COLLEGE NAME: MEENAKSHI COLLEGE OF ENGINEERING

COLLEGE CODE: 3114

YOUR NAME: ARIVANANTHAM.J

NAAN MUDHALVAN (NM) ID:
C243912FBA9215698764EA53EB29B62B

PHONE NO: 8807194452

EMAIL ID: nandhamarivu06@gmail.com

GIT HUB LINK: https://github.com/Arivu611/crud.git

COLLEGE COUNSELING MANAGEMENT SYSTEM

1. INTRODUCTION

The College Counseling System is a web-based application developed to help manage the college counseling process in an organized and efficient manner.

This system allows administrators to store, update, view, and delete student information digitally.

It reduces manual work and improves accuracy in handling student data during counseling.

2. PROBLEM STATEMENT

Traditional college counseling is mostly done manually, which leads to:

Data duplication

Human errors

Time consumption

Difficulty in managing large student records

Hence, there is a need for an automated system to manage the counseling process efficiently.

3. OBJECTIVES OF THE PROJECT

The main objectives of the College Counseling System are:

To automate the student counseling process

To store student data securely

To perform fast data retrieval

To reduce manual errors

To provide a structured counseling workflow

4. SCOPE OF THE PROJECT

The scope of this project includes:

Managing student details

Performing CRUD operations (Create, Read, Update, Delete)

Providing API-based backend support

Testing APIs using Postman

This system can be enhanced in the future by:

Adding a frontend user interface

Connecting to a database

Implementing real-time counseling logic

5. SYSTEM REQUIREMENTS

5.1 Hardware Requirements

Computer or Laptop

Minimum 4 GB RAM

Internet connection (optional)

5.2 Software Requirements

Operating System: Windows / Linux

Node.js

Visual Studio Code

Postman

Web Browser

6. TECHNOLOGIES USED

Programming Language: JavaScript

Backend Runtime: Node.js

Framework: Express.js

API Testing Tool: Postman

Development Tool: Visual Studio Code

7. SYSTEM ARCHITECTURE

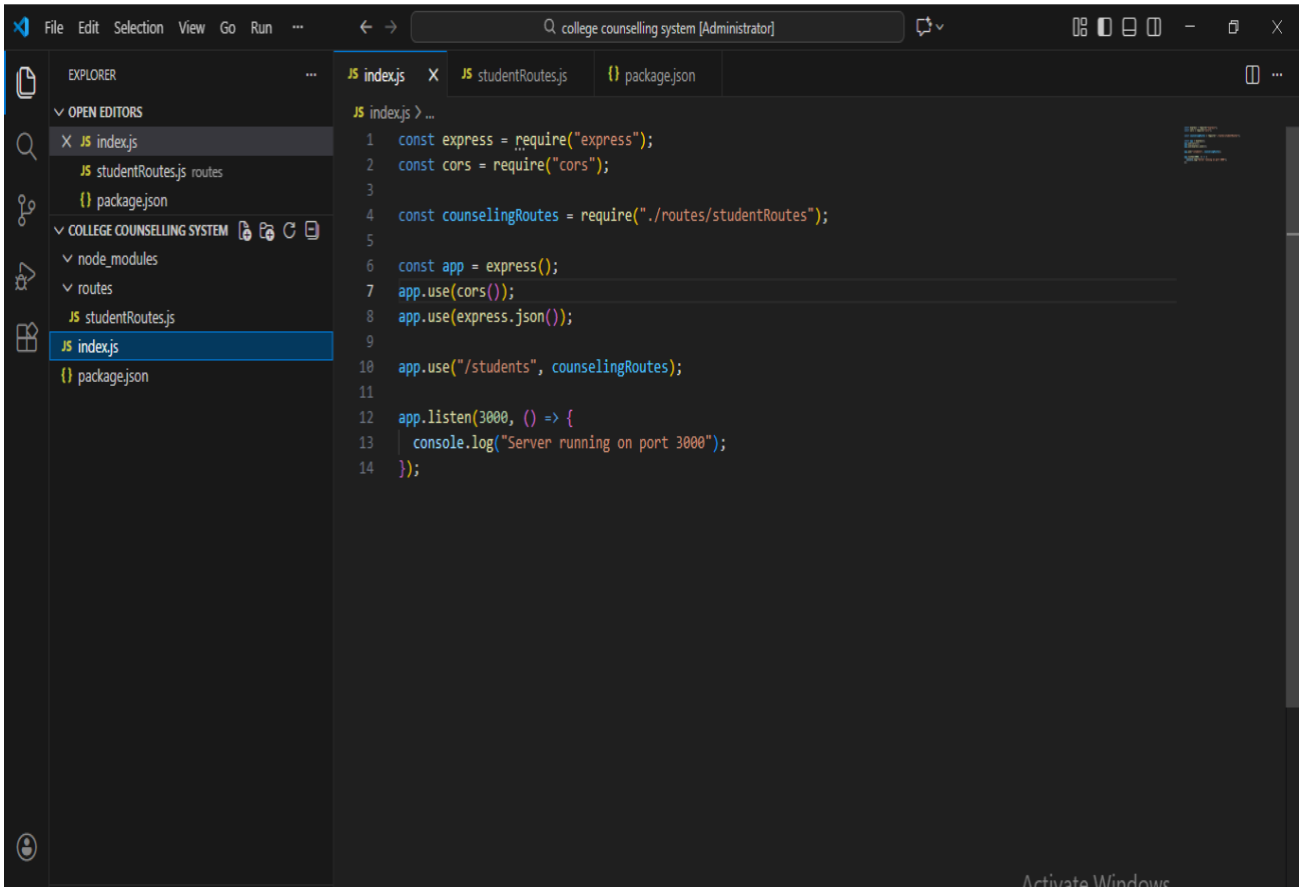
The College Counseling System follows a client-server architecture.

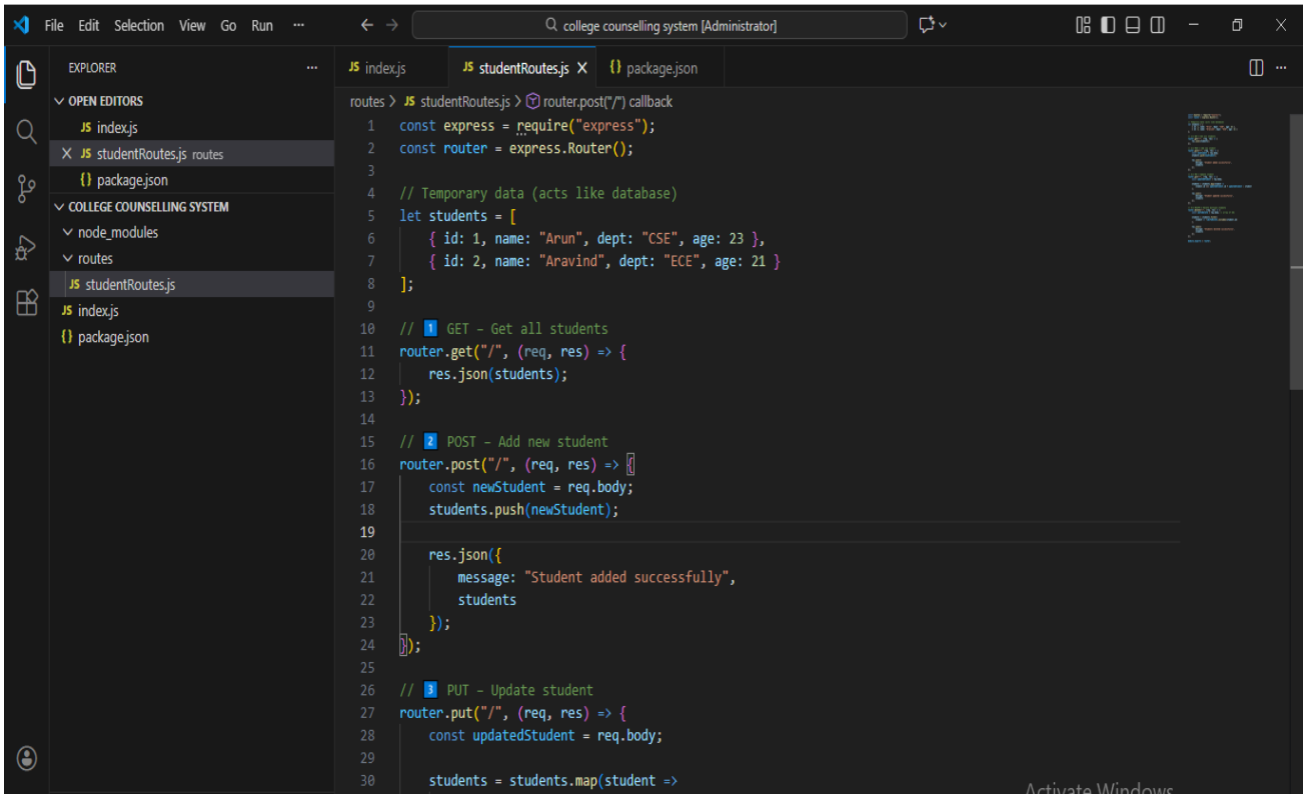
The client sends HTTP requests.

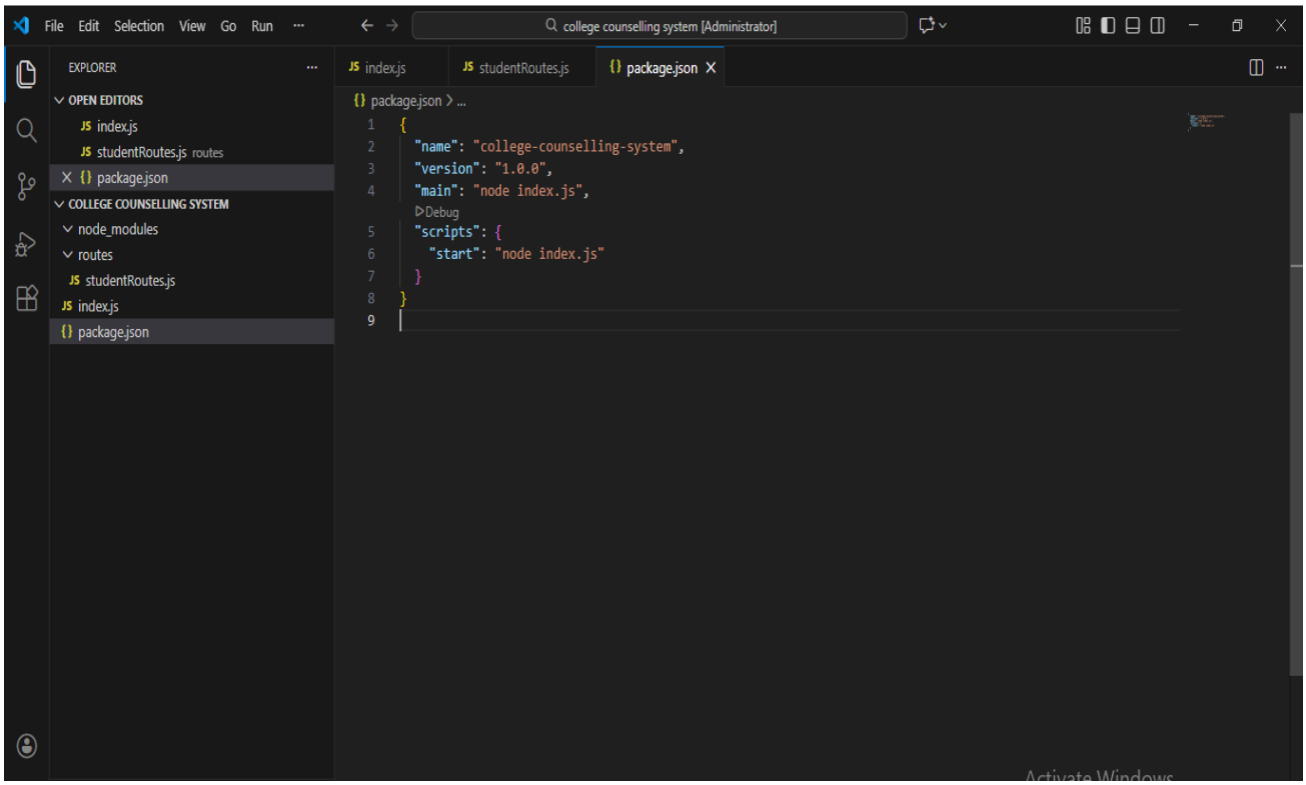
The server processes the requests.

The server returns responses in JSON format.

The system uses REST APIs to communicate between client and server.




File Edit Selection View Go Run
college counselling system [Administrator]
EXPLORER
OPEN EDITORS
index.js
studentRoutes.js routes
package.json
COLLEGE COUNSELLING SYSTEM
node_modules
routes
studentRoutes.js
index.js
package.json
routes > studentRoutes.js > router.post("/") callback
const express = require("express");
const router = express.Router();
// Temporary data (acts like database)
let students = [
{ id: 1, name: "Arun", dept: "CSE", age: 23 },
{ id: 2, name: "Aravind", dept: "ECE", age: 21 }
];
// 1 GET - Get all students
router.get("/", (req, res) => {
res.json(students);
});
// 2 POST - Add new student
router.post("/", (req, res) => {
const newStudent = req.body;
students.push(newStudent);
res.json({
message: "Student added successfully",
students
});
});
// 3 PUT - Update student
router.put("/", (req, res) => {
const updatedStudent = req.body;
students = students.map(student =>


File Edit Selection View Go Run
college counselling system [Administrator]
EXPLORER
OPEN EDITORS
index.js
studentRoutes.js routes
package.json
COLLEGE COUNSELLING SYSTEM
node_modules
routes
studentRoutes.js
index.js
package.json
package.json > ...
{
"name": "college-counselling-system",
"version": "1.0.0",
"main": "node index.js",
Debug
"scripts": {
"start": "node index.js"
}
}

8. MODULE DESCRIPTION

Student Management Module

This module handles all student-related operations:

Add student details

View student list

Update student information

Delete student records

9. WORKING PRINCIPLE

The server is started using Node.js.

APIs are created using Express.js.

Postman is used to send requests to the server.

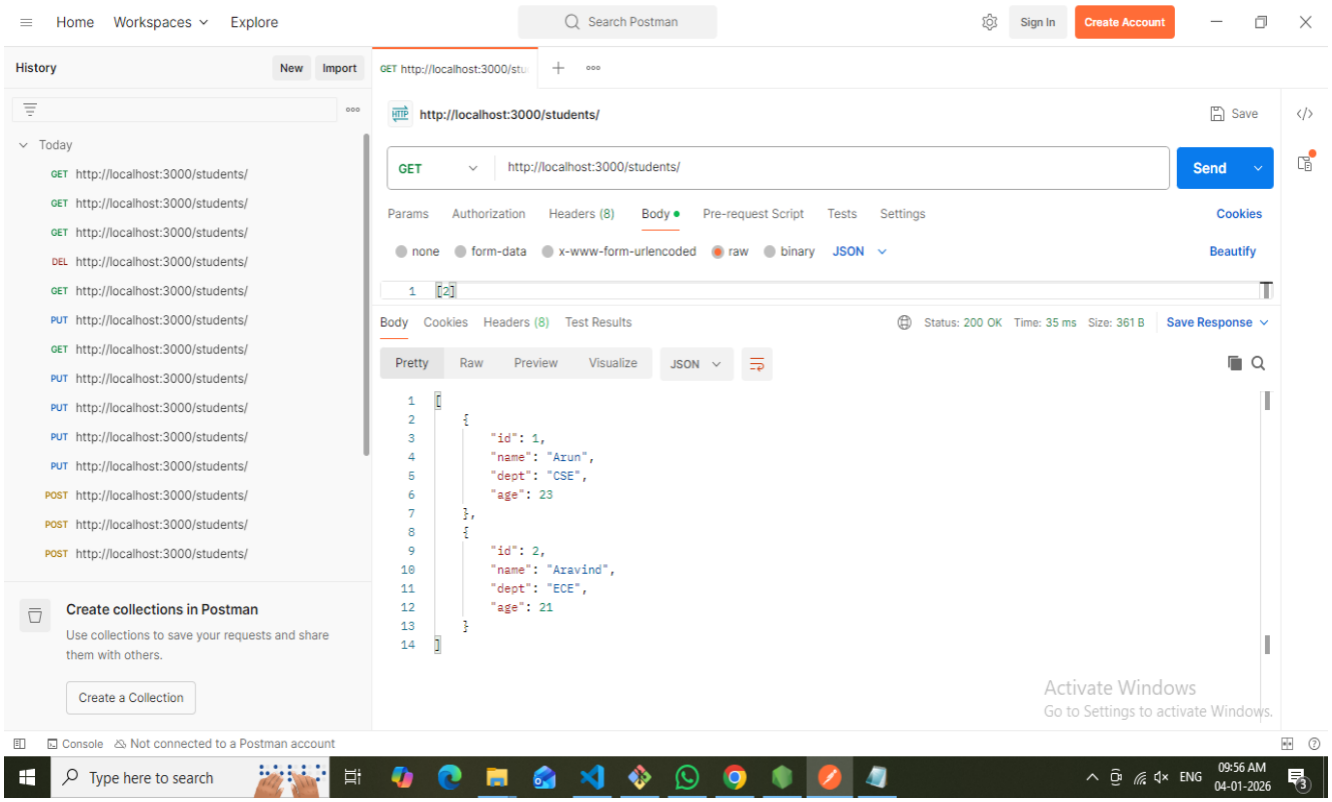
The server processes the request.

The response is sent back in JSON format.

10. API OPERATIONS

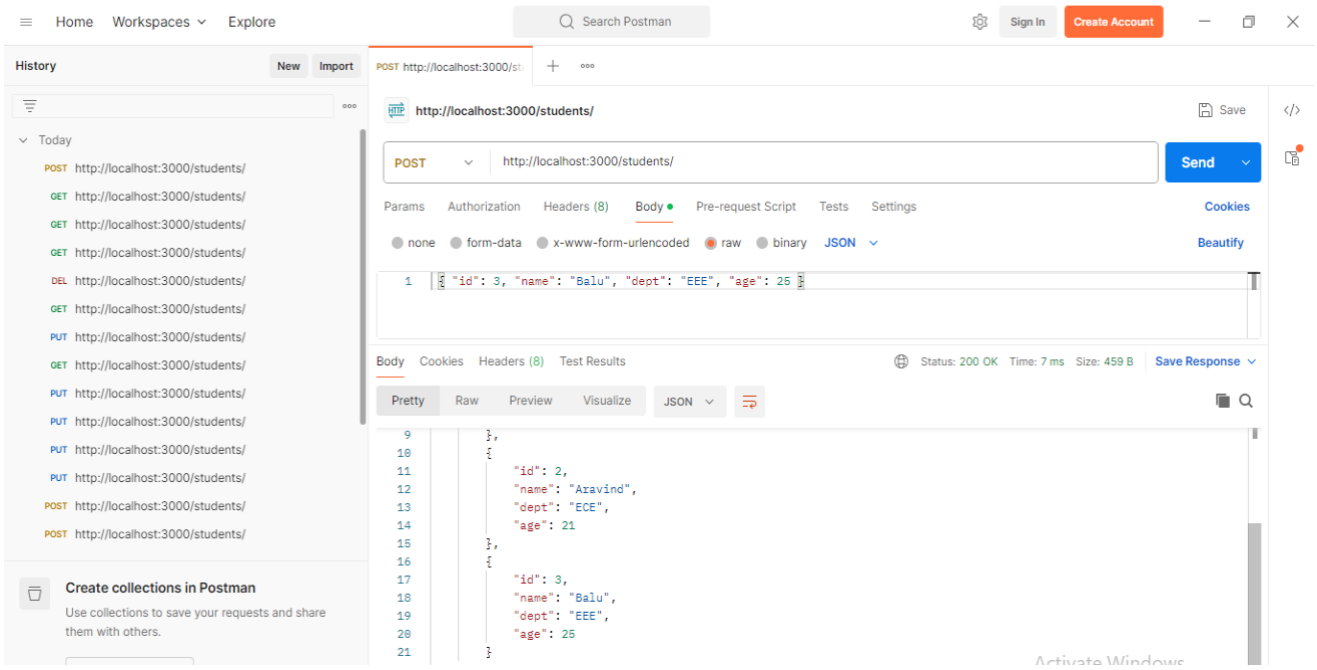
GET Operation

Used to fetch student details



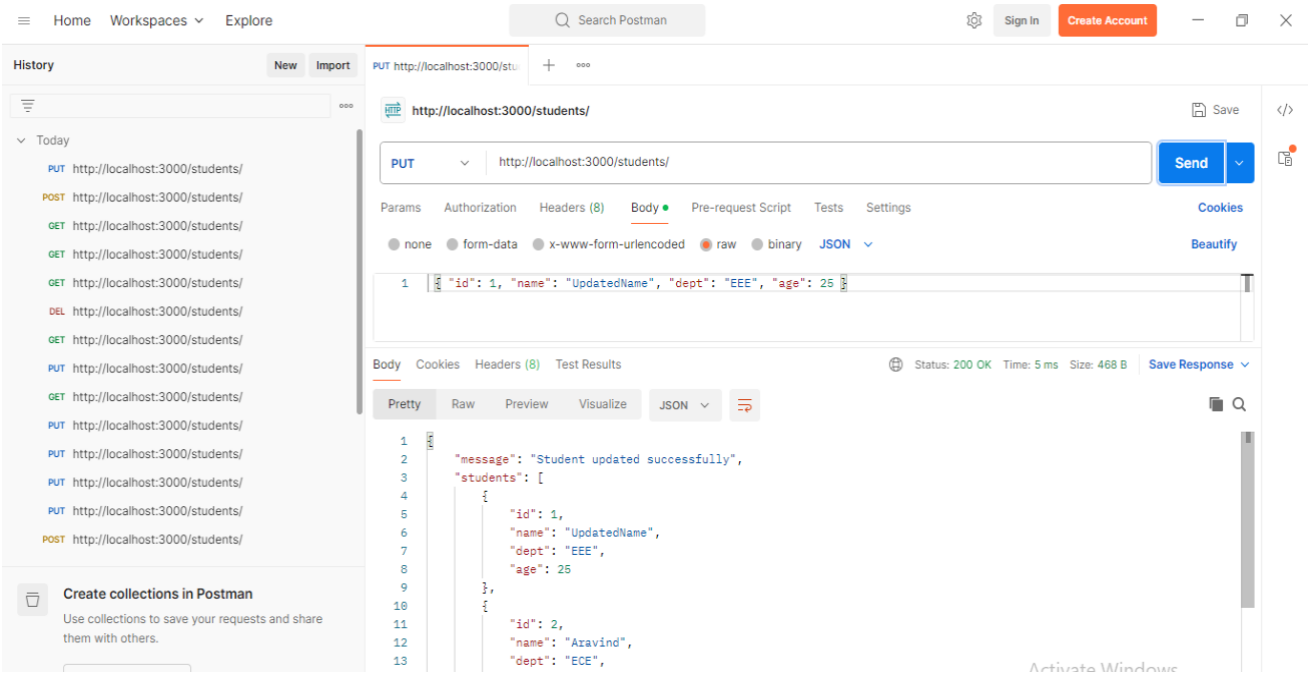
POST Operation

Used to add new student information



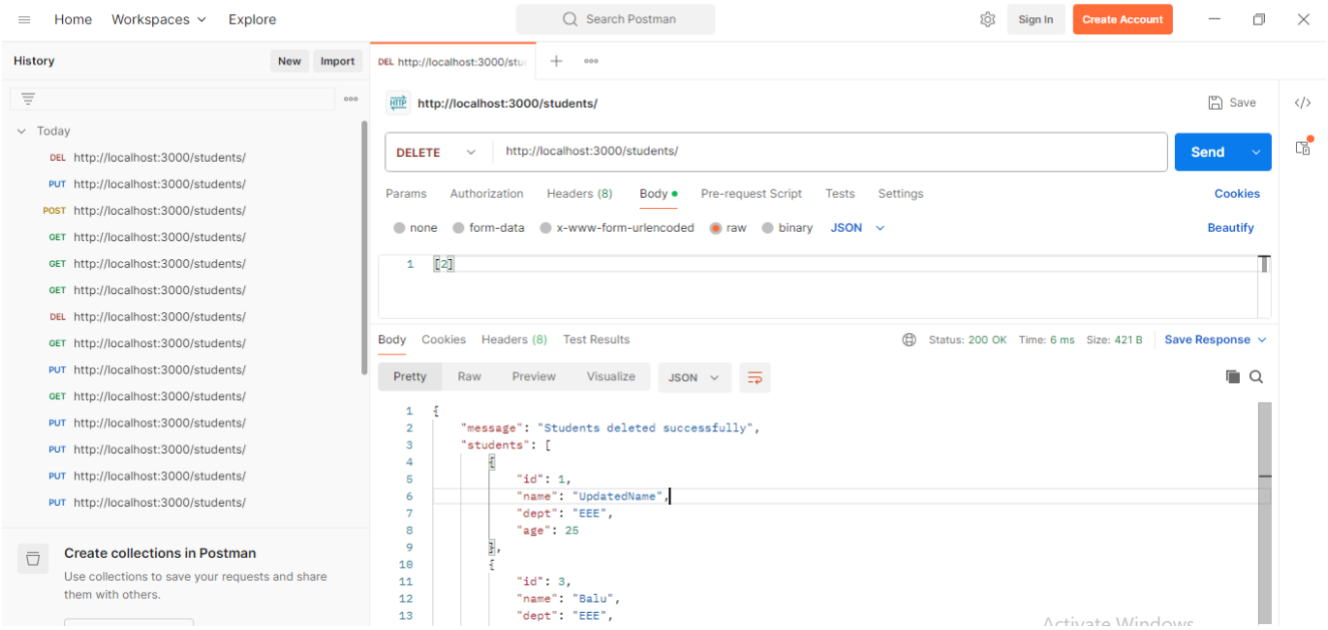
PUT Operation

Used to update existing student data



DELETE Operation

Used to delete student records



11. ADVANTAGES OF THE SYSTEM

Easy to use

Saves time

Reduces manual errors

Organized data handling

Scalable for future enhancement

12. LIMITATIONS

No frontend UI

No permanent database

Limited to backend operations

13. FUTURE ENHANCEMENTS

Database integration (MySQL / Mongo DB)

User authentication

Frontend interface

College cutoff logic

Role-based access (Admin / Student)

14. CONCLUSION

The College Counseling System provides an efficient solution for managing student counseling data.

It simplifies the counseling process and reduces manual effort.

This project demonstrates the practical implementation of REST APIs using Node.js and Express.

15. REFERENCES

Node.js Official Documentation

Express.js Documentation

Postman Learning Center