**1. Introduction**

This test plan outlines the strategy for testing the Bug Tracking Website, which is designed to log, track, and manage bug reports in software projects. The website allows users (testers, developers, and managers) to create, view, update, and resolve bugs. The goal of this test plan is to ensure that the bug tracking system functions correctly, is user-friendly, and meets the project requirements.

**2. Test Objectives**

* Ensure that the core functionalities of bug reporting, tracking, updating, and closing bugs work as expected.
* Validate the user interface (UI) and user experience (UX) for ease of use.
* Ensure data security, particularly around user data and bug details.
* Test for cross-browser compatibility.
* Confirm that proper notifications are triggered based on changes in bug status.
* Ensure the system can handle multiple users and large volumes of bug reports.

**3. Scope**

Testing will cover:

* Functional testing of core modules (bug creation, bug status update, reporting).
* Non-functional testing, including usability, performance, and security tests.
* User roles and permissions for different types of users (Tester, Developer, Manager).
* Compatibility testing on different browsers and devices.

**4. Test Strategy**

**4.1 Types of Testing**

* **Functional Testing**: Ensuring that each feature works as per the specifications.
* **Usability Testing**: Evaluating the user interface and overall user experience.
* **Security Testing**: Testing for vulnerabilities like SQL injection, unauthorized access, etc.
* **Performance Testing**: Checking if the system performs well under various loads.
* **Cross-browser Testing**: Ensuring compatibility across Chrome, Firefox, Edge, Safari, etc.
* **Regression Testing**: Ensuring that new changes don’t break existing functionality.

**4.2 Test Approach**

* **Manual Testing**: To cover usability, exploratory, and functional testing.
* **Automation Testing**: For repetitive regression tests using tools like Selenium or Cypress.

**5. Test Environment**

* **Test Servers**: The application will be deployed on a test server for testing.
* **Browsers**: Chrome, Firefox, Safari, and Edge.
* **Operating Systems**: Windows, macOS, Linux.
* **Devices**: Desktop, Tablet, and Mobile.
* **Test Data**: Dummy bug reports with various fields like bug type, priority, status, etc.

**6. Test Cases**

Here are some examples of test cases for different modules:

**6.1 Bug Creation**

* **Test Case ID**: TC01
* **Objective**: Verify that a bug can be successfully created with valid inputs.
* **Steps**:
  1. Log in as a Tester.
  2. Navigate to the "Report Bug" page.
  3. Fill in bug details (Title, Description, Severity, Priority).
  4. Submit the bug.
* **Expected Result**: The bug is created, and the system generates a unique Bug ID.

**6.2 Bug Status Update**

* **Test Case ID**: TC02
* **Objective**: Verify that a developer can update the status of a bug.
* **Steps**:
  1. Log in as a Developer.
  2. Open an existing bug.
  3. Change the status from “Open” to “In Progress.”
  4. Save the changes.
* **Expected Result**: The status of the bug is updated successfully.

**6.3 Permission Testing**

* **Test Case ID**: TC03
* **Objective**: Verify that a Tester cannot close a bug.
* **Steps**:
  1. Log in as a Tester.
  2. Try to change the status of a bug to "Closed."
* **Expected Result**: The Tester should not have the option to close a bug.

**6.4 Search and Filter Bugs**

* **Test Case ID**: TC04
* **Objective**: Verify that users can search and filter bugs based on criteria like status, priority, or assignee.
* **Steps**:
  1. Log in as any user.
  2. Use the search/filter function on the "All Bugs" page.
  3. Search for bugs assigned to a specific developer or bugs with "High" priority.
* **Expected Result**: The search results display the correct bugs matching the criteria.

**6.5 Security Testing**

* **Test Case ID**: TC05
* **Objective**: Ensure that unauthorized users cannot access the system.
* **Steps**:
  1. Try to access the bug-tracking page without logging in.
* **Expected Result**: The user is redirected to the login page.

**7. Test Deliverables**

* Test Plan document.
* Test cases with test data.
* Test execution results (logs and reports).
* Bug reports.

**8. Schedule**

The testing will be completed over the following timeline:

* Test Planning: 2 days.
* Test Case Design: 3 days.
* Test Execution: 5 days.
* Bug Fixing and Retesting: As needed.

**9. Roles and Responsibilities**

* **Test Lead**: Responsible for planning, scheduling, and managing the testing process.
* **Testers**: Execute test cases, report bugs, and perform regression testing.
* **Developers**: Fix the reported bugs and work closely with testers during the retesting phase.

**10. Risks**

* Delays in test environment setup.
* Changes in requirements during the testing phase.
* Unavailability of resources for cross-browser and mobile testing.