Project Description

This project is a basic blockchain written in Python to understand the importance of blockchain technology through hashing, blocks, mining (proof-of-work) and chain checking.

Main Concepts Introduced

1. **Block A structure for a block consists of**

an index (where it is in the chain),

a timestamp (when it was created),

data (what is within the block which can be anything) as well as

the previous hash (the hash of the block before it in order to keep the integrity of the chain).

The nonce (how many times the algorithm has been run to mine for a block) and its hash (the sha256 of the block).

2. **Hashing**

Block hashes are created using a built-in method called hashlib.sha256() for Python. The hash is reimplemented when the nonce changes during the mining process to achieve the difficulty required. This keeps data intact because if you ever change something in a block, it will change its hash.

3. **Proof-of-Work**

Mining is represented by incrementing the nonce until the final sha256 of the block starts with predetermined integers (like "00"). This means that it will take a certain amount of time/computational effort to mine a block which shows that it's verified.

4. **The Blockchain**

The blockchain itself is essentially a list of blocks in Python with methods, including:

make_genesis(): to make the first entry in the chain,

add_block(data): to add more entries into the chain and mine those and

check_chain(): to check the integrity of the chain from start to finish.

5. **Validation**

This blockchain checks that the previous hash in one block matches up to the hash created in another block, that the hashing of blocks meets the difficulty requirement and that all hashes rehashed from stored ones meet expected changes.

6. **How it Shows Tampering**

The program indicates how the chain becomes broken if someone tampers with it. If someone changes data in a block, check_chain() will show how everything is disconnected, showing one of the elements of blockchain technology (that it's immovable).

7. **Purpose**

The purpose of this project is for educational reasons to understand how blockchain works without third party libraries, how blocks are linked and confirmed and how mining works through calculation hashes.

```
C:\Users\jozef\Documents\univ\blockchain\mini-blockchain>git pull
Already up to date.

C:\Users\jozef\Documents\univ\blockchain\mini-blockchain>python blockchain.py
Mining block 1...
Block 1 mined: 007089f9dc5a26864cb4db4c503a6f731056a83802d50dfc9abcf9de858e496
d
Mining block 2...
Block 2 mined: 00c314e44a16c02f55767f78b74174e316f3e09b82c55ea019e3dd6ce0ec637
3
Mining block 3...
Block 3 mined: 0030f64c47250b8c7f2f78de6bb295eb5f6db9322f76130d06a3bb9958850ad
e

Full chain:
{'idx': 0, 'ts': 1763820438.6434307, 'dat': 'Genesis Block', 'prev_hash': '0',
 'nonce': 0, 'hash': 'fbf9438824b09150e48b39228f1f3e04a42075eca9a5cbcdf57f057b
c1b26edc'}
{'idx': 1, 'ts': 1763820438.6434484, 'dat': 'First real block', 'prev_hash': '
fbf9438824b09150e48b39228f1f3e04a42075eca9a5cbcdf57f057bc1b26edc', 'nonce': 96
, 'hash': '007089f9dc5a26864cb4db4c503a6f731056a83802d50dfc9abcf9de858e496d'}
{'idx': 2, 'ts': 1763820438.643823, 'dat': 'Another block', 'prev_hash': '0070
89f9dc5a26864cb4db4c503a6f731056a83802d50dfc9abcf9de858e496d', 'nonce': 399, '
hash': '00c314e44a16c02f55767f78b74174e316f3e09b82c55ea019e3dd6ce0ec6373'}
{'idx': 3, 'ts': 1763820438.6447244, 'dat': 'Hello Blockchain', 'prev_hash': '
00c314e44a16c02f55767f78b74174e316f3e09b82c55ea019e3dd6ce0ec6373', 'nonce': 33
, 'hash': '0030f64c47250b8c7f2f78de6bb295eb5f6db9322f76130d06a3bb9958850ade'}
Checking chain...
Chain is good ✅

Tampering with block 2...
Checking chain...
Difficulty not met at block 1!

C:\Users\jozef\Documents\univ\blockchain\mini-blockchain>
```