Manual Test Plan for Scopex Money

1. Introduction

This document outlines the manual testing approach for the **Scopex Money** web application, focusing on the following functionalities:

- **User Registration**
- **Adding a Recipient**
- **Logout**

The goal is to validate these features against the defined acceptance criteria and ensure they meet business requirements before proceeding with automation testing.

2. Test Approach

The testing approach includes:

- Identifying **key test scenarios** for each functionality.
- Defining **test data requirements** and **preconditions**.
- Validating **expected outcomes** against actual results.
- Testing across multiple **browsers** and **Windows versions**.
- Documenting **assumptions**, **risks**, and **constraints**.

3. Scope of Testing

The testing scope includes:

- **Functional Testing**: Validating core features (registration, adding a recipient, logout).
- **UI/UX Testing**: Ensuring consistency and responsiveness across different screen resolutions.
- **Input Validation**: Testing with valid and invalid inputs to verify error handling.
- **Cross-Browser Testing**: Ensuring compatibility across Chrome, Edge, and Firefox.
- **Exploratory Testing**: Identifying issues beyond predefined test cases.

4. Testing Methodology

The following testing techniques will be applied:

- **Positive Testing**: Validating functionality with valid inputs.

- **Negative Testing**: Testing with invalid inputs to ensure proper error handling.

- **Boundary Value Analysis**: Testing edge cases for input fields.

- **Usability Testing**: Evaluating user experience and ease of use.

- **Regression Testing**: Ensuring new changes do not break existing functionality.

5. Test Execution Process

1. **Environment Setup**: Configure the test environment (Windows 10/11, browsers, network conditions).

2. **Test Data Preparation**: Prepare valid and invalid test data for each scenario.

3. **Test Case Execution**: Execute test cases and log results.

4. **Defect Logging**: Document defects with details (steps to reproduce, expected vs. actual results, severity).

5. **Retesting**: Verify resolved defects and conduct regression testing.

6. **Reporting**: Summarize test execution results and share with stakeholders.

6. Test Environment

- **Platform**: Windows 10, Windows 11

- **Browsers**: Chrome (latest), Edge (latest), Firefox (latest)

- **Network**: Wi-Fi, Ethernet

- **Test Data**:

  - **User Registration**: Valid and invalid email addresses, passwords,First Name,Last Name.

  - **Adding a Recipient**: Valid and invalid recipient names, account numbers, bank details,Country.

  - **Logout**: Precondition – user must be logged in.

7. Assumptions & Constraints

- **Assumptions**:

  - The application is stable and accessible for testing.

  - Test data is accurate and available.
  - Authentication services are functional.
- **Constraints**:
  - Limited access to specific devices or OS versions.
  - Time constraints for test execution.

8. Defect Management

- **Defect Logging**: Defects will be logged in tracking tool.
- **Defect Details**:
  - **Bug ID**: Unique identifier.
  - **Title**: Brief description of the issue.
  - **Steps to Reproduce**: Detailed steps to recreate the issue.
  - **Expected Result**: What should happen.
  - **Actual Result**: What actually happens.
  - **Severity**: Critical, High, Medium, Low.
  - **Screenshots/Video**: Attach visual evidence.
- **Defect Resolution**: Assign bugs to developers, retest fixes, and close resolved issues.

9. Test Completion Criteria

Testing will be considered complete when:

- All test cases have been executed and meet expected results.
- Critical and high-priority defects are resolved and retested.
- No major functionality or usability issues remain.
- Test results are approved by stakeholders.

10. Acceptance Criteria

- **User Registration**:
  - Successful registration redirects the user to the Home.
  - Duplicate email addresses are not allowed.

- **Adding a Recipient**:
  - Recipient is added successfully and displayed in the recipient list.
  - Invalid recipient details trigger appropriate error messages.
- **Logout**:
  - User is logged out and redirected to the login screen.
  - Session is terminated, and the user cannot navigate back to the Home.

11. Risks & Mitigation

- **Risk**: Network issues may disrupt testing.
  - **Mitigation**: Test under stable network conditions.
- **Risk**: Time constraints may limit test coverage.
  - **Mitigation**: Prioritize critical test cases and focus on high-risk areas.

12. Conclusion

This document provides a structured approach to validate the **User Registration**, **Adding a Recipient**, and **Logout** functionalities of the Scopex Money web application. Once approved, the validated scenarios will serve as the foundation for automation testing.