Blockchain Lab Exp 2

Name: Rohan Lalchandani **Class:** D20A
Roll no: 31 **Batch:** A

Aim: Create a Blockchain using Python

Theory:

1. What is a Blockchain?

A **Blockchain** is a modern digital technology used to store information in a secure and transparent way.

Definition

A blockchain is a **distributed digital ledger** that records transactions or data in the form of blocks, where each block is connected to the previous one using cryptographic hashes.

Key Features of Blockchain

- **Blocks contain data** (transactions, records, etc.)
- **Each block is linked** to the previous block, forming a chain
- **Decentralized system** (no single authority controls it)
- **Secure and tamper-proof** due to cryptography
- **Transparent and verifiable** by all participants

Structure of Blockchain

Each block generally contains:

- Transaction data
- Timestamp
- Hash of the current block
- Hash of the previous block

So, if one block is changed, the entire chain becomes invalid.

Uses of Blockchain

- Cryptocurrencies (Bitcoin, Ethereum)
- Supply chain tracking
- Digital identity verification
- Secure voting systems
- Smart contracts

2. Process of Mining:

1. **Transaction Collection**
   Miners collect new and unconfirmed transactions from the network.

2. **Block Formation**
   These transactions are grouped together to form a new block.

3. **Hash Calculation**
   Miners generate a cryptographic hash for the block using the block data.

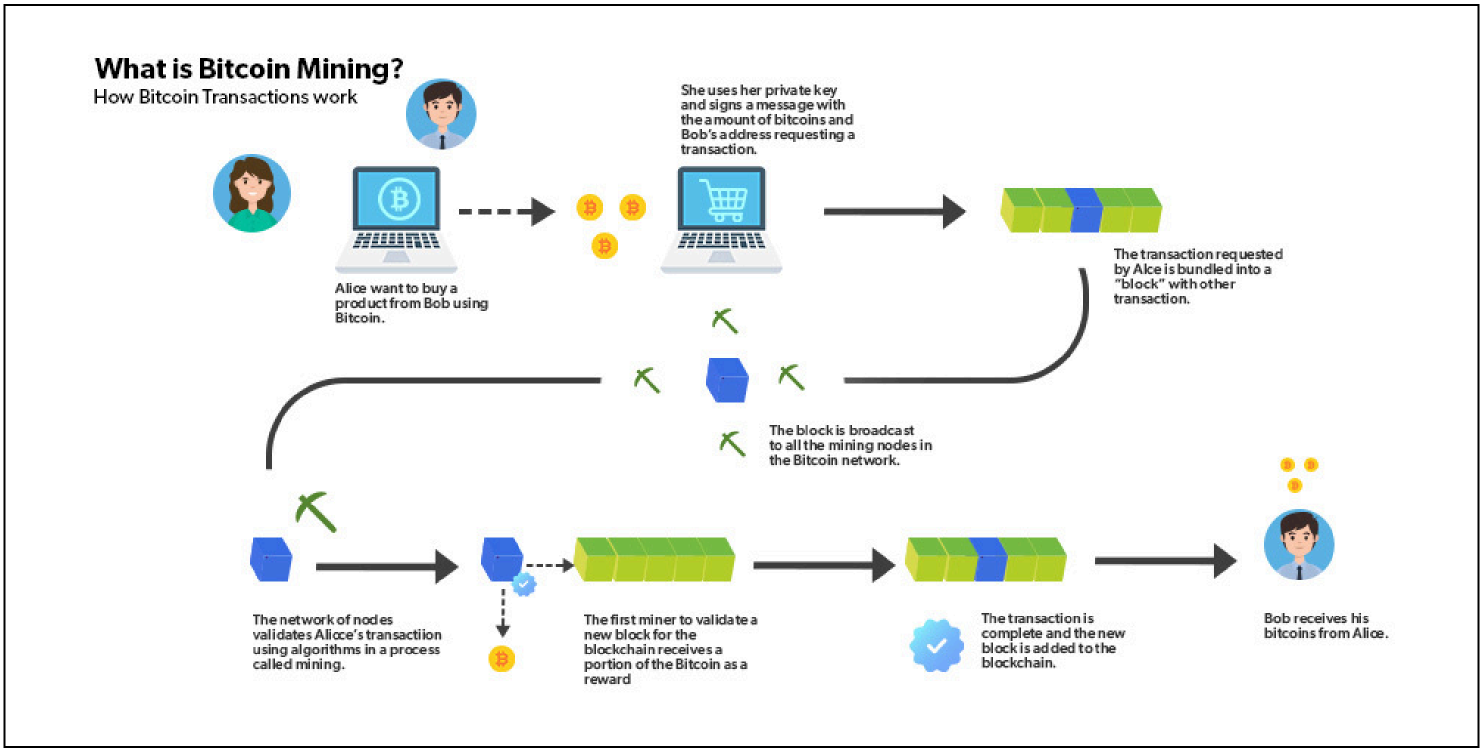
4. **Solving the Puzzle (Proof of Work)**
   Miners compete to find a special number called a **nonce** that produces a valid hash (meeting difficulty conditions).

5. **Validation by Network**
   Once a miner finds the correct solution, the block is broadcast to the network for verification.

6. **Block Addition to Blockchain**
   After verification, the new block is added to the existing blockchain.

7. **Reward to Miner**
   The successful miner receives a reward in the form of cryptocurrency and transaction fees.

8. **Chain Continues**
   The mining process repeats for the next set of transactions and blocks.



What is Bitcoin Mining?

How Bitcoin Transactions work

Alice want to buy a product from Bob using Bitcoin.

She uses her private key and signs a message with the amount of bitcoins and Bob's address requesting a transaction.

The transaction requested by Alce is bundled into a "block" with other transaction.

The block is broadcast to all the mining nodes in the Bitcoin network.

The network of nodes validates Alicce's transactiion using algorithms in a process called mining.

The first miner to validate a new block for the blockchain receives a portion of the Bitcoin as a reward

The transaction is complete and the new block is added to the blockchain.

Bob receives his bitcoins from Alice.

3. How to Check the Validity of Blocks in a Blockchain

The validity of blocks in a blockchain is checked to ensure that the data is secure and has not been tampered with. This is done through the following methods:

1. **Hash Verification**
   Each block has its own unique hash. The system recalculates the hash to confirm that the block data has not been changed.

2. **Previous Hash Matching**
   Every block contains the hash of the previous block. If the previous hash stored in the block matches the actual hash of the previous block, the chain remains valid.

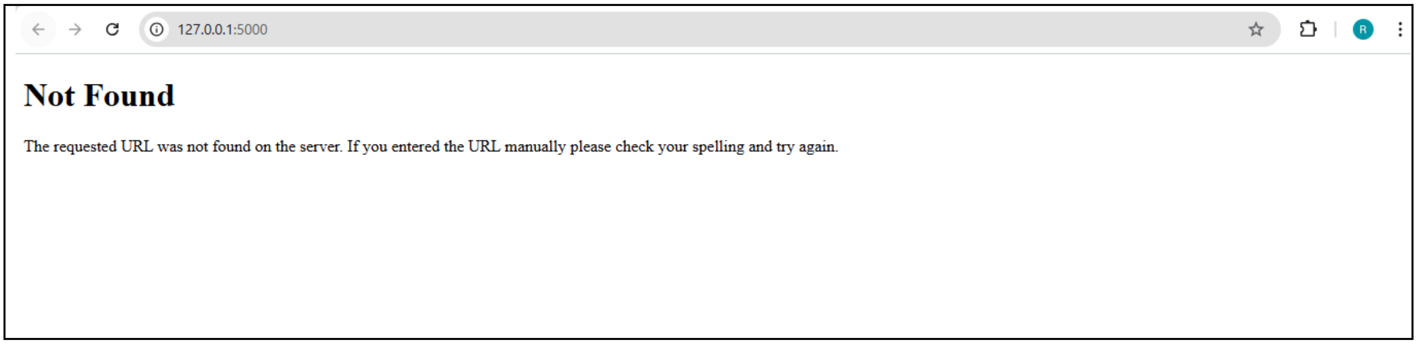
3. **Transaction Validation**
   All transactions inside the block are checked to ensure they are genuine, properly signed, and follow network rules.

4. **Consensus Mechanism Check**
   The block must satisfy the consensus protocol (like Proof of Work or Proof of Stake) to prove it was added legitimately.

5. **Merkle Root Verification**
   The Merkle root stored in the block header is verified to ensure all transactions inside the block are correct and unchanged.

6. **Timestamp and Block Structure Check**
   The timestamp and format of the block are examined to confirm it follows blockchain standards.

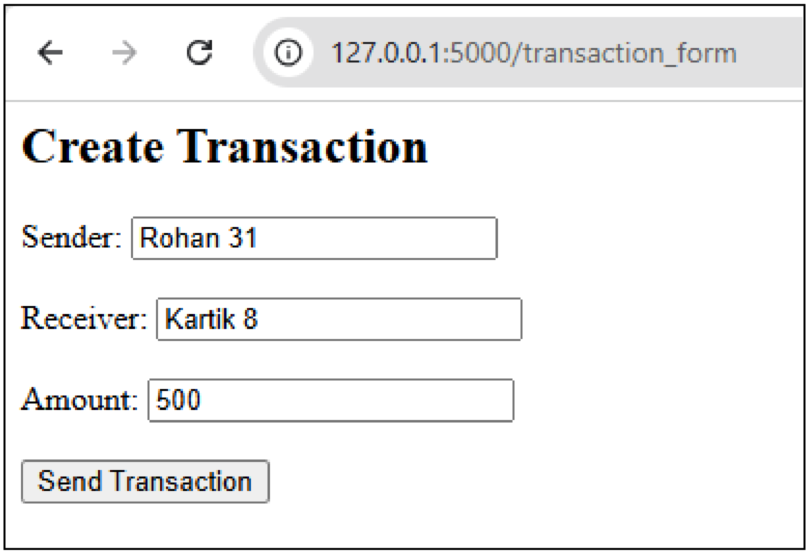
Implementation:



127.0.0.1:5000

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.



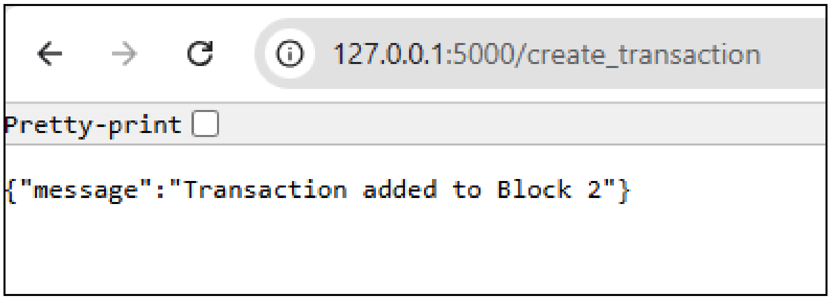
127.0.0.1:5000/transaction_form

Create Transaction

Sender: Rohan 31

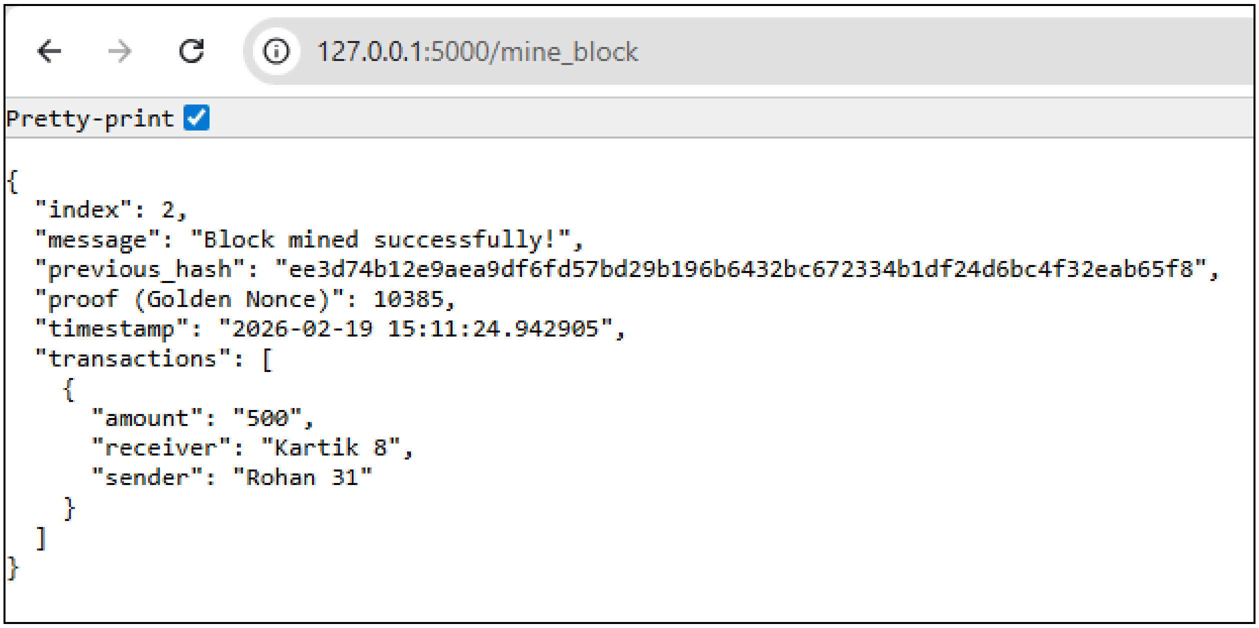
Receiver: Kartik 8

Amount: 500



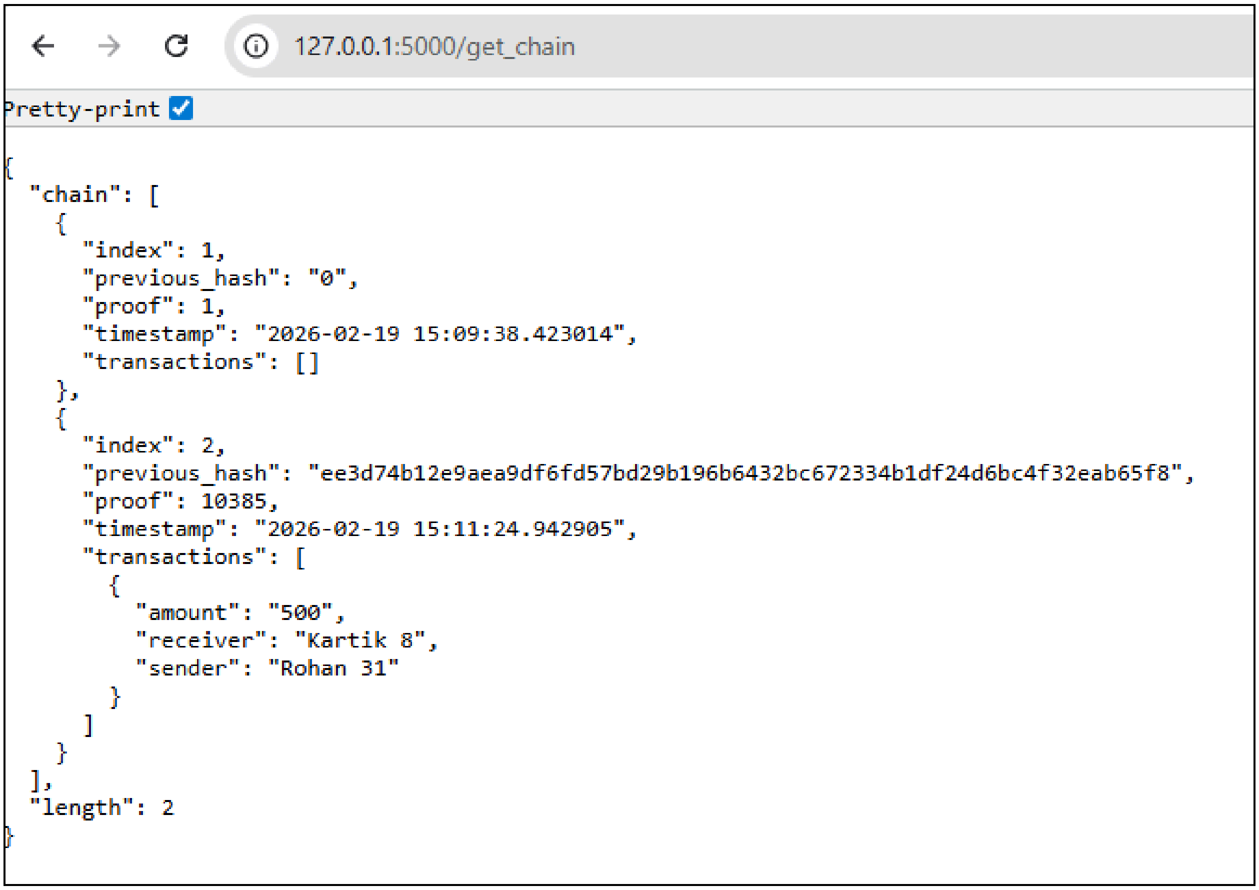
127.0.0.1:5000/create_transaction

```
{"message":"Transaction added to Block 2"}
```



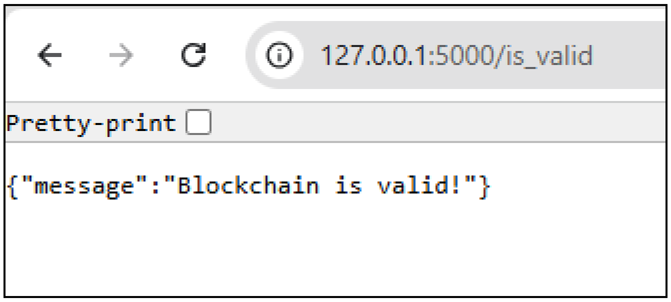
Pretty-print ☑

```
{
  "index": 2,
  "message": "Block mined successfully!",
  "previous_hash": "ee3d74b12e9aea9df6fd57bd29b196b6432bc672334b1df24d6bc4f32eab65f8",
  "proof (Golden Nonce)": 10385,
  "timestamp": "2026-02-19 15:11:24.942905",
  "transactions": [
    {
      "amount": "500",
      "receiver": "Kartik 8",
      "sender": "Rohan 31"
    }
  ]
}
```



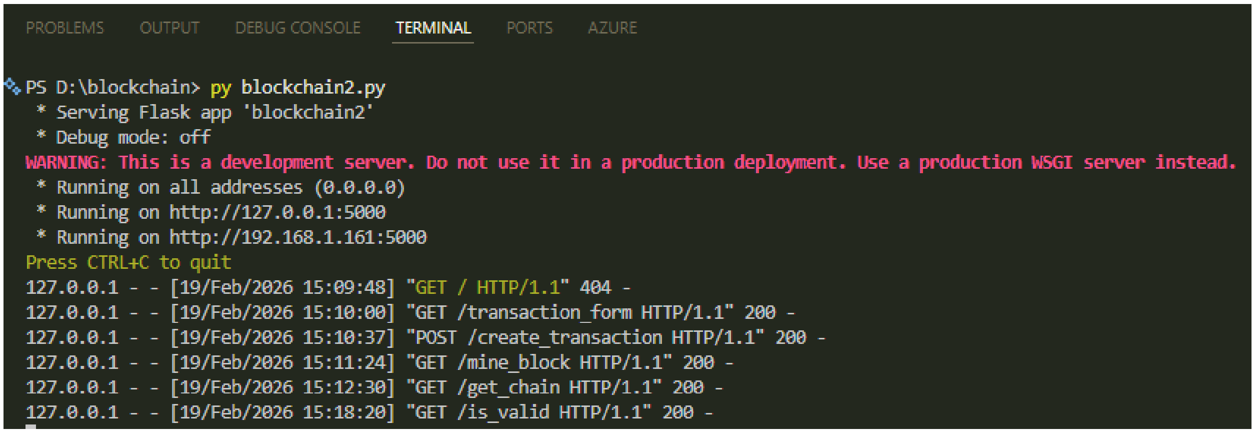
Pretty-print ☑

```
{
  "chain": [
    {
      "index": 1,
      "previous_hash": "0",
      "proof": 1,
      "timestamp": "2026-02-19 15:09:38.423014",
      "transactions": []
    },
    {
      "index": 2,
      "previous_hash": "ee3d74b12e9aea9df6fd57bd29b196b6432bc672334b1df24d6bc4f32eab65f8",
      "proof": 10385,
      "timestamp": "2026-02-19 15:11:24.942905",
      "transactions": [
        {
          "amount": "500",
          "receiver": "Kartik 8",
          "sender": "Rohan 31"
        }
      ]
    }
  ],
  "length": 2
}
```



Pretty-print □

```
{"message":"Blockchain is valid!"}
```



```
PS D:\blockchain> py blockchain2.py
 * Serving Flask app 'blockchain2'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.1.161:5000
Press CTRL+C to quit
127.0.0.1 - - [19/Feb/2026 15:09:48] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [19/Feb/2026 15:10:00] "GET /transaction_form HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2026 15:10:37] "POST /create_transaction HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2026 15:11:24] "GET /mine_block HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2026 15:12:30] "GET /get_chain HTTP/1.1" 200 -
127.0.0.1 - - [19/Feb/2026 15:18:20] "GET /is_valid HTTP/1.1" 200 -
```

Conclusion:

In this practical, we successfully enhanced the blockchain application by adding transaction support and implementing a mining mechanism using the golden nonce concept. Blocks are mined only when pending transactions exist, ensuring meaningful block creation. The proof-of-work algorithm was modified to solve a cryptographic puzzle requiring the block hash to start with “000”, improving security and demonstrating how mining validates and strengthens blockchain integrity.