

DOKUMENTATION

„Erinnerungsliste“

Kathrin Falch

ON21

26.November 2022

1. Thema des Projekts

2. Ausgangssituation

3. Vorgehen

4. Anforderungsliste

5. Konzeption

5.1. Technologie- / Werkzeugwahl

5.2. Entwurf

6. Ergebnis des Projekts

7. Reflexion

7.1. Herausforderungen

7.2. Unterstützung

7.3. Lernerfolge / Fazit

8. Quellen

A. Installationsanleitung





B. Benutzerdokumentation

1. Thema des Projekts

Das Thema meines Projektes ist eine Erinnerungsliste mit Login-Funktion. Die Erinnerungsliste soll dabei helfen, kleine Erinnerungen mit einer Notiz zu dokumentieren und sie schließlich entweder zu bearbeiten oder als erledigt zu markieren und somit zu löschen.

Für das Thema habe ich mich entschieden, da ich selbst sehr vergesslich bin und gerne die „Erinnerungen-App“ auf meinem iPhone benutze. Eine Nachentwicklung von dieser Idee hat mich sehr interessiert, da ich wissen wollte, wie der Hintergrund einer solchen Applikation aussieht.

2. Ausgangssituation

	HTML	CSS	Bootstrap	PHP
Skillsterne				

Vorerfahrung im Bereich Programmieren bzw. PHP und Datenbanken habe ich keine. Ebenfalls fällt mir teilweise das Verstehen, Unterscheiden und Anwenden der verschiedenen Programmiersprachen schwer.

3. Vorgehen

Um mich mit dem Thema bekannt zu machen, habe ich mir zunächst viele YouTube-Tutorials angesehen. Es war für mich von Anfang an klar, dass ich eine Webanwendung mit PHP realisieren möchte, da mir diese Programmiersprache in den jeweiligen Vorlesungen am verständlichsten erläutert wurde. Dass ich jedoch aufgrund meiner sehr geringen Programmiererfahrung viel Starthilfe benötigte, war mir ebenfalls von Beginn

an bewusst. Nachdem ich meine nötigen Ordner erstellt habe, legte ich in der Datei „docker-compose.yml“ meine Datenbankdaten fest. Anschließend öffnete ich Docker Desktop und konnte durch die Terminal-Eingabe „docker-compose up“ starten. Nach der Fertigstellung des Logins erfolgte das Erstellen der Funktion, Aufgaben in die Erinnerungsliste anzulegen. Schließlich kamen das Verarbeiten, Bearbeiten, Anzeigen und Löschen. Damit waren alle CRUD-Methoden abgedeckt, die ich jedoch nicht ohne diverse YouTube Tutorials und Hilfe von Kommilitonen geschafft hätte. Dadurch, dass wir uns zusammen an unseren Code gesetzt haben, konnten wir uns gegenseitig bei Fehlercodes aushelfen und Lösungen finden, was mir den Wissenserwerb und die Wissensweitergabe ermöglicht hat.

4. Anforderungsliste

	Anforderung	Fertigstellungsgrad
1	Login- und Logout Funktion	Funktioniert
2	Erinnerungen anlegen	Funktioniert
3	Erinnerungsliste ansehen	Funktioniert
4	Erinnerungen bearbeiten	Funktioniert
5	Erinnerungen löschen	Funktioniert
6	AJAX-Einbindung	Funktioniert nur teilweise

Eine sinnvoll asynchrone Kommunikation (AJAX) habe ich nur für die „aufgabe_löschen.php“-Funktion umsetzen können. Dazu nutzte ich jQuery, da ich dachte, dass dies einfacher wäre. Ein Kommilitone erklärte mir JavaScript und so konnte ich mit dessen Hilfe AJAX einsetzen. Tatsächlich habe ich den Einsatz für alle weiteren Funktionen nicht mehr verstanden und den Durchblick schlussendlich verloren, weshalb ich mich dazu entschied, aufzugeben. Daher gilt der Fertigstellungsgrad für diese Anforderung leider als unfertig.

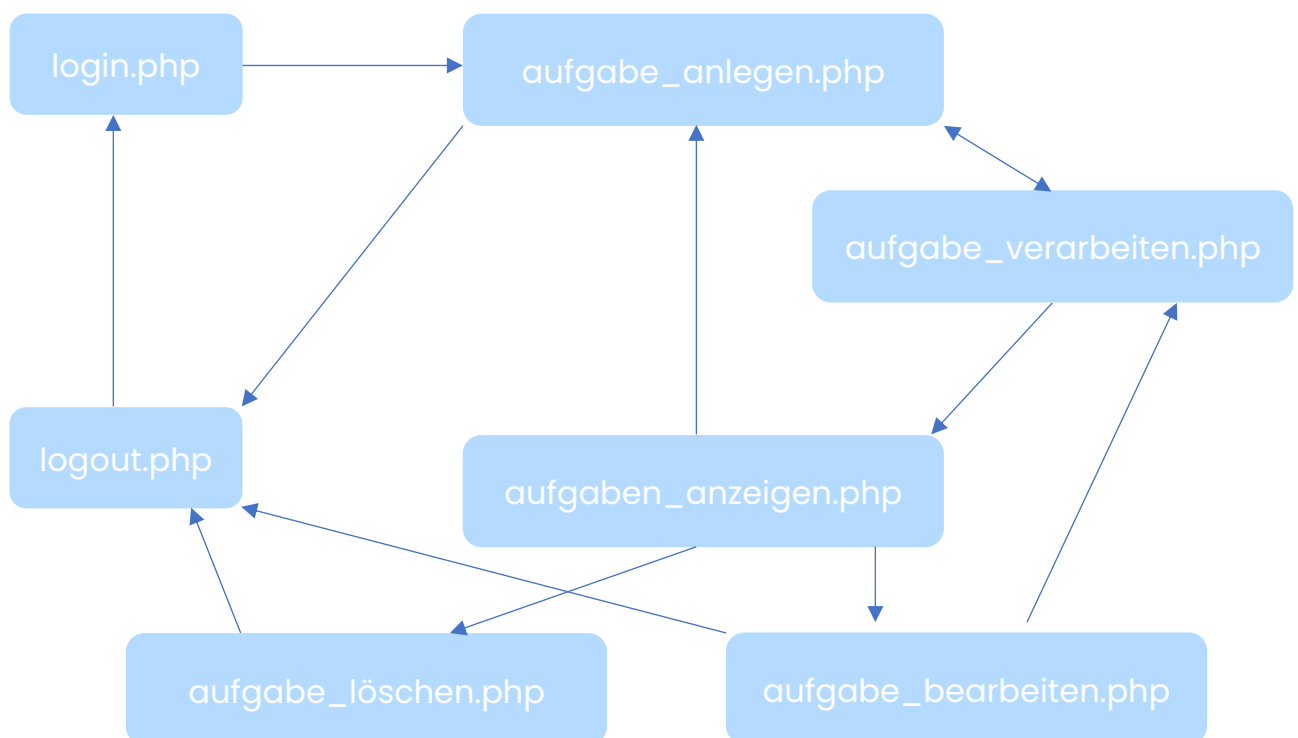
5. Konzeption

5.1. Technologie-/Werkzeugauswahl

Für die Entwicklung der Webanwendung habe ich plain PHP mit MySQL als Datenbank über Docker benutzt. Werkzeuge, die mir beim Codieren geholfen haben, waren YouTube, W3Schools, das classless CSS Framework water.css und meine Kommilitonen. Ich habe sehr viel Input benötigt, um zu verstehen, wie ich meine Erinnerungs-Anwendung aufzubauen habe. Durch die Visualisierungen und Erklärungen auf YouTube verstand ich schließlich mehr. Da das Aussehen nicht bewertet wurde, übernahm ich das CSS Framework von water.css. Dadurch konnte ich mir etwas Zeit sparen und mich voll auf meinen Code konzentrieren. Meine Kommilitonen und ich hatten keine Vorerfahrung beim Programmieren, wodurch wir uns gegenseitig halfen und unterrichteten.

5.2. Entwurf

Architektur

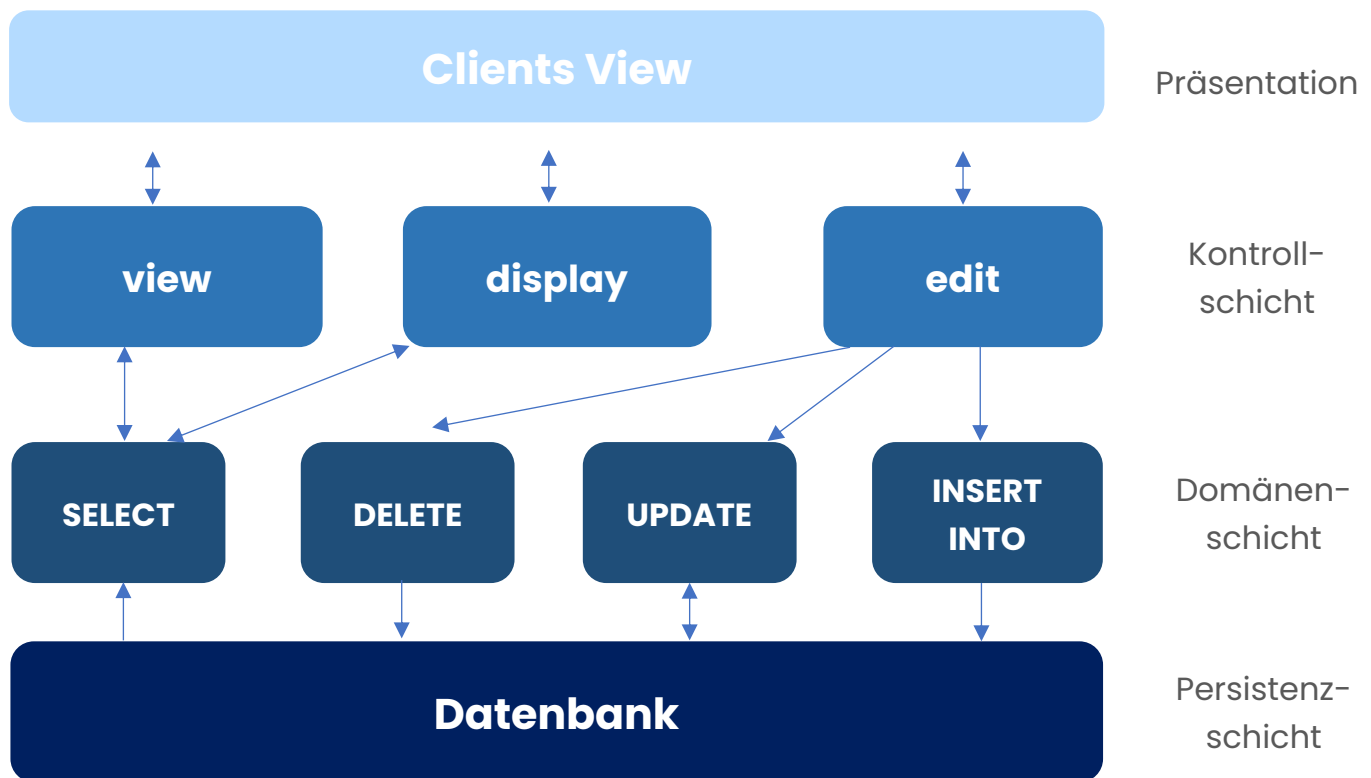


Nach einem erfolgreichen Login auf der login.php wird man weitergeleitet auf aufgabe_anlegen.php um eine Erinnerung zu hinterlegen. Hier besteht sowohl die Möglichkeit, mich durch einen Klick auf den Abmelde-Button auszuloggen, aber auch das Erinnerungsformular auszufüllen und dadurch dann auf aufgabe_verarbeiten.php zu gelangen. Hier habe ich wiederum die Möglichkeiten, entweder eine neue Erinnerung anzulegen (aufgabe_anlegen.php) oder mir die gesamte Erinnerungsliste anzeigen zu lassen (aufgaben_anzeigen.php). Lasse ich mir die gesamte Liste auf aufgaben_anzeigen.php anzeigen, kann ich die Aufgabe entweder bearbeiten (aufgabe_bearbeiten.php) und nach der Eingabe automatisch wieder bei aufgabe_verarbeiten.php landen, oder eine Aufgabe als erledigt markieren und somit löschen (aufgabe_loeschen.php). Hier besteht, wie auch bei aufgabe_bearbeiten.php, die Möglichkeit, mich wieder auszuloggen und wieder bei der Anmelde-Maske zu landen.

Dank `header(„xyz.php“); exit;` konnte ich eine direkte Überleitung verwirklichen, was mir am Anfang noch ein Rätsel war. Hierbei half Google jedoch gut weiter.

Das User Interface ist einfach gehalten und die Bedienung durch den Nutzer recht leicht verständlich. Durch water.css wurde das UI auch optisch unterstützt.

Datenmodell / Persistenzüberlegung



6. Ergebnis des Projekts

Das Projekt ist dank diverser YouTube Tutorials plastisch erfahrbar. Als User versteht man einen Zusammenhang innerhalb der Anwendung. Die Applikation läuft mithilfe von Docker-Containern. Der Nutzer ist in der Lage, sich einzuloggen, Erinnerungen mit einer kleinen Notiz und Wertung der Dringlichkeit zu hinterlegen, die Liste aufzurufen und Erinnerungen zu bearbeiten sowie zu löschen.

7. Reflexion

7.1 Herausforderungen

Meine größte Herausforderung während des gesamten Projektes war meine fehlende Vorerfahrung. Tatsächlich merke ich oft selbst, wie schnell ich mit den verschiedenen Programmiersprachen und Funktionen durcheinanderkomme. Manchmal fehlt das Verständnis zum Zusammenhang diverser Codepassagen. Ebenfalls hat mich mein Zeitmanagement stark herausgefordert. Das Abarbeiten anderer Projekte ließ mich die benötigte Zeit für das T4-Projekt falsch einschätzen, wodurch ich letzten Endes nicht mehr genug Zeit für die ausführliche Auseinander- und Durchsetzung mit der AJAX-Einbindung hatte. Trotzdem habe ich in jedem Punkt versucht, durch viele Erklärvideos und Hilfestellungen seitens meiner Kommilitonen, das Beste aus meiner Ausgangssituation zu machen.

7.2. Unterstützung

Schon früh war ich abhängig von dem Teamgeist meiner Kommilitonen Sarah Wirth, Andreas Matejcek und Botan Mustafa. Als wir nicht mehr weiterwussten und selbst Google keine Antworten mehr hatte, trafen wir uns gemeinsam auf Discord und erklärten uns verschiedene Probleme gegenseitig. Vor Allem konnten Fabian Naglo und Sven Schuppel uns aus aussichtslosen Situationen helfen, da beide schon die nötigen Vorerfahrungen haben. Botan Mustafa und ich konnten die meisten Passagen überwiegend nur mit YouTube Tutorials realisieren, weshalb sich unser Code ähnelt. Wir konnten jedoch Zusammenhänge verstehen und sogar Probleme selbst lösen, was uns beide sehr stolz gemacht hat, da das Projekt für uns anfangs unmöglich wirkte. Als wir allmählich „den Dreh raus hatten“ machte das gemeinsame Arbeiten sogar sehr viel Spaß, da wir endlich Zusammenhänge erkannten und kleine Erfolge feierten, als unsere Webanwendung endlich funktionierte.

7.3. Lernerfolge / Fazit

Als Fazit des Projektes nehme ich mit, dass Teamarbeit und gegenseitiges Helfen sehr wichtig sind. Es sei zu erwähnen, dass mir bewusst ist, dass das Nacharbeiten und Zusammenarbeiten nicht der Sinn dieses Projektes sind. Jedoch bin ich froh, dass es dadurch trotzdem geschafft habe, eine laufende Webanwendung zu entwickeln und den Hintergrund dadurch verstanden zu haben. Selbst, wenn ich nicht fertig geworden bin, macht es mich stolz, solange am Code getüftelt zu haben, bis jede Funktion genau so funktioniert hat, wie ich sie mir vorgestellt hatte. Außerdem bin ich meinen Kommilitonen, die bereits Vorerfahrung hatten, sehr dankbar für jede helfende Minute. Das Programmieren wird mir in Zukunft zwar immer noch nicht besonders liegen, jedoch würde ich behaupten, dass ich nun PHP-Grundbausteine verstehen und wiedergeben kann. Allein das ist für mich Erfolg genug.

8. Quellen

<https://www.youtube.com/watch?v=Y9yE98etanU&t=4s>

<https://www.youtube.com/watch?v=5L9UhOnuos0&t=2003s>

<https://www.youtube.com/watch?v=pdsfCLtNqus>

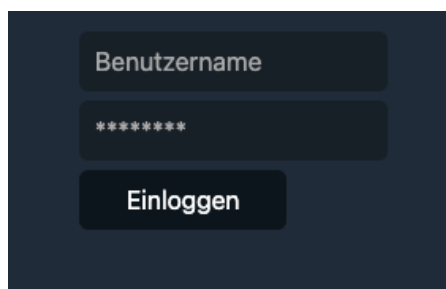
<https://youtu.be/NqP0-UkIQS4>

A. Installationsanleitung

1. Die Zip-Datei des Projekt Ordners entpacken.
2. Docker Desktop starten.
3. Visual Studio Code starten.
4. Das Terminal in der Console aufrufen und **docker-compose-up** eingeben.
5. Den Docker Desktop beobachten und warten, bis **Running** erscheint.
6. Nun ist ein Zugriff auf localhost:8000 möglich.
7. Jetzt muss der Nutzer k_erinnerungen.sql und k_user.sql in dem Ordner öffnen und in die Datenbank einbringen.
8. Auf localhost:8080 kann man sich nun mit den Anmeldedaten
Benutzer: kathrin
Passwort: kathrinphp
anmelden.

B. Benutzerdokumentation

Login



Benutzername

Einloggen

Benutzername: admin

Passwort: test

```
login.php .../php X login.php projekt
projekt > php > login.php
1  ?php
2  session_start();
3
4  require("conn.php");
5
6  $login_user    = $_POST["username"];
7  $login_password = $_POST["password"];
8
9  $abfrage = "SELECT * FROM k_user WHERE username = '" . $login_user . "' ";
10 $result = $sql->query($abfrage);
11 $user = $result->fetch_assoc();
12
13 if($user){
14     if(password_verify($login_password, $user['password'])){
15         $_SESSION['userid'] = $user['id'];
16         header('Location: ../php/aufgabe_anlegen.php');
17         exit();
18     }
19     else{
20         echo "Passwort stimmt nicht.  

21         Jetzt <a href = '../login.php'> nochmal probieren </a>.";
22     }
23 }
24 else{
25     echo " <br> User nicht gefunden";
26     echo " <br> Jetzt <a href = '../login.php'> nochmal probieren </a>.";
27 }
28 >
29
30 !DOCTYPE html>
31 html lang="de">
32 <head>
33     <meta charset="utf-8">
34     <meta name="viewport" content="width=device-width, initial-scale=1.0">
35     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/water.css@2/out/water.css">
36 </head>
37 <body>
38 </body>
39 </html>
```

Der Nutzer „admin“ loggt sich mit seinem Passwort „test“ bei der Erinnerungsliste an. Der Server zieht aus der Datenbank k_user die nötigen Daten um zu überprüfen, ob die Eingaben korrekt oder inkorrekt sind.

Erinnerungen eintragen

Stimmen die Eingaben, erfolgt eine Weiterleitung zu /aufgabe_anlegen.php über die Datenbank k_erinnerungen, wo der User eine neue Erinnerung anlegen kann.

The screenshot shows a web form titled "Meine Erinnerungen eintragen" with a dark blue header. In the top right corner of the header is a button labeled "Abmelden". The form contains three main sections: 1. "Erinnerung" with a text input field. 2. "Notiz" with a large text area. 3. "Dringlichkeit" with a dropdown menu currently set to "Mittel". At the bottom of the form is a button labeled "Bestätigen".

```
projekt > php >  aufgabe_anlegen.php
1  <?php
2      session_start();
3  >
4
5  <!DOCTYPE html>
6  <html lang="de">
7      <head>
8          <meta charset="utf-8">
9          <meta name="viewport" content="width=device-width, initial-scale=1.0">
10         <link rel="stylesheet"
11             href="https://cdn.jsdelivr.net/npm/water.css@2/out/water.css">
12         <title>Erinnerungsliste</title>
13     </head>
14     <body>
15
16         <div style="display: flex; justify-content: space-between; margin-bottom: 10px;">
17             <h1>Meine Erinnerungen eintragen</h1>
18             <button><a href='../logout.php'>Abmelden</a></button>
19         </div>
20
21         <br>
22
23         <form action="aufgabe_verarbeiten.php" method="post">
24             <label for="erinnerung">Erinnerung</label>
25             <input type="text" id="erinnerung" name="erinnerung">
26
27             <label for="notiz">Notiz</label>
28             <textarea id="notiz" name="notiz"></textarea>
29
30             <label for="dringlichkeit">Dringlichkeit</label>
31             <select id="dringlichkeit" name="dringlichkeit">
32                 <option value="1">Niedrig</option>
33                 <option value="2" selected="selected">Mittel</option>
34                 <option value="3">Hoch</option>
35             </select>
36
37             <br>
38
39             <button>Bestätigen</button>
40         </form>
41     </body>
42 </html>
```

Der User kann die Erinnerung eintragen, eine Notiz dazu festhalten und zwischen den Dringlichkeitswerten „Niedrig“, „Mittel“ und „Hoch“ entscheiden und seine Eingaben anschließend mit „Bestätigen“ speichern. Hier kann er sich außerdem auch umentscheiden und sich abmelden.

Erinnerung erfolgreich gespeichert

Eingabe erfolgreich gespeichert.
Eine neue [Erinnerung anlegen](#) .
Klicke [hier](#) um dir deine Erinnerungsliste anzusehen.

```
projekt > php > aufgabe_verarbeiten.php
1  ?php
2      session_start();
3
4      $erinnerung = $_POST["erinnerung"];
5      $notiz = $_POST["notiz"];
6      $dringlichkeit = filter_input(INPUT_POST, "dringlichkeit", FILTER_VALIDATE_INT);
7
8      $db_host = "db";
9      $db_user = "kathrin";
10     $db_passwort = "kathrinphp";
11     $db_name = "erinnerungsliste_kathrinphp";
12
13     $conn = new MySQLi($db_host, $db_user, $db_passwort, $db_name);
14
15     if (mysqli_connect_errno()) {
16         die("Verbindungsfehler: " . mysqli_connect_error());
17     }
18
19     $sql = "INSERT INTO k_erinnerung (erinnerung, body, dringlichkeit)
20         VALUES (?, ?, ?)";
21
22     $stmt = mysqli_stmt_init($conn);
23
24     if ( ! mysqli_stmt_prepare($stmt, $sql)) {
25         die(mysqli_error($conn));
26     }
27
28     mysqli_stmt_bind_param($stmt, "ssi", $erinnerung, $notiz, $dringlichkeit);
29
30     mysqli_stmt_execute($stmt);
31
32     echo "Eingabe erfolgreich gespeichert.";
33     echo "<br> Eine neue <a href = '../php/aufgabe_anlegen.php'> Erinnerung anlegen";
34     echo "<br> Klicke <a href = '../php/aufgaben_anzeigen.php'> hier </a> um dir deine Erinnerungsliste anzusehen.";
35
36
37 <!DOCTYPE html>
38 <html lang="de">
39     <head>
40         <meta charset="utf-8">
41         <meta name="viewport" content="width=device-width, initial-scale=1.0">
42         <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/water.css@2/out/water.css">
43     </head>
44     <body>
45     </body>
46 </html>
```

Nach dem Bestätigen erscheint die Oberfläche der PHP-Funktion `aufgabe_verarbeiten`. Hier ist eine Einbindung zur Datenbank gegeben, weshalb die Eingaben in dieser gespeichert sind und unter `localhost:8080` auch ersichtlich sind (`INSERT INTO`).

Sollte man eine neue Erinnerung anlegen wollen, bietet die direkte Verlinkung zu `/php/aufgabe_anlegen.php` diese Möglichkeit. Durch einen Klick auf „hier“ kann der User seine Erinnerungsliste ansehen. Dies geschieht über die Verlinkung zu `/php/aufgaben_anzeigen.php`.

Erinnerungsliste anzeigen

Alle Erinnerungen

Abmelden

Neue Erinnerung

Nummer	Erinnerung	Notiz	Dringlichkeit	
59	Einkaufen	Wasser, Katzenfutter, SmoothieBowl	2	Bearbeiten Erledigt
61	Katze zum Tierarzt bringen	Kastration kostet 110 €!	3	Bearbeiten Erledigt

```
projekt > php >  aufgaben_anzeigen.php
1  <?php
2      session_start();
3  ?>
4
5  <!DOCTYPE html>
6  <html lang="de">
7      <head>
8          <meta charset="utf-8">
9          <meta name="viewport" content="width=device-width, initial-scale=1.0">
10         <link rel="stylesheet"
11             href="https://cdn.jsdelivr.net/npm/water.css@2/out/water.css">
12     </head>
13     <body style="margin: 50px;">
14
15         <div style="display: flex; justify-content: space-between; margin-bottom: 5px;">
16             <h1>Alle Erinnerungen</h1>
17             <button><a href='../logout.php'>Abmelden</a></button>
18         </div>
19
20         <button><a href='../php/aufgabe_anlegen.php'>Neue Erinnerung</a></button>
21
22         <table class="table">
23             <thead>
24                 <tr>
25                     <th>Nummer</th>
26                     <th>Erinnerung</th>
27                     <th>Notiz</th>
28                     <th>Dringlichkeit</th>
29                 </tr>
30             </thead>
31
32             <tbody>
33                 <?php
34                     $db_host =      "db";
35                     $db_user =      "kathrin";
36                     $db_passwort =  "kathrinphp";
37                     $db_name =      "erinnerungsliste_kathrinphp";
38
39                     $conn = new MySQLi($db_host, $db_user, $db_passwort, $db_name);
40
41                     if ($conn->connect_error) {
42                         die ("Verbindung fehlgeschlagen: " . $conn->connect_error);
43                     }
44
45                     $sql = "SELECT * FROM k_erinnerung";
46                     $result = $conn->query($sql);
47
48                     if (!$result) {
```

```

49         die("Ungültig: " . $conn->error);
50     }
51
52     while($row = $result->fetch_assoc()) {
53         echo "<tr>
54             <td>" . $row ["id"] . "</td>
55             <td>" . $row ["erinnerung"] . "</td>
56             <td>" . $row ["body"] . "</td>
57             <td>" . $row ["dringlichkeit"] . "</td>
58             <td>
59                 <a href='/php/aufgabe_bearbeiten.php?aufgabe=" . $row["id"] . "'>Bearbeiten</a> <br>
60                 <a class='erledigt-button' href='/php/aufgabe_loeschen.php?aufgabe=" . $row["id"] . "'
61             </td>
62         </tr>";
63     }
64     ?>
65 </tbody>
66 </table>
67 <script src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
68 <script language="javascript" type="text/javascript" src="/script.js"></script>
69 </body>
70 </html>

```

Die Erinnerungsliste greift auf die Datenbank k_erinnerung und kann so wiedergeben, welche Erinnerungen noch nicht erledigt wurden. Der User kann nach diesem Einblick entscheiden, ob er sich abmelden will, eine neue Erinnerung anlegen möchte, eine bereits bestehende Erinnerung bearbeiten will oder als erledigt markieren will. Entscheidet er sich für die ersten beiden Fälle, landet er entweder wieder auf /php/aufgabe_anlegen.php oder wieder im Login. Sollte er eine Erinnerung bearbeiten wollen, wird er zu /php/aufgabe_bearbeiten.php navigiert.

Erinnerungen bearbeiten

Meine Erinnerungen eintragen

Abmelden

Erinnerung

Einkaufen

Notiz

Wasser, Katzenfutter, SmoothieBowl

Dringlichkeit

Mittel

Bestätigen

aufgabe_bearbeiten.php

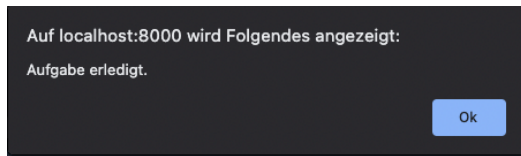
```
php > aufgabe_bearbeiten.php
1  <?php
2      session_start();
3
4      if ( $_SESSION['userid']) {
5          $id = $_GET["aufgabe"];
6
7          require("conn.php");
8
9          $befehl = "SELECT * FROM k_erinnerung WHERE id = $id";
10         $ergebnis = $sql->query($befehl);
11         $ergebnis = $ergebnis->fetch_assoc();
12     }
13     else {
14         header ("location: /login.php");
15         exit;
16     }
17
18
19 <!DOCTYPE html>
20 <html lang="de">
21     <head>
22         <meta charset="utf-8">
23         <meta name="viewport" content="width=device-width, initial-scale=1.0">
24         <link rel="stylesheet"
25             href="https://cdn.jsdelivr.net/npm/water.css@2/out/water.css">
26         <title>Erinnerungsliste</title>
27     </head>
28     <body>
29
30         <div style="display: flex; justify-content: space-between; margin-bottom: 5%;">
31             <h1>Meine Erinnerungen eintragen</h1>
32             <button><a href='../logout.php'>Abmelden</a></button>
33         </div>
34
35         <br>
36
37         <form action="aufgabe_bearbeiten_funktion.php?aufgabe=?php echo $id; ?>" method="post">
38             <input type="hidden" name="id" value=?php echo $id;?>
39             <label for="erinnerung">Erinnerung</label>
40             <input type="text" id="erinnerung" name="erinnerung" value=?php echo $ergebnis["erinnerung"];?>
41
42             <label for="notiz">Notiz</label>
43             <textarea id="notiz" name="notiz">?php echo $ergebnis["body"];?</textarea>
44
45             <label for="dringlichkeit">Dringlichkeit</label>
46             <select id="dringlichkeit" name="dringlichkeit">
47                 <option value="1" <?php if( $ergebnis["dringlichkeit"] == 1) { echo "selected"; } ?>>Niedrig</option>
48
49                 <option value="2" <?php if( $ergebnis["dringlichkeit"] == 2) { echo "selected"; } ?>>Mittel</option>
50                 <option value="3" <?php if( $ergebnis["dringlichkeit"] == 3) { echo "selected"; } ?>>Hoch</option>
51             </select>
52
53             <br>
54             <button>Bestätigen</button>
55
56         </form>
57     </body>
58 </html>
59
```

aufgabe_bearbeiten_funktion.php

```
php > aufgabe_bearbeiten_funktion.php
1  <?php
2      session_start();
3
4      if ( $_SESSION['userid']) {
5          $id = $_GET["aufgabe"];
6
7          require("conn.php");
8
9          $erinnerung = $_POST["erinnerung"];
10         $notiz = $_POST["notiz"];
11         $dringlichkeit = $_POST["dringlichkeit"];
12
13         $befehl = "UPDATE k_erinnerung SET Erinnerung='$erinnerung', body='$notiz', dringlichkeit='$dringlichkeit' WHERE k_id=$id";
14         $ergebnis = $sql->prepare($befehl);
15         $ergebnis = $ergebnis->execute();
16
17         if ($ergebnis) {
18             echo "Eingabe erfolgreich gespeichert.";
19             echo "<br> Eine neue <a href = '../php/aufgabe_anlegen.php'> Erinnerung anlegen </a>.";
20             echo "<br> Klicke <a href = '../php/aufgaben_anzeigen.php'> hier </a> um dir deine Erinnerungsliste anzusehen";
21         }
22     }
23     else {
24         header ("location: /login.php");
25         exit;
26     }
27 ?>
28
29 <!DOCTYPE html>
30 <html lang="de">
31     <head>
32         <meta charset="utf-8">
33         <meta name="viewport" content="width=device-width, initial-scale=1.0">
34         <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/water.css@2/out/water.css">
35     </head>
36     <body>
37     </body>
38 </html>
```

Mit UPDATE ist der Nutzer in der Lage, seine Erinnerung zu bearbeiten. Nachdem der User seine Erinnerung fertig bearbeitet hat, kann er diese wieder bestätigen, wodurch sie in der Datenbank abgeändert und gespeichert wird. Er kann sich außerdem auch direkt ausloggen und landet somit wieder auf der Anmeldemaske.

Erinnerung als erledigt markieren (löschen)



```
1 <?php
2 session_start();
3
4 if ( $_SESSION['userid'] ) {
5     $id = $_GET["aufgabe"];
6
7     require("conn.php");
8
9     $befehl = "DELETE FROM k_erinnerung WHERE id=$id";
10    $ergebnis = $sql->prepare($befehl);
11    $ergebnis = $ergebnis->execute();
12
13    if ($ergebnis) {
14        echo "Aufgabe erledigt.";
15    }
16 }
17 else {
18     echo "Bitte einloggen.";
19 }
20 ?>
```

Dank DELETE FROM kann der User nun seine Erinnerung löschen, indem er auf den Button „Erledigt“ klickt.