

Documentation

Test Plan Outline for Online Registration Form

1. Introduction

This plan explains how we will test the online registration form. The form collects user information and should check that all inputs are correct before submission.

2. Objectives

- Make sure the form works correctly.
- Check that all fields accept valid data and show error messages for invalid data.
- Confirm that the form submits data to the server when all inputs are correct.

3. Scope

- We will test the functionality, usability, and performance of the form.
- The form fields include: Full Name, Email Address, Password, Confirm Password, Date of Birth, Gender, Newsletter Subscription, and Submit Button.

4. Test Environment

- Browsers: Chrome, Firefox, Safari, Edge
- Devices: Desktop, Mobile, Tablet
- Operating Systems: Windows, macOS, Android, iOS

5. Test Data

Example data for each field:

Full Name: "hamza"

Email Address: "hamza@example.com"

Password: "password123"

Confirm Password: "password123"

Date of Birth: "01/01/1990"

Gender: "Male"

Newsletter Subscription: "Yes"

6. Test Scenarios

Positive and negative test scenarios for checking form validation and submission.

7. Test Execution Schedule

Start Date: 21-09-2009

End Date: 21-01-2010

Daily test execution from 9 AM to 5 PM.

8. Risks and Assumptions

Risks: Browser compatibility issues, server downtime.

Assumptions: Stable internet connection, accurate test data.

Test Cases

Test Case 1: Valid Registration

Test Case ID: TC001

Test Case Description: Test the registration form with valid inputs.

Preconditions: Browser is open, user is on the registration page.

Test Steps:

1. Enter "sami" in the Full Name field.
2. Enter "sami@example.com" in the Email Address field.
3. Enter "password123" in the Password field.
4. Enter "password123" in the Confirm Password field.
5. Enter "01/01/1990" in the Date of Birth field.
6. Select "Male" for Gender.
7. Select "Yes" for Newsletter Subscription.
8. Click the Submit button.

Expected Results: Form submits successfully, user is redirected to a confirmation page.

Postconditions: User receives a confirmation email.

Test Case 2: Password Mismatch

Test Case ID: TC002

Test Case Description: Test the registration form with mismatched passwords.

Preconditions: Browser is open, user is on the registration page.

Test Steps:

1. Enter "Jami" in the Full Name field.
2. Enter "jami@example.com" in the Email Address field.
3. Enter "password123" in the Password field.
4. Enter "password321" in the Confirm Password field.
5. Enter "01/01/1990" in the Date of Birth field.
6. Select "Female" for Gender.
7. Select "No" for Newsletter Subscription.

8. Click the Submit button.

Expected Results: Error message "Passwords do not match" is displayed.

Postconditions: Form is not submitted.

Test Case 3: Invalid Email Format

Test Case ID: TC003

Test Case Description: Test the registration form with an invalid email format.

Preconditions: Browser is open, user is on registration page.

Test Steps:

1. Enter "Ali" in the Full Name field.
2. Enter "ali.9@invalid" in the Email Address field.
3. Enter "password123" in the Password field.
4. Enter "password123" in the Confirm Password field.
5. Enter "01/01/1990" in the Date of Birth field.
6. Select "Other" for Gender.
7. Select "Yes" for Newsletter Subscription.
8. Click the Submit button.

Expected Results: Error message "Invalid email address" is displayed.

Postconditions: Form is not submitted.

Scenario Identification

Positive Scenarios:

1. Valid Registration

Steps: Enter valid data for all fields and click Submit.

Expected Outcome: Form submits successfully, user is redirected to a confirmation page.

2. Newsletter Subscription

Steps: Enter valid data, select "Yes" for Newsletter Subscription, and click Submit.

Expected Outcome: Form submits successfully, user receives a confirmation email with subscription details.

Negative Scenarios:

1. Password Mismatch

Steps: Enter valid data except for mismatched passwords, and click Submit.

Expected Outcome: Error message "Passwords do not match" is displayed, form is not submitted.

2. Invalid Email Format

Steps: Enter valid data except for an invalid email format, and click Submit.

Expected Outcome: Error message "Invalid email address" is displayed, form is not submitted.

Test Writing

Automated Test Scripts for Positive Scenarios

Using Cypress (JavaScript):

```
``javascript
describe('Registration Form', () => {
  it('should submit the form with valid inputs', () => {
    cy.visit('/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('19900101');
    cy.get('select[name="gender"]').select('Male');
    cy.get('input[name="newsletter"]').check();
    cy.get('button[type="submit"]').click();
    cy.url().should('include', '/confirmation');
  });

  it('should submit the form with newsletter subscription', () => {
    cy.visit('/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('19900101');
    cy.get('select[name="gender"]').select('Male');
    cy.get('input[name="newsletter"]').check();
    cy.get('button[type="submit"]').click();
    cy.url().should('include', '/confirmation');
    cy.get('.confirmation-email').should('contain', 'Subscription confirmed');
  });
});
```

```
});  
});  
'''
```

Submission

The test plan outline, test cases, scenario identification, and test scripts are provided above. Any additional information or assumptions were included within the documentation sections.