How to Use the MiniBlockchain Program

Sara JAN

Start the Program:

- Open your terminal or command prompt.
- Navigate to the directory where the Python script miniBlockChain_RSA.py is located.
- Run the script using python miniBlockChain_RSA.py.

You will see the following menu:

```
Mini Blockchain
1. Add Transaction
2. Mine Pending Transactions
3. Print Blockchain
4. Validate Blockchain
5. Exit
Enter your choice:
```

Step 1: Add a Transaction

Choose option 1 to add a transaction.

- Enter the sender (e.g., Alice).
- Enter the receiver (e.g., Bob).
- Enter the amount (e.g., 10).

```
Enter your choice: 1
Enter sender (Alice/Bob): Alice
Enter receiver (Alice/Bob): Bob
Enter amount: 10
Transaction added: Alice sends 10.00 to Bob | Signature: 7
f33a464ffebd61d1854e8a20c1ee0df60177c450cf63d82f6a48a471b7
bb42444a27529f51e91b54d47a77c4848b06179aacb94e6e0fea027a42
ee3090cffc3
Alice's temp balance: 90.00
Alice's balance: 100.00
Bob's balance: 50.00
```

The program will verify the transaction using RSA encryption to ensure the correct sender and receiver. Once the verification is successful, the transaction will be added to the pending transactions, and the temporary balance of the sender will be displayed. It will be during the minning that the temporary balance of the sender will be the real one, same as for the receiver.

Step 2: Mine Pending Transactions

- Choose option 2 to mine the pending transactions.
- The program will create a new block containing the pending transactions.

You will see a confirmation message indicating that the transactions have been mined into a new block.

```
Enter your choice: 2
Transaction verified: Alice sends 10.00 to Bob
Nonce found: 1109
Block 1 added to the blockchain.
Pending transactions mined into a new block.
Alice's balance: 90.00
Bob's balance: 60.00
```

Step 3: Print the Blockchain

- Choose option 3 to print the blockchain.
- The program will display the current state of the blockchain, including the genesis block and any blocks created by mining transactions.

```
Block 0 [Hash: a9b921b4344de357ae0cd537936826fb032c6000311c6c3620f239f9649f1873
]
Data: Genesis Block
Previous Hash: 0
Timestamp: 1733147080
Nonce: 0

Block 1 [Hash: 00084bb4bee4b0dc71ce36d1495f273ed97e7492310069482dabcb3ef1566fdf
]
Data: Alice sends 10 to Bob
Previous Hash: a9b921b4344de357ae0cd537936826fb032c6000311c6c3620f239f9649f1873
Timestamp: 1733147466
Nonce: 8493
```

You will see the genesis block and the blocks created by mining the transactions added in the previous steps.

Step 4: Validate the Blockchain

- Choose option 4 to validate the blockchain.
- The program will verify the integrity of the blockchain and confirm that all blocks are valid.

You will see a confirmation message indicating that the blockchain is valid.