```
        ┌─────────────┐
        │    start    │
        └──────┬──────┘
               │
               ▼
       ┌───────────────┐
       │ Enter full name│
       └───────┬────────┘
               │
               ▼
       ┌───────────────┐
       │  Enter email  │
       └───────┬────────┘
               │
               ▼
       ┌───────────────┐
       │ Enter password│
       └───────┬────────┘
               │
               ▼
       ┌────────────────┐
       │confirm password│
       └───────┬────────┘
               │
               ▼
       ┌───────────────┐
       │   Enter DOB   │
       └───────┬────────┘
               │
               ▼
       ┌───────────────┐
       │Selected Gender│
       └───────┬────────┘
               │
               ▼
         Newsletter Subscription (Yes/No)

                    yes              no

              ┌──────────────┐
              │   Submitt    │
              └──────────────┘
```

Documentation

a. Test Plan Outline

1. Introduction

The purpose of this test plan is to outline the strategy, scope, and objectives for testing the online registration form of a website. The registration form collects user information and validates inputs to ensure data integrity and a smooth user experience.

2. Objectives

- To verify that the registration form correctly validates all input fields.

- To ensure that error messages are displayed appropriately for invalid inputs.

- To confirm that valid data is successfully submitted to the server.

- To validate the overall functionality, usability, and reliability of the registration form.

3. Scope

- Functional testing of the registration form fields: Full Name, Email Address, Password, Confirm Password, Date of Birth, Gender, and Newsletter Subscription.

- Validation of input fields and error message displays.

- Verification of data submission to the server upon correct completion.

4. Test Environment

- Operating Systems: Windows, macOS, Linux

- Browsers: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari

- Test Tools: Cypress (for automated testing), Selenium WebDriver, Postman (for API testing)

- Test Data: Different sets of user information for various test scenarios.

5. Test Data

- Valid and invalid email addresses.

- Passwords of different lengths and complexities.

- Various combinations of Full Name, Date of Birth, and Gender inputs.

- Different selections for Newsletter Subscription.

6. Test Scenarios

Positive Scenarios:

- Valid input for all fields and successful submission.

- Valid input for all fields except Newsletter Subscription (both options tested).

Negative Scenarios:

- Invalid email address format.

- Password and Confirm Password do not match.

- Password less than 8 characters.

- Empty required fields.

7. Test Execution Schedule

Activity	Start Date	End Date	Responsible
Test Planning	2024-07-16	2024-07-17	QA Engineer
Test Case Development	2024-07-18	2024-07-19	QA Engineer
Test Environment Setup	2024-07-20	2024-07-21	QA Engineer
Test Execution	2024-07-22	2024-07-25	QA Engineer
Test Report & Review	2024-07-26	2024-07-27	QA Engineer

8. Risks and Assumptions

Risks:

- Potential delays in environment setup.

- Incomplete or ambiguous requirements.

Assumptions:

- All necessary resources (test tools, environment access) will be available.

- Requirements for the registration form are clear and complete.

b. Test Cases

Test Case 1: TC_001

Test Case Description: Verify successful registration with valid inputs.

Preconditions: Registration form is accessible.

Test Steps:

- Enter a valid Full Name.

- Enter a valid Email Address.

- Enter a valid Password (at least 8 characters).

- Enter the same Password in Confirm Password.

- Enter a valid Date of Birth.

- Select a Gender.

- Select Yes for Newsletter Subscription.

- Click the Submit Button.

Expected Results: Registration is successful, and a confirmation message is displayed.

Post-conditions: User data is submitted to the server.

Test Case 2: TC_002

Test Case Description: Verify error message for mismatched Password and Confirm Password.

Preconditions: Registration form is accessible.

Test Steps:

- Enter a valid Full Name.

- Enter a valid Email Address.

- Enter a valid Password (at least 8 characters).

- Enter a different Password in Confirm Password.

- Enter a valid Date of Birth.

- Select a Gender.

- Select No for Newsletter Subscription.

- Click the Submit Button.

Expected Results: An error message indicating that the passwords do not match is displayed.

Post-conditions: Form is not submitted.

Test Case 3: TC_003

Test Case Description: Verify error message for invalid email format.

Preconditions: Registration form is accessible.

Test Steps:

- Enter a valid Full Name.

- Enter an invalid Email Address (e.g., "invalidemail.com").

- Enter a valid Password (at least 8 characters).

- Enter the same Password in Confirm Password.

- Enter a valid Date of Birth.

- Select a Gender.

- Select No for Newsletter Subscription.

- Click the Submit Button.

Expected Results: An error message indicating that the email address is invalid is displayed.

Post-conditions: Form is not submitted.

Scenario Identification

Positive Scenarios

Scenario 1:

Steps: User enters valid Full Name, Email Address, Password, Confirm Password, Date of Birth, selects Gender, and subscribes to Newsletter.

Expected Outcome: Registration is successful, and a confirmation message is displayed.

Scenario 2:

Steps: User enters valid Full Name, Email Address, Password, Confirm Password, Date of Birth, selects Gender, and does not subscribe to Newsletter.

Expected Outcome: Registration is successful, and a confirmation message is displayed.

Negative Scenarios

Scenario 1:

Steps: User enters an invalid Email Address format.

Expected Outcome: An error message indicating that the email address is invalid is displayed.

Scenario 2:

Steps: User enters mismatched Password and Confirm Password.

Expected Outcome: An error message indicating that the passwords do not match is displayed.

Test Writing

Automated Test Scripts

Scenario 1: Valid Registration with Newsletter Subscription

(using Selenium)

```
import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;
```

```java
import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;


public class RegistrationFormTests {

    WebDriver driver;


    @BeforeClass
    public void setUp() {

        System.setProperty("webdriver.chrome.driver", "path/to/chromedriver"); // replace with the actual path to chromedriver

        driver = new ChromeDriver();

        driver.get("http://example.com/register"); // replace with the actual URL

    }


    @Test
    public void shouldRegisterWithValidInputsAndSubscribeToNewsletter() {

        driver.findElement(By.name("fullname")).sendKeys("John Doe");

        driver.findElement(By.name("email")).sendKeys("john.doe@example.com");

        driver.findElement(By.name("password")).sendKeys("Password123");

        driver.findElement(By.name("confirmPassword")).sendKeys("Password123");

        driver.findElement(By.name("dob")).sendKeys("1990-01-01");

        driver.findElement(By.name("gender")).sendKeys("Male");

        WebElement newsletterCheckbox = driver.findElement(By.name("newsletter"));

        if (!newsletterCheckbox.isSelected()) {

            newsletterCheckbox.click();
```

```java
        }

        driver.findElement(By.cssSelector("button[type='submit']")).click();


        WebElement successMessage =
driver.findElement(By.xpath("//div[contains(text(),'Registration successful')]"));

        Assert.assertTrue(successMessage.isDisplayed(), "Registration was not successful.");

    }


    @AfterClass

    public void tearDown() {

        driver.quit();

    }

}
```