Mylene L. Paderna
BSIT-3E

Task 1:

1. Verify if login accepts valid credentials.
2. Verify if login declines invalid credentials.
3. Declines empty fields.
4. Test short password rejection.
5. Account lock after 3 attempts.

Task 2:

Verify if login accepts valid credentials.

Step	Test Case 1	Test Case 2	Test Case 3
Identify Inputs	Username: mylenelargo7@gmail.com Password: **********	Username: mylenelargo7@gmail.com Password: **********	Username: mylenelargo7@gmail.com Password: **********
Expected Output	User is successfully logged in and redirected to the dashboard/homepage.	User is successfully logged in and redirected to the dashboard/homepage.	User is successfully logged in and redirected to the dashboard/homepage.
Actual Output	User logged in and redirected correctly.	User logged in and redirected correctly.	User logged in and redirected correctly.
Comparison (Pass/Fail)	Pass	Pass	Pass

Task 3:

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status
TC001	**Valid login**	1. Open login page 2. Enter valid username 3. Enter valid password 4. Click Login	Username: mylenelargo7@gmail.com Password: *********	User successfully logs in and is redirected to the dashboard.	As expected	Pass
TC002	**Invalid login**	1. Open login page 2. Enter invalid username or password 3. Click Login	Username: wrongUser Password: WrongPass!	System displays an error message: "Invalid login credentials."	As expected	Pass
TC003	**Empty fields**	1. Open login page 2. Leave both fields empty 3. Click Login	Username: Password:	System displays validation messages: "Username is required" and "Password is required."	System displayed: "Please Enter your ID(or Email) and password"	Pass
TC004	**Short password**	1. Open login page 2. Enter valid username 3. Enter a password shorter than required length 4. Click Login	Username: mylenelargo7@gmail.com Password: largo1234	System shows error: "Password must be at least 8 characters."	System displayed an error message: "Invalid login credentials."	Pass
TC005	**Account lock after 3 attempts**	1. Open login page 2. Enter invalid credentials and click login	Username: mylenelargo7@gmail.com Password: wrongPass	After 3 failed attempts, system locks the account.	The system did not lock after 10 attempts	Fail

		(repeat 3 times)				

Task 4:

TC001 – Valid Login

Technique Used: System Testing

Justification: The valid login test checks the entire login functionality working together (UI → backend → authentication logic → redirect). This evaluates the behavior of the whole system as a complete unit, not just a single component.

TC002 – Invalid Login

Technique Used: Sanity Testing

Justification: This test quickly verifies that a basic and essential function (login validation) behaves correctly after builds or changes. It's a small, focused test on core functionality.

TC003 – Empty Fields

Technique Used: Integration Testing

Justification: This test verifies how different modules interact: UI input fields Validation module Error message component It checks the flow between modules, making it an integration test.

TC004 – Short Password

Technique Used: Smoke Testing

Justification: Smoke testing checks whether critical functionalities work before deeper testing. Password validation is one of the core login checks, so confirming that it works is part of smoke test readiness.

TC005 – Account Lock After 3 Failed Attempts

Technique Used: Regression Testing

Justification: This scenario ensures that after multiple updates to login or authentication, the existing feature—account lockout—still functions. It validates that changes did not break earlier implemented behavior.

Task 5:

Functional Tests

1. **Valid Login Test**
   Ensures the system accepts correct username and password and redirects the user to the dashboard.
2. **Invalid Login Test**
   Checks that the system correctly rejects wrong credentials and displays an error message.
3. **Empty Field Validation Test**
   Confirms that the login button is disabled or error messages appear when username/password are empty.

Non-Functional Tests

1. **Performance Test (Response Time)**
   Measures how fast the login page loads and how quickly the system processes login requests under normal load.
2. **Security Test (Brute-Force Protection)**
   Ensures the login page resists attacks such as repeated login attempts, injection attacks, or credential stuffing.
3. **Usability Test (User Experience)**
   Evaluates whether the login page is easy to understand, has clear instructions, readable labels, and intuitive design.

Task 6:

1. **How does functional testing improve software quality in the login module?**

Functional testing improves the quality of the login module in several ways:

- **Ensures correct behavior of core features** -bIt verifies that essential login functions—such as validating credentials, handling empty fields, and redirecting users—work exactly as intended.
- **Detects defects early** - Errors in authentication logic, field validation, session handling, or error messages can be found before the system is released.
- **Validates user requirements** - Since functional testing uses the actual requirements (e.g., correct password rules, lockout rules), it ensures the login system meets business expectations.

- **Improves reliability and stability -** By testing different input scenarios (valid, invalid, edge cases), the login module becomes more stable and less likely to fail during real-world use.
- **Enhances user experience** - Proper functional testing ensures users receive meaningful feedback (e.g., “Incorrect password”), reducing confusion and frustration.

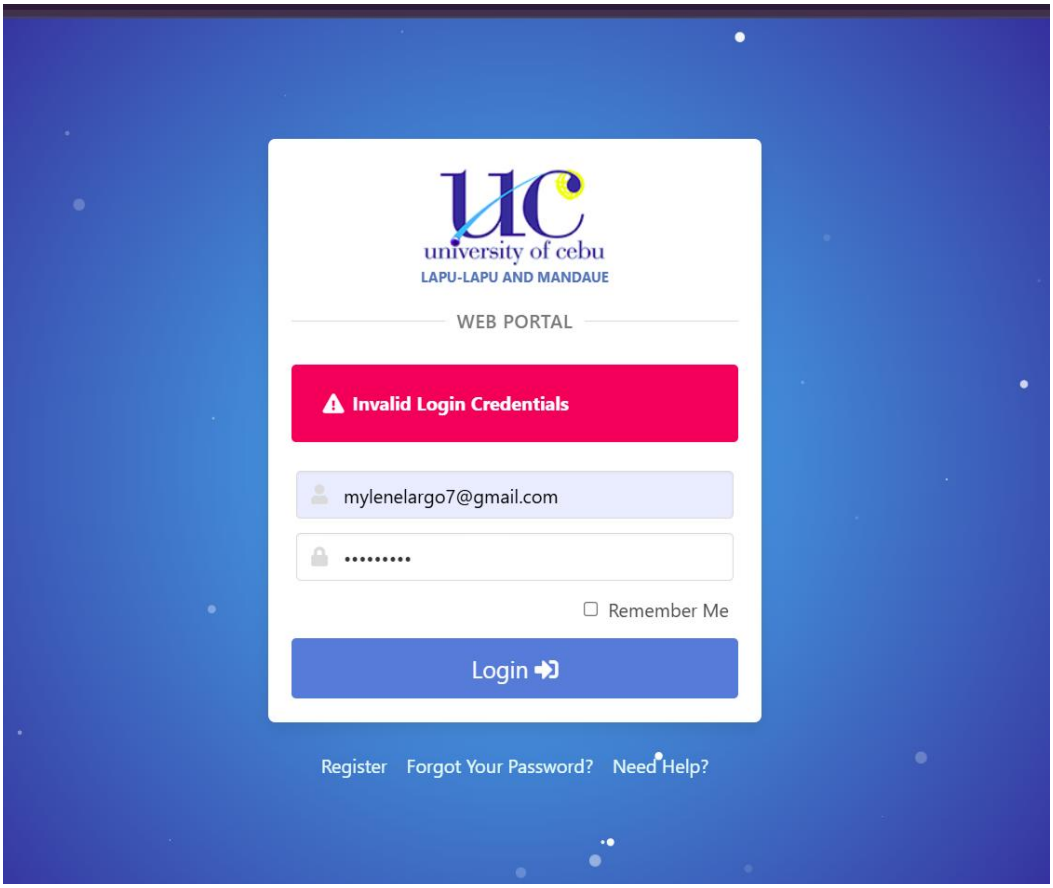
2. **What limitations may affect the accuracy of your test results?**

Several limitations of functional testing can impact how accurate or complete your test results are:

- **Cannot guarantee full coverage** - Functional testing typically checks expected behaviors, but it cannot cover every possible user action or input combination.
- **Limited to documented requirements -** If a requirement is missing, unclear, or incorrect, functional testing may miss defects related to those gaps.
- **Does not test performance or usability** - Functional testing focuses on what the system does, not how well it performs. Slow response times or poor user experience might still go undetected.
- **Dependent on test data quality -** If the test data is incomplete or unrealistic, the test results may not reflect real user scenarios accurately.
- **May overlook integration or environment issues** - Functional tests often assume the environment is stable; real-world issues (network delays, server load, API failures) may not be captured.

Task 7:

Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status
Invalid login	1. Open login page 2. Enter invalid username or password 3. Click Login	Username: wrongUser Password: WrongPass!	System displays an error message: “Invalid login credentials.”	As expected	Pass



Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status
Empty fields	1. Open login page 2. Leave both fields empty 3. Click Login	Username: Password:	System displays validation messages: "Username is required" and "Password is required."	System displayed: "Please Enter your ID(or Email) and password"	Pass