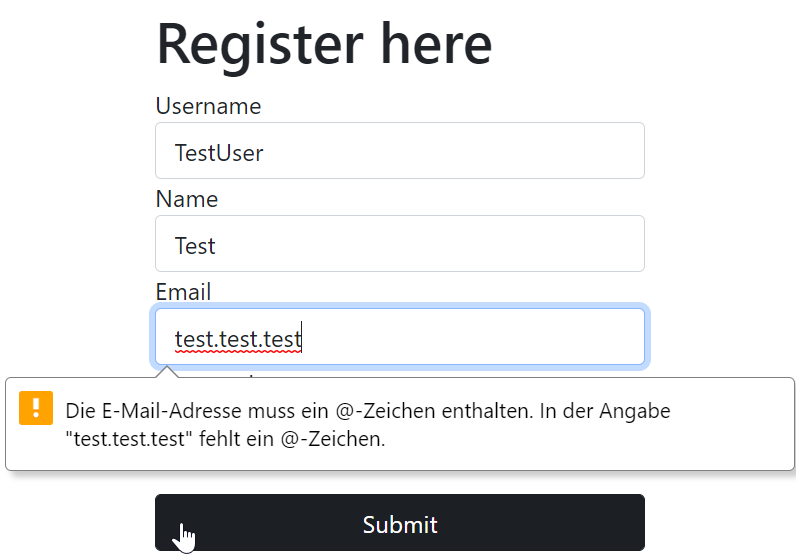
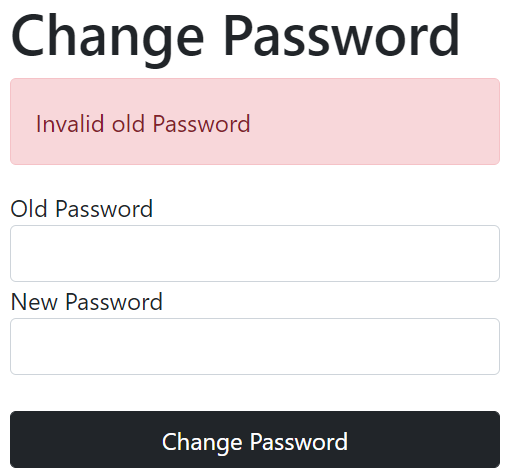
## Achtet auf Usability der Applikation, leitet den Benutzer mit sinnvollen Meldungen durch alle Prozesse.



Eine Clientseitige Fehlermeldung erscheint, da der User nicht eine Korrekte E-Mail-Adresse angegeben hat



Bei Angabe eines falschen alten Passworts wird dem User mit einer Fehlermeldung mitgeteilt, dass die Eingabe nicht korrekt war.

        if (strlen($error)) {

            echo "<div class=\"alert alert-danger\" role=\"alert\">" . $error . "</div>";

        }

Bei Eingabe eines falschen Wertes wird mit diesem if statement eine Fehlermeldung ausgegeben.

## Prüft alle Benutzereingaben client- und serverseitig (Pflichtfelder, Länge, Datentyp). Verwendet sinvolle Datenypen und Speichergrössen zur Speicherung der Informationen.

<input type="text" name="userid" class="form-control" id="userid"

pattern="^[a-zA-Z]+$" minLength="3" maxLength="20" value="" required />

Die Benutzereingaben werden mit den Angaben pattern, minLength, maxLength und required im <input/> clientseitig validiert.

  $passwordPattern = "/^[ -~]+$/";

Hier bilden wir ein Pattern welches für das Passwort benötigt wird.

   if (

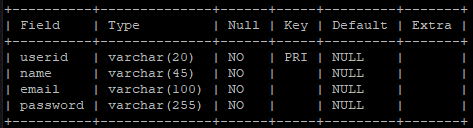
        !empty($user["password"]) &&

        strlen($user["password"]) >= 8 && strlen($user["password"]) <= 255 &&

        preg\_match($passwordPattern, $user["password"])

    )

Serverseitig wird der mitgegebene Wert überprüft ob dieser die richtige Länge enthält. Danach wird der mitgegebene Wert mit dem Pattern geprüft.



Für jede Information nutzen wir eine jeweilig unterschiedliche Speichergrösse welche für die Information am besten geeignet ist.

## Verwendet Sessionhandling zur Sitzungsverfolgung und stellt für authorisierte Benutzer neben der Möglichkeit das Passwort zu ändern, zusätzliche Funktionen zur Verfügung.

if (isset($\_SESSION['loggedin']) and $\_SESSION['loggedin'])

Die ganze Applikation ist erst nutzbar wenn der User sich registriert und angemeldet hat.

                session\_start();

                $\_SESSION['loggedin'] = true;

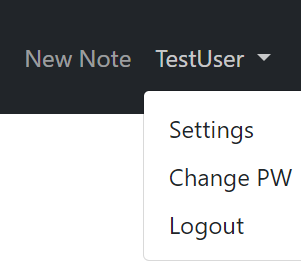
                $\_SESSION['userid'] = $firstRow['userid'];

                $\_SESSION['name'] = $firstRow['name'];

                $\_SESSION['email'] = $firstRow['email'];

                session\_regenerate\_id();

Nachdem sich der User eingeloggt hat, wird die Session gestartet.

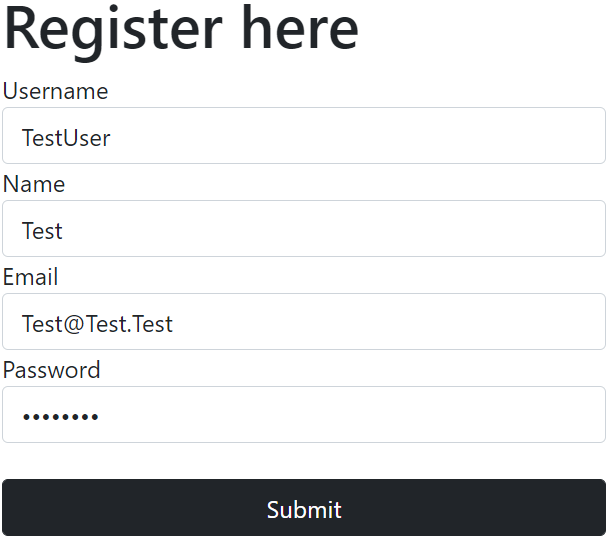


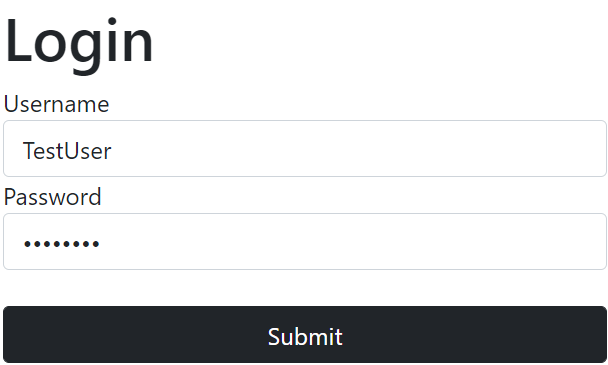
Falls der Benutzer eingeloggt ist, kann er die Applikation nutzen und Notizen erstellen. Zudem erhält er die Möglichkeit sein Password, seinen Namen oder seine E-Mail-Adresse zu ändern. In Settings gibt es auch die Möglichkeit den eigenen Benutzer zu löschen. Wenn der Benutzer sich ausloggt, sind diese Möglichkeiten nicht mehr verfügbar, sondern nur noch die Möglicheiten zum Login oder zur Registrierung.

## Speichert schutzwürdige Informationen sicher und nach neustem Stand der Technik in der Datenbank. Achtet darauf, das der Datenbankbenutzer die für die Kommunikation zwischen Web-Applikation und Datenbank minimal nötigen Berechtigungen besitzt.

## Verhindert SQL-Injection, Script-Injection und Session-Hijacking.

## Bietet die Möglichkeit zur Registrierung und Login an der Web-Applikation an. Dazu werden sinnvolle Informationen zum Benutzer erfasst.

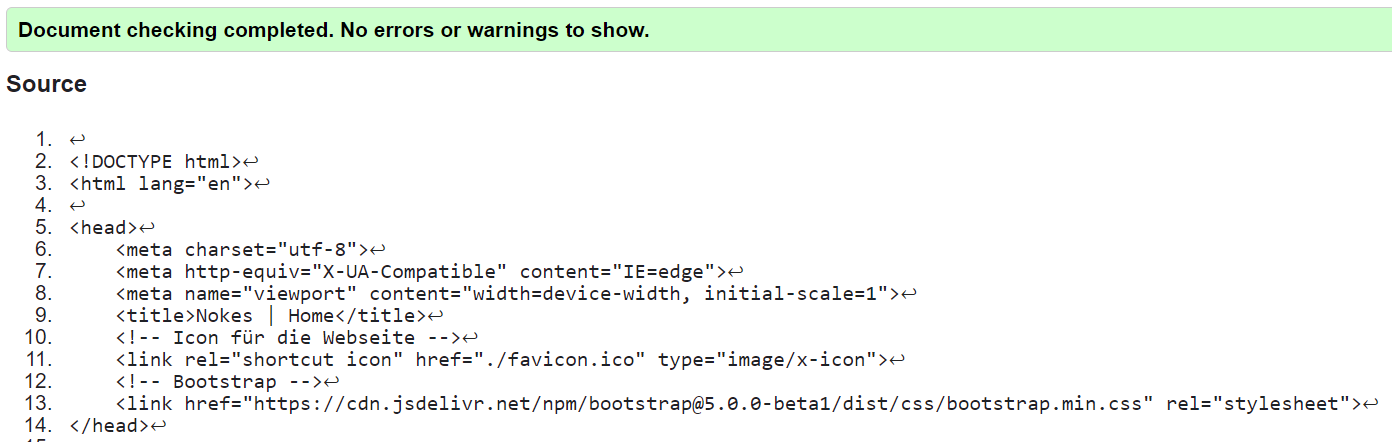
Bei der Registrierung muss der User seinen Username, seinen Namen, seine E-Mail-Adresse und ein Passwort angeben.

Beim Login benötigt der User nur noch seinen Username (userid) und sein Passwort um sich bei der Applikation anzumelden. Nach der Anmeldung wird die Session gestartet.

## Ermöglicht dass Erfassen, Ändern und Löschen zusätzlicher Daten über die Webseite in der Datenbank. Dieses Daten können von nicht autorisierten Benutzern angesehen werden.

## Kann einen Testplan für die Applikationen erstellen und überprüft die für dieses Modul relevanten Funktionen der Applikation anhand dieses Testplanes.

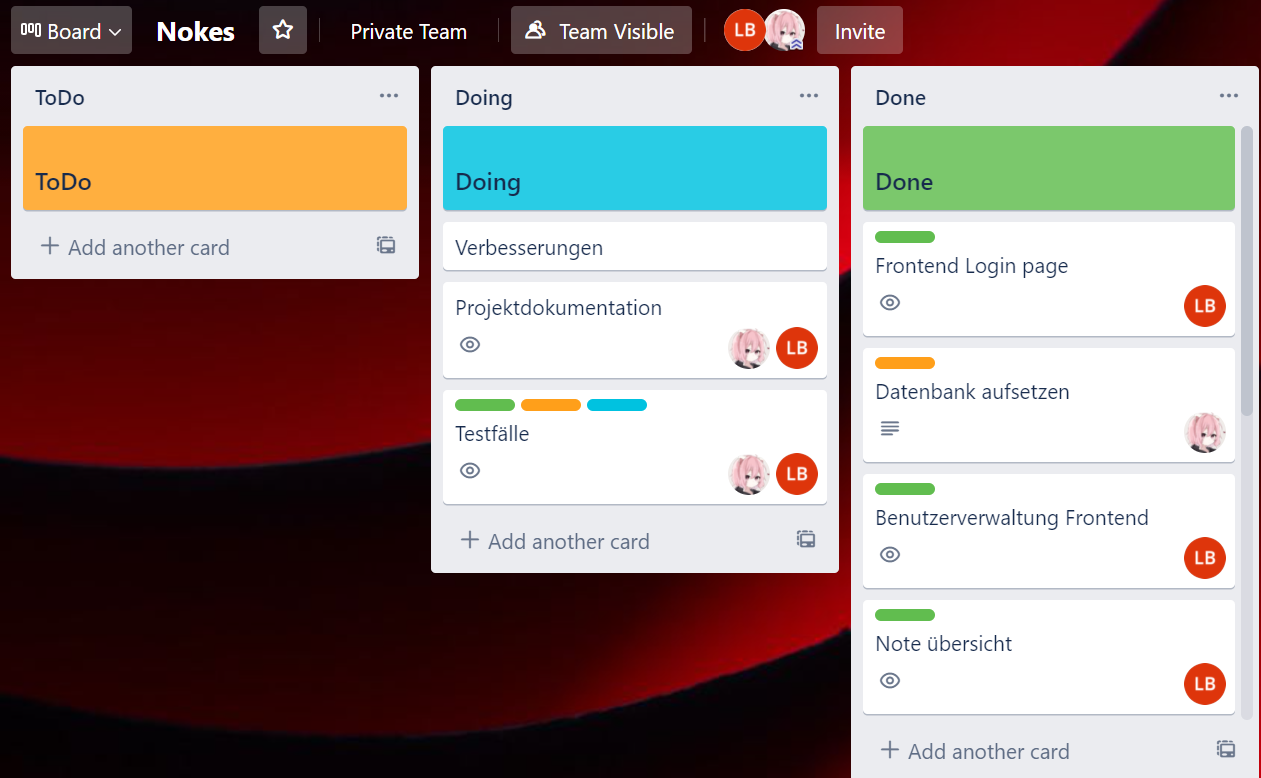
## Kennt Tools zur Validierung von HTML und CSS und validiert seinen / ihren Code.



Der Code wurde mithilfe des HTML Validators von <https://validator.w3.org/> validiert. Keine Fehlermeldungen oder Warnungen wurden erkannt.

CSS mussten wir nicht validieren, da wir Bootstrap nutzten.

## Setzt für die Projektplanung ein geeignetes Planungstool ein und verwendet dieses aktiv während der Entwicklung des Projektes.

Für die Projektplanung nutzen wir das Tool Trello, da wir beide auch schon miteinander Projekte mit diesem Tool durchgeführt haben. Durch Trello hatten wir beide die Möglichkeit unseren Fortschritt grob dem Anderen mitzuteilen.

## Kennt Möglichkeiten zur Strukturierung von Quellcode und der Umsetzung unter Berücksichtigung von Codierungsrichtlinien und wendet diese in seinem Projekt an.

Wir versuchten, logische Funktionen ohne HTML in getrennte auszulagern.

function loginUser($loginUser)

{

    // Database connection

    include('sqlConnection.php');

    $errorString = '';

Bei der Benennung der Variablen und Funktionen hielten wir uns an die gängigen Methoden. Alles sollte einen möglichst simplen und selbsterklärenden Namen bekommen.

## Kommentiert seinen Code.

    // Validation of userid

    // not empty

    // length: 3-20

    // only letters($letterOnly regex)

    $trimmedUserId = trim($user["userid"]);

Code wurde an den benötigten Stellen kommentiert.