SER421: Web-based Applications Spring 2021 Session B
Lab 6

The goal of Lab 6 is to utilize concepts and patterns learned throughout the course to produce an application through the lens of Test-Driven Development (TDD) using React.js.

Important: Be sure to clone the scaffolding code for this lab. Once you have it on your system, in the root directory, run the `npm install` command.

Activity 1 (40 points): Component Library

For this activity you will create a library of React components. Each component will be built to the specifications provided.

Component Library Requirements (each requirement is worth 5 points)

Components are snippets of code that amount to reusable HTML, CSS, and JavaScript. The library must contain the following React components, with the specifications outlined in their respective image files (located in the prototypes folder in the root directory of the lab repository):

- R1. SelectField.
- R2. Button.
- R3. Checkbox.
- R4. Image
- R5. InfoPanel
- R6. UserInputPanel
- R7. KeyValueList
- R8. TESTS MUST BE WRITTEN USING TDD
  - a. Each test must be written using the Jest testing framework and the React Testing Library.
  - b. A test must be written for every behavior, static attribute, and style as described in the requirements outlined in the prototypes for each component. All styles for a particular component can be tested within the same test.

Constraints:

- C1. Each component must use Styled Components, or the built-in React styling.
- C2. Each component must be in its own file and have a corresponding markdown (.md) file in the same directory.
- C3. Each component must have a corresponding test file in the form of <component name>.test.jsx, and each test file must be in a folder named "test" within the root directory.

Activity 2 (35 points): Implement a web application using Next.js

For this activity you will implement a web application that will gather information from an external API and show an image and contextual information on a web page hosted by Next.js.

Background:

There are 2 external APIs to choose from:

- a. The Cat API
- b. The Dog API

You will need to go to one of these websites and obtain an API key. Please see the lab video lecture for more information.

Web Application Requirements (each requirement is worth 5 points):

If it does not already exist, create an index.jsx file in the pages folder in the root directory.

- R1. Create the index page by following the requirements outlined in the corresponding Index.png file located in the prototypes folder in the root directory.
- R2. A user must be able to get a random image by clicking on the "fetch" button.
- R3. A user must be shown details about the animal in the image if provided in the response from the API.
- R4. A user must be able to choose an image based on a selected breed.
- R5. A user must be able to choose an animated .gif file to be shown.
- R6. Choosing a breed or an animated .gif must be mutually exclusive.
- R7. TESTS MUST BE WRITTEN USING TDD
  - a. Each test must be written using the Jest testing framework and the React Testing Library.

b. A test must be written for every behavior, static attribute, and style as described in the requirements outlined in the prototype. All styles can be tested within the same test.
c. A test must be written for the three different API calls: random, by breed, and animated .gif. Each of these tests must mock the API call and trigger the call through simulated user behavior (button click).

Constraints:

C1. The index.jsx file must use Styled Components, or the built-in React styling.
C2. The index.jsx file must have a corresponding index.test.jsx, and each test file must be in a folder named "test" within the root directory.

SUBMISSION INSTRUCTIONS (READ CAREFULLY and ASK QUESTIONS!):

1. Create a zip file named <asurite>_421Lab6.zip where <asurite> is your ASURITE id. No RAR or 7zip files!
2. The zip file should have a root folder with the NodeJS code and related files.
3. In the root folder include a README.md (update or replace the one give if necessary) with any information you want us to know and requested by the lab specification.
4. I strongly suggest, especially on programming problems, that you get a stable solution to a part, save it, and then move on. We do give partial credit. We allow as many submissions as you want to do and only grade the latest!
5. DO NOT INCLUDE THE node_modules OR .next FOLDERS!