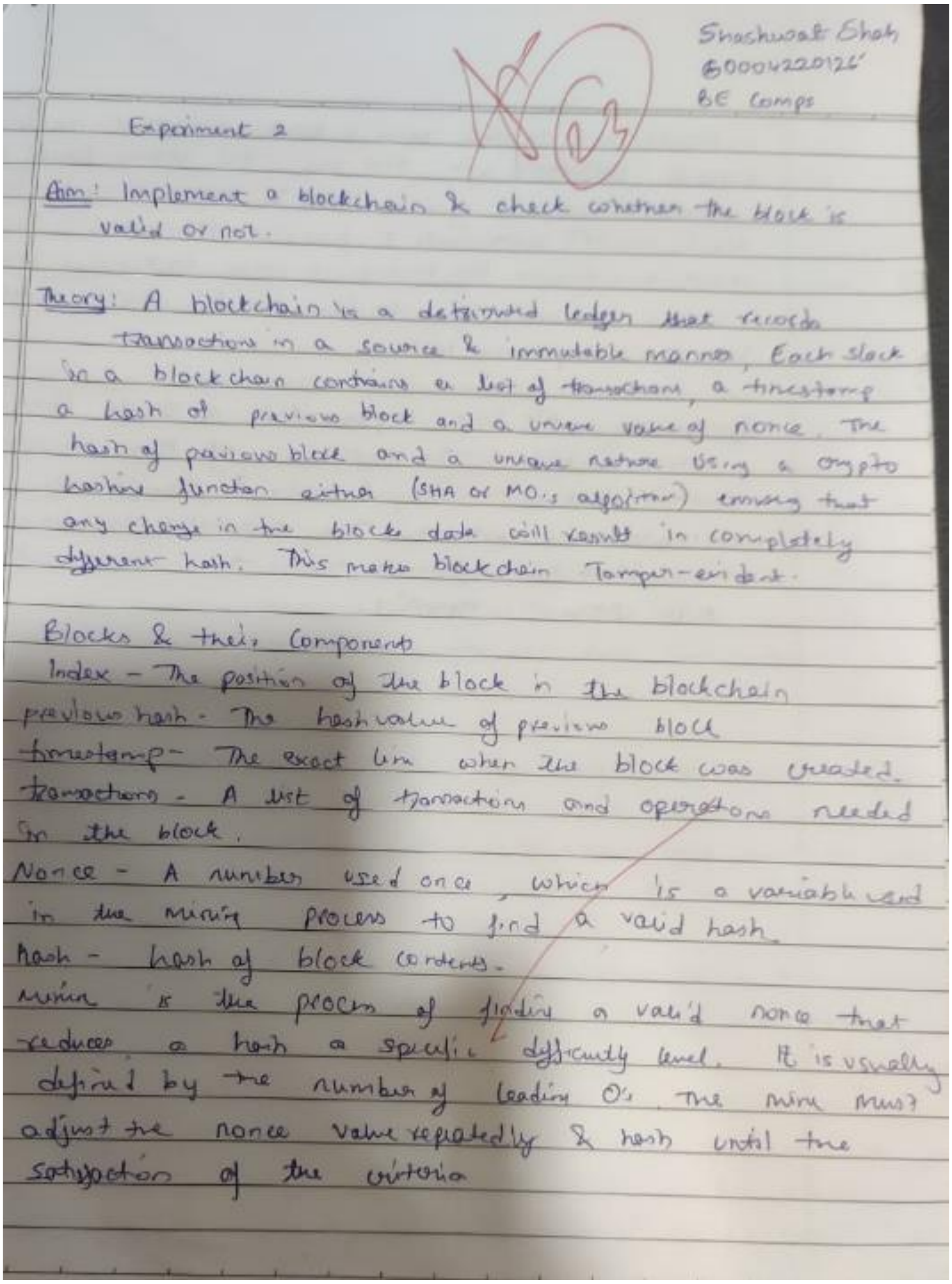
BLOCKCHAIN TECHNOLOGY

Name:-Shashwat Shah
Sapid:-60004220126
Branch:-Computer Engineering
Div/Batch:-C2-2

EXPERIMENT NO.02



Shashwat Shah
60004220126
BE Comps

Experiment 2

Aim: Implement a blockchain & check whether the block is valid or not.

Theory: A blockchain is a distributed ledger that records transactions in a source & immutable manner, Each block in a blockchain contains a list of transactions, a timestamp a hash of previous block and a unique value of nonce. The hash of previous block and a unique nature using a crypto hashing function either (SHA or MD5 algorithm) ensuring that any change in the block data will result in completely different hash. This makes blockchain Tamper-evident.

Blocks & their Components

Index - The position of the block in the blockchain

previous hash - The hash value of previous block

timestamp - The exact time when the block was created.

transactions - A list of transactions and operations needed in the block.

Nonce - A number used once, which is a variable used in the mining process to find a valid hash.

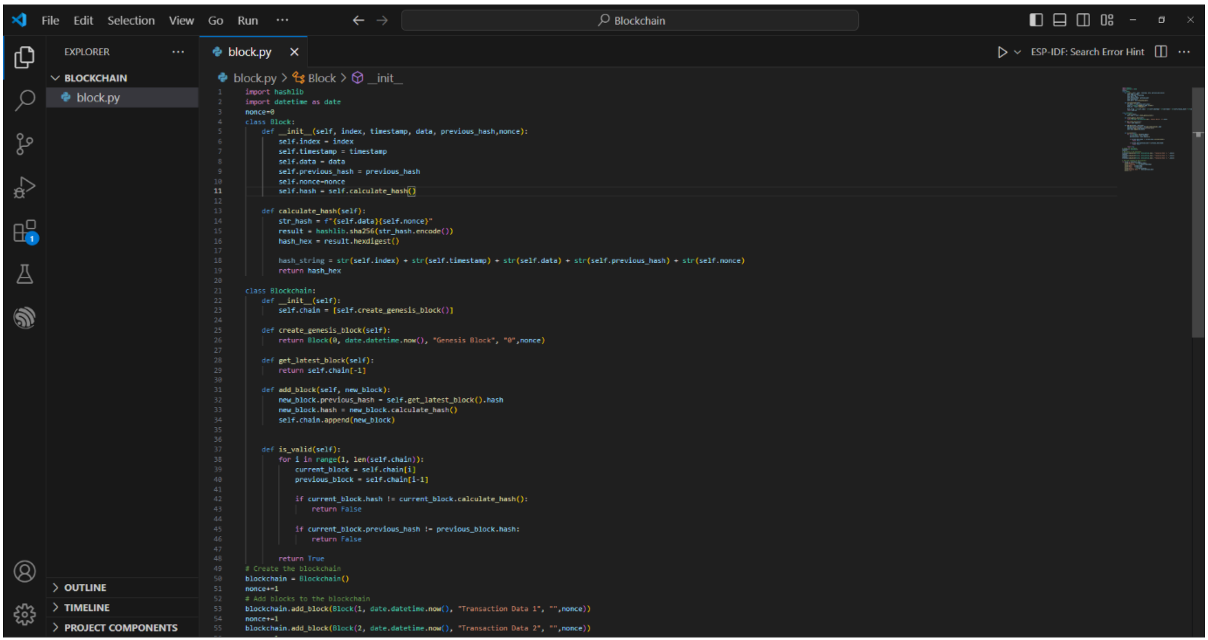
hash - hash of block contents.

mining is the process of finding a valid nonce that reduces a hash a specific difficulty level. It is usually defined by the number of leading 0's. The miner must adjust the nonce value repeatedly & hash until the satisfaction of the criteria

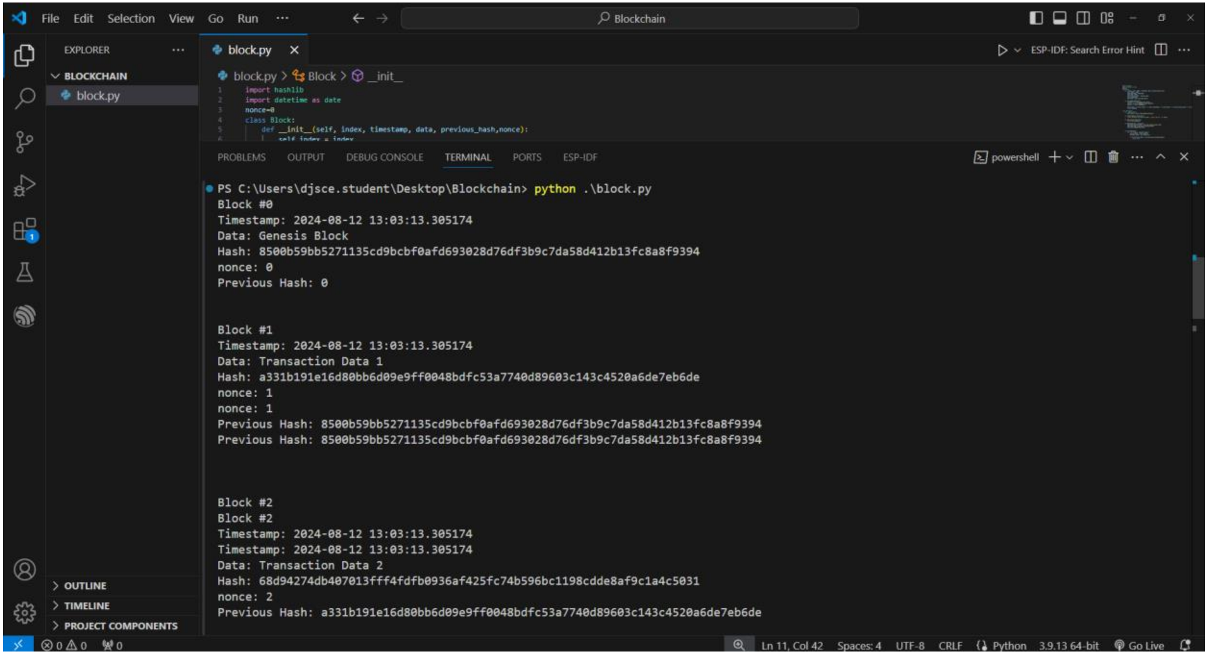
CODE & OUTPUT:-

```
import hashlib
import datetime as date
nonce=0
class Block:
    def __init__(self, index, timestamp, data, previous_hash,nonce):
        self.index = index
        self.timestamp = timestamp
        self.data = data
        self.previous_hash = previous_hash
        self.nonce=nonce
        self.hash = self.calculate_hash()

    def calculate_hash(self):
        str_hash = f"{self.data}{self.nonce}"
        result = hashlib.sha256(str_hash.encode())
        hash_hex = result.hexdigest()

        hash_string = str(self.index) + str(self.timestamp) + str(self.data) + str(self.previous_hash)
+ str(self.nonce)
        return hash_hex

class Blockchain:
    def __init__(self):
        self.chain = [self.create_genesis_block()]

    def create_genesis_block(self):
        return Block(0, date.datetime.now(), "Genesis Block", "0",nonce)

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

    def add_block(self, new_block):
        new_block.previous_hash = self.get_latest_block().hash
        new_block.hash = new_block.calculate_hash()
        self.chain.append(new_block)


    def is_valid(self):
        for i in range(1, len(self.chain)):
            current_block = self.chain[i]
            previous_block = self.chain[i-1]

            if current_block.hash != current_block.calculate_hash():
                return False

            if current_block.previous_hash != previous_block.hash:
                return False

        return True
# Create the blockchain
blockchain = Blockchain()
nonce+=1
# Add blocks to the blockchain
blockchain.add_block(Block(1, date.datetime.now(), "Transaction Data 1", "",nonce))
nonce+=1
```

```
blockchain.add_block(Block(2, date.datetime.now(), "Transaction Data 2", "",nonce))
nonce+=1
blockchain.add_block(Block(3, date.datetime.now(), "Transaction Data 3", "",nonce))

# Print the contents of the blockchain
for block in blockchain.chain:
    print("Block #" + str(block.index))
    print("Timestamp: " + str(block.timestamp))
    print("Data: " + block.data)
    print("Hash: " + block.hash)
    print("nonce: " + str(block.nonce))
    print("Previous Hash: " + block.previous_hash)
    print("\n")
```


File Edit Selection View Go Run ···
Blockchain
EXPLORER
BLOCKCHAIN
block.py
block.py > Block > __init__
ESP-IDF: Search Error Hint
OUTLINE
TIMELINE
PROJECT COMPONENTS

```
import hashlib
import datetime as date
nonce=0
class Block:
    def __init__(self, index, timestamp, data, previous_hash,nonce):
        self.index = index
        self.timestamp = timestamp
        self.data = data
        self.previous_hash = previous_hash
        self.nonce=nonce
        self.hash = self.calculate_hash()

    def calculate_hash(self):
        str_hash = f"{self.data}{self.nonce}"
        result = hashlib.sha256(str_hash.encode())
        hash_hex = result.hexdigest()

        hash_string = str(self.index) + str(self.timestamp) + str(self.data) + str(self.previous_hash) + str(self.nonce)
        return hash_hex

class Blockchain:
    def __init__(self):
        self.chain = [self.create_genesis_block()]

    def create_genesis_block(self):
        return Block(0, date.datetime.now(), "Genesis Block", "0",nonce)

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

    def add_block(self, new_block):
        new_block.previous_hash = self.get_latest_block().hash
        new_block.hash = new_block.calculate_hash()
        self.chain.append(new_block)


    def is_valid(self):
        for i in range(1, len(self.chain)):
            current_block = self.chain[i]
            previous_block = self.chain[i-1]

            if current_block.hash != current_block.calculate_hash():
                return False

            if current_block.previous_hash != previous_block.hash:
                return False

        return True
# Create the blockchain
blockchain = Blockchain()
nonce+=1
# Add blocks to the blockchain
blockchain.add_block(Block(1, date.datetime.now(), "Transaction Data 1", "",nonce))
nonce+=1
blockchain.add_block(Block(2, date.datetime.now(), "Transaction Data 2", "",nonce))
```


File Edit Selection View Go Run ···
Blockchain
EXPLORER
BLOCKCHAIN
block.py
block.py > Block > __init__
ESP-IDF: Search Error Hint
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ESP-IDF
powershell
OUTLINE
TIMELINE
PROJECT COMPONENTS
Ln 11, Col 42 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit Go Live

```
import hashlib
import datetime as date
nonce=0
class Block:
    def __init__(self, index, timestamp, data, previous_hash,nonce):
        self.index = index
```

```
PS C:\Users\djsce.student\Desktop\Blockchain> python .\block.py
Block #0
Timestamp: 2024-08-12 13:03:13.305174
Data: Genesis Block
Hash: 8500b59bb5271135cd9bcbf0afd693028d76df3b9c7da58d412b13fc8a8f9394
nonce: 0
Previous Hash: 0


Block #1
Timestamp: 2024-08-12 13:03:13.305174
Data: Transaction Data 1
Hash: a331b191e16d80bb6d09e9ff0048bdfc53a7740d89603c143c4520a6de7eb6de
nonce: 1
nonce: 1
Previous Hash: 8500b59bb5271135cd9bcbf0afd693028d76df3b9c7da58d412b13fc8a8f9394
Previous Hash: 8500b59bb5271135cd9bcbf0afd693028d76df3b9c7da58d412b13fc8a8f9394



Block #2
Block #2
Timestamp: 2024-08-12 13:03:13.305174
Timestamp: 2024-08-12 13:03:13.305174
Data: Transaction Data 2
Hash: 68d94274db407013fff4fdfb0936af425fc74b596bc1198cdde8af9c1a4c5031
nonce: 2
Previous Hash: a331b191e16d80bb6d09e9ff0048bdfc53a7740d89603c143c4520a6de7eb6de
```