VWO Login Module — Test Strategy

# 1. Introduction

This test strategy defines the approach for validating the login module of the VWO web application. It outlines the testing scope, techniques, resources, environments, and quality metrics required to ensure a stable and secure login experience.

# 2. Test Objectives

- Verify the login flow works as expected for different user types.  
- Validate the correctness of input validations and error messaging.  
- Ensure session and authentication handling complies with security best practices.  
- Confirm the UI behaves consistently across supported devices and browsers.

# 3. Testing Scope

* In Scope:
* Functional testing of login (email/password, SSO)
* UI responsiveness and accessibility checks
* Validation of error handling and session management
* Cross-browser compatibility testing
* Out of Scope:
* Deep API or infrastructure-level testing (DevOps responsibility)
* Load testing beyond 500 concurrent users

# 4. Testing Approach

- Manual functional testing using well-defined test cases.  
- Exploratory testing for UI and edge cases.  
- Use of BrowserStack for cross-browser testing.  
- Regression testing after each release build.  
- Sanity checks on staging before production push.

# 5. Test Types

- Functional Testing  
- UI/UX Validation  
- Security Testing (manual + OWASP scan)  
- Compatibility Testing  
- Smoke and Regression Testing

# 6. Roles & Responsibilities

|  |  |
| --- | --- |
| Role | Responsibility |
| QA Engineer | Write and execute test cases, log bugs, verify fixes |
| QA Lead | Review test coverage, ensure timely execution, risk tracking |
| Dev Team | Fix defects, clarify flows, support test data and logs |
| Product Manager | Provide business flow expectations and edge cases |

# 7. Tools & Resources

- TestRail for case management  
- JIRA for bug tracking  
- BrowserStack for cross-browser validation  
- OWASP ZAP for basic security checks  
- Slack for team communication

# 8. Test Metrics

- Test Case Execution Rate = (Executed / Total) \* 100  
- Pass Percentage = (Passed / Executed) \* 100  
- Defect Density = Total Defects / Total Test Cases  
- Open Defects by Severity (Sev1, Sev2, Sev3)

# 9. Risk Analysis

|  |  |  |
| --- | --- | --- |
| Risk | Impact | Mitigation |
| Unsupported browser behavior | High | Use BrowserStack in early test cycles |
| Unexpected UI regression | Medium | Include key UI flows in regression suite |
| Auth token expiration bugs | High | Test token refresh and invalid session flow |