React with SpringBoot Integration
++++++++++++++++++++++++++++++++

To-Do-List
++++++++++
1. Understand the Architecture of Integrating SpringBoot(Producer) with
React(Consumer)
      a. ReactApp will send the request using Axios(HTTP Library) to SpringBootApp
      b. Response will be sent in the form of JSON to ReactApp
      c. SpringBootApp(Producer::Java) and ReactApp(Consumer::JavaScript) --->
[Distrubuted Application environment(JSON)]

2. Flow diagram of SpringBoot
      a. Controller(SpringRest)
      b. Service(SpringAOP)
      c. DAO (DataJPA)
   Flow diagram of React
      a. Router(guides for what request what component should be rendered)
      b. Component(Tells what each thing represent of page)
      c. Service[AxiosHttpLibrary](Makes a API call to backend and gets the data)

3. Pre-Requistes for Backend and front-end
       refer diagram

4. Creating a SpringBoot project to make backend Ready
      ENDPOINT
      1. //GET ALL EMPLOYEES :: http://localhost:9999/api/v1/employees

5. Creating a UI using React
      D:\UI-Frontend> npx create-react-app react-frontend
      We suggest that you begin by typing:
      cd react-frontend
      npm start

        D:\UI-Frontend> cd react-frontend
      D:\UI-Frontend\react-frontend> npm start
      You can now view react-frontend in the browser.
            Local: http://localhost:3000
            On Your Network: http://192.168.0.102:3000

Scene1:Go to App.js and do the following changes to understand the control flow
++++++
App.js
++++++
```
import './App.css';
function App() {
  return (
    <div className="App">
         <h1>Welcome to the world of React</h1>
    </div>
  );
}
export default App;
```

Output: Welcome to the world of React

Control flow
  1> index.html-------id=root------> index.js --------root.render(<App/>)------

Component------> App.js

Scene2: Adding bootstrap4 to React App
Open new terminal and navigate to react-frontend

```
D:\UI-Frontend> cd react-frontend
D:\UI-Frontend\react-frontend> npm i bootstrap@4
```

Note: refer package.json file to see whether bootstrap is added or not to our project.

```
dependencies": {
    "@testing-library/jest-dom": "^5.17.0",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "bootstrap": "^4.6.2", //added to our project
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  }
```

Scene3:
1. Create a folder called component in src
2. Create a file called ListEmployeeComponent.jsx

ListEmployeeComponent.jsx

```
++++++++++++++++++++++++
import React, { useState } from 'react';

function ListEmployeeComponent() {

  const [employees, setEmployees] = useState([]);

  return (
    <div className="m-4">
      <h2 className="text-center">Employees List</h2>
      <div className="row">
        <table className="table table-striped table-bordered">
          <thead>
            <tr>
              <th> Employee First Name</th>
              <th> Employee Last Name</th>
              <th> Employee Email Id</th>
              <th> Actions</th>
            </tr>
          </thead>
          <tbody>
            {employees.map((employee) => (
              <tr key={employee.id}>
                <td> {employee.firstName}</td>
                <td> {employee.lastName}</td>
                <td> {employee.emailId}</td>
              </tr>
            ))}
          </tbody>
        </table>
      </div>
    </div>
  );
```

```
}

export default ListEmployeeComponent;
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++
In App.js make the following changes

App.js
+++++
import './App.css';
import 'bootstrap/dist/css/bootstrap.min.css'
import ListEmployeeComponent from './component/ListEmployeeComponent';


function App() {
  return (
    <div className="App">
         <ListEmployeeComponent/>
    </div>
  );
}

export default App;

output: Check the browser, it should display the empty page with the Employee Table
with no data(only columns)
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++
3. install axios library to our react project
D:\UI-Frontend\react-frontend> npm install axios --save

4.Create a folder called services in src folder

   Create a file called EmployeeService.js

++++++++++++++++++
EmployeeService.js
++++++++++++++++++

import axios from 'axios';

const EMPLOYEE_API_BASE_URL = "http://localhost:9999/api/v1/employees";

export const getEmployees = () => {
  return axios.get(EMPLOYEE_API_BASE_URL);
};

5. Make suitable changes in ListEmployeeComponent.jsx to call and API and render
the data on the page

++++++++++++++++++++++++
ListEmployeeComponent.jsx
++++++++++++++++++++++++
import React, { useState,useEffect } from 'react';
import {getEmployees} from "../services/EmployeeService"

function ListEmployeeComponent() {

  //employees <----setEmployees()
```

```
  const [employees, setEmployees] = useState([]);

  //React hook to get the JSON data
  useEffect(() => {
    //Calling API From BackEnd to get the data of Employees
    getEmployees().then((emp) => {
     //Inject data to employees varaible
      setEmployees(emp.data);
    });
  }, []);

  return (
    <div className="m-4">
      <h2 className="text-center">Employees List</h2>
      <div className="row">
        <table className="table table-striped table-bordered">
          <thead>
            <tr>
              <th> Employee First Name</th>
              <th> Employee Last Name</th>
              <th> Employee Email Id</th>
              <th> Actions</th>
            </tr>
          </thead>
          <tbody>
          //Rendering the data from employees variable
            {employees.map((employee) => (
              <tr key={employee.id}>
                <td> {employee.firstName}</td>
                <td> {employee.lastName}</td>
                <td> {employee.emailId}</td>
              </tr>
            ))}
          </tbody>
        </table>
      </div>
    </div>
  );
}

export default ListEmployeeComponent;
```

refresh the page to see the output(rendering of employees data coming from backend)

+++
++++++++++++++++++++++++++

7. Create header and footer component

+++++++++++++++++++

FooterComponent.jsx

+++++++++++++++++++

```
import React from 'react';

function FooterComponent() {
  return (
    <div>
      <footer className="footer">
        <span className="text-muted">All Rights Reserved 2023 @Ineuron</span>
      </footer>
```

```
    </div>
  );
}

export default FooterComponent;

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++

++++++++++++++++++
HeaderComponent.jsx
++++++++++++++++++
import React from 'react';
import { Link } from 'react-router-dom';

function HeaderComponent() {
  return (
    <div>
      <header>
        <nav className="navbar navbar-expand-md navbar-dark bg-dark">
          <div><Link to={"https://ineuron.ai"} className="navbar-brand">Employee
Management App</Link></div>
        </nav>
      </header>
    </div>
  );
}

export default HeaderComponent;

Make changes in App.css
++++++++++++++++++++++
.footer {
  position: absolute;
  bottom: 0;
  width:100%;
  height: 50px;
  background-color: black;
  text-align: center;
  color: white;
}
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++
Render App.js by linking HeaderComponent and FooterComponent(install react-router-
dom)
=> D:\UI-Frontend\react-frontend> npm install react-router-dom@6

++++++
App.js
++++++
import ListEmployeeComponent from './component/ListEmployeeComponent';
import HeaderComponent from './component/HeaderComponent';
import FooterComponent from './component/FooterComponent';
import { BrowserRouter as Router } from 'react-router-dom';
import { Routes, Route } from "react-router";

import 'bootstrap/dist/css/bootstrap.min.css'
import './App.css';
```

```
function App() {
  return (
    <div>
      <Router>
        <HeaderComponent />
            <div class="container">
            <Routes>
               <Route path='/' element={<ListEmployeeComponent />} />
            </Routes>
          </div>
        <FooterComponent />
      </Router>
    </div>
  );
}

export default App;
```

+++
+++++++++++++++++++++++++++++++++++

Create Employee Functionality
  a. ENDPOINTS : POST -> http://localhost:9999/api/v1/employees