Name: Khawaja Abdullah
Email: k200385@nu.edu.pk

Linkedin: https://www.linkedin.com/in/khawajaabdullah/

Task:

Documentation:

Part a)

Introduction

This test plan outlines the testing strategy for the online registration form. The objective is to ensure that the registration form functions correctly, validates user input effectively, and successfully submits data to the server.

Objectives

- Verify that all mandatory fields are correctly identified and required for form submission.
- Validate input data for accuracy and consistency.
- Ensure appropriate error messages are displayed for invalid input.
- Confirm successful data submission to the server.

Scope

- Functional testing of the registration form.
- Data validation testing.
- Error message verification.
- Data submission testing.

Test Environment

- Browser compatibility (e.g., Chrome, Firefox, Edge, Safari)
- Operating systems (e.g., Windows, macOS, Linux)
- Hardware specifications (e.g.processor, memory)
- Test data environment (e.g. database)
- Network conditions

Test Data

- Valid and invalid test data for all fields (e.g., full names, email addresses, passwords, dates of birth, genders).
- Boundary value test data (e.g., minimum/maximum password length, date of birth range).
- Special character test data.
- Performance test data (e.g., multiple concurrent users).

Test Scenarios

- Mandatory field validation
- Data type validation (e.g., email format, date format)
- Data length validation (e.g., password length, name length)
- Password matching validation
- Error message verification
- Successful registration and data submission
- Duplicate email validation

Test Execution Schedule

- Define test phases (unit, integration, system, acceptance)
- Assign responsibilities
- Create a detailed test case execution matrix
- Determine test environment availability
- Schedule test execution and defect reporting

Risks and Assumptions

- Assumptions about the application's behavior (e.g., expected error messages, data format)
- Potential risks (e.g., data loss, performance issues, security vulnerabilities)
- Mitigation strategies for identified risks

Part b)

Test Case 1: Mandatory Field Validation

Test Case ID: TC_001
Test Case Description: Verify that all mandatory fields are required for form submission.
Preconditions: Registration form is accessible.
Test Steps:

1. Leave all fields blank.

2. Click the "Submit" button.

Expected Results:

- Error messages are displayed for all mandatory fields.
- Form submission is prevented.

Post-conditions: User remains on the registration page with error messages displayed.

Test Case 2: Data Type Validation (Email)

Test Case ID: TC_002
Test Case Description: Verify that the email field accepts valid email formats and rejects invalid ones.
Preconditions: Registration form is accessible.
Test Steps:

1. Enter an invalid email format (e.g., "invalid_email") in the email field.
2. Click the "Submit" button.

Expected Results:

- An error message is displayed indicating an invalid email format.
- Form submission is prevented.

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

Test Case 3: Password Validation

Test Case ID: TC_003
Test Case Description: Verify that the password field meets the minimum length requirement and matches the confirm password field.
Preconditions: Registration form is accessible.
Test Steps:

1. Enter a password shorter than the minimum required length in the password field.
2. Enter the same password in the confirm password field.
3. Click the "Submit" button.

Expected Results:

- An error message is displayed indicating that the password is too short.
- Form submission is prevented.

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

Scenario Identification:

Part a)

Positive Scenarios

Scenario 1: Successful Registration

- **Steps:**
    - User provides all required information in the registration form with valid data types and formats.
    - User clicks the "Submit" button.
- **Expected Outcome:**
    - Form is submitted successfully.
    - User receives a confirmation message or is redirected to a confirmation page.
    - User account is created in the system.

Scenario 2: Newsletter Subscription

- **Steps:**
    - User provides all required information in the registration form.
    - User checks the "Newsletter Subscription" checkbox.
    - User clicks the "Submit" button.
- **Expected Outcome:**
    - Form is submitted successfully.
    - User account is created and added to the newsletter subscription list.

Negative Scenarios

Scenario 1: Invalid Email Format

- **Steps:**
    - User enters an invalid email address format (e.g., "invalid_email").
    - User provides all other required information with valid data.
    - User clicks the "Submit" button.
- **Expected Outcome:**
    - Form submission is prevented.
    - An error message is displayed indicating an invalid email format.

Scenario 2: Password Mismatch

- **Steps:**
    - User enters a password in the "Password" field.
    - User enters a different password in the "Confirm Password" field.
    - User provides all other required information with valid data.
    - User clicks the "Submit" button.
- **Expected Outcome:**
    - Form submission is prevented.
    - An error message is displayed indicating a password mismatch.

Test Writing:

Part a)

Using cypress for automated testing

```
describe('Registration Form', () => {
  it('Successful Registration', () => {
    const userData = {
      fullName: 'Test User',
      email: 'test@gmail.com',
      password: 'password123',
      confirmPassword: 'password123',
      dateOfBirth: '1998-01-01',
      gender: 'Male',
      newsletterSubscription: true
    };

    cy.visit('http://localhost:3000/register');

    // Fill in registration form
    cy.get('[data-cy="fullName"]').type(userData.fullName);
    cy.get('[data-cy="email"]').type(userData.email);
    cy.get('[data-cy="password"]').type(userData.password);
    cy.get('[data-cy="confirmPassword"]').type(userData.confirmPassword);
    cy.get('[data-cy="dateOfBirth"]').type(userData.dateOfBirth);
    cy.get('[data-cy="gender"]').select(userData.gender);
    cy.get('[data-cy="newsletterSubscription"]').check();

    // Submit the form
    cy.get('[data-cy="submitButton"]').click();
    cy.contains('Registration successful').should('be.visible');
  });
  it('Newsletter Subscription', () => {
    const userData = {
 fullName: 'Test User',
      email: 'test@gmail.com',
      password: 'password123',
      confirmPassword: 'password123',
      dateOfBirth: '1998-01-01',
      gender: 'Male'
    };

    cy.visit('http://localhost:3000/register');

    // Fill in registration form and submit
    // ... same as above

    // Assert newsletter subscription
    cy.contains('Thank you for subscribing to our newsletter').should('be.visible');
  });
});
```