CSE 2102: Lab 6 (Final Project)

Tom McCarthy

Code located in my repo: <https://github.com/tommcc314/CSE-2102-Software-Engineering>

Libraries/Frameworks/Tools Used:

* React Router: A React library for handling multiple pages. Some example code I based my router off of is from <https://stackoverflow.com/questions/41956465/how-to-create-multiple-page-app-using-react>
* Jest/testing-library: For unit tests related to rendering and functionality of elements.
* Bootstrap: For portable UI like buttons and the navigation bar.

Description of Quiz System:

There are three pages: home, quiz and signup.

The home page contains a welcome message with a brief description of what the quiz system offers. It has a call to try a quiz.

A screenshot of a quiz test

Description automatically generated

The signup page has an email and password field. These are validated by a similar method to Lab 5, where the email format and password length/complexity are checked.

This page utilizes React state to track the inputted email and passwords, as well as the outcome of validating those fields. When the validation function receives the values and detects errors, it sets the validation states to contain those errors, which display on the page accordingly.

A screenshot of a computer login form

Description automatically generated

The quiz page has 10 questions related to the content of this course (some are traditional multiple choice, some are true/false). When the user clicks the “submit” button, the score appears.

A screenshot of a survey

Description automatically generated

The system keeps track of the score by updating a Boolean array of results. Every time a user clicks an answer choice, the system dynamically updates the corresponding result in the array. When the user submits, the system sums up the number of correct answers in results and displays the score.

A screenshot of a computer

Description automatically generated

Unit Tests:

App.test.js contains React tests. Four of the tests test for proper rendering of components, like the home page and navigation bar. The test for the quiz page goes through every question and tests whether it is displayed on the page.

I also added a functional test for scoring, where the test clicks on the first option for every question, clicks “submit” and expects the proper score to display.

Architecture Diagrams:

A diagram of a network

Description automatically generatedApp Structure:

A screenshot of a test

Description automatically generatedQuiz Close-up:

Signup Close-up:

A screenshot of a document

Description automatically generated