

Masterarbeit

LERNMANAGEMENTSYSTEM FÜR DIE FEUERWEHRJUGEND DES LANDESFEUERWEHRVERBANDES STEIERMARK

ausgeführt am



Fachhochschul-Masterstudiengang
Automatisierungstechnik-Wirtschaft

von

Ing. Christoph Franz Beichler, BSc
2010321001

betreut und begutachtet von
DI (FH) Gernot Hofer

Graz, November 2024



Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.



Unterschrift

DANKSAGUNG

Mein tiefster Dank gilt meiner wunderbaren Frau Katharina Beichler, deren bedingungslose Unterstützung und endlose Geduld mich durch die Höhen und Tiefen dieser Masterarbeit getragen haben. Ihre Liebe und Ermutigung waren die Grundpfeiler, auf denen ich diese spannende Reise aufgebaut habe.

Ein besonderer Dank gebührt auch Herrn DI (FH) Gernot Hofer, der mir nicht nur als Betreuer zur Seite stand, sondern dessen Fachwissen und fachliche Anleitung entscheidend dazu beigetragen haben, diese Arbeit auf ein höheres Niveau zu heben. Seine Hingabe zu diesem Thema und seine Bereitschaft, mich zu unterstützen, waren von unschätzbarem Wert.

Ein herzliches Dankeschön geht auch an Frau Christina Egger, BEd für ihren wertvollen Beitrag zum didaktischen Teil meiner Arbeit. Ihre Einsichten und Ratschläge haben mir geholfen, meine Konzepte zu verfeinern und meine Ziele klarer zu definieren.

Ein weiterer Dank gilt meinen geschätzten Kommilitonen, insbesondere Herrn Larcher Markus, BSc, dessen gemeinsame Arbeit und kollegiale Begleitung mir nicht nur geholfen haben, fachliche Hürden zu überwinden, sondern den Prozess der Masterarbeit zu einer bereichernden und erfüllenden Erfahrung gemacht haben.

Nicht zuletzt möchte ich auch allen anderen Freunden, Familienmitgliedern und Mentoren danken, die mich auf meinem akademischen Weg begleitet und ermutigt haben. Ihr moralischer Beistand und die Ermutigungen haben mir in Zeiten der Unsicherheit und des Zweifels Kraft und Zuversicht gegeben.

Ohne die Hilfe und Unterstützung dieser wunderbaren Menschen wäre diese Arbeit nicht möglich gewesen. Ihnen allen gebührt mein aufrichtiger Dank.

KURZFASSUNG

Der Schwerpunkt dieser Masterarbeit liegt auf der Entwicklung eines Lernmanagementsystems für die Jugend des Landesfeuerwehrverbandes Steiermark. Ziel ist es, eine webbasierte Plattform zu schaffen, welche die Mitglieder der Jugendfeuerwehr während ihrer Lerntätigkeiten unterstützt.

Durch den Einsatz moderner Didaktik und der Implementierung einer adäquaten Methodik soll das Lernerlebnis auf dem Niveau des Wissenstests in allen drei Stufen (Bronze, Silber und Gold) verbessert werden. Über die Lernfunktion können die Jugendlichen über jedes internetfähige Endgerät ihr Wissen aneignen. Mit der Wettkampf-Funktion können die Jung-Florianis die erlernten Kenntnisse spielerisch mit ihren Kameraden*innen unter Beweis stellen. Darüber hinaus wird ein eigenes Bewertungstool für die engagierten Ausbilder implementiert, mit dessen Hilfe laufend auf die vorhandenen Lerndefizite der Jugendmitglieder eingegangen werden kann.

Zusätzlich musste zum Grundkonzept der Plattform die Datenbank des Landesfeuerwehrverbandes Steiermark eingebunden werden, um eine nicht-passwortgebundene Anmeldung zu realisieren.

Das Resultat ist eine lauffähige Lernplattform, welche auf einem lokalen Server integriert wurde und alle essenziellen Themengebiete abdeckt.

ABSTRACT

The focus of this master's thesis is the development of a learn-management-system for the youth of the Styrian State Fire Brigade Association. The goal is to create a web-based platform that supports junior fire brigade members in their learning activities.

By utilizing modern didactics and implementing an appropriate methodology, the learning experience for the Wissenstest across all three levels (bronze, silver, gold) is intended to be enhanced. Through the learning function, young members can expand their knowledge on any internet-enabled device. The competition function allows the young firefighters to demonstrate their newly acquired knowledge in a playful way with their peers. Furthermore, a dedicated evaluation tool has been implemented for committed instructors, enabling them to continuously address any learning gaps among their youth members.

Additionally, the platform's core concept required the integration of the Styrian Firefighter's Association's database to facilitate a non-password-based login.

The result is a functional website that is currently integrated on a local server covering all essential subject areas.

INHALTSVERZEICHNIS

1	Einleitung.....	1
2	Österreichischer Bundesfeuerwehrverband	2
2.1	Organisation und Gliederung	2
2.2	Landesfeuerwehrverband Steiermark	3
2.2.1	Organisation.....	3
2.2.2	Aufgaben.....	4
2.3	Sachgebiet Feuerwehrjugend.....	4
2.4	Wissenstest.....	5
2.4.1	Wissenstest Stufe Bronze.....	5
2.4.2	Wissenstest Stufe Silber	6
2.4.3	Wissenstest Stufe Gold	6
3	Didaktik und Methodik	7
3.1	Rechtsgrundlage zur digitalen Grundbildung	7
3.2	Publikationen im Feuerwehrwesen.....	8
3.3	Die methodischen Prinzipien im Lernmanagementsystem	9
4	Datenbanken	11
4.1	Feuerwehrdateninformationssystem und Katastrophenschutzmanagement	12
4.1.1	Funktionäre*innen im Bereich Jugend.....	13
4.1.2	Jugend	13
4.1.3	Fahrzeuge und Stützpunkte.....	13
4.2	MySQL™	15
4.2.1	Datenbankverknüpfung.....	16
4.2.2	Tabelle: users	16
4.2.3	Tabelle: mappingtable_wt.....	17
4.2.4	Tabelle: mappingtable_user	18
4.2.5	Tabelle: user_progress	18
4.2.6	Tabelle: vehicles_bases	19
5	Websites und das Internet	20
5.1	Domain.....	20
5.2	Website und Webseite	20
5.3	Host, Datenbank-Host und Server.....	21
5.4	FTP-Server	22
5.5	HTTP und HTTPS.....	22
5.6	Cookies	23
6	Die Lernmanagementsystem Website	25
6.1	WordPress®	25
6.1.1	Hauptvarianten von WordPress.....	25
6.1.2	Aufbau und Struktur von WordPress	26
6.1.3	Benutzerverwaltung	27

6.1.4	Medienverwaltung.....	27
6.1.5	Sicherheit und Wartung	28
6.1.6	Vor- und Nachteile von WordPress	28
6.2	Content-Management-System.....	29
6.3	Datenschutzgrundverordnung	29
6.3.1	DSGVO Artikel 30.....	30
6.3.2	DSGVO Artikel 32.....	30
6.3.3	DSGVO Artikel 33, 34, 35 und 36.....	31
6.4	Lokale Entwicklungsumgebung	31
6.4.1	Local®.....	31
6.4.2	PhpStorm®	33
6.5	PHP und HTML.....	34
6.5.1	Hypertext Preprocessor (PHP)	34
6.5.2	Hypertext Markup Language (HTML)	35
6.6	Plugin und Skripte.....	36
6.6.1	Plugin: Benutzeranmeldung.....	36
6.6.1.1	Anmeldekriterien.....	36
6.6.1.2	Funktionsbeschreibung der Benutzeranmeldung.....	37
6.6.2	Skript: Datenbank Handshake	41
6.6.2.1	Funktionsbeschreibung des Datenbank Handshake.....	42
6.6.2.2	Vigenère-Verschlüsselung.....	44
6.6.3	Skript: Lernmodus.....	46
6.6.4	Skript: Prüfungsmodus	50
6.6.4.1	Funktionsbeschreibung des Prüfungsmodus	50
6.6.4.2	Beschreibung des HTML-Anteils im Modul Prüfungsmodus.....	53
6.6.5	Skript: Auswertung.....	54
6.6.5.1	Funktionsbeschreibung der Auswertung für Jugendliche	54
6.6.5.2	Funktionsbeschreibung der Auswertung für Beauftragte und Funktionäre*innen.....	54
6.6.5.3	Vigenère-Entschlüsselung.....	56
6.7	Aufbau der Website	57
6.7.1	Einstiegsseite.....	57
6.7.2	Benutzeranmeldung.....	59
6.7.3	Dashboard Jugend	60
6.7.3.1	Lernmodus.....	60
6.7.3.2	Prüfungsmodus	62
6.7.3.3	Ergebnisse der Station	63
6.7.4	Dashboard Funktionäre	64
6.7.4.1	Auswertung Übersicht	64
6.7.4.2	Auswertung Einzelne.....	65
7	Tests.....	66
8	Rückmeldungen	67

8.1	Feedback der Feuerwehrjugend der FF Tillmitsch	67
8.2	Einarbeitung des Feedbacks	68
9	Ergebnisse und Ausblick	69
	Literaturverzeichnis	70
	Abbildungsverzeichnis.....	73
	Tabellenverzeichnis	77
	Abkürzungsverzeichnis.....	78
	Anhang 1: Quellcode Benutzeranmeldung (Plugin)	79
	Anhang 2: Quellcode Datenbank Handshake	83
	Anhang 3: Quellcode Lernmodus für die drei Stufen Bronze, Silber und Gold – Beispiel für Bronze	85
	Anhang 4: Quellcode Prüfungsmodus für die drei Stufen	92
	Anhang 5: Quellcode Auswertung Jugendliche.....	97
	Anhang 6: Quellcode Auswertung Beauftragte und Funktionäre*innen.....	101

1 EINLEITUNG

Die Digitalisierung schreitet in allen Lebensbereichen der Menschheit schnell voran. Sie soll den Alltag der Menschen erleichtern, Abläufe automatisieren und den Zugriff auf Daten orts- und zeitunabhängig gestalten. Während beinahe alle Bildungsbereiche bereits digitale Lehrmittel verwenden, ist die Digitalisierung in der Feuerwehrjugend des Landesfeuerwehrverbandes Steiermark noch nicht stark ausgeprägt. Um die Inhalte für den Wissenstest zu lehren, wird hier seit eh und je auf Druckwerke zurückgegriffen. Damit es zukünftig eine Alternative gibt und die Arbeit im Jugendbereich wieder attraktiver und moderner gestaltet werden kann, soll eine digitale Plattform erstellt werden. Dieses Lernsystem soll den Jugendlichen einen zeitgemäßen Zugang zu Wissen und interaktiven Lernmethoden bieten.

Augenscheinlich gibt es schier unendlich viele Möglichkeiten, wie solch eine Applikation umgesetzt werden kann. Werden aber alle notwendigen Aspekte betrachtet, so wird klar, dass sich die Verwirklichung des Projekts als sehr komplex gestaltet. Eine Herausforderung ergibt sich aus der Vielfalt an Endgeräten und deren unterschiedlichen Betriebssystemen. Es müssten für eine Smartphone-App zumindest zwei Betriebssysteme (Android und iOS) bedient werden. Eine weitere Problematik stellt die Integration der unterschiedlichen Datenbanken und die damit verbundenen Automatisierungsaufgaben dar. Es sollen so viele Informationen wie möglich aus der Datenbank der Feuerwehr automatisch in der Plattform verwendet werden. Zusätzlich soll auch eine nicht-passwortbasierte Anmeldung etabliert werden, damit einem "vergessenen Passwort" effektiv entgegengewirkt werden kann.

Um die Herausforderung mit den unterschiedlichen Betriebssystemen zu umgehen, soll eine Website als zentrale Plattform erstellt werden. Auf dieser Lernplattform sollen sich nur Mitglieder der Feuerwehrjugend und deren Funktionäre anmelden können. Für die Authentisierung soll ein eigenes Verfahren erstellt werden. Ein Lernmodus, mit dem die Fragen der jeweiligen Wissensteststufe geübt werden können und ein Prüfungsmodus, mit dem das Erlernte überprüft werden kann, soll implementiert werden. Für die Jugendbetreuer*innen soll ein Evaluierungstool entwickelt werden, mit dem die Lerndefizite der Jugendlichen besser identifiziert werden können. Es sei an dieser Stelle noch erwähnt, dass dieses Lernsystem modular aufgebaut werden soll, damit es jederzeit erweiterbar ist, um auf zukünftige Anforderungen reagieren zu können.

Im ersten Teil dieser Arbeit werden das österreichische und steirische Feuerwehrwesen genauer erklärt. Es folgt ein Einblick in das Themengebiet der Didaktik und Methodik. Am Ende des ersten Segments werden die technischen Grundlagen, die für die Erstellung einer internetbasierten Lernplattform notwendig sind, genauer erläutert. Der zweite Teil der Arbeit beschreibt die praktische Umsetzung des Systems. Im Detail geht es hier um die Entwicklung der verwendeten Plugins und den Aufbau der Website. Den Abschluss dieser Arbeit bilden die Testergebnisse und Rückmeldung der Testserien und ein Ausblick in die Zukunft dieser Lernplattform.

2 ÖSTERREICHISCHER BUNDESFEUERWEHRVERBAND

Der Österreichische Bundesfeuerwehrverband (ÖBFV) mit dem Hauptsitz in Wien wurde im Jahr 1889 gegründet und ist ein rechtsfähiger Verein. Die Ziele des ÖBFV umfassen die Vereinheitlichung der Organisation, Ausrüstung und Ausbildung der österreichischen Feuerwehren. Dabei wird die föderale Struktur der semipräsidentiellen Republik Österreich berücksichtigt. Zudem vertritt der ÖBFV die Interessen der Feuerwehren, die über die Grenzen eines Bundeslandes und des Bundes hinausgehen.¹

2.1 Organisation und Gliederung

Das Präsidium des ÖBFV setzt sich aus dem Präsidenten*in, den Vizepräsidenten*innen und den Landesfeuerwehrkommandanten*innen zusammen. Mit der Unterstützung der Referatsleiter*innen und den Vorsitzenden der Fachausschüsse berät die Führungsspitze über grundlegende Angelegenheiten des Feuerwehrwesens und arbeitet an deren Lösung. Angeführt wird das Präsidium vom Feuerwehrpräsidenten Robert Mayer, MSc.

Wie bereits einleitend erwähnt, organisiert sich der ÖBFV föderalistisch. Das bedeutet, dass sich jeder Landesfeuerwehrverband (LFV) selbst verwalten muss und selbst entscheiden kann, welche Rechtsform er in Anbetracht der länderspezifischen Gesetze annimmt.

In der Republik Österreich werden etwa 99% der Einsätze durch freiwillige Feuerwehren bewältigt, wobei 77% dieser Einsätze ohne Bezahlung erfolgen. In Europa steht Österreich dadurch unangefochten auf dem ersten Platz. Etwa jede*r 33. Österreicher*in ist Mitglied einer freiwilligen Feuerwehr.

Aufgeteilt sind die insgesamt 4774 Feuerwehren des ÖBFV, wie in Tabelle 1 ersichtlich, auf sechs Berufsfeuerwehren (BF), 313 Betriebsfeuerwehren und 4.455 freiwillige Feuerwehren. Die Berufsfeuerwehren befinden sich in Wien, Linz, Graz, Klagenfurt, Salzburg und Innsbruck.²

Feuerwehren	
Gesamtanzahl	4.774
Aufteilung	
Berufsfeuerwehren	6
Betriebsfeuerwehren	313
Freiwillige Feuerwehren	4.455

Tabelle 1: Übersicht Feuerwehren in Österreich, Quelle: Eigene Darstellung.

¹ Vgl. Österreichischer Bundesfeuerwehrverband (2022): Online-Quelle [19.05.2024].

² Vgl. Österreichischer Bundesfeuerwehrverband (2018): Online-Quelle [19.05.2024].

In der Statistik des ÖBFVs waren mit Ende 2023, 353.798 Mitglieder bestehend aus Aktivstand, Reserve und Feuerwehrjugend (FJ), gemeldet. Mittlerweile beträgt die Frauenquote etwa 9,5%, was einen Zuwachs von 170% im Vergleich zum Jahr 2008 darstellt. Im Jahr 2023 wurden von den Kameraden*innen 303.666 Einsätze erfolgreich gemeistert.³

2.2 Landesfeuerwehrverband Steiermark

Der Landesfeuerwehrverband Steiermark und dessen einzelne Feuerwehren besitzen die Rechtsform einer Körperschaft des öffentlichen Rechts. Dies bedeutet, dass ihre Existenzen gesetzlich verankert sind, da sie behördliche Aufgaben erfüllen und zusätzlich im Einsatzfall mit hoheitlichen Befugnissen ausgestattet sind. Dazu zählt als Beispiel auch, dass Feuerwehrmitglieder während der Ausübung ihrer Pflicht gemäß § 74 Z 4a Strafgesetzbuch (StGB) verbeamtet sind.⁴

Im Jahr 2023 wurden die Mitglieder des LFV Steiermark zu 51.699 von den österreichweiten 303.666 Einsätzen gerufen.

2.2.1 Organisation

Der LFV Steiermark organisiert sich in drei untergeordnete Verbände. Die 17 Bereichsfeuerwehrverbände (BFV) stellen die einzelnen alten Bezirke des Bundeslandes Steiermark dar. Es wurde hier bewusst auf eine Zusammenlegung der BFV im Vergleich zu den Bezirken in den Jahren 2012 und 2013 verzichtet, da die bestehenden Organisationen mit deren Infrastrukturen verändert werden hätten müssen (siehe Abbildung 1).

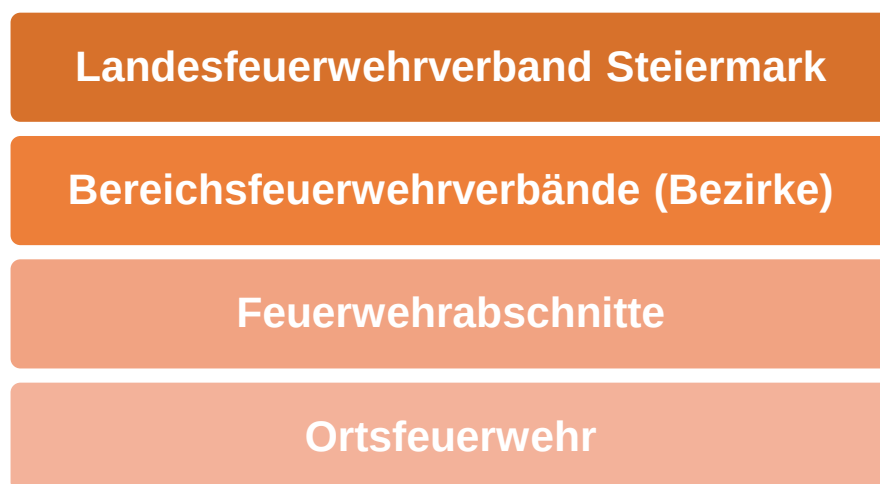


Abbildung 1: Organisation des Landesfeuerwehrverbandes Steiermark, Quelle: Eigene Darstellung.

³ Vgl. Österreichischer Bundesfeuerwehrverband (2023): Online-Quelle [19.05.2024].

⁴ Vgl. Johannes Kepler Universität Linz (2021): Online-Quelle [19.05.2024].

Eine Ortsfeuerwehr wird von den Offizieren dem*der Kommandanten*in und dessen*deren Stellvertreter*in generell vertreten (siehe Abbildung 2). Gemeinsam mit dem Feuerwehrausschuss, welcher aus Vertretern*innen der unterschiedlichen Sachgebiete besteht, bilden sie die Führung der ortsansässigen Feuerwehr und tragen die Verantwortung für alle internen Entscheidungen. Die Wahl des Kommandos sowie die Bestellung des Feuerwehrausschusses erfolgt regulär alle fünf Jahre bei einer Jahreshauptversammlung.

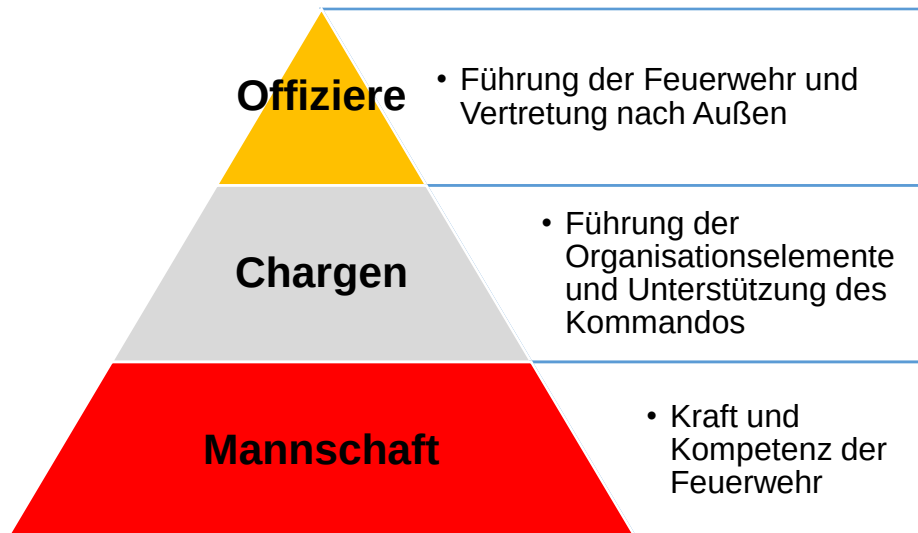


Abbildung 2: Hierarchiestruktur der österreichischen Feuerwehren, Quelle: Eigene Darstellung.

2.2.2 Aufgaben

Die Freiwillige Feuerwehr ist dafür zuständig, Brände zu bekämpfen und bei der Verhinderung dieser mitzuwirken. Außerdem ist sie verantwortlich für den Schutz vor anderen Gefahren, die lokal oder überregional auftreten und die Allgemeinheit einzelne Personen, Tiere, Sachwerte oder die Umwelt bedrohen könnten. Nicht nur Brandeinsätze und technische Hilfestellungen nach Verkehrsunfällen, sondern auch Spezialaufgaben wie zum Beispiel der Strahlenschutz, die Menschenrettung und Absturzsicherung, der Flugdienst oder der Tauchdienst zählen zu den Einsatzgebieten der freiwilligen Feuerwehren in Österreich. Diese sehr vielfältigen Aufgabengebiete erfordern eine ausgezeichnete und intensive Ausbildung, welche teilweise in der eigenen Feuerwehr, hauptsächlich aber in der Feuerwehr- und Zivilschule (FWZS) in Lebring-Sankt Margarethen erfolgen.

2.3 Sachgebiet Feuerwehrjugend

In diesem Kapitel soll nun näher auf das Sachgebiet der Feuerwehrjugend des Bundeslandes Steiermark eingegangen werden. Bereits ab einem Alter von zehn Jahren kann der ortsansässigen Feuerwehrjugend beigetreten werden. Grundvoraussetzung sind eine medizinische Tauglichkeitsuntersuchung sowie eine Abstimmung über die Aufnahme bei einer ordentlichen Ausschusssitzung. Bis zum Beginn des 16. Lebensjahres bleibt der*die Jugendliche in der Feuerwehrjugend, wobei ein Aufrücken in den Aktivstand der Feuerwehr erst durch einen Beschluss des Feuerwehrausschusses genehmigt oder aber auch abgelehnt werden kann. Neben den allgemeinen Übungen und diversen Veranstaltungen gibt es eine Menge für die Jung-Florianis zu tun. Jedes Jahr werden den Jugendlichen zahlreiche Wettbewerbe

angeboten. Für die Zehn bis Zwölfjährigen gibt es das sogenannte Wissenstest-Spiel in den Stufen Bronze und Silber. Bei diesem Bewerb werden den jüngsten Florianis die rudimentären Grundlagen des Feuerwehrwesens nähergebracht.⁵ Den Kern der Jugendausbildung bildet der in Kapitel 2.4 beschriebene Wissenstest mit seinen begehrten Abzeichen in den jeweiligen Stufen (siehe Abbildung 3).

Neben diesen beiden Wettbewerben gibt es zusätzlich noch das Feuerwehrjugendleistungsabzeichen (FJLA) in den Stufen Bronze, Silber und Gold und das Jugend-Funkleistungsabzeichen.

2.4 Wissenstest

Der Wissenstest der Feuerwehrjugend ist der Grundstein der Jugendausbildung im Feuerwehrwesen. Ab dem zwölften Lebensjahr können die Jugendlichen sich jährlich dem Wissenstest stellen und dabei schrittweise ihr Wissen in folgenden Gebieten überprüfen:

- Organisation der Feuerwehr
- Formalexerzieren
- Fahrzeug- und Gerätekunde
- Kenntnisse der Dienstgrade
- Warn- und Alarmsysteme
- Nachrichtendienst
- Einsatz- und Dienstbekleidung
- Knotenkunde

Die genannten Bereiche sind in den folgenden Wissenstest Stufen auf verschiedene Weise verteilt. Zum einen unterteilen sich die Klassen auf Orts-, Bereichs- und Landesebene und zum anderen verteilen sich die Themengebiete nach Komplexität des zu behandelten Sachgebietes.

2.4.1 Wissenstest Stufe Bronze

Im Fokus des Wissenstests der Stufe Bronze stehen im Allgemeinen das Kennenlernen der eigenen Feuerwehr und die Grundlagen des Feuerwehrwesens. Inhaltlich werden verschiedene Themen behandelt, die für die Arbeit der Feuerwehr wichtig sind. Dazu gehört das Wissen über die Hierarchie innerhalb der Feuerwehr, also wer welche Position innehat und welche Aufgaben dabei erfüllt werden. Auch die Verwendung und Funktionsweisen von Fahrzeugen und Ausrüstungen, wie zum Beispiel Feuerlöschern, wird erklärt, damit die Jugendlichen verstehen, wie sie im Ernstfall eingesetzt werden. Ein weiterer Bestandteil des Wissenstests der Stufe Bronze ist das Erlernen von praktischen Fähigkeiten, wie dem Binden von Knoten oder dem Erkennen wichtiger Warnzeichen, die in Gefahrensituationen verwendet werden. Zudem wird den Jugendlichen beigebracht, wie sie sich in Notfällen richtig verhalten, zum Beispiel bei einem Brand oder wenn sie Hilfe über den Notruf anfordern müssen.⁶

⁵ Vgl. Landesfeuerwehrverband Steiermark (2023): Online-Quelle [19.05.2024].

⁶ Vgl. Landesfeuerwehrverband Steiermark (2016): Online-Quelle [19.05.2024].

2.4.2 Wissenstest Stufe Silber

Nachdem der Wissenstest der Stufe Bronze erfolgreich abgelegt wurde, können sich die Jugendlichen im Folgejahr dem Wissenstest der Stufe Silber stellen. Aufbauend auf dem erlernten Wissen aus der Stufe Bronze wird nun die Organisation der Feuerwehr auf Abschnitts- und Bereichsebene gelehrt. Zudem werden die Kenntnisse in den Bereichen Dienstgrade, Gerätekunde, Knotenkunde und Formalexerzieren weiter vertieft. Ergänzend kommt das Fachgebiet Nachrichtendienst hinzu, bei dem die Jugendlichen lernen, welche Funksysteme und Geräte verwendet werden und wie die Funkordnung im steirischen LFV organisiert ist.⁷

2.4.3 Wissenstest Stufe Gold

Der Wissenstest der Stufe Gold stellt die Königsdisziplin in der Feuerwehrjugend dar. Nach erfolgreicher Absolvierung des Wissenstests der Stufe Silber kann im darauffolgenden Jahr das begehrte Abzeichen in Gold erworben werden. In dieser Stufe werden die Jugendlichen, welche sich dann bereits im Alter zwischen 14 und 15 Jahren befinden, auf den aktiven Dienst bei der Feuerwehr vorbereitet. Der Fokus des Wissenstests der Stufe Gold richtet sich auf die Organisation der Feuerwehr auf Landesebene und einsatzrelevanter Grundlagen.⁸



Abbildung 3: Wissenstestabzeichen in der Stufe Bronze, Silber und Gold,
Quelle: Niederösterreichischer Landesfeuerwehrverband (o.J.), Online-Quelle [19.05.2024].

⁷ Vgl. Landesfeuerwehrverband Steiermark (2016): Online-Quelle [19.05.2024].

⁸ Vgl. Landesfeuerwehrverband Steiermark (2016): Online-Quelle [19.05.2024].

3 DIDAKTIK UND METHODIK

Didaktik und Methodik sind zwei zentrale Begriffe, die eng miteinander verbunden sind, wenn es um das Lehren und Lernen geht. Dabei beschäftigt sich die Didaktik mit den Inhalten und den Zielsetzungen der zu lehrenden Themengebiete. Es wird mithilfe von Planung und Strukturierung des Lernprozesses überlegt, welcher konkrete Content den Lernenden vermittelt werden und welche Ziele damit erreicht werden sollen. Bei der Didaktik wird immer nach dem “Was” und “Warum” gefragt.⁹

Die Methodik beschäftigt sich dagegen mit der praktischen Umsetzung der zu erlernenden Inhalte. Der Fokus liegt dabei auf den verschiedenen Methoden und Techniken, mit denen der Lernstoff optimal vermittelt werden kann, um den Lernprozess bestmöglich zu unterstützen. Hier gibt es die unterschiedlichsten Formen wie zum Beispiel Vorträge, Gruppenarbeiten oder den Einsatz von digitalen Medien. Im Vergleich zur Didaktik wird bei der Methodik nach dem “Wie” gefragt.¹⁰

Nachdem die Inhalte des Wissenstests in den jeweiligen Stufen vom LFV Steiermark fixiert sind, konnte in dieser Arbeit nicht weiter auf die didaktischen Themen eingegangen werden. Eine Überarbeitung in diesem Bereich müsste über den zuständigen Arbeitskreis im LFV Steiermark erfolgen.

3.1 Rechtsgrundlage zur digitalen Grundbildung

Bereits am 06. Juli 2022 wurde vom Bundesministerium für Bildung, Wissenschaft und Forschung eine Verordnung in Form eines Bundesgesetzblattes (BGBlA) erlassen, in welcher die digitale Grundbildung genauestens definiert wurde. Alle Schüler*innen, welche sich im Alter zwischen zehn und fünfzehn Jahren befinden, werden unter der Sekundarstufe 1 zusammengefasst (siehe Abbildung 4). Somit können auch

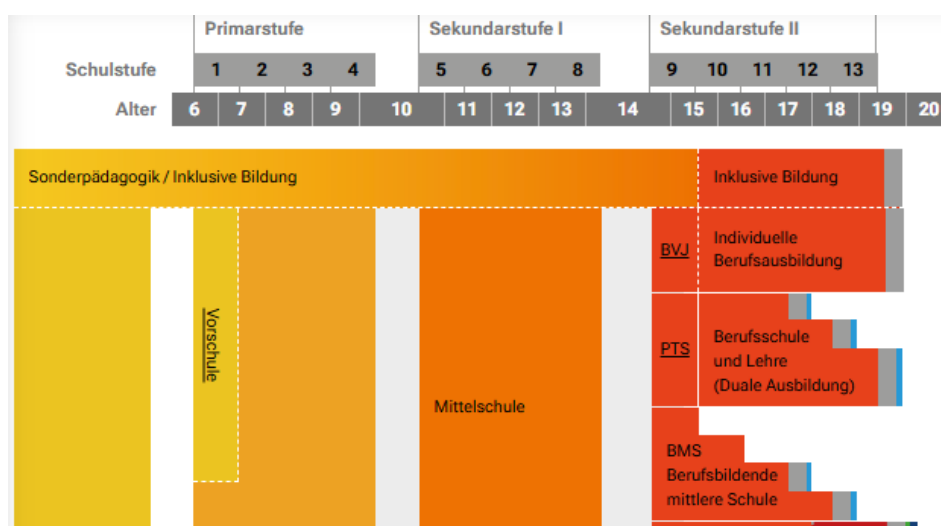


Abbildung 4: Ausschnitt aus dem österreichischen Bildungssystem,
Quelle: Agentur für Bildung und Internationalisierung (o.J.), Online-Quelle [30.10.2024].

⁹ Vgl. Studyflix GmbH (o.J.): Online Quelle [12.01.2025]

¹⁰ Vgl. Cornelsen Verlag (o.J.): Online Quelle [12.01.2025]

die Jung-Florianis dieser Stufe zugeordnet werden und es sollen Teile dieser Grundsätze im Lernmanagementsystem verwendet werden.

Ziel der digitalen Grundbildung ist, dass die Jugendlichen einen verantwortungsvollen Umgang mit den technologischen Entwicklungen erlernen. Zusätzlich soll ihnen auch eine kritische Denkweise gegenüber dieser digital geprägten Welt beigebracht werden.

Ein wichtiger Schritt ist die Verwendung von modernen didaktischen Grundsätzen. Durch den Einsatz von interaktiven und prozessualen Lernmethoden wie dem forschenden Lernen, dem spielerischen Lernen und dem kritischen Denken soll dieser Unterricht zukünftig fächerübergreifend gestaltet werden.¹¹

Bei der Umsetzung des LMSs soll überprüft werden, ob es möglich ist, Elemente dieser Grundlagen zu integrieren. Dies könnte sich jedoch schwierig gestalten, da der Wissenstest in seiner eigentlichen Form (Art der Fragestellung, Aufbau, usw.) nicht verändert werden darf. In Anbetracht dessen kann hier am ehesten auf das spielerische Lernen der Gamification zurückgegriffen werden.

3.2 Publikationen im Feuerwehrwesen

Im Zuge dieser Arbeit wurde versucht, Informationen zur Digitalisierung und den angewandten Lernvarianten bei Feuerwehren in und außerhalb Europas zu finden. Leider kann neben den klassischen Druckwerken wie dem "The Firefighter's Handbook", welches als Referenzwerk für Feuerwehren in den USA, dem Vereinigten Königreich, Australien, Kanada, Mexiko, Singapur und Spanien angepriesen wird, nicht viel gefunden werden. Der Begriff Handbuch ist dabei eher eine Untertreibung, da hier Wissen auf knapp 1000 Seiten vermittelt wird.¹²

Vom Schweizer Kanton Schaffhausen gibt es immerhin einen Leitfaden für die Feuerwehrausbildung. In diesem Nachschlagewerk werden Verantwortlichkeiten geregelt sowie allgemeine Grundsätze, Elemente der Ausbildung und Ausbildungsmethodiken genauer erklärt.¹³ Es wird aber auch in der Schweiz nach wie vor immer im klassischen Stile Wissen vermittelt.

Was jedoch im Allgemeinen fehlt, ist der Zugang zur digitalen Ausbildung im Feuerwehrwesen. Es konnte nur eine einzige Publikation aus dem Jahr 2017 zu diesem Thema gefunden werden. Dabei wurde eine fünfjährige Studie zur Umsetzung eines Online-Lernprogramms in der schwedischen Feuerwehrausbildung analysiert. Mithilfe dieser Studie konnten zwei Phasen, die Implementierungsphase und die Verarbeitungsphase bei der Verwendung von digitalen Lernplattformen identifiziert werden. In der Implementierungsphase zeigte sich ein deutlicher Widerspruch in Bezug auf die Interpretation der Ausbildungsziele. Dies führte die Ausbilder*innen zu einem kritischen Überdenken ihrer früheren Überzeugungen in Anbetracht des Kursdesigns. Durch die Unterstützung digitaler Technologien konnten theoretischere, prozessorientiertere Ansätze, die das Kursformat verbesserten, entwickelt werden. In der Verarbeitungsphase gab es jedoch einige Konflikte und Dilemmata, wodurch die zuvor getroffenen

¹¹ Vgl. Rechtsinformationssystem des Bundes (2022): Online-Quelle [20.05.2024].

¹² Vgl. Next Step NH (2020): Online-Quelle [20.05.2024].

¹³ Vgl. Feuerwehr Neuhaus (2014): Online-Quelle [23.11.2024].

Veränderungen teilweise wieder zurückgezogen wurden. Diese Problematik zeigte, dass die Digitalisierung in dieser Phase ohne ausreichende Unterstützung durch die Ausbilder*innen und deren Einschulung zum Scheitern verurteilt ist. Die Conclusio der Studie ist, dass sehr hohe Aufwände für die dauerhafte Integration einer Lernplattform nötig sind, um alle auftretenden Konflikte positiv lösen zu können.¹⁴

3.3 Die methodischen Prinzipien im Lernmanagementsystem

Als grundlegenden methodischen Prinzipien gibt es folgende drei bekannte äußerst wirkungsvolle Arten:¹⁵

- Prinzip der Anschaulichkeit
- Prinzip der Wiederholung
- Prinzip der Praxisnähe

Das Prinzip der Anschaulichkeit findet seinen größten Anwendungsbereich in der Knotenkunde. Initial wird diese Thematik den Jugendlichen nur über sehr schlechte Abbildungen nähergebracht (siehe Abbildung 5).

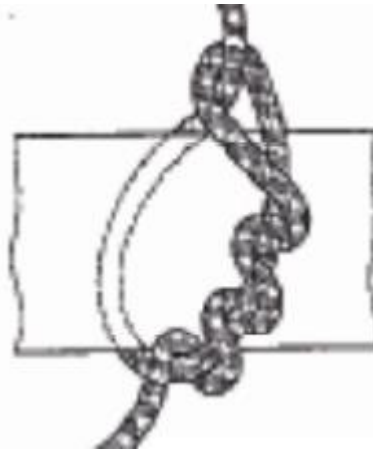


Abbildung 5: Originaldarstellung des Zimmermannsklank (Zimmermannsstek) aus dem Wissenstest Silber, Quelle: Eigene Darstellung.

Erst durch das praktische Mitwirken der Jugendbetreuer*innen und Ortsjugendbeauftragten kann das Anlegen der Knoten verinnerlicht werden. An dieser Stelle sollen Lernvideos, welche das Anfertigen des jeweiligen Knotens zeigen und über die Tonspur dessen Verwendung erläutern, ins Spiel kommen.

Mithilfe des Evaluierungstools im Lernmanagementsystem wird das Prinzip der Wiederholung angewendet. Die zuständigen Beauftragten können mit Hilfe des Tools laufend die Ergebnisse der Jung-Florianis verfolgen und zur Wiederholung animieren. Die ständige Wiederholung sorgt dafür, dass das Gelernte nicht nur kurzfristig im Gedächtnis bleibt, sondern langfristig abrufbar wird.

Eine Methode, die bereits fest in der Ausbildung der Jugendlichen verankert ist, ist das Prinzip der Praxisnähe. So werden zum Beispiel bei der Gerätekunde die Mechanismen und Funktionen der jeweiligen Objekte direkt am Gerät erklärt. Einsatztechnische Themen werden den Jung-Florianis durch realitätsnahe

¹⁴ Vgl. Taylor & Francis Online (2018): Online-Quelle [20.05.2024].

¹⁵ Vgl. Studyflix GmbH (o.J.): Online-Quelle [08.06.2024].

Praxisübungen gelehrt. Im LMS wird auch abseits der Knotenkunde auf Lernvideos zurückgegriffen, damit die Jugendlichen diese Themengebiete auch von zu Hause aus lernen können.

4 DATENBANKEN

In den 1960er-Jahren wurde das Konzept der Datenverwaltung durch eine separate Softwareschicht zwischen Betriebssystem und Anwendungsprogramm eingeführt, da Probleme bei der Verarbeitung von Daten in einfachen Dateien auftraten. Dieses Konzept stieß darauf, dass Datenspeicher in Form von Dateien normalerweise für eine bestimmte Anwendung bestimmt waren und dass das Umkopieren, Mischen und Restrukturieren von Dateien ein beträchtlicher Teil des täglichen Geschäfts darstellten.¹⁶

Eine Datenbank bezeichnet eine strukturierte Ansammlung von Daten (siehe Abbildung 6), die üblicherweise in einem Computersystem elektronisch gespeichert werden. In der Regel wird eine Datenbank durch ein Datenbank-Management-System (DBMS) verwaltet. Die Daten, das DBMS und die dazugehörigen Anwendungen werden gemeinsam als Datenbanksystem bezeichnet. Dieses ist häufig auf die bloße Datenbank beschränkt. Typischerweise werden Daten in den am weitesten verbreiteten Datenbanktypen in Zeilen und Spalten in mehreren Tabellen modelliert, um eine effiziente Verarbeitung und Datenabfrage zu ermöglichen. Dann ist es möglich, die Informationen problemlos abzurufen, zu verwalten, zu modifizieren, zu aktualisieren, zu kontrollieren und zu organisieren. Die Structured Query Language (SQL) wird von den meisten Datenbanken genutzt, um Daten zu schreiben und abzufragen.¹⁷

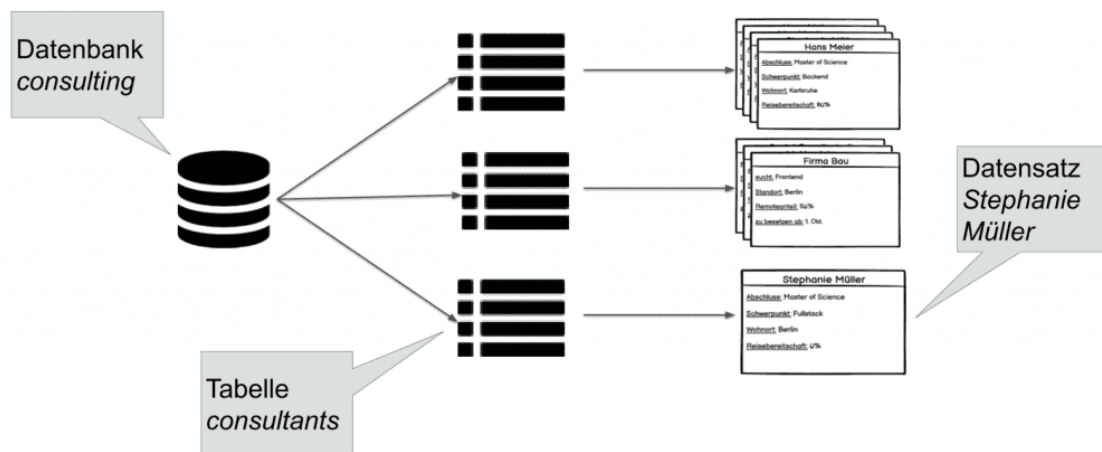


Abbildung 6: Beispielhafte Darstellung der Datenbankstruktur,
Quelle: BCM Solutions (o.J.), Online-Quelle [08.06.2024].

In das LMS wurde zum einen das veraltete Datenbanksystem der Feuerwehr, das Feuerwehrdateninformationssystem und Katastrophenschutzmanagement (FDISK) integriert und zum anderen eine neue Datenbank für die Einbindung der Fragen mit MySQL kreiert.

¹⁶ Vgl. Oracle Corporation (2020): Online-Quelle [09.06.2024].

¹⁷ Vgl. Oracle Corporation (2020): Online-Quelle [09.06.2024].

4.1 Feuerwehrdateninformationssystem und Katastrophenschutzmanagement

Bereits 1977 wurden die ersten Verwaltungstools bei einigen Feuerwehren in Österreich eingeführt, jedoch wurden diese nur lokal betrieben, da es zu diesem Zeitpunkt noch keine flächendeckende IT-Infrastruktur gab. Im Jahr 1993 wurde dann das erste überwiegend zentral verwendete Verwaltungssystem, das sogenannte "Notruf – 122", eingeführt. Schon damals gab es die Möglichkeit, die zentrale Datenbank durch eine Modem-Einwahlverbindung mit der Datenbank des Landesfeuerwehrkommandos zu synchronisieren.

Im Jahr 2002 gelang es dann, eine Online-Plattform in das Feuerwehrwesen zu integrieren, welche alle Teilbereiche der Feuerwehrverwaltung abdeckt. Das Feuerwehrdateninformationssystem und Katastrophenschutzmanagement ist seitdem nicht mehr wegzudenken, da es von der Mitgliederverwaltung über die Einsatzdokumentation bis hin zu den Katastrophenplänen und vielem mehr alles in Echtzeit verwalten kann (siehe Abbildung 7).¹⁸

The screenshot shows the FDISK web application. The top header features a fire truck image and the FDISK logo. The left sidebar contains a navigation menu with categories like HOME, FW, BEWERBSVERWALTUNG, GLOBALE SUCHE, JUGENDLAGER, LOGINS, RECHNUNGSWESEN, and WIKI. The main content area is titled 'Feuerwehr' and shows a form for editing the data for 'Tillmitsch'. The form includes fields for 'Feuerwehrart', 'Name', 'Gemeinde', 'AFKDO', 'Strasse, Nr., Stiege/Stock/Tür', 'Katastralgemeinde', 'Plz, Ort', 'Bemerkungen', 'Name (Kurz)', 'Nummer (Kurz)', 'Nummer', 'DVR-Nr.', 'Gründungsjahr', 'Feuerwehrjugend von - bis', 'Verwendet FDISK (ja/nein)', 'Sonderdienstpostenplan (ja/nein)', 'Lücken in Standesbuchnummern füllen (ja/nein)', 'Atemluftfüllstelle (ja/nein)', and 'Logo'. A 'Speichern' button is at the bottom. The right sidebar contains several informational and administrative links under headings like 'Informationen', 'Weitere Eigenschaften', 'Mitgliederverwaltung', and 'Geräteverwaltung'.

Abbildung 7: Darstellung des Feuerwehrdateninformationssystem und Katastrophenschutzmanagement, Quelle: Eigene Darstellung.

Das FDISK erfüllt nach wie vor seinen Zweck, mittlerweile ist es jedoch in die Jahre gekommen und muss in den nächsten Jahren grundlegend modernisiert werden. Eine Herausforderung, welches im Zuge dieser Masterarbeit auftrat, stellte die direkte Integration der Datenbank in das LMS dar, da sich der Entwicklungsstand für den direkten Datenimport und -export noch auf dem Stand von 2002 / 2003 befindet.

¹⁸ Vgl. Niederösterreichischer Landesfeuerwehrverband (o.J.): Online-Quelle [13.06.2024].

Zusätzlich wurden hier aufgrund der Datenschutzgrundverordnung (DSGVO) nur die essenziellsten Informationen, die für das LMS benötigt werden, vom LFV Steiermark freigegeben. Die Bereitstellung der Daten erfolgte durch eine CSV-Datei.

Diese CSV-Datei wurde für das Lernmanagementsystem entsprechend aufbereitet und anschließend in das Plugin für die Benutzeranmeldung (siehe Kapitel 6.6.1) integriert.

Die berechtigten Funktionäre*innen einer Feuerwehr haben nur auf ihren eigenen Wirkungsbereich im FDISK Zugriff. Dies verhindert einen möglichen Wettkampf unter den Wehren, das Teilen von wichtigen internen Daten (persönliche Daten, Fahrzeugdaten, usw.) und möglichen böswilligen Interaktionen.

4.1.1 Funktionäre*innen im Bereich Jugend

In der vom LFV Steiermark zur Verfügung gestellten CSV-Datei gibt es eine Auflistung aller Jugendbeauftragten auf Orts-, Abschnitts-, Bereichs- und Landesebene. Im FDISK-Auszug vom 10. Juli 2024 gibt es insgesamt 1430 Einträge, wobei einige Namen doppelt vorkommen, da diese zum Beispiel die Funktion als Orts- und Bereichsjugendbeauftragte*r bekleiden.

Für die Anmeldung der Funktionäre*innen wurden folgende Daten zur Verfügung gestellt:

- Funktion
- Vorname
- Nachname
- Feuerwehr
- Dienstgrad
- Standesbuchnummer

Mit diesen Daten sollen im LMS neben der Anmeldung auch die Berechtigungen der*s jeweiligen Funktionär*in geregelt werden. Zusätzlich gibt es Fragen im Wissenstest, bei denen die Namen der zuständigen Orts-, Abschnitts-, Bereichs- und Landesjugendbeauftragte*n, genannt werden müssen. Dabei werden die eingegebenen Antworten automatisch mit den Daten der CSV-Datei abgeglichen. Dadurch soll später sichergestellt werden, dass die Jugendlichen immer auf dem neuesten Stand ausgebildet werden.

4.1.2 Jugend

In derselben CSV-Datei befinden sich auf einem anderen Arbeitsblatt dieselben Daten für alle Jugendlichen. Zu dem bereits in Abschnitt 4.1.1 genannten Auszugsdatum gab es in der Steiermark 6.355 Jung-Florianis. Durch die Zugehörigkeit zu einer Feuerwehr können die Mitglieder der Feuerwehrjugend ihrer*m verantwortlichen Ortsjugendbeauftragte*n zugeordnet werden.

Auch für die Jugendlichen gibt es die gleichen Anmeldebedingungen wie für die Funktionäre*innen.

4.1.3 Fahrzeuge und Stützpunkte

Um die Fragen beim Wissenstest Gold, welche sich auf die Standorte von Sonderfahrzeugen und Stützpunktfeuerwehren beziehen, überprüfen zu können, musste eine eigene Tabelle in der Datenbank erstellt werden (siehe Abschnitt 4.2.6). Leider war es nicht möglich diese Informationen über den LFV

Steiermark zu beziehen, somit wurde dieses Arbeitsblatt mit den Infos von der Homepage des LFV Steiermark zusammengetragen.

Sonderfahrzeuge sind Fahrzeuge, die für besondere Aufgaben in den jeweiligen Bereichen eingesetzt werden. Dabei handelt es sich zum Beispiel um Drehleitern, Einsatzleitfahrzeuge oder Ölschadensfahrzeuge (siehe Abbildung 8 und 9).



Abbildung 8: Gefährliche Stoffe Fahrzeug (GSF) der FF Lebring,
Quelle: Freiwillige Feuerwehr Lebring (o.J.), Online-Quelle [20.05.2024].

Zusätzlich zu den Sonderfahrzeugen in den jeweiligen Bereichen existieren auch Stützpunktfeuerwehren. Diese Stützpunkte decken spezialisierte Aufgabenbereiche ab, wie etwa Tauchstaffeln, den Wasserdienst, den Atemschutz und den Strahlenschutz.

2) Nenne den Standort folgender Sonderfahrzeuge in deinem Bereich:

- a) schweres Rüstfahrzeug (SRF)
- b) Gefährliches-Stoffe-Fahrzeug (GSF) bzw. Ölschadensfahrzeug (ÖF)
- c) Einsatzleitfahrzeug (ELF)
- d) Wechselladefahrzeug (WLF)
- e) Hubsteiger oder Drehleiter (DL)

Antwort:	a) SRF bei Feuerwehr	_____
	b) GSF bei Feuerwehr	_____
	ÖF bei Feuerwehr	_____
	c) ELF bei Feuerwehr	_____
	d) WLF bei Feuerwehr	_____
	e) DL bei Feuerwehr	_____

Abbildung 9: Frage aus dem Wissenstest Stufe Gold, Quelle: Eigene Darstellung.

4.2 MySQL™

Ein weithin anerkanntes System zur Verwaltung von Daten ist MySQL, welches auf der Datenbanksprache SQL basiert. Es wurde entwickelt, um eine effektive Organisation, Speicherung und den Abruf umfangreicher Datenmengen zu ermöglichen. MySQL findet Verwendung in verschiedenen Bereichen von kleinen Blogs bis hin zu komplexen Geschäftsanwendungen und ist bekannt für seine Leistungsfähigkeit, Zuverlässigkeit und Flexibilität. Ein wesentlicher Bestandteil von MySQL ist der Primärschlüssel, der sicherstellt, dass jeder Datensatz eindeutig identifiziert wird. Dadurch können Duplikate vermieden werden. Diese Funktion wird besonders wichtig, wenn mehrere Tabellen verknüpft werden müssen, was oft durch Fremdschlüssel geschieht, die auf den Primärschlüssel einer anderen Tabelle verweisen. MySQL ist äußerst beliebt in Webanwendungen wie WordPress und bildet häufig einen zentralen Bestandteil des LAMP-Stacks, einer bewährten Kombination von Technologien zur Erstellung von Websites. Neben dem Betriebssystem Linux und dem Webserver Apache kommen auch PHP oder Perl zur Verarbeitung der Website-Logik zum Einsatz. In diesem Kontext agiert MySQL als das System, das alle notwendigen Daten speichert. Eine herausragende Stärke von MySQL liegt in seiner Fähigkeit, Transaktionen zu handhaben, was bedeutet, dass Änderungen entweder vollständig umgesetzt oder zurückgezogen werden können. Dies ist bei kritischen Abläufen wie Online-Zahlungen von großer Bedeutung.¹⁹

Die Verwaltung einer MySQL-Datenbank kann auf unterschiedliche Weise erfolgen. Für die Verwaltung und Bearbeitung der in den Kapiteln 4.2.2 bis 4.2.6 beschriebenen Datenbanktabellen wurde die MySQL Workbench verwendet (siehe Abbildung 10).

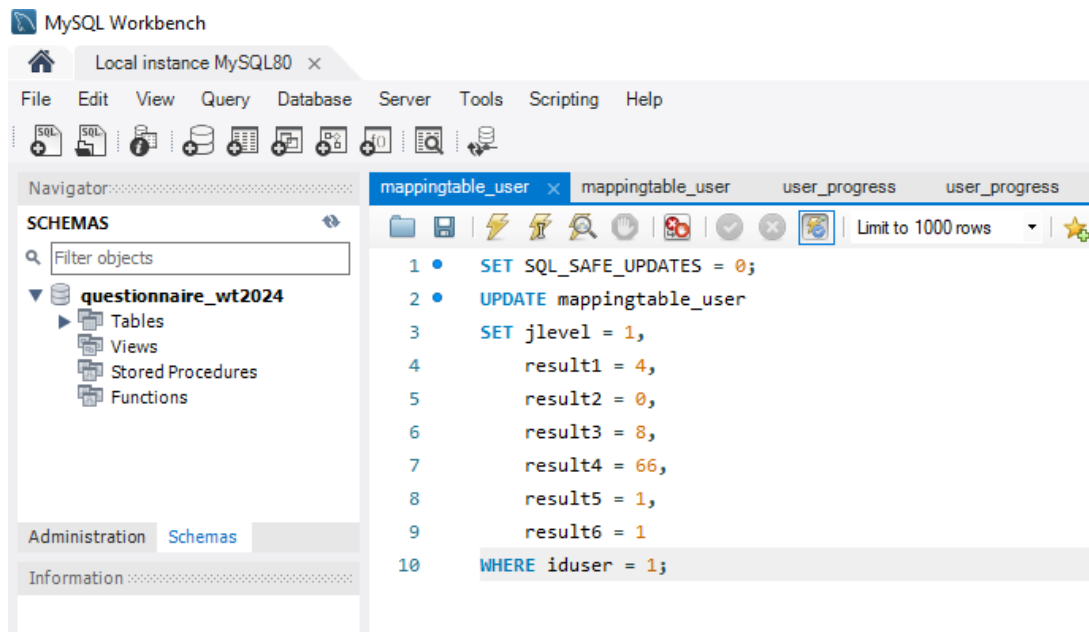


Abbildung 10: Darstellung der MySQL Workbench des LMSS, Quelle: Eigene Darstellung.

Diese bietet durch ihre grafische Oberfläche eine benutzerfreundliche Alternative zu den klassischen Befehlen, welche über die Kommandozeile eingegeben werden. Diese Programme erleichtern die Arbeit

¹⁹ Vgl. Oracle Corporation (o.J.): Online-Quelle [24.06.2024].

mit der Datenbank erheblich und machen MySQL zu einer soliden Wahl für sowohl kleine Projekte als auch große Unternehmen.

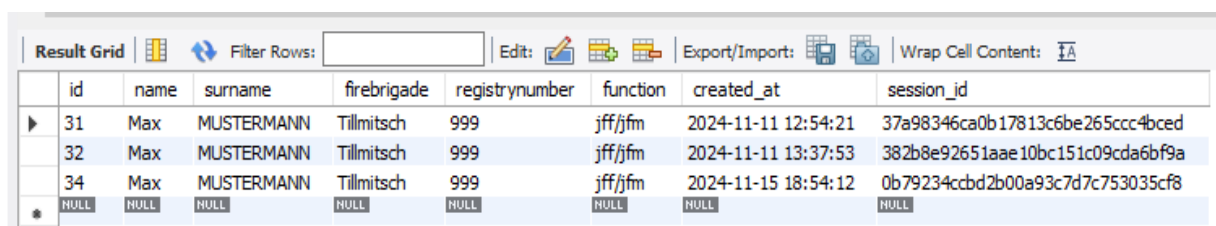
4.2.1 Datenbankverknüpfung

Da die Daten vom LFV Steiermark nur als CSV-Datei zur Verfügung gestellt wurden und die Vorgehensweise der Integration des FDISK noch nicht final geklärt ist, wird im Lernmanagementsystem die CSV-Datei unverändert verarbeitet. Es wäre an dieser Stelle eine Option gewesen, die CSV-Datei direkt in die Datenbank des LMSs als eigene Tabelle zu integrieren. Mit dem IST-Stand kann jedoch eine externe Datenquelle dargestellt werden und die korrekte Verarbeitung dieser Daten betrachtet werden (siehe Kapitel 6.6.1).

In den folgenden Abschnitten werden die jeweiligen Tabellen, welche im LMS verwendet wurden, genauer erklärt. Besonders wichtig ist dabei das Zusammenspielen der einzelnen Tabellen, damit alle Funktionalitäten des LMSs abgedeckt werden können.

4.2.2 Tabelle: users

Die Tabelle users wird in erste Linie für das Mapping der CSV-Datei in eine Datenbanktabelle benötigt. Das Beschreiben dieser Tabelle geschieht während des Authentifizierungsprozesses. Dabei wird bei jeder erfolgreichen Authentifizierung geprüft, ob es bereits eine Eintragung mit den Daten gibt, die nicht älter als 30 Minuten ist (siehe Abbildung 11). Ist dies der Fall, so wird lediglich der Eintrag in created_at aktualisiert. Im Allgemeinen werden eben die Daten aus der CSV-Datei in diese Datenbanktabelle übertragen. Um auf den nachfolgenden Seiten den Benutzer weiterhin authentifizieren zu können, wird zusätzlich ein Hash zur Erstellung einer Session-ID verwendet. Es wird auf allen nicht frei zugänglichen Seiten geprüft, ob ein gültiger Hash vorhanden ist. Sollte das nicht der Fall sein, so wird die*der Benutzer*in automatisch auf die Startseite umgeleitet.



	id	name	surname	firebrigade	registrynumber	function	created_at	session_id
▶	31	Max	MUSTERMANN	Tillmitsch	999	jff/jfm	2024-11-11 12:54:21	37a98346ca0b17813c6be265ccc4bcd
	32	Max	MUSTERMANN	Tillmitsch	999	jff/jfm	2024-11-11 13:37:53	382b8e92651aae10bc151c09cda6bf9a
	34	Max	MUSTERMANN	Tillmitsch	999	jff/jfm	2024-11-15 18:54:12	0b79234ccbd2b00a93c7d7c753035cf8
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Abbildung 11: Darstellung der Datenbanktabelle users, Quelle: Eigene Darstellung.

4.2.3 Tabelle: mappingtable_wt

In der Datenbanktabelle mappingtable_wt wurden alle Inhalte des Wissenstests übertragen. Da die Unterlagen der Wissensteste nur als PDF vorhanden sind, wurde zuerst eine Excel-Datei erstellt, in der alle Stationen, Fragen, Antworten und Punkteverteilungen eingetragen wurden. Die Tabelle wurde dann um folgende Spalten erweitert (siehe Abbildung 12):

- idwt - Aufsteigende ID aller Wissenstestfragen.
- j_level - Darstellung der Wissensteststufe, Bronze = 1, Silber = 2, Gold = 3.
- question_no - Nummerierung der Frage in der jeweiligen Station.
- station_number - Nummerierung der jeweiligen Station.
- truefalse - Indikator, ob es sich um die richtige Antwort handelt.
- flag - Verweis, ob Antwortoptionen von einer anderen Datenbank geholt werden können.

idwt	j_level	question_no	station_number	station
1	1	1	1	Organisation der Feuerwehr
2	1	2	1	Organisation der Feuerwehr
3	1	3	1	Organisation der Feuerwehr
4	1	4	1	Organisation der Feuerwehr
5	1	5	1	Organisation der Feuerwehr
6	1	6	1	Organisation der Feuerwehr
7	1	7	1	Organisation der Feuerwehr
8	1	8	1	Organisation der Feuerwehr
9	1	1	2	Formalexerzieren
10	1	1	3	Fahrzeug- und Gerätekunde
11	1	2	3	Fahrzeug- und Gerätekunde

Abbildung 12: Darstellung der Datenbanktabelle mappingtable_wt Teil 1, Quelle: Eigene Darstellung.

Nachdem alle wissenstestspezifischen Daten in die Excel-Datei eingepflegt und diese als CSV-Datei gespeichert wurde, kam die MySQL Workbench zum Zug (siehe Kapitel 4.2). Mit der Import Funktion der MySQL Workbench konnten die Daten im Handumdrehen in die Datenbanktabelle migriert werden.

In Abbildung 13 kann in der Spalte answer bei den Antworten ein vermehrtes Vorkommen von Schrägstrichen (Slashes) erkannt werden. Diese Schrägstriche dienen als Trennzeichen für die PHP-Skripte, damit an dieser Stelle ein Zeilenumbruch stattfinden kann. Normal würde hier ein Semikolon zum Einsatz kommen, jedoch würde dies zu Problemen beim Migrieren der CSV-Datei führen, da das Trennzeichen der Spalten in einer CSV-Datei klassischerweise ein Komma ist. Dadurch würde es auf eine nicht korrekte Übertragung der Daten aus der CSV-Datei hinauslaufen.

question	answer	points	truefalse	flag
Nenne die wichtigsten Funktionen in deiner Feuerwehr:	Der Feuerwehrkommandant,/der Feuerwehrko...	1	T	
Nenne die Organe der Feuerwehr:	Der Feuerwehrkommandant,/der Feuerwehrko...	1	T	
Was sind die wichtigsten Regeln für die Benützung von Feuerwehrhaus u...	Feuerwehrhaus und Geräte dürfen nur bei Eins...	1	T	
Welche Dienstgrade gibt es in deiner Feuerwehr?	Offiziere,/Chargen,/Mannschaftsdienstgrade,/...	1	T	
Wie heißen dein Feuerwehrkommandant und dein Feuerwehrkommandan...	Feuerwehrkommandant: /Feuerwehrkommanda...	1	T	F
Wie heißen dein Ortsfeuerwehrjugendbeauftragter und der Geräte- und ...	Ortsfeuerwehrjugendbeauftragter: /Geräte- un...	1	T	F
Was ist eine Hauptaufgabe der*s Feuerwehrkommandantin*en?	Er*Sie ist für die Einsatzbereitschaft und Leistu...	1	T	
Was ist eine Hauptaufgabe der*s Ortsfeuerwehrjugendbeauftragten?	Vorbereitung der Feuerwehrjugend für den akti...	1	T	
Die Bewerber treten gruppenweise an, ein Bewerber kommandiert die Gr...	Der*ie Bewerber*in gibt folgende Kommandos:/...	4	T	
Wie werden Fahrzeuge eingeteilt?	Fahrzeuge zur Brandbekämpfung,/Fahrzeuge z...	1	T	

Abbildung 13: Darstellung der Datenbanktabelle mappingtable_wt Teil 2, Quelle: Eigene Darstellung.

Um auch bild- und videobasierte Antwortmöglichkeiten zu realisieren, wurde bei den betroffenen Antworten der Pfad zu den jeweiligen Dateien abgelegt. Der Vorgang der Medienverwaltung wird in Kapitel 6.1.4 genauer beschrieben.

4.2.4 Tabelle: mappingtable_user

Die mappingtable_user Tabelle besitzt ähnliche Inhalte wie die in Abschnitt 4.2.2 Tabelle users. Da die DSGVO eine Pseudonymisierung von personenbezogenen Daten vorsieht (siehe Kapitel 6.3), wurde der eigentliche Nachname verschlüsselt in der Datenbank abgelegt (siehe Abbildung 14). Es handelt sich hierbei um eine Vigenère-Verschlüsselung, welche in Abschnitt 6.6.2.2 genauer beschrieben wird.

iduser	name	codedsurname	firebrigade
2	Milki	RFGKMRZISY	Tillmitsch
3	Max	RFGKMRZISY	Tillmitsch

Abbildung 14: Darstellung der Datenbanktabelle mappingtable_user, Quelle: Eigene Darstellung.

Der Nachname muss prinzipiell erst wieder bei der Ergebnisauswertung, welche für die Betreuer*innen und die*den Ortsjugendbeauftragte*n relevant sind, entschlüsselt werden.

In der Datenbanktabelle mappingtable_user werden außerdem noch die Anzahl der Durchläufe des Lernmodus und des Prüfungsmodus mit dessen Ergebnisse abgespeichert (siehe Abbildung 15). Es werden nur die fünf letztgültigen Resultate in einer Art Ringspeicher gesichert.

	id	user_id	station_num	current_question_index	level
▶	1	1	1	5	1
	2	1	3	0	1
	3	1	5	0	2
	4	1	1	0	2
	-	-	-	-	-

Abbildung 15: Darstellung der mappingtable_user, Quelle: Eigene Darstellung.

4.2.5 Tabelle: user_progress

Um einen reibungslosen Ablauf des Lern- und Prüfungsmodus zu gewährleisten, wurde eine Tabelle, welche den aktuellen Fortschritt im jeweiligen Modus sichert, angelegt. Dafür werden Informationen über die Wissensteststufe, die Station, die Nummer der Frage und die Benutzer-ID in Zusammenhang mit dem gültigen Hash gespeichert.

4.2.6 Tabelle: vehicles_bases

Abschließend wurden noch die nicht zu Verfügung gestellten Informationen zu den Sonderfahrzeugen und Stützpunkten in der Tabelle vehicles_bases grob zusammengefasst. Diese Informationen werden zur Lösung der im Abschnitt 4.1.3 beschriebenen Aufgaben benötigt. Es wurde, wie bereits bei allen anderen Tabellen wieder eine ID vergeben. Die Einträge differenzieren sich durch die Spalten vehicle und base voneinander. Handelt es sich um ein Sonderfahrzeug, wurde die Bezeichnung des Fahrzeugs nur der Spalte vehicles zugeordnet. Bei den Stützpunkten bleibt dagegen die Spalte vehicles leer. Neben diesen Informationen wurden auch die zuständige Feuerwehr und der entsprechende Bereichsfeuerwehrverband hinzugefügt (siehe Abbildung 16).

i>id	vehide	base	firebrigade	bfv	last_edit
1		Atemschutz	Feuerwehr- und Zivilschutzschule	Leibnitz	05.11.2024
2		Strahlenschutz	Schoenberg	Leibnitz	05.11.2024
3		Wasserdienst	Tillmitsch	Leibnitz	05.11.2024
4	SRF		Leibnitz	Leibnitz	05.11.2024
5	GSF		Lebring	Leibnitz	05.11.2024
6	ELF		Gleinstaetten	Leibnitz	05.11.2024
7	WLF		Leutschach	Leibnitz	05.11.2024
8	DL		Leibnitz	Leibnitz	05.11.2024

Abbildung 16: Darstellung der Datenbanktabelle vehicles_bases, Quelle: Eigene Darstellung.

5 WEBSITES UND DAS INTERNET

Im Nachfolgenden soll nun etwas ausführlicher auf die essenziellen Komponenten, welche zur Erstellung des Lernmanagementsystems verwendet wurden, eingegangen werden. Dabei sollen vor allem die Begriffe mit den dazugehörenden Funktionen genauer erläutert werden.

5.1 Domain

Der Ausdruck Domain definiert einen eindeutigen Namen, den eine Website im Internet haben kann. Vergleichbar mit einem Navigationssystem und einer Wohnadresse führt die Eingabe der Domain im Webbrowser oder in der Suchmaschine direkt zur gewünschten Adresse. Generell besteht eine Domain aus zwei Teilen, dem Domainnamen oder auch Second-Level-Domain (SLD) genannt und der Top-Level-Domain (TLD), wobei der TLD wiederum Auskunft über das Herkunftsland der Website geben kann. So steht der TLD “.at“ für Websites aus Österreich und “.com“ für kommerzielle Webauftritte. Hinter einer Domain versteckt sich jedoch die IP-Adresse des Servers, auf dem sich die Website befindet. Somit dient eine Domain prinzipiell nur der besseren Darstellung und Lesbarkeit.²⁰

Unter Einhaltung gewisser Regeln kann fast jeder Domainname frei gewählt werden und über offizielle Registrierungsstellen (z.B.: nic.at) eingetragen werden. Wird eine neue Domain registriert fallen in Abhängigkeit der gewählten Registrierungsstelle und des ausgesuchten TLDs unterschiedliche Kosten an. Die Gebühren für eine Domain sind jährlich zu leisten, da diese ansonsten wieder für alle freigegeben werden kann.

Für die Erstellung des Lernmanagementsystems im Zuge dieser Masterarbeit wurde keine Domain gekauft, da das LMS später direkt an die Internetseite des LFV Steiermark angeknüpft werden soll.

5.2 Website und Webseite

Der Unterschied zwischen den beiden Begriffen Website und Webseite liegt hauptsächlich in ihrer deutschsprachigen Bedeutung. So steht eine Website oder im deutschen auch oft als Internetseite bezeichnet für eine Sammlung von mehreren unterschiedlichen Webseiten.²¹

Dies bedeutet, dass beim Aufrufen einer bestimmten Domain oder beim Öffnen des bevorzugten Webbrowsers immer nur eine einzelne Webseite angezeigt wird. Dahingegen umfasst eine Website das gesamte System, welches alle Webseiten miteinander verknüpft, unter einer Domain zusammen.

Als abstrakte Vorstellungsweise kann hier der Vergleich mit einem Buch herangezogen werden. So stellt dieses Druckwerk in seiner Gesamtheit die Website dar. Jede Webseite, die zur selben Website gehört, wird wiederum von einer Seite aus diesem Buch repräsentiert.

²⁰ Vgl. Ionos SE (2024): Online-Quelle [29.05.2024].

²¹ Vgl. Müller, Peter (2024): S. 33.

5.3 Host, Datenbank-Host und Server

Der Begriff „Host“ ist nicht eindeutig definiert, da er je nach Kontext unterschiedliche Bedeutungen aufweisen kann. Einerseits werden als Host Endgeräte wie Computer, Laptops, Smartphones, usw. welche mit einem Netzwerk verbunden sind und Dateien senden und empfangen bezeichnet. Andererseits handelt es sich dabei um Datenbankanbieter oder Hosting-Anbieter, welche ihre Server und Infrastruktur bereitstellen, um Datenbanken zu hosten und zugänglich zu machen. Soll eine Website online gehen, so muss erst ein passender Datenbank-Host gefunden werden. Dabei sollten stets folgende Punkte beachtet werden:

- Kosten
- Leistung und Skalierbarkeit
- Zuverlässigkeit und Verfügbarkeit
- Sicherheit
- Unterstützte Datenbanktypen
- Wiederherstellungsoptionen
- Rechtliche Anforderungen
- Kundensupport

Rein technisch betrachtet ist ein jeder Server auch ein Host, da er mit dem Netzwerk oder Internet verbunden ist, jedoch ist nicht jeder Host ein Server. Im Vergleich zu den oben angeführten Endgeräten ist ein Server normalerweise ein spezieller Computer, welcher für den Dauerbetrieb ausgelegt ist und eine leistungsstarke Hardware besitzt (siehe Abbildung 17). Es gibt aber virtuelle Server, welche wiederum auf physischen Computern laufen, um dort bestimmte Aufgaben (z.B.: Mailserver wie Microsoft Exchange) zu erfüllen.²²

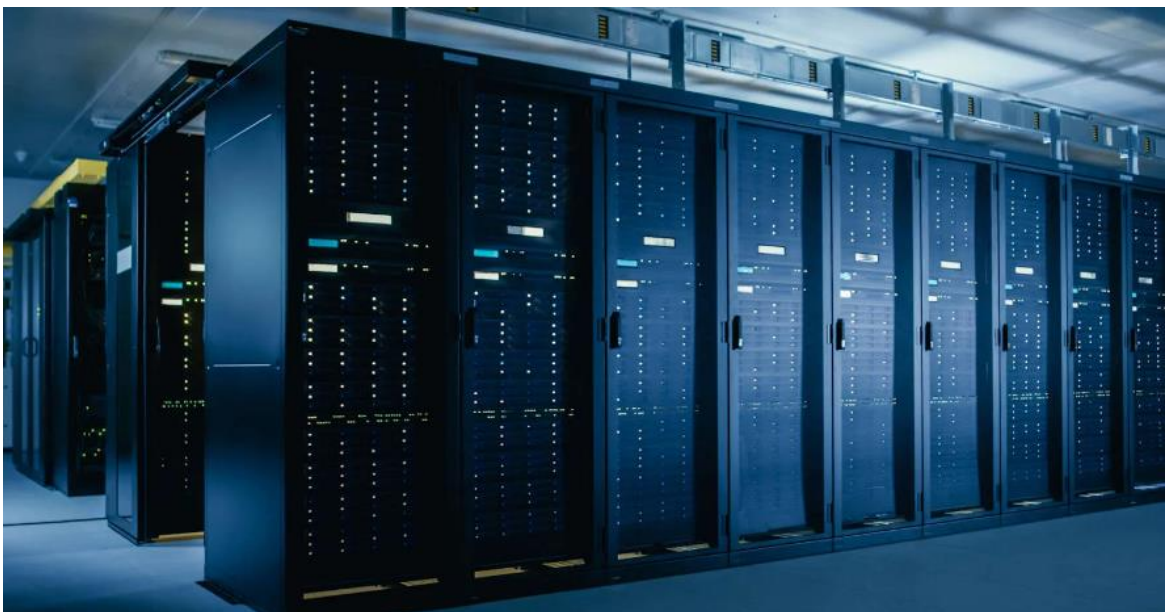


Abbildung 17: Darstellung eines Serverraums, Quelle: Adobe Stock (o.J.), Online-Quelle [22.11.2024]

²² Vgl. Fuchs Media Solutions (o.J.): Online-Quelle [29.05.2024].

5.4 FTP-Server

Der File Transfer Protocol-Server (FTP-Server) kann eine spezielle Art von Server oder auch eine Software sein, die dafür entwickelt wurde, Dateien über ein lokales Netzwerk oder Internet zu übertragen. Ein wichtiger Aspekt des FTP-Servers ist, dass er normalerweise mit Benutzerkonten arbeitet. Dadurch muss sich jede*r vor dem Zugriff auf den Server mit einem Benutzernamen und einem Passwort authentifizieren. Erst danach können Dateien vom beziehungsweise auf den FTP-Server geladen werden. Der Datentransfer wird dabei über eine Transmission Control Protocol Verbindung (TCP-Verbindung) abgewickelt. Die Kommunikation zwischen dem FTP-Client und dem FTP-Server erfolgt dabei über zwei separate Kanäle, dem Steuerkanal und dem Datenkanal. Wie es der Name schon verrät, werden über den Steuerkanal Befehle wie "Datei herunterladen" oder "Datei hochladen" gesendet. Über den Datenkanal werden nach dem Erhalt der Anweisungen anschließend die Daten geschickt. Um einen höheren Sicherheitsstandard bei der Datenübertragung zu gewährleisten, kann auch das SFTP (Secure File Transfer Protocol) oder FTPS (File Transfer Protocol Secure) verwendet werden.²³ Im Allgemeinen werden alle Dateien, die für eine korrekte Ausführung einer Website benötigt, werden auf einem FTP-Server gespeichert. Dazu zählen HTML-Dateien, Plugins, Themes, Medien (z.B.: Bilder, Videos, usw.), Datenbankdateien und viele mehr.

Da das Lernmanagementsystem auf einem lokalen Server implementiert wurde, stellt und verwaltet die in Kapitel 6.4.1 beschriebene Software Local® den lokalen FTP-Server.

5.5 HTTP und HTTPS

Um eine Webseite im Browser anzeigen zu können, wird das Hypertext Transfer Protocol (HTTP) verwendet. Dabei werden die zur Anzeige der Internetseite benötigten Inhalte und Daten wie Texte, Bilder und Videos über das HTTP ausgetauscht. Das einzige Problem bei HTTP ist, dass Informationen nicht verschlüsselt übertragen werden. An dieser Stelle wird das Hypertext Transfer Protocol Secure (HTTPS) ins Spiel gebracht. Beide Typen werden als Protokoll-Präfix verwendet (siehe Abbildung 18).

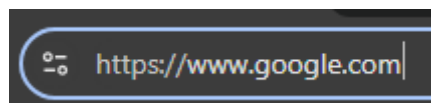


Abbildung 18: Darstellung des Protokoll-Präfix HTTPS im Webbrowser, Quelle: Eigene Darstellung.

HTTPS setzt eine Mischung aus symmetrischer und asymmetrischer Verschlüsselung ein, welche unterschiedlich zur Anwendung kommen. Zuerst wird eine asymmetrische Codierung verwendet, mit der eine sichere Verbindung zwischen dem eigenen Browser und dem Zielsystem hergestellt wird. Nachdem diese Verbindung durch Überprüfen des Server-Zertifikates zustande gekommen ist, werden alle weiteren Daten, welche übertragen werden, symmetrisch verschlüsselt. Die genaueren Funktionen von HTTP und HTTPS werden in den Request For Comments Dokumenten (RFC-Dokumenten) beschrieben. Bei HTTP handelt es sich um das RFC 2616 und bei HTTPS um das RFC 8446 Dokument.²⁴

²³ Vgl. Ionos SE (2023): Online-Quelle [30.05.2024].

²⁴ Vgl. Myra Security GmbH (o.J.): Online-Quelle [30.05.2024].

5.6 Cookies

Cookies sind kleine Textdateien, die von einer Website auf dem aufrufenden Endgerät gespeichert werden. Benötigt werden diese Dateien, um das Surfen im Internet im Allgemeinen bequemer zu gestalten. Es gibt eine Vielzahl an unterschiedlichen essenziellen und nicht-essenziellen Cookies, die in den verschiedensten Bereichen eingesetzt werden können.

- Essenzielle Cookies
 - Authentifizierungs-Cookies
 - Sicherheits-Cookies
 - Session-Cookies
- Funktionale Cookies
 - Sprachpräferenzen
 - Speichern von Warenkörben
 - Layouteinstellungen (z.B.: Dark-Mode)
- Marketing- und Tracking-Cookies
 - Google Analytics-Cookies (Werbefunktion)
 - Facebook Pixel
 - LinkedIn Insight Tag

Bei der Einbindung von Cookies in eine Website ist jedoch darauf zu achten, dass deren Verwendung sich an die geltenden EU-Datenschutz-Grundverordnung (DSGVO) und der E-Privacy- / Cookie-Richtlinie halten (siehe Kapitel 6.3). Der Betreiber einer Website muss sicherstellen, dass Nutzer ihre ausdrückliche Zustimmung zu den jeweiligen angebotenen Cookies geben.²⁵ Dies geschieht über ein sogenanntes Cookie-Banner (siehe Abbildung 19).

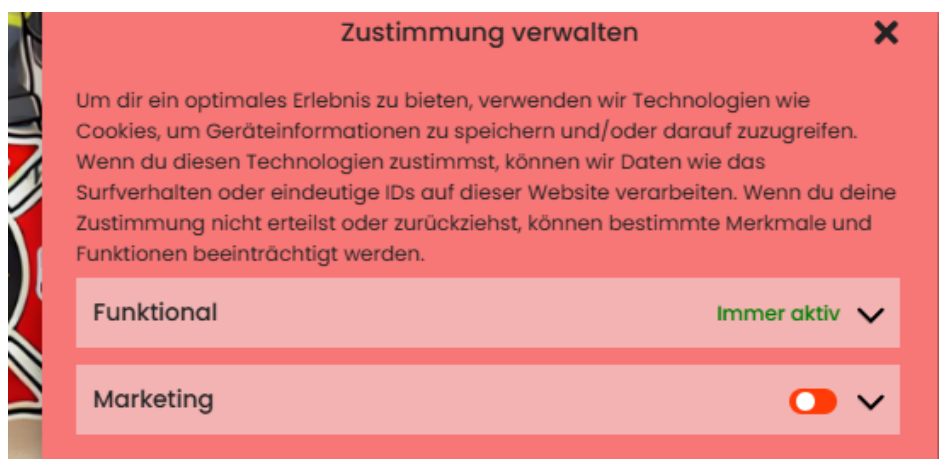


Abbildung 19: Das Cookie-Banner des Lernmanagementsystems, Quelle: Eigene Darstellung.

In diesem Cookie-Banner muss der Nutzer über die Notwendigkeit (essenziell oder nicht-essenziell) des Cookies, zu welchem Zweck dieser verwendet wird und welche Daten an möglicherweise Dritte

²⁵ Vgl. Suske, Sophie (2023): Online-Quelle [12.07.2024].

weitergegeben werden, informiert werden. Zudem haben Nutzer jederzeit die Möglichkeit, ihre Cookie-Einstellungen zu ändern oder die Zustimmung zu widerrufen.

Das Cookie-Banner wurde mit dem kostenlosen Plugin Complianz generiert. Ein Vorteil solcher vorgefertigten Plugins ist die schnelle Implementierung auf einer Wordpress Website. Schon nach nur wenigen Klicks wird das entsprechende Cookie-Banner generiert und länderspezifische Eigenschaften übernommen. Für das LMS wurde zum Beispiel die Farbe des Banners automatisch an die Homepage angepasst. Zusätzlich können die Webseiten für die Cookie-Richtlinien, das Impressum und die Datenschutzerklärung direkt über dieses Plugin editiert und kreiert werden. Dabei werden alle wichtigen gesetzlichen Themen gemäß der DSGVO abgedeckt.

Da das Lernmanagementsystem zurzeit auf einem lokalen Host läuft, ist der Einsatz von Cookies derzeit noch nicht notwendig. Ausnahmen sind die von Wordpress und Complianz automatisch generierten Cookies. Hier werden zum Beispiel die Wordpress Benutzerdaten und die Daten vom Cookie-Banner bereits gespeichert.

Später werden jedoch noch Cookies zur Benutzerverwaltung des LMSs und von Google Analytics zur Analyse des Besucherverhaltens und der Besucherinteraktionen hinzugefügt.

Zum Schluss muss ein weiterer wichtiger Punkt erwähnt werden. So praktisch Cookies in der Verwendung auch sind, Benutzer haben das Recht, diese auch abzulehnen. Dadurch ist sicherzustellen, dass Websites ohne den Einsatz von Cookies tadellos und uneingeschränkt funktionieren.²⁶

²⁶ Vgl. Wenz, Christian; Hauser, Tobias (2024): S. 489-492.

6 DIE LERNMANAGEMENTSYSTEM WEBSITE

Das LMS soll auf Basis einer Website erstellt werden, da so eine Umsetzung, geräte- und plattformunabhängig ist. Dadurch kann das Lernmanagementsystem auf unterschiedlichen Endgeräten vom Smartphone bis hin zum Smart-TV verwendet werden. Zusätzlich spielt auch die Verwendung des am Endgerät installierten Betriebssystems keine Rolle. Die Vorteile gegenüber einer eigens programmierten App liegen dabei auf der Wartbarkeit und der Einfachheit der Umsetzung. Würde das LMS über eine App realisiert werden, so müsste ein eigener Quellcode für Endgeräte mit Android und iOS angefertigt werden, da diese beiden Systeme unterschiedliche Programmiersprachen verwenden. Bei inhaltlichen Änderungen am LMS müssten diese dann separat auf jeder Plattform angepasst werden.

6.1 WordPress®

WordPress betreibt weltweit Millionen von Websites und ist ein sehr beliebtes und weitverbreitetes Content-Management-System (CMS). Es wurde 2003 von Mike Little und Matt Mullenweg als Blogging-Plattform ins Leben gerufen. Mittlerweile ist es zu einem der vielseitigsten Tools für die Erstellung und Verwaltung von Websites aller Art geworden – von simplen Blogs bis hin zu komplexen E-Commerce-Websites und Unternehmensportalen. Mehr als 43% der Websites, deren CMS bekannt ist, verwenden WordPress. Es handelt sich daher um das am häufigsten verwendete System für den Betrieb von Websites mit CMS.²⁷

WordPress wird auch von sehr vielen namhaften Unternehmen wie Microsoft, der NASA oder der Harvard University verwendet. Es gibt zwei Versionen von WordPress, die in Bezug auf ihre Handhabung, Kontrolle und Kostenstruktur variieren.

6.1.1 Hauptvarianten von WordPress

- **WordPress.org**

Diese Open-Source-Software ist kostenlos auf der offiziellen Website erhältlich. Diese Version muss auf ihrem eigenen Webserver installiert werden, wodurch sie das System vollständig kontrollieren können.

Vorteile:

Vollständige Steuerung des Hostings, des Designs und der Funktionen. Alle Themes und Plugins können ohne Einschränkungen installiert und angepasst werden. Die Option besteht darin, die Website durch Werbung zu monetarisieren.

Nachteile:

Die Nutzer müssen sich mit dem Webhosting, den Sicherheitsaktualisierungen und den Sicherungen befassen. Um anspruchsvollere Anpassungen durchzuführen, ist ein technisches Verständnis oder die Verwendung eines Webentwicklers notwendig.

²⁷ Vgl. WordPress.org (o.J.): Online-Quelle [21.03.2024].

- **WordPress.com**

Das Unternehmen Automattic, das auch WordPress.org unterstützt, ist für diese Version verantwortlich. Diese Plattform übernimmt den technischen Aufwand der Nutzer, indem sie die Website auf ihren Servern hostet und verwaltet.

Vorteile:

Aufgrund der Verwaltung des Hostings und der Sicherheitsupdates von WordPress.com ist kein technisches Fachwissen erforderlich. Perfekt geeignet für Anfänger und für kleine Webseiten oder Blogs.

Nachteile:

Es gibt Einschränkungen bei der Anpassung, insbesondere in der kostenlosen Version. Die Monetarisierung und die Einbindung von Werbung werden weniger kontrolliert.

6.1.2 Aufbau und Struktur von WordPress

Die Erstellung und Strukturierung von mit WordPress erstellten Websites erfolgt mithilfe der folgenden beiden Hilfsmittel:

- **Themes (Designvorlagen):**

Themes stellen bereits erstellte Designvorlagen dar, die die Gestaltung und das Aussehen einer Website bestimmen. Es existieren zahlreiche kostenlose und bezahlte Themes, die sich für verschiedene Zwecke wie Blogs, Portfolios, Unternehmensseiten oder E-Commerce eignen. Mit dem Backend (Administrativer Bereich eines CMS) von WordPress können Themes problemlos installiert und konfiguriert werden. Viele Themes verfügen über einen eingebauten Editor, der eine benutzerfreundliche Anpassung ermöglicht. Dadurch ist es möglich, Farben, Schriftarten und Layouts ohne Programmierkenntnisse schnell und einfach anzupassen.²⁸

- **Plugins (Erweiterungen):**

Plugins sind Add-ons, die zusätzlichen Funktionen und Features zu einer WordPress-Website hinzufügen. Es gibt über 60.000 Plugins im offiziellen WordPress-Plugin-Verzeichnis, die nahezu jede denkbare Funktionalität bieten. Nachfolgend eine kleine Anführung von Plugins:²⁹

- SEO-Optimierung (z.B.: Yoast SEO, SEOPress, WordLift)
- E-Commerce (z.B.: WooCommerce, ShopEngine, MemberPress)
- Sicherheitsfeatures (z.B.: Wordfence, Sucuri, Security Ninja)
- Kontaktformulare (z.B.: Contact Form 7, Ninja Forms, QuForm)
- Performance-Optimierung (z.B.: Perfmatters, WP Rocket)

Benutzer können Plugins einfach über das Dashboard installieren, aktivieren und verwalten.

²⁸ Vgl. Simeth, Florian (2022): S. 175-176.

²⁹ Vgl. Müller, Peter (2024): S. 407-439.

6.1.3 Benutzerverwaltung

WordPress bietet die Möglichkeit, verschiedene Benutzerrollen zu unterstützen, was insbesondere für große Websites mit mehreren Autoren oder Administratoren von Nutzen ist. Die üblichen Rollen sind:³⁰

- Administrator: Besitzt Rechte für die gesamte Website.
- Redakteur: Hat die Möglichkeit, Inhalte von sich und anderen zu modifizieren und zu veröffentlichen.
- Autor: Es ist ihm nur gestattet, seine eigenen Beiträge zu veröffentlichen.
- Mitarbeiter: Ermöglicht das Erstellen von Beiträgen, veröffentlicht diese jedoch nicht.
- Abonnent: Er hat nur Zugang zum Lesen und kann beispielsweise Kommentare schreiben.

Diese Benutzerverwaltung ist nach der Beendigung dieser Masterarbeit besonders wichtig, da die Website auf einen neuen Host transferiert werden soll. Dort wird das LMS dann von mehreren Mitarbeitern des LFV Steiermark kontinuierlich bearbeitet und erweitert werden.

6.1.4 Medienverwaltung

Eine eingebaute Medienbibliothek von WordPress ermöglicht das Hochladen und Verwalten von Bildern, Videos, PDFs und anderen Mediendateien. Das Einbinden dieser Dateien in Beiträge und Seiten macht es einfacher, Lernunterlagen welche Bild- oder Videomaterial benötigen, zu erstellen, die visuell ansprechend gestaltet sind. Dies wird vor allem für die Geräte-, Knoten- und Dienstgradkunde eingesetzt (siehe Kapitel 6.7.4).

Grundsätzlich können Medien auf drei Arten integriert werden. Eine Variante wäre die Integration über die Datenbank. Dabei wird zum Beispiel ein Bild als LOB (Large Object) Format gespeichert. Folgende Datentypen können als LOB gewählt werden:

Datentyp	Maximalgröße in Bytes	Maximalgröße in Megabytes
TINYBLOB	256 B	0,0002 MB
MEDIUMBLOB	655.536 B	0,6000 MB
BLOB	1.677.216 B	1,7000 MB
LOB	4.294.967.296 B	4.295,0000 MB

Tabelle 2: Übersicht der LOB Datentypen, Quelle: Hauser / Wenz (2024), S. 624 (leicht modifiziert).

Sollten nur vereinzelt Mediendateien auf der Website verwendet, empfiehlt sich das BLOB (Binary Large Object). Das Bild wird beim BLOB als reine Binärdatei ohne dem inhaltspezifischen Format gespeichert.³¹

³⁰ Vgl. World4You Internet Services GmbH (2024): Online-Quelle [20.11.2024].

³¹ Vgl. Wenz, Christian; Hauser, Tobias (2024): S. 624.

Werden größere Videos verwendet, so gibt es die Möglichkeit diese von anderen Websites wie YouTube direkt einzubetten. Diese Methode hat den Vorteil, dass der eigene FTP-Server nicht überladen wird und die Performance der Website nicht unter den großen Datenmengen leidet.

Die letzte Variante eignet sich ausgezeichnet für die Verwendung im Lernmanagementsystem. In der Datenbank wird nur der Pfad zur Mediendatei gespeichert. Die Bilddatei selbst wird am FTP-Server abgelegt. Dies hat den Vorteil, dass Bilder einfach am Server ausgetauscht werden können, ohne dass die Links in den Datenbanken angepasst werden müssen. Die einzige Prämisse dabei ist, dass die Dateien wieder unter demselben Namen gespeichert werden. Die vier Videodateien, welche beim Formalexerzieren und der Knotenkunde zum Einsatz kommen, wurden auf die gleiche Art und Weise eingebunden.

6.1.5 Sicherheit und Wartung

WordPress ist selbstverständlich ein sicheres System, doch aufgrund seiner weltweiten Beliebtheit als CMS ist es auch für Hacker ein bevorzugtes Ziel. Es ist unverzichtbar, die Website regelmäßig zu aktualisieren und Sicherheits-Plugins wie Wordfence oder Sucuri (siehe Kapitel 6.1.2) zu nutzen, um sie abzusichern.

Einige bedeutende Sicherheitsvorkehrungen für WordPress:

- Kontinuierliche Sicherung von Website-Daten und Datenbank.
- Nutzung leistungsstarker Kennwörter und Zwei-Faktor-Authentifizierung.
- SSL-Zertifikate für einen verschlüsselten Anschluss installieren.
- Aktualisierungen sowohl von WordPress als auch von Themes und Plugins.

6.1.6 Vor- und Nachteile von WordPress

Nachfolgende Vor- und Nachteile gilt es bei der Verwendung von WordPress zu beachten:³²

- **Vorteile**
 - Einfachheit
Nutzer, die keinerlei technisches Wissen besitzen, können Inhalte mühelos erstellen und verwalten.
 - Flexibilität
WordPress ist für eine Vielzahl von Website-Typen geeignet – von Blogs bis zu E-Commerce-Lösungen.
 - Erweiterbarkeit
Aufgrund der großen Auswahl an Plugins und Themes ist es unkompliziert, die Funktionen zu erweitern.
 - SEO-Freundlichkeit
WordPress ist für Suchmaschinen gut geeignet und verfügt über Erweiterungen, die das SEO-Ranking verbessern. Auf eine SEO-Optimierung wurde beim Lern-Management-System bewusst verzichtet, da nur berechtigte Nutzer Zugriff auf die Website und deren Inhalte haben.

³² Vgl. Siever, Friedrich (2024): Online-Quelle [11.09.2024].

- Große Community

Die globale WordPress-Community bietet Hilfe, Anleitungen und viele Ressourcen an.

- **Nachteile**

- Sicherheitsrisiken

Aufgrund seiner weiten Verbreitung wird WordPress häufig Gegenstand von Hackerangriffen. Daher sind Sicherheitsaktualisierungen und Sicherheitsmaßnahmen unverzichtbar.

- Wartungsaufwand

Nutzer müssen insbesondere bei selbst gehosteten Installationen regelmäßig Aktualisierungen und Sicherungen durchführen.

- Performance

Abhängig von der Anzahl und der Qualität der Plugins kann sich die Leistung der Website verschlechtern, was wiederum zu einer Verlangsamung der Ladezeit führen kann.

6.2 Content-Management-System

Ein Content-Management-System ist ein Programm, mit dem Nutzer*innen Inhalte auf einer Website erstellen, verwalten und veröffentlichen können. Dabei müssen sie nicht einmal über umfangreiche Programmierkenntnisse verfügen. WordPress ermöglicht es Benutzern, ihre Inhalte ihrer Website mühelos zu aktualisieren, neue Seiten zu erstellen und das Design anzupassen, ohne dass sie einen Eingriff in den Code benötigen.³³

6.3 Datenschutzgrundverordnung

Bei der DSGVO handelt es sich um eine Verordnung, welche am 25. Mai 2018 in der gesamten Europäischen Union (EU) eingeführt wurde. Ziel ist es, den Schutz personenbezogener Daten zu stärken und die Privatsphäre der EU-Bürger*innen zu schützen.

Bei der Verwendung von Websites werden wie bereits in Kapitel 5.6 Cookies nur nach ausdrücklicher Zustimmung gespeichert. Dabei handelt es sich mitunter um eine Auflage aus der DSGVO. Da das Lernmanagementsystem Zugriff auf personenbezogene Daten aus dem Datenbankausschnitt des LFV Steiermark hat, müssen diese entsprechend verarbeitet werden. Grundsätzlich müssen folgende Punkte gewährleistet sein:

- Rechtmäßigkeit und Zweckbindung (Artikel 5 (1) und Artikel 6)
- Datenminimierung (Artikel 5 (1c))
- Speicherbegrenzung und Löschung (Artikel 5 (1e))
- Sicherheitsmaßnahmen (Artikel 32)
- Transparenz und Betroffenenrechte (Artikel 12 bis 22 und Artikel 34)

³³ Vgl. Kinsta Inc. (2023): Online-Quelle [11.09.2024].

Die Rechtmäßigkeit und Zweckbindung stellt keine große Problematik dar, da alle Mitglieder des LFV Steiermark einer sachgemäßen Verwendung ihrer Daten zugestimmt haben. Eine Datenminimierung hat bereits im Zuge der Anmeldekriterien stattgefunden (siehe Kapitel 6.6.1.1).³⁴

Nach einer erfolgreichen Authentifizierung werden bei allen weiteren Datenverarbeitungsschritten im LMS hauptsächlich der Vor- und Nachname sowie die Standesbuchnummer benötigt.

Der Punkt Speicherbegrenzung und Löschung beim LMS trifft nur auf die Auswertung des Lernfortschrittes und der Prüfungsergebnisse zu. Sechs Monate nach Anmeldung am Lernmanagementsystem werden die gespeicherten Daten automatisch gelöscht (siehe Kapitel 4.2.2). Dadurch wird zusätzlich einem ständigen Wachsen der Datenbank des LMSs entgegengewirkt.

Bei den Sicherheitsmaßnahmen wird hier auf eine einfache Verschlüsselung zurückgegriffen. Dabei werden der Vor- und Nachname durch eine Zahlenkombination ersetzt und verschlüsselt. Der Algorithmus wurde im entsprechenden PHP-Plugin implementiert (siehe Kapitel 6.6.3). Der größte Vorteil liegt hierbei in der Trennung des FDISK von der LMS-Datenbank.

Im Infobereich des Lernmanagementsystems gibt es eine Erklärung darüber, wie mit den **personenbezogenen** Daten umgegangen wird (siehe Kapitel 6.7.1). Sollte sich ein Mitglied dazu entscheiden, seine Feuerwehr zu verlassen, so werden seine*ihre Informationen zuerst aus dem FDISK gelöscht. Spätestens nach sechs Monaten würden aber auch die pseudonymisierten Daten aus der LMS-Datenbank automatisch entfernt werden. Ein Bonus ist hier die Kombination aus FDISK und LMS-Datenbank, da es nicht möglich ist, sich ohne FDISK-Eintrag am Lernmanagementsystem anzumelden.

Die nächsten drei Abschnitte erläutern vereinfacht, welche DSGVO Artikel vom LFV Steiermark als "unbedingt einzuhalten" eingestuft wurden. Zusätzlich wurden noch die Artikel 33 bis 36 bei der praktischen Umsetzung der Masterarbeit berücksichtigt.

6.3.1 DSGVO Artikel 30

Um den Artikel 30 der DSGVO erfüllen zu können, muss ein Verzeichnis mit allen Tätigkeiten, bei denen personenbezogene Daten verarbeitet werden, geführt werden. Es müssen der Zweck der Verarbeitung, die Kategorie der betroffenen Daten (Name, Geburtsdatum, etc.), Empfänger und geplante Löschfristen angegeben werden. Dieser Artikel dient der Transparenz und Nachweisführung gegenüber der zuständigen Datenschutzbehörde.³⁵

6.3.2 DSGVO Artikel 32

Damit Risiken für die Rechte und Freiheiten von betroffenen Personen minimiert werden können, müssen geeignete technische und organisatorische Maßnahmen getroffen werden. Dazu zählt zum Beispiel die

³⁴ Vgl. intersoft consulting services AG (2021): Online-Quelle [25.08.2024].

³⁵ Vgl. Intersoft consulting services AG (2021): Online-Quelle [25.08.2024].

Pseudonymisierung des Vor- und Nachnamens, aber auch die regelmäßige Überprüfung der Sicherheitsmaßnahmen.³⁶

6.3.3 DSGVO Artikel 33, 34, 35 und 36

Die Artikel 33, 34, 35 und 36 der DSGVO beziehen sich auf folgende Punkte, auf die nicht genauer eingegangen wird:³⁷

- Meldung von Verletzungen des Schutzes personenbezogener Daten an die Aufsichtsbehörde
- Benachrichtigung der von einer Verletzung des Schutzes personenbezogener Daten betroffenen Person
- Datenschutz-Folgeabschätzungen
- Vorherige Konsultation

6.4 Lokale Entwicklungsumgebung

Die Wahl der richtigen Entwicklungsumgebung ist für Projekte wie das Lernmanagementsystem ein entscheidender Faktor. Dabei geht es nicht nur um die Kosten, sondern auch um die Verbreitung der Software. Es wurden ausschließlich die beiden Programme Local® und PhpStorm® zur Erstellung der Website verwendet. Für die Datenbank im Hintergrund wurde die MySQL Workbench eingesetzt.

6.4.1 Local®

Um den in Kapitel 6.6 beschriebenen Aufbau der Website umsetzen zu können, wurde ein Host benötigt. Zum einen können Hostsysteme mit dazu gehörendem Webspace gemietet werden, zum anderen gibt es aber auch die Möglichkeit, die Freeware Local® zu verwenden (siehe Abbildung 20).

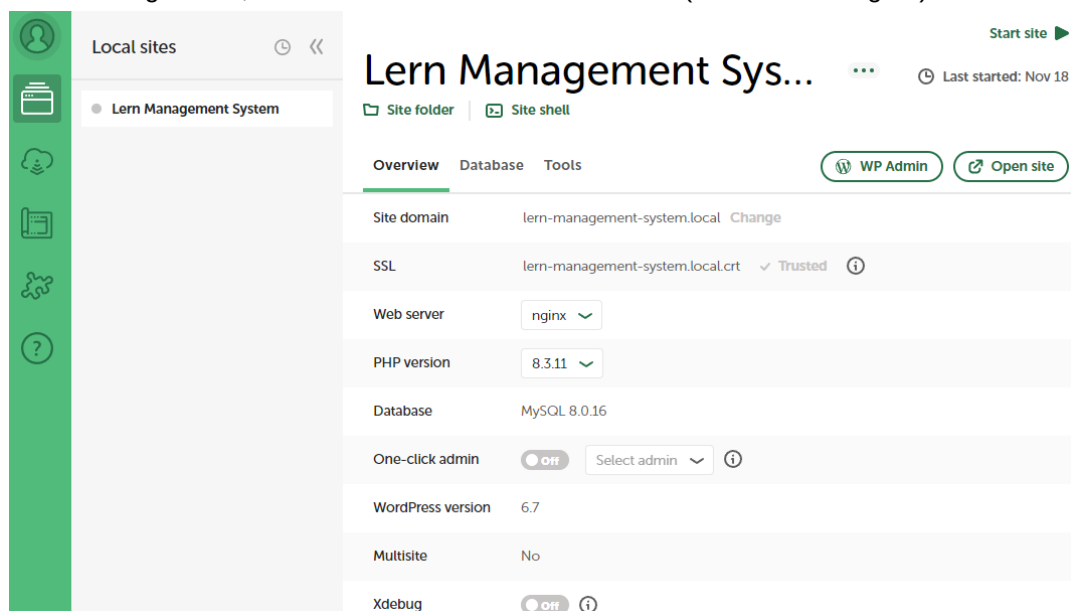


Abbildung 20: Darstellung Local® des Lernmanagementsystems, Quelle: Eigene Darstellung.

³⁶ Vgl. intersoft consulting services AG (2021): Online-Quelle [25.08.2024].

³⁷ Vgl. intersoft consulting services AG (2021): Online-Quelle [25.08.2024].

Local® ist eine Software, die speziell entwickelt wurde, um eine lokale Entwicklungsumgebung für WordPress bereitzustellen. Sie ermöglicht es Entwicklern und Designern, WordPress-Websites direkt auf deren Computer zu kreieren und zu testen, ohne dass eine Internetverbindung oder ein Live-Server erforderlich ist.

Die Hauptfunktion von Local® besteht darin, eine schnelle und benutzerfreundliche Möglichkeit zu bieten, eine lokale WordPress-Installation zu erstellen. Die Software kümmert sich automatisch um die Einrichtung von Webservern (wie Apache oder Nginx), Datenbanken (wie MySQL) und PHP-Versionen, sodass der Nutzer diese nicht manuell konfigurieren muss. Dadurch ist es möglich, verschiedene WordPress-Projekte parallel zu verwalten und zu entwickeln, ohne den Aufwand einer komplexen Serverkonfiguration. Ein weiteres wichtiges Merkmal ist die Integration von SSL-Zertifikaten, die es ermöglicht, lokale Websites unter HTTPS zu testen. Local® unterstützt außerdem den einfachen Wechsel zwischen verschiedenen PHP-Versionen und Webserver-Konfigurationen, was für die Kompatibilitätsprüfung oder das Debugging hilfreich ist. Die Software bietet eine One-Click-Deployment-Funktion, mit der lokale Websites schnell zu einem Live-Server migriert werden können, sei es auf einen selbst gehosteten Server oder zu Managed-WordPress-Hosting-Plattformen wie Flywheel oder WP-Engine. Dadurch entfällt der manuelle Prozess der Datenübertragung und des Datenbankexports. Zusätzlich verfügt Local® über Werkzeuge für die Datenbanksicherung und -wiederherstellung sowie Debugging-Tools für die Fehlersuche in WordPress. Es unterstützt Add-ons, mit denen Nutzer zusätzliche Funktionen integrieren können, wie etwa die Integration von Performance-Testing-Tools oder Entwicklungstools für bestimmte WordPress-Plugins oder -Themes. Insgesamt macht Local® die Entwicklung und Verwaltung von WordPress-Seiten einfacher, schneller und effizienter, insbesondere für Entwickler, die lokal an Projekten arbeiten möchten, bevor sie auf einen Live-Server migrieren.³⁸

³⁸ Vgl. WP Engine Inc. (o.J.): Online-Quelle [15.09.2024].

6.4.2 PhpStorm®

PhpStorm® ist eine kostenpflichtige integrierte Entwicklungsumgebung (IDE), die speziell für die Entwicklung von PHP-Projekten entwickelt wurde. Sie bietet eine Vielzahl von Funktionen, die Entwicklern helfen, effizienter zu programmieren. Dazu gehört ein Code-Editor, der intelligente Code-Vervollständigung, Syntax-Highlighting und Fehlererkennung für PHP bereitstellt (siehe Abbildung 21). PhpStorm unterstützt außerdem moderne PHP-Frameworks wie Laravel, Symfony oder Zend, was die Entwicklung in diesen Frameworks beschleunigt.

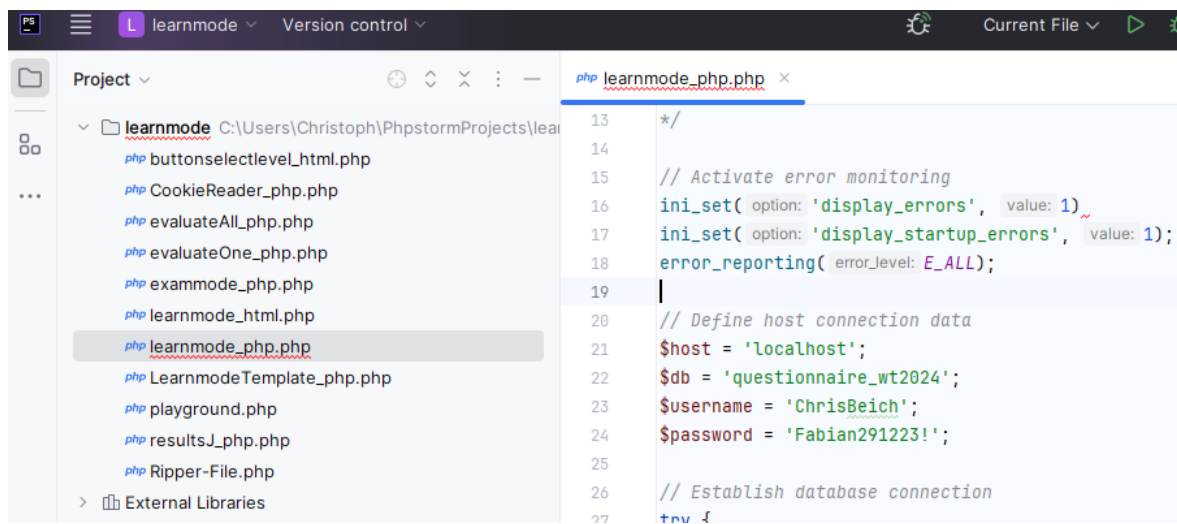


Abbildung 21: Darstellung der IDE PhpStorm® mit Syntax-Highlighting und Fehlererkennung, Quelle: Eigene Darstellung.

PhpStorm® bietet auch die Möglichkeit, mit Datenbanken wie MySQL oder PostgreSQL zu interagieren. Dadurch ist es möglich, SQL-Abfragen direkt in der IDE auszuführen und auszuwerten. Darüber hinaus unterstützt die IDE Test-Frameworks wie PHPUnit und ein eingebautes Debugging-Tool, das die Fehlerbehebung vereinfacht. Die Verwaltung von Codeänderungen und Branches wird durch die reibungslose Einbindung von Versionskontrollsystemen wie Git oder SVN erleichtert.

Des Weiteren verfügt PhpStorm® über die Unterstützung von HTML, CSS, JavaScript und anderen Webtechnologien, wodurch es eine vielseitige Lösung für die Full-Stack-Webentwicklung darstellt. Die Integration von Composer erleichtert die Verwaltung von PHP-Abhängigkeiten, und PhpStorm® bietet zusätzlich Tools wie PHP CS Fixer zur Einhaltung von Coding-Standards. PhpStorm stellt auch Funktionen für das direkte Deployment auf Servern zur Verfügung und unterstützt Remote-Entwicklung und Debugging, was es zu einer umfassenden Lösung für die Entwicklung moderner Webanwendungen macht. In Kombination mit der in Abschnitt 6.4.1 vorgestellten Software Local® ist PhpStorm® eine nahezu perfekte Lösung, um Webanwendungen wie dieses LMS zu erstellen.³⁹

Neben den bereits genannten Funktionalitäten bietet die IDE zusätzlich noch die Möglichkeit, eine direkte Verbindung zu allen gängigen Datenbankvarianten aufzubauen. Dies stellt einen enormen Vorteil beim Debuggen von Skripten dar, die auf Datenbanken zugreifen und deren Inhalte manipulieren. Dadurch konnten die PHP-Skripte, welche eine Datenbankverbindung benötigen, direkt in der Entwicklungsumgebung getestet werden.

³⁹ Vgl. JetBrains s.r.o. (o.J.): Online-Quelle [15.09.2024].

6.5 PHP und HTML

Damit ein erfolgreicher Webauftritt gelingt, wird die Programmiersprache PHP und die Auszeichnungssprache HTML benötigt. Gemeinsam strukturieren und gestalten sie die Inhalte einer Website.

6.5.1 Hypertext Preprocessor (PHP)

Der Hypertext Preprocessor (PHP) ist eine Skriptsprache, die hauptsächlich für die Erstellung dynamischer Webseiten oder Webanwendungen genutzt wird und eine Syntax hat, die an C und Perl angelehnt ist.

Der Server führt die Sprache aus und das Resultat, in der Regel in HTML-Form, wird dem Browser des Benutzers übermittelt. PHP lässt sich direkt in HTML integrieren, wodurch die Integration dynamischer Inhalte in Webseiten erleichtert wird. PHP bietet die Möglichkeit zur Interaktion mit Datenbanken, was die Datenverwaltung und Datenverarbeitung vereinfacht (siehe Abbildung 22). Außerdem kann PHP auf zahlreichen Webservern und Betriebssystemen verwendet werden und wird von einer großen Community von Entwicklern unterstützt. Die Sprache ist vergleichsweise unkompliziert zu erlernen und ermöglicht es gleichzeitig, komplexe Webanwendungen flexibel und effizient zu entwickeln.⁴⁰ Für die Programmierung der im Lernmanagementsystems verwendeten Plugins wurde die neueste PHP-Version (Version 8) verwendet. Bereits seit Version 5 unterstützt PHP die objektorientierte Programmierung (OOP).

```
// Establish database connection
try {
    $pdo = new PDO( dsn: "mysql:host=$host;dbname=$db", $username, $password);
    $pdo->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Datenbankverbindung fehlgeschlagen: " . $e->getMessage());
}
```

Abbildung 22: Beispiel einer Datenbankinteraktion mit PHP, Quelle: Eigene Darstellung.

⁴⁰ Vgl. one.com Group AB (o.J.): Online-Quelle [15.09.2024].

6.5.2 Hypertext Markup Language (HTML)

Die Hypertext Markup Language ist ein Markup-Tool, das zur Erstellung der Struktur und des Inhalts von Websites genutzt wird. Durch die Definition von Textelementen wie Überschriften, Absätzen, Listen, Links, Bildern und Formularen erklärt sie die grundlegenden Elemente einer Webseite (siehe Abbildung 23). Um dem Browser zu zeigen, wie spezifische Inhalte dargestellt werden sollen, benutzt HTML sogenannte Tags. Diese Tags bestimmen, welche Abschnitte des Textes beispielsweise als Überschrift, Absatz oder Bild zu sehen sind. HTML ist selbst statisch, was bedeutet, dass der Inhalt, den der Benutzer sieht, immer gleich bleibt, sofern er nicht von anderen Technologien wie CSS oder JavaScript modifiziert wird. HTML, die Standardsprache des Internets, bildet die Basis für die Erstellung von Webseiten.⁴¹

```

264 <!DOCTYPE html>
265 <html lang="de">
266 <head>
267   <meta charset="UTF-8">
268   <title>Fragen für Wissenstest</title>
269   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
270   <style>
271     <!-- Define style of Q&A area including the 3 buttons -->
272     .question-container {
273       margin-top: 50px;
274     }
275     .answer {
276       display: none;
277       margin-top: 20px;
278       font-weight: bold;
279       color: #007bff;
280     }
281     .btn-previous, .btn-next, .btn-answer {
282       margin-top: 20px;
283       width: 150px;
284       font-size: 1.2rem;
285       background-color: #6c757d;
286       color: white;
287       border: none;
288       border-radius: 5px;
289       text-align: center;
290       text-decoration: none;
291       display: inline-block;
292     }
293     .btn-back {
294       width: 200px;
295       font-size: 1.1rem;
296       font-weight: bold;
297       background-color: #e57373;
298       color: white;

```

Abbildung 23: Darstellung eines HTML-Codes aus dem LMS, Quelle: Eigene Darstellung.

⁴¹ Vgl. SEO-Küche Internet Marketing GmbH & Co. KG (2023): Online-Quelle [15.09.2024].

6.6 Plugin und Skripte

Den Kern des Lernmanagementsystems bilden neben den Datenbanken die eigens entwickelten Plugins und Skripte. Plugins sind Funktionen, welche in eine Website integriert werden. Diese können, wie in Kapitel 6.5.1 beschrieben, in der Skriptsprache PHP erstellt werden. Daneben gibt es für dynamische Anwendungen noch JavaScript als Programmiersprache. Neben Eigenkreationen kann online auch auf öffentlich zugängliche oder käufliche Plugins zurückgegriffen werden. Generell werden Skripts in Englisch verfasst, da dies den Lesefluss zwischen den englischsprachigen Befehlen und Funktionen weniger beeinträchtigt. Das Plugin für die Benutzeranmeldung wurde als solches in WordPress® implementiert, da auf diese Weise die Interaktion mit der CSV-Datei bewerkstelligt werden kann. Dagegen wurden alle anderen Skripte über das Plugin "PHP Code Snippets" als sogenannte Snippets eingearbeitet. Der Vorteil dieser kompakten Programme ist, dass sie über einen Shortcode an jeder beliebigen Stelle der Website eingepflegt werden können.

Die in diesem Kapitel beschriebenen Plugins und Skripte wurden eigens für das LMS entwickelt, um den hohen Anforderungen des LFV Steiermark zu entsprechen.

6.6.1 Plugin: Benutzeranmeldung

Der Grundstein des Lernmanagementsystems wird mit der Benutzeranmeldung gelegt. Die Vorgabe des LFV Steiermark war es, eine nicht-passwortbasierte Benutzerauthentifizierung zu verwenden. Die aus dem FDISK bereitgestellten Nutzerdaten wurden bereits in den Unterkapiteln des Kapitels 4.1 detailliert betrachtet.

6.6.1.1 Anmeldekriterien

Um eine*n Nutzer*in nun eindeutig mit den zur Verfügung gestellten Daten identifizieren zu können, wurden folgende fünf Eingabekriterien ausgewählt:

- Vorname
- Nachname
- Feuerwehr
- Funktion
- Standesbuchnummer

Zum einen handelt es sich hierbei um Werte, die sowohl von dem*r eigentlichen Nutzer*in aber auch von Fremden eingegeben werden können. Interessant wird es jedoch bei der Standesbuchnummer. Die Standesbuchnummer ist eine Nummer, die in jeder Feuerwehr des LFV Steiermark exakt einmal vorkommen kann. Dies bedeutet, dass es theoretisch in jeder Wehr einen FDISK-Eintrag mit der Standesbuchnummer "1" geben kann (siehe Abbildung 24).

Die Standesbuchnummer ist generell nur dem jeweiligen Feuerwehrmitglied und den Funktionären*innen der eigenen Feuerwehr, welche Zugriff auf das FDISK haben, bekannt. Dadurch wird es nahezu unmöglich, sich mit den Daten von jemand anderen anzumelden.

Standesbuchnummer	FUBWEHRSTB	instanzart	instanzname
1	41003001	FF	Deutschfeistritz
1	41019001	FF	Haselbach
1	41041001	FF	Stattegg
1	41042001	FF	Steinberg-Rohrbach
1	41044001	FF	St. Marein b. Graz
1	41052001	FF	Vasoldsberg
1	42017001	FF	Pernegg a.d. Mur
1	44013001	FF	Graggerer
1	44027001	FF	Michlgleinz

Abbildung 24: Darstellung der CSV-Datei aus dem Auszug des FDISK, Quelle: Eigene Darstellung.

In Kombination mit dem Vor- und Nachnamen sowie der zugehörigen Feuerwehr kann der*die Nutzer*in eindeutig identifiziert werden. Zusätzlich gibt es noch die Abfrage der Funktion, mit der zwischen einem Jugendmitglied, einer*m Feuerwehrjugend-Betreuer*in oder der*m Orts-, Abschnitts-, Bereichs-, Landes-Jugendbeauftragten*m unterschieden wird. Dies ist insofern wichtig, da der*die Nutzer*in auf unterschiedliche Websites zugreifen kann. Auch das Hauptmenü weist andere Funktionalitäten auf (siehe Kapitel 6.7.2).

6.6.1.2 Funktionsbeschreibung der Benutzeranmeldung

Zu Beginn des Benutzeranmeldungs-Skripts wird wie bei allen sonstigen Anwendung des LMSs mit dem Verbindungsaufbau zur MySQL Datenbank gestartet (siehe Abbildung 25). In den nachfolgenden Kapiteln wird deshalb nicht mehr näher auf diesen Teil des jeweiligen Skripts eingegangen.

```
// Establish connection to database
$host = 'localhost';
$db = 'questionnaire_wt2024';
$username = 'ChrisBeich';
$password = 'XXXXXXXXXX!';

// Try to connect with database with given information
try {
    $pdo = new PDO( dsn: "mysql:host=$host;dbname=$db", $username, $password);
    $pdo->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    // If database connection could not get established -> send error message to screen
    die("Datenbankverbindung fehlgeschlagen: " . $e->getMessage());
}
```

Abbildung 25: Verbindungsaufbau mit der MySQL Datenbank, Quelle: Eigene Darstellung.

Durch die Variablendeklaration am Anfang werden die für die Verbindungsherstellung notwendigen Parameter gesetzt. Da zu diesem Zeitpunkt nicht sichergestellt werden kann, ob eine Datenbankverbindung überhaupt möglich ist, kommt an dieser Stelle eine try-catch-Anweisung zum Einsatz. Es wird versucht, mit den zuvor deklarierten Variablen eine Verbindung mit der MySQL Datenbank herzustellen. Sollte der Verbindungsaufbau aus welchen Gründen auch immer fehlschlagen, so kommt die catch-Bedingung zur Anwendung. Dabei beendet die die-Funktion das PHP-Skript sofort und gibt den Text "Datenbankverbindung fehlgeschlagen: ", gefolgt von der Fehlermeldung, welche ferner eine beschreibende Nachricht zum Fehler liefert. Um Zugriff auf die hinterlegte CSV-Datei mit den

Benutzerdaten zu erhalten, muss diese wie in Abbildung 26 gezeigt definiert werden. In diesem Fall verweist der Pfad auf eine CSV-Datei, die im Ordner dieses Plugins am lokalen FTP-Server abgelegt wurde.

```
// Path for CSV-file from LFV Steiermark
$csvFile = 'C:\Users\Christoph\Local Sites\lern-management-system\app\public\wp-content\plug:
```

Abbildung 26: Pfadbeschreibung des CSV-Files, Quelle: Eigene Darstellung.

Mit der ersten Funktion dieses Skripts wurde bereits eine Maßnahme für den Onlinebetrieb des LMSs getroffen. Es wird ein zufälliger 32 Zeichen langer Hexadezimalcode generiert und mit der return-Funktion am Ort des Funktionsaufrufs zurückgegeben und der Variablen `$sessionID` zugewiesen (siehe Abbildung 27).

```
// Function to generate a unique session ID - will be needed when the website goes online
no usages
function generateSessionID() {
    return bin2hex(random_bytes( 32)); // Generates a 32-character hexadecimal string
}
// Function for user authentication
no usages
function authenticateUser($name, $surname, $firebrigade, $registrynumber, $function, $csvFile): bool {
    if (!file_exists($csvFile)) {
        die("Die CSV-Datei konnte nicht geladen werden! " . $csvFile);
    }
}
```

Abbildung 27: Darstellung der Funktionen "Erstellen der Session ID" und "Authentifizieren des Benutzers",
Quelle: Eigene Darstellung

Der nachfolgende Block (siehe Abbildung 28) beinhaltet alle notwendigen Schritte, um den*die Benutzer*in, welcher durch das Eingabefenster (siehe Kapitel 6.7.3) versucht, sich anzumelden, zu authentifizieren. In dem ersten if-Befehl wird mit der fopen-Funktion versucht, die CSV-Datei im Lesemodus ('r') zu öffnen. Es kommt zu einem Rückgabewert an `$handle` der entweder ein Array oder ein FALSE sein kann. Kommt es zu einem erfolgreichen öffnen der CSV-Datei, so wird die if-Bedingung erfüllt denn der Wert ist ungleich ('!=') FALSE. Als nächstes wird mit der while-Schleife Zeile für Zeile aus der CSV-Datei gelesen. Damit die Datei richtig ausgelesen werden kann, muss noch das verwendete Trennzeichen der Spalten, ein Semikolon und die maximale Leselänge pro Zeile in Bytes übergeben werden. Es wird davon ausgegangen, dass pro ASCII-Zeichen ein Byte in Anspruch genommen wird, weswegen eine maximale Länge von 75 Zeichen verwendet wurde. Dies sollte alle Punkte der Anmeldekriterien (siehe 6.6.1.1) abdecken.

```
// Read from CSV-file
if (($handle = fopen($csvFile, mode: 'r')) != FALSE) {
    while (($data = fgetcsv($handle, length: 75, separator: ';')) != FALSE) {
        // Get data, convert them to lower string and remove all unnecessary spaces
        $csvName = trim(strtolower($data[0]));
        $csvSurname = trim(strtolower($data[1]));
        $csvFirebrigade = trim(strtolower($data[2]));
        $csvRegistryNumber = trim(strtolower($data[3]));
        $csvFunction = trim(strtolower($data[4]));
    }
}
```

Abbildung 28: Darstellung des ersten Teils des Datenabgleichs zwischen Benutzereingabe und CSV-Datei,
Quelle: Eigene Darstellung.

Zur Gewährleistung, dass alle Einträge der CSV-Datei richtig weiterverarbeitet werden können, wird beim Leseprozess mit der String-to-Lower-Funktion ("strtolower") sichergestellt, dass alle gelesenen Zeichen in Kleinbuchstaben gespeichert werden. Nachfolgend wird dieser Vorgang für alle im Anmeldefenster eingegebenen Daten wiederholt. Zusätzlich zur String-to-Lower-Funktion gibt es noch die trim-Funktion, welche alle überflüssigen Leerzeichen am Anfang und Ende des Strings entfernt.

Mit der zweiten if-Abfrage wird nun geprüft, ob die eingegebenen Daten mit den Daten der CSV-Datei übereinstimmen. Findet sich ein passender Datensatz in der CSV-Datei, wird TRUE aus der if-Funktion zurückgegeben und die while-Schleife dadurch unterbrochen (siehe Abbildung 29).

```
$inputName = trim(strtolower($name));
$inputSurname = trim(strtolower($surname));
$inputFirebrigade = trim(strtolower($firebrigade));
$inputRegistryNumber = trim(strtolower($registrynumber));
$inputFunction = trim(strtolower($function));

if ($csvName == $inputName &&
    $csvSurname == $inputSurname &&
    $csvFirebrigade == $inputFirebrigade &&
    $csvRegistryNumber == $inputRegistryNumber &&
    $csvFunction == $inputFunction) {
    fclose($handle);
    return true;
}
```

Abbildung 29: Darstellung des zweiten Teils des Datenabgleichs zwischen Benutzereingabe und CSV-Datei, Quelle: Eigene Darstellung.

Bei der Ausführung des Skripts wird, nachdem der Verbindungsaufbau zur Datenbank gelungen ist, damit begonnen, die Daten aus dem Anmeldefenster an Variable im Programm zu übergeben. Dazu wird die POST-Methode eingesetzt. Bei der POST-Methode handelt es sich um eine HTTP-Methode, die sehr häufig zur Datenübertragung zwischen Webbrowser und Server verwendet wird.⁴² Ob tatsächlich Daten von der Benutzeranmeldung übergeben wurden, überprüft die if-Abfrage (siehe Abbildung 30). Ist die Überprüfung in Ordnung, werden die Anmeldedaten aus dem POST-Array gelesen und an die entsprechenden Variablen übertragen. Bleibt ein Eingabefeld leer, wird der Dialog aktualisiert und der*die Benutzer*in zur erneuten Eingabe aufgefordert.

```
// Process data from POST-request
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $name = $_POST['name'];
    $surname = $_POST['surname'];
    $firebrigade = $_POST['firebrigade'];
    $registrynumber = $_POST['registrynumber'];
    $function = $_POST['function'];
}
```

Abbildung 30: Darstellung der Übergabe der Daten aus dem POST-Array, Quelle: Eigene Darstellung.

⁴² Vgl. IBM Deutschland GmbH (2024): Online-Quelle [01.11.2024].

Somit wurden alle vorbereitenden Maßnahmen für die Authentifizierung getroffen und es können nun die beiden Datensätze (Eingabe und CSV-File) miteinander verglichen werden. In einer neuen if-Abfrage, welche sich verschachtelt in der letzten befindet, werden Anfragen an die Datenbank gesendet. Solche Anfragen werden immer in zwei Stufen durchgeführt. Im ersten Teil, dem prepare, wird der Variablen \$stmt ein MySQL Befehl zur Überprüfung der*des Benutzers*in mit der Datenbanktabelle, zugewiesen. Der ausführende zweite Teil beinhaltet erst den execute-Befehl mit den zu verwendenden Informationen (siehe Abbildung 31).

```
// User authentication -> CSV X database
if (authenticateUser($name, $surname, $firebrigade, $registrynumber, $function, $csvFile)) {
    $stmt = $pdo->prepare( query: "SELECT id, TIMESTAMPDIFF(MINUTE, created_at, NOW()) AS minutes
    $stmt->execute([
        ':name' => $name,
        ':surname' => $surname,
        ':firebrigade' => $firebrigade,
        ':registrynumber' => $registrynumber,
        ':function' => $function
    ]);
}
```

Abbildung 31: Darstellung der Funktion Benutzerauthentifizierung, Quelle: Eigene Darstellung.

In der aktuellen Version des LMSs wurde zusätzlich ein Mechanismus integriert, der die verstrichene Zeit zwischen zwei Anmeldungen mit denselben Benutzerinformationen überprüft. Sollten zwei idente Authentifizierungen innerhalb von 30 Minuten stattfinden, so wird nur der Zeitstempel im entsprechenden Tabelleneintrag der Datenbank aktualisiert. Vergeht jedoch mehr Zeit, wird die Anmeldung nicht mehr als Wiederholung gewertet und ein neuer Eintrag erstellt. Dies verhindert zum einen, dass es in der Interims-Datenbank durch ungewolltes Abmelden oder Verbindungsproblemen zu vermehrten Einträgen bei einer Sitzung kommt (siehe Abbildung 32). Zum anderen hat es auch einen sicherheitstechnischen Hintergrund, da sich die Codierung der einzelnen Anmeldung regelmäßig ändert.

```
if ($existingUser && $existingUser['minutes_since_created'] < 30) {
    $updateStmt = $pdo->prepare( query: "UPDATE users SET created_at = NOW() WHERE id = :id");
    $updateStmt->execute([':id' => $existingUser['id']]);
    // Retrieve the existing session ID
    $sessionID = $existingUser['session_id'];
} else {
    $sessionID = generateSessionID(); // Generate a new session ID
    $insertStmt = $pdo->prepare( query: "INSERT INTO users (name, surname, firebrigade, registrynumber, function, session_id) VALUES (:name, :surname, :firebrigade, :registrynumber, :function, :session_id)");
    $insertStmt->execute([
        ':name' => $name,
        ':surname' => $surname,
        ':firebrigade' => $firebrigade,
        ':registrynumber' => $registrynumber,
        ':function' => $function,
        ':session_id' => $sessionID,
    ]);
}
```

Abbildung 32: Darstellung der Benutzerauthentifizierung und der Datenbankeintragung, Quelle: Eigene Darstellung.

Ist die Session-ID älter als 30 Minuten, werden die Anweisungen in der else-Block ausgeführt. Es wird eine neue Session-ID über die Funktion `generateSessionID` generiert (siehe Abbildung 32). Im Anschluss wird wie bereits zuvor beschrieben zuerst der Vorbereitungsbefehl, mit allen Daten die neu in der Datenbanktabelle angelegt werden sollen ausgeführt. Es folgt der execute-Befehl, durch den die Daten an die Datenbank übertragen werden (siehe Abbildung 32 und Abbildung 33).

id	name	surname	firebrigade	registrynumber	function	created_at	session_id
26	Max	MUSTERMANN	Tillmitsch	999	jff/jfm	2024-11-07 20:05:22	fcf30c744e9ee5f242ab11b8312363f3

Abbildung 33: Darstellung der Datenbanktabelle nach erfolgreicher Anmeldung, Quelle: Eigene Darstellung.

Dieser Zwischenschritt mit der unverschlüsselten Übertragung der Daten an die Datenbank wurde ausschließlich für diese Masterarbeit eingeführt. Wird das Lernmanagementsystem direkt in die IT-Landschaft des LFV Steiermark integriert, so wird das LMS einen Zugriff auf das FDISK bekommen. Dadurch wird der Abgleich mit der CSV-Datei entfallen und der Vergleich erfolgt direkt mit den Daten aus dem FDISK.

Nach einer erfolgreichen Authentifizierung erfolgt die Weiterleitung zum entsprechenden Dashboard. Hier gibt es derzeit nur eine Unterscheidung zwischen der*m Jugend-Feuerwehrfrau*mann sowie der*m Ortsjugendbeauftragte*n und der*m Feuerwehrjugend-Betreuer*in. Welche Funktionen später der*m Abschnitts-, Bereichs- und Landes-Jugendbeauftragte*n zur Verfügung stehen werden, werden erst vor der Publizierung des Lernmanagementsystems Ende 2025 definiert werden. Wird im Anmeldefenster im Dropdown-Menü der Funktionsauswahl die Funktion JFF/JFM (Jugend-Feuerwehrfrau / Jugend-Feuerwehrmann) ausgewählt, so wird der*die Benutzer*in zum Dashboard der Jugend weitergeleitet (siehe Abbildung 61). Sollte die Anmeldung fehlschlagen, wird über den echo-Befehl im letzten else-Block eine Ausgabe direkt im Browser ausgegeben (siehe Abbildung 34).

```
// Forwarding after successful comparison of user data
if ($function == "jff/jfm") {
    header( header: "Location: http://lern-management-system.local/dashboard-jugend");
    exit();
} elseif ($function == "orts-jugendbeauftragte*r" || $function == "feuerwehrjugend-betreuer*in") {
    header( header: "Location: http://lern-management-system.local/dashboard-funktionaere");
    exit();
}
} else {
    echo "Anmeldung fehlgeschlagen. Die eingegebenen Daten stimmen nicht mit der CSV-Datei überein.<br>";
}
```

Abbildung 34: Darstellung der Weiterleitung nach erfolgreicher Anmeldung, Quelle: Eigene Darstellung.

6.6.2 Skript: Datenbank Handshake

Wie bereits in Kapitel 6.6.1 beschrieben, wurde ein Zwischenschritt bei der Datenmanipulation eingeführt. Das in diesem Abschnitt beschriebene Skript beginnt an dem Punkt, wo später einmal die Interaktion zwischen dem FDISK und dem LMS stattfinden wird. Das bedeutet, dass beim Aufruf dieses Skripts zukünftig die verwendeten Daten nicht von der Datenbanktabelle aus Abbildung 33 kommen werden, sondern aus der Datenbank des LFV Steiermark. Da das LMS nicht alle Daten aus dem FDISK verwendet und selbst eine Datenbank besitzt, wird mit diesem Skript eine neue Datenbanktabelle befüllt. Zum einen

wird dadurch der Grundsatz der Datenminimierung nach DSGVO Artikel 5 (1c) gewahrt, zum anderen kommt zusätzlich eine Verschlüsselung (siehe Abschnitt 6.6.2.2) der Daten nach DSGVO Artikel 32 ins Spiel (siehe Kapitel 6.3.2).

6.6.2.1 Funktionsbeschreibung des Datenbank Handshake

Der Verbindungsaufbau zur Datenbank erfolgt wie bereits in Abbildung 25 gezeigt. Danach werden mit der try-catch-Methode, dem SELECT-Befehl und dem fetchAll-Befehl alle Datensätze der Datenbanktabelle users abgerufen und als \$users Array gespeichert. Es werden nur der Vor- und Nachname sowie die Feuerwehr des letzten Eintrages der gespeicherten Daten extrahiert. Nach der Extraktion wird durch den Einsatz einer foreach-Schleife das \$users Array zerlegt und jeder Eintrag in ein assoziatives Array übertragen. Die dadurch erhaltenen Werte der einzelnen \$user Arrays werden anschließend einer Variablen zugewiesen. Eine Besonderheit weist die letzte Zeile in Abbildung 35 auf. Bei der Variablendeklaration der Variablen \$codedsurname wird die Funktion vigenereEncrypt aufgerufen. Es werden der Funktion die Variable \$surname, als zu verschlüsselndes Element und der Schlüssel als Key übergeben. Sollte das LMS zu einem späteren Zeitpunkt mit dem FDISK verbunden werden, findet hier die letzte Interaktion der beiden Systeme statt. Danach wird keine Verbindung zum FDISK mehr benötigt.

```
// Get data from table 'users'
try {
    $stmt = $pdo->query( query: "SELECT id, name, surname, firebrigade FROM users");
    $users = $stmt->fetchAll( mode: PDO::FETCH_ASSOC);

    foreach ($users as $user) {
        $name = $user['name'];
        $surname = $user['surname'];
        $firebrigade = $user['firebrigade'];
        $codedsurname = vigenereEncrypt($surname, key: 'Floriani');
```

Abbildung 35: Darstellung der Datenextraktion aus der Datenbanktabelle und des Aufrufs der Verschlüsselungsfunktion, Quelle: Eigene Darstellung.

Der Abschnitt 6.6.2.2 beschreibt die detaillierten Hintergründe der Vigenère-Verschlüsselung. Die für die Verschlüsselung verantwortliche Funktion kann in zwei Teile unterteilt werden (siehe Abbildung 36). Im ersten Teil werden zur Vereinfachung des Verschlüsselungsalgorithmus der zu codierende Name und der Schlüssel mit strtoupper in einen String, der nur aus Großbuchstaben besteht, umgewandelt. Dies hat den Vorteil, dass bei der Codierung die Schreibweise nicht berücksichtigt werden muss. Ansonsten müsste zusätzlich bei jedem Zeichen geprüft werden, ob es sich um einen Groß- oder Kleinbuchstaben handelt, da bei Kleinbuchstaben vom ASCII-Wert die Zahl 97 abgezogen werden muss. Wird dieser kleine Unterschied nicht berücksichtigt, erfolgt nach der Decodierung die Rückgabe eines völlig falschen Wertes. Um die Länge der beiden an die Funktion übergebenen Strings zu ermitteln, wird mit strlen die Länge des Strings ermittelt und an die jeweilige Variable übergeben.

Die eigentliche Verschlüsselung findet im zweiten Teil der Abbildung 36 statt. Mithilfe der for-Schleife wird in Abhängigkeit der zuvor ermittelten Länge des zu verschlüsselnden Texts, dieser in seine Einzelteile zerlegt und im \$textChar Array abgelegt. Anschließend wird mit einer if-Abfrage und der ctype_alpha

Funktion überprüft, ob es sich bei jedem Element im `$textChar` Array um einen alphabetischen Buchstaben handelt. Es sei angemerkt, dass die `ctype_alpha`-Funktion nur Buchstaben von A bis Z (groß und klein geschrieben) erkennt. Umlaute, Akzente, diakritische Zeichen oder Sonderzeichen fallen nicht in den Definitionsbereich dieser Funktion. Ist diese Bedingung erfüllt, wird die Vigenère-Verschlüsselung des übergebenen Textes, wie in Abschnitt 6.6.2.2 beschrieben, für jeden Buchstaben durchgeführt. Der Rückgabewert der Funktion beinhaltet den zusammengesetzten verschlüsselten Text in Großbuchstaben.

```
// Vigenère-encryption
2 usages
function vigenereEncrypt($text, $key) {
    $text = strtoupper($text);
    $key = strtoupper($key);
    $keyLength = strlen($key);
    $textLength = strlen($text);
    $encryptedText = '';

    for ($i = 0; $i < $textLength; $i++) {
        $textChar = $text[$i];
        if (ctype_alpha($textChar)) {
            $textCharCode = ord($textChar) - 65;
            $keyCharCode = ord($key[$i % $keyLength]) - 65;
            $encryptedCharCode = ($textCharCode + $keyCharCode) % 26 + 65;
            $encryptedText .= chr($encryptedCharCode);
        } else {
            $encryptedText .= $textChar;
        }
    }
    return $encryptedText;
}
```

Abbildung 36: Darstellung der Funktion der Vigenère-Verschlüsselung, Quelle: Eigene Darstellung.

Da nun alle relevanten Daten, den dafür vorgesehenen Variablen übergeben wurden, kann mit dem Übertragen der Daten auf die für das LMS relevante Datenbanktabelle begonnen werden. Dafür werden zuerst wieder die Strings für den Namen, den verschlüsselten Nachnamen und die Feuerwehr mit `strtolower` zu Strings mit ausschließlich Kleinbuchstaben formatiert (siehe Abbildung 37). Danach wird die Tabelle `mappingtable_user` auf bereits idente vorhandene Einträge mit den Befehlen `prepare`, `SELECT` und `execute` überprüft (siehe Abbildung 38).

```
// Prepare data for database query
$lowercase_name = strtolower($name);
$lowercase_codedsurname = strtolower($codedsurname);
$lowercase_firebrigade = strtolower($firebrigade);
```

Abbildung 37: Darstellung der "String to lower" Funktion, Quelle: Eigene Darstellung.

Ob es eine Übereinstimmung in der Datenbanktabelle `mappingtable_user` gibt, wird mit einer if-Abfrage festgestellt. Sollte die Suche negativ verlaufen, wird der Tabelle eine neue Zeile mit den drei definierten Werten hinzugefügt.

```
if ($existingEntry) {
    // Refresh of parameters in case of identified user currently not needed
    $current_iduser = $existingEntry['iduser'];
    $updateStmt = $pdo->prepare( query: "
        UPDATE mappingtable_user
        SET name = :name, codedsurname = :surname, firebrigade = :firebrigade
        WHERE iduser = :iduser
    ");
    $updateStmt->bindParam( param: ':iduser', &var: $current_iduser, type: PDO::PARAM_INT);
    $updateStmt->bindParam( param: ':name', &var: $name);
    $updateStmt->bindParam( param: ':surname', &var: $codedsurname);
    $updateStmt->bindParam( param: ':firebrigade', &var: $firebrigade);
```

Abbildung 39: Darstellung Übergabe der Werte an die Datenbanktabelle `mappingtable_user`,
Quelle: Eigene Darstellung.

Dies geschieht mit den bereits bekannten Datenbankmechanismen `query`, `fetch`, `prepare` und `execute` (siehe Abbildung 39).

```
// Check for existing entries in 'mappingtable_user'
$checkStmt = $pdo->prepare( query: "
    SELECT iduser FROM mappingtable_user
    WHERE LOWER(TRIM(name)) = :name
    AND LOWER(TRIM(codedsurname)) = :codedsurname
    AND LOWER(TRIM(firebrigade)) = :firebrigade
");
$checkStmt->bindParam( param: ':name', &var: $lowercase_name);
$checkStmt->bindParam( param: ':codedsurname', &var: $lowercase_codedsurname);
$checkStmt->bindParam( param: ':firebrigade', &var: $lowercase_firebrigade);
$checkStmt->execute();
```

Abbildung 38: Darstellung der Datenbanktabellenabfrage, Quelle: Eigene Darstellung.

Sollte ein Fehler bei der Datenverarbeitung auftreten, so wird dieser durch den catch-Block am Ende dieses Skripts verarbeitet (siehe Abbildung 40). Es kommt dabei zu einer direkten Fehlerbeschreibung im Webbrowser.

```
} catch (PDOException $e) {
    echo "Fehler bei der Datenverarbeitung: " . $e->getMessage();
}
```

Abbildung 40: Darstellung des catch-Blocks am Skriptende, Quelle: Eigene Darstellung.

6.6.2.2 Vigenère-Verschlüsselung

Bei der Vigenère-Verschlüsselung handelt es sich um ein monografisches polyalphabetisches Substitutionsverfahren, welches bereits Anfang des 16. Jahrhundert vom deutschen Benediktinerabt Johannes Trithemius beschrieben wurde. Damals war es aber noch eine starre Verschlüsselung ohne Schlüssel. Die Grundidee wurde im Lauf des Jahrhunderts von mehreren Kryptografen aufgegriffen und

erweitert. Die entscheidende Idee zur Verwendung eines beliebigen Schlüssels wurde 1585 vom Franzosen Blaise de Vigenère vorgeschlagen.⁴³

Zur Verschlüsselung des Namens Mustermann wird ein Schlüssel (Key) mit dem Namen Student verwendet. Zuerst wird der Nachname Mustermann in seine Bestandteile zeilenweise aufgetrennt. Dasselbe passiert mit dem Key. Je nach Länge des gewählten Schlüssels kann sich dieser mehrfach wiederholen. Danach wird jeder Buchstabe des Nachnamens und des Keys entsprechend seiner Position im Alphabet in eine Zahl umgewandelt. Da unsere Computersysteme mit dem ASCII (American Standard Code for Information Interchange) als Zeichencodierungsschema arbeiten, wird nach der Umwandlung grundsätzlich dieser ASCII-Wert angezeigt. Damit der Vergleich mit dem bekannten Alphabet übereinstimmt, wird deshalb vom umgewandelten Zahlenwert der Wert 65 subtrahiert. Die dadurch neu erhaltenen Werte werden miteinander addiert, durch 26 dividiert und davon der ganzzahlige Rest als Verschiebung verwendet. Zur neu erhaltenen Zahl wird wiederum 65 addiert und der neue Wert anschließend als verschlüsselter Buchstabe dargestellt.

Die Tabelle 3 dient zur Veranschaulichung der Vigenère-Verschlüsselung.

Name	Key	Alphabet Name	Alphabet Key	Verschiebung	Codierung
M	S	12	18	4	E
U	T	20	19	13	N
S	U	18	20	12	M
T	D	19	3	22	W
...	...	...	...	...	...

Tabelle 3: Beispiel der Vigenère-Verschlüsselung, Quelle: Eigene Darstellung.

Folgende Berechnungsschritte werden für die Vigenère-Verschlüsselung benötigt. Als Beispiel werden die Werte der ersten Zeile der Tabelle berechnet.

Ermittlung des Stellenwertes des Buchstaben M und S im Alphabet mit ASCII-Codierung:

$$M_{ASCII} = 77$$

$$S_{ASCII} = 83$$

$$Alphabet_{Name} = M_{ASCII} - 65$$

$$Alphabet_{Name} = 77 - 65 = \mathbf{12}$$

$$Alphabet_{Key} = S_{ASCII} - 65$$

$$Alphabet_{Key} = 83 - 65 = \mathbf{18}$$

⁴³ Vgl. Studyflix GmbH (o.J.): Online-Quelle [09.11.2024].

Addition der beiden Werte:

$$\begin{aligned} Summe_{ASCII} &= M_{ASCII} + S_{ASCII} \\ Summe_{ASCII} &= 12 + 18 = \mathbf{30} \end{aligned}$$

Division durch die Anzahl an Buchstaben im Alphabet (26) und Ermittlung des ganzzahligen Rest, der Verschiebung:

$$\begin{aligned} Verschiebung &= (M_{ASCII} + S_{ASCII}) \bmod 26 \\ Verschiebung &= (12 + 18) \bmod 26 = \mathbf{4} \\ \mathbf{4} &\triangleq \mathbf{E} \end{aligned}$$

6.6.3 Skript: Lernmodus

Auch das Skript für den Lernmodus benötigt eine funktionierende Datenbankverbindung. Diese wird nach dem in Abbildung 25 gezeigten Schema umgesetzt. Da für jede Wissensteststufe dasselbe Grundkonstrukt dieses Codes benutzt wird, wurde die Stufe zu Beginn des Skripts hart codiert. Da die Verbindung zur FDISK Datenbank noch nicht besteht, wird für die Benutzerzuordnung der*die Benutzer*in mit der höchsten Benutzer-ID aus der Datenbanktabelle mappingtable_user verwendet. Nachdem die Datenbankverbindung hergestellt wurde, wird bei der Auswahl einer Wissensteststufe in der Spalte learnmodecount der Tabelle mappingtable_user der für diese*n User*in vorhandene Zellenwert um eins erhöht. Die dafür verwendete Methode sollte bereits aus den letzten Kapiteln bekannt sein (siehe Kapitel 6.6.1 und 6.6.2). Nachdem die Funktion incrementLearnmodeCount definiert wurde, wird diese direkt aufgerufen (siehe Abbildung 41).

```
// Increment learn mode count or initialize it to 1 if NULL
1 usage
function incrementUserCount($pdo, $iduser) {
    try {
        // Check if a value in learnmodecount is already existing
        $stmt = $pdo->prepare("SELECT learnmodecount FROM mappingtable_user WHERE iduser = :iduser");
        $stmt->execute([':iduser' => $iduser]);
        $currentLearnModeCount = $stmt->fetch(PDO::FETCH_ASSOC)['learnmodecount'];

        if ($currentLearnModeCount === null) {
            // If learnmodecount is NULL -> set value to 1
            $stmt = $pdo->prepare("UPDATE mappingtable_user SET learnmodecount = 1 WHERE iduser = :iduser");
        } else {
            // Else increase value by 1
            $stmt = $pdo->prepare("UPDATE mappingtable_user SET learnmodecount = learnmodecount + 1 WHERE iduser = :iduser");
        }
        $stmt->execute([':iduser' => $iduser]);
    } catch (PDOException $e) {
        echo "Fehler beim Aktualisieren von learnmodecount: " . $e->getMessage() . "<br>";
    }
}

// Call incrementUserCount function
incrementUserCount($pdo, $iduser);
```

Abbildung 41: Darstellung der Funktion incrementLearnmodeCount und Funktionsaufruf, Quelle: Eigene Darstellung.

Ein wichtiges Element des Lernmodus stellt die automatisierte Abfrage der in der gewählten Stufe vorhandenen Stationen dar. Mittels if-Abfrage wird geprüft, ob aktuell keine Station (!isset(...)) angewählt wurde. Ist diese Bedingung erfüllt, so werden aus der Datenbanktabelle mappingtable_wt alle Stationen die dem konstanten Wert von \$level entsprechen, herausgefiltert (siehe Abbildung 42).

```
// Show stations if no station was chosen
if (!isset($_POST['station_num'])) {
    // Get all stations for bronze level
    $stmt = $pdo->prepare( query: "SELECT DISTINCT station_number, station FROM mappingt
    $stmt->execute(['level' => $level]);
    $stations = $stmt->fetchAll( mode: PDO::FETCH_ASSOC);
```

Abbildung 43: Ausschnitt Stationsabfrage der jeweiligen Wissenstest Stufe, Quelle: Eigene Darstellung.

Nach der erfolgreichen Abfrage beinhaltet dieser if-Block einen HTML-Code. In diesem Code ist die Darstellung der Auswahl Schaltflächen (Buttons) definiert. Bei der Erstellung einer Visualisierung mit HTML müssen einige Punkte beachtet werden. In der Programmierung wird zuerst mit der Definition der einzelnen Elemente, die später verwendet werden sollen, begonnen. Erst am Ende des HTML-Skripts werden diese Einzelteile dann zu einer anschaulichen Benutzeroberfläche zusammengesetzt. Für die Ansicht der Stationsauswahl wurden zuerst zwei Container definiert. Im Allgemeinen beinhaltet ein Container alle relevanten Informationen zur Positionierung, Größe, Strukturierung und dem Design der später inkludierten Elemente. Der `main-container` dient als transparenter Hauptrahmen der Struktur und sorgt für gleichmäßige Abstände (`padding`), während der `button-container` für das Arrangement der Buttons zuständig ist (siehe Abbildung 43).

```
<style>
    .main-container {
        padding-top: 40px;
        padding-bottom: 40px;
        background-color: transparent;
    }
    .button-container {
        display: flex;
        flex-direction: column;
        align-items: center;
        gap: 10px;
        margin-top: 20px;
    }
    .btn-station {
        width: 300px;
        font-size: 1.2rem;
        background-color: #6c757d;
        color: white;
        border: none;
        border-radius: 5px;
    }
}
```

Abbildung 42: Darstellung des HTML-Teils zur Definition der verwendeten Buttons, Quelle: Eigene Darstellung.

Für das Design der Schaltflächen wurden drei unterschiedliche Formatierungsblöcke angelegt. Es gibt einen Block für die Stations-Buttons und zwei für die Navigationsbuttons, wobei der Abschnitt `navigation-buttons` als Container für die zwei Buttons dient.

Zum Schluss wurden nun die Elemente wie in Abbildung 43 ersichtlich zusammengefügt. Um die Stations-Buttons richtig darstellen zu können, wurde im HTML-Skript ein PHP-Skript integriert, mit dem für jeden Stationseintrag in der Datenbanktabelle eine eigene Schaltfläche angezeigt wird. Damit die beiden Navigationsbuttons auch die Aufgabe der Weiterleitung erfüllen können, müssen ihnen noch die entsprechenden Links und eine Beschriftung hinzugefügt werden (siehe Abbildung 44 und Abbildung 64).

```
<!-- Identify the stations stored in the database -->
<form method="POST" action="">
  <div class="button-container">
    <?php foreach ($stations as $station): ?>
      <button type="submit" name="station_num" value="<?= htmlspecialchars($station['station_number'])
        <?= htmlspecialchars($station['station']) ?>
      </button>
    <?php endforeach; ?>
  </div>
</form>

<!-- Additional navigation buttons -->
<div class="navigation-buttons">
  <a href="http://lern-management-system.local/lernmodus/" class="btn btn-nav">Auswahl Wissenstest</a>
  <a href="http://lern-management-system.local/dashboard-jugend/" class="btn btn-nav">Lernmodusauswahl</a>
</div>
```

Abbildung 44: Darstellung des HTML-Teils zur Generierung der verwendeten Buttons,
Quelle: Eigene Darstellung.

Nachdem eine Station aus dem Wissenstest ausgewählt wurde, findet eine Überprüfung des letzten Lernfortschritt-Eintrages der*s Benutzer*in statt. Dadurch wird gewährleistet, dass der*die User*in nach einer Unterbrechung des Lernmodus wieder an derselben Stelle weitermachen kann. Dies geschieht mit den bereits bekannten Befehlen zur Datenbankabfrage und der Datenbanktabelle `user_progress`. Daneben gibt es auch eine eigene Funktion mit denselben Mechanismen zum Auslesen der entsprechenden Fragen und Antworten aus der `mappingtable_wt` Tabelle (siehe Abbildung 45).

```
// Load questions for the selected station
2 usages
function getQuestions($pdo, $level, $station_num) {
    $stmt = $pdo->prepare("SELECT idwt, question, answer FROM mappingtable_wt WHERE
    $stmt->execute(['level' => $level, 'station_num' => $station_num]);
    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

$questions = getQuestions($pdo, $level, $station_num);
// Get user progress and refresh data in database
$stmt = $pdo->prepare( query: "SELECT * FROM user_progress WHERE user_id = :user_id AND
$stmt->execute(['user_id' => $iduser, 'station_num' => $station_num, 'level' => $level]);
$progress = $stmt->fetch( mode: PDO::FETCH_ASSOC);

// Check if progress for this station is existing -> create new entry if not
if(!$progress) {
    // Create new entry in database for user progress
    $currentQuestionIndex = 1;
    $stmt = $pdo->prepare( query: "INSERT INTO user_progress (user_id, station_num, current_question_index)
    $stmt->execute(['user_id' => $iduser, 'station_num' => $station_num, 'current_question_index' => $currentQuestionIndex]);
} else {
    // Load progress if entry is existing
    $currentQuestionIndex = $progress['current_question_index'];
}
```

Abbildung 45: Darstellung der Datenbankabfragen für den Lernmodus, Quelle: Eigene Darstellung.

Im Lernmodus gibt es in der Anzeigeebene neben der Fragennummer der Fragestellung und der Lösung noch zusätzlich zwei Navigations-Schaltflächen und eine Antwort-Schaltfläche (siehe Abbildung 65). Mit diesen Buttons kann zwischen den einzelnen Fragen hin- und hergeschaltet und die Antwort eingeblendet werden. Um diese Funktionalitäten zu integrieren, wird eine verschachtelte if-Abfrage benötigt (siehe Abbildung 46).

```
// Identification of which button got pressed (previous, answer, forward)
if (isset($_POST['action'])) {
    if ($_POST['action'] == 'previous') {
        $currentQuestionIndex = max( value: 1, ...values: $currentQuestionIndex - 1); // Avoids a qu
    } elseif ($_POST['action'] == 'next') {
        // Check if current question is last question of station
        if ($currentQuestionIndex >= count($questions)) {
            echo "Letzte Frage erreicht. Eintrag wird gelöscht.";
            $pdo->beginTransaction();
            try {
                // Delete values for station_num and current_question_index after last question
                $stmt = $pdo->prepare( query: "DELETE FROM user_progress WHERE user_id = :user_id
                $stmt->execute(['user_id' => $idUser, 'station_num' => $station_num, 'level' =>
                $pdo->commit();

                // This header is needed to move back to the station overview after the last Q
                header( header: "Location: http://lern-management-system.local/lernmodus-wissens
                exit();
            } catch (PDOException $e) {
                $pdo->rollBack();
                die("Fehler beim Löschen des Fortschritts: " . $e->getMessage());
            }
        }
    } else {
        $currentQuestionIndex = min(count($questions), ...values: $currentQuestionIndex + 1);
    }
}
```

Abbildung 46: Darstellung der verschachtelten if-Abfrage, Quelle: Eigene Darstellung.

Da die Aktionen der Schaltflächen “Weiter” und “Zurück” im Array `$_POST` gespeichert ist, muss in den verschachtelten if-Abfragen Folgendes geprüft werden:

- Wurde einer der beiden Navigationsbuttons gedrückt?
- Wurde der Zurückbutton gedrückt?
- Wurde der Weiterbutton gedrückt?
- Handelt es sich bei der derzeitigen Frage um die erste oder letzte Frage?

Dem gedrückten Button entsprechend wird der `$currentQuestionIndex` entweder um eins erhöht oder gesenkt, insofern die aktuelle Fragestellung weder die erste noch letzte Frage in ihrer Station ist. Sollte es sich um die Erste handeln, so wird der Zurückbutton deaktiviert und anders dargestellt. Wird der Weiterbutton bei der letzten Fragestellung gedrückt, löscht der Datenbankbefehl im try-catch-Block die Werte für die Stationsnummer und den derzeitigen Fragenindex. Die Station kann dadurch als abgeschlossen angesehen werden.

Abschließend fehlt vom dynamisch gestalteten Teil des Lernmodus-Skripts noch die Überleitung der passenden Antwort. Da beim Wissenstest nicht ausschließlich textbasierte Lösungen, sondern auch bildbasierte und im Zuge des Upgrades durch das LMS ebenso videobasierte Antwortmöglichkeiten

realisiert wurden, muss an dieser Stelle ein spezieller Mechanismus implementiert werden. Wie bereits in Kapitel 4.2.3 und 6.1.4 beschrieben ist bei den Antworten der Datenbanktabelle `mappingtable_wt` nur der Pfad zur entsprechenden Datei hinterlegt. Die Differenzierung des Dateiformats geschieht in einer verschachtelten if-Abfrage. Es wird zwischen Bild- und Videodateien in den Formaten JPG, JPEG, PNG, GIF, BMP, MOV und MP4 unterschieden. Zusätzlich zur Unterscheidung wird außerdem die Größe der anzuzeigenden Elemente über den HTML-Code mit übergeben (siehe Abbildung 47).

```
// Check if the answer is a pic, vid or text
if (preg_match( pattern: '/\.(jpg|jpeg|png|gif|bmp)$/i', $currentAnswer)) {
    // Show picture
    $currentAnswer = "<img src='" . htmlspecialchars($currentAnswer) . "' alt='Antwort-Bild' s"
} elseif (preg_match( pattern: '/\.(mov|mp4)$/i', $currentAnswer)) {
    $currentAnswer = "<video controls style='max-width:100%; height:auto;' src='" . htmlspecialchars($currentAnswer)
} else {
    $currentAnswer = str_replace( search: '/', replace: '<br>', htmlspecialchars($currentAnswer));
}
```

Abbildung 47: Darstellung der Unterscheidung des Dateiformats der Antwort, Quelle: Eigene Darstellung.

Sowohl bei den Fragen als auch bei den Antworten gilt, wie in Kapitel 4.2.3 beschrieben, der Schrägstrich (Slash) als Trennzeichen. Bei der dementsprechenden Ausgabe wird dieser in beiden Fällen durch einen Zeilenumbruch ersetzt, um eine bessere Formatierung sicherzustellen.

Die Darstellung der Fragestellungen, der Antworten und der bereits beschriebenen Schaltflächen erfolgt ähnlich wie bei der Stationsauswahl. Um die Navigation zwischen den Stationen zu erleichtern, gibt es noch einen Button, mit dem auch zwischen den Fragen zur Stationsauswahl zurückgekehrt werden kann.

6.6.4 Skript: Prüfungsmodus

Der Prüfungsmodus wurde in Bezug auf die Grundlagen und den Aufbau an den Lernmodus angelehnt. Dadurch wird sichergestellt, dass es ein homogenes Gesamtbild auf der Website gibt. Da es sich beim Wissenstest nur um die eigentliche Fragestellung und deren korrekte Lösung handelt, mussten für den Prüfungsmodus zusätzlich drei falsche Antwortmöglichkeiten generiert werden. Diese wurden leicht von den richtigen Antworten abgeleitet, damit die Jugendlichen zum aufmerksamen Lesen aller Auswahlmöglichkeiten animiert werden.

6.6.4.1 Funktionsbeschreibung des Prüfungsmodus

Die bereits in Kapitel 6.6.1.2, 6.6.2.1 und 6.6.4.1 erläuterte Datenbankverbindung wird auch für den Prüfungsmodus benötigt, weshalb an dieser Stelle nicht mehr genauer darauf eingegangen wird. Nachdem die Verbindung hergestellt wurde, wird die zuletzt gültige Benutzer-ID aus der Datenbanktabelle `mappingtable_user` mittels einer try-catch-Anweisung geladen und vom Statement-Objekt `$idUserStmt` verarbeitet. Anschließend wird die Stufe des Wissenstests über die hart codierte Variable `$level` an die Tabelle übertragen.

Falls es einen Fehler mit der Datenbanktabelle geben sollte, wird entweder nur eine Ausgabe mit dem echo-Befehl erzeugt oder das Skript durch den die-Befehl im catch-Block direkt beendet (siehe Abbildung 48). Prinzipiell handelt es sich hier um einen ähnlichen Vorgang wie bei dem Lernmodus in Kapitel 6.6.3.

```
// Exam level
$level = 1;

// Get the latest user ID
try {
    $idUserStmt = $pdo->prepare( query: "SELECT MAX(iduser) AS max_iduser FROM mappingtable_user");
    $idUserStmt->execute();
    $maxIdUser = $idUserStmt->fetch( mode: PDO::FETCH_ASSOC)['max_iduser'];

    if ($maxIdUser != null) {
        // Update the jlevel entry in database with $j_level
        $updateLevelStmt = $pdo->prepare( query: "UPDATE mappingtable_user SET jlevel = :level WHERE iduser = :iduser");
        $updateLevelStmt->execute(['level' => $level, 'iduser' => $maxIdUser]);
    } else {
        echo "Kein Benutzer gefunden.";
    }
} catch (PDOException $e) {
    die("Fehler beim Setzen des Levels: " . $e->getMessage());
}
```

Abbildung 48: Darstellung der Benutzer-ID Abfrage, Quelle: Eigene Darstellung.

Die Auswahl der im Prüfungsmodus zu bewältigenden Station erfolgt ident wie im Lernmodus. Die Stationen werden aus der Datenbanktabelle mappingtable_wt ausgelesen und als Buttons angeordnet (siehe Kapitel 6.7.3.2). Der dazu verwendete HTML-Code entspricht dem aus dem Lernmodus.

Im Gegensatz zum Lernmodus müssen beim Prüfungsmodus die erreichten Punkte der Station mitgezählt werden. Dafür muss zum einen die maximal zu erreichende Punkteanzahl pro Station ermittelt werden und zum anderen müssen auch die Bewertungen aus den richtigen Antworten addiert und gesichert werden. Diese Werte werden in den Arrays \$max_points und \$total_points gespeichert (siehe Abbildung 49). Die Funktion zur Ermittlung der aktuellen Punktezahl darf nur bei der Auswahl der richtigen Antwort aufgerufen werden.

```
if (!isset($_POST['max_points'])) {
    $stmt = $pdo->prepare( query: "SELECT SUM(points) AS max_points FROM mappingtable_wt WHERE j_level = :level");
    $stmt->execute(['level' => $level, 'station_num' => $station_num]);
    $result = $stmt->fetch( mode: PDO::FETCH_ASSOC);
    $max_points = (int)$result['max_points'];
} else {
    $max_points = (int)$_POST['max_points'];
}

$stmt = $pdo->prepare( query: "SELECT question, answer, answer_no, points FROM mappingtable_wt WHERE j_level = :level AND station_num = :station_num");
$stmt->execute(['level' => $level, 'station_num' => $station_num]);
$allQuestions = $stmt->fetchAll( mode: PDO::FETCH_ASSOC);
```

Abbildung 49: Berechnung der Punktezahl im Prüfungsmodus, Quelle: Eigene Darstellung.

Neben dem Abrufen der Fragen und den dazugehörigen Antworten, welches im Kapitel 6.6.3 ausführlich beschrieben wurde, müssen im Prüfungsmodus auch die falschen Antwortmöglichkeiten aus der Datenbanktabelle gelesen werden. Um nun die Antwortmöglichkeiten bei jedem Aufruf durchzumischen, wird die shuffle-Funktion verwendet.

Das Mischen der potenziellen Lösungen stellt in erster Linie kein Problem dar, jedoch musste die neu generierte Anordnung der vermischten Antwortmöglichkeiten aus dem `$shuffledOrder` Array in der Datenbank zwischen gespeichert werden. Prinzipiell sind die ungültigen Lösungen immer mit einem F für FALSE in der Spalte `truefalse` der Tabelle `mappingtable_wt` markiert (vergleiche Abbildung 13).

Damit die gewünschte Funktionalität realisiert werden kann, gibt es eine verschachtelte if-Abfrage mit drei Instanzen. Als erstes wird der Fortschritt der Fragen aus der Datenbanktabelle `user_progress` und der Spalte `current_question_index` ausgelesen. Dadurch wird überprüft, ob der*die Benutzer*in bereits einen gespeicherten Fortschritt in der gewählten Station besitzt. Dies passiert, wenn die Variable `$progress` definiert ist und weder NULL noch FALSE zurückgibt. In der nächsten Stufe wird ermittelt, ob der Button "Weiter" nach der Auswahl einer Antwort gedrückt wurde. Wird die Schaltfläche betätigt, erhöht sich der `$currentQuestionIndex` um eins und es wird die nächste Frage aus der Tabelle `mappingtable_wt` abgerufen. Mit der letzten if-Abfrage wird das Ende der ausgewählten Station festgestellt. Solange der `$currentQuestionIndex` kleiner als die Anzahl der Fragen der gewählten Station ist, werden die Antwortoptionen mit der shuffle-Funktion in zufälliger Reihenfolge angeordnet (siehe Abbildung 50). Die Anordnung der Antworten wird in der Spalte `answer_order` der Datenbanktabelle `user_progress` abgelegt. Dieser Schritt ist notwendig, um bei der Evaluierung festzustellen, ob die richtige Antwort ausgewählt wurde und um die korrekte Reihenfolge der vermischten Antworten wiedergeben zu können.

```
if ($progress) {
    $currentQuestionIndex = $progress['current_question_index'];
    $shuffledOrder = json_decode($progress['answer_order'], associative: true);

    if (isset($_POST['next_question']) && $_POST['next_question'] == "1") {
        $currentQuestionIndex += 1;

        if ($currentQuestionIndex < count($questions)) {
            $shuffledOrder = array_keys($questions[array_keys($questions)[$currentQuestionIndex]]['answers']);
            shuffle($shuffledOrder);

            $updateStmt = $pdo->prepare( query: "UPDATE user_progress SET current_question_index = :question_index");
            $updateStmt->execute([
                'question_index' => $currentQuestionIndex,
                'answer_order' => json_encode($shuffledOrder),
                'id' => $progress['id']
            ]);
        }
    }
} else {
```

Abbildung 50: Ausschnitt aus dem Skript des Prüfungsmodus, Quelle: Eigene Darstellung.

Sollte die Variable `$progress` jedoch NULL oder FALSE zurückgeben, wird im else-Zweig ein neuer Eintrag für den Benutzerfortschritt in der Datenbanktabelle angelegt. Im Programmablauf fehlten an dieser Stelle nur noch die Ermittlung der erreichten Punkte und die Gesamtanzahl, damit diese dann in den Skripts der Auswertung zur Verfügung stehen.

6.6.4.2 Beschreibung des HTML-Anteils im Modul Prüfungsmodus

Da sich der HTML-Teil im Skript des Prüfungsmodus etwas komplexer gestaltet, wird in diesem Abschnitt genauer darauf eingegangen. Die Auswahl der Stationen wurde, wie bereits im Kapitel Lernmodus erläutert (siehe Abschnitt 6.6.3), implementiert.

Im zweiten Teil sind jedoch einige speziellerer Mechanismen erforderlich, um die Interaktion mit den Antwortbuttons zu realisieren. Neben der Anzeige der aktuellen Frage werden mit einer foreach-Schleife die möglichen Antworten in der bereits vermischten Reihenfolge als Schaltfläche generiert und dargestellt. Zusätzlich gibt es ein paar versteckte Felder ("hidden"), mit denen im Hintergrund Daten in der Anwendung übertragen werden. Dazu zählen die derzeitige Station im Quiz, der Index der aktuellen Frage, die bisher erreichten Punkte und die maximal mögliche Punktezahl (siehe Abbildung 51).

```
<form method="POST" action="">
  <?php foreach ($displayAnswers as $index => $answer): ?>
    <button type="submit" name="answer" value="<?= $index ?>" class="answer-button <?= $selectedAnswerIndex
      <?= $answer['text'] ?>
    </button>
  <?php endforeach; ?>

  <input type="hidden" name="station_num" value="<?= htmlspecialchars($station_num) ?>">
  <input type="hidden" name="current_question_index" value="<?= $selectedAnswerIndex !== null ? $currentQuestionIndex : 0 ?>">
  <input type="hidden" name="total_points" value="<?= $total_points ?>">
  <input type="hidden" name="max_points" value="<?= $max_points ?>">
```

Abbildung 51: Darstellung des HTML-Teils des Prüfungsmodus, Quelle: Eigene Darstellung.

Wird eine Antwort ausgewählt, so muss überprüft werden, ob diese richtig oder falsch ist. Dafür wurde ein PHP-Code im HTML-Bereich integriert, der in Abhängigkeit der gewählten Lösung die Richtigkeit des ausgewählten Buttons kontrolliert. Nach der Auswahl eines Antwortbuttons wird mit der Funktion `$isCorrect` abgefragt, ob die Auswahl korrekt ist (siehe Abbildung 52).

```
<?php if ($selectedAnswerIndex !== null): ?>
  <p><?= $isCorrect ? "Richtig!" : "Falsch! Die richtige Antwort ist grün markiert." ?></p>
<?php endif; ?>

<?php if ($selectedAnswerIndex !== null && $currentQuestionIndex + 1 >= count($questions)): ?>
  <a href="http://lern-management-system.local/ergebnisse-dieser-station/?total_points=<?= $total_points ?>">
<?php elseif ($selectedAnswerIndex !== null): ?>
  <button type="submit" name="next_question" value="1" class="btn btn-secondary btn-next">Weiter</button>
<?php endif; ?>
```

Abbildung 52: Ausschnitt aus dem HTML-Teil des Skript des Prüfungsmodus, Quelle: Eigene Darstellung.

Ist dies der Fall, wird der gewählte Button grün visualisiert, es kommt zu einer Textausgabe und der Weiterbutton erscheint. Andernfalls wird die ausgewählte Antwort rot und die richtige Lösung in Grün gezeigt. Am Ende des HTML-Teils taucht anstelle des Weiterbuttons eine Schaltfläche zur Weiterleitung an die Seite der Ergebnisauswertung auf.

6.6.5 Skript: Auswertung

Das letzte Skript unterstützt den*die Anwender*in bei der Auswertung der Einzelergebnisse beziehungsweise aller Ergebnisse der eigenen Feuerwehr. Hier wird zwischen den Berechtigungen der Jugendlichen und derer der Funktionäre*innen unterschieden. Als präventive Maßnahme gegen diskriminierendes Verhalten können die Jung-Florianis nur ihre eigenen Ergebnisse am Ende der absolvierten Station sehen. Den Betreuern*innen und den Funktionären*innen steht zur Wahl, ob sie den Lernfortschritt eines einzelnen Mitglieds ihrer Feuerwehrjugend begutachten wollen, oder ob sie sich den Mittelwert von allen Jugendlichen ihrer Feuerwehr ansehen wollen. Dadurch können Defizite und Missverständnisse schneller erfasst werden und entsprechende Gegenmaßnahmen bei den Jugendübungen durchgeführt werden.

6.6.5.1 Funktionsbeschreibung der Auswertung für Jugendliche

Nachdem die Datenbankverbindung erfolgreich hergestellt wurde, werden die Ergebnisse aus den Spalten result1 bis result6 der mappingtable_user des jeweiligen Benutzers, mit den bereits bekannten Anweisungen ausgelesen. Da an dieser Stelle der Nachname nur in verschlüsselter Form verwendbar ist, muss dieser noch decodiert werden. Dies geschieht mit der Funktion `decryptVigenere`, deren genauerer Ablauf in Abschnitt 6.6.5.3 beschrieben ist (siehe Abbildung 53).

```
function decryptVigenere($ciphertext, $key) {
    $key = strtoupper($key);
    $keyLength = strlen($key);
    $plaintext = '';

    for ($i = 0, $j = 0; $i < strlen($ciphertext); $i++) {
        $c = strtoupper($ciphertext[$i]);
        if (ctype_alpha($c)) {
            $shift = ord($key[$j % $keyLength]) - ord(character: 'A');
            $decryptedChar = chr( codepoint: ((ord($c) - $shift + 26 - ord(character: 'A')) % 26) + ord(character: 'A'));
            $plaintext .= $decryptedChar;
            $j++;
        } else {
            $plaintext .= $c;
        }
    }
    return $plaintext;
}
```

Abbildung 53: Darstellung der Funktion für die Vigenère-Verschlüsselung aus dem Auswertungs-Skript,
Quelle: Eigene Darstellung.

Bevor es zur Ausgabe der Ergebnisse kommt, müssen noch die gespeicherten Stationen der absolvierten Wissensteststufe geladen werden. Im HTML-Teil des Skripts wird die Anordnung der Ausgabeelemente definiert und die bereits vorbereiteten Parameter werden ausgegeben (siehe Kapitel 6.7.3.3).

6.6.5.2 Funktionsbeschreibung der Auswertung für Beauftragte und Funktionäre*innen

Grundlegend funktioniert die Auswertung für die Beauftragten und Funktionäre*innen gleich wie bei den Jugendlichen. Es wurde auch hier versucht, ein homogenes Gesamtbild bei gleichen Funktionen zu etablieren. Nach der erfolgreichen Anmeldung als Betreuer*in oder Ortsjugendbeauftragte*r kann zwischen einer Einzel- oder Gruppenauswertung entschieden werden.

Bei der Einzelauswertung werden alle Einträge der Jugendlichen, welche zur selben Feuerwehr wie der*die Betreuer*in oder der*die Ortsjugendbeauftragte*r gehören, aus der Datenbanktabelle `mappingtable_user`

in einem Dropdown-Menü zur Auswahl bereitgestellt. Wie bereits in Abbildung 53 gezeigt, werden auch hier die codierten Nachnamen mit der Vigenère-Entschlüsselung (siehe Abschnitt 6.6.5.3) auf dieselbe Art und Weise decodiert. Wird eine Person im Dropdown-Menü ausgewählt, erscheint dieselbe Punkteübersicht wie bei den Jugendlichen (siehe Kapitel 6.7.4.2).

Im Vergleich zur Einzelauswertung werden bei der Gruppenauswertung ähnliche Mechanismen verwendet, jedoch wird hier bei jeder Station der Mittelwert der erreichten Punkte ausgegeben. Damit die Punkte der unterschiedlichen Wissensteststufen bei der Gruppenauswertung nicht miteinander vermischt werden, wird die Ausgabe für jede hinterlegte Stufe separat gebildet (siehe Abbildung 71). Gibt es für eine der drei Wissensteststufen keine Daten, wird diese auch nicht angezeigt. Zuerst müssen dafür alle Einträge aus der Datenbanktabelle `mappingtable_user` ausgelesen und entsprechend der Spalte `jLevel` sortiert werden. Anschließend wird von allen Stationen und in allen Stufen der Mittelwert aller Ergebnisse gebildet und zwischengespeichert (siehe Abbildung 54).

```
// jLevel of users with firebrigade = 'Tillmitsch' and valid jLevel (1, 2 or 3)
$stmt = $pdo->prepare( query: "
    SELECT jlevel,
           AVG(result1) AS avg_result1,
           AVG(result2) AS avg_result2,
           AVG(result3) AS avg_result3,
           AVG(result4) AS avg_result4,
           AVG(result5) AS avg_result5,
           AVG(result6) AS avg_result6
    FROM mappingtable_user
    WHERE firebrigade = :firebrigade AND jlevel IS NOT NULL AND jlevel IN (1, 2, 3)
    GROUP BY jlevel
```

Abbildung 54: Darstellung des Skript-Teils zur Trennung der Wissensteststufen, Quelle: Eigene Darstellung.

Damit die Stationsnamen in den unterschiedlichen Wissensteststufen und die dazugehörigen maximale Punktezahln berechnet werden kann, wird jeweils eine foreach-Schleife benötigt. In der ersten foreach-Schleife werden alle Stationsnamen in Abhängigkeit der vorhandenen `jlevel` in der Datenbanktabelle `mappingtable_wt` ermittelt und im Array `$stationsData` abgespeichert (siehe Abbildung 55).

```
foreach ($averageResults as $avgResult) {
    $jLevel = $avgResult['jlevel'];

    // Get station name and max. points for current jLevel
    $stationStmt = $pdo->prepare( query: "
        SELECT station, SUM(points) AS max_points
        FROM mappingtable_wt
        WHERE j_level = :jlevel
        GROUP BY station
        ORDER BY MIN(station_number)
        LIMIT 6
    ");
    $stationStmt->execute(['jlevel' => $jLevel]);
    $stationsData = $stationStmt->fetchAll( mode: PDO::FETCH_ASSOC);
```

Abbildung 55: Darstellung des Skript-Teils zur Ermittlung der Stationen und maximalen Punktezahln, Quelle: Eigene Darstellung.

Anhand der ermittelten Stationen lässt sich nun auf die maximal erreichbaren Punkte schließen. Diese Rückführung wird mit der zweiten foreach-Schleife realisiert. Zusätzlich werden in diesem Block noch die durchschnittlich erreichten Punkte mit in das Ergebnis übernommen und für die Ausgabe vorbereitet (siehe Abbildung 56).

```
foreach ($stationsData as $index => $station) {
    $stationName = $station['station'];
    $maxPoints = $station['max_points'];
    $avgPoints = isset($avgResult["avg_result" . ($index + 1)]) ? round($avgResult["avg_result" . ($index + 1)],

    $result['stations'][] = [
        'name' => $stationName,
        'avg_points' => $avgPoints,
        'max_points' => $maxPoints
    ];
}
```

Abbildung 56: Darstellung des Skript-Teils zur Ausgabevorbereitung, Quelle: Eigene Darstellung.

Im HTML-Teil werden für beide Auswertungen dieselben Codesegmente mit kleinen Abweichungen entsprechend der Einzel- oder Gruppenauswertung verwendet.

6.6.5.3 Vigenère-Entschlüsselung

Da, wie bereits in Kapitel 6.6.2.2 beschrieben, aufgrund der DSGVO die Nachnamen der Benutzer*innen verschlüsselt gespeichert wurden, müssen diese bei der Ausgabe wiederum entschlüsselt werden. Dies geschieht bei der Vigenère-Verschlüsselung mit demselben Schlüssel, welcher schon bei der Codierung verwendet wurde. Bei der Decodierung werden der gespeicherte Nachname und der Key wieder in seine einzelnen Buchstaben zerlegt. Für beide folgt die Umwandlung in den jeweiligen Zahlenwert der Position des Buchstabens im Alphabet. Anders als bei der Verschlüsselung wird vom Wert des Buchstabens des codierten Nachnamens der Wert des Buchstabens des Schlüssels subtrahiert. Da es dabei zu negativen Zahlen kommen kann, wird anschließend immer die Zahl 26 addiert. Erst danach wird der ganzzahlige Rest von dieser Zahl dividiert durch 26 ermittelt. Das Ergebnis sollte demselben Zahlenwert wie vor der Codierung entsprechen.

Die Tabelle 4 soll bei der Veranschaulichung der Vigenère-Entschlüsselung helfen.

Name	Key	Alphabet Name	Alphabet Key	Verschiebung	Codierung
E	S	4	18	12	M
N	T	13	19	20	U
M	U	12	20	18	S
TW	D	22	3	19	T
...	...	...	...	...	...

Tabelle 4: Beispiel der Vigenère-Entschlüsselung, Quelle: Eigene Darstellung.

6.7 Aufbau der Website

In der heutigen digitalen Welt sind Websites und ein professioneller Webauftritt unverzichtbar, um erfolgreich zu sein. Egal, ob es sich um ein Unternehmen, eine Organisation oder eine Privatperson handelt. Die Online-Präsenz ist oft der erste Eindruck, den potenzielle Kunden oder Partner gewinnen. Eine Website dient als digitale Visitenkarte, die rund um die Uhr erreichbar ist und wichtige Informationen zugänglich macht. Bei der Homepage des Lernmanagementsystems wurde versucht, die Handhabung so einfach wie möglich zu gestalten und den Informationsfluss bestmöglich zu automatisieren. In den nachfolgenden Abschnitten soll der Aufbau der einzelnen Webseiten des LMSs genauer beschrieben werden, um ein besseres Verständnis für die in Kapitel 6.6 beschriebenen Skripte zu geben.

6.7.1 Einstiegsseite

Wird die Hauptdomain des Lernmanagementsystems aufgerufen, erscheint die in Abbildung 57 gezeigte Startseite. Aktuell befindet sich die Domain noch am lokalen Host und besitzt daher keine eigentliche Webadresse wie in Kapitel 5.1 beschrieben. Im Allgemeinen soll die Hauptseite einer Website so einfach wie möglich aufgebaut sein, damit sich Benutzer*innen schnell zurechtfinden können. Den Kern bildet aber die rote Schaltfläche zur Weiterleitung auf die Anmeldeseite. Das angezeigte Bild soll in Zukunft durch ein Nachrichtenlaufband (News-Ticker) ersetzt werden. In diesem News-Ticker sollen die fünf neuesten Beiträge der Feuerwehrjugend Steiermark angezeigt werden.



Abbildung 57: Startseite des Lernmanagementsystems, Quelle: Eigene Darstellung.

Der Kopf- und Fußbereich (Header und Footer) wird auf allen Webseiten des LMSs dargestellt und besitzt überall dieselben Funktionalitäten und Eigenschaften. Beim Header gibt es die Möglichkeit, mit einem Klick auf den Text "Lern Management System" direkt auf die Startseite zurückzukehren. Mit den fünf

Schaltflächen im oberen rechten Bereich kann zu den einzelnen Webseiten des LMSs oder aber auch zur Website des LFV Steiermark navigiert werden (siehe Abbildung 58).

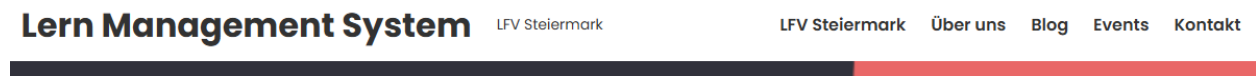


Abbildung 58: Kopfbereich des Lernmanagementsystems, Quelle: Eigene Darstellung.

Im Fußbereich gibt es neben den bereits in der Kopfzeile verwendeten Links noch die Möglichkeit, zu den rechtlich relevanten Themengebieten zu gelangen (siehe Abbildung 59). Dazu zählen das Impressum, die Cookie-Richtlinien und die Datenschutzerklärung. Das Impressum sorgt dafür, dass die für die Homepage verantwortlichen Personen von der*m Benutzer*in klar identifiziert werden können. Bei den Cookie-Richtlinien muss, wie bereits in Kapitel 5.6 erklärt, die Nutzung und genaue Verwendung der von den Cookies erfassten Daten angeführt werden. Abschließend fehlt laut DSGVO nur noch die Datenschutzerklärung in der der*die Nutzer*in über die Nutzung seiner*ihrer Daten informiert werden. Es wird ebenso erläutert, wer Zugriff auf die Daten hat und welche Rechte der*die Benutzer*in hat.

In der rechten Hälfte des Footers sind neben den schnellen Kontaktmöglichkeiten auch noch zwei Links für die häufig gestellten Fragen (FAQs) und den Release Status des Lernmanagementsystems angeführt. Ziel des Release Status ist es, einen Bereich zu schaffen, in dem Benutzer*innen auf einen Blick die neuesten Funktionalitäten des LMSs in Erfahrung bringen können.

Wir sind für Sie da	Informationen	Support	Kontakt
LFV Steiermark	Lern Management System	FAQs	✉ support@lms.com
Über uns	Impressum	Release Status	+43 664 12 34 567
Blog	Cookie-Richtlinien		Florianistraße 22
Events	Datenschutzerklärung		Lebring
Kontakt			

Abbildung 59: Fußbereich des Lernmanagementsystems, Quelle: Eigene Darstellung.

6.7.2 Benutzeranmeldung

Nachdem auf der Startseite der Button Anmeldung geklickt wurde, erfolgt die Weiterleitung zur Benutzeranmeldung (siehe Abbildung 60). Wie bereits im vorherigen Abschnitt 6.7.1 kurz geschildert, werden auf allen nachfolgenden Webseiten derselbe Kopf- und Fußbereich angezeigt. Fokus der Benutzeranmeldung liegt auf der Darstellung des Anmeldeformulars und der Interaktion mit der*dem Benutzer*in. Es müssen alle Eingabefelder entsprechend befüllt werden, damit es zu einer erfolgreichen Anmeldung kommen kann. Den Rest erledigt das in Kapitel 6.6.1 beschriebene Plugin.

Abbildung 60 zeigt ein Login-Formular mit dem Titel "Login". Es enthält folgende Felder:

- Vorname:** Ein Textfeld mit dem Inhalt "Max".
- Nachname:** Ein Textfeld mit dem Inhalt "MUSTERMANN".
- Feuerwehr:** Ein Textfeld mit dem Inhalt "Tillmitsch".
- Standesbuchnummer:** Ein Textfeld mit dem Inhalt "999".
- Funktion:** Ein Dropdown-Menü mit der Auswahl "JFF/JFM".

Unter den Feldern befindet sich ein roter Button mit der Aufschrift "Anmelden".

Abbildung 60: Benutzeranmeldung des Lernmanagementsystems, Quelle: Eigene Darstellung.

Lediglich die Auswahl der Funktion wird mithilfe eines Dropdown-Menüs, in dem die für die Arbeit mit den Jugendlichen vordefinierten Rollen angezeigt werden, unterstützt (siehe Abbildung 61).

Abbildung 61 zeigt das Dropdown-Menü für die Funktion. Die Liste der Funktionen lautet:

- Bitte wählen Sie eine Funktion
- JFF/JFM
- Feuerwehrjugend-Betreuer*in
- Orts-Jugendbeauftragte*r
- Abs-Jugendbeauftragte*r
- BFV-Jugendbeauftragte*r
- LFV-Jugendbeauftragte*r

Unter der Liste befindet sich ein roter Button mit der Aufschrift "Anmelden".

Abbildung 61: Darstellung des Dropdown-Menüs der Benutzeranmeldung, Quelle: Eigene Darstellung.

6.7.3 Dashboard Jugend

Wird bei der Benutzeranmeldung im Dropdown-Menü die Funktion Jungfeuerwehrfrau / Jungfeuerwehrmann (JFF / JFM) ausgewählt, so wird der*die Benutzer*in zum Dashboard der Jugend weitergeleitet. Natürlich nur, wenn der*die Nutzer*in erfolgreich authentifiziert werden konnte.

Zurzeit kann auf der Übersichtsseite der Jugendlichen zwischen dem Lernmodus und dem Prüfungsmodus gewählt werden (siehe Abbildung 62). Je nachdem welcher Modus gewünscht wird, kann die Weiterleitung per Mausklick initialisiert werden. Die Buttons selbst besitzen nur den Link zur entsprechenden Webseite.



Abbildung 62: Darstellung der Webseite des Dashboards für die Jugendlichen, Quelle: Eigene Darstellung.

6.7.3.1 Lernmodus

Nachdem, wie in Abschnitt 6.7.3 erläutert, der Lernmodus ausgewählt wurde, wird die Webseite aktualisiert und es kommt zur Auswahl der gewünschten Wissensteststufe (siehe Abbildung 63). Zukünftig soll dieser Teil im Lern- und Prüfungsmodus übersprungen werden, da anhand der Daten aus dem FDISK ermittelt werden kann, welche Stufe der*die authentifizierte Benutzer*in absolvieren darf.

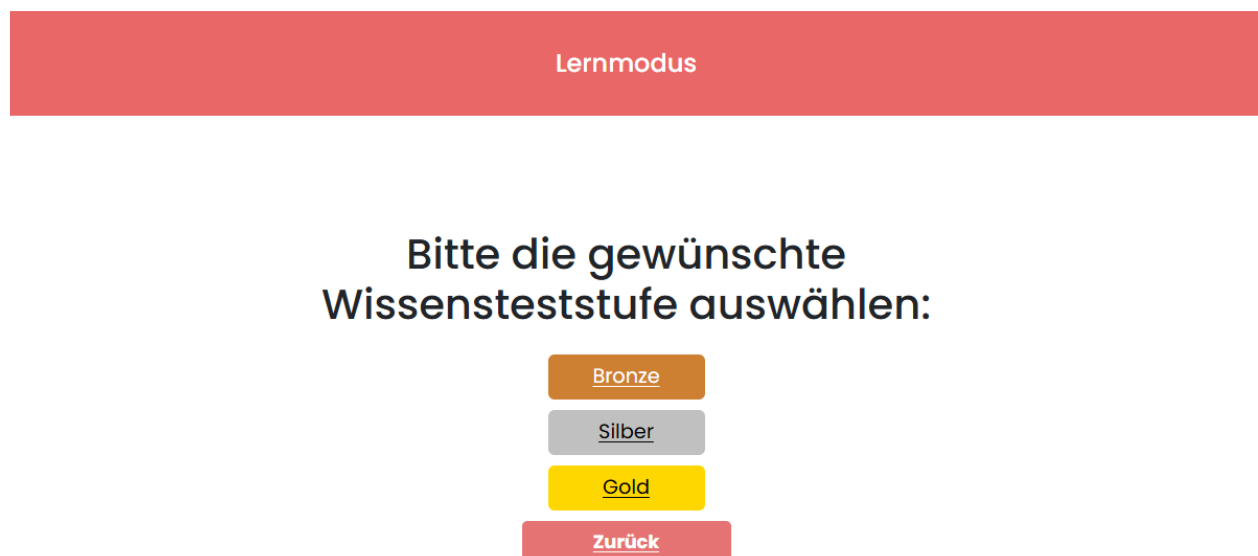


Abbildung 63: Auswahl der Wissensteststufe im Lernmodus, Quelle: Eigene Darstellung.

Nach der Wahl der Wissensteststufe werden die passenden Stationen durch das in Kapitel 6.6.3 beschriebene Skript aus der Datenbanktabelle ausgelesen und als Schaltfläche dargestellt (siehe Abbildung 64). Zusätzlich gibt es zwei Navigationsbuttons, mit denen die Auswahl der gewählten Stufe und ein Wechsel des Modus durchgeführt werden kann. Wie bereits zuvor beschrieben wird auch der Button zur Wahl der Wissensteststufe später entfallen.

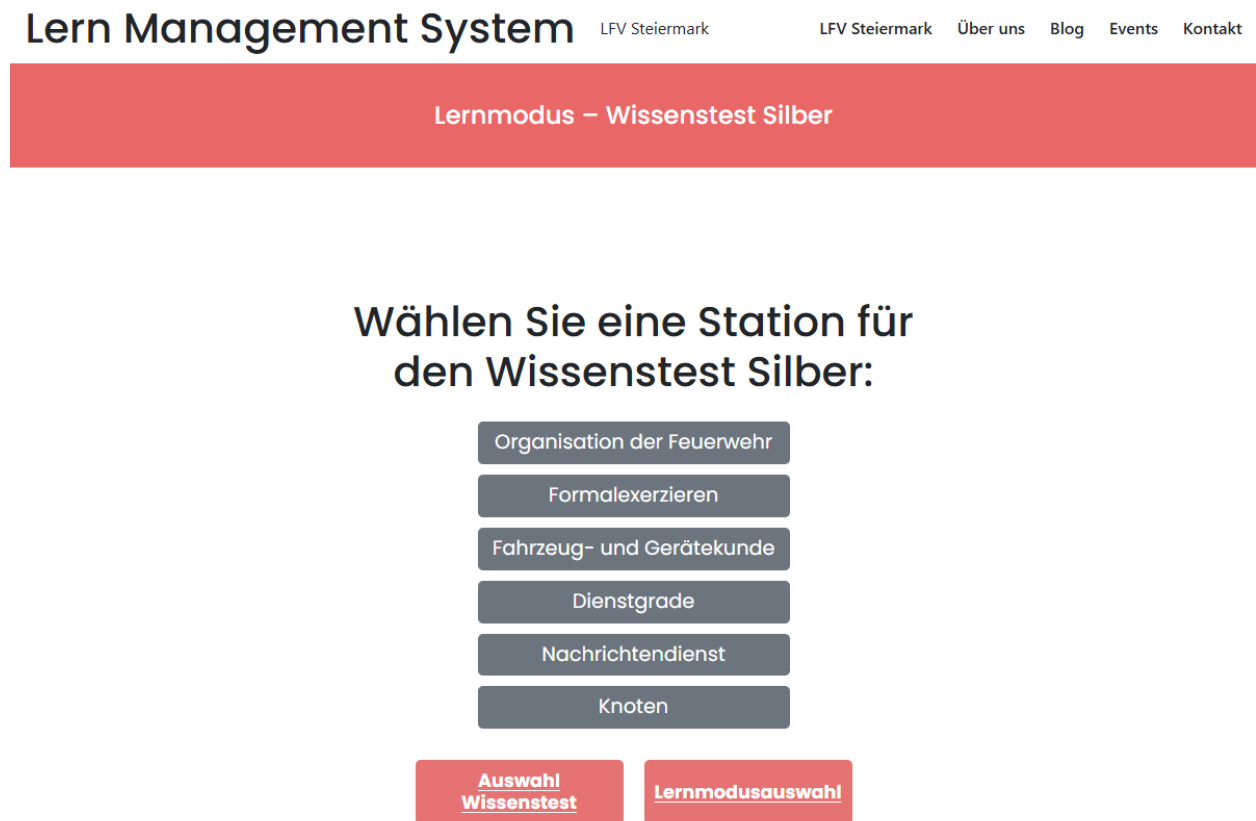


Abbildung 64: Stationsübersicht der Wissensteststufe Silber im Lernmodus, Quelle: Eigene Darstellung.

Wurde eine Station ausgewählt, wird die erste Frage aus der Datenbank ausgelesen und angezeigt. Die korrekte Antwort dagegen wird erst nach einem Klick auf den Antwort-Button ausgegeben. Dies soll das aufmerksame Lesen und Verstehen der Fragestellung fördern. Der Zurückbutton ist bei der ersten Frage noch deaktiviert und mit dem Weiterbutton kann zur nächsten Fragestellung weitergesprungen werden (siehe Abbildung 65).



Abbildung 65: Darstellung der ersten Frage aus dem Wissenstest Bronze im Lernmodus, Quelle: Eigene Darstellung.

Während der Abarbeitung der Fragen im Lernmodus kann immer auf die Stationsübersicht über die dafür integrierte Schaltfläche zurückgekehrt werden. Der Fortschritt wird in der Datenbanktabelle zwischengespeichert und es kann direkt zur letzten angezeigten Frage navigiert werden.

6.7.3.2 Prüfungsmodus

Der Aufbau des Prüfmodus fußt auf der Struktur des Lernmodus. Dadurch ist die Darstellung für die Auswahl der Wissensteststufe und der gewünschten Station ident. Was sich allerdings bei der Ansicht der Fragestellungen ändert, ist zum einen die Überschrift und zum anderen die Wahlmöglichkeit, aus vier potenziellen Antworten zu wählen. Durch diese Änderung und der gewünschten Funktion des Prüfungsmodus kommt es auch zum Entfall der Navigationsbuttons aus dem Lernmodus (siehe Abbildung 66).

Prüfungsmodus Bronze

Frage 3:

Was bedeutet das der taktischen Bezeichnung nachgesetzte A?

Das Fahrzeug verfügt über einen Allradantrieb.

Das Fahrzeug verfügt über eine Anfahrhilfe für steiles Gelände.

Das Fahrzeug verfügt über einen Autopiloten.

Das Fahrzeug verfügt über einen Akku und wird elektrisch betrieben.

[Zurück zur Stationsübersicht](#)

Abbildung 66: Darstellung einer Fragestellung im Prüfungsmodus des Wissenstest Bronze, Quelle: Eigene Darstellung.

Durch einen Klick auf eine der vier Antwortmöglichkeiten ergeben sich zwei Szenarien, wie sich die Visualisierung der derzeitigen Ansicht verändern kann. Wurde die richtige Antwort ausgewählt, so wird diese grün dargestellt und es erscheinen eine Textausgabe mit dem Inhalt "Richtig!" und ein Weiterbutton unter den Schaltflächen (siehe Abbildung 67).

Frage 3:

Was bedeutet das der taktischen Bezeichnung nachgesetzte A?

Das Fahrzeug verfügt über einen Allradantrieb.

Das Fahrzeug verfügt über eine Anfahrhilfe für steiles Gelände.

Das Fahrzeug verfügt über einen Autopiloten.

Das Fahrzeug verfügt über einen Akku und wird elektrisch betrieben.

Richtig!

Weiter

[Zurück zur Stationsübersicht](#)

Abbildung 67: Darstellung bei der Auswahl einer richtigen Antwort im Prüfungsmodus, Quelle: Eigene Darstellung.

Ist die gewählte Lösung inkorrekt, wird diese rot eingefärbt. Die korrekte Antwort wird grün koloriert und es erscheinen ein Textfeld mit der Information "Falsch! Die richtige Antwort ist grün markiert" sowie der Weiterbutton (siehe Abbildung 68).

Frage 5:

Wer erteilt den Auftrag zum Auf- und Absitzen auf das und vom Feuerwehrfahrzeug?

Der*Die Maschinist*in

Der*Die Feuerwehrkassierer*in beziehungsweise der*die Schriftführer*in

Der*Die Gruppenkommandant*in beziehungsweise der*die Jugendbeauftragte*r.

Der*Die Feuerwehrkommandant*in beziehungsweise dessen Stellvertreter*in

Falsch! Die richtige Antwort ist grün markiert.

Weiter

Zurück zur Stationsübersicht

Abbildung 68: Darstellung bei der Auswahl einer falschen Antwort im Prüfungsmodus, Quelle: Eigene Darstellung.

6.7.3.3 Ergebnisse der Station

Wird eine Station im Prüfungsmodus beendet, so erscheint automatisch die Webseite mit den Ergebnissen der Stationen. Es werden der Vorname, der entschlüsselte Nachname (siehe Kapitel 6.6.5.3) und die bisher erreichten Punkte aller Stationen sowie deren Summe ausgegeben. Neben einem Button, der zur Stationsübersicht zurückführt, gibt es noch eine weitere Schaltfläche, mit der die erzielte Punkteanzahl an die Datenbanktabelle übertragen werden kann. Wird dieser Button nicht gedrückt, werden auch keine Punkte bei der Auswertung der Funktionäre angezeigt, da die Punkteanzahl nicht in der Datenbank gesichert wird (siehe Abbildung 69).

Ergebnisse

Vorname:	Milki
Nachname:	MUSTERMANN
Organisation der Feuerwehr:	7
Formalexerzieren:	4
Fahrzeug- und Gerätekunde:	1
Dienstgrade (Feuerwehrjugend und Branddienst):	0
Warn- und Alarmsysteme:	1
Einsatz und Dienstbekleidung:	1
Gesamtpunkte: 14	
Zurück zur Stationsübersicht Gesamtpunkte speichern	

Abbildung 69: Darstellung der Ergebnisausgabe der Jugendlichen, Quelle: Eigene Darstellung.

6.7.4 Dashboard Funktionäre

Meldet sich ein*e Feuerwehrjugend-Betreuer*in oder ein*e Ortsjugendbeauftragte*r erfolgreich an, werden diese zum Dashboard der Funktionäre weitergeleitet. Im Gegensatz zum Dashboard der Jugendlichen können die Funktionäre das Lernmanagementsystem derzeit ausschließlich zur Evaluierung des Lernfortschrittes ihrer Feuerwehrjugend nutzen. Deswegen gibt es wie in Abbildung 70 dargestellt nur zwei Buttons, mit denen entweder eine Auswertung aller Ergebnisse oder die Resultate eines Einzelnen abgerufen werden können.



Abbildung 70: Darstellung des Dashboards der Funktionäre, Quelle: Eigene Darstellung.

6.7.4.1 Auswertung Übersicht

Zweck dieser Auswertung ist es, die Jugendgruppe als Ganzes bewerten zu können. Dadurch können die Funktionäre die Stationen schneller identifizieren, bei denen tendenziell weniger Punkte erreicht wurden und diese verstärkt mit den Jugendlichen beüben. Bei der Visualisierung wird der Durchschnitt der Punkte aller Jung-Florianis in der jeweiligen Wissensteststufe ausgegeben. In Abbildung 71 wird die Auswertung von zwei in der Datenbank generierten Datensätzen abgefragt und angezeigt. Die beiden Einträge wurden für die Stufe Bronze angelegt. Würde es zusätzliche Daten für die Stufen Silber und Gold geben, so würden diese unter der Bronze-Auswertung dargestellt werden. Der Name der Feuerwehr wird aus der Datenbanktabelle erfasst und in der Überschrift automatisch eingesetzt.

Auswertung Übersicht		
Durchschnittliche Ergebnisse für Feuerwehr Tillmitsch		
Bronze		
Station	Durchschnittliche Punkte	Maximale Punkte
Organisation der Feuerwehr	5.5 Pkt.	8 Pkt.
Formalexerzieren	1.5 Pkt.	4 Pkt.
Fahrzeug- und Gerätekunde	6.5 Pkt.	8 Pkt.
Dienstgrade (Feuerwehrjugend und Branddienst)	72 Pkt.	89 Pkt.
Warn- und Alarmsysteme	1 Pkt.	1 Pkt.
Einsatz und Dienstbekleidung	1 Pkt.	1 Pkt.

Abbildung 71: Darstellung der durchschnittlichen Ergebnisse, Quelle: Eigene Darstellung.

6.7.4.2 Auswertung Einzel

Die Einzelauswertung soll den Funktionären einen besseren Einblick in den Lernfortschritt jedes einzelnen Jugendlichen geben. Anfänglich werden auf dieser Webseite nur ein Dropdown-Menü und eine Schaltfläche dargestellt (siehe Abbildung 72). Die Daten des Auswahlmenüs werden aus der Datenbanktabelle in Abhängigkeit der Feuerwehruzugehörigkeit entnommen.

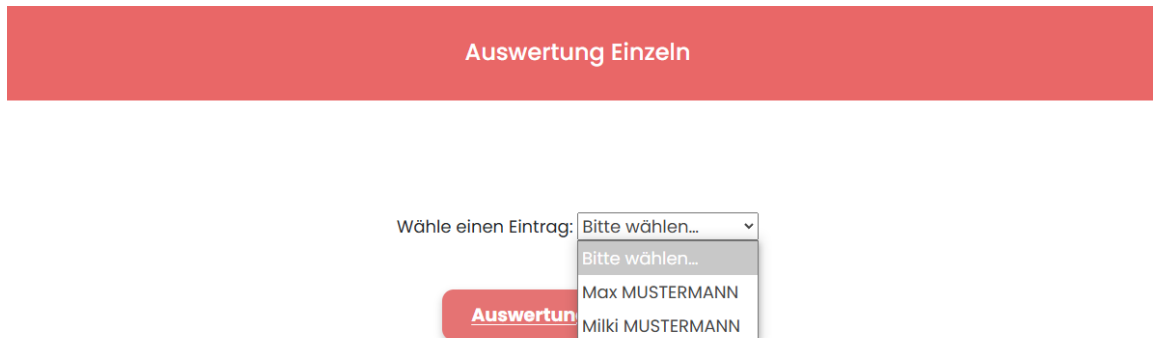


Abbildung 72: Darstellung der Einzelauswertung ohne Auswahl einer*s Jugendlichen, Quelle: Eigene Darstellung.

Wird im Dropdown-Menü ein*e Jugendliche*r ausgewählt, so wird die bereits bekannte Auswertung der Stationsergebnisse angezeigt (siehe Abbildung 73). Kommt es zu einem Wechsel des Namens im Auswahlmenü, werden die Daten automatisch an die Auswahl angepasst. Die in Abbildung 72 teilweise ersichtliche Schaltfläche leitet den*die Benutzer*in wieder zurück zum Dashboard der Funktionäre.

Wähle einen Eintrag: Max MUSTERMANN ▾

Ergebnisse

Vorname:	Max
Nachname:	MUSTERMANN
Organisation der Feuerwehr:	4 Pkt. von 8 Pkt.
Formalexerzieren:	0 Pkt. von 4 Pkt.
Fahrzeug- und Gerätekunde:	8 Pkt. von 8 Pkt.
Dienstgrade (Feuerwehrjugend und Branddienst):	66 Pkt. von 89 Pkt.
Warn- und Alarmsysteme:	1 Pkt. von 1 Pkt.
Einsatz und Dienstbekleidung:	1 Pkt. von 1 Pkt.
Gesamtpunkte: 77	

Abbildung 73: Darstellung der Einzelauswertung mit Auswahl einer*s Jugendlichen, Quelle: Eigene Darstellung.

7 TESTS

Nach Abschluss des Praxisteils dieser Masterarbeit fanden zwei Testserien, bei denen das Lernmanagementsystem auf Fehler und Funktionalität geprüft wurde, statt. Ziel der eigenen Testserie war es, Fehlverhalten im Ablauf zu identifizieren und zu beheben. Dabei konnten einige fehlerhafte Links, welche bei der Weiterleitung auf andere Webseiten verwendet wurden, erkannt und berichtigt werden. Dies war vor allem bei den Seiten der Fall, die mit den unterschiedlichen Wissensteststufen arbeiten, da beim Kopieren der Skripte auf die Adaption der Links vergessen wurde. Dasselbe Problem fiel auch bei der Textausgabe im Lernmodus Skript auf. Bei den Überschriften der drei verschiedenen Stufen wurde nur der Wissenstest in Bronze ausgegeben. Neben solchen Kleinigkeiten gab es aber auch noch ein größeres Problem mit dem Prüfungsmodus. Nachdem die letzte Frage beantwortet wurde, wird der `current_question_index` nicht auf null zurückgesetzt. Dies hatte zur Folge, dass beim erneuten Aufruf der Station ausschließlich mit der letzten Fragestellung gestartet werden konnte. Das Fehlverhalten wurde durch eine Positionsänderung der Rücksetzfunktion behoben. Das Problem war, dass es an dieser Stelle zu einer ungünstigen Überlagerung von Befehlen kam und das Rücksetzen dadurch nicht ausgeführt wurde.

Am Samstag, den 9. November 2024, fand eine Testung durch drei Mitglieder der Feuerwehrjugend und dem Ortsjugendbeauftragten der FF Tillmitsch statt. Bei der Testserie mit der Feuerwehrjugend der Freiwilligen Feuerwehr (FF) Tillmitsch lag der Fokus auf der Handhabung der einzelnen Funktionen des LMSs. Mithilfe der TeamViewer Remote Software, einem speziellen Programm für sichere Remote-Zugriffe, wurde eine Verbindung zwischen dem lokalen Server, auf dem sich das LMS befindet und dem Computer der FF Tillmitsch hergestellt. Dadurch konnten die Tests direkt im Rüsthaus durchgeführt werden. Zuerst bekamen die Jung-Florianis eine kurze Einführung in die Aufgaben und Funktionalitäten des Lernmanagementsystem. Im Anschluss durfte dann jedes Mitglied die Website auf Herz und Nieren testen. Die Ergebnisse dieser Testserie werden in Kapitel 8.1 erläutert.

8 RÜCKMELDUNGEN

Die Rückmeldungen beziehen sich ausschließlich auf das Feedback der Testserie aus Kapitel 7. Ein Meeting mit den Verantwortlichen des Landesfeuerwehrverbandes Steiermark wird aus zeitlichen Gründen voraussichtlich erst im Dezember 2024 oder im Jänner 2025 stattfinden.

8.1 Feedback der Feuerwehrjugend der FF Tillmitsch

Wie bereits in Kapitel 7 beschrieben, gab es eine eigene Testserie mit drei Mitgliedern der Feuerwehrjugend und dem Ortsjugendbeauftragten der FF Tillmitsch. Das LMS erhielt ein überwiegend positives Feedback, da es solch eine Plattform bis jetzt noch nicht gegeben hat und alle sehr begeistert vom Lern- und Prüfungsmodus waren. Besonders gut kam die Verwendung von Videomaterial bei der Knotenkunde und dem Formalexerzieren an. Auch die Aktualisierung des Bildmaterials bei der Gerätekunde und den Dienstgraden sorgte für Wohlwollen.

Zukünftig sollten der Lern- und Prüfungsmodus jedoch durch ein paar zusätzliche Schaltflächen erweitert werden. So soll es generell auf jeder Seite die Möglichkeit geben, sich vom Lernmanagementsystem abzumelden. Weiters wäre es beim Prüfungsmodus sinnvoll, einen Button bei der Stationsauswahl einzufügen, um damit direkt auf die Ergebnisseite zu gelangen. Dadurch könnten die Jugendlichen ihre letzten Punktestände prüfen und müssten sich nicht erst durch die Fragestellungen einer Station durchklicken. Im Lernmodus wird der Fortschritt bei den Fragen in der Datenbank zwischengespeichert. Dieser Zwischenstand sollte auch per Knopfdruck zurückgesetzt werden können, damit nicht immer bei der zuletzt angesehenen Frage der ausgewählten Station fortgesetzt werden muss.

Als kleine Gedächtnisstütze würden sich die Jung-Florianis einen integrierten Kalender wünschen, der ihnen die wichtigsten Termine auf Bereichs- und Landesebene zeigt. Dort sollen Daten zu geplanten Jugendaktivitäten wie dem Wissenstest im eigenen Bereich, aber auch das beliebte Landesjugendzeltlager und ähnliches abgelegt sein.

Eine weitere Forderung war die Integration der restlichen Jugendbewerbe in die Website. So sollen das Feuerwehrjugendleistungsabzeichen (FJLA), die Wissenstestspiele und das Funkleistungsabzeichen (FULA) der Jugend zukünftig auch integriert werden. Vor allem beim FJLA wäre eine detaillierte Beschreibung des Ablauf und Videos für die zu besetzenden Positionen von großem Vorteil für die Jugendlichen.

Dem Ortsjugendbeauftragten fiel auf, dass er die Ergebnisse seiner Jugendmannschaft nicht zurücksetzen konnte. Diese Funktion wäre von großem Vorteil, da dadurch bei Übungen die aktuellen Datenstände auf null gestellt und eine Zwischenkontrolle der Lernziele effizienter durchgeführt werden könnte.

Beim Feuerwehrjugendleistungsabzeichen kam die Bitte, die Ausbildungsunterlagen, in denen Details wie der Aufbau der Bewerbsbahn beschrieben sind, für die Betreuer*innen und Ortsjugendbeauftragten zu digitalisieren.

8.2 Einarbeitung des Feedbacks

Wie bereits zu Beginn dieses Kapitels erläutert, findet die Präsentation des LMSs vor dem LFV Steiermark frühestens im Dezember 2024 statt. Bis dahin sollen die Punkte betreffend der fehlenden Schaltflächen eingearbeitet werden. Alle weiteren Anregungen werden für dieses Meeting aufbereitet, um diese bei der Vorstellung vor dem Gremium auch ansprechen zu können. Dabei sollte das erhaltene Feedback aus den Reihen der Feuerwehrjugend die größte Aufmerksamkeit erhalten.

Die Rückmeldungen des Landesfeuerwehrverbandes Steiermark werden in Abhängigkeit einer Beauftragung zur Vervollständigung des Lernmanagementsystems nachhaltig in die Weiterentwicklung einfließen. Sollte es entgegen den Erwartungen zu keiner Auftragserteilung, zur Integration und zum Ausbau des Lernmanagementsystems kommen, wird dieses dennoch als Hilfsmittel in der Freiwilligen Feuerwehr Tillmitsch etabliert werden. Die Jung-Florianis aus Tillmitsch können alle Funktionen dann direkt über die Website der FF Tillmitsch abrufen.

9 ERGEBNISSE UND AUSBLICK

Im Allgemeinen ist das Lernmanagementsystem für sich eine sehr positive Sache und ein möglicher Wegweiser in eine digitale Zukunft. Jugendliche der Feuerwehrjugend des Landes Steiermark können auf einfachem Wege für den Wissenstest in der jeweiligen Stufe lernen und ihr Wissen im Prüfungsmodus auf die Probe stellen. Durch die Verwendung von Lernvideos wird den Jung-Florianis die Möglichkeit eröffnet, auch von zu Hause aus das Formalexerzieren und die Knoten zu üben. Die für diese Masterarbeit verwendeten Lernvideos müssen jedoch erneut aufgenommen werden, da diese nur zu Testzwecken produziert und integriert wurden. Die neuen Filme sollten zudem eine Audiospur mit einer exakten und verständlichen Beschreibung des Gezeigten besitzen. Das verwendete Bildmaterial in den Stationen der Gerätekunde sollte ebenfalls ersetzt werden. Es sollte zukünftig darauf geachtet werden, dass die Geräte so herstellerneutral wie möglich abgebildet werden, da es hier durch farbliche Unterschiede zu Verwechslungen kommen kann.

Für die Betreuer*innen und die Funktionäre*innen ergibt sich durch die Auswertung der Ergebnisse eine neue Möglichkeit, ihre Jugendlichen besser zu unterstützen und auf den Wissenstest vorzubereiten. Defizite können frühzeitig erkannt und lernschwächere Jugendmitglieder intensiver gefördert werden. Die Nutzung des Lernmanagementsystems als zentrale Informationsplattform für alle Anliegen rund um die Bewerbe der Feuerwehrjugend wäre ein großer Schritt in die richtige Richtung.

Sollte das Lernmanagementsystem aus welchen Gründen auch immer nicht zum aktiven Einsatz im Landesfeuerwehrverband Steiermark kommen, so wäre es wünschenswert, dass die Grundsätze und Rückmeldungen in eine alternative Lösung miteinfließen würden.

LITERATURVERZEICHNIS

Gedruckte Werke

Eisenmenger, Richard; Brinkmann, Florian (2022): *WordPress 6 – Das umfassende Handbuch*, 2. Auflage, Rheinwerk Verlag, Bonn

Müller, Peter (2024): *Einstieg in WordPress 6*, 5. Auflage, Rheinwerk Verlag, Bonn

Simeth, Florian (2022): *WordPress – Plugins, Themes und Blöcke entwickeln*, 1. Auflage, Rheinwerk Verlag, Bonn

Wenz, Christian; Hauser, Tobias (2024): *PHP 8 und MySQL – Das umfassende Handbuch*, 5. Auflage, Rheinwerk Verlag, Bonn

Online-Quellen

Cornelsen Verlag (2024): *Methodik und Methoden*,
<https://www.cornelsen.de/empfehlungen/paedagogik/methodik> [12.01.2024]

Feuerwehr Neuhaus (2014): *Leitfaden für die Feuerwehrausbildung*,
https://feuerwehr-neuhausen.ch/fileupload/Leitfaden_Feuerwehrausbildung.pdf [Stand: 23.11.2024]

Fuchs Media Solutions (o.J.): *Host – Begriffserklärung und Definition*,
<https://www.seo-analyse.com/seo-lexikon/h/host/#geschichtliches-zum-begriff-host-vom-grossrechner-zum-server> [Stand: 29.05.2024]

Holmgren, Robert (2018): *Firefighting training at a distance – a longitudinal study*,
<https://www.tandfonline.com/doi/full/10.1080/13636820.2018.1464054#d1e584> [Stand: 20.05.2024]

IBM Deutschland GmbH (2024): *Ressource 'POST-Methode für bestimmte Objekttypen'*,
<https://www.ibm.com/docs/de/baw/23.x?topic=potr-post-method> [Stand: 09.11.2024]

intersoft consulting services AG (2021): *Art. 30 DSGVO – Verzeichnis von Verarbeitungstätigkeiten*,
<https://dsgvo-gesetz.de/art-30-dsgvo/> [Stand: 25.08.2024]

intersoft consulting services AG (2021): *Art. 32 DSGVO – Sicherheit der Verarbeitung*,
<https://dsgvo-gesetz.de/art-32-dsgvo/> [Stand: 25.08.2024]

intersoft consulting services AG (2021): *Datenschutz-Grundverordnung DSGVO*,
<https://dsgvo-gesetz.de/> [Stand: 25.08.2024]

Ionos SE (2023): *FTP: Das File Transfer Protocol erklärt*,
<https://www.ionos.at/digitalguide/server/knowhow/ftp-file-transfer-protocol/> [Stand: 30.05.2024]

Ionos SE (2024): *Was ist eine Domain?*,
<https://www.ionos.at/digitalguide/domains/domaintipps/was-ist-eine-domain/> [Stand: 29.05.2024]

JetBrains s.r.o. (o.J.): *Funktionen von PhpStorm*,
<https://www.jetbrains.com/de-de/phpstorm/features/> [Stand: 15.09.2024]

Johannes Kepler Universität Linz (2021): *Die hoheitlichen Befugnisse der Feuerwehren im Einsatz*,
<https://epub.jku.at/obvulihs/download/pdf/6148477?originalFilename=true> [Stand: 19.05.2024]

- Kinsta Inc. (2023): *Was ist ein CMS?*,
<https://kinsta.com/de/wissensdatenbank/content-management-system/> [Stand: 11.09.2024]
- Landesfeuerwehrverband Steiermark (2023): *Richtlinie - Feuerwehr mit Jugendmitgliedern*,
https://www.lfv.steiermark.at/PortalData/1/Resources/dokumente/6_innere_angelegenheiten/jugend/RL_Feuerwehren_mit_Jugendmitgliedern_2023.pdf [Stand: 19.05.2024]
- Landesfeuerwehrverband Steiermark (2016): *Fragenkatalog Wissenstest Bronze*,
[https://www.lfv.steiermark.at/PortalData/1/Resources/dokumente/5_aus_u_weiterbildung/9_fwjugend_fjla/info/allgemeine_info_\(ab_2011\)/FO_Fragenkatalog_Wissenstest_Bronze_v16.02.pdf](https://www.lfv.steiermark.at/PortalData/1/Resources/dokumente/5_aus_u_weiterbildung/9_fwjugend_fjla/info/allgemeine_info_(ab_2011)/FO_Fragenkatalog_Wissenstest_Bronze_v16.02.pdf) [Stand: 19.05.2024]
- Landesfeuerwehrverband Steiermark (2016): *Fragenkatalog Wissenstest Gold*,
[https://www.lfv.steiermark.at/PortalData/1/Resources/dokumente/5_aus_u_weiterbildung/9_fwjugend_fjla/info/allgemeine_info_\(ab_2011\)/FO_Fragenkatalog_Wissenstest_Gold_v16.02.pdf](https://www.lfv.steiermark.at/PortalData/1/Resources/dokumente/5_aus_u_weiterbildung/9_fwjugend_fjla/info/allgemeine_info_(ab_2011)/FO_Fragenkatalog_Wissenstest_Gold_v16.02.pdf) [Stand: 19.05.2024]
- Landesfeuerwehrverband Steiermark (2016): *Fragenkatalog Wissenstest Silber*,
[https://www.lfv.steiermark.at/PortalData/1/Resources/dokumente/5_aus_u_weiterbildung/9_fwjugend_fjla/info/allgemeine_info_\(ab_2011\)/FO_Fragenkatalog_Wissenstest_Silber_v16.02.pdf](https://www.lfv.steiermark.at/PortalData/1/Resources/dokumente/5_aus_u_weiterbildung/9_fwjugend_fjla/info/allgemeine_info_(ab_2011)/FO_Fragenkatalog_Wissenstest_Silber_v16.02.pdf) [Stand: 19.05.2024]
- Myra Security GmbH (o.J.): *Was ist HTTP & HTTPS?*,
<https://www.myrasecurity.com/de/knowledge-hub/httphttps/> [Stand: 30.05.2024]
- Next Step NH (2020): *The Firefighter's Handbook*,
<https://beyondclassroom.org/wp-content/uploads/2020/08/firefighters-handbook-pages.pdf>
[Stand: 20.05.2024]
- Niederösterreichischer Landesfeuerwehrverband (o.J.): *FDISK*,
<https://www.noel122.at/fachinfos/informationstechnologie/fdisk> [Stand: 13.06.2024]
- one.com Group AB (o.J.): *Was ist PHP?*, <https://www.one.com/de/hosting/was-ist-php/>
[Stand: 15.09.2024]
- Oracle Corporation (2020): *Was ist ein Datenbankverwaltungssystem (DBMS)?*,
<https://www.oracle.com/de/database/what-is-database/#dbms> [Stand: 09.06.2024]
- Oracle Corporation (2020): *Was ist eine Datenbank?*,
<https://www.oracle.com/at/database/what-is-database/> [Stand: 09.06.2024]
- Oracle Corporation (o.J.): *Was ist MySQL?*, <https://www.oracle.com/at/mysql/what-is-mysql/>
[Stand: 24.06.2024]
- Österreichischer Bundesfeuerwehrverband (2018): *Erklärvideo zum österreichischen Feuerwehrwesen*,
<https://www.bundesfeuerwehrverband.at/2018/10/03/erklaervideo-zum-oesterreichischen-feuerwehrwesen-online/> [Stand: 19.05.2024]
- Österreichischer Bundesfeuerwehrverband (2022): *Präsidium des ÖBFV*,
<https://www.bundesfeuerwehrverband.at/homepage-oebfv-2/oebfv/praesidium-des-oebfv/>
[Stand: 19.05.2024]

Österreichischer Bundesfeuerwehrverband (2023): *Feuerwehren in Österreich*,
https://www.bundesfeuerwehrverband.at/wp-content/uploads/2024/02/Statistik_2023.pdf
[Stand: 19.05.2024]

Rechtsinformationssystem des Bundes (2022): *Bundesgesetzblatt 267/2022*,
<https://www.ris.bka.gv.at/eli/bgbl/II/2022/267/20220706> [Stand: 20.05.2024]

SEO-Küche Internet Marketing GmbH & Co. KG (2023): *Was ist HTML?*,
<https://www.seo-kueche.de/lexikon/html/> [Stand: 15.09.2024]

Siever, Friedrich (2024): *Die 15 WordPress Vorteile und Nachteile für Dein Website Projekt*,
<https://www.exovia.de/journal/wordpress-vorteile-nachteile/> [Stand: 11.09.2024]

Studyflix GmbH (o.J.): *Didaktische Prinzipien*,
<https://studyflix.de/paedagogik-psychologie/didaktische-prinzipien-5256> [Stand: 08.06.2024]

Studyflix GmbH (o.J.): *Vigenère Verschlüsselung*,
<https://studyflix.de/informatik/vigenere-verschlüsselung-1607> [Stand: 09.11.2024]

Suske, Sophie (2023): *Welche Arten von Cookies gibt es – und welche Funktion haben sie?*,
<https://www.e-recht24.de/tracking-cookies/13147-cookie-arten.html> [Stand: 12.07.2024]

WordPress.org (o.J.): *Unsere Geschichte*, <https://de.wordpress.org/about/> [Stand: 21.03.2024]

World4You Internet Services GmbH (2024): *Benutzer:innen und Rollen in WordPress einfach verwalten*,
<https://www.world4you.com/blog/wordpress-benutzer-rollen/> [Stand: 20.11.2024]

WP Engine Inc. (o.J.): *All features*, <https://localwp.com/features/> [Stand: 15.09.2024]

ABBILDUNGSVERZEICHNIS

Abbildung 1: Organisation des Landesfeuerwehrverbandes Steiermark, Quelle: Eigene Darstellung.....	3
Abbildung 2: Hierarchiestruktur der österreichischen Feuerwehren, Quelle: Eigene Darstellung.	4
Abbildung 3: Wissenstestabzeichen in der Stufe Bronze, Silber und Gold, Quelle: Niederösterreichischer Landesfeuerwehrverband (o.J.), Online-Quelle [19.05.2024].	6
Abbildung 4: Ausschnitt aus dem österreichisches Bildungssystem, Quelle: Agentur für Bildung und Internationalisierung (o.J.), Online-Quelle [30.10.2024].....	7
Abbildung 5: Originaldarstellung des Zimmermannsklank (Zimmermannsstek) aus dem Wissenstest Silber, Quelle: Eigene Darstellung.	9
Abbildung 6: Beispielhafte Darstellung der Datenbankstruktur, Quelle: BCM Solutions (o.J.), Online-Quelle [08.06.2024].	11
Abbildung 7: Darstellung des Feuerwehrdateninformationssystem und Katastrophenschutzmanagement, Quelle: Eigene Darstellung.....	12
Abbildung 8: Gefährliche Stoffe Fahrzeug (GSF) der FF Lebring, Quelle: Freiwillige Feuerwehr Lebring (o.J.), Online-Quelle [20.05.2024].	14
Abbildung 9: Frage aus dem Wissenstest Stufe Gold, Quelle: Eigene Darstellung.	14
Abbildung 10: Darstellung der MySQL Workbench des LMSs, Quelle: Eigene Darstellung.	15
Abbildung 11: Darstellung der Datenbanktabelle users, Quelle: Eigene Darstellung.	16
Abbildung 12: Darstellung der Datenbanktabelle mappingtable_wt Teil 1, Quelle: Eigene Darstellung. ..	17
Abbildung 13: Darstellung der Datenbanktabelle mappingtable_wt Teil 2, Quelle: Eigene Darstellung. ..	17
Abbildung 14: Darstellung der Datenbanktabelle mappingtable_user, Quelle: Eigene Darstellung.	18
Abbildung 15: Darstellung der mappingtable_user, Quelle: Eigene Darstellung.	18
Abbildung 16: Darstellung der Datenbanktabelle vehicles_bases, Quelle: Eigene Darstellung.	19
Abbildung 17: Darstellung eines Serverraums, Quelle: Adobe Stock (o.J.), Online-Quelle [22.11.2024] .	21
Abbildung 18: Darstellung des Protokoll-Präfix HTTPS im Webbrowser, Quelle: Eigene Darstellung.....	22
Abbildung 19: Das Cookie-Banner des Lernmanagementsystems, Quelle: Eigene Darstellung.	23
Abbildung 20: Darstellung Local® des Lernmanagementsystems, Quelle: Eigene Darstellung.	31
Abbildung 21: Darstellung der IDE PhpStorm® mit Syntax-Highlighting und Fehlererkennung, Quelle: Eigene Darstellung.	33
Abbildung 22: Beispiel einer Datenbankinteraktion mit PHP, Quelle: Eigene Darstellung.	34
Abbildung 23: Darstellung eines HTML-Codes aus dem LMS, Quelle: Eigene Darstellung.....	35
Abbildung 24: Darstellung der CSV-Datei aus dem Auszug des FDISK, Quelle: Eigene Darstellung.	37

Abbildung 25: Verbindungsaufbau mit der MySQL Datenbank, Quelle: Eigene Darstellung.	37
Abbildung 26: Pfadbeschreibung des CSV-Files, Quelle: Eigene Darstellung.	38
Abbildung 27: Darstellung der Funktionen "Erstellen der Session ID" und "Authentifizieren des Benutzers", Quelle: Eigene Darstellung.	38
Abbildung 28: Darstellung des ersten Teils des Datenabgleichs zwischen Benutzereingabe und CSV-Datei, Quelle: Eigene Darstellung.	38
Abbildung 29: Darstellung des zweiten Teils des Datenabgleichs zwischen Benutzereingabe und CSV-Datei, Quelle: Eigene Darstellung.	39
Abbildung 30: Darstellung der Übergabe der Daten aus dem POST-Array, Quelle: Eigene Darstellung.	39
Abbildung 31: Darstellung der Funktion Benutzerauthentifizierung, Quelle: Eigene Darstellung.	40
Abbildung 32: Darstellung der Benutzerauthentifizierung und der Datenbankeintragung, Quelle: Eigene Darstellung.	40
Abbildung 33: Darstellung der Datenbanktabelle nach erfolgreicher Anmeldung, Quelle: Eigene Darstellung.	41
Abbildung 34: Darstellung der Weiterleitung nach erfolgreicher Anmeldung, Quelle: Eigene Darstellung.	41
Abbildung 35: Darstellung der Datenextraktion aus der Datenbanktabelle und des Aufrufs der Verschlüsselungsfunktion, Quelle: Eigene Darstellung.	42
Abbildung 36: Darstellung der Funktion der Vigenère-Verschlüsselung, Quelle: Eigene Darstellung.	43
Abbildung 37: Darstellung der "String to lower" Funktion, Quelle: Eigene Darstellung.	43
Abbildung 39: Darstellung der Datenbanktabellenabfrage, Quelle: Eigene Darstellung.	44
Abbildung 38: Darstellung Übergabe der Werte an die Datenbanktabelle mappingtable_user, Quelle: Eigene Darstellung.	44
Abbildung 40: Darstellung des catch-Blocks am Skriptende, Quelle: Eigene Darstellung.	44
Abbildung 41: Darstellung der Funktion incrementLearnmodeCount und Funktionsaufruf, Quelle: Eigene Darstellung.	46
Abbildung 42: Darstellung des HTML-Teils zur Definition der verwendeten Buttons, Quelle: Eigene Darstellung.	47
Abbildung 43: Ausschnitt Stationsabfrage der jeweiligen Wissenstest Stufe, Quelle: Eigene Darstellung.	47
Abbildung 44: Darstellung des HTML-Teils zur Generierung der verwendeten Buttons, Quelle: Eigene Darstellung.	48
Abbildung 45: Darstellung der Datenbankabfragen für den Lernmodus, Quelle: Eigene Darstellung.	48
Abbildung 46: Darstellung der verschachtelten if-Abfrage, Quelle: Eigene Darstellung.	49

Abbildung 47: Darstellung der Unterscheidung des Dateiformats der Antwort, Quelle: Eigene Darstellung.	50
Abbildung 48: Darstellung der Benutzer-ID Abfrage, Quelle: Eigene Darstellung.	51
Abbildung 49: Berechnung der Punktezah im Prüfungsmodus, Quelle: Eigene Darstellung.	51
Abbildung 50: Ausschnitt aus dem Skript des Prüfungsmodus, Quelle: Eigene Darstellung.	52
Abbildung 51: Darstellung des HTML-Teils des Prüfungsmodus, Quelle: Eigene Darstellung.	53
Abbildung 52: Ausschnitt aus dem HTML-Teil des Skript des Prüfungsmodus, Quelle: Eigene Darstellung.	53
Abbildung 53: Darstellung der Funktion für die Vigenère-Verschlüsselung aus dem Auswertungs-Skript, Quelle: Eigene Darstellung	54
Abbildung 54: Darstellung des Skript-Teils zur Trennung der Wissensteststufen, Quelle: Eigene Darstellung.	55
Abbildung 55: Darstellung des Skript-Teils zur Ermittlung der Stationen und maximalen Punktezah, Quelle: Eigene Darstellung	55
Abbildung 56: Darstellung des Skript-Teils zur Ausgabevorbereitung, Quelle: Eigene Darstellung.	56
Abbildung 57: Startseite des Lernmanagementsystems, Quelle: Eigene Darstellung.	57
Abbildung 58: Kopfbereich des Lernmanagementsystems, Quelle: Eigene Darstellung.	58
Abbildung 59: Fußbereich des Lernmanagementsystems, Quelle: Eigene Darstellung.	58
Abbildung 60: Benutzeranmeldung des Lernmanagementsystems, Quelle: Eigene Darstellung.	59
Abbildung 61: Darstellung des Dropdown-Menüs der Benutzeranmeldung, Quelle: Eigene Darstellung.	59
Abbildung 62: Darstellung der Webseite des Dashboards für die Jugendlichen, Quelle: Eigene Darstellung.	60
Abbildung 63: Auswahl der Wissensteststufe im Lernmodus, Quelle: Eigene Darstellung.	60
Abbildung 64: Stationsübersicht der Wissensteststufe Silber im Lernmodus, Quelle: Eigene Darstellung.	61
Abbildung 65: Darstellung der ersten Frage aus dem Wissenstest Bronze im Lernmodus, Quelle: Eigene Darstellung.	61
Abbildung 66: Darstellung einer Fragestellung im Prüfungsmodus des Wissenstest Bronze, Quelle: Eigene Darstellung.	62
Abbildung 67: Darstellung bei der Auswahl einer richtigen Antwort im Prüfungsmodus, Quelle: Eigene Darstellung.	62
Abbildung 68: Darstellung bei der Auswahl einer falschen Antwort im Prüfungsmodus, Quelle: Eigene Darstellung.	63
Abbildung 69: Darstellung der Ergebnisausgabe der Jugendlichen, Quelle: Eigene Darstellung.	63

Abbildung 70: Darstellung des Dashboards der Funktionäre, Quelle: Eigene Darstellung.	64
Abbildung 71: Darstellung der durchschnittlichen Ergebnisse, Quelle: Eigene Darstellung.	64
Abbildung 72: Darstellung der Einzelauswertung ohne Auswahl einer*s Jugendlichen, Quelle: Eigene Darstellung.	65
Abbildung 73: Darstellung der Einzelauswertung mit Auswahl einer*s Jugendlichen, Quelle: Eigene Darstellung.	65

TABELLENVERZEICHNIS

Tabelle 1: Übersicht Feuerwehren in Österreich, Quelle: Eigene Darstellung.....	2
Tabelle 2: Übersicht der LOB Datentypen, Quelle: Hauser / Wenz (2024), S. 624 (leicht modifiziert).....	27
Tabelle 3: Beispiel der Vigenère-Verschlüsselung, Quelle: Eigene Darstellung.....	45
Tabelle 4: Beispiel der Vigenère-Entschlüsselung, Quelle: Eigene Darstellung.....	56

ABKÜRZUNGSVERZEICHNIS

BFV	Bereichsfeuerwehrverband
BGBLA	Bundesgesetzblatt
CMS	Content-Management-System
DBMS	Datenbank-Management-System
DSGVO	Datenschutz-Grundverordnung
FAS	Feuerwehrrabschnitte
FDISK	Feuerwehrdateninformationssystem und Katastrophenschutzmanagement
FF	Freiwillige Feuerwehr
FJLA	Feuerwehr Jugendleistungsabzeichen
FTP	File Transfer Protocol / Dateiübertragungsprotokoll
FTPS	File Transfer Protocol Secure
FWZS	Feuerwehr- und Zivilschutzschule
HTTP	Hypertext Transfer Protocol / Hypertextübertragungsprotokoll
HTTPS	Hypertext Transfer Protocol Secure / Sicheres Hypertextübertragungsprotokoll
IDE	Integrated Development Environment / integrierte Entwicklungsumgebung
LFV	Landesfeuerwehrverband
LMS	Lernmanagementsystem
OOP	Object Oriented Programming / Objektorientierte Programmierung
ÖBFV	Österreichischer Bundesfeuerwehrverband
RFC	Request For Comments
SEO	Search Engine Optimization / Suchmaschinenoptimierung
SFTP	Secure File Transfer Protocol
SLD	Second-Level-Domain
STMK	Steiermark
TLD	Top-Level-Domain
WT	Wissenstest
WTS	Wissenstest-Spiele

ANHANG 1: QUELLCODE BENUTZERANMELDUNG (PLUGIN)

```
<?php
/*
Header: Master Thesis - Learn Management System 2024
Plugin Name: Benutzeranmeldung.php
Description: This plugin should handle the user verification for the member login and redirection
Version: 4.3 - Change from cookie to database session handling
Author: Beichler Christoph Franz
PHP Version: 8.3
DB Version: MySQL 8.0.16
WP Version: 6.6.2
Web Server: nginx
*/

// Activation of error viewer
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

// Start of output-buffer
ob_start();
session_start();

// Establish connection to database
$host = 'localhost';
$db = 'questionnaire_wt2024';
$username = 'ChrisBeich';
$password = 'XXXXXXXXXXXX';

// Try to connect with database with given information
try {
    $pdo = new PDO("mysql:host=$host;dbname=$db", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    // If database connection could not get established -> send error message to screen
    die("Datenbankverbindung fehlgeschlagen: " . $e->getMessage());
}

// Path for CSV-file from LFV Steiermark
$csvFile = 'C:\Users\Christoph\Local Sites\lern-management-system\app\public\wp-content\plugins\userlogin\datenbank_mitglieder.csv';

// Function to generate a unique session ID - will be needed when the website goes online
function generateSessionID() {
    return bin2hex(random_bytes(16)); // Generates a 32-character hexadecimal string
}

// Function for user authentication
function authenticateUser($name, $surname, $firebrigade, $registrynumber, $function, $csvFile):
bool {
    if (!file_exists($csvFile)) {
        die("Die CSV-Datei konnte nicht geladen werden! " . $csvFile);
    }

    // Read from CSV-file
    if (($handle = fopen($csvFile, 'r')) !== FALSE) {
        while (($data = fgetcsv($handle, 75, ';')) !== FALSE) {
            // Get data, convert them to lower string and remove all unnecessary spaces
            $csvName = trim(strtolower($data[0]));
            $csvSurname = trim(strtolower($data[1]));
            $csvFirebrigade = trim(strtolower($data[2]));
            $csvRegistryNumber = trim(strtolower($data[3]));
            $csvFunction = trim(strtolower($data[4]));

            $inputName = trim(strtolower($name));
            $inputSurname = trim(strtolower($surname));
            $inputFirebrigade = trim(strtolower($firebrigade));
            $inputRegistryNumber = trim(strtolower($registrynumber));
            $inputFunction = trim(strtolower($function));

            if ($csvName == $inputName &&
                $csvSurname == $inputSurname &&
                $csvFirebrigade == $inputFirebrigade &&
                $csvRegistryNumber == $inputRegistryNumber &&
                $csvFunction == $inputFunction) {
                fclose($handle);
                return true;
            }
        }
    }
}
```

```

    }
    fclose($handle);
}
return false;
}
// Process data from POST-request
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $name = $_POST['name'];
    $surname = $_POST['surname'];
    $firebrigade = $_POST['firebrigade'];
    $registrynumber = $_POST['registrynumber'];
    $function = $_POST['function'];
    // User authentication -> CSV X database
    if (authenticateUser($name, $surname, $firebrigade, $registrynumber, $function, $csvFile)) {
        $stmt = $pdo->prepare("SELECT id, TIMESTAMPDIFF(MINUTE, created_at, NOW()) AS
minutes_since_created FROM users WHERE name = :name AND surname = :surname AND firebrigade =
:firebrigade AND registrynumber = :registrynumber AND `function` = :function");
        $stmt->execute([
            ':name' => $name,
            ':surname' => $surname,
            ':firebrigade' => $firebrigade,
            ':registrynumber' => $registrynumber,
            ':function' => $function
        ]);

        $existingUser = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($existingUser && $existingUser['minutes_since_created'] < 30) {
            $updateStmt = $pdo->prepare("UPDATE users SET created_at = NOW() WHERE id = :id");
            $updateStmt->execute([':id' => $existingUser['id']]);
            // Retrieve the existing session ID
            $sessionID = $existingUser['session_id'];
        } else {
            $sessionID = generateSessionID(); // Generate a new session ID
            $insertStmt = $pdo->prepare("INSERT INTO users (name, surname, firebrigade,
registrynumber, `function`, created_at, session_id) VALUES (:name, :surname, :firebrigade,
:registrynumber, :function, NOW(), :session_id)");
            $insertStmt->execute([
                ':name' => $name,
                ':surname' => $surname,
                ':firebrigade' => $firebrigade,
                ':registrynumber' => $registrynumber,
                ':function' => $function,
                ':session_id' => $sessionID,
            ]);
        }

        session_regenerate_id(true);
        $_SESSION['session_id'] = $sessionID;
        $_SESSION['user_id'] = $existingUser ? $existingUser['id'] : $pdo->lastInsertId();

        // Forwarding after successful comparison of user data
        if ($function == "jff/jfm") {
            header("Location: http://lern-management-system.local/dashboard-jugend");
            exit();
        } elseif ($function == "orts-jugendbeauftragte*r" || $function == "feuerwehrjugend-
betreuer*in") {
            header("Location: http://lern-management-system.local/dashboard-funktionaere");
            exit();
        }
    } else {
        echo "Anmeldung fehlgeschlagen. Die eingegebenen Daten stimmen nicht mit der CSV-Datei
überein.<br>";
    }
}
// Stop output-buffer and send data
ob_end_flush();
?>

<!-- |#|#|#|#|#| Additional HTML code |#|#|#|#|#| -->
<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Page</title>
    <style>
        body {

```

```

        font-family: Arial, sans-serif;
        display: flex;
        justify-content: center;
        align-items: center;
        min-height: 100vh;
        margin: 0;
        background-color: #f4f4f9;
    }
    .login-container {
        width: 100%;
        max-width: 400px;
        background: #fff;
        padding: 2rem;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        border-radius: 8px;
    }
    h2 {
        text-align: center;
        color: #333;
    }
    .error-message {
        color: red;
        text-align: center;
        margin-bottom: 1rem;
    }
    label {
        display: block;
        margin-bottom: 0.5rem;
        color: #333;
    }
    input[type="text"], select {
        width: 100%;
        padding: 0.5rem;
        margin-bottom: 1rem;
        border: 1px solid #ddd;
        border-radius: 4px;
        box-sizing: border-box;
    }
    input[type="submit"] {
        width: 100%;
        padding: 0.7rem;
        background-color: #007bff;
        color: white;
        border: none;
        border-radius: 4px;
        cursor: pointer;
        font-size: 1rem;
    }
    input[type="submit"]:hover {
        background-color: #0056b3;
    }
</style>
</head>
<body>

<div class="login-container">
    <h2>Login</h2>

    <?php if (isset($error)): ?>
        <p class="error-message"><?php echo $error; ?></p>
    <?php endif; ?>

    <form method="post" action="userlogin.php">
        <label for="name">Vorname:</label>
        <input type="text" name="name" id="name" required>

        <label for="surname">Nachname:</label>
        <input type="text" name="surname" id="surname" required>

        <label for="firebrigade">Feuerwehr:</label>
        <input type="text" name="firebrigade" id="firebrigade" required>

        <label for="registrynumber">Standesbuchnummer:</label>
        <input type="text" name="registrynumber" id="registrynumber" required>

        <label for="function">Funktion:</label>
        <select name="function" id="function" required>
            <option value="" disabled selected>Bitte wählen Sie eine Funktion</option>
            <option value="JFM/JFF">JFM/JFF</option>
        </select>
    </form>
</div>

```

```
        <option value="Feuerwehrjugend-Betreuer*in">Feuerwehrjugend-Betreuer*in</option>
        <option value="Orts-Jugendbeauftragte*r">Orts-Jugendbeauftragte*r</option>
        <option value="Abs-Jugendbeauftragte*r">Abs-Jugendbeauftragte*r</option>
        <option value="BFV-Jugendbeauftragte*r">BFV-Jugendbeauftragte*r</option>
        <option value="LFV-Jugendbeauftragte*r">LFV-Jugendbeauftragte*r</option>
    </select>

    <input type="submit" value="Anmelden">
</form>
</div>

</body>
</html>
```


ANHANG 2: QUELLCODE DATENBANK HANDSHAKE

```
<?php
/*
Header: Master Thesis - Learn Management System 2024
Plugin Name: Datenbank_Handshake.php
Description: This code is used to transfer data from the 'users' table to 'mappingtable_user',
with encrypted surnames.
Version: 2.13
Author: Beichler Christoph Franz
PHP Version: 8.3
DB Version: MySQL 8.0.16
WP Version: 6.6.2
Web Server: nginx
*/

$host = 'localhost';
$db = 'questionnaire_wt2024';
$username = 'ChrisBeich';
$password = 'XXXXXXXXXX';
// Establish host connection
try {
    $pdo = new PDO("mysql:host=$host;dbname=$db", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Datenbankverbindung fehlgeschlagen: " . $e->getMessage());
}

// Vigenère-encryption
function vigenereEncrypt($text, $key) {
    $text = strtoupper($text);
    $key = strtoupper($key);
    $keyLength = strlen($key);
    $textLength = strlen($text);
    $encryptedText = '';

    for ($i = 0; $i < $textLength; $i++) {
        $textChar = $text[$i];
        if (ctype_alpha($textChar)) {
            $textCharCode = ord($textChar) - 65;
            $keyCharCode = ord($key[$i % $keyLength]) - 65;
            $encryptedCharCode = ($textCharCode + $keyCharCode) % 26 + 65;
            $encryptedText .= chr($encryptedCharCode);
        } else {
            $encryptedText .= $textChar;
        }
    }
    return $encryptedText;
}

// Get data from table 'users'
try {
    $stmt = $pdo->query("SELECT id, name, surname, firebrigade FROM users");
    $users = $stmt->fetchAll(PDO::FETCH_ASSOC);

    foreach ($users as $user) {
        $name = $user['name'];
        $surname = $user['surname'];
        $firebrigade = $user['firebrigade'];
        $codedsurname = vigenereEncrypt($surname, 'Floriani');

        // Prepare data for database query
        $lowercase_name = strtolower($name);
        $lowercase_codedsurname = strtolower($codedsurname);
        $lowercase_firebrigade = strtolower($firebrigade);

        // Check for existing entries in 'mappingtable_user'
        $checkStmt = $pdo->prepare("
            SELECT iduser FROM mappingtable_user
            WHERE LOWER(TRIM(name)) = :name
            AND LOWER(TRIM(codedsurname)) = :codedsurname
            AND LOWER(TRIM(firebrigade)) = :firebrigade
        ");
        $checkStmt->bindParam(':name', $lowercase_name);
        $checkStmt->bindParam(':codedsurname', $lowercase_codedsurname);
        $checkStmt->bindParam(':firebrigade', $lowercase_firebrigade);
        $checkStmt->execute();
        $existingEntry = $checkStmt->fetch(PDO::FETCH_ASSOC);
```

```

if ($existingEntry) {
    // Refresh of parameters in case of identified user currently not needed
    $current_iduser = $existingEntry['iduser'];

    $updateStmt = $pdo->prepare("
        UPDATE mappingtable_user
        SET name = :name, codedsurname = :surname, firebrigade = :firebrigade
        WHERE iduser = :iduser
    ");
    $updateStmt->bindParam(':iduser', $current_iduser, PDO::PARAM_INT);
    $updateStmt->bindParam(':name', $name);
    $updateStmt->bindParam(':surname', $codedsurname);
    $updateStmt->bindParam(':firebrigade', $firebrigade);

    /*
        if ($updateStmt->execute()) {
            echo "Eintrag mit iduser=$current_iduser aktualisiert.\n";
        } else {
            echo "Fehler beim Aktualisieren des Eintrags.\n";
        }
    */

} else {
    // echo "Kein Eintrag gefunden, füge neuen Eintrag hinzu...\n";
    $stmt = $pdo->query("SELECT MAX(iduser) AS max_id FROM mappingtable_user");
    $row = $stmt->fetch(PDO::FETCH_ASSOC);
    $new_iduser = $row['max_id'] + 1;

    $insertStmt = $pdo->prepare("
        INSERT INTO mappingtable_user (iduser, name, codedsurname, firebrigade)
        VALUES (:iduser, :name, :surname, :firebrigade)
    ");
    $insertStmt->bindParam(':iduser', $new_iduser, PDO::PARAM_INT);
    $insertStmt->bindParam(':name', $name);
    $insertStmt->bindParam(':surname', $codedsurname);
    $insertStmt->bindParam(':firebrigade', $firebrigade);
    $insertStmt->execute();

    /*
        if ($insertStmt->execute()) {
            echo "Neuer Eintrag erfolgreich mit iduser=$new_iduser in die Datenbank
eingefügt!\n";
        } else {
            echo "Fehler beim Einfügen der Daten...\n";
        }
    */

}
}
} catch (PDOException $e) {
    echo "Fehler bei der Datenverarbeitung: " . $e->getMessage();
}
?>

```

ANHANG 3: QUELLCODE LERNMODUS FÜR DIE DREI STUFEN BRONZE, SILBER UND GOLD – BEISPIEL FÜR BRONZE

```
<?php
/*
Header: Master Thesis - Learn Management System 2024
Plugin/Skript Name: Lernmodus_php.php
HTML File: included
Description: This code is used for the learn mode - bronze level
Version: 7.3
Author: Beichler Christoph Franz
PHP Version: 8.3
DB Version: MySQL 8.0.16
WP Version: 6.6.2
Web Server: nginx
*/

// Activate error monitoring
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

// Define host connection data
$host = 'localhost';
$db = 'questionnaire_wt2024';
$username = 'ChrisBeich';
$password = 'XXXXXXXXXX';

// Establish database connection
try {
    $pdo = new PDO("mysql:host=$host;dbname=$db", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Datenbankverbindung fehlgeschlagen: " . $e->getMessage());
}

// Define level for test purpose - will get information from FDISK in the future
$level = 1; // Edit for different level

// Get latest user id - different approach after FDISK connection -> via Session
$stmt = $pdo->query("SELECT MAX(iduser) AS last_iduser FROM mappingtable_user");
$lastUser = $stmt->fetch(PDO::FETCH_ASSOC);
$iduser = $lastUser['last_iduser'];

// Get user information
$stmt = $pdo->prepare("SELECT name, codedsurname, firebrigade FROM mappingtable_user WHERE iduser = :iduser");
$stmt->execute([':iduser' => $iduser]);
$userData = $stmt->fetch(PDO::FETCH_ASSOC);

// Debugging - check if user data got loaded
if (!$userData) {
    echo "<p>Benutzerinformationen nicht gefunden für iduser = $iduser. Überprüfen Sie die Datenbank und die Spaltennamen.</p>";
    // Define standard value to avoid errors
    $name = 'Unbekannt';
    $codedsurname = 'Unbekannt';
    $firebrigade = 'Unbekannt';
} else {
    // Write results to variables
    $name = isset($userData['name']) ? $userData['name'] : 'Unbekannt';
    $codedsurname = isset($userData['codedsurname']) ? $userData['codedsurname'] : 'Unbekannt';
    $firebrigade = isset($userData['firebrigade']) ? $userData['firebrigade'] : 'Unbekannt';
}

// Increment learn mode count or initialize it to 1 if NULL
function incrementUserCount($pdo, $iduser) {
    try {
        // Check if a value in learnmodecount is already existing
        $stmt = $pdo->prepare("SELECT learnmodecount FROM mappingtable_user WHERE iduser = :iduser");
        $stmt->execute([':iduser' => $iduser]);
        $currentLearnModeCount = $stmt->fetch(PDO::FETCH_ASSOC)['learnmodecount'];

        if ($currentLearnModeCount === null) {
            // If learnmodecount is NULL -> set value to 1

```

```

        $stmt = $pdo->prepare("UPDATE mappingtable_user SET learnmodecount = 1 WHERE iduser = :iduser");
    } else {
        // Else increase value by 1
        $stmt = $pdo->prepare("UPDATE mappingtable_user SET learnmodecount = learnmodecount + 1 WHERE iduser = :iduser");
    }
    $stmt->execute([':iduser' => $iduser]);
} catch (PDOException $e) {
    echo "Fehler beim Aktualisieren von learnmodecount: " . $e->getMessage() . "<br>";
}

}

// Call incrementUserCount function
incrementUserCount($pdo, $iduser);

// Show stations if no station was chosen
if (!isset($_POST['station_num'])) {
    // Get all stations for bronze level
    $stmt = $pdo->prepare("SELECT DISTINCT station_number, station FROM mappingtable_wt WHERE j_level = :level");
    $stmt->execute([':level' => $level]);
    $stations = $stmt->fetchAll(PDO::FETCH_ASSOC);

    ?>

<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Lernmodus - Wissenstest Bronze</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <style>
        .main-container {
            padding-top: 40px;
            padding-bottom: 40px;
            background-color: transparent;
        }
        .button-container {
            display: flex;
            flex-direction: column;
            align-items: center;
            gap: 10px;
            margin-top: 20px;
        }
        .btn-station {
            width: 300px;
            font-size: 1.2rem;
            background-color: #6c757d;
            color: white;
            border: none;
            border-radius: 5px;
        }
        .navigation-buttons {
            margin-top: 30px;
            display: flex;
            gap: 20px;
            justify-content: center;
        }
        .btn-nav {
            width: 200px;
            font-size: 1.1rem;
            font-weight: bold;
            background-color: #e57373;
            color: white;
            text-align: center;
            display: flex;
            align-items: center;
            justify-content: center;
            height: auto;
            padding: 10px 20px;
            border: none;
            border-radius: 5px;
            line-height: 1.2;
            white-space: normal;
        }
    </style>
</head>

```

```

<body>

<div class="container main-container text-center">
  <h1 class="mb-4">Wählen Sie eine Station für<br>den Wissenstest Bronze:</h1>

  <!-- Debugging only for test reasons
  <p>Name: <?= htmlspecialchars($name) ?></p>
  <p>Codierter Nachname: <?= htmlspecialchars($codedsurname) ?></p>
  <p>Feuerwehr: <?= htmlspecialchars($firebrigade) ?></p>
  -->

  <!-- Identify the stations stored in the database -->
  <form method="POST" action="">
    <div class="button-container">
      <?php foreach ($stations as $station): ?>
        <button type="submit" name="station_num" value="<?=
htmlspecialchars($station['station_number']) ?>" class="btn btn-station">
          <?= htmlspecialchars($station['station']) ?>
        </button>
      <?php endforeach; ?>
    </div>
  </form>

  <!-- Additional navigation buttons -->
  <div class="navigation-buttons">
    <a href="http://lern-management-system.local/lernmodus/" class="btn btn-nav">Auswahl
Wissenstest</a>
    <a href="http://lern-management-system.local/dashboard-jugend/" class="btn btn-
nav">Lernmodusauswahl</a>
  </div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

<?php
} else {
  // Following part contains the Q&A if a station got chosen
  $station_num = $_POST['station_num'];

  // Load questions for the selected station
  function getQuestions($pdo, $level, $station_num) {
    $stmt = $pdo->prepare("SELECT idwt, question, answer FROM mappingtable_wt WHERE j_level =
:level AND station_number = :station_num ORDER BY idwt");
    $stmt->execute(['level' => $level, 'station_num' => $station_num]);
    return $stmt->fetchAll(PDO::FETCH_ASSOC);
  }

  $questions = getQuestions($pdo, $level, $station_num);

  // Get user progress and refresh data in database
  $stmt = $pdo->prepare("SELECT * FROM user_progress WHERE user_id = :user_id AND station_num =
:station_num AND level = :level");
  $stmt->execute(['user_id' => $iduser, 'station_num' => $station_num, 'level' => $level]);
  $progress = $stmt->fetch(PDO::FETCH_ASSOC);

  // Check if progress for this station is existing -> create new entry if not
  if(!$progress) {
    // Create new entry in database for user progress
    $currentQuestionIndex = 1;
    $stmt = $pdo->prepare("INSERT INTO user_progress (user_id, station_num,
current_question_index, level) VALUES (:user_id, :station_num, :current_question_index,
:level)");
    $stmt->execute(['user_id' => $iduser, 'station_num' => $station_num,
'current_question_index' => $currentQuestionIndex, 'level' => $level]);
  } else {
    // Load progress if entry is existing
    $currentQuestionIndex = $progress['current_question_index'];
  }

  // $currentQuestionIndex = $progress ? $progress['current_question_index'] : 1; // Starts with
1 and not 0

  // Identification of which button got pressed (previous, answer, forward)
  if (isset($_POST['action'])) {
    if ($_POST['action'] == 'previous') {
      $currentQuestionIndex = max(1, $currentQuestionIndex - 1); // Avoids a question index

```

```

below 1
    } elseif ($_POST['action'] == 'next') {
        // Check if current question is last question of station
        if ($currentQuestionIndex >= count($questions)) {
            echo "Letzte Frage erreicht. Eintrag wird gelöscht.";
            $pdo->beginTransaction();
            try {
                // Delete values for station_num and current_question_index after last
                question got passed
                $stmt = $pdo->prepare("DELETE FROM user_progress WHERE user_id = :user_id AND
station_num = :station_num AND level = :level");
                $stmt->execute(['user_id' => $idUser, 'station_num' => $station_num, 'level'
=> $level]);
                $pdo->commit();

                // This header is needed to move back to the station overview after the last
                Q&A got finished
                header("Location: http://lern-management-system.local/lernmodus-wissenstest-
bronze/");
                exit();
            } catch (PDOException $e) {
                $pdo->rollBack();
                die("Fehler beim Löschen des Fortschritts: " . $e->getMessage());
            }
        } else {
            $currentQuestionIndex = min(count($questions), $currentQuestionIndex + 1); //
            Max. count only up to last question
        }
    }
}

// Load current Q&A based on the current_question_index in the database
$currentQuestionRaw = $questions[$currentQuestionIndex - 1]['question']; // -1 cause index
starts with 1
$currentQuestion = htmlspecialchars($currentQuestionRaw);
$currentQuestion = str_replace('/', '<br>', $currentQuestion);
// Use only for text based answers - got replaced by text + pics + vids
// $currentAnswer = str_replace('/', '<br>', $questions[$currentQuestionIndex - 1]['answer']);

$currentAnswer = $questions[$currentQuestionIndex - 1]['answer'];

// Check if the answer is a pic, vid or text
if (preg_match('/\.(jpg|jpeg|png|gif|bmp)$/i', $currentAnswer)) {
    // Show picture
    $currentAnswer = "<img src='" . htmlspecialchars($currentAnswer) . "' alt='Antwort-Bild'
style='max-width:100%; height:auto;'>";
} elseif (preg_match('/\.(mov|mp4)$/i', $currentAnswer)) {
    $currentAnswer = "<video controls style='max-width:100%; height:auto;' src='" .
htmlspecialchars($currentAnswer) . "' type='video/mp4'>Ihr Browser unterstützt dieses Videoformat
nicht.</video>";
} else {
    $currentAnswer = str_replace('/', '<br>', htmlspecialchars($currentAnswer));
}

// Save progress in database
$stmt = $pdo->prepare("UPDATE user_progress SET current_question_index =
:current_question_index WHERE user_id = :user_id AND station_num = :station_num AND level =
:level");
$stmt->execute(['current_question_index' => $currentQuestionIndex, 'user_id' => $idUser,
'station_num' => $station_num, 'level' => $level]);
?>

<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Fragen für Wissenstest</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <style>
        <!-- Define style of Q&A area including the 3 buttons -->
        .question-container {
            margin-top: 50px;
        }
        .answer {
            display: none;
            margin-top: 20px;
            font-weight: bold;

```

```

        color: #007bff;
    }
    .btn-previous, .btn-next, .btn-answer {
        margin-top: 20px;
        width: 150px;
        font-size: 1.2rem;
        background-color: #6c757d;
        color: white;
        border: none;
        border-radius: 5px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
    }
    .btn-back {
        width: 200px;
        font-size: 1.1rem;
        font-weight: bold;
        background-color: #e57373;
        color: white;
        text-align: center;
        display: flex;
        align-items: center;
        justify-content: center;
        height: auto;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        line-height: 1.2;
        white-space: normal;
    }
    .button-group {
        display: flex;
        gap: 10px;
        justify-content: center;
    }
    .back-button-container {
        display: flex;
        justify-content: center;
        margin-top: 35px;
    }
}
</style>
</head>
<body>

<div class="container text-center question-container">
    <h2>Frage <?= $currentQuestionIndex ?>:</h2>
    <p id="question"><?= $currentQuestion ?></p>
    <p id="answer" class="answer"><?= nl2br($currentAnswer) ?></p>

    <!-- Button creation and function description -->
    <form method="POST" action="">
        <div class="button-group">
            <!-- Back button -->
            <button type="submit" name="action" value="previous" class="btn btn-previous" <?=
$currentQuestionIndex === 1 ? 'disabled' : '' ?>>Zurück</button>
            <!-- Show answer button -->
            <button type="button"
onclick="document.getElementById('answer').style.display='block';" class="btn btn-
answer">Antwort</button>
            <!-- Forward button -->
            <button type="submit" name="action" value="next" class="btn btn-
next">Weiter</button>
        </div>
        <input type="hidden" name="station_num" value="<?= htmlspecialchars($station_num)
?>">
    </form>

    <!-- Button to go directly back to station overview -->
    <div class="back-button-container">
        <a href="http://lern-management-system.local/lernmodus-wissenstest-bronze/"
class="btn btn-back">Zur Stationsübersicht</a>
    </div>
</div>

<script>
    function showAnswer() {
        document.getElementById("answer").style.display = "block";
    }

```

```

    </script>
  </body>
</html>

  <?php
}

    <!-- ||#||#||#||#|| Additional HTML code ||#||#||#||#|| -->

<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>Quiz</title>
</head>
<body>

<h1>Wähle eine Kategorie und Unterkategorie für dein Quiz</h1>

<form method="POST" action="">
  <label for="category">Kategorie:</label>
  <select name="category" id="category" required>
    <?php foreach ($categories as $category): ?>
      <option value="<?= htmlspecialchars($category) ?>"><?= htmlspecialchars($category)
?></option>
    <?php endforeach; ?>
  </select>

  <label for="subcategory">Unterkategorie:</label>
  <input type="text" name="subcategory" id="subcategory" required>

  <button type="submit">Quiz starten</button>
</form>

<?php
// Quiz anzeigen, wenn das Formular abgeschickt wurde
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
  $category = $_POST['category'];
  $subcategory = $_POST['subcategory'];
  $questions = getQuestions($pdo, $category, $subcategory);

  if ($questions) {
    echo "<h2>Quiz: $category - $subcategory</h2>";
    echo "<form>";
    foreach ($questions as $index => $question) {
      echo "<p><strong>Frage " . ($index + 1) . ":</strong> " .
htmlspecialchars($question['question']) . "</p>";
      echo "<input type='text' name='answer_" . $index . "' placeholder='Antwort
eingeben'><br><br>";
    }
    echo "<button type='submit'>Quiz abschließen</button>";
    echo "</form>";
  } else {
    echo "<p>Keine Fragen in dieser Kategorie und Unterkategorie gefunden.</p>";
  }
}
?>

</body>
</html>

    <!-- ||#||#||#||#|| Additional HTML code for level selction ||#||#||#||#|| -->

<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>Lernmodus Auswahl</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">
  <style>
    .btn-bronze {
      background-color: #cd7f32;
      color: white;
      width: 150px;
      font-size: 1.2rem;
    }
    .btn-silver {
      background-color: #c0c0c0;
      color: black;

```



```

        width: 150px;
        font-size: 1.2rem;
    }
    .btn-gold {
        background-color: #ffd700;
        color: black;
        width: 150px;
        font-size: 1.2rem;
    }
</style>
</head>
<body>

<div class="container text-center mt-5">
    <h1 class="mb-4">Bitte die gewünschte Wissensteststufe auswählen</h1>
    <div class="d-flex justify-content-center gap-3">
        <!-- Bronze-Button with link -->
        <a href="http://lern-management-system.local/lernmodus-wissenstest-bronze/" class="btn btn-bronze">Bronze</a>

        <!-- Silver-Button with link -->
        <a href="http://lern-management-system.local/lernmodus-wissenstest-silber/" class="btn btn-silver">Silber</a>

        <!-- Gold-Button with link -->
        <a href="http://lern-management-system.local/lernmodus-wissenstest-gold/" class="btn btn-gold">Gold</a>
    </div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

ANHANG 4: QUELLCODE PRÜFUNGSMODUS FÜR DIE DREI STUFEN

```
<?php
/*
Header: Master Thesis - Learn Management System 2024
Plugin/Skript Name: Prüfungsmodus_php.php
HTML File: included
Description: This code is used for the exam mode - bronze level
Version: 18.5 - Many issues with shuffle function
Author: Beichler Christoph Franz
PHP Version: 8.3
DB Version: MySQL 8.0.16
WP Version: 6.6.2
Web Server: nginx
*/

ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

$host = 'localhost';
$db = 'questionnaire_wt2024';
$username = 'ChrisBeich';
$password = 'XXXXXXXXXX';
$level = 1;

try {
    $pdo = new PDO("mysql:host=$host;dbname=$db", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Datenbankverbindung fehlgeschlagen: " . $e->getMessage());
}

// Get user ID
try {
    $idUserStmt = $pdo->prepare("SELECT MAX(iduser) AS max_iduser FROM mappingtable_user");
    $idUserStmt->execute();
    $maxIdUser = $idUserStmt->fetch(PDO::FETCH_ASSOC)['max_iduser'];

    if ($maxIdUser !== null) {
        $updateLevelStmt = $pdo->prepare("UPDATE mappingtable_user SET jlevel = :level WHERE
idUser = :iduser");
        $updateLevelStmt->execute(['level' => $level, 'iduser' => $maxIdUser]);
    } else {
        echo "Kein Benutzer gefunden.";
    }
} catch (PDOException $e) {
    die("Fehler beim Setzen des Levels: " . $e->getMessage());
}

$station_num = $_POST['station_num'] ?? null;
$total_points = $_POST['total_points'] ?? 0;

if ($station_num === null) {
    $stmt = $pdo->prepare("SELECT DISTINCT station_number, station FROM mappingtable_wt WHERE
j_level = :level AND station IS NOT NULL AND station != ''");
    $stmt->execute(['level' => $level]);
    $stations = $stmt->fetchAll(PDO::FETCH_ASSOC);
    ?>

<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Prüfungsmodus - Wissenstest Bronze</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <style>
        .main-container {
            padding-top: 40px;
            padding-bottom: 40px;
            background-color: transparent;
        }
        .button-container {
            display: flex;
            flex-direction: column;
            align-items: center;
        }
    </style>

```

```

        gap: 10px;
        margin-top: 20px;
    }
    .btn-station {
        width: 300px;
        font-size: 1.2rem;
        background-color: #6c757d;
        color: white;
        border: none;
        border-radius: 5px;
    }
    .extra-buttons {
        display: flex;
        justify-content: center;
        gap: 20px;
        margin-top: 30px;
    }
    .extra-button {
        font-size: 1.1rem;
        padding: 10px 20px;
        border-radius: 5px;
        font-weight: bold;
    }
    .btn-wissenstest, .btn-lernmodus {
        width: 200px;
        font-size: 1.1rem;
        font-weight: bold;
        background-color: #e57373;
        color: white;
        text-align: center;
        display: flex;
        align-items: center;
        justify-content: center;
        height: auto;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        line-height: 1.2;
        white-space: normal;
    }
    .btn-station:hover {
        background-color: #fff;
        color: #000;
    }
    .btn-wissenstest:hover, .btn-lernmodus:hover {
        background-color: #fff;
    }
}
</style>
</head>
<body>
<div class="container main-container text-center">
    <h1>Wählen Sie eine Station für<br>den Wissenstest Bronze<br>(Prüfungsmodus):</h1>
    <form method="POST" action="">
        <div class="button-container">
            <?php foreach ($stations as $station): ?>
                <button type="submit" name="station_num" value="<?=
htmlspecialchars($station['station_number']) ?>" class="btn btn-secondary btn-station">
                    <?= htmlspecialchars($station['station']) ?>
                </button>
            <?php endforeach; ?>
        </div>
    </form>
</div>
</body>
</html>

<?php
} else {

    if (!isset($_POST['max_points'])) {
        $stmt = $pdo->prepare("SELECT SUM(points) AS max_points FROM mappingtable_wt WHERE
j_level = :level AND station_number = :station_num");
        $stmt->execute(['level' => $level, 'station_num' => $station_num]);
        $result = $stmt->fetch(PDO::FETCH_ASSOC);
        $max_points = (int)$result['max_points'];
    } else {
        $max_points = (int)$_POST['max_points'];
    }
}

```

```

$stmt = $pdo->prepare("SELECT question, answer, answer_no, points FROM mappingtable_wt WHERE
j_level = :level AND station_number = :station_num ORDER BY idwt");
$stmt->execute(['level' => $level, 'station_num' => $station_num]);
$allQuestions = $stmt->fetchAll(PDO::FETCH_ASSOC);

$questions = [];
foreach ($allQuestions as $row) {
    $question_text = $row['question'];
    if (!isset($questions[$question_text])) {
        $questions[$question_text] = [
            'question' => $row['question'],
            'points' => (int)$row['points'],
            'answers' => []
        ];
    }

    $answer_text = str_replace('/', '<br>', $row['answer']);
    $questions[$question_text]['answers'][] = [
        'text' => $answer_text,
        'correct' => $row['answer_no'] == 1
    ];
}

$stmt = $pdo->prepare("SELECT id, current_question_index, answer_order FROM user_progress
WHERE user_id = :user_id AND station_num = :station_num AND level = :level LIMIT 1");
$stmt->execute([
    'user_id' => $maxIdUser,
    'station_num' => $station_num,
    'level' => $level
]);
$progress = $stmt->fetch(PDO::FETCH_ASSOC);

if ($progress) {
    $currentQuestionIndex = $progress['current_question_index'];
    $shuffledOrder = json_decode($progress['answer_order'], true);

    if (isset($_POST['next_question']) && $_POST['next_question'] == "1") {
        $currentQuestionIndex += 1;

        if ($currentQuestionIndex < count($questions)) {
            $shuffledOrder =
array_keys($questions[array_keys($questions)[$currentQuestionIndex]]['answers']);
            shuffle($shuffledOrder);

            $updateStmt = $pdo->prepare("UPDATE user_progress SET current_question_index =
:question_index, answer_order = :answer_order WHERE id = :id");
            $updateStmt->execute([
                'question_index' => $currentQuestionIndex,
                'answer_order' => json_encode($shuffledOrder),
                'id' => $progress['id']
            ]);
        }
    }
} else {
    $currentQuestionIndex = 0;
    $shuffledOrder =
array_keys($questions[array_keys($questions)[$currentQuestionIndex]]['answers']);
    shuffle($shuffledOrder);

    $insertStmt = $pdo->prepare("INSERT INTO user_progress (user_id, station_num,
current_question_index, level, answer_order) VALUES (:user_id, :station_num, :question_index,
:level, :answer_order)");
    $insertStmt->execute([
        'user_id' => $maxIdUser,
        'station_num' => $station_num,
        'question_index' => $currentQuestionIndex,
        'level' => $level,
        'answer_order' => json_encode($shuffledOrder)
    ]);
}

$questionKeys = array_keys($questions);
$currentQuestionKey = $questionKeys[$currentQuestionIndex];
$currentQuestion = $questions[$currentQuestionKey];

$displayAnswers = array_map(fn($i) => $currentQuestion['answers'][$i], $shuffledOrder);

// Check if $selectedAnswerIndex is defined
$selectedAnswerIndex = isset($_POST['answer']) ? (int)$_POST['answer'] : null;

```

```

// Set std. value for $isCorrect
$isCorrect = false;

// Check answer
if ($selectedAnswerIndex !== null) {
    $isCorrect = $displayAnswers[$selectedAnswerIndex]['correct'];

    // Add points if answer is correct
    if ($isCorrect) {
        $total_points += $currentQuestion['points'];
    }
}

?>
<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Prüfungsmodus Frage</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <style>
        .question-container { margin-top: 50px; }
        .answer-button {
            width: 50%;
            font-size: 1.2rem;
            margin: 10px auto; padding: 10px;
            display: block;
        }
        .correct {
            background-color: green;
            color: white;
        }
        .incorrect {
            background-color: red;
            color: white;
        }
        .btn-next {
            margin-top: 20px;
            width: 150px;
            font-size: 1.2rem;
            border-radius: 5px;
        }
        .btn-back {
            margin-top: 20px;
            width: 200px;
            font-size: 1.2rem;
            border-radius: 5px;
        }
    </style>
</head>
<body>

<div class="container question-container text-center">
    <h2>Frage <?= $currentQuestionIndex + 1 ?>:</h2>
    <p><?= htmlspecialchars($currentQuestion['question']) ?></p>

    <form method="POST" action="">
        <?php foreach ($displayAnswers as $index => $answer): ?>
            <button type="submit" name="answer" value="<?= $index ?>" class="answer-button
<?= $selectedAnswerIndex !== null ? ($index === $selectedAnswerIndex ? ($isCorrect ? 'correct' :
'incorrect') : ($answer['correct'] ? 'correct' : '')) : '' ?>">
                <?= $answer['text'] ?>
            </button>
        <?php endforeach; ?>

        <input type="hidden" name="station_num" value="<?= htmlspecialchars($station_num)
?>">
        <input type="hidden" name="current_question_index" value="<?= $selectedAnswerIndex
!== null ? $currentQuestionIndex + 1 : $currentQuestionIndex ?>">
        <input type="hidden" name="total_points" value="<?= $total_points ?>">
        <input type="hidden" name="max_points" value="<?= $max_points ?>">

        <?php if ($selectedAnswerIndex !== null): ?>
            <p><?= $isCorrect ? "Richtig!" : "Falsch! Die richtige Antwort ist grün
markiert." ?></p>
        <?php endif; ?>
    </form>
</div>

```

```
        <?php if ($selectedAnswerIndex !== null && $currentQuestionIndex + 1 >=
count($questions)): ?>
            <a href="http://lern-management-system.local/ergebnisse-dieser-
station/?total_points=<?= $total_points ?>&max_points=<?= $max_points ?>" class="btn btn-
secondary btn-next">Weiter zur Ergebnisseite</a>
            <?php elseif ($selectedAnswerIndex !== null): ?>
                <button type="submit" name="next_question" value="1" class="btn btn-secondary
btn-next">Weiter</button>
            <?php endif; ?>
        </form>

        <a href="http://lern-management-system.local/pruefungsmodus-bronze/" class="btn btn-
secondary btn-back">Zurück zur Stationsübersicht</a>
    </div>
</body>
</html>
<?php
}
?>
```

ANHANG 5: QUELLCODE AUSWERTUNG JUGENDLICHE

```

<?php
/*
Header: Master Thesis - Learn Management System 2024
Plugin/Skript Name: resultJ_php.php
HTML File: included
Description: This code is used for the evaluation of the youth members
Version: 1.4
Author: Beichler Christoph Franz
PHP Version: 8.3
DB Version: MySQL 8.0.16
WP Version: 6.6.2
Web Server: nginx
*/

// Database connection
$host = 'localhost';
$db = 'questionnaire_wt2024';
$username = 'ChrisBeich';
$password = 'XXXXXXXXXX';

try {
    $pdo = new PDO("mysql:host=$host;dbname=$db", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Datenbankverbindung fehlgeschlagen: " . $e->getMessage());
}

// Vigenère-decryption
function decryptVigenere($ciphertext, $key) {
    $key = strtoupper($key);
    $keyLength = strlen($key);
    $plaintext = '';

    for ($i = 0, $j = 0; $i < strlen($ciphertext); $i++) {
        $c = strtoupper($ciphertext[$i]);
        if (ctype_alpha($c)) {
            $shift = ord($key[$j % $keyLength]) - ord('A');
            $decryptedChar = chr(((ord($c) - $shift + 26 - ord('A')) % 26) + ord('A'));
            $plaintext .= $decryptedChar;
            $j++;
        } else {
            $plaintext .= $c;
        }
    }

    return $plaintext;
}

// Get user information
try {
    $stmt = $pdo->prepare("
        SELECT name, codedsurname, result1, result2, result3, result4, result5, result6, jlevel
        FROM mappingtable_user
        WHERE iduser = (SELECT MAX(iduser)
        FROM mappingtable_user)
    ");
    $stmt->execute();
    $userData = $stmt->fetch(PDO::FETCH_ASSOC);
    $jlevel = $userData['jlevel'];
} catch (PDOException $e) {
    die("Fehler beim Abrufen der Daten: " . $e->getMessage());
}

// Decode surname
$encryptedSurname = $userData['codedsurname'];
$decryptedSurname = decryptVigenere($encryptedSurname, 'FLORIANI');

// Get stations
$stationNames = [];
if ($jlevel !== null) {
    try {
        $stationStmt = $pdo->prepare("
            SELECT DISTINCT station
            FROM mappingtable_wt
            WHERE j_level = :jlevel LIMIT 6
        ");
        $stationStmt->execute(['jlevel' => $jlevel]);
    }
}

```

```

        $stationNames = $stationStmt->fetchAll(PDO::FETCH_COLUMN);
    } catch (PDOException $e) {
        die("Fehler beim Abrufen der Stationsnamen: " . $e->getMessage());
    }
}
// Get total points
$totalPoints = 0;
if ($userData) {
    $totalPoints = array_sum(array_slice($userData, 2, 6));
}
?>

<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Ergebnisse</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">
    <style>
        .container {
            max-width: 600px;
            margin-top: 50px;
            padding: 20px;
            background-color: #f9f9f9;
            border-radius: 8px;
            box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
            text-align: center;
        }
        .result-title {
            font-weight: bold;
            font-size: 1.5rem;
            color: #333;
            margin-bottom: 20px;
        }
        .result-item {
            font-size: 1.1rem;
            color: #555;
            display: flex;
            justify-content: space-between;
            padding: 8px 0;
            border-bottom: 1px solid #e0e0e0;
        }
        .result-item strong {
            color: #333;
            font-weight: 600;
            flex: 1;
            text-align: left;
        }
        .result-value {
            color: #333;
            font-weight: 600;
            flex: 0;
            text-align: right;
        }
        .total-points {
            font-size: 1.2rem;
            font-weight: bold;
            margin-top: 20px;
            color: #007bff;
        }
        .btn-container {
            display: flex;
            justify-content: space-between;
            margin-top: 30px;
        }
        .btn-container .btn {
            width: 48%;
            font-size: 1rem;
            padding: 10px 0;
        }
        .success-message {
            color: #28a745;
            text-align: center;
            margin-top: 20px;
            font-weight: bold;
        }
        .info-section {
            margin-top: 15px;

```



```

        margin-bottom: 20px;
        text-align: left;
    }
</style>
</head>
<body>
<div class="container">
    <h2 class="result-title">Ergebnisse</h2>
    <?php if ($userData): ?>
        <div class="info-section">
            <p class="result-item"><strong>Vorname:</strong> <span class="result-value"><?=
htmlspecialchars($userData['name']) ?></span></p>
            <p class="result-item"><strong>Nachname:</strong> <span class="result-value"><?=
htmlspecialchars($decryptedSurname) ?></span></p>
        </div>

        <div style="text-align: left;">
            <?php for ($i = 1; $i <= 6; $i++): ?>
                <p class="result-item">
                    <strong><?= htmlspecialchars($stationNames[$i - 1] ?? "Station $i")
?></strong>
                    <span class="result-value"><?= htmlspecialchars($userData["result$i"] ?? '0')
?></span>
                </p>
            <?php endfor; ?>
        </div>

        <p class="total-points">Gesamtpunkte: <?= $totalPoints ?></p>
    <?php else: ?>
        <p>Keine Daten gefunden.</p>
    <?php endif; ?>

    <form method="POST" action="">
        <div class="btn-container">
            <a href="http://lern-management-system.local/pruefungsmodus-bronze/" class="btn btn-
secondary">Zurück zur Stationsübersicht</a>
            <button type="submit" name="save_total" class="btn btn-primary">Gesamtpunkte
speichern</button>
        </div>
    </form>

    <?php
    // Save total points
    if (isset($_POST['save_total'])) {
        try {
            $idUserStmt = $pdo->prepare("
                SELECT MAX(iduser)
                AS max_iduser
                FROM mappingtable_user
                ");
            $idUserStmt->execute();
            $maxIdUser = $idUserStmt->fetch(PDO::FETCH_ASSOC) ['max_iduser'];

            if ($maxIdUser !== null) {
                $pdo->beginTransaction();
                $pdo->prepare("
                    UPDATE mappingtable_user
                    SET lowpoint5 = lowpoint4, lowpoint4 = lowpoint3, lowpoint3 = lowpoint2,
lowpoint2 = lowpoint1
                    WHERE iduser = :iduser
                    ");
                ->execute(['iduser' => $maxIdUser]);

                $saveStmt = $pdo->prepare("
                    UPDATE mappingtable_user
                    SET lowpoint1 = :total_points
                    WHERE iduser = :iduser
                    ");
                $saveStmt->execute(['total_points' => $totalPoints, 'iduser' => $maxIdUser]);

                $pdo->commit();
                echo "<p class='success-message'>Gesamtpunkte wurden erfolgreich
gespeichert.</p>";
            } else {
                echo "<p class='text-danger'>Kein Benutzer gefunden.</p>";
            }
        } catch (PDOException $e) {
            $pdo->rollback();
            die("Fehler beim Speichern der Gesamtpunkte: " . $e->getMessage());
        }
    }

```

```
    }  
  }  
  ?>  
</div>  
</body>  
</html>
```

ANHANG 6: QUELLCODE AUSWERTUNG BEAUFTRAGTE UND FUNKTIONÄRE*INNEN

```

<?php
/*
Header: Master Thesis - Learn Management System 2024
Plugin/Skript Name: ErgebnisEinzel_php.php
HTML File: included
Description: This code is used to get evaluation of dedicated youth members
Version: 1.2
Author: Beichler Christoph Franz
PHP Version: 8.3
DB Version: MySQL 8.0.16
WP Version: 6.6.2
Web Server: nginx
*/

// Standard-setup for error monitoring
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

// Database connection
$host = 'localhost';
$db = 'questionnaire_wt2024';
$username = 'ChrisBeich';
$password = 'XXXXXXXXXX';
try {
    $pdo = new PDO("mysql:host=$host;dbname=$db", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // Get all entries via sql-request with firebrigade = 'Tillmitsch'
    $stmt = $pdo->prepare("SELECT iduser, name, codedsurname FROM mappingtable_user WHERE
firebrigade = :firebrigade");
    $stmt->execute(['firebrigade' => 'Tillmitsch']);
    $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
} catch (PDOException $e) {
    die("Fehler bei der Datenbankverbindung: " . $e->getMessage());
}

// Vigenère-decryption
function decryptVigenere($ciphertext, $key) {
    $key = strtoupper($key);
    $keyLength = strlen($key);
    $plaintext = '';
    for ($i = 0, $j = 0; $i < strlen($ciphertext); $i++) {
        $c = strtoupper($ciphertext[$i]);
        if (ctype_alpha($c)) {
            $shift = ord($key[$j % $keyLength]) - ord('A');
            $decryptedChar = chr(((ord($c) - $shift + 26 - ord('A')) % 26) + ord('A'));
            $plaintext .= $decryptedChar;
            $j++;
        } else {
            $plaintext .= $c;
        }
    }
    return $plaintext;
}

// Get user results if request got sent
if (isset($_GET['id'])) {
    $userId = $_GET['id'];
    $stmt = $pdo->prepare("SELECT name, codedsurname, result1, result2, result3, result4,
result5, result6, jlevel FROM mappingtable_user WHERE iduser = :iduser");
    $stmt->execute(['iduser' => $userId]);
    $userData = $stmt->fetch(PDO::FETCH_ASSOC);

    // Decrypt codedsurname -> surname
    $decryptedCodedsurname = decryptVigenere($userData['codedsurname'], 'FLORIANI');

    // Get station name and max. points
    $stationStmt = $pdo->prepare("
    SELECT station, SUM(points) AS max_points
    FROM mappingtable_wt
    WHERE j_level = :jlevel
    GROUP BY station

```

```

ORDER BY MIN(station_number)
LIMIT 6
");
$stmt->execute(['jlevel' => $userData['jlevel']]);
$stationsData = $stmt->fetchAll(PDO::FETCH_ASSOC);

// Get total points and assemble the output
$totalPoints = array_sum(array_slice($userData, 3, 6));
$stations = [];
foreach ($stationsData as $index => $station) {
    $stations[] = [
        'name' => $station['station'],
        'points' => $userData["result" . ($index + 1)],
        'max_points' => $station['max_points']
    ];
}

// Sent JSON-answer
echo json_encode([
    'vorname' => $userData['name'],
    'codedsurname' => $decryptedCodedsurname,
    'stations' => $stations,
    'total_points' => $totalPoints
]);
exit; // End script so only JSON-answer got shown
}
?>

<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Ergebnisse anzeigen</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">
    <style>
        /* Formating of dropdown results */
        .dropdown-container {
            display: flex;
            justify-content: center;
            align-items: center;
            height: 10vh;
        }
        .form-content {
            text-align: center;
        }
        .container {
            max-width: 600px;
            margin: 20px auto;
            padding: 20px;
            background-color: #f9f9f9;
            border-radius: 8px;
            box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
            text-align: center;
            display: none;
        }
        .result-title {
            font-weight: bold;
            font-size: 1.5rem;
            color: #333;
            margin-bottom: 20px;
        }
        .result-item {
            font-size: 1.1rem;
            color: #555;
            display: flex;
            justify-content: space-between;
            padding: 8px 0;
            border-bottom: 1px solid #e0e0e0;
        }
        .result-item strong {
            color: #333;
            font-weight: 600;
            flex: 1;
            text-align: left;
        }
        .result-value {
            color: #333;
            font-weight: 600;

```

```

        text-align: right;
    }
    .points-info {
        color: #007bff;
        font-size: 1rem;
    }
    .total-points {
        font-size: 1.3rem;
        font-weight: bold;
        margin-top: 20px;
        color: #007bff;
    }
}
</style>
</head>
<body>

<div class="dropdown-container">
    <div class="form-content">
        <form>
            <label for="firebrigade-dropdown">Wähle einen Eintrag:</label>
            <select id="firebrigade-dropdown" name="firebrigade" onchange="fetchResults()">
                <option value="" selected disabled>Bitte wählen...</option> <!-- First line in
dropdown -->
                <!-- PHP: Dynamic generation of dropdown-options -->
                <?php if (!empty($results)): ?>
                    <?php foreach ($results as $row):
                        $decryptedCodedsurname = decryptVigenere($row['codedsurname'],
'FLORIANI'); // Decrypt surname ?>
                        <option value="<?php echo htmlspecialchars($row['iduser']); ?>">
                            <?php echo htmlspecialchars($row['name'] . ' ' .
$decryptedCodedsurname); ?>
                        </option>
                    <?php endforeach; ?>
                <?php else: ?>
                    <option value="">Keine Einträge gefunden</option>
                <?php endif; ?>
            </select>
        </form>
    </div>
</div>

<div class="container" id="result-container">
    <h2 class="result-title">Ergebnisse</h2>
    <div id="result-content"></div>
    <p class="total-points" id="total-points"></p>
</div>

<script>
    function fetchResults() {
        const userId = document.getElementById('firebrigade-dropdown').value;
        if (userId) {
            fetch(`?id=${userId}`)
                .then(response => response.json())
                .then(data => {
                    // Sent results
                    let resultContent = `
                        <p class="result-item"><strong>Vorname:</strong> <span class="result-
value">${data.vorname}</span></p>
                        <p class="result-item"><strong>Nachname:</strong> <span class="result-
value">${data.codedsurname}</span></p>
                    `;

                    // Show station name + total points + max. points
                    data.stations.forEach((station, index) => {
                        resultContent += `
                            <p class="result-item">
                                <strong>${station.name}</strong>
                                <span class="result-value">${station.points}</span> Pkt. <span
class="points-info">von ${station.max_points}</span> Pkt.</span></p>
                        `;
                    });

                    document.getElementById('result-content').innerHTML = resultContent;
                    document.getElementById('total-points').innerText = `Gesamtpunkte:
${data.total_points}`;
                    document.getElementById('result-container').style.display = 'block';
                })
                .catch(error => console.error('Fehler beim Abrufen der Daten:', error));
        }
    }

```

```
    } else {
        document.getElementById('result-container').style.display = 'none';
    }
}
</script>

</body>
</html>

<!-- |#|#|#|#|#| Additional PHP code for all members |#|#|#|#|#| -->

<?php
/*
Header: Master Thesis - Learn Management System 2024
Plugin/Skript Name: ErgebnisAlle_php.php
HTML File: included
Description: This code is used to get evaluation of all youth members
Version: 1.5
Author: Beichler Christoph Franz
PHP Version: 8.3
DB Version: MySQL 8.0.16
WP Version: 6.6.2
Web Server: nginx
*/

// Standard-setup for error monitoring
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

// Database connection
$host = 'localhost';
$db = 'questionnaire_wt2024';
$username = 'ChrisBeich';
$password = 'XXXXXXXXXX';

try {
    $pdo = new PDO("mysql:host=$host;dbname=$db", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // Define name of jlevel - number into string
    function getLevelName($jLevel) {
        switch ($jLevel) {
            case 1: return "Bronze";
            case 2: return "Silber";
            case 3: return "Gold";
            default: return "Unbekannt";
        }
    }

    // Vigenère-decryption
    function decryptVigenere($ciphertext, $key) {
        $key = strtoupper($key);
        $keyLength = strlen($key);
        $plaintext = '';

        for ($i = 0, $j = 0; $i < strlen($ciphertext); $i++) {
            $c = strtoupper($ciphertext[$i]);
            if (ctype_alpha($c)) {
                $shift = ord($key[$j % $keyLength]) - ord('A');
                $decryptedChar = chr(((ord($c) - $shift + 26 - ord('A')) % 26) + ord('A'));
                $plaintext .= $decryptedChar;
                $j++;
            } else {
                $plaintext .= $c;
            }
        }
        return $plaintext;
    }

    // jLevel of users with firebrigade = 'Tillmitsch' and valid jLevel (1, 2 or 3)
    $stmt = $pdo->prepare("
        SELECT jlevel,
            AVG(result1) AS avg_result1,
            AVG(result2) AS avg_result2,
            AVG(result3) AS avg_result3,
            AVG(result4) AS avg_result4,
            AVG(result5) AS avg_result5,
            AVG(result6) AS avg_result6
    ");
}
```

```

        FROM mappingtable_user
        WHERE firebrigade = :firebrigade AND jlevel IS NOT NULL AND jlevel IN (1, 2, 3)
        GROUP BY jlevel
    ");
    $stmt->execute(['firebrigade' => 'Tillmitsch']);
    $averageResults = $stmt->fetchAll(PDO::FETCH_ASSOC);

    // Array to store the results
    $allResults = [];

    foreach ($averageResults as $avgResult) {
        $jLevel = $avgResult['jlevel'];

        // Get station name and max. points for current jLevel
        $stationStmt = $pdo->prepare("
            SELECT station, SUM(points) AS max_points
            FROM mappingtable_wt
            WHERE j_level = :jlevel
            GROUP BY station
            ORDER BY MIN(station_number)
            LIMIT 6
        ");
        $stationStmt->execute(['jlevel' => $jLevel]);
        $stationsData = $stationStmt->fetchAll(PDO::FETCH_ASSOC);

        // Store result for jLevel
        $result = [
            'jLevel' => getLevelName($jLevel),
            'stations' => []
        ];

        foreach ($stationsData as $index => $station) {
            $stationName = $station['station'];
            $maxPoints = $station['max_points'];
            $avgPoints = isset($avgResult["avg_result" . ($index + 1)]) ?
round($avgResult["avg_result" . ($index + 1)], 2) : 0;

            $result['stations'][] = [
                'name' => $stationName,
                'avg_points' => $avgPoints,
                'max_points' => $maxPoints
            ];
        }

        $allResults[] = $result;
    }

} catch (PDOException $e) {
    die("Fehler bei der Datenbankverbindung: " . $e->getMessage());
}
?>

<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Durchschnittliche Ergebnisse</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet">
    <style>
        .container {
            max-width: 800px;
            margin: 20px auto;
            padding: 20px;
            background-color: #f9f9f9;
            border-radius: 8px;
            box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
            text-align: center;
        }
        .result-title {
            font-weight: bold;
            font-size: 1.5rem;
            color: #333;
            margin-bottom: 20px;
        }
        .result-table {
            width: 100%;
            margin-bottom: 20px;
        }
    </style>

```

```

        .result-table th, .result-table td {
            padding: 10px;
            text-align: left;
            border-bottom: 1px solid #ddd;
        }
        .level-title {
            font-size: 1.3rem;
            font-weight: bold;
            margin-top: 20px;
            color: #555;
        }
    }
</style>
</head>
<body>

<div class="container">
    <h2 class="result-title">Durchschnittliche Ergebnisse für<br>Feuerwehr Tillmitsch</h2>

    <?php if (!empty($allResults)): ?>
        <?php foreach ($allResults as $result): ?>
            <h3 class="level-title"><?= htmlspecialchars($result['jLevel']) ?></h3>
            <table class="result-table">
                <thead>
                    <tr>
                        <th>Station</th>
                        <th>Durchschnittliche Punkte</th>
                        <th>Maximale Punkte</th>
                    </tr>
                </thead>
                <tbody>
                    <?php foreach ($result['stations'] as $station): ?>
                        <tr>
                            <td><?= htmlspecialchars($station['name']) ?></td>
                            <td><?= htmlspecialchars($station['avg_points']) ?> Pkt.</td>
                            <td><?= htmlspecialchars($station['max_points']) ?> Pkt.</td>
                        </tr>
                    <?php endforeach; ?>
                </tbody>
            </table>
        <?php endforeach; ?>
    <?php else: ?>
        <p>Keine gültigen Einträge gefunden.</p>
    <?php endif; ?>
</div>

</body>
</html>

```