Test Plan for SecureBank

1 Introduction

This test plan explains how testing will be done for the secure bank website, including the goals and areas to be tested. It serves as a guide for checking the website's features, usability, compatibility. Since the secure bank website is used for transactions, it's important to ensure it works well and offers a smooth experience for users.

- **Application overview**

  SecureBank is a financial web application designed for secure online transactions. It helps developers and security professionals learn how to identify and address security vulnerabilities while ensuring safe financial operations for its users.

- **Purpose**

  The primary purpose of this test is to ensure the quality and reliability of the SecureBank application by thoroughly evaluating its various components and features. The testing process aims to confirm that the application functions correctly, is easy to use, performs efficiently under different conditions, and follows security best practices to protect user data and financial transactions.

2 Scope

This test plan will cover all key components and functionalities of the SecureBank web application to ensure it meets the required quality standards. The areas included in this scope are as follows:

- **Functional Testing:**
  - Verify all features such as login, signup, transaction history viewing, file uploads, and search functionalities work correctly.
  - Test the ability to create, modify, and delete transactions without errors.
  - Validate that the store history and service listings function as expected.

- **UI Testing:**
  - Ensure the design layout is consistent across all pages.
  - Verify proper alignment, fonts, colors, and element placements on different screen sizes and resolutions.
  - Confirm interactive elements like buttons and links are functional and correctly labeled.

- **Performance Testing:**
  - Measure the system's response time under normal and high user load conditions.

    - Conduct stress testing to determine system behavior under extreme traffic conditions.
    - Test how the application handles bulk data uploads and large transaction histories.
- **Compatibility Testing:**
    - Verify the application’s compatibility across major browsers including Chrome, Firefox, and Edge.
    - Confirm the application works seamlessly across devices such as desktops, tablets, and mobile phones.
- **Regression Testing:**
    - Ensure that bug fixes and new features do not break existing functionalities.
    - Re-run critical test cases after every major update or change.
- **Global Components Testing:**
    - Verify elements like the header, footer, and navigation panel for consistency across all pages.
    - Ensure the copyright information, links, and branding are uniform throughout the application.

This scope ensures the SecureBank application is safe, easy to use, and works efficiently for its users.

3 Assumptions and Risks:

The test plan assumes that all requirements are clearly defined, test data is accurate and sufficient, and the test environment mirrors production. It also expects collaboration between teams for defect resolution and availability of necessary testing resources.

Several risks could impact the testing process. Changes in project scope or requirements may affect the testing timeline. Limited access to third-party services and APIs could delay integration testing, while insufficient test data may result in incomplete coverage of test scenarios. There is also a risk that certain security vulnerabilities may go undetected due to limited penetration testing resources. Browser and device compatibility issues may arise if certain configurations are untested, and delays in bug fixing and retesting cycles could impact the overall project schedule.

4 Test Approach:

The test approach will follow a phased process starting with requirement analysis, test planning, and test case creation. Testing will be conducted in cycles, covering functional, security, performance, and compatibility aspects. Defects will be logged and tracked to closure. Regression testing will be performed after fixes, and no automation will be for this stage. Testing progress will be regularly monitored and reported to stakeholders.

4.1 Test Automation

Automated unit tests are part of the development process, but no automated functional tests are planned at this time.

5 Test Environment

The test environment for SecureBank focuses entirely on manual testing. Testing will be conducted on browsers such as Chrome, Firefox, and Edge to ensure compatibility across different platforms. The testing will primarily focus on verifying the core functionalities of the application, ensuring that it works smoothly and securely across the specified browsers. Database testing has not been performed yet, and no API testing or automation is involved in this phase.

Behavioral Questions:

- Can you describe a challenging bug you found and how you resolved it?
- Have you ever disagreed with a developer about a defect? How did you handle the situation?
- How do you ensure the quality of a product when working under tight deadlines?
- What steps do you take to improve your testing skills?
- How do you manage multiple testing projects simultaneously?

Scenario-Based Questions:

- How would you test a login page for an application?
- How do you test an e-commerce website for functionality and performance? What is your approach to testing APIs?

How would you test a search feature in an application?
You have limited time to test a release. How do you prioritize your tasks?