CS 513 Software Systems(ESD)

Lab 5 - Frontend Development

Lab Objective:

In this lab session, we will discuss HTML, CSS , Javascript and React framework for the frontend development. As you all know about html, css and javascript, we will try to focus on React framework and try to implement basic functionalities of that.

ReactJS Framework

React is an open-source JavaScript library which is useful in developing user interfaces specifically for applications with a single page. It is helpful in building complex and reusable user interface(UI) components of mobile and web applications as it follows the component-based approach. React is an efficient, flexible, and open-source JavaScript framework library that allows developers to create simple, fast, and scalable web applications.

Important features of React are:

- It supports server side rendering.
- It uses Virtual DOM rather than Real Dom(Data object Model).
- It uses component based architecture for developing applications. Components are independent and reusable bits of code.

Prerequisite

- Environment setup lab handson
- Database setup(MySQL) - Lab 2
- Hibernate and JPA setup - Lab 3
- Rest Services lab- Lab 4

Lab Activity

To create a React application, first of all install Nodejs.

Node.js provides a runtime environment to execute JavaScript code from outside a browser. NPM, the Node package manager, is used for managing and sharing the packages for either React or Angular.

Install NodeJS in Ubuntu:

```
$ sudo apt update
$ sudo apt install nodejs
```

Check the installation of NodeJS:

```
$ node -v
```

You must get the appropriate nodeJS version as an output.

Install the NPM package:

```
$ sudo apt install npm
```

The next step is to install a tool called create-react-app using NPM. This tool is used to create react applications easily from our system. You can install this at the system level or temporarily at a folder level. We will install it globally by using the following command.

```
$ npm install -g create-react-app
```

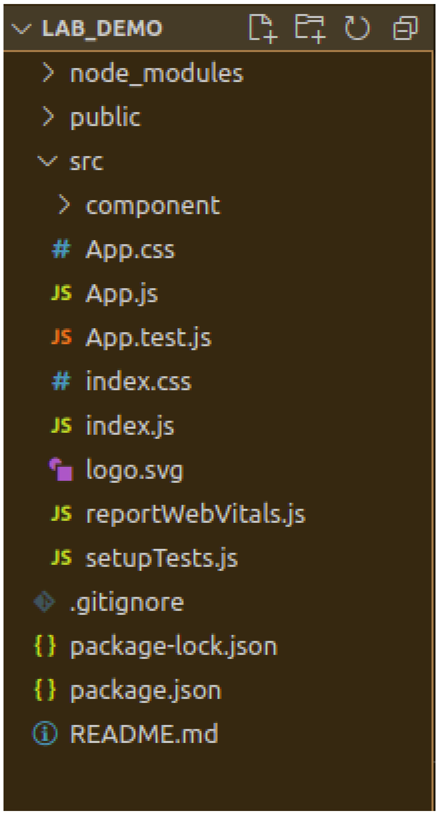
After the create-react-app is installed, we can create our first react application. create a new Project now using the following command:

```
$ create-react-app lab_demo
```

Note: if you get the version related error while running the command, then try the following commands and rerun the above command:

```
$ npm cache clean -f
$ sudo npm install -g n
$ sudo n latest
```

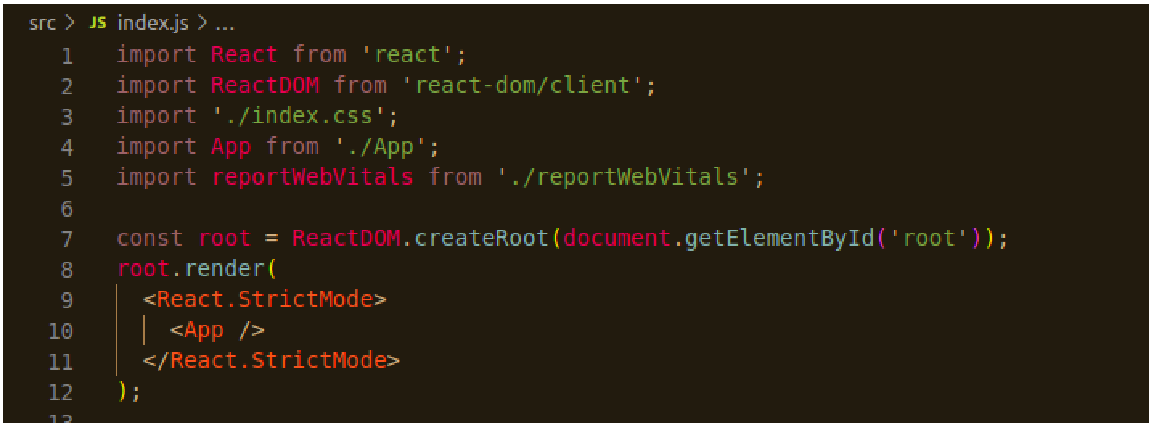
Now open lab_demo in VS Code, you can see the folder structure as shown below.



```
LAB_DEMO
  node_modules
  public
  src
    component
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    reportWebVitals.js
    setupTests.js
  .gitignore
  package-lock.json
  package.json
  README.md
```

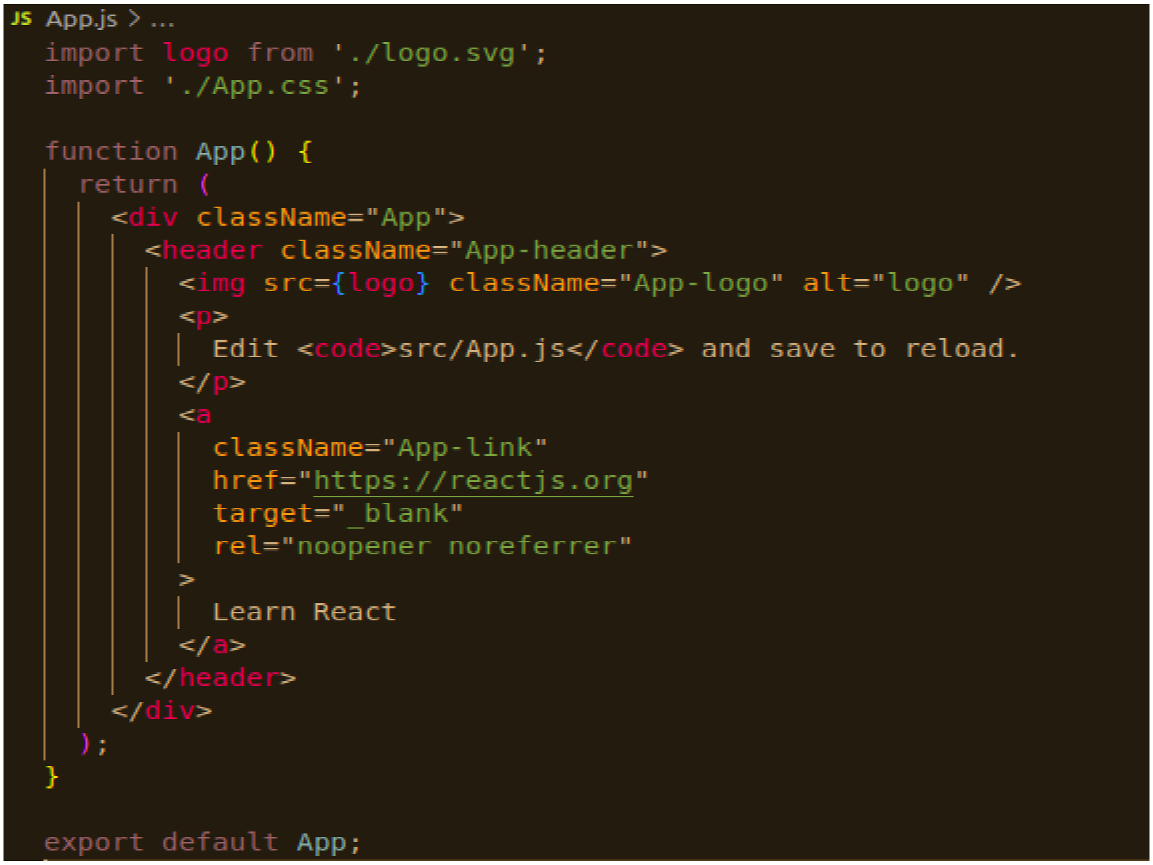
node_modules	External modules that our project depends upon
public	Contains static files such as index.html, library files,public images etc
src	Heart of react application.Main component that contains entry point(App.js) and all the components.
package.json	Contains dependencies and scripts required for the project.

Index.js under the src folder is the main entry point of the React application. In this file, the main DOM is initiated with the name ‘root’ and renders the first module(i.e. component). Following is the demo index.js file that calls the App component.



```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

Following is the demo App.js that generates the main page which will be shown on the browser when the application starts.

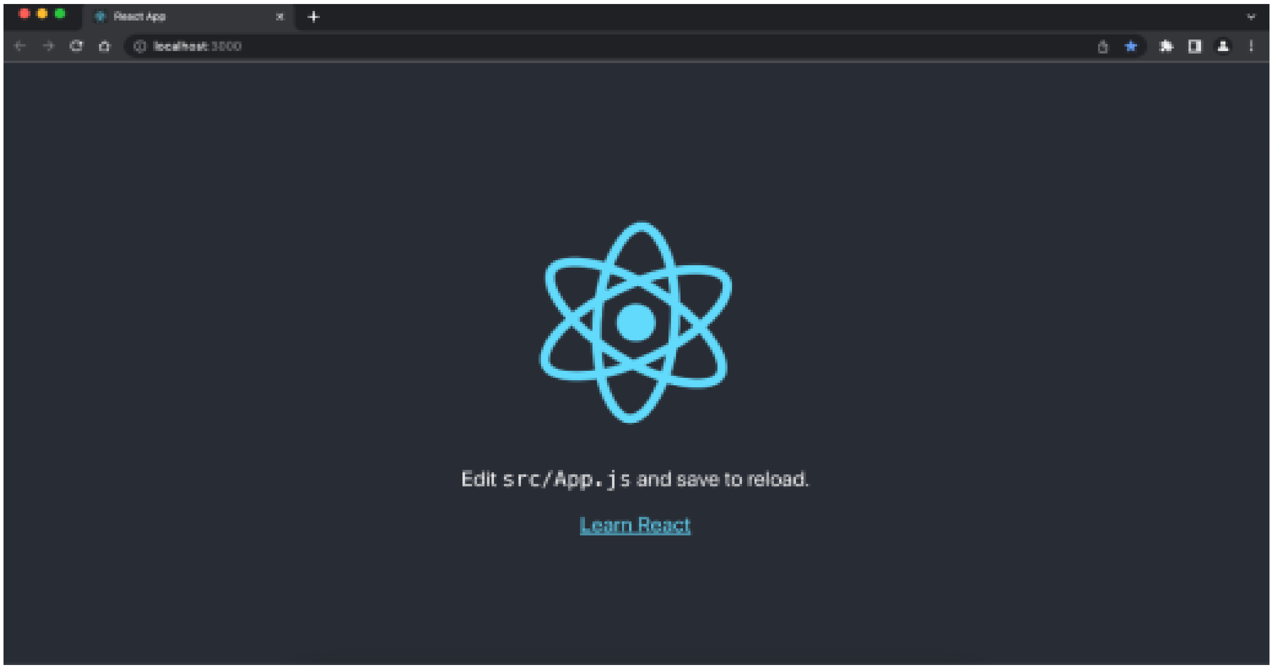


```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

export default App;
```

Go to Terminal->New Terminal, switch to the project directory and run the following command:

```
$ npm start
```

It will start our react app on port localhost:3000



Edit `src/App.js` and save to reload.

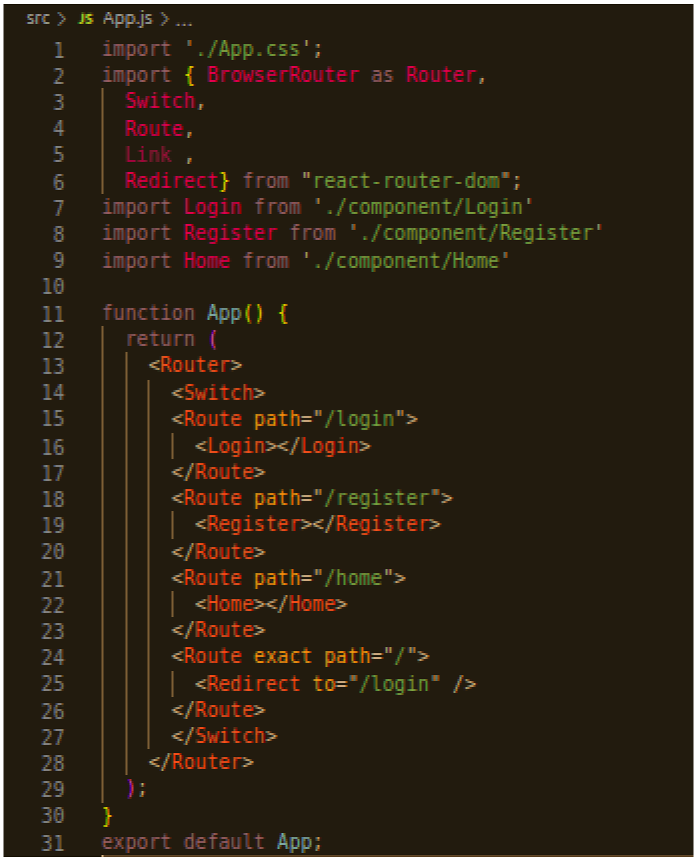
This is the output of create react app. Now you can edit App.js and create your own application.

Demo Project(Only for reference):

GitHub Repository: https://github.com/sarika476/lab_demo

Following is the demo App.js file that defines the multiple URL path and specifies the component which will be called for each URL path. Based on the code:

- localhost:3000/login -> will call Login component
- localhost:3000/register -> will call Register component
- localhost:3000/home -> will call Home component
- localhost:3000/ -> will redirect to the /login path

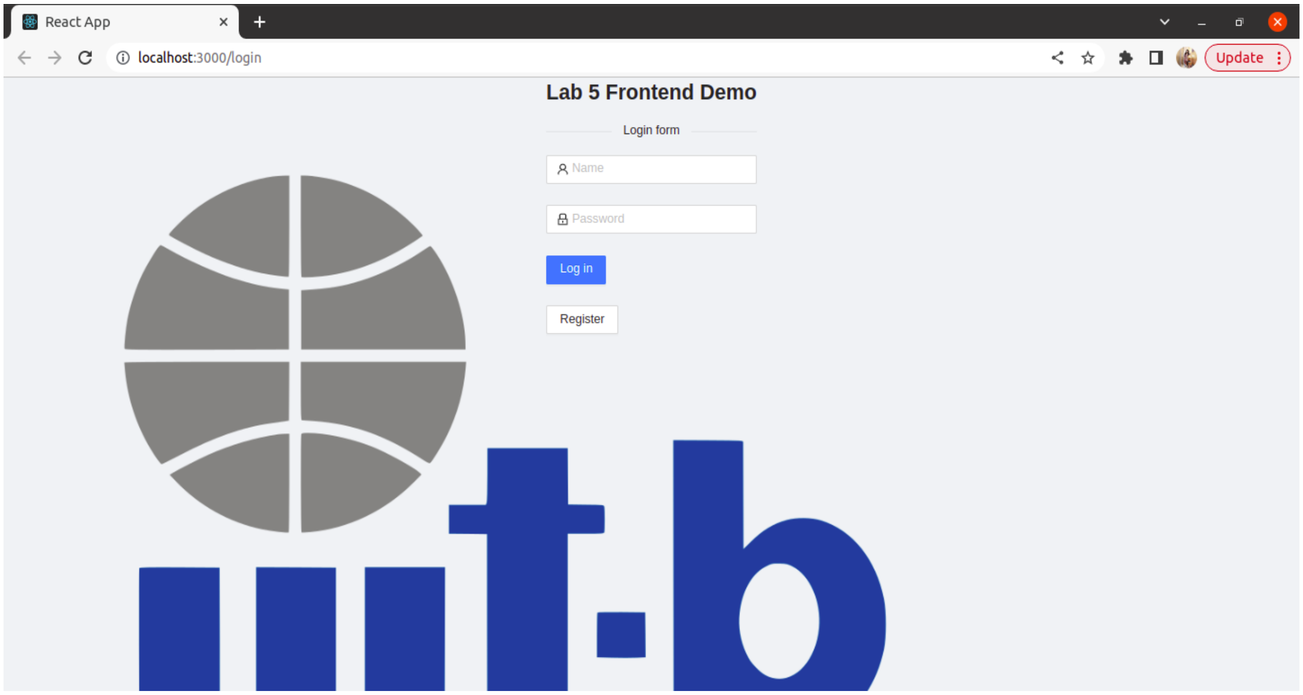


```
import './App.css';
import { BrowserRouter as Router,
  Switch,
  Route,
  Link ,
  Redirect} from "react-router-dom";
import Login from './component/Login'
import Register from './component/Register'
import Home from './component/Home'

function App() {
  return (
    <Router>
      <Switch>
        <Route path="/login">
          <Login></Login>
        </Route>
        <Route path="/register">
          <Register></Register>
        </Route>
        <Route path="/home">
          <Home></Home>
        </Route>
        <Route exact path="/">
          <Redirect to="/login" />
        </Route>
      </Switch>
    </Router>
  );
}
export default App;
```

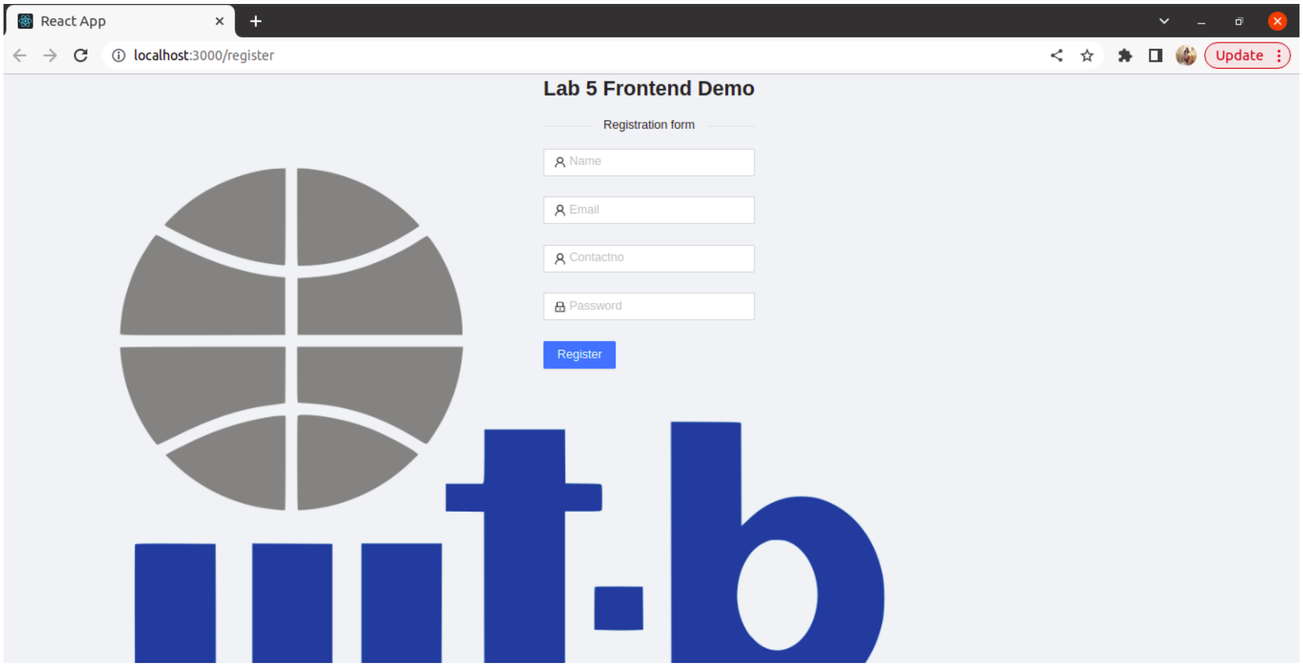
src/App.js

I have created three components for your reference namely Login, Register and Home. Login renders the login page for the application, Register component renders the register page for the application and

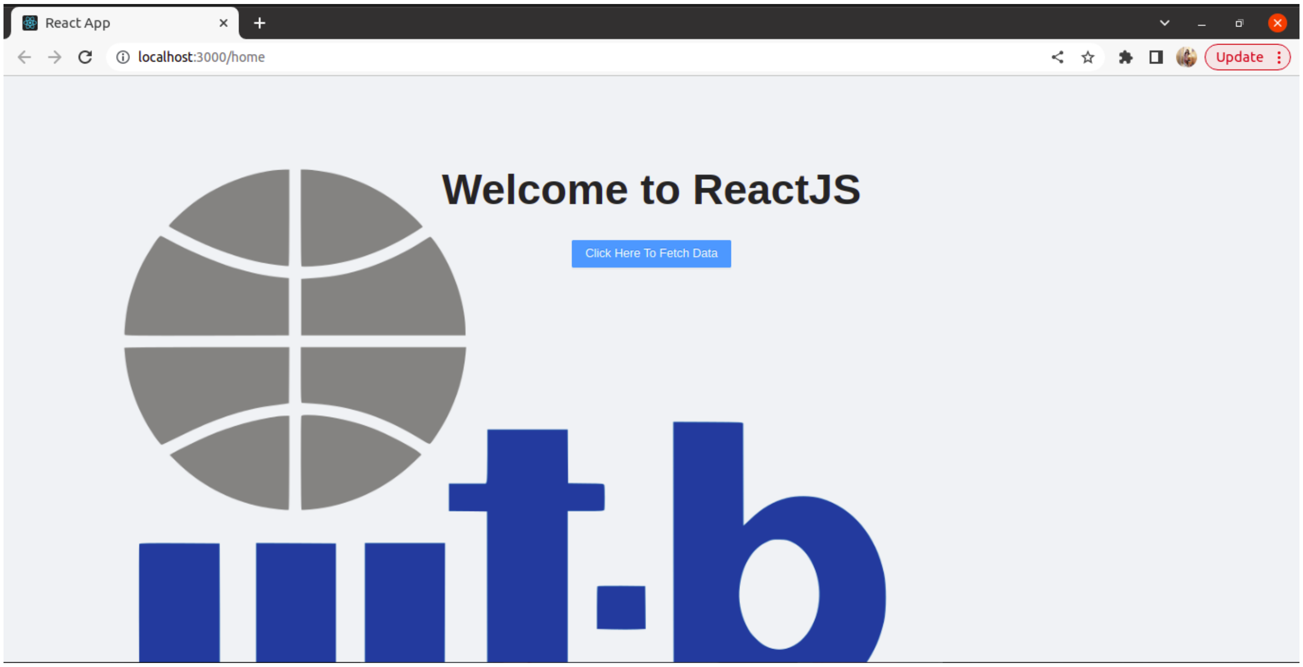
Home component renders the main page after successful login. Following are some output screenshots for each component; you can check the files here.



Login screen



Registration screen



Home Screen

Links:

- https://reactjs.org/docs/getting-started.html
- https://ant.design/docs/react/introduce
- https://www.w3schools.com/js/
- Some other front end development frameworks: https://technostacks.com/blog/best-frontend-frameworks/