### **Test Plan Outline**

#### **1. Introduction**

This test plan outlines the strategy and approach for testing the online registration form of a website. The form collects user details such as Full Name, Email Address, Password, Confirm Password, Date of Birth, Gender, and Newsletter Subscription preferences.

#### **2. Objectives**

* Ensure that the registration form functions correctly and efficiently.
* Verify that all fields validate inputs appropriately.
* Confirm that appropriate error messages are displayed for invalid inputs.
* Ensure that the form submits data successfully upon correct completion.

#### **3. Scope**

* Functional testing of the registration form.
* Validation of input fields.
* Error message verification.
* Form submission and server response handling.

#### **4. Test Environment**

* Browser: Chrome, Firefox, Safari, Edge
* Operating System: Windows, macOS, Linux
* Devices: Desktop, Mobile

#### **5. Test Data**

* Valid and invalid Full Names, Email Addresses, Passwords, Dates of Birth.
* Different combinations of gender and newsletter subscription preferences.

#### **6. Test Scenarios**

* Positive Scenarios: Successful registration with valid inputs, checking successful form submission.
* Negative Scenarios: Registration with invalid inputs, checking error messages for invalid inputs.

#### **7. Test Execution Schedule**

* Test Case Design: 3 days
* Test Case Execution: 5 days
* Bug Reporting and Fixing: 3 days
* Regression Testing: 2 days

#### **8. Risks and Assumptions**

* Assumption that the registration form's backend is functioning correctly.
* Risk of unavailability of the testing environment.
* Assumption that all required fields are mandatory.

### **Test Cases**

#### **Test Case 1**

**Test Case ID**: TC01

**Test Case Description**: Verify successful registration with valid inputs.

**Preconditions**: User is on the registration page.

**Test Steps**:

1. Enter "Imran Ullah" in the Full Name field.
2. Enter "imranullah114@gmail..com" in the Email Address field.
3. Enter "Password123" in the Password field.
4. Enter "Password123" in the Confirm Password field.
5. Enter "01/01/1997" in the Date of Birth field.
6. Select "Male" as Gender.
7. Select "Yes" for Newsletter Subscription.
8. Click on the Submit button.

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

**Post-conditions**: User account is created.

#### **Test Case 2**

**Test Case ID**: TC02

**Test Case Description**: Verify error message for mismatched passwords.

**Preconditions**: User is on the registration page.

**Test Steps**:

1. Enter "Imran Ullah" in the Full Name field.
2. Enter "imranullah114@gmail.com" in the Email Address field.
3. Enter "Password123" in the Password field.
4. Enter "Password456" in the Confirm Password field.
5. Enter "02/02/1997" in the Date of Birth field.
6. Select "Female" as Gender.
7. Select "No" for Newsletter Subscription.
8. Click on the Submit button.

**Expected Results**: An error message "Passwords do not match" is displayed.

**Post-conditions**: User remains on the registration page.

#### **Test Case 3**

**Test Case ID**: TC03

**Test Case Description**: Verify error message for invalid email address.

**Preconditions**: User is on the registration page.

**Test Steps**:

1. Enter "Muhammad Ali" in the Full Name field.
2. Enter "muhammad.ali" in the Email Address field.
3. Enter "Password123" in the Password field.
4. Enter "Password123" in the Confirm Password field.
5. Enter "03/03/1993" in the Date of Birth field.
6. Select "Other" as Gender.
7. Select "Yes" for Newsletter Subscription.
8. Click on the Submit button.

**Expected Results**: An error message "Invalid email address" is displayed.

**Post-conditions**: User remains on the registration page.

### **Scenarios**

#### **Positive Scenarios**

**Scenario 1: Successful Registration**

* Steps: User enters valid details in all fields and submits the form.
* Expected Outcome: Registration is successful, and a success message is displayed.

**Scenario 2: Successful Registration without Newsletter Subscription**

* Steps: User enters valid details but selects "No" for the Newsletter Subscription.
* Expected Outcome: Registration is successful, and a success message is displayed.

#### **Negative Scenarios**

**Scenario 1: Mismatched Passwords**

* Steps: User enters mismatched passwords and submits the form.
* Expected Outcome: An error message "Passwords do not match" is displayed.

**Scenario 2: Invalid Email Address**

* Steps: User enters an invalid email address format and submits the form.
* Expected Outcome: An error message "Invalid email address" is displayed.

### **Automated Test Scripts**

Below is an example of an automated test script using Java with Selenium, Maven and TestNG:

**Dependencies**:

<dependencies>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.1.0</version>

</dependency>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.4.0</version>

</dependency>

</dependencies>

**Test Script**:

Registration Page Class

public class RegistrationPage {

WebDriver driver;

By fullNameField = By.id("fullName");

By emailField = By.id("email");

By passwordField = By.id("password");

By confirmPasswordField = By.id("confirmPassword");

By dobField = By.id("dob");

By genderMaleRadio = By.id("genderMale");

By newsletterYesCheckbox = By.id("newsletterYes");

By submitButton = By.id("submit");

By successMessage = By.id("successMessage");

public RegistrationPage(WebDriver driver) {

this.driver = driver;

}

public void enterFullName(String fullName) {

driver.findElement(fullNameField).sendKeys(fullName);

}

public void enterEmail(String email) {

driver.findElement(emailField).sendKeys(email);

}

public void enterPassword(String password) {

driver.findElement(passwordField).sendKeys(password);

}

public void enterConfirmPassword(String confirmPassword) {

driver.findElement(confirmPasswordField).sendKeys(confirmPassword);

}

public void enterDOB(String dob) {

driver.findElement(dobField).sendKeys(dob);

}

public void selectGenderMale() {

driver.findElement(genderMaleRadio).click();

}

public void selectNewsletterYes() {

driver.findElement(newsletterYesCheckbox).click();

}

public void clickSubmit() {

driver.findElement(submitButton).click();

}

public boolean isSuccessMessageDisplayed() {

return driver.findElement(successMessage).isDisplayed();

}

}

**Base Class**

public class BaseClass {

WebDriver driver;

RegistrationPage registrationPage;

@BeforeMethod

public void setup() {

System.setProperty("webdriver.chrome.driver", "chromedriver");

driver = new ChromeDriver();

driver.get("http://example.com/registration");

registrationPage = new RegistrationPage(driver);

}

@AfterMethod

public void teardown() {

driver.quit();

}

}

**Registration From Test class**

public class RegistrationFormTest extends BaseClass {

@Test(priority=1)

public void testEnterFullName() {

registrationPage.enterFullName("khan khan");

Assert.assertTrue(true, "Full Name entered successfully");

}

@Test(priority=2)

public void testEnterEmail() {

registrationPage.enterEmail("khan@gmail.com");

Assert.assertTrue(true, "Email entered successfully");

}

@Test(priority=3)

public void testEnterPassword() {

registrationPage.enterPassword("Password123");

Assert.assertTrue(true, "Password entered successfully");

}

@Test(priority=4)

public void testEnterConfirmPassword() {

registrationPage.enterConfirmPassword("Password123");

Assert.assertTrue(true, "Confirm Password entered successfully");

}

@Test(priority=5)

public void testEnterDOB() {

registrationPage.enterDOB("01/01/1990");

Assert.assertTrue(true, "Date of Birth entered successfully");

}

@Test(priority=6)

public void testSelectGender() {

registrationPage.selectGenderMale();

Assert.assertTrue(true, "Gender selected successfully");

}

@Test(priority=7)

public void testSelectNewsletter() {

registrationPage.selectNewsletterYes();

Assert.assertTrue(true, "Newsletter preference selected successfully");

}

@Test(priority=8)

public void testSubmit() {

registrationPage.clickSubmit(); Assert.assertTrue(registrationPage.isSuccessMessageDisplayed(), "Form submitted successfully"); }}