Lab: React

Lab Overview

This Lab consists of 1 React project, which you will create in VS Code.

While performing this lab, incorporate the principles and practices that you've learned from prior lessons in this course. For this Lab, be sure to consider:

- HTML and CSS – The ability to work with and craft user interfaces is necessary to every organization
- JSX – In React, you never really touch HTML proper, you work with this syntax extension
- JavaScript Fundamentals + ES6 – essential skills: variables, scoping, arrays, objects, functions and arrow functions, DOM manipulation and event handlers, the 'this' keyword and callback functions.

It is highly recommended that you **design your solution first**, before starting to code, by creating user stories, a site map or mockup, etc. This doesn't have to be reflective of your final product but will help you think through how to fit all the pieces together.

While not required, it's also recommended that you include this design in your pull request, if you can do so in a small, self-contained way (e.g.: no links to an online site, unless it doesn't require the reviewer to sign in; no files/images/screenshots unless they're only a MB or 2).

Submission Instructions

Create a new feature branch in your TEKbootcamp repository. Commit and push frequently. When your work is done and ready for review, please merge the feature branch into the develop branch.

Following instructions for CodeGrade submission:

1. Click on the "Launch exercise" button on the exercise page at TEKsystems Academy.
2. In the new CodeGrade window, click on the "Connect Git" button.
3. Select GitHub as your Git host. Connect your TEKbootcamp repository to the exercise*.

After selecting a repository, this repository will be cloned to CodeGrade as a first submission. After this, you can start to use Git like you usually would. Every time you push to the develop branch, it will automatically result in a new submission in CodeGrade. Pushes made after the submission deadline will not be taken into consideration.

* You will only have to log in and authorize CodeGrade once; after that, it will be available for all your other assignments inside CodeGrade too.

WARNING: Be sure to connect the proper repository to the exercise. You will not be able to connect to another one immediately using the CodeGrade connection. You can undo your Git connection with CodeGrade by revoking access to the External Tool "CodeGrade" in your GitHub account only.

Grading Expectations

A solution that meets all the requirements in this lab constitutes a grade of Proficient (3/5). To earn a higher grade, your solution must also apply the code quality concepts you've learned up to this point. Additional functionality that does not improve the quality of the code and only serves to make your assignment look more impressive will not help to raise your grade.

Plagiarism Warning

This assignment is a demonstration of your understanding of the topics covered to-date in the boot camp. You may need to reference tutorials or Q&A sites such as StackOverflow to complete parts of the assignment. It is acceptable to use tutorials/Q&As to learn how to add specific functionality to your project, but you cannot simply copy a tutorial repository, follow a single tutorial from start to finish, or copy the code from a Q&A site to develop your application. See guidelines below for using tutorials:

- Do not fork a tutorial repository. This would be considered plagiarism and result in an automatic grade of zero and potential removal from the boot camp.
- Do not simply follow the tutorial from start to finish.
- Do not copy code from a Q&A site and then change it for your application (a few lines of code are permissible)
- You must be able to explain how all functionality included in your project works.
- List ALL tutorials or aids used, either in your README file, and/or as code comments. Include:
    - The name of the tutorial,
    - Which section(s) you used, and
    - The functionality you learned from the tutorial.
- Failure to identify tutorials/Q&A responses will result in significant reduction of your grade.
- If you work with other boot camp members – or instructional team members – to find solutions, that is acceptable, but must also be clearly identified

Estimated Duration

8 to 12 hours

Resources

None

Project: *Social Media Site*

Project Overview

Your exercise is to create your own social media site, or a clone of an already existing site (Ex: Twitter, Facebook, Instagram). The site DOES NOT need to "work". The site should display at least 15 posts (from multiple users) each with their own comments. Users do not need to be able to create or update their profile, or like posts. The site merely needs to *appear* as though it is a working social media site. You will use data from the DummyAPI to mock the users, posts, and comments.

Project Setup Instructions

Within your feature branch for this assignment, use npm to create a react app called 'social-media-site'. As you complete a section of your app commit and push your code to your GitHub repo frequently

Instructions: Complete the following steps using the DummyAPI:

- Create a dashboard view that displays post from multiple users that include the following info:
  - Post body
  - Post owner
  - Comments
  - Like count
  - Post date
- Create functionality to allow users to add comments to posts using hooks (does not need to persist on page refresh) (hint: https://dummyapi.io/docs/comment POST)
- Create a profile view that displays the current user's full profile details such as:
  - Full name
  - Gender
  - Birthdate
  - City, State, and Country
- Use React Router to allow users to navigate between the current user's profile and dashboard
- Make the site responsive so it can be viewed on desktop and mobile devices. If you're not much of a designer, look into CSS libraries such as Material UI, Bootstrap, or Semantic UI.

Exercise Notes:

- Start by creating a site map and a mockup so you don't have to figure out the design while you're coding, sketch it out or use a tool like Figma
- Complete pieces of the app as you follow along in the Udemy course and learn the skills necessary to complete each part of this project
  - Day 1: Create basic components with temporary filler data
  - Day 2: Make the site responsive
  - Day 3: Make API calls to replace temporary filler data with data from DummyAPI
  - Day 4: Create new comment functionality
  - Day 5: Set up routing between pages