



GlobalBoost Coding Hacks

13. Implement a Simple Blockchain in Python for Educational Prototyping

Why: Understanding blockchain internals aids in auditing or customizing protocols. A basic implementation demonstrates proof-of-work, hashing, and chaining, valuable for developers entering Web3 in 2026, where custom chains like layer-2 solutions are prevalent.

How to Implement: Define Block class with hash, add chaining in Blockchain class, and mine with difficulty. Use for simulations or teaching, extending with Merkle trees for realism.

```
python
import hashlib
import time

class Block:
    def __init__(self, index, previous_hash, timestamp, data, nonce=0):
        self.index = index
        self.previous_hash = previous_hash
        self.timestamp = timestamp
        self.data = data
        self.nonce = nonce
        self.hash = self.calculate_hash()

    def calculate_hash(self):
        return
        hashlib.sha256(f"{self.index}{self.previous_hash}{self.timestamp}{self.data}{self.nonce}".encode()).hexdigest()

class Blockchain:
    def __init__(self):
        self.chain = [self.create_genesis_block()]

    def create_genesis_block(self):
        return Block(0, "0", time.time(), "Genesis Block")
```



GlobalBoost Coding Hacks

```
def add_block(self, data):
    previous_block = self.chain[-1]
    new_block = Block(len(self.chain), previous_block.hash, time.time(), data)
    self.mine_block(new_block)
    self.chain.append(new_block)

def mine_block(self, block, difficulty=4):
    while block.hash[:difficulty] != "0" * difficulty:
        block.nonce += 1
        block.hash = block.calculate_hash()

# Hack: Usage
bc = Blockchain()
bc.add_block("Transaction 1")
bc.add_block("Transaction 2")
for block in bc.chain:
    print(f"Block {block.index}: {block.hash}")
```

Analysis: This models Bitcoin's core without network layers, mining in seconds for small difficulties. Extensible to PoS; in educational contexts, it clarifies immutability, with SHA-256 ensuring tamper-resistance, prepping devs for tools like Solidity.