



2023

SOFTWARE ENGINEERING
PROJECT

MILESTONE 5

TEST CASES



PRESENTED BY TEAM : SE-SEPT-04

**NIDHISH KUMAR (21F1003758), SHRI KRISHNA
PANDEY (21F1006966), SHREYOSI SARKAR
(21F1003968), NIWESH BARAJ (21F1001463),
PRATHAM BHALLA (21F1003052)**

API Endpoint being tested: <http://127.0.0.1:8000/api/register>

```
def test_user_registration_valid_data():
    input_dict = {
        "username": "string",
        "name": "string",
        "password": "string",
        "gender": 0,
        "primary_email": "string"
    }
    data = json.dumps(input_dict)
    header = {"Content-Type": "application/json"}
    response = client.post(user_registration_endpoint, data=data, headers=header)
    assert response.status_code == 201
    assert response.json() == "Successfully Registered!"
```

Inputs:

- Request Method: POST
- JSON: {"username": "string", "name": "string", "password": "string", "gender": 0, "primary_email": "string"}

Expected Output:

- HTTP Status Code: 201
- JSON: {"Successfully Registered!"}

Actual Output:

- HTTP Status Code: 201
- JSON: {"Successfully Registered!"}

Result: Success

API Endpoint being tested: <http://127.0.0.1:8000/api/login>

```
def test_user_login_valid_data():
    input_dict = {
        "email": "string",
        "password": "string"
    }
    data = json.dumps(input_dict)
    header = {"Content-Type": "application/json"}
    response = client.post(user_login_endpoint, data=data, headers=header)
    assert response.status_code == 200
    assert response.json()['id'] is not None
    assert response.json()['token'] is not None
```

Inputs:

- Request Method: POST
- JSON: {"email": "dummy@gmail.com", "password": "string"}

Expected Output:

- HTTP Status Code: 200
- JSON: {"id": 1, "token": "some string"}

Actual Output:

- HTTP Status Code: 200
- JSON: {"id": 1, "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbnRpbiCI6IjIxZjEwMDM3NThAZHMuYmV4cGlzMmE4ODUxNTUuMjE3NzNmNX0._HtOU-TXm1t1qGE2ICwnuAUOfodbKx00n5HPUHhCGBA"}"

Result: Success

API Endpoint being tested: <http://127.0.0.1:8000/api/forget-password>

```
def test_user_forget_password_get_valid_data():  
  
    endpoint = f"{user_forget_password_endpoint}?email={valid_email}"  
    header = {"Content-Type": "application/json"}  
    response = client.get(endpoint, headers=header)  
    assert response.status_code == 200  
    assert response.json()
```

Inputs:

- Request Method: GET
- JSON: {"email": "dummy@gmail.com"}

Expected Output:

- HTTP Status Code: 200
- JSON: {true}

Actual Output:

- HTTP Status Code: 200
- JSON: {true}

Result: Success

API Endpoint being tested: <http://127.0.0.1:8000/api/forget-password>

```
def test_user_forget_password_put_valid_data():

    input_dict = {
        "email": valid_email,
        "new_password": "string"
    }

    data = json.dumps(input_dict)
    header = {"Content-Type": "application/json"}
    response = client.put(user_forget_password_endpoint, data=data, headers=header)
    assert response.status_code == 202
    assert response.json() == "Successfully Changed!"
```

Inputs:

- Request Method: PUT
- JSON: {"email": "dummy@gmail.com", "new_password": "random_password"}

Expected Output:

- HTTP Status Code: 202
- JSON: {"Successfully Changed!"}

Actual Output:

- HTTP Status Code: 202
- JSON: {"Successfully Changed!"}

Result: Success

Pytest Outputs for the above endpoints

```
$ python -m pytest -v tests/test_user_api.py
===== test session starts =====
platform win32 -- Python 3.10.11, pytest-7.2.1, pluggy-1.0.0 -- C:\Users\HP\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\python.exe
cachedir: .pytest_cache
rootdir: D:\Github\Academic\Recommendation-System
plugins: anyio-3.7.0, shutil-1.7.0, typeguard-2.13.3
collected 4 items

tests/test_user_api.py::test_user_registration_valid_data PASSED [ 25%]
tests/test_user_api.py::test_user_login_valid_data PASSED [ 50%]
tests/test_user_api.py::test_user_forget_password_get_valid_data PASSED [ 75%]
tests/test_user_api.py::test_user_forget_password_put_valid_data PASSED [100%]

===== warnings summary =====
tests/test_user_api.py::test_user_registration_valid_data
tests/test_user_api.py::test_user_login_valid_data
tests/test_user_api.py::test_user_forget_password_put_valid_data
  C:\Users\HP\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages\Python310\site-packages\httpx\_content.py:204: DeprecationWarning: Use 'content' to upload raw bytes/text content.
    warnings.warn(message, DeprecationWarning)

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 4 passed, 3 warnings in 1.17s =====
```

API Endpoint being tested: http://127.0.0.1:8000/api/profile/{user_id}

Request Method: GET

```
def test_get_valid_user_data():  
  
    header = {"token": get_token(), "Content-Type": "application/json"}  
    response = client.get(f"{user_profile_endpoint}/{valid_user_id}", headers=header)  
    assert response.status_code == 200  
    assert response.json()['username'] is not None  
    assert response.json()['name'] is not None  
    assert response.json()['primary_email'] is not None  
    assert response.json()['profile_pic'] is not None
```

API Endpoint being tested: <http://127.0.0.1:8000/api/logout>

Request Method: POST

```
def test_valid_user_logout():  
  
    header = {"token": get_token(), "Content-Type": "application/json"}  
    response = client.post(user_logout_endpoint, headers=header)  
    assert response.status_code == 200  
    assert response.json() == "Successfully logged out!"
```

API Endpoint being tested: <http://127.0.0.1:8000/api/courses>

Request Method: GET

```
def test_get_all_courses():  
  
    header = {"Content-Type": "application/json"}  
    response = client.get(courses_endpoint, headers=header)  
    assert response.status_code == 200  
    assert response.json() is not []
```

API Endpoint being tested:

http://127.0.0.1:8000/api/courses/{course_id}

Request Method: GET

```
def test_get_valid_course():  
  
    header = {"Content-Type": "application/json"}  
    response = client.get(f"{courses_endpoint}/{valid_course_id}", headers=header)  
    assert response.status_code == 200  
    assert response.json()['name'] != ""  
    assert response.json()['course_id'] != ""
```

API Endpoint being tested:

http://127.0.0.1:8000/api/courses/{course_id}

Request Method: PUT

```
def test_update_valid_course():  
  
    input_data = {  
        "course_id": "string",  
        "name": "string",  
        "description": "string",  
        "duration": 0,  
        "course_credit": 0,  
        "is_project": True,  
        "status": True,  
        "level": 0  
    }  
  
    header = {"token": get_token(), "Content-Type": "application/json"}  
    response = client.put(f"{courses_endpoint}/{valid_course_id}", headers=header)  
    assert response.status_code == 202  
    assert response.json() == "Course details updated successfully!"
```

API Endpoint being tested: <http://127.0.0.1:8000/api/admin/students>

Request Method: GET

```
def test_get_all_students_details():  
  
    header = {"token": get_token(), "Content-Type": "application/json"}  
    response = client.get(all_students_endpoint, headers=header)  
    assert response.status_code == 200  
    assert response.json() != []
```

API Endpoint being tested:

<http://127.0.0.1:8000/api/admin/students/{roll}>

Request Method: GET

```
def test_get_student_details():  
  
    header = {"token": get_token(), "Content-Type": "application/json"}  
    response = client.get(specific_student_endpoint, headers=header)  
    assert response.status_code == 200  
    assert response.json()['name'] != ""  
    assert response.json()['roll'] != ""  
    assert response.json()['level'] != ""  
    assert response.json()['status'] is not None
```

API Endpoint being tested:

<http://127.0.0.1:8000/api/admin/students-graph-data>

Request Method: GET

```
def test_get_students_graph_data():  
  
    header = {"token": get_token(), "Content-Type": "application/json"}  
    response = client.get(students_graph_endpoint, headers=header)  
    assert response.status_code == 200
```

API Endpoint being tested:

<http://127.0.0.1:8000/api/admin/course-graph-data>

Request Method: GET

```
def test_get_courses_graph_data():  
  
    header = {"token": get_token(), "Content-Type": "application/json"}  
    response = client.get(courses_graph_endpoint, headers=header)  
    assert response.status_code == 200
```


API Endpoint being tested:

<http://127.0.0.1:8000/api/advisor/recommendation-paths>

Request Method: GET

```
def test_get_all_recommendation_paths():  
  
    header = {"token": get_token(), "Content-Type": "application/json"}  
    response = client.get(all_recommendation_paths_endpoint, headers=header)  
    assert response.status_code == 200
```

API Endpoint being tested:

http://127.0.0.1:8000/api/advisor/recommendation-path/{path_id}

Request Method: POST

```
def test_approve_recommendation_path():  
  
    header = {"token": get_token(), "Content-Type": "application/json"}  
    response = client.post(f"{recommendation_path_endpoint}/{valid_path_id}", headers=header)  
    assert response.status_code == 204  
    assert response.json() in ["Recommended path approved successfully!",  
                               "Recommended path disapproved successfully!"]
```

API Endpoint being tested:

<http://127.0.0.1:8000/api/recommendation-path>

Request Method: POST

```
def test_get_recommendation_paths():  
  
    input_dict = {  
        "n_hours": "string"  
    }  
  
    data = json.dumps(input_dict)  
    header = {"token": get_token(), "Content-Type": "application/json", "data": data}  
    response = client.post(recommendation_path_endpoint, headers=header)  
    assert response.status_code == 200
```

API Endpoint being tested:

<http://127.0.0.1:8000/api/recommendation-paths/status>

Request Method: GET

```
def test_get_recommendation_paths_status():  
  
    header = {"token": get_token(), "Content-Type": "application/json"}  
    response = client.get(recommendation_path_status_endpoint, headers=header)  
    assert response.status_code == 200
```

API Endpoint being tested:

http://127.0.0.1:8000/api/feedback/{course_id}

Request Method: POST

```
def test_valid_course_feedback():  
  
    input_dict = {  
        "feedback": "string"  
    }  
  
    data = json.dumps(input_dict)  
    header = {"token": get_token(), "Content-Type": "application/json", "data": data}  
    response = client.post(f"{course_feedback_endpoint}/{valid_course_id}", headers=header)  
    assert response.status_code == 200  
    assert response.json() == "Feedback provided successfully!"
```