



## Modul 13 Datenbanken



# Modulverwaltung FST

Daniel Otto, Frank Belyea

---

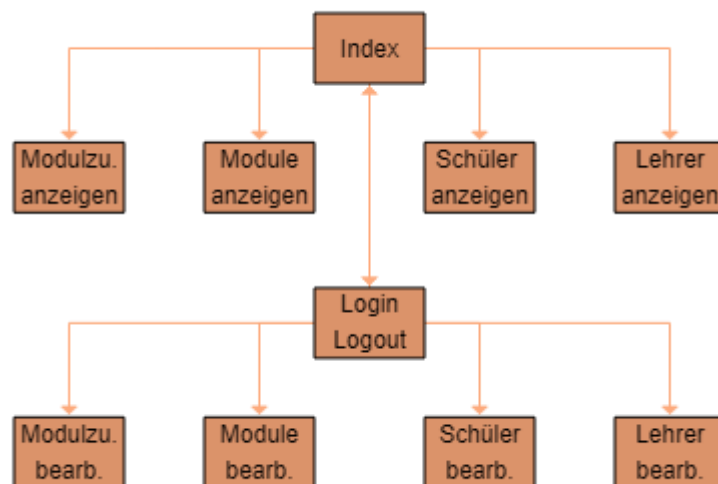
Meisterschule für Handwerker  
FSTINF 21

|                                 |    |
|---------------------------------|----|
| Aufgabenstellung.....           | 2  |
| Seitenbaum .....                | 2  |
| Seitenaufbau .....              | 3  |
| Navbar .....                    | 3  |
| Schüler Anzeigen .....          | 4  |
| Lehrer Anzeigen.....            | 5  |
| Code-Ausschnitte.....           | 5  |
| Module Anzeigen.....            | 6  |
| Modulzuordnung.....             | 7  |
| Loginbereich .....              | 8  |
| Code-Ausschnitte.....           | 8  |
| Schüler bearbeiten .....        | 9  |
| Lehrer bearbeiten.....          | 11 |
| Module bearbeiten.....          | 12 |
| Modulzuordnung bearbeiten ..... | 13 |

## Aufgabenstellung

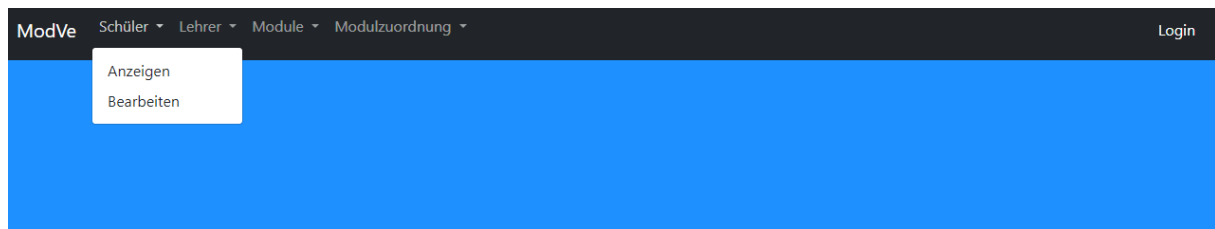
Es soll im Rahmen des Abschlussprojekts eine Web-Anwendung entwickelt werden, welche größtenteils die vorgeschlagene Entwurfslösung umsetzt. Es sind ausreichend Musterdaten für die Datenbanktabellen zu erstellen und mithilfe eines Installationsskripts soll die Datenbank inklusive der Tabellen und Musterdaten, sowie der Logindaten der Benutzer mit Standardpasswort, angelegt werden. Dem Benutzer soll die Möglichkeit geboten werden, alle Tabellen als PDF-Datei herunterzuladen

## Seitenbaum



# Seitenaufbau

## Navbar



Die Navigationsleiste ermöglicht den Zugriff auf diverse Seiten, wie beispielsweise "Schüler", "Lehrer", "Module", "Modulzuordnung" und den Login-Bereich. Jede dieser Seiten ist mit einem Dropdown-Menü versehen, das sowohl eine Anzeigen-Seite als auch eine Bearbeiten-Seite beinhaltet. Der Zugriff auf die Bearbeiten-Seite ausschließlich ist nur Benutzern nach erfolgreichem Login möglich.

## Schüler Anzeigen

| Filter nach PLZ: <input type="text"/> <a href="#">Filter Anwenden</a> <a href="#">PDF Download</a> |             |         |              |                      |       |                |
|--|-------------|---------|--------------|----------------------|-------|----------------|
| Snr.   | Name        | Vorname | Geburtsdatum | Straße               | PLZ   | Ort            |
| 1  | Otto        | Daniel  | 1995-08-07   | Am Alten Friedhof 1  | 67655 | Kaiserslautern |
| 2  | Belyea      | Frank   | 1992-10-27   | Gotherstraße 3       | 67663 | Kaiserslautern |
| 3  | Strelcov    | Artem   | 1992-05-22   | Münchstraße 14       | 67655 | Kaiserslautern |
| 4  | Fischer     | Michael | 1994-09-30   | Carl-Zeiss-Straße 10 | 67227 | Frankenthal    |
| 5  | Brusdeilins | Jan     | 2000-01-10   | Schreberstraße 96    | 67075 | Ludwigshafen   |

Auf der Seite "Schüler anzeigen" werden alle Schüler in einer tabellarischen Form dargestellt. Mit Hilfe des Filters kann die Anzeige auf eine bestimmte Postleitzahl beschränkt werden.

Die Tabelle enthält verschiedene Spalten, darunter "Snr." (Schülernummer), "Name", "Vorname", "Geburtsdatum", "Straße", "PLZ" und "Ort". Jeder Schüler wird in einer eigenen Zeile dargestellt.

Darüber hinaus enthält die Seite einen Button "PDF Download", der es ermöglicht, die angezeigten Schülerdaten als PDF herunterzuladen. Beim Klicken auf den Button wird ein PDF generiert, das die gleichen Daten enthält wie die Tabelle auf der Seite.

## Lehrer Anzeigen

| Filter nach Nachname: <input type="text"/> <a href="#">Filter Anwenden</a> <a href="#">PDF Download</a> |           |         |
|---|-----------|---------|
| Lid   | Name      | Vorname |
| 1   | Ritter    | Karl    |
| 2   | Obeldobel | Joachim |
| 3   | Clos      | Jürgen  |
| 4   | Hammen    | Andreas |

Auf der Seite "Lehrer anzeigen" werden alle Lehrer in einer tabellarischen Form dargestellt. Mit Hilfe des Filters kann nach bestimmten Lehrern gesucht werden. Die Suchergebnisse berücksichtigen nicht nur exakte Übereinstimmungen, sondern auch alle Lehrerdaten, deren Nachname mit der eingegebenen Zeichenfolge beginnt.

### Code-Ausschnitte

```
//Filter Anzeige
$filterLname = isset($_GET['lname']) ? $_GET['lname'] : '';

//fügt Wildcards am Anfang und Ende des namens hinzu, wählt entsprechend aus der Datenbank aus
if (!empty($filterLname)) {
    $stmt = $pdo->prepare('SELECT * FROM lehrer WHERE Lname LIKE :lname');
    $stmt->bindValue(':lname', '%' . $filterLname . '%');
} else {
    $stmt = $pdo->prepare('SELECT * FROM lehrer');
}

<!--Link zur PDF-Erstellung-->
<a href="./form/download_pdf.php" class="btn btn-success" style="margin-left: 0.5em;">PDF Download</a>

//Die Tabelle wird in einer Variable gespeichert die zum Erstellen der PDF verwendet wird
$html = '
    <table class="table table-striped table-dark table-hover table-sm rounded">
        <thead>
            <tr>
                <th>Lid</th>
                <th>Name</th>
                <th>Vorname</th>
            </tr>
        </thead>
        <tbody>';
```

## Module Anzeigen

| PDF Download |                               |              |           |      |
|--------------|-------------------------------|--------------|-----------|------|
| Modnr.       | Modul                         | Modulstunden | Lehrer ID | Raum |
| 1            | Technisches Singen            | 100          | 2         | 5215 |
| 2            | Angewandte Philosophie        | 0            | 1         | 3228 |
| 3            | Objektorientierte Messsysteme | 1200         | 3         | 5210 |
| 4            | Zustandslose Datenbanken      | 300          | 1         | 5215 |
| 5            | Metallkunde                   | 150          | 4         | 3228 |

Die Seite "Module anzeigen" ermöglicht die Anzeige aller verfügbaren Module. Die Moduldaten werden aus der Datenbank abgerufen und in einer übersichtlichen Tabelle dargestellt.

Die Tabelle enthält die folgenden Spalten:

- **Modnr.:** Die Modulnummer, die jedes Modul eindeutig identifiziert.
- **Modul:** Der Name des Moduls.
- **Modulstunden:** Die Anzahl der Stunden, die für das Modul vorgesehen sind.
- **Lehrer ID:** Die ID des Lehrers, der das Modul unterrichtet.
- **Raum:** Die Raumnummer, in der das Modul stattfindet.

Jedes Modul wird in einer eigenen Tabellenzeile mit den entsprechenden Daten angezeigt.

### Code-Ausschnitte

```
$stmt = $pdo->prepare('SELECT * FROM module');  
$stmt->execute();  
foreach ($stmt as $result) {  
    $html .= "<tr>  
        <td>" . $result['Modnr'] . "</td>  
        <td>" . $result['Modu'] . "</td>  
        <td>" . $result['Mstd'] . "</td>  
        <td>" . $result['Lid'] . "</td>  
        <td>" . $result['Rnr'] . "</td>  
    </tr>";  
}
```

Hier wird nur eine simple Datenbankabfrage gesendet und im Nachhinein mit einer foreach-Schleife die Daten als Tabelle ausgegeben.

## Modulzuordnung

PDF Download

| Modnr. | Modul                         | Snr | Schüler Vorname | Schüler Nachname |
|--------|-------------------------------|-----|-----------------|------------------|
| 1      | Technisches Singen            | 3   | Artem           | Strelcov         |
| 2      | Angewandte Philosophie        | 2   | Frank           | Belyea           |
| 3      | Objektorientierte Messsysteme | 3   | Artem           | Strelcov         |
| 5      | Metallkunde                   | 1   | Daniel          | Otto             |

Die Seite "Modulzuordnung bearbeiten" ermöglicht die Anzeige der Zuordnung von Modulen zu Schülern. Diese Zuordnungsinformationen werden aus der Datenbank abgerufen und in einer übersichtlichen Tabelle dargestellt.

Die Tabelle enthält die folgenden Spalten:

- Modnr.: Die Modulnummer, die jedes Modul eindeutig identifiziert.
- Modul: Der Name des Moduls.
- Snr: Die Schülernummer, die jeden Schüler eindeutig identifiziert.
- Schüler Vorname: Der Vorname des Schülers, dem das Modul zugeordnet ist.
- Schüler Nachname: Der Nachname des Schülers, dem das Modul zugeordnet ist.

Jede Modulzuordnung wird in einer eigenen Tabellenzeile mit den entsprechenden Daten angezeigt.

### Code-Ausschnitte

Die Modulzuordnung wird ähnlich wie bei der Seite „Module anzeigen“ gehandhabt. Nur die Datenbankabfrage arbeitet mit zwei Inner Joins.

```
$stmt = $pdo->prepare('SELECT * FROM module INNER JOIN modulzuordnung on module.Modnr =  
modulzuordnung.Modnr  
INNER JOIN schueler on modulzuordnung.Snr = schueler.Snr');
```

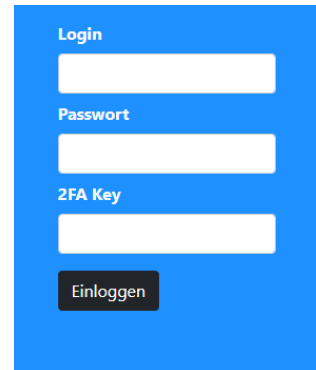
Hier holen wir zuerst die Informationen der „module“ Tabelle und verbinde sie mithilfe des join-Befehls mit der Tabelle „modulzuordnung“ mit der Bedingung das die Modnr übereinstimmt. Darauf folgt ein weiterer join-Befehl der die „schueler“ Tabelle mit der „modulzuordnung“ Tabelle verbindet. Somit kriegen wir alle nötigen Informationen die wir zum ausgeben der gewünschten Daten benötigen.



## Loginbereich

Nachdem der Benutzer die erforderlichen Informationen eingegeben hat, kann er auf den "Einloggen"-Button klicken, um den Anmeldevorgang abzuschließen.

Bevor der Benutzer angemeldet wird, werden die eingegebenen Informationen mit den Daten in der MySQL-Datenbank überprüft. Es wird geprüft, ob die eingegebene Login-ID mit einem gültigen Benutzernamen übereinstimmt und ob das eingegebene Passwort korrekt ist. Darüber hinaus wird auch überprüft, ob der Benutzer über die erforderliche Zugriffsberechtigung verfügt und ob der eingegebene 2FA-Schlüssel mit dem aktuellen 2FA-Code übereinstimmt.

A blue rectangular login form. It contains three white input fields stacked vertically. The first field is labeled 'Login', the second 'Passwort', and the third '2FA Key'. Below the third field is a dark blue button with the text 'Einloggen' in white.

Wenn alle Überprüfungen erfolgreich sind, wird eine Sitzung gestartet und der Benutzer wird zur Hauptseite der Modulverwaltung weitergeleitet. Andernfalls wird eine Fehlermeldung angezeigt, die den Benutzer darüber informiert, dass der Login-Versuch gescheitert ist und er seine Login-ID und sein Passwort überprüfen soll.

## Code-Ausschnitte

```
$current_code_2fa = trim(shell_exec("oathtool --base32 --totp $fakey"));  
$pwdVerify = password_verify($_POST['passwd'], $password);
```

### 1. `$current_code_2fa = trim(shell_exec("oathtool --base32 --totp $fakey"));`

- Diese Zeile generiert den aktuellen 2FA-Code (Two-Factor Authentication) basierend auf dem gespeicherten 2FA-Schlüssel (**\$fakey**).
- Das Kommandozeilentool "oathtool" wird verwendet, um den 2FA-Code zu generieren.
- **--base32** gibt an, dass der 2FA-Schlüssel im Base32-Format vorliegt.
- **--totp** steht für "Time-Based One-Time Password" und gibt an, dass ein zeitbasiertes Einmalpasswort generiert werden soll.
- **\$fakey** ist der gespeicherte 2FA-Schlüssel, der an das "oathtool"-Kommando übergeben wird.
- **shell\_exec()** führt das Shell-Kommando aus und gibt das Ergebnis zurück.
- **trim()** wird verwendet, um Leerzeichen am Anfang und Ende des generierten Codes zu entfernen.
- Der generierte 2FA-Code wird der Variablen **\$current\_code\_2fa** zugewiesen.

### 2. `$pwdVerify = password_verify($_POST['passwd'], $password);`

- Diese Zeile überprüft das eingegebene Passwort (**\$\_POST['passwd']**) mit dem gehashten Passwort, das in der Datenbank (**\$password**) gespeichert ist.
- **password\_verify()** ist eine Funktion in PHP, die verwendet wird, um das eingegebene Passwort mit dem gehashten Passwort zu vergleichen.
- **\$\_POST['passwd']** enthält das eingegebene Passwort aus dem Formular.
- **\$password** enthält das gehashte Passwort aus der Datenbank.
- Das Ergebnis der Überprüfung, ob das Passwort übereinstimmt oder nicht, wird der Variablen **\$pwdVerify** zugewiesen. Wenn das Passwort übereinstimmt, ist der Wert von **\$pwdVerify true**, andernfalls **false**.

## Schüler bearbeiten

| Hinzufügen |             |         |              |                      |       |                |  |
|------------|-------------|---------|--------------|----------------------|-------|----------------|--|
| Snr.       | Name        | Vorname | Geburtsdatum | Straße               | PLZ   | Ort            | Actions  |
| 1          | Otto        | Daniel  | 1995-08-07   | Am Alten Friedhof 1  | 67655 | Kaiserslautern | <button>Bearbeiten</button> <button>Löschen</button> |
| 2          | Belyea      | Frank   | 1992-10-27   | Gotherstraße 3       | 67663 | Kaiserslautern | <button>Bearbeiten</button> <button>Löschen</button> |
| 3          | Strelcov    | Artem   | 1992-05-22   | Münchstraße 14       | 67655 | Kaiserslautern | <button>Bearbeiten</button> <button>Löschen</button> |
| 4          | Fischer     | Michael | 1994-09-30   | Carl-Zeiss-Straße 10 | 67227 | Frankenthal    | <button>Bearbeiten</button> <button>Löschen</button> |
| 5          | Brusdeilins | Jan     | 2000-01-10   | Schreiberstraße 96   | 67075 | Ludwigshafen   | <button>Bearbeiten</button> <button>Löschen</button> |

| Snr. | Name                              | Vorname                              | Geburtsdatum                              | Straße                              | PLZ                              | Ort                              | Actions   |
|------|-----------------------------------|--------------------------------------|---|-------------------------------------|----------------------------------|----------------------------------|---|
| x    | <input type="text" value="Name"/> | <input type="text" value="Vorname"/> | <input type="text" value="Geburtsdatum"/> | <input type="text" value="Straße"/> | <input type="text" value="PLZ"/> | <input type="text" value="Ort"/> | <button>Speichern</button> <button>Abbrechen</button> |

| Snr. | Name                              | Vorname                             | Geburtsdatum                            | Straße   | PLZ                                | Ort   | Actions   |
|------|-----------------------------------|-------------------------------------|---|--|------------------------------------|---|---|
| 1    | <input type="text" value="Otto"/> | <input type="text" value="Daniel"/> | <input type="text" value="1995-08-07"/> | <input type="text" value="Am Alten Friedhof 1"/> | <input type="text" value="67655"/> | <input type="text" value="Kaiserslautern"/> | <button>Speichern</button> <button>Verwerfen</button> |

Die Seite "Schüler bearbeiten" enthält eine Tabelle, die Schülerdaten anzeigt.

Die Tabelle enthält auch eine versteckte Zeile, die für das Hinzufügen eines neuen Schülers verwendet wird. Wenn der Benutzer auf die "Hinzufügen" Schaltfläche klickt, wird diese Zeile angezeigt und ermöglicht das Eingeben neuer Schülerdaten.

Um einen vorhandenen Schüler zu bearbeiten, kann der Benutzer auf die "Bearbeiten" Schaltfläche in der entsprechenden Zeile klicken. Die Zeile wechselt in den Bearbeitungsmodus, in dem die Schülerdaten als Eingabefelder angezeigt werden. Der Benutzer kann die Daten bearbeiten und anschließend auf die "Speichern" Schaltfläche klicken, um die Änderungen zu speichern. Alternativ kann der Benutzer auf die "Abbrechen" Schaltfläche klicken, um die Bearbeitung abzubrechen und die ursprünglichen Daten wiederherzustellen.

Wenn der Benutzer auf die "Löschen" Schaltfläche klickt, wird eine Bestätigungsmeldung angezeigt. Wenn der Benutzer die Löschung bestätigt, wird der entsprechende Schüler aus der Datenbank entfernt. Es wird überprüft, ob der Schüler Module belegt. Wenn dies der Fall ist, wird das Löschen verhindert und eine entsprechende Fehlermeldung angezeigt.

Zusätzlich zur HTML-Datei enthält das Skript am Ende PHP-Code, der für die Aktualisierung und das Löschen von Schülerdaten zuständig ist. Die Aktualisierung erfolgt über einen AJAX-Anruf an `./ajax/updateSchulData.php`, während das Löschen über einen AJAX-Anruf an `./ajax/removeSchulData.php` erfolgt.

## Code-Ausschnitte

```
<!--Der Style dient zum Ausblenden der Hinzufügen-Zeile-->
<style>
    .add-row {
        display: none;
    }
</style>
</head>
<!-- Versteckte Zeile zum Hinzufügen eines Lehrers -->
    <tr class="add-row ">

// AJAX request to update the database
```

```
$.ajax({
    type: 'POST',
    url: './ajax/removeSchulData.php',
    data: data,
    success: function(response) {
        // Handle success if necessary
        alert(response);
    },
    error: function(xhr, status, error) {
        console.log(xhr.responseText);
        // Handle error if necessary
    }
})
```

Die Funktion \$.ajax() wird verwendet, um eine asynchrone Anfrage an den Server zu senden und auf Erfolg oder Fehler zu reagieren, indem eine PHP-Datei aufgerufen wird, um Daten zu verarbeiten und eine Serverantwort zurückzugeben.

```
$snr = $_POST['snr'];

$sqlCheck = "SELECT * from modulzuordnung WHERE Snr=:snr";
$stmtCheck = $pdo->prepare($sqlCheck);
$stmtCheck->bindParam(':snr', $snr);
$stmtCheck->execute();

if ($stmtCheck->rowCount() > 0) {
    $response = "Löschen nicht möglich: Der Schüler belegt ein oder mehrere Module";
} else {
    $sql = "DELETE FROM schueler WHERE Snr=:snr";
    $stmt = $pdo->prepare($sql);
    $stmt->bindParam(':snr', $snr);
    $stmt->execute();
    $response = "Löschen Erfolgreich";
}
echo $response;
```

## Lehrer bearbeiten

| Hinzufügen |           |         |  |
|------------|-----------|---------|--|
| Lid        | Name      | Vorname | Actions  |
| 1          | Ritter    | Karl    | <button>Bearbeiten</button> <button>Löschen</button> |
| 2          | Obeldobel | Joachim | <button>Bearbeiten</button> <button>Löschen</button> |
| 3          | Clos      | Jürgen  | <button>Bearbeiten</button> <button>Löschen</button> |
| 4          | Hammen    | Andreas | <button>Bearbeiten</button> <button>Löschen</button> |

| Lid | Name                                   | Vorname                              | Actions                     |                            |
|-----|--|--------------------------------------|-----------------------------|----------------------------|
| x   | <input type="text" value="Name"/>      | <input type="text" value="Vorname"/> | <button>Speichern</button>  | <button>Abbrechen</button> |
| 1   | Ritter                                 | Karl                                 | <button>Bearbeiten</button> | <button>Löschen</button>   |
| 2   | <input type="text" value="Obeldobel"/> | <input type="text" value="Joachim"/> | <button>Speichern</button>  | <button>Verwerfen</button> |
| 3   | Clos                                   | Jürgen                               | <button>Bearbeiten</button> | <button>Löschen</button>   |
| 4   | Hammen                                 | Andreas                              | <button>Bearbeiten</button> | <button>Löschen</button>   |

Die Seite "Lehrer bearbeiten" weist einen identischen Aufbau wie die Seite "Schüler bearbeiten" auf und ermöglicht die Bearbeitung von Lehrerdaten in der Datenbank. Der maßgebliche Unterschied besteht darin, dass das Löschen eines Lehrers auf dieser Seite nur gestattet ist, sofern der betreffende Lehrer keine Module anbietet. Sollte ein Lehrer Module anbieten, wird dem Benutzer eine entsprechende Meldung angezeigt, und das Löschen wird verhindert. Abgesehen von dieser Bedingung funktioniert die Seite "Lehrer bearbeiten" analog zur Seite "Schüler bearbeiten".

## Module bearbeiten

| <div>Hinzufügen Löschen</div> |                               |              |           |      |                             |
|-------------------------------|-------------------------------|--------------|-----------|------|-----------------------------|
| Modnr.                        | Modul                         | Modulstunden | Lehrer ID | Raum |                             |
| 1                             | Technisches Singen            | 100          | 2         | 5215 | <button>Bearbeiten</button> |
| 2                             | Angewandte Philosophie        | 0            | 1         | 3228 | <button>Bearbeiten</button> |
| 3                             | Objektorientierte Messsysteme | 1200         | 3         | 5210 | <button>Bearbeiten</button> |
| 4                             | Zustandslose Datenbanken      | 300          | 1         | 5215 | <button>Bearbeiten</button> |
| 5                             | Metallkunde                   | 150          | 4         | 3228 | <button>Bearbeiten</button> |

Die Seite "Modul Bearbeiten" ermöglicht es dem Benutzer, bestehende Modulinformationen zu ändern. Die Seite zeigt eine Tabelle mit verschiedenen Modulen an, die aus einer Datenbank abgerufen werden. Jedes Modul in der Tabelle hat eine "Bearbeiten" Schaltfläche, die beim Klicken ein Modalfenster öffnet. Im Modalfenster werden die vorhandenen Moduldaten angezeigt und der Benutzer kann die Informationen bearbeiten.

Modul löschen

Modul: Zustandslose Datenbanken

Nur Module ohne Belegung lassen sich löschen!

Abbrechen

Modul löschen

Modul bearbeiten

Modul: Technisches Singen

Modulstunden: 100

Lehrer: Ritter

Raum: 5215

Abbrechen

Änderungen speichern

Das Modalfenster enthält Eingabefelder für den Modulnamen, die Modulstunden, den Lehrer und den Raum. Die Eingabefelder sind vorab mit den aktuellen Modulwerten gefüllt. Der Benutzer kann die Werte in den Eingabefeldern bearbeiten. Nachdem der Benutzer die Änderungen vorgenommen hat, kann er auf die Schaltfläche "Änderungen speichern" klicken, um die Aktualisierung durchzuführen.

Die Bearbeitungsfunktion verwendet AJAX-Anfragen, um die Moduldaten abzurufen und zu aktualisieren, ohne die Seite neu zu laden. Wenn der Benutzer auf die Schaltfläche "Bearbeiten" klickt, wird eine AJAX-Anfrage gesendet, um die aktuellen Moduldaten abzurufen und im Modalfenster anzuzeigen. Nachdem der Benutzer die Änderungen vorgenommen hat und auf "Änderungen speichern" klickt, wird eine weitere AJAX-Anfrage gesendet, um die aktualisierten Moduldaten an den Server zu senden und in der Datenbank zu speichern.

Modul hinzufügen

Modul: Modulname

Modulstunden: Stundenzahl

Lehrer: Ritter

Raum: 5215

Abbrechen

Hinzufügen

Nachdem die Änderungen gespeichert wurden, wird der Benutzer zur Modulbearbeitungsseite zurückgeleitet, wo die aktualisierten Modulinformationen angezeigt werden

## Modulzuordnung bearbeiten

| <div>Hinzufügen Löschen</div> |                               |     |                 |                  |
|-------------------------------|-------------------------------|-----|-----------------|------------------|
| Modnr.                        | Modul                         | Snr | Schüler Vorname | Schüler Nachname |
| 1                             | Technisches Singen            | 3   | Artem           | Strelcov         |
| 2                             | Angewandte Philosophie        | 2   | Frank           | Belyea           |
| 3                             | Objektorientierte Messsysteme | 3   | Artem           | Strelcov         |
| 5                             | Metallkunde                   | 1   | Daniel          | Otto             |

Die Seite "Modulzuordnung bearbeiten" ermöglicht es den Benutzern, die Modulzuordnungen anzupassen. Durch Klicken auf die Buttons "Hinzufügen" oder "Löschen" werden Modals geöffnet, die es ermöglichen, neue Modulzuordnungen hinzuzufügen oder vorhandene Modulzuordnungen zu entfernen.

Das Hinzufügen-Modal enthält ein Formular, in dem Benutzer die erforderlichen Informationen eingeben können, um eine neue Modulzuordnung hinzuzufügen. Das Löschen-Modal zeigt eine Liste der aktuell löschbaren Modulzuordnungen an, aus der der Benutzer eine Auswahl treffen kann, um die entsprechende Modulzuordnung zu entfernen.

Modulzuordnung hinzufügen

Modul: 

Technisches Singen

Schülernr: 

1

2

3

4

5

Abbrechen

Hinzufügen

Modulzuordnung löschen

Modul: 

Technisches Singen

Schüler ID: 

3

Abbrechen

Löschen

Um die Modals mit den erforderlichen Daten zu füllen, werden AJAX-Anfragen verwendet. Wenn der Benutzer auf den "Hinzufügen"-Button klickt, wird die "addModzuData.php" aufgerufen, um die Dropdown-Listen mit den verfügbaren Modulen und Schülern für die Modulzuordnung zu generieren. Wenn der Benutzer auf den "Löschen"-Button klickt, wird die "removeModzuData.php" aufgerufen, um die Dropdown-Listen mit den vorhandenen

Modulzuordnungen und Schülern für das Löschen der Modulzuordnung zu generieren.

Diese Funktionalitäten bieten den Benutzern eine benutzerfreundliche Möglichkeit, die Modulzuordnungen zu bearbeiten, indem sie neue Zuordnungen hinzufügen oder bestehende entfernen. Durch die Verwendung von Modals und AJAX-Anfragen wird eine interaktive und reaktionsschnelle Benutzererfahrung gewährleistet.

## Installationskript

Gewünscht war ein Installationsskript, mit dem man die Datenbank einfach einspielen kann in der schon Beispieldaten schon vorhanden sind sowie ein Standard-Userlogin.

```
#!/bin/bash

echo "Dies ist ein Installationsskript für ein Modulverwaltungssystem."
read -p "Bitte gib deinen MySQL Nutzernamen ein: " db_login
echo -n "Bitte gib dein MySQL Passwort ein: "
read -s db_pass
echo

mysql --user="$db_login" --password="$db_pass" --verbose <
base_database.sql || exit 1

echo "Datenbank wurde eingespielt!"
echo "Standardnutzer mit Passwort wurde erstellt!"
echo "Die Logindaten fuer diesen Nutzer sind der Dokumentation zu
entnehmen."
```

Hier sehen wir das von uns erstellte Skript, bei diesem wird der Nutzer gebeten seine Logindaten der Datenbank anzugeben, darauf wird bei korrekter Angabe die Datenbank eingespielt und mit einer Erfolgsmeldung in der Konsole begrüßt.

Die Standard Anmeldedaten wären:

User: ritter

Passwort: test

2FA Key:



## PDF-Funktion

Wir haben uns dazu entschieden die TCPDF-Bibliothek zu benutzen, da diese sehr einfach PDF-Dateien mithilfe von HTML erstellen kann.

```
$pdf = new TCPDF(PDF_PAGE_ORIENTATION, PDF_UNIT, PDF_PAGE_FORMAT, true, 'UTF-8', false);
$pdf->setPrintHeader(false);
$pdf->AddPage();
$pdf->writeHTMLCell(0, 0, '', '', $_SESSION['pdf_html'], 0, 1, 0, true, '', true);
$pdf->Output('test.pdf', 'I');
```

Es wird nur der HTML-Code der gewünschten Tabelle an die Bibliothek übergeben und diese generiert die gewünschte PDF und stellt diese bereit. Der HTML-Code von der Tabelle wird von jeder „Anzeigen“-Seite in eine Variable gespeichert und beim betätigen des Button „PDF laden...“ an die „download\_pdf“-Form geschickt.