**Test Plan Outline**

**Introduction**

The test plan aims to outline the testing approach for the online registration form, ensuring it collects and validates user data accurately before submission.

**Objectives**

1. Validate the functionality of the registration form fields.
2. Ensure proper error handling and validation messages.
3. Verify successful data submission to the server.
4. Confirm the usability and user experience aspects of the form.

**Scope**

* **Inclusions**: Testing all fields of the registration form (Full Name, Email Address, Password, Confirm Password, Date of Birth, Gender, Newsletter Subscription).
* **Exclusions**: Payment processing, backend database verification beyond successful submission.

**Test Environment**

* **Browser**: Chrome, Firefox.
* **Tools**: Cypress for automated testing, Chrome Developer Tools for manual inspection.
* **Operating Systems**: Windows 10, macOS.

**Test Data**

* **Valid Data**: Properly formatted entries for all fields.
* **Invalid Data**: Incorrect formats for email, passwords that do not match, missing required fields.

**Test Scenarios**

1. Successful registration with all fields correctly filled.
2. Registration with invalid email format.
3. Registration with passwords that do not match.

**Test Execution Schedule**

* **Day 1**: Setup test environment and tooling.
* **Day 2-3**: Write and execute test cases manually.
* **Day 4-5**: Develop and execute automated scripts.
* **Day 6**: Review and finalize test reports.

**Risks and Assumptions**

* **Risks**: Dependency on stable internet connection for testing.
* **Assumptions**: Server-side validation and database handling are functioning correctly.

**Test Cases**

**TC\_001: Successful Registration**

* **Description**: User successfully registers with valid data.
* **Preconditions**: Registration form is accessible.
* **Test Steps**:
  1. Enter valid Full Name, Email Address, Password (at least 8 characters), Confirm Password, Date of Birth, select Gender, and opt for Newsletter Subscription.
  2. Click Submit button.
* **Expected Results**: User is successfully registered.
* **Post-conditions**: User data is stored in the server database.

**TC\_002: Invalid Email Format**

* **Description**: User attempts to register with an invalid email format.
* **Preconditions**: Registration form is accessible.
* **Test Steps**:
  1. Enter valid Full Name, invalid Email Address format, valid Password, Confirm Password, Date of Birth, select Gender, and opt for Newsletter Subscription.
  2. Click Submit button.
* **Expected Results**: Error message displays indicating invalid email format.
* **Post-conditions**: User data is not submitted.

**TC\_003: Passwords Do Not Match**

* **Description**: User attempts to register with passwords that do not match.
* **Preconditions**: Registration form is accessible.
* **Test Steps**:
  1. Enter valid Full Name, Email Address, valid Password, different Confirm Password, Date of Birth, select Gender, and opt for Newsletter Subscription.
  2. Click Submit button.
* **Expected Results**: Error message displays indicating passwords do not match.
* **Post-conditions**: User data is not submitted.

**Scenario Identification**

**Positive Scenarios**

1. **Scenario 1: Successful Registration**
   * **Steps**: User fills all fields correctly and submits.
   * **Expected Outcome**: User is registered successfully.
2. **Scenario 2: Newsletter Subscription**
   * **Steps**: User opts for newsletter subscription during registration.
   * **Expected Outcome**: User data shows newsletter subscription preference.

**Negative Scenarios**

1. **Scenario 1: Invalid Email Format**
   * **Steps**: User enters an email address without '@' symbol.
   * **Expected Outcome**: Error message prompts user to enter a valid email.
2. **Scenario 2: Passwords Do Not Match**
   * **Steps**: User enters two different passwords in the 'Password' and 'Confirm Password' fields.
   * **Expected Outcome**: Error message alerts user that passwords do not match.

**Automated Test Scripts (using Cypress)**

javascript

// positive\_scenarios.spec.js

describe('Registration Form Positive Scenarios', () => {

beforeEach(() => {

cy.visit('/'); // Replace with your registration form URL

});

it('Successfully registers a new user', () => {

cy.get('#fullName').type('John Doe');

cy.get('#email').type('johndoe@example.com');

cy.get('#password').type('securepassword');

cy.get('#confirmPassword').type('securepassword');

cy.get('#dob').type('1990-01-01');

cy.get('#gender').select('Male');

cy.get('#newsletter').check();

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

cy.url().should('include', '/success');

// Add assertions for successful registration

});

it('Registers with newsletter subscription', () => {

cy.get('#fullName').type('Jane Smith');

cy.get('#email').type('janesmith@example.com');

cy.get('#password').type('anothersecurepassword');

cy.get('#confirmPassword').type('anothersecurepassword');

cy.get('#dob').type('1985-05-15');

cy.get('#gender').select('Female');

cy.get('#newsletter').check();

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

cy.url().should('include', '/success');