**DOCUMENTATION FOR CRUD OPERATION PROJECT**

**INTRODUCTION :**

A CRUD (Create, Read, Update, Delete) application is a fundamental component of web development, and in this context, we'll explore how to build one using Spring Tool Suite (STS), MySQL as the database, and Postman for testing API endpoints.

Introduction to the Components:Spring Tool Suite (STS): STS is an integrated development environment (IDE) based on Eclipse that simplifies the development of Spring-based applications. It provides tools and templates for building robust Java applications, including Spring Boot projects.

MySQL: MySQL is a popular open-source relational database management system. In our CRUD application, we'll use MySQL to store and manage data.Postman: Postman is a powerful API testing tool that allows you to send HTTP requests to your application's endpoints and receive responses. It's an essential tool for testing and debugging RESTful APIs.

Building the CRUD Application:To create a CRUD application using these components, follow these general steps:

Set Up Your Development Environment:Install Spring Tool Suite (STS) and MySQL on your system.

Create a new Spring Boot project in STS.Database Configuration:Configure the MySQL database connection in your Spring Boot application's properties or YAML file.

Create Entity Classes: Define Java entity classes that represent the data you want to store in the database. Annotate them with JPA annotations. Repository Interfaces: Create repository interfaces using Spring Data JPA to perform database operations (CRUD) on your entity classes.

Service Layer: Create service classes to encapsulate the business logic of your application. These classes will interact with the repository interfaces.

Controller Layer: Develop RESTful controllers that handle HTTP requests and use the service layer to perform CRUD operations.

Testing with Postman:Use Postman to send HTTP requests (GET, POST, PUT, DELETE) to your API endpoints (e.g., /api/users).Verify that you can create, read, update, and delete data through these endpoints.

SETTING OF THE DEVELOPMENT ENVIRONMENT:

To set up a development environment for a CRUD (Create, Read, Update, Delete) application using Spring Tool Suite, MySQL, and Postman, you can follow these general steps:

1. \*Install Required Software\*:

- Install Spring Tool Suite (STS) if you haven't already.

- Install MySQL Server and MySQL Workbench or any MySQL client of your choice.

- Install Postman for testing your API endpoints.

2. \*Create a Spring Boot Project\*:

- Open STS and create a new Spring Boot project. You can use the Spring Initializer wizard within STS to generate a basic project structure.

3. \*Set Up Database\*:

- Configure your MySQL database connection in your Spring Boot application properties (usually `application.properties` or `application.yml`).

properties

spring.datasource.url=jdbc:mysql://localhost:3306/your\_database

spring.datasource.username=your\_username

spring.datasource.password=your\_password

4. \*Create Entity Classes\*:

- Create Java classes representing your data entities (e.g., User, Product) with JPA annotations.

5. \*Create Repository Interfaces\*:

- Create repository interfaces that extend JpaRepository to handle database operations.

6. \*Implement Service Layer\*:

- Create service classes to contain business logic and interact with repositories.

7. \*Create RESTful Controllers\*:

- Create controllers with Spring MVC to expose RESTful endpoints.

8. \*Test Using Postman\*:

- Build and run your Spring Boot application.

- Use Postman to send HTTP requests (GET, POST, PUT, DELETE) to your API endpoints to test CRUD operations.

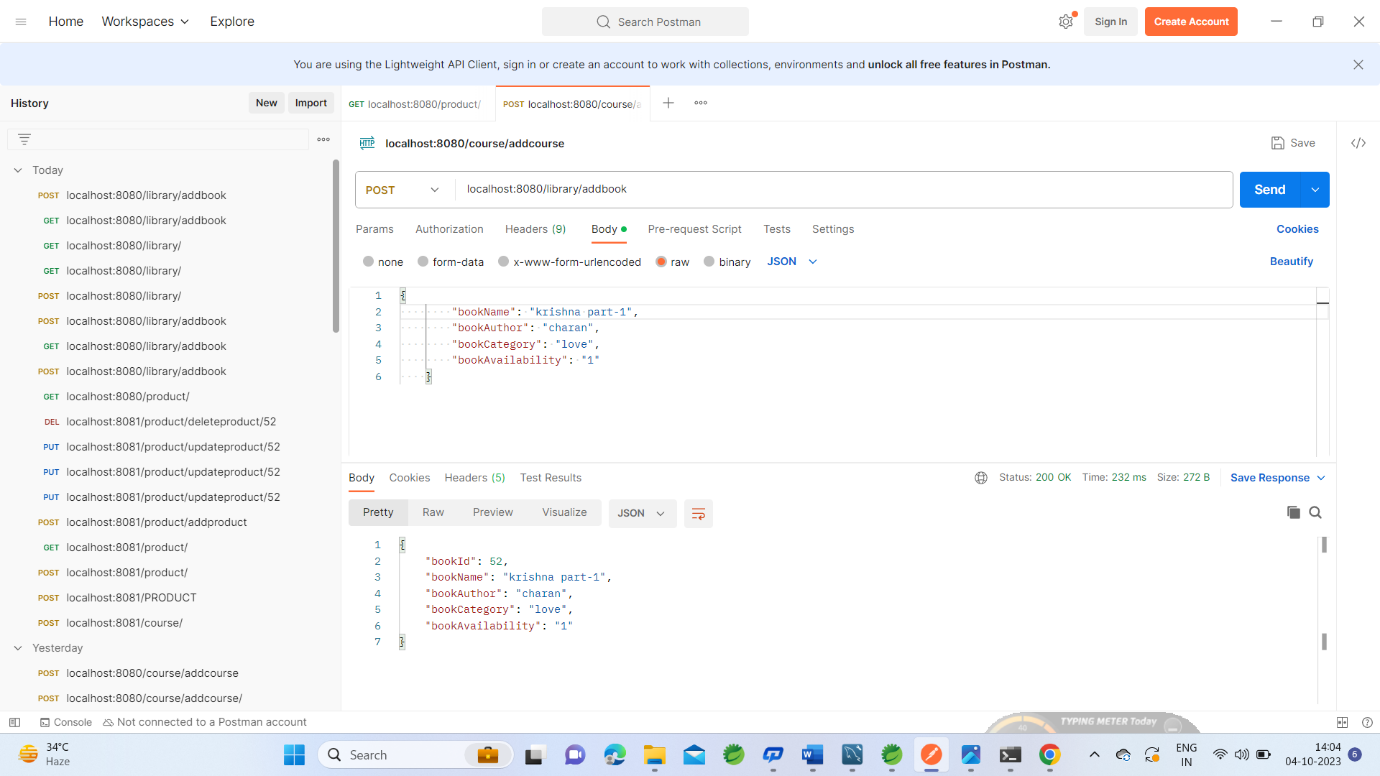
Remember to adapt these steps according to your specific project requirements and architecture. This is a high-level overview, and the actual implementation details may vary depending on your project's complexity and needs.

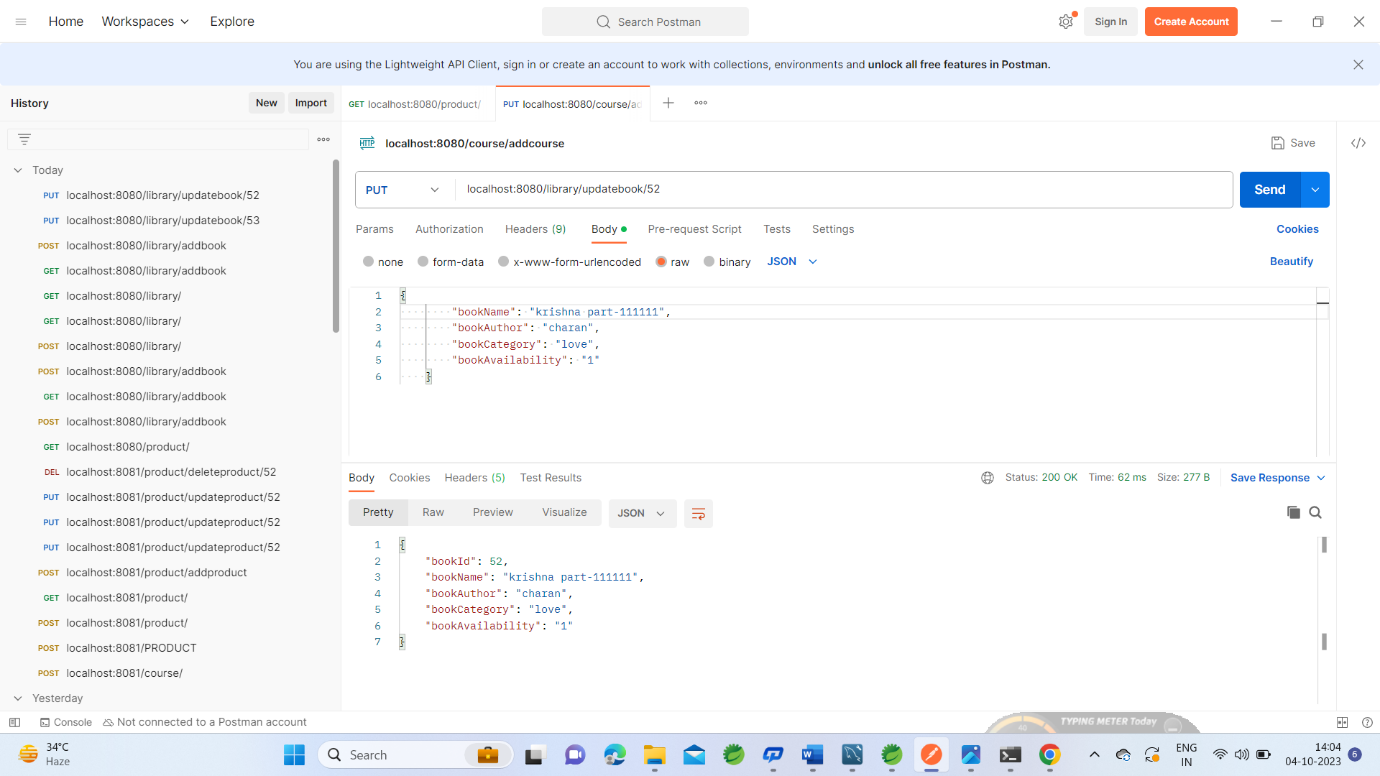
PROJECTPROCEDURE :

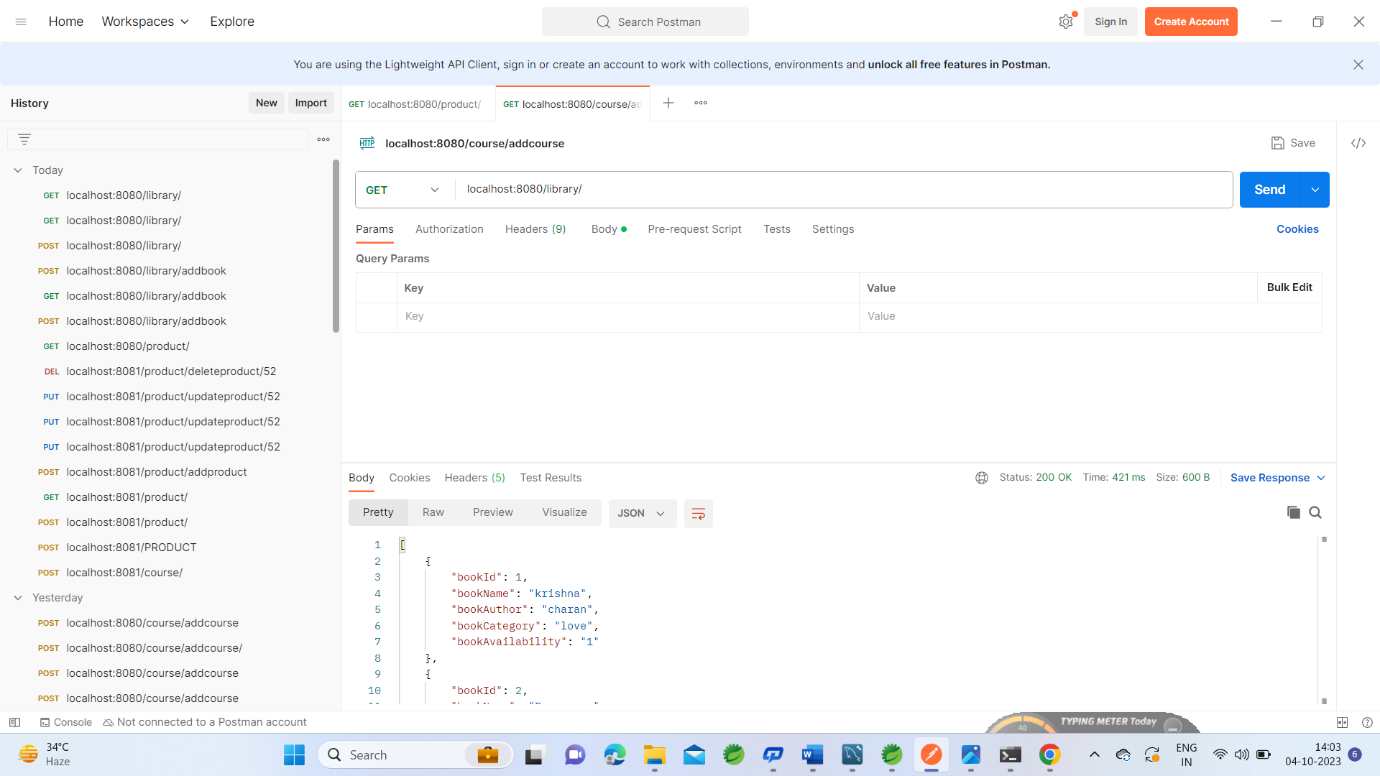
1. We need to downloaded the required software into machines.
2. Then open the Spring Tool Suite for Developing the business logic using the Spring Boot
3. You need to create the project in STS and you need add all header files And developer the code.
4. We create the project with layer like Entites layer, Service layer, Repository layer, Controller layer
5. Entites layer have the Variable Declaration and Lombok (toString , Setter , Getter ,AllArgsConstructor ,NoArgsConstructor) to avoid bowl code.
6. Service layer we have the develop the business logic code
7. Repository layer we have create the JPA code.
8. Controller layer we have develop the Controllers of the Http controlers.
9. Application.properites we have linked JDBC connectivity.

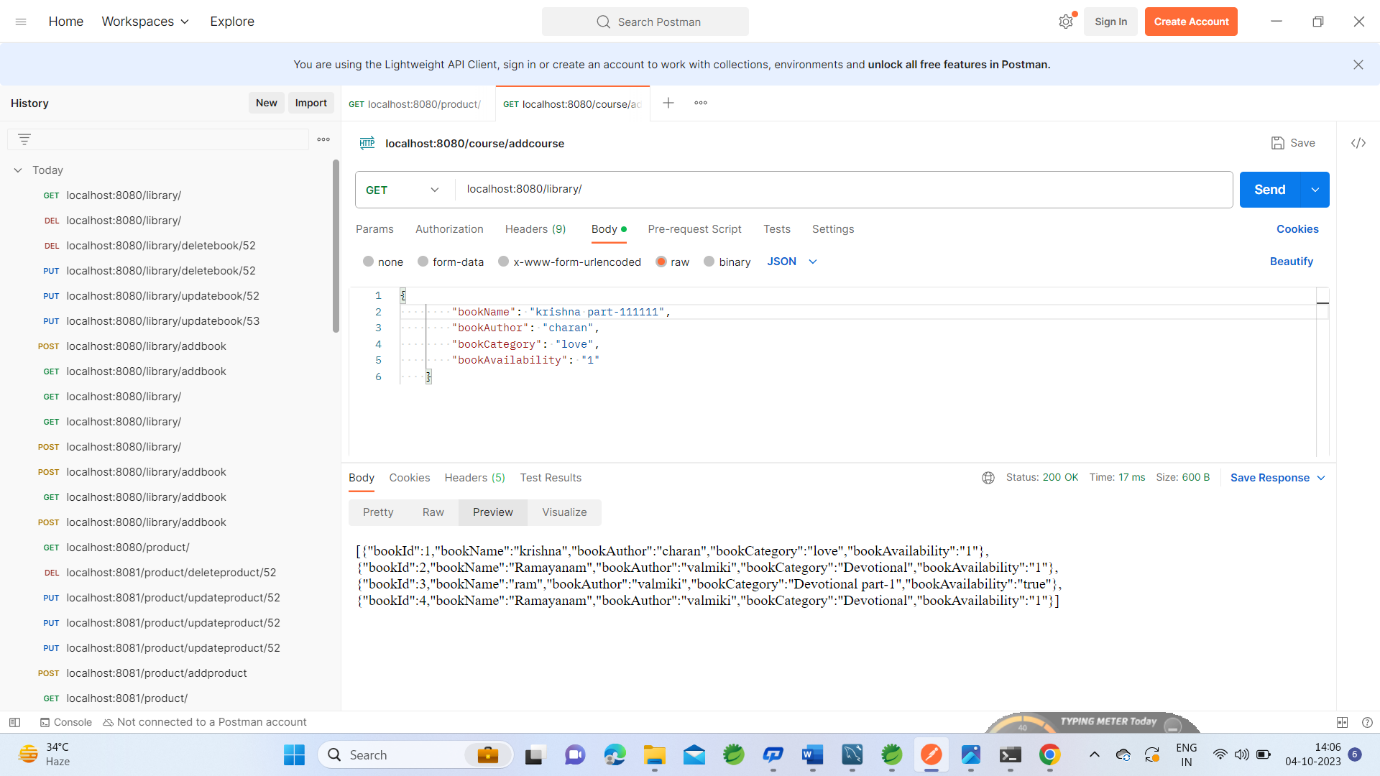
Screenshot:

Adding the book



Update the book

View the data 

View all the books

Delete the book