(1) Each block will have a constructor which will receive the properties of the block . Each block will have index , timestamp , data , previous hash . Timestamp tells us when this block was created . Data will contain details of transaction i.e. sender , receiver and amount transacted . Previous hash is a string which contains the hash of the block before this one . This ensures the integrity of our entire blockchain .

(2) Next , we have calculate hash method which will calculate the hash of each block created and help to identify it in the chain . We are using SHA 256 as our hash function . For this we need to install crypto.js module of javascript .

(3) The first block in our blockchain is genesis block which is added manually . Since , this is the first block , so previous hash wont exist . We will have get_latest_block() method , add_new_block() method , and also a method which validates whether our blockchain is valid or not at each step .

(4) Modern computers can create houndred of thousands of blocks per second and spam our blockchain . There's also a security issue that contents of the block can be changed and re calculate the hash for each block and ends with a valid chain even though it is tampered with it . To solve this issue , blockchain have proof of work . This is also called mining . Therefor , bitcoins has a certain number of zeroes to start with in their hash i.e difficulty .

(5) Increasing difficulty will take a lot longer in mining the blocks that could be a solution against the adversary . Our blocks now has hash value starting with 4 zeroes since diificulty by default is set to 4 . With this mechanism , we can control how fast new blocks can be added to our blockchain . This is essential for safety of the blockchain .

(6) Now , we are gonna add rewards for miners . When we start a cryptocurrency , we need to have virtual money or coins . Mining rewards steadily increases new coins into the system . We have created a new class transaction . A transaction always goes from someone , comes from someone and carries some coins .

(7) We only create blocks on a specific interval . In bitcoins case , proof of algorithm makes sure that we only create a new block every 10 minutes . All the transactions that are made in between the blocks , are temporarily stored in pending transaction array , so that they can be included in next block .

(8) Many people think that if you send some bitcoins around , they actually move away from our wallet balance to some balance . But in reality , we don't really have a balance . The transaction is just stored on the blockchain and if want to check our balance , we have to go through all the transaction that involve our address and calculate it that way .

(9) After a block is mined , we create a new Transaction to give the user mining reward , but that is added to pending transaction array . So ,the mining reward will he sent , when the block is mined . But , after mining in the second block we again get a new reward , which is in the pending transaction array state and will be included in the next block that is mined .

(10) But currently anyone can make any transaction that he wants. So effectively a user can spend coins that aren't his . So , for that we are making it mandatory for transaction to be

signed with a private and public key . That way we can only spend coins in the wallet if we have a private key to it .

(11) Next , we import a new library ELLIPTIC . This library will allow us to generate a public and private key . It also has methods to sign something and verify a signature . We can use any elliptic curve which is used in bitcoin . We than generate a new key pair and convert them into hex strings. We don’t need this only to sign transaction but also to get the balance in our wallet .

(12) Our signTransaction method will receive a signingkey , which will be our private and public key pair . Signing key will be the object that we got from our elliptic library . Before we sign a transaction we check if our public key equals the from address . We can only spend the coins from the wallet for which we have the private key .