The Quiz app is made using HTML, CSS (Bootstrap) and JavaScript. It consists of three views - index.html (the starting page), quiz.html (the quiz page) and end.html (the page where the result is displayed). In addition there are three .js files, one for each view, and a css file with additional styling.

index.html, index.js

The index.html file contains a small form where the user has to provide a username and choose one of four available quiz topics.

The index.js file collects all the needed data from the index.html form using the *getElementById* function. Both form fields have an *addEventListener* function assigned to them, which checks whether the user has interacted with both fields. If yes, the *formValidation* function is started which checks whether both fields have values in them, enables the submit button and changes its color from red to green by removing and assigning a class to it.

The Start button in index.html starts the *savePlayerAndTopic* function in the js file that stores the name and chosen topic with the help of *localStorage.setItem* and redirects the user to the next view.

quiz.html, quiz.js

In quiz.html there are placeholders for the topic, question, buttons for answers and a progress bar.

The quiz.js file get elements from quiz.html using *getElementById* and pulls the chosen topic and the user’s provided name from local storage using *localStorage.getItem*. The topic placeholder in quiz.html if filled with the stored value from the first view. Several other variables are declared as well.

The *questionList* variable is an array which contains all the possible questions each of which have four possible answers, one correct answer and the topic each question falls under. The *filter* function selects all questions that fall under the chosen topic, comparing it to the stored value from index.html, and makes a new variable *questions* with them.

The *CORRECT_BONUS* and *MAX_QUESTIONS* variables determine how many points the user gets for a correct answer and the maximum number of questions. The code can be modified to add more questions, choose a different amount for the maximum number of questions and the way the user gets scored.

The *startGame* function sets the question counter (variable *questionCounter*) and the score (variable *score*) to 0 and declares the *availableQuestions* variable as an array of the *questions* variable. The *getNewQuestion* function is started.

At the beginning of *getNewQuestion* function the program checks whether the maximum number of questions is reached, and if that's true stores the score in local storage and redirects the user to the last view.

If the maximum number of questions is not reached the app proceeds to selecting a new question for the user. A random number between 1 and the number of remaining available questions is chosen with the help of *Math.floor* function and its value is assigned the variable *questionIndex*. With the help of *questionIndex* a question is chosen from the array of available questions and the question is removed from available questions using the *splice* function. With the help of the *foreach* function the values of all possible answers (*choice1*, *choice2*,...) are filled in the buttons in quiz.html.

In quiz.html each button tag contains a data-number attribute and the way the program determines whether the user has answered correctly is by comparing the value of this attribute with the value of *answer* in the *questions* array in quiz.js. If they match the score gets increased by the amount set in the *CORRECT_BONUS* variable, the color of the button is set to green and the progress bar gets filled.

end.html, end.js

In the last view the user is told how many questions out of the maximum number of questions he answered correctly and is given the possibility to redo the quiz by pressing Start over. The values for the name of the user, the correct answers and maximum number of questions are pulled from local storage using *localStorage.getItem*.