**Testing Documentation for FinanceMan Dashboard**

## **Project Title**

FinanceMan Dashboard

## **Introduction**

The FinanceMan Dashboard is a web-based financial management tool that enables users to efficiently manage and monitor their finances. The dashboard allows users to track financial data, such as account balances, income, expenses, and transaction history. It also provides key features like a sidebar menu for navigation, a responsive design that adapts to different screen sizes, a search functionality for transaction filtering, and pagination for transaction history.

The objective of this testing documentation is to outline the structure, features, and testing phases performed to ensure the FinanceMan Dashboard meets its functional, usability, and reliability requirements. This document covers the key components tested, the testing approach, and how the features of the dashboard were verified for different devices and browsers.

## **Project Type**

Testing/Verification

## **Deployed App**

Frontend:https://finance-man.netlify.app/dashboard

## 

## 

## **Directory Structure**

/ProjectDirectory

├── /TestCases # Contains detailed test cases

│ ├── functional\_tests.xlsx # Test cases for functionalities like login, data display, etc.

│ └── negative\_tests.xlsx # Test cases for negative scenarios like invalid input

├── /TestReports # Contains all reports after test execution

│ ├── TestExecutionReport.docx

│ ├── TestSummaryReport.docx

│ └── BugDefectReport.docx

├── /Screenshots # Screenshots from test execution, validations, and responses

│ └── screenshot1.png

├── /TestPlan # Test plan documents

│ ├── TestPlan\_v1.0.docx

│ └── TestSchedule.docx

├── /TestData # Sample data used for testing (user data, transactions)

│ ├── SampleUserData.xlsx

│ └── SampleTransactions.xlsx

└── /Scripts # Scripts used to execute automated testing (if applicable)

## 

## **Features**

1. **Dashboard Page Load**
   * **Description:** Verifies that the dashboard loads successfully without any errors.
   * **Expected Outcome:** The page should load with the dashboard content visible and without any issues.
2. **Header Branding & Title**
   * **Description:** Verifies the correct display of the title and branding in the header.
   * **Expected Outcome:** The header should display the appropriate title (e.g., "Finance Dashboard") and branding/logo.
3. **Sidebar Menu Items**
   * **Description:** Verifies the display and functionality of the sidebar menu.
   * **Expected Outcome:** Menu items like "Dashboard," "Transactions," "Settings," etc., should be present and functional.
4. **Financial Data Display**
   * **Description:** Ensures the dashboard displays the user's financial data correctly.
   * **Expected Outcome:** The financial data should be accurate, such as balance, expenses, and income, as per the logged-in user.
5. **Transaction History**
   * **Description:** Verifies the correct population and display of transaction history.
   * **Expected Outcome:** Transaction records should be shown with accurate information like date, amount, and transaction type.
6. **Search Functionality**
   * **Description:** Tests the search bar in the transaction history section to filter results based on keywords.
   * **Expected Outcome:** The search should return relevant transaction results based on the keyword.
7. **Pagination Controls**
   * **Description:** Verifies that pagination works correctly with multiple pages of transaction data.
   * **Expected Outcome:** Pagination controls should function as expected (e.g., "Next" and "Previous" buttons), showing up to 10 items per page.
8. **Logout Functionality**
   * **Description:** Ensures the user can log out and is redirected to the login page.
   * **Expected Outcome:** The user should be logged out successfully and directed to the login page.
9. **Responsiveness**
   * **Description:** Tests the responsiveness of the dashboard across different screen sizes (desktop, tablet, mobile).
   * **Expected Outcome:** The dashboard layout should adjust and display correctly across different devices.
10. **Error Handling for Invalid Input**
    * **Description:** Verifies that the system correctly handles invalid input scenarios.
    * **Expected Outcome:** The system should display an error message when invalid data (e.g., missing fields, negative amount) is entered.

### **Test Strategy Overview**

1. **Test Types:**
   * **Functional Testing:** Focused on verifying the correctness of all features such as the dashboard, sidebar, transaction history, and more.
   * **Negative Testing:** Ensures that the system behaves as expected when given invalid input or erroneous conditions.
   * **Responsiveness Testing:** Verifies the application works smoothly across multiple device screen sizes and browsers.
2. **Testing Methodology:**
   * **Manual Testing:** All functional and negative test cases were executed manually.
   * **Cross-Browser Testing:** Ensured compatibility on Google Chrome, Mozilla Firefox, and Microsoft Edge.
   * **Device Testing:** The dashboard was tested across desktop, tablet, and mobile devices to ensure proper responsiveness.
3. **Tools Used:**
   * **Test Management:** Test cases and execution tracking were managed using TestRail.
   * **Bug Tracking:** Jira was used for tracking any defects and issues.
   * **Browsers:** Google Chrome, Mozilla Firefox, and Microsoft Edge were used for testing.
4. **Test Data:**
   * Sample data was provided for user accounts and transactions, ensuring realistic and relevant test scenarios were used.

### **Test Execution Details**

* **Test Execution Phase:** All test cases were executed from **2024-12-16 to 2024-12-18** across different browsers and devices.
* **Test Execution Results:** 90% pass rate across all functional and negative tests. 1 - No issues were identified during the testing cycle.
* **Test Coverage:** The tests covered all core functionalities including the dashboard page load, menu items, search functionality, transaction history, and responsiveness across devices.

### **Conclusion**

The testing was successfully completed with 1 - defect or failures. The FinanceMan Dashboard met all functional, usability, and performance requirements. The application is not ready for deployment,

The majority of the test cases passed successfully.

* TC011, which verifies the functionality of several buttons (Delete, Save, No Sort, Sort by Date, and Sort by Amount), failed due to bugs causing the buttons not to trigger any action or display an error message. and the following steps include moving forward with production deployment, followed by user acceptance testing (UAT).

### **Next Steps**

1. **Production Deployment:** Since 90% all test cases passed, But the application is not ready to be deployed to production. because TC011, which verifies the functionality of several buttons (Delete, Save, No Sort, Sort by Date, and Sort by Amount), failed due to bugs causing the buttons not to trigger any action or display an error message.
2. **Regression Testing:** Will be conducted for future releases to ensure no existing functionality is affected by new features or changes.
3. **User Acceptance Testing (UAT):** Will be carried out post-deployment to ensure the dashboard meets user expectations.

#### **Approval**

| **Name** | **Role** | **Date** | **Signature** |
| --- | --- | --- | --- |
| Stephen Chitaranjan | Test Lead | 2024-12-13 | Stephen Chitaranjan |
| Biju | Project Manager | 2024-12-13 | Biju |