TEST STRATEGY DOCUMENT
Product: app.vwo.com (A/B Testing Platform)
Prepared By: Senior QA Strategist (20+ Years Experience)
Date: 10-May-2025

1. Objective

The objective of this Test Strategy is to define the testing approach, scope, resources, and deliverables to ensure the quality and readiness of app.vwo.com — an A/B testing platform. This includes validating the critical business flows, APIs, UI, performance, and data accuracy prior to production release.

2. Scope

In Scope:

- End-to-end testing of core workflows:
  - User Sign-up
  - User Login / Logout
  - Dashboard Navigation
  - A/B Test Creation, Editing, Execution
  - Analytics & Reporting for Experiments
- API Validation (Test management, User management, Goal tracking)
- UI Testing (Web - Desktop and Mobile browsers)
- Backend validation (DB integrity, Logs, Analytics calculations)
- Security & performance validations
- Cross-browser and accessibility testing

Out of Scope:

- Native mobile applications (iOS/Android)
- Admin or internal configuration dashboards
- External billing/payment systems

3. Test Objectives

- Verify functional correctness of all user and admin workflows

- Validate statistical data and result consistency of A/B tests
- Ensure system behavior under load and concurrent users
- Identify and address cross-browser inconsistencies
- Detect and mitigate critical vulnerabilities (OWASP Top 10)
- Verify API behavior, contracts, and error handling

4. Testing Types & Approach

Testing Type	**Strategy / Tools**
Functional Testing	Manual testing based on user stories & workflows
API Testing	Postman, Rest Assured (contract, CRUD, auth)
Automation (UI/API)	Selenium + TestNG (Java), API test runners
Performance Testing	JMeter, BlazeMeter (1000+ concurrent users)
Security Testing	OWASP ZAP, Burp Suite (auth, inputs, tokens)
Compatibility Testing	BrowserStack (Chrome, Firefox, Safari, Edge)
Accessibility Testing	axe-core, Lighthouse, WCAG 2.1 validations
Database Testing	SQL queries using DBeaver
Log Validation	Splunk / Kibana for backend errors and API traces
Exploratory Testing	Session-based testing (Risk-based priority)
Regression Testing	Automated suites + manual sanity round

5. Key Focus Areas

- Functional validation of login, signup, dashboard, and test creation
- Accuracy of A/B test result metrics (conversion rates, CTR, etc.)
- Error handling and validation messaging
- Session management and security (token validation, expired sessions)
- UI consistency, responsive behavior, intuitive navigation

- Backend verification of data integrity post A/B execution
- SLA compliance for performance (avg < 2s response @ 1000 users)
- Browser/device compatibility (Win/Mac/Android/iOS)
- Usability for end users (accessibility support, screen readers)

6. Test Environments

- QA/Staging Environment: with production-like data
- Tools Integration: GitHub, Jenkins, Zephyr, Jira
- Environments include all services: frontend, backend, analytics, APIs

7. Deliverables

- Test Plan & Test Case Documentation (Zephyr)
- API Test Results (Postman/Newman Reports)
- Functional Test Summary Reports
- Performance Test Report
- Security Vulnerability Report
- Compatibility Report (Browser/Device Matrix)
- Usability/Accessibility Evaluation
- UAT Sign-off Report
- Regression Test Execution Logs
- Defect Summary Report
- Final Go/No-Go Recommendation

8. Entry & Exit Criteria

Entry Criteria:

- Functional requirements/user stories signed off
- Test environments and test data ready
- APIs deployed and accessible for testing
- UI finalized with minimal design churn

Exit Criteria:

- 100% planned test cases executed
- 95%+ test case pass rate
- 100% blocker/critical issues fixed & retested
- No high/critical severity security issues
- SLA met for performance (response time & throughput)
- UAT completed with formal sign-off
- Final regression suite passes (manual + automated)

9. QA Team Structure & Timeline

Team Composition:

- 2 Functional Testers (Manual)
- 1 Test Automation Engineer
- 1 Performance Engineer
- 1 Security Consultant
- 1 Test Lead

Timeline Overview:

Week	**Activity**
Week 1–2	Functional & API Testing (Core modules)
Week 3	Performance Testing (JMeter)
Week 4	Cross-Browser & Accessibility Testing
Week 5	Regression Testing + Security Testing
Week 6	UAT, Final Defect Triage & Sign-off

10. Metrics to Track

- Test Coverage (% of requirements covered)
- Test Case Execution (% complete/pass/fail)

- Defect Density (open/closed by severity)
- Defect Leakage (defects missed in prior cycles)
- Automation Coverage (% of scenarios automated)
- API Response Success Rate & Latency
- Security Issues by Severity
- Usability Violations (per WCAG)

11. Risks & Mitigation

Risk	**Mitigation**
Rapid UI/API changes	Close alignment with dev; flexible test plans
Incomplete test data	Use mock data generators and DB scripts
Limited UAT participation	Pre-schedule and provide UAT guides
Performance degradation at scale	Early performance profiling + load simulations

12. Tools Stack

Area	**Tools**
Test Management	Jira + Zephyr
Automation	Selenium + TestNG + Java
API Testing	Postman, Rest Assured
DB Validation	DBeaver
Logs Monitoring	Kibana, Splunk
Security Testing	OWASP ZAP, Burp Suite
Performance	JMeter, BlazeMeter
Compatibility	BrowserStack
CI/CD Integration	Jenkins

Area	**Tools**
Accessibility	axe-core, Lighthouse