| **Project Name –** Hack-hurdles  **Prepared By –** Vikash Kumar Bharti  **Date –** 08/10/2024  **Reviewed By:** Self-Reviewed |
| --- |

**1. Objective**The objective of this testing cycle was to evaluate the functionality, performance, usability, UI/UX, and compatibility of the **Hack-hurdles**.

This report highlights the results of test execution, key issues found, and overall recommendations for improvements.

**2. Scope of Testing**

Testing was conducted across various areas of the website, including:

* **Functional Testing**
* **Responsive Testing**
* **Compatibility Testing**
* **Usability Testing**

The testing covered multiple devices (desktop, mobile, tablet) and browsers (Chrome, Firefox, Edge).

**3. Test Strategy**

#### Testing Type

* **Functional Testing** – Validating the core features such as login, text formatting, and document saving.
* **Responsive Testing** – Ensuring layout adjusts across Desktop, Tablet, and mobile.
* **Compatibility Testing** – Ensuring the editor works on Chrome, Firefox, and Edge.
* **Usability Testing** – Verifying accessibility and user-friendliness of features like the color palette and filters.

**4. Test Execution Summary**

| **Total Test cases** | **Passed** | **Failed** |
| --- | --- | --- |
| **45** | **43** | **2** |

**5. Key Findings and Issues**

| **Issue ID** | **Category** | **Description** | **Expected Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- |
| TC\_0035 | Color Selection | Selecting an invalid color does not trigger an error prompt | Users should receive an error message for invalid colors. | No error message appears, causing user confusion. | Fail |
| TC\_0045 | Filter Application | Applying filters to unsupported formats leads to errors | Filters should only apply to supported formats or show an error. | The application crashes or behaves unexpectedly. | Fail |

**6. Performance Metrics**

| **Test Type** | **Metric** | **Result** | **Acceptable Limit** |
| --- | --- | --- | --- |
| Load Testing | Page Load Time | 1.8s | < 2 seconds |
| Functionality Test | Document Save Time | 0.8s | < 1 second |
| Error Handling | Error Message Display | 0.6s | < 0.5 seconds |
| Compatibility Test | Browser Compatibility | Pass | Pass on all browsers |
| Resource Usage | Memory Using | 150 MB | < 200MB |

|  |
| --- |

**7. Test Coverage**

### **Coverage Achieved**

**Coverage Percentage:** 95% of core functionalities were tested, including:

* User Authentication (Login/Logout)
* Document Creation and Editing
* Text Formatting Features
* Document Saving and Loading
* Image Insertion
* Filtering Options

**Areas Not Covered:**

* **API Integrations:** Functionality related to backend API calls for document management.
* **Payment Gateway:** Any payment processing features if applicable.
* **User Role Management:** Features related to user permissions and roles.

**8. Risk Assessment**

| **Risk** | **Impact** | **Mitigation** |
| --- | --- | --- |
| Integration Failures | High: Affects data retrieval and functionality | Conduct thorough integration testing and use mock APIs for initial testing. |
| Browser Compatibility | Medium: May lead to user frustration | Perform cross-browser testing regularly and fix identified issues promptly. |
| User Adoption | High: Low engagement if interface is not user-friendly | Gather user feedback and iterate on UI/UX design based on user insights. |
| Performance Issues | Medium: Slow load times can drive users away | Optimize website performance through code optimization and efficient asset loading. |
| Data Breaches | Very High: Could lead to loss of user trust and legal issues | Implement strong authentication methods, regular security audits, and data encryption. |
| Insecure Data Transmission | High: Vulnerabilities during data exchange | Use HTTPS for all data transfers and educate developers about secure coding practices. |

**9. Recommendations**

**Enhance User Interface (UI):**

* Simplify navigation by grouping related features together.
* Use consistent color schemes and fonts to improve readability.

**Improve User Onboarding:**

* Create a step-by-step tutorial for new users to help them understand key features.
* Implement tooltips to provide guidance on specific functionalities.

**Optimize Performance:**

* Regularly monitor load times and optimize images and assets for faster performance.
* Implement lazy loading for images to improve initial load times.

**Strengthen Security Measures:**

* Ensure all user data is encrypted and secure.
* Conduct regular security audits and vulnerability assessments.

**10. Next Step**

**●**  Bug Fixes: Address the critical bugs related to functionality, performance, and compatibility.

● Regression Testing: Conduct regression testing after fixes to ensure all issues are resolved.

● Additional Testing: Plan for security and API testing in the next phase.

### **Final Notes:**

This test summary report includes real bug data from my project and is structured to provide a professional and clear overview.