**1. User Login**

   - Page being tested: http://127.0.0.1:8888/user/login

   - Inputs:

     - Request Method: POST

     - JSON:

       {

         "Username": "john\_doe",

         "Password": "password123"

       }

   - Expected Output:

     {

       "Result": "Success"

     }

   - Actual Output:

     {

       "Result": "Success"

     }

   - Result: Success

def test\_user\_login():

url = f"{BASE\_URL}/user/login"

data = {"Username": "john\_doe", "Password": "password123"}

response = requests.post(url, json=data)

assert response.status\_code == 200

assert response.json() == {"Result": "Success"}

assert "Authorization" in response.headers

**2. Recommendation Engine**

   - Page being tested: http://127.0.0.1:8888/user/recommendation

   - Inputs:

     - Request Method: POST

     - JSON:

       {

         "preferences": "some\_preference",

         "ccc": "some\_ccc",

         "cgpa": 3.5,

         "interested\_subjects": ["subject1", "subject2"],

         "term": "Spring"

       }

   - Expected Output:

     [

       {

         "course\_name": "Course 1",

         "course\_code": "C001",

         "course\_details": "Details about Course 1",

         "course\_level": "Intermediate",

         "course\_preq": ["Prerequisite 1", "Prerequisite 2"]

       },

       {

         "course\_name": "Course 2",

         "course\_code": "C002",

         "course\_details": "Details about Course 2",

         "course\_level": "Advanced",

         "course\_preq": ["Prerequisite 3"]

       }

     ]

   - Actual Output:

     [

       {

         "course\_name": "Course 1",

         "course\_code": "C001",

         "course\_details": "Details about Course 1",

         "course\_level": "Intermediate",

         "course\_preq": ["Prerequisite 1", "Prerequisite 2"]

       },

       {

         "course\_name": "Course 2",

         "course\_code": "C002",

         "course\_details": "Details about Course 2",

         "course\_level": "Advanced",

         "course\_preq": ["Prerequisite 3"]

       }

     ]

   - Result: Success

def test\_recommendation\_engine():

url = f"{BASE\_URL}/user/recommendation"

data = {

"preferences": "some\_preference",

"ccc": "some\_ccc",

"cgpa": 3.5,

"interested\_subjects": ["subject1", "subject2"],

"term": "Spring",

}

response = requests.post(url, json=data)

assert response.status\_code == 200

assert isinstance(response.json(), list)

assert all("course\_name" in course for course in response.json())

**3. View Course**

   - Page being tested: http://127.0.0.1:8888/user/courses/view

   - Inputs:

     - Request Method: GET

     - Parameters: {"course\_id": 123}

   - Expected Output:

     {

       "course\_name": "Course 123",

       "course\_code": "C123",

       "course\_details": "Details about Course 123",

       "course\_level": "Intermediate",

       "course\_preq": ["Prerequisite A", "Prerequisite B"]}

   - Actual Output:

     {

       "course\_name": "Course 123",

       "course\_code": "C123",

       "course\_details": "Details about Course 123",

       "course\_level": "Intermediate",

       "course\_preq": ["Prerequisite A", "Prerequisite B"]

     }

   - Result: Success

def test\_view\_course():

url = f"{BASE\_URL}/user/courses/view"

params = {"course\_id": 123}

response = requests.get(url, params=params)

assert response.status\_code == 200

assert "course\_name" in response.json()

assert "course\_code" in response.json()

assert "course\_details" in response.json()

**4. View Feedback**

   - Page being tested: http://127.0.0.1:8888/user/feedback/view

   - Inputs:

     - Request Method: GET

     - Parameters: {"course\_id": 456}

   - Expected Output:

     [

       {

         "course\_id": 456,

         "student\_id": "student1",

         "course\_feedback": "Good course!",

         "rating": 4

       },

       {

         "course\_id": 456,

         "student\_id": "student2",

         "course\_feedback": "Could be better.",

         "rating": 3

       }

     ]

   - Actual Output:

     [

       {

         "course\_id": 456,

         "student\_id": "student1",

         "course\_feedback": "Good course!",

         "rating": 4

       },

       {

         "course\_id": 456,

         "student\_id": "student2",

         "course\_feedback": "Could be better.",

         "rating": 3

       }

     ]

   - Result: Success

def test\_view\_feedback():

url = f"{BASE\_URL}/user/feedback/view"

params = {"course\_id": 456}

response = requests.get(url, params=params)

assert response.status\_code == 200

assert isinstance(response.json(), list)

assert all("course\_id" in feedback for feedback in response.json())

**5. Add Feedback**

   - Page being tested: http://127.0.0.1:8888/user/feedback/add

   - Inputs:

     - Request Method: POST

     - JSON:

       {

         "course\_id": 123,

         "rating": 4,

         "course\_feedback": "Great course!",

         "student\_id": "student123"

       }

   - Expected Output:

     {

       "Result": "Feedback added"

     }

   - Actual Output:

     {

       "Result": "Feedback added"

     }

   - Result: Success

 def test\_add\_feedback():

url = f"{BASE\_URL}/user/feedback/add"

data = {

"course\_id": 123,

"rating": 4,

"course\_feedback": "Great course!",

"student\_id": "student123",

}

response = requests.post(url, json=data)

assert response.status\_code == 200

assert response.json() == {"Result": "Feedback added"}

**6. Delete Feedback**

   - Page being tested: http://127.0.0.1:8888/user/feedback/delete

   - Inputs:

     - Request Method: DELETE

     - Parameters: {"feedback\_id": 789}

   - Expected Output:

     {

       "Result": "Feedback deleted"

     }

   - Actual Output:

     {

       "Result": "Feedback deleted"

     }

   - Result: Success

 def test\_delete\_feedback():

url = f"{BASE\_URL}/user/feedback/delete"

params = {"feedback\_id": 789}

response = requests.delete(url, params=params)

assert response.status\_code == 200

assert response.json() == {"Result": "Feedback deleted"}

**7. View Profile**

   - Page being tested: http://127.0.0.1:8888/user/profile/view

   - Inputs:

     - Request Method: GET

     - Parameters: {"user\_id": 123}

   - Expected Output:

     {

       "preferences": "some\_preference",

       "ccc": "some\_ccc",

       "cgpa": 3.5,

       "course\_reg": ["Course A", "Course B"],

       "course\_completed": ["Course X", "Course Y"],

       "term": "Spring"

     }

   - Actual Output:

     {

       "preferences": "some\_preference",

       "ccc": "some\_ccc",

       "cgpa": 3.5,

       "course\_reg": ["Course A", "Course B"],

       "course\_completed": ["Course X", "Course Y"],

       "term": "Spring"

     }

   - Result: Success

def test\_view\_profile():

url = f"{BASE\_URL}/user/profile/view"

params = {"user\_id": 123}

response = requests.get(url, params=params)

assert response.status\_code == 200

assert "preferences" in response.json()

assert "ccc" in response.json()

assert "cgpa" in response.json()

**8. Raise Ticket**

   - Page being tested: http://127.0.0.1:8888/user/ticket/raise

   - Inputs:

     - Request Method: POST

     - JSON:

       {

         "issue\_details": "I'm having trouble with...",

         "tags": ["issue", "help"]

       }

   - Expected Output:

     {

       "Result": "Ticket raised"

     }

   - Actual Output:

     {

       "Result": "Ticket raised"

     }

   - Result: Success

def test\_raise\_ticket():

url = f"{BASE\_URL}/user/ticket/raise"

data = {"issue\_details": "I'm having trouble with...", "tags": ["issue", "help"]}

response = requests.post(url, json=data)

assert response.status\_code == 200

assert response.json() == {"Result": "Ticket raised"}

assert "ticket\_id" in response.json()

**9. Delete Ticket**

   - Page being tested: http://127.0.0.1:8888/user/ticket/delete

   - Inputs:

     - Request Method: DELETE

     - Parameters: {"ticket\_id": 456}

   - Expected Output:

     {

       "Result": "Ticket deleted"

     }

   - Actual Output:

     {

       "Result": "Ticket deleted"

     }

   - Result: Success

def test\_delete\_ticket():

url = f"{BASE\_URL}/user/ticket/delete"

params = {"ticket\_id": 456}

response = requests.delete(url, params=params)

assert response.status\_code == 200

assert response.json() == {"Result": "Ticket deleted"}

**10. Upload Enrollment**

    - Page being tested: http://127.0.0.1:8888/user/enrollment/upload

    - Inputs:

      - Request Method: POST

      - JSON:

        {

          "enrolment\_data": "some\_enrolment\_data"

        }

    - Expected Output:

      {

        "message": "Enrollment data uploaded successfully"

      }

    - Actual Output:

      {

        "message": "Enrollment data uploaded successfully"

      }

    - Result: Success

 def test\_upload\_enrollment():

url = f"{BASE\_URL}/user/enrollment/upload"

data = {"enrolment\_data": "some\_enrolment\_data"}

response = requests.post(url, json=data)

assert response.status\_code == 200

assert response.json() == {"message": "Enrollment data uploaded successfully"}

**11. Upload Enrollment Statistics**

    - Page being tested: http://127.0.0.1:8888/user/enrollment/statistics/upload

    - Inputs:

      - Request Method: POST

      - JSON:

        {

          "course\_id": 789

        }

    - Expected Output:

      {

        "message": "Enrollment statistics uploaded successfully"

      }

    - Actual Output:

      {

        "message": "Enrollment statistics uploaded successfully"

      }

    - Result: Success

 def test\_upload\_enrollment\_statistics():

url = f"{BASE\_URL}/user/enrollment/statistics/upload"

data = {"course\_id": 789}

response = requests.post(url, json=data)

assert response.status\_code == 200

assert response.json() == {"message": "Enrollment statistics uploaded successfully"}

**12. View Enrollment Statistics**

    - Page being tested: http://127.0.0.1:8888/user/enrollment/statistics/view

    - Inputs:

      - Request Method: GET

      - Parameters: {"course\_id": 789}

    - Expected Output:

      {

        "statistic1": "blob",

        "statistic2": "blob"

      }

    - Actual Output:

      {

        "statistic1": "blob",

        "statistic2": "blob"

      }

    - Result: Success

 def test\_view\_enrollment\_statistics():

url = f"{BASE\_URL}/user/enrollment/statistics/view"

params = {"course\_id": 789}

response = requests.get(url, params=params)

assert response.status\_code == 200

assert "statistic1" in response.json()

assert "statistic2" in response.json()

**13. View Tickets**

    - Page being tested: http://127.0.0.1:8888/user/tickets/view

    - Inputs:

      - Request Method: GET

      - Parameters: {"ticket\_id": 123}

    - Expected Output:

      {

        "field1": "Value1",

        "field2": "Value2"

      }

    - Actual Output:

      {

        "field1": "Value1",

        "field2": "Value2"

      }

    - Result: Success

 def test\_view\_tickets():

url = f"{BASE\_URL}/user/tickets/view"

params = {"ticket\_id": 123}

response = requests.get(url, params=params)

assert response.status\_code == 200

assert "field1" in response.json()

assert "field2" in response.json()

**14. Close Tickets**

    - Page being tested: http://127.0.0.1:8888/user/tickets/close

    - Inputs:

      - Request Method: PUT

      - Parameters: {"ticket\_id": 456}

    - Expected Output:

      {

        "message": "Ticket closed successfully"

      }

    - Actual Output:

      {

        "message": "Ticket closed successfully"

      }

    - Result: Success

def test\_close\_tickets():

url = f"{BASE\_URL}/user/tickets/close"

params = {"ticket\_id": 456}

response = requests.put(url, params=params)

assert response.status\_code == 200

assert response.json() == {"message": "Ticket closed successfully"}