# **Test Plan for Online Registration Form**

Contents

[**Test Plan for Online Registration Form** 1](#_Toc171891596)

[**Objective** 2](#_Toc171891597)

[**Scope** 3](#_Toc171891598)

[**Test Strategy** 3](#_Toc171891599)

[**Test Environments** 6](#_Toc171891600)

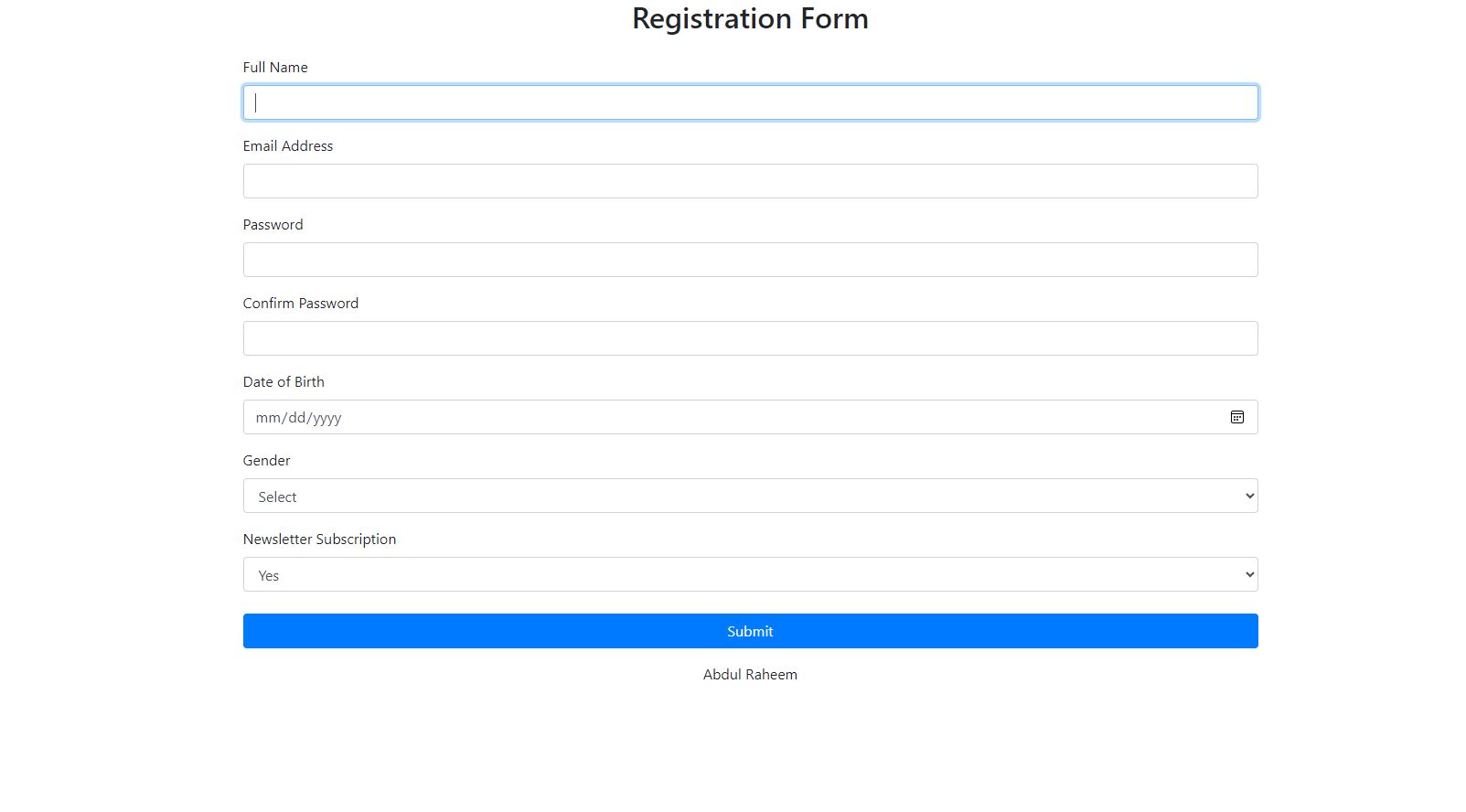
#### 

**Introduction**

* This test plan outlines the approach to testing the online registration form of a website. The form is designed to collect user information, validate inputs, and submit data to the server. Ensuring the functionality and reliability of this form is critical for user onboarding and data integrity.

## **Objective**

* Verify that all form fields validate inputs correctly.
* Ensure appropriate error messages are displayed for invalid inputs.
* Confirm successful data submission upon correct completion of the form.
* Assess the form’s usability, security, and performance.



## 

## **Scope**

* **Functional Testing:** Validate each field’s input requirements and overall form submission.
* **Usability Testing:** Ensure the form is user-friendly and accessible.
* **Security Testing:** Check for vulnerabilities such as SQL injection and XSS attacks.
* **Performance Testing:** Evaluate the form’s response time under different load conditions.

### **Test Strategy**

The test strategy outlines the approach and methodology for testing the online registration form. This strategy ensures that all functional, usability, security, and performance aspects are thoroughly tested to deliver a robust and user-friendly registration form.

#### **1. Functional Testing**

**Objective:** Ensure that each field in the registration form meets its input requirements and that the form can be submitted successfully when all fields are correctly filled.

**Approach:**

* **Field Validation:**
  + Verify that required fields cannot be left empty.
  + Validate specific input requirements for each field (e.g., email format, password length).
  + Check error messages for invalid inputs.
* **Form Submission:**
  + Test form submission with all valid inputs.
  + Ensure that the form data is correctly submitted to the server.
  + Confirm that a success message or redirection occurs upon successful submission.

**Examples of Functional Test Cases:**

* Test case for validating empty fields.
* Test case for incorrect email format.
* Test case for password length requirements.

#### **2. Usability Testing**

**Objective:** Ensure that the registration form is user-friendly, intuitive, and accessible to all users, including those with disabilities.

**Approach:**

* **User Interface (UI) Design:**
  + Verify that the form layout is clear and visually appealing.
  + Ensure that labels, placeholders, and instructions are clear and concise.
* **Accessibility:**
  + Test the form for compatibility with screen readers.
  + Ensure that the form can be navigated using a keyboard.
  + Verify color contrast and font size for readability.
* **User Experience (UX):**
  + Conduct user testing sessions to gather feedback on ease of use.
  + Identify any pain points or confusion during the registration process.

**Examples of Usability Test Cases:**

* Test case for keyboard navigation.
* Test case for screen reader compatibility.
* Test case for user feedback on form layout.

#### **3. Security Testing**

**Objective:** Identify and mitigate vulnerabilities that could compromise the security of the registration form and user data.

**Approach:**

* **SQL Injection Testing:**
  + Attempt to inject malicious SQL queries into input fields to ensure the form is secure against SQL injection attacks.
  + Verify that the form properly sanitizes inputs and rejects malicious queries.
* **Cross-Site Scripting (XSS) Testing:**
  + Inject XSS payloads into input fields to check if the form is vulnerable to XSS attacks.
  + Ensure that the form properly escapes and sanitizes inputs to prevent script execution.
* **Data Privacy:**
  + Verify that sensitive data (e.g., passwords) is encrypted during transmission.
  + Ensure that user data is stored securely on the server.

**Examples of Security Test Cases:**

* Test case for SQL injection attempts.
* Test case for XSS payload injection.
* Test case for secure data transmission.

#### **4. Performance Testing**

**Objective:** Assess the performance of the registration form under different load conditions to ensure it can handle a high number of users without degradation in performance.

**Approach:**

* **Load Testing:**
  + Simulate a high number of users submitting the form simultaneously to measure the response time and performance.
  + Identify any bottlenecks or slow responses.
* **Stress Testing:**
  + Test the form under extreme conditions to determine its breaking point.
  + Ensure the form gracefully handles errors and recovers from failures.
* **Scalability Testing:**
  + Evaluate how the form performs as the number of users increases over time.
  + Ensure the system can scale to accommodate growing user numbers.

**Examples of Performance Test Cases:**

* Test case for load testing with 1000 simultaneous users.
* Test case for stress testing to find the breaking point.
* Test case for monitoring form performance over time.

### **Test Environments**

* **Browsers:** Latest versions of Chrome, Firefox, Safari, and Edge.
* **Devices:** Desktop, tablet, and mobile devices.
* **Server:** Test server replicating the production environment.

Windows 10 – Chrome, Firefox and Edge

• Mac OS – Safari Browser

• Android Mobile OS – Chrome

• IPhone Mobile OS - Safari

**Test Execution Schedule**

* **Test case design:** 1 day
* **Test execution:** 1 day
* **Bug reporting and fixing:** 1 to 2 days

**Risks and Assumptions**

* **Risks:** Delays in server setup, browser compatibility issues, changes in requirements.
* **Assumptions:** Form fields and validations are as per the requirements, the test environment is stable, and necessary test data is available.