**Simple React Exercise**

*(Last updated on Jan. 16, 2024)*

**Objectives**

* Learn how to use React to implement two simple frontend form examples that are needed to start the Spring Tutorial in class. (i.e., Consider this as your entry ticket to Spring Tutorial.)

**Tasks**

1. Complete the following two parts of his exercise;

* Part I is a React walkthrough. You are required to follow the given tasks with all required reading and copy/paste the code as instructed.
* For part II, you must create a new page with a new form and add it to a separate route location.
* You will need the React code from this tutorial for Assignment 2 (Spring Tutorial). So, make sure to save it for your future use.

1. At the end of each part, take a screenshot of your homepage and the created page (either Student or Course).
2. Submit all screenshots in a file named **YourLastName\_A1.docx** or **.pdf** to Canvas (here for Assignment 1).

**Part I**

React is a JavaScript library for creating user interfaces. It is a component-based library that focuses on building reusable, encapsulated components to make UIs.

**Section 1 Create a React App**

If you don’t have npm installed, please download nodejs (<https://nodejs.org/en/download/>) and install it. This will install npm and node js.

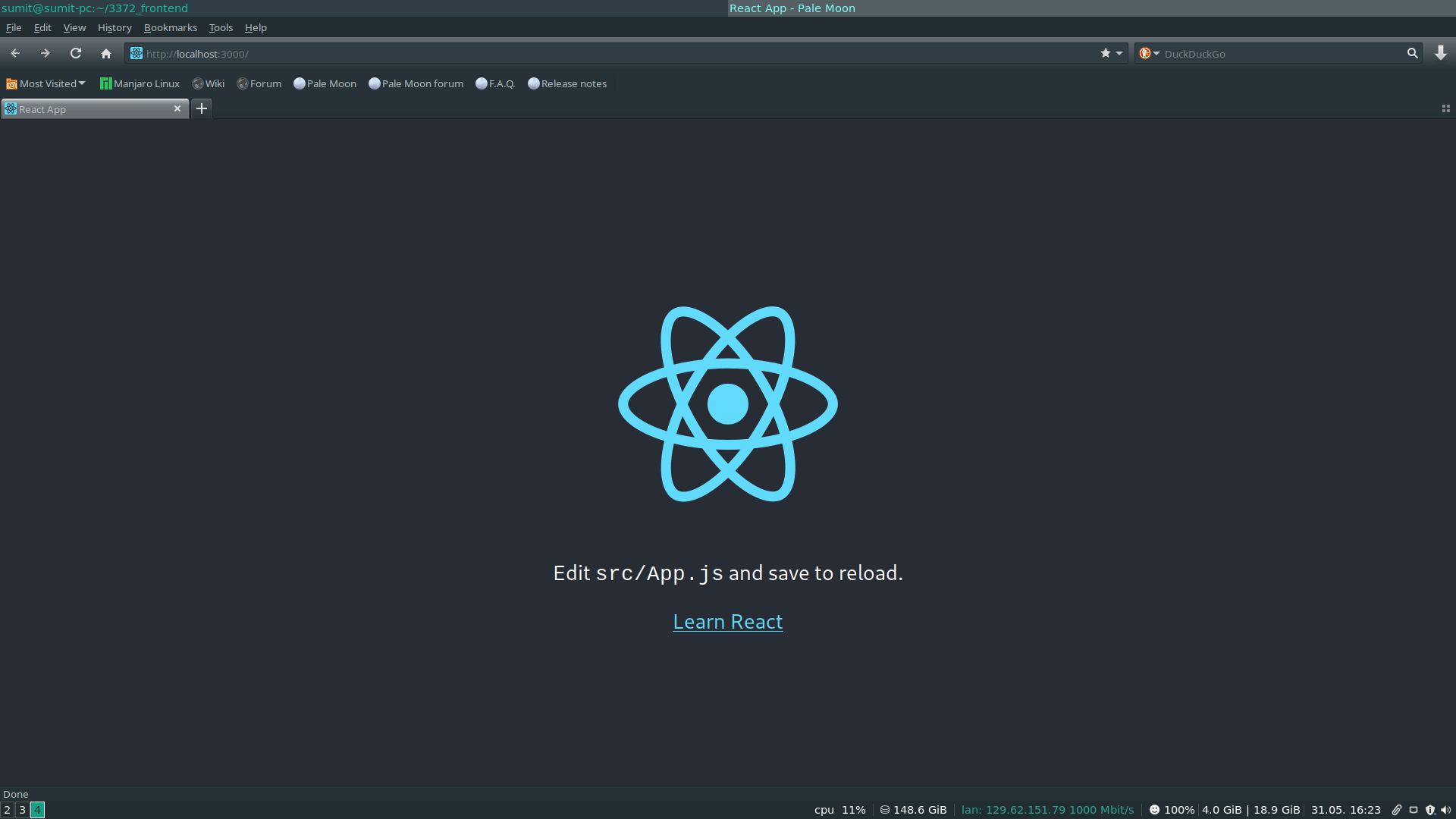
Run the following commands to create a basic React application and start it.

You can run the following commands in the terminal. Just open the terminal or command prompt (cmd) and go to the desired directory using “cd path\_to\_your\_directory”. Then run the following commands. You can also use the console of your IDE (Intellij or Visual Studio Code or any other) to the run the following commands.

* npm install npx
* npx create-react-app frontend
* cd frontend
* npm start

At this point the project has been initialized. Open the project you just created using your favorite IDE (Intellij or Visual Studio Code ) to see the files that have been created.

Open your browser and go to localhost:3000 and if you have the following screenshot then the react app has run successfully.



If you get this screenshot move to Section 2.

**Note**: Due to a bug in older versions of npx, you might run into issues on Windows machines. Either update npx, temporarily install npx, or temporarily install create-react-app. In order to temporarily install npx:

* Run ‘ npm install npx ‘ .
* Run ‘ .\node\_modules\.bin\npx create-react-app frontend ‘.
* After everything has been set-up, you can delete the package-lock.json file and everything in the node\_modules directory.
* Run ‘cd frontend’.
* Run ‘npm start’. You should get the UI same as above screenshot at localhost:3000. (Note: when you make changes to your code while the application is running, the application detects this change and it is reflected automatically. You don’t have to close and run ‘npm start’ every time)

Also, please refer [here](https://reactjs.org/docs/getting-started.html) for some additional references. Now, we have initialized our project. From here, you can make changes in the source code or can download the source code provided.

**Section 2 React app overview**

**React Components**

React Components break the entire UI system into reusable pieces. React lets you define components as classes and functions. In this exercise, we will be using a function component.

Please do the following:

1. Download the source code provided [[HERE](https://baylor.instructure.com/courses/200752/files/folder/Assignments/A1-React%20Tutorial?preview=14884748)].
2. Unzip and open the project in an IDE. (Preferably Visual Studio Code)
3. Run a terminal inside the IDE.
4. Run *npm install*.This will install all the dependencies required. Do not worry about knowing the details for now. FYI, the package.json contains all the necessary dependencies. Once you run the above statement, you should see the creation of *node\_modules* and *package-lock.json*
5. Run npm start.
6. Load <http://localhost:3000/> in your browser.
7. You should see a screen that has two links: *Student* and *Course*.

**Section 3 App.js**

The App.js is basically our main component that renders all the components needed for our project.

If you see, there are components called Home and Student inside the components folder under src.

**Home**: This is the homepage that has the component Student and Course in it.

import React, { Component } from "react";

export default function Home(props) {

return (

<div>

<div>

{" "}

<a href="/student">Student</a>

</div>

<div>

{" "}

<a href="/course">Course</a>

</div>

</div>

);

}

The href takes you to the destination.

Student:

This component contains a form that where the user is required to input name.

import React, { useState } from "react";

export default function Student(props) {

const [name, setName] = useState();

function onSubmit(event) {

event.preventDefault();

window.alert("Welcome " + name);

}

function handleNameChange(event) {

setName(event.target.value);

}

return (

<div>

<h3> Student</h3>

<form onSubmit={onSubmit}>

<label for="name" required>

Name

</label>

<input type="text" name="name" onChange={handleNameChange} />

<button variant="contained" type="submit">

{" "}

Save{" "}

</button>

</form>

</div>

);

}

Here, you see the use of useState. This is called hooks. We will be making use of these hooks a lot in the future. Do not investigate this for now. This will be dealt in future. You can refer to <https://reactjs.org/docs/hooks-reference.html#usestate> for references.

**Section** **4. Implementing Routes**

Routes are mechanisms to provide access to different parts of an application. There are multiple ways to route through your react application. We will use the React router for managing routes.

If you click on Student, you will see the student form. How does this form render when clicking on the link? This is where Routes come into play.

If you see, the App.js has code that look like this:

<BrowserRouter>

<Routes>

<Route path="/" element={<Home />}></Route>

<Route path="/student" element={<Student />}></Route>

{/\* <Route path="/course" element={<Course />}></Route> \*/}

</Routes>

</BrowserRouter>

This is a part of react-router-dom, which is responsible for routing our pages. Find the documentation [here](https://www.npmjs.com/package/react-router-dom).

**Task: Capture the screenshot showing your homepage and Student page.**

**Part II**

Please make sure to complete part I before moving to part II.

1. Create a separate form for ‘Course’ similar to the Student form above.
2. Add some fields, such as course number, grade mode, course name, Professor, etc.
3. Manage the route for this ‘Course’ form so that it can be accessible at <http://localhost:3000/course>.
4. Add a link to the course page in your home page (in App.js)

**Task: Capture the screenshot showing your homepage and Course page.**

**Submission:**

Submit all screenshots in a file named **YourLastName\_A1.docx** or **.pdf** to Canvas through the Assignment 1 (A1) submission link.