**Test Plan (VWO.com)**

**Test Plan for Login Functionality and Dashboard Page of app.vwo.com**

**1. Introduction**

This test plan outlines the testing strategy, scope, test scenarios, and approach for verifying the **Login Functionality** and **Dashboard Page** of [app.vwo.com](https://app.vwo.com). The objective is to ensure the reliability, security, and usability of these features.

**2. Scope**

**In-Scope**

* Testing Login Page functionality (Valid, Invalid, Edge Cases)
* Authentication and Authorization checks
* UI/UX validation of the Login and Dashboard pages
* Functional testing of Dashboard widgets, navigation, and data display
* Performance and Security testing for Login and Dashboard

**Out of Scope**

* Load Testing for multiple concurrent users
* Database performance testing

**3. Test Strategy**

**Testing Types:**

* Functional Testing
* UI/UX Testing
* Security Testing
* Performance Testing
* Compatibility Testing (across browsers and devices)

**Test Environments:**

* OS: Windows 10, macOS, Linux
* Browsers: Chrome, Firefox, Edge, Safari
* Devices: Desktop, Mobile (Responsive Testing)

**4. Test Scenarios**

**Login Functionality**

1. Verify login with valid credentials
2. Verify login with invalid credentials (wrong password/username)
3. Verify login with empty fields
4. Check case sensitivity of credentials
5. Validate error messages for incorrect login attempts
6. Verify login with special characters in username/password
7. Check session timeout behaviour
8. Validate Remember Me functionality
9. Verify account lockout after multiple failed attempts
10. Check forgot password functionality
11. Validate user redirection after successful login
12. Verify security measures (e.g., password encryption, secure connections)
13. Test login with different roles (Admin, User, Read-Only Access)
14. Verify login across different browsers
15. Check mobile responsiveness of the login page

**Dashboard Page**

1. Verify successful redirection to Dashboard after login
2. Check the presence of essential widgets and components
3. Validate data displayed in widgets against backend database
4. Verify UI responsiveness across devices
5. Test navigation between different sections
6. Check the functionality of filters, sorting, and search options
7. Validate user-specific data visibility based on roles
8. Test logout functionality and session expiry
9. Check error handling for API failures in the Dashboard
10. Verify notifications and alerts functionality

**5. Test Data**

* Sample valid user credentials (Admin, Standard User, Read-Only User)
* Invalid usernames/passwords
* Expired or locked accounts
* Various screen resolutions for UI testing

**6. Defect Tracking**

Defects will be logged in a defect tracking tool (e.g., Jira, Bugzilla) with severity and priority based on impact.

**7. Test Deliverables**

* Test Scenarios and Test Cases Document
* Test Execution Report
* Defect Report
* Test Summary Report

**8. Exit Criteria**

* All high and critical severity defects must be resolved
* Successful execution of functional and security test cases
* Approval from stakeholders

**9. Risks and Mitigation**

| **Risk** | **Mitigation Strategy** |
| --- | --- |
| Login failures due to backend issues | Coordinate with backend team for debugging |
| Browser compatibility issues | Perform cross-browser testing thoroughly |
| Security vulnerabilities | Conduct security testing and implement best practices |

**10. Conclusion**

This test plan ensures that the login functionality and dashboard page work as expected, providing a seamless user experience while maintaining security and performance standards.