

PHP & HTML

Nun widmen wir uns der Integration des PHP-Code in das HTML-Gerüst.

Wir werden dabei drei Bereiche abdecken:

1. Durchschnitt berechnen
2. Tabelle mit allen Schülern anzeigen
3. Neuen Eintrag hinzufügen

Ausgangssituation

Um die weiteren Schritte zu absolvieren, brauchen wir folgenden Dateien:

`index.php`

```
<!DOCTYPE html>
<html lang="de" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Notentracker</title>

    <link rel="stylesheet" href="index.css">
  </head>

  <body>
    <header>
      <!--Übersichtskasten links oben für Kurs und Jahrgang-->
      <div class="Kurs-Overview">
        <b>Kurs/Klasse</b><span class="Teiler"/><b>Jahrgang</b>
      </div>

      <!--Übersichtskasten links oben für Lehrer und Halbjahr-->
      <div class="Lehrer-Semester">
        <b>Lehrkraft: </b>
        <span>LehrerVariabel</span><br/>
        <b>Halbjahr: </b>
        <span>Zeitraum</span>
        <RiPencilFill class="PencilIcon" size="1.5em"/>
      </div>

      <!--Übersichtskasten rechts oben für Username und abmelden-->
      <div class="user-block">
        <FaUser class="UserIcon"/><b>Username</b><span class="Teiler"/>
        <b>abmelden</b>
      </div>
    </header>

    <div class="Uebersicht">

      <!--Übersichtskasten über Tabelle für Notendurchschnitt des Kurses-->
      <div class="UebersichtBox">
```

```

        <!--Hier kommt der Durchschnitt rein-->
        Durchschnitt:
    </div>
</div>

<!--Tabelle mit allen Schülern eines Kurses-->
<div class="containerDiv">
<table class="Tabelle">
    <thead>
        <tr>
            <th>Vorname</th>
            <th>Nachname</th>
            <th>Punkte</th>
            <th>Datum</th>
            <th>Art der Leistung</th>
        </tr>
    </thead>
    <tbody>

        <!--Hier kommt Code für die Tabelle rein-->

        <tr>
            <td>Vorname</td>
            <td>Nachname</td>
            <td>Punkte</td>
            <td>Datum</td>
            <td>Art_der_Leistung</td>
        </tr>

    </tbody>

</table>
</div>

<!--Eingabefeld für das Einfügen neuer Schüler und Noten-->

<form class="NeueNoteOverlay" action="/eintrag-hinzufügen.php">
    <h2 align="middle">Note hinzufügen</h2>
    <!--Vorname-->
        <label class="LabelOben">Vorname</label>
        <input id="neuer_vorname" class="NeueNoteBox" name="neuer_vorname"
type="text" required="text" value="">
        <label class="LabelUnten">max. 50 Zeichen</label>
    <!--Nachname-->
        <label class="LabelOben">Nachname</label>
        <input id="neuer_nachname" class="NeueNoteBox" name="neuer_nachname"
type="text" required="text" value="">
        <label class="LabelUnten">max. 50 Zeichen</label>
    <!--Punkte-->
        <label class="LabelOben">Punkte</label>
        <input id="neue_punkte" class="NeueNoteBox" name="neue_punkte" min="0"
max="15" required="number" type="number" value="">
        <label class="LabelUnten">0 bis 15</label>
    <!--Art der Leistung-->
        <label class="LabelOben">Leistung</label>
        <input id="neue_art" class="NeueNoteBox" name="neue_art"
maxlength="50" required="required" type="text" value="">
        <label class="LabelUnten">max. 50 Zeichen</label>

```

```

<!--Datum-->
    <label class="LabelOben">Datum</label>
    <input id="neues_datum" class="NeueNoteBox" name="neues_datum"
required="required" type="date" value="">
    <label class="LabelUnten"></label>

<!--Eingaben löschen oder in Datenbank übernehmen-->
    <span>
        <input class="NeueNoteSubmit" type="submit" value="Note hinzufügen">
        <!-- <button class="NeueNoteReset" type="reset">Reset</button> -->
    </span>
</form>

</body>
</html>

```

Mit dem Stylesheet `index.css`

```

body {
    background-color: #1D2025;
    font-family: Arial;
    color: #FFFFFF;
}

/*Übersichts-Dinger*/
.uebersicht {
    position: absolute;
    padding: .5%;
    width: 30%;
    margin-top: 17%;
}

/*Boxen um den Text*/
.uebersichtBox {
    background-color: #39424B;
    border-radius: 10px;
    padding: 5%;
    margin: 5%;
    font-weight: bold;
}

.NoteHinzufuegen {
    background-color: #02D2BF;
    border-radius: 100em;
    padding: 5%;
    margin: 5%;
    border: none;
    font-weight: bold;
    color: #FFFFFF;
    cursor: pointer;
    float: right;
}

/* Variablen */
.variablen {

```

```
font-weight: normal;
}

/*Overlay fuer neue Noten*/
.NeueNoteOverlay {
  position: absolute;
  border-radius: 10px;
  padding: 2.5%;
  width: 25%;
  bottom: 2.5%;
  right: 2.5%;
  background-color: #39424B;
}

.NeueNoteBox {
  border: solid #02D2BF;
  border-radius: 10px;
  padding: 5%;
  margin: 5%;
  color: #FFFFFF;
  background-color: transparent;
  width: 79%;
}

.LabelOben {
  position: relative;
  top: 1.75em;
  left: 5%;
  background-color: #39424B;
  padding: 1%;
  color: #02D2BF;
}

.LabelUnten {
  position: relative;
  top: -1em;
  left: 2.5em;
  color: #02D2BF;
  font-size: 75%;
}

.NeueNoteReset {
  cursor: pointer;
  width: 40%;
  font-weight: bold;
  color: #FFFFFF;
  border: none;
  background-color: #FF5757;
  border-radius: 10em;
  padding: 5%;
  margin: 5%;
}

.NeueNoteSubmit {
  cursor: pointer;
  font-weight: bold;
  width: 40%;
  color: #FFFFFF;
```

```
border: none;
background-color: #02D2BF;
border-radius: 10em;
padding: 5%;
margin: 5%;
}
```

```
/*Übersicht Ordner*/
```

```
.Kurs-Overview {
padding: 1%;
background-color: #39424B;
position: absolute;
left: 2.5%;
top: 2.5%;
border-radius: 10px;
}
```

```
.Teiler {
border: 1px solid white;
width: 1%;
margin: 2%;
}
```

```
.Lehrer-Semester {
background-color: #39424B;
position: absolute;
top: 15%;
left: 2.5%;
border-radius: 10px;
padding: 2%;
line-height: 200%;
}
```

```
.PencilIcon {
color: #FFFFFF;
background-color: #02D2BF;
position: absolute;
top: -0.5em;
right: -0.5em;
border-radius: 50%;
padding: 2.5%;
}
```

```
.user-block {
background-color: #39424B;
position: absolute;
top: 2.5%;
right: 2.5%;
border-radius: 10px;
padding: 1%;
cursor: pointer;
}
```

```
.UserIcon {
vertical-align: middle;
}
```

```
/*Tabelle CSS*/
```

```

.containerDiv {
    bottom: 2.5em;
    left: 2.5em;
    width: 60%;
    position: absolute;
    padding-top: 1em;
    border-top: 0.15em solid #02D2BF;
}

.Tabelle {
    width: 100%;
    background-color: #FFFFFF;
    border-spacing: 0.1em;
    table-layout: fixed;
    border-radius: 0.2em;
}

.Tabelle td, th {
    color: #FFFFFF;
    background-color: #8EA1B4;
    padding: 1%;
    text-align: center;
}

```

PHP-Code einfügen

Um die nächsten drei Abschnitte erfolgreich meistern zu können, muss noch bevor der Code geöffnet wird folgende Zeile eingefügt werden:

```

<?php // Verbindung zur Datenbank aufbauen
include "verbindung_db_notentracker.php"; ?>

```

Damit binden wir die Datei `verbindung_db_notentracker.php` ein.

Aktuell existiert diese Datei noch nicht. Erstelle also eine neue Datei in gleichen Verzeichnis wie die Datei `index.php` mit den Namen `verbindung_db_notentracker.php`

Zu Beginn bestimmen wir, aus welcher Datenbank Daten ausgelesen werden. Dafür verwenden wir vier Variablen: `$servername`, `$username`, `password` und `$database_in_use`.

Da wir nur einen lokalen Webserver betreiben ist der `$servername` unser eigener Computer. Das wird mit Hilfe von `localhost` angegeben.

Um auf den Datenbankserver zugreifen zu können, brauchen wir wie bei fast jedem Computer einen Benutzernamen (`$root`) und ein Passwort (`$usbw`). Der Benutzer `root` ist der Administrator-Benutzer der Datenbank. Dieser kann alles in der Datenbank bearbeiten oder löschen. Deshalb wird bei einem richtigen Webserver auch immer ein anderer Benutzer mit weniger Rechten angelegt. Für unsere Zwecke reicht das aber :-)

Zum Schluss definieren wir noch, auf welche Datenbank das Programm zugreifen soll (`database_in_use`). In unserem Fall ist das die Datenbank `notentracker`.

```
<?php
// Variablen für Verbindung
$servername = "localhost";
$username = "root";
$password = "usbw";
$database_in_use = "notentracker";
```

Anschließend wird die Funktion `mysqli_connect` aufgerufen. Wir übergeben die gerade eben festgelegten Parameter. Das "Ergebnis" speichern wir in der Variable `mysqli`. Damit ist die Verbindung zur Datenbank aufgebaut.

```
// Verbindung aufbauen
$mysqli = mysqli_connect($servername, $username, $password, $database_in_use);
```

Zum Schluss wird geprüft, ob die Verbindung funktioniert. Sollte das nicht der Fall sein, wird eine Fehlermeldung ausgegeben.

```
// Verbindung überprüfen
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
?>
```

Jetzt haben wir eine Verbindung zu unserer Datenbank und können mit Hilfe der Variable `mysqli` Anfragen an diese schicken.

1. Durchschnitt

Als erstes wollen wir den Durchschnitt aller Noten in diesem Kurs bestimmen und ausgeben.

Dafür bauen wir im HTML-Code einige Zeilen PHP-Code ein.

Um den Durchschnitt zu berechnen, benötigen wir alle in der Tabelle aufgeführten Punkte:

```
$sql = "SELECT Punkte FROM alles where 1";
```

Diesen String speichern wir in der Variable `$sql`. In der nächsten Zeile übergeben wir `$sql` an die Funktion `mysqli_query()` zusammen mit dem "Ergebnis" `$mysqli` der Datenbankverbindung aus der Datei `verbindung_db_notentracker.php`.

```
$result = mysqli_query($mysqli, $sql);
```

Die Ausgabe der Funktion wird in `$result` gespeichert. Da es sich dabei um ein assoziatives Array handelt, müssen wir es erst noch mit einer weiteren Funktion "auseinandernehmen" (`fetch_assoc()`). Danach speichern wir Schritt für Schritt mit einer while-Schleife alle Punkte aus jeder Reihe in `$punkte_ges`. Gleichzeitig speichern wir die Anzahl der Reihen in der Variable `$anzahl`

```
while($row = $result->fetch_assoc()){
    $anzahl = $anzahl+1;
    $punkte_ges = $punkte_ges + $row["Punkte"];
}
```

Nun können wir ganz einfach den Durchschnitt berechnen:

```
// Durchschnitt errechnen
$durchschnitt = ($punkte_ges / $anzahl);
```

Wenn der Durchschnitt so ausgegeben würde, würde ein Kommazahl mit vielen Stellen dabei herauskommen. Mit der PHP-Funktion `round()` können wir einstellen über den Parameter `$precision`, dass nur zwei Stellen nach dem Komma ausgegeben werden sollen:

```
echo "Durchschnitt: " .round($durchschnitt, $precision=2 ). "<br>";
```

Mit `echo` geben wir nun das Ergebnis unserer Berechnung aus.

Der gesamte Abschnitt sieht wie folgt aus:

```
<div class="UebersichtBox">

    <?php // Durchschnitt von allen Punkten in Datenbank berechnen
    $sql = "SELECT Punkte FROM alles where 1";
    $result = mysqli_query($mysqli, $sql);
    while($row = $result->fetch_assoc()){
        $anzahl = $anzahl+1;
        $punkte_ges = $punkte_ges + $row["Punkte"];
    }
    // Durchschnitt errechnen
    $durchschnitt = ($punkte_ges / $anzahl);

    // Durchschnitt aller Noten anzeigen und auf die zweite Stelle runden
    echo "Durchschnitt: " .round($durchschnitt, $precision=2 ). "<br>";
    ?>
</div>
```

2. Tabelle mit allen Schülern anzeigen

Auch hier fügen wir wieder ein kleinen Abschnitt PHP in das HTML-Gerüst ein.

Da wir alle Schüler in einer Tabelle ausgeben wollen ist das SQL-Statement recht einfach:

```
<?php
$sql = "SELECT * FROM alles";
```

Auch diese Anfrage schicken wir an die Datenbank und speichern das Ergebnis in `$result`:

```
$result = mysqli_query($mysqli, $sql);
```


Beim Anzeigen des Ergebnis verfahren wir ähnlich wie in 1. Das assoziative Array wird Schritt für Schritt mit Hilfe der while-Schleife auseinandergenommen und angezeigt:

```
while ($row = mysqli_fetch_object($result)) {
    echo "<tr>";
    echo "<td>".$row->Vorname . "</td>";
    echo "<td>".$row->Nachname . "</td>";
    echo "<td>".$row->Punkte . "</td>";
    echo "<td>".$row->Datum . "</td>";
    echo "<td>".$row->Art_der_Leistung . "</td>";
    echo "</tr>";
} ?>
```

3. Neuen Eintrag hinzufügen

Dafür verwenden wir das Eingabefeld im HTML-Code. In der ersten Zeile ist aufgeführt, was passieren soll, wenn das Feld abgeschickt wird; die Datei `eintrag-hinzufügen.php` wird aufgerufen. Es sollte also eine neue Datei mit diesem Namen im selben Verzeichnis wie die `index.php` erstellt werden. Jetzt beschäftigen wir uns damit, was da hinein kommt.

Auch hier müssen wir zunächst eine Verbindung zur Datenbank herstellen:

```
<?php
include "verbindung_db_notentracker.php";
```

Bei der Übertragung von Daten gibt zwei Modi: `Get` und `Post`. Diese entscheiden, wie die Daten von der Website zur Datenbank gelangen. Wir verwenden für dieses Beispiel `Get`, da man hierbei in der URL sehen kann, welche Daten übertragen werden. Für eine richtige Website ist das natürlich nicht empfehlenswert... Da `Get` der Standardfall für HTML-Forms ist, brauchen wir auf Seiten des HTML-Codes keine weiteren Einstellungen zu machen.

Beim Auslesen der Parameter für die Datenbank ist es natürlich wichtig. Um die einzelnen Daten in ein SQL-Statement verpacken zu können, ist es sinnvoll sie in einer Variable zwischen zu speichern. In diesen Variablen speichern wir mit Hilfe des `Get`-Parameters die einzelnen Daten:

```
$neuer_vorname_eingabe = $_GET["neuer_vorname"];
$neuer_nachname_eingabe = $_GET["neuer_nachname"];
$neue_punkte_eingabe = $_GET["neue_punkte"];
$neues_datum_eingabe = $_GET["neues_datum"];
$neue_art_eingabe = $_GET["neue_art"];
```

In diesem Fall wollen wir keine Daten abfragen sondern welche der Datenbank übergeben. Wir arbeiten also mit dem SQL-Befehl `INSERT INTO` Trotzdem ist das Vorgehen zu den ersten beiden Fällen sehr ähnlich:

```
// SQL-Anfrage
$sql = "INSERT INTO `alles` (`Vorname`, `Nachname`, `Datum`, `Punkte`,
`Art_der_Leistung`) VALUES ('$neuer_vorname_eingabe', '$neuer_nachname_eingabe',
'$neues_datum_eingabe', '$neue_punkte_eingabe', '$neue_art_eingabe' );";
$result = $mysqli->query($sql);
?>
```

Ab diesem Zeitpunkt sollte es nun möglich sein, neue Daten in die Tabelle über die Seite einzutragen. Noch muss man jedoch manuell auf die Ursprungsseite zurückkehren. Wir wollen aber, dass der Nutzer im besten Fall garnicht merkt, dass er gerade auf einer andere Seite war.

Dafür öffnen wir unterhalb unseres PHP-Code nun ein ganz normales HTML-Dokument. Wir binden auch die Datei `index.css` ein, damit es keinen großen Farbunterschied gibt.

```
<!DOCTYPE html>
<html lang="de" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Gespeichert</title>
    <!--bindet CSS-Datei ein-->
    <link rel="stylesheet" href="index.css">
  </head>
  <body>

  </body>
</html>
```

Damit ein Nutzer jedoch umgeleitet wird ist noch eine kleine Änderung nötig. Im des Dokuments fügen wir

```
<!--Leitet User auf die Homepage-->
<meta http-equiv="refresh" content="0; URL=http://localhost:8080">
```

ein.

Das sorgt dafür, dass sich die Seite unmittelbar nach dem öffnen neu lädt. Mit dem Parameter `URL=http://localhost:8080` von `content` wird jedoch nicht die aktuelle sondern die angegebene Seite neu geladen. Damit sieht es für den Benutzer so aus, als ob die Seite nie verlassen worden wäre.