Unit Testing

Frameworks used: ***Jest and Supertest***
Postman

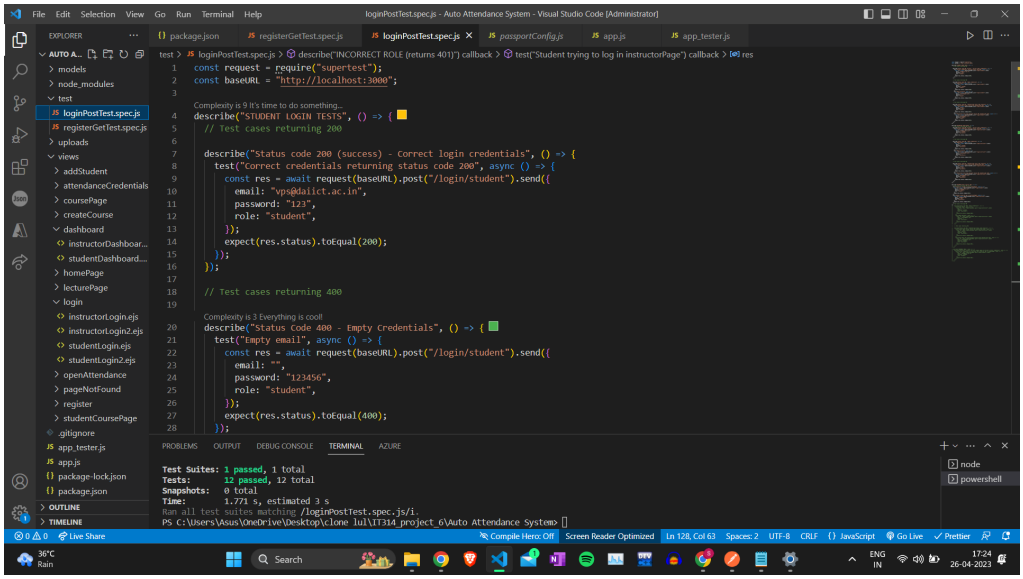
Login Unit Testing

The following testing module contains different scenarios for:

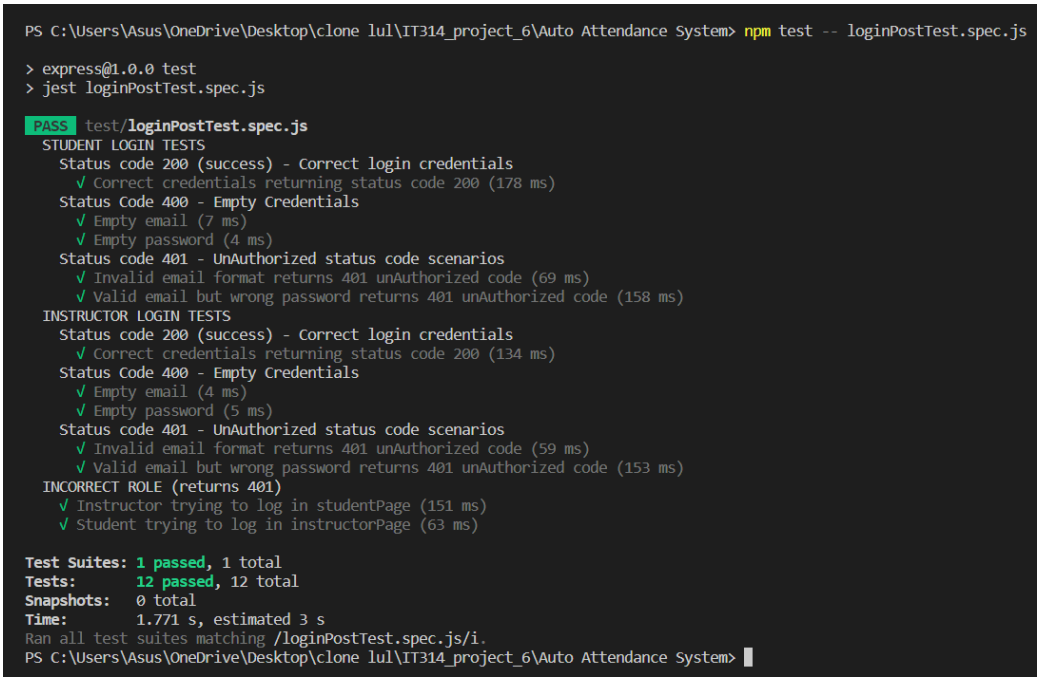
1. Instructor Login
2. Student Login

Code is saved as *loginPostTest.spec.js*

Code snippet:



Results:



The same is done for the register module.

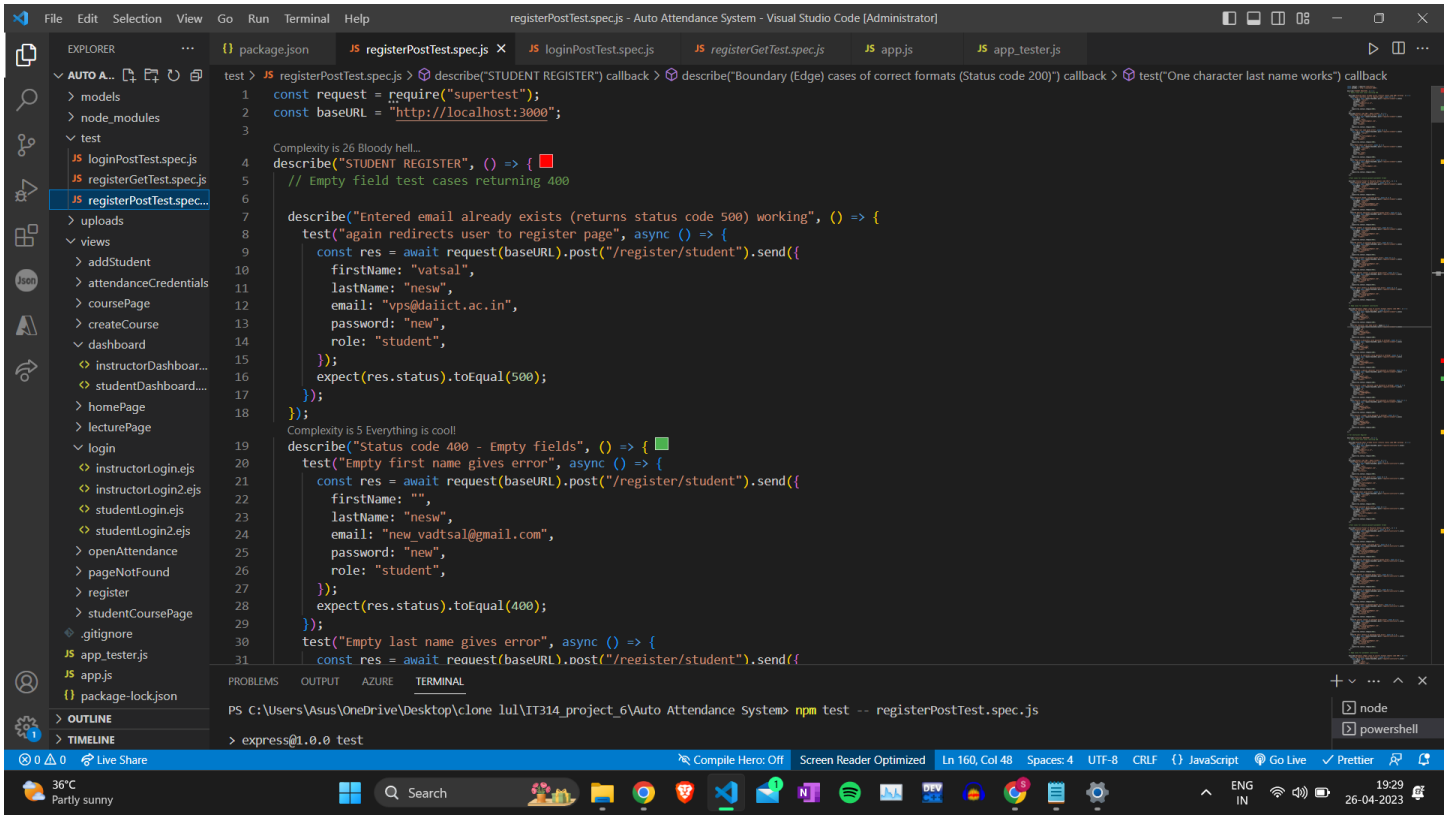
Register Unit Testing

The following testing module contains different scenarios for:

1. Instructor Register
2. Student Register

Code is saved as *registerPostTest.spec.js*

Code Snippet:

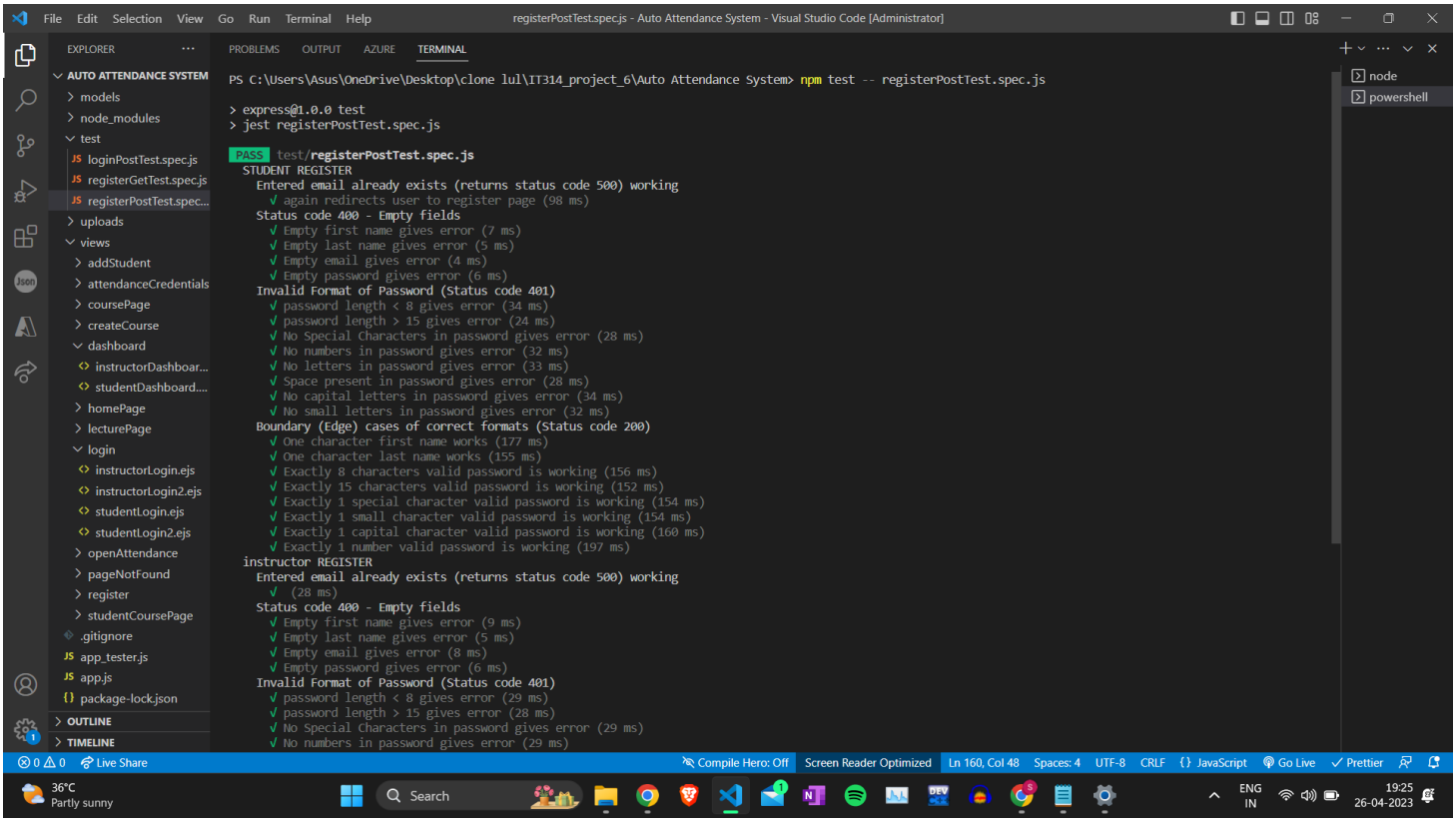


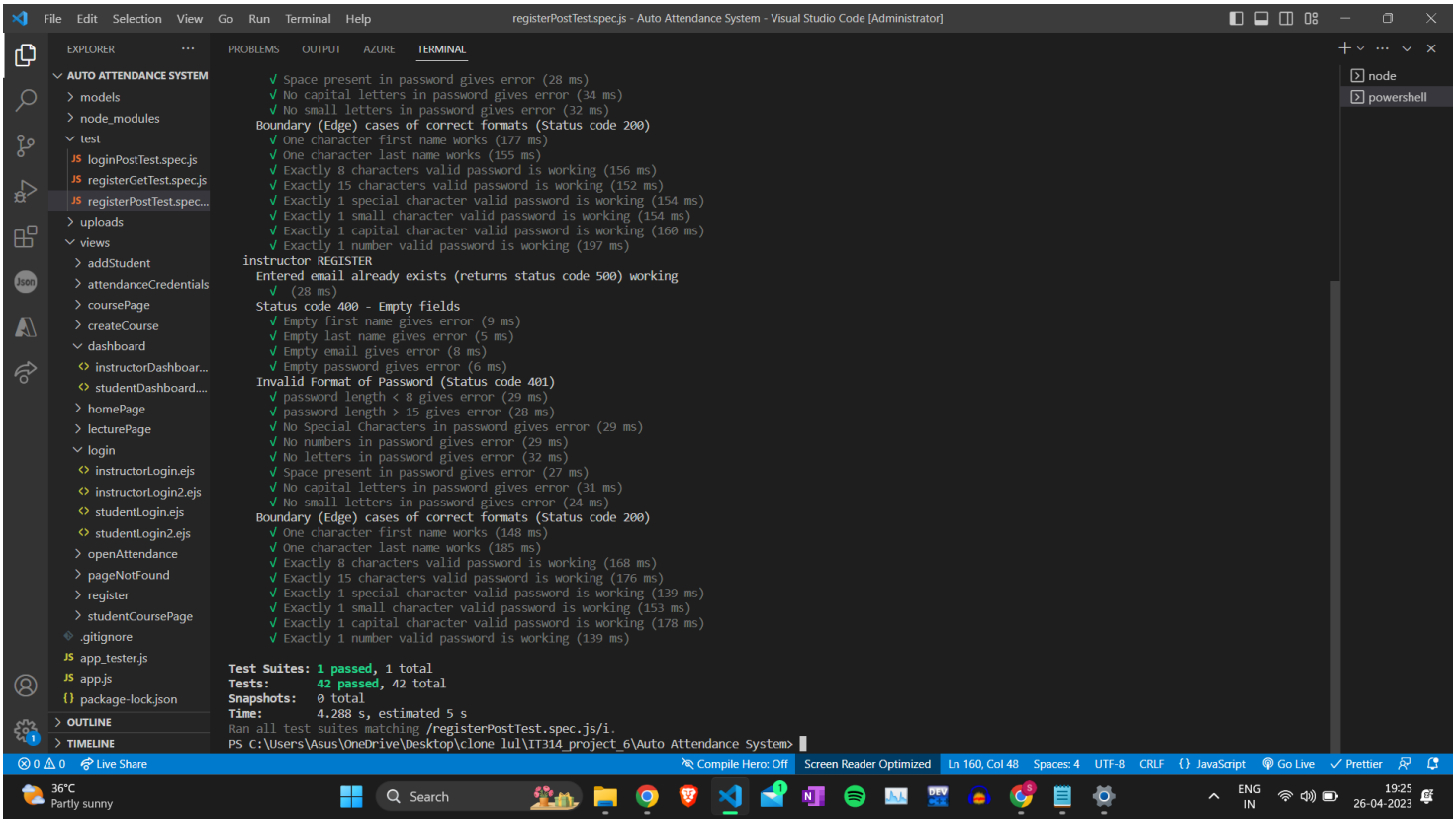
We took all kinds of scenarios possible and generated 42 test cases and hard coded them using *Jest and Supertest.*

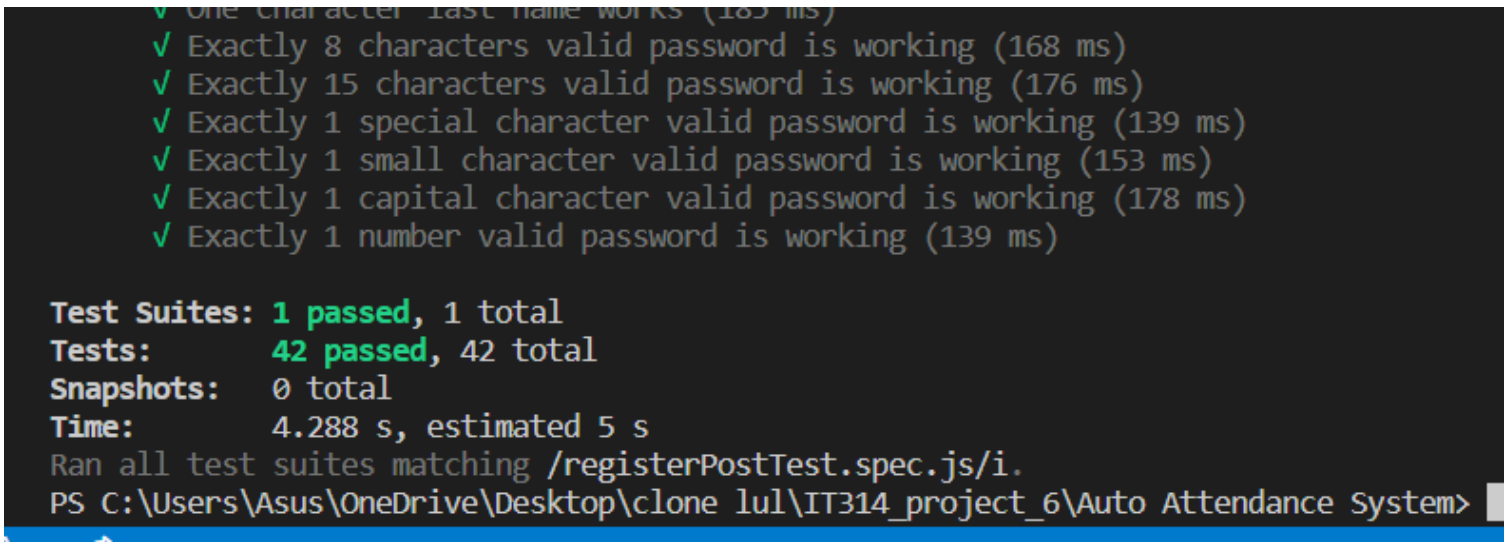
Also entered the expected output for each test cases/scenarios and if the output matches the expected output then the test case will be shown as ✔passed

The results are shown below in screenshots.

Result after running 42 test cases



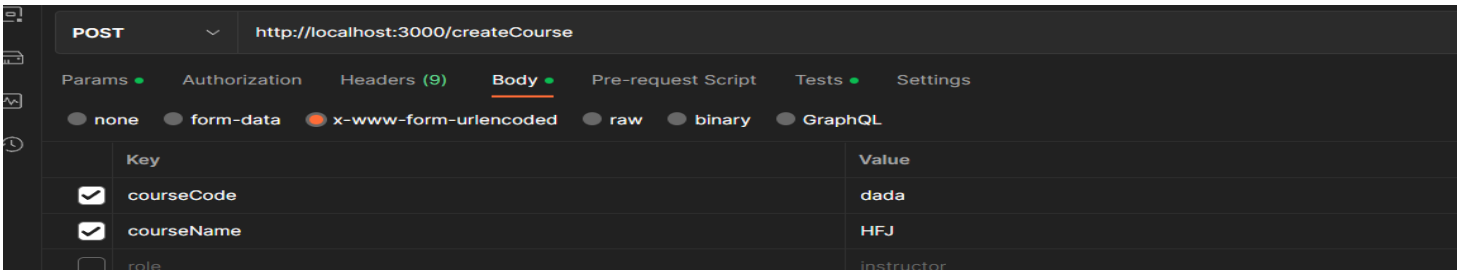


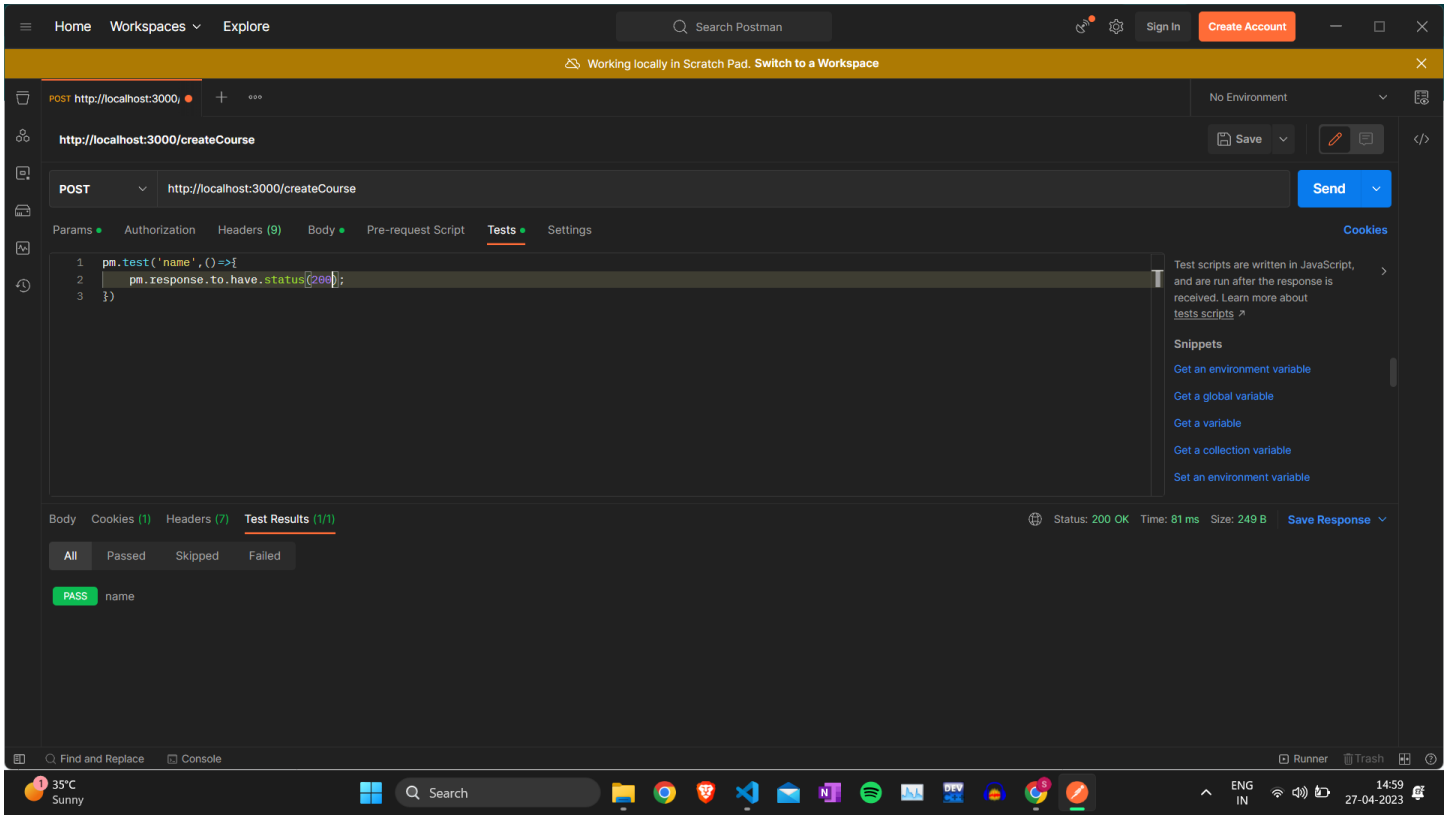


The next two modules are tested in *Postman* because the terminal will not store the data for login and it was a tedious code to implement the scenario in *Jest and Supertest*.

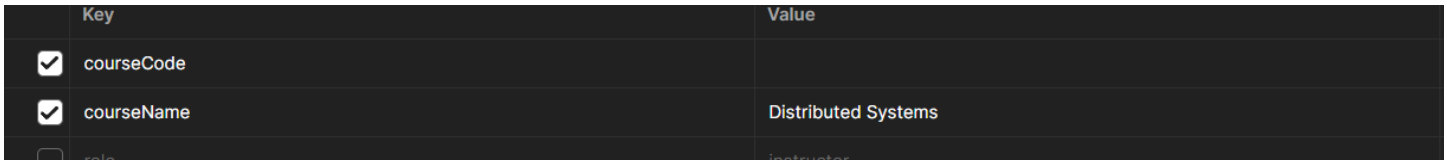
Testing for Create Course

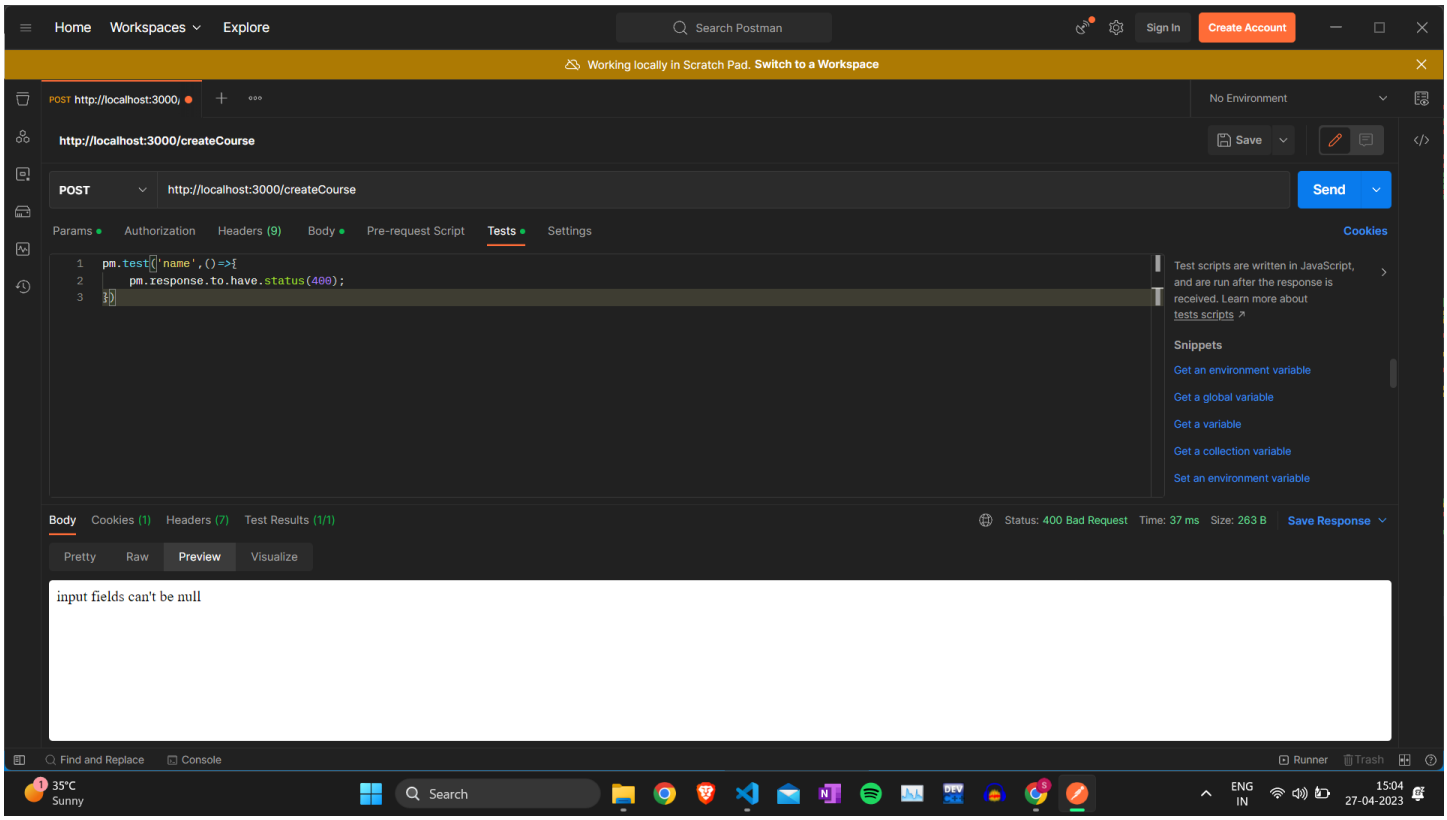
Test Case 1:
courseCode: valid
courseName: valid





Test case2:
courseCode: empty
courseName: valid



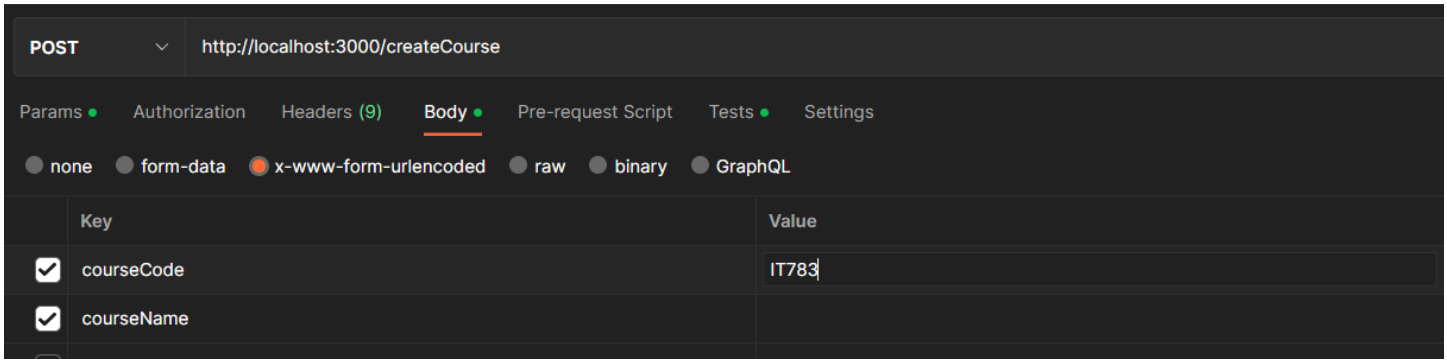


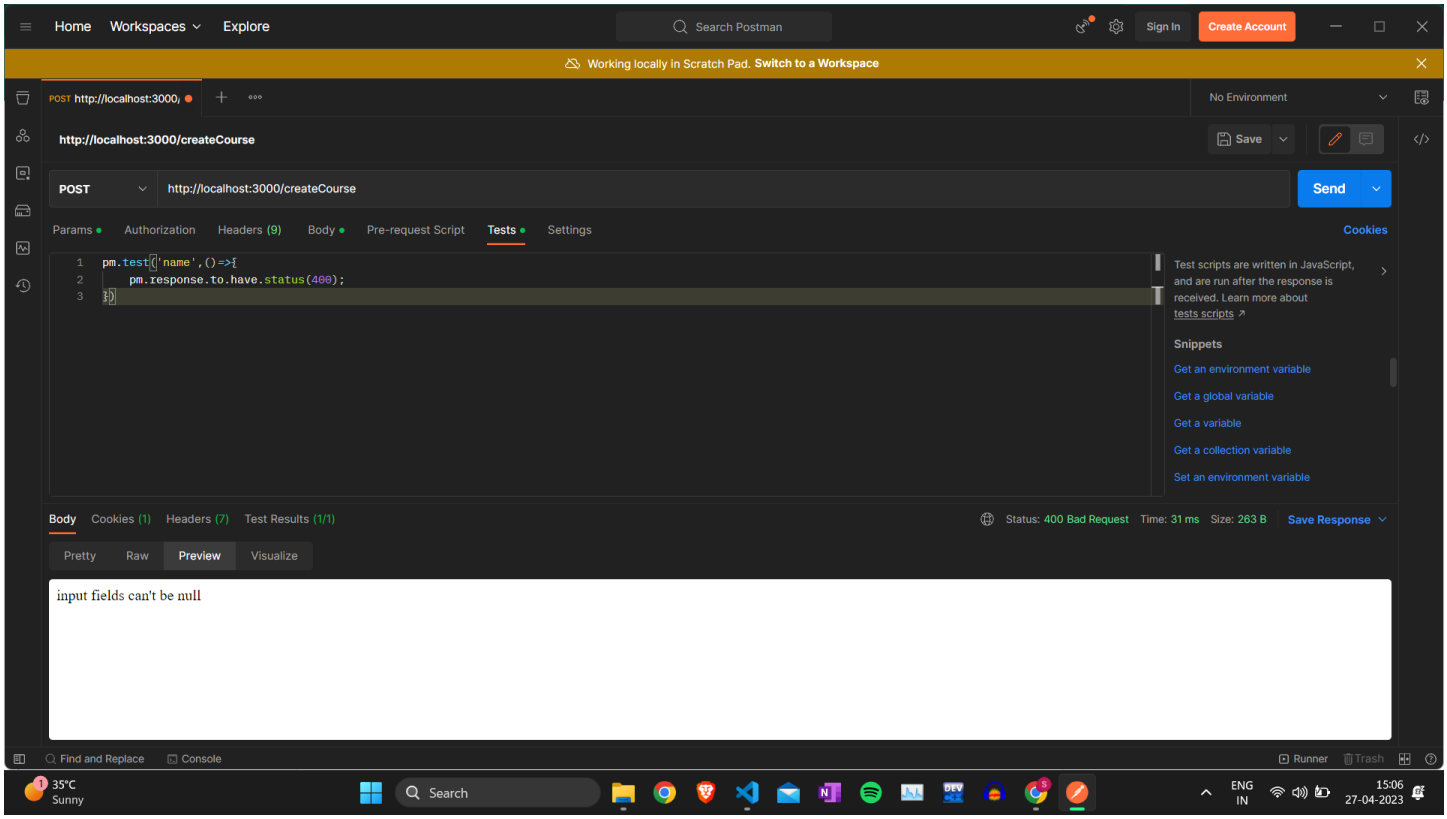
Test case 3:

When courseName is empty and courseCode is valid.

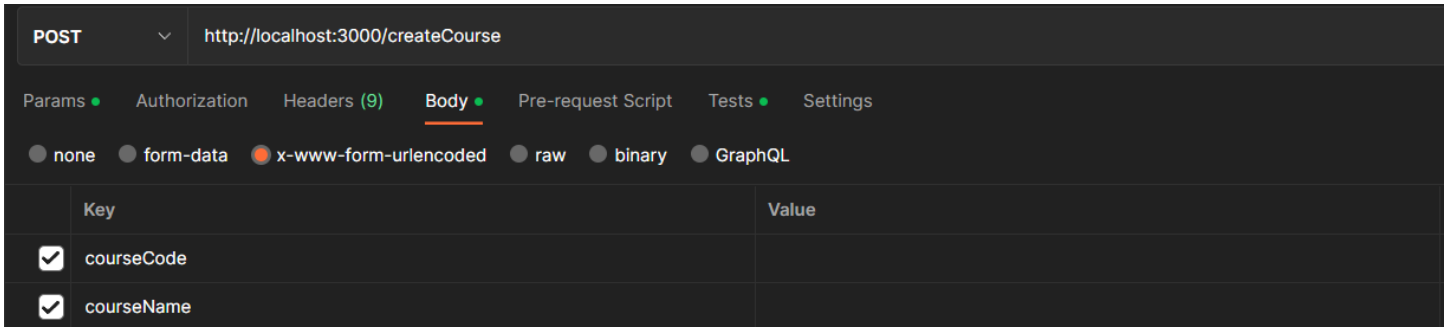
courseCode: valid

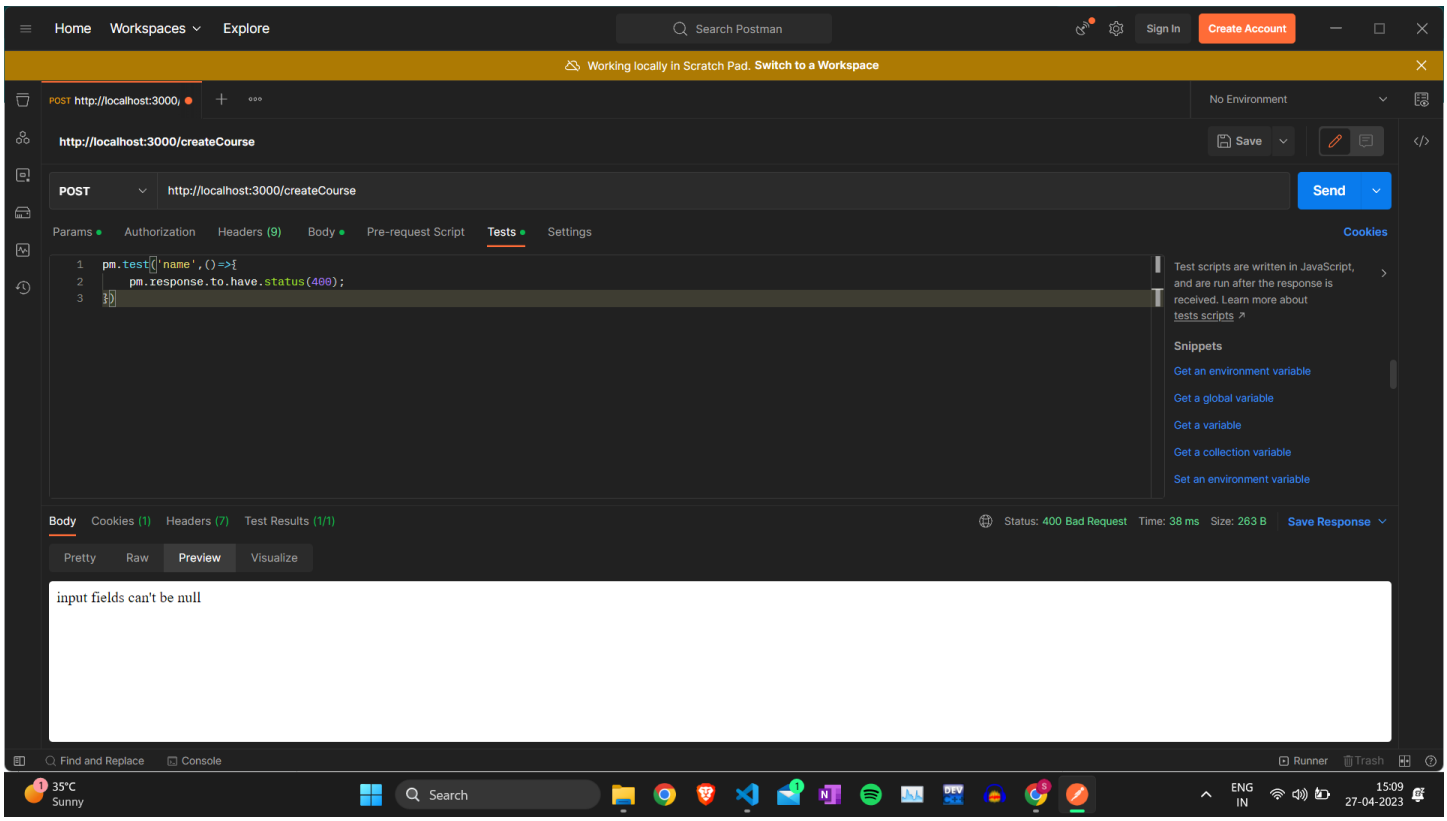
courseName: empty





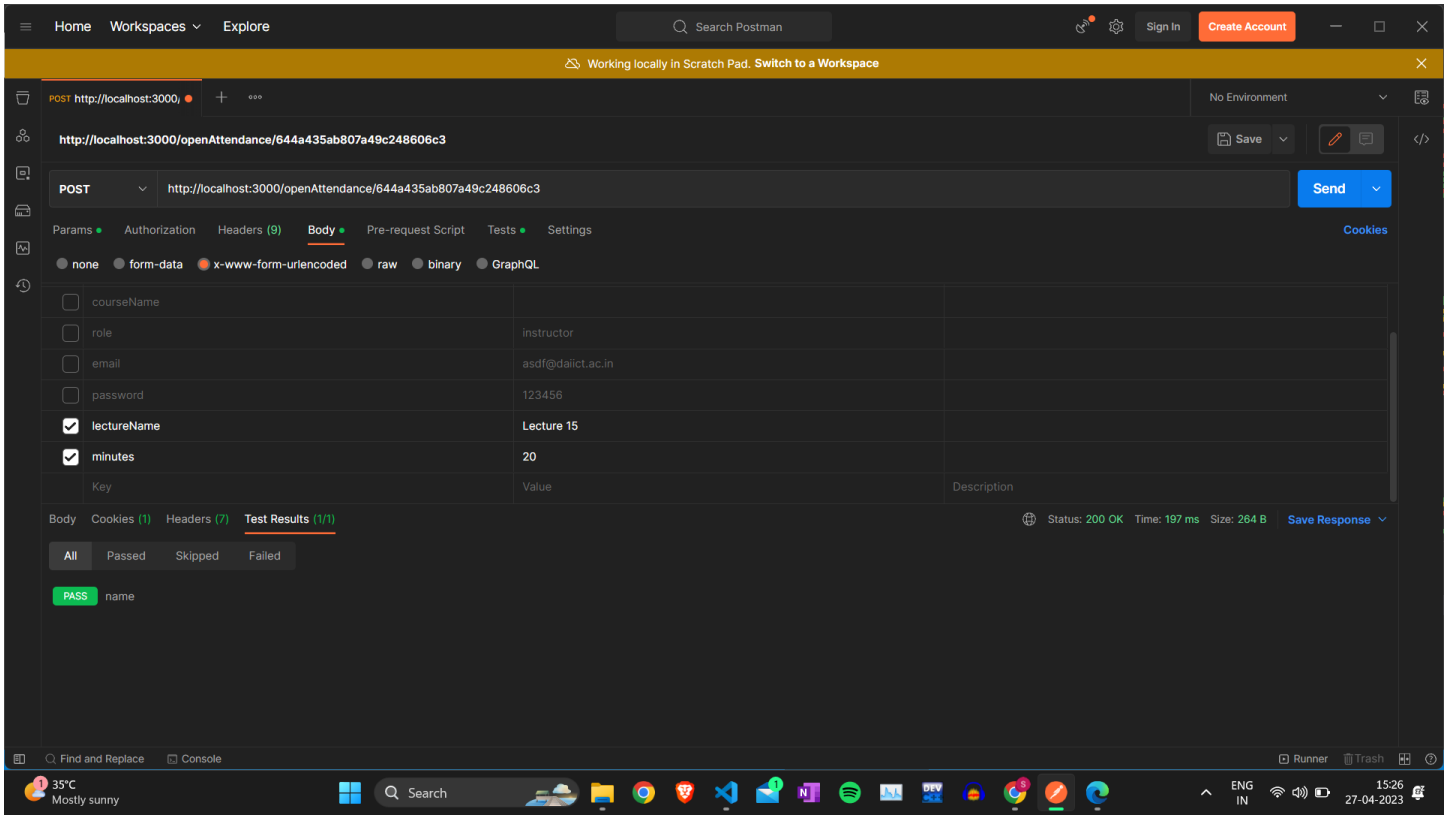
Test Case 4:
courseCode: empty
courseName: empty

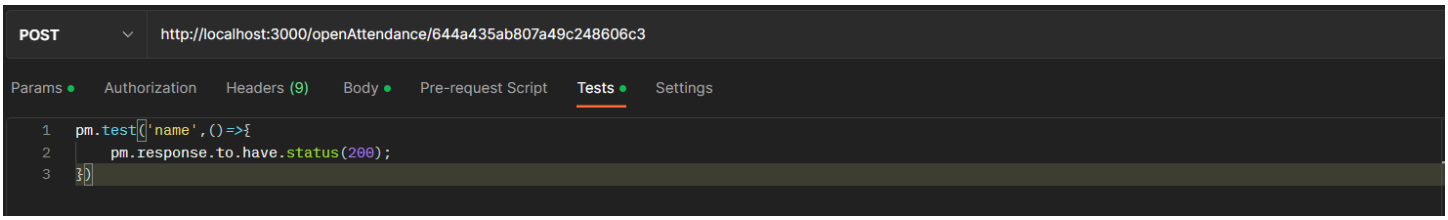




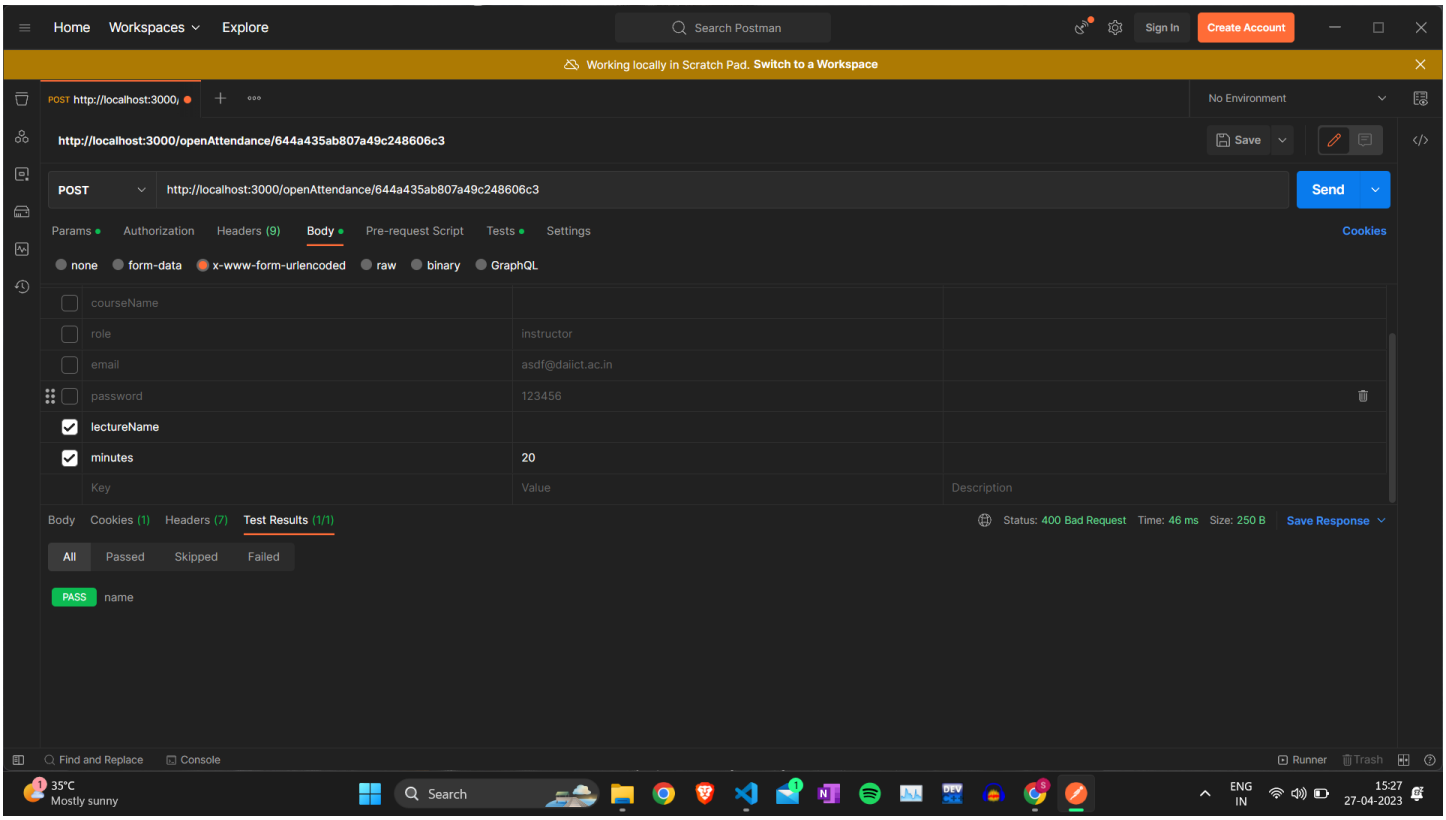
Testing for creating new Attendance

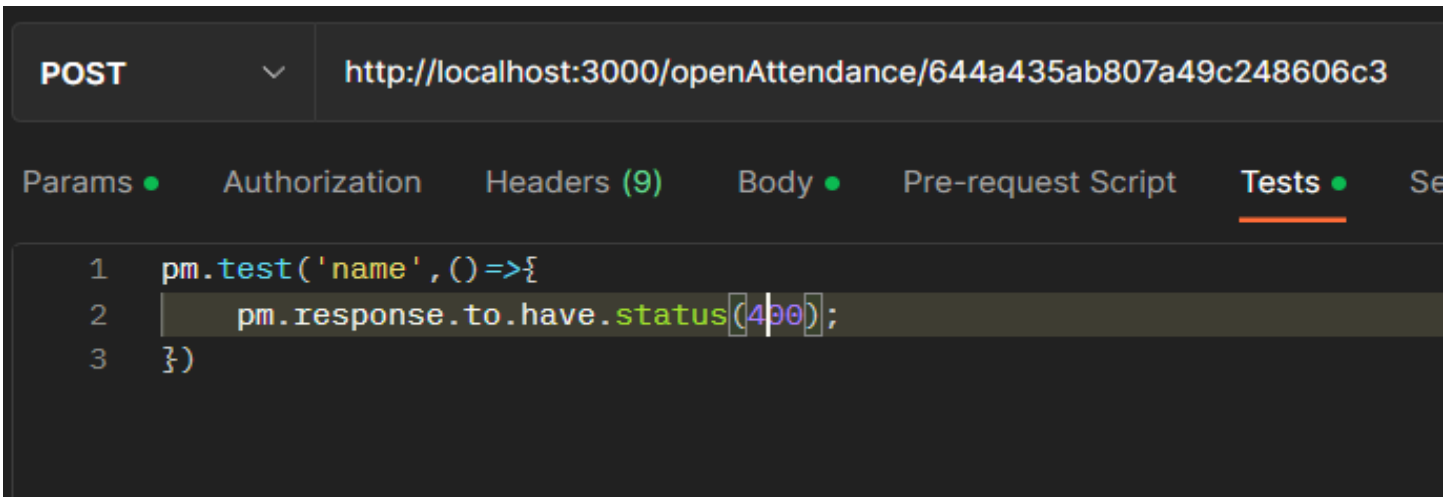
Test case 1:
lectureName: valid
Minutes: valid





Test case 2:
lectureName: empty
minutes: valid

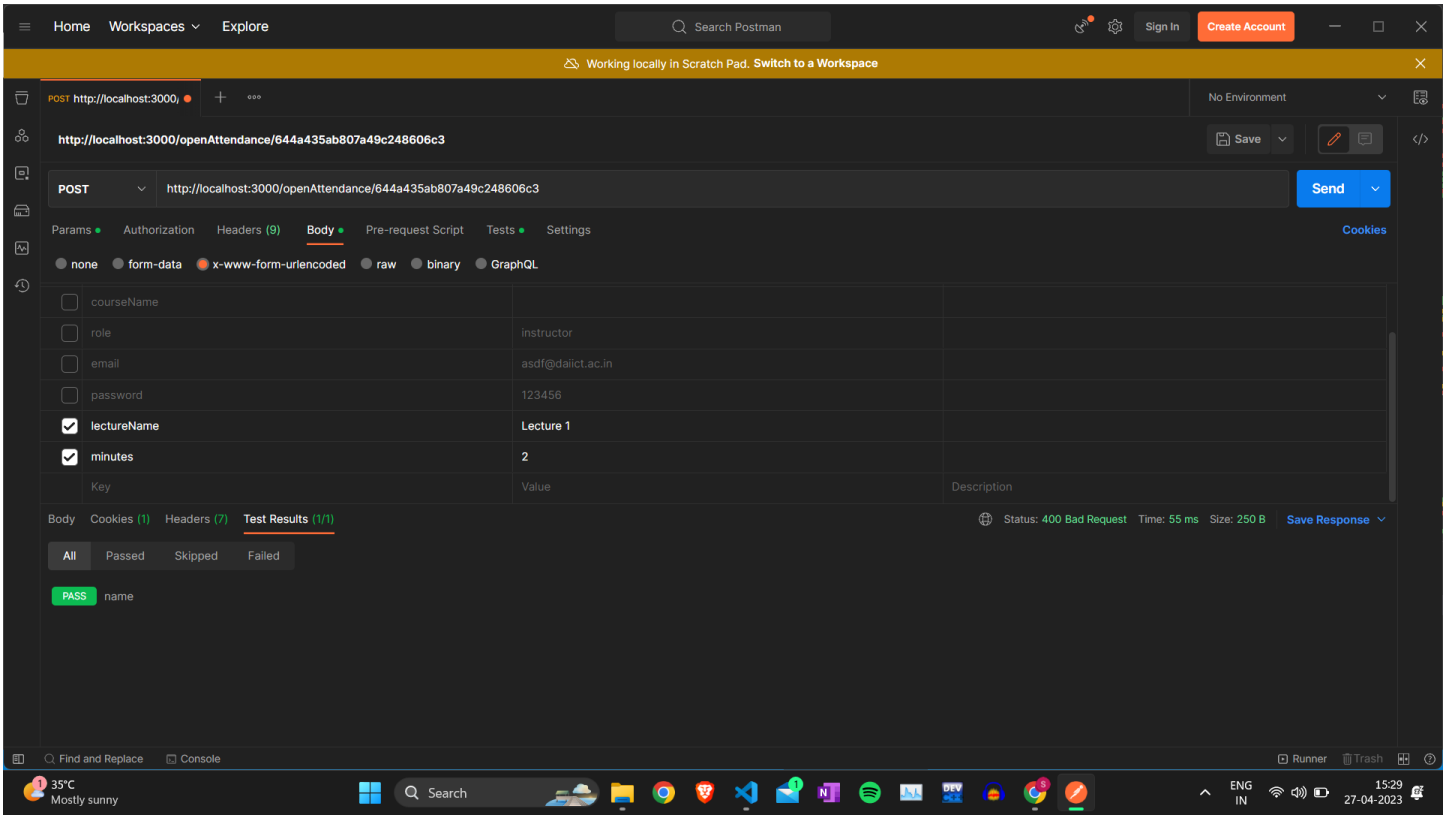


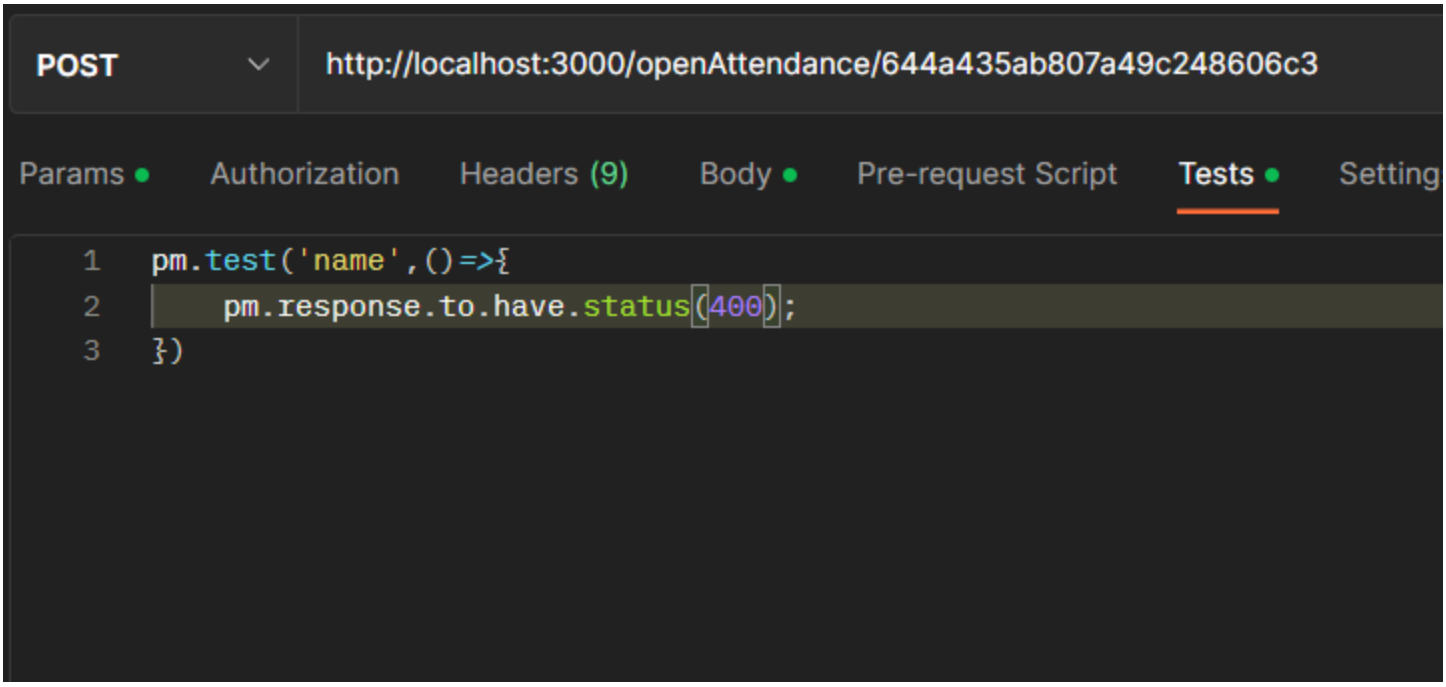


Test case 3:

lectureName: valid

minutes : not valid

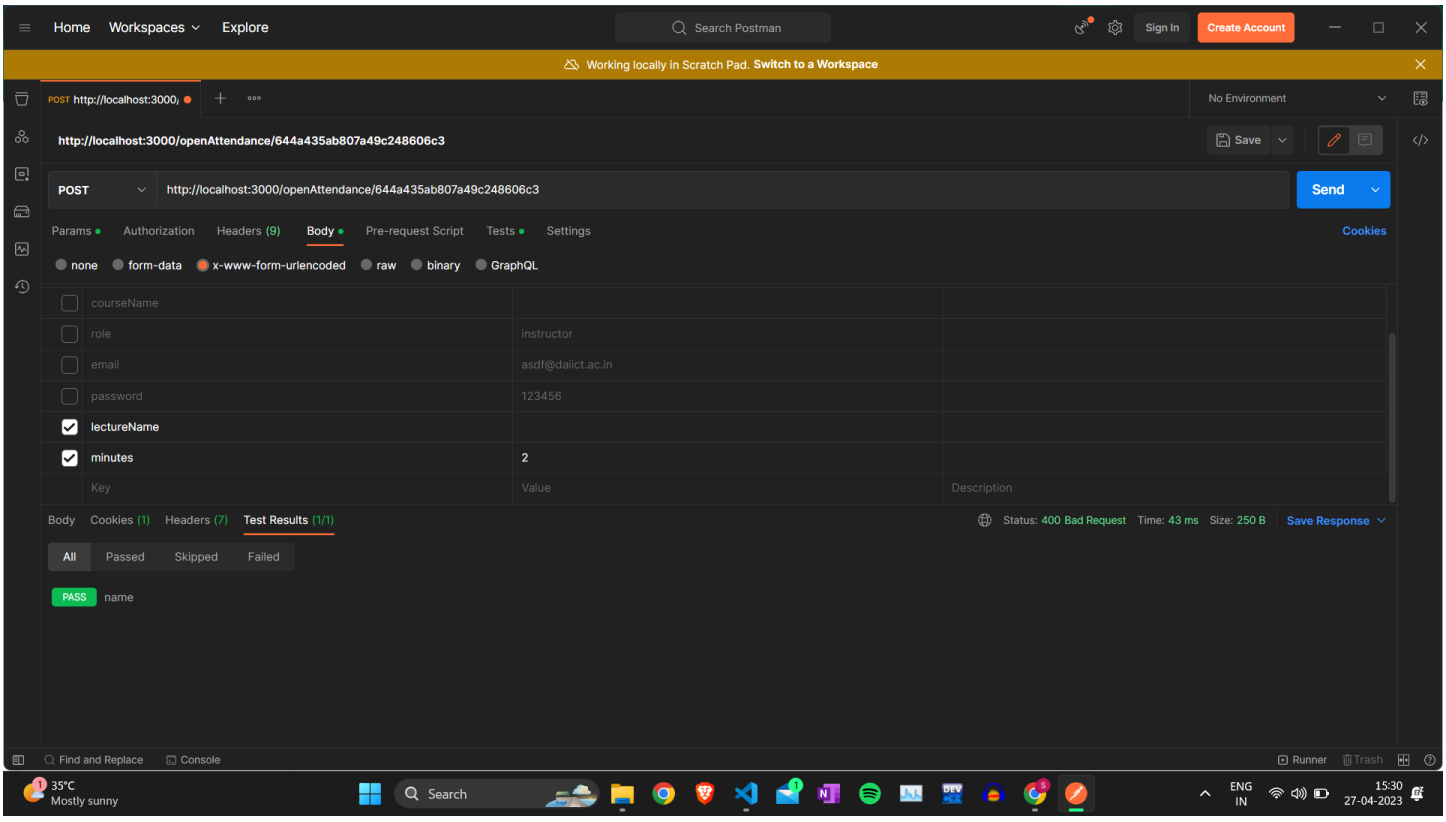


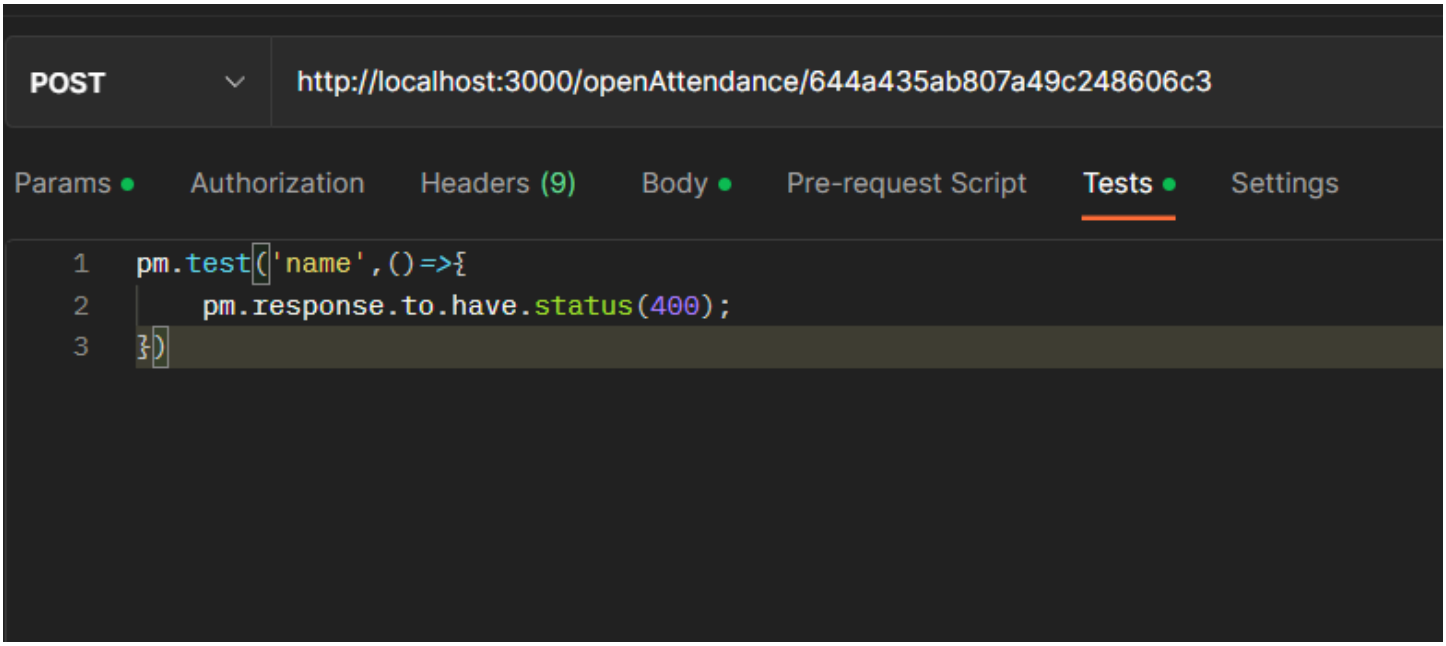


Test case 4:

lectureName: not valid

minutes: not valid

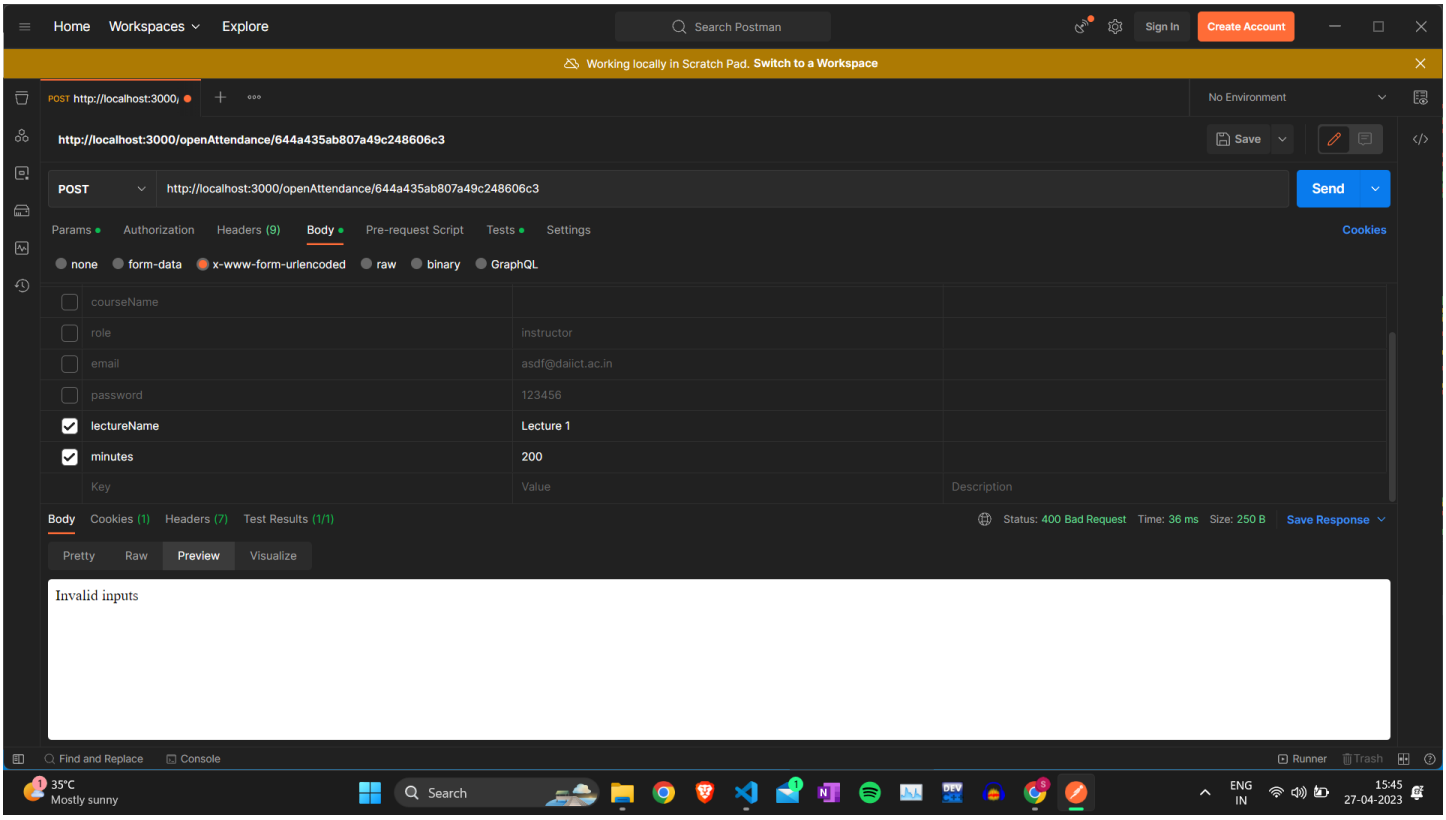




Test case 5:

lectureName: Valid

Minutes: >75



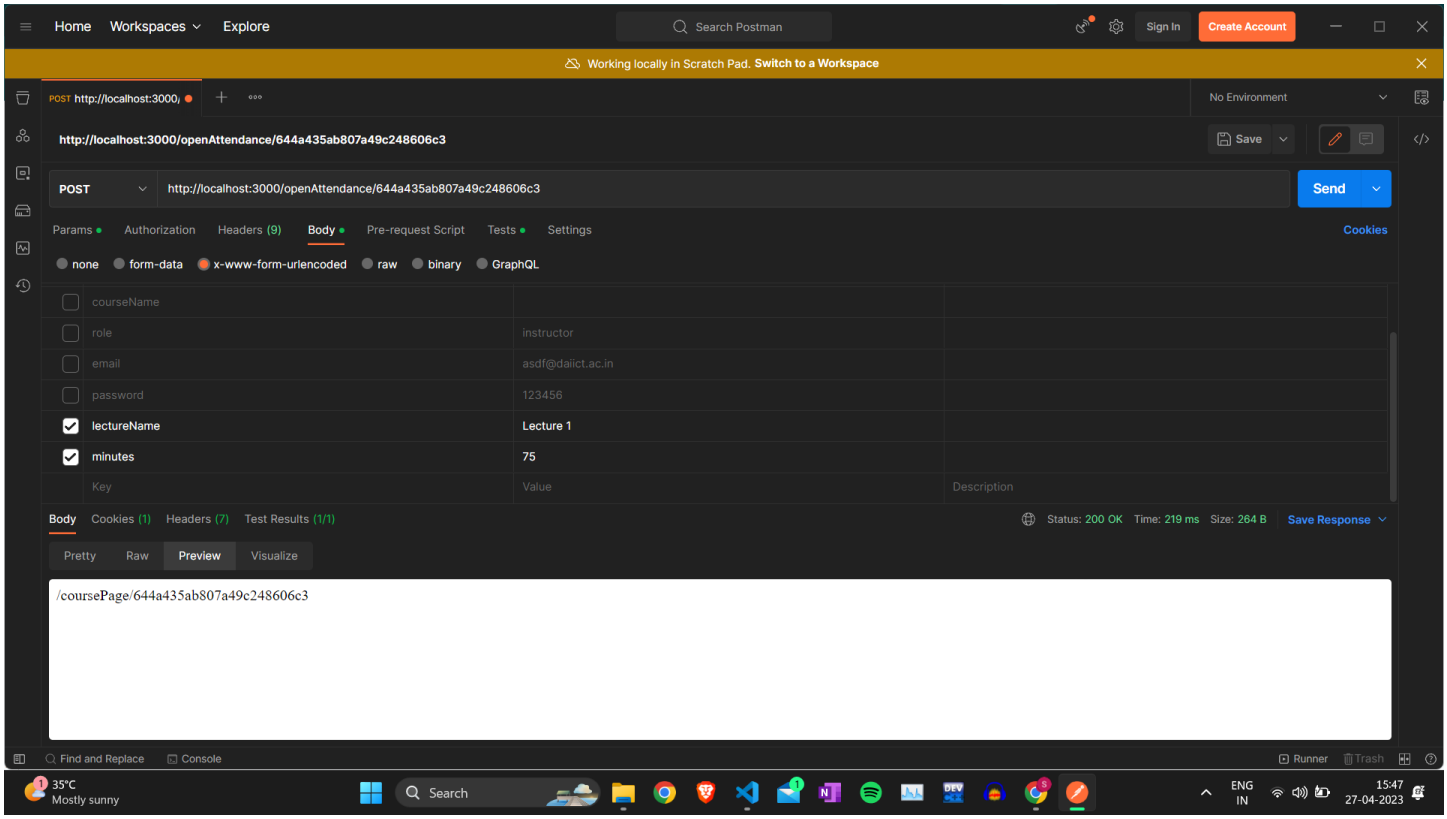
POST http://localhost:3000/openAttendance/644a435ab807a49c248606c3

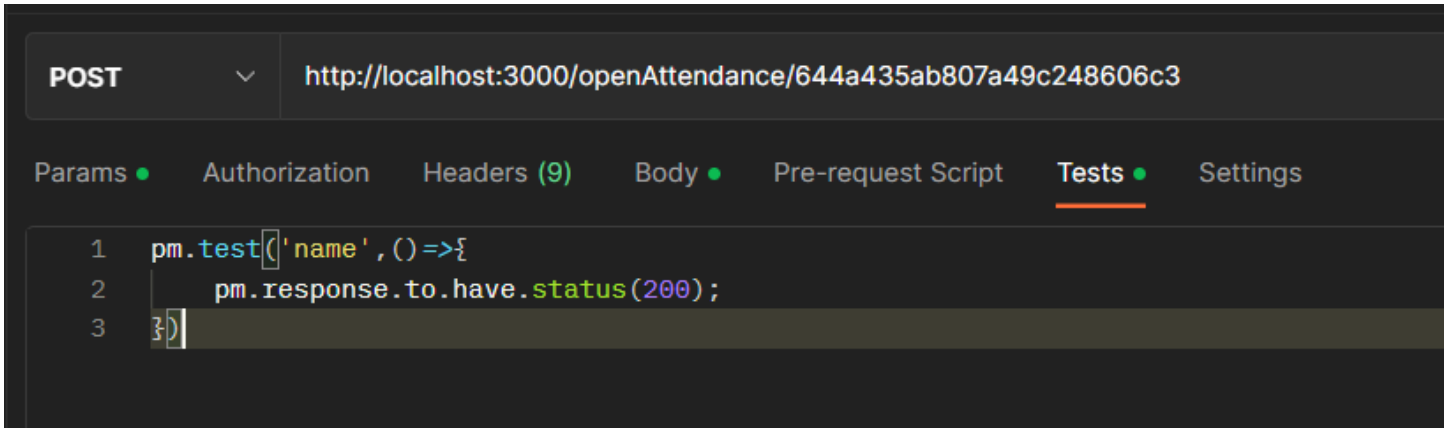
Params • Authorization Headers (9) Body • Pre-request Script Tests • Settings

```
1  pm.test('name',()=>{
2      pm.response.to.have.status(400);
3  })
```

Test case 6:

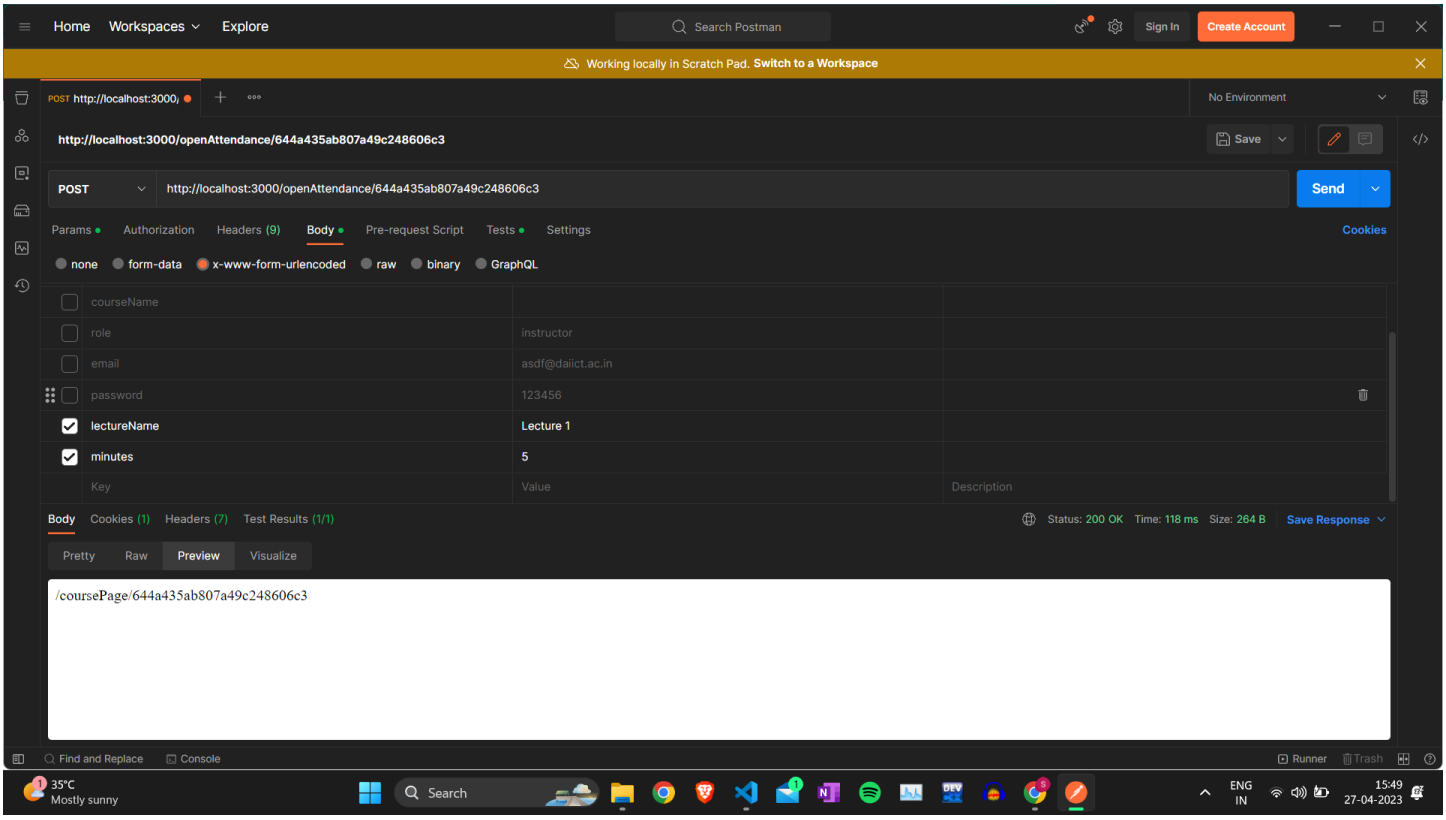
When lecture name is valid and minutes is exactly equal to 75.

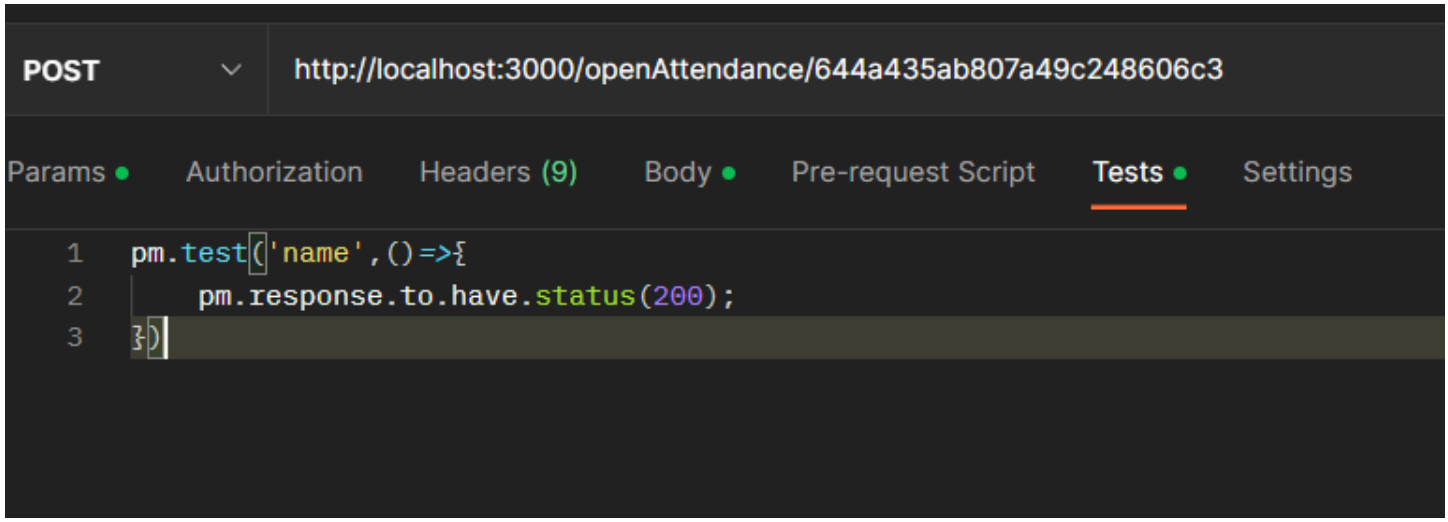




Test 7:

When input is exactly equal to 5 and lecture name is valid.




POST
http://localhost:3000/openAttendance/644a435ab807a49c248606c3
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
1 pm.test('name',()=>{
2 pm.response.to.have.status(200);
3 })