# BLOCKCHAIN EXPERIMENT - 03
## AAYUSH TALREJA (D17C/56)

## AIM

Create a Cryptocurrency using Python and perform mining in the Blockchain created

## THEORY

### Blockchain

- A blockchain is a distributed, decentralized ledger that records transactions across multiple computers in a secure and tamper-resistant manner.
- Each block in the blockchain contains a set of transactions, a timestamp, and a reference to the previous block (forming a chain).
- Blockchain technology ensures transparency, security, and immutability of data.

### Public vs. Private Blockchain

- Public Blockchains (e.g., Bitcoin, Ethereum): Open to anyone, permissionless, and decentralized.
- Private Blockchains: Restricted access, typically used within organizations, and often permissioned.

### Peer-to-Peer (P2P) Networks

- P2P networks consist of nodes (computers) that communicate directly with each other without relying on a central server.
- In blockchain, P2P networks facilitate the distribution of the ledger and consensus among nodes.

### Transactions in a Blockchain

- Transactions represent the transfer of assets (cryptocurrencies or data) between participants on the blockchain.
- Each transaction includes sender and receiver addresses, amount, and a digital signature to verify authenticity.

### Mempools

- Mempools (Memory Pools) are temporary storage areas within nodes where pending transactions are held before being included in a block.
- Miners select transactions from the mempool to add to the next block they mine.

## Tools and Libraries

### Flask

- Flask is a Python web framework used to create the HTTP server for your blockchain nodes.

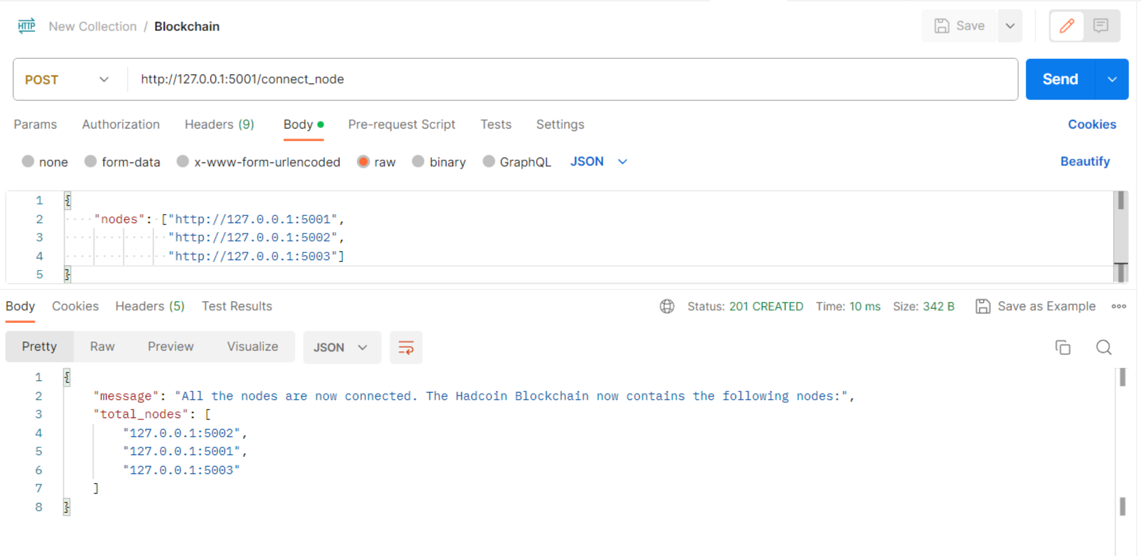
### Postman

- Postman is a popular API testing tool that allows you to make HTTP requests to interact with your blockchain nodes.

### Python Libraries

- `**datetime**`: Used for timestamping blocks.
- `**jsonify**`: Helps with converting data to JSON format for API responses.
- `**hashlib**`: Used for hashing data within blocks.
- `**uuid4**`: Generates unique identifiers for transactions or nodes.
- `**urlparse**`: Parses URLs for managing peers.
- `**requests**`: Makes HTTP requests to interact with other nodes in the network.

## OUTPUT

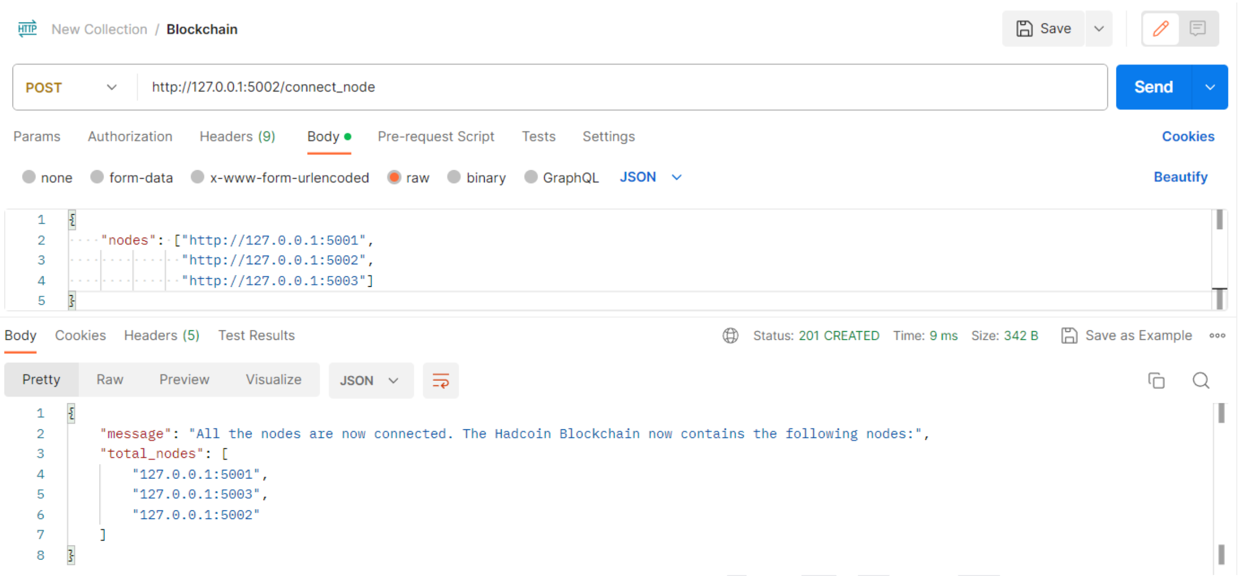


POST http://127.0.0.1:5001/connect_node

```
{
    "nodes": ["http://127.0.0.1:5001",
              "http://127.0.0.1:5002",
              "http://127.0.0.1:5003"]
}
```

Status: 201 CREATED Time: 10 ms Size: 342 B

```
{
    "message": "All the nodes are now connected. The Hadcoin Blockchain now contains the following nodes:",
    "total_nodes": [
        "127.0.0.1:5002",
        "127.0.0.1:5001",
        "127.0.0.1:5003"
    ]
}
```

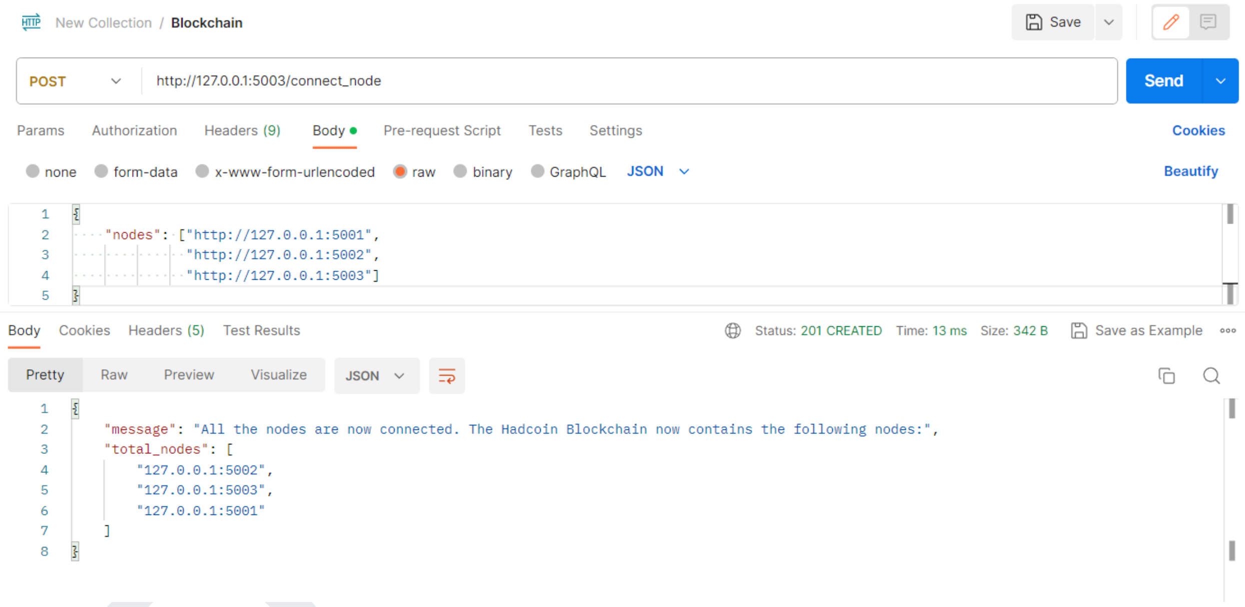


POST http://127.0.0.1:5002/connect_node

```
{
    "nodes": ["http://127.0.0.1:5001",
              "http://127.0.0.1:5002",
              "http://127.0.0.1:5003"]
}
```

Status: 201 CREATED Time: 9 ms Size: 342 B

```
{
    "message": "All the nodes are now connected. The Hadcoin Blockchain now contains the following nodes:",
    "total_nodes": [
        "127.0.0.1:5001",
        "127.0.0.1:5003",
        "127.0.0.1:5002"
    ]
}
```



POST http://127.0.0.1:5003/connect_node

```
{
    "nodes": ["http://127.0.0.1:5001",
              "http://127.0.0.1:5002",
              "http://127.0.0.1:5003"]
}
```

Status: 201 CREATED Time: 13 ms Size: 342 B

```
{
    "message": "All the nodes are now connected. The Hadcoin Blockchain now contains the following nodes:",
    "total_nodes": [
        "127.0.0.1:5002",
        "127.0.0.1:5003",
        "127.0.0.1:5001"
    ]
}
```

```
{
  "chain": [
    {
      "index": 1,
      "previous_hash": "0",
      "proof": 1,
      "timestamp": "2023-08-17 12:45:33.901314",
      "transactions": []
    },
    {
      "index": 2,
      "previous_hash": "ad06b2e37c53b64610a7f1b82ce13f6e9df5a850250dd743e71bde58973ab620",
      "proof": 533,
      "timestamp": "2023-08-17 12:52:14.432105",
      "transactions": [
        {
          "amount": 1000,
          "receiver": "Aayush",
          "sender": "Talreja"
        },
        {
          "amount": 1,
          "receiver": "you",
          "sender": "276d870ada2b41849e9289bd2a2c6176"
        }
      ]
    }
  ],
  "length": 2
}
```

**CONCLUSION**

This experiment  helped us understand the fundamental concepts of blockchain technology, including peer-to-peer networks, transaction handling, and the role of mempools. It also introduced us to the tools and libraries commonly used in blockchain development. Additionally, we gained hands-on experience in creating and interacting with a blockchain network.