LAB TASK 6

SUFYAN HUMMAM MUSHTAQ

SOFTWARE CONSTRUCTION

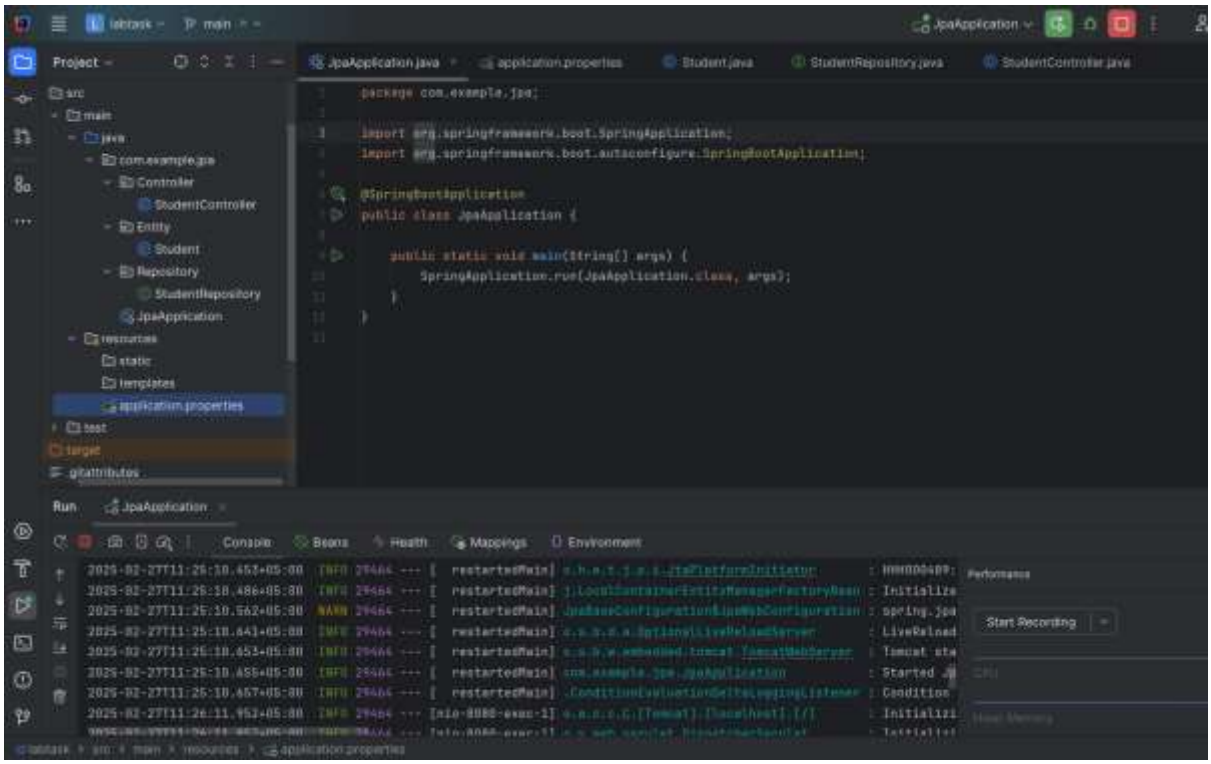
SOFTWARE ENGINEERING GREEN

SUBMITTED TO: MAM ZAINAB TAHIR

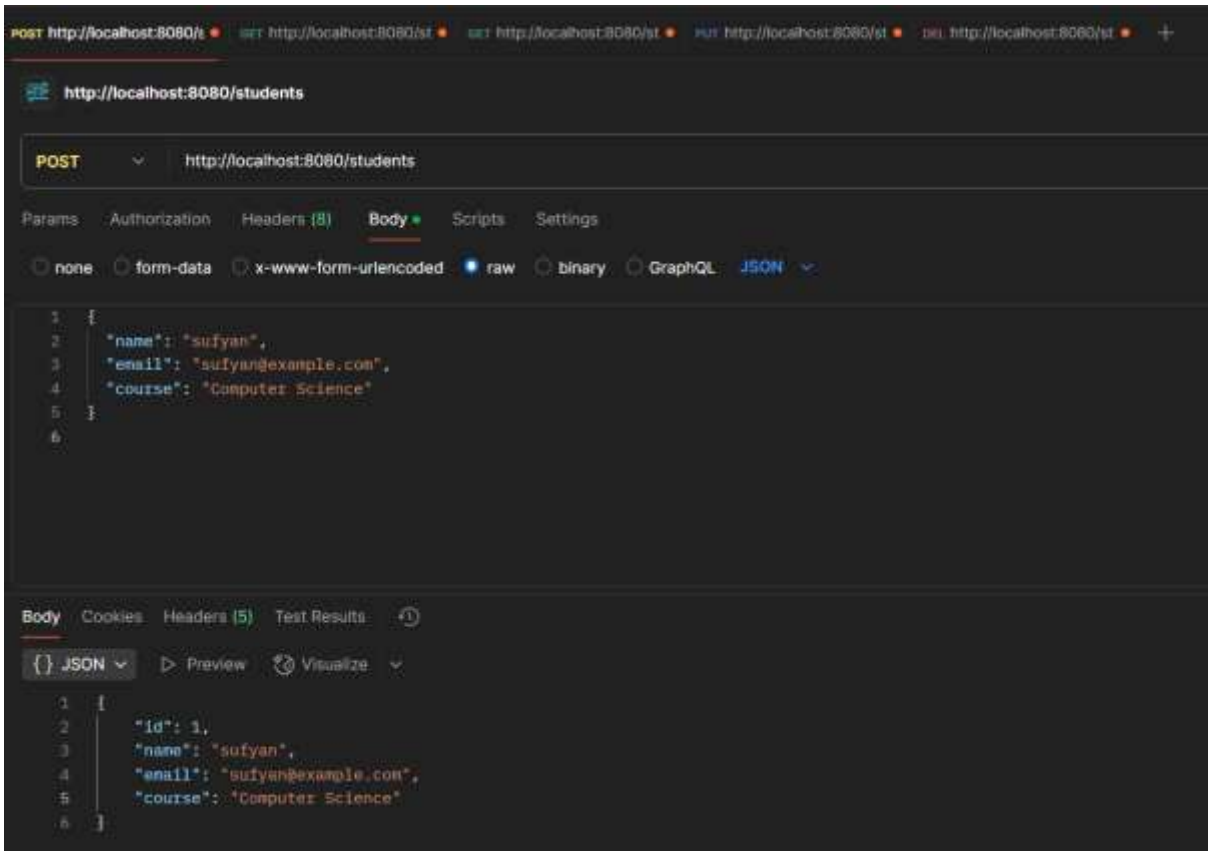
DATE: 27 FEB, 2025

GitHub Repository: https://github.com/sufyan-hummam/SC-D-LAB_6

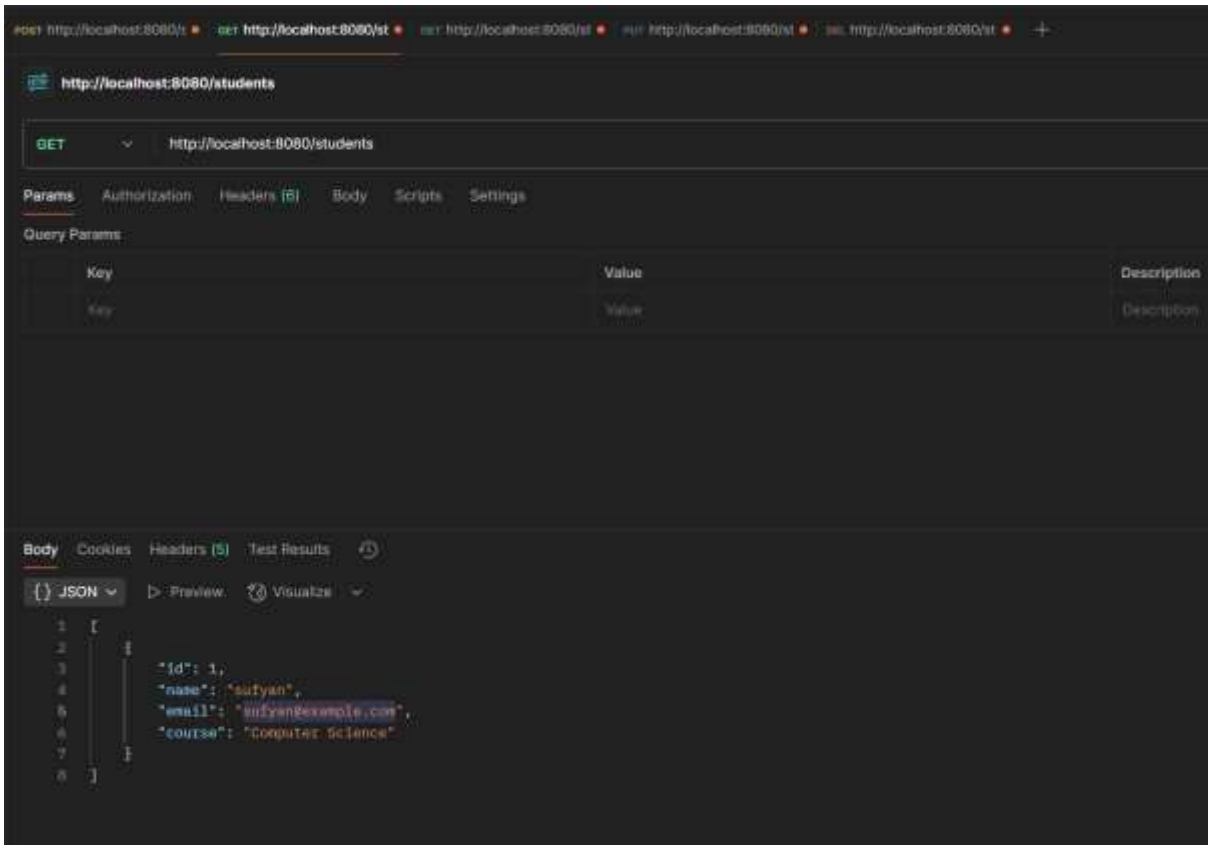
CODE SETUP:



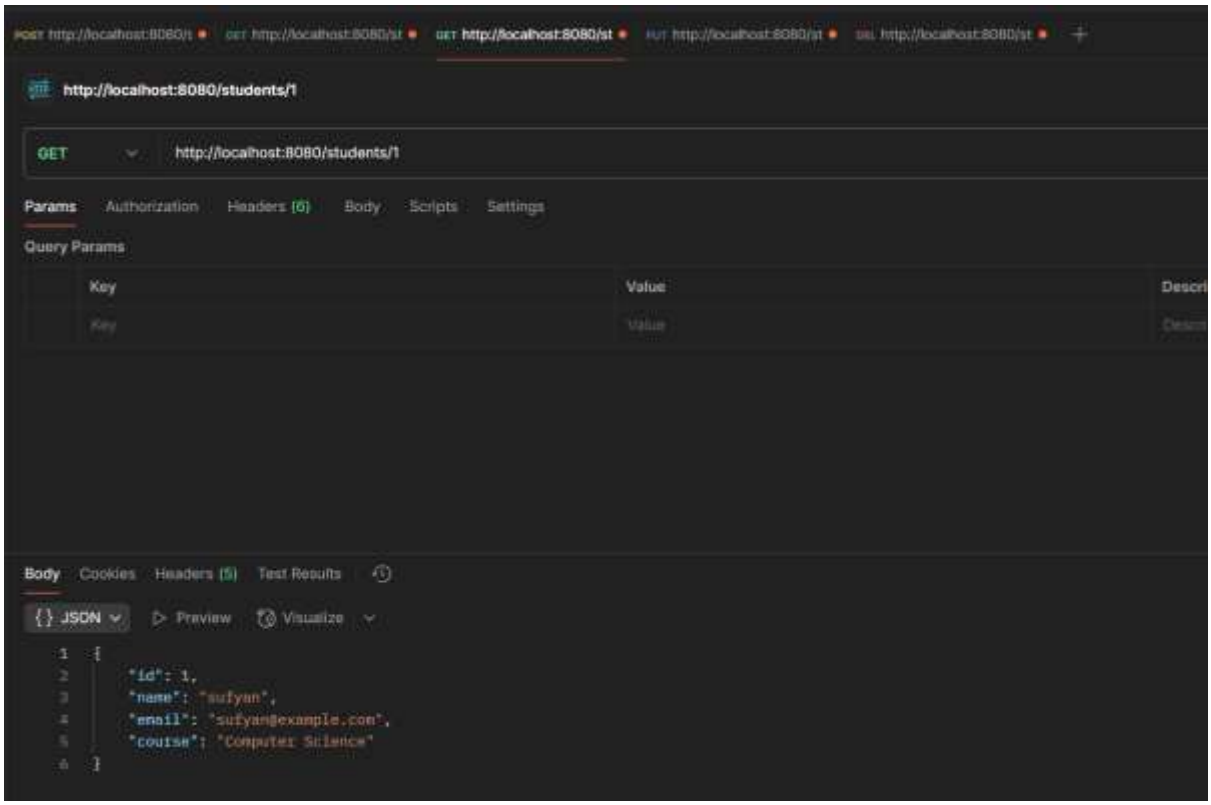
GET METHOD:


http://localhost:8080/students
POST
http://localhost:8080/students
Params
Authorization
Headers (8)
Body
Scripts
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"name": "sufyan",
"email": "sufyan@example.com",
"course": "Computer Science"
}
Body
Cookies
Headers (5)
Test Results
{} JSON
Preview
Visualize
[
"id": 1,
"name": "sufyan",
"email": "sufyan@example.com",
"course": "Computer Science"
]

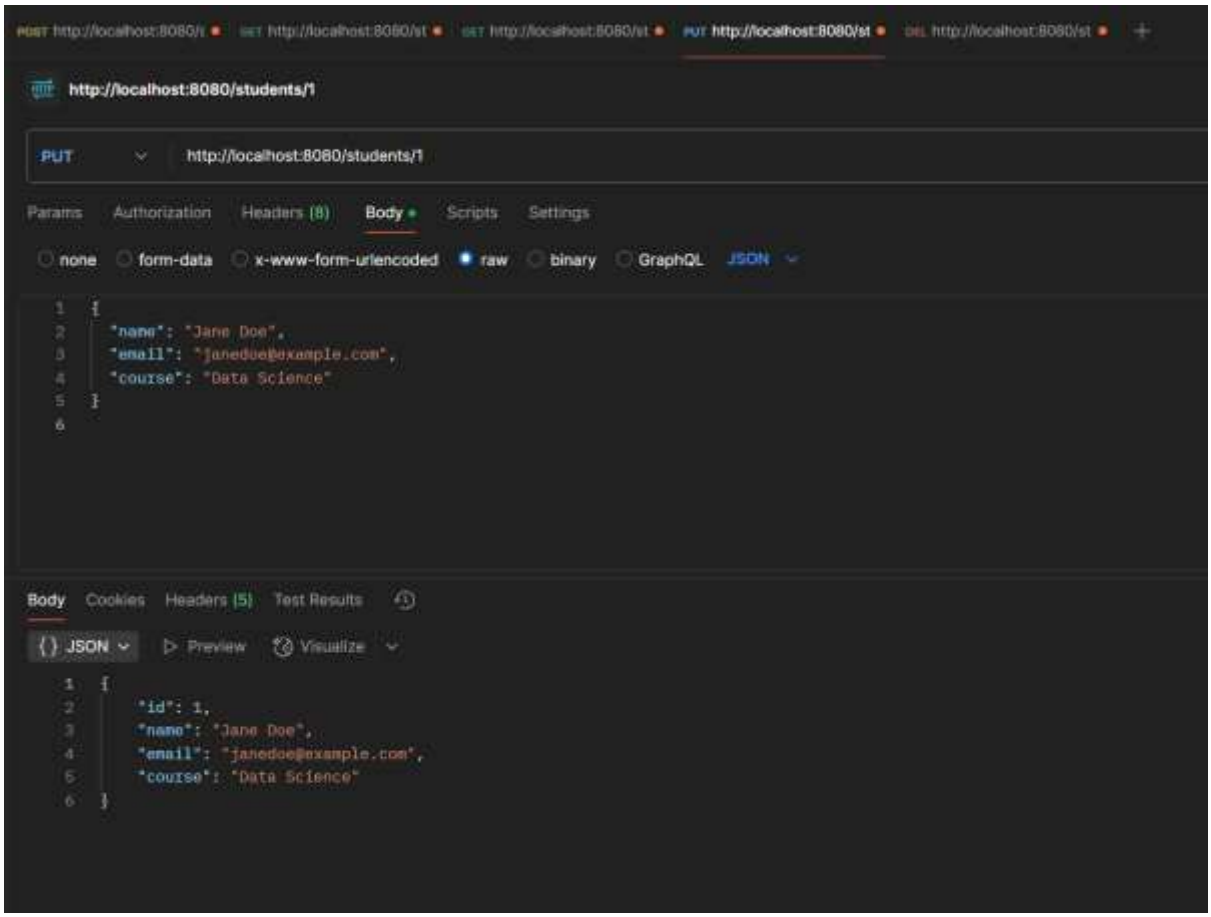
GET METHOD:


http://localhost:8080/students
GET
http://localhost:8080/students
Params
Authorization
Headers (6)
Body
Scripts
Settings
Query Params
Key
Value
Description
Body
Cookies
Headers (5)
Test Results
JSON
Preview
Visualize
[
{
"id": 1,
"name": "sufyan",
"email": "sufyan@example.com",
"course": "Computer Science"
}
]

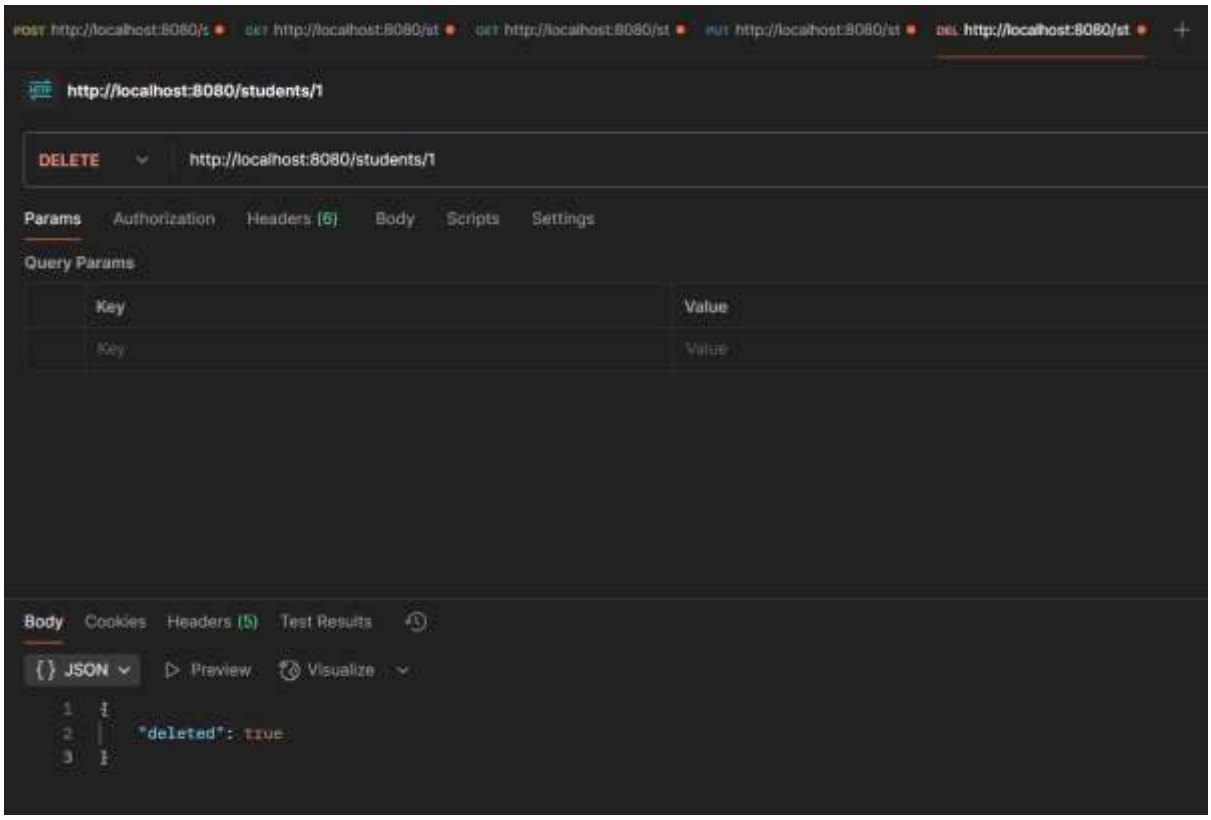
GET METHOD BY ID:



PUT METHOD:


http://localhost:8080/students/1
PUT
http://localhost:8080/students/1
Params
Authorization
Headers (8)
Body
Scripts
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"name": "Jane Doe",
"email": "janedoe@example.com",
"course": "Data Science"
}
Body
Cookies
Headers (5)
Test Results
JSON
Preview
Visualize
{
"id": 1,
"name": "Jane Doe",
"email": "janedoe@example.com",
"course": "Data Science"
}

DELETE METHOD:



Summary of Student Management System API

The Student Management System API is a RESTful web service developed using Spring Boot, Java, and MySQL. It provides basic CRUD (Create, Read, Update, Delete) functionalities to manage student records in a database.

Project Structure

The project follows a layered architecture:

CONTROLLER LAYER: Handles HTTP requests and responses.

SERVICE LAYER: Manages business logic (optional but recommended).

REPOSITORY LAYER: Connects with MySQL using Spring Data JPA.

ENTITY LAYER: Defines the Student model (table).

Database Configuration

The application uses MySQL for data storage, and the database connection is configured in the application.properties file. The Spring Boot Hibernate dialect is used to manage database schema updates automatically.

API Functionalities

The API provides the following endpoints, accessible via Postman:

POST /students: Add a new student.

GET /students: Retrieve all students.

GET /students/1: Retrieve a student by ID.

PUT /students/1 Update student details.

DELETE /students/1 Remove a student from the database.

Working Process

MySQL Database Setup: The database (student_db) is created using XAMPP or MySQL Workbench.

Running the Application: The Spring Boot application is started from IntelliJ IDEA or VS Code.

API Testing in Postman: Different endpoints are tested using JSON requests.

Conclusion

The Student Management System API is a simple yet efficient Spring Boot application that demonstrates CRUD operations using REST principles. It follows a structured architecture, making it easy to maintain and extend. The use of Postman ensures that all API functions work correctly before deployment.