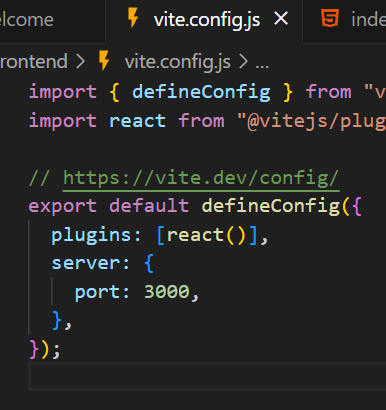


Command to create react app => **npm create vite@latest ems-frontend**

App start run on <http://localhost:5173/>, we can change the port in vite.config.js file –

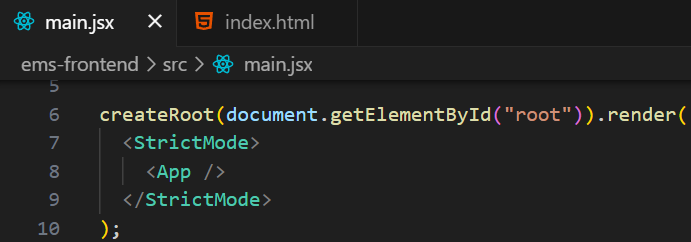
 Command to start App => npm run dev

Node\_modules – whenever we run npm install then npm will create this folder and keep all the downloaded JS libraries and packages inside it.

Public folder – we keep all images and static files like images.

Index.css – styles for index.html app.css – styles applied globally for all components

Main.jsx – main entry point for Js related code. Whenever we run our application, then index.html will get served in browser and it internally call main.jsx file –

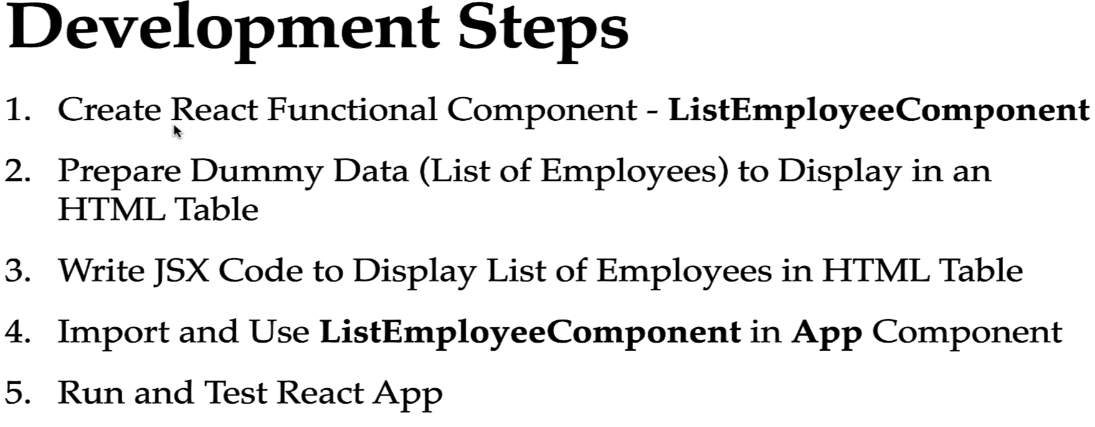
 getting the div with id = “root” and render App component within it.

App.jsx – this file contains code for App component (root component)

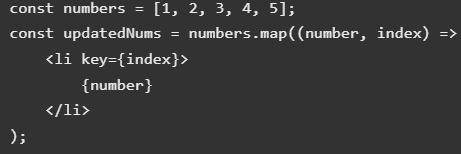
Adding bootstrap in project => npm install bootstrap –- save

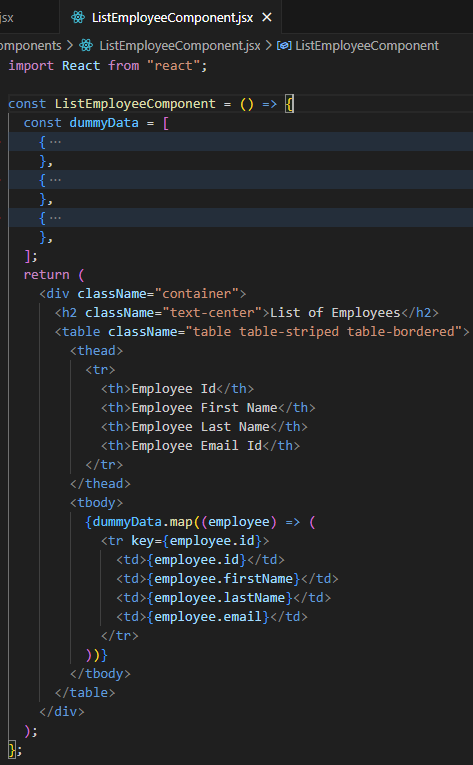
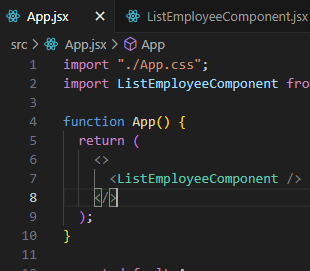
import "bootstrap/dist/css/bootstrap.min.css" in main.jsx file

**Create React ListEmployeeComponent and Display Data**

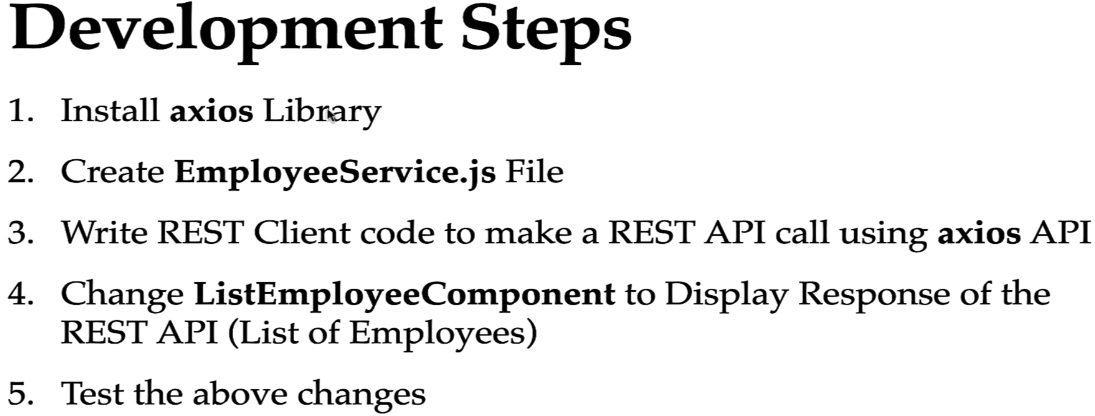
****

Keys in React are special attributes used to identify the items in a list.



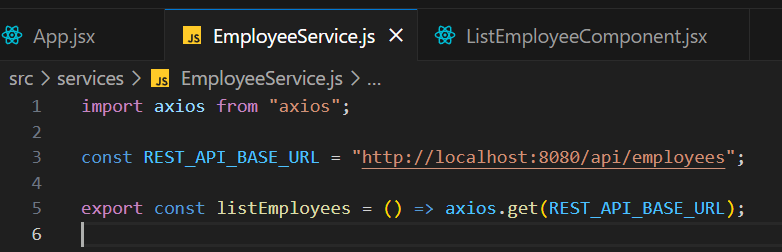
 

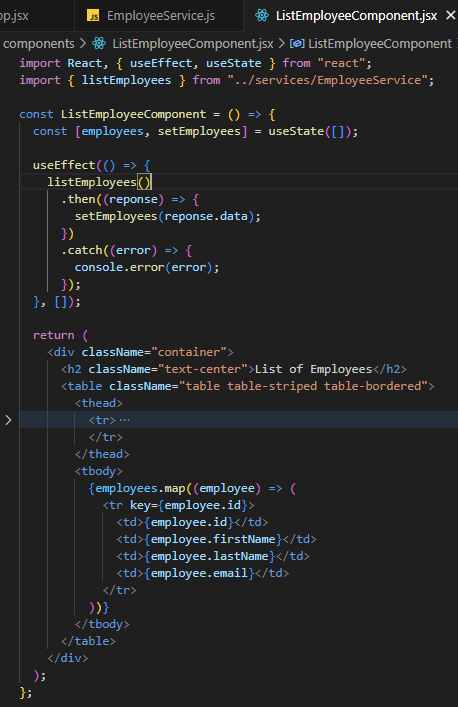
**Connect React App with Get All Employees REST API**

****

Inorder to hold the response of REST Api we have to use State variable. In functional component we use useState hook to define state variables.

Inorder to make REST Api call in functional component we use useEffect hook. It takes 2 arguments – callback function and dependencies list. The useEffect hook is triggered automatically **after the component is rendered**.

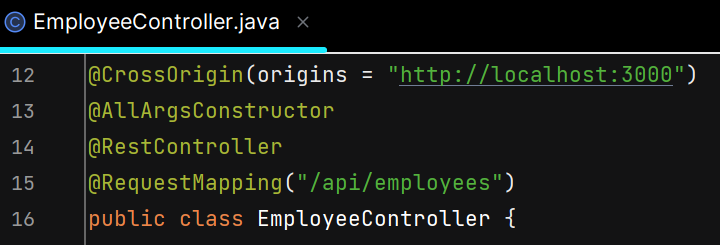




On running app, we get error – “Access to XMLHttpRequest at 'http://localhost:8080/api/employees' from origin 'http://localhost:3000' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.”

This is due to Cross-Origin Resource sharing policy which is enforcesd by browser for security. This happens when your React application (running on http://localhost:3000) tries to make a request to your Spring Boot backend (http://localhost:8080), which are considered different origins.

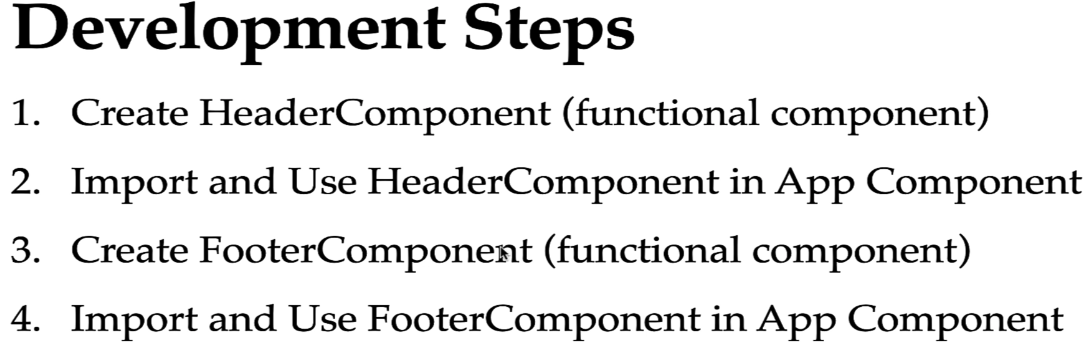
We can handle such CORS issue at springboot backend –

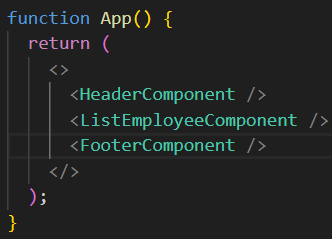
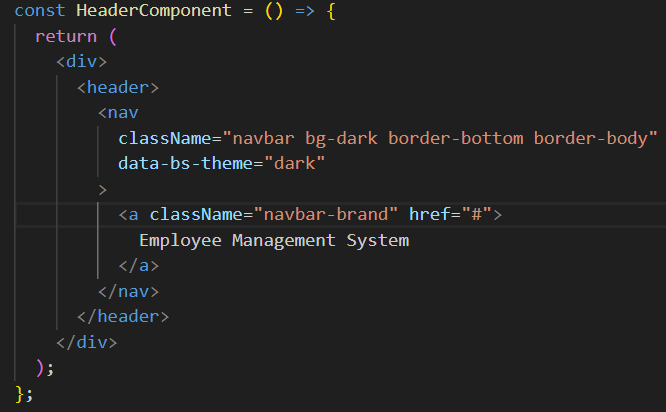


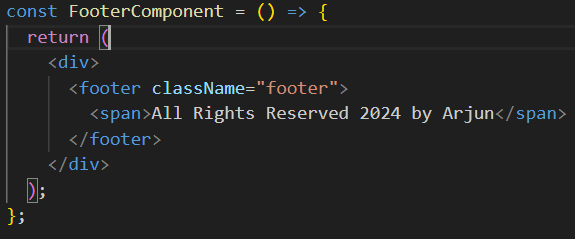
“\*” means all the clients/origin can call employee controller REST APIs

We can also specify origins like - @CrossOrigin(origins = "http://localhost:3000")

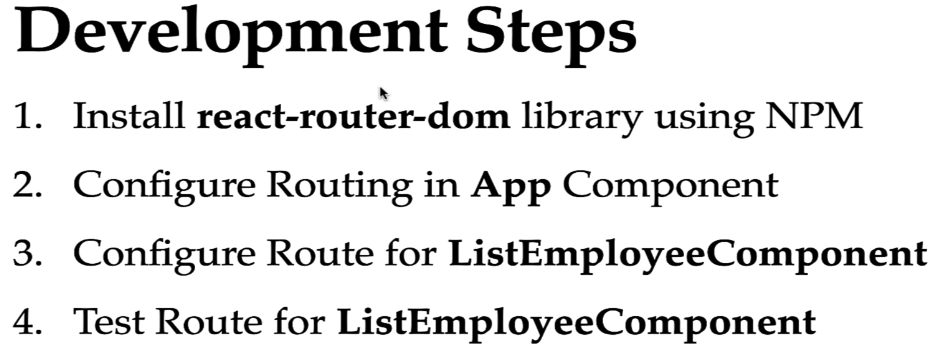
**Adding Header and Footer to React App**







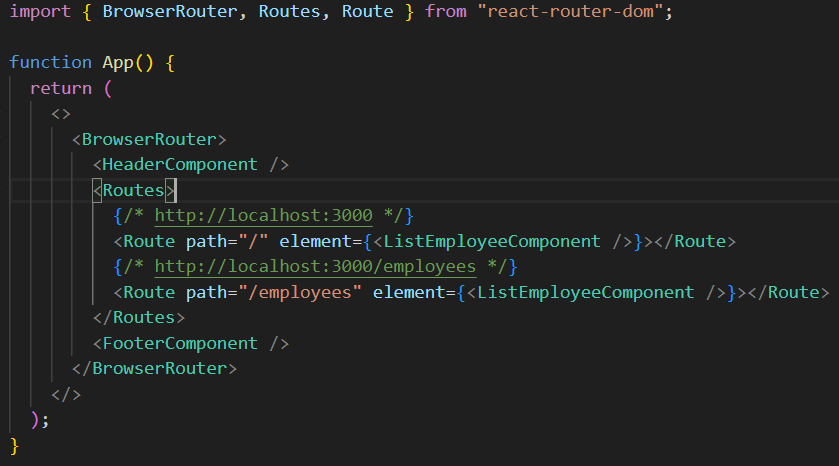
**Configure Routing in a React App**

****

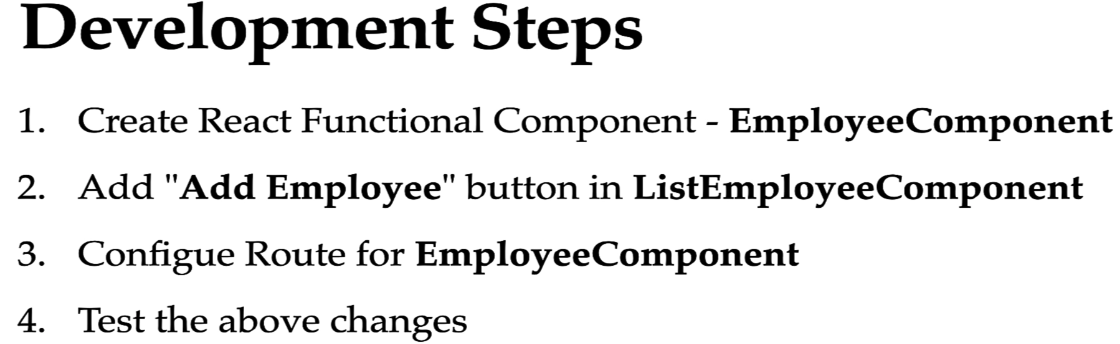
Inorder to configure routing, we have to use routing related components from react-router-dom

Steps – 1) enclose all the components within BrowserRouter component.

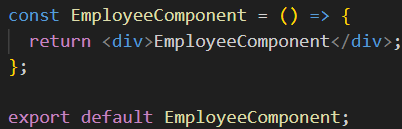
2) Routes is basically a container or a parent for all the individual routes. Within this Routes, we can define the individual Routes –



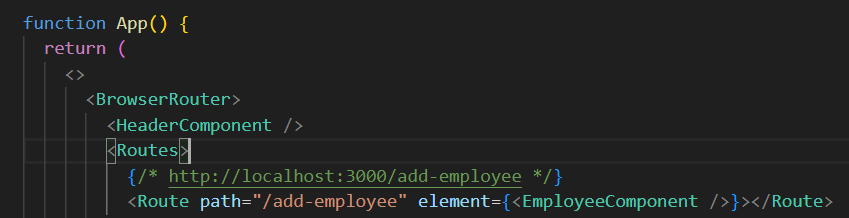
**Create EmployeeComponent and Configure Route**

****

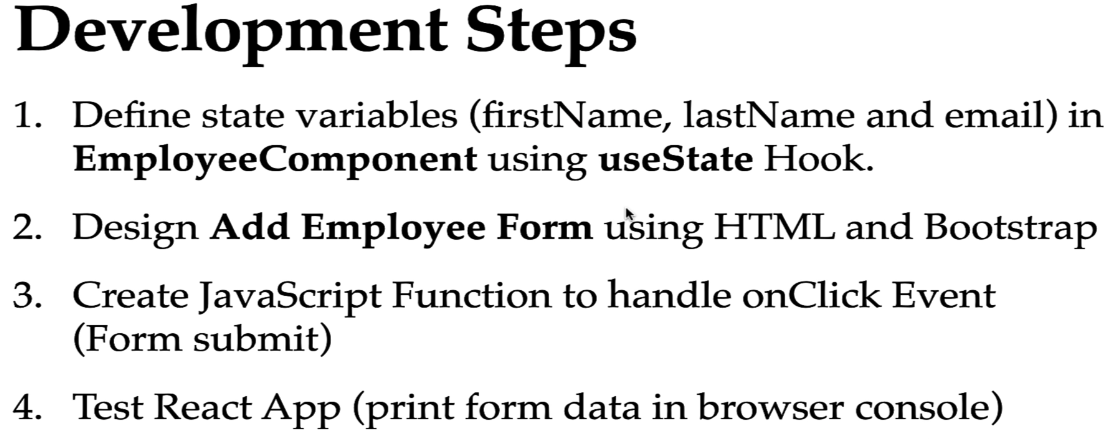
In order to navigate user form one page to another, we can use useNavigate() hook.



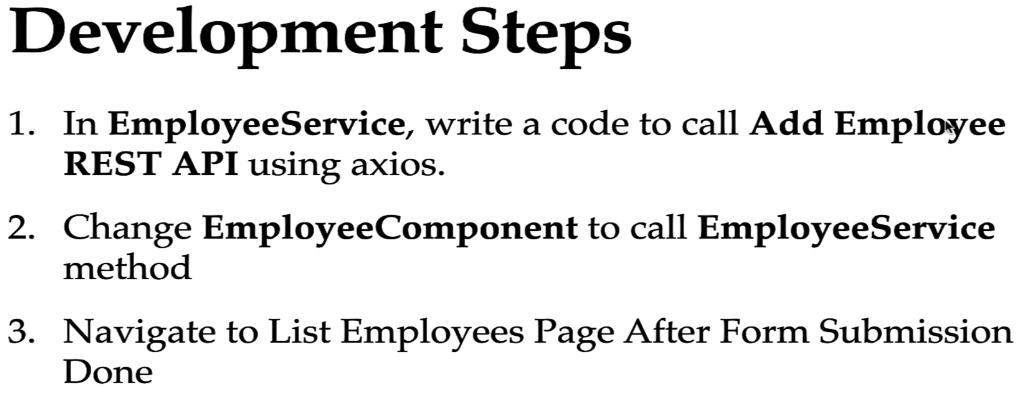




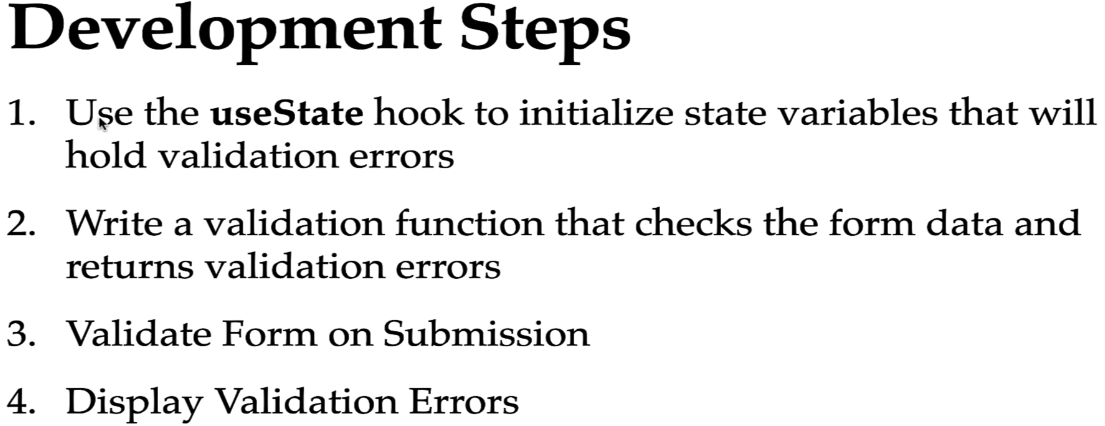
**Add employee Form Handling**

****

**Connect React App to Add Employee REST API**

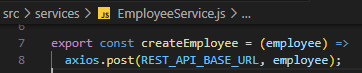
****

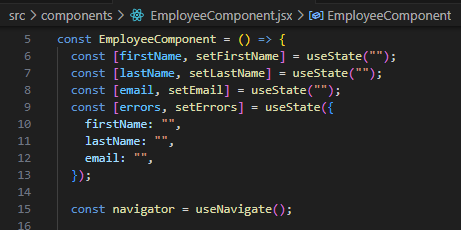
**Add Employee Form Validations**

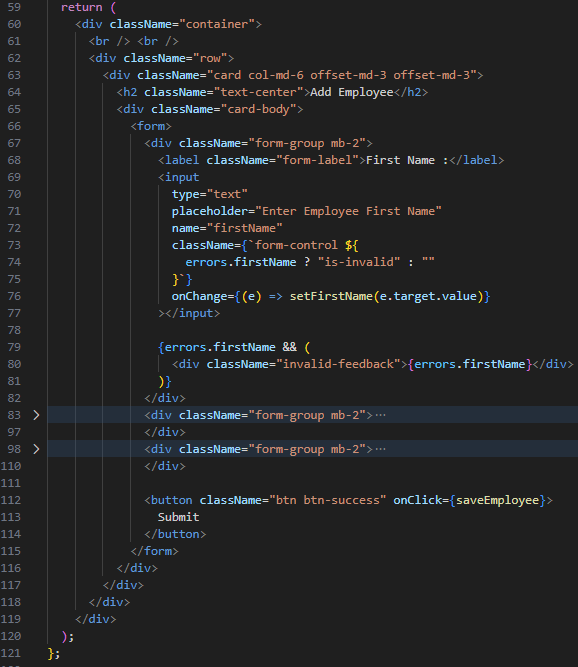
****

The useState() hook is commonly used for managing form input state in React. It allows you to declare a state variable and a corresponding setter function to update the value of the state variable. By using useState(), you can keep track of the form input values and update them when the user interacts with the form.

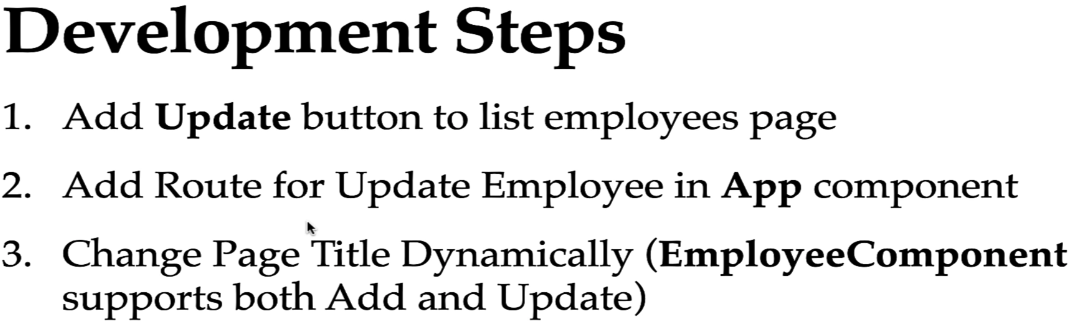
We have to dynamically add the css on input tag.

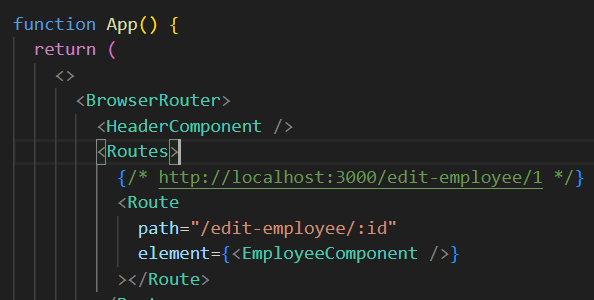






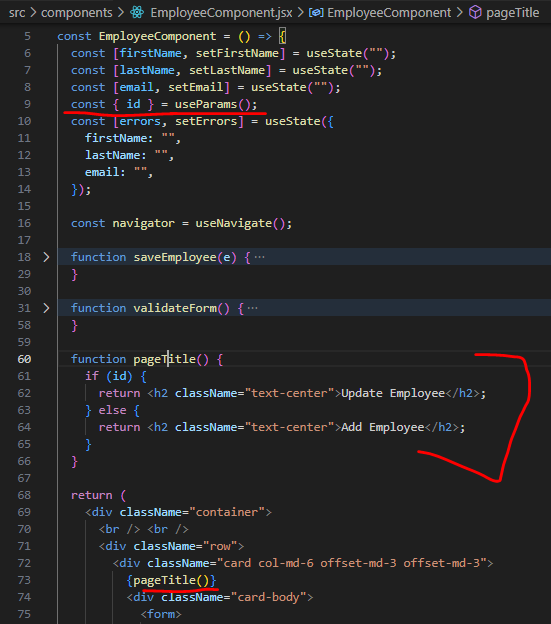
**Update Employee Feature – Adding Update button, Title and Route**

****

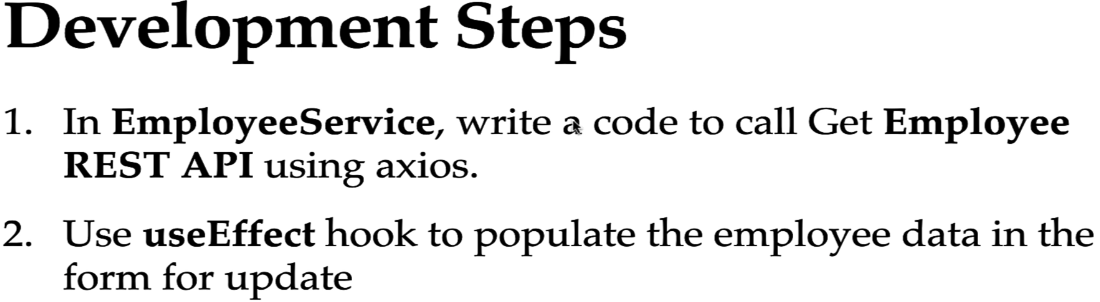


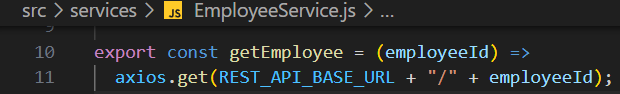
**useParams()** hook from react-router-dom library is used to get the query parameters from url. This returns object with key value pair.





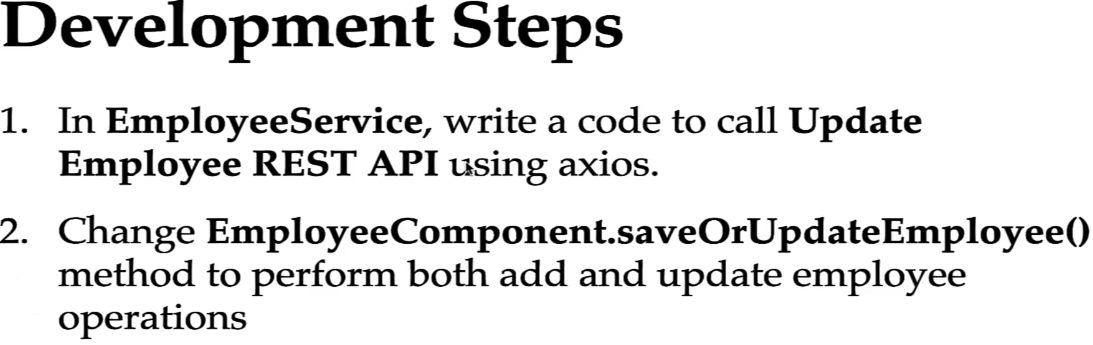
**Connect React app to Get Employee REST API**

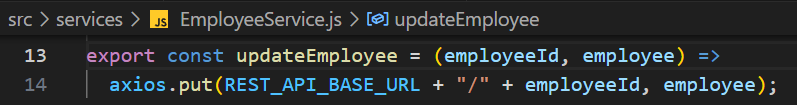
****

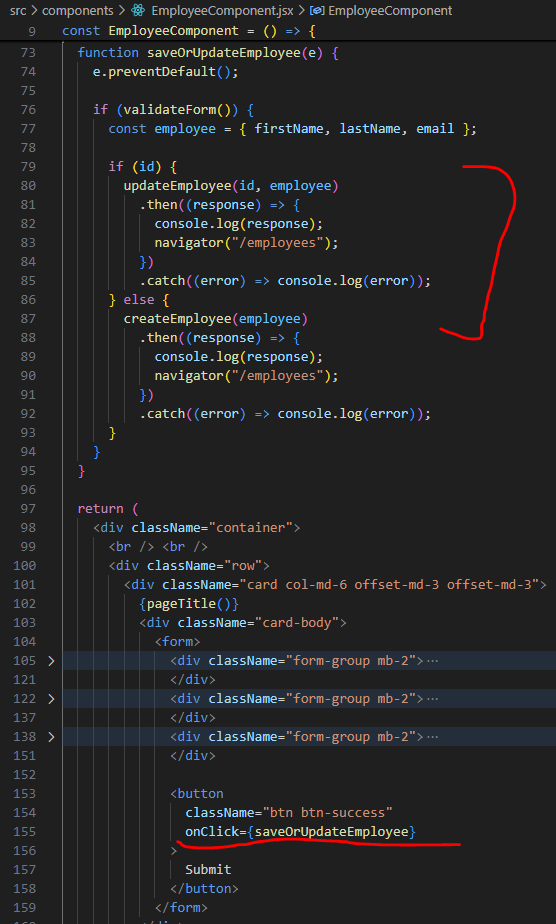




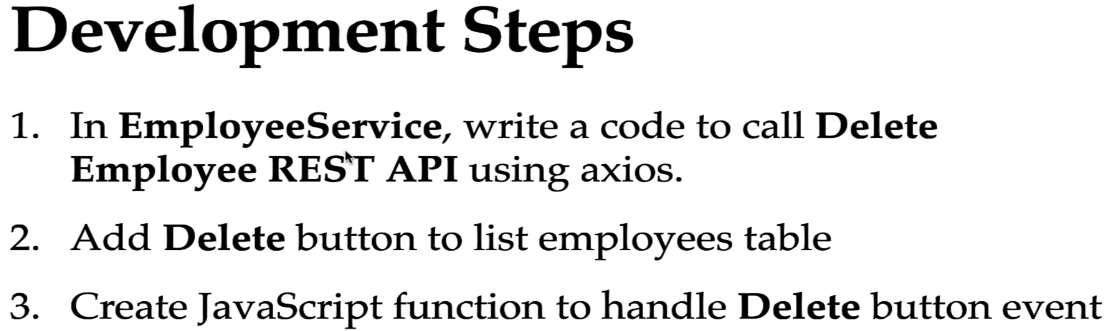
**Connect React App to Update Employee REST API**

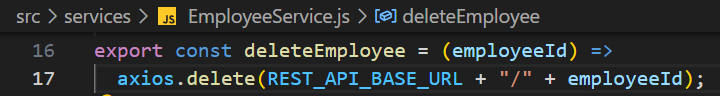
****



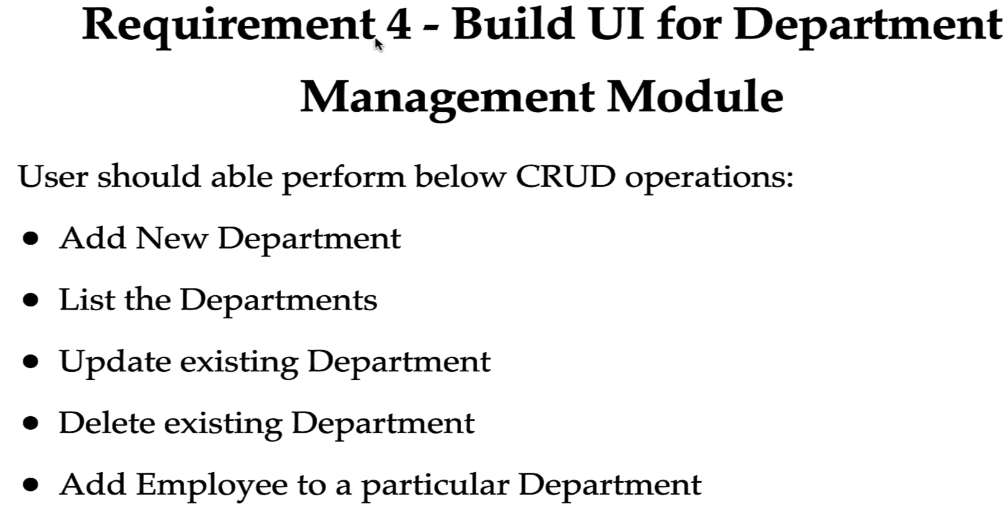


**Delete Employee feature**

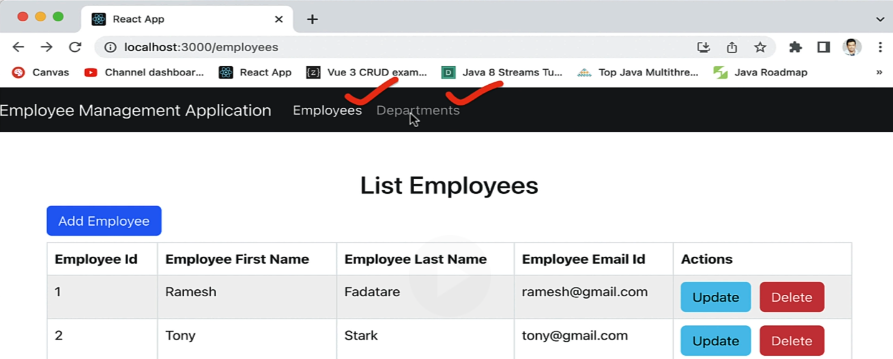
****

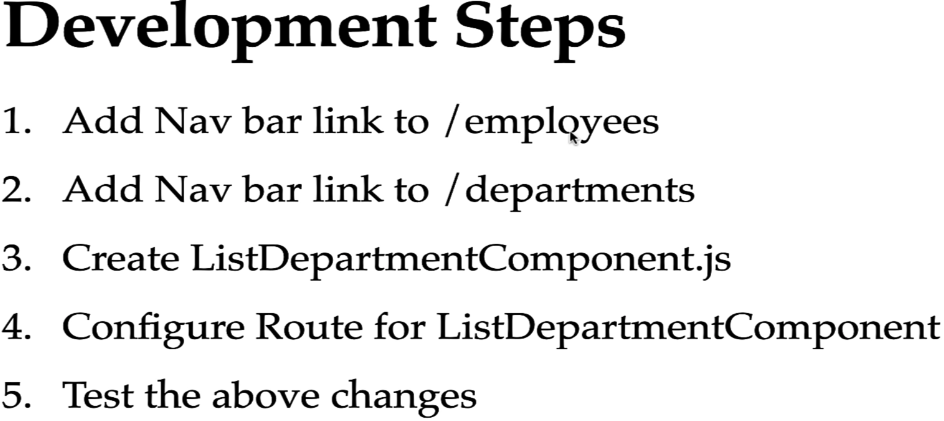




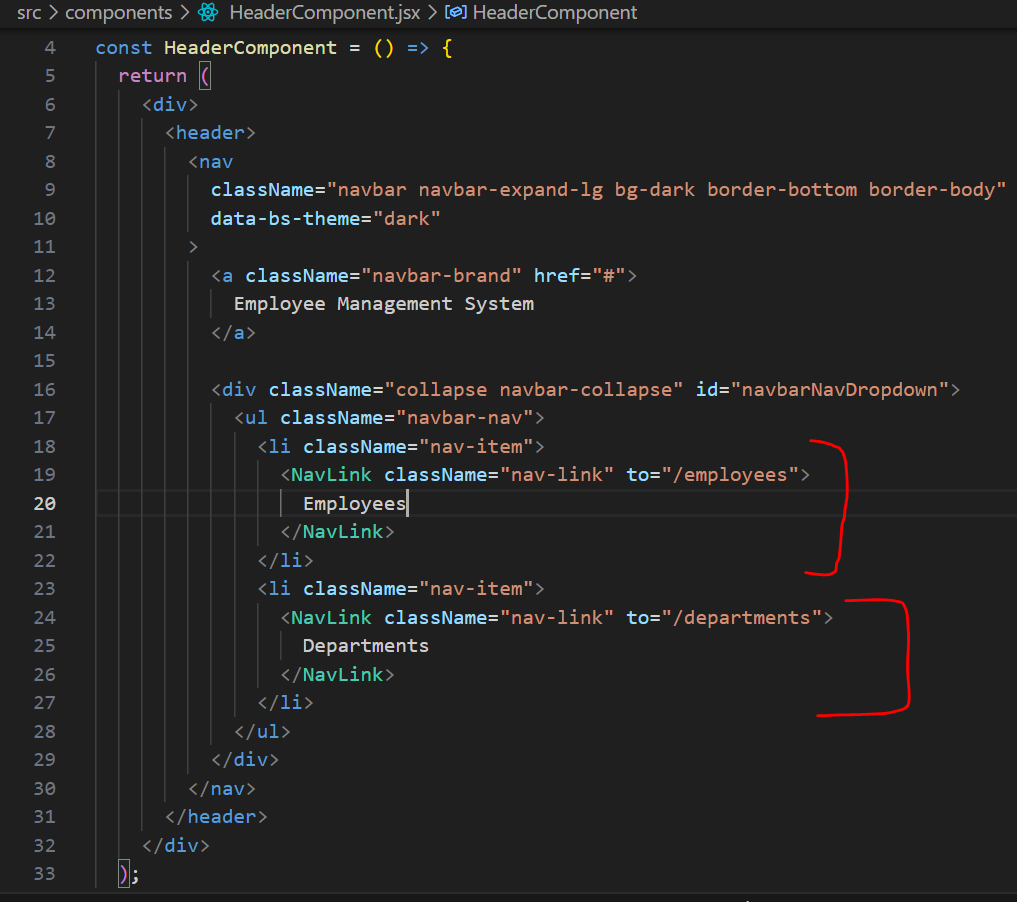
****

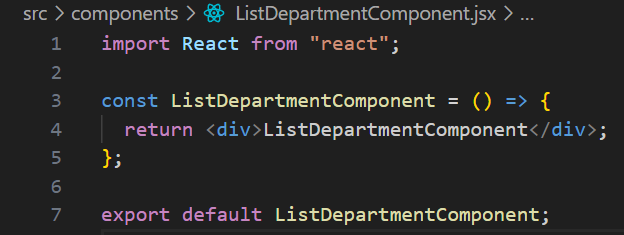
**Add Navigation Links in Header**

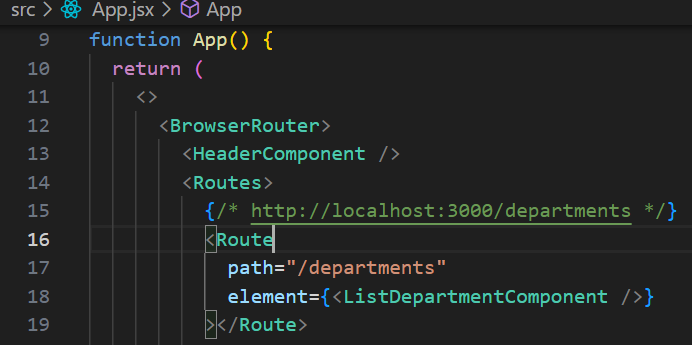
****

****

We used NavLink component from React-router-dom library to navigate user to the particular page.

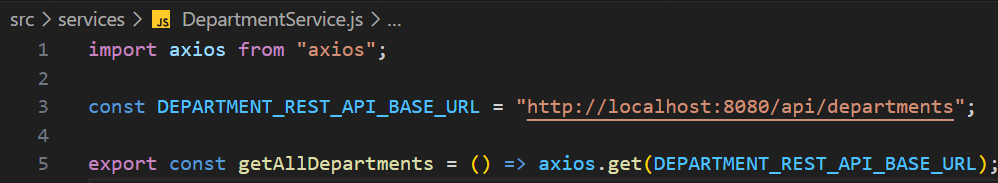






**Design List Department Component & Connect with Get All Departments REST API**

****

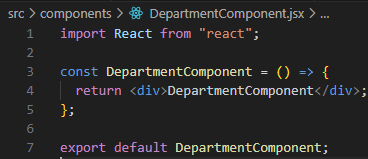


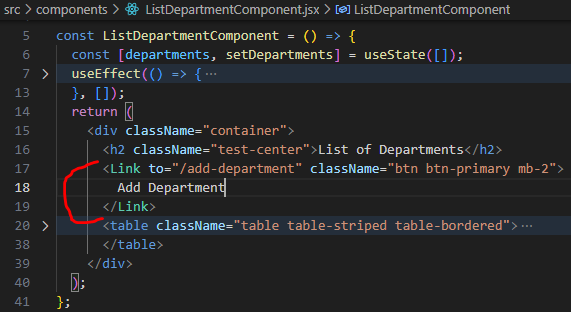


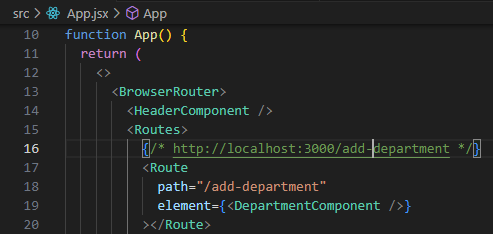
**Add Department Feature**

****

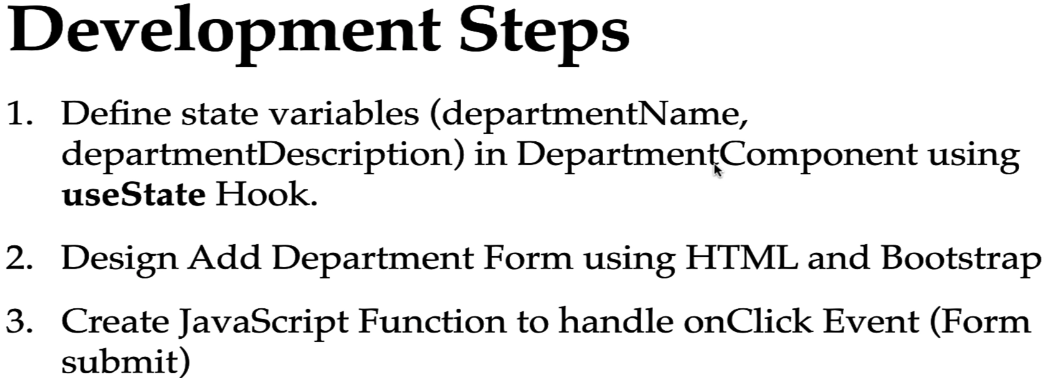
Inorder to add link, we’ll use Link component from react-router-dom library



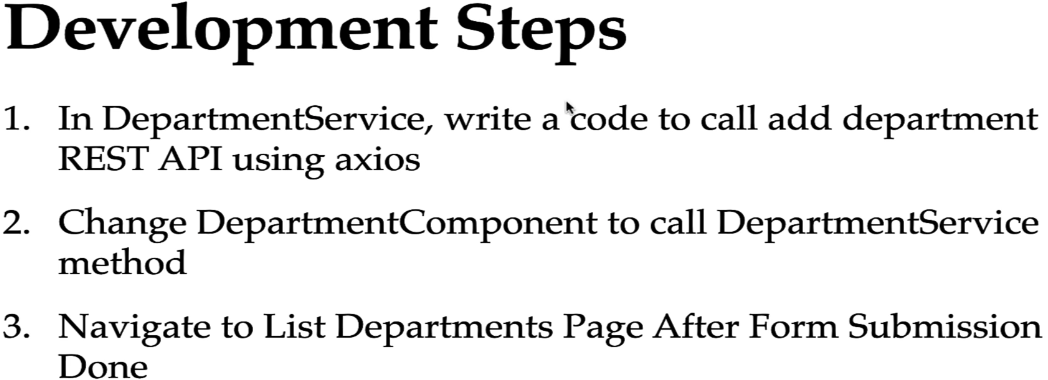


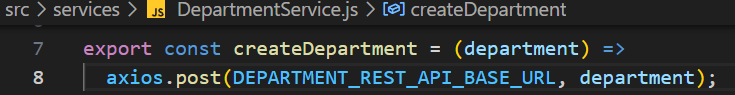


**Add Department Form Handling**



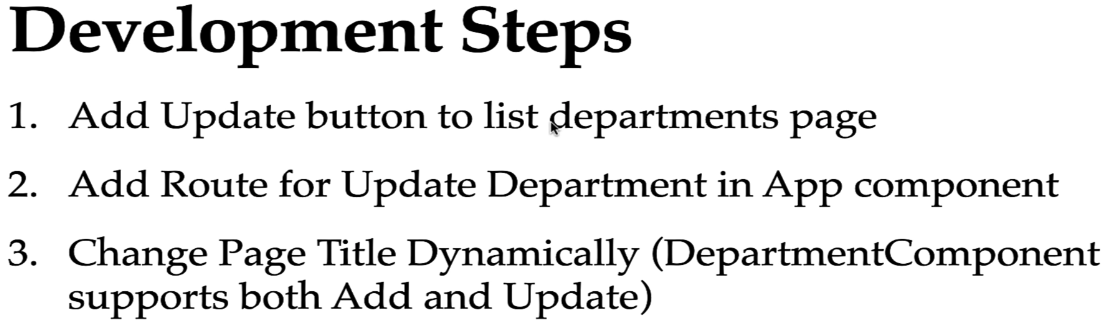
**Connect React to Add Department REST API**

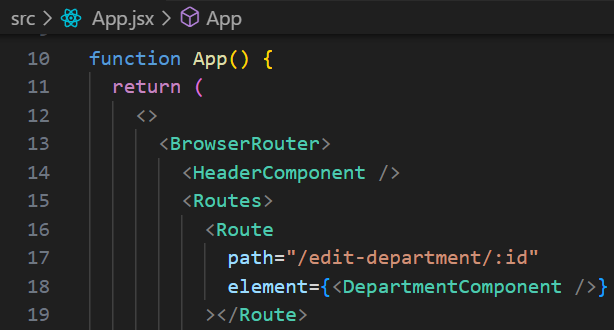
****

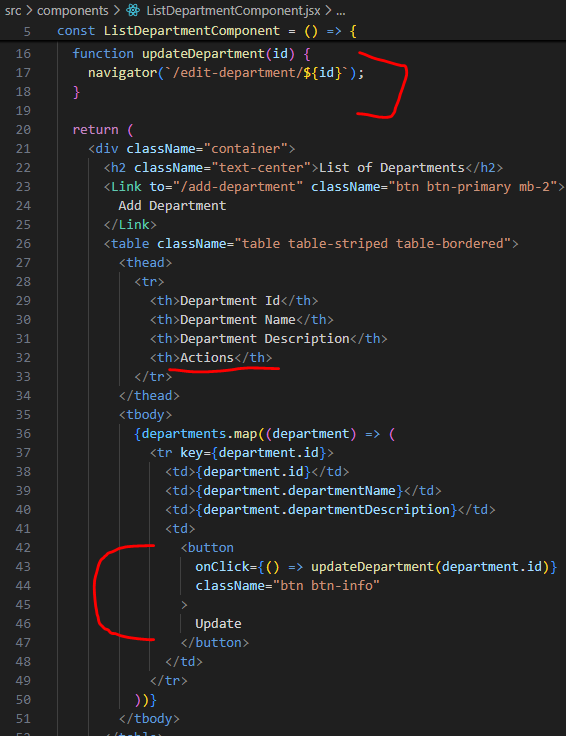


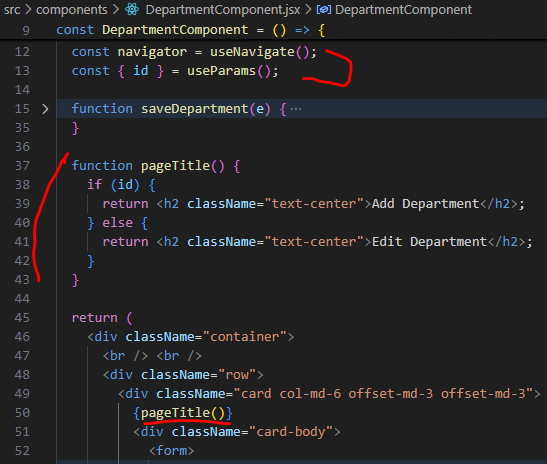


**Adding Update Button, Title and Route**

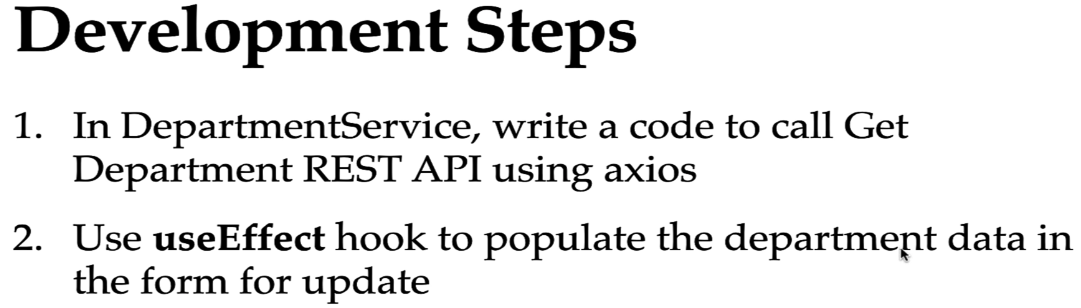
****

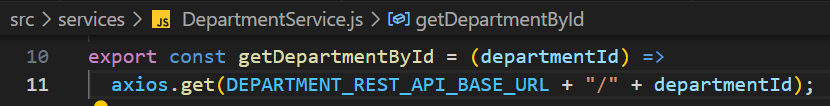


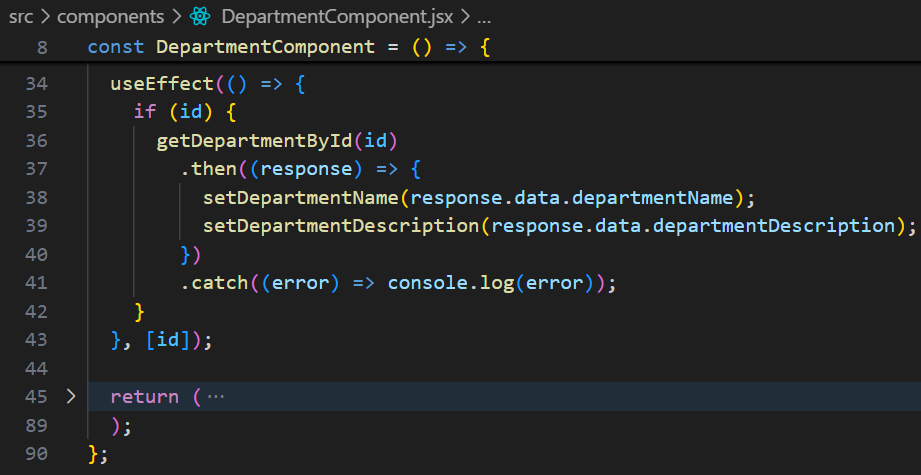




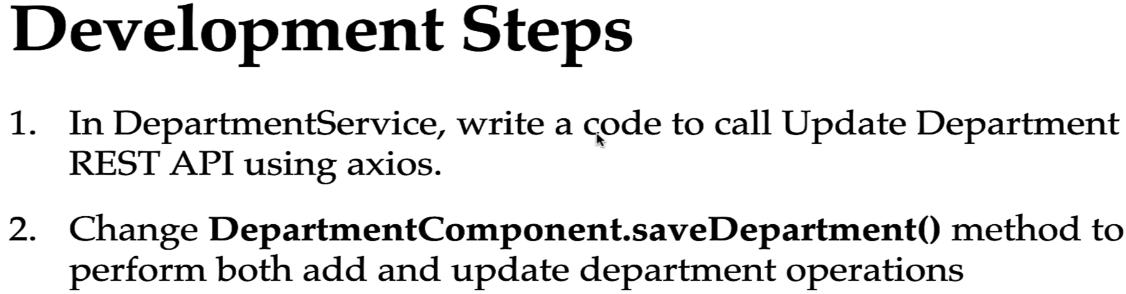
**Connect React App to Get Department REST API**

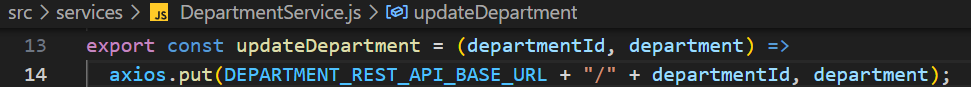
****

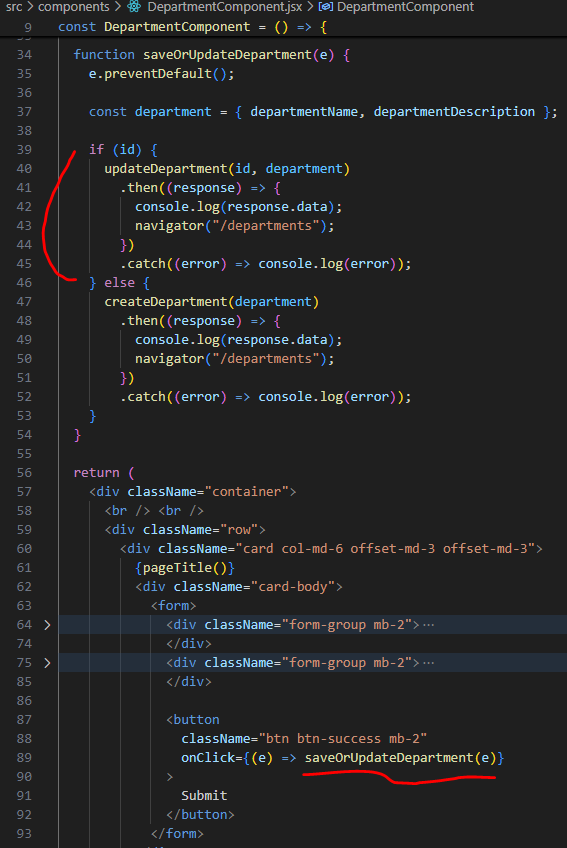




**Connect React App to Update Department REST API**

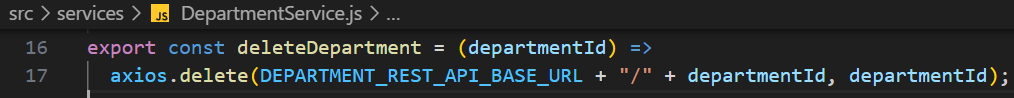
****





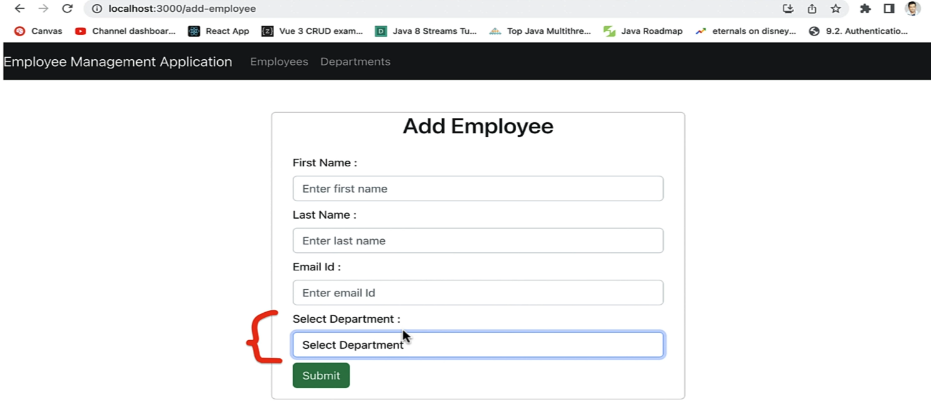
**Implement Delete Department Feature**

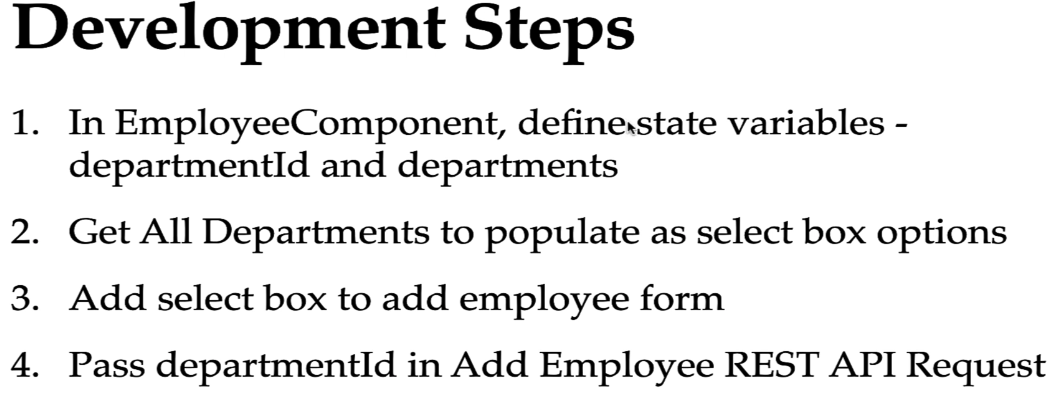
****





**Change Add & Update Employee Feature to use Department**

****

****

