

SciRead – Eine Webanwendung zur Verwaltung wissenschaftlicher Literatur

Erstellt von: Jonas Kriefoth, Cem Cetin, Viktoria Lau, Kevin-Jörg Dreeßen

102202216, 102207269, 102204450, 102202394

Dozent: Prof.Dr-Ing.Kuhlen

Kurs: DSGWP042201

Gliederung

Einleitung

- Aufgabenstellung
- Lösung der Aufgabenstellung

Architektur der Anwendung

- Flussdiagramm der Seitenstruktur
- Übersicht und Erläuterung der Datenbank

Vorstellung der Website

- Übersicht der geplanten Implementierung
- Implementierung der Anwendung
- Live-Demo der wichtigsten Seiten
- Veranschaulichung und Erklärung des Codes

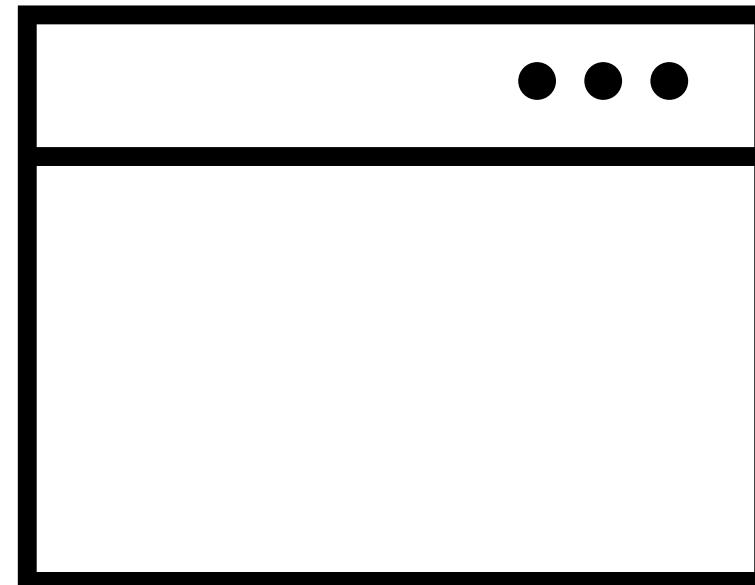
Reflektion

- Technologieentscheidungen
- Herausforderungen und Lösungen

Einleitung

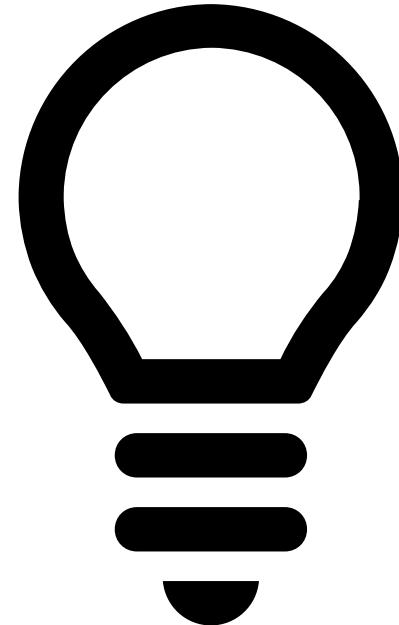
Aufgabenstellung

Entwicklung einer
benutzerfreundlichen
Webanwendung zur Verwaltung von
wissenschaftlicher Literatur.



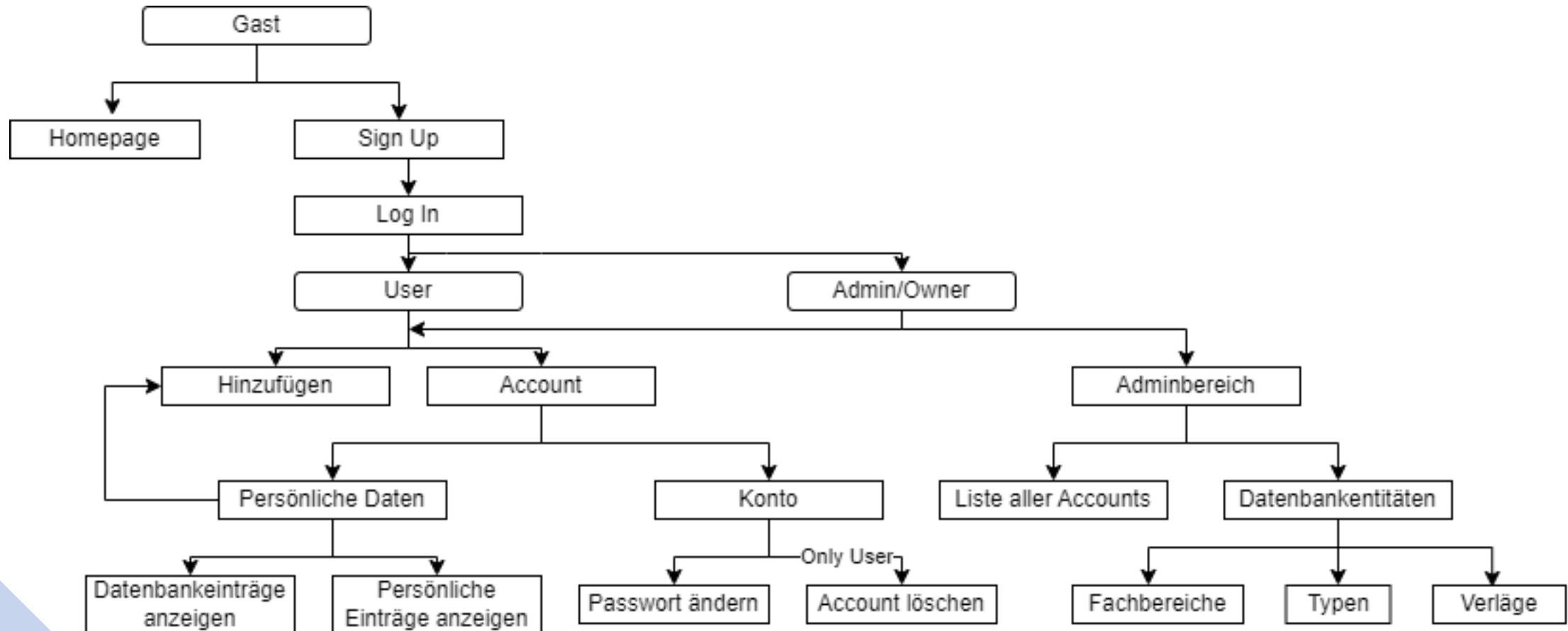
Lösung der Aufgabenstellung

- Implementierung eines Anmeldesystems
- Möglichkeit eigene Einträge zu erstellen
- Möglichkeit Einträge in eine Merkliste zu speichern

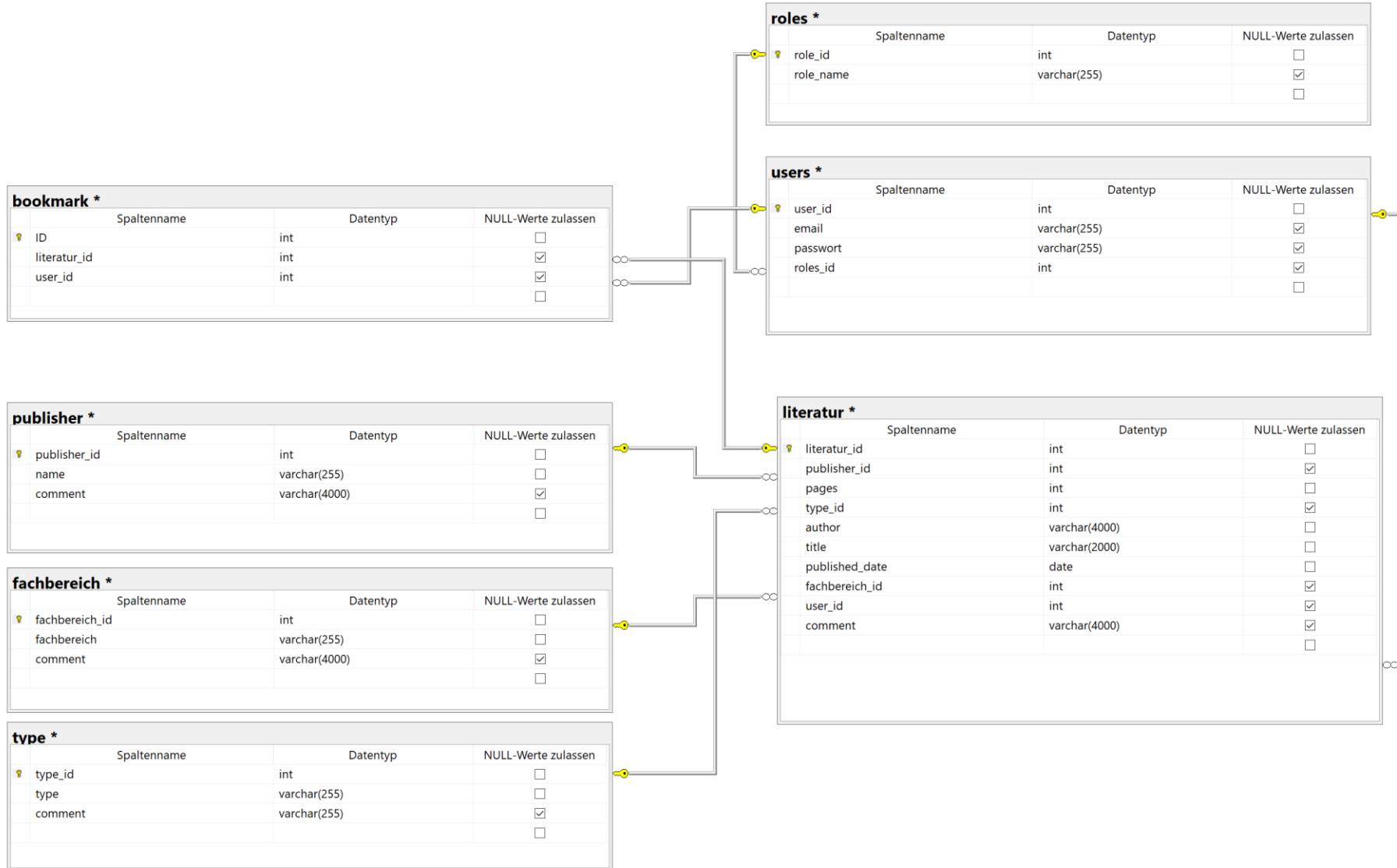


Architektur der Anwendung

Flussdiagramm der Seitenstruktur



Übersicht und Erläuterung der Datenbank



Vorstellung der Website

Übersicht der geplanten Implementierungen



Implementierung eines Anmeldesystems

- Möglichkeit dich zu Registrieren
- Möglichkeit dich Anzumelden
- Möglichkeit dein Passwort zu ändern
- Möglichkeit dein Account zu löschen
- Einträge zu erstellen und Lesezeichen für sich selber zu erstellen



Implementierung eines Administratorsystems

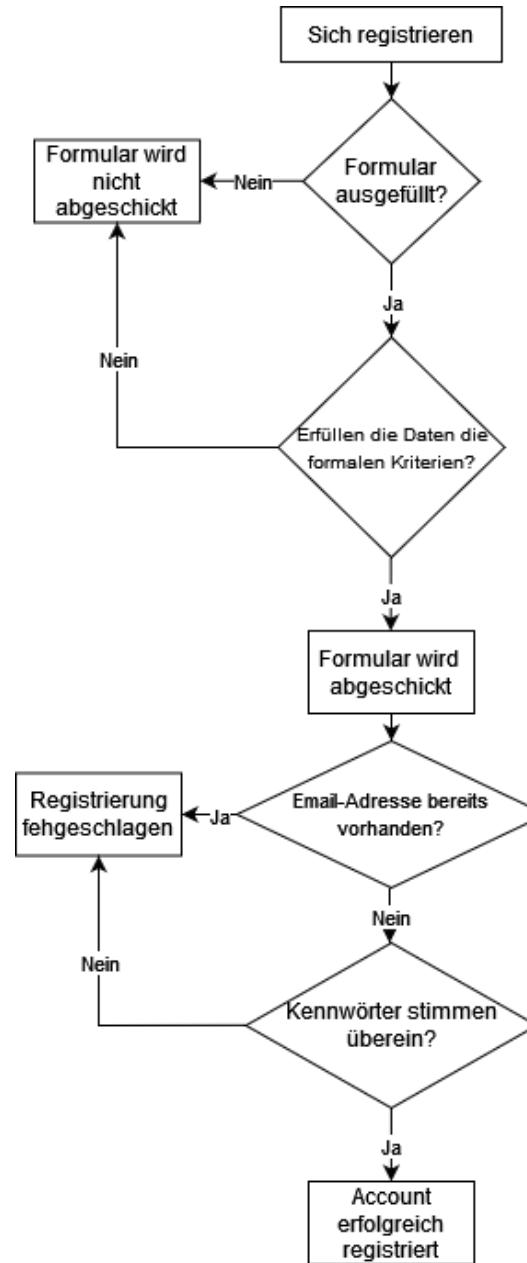
- Administratoren können fremde Einträge bearbeiten und löschen
- Administratoren können alle User sehen und deren Accounts löschen
- Der „Owner“ hat die Möglichkeit Administratoren zu ernennen, die Rechte wegzunehmen und zu löschen
- Administratoren können in bestimmte Tabellen weitere Einträge hinzufügen, löschen oder bearbeiten

Implementierung eines Anmeldesystems

Registrierung

Nutzen und Funktionen:

- Es wird ein INSERT-Befehl ausgeführt.
 - Einen Account erstellen und in der Datenbank abspeichern
 - Gespeicherte Passwort in der Datenbank, muss verschlüsselt sein
 - Keine doppelten Einträge erlauben



Wie wird für die Registrierung die Sicherheit gewährleistet?

- Das Passwort wird als „Hash“ in der Datenbank gespeichert
- Falls die Einträge in der Datenbank öffentlich werden, sind die Passwörter nicht direkt ersichtlich
- Vor SQL-Injection mit der prepare() Funktion und Parametern schützen
- Passwort muss mindestens 8-Zeichen lang sein

- Passwort hashen

```
1 $passwordHash = password_hash($password, PASSWORD_DEFAULT);
```

- Daten in die Datenbank schreiben

```
1 $sql = "INSERT INTO users (email, passwort, roles_id) VALUES (:email, :passwordHash, :roles_id)";
2 $stmt = $dbh->prepare($sql);
3 $stmt->bindValue(':email', $email);
4 $stmt->bindValue(':passwordHash', $passwordHash);
5 $stmt->bindValue(':roles_id', 2);
6 $success = $stmt->execute();
```

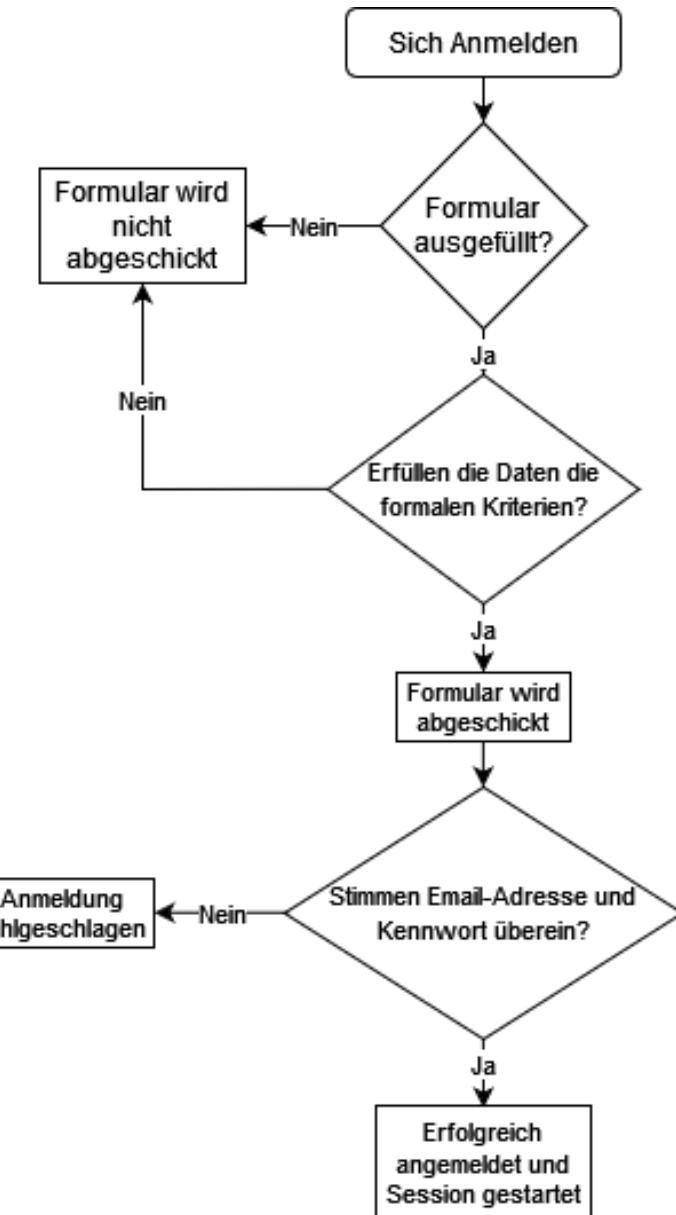
- Acht Zeichen Anforderung

```
1 <input type="password" id="kennwort" name="kennwort" class="form-control form-control-lg"
2           minlength="8" required pattern=".{8,}" title="Das Passwort muss mindestens 8 Zeichen lang sein."/>
```

Anmeldung

Nutzen und Funktionen:

- Es wird ein SELECT-Befehl ausgeführt.
 - Die Anmelddaten in der Datenbank suchen und verifizieren
 - Eine Session für den User starten
 - Wichtige Informationen für die Dauer der Session speichern



Wie wird für die Anmeldung die Sicherheit gewährleistet?

- Vor SQL-Injection mit der `prepare()` Funktion und Parametern schützen
- Um Brute-Force-Angriffe zu vermeiden, wird bei fehlgeschlagenem Log In die selbe Fehlermeldung ausgegeben
- Bei erfolgreichem Log In werden wichtige Daten abgespeichert, die zur weiteren Identifizierung dienen

- Daten in die Datenbank schreiben

```
1 // Nach Anmelddaten in der Datenbank suchen
2 $query = 'SELECT * FROM users WHERE (email = :email)'
3 // Fehler abfangen falls ein Fehler auftritt
4 try
5 {
6 // Query ausführen
7 $stmt = $dbh->prepare($query);
8 $stmt->bindValue(':email', $email);
9 $stmt->execute();
10 }
```

- Fehlermeldung ist die selbe

```
1 // fetch(PDO::FETCH_ASSOC) liefert ein Array mit Spaltennamen als Schlüssel zurück.
2 $row = $stmt->fetch(PDO::FETCH_ASSOC);
3 // Prüft ob ein Array zurückgegeben wurde
4 $isArray = is_array($row);
5 if (!$isArray)
6 {
7 // Kein Array zurückgegeben, Benutzer existiert nicht.
8 ?>
9 <script>
10 // Aus Sicherheitsgründen wir nur eine allgemeine Fehlermeldung ausgegeben
11 alert("Anmelddaten falsch!");
12 window.location.href = "page_login.php";
13 </script>
14 <?php
15 return;
16 }
17 // Überprüft ob das Passwort mit dem Passwort in der Datenbank übereinstimmt
18 $isPasswordCorrect = password_verify($password, $row['passwort']);
19 if (!$isPasswordCorrect)
20 {
21 // Passwort ist falsch.
22 ?>
23 <script>
24 // Aus Sicherheitsgründen wir nur eine allgemeine Fehlermeldung ausgegeben
25 alert("Anmelddaten falsch!");
26 window.location.href = "page_login.php";
27 </script>
28 <?php
29 return;
30 }
```

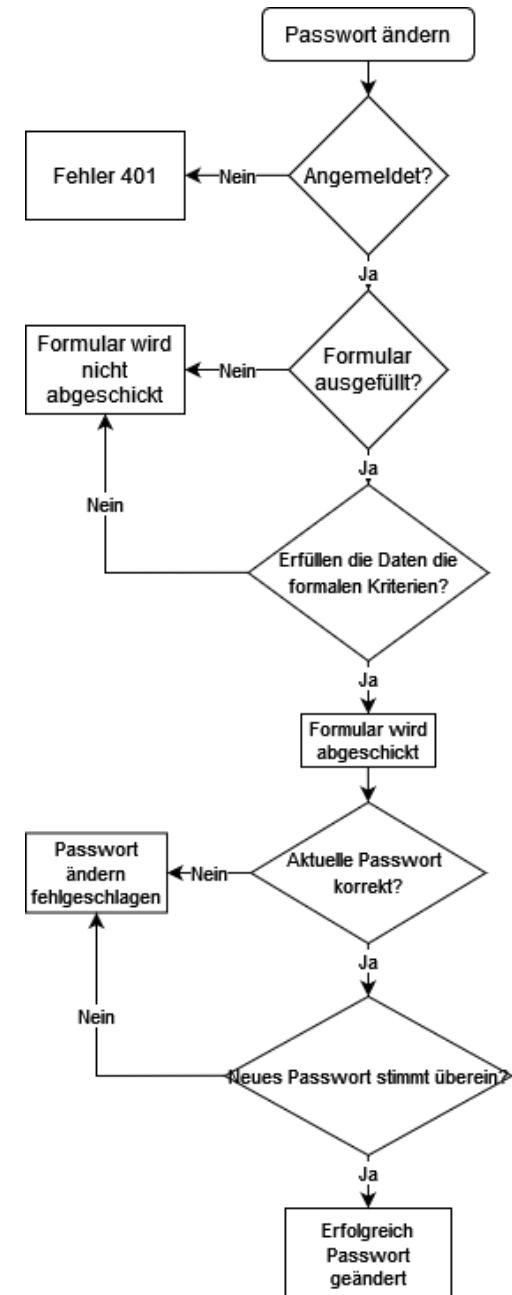
- Session starten und Variablen setzte

```
1 // Session starten und Variablen setzen.
2 session_start();
3 $_SESSION['email'] = $email;
4 $_SESSION['passwort'] = $password;
5 $_SESSION['user_id'] = $row['user_id'];
6 $_SESSION['roles_id'] = $row['roles_id'];
7 $_SESSION['login'] = TRUE;
```

Passwort ändern

Nutzen und Funktionen:

- Es wird ein UPDATE-Befehl ausgeführt.
 - Passwort des eingeloggten Nutzers wird geändert



Wie wird für die Passwort ändern Seite die Sicherheit gewährleistet?

- Vor SQL-Injection mit der `prepare()` Funktion und Parametern schützen
- Aktuelle Passwort muss für die Verifizierung genutzt werden
- Passwort muss mindestens 8-Zeichen lang sein

- Code zur Verifizierung

```
1 // Kontrolliert ob das aktuelle Passwort mit dem Passwort in der Datenbank übereinstimmt
2 $verify = password_verify($passwort_aktuell, $row['passwort']);
3 if (!$verify) {
4     // Falls das Passwort falsch ist, wird der Benutzer hierher weitergeleitet
5     header('Location: page_change_password.php?success=2');
6     return;
7 }
8 // Kontrolliert ob das neue Passwort mit dem Bestätigungspswrd übereinstimmt
9 $isPasswordCorrect = $passwort_neu == $passwort_confirm;
10 if (!$isPasswordCorrect) {
11     // Falls Passwörter nicht übereinstimmen, wird der Benutzer hierher weitergeleitet
12     header('Location: page_change_password.php?success=1');
13     return;
14 }
```

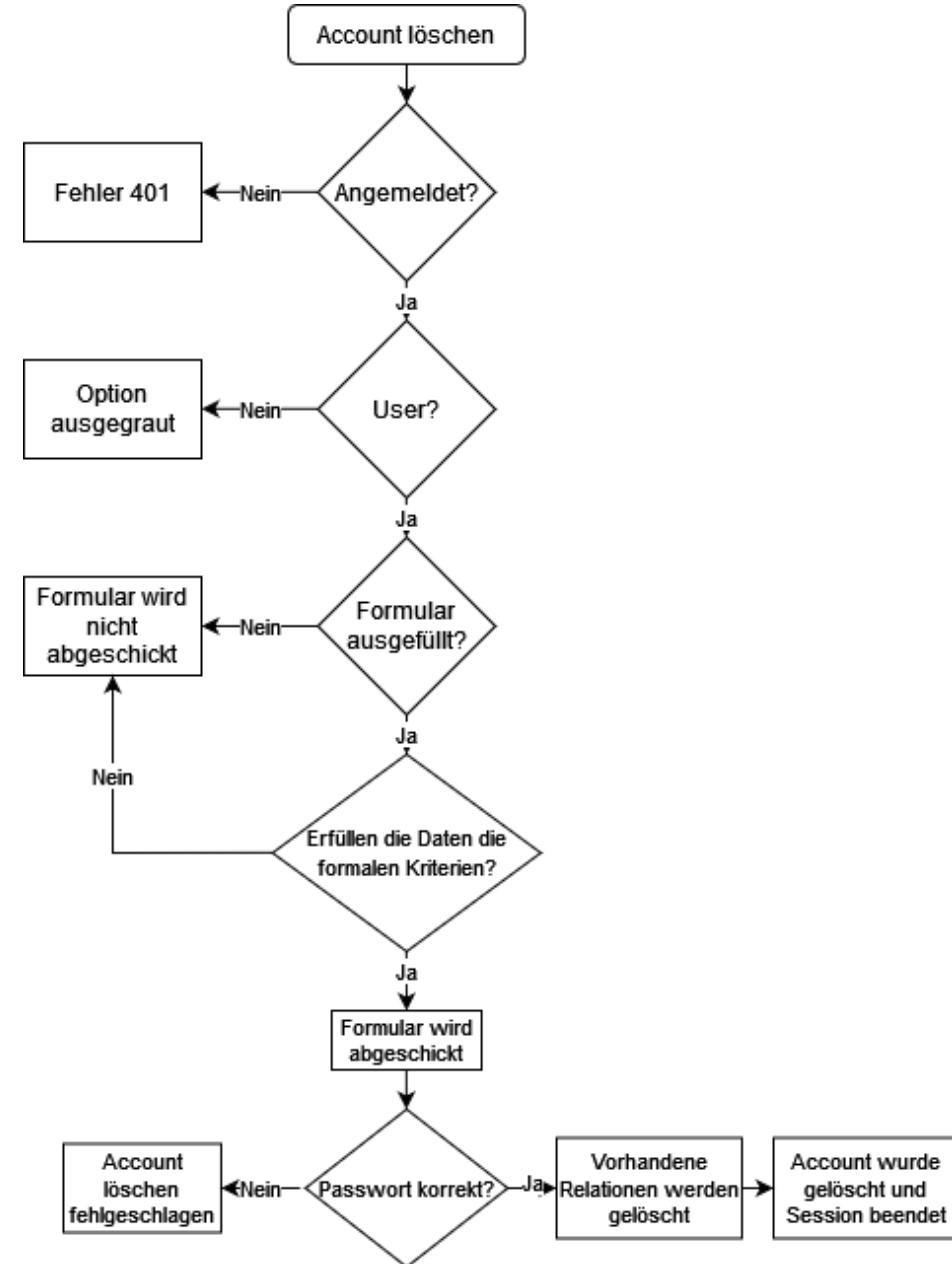
- Datenbankeintrag wird geupdated

```
1 // Updatet das Passwort in der Datenbank
2 try{
3     $sql = "UPDATE users SET passwort = :hashed_password WHERE user_id = :user_id";
4     $stmt = $dbh->prepare($sql);
5     $stmt->bindValue(':hashed_password', $hashed_password);
6     $stmt->bindValue(':user_id', $user_id);
7     $stmt->execute();
8 }
9 catch(PDOException $e){
10     echo "Error: " . $e->getMessage();
11 }
```

Account löschen als User

Nutzen und Funktionen:

- Es wird ein DELETE-Befehl ausgeführt.
 - Alle vorhandenen Relationen werden gelöscht
 - Account wird gelöscht



Wie wird für die Account löschen Seite die Sicherheit gewährleistet?

- Vor SQL-Injection mit der `prepare()` Funktion und Parametern schützen
- Aktuelle Passwort muss für die Verifizierung genutzt werden
- Die aktuelle Session wird beendet
- Damit keine Fehler auftreten, müssen vor dem löschen die Relationen gelöscht werden

- Datenbankeintrag wird gelöscht

```
1 // Falls das Passwort richtig ist, wird der Account gelöscht
2 $id = $_SESSION['user_id'];
3 $query = 'DELETE FROM users WHERE user_id = :user_id';
4 try{
5     $stmt = $dbh->prepare($query);
6     $stmt->bindValue(':user_id', $id);
7     $stmt->execute();
8 } catch (PDOException $e){
9     echo "Verbindung fehlgeschlagen: " . $e->getMessage();
10    die();
11 }
12 // Die Session wird beendet
13 session_destroy();
14 ?>
```

- Relationen werden gelöscht

```
1 //Spalte user_id in der Tabelle literatur wird auf NULL gesetzt
2 $changesql = 'UPDATE literatur SET user_id = :filler WHERE user_id = :userE_id';
3 try {
4     $stmt = $dbh->prepare($changesql);
5     $stmt->bindValue(':filler', NULL);
6     $stmt->bindValue(':userE_id', $user_id);
7     $stmt->execute();
8 } catch (PDOException $e) {
9     echo "Verbindung fehlgeschlagen: " . $e->getMessage();
10    die();
11 }
12 //Bookmark wird gelöscht
13 $changesql2 = 'DELETE FROM bookmark WHERE user_id = :userE_id';
14 try {
15     $stmt = $dbh->prepare($changesql2);
16     $stmt->bindValue(':userE_id', $user_id);
17     $stmt->execute();
18 } catch (PDOException $e) {
19     echo "Verbindung fehlgeschlagen: " . $e->getMessage();
20    die();
21 }
```

Log Out Button

Nutzen und Funktionen:

- Schließt die Sitzung des aktuellen Users

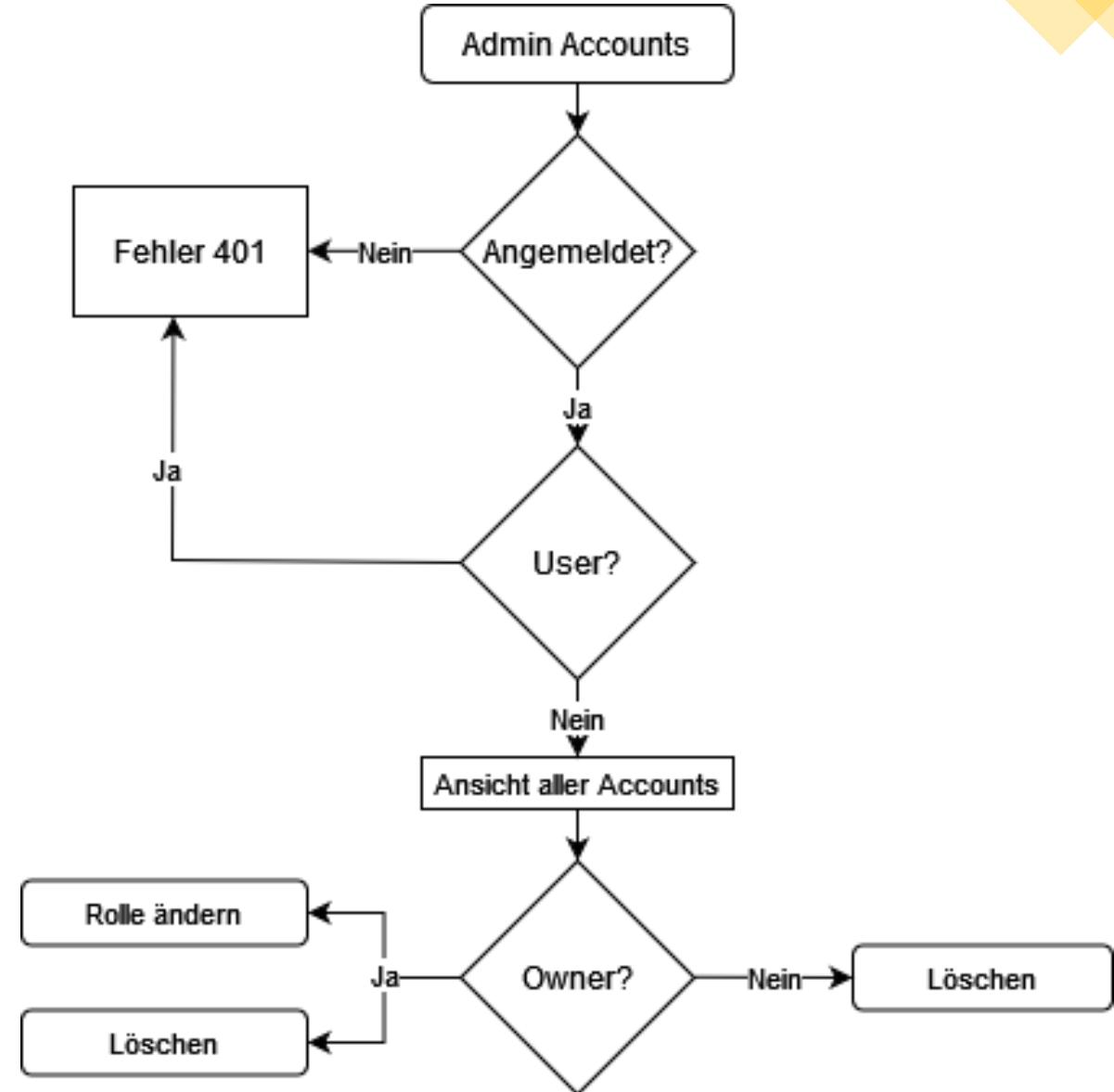
```
1 <?php  
2 session_start();  
3 session_destroy();  
4 header("Location: ../Dreessen/Homepage.php");  
5 ?>
```

Implementierung eines Administratorsystems

Funktion der Admin Accounts Seite

Nutzen und Funktionen:

- Es wird ein SELECT-Befehl ausgeführt.
 - Übersicht über alle Nutzerprofile in der Datenbank
 - Option verfügbar einen Account zu löschen
 - Option verfügbar die Rolle eines Accounts zu ändern



Wie wird für die Admin Accounts Seite die Sicherheit gewährleistet?

- Vor SQL-Injection mit der prepare() Funktion und Parametern schützen
- Man benötigt die Rollen „Admin“ oder „Owner“ um Berechtigt zu sein die Seite zu betreten
- Sowohl für die Seite als auch für die spezifische Funktion werden bestimmte Rechte vorausgesetzt

- Um die Seite sehen zu können, benötigt man Rechte

```
1 // Prüft ob der User eingelogt ist
2 if(!isset($_SESSION['roles_id'])){
3     include_once "401.php";
4     return;
5 }
6 // Holt die Rolle des Users aus der Session
7 $role = $_SESSION['roles_id'];
8 // Falls der User ein Admin ist, wird ihn der Inhalt der Seite angezeigt
9 if(($role == '1' || $role == '3')){
```

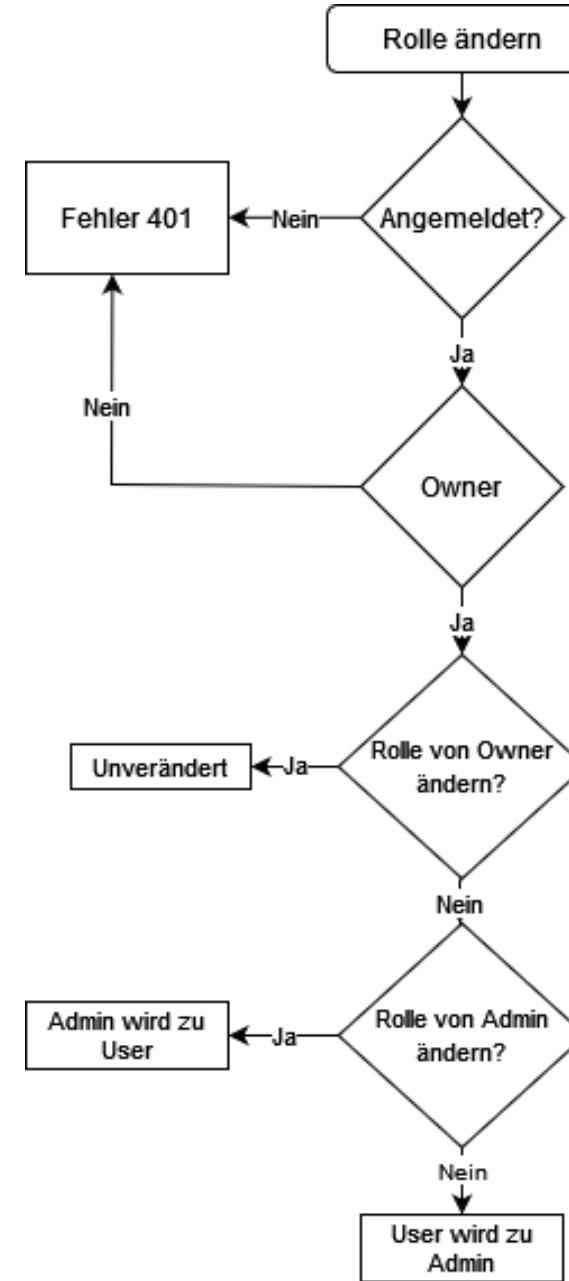
- Darstellung der Tabelle

```
1 while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
2     echo '<tr>';
3     echo '<td>' . $row['user_id'] . '</td>';
4     echo '<td>' . $row['email'] . '</td>';
5     //Da es nur zwei Rollen gibt, wurde hier eine if-else-Abfrage verwendet und keine foreach-Schleife
6     if ($row['roles_id'] == 1) {
7         echo '<td>Admin</td>';
8     } else if ($row['roles_id'] == 2){
9         echo '<td>User</td>';
10    } else {
11        echo '<td>Owner</td>';
12    }
13    if($role == 3){
14        echo '<td><button class="btn btn-warning" onclick="switchRole(' . $row['user_id'] . ')>Rolle ändern</button>';
15        echo '<button class="btn btn-danger" onclick="deleteAccount(' . $row['user_id'] . ')>Löschen</button></td>';
16        echo '</tr>';
17    }
18    if($role == 1){
19        echo '<td><button class="btn btn-danger" onclick="deleteAccount(' . $row['user_id'] . ')>Löschen</button></td>';
20        echo '</tr>';
21    }
22}
```

Funktion um die Rolle zu ändern

Nutzen und Funktionen:

- Es wird ein UPDATE-Befehl ausgeführt.
 - Adminrechte können vergeben werden
 - Adminrechte können entzogen werden



Wie wird für die Funktion die Rolle zu ändern die Sicherheit gewährleistet?

- Vor SQL-Injection mit der `prepare()` Funktion und Parametern schützen
- Man benötigt die Rollen „Owner“ um Berechtigt zu sein die Funktion sehen zu können
- Man benötigt die Rollen „Owner“ um Berechtigt zu sein die Funktion nutzen zu können
- Die Funktion funktioniert nur bei „Admins“ und „Usern“

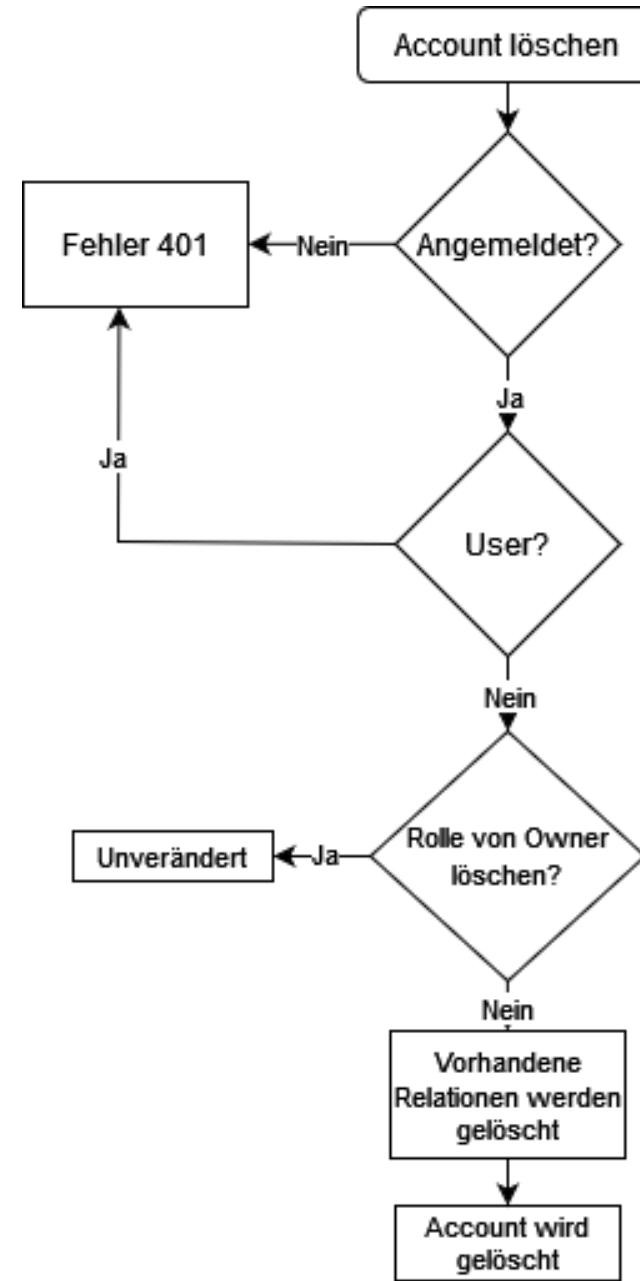
- Um die Seite sehen zu können, benötigt man Rechte

```
1 $query = 'UPDATE users SET roles_id = :newRole WHERE user_id = :user_id';
2 $isAdmin = 'SELECT roles_id FROM users WHERE user_id = :user_id';
3 // Führt die Query aus
4 try {
5     //Guckt ob der Benutzer ein Admin ist
6     $stmt = $dbh->prepare($isAdmin);
7     $stmt->bindValue(':user_id', $id);
8     $stmt->execute();
9     $row = $stmt->fetch(PDO::FETCH_ASSOC);
10    $UserRole = $row['roles_id'];
11    if($UserRole == 3){
12        header('Location: page_admin_accounts.php');
13        return;
14    }
15    $stmt = $dbh->prepare($query);
16    //Falls der Benutzer ein Admin ist, wird er zu einem User
17    if($UserRole == 1){
18        $stmt->bindValue(':newRole', 2);
19    }
20    //Falls der Benutzer ein User ist, wird er zu einem Admin
21    else if($UserRole == 2){
22        $stmt->bindValue(':newRole', 1);
23    }
24    $stmt->bindValue(':user_id', $id);
25    $stmt->execute();
26    header('Location: page_admin_accounts.php');
```

Funktion um Accounts als Admin zu löschen

Nutzen und Funktionen:

- Es wird ein DELETE-Befehl ausgeführt.
 - Alle vorhandenen Relationen werden gelöscht
 - „Admin“ und „Owner“ haben die Möglichkeit einen fremden Account zu löschen



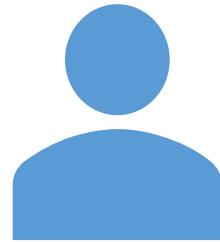
Wie wird für die Funktion um einen fremden Account zu löschen die Sicherheit gewährleistet?

- Vor SQL-Injection mit der `prepare()` Funktion und Parametern schützen
- Man benötigt die Rollen „Admin“ oder „Owner“ um Berechtigt zu sein die Funktion zu nutzen
- Die Funktion funktioniert nur bei „Admins“ und „Usern“
- Damit keine Fehler auftreten, müssen vor dem Löschen die Relationen gelöscht werden

- Um die Seite sehen zu können, benötigt man Rechte

```
1 $checkOwner = 'SELECT roles_id FROM users WHERE user_id = :user_id';
2     try {
3         $stmt = $dbh->prepare($checkOwner);
4         $stmt->bindValue(':user_id', $id);
5         $stmt->execute();
6         $row = $stmt->fetch(PDO::FETCH_ASSOC);
7         $UserRoleOwner = $row['roles_id'];
8     } catch (PDOException $e) {
9         echo "Verbindung fehlgeschlagen: " . $e->getMessage();
10        die();
11    }
12
13    if($UserRoleOwner == 3){
14        header('Location: page_admin_accounts.php');
15        return;
16    }
```

Übersicht der geplanten Implementierungen



Erweiterung des Administratorsystems

Möglichkeit Einträge hinzuzufügen

Möglichkeit Einträge zu löschen

Möglichkeit Einträge zu bearbeiten

Erweiterung des Administratorsystems

Datenbankentitäten



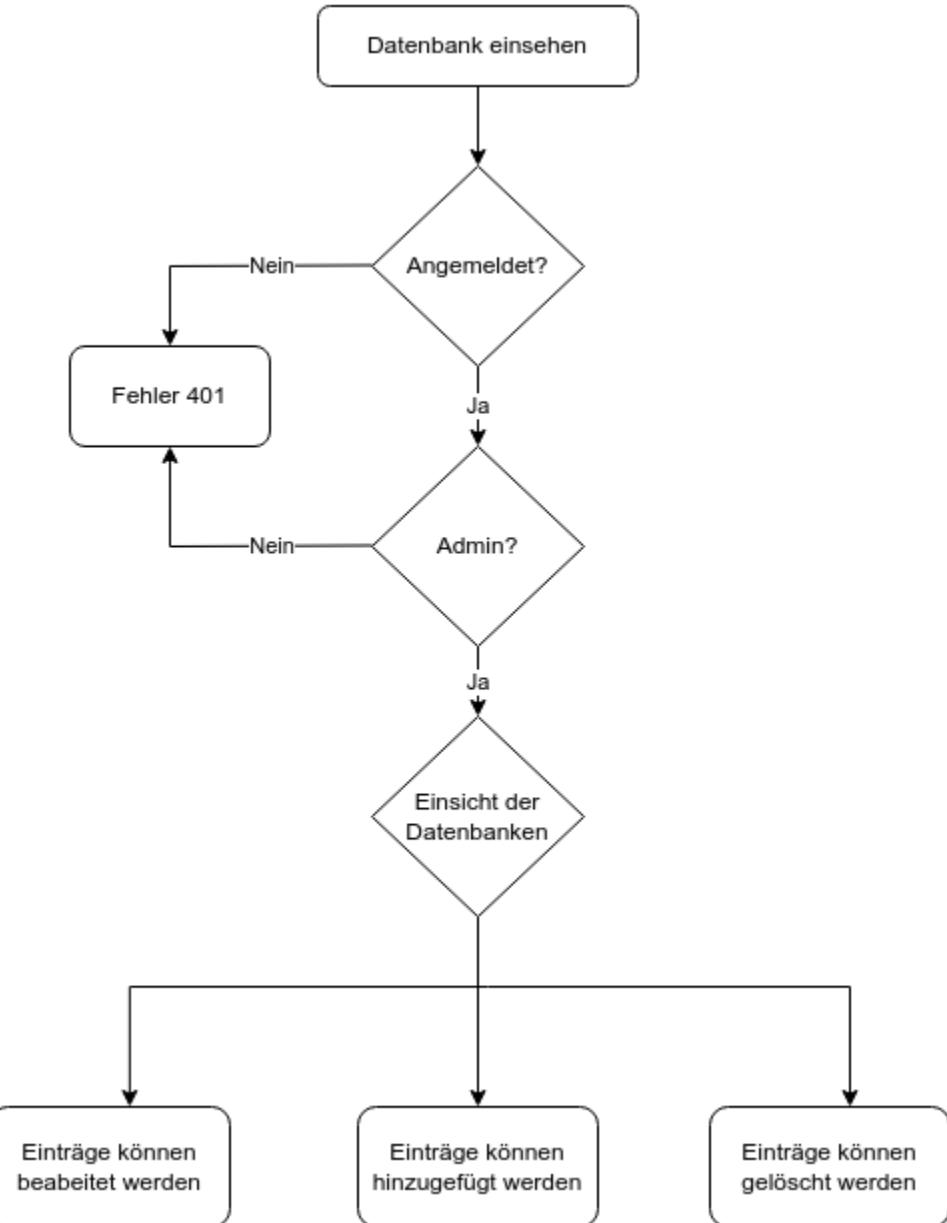
Nutzen:

- Administratoren und Owner können Datenbankeinträge einsehen und hinzufügen
- Möglichkeit des Löschens und der Bearbeitung

Datenbankentitäten

Nutzen:

- Datenbankeinträge einsehen
- Datenbankeinträge hinzufügen
- Datenbankeinträge bearbeiten
- Datenbankeinträge löschen



Datenbankentitäten - Hinzufügen

```
38 <form style="margin-left:20px" action="savefachbereich.php" method="post">
39   <div class="row align-items-end">
40     <div class="col-md-4 mb-3">
41       <label for="fachbereich">Fachbereich:</label>
42       <input type="text" class="form-control" name="fachbereich" required>
43     </div>
44     <div class="col-md-4 mb-3">
45       <label for="comment">Anmerkungen</label>
46       <input type="text" class="form-control" name="comment" id="comment">
47     </div>
48
49     <div class="col-auto">
50       <button type="submit" name="submit" class="btn btn-light" onclick="alert('Neuer Fachbereich hinzugefügt!'); location.href = 'addFachbereich.php'; return true;">Upload</button><br><br>
51     </div>
52   </div>
53 </div>
54 </form>
55 <?php
56 include('addFachbereich_display.php');
57 ?>
```

Datenbankentitäten - Hinzufügen

```
40     $sql = "SELECT * FROM fachbereich ORDER BY fachbereich_id";
41     $typen = executeSQL($sql)->fetchAll();
42 }
43 foreach ($typen as $row) {
44     echo "<tr>";
45     echo "<td>" . $row['fachbereich_id'] . "</td>";
46     echo "<td>" . $row['fachbereich'] . "</td>";
47     echo "<td>" . $row['comment'] . "</td>";
48     echo '<td><div class="row"><form action="fachbereich-edit-form.php" method="GET" class="px-2">';
49     echo '<input type="hidden" name="fachbereich_id" value="' . $row['fachbereich_id'] . '">';
50     echo '<input type="hidden" name="fachbereich" value="' . $row['fachbereich'] . '">';
51     echo '<input type="hidden" name="comment" value="' . $row['comment'] . '">';
52     echo '<input type="submit" class="btn btn-primary" name="edit" value="Bearbeiten">';
53     echo '</form>';
54     echo '<form action="loeschenrow-fachbereich.php" method="post">';
55     echo '<input type="hidden" name="fachbereich_id" value="' . $row['fachbereich_id'] . '">';
56     echo '<input type="submit" class="btn btn-danger" value="Löschen" onclick="return confirm(\\'Sicher das Sie diesen Eintrag löschen möchten?\')">';
57     echo '</form></div></td>';
58     echo "</tr>";
59 }
60 ?
61
62 ?>
```

Datenbankentitäten - Hinzufügen

```
3 <?php
4     $varfachbereich= $_POST['fachbereich'];
5     $varanmerkung= $_POST['comment'];
6
7     require_once 'connect-server.php';
8     $newID= (executeSQL("SELECT max(fachbereich_id) FROM fachbereich")->fetch())[0]+1;
9     $insertLine = "INSERT INTO fachbereich(fachbereich_id, fachbereich, comment)
10             VALUES ($newID, '$varfachbereich', '$varanmerkung');";
11     executeSQL($insertLine);
12     header("Location: addFachbereich.php")
13 ?>
```

Datenbankentitäten - Löschen

```
1 <?php
2 require_once 'connect-server.php';
3 $id = $_POST['fachbereich_id'];
4 $sql = "UPDATE literatur SET fachbereich_id = :filler WHERE fachbereich_id = :id";
5 $stmt = $dbConnection->prepare($sql);
6 $stmt->bindValue(':filler', NULL);
7 $stmt->bindValue(':id', $id);
8 $stmt->execute();
9
10 $sql = "DELETE FROM fachbereich WHERE fachbereich_id = $id";
11 $stmt = $dbConnection->prepare($sql);
12 $stmt->execute();
13 header("Location: addFachbereich.php");
14 ?>
```

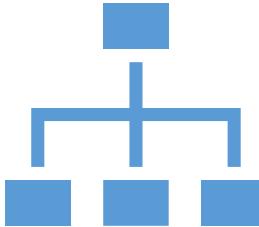
Datenbankentitäten - Bearbeiten

```
29 <?php
30 $id=$_GET['fachbereich_id'];
31 $fachbereich=$_GET['fachbereich'];
32 $comment=$_GET['comment'];
33 ?>
34 <br><br>
35 <form tyle="margin-left:20px" action="update-fachbereich.php" method="post">
36 <div class="col-md-12 mb-lg-4 ">
37 <label for="fachbereich">Fachbereich:</label>
38 <input type="hidden" name="fachbereich_id" value="<?php echo $id; ?>">
39 | <input type="text" class="form-control" name="newFachbereich" id="newFachbereich" value="<?php echo $fachbereich; ?>" required>
40 </div>
41 <div class="col-md-12 mb-lg-4 ">
42 | | | | <label for="comment">Anmerkungen</label>
43 | <input type="text" class="form-control" name="newComment" id="newComment" value="<?php echo $comment; ?>">
44 </div>
45 <div class="col-md-12 mb-lg-4 ">
46 <input type="submit" value="Update" name="update" class="btn btn-light"
47 onclick="alert('Fachbereich wurde aktualisiert!'); location.href = 'addFachbereich.php'; return true;" /><br><br>
```

Datenbankentitäten - Bearbeiten

```
2 <?php
3 require_once 'connect-server.php';
4 $id = $_POST['fachbereich_id'];
5 $newFachbereich=$_POST['newFachbereich'];
6 $newComment=$_POST['newComment'];
7 $sql = "UPDATE fachbereich SET fachbereich = :newFachbereich, comment = :newComment WHERE fachbereich_id = :fachbereich_id";
8 $stmt = $dbConnection->prepare($sql);
9 $stmt->bindParam(':newFachbereich', $newFachbereich);
10 $stmt->bindParam(':newComment', $newComment);
11 $stmt->bindParam(':fachbereich_id', $id);
12 $stmt->execute();
13 header("Location: addFachbereich.php");
14 ?>
```

Übersicht der geplanten Implementierungen



Implementierung von Navbar und Footer

Aufbau des Körpers der Seite

Möglichkeit des Navigierendes zwischen den Funktionen

Ende der Website durch ein Footer



Implementierung der Funktion Hinzufügen

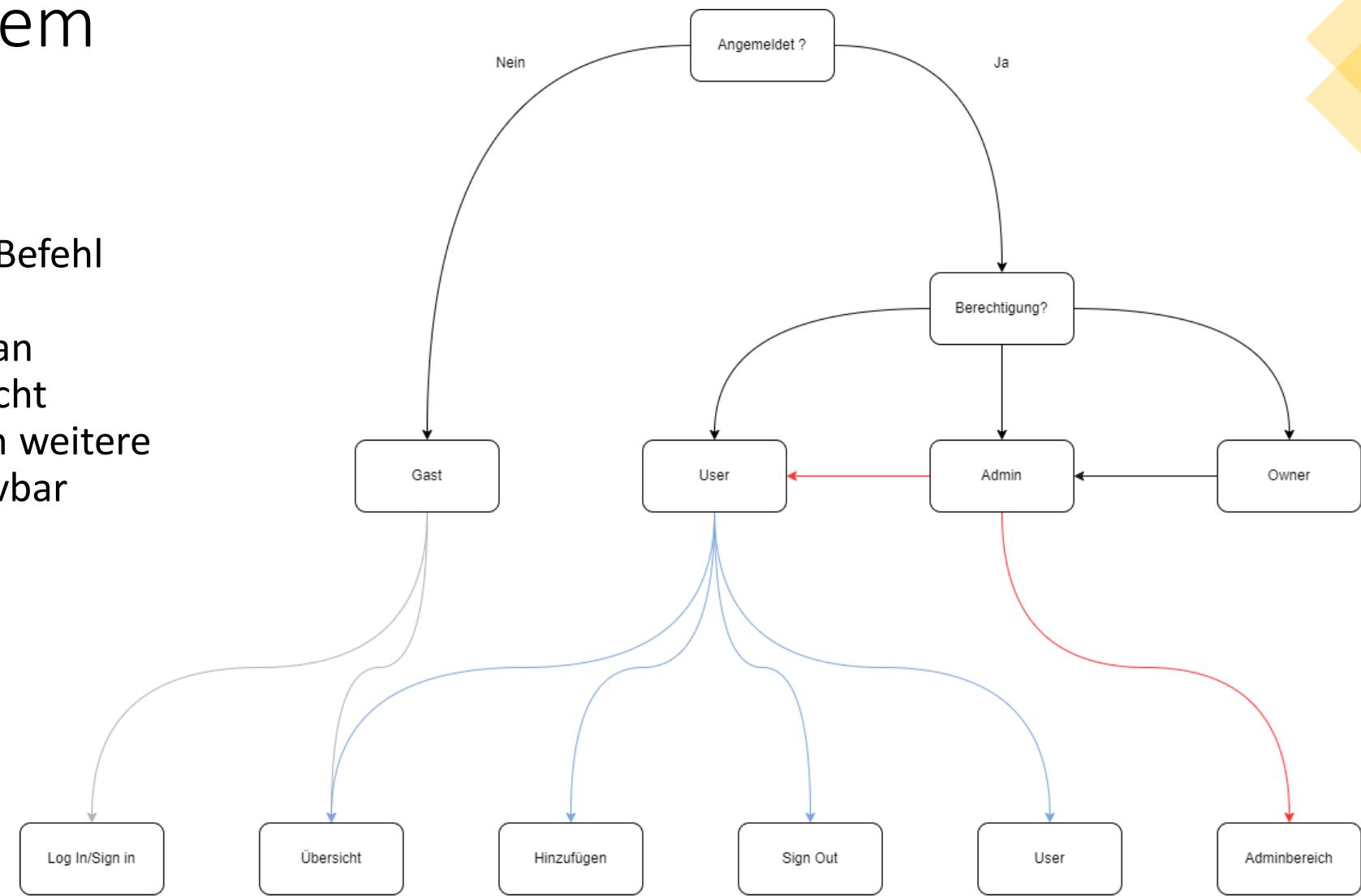
Möglichkeit von Usern das Eintragen von neuen Literaturen

Implementierung der Navbar

Navigationssystem

Funktionen und Nutzen:

- Es wird ein Session Start Befehl ausgeführt
 - Es wird gesehen, ob man angemeldet ist oder nicht
 - Je nach User ID werden weitere Funktionen auf der Navbar angezeigt



Funktion des Session_start() in der Navbar:

- Der session_start Befehl erzeugt eine Session oder nimmt eine erstellte Session wieder auf.
 - Durch den if Befehl wird erstmal geguckt ob eine Session schon besteht indem er guckt ob eine Session namens roles_id existiert
 - Wenn es existiert dann werden der Session die schon existierenden Daten gegeben
 - Wenn nicht, werden die die Daten automatisch zu den Werten Null und FALSE

```
1 <?php
2 session_start();
3 if (isset($_SESSION['roles_id'])) {
4     $user_role = $_SESSION['roles_id'];
5     $login = $_SESSION['login'];
6 } else {
7     $user_role = 0;
8     $login = FALSE;
9 }
10 ?>
```

Header der Webseite:

- Der Header ist der Anfang der Website, wo das Logo der Website zu sehen ist
- Die if(\$login) vergleicht ob der Login Befehl True ist
 - Wenn zeigt er den Button Sign Out
 - Wenn False dann zeigt er die beiden Buttons Log In und Sign Up an

```
1 <header class="blog-header py-3">
2   <div class="row flexnowrap justify-content-between align-items-center">
3     <div class="col-4 pt-1">
4       <a class="text-muted" href="#"></a>
5     </div>
6     <div class="col-4 text-center">
7       <a class="blog-header-logo text-dark" href="../Dreessen/Homepage.php">SciRead</a>
8     </div>
9     <div class="col-4 d-flex justify-content-end align-items-center">
10
11       <?php if ($login) {
12         echo '<a class="btn btn-sm btn-outline-secondary" href="../Cetin/function_logout.php">Sign Out</a>';
13       } else {
14         echo '<a class="btn btn-sm btn-outline-secondary" href="../Cetin/page_login.php">Log In</a>';
15         echo '<a class="btn btn-sm btn-outline-secondary" href="../Cetin/page_register.php">Sign Up</a>';
16       }
17       ?>
18     </div>
19   </div>
20 </header>
```

Navbar das Navigationssystem der Website:

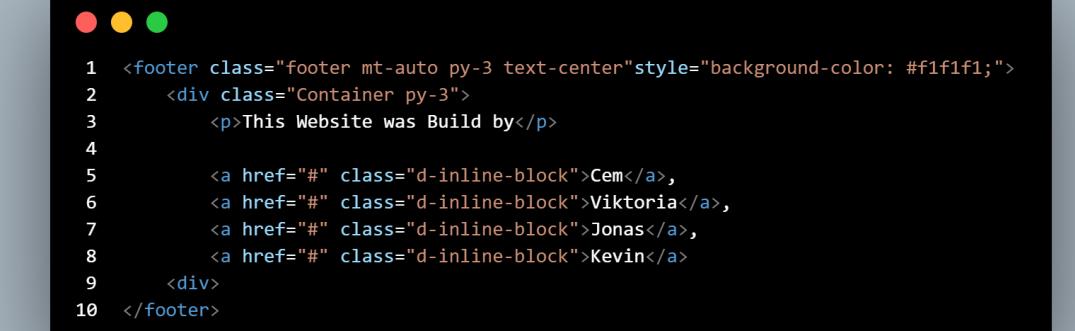
- Der If Befehl verändert den Link zu den Funktionen.
 - Login=True: Man kommt zu den Funktionen Hinzufügen und User
 - Login = False: Man wird zur Log in Page befördert.
- Der If Befehl guckt auf die verteilte User_role und zeigt nur den Navbar Text Adminbereich an wenn die user_role nicht 0 oder 2 ist

```
1 <div class="nav-scroller py-1 mb-2">
2   <nav class="nav d-flex justify-content-between">
3     <a class="p-2 text-muted" href="../Dreessen/Homepage.php">Übersicht</a>
4     <?php if ($login) {
5       echo '<a class="p-2 text-muted" href="../Dreessen/Submit_page.php">Hinzufügen</a>';
6       echo '<a class="p-2 text-muted" href="../Cetin/page_profile.php">User</a>';
7     } else {
8       echo '<a class="p-2 text-muted" href="../Cetin/page_login.php">Hinzufügen</a>';
9       echo '<a class="p-2 text-muted" href="../Cetin/page_login.php">User</a>';
10    }
11  ?>
12  <?php if ($user_role == "0" || $user_role == '2') {
13    echo '<a class="p-2 text-muted" href="#" style="display:none;">Adminbereich</a>';
14  } else {
15    echo '<a class="p-2 text-muted" href="../Cetin/page_admin.php">Adminbereich</a>';
16  } ?>
17  <!--Einrücken der Funktionen-->
18  <a class="p-2 text-muted" href="#"></a>
19  <a class="p-2 text-muted" href="#"></a>
20  <a class="p-2 text-muted" href="#"></a>
21  <a class="p-2 text-muted" href="#"></a>
22  <a class="p-2 text-muted" href="#"></a>
23  <a class="p-2 text-muted" href="#"></a>
24  <a class="p-2 text-muted" href="#"></a>
25  <a class="p-2 text-muted" href="#"></a>
26  </nav>
27 </div>
```

Implementierung des Footers

Footer

- Erstellt ein Footer, der das Ende der Website andeutet.



```
1 <footer class="footer mt-auto py-3 text-center" style="background-color: #f1f1f1;">
2   <div class="Container py-3">
3     <p>This Website was Build by</p>
4
5     <a href="#" class="d-inline-block">Cem</a>,
6     <a href="#" class="d-inline-block">Viktoria</a>,
7     <a href="#" class="d-inline-block">Jonas</a>,
8     <a href="#" class="d-inline-block">Kevin</a>
9   <div>
10  </div>
11 </div>
12 </div>
13 </div>
14 </div>
```

Aufbau einer Website mit Navbar und Footer



```
1 include_once 'navbar.php';
```



```
1 <?php include_once 'footer.php' ?>
```

Homepage

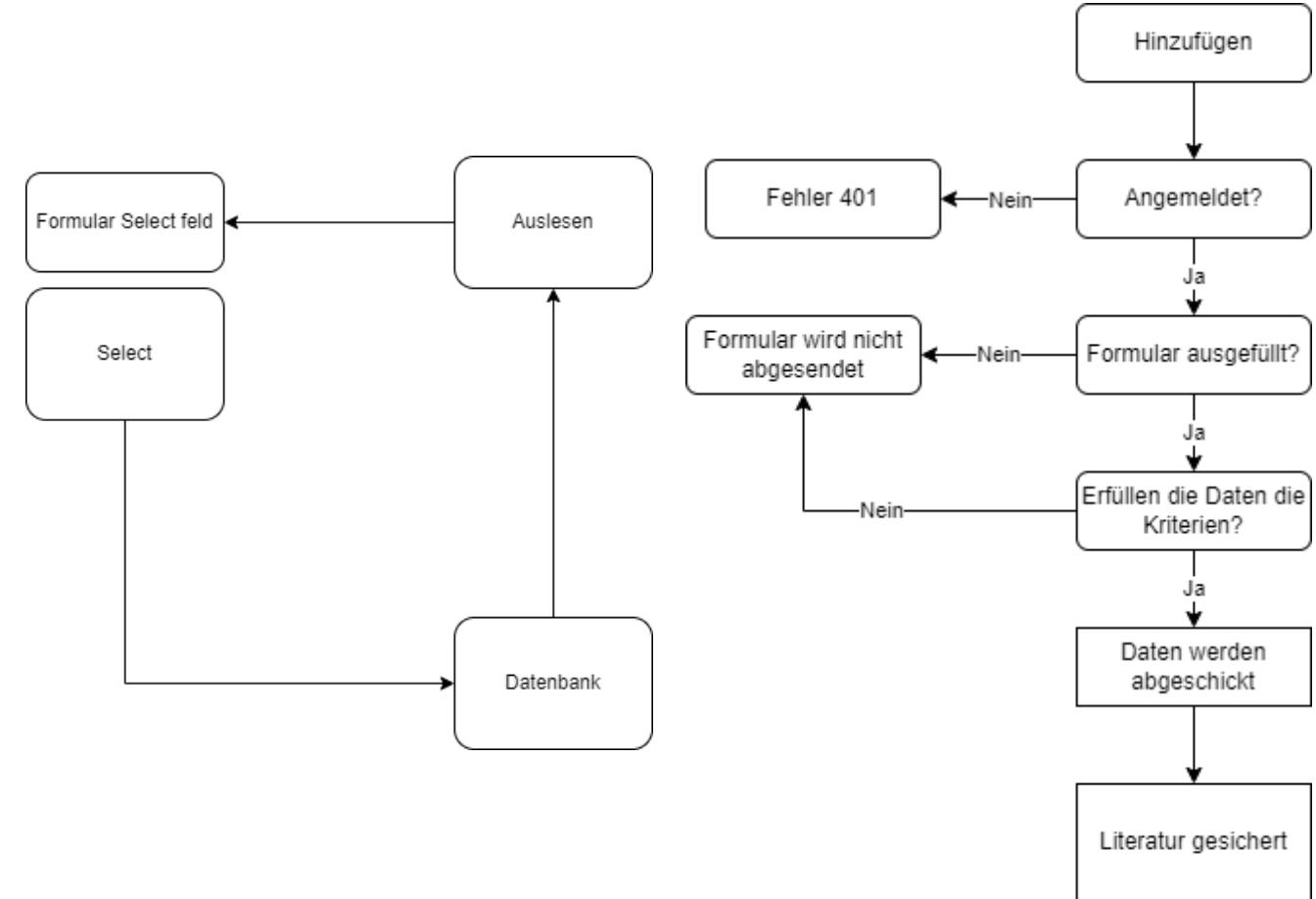
```
● ● ●  
1  <head>  
2      <title>SciRead|Homepage </title>  
3  </head>  
4  <body>  
5      <div class="container">  
6          <div class="jumbotron p-3 p-md-5 text-white rounded bg-dark">  
7              <div class="col-md-6 px-0">  
8                  <h1 class="display-4 font-italic">Willkommen zu SciRead</h1>  
9                  <p class="lead my-3">Die erste Datenbank, bei der man wissenschaftliche Literatur suchen und speichern kann.</p>  
10             </div>  
11         </div>  
12         <?php include_once '../Kliefoth/search.php'?>  
13     </div>
```

Implementierung der Hinzufügen Funktion

Hinzufügen

Nutzen:

- Fügt neue Literatur zu
- Liest Daten aus der Datenbank und fügt sie in ein Auswahl Formular ab.



Aufbau des Hinzufügen Felds

```
● ● ●  
1 <form action="Submit_database.php" method="post">  
  
● ● ●  
1 <div class="col-md-4 mb-3">  
2   <label for="author">Autor:</label>  
3   <input type="text" class="form-control" name="author" placeholder="Autor" required>  
4   <div class="invalid-feedback">  
5     Bitte gebe den Autor an.  
6   </div>  
7 </div>  
  
● ● ●  
1 <button class="btn btn-primary" type="submit">Abschicken</button>
```

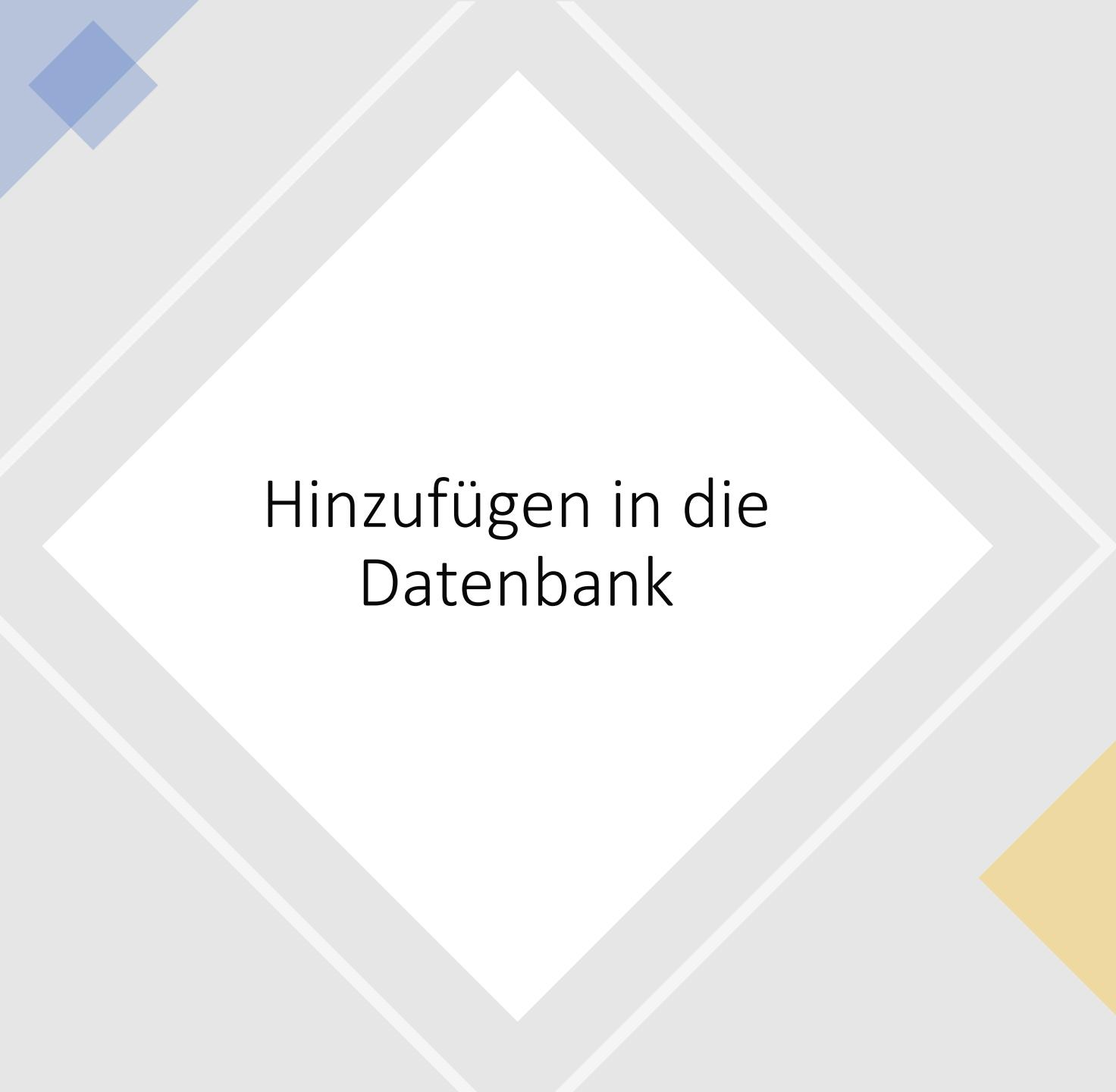
Aufbau des Auswahl Input

Select Befehl zeigt die angegebenen Daten in der Datenbank als eine Auswahl Feld an.

Die AJAX Post request verändert die HTML ohne sie nachzuladen.

```
1 <label for publisher>Verlag:</label>
2 <select class="custom-select mr-sm-2" name="verlag" required onchange="getVerlagId(this.value)">
3   <option value="" disabled selected hidden>Wähle...</option>
4   <?php
5     $stmt = executeSQL("SELECT name FROM publisher ORDER BY name ASC");
6     while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
7       echo '<option value="' . $row['name'] . '">' . $row['name'] . '</option>';
8     }
9   ?>
10  </select>
```

```
1 function getVerlagId(publisher) {
2   $.ajax({
3     type: 'POST',
4     url: 'Submit.php',
5     data: {publisher: publisher},
6     success: function(response) {
7       document.getElementById("publisher_id").value = response;
8     }
9   });
}
```



Hinzufügen in die
Datenbank

Aufbau der Submit Datenbank

- Angegebene Daten werden aus dem Input Feld entnommen und in eine String gespeichert
- Datei die zur Datenbank anbindet wird hinzugefügt
- Die aus der Select class werden mit deren ID vergleicht und getauscht
- ID wird den in String getan
- Daten werden nun in Form eingegeben und in die Datenbank eingetragen
- Wenn erfolgreich wird Literatur gesichert ausgegeben

```
1 $varauthor = $_POST['author'];
```



```
1 require_once 'Server_connect.php';
```



```
1 $stmt = $dbConnection->prepare("SELECT fachbereich_id FROM fachbereich WHERE fachbereich = ?");  
2 $stmt->execute([$fachbereich]);  
3 $id = $stmt->fetch(PDO::FETCH_ASSOC);  
4 $varfachbereichid = $id['fachbereich_id'];
```



```
1 $newId = (executeSQL("SELECT max(literatur_id) FROM literatur")->fetch())[0]+1;  
2 $insertLine = "INSERT INTO literatur(literatur_id, pages, author, title, published_date, comment, fachbereich_id, type_id, publisher_id, user_id)  
3 VALUES ($newId, '$varpages', '$varauthor', '$vartitle', '$varpublished', '$vardescription', '$varfachbereichid', '$vartypeid', '$varverlagid', '$user_id');";  
4 executeSQL($insertLine);  
5 echo "<div class= container><div class=card-header><h3>Literatur gesichert </h3></div></div>";
```

Funktionen und Live Demo der Suchfunktionen

Übersicht der geplanten Implementierungen



Implementierung einer Suchfunktion

- Soll Namen suchen können
- Soll Filtern können
- Soll sortieren können



Implementierung einer Merkliste

- Soll Literatur sich merken können
- Soll nach gemerkter Literatur suchen können

Allgemeine Suche



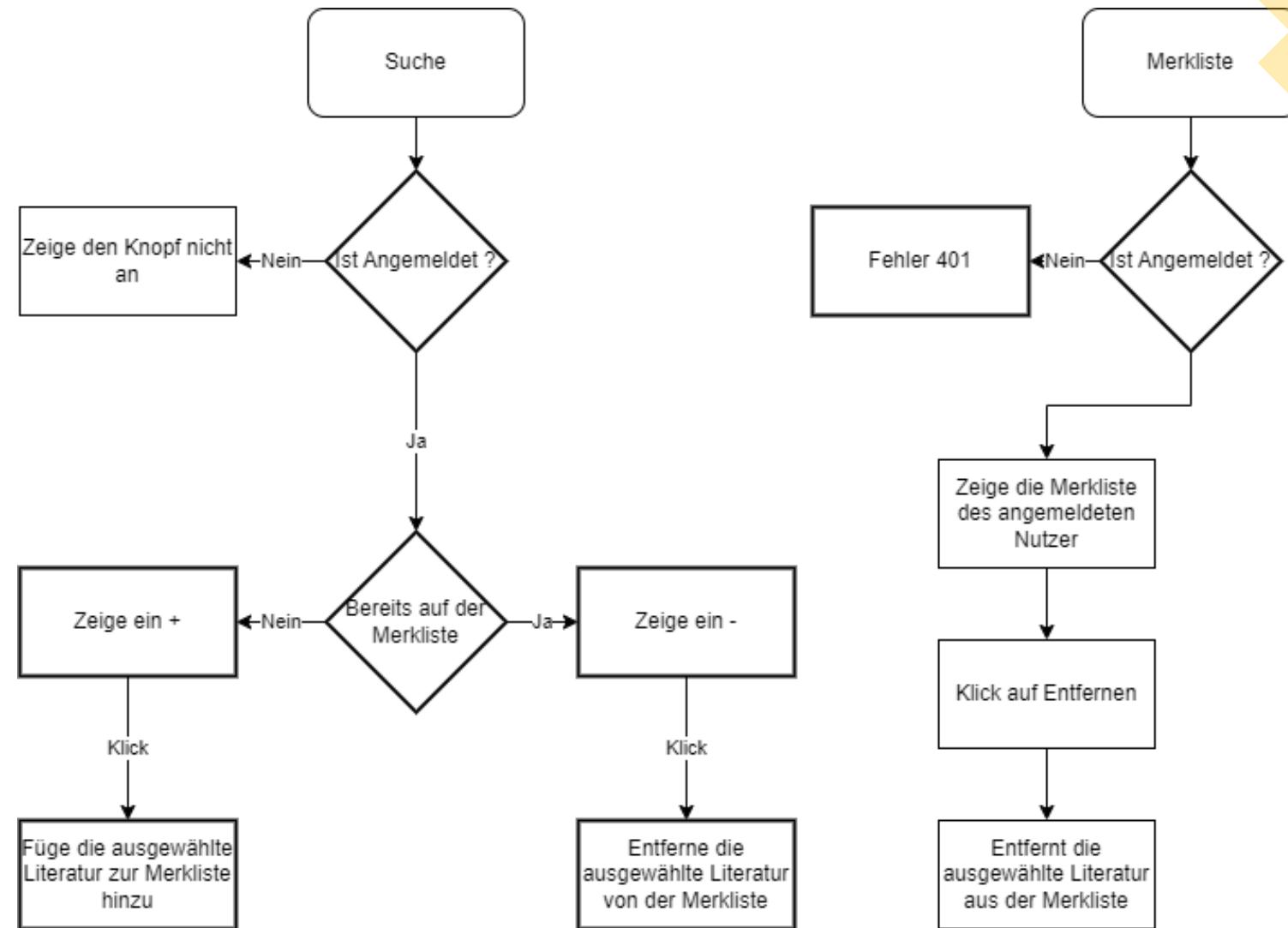
Nutzen:

- Auflistung aller Literaturen
- Literaturen zur Merkliste hinzufügen

Merkliste

Nutzen:

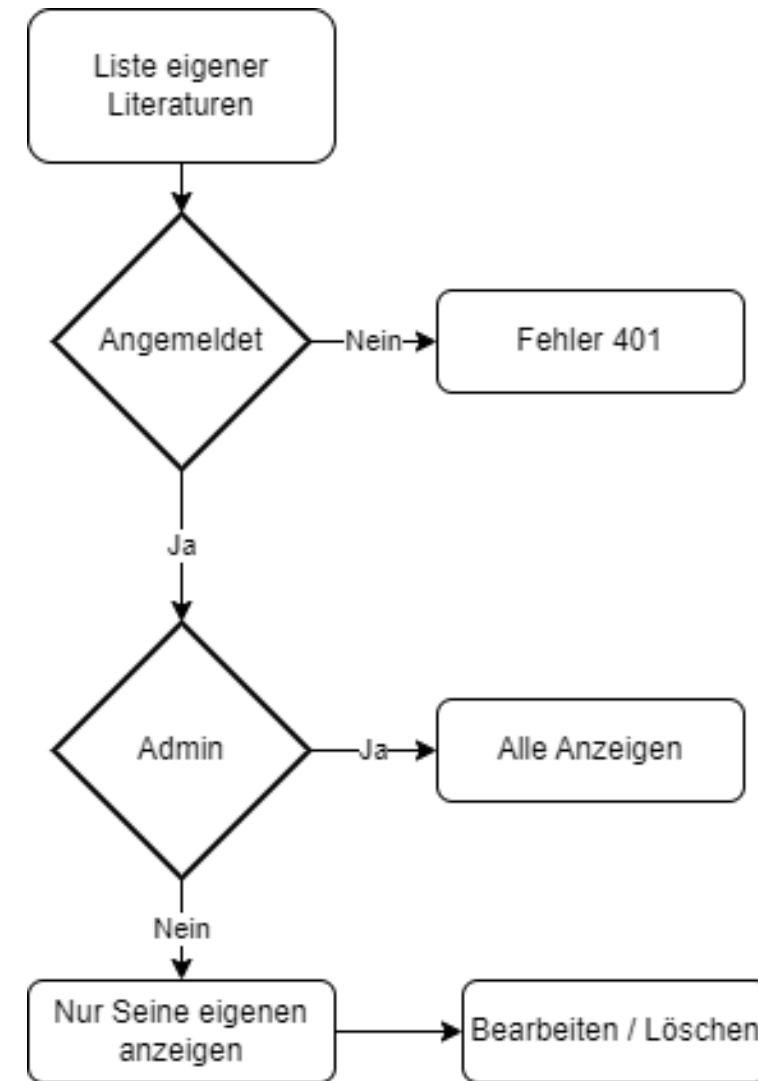
- Literaturen zur Merkliste hinzufügen
- Literaturen von der Merkliste entfernen



Suche eigener Literaturen

Nutzen/Funktionen:

- Einträge bearbeiten
- Einträge löschen
- Admins: Einträge verwalten



Ausgewählte Code- Blöcke

Suche



```
1 <?php require_once "../Dreessen/Server_connect.php";
2 if (isset($_POST['titel']) || isset($_GET['titel'])) {
3     if (isset($_POST['titel'])) {
4         $titel = $_POST['titel'];
5     } else {
6         $titel = $_GET['titel'];
7     }
8 } else {
9     $titel = "";
10 }
11 if (isset($_POST['search-filter'])) {
12     $sort = "ORDER BY " . $_POST['search-filter'];
13 } else {
14     $sort = "ORDER BY Title ASC";
15 }
16 $books = executeSQL("SELECT title,pub.name,pages,ty.type,author,published_date,fb.fachbereich,lit.literatur_id
17                     FROM public.\"literatur\" lit
18                     JOIN fachbereich fb USING (fachbereich_id)
19                     JOIN publisher pub USING (publisher_id)
20                     JOIN type ty USING (type_id)
21                     WHERE title LIKE '$titel%' $sort")->fetchAll();
22 ?>
```



```
23 <form action="search.php" method="post" class="ms-5 me-5 border-top">
24     <div class="mb-3 mt-3 row align-items-end">
25         <div class="col-auto">
26             <div class="form-group">
27                 <label for="titel" class="col-form-label">Suche:</label>
28                 <input class="form-control" type="search" name="titel" id="titel" placeholder="Titel eingeben" value="=php echo $titel?" autocomplete="off">
29             </div>
30         </div>
31         <div class="col-auto">
32             <div class="form-group">
33                 <label for="search-filter" class="col-form-label">Sortieren:</label>
34                 <select class="form-control" name="search-filter" id="search-filter">
35                     <option value="title ASC" <?php if ($sort === "ORDER BY title ASC") echo "selected" ?>>Titel ↓</option>
36                     <option value="title DESC" <?php if ($sort === "ORDER BY title DESC") echo "selected" ?>>Titel ↑</option>
37                     <option value="author ASC" <?php if ($sort === "ORDER BY author ASC") echo "selected" ?>>Autor ↓</option>
38                     <option value="author DESC" <?php if ($sort === "ORDER BY author DESC") echo "selected" ?>>Autor ↑</option>
39                     <option value="pages ASC" <?php if ($sort === "ORDER BY pages ASC") echo "selected" ?>>Seitenanzahl ↓</option>
40                     <option value="pages DESC" <?php if ($sort === "ORDER BY pages DESC") echo "selected" ?>>Seitenanzahl ↑</option>
41                     <option value="published_date ASC" <?php if ($sort === "ORDER BY published_date ASC") echo "selected" ?>>Veröffentlichungsdatum ↓</option>
42                     <option value="published_date DESC" <?php if ($sort === "ORDER BY published_date DESC") echo "selected" ?>>Veröffentlichungsdatum ↑</option>
43             </select>
44         </div>
45     </div>
46     <div class="col-auto">
47         <div class="form-group">
48             <button type="submit" class="btn btn-primary">Abschicken</button>
49         </div>
50     </div>
51 </div>
52 </form>
```

```
● ● ●
1  <?php if (count($books) > 0) { ?>
2      <table class="table">
3          <thead>
4              <tr>
5                  <th scope="col">Titel</th>
6                  <th scope="col">Verlag</th>
7                  <th scope="col">Seitenanzahl</th>
8                  <th scope="col">Typ</th>
9                  <th scope="col">Autor</th>
10                 <th scope="col">Veröffentlichungsdatum</th>
11                 <th scope="col">Fachbereich</th>
12                 <?php if (isset($user_id)) { ?>
13                     <th></th>
14                 <?php } ?>
15             </tr>
16         </thead>
17         <tbody>
18             <?php
19             foreach ($books as $b) {
20
21                 echo "<tr>";
22                 echo "<td>" . $b[0] . "</td>";
23                 echo "<td>" . $b[1] . "</td>";
24                 echo "<td>" . $b[2] . "</td>";
25                 echo "<td>" . $b[3] . "</td>";
26                 echo "<td>" . $b[4] . "</td>";
27                 echo "<td>" . $b[5] . "</td>";
28                 echo "<td>" . $b[6] . "</td>";
29                 if (isset($user_id)) {
30                     $bookmark = executeSQL("SELECT literatur_id FROM public.\"bookmark\""
31                                         WHERE literatur_id = $b[7]
32                                         and user_id = $user_id")->fetch();
33                     if ($bookmark == null) {
34                         echo "<td><a class='btn border' href='search.php?use=addbookmark&id=$b[7]'>+</a></td>";
35                     } else {
36                         echo "<td><a class='btn border' href='search.php?use=removebookmark&id=$b[7]'>-</a>
37 </td>";
38                     }
39                     echo "</tr>";
40                 }
41             ?>
42         </tbody>
43     </table>
44 <?php } else { ?>
45     <p> Keine Einträge gefunden </p>
46 <?php } ?>
```

Merkliste



```
1  <?php
2  if ($use === "delete" && isset($id)) {
3      if (!isset($agreed)) {
4          $book = executeSQL("SELECT title,pub.name,pages,ty.type,author,published_date,fb.fachbereich,book.user_id FROM public.\"bookmark\" book
5                          JOIN public.\"literatur\" lit  USING (literatur_id)
6                          JOIN fachbereich fb USING (fachbereich_id)
7                          JOIN publisher pub USING (publisher_id)
8                          JOIN type ty USING (type_id)
9                          WHERE literatur_id='".$id"')->fetch();
10     if (isset($user_id)) {
11         if ($user_id == $book[7]) {
12     ?>
13         <div class="container">
14             <div class="card">
15                 <div class="card-header">
16                     <h3>Merkliste</h3>
17                 </div>
18                 <div class="card-body container-fluid">
19                     <div class="row align-items-center ms-5 me-5 mt-5 border-top">
20                         <p class="col pt-3 pb-3 mb-0 text-center">Wollen Sie wirklich das Buch von der Merkliste entfernen?</p>
21                     </div>
22                     <div class="row ms-5 me-5 text-center">
23                         <table class="table">
24                             <thead>
25                                 <tr>
26                                     ... <!-- Alle Reihen Benennen wie in der Suche-->
27                                 </tr>
28                             </thead>
29                             <tbody>
30                                 <?php
31                                     echo "<tr>";
32                                     ... // Alle Reihen ausgeben wie in der Suche
33                                     echo "</tr>";
34                                 ?>
35                             </tbody>
36                         </table>
37                     </div>
38                     <div class="row ms-5 me-5 justify-content-evenly">
39                         <a href="bookmark_search.php" class="col-4 btn btn-primary">Zurück </a>
40                         <a class="col-4 btn btn-danger" href="bookmark_search.php?use=delete&id=<?php echo $id ?>&agreed=1">Entfernen?</a
41     >
42                         </div>
43                     </div>
44                 </div>
```



```
60 $book_userid = executeSQL("SELECT user_id FROM public.bookmark WHERE literatur_id='".$id."")->fetch()[0];
61     if (isset($user_id)) {
62         if ($user_id === $book_userid) {
63             executeSQL('DELETE FROM public.bookmark WHERE literatur_id=' . $id . ' AND user_id=' . $user_id)->fetch
64 ();        }
65     }
66     echo "<script>window.location.href='bookmark_search.php'</script>";
67 
```

Reflektion

Technologieentscheidungen

- Bootstrap
 - CSS library:
 - Vereinfacht responsive Design
- GitHub
 - Vereinfachte Verwaltung des Source-codes
 - Version Control System
- AJAX
 - Veränderung der HTML ohne nachzuladen

Herausforderungen und Lösungen

Herausforderungen

- Koordinierung des Codes
- Datenbank Korrektur
- Bugfixing

Lösungen:

- Regelmäßige Treffen zur Problembehebung
- Nutzung der Website
- GitHub



Vielen Dank für Ihre
Aufmerksamkeit