

Test Plan

Created by : Ruksar Ibtesam

Test Plan for : app.vwo.com

Table of Content

Title	Pg. no
Objective	2
Scope	2
Features to be Tested	2
Types of Testing	2
Environments	2
Evaluation Criteria	2
Team Roles & Responsibilities	3
Inclusions	3
Exclusions	3
Test Environments	3-4
Defect Reporting Procedure	4
Defect Identification	4
Reporting Steps	4
Metrics Tracked	4
Test Strategy	4-5
Test Schedule	5
Test Deliverables	5
Entry & Exit Criteria	6
Tools	6
Risks & Mitigations	6
Approvals	6

Objective

This document outlines the test plan for app.vwo.com, an A/B testing platform that helps businesses optimize website conversions. The objective is to ensure all features function as expected, providing a seamless experience for users managing A/B tests.

Scope

Features to be Tested

- ☐ Login Page
- ☐ Forgot Password
- ☐ Create Account Page
- ☐ Dashboard Page
- ☐ A/B Testing Campaign (Creation, Execution, Reporting)

Types of Testing

- ☐ Manual Testing
- ☐ Automated Testing (Selenium/Cypress)
- ☐ Performance Testing (JMeter)
- ☐ Accessibility Testing (WCAG Compliance)

Environments

Browsers	Chrome, Firefox, Edge, Safari
OS	Windows 10/11, macOS, Linux
Devices	Desktop, Laptop, Mobile

Evaluation Criteria

- ☐ Defects found & fixed
- ☐ Test execution time
- ☐ User satisfaction (UX testing)

Test Plan

Team Roles & Responsibilities

Test Lead	Oversees test strategy & reporting
Testers	Execute test cases, log defects
Developers	Fix defects, assist in debugging
Stakeholders	Review test results

Inclusions

Introduction

- ☐ Purpose
- ☐ Scope
- ☐ Goals of Testing

Test Objectives

- ☐ Identify defects in critical workflows.
- ☐ Ensure UI consistency across devices.
- ☐ Validate A/B test creation & reporting accuracy.
- ☐ Check security (login, session management).

Exclusions

- ☐ Third-party integrations (unless critical to A/B tests).
- ☐ Backend database testing (covered separately).
- ☐ Full regression suite (only priority test cases included).

Test Environments

Category	Details
Operating Systems	Windows 10/11, macOS, Linux
Browsers	Chrome, Firefox, Edge, Safari (latest versions)

Devices	Desktop, iPhone, Android, iPad
Network	Wi-Fi, 4G/5G, Low-bandwidth simulation
Security	Password policies, Session timeout, CSRF checks
Permissions	Admin vs. User role testing

Defect Reporting Procedure

Defect Identification

- ☐ Deviations from requirements.
- ☐ UI/UX inconsistencies.
- ☐ Functional failures (e.g., broken workflows).

Reporting Steps

- ☐ Log defects in JIRA with:
 1. Steps to reproduce.
 2. Screenshots/videos.
 3. Severity (Blocker, Critical, Major, Minor).
- ☐ Triage: Daily review with dev team.
- ☐ Tracking: JIRA dashboards for metrics.

Metrics Tracked

- ☐ Defects opened/closed.
- ☐ Avg. resolution time.
- ☐ Test coverage %.

Test Strategy

Step 1: Test Design

Techniques Used

- ☐ Equivalence Partitioning (e.g., valid/invalid emails).
- ☐ Boundary Value Analysis (e.g., password length).
- ☐ Exploratory Testing (unscripted scenarios).

Test Plan

Step 2: Testing Phases

Phase	Description
Smoke Testing	Quick check on build stability
Functional Testing	Validate login, dashboard, A/B test flows
Regression Testing	Post-fix verification
Performance Testing	Load test with 500+ concurrent users

Step 3: Best Practices

- ☐ Shift-Left Testing: Early QA involvement.
- ☐ End-to-End Flows: Test user journeys (e.g., Login → Create Test → Analyze Report).

Test Schedule

Task	Duration
Test Plan Creation	2 Days
Test Case Design	5 Days
Execution	10 Days
Defect Fixing & Retest	5 Days
Final Report	1 Day

Test Deliverables

- ☐ Test Cases (Excel/TestRail).
- ☐ Defect Reports (JIRA).
- ☐ Test Summary Report.

Entry & Exit Criteria

Test Plan

Phase	Entry Criteria	Exit Criteria
Test Planning	Requirements finalized	Approved test plan
Test Execution	Build deployed to QA env.	All test cases executed.
Test Closure	All defects resolved.	Test summary shared.

Tools

- ☐ Test Management: TestRail
- ☐ Defect Tracking: JIRA
- ☐ Automation: Selenium/Cypress
- ☐ Performance: JMeter
- ☐ Accessibility: Axe

Risks & Mitigations

Risk	Mitigation
Delayed build delivery	Use mock APIs for early testing
Browser-specific defects	Prioritize cross-browser testing
Insufficient test data	Synthetic data generation tools

Approvals

Approved by

- ☐ [QA Lead Name] – [Date]
- ☐ [Product Manager Name] – [Date]