VWO Test Plan

1. Objective

This test plan ensures the VWO application functions correctly by verifying its features, tracking key metrics, and preventing bad experiments early.

Technologies Used:

- **Frontend:** React 18.2.0, jQuery 2.1.1, JavaScript
- **Backend:** PostgreSQL, Apache (suggested), Nginx

2. Scope

This plan covers:

- ✅ **Features to be tested:** User interface, checkout process, search functionality, mobile compatibility
- ✅ **Types of testing:** Manual, automated, performance, accessibility testing
- ✅ **Test environments:** Different browsers, operating systems, and device types
- ✅ **Success criteria:** Defect count, test duration, user satisfaction ratings
- ✅ **Roles & responsibilities:** Test lead, testers, developers
- ✅ **Schedule & milestones:** Start and end dates, planned testing activities
- ✅ **Tools & equipment:** Testing software, hardware, documentation templates

3. Inclusions

✓ Core Features Tested:

- Login
- Dashboard Page
- Create Account

❌ **Exclusions:**

- Support Page
- ZOHO Chat Widget

4. Test Environments

Supported Platforms:

- 🖥️ Windows 10 – Chrome, Firefox, Edge
- 🍏 macOS – Safari
- 📱 Android – Chrome
- 📱 iPhone – Safari

Test URLs:

Environment	URL
QA	qa.vwo.com
Pre Prod	preprod.vwo.com
UAT	uat.vwo.com
Prod	app.vwo.com

5. Defect Reporting Procedure

🔍 **Defect Identification:**

- Deviation from requirements, UX issues, technical errors

📌 **Defect Logging:**

- Report using templates, detailed steps, screenshots/logs

⚡ **Defect Triage & Fixing:**

- Assign severity & priority levels
- Track in **JIRA**
- Daily defect status updates

👥 **Defect Ownership:**

Area	POC
Frontend	Devesh
Backend	Sonal
DevOps	Prajeeth

6. Test Strategy

Test Design Techniques:

✅ Equivalence Class Partitioning

✅ Boundary Value Analysis

✅ Decision Table Testing

✅ State Transition Testing

✅ Use Case Testing

Execution Flow:

1️⃣ **Smoke Testing:** Verify critical functionalities before full testing

2️⃣ **Regression & Retesting:** Ensure new changes don't break existing features

3️⃣ **Usability, Functional, UI Testing:** Validate user experience

4️⃣ **End-to-End Flow Testing:** Simulate real user journeys

Best Practices:

🚀 **Shift Left Testing:** Start testing early in development

🧐 **Exploratory Testing:** Leverage tester expertise

🎯 **Context-Driven Testing:** Align with application needs

7. Test Schedule

🗓️ **Planned Tasks & Timeline:**

- 📌 **Test Plan Creation**
- 📌 **Test Case Creation**
- 📌 **Test Execution**
- 📌 **Summary Report Submission**

🕓 **Duration:** 2 Sprints

8. Test Deliverables

The following are to be delivered to the client:

Deliverables	Description
Test Plan	Details on scope, test strategy, schedule, resources, deliverables
Functional Test Cases	Test cases created for the defined scope
Defect Reports	Detailed defects report with screenshots and steps to reproduce (daily)
Summary Reports	Overview of testing, defect statistics, and progress updates
Bug Reports	Categorization of bugs (by Bug#, Functional Area, and Priority)

9. Entry & Exit Criteria

Requirement Analysis:

✅ **Entry:** Requirements received

✅ **Exit:** Team understands requirements, clarifications resolved

Test Execution:

✅ **Entry:** Approved test cases, stable application build

✅ **Exit:** Test reports, defect logs completed

Test Closure:

✅ **Entry:** All test reports finalized

✅ **Exit:** Test summary reports submitted

10. Tools Used

🛠️ **Testing Tools:**

- JIRA (Bug Tracking)
- Mind Map Tool
- Snipping Screenshot Tool
- Word & Excel Documents

11. Risks & Mitigation Strategies

Potential Risk	Mitigation Plan
Resource unavailability	Backup resource planning
Inaccessible build URL	Prioritize alternative tasks
Limited testing time	Scale up resources dynamically

12. Approvals

✅ **Required Approvals for:**

- Test Plan
- Test Scenarios
- Test Cases
- Reports

🔔 **Testing proceeds only after client approvals.**