**Reflection (~300 words)**

Working on this mini blockchain gave me practical insight into the fundamental ideas behind blockchain technology. I learned how hashing is used to guarantee immutability: once a block is mined, its hash depends on all of its contents (including transactions, timestamp, and nonce). If even a small change is made to the block, the hash changes completely. Because every block also stores the previous block’s hash, changing one block breaks the entire chain. This illustrates how blockchains achieve immutability and why tampering is detectable.

I also saw how Proof-of-Work adds security. Mining requires finding a hash with a certain number of leading zeros, which can only be achieved through trial and error by incrementing the nonce. This makes it computationally expensive to modify blocks. If someone changes a transaction, they must re-mine that block and every block after it. As the chain grows, this becomes nearly impossible in practice, which is why Proof-of-Work provides strong protection against tampering and ensures trust without a central authority.

What surprised me was how little code is needed to demonstrate these concepts. In less than a hundred lines of JavaScript, I was able to create a simple blockchain, mine blocks, and validate integrity. It was striking to see how altering one number in a transaction caused the entire chain to become invalid. This exercise made me appreciate the elegant simplicity behind blockchain’s design, while also understanding why real-world I also saw how Proof-of-Work adds security. Mining requires finding a hash with a certain number of leading zeros, which can only be achieved through trial and error by incrementing the nonce. This makes it computationally expensive to modify blocks. If someone changes a transaction, they must re-mine that block and every block after it. As the chain grows, this becomes nearly impossible in practice, which is why Proof-of-Work provides strong protection against tampering and ensures trust without a central authority.