Test Plan (VWO.com)

Objective

The **Test Plan** for the **VWO application** enables A/B testing of any element while measuring its impact across various metrics.

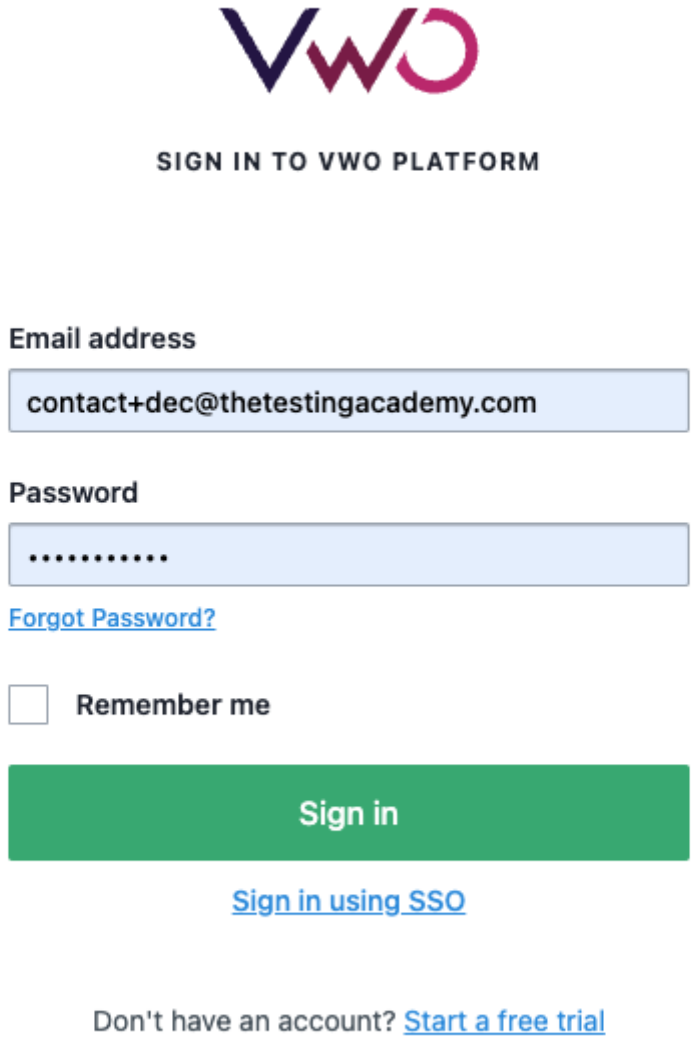
- **Import and aggregate** all relevant metrics to analyse how they are influenced by experiments.
- **Monitor leading and lagging indicators** to assess the experiment's overall impact.
- **Track guardrail metrics** to detect and halt ineffective experiments early.

Technology Stack:

- **Frontend:** React 18.2.0, jQuery 2.1.1, JavaScript
- **Database:** PostgreSQL
- **Web Server:** Apache (recommended) or Nginx

Scope

 Features and Functionality to be Tested: This includes the **user interface, checkout process, search functionality, and mobile compatibility**.
 Types of Testing: Various testing approaches such as **manual testing, automated testing, performance testing, and accessibility testing** will be conducted.
 Testing Environments: The application will be tested across **different browsers, operating systems, and device types** to ensure compatibility.
 Success Criteria: The effectiveness of testing will be measured based on **the number of defects identified, testing completion time, and user satisfaction ratings**.
 Roles and Responsibilities: The testing team will consist of **a test lead, testers, and developers**, each assigned specific responsibilities.
 Testing Schedule and Milestones: The plan will outline **start and end dates along with key testing activities** to track progress.
 Testing Tools and Resources: Various **testing software, hardware, and documentation templates** will be utilized throughout the process.

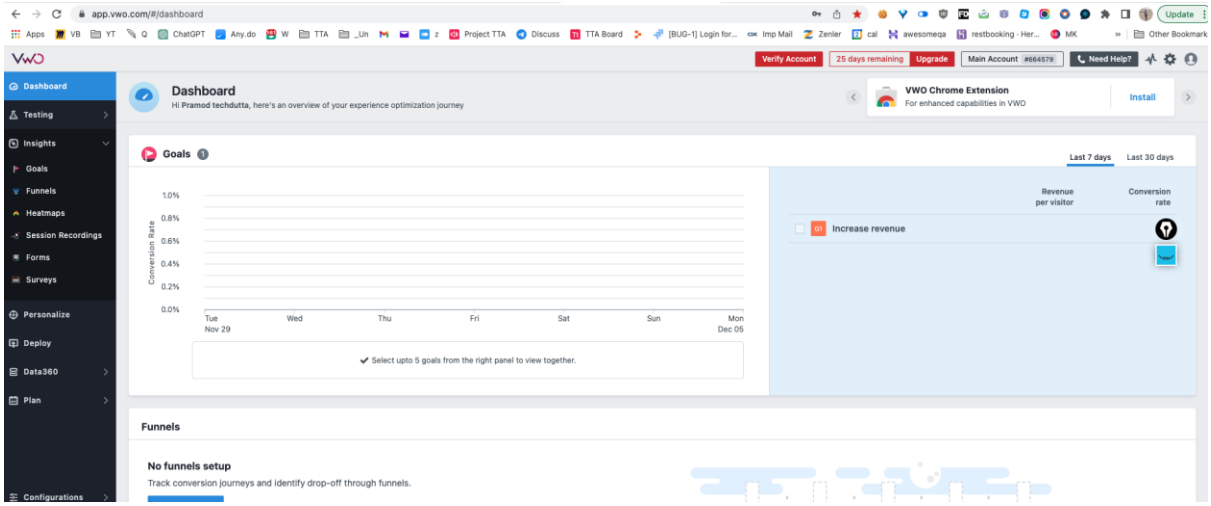


Inclusions

 Introduction: This section provides an overview of the test plan, detailing its **purpose, scope, and objectives**.
 Test Objectives: This section defines the **specific goals of testing**, such as **detecting and resolving defects, enhancing user experience, and ensuring optimal performance levels**.

- Login
- Dashboard Page
- Create Account

-

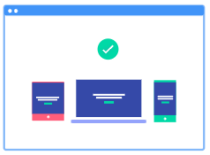


A/B Tests

Test multiple variations of your landing page and find which one converts the best

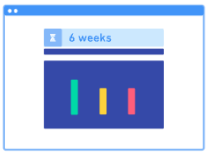
Easy-to-use Editor

Easily create tests through visual editor or use our developer-friendly code editor to build advanced tests.



Cross-platform testing

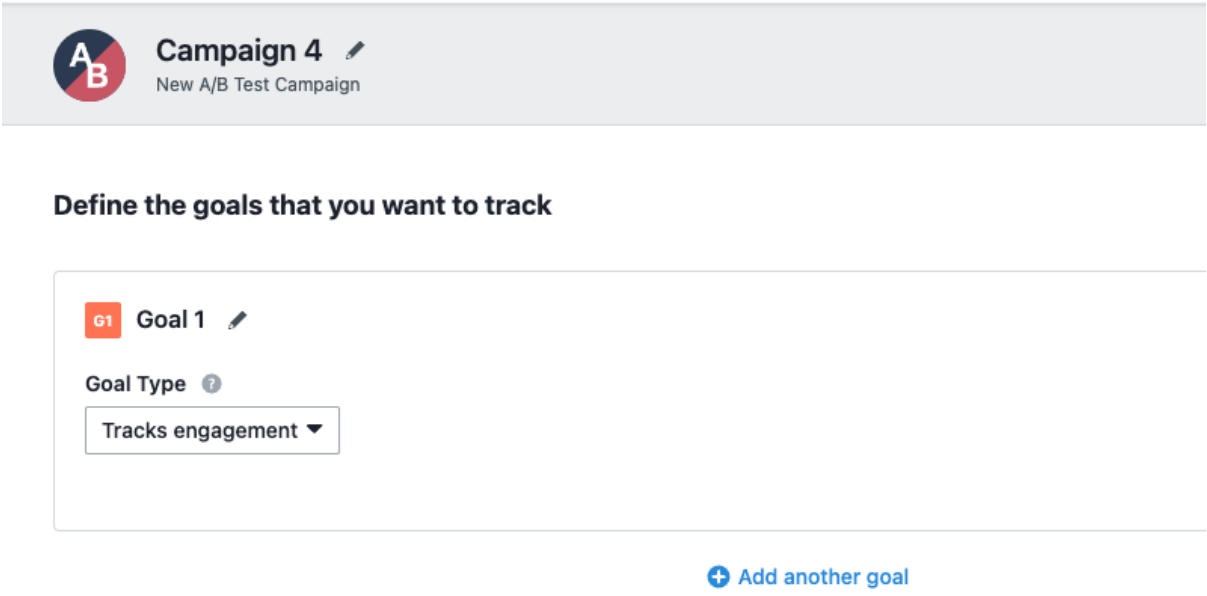
Test your website traffic across mobiles, tablets and desktops.

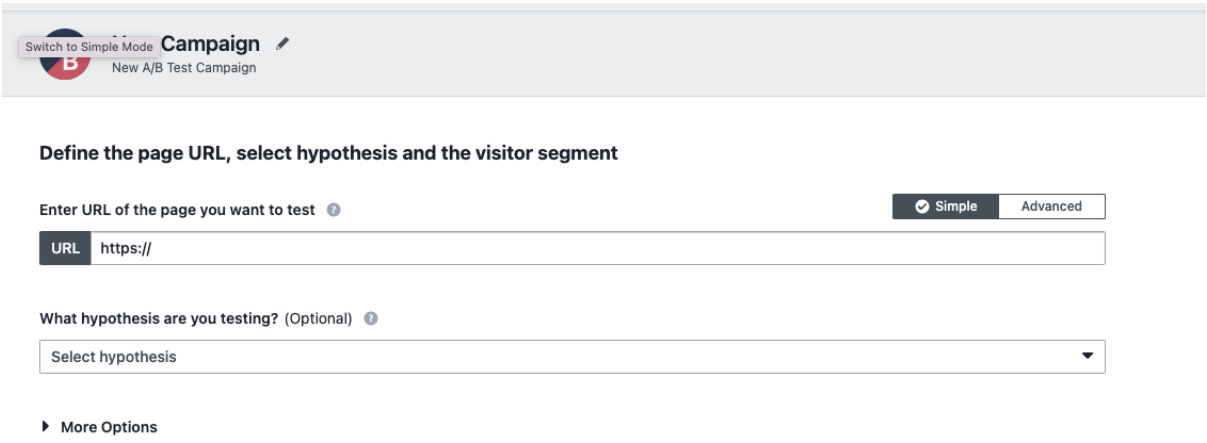


Powered by SmartStats

Bayesian statistics to provide you with accurate and transparent results.

Create an A/B Test | View Sample Report





Exclusion

1. Support Page
2. Support Widget - ZOHO chat

Test Environments

 Operating Systems: Testing will be conducted on various OS versions, including **Windows 10, macOS, and Linux**.
 Browsers: Compatibility will be tested across multiple browsers such as **Google Chrome, Mozilla Firefox, and Microsoft Edge**, along with their respective versions.
 Devices & Screen Sizes: The application will be tested on different device types, including **desktops, laptops, tablets, and smartphones**.
 Network Conditions: Various network setups, such as **Wi-Fi, cellular, and wired connections**, will be considered during testing.

 Hardware & Software Requirements: The test environment will require **specific processor capabilities, memory, and storage capacity** to run test cases efficiently.
 Security & Authentication: Access to the test environment will be secured using **passwords, tokens, or certificates** for authentication.
 User Access & Roles: Team members, including **testers, developers, and stakeholders**, will have designated **access permissions** based on their roles.

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

Windows 10 – Chrome, Firefox and Edge
• Mac OS – Safari Browser
• Android Mobile OS – Chrome
• iPhone Mobile OS - Safari

Defect Reporting Procedure

 Defect Identification Criteria: A defect will be identified based on **requirement deviations, user experience issues, or technical errors**.
 Defect Reporting Process: Issues will be logged using a **designated template**, including **detailed reproduction steps, screenshots, and logs**.
 Defect Triage & Prioritization: Defects will be classified by **severity and priority levels** and assigned to the appropriate team for resolution.
 Defect Tracking Tools: A **defect tracking or project management tool** will be used to log, monitor, and manage defects.
 Roles & Responsibilities: Team members, including **testers, developers, and the test lead**, will have specific responsibilities in defect management.
 Stakeholder Communication: Regular updates on defect status will be shared via **defined communication channels and update frequencies**.
 Defect Reporting Metrics: Effectiveness will be measured using **metrics such as defect count, resolution time, and fix success rate**.

Defect Process	**POC**
New Frontend	Devesh
Backend	Sonal
Dev Ops	Prajeeth

Tools – JIRA, ALM

Test Strategy

Step 1: Test Scenario & Test Case Creation

- The first step involves designing **test scenarios and test cases** for all features within scope.
- Various **test design techniques** will be applied, including:
    - **Equivalence Class Partitioning (ECP)**
    - **Boundary Value Analysis (BVA)**
    - **Decision Table Testing**
    - **State Transition Testing**
    - **Use Case Testing**
- Additionally, we will leverage our expertise in:
    - **Error Guessing**
    - **Exploratory Testing**
- Test cases will be prioritized based on criticality and impact.

Step 2: Testing Execution Process

- Upon receiving a testing request, we begin with **Smoke Testing** to verify the core functionalities.
- If the build fails **Smoke Testing**, it will be **rejected**, and testing will resume once a stable build is received.
- Once a stable build **passes Smoke Testing**, we conduct **in-depth functional testing** using the predefined test cases.
- Multiple test engineers will execute tests **simultaneously across multiple supported environments**.
- Any defects found will be **logged in the bug tracking tool**, and a **daily status report** will be sent to the development and management teams.
- The testing process will include:
    - **Smoke & Sanity Testing**
    - **Regression Testing & Retesting**
    - **Usability, Functional, and UI Testing**
- Testing cycles will be repeated until a **high-quality product** is achieved.

Step 3: Best Practices for Efficient Testing

- **Context-Driven Testing:** Testing will be adapted based on the specific requirements and context of the application.
- **Shift-Left Testing:** Testing will begin **early in the development cycle** to detect and address defects sooner.
- **Exploratory Testing:** Beyond scripted test execution, **ad-hoc testing** will be performed to uncover hidden defects.
- **End-to-End Testing:** Comprehensive testing of **user flows** involving multiple functionalities to replicate real-world scenarios.

Test Schedule

Following is the test schedule planned for the project –
Task Time Duration

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

2-4 Sprints to Test the Application

Test Deliverables.

The following are to be delivered to the client:

Deliverables	Description	Target Completion Date
Test Plan	Details on the scope of the Project, test strategy, test schedule, resource requirements, test deliverables and schedule	Date
Functional Test Cases	Test Cases created for the scope defined	Date
Defect Reports	Detailed description of the defects identified along with screenshots and steps to reproduce on a daily basis.	NA
Summary Reports	Summary Reports – Bugs by Bug#, Bugs by Functional Area and Bugs by Priority	Date

Entry and Exit Criteria

The below are the entry and exit criteria for every phase of Software Testing Life Cycle:

Requirement Analysis

Entry Criteria:

- The testing team receives the Requirement Documents or detailed project specifications.

Exit Criteria:

- The requirements have been thoroughly analysed and understood by the testing team.
- Any queries or ambiguities have been clarified.

Test Execution

Entry Criteria:

• **Test Scenarios and Test Cases** have been **approved** by the client.
• The **application is prepared** and ready for testing.

Exit Criteria:

• **Test Case Reports** and **Defect Reports** are documented and finalized.

Test Closure

Entry Criteria:

• **Test Case Reports** and **Defect Reports** are finalized.

Exit Criteria:

• **Test Summary Report** is prepared and documented.

Tools

The following tools will be utilized throughout the project:

- **JIRA** – Bug tracking and project management tool
- **Zephyr/Alm** – For storing the test scripts.
- **Mind Mapping Tool** – For visualizing test scenarios and strategies
- **Snipping Tool** – For capturing and sharing screenshots
- **Microsoft Word & Excel** – For documentation and reporting

Risks and Mitigations

Below are the potential risks and their corresponding mitigation approaches:

- **Risk:** Unavailability of a resource
  Mitigation: Implement **backup resource planning** to ensure continuity.
- **Risk:** Build URL is inaccessible
  Mitigation: Allocate resources to **other pending tasks** until the issue is resolved.
- **Risk:** Limited time for testing
  Mitigation: Scale up resources dynamically based on client requirements.

Approvals

The team will submit various documents for **client approval**, including:

- **Test Plan**
- **Test Scenarios**
- **Test Cases**
- **Reports**

Testing will **proceed to the next phase only after receiving the necessary approvals**.