



LERNJOURNAL

IMS Modularbeit M151



TOBIAS ODERMATT
IMS I2A

Inhalt

Modularbeit M151 ToDo Verwaltungssystem	2
Datenbankaufbau	2
Errichtung und Implementierung der Datenbankstruktur.....	2
Erstellen des Admin-user	2
Webseitengrundstruktur / Grundfunktionalitäten.....	2
Login Seite implementiert.....	2
Bootstrap Admin Grundgerüst errichtet.....	2
Grundgerüst der Webseitenordnerstruktur aufgebaut.....	2
Weiterleitung nach Login Session.....	3
User anlegen Formular in Admin Grundgerüst implementiert	3
Abgabe des geforderten ERM.	3
ERM Abgeben über E-Mail 07.02.2022.....	3
Adminpanel Aufbau	3
Anzeige der Nutzer Funktion implementiert.....	3
Buttons zur bearbeitung und löschung eines Nutzers implementiert	4
Zwischen Ergebnis der «Admin Page Nutzer anzeigen».....	4
Verbesserung des Fehlermanagement beim Löschen	4
Kategorien erstellen/bearbeiten/löschen/anzeigen System implementiert	4
Zwischenstand der Kategorie anlegen Funktion sowie Kategorie anzeigen Funktion.....	5
Administrative Aufgaben	5
Veränderung der Datenbank aufgrund eines Bewertungsschema.....	5
Weitere Kommentierungen in meinem Code hinzugefügt.....	5
Adminpanel zuweisen Funktion	6
Erste Implementierung der Zuweisungen V1.....	6
Zweite Implementierung der Zuweisungen V2	6
Userpanel Grundfunktion	8
ToDo anzeigen welche der Nutzer sehen darf:	8
ToDo Reihenfolge richtig anzeigen	8
ToDo erstellen:.....	8
Todos Bearbeiten/Löschen:	9
Archivieren Funktion:	9
Datum richtig berechnen:	10
Suchmaske:.....	10
ToDo Beschreibung:	11

Modulararbeit M151 ToDo Verwaltungssystem

Datenbankaufbau

Errichtung und Implementierung der Datenbankstruktur 31.01.2022 09.00-11.00

Das ERM wurde fertiggestellt. Nach dem Versuch das ERM in meine DB zu importieren, kam es zu einem Fehler. Ich brauchte ca. 30 Minuten, um anschliessend herausfinden, dass das Problem durch das Erstellen eines Indexs ausgelöst wird. Das Probleme konnte durch das Löschen dieser genannten Zeilen behoben werden. Die Datenbankstruktur wurde dann später erfolgreich in MySQL implementiert. Ich lernte dadurch das Problemlösung auch darin bestehen kann, vorhanden Zeilen zu löschen.

Erstellen des Admin-user 31.01.2022 11.00-12.00

Admin-user in die DB hinzugefügt. Das Password hashte ich provisorisch in einer PHP File. Dieser Hash habe ich dann per «INSERT» Operation mit den anderen nötigen Attributen in die DB hinzugefügt. Dabei lernte ich nicht neues dazu. Mir war klar, dass hier eine Provisorische Lösung die beste Lösung ist, da ich nur ein geshashten String brauche, um den Admin Nutzer anzulegen.

Webseitengrundstruktur / Grundfunktionalitäten

Login Seite implementiert 01.02.2022

Dazu kopierte ich meinen vorhandenen Code, den ich eine Woche zuvor geschrieben habe. Der Code besteht zum einen Teil aus einem Part, den ich selber geschrieben habe, sowie einen kopierten Part von OneNote, der dazu dient, meine Formfelder Server- sowie Clientseitig zu validieren.

Login

Bitte melden Sie sich mit Benutzernamen und Passwort an.

Benutzername *

Gross- und Kleinbuchstaben, min 6 Zeichen.

Password *

Gross- und Kleinbuchstaben, Zahlen, Sonderzeichen, min. 8 Zeichen, keine Umlaute

Senden

Löschen

Bootstrap Admin Grundgerüst errichtet 02.01.2022

Grundgerüst für UI Adminpanel in Bootstrap erstellt. Ich habe dazu das Beispielbild der Modulararbeit in OneNote angesehen und überlegt, wie ich meine Webseite designtechnisch strukturiere.

Login-Seite

Zum Login

Grundgerüst der Webseitenordnerstruktur aufgebaut 02.01.2022

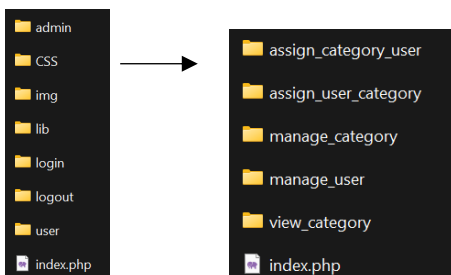
Ich habe mir Gedanken gemacht, welche PHP Dateien ich für diese Modulararbeit brauche. Ich bin zum Entschluss gekommen, dass es wenig Sinn ergeben würde, wenn ich schon die vollständige Ordnerstruktur entwerfen würde. Dies ist dem Grund geschuldet, dass ich von meiner Arbeitsweise her, Stück für Stück herangehen möchte. Ich möchte Multitasking vermeiden, deshalb habe ich lediglich den Hauptordner, sowie den Admin Ordner strukturiert. Die Überlegungen konnte ich nur treffen, da ich mir schon akribisch Gedanken zum Design und der Funktionalität des Admin Panels gemacht habe.

Bild 1 Ordnerstruktur:

M151_Modulararbeit_Tobias_Odermatt/

Bild 2 Ordnerstruktur

M151_Modulararbeit_Tobias_Odermatt/admin



Weiterleitung nach Login Session 03.01.2022

Nutzer werden nun entweder nach dem Login auf die Admin Page oder User Page weitergeleitet. Dies habe ich mit Session schnell realisiert. Nach dem Login wird die Rolle des Benutzers zur Session gesetzt. Dann leite ich den Nutzer, vom Login Skript aus, auf die Admin/User Seite um. In dem Admin und in dem User Bereich wird auch da die Session überprüft. Ist die Berechtigung ungültig, wird der User/Admin auf die Startseite weitergeleitet.

User anlegen Formular in Admin Grundgerüst implementiert 04.01.2022 14.00-18.20

Ich habe dem Admin die Möglichkeit geben, einen Nutzer zu erstellen. Das Fehlermanagement wurde hier schon mit einbezogen. Dies mit einer client- und serverseitigen Validierung. Sowie wird abgefragt, ob der Nutzernamen bereits existiert. Ist jenes der Fall, bekommt der Admin dies als Fehlermeldung zurück.

Nutzer hinzufügen.

Hier können Sie einen neuen Nutzer anlegen.

Vorname *

Nachname *

Benutzername *

Password *

Abgabe des geforderten ERM.

ERM Abgegeben über E-Mail 07.02.2022

Adminpanel Aufbau

Optimierung und Erweiterung des Adminsusermanagement 07.02.2022 09.00-11.00

Der Admin kann nun auch Benutzer bearbeiten sowie löschen. Entweder wird bei einem GET-Request das Attribut «ID» mit der User ID mitgeben, dann kann der Benutzer bearbeitet werden oder es wird das Attribut «DEL» mitgeben, dann wird der Nutzer gelöscht. Wurde ein Nutzer gelöscht, wird der Admin automatisch wieder auf «Index.php» weitergeleitet. Selbstverständlich wurden hier alle Fehler abgefangen. Bevor die ID bearbeitet oder gelöscht wird, überprüfe ich, ob ein Nutzer unter dieser ID existiert, falls nicht wird dieser Fehler abgefangen. Ich bin bis jetzt sehr zufrieden, wie es mit meiner aktuellen Arbeitsweise läuft. Ich finde, dass ich schnell vorankomme und habe bis jetzt noch genug Motivation weiter zu arbeiten und meine eigenen Arbeiten zu Reflektieren und zu optimieren. So habe ich beispielweise heute in CSS und HTML sicherlich über eine Stunde gebraucht, um die Webseite optisch ansprechender zu gestalten.

Anzeige der Nutzer Funktion implementiert 08.02.2022 16.30-20.30

Damit die Nutzer, die erstellt worden sind, angezeigt werden können, führe ich auf der Startseite ein SQL Statement aus, damit alle relevante Nutzerinformationen in einem Assoziatives Array ersichtlich werden. Danach loope ich durch das Assoziatives Array und erzeuge jedes Mal ein Element für die Tabelle, sowie Buttons, die auf die «manage_user page» weiterleiten. Mit den Attributen «DEL» / «ID» können die Benutzer entweder gelöscht oder bearbeitet werden, auch hier wurde der Fehler abgefangen, dass falls kein Nutzer existiert, die Tabelle leer bleibt. Wurden keine Nutzer gefunden, wird dies angezeigt. Es werden nur Nutzer der Rolle «User» angezeigt, dies habe ich durch ein IF-Statement einfach realisiert. (Bild zum Skript unten)

Buttons zur Bearbeitung und Löschung eines Nutzers implementiert 08.02.2022 16.30-20.30

Wie genau die Klassen in Bootstrap von einer «schöne» Tabelle benannt sind, konnte ich durch Internet recherchieren herausfinden. Mir war von Anfang an klar, dass ich für jedes einzelne Element Buttons hinzufügen werde, da ich in meiner zuvorhigen Modulararbeit auch für jedes einzelne Element ein Button erstellt habe für das Löschen und Bearbeiten.

```
function viewUSER($conn){
    $userlist = $conn->query("SELECT ID,firstname,lastname,username,role FROM user");
    while($row = $userlist->fetch_array()){
        if($row['role'] == 'admin'){continue;}
        echo "<tr>";
        "<td>". $row['ID']. "</td>";
        "<td>". $row['username']. "</td>";
        "<td>". $row['firstname']. "</td>";
        "<td>". $row['lastname']. "</td>";
        "<td style='width: 20%;>";
        "<a href='\"assign_user_category?ID=".$row['ID']."'>\" class='\"table-link'>";
        "<span class='\"fa-stack'>";
        "<i class='\"fa fa-square fa-stack-2x'>";
        "<i class='\"fa fa-tag fa-stack-1x fa-inverse'>";
        "</span>";
        "<a href='\"manage_user?ID=".$row['ID']."'>\" class='\"table-link'>";
        "<span class='\"fa-stack'>";
        "<i class='\"fa fa-square fa-stack-2x'>";
        "<i class='\"fa fa-pencil fa-stack-1x fa-inverse'>";
        "</span>";
        "<a href='\"manage_user?DEL=".$row['ID']."'>\" class='\"table-link danger'>";
        "<span class='\"fa-stack'>";
        "<i class='\"fa fa-square fa-stack-2x'>";
        "<i class='\"fa fa-trash-o fa-stack-1x fa-inverse'>";
        "</span>";
        "</td>";
    }
}
```

```
<table class="table table-striped table-hover">
  <thead>
    <tr>
      <th>ID</th>
      <th>Nutzername:</th>
      <th>Vorname:</th>
      <th>Nachname:</th>
      <th>Aktion:</th>
    </tr>
  </thead>
  <tbody>
    <?php viewUSER($conn);?>
  </tbody>
</table>
```

Auf dem ersten Bild sieht man die Funktion, die alle Nutzer auflistet, indem HTML Eintrags Elemente erstellt werden. Auf dem zweiten Bild sieht man, die vorgegebenen Kategorien eines Nutzers.

Zwischenergebnis der Admin Page Nutzer anzeigen

Herzlich Willkommen Administrator #1				
ID:	Nutzername:	Vorname:	Nachname:	Aktion:
43	UserTest	Test Nutzer 1 Vorname	Test Nutzer 1 Nachname	 
44	UserTestZwei	Zweiter User Test Vorname	Zweiter User Test Nachname	 

Verbesserung des Fehlermanagement beim Löschen 08.02.2022 21.00-21.30

Bei dem aktuellen Löschen wurde der Nutzer wieder auf die Administratorseite weitergeleitet. Der Admin bekam keine Rückmeldung, ob der Benutzer gelöscht wurde. Mir gefiel dies nicht. Dieses Problem konnte ich einfach lösen. Versucht man ein Benutzer zu löschen, wird ein GET Request zur Startseite gesendet, mit dem Attribut «Status» und dem Wert «delsucces» oder «delfail».

```
if(isset($_GET['status'])){
    if($_GET['status'] == "delsuccess"){
        $alert_class_name = "alert alert-success";
        $alert_message = "Der Benutzer wurde erfolgreich gelöscht.";
    }
    else if($_GET['status'] == "delfail"){
        $alert_class_name = "alert alert-danger";
        $alert_message = "Der Benutzer existiert nicht.";
    }
}
```

Nun wird der Admin informiert, ob das Löschen erfolgreich war.

Kategorien erstellen/bearbeiten/löschen/anzeigen System implementiert

09.02.2022 15.00-17.00

Das System für das Erstellen/Bearbeiten/Löschen/Anzeigen der Kategorie, dauerte sehr schnell, da der Grundstein der Funktionen durch das Programmieren der PHP File «manage_user/index.php» schon gelegt worden war. In kürzester Zeit war die Funktion implementiert. Ich konnte viele Funktion von dem einen, in das andere PHP File kopieren. Ich habe auch mein Fehlermanagement mit kopiert, zum Beispiel die Abfrage, ob dieser Kategorie Namen schon existiert. das Auflisten sowie bearbeiten der Kategorie war in weniger als einer halben Stunde schon implementiert.

Zwischenstand der Kategorie anlegen Funktion sowie Kategorie anzeigen Funktion

Home | Benutzerverwaltung | Kategorien ansehen | Kategorie erstellen | Benutzer erstellen | Log out

Kategorie hinzufügen.

Hier können Sie eine neue Kategorie anlegen.

Kategorienname *

Geben Sie eine Kategorie ein.

Senden Löschen

Adminpanel / Benutzerverwaltung

Hier können alle Kategorien eingesehen werden.

ID:	Name:	Aktion:
35	Test_Kategorie_1	
36	Test_Kategorie_2	

Info: Existiert schon eine Kategorie unter diesem Namen wird dies als Fehler abgefangen siehe Bild.

Kategorie existiert bereits.

Administrative Aufgaben

Hinzufügung von Kommentierungen in meinem Code 13.02.2022 18.00-19.00

Ich habe meine Skripte durch Kommentare leesbarer gemacht. Zudem hab ich die Namenskonventionen in meinen Skripten überarbeitet, um die Lesbarkeit zu verbessern. Ich habe bemerkt, dass ich mit der Zeit immer besser werde Variabel, einen «passenden» Namen zu geben.

Veränderung der Datenbank aufgrund eines Bewertungsschema 14.02.2022 08.00-11.00

Heute habe ich das Bewertungsschema durchgelesen, da ist mir folgender Punkt aufgefallen:

o Können mehr als 255 Zeichen in die Beschreibung der TODOs eingegeben werden?

Das stellt für meine aktuellen DB ein Problem dar, da ich davon ausgegangen bin, dass die Beschreibung eines ToDOs maximal 45 Zeichen beanspruchen wird. Dieser Fehler konnte ich in MySQL schnell beheben. Durch das Ändern des Datentyp VARCHAR Länge, konnte ich dieses Problem lösen. Der VARCHAR ist nun auf 300 Zeichen begrenzt.

Weitere Kommentierungen in meinem Code hinzugefügt 15.02.2022 17.00-18.00

Ich habe ungefähr 3 Funktionen unbenannt, da ich nicht zufrieden mit der Namensgebung war. Zudem habe ich 2 Variablen gefunden die Fehlerhaft benannt worden waren. Diese Fehler habe ich direkt ausgebessert und meine Skripte durch Kommentierungen ergänzt.

Adminpanel zuweisen Funktion

Erste Implementierung der Zuweisungen V1 16.02.2022 19.00-21.00

Ich möchte mich nun der Funktion für die Zuweisungen widmen, dazu habe ich schon einige Vorbereitung getätigt. Ich habe bei der Funktion des Anzeigens der Benutzer/Kategorien jeweils ein Button bei jedem Element eingebaut, dass es ermöglicht, die ID des behandelten Elements mittels GET-Request an das Script «assign_user_category» oder an das Script zu senden assign_category_user.

Herzlich Willkommen Administrator #1				
ID:	Nutzername:	Vorname:	Nachname:	Aktion:
5	CreepyTobi	Tobias	Odermatt	  
6	Tobias	Tobias	Odermatt	  

In diesen zwei Bildern sieht man, dass man einem Nutzer eine Kategorie hinzufügen kann. Das Ganze geht natürlich auch umgekehrt. ich habe mein Script einfach kopiert und verändert. Einer Kategorie kann auch unter gleichem Layout, ein Nutzer hinzugefügt werden anhand dem Nutzernamen oder ID.

In diesem Bild sieht man den Button. Wird der Button gedrückt, wird man auf diese Seite weitergeleitet.

Kategorie dem Nutzer hinzufügen.

Klicken Sie die Kategorien an, die Sie dem Nutzer #6 hinzufügen möchten

Kategorienname *

Geben Sie den Kategorienamen oder die Kategorien ID ein zum bsp: #32.

Die Zuweisung funktioniert so, dass wenn der eingegebene String ein «#» enthält, wird das Hashtag entfernt und der String wird in ein Integer umgewandelt. Der Integer wird dann in die Tabelle «user_has_category» geschrieben, zusammen mit der ID des zuvor angeklickten Elements, der per GET-Request vorher an das Skript übertragen wurden.

Wird ein String in der Form von dem Kategorienamen/Benutzernamen eingegeben wird die ID in der Datenbank dazu gesucht und dann auch zusammen mit der ID des zuvor angeklickten Elements, der per GET-Request, vorher an das Skript übertragen wurden.

Diese Lösung gefiel mir nicht, da jeder Benutzer oder Kategorie von Hand einzeln eingegeben werden musste. Ich wollte ein System, dass es ermöglicht, mehrere Kategorien einem Nutzer zuzuweisen oder mehrere Nutzer eine Kategorie zuzuweisen.

Mir gefiel es auch nicht, dass es mit der «#» ID Eingabe Funktion zu Fehlern führen hätte können, falls ein Nutzernamen oder Kategorie Name ein «#» enthalten würden.

Durch das Abwiegen dieser Nachteile, entschloss ich mich das ganze Zuweisen System nochmal vollständig umzuschreiben, doch dies mache ich erst Morgen da es schon spät ist.

Zweite Implementierung der Zuweisungen V2


17.02.2022.18.00-23.00

Das Zuweisungssystem

Nach vielen Überlegungen über die genaue Umsetzung dieser Funktion, entschied ich mich, dass es am meisten Sinn machen würde, wenn für jeden Nutzer oder jede Kategorie eine eigene Checkbox erstellt wird, die angewählt werden kann.

Aufgrund der vorliegenden Funktion für das Anzeigen von Benutzer und Kategorien, wusste ich direkt wie ich schnell automatisiert eine Tabelle in Bootstrap erstelle sowie die Checkboxes, nur wusste ich nicht wie eine Dynamische Anzahl von Variablen erstelle und deren die Werte einzeln auslese, doch erst einmal wollte ich die Checkboxes an sich implementieren.

Checkboxes Design

Mir war es wichtig das die Checkbox «schön» aussieht deshalb Recherchierte ich über Bootstrap Checkboxes V.3 und fand diese [Checkboxes](#) ->  dieser HTML & CSS Snippet habe ich kopiert.

```
<div class="material-switch pull-right">
  <input id="someSwitchOptionSuccess" name="someSwitchOption001" type="checkbox"/>
  <label for="someSwitchOptionSuccess" class="label-success"></label>
</div>
```

Werte auslesen/verwalten

Durch Recherche wie ich dieses Problem lösen konnte stoss ich auf diesen [stack overflow](#) Beitrag, dort lernte ich etwas entscheidendes, ich wusste nicht das das ich bei einem HTML Formular die Variabel (Name) ein Assoziatives Array zuweisen kann, in meinem Fall heisst es hier «SwitchForm[]», ich diese Variable durch loopen, es sind nur die Werte drin die ein Checkbox trägt falls sie gechecked wurde, ist das Array leer heisst das dass kein Benutzer angewählt wurden und alle Einträge in der Tabelle «user_has_category» mit der Benutzer ID werden gelöscht.

Die Werte die von dem Assoziativen Array mitgeliefert werden, werden in einem loop automatisch in die Tabelle «user_has_category» eingetragen. (siehe Bild 3)

Wurde ein Feld nicht angewählt wird diese Zuweisung gelöscht, dies geschieht durch die in_array Funktion, sie überprüft, ob die ID von der Datenbank in dem Assoziativen Array der Checkboxes als Wert vorhanden ist, wenn nicht dann wird die Zuweisung gelöscht.

```
while ($row = $query->fetch_array()) {
    echo "<tr>" .
        "<td>" . $row['name'] . "</td>" . //Benutzername Anzeige
        "<td><div class=\"material-switch pull-right\">" . //Switch für jeden Benutzer
        "<input id=\"" . $row["category_id"] . "\" name=\"SwitchForm[]\" " .
        " value=\"" . $row["category_id"] . "\" type=\"checkbox\"/>" .
        "<label for=\"" . $row["category_id"] . "\" class=\"label-success\"></label></div></td>" .
        "</tr>";
}
```

```
if (!empty($_POST['SwitchForm'])) {
    readCheckBox($_POST['SwitchForm'], $conn, $user_id);
} else {
    $alert_message = "Nutzer ist keiner Kategorie zugeordnet";
    uncheckEverything($conn, $user_id);
}
```

```
function readCheckBox($ArrayCheckbox, $conn, $user_id)
{
    global $alert_message; //Ruft die Zuweisungsfunktion auf
    foreach ($ArrayCheckbox as $key => $category_id) {
        assignUserCategory($user_id, $category_id, $conn);
        $alert_message = "Änderungen wurden gespeichert";
    }

    $query = $conn->query("SELECT category_id FROM category");
    if (!($query->num_rows == 0)) {
        while ($row = $query->fetch_array()) {
            if (!in_array($row["category_id"], $ArrayCheckbox)) { // Wurde ein Feld nicht angewählt wird die Zuweisung gelöscht
                delAssign($conn, $user_id, $row["category_id"]);
            }
        }
    }
}
```

Dieses Checkboxes Zuweisensystem hier zuprogrammieren war bis jetzt das schwerste für mich, doch ich lernte dadurch extrem viel, wie das ein Assoziative Array in einem Super Globalen Assoziative POST_Array gespeichert werden kann.

Ich bin mit meiner Überarbeitung zufrieden und bin der Meinung das dass Ergebnis sich bezahlt hat, ich finde die Mühe hat sich gelohnt, und ich habe wichtiges wissen dazu gewonnen.

Einem Nutzer ein oder mehrere Kategorien zuweisen.

Klicken Sie die Kategorien an, die dem Nutzer #5 hinzugefügt werden sollen

Kategorie Namen:	Auswahl:
Haushalt	<input checked="" type="checkbox"/>
IMS	<input type="checkbox"/>
Administration	<input checked="" type="checkbox"/>
Generell	<input type="checkbox"/>

Userpanel Grundfunktion

ToDo anzeigen welche der Nutzer sehen darf: 18.02.2022 15.00-16.00

Da ich die Seite die Todos erstellen soll noch nicht implementiert habe, habe ich Testweise ein ToDo per INSERT Operation Manuell in die Datenbank eingetragen, nun kann es losgehen mit der Implementierung der Anzeige.

Durch das mehrmalige Durchlesen des Bewertungsraster, habe ich mich dazu entschieden eine Funktion zu programmieren, die mir alle Todos IDs in ein Array schreibt, die der Nutzer sehen darf, also die Todos mit der der Nutzer eine Kategorie teilt.

```
function getAllowedToDoIDs($conn,$user_ID){
    $ToDoIDArray = [];
    $query = $conn->query("SELECT category_category_id FROM user_has_categories where user_ID = '$user_ID'");
    while($row = $query->fetch_array()){
        $category_id = $row["category_category_id"];
        $selectToDoID = $conn->query("SELECT ID from to_do where category_category_id = '$category_id'");
        while($toDO_ID = $selectToDoID->fetch_array()){
            array_push($ToDoIDArray,$toDO_ID["ID"]);} }
    return $ToDoIDArray;
}
```

ToDo Reihenfolge richtig anzeigen: 18.02.2022 15.00-16.00

Die Funktion für das Anzeigen der Daten war mir aufgrund der Erstellung des Adminpanel vertraut, ich musste mir nun nur Mühe geben das die Daten schön und in der richtigen Reihenfolge dargestellt werden, Dies habe ich durch ein einfach SQL query sichergestellt.

Die Funktion, die oben beschrieben wurde, konnte ich hier verwenden, ich bin übergücklich das meine Todos Fehler frei nun angezeigt werden können, wenn auch mit Verbesserungsbedarf des Aussehens.

```
//Erstellt ein ToDo Beitrag, ruft dazu alle nötigen Funktionen zum Anzeigen auf, wird nach Priorität und nach Datum sortiert.
function viewTodos($conn)
{
    $user_ID = $_SESSION['id'];
    $query = $conn->query("SELECT * FROM to_do where archive = 'false' order by priority DESC, finishDate ASC");
    $AllowedToDoIDArray = getAllowedToDoIDs($conn,$user_ID);
    while($row = $query->fetch_array()){
        if(!in_array($row['ID'],$AllowedToDoIDArray)){continue;};
    }
}
```

ToDo erstellen: 19.02.2022 14.00-18.00

Für das Generieren meiner ToDo erstell Formular, kopierte ich, um Zeit zu sparen mein user erstell Formular, und veränderte meine Attribute zu den Attributen des Todos, ich musste jedoch selbst noch eigene Funktionen dazu schreiben, die Beispielweise überprüfen ob dem Nutzer überhaupt Kategorien zugewiesen worden sind, dann eine Funktion welche Automatisch ein Dropdown Menü generiert welche es ermöglicht eine Kategorie anzuwählen, hier wollte ich schon vorausplanen und es schonmal möglich machen das wenn ein ToDo bearbeitet wird, standartmässig die von zuvor gewählte Kategorie angewählt wird.

```
//Macht aus der Kategorie ID ein Kategorie Namen
function getCategoryname($conn,$category_ID){
    $result = mysqli_query($conn, "SELECT Name from Category WHERE category_id = '$category_ID'");
    return mysqli_fetch_assoc($result)["Name"];
}

//Generiert eine Kategorie ID liste mit den Kategorien die der Benutzer erstellt hat.
function generateCategoryList($conn,$user_ID,$edit_category_ID){
    $query = $conn->query("SELECT category_category_id from user_has_categories where user_ID = '$user_ID'");
    while($row = $query->fetch_array()){
        $categoryID = $row["category_category_id"];
        $categoryname = getCategoryname($conn,$categoryID);
        if($edit_category_ID == $categoryID){
            echo "<option value=\"$categoryID\" selected>".$categoryname."</option>";
        }else{echo "<option value=\"$categoryID\">".$categoryname."</option>";}
    }
}
```

← Siehe PHP Code

Um das Datum anzuwählen, Recherchierte ich im Internet, sowie schaute mir den Code im OneNote dazu an, nach dem Einbinden des JQuery und des Schreibens einer Zeile HTML war dieses Form Feld schon fertig. Für die Priorität erstellte ich ebenfalls ein Dropdown Menü, dieses konnte aber statisch sein da die Prioritäten sich ja nie verändern.

Nun fehlte nur noch der Slider, dafür Recherchierte ich im Netz nach Slidern, nach ungefähr 10 Minuten wurde ich fündig, ich entschied mir diese 2 zwei Zeilen in meinem Code zu implementieren, das Grundgerüst war nun fertig.

Um das Skript auf das Erstellen von Todos anzupassen half es mir enorm CTRL + F die Variablen Namen vom vorigen Skript mit den aktuellen zu ersetzen.

Bild ist das Ergebnis: Ich bin über das Aussehen beziehungsweise die Usability sehr zufrieden.

ToDo hinzufügen.

Titel *
Geben Sie einen Titel ein maximal 45 Zeichen.

Beschreibung *
Geben Sie eine Beschreibung an maximal 300 Zeichen.

Priorität: 1

Kategorien:

Status des Todos:

ToDo's Bearbeiten/Löschen: 20.02.2022 18.00-19.00

Um das ToDo zu bearbeiten/löschen, programmierte ich zuerst eine Funktion, die überhaupt überprüft ob der Benutzer die Berechtigung hat ein ToDo zu bearbeiten/löschen, falls der Benutzer keine Berechtigung dazu hat wird dies per GET-Request als Fehler zurück an die Startseite gegeben, beim Bearbeiten war es mir wichtig das alle Werte in das Formular geladen werden.

Dadurch das ich ein bearbeite Funktion für das User Formular in PHP schon programmiert habe, ging das implementieren der ToDo bearbeiten Reibungslos von statten, und ich konnte mich direkt wieder auf andere Dinge konzentrieren

Archivieren Funktion: 20.02.2022 22.00-22.30

Wird ein ToDo (un)archiviert, wird ein GET-Request zu «manage_todo» gesendet, dort wird überprüft, der Eigentümer dieses ToDo wird überprüft, je nachdem wird es (un)archiviert oder ein GET-Request wird zur Startseite gesendet mit dem Status «archieved», «unarchieved», «permissionfail».

Die Funktion für das Archivieren funktionier wie ein Switch, falls das ToDo archiviert ist und die Funktion wird es gestartet wird es unarchiviert, oder falls es archiviert ist, wird es archiviert.

Ich bin zufrieden mit meiner Lösung und war wieder mal überrascht wie einfach ich das umsetzen konnte.

Datum richtig berechnen: 20.02.2022 23.00-23.30

Damit das Datum richtig berechnet wird, habe ich dazu selbstverständlich eine Funktion erstellt, mein Ziel war es, das das Datum entweder Grün oder Rot angezeigt wird, je nachdem ob das Datum schon war oder nicht, zudem soll nicht 0 Tage stehen, sondern «Heute», wurde ein ToDo erledigt sollte «Erledigt» stehen.

Unten sieht man meine Umsetzung

```
//Generiert den Fälligkeitstext, wurde ein ToDo archiviert steht nur "erledigt"
function getFinishDate_Text($FinishDate, $archieved = null) {
    $FinishTimeStamp = strtotime($FinishDate); $today = time();
    $difference = $FinishTimeStamp - $today;
    $difference = $difference/60/60/24+1; //Berechnung für wie viele Tage
    if($archieved = null){
        if ($difference > 1) { //Zukunft
            return "<p class=\"text-success\"> in ".(floor($difference))." Tagen </p>";
        }
        else if ($difference ≤ 1 && $difference > 0){ //Heute
            return "<p class=\"text-success\">Heute</p>";
        }
        else if ($difference < 0){ //Vergangenheit
            return "<p class=\"text-danger\">".(abs(floor($difference)))." Tagen! </p>";
        }
    }
    if($archieved = "true"){ return "<p class=\"text-success\">erledigt</p>"; } //Falls archiviert
}
```

Suchmaske: 21.02.2022 08.30-11.00

Im Informatikunterricht programmierte ich eine Suchmaske, Ziel der Suchmaske ist es Todos auf die der Nutzer Zugriff hat, nach den Attributen «Text» sowie «Title» zu durchsuchen, es sollten alle Todos gelistet werden, auch welche die Archiviert worden sind, dazu habe ich eine Searchbar oberhalb meiner ToDo Liste eingefügt.

Zudem kopierte ich meine Funktion «ViewTodos» und gab ihr den Namen «searchToDo», mit dem weiteren Parameter «\$searchString», werden nun Todos gesucht wird die Funktion «searchToDo» aufgerufen und der Validierte Suchtext der Funktion übergeben.

```
<table class="table table-striped">
    <thead>
        <tr>
            <th style = "width: 4%"></th>
            <th style = "width: 4%">ID:</th>
            <th style = "width: 8%">Priorität:</th>
            <th style = "width: 9%">Kategorie:</th>
            <th style = "width: 22%">Titel:</th>
            <th style = "width: 12%">Erstellt:</th>
            <th style = "width: 10%">Fällig:</th>
            <th style = "width: 16%">Status:</th>
            <th style = "width: 15%">Aktion:</th>
        </tr>
    </thead>
    <tbody>
        <?php if(empty($searchString)){viewTodos($conn);}
        else{searchToDo($conn, $searchString);}?>
    </tbody>
</table> </div>
```

Die Funktion «searchToDo» bereitet ein prepared SQL Statement vor bei dem der Inhalt der Todos sowie der Titel der Todos gesucht werden, dies konnte ich mit der SQL Operation Like erreichen, ich hatte zuerst mühe die Prozentzeichen richtig zu platzieren, doch mit dem einfach zusammenfügen vor des eigentlichen SQL Query funktionierte das System

ToDo Beschreibung: 22.02.2022 17.00-20.00

Um die Beschreibung eines ToDos anzuzeigen war es mir sehr wichtig auch hier Wert auf die Usability zulegen, ich war/bin überzeugt davon, dass sich dies hervorragend durch sogenannte «collapse elemente» umsetzen liesse, also recherchierte ich unter den Suchbegriffen («Bootstrap 3 + table + collapse») Schliesslich wurde ich fündig und ich fand ein Beispiel, das man live ausprobieren konnte [Link](#).

Das umsetzen erwies sich als schwieriger als gedacht, den durch das Aufklappen des Beschreibungstextes, veränderte sich auch das Layout meiner Tabelle stark, durch sehr lange Recherchen wie es verhindern konnte das ein Element eine Tabelle beeinflusst und durch das viele ausprobieren von CSS fand ich die Lösung. `.table{table-layout:fixed;}`

Diese CSS Eigenschaft verhindert das sich die Tabelle anders verhält als geplant.

```
$searchedTitle = "%".$SearchString."%";$searchedText = "%".$SearchString."%";
```