Final Mod Project
(https://perscholas.instructure.com/courses/630/assignments)

Mod-2 Project

–/100 Points

Calculated by: Most Recent

Unlimited Attempts

Attempt 1

Available | Upload | Submit | Not Graded Yet

Details

Module 2 Final Project Evaluation

You must meet all of the following requirements to pass Module 2 Project

Minimum Project Requirements

- Project must be approved by Instructor approval at the start of project week
- Project code is pushed to a public Git repository
- The code should be written in ES6 as much as possible
- Use the create-react-app generator to start your project.
- Follow the instructions on this repo to setup the generator: create-react-app
- Your app should have one HTML page to render your react-redux application
- There should be 2 container components
- There should be 5 stateless components
- There should be 3 or more routes
- The Application must make use of react-router and proper RESTful routing
- Use Redux middleware to respond to and modify state change
- Make use of async actions to send data to and receive data from a server
- Your client-side application should handle the display of data with minimal data manipulation
- Uses React, and Redux
- Uses CSS or styled components to style application
- App renders in browser

- Has readme.md file at the root of repository
- readme file clearly documents the purpose of the project as a problem statement
- readme file clearly documents the target end user for this program.
- readme file has embedded link (not url) to publicly deployed app

Exceeding Project Requirements

- Using a 3rd party api
- Incorporating a new technology not taught in the program

Presentation Requirements

A minimum score of 1 must be met for every item below to pass.

0 - does not meet 1 - meets 2 - exceeds

Problem Statement

- - 0 - Does not state the problem that their program solves
- - 1 - Clearly states the problem that the application solves
- - 2 - Describes the end-user of their application and what problems their application could solve

Technical Problem Solution

- - 0 - Does not state the core solution of their application
- - 1 - Overviews the architecture of their program and how it solves the user's problems
- - 2 - Details the architecture of their program including data structures and algorithms used to solve the problem of the user

Technical Language

- - 0 - Does not demonstrate accurate programming language during the presentation
- - 1 - Uses language that demonstrates general programming concepts during presentation
- - 2 - Uses accurate programming terminology during presentation that demonstrates work ready skills.

Meetings with instructors

An instructor will contact you to setup a meeting time to approve your project.

You will meet with an instructor for 15 minutes to get your game idea approved. Be sure to write out what features you will need to build in order to meet MVP and some stretch goal ideas.

How to Submit Your Project

- Submit your github link.

You will present your project and show your code to classmates and instructors.

Where to go for help during project week

1. Seek out help online
2. Seek out help with your classmates
3. Seek out help with our class TA

TA Hours

1. Will remain the same!

Suggested Ways to Get Started

- **Wireframe** Make a drawing of what your app will look like in all of the stages of the game (what does it look like as soon as you log on to the site? What does it look like while the player is playing? What does it look like when the player wins / loses?).

- **Break the project down into different components** (data, presentation, views, style, DOM manipulation) and brainstorm each component individually.

- **Commit early, commit often.** Don't be afraid to break something because you can always go back in time to a previous version.

- **Consult documentation resources** (MDN, jQuery, etc.) at home to better understand what you'll be getting into.

(https://git.generalassemb.ly/misteroh/SEIR-FLEX-ADA/tree/master/unit_1/w07d3/homework/project_1#think-about)

Think about...

- **Creativity**
  Did you add a personal spin or creative element into your project submission? Did you deliver something of value to the end user?

- **Code Quality**
  Did you follow code style guidance and best practices covered in class, such as spacing, indentation, modularity, and semantic naming? Did you comment your code as your instructors have in class?

- **Problem Solving**
  Are you able to defend why you implemented your solution in a certain way? Can you demonstrate that you thought through alternative implementations?

Attempt 1 Comments

Website Url

http://

More Options