Projektdokumentation

Spezifische Funktionalität einer Webapplikation Mit Session-Handling, Authentifizierung und Formularüberprüfung realisieren.

- In meinem Projekt werden alle Benutzereingaben client- und serverseitig validiert.
    - Die Clientseitige Validierung wird durch ein Pattern und eine Maxlength durchgeführt.

```
pattern="(?=.*[a-z])(?=.*[A-Z])[a-zA-Z]{6,}" title="Gross- und Keinbuchstaben, min 6 Zeichen." maxlength="30" required="true">
```

    - Die Serverseitige Validierung wird durch isset, htmlspecialchars, trim, strlen, empty, filter_var, preg_match und password_hash

```
if (isset($_POST['firstname'])) {
  //trim and sanitize
  $firstname = htmlspecialchars(trim($_POST['firstname']));
```

```
//mindestens 1 Zeichen und maximal 100 Zeichen lang, gültige Emailadresse
if (empty($email) || strlen($email) > 100 || filter_var($email, filter: FILTER_VALIDATE_EMAIL) === false) {
  $error .= "Geben Sie bitte eine korrekten Emailadresse ein.<br />";
}
```

```
if (empty($username) || !preg_match( pattern: "/(?=.*[a-z])(?=.*[A-Z])[a-zA-Z]{6,30}/", $username)) {
  $error .= "Geben Sie bitte einen korrekten Usernamen ein.<br />";
}
```

```
$password_hash = password_hash($password, algo: PASSWORD_DEFAULT);
```

- In meinem Projekt wird das Session-Handling korrekt eingesetzt. Ein angemeldeter Benutzer hat Zugriff auf weitere Funktionen, welche nicht angemeldeten Benutzern verwehrt bleiben.
    - In der Navigation werden unterschiedliche Optionen angezeigt abhängig davon, ob man angemeldet ist oder nicht

```
<?php
if (isset($_SESSION['loggedin']) and $_SESSION['loggedin']) {
    echo '<li class="nav-item"><a class="nav-link" href="logout.php">Logout</a></li>';
    echo '<li class="nav-item"><a class="nav-link" href="meineWitze.php">Meine Witze</a></li>';
    echo '<li class="nav-item"><a class="nav-link" href="passwort.php">Passwort ändern</a></li>';
}else {
    // TODO - wenn Session nicht personalisiert
    echo '<li class="nav-item"><a class="nav-link" href="register.php">Registrierung</a></li>';
    echo '<li class="nav-item"><a class="nav-link" href="login.php">Login</a></li>';
}
?>
```

- An anderen Orten wird man direkt auf eine andere Seite weitergeleitet, falls man nicht angemeldet ist

```php
<?php
if (isset($_SESSION['loggedin']) and $_SESSION['loggedin']) {
    echo '<li class="nav-item"><a class="nav-link" href="logout.php">Logout</a></li>';
    echo '<li class="nav-item"><a class="nav-link" href="index.php">Home</a></li>';
    echo '<li class="nav-item"><a class="nav-link" href="passwort.php">Passwort ändern</a></li>';
}else {
    header( header: 'Location: index.php');
}
?>
```

- Ein angemeldeter Benutzer kann sich wieder abmelden. Die Session wird dabei korrekt beendet.
    - Bei Ausloggen wird die Session korrekt beendet

```php
<?php
// TODO - Session starten
session_start();
// TODO - Session leeren
$_SESSION = array();
session_destroy();
// TODO - Weiterleiten auf login.php
header( header: 'Location: index.php');
?>
```

- In meinem Projekt werden Script-Injection, Session-Fixation und Session-Hijacking konsequent verhindert.
    - Script-Injection wird htmlspecialchars verhindert
    - Session-Fixation und Session-Hijacking werden mit session_regenerate_id(true); beim Erstellen der Session verhindert.

Schutz und sicherheitswürdige Informationen unter Beachtung des Datenschutzes identifizieren und Massnahmen definieren

- In meinem Projekt werden sensible Daten wie das Passwort mit sicheren und aktuellen Methoden gehasht und gesaltet.
    - Das Passwort wird mit password_hash($password, PASSWORD_DEFAULT); gehashed und gesalted

Anbindung der Web-Applikation an die Datenbank realisieren, Datenschutz und Datensicherheit beachten

- Ein Benutzer kann sich an meinem Projekt registrieren. Dazu erfasse ich sinvolle Daten des Benutzers.
    - Es werden folgende Daten erfasst:
        - Vorname
        - Nachname
        - Benutzername
        - Passwort
        - E-Mail-Adresse

```
'Insert into tbl_benutzer (vorname, nachname, benutzername, passwort, email)
```

- Ein Benutzer kann sich an meinem Projekt anmelden. Nach der Anmeldung stehen dem Benutzer weitere Funktionen zur Verfügung.
    - Nach dem registrieren und Einloggen können eigene Witze erfasst, bearbeitet und gelöscht werden.

```
if (isset($_SESSION['loggedin']) and $_SESSION['loggedin']) {
    echo '<li class="nav-item"><a class="nav-link" href="logout.php">Logout</a></li>';
    echo '<li class="nav-item"><a class="nav-link" href="meineWitze.php">Meine Witze</a></li>';
    echo '<li class="nav-item"><a class="nav-link" href="passwort.php">Passwort ändern</a></li>';
```

```
<a href="bearbeiten.php?id='.$row["id"].'">bearbeiten</a> </BLOCKQUOTE> <a href="loeschen.php?id='.$row["id"].'">löschen</a>'
```

- In meinem Projekt kann ein angemeldeter Benutzer sein Passwort ändern.

```
'UPDATE tbl_benutzer SET passwort = ? WHERE id = ?";
```

- In meinem Projekt können angemeldete Benutzer weitere Informationen in der Datenbank erfassen. Diese Datensätze können ausschliesslich vom Ersteller dieser Datensätze einzeln geändern und gelöschen werden.
    - Die Witze können ausschliesslich von den Erstellern bearbeitet und gelöscht werden, die geschieht dadurch, dass die entsprechenden Seiten die ID des Witzes und aus der Session die BenutzerID erhalten.

```
"UPDATE tbl_witze SET titel = ?, inhalt = ? WHERE id = ? and benutzerId = ?";
```

- In meinem Projekt können diese zusätzlichen Informationen von nicht angemeldeten Benutzern angesehen werden.
    - Man wird automatisch auf index.php umgeleitet sollte man nicht angemeldet sein

```php
<?php
if (isset($_SESSION['loggedin']) and $_SESSION['loggedin']) {
    echo '<li class="nav-item"><a class="nav-link" href="logout.php">Logout</a></li>';
    echo '<li class="nav-item"><a class="nav-link" href="index.php">Home</a></li>';
    echo '<li class="nav-item"><a class="nav-link" href="passwort.php">Passwort ändern</a></li>';
}else {
    header( header: 'Location: index.php');
}
?>
```

- In meinem Projekt wird SQL-Injection konsequent verhindert.
    - Die SQL- Injection wird durch Prepared Statemnts verhindert, bei welchen die Benutzereingaben nur in das SQL-Statement eingefügt werden, welche dem Server schon bekannt sind. Es wird also kein anderes Statement als das vordefinierte ausgeführt.

```php
// Query vorbereiten
$query = "UPDATE tbl_witze SET titel = ?, inhalt = ? WHERE id = ? and benutzerId = ?";
$stmt = $mysqli->prepare($query);
if ($stmt === false) {
    $error .= 'prepare() failed ' . $mysqli->error . '<br />';
}
// Parameter an Query binden
if (!$stmt->bind_param( types: "ssii", &var1: $titel, &...: $inhalt, $id, $user_id)) {
    $error .= 'bind_param() failed ' . $mysqli->error . '<br />';

}// Query ausführen
if (!$stmt->execute()) {
    $error .= 'execute() failed ' . $mysqli->error . '<br />';
}
```