**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 Rest Assured** to verify the reliability and robustness of the API.

**2. Scope of Testing**

**2.1 Admin Functionalities:**

* 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
* Selecting Loan Processor Method
* Set Loan Approval Threshold

**2.2 User Functionalities:**

* Login
* Account Creation
* Edit Account Details
* Depositing 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
  + Entering incorrect credentials
  + Leaving the required fields empty
  + Entering invalid data types (e.g., letters in numeric fields)
* **Session Timeout Handling:** Ensure users are logged out after a period of inactivity

**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).
* **Endurance Testing:** Monitor system stability over extended periods (e.g., 24-48 hours). ##
* **Spike Testing:** Test system response to sudden spikes in user activity. ##
* **Response Time Testing:** Measure the time taken for key operations (e.g., login, money transfer, account updates).

**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.

**2.5 Security Testing**

* **Authentication:** Test login and API authentication for vulnerabilities like SQL injection and weak passwords.
* **Session Management:** Ensure session timeout and secure handling of web and API sessions.
* **Authorization:** Verify users can only access allowed features and data.
* **Data Security:** Ensure sensitive data is encrypted and protected from SQL injection and XSS.
* **API Security:** Test API endpoints and tokens for security risks.
* **Error Handling:** Ensure error messages do not expose sensitive information.
* **Security Logs:** Verify logging of security-related events.

**2.6 Automation Testing:**

* **Booking Management:** Automate creating, retrieving, updating, and deleting bookings.
* **Account Management:** Automate creating, viewing, updating, and deleting accounts.
* **Authentication:** Automate login, token generation, and validation.
* **Transactions:** Automate deposit, withdrawal, and transfer operations.
* **Search and Filter:** Automate search and filter functions for bookings and transactions.

**2.7 Regression Testing**

* **Restful Booker:** Automate regression tests for bookings, authentication, and search/filter.
* **Para Bank:** Automate regression tests for accounts, transactions, and authentication.
* **Common:** Ensure new changes do not break existing functionality.

**2.8 Roles:**

* **User Functional: Gehad, Menna , Rewan**
* **Security Testing: Gehad**
* **Admin Functional Testing:**
* **Compatibility Testing: Menna, Gehad**
* **Automation Testing:**
* **Risk Management:**
* **KPI Tracking:**

**3. Risks**

**3.1 Payment Processing Failure:** Payments may fail or be duplicated due to network issues or API errors.

**3.2 Unauthorized Access:** Weak authentication allows hackers to access customer accounts.

* 1. **System Downtime:** Unexpected downtime can disrupt transactions.
  2. **Regulatory Violation:** The system fails to comply with financial regulations.

**4. KPIs**

**4.1Functional Testing KPIs:**

* Test Case Pass Rate – Measures the percentage of successfully passed test cases.
  + - Formula: (Passed Test Cases / Total Test Cases) \* 100
* 4.2Defect Detection Rate – Indicates the number of defects identified per test cycle.
  + - Formula: (Defects Found / Total Test Cases Executed) \* 100
* Critical Defect Percentage – Tracks the percentage of defects categorized as high/critical severity.
  + - Formula: (Critical Defects / Total Defects) \* 100

**4.2Performance Testing KPIs:**

* Transaction Response Time – Measures the average response time for key banking operations (e.g., fund transfers, withdrawals).
  + Unit: Milliseconds (ms) / Seconds (s)
* System Uptime – Tracks the percentage of time the system remains operational and available.
  + Formula: (System Uptime / Total Time) \* 100

**4.3Security Testing KPIs:**

* Number of Security Vulnerabilities – Counts the total security issues detected (e.g., SQL Injection, XSS).
* Time to Fix Security Issues – Measures the average time required to resolve identified security vulnerabilities.
  + Unit: Hours / Days

**4.4User Experience Testing KPIs:**

* Usability Issues Reported – Tracks the number of UI/UX-related issues affecting the customer experience.
* Error Rate – Measures the percentage of failed transactions due to system issues.
  + Formula: (Failed Transactions / Total Transactions) \* 100

**5. Reporting:**

* **Test Reports:** 
  + Document test results using tools like TestRail or JIRA.
* **Defect Reports:**
  + Log defects with detailed steps to reproduce, screenshots, and logs.
* **KPI Tracking:** 
  + Regularly update KPIs and share progress with the team.

**6. Tools:**

* **Functional Testing:** Selenium.
* **API Testing:** Postman, SoapUI
* **Test Management:** JIRA

**7. Conclusion**

This test plan ensures comprehensive coverage of functional, performance, security, and compatibility testing for the Para Bank website. It also includes specific scenarios for training in manual and automated testing, making it suitable for both beginners and experienced testers.