Testing - code completed by Zai

* Overall Fault - jest testing framework was used which meant many of the components rendered from external components in reacting could not be properly run within a test.
* Faults:
  + useNavigation() which was used by react within a <Router> block to navigate between all the pages of the app was not accessible within the testing components
  + I tried to make a mock navigate function to mirror the functionality of navigation in many different ways and all of them were having the same issue that seemed to be caused by a package alias being created by a function called mapModuleNames within jest’s folder within node\_modules. The error would prevent the test suite from running and say that it couldn’t find react-native mapped as react-native web. Seeing as this was a problem with react and the functionality that the jest test framework allowed for it was a difficult issue to address. There are thousands of files within node\_modules and messing something up could cause a lot of issues with dependencies.
* Fix: After hours of troubleshooting, the relatively simple fix of creating a configuration file for jest came to me. I have worked with configuration files in the past but I have not used jest or react so it was not my first idea when trying to fix the testing issues. I made a jest config that created a navigation component to mock the nav component from react.

Original questionnaire quiz.js state issues - addressed by Zai and Ethan

* Original Questionnaire (At the time all the questions were displayed on one screen with a scroll bar).
* Faults
  + Issues keeping track of information relayed from the home page and difficulty keeping it in the proper order. Many of the issues came from the backend and frontend having difficulty communicating (JSON quiz files and JS application)
  + Difficulty in managing complex variables and generating results from variable answers. Originally we planned on having quizzes that could have dynamic variables for answers, but this design proved to be more complex than previously thought when implementing a quiz generator.
* Fix
  + separating each question to its own screen using useEffect() in react would've helped to keep answer information in proper order.
  + Opted to use static questions and answers that have only one correct answer. Using an API to integrate quiz questions proved to be the logical choice when implementing quizzes to have right or wrong answers.

Third code segment to review - Completed by Zai

* React App page/variables rendering/updating
* Faults
  + useState() function was not updating the variables, which caused an error when the user was interacting with the quiz and selecting answers.
  + The app rendered the page only once: at the beginning. This caused problems, since the page would not update properly.
* Fix
  + Each time a component is rendered, useEffect() was used to force the page to render each time an action was performed.
  + Implementing this, forced the pages to update properly, and display the correct information to the page.

Documentation segment to review - completed by Ethan

* Issue: outdated documentation not required from rubric.
* Fix: The majority of the documentation from project 3, regarding the design and descriptions of the project was moved to a new folder. Some folders were tweaked to reflect updates made to the project.