

# Employee Management System

---

**SPRING BOOT + JPA + H2 + REACT IMPLEMENTATION**

RAKSHA PAHARIYA

# AGENDA

---

- Project Overview
- Learning Objectives
- JPA & H2 Database
- CRUD Operations
- REST vs SOAP APIs
- Results & Demo Output

# PROJECT OVERVIEW

---

- **Purpose:**
  - Build a working project to explore Spring ecosystem.
- **Learning Goals:**
  - Understand Spring Boot project structure.
  - Practice REST API development.
  - Apply JPA for database mapping.
  - Use H2 DB for testing CRUD features.
  - React UI integration with backend APIs.
- **Scope:**
  - Small HR-style application managing employee data.

# JPA & H2 DATABASE

---

## JPA (Java Persistence API):

- Maps Java classes → DB tables.
- Eliminates raw SQL queries.
- Works with Hibernate under the hood.

## H2 Database:

- In-memory DB, resets each run.
- Auto-configured with Spring Boot.
- Ideal for learning & testing without external setup.

## Key Relationship Between Them

- JPA defines how to interact with data (objects → tables).
- H2 acts as the storage engine where the data is actually stored.
- Together, they allow rapid development and testing without complex setup.

# CRUD OPERATIONS

---

- **Create (POST):**
  - Add new employee.
- **Read (GET):**
  - Fetch all employees or one by ID.
- **Update (PUT):**
  - Modify existing employee details.
- **Delete (DELETE):**
  - Remove an employee record.
- **Implementation:**
  - Controller: Handles REST endpoints.
  - Service: Business logic.
  - Repository: DB operations via JPA.

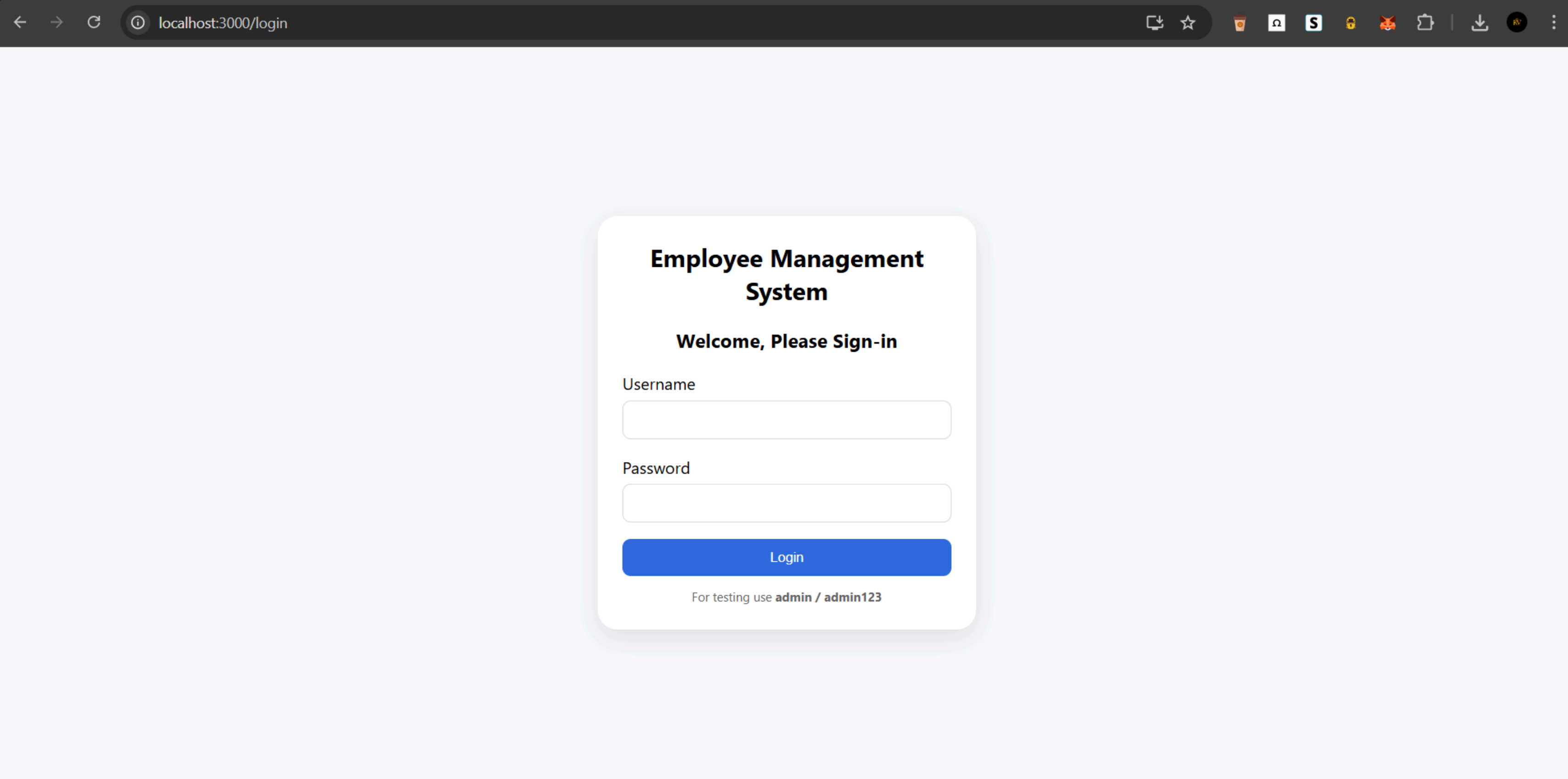
# REST VS SOAP API

---

- **REST (Implemented):**
  - Simple, lightweight, JSON-based.
  - Follows HTTP verbs (GET, POST, PUT, DELETE).
  - Easy to integrate & modern standard.
- **SOAP (For comparison):**
  - XML-based, uses WSDL.
  - Heavier but more secure.
  - Suitable for enterprise contracts

**D E M O**

# LOGIN SCREEN



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/login'. The browser's toolbar includes navigation icons (back, forward, refresh) and a series of extension icons. The main content area features a light blue background with a white, rounded rectangular login card centered on the screen. The card has a subtle drop shadow. Inside the card, the text 'Employee Management System' is displayed in a bold, black font. Below this, a welcome message 'Welcome, Please Sign-in' is shown. There are two input fields: one for 'Username' and one for 'Password'. A prominent blue button with the text 'Login' is positioned below the password field. At the bottom of the card, a note states 'For testing use admin / admin123'.

← → ↻ ⓘ localhost:3000/login

📄 ☆ 🍷 Ω S 🔒 🦊 🛒 | ⬇️ ⬛ RV ⋮

# Employee Management System

Welcome, Please Sign-in

Username

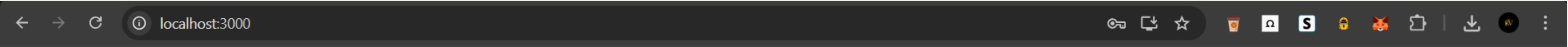
Password

Login

For testing use **admin / admin123**



# DASHBOARD




## Employee Dashboard

 Logout

### Current Employees


+ Add Employee

 **Employee One**  
emp.one@wf.com



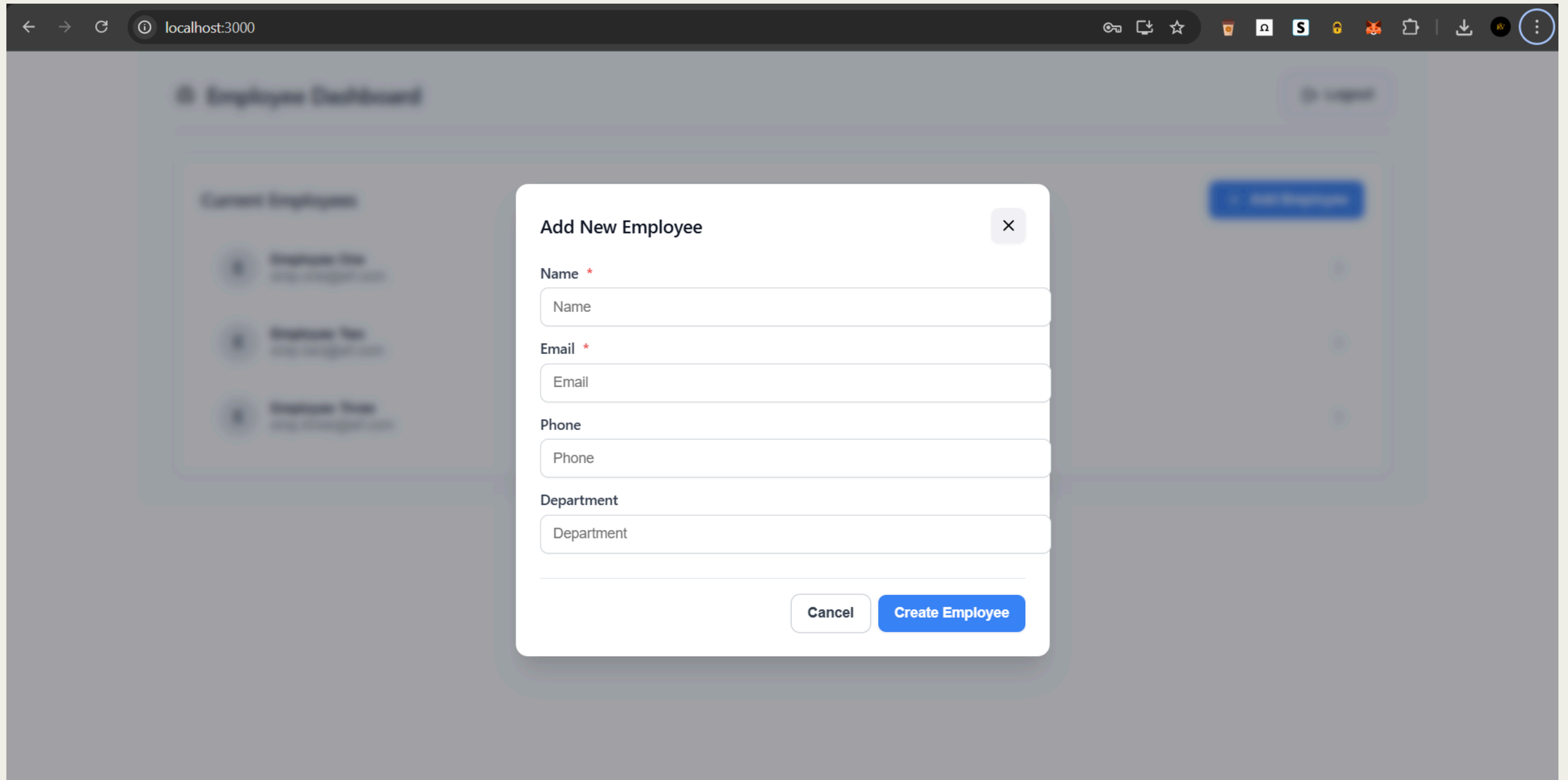
 **Employee Two**  
emp.two@wf.com



 **Employee Three**  
emp.three@wf.com



# ADD EMPLOYEE



# UPDATE/DELETE EMPLOYEE

← → ↻ ⓘ localhost:3000 🔑 📄 ☆ 🍷 📧 S 🔒 🦊 📁 | ⬇️ 🌙 ⋮

## Employee Dashboard

### Current Employees

Employee One

Employee Two

Employee Three

➕ Add Employee

### Edit Employee Details

✕

Name \*

Employee One

Email \*

emp.one@wf.com

Phone

987654321

Department

CCIBT

🗑️ Delete

Cancel

Save Changes



# RESULTS & CONCLUSION

---

## Results:

- Full-stack CRUD app with React frontend & Spring backend.
- Tested via React UI + Postman + H2 Console.
- Learnt how to connect frontend ↔ backend ↔ database.

## Key Takeaways:

- Gained practical knowledge of Spring Boot, REST, JPA, H2, and React.
- Built layered architecture with real UI integration.

# Thank you!

---

RAKSHA PAHARIYA