



TEST TASK: DOCUMENTATION AND TEST SCENARIO IDENTIFICATION



JULY 14, 2024

Test Plan Outline

Test Plan Identifier

TP01

Introduction:

The purpose of this test plan is to ensure the quality and functionality of the online registration form. As a Quality Assurance Engineer, my responsibilities include validating user inputs, verifying error handling, and confirming successful data submission. The registration form collects essential information such as full names, email addresses, passwords, dates of birth, and gender. By meticulously examining each aspect of the form, we aim to enhance the overall user experience and contribute to a seamless registration process. Our testing efforts will focus on both positive and negative scenarios, ensuring that the form validates inputs, displays appropriate error messages, and successfully submits data to the server upon correct completion.

Objectives

- Ensure that users can input valid data for all required fields (e.g., Full Name, Email, Password).
- Verify that the form handles special characters, spaces, and other edge cases appropriately.
- Confirm that the form displays relevant error messages when users provide invalid or incomplete information.
- Validate error messages for each field (e.g., "Please enter a valid email address," "Password must be at least 8 characters," etc.).
- Test the form's submission process to ensure that user data is correctly sent to the server.
- Verify that users receive confirmation upon successful submission.

Scope

In scope: Form fields, validation rules, and error messages.

Out of scope: Server-side validation.

Environments

Browsers: Ensure compatibility with popular browsers such as Chrome, Firefox, and Edge.

Devices: Test on both desktop and mobile devices.

Operating Systems: Validate functionality across Windows, macOS, Android, and iOS.

Testing tools: Use different automated testing tools.

Test Data

Valid Data:

- Sample names, valid email addresses, and strong passwords.
- Include edge cases (e.g., special characters, long names).

Invalid Data:

- Empty fields, invalid email formats, and short passwords.
- When the user provides incorrect information.
- When a user enters different data in contrast to what is required.

Test Scenarios

1. The User enters a password that doesn't meet the minimum length requirement (less than 8 characters).
2. The User enters mismatched passwords in the "Password" and "Confirm Password" fields.
3. The user enters an invalid date of birth (e.g., future date or incorrect format).
4. The user submits the form without selecting a gender.
5. The user submits the form with empty or blank fields individually.
6. The user submits the response with all fields empty.
7. The user enters the invalid email
8. The user enters the email with the wrong format
9. Verify that the registration page properly handles passwords with spaces, special characters, or non-alphanumeric characters.
10. Verify that the registration page detects and prevents the use of commonly used or easily guessable passwords
11. The user did not select any of the options of the newsletter.
12. The Registration page detects and prevents the use of commonly used or easily guessable passwords.
13. Testing the Response when traffic is high.
14. Verify the error messages are clear and related to the incorrect field.
15. Test to prevent accounts from using the duplicate email address.

Test Execution Schedule

Start Date:

End Date:

Allocate time for each test scenario based on priority and complexity.
(We can also use a Gantt chart to display the schedule)

Risks and Assumptions

Risks:

- Incomplete validation logic may lead to unexpected behavior.
- Different OS problems.

- Browser compatibility issues could affect user experience.

Assumptions:

- The testing environment remains stable during the testing period.

Test Cases

Test Case ID: TC001 - Valid Password Length
Description: Verify that the registration form correctly validates the password length.
Preconditions: The registration form is open, and the user is on the password input field.
Test Steps:
<ul style="list-style-type: none">• Enter a password with less than 8 characters.• Click the submit button.
Expected Results:
An error message should appear, indicating that the password must be at least 8 characters long.
Post-conditions: The user remains on the registration form.

Table 1 TC001

Test Case ID: TC002 - Mismatched Passwords
Description: Confirm that the form detects mismatched passwords.
Preconditions: The registration form is open, and the user has entered a valid password.
Test Steps:
<ul style="list-style-type: none">• Enter a valid password in the "Password" field.• Enter a different password in the "Confirm Password" field.• Click the submit button.
Expected Results:
An error message should appear, indicating that the passwords do not match.
Post-conditions: The user remains on the registration form.

Table 2 TC002

Test Case ID: TC003 - Invalid Date of Birth
Description: Validate the handling of incorrect date formats.
Preconditions: The registration form is open, and the user is on the date of birth input field.
Test Steps:
<ul style="list-style-type: none">• Enter an invalid date (e.g., "30-Feb-2000" or "2022-13-01").• Click the submit button.

Expected Results:
An error message should appear, indicating that the date format is incorrect.
Post-conditions: The user remains on the registration form.

Table 3 TC003

Scenario Identification

Positive Scenarios:

Successful Registration:

Steps:

1. The user opens the registration form.
2. Enters valid information (full name, email, password, date of birth, gender).
3. Click the submit button.

Expected Outcome:

- The form validates the data, displays a success message (e.g., "Registration successful!"), and redirects the user to the next page.

Valid Password and Confirmation:

Steps:

1. The user enters a strong password (e.g., "P@ssw0rd").
2. Confirm the same password in the "Confirm Password" field.
3. Submit the form.

Expected Outcome:

- The form accepts the password, validates the confirmation, and proceeds to the next step.

Negative Scenarios:

Invalid Email Format:

Steps:

1. The user enters an invalid email address e.g., without the @ symbol
2. Submit the form.

Expected Outcome:

- The form displays an error message (e.g., "Please enter a valid email address") and highlights the email field.

Empty Required Fields:

Steps:

1. The user leaves the "Full Name" and "Email" fields empty.
2. Submit the form.

Expected Outcome:

- The form shows error messages for the empty fields (e.g., "Full Name is required," "Email is required").

Test Script

Positive Scenario

```
describe('Registration Form', () => {  
  it('should successfully register Ayesha', () => {  
    cy.visit('/registration');  
  
    // Fill out the form fields  
    cy.get('[data-testid="full-name"]').type('Ayesha');  
    cy.get('[data-testid="email"]').type('ayeshariaz6306@gmail.com');  
    cy.get('[data-testid="password"]').type('P@ssw0rd');  
    cy.get('[data-testid="confirm-password"]').type('P@ssw0rd');  
    cy.get('[data-testid="date-of-birth"]').type('2004-08-01');  
    cy.get('[data-testid="gender"]').select('Female');  
  
    // Submit the form  
    cy.get('[data-testid="submit-button"]').click();  
  
    // success message  
    cy.url().should('include', '/success');  
    cy.contains('Registration successful!').should('be.visible');  
  });  
});
```

Negative Scenario

```
describe('Registration Form', () => {  
  it('should display an error for an invalid password length', () => {  
    cy.visit('/registration'); // Replace with your registration page URL  
  
    // Fill out the form fields  
    cy.get('[data-testid="full-name"]').type('Invalid User');  
    cy.get('[data-testid="email"]').type('invalid@example.com');  
    cy.get('[data-testid="password"]').type('short'); // Invalid password (less than 8  
characters)  
    cy.get('[data-testid="confirm-password"]').type('short');  
    cy.get('[data-testid="date-of-birth"]').type('2000-01-01');  
    cy.get('[data-testid="gender"]').select('Other');  
  
    // Submit the form  
    cy.get('[data-testid="submit-button"]').click();
```



```
// Error message for password
cy.contains('Password must be at least 8 characters').should('be.visible');
});
});
```