Test Strategy for app.vwo.com  
Created by: Sharik Sayyad  
Version: 1.0  
Date: May 4, 2025

—

1. Introduction  
   This Test Strategy outlines the testing approach, methods, and activities for app.vwo.com. It supports the Test Plan by defining how testing will be conducted to ensure a stable, secure, performant, and user-friendly application for users conducting A/B testing experiments.

—

1. Objectives

* Validate functionality, usability, security, accessibility, and performance of key modules: Login, Dashboard, Account Creation, and AB Testing Campaigns.
* Identify critical defects early by Shift-Left Testing practices.
* Ensure cross-browser, cross-device compatibility.
* Deliver a high-quality, production-ready application on schedule.

—

1. Scope

* In-Scope:
  + Functional Testing (Login, Dashboard, Account Creation, AB Campaigns)
  + Non-Functional Testing (Performance, Accessibility, Security)
  + API Validation (Authentication, Campaign APIs)
  + Browser and Device Compatibility Testing
* Out-of-Scope:
  + Backend infrastructure validation
  + Testing third-party tool internal functionality

—

1. Testing Types and Approach  
   A. Functional Testing

* Smoke Testing after each build deployment.
* Detailed Functional Testing based on business requirements and user stories.
* Regression Testing after every release.
* End-to-End Flow Testing simulating real-world use.

B. Automation Testing

* Playwright with TypeScript for Regression and Smoke Suites.
* Critical User Journeys automated: Login, Campaign Creation, Reporting.

C. Performance Testing

* JMeter scripts simulating 1000 concurrent users.
* Target KPIs:
  + Response time < 3 seconds
  + 95% success rate under load

D. Accessibility Testing

* WCAG 2.1 AA compliance verified using Axe DevTools and Lighthouse.
* Focus Areas: Forms (Login, Create Account), Campaign Setup, Reports.

E. Security Testing

* Manual security tests covering:
  + Input Validation (SQL Injection, XSS)
  + Session Management
  + Authentication Mechanisms (SSO, OAuth)
* Basic Penetration Testing using OWASP ZAP.

F. Usability Testing

* Ease of navigation, intuitive campaign setup, clarity of results dashboards.

—

1. Test Levels

* Unit Testing (owned by Developers)
* Integration Testing (QA supports API integration validation)
* System Testing (owned by QA team)
* User Acceptance Testing (UAT) (owned by Stakeholders)

—

1. Test Design Techniques

* Boundary Value Analysis (e.g., password field length, username length)
* Equivalence Partitioning (valid vs invalid inputs)
* Decision Table Testing (login success/failure combinations)
* State Transition Testing (campaign status changes)
* Error Guessing (common UX pitfalls, race conditions)

—

1. Test Data Management

* Dummy user profiles (Admin, Editor, Viewer roles)
* Sample dummy websites for AB testing setup
* Campaigns pre-configured for quick validation
* Data anonymization where required

—

1. Environment Strategy

* Primary Testing Environments: QA, UAT
* Browsers: Chrome, Firefox, Edge, Safari (latest and N-1 versions)
* Devices: Desktops, Laptops, Tablets, Mobile (iOS & Android)
* Responsive Testing Tools: BrowserStack / local devices

—

1. Defect Management

* All defects logged in JIRA.
* Severity and Priority clearly defined.
* Daily Defect Triage Meetings.
* SLA-driven resolution timelines.

—

1. Exit Criteria

* 95% Test Cases Passed.
* All Severity 1 and Severity 2 defects fixed and verified.
* Traceability Matrix covers 100% requirements.
* Performance Benchmarks Met.
* Stakeholder Sign-off obtained.

—

1. Risks and Mitigation  
   | Risk | Mitigation |  
   |------|------------|  
   | Build instability | Smoke tests before deep testing |  
   | Resource unavailability | Cross-training team members |  
   | Browser version updates mid-cycle | Continuous monitoring, adjusting compatibility matrix |  
   | Late changes to requirements | Agile Change Management process in place |

—

1. Tools Used

* TestRail (Test Management)
* Playwright + TypeScript (Test Automation)
* JMeter (Performance Testing)
* Axe DevTools, Lighthouse (Accessibility)
* JIRA (Defect Tracking)
* Confluence (Documentation)
* Slack/Zoom (Communication)

—

1. Metrics & Reporting

* Test Execution Progress (% executed, % passed, % failed)
* Defect Density
* Test Coverage
* Requirements to Defect Mapping
* Performance Metrics
* Accessibility Compliance Score

—

1. Approval  
   | Name | Role | Approval Date |  
   |------|------|---------------|  
   | Sharik Sayyad | QA Lead | May 5, 2025 |  
   | Project Manager | Stakeholder | May 6, 2025 |