**Assignment Task: Develop a Flash USDT/BTC Wallet Transaction System**

**Task Description:**

Develop a blockchain-based wallet transaction system that allows **Flash USDT/BTC transactions** to any wallet address on **TRC20 (TRON Network)**. The flash transactions should have a **20-day validation period** before expiring. The system should also **verify and validate transactions** with Binance or another blockchain explorer.

**Task Requirements:**

**1. Wallet Generation & Management**

* Implement **wallet generation** for **USDT (TRC20), BTC, and ETH**.
* Use **TRON (TRC20)** network for USDT transactions.
* Provide **private keys & public addresses** securely.

**2. Flash Transaction System (Temporary Transactions)**

* Allow users to **send & receive** USDT/BTC.
* Transactions should appear as valid for **20 days** before **expiring**.
* Transactions should be recorded in the system but should not be **permanently confirmed on the blockchain**.

**3. Transaction Verification & Validation**

* Validate transactions using blockchain explorers (**TronScan, BTC Explorer, Etherscan**).
* Implement a system to check if a transaction is still valid within the **20-day period**.
* If 20 days pass, the transaction should become invalid.

**4. Transaction History & Logs**

* Store and display **transaction history** (sent, received, pending, expired).
* Each transaction should include:
  + Sender & receiver address
  + Transaction amount
  + Transaction ID
  + Status (**Valid, Pending, Expired**)

**5. Integration with Binance or Other Exchanges**

* Allow users to verify transactions with **Binance API** or other blockchain explorers.
* Provide **real-time transaction tracking**.

**6. Security Measures**

* Use **secure encryption** for private keys.
* Implement **2FA authentication** for transactions.
* Ensure **anti-replay attacks** are prevented.

**Tech Stack Recommendations:**

* **Blockchain**: TRON (for USDT TRC20), Bitcoin, Ethereum
* **Backend**: Node.js, Python (Flask or FastAPI)
* **Database**: MongoDB, PostgreSQL
* **API Integration**: Binance API, TronScan API, Blockcypher for BTC
* **Frontend**: React.js or Flutter (if mobile app)

**Expected Deliverables:**

✔ **Source Code** (GitHub/ZIP)  
✔ **Documentation** (Setup Guide, APIs Used, Architecture)  
✔ **Demo Video** (Screen Recording of Flash Transactions)  
✔ **Testing Report** (How validation works for 20 days)