

Block-Chain Lab 2

In this lab exercise, you will implement a simple blockchain from scratch using a Python programming language. The blockchain will consist of 8 transactions. Your task is to create the necessary data structures and functions to build and validate the blockchain.

Requirements

Block Structure: Define a block structure that includes the following fields:

Index: The index of the block in the blockchain.

Timestamp: The time when the block was created.

Previous Hash: The hash of the previous block in the chain.

Transactions: A list of transactions included in the block.

Nonce: A value that will be adjusted to meet the proof-of-work requirement.

Hash: The hash of the block's contents.

Transactions: Create 8 sample transactions with the following format:

Sender: The sender's address.

Receiver: The receiver's address.

Amount: The amount of currency transferred.

Genesis Block: Create a genesis block (the first block) with arbitrary values for the fields.

Hash Function: Implement a hash function that takes a block's contents and produces a hash value. You can use a simple hash algorithm like SHA-256.

Proof of Work (PoW): Implement a proof-of-work mechanism. The nonce value should be adjusted to find a hash that starts with a specified number of leading zeros (difficulty). You can start with a low difficulty level.

Block Validation: Create a function to validate if a block's hash meets the proof-of-work requirement and if the previous block's hash matches the hash of the previous block.

Blockchain Creation: Construct the blockchain by linking blocks together. Ensure that each new block is correctly linked to the previous one.

Blockchain Validation: Implement a function to validate the entire blockchain. This function should check if each block's hash is valid, if the previous hash matches, and if the proof-of-work requirement is met.

Your Task

Your task is to write the necessary code to implement the requirements outlined above. Start by defining the block structure, creating sample transactions, and implementing the blockchain creation process. Finally, validate the entire blockchain to ensure its integrity.

Remember, this is a simplified implementation for educational purposes, and real-world blockchains have many additional features and complexities.