Test Plan for VWO Application

1. Objective

The purpose of this test plan is to ensure the functionality, usability, and performance of the VWO application's login and dashboard features. The application enables users to A/B test various elements and measure their impact. The test plan aims to validate that users can log in securely and access the dashboard seamlessly.

Key Testing Objectives:

- Verify the correctness and security of the login functionality.
- Ensure smooth navigation and data visualization on the dashboard.
- Assess performance across different environments and devices.
- Identify and address defects, ensuring a high-quality user experience.
- Maintain application stability by monitoring guardrail metrics.

2. Scope

In-Scope Features:

- **Login Page:** User authentication via email and password, error handling, session management, and security measures.
- **Dashboard Page:** User interface elements, data aggregation, navigation, performance, and metrics tracking.
- **Compatibility Testing:** Across various browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, tablet, mobile).
- **Performance Testing:** Assess loading times, response speeds, and database query efficiency.
- **Security Testing:** Verify authentication security, session management, and vulnerability assessment.
- **Usability Testing:** Evaluate user experience, intuitive navigation, and accessibility compliance.

Out-of-Scope Features:

- Payment gateway integration.
- Advanced experiment setup and analytics beyond dashboard overview.
- Non-core VWO functionalities unrelated to login and dashboard.

3. Test Strategy

3.1 Testing Types

- **Functional Testing:** Ensure that login and dashboard functionalities work as expected.
- **Regression Testing:** Verify that new updates do not break existing features.
- **Performance Testing:** Load and stress testing of the login mechanism and dashboard data aggregation.
- **Security Testing:** Validate user authentication, session expiration, and data protection.
- **Cross-Browser Testing:** Ensure compatibility with Chrome, Firefox, Safari, and Edge.
- **Accessibility Testing:** Confirm compliance with WCAG standards.

4. Test Environment

- **Front-End Technologies:** React 18.2.0, jQuery 2.1.1, JavaScript.
- **Back-End:** PostgreSQL database.
- **Web Server:** Apache (suggested) and Nginx.
- **Test Environments:**
    - Development Environment
    - Staging Environment
    - Production (Monitoring Only)

5. Test Deliverables

- Test Plan Document (this document)
- Test Cases
- Test Execution Reports
- Bug Reports
- Performance Analysis Reports
- Final Test Summary Report

6. Roles and Responsibilities

- **Test Lead:** Oversees test planning and execution, reports test status.
- **QA Engineers:** Design, execute, and document test cases.
- **Developers:** Fix identified defects and assist with debugging.
- **Product Manager:** Reviews test outcomes and defines acceptance criteria.

7. Test Schedule and Milestones

Milestone	Start Date	End Date
Test Plan Preparation	Day 1	Day 3
Test Case Creation	Day 4	Day 7
Test Execution	Day 8	Day 14
Bug Fixing and Re-testing	Day 15	Day 20
Test Summary Report	Day 21	Day 22

8. Entry and Exit Criteria

Entry Criteria:

- Development of login and dashboard is completed.
- Test environment is set up and ready.
- Test data is prepared.
- Required test tools are available.

Exit Criteria:

- All critical and high-severity defects are resolved.
- Functional and regression testing is successfully completed.
- Performance and security benchmarks are met.
- Test summary report is signed off.

9. Test Tools and Equipment

- **Test Management Tools:** Jira, TestRail
- **Automation Tools:** Selenium, Cypress (for functional automation)
- **Performance Testing Tools:** JMeter, Lighthouse
- **Security Testing Tools:** OWASP ZAP, Burp Suite
- **Browser Testing Tools:** BrowserStack, Cross-BrowserTesting

10. Test Cases (Example)

Login Page

1. Verify that a user can successfully log in with valid credentials.
2. Validate error messages for incorrect username/password.
3. Ensure session expiration works as expected.
4. Check login functionality across different browsers and devices.

5. Test SQL injection and XSS vulnerabilities in login fields.

Dashboard Page

1. Verify that the dashboard loads correctly after login.
2. Ensure all key metrics and graphs display properly.
3. Validate responsiveness and layout across different screen sizes.
4. Assess performance when handling large datasets.
5. Check that logout functionality works as expected.

11. Risk Analysis

- **Authentication failures due to security vulnerabilities.**
- **Performance issues when aggregating large amounts of data.**
- **Compatibility problems across different browsers and devices.**
- **Usability concerns affecting the user experience.**

12. Conclusion

This test plan provides a structured approach to validating the login and dashboard functionality of the VWO application. Ensuring a robust testing process will lead to improved application reliability, security, and performance, ultimately enhancing the user experience.