HANDS ON EXERCISE WEEK-7

**1. How can we fetch and display a list of users from a Spring Boot backend using React?**

**Backend: Spring Boot (Java)**

java

// User.java (Model)

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String email;

// getters and setters

}

java

// UserRepository.java

public interface UserRepository extends JpaRepository<User, Long> {}

java

// UserController.java

@RestController

@RequestMapping("/api/users")

@CrossOrigin(origins = "http://localhost:3000")

public class UserController {

@Autowired

private UserRepository repo;

@GetMapping

public List<User> getAllUsers() {

return repo.findAll();

}

}

**Frontend: React**

jsx

// UserList.js

import React, { useEffect, useState } from 'react';

import axios from 'axios';

function UserList() {

const [users, setUsers] = useState([]);

useEffect(() => {

axios.get("http://localhost:8080/api/users")

.then(response => setUsers(response.data))

.catch(error => console.error("Error fetching users:", error));

}, []);

return (

<div>

<h2>User List</h2>

<ul>

{users.map(user =>

<li key={user.id}>{user.name} ({user.email})</li>

)}

</ul>

</div>

);

}

export default UserList;

**2. How to submit a form from React and save the data in a MySQL DB using Spring Boot?**

**Backend (Spring Boot)**

java

@PostMapping

public User addUser(@RequestBody User user) {

return repo.save(user);

}

**Frontend (React Form)**

jsx

// AddUser.js

import React, { useState } from 'react';

import axios from 'axios';

function AddUser() {

const [user, setUser] = useState({ name: "", email: "" });

const handleChange = e => {

setUser({ ...user, [e.target.name]: e.target.value });

};

const handleSubmit = e => {

e.preventDefault();

axios.post("http://localhost:8080/api/users", user)

.then(res => alert("User added!"))

.catch(err => console.error(err));

};

return (

<form onSubmit={handleSubmit}>

<input name="name" onChange={handleChange} placeholder="Name" />

<input name="email" onChange={handleChange} placeholder="Email" />

<button type="submit">Add</button>

</form>

);

}

export default AddUser;

**3. How to update a user using React form and PUT API in Spring Boot?**

**Backend (Spring Boot)**

java

@PutMapping("/{id}")

public ResponseEntity<User> updateUser(@PathVariable Long id, @RequestBody User updatedUser) {

return repo.findById(id).map(user -> {

user.setName(updatedUser.getName());

user.setEmail(updatedUser.getEmail());

repo.save(user);

return ResponseEntity.ok(user);

}).orElse(ResponseEntity.notFound().build());

}

**Frontend (React Update Form)**

jsx

// UpdateUser.js

import React, { useState, useEffect } from 'react';

import axios from 'axios';

function UpdateUser({ userId }) {

const [user, setUser] = useState({ name: "", email: "" });

useEffect(() => {

axios.get(`http://localhost:8080/api/users/${userId}`)

.then(res => setUser(res.data));

}, [userId]);

const handleChange = e => {

setUser({ ...user, [e.target.name]: e.target.value });

};

const handleSubmit = e => {

e.preventDefault();

axios.put(`http://localhost:8080/api/users/${userId}`, user)

.then(() => alert("User updated"));

};

return (

<form onSubmit={handleSubmit}>

<input name="name" value={user.name} onChange={handleChange} />

<input name="email" value={user.email} onChange={handleChange} />

<button type="submit">Update</button>

</form>

);

}

export default UpdateUser;

**4.How can we delete a user from a database using React and a Spring Boot backend?**

**Backend (Spring Boot - DELETE API)**

java

@DeleteMapping("/{id}")

public ResponseEntity<Void> deleteUser(@PathVariable Long id) {

if (repo.existsById(id)) {

repo.deleteById(id);

return ResponseEntity.ok().build();

} else {

return ResponseEntity.notFound().build();

}

}

**Frontend (React)**

jsx

// UserListWithDelete.js

import React, { useEffect, useState } from 'react';

import axios from 'axios';

function UserListWithDelete() {

const [users, setUsers] = useState([]);

const fetchUsers = () => {

axios.get("http://localhost:8080/api/users")

.then(res => setUsers(res.data));

};

const deleteUser = (id) => {

axios.delete(`http://localhost:8080/api/users/${id}`)

.then(() => fetchUsers());

};

useEffect(fetchUsers, []);

return (

<div>

<h2>User List with Delete</h2>

<ul>

{users.map(user => (

<li key={user.id}>

{user.name} - {user.email}

<button onClick={() => deleteUser(user.id)}>Delete</button>

</li>

))}

</ul>

</div>

);

}

export default UserListWithDelete;

**5. How can we search for users by name using a search bar in React with a backend filter?**

**Backend: Search Endpoint**

java

@GetMapping("/search")

public List<User> searchUsers(@RequestParam String keyword) {

return repo.findByNameContainingIgnoreCase(keyword);

}

java

// In UserRepository.java

List<User> findByNameContainingIgnoreCase(String keyword);

**Frontend: React Search Bar**

jsx

// UserSearch.js

import React, { useState } from 'react';

import axios from 'axios';

function UserSearch() {

const [keyword, setKeyword] = useState("");

const [results, setResults] = useState([]);

const search = () => {

axios.get(`http://localhost:8080/api/users/search?keyword=${keyword}`)

.then(res => setResults(res.data));

};

return (

<div>

<input value={keyword} onChange={e => setKeyword(e.target.value)} placeholder="Search name..." />

<button onClick={search}>Search</button>

<ul>

{results.map(u => <li key={u.id}>{u.name} - {u.email}</li>)}

</ul>

</div>

);

}

export default UserSearch;

**6. How can you implement pagination in a React table using a Spring Boot paginated API?**

**Backend: Spring Pagination**

java

@GetMapping("/paged")

public Page<User> getPagedUsers(@RequestParam int page, @RequestParam int size) {

return repo.findAll(PageRequest.of(page, size));

}

**Frontend: React Pagination**

jsx

// PagedUserList.js

import React, { useState, useEffect } from 'react';

import axios from 'axios';

function PagedUserList() {

const [users, setUsers] = useState([]);

const [page, setPage] = useState(0);

const fetchUsers = () => {

axios.get(`http://localhost:8080/api/users/paged?page=${page}&size=5`)

.then(res => setUsers(res.data.content));

};

useEffect(fetchUsers, [page]);

return (

<div>

<h2>Paged User List</h2>

<ul>

{users.map(user => <li key={user.id}>{user.name} - {user.email}</li>)}

</ul>

<button disabled={page === 0} onClick={() => setPage(p => p - 1)}>Previous</button>

<button onClick={() => setPage(p => p + 1)}>Next</button>

</div>

);

}

export default PagedUserList;