Online Payment Using Blockchain

Karthikeya Thanapal, *Dhiraj* Mehta, *Karthik* Mudaliar, and *Bushra* Shaikh
SIES Graduate School of Technology,
University of Mumbai,
Maharashtra, India.

***Abstract*. Increasing list of records is with blockchain where each record is linked with the help of cryptography. Every block in the chain contains timestamp, transaction details and hash of a previous block, hash is cryptographic hash. This is a secured system, which we plan to replace the current online payment system. A current online payment gateway is prone to hackers where the attacker can tamper into the network, thus creating money loss. And not only this but also the transaction has to go through multiple payment systems which consumes time, also creating a risk of transaction getting failed. So, our system would be using blockchain that allows online transactions which would allow online payments to be sent directly from one party to another without going through a financial institution and in a secured way. This system allows online transactions between two parties based on cryptographic proof without relying and trusting for a third party. To record transactions, we use proof of work algorithm which makes computationally impractical for an attacker to change. Digital signatures provide part of the solution for ensuring the security and integrity of the data that is recorded onto a blockchain.**

I. INTRODUCTION

A blockchain is a public ledger of information collected through a network that sits on top of the internet. It is how this information is recorded that gives blockchain its groundbreaking potential.

Blockchain [1], as its name suggests, consists of multiple blocks strung together. The words “block” (digital pieces of information) and “chain” (stored in a public database). Each block in the network containing the data is secured and connected to each other with the help of cryptography principles Data cannot be changed or altered once recorded in a block, making it impossible to do so without it being seen by the other participants on the network

A node can be any electronic device, including a computer, phone, a printer or even a fridge, as long as it is connected to the internet. All nodes are equal in importance on a blockchain, but a node can have different roles in making a blockchain work. In comparison to the normal network, the blockchain network is completely different having no central point that stores and controls information. Instead, the responsibility to look after the network and store information is shared by different devices, known as peers, on that network. This is why a blockchain network is known as a peer-to-peer network. Instead, information is being constantly recorded and interchanged between all of the participants on the network.

Cryptography is used in blockchain as a means of ensuring transactions are done safely, while securing all information and storages of value. Therefore, anyone using blockchain can have complete confidence that once something is recorded on a blockchain, it is done so legitimately and in a manner that preserves security. Integrity of the data is ensured using digital signature, which is the main aspect of data recorded in blockchain Genesis block is the first block of blockchain, contains its transactions that, when combined and validated, produce a unique hash. This hash and all the new transactions that are being processed are then used as input to create a unique hash that is used in the next block in the chain. This ensures that each block links to its previous block through its hash, forming a chain back to the genesis block, so the name blockchain.

Online transaction has gained a huge market for payment and hence becomes important to look into its cons as well as flaws. The current payment gateway systems include various third party system, which is time consuming because it happens that the transaction has to go through multiple third parties which also creates the risk of transaction getting failed. The very next important factor is security, where the current system is not fulfilling the expectations of the customer. This is because there are various cases breaking the security of the transaction where the attacker tampers into the network and leading to money loss and also the faith of the customers is lost. Then there comes additional transactional charges which is point to be looked upon from the customer’s point of view which can be reduced using blockchain. Again, improving financial management is a need which can be accomplished using blockchain where we look upon to create a decentralized application. Blockchain makes it easy to maintain the transactions on a whole, and also fastens the transactions where the current system fails i.e. it is much slower compared to blockchain.

So we plan at introducing blockchain for online payment. Each node i.e. customer side transaction would be recorded in the blockchain. The current public blockchains available are Ethereum, Bitcoin etc. Though Ethereum provides a platform to create our own smart contracts, but since it is a public network the transaction details get visible to all rather than just showing it to a sub-network where only respective people get to view their transactions.

We aim at creating our own public network where MongoDB acts as the blockchain database. All the transactions would be first validated through miners and then stored as a chain in MongoDB. Every time a user tries to have a transaction first its been shared with the miners. There would be specific number of miners who would be using of proof of work for validation and then their results will be shared with each other to finally validate the transaction and then added to the chain in the MongoDB.

II. LITERATURE SURVEY

Nakamoto, Satoshi [2]. In this paper, the complete mechanism of blockchain technology for a electronic cash system that basically allows online payments to be sent directly from one party to another without going through a financial institution is presented. It explains a network system which is distributed i.e. peer to peer network which resulted to be a solution for double spending and the Proof of Work algorithm for carrying out safe and secure transactions.

Judmayer, Aljosha et.al [3] presented an overview of blockchain technology in technical point of view also introduced the concepts of cryptographic currencies and the consensus ledgers. This paper mainly focused on the Bitcoin cryptographic currencies saying that the current scientific community is relatively slowly to this emerging and fast-moving field of blockchain technology reason as not sufficient resources available other than bitcoin. It explained deeply about bitcoin and why it has gained a huge market and interest in today’s technology and also highlights the challenges in the area of digital assets management and presents a discussion of Bitcoin usability, privacy, and security challenges from the user’s perspective, the concept, characteristics, need of Blockchain and how Bitcoin works. It attempts to highlights role of Blockchain in shaping the future of banking, financial institutions.

Zibin Zheng et al. [4] provided an overview of blockchain architecture firstly and compared some typical consensus algorithms used in different blockchains. Also discussed various blockchain based applications that are covering numerous fields like financial services, reputation system, IOT so on. Furthermore, technical challenges of blockchain technology such as scalability of security problems waiting to be overcome and recent advances are briefly listed and possible future trends for blockchain.

III. SYSTEM DESIGN

Fig 1. explains the workflow of usage of Blockchain in Online payment system. When the user initializes a transaction, a block is been created which is sent for the confirmation or verified using the consensus algorithm. After the successful verification the block gets added in the chain and transaction is done.

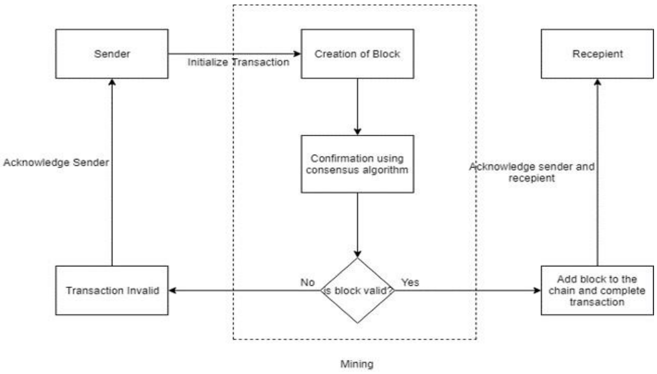

Sender
Initialize Transaction
Creation of Block
Recepient
Acknowledge Sender
Confirmation using consensus algorithm
Acknowledge sender and recepient
Transaction Invalid
No
is block valid?
Yes
Add block to the chain and complete transaction
Mining

Fig 1. System Design

A. METHODOLOGY

Every time a new user starts with our application, he/she has to register in the application with the account details. As soon as the user registers himself/herself a private key is generated for account recovery. This private key is generated using random-words npm package which generates a 12-words private key. And the input to this random-words package is the account details added by the user. This private key has to be privately stored by the user for account recovery in the future, and also this secures the user account from being compromised.

After successful registration, the user can login into the application using the username and password, where each user is uniquely identified by his/her username. Post successful login the user can initiate the transaction, where he/she can send the money to the other users with in the application. Once user initiates the transaction, it is been shared with all the miners. Initially, a hash value is generated for the transaction using the sender’s username, receiver’s username, the amount and timestamp. Timestamp parameter makes sure that each block is unique which in turn generates a unique hash value.

Now in the MongoDB we have 3 collections. One is the user collection having all the user account details. Then we have transaction collection which contains the validated transactions. Here, the UserID is considered as the key, and the value is transaction ID and its relevant details. And the third table is the validation table, where after a transaction is initiated it is been added to the table. UserID again remains key and value for this key is again the transaction ID and the hash value generated after transaction initialization.

As soon as transaction is initiated, the transaction details are updated in a real time database which then triggers the miner's application. Now we have 5 predetermined miners, where we have a NodeJS application running in the system. The application here uses proof of work as the consensus algorithm to validate the transaction.

User chain is the complete transaction details which is been encrypted using user's private key and distributed among all the miners. Once the transaction is initiated by the user, his/her user chain is validated. Here validation is done by first checking the encrypted chain stored at the miner side with the user's current chain in the blockchain. If yes, then transactions proceeds for further validation using proof of work algorithm. If the user chain with in the miner is not same then it may be a case of some attack or malicious activity and the transaction is failed. Our system has 5 miners, so if suppose there are 2 miners who have a different user chain, then we take those encrypted user chain, and try decrypting it using user's private key. If its getting decrypted than that user chain is correct and others are being the wrong ones, so accordingly we update the correct user chain among the miners and roll back the transaction. After the chain validation, we go with proof of work.

Going deeper, proof of work is a requirement to define an expensive computer calculation, also called mining, that needs to be performed in order to create a new group of trustless transactions (the so-called block) on a distributed ledger called blockchain.
Mining serves as two purposes:

1. To verify the legitimacy of a transaction, or avoiding the so-called double-spending;
2. To create new digital currencies by rewarding miners for performing the previous task.

So in our application the difficulty level is set to 4. So the hash value must have four zeros in the start and then the algorithm tries to find the nonce value. For example if this is our hash is 4dd3426129639082239efd583b5273b1bd75e8d78ff2e8d then the miner tries to find the nonce value where after appending random value if the resultant hash has four zeros in the start then that value is considered as the nonce value. Once nonce value is determined, the value is shared between the miners. We again have a database, where for each transaction we have the miner and their nonce value. If the maximum number of miners have the same nonce value then the transaction is validated or it is rejected. The difficulty level is set to 4 precisely because if it is kept more than 4, the time consumption would be more and if set less than 4 probability of finding the nonce value becomes easy.

Now, the hash sent by the user is encrypted using the generated nonce value. Then at the server side, we try to decrypt the hash value using the nonce value sent by the miners. This is just to confirm that, no attack is been performed or data is been manipulated while sharing it between the users and miners. So if we are able to decrypt the hash value using the nonce value the transaction is added to the transaction table with respect to the particular UserID as the key who initiated the transaction. And these transactions are fetched and viewed in the application. Else the transaction is failed. This way we can also tackle the 51% attack.

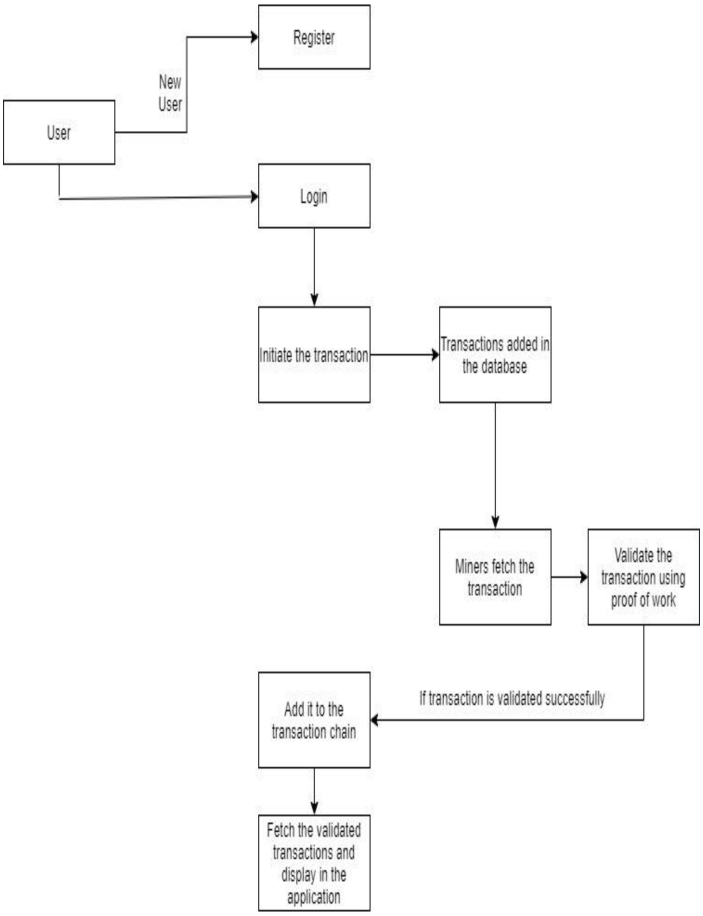
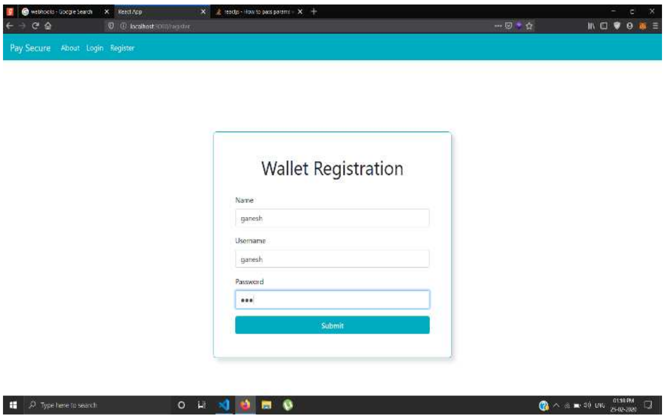

Register
New User
User
Login
Initiate the transaction
Transactions added in the database
Miners fetch the transaction
Validate the transaction using proof of work
If transaction is validated successfully
Add it to the transaction chain
Fetch the validated transactions and display in the application

Fig 2. Workflow

B .RESULTS AND DISCUSSIONS



Wallet Registration

Name
ganesh

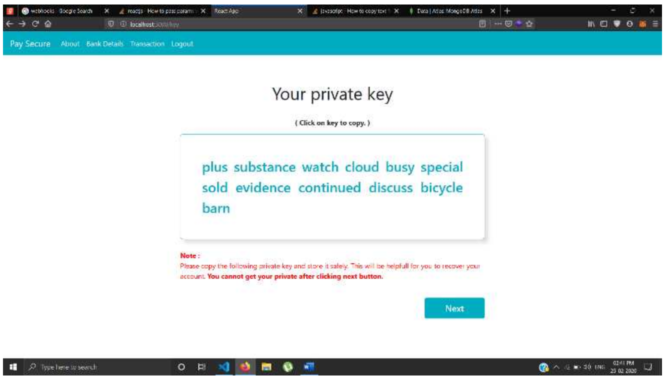
Username
ganesh

Password
•••

Submit

Fig 3. Registration Page

As shown in Fig 3. The user registers himself/herself first where the following details are mandatory and then the private key is generated. Here the username has to be unique, as the user is identified by the username.



Your private key

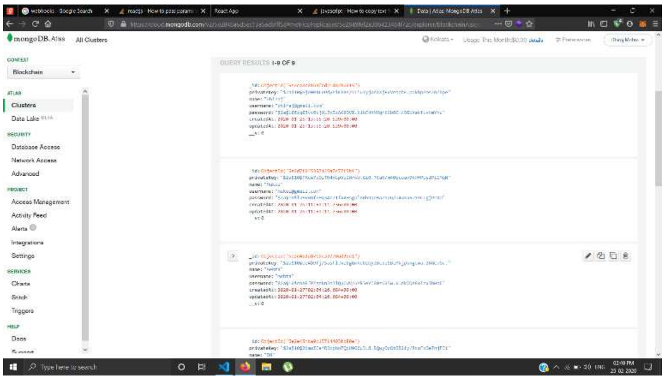
(Click on key to copy.)

plus substance watch cloud busy special sold evidence continued discuss bicycle barn

Note :
Please copy the following private key and store it safely. This will be helpfull for you to recover your account. **You cannot get your private after clicking next button.**

Fig 4. Private Key

In Fig 4. The Private key has to be stored separately by the user, as this private key is used for account recovery.

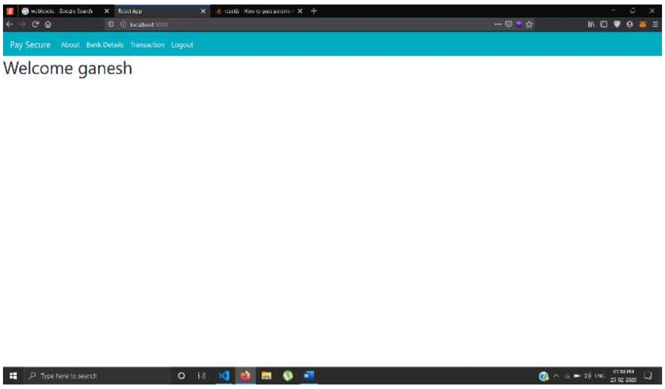


QUERY RESULTS 1-8 OF 8

```
_id: ObjectId("[illegible]")
privateKey: "[illegible]"
name: "[illegible]"
username: "[illegible]@gmail.com"
password: "[illegible]"
createdAt: 2020-03-2[illegible]
updatedAt: 2020-03-2[illegible]
__v: 0

_id: ObjectId("[illegible]")
privateKey: "[illegible]"
name: "[illegible]"
username: "[illegible]@gmail.com"
password: "[illegible]"
createdAt: [illegible]
updatedAt: [illegible]
__v: 0

_id: ObjectId("[illegible]")
privateKey: "[illegible]"
name: "[illegible]"
username: "[illegible]"
password: "[illegible]"
createdAt: 2020-03-27T02:04:26.804+00:00
updatedAt: 2020-03-27T02:04:26.804+00:00
__v: 0

_id: ObjectId("[illegible]")
privateKey: "[illegible]"
name: "[illegible]"
```

Fig 5. User Objects

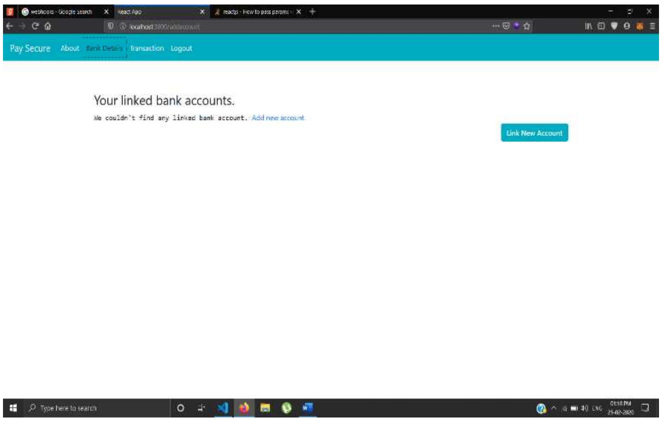
The user details are added into collections shown in Fig 5.



Welcome ganesh

Fig 6. Home Page

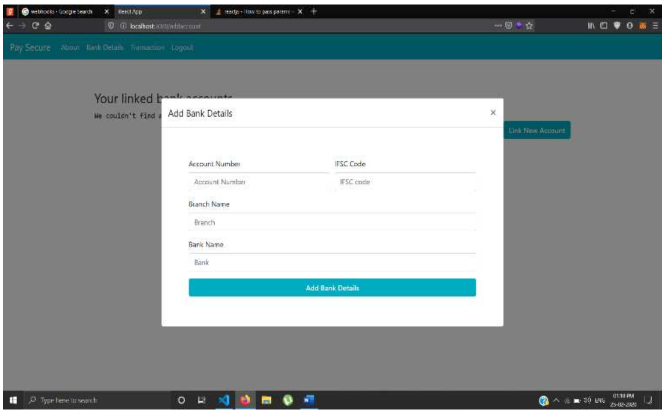
Once the user is logged in successfully he/she is redirected to the homepage as shown in Fig 6. Here the user can link the virtual bank account or perform the transactions.



Your linked bank accounts.

We couldn't find any linked bank account. Add new account

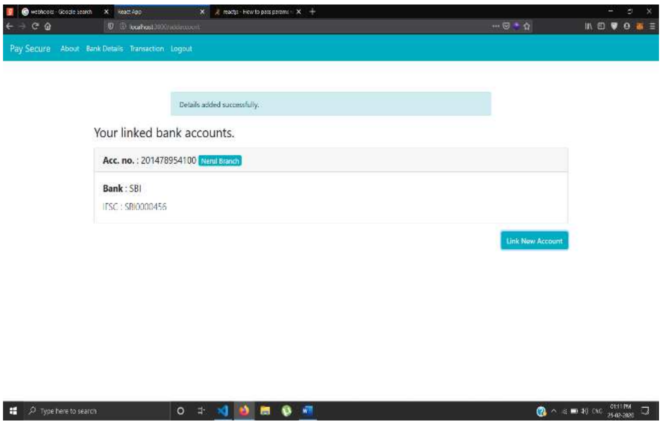
Fig 7.1. Bank Details



Add Bank Details

Account Number: Account Number
IFSC Code: IFSC code
Branch Name: Branch
Bank Name: Bank

Fig 7.2. Bank Details Form



Details added successfully.

Your linked bank accounts.

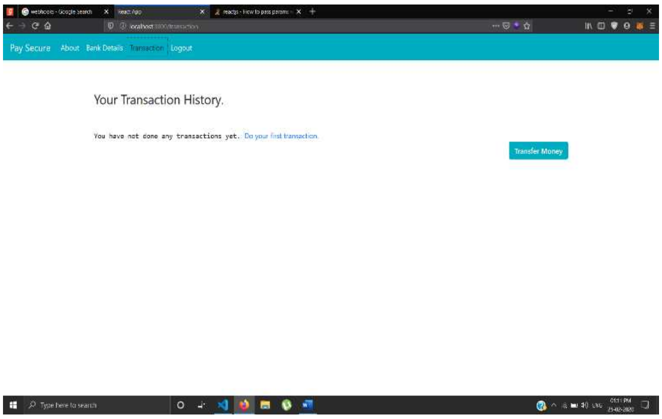
Acc. no. : 201478954100 Nerul Branch

Bank : SBI

ITSC : SBI0000456

Fig 7.3. Bank Details

The above Fig 7.1,2,3 shows how the user can link to the virtual bank account which can be considered as wallet through which money would be detected or added.

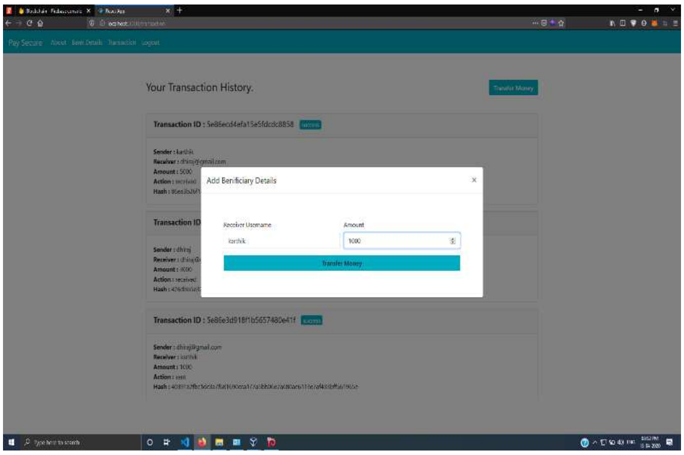


Your Transaction History.

You have not done any transactions yet. Do your first transaction.

Fig 8.1. Transaction Page

Here the valid transactions from the Transaction chain is fetch according to respective UserID and displayed in this page in Fig 8.1.



Your Transaction History.

Transaction ID : 5e86ecd4efa15e5fdcdc8858

Sender : karthik
Receiver : dhiraj@gmail.com
Amount : 5000
Action : received
Hash : [illegible]

Transaction ID [illegible]

Sender : dhiraj
Receiver : [illegible]
Amount : [illegible]
Action : received
Hash : [illegible]

Transaction ID : 5e86e3d918f1b5657480e41f

Sender : dhiraj@gmail.com
Receiver : karthik
Amount : 1000
Action : sent
Hash : [illegible]

Add Benificiary Details

Receiver Username: karthik
Amount: 1000

Fig 8.2. Transactions Form

In Fig 8.2. The user may mention the beneficiary's username and amount to be added and then the transaction would be initiated. Post this the user transaction chain at each miner is validated.

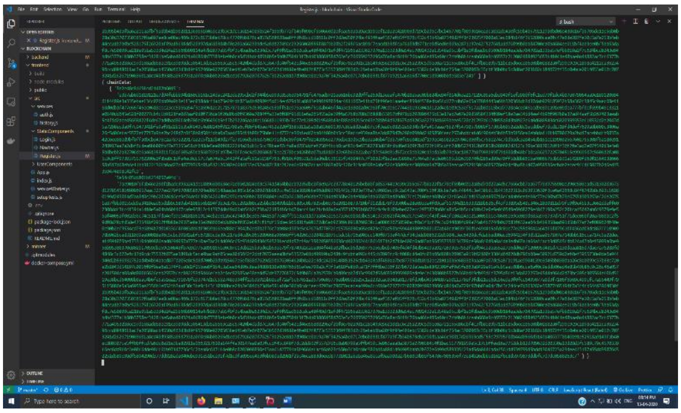
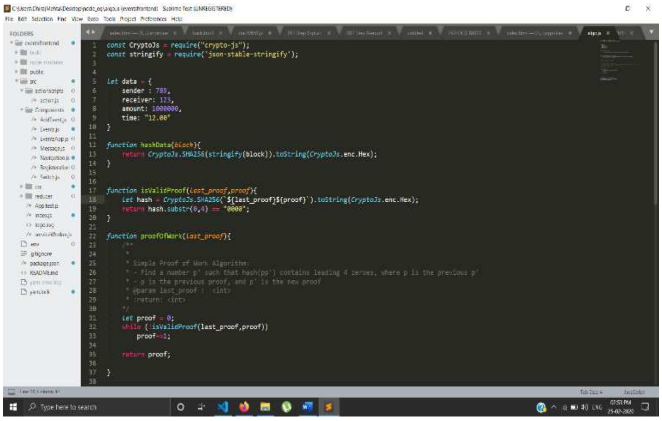


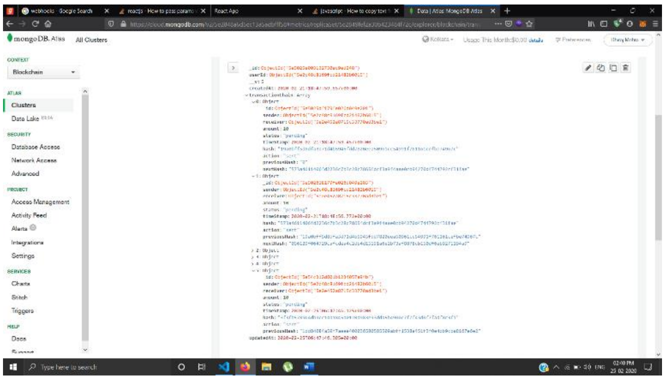
Fig 9. Encrypted chain saved on miner's side



```
const CryptoJs = require("crypto-js");
const stringify = require('json-stable-stringify');


let data = {
    sender : 789,
    receiver: 123,
    amount: 1000000,
    time: "12.00"
}

function hashData(block){
    return CryptoJs.SHA256(stringify(block)).toString(CryptoJs.enc.Hex);
}


function isValidProof(last_proof,proof){
    let hash = CryptoJs.SHA256(`${last_proof}${proof}`).toString(CryptoJs.enc.Hex);
    return hash.substr(0,4) == "0000";
}

function proofOfWork(last_proof){
    /**
     *
     * Simple Proof of Work Algorithm:
     * - Find a number p' such that hash(pp') contains leading 4 zeroes, where p is the previous p'
     * - p is the previous proof, and p' is the new proof
     * @param last_proof : <int>
     * :return: <int>
     */
    let proof = 0;
    while (!isValidProof(last_proof,proof))
        proof+=1;

    return proof;

}
```

Fig 10. Mining Consensus

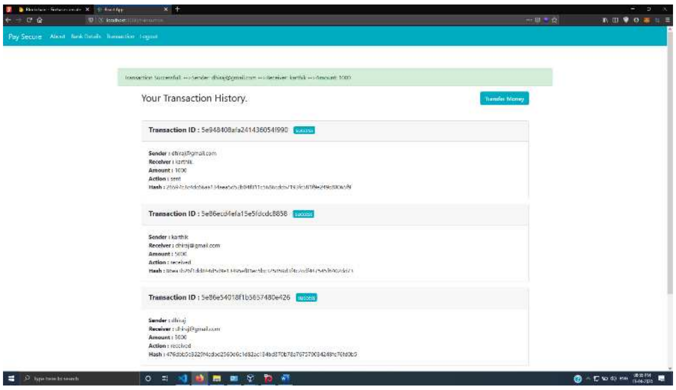
Fig 9&10. Explains the proof of work algorithm used by the miners to validate the transactions. Every time it generates the nonce value based on the difficulty value set. The nonce value from the different miners are checked and if the maximum number of nonce value is same, then the transaction is validated.



```
_id: ObjectId("[illegible]")
userId: ObjectId("[illegible]")
__v: 5
createdAt: 2020-02-21T[illegible]
transactionChain: Array
  0: Object
    _id: ObjectId("[illegible]")
    sender: ObjectId("[illegible]")
    receiver: ObjectId("[illegible]")
    amount: 10
    status: "pending"
    timeStamp: 2020-02-21T[illegible]
    hash: "[illegible]"
    action: "sent"
    previousHash: "0"
    nextHash: "[illegible]"
  1: Object
    _id: ObjectId("[illegible]")
    sender: ObjectId("[illegible]")
    receiver: ObjectId("[illegible]")
    amount: 10
    status: "pending"
    timeStamp: 2020-02-21T[illegible]
    hash: "[illegible]"
    action: "sent"
    previousHash: "[illegible]"
    nextHash: "[illegible]"
  2: Object
  3: Object
  4: Object
  5: Object
    _id: ObjectId("[illegible]")
    sender: ObjectId("[illegible]")
    receiver: ObjectId("[illegible]")
    amount: 10
    status: "pending"
    timeStamp: 2020-02-25T[illegible]
    hash: "[illegible]"
    action: "sent"
    previousHash: "[illegible]"
updatedAt: 2020-02-25T[illegible]
```

Fig 11. Transactions Object

In Fig 11. The validated transactions are added into the chain and then fetched and displayed to the user .



Transaction Successful --> Sender: dhiraj@gmail.com --> Receiver: karthik --> Amount: 1000

Your Transaction History.

Transaction ID : 5e948408afa2414360541990

Sender : dhiraj@gmail.com
Receiver : karthik
Amount : 1000
Action : sent
Hash : [illegible]

Transaction ID : 5e86ecd4efa15e5fdcdc8858

Sender : karthik
Receiver : dhiraj@gmail.com
Amount : 5000
Action : received
Hash : [illegible]

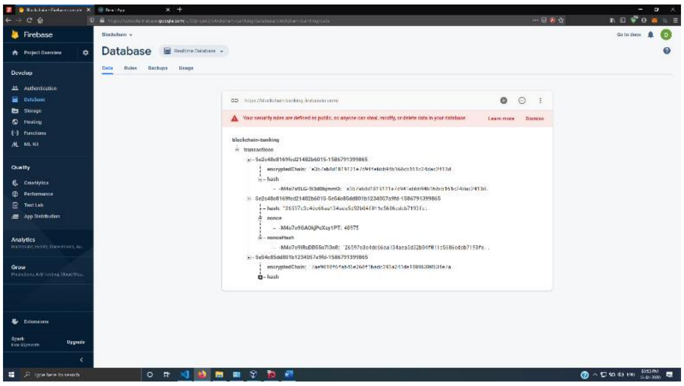
Transaction ID : 5e86e54018f1b5657480e426

Sender : dhiraj
Receiver : dhiraj@gmail.com
Amount : 3000
Action : received
Hash : 476dbb5c3229f4cdbe2560d6c1d82ac184bd870b78a76757903424 8fe76fd0b5

Fig 12. Transactions Page

Fig 12. shows that after successful validation and mining the transaction is successfully added to the user's transaction chain.

Also post transaction initialization when the attack is performed the transaction completely fails and it is not been added to the user's transaction chain.

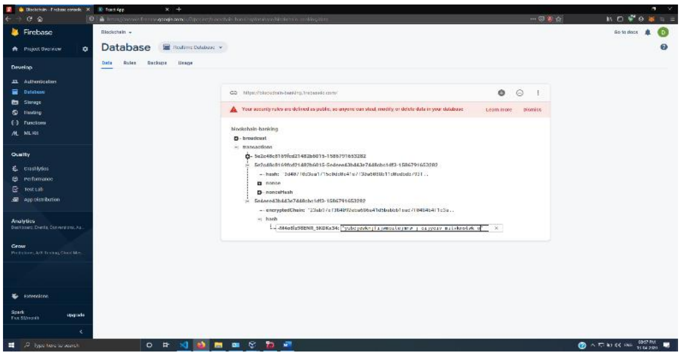


Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

```
blockchain-banking
 ⊟ transactions
    ⊟- 5e2c48c8169fcd21482b6015-1586791399865
         encryptedChain: "e3b7ebfd1[illegible]2413d..
       ⊟- hash
           – -M4o7o9LG-5i3dDbjmmO: "e3b7ebfd1[illegible]2413d..
    ⊟ 5e2c48c8169fcd21482b6015-5e54c85dd801b1234057a9fd-1586791399865
       ⊢- hash: "26597c3c4dc66aa134aea5d52b94f811c5686cdcb7193fc..
       ⊟ nonce
           – -M4o7o9GAOkjPcXsy1PT: 48975
       ⊟- nonceHash
           – -M4o7o9IRuDB66o7I3n8: "26597c3c4dc66aa134aea5d52b94f811c5686cdcb7193fc..
    ⊟- 5e54c85dd801b1234057a9fd-1586791399865
         encryptedChain: "7ae981[illegible]e7a..
       ⊞- hash
```

Fig 13.1 Data uploaded by server and updated by miners

After validation and performing consensus algorithm. Miners update the nonce values and encrypted hash values into the server as in Fig 13.2.

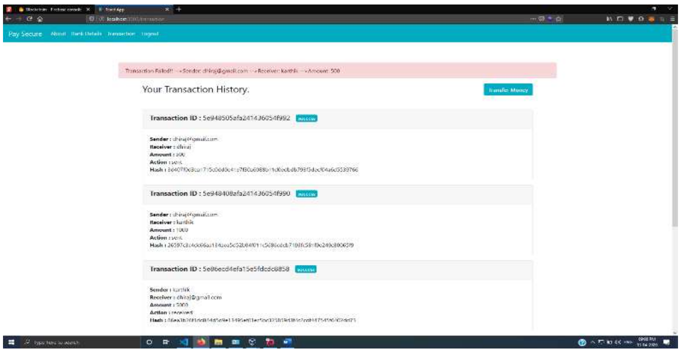


Your security rules are defined as public, so anyone can steal, modify or delete data in your database

```
blockchain-banking
 ⊞- broadcast
 ⊟ transactions
    ⊞- 5e2c48c8169fcd21482b6015-1586791653282
    ⊟ 5e2c48c8169fcd21482b6015-5edeed3bd43e7448cbc1df3-1586791653282
       –- hash: "3d407[illegible]..
       ⊞ nonce
       ⊞- nonceHash
    ⊟ 5edeed3bd43e7448cbc1df3-1586791653282
       –- encryptedChain: "23ab[illegible]..
       ⊟ hash
          └- -M4o8lz98BNR_5KDKs34: "[illegible]"
```

Figure 13.2 Data tampering in database

When some malicious attack is been performed, like here the database is been tampered by changing the value updated by user, the transaction fails which is shown in Fig 13.3.



Transaction Failed!! -->Sender: dhiraj@gmail.com -->Receiver: karthik -->Amount: 500

Your Transaction History.

Transaction ID : 5e948505afa241436054f992

Sender : dhiraj@gmail.com
Receiver : dhiraj
Amount : 500
Action : sent
Hash : 3d407[illegible]

Transaction ID : 5e948408afa241436054f990

Sender : dhiraj@gmail.com
Receiver : karthik
Amount : 1000
Action : sent
Hash : 26597c3c4dc66aa134aea5d52b94f811c5686cdcb7193fc58[illegible]

Transaction ID : 5e86ecd4efa15e5fdcdc8858

Sender : karthik
Receiver : dhiraj@gmail.com
Amount : 5000
Action : received
Hash : [illegible]

Figure 13.3 Transaction failed on data tampering

Thus, as we can see performing any attacks or malicious activities make transaction fail and hence creating a secured system for online transactions. Thus, we had complete white box testing checking the system's internal functionalities thoroughly whether it is securing the transactions properly

IV. Conclusion

Thus, we have successfully implemented the online payment transactions using blockchain, which aims at securing the complete process. The use of one-way hashing algorithm helps in securely sending the data to the miners and further the miners use the proof of work algorithm to validate the transactions using the hash value sent. The validated transactions are then stored into the blockchain and once stored the transactions in the chain cannot be tampered. Thus, the application aims at giving a secure process for online transactions by overcoming the attacks such as man-in-the-middle attack and also eliminates third-party gateways which makes the entire process of online money transfer faster.

References

[1] What is Blockchain Technology? A Step-by-Step Guide For Beginners. Available Online: https://blockgeeks.com/guides/ is-blockchain-technology/

[2] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008). Available Online: https://bitcoin.org/bitcoi

[3] Judmayer, Aljosha, Nicholas Stifter, Katharina Krombholz, and Edgar Weippl. "Blocks and chains: introduction to bitcoin, cryptocurrencies, and their consensus mechanisms." Synthesis Lectures on Information Security, Privacy, & Trust 9, no. 1 (2017): 1-123.

[4] Zheng, Zibin, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. "An overview of blockchain technology: Architecture, consensus, and future trends." In 2017 IEEE International Congress on Big Data (BigData Congress), pp. 557-564. IEEE, 2017.