DEPARTMENT OF CSE
COURSE CODE: 23SDCS12A / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/_____ **Time of The Session:_____to_____**

LAB - 9 ➔ Spring Boot with ReactJS Integration

Prerequisites:

Implementation skill in Reactjs

Implementation skill in Spring Boot Application with JPA and Database

Exercise:

Implement the necessary ReactJS front end pages for sending the request and receive responses from the back end (spring boot app) designed as per the below requirements.

Develop a Spring Boot web application to manage a list of products in a warehouse. The application should handle CRUD operations to manage product details such as Product ID, Name, Description, Price, and Quantity. The application should include features to add new products, display a list of all products, update existing product details, and delete products from the database. Use Spring Web MVC for handling HTTP requests, Spring Data JPA for database interactions. Ensure the application is configured to connect to a MySQL/PostgreSQL database and implement both setter-based or constructor-based dependency injections to manage service and repository layers effectively.

❖ Watch The Video And Do In Eclipse Workspace

9 https://youtu.be/cl9wbkNWqPs?si=wS_4-3fSaU4rjY1Q

Insert.jsx

```
import React from "react";
import axios from "axios";

export default function Insert() {
 function saveData() {
  var id = document.getElementsByName("idn")[0].value;
  var name = document.getElementsByName("namen")[0].value;

  axios
   .post("http://localhost:8081/product", {
     "id": parseInt(id),
     "name": name
     }).then((res)=>{

       alert(res.data)
     } )
    }

 return (
  <div>
   Id: <input type="text" name="idn" />
   <br />
   Name: <input type="text" name="namen" />
   <br />
   <button onClick={saveData}>Save</button>
  </div>
 );
}
```

Update.jsx

```
import React from "react";
import axios from "axios";

export default function Update() {
 function updateData() {
  var id = document.getElementsByName("idn")[0].value;
  var name = document.getElementsByName("namen")[0].value;

  axios
   .put("http://localhost:8081/product", {
    "id": parseInt(id),
    "name": name
    }).then((res)=>{

      alert(res.data)
    } )
   }

 return (
  <div>
   Id: <input type="text" name="idn" />
   <br />
   Name: <input type="text" name="namen" />
   <br />
   <button onClick={updateData}>Save</button>
  </div>
 );
}
```

Delete.jsx

```
import React from "react";
import axios from "axios";

export default function Insert() {
 function DeleteData() {
  var id = document.getElementsByName("idn")[0].value;

  axios
   .delete(`http://localhost:8081/product/${id}`)
   .then((res) => {
    alert(res.data);
   });
 }

 return (
  <div>
   Id: <input type="text" name="idn" />
   <br />
   <button onClick={DeleteData}>Delete</button>
  </div>
 );
}
```

Show.jsx

```
import React, { useState, useEffect } from "react";
import axios from "axios";

export default function Show() {
  const [result, setResult] = useState(null);

  useEffect(() => {
    axios.get("http://localhost:8081/product")
      .then((res) => {
        setResult(res.data);
      })
      .catch((err) => console.error("Error fetching data:", err));
  }, []);

  if (result === null) {
    return <div>Data is Fetching</div>;
  }

  return (
    <table border="1">
      <thead>
        <tr>
          <th>ID</th>
          <th>Name</th>
        </tr>
      </thead>
      <tbody>
        {result.map((element, index) => (
          <tr key={index}>
            <td>{element.id}</td>
            <td>{element.name}</td>
          </tr>
        ))}
      </tbody>
    </table>
  );
}
```

VIVA QUESTIONS:

1. How does data flow between a ReactJS frontend and a Spring Boot backend in a full-stack application?
2. How do you configure CORS in a Spring Boot application to allow requests from a ReactJS frontend?
3. What is the role of JPA in a Spring Boot application, and how does it interact with a database?
4. How do you handle asynchronous operations in React when fetching data from the Spring Boot API?
5. Can you describe a typical CRUD operation cycle from the ReactJS frontend to the database via Spring Boot and JPA?

1. React sends API requests → Spring Boot processes → JPA interacts with DB → Response to React.
2. Use `@CrossOrigin` or configure `CorsFilter` in `WebMvcConfigurer`.
3. ORM for DB operations, `JpaRepository` handles CRUD.
4. Use `useEffect`, `fetch/axios`, and `useState` for state management.
5. **CRUD:**
   - `POST` → Save via JPA.
   - `GET` → Fetch data.
   - `PUT` → Modify DB.
   - `DELETE` → Remove entry.

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured _____out of 50 Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation: