**Name: Ushna Karim**

**Test Plan Outline**

**Introduction**

This document outlines the strategy and approach for testing the website's online registration form. The primary goal is to guarantee a seamless user experience by ensuring the registration form.

**Objectives**

* Verify that the registration form accepts valid input and rejects invalid input.
* Displays appropriate error messages for each type of invalid input.
* Transmits validated data upon successful form completion.

**Scope**

This test plan focuses on evaluating the functionality and user experience of the online registration form. The following areas will be thoroughly examined:

* **Input Field Validation:** Verify that the form enforces proper data formats and rejects invalid or incomplete entries with clear error messages.
* **Error Message Display:** Ensure appropriate and informative error messages are displayed for each type of invalid input, guiding users towards correcting their information.
* **Successful Form Submission**: Confirm that the form successfully transmits validated data to the server upon completion and provides a confirmation message to the user.
* **Cross-Browser Compatibility:** Test the form across various popular browsers (specify which browsers) to ensure consistent functionality and user experience.

By covering these critical aspects, this test plan aims to guarantee a reliable and user-friendly registration process.  
  
**Test Environment**

* Web Browsers: Chrome, Edge
* Devices: Desktop, Mobile
* Operating Systems: Windows, iOS
* Tools: Selenium WebDriver

**Test Data**

* Email addresses (valid and invalid)
* Passwords (strong, weak, mismatched)
* Dates of birth (valid range, future dates)
* Gender and newsletter subscription options (all combinations)

**Test Scenarios**

* Valid User Registration (combines Valid Input & Successful Submission)
* Invalid Email Address
* Password Mismatch
* Missing Required Fields

**Test Execution Schedule**

* Test Planning: 1 day
* Test Case Design: 2 days
* Test Data Preparation: 1 day
* Test Execution: 4 days
* Reporting: 2 days

**Risks and Assumptions**

1. Assumptions:  
   Stable build provided by development team.
2. Risks:   
   Form changes during testing.
3. Mitigation:  
   Frequent communication with development team.  
   Adapting tests to accommodate changes.

**Test Cases**

**1. Valid User Registration (TC-01)**

**Description:** To verify successful registration with valid data.

**Preconditions:** None

**Test Steps**:

* 1. Access the registration page.
  2. Enter valid data for all fields.
  3. Click "Submit."

**Expected Results:**

1. Form successfully submits data.
2. Confirmation message appears (e.g., "Registration successful!").

**Post-conditions:** User is logged in or receives a confirmation email.

**2. Invalid Email Address (TC-02)**

**Description:** To verify registration is not possible with invalid email address.

**Preconditions:** None

**Test Steps**:

* 1. Go to registration page
  2. Enter valid information except for email (e.g., "username" instead of "username@email.com").
  3. Click "Submit."

**Expected Results:**

1. Form does not submits data.
2. Error message displays for invalid email format.

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

**3. Password Mismatch (TC-03)**

**Description:** To verify registration is not possible with mismatched passwords.

**Preconditions:** None

**Test Steps**:

1. Go to registration page
2. Enter valid information for all fields.
3. Enter a different password in the "Confirm Password" field.
4. Click "Submit."

**Expected Results:**

1. Form does not submits data.
2. Error message displays for password mismatch.

**Post-conditions:** User remains on the registration page with the error message

**Scenario Identification**

**Positive Scenarios:**

* Valid User Registration: User enters all information correctly and successfully registers.
* Strong Password: User creates a strong password that meets complexity requirements.

**Negative Scenarios:**

* Missing Required Field: User omits a mandatory field (e.g., email address) and receives an error message.
* Weak Password: User creates a weak password that doesn't meet complexity requirements and receives an error message.

**Automated Scripts (Python, Selenium)**

#VALID USER REGISTRATION  
  
  
from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

import time

# Set up the webdriver

driver = webdriver.Chrome()

# Open the registration page

driver.get("https://eliteform.com/register")

# Fill in the registration form with valid data

driver.find\_element(By.NAME, "fullName").send\_keys("Ushna Karim")

driver.find\_element(By.NAME, "email").send\_keys("ushnakarim@example.com")

driver.find\_element(By.NAME, "password").send\_keys("UshnaPassword")

driver.find\_element(By.NAME, "confirmPassword").send\_keys("UshnaPassword")

driver.find\_element(By.NAME, "dob").send\_keys("2001-02-08")

driver.find\_element(By.NAME, "gender").send\_keys("female")

driver.find\_element(By.XPATH, "//input[@name='newsletter' and @value='Yes']").click()

# Submit the form

driver.find\_element(By.XPATH, "//button[@type='submit']").click()

# Wait for the response and check for success message

time.sleep(5)

assert "Registration successful" in driver.page\_source

# Close the browser

driver.quit()

#STRONG PASSWORD

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

import time

# Set up the webdriver

driver = webdriver.Chrome()

# Open the registration page

driver.get("https://eliteform.com/register")

# Fill in the registration form with a strong password

driver.find\_element(By.NAME, "fullName").send\_keys("Ushna Karim")

driver.find\_element(By.NAME, "email").send\_keys("ushnakarim@example.com")

driver.find\_element(By.NAME, "password").send\_keys("UshnaPassword")

driver.find\_element(By.NAME, "confirmPassword").send\_keys("UshnaPassword")

driver.find\_element(By.NAME, "dob").send\_keys("2001-02-08")

driver.find\_element(By.NAME, "gender").send\_keys("female")

driver.find\_element(By.XPATH, "//input[@name='newsletter' and @value='Yes']").click()

# Submit the form

driver.find\_element(By.XPATH, "//button[@type='submit']").click()

# Wait for the response and check for success message

time.sleep(5)  # Adjust based on the actual response time

assert "Registration successful" in driver.page\_source

# Close the browser

driver.quit()