# **Test Plan for Write Wave Application**

## **1. Introduction**

The purpose of this test plan is to outline the testing approach for the WriteWave application, a web-based collaborative writing platform. The goal is to ensure the application meets functional, usability, performance, and security requirements. This test plan provides a structured methodology for identifying, documenting, and resolving defects to enhance software quality and reliability.

## **2. Objective**

* To verify that all features of Write Wave work as intended.
* To ensure the application meets functional, performance, and security standards.
* To identify and resolve defects before release.
* To validate the user interface (UI) and overall user experience (UX).
* To ensure the application is compatible across various devices and browsers.

## **3. Scope of Testing**

* **Functional Testing:** Validates core features like user registration, login, content creation, and publishing.
* **Usability Testing:** Ensures the UI/UX is intuitive and user-friendly.
* **Performance Testing:** Tests application speed and response under varying loads.
* **Security Testing:** Identifies vulnerabilities such as unauthorized access and data leaks.
* **Compatibility Testing:** Ensures functionality across different browsers, devices, and operating systems.
* **Regression Testing:** Ensures that changes do not negatively impact existing functionalities.

## **4. Features to be Tested**

* **User Authentication:** Registration, login, password reset.
* **Content Management:** Creating, editing, saving, and deleting text documents.
* **Collaboration:** Sharing and real-time updates (if applicable).
* **UI & UX:** Navigation, responsiveness, color schemes, and accessibility.
* **Performance:** Load times, responsiveness, and stability under load.
* **Security:** Data protection, authentication, and session management.

## **5. Testing Strategy**

### **Functional Testing**

* Unit Testing for individual components.
* Integration Testing for interaction between components.
* End-to-End Testing to validate complete user journeys.

### **Usability Testing**

* Validate user interface consistency and readability.
* Verify that all links, buttons, and interactive elements work correctly.

### 

### 

### **Performance Testing**

* Load Testing with simultaneous users.
* Stress Testing by increasing requests to assess stability.

### **Security Testing**

* Authentication and authorization validation.
* SQL injection and XSS vulnerability testing.

### **Compatibility Testing**

* Cross-browser testing on Chrome, Firefox, Safari, and Edge.
* Cross-device testing on mobile (iOS, Android) and desktop.

## **6. Testing Environment Setup**

* **Operating Systems:** Windows, macOS, Linux
* **Browsers:** Chrome, Firefox, Safari, Edge
* **Devices:** Desktop, Tablet, Mobile
* **Test Data:** Sample user accounts, content drafts, invalid inputs

## **7. Test Deliverables**

* Test Plan Document
* Test Scenarios & Cases
* Test Execution Reports
* Defect Reports
* Test Summary Report

## **8. Test Schedule**

| **Phase** | **Start Date** | **End Date** |
| --- | --- | --- |
| Test Planning | 13/03/2025 | 13/03/2025 |
| Test Case Design | 13/03/2025 | 13/03/2025 |
| Test Execution | 13/03/2025 | 13/03/2025 |
| Defect Reporting & Fixing | 13/03/2025 | 13/03/2025 |
| Final Testing & Sign-off | 13/03/2025 | 13/03/2025 |

## **9. Entry & Exit Criteria**

### **Entry Criteria**

* Development team delivers a stable build.
* Required test data is available.
* Test environment is ready.

### **Exit Criteria**

* All test cases executed.
* All critical and major defects are fixed.
* Performance and security standards met.
* Stakeholders approve test results.

## 

## 

## **10. Risk & Mitigation**

| **Risk** | **Mitigation Strategy** |
| --- | --- |
| Delays in development | Align with developers for early defect identification |
| Environment issues | Maintain backup environments |
| High defect rate | Prioritize and fix critical issues first |
| Performance bottlenecks | Optimize database queries and front-end performance |

## **11. Roles & Responsibilities**

| **Role** | **Responsibility** |
| --- | --- |
| Test Manager | Oversees test plan execution and reporting |
| Test Engineers | Design, execute, and report test cases |
| Developers | Fix reported defects and collaborate on debugging |
| Product Owner | Validate application functionality and approve test results |

## 

## **12. Tools & Resources**

* **Test Management:** Jira
* **Automation Testing:** Selenium
* **Performance Testing:** JMeter
* **Cross-Browser Testing:** BrowserStack
* **Google Sheets**: For Documentation

**13. Approvals & Sign-off**

* **Stakeholders:** Project Manager, QA Lead, Development Lead
* **Sign-off Criteria:** All planned tests executed, no high-priority defects remain, stakeholders accept results.