Test Plan for AB Testing Website App.vwo.com

Created by: Sushma Yadav

1. Objective

This document outlines the test plan for the AB Testing website **App.vwo.com**. The objective is to ensure that all features and functionalities work as expected for the target audience, including digital marketers, product managers, and developers who use the platform to run A/B tests on websites and mobile websites.

2. Scope

***** Features to be tested:

Feature	Description
User Authentication	Login, Logout, Password Reset
■ Dashboard Functionality	Access to insights and reports
Experiment Creation	A/B Testing, Split Testing, Multivariate Testing
o Goal Tracking & Analytics	Monitor conversion rates and engagement
Third-Party Integrations	Google Analytics, Segment, etc.
user Role & Permissions	Admin, Editor, Viewer management
Performance & Responsiveness	Across different devices
Security & Compliance	Data privacy and protection

✓ Types of Testing:

- Manual Testing
- Automated Testing
- Performance Testing +
- Accessibility Testing <a>6

Environments:

• Browsers: Chrome, Firefox, Edge, Safari

• Operating Systems: Windows, macOS, Linux

• **Devices:** Desktops, Laptops, Tablets, Smartphones

ii Evaluation Criteria:

Metric	Measurement
Number of Defects Found	Count of bugs detected
Ime to Complete Testing	Total duration taken
type User Satisfaction Ratings	Feedback from testers

Team Roles & Responsibilities:

Role	Responsibility
🌀 Test Lead	Define test strategy & oversee execution
🏂 Testers	Execute test cases & report defects
<u>Language</u> Developers	Fix reported issues
★ Product Managers	Provide feature requirements
Stakeholders	Review test reports

3. Inclusions

Q Introduction:

This test plan provides an overview of the testing approach for App.vwo.com, ensuring a high-quality experience for end users.

© Test Objectives:

Identify and fix defects before release

- Validate user experience and performance
- Ensure application stability and reliability

4. Exclusions

Solution Services Separately in API testing plans Services Separately in API testing plans Services Service

5. Test Environments

Operating Systems:

- Windows 10 & 11 =
- macOS Monterey and later
- Ubuntu Linux 💍

Browsers:

- Google Chrome (Latest 3 versions)
- Mozilla Firefox (Latest 3 versions)
- Microsoft Edge (Latest version)
- Safari (Latest version)

Devices:

- Desktop Computers & Laptops
- Tablets & Smartphones

Network Connectivity:

• Wi-Fi, Wired, and Cellular (4G/5G) connections 📶

Security Protocols:

- Multi-Factor Authentication
- Secure HTTPS connections
- User session management

Access Permissions:

• Admin, Editor, Viewer roles for testing user access controls

6. Defect Reporting Procedure

Criteria for Identifying Defects:

- Deviation from expected functionality
- UI/UX inconsistencies
- Performance issues (slow response, lagging UI)
- Security vulnerabilities 🕡

Steps for Reporting Defects:

- 1. Document the defect with screenshots and logs ion
- 2. Assign severity and priority levels 6
- 3. Log the defect in JIRA 🦰
- 4. Assign the defect to the respective developer 💄
- 5. Track resolution status and retest after fixes 😉

Metrics Tracked:

Metric	Description
Number of defects	Total bugs found
Resolution Time	Average time taken to fix
✓ Fix Rate	Percentage of defects resolved

7. **Test Strategy**

Step 1: Test Scenarios & Test Cases Creation

Techniques Used:

- Equivalence Class Partitioning
- Boundary Value Analysis
- Decision Table Testing

- State Transition Testing
- Use Case Testing

Additional Methods:

- Error Guessing
- Exploratory Testing

Step 2: Testing Procedure

- 1. **Smoke Testing** Verify critical functionalities (Login, Dashboard, Experiment Creation)
- 2. **Detailed Functional Testing** <a> Execute all test cases
- 3. **Performance Testing** \(\daggered{f} \) Evaluate response time and load handling
- 4. **Security Testing** Verify authentication and data protection
- 5. **Regression Testing** Ensure new updates do not break existing features

Step 3: Best Practices

- **Context-Driven Testing** Testing based on application scenarios
- Shift-Left Testing Early-stage testing
- **Exploratory Testing** Unscripted testing for new defects
- End-to-End Flow Testing Simulate real user journeys

8. **X** Test Schedule

Task	Duration
> Test Plan Creation	3 Days
Test Case Creation	5 Days
* Test Execution	10 Days
Defect Fixing & Retest	5 Days
Test Summary Report	2 Days

9. Test Deliverables

- 🀞 Defect Reports
- **Test Summary Report**

10. Entry and Exit Criteria

Phase	Entry Criteria	Exit Criteria
Requirement Analysis	Requirements received 🔽	Requirements signed off 🚣
Test Execution	Test cases approved 🍃	All cases executed 🎯
Test Closure	Testing complete 🔽	Summary reports finalized 📊

11. 💥 Tools

- JIRA Bug Tracking 🐞
- Mind Mapping Tool Test Strategy Planning 🧠
- Snipping Tool Capturing Screenshots 📷
- Selenium/Cypress Automation Testing
- **JMeter** Performance Testing **\(\rightarrow\$**

12. A Risks & Mitigations

Risk	Mitigation Strategy
Resource unavailability	Assign backup resources
URL not accessible	Ensure backup test environment
Ime constraints	Prioritize high-impact test cases

13. **Approvals**

Approval required from:

- A QA Manager
- Product Manager
- 📌 Development Team

XX End of Test Plan