Submitted by:

Aiman Sadia

Domain:

Back-End Development

Internship Program:

Cosmicode

Date:

June 27, 2025

Week 1: Introduction to Back-End Development (Beginner)

Objective: Understand back-end concepts, tools, and simple server setup.

Tasks:

1. Write a document explaining **what back-end development is**, how it works with front-end, and list popular back-end languages and frameworks.
2. Install **Node.js / Python / Java (pick one stack)** and set up your development environment.
3. Create your **first "Hello World" server** using Express (Node.js) or Flask (Python) or Spring Boot (Java).
4. Set up **basic routing (GET, POST)** for your server.
5. Connect your back-end app to a **local database (SQLite/MySQL/PostgreSQL)** and perform basic CRUD (Create, Read, Update, Delete) operations.

TASK 01:

Write a document explaining what back-end development is, how it works with front-end, and list popular back-end languages and frameworks.

What is Back-End Development?

Back-end development refers to the server-side part of a web application. It is responsible for managing the logic, database interactions, authentication, and server configuration that make a website or app function behind the scenes. Unlike front-end development, which focuses on what users see and interact with, back-end development deals with the data processing and system integration that users don't see directly.

Back-end developers build and maintain the core functional logic of a system. This includes:

- Server-side application logic
- Databases and data storage
- APIs (Application Programming Interfaces
- Authentication and authorization
- Integration with third-party services

How Back-End Works with Front-End

Front-end and back-end development work together to create complete web applications:

1. **Front-End (Client-Side)**: This is what users see HTML, CSS, JavaScript, and UI frameworks. It handles the layout, design, and user interaction.
2. **Back-End (Server-Side)**: This handles the business logic, retrieves data from databases, processes it, and sends it back to the front-end.

Example Interaction:

1. A user submits a login form (front-end).

2. The data (username and password) is sent to the back-end server.
3. The back-end validates the credentials against stored records in a database.
4. The server responds with success or failure, which is then shown to the user by the front-end.

Together, both sides create a seamless user experience while ensuring data is handled securely and efficiently.

Popular Back-End Languages and Frameworks

Below are some widely used back-end programming languages and their corresponding frameworks:

Languages:

- **JavaScript (Node.js)** – Popular for full-stack development using JavaScript on both front-end and back-end.
- **Python** – Known for its simplicity and readability; widely used in web and data applications.
- **Java** – A mature, object-oriented language often used in enterprise-level applications.
- **PHP** – A scripting language mainly used for server-side web development.
- **Ruby** – Valued for its developer-friendly syntax.
- **C#** – A Microsoft language used mainly for Windows and ASP.NET development.
- **Go (Golang)** – Known for high performance and simplicity in scalable systems.

Frameworks:

JavaScript/Node.js:

- **Express.js** – Minimalist and fast Node.js web framework.
- **NestJS** – A scalable framework built with TypeScript.

Python:

- **Django** – High-level framework that emphasizes rapid development and clean design.

- **Flask** – Lightweight and flexible microframework.

Java:

- **Spring Boot** – Modern framework that simplifies Java application development.

PHP:

- **Laravel** – A modern, elegant framework with robust features.
- **Symfony** – A flexible and powerful PHP framework.

Ruby:

- **Ruby on Rails** – A full-stack framework that follows convention over configuration.

C#:

- **ASP.NET Core** – Cross-platform, high-performance framework for building web apps with .NET.

Go:

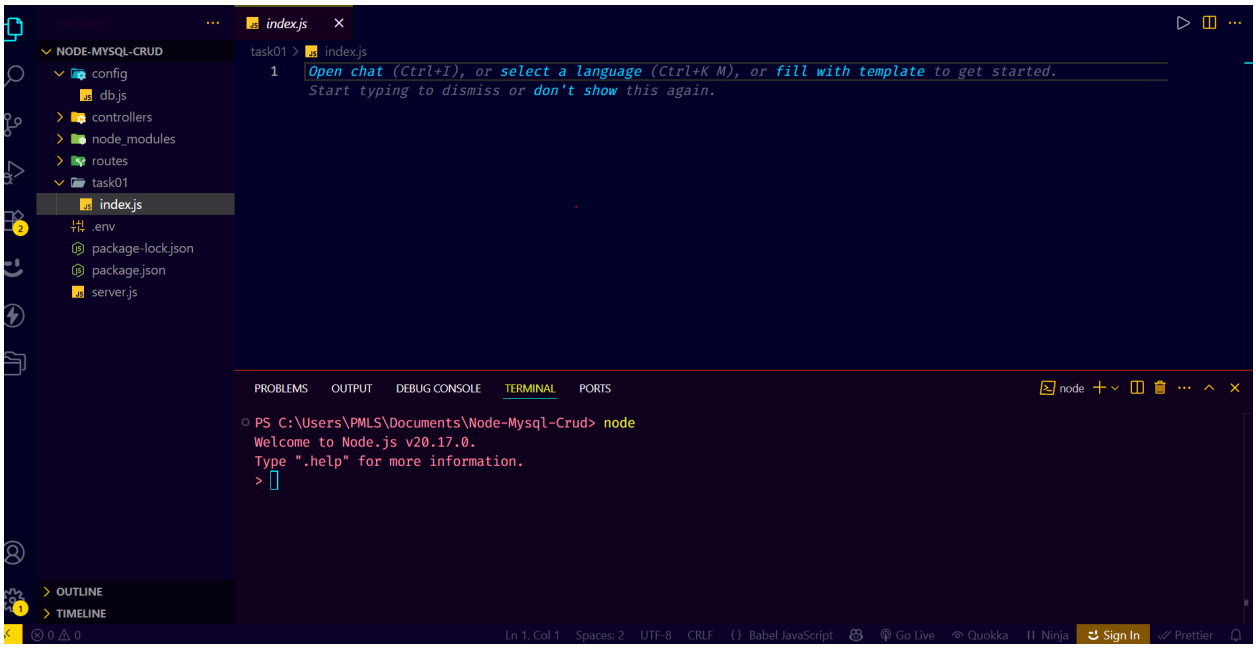
- **Gin** – A lightweight and high-performance Go web framework.
- **Echo** – Another fast, extensible framework for Go.

Conclusion

Back-end development is essential for building robust, secure, and efficient web applications. It works hand-in-hand with front-end development to deliver dynamic and interactive digital experiences. Understanding the languages and frameworks used in the back end can help developers choose the right tools for their projects and ensure seamless communication between user interfaces and server logic.

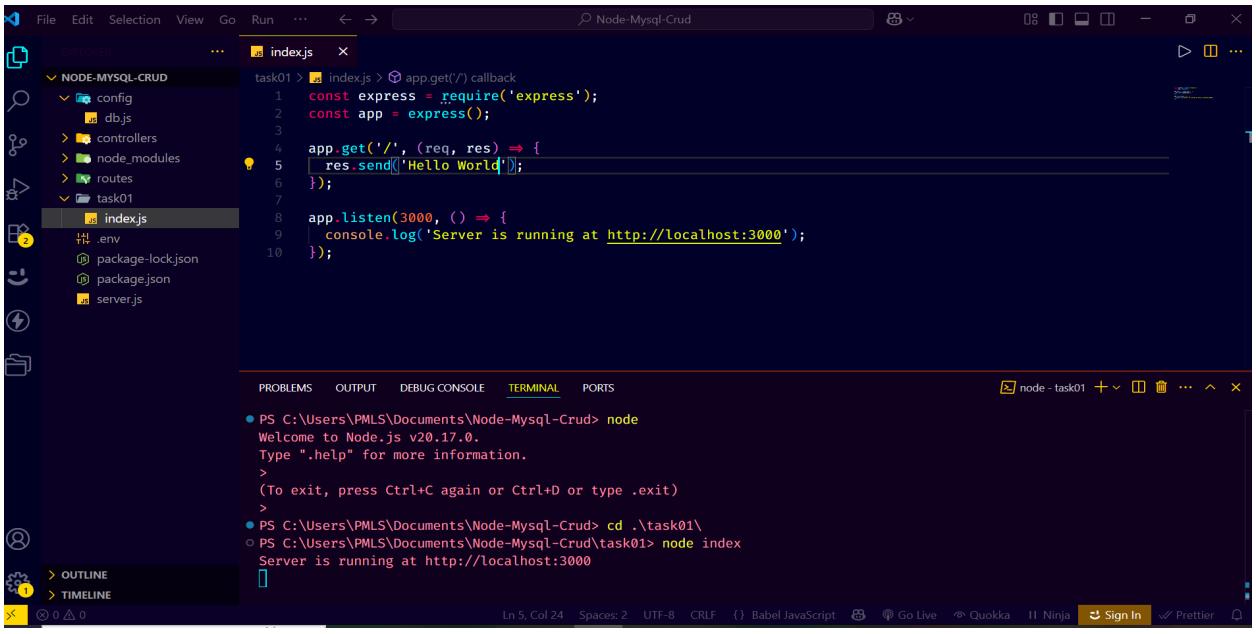
TASK 02

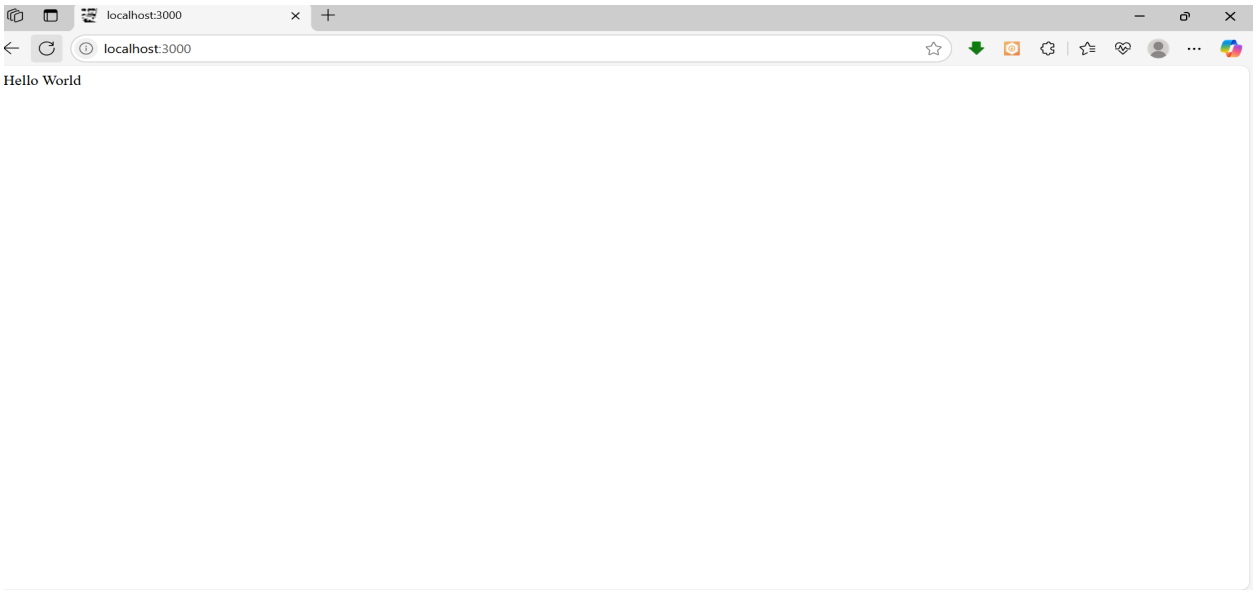
Install **Node.js / Python / Java (pick one stack)** and set up your development environment.



TASK 03

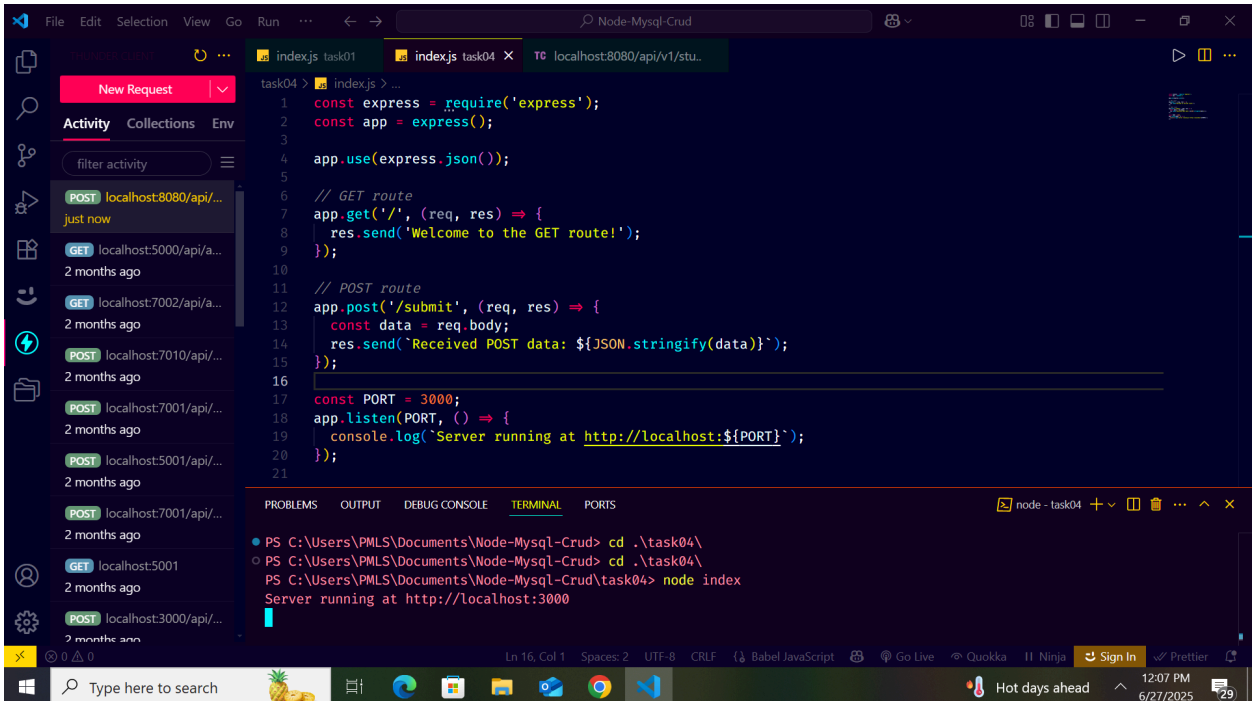
Create your **first "Hello World" server** using Express (Node.js) or Flask (Python) or Spring Boot (Java).

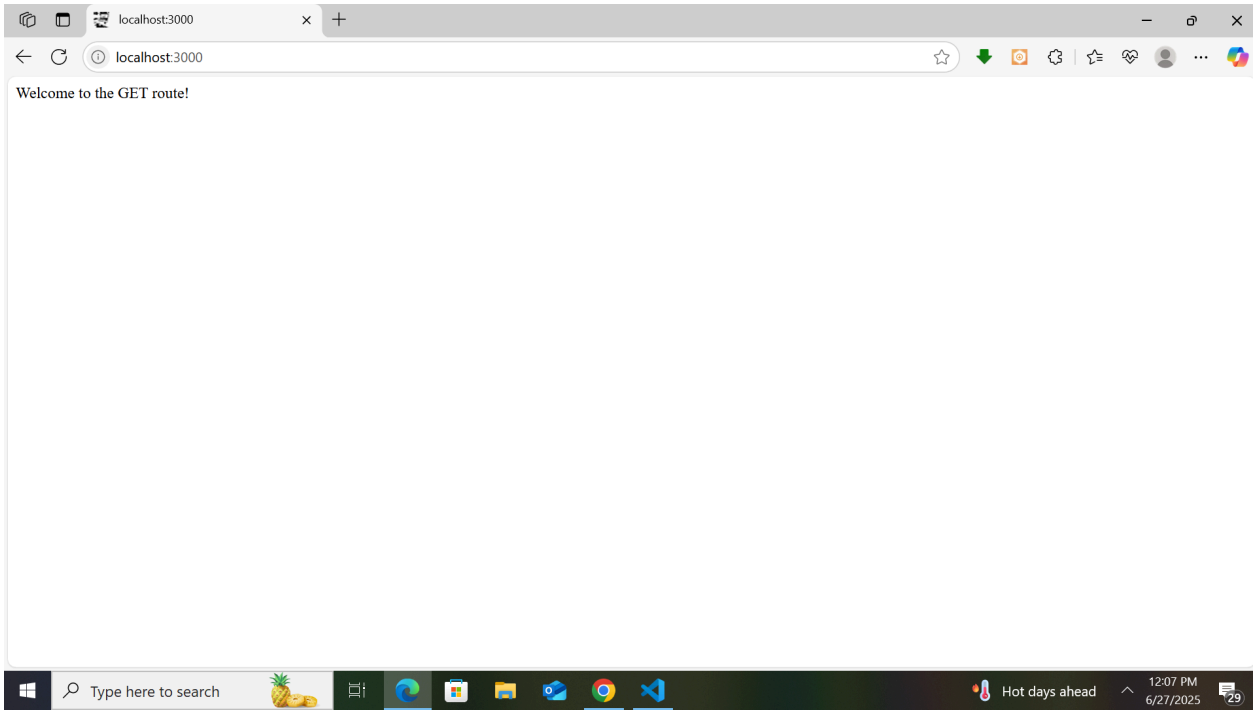


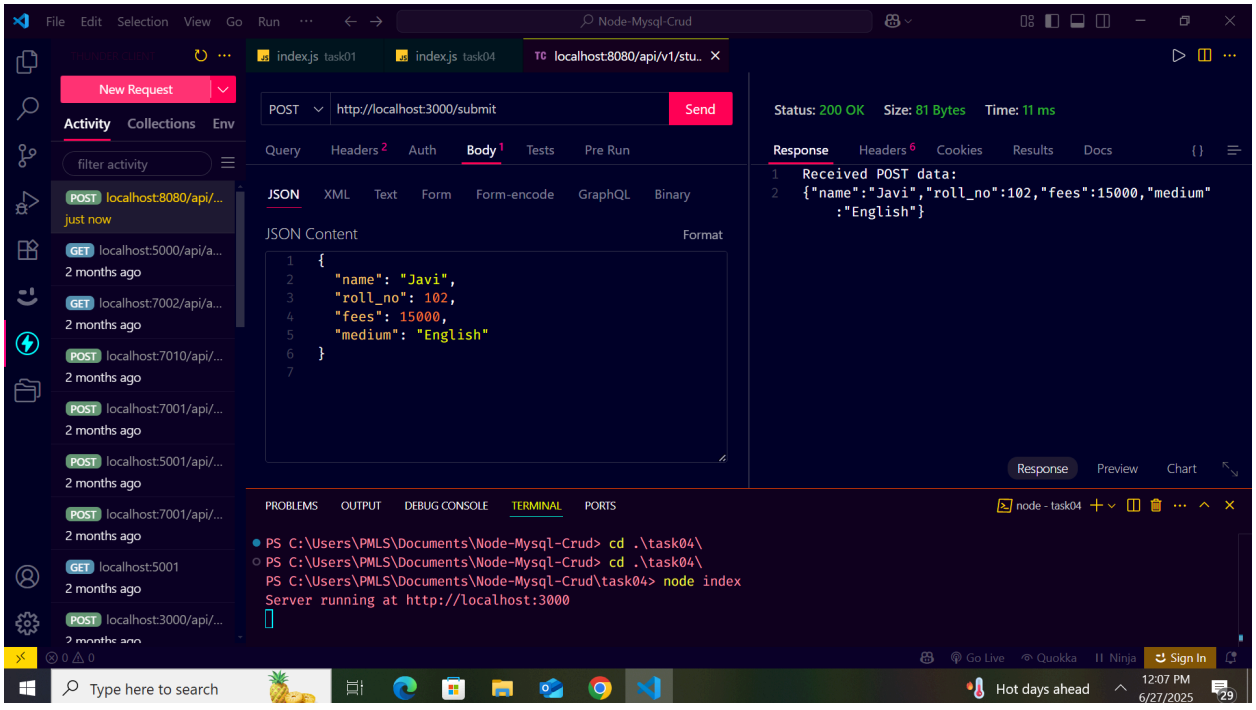


TASK 04

Set up **basic routing (GET, POST)** for your server.

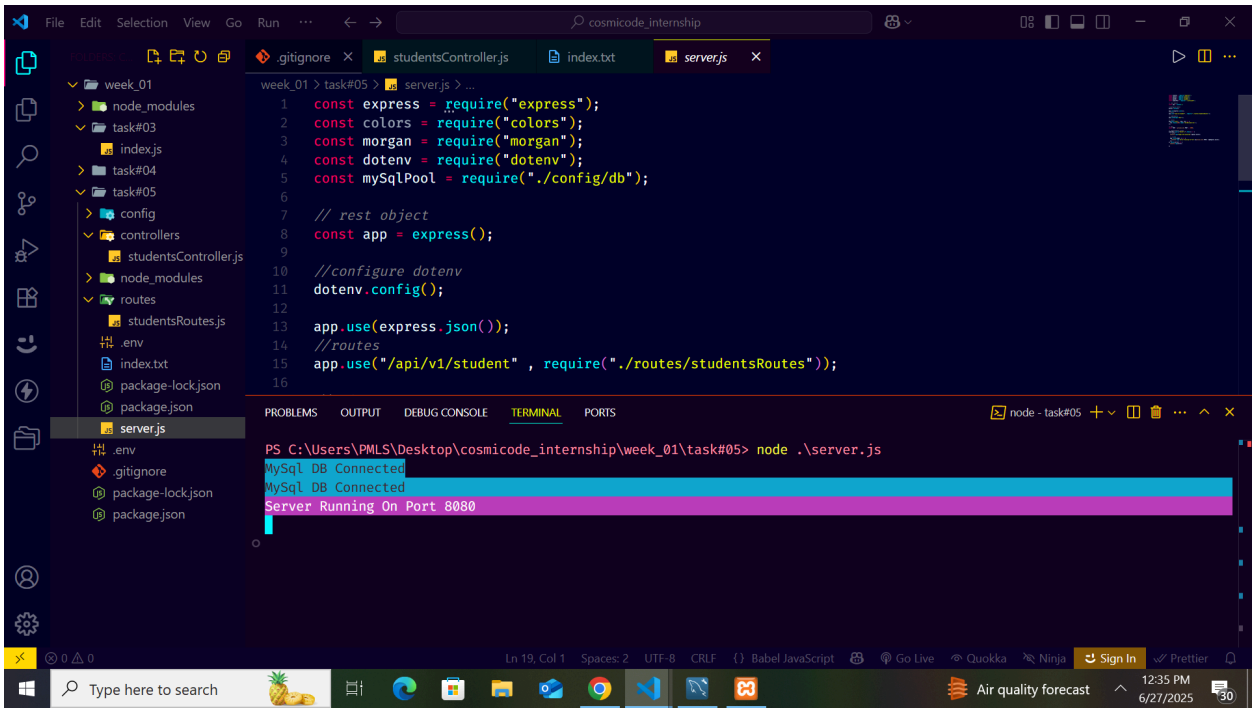


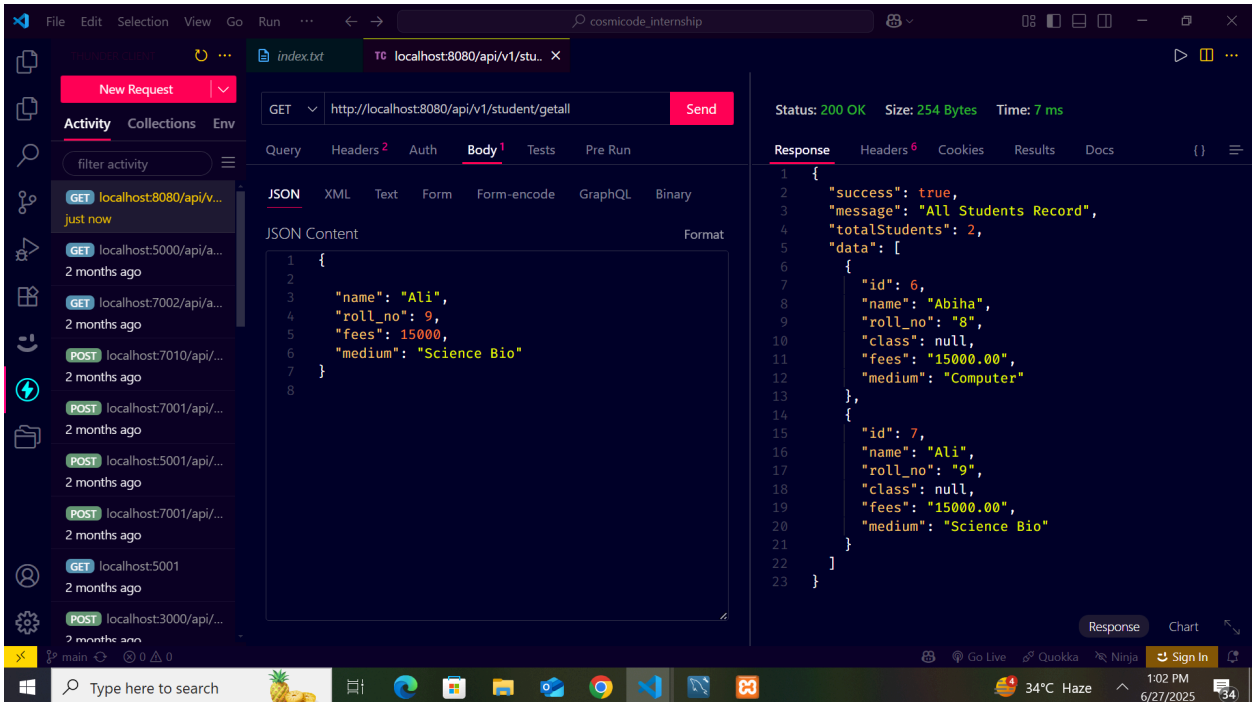

localhost:3000
localhost:3000
Welcome to the GET route!
Type here to search
Hot days ahead
12:07 PM
6/27/2025

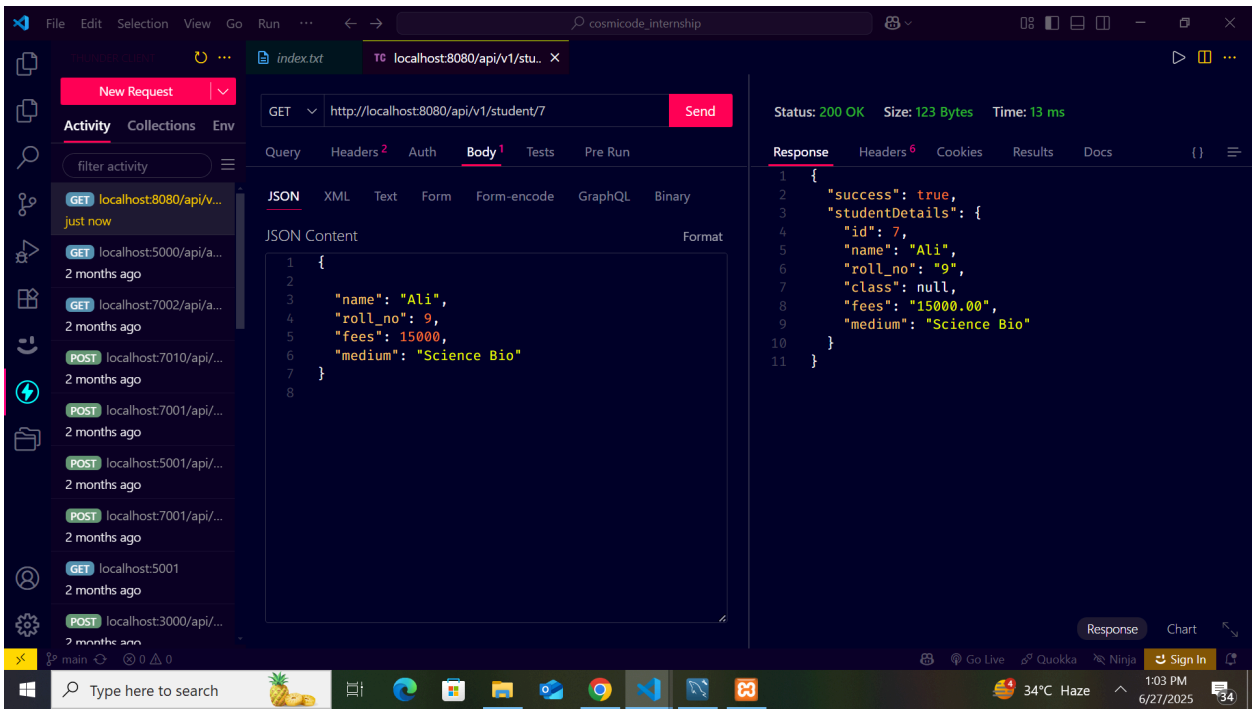

File Edit Selection View Go Run
Node-Mysql-Crud
index.js task01
index.js task04
localhost:8080/api/v1/stu..
New Request
Activity Collections Env
filter activity
POST localhost:8080/api/...
just now
GET localhost:5000/api/a...
2 months ago
GET localhost:7002/api/a...
2 months ago
POST localhost:7010/api/...
2 months ago
POST localhost:7001/api/...
2 months ago
POST localhost:5001/api/...
2 months ago
POST localhost:7001/api/...
2 months ago
GET localhost:5001
2 months ago
POST localhost:3000/api/...
POST http://localhost:3000/submit
Send
Query Headers 2 Auth Body 1 Tests Pre Run
JSON XML Text Form Form-encode GraphQL Binary
JSON Content
Format
{
"name": "Javi",
"roll_no": 102,
"fees": 15000,
"medium": "English"
}
Status: 200 OK Size: 81 Bytes Time: 11 ms
Response Headers 6 Cookies Results Docs
Received POST data:
{"name":"Javi","roll_no":102,"fees":15000,"medium":"English"}
Response Preview Chart
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node - task04
PS C:\Users\PMLS\Documents\Node-Mysql-Crud> cd .\task04\
PS C:\Users\PMLS\Documents\Node-Mysql-Crud> cd .\task04\
PS C:\Users\PMLS\Documents\Node-Mysql-Crud\task04> node index
Server running at http://localhost:3000
Go Live Quokka Ninja Sign In
Type here to search
Hot days ahead
12:07 PM
6/27/2025

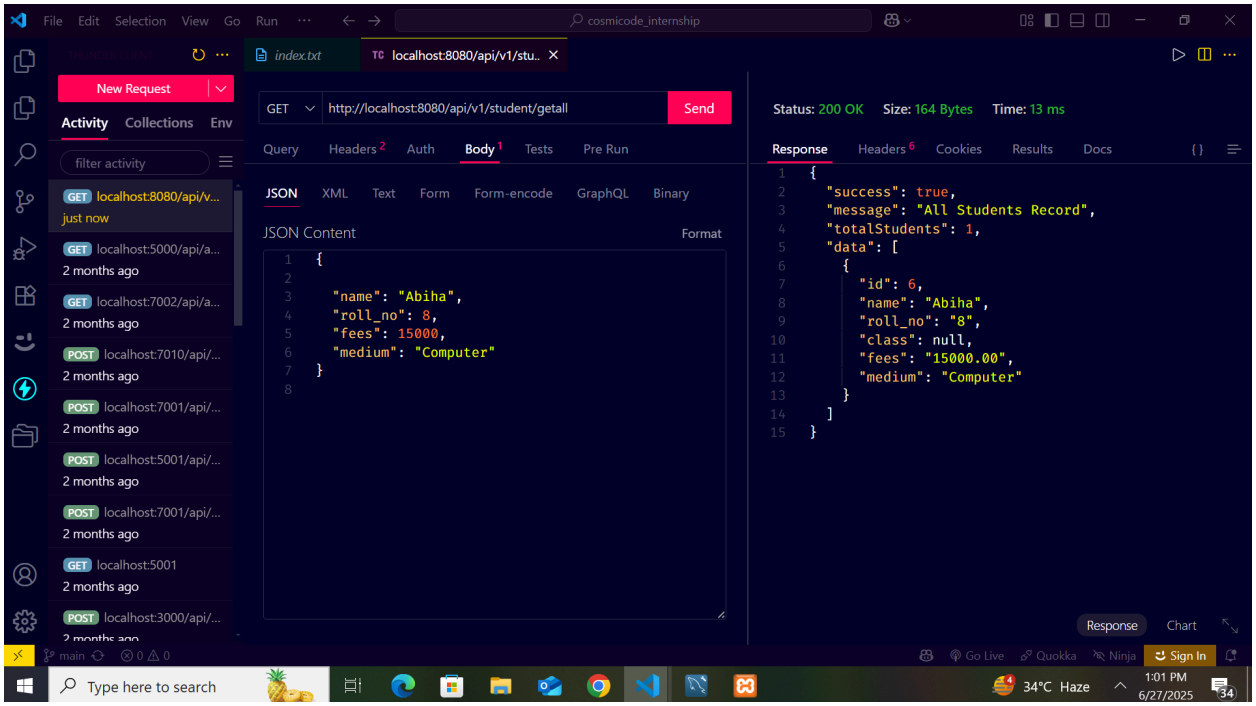
TASK 05

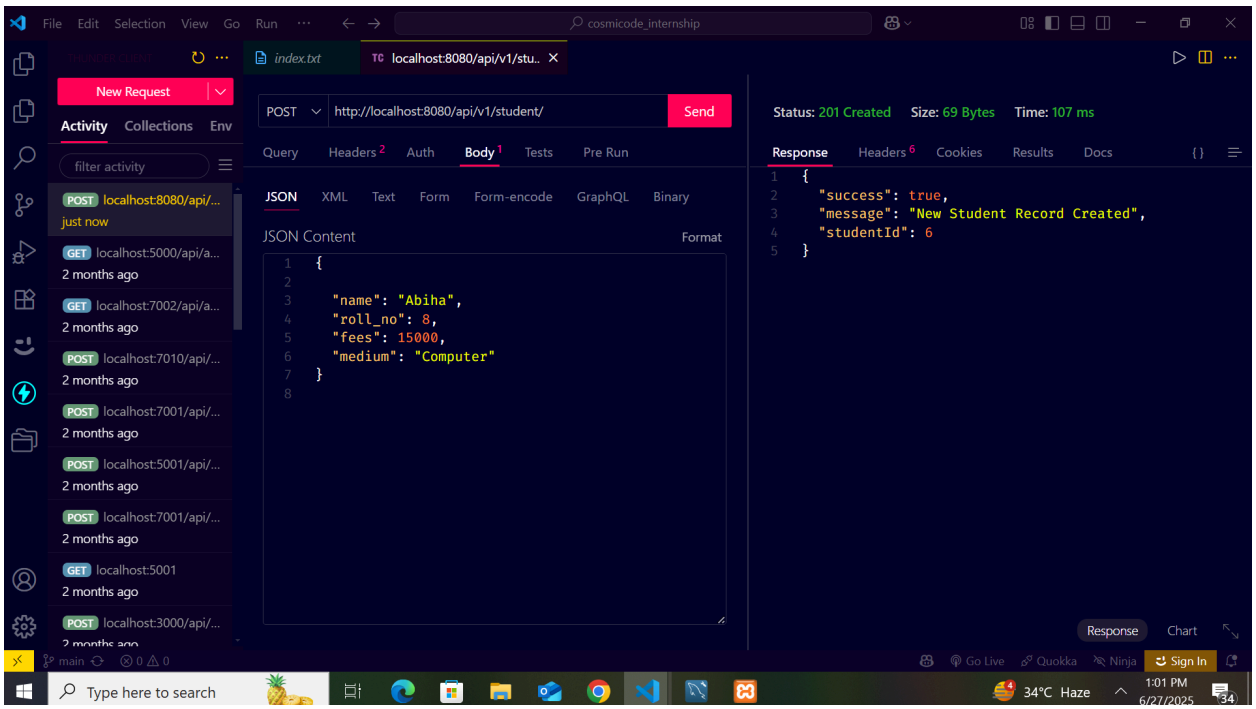
Connect your back-end app to a **local database (SQLite/MySQL/PostgreSQL)** and perform basic CRUD (Create, Read, Update, Delete) operations.

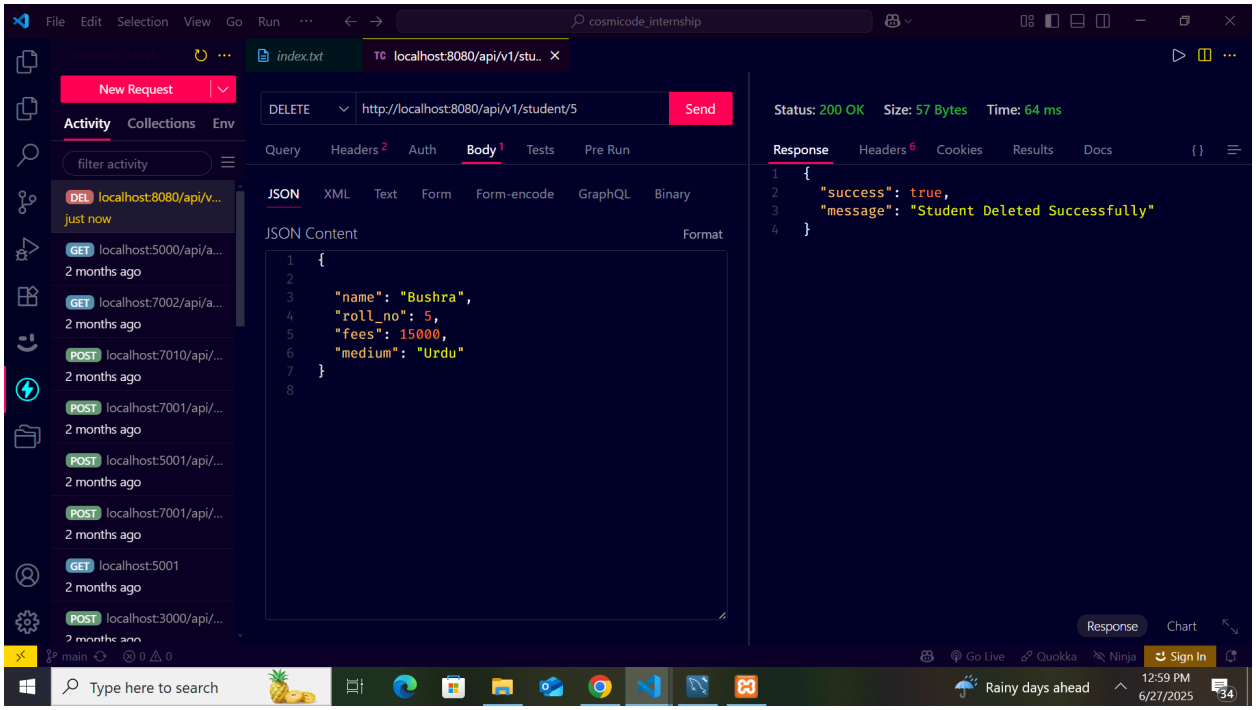



File Edit Selection View Go Run
cosmicode_internship
index.txt
localhost:8080/api/v1/stu..
New Request
Activity Collections Env
filter activity
GET localhost:8080/api/v...
just now
GET localhost:5000/api/a...
2 months ago
GET localhost:7002/api/a...
2 months ago
POST localhost:7010/api/...
2 months ago
POST localhost:7001/api/...
2 months ago
POST localhost:5001/api/...
2 months ago
POST localhost:7001/api/...
2 months ago
GET localhost:5001
2 months ago
POST localhost:3000/api/...
GET
http://localhost:8080/api/v1/student/getall
Send
Query Headers 2 Auth Body 1 Tests Pre Run
JSON XML Text Form Form-encode GraphQL Binary
JSON Content
Format
{
"name": "Ali",
"roll_no": 9,
"fees": 15000,
"medium": "Science Bio"
}
Status: 200 OK Size: 254 Bytes Time: 7 ms
Response Headers 6 Cookies Results Docs
{
"success": true,
"message": "All Students Record",
"totalStudents": 2,
"data": [
{
"id": 6,
"name": "Abiha",
"roll_no": "8",
"class": null,
"fees": "15000.00",
"medium": "Computer"
},
{
"id": 7,
"name": "Ali",
"roll_no": "9",
"class": null,
"fees": "15000.00",
"medium": "Science Bio"
}
]
}
Response Chart
main
Go Live Quokka Ninja Sign In
Type here to search
34°C Haze
1:02 PM
6/27/2025


File Edit Selection View Go Run
cosmicode_internship
index.txt
localhost:8080/api/v1/stu..
New Request
Activity Collections Env
filter activity
GET localhost:8080/api/v...
just now
GET localhost:5000/api/a...
2 months ago
GET localhost:7002/api/a...
2 months ago
POST localhost:7010/api/...
2 months ago
POST localhost:7001/api/...
2 months ago
POST localhost:5001/api/...
2 months ago
POST localhost:7001/api/...
2 months ago
GET localhost:5001
2 months ago
POST localhost:3000/api/...
GET
http://localhost:8080/api/v1/student/7
Send
Query Headers 2 Auth Body 1 Tests Pre Run
JSON XML Text Form Form-encode GraphQL Binary
JSON Content
Format
{
"name": "Ali",
"roll_no": 9,
"fees": 15000,
"medium": "Science Bio"
}
Status: 200 OK Size: 123 Bytes Time: 13 ms
Response Headers 6 Cookies Results Docs
{
"success": true,
"studentDetails": {
"id": 7,
"name": "Ali",
"roll_no": "9",
"class": null,
"fees": "15000.00",
"medium": "Science Bio"
}
}
Response Chart
main
Go Live Quokka Ninja Sign In
Type here to search
34°C Haze
1:03 PM
6/27/2025


File Edit Selection View Go Run
cosmicode_internship
index.txt
localhost:8080/api/v1/stu..
New Request
Activity Collections Env
filter activity
GET localhost:8080/api/v...
just now
GET http://localhost:8080/api/v1/student/getall
Send
Query Headers 2 Auth Body 1 Tests Pre Run
JSON XML Text Form Form-encode GraphQL Binary
JSON Content Format
{
"name": "Abiha",
"roll_no": 8,
"fees": 15000,
"medium": "Computer"
}
Status: 200 OK Size: 164 Bytes Time: 13 ms
Response Headers 6 Cookies Results Docs
{
"success": true,
"message": "All Students Record",
"totalStudents": 1,
"data": [
{
"id": 6,
"name": "Abiha",
"roll_no": "8",
"class": null,
"fees": "15000.00",
"medium": "Computer"
}
]
}
Response Chart
Type here to search
34°C Haze
1:01 PM
6/27/2025


File Edit Selection View Go Run
cosmicode_internship
index.txt
localhost:8080/api/v1/stu..
New Request
Activity Collections Env
filter activity
POST localhost:8080/api/...
just now
POST http://localhost:8080/api/v1/student/
Send
Query Headers 2 Auth Body 1 Tests Pre Run
JSON XML Text Form Form-encode GraphQL Binary
JSON Content Format
{
"name": "Abiha",
"roll_no": 8,
"fees": 15000,
"medium": "Computer"
}
Status: 201 Created Size: 69 Bytes Time: 107 ms
Response Headers 6 Cookies Results Docs
{
"success": true,
"message": "New Student Record Created",
"studentId": 6
}
Response Chart
Type here to search
34°C Haze
1:01 PM
6/27/2025


File Edit Selection View Go Run
cosmicode_internship
index.txt
localhost:8080/api/v1/stu..
New Request
Activity Collections Env
filter activity
DEL localhost:8080/api/v...
just now
GET localhost:5000/api/a...
2 months ago
GET localhost:7002/api/a...
2 months ago
POST localhost:7010/api/...
2 months ago
POST localhost:7001/api/...
2 months ago
POST localhost:5001/api/...
2 months ago
POST localhost:7001/api/...
2 months ago
GET localhost:5001
2 months ago
POST localhost:3000/api/...
DELETE
http://localhost:8080/api/v1/student/5
Send
Query Headers 2 Auth Body 1 Tests Pre Run
JSON XML Text Form Form-encode GraphQL Binary
JSON Content
Format
{
"name": "Bushra",
"roll_no": 5,
"fees": 15000,
"medium": "Urdu"
}
Status: 200 OK Size: 57 Bytes Time: 64 ms
Response Headers 6 Cookies Results Docs
{
"success": true,
"message": "Student Deleted Successfully"
}
Response Chart
main
Go Live Quokka Ninja Sign In
Type here to search
Rainy days ahead
12:59 PM
6/27/2025