# **Test Plan for Registration Form**

## **Introduction**

**1.1 Purpose**

This document outlines the strategy and approach for testing the online registration form on the website. The primary goal is to ensure the form functions correctly, validates inputs, displays appropriate error messages, and submits data successfully.

**1.2 Background**

The registration form is a critical component of the website, enabling new users to create accounts and access the services provided. Ensuring its reliability and usability is essential for user satisfaction and retention.

1. Objectives

**2.1 Functional Verification**

Ensure all form fields accept valid inputs and reject invalid inputs.

**2.2 Error Handling**

Verify that error messages are displayed appropriately for invalid inputs.

**2.3 Data Submission**

Confirm successful data submission upon correct completion of the form.

**2.4 Security**

Ensure the password is stored securely and meets security guidelines.

**2.5 Usability**

Verify that the form is user-friendly and accessible.

1. Scope

**3.1 Functional Testing**

Ensure each form field works correctly and validates inputs appropriately.

**3.2 Input Validation**

Test the form's ability to handle both valid and invalid inputs across all fields.

**3.3 Error Messaging**

Verify that appropriate error messages are displayed for various validation failures.

**3.4 Data Submission**

Test the form’s ability to successfully submit data to the server.

**3.5 Cross-Browser Compatibility**

Ensure the form works across different browsers.

**3.6 Responsiveness**

Ensure the form is usable on different devices (desktop, tablet, mobile).

1. Test Environment

**4.1 Browsers**

* Latest versions of Chrome, Firefox, Safari, and Edge.

**4.2 Operating Systems**

* Windows 10
* macOS
* Linux

**4.3 Devices**

* Desktop
* Tablet
* Mobile

**4.4 Test Tools**

* Cypress for automated testing
* Chrome DevTools for debugging

1. Test Data

**5.1 Full Name**

* **Valid**: "John Doe", "Jane Smith".
* **Invalid**: "12345", "<script>alert('XSS')</script>".

**5.2 Email Address**

* **Valid**: "john.doe@example.com", "jane.smith@domain.org".
* **Invalid**: "john.doe@", "jane.smith@domain", "user@@domain.com".

**5.3 Password**

* **Valid**: "Password123", "Secure!Pass987".
* **Invalid**: "short", "1234567", "password".

**5.4 Confirm Password**

* Matching and non-matching cases.

**5.5 Date of Birth**

* Various valid dates and edge cases (e.g., 29/02/2000).

**5.6 Gender**

* Selection of "Male", "Female", "Other".

**5.7 Newsletter Subscription**

* Choices of "Yes" and "No".

1. Test Scenarios

**6.1 Positive Scenarios**

* Successful registration with valid data.
* Successful registration with newsletter subscription.

**6.2 Negative Scenarios**

* Registration with missing required fields.
* Registration with invalid email format.
* Password and confirm password mismatch.
* Password less than 8 characters.

1. Test Execution Schedule

**7.1 Preparation**

* **Duration**: 1 day
* **Activities**: Setting up the test environment, preparing test data.

**7.2 Test Execution**

* **Duration**: 3 days
* **Activities**: Executing test cases, logging defects.

**7.3 Defect Retesting**

* **Duration**: 2 days
* **Activities**: Retesting defects, verifying fixes.

**7.4 Reporting**

* **Duration**: 1 day
* **Activities**: Preparing and distributing the final test report.

1. Risks and Assumptions

**8.1 Risks**

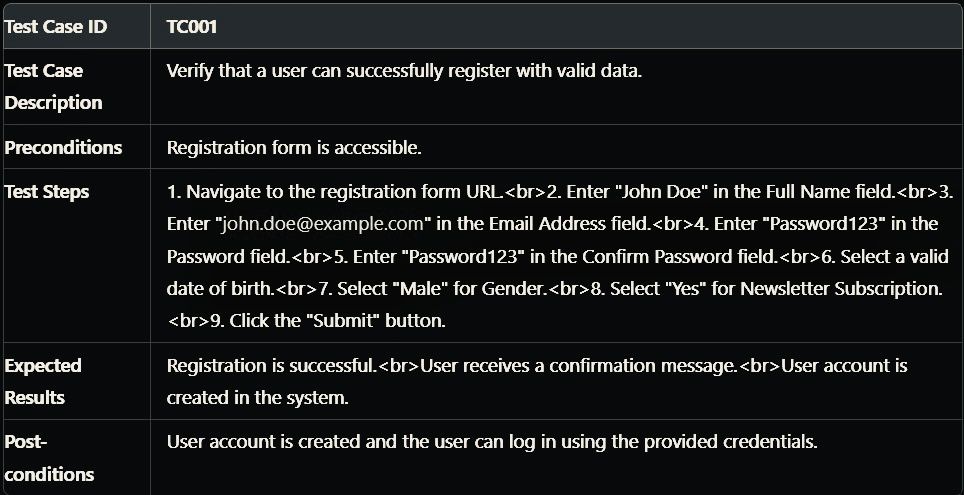
* Potential browser-specific issues.
* Possible downtime of the server during testing.

**8.2 Assumptions**

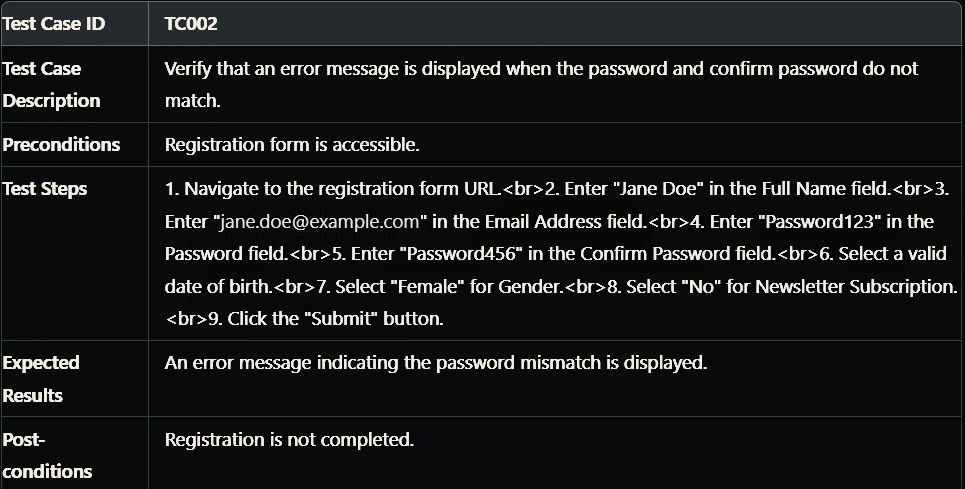
* The server is available for testing.
* Test data is available and can be reused.

1. Test Cases

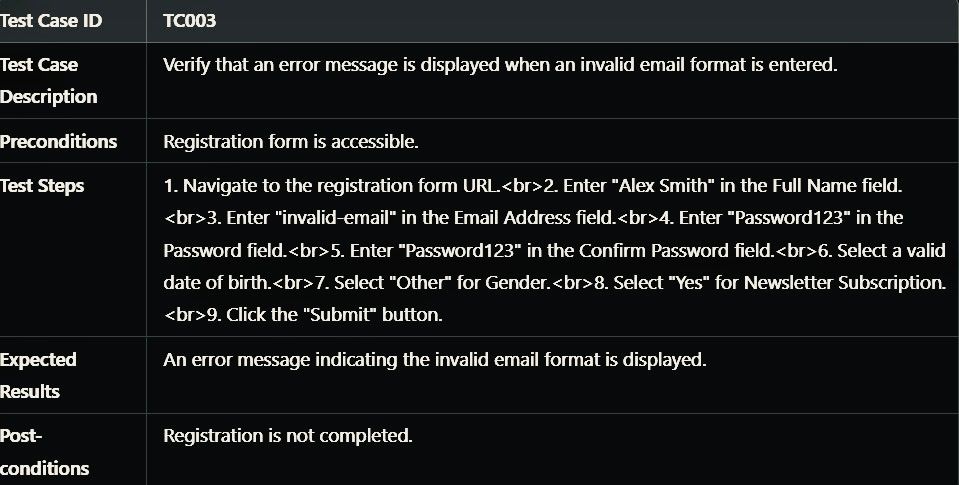
**Test Case 1: Successful Registration**

****

**Test Case 2: Password Mismatch**

****

**Test Case 3: Invalid Email FormatScenario Identification**

****

**10.1 Positive Scenarios**

**10.1.1 Successful Registration with Valid Data**

* **Steps**:
  1. Enter all valid data (full name, email, password, confirm password, date of birth, gender, newsletter subscription).
  2. Submit the form.
* **Expected Outcome**: User is successfully registered and receives a confirmation message.

**10.1.2 Successful Registration with Newsletter Subscription**

* **Steps**:
  1. Enter all valid data including selecting "Yes" for the newsletter subscription.
  2. Submit the form.
* **Expected Outcome**: User is successfully registered, receives a confirmation message, and is subscribed to the newsletter.

**10.2 Negative Scenarios**

**10.2.1 Registration with Missing Required Fields**

* **Steps**:
  1. Leave one or more required fields empty.
  2. Submit the form.
* **Expected Outcome**: Error messages are displayed for the missing required fields.

**10.2.2 Registration with Invalid Email Format**

* **Steps**:
  1. Enter an invalid email format.
  2. Submit the form.
* **Expected Outcome**: An error message indicating the invalid email format is displayed.

1. Test Writing

**11.1 Automated Test Scripts**

javascript

Copy code

describe('Registration Form Tests', () => {

  it('Should successfully register with valid data', () => {

    cy.visit('https://example.com/register');

    cy.get('input[name="fullName"]').type('John Doe');

    cy.get('input[name="email"]').type('john.doe@example.com');

    cy.get('input[name="password"]').type('Password123');

    cy.get('input[name="confirmPassword"]').type('Password123');

    cy.get('input[name="dob"]').type('1990-01-01');

    cy.get('input[name="gender"]').check('Male');

    cy.get('input[name="newsletter"]').check('Yes');

    cy.get('button[type="submit"]').click();

    cy.contains('Registration successful').should('be.visible');

  });

  it('Should display error for password mismatch', () => {

    cy.visit('https://example.com/register');

    cy.get('input[name="fullName"]').type('Jane Doe');

    cy.get('input[name="email"]').type('jane.doe@example.com');

    cy.get('input[name="password"]').type('Password123');

    cy.get('input[name="confirmPassword"]').type('Password456');

    cy.get('input[name="dob"]').type('1990-01-01');

    cy.get('input[name="gender"]').check('Female');

    cy.get('input[name="newsletter"]').check('No');

    cy.get('button[type="submit"]').click();

    cy.contains('Passwords do not match').should('be.visible');

  });

  it('Should display error for invalid email format', () => {

    cy.visit('https://example.com/register');

    cy.get('input[name="fullName"]').type('Alex Smith');

    cy.get('input[name="email"]').type('invalid-email');

    cy.get('input[name="password"]').type('Password123');

    cy.get('input[name="confirmPassword"]').type('Password123');

    cy.get('input[name="dob"]').type('1990-01-01');

    cy.get('input[name="gender"]').check('Other');

    cy.get('input[name="newsletter"]').check('Yes');

    cy.get('button[type="submit"]').click();

    cy.contains('Invalid email format').should('be.visible');

  });

});