SOFTWARE ENGINEERING PROJECT Term : Sept 2023 - Dec 2023

Milestone - 5

Contributors:

JS Ranjith (21f1002171)
Mohd. Ikram (21f2000712)
Dhruv Gupta (21f1004011)
Bimlesh KR Kanth (21f1005524)
Rahul Chakraborty (21f1002786)

Test Case Documentation:

Objective: To login with correct credentials

Api being tested: http://127.0.0.1:5001/api/login

Input:

- Request Method : POST
- JSON: {"username": "mdikram888", "password": "12345678"}

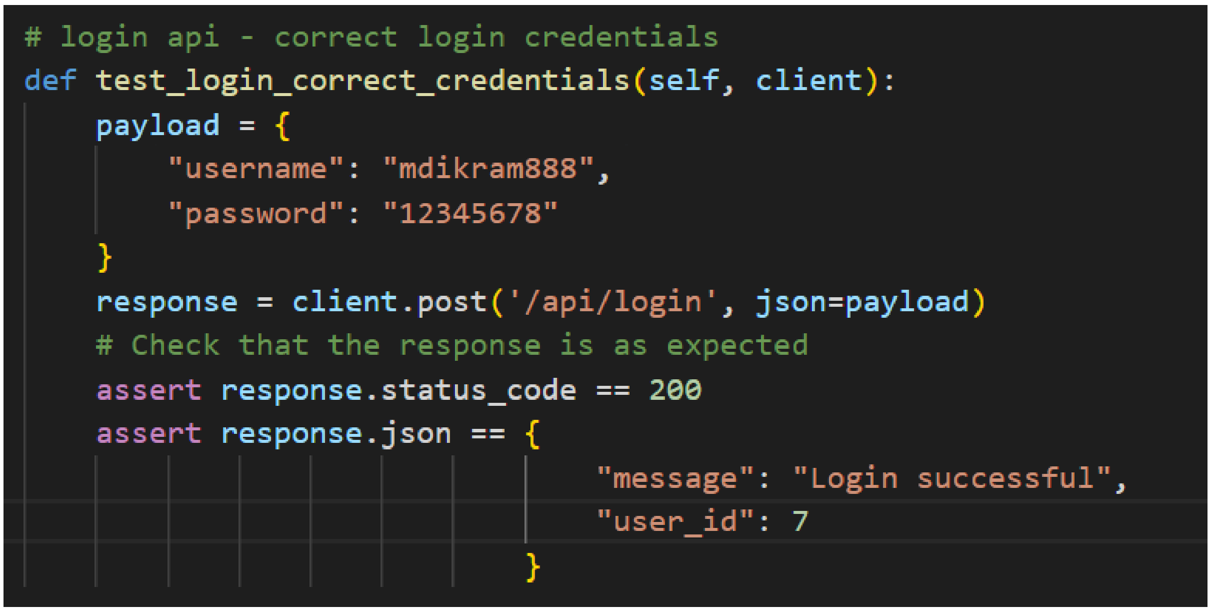
Expected Output:

- HTTP Status Code : 200
- JSON : {"message": “Login successful”, “user_id”: 1}

Actual Output:

- HTTP Status Code : 200
- JSON : {"message": “Login successful” ,“user_id”: 1}

Result: Success



```python
# login api - correct login credentials
def test_login_correct_credentials(self, client):
    payload = {
        "username": "mdikram888",
        "password": "12345678"
    }
    response = client.post('/api/login', json=payload)
    # Check that the response is as expected
    assert response.status_code == 200
    assert response.json == {
                                "message": "Login successful",
                                "user_id": 7
                            }
```

Objective: Try new user signup with missing roll number

Api being tested: http://127.0.0.1:5001/api/signup

Input:

- Request Method : POST
- JSON: { "username": "username4","email": "stone388019@gmail.com", "password": "acdb1234", "name": "bimleshkanth","reset_code": "202001","dual_degree": False, "side_work": False,"additional_education": "PhD"}

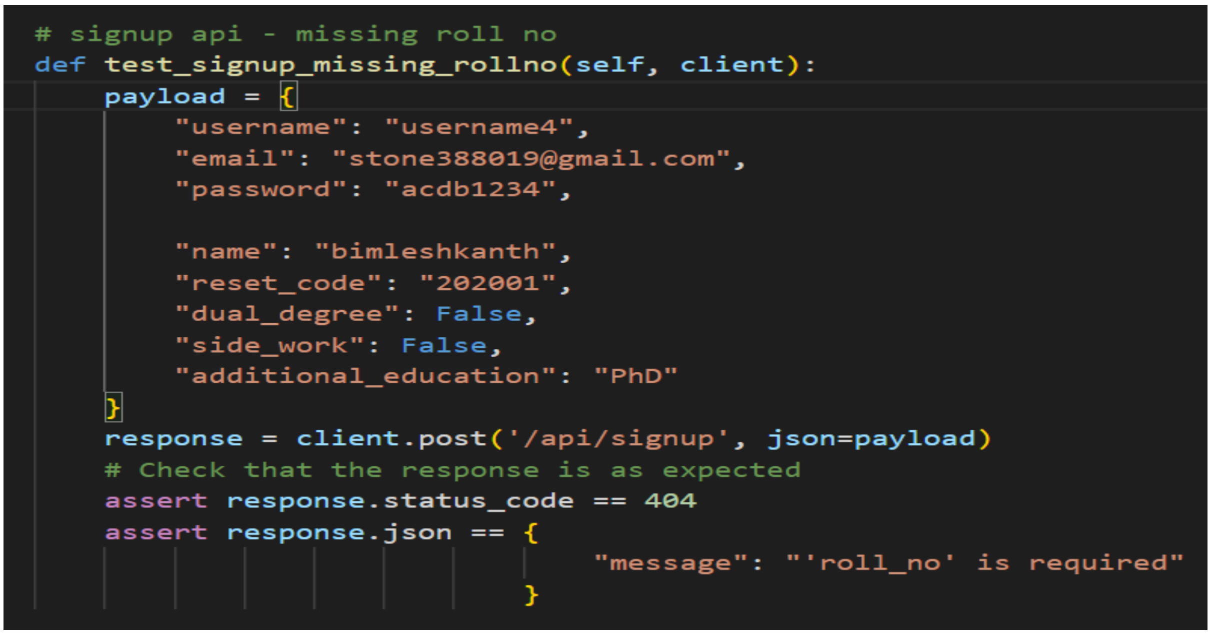
Expected Output:

- HTTP Status Code : 404
- JSON : {"message": " 'roll_no' is required"}

Actual Output:

- HTTP Status Code : 404
- JSON : {"message": " 'roll_no' is required"}

Result: Success



```python
# signup api - missing roll no
def test_signup_missing_rollno(self, client):
    payload = {
        "username": "username4",
        "email": "stone388019@gmail.com",
        "password": "acdb1234",

        "name": "bimleshkanth",
        "reset_code": "202001",
        "dual_degree": False,
        "side_work": False,
        "additional_education": "PhD"
    }
    response = client.post('/api/signup', json=payload)
    # Check that the response is as expected
    assert response.status_code == 404
    assert response.json == {
                                "message": "'roll_no' is required"
                            }
```

Objective: Try entering non-alphanumeric username while doing a password reset

Api being tested: http://127.0.0.1:5001/api/forget_password

Input:

- Request Method : POST
- JSON : { "username": "username4@#", "new_password": "newpwd1234", "reset_code": "202001"}

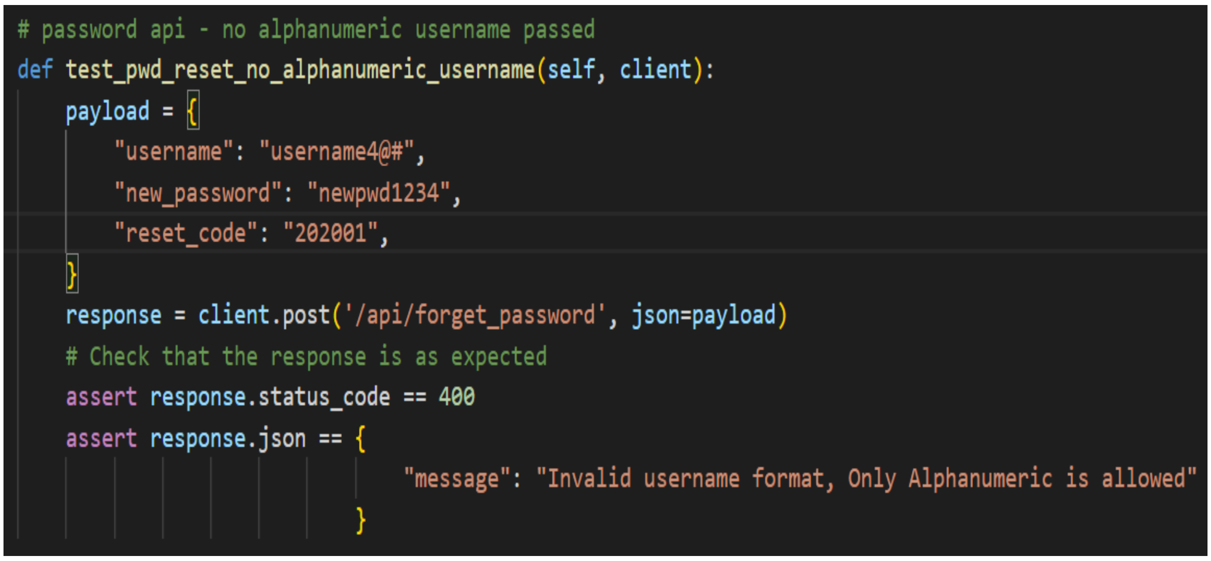
Expected Output:

- HTTP Status Code : 400
- JSON : {"message": " Invalid username format, Only Alphanumeric is allowed"}

Actual Output:

- HTTP Status Code : 400
- JSON : {"message": " Invalid username format, Only Alphanumeric is allowed"}

Result: Success



```python
# password api - no alphanumeric username passed
def test_pwd_reset_no_alphanumeric_username(self, client):
    payload = {
        "username": "username4@#",
        "new_password": "newpwd1234",
        "reset_code": "202001",
    }
    response = client.post('/api/forget_password', json=payload)
    # Check that the response is as expected
    assert response.status_code == 400
    assert response.json == {
                                "message": "Invalid username format, Only Alphanumeric is allowed"
                            }
```

Objective: Try updating user profile email with an existing email in database

Api being tested: http://127.0.0.1:5001/api/profile/8

Input:

- Request Method : PUT
- JSON: { "email": "game388019@gmail.com", "password": "acdbe1234", "username": "username4", "name": "bimleshkanth", "reset_code": "202001","dual_degree": False, "side_work": False, "additional_education": "PhD"}

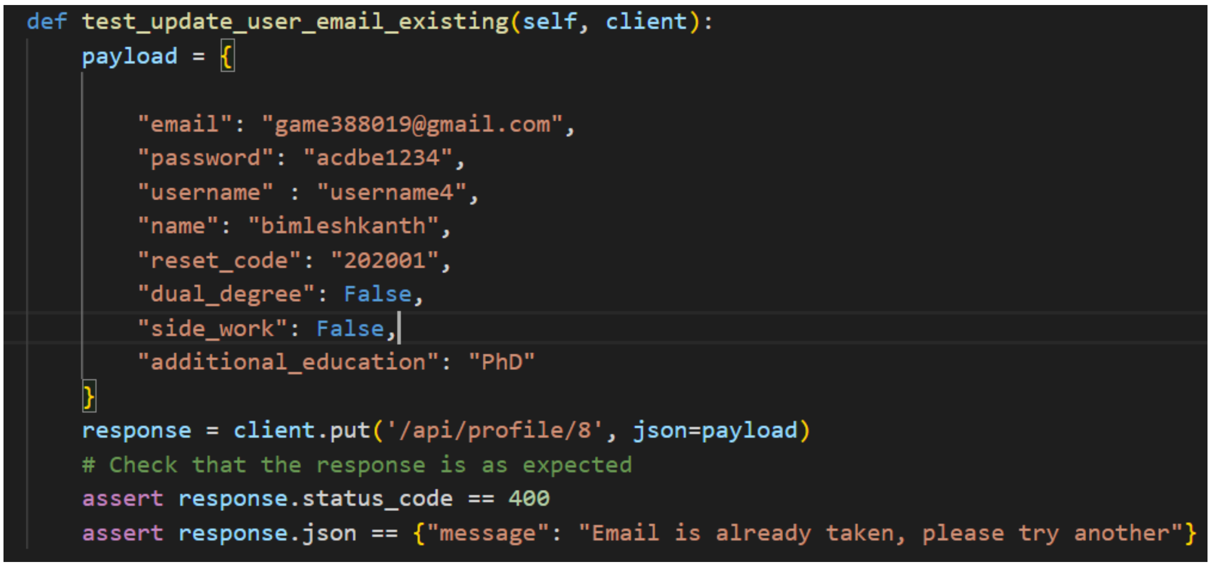
Expected Output:

- HTTP Status Code : 400
- JSON : {"message": " Email is already taken, please try another"}

Actual Output:

- HTTP Status Code : 400
- JSON : {"message": Email is already taken, please try another"}

Result: Success



```python
def test_update_user_email_existing(self, client):
    payload = {

        "email": "game388019@gmail.com",
        "password": "acdbe1234",
        "username" : "username4",
        "name": "bimleshkanth",
        "reset_code": "202001",
        "dual_degree": False,
        "side_work": False,
        "additional_education": "PhD"
    }
    response = client.put('/api/profile/8', json=payload)
    # Check that the response is as expected
    assert response.status_code == 400
    assert response.json == {"message": "Email is already taken, please try another"}
```

Objective: To approve new student success

Api being tested: http://127.0.0.1:5001 /api/student/approve

Input:

- Request Method : PUT
- JSON: {"user_id": 1}

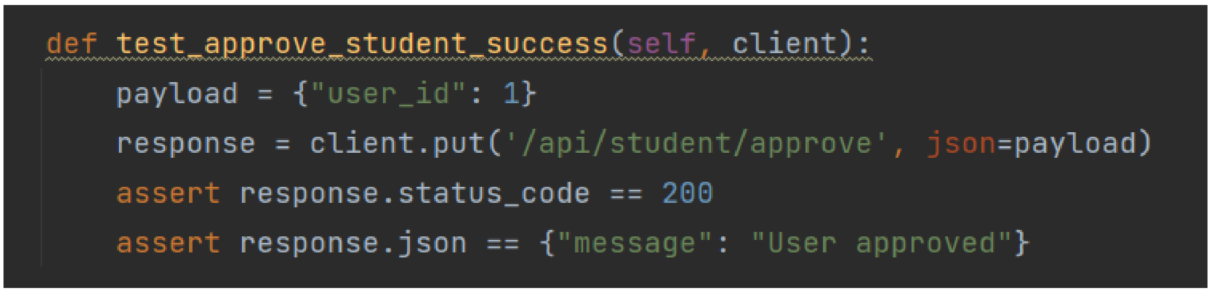
Expected Output:

- HTTP Status Code : 200
- JSON : {"message": "User approved"}

Actual Output:

- HTTP Status Code : 400
- JSON : {"message": Email is already taken, please try another"}

Result: Success



```python
def test_approve_student_success(self, client):
    payload = {"user_id": 1}
    response = client.put('/api/student/approve', json=payload)
    assert response.status_code == 200
    assert response.json == {"message": "User approved"}
```

Objective: To get the Course Details

Page being tested: http://127.0.0.1:5000/api/course/-1

Input:

- Request Method : GET

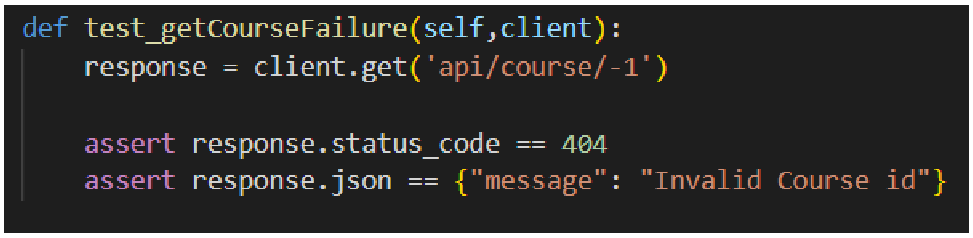
Expected Output:

- HTTP Status Code : 404
- JSON : {"message": "Invalid Course id"}

Actual Output:

- HTTP Status Code : 404
- JSON : {"message": "Invalid Course id"}

Result: Success



```
def test_getCourseFailure(self,client):
    response = client.get('api/course/-1')

    assert response.status_code == 404
    assert response.json == {"message": "Invalid Course id"}
```

Objective: To add a new course with already taken course code

Page being tested: http://127.0.0.1:5000/api/course

Input:

- Request Method : POST
- JSON : {

  "code": "CS001215", "course_id": 2,

  "course_type": "Diploma in Programming",

  "description": "Testing API POST", "duration": "8 weeks",

  "is_active": True, "level": "1",

  "name": "Python", "professor": "It is Me"

  }

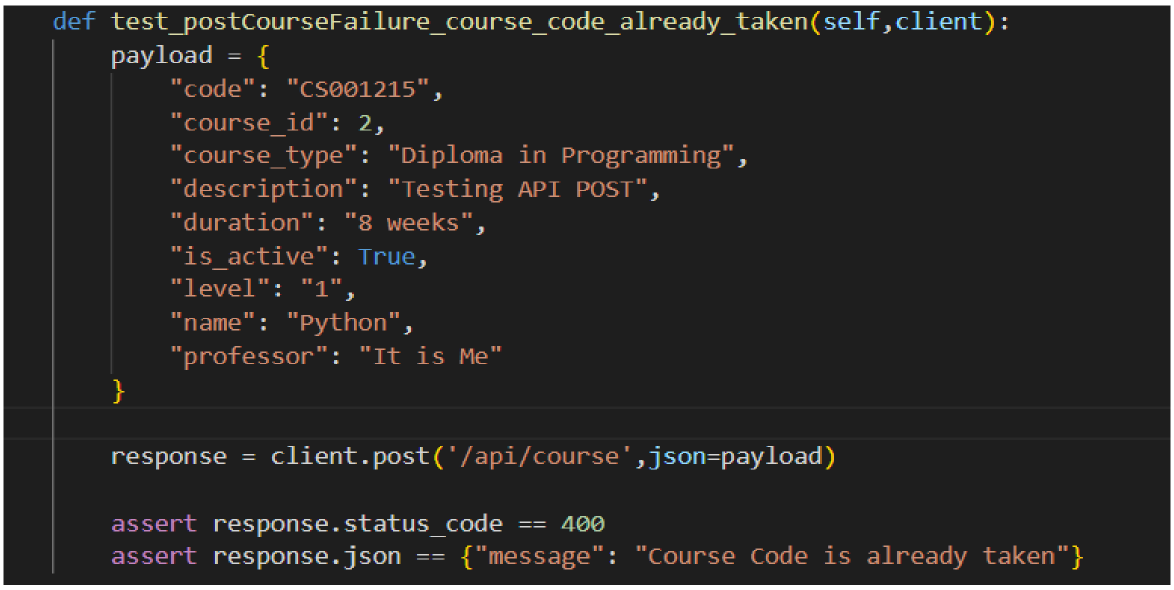
Expected Output:

- HTTP Status Code : 400
- JSON : {"message": "Course Code is already taken"}

Actual Output:

- HTTP Status Code : 400
- JSON : {"message": "Course Code is already taken"}

Result: Success



```
def test_postCourseFailure_course_code_already_taken(self,client):
    payload = {
        "code": "CS001215",
        "course_id": 2,
        "course_type": "Diploma in Programming",
        "description": "Testing API POST",
        "duration": "8 weeks",
        "is_active": True,
        "level": "1",
        "name": "Python",
        "professor": "It is Me"
    }

    response = client.post('/api/course',json=payload)

    assert response.status_code == 400
    assert response.json == {"message": "Course Code is already taken"}
```

Objective: Delete a course

Page being tested: http://127.0.0.1:5000/api/course

Input:

- Request Method : DELETE

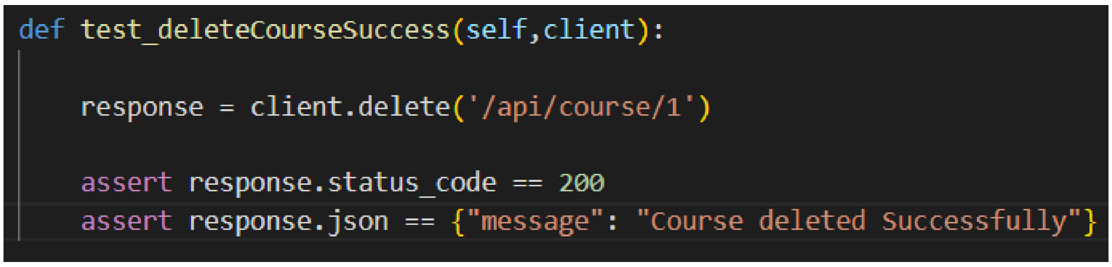
Expected Output:

- HTTP Status Code : 200
- JSON : {"message": "Course deleted Successfully"}

Actual Output:

- HTTP Status Code : 200
- JSON : {"message": "Course deleted Successfully"}

Result: Success



```python
def test_deleteCourseSuccess(self,client):

    response = client.delete('/api/course/1')

    assert response.status_code == 200
    assert response.json == {"message": "Course deleted Successfully"}
```

Objective: Update Course details

Page being tested: http://127.0.0.1:5000/api/course

Input:

- Request Method : PUT
- JSON: {

    "code": "CS123456", "course_id": 2,

    "course_type": "Diploma in Programming",

    "description": "Testing API POST",

    "duration": "8 weeks", "is_active": True,

    "level": "1", "name": "SE",

    "professor": "It is Ranjith"

  }

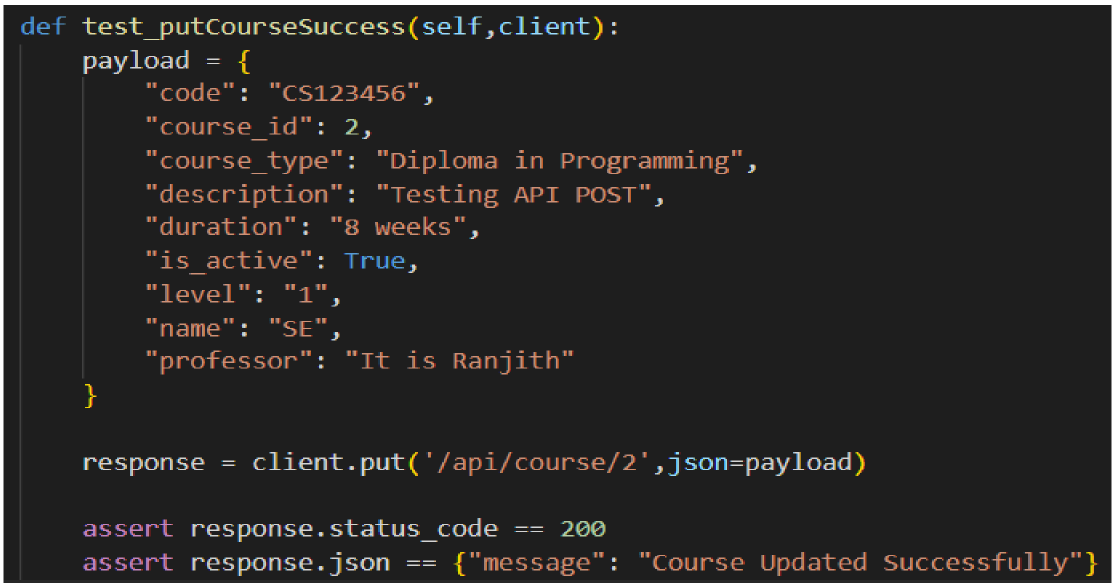
Expected Output:

- HTTP Status Code : 200
- JSON : {"message": "Course Updated Successfully"}

Actual Output:

- HTTP Status Code : 200
- JSON : {"message": "Course Updated Successfully"}

Result: Success



```python
def test_putCourseSuccess(self,client):
    payload = {
        "code": "CS123456",
        "course_id": 2,
        "course_type": "Diploma in Programming",
        "description": "Testing API POST",
        "duration": "8 weeks",
        "is_active": True,
        "level": "1",
        "name": "SE",
        "professor": "It is Ranjith"
    }

    response = client.put('/api/course/2',json=payload)

    assert response.status_code == 200
    assert response.json == {"message": "Course Updated Successfully"}
```

Objective: Get all courses registered by certain student

Page being tested: http://127.0.0.1:5000/courses/student/1

Input:

- Request Method : GET

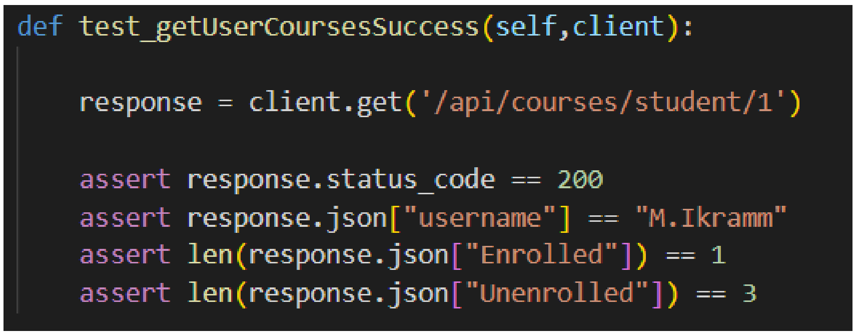
Expected Output:

- HTTP Status Code : 200

Actual Output:

- HTTP Status Code : 200

Result: Success



```python
def test_getUserCoursesSuccess(self,client):

    response = client.get('/api/courses/student/1')

    assert response.status_code == 200
    assert response.json["username"] == "M.Ikramm"
    assert len(response.json["Enrolled"]) == 1
    assert len(response.json["Unenrolled"]) == 3
```

**Objective : **To get the enrollments of a particular student which does not exist

Page being tested: http://127.0.0.1:5000/api/enrollments/student/1000

Input:

- Request Method : GET

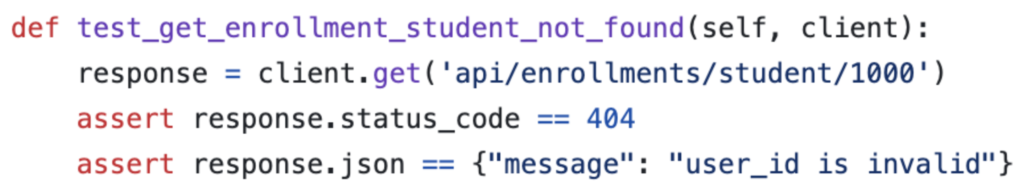
Expected Output:

- HTTP Status Code : 404
- JSON : { “message” : “user_id is invalid” }

Actual Output:

- HTTP Status Code : 404
- JSON : { “message” : “user_id is invalid” }

Result: Success



```
def test_get_enrollment_student_not_found(self, client):
    response = client.get('api/enrollments/student/1000')
    assert response.status_code == 404
    assert response.json == {"message": "user_id is invalid"}
```

**Objective : **To get the enrollments of all students

Page being tested: http://127.0.0.1:5000/api/enrollments/admin

Input:

- Request Method : GET

Expected Output:

- HTTP Status Code : 200
- JSON : { “data” : [{“course_name” : “Java” , “user_name” : “M.Ikramm”} , {“course_name” : “SE” , “user_name” : “Mohd Ikram”} , {“course_name” : “FastX” , “user_name” : “M.Ikramm”} , “course_name” : “SE” , “user_name” : “Rahul”] }

Actual Output:

- HTTP Status Code : 200
- JSON : { “data” : [{“course_name” : “Java” , “user_name” : “M.Ikramm”} , {“course_name” : “SE” , “user_name” : “Mohd Ikram”} , {“course_name” : “FastX” , “user_name” : “M.Ikramm”} , “course_name” : “SE” , “user_name” : “Rahul”] }

Result: Success

```python
def test_all_enrollment(self, client):
    response = client.get('api/enrollments/admin')
    assert response.status_code == 200
    assert "data" in response.json
    assert type(response.json["data"]).__name__ == 'list'
```

**Objective : ** To update enrollment details of a preexisting enrollment.

Page being tested: http://127.0.0.1:5000/api/enrollments/5

Input:

- Request Method : PUT
- Payload : { “marks” : 70, “term” : “Sept”, “year” : 2022, “study_hours” : 30 }

Expected Output:

- HTTP Status Code : 202
- JSON : { “message” : “Enrollment Updated Successfully” }

Actual Output:

- HTTP Status Code : 404
- JSON : { “message” : “Enrollment Updated Successfully” }

Result: Success

```
def test_put_enrollment_id_success(self, client):
    payload = {
                    "marks": 70,
                    "term": "Sept",
                    "year": 2022,
                    "study_hours": 30
                }
    response = client.put('api/enrollments/5', json=payload)
    assert response.status_code == 202
    assert response.json == {"message": "Enrollment Updated Successfully"}
```

Objective : To add enrollment details of a user with invalid course ID.

Page being tested: http://127.0.0.1:5000/api/enrollments/

Input:

- Request Method : POST
- Payload : { “course_id” : 200, “user_id” : 3, “marks”: 70, “term” : “Sept”, “year” : 2022, “study_hours” : 20 }

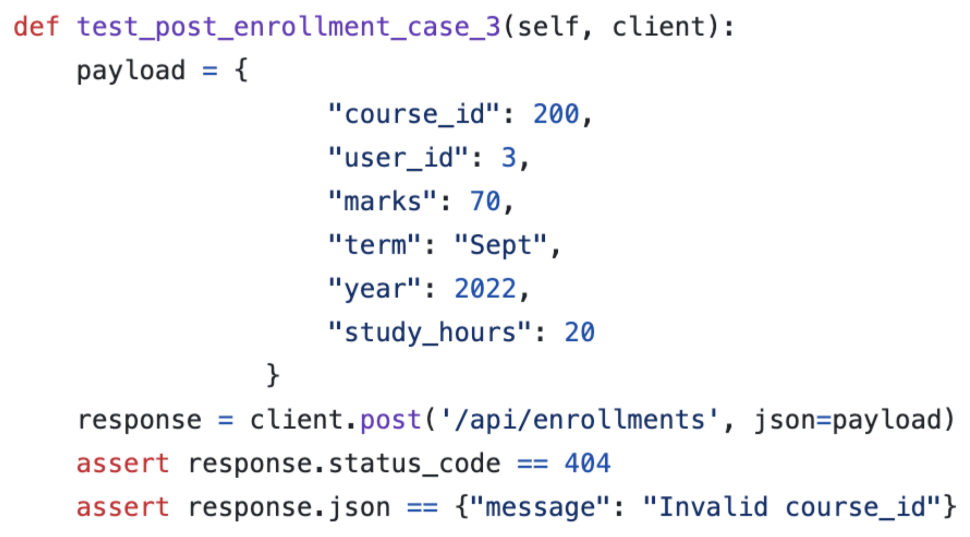
Expected Output:

- HTTP Status Code : 404
- JSON : { “message” : “Invalid course_id” }

Actual Output:

- HTTP Status Code : 404
- JSON : { “message” : “Invalid course_id” }

Result: Success



```
def test_post_enrollment_case_3(self, client):
    payload = {
                    "course_id": 200,
                    "user_id": 3,
                    "marks": 70,
                    "term": "Sept",
                    "year": 2022,
                    "study_hours": 20
                }
    response = client.post('/api/enrollments', json=payload)
    assert response.status_code == 404
    assert response.json == {"message": "Invalid course_id"}
```

Objective : To add enrollment details of a user with a new course.

Page being tested: http://127.0.0.1:5000/api/enrollments/

Input:

- Request Method : POST
- Payload : { “course_id” : 1, “user_id” : 3, “marks”: 100, “term” : “Sept”, “year” : 2023, “study_hours” : 40 }

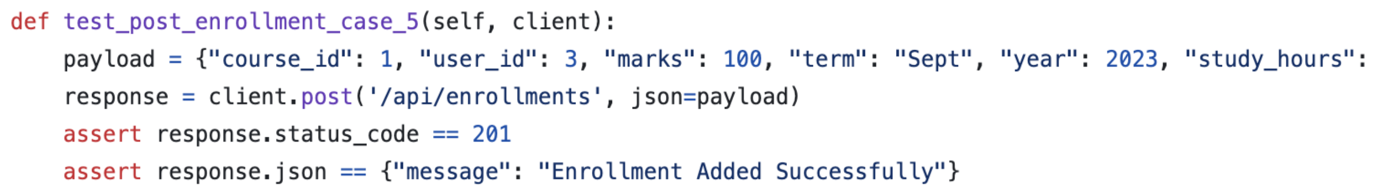
Expected Output:

- HTTP Status Code : 201
- JSON : { “message” : “Enrollment Added Successfully” }

Actual Output:

- HTTP Status Code : 201
- JSON : { “message” : “Enrollment Added Successfully” }

Result: Success



```
def test_post_enrollment_case_5(self, client):
    payload = {"course_id": 1, "user_id": 3, "marks": 100, "term": "Sept", "year": 2023, "study_hours":
    response = client.post('/api/enrollments', json=payload)
    assert response.status_code == 201
    assert response.json == {"message": "Enrollment Added Successfully"}
```

Objective : To delete an enrollment.

Page being tested: http://127.0.0.1:5000/api/enrollments/6

Input:

- Request Method : DELETE

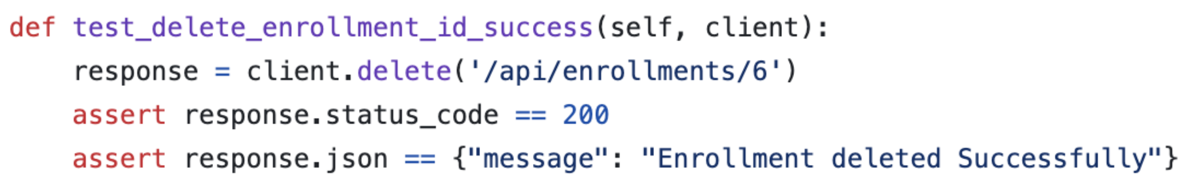
Expected Output:

- HTTP Status Code : 200
- JSON : { “message” : “Enrollment deleted Successfully” }

Actual Output:

- HTTP Status Code : 200
- JSON : { “message” : “Enrollment deleted Successfully” }

Result: Success



```python
def test_delete_enrollment_id_success(self, client):
    response = client.delete('/api/enrollments/6')
    assert response.status_code == 200
    assert response.json == {"message": "Enrollment deleted Successfully"}
```

Objective : Add review Successfully.

Api being tested: http://127.0.0.1:5001/api/review

Input:

- Request Method : POST
- JSON: {“user_id”: 1, “course_id”: 1, “difficulty”: 5, “support”: 3, “rating”: 4,

  “review”: “blab blabalba”

Expected Output:

- HTTP Status Code : 201
- JSON : {"message": "Review added successfully"

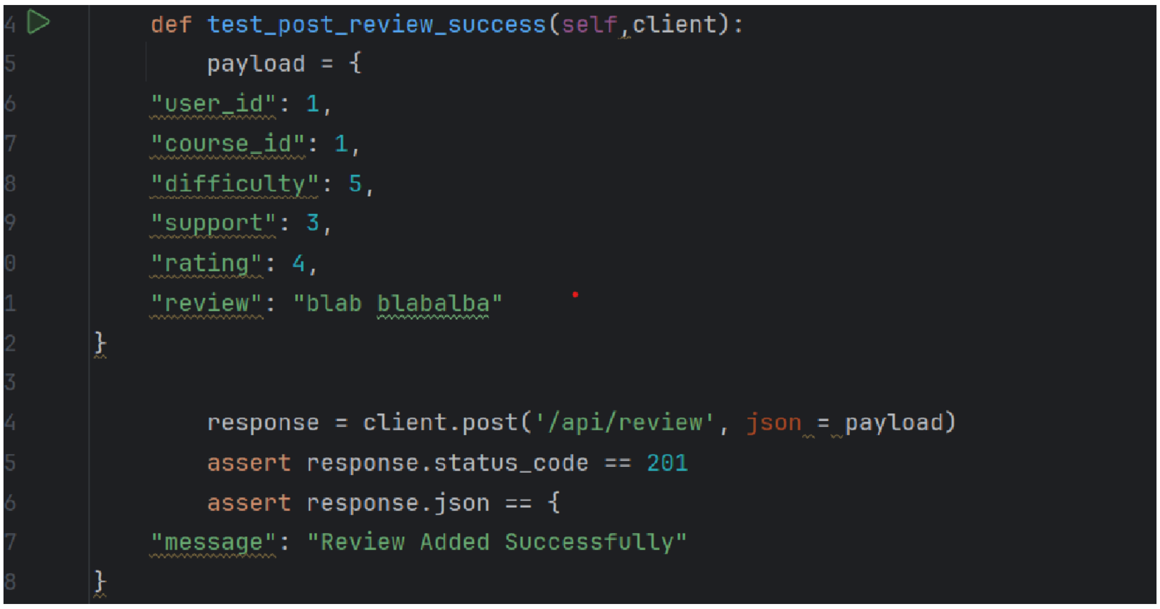
  }

Actual Output:

- HTTP Status Code : 201
- JSON : {"message": "Review added successfully"

  }

Result: Success



```python
    def test_post_review_success(self,client):
        payload = {
    "user_id": 1,
    "course_id": 1,
    "difficulty": 5,
    "support": 3,
    "rating": 4,
    "review": "blab blabalba"
}

        response = client.post('/api/review', json = payload)
        assert response.status_code == 201
        assert response.json == {
    "message": "Review Added Successfully"
}
```

Objective: Get Review - to retrieve review from the database Successfully

Api being tested: 127.0.0.1:5001/api/review/2

Input:

- Request Method : GET

Expected Output:

- HTTP Status Code : 200
- JSON : { "data": { "course_id": 1, "course_name": "Java", "difficulty": 1,

  "rating": 5, "review": "23", "review_id": 2,
  "support": 4,

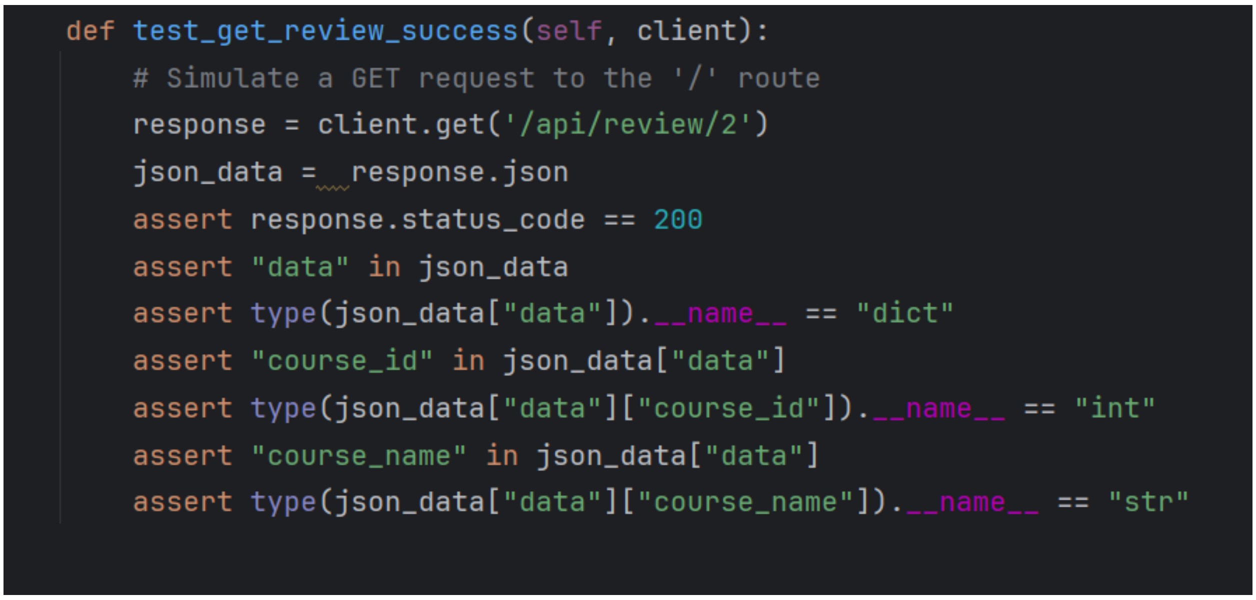
  "user_id": 5 } }

Actual Output:

- HTTP Status Code : 200
- JSON : { "data": { "course_id": 1, "course_name": "Java", "difficulty": 1,

  "rating": 5, "review": "23", "review_id": 2, "support": 4,
  "user_id": 5} }

- **Result:** Success



```python
def test_get_review_success(self, client):
    # Simulate a GET request to the '/' route
    response = client.get('/api/review/2')
    json_data =  response.json
    assert response.status_code == 200
    assert "data" in json_data
    assert type(json_data["data"]).__name__ == "dict"
    assert "course_id" in json_data["data"]
    assert type(json_data["data"]["course_id"]).__name__ == "int"
    assert "course_name" in json_data["data"]
    assert type(json_data["data"]["course_name"]).__name__ == "str"
```

Objective: Edit Review Success

Api being tested: 127.0.0.1:5001/api/review

Input:

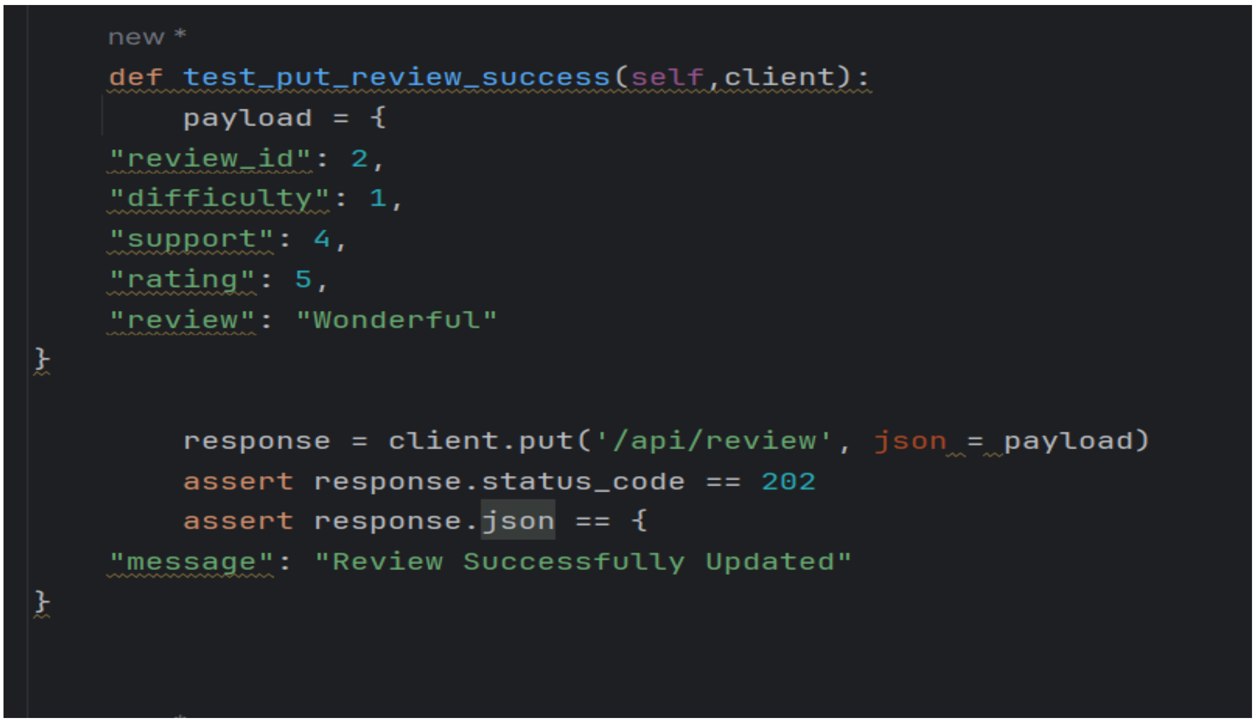
- Request Method : PUT
- JSON: { "review_id": 2, "difficulty": 1, "support": 4, "rating": 5, "review": "Wonderful"}

Expected Output:

- HTTP Status Code : 202
- JSON : { "message": "Review Successfully Updated" }

Actual Output:

- HTTP Status Code : 202
- JSON : { "message": "Review Successfully Updated"}
- **Result:** Success



```
    new *
    def test_put_review_success(self,client):
        payload = {
    "review_id": 2,
    "difficulty": 1,
    "support": 4,
    "rating": 5,
    "review": "Wonderful"
}

        response = client.put('/api/review', json = payload)
        assert response.status_code == 202
        assert response.json == {
    "message": "Review Successfully Updated"
}
```

Objective: Get Student Review Success

API being tested: 127.0.0.1:5001/api/reviews/student/1

Input:

- Request Method : GET

Expected Output:

- HTTP Status Code : 200
- JSON :{ "data": [

  {"course_id": 2, "course_name": "SE","difficulty": 5, "rating": 4, "review": "blab blabalba", "review_id": 3, "support": 3, "user_id": 1}

  ,

  { "course_id": 1, "course_name": "Java", "difficulty": 5, "rating": 4,

  "review": "blab blabalba", "review_id": 5, "support": 3, "user_id":

  1

  }]}

Actual Output:

- HTTP Status Code : 200
- JSON : { "data": [

  {"course_id": 2, "course_name": "SE","difficulty": 5, "rating": 4, "review": "blab blabalba", "review_id": 3, "support": 3, "user_id": 1}

  ,

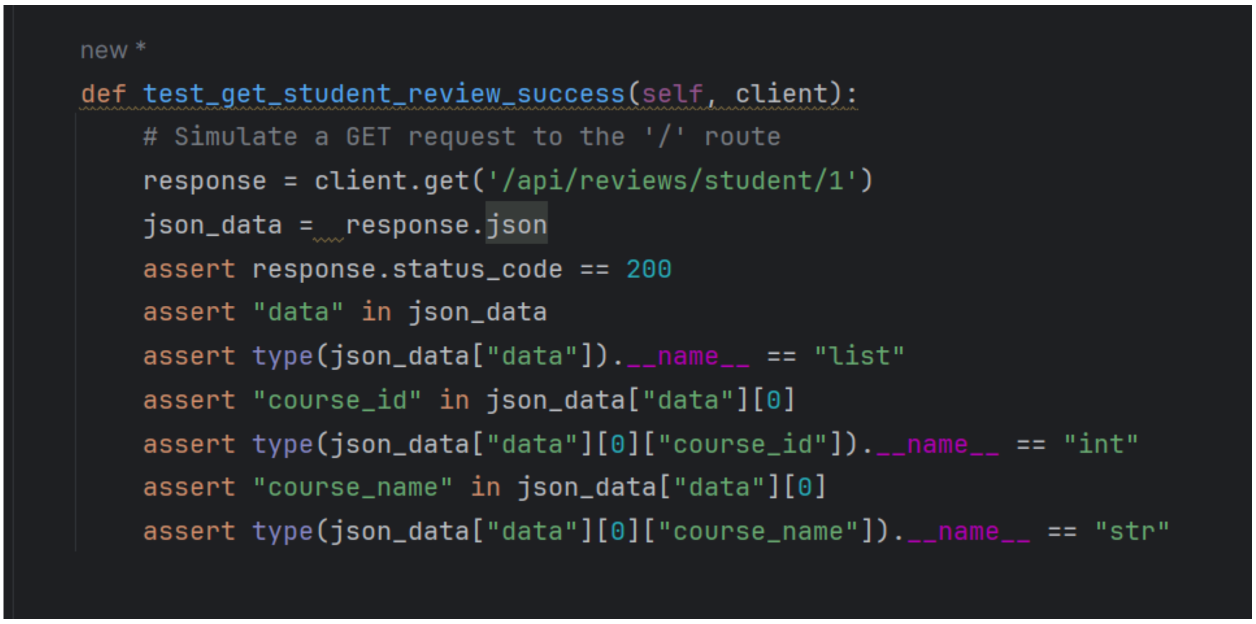
  { "course_id": 1, "course_name": "Java", "difficulty": 5, "rating": 4,

  "review": "blab blabalba", "review_id": 5, "support": 3, "user_id":

  1

  }]}

- **Result:** Success



```
new *
def test_get_student_review_success(self, client):
    # Simulate a GET request to the '/' route
    response = client.get('/api/reviews/student/1')
    json_data =  response.json
    assert response.status_code == 200
    assert "data" in json_data
    assert type(json_data["data"]).__name__ == "list"
    assert "course_id" in json_data["data"][0]
    assert type(json_data["data"][0]["course_id"]).__name__ == "int"
    assert "course_name" in json_data["data"][0]
    assert type(json_data["data"][0]["course_name"]).__name__ == "str"
```

Objective: Delete review Success/Failure

Api being tested: 127.0.0.1:5001/api/review/5

Input:

- Request Method : DELETE

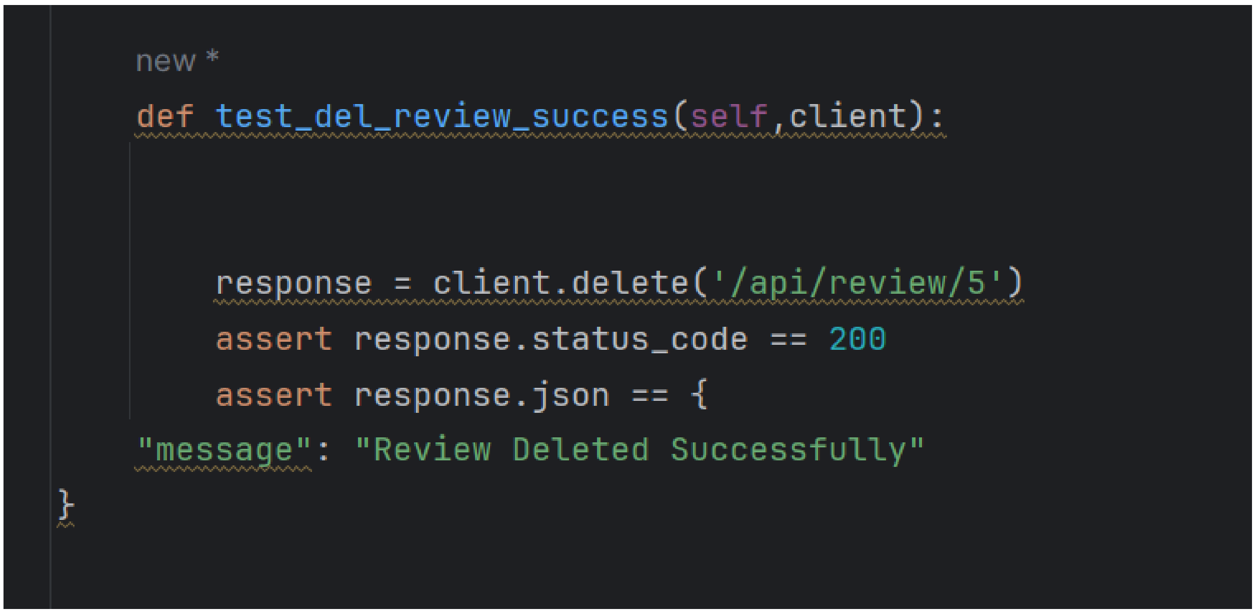
Expected Output:

- HTTP Status Code : 200
- JSON : {"message": "Review Deleted Successfully"}

Actual Output:

- HTTP Status Code : 200
- JSON : {"message": "Review Deleted successfully" }

Result: Success



```
new *
def test_del_review_success(self,client):


    response = client.delete('/api/review/5')
    assert response.status_code == 200
    assert response.json == {
"message": "Review Deleted Successfully"
}
```

Objective: To get the Admin Course List

Page being tested: http://127.0.0.1:5000/api/admin_courses_list

Input:

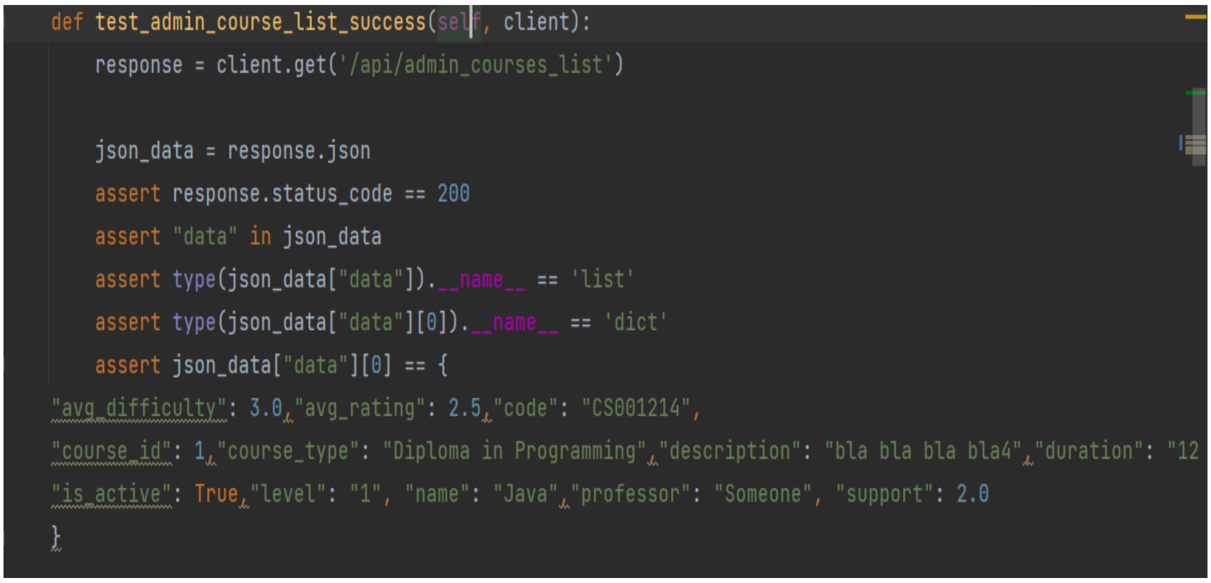
- Request Method : GET

Expected Output:

- HTTP Status Code : 200
- JSON :
```
{“data”: [ {
    "avg_difficulty": 3.0,
    "avg_rating": 2.5,
    "code": "CS001214",
    "course_id": 1,
    "course_type": "Diploma in Programming",
    "description": "bla bla bla bla4",
    "duration": "12 weeks",
    "is_active": True,
    "level": "1",
    "name": "Java",
    "professor": "Someone",
    "support": 2.0
        }    ] }
```

- **Actual Output:**
- HTTP Status Code : 200
- JSON :
```
{ “data” : [ {
    "avg_difficulty": 3.0,
    "avg_rating": 2.5,
    "code": "CS001214",
    "course_id": 1,
    "course_type": "Diploma in Programming",
    "description": "bla bla bla bla4",
    "duration": "12 weeks",
    "is_active": True,
    "level": "1",
    "name": "Java", "professor": "Someone",  "support":
    2.0  } ]  }
```

Result: Success



```
def test_admin_course_list_success(self, client):
    response = client.get('/api/admin_courses_list')

    json_data = response.json
    assert response.status_code == 200
    assert "data" in json_data
    assert type(json_data["data"]).__name__ == 'list'
    assert type(json_data["data"][0]).__name__ == 'dict'
    assert json_data["data"][0] == {
"avg_difficulty": 3.0,"avg_rating": 2.5,"code": "CS001214",
"course_id": 1,"course_type": "Diploma in Programming","description": "bla bla bla bla4","duration": "12
"is_active": True,"level": "1", "name": "Java","professor": "Someone", "support": 2.0
}
```

Objective: To Update Course Status by admin

Page being tested: http://127.0.0.1:5000/api/update_course_status

Input:

- Request Method : PUT
- JSON : {"course_id": -1}

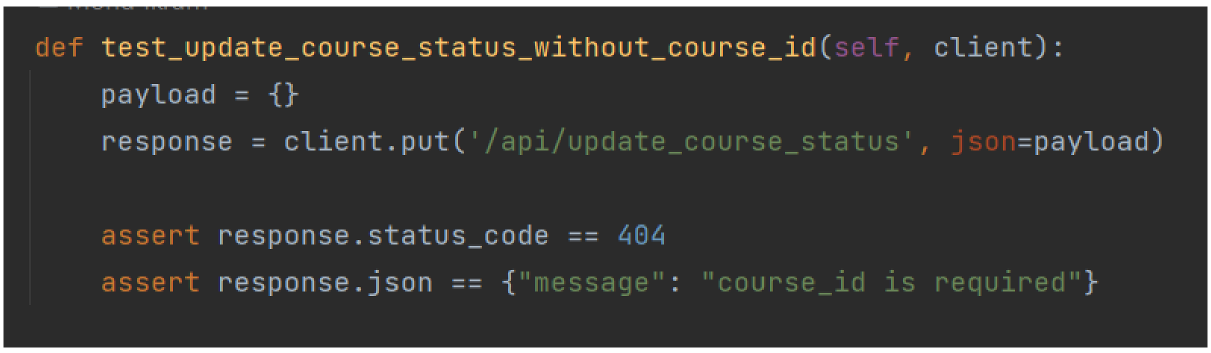
Expected Output:

- HTTP Status Code : 404
- JSON : {"message": "Invalid Course Id"}

Actual Output:

- HTTP Status Code : 404
- JSON : {"message": "Invalid Course Id"}

Result: Success



```
def test_update_course_status_without_course_id(self, client):
    payload = {}
    response = client.put('/api/update_course_status', json=payload)

    assert response.status_code == 404
    assert response.json == {"message": "course_id is required"}
```

Objective: Get basic user Info

Page being tested: http://127.0.0.1:5000/api/get_user_name/2

Input:

- Request Method : GET

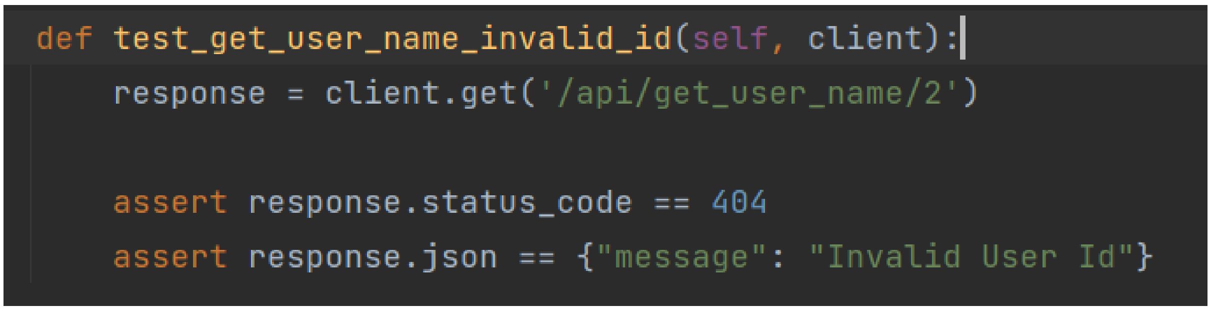
Expected Output:

- HTTP Status Code : 404
- JSON : {"message": "Invalid User Id"}

Actual Output:

- HTTP Status Code : 404
- JSON : {"message": "Invalid User Id"}

Result: Success



```
def test_get_user_name_invalid_id(self, client):
    response = client.get('/api/get_user_name/2')

    assert response.status_code == 404
    assert response.json == {"message": "Invalid User Id"}
```

Objective: To Use Course Recommender

Page being tested: http://127.0.0.1:5000/api/course_recommender

Input:

- Request Method : POST
- JSON: {
- "user_id": 1,
- "no_of_courses": 1,
- "study_hour": 20,
- "level": 1
- }

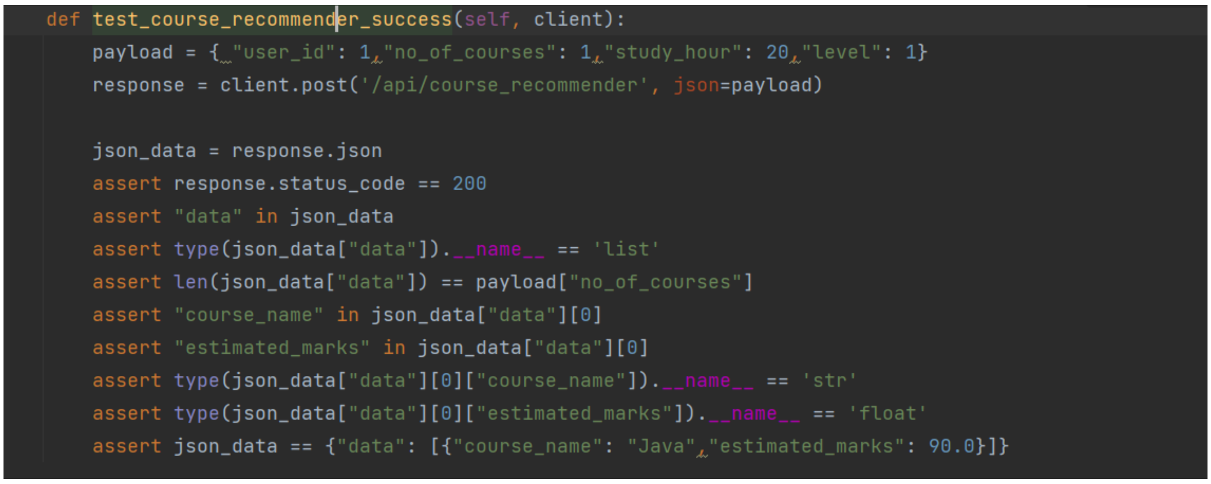
Expected Output:

- HTTP Status Code : 200
- JSON : { "data": [{ "course_name": "Java", "estimated_marks": 90.0 }] }

Actual Output:

- HTTP Status Code : 200
- JSON : { "data": [{ "course_name": "Java", "estimated_marks": 90.0 }] }

Result: Success



```
def test_course_recommender_success(self, client):
    payload = { "user_id": 1,"no_of_courses": 1,"study_hour": 20,"level": 1}
    response = client.post('/api/course_recommender', json=payload)

    json_data = response.json
    assert response.status_code == 200
    assert "data" in json_data
    assert type(json_data["data"]).__name__ == 'list'
    assert len(json_data["data"]) == payload["no_of_courses"]
    assert "course_name" in json_data["data"][0]
    assert "estimated_marks" in json_data["data"][0]
    assert type(json_data["data"][0]["course_name"]).__name__ == 'str'
    assert type(json_data["data"][0]["estimated_marks"]).__name__ == 'float'
    assert json_data == {"data": [{"course_name": "Java","estimated_marks": 90.0}]}
```

Objective: Get all courses registered by certain student

Page being tested: http://127.0.0.1:5000/courses/student/1

Input:

- Request Method : GET

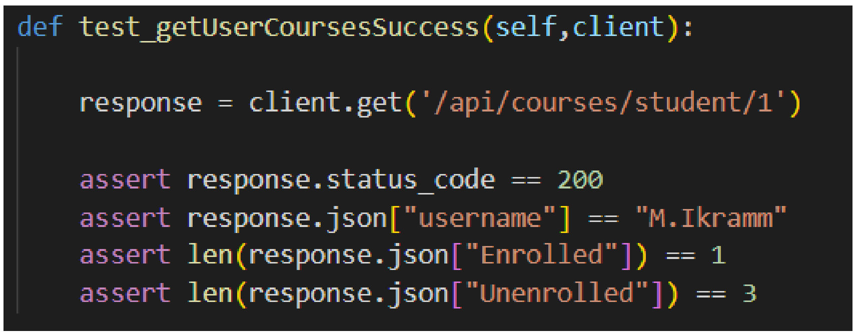
Expected Output:

- HTTP Status Code : 200

Actual Output:

- HTTP Status Code : 200

Result: Success



```
def test_getUserCoursesSuccess(self,client):

    response = client.get('/api/courses/student/1')

    assert response.status_code == 200
    assert response.json["username"] == "M.Ikramm"
    assert len(response.json["Enrolled"]) == 1
    assert len(response.json["Unenrolled"]) == 3
```