POC

Project Overview

You will develop a simple "Student Management System" which allows a user to:

- Create a new student entry
- Read all student entries
- Update a student's information
- Delete a student entry

Tech Stack

Frontend:

- Angular (v15+)
- Bootstrap (for UI styling)

Backend:

- Java (JDK 21)
- Spring Boot (v3+)
- Maven
- H2 in-memory database

Deliverables

- Functional Angular + Spring Boot CRUD project

Timeline & Tasks

Day 1: Environment Setup & Project Initialization

Backend: Install Java & Spring Boot. Initialize Spring Boot project with:

- Spring Web
- Spring Data JPA
- H2 Database

Frontend: Install Node.js and Angular CLI. Create Angular project. Create basic UI with navigation bar

Tasks:

- Setup backend project structure
- Setup Angular app and basic routes/components

Day 2: Backend – Create & Read APIs

Create Student entity with fields: id, name, email, department

Implement:

- POST /students – Create

- GET /students – Read all
- Use H2 DB with schema.sql & data.sql (optional)

Tasks:
• Create Student entity, repository, service, controller
• Test APIs using Postman

Day 3: Frontend – Connect Angular to Backend (Read & Create)

Use Angular service to consume backend APIs

Create form to add a new student

Display students in a table

Tasks:
• Create Angular service using HttpClient
• Use Reactive or Template Driven Forms
• Display data in HTML table

Day 4: Implement Update & Delete Operations

Backend:
- PUT /students/{id} – Update
- DELETE /students/{id} – Delete

Frontend:
- Add buttons for Edit and Delete
- Populate form for update
- Delete row on button click

Tasks:
• Implement update form logic
• Delete student and refresh list

Day 5: Final Testing, Cleanup & Documentation

•Use pagination in the student list
Add search or filter functionality
Test all CRUD operations

UI polishing with Bootstrap

Add validations (required fields, email format)

Tasks:
• Final test end-to-end

Sample API Definitions

Method	URL	Description
POST	/students	Add new student
GET	/students	Get all students
PUT	/students/{id}	Update student
DELETE	/students/{id}	Delete student

Learning Objectives

- How to set up full-stack applications
- RESTful API development using Spring Boot
- HTTP communication between Angular and backend
- Implementing CTUD operations
- Using H2 for in-memory storage