**Test Plan for AB Testing Website App.vwo.com**

**Created by: Kathiravan V P**

**1. Objective**

This document outlines the test plan for the AB Testing website **App.vwo.com**. The objective is to ensure that all features and functionalities work as expected for the target audience, including digital marketers, product managers, and developers who use the platform to run AB tests on websites and mobile websites.

**2. Scope**

The scope of this test plan includes:

**Features to be tested:**

* User authentication (Login, Logout, Password Reset)
* Dashboard functionality
* Experiment creation (A/B Testing, Split Testing, Multivariate Testing)
* Goal tracking and analytics
* Integration with third-party tools (Google Analytics, Segment, etc.)
* User role and permissions management
* Performance and responsiveness across devices
* Security and data privacy compliance

**Types of testing:**

* Manual Testing
* Automated Testing
* Performance Testing
* Accessibility Testing

**Environments:**

* Different browsers (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari)
* Operating systems (Windows, macOS, Linux)
* Device types (Desktops, Laptops, Tablets, Smartphones)

**Evaluation criteria:**

* Number of defects found
* Time taken to complete testing
* User satisfaction ratings

**Team roles and responsibilities:**

* Test Lead: Define test strategy and oversee execution
* Testers: Execute test cases and report defects
* Developers: Fix reported issues
* Product Managers: Provide feature requirements
* Stakeholders: Review test reports

**3. Inclusions**

**Introduction:**

This test plan provides an overview of the testing approach for App.vwo.com, ensuring a high-quality experience for end users.

**Test Objectives:**

* Identify and fix defects before release
* Validate user experience and performance
* Ensure application stability and reliability

**4. Exclusions**

* Backend API testing (covered separately in API testing plans)
* Third-party tool functionalities beyond VWO integrations
* Internal administrative configurations

**5. Test Environments**

**Operating Systems:**

* Windows 10 & 11
* macOS Monterey and later
* Ubuntu Linux

**Browsers:**

* Google Chrome (Latest 3 versions)
* Mozilla Firefox (Latest 3 versions)
* Microsoft Edge (Latest version)
* Safari (Latest version)

**Devices:**

* Desktop Computers & Laptops
* Tablets & Smartphones (iOS, Android)

**Network Connectivity:**

* Wi-Fi, Wired, and Cellular (4G/5G) connections

**Security Protocols:**

* Multi-Factor Authentication
* Secure HTTPS connections
* User session management

**Access Permissions:**

* Admin, Editor, Viewer roles for testing user access controls

**6. Defect Reporting Procedure**

**Criteria for Identifying Defects:**

* Deviation from expected functionality
* UI/UX inconsistencies
* Performance issues (slow response time, lagging UI)
* Security vulnerabilities

**Steps for Reporting Defects:**

1. Document the defect with screenshots and logs
2. Assign severity and priority levels
3. Log the defect in JIRA (or another defect tracking tool)
4. Assign the defect to the respective developer
5. Track resolution status and retest after fixes

**Metrics Tracked:**

* Number of defects found
* Average time to resolve defects
* Percentage of defects fixed

**7. Test Strategy**

**Step 1: Test Scenarios and Test Cases Creation**

**Techniques Used:**

* Equivalence Class Partitioning
* Boundary Value Analysis
* Decision Table Testing
* State Transition Testing
* Use Case Testing

**Additional Methods:**

* Error Guessing
* Exploratory Testing

**Step 2: Testing Procedure**

1. **Smoke Testing** - Verify critical functionalities (Login, Dashboard, Experiment Creation)
2. **Detailed Functional Testing** - Execute all test cases
3. **Performance Testing** - Evaluate response time and load handling
4. **Security Testing** - Verify authentication and data protection
5. **Regression Testing** - Ensure new updates do not break existing features

**Step 3: Best Practices**

* **Context-Driven Testing** - Testing based on application scenarios
* **Shift-Left Testing** - Early-stage testing
* **Exploratory Testing** - Unscripted testing for new defects
* **End-to-End Flow Testing** - Simulate real user journeys

**8. Test Schedule**

|  |  |
| --- | --- |
| **Task** | **Duration** |
| Test Plan Creation | 3 Days |
| Test Case Creation | 5 Days |
| Test Execution | 10 Days |
| Defect Fixing & Retest | 5 Days |
| Test Summary Report | 2 Days |

**9. Test Deliverables**

* Test Plan Document
* Test Scenarios & Cases
* Defect Reports
* Test Summary Report

**10. Entry and Exit Criteria**

**Requirement Analysis:**

* **Entry:** Requirements document received
* **Exit:** Requirements clarified and signed off

**Test Execution:**

* **Entry:** Test cases approved, application ready for testing
* **Exit:** All test cases executed, defects logged and retested

**Test Closure:**

* **Entry:** Test execution complete, defect reports ready
* **Exit:** Test summary reports finalized

**11. Tools**

* **JIRA** - Bug Tracking
* **Mind Mapping Tool** - Test Strategy Planning
* **Snipping Tool** - Capturing Screenshots
* **Word/Excel** - Documentation
* **Selenium/Cypress** - Automation Testing
* **JMeter** - Performance Testing

**12. Risks and Mitigations**

|  |  |
| --- | --- |
| Risk | Mitigation Strategy |
| Resource unavailability | Assign backup resources |
| URL not accessible | Ensure backup test environment |
| Time constraints | Prioritize high-impact test cases |

**13. Approvals**

**Documents for Approval:**

* Test Plan
* Test Scenarios
* Test Cases
* Test Reports

**Approval Required From:**

* QA Manager
* Product Manager
* Development Team