**Business Requirements Document (BRD) for QA Testing**

**1. Introduction**

* **Project Name:** Web Application QA Automation
* **Prepared By:** QA Team
* **Date:** [DD/MM/YYYY]
* **Version:** 1.0
* **Objective:** The purpose of this document is to define the business and functional requirements for QA testing of a web application, ensuring it meets the expected quality standards before release.

**2. Business Requirements** The application should be thoroughly tested to ensure its functionality, performance, security, and user experience meet the required standards. The key requirements include:

* Ensuring compatibility across multiple browsers such as Chrome, Firefox, and Edge, as well as different devices including desktops, tablets, and mobile phones.
* Implementing automated scripts to capture screenshots for every user click and interaction within the application.
* Verifying user authentication and access control mechanisms to prevent unauthorized access.
* Testing form submissions with valid and invalid data to validate proper error handling and data processing.
* Ensuring accurate database storage and retrieval for submitted data.
* Conducting API response validation to ensure expected data exchange between the frontend and backend.
* Performing error handling tests to confirm appropriate messages are displayed for invalid inputs.
* **Adding the sum of points 1, 2, and 3 to validate proper aggregation functionality.**
* **Adding results with point 5 from the first requirement to ensure correct calculations.**

**3. Functional Requirements**

* **User Login:** Users should be able to log in with valid credentials.
* **Form Submission:** Users should be able to submit forms with valid data.
* **Data Validation:** Submitted data should be stored correctly in the database.
* **Screenshot Capture:** Each user interaction should be logged with screenshots.
* **Error Handling:** System should provide appropriate error messages for invalid inputs.

**4. Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Result** | **Status** |
| TC\_001 | User Login - Valid Data | 1. Open the web app 2. Enter valid username/password 3. Click login | User should be logged in successfully | Pass |
| TC\_002 | User Login - Invalid Data | 1. Open the web app 2. Enter invalid username/password 3. Click login | System should display an error message | Pass |
| TC\_003 | Form Submission - Valid Data | 1. Navigate to form page 2. Fill in valid details 3. Click Submit | Form should be submitted successfully | Pass |
| TC\_004 | Form Submission - Invalid Data | 1. Navigate to form page 2. Leave required fields blank 3. Click Submit | System should display validation errors | Pass |
| TC\_005 | Screenshot Capture | 1. Perform user interactions 2. Verify screenshots are saved | Screenshots should be captured for every action | Pass |
| TC\_006 | API Response Validation | 1. Send API request 2. Verify response format and data | API should return correct response with HTTP 200 | Pass |
| TC\_007 | Sum Calculation | 1. Input values for points 1, 2, and 3 2. Perform sum operation | System should correctly add the values | Pass |
| TC\_008 | Result Addition with Point 5 | 1. Compute sum of previous values 2. Add with point 5 3. Verify final result | Final output should be correct | Pass |

**5. Assumptions & Dependencies**

* The test environment is stable and mirrors the production environment.
* Test data is available and accessible.
* Required automation tools (Selenium, Puppeteer) are installed.

**6. Conclusion**

* This document provides a structured approach to QA validation.
* Test cases ensure all functionalities are verified before release.
* Any issues found should be documented and reported for resolution.