

7. Übung (Datenbanken)

Laden Sie aus Moodle **Aufgabe7.zip** herunter. Dieses enthält die Dateien

- **Aufgabe7.php**,
- **mockupdatatable.sql** und
- **DB.php**.
- Außerdem noch **css**-Styles → belassen Sie es am besten beim Einbinden dieser Dateien und binden nicht selbst Bootstrap und eigene css-Dateien ein

Ziel der Übung:

- Verwenden von MySQL und phpMyAdmin
- Anwenden von PDO
- Verwenden von Klassen in PHP
- Verwenden von prepared Statements für Datenbankabfragen

Vorbereitung:

1. Loggen Sie sich auf dem Studi-Server ein und wählen Sie „**MySQL**“. Legen Sie dort eine Datenbank „**mockupdatadb**“ an.
2. Rufen Sie auf dem Studiserver **phpMyAdmin** auf und loggen sich ein.
3. Wählen Sie unter Datenbanken die „**__Matrikelnummer__mockupdatadb**“-Datenbank aus.
4. Importieren Sie über phpMyAdmin die „**mockupdatatable.sql**“.
5. Prüfen Sie, ob eine Tabelle „**mockupdatatable**“ erstellt und mit 50 Datensätzen gefüllt wurde.

Aufgabe 7:

1. Ergänzen Sie die Klasse **DB.php** (hauptsächlich um SQL-Anfragen und das Binden der Parameter). Diese Klasse enthält:
 1. Ein Feld (Variable) **\$connection** für die DB-Verbindung
 2. Einen Konstruktor, dem der Name der Datenbank und als optionale Parameter der Host, der User und das Passwort übergeben werden
Innerhalb des Konstruktors erzeugen Sie ein neues PDO-Objekt. Dabei dient die Variable **\$connection** als Datenbankhandle
(\$this->connection
= new PDO("mysql:host=\$host;dbname=\$name;charset=utf8", \$user,
\$pass);)
 3. Eine Funktion **all()**, die alle Einträge der Tabelle mockupdatatable zurückgibt (fetchAll())
 4. Eine Funktion **get(\$id)**, die den Datensatz zurückgibt, bei dem die id gleich \$id ist
 5. Eine Funktion **add(array \$daten)**, die die im Array \$daten übergebenen Daten (vorname, nachname, email, ipnr) in die Tabelle einfügt (hier ohne id).
 6. Eine Funktion **edit(array \$daten)**, die den Datensatz ändert, dessen id der im Array \$daten enthaltenen id entspricht (hier mit id).
 7. Eine Funktion **delete(\$id)**, die den Datensatz mit der id == \$id löscht.

- Erstellen Sie eine Webseite, die aus zwei Teilen besteht:
 - Im oberen Teil der Seite erscheint ein Formular. Dieses Formular kann benutzt werden, um weitere Teilnehmerinnen zur Datenbank hinzuzufügen (d.h. wenn Sie das Formular ausfüllen und absenden, wird die Funktion **add(array \$daten)** aufgerufen).
 - Im unteren Teil der Seite erscheinen alle Einträge aus der Datenbank als „Karten“. Fährt man über eine „Karte“ erscheinen zwei kleine Hyperlinks in Form von Buttons. Der eine Button heißt „Edit“ und der andere „Delete“. Klickt man auf den Edit-Button, erscheint der Datensatz im Formular und kann geändert werden (Absenden des Formulars bewirkt dann eine Änderung des Datensatzes → **edit(array \$daten)** verwenden). Klickt man auf den Delete-Button wird der Datensatz gelöscht.

Screenshots:

The screenshot shows a web application interface. At the top, there is a green header bar with the text "Teilnehmerin hinzufügen". Below this is a form with four input fields: "Vorname", "Nachname", "E-Mail-Adresse", and "IP-Adresse". A green button labeled "Anlegen" is positioned below the form. Below the form, there is a grid of 12 participant cards arranged in 3 rows and 4 columns. Each card displays a first name, a last name, an email address, and an IP address.

| Vorname | Nachname | E-Mail-Adresse | IP-Adresse |
|---------|----------|-----------------------------|----------------|
| Adam | Anderson | aanderson8@google.fr | 118.93.83.157 |
| Susan | Andrews | sandrewsn@google.co.jp | 228.214.9.251 |
| Alan | Bradley | abradley1c@globo.com | 229152117127 |
| Anne | Brooks | abrooks16@bravesites.com | 243.159.39.234 |
| Russell | Brown | rbrownq@nifty.com | 215.38.120.242 |
| Ryan | Burton | rburton18@foxnews.com | 159.60.107.14 |
| Roy | Campbell | rcampbell1@geocities.com | 237.232.34.20 |
| Russell | Campbell | rcampbell17@eventbrite.com | 251.2.92.63 |
| Bonnie | Coleman | bcoleman11@fc2.com | 109150122102 |
| Ernest | Coleman | ecoleman15@businessweek.com | 213.173.4.7 |
| Richard | Cruz | rcruz7@unc.edu | 235.124.23.221 |
| Sean | Cruz | scruz10@answers.com | 92.255.49.227 |

Abbildung 1: Ansicht nach Start und nach Absenden des Formulars

This screenshot shows a portion of the participant cards from the previous image. The card for Roy Campbell now includes two buttons: "Edit" and "Delete".

| Vorname | Nachname | E-Mail-Adresse | IP-Adresse | Buttons |
|---------|----------|----------------------------|----------------|--------------|
| Russell | Brown | rbrownq@nifty.com | 215.38.120.242 | |
| Ryan | Burton | rburton18@foxnews.com | 159.60.107.14 | |
| Roy | Campbell | rcampbell1@geocities.com | 237.232.34.20 | Edit, Delete |
| Russell | Campbell | rcampbell17@eventbrite.com | 251.2.92.63 | |

Abbildung 2: Mouse-Over auf eine "Karte"

The image shows a web application interface. At the top, there is a green header bar with the text "Eintrag bearbeiten". Below this is a form with four input fields containing the following data: "Roy", "Campbell", "rcampbell1@geocities.com", and "237.232.34.20". A green button labeled "Aktualisieren" is positioned below the form. Below the form is a grid of 12 data cards arranged in 3 rows and 4 columns. Each card displays a name, a last name, an email address, and an IP address. The card for "Roy Campbell" (rcampbell1@geocities.com, 237.232.34.20) is highlighted with a mouse-over effect, showing "Edit" and "Delete" buttons in the top right corner.

| Name | Last Name | Email | IP Address |
|---------|-----------|----------------------------|----------------|
| Adam | Anderson | aanderson8@google.fr | 118.93.83.157 |
| Susan | Andrews | sandrewsn@google.co.jp | 228.214.9.251 |
| Alan | Bradley | abradley1c@globo.com | 229152117127 |
| Anne | Brooks | abrooks16@bravesites.com | 243.159.39.234 |
| Russell | Brown | rbrownq@nifty.com | 215.38.120.242 |
| Ryan | Burton | rburton18@foxnews.com | 159.60.107.14 |
| Roy | Campbell | rcampbell1@geocities.com | 237.232.34.20 |
| Russell | Campbell | rcampbell17@eventbrite.com | 251.2.92.63 |
| Bonnie | Coleman | | |
| Ernest | Coleman | | |
| Richard | Cruz | | |
| Sean | Cruz | | |

Abbildung 3: Nach Klick auf Button "Edit"

Tipps:

für die Verwaltung, welcher der beiden Hyperlinkbuttons (Edit oder Delete) für welchen Datensatz angeklickt wurde, empfiehlt sich die Verwendung der GET-Methode. An die URL werden zwei Parameter angehängt: **command** (kann den Wert **edit** oder **delete** haben) und die **id** (mit dem Wert der **id** des Datensatzes)

Zur Verwaltung des Formulars empfiehlt sich die POST-Methode. Bedenken Sie, dass Sie beim Ändern eines Datensatzes die **id** kennen, beim Hinzufügen eines neuen Datensatzes die **id** nicht kennen -> somit ist eine Fallunterscheidung möglich (**if (isset(\$id)) // edit else // add**) → für den Fall, dass Sie die **id** kennen (edit), können Sie wieder ein hidden-Feld in das Formular einfügen