TEST PLAN

Product Name: VWO (Frontend)

Prepared by: Rachna Maithani

Date: Dec 3, 2024

TABLE OF CONTENTS

OVERVIEW

This Test Plan document for the VWO application aims to validate the platform's ability to conduct comprehensive A/B testing, measure its impact, and provide actionable insights. Specifically, this plan will verify that VWO allows users to:

- Conduct A/B testing on any element

- Import and aggregate relevant metrics to measure experiment impact

- Track leading and lagging indicators to assess experiment effectiveness

- Monitor guardrail metrics to identify and terminate underperforming experiments

SCOPE

The following aspects of app.vwo.com will be tested:

- User interface and user experience

- Login and authentication processes

- Mobile compatibility and responsiveness

Testing Approach

The testing will involve a combination of:

- Manual testing to validate functionality and usability

- Performance testing to evaluate speed and efficiency

Test Environments

Testing will be conducted on:

- Multiple browsers (e.g. Chrome, Firefox, Safari)

- Various operating systems (e.g. Windows, macOS, Linux)

- Different device types (e.g. desktops, laptops, mobile devices)

Evaluation Criteria

The success of the testing will be evaluated based on:

- Number of defects identified and resolved

- Time taken to complete testing activities

- User satisfaction ratings and feedback

Roles and Responsibilities

The following team members will be involved in the testing:

- Test lead

- Testers

- Developers

Testing Schedule

The testing will commence on 8-12-2024 and is expected to be completed by 16-12-2024. The following milestones are planned:

- [Milestone 1: Completion of test planning]

- [Milestone 2: Completion of testing activities]

- [Milestone 3: Defect resolution and re-testing]

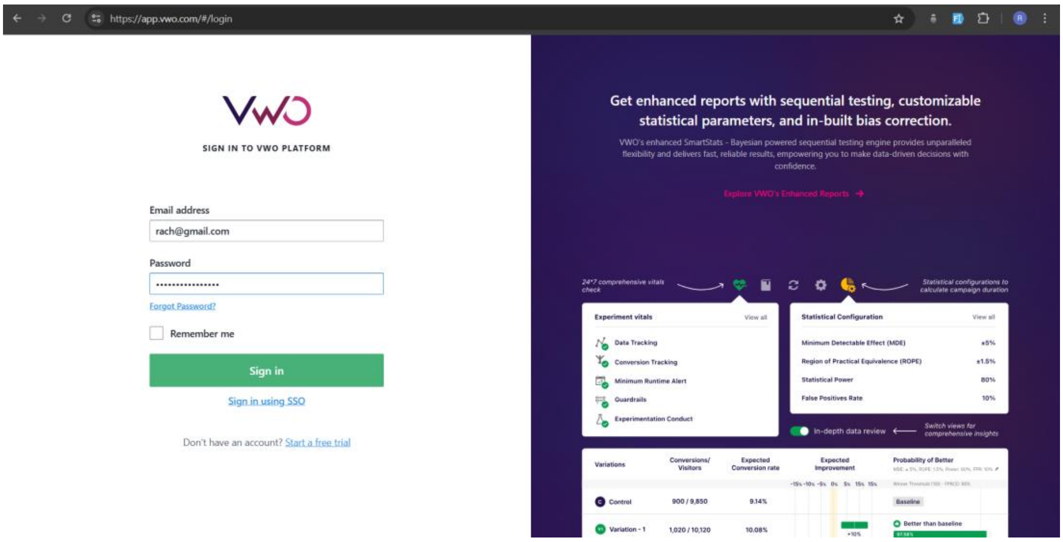
Testing Tools and Equipment

The following tools and equipment will be used for testing:

- Testing software (e.g. JIRA)

- Hardware (e.g. laptops, mobile devices)

- Documentation templates (e.g. test plans, test cases)



https://app.vwo.com/#/login

SIGN IN TO VWO PLATFORM

Email address

rach@gmail.com

Password

••••••••••••••••

Forgot Password?

□ Remember me

Sign in

Sign in using SSO

Don't have an account? Start a free trial

Get enhanced reports with sequential testing, customizable statistical parameters, and in-built bias correction.

VWO's enhanced SmartStats - Bayesian powered sequential testing engine provides unparalleled flexibility and delivers fast, reliable results, empowering you to make data-driven decisions with confidence.

Explore VWO's Enhanced Reports →

24*7 comprehensive vitals check

Statistical configurations to calculate campaign duration

Experiment vitals — View all

- Data Tracking
- Conversion Tracking
- Minimum Runtime Alert
- Guardrails
- Experimentation Conduct

Statistical Configuration	View all
Minimum Detectable Effect (MDE)	±5%
Region of Practical Equivalence (ROPE)	±1.5%
Statistical Power	80%
False Positives Rate	10%

In-depth data review ← Switch views for comprehensive insights

Variations	Conversions/ Visitors	Expected Conversion rate	Expected Improvement	Probability of Better
Control	900 / 9,850	9.14%		Baseline
Variation - 1	1,020 / 10,120	10.08%	+10%	Better than baseline

The scope of the project includes testing the following features of ‘https://demo.opencart.com/’ web application.

INCLUSIONS

- Login
- Forget Password
- Start a free Trail (Register)

Name	Env url
QA	qa.vwo.com
Pre Prod	preprod.vwo.com
UAT	uat.vwo.com
Prod	app.vwo.com

EXCLUSIONS

- All the features except that are mentioned under ‘Inclusions’
- Dashboard, Surveys and Feedback, Support Page
- Support Widget - ZOHO chat
- Test Automation

TEST STRATEGY

Our approach to functional testing is simple and structured to make sure all features are thoroughly tested.

STEP 1: Creating Test Scenarios and Test Cases

We will use different techniques to write test cases, such as:

- Equivalence Class Partitioning
- Boundary Value Analysis
- Decision Table Testing
- State Transition Testing
- Use Case Testing

Additionally, we'll use:

- Error Guessing
- Exploratory Testing

Test cases will be prioritized based on their importance to the business and the risk involved.

STEP 2: Testing Process

1. Smoke Testing:
   When we get the application, we will first test the basic and critical features (smoke testing). If these features do not work, we will reject the build and wait for a fixed version.

2. In-depth Testing:
   Once we have a stable build, we will:
   - Test the application in detail using our prepared test cases.
   - Test on multiple devices and environments at the same time.

3. Defect Reporting:
   - Report any issues (bugs) using a bug-tracking tool.
   - Share a daily email update with the development team about the bugs found and the testing progress.

4. Types of Testing We Will Perform:
   - Smoke Testing: Testing critical features.
   - Sanity Testing: Checking if minor changes work well.
   - Regression Testing: Ensuring new changes have not broken existing features.
   - Retesting: Confirming bugs are fixed.
   - Usability Testing: Checking if the app is easy to use.
   - Functionality Testing: Making sure all features work correctly.
   - UI Testing: Verifying the design and layout look good.

Testing will continue until we achieve a high-quality product.

STEP 3: Best practices to make our Testing better :

To ensure efficient testing, we will follow:

• Context Driven Testing – We will be performing Testing as per the context

of the given application.

• Shift Left Testing – We will start testing from the beginning stages of the development itself, instead of waiting for the stable build.

• Exploratory Testing – Using our expertise we will perform Exploratory

Testing, apart from the normal execution of the Test cases.

• End to End Flow Testing – We will test the end-to-end scenario which

involve multiple functionalities to simulate the end user flows.

DEFECT REPORTING PROCEDURE:

Defect Identification:
A defect is identified when there is:

- A mismatch with the requirements.
- Issues affecting user experience.
- Technical errors or system failures.

Defect Reporting:

- Use a standard template for reporting.
- Include clear steps to reproduce the issue.
- Attach screenshots, logs, or any relevant evidence.

Defect Prioritization:

- Assign severity and priority levels to each defect.
- Assign defects to the right team members for investigation and fixing.

Tools for Tracking:

- Use defect tracking or project management tools to record and monitor defects.

Team Roles:

- Testers
- Developers
- Test Lead

Communication:

- Regular updates will be shared with stakeholders through designated communication channels.

Performance Metrics:

- Number of defects found.
- Time taken to fix defects.

- Percentage of defects successfully resolved.

Defect Process	POC
New Frontend	Rachna
Backend	-
Dev Ops	-

Tools - JIRA

TEST SCHEDULE

Following is the test schedule planned for the project –

Task	Time Duration
▪ Creating Test Plan	
▪ Test Case Creation	
▪ Test Case Execution	
▪ Summary Reports Submission Date	

TEST DELIVERABLES

The following are to be delivered to the client:

Deliverables	Description	Target Completion Date
Test Plan	A document with details about the project scope, testing strategy, schedule, resources, and what we will deliver.	DATE
Functional Test Cases	A set of test cases created to cover all features in scope.	A set of test cases created to cover all features in scope.
Defect Reports	Daily reports of bugs found, including screenshots and steps to reproduce them.	NA
Summary Report	A final report showing: - Bugs by type. - Bugs by feature or area. - Bugs by priority level.	DATE

ENTRY AND EXIT CRITERIA FOR SOFTWARE TESTING LIFE CYCLE

Phase	Entry Criteria	Exit Criteria
Requirement Analysis	- Requirements Documents or project details are provided to the testing team. - Necessary resources are allocated to the testing team.	- The testing team has thoroughly reviewed and understood the requirements. - All questions or doubts regarding the requirements are clarified.
Test Planning	- Testable requirements are derived from the provided requirements or project details. - Any ambiguities in the requirements are resolved. - Resources, tools, and timelines are identified for testing.	- A comprehensive Test Plan document, including Test Strategy, is created and approved by the client.
Test Designing	- The Test Plan document is finalized and approved. - Application workflows and feature details are clearly understood. - Test data and environment requirements are identified.	- Test Scenarios and Test Cases are written, reviewed, and signed off by the client.
Test Execution	- Test Scenarios and Test Cases are finalized and signed off. - A stable application build is available for testing. - Required test environment is set up and validated.	- All Test Cases are executed, and their results are documented in Test Case Reports. - Defect Reports are created for all identified bugs, including steps to reproduce and relevant details.
Test Closure	- All Test Case Reports and Defect Reports are completed. - Outstanding defects are reviewed, and their status is communicated to stakeholders.	- A Test Summary Report, detailing the testing process and outcomes, is created and approved. - Lessons learned and recommendations for future projects are documented.

TOOLS

The following are the list of Tools we will be using in this Project:

- JIRA - Bug Tracking Tool
- Mind map Tool
- Snipping Screenshot Tool
- Word and Excel documents

RISKS AND MITIGATIONS

The following are the list of risks possible and the ways to mitigate them:

Risk: Non-Availability of a Resource

Mitigation: Backup Resource Planning

Risk: Build URL is not working

Mitigation: Resources will work on other tasks

Risk: Less time for Testing

APPROVALS

Team will send different types of documents for Client Approval like below:

- Test Plan
- Test Scenarios
- Test Cases
- Reports

Testing will only continue to the next steps once these approvals are done.