**1. Introduction**

This test plan outlines the approach for testing the **Para Bnak Website** based on the functional requirements and the provided user stories for **User Login**, **Transfer Funds Between Accounts**, **View Transaction History and Open a New Account.**

The focus is on **functional testing** of API endpoints, UI behavior ensuring compliance with acceptance criteria, and identifying bugs and **non-functional testing** of website performance.

**2. Test Scope**

**In-Scope:**

**US1: User Login**

**Description:**  
As a registered user, I want to log in to my account so that I can access my banking dashboard.

**Test Coverage:**

* Validate successful login with correct username and password.
* Verify login failure with invalid credentials and that an appropriate error message is displayed.
* Confirm that the session remains active until logout or idle timeout.
* Check that the UI displays the logged-in user's name on the dashboard.

**US2: Transfer Funds Between Accounts**

**Description:**  
As a user, I want to transfer funds between my accounts so that I can manage my money efficiently.

**Test Coverage:**

* Verify selection of source and destination accounts.
* Validate transfer with a valid amount and ensure accurate balance updates.
* Ensure an error message appears when attempting a transfer with insufficient funds.
* Confirm that the successful transaction appears instantly in the transaction history.

**US3: View Transaction History**

**Description:**  
As a user, I want to view my transaction history so that I can track my spending and verify transactions.

**Test Coverage:**

* Ensure this feature is accessible only after successful login.
* Validate that the transaction list displays: date, description, amount, and updated balance.
* Check that transactions are ordered with the most recent first.
* Display appropriate message if no transactions exist.

**US4: Open a New Account**

**Description:**  
As a user, I want to open a new account so that I can create separate accounts for different purposes.

**Test Coverage:**

* Verify user can select the account type (checking/savings).
* Confirm that a new account is successfully created and listed under “Accounts Overview”.
* Validate that a confirmation message is displayed after creation.
* Ensure the new account has the default opening balance.
* Confirm the user can transfer funds into the newly created account.

**3. Test Approach**

* **Manual API Testing** using **Postman**
* **Automation Testing** using **Selenium**
* **Performance testing** using **jMeter.**
* **Black-box Testing**
* **Positive & Negative Test Cases** (valid/invalid inputs)
* **Edge Cases** (boundary conditions, error handling)

**4. Test Environment**

|  |  |
| --- | --- |
| **Application Under Test** | Online Banking System (ParaBank) |

|  |  |
| --- | --- |
| **Test Type** | Functional, API, Automation, Performance, and Black-box Testing |

|  |  |
| --- | --- |
| **Frontend Browser** | Google Chrome (latest stable version) |

|  |  |
| --- | --- |
| **Operating System** | Windows 10 / 11 or Ubuntu (depending on team setup) |

|  |  |
| --- | --- |
| **Automation Tool** | Selenium WebDriver with TestNG (Java-based) |

|  |  |
| --- | --- |
| **API Testing Tool** | Postman |

|  |  |
| --- | --- |
| **Performance Testing Tool** | Apache JMeter |

|  |  |
| --- | --- |
| **Build Tool** | Apache Maven |

|  |  |
| --- | --- |
| **Test Management Tool** | (Optional) Excel, Jira, TestRail, or similar |

|  |  |
| --- | --- |
| **Version Control** | Git (optional, if source code/scripts are version-controlled) |

**5. Test Cases**

All test cases will be documented and managed in a structured format using **Google Sheets** for easy access, collaboration, and tracking. Each test case will include:

* Test Case ID
* User Story Reference
* Test Description
* Preconditions
* Test Steps
* Expected Result
* Actual Result
* Status (Pass/Fail)
* Remarks (if any)

The sheet will cover all functional flows, edge cases, positive and negative scenarios aligned with the acceptance criteria of each user story.

**6. Bug Reporting Format**

If a bug is found, document it as follows:

**Bug ID**: BUG-001  
**Description**: Issue details will be explained   
**Steps to Reproduce**:

1. Step by step on what is done to reproduce the bug once more.

**Expected Result**: what should be happened.  
**Actual Result**: what is actually happened.  
**Severity**: how important this bug from the technical perspective.

**7. Exit Criteria**

All test cases executed  
 All critical bugs reported  
 Documentation completed (Test Plan, Bug Reports)