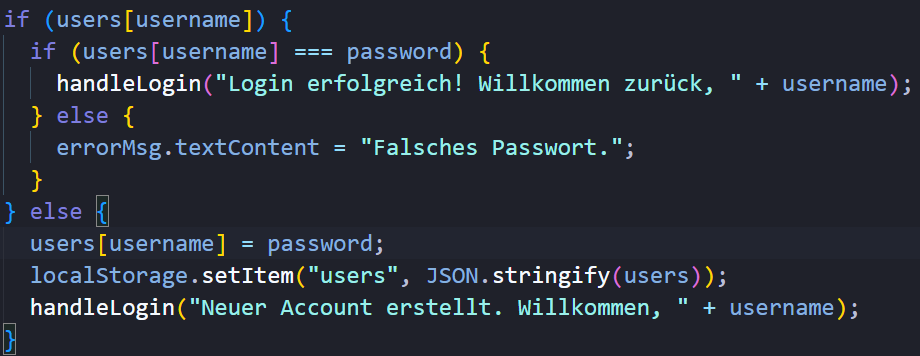
**Lab01**

**The Login Page**

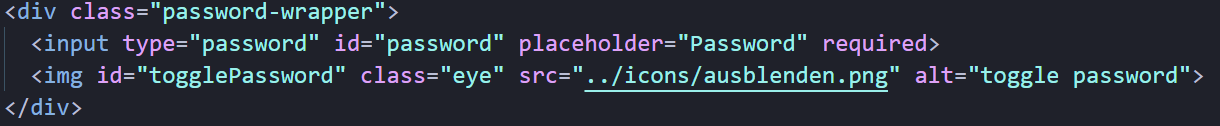
First the user gets prompted to input a name and a password, this password is saved in the localStorage and later used for the leaderboard. Furthermore, one needs an active user ("currentUser") to access the quizzes and just the rest of the website in general. We first read the input from the two DOM-elements (user and password), the password must be at least 4 chars long. We also check if there already is a user with the same username in the localStorage, if there is we check if the input and saved password are the same if they are we let the user in and welcome them back. In the case of the user not existing we create a new account and update the JSON including the past usernames and passwords with the new entry.



One can also change the view of the password-field from secure to plain to check their input. I have downloaded two pngs and switch them in respect to the current view.

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.



Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

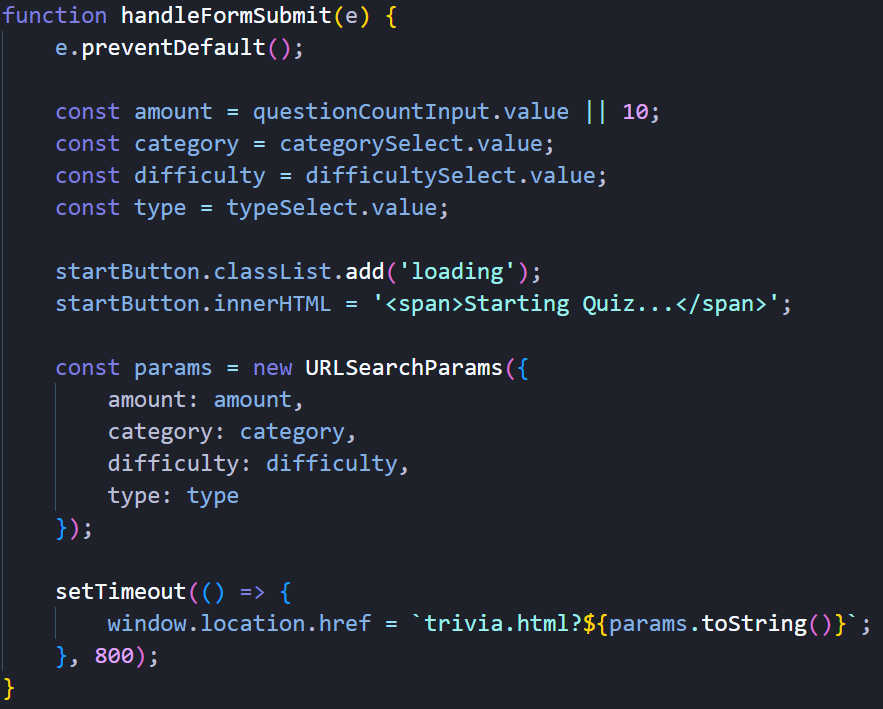
**The Main Page**

On the main page the user can decide how many questions there are, how hard those questions are and what category those questions should cover. This creates a custom URL.

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

After deciding what kind of quiz, they want the play the player clicks on “Start Quiz”, there is a little build in delay, for we want the player to think that we are loading something, after the delay we redirect the user to the trivia HTML side and the quiz is started.



**The Quiz**

Once user has started the quiz, we use the link located in the window.location and send out a request concerning the API-questions.

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Then we fetch and display the questions.

Ein Bild, das Text, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

After completing the quiz, we show the results.

Ein Bild, das Text, Screenshot, Schrift enthält.

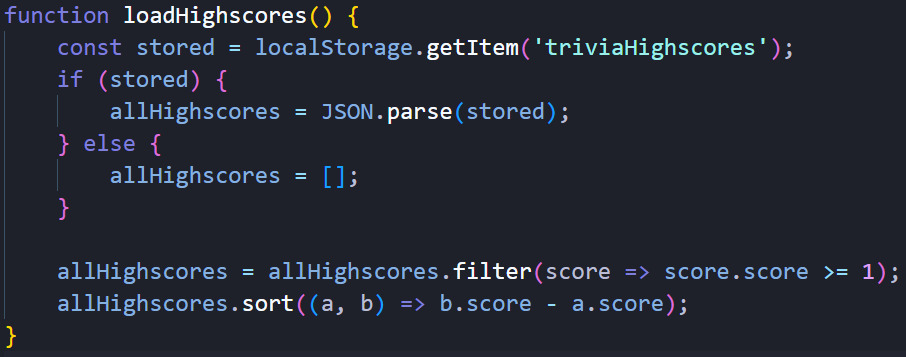
KI-generierte Inhalte können fehlerhaft sein.

**The Leaderboard**

We load the stored scores and sort them accordingly.

The function calculates the difference between b.score and a.score.

* If the result is positive (b.score > a.score), b is placed before a in the sorted array.
* If the result is negative (b.score < a.score), a is placed before b.
* If the result is 0, the order of a and b remains unchanged relative to each other.



We also give the user the option to filter the highscores by category. Here we create a new filteredHighscores array that consists of all the elements in allHighscores, we use the new array to filter the elements by their category.



This is the function we use for the nice “counting-up” animation for the numbers

