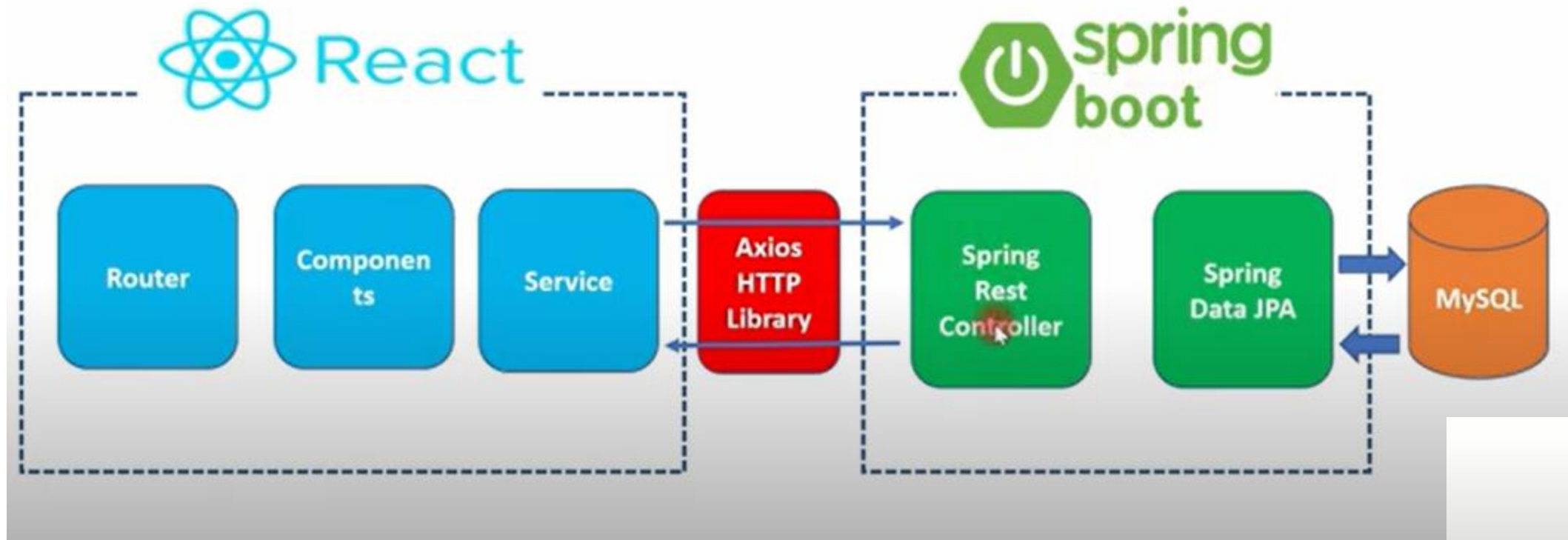


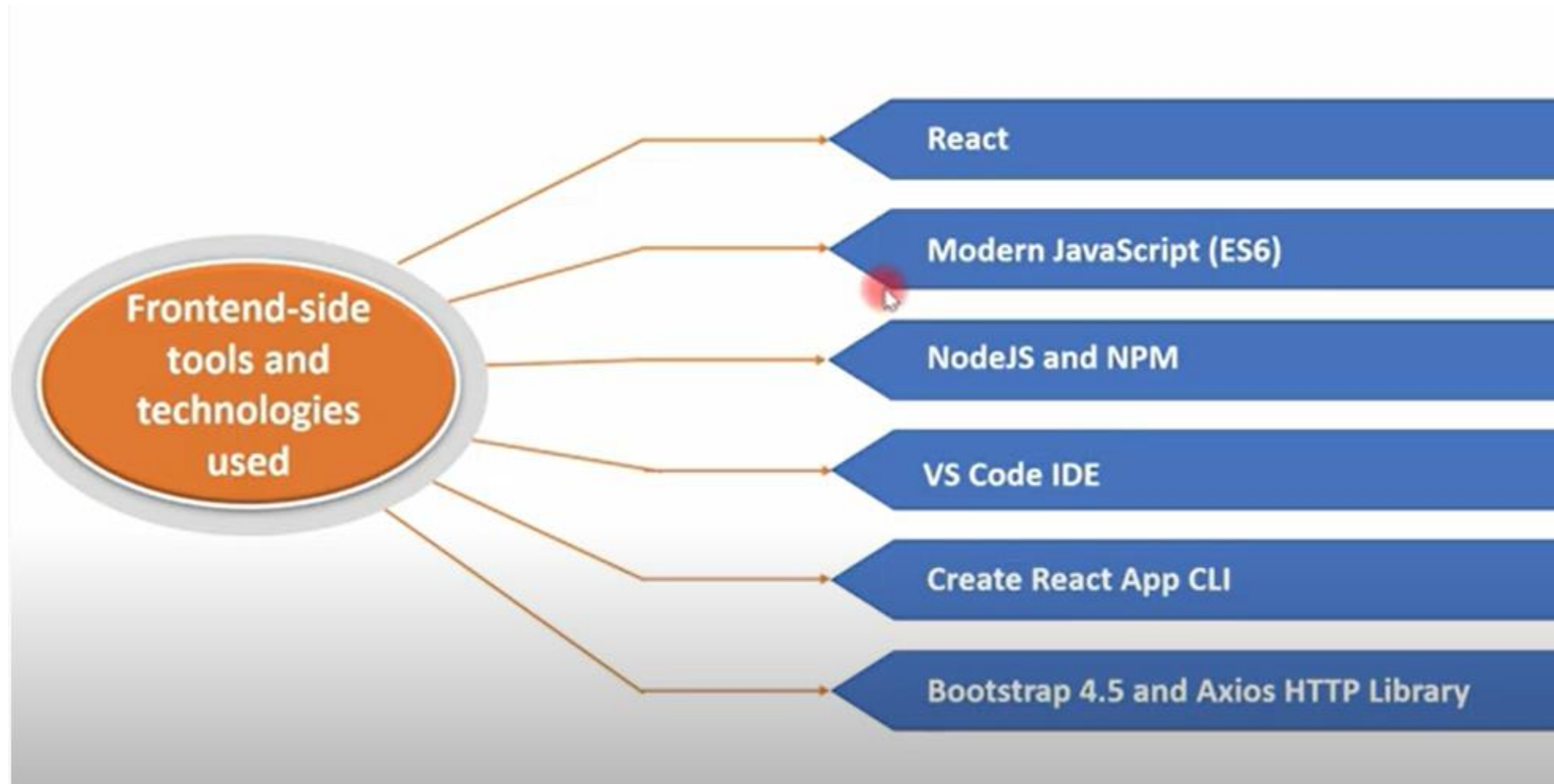
Employee Management System – Spring Boot, MySQL, ReactJS

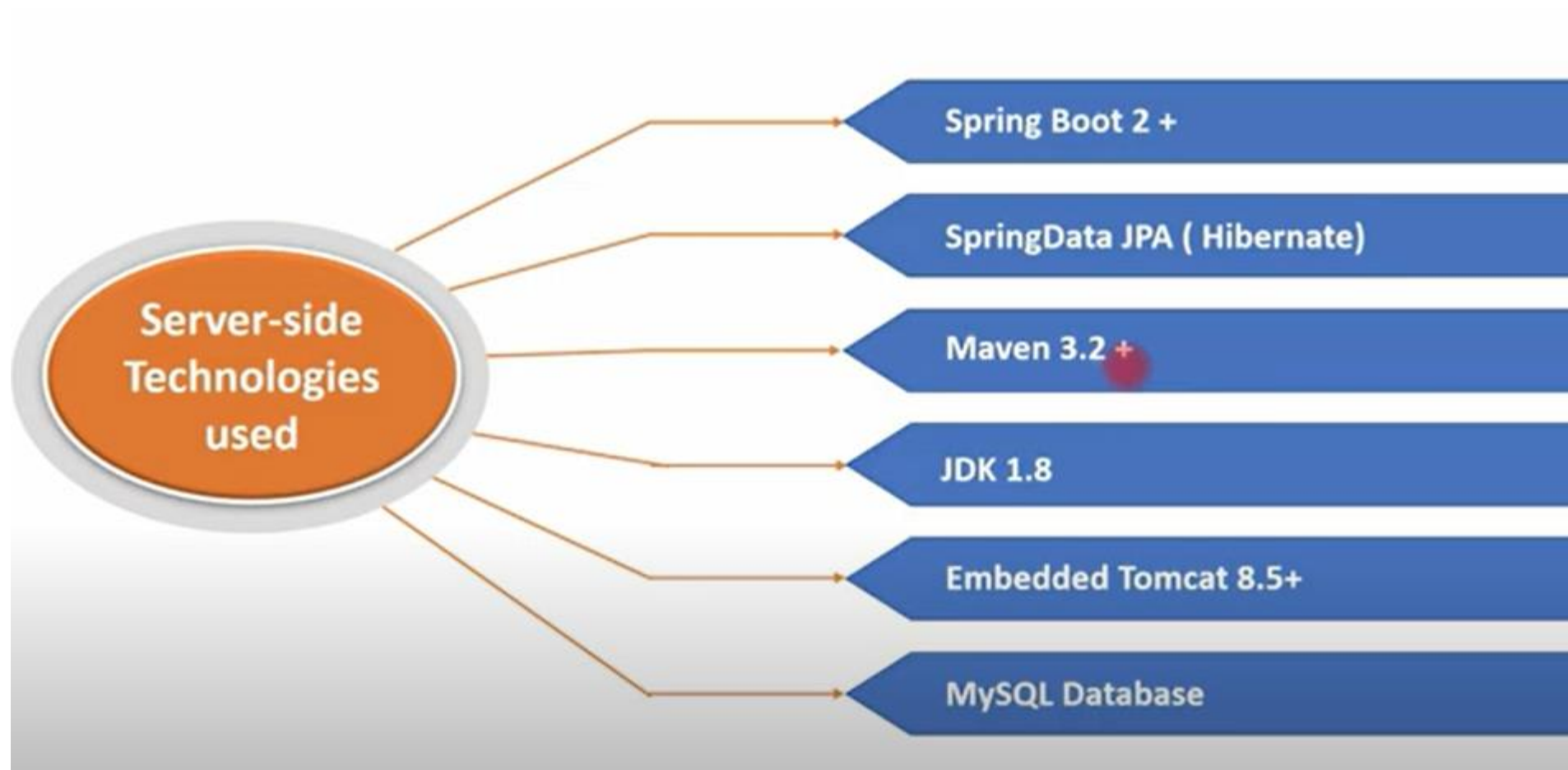
Client server Architecture



Spring Boot + React Full Stack App Architecture







Employee Management System –MySQL

```
1 • create database suresh;
2 • use suresh;
3 • drop table employees;
4 • create table employees(id int primary key auto_increment,first_name varchar(25),last_name varchar(25),email_id varchar(25));
5 • insert into employees(first_name,last_name,email_id)values('ram','kumar','ram@gmail.com');
6 • insert into employees(first_name,last_name,email_id)values('shiva','kumar','shiva@gmail.com');
7 • insert into employees(first_name,last_name,email_id)values('krishna','kumar','krishna@gmail.com');
8 • select * from employees;
```

id	first_name	last_name	email_id
1	ram	kumar	ram@gmail.com
2	shiva	kumar	shiva@gmail.com
3	krishna	kumar	krishna@gmail.com

Employee Management System – Spring Boot

EMS

```
springboot-backend [boot] [devtools]
├── src/main/java
│   ├── com.suresh.demos.springboot
│   │   └── SpringbootBackendApplication.java
│   ├── com.suresh.demos.springboot.controller
│   │   └── EmployeeController.java
│   ├── com.suresh.demos.springboot.exception
│   │   └── ResourceNotFoundException.java
│   ├── com.suresh.demos.springboot.model
│   │   └── Employee.java
│   ├── com.suresh.demos.springboot.repository
│   │   └── EmployeeRepository.java
│   └── src/main/resources
│       └── application.properties
├── src/test/java
│   └── com.suresh.demos.springboot
│       └── SpringbootBackendApplicationTests.java
├── JRE System Library [JavaSE-1.8]
├── Maven Dependencies
├── bin
├── src
│   ├── main
│   └── test
├── target
├── mvnw
├── mvnw.cmd
└── pom.xml
```

```
<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.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
    <exclusions>
      <exclusion>
        <groupId>org.junit.vintage</groupId>
        <artifactId>junit-vintage-engine</artifactId>
      </exclusion>
    </exclusions>
  </dependency>
</dependencies>
```

```
1 package com.suresh.demos.springboot.model;
2
3 import javax.persistence.*;
4
5 @Entity
6 @Table(name = "employees")
7 public class Employee {
8     @Id
9     @GeneratedValue(strategy = GenerationType.IDENTITY)
10    private long id;
11    @Column(name = "first_name")
12    private String firstName;
13
14    @Column(name = "last_name")
15    private String lastName;
16
17    @Column(name = "email_id")
18    private String emailId;
19
20    //getters and setters
21    //constructors
22 }
```

```
1 spring.datasource.url=jdbc:mysql://localhost:3306/suresh?useSSL=false
2 spring.datasource.username=root
3 spring.datasource.password=root
4
5 spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8Dialect
6
7 spring.jpa.hibernate.ddl-auto = update
8 server.port=8080
```

```
1 package com.suresh.demos.springboot.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4 import org.springframework.stereotype.Repository;
5
6 import com.suresh.demos.springboot.model.Employee;
7
8 @Repository
9 public interface EmployeeRepository extends JpaRepository<Employee, Long>{
10
11 }
```

```

1 package com.suresh.demos.springboot.controller;
2 import java.util.HashMap;
3 import java.util.List;
4 import java.util.Map;
5
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.http.ResponseEntity;
8 import org.springframework.web.bind.annotation.CrossOrigin;
9 import org.springframework.web.bind.annotation.DeleteMapping;
10 import org.springframework.web.bind.annotation.GetMapping;
11 import org.springframework.web.bind.annotation.PathVariable;
12 import org.springframework.web.bind.annotation.PostMapping;
13 import org.springframework.web.bind.annotation.PutMapping;
14 import org.springframework.web.bind.annotation.RequestBody;
15 import org.springframework.web.bind.annotation.RequestMapping;
16 import org.springframework.web.bind.annotation.RestController;
17
18 import com.suresh.demos.springboot.exception.ResourceNotFoundException;
19 import com.suresh.demos.springboot.model.Employee;
20 import com.suresh.demos.springboot.repository.EmployeeRepository;

```

```

22 @CrossOrigin(origins = "http://localhost:3000")
23 @RestController
24 @RequestMapping("/api/v1/")
25 public class EmployeeController {
26     @Autowired
27     EmployeeRepository employeeRepository;
28
29     @GetMapping("/employees")
30     public List<Employee> getAllEmployees() {
31         return employeeRepository.findAll();
32     }
33     @PostMapping("/employees")
34     public Employee createEmployee(@RequestBody Employee employee) {
35         return employeeRepository.save(employee);
36     }
37     @GetMapping("/employees/{id}")
38     public ResponseEntity<Employee> getEmployeeById(@PathVariable Long id) {
39         Employee employee = employeeRepository.findById(id)
40             .orElseThrow(() ->
41                 new ResourceNotFoundException("Employee not exist with id :" + id));
42         return ResponseEntity.ok(employee);
43     }

```

```

45     @PutMapping("/employees/{id}")
46     public ResponseEntity<Employee> updateEmployee(@PathVariable Long id,
47         @RequestBody Employee employeeDetails) {
48         Employee employee = employeeRepository.findById(id)
49             .orElseThrow(() ->
50                 new ResourceNotFoundException("Employee not exist with id :" + id));
51
52         employee.setFirstName(employeeDetails.getFirstName());
53         employee.setLastName(employeeDetails.getLastName());
54         employee.setEmailId(employeeDetails.getEmailId());
55
56         Employee updatedEmployee = employeeRepository.save(employee);
57         return ResponseEntity.ok(updatedEmployee);
58     }
59
60     @DeleteMapping("/employees/{id}")
61     public ResponseEntity<Map<String, Boolean>> deleteEmployee(@PathVariable Long id) {
62         Employee employee = employeeRepository.findById(id)
63             .orElseThrow(() ->
64                 new ResourceNotFoundException("Employee not exist with id :" + id));
65
66         employeeRepository.delete(employee);
67         Map<String, Boolean> response = new HashMap<>();
68         response.put("deleted", Boolean.TRUE);
69         return ResponseEntity.ok(response);
70     }
71
72 }

```

```

1 package com.suresh.demos.springboot.exception;
2
3 import org.springframework.http.HttpStatus;
4 import org.springframework.web.bind.annotation.ResponseStatus;
5
6 @ResponseStatus(value = HttpStatus.NOT_FOUND)
7 public class ResourceNotFoundException extends RuntimeException{
8
9     private static final long serialVersionUID = 1L;
10
11     public ResourceNotFoundException(String message) {
12         super(message);
13     }
14 }

```

Sample for NativeQuery

```
List<Employee> findByDesig(String desig);
```

```
@Query(value = "select * from employee where salary>=?",nativeQuery = true)  
List<Employee> findSalaryAbove(@Param("salary")int salary);
```

```
@GetMapping("/findEmployeesByDesig/{desig}")  
public List<Employee> findEmployeeByDesig(@PathVariable("desig") String desig) {  
    return employeeRepository.findByDesig(desig);  
}  
  
@GetMapping("/findEmployeeBySalaryRange")  
public List<Employee> findEmployeeBySalaryRange(@RequestParam("salary") int salary) {  
    return employeeRepository.findSalaryAbove(salary);  
}
```

Employee Management System – ReactJS

```
▼ EMS-CRUD
  > node_modules
  ▼ public
    <> index.html
    {} manifest.json
    ≡ robots.txt
  ▼ src
    ▼ components
      JS AddEmployeeComponent.js
      JS FooterComponent.js
      JS HeaderComponent.js
      JS ListEmployeeComponent.js
      JS ViewEmployeeComponent.js
    ▼ services
      JS EmployeeService.js
  # App.css
  JS App.js
  JS App.test.js
  # index.css
  JS index.js
  logo.svg
  JS reportWebVitals.js
  JS setupTests.js
  .eslintcache
  .gitignore
  {} package-lock.json
  {} package.json
  i README.md
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8" />
5    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6    <meta name="viewport" content="width=device-width, initial-scale=1" />
7    <meta name="theme-color" content="#000000" />
8    <meta
9      name="description"
10     content="Web site created using create-react-app"
11   />
12   <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
13   <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
14
15   <title>React App</title>
16 </head>
17 <body>
18   <noscript>You need to enable JavaScript to run this app.</noscript>
19   <div id="root"></div>
20 </body>
21 </html>
```


EMS

EMS-CRUD

- > node_modules
- ✓ public
 - <> index.html
 - { } manifest.json
 - ≡ robots.txt
- ✓ src
 - ✓ components
 - JS AddEmployeeComponent.js
 - JS FooterComponent.js
 - JS HeaderComponent.js
 - JS ListEmployeeComponent.js
 - JS ViewEmployeeComponent.js
 - ✓ services
 - JS EmployeeService.js
 - # App.css
 - JS App.js ✓
 - JS App.test.js
 - # index.css
 - JS index.js
 - 📁 logo.svg
 - JS reportWebVitals.js
 - JS setupTests.js
 - ≡ .eslintcache
 - 💎 .gitignore
 - { } package-lock.json
 - { } package.json
 - 📄 README.md

```
1  import './App.css';
2  import FooterComponent from './components/FooterComponent';
3  import HeaderComponent from './components/HeaderComponent';
4  import ListEmployeeComponent from './components/ListEmployeeComponent';
5  import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';
6  import AddEmployeeComponent from './components/AddEmployeeComponent';
7  import ViewEmployeeComponent from './components/ViewEmployeeComponent';
8
9  function App() {
10     return (
11         <div>
12             <Router>
13                 <HeaderComponent />
14                 <div class="container">
15                     <Switch>
16                         <Route path="/" exact component={ListEmployeeComponent} />
17                         <Route path="/employees" component={ListEmployeeComponent} />
18                         <Route path="/add-employee/:id" component={AddEmployeeComponent} />
19                         <Route path="/view-employee/:id" component={ViewEmployeeComponent} />
20                     </Switch>
21                 </div>
22                 <FooterComponent />
23             </Router>
24         </div>
25     );
26 }
27
28
29 export default App;
30 <ListEmployeeComponent />
```

EMS

▼ EMS-CRUD

➤ node_modules

▼ public

🔗 index.html

{ } manifest.json

☰ robots.txt

▼ src

▼ components

JS AddEmployeeComponent.js

JS FooterComponent.js ✓

JS HeaderComponent.js ✓

JS ListEmployeeComponent.js

JS ViewEmployeeComponent.js

▼ services

JS EmployeeService.js

App.css

JS App.js

JS App.test.js

index.css

JS index.js

🖼 logo.svg

JS reportWebVitals.js

JS setupTests.js

☰ .eslintcache

🔒 .gitignore

{ } package-lock.json

{ } package.json

📘 README.md

```
1  import React, { Component } from 'react'
2
3  export default class HeaderComponent extends Component {
4    render() {
5      return (
6        <div>
7          <header>
8            <nav className="navbar navbar-expand-md navbar-dark bg-dark">
9              <div>
10               <a href="http://vlaksush.in" className="navbar-brand" >Employee Management Application</a>
11             </div>
12           </nav>
13         </header>
14       </div>
15     )
16   }
17 }
18
```


```
1  import React, { Component } from 'react'
2
3  export default class FooterComponent extends Component {
4    render() {
5      return (
6        <div>
7          <footer className="footer">
8            <span className="text-muted">All rights reserved 2021 @vlaksush@gmail.com</span>
9          </footer>
10        </div>
11      )
12    }
13  }
14
```



```

1 import React, { Component } from 'react';
2 import EmployeeService from '../services/EmployeeService';
3 export default class ListEmployeeComponent extends Component {
4     constructor(props) {
5         super(props);
6         this.state = {
7             employees: []
8         }
9         this.addEmployee = this.addEmployee.bind(this);
10        this.editEmployee = this.editEmployee.bind(this);
11        this.viewEmployee = this.viewEmployee.bind(this);
12    }
13    componentDidMount() {
14        EmployeeService.getEmployees()
15            .then((res) => {
16                this.setState({
17                    employees: res.data
18                });
19            })
20    }
21    addEmployee() {
22        this.props.history.push('/add-employee/_add');
23    }
24    editEmployee(id) {
25        this.props.history.push(`/add-employee/${id}`);
26    }

```



```

v EMS-CRUD
  > node_modules
  v public
    <> index.html
    {} manifest.json
    ≡ robots.txt
  v src
    v components
      JS AddEmployeeComponent.js
      JS FooterComponent.js
      JS HeaderComponent.js
      JS ListEmployeeComponent.js
      JS ViewEmployeeComponent.js
    v services
      JS EmployeeService.js
  # App.css
  JS App.js
  JS App.test.js
  # index.css
  JS index.js
  logo.svg
  JS reportWebVitals.js
  JS setupTests.js
  .eslintcache
  .gitignore
  {} package-lock.json
  {} package.json
  README.md

```

```

27 deleteEmployee(id) {
28     EmployeeService.deleteEmployee(id)
29         .then((res) => {
30             this.setState({
31                 employees: this.state.employees
32                     .filter(employee => employee.id !== id)
33             });
34         })
35     }
36     viewEmployee(id) {
37         this.props.history.push(`/view-employee/${id}`);
38     }

```

```

39   render() {
40       return (
41           <div class="container">
42               <h2>Employee List</h2>
43               <div className="row">
44                   <button className="btn btn-primary" onClick={this.addEmployee}>
45                       Add Employee
46                   </button>
47               </div>
48               <br /><br />
49               <div className="row">
50                   <table className="table table-stiped table-bordered">
51                       <thead>
52                           <tr><th>Firstname</th><th>Lastname</th>
53                           <th>email</th> <th>Actions</th> </tr>
54                       </thead>
55                       <tbody>
56                           { this.state.employees.map(
57                               employee =>
58                                   <tr key={employee.id}>
59                                       <td>{employee.firstName}</td>
60                                       <td>{employee.lastName}</td>
61                                       <td>{employee.emailId}</td>
62                                       <td>
63                                           <button className="btn btn-info"
64                                               onClick={() => this.editEmployee(employee.id)}>Update</button>
65                                           <button style={{ marginLeft: "10px" }} onClick={() => this.deleteEmployee(employee.id)} className="btn btn-danger">Delete</button>
66                                           <button style={{ marginLeft: "10px" }} onClick={() => this.viewEmployee(employee.id)} className="btn btn-info">View</button>
67                                       </td>
68                                   </tr>
69                               ) }
70                       </tbody>
71                   </table>
72               </div>
73           </div>
74       )
75   }
76 }

```