Documentation for Flask REST API

This part of documentation involves the test cases and expected outcomes.

- GET: should be for data retrieval.
- POST: should be used to create a new resource.
- PUT: should be used to update information about a specific resource.
- DELETE: should be used to delete a particular resource.
- PATCH: should be used to update partial information about a particular resource.

The URLs for

Admin Login: http://127.0.0.1:5000/admin/login

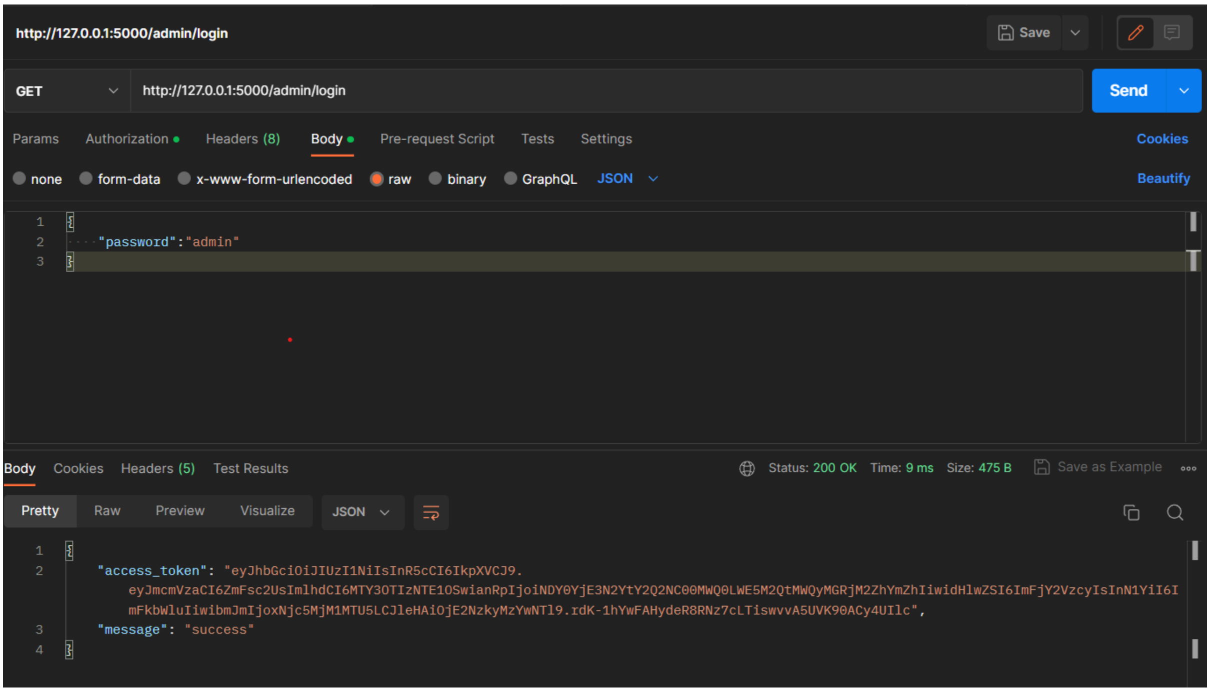
Student Login: http://127.0.0.1:5000/student/login

Student to view course list: http://127.0.0.1:5000/student/list

Student API: http://127.0.0.1:5000/student

Admin API: http://127.0.0.1:5000/admin

Case 1:



http://127.0.0.1:5000/admin/login

GET http://127.0.0.1:5000/admin/login

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

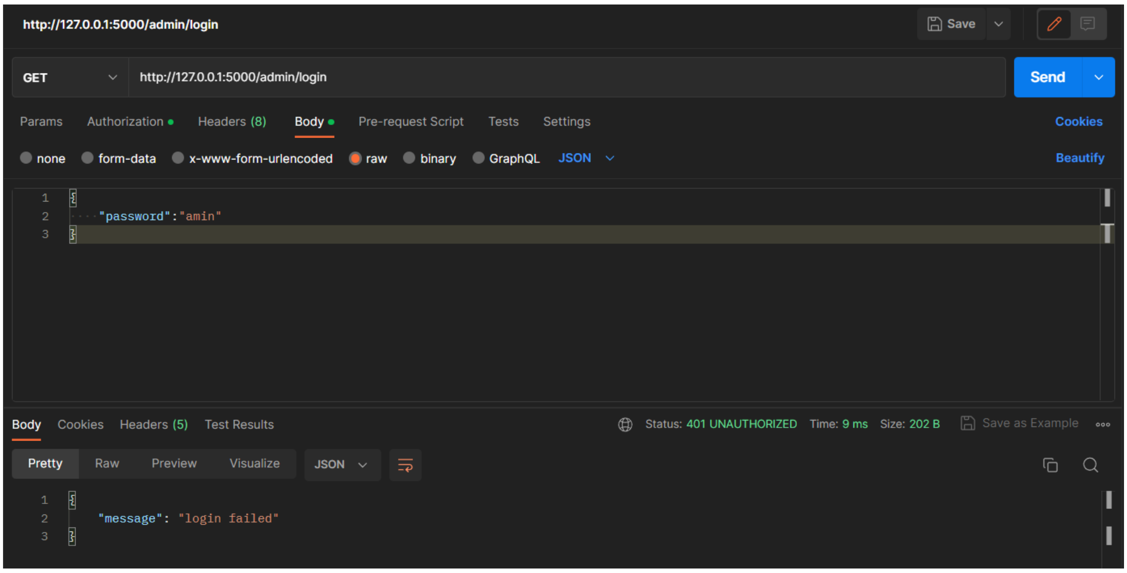
```
{
    "password":"admin"
}
```

Status: 200 OK Time: 9 ms Size: 475 B

```
{
    "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY3OTIzNTE1OSwianRpIjoiNDY0YjE3N2YtY2Q2NC00MWQ0LWE5M2QtMWQyMGRjM2ZhYmZhIiwidHlwZSI6ImFjY2VzcyIsInN1YiI6I
        mFkbWluIiwibmJmIjoxNjc5MjM1MTU5LCJleHAiOjE2NzkyMzYwNTl9.rdK-1hYwFAHydeR8RNz7cLTiswvvA5UVK90ACy4UIlc",
    "message": "success"
}
```

When we try to login with the credentials.

And when the password matches with the correct password we get the access token using which we can perform the CRUD operations of admin.



http://127.0.0.1:5000/admin/login

GET http://127.0.0.1:5000/admin/login

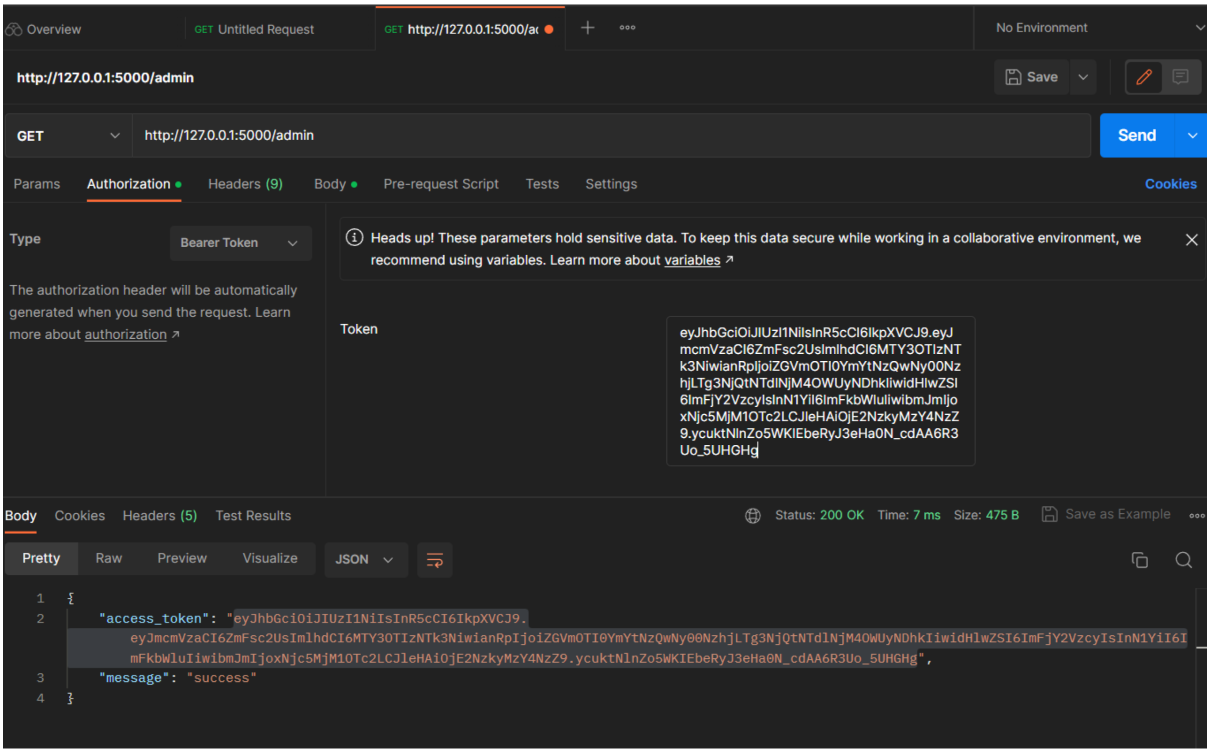
```
{
    "password":"amin"
}
```

Status: 401 UNAUTHORIZED Time: 9 ms Size: 202 B

```
{
    "message": "login failed"
}
```

This the case when the password doesn’t match. We receive a message stating that the login failed

Case 2:



http://127.0.0.1:5000/admin

GET http://127.0.0.1:5000/admin

Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

ⓘ Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

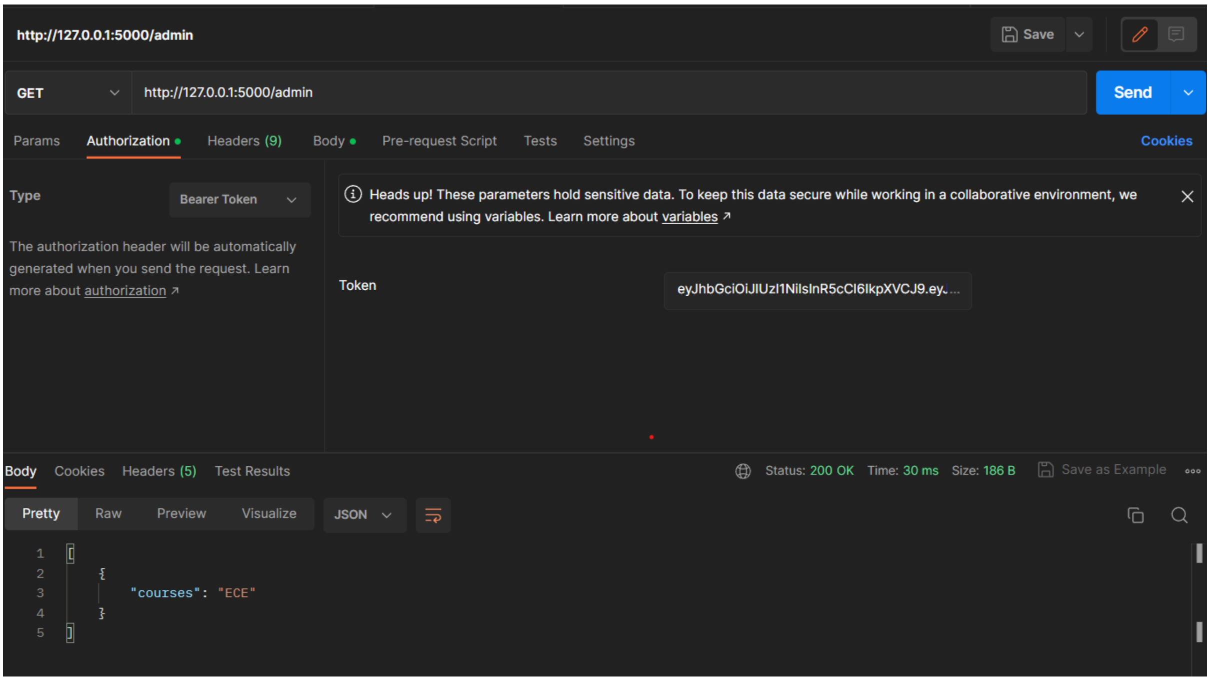
Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY3OTIzNTk3NiwianRpIjoiZGVmOTI0YmYtNzQwNy00NzhjLTg3NjQtNTdlNjM4OWUyNDhkIiwidHlwZSI6ImFjY2VzcyIsInN1YiI6ImFkbWluIiwibmJmIjoxNjc5MjM1OTc2LCJleHAiOjE2NzkyMzY4NzZ9.ycuktNlnZo5WKIEbeRyJ3eHa0N_cdAA6R3Uo_5UHGHg

Status: 200 OK Time: 7 ms Size: 475 B

```
{
    "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTY3OTIzNTk3NiwianRpIjoiZGVmOTI0YmYtNzQwNy00NzhjLTg3NjQtNTdlNjM4OWUyNDhkIiwidHlwZSI6ImFjY2VzcyIsInN1YiI6I
        mFkbWluIiwibmJmIjoxNjc5MjM1OTc2LCJleHAiOjE2NzkyMzY4NzZ9.ycuktNlnZo5WKIEbeRyJ3eHa0N_cdAA6R3Uo_5UHGHg",
    "message": "success"
}
```

For authorization we have copy pasted the access token in the token box of authorization

Then If we give the GET request for admin we will receive the list of courses



http://127.0.0.1:5000/admin

GET http://127.0.0.1:5000/admin

Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

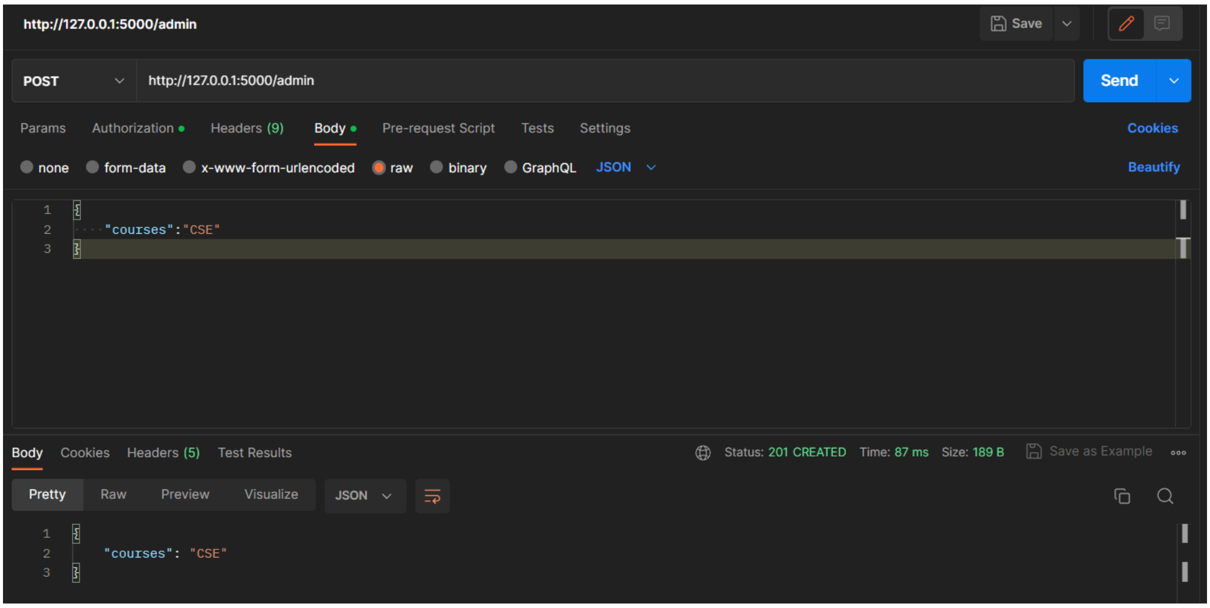
Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey...

Status: 200 OK Time: 30 ms Size: 186 B

```
[
    {
        "courses": "ECE"
    }
]
```

The courses are listed.

Case 3:



http://127.0.0.1:5000/admin

POST http://127.0.0.1:5000/admin

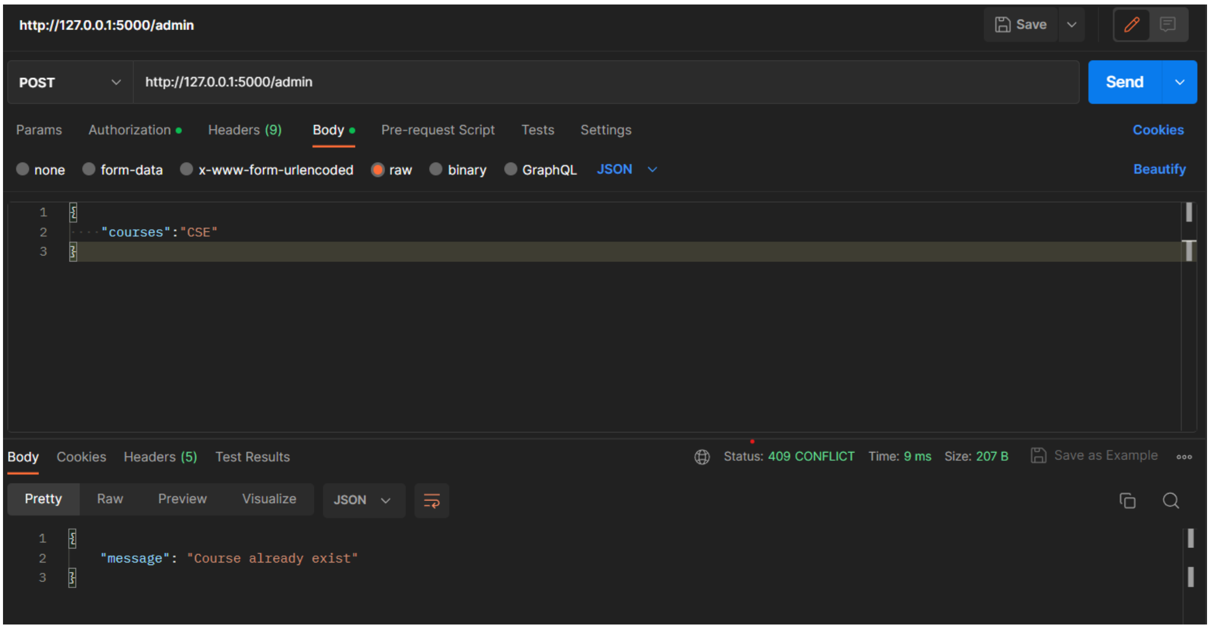
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "courses":"CSE"
}
```

Status: 201 CREATED Time: 87 ms Size: 189 B

```
{
    "courses": "CSE"
}
```

This is the POST request or the request for creating or adding the course CSE into the course table.

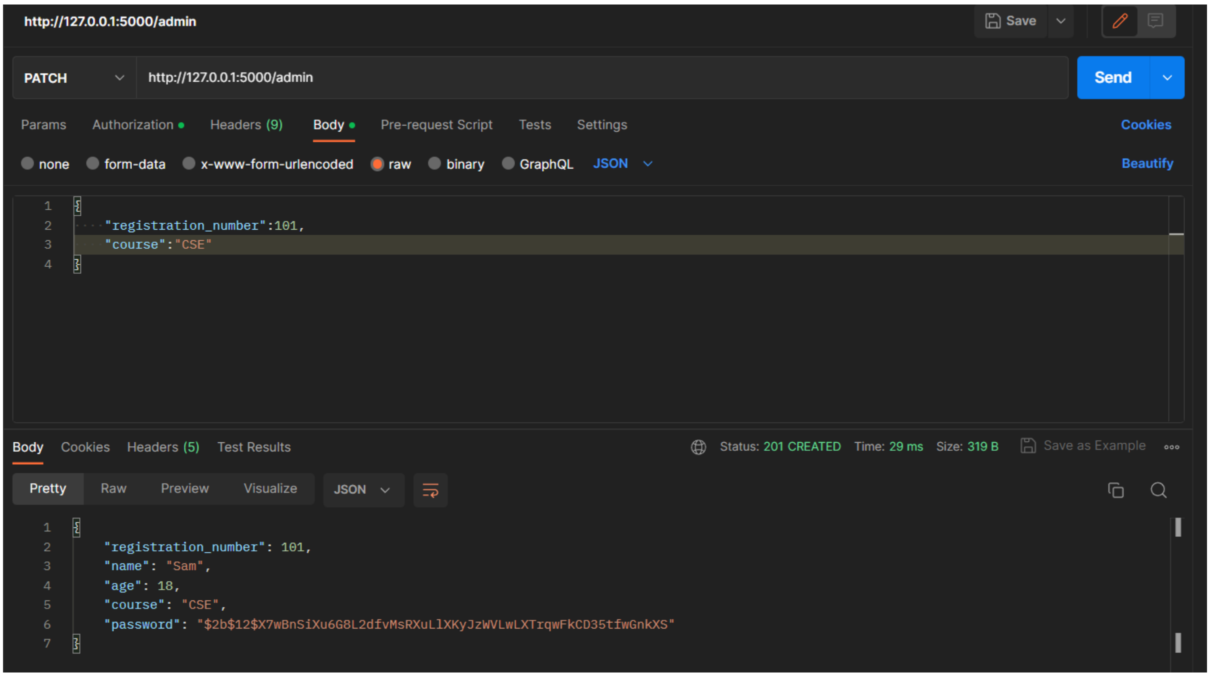


```
{
    "courses":"CSE"
}
```

```
{
    "message": "Course already exist"
}
```

When we try to add the same course, we receive a message stating that the course already exists.

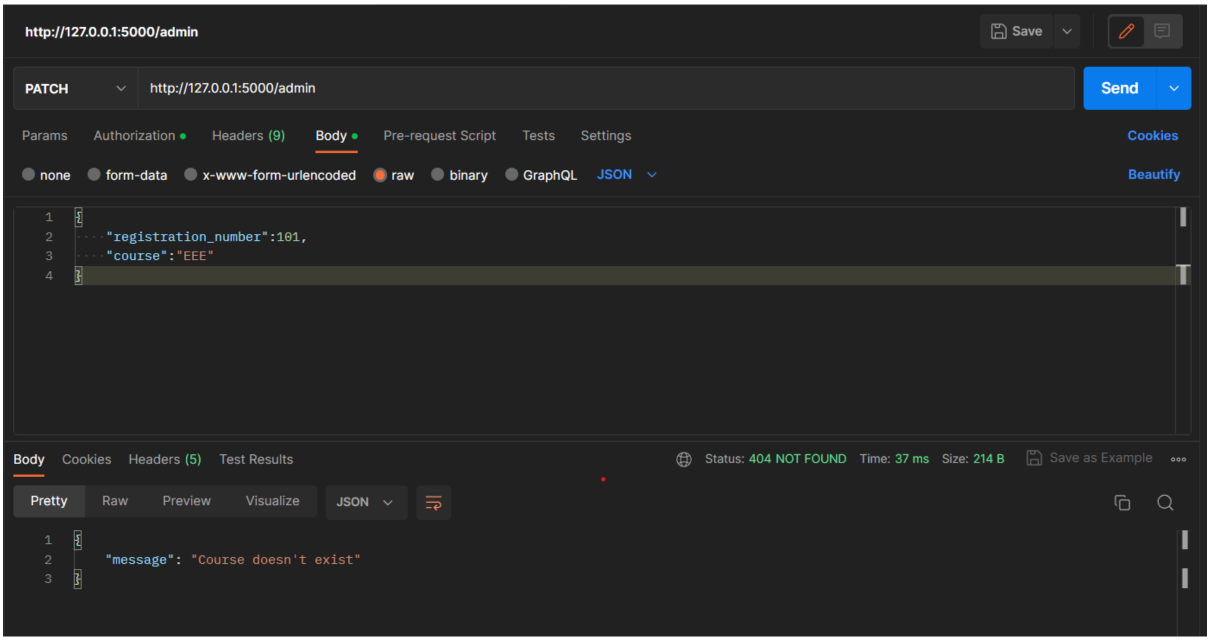
Case 4:



```
{
    "registration_number":101,
    "course":"CSE"
}
```

```
{
    "registration_number": 101,
    "name": "Sam",
    "age": 18,
    "course": "CSE",
    "password": "$2b$12$X7wBnSiXu6G8L2dfvMsRXuLlXKyJzWVLwLXTrqwFkCD35tfwGnkXS"
}
```

In this case we try to change the course of the student which can be done by the admin and not by the user.



PATCH http://127.0.0.1:5000/admin

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

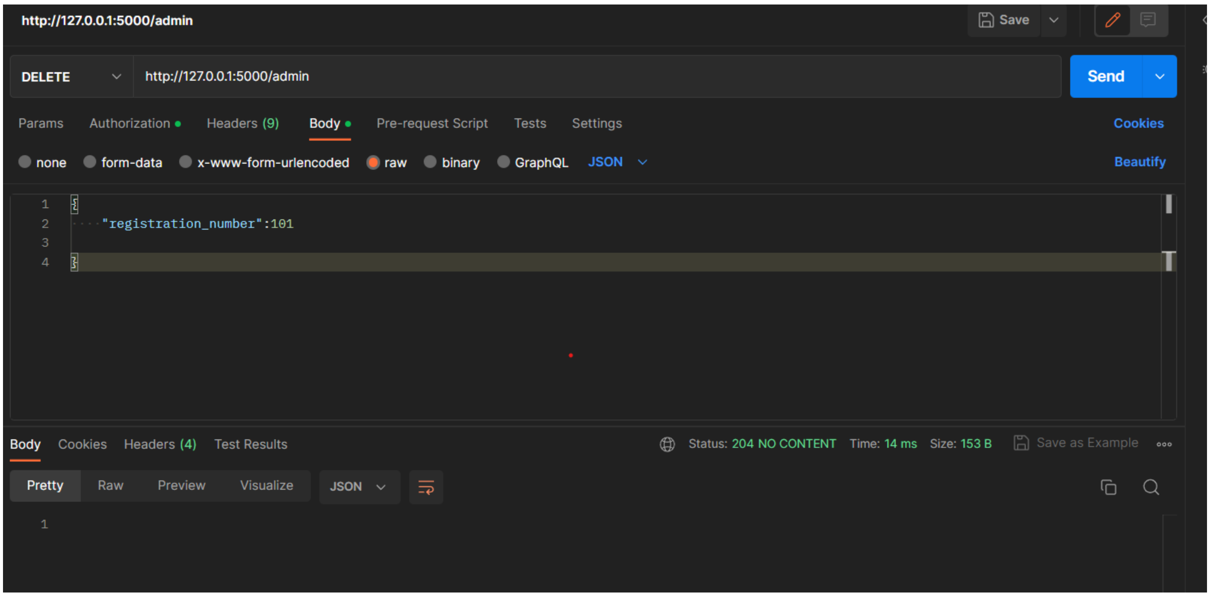
```
{
    "registration_number":101,
    "course":"EEE"
}
```

Body Cookies Headers (5) Test Results — Status: 404 NOT FOUND Time: 37 ms Size: 214 B Save as Example

```
{
    "message": "Course doesn't exist"
}
```

The message we get when we try to change the course that doesn't exist

Case 5:



DELETE http://127.0.0.1:5000/admin

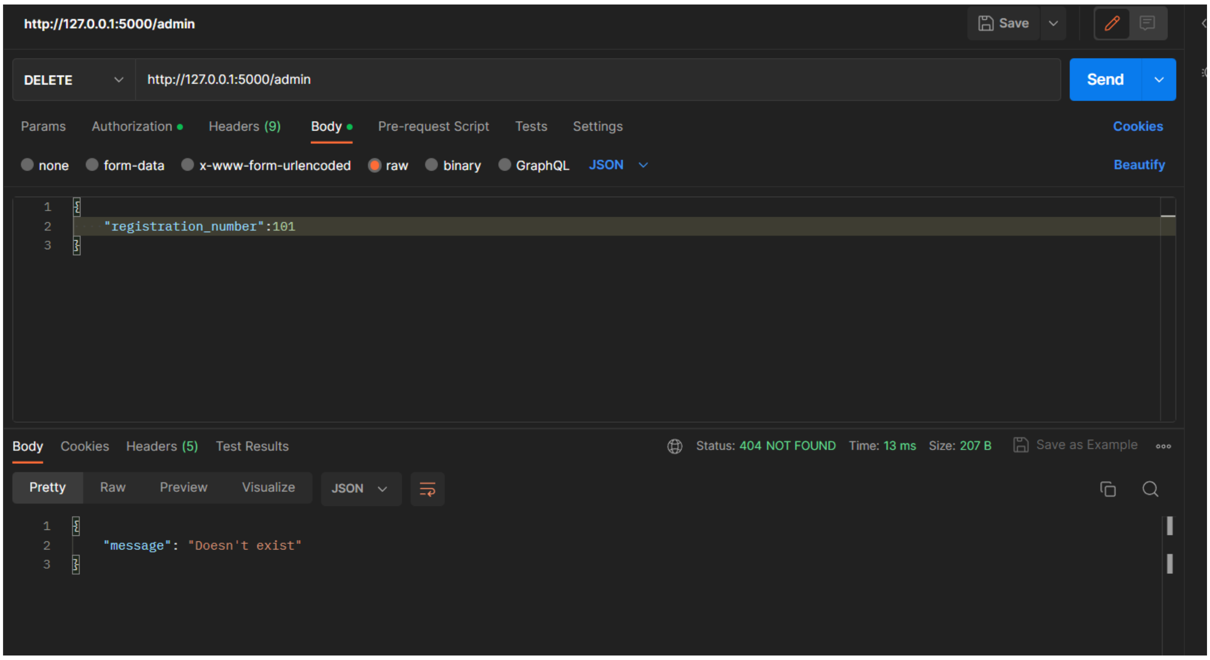
Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "registration_number":101

}
```

Body Cookies Headers (4) Test Results — Status: 204 NO CONTENT Time: 14 ms Size: 153 B Save as Example

Deleting a student's details from the student table



DELETE http://127.0.0.1:5000/admin

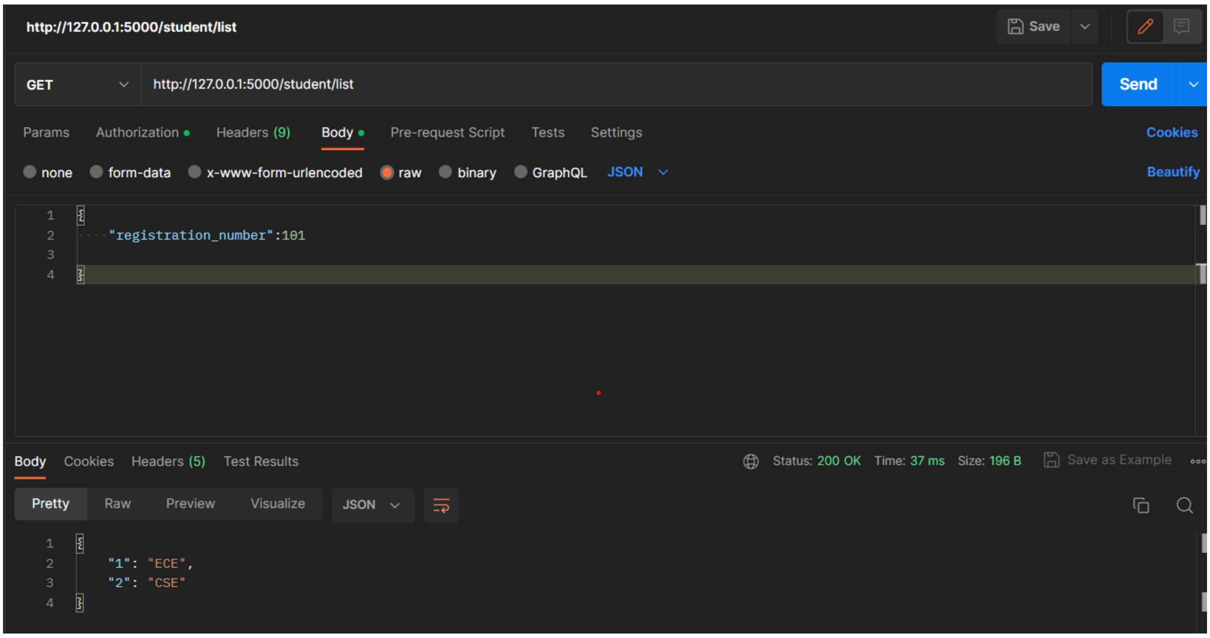
```
{
    "registration_number":101
}
```

Status: 404 NOT FOUND Time: 13 ms Size: 207 B

```
{
    "message": "Doesn't exist"
}
```

Trying to delete the same student details gives us this message.

Case 6:



GET http://127.0.0.1:5000/student/list

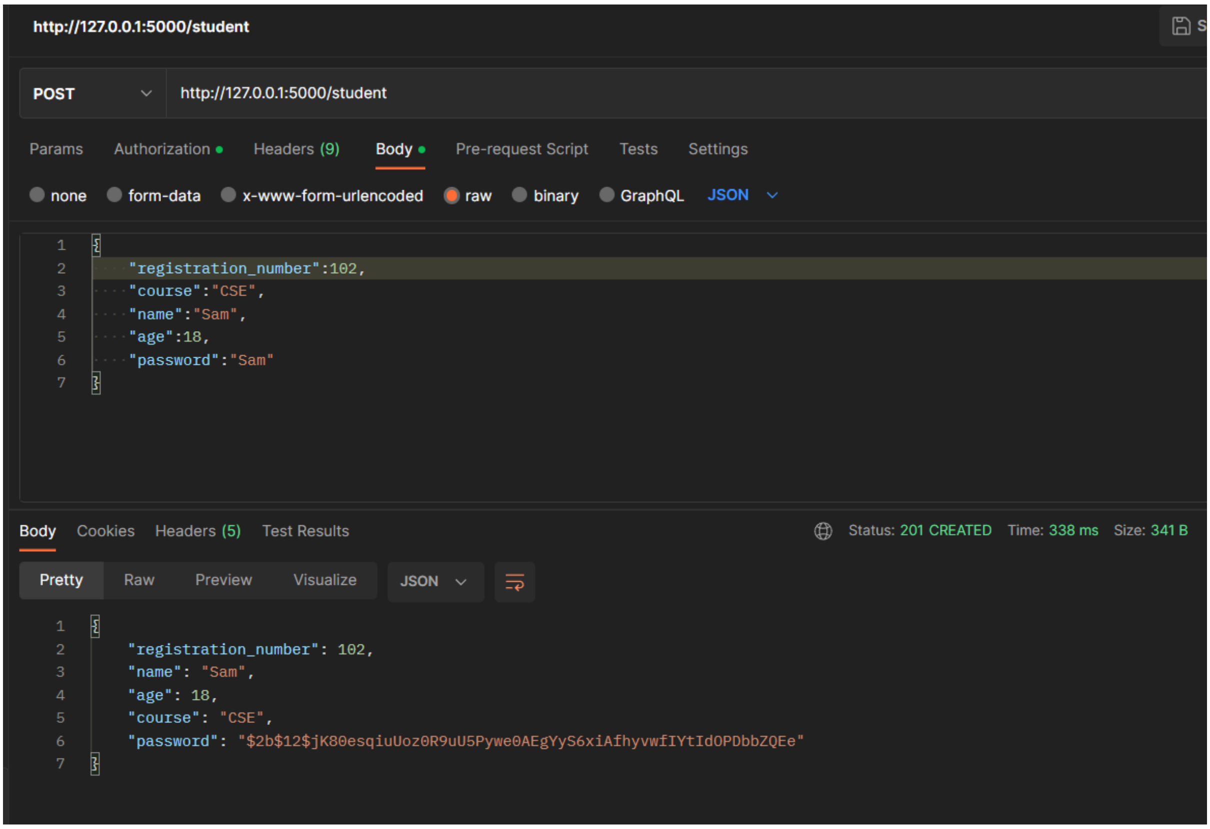
```
{
    "registration_number":101

}
```

Status: 200 OK Time: 37 ms Size: 196 B

```
{
    "1": "ECE",
    "2": "CSE"
}
```

Request for viewing the courses.



POST http://127.0.0.1:5000/student

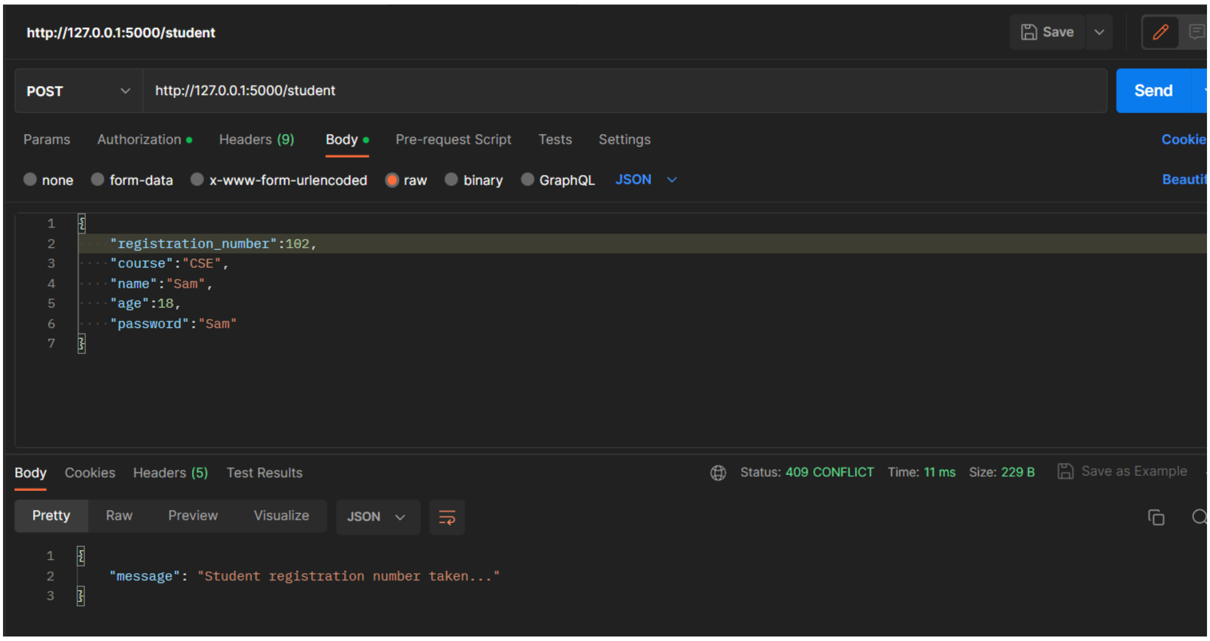
Body: raw, JSON

```
{
    "registration_number":102,
    "course":"CSE",
    "name":"Sam",
    "age":18,
    "password":"Sam"
}
```

Status: 201 CREATED Time: 338 ms Size: 341 B

```
{
    "registration_number": 102,
    "name": "Sam",
    "age": 18,
    "course": "CSE",
    "password": "$2b$12$jK80esqiuUoz0R9uU5Pywe0AEgYyS6xiAfhyvwfIYtIdOPDbbZQEe"
}
```

Adding a student’s data into data base using POST request.



POST http://127.0.0.1:5000/student

Body: raw, JSON

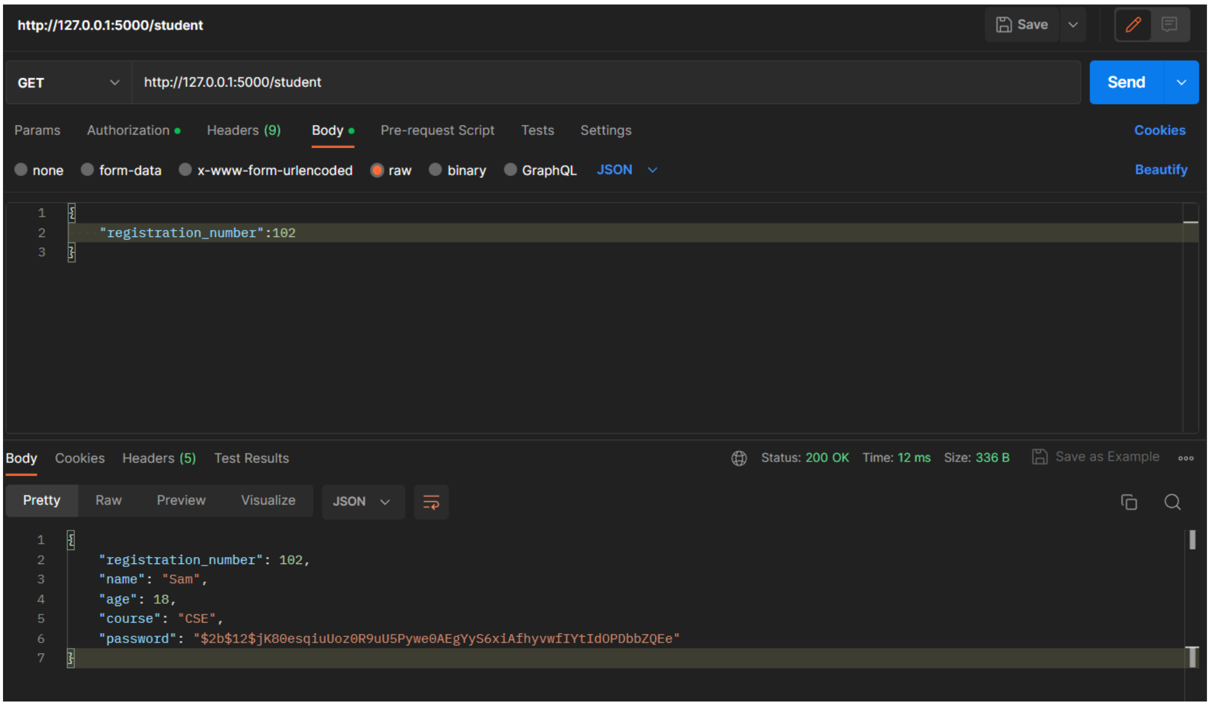
```
{
    "registration_number":102,
    "course":"CSE",
    "name":"Sam",
    "age":18,
    "password":"Sam"
}
```

Status: 409 CONFLICT Time: 11 ms Size: 229 B

```
{
    "message": "Student registration number taken..."
}
```

When we try to add the same student details we receive a message stating that the registration number is taken.

Case 7:


http://127.0.0.1:5000/student
Save
GET
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON
Beautify

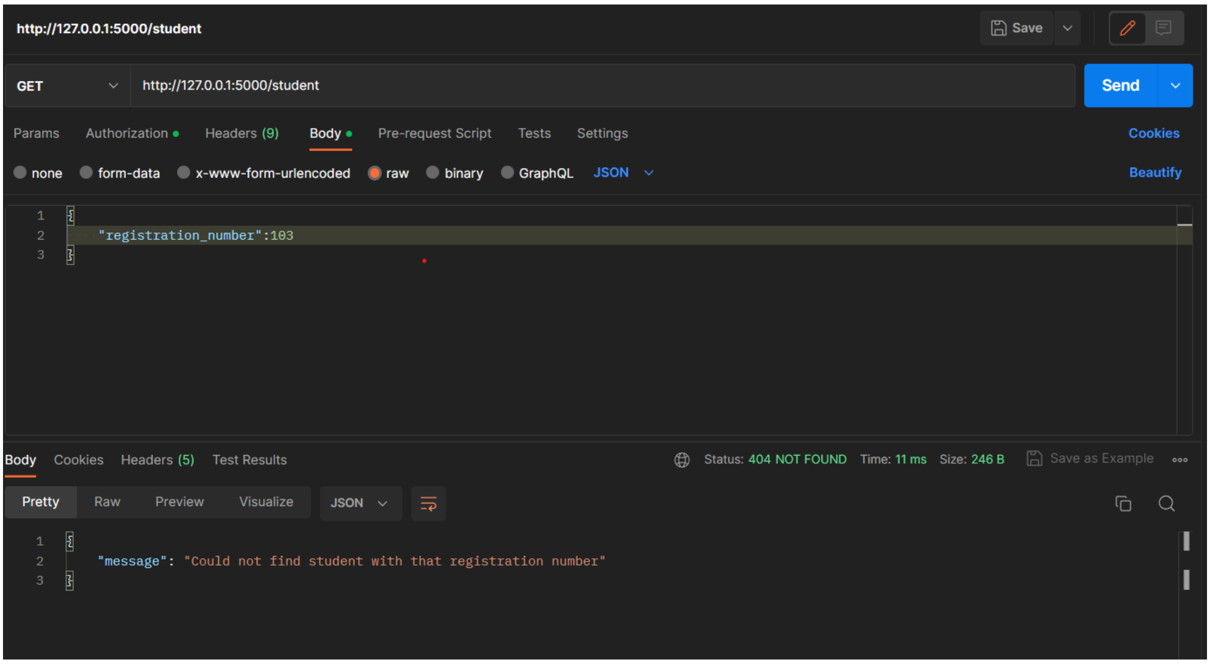
```
http://127.0.0.1:5000/student
```

```
{
    "registration_number":102
}
```

Status: 200 OK Time: 12 ms Size: 336 B

```
{
    "registration_number": 102,
    "name": "Sam",
    "age": 18,
    "course": "CSE",
    "password": "$2b$12$jK80esqiuUoz0R9uU5Pywe0AEgYyS6xiAfhyvwfIYtIdOPDbbZQEe"
}
```

When we try to display the student details using GET request we receive the information in the JSON Format


http://127.0.0.1:5000/student
Save
GET
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON
Beautify

```
http://127.0.0.1:5000/student
```

```
{
    "registration_number":103
}
```

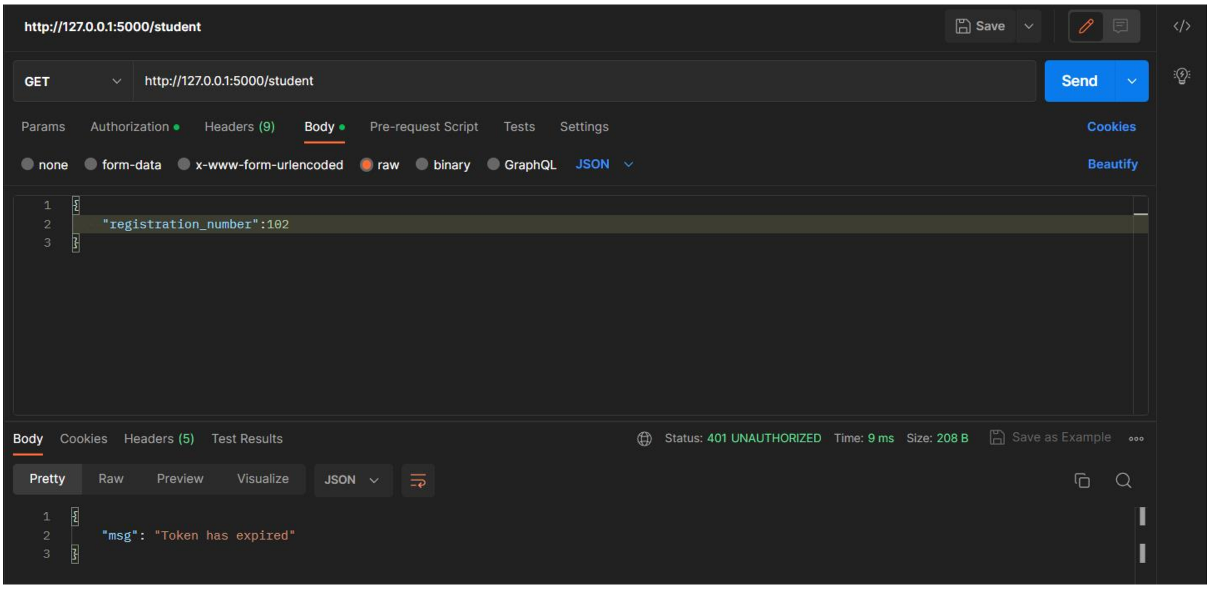
Status: 404 NOT FOUND Time: 11 ms Size: 246 B

```
{
    "message": "Could not find student with that registration number"
}
```

When the registration number doesn’t match with the numbers present in database.

We receive a message “Could not find student with that register number”.

Token Expiry Case:



http://127.0.0.1:5000/student

GET http://127.0.0.1:5000/student

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

```
{
    "registration_number":102
}
```

Status: 401 UNAUTHORIZED Time: 9 ms Size: 208 B

```
{
    "msg": "Token has expired"
}
```