COLLEGE MANAGEMENT SYSTEM

Project Member:

Ms. Neha Gupta
Mr. Gopal Singh Devada
Mr. Suresh
Mr. Shashank
Mr. Pradeep
Mr. Harish Kumar

Guided By :

Prof. Indrakka Mali

Title :

- Introduction
- Objective
- Technology to be Used
- Proposed System
- Snapshots
- Diagrams
- Advantages
- Conclusion

Introductoin:

- The college management system helps Educational Institutions especially colleges in various ways, such as storing data, maintain student profiles, analyzing administrative and academic data, improving communication, and engaging students.

- A college management system is **a cloud based educational ERP software that enables HEIs to manage online admission & fees, students' attendance, library books, etc**. It can also generate students' performance reports & simplify the hassles of faculty.

- College Management System is an end-to-end solution for colleges to improve operational efficiency & institutional outcomes by automating Student-Faculty lifecycle & campus administration.

- The COLLEGE MANAGEMENT SYSTEM can be used to store student information like attendance and student details etc.

- Admin can create report regarding any student any time using this system. Using this system you can register new student and their course details.

Objective :

- The main objective of the College Management System is **to manage the details of College,Faculty,Course,Batch,Session**.

- It manages all the information about College, Student, Session, College.

- The project is totally built at administrative end and thus only the administrator is guaranteed the access

- **This project deals with the various functioning in College management process**.

- The main idea is to implement a proper process to system .

- In our existing system contains a many operations registration, student search, fees, details of the student etc.

- The organization can maintain computerized records without redundant entries.

Software & Hardware used:

Software Requirements :

- Operating System : Windows 7 or Windows 10
- Language : Java
- IDE : Spring Boot, PostMan
- Backend : Microsoft MySQL server

Hardware Requirements :

- CPU : Intel Pentium IV processor
- RAM : 512 MB or above
- Hard Disk : 2 GB hard disk space or minimum

Proposed System:

- The college management system helps Educational Institutions especially colleges in various ways, such as storing data, maintain student profiles, analyzing administrative and academic data, improving communication, and engaging students.

- **This is a web-related application that permits us to approach the entire knowledge regarding the college, students, faculties etc**.

- This application is also called as College management system. It offers an actual trip of the college campus.

- First we have to run the Application on the browser, the index page will be shown after that if the user is registered then he/she can login the page and he/she can entering in the application.

- If the user is not registered then he/she needs to register themselves and then he/she can continuing to login and he/she can access any time.

- After that related pages will be there like if user is admin so he/she can access all the operations like insert, delete, update and search the details.

- If the user is customer then he/she can view the related record and he/she can update the record.

- And after that he/she can logged out by themselves.

Back-End :

- Core Java
- Spring Boot
- Spring Data JPA
- Hibernate

Spring Boot Web

MySQL Database

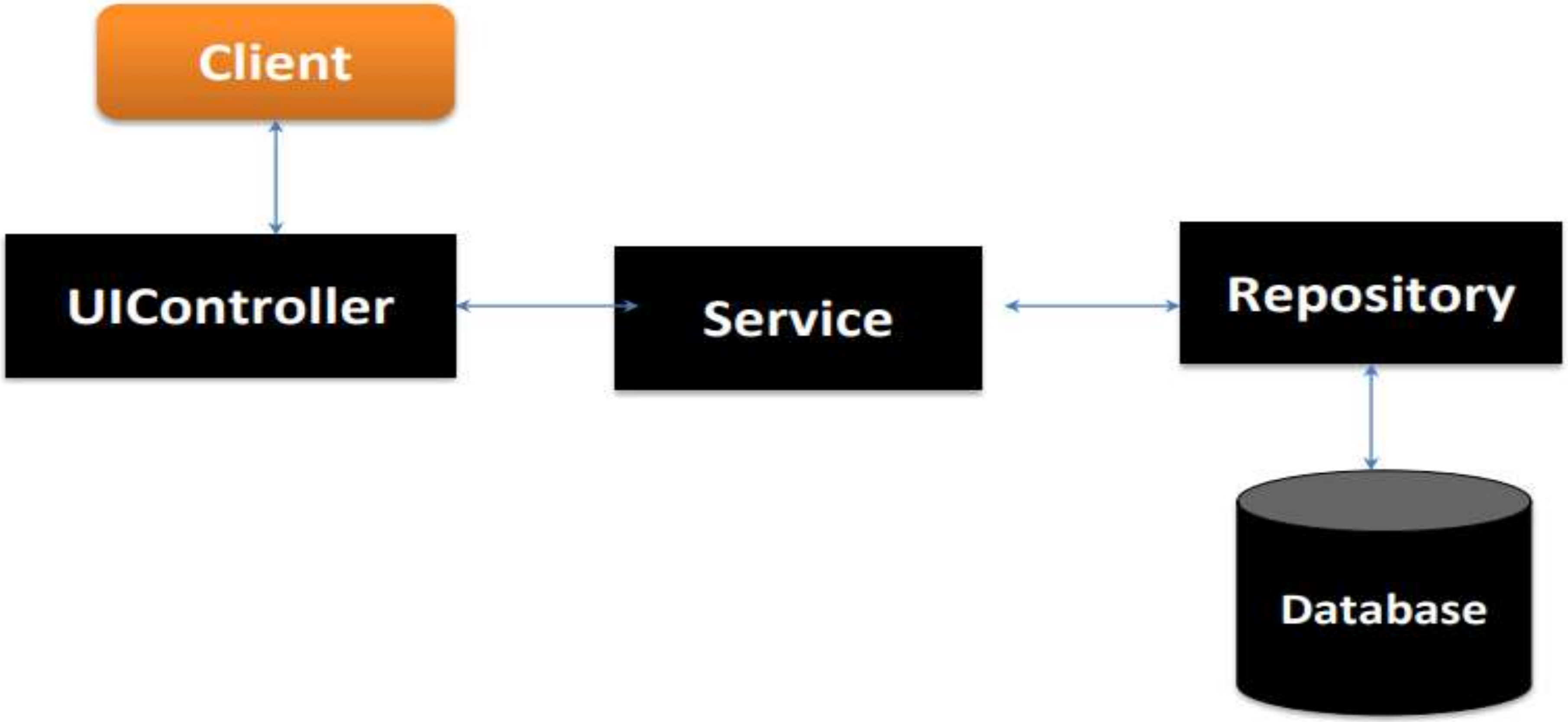
JUnit

About Back-End:

- There are mainly four operations will be performed by the user and admin Ex. Insert, update, retrieve/fetch, delete.
- These operation will be performed using the spring boot framework, core java and spring boot web , spring data Jpa , and hibernate.
- And thymeleaf is the design pattern for the user interface.
- For connectivity purpose we are using MySQL database. It is connected to the java and database

Working Of Back-End ;

- There are mainly three stages



- UIController - In Spring Boot, the UIController class is responsible for processing incoming requests, preparing a model, and returning the view to be rendered as a response.

- Service - Components are the class file which contains @Service annotation.

- These class files are used to write business logic in a different layer.

- Repository - Repository is a specialization of @Component annotation which is used to indicate that the class provides the mechanism for storage, retrieval, update, delete and search operation on objects

- Repository is directly connected with the database and then it can return to the repository and response to the service and it can return to the controller and then response send to the client.

Advantages:

- ➢ Better and improved admission rate
- ➢ Better teacher performance
- ➢ Drastic reduction in manual tasks
- ➢ Elimination of paper based tasks
- ➢ Improved fee management

- Makes the institute smart and ready for the future.

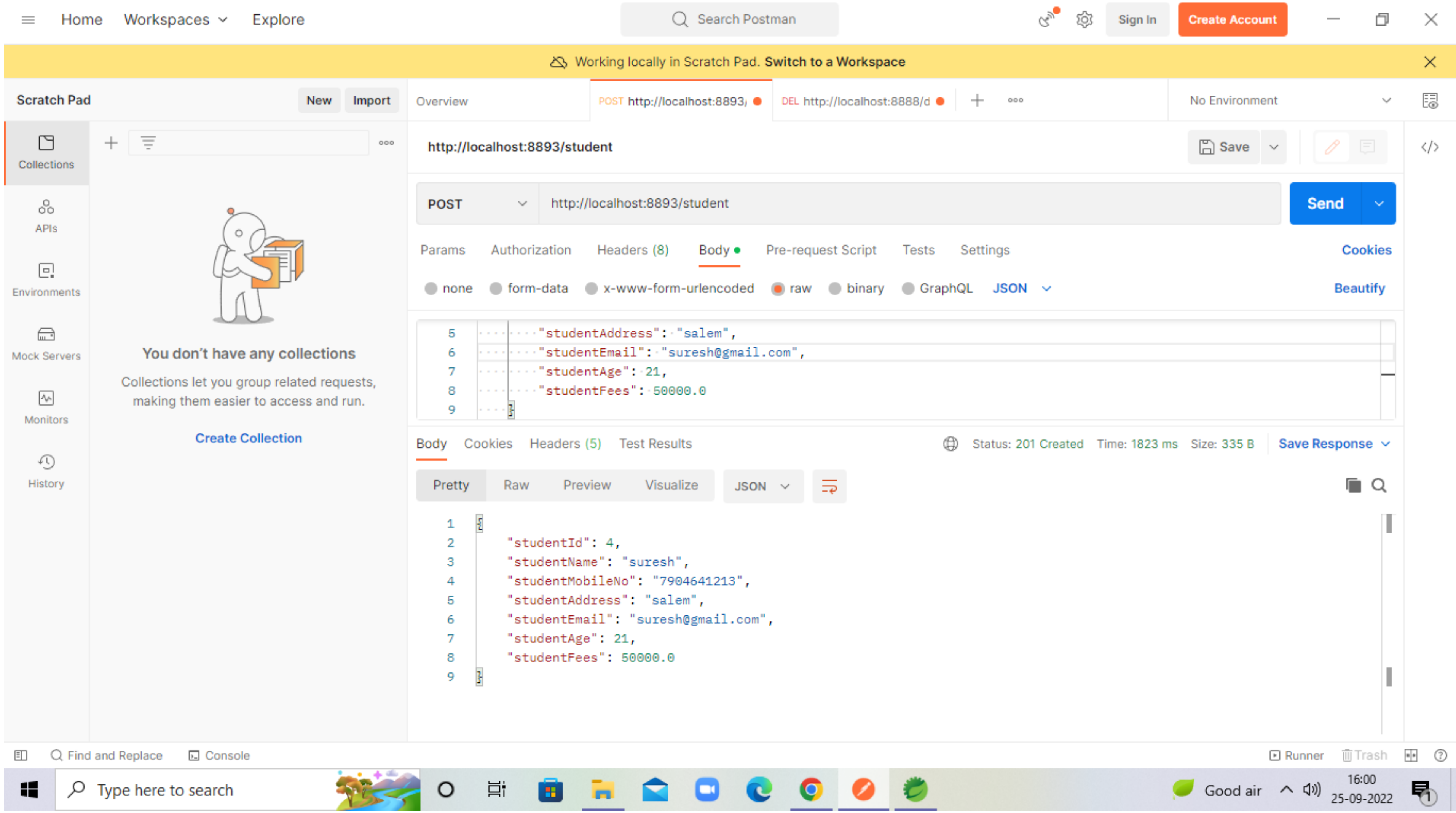
Improved institute reputation

better resource utilization

- Duplication of the student and teacher data is avoided.

snap shots:

Add Student:



http://localhost:8893/student

POST http://localhost:8893/student

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

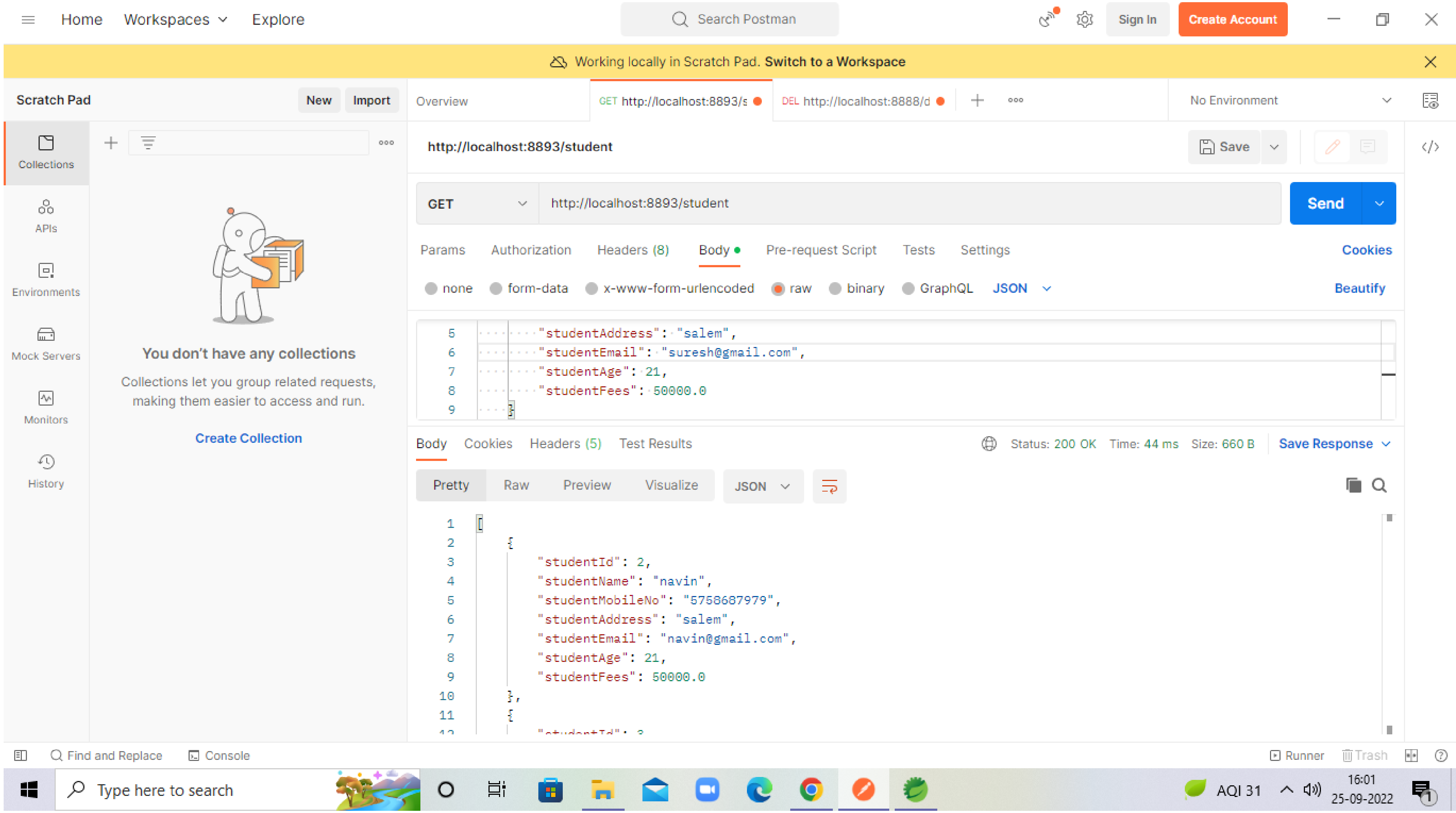
```
    "studentAddress": "salem",
    "studentEmail": "suresh@gmail.com",
    "studentAge": 21,
    "studentFees": 50000.0
}
```

Body — Status: 201 Created Time: 1823 ms Size: 335 B

Pretty JSON

```
{
    "studentId": 4,
    "studentName": "suresh",
    "studentMobileNo": "7904641213",
    "studentAddress": "salem",
    "studentEmail": "suresh@gmail.com",
    "studentAge": 21,
    "studentFees": 50000.0
}
```

Get All Student Details:



http://localhost:8893/student

GET http://localhost:8893/student

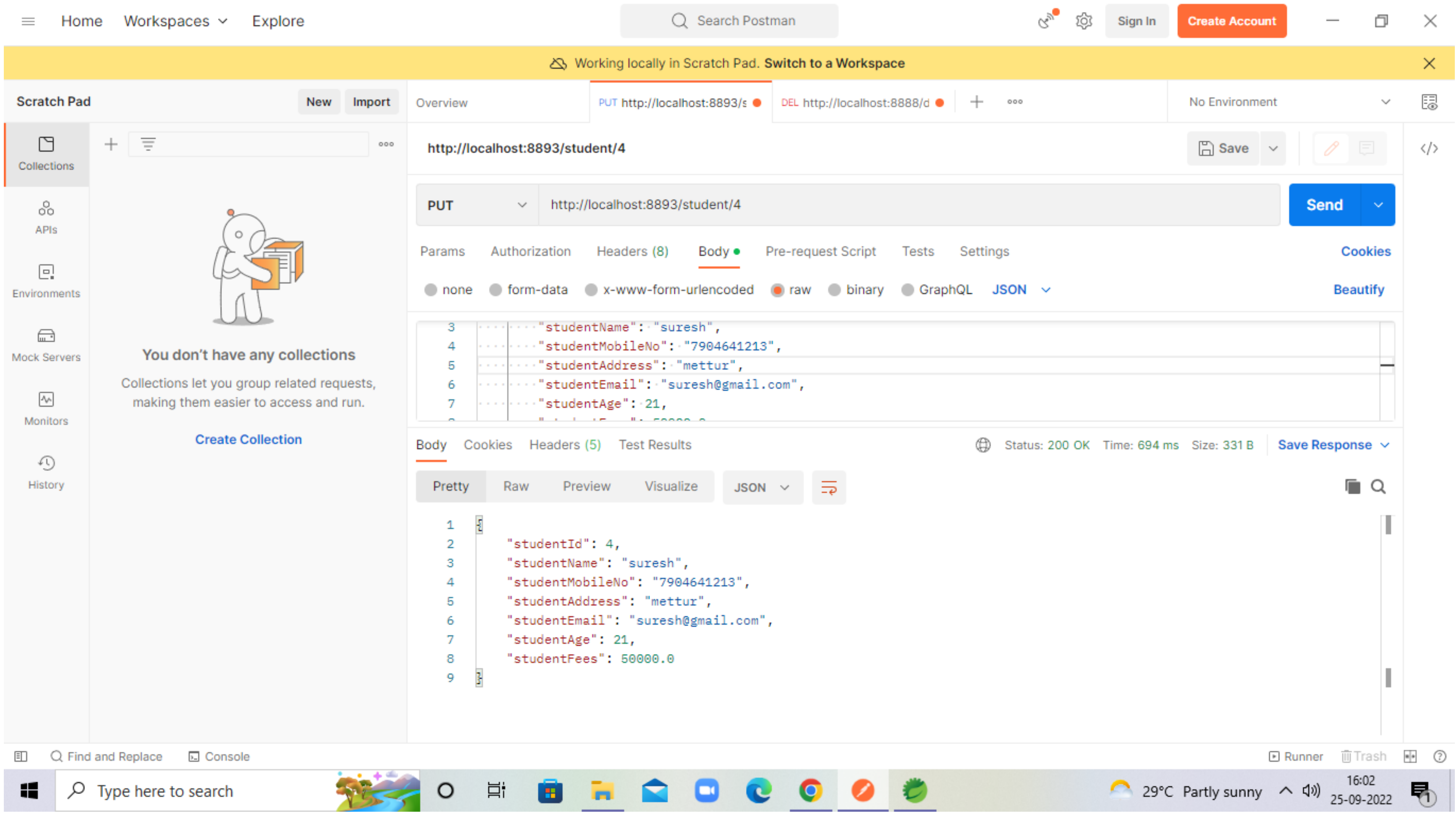
Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
    "studentAddress": "salem",
    "studentEmail": "suresh@gmail.com",
    "studentAge": 21,
    "studentFees": 50000.0
}
```

Body — Status: 200 OK Time: 44 ms Size: 660 B — JSON

```
[
    {
        "studentId": 2,
        "studentName": "navin",
        "studentMobileNo": "5758687979",
        "studentAddress": "salem",
        "studentEmail": "navin@gmail.com",
        "studentAge": 21,
        "studentFees": 50000.0
    },
    {
        "studentId": 3
```

Update Student:



http://localhost:8893/student/4

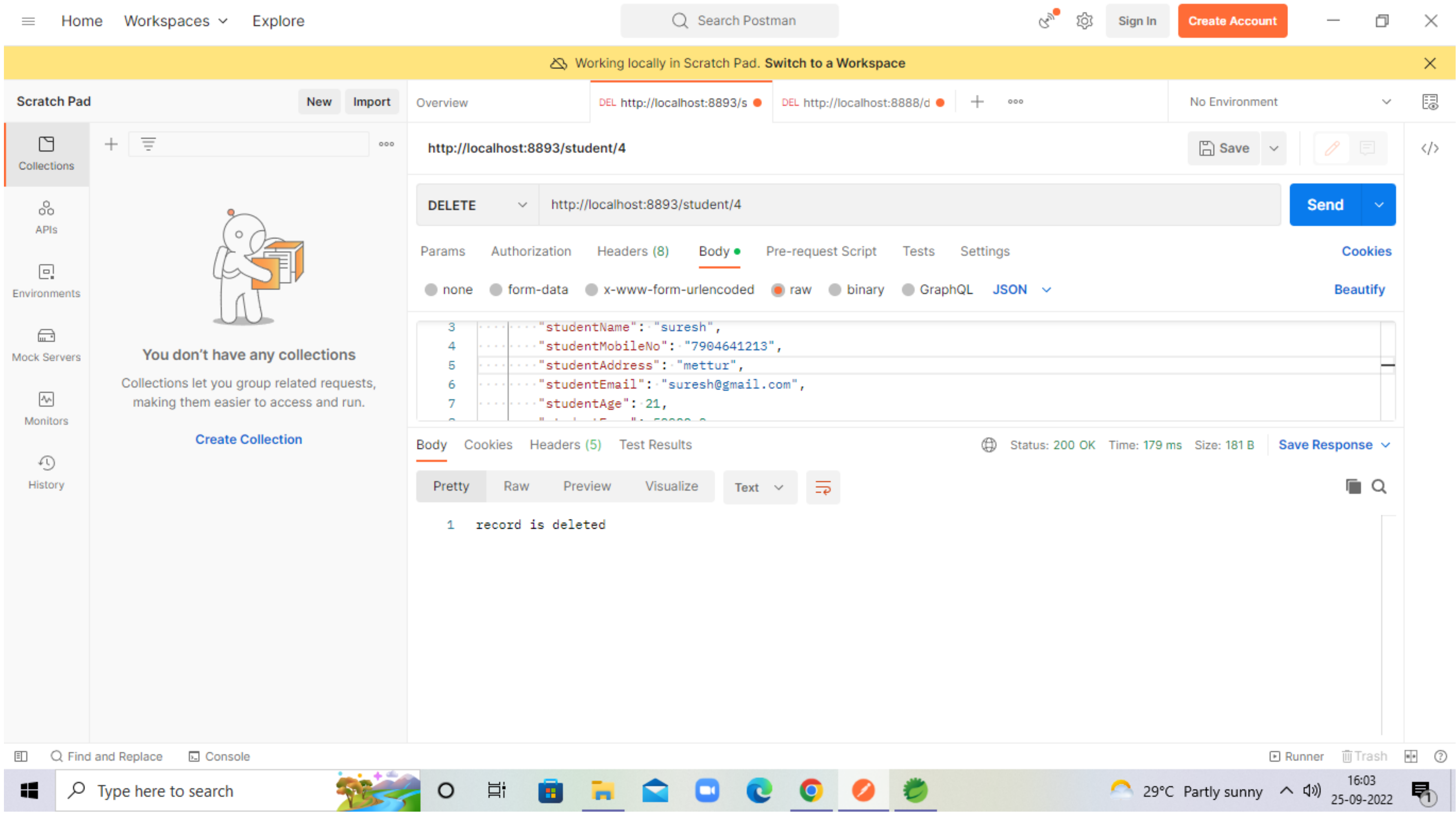
PUT http://localhost:8893/student/4

```
    "studentName": "suresh",
    "studentMobileNo": "7904641213",
    "studentAddress": "mettur",
    "studentEmail": "suresh@gmail.com",
    "studentAge": 21,
```

Status: 200 OK Time: 694 ms Size: 331 B

```
{
    "studentId": 4,
    "studentName": "suresh",
    "studentMobileNo": "7904641213",
    "studentAddress": "mettur",
    "studentEmail": "suresh@gmail.com",
    "studentAge": 21,
    "studentFees": 50000.0
}
```

Delete Student By Id:



http://localhost:8893/student/4

DELETE http://localhost:8893/student/4

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

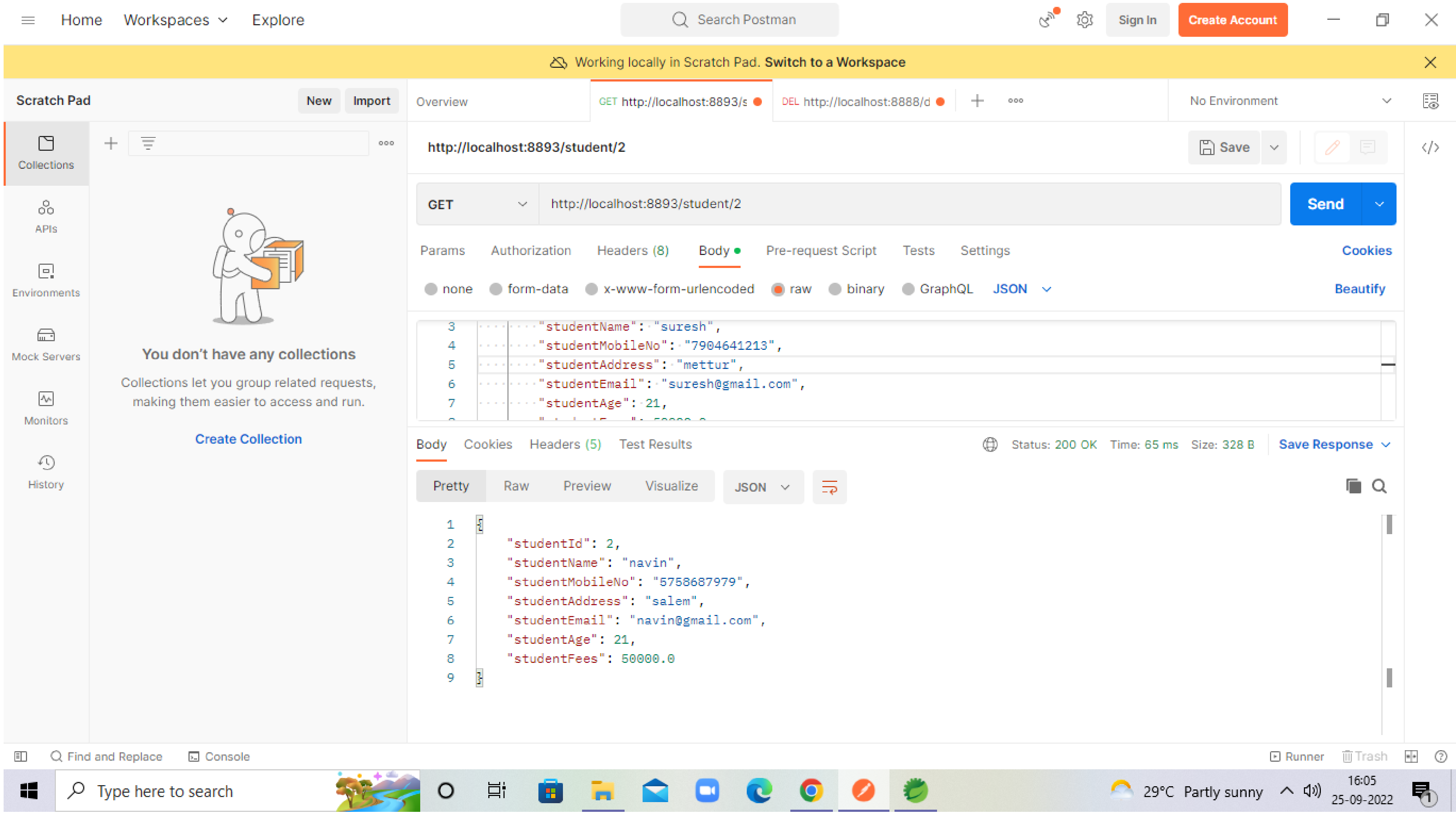
```
    "studentName": "suresh",
    "studentMobileNo": "7904641213",
    "studentAddress": "mettur",
    "studentEmail": "suresh@gmail.com",
    "studentAge": 21,
```

Body Cookies Headers (5) Test Results — Status: 200 OK Time: 179 ms Size: 181 B Save Response

Text

```
record is deleted
```

Get Student By Id:



http://localhost:8893/student/2

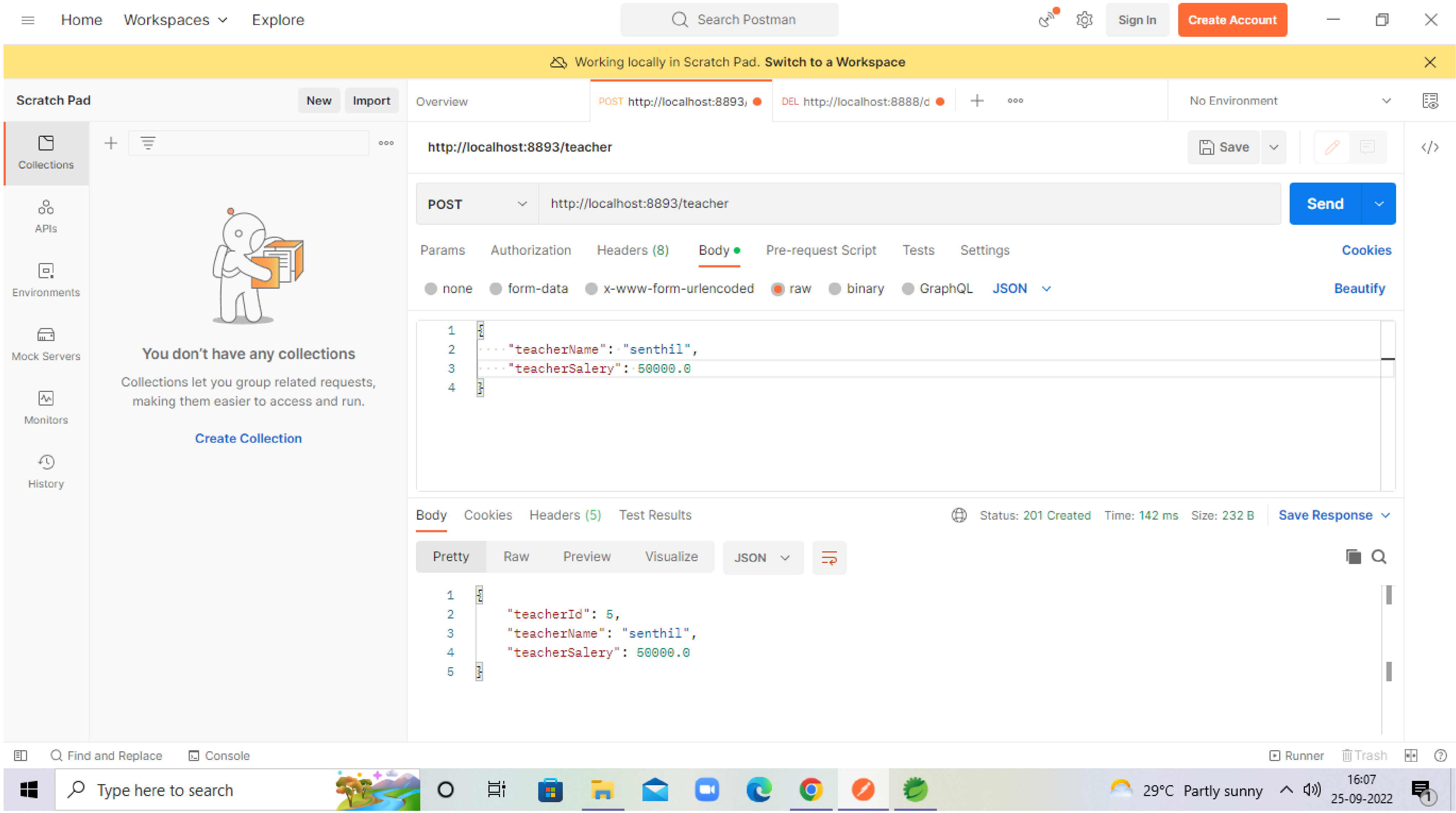
GET http://localhost:8893/student/2

```
    "studentName": "suresh",
    "studentMobileNo": "7904641213",
    "studentAddress": "mettur",
    "studentEmail": "suresh@gmail.com",
    "studentAge": 21,
```

Status: 200 OK Time: 65 ms Size: 328 B

```
{
    "studentId": 2,
    "studentName": "navin",
    "studentMobileNo": "5758687979",
    "studentAddress": "salem",
    "studentEmail": "navin@gmail.com",
    "studentAge": 21,
    "studentFees": 50000.0
}
```

Add Teacher:



http://localhost:8893/teacher

POST http://localhost:8893/teacher

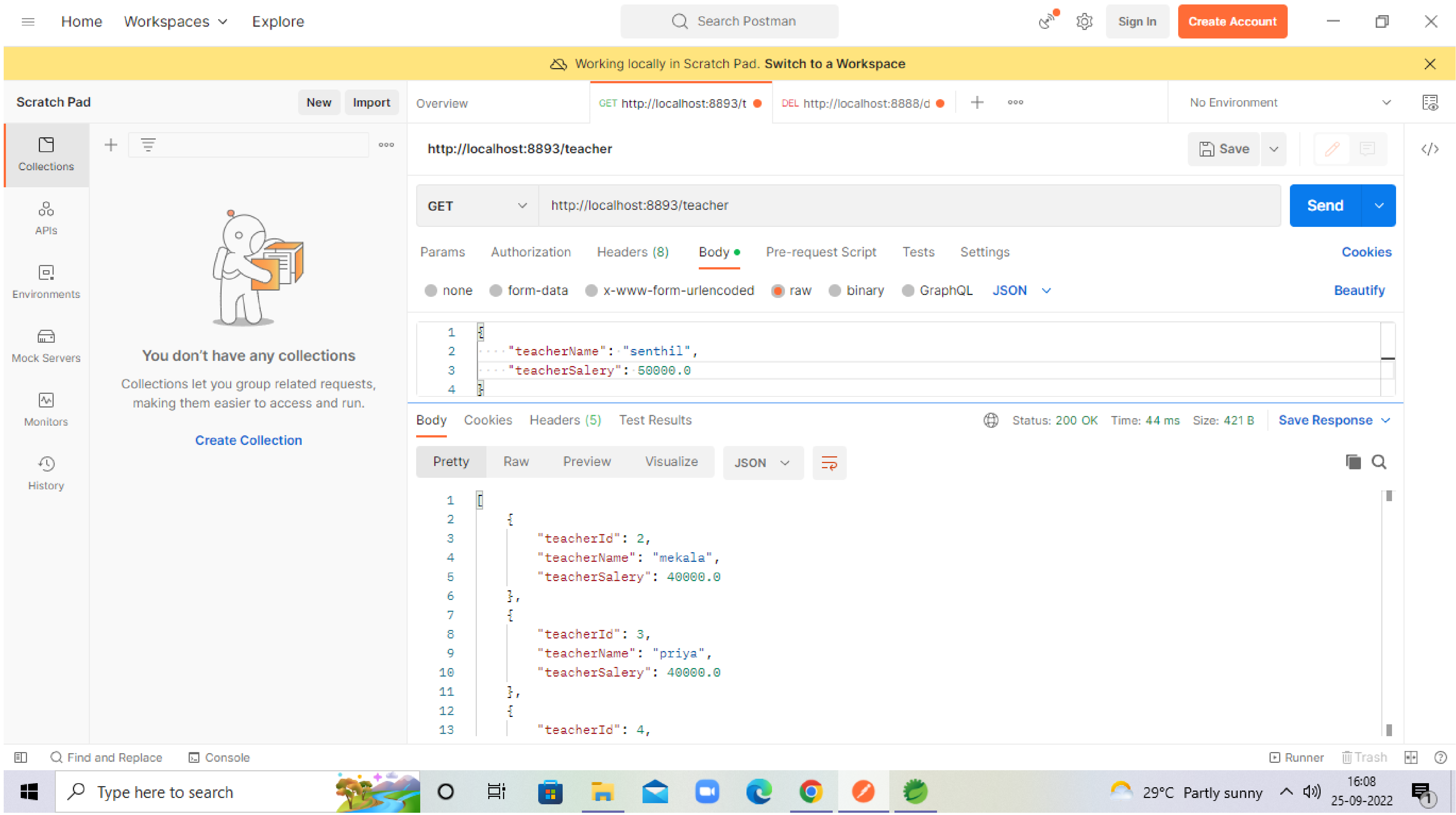
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "teacherName": "senthil",
    "teacherSalery": 50000.0
}
```

Status: 201 Created Time: 142 ms Size: 232 B

```
{
    "teacherId": 5,
    "teacherName": "senthil",
    "teacherSalery": 50000.0
}
```

Get All Teacher:



http://localhost:8893/teacher

GET http://localhost:8893/teacher Send

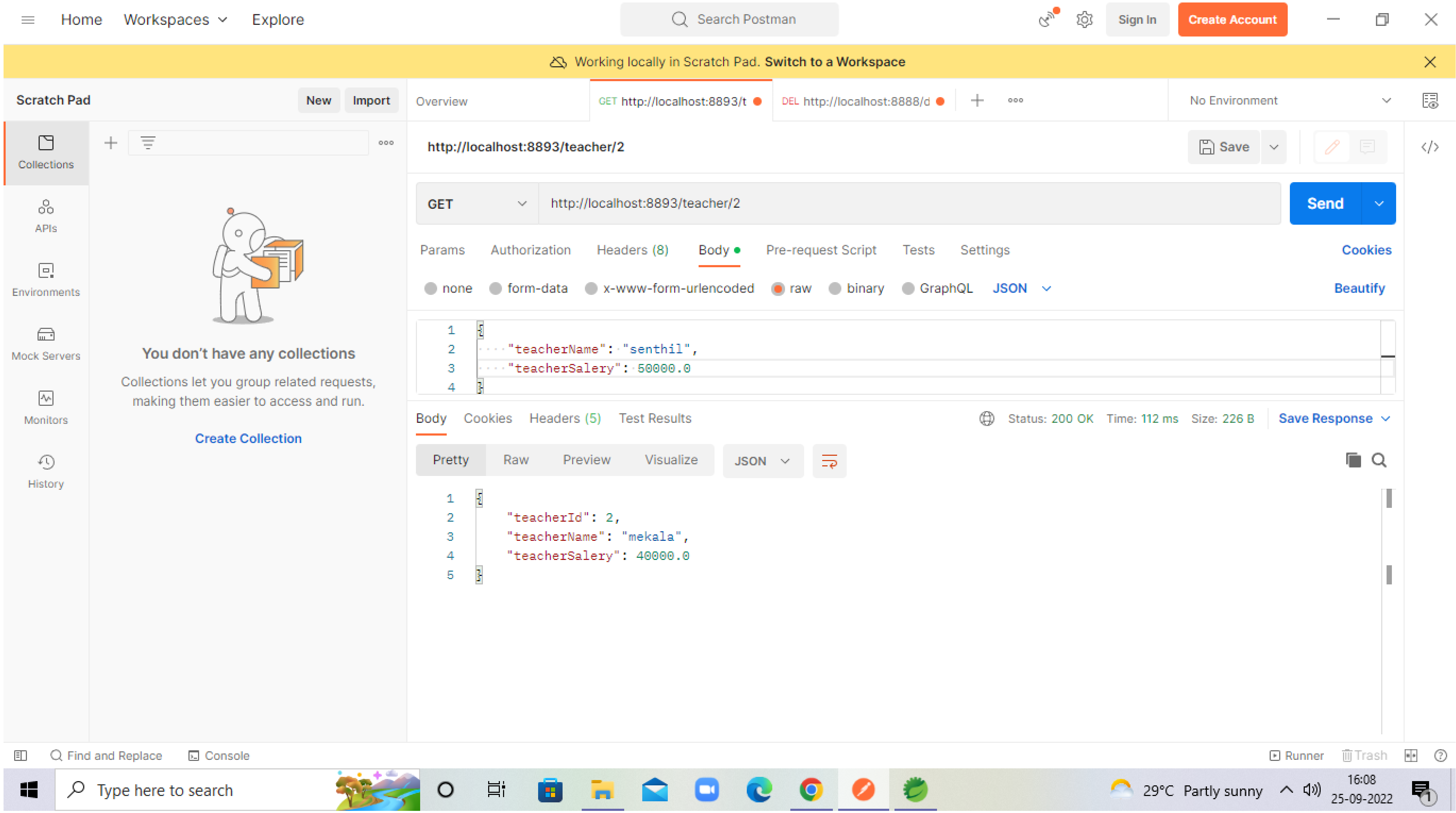
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "teacherName": "senthil",
    "teacherSalery": 50000.0
}
```

Body Cookies Headers (5) Test Results — Status: 200 OK Time: 44 ms Size: 421 B

```
[
    {
        "teacherId": 2,
        "teacherName": "mekala",
        "teacherSalery": 40000.0
    },
    {
        "teacherId": 3,
        "teacherName": "priya",
        "teacherSalery": 40000.0
    },
    {
        "teacherId": 4,
```

Get Teeacer By Id:



http://localhost:8893/teacher/2

GET http://localhost:8893/teacher/2

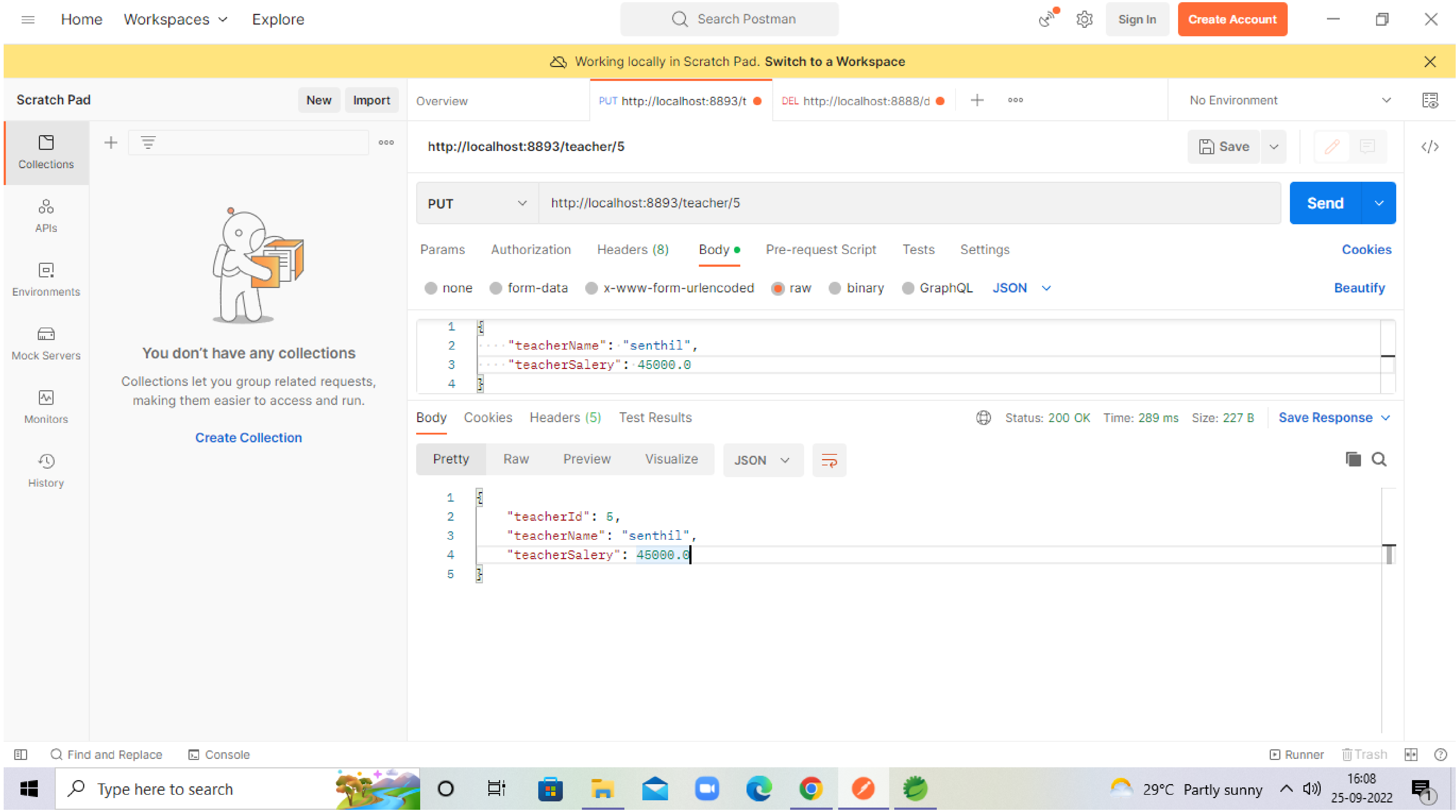
Body: ● raw JSON

```
{
    "teacherName": "senthil",
    "teacherSalery": 50000.0
}
```

Status: 200 OK Time: 112 ms Size: 226 B

```
{
    "teacherId": 2,
    "teacherName": "mekala",
    "teacherSalery": 40000.0
}
```

Update Teacher:



PUT http://localhost:8893/t

DEL http://localhost:8888/d

http://localhost:8893/teacher/5

PUT http://localhost:8893/teacher/5

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "teacherName": "senthil",
    "teacherSalery": 45000.0
}
```

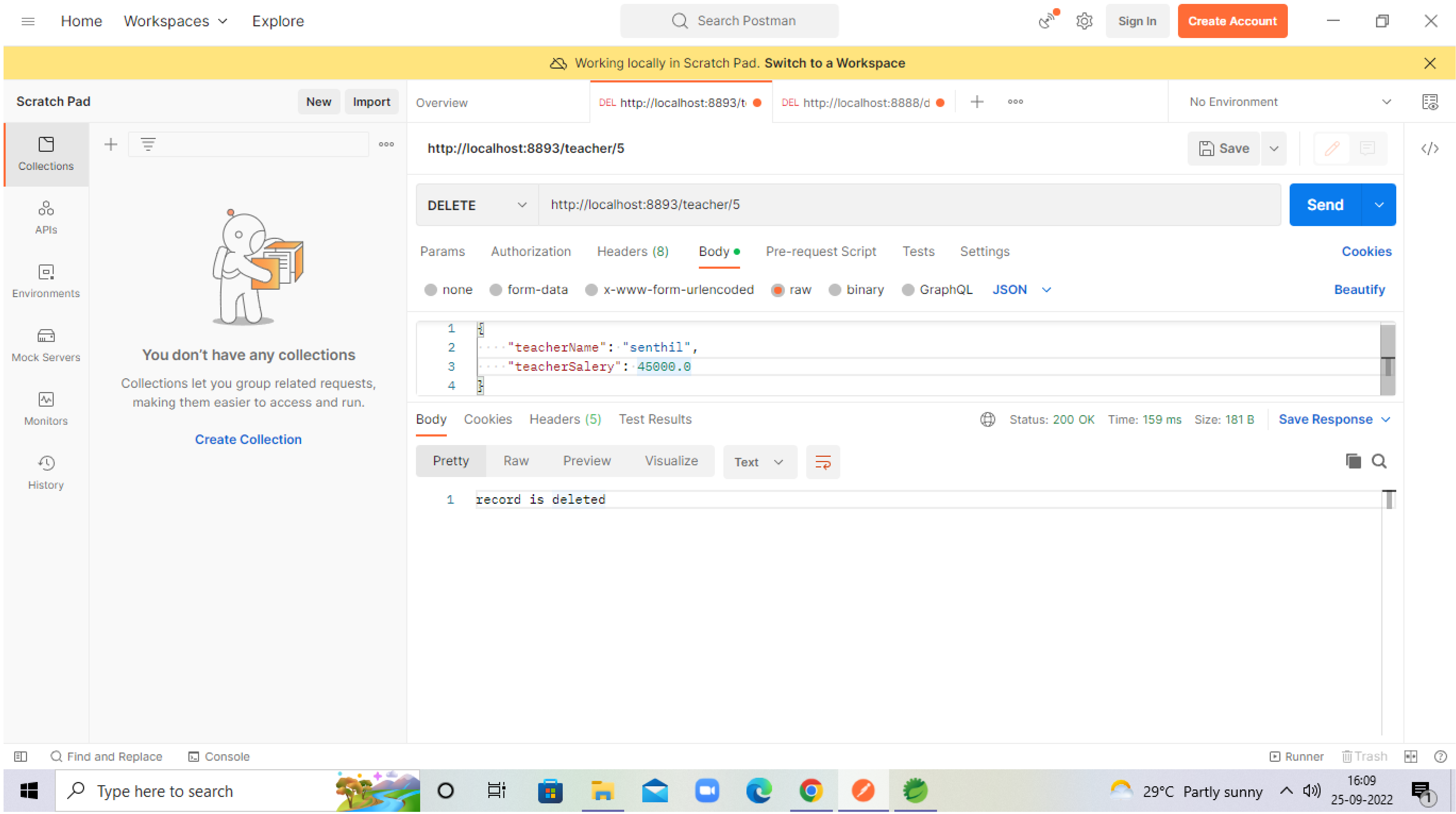
Body

Status: 200 OK Time: 289 ms Size: 227 B

JSON

```
{
    "teacherId": 5,
    "teacherName": "senthil",
    "teacherSalery": 45000.0
}
```

Delete Teacher By Id:



http://localhost:8893/teacher/5

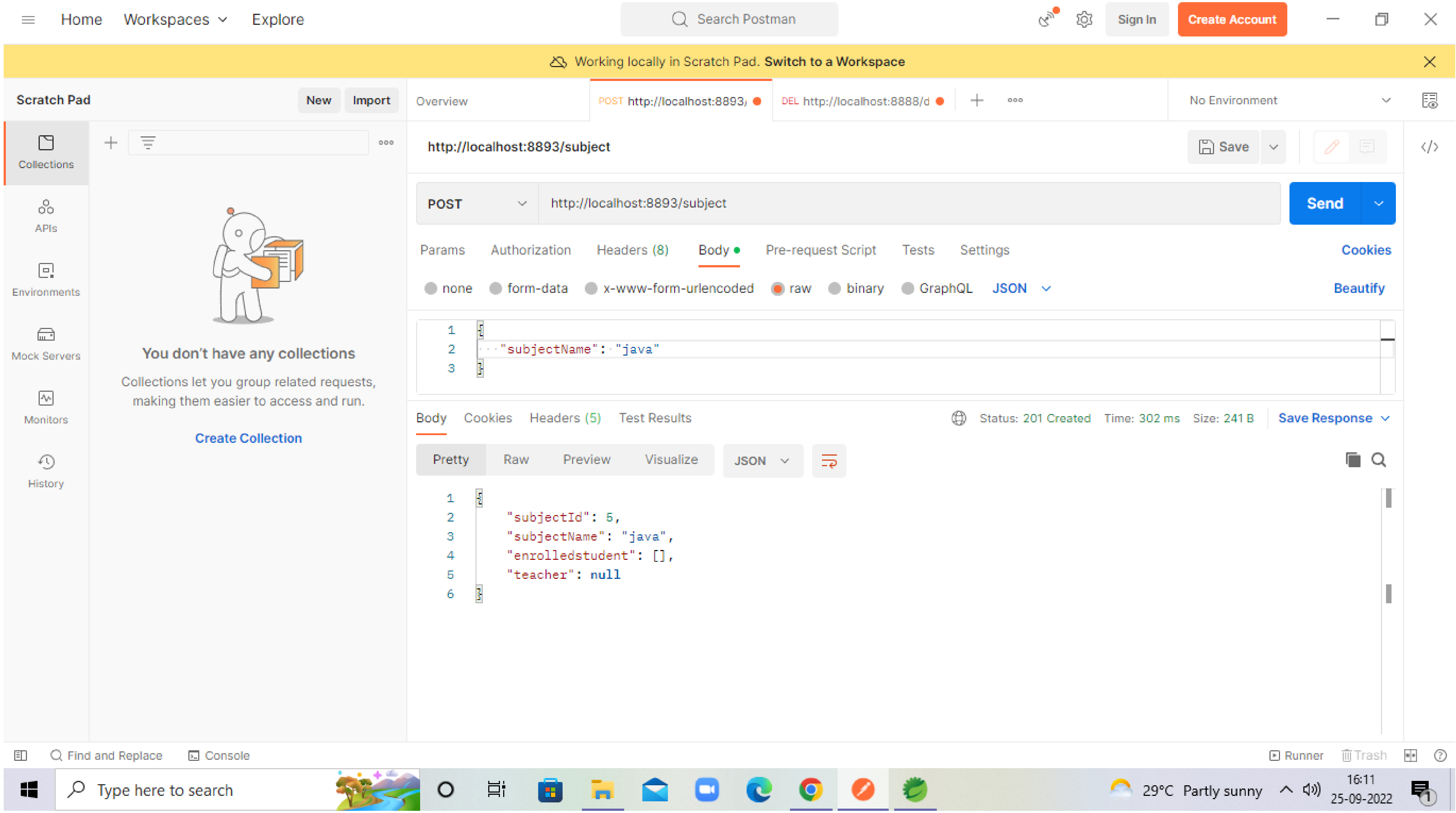
DELETE http://localhost:8893/teacher/5

```
{
    "teacherName": "senthil",
    "teacherSalery": 45000.0
}
```

Status: 200 OK Time: 159 ms Size: 181 B

```
record is deleted
```

Add Subject:



http://localhost:8893/subject

POST http://localhost:8893/subject

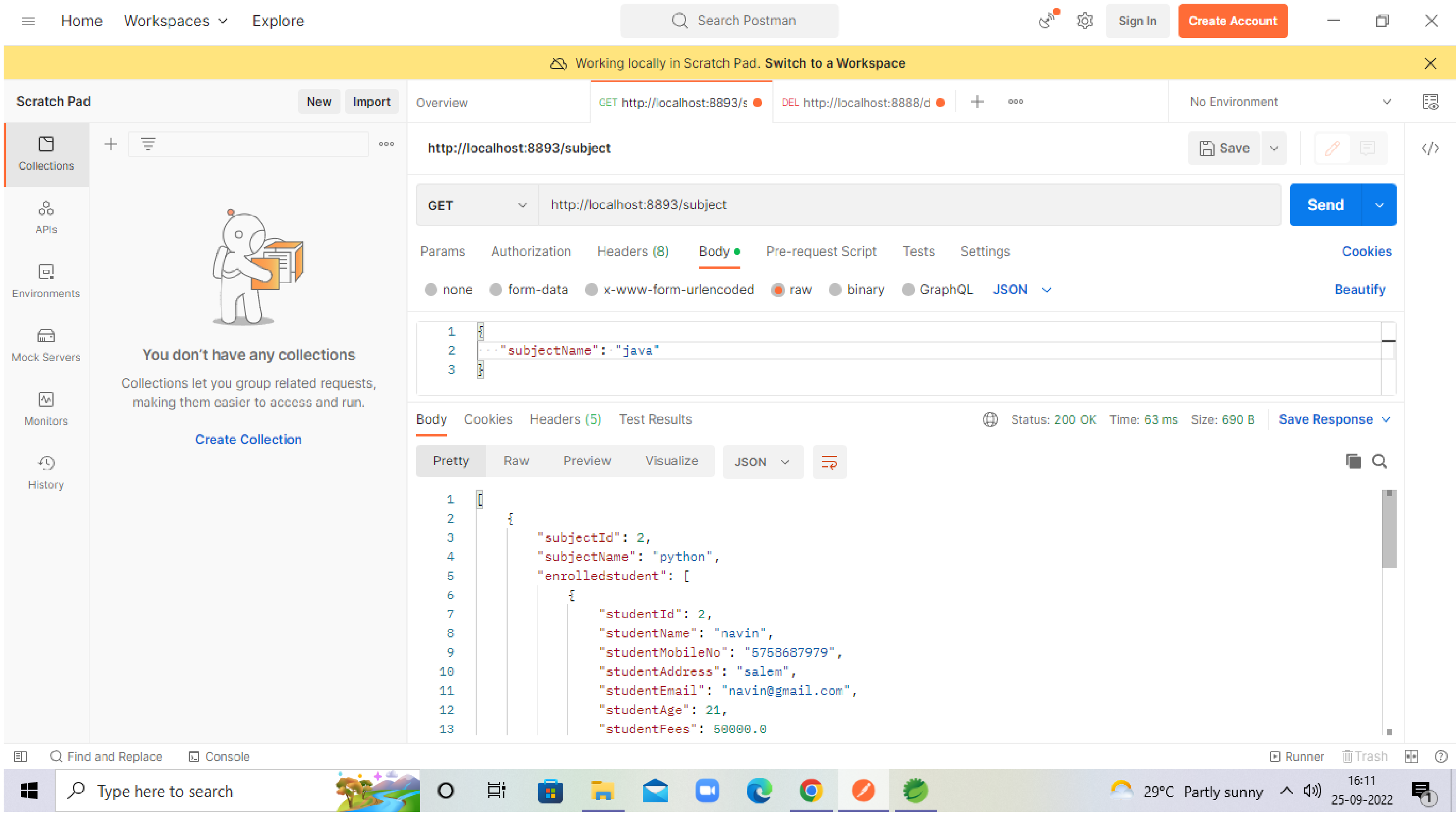
Body: ● raw JSON

```
{
  "subjectName": "java"
}
```

Status: 201 Created Time: 302 ms Size: 241 B

```
{
    "subjectId": 5,
    "subjectName": "java",
    "enrolledstudent": [],
    "teacher": null
}
```

Get All Subject:



http://localhost:8893/subject

GET http://localhost:8893/subject

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

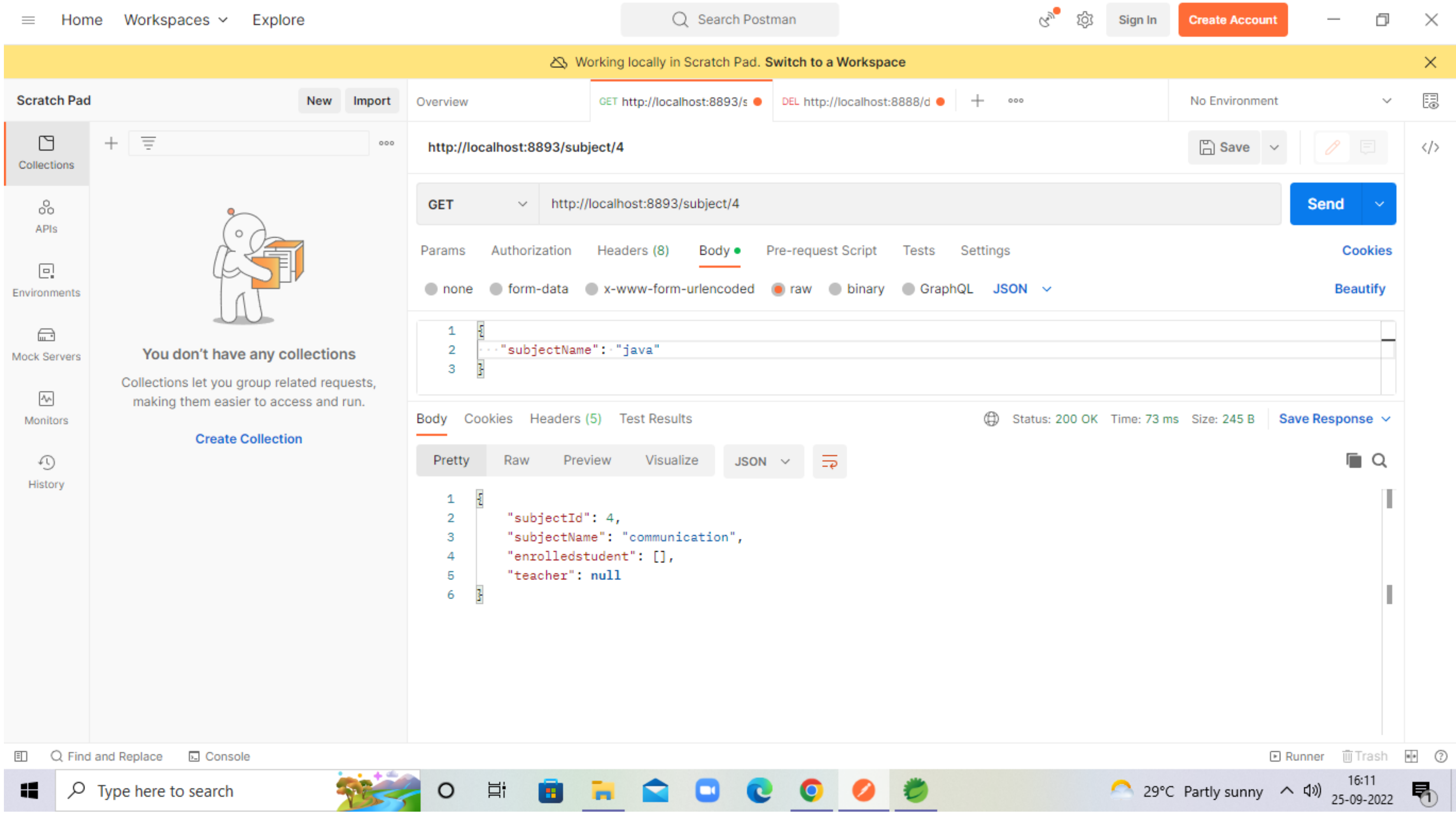
```
{
    "subjectName": "java"
}
```

Body   Status: 200 OK   Time: 63 ms   Size: 690 B

JSON

```
[
    {
        "subjectId": 2,
        "subjectName": "python",
        "enrolledstudent": [
            {
                "studentId": 2,
                "studentName": "navin",
                "studentMobileNo": "5758687979",
                "studentAddress": "salem",
                "studentEmail": "navin@gmail.com",
                "studentAge": 21,
                "studentFees": 50000.0
```

Get Subject By Id:



http://localhost:8893/subject/4

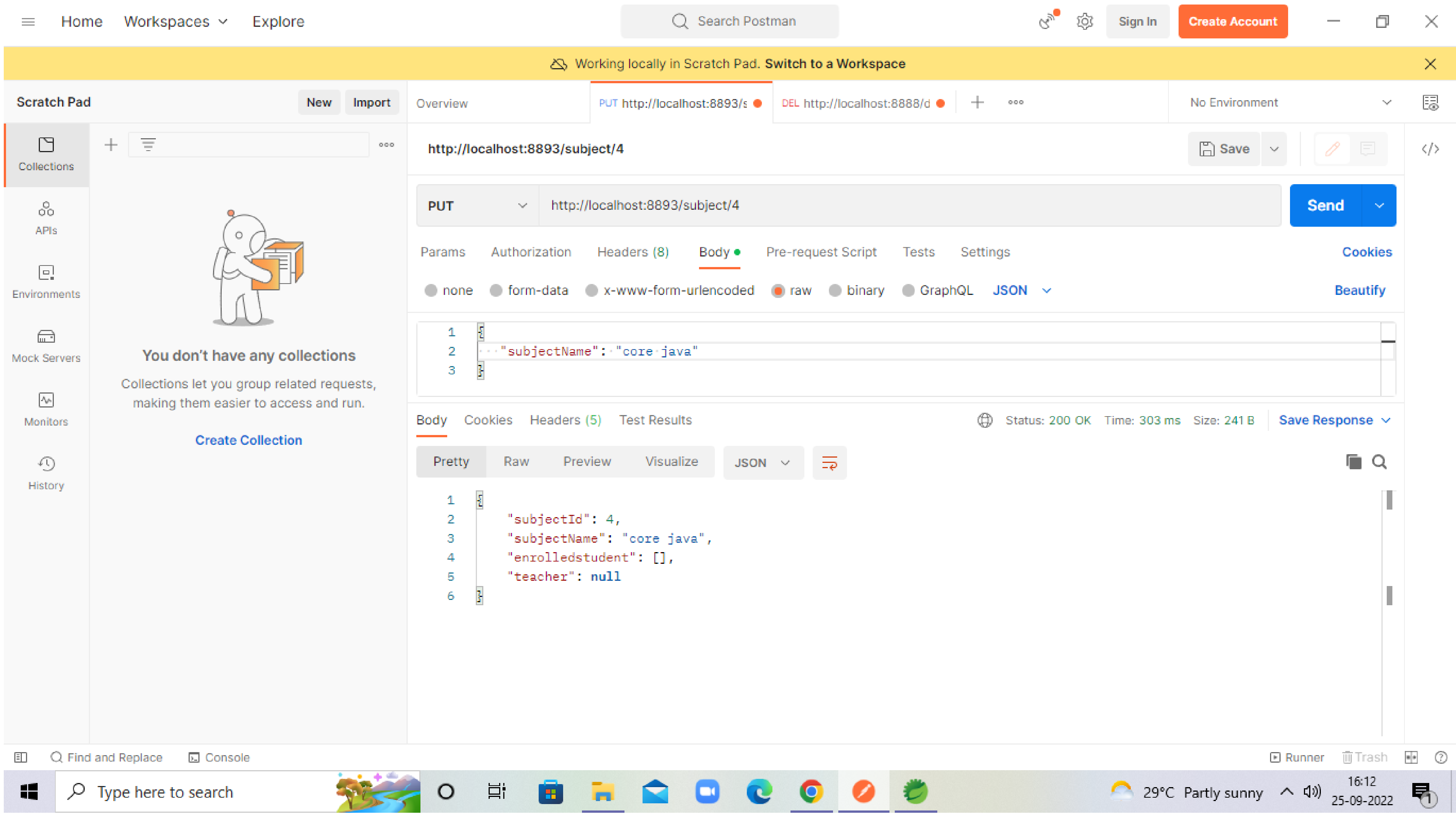
GET http://localhost:8893/subject/4

```
{
    "subjectName": "java"
}
```

Status: 200 OK   Time: 73 ms   Size: 245 B

```
{
    "subjectId": 4,
    "subjectName": "communication",
    "enrolledstudent": [],
    "teacher": null
}
```

Update Subject:



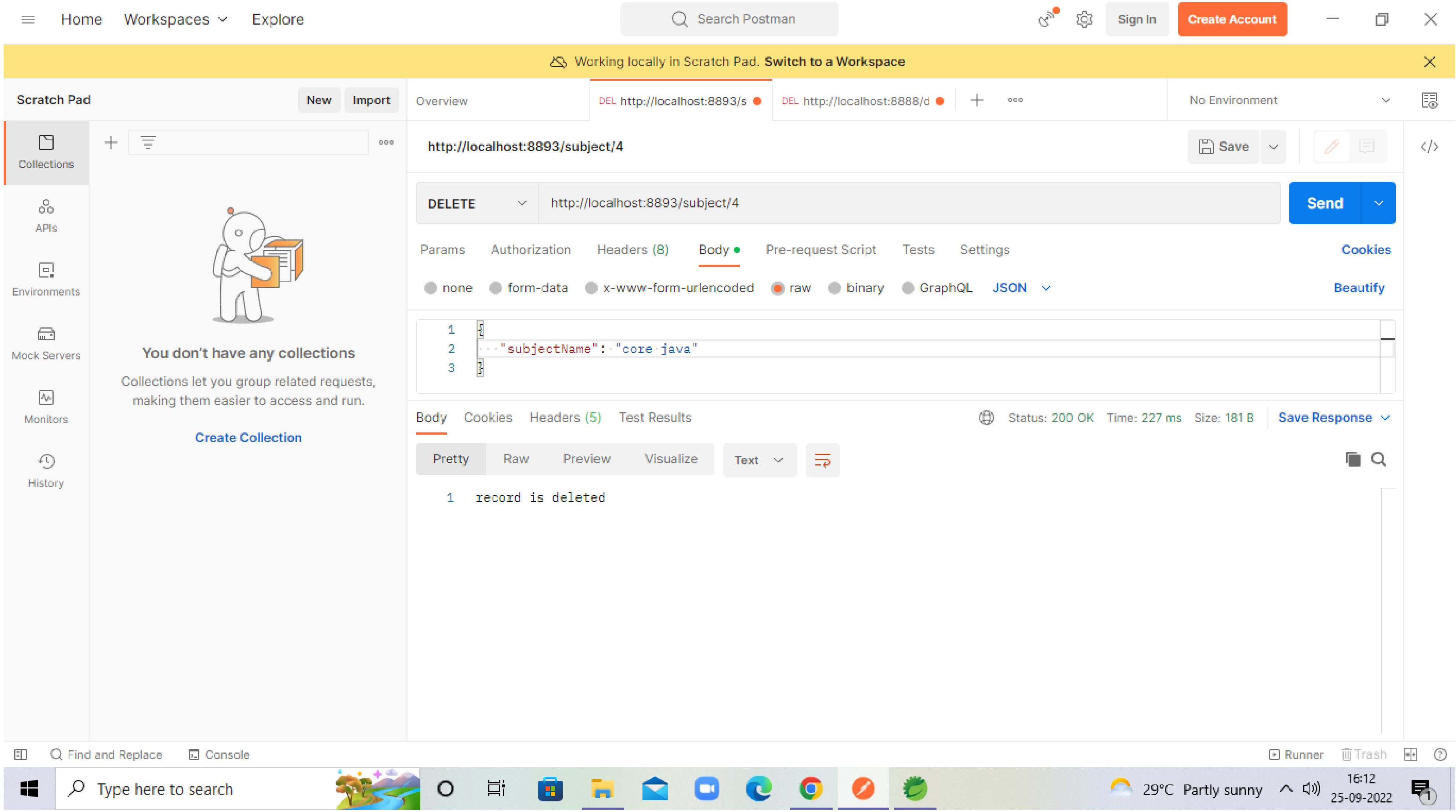
PUT http://localhost:8893/subject/4

```
{
  "subjectName": "core java"
}
```

Status: 200 OK Time: 303 ms Size: 241 B

```
{
    "subjectId": 4,
    "subjectName": "core java",
    "enrolledstudent": [],
    "teacher": null
}
```

Delete Subject By Id:



http://localhost:8893/subject/4

DELETE http://localhost:8893/subject/4

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

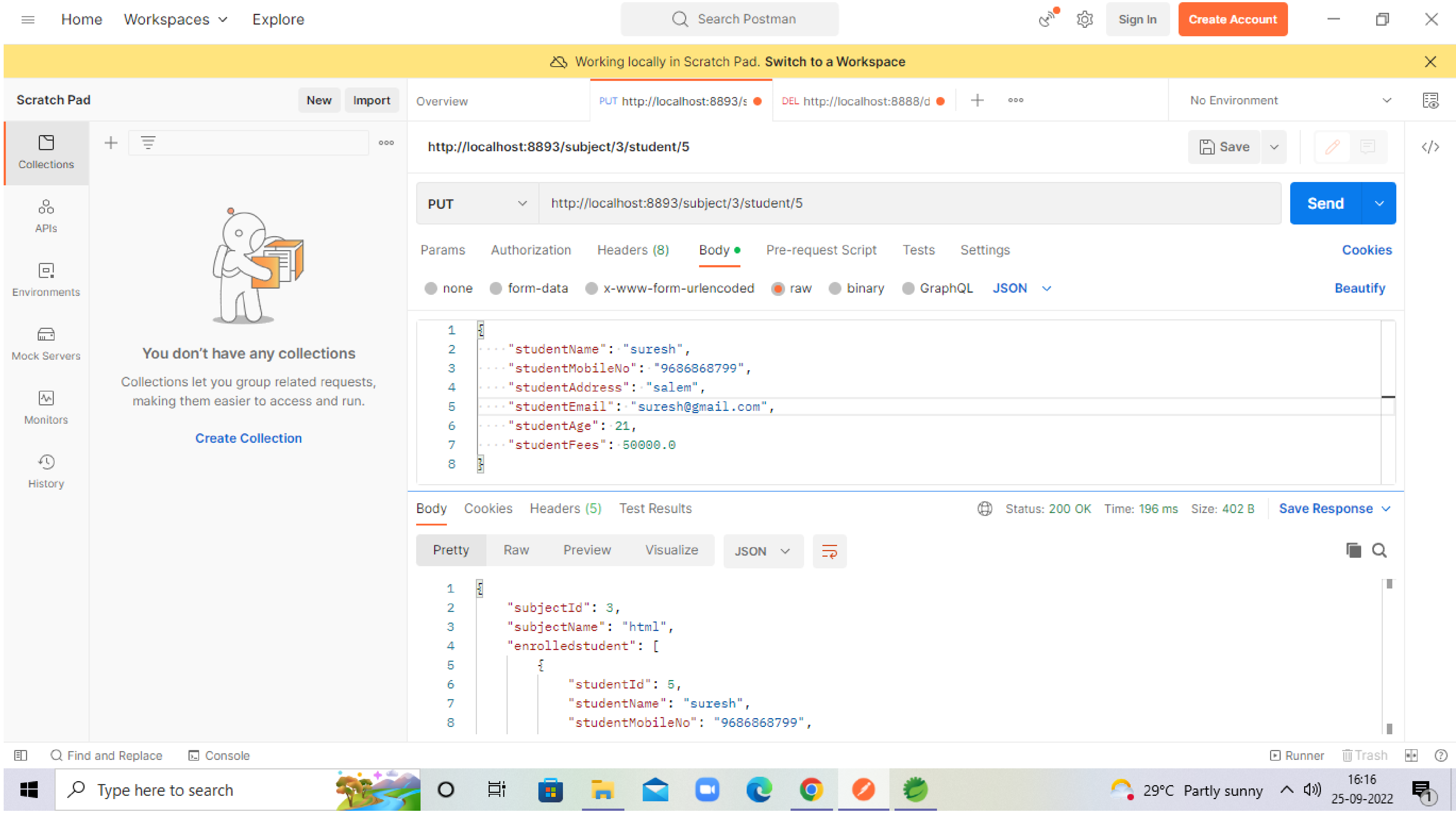
```
{
    "subjectName": "core java"
}
```

Status: 200 OK Time: 227 ms Size: 181 B

Body (Pretty, Text):

```
record is deleted
```

Assign Subject To Student:



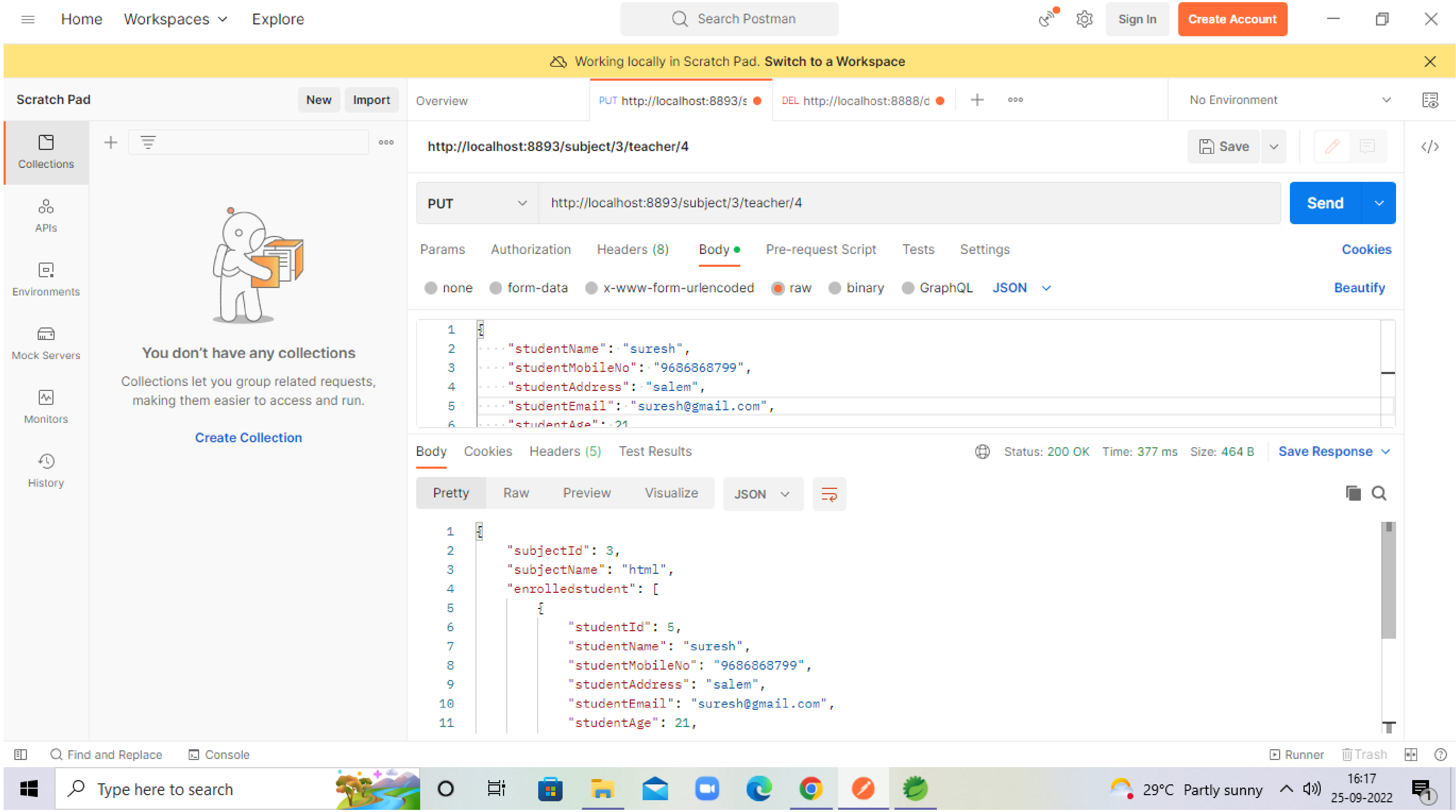
PUT http://localhost:8893/subject/3/student/5

```
{
    "studentName": "suresh",
    "studentMobileNo": "9686868799",
    "studentAddress": "salem",
    "studentEmail": "suresh@gmail.com",
    "studentAge": 21,
    "studentFees": 50000.0
}
```

Status: 200 OK Time: 196 ms Size: 402 B

```
{
    "subjectId": 3,
    "subjectName": "html",
    "enrolledstudent": [
        {
            "studentId": 5,
            "studentName": "suresh",
            "studentMobileNo": "9686868799",
```

Assign Subject To Teacher:



http://localhost:8893/subject/3/teacher/4

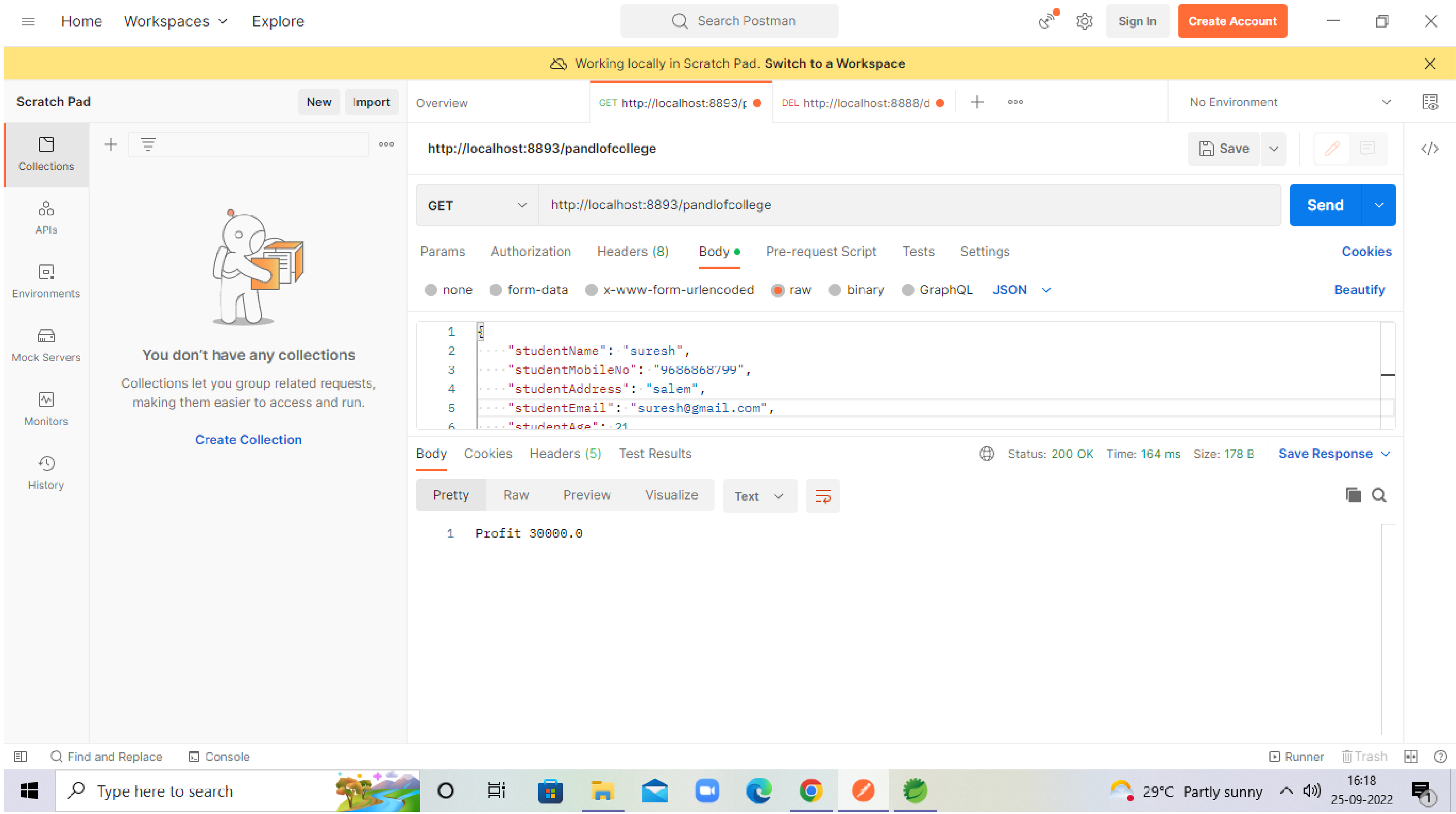
PUT http://localhost:8893/subject/3/teacher/4

```
{
    "studentName": "suresh",
    "studentMobileNo": "9686868799",
    "studentAddress": "salem",
    "studentEmail": "suresh@gmail.com",
    "studentAge": 21
```

Status: 200 OK Time: 377 ms Size: 464 B

```
{
    "subjectId": 3,
    "subjectName": "html",
    "enrolledstudent": [
        {
            "studentId": 5,
            "studentName": "suresh",
            "studentMobileNo": "9686868799",
            "studentAddress": "salem",
            "studentEmail": "suresh@gmail.com",
            "studentAge": 21,
```

Profit and Loss of College:



GET http://localhost:8893/pandlofcollege

```
{
    "studentName": "suresh",
    "studentMobileNo": "9686868799",
    "studentAddress": "salem",
    "studentEmail": "suresh@gmail.com",
    "studentAge": 21
```

Status: 200 OK Time: 164 ms Size: 178 B

```
Profit 30000.0
```

Conclusion :

- **The project entitled as College Management System is the system that deals with the issues related to a particular institution**.

- This project is successfully implemented with all the features mentioned in system requirements specification.

- Awareness and right information about any college is essentialfor both the development of student as well as faculty.

- So this serves the right purpose in achieving the desired requirements of both the communities.