# Login Page Test Plan

## Objective

The objective of this test plan is to verify the functionality, usability, security, and performance of the login page.

## Scope

The scope of this test plan includes testing the following aspects of the login page:

* Authentication and authorization
* User interface and usability
* Security and protection against common attacks
* Compatibility with different devices and browsers

## Deliverables

* Test cases, test data, and results documentation.
* A summary report detailing the pass/fail status and any defects identified.
* Recommendations for improvements and mitigations.

## Strategies

Testing will be performed manually. A combination of functional, security, usability, and compatibility testing will be conducted.

## Test Cases

The following test cases will be executed during the testing process:

### Functional Testing

* Verify that valid user credentials allow access to the application.
* Verify that invalid credentials result in an error message.
* Test the "Remember Me" feature.
* Test the "Forgot Password" functionality.
* Check for session timeout behavior.

### Security Testing

* Test for SQL injection attacks.
* Test for Cross-Site Scripting (XSS) vulnerabilities.
* Verify that the application mitigates against Cross-Site Request Forgery (CSRF) attacks.
* Check for brute force attack prevention.
* Test for account lockout functionality.

### Usability Testing

* Evaluate the login page for accessibility compliance.
* Test the page's responsiveness on different devices.
* Ensure the page is user-friendly and easy to understand.
* Check for browser autofill compatibility.

### Compatibility Testing

* Test the login page on different browsers (e.g., Chrome, Firefox, Safari, Edge).
* Verify compatibility on various devices (desktop, mobile, tablet).
* Test the page in different screen resolutions.

## Test Data

Test data will include both valid and invalid usernames and passwords. Additional test data may be used to validate the "Forgot Password" feature.

Test cases for the login page:

1. Verify that a user with the correct credentials can log in successfully.

- Input: Valid username and password.

- Expected Output: The user is logged in and directed to the homepage.

2. Ensure the system doesn't allow empty fields.

- Input: Empty username and password fields.

- Expected Output: An error message pops, such as "Both fields are required."

3. Test if the system handles an invalid username.

- Input: Valid password and an invalid username.

- Expected Output: Display an error message, e.g., "Invalid username."

4. Test if the system handles an invalid password.

- Input: Valid username and an incorrect password.

- Expected Output: Display an error message, e.g., "Invalid password."

5. Clicking on "Forgot Password" should navigate to the password reset page.

- Input: Click on the "Forgot Password" link.

- Expected Output: Redirect to the password reset page.

6. Verify that the "Remember Me" checkbox keeps the user logged in after the session ends.

- Input: Check the "Remember Me" box and log in.

- Expected Output: The user remains logged in even after the browser is closed.

7. After a session timeout, the user should be prompted to log in again.

- Input: Wait for the session timeout period to expire.

- Expected Output: After a timeout, redirect the user to the login page.

8. Test if the system supports multiple sessions for the same user.

- Input: Login from two different devices simultaneously.

- Expected Output: Both sessions are valid.

9. Ensure that the system enforces strong password requirements.

- Input: Enter a weak password (e.g., "password123").

- Expected Output: Display a message that indicates the password is weak and specify the requirements.

10. After a specified number of failed login attempts, lock the user account.

- Input: Exceed the maximum number of failed login attempts.

- Expected Output: Display a message like "Account locked due to too many failed attempts."

11. Test the login page on different browsers (e.g., Chrome, Firefox, Safari, Edge).

- Input: Use various browsers to access the login page.

- Expected Output: Ensure consistent functionality and appearance.

12. Verify that the system prevents or delays brute force attacks.

- Input: Repeated login attempts with incorrect credentials.

- Expected Output: The system should either introduce a delay between login attempts or lock the account after a certain number of failures.

13. Test the "Forgot Password" functionality.

- Input: Click on "Forgot Password" and follow the password recovery process.

- Expected Output: Receive a password reset link or code via email and successfully reset the password.

14. If your application supports social login (e.g., via Google or Facebook), ensure it works correctly.

- Input: Click on a social login button (e.g., "Login with Google").

- Expected Output: Redirect to the respective social media for authentication and return to your app upon success.

15. Test how the login page behaves with browser autofill.

- Input: Use a browser with autofill enabled to populate the login fields.

- Expected Output: Ensure that autofill works as expected, filling in the username and password correctly.

16. Verify that the session times out after a period of inactivity.

- Input: Login, then leave the application inactive for the specified timeout duration.

- Expected Output: After timeout, redirect the user to the login page.

17. Accessibility Testing: Ensure that the login page is accessible to users with disabilities.

- Input: Use accessibility tools (e.g., screen readers) to navigate and interact with the login page.

- Expected Output: Verify that all page elements are accessible and that users can log in using accessibility tools.