COMP 9783 Front-end Development

Lab-3-1 - First React Lab. (4% of the course mark)

Name:
Student Number:

The First React Lab is an immersive hands-on workshop designed to introduce participants to the fundamentals of building dynamic web applications using React.js, a popular JavaScript library for building user interfaces. This lab provides an interactive learning environment where participants can gain practical experience working with React.js through a series of guided exercises and projects.

Lab objectives:

1. Understand the basics of React.js, including its component-based architecture and virtual DOM.
2. Learn how to set up a local development environment with Node.js and npm to build React applications.

Create a react app:

On **VSCode**, open the **terminal** and type the following commands:

1. **npm install -g create-react-app** and **press enter**.
2. **create-react-app --version** and **press enter**. Take a screenshot of this and name it **cra-version.png**.
3. **create-react-app first-react-app** and **press enter**. This will create the files and folders required by React.

Starting a react app:

On **VSCode**, open the **terminal** and type the following commands:

1. Ensure that you are inside the **first-react-app** project directory, if not change to the **first-react-app** directory by typing: cd **first-react-app** and **press enter**.
2. **npm run start** and **press enter**. This should open a browser window and display the generate react app. Take a screenshot of this and name it: **react-app.png**.
3. Edit the file **src/App.js** and look for:

```
<p>
  Edit <code>src/App.js</code> and save to reload.
</p>
```

4. And add the HTML code below:

```
<p>YourFirstname YourLastname's first react app.</p>
```

5. Update **YourFirstname YourLastname** to your **first** and **last names**.
6. Save the changes, the changes should reflect immediately on the browser, take a screenshot of this and name it **react-app-modified.png**.
7. Press **Ctrl-C** to stop the app.

Testing a react app:

On **VSCode**, open the **terminal** and type the following commands:

1. Ensure that you are inside the **first-react-app** project directory, if not change to the **first-react-app** directory by typing: cd **first-react-app** and **press enter**.
2. **npm run test** and **press enter**. This should run the test and display the results on the terminal. Take a screenshot of this and name it: **test-react-app.png**.
3. Edit the file **src/App.test.js** and at the end of the file add the following:

```
test("check custom text", () => {
  render(<App />);
  const linkElement = screen.getByText(/YourFirstname YourLastname's first react app../i);
  expect(linkElement).toBeInTheDocument();
});
```

4. Make sure to change **YourFirstname YourLastname** to the names used on **Starting react app step 5**.
5. Save the changes, this should run the test and display the results on the terminal. Take a screenshot of this and name it: **test-modified-react-app.png**.
6. Press **Ctrl-C** to stop the app testing.

Building a react app:

On **VSCode**, open the **terminal** and type the following commands:

1. Ensure that you are inside the **first-react-app** project directory, if not change to the **first-react-app** directory by typing: cd **first-react-app** and **press enter**.
2. **npm run build** and **press enter**. This should build an optimized version of the app.
3. **npm install -g serve** and **press enter**. This should install the **serve** app, which is an http server for the files from built step 2.
4. **serve -s build** and **press enter**. Links to the react app are displayed on the terminal, these links can be opened by **clicking** and pressing the **ctrl key** at the same time.
5. Take a screenshot of the results on the **browser** and **terminal**, name them **react-build-browser.png** and **react-build-terminal.png**.

Submission:

1. Use the html template: **index.html** and **write HTML codes for each screenshot**:
   a. Write a title and short description.
   b. Display the screenshot.

Note: Feel free to use any component from a CSS framework of your choice.

2. Create a new folder named **html** and copy all the **HTML**, **CSS** and **PNG** files used in the previous step.
3. Create a **zip file** of the **html folder**.
4. Submit the **zip file** to **GBC - D2L**.