
Detailed Test Plan Document

1. Introduction The purpose of this test plan is to outline the testing strategies, objectives, resources, schedule, and scope of testing for the FinanceMan web application. The primary focus of the application is on the functionalities related to Sign-Up, Login, and Dashboard/Transactions, ensuring that all features work as intended and meet the business requirements.

2. Scope This document covers the following functionalities of the FinanceMan application:

- **Sign-Up Page**
 - **Login Page**
 - **Dashboard Page** (including transaction management)
-

3. Test Objectives The objectives of testing include:

- Validating the correctness and functionality of the Sign-Up, Login, and Dashboard features.
 - Ensuring proper validation mechanisms are in place for user inputs.
 - Confirming compatibility with multiple browsers and devices.
 - Verifying error handling and security mechanisms.
-

4. Test Scope Inclusions and Exclusions

In Scope:

- Functional testing for all features of the Sign-Up, Login, and Dashboard pages.
- Validation testing for all mandatory and optional fields.
- User interface testing for responsiveness.
- API validation (if applicable).
- Browser compatibility testing (Chrome, Firefox, Safari, Edge).

Out of Scope:

- Performance and load testing.
- Testing for mobile applications (if not a web-based app).
- Backend testing beyond API verification.

5. Test Approach

- **Manual Testing:** For functional testing, validations, and UI responsiveness.
 - **Automation Testing:** (Optional) Automating repetitive test cases for login, sign-up, and dashboard functionalities.
 - **Exploratory Testing:** To identify issues not covered in test cases.
 - **Regression Testing:** To ensure new changes do not affect existing functionalities.
-

6. Test Environment

- **Test Environment Details:**
 - Operating Systems: Windows, macOS.
 - Browsers: Chrome (latest), Firefox (latest), Safari (latest), Edge (latest).
 - Devices: Desktop and laptop (responsiveness checks).
 - Test Data: Sample names, emails, passwords, and CSV files.
-

7. Test Scenarios and Test Cases

Below are the detailed test scenarios:

Sign-Up Page:

1. Verify the Sign-Up form loads successfully.
2. Verify all input fields are present (Name, Email, Password, Confirm Password).
3. Verify user can successfully sign up with valid data.
4. Verify validation for mandatory fields when left blank.
5. Verify validation for invalid email format.
6. Verify password and confirm password mismatch triggers error.
7. Verify minimum password length validation.
8. Verify proper error message displays for already registered email.
9. Verify successful account creation redirects to the login page.

Login Page:

1. Verify the Login form loads successfully.
2. Verify the presence of Email and Password fields.
3. Verify user can log in with valid credentials.

-
4. Verify validation for blank Email or Password fields.
 5. Verify validation for incorrect email format.
 6. Verify error message for invalid credentials.
 7. Verify the "Login with Google" functionality.
 8. Verify "Click here" link redirects to the Sign-Up page.

Dashboard Page (Transactions):

1. Verify dashboard initializes correctly with no transactions.
 2. Verify adding income updates Total Income and Current Balance.
 3. Verify adding expense updates Total Expenses and Current Balance.
 4. Verify current balance calculation (Total Income - Total Expenses).
 5. Verify "No Transactions Available" message when no records exist.
 6. Verify sorting transactions by Date.
 7. Verify sorting transactions by Amount.
 8. Verify search functionality filters transactions by name.
 9. Verify exporting transactions to a CSV file.
 10. Verify importing transactions from a valid CSV file.
 11. Verify importing invalid CSV files triggers an error.
 12. Verify transaction details (amount, description, date) display correctly.
 13. Verify deleting transactions updates the Current Balance.
 14. Verify editing an existing transaction.
 15. Verify description field has a maximum character limit.
 16. Verify invalid transaction inputs trigger appropriate error messages.
 17. Verify "No Sort" resets transactions to the default order.
-

8. Test Deliverables

- Test Scenarios and Test Cases document.
 - Test Execution Report (Pass/Fail Status).
 - Bug Report (if applicable).
 - Summary Report.
-

9. Entry and Exit Criteria

Entry Criteria:

- Development is complete and application is deployed in the test environment.
- Test data is available.

-
- Test cases are reviewed and approved.

Exit Criteria:

- All planned test cases are executed.
 - Critical and high-priority bugs are resolved and verified.
 - Test execution report is prepared and shared.
-

10. Test Schedule

- **Test Planning:** 1 day.
 - **Test Case Creation:** 3 days.
 - **Test Execution:** 5 days.
 - **Bug Verification and Regression Testing:** 2 days.
 - **Final Report Preparation:** 1 day.
-

11. Roles and Responsibilities

- **Test Lead:** Oversee the entire testing process, review test cases, and ensure adherence to timelines.
 - **Test Engineer:** Design and execute test cases, log bugs, and retest fixes.
 - **Developer:** Resolve bugs identified during testing.
 - **Project Manager:** Ensure alignment with project goals and deadlines.
-

12. Risks and Mitigation

Risks:

- Delay in receiving the test environment.
- Lack of proper documentation for business requirements.
- High number of defects found late in the testing cycle.

Mitigation Plan:

- Proactively communicate with the development team for environment readiness.
 - Conduct requirement clarification meetings early.
 - Focus on early testing of high-risk areas.
-

13. Tools

- Test Management Tool: JIRA/TestRail.
 - Bug Tracking Tool: JIRA.
 - Browser Compatibility Tool: BrowserStack.
 - Automation Tool (optional): Selenium/Cypress.
-

14. Approvals

- This test plan must be reviewed and approved by:
 - Project Manager
 - Test Lead
 - Development Lead