



**Software Engineering IT314**  
**Project: Student leave and TA assistantship**  
**management**  
**Group: 9**

● **Group Members:**

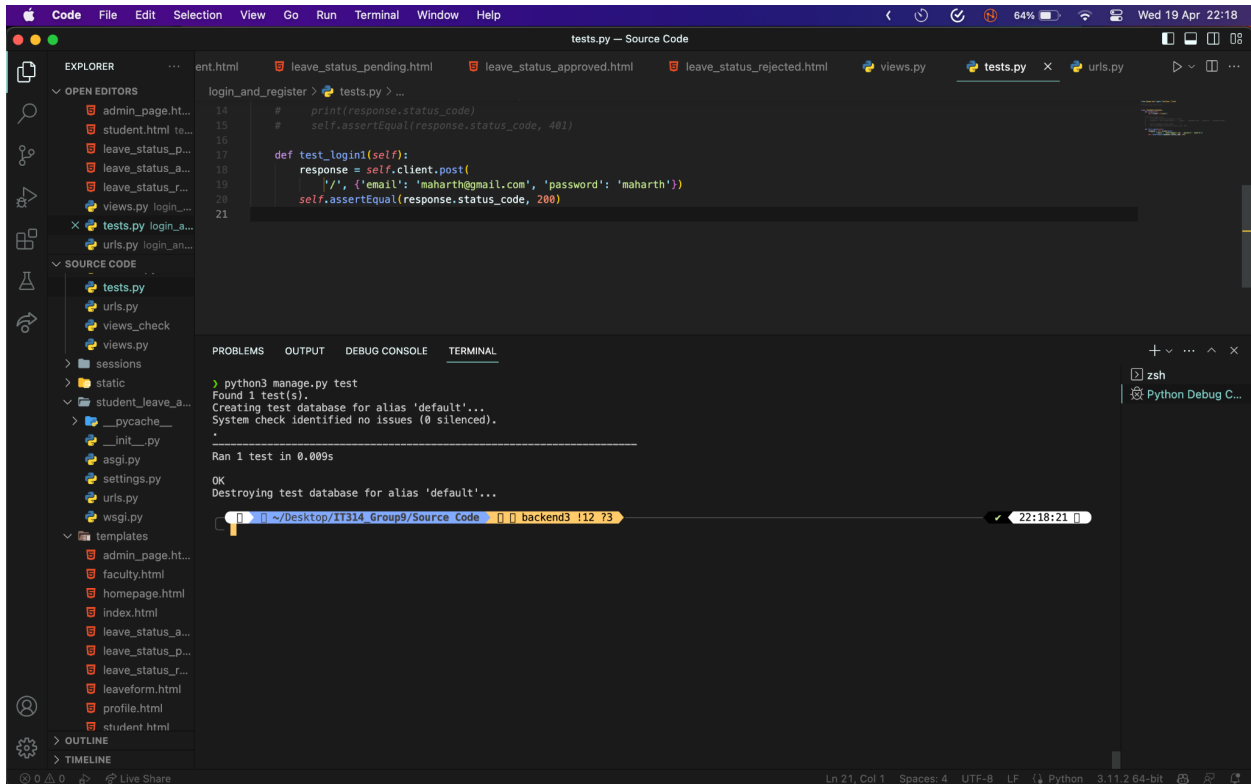
- Preksha Anand - 202001061
- Maharth Thakar - 202001069
- **Meet Patel - 202001074**
- Kartik Dangi - 202001079
- Harsh Kathiriya - 202001088
- Kuldipsinh Gohil - 202001092
- Neha Miglani - 202001097
- Harsh Anand - 202001101
- Darshan Kheni - 202001109

**Lab: 08**

**Date: 19-04-2023**

## ❖ Taking the module of Login form for testing the test case.

### ➤ Test Case 1



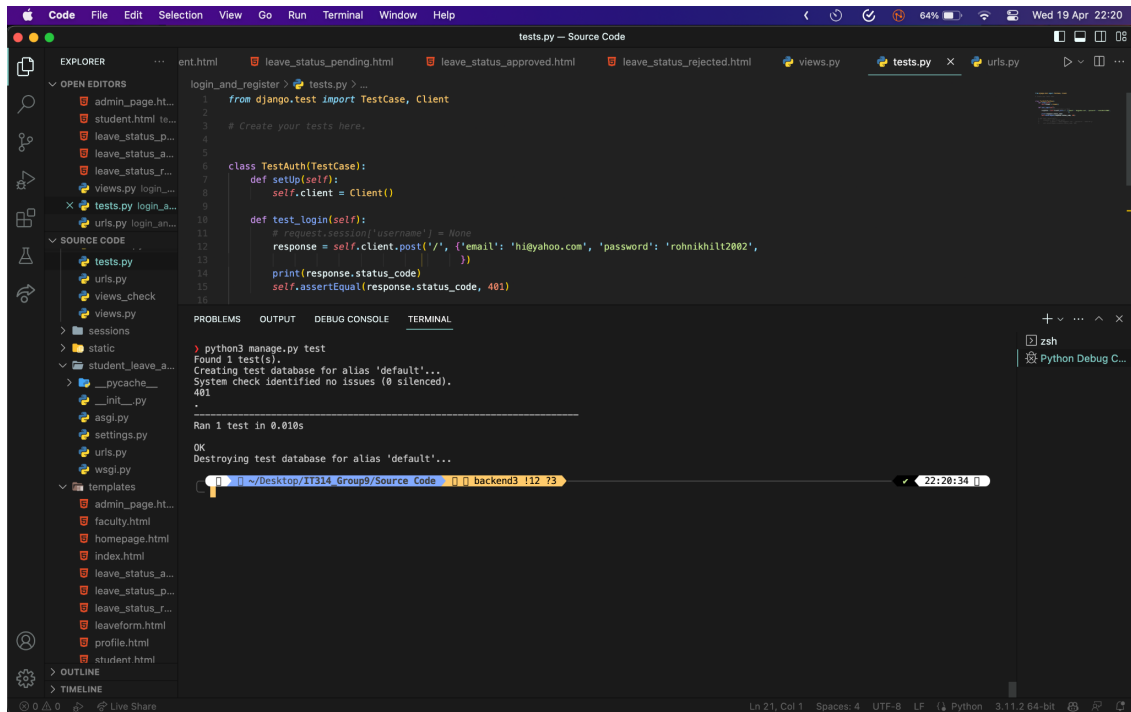
```
tests.py — Source Code
login_and_register > tests.py > ...
14 # print(response.status_code)
15 # self.assertEqual(response.status_code, 401)
16
17 def test_login1(self):
18     response = self.client.post(
19         '/', {'email': 'mahartha@gmail.com', 'password': 'maharth'})
20     self.assertEqual(response.status_code, 200)
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
> python3 manage.py test
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.
Ran 1 test in 0.009s
OK
Destroying test database for alias 'default'...
```

- The above test case passes with response code 200 (OK) as the username and password credentials match the database values.

## ➤ Test Case 2



The screenshot shows a code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure with files like `admin_page.html`, `student.html`, `leave_status_p...`, `leave_status_a...`, `leave_status_r...`, `views.py`, `urls.py`, `login_and_register`, `leave_status_pending.html`, `leave_status_approved.html`, `leave_status_rejected.html`, `views.py`, `tests.py`, `urls.py`, `views_check`, `views.py`, `sessions`, `static`, `student_leave_a...`, `__pycache__`, `__init__.py`, `asgi.py`, `settings.py`, `urls.py`, `wsgi.py`, `templates`, `admin_page.ht...`, `faculty.html`, `homepage.html`, `index.html`, `leave_status_a...`, `leave_status_p...`, `leave_status_r...`, `leaveform.html`, `profile.html`, and `student.html`. The `tests.py` file is open in the editor, showing the following code:

```
1 login_and_register > tests.py > ...
2 from django.test import TestCase, Client
3 # Create your tests here.
4
5 class TestAuth(TestCase):
6     def setUp(self):
7         self.client = Client()
8
9     def test_login(self):
10         # request.session['username'] = 'hoo
11         response = self.client.post('/', {'email': 'hi@yahoo.com', 'password': 'rohnikhilt2002',
12                                           })
13         print(response.status_code)
14         self.assertEqual(response.status_code, 401)
```

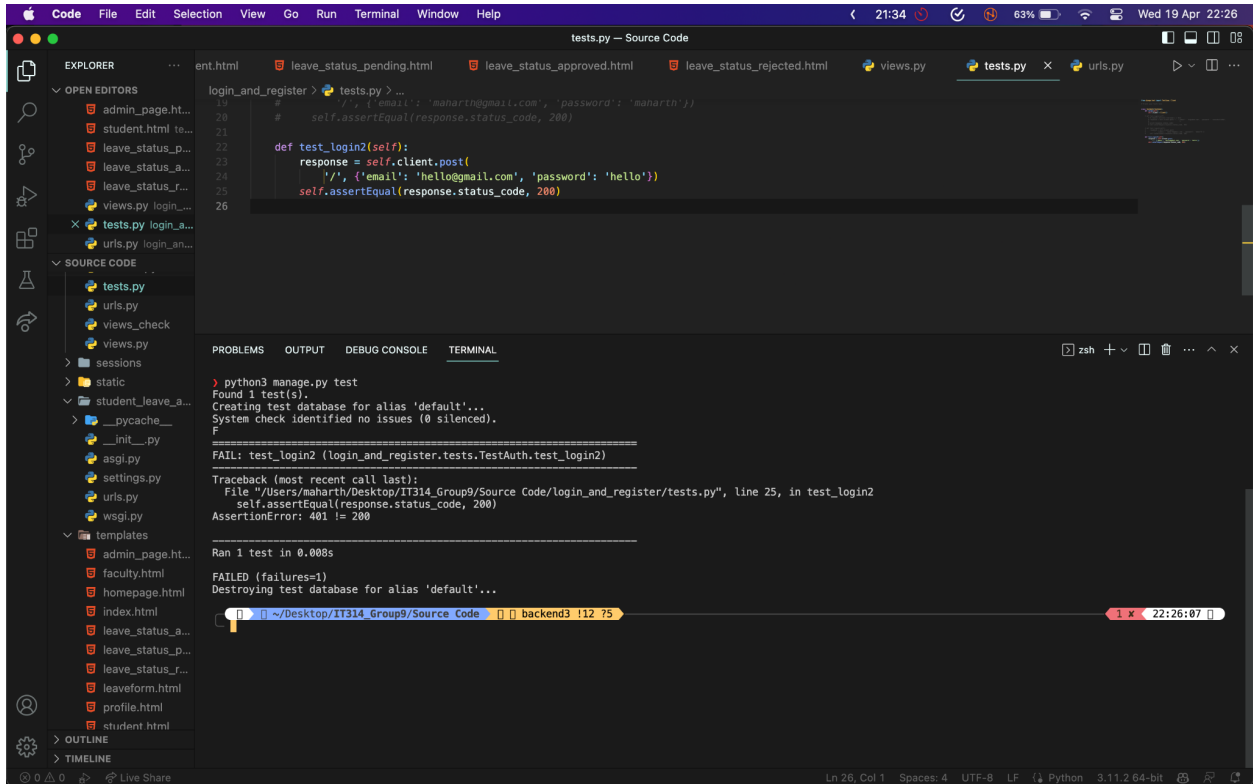
The terminal at the bottom shows the output of running the test:

```
> python3 manage.py test
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
401
Ran 1 test in 0.010s
OK
Destroying test database for alias 'default'...
```

The terminal also shows the file path `~/Desktop/IT314_Group9/Source Code` and the backend3 !12 73.

- The above test case passed with the response code 401 (Invalid Authentication) as the credentials are wrong in the above test case.
- But the result shows as OK as we predicted the wrong result.

## ➤ Test Case 3



```
tests.py -- Source Code
19 # /, ('email': 'manarth@gmail.com', 'password': 'manarth'))
20 # self.assertEqual(response.status_code, 200)
21
22 def test_login2(self):
23     response = self.client.post(
24         '/', {'email': 'hello@gmail.com', 'password': 'hello'})
25     self.assertEqual(response.status_code, 200)
26
```

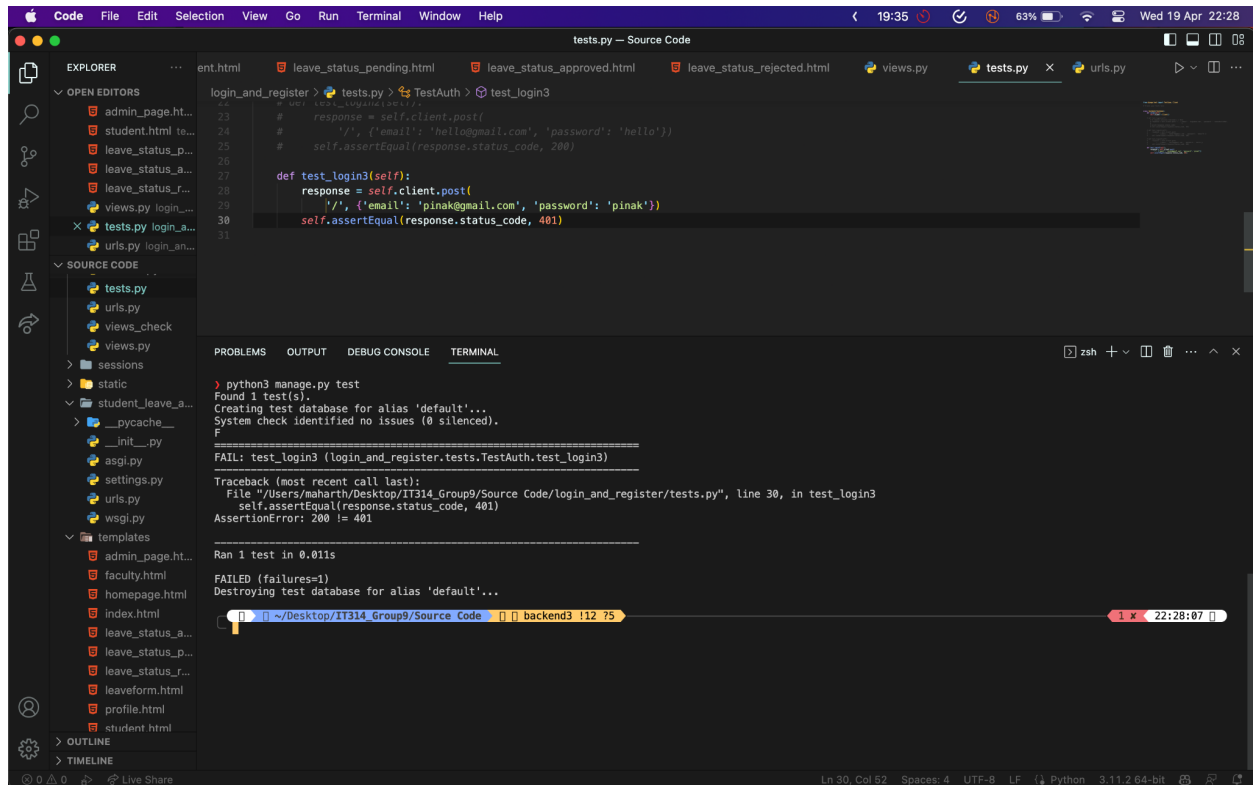
```
python3 manage.py test
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
F
=====
FAIL: test_login2 (login_and_register.tests.TestAuth.test_login2)
Traceback (most recent call last):
  File "/Users/maharth/Desktop/IT314_Group9/Source Code/login_and_register/tests.py", line 25, in test_login2
    self.assertEqual(response.status_code, 200)
AssertionError: 401 != 200

Ran 1 test in 0.008s

FAILED (failures=1)
Destroying test database for alias 'default'...
```

- The above code test case fails as we predicted the correct result with the response code of 200 (OK) but eh provided credentials are wrong, which ultimately gives us the failure.

## ➤ Test Case 4



```
tests.py — Source Code
login_and_register > tests.py > TestAuth > test_login3
23 # response = self.client.post(
24 #     '/', {'email': 'hellogmail.com', 'password': 'hello'})
25 # self.assertEqual(response.status_code, 200)
26
27 def test_login3(self):
28     response = self.client.post(
29         '/', {'email': 'pinak@gmail.com', 'password': 'pinak'})
30     self.assertEqual(response.status_code, 401)
31

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
> python3 manage.py test
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
F
=====
FAIL: test_login3 (login_and_register.tests.TestAuth.test_login3)
Traceback (most recent call last):
  File "/Users/maharth/Desktop/IT314 Group9/Source Code/login_and_register/tests.py", line 30, in test_login3
    self.assertEqual(response.status_code, 401)
AssertionError: 200 != 401

Ran 1 test in 0.011s
FAILED (failures=1)
Destroying test database for alias 'default'...
```

- The above test case failed as we predicted wrong with the response code 401, but we tested it with the correct credentials, which ultimately failed.

❖ The test.py file in the Django framework supports unit testing, and here is the code for the same:

```
from django.test import TestCase, Client

# Create your tests here.

class TestAuth(TestCase):
    def setUp(self):
        self.client = Client()

    def test_login(self):
        # request.session['username'] = None
        response = self.client.post('/', {'email': 'hi@yahoo.com', 'password':
'rohnikhilt2002',
                                         })

        print(response.status_code)
        self.assertEqual(response.status_code, 401)

    def test_login1(self):
        response = self.client.post(
            '/', {'email': 'maharth@gmail.com', 'password': 'maharth'})
        self.assertEqual(response.status_code, 200)

    def test_login2(self):
        response = self.client.post(
            '/', {'email': 'hello@gmail.com', 'password': 'hello'})
        self.assertEqual(response.status_code, 200)

    def test_login3(self):
        response = self.client.post(
            '/', {'email': 'pinak@gmail.com', 'password': 'pinak'})
        self.assertEqual(response.status_code, 401)
```

If a test is failed, I fix it by making the correct combinations of usernames and passwords.