Batch:D-2	**Roll No.:16010123325**
Experiment / assignment / tutorial No._7___	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

Title: Create a RESTful API server in Express and Node.js. Implementation + Testing application u postman/Thurderclient

AIM: Create a RESTful API server in Express and Node.js. Implementation + Testing application using postman/Thurderclient

Problem Definition: To develop and test a RESTful API using Node.js and Express.js that performs CRUD (Create, Read, Update, Delete) operations on student data. The API should handle JSON requests and responses, and be tested using Thunder Client to verify proper backend functionality.

Resources used:

- Software:
  - Node.js
  - Visual Studio Code
  - Thunder Client (VS Code Extension)
- Libraries:
  - Express.js
  - Body-Parser (for JSON parsing)

Expected OUTCOME of Experiment:

Successful implementation of a RESTful API that performs CRUD operations using Express.js and Node.js, with verification through Thunder Client showing correct HTTP responses and JSON data for each operation.

CO 3: Test the concepts and components of various front-end, back-end web app development technologies & frameworks using web development tools.

Books/ Journals/ Websites referred:

1. Official Express.js Documentation – https://expressjs.com
2. MDN Web Docs – REST API Basics and HTTP Methods

Pre Lab/ Prior Concepts:

Thunder Client is a lightweight REST API client available as an extension inside VS Code. It allows developers to test their backend APIs by sending HTTP requests (GET, POST, PUT, DELETE) and receiving responses directly in JSON format. It simplifies testing without needing an external tool like Postman.

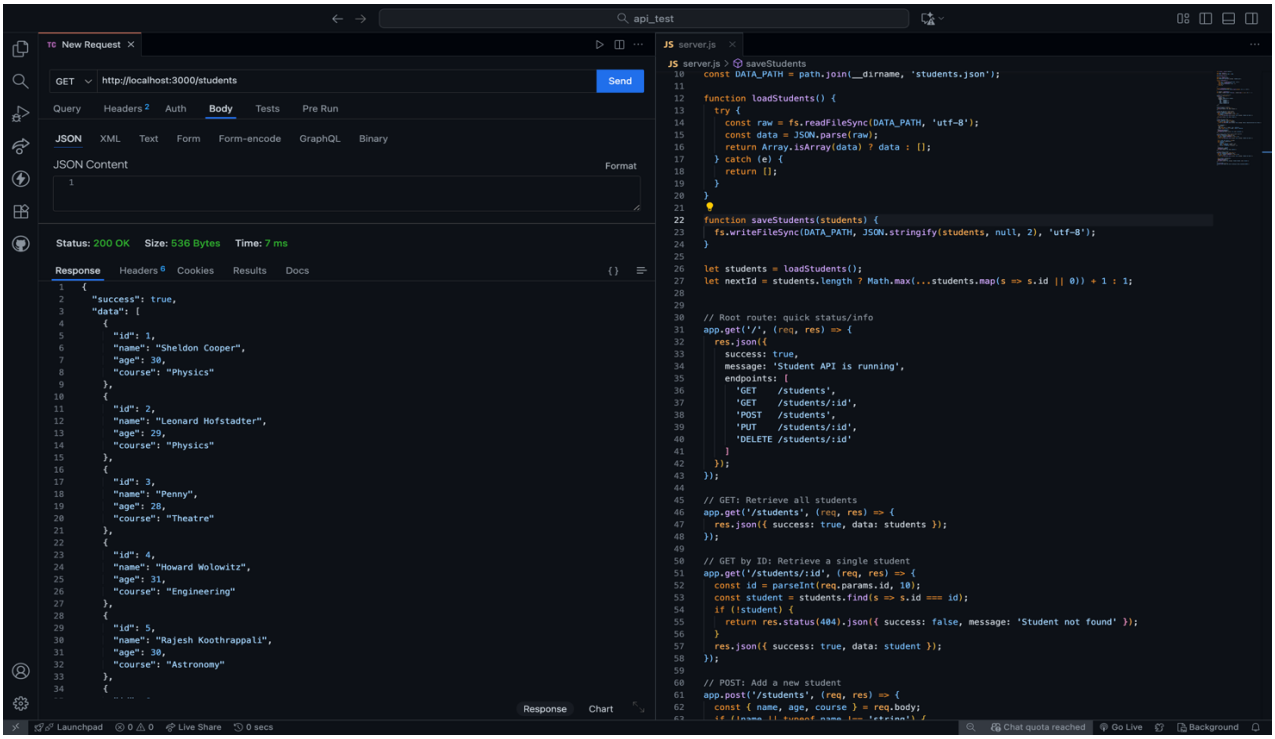
Methodology:

1. Initialize a Node.js project using npm init -y.
2. Install Express.js using npm install express.
3. Create a server.js file to define CRUD (Create, Read, Update, Delete) routes.
4. Run the server using node server.js.
5. Use Thunder Client to test each API endpoint.

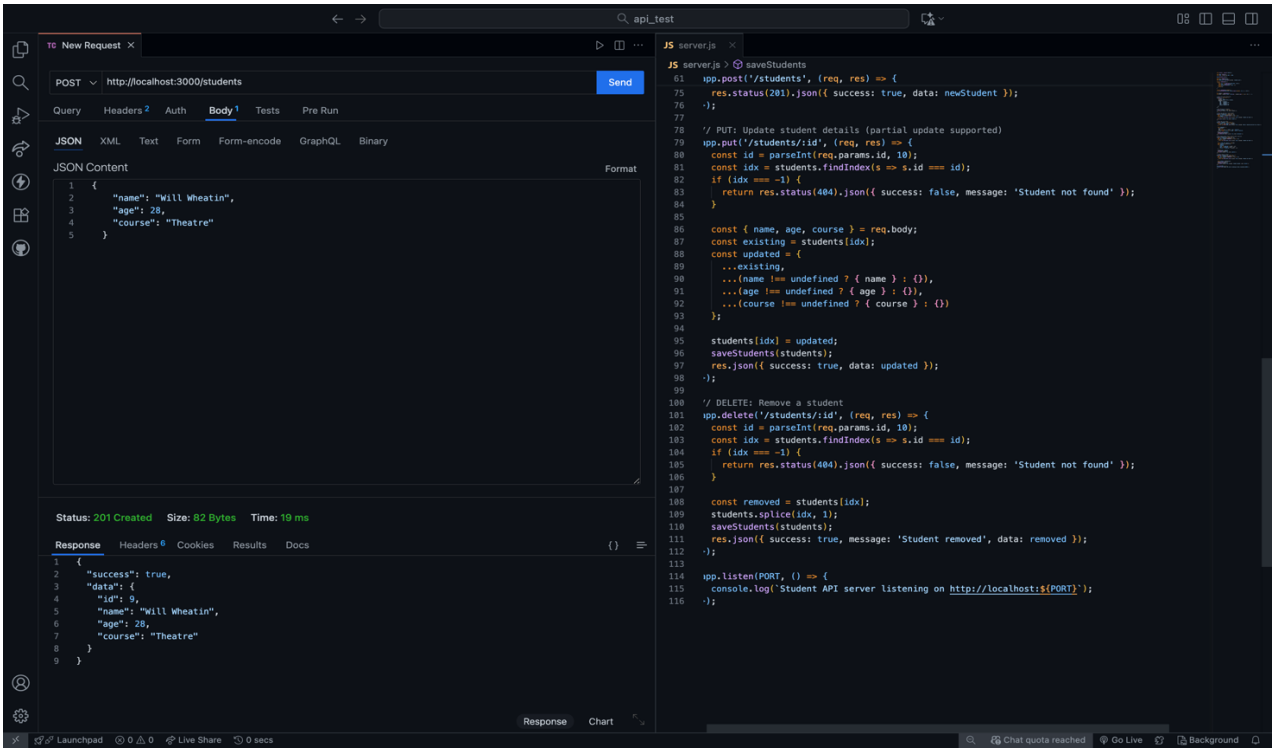
Implementation Details:
The implementation includes defining RESTful endpoints using Express.js. Each endpoint corresponds to a CRUD operation on a simple student dataset.

Steps for execution:
1. Start the server using the command: node server.js
2. Open Thunder Client from the VS Code sidebar.
3. Perform the following operations:
   - GET: Retrieve all students.
   - POST: Add a new student.
   - GET by ID: Retrieve a single student.
   - PUT: Update student details.
   - DELETE: Remove a student.

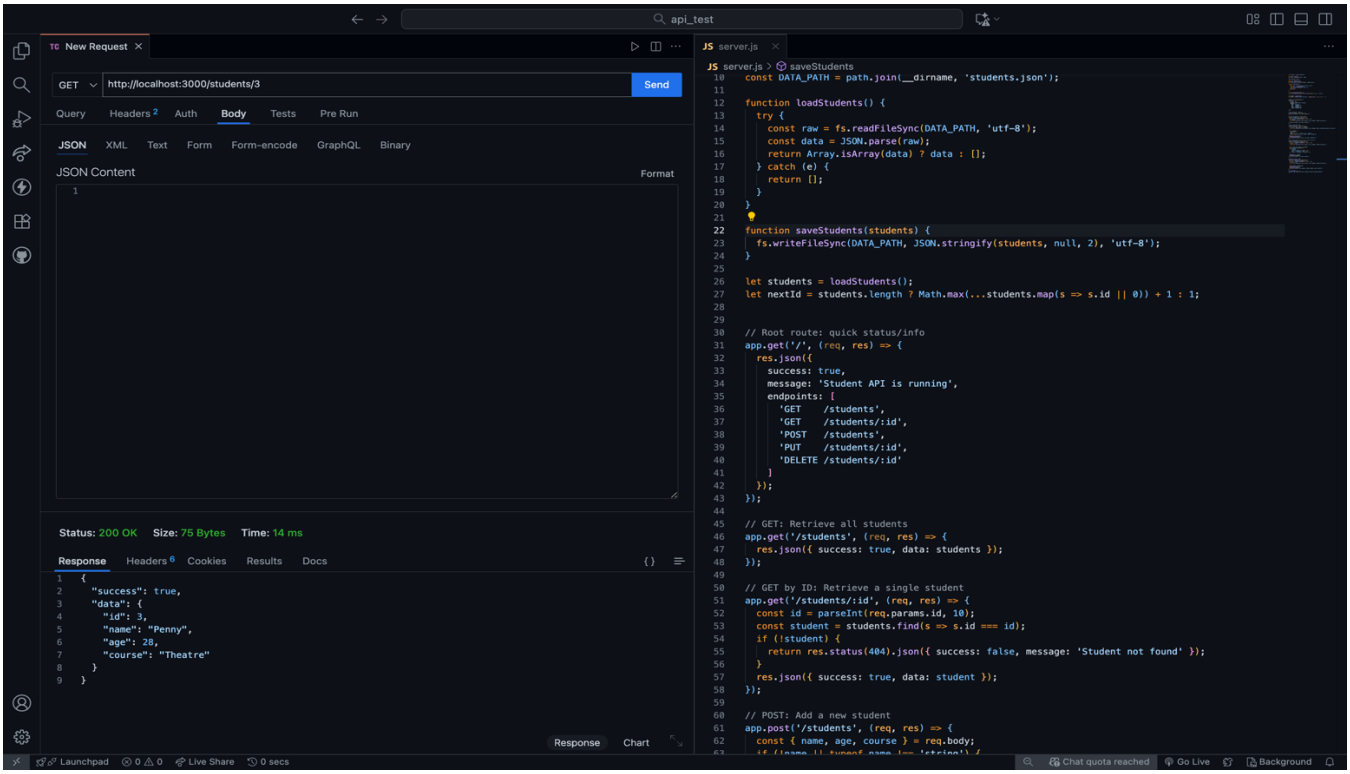
GET Request – Fetch All Students



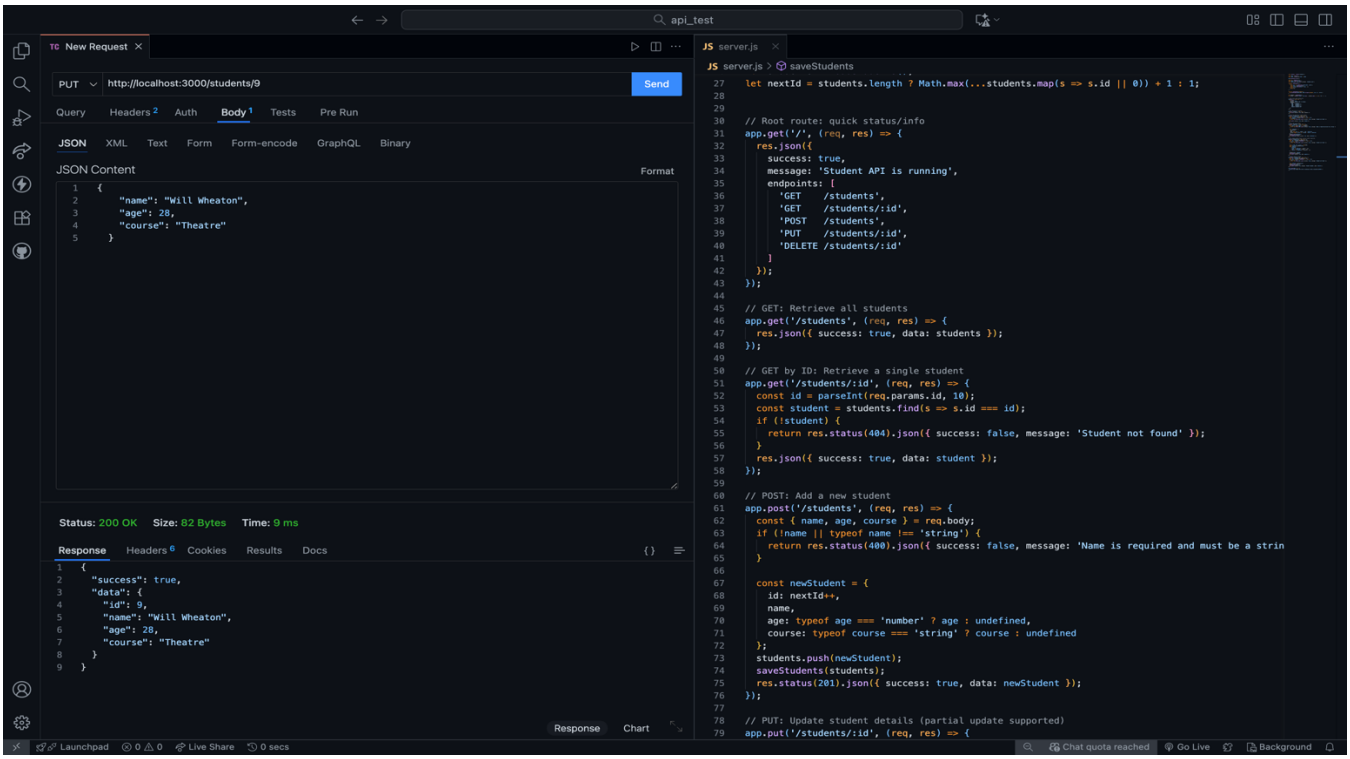
POST Request – Add New Student



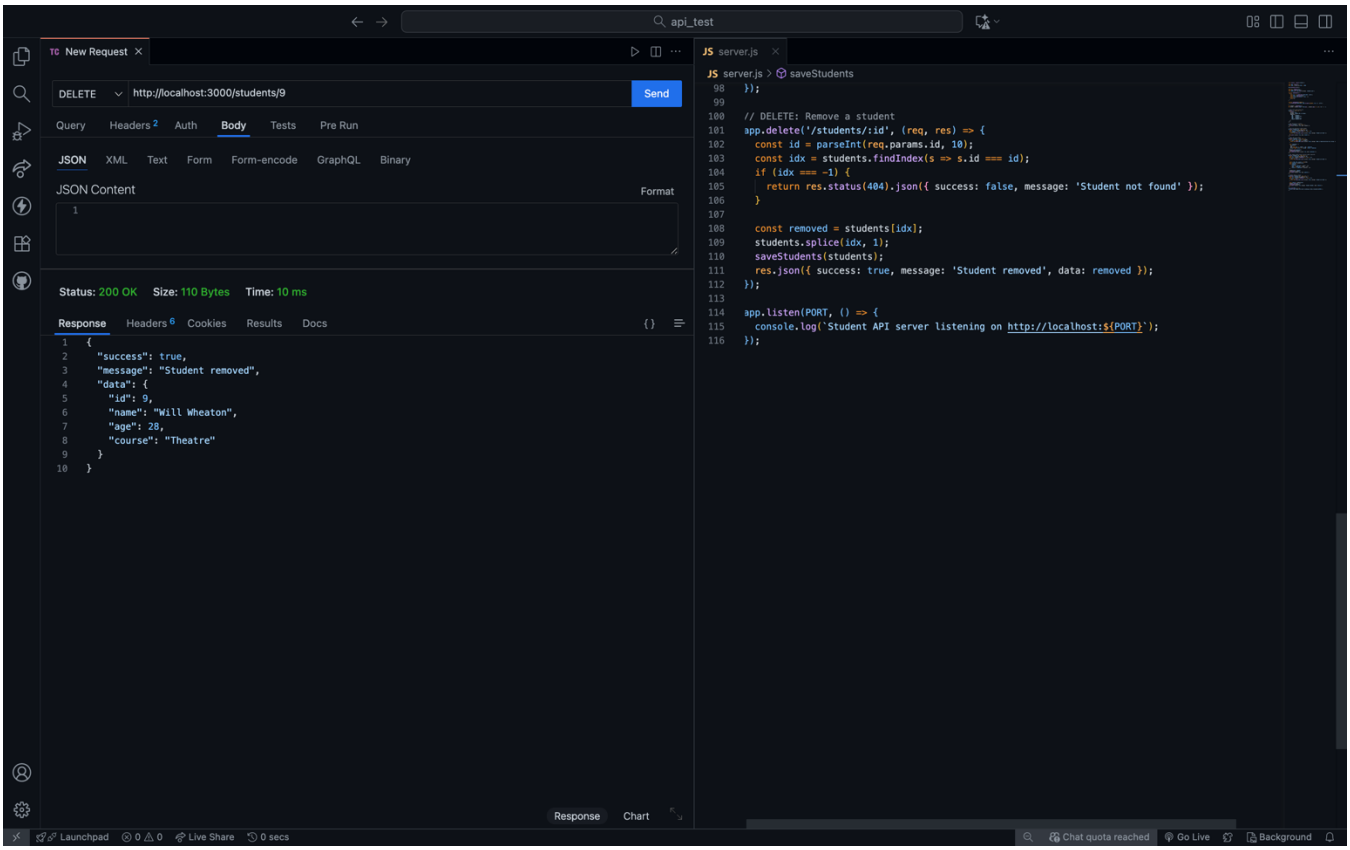
GET by ID Request – Fetch Single Student



PUT Request – Update Student Data



DELETE Request – Remove Student Record



Conclusion:

In this experiment, a RESTful API was successfully implemented using Express.js and Node.js. The API handled CRUD operations effectively and was tested using Thunder Client. All endpoints returned the expected responses, demonstrating the core functionality of RESTful services.