**DOCUMENTATION FOR CRUD APPLICATION**

**INTRODUCTION:**

The CRUD (Create, Read, Update, Delete) application built with Spring Boot, SpringToolSuite, and MySQL serves the purpose of providing a robust and user-friendly system for managing data. Its scope encompasses:

1. Data Management: Allowing users to create, retrieve, update, and delete records in a MySQL database.

2. Technological Stack: Leveraging STS, and MySQL as the relational database management system.

3. Scalability: Designed to be scalable, the application can handle various types of data and grow to accommodate evolving needs.

4. Customization: Potential for customization and extension to suit specific business requirements or use cases.

Overall, the CRUD application streamlines data management operations, making it an essential tool for businesses and organizations to efficiently handle their data-related tasks.

**SETTING UP THE DEVELOPMENT ENVIRONMENT**

To set up the development environment for the CRUD application using Spring Tool Suite (STS), follow these instructions, including the necessary prerequisites:

Prerequisites:

1. Java Development Kit (JDK):

- Ensure you have the JDK (Java Development Kit) installed. The application is built with Java, and you'll need this to compile and run the code. You can download the JDK from the official Oracle website or use OpenJDK.

2. Spring Tool Suite (STS):

- Download and install Spring Tool Suite (STS) from the official website (https://spring.io/tools).

3. MySQL Database:

- Install MySQL if not already installed. You can download the MySQL Community Server from the official MySQL website (https://dev.mysql.com/downloads/installer/).

**Setting Up the Development Environment:**

1. Clone the Project:

- Clone the CRUD application project repository from your version control system (e.g., Git) to your local machine.

2. Importing the Project into STS:

- Open Spring Tool Suite.

- Go to `File > Import`.

- Select `Existing Gradle Project` or `Existing Maven Project`, depending on the build tool used in your project.

- Browse to the location of the cloned project and click `Finish`.

3. Configuring MySQL Database:

- Open MySQL Workbench or your preferred MySQL client.

- Create a new database for your application.

- Update the application's database configuration in the project (e.g., in `application.properties` or `application.yml`) to point to your MySQL database.

5. Build and Run:

- Build the project within STS.

- Run the application as a Spring Boot application. Right-click the main application class and select "Run as > Spring Boot App."

6. Access the Application:

- Once the application is running, open a web browser and go to the specified URL (http://localhost:8081) to access the CRUD application.

**PROJECT PROCEDURE OF CRUD APPLICATION FOR PRODUCT**

1. Firstly, I had setup the required software into machine.

2. Then opened the Spring Tool Suite for Developing the business logic using the Spring Boot.

3. Created the project in STS and added all header files and developed the code.

4. Created the project with layer like Entity layer, Service layer, Repository layer, Controller layer.

5. Entity layer have the Variable Declaration and Lombok (toString , Setter , Getter , AllArgsConstructor, NoArgsConstructor) to avoid bowl code.

6. In Service layer, developed the business logic code

7. In Repository layer, created the JPA code.

8. In Controller layer, developed the Controllers of the Http controllers.

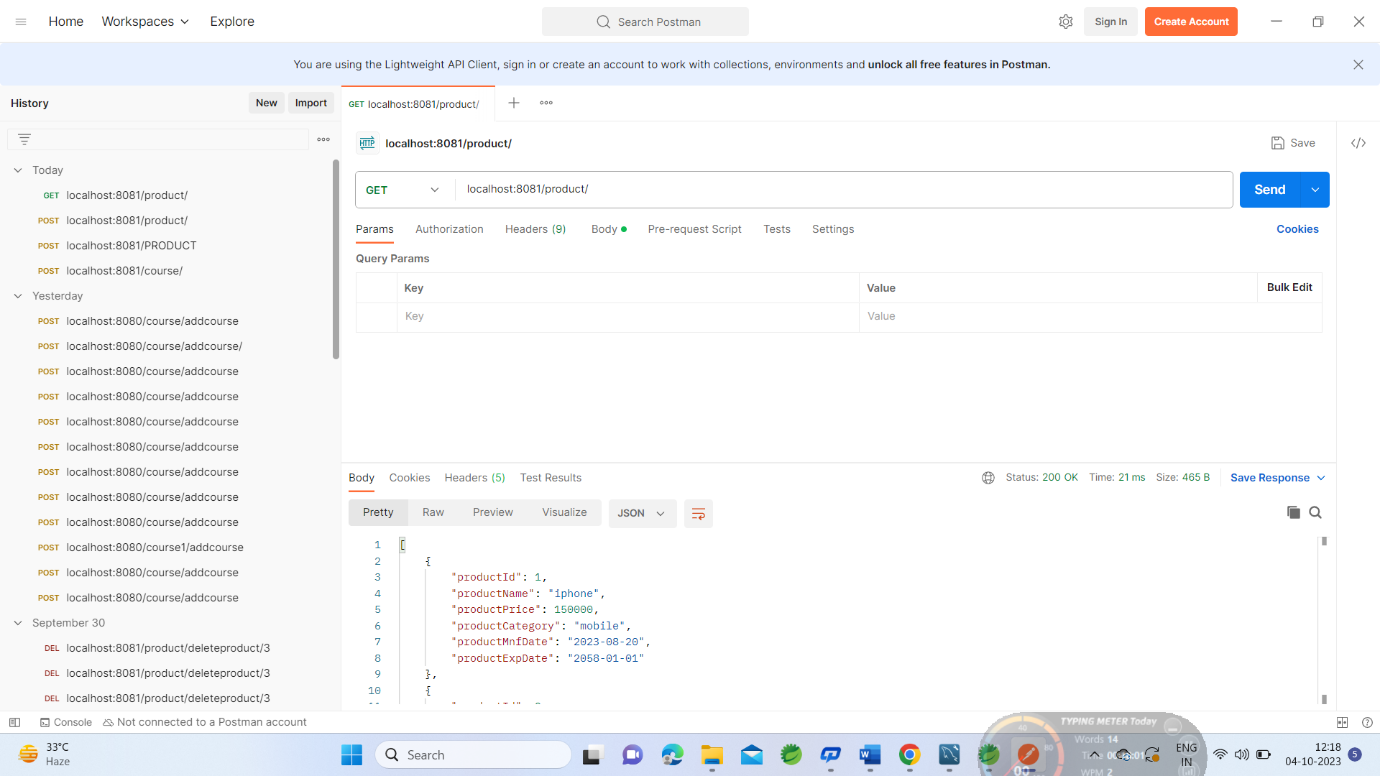
9. Using Application.properites linked JDBC connectivity.

10. Created the Database in MySQL for storing the data into Database.

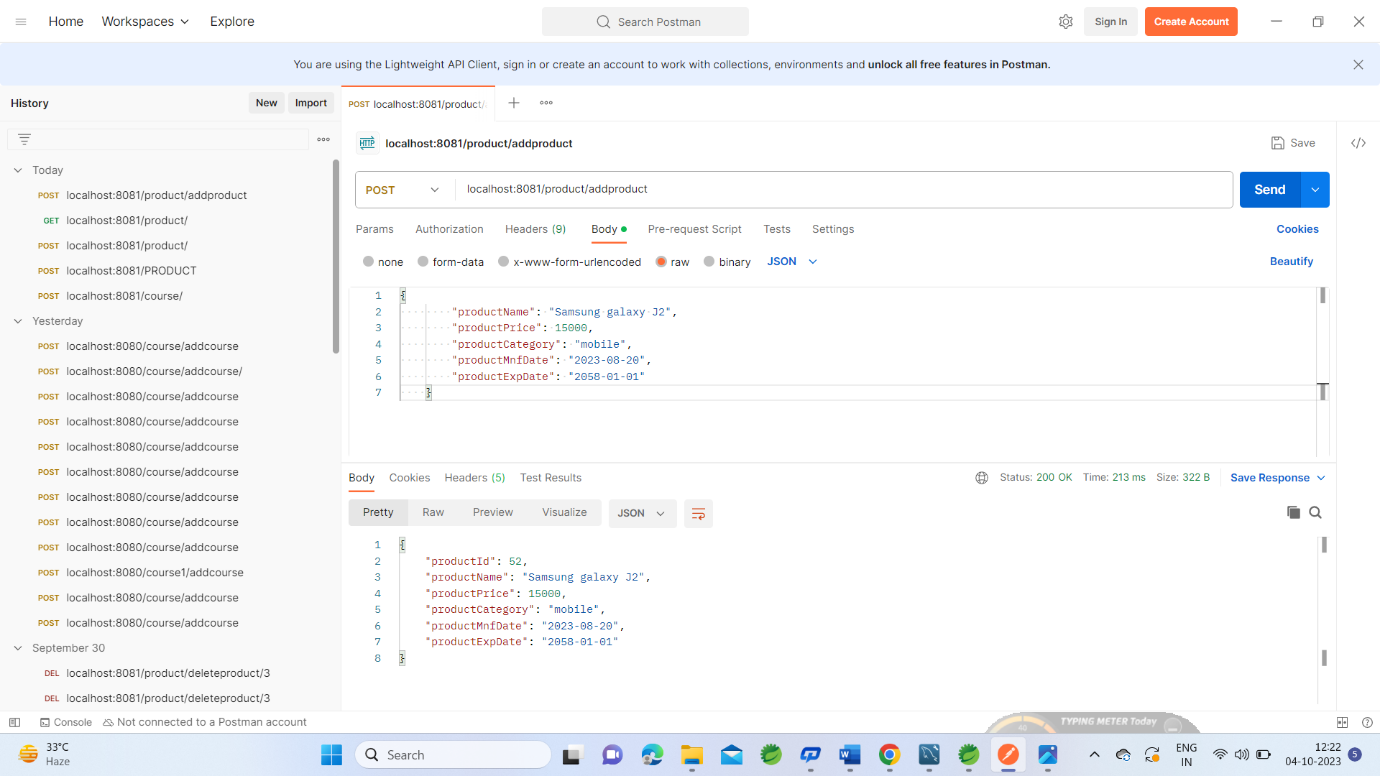
11. After completion of backend code, we need to check the project by using Postman API for backend purpose because every business logic is working or not, we need to find out.

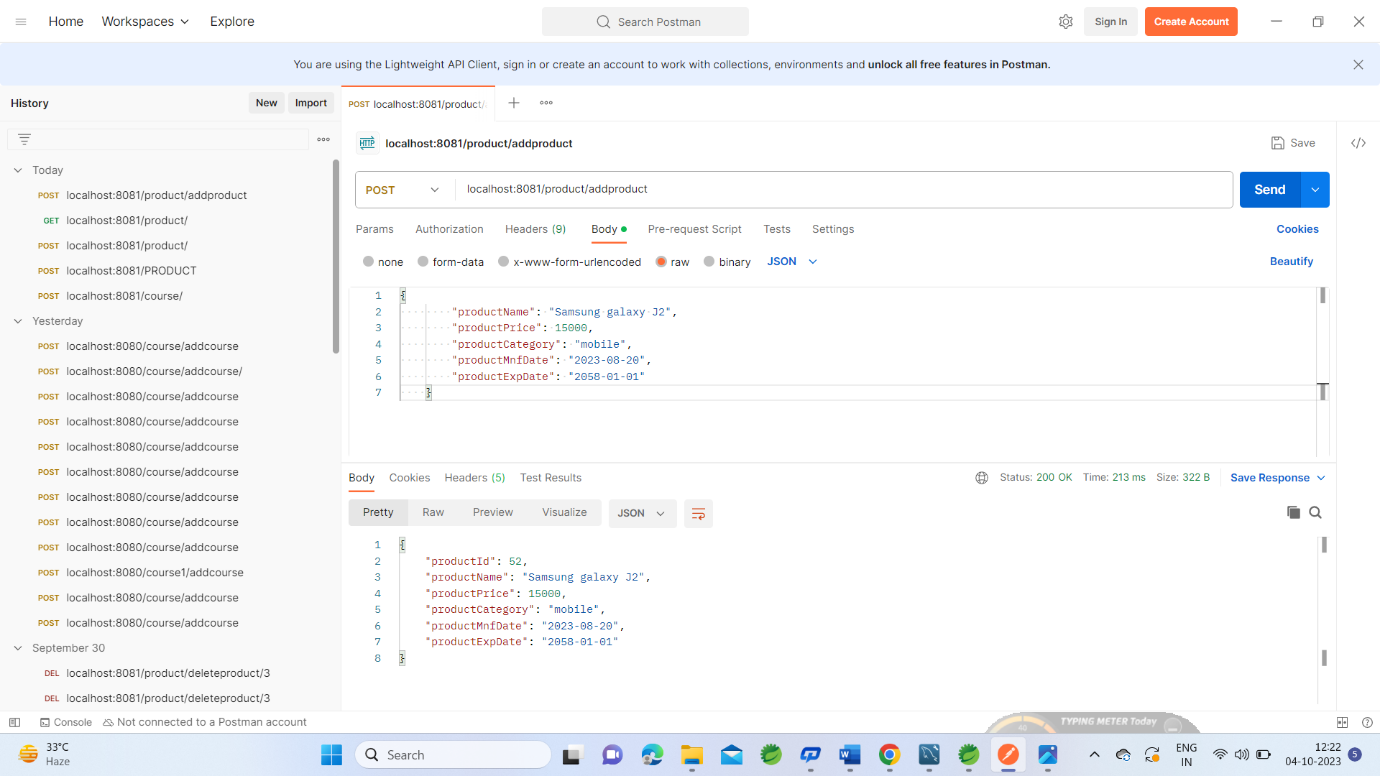
12. By using Postman API, data is pushed into database, we need to check it and in the given below pictures, you can see the CRUD operation is working.

**DISPLAYING PRODUCT**

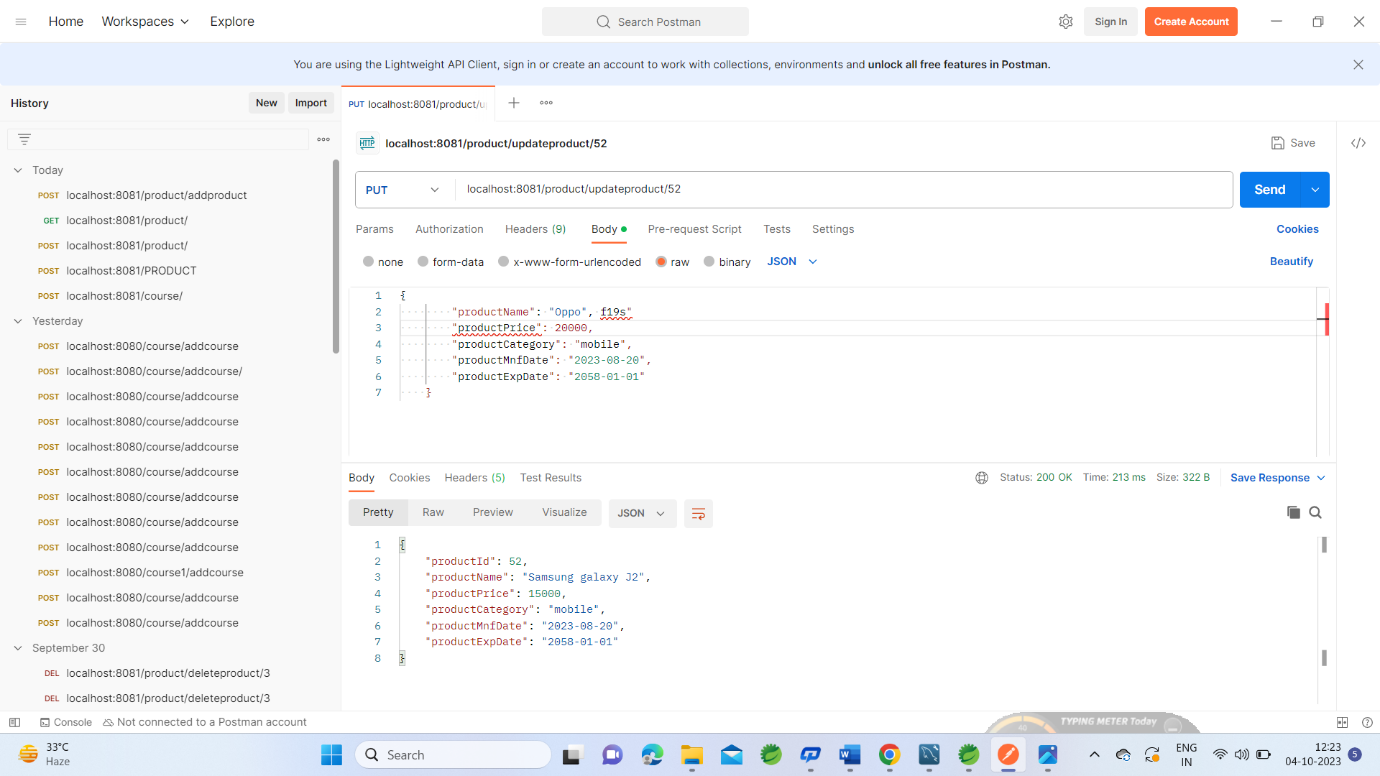
****

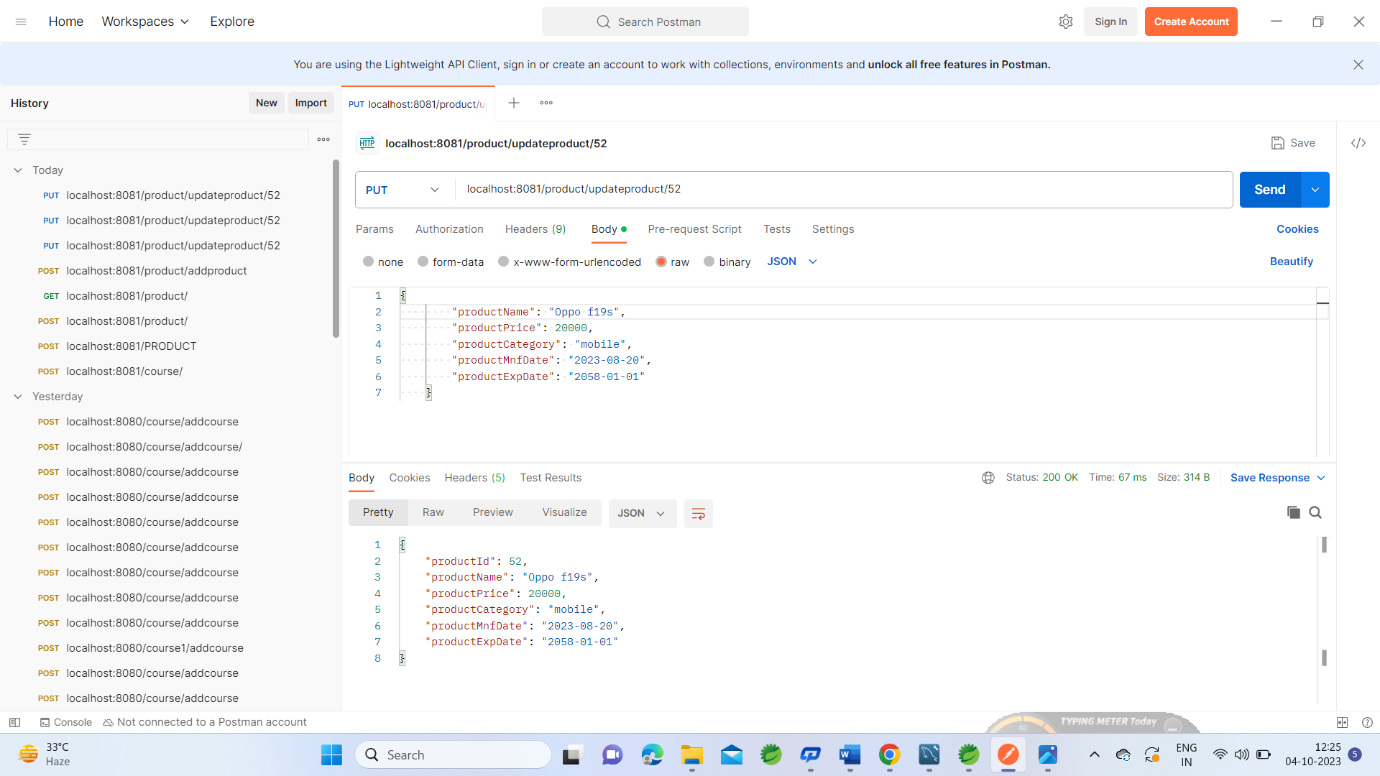
**ADD PRODUCT**





**UPDATE PRODUCT**





**DELETE PRODUCT**

