

# Test Strategy Document

**Project:** app.vwo.com

**Prepared By:** QA Lead – Juhi Danak

**Date:** 03/05/2025

**Version:** 1.0

---

## 1. Purpose

This Test Strategy document defines the approach and standards to be followed during the testing of the VWO application (app.vwo.com). It aims to ensure that all features, especially those related to A/B testing workflows, traffic segmentation, and campaign reporting, function reliably across supported environments and use cases.

---

## 2. Test Scope

### In-Scope Features:

- Login/Authentication
- User Dashboard
- Account Creation
- A/B Testing Campaign Creation & Management
- Campaign Launch and Variation Tracking
- User Segmentation Rules
- Campaign Report & Analytics Dashboard

### Out of Scope:

- Backend microservices not exposed to UI
  - API stress testing (handled by DevOps)
  - Multivariate testing (MVT) module for this phase
  - In-app product tours or guides
-

### 3. Testing Types and Coverage

Type	Description
Functional Testing	To validate UI flows, campaign behavior, report generation, and variations
Regression Testing	Automated suite run after each major code merge
Smoke Testing	Run post-deployment to check major flows (Login, Campaign launch, Reports)
Sanity Testing	Targeted testing for new features
Compatibility Testing	Across OS, browsers, screen resolutions
Accessibility Testing	WCAG 2.1 validation using tools like Axe, NVDA
Performance Testing	Response time of campaign creation, report loading
Security Testing	Login brute-force prevention, session timeout, secure cookie flags
Data Validation	Accuracy of A/B test results, segmentation, traffic split integrity
Usability Testing	Ease of campaign setup, dashboard UX, error messages

---

### 4. Test Design Techniques

- **Equivalence Partitioning** and **Boundary Value Analysis**
  - **Decision Table Testing** for segmentation rules and traffic logic
  - **State Transition Testing** for campaign states (Draft → Running → Completed)
  - **Use Case Testing** for user journeys
  - **Error Guessing** and **Exploratory Testing** for unexpected behaviors
- 

### 5. Test Environments

Layer	Tools/Setup
OS	Windows 10, 11, macOS Ventura, Ubuntu 20.04+

Browsers	Chrome (latest 3 versions), Firefox, Edge, Safari
Devices	Desktop, Laptop, Tablet (iPad), Mobile (Android, iOS Safari/Chrome)
Network	Simulate 3G, 4G, Wi-Fi; throttling via Chrome DevTools
Testing Tools	Selenium, Cypress, Postman, BrowserStack, Axe DevTools, JIRA, TestRail
Automation CI/CD	Jenkins Pipeline for test run triggers; GitHub repo for test suite

---

## 6. Roles & Responsibilities

Role	Responsibility
QA Lead	Approve test strategy, review execution, communicate with stakeholders
Test Engineers	Author test cases, execute, log defects, automate regression
Automation QA	Maintain regression suites, develop new test scripts
Dev Team	Bug fixes, code reviews, participate in triage
Product Manager	Requirement clarification, priority input on defects

---

## 7. Defect Life Cycle and Severity Matrix

Severity	Description	SLA (Fix Target)
Critical	App crash, login failure, data loss	< 8 hours
High	Broken flows, incorrect calculations	< 24 hours
Medium	UI issues, inconsistent states	< 48 hours
Low	Cosmetic/typo issues	Next sprint

All defects will be logged in **JIRA**, tagged with release version and build number.

---

## 8. Test Data Management

- **Account-Based Testing:** Use pre-created test accounts for Admin, User roles
  - **Dynamic Data:** Generated via automation scripts (UUIDs, email aliases)
  - **Data Reset:** Nightly job to reset campaign/test data for clean regression runs
- 

## 9. Test Execution Approach

### Phase-wise Breakdown:

1. **Smoke Testing** post-build deployment
2. **Functional Testing** (manual + automation)
3. **Cross-Browser and Device Testing**
4. **Performance & Accessibility**
5. **Regression Suite Execution**
6. **Final Sanity before UAT sign-off**

Parallel testing across environments will be managed using **BrowserStack Live sessions and Cypress parallel runs**.

---

## 10. Quality Metrics to Track

Metric	Target
Test Case Execution Rate	>95%
Defect Leakage (to UAT)	<2%
Automation Coverage (critical)	≥85%
Average Defect Fix Time	<24h
Failed Test Re-execution Window	<1 day

---

## 11. Compliance & Privacy Testing

- Cookie Consent Validation (GDPR Compliance)
- Data Encryption at REST & in Transit

- Secure Session Management
  - Third-party scripts validated for opt-out/opt-in
- 

## 12. Test Milestones

Activity	Date Range
Test Case Design	[Start Date] – [End Date]
Environment Setup	[Start Date]
Smoke + Functional	[Start Date] – [End Date]
Regression Testing	[Start Date]
Final Sign-Off	[Date]

---

## 13. Change Management

- Any changes in requirements must be accompanied by updated user stories and test cases.
  - Regression and sanity cycles will be executed post-change deployment.
- 

## 14. Risks and Mitigations

Risk	Mitigation Strategy
Analytics/Tracking Code Not Triggered	Add network listeners in automation
Split Traffic Not Balanced (A vs B)	Backend logs + test hooks to validate split
Limited Device Coverage	Use of BrowserStack & real devices
Unexpected UI Changes	Weekly UI review sync with dev

---

# Approval

Reviewer	Designation	Signature	Date
[QA Lead Name]	QA Lead		
[Product Owner]	Product Manager		
[Dev Lead]	Development Lead		