# Test Plan (VWO.com)

**[Test Plan](#_heading=h.30j0zll)**

[Objective](#_heading=h.3znysh7)

[Scope](#_heading=h.2et92p0)

[Inclusions](#_heading=h.tyjcwt)

[Test Environments](#_heading=h.3dy6vkm)

Testing Types

[Defect Reporting Procedure](#_heading=h.1t3h5sf)

[Test Strategy](#_heading=h.4d34og8)

[Test Schedule](#_heading=h.2s8eyo1)

[Test Deliverables.](#_heading=h.17dp8vu)

[Entry and Exit Criteria](#_heading=h.3rdcrjn)

[Entry Criteria:](#_heading=h.26in1rg)

[Exit Criteria:](#_heading=h.lnxbz9)

[Test Execution](#_heading=h.35nkun2)

[Entry Criteria:](#_heading=h.1ksv4uv)

[Exit Criteria:](#_heading=h.44sinio)

[Test Closure](#_heading=h.2jxsxqh)

[Entry Criteria:](#_heading=h.z337ya)

[Exit Criteria:](#_heading=h.3j2qqm3)

[Tools](#_heading=h.1y810tw)

[Risks and Mitigations](#_heading=h.4i7ojhp)

[Approvals](#_heading=h.2xcytpi)

**Test Plan**

This test plan outlines the objectives, scope, testing strategy, test execution approach, and deliverables for the VWO.com application.

**Objective**

* Ensure that the VWO.com application meets functional and non-functional requirements.
* Identify and report defects to improve application quality.
* Verify compatibility across various devices and browsers.
* Ensure security, usability, and performance of the application.
* Validate that all functional and non-functional requirements are met.
* Ensure compatibility across multiple devices, browsers, and operating systems.
* Identify defects and ensure a high-quality user experience.
* Verify API responses and integration with third-party services.
* Ensure the application is secure against potential cyber threats.
* Conduct regression testing to prevent defects in existing functionality

**Scope**

* **Scope for Login Page:**
* **Functional Testing:** Valid/Invalid login, forgot password, account lockout, two-factor authentication.
* **UI/UX Testing:** Responsive design, error messages, accessibility.
* **Security Testing:** Brute force prevention, SQL injection, session expiration.
* **Performance Testing:** Login response time, concurrent user load.
* **Compatibility Testing:** Cross-browser and device testing.
* **Scope for Dashboard Application:**
* **Functional Testing:** Navigation, role-based access, widgets, notifications, data export.
* **UI/UX Testing:** Proper alignment, readability, responsive design.
* **Security Testing:** Access control, unauthorized data access prevention, encryption.
* **Performance Testing:** Dashboard load time, stress testing.
* **Compatibility Testing:** Browser and device compatibility.

**Inclusions**

* Functional Testing (User authentication, checkout, payment, notifications, etc.)
* UI/UX Testing (Layout consistency, responsive design)
* Compatibility Testing (Different browsers and devices)
* Performance Testing (Load, stress, and scalability)
* Security Testing (Vulnerability assessment, authentication)
* API Testing (Validation of request-response handling)

**Test Environments**

* **Browsers:** Chrome, Firefox, Safari, Edge
* **Devices:** Windows, macOS, Android, iOS
* **Tools:** Selenium, JMeter, Postman, Burp Suite
* **Test Data:** Sample user accounts, dummy payment details

**Testing Types**

* **Functional Testing:** Validate core features, user interactions, workflows.
* **UI/UX Testing:** Ensure proper layout, responsiveness, and accessibility.
* **Compatibility Testing:** Test across browsers, devices, and OS platforms.
* **Performance Testing:** Evaluate speed, scalability, and load handling.
* **Security Testing:** Identify vulnerabilities, ensure data protection.
* **Database Testing:** Verify data integrity, queries, CRUD operations.
* **API Testing:** Validate request/response handling, integration points.
* **Regression Testing:** Re-test after updates to ensure stability.

**Defect Reporting Procedure**

* All defects will be logged in a defect tracking tool (e.g., JIRA).
* Defects will be categorized by severity and priority.
* Developers will be notified, and defects will be retested after fixes.

**Test Strategy**

* **Manual Testing:** Exploratory, functional, and usability testing.
* **Automation Testing:** Regression and API automation using Selenium & Postman.
* **Performance Testing:** Load testing using JMeter.
* **Security Testing:** Penetration testing using Burp Suite.

**Test Schedule**

|  |  |
| --- | --- |
| **Phase** | **Duration** |
| Test Planning | 1 week |
| Test Case Design | 2 weeks |
| Test Execution | 3 weeks |
| Regression Testing | 1 week |
| Test Closure | 1 week |

**Test Deliverables**

* Test Plan Document
* Test Cases & Test Scenarios
* Test Execution Reports
* Defect Reports
* Performance Test Reports
* Security Test Reports
* Final Test Summary Report

**Entry and Exit Criteria**

**Entry Criteria:**

* Requirements are finalized and approved.
* Test environment is ready.
* Test cases are written and reviewed.

**Exit Criteria:**

* All test cases executed.
* No critical defects remain open.
* Final test summary report prepared.

**Test Execution**

**Entry Criteria:**

* Test plan and test cases are approved.
* Test environment is stable.

**Exit Criteria:**

* All functional and non-functional tests completed.
* All high-priority defects are fixed and retested.

**Test Closure**

**Entry Criteria:**

* All test cases executed with acceptable pass percentage.
* All critical defects closed.

**Exit Criteria:**

* Final test summary report signed off.
* Test completion certificate issued.

**Tools**

* **Defect Tracking:** JIRA
* **Test Automation:** Selenium, Appium
* **API Testing:** Postman, Rest Assured
* **Performance Testing:** JMeter
* **Security Testing:** Burp Suite

**Risks and Mitigations**

|  |  |
| --- | --- |
| **Risk** | **Mitigation Strategy** |
| Changing requirements | Regular sync-ups with stakeholders |
| Test environment issues | Use cloud-based testing |
| Third-party service failures | Mock API testing |

**Approvals**

* **Test Lead:** [Name]
* **Project Manager:** [Name]
* **Business Stakeholder:** [Name]