

# SE-Sept-19: Milestone 5: Test Cases

Descriptions of relevant API tests are given below.

Page being tested: <http://127.0.0.1:8080/login/>

Inputs:

- Request Method: POST
- JSON: {"username": "john123", "password": "pass123"}

Expected Output:

- HTTP Status Code: 200

Actual Output:

- HTTP Status Code: 200

Result: Success

Python

```
def test_login_success():  
    url = base_url + "login"  
    payload = {"username": "john123", "password": "pass123"}  
    response = requests.post(url, json=payload)  
    assert response.status_code == 200
```

---

Page being tested: <http://127.0.0.1:8080/login>

Inputs:

- Request Method: POST
- JSON: {"username": "john123", "password": "wrongpass"}

Expected Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Actual Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Result: Success

Python

```
def test_login_failure():  
    url = base_url + "login"  
    payload = {"username": "john123", "password": "wrongpass"}  
    response = requests.post(url, json=payload)  
    assert response.status_code == 401  
    assert response.json() == "Incorrect login"
```

---

Page being tested: <http://127.0.0.1:8080/profile>

Inputs:

- Request Method: GET
- Header: {"authToken": "123abc"}

Expected Output:

- HTTP Status Code: 200

Actual Output:

- HTTP Status Code: 200

Result: Success

Python

```
def test_get_student_profile():  
    url = base_url + "profile"  
    headers = {"authToken": "123abc"}  
    response = requests.get(url, headers=headers)  
    assert response.status_code == 200  
    assert "profile_id" in response.json()
```

---

Page being tested: <http://127.0.0.1:8080/profile>

Inputs:

- Request Method: GET
- Header: {"authToken": "123abc"}

Expected Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Actual Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Result: Success

Python

```
def test_get_profile_unauthorized():  
    url = base_url + "profile"  
    headers = {"authToken": "invalidtoken"}  
    response = requests.get(url, headers=headers)  
    assert response.status_code == 401  
    assert response.json() == "Invalid credentials"
```

---

Page being tested: <http://127.0.0.1:8080/notifications>

Inputs:

- Request Method: GET
- Header: {"authToken": "123xyz"}

Expected Output:

- HTTP Status Code: 200

Actual Output:

- HTTP Status Code: 200

Result: Success

Python

```
def test_get_notifications():  
    url = base_url + "notifications"  
    headers = {"authToken": "123xyz"}  
    response = requests.get(url, headers=headers)  
    assert response.status_code == 200  
    assert type(response.json()) is list
```

---

Page being tested: <http://127.0.0.1:8080/notifications>

Inputs:

- Request Method: GET
- Header: {"authToken": "invalid"}

Expected Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Actual Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Result: Success

Python

```
def test_get_notifications_unauthorized():  
    url = base_url + "notifications"  
    headers = {"authToken": "invalid"}  
    response = requests.get(url, headers=headers)  
    assert response.status_code == 401  
    assert response.json() == "Invalid credentials"
```

---

Page being tested: <http://127.0.0.1:8080/recommendations>

Inputs:

- Request Method: GET
- Header: {"authToken": "456def"}

Expected Output:

- HTTP Status Code: 200

Actual Output:

- HTTP Status Code: 200

Result: Success

Python

```
def test_get_recommendations_success():  
    url = base_url + "recommendations"  
    headers = {"authToken": "456def"}  
    response = requests.get(url, headers=headers)  
    assert response.status_code == 200  
    assert type(response.json()) is list
```

---

Page being tested: <http://127.0.0.1:8080/recommendations>

Inputs:

- Request Method: GET
- Header: {"authToken": "invalid"}

Expected Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Actual Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Result: Success

Python

```
def test_get_recommendations_unauthorized():  
    url = base_url + "recommendations"  
    headers = {"authToken": "invalid"}  
    response = requests.get(url, headers=headers)  
    assert response.status_code == 401  
    assert response.json() == "Invalid credentials"
```

---

Page being tested: <http://127.0.0.1:8080/preferences>

Inputs:

- Request Method: PUT
- Header: {"authToken": "123xyz"}
- JSON: {"domains": ["science", "math"], "availability": 10}

Expected Output:

- HTTP Status Code: 200

Actual Output:

- HTTP Status Code: 200

Result: Success

Python

```
def test_update_preferences_success():  
    url = base_url + "preferences"  
    headers = {"authToken": "123xyz"}  
    payload = {"domains": ["science", "math"], "availability": 10}  
    response = requests.put(url, json=payload, headers=headers)  
    assert response.status_code == 200
```

---

Page being tested: <http://127.0.0.1:8080/preferences>

Inputs:

- Request Method: PUT
- Header: {"authToken": "123xyz"}
- JSON: {"domains": ["science", "math"]}

Expected Output:

- HTTP Status Code: 400
- JSON: "Error in input data"

Actual Output:

- HTTP Status Code: 400
- JSON: "Error in input data"

Result: Success

Python

```
def test_update_preferences_bad_request():  
    url = base_url + "preferences"  
    headers = {"authToken": "123xyz"}  
    payload = {"domains": ["science", "math"]}  
    response = requests.put(url, json=payload, headers=headers)  
    assert response.status_code == 400  
    assert response.json() == "Error in input data"
```

---

Page being tested: <http://127.0.0.1:8080/feedback/{courseId}>

Inputs:

- Request Method: POST
- Header: {"authToken": "123abc"}
- JSON: {"rating": 4, "comment": "Good course"}

Expected Output:

- HTTP Status Code: 200

Actual Output:

- HTTP Status Code: 200

Result: Success

Python

```
def test_submit_feedback():
    url = base_url + "feedback/5252"
    headers = {"authToken": "123abc"}
    payload = {"rating": 4, "comment": "Good course"}
    response = requests.post(url, json=payload, headers=headers)
    assert response.status_code == 200
```

---

Page being tested: <http://127.0.0.1:8080/feedback/{courseId}>

Inputs:

- Request Method: POST
- Header: {"authToken": "123abc"}
- JSON: {"rating": 4, "comment": "Good course"}

Expected Output:

- HTTP Status Code: 404
- JSON: "Course not found!"

Actual Output:

- HTTP Status Code: 404
- JSON: "Course not found!"

Result: Success

Python

```
def test_submit_feedback_not_found():
    url = base_url + "feedback/999"
    headers = {"authToken": "123abc"}
    payload = {"rating": 4, "comment": "Good course"}
    response = requests.post(url, json=payload, headers=headers)
    assert response.status_code == 404
    assert response.json() == "Course not found!"
```

---

Page being tested: <http://127.0.0.1:8080/data/anonymize>

Inputs:

- Request Method: POST
- Header: {"authToken": "123abc"}

Expected Output:

- HTTP Status Code: 200

Actual Output:

- HTTP Status Code: 200

Result: Success

Python

```
def test_anonymize_data_success():  
    url = base_url + "data/anonymize"  
    headers = {"authToken": "123abc"}  
    response = requests.post(url, headers=headers)  
    assert response.status_code == 200
```

---

Page being tested: <http://127.0.0.1:8080/data/anonymize>

Inputs:

- Request Method: POST
- Header: {"authToken": "123abc"}

Expected Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Actual Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Result: Success

Python

```
def test_anonymize_data_unauthorized():  
    url = base_url + "data/anonymize"  
    headers = {"authToken": "invalid"}  
    response = requests.post(url, headers=headers)  
    assert response.status_code == 401  
    assert response.json() == "Invalid credentials"
```

---

Page being tested: <http://127.0.0.1:8080/enrollments/upload>

Inputs:

- Request Method: POST
- Header: {"authToken": "admin123"}
- File: CSV\_File

Expected Output:

- HTTP Status Code: 200

Actual Output:

- HTTP Status Code: 200

Result: Success

Python

```
def test_upload_enrollments_success():  
    url = base_url + "enrollments/upload"  
    files = {'file': open('test_data.csv', 'rb')}  
    headers = {"authToken": "admin123"}  
    response = requests.post(url, files=files, headers=headers)  
    assert response.status_code == 200
```

---

Page being tested: <http://127.0.0.1:8080/enrollments/upload>

Inputs:

- Request Method: POST
- Header: {"authToken": "admin123"}
- File: CSV\_File

Expected Output:

- HTTP Status Code: 400
- JSON: "Error in input data"

Actual Output:

- HTTP Status Code: 400
- JSON: "Error in input data"

Result: Success

Python

```
def test_upload_enrollments_bad_request():  
    url = base_url + "enrollments/upload"  
    files = {'file': 'invalid_file'}  
    headers = {"authToken": "admin123"}  
    response = requests.post(url, files=files, headers=headers)  
    assert response.status_code == 400
```

---



Page being tested: <http://127.0.0.1:8080/analytics>

Inputs:

- Request Method: GET
- Header: {"authToken": "admin123"}

Expected Output:

- HTTP Status Code: 200

Actual Output:

- HTTP Status Code: 200

Result: Success

Python

```
def test_get_analytics_success():
    url = base_url + "analytics"
    headers = {"authToken": "admin123"}
    response = requests.get(url, headers=headers)
    assert response.status_code == 200
    assert type(response.json()) is dict
```

---

Page being tested: <http://127.0.0.1:8080/analytics>

Inputs:

- Request Method: GET
- Header: {"authToken": "invalid"}

Expected Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Actual Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Result: Success

Python

```
def test_get_analytics_unauthorized():
    url = base_url + "analytics"
    headers = {"authToken": "invalid"}
    response = requests.get(url, headers=headers)
    assert response.status_code == 401
    assert response.json() == "Invalid credentials"
```

---

Page being tested: <http://127.0.0.1:8080/errors>

Inputs:

- Request Method: GET
- Header: {"authToken": "itsupport123"}
- File: CSV\_File

Expected Output:

- HTTP Status Code: 200

Actual Output:

- HTTP Status Code: 200

Result: Success

Python

```
def test_get_errors_success():  
    url = base_url + "errors"  
    headers = {"authToken": "itsupport123"}  
    response = requests.get(url, headers=headers)  
    assert response.status_code == 200
```

---

Page being tested: <http://127.0.0.1:8080/errors>

Inputs:

- Request Method: GET
- Header: {"authToken": "invalid"}

Expected Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Actual Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Result: Success

Python

```
def test_get_analytics_unauthorized():  
    url = base_url + "analytics"  
    headers = {"authToken": "invalid"}  
    response = requests.get(url, headers=headers)  
    assert response.status_code == 401  
    assert response.json() == "Invalid credentials"
```

---

Page being tested: <http://127.0.0.1:8080/feedback/negative>

Inputs:

- Request Method: GET
- Header: {"authToken": "advisor123"}

Expected Output:

- HTTP Status Code: 200

Actual Output:

- HTTP Status Code: 200

Result: Success

Python

```
def test_get_negative_feedback_success():  
    url = base_url + "feedback/negative"  
    headers = {"authToken": "advisor123"}  
    response = requests.get(url, headers=headers)  
    assert response.status_code == 200
```

---

Page being tested: <http://127.0.0.1:8080/feedback/negative>

Inputs:

- Request Method: GET
- Header: {"authToken": "invalid"}

Expected Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Actual Output:

- HTTP Status Code: 401
- JSON: "Invalid credentials"

Result: Success

Python

```
def test_get_negative_feedback_unauthorized():  
    url = base_url + "feedback/negative"  
    headers = {"authToken": "invalid"}  
    response = requests.get(url, headers=headers)  
    assert response.status_code == 401
```

---