Test Strategy Document: apo.vwo.com

# 1. Objective

To ensure high-quality delivery of apo.vwo.com — an A/B testing platform — through structured testing that ensures functional completeness, performance resilience, security assurance, and superior user experience, across platforms.

# 2. Scope

In-Scope Functional Areas:

- Authentication: Login, Sign-up, Forgotten Password, Session handling

- Dashboard: Analytics, campaign metrics, experiment overview

- A/B Testing Page: Experiment setup, variation handling, preview, goals

- Account Management: Profile updates, password change, logout

- Third-party Integrations: Google Analytics, Segment, Hotjar, etc.

- Web & Mobile (Responsive Web Design)

Out of Scope:

- Native mobile applications (iOS/Android)

- Backend admin or internal support portals

- Deprecated APIs or features marked for removal

# 3. Test Objectives & Success Metrics

|  |  |
| --- | --- |
| Objective | Success Metric |
| Ensure functional correctness | >95% test case pass rate across modules |
| Confirm system stability under load | Stable under 500+ concurrent users |
| Validate security compliance | Zero critical/high OWASP issues |
| Ensure cross-browser/device compatibility | No major UI or behavior issues |
| Meet accessibility standards | WCAG 2.1 Level AA compliance |
| Deliver confidence for release | UAT sign-off with zero major blockers |

# 4. Test Approach

Test Types and Techniques:

|  |  |  |
| --- | --- | --- |
| Test Type | Technique(s) Used | Owner |
| Unit Testing | White-box, TDD | Dev |
| API Testing | Contract testing, JSON schema validation | QA |
| Functional Testing | BDD, Data-driven | QA |
| Regression | Automation first | QA |
| Integration Testing | Middleware, 3P Services validation | QA |
| Security Testing | Static + Dynamic Analysis | Security |
| Performance Testing | JMeter + scenario modeling | QA Perf |
| Usability Testing | A/B comparison, heuristics | UX Team, QA |
| Accessibility | Axe, NVDA, contrast checks | QA A11y |

# 5. Test Tools & Frameworks

|  |  |
| --- | --- |
| Purpose | Tools/Tech |
| Test Management | TestRail, Zephyr |
| Automation | Playwright + TypeScript / Cypress |
| CI/CD | GitHub Actions + Allure |
| API Testing | Postman, Newman, Swagger, Pact |
| Performance | Apache JMeter + BlazeMeter |
| Security | OWASP ZAP, Burp Suite, SonarQube |
| Accessibility | Axe-core, Lighthouse, NVDA, Wave |
| Defect Tracking | Jira |

# 6. Environments & Data Strategy

- Environments: Dev, QA, Staging, Pre-Prod

- Data: Synthetic test data with Faker.js, Factory Bot

- Checklist for environment readiness

# 7. Cross-Browser/Device Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Desktop Browser | Version | Mobile Browser | OS |
| Chrome | Latest, -1 | Chrome | Android 12–14 |
| Firefox | Latest, -1 | Safari | iOS 16–17 |
| Safari (macOS) | Latest | Edge | Android |
| Edge (Chromium) | Latest, -1 |  |  |

# 8. Accessibility Compliance

- WCAG 2.1 Level AA baseline

- Contrast checks, focus states, ARIA roles, screen-reader support

# 9. Deliverables

|  |  |
| --- | --- |
| Deliverable | Frequency / Timing |
| Test Strategy & Plan | Project start |
| Test Cases & Scripts | Sprint-wise |
| Daily/Weekly QA Reports | Ongoing |
| Final QA Summary Report | At release cut |
| Defect Reports | Sprint-wise + graphs |
| Automation Report | Per CI/CD trigger |
| UAT Report | Post-UAT execution |
| Security & Accessibility Audit | Release candidate stage |

# 10. Entry/Exit Criteria

Entry Criteria:

- Approved feature stories, stable builds, updated API docs

Exit Criteria:

- 100% test execution, 0 critical defects, all thresholds met

# 11. Team Structure

|  |  |  |
| --- | --- | --- |
| Role | Count | Responsibility |
| QA Lead | 1 | Strategy, planning, reporting |
| Sr. Test Engineer | 1 | Automation, complex scenarios |
| Manual QA Engineers | 2 | Functional, compatibility |
| Performance Analyst | 1 | Load, spike testing |
| Security Analyst | 1 | OWASP, penetration testing |

# 12. Test Schedule

|  |  |
| --- | --- |
| Phase | Timeframe |
| Test Planning & Strategy | Week 1 |
| Functional Testing | Week 2–4 |
| Security & API Testing | Week 3–5 |
| Performance Testing | Week 4–5 |
| Usability & Compatibility | Week 5–6 |
| Regression Testing | Week 6–7 |
| Final UAT & Release Readiness | Week 8 |

# 13. Risk Mitigation Plan

|  |  |
| --- | --- |
| Risk | Mitigation Strategy |
| Unclear requirements | Early grooming, raise RFCs |
| Unstable integrations | Use mocks, sandbox testing |
| Frequent UI changes | Use resilient selectors |
| Environment delays | Parallel test scripting |
| Performance gaps | Baseline early, simulate realistic loads |