BLOCKCHAIN SECURITY

MERTCAN DURMUŞ/141180024

BLOCKCHAIN

Blockchain technology is a distributed database that allows us to transfer assets that we value

Blockchain is an encrypted, ever-growing, distributed database that does not have any central authority, which was introduced in the article titled “Bitcoin: Peer-to-Peer Electronic Cash System” published in 2008 under the nickname Satoshi Nakamoto

Blockchain Security

- Proof of work algorithm
- Proof of stake algorithm
- Decentralized systems
- Select the longest chain algorithm

How it works?

- Transactions occur
- Transactions verified
- Transactions store the block
- Block added the longest chain

Blockchain application

- Crypto currencies
- Voting mechanism
- Smart contract
- Anti-money laundering tracking system
- Real-time IoT operating systems

Our application-Server

- The python language and flask framework were used in the server part

```python
class Block:
    def __init__(self, index, transactions, timestamp, previous_hash, nonce=0):
        self.index = index
        self.transactions = transactions
        self.timestamp = timestamp
        self.previous_hash = previous_hash
        self.nonce = nonce

    def compute_hash(self):
        block_string = json.dumps(self.__dict__, sort_keys=True)
        return sha256(block_string.encode()).hexdigest()
```

```python
@staticmethod
def proof_of_work(block):
    block.nonce = 0
    computed_hash = block.compute_hash()
    while not computed_hash.startswith('0' * Blockchain.difficulty):
        block.nonce += 1
        computed_hash = block.compute_hash()

    return computed_hash
```

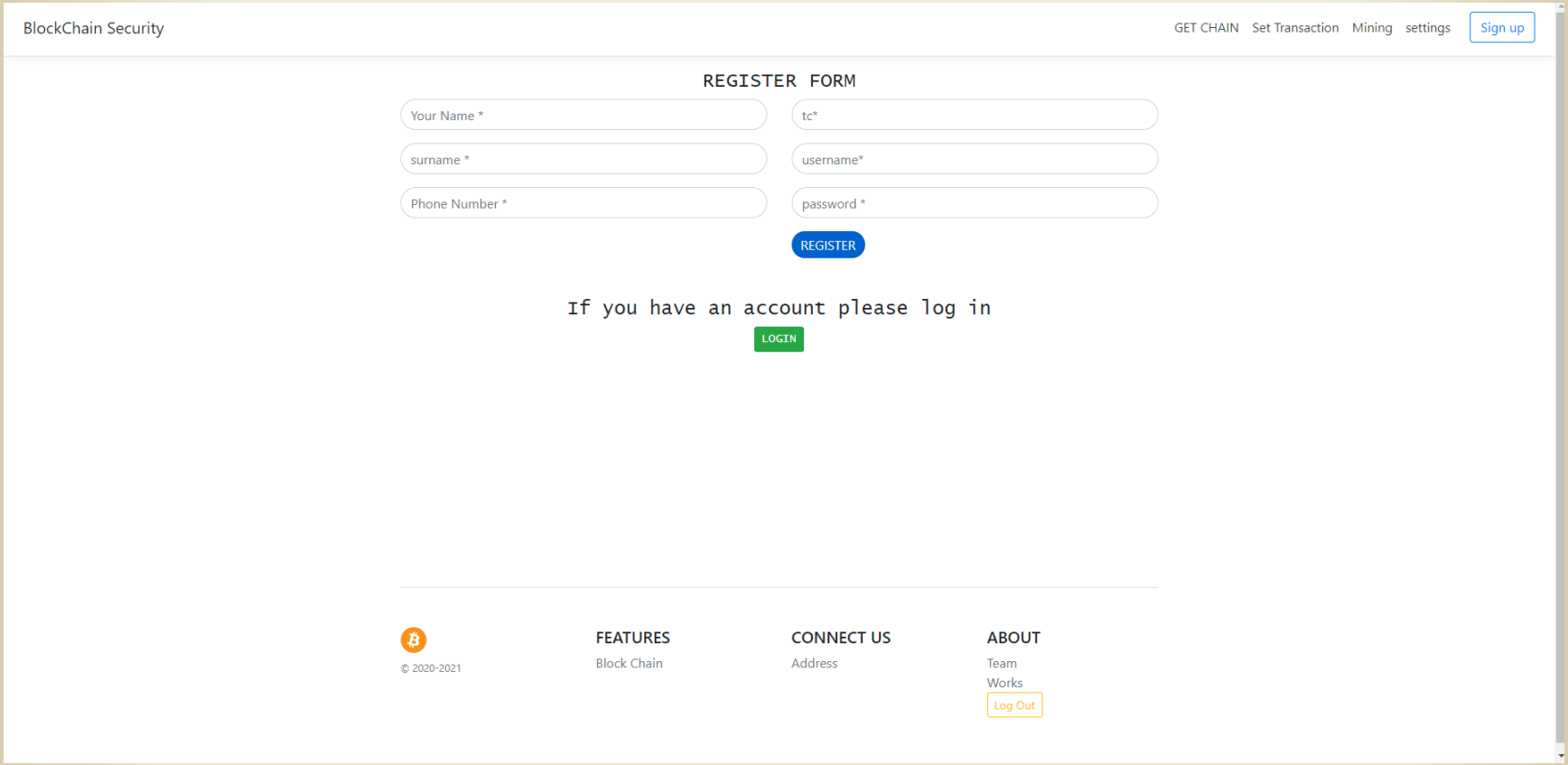
```python
@classmethod
def check_chain_validity(cls, chain):
    result = True
    previous_hash = "0"

    for block in chain:
        block_hash = block.hash
        delattr(block, "hash")

        if not cls.is_valid_proof(block, block_hash) or \
                previous_hash != block.previous_hash:
            result = False
            break

        block.hash, previous_hash = block_hash, block_hash

    return result
```

```python
def announce_new_block(block):
    for peer in peers:
        url = "{}add_block".format(peer)
        headers = {'Content-Type': "application/json"}
        requests.post(url,
                      data=json.dumps(block.__dict__, sort_keys=True),
                      headers=headers)
```

```python
def consensus():
    global blockchain

    longest_chain = None
    current_len = len(blockchain.chain)

    for node in peers:
        response = requests.get('{}chain'.format(node))
        length = response.json()['length']
        chain = response.json()['chain']
        if length > current_len and blockchain.check_chain_validity(chain):
            current_len = length
            longest_chain = chain

    if longest_chain:
        blockchain = longest_chain
        return True

    return False
```

```python
@app.route("/userIp",methods=['POST'])
def userIp():
    try:
        tx_data = request.json
        print(tx_data)
        required_fields = ["username", "ip"]
        db = MySQLdb.connect(host="localhost",  user="root",passwd="toor",db="blockchainUser")
        cur = db.cursor()
        username=tx_data.get("username")
        ipAddress=tx_data.get("ipAddress")
        sql = "INSERT INTO ipTable (username, ipAddress) VALUES(%s,%s)"
        val=(format(str(tx_data['username'])),format(str(tx_data['ipAddress'])) )
        cur.execute(sql,val)
        db.commit()
        print(cur.rowcount, "record inserted.")
        return json.dumps("success")
    except (MySQLdb.Error, MySQLdb.Warning) as e:
        try:
            print("exceptinossssss")
            print(e)
            error=str(e)
            response = jsonify(error)
            response.status_code = 200
            return response
        except IndexError:
            error=str(e)
            response = jsonify(error)
            return response
    except TypeError as e:
        print(e)
        error=str(e)
        response = jsonify(error)
        return response
    except ValueError as e:
        print(e)
        error=str(e)
        response = jsonify(error)
        return response
    finally:
        cur.close()
        db.close()
```

Our application-Client



BlockChain Security

GET CHAIN Set Transaction Mining settings Sign up

Giriş yap !

User name : 123456789 Password : ••••••••• . LOGIN

User succesfully log in

© 2020-2021

FEATURES

Block Chain

CONNECT US

Address

ABOUT

Team

Works

Log Out

BlockChain Security

GET CHAIN Set Transaction Mining settings Sign up

block 1 succesfully mined (connected to chain)

FEATURES

Block Chain

CONNECT US

Address

ABOUT

Team

Works

Log Out

BlockChain Security

GET CHAIN Set Transaction Mining settings Sign up

All blocks

index	transactions	timestamp	previousHash	nonce	hash
0		0	pre_hash	0	hash
1	[object Object],[object Object],[object Object]	1589030789.9371996	pre_hash	796	hash

FEATURES

Block Chain

CONNECT US

Address

ABOUT

Team

Works

Log Out

BlockChain Security

GET CHAIN Set Transaction Mining settings Sign up

New Transaction

>

AUTHOR

Author Name

TRANSACTION

Transaction Content

send

Transaction send succesfully

FEATURES

Block Chain

CONNECT US

Address

ABOUT

Team

Works

Log Out

Thank You!