User Guide: Banking System with React & Node.js

1. Introduction

This document provides a step-by-step guide on using the Asset-x banking system with a React frontend and a Node.js backend. Users will learn how to sign up, manage accounts, and perform transactions securely.

2. Setting Up the Project

To get started, install Node.js and create a React project:

1. Install Node.js from https://nodejs.org/

2. Create a new React project: `npx create-react-app bank-frontend`

3. Set up a backend: `mkdir bank-backend && cd bank-backend && npm init -y`

3. User Signup & Account Creation

Users must first sign up and create an account. The backend API provides an endpoint:-

`POST /api/signup` - Creates a new account.

- `POST /api/login` - Authenticates users and returns a token.

4. Performing Transactions

Once logged in, users can perform the following transactions:-

`POST /api/deposit` - Deposit money into an account.

- `POST /api/withdraw` - Withdraw money from an account.-

`GET /api/balance` - Check account balance.

5. Testing the System

Users can test the banking system using tools like Postman to make API requests. Ensure correct authentication by including the token in requests.

6. Security & Best Practices

For security, users should:

- Use strong passwords.

- Enable two-factor authentication (2FA).

- Avoid sharing login credentials.

7. ER-Diagram

8. User Flow

1. The user signs up and creates an account.

2. Upon registration, the system assigns an account to the user.

3. The user can deposit money using the **Deposit API**.

4. The user can withdraw money using the **Withdraw API**.

5. The user can check their balance using the **Balance API**.

9. Key Features

1. **User API** - Allows users to sign up and authenticate. 2.

Account API - Creates an account for a registered user. 3.

Deposit API - Deposits money into an account.

4. **Withdraw API** - Withdraws money from an account. 5.

Balance API - Fetches the current balance of an account.