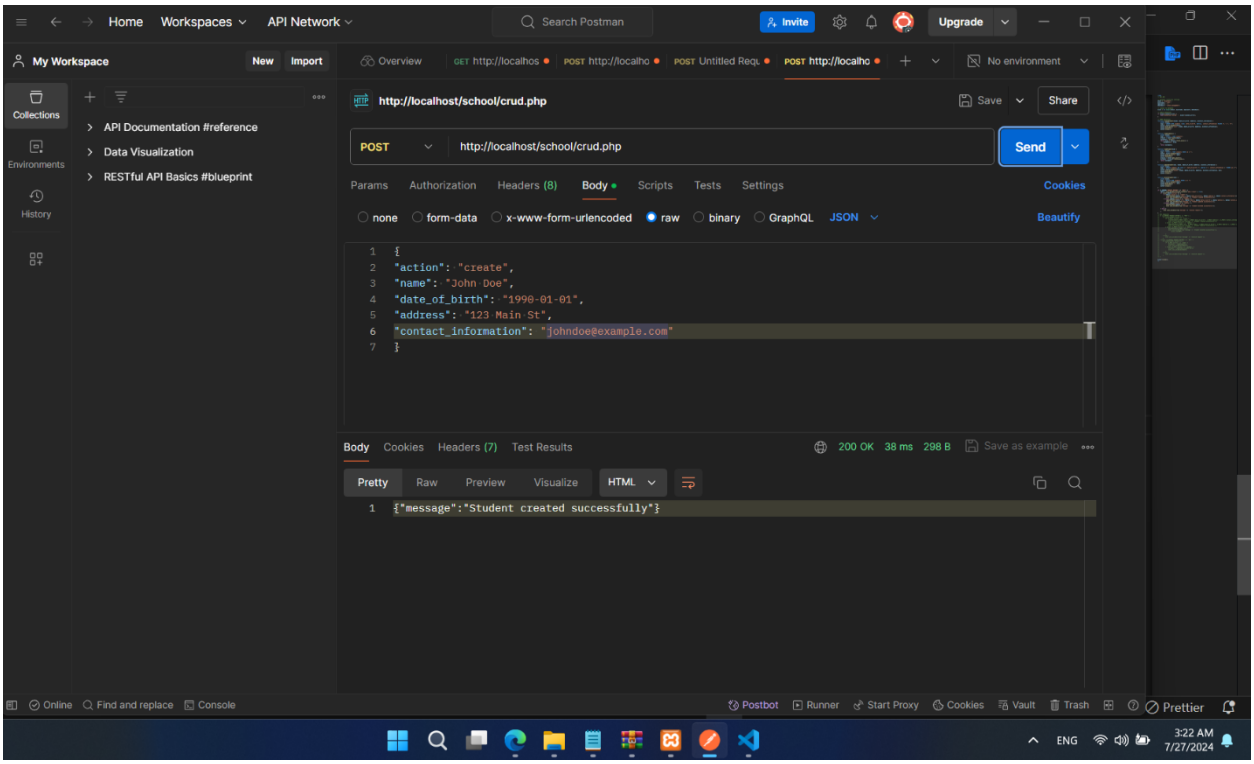
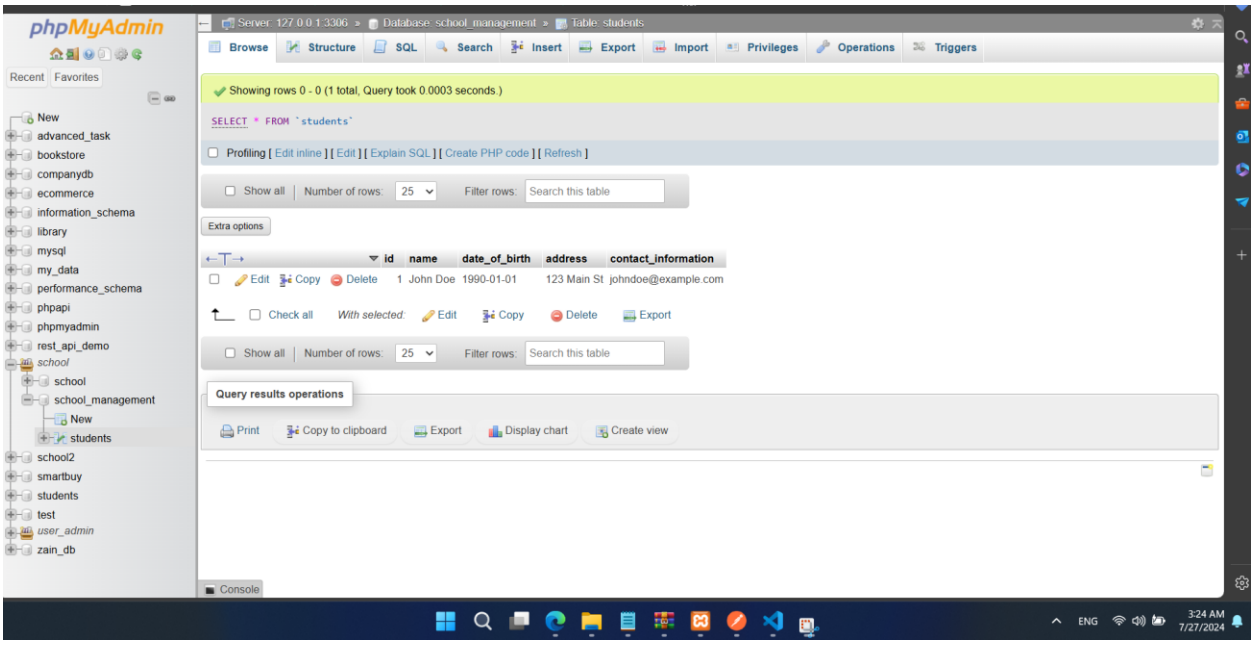
Task 1: School Management System CRUD Operations and API Development

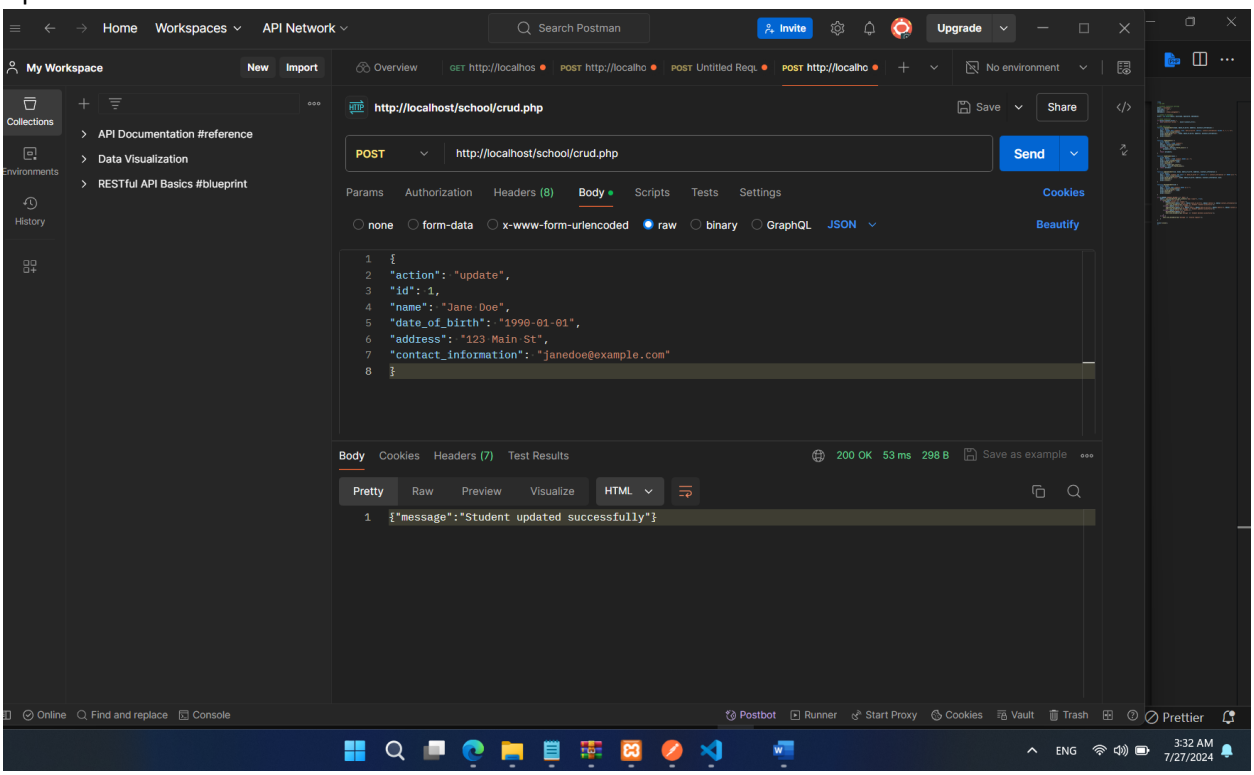
use postman to test each CRUD operation:

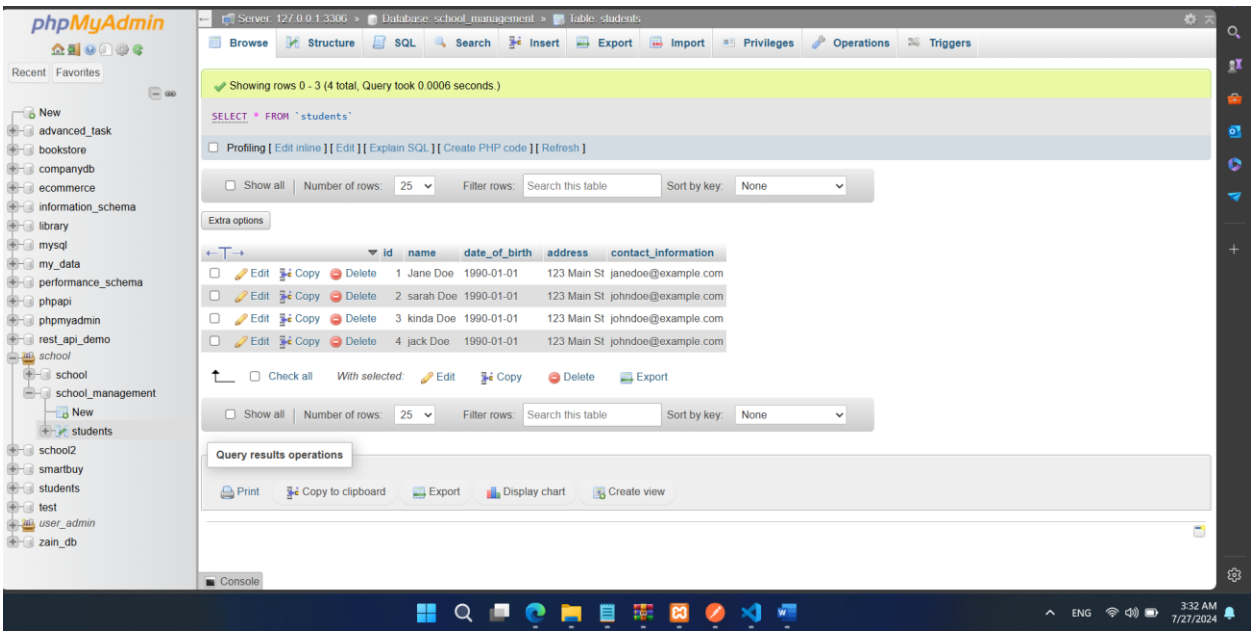
1-creat:



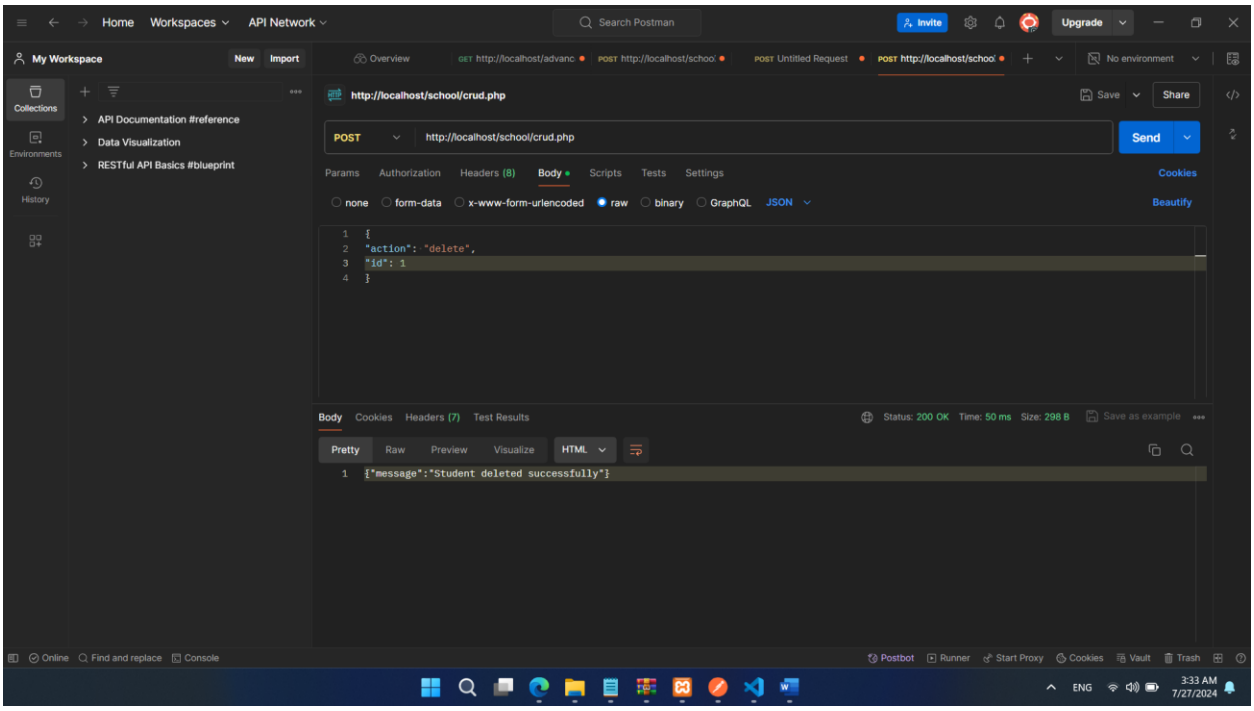


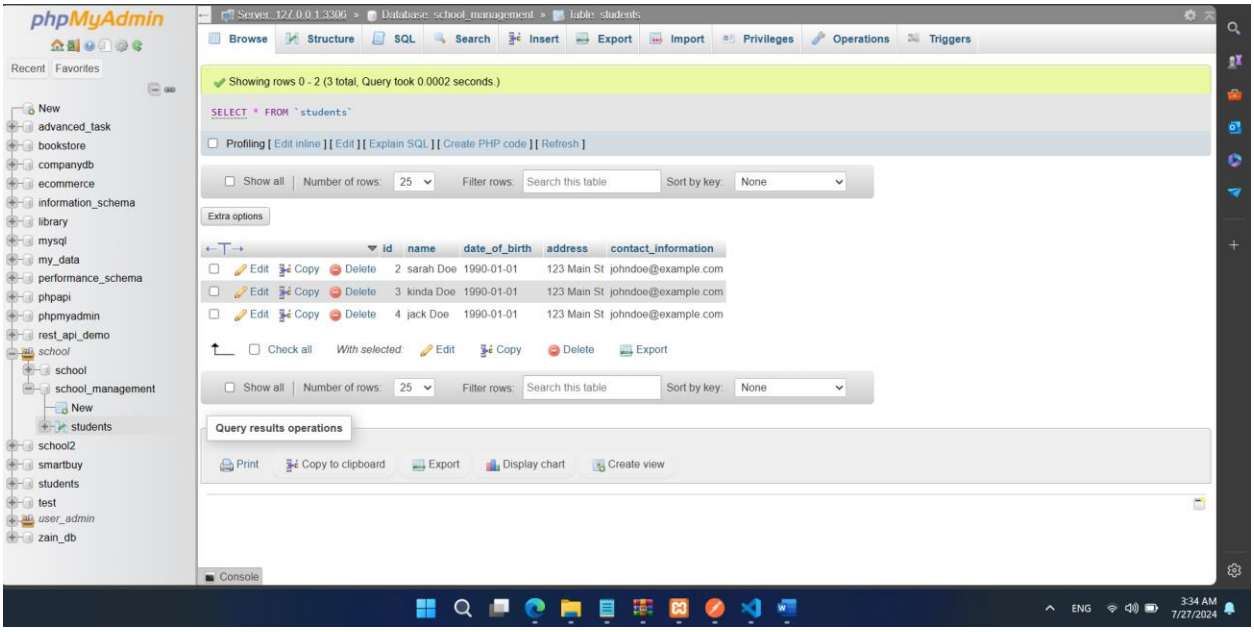
Update:





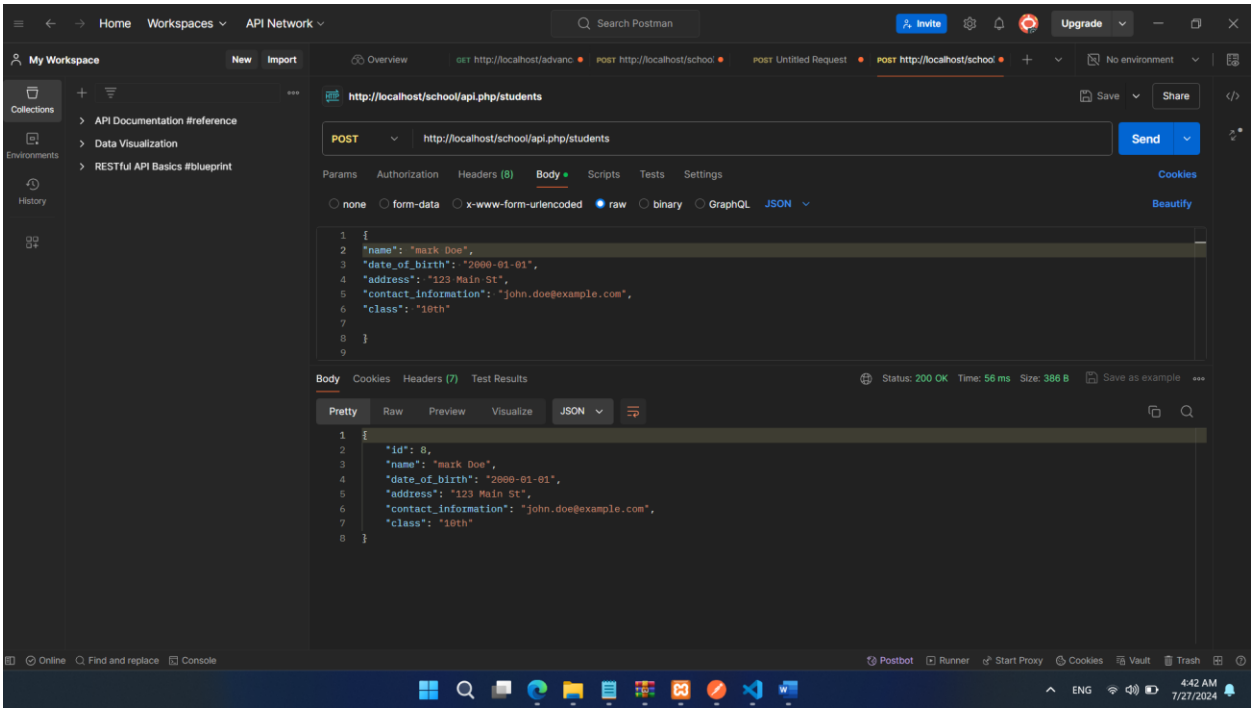
delete:

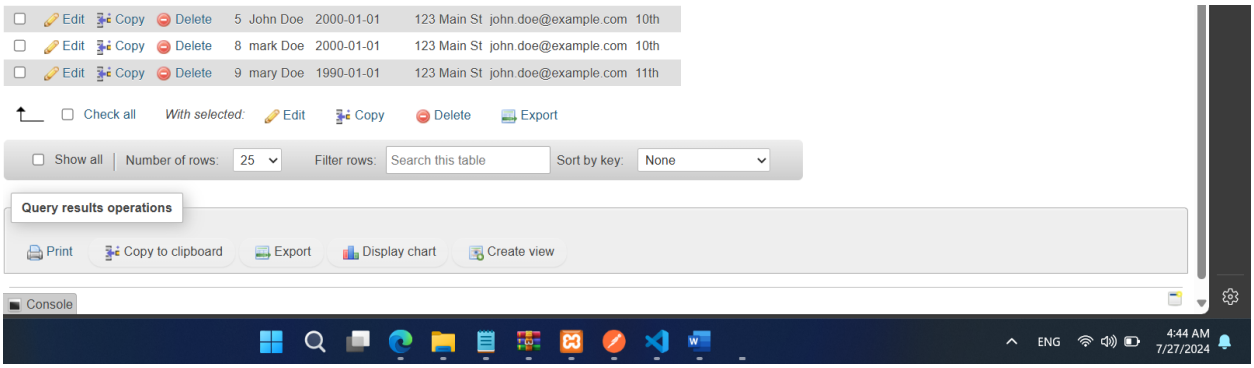

Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
History
API Documentation #reference
Data Visualization
RESTful API Basics #blueprint
Overview
GET http://localhost/advanc
POST http://localhost/school
POST Untitled Request
POST http://localhost/school
No environment
http://localhost/school/crud.php
Save
Share
POST
http://localhost/school/crud.php
Send
Params
Authorization
Headers (8)
Body
Scripts
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"action": "delete",
"id": 1
}
Body
Cookies
Headers (7)
Test Results
Status: 200 OK
Time: 50 ms
Size: 298 B
Save as example
Pretty
Raw
Preview
Visualize
HTML
{"message":"Student deleted successfully"}
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash
ENG
3:33 AM
7/27/2024


phpMyAdmin
Recent
Favorites
New
advanced_task
bookstore
companydb
ecommerce
information_schema
library
mysql
my_data
performance_schema
phpapi
phpmyadmin
rest_api_demo
school
school
school_management
New
students
school2
smartbuy
students
test
user_admin
zain_db
Server: 127.0.0.1:3306 » Database: school_management » Table: students
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Triggers
Showing rows 0 - 2 (3 total, Query took 0.0002 seconds.)
SELECT * FROM `students`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
Extra options
id
name
date_of_birth
address
contact_information
Edit
Copy
Delete
2 sarah Doe 1990-01-01 123 Main St johndoe@example.com
3 kinda Doe 1990-01-01 123 Main St johndoe@example.com
4 jack Doe 1990-01-01 123 Main St johndoe@example.com
Check all
With selected:
Export
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Console
ENG
3:34 AM
7/27/2024

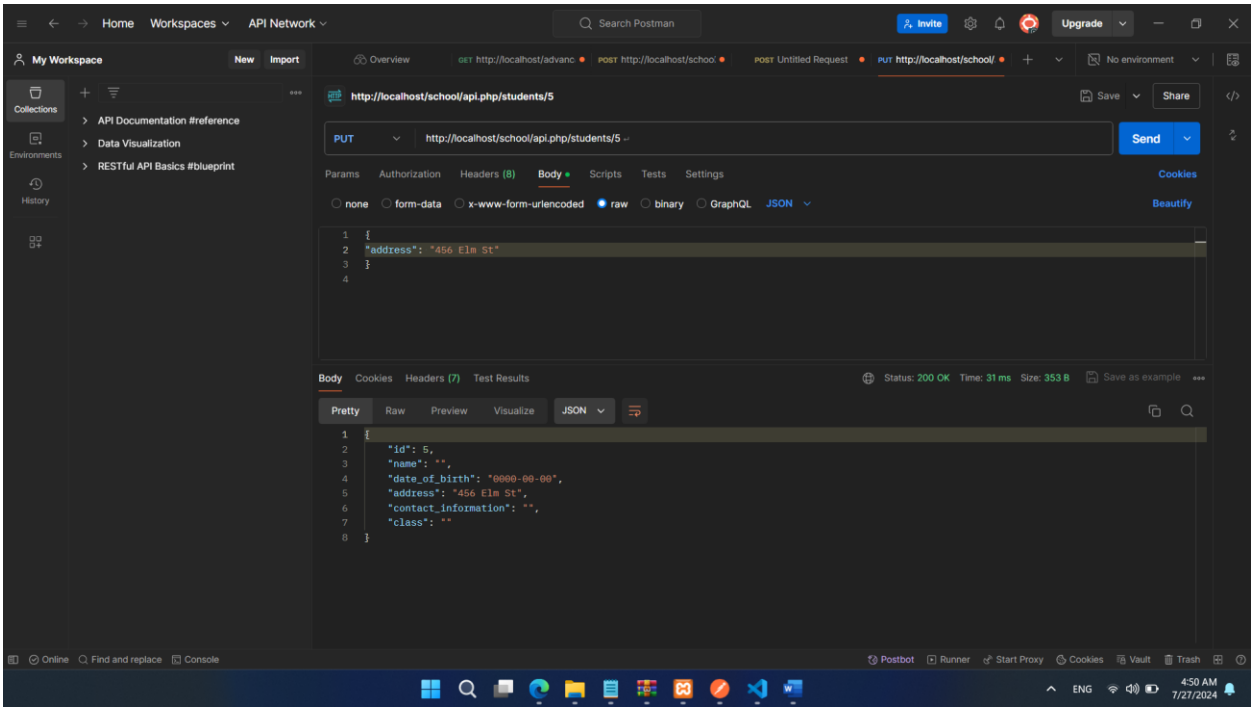
Task2:

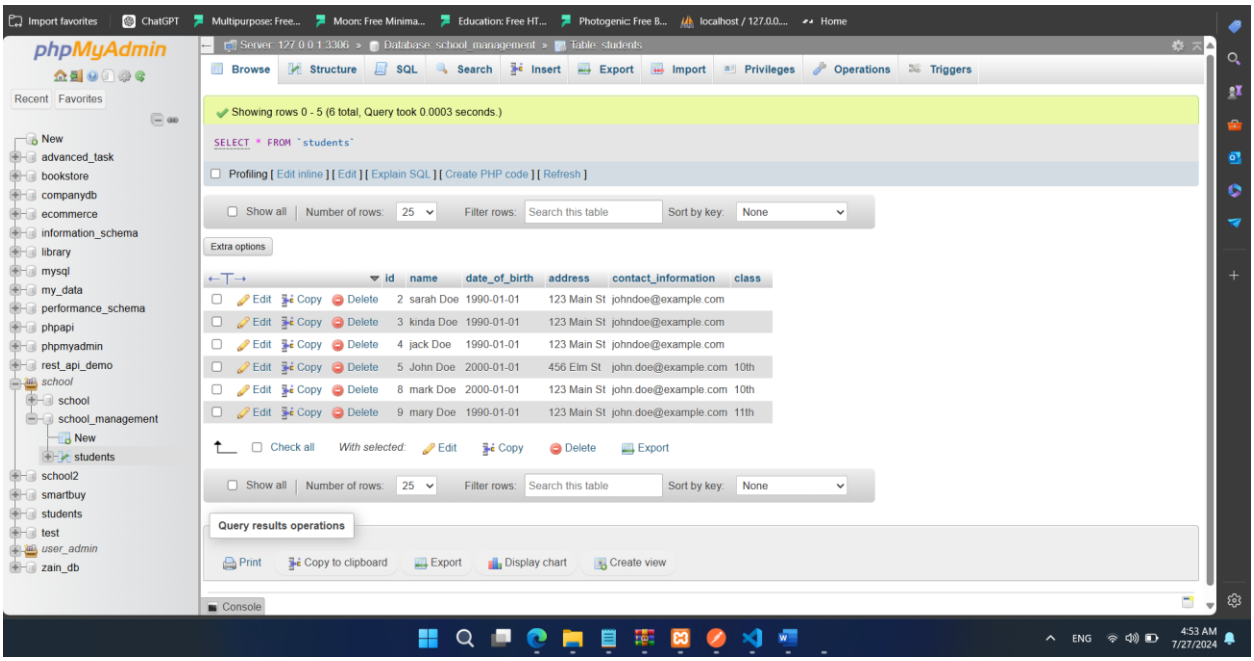
1. 1. Create Student (POST):



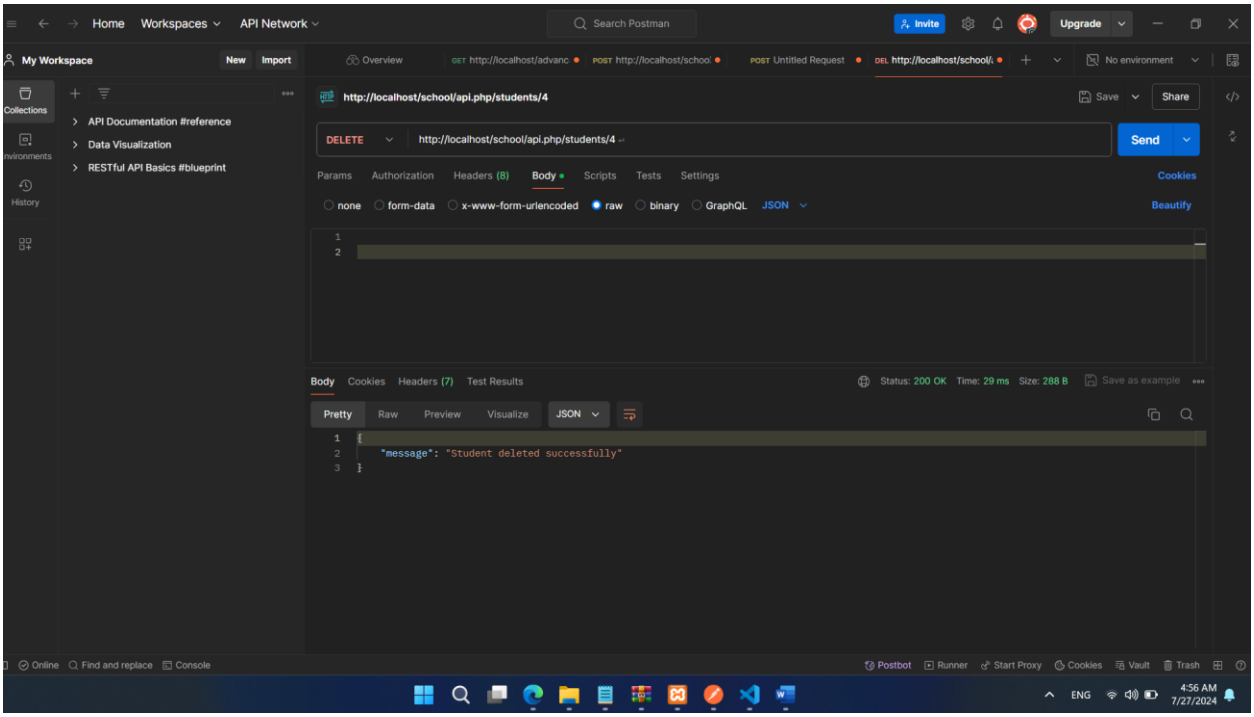


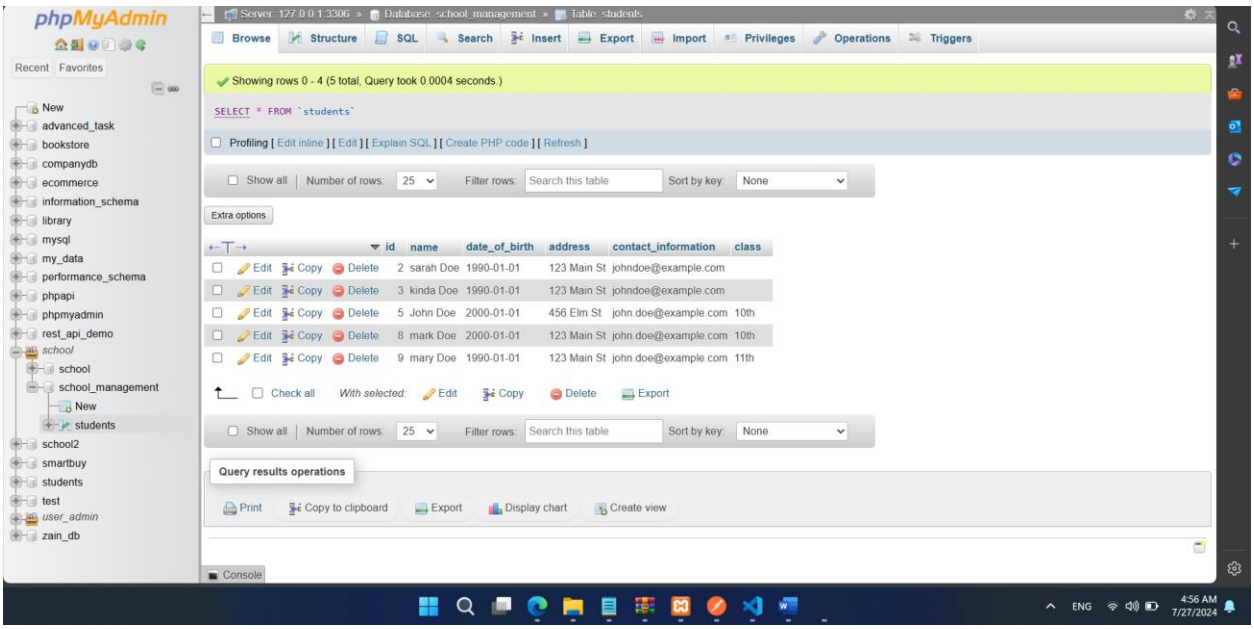
2. Update using put





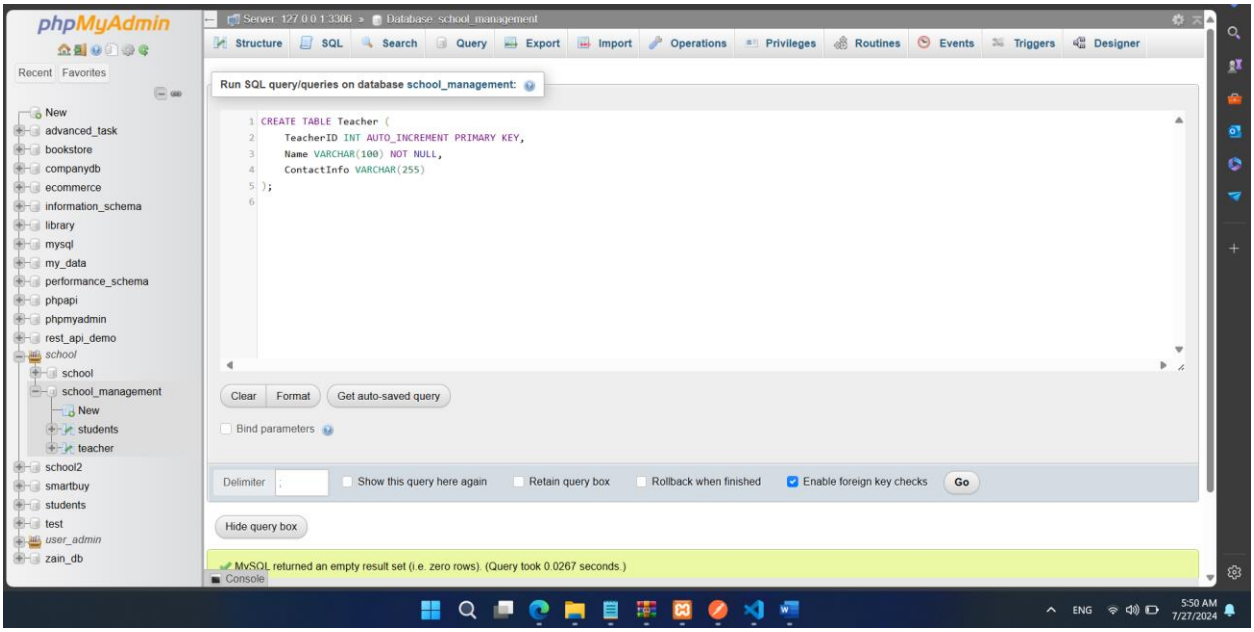
3. 3. Delete Student (DELETE):

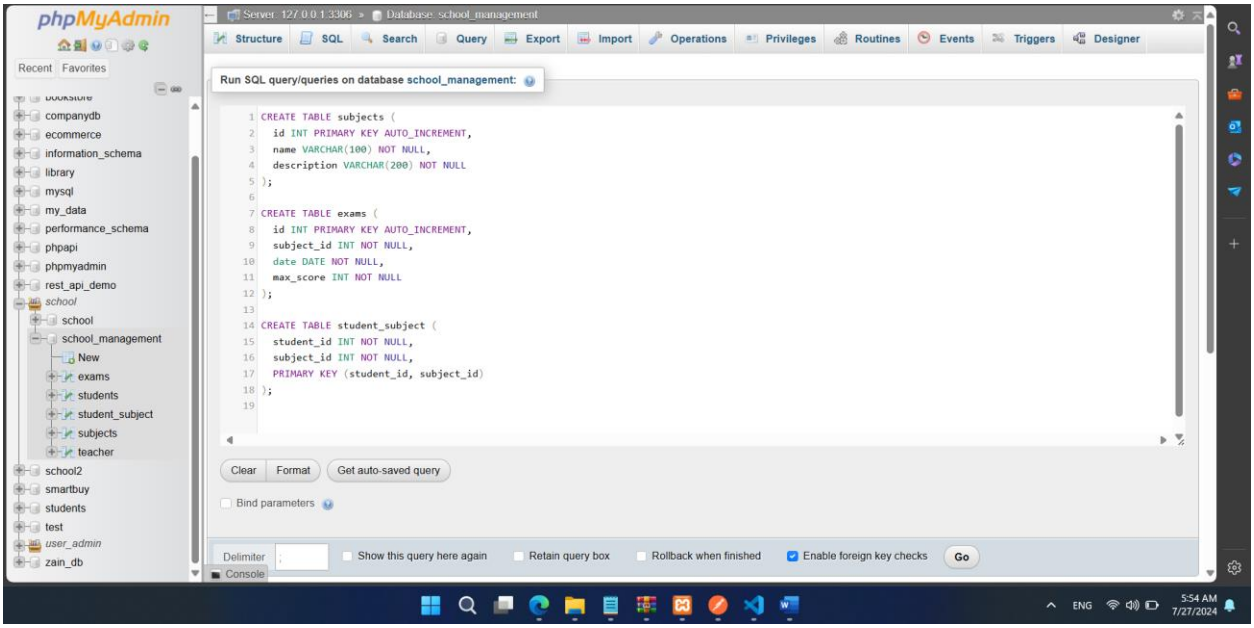




Task3:

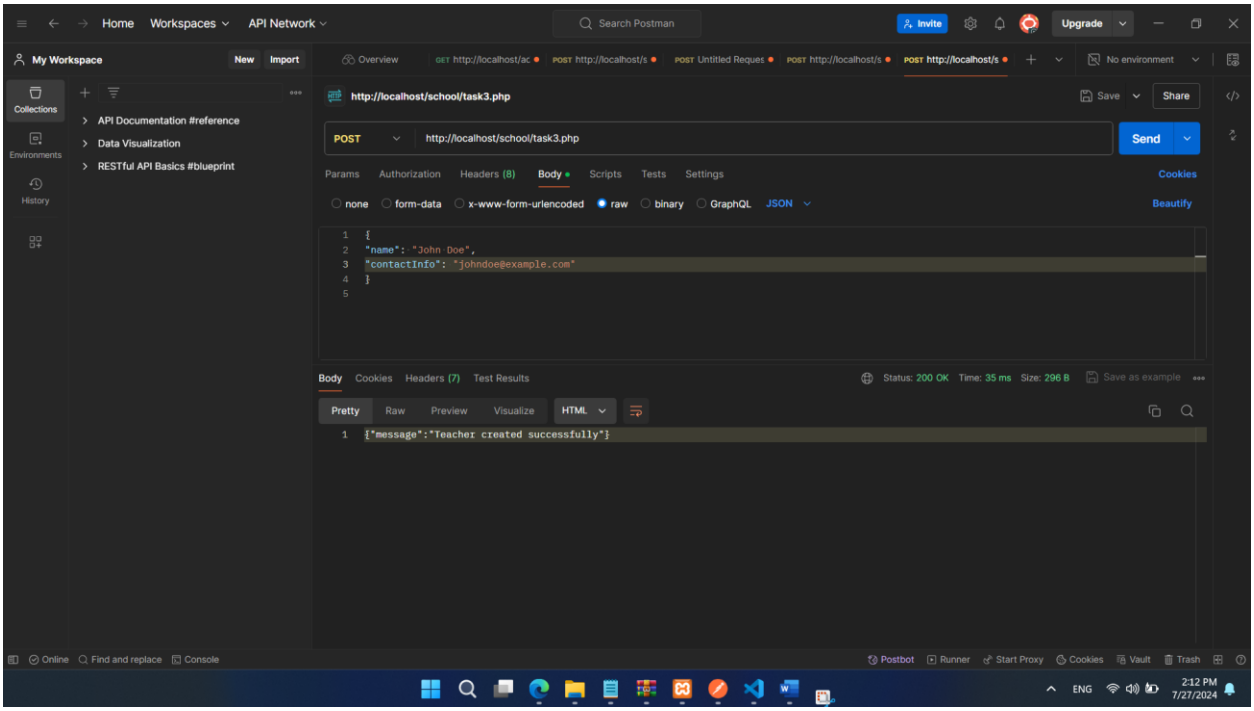
create tables and Relationships:

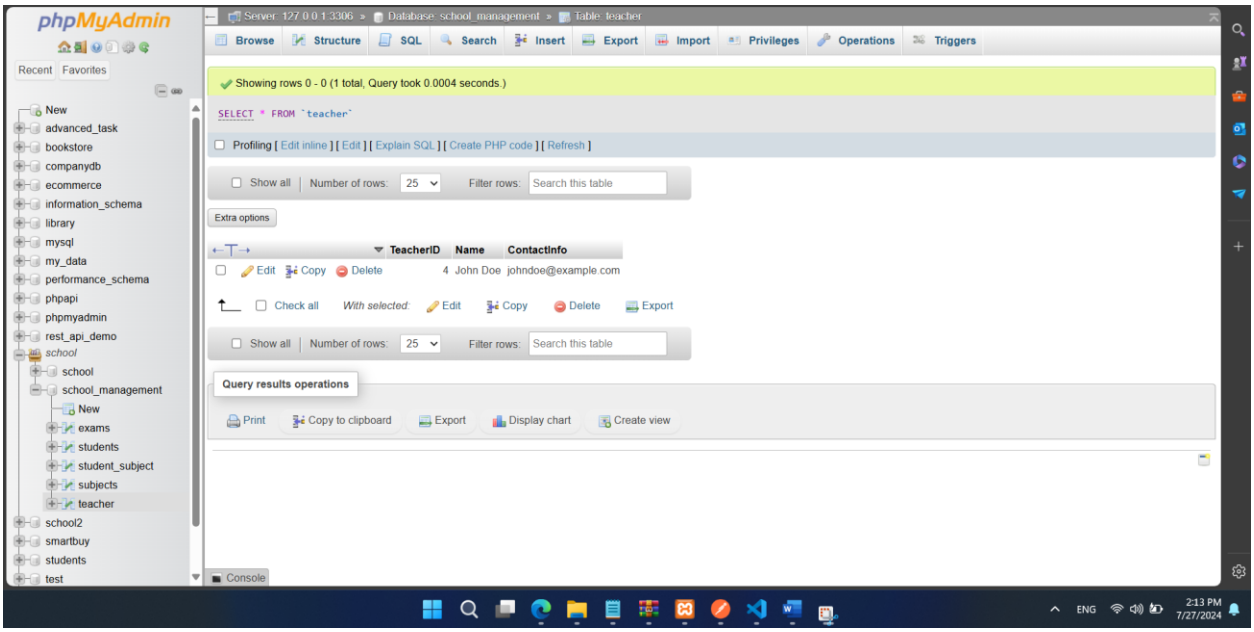




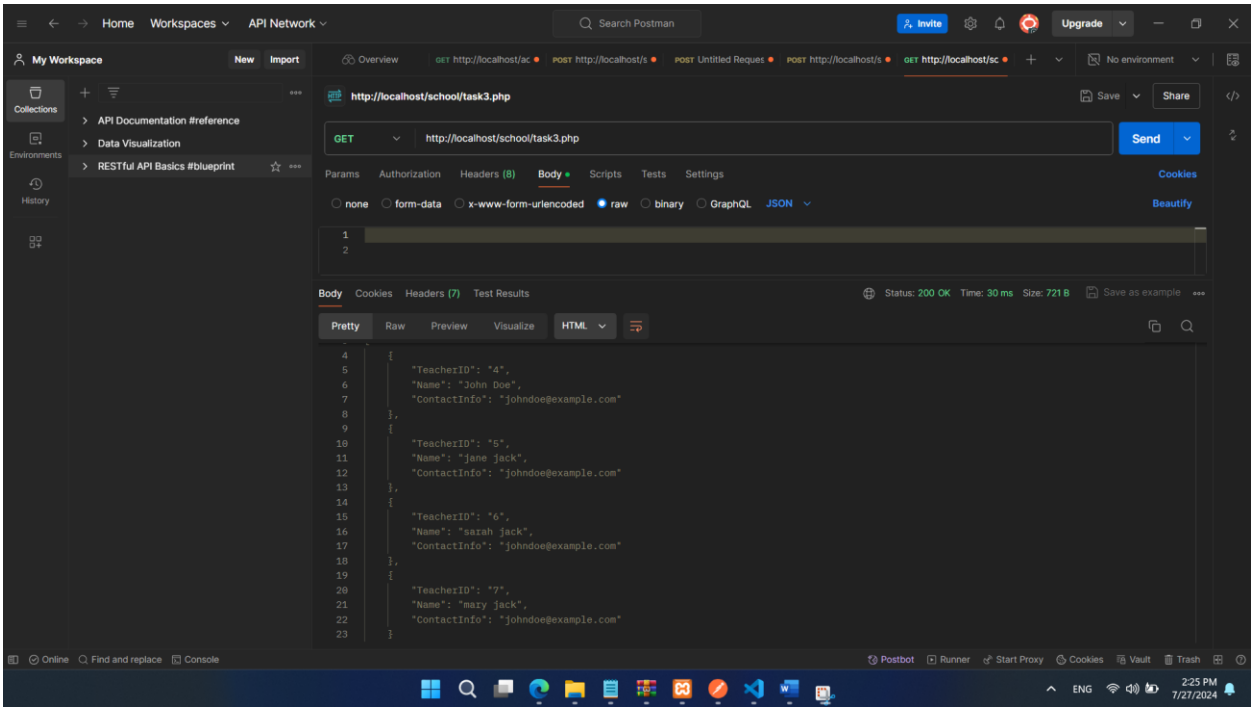
Create API Endpoints

1. Create Teacher (POST):

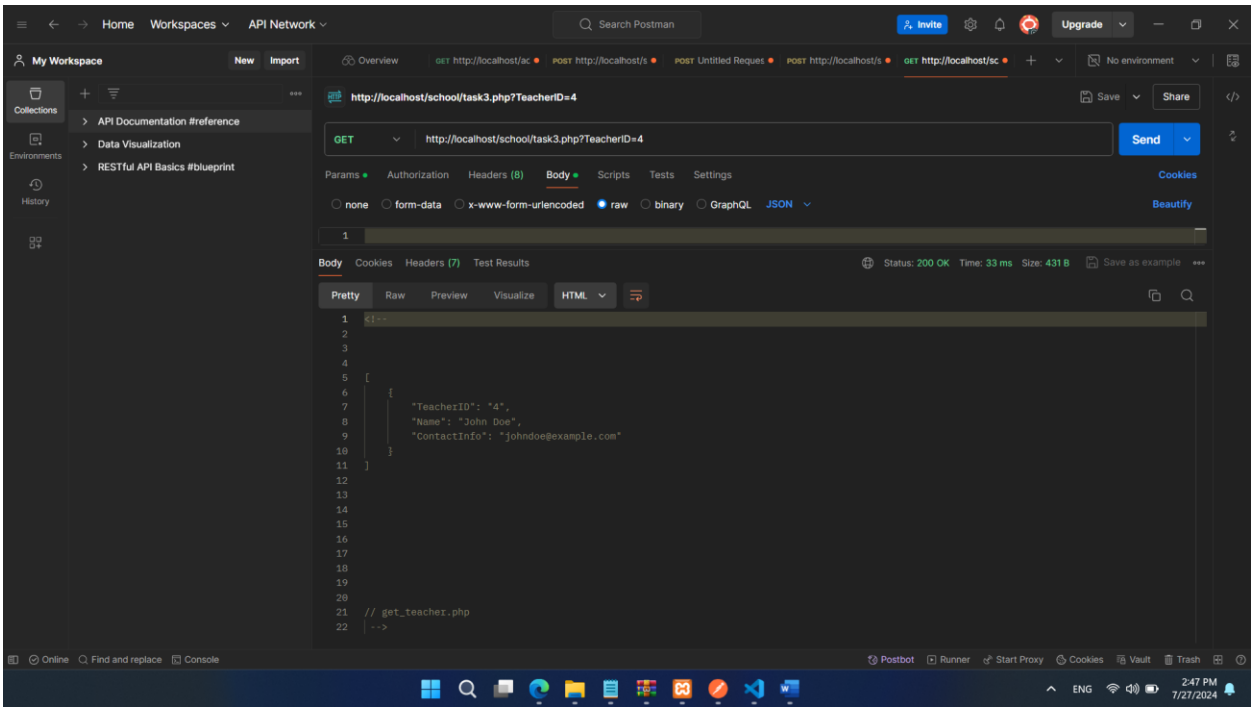




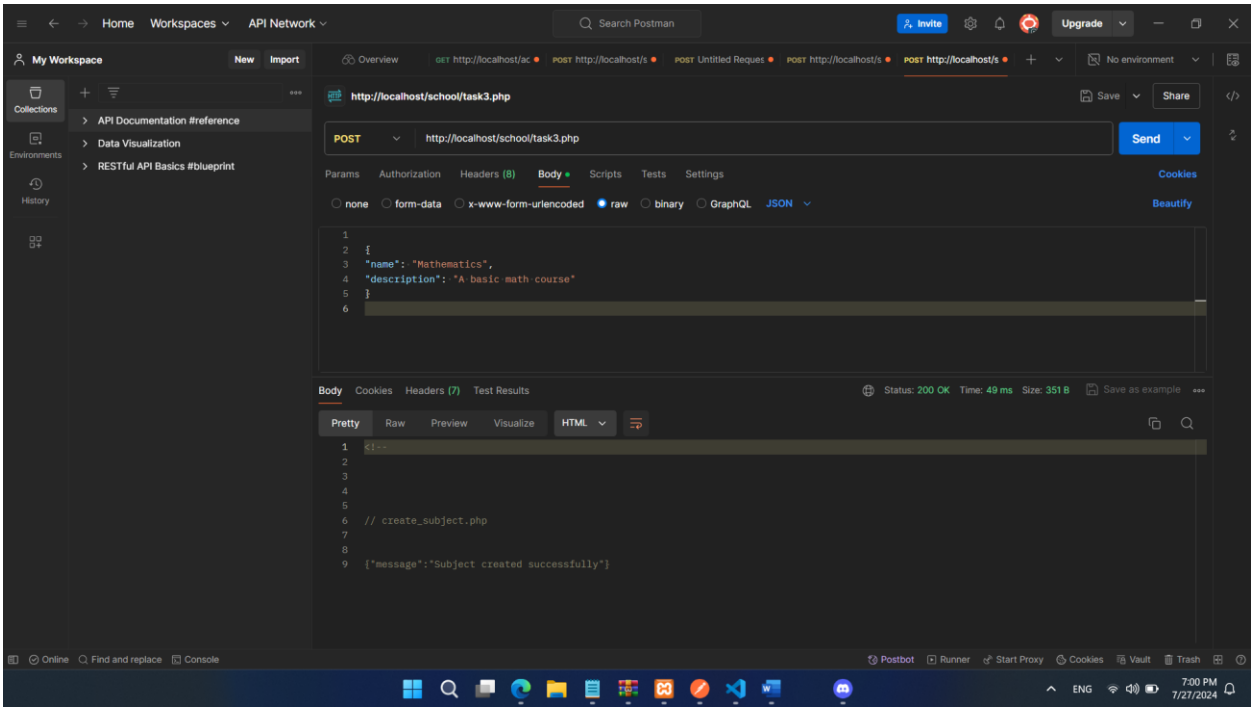
2. Get All Teachers (GET):

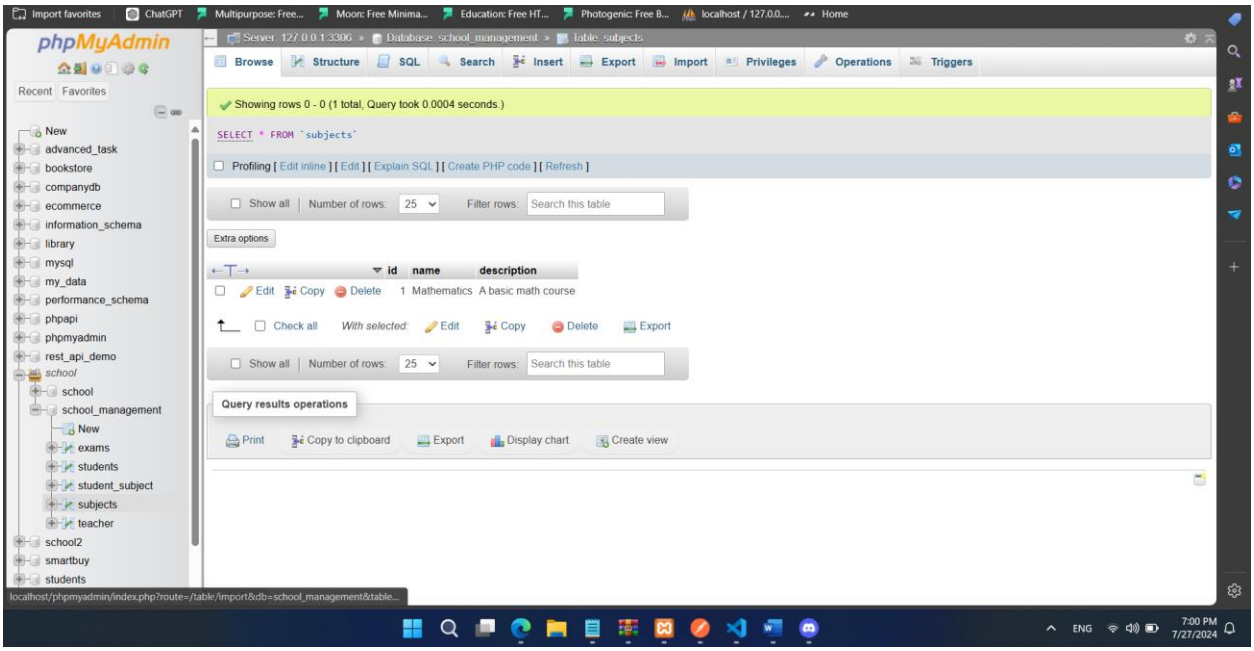


3. Get Teacher by ID (GET):

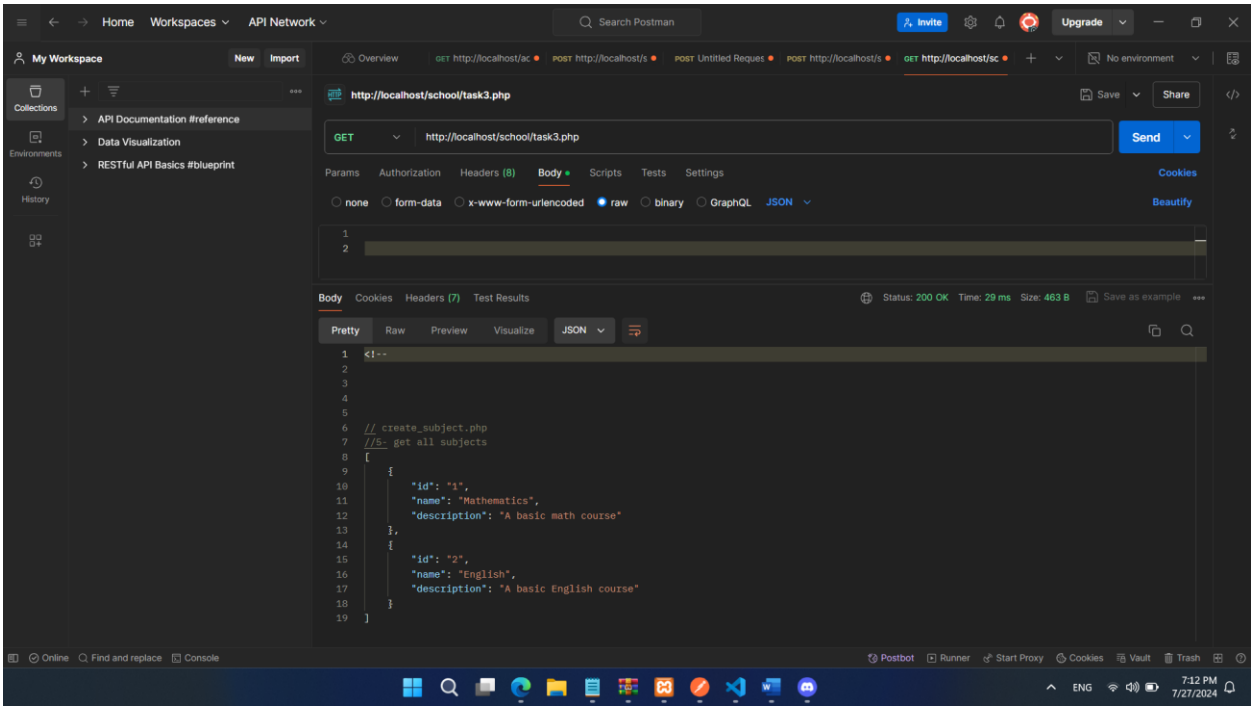


4. Create Subject (POST):

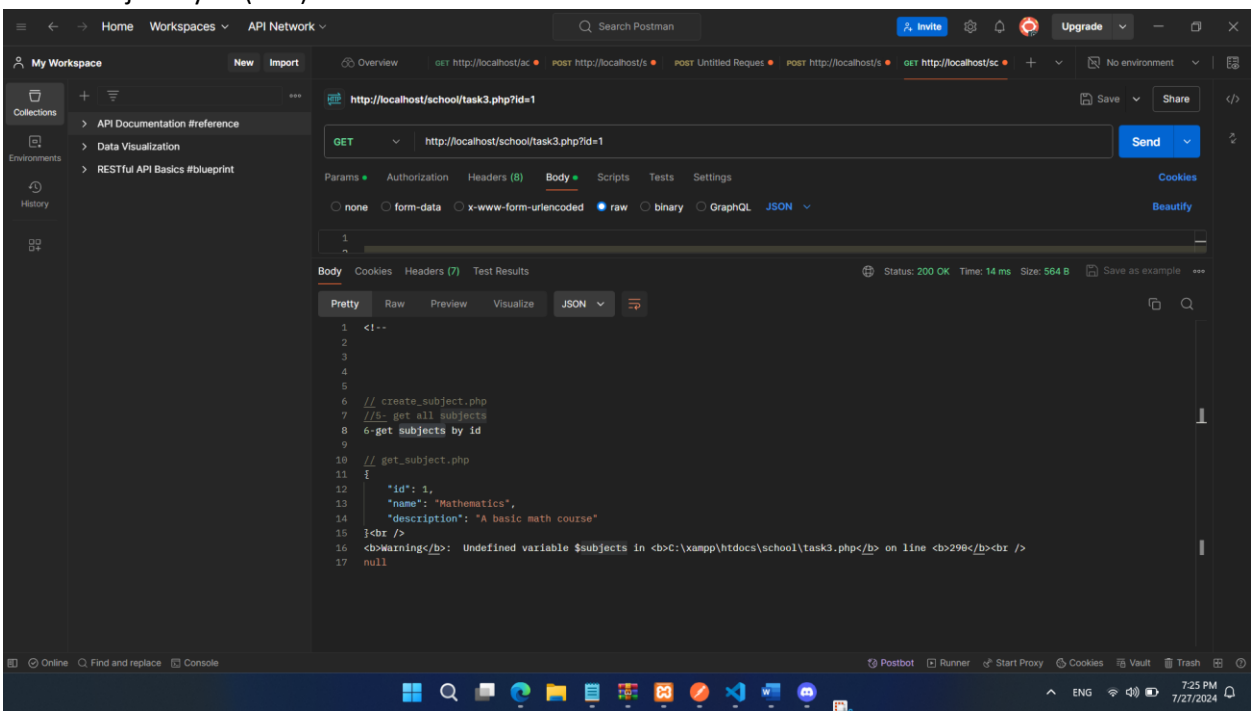




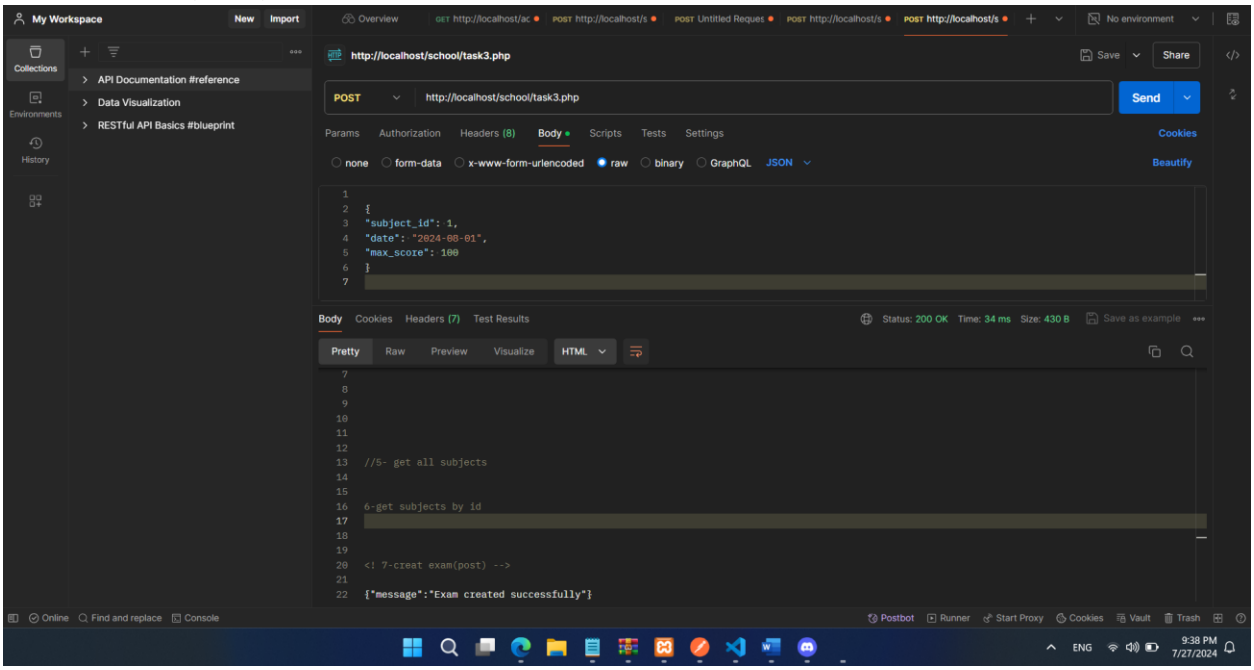
5. Get All Subjects (GET)

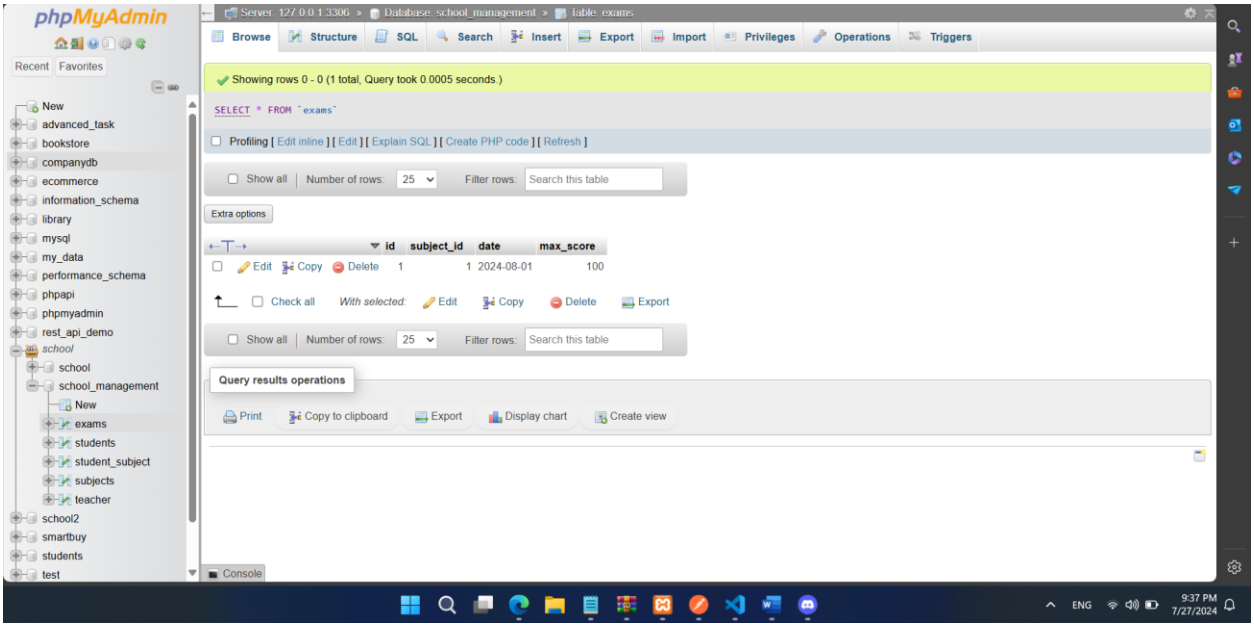


6. Get Subject by ID (GET):

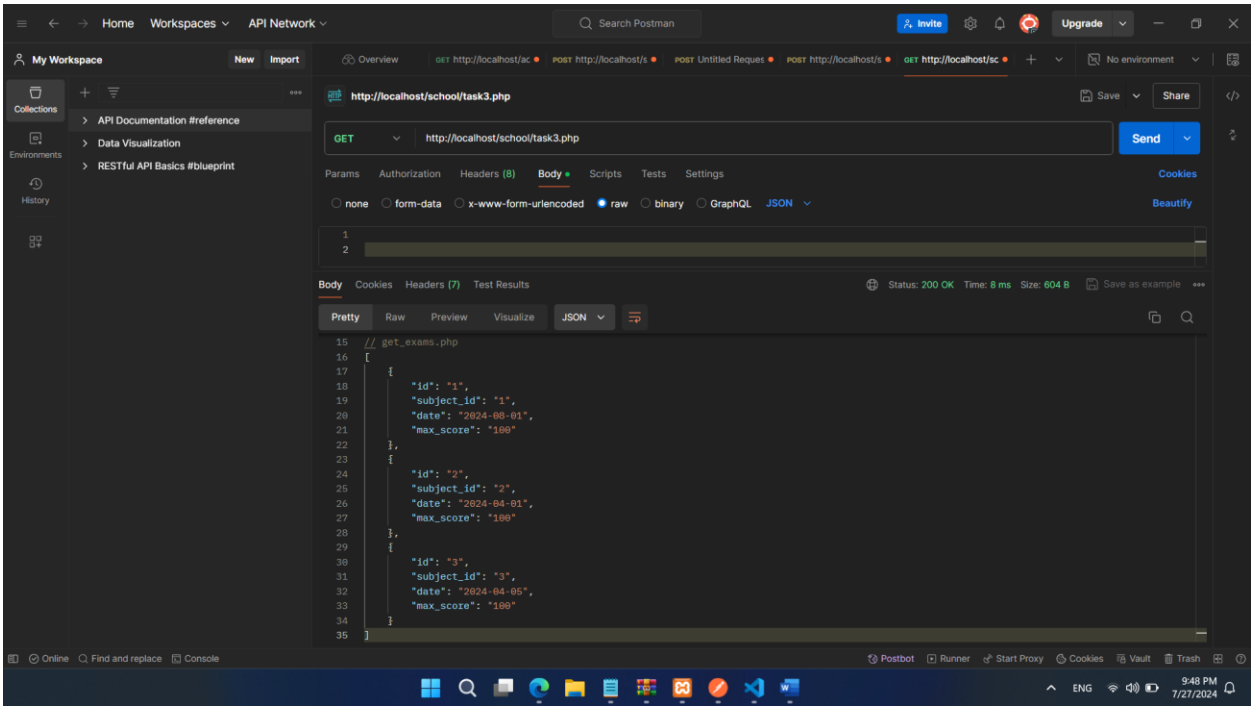


7. Create Exam (POST):

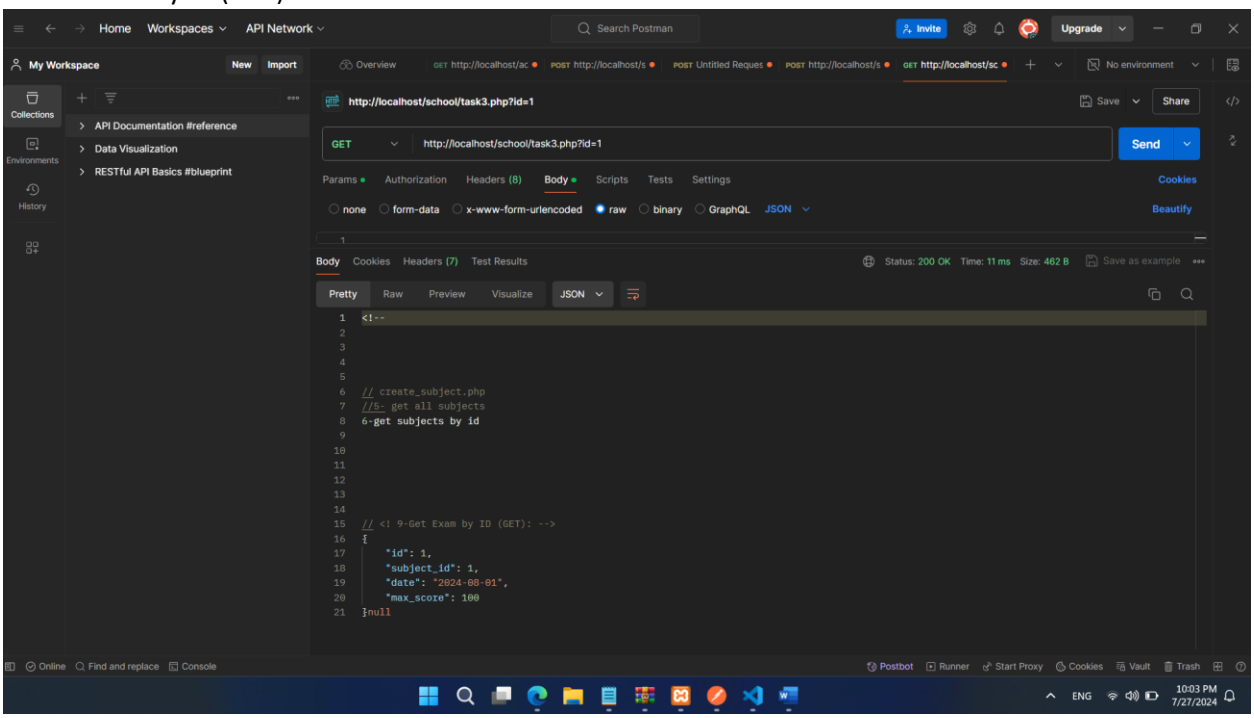




8. Get All Exams (GET):

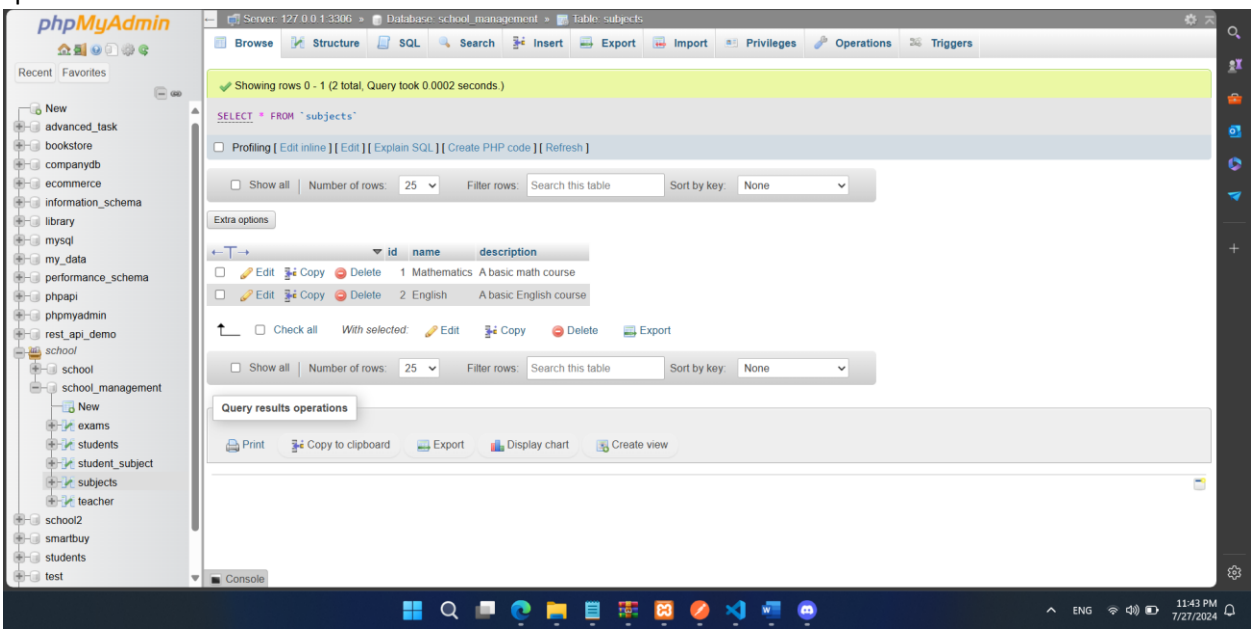


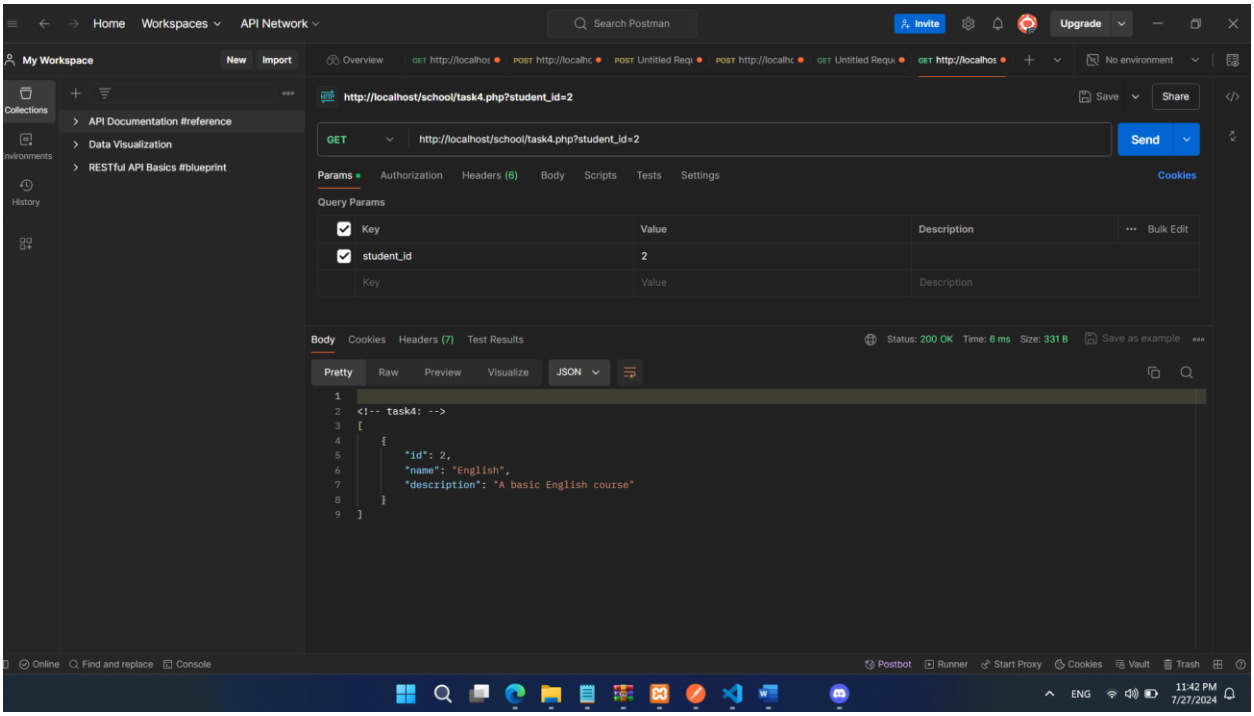
9. Get Exam by ID (GET):



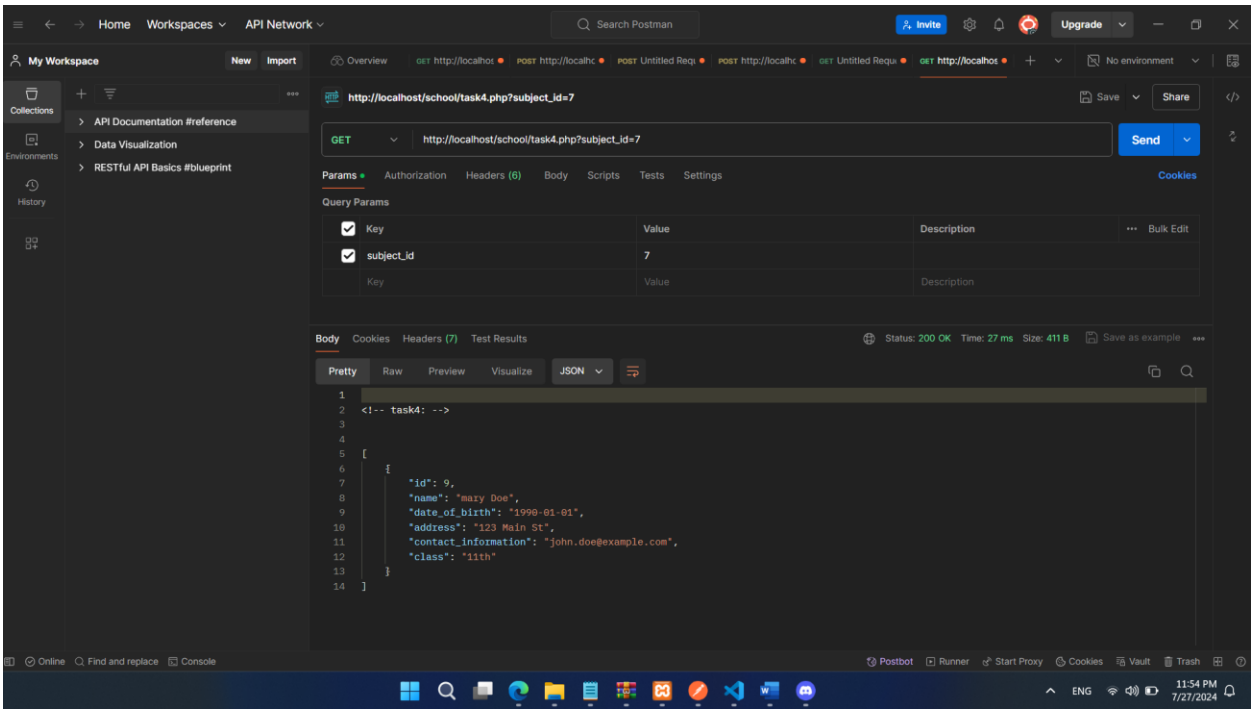
Task 4: Make these new APIs based on our customer requests

1. Retrieve a Student's Subjects (GET): Create an endpoint to get all the subjects associated with a specific student.

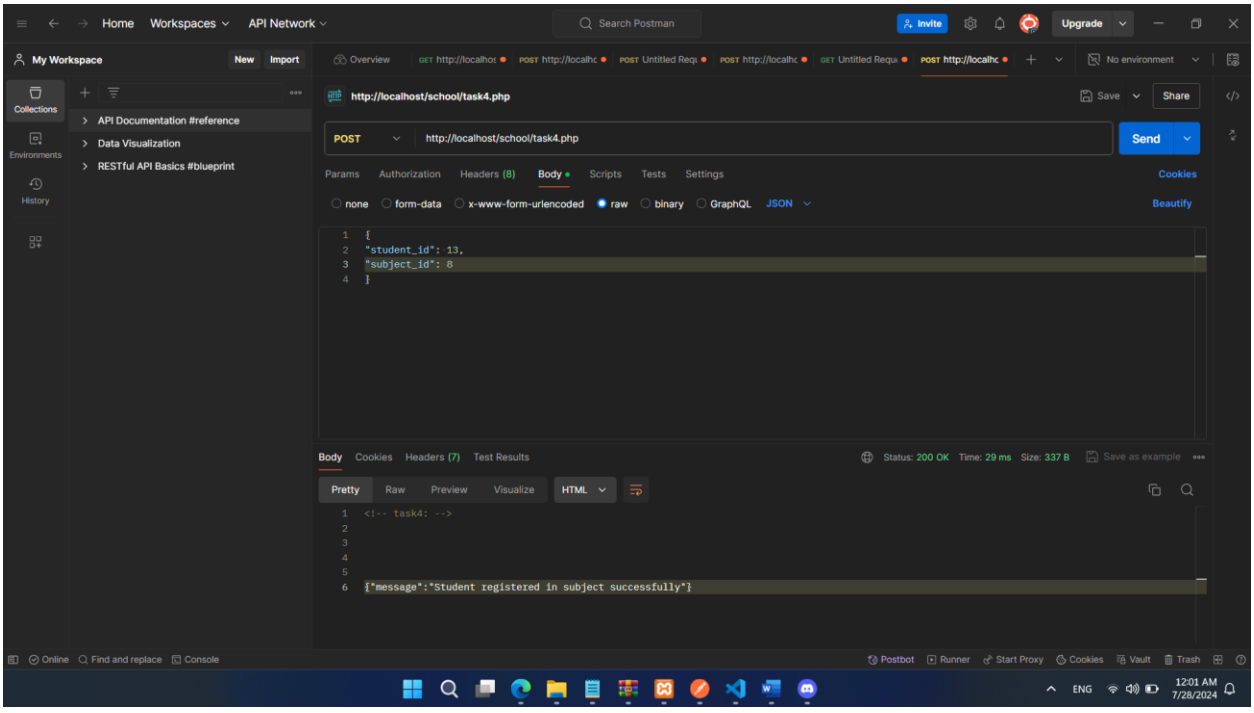


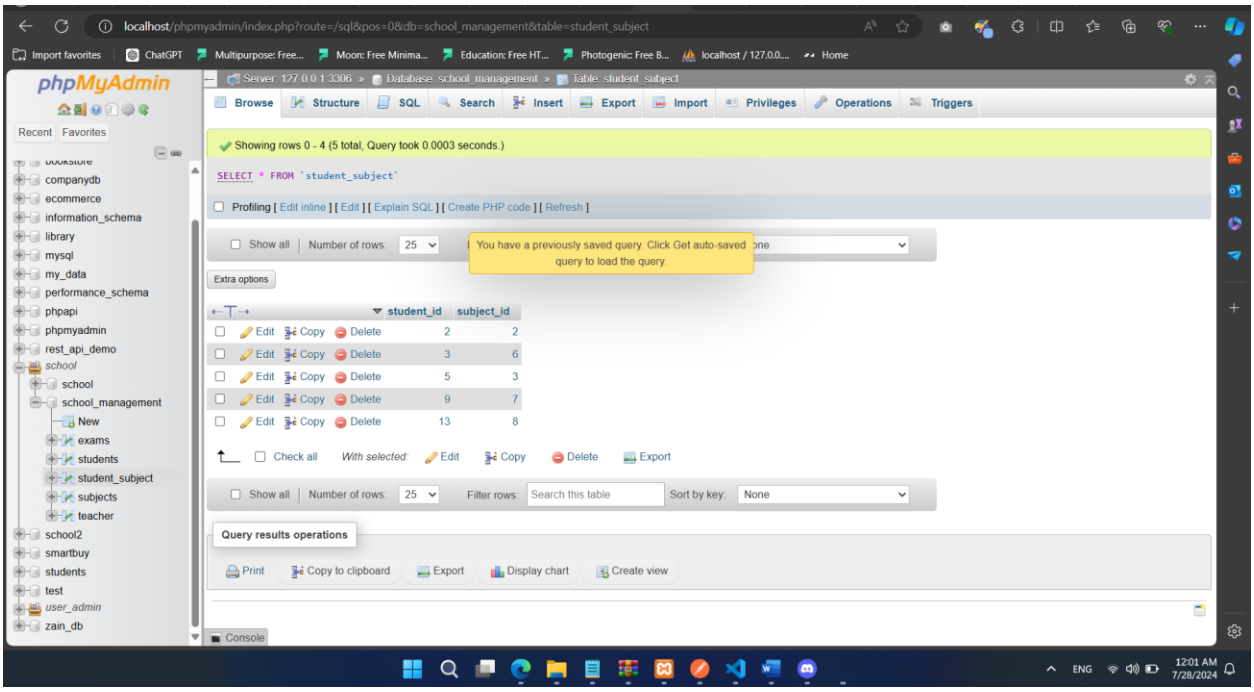


2. Retrieve a Subject's Students (GET): Create an endpoint to get all the students enrolled in a particular subject.

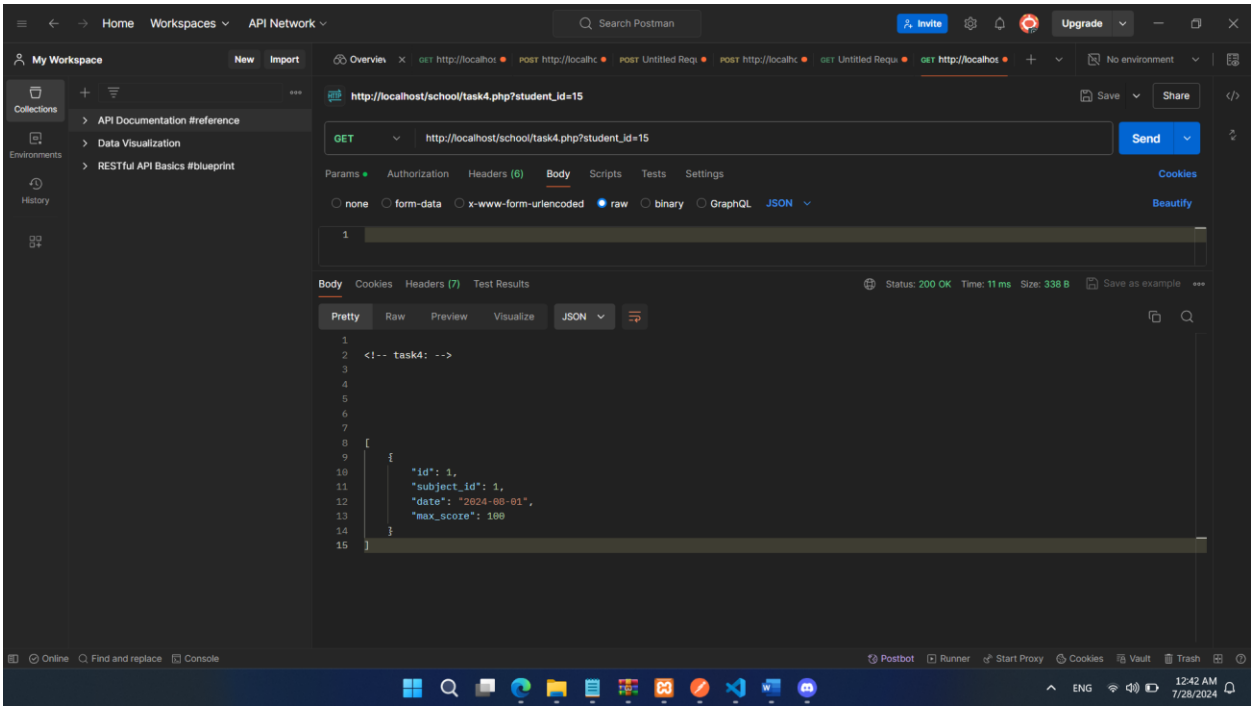


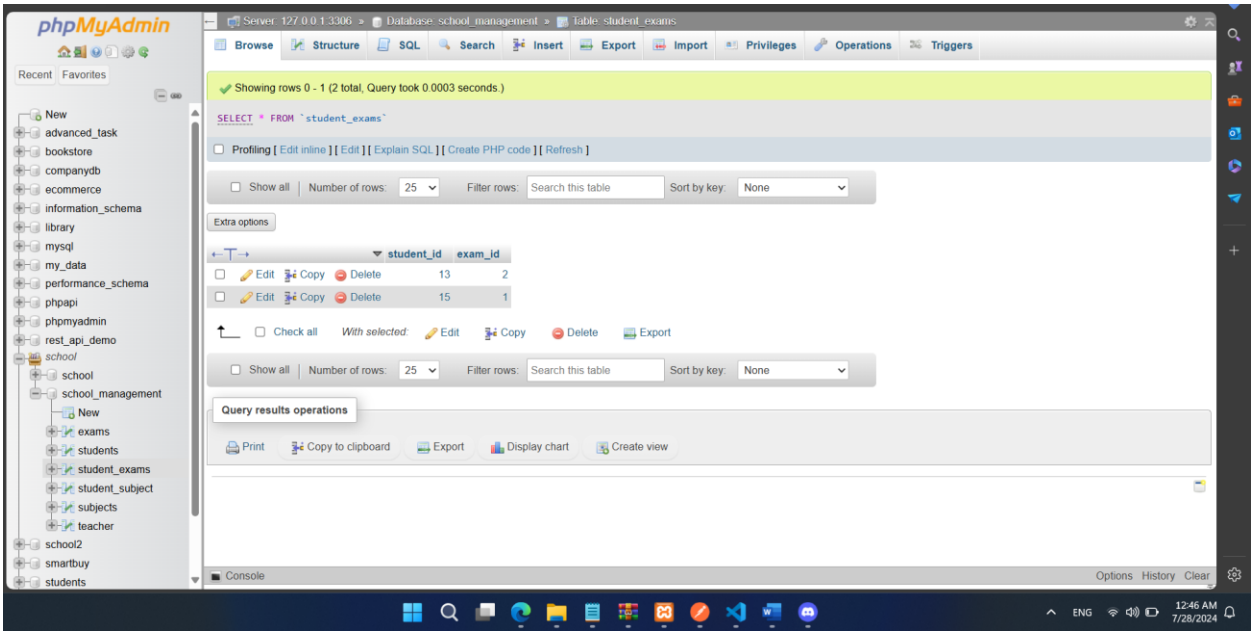
3. Register Students in Subjects (POST): Implement an API endpoint to register students in subjects (many-to-many relationship).



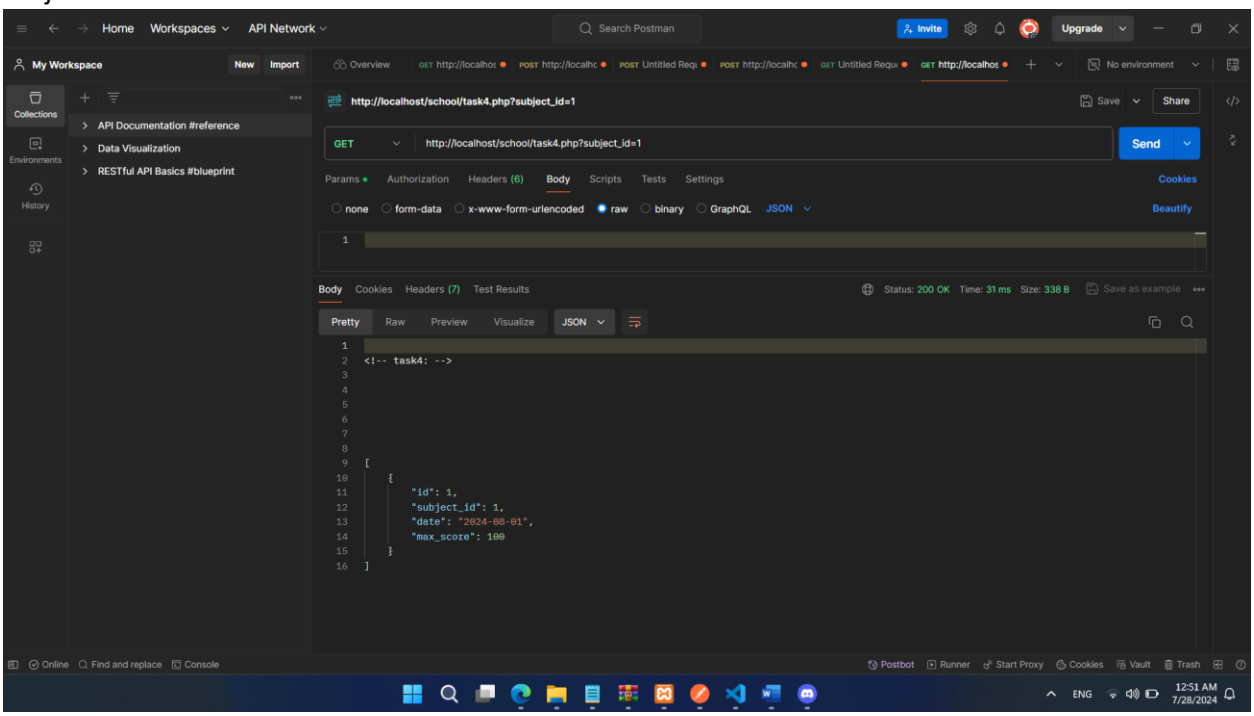


4. Retrieve a Student's Exams (GET): Create an endpoint to get all the exams taken by a specific student.





5. Retrieve a Subject's Exams (GET): Create an endpoint to get all the exams conducted for a particular subject.



6. Update Exam: Allow updating exam details, including scores