Test Plan for Registration Form

Introduction

This test plan describes the methodology and approach for testing a website's online registration form. Information about the user, including their full name, email address, password, password confirmation, date of birth, gender, and newsletter subscription, is gathered through the registration form. After filling out correctly, the form must successfully send the data to the server, validate inputs, and provide the relevant error messages.

Objectives

- Verify that the registration form is operating properly.
- Verify the entry of correct information in the input fields.
- Check that the error messages are shown correctly.
- Verify that the data you sent to the server was successful.

Scope

- Conduct functional testing for all input fields and validation rules.
- Perform UI/UX testing to ensure a user-friendly design and intuitive flow.
- Carry out compatibility testing across various browsers and devices.
- Execute security testing to ensure data privacy and protection.

Test Levels and Strategies

- Unit Testing: software module form will be tested to ensure they function correctly according to their specifications (white-box testing).
- Integration Testing: Form will be integrated to the website sign up flow and tested together to identify any integration issues (black-box testing).
- System Testing: The entire feature will be tested as a whole to ensure all functionalities work seamlessly (black-box testing)
- Acceptance Testing: User representatives will test the feature to confirm it meets business requirements and user needs (user acceptance testing).

Test Environment

- Testing will be conducted on a dedicated test server environment replicating the production environment. I.e. QA Server/Staging Server
- The environment will include:
    - Database: MongoDB
    - OS: Windows Server 2022
    - Web browsers: Chrome, Firefox, Safari, Edge
    - Devices: Desktop, Tablet, Mobile
    - OS: Windows, macOS, iOS, Android

Test Data

- Valid and invalid email addresses
- Valid and invalid passwords

- Various combinations of user input (e.g., empty fields, incorrect formats)
- Valid and Invalid DOB format

Test Scenarios

- Valid input data
- Invalid email format
- Password and confirm password mismatch
- Empty required fields
- Date of Birth validation
- Gender selection validation
- Newsletter subscription toggle
- Successful form submission

Test Execution Schedule

Strategy	Estimates
Test Planning	1 Day
Test Case Design	2 Days
Test Case Executions	2 Days
Bug Reporting	1 Day
Bug Fixing	½ Day
Retesting	½ Day
Test Review	½ Day
Test Closure	½ Day

Defect Management

- Defects identified during testing will be reported in a Jira.
- Each defect report will include:
    - Description of the defect
    - Steps to reproduce the defect
    - Severity level (critical, major, minor)
    - Priority level (high, medium, low)

Test Reporting

- Test reports will be generated after each sprint, summarising:
    - Number of test cases executed
    - Number of passed and failed test cases
    - Identified defects and their status
    - Overall test coverage achieved

Schedule and Resources

The testing phase is estimated to take 1 week.

- Resources required include:
    - 1 QA Tester
    - Bug Tracking System (Jira)

Risks and Assumptions

- Requirement Gaps: Incomplete or ambiguous requirements can result in overlooked test scenarios, leading to potential defects in the final product.
- Test Environment Instability: Unreliable or unavailable test environments can hinder testing efforts, causing delays and impacting test coverage.
- Form Validation Inconsistencies: Deviations from standard form validation rules can lead to unexpected user experiences and potential security vulnerabilities.

Test Cases

Test Case 1: Valid Input Data

- **Test Case ID**: TC001
- **Test Case Description**: Verify the registration form submission with valid input data.
- **Preconditions**: The registration form is accessible.
- **Test Steps**:
    1. Enter a valid Full Name.
    2. Enter a valid Email Address.
    3. Enter a valid Password (at least 8 characters).
    4. Enter the same password in Confirm Password.
    5. Select a valid Date of Birth.
    6. Select a Gender.
    7. Choose 'Yes' or 'No' for Newsletter Subscription.
    8. Click the Submit button.
- **Expected Results**: The form is submitted successfully and a success message is displayed.
- **Post-conditions**: User account is created.

Test Case 2: Invalid Email Format

- **Test Case ID**: TC002
- **Test Case Description**: Verify the error message for an invalid email format.
- **Preconditions**: The registration form is accessible.
- **Test Steps**:
    1. Enter a valid Full Name.
    2. Enter an invalid Email Address (e.g., "invalidemail").
    3. Enter a valid Password (at least 8 characters).
    4. Enter the same password in Confirm Password.
    5. Select a valid Date of Birth.
    6. Select a Gender.
    7. Choose 'Yes' or 'No' for Newsletter Subscription.
    8. Click the Submit button.

- **Expected Results**: An error message "Please enter a valid email address" is displayed.
- **Post-conditions**: Form is not submitted.

Test Case 3: Password and Confirm Password Mismatch

- **Test Case ID**: TC003
- **Test Case Description**: Verify the error message when Password and Confirm Password do not match.
- **Preconditions**: The registration form is accessible.
- **Test Steps**:
    1. Enter a valid Full Name.
    2. Enter a valid Email Address.
    3. Enter a valid Password (at least 8 characters).
    4. Enter a different password in Confirm Password.
    5. Select a valid Date of Birth.
    6. Select a Gender.
    7. Choose 'Yes' or 'No' for Newsletter Subscription.
    8. Click the Submit button.
- **Expected Results**: An error message "Passwords do not match" is displayed.
- **Post-conditions**: Form is not submitted.

Scenario Identification

Positive Scenarios

1. **Scenario 1: Successful Registration**
    - Steps: User fills in all fields with valid data and submits the form.
    - Expected Outcome: Form submits successfully, and a success message is displayed.
2. **Scenario 2: Newsletter Subscription**
    - Steps: User fills in all fields with valid data, chooses 'Yes' for Newsletter Subscription, and submits the form.
    - Expected Outcome: Form submits successfully, and the user is subscribed to the newsletter.

Negative Scenarios

1. **Scenario 1: Invalid Email Format**
    - Steps: User enters an invalid email format and submits the form.
    - Expected Outcome: Error message "Please enter a valid email address" is displayed.
2. **Scenario 2: Password and Confirm Password Mismatch**
    - Steps: User enters different passwords in the Password and Confirm Password fields and submits the form.
    - Expected Outcome: Error message "Passwords do not match" is displayed.
3. **Scenario 3: Future Dates for DOB**
    - Steps: User try to choose DOB from future dates

    - Expected Outcome: Users should not be able to choose DOB from future dates. They all should be disabled.
4. **Scenario 4: Double Click Submit button**
    - Steps: User try to double click the submit button
    - Expected Outcome: there should not be two entries in the system. Users should not be able to double click the submit button.

Test Writing

Automated Test Scripts - Link to Github repo https://github.com/rafia08/QAtask

Test Plan for Automation Testing

Tools and Technologies: A suitable automated testing framework (e.g., Selenium) will be chosen based on system technology.

Tools: IntelliJ will be used for writing automated scripts using Java language.

Scope of Automation: Automation will focus on regression testing of registration form.
