Design, Analysis and Implementation of Online Competitive Examination Using Heterogeneous Consortium Blockchain

Anmol Singh
Department of Computer Science and Engineering
Amity University Jharkhand
Ranchi, India

***Abstract*— This paper provides an alternative design, analysis and implementation of present E-examination system using concepts of consortium blockchain mixed with public blockchain concepts like PoW consensus protocol giving a heterogeneous solution which is more secure in terms of data mutability of the candidates. The experiment performed propose a decentralized application(D-APP) which use SHA-256 hashing cryptography to encode the blocks. Also, it covers the core concepts of distributed ledger containing student details in a decentralized network of examination centers forming different levels of blockchain and how it's distribution throughout the network.**

The proposed model diversifies the future of blockchain in the field of education.

***Keywords*— D-APP, Blockchain, SHA-256, PoW**

I. INTRODUCTION

With emergence of technology boosted by internet led a paradigm shift in work flow of educational institution in India and all around the world. Today's E-examination System offers high scalability with higher reach of candidates and have a boon of AI based proctoring preventing cheating and giving real time analytics of result have been considered a reliable tool for conducting examination country wise and worldwide. It has been seen that E-Learning Management System (E-LMS) has been adopted by various Colleges, Examination Institutions and schools during and prior of Covid-19 outbreak.

Still many prominent National examination board doesn't consider the present e- examinations system reliable for conducting national level or state level examination. One of the most recent examples could be seen during covid-19 outbreak worldwide where various examinations which was supposed to be conducted offline were postponed even after the situation was under control. It could be the noticed even these examination boards didn't adopt the E-examination system prior to covid-19 outbreak also. These board of examination might not have accepted the present system due to the challenges that could be faced in an E-examination system, which many include malpractice from both the server side (mutability of data, connectivity issues) and client side (open-book exam, difficulty in grading long answers).

The present centralized client server model RDBMS has been considered robust when it comes to performance and storing and retrieval of data. It takes various measures to manage and secure the data by using encryption, authentication, data masking and various access control but still could be prone to data alteration by authorized authority.

When it comes to data integrity and transparency blockchain has been considered as one of the most emerging technology in this modern era. With its decentralized peer to peer architecture and advanced cryptography techniques data is not only secure but also stored and distributed among all the people of the network forming an immutable ledger. From emergence of cryptocurrencies like bitcoin to supply chain management have drawn the attention of corporations and individuals. The all three types of blockchain (public, private, consortium) are leaving a huge impact on the future of data security and access.

This paper is based upon a proposed model of E-examination system implemented using heterogeneous consortium blockchain which is partial combination of public, private and consortium blockchain.

II. LITERATURE REVIEW

Haber, S. Stornetta [1] describes the complete procedure of timestamping of media and text file by creating immutability in files which is independent of the medium also, stamping the date and time during the creation of these digital documents. The concept of blockchain and some of its features and ideas behind hashing are actually present in this paper.

Penard and Tim [2] explains the working of Secure Hash Algorithm (SHA) family which is now widely used is blockchain technology. They have explained the fault found SHA-0 and how it was broken by the French researchers Chabaud and Joux by testing it with mathematical collision. They have also explained the improvements, drawbacks and complete mathematical explanation of SHA-1 and how SHA-2 family is more secure than the rest.

In [3] Satoshi Nakamoto brings blockchain technology from theory to practical, where he explained how a network of people can interact with each other and transact as they trust the technology, eliminating the banking intermediatory.

Xi, Zeyu [4] explains the not only the physical architecture comparison of centralized and decentralized network but also in terms of practical implementations and concluding how decentralized network is more adaptable in all situations. These days banking sector and military are also using decentralized architecture as they provide fault tolerance and also least prone to attack or collision.

In [5] the authors have analyzed PoW consensus protocol in terms of security and performance. Their experiments show how throughput could be increased in terms of transaction per second without affecting the security of the block chain. They have talked about the synchronization od miners in the block chain and network partitioning in case of competing chain. If crypto world if the block generation is faster payments but prone to security variabilities where as if the block size is larger, then the propagation rate is slower. Their quantitative framework consists of a comparative analysis of different security provisions of different proof of work blockchain. their findings suggest a correlation of block reward with security and also how they can scale bitcoin with an effect of 10 time in terms of block generation without affecting the security.

A detailed overview of consortium blockchain is been discussed in terms of architecture and its components and also its application specific to E-government [6]. Authors discuss the present E-government system which is subject to single point of failure. The proposed E-government system is based on consortium blockchain. This decentralized system consists of preselected nodes that governs the entire system and giving limited access to the other entities.

III. PROPOSED SYSTEM ARCHITECTURE

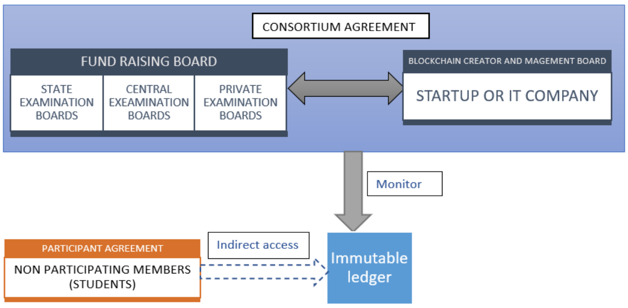


Fig. 1. Basic Blockchain design of the proposed system.

1. **Fund raising board:** It consist of different examination boards that are willing to be a part of the Examination System using blockchain technology.
2. **Blockchain creator:** They could be IT sector company who take the initiative of developing such an ecosystem by raising funds from various examination boards for management of the entire blockchain system and also for conducting online examination and storing data in the form of distributed ledger. They could completely monitor and manage the entire system.
3. **Non-Participating members:** They are the students who are giving examination.
4. **Consortium agreement:** It is the set of rights and policies under which Fund-raising board and blockchain creator and monitoring board come to an agreement for the access right and monitoring policies of the network that are part of the blockchain and immutable ledger. The main purpose of this agreement is to conduct examination by the blockchain creator in agreement with the fund-raising board and to keep the entire system as democratic as possible.
5. **Participant agreement:** Consist of the indirect access agreement of the non-participant members i.e., the students who can get access to their marks.
6. **Immutable ledger:** Consist of immutable records (marks and student details) distributed among all the nodes in the decentralized network.

IV. PROPOSED METHODOLOGY

A. 3-Layer Blockchain Model

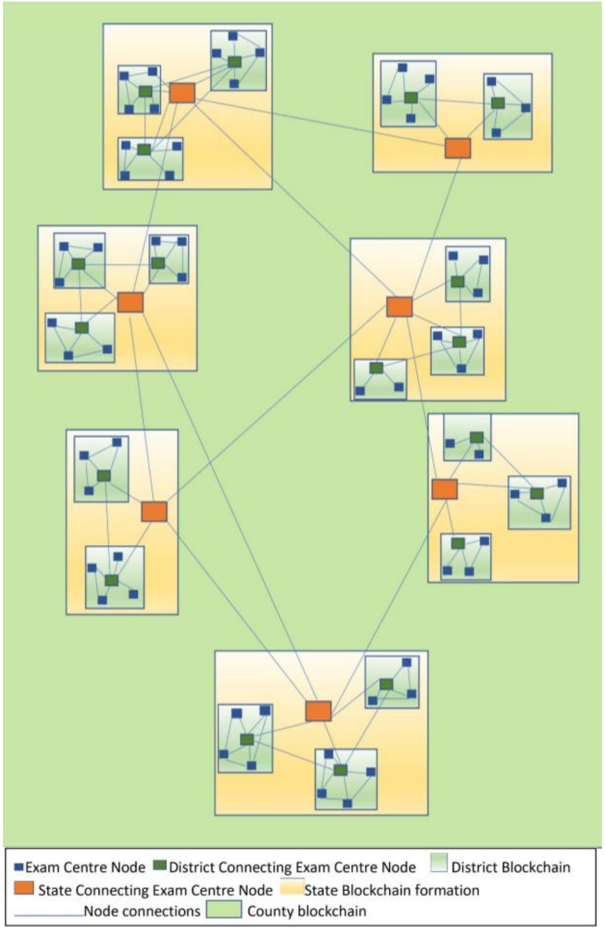

■ Exam Centre Node
■ District Connecting Exam Centre Node
District Blockchain
■ State Connecting Exam Centre Node
State Blockchain formation
Node connections
County blockchain

Fig. 2. 3-Layer Blockchain Architecture

The above Fig.2. demonstrate the 3-layers Blockchain formation and propagation summing up to form a single blockchain which gets distributed throughout the network. It can be clearly seen that there are various types of nodes that are part of the same decentralized network connected to its peer nodes. The nodes which are exam centre node, district exam centre node, state centre node is used for self-explanatory naming purpose and is subjected to same architectural prospect.

This 3- layer Blockchain formation not only adds a smooth propagation of blocks within the network but also adds additional security measures that could be adopted in terms of practical prospect. Here a naïve algorithm showing how the 3-layer blockchain formation and propagation adds to the security prospect of the blockchain system at a larger scale.

B. *Working Algorithm for 3-layer Blockchain Model*

Predefined assumptions :

a) C contains set of all nodes in a network
b) S_i is a set of nodes in a particular state,
 where i =1,2,3,4 …. m
c) D_j is a set of nodes in a particular district,
 where j=1,2,3,4 .... n
d) $S_i \subset C$
e) $D_j \subset S_i$
f) $D_j \subset C$

Example :

$C= \{node_1, node_2, node_3\ node_4, node_5\ , node_6\}$
$S_1= \{node_2, node_3, node\ _4\}$, where i =1
$S_2= \{node_2, node_3, node\ _8, node\ _4\}$, where i =2
$D_1 = \{node_4\}$, where j=1
$D_2 = \{node_4, node_8, node_3\}$, where j=2

Case 1:
For D_1 and S_1,
$D_1 \subset S_1 \rightarrow$ Pass.
$S_1 \subset C \rightarrow$ Pass
$D_1 \subset C \rightarrow$ Pass
⇨ Block added

Case 2:
For D_1 and S_2
$D_1 \subset S_2 \rightarrow$ Pass.
$S_1 \not\subset C \rightarrow$ Fail
⇨ Block not added

Case 3:
For D_2 and S_2
$D_2 \subset S_2 \rightarrow$ Pass.
$S_2 \not\subset C \rightarrow$ Fail
⇨ Block not added

Case 4:
For D_1 and S_1,
$D_1 \subset S_1$ and $n(D_1) < 0.5 \times n(S_1) \rightarrow$ Pass.
$S_1 \subset C$ and $n(S_1) < 0.5 \times n(C) \rightarrow$ Fail
⇨ Block not added

From above observation,

Case 1 denotes successful block addition passing all the three layers.

Case 2 denotes malicious node presence at layer-2, therefore block not added.

Case 3 denotes malicious node presence at layer-1 and layer-2, prevented by layer-3. Therefore, block not added.

Case 4 denotes protection of node in blockchain from 51% attack in PoW consensus protocol.

C. *Protocols and Cryptographyic Methods Used.*

1) PoW (Proof of Work)

Proof of work (PoW) is the consensus protocol in the above proposed model. This protocol is vastly used in public blockchain and is not preferred for consortium blockchain very often as it is considered as wastage of computational resources and required high consumption of electricity. It is widely used by leading crypto currencies like bitcoin. The miners have to solve a cryptographic puzzle and find a value called Nonce (number used only once).

$$Mining\ difficulty = \frac{\text{Current target}}{\text{Max target}}$$

From the above equation is clearly seen that mining difficulty will decrease if current target decrease. Also mining difficulty decreases if we increase the max target value. With reference to the case 4 of the 3-layer blockchain formation and propagation, it is clearly seen that we could reduce chances of 51% attack to minimal if we use the 3-layer architecture of proposed system and PoW consensus. Since funds are being raised by the blockchain management committee for managing and conduction online exams it is puts no point for rewards for the miners who would be an integral part of the blockchain management committee. We can also use different cryptographic hash puzzles for node specific where only the miners who are part of the system could be allowed to mine the block for their node.

2) SHA-256 cryptographic hash function

It is the most widely used cryptographic methods to encode a block which takes 256 bits of memory and is 64 character long hexadecimal hash. Each character in the resulting hash takes up four bits (4*4*4*) i.e., 4^4. The avalanche effect is one of the major reasons for the choice of using it in this project as each student would have a unique attribute that could be easily sufficient enough to withstand collision. Increasing the unique attributes can increase less chances of collision in the block chain.

V. Basic Implementation

This implementation is based upon a basic idea on how one can simulate the proposed system irrespective of a standard blockchain technologies such as Ethereum, IBM Blockchain etc.

The idea is to build a decentralized network of nodes, which are running on different port of the same computer. For different nodes operating on different port, we can create a standard web applications web application that are running on different ports in the same network.

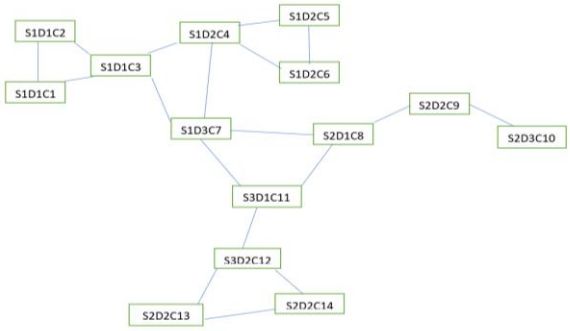


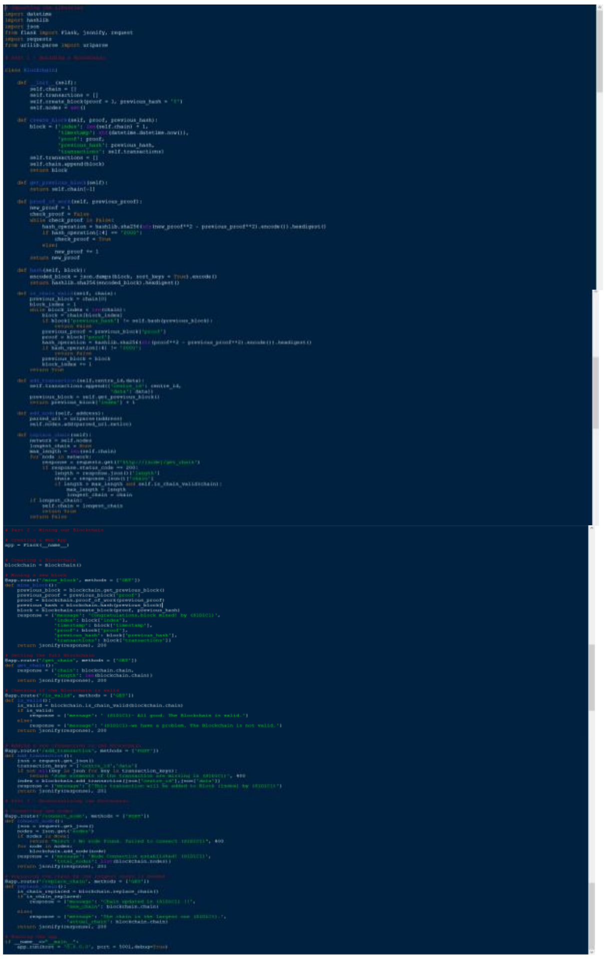
Fig. 3. Example of Decentralized Nodes Connectivity

From Fig.3. each node has been named with $S_nD_mC_k$ naming convention.
Where n is the state number,
m is the district number within a state S_n,
k is the center number within a district D_m of state S_n.
Assuming each node is an examination center with same functionality and computing power. The set of nodes within the network is within the control of the fund raising and blockchain management committee being monitored continuously.

A. Building D-APP

The decentralized nodes are web applications running on different port having same code and have same functionality serving as the smart contract. The following is an example of how one can create a Decentralized Web Application (D-APP) :
Requirements:

- An Object-Oriented Programing language.
- A module for using datetime
- A module for using SHA256 Cryptographic hash function
- A web application module
- A module to send HTTP request
- A module for URL parsing
- A json module for data transfer between server and client



```
# Importing the libraries
import datetime
import hashlib
import json
from flask import Flask, jsonify, request
import requests
from urllib.parse import urlparse

# Part 1 - Building a Blockchain

class Blockchain:

    def __init__(self):
        self.chain = []
        self.transactions = []
        self.create_block(proof = 1, previous_hash = '0')
        self.nodes = set()

    def create_block(self, proof, previous_hash):
        block = {'index': len(self.chain) + 1,
                 'timestamp': str(datetime.datetime.now()),
                 'proof': proof,
                 'previous_hash': previous_hash,
                 'transactions': self.transactions}
        self.transactions = []
        self.chain.append(block)
        return block

    def get_previous_block(self):
        return self.chain[-1]

    def proof_of_work(self, previous_proof):
        new_proof = 1
        check_proof = False
        while check_proof is False:
            hash_operation = hashlib.sha256(str(new_proof**2 - previous_proof**2).encode()).hexdigest()
            if hash_operation[:4] == '0000':
                check_proof = True
            else:
                new_proof += 1
        return new_proof

    def hash(self, block):
        encoded_block = json.dumps(block, sort_keys = True).encode()
        return hashlib.sha256(encoded_block).hexdigest()

    def is_chain_valid(self, chain):
        previous_block = chain[0]
        block_index = 1
        while block_index < len(chain):
            block = chain[block_index]
            if block['previous_hash'] != self.hash(previous_block):
                return False
            previous_proof = previous_block['proof']
            proof = block['proof']
            hash_operation = hashlib.sha256(str(proof**2 - previous_proof**2).encode()).hexdigest()
            if hash_operation[:4] != '0000':
                return False
            previous_block = block
            block_index += 1
        return True

    def add_transaction(self,centre_id,data):
        self.transactions.append({'centre_id': centre_id,
                                  'data': data})
        previous_block = self.get_previous_block()
        return previous_block['index'] + 1

    def add_node(self, address):
        parsed_url = urlparse(address)
        self.nodes.add(parsed_url.netloc)

    def replace_chain(self):
        network = self.nodes
        longest_chain = None
        max_length = len(self.chain)
        for node in network:
            response = requests.get(f'http://{node}/get_chain')
            if response.status_code == 200:
                length = response.json()['length']
                chain = response.json()['chain']
                if length > max_length and self.is_chain_valid(chain):
                    max_length = length
                    longest_chain = chain
        if longest_chain:
            self.chain = longest_chain
            return True
        return False

# Part 2 - Mining our Blockchain

# Creating a Web App
app = Flask(__name__)


# Creating a Blockchain
blockchain = Blockchain()

# Mining a new block
@app.route('/mine_block', methods = ['GET'])
def mine_block():
    previous_block = blockchain.get_previous_block()
    previous_proof = previous_block['proof']
    proof = blockchain.proof_of_work(previous_proof)
    previous_hash = blockchain.hash(previous_block)
    block = blockchain.create_block(proof, previous_hash)
    response = {'message': 'Congratulations,block mined! by (S1D1C1)',
                'index': block['index'],
                'timestamp': block['timestamp'],
                'proof': block['proof'],
                'previous_hash': block['previous_hash'],
                'transactions': block['transactions']}
    return jsonify(response), 200

# Getting the full Blockchain
@app.route('/get_chain', methods = ['GET'])
def get_chain():
    response = {'chain': blockchain.chain,
                'length': len(blockchain.chain)}
    return jsonify(response), 200

# Checking if the Blockchain is valid
@app.route('/is_valid', methods = ['GET'])
def is_valid():
    is_valid = blockchain.is_chain_valid(blockchain.chain)
    if is_valid:
        response = {'message': ' (S1D1C1)- All good. The Blockchain is valid.'}
    else:
        response = {'message': '(S1D1C1)-we have a problem. The Blockchain is not valid.'}
    return jsonify(response), 200


# Adding a new transaction to the Blockchain
@app.route('/add_transaction', methods = ['POST'])
def add_transaction():
    json = request.get_json()
    transaction_keys = ['centre_id','data']
    if not all(key in json for key in transaction_keys):
        return 'Some elements of the transaction are missing in (S1D1C1)', 400
    index = blockchain.add_transaction(json['centre_id'],json['data'])
    response = {'message': f'This transaction will be added to Block {index} by (S1D1C1)'}
    return jsonify(response), 201

# Part 3 - Decentralizing our Blockchain

# Connecting new nodes
@app.route('/connect_node', methods = ['POST'])
def connect_node():
    json = request.get_json()
    nodes = json.get('nodes')
    if nodes is None:
        return "Error : No node found. Failed to connect (S1D1C1)", 400
    for node in nodes:
        blockchain.add_node(node)
    response = {'message': 'Node Connection established! (S1D1C1)',
                'total_nodes': list(blockchain.nodes)}
    return jsonify(response), 201

# Replacing the chain by the longest chain if needed
@app.route('/replace_chain', methods = ['GET'])
def replace_chain():
    is_chain_replaced = blockchain.replace_chain()
    if is_chain_replaced:
        response = {'message': 'Chain updated in (S1D1C1) !!',
                    'new_chain': blockchain.chain}
    else:
        response = {'message': 'The chain is the largest one (S1D1C1).',
                    'actual_chain': blockchain.chain}
    return jsonify(response), 200

# Running the app
if __name__=="__main__":
    app.run(host = '0.0.0.0', port = 5001,debug=True)
```

Fig. 3. Source Code

The above decentralized application from Fig 3. is same for all the nodes in the network. The above code is implemented using Python 3.8. and Flask which is Microframework used to create the required web application. There are different modules used which satisfy the requirement mentioned.

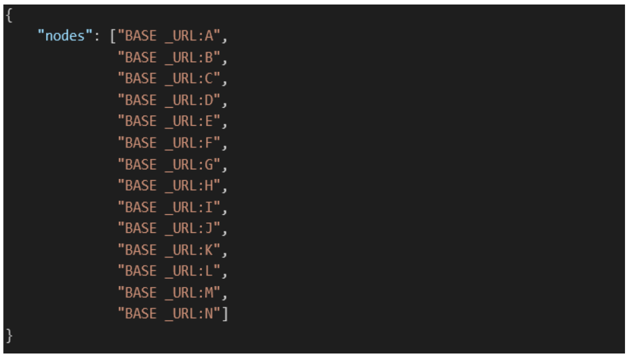
B. Node connection

The below table shows the port-to-port connectivity of various nodes in the network. The port numbers can be selected as per the user choice excluding the reserved port numbers. Here A,B,C,D…N has been taken for representation purpose only .

TABLE I. THE NODE (PORT) AND ITS PEER'S CONNECTION

Node on Port	Connection 1	Connection 2	Connection 3	Connection 4
S1D1C1(Port:A)	(Port : B)	(Port : C)	-	-
S1D1C2 (Port:B)	(Port : A)	(Port : C)	-	-
S1D1C3(Port: C)	(Port : D)	(Port : G)	(Port : A)	(Port : B)
S1D2C4(Port:D)	(Port : E)	(Port : F)	(Port : C)	(Port : G)
S1D2C5(Port:E)	(Port : D)	(Port : F)	-	-
S1D2C6(Port: F)	(Port : D)	(Port : E)	-	-
S1D3C7(Port:G)	(Port : H)	(Port : K)	(Port : C)	(Port : D)
S2D1C8(Port:H)	(Port : I)	(Port : K)	(Port : G)	-
S2D2C9(Port:I)	(Port : J)	(Port : H)	-	-
S2D2C10(Port:J)	(Port : I)	-	-	-
S3D1C11(Port: K)	(Port : L)	(Port : G)	(Port : H)	-
S3D2C12(Port : L)	(Port : K)	(Port : M)	(Port : N)	-
S3D2C13(Port : M)	(Port : L)	(Port : N)	-	-
S3D2C14(Port : N)	(Port : L)	(Port : M)	-	-

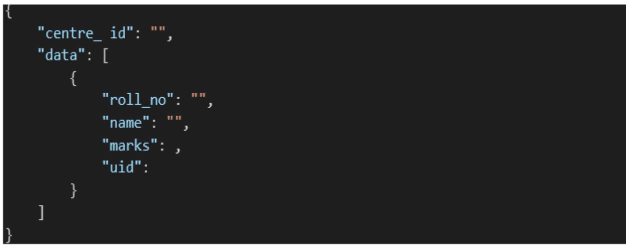
C. Network and Transaction



```
{
    "nodes": ["BASE _URL:A",
              "BASE _URL:B",
              "BASE _URL:C",
              "BASE _URL:D",
              "BASE _URL:E",
              "BASE _URL:F",
              "BASE _URL:G",
              "BASE _URL:H",
              "BASE _URL:I",
              "BASE _URL:J",
              "BASE _URL:K",
              "BASE _URL:L",
              "BASE _URL:M",
              "BASE _URL:N"]
}
```

Fig. 4. Total Nodes in Network

The nodes json file contain list of nodes in the network, which is under the control of the fundraising committee and blockchain management company can be seen in Fig.4. Here BASE_URL is can be the IPv4 address of the device connected to internet and A,B,C,D,E…N are the port numbers. The BASE_URL:PORT gives a unique identification to each node.



```
{
    "centre_ id": "",
    "data": [
        {
            "roll_no": "",
            "name": "",
            "marks": ,
            "uid":
        }
    ]
}
```

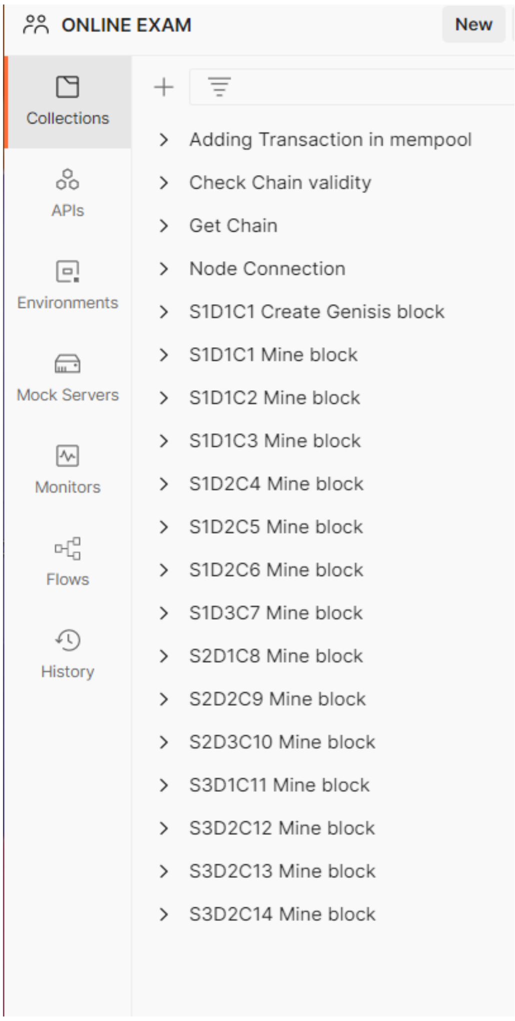
Fig. 5. Transaction Details

The transactions consist dataset of students who are giving examination on a particular centre. The transactions are 1st stored in the memory pool of the exam centre node and then automated iterative mining and block propagation in different nodes is done as assigned by the management of the blockchain system.

D. Testing and Automation

Once all the 14 servers are active and the web application running in these 14 nodes(ports) I have tested the APIs using postman and created various collection(Fig. 6.) represent a particular task assigned .

The sequence of execution followed is :

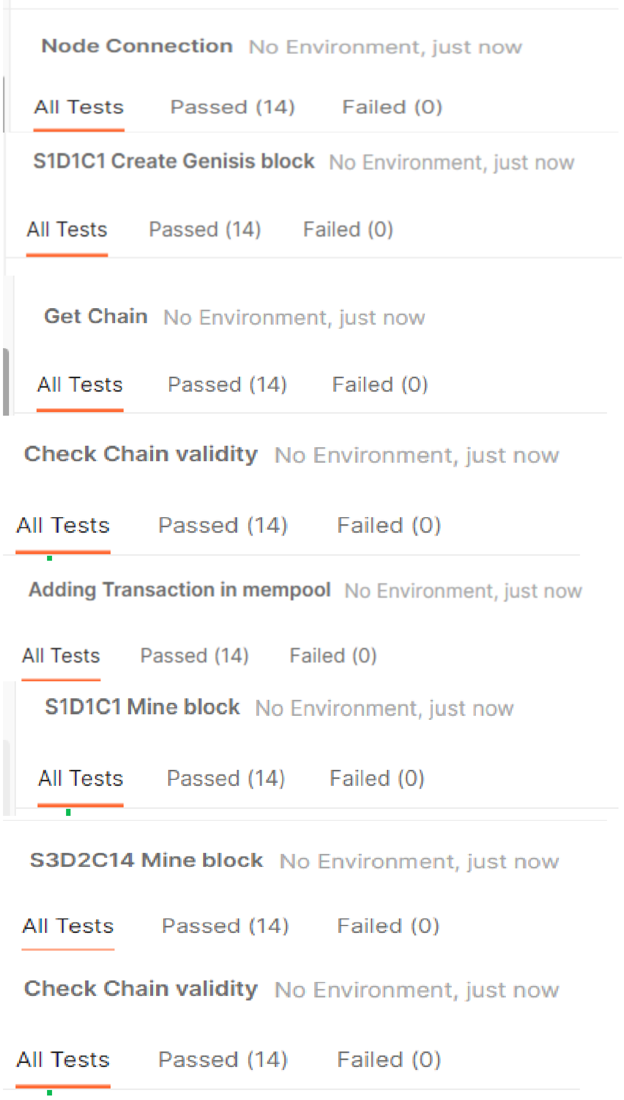


- Adding Transaction in mempool
- Check Chain validity
- Get Chain
- Node Connection
- S1D1C1 Create Genisis block
- S1D1C1 Mine block
- S1D1C2 Mine block
- S1D1C3 Mine block
- S1D2C4 Mine block
- S1D2C5 Mine block
- S1D2C6 Mine block
- S1D3C7 Mine block
- S2D1C8 Mine block
- S2D2C9 Mine block
- S2D3C10 Mine block
- S3D1C11 Mine block
- S3D2C12 Mine block
- S3D2C13 Mine block
- S3D2C14 Mine block

Fig. 6. API Collections in Postman

- Node connection (establish node connections with its respective peers)
- $S_1D_1C_1$ Create Genesis Block (here the genesis block is used for testing purpose of chain propagation in the system.)
- Adding Transaction in Mempool. (All the nodes add the students details in their memory pool)
- Check Chain Validity (check control)
- $S_ND_MC_K$, Mine Block(any one exam center mines a block and it gets propagated to its peers)
- Check chain validity (used frequently during mining)
- Get chain (could display the chain , used for monitoring)

The result of each collection runner has executed successfully (Fig 7.) and a unified immutable ledger could be seen of students participating in blockchain.



Node Connection No Environment, just now
All Tests Passed (14) Failed (0)

S1D1C1 Create Genisis block No Environment, just now
All Tests Passed (14) Failed (0)

Get Chain No Environment, just now
All Tests Passed (14) Failed (0)

Check Chain validity No Environment, just now
All Tests Passed (14) Failed (0)

Adding Transaction in mempool No Environment, just now
All Tests Passed (14) Failed (0)

S1D1C1 Mine block No Environment, just now
All Tests Passed (14) Failed (0)

S3D2C14 Mine block No Environment, just now
All Tests Passed (14) Failed (0)

Check Chain validity No Environment, just now
All Tests Passed (14) Failed (0)

Fig. 7. Collection Runner Result .

VI. CONCLUSION

Here only one cryptographic puzzle is assigned to all the nodes, but if different nodes have different cryptographic puzzle, it could lead enhancement of security in the layer 3 model of block propagation and chain formation. However, the implementation of this model has not been fully implemented. Also , advance asymmetric cryptography algorithm could be used like RSA for validation of transaction and block propagating in each layer. Other consensus protocol like PoA (proof of authority could also be used in place of PoW. Ecosystem especially made for blockchain such as Ethereum could be used. Furthermore the 3-layer blockchain model could be decomposed into more layers to make the system more reliable.

Thus, this technology can be reliable for many in coming years in India and around the world. The proposed model could be a source of additional information for people working or researching on this domain. This research project could be an idea for a startup to boost their growth. This project denotes democracy because the authorities have no control on mutating or changing the data. also, they aren't part of the blockchain system. their role is to only monitor the security aspects of the blockchain giving an immutable ledger of student participating in an examination forming a complete democratic system.

REFERENCE

[1] Haber, S., Stornetta, W.S. How to time-stamp a digital document. *J. Cryptology* **3,** 99–111 (1991)

[2] Wouter Penard and Tim van Werkhoven. On the secure hash algorithm family. In Cryptography in Context, pages 1-18. 2008.

[3] Nakamoto, Satoshi. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System. Cryptography Mailing list at

[4] Xi, Zeyu. (2020). The comparison of decentralized and centralized structure of network communication in different application fields. 10.2991/msie-19.2020.10.

[5] Gervais, Arthur & Karame, Ghassan & Wüst, Karl & Glykantzis, Vasileios & Ritzdorf, Hubert & Capkun, Srdjan. (2016). On the Security and Performance of Proof of Work Blockchains. 3-16. 10.1145/2976749.2978341.

[6] Elisa, N & Yang, Longzhi & Li, Honglei & Chao, Fei & Naik, N & Nnko, Noe & Yang, Li & Chao,. (2019). Consortium Blockchain for Security and Privacy-Preserving in E-government Systems.