**STUDENT CRUD MANAGEMENT SYSTEM USING SPRINGBOOT**

**1. Project Overview**

The **Student CRUD Project** is a simple RESTful application developed using **Spring Boot**. Its main purpose is to perform basic **CRUD operations** (Create, Read, Update, Delete) on student records. Each student has the following three attributes:

* **Name:** Full name of the student
* **Email:** Unique identifier for the student
* **Course:** The course in which the student is enrolled

This project helps beginners understand **REST API concepts** and the flow of data using **JSON format**.

**2. Tools and Technologies Used**

* **Java 8+**: Programming language
* **Spring Boot**: Framework for building RESTful applications
* **H2 Database**: In-memory database for development and testing
* **Postman / CURL / PowerShell**: For testing the REST APIs
* **IntelliJ IDEA**: For coding and project management

**3. Project Structure**

The project follows a **layered architecture**:

1. **Model Layer:** Represents the structure of the student entity.
2. **Repository Layer:** Handles interaction with the database.
3. **Controller Layer:** Manages the API requests and responses.

This structure ensures separation of concerns and makes the application maintainable.

**4. Functionality of CRUD Operations**

1. **Create (POST):**  
   Adds a new student to the database. The student’s data is sent in **JSON format** and saved.
2. **Read (GET):**
   * **Read All:** Retrieves all student records from the database.
   * **Read One:** Retrieves a specific student’s record using the unique email ID.
3. **Update (PUT):**  
   Updates existing student details. The email serves as a unique identifier, while other fields like name and course can be updated.
4. **Delete (DELETE):**  
   Deletes a student record based on the unique email ID.

**5. Data Format**

All CRUD operations use **JSON format** for sending and receiving data. This ensures a standard structure for requests and responses and allows easy integration with other systems or front-end applications.

**6. Advantages of the Project**

* Demonstrates the complete **CRUD workflow**
* Uses **RESTful APIs**, which are widely used in modern web applications
* Allows practice with **JSON data format**
* Can be extended easily to include additional fields, validations, or authentication

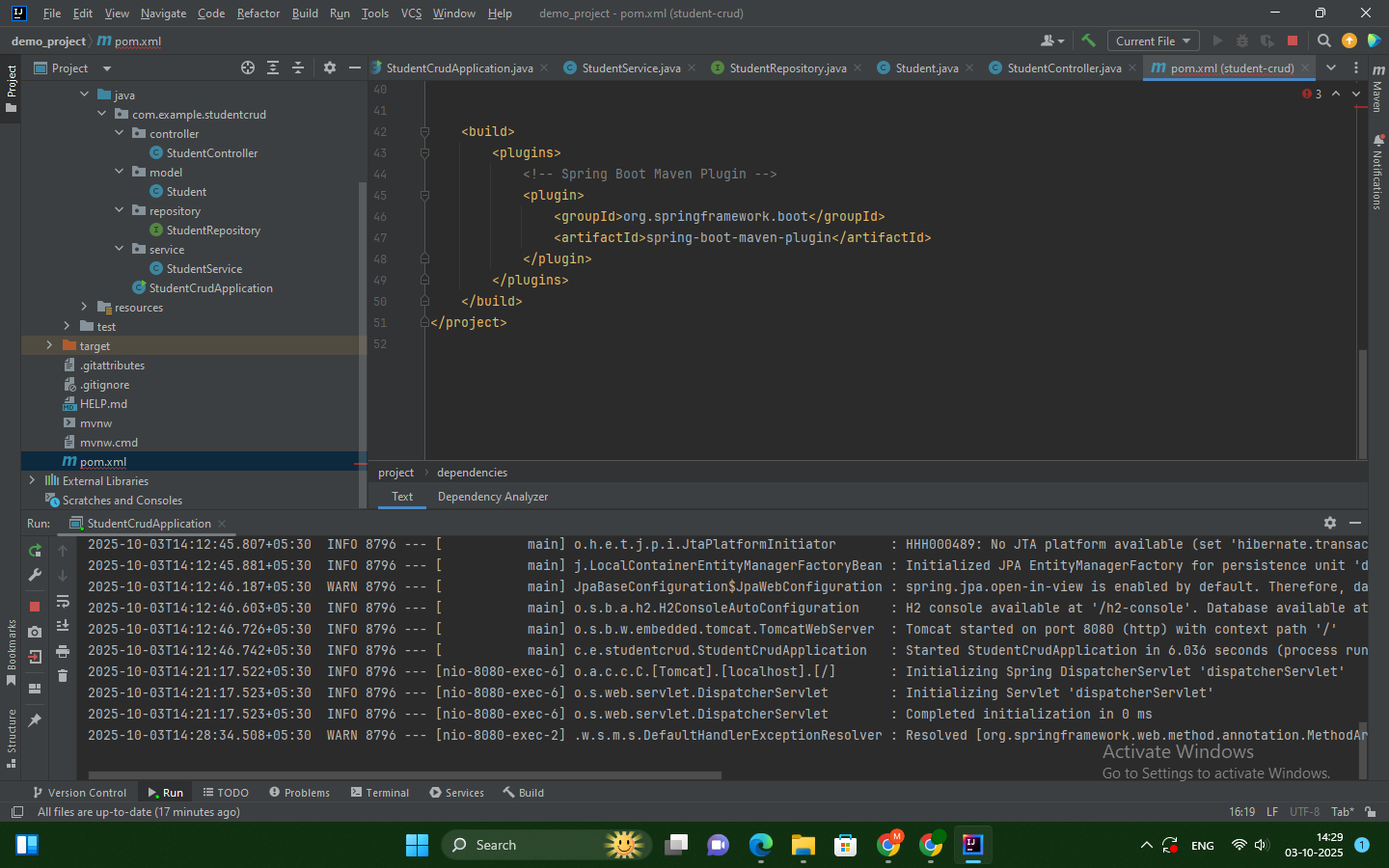
**7. Learning Outcomes**

After completing this project, a student or developer will be able to:

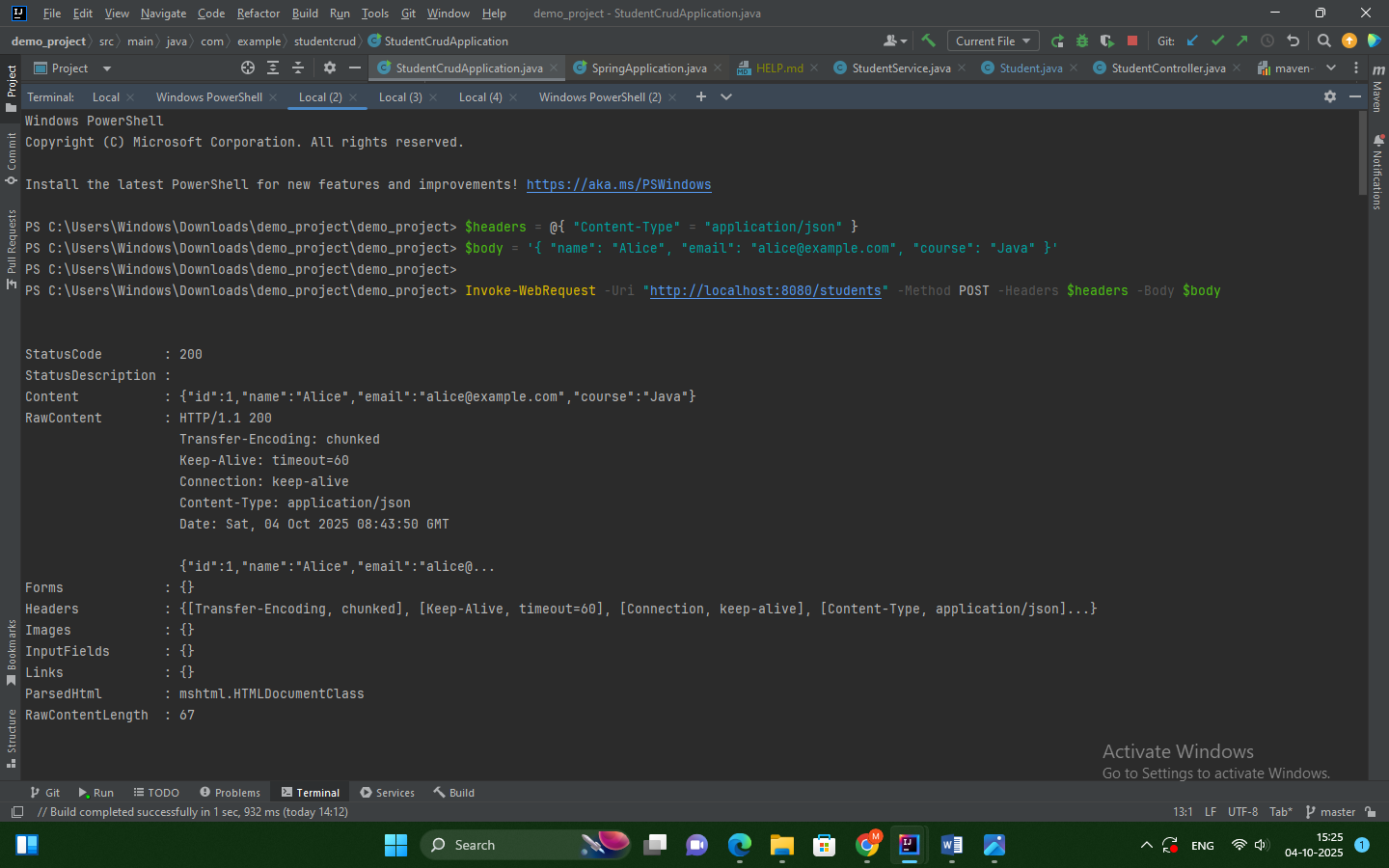
* Understand **basic REST API design**
* Perform CRUD operations on an in-memory database
* Handle **JSON requests and responses**
* Implement a simple layered architecture using Spring Boot
* Prepare for building more **complex applications**

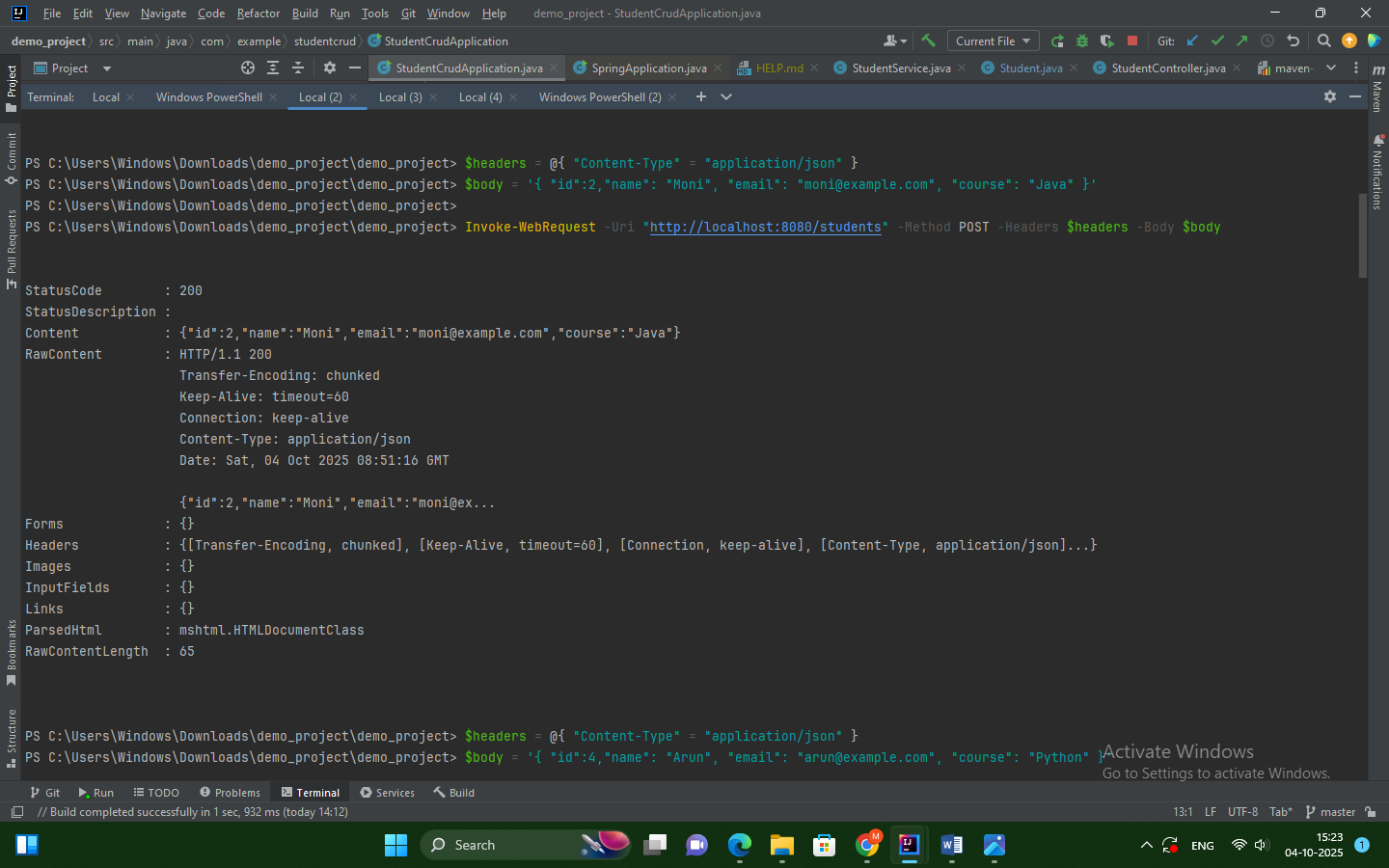
**OUTPUT:**

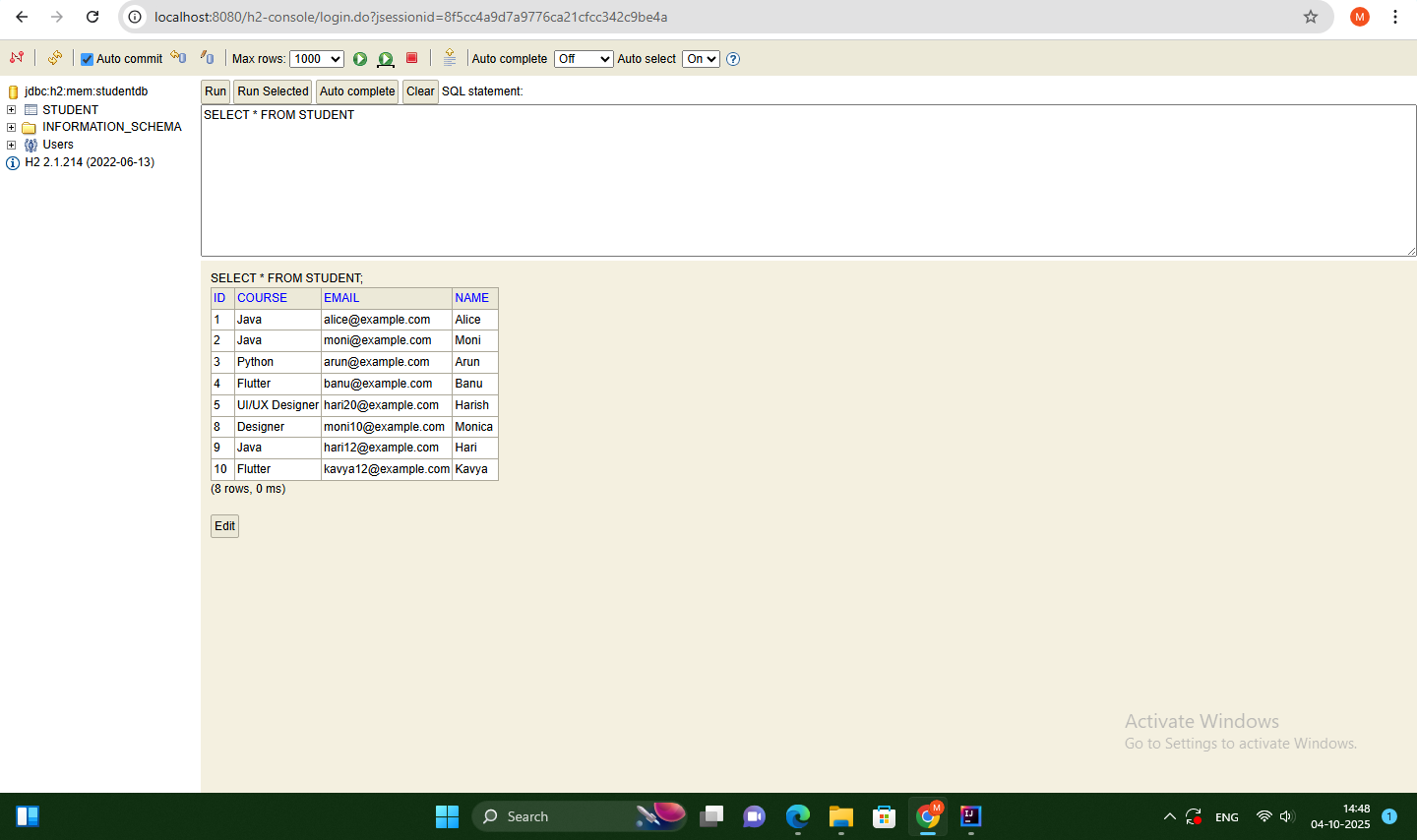
* To run this application in Intellij,first needs to update maven in pom.xml using (mvn clean install) in terminal
* After that we have to run mainApplication using green button for starting the application
* The application will start, and tested the APIs using **PowerShell**.Following screenshots are output of this application

****

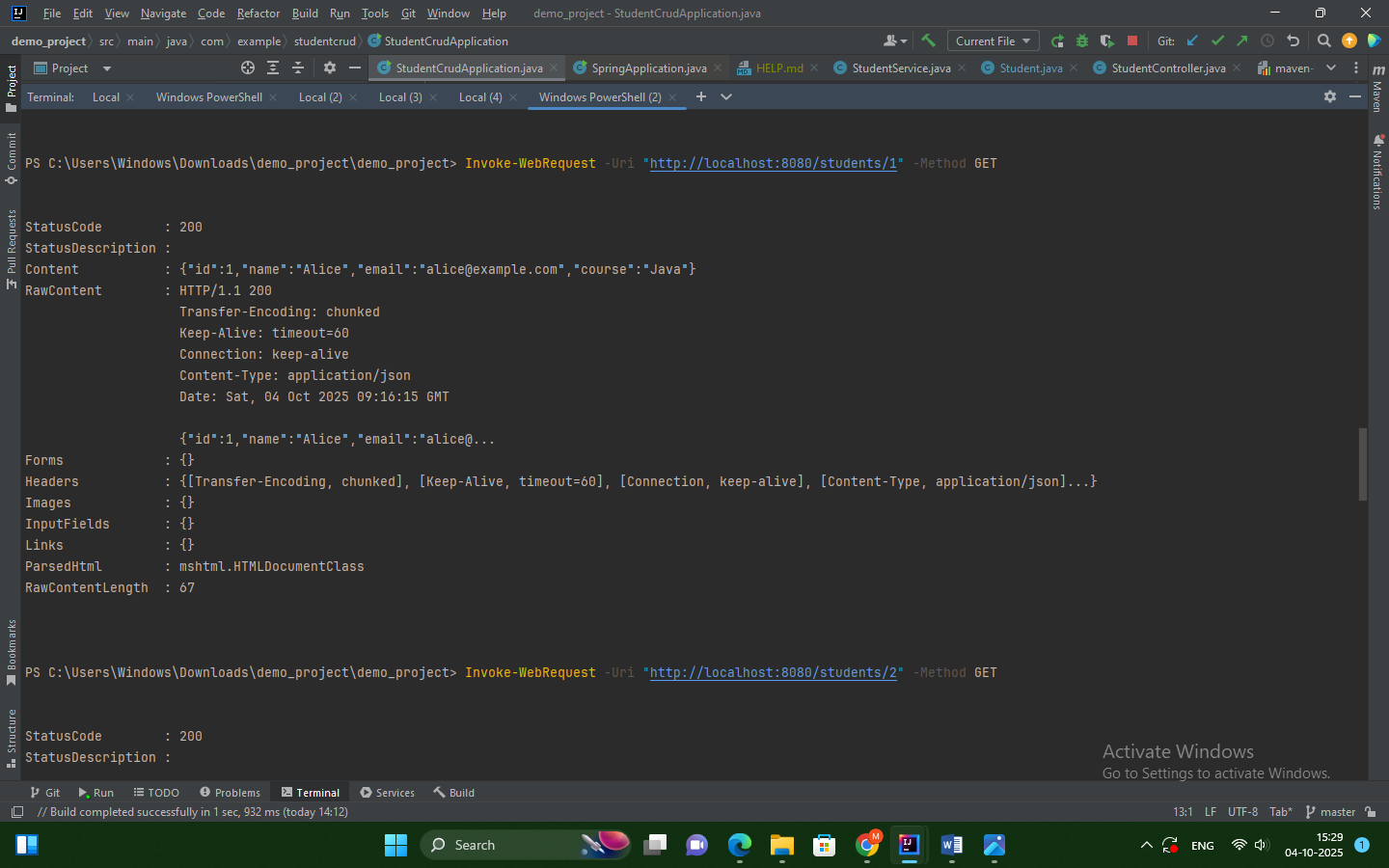
**In Terminal:1.POST:**

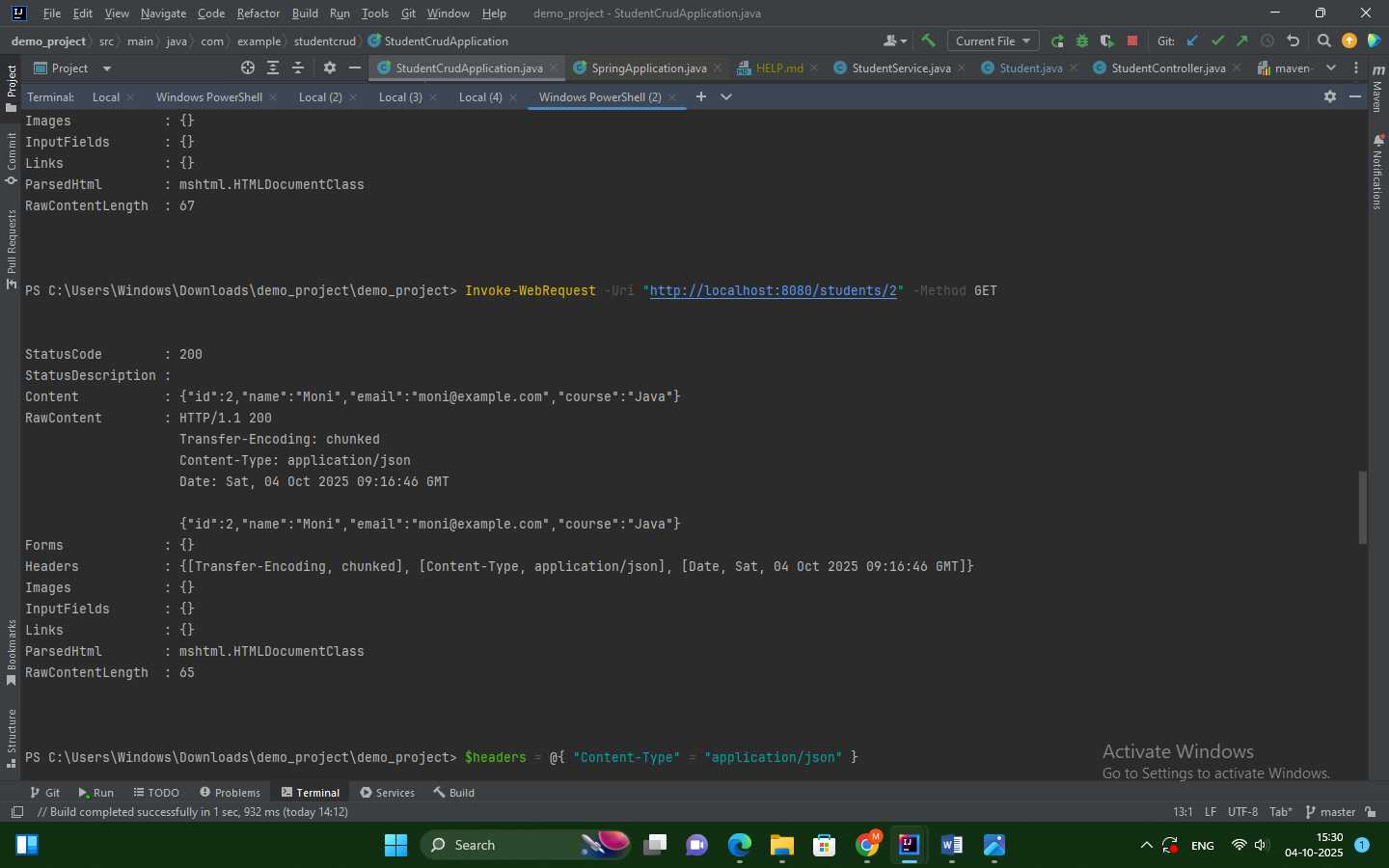




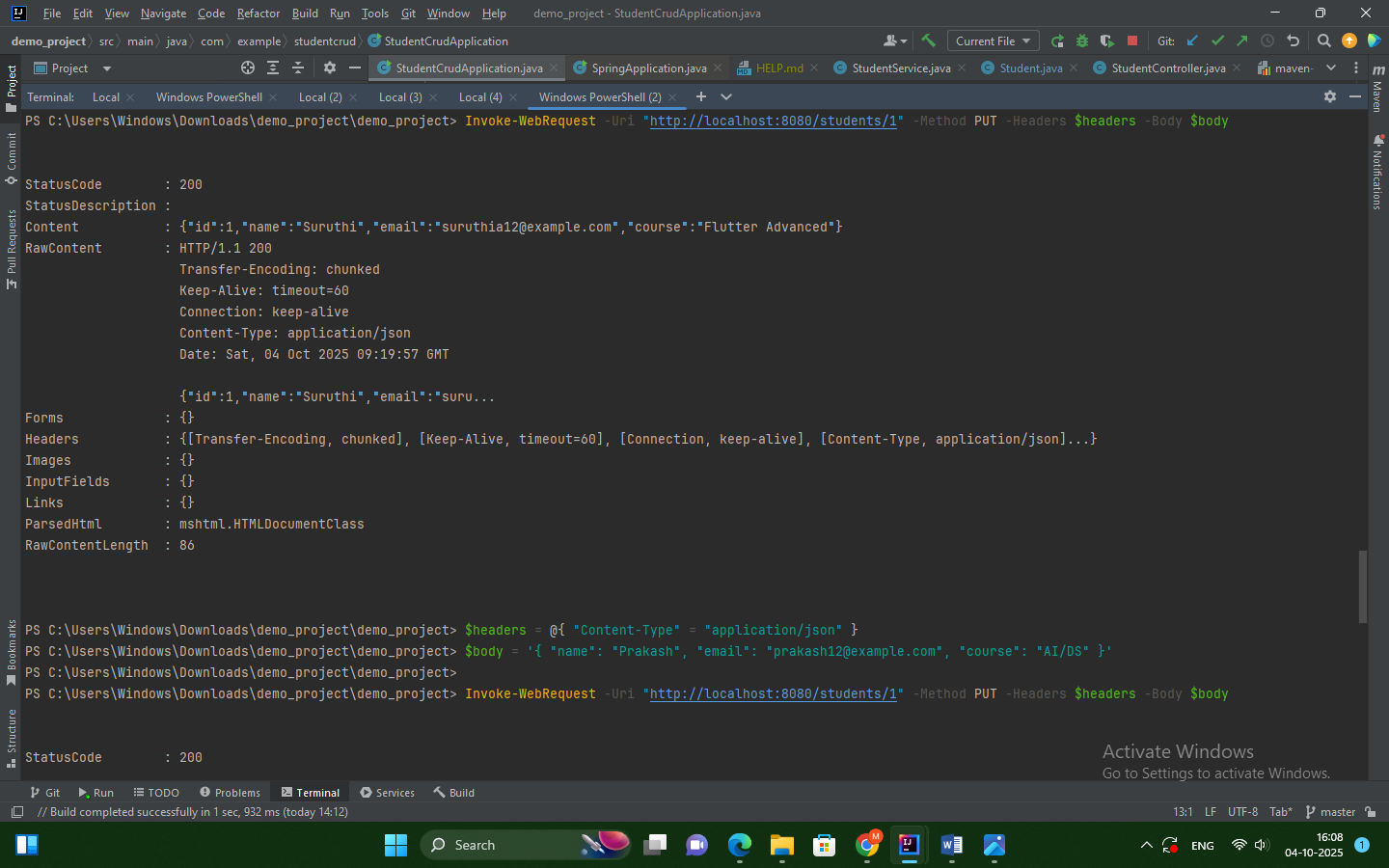


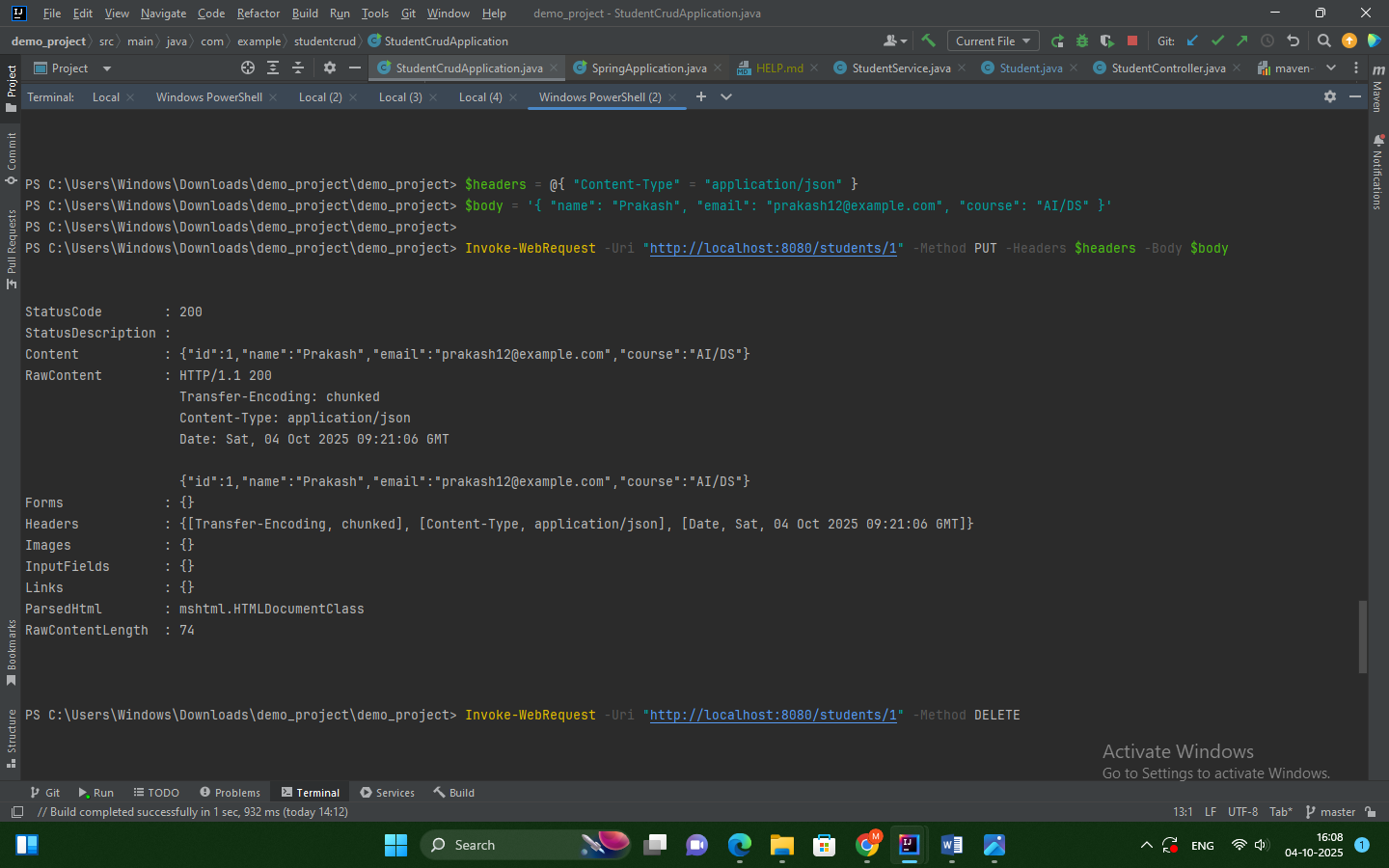
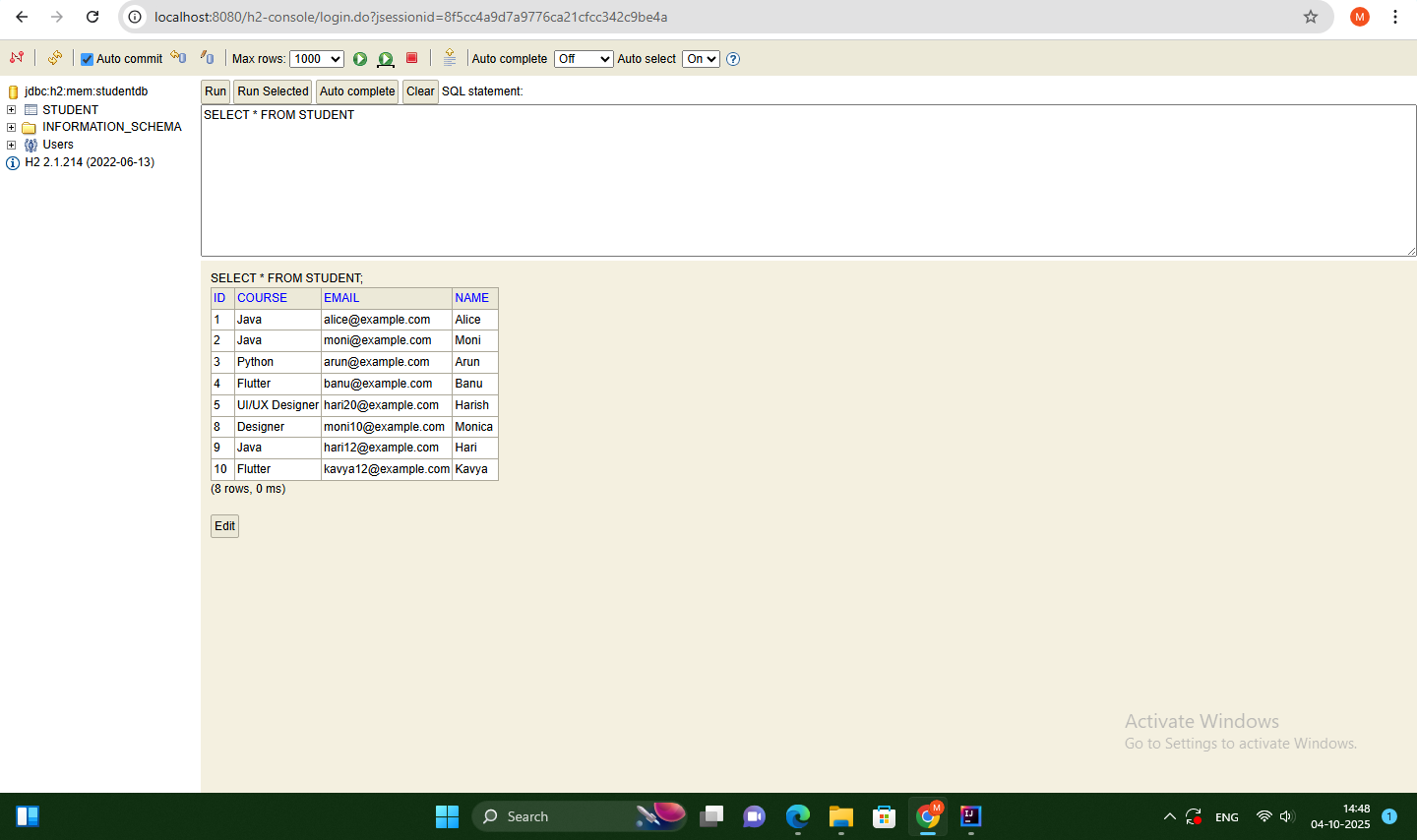
**2.GET:**



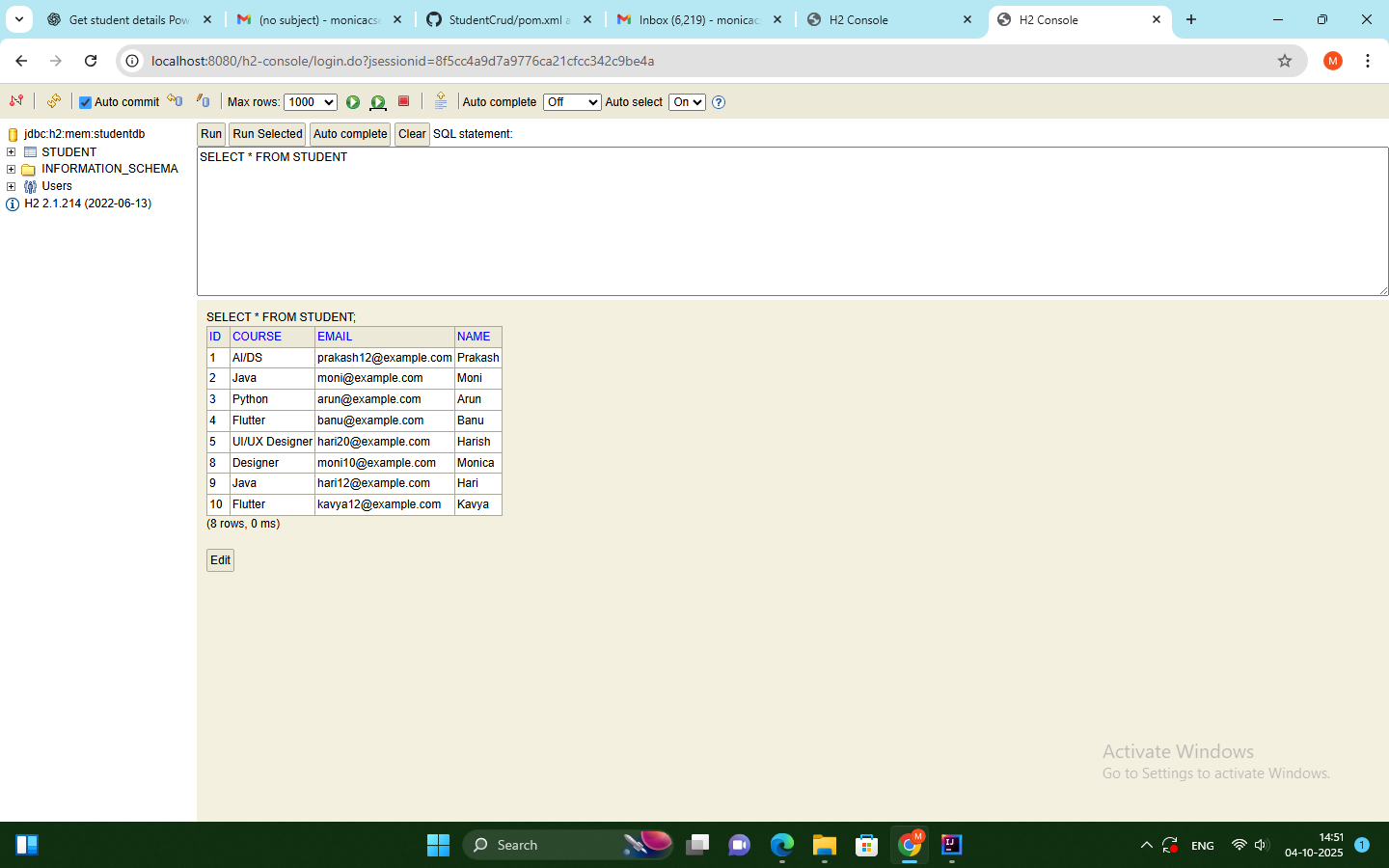


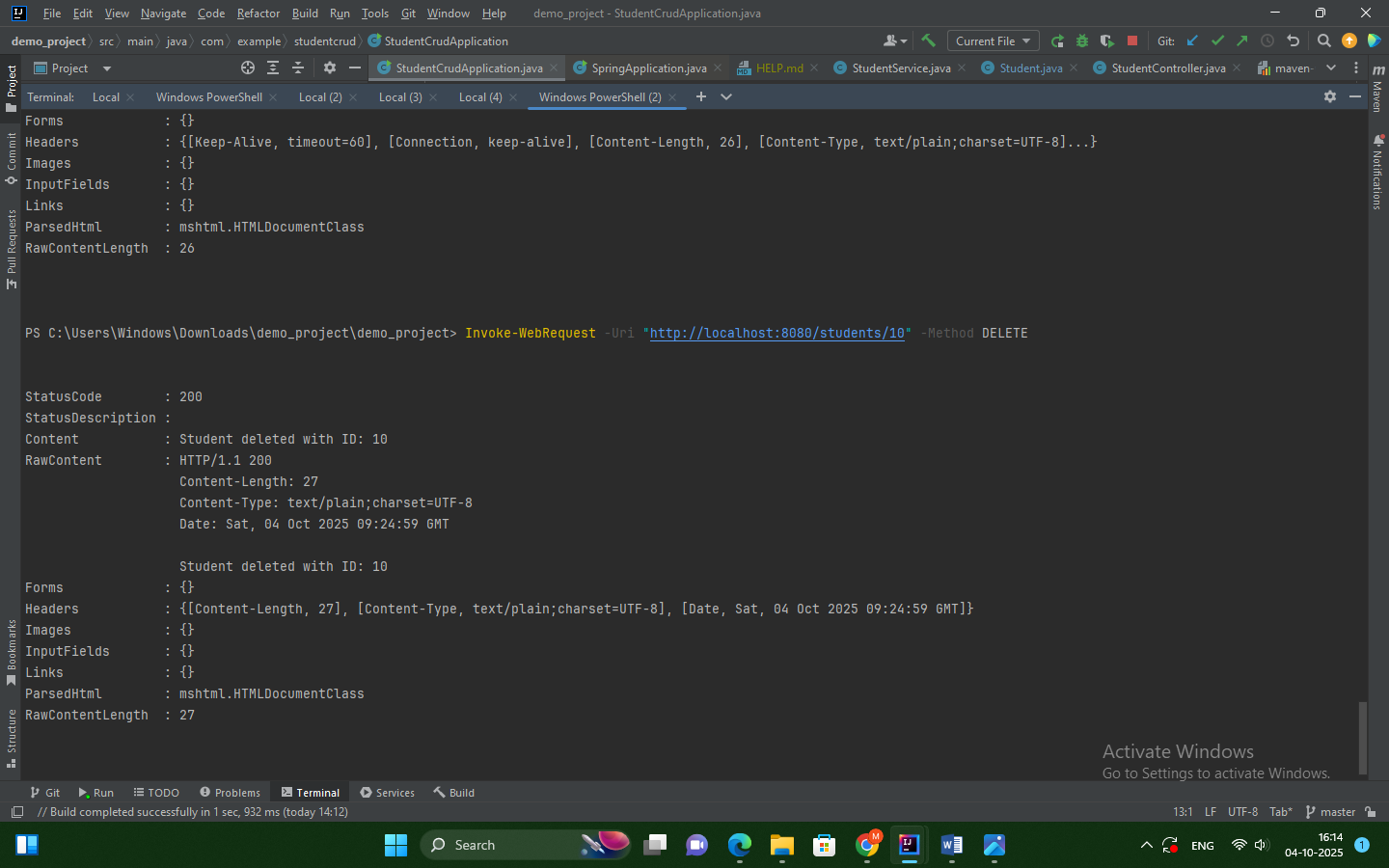
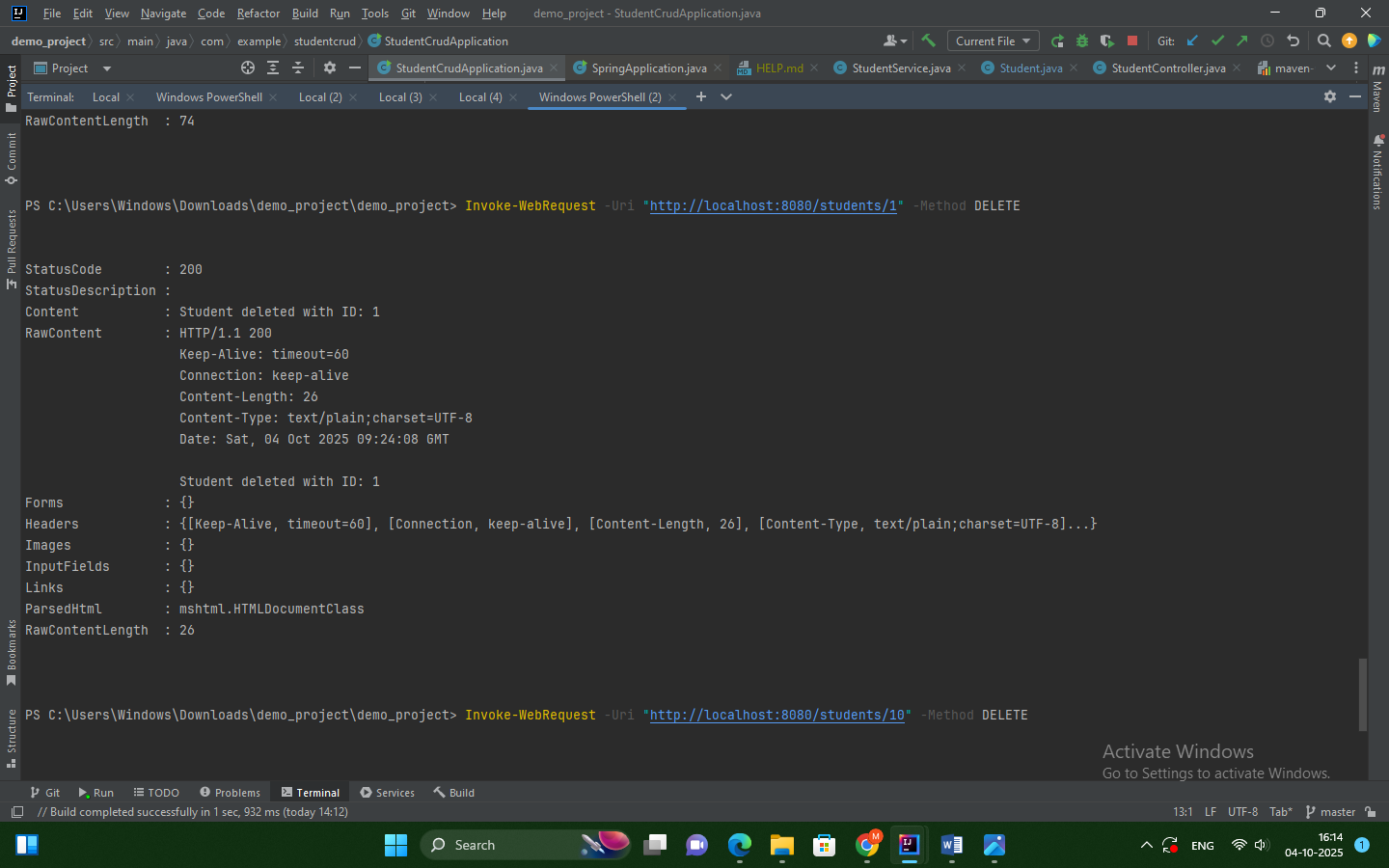
**3.PUT**



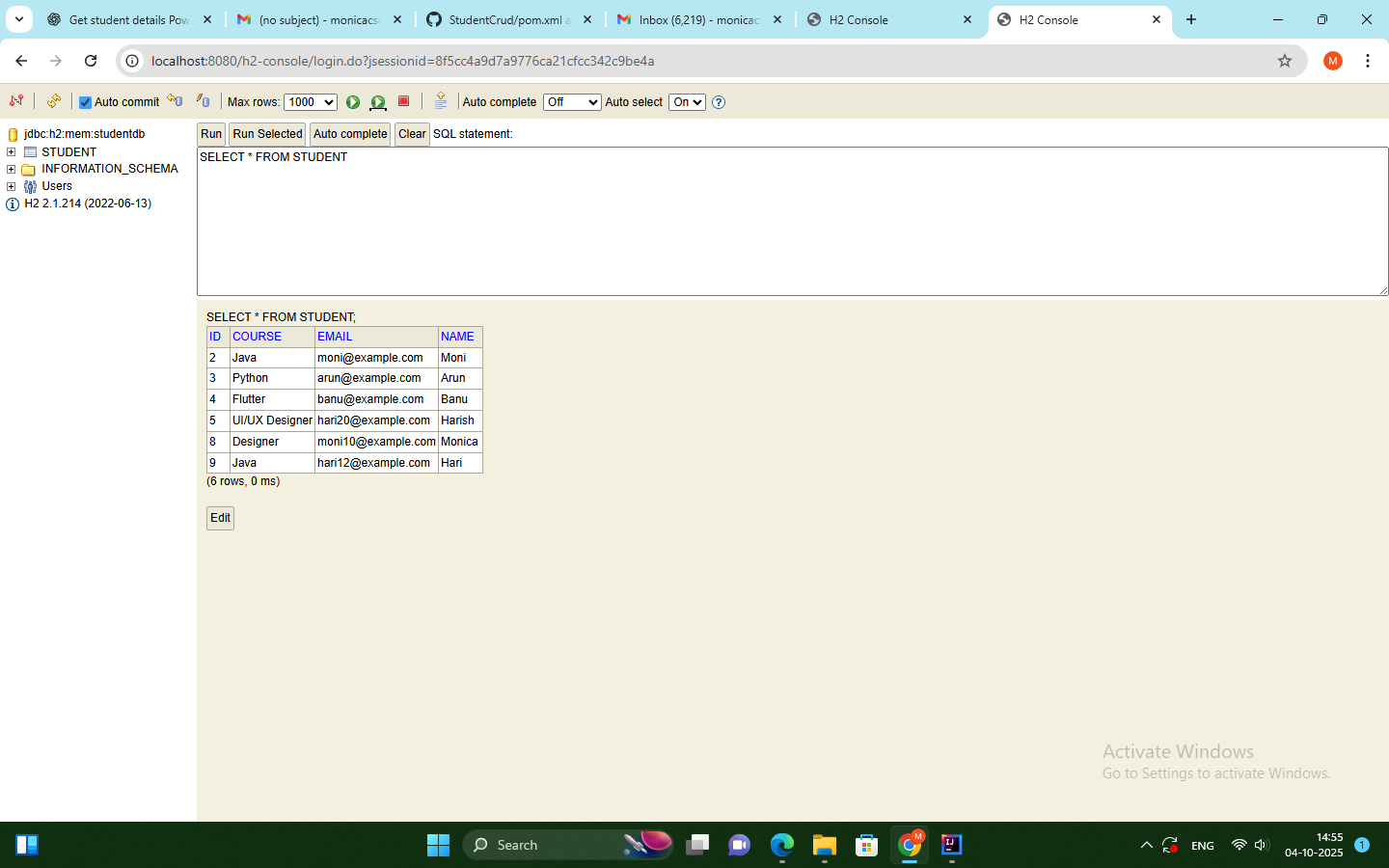
**BEFORE UPDATE:**

**AFTER UPDATE:**



**4.DELETE:**

**AFTER DELETING IN H2 CONSOLE:**

****

**8. Conclusion**

The Student CRUD Project is a **beginner-friendly application** to understand the core concepts of **Spring Boot, REST APIs, and JSON-based data handling**. It forms the foundation for more advanced projects involving multiple entities, relational databases, and web applications.