**Test Plan for Para Bank & Restful Booker Websites**

1. **Introduction**

This test plan is designed for the Para Bank website, a demo banking system used for training in manual and automated testing. The system allows users to perform various banking operations such as account creation, deposits, withdrawals, transfers, and viewing transaction history. The testing approach will include **Black Box Testing** techniques, focusing on **functional, performance, security, and compatibility** testing. Since no formal SRS (Software Requirements Specification) is provided, exploratory testing will be conducted to identify key functionalities and potential risks.

Additionally, **API testing** will be performed on the **Restful Booker** website to validate its endpoints, ensuring proper functionality, response accuracy, and performance. This will include testing **CRUD operations (Create, Read, Update, Delete), authentication mechanisms, error handling, response times, and security vulnerabilities**. API testing will be conducted using tools such as **Postman, JMeter, and RestAssured** to verify the reliability and robustness of the API.

1. **Scope of Testing**

**2.1 Admin Functionals:**

* Login
* Initialize Database
* Clean Database
* Start JMS Service
* Shut Down JMS Service
* Select Data Access Mode (SOAP, REST XML, REST JSON, JDBC)
* Set SOAP API Endpoint
* Set REST API Endpoint
* Set Loan Processor Service Endpoint
* Configure Initial Account Balance
* Configure Minimum Account Balance
* Select Loan Provider
* Select Loan Processor Method
* Set Loan Approval Threshold

**2.2 User Functionals:**

* Login
* Account Creation
* Edit Account Details
* Deposit Money Into The Account
* Transfer Money To Another User
* Withdraw Money
* View Transaction History And Account Details
* Check Account Balance
* OTP Verification During Sensitive Operations
* Password Reset Process
* Handling Incorrect Data Input
* Notifications And Messages For Successful Transactions

**2.3 Performance Testing:**

* **Load Testing:** Simulate multiple users and transactions to monitor system behavior under expected load.
* **Stress Testing:** Evaluate system performance under extreme conditions (e.g., high user load, large transactions).
* **Scalability Testing:** Check system scalability by adding resources and increasing transaction volumes.
* **Endurance Testing:** Monitor system stability over extended periods (e.g., 24-48 hours).
* **Spike Testing:** Test system response to sudden spikes in user activity.

**2.4 Compatibility Testing:**

* **Browser Compatibility:** Test website functionality across browsers (Chrome, Firefox, Safari, Edge).
* **Device & OS Compatibility:** Ensure the website works on different devices (desktop, mobile, tablet) and operating systems.
* **Network Compatibility:** Ensure the website performs well under different network conditions.
* **Screen Resolution Compatibility:** Ensure the website displays correctly on various screen resolutions.