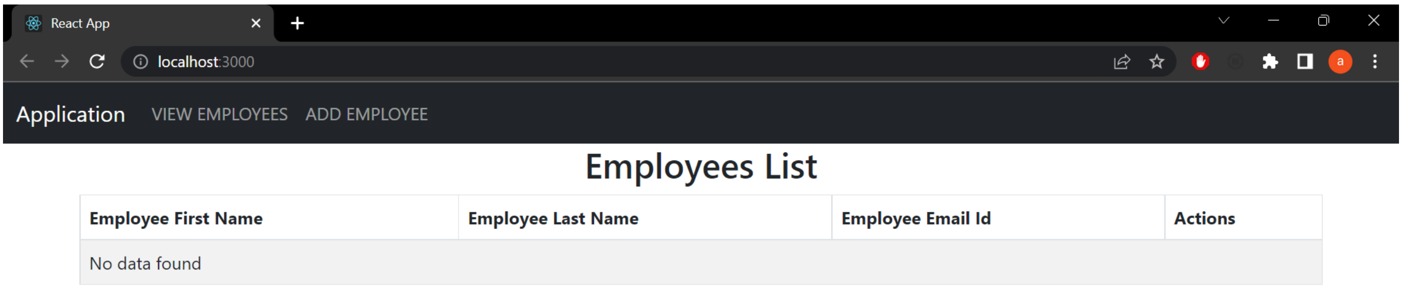
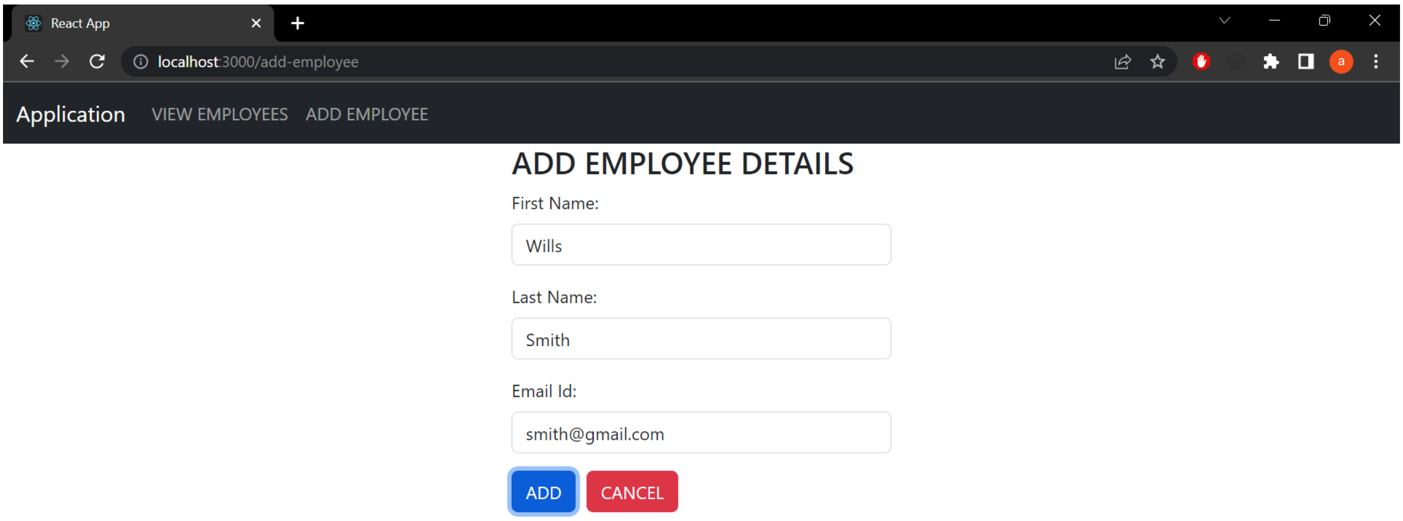
Integration of React and Spring Boot



Employees List

Employee First Name	Employee Last Name	Employee Email Id	Actions
No data found			

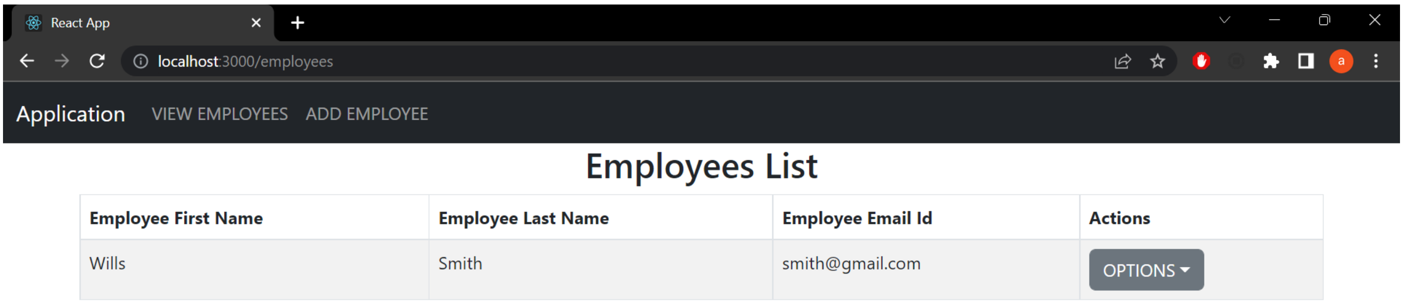


ADD EMPLOYEE DETAILS

First Name: Wills

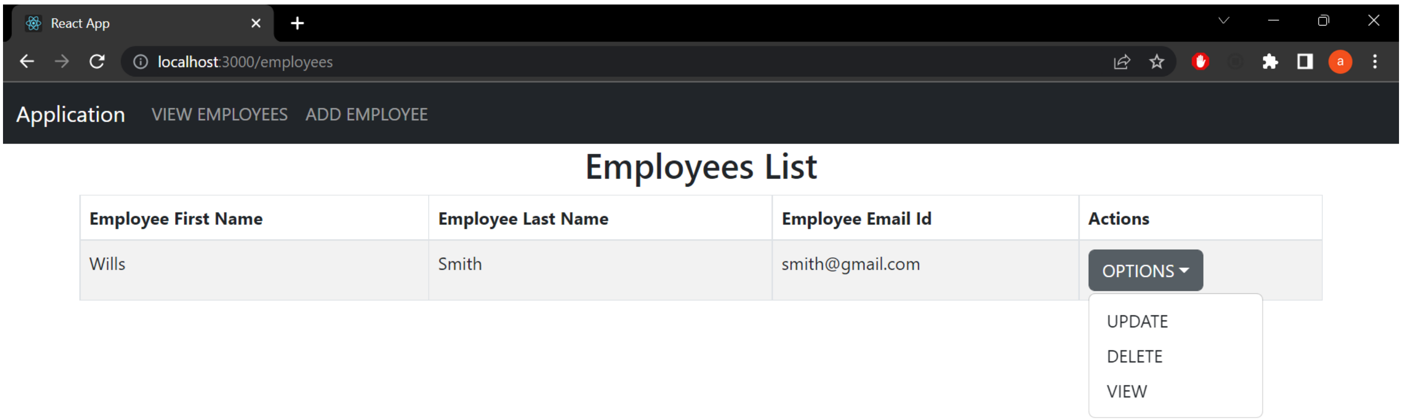
Last Name: Smith

Email Id: smith@gmail.com



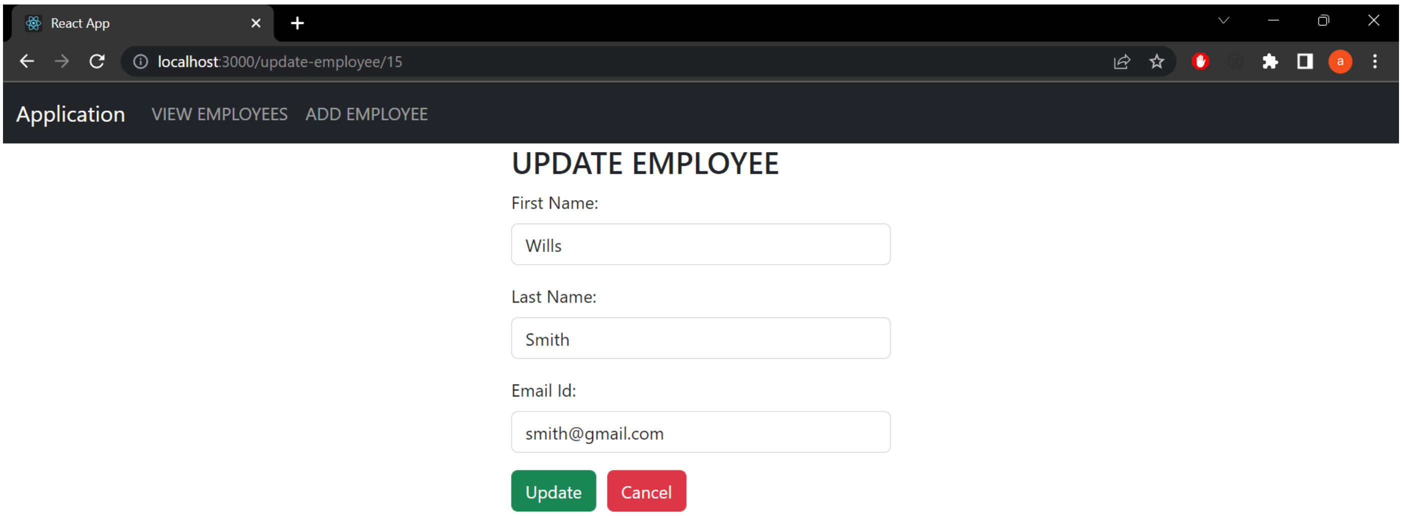
Employees List

Employee First Name	Employee Last Name	Employee Email Id	Actions
Wills	Smith	smith@gmail.com	OPTIONS ▾



Employees List

Employee First Name	Employee Last Name	Employee Email Id	Actions
Wills	Smith	smith@gmail.com	OPTIONS

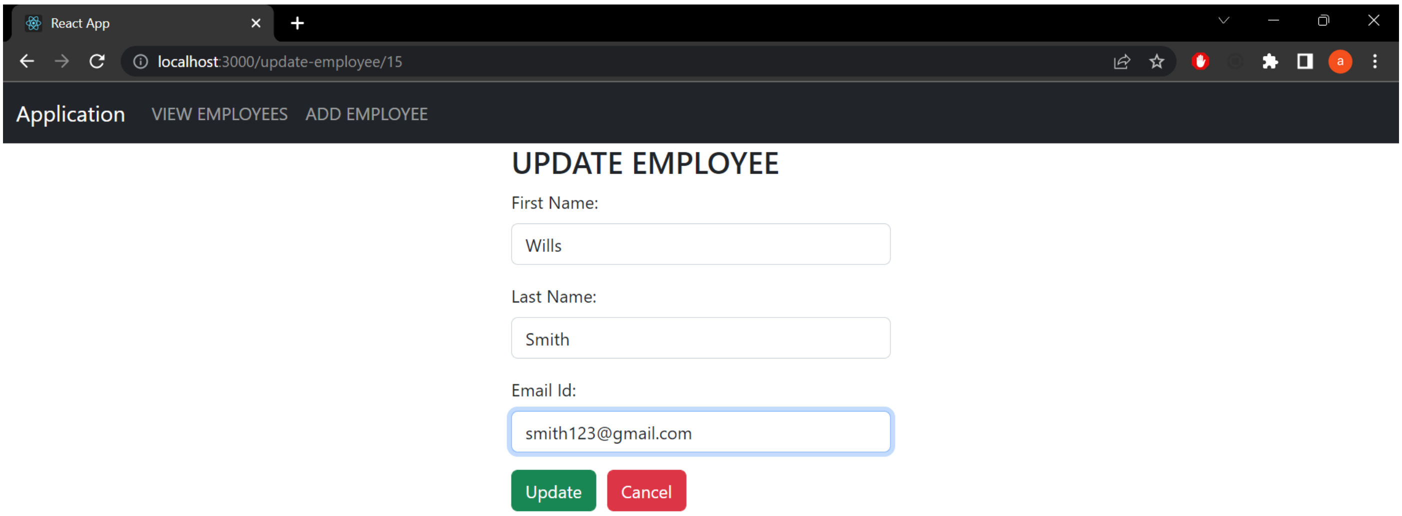


UPDATE EMPLOYEE

First Name: Wills

Last Name: Smith

Email Id: smith@gmail.com

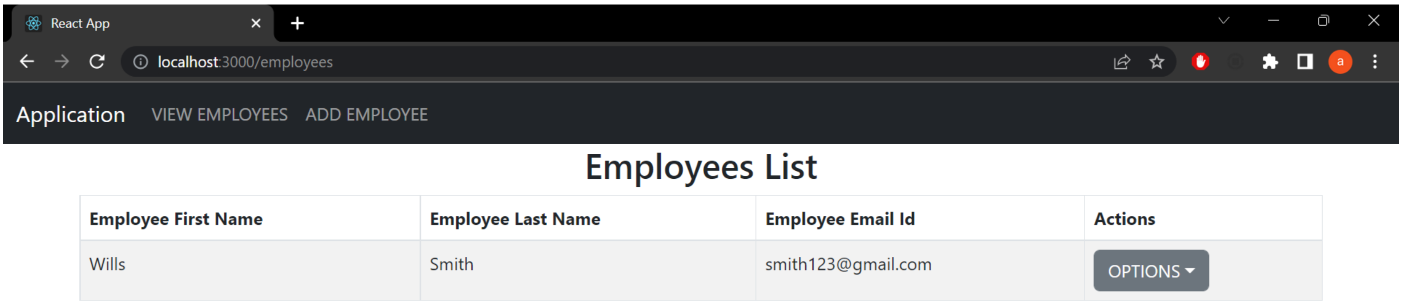


UPDATE EMPLOYEE

First Name: Wills

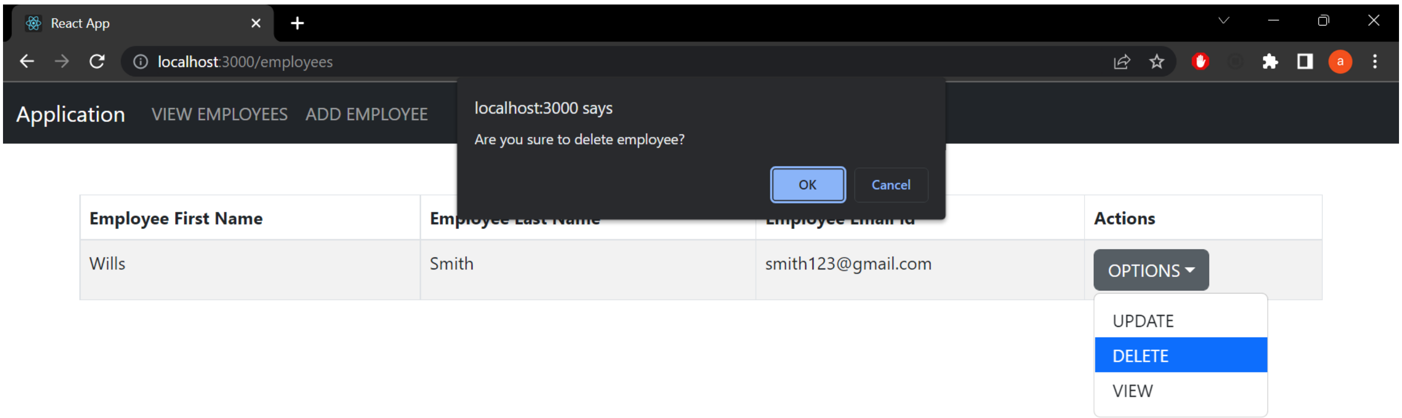
Last Name: Smith

Email Id: smith123@gmail.com



Employees List

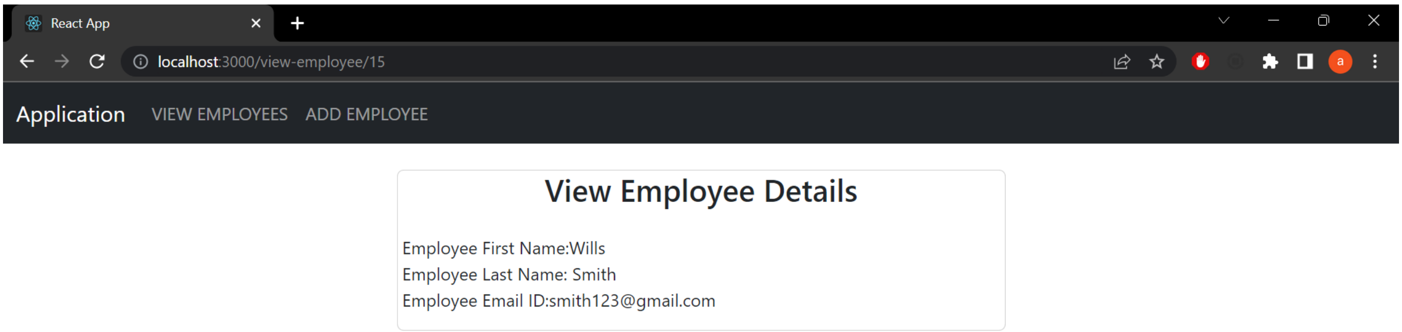
Employee First Name	Employee Last Name	Employee Email Id	Actions
Wills	Smith	smith123@gmail.com	OPTIONS



localhost:3000 says

Are you sure to delete employee?

Employee First Name	Employee Last Name	Employee Email Id	Actions
Wills	Smith	smith123@gmail.com	OPTIONS



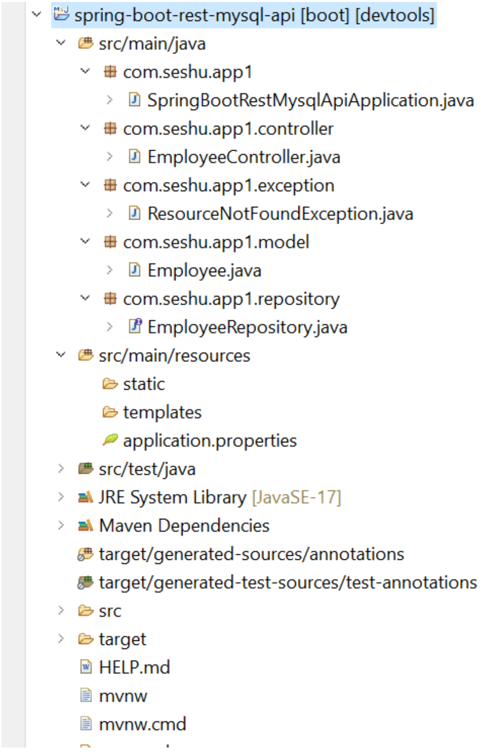
View Employee Details

Employee First Name:Wills
Employee Last Name: Smith
Employee Email ID:smith123@gmail.com

Developing Spring Boot Backend Application;

Create backend application with fallowing dependecies;

Spring-data-jpa
Spring-web
Devtools
Mysql connector



```
spring-boot-rest-mysql-api [boot] [devtools]
  src/main/java
    com.seshu.app1
      SpringBootRestMysqlApiApplication.java
    com.seshu.app1.controller
      EmployeeController.java
    com.seshu.app1.exception
      ResourceNotFoundException.java
    com.seshu.app1.model
      Employee.java
    com.seshu.app1.repository
      EmployeeRepository.java
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  JRE System Library [JavaSE-17]
  Maven Dependencies
  target/generated-sources/annotations
  target/generated-test-sources/test-annotations
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
     <modelVersion>4.0.0</modelVersion>
     <parent>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-starter-parent</artifactId>
          <version>2.7.4</version>
          <relativePath/> <!-- lookup parent from repository -->
     </parent>
     <groupId>com.seshu</groupId>
     <artifactId>spring-boot-rest-mysql-api</artifactId>
     <version>0.0.1-SNAPSHOT</version>
     <name>spring-boot-rest-mysql-api</name>
     <description>Demo project for Spring Boot</description>
     <properties>
          <java.version>17</java.version>
     </properties>
     <dependencies>
          <dependency>
               <groupId>org.springframework.boot</groupId>
```

```
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
                <configuration>
                    <excludes>
                        <exclude>
                            <groupId>org.projectlombok</groupId>
                            <artifactId>lombok</artifactId>
                        </exclude>
                    </excludes>
                </configuration>
            </plugin>
        </plugins>
    </build>

</project>
```

application.properties

```
server.port=8181

spring.datasource.url=jdbc:mysql://localhost:3306/adidb
spring.datasource.username=root
spring.datasource.password=seshu

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
```

Employee.java

```
package com.seshu.app1.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "employees")
public class Employee {

     @Id
     @GeneratedValue
     private long id;

     @Column(name = "first_name")
     private String firstName;

     @Column(name = "last_name")
     private String lastName;

     @Column(name = "email_id")
     private String emailId;

     public Employee() {

     }

     public Employee(String firstName, String lastName, String emailId)
{
          super();
          this.firstName = firstName;
          this.lastName = lastName;
          this.emailId = emailId;
     }
     public long getId() {
          return id;
     }
     public void setId(long id) {
          this.id = id;
     }
     public String getFirstName() {
          return firstName;
     }
     public void setFirstName(String firstName) {
          this.firstName = firstName;
     }
     public String getLastName() {
          return lastName;
     }
```

```
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public String getEmailId() {
        return emailId;
    }
    public void setEmailId(String emailId) {
        this.emailId = emailId;
    }
}
```

EmployeeRepository.java

```
package com.seshu.app1.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.seshu.app1.model.Employee;

@Repository
public interface EmployeeRepository extends JpaRepository<Employee,
Long>{

}
```

ResourceNotFoundException.java

```
package com.seshu.app1.exception;

import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.ResponseStatus;

@ResponseStatus(value = HttpStatus.NOT_FOUND)
public class ResourceNotFoundException extends RuntimeException{

    private static final long serialVersionUID = 1L;

    public ResourceNotFoundException(String message) {
        super(message);
    }
}
```

EmployeeController.java

```
package com.seshu.app1.controller;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.seshu.app1.exception.ResourceNotFoundException;
import com.seshu.app1.model.Employee;
import com.seshu.app1.repository.EmployeeRepository;

@CrossOrigin(origins = "http://localhost:3000")
@RestController
@RequestMapping("/api/v1/")
public class EmployeeController {

     @Autowired
     private EmployeeRepository employeeRepository;

     @GetMapping("/employees")
     public List<Employee> getAllEmployees(){
          return employeeRepository.findAll();
     }

     @PostMapping("/employees")
     public Employee createEmployee(@RequestBody Employee employee) {
          return employeeRepository.save(employee);
     }

     @GetMapping("/employees/{id}")
     public ResponseEntity<Employee> getEmployeeById(@PathVariable Long
id) {
          Employee employee = employeeRepository.findById(id)
                    .orElseThrow(() -> new
ResourceNotFoundException("Employee not exist with id :" + id));
          return ResponseEntity.ok(employee);
     }

     @PutMapping("/employees/{id}")
     public ResponseEntity<Employee> updateEmployee(@PathVariable Long 
id, @RequestBody Employee employeeDetails){
          Employee employee = employeeRepository.findById(id)
                    .orElseThrow(() -> new 
ResourceNotFoundException("Employee not exist with id :" + id));
          
          employee.setFirstName(employeeDetails.getFirstName());
          employee.setLastName(employeeDetails.getLastName());
          employee.setEmailId(employeeDetails.getEmailId());
          
          Employee updatedEmployee = employeeRepository.save(employee);
          return ResponseEntity.ok(updatedEmployee);
     }
     
     @DeleteMapping("/employees/{id}")
     public ResponseEntity<Map<String, Boolean>> 
deleteEmployee(@PathVariable Long id){
          Employee employee = employeeRepository.findById(id)
                    .orElseThrow(() -> new 
ResourceNotFoundException("Employee not exist with id :" + id));
          
          employeeRepository.delete(employee);
          
          Map<String, Boolean> response = new HashMap<>();
          response.put("deleted", Boolean.TRUE);
          
          return ResponseEntity.ok(response);
     }
}
```

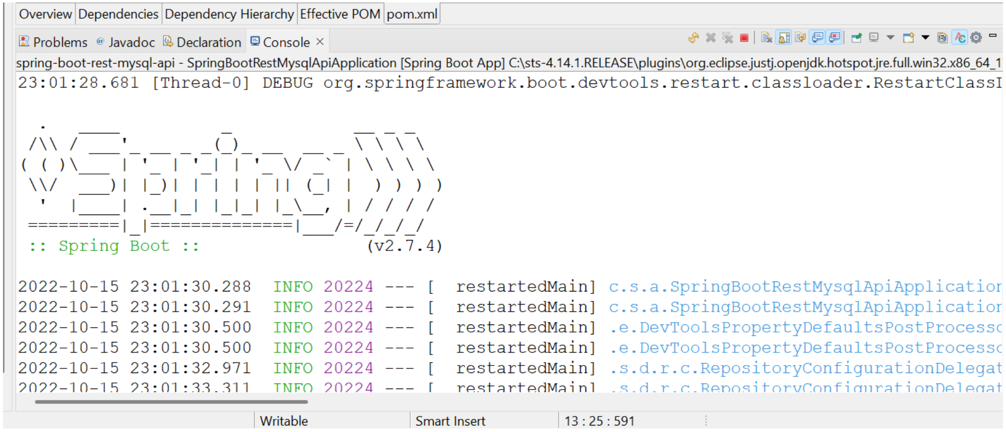
```
package com.seshu.app1;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class SpringBootRestMysqlApiApplication {

     public static void main(String[] args) {
     
     SpringApplication.run(SpringBootRestMysqlApiApplication.class, 
args);
     }
}
```

Run the spring boot starter;



spring-boot-rest-mysql-api - SpringBootRestMysqlApiApplication [Spring Boot App] C:\sts-4.14.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_1

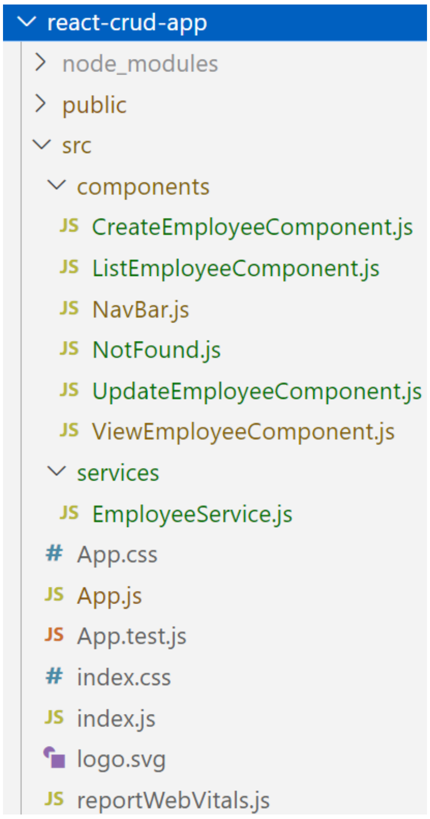
```
23:01:28.681 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassI

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.7.4)

2022-10-15 23:01:30.288  INFO 20224 --- [  restartedMain] c.s.a.SpringBootRestMysqlApiApplicatior
2022-10-15 23:01:30.291  INFO 20224 --- [  restartedMain] c.s.a.SpringBootRestMysqlApiApplicatior
2022-10-15 23:01:30.500  INFO 20224 --- [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessc
2022-10-15 23:01:30.500  INFO 20224 --- [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessc
2022-10-15 23:01:32.971  INFO 20224 --- [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegat
2022-10-15 23:01:33.311  INFO 20224 --- [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegat
```

Writable | Smart Insert | 13 : 25 : 591

Developing Front-end application (React Application)



```
react-crud-app
  node_modules
  public
  src
    components
      CreateEmployeeComponent.js
      ListEmployeeComponent.js
      NavBar.js
      NotFound.js
      UpdateEmployeeComponent.js
      ViewEmployeeComponent.js
    services
      EmployeeService.js
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    reportWebVitals.js
```

Steps:

Generate react application

```
>npx create-react-app react-crud-app
```

Install axios and react-router-dom

```
react-crud-app>npm install axios
react-crud-app >npm install react-router-dom
```

https://reactrouter.com/en/v6.3.0/getting-started/overview

Add bootstrap cdn links

react-crud-app\public\index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <title>React App</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
      crossorigin="anonymous"
    />
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>

    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-
u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
      crossorigin="anonymous"
    ></script>
  </body>
</html>
```

Create EmployeeService

react-crud-app\src\services\EmployeeService.js

```
import axios from "axios";

const EMPLOYEE_API_BASE_URL = "http://localhost:8181/api/v1/employees";

class EmployeeService {
  getEmployees() {
    return axios.get(EMPLOYEE_API_BASE_URL);
  }

  createEmployee(employee) {
    return axios.post(EMPLOYEE_API_BASE_URL, employee);
  }

  getEmployeeById(employeeId) {
    return axios.get(EMPLOYEE_API_BASE_URL + "/" + employeeId);
  }

  updateEmployee(employee, employeeId) {
    return axios.put(EMPLOYEE_API_BASE_URL + "/" + employeeId, employee);
  }

  deleteEmployee(employeeId) {
    return axios.delete(EMPLOYEE_API_BASE_URL + "/" + employeeId);
  }
}

export default new EmployeeService();
```

Developing ListEmployeeComponent;

react-crud-app\src\components\ListEmployeeComponent.js

```
import React, { useState, useEffect } from "react";
import EmployeeService from "../services/EmployeeService";
import { Link } from "react-router-dom";

const ListEmployeeComponent = () => {
  const [employees, setEmployees] = useState([]);

  useEffect(() => {
    EmployeeService.getEmployees().then((res) => {
      setEmployees(res.data);
    });
  }, []);

  const deleteEmployee = (id) => {
    if (window.confirm("Are you sure to delete employee?")) {
      EmployeeService.deleteEmployee(id).then((res) => {
        setEmployees(employees.filter((employee) => employee.id !== id));
      });
    } else {
      return;
    }
  };

  return (
    <div className="container">
      <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>
                <td>
                  <div className="dropdown">
                    <button
                      className="btn btn-secondary dropdown-toggle"
                      type="button"
                      data-bs-toggle="dropdown"
                      aria-expanded="false"
                    >
                      OPTIONS
                    </button>
                    <ul className="dropdown-menu">
                      <li>
                        <Link
                          className="btn btn-primary dropdown-item"
                          to={`/update-employee/${employee.id}`}
                        >
                          UPDATE
                        </Link>
                      </li>
                      <li>
                        <button
                          onClick={() => deleteEmployee(employee.id)}
                          className="btn btn-danger dropdown-item"
                        >
                          DELETE
                        </button>
                      </li>
                      <li>
                        <Link
                          className="btn btn-success dropdown-item"
                          to={`/view-employee/${employee.id}`}
                        >
                          VIEW
                        </Link>
                      </li>
                    </ul>
                  </div>
                </td>
              </tr>
            ))}
            {employees.length === 0 && (
              <tr>
                <td colSpan="4">No data found</td>
              </tr>
            )}
          </tbody>
        </table>
      </div>
    </div>
  );
};

export default ListEmployeeComponent;
```

Develop CreateEmployeeComponent;

react-crud-app\src\components\CreateEmployeeComponent.js

```
import React, { useState } from "react";
import { useNavigate } from "react-router-dom";
import EmployeeService from "../services/EmployeeService";

const CreateEmployeeComponent = (props) => {
  const navigate = useNavigate();
  const initialData = {
    firstName: "",
    lastName: "",
    emailId: "",
  };
  const [employee, setEmployee] = useState(initialData);
  const { firstName, lastName, emailId } = employee;

  const handleInputChange = (e) => {
    setEmployee({ ...employee, [e.target.name]: e.target.value });
  };

  const saveEmployee = (e) => {
    e.preventDefault();
    EmployeeService.createEmployee(employee).then((res) => {
      navigate("/employees");
    });
  };

  return (
    <div className="container col-4">
      <h3>ADD EMPLOYEE DETAILS</h3>
      <div>
        <form>
          <div className="mb-3">
            <label className="form-label"> First Name: </label>
            <input
              name="firstName"
              className="form-control"
              value={firstName}
              onChange={(e) => handleInputChange(e)}
            />
          </div>
          <div className="mb-3">
            <label className="form-label"> Last Name: </label>
            <input
              name="lastName"
              className="form-control"
              value={lastName}
              onChange={(e) => handleInputChange(e)}
            />
          </div>
          <div className="mb-3">
            <label className="form-label"> Email Id: </label>
            <input
              name="emailId"
              className="form-control"
              value={emailId}
              onChange={(e) => handleInputChange(e)}
            />
          </div>

          <button className="btn btn-primary" onClick={(e) => saveEmployee(e)}>
            ADD
          </button>
        </form>
      </div>
    </div>
  );
};

export default CreateEmployeeComponent;
```

Develop ViewEmployeeComponent;

react-crud-app\src\components\ViewEmployeeComponent.js

```
import React, { useState, useEffect } from "react";
import { useParams } from "react-router-dom";
import EmployeeService from "../services/EmployeeService";

const ViewEmployeeComponent = () => {
  const initialData = {
    firstName: "",
    lastName: "",
    emailId: "",
  };

  const [employee, setEmployee] = useState(initialData);

  const params = useParams();

  useEffect(() => {
    EmployeeService.getEmployeeById(params.id).then((res) => {
      setEmployee(res.data);
    });
  }, []);

  return (
    <div>
      <br></br>
      <div className="card col-md-6 offset-md-3">
        <h3 className="text-center"> View Employee Details</h3>
        <div className="card-body">
          <div className="row">
            Employee First Name:
            {employee.firstName}
          </div>
          <div className="row">Employee Last Name: {employee.lastName}</div>
          <div className="row">
            Employee Email ID:
            {employee.emailId}
          </div>
        </div>
      </div>
    </div>
  );
};

export default ViewEmployeeComponent;
```

Develop UpdateEmployeeComponent;

react-crud-app\src\components\UpdateEmployeeComponent.js

```
import React, { useState, useEffect } from "react";
import { useNavigate, useParams } from "react-router-dom";
import EmployeeService from "../services/EmployeeService";

const UpdateEmployeeComponent = () => {
  const params = useParams();
  const navigate = useNavigate();

  const initialData = {
    firstName: "",
    lastName: "",
    emailId: "",
  };

  const [employee, setEmployee] = useState(initialData);

  const { firstName, lastName, emailId } = employee;

  const onInputChange = (e) => {
    setEmployee({ ...employee, [e.target.name]: e.target.value });
  };

  useEffect(() => {
    EmployeeService.getEmployeeById(params.id).then((res) => {
      setEmployee(res.data);
    });
  }, []);

  const onSubmit = (e) => {
    e.preventDefault();
    EmployeeService.updateEmployee(employee, params.id).then((res) => {
      navigate("/employees");
    });
  };

  const cancel = () => {
    navigate("/employees");
  };

  return (
    <div className="container col-4">
      <h3>UPDATE EMPLOYEE</h3>
      <form>
        <div className="mb-3">
          <label className="form-label"> First Name: </label>
          <input
            placeholder="First Name"
            name="firstName"
            className="form-control"
            value={firstName}
            onChange={(e) => onInputChange(e)}
          />
        </div>
        <div className="mb-3">
          <label className="form-label"> Last Name: </label>
          <input
            placeholder="Last Name"
            name="lastName"
            className="form-control"
            value={lastName}
            onChange={(e) => onInputChange(e)}
          />
        </div>
        <div className="mb-3">
          <label className="form-label"> Email Id: </label>
          <input
            placeholder="Email Address"
            name="emailId"
            className="form-control"
            value={emailId}
            onChange={(e) => onInputChange(e)}
          />
        </div>

        <button className="btn btn-success" onClick={(e) => onSubmit(e)}>
          Update
        </button>
        <button
          className="btn btn-danger"
          onClick={() => cancel()}
          style={{ marginLeft: "10px" }}
        >
          Cancel
        </button>
      </form>
    </div>
  );
};

export default UpdateEmployeeComponent;
```

Develop NavBar;

react-crud-app\src\components\NavBar.js

```
import { Outlet, Link } from "react-router-dom";

const NavBar = () => {
  return (
    <>
      <nav className="navbar navbar-expand-lg navbar-dark bg-dark">
        <div className="container-fluid">
          <a className="navbar-brand" href="#">
            Application
          </a>
          <button
            className="navbar-toggler"
            type="button"
            data-bs-toggle="collapse"
            data-bs-target="#navbarSupportedContent"
            aria-controls="navbarSupportedContent"
            aria-expanded="false"
            aria-label="Toggle navigation"
          >
            <span className="navbar-toggler-icon"></span>
          </button>
          <div className="collapse navbar-collapse" id="navbarSupportedContent">
            <ul className="navbar-nav me-auto mb-2 mb-lg-0">
              <li className="nav-item">
                <Link to="/employees" className="nav-link">
                  VIEW EMPLOYEES
                </Link>
              </li>
              <li className="nav-item">
                <Link to="/add-employee" className="nav-link">
                  ADD EMPLOYEE
                </Link>
              </li>
            </ul>
          </div>
        </div>
      </nav>

      <Outlet />
    </>
  );
};
export default NavBar;
```

NotFound Component;

react-crud-app\src\components\NotFound.js

```
const NotFound = () => {
  return (
    <>
      <h2>Page Not Found</h2>
    </>
  );
};

export default NotFound;
```

react-crud-app\src\App.js

```
import React from "react";
import "./App.css";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import ListEmployeeComponent from "./components/ListEmployeeComponent";
import CreateEmployeeComponent from "./components/CreateEmployeeComponent";
import UpdateEmployeeComponent from "./components/UpdateEmployeeComponent";
import ViewEmployeeComponent from "./components/ViewEmployeeComponent";
import NavBar from "./components/NavBar";
import NotFound from "./components/NotFound";

function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<NavBar />}>
          <Route path="employees" element={<ListEmployeeComponent />} />
          <Route path="add-employee" element={<CreateEmployeeComponent />} />
          <Route path="view-employee/:id" element={<ViewEmployeeComponent />} />
          <Route
            path="update-employee/:id"
            element={<UpdateEmployeeComponent />}
          />
          <Route path="*" element={<NotFound />} />
        </Route>
      </Routes>
    </BrowserRouter>
  );
}

export default App;
```

react-crud-app\src\index.js

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

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById("root")
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

Run the react application;

react-crud-app>npm start