Project Title: Academic feedback system

Student Name: Nidhi Dattani

Enrolment no.: 92200133019

Subject: Capstone Project

Department: Information and Communication Technology

Academic Year: 2025-26

MARWADI UNIVERSITY
Rajkot-Morbi Road, At & Po. Gauridad,
Rajkot-360003, Gujarat, India

Testing and Validation

Introduction

This section outlines my plan for rigorously testing and validating the **Academic feedback System**. The goal is to prove that the system is not only functional but also reliable and that it effectively meets all the project's original objectives and stakeholder needs. I will use a systematic approach, including specific tests and performance measurements, to provide clear evidence of the system's quality.

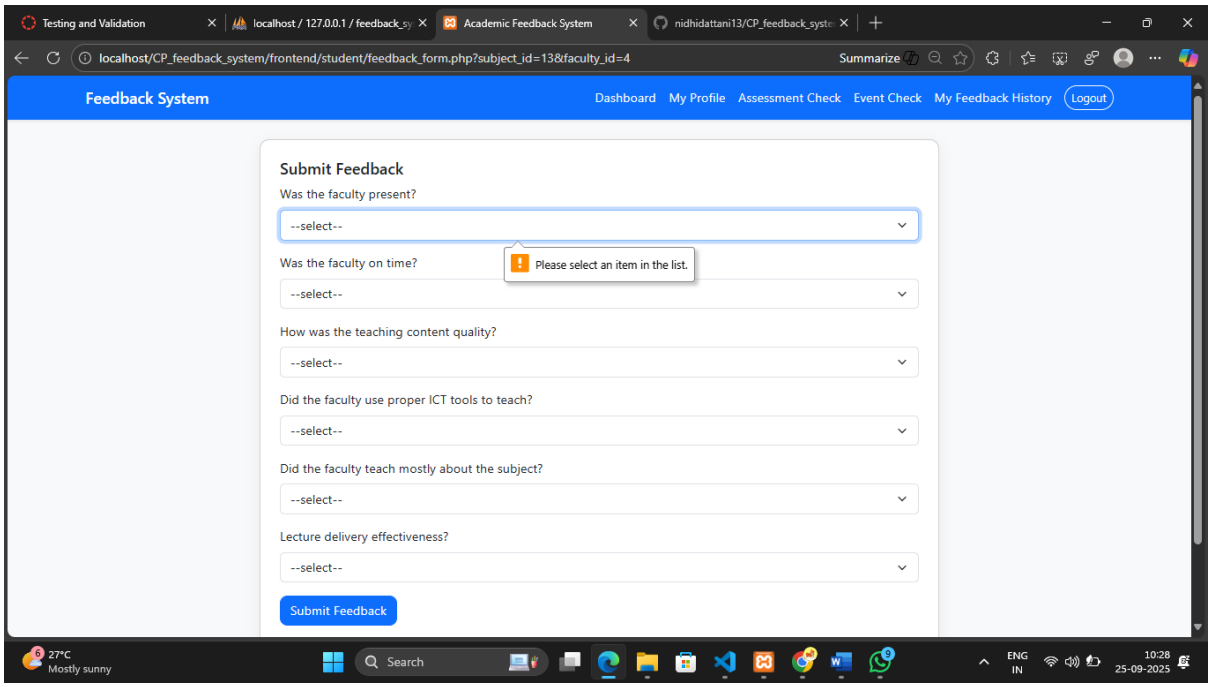
Testing Methodology

My testing strategy will be thorough and follow a two-step process: unit testing and integration testing. I will primarily use manual testing and browser developer tools for front-end validation, and for the back-end, I will use custom PHP scripts to test functions.

Unit Tests

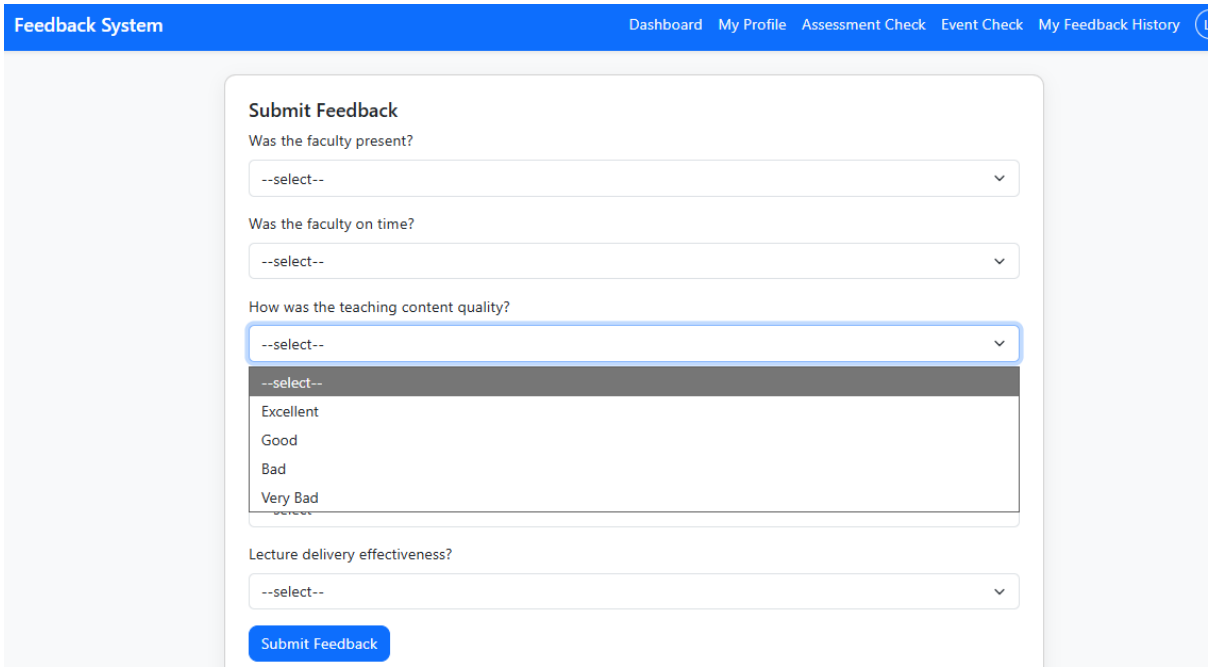
Unit tests will focus on checking the smallest, individual parts of the system. This ensures that each component works on its own before being connected to others.

- **Test Case 1: Student Feedback Form Validation**
    - **Input:** An empty submission from a student.



    - **Expected Output:** The system should show an error message and refuse to submit the form, preventing incomplete data from being saved.
    - **Actual Result:** I will manually verify that the form's required fields prevent submission when empty.
- **Test Case 2: Data Sanitization**

    - **Input:** A string containing malicious code (e.g., `<script>alert('XSS');</script>`) submitted through a feedback field.
    - **Expected Output:** The malicious code should be sanitized or stripped away before being saved in the database.
    - **Actual Result:** I will check the database to ensure the data is stored as plain text, proving that the system is protected from SQL injection and XSS attacks.
- **Test Case 3: Average Rating Calculation**
    - **Input:** A set of test feedback data for a faculty member (e.g., ratings of 5, 4, and 3).

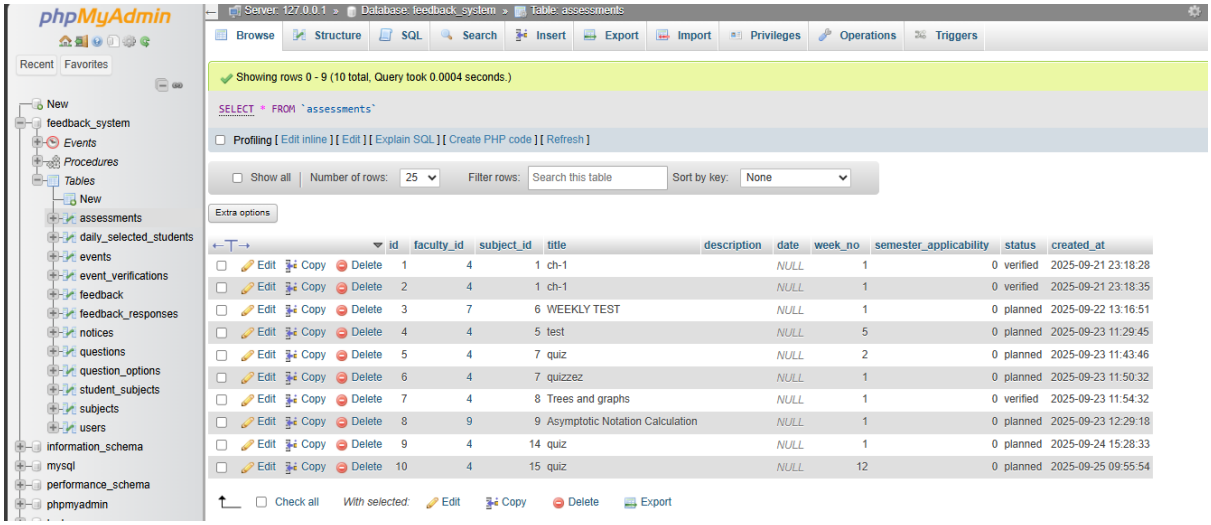


    - **Expected Output:** The PHP function that calculates the average rating should correctly return the value of 4.
    - **Actual Result:** I will use a test script to call the calculation function with the given data and verify that the output is as expected.

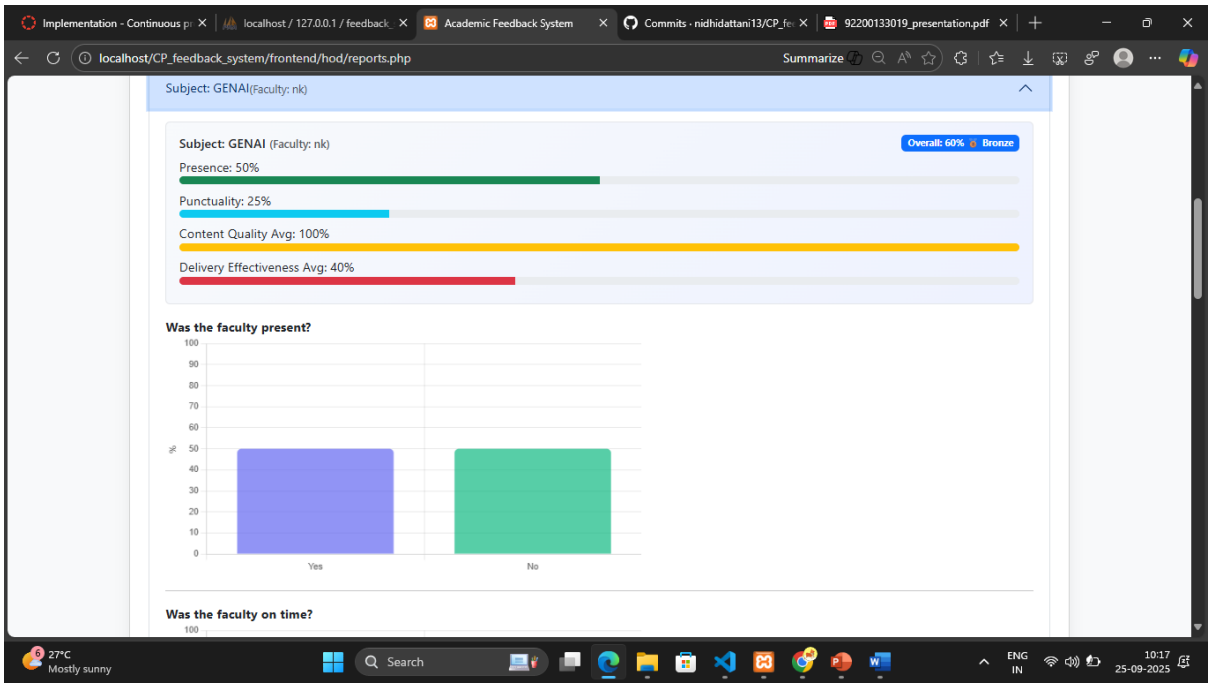
Integration Tests

Integration tests will ensure that the different modules of the system—the front-end, back-end, and database work together without any problems.

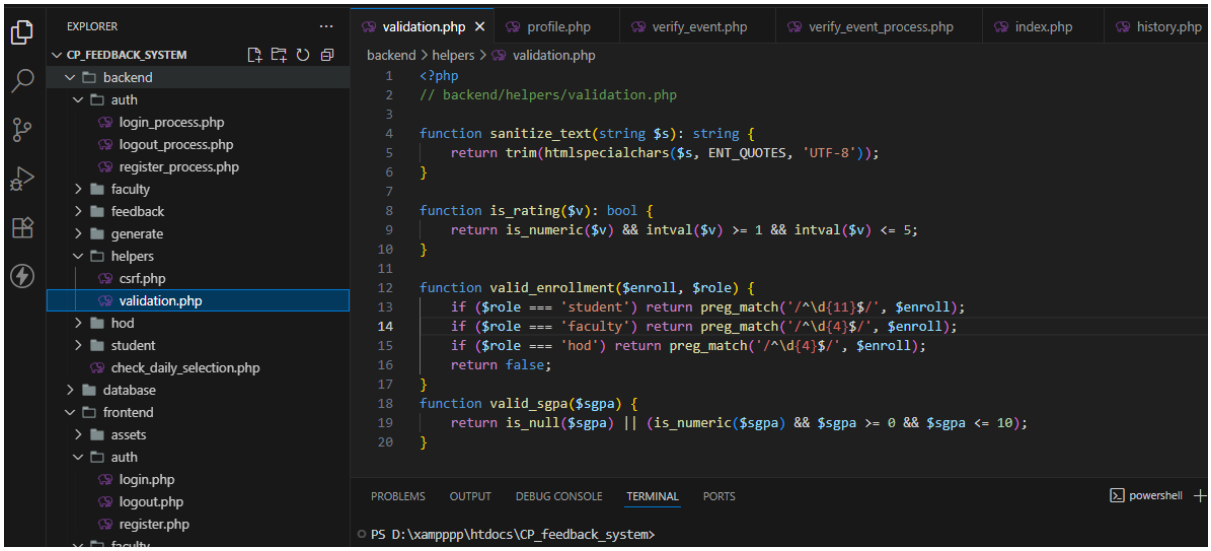
- **Test Case 1: End-to-End Feedback Submission**
    - **Inputs:** A complete and valid feedback form submitted by a student.
    - **Expected Output:** The data should be successfully stored in the MySQL database, and the HOD dashboard should immediately reflect the new data.



  - **Actual Result:** I will manually submit a form and then log in to the HOD dashboard to confirm that the new feedback has been accurately added and the charts have been updated.
- **Test Case 2: HOD Dashboard Data Display**
  - **Inputs:** An HOD logs into the dashboard.
  - **Expected Output:** All charts and graphs should load correctly and display the most current data from the database without any errors.
  - **Actual Result:** I will log in to the HOD dashboard and check that all visualizations are functional and show the correct, live data.



- **Test Case 3: Secure Login**
  - **Inputs:** An incorrect username or password for the HOD.



- **Expected Output:** The system should deny access and display a clear error message without revealing any sensitive information.
- **Actual Result:** I will attempt to log in with incorrect credentials and verify that the system behaves as expected.

Performance Metrics

I will measure the system's performance to ensure it provides a fast and reliable user experience.

- **Response Time:** I will measure the time it takes for a user's request to be completed (e.g., how long it takes for a form to submit or a dashboard to load). I will use browser developer tools for this.
- **Load Testing:** I will simulate multiple concurrent users to see how the system performs under stress. This will help identify any potential bottlenecks.

Validation Against Objectives

Finally, I will validate the project's success by linking my test results back to my original objectives.

- **Objective 1 (Develop a Secure Web Application):** This will be validated through my unit tests on data sanitization and integration tests for secure login. The test results will prove that the system is secure and protects against common vulnerabilities.
- **Objective 2 (Implement a Data Analytics Dashboard):** The HOD Dashboard Integration Test and Performance Metrics will show that the dashboard is functional, reliable, and provides clear, visual insights as intended.
- **Objective 3 (Ensure Anonymity and Data Privacy):** The unit test for data sanitization will provide evidence that no personally identifiable information is stored with the feedback, proving that anonymity is maintained.

Conclusion

Through this rigorous testing and validation process, I am confident that I can deliver a project that is not only well-designed and implemented but also fully functional and reliable. The documented test cases and performance metrics will provide concrete evidence that the **Academic feedback System** successfully solves the problem and meets all project objectives.