**Blockchain using Node.js and Socket.io**

Blockchain is a decentralized trust based system to transfer and verify the data sent over a network.

It is a decentralized system in which a record of transactions made in bitcoin or another cryptocurrency are maintained across several computers that are linked in a peer-to-peer network.

Blockchain’s impact on traditional businesses has been swift, as it has already disrupted a lot of fields like finance, medicine, logistics, education and artificial intelligence (AI).

Blockchains are meant to be decentralized and secure on a P2P network, so we’ll need to implement a hashing algorithm.

**Why hashing over other security methods?**

Hashing allows us to input any size data and always get in return a string of a determined length, thus making it very easy to work with our data since the same input will always return the same output.

**Blockchain 101**

First off Blockchain is not Cryptocurrency.

Blockchain is the technology behind Cryptocurrencies.

An analogy would be how Internet is the technology behind Twitter.

It is the process of maintaining a distributed digital ledger which contains a series of records that are linked together and cannot be corrupted.

Let's take a look at the components of a blockchain.

**Components of a Blockchain**

Block: It is a collection of data usually capped at an upper limit either by the size or number of data units.

Chain: It is a collection of blocks that are linked together by an algorithm that uses the information of the previous block.

Node: A system in the blockchain that is used to verify the authenticity of blocks as well as maintain the digital ledger that keeps record of all the blocks in chronological order.

**Advanced Blockchain;**

The above are only components that make up the blockchain.

What truly makes the blockchain secure are the processes that are involved.

**Mining a block**

Mining is process of generating a new block and adding it to the network.

It involves two processes;

**Proof of Work Generation:**

As mentioned earlier, blocks are linked using an algorithm that makes use of the information of the previous block to generate the next block.

This algorithm is known as proof of work, generally designed in such a way that output is hard to generate, but fairly easy to verify its output.

**Proof of Work Verification:**

When one of the nodes in the network successfully generates the block, the rest of nodes in the network must verify the authenticity of the proof and chain.

So the nodes verify whether the proof is right and checks the authenticity of the chain. If everything is alright, the block is mined and all other nodes update their ledger to include the newly mined block.

**Event Flow of Blockchain**

Let's look at how the events unfold when you add data to a blockchain.

A request to exchange data is received from the sender to one of the nodes in the chain.

The node then broadcasts to other nodes about the incoming data and adds it to the current transaction pool.

Once the limit of the block is reached (size or number of units), the nodes start mining the block.

The nodes compete against each other to find a proof of work solution.

When one of the nodes succeeds in mining it broadcasts the solution.

The other nodes then verify the output and check whether it’s valid.

Then they verify the blocks of the chain and add the newly mined block.

**Requirements:**

This app is built in Node.js.

We’ll also be making use of the sha256 npm package to hash the blocks.

If you’re new to Node.js, first create a new directory, create an index.js file, and then from your terminal, run npm init -y from within your new directory. This creates a package.json file, which will allow us to install the sha256 package.

Run npm i sha256 to install the sha256 library.

If you’re on Repl.it, create a new Nodejs repl, install thejs-sha256 package, and require it.

**Blockchain Security**

The special thing about the blockchain is that it cannot easily be tampered with.

The hash of each block gives it a unique identity like a fingerprint. And this hash is generated only once when the block is created.

It is generated using the values of all the other attributes of the block. So, if any of the values are changed, the hash will no more be valid and hence the entire block will be invalid. So, this way nobody can tamper with the values stored in the block.

Another thing is if the hash is also recalculated, then it will not be detected as tampering. To avoid this, we store the hash of the previous block. This creates a chain of the blocks and if the hash of any block changes, the chain will be broken and the entire thing will be invalid.

So, to successfully hack a blockchain, the hacker has to know the location of all blocks and simultaneously change all their hashes and relink the whole blockchain. This is not impossible but extremely tedious as there is a mandatory gap of 10 minutes between each transaction. So it beats the whole purpose of life.

**Decentralized Blocks**

These blockchains are not stored in a centralized server. But the blocks are distributed across the internet and they are connected using a peer-to-peer network.

When someone joins the network, they are given the full copy of the network. When they create a new block, the block is sent to all the peers and the validity of the chain is checked. If everything checks out, the block is connected to the chain.