**Test Plan for Ultimate QA automation website**

***https://ultimateqa.com/automation***

[1. Introduction 2](#_Toc164337499)

[2. Objectives 2](#_Toc164337500)

[3. Scope 2](#_Toc164337501)

[4. Risk-Based Testing 2](#_Toc164337502)

[4.1 Identification of Risks 2](#_Toc164337503)

[4.2 Risk Mitigation Strategy 3](#_Toc164337504)

[5. Testing Strategy 3](#_Toc164337505)

[6. Testing Scenarios 3](#_Toc164337506)

[6.1 Functional Testing 3](#_Toc164337507)

[6.2 Security Testing 4](#_Toc164337508)

[6.3 Performance Testing 5](#_Toc164337509)

[6.4 Usability Testing 5](#_Toc164337510)

[6.5 Compatibility Testing 6](#_Toc164337511)

[7. Resources 6](#_Toc164337512)

[8. Timelines 6](#_Toc164337513)

[9. Conclusion 6](#_Toc164337514)

# 1. Introduction

This document outlines the test plan for the Ultimate QA Automation Practice website. The purpose of this test plan is to ensure that the website functions correctly, meets its intended objectives, and provides valuable learning experience for testers practicing automation.

# 2. Objectives

The objectives of this test plan are as follows:

* To verify that all features of the web application work as intended.
* To identify and mitigate risks associated with the application.
* To ensure the usability, reliability, and performance of the application.
* To validate the compatibility of the application with different browsers and devices.
* To provide confidence to stakeholders regarding the quality of the application.

# 3. Scope

The scope of testing includes:

* Functional testing of all features and functionalities.
* Non-functional testing including performance, security, usability, and compatibility testing.
* Testing across different browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, mobile, tablet).
* Automation testing to cover all functionalities.

# 4. Risk-Based Testing

## 4.1 Identification of Risks

The following risks have been identified:

1. Security vulnerabilities: Due to the sensitive nature of user data, security breaches could lead to reputational damage and legal consequences.
2. Performance issues under load: High traffic could result in slow response times or system crashes.
3. Compatibility issues: Variations in browser behavior and device configurations may affect the user experience.
4. Incomplete functionality: Features may not work as expected due to incomplete implementation or changes in requirements.

## 4.2 Risk Mitigation Strategy

1. Security testing will be conducted to identify and address vulnerabilities.
2. Performance testing will simulate various levels of user load to assess system behavior under stress.
3. Compatibility testing will cover a range of browsers and devices to ensure a consistent user experience.
4. Continuous communication with development teams will ensure that any incomplete functionality is identified and prioritized for testing.

# 5. Testing Strategy

The testing strategy will include the following approaches:

* Black-box testing: Functional testing will be conducted without knowledge of the internal code to simulate user interactions and verify expected behavior.
* White-box testing: Code reviews and unit tests will be performed to ensure the integrity of the codebase and identify potential defects.
* Regression testing: Automated regression tests will be executed after each code change to ensure that new features do not introduce regressions in existing functionality.
* Exploratory testing: Ad-hoc testing will be conducted to explore the application and identify potential defects that may not be covered by scripted tests.

# 6. Testing Scenarios

## 6.1 Functional Testing

**Navigation to Different Pages:**

* Verify that all links on the parent page lead to the correct destinations, including the Fake Landing Page and Fake Pricing Page.
* Ensure that the navigation menu or buttons allow users to easily switch between different pages.

**Form Interactions:**

* Fill out the forms available on the website, including any registration or contact forms.
* Verify that form submission works correctly and that users receive appropriate feedback (success/error messages) after submission.
* Test form validation by entering invalid data and ensuring that appropriate error messages are displayed.

**Login Automation:**

* Test the login functionality by automating the login process using provided credentials.
* Verify that users are able to successfully log in and access restricted content after authentication.
* Test for various scenarios such as incorrect username/password, expired sessions, and successful login attempts.

**Interactions with Simple Elements:**

* Interact with simple elements on the website such as buttons, checkboxes, radio buttons, and dropdown menus.
* Verify that interactions with these elements produce the expected results, such as toggling checkboxes, selecting options from dropdown menus, and triggering actions with buttons.

## 6.2 Security Testing

1. **Injection**

* Attempt SQL injection attacks on forms and input fields to verify if the website is vulnerable to SQL injection.
* Test for other injection vulnerabilities such as command injection by injecting malicious commands into input fields.

1. **Broken Authentication**

* Test for weak password policies by attempting to create accounts with weak passwords.
* Attempt to bypass authentication mechanisms by manipulating session tokens or cookies.
* Test for account enumeration vulnerabilities by identifying differences in error messages for valid and invalid usernames during login attempts.

1. **Sensitive Data Exposure**

* Test for insecure transmission of sensitive data by intercepting network traffic to check if data is transmitted over HTTP instead of HTTPS.
* Verify that sensitive data such as passwords and payment information is properly encrypted during transmission and storage.
* Test for exposure of sensitive data in error messages, logs, or response headers.

1. **Cross-Site Scripting (XSS)**

* Test for reflected and stored XSS vulnerabilities by injecting malicious scripts into input fields or URLs.
* Verify that the application properly encodes user input to prevent XSS attacks.
* Test for DOM-based XSS vulnerabilities by analyzing client-side scripts for insecure data handling.

1. **Insufficient Logging and Monitoring**

* Test for insufficient logging by verifying that security-relevant events such as failed login attempts and access control failures are logged.
* Check the adequacy of log entries by reviewing their content and level of detail.
* Verify that the application has monitoring mechanisms in place to detect and respond to security incidents in real-time.

## 6.3 Performance Testing

* Test the website's performance by simulating different levels of user traffic.
* Measure the response time of critical actions such as page loading, form submission, and navigation between pages.
* Ensure that the website remains responsive and stable under normal and peak load conditions.

## 6.4 Usability Testing

* Evaluate the clarity and intuitiveness of the website's layout and design.
* Test the readability of text content and the visibility of important elements.
* Verify that users can easily understand and navigate through the website to achieve their goals.

## 6.5 Compatibility Testing

* Test the website's compatibility across different browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, mobile, tablet).
* Verify that the website's layout and functionality remain consistent across various platforms and screen sizes.

# 7. Resources

The following resources will be required for testing:

* Test environment (development, staging, production)
* Test data (sample user accounts, content)
* Testing tools (automation tools, security scanners, performance testing tools)
* Testing team (2 testers, developer for white-box testing, stakeholders for feedback)

# 8. Timelines

The testing timeline will be as follows:

|  |  |
| --- | --- |
| **Planning and preparation** | April 20, 2024 - April 22, 2024 |
| **Test execution** | April 23, 2024 - April 29, 2024 |
| **Automation implementation** | Ongoing throughout the testing phase |
| **Reporting and documentation** | April 30, 2024 |

# 9. Conclusion

This test plan outlines the approach, strategies, and resources required to ensure the quality and usability of the Ultimate QA Automation Practice website. By following this plan, we aim to provide a valuable learning experience for testers practicing automation techniques and ensure the reliability of the website for future users.