



COURSE CODE: (DJS22ADL7013)

COURSE NAME: Blockchain Technology Laboratory CLASS: B.Tech

EXPERIMENT NO. 2

CO/LO: Describe basic knowledge of Blockchain technology.

AIM / OBJECTIVE: To create a basic Blockchain with sample transactions and print it.

DESCRIPTION OF EXPERIMENT:

This experiment involves creating a basic blockchain using Python to understand its structure and working. Each block will contain sample transactions, a timestamp, and the hash of the previous block. The experiment prints the blockchain to demonstrate how data is securely linked in a chain..

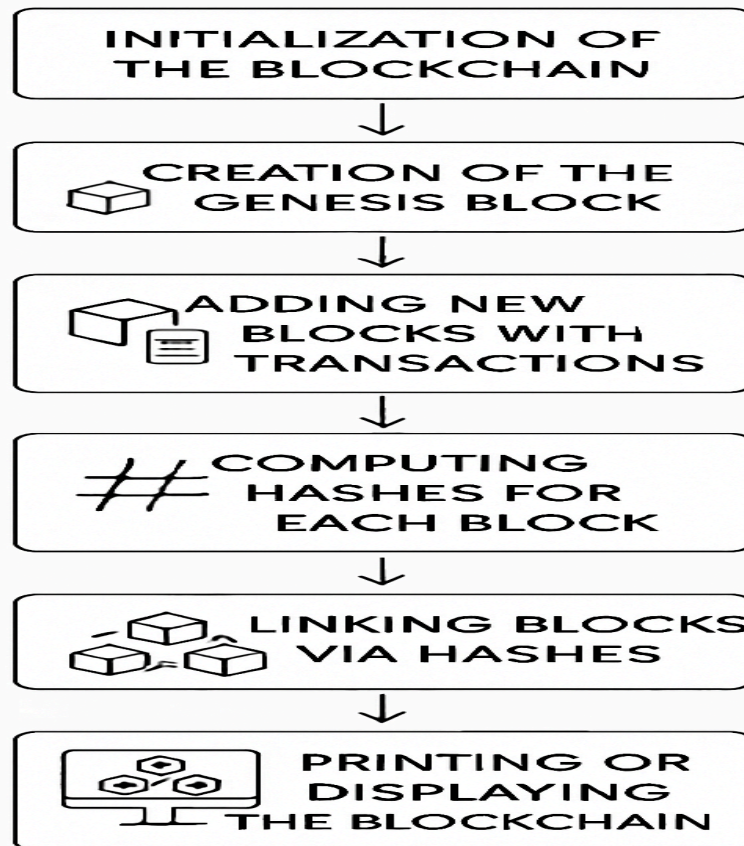
Overview of Libraries:

Library	Purpose
Hashlib	To generate SHA-256 hashes for securing each block.
datetime	To timestamp each block when it is created.
json	(Optional) To format and display blockchain data cleanly.

EXERCISE



Department: Artificial Intelligence (AI) and Data Science





Step:1 Simple Hashing

```
import hashlib
def create_hash(string):
    hash_object = hashlib.sha256()
    hash_object.update(string.encode('utf-8'))
    hash_string= hash_object.hexdigest()
    return hash_string

input_string = input("Enter a string: ")
hash_result = create_hash(input_string)
print("Hash: ",hash_result)
```

Enter a string: DASHRATH KALE

Hash: d7df8dd69ddc5b1bdf438a5d1086dba4bebc9bb82de23dcb69a3eb59620b0178

Step:2 Hashing with NONCE

```
import hashlib
input_string = input("Enter a string: ")
nonce = input("Enter a nonce: ")
hash_string = input_string + nonce
hash_object = hashlib.sha256(hash_string.encode('utf-8'))
hash_code= hash_object.hexdigest()
print("Hash: ",hash_code)
```

Enter a string: RAM SINGH

Enter a nonce: 123

Hash: fa8c4f0e1ac343f477924e01d66609e0196aa2b79a01deac77535b3fb847a167



Step 3: Hashing with random NONCE

```
import hashlib
import random
input_string = input("Enter a string: ")
nonce = str(random.randint(0,1000))
hash_string = input_string + nonce
hash_object = hashlib.sha256(hash_string.encode('utf-8'))
hash_code = hash_object.hexdigest()
print("NONCE: ", nonce)
print("Hash: ", hash_code)
```

Enter a string: SUDHIT SIR

NONCE: 716

Hash: 024ea26dfe01a1f2ce269101276c492c9ac9fd06fcd8046e155e3f8b8c87271a

Step 4: Simple Blockchain implementation



```
import hashlib
class Block:
    def __init__(self, transactions, previous_hash):
        self.transactions = transactions
        self.previous_hash = previous_hash
        self.hash = self.calculate_hash()
    def calculate_hash(self):
        data = (self.transactions) + str(self.previous_hash)
        return hashlib.sha256(data.encode()).hexdigest()
class Blockchain:
    def __init__(self):
        self.chain = []
        self.create_genesis_block()
    def create_genesis_block(self):
        transactions = "Genesis block"
        previous_hash = "0"
        self.chain.append(Block(transactions,previous_hash))
    def add_block(self, transactions):
        previous_block = self.chain[-1]
        previous_hash = previous_block.hash
        self.chain.append(Block(transactions,previous_hash))
    def print_chain(self):
        for block in self.chain:
            print("Transactions: ",block.transactions)
            print("Hash: " , block.hash)
            print("Previous Hash: " , block.previous_hash)
            print()
blockchain = Blockchain()
# Add blocks to the blockchain
blockchain.add_block("Transaction Data 1")
blockchain.add_block("Transaction Data 2")
blockchain.add_block("Transaction Data 3")
blockchain.add_block("Transaction Data 4")
blockchain.add_block("Transaction Data 5")
blockchain.print_chain()
```



Transactions: Genesis block

Hash: 29b4f27c76066fc0c3b4ef7617216202930ed69e99e42aa94da0565254852350

Previous Hash: 0

Transactions: Transaction Data 1

Hash: 501a452df5cbbccce5b68b4409dc2c713023945c25dda690ff7ff5178fd7201d

Previous Hash: 29b4f27c76066fc0c3b4ef7617216202930ed69e99e42aa94da0565254852350

Transactions: Transaction Data 2

Hash: 7b07691276851c652038e31ab13d64d55b8c00d57e9cb1a90aea867958997196

Previous Hash: 501a452df5cbbccce5b68b4409dc2c713023945c25dda690ff7ff5178fd7201d

Transactions: Transaction Data 3

Hash: 1956ad570f0f7d0c4875ee08391636a2ceaa296ccd9575d93f5138349cb76178

Previous Hash: 7b07691276851c652038e31ab13d64d55b8c00d57e9cb1a90aea867958997196

Transactions: Transaction Data 4

Hash: a0cd89368ddacc68a4dc6a89655a354dced4bb7b30389c20b152639617d1ef90

Previous Hash: 1956ad570f0f7d0c4875ee08391636a2ceaa296ccd9575d93f5138349cb76178

Transactions: Transaction Data 5

Hash: 0aea5431f66cfd5cc625a9dc406fdd2bf2d94f4d6e8183e1ba3af626d3dc253

Previous Hash: a0cd89368ddacc68a4dc6a89655a354dced4bb7b30389c20b152639617d1ef90

QUESTIONS:

1. Difference between Centralized Systems, Decentralized Systems, and Distributed Systems
2. Explain the types of Blockchain

REFERENCE:

Website References:

1. <https://www.youtube.com/watch?v=MViBvQXQ3mM>
2. <https://www.youtube.com/watch?v=o0hp-fRyLOc>