Test Task: Documentation and Test Scenario Identification

Documentation:

a. Provide a test plan outline for testing the registration form. Include the following sections:

Introduction

The test plan outlines the testing strategy for the online registration form, allowing users to create an account successfully by providing their personal information.

Objectives

- Ensure the registration form functions as intended.
- Verify that the input validations are correctly implemented.
- Verify that the appropriate error messages are displayed for invalid inputs.
- Ensure that the data is being stored on the server while submitting the form.
- Validate the security measures for the password fields.

Scope

- Functional testing of each field.
- The submit button must store the data on the server.
- Usability testing of the user interface.
- Compatibility testing across different browsers and devices.

Test Environment

The test environment includes devices, e.g., desktops, laptops, and phones having latest browser version with a stable internet connection.

Test Data

The test data includes the valid and invalid input for the required fields to test the form's functionality.

Test Scenario

1. Check the fields validation.
2. Check the submit button stores data.
3. Check the password field security.
4. Verify the form is accessible and easy to use for all the users.

Test Execution Schedule

Preparation Phase: 2-3 days

- Set up the test environment.
- Prepare test data.
- Write test cases.

Execution Phase: 1-2 days

- Execute test cases.
- Log defects.

Reporting Phase: 2-3 days

- Compile test results.
- Prepare and deliver the test report.

Risks and Assumptions

- **Risk**
    - Delays in test environment setup.
    - Changes in requirements during the testing phase.
    - Limited availability of testing resources.
- **Assumption**
    - Requirements are stable and well-documented.
    - Test data accurately represents real-world scenarios.

b. Write test cases for at least three scenarios covering different aspects of the registration form.

Scenario 1: Valid Input for all fields

Test Case ID	TC_RF_001
Test Case Description	Verify that the registration form successfully submits data to the server with valid data for all fields.
Preconditions	The registration form is accessible.
Test Steps	1. Open the registration form. 2. Enter "John Doe" in the Full Name field. 3. Enter "john.doe@example.com" in the Email Address field. 4. Enter "Password123" in the Password field. 5. Enter "Password123" in the Confirm Password field. 6. Select "1990-01-01" as the Date of Birth. 7. Select "Male" as the Gender. 8. Select "Yes" for Newsletter Subscription.

	9. Click the Submit button.
Expected Results	1. Form is successfully submitted. 2. User recieves a confirmation message. 3. Server receives the submitted data correctly.
Post-conditions	User account is created successfully with the provided data.

Scenario 2: Invalid email address

Test Case ID	TC_RF_002
Test Case Description	Verify that the appropriate error message shown to the user.
Preconditions	The registration form is accessible.
Test Steps	1. Open the registration form. 2. Enter "John Doe" in the Full Name field. 3. Enter "john.doe#example.com" in the Email Address field. 4. Enter "Password123" in the Password field. 5. Enter "Password123" in the Confirm Password field. 6. Select "1990-01-01" as the Date of Birth. 7. Select "Female" as the Gender. 8. Select "No" for Newsletter Subscription. 9. Click the Submit button.
Expected Results	1. An error message is displayed below the Email Address field indicating that the email format is invalid. 2. Form not submitted 3. Other fields retain their input values.
Post-conditions	The user remains on the registration form page with the error message displayed.

Scenario 3: Password and confirm password mismatch

Test Case ID	TC_RF_003
Test Case Description	Verify that the registration form displays an appropriate error message when the Password and Confirm Password fields do not match.
Preconditions	The registration form is accessible.
Test Steps	1. Open the registration form. 2. Enter "John Doe" in the Full Name field. 3. Enter "john.doe@example.com" in the Email Address field. 4. Enter "Password123" in the Password field. 5. Enter "Password143" in the Confirm Password field.

	6. Select "1990-01-01" as the Date of Birth. 7. Select "Female" as the Gender. 8. Select "No" for Newsletter Subscription. 9. Click the Submit button.
Expected Results	1. An error message is displayed below the Confirm Password field indicating the passwords don't match. 2. Form not submitted 3. Other fields retain their input values.
Post-conditions	The user remains on the registration form page with the error message displayed.

Scenario Identification:

a. Identify at least two positive and two negative scenarios for the registration form. For each scenario, briefly describe the steps a user would take and the expected outcome.

Positive Scenarios

1. Successful Registration with Valid Data in all fields

1. Open the registration form.
2. Enter "John Doe" in the Full Name field.
3. Enter "john.doe@example.com" in the Email Address field.
4. Enter "Password123" in the Password field.
5. Enter "Password123" in the Confirm Password field.
6. Select "1990-01-01" as the Date of Birth.
7. Select "Male" as the Gender.
8. Select "Yes" for Newsletter Subscription.
9. Click the Submit button.

The expected outcome will be:
1. Form is successfully submitted.
2. User recieves a confirmation message.
3. Server receives the submitted data correctly.

2. Successful Registration with minimum required data

1. Open the registration form.
2. Enter "John Doe" in the Full Name field.
3. Enter "john.doe@example.com" in the Email Address field.
4. Enter "Password123" in the Password field.
5. Enter "Password123" in the Confirm Password field.
6. Select "1990-01-01" as the Date of Birth.
7. Select "Male" as the Gender.
8. Leave the Newsletter Subscription filed unchecked.
9. Click the Submit button.

The expected outcome will be:
1. Form is successfully submitted.
2. User recieves a confirmation message.
3. Server receives the submitted data correctly.

Negative Scenarios

1. Password format don’t match

1. Open the registration form.
2. Enter "John Doe" in the Full Name field.
3. Enter "john.doe@example.com" in the Email Address field.
4. Enter "pass12" in the Password field.
5. Enter "pass12" in the Confirm Password field.
6. Select "1990-01-01" as the Date of Birth.
7. Select "Male" as the Gender.
8. Select "Yes" for Newsletter Subscription.
9. Click the Submit button.

The expected outcome will be:
1. Form not submitted.
2. User recieves an error message at password field that it requires atleast 8 characters.
3. Other fields retain their input values.

2. Leaving an empty field that is required

1. Open the registration form.
2. Enter "John Doe" in the Full Name field.
3. Leave the Email Address field empty.
4. Enter "pass12" in the Password field.
5. Enter "pass12" in the Confirm Password field.
6. Select "1990-01-01" as the Date of Birth.
7. Select "Male" as the Gender.
8. Select "Yes" for Newsletter Subscription.
9. Click the Submit button.

The expected outcome will be:
1. Form not submitted.
2. User recieves an error message showing the required email field is empty.
3. Other fields retain their input values.

Test Writing:

a. Write automated test scripts (using a language and testing framework of your choice, e.g Cypress for JavaScript) for the positive scenarios identified in the previous step. Ensure that the scripts cover the key functionalities of the registration form.

1. Successful Registration with Valid Data in all fields

```
using System
using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;

namespace RegistrationFormTests
{
   [TestClass]
   public class RegistrationForm
   {
      private IWebDriver driver;

      [TestMethod]
      public void SuccessfulRegistrationWithValidDataInAllFields()
      {
         driver = new ChromeDriver();
         driver.Url="http://example.com/register";
         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");
         new SelectElement(driver.FindElement(By.Name("gender"))).SelectByValue("Male");
         driver.FindElement(By.Name("newsletter")).Click();

         driver.FindElement(By.CssSelector("button[type='submit']")).Click();

         Assert.IsTrue(driver.PageSource.Contains("Registration successful"));
         driver.Close();
      }
   }
}
```

2. Successful Registration with minimum required data

```
using System
using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;

namespace RegistrationFormTests
{
   [TestClass]
   public class RegistrationForm
   {
```

```
    private IWebDriver driver;

    [TestMethod]
    public void SuccessfulRegistrationWithValidDataInAllFields()
    {
        driver = new ChromeDriver();
        driver.Url="http://example.com/register";
        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");
        new SelectElement(driver.FindElement(By.Name("gender"))).SelectByValue("Male");
        driver.FindElement(By.Name("newsletter")).Click();

        driver.FindElement(By.CssSelector("button[type='submit']")).Click();

        Assert.IsTrue(driver.PageSource.Contains("Registration successful"));
        driver.Close();
    }
  }
}
```