

# **Modul 151 | Projekt**

## **Technische Dokumentation für eine Trainingsplan-Webseite und Installationsanleitung**



Vorgelegt von:

**Marc Schmidlin und Flavio Schaffer**

Projektleiter:

Marc Schmidlin und Flavio Schaffer

Pratteln, März 23.03.2023

## Management Summary

### **Wo stehen wir heute (IST Situation, Projekt-Beweggrund)**

Zu Beginn des Projektes (Ist Zustand), gibt es keine einfache Möglichkeit einen Trainingsplan für sich zu erstellen und auch das Gewicht zu tracken. Deswegen werden wir eine einfache und Flexible Lösung dafür suchen und diese auf einer Webseite festlegen.

### **Was ändert das neue Produkt/Projekt innerhalb des betroffenen Bereichs**

Das neue Projekt bietet Benutzern die Möglichkeit auf eine einfache Art und Weise einen Trainingsplan zu erstellen, wobei er dem Trainingsplan ein Name, Übungen und den Zielmuskel angeben kann. Auch kann der Benutzer durch Eingabe des Gewichts sein Gewicht einfach verfolgen und weiss immer, ob es nach oben oder nach unten geht.

### **Wie lange dauert die Änderung/Implementierung (Termin)**

Die Implementierung für das Tool, ist nach der Vereinbarung sehr schnell erledigt und es kann direkt damit begonnen werden. Dieses Dokument wird eine Kurzanleitung enthalten, wo bei der Installation und Einrichtung behilflich sein wird.

### **Wie viel wird das Projekt im Total kosten**

Die Kosten für das neue Projekt der Firma Modul 151 AG wird sich auf 0 Franken halten, da dies ein Schulmodul ist und nur für Testzwecke durchgeführt wird.

## Dokumentmanagement

Version/Status: Siehe unten

Datum: 22.12.2022

Autoren: Marc Schmidlin | Flavio Schaffer

Dateiname:

## Änderungsgeschichte

Vers.	Datum	Autoren	Status	Änderung
0.1	16.03.2023	Flavio Schaffer	Initial	Formatvorlagen eingesetzt, Gliederung aktiviert, Deckblatt eingefügt, Layout angepasst, Logo in Kopfzeile ergänzt und Fusszeile eingefügt.
0.2	16.03.2023	Flavio Schaffer	Abgeschlossen	Inhaltsverzeichnis eingefügt und formatiert, Alle Überschriften erstellt und Dokument zur Bearbeitung fertiggestellt und Kapitel Einleitung abgeschlossen.
03	30.03.2023	Flavio Schaffer	In Progress	Kompetenzen hinzugefügt und mit dem Dokumentieren begonnen
04	15.03.2023	Flavio Schaffer	In Progress	Installationsanleitung begonnen zu schreiben und Informationen mit Bildern hinzugefügt

# Inhaltverzeichnis

<b>Einleitung .....</b>	<b>5</b>
Kurzeinführung Projekt.....	5
Dokumentation Inhalt .....	5
<b>Installationsanleitung .....</b>	<b>6</b>
Einleitung .....	6
Datenbank .....	6
Projekt .....	6
<b>Technische Dokumentation .....</b>	<b>7</b>
Einleitung .....	7
Kompetenz B1 und B2.....	7
Kompetenz C1 .....	7
Kompetenz C2.....	8
Kompetenz C3.....	9
Kompetenz C4.....	10
Kompetenz C5.....	11
Kompetenz C6.....	11
Kompetenz C7.....	12
Kompetenz C8.....	13
Kompetenz C9.....	13
Kompetenz C10.....	14
Kompetenz C11 .....	15
Kompetenz C12.....	15
Kompetenz C13.....	16
Kompetenz C14.....	17
Kompetenz C15.....	18
Kompetenz C16.....	18
<b>Lernprozesse .....</b>	<b>19</b>
Flavio Schaffer .....	19
Marc Schmidlin .....	20
<b>Reflexion .....</b>	<b>21</b>
Flavio Schaffer .....	21
Marc Schmidlin .....	21
<b>Geschätzte Note .....</b>	<b>22</b>
<b>Selbstständigkeitserklärung.....</b>	<b>23</b>
1.1   Verwendung von Quellen und Sekundärliteratur .....	23

1.2	Sanktionen .....	23
-----	------------------	----

## Einleitung

### Kurzeinführung Projekt

Wir haben eine Website entwickelt, die es dem Nutzer ermöglicht, sich zu registrieren und anzumelden, um auf weitere Funktionen zugreifen zu können. Nach erfolgreicher Anmeldung kann der eingeloggte Benutzer sein Gewicht eintragen und dessen Veränderungen im Laufe der Zeit verfolgen. Darüber hinaus hat der Nutzer die Möglichkeit, einen individuellen Trainingsplan mit drei Übungen zu erstellen und sich diesen anzusehen. Für den Fall, dass Änderungen an den Anmeldedaten vorgenommen werden müssen, besteht die Option, den Benutzernamen und das Passwort zu bearbeiten.

### Dokumentation Inhalt

Dieses Dokument beinhaltet die Installationsanleitung und die Technische Dokumentation. Die Installationsanleitung, zeigt wie und mit welchen schritten das Projekt installiert werden kann, sodass es für den gebrauch parat ist. Die Technische Dokumentation zeigt alle Kompetenzen, welches für das Projekt zugeordnet sind und anhand von verschiedenen Dingen wie sie in das Projekt einbezogen wurden. .

## Installationsanleitung

### Einleitung

Dieser Dokumententeil beinhalten die Installationsanleitung und zeigt auf, wie unser Projekt installiert werden kann. Dazu gehören alle Files wie html, css, jpegs aber auch die Datenbank. Dabei wird beschreiben, wie das Projekt angelegt werden muss, damit es funktionsbereit ist.

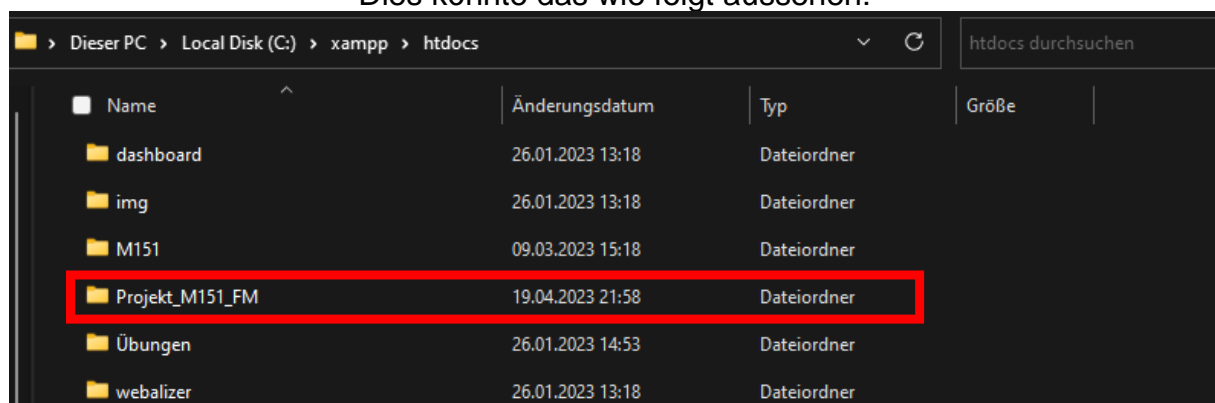
### Vorbereitungsschritte:

- Laden Sie das vorgegebene ZIP-File herunter (Projekt\_m151\_FM.zip)
- Öffnen Sie XAMPP und aktivieren Sie der Apache und MySQL dienst.

### Projekt

Das Projekt sollte, bevor es verwendet, wird richtig abgespeichert werden. Hierzu nehmen sie das erhaltene Zip und verschieben es in den Ordner: **xampp\htdocs** Entzippen Sie nun den kompletten Ordner. Nun sollten Sie bereits den richtigen Ordner in der Ordnerstruktur von htdocs sehen.

Dies könnte das wie folgt aussehen:



### Datenbank

Um die Datenbank anzulegen, wurde in dem abgegebenen ZIP ein Ordner Namens Projektdokumentation abgegeben. In diesem Ordner befindet sich eine Datei Namens «VorlageDatenbankProjekt» welche ein Export der Datenbank ist.

Diese Datenbank muss nun auf phpMyAdmin hinzugefügt werden.

Öffnen die dazu Ihren Browser und gehen sie auf Localhost. Rufen sie von da auf die Webseite phpMyAdmin auf (Verlinkung im Header).

Um nun die Datenbank hinzuzufügen, klicken Sie auf «Importieren» im Header:



Danach können sie das mitgegebene File VorlageDatenbankProjekt.sql auswählen und mit Ok bestätigen. Dies wird die Komplette Datenbank mit allen Tabellen, einem Testbenutzer und Testdaten hinzufügen.

## Technische Dokumentation

### Einleitung

Dieser Dokumentpart übernimmt die Technische Dokumentation. Hier werden auf die verschiedenen Kompetenzen des Projektes eingegangen und beschrieben

#### Kompetenz B1 und B2

Die beiden Kompetenzen B1 und B2 werden mit diesem Dokument abgelegt. B1 wird erfüllt durch das Erklären von den relevanten Funktionen und B2 durch die Installationsanleitung zu Beginn des Dokumentes. Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*B1: Ich kann den Projektantrag in eine Projekt-Dokumentation überführen.  
In der Projekt-Dokumentation werden die für dieses Modul relevanten Funktionen der Projektarbeit dokumentiert und erklärt.*

*B2: Ich erstelle für mein Projekt eine Installationsanleitung, welche die Inbetriebnahme detailliert erklärt*

#### Kompetenz C1

Wir haben uns verschiedene sinnvolle Testpläne überlegt, erstellt und durchgeführt. Diese wurden in einem extra Dokument festgehalten und bei Projektantrag mit abgegeben. Siehe hierzu das Dokument «Testprotokoll». Durch das Durchführen der Tests konnte erkannt werden, ob Fehler vorhanden sind. Die Testfälle sind alle durchgeführt worden und es gab kein Problem.

Für das Erstellen der verschiedenen Testfälle, haben wir überlegt, welche Funktionen für den Benutzer der Applikation wichtig sind und wie diese getestet werden können. Danach haben wir diese im Testprotokoll Dokument festgehalten.

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*C1: Ich kann einen Testplan für mein Projekt erstellen und überprüfe die für dieses Modul relevanten Funktionen meines Projektes anhand dieses Testplanes.*

#### Inhaltsverzeichnis

1	Testfall Registrierung .....
2	Testdurchführung und Testergebnis .....
3	Testfall Anmelden .....
4	Testdurchführung und Testergebnis .....
5	Testfall Passwort ändern .....
6	Testdurchführung und Testergebnis .....
7	Testfall Daten erfassen ändern .....
8	Testdurchführung und Testergebnis .....
9	Testfall Daten ändern von Benutzer .....
10	Testdurchführung und Testergebnis .....
11	Testfall Daten löschen .....
12	Testdurchführung und Testergebnis .....
13	Testfall SQL Injection .....
14	Testdurchführung und Testergebnis .....



### Kompetenz C2

Wir haben den Code mit W3C validiert und Fehler, die es gab verbessert. Nach der Verbesserung aller Validierungsfehler ist der Coder nun fehlerfrei und kann ohne Probleme verwendet werden. Hier ein Beispiel der Meldung nach Fehlerfreien Validierung

5555555555555555

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*C2: Ich kenne Tools zur Validierung von HTML und CSS und validiere den Code mit diesen Tools. Der generierte HTML-Code ist fehlerfrei.*

### Kompetenz C3

In unserem Projekt haben wir eine umfassende Validierung aller Benutzereingabe-Formulare auf der Client-Seite durchgeführt. Hierfür haben wir verschiedene Funktionen genutzt, darunter Input Types wie Text, Number, Email und Date sowie Attribute wie Required und Maxlength. Ebenfalls wurden für ein Passwort Feld Pattern verwendet Dies wurde durch alle Formula durchgeführt. In unserem Code sieht das dann wie folgt aus (Das Beispiel stammt aus dem signun.php file, welches zum einloggen verwendet wird).

```
<label for="firstname">Vorname *</label>
  <input type="text" name="firstname" class="form-control" id="firstname"
value="<?php echo $firstname ?>" placeholder="Geben Sie Ihren Vornamen an." maxlength="30"
required="true">
</div>
<!-- nachname -->
<div class="form-group">
  <label for="lastname">Nachname *</label>
  <input type="text" name="lastname" class="form-control" id="lastname" value="<?php
echo $lastname ?>" placeholder="Geben Sie Ihren Nachnamen an" maxlength="30"
required="true">
</div>
<!-- email -->
<div class="form-group">
  <label for="email">Email *</label>
  <input type="email" name="email" class="form-control" id="email" value="<?php echo
$email ?>" placeholder="Geben Sie Ihre Email-Adresse an." maxlength="100" required="true">
</div>
<!-- benutzername -->

<!-- password -->
<div class="form-group">
  <label for="password">Password *</label>
  <input type="password" name="password" class="form-control" id="password"
placeholder="Gross- und Kleinbuchstaben, Zahlen, Sonderzeichen, min. 8 Zeichen, keine
Umlaute" pattern="(?=^.{8,}$)((?=.*\d+)(?=.*\W+))(![.\n])(?=.*[A-Z])(?=.*[a-z]).*$"
title="mindestens einen Gross-, einen Kleinbuchstaben, eine Zahl und ein Sonderzeichen,
mindestens 8 Zeichen lang,keine Umlaute." maxlength="255" required="true">
</div>
```

Die clientseitige Validierung ist wichtig, um sicherzustellen, dass Benutzereingaben korrekt und im erwarteten Format eingegeben werden. Dadurch können Fehler und Probleme vermieden werden, bevor Daten an den Server gesendet werden, was Zeit spart, und die Benutzerfreundlichkeit verbessert. Dies hilft uns bei der Datenweitergabe und ermöglicht uns Fehler im Voraus herauszufiltern.

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*In meinem Projekt werden alle Benutzereingaben clientseitig validiert.*

## Kompetenz C4

Wir haben in unserem Projekt alle eingaben serverseitig validiert. Dies gibt uns Sicherheit über die richtige eingaben der Daten. Für die Validierung haben wir `isset()`, `empty()`, `trim()` und `strlen()` verwenden. Die Validierung auf der Serverseite ist besonders wichtig, da dies der letzte schritt ist, bevor die Daten in der Datenbank verarbeitet werden. Die Serverseitige Validierung wurde durch alle Formulare durchgezogen. In unserem Code sieht das wie folgt aus (Beispiel aus `signup.php` file, welches vor dem Login E-Mail und Passwort überprüft):

```
// Email ausgefüllt?
if (isset($_POST['email'])) {
    //trim and sanitize
    $email = htmlspecialchars(trim($_POST['email']));

    //mindestens 1 Zeichen und maximal 100 Zeichen lang, gültige Emailadresse
    if (empty($email) || strlen($email) > 100 || filter_var($email, FILTER_VALIDATE_EMAIL)
    === false) {
        $error .= "Geben Sie bitte eine korrekten Emailadresse ein.<br />";
    }
} else {
    $error .= "Geben Sie bitte eine Emailadresse ein.<br />";
}

// Passwort ausgefüllt
if (isset($_POST['password'])) {
    //trim and sanitize
    $password = trim($_POST['password']);

    //mindestens 1 Zeichen , entsprich RegEX
    if (empty($password) ||
    !preg_match("/(?!.*\d+)(?!.*\W+)(?![\.\n])(?!.*[A-Z])(?!.*[a-z]).*$/",
    $password)) {
        $error .= "Geben Sie bitte ein korrektes Passwort ein.<br />";
    }
} else {
    $error .= "Geben Sie bitte ein Passwort ein.<br />";
}
```

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):  
*In meinem Projekt werden alle Benutzereingaben serverseitig validiert.*

## Kompetenz C5

Script Injection wurde durch das verwenden der funktion htmlspecialchars() in unserem gesamten Projekt verhindert. Somit werden sicherheitslücken in dem System verhindert und es können keine Scripts bei der Eingabe mitgegeben werden. In einem Beispiel von dem File signup.php sieht das ganze wie folgt aus:

```
// Vorname ausgefüllt?  
if (isset($_POST['firstname'])) {  
    //trim and sanitize  
    $firstname = htmlspecialchars(trim($_POST['firstname']));  
}
```

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):  
*Script-Injection wird in meinem Projekt konsequent verhindert.*

## Kompetenz C6

**Funktioniert noch nicht**

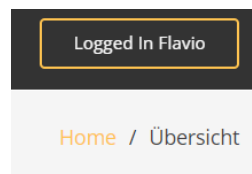
Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):  
*In meinem Projekt wird das Session-Handling korrekt eingesetzt. Ein angemeldeter Benutzer hat Zugriff auf weitere Funktionen, welche nicht angemeldeten Benutzern verwehrt bleiben.*

### Kompetenz C7

In unserem Projekt wurde ein Button hinzugefügt, wo sich ein User abmelden kann. Dabei wird die Session beendet und der User wird auf die Startseite weitergeleitet. Die Session wird mit der funktion `session_destroy()` beendet. Dies sieht in unserem Code wie folgt aus (Der Code stammt aus dem Fiel `logout.php`):

```
<?php
// TODO - Session starten
session_start();
// TODO - Session leeren
$_SESSION = array();
session_destroy();
header('Location: ./index.php');
?>
```

Der Benutzer kann mit folgendem Button sich ausloggen (Dabei wird der obenstehende Code durchgeführt). Die Funktion wird freigeschaltet, sobald man eingeloggt ist.



Hierbei kann man auf Logged In «Username» auf der oberen Seite klicken. Von da aus sieht dann das Fenster wie folgt aus:



Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*Ein angemeldeter Benutzer kann sich wieder abmelden. Die Session wird dabei korrekt beendet.*

### Kompetenz C8

In unserer Webseite haben wir Session-Fixation und Session-Hijacking mit dem folgenden code erschwert:

```
session_regenerate_id(true);
```

Der Code "session\_regenerate\_id(true);" erzeugt eine neue Session-ID für den aktuellen Benutzer und löscht gleichzeitig die alte ID. Dies erschwert das Hijacking.

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*In meinem Projekt wird Session-Fixation und Session-Hijacking erschwert.*

### Kompetenz C9

In unserem Projekt haben wir die Sensiblen Daten (Passwörter) mit einem Hash versehen und so abgespeichert. Dies verhindert das Passwörter mit Text in der Datenbank stehen. Es wird nur der Hash-Code abgespeichert und so kann ein Passwort nicht entschlüsselt werden. Diese Funktion verwenden wir in unserem code in dem file signup.php wo das Passwort als Hast in die Datenbank geschrieben wird:

```
$password_hash = password_hash($password, PASSWORD_DEFAULT);
```

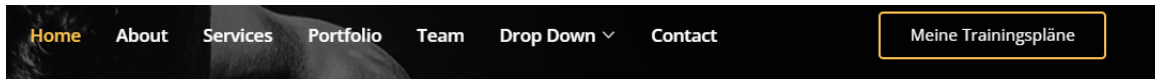
Hierfür verwenden wird die funktion password\_hash(). Diese wird von PHP zur Verfügung gestellt und ist ziemlich sicher, da noch salt hinzugefügt wird.

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*C9: In meinem Projekt werden sensible Daten wie das Passwort mit sicheren und aktuellen Methoden gehasht und gestaltet.*

### Kompetenz C10

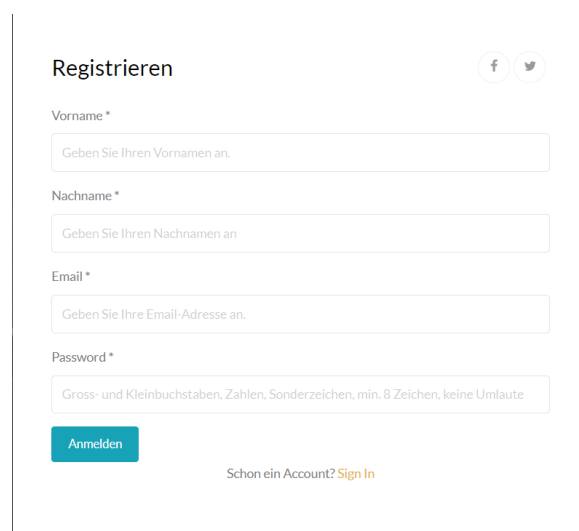
Ein Benutzer kann sich auf unserer Webseite registrieren. Damit ein User sich registrieren kann, sollte dieser die Startseite öffnen und auf «Meine Trainingspläne» klicken.



Danach wird er zu Login Seite weitergeleitet. Um sich nun zu registrieren kann Sign Up ausgewählt werden.

Noch kein Account? [Sign Up](#)

Der User wird dann zur Registrierungsseite weitergeleitet. Diese ist wie folgt aufgebaut:



Es werden Vorname Nachname, E-Mail und Passwort von dem Benutzer aufgenommen. Diese sind für unsere Anwendung Sinnvoll. Für das Login ist dies ebenfalls praktisch, da die E-Mail und Passwort verwendet werden kann. Mit dem Anmelden Button werden die Daten in die Datenbank geschrieben und der Benutzer ist erfolgreich registriert.

Der Code für das Registrieren wurde im file signup.php geschrieben.

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*C10: Ein Benutzer kann sich an meinem Projekt registrieren. Dazu erfasse ich sinnvolle Daten des Benutzers.*

### Kompetenz C11

Nach dem erfolgreichen Registrieren in Kompetenz C10 kann sich der Benutzer mit dem zuvor erstellten Konto anmelden. Hierfür wird die Email und das Passwort als Anmelde Faktoren verwendet:

Das Login überprüft dann durch ein select statement ob email und passwort mit einem Account übereinstimmen. Nach dem erfolgreichen Eingeben der Daten wird man auf eine Seite weitergeleitet, wo nur angemeldete Benutzer durch die Session Zugriff, haben. Von da aus hat der Benutzer mehrere Funktionen zur Verfügung, die ihm zuvor nicht zur Verfügung gestanden sind.

Der Code für das Login wurde im File signin.php geschrieben.

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*Ein Benutzer kann sich an meinem Projekt anmelden. Nach der Anmeldung stehen dem Benutzer weitere Funktionen zur Verfügung.*

### Kompetenz C12

Ein angemeldeter Benutzer kann sein Passwort ändern, indem er auf seinen Account auf der rechten Seite in der Navigationsleiste klickt. Von da aus kann der Benutzer Passwort wechseln auswählen und ein gewünschtes neues Passwort eingeben. Dies wurde im Code mit einem update gemacht. Dieser Code kann im File PasswordChange.php gefunden werden

Nach der erfolgreichen Eingabe des alten und neuen Passworts und dem Klicken auf senden wird das Passwort in der Datenbank geändert und der Benutzer weitergeleitet. Als query zum Updaten des Passworts wurde folgender code verwendet:

```
$query = "UPDATE benutzer SET Passwort = ? WHERE email = ?";
```

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*In meinem Projekt kann ein angemeldeter Benutzer sein Passwort ändern.*

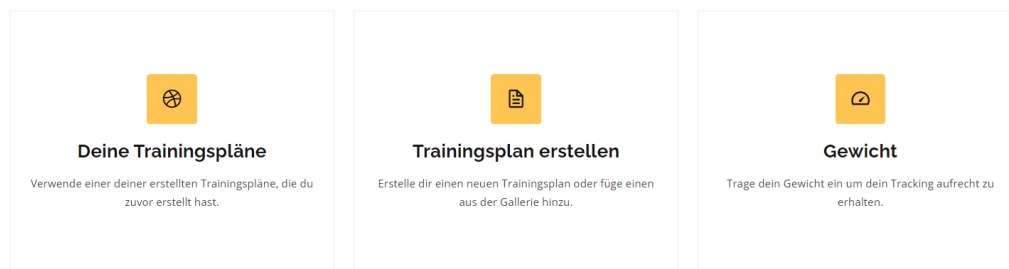


### Kompetenz C13

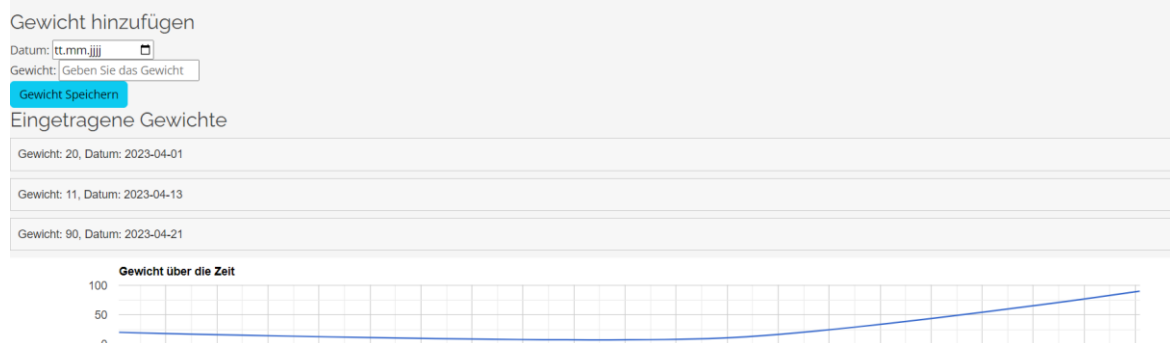
Auf der Webseite gibt es weitere Möglichkeiten Daten auf der Webseite zu erfassen. Hierfür kann als Beispiel das Gewicht eines Benutzers eingetragen und getrackt werden oder ein Trainingsplan mit 3 Übungen erstellt werden. Dazu kann der angemeldete Benutzer zwischen Trainingsplan erstellen und Gewicht auswählen:

FUNKTIONEN ÜBERSICHT

#### WAS WILLST DU ALS NÄCHSTES MACHEN?



In beiden Fällen können durch ein Formular Daten eingegeben werden, welche danach in die Datenbank geschrieben werden und auf der Webseite angezeigt werden. Als Beispiel wird das Gewicht verwendet welches in der untenstehenden Grafik noch einmal aufgezeigt wird.



Durch die verschiedenen Formulare können mehrere Informationen in der Datenbank erfasst werden. In dem Code wird das durch Inserts und Selects geregelt. Der Code kann in den Dateien `gewicht.php` und `trainingsplan-erstellen.php` gefunden werden.

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*In meinem Projekt können angemeldete Benutzer weitere Informationen in der Datenbank erfassen.*

#### Kompetenz C14

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

**Funktioniert noch nicht**

*Diese Datensätze können ausschliesslich vom Ersteller dieser Datensätze einzeln geändert werden*

### Kompetenz C15

Ein angemeldeter Benutzer kann unter Trainingsplan seine eigenen Trainingspläne anschauen. Dabei kann der Benutzer diese als einzige sehen und auch als einziger mit dem delete Button Löschen. Mit dem klicken auf den Lösch Button wird der Gesamte Trainingsplan aus der Datenbank gelöscht. Dies wird über ein Delete Statement im code gemacht:



Mit dem Klicken auf code werden zwei verschiedene Delete Statements ausgeführt, damit es in der Datenbank in der Uebungstabelle und in der Trainingsplan Tabelle gelöscht wird. Ein Beispiel Statement zum Löschen aus dem File trainingplan.php ist:

```
$query = "DELETE FROM uebungen WHERE Trainingplan_idTrainingplan=?";
```

Dies ermöglicht dem User das Löschen der Trainingspläne, die nicht benötigt werden.

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*Diese Datensätze können ausschliesslich vom Ersteller dieser Datensätze einzeln gelöscht werden.*

### Kompetenz C16

Es wurden SQL-Injektion konsequent verhindert durch das Verwenden von Parametrisierten abfragen und den Client und serverseitigen Validation. Ein Beispiel für eine Parametrisierten abfrage sieht wie folgt aus:

```
if (!$stmt->bind_param('sss',$idBenutzer, $datumgewicht, $gewicht))
```

Die Kompetenzen hierfür lauten (Zitat aus Kompetenzraster):

*In meinem Projekt wird SQL-Injection konsequent verhindert.*

## Lernprozesse

Flavio Schaffer


Marc Schmidlin


## Reflexion

Flavio Schaffer

Ich konnte in diesem Modul sehr viel neues und Spannendes dazu lernen. Ich habe insbesondere im Bereich PHP sehr viele neue Kenntnisse dazu gelernt. Durch das Projekt konnte ich die zuvor angeschauten Übungen sehr gut verstehen und selber anwenden.

Mir hat das gesamte Modul sehr viel Freude bereitet. Die Gruppenarbeit lief sehr gut und wir hatten kein Problem innerhalb der Gruppe. Die Zeit wurde jedoch nicht so gut eingeteilt und wir hatten zum Schluss etwas Stress mit dem Fertigstellen des Projektes. Trotz diesen Zeitschwierigkeiten bin ich sehr zufrieden mit dem gesamten Projekt und dem Endergebnis.

Für das nächste mal würde ich die Zeitplanung etwas anders gestalten und noch mehr mit dem Trello Arbeiten.

Besondere gut gefallen hat mir das Erstellen des Logins und die Zusammenarbeit im Team. Ich hoffe für die Zukunft, dass es weitere solche Projektarbeiten gibt und diese ebenfalls so gut laufen.

Marc Schmidlin

## Geschätzte Note

Als Schätznote würden wir uns selbst auf eine **5.5** Schätzen.

Wir sind als Team sehr zufrieden mit der Arbeit. Besonders durch die gute Strukturierung konnten wir auch gemeinsam eine gute Projektarbeit schreiben.

## Selbstständigkeitserklärung

Ich erkläre hiermit, dass es sich bei der von mir eingereichten schriftlichen Arbeit mit dem Titel

Dokumentation\_ProjektM151\_FM.pdf

um eine von mir selbst und ohne unerlaubte Beihilfe sowie in eigenen Worten verfasste Originalarbeit handelt.

Ich bestätige überdies, dass die Arbeit als Ganzes oder in Teilen noch nie zur Bewertung einer anderen schulischen Leistung an der GIBM oder an einer anderen Ausbildungseinrichtung verwendet wurde.

### 1.1 Verwendung von Quellen und Sekundärliteratur

Ich erkläre weiterhin, dass ich sämtliche in der eingereichten Arbeit enthaltenen Bezüge auf Quellen und Sekundärliteratur als solche kenntlich gemacht habe. Insbesondere bestätige ich, dass ich ausnahmslos und nach bestem Wissen sowohl bei wörtlich übernommenen Aussagen (Zitaten) als auch bei in eigenen Worten wiedergegebenen Aussagen anderer Autorinnen oder Autoren (Paraphrasen) die Urheberschaft angegeben habe.

### 1.2 Sanktionen

Ich nehme zur Kenntnis, dass Arbeiten, welche die Grundsätze der Selbstständigkeitserklärung verletzen – insbesondere solche, die Zitate oder Paraphrasen ohne Herkunftsangaben enthalten –, als Plagiat betrachtet werden und entsprechende rechtliche und disziplinarische Konsequenzen nach sich ziehen können.

Ich bestätige mit meiner Unterschrift die Richtigkeit dieser Angaben.

	<b>Autor 1</b>	<b>Autor 2</b>
Name:	Schaffer _____	Marc _____
Vorname:	Flavio _____	Schmidlin _____
Datum:	21.12.2022 _____	21.12.2022 _____
Unterschrift:	<u>f. Schaffer</u>	_____