Name: Vemburaj Konar **Class**: D16AD **Roll**: 28

Blockchain Lab 6

Aim: To develop a web-based gateway that serves as an entry point for users to access and interact with Ethereum Mainnet.

Theory:

Etherscan is a popular blockchain explorer specifically designed for the Ethereum blockchain. It serves as a search engine, analytics platform, and API provider for Ethereum transactions, addresses, tokens, decentralized applications (DApps), and smart contracts.

Working of Etherscan:

1. Data Retrieval: Etherscan continuously synchronizes with the Ethereum blockchain, collecting and indexing data like transactions, contract interactions, and token transfers.

2. User Interface: Users can search for specific transactions, addresses, or smart contracts using Etherscan's user-friendly interface. They can view real-time transaction details, contract source code, and token balances.

3. Verification: Etherscan provides a platform for developers to verify their smart contracts' source code, ensuring transparency and security. Verified contracts show that the deployed code matches the published source.

4. API Services: Etherscan offers APIs that developers can use to access Ethereum data programmatically. These APIs allow applications to interact with the Ethereum blockchain without running a full node.

Why Do We Need Etherscan:

1. Transparency: Etherscan enhances the transparency of the Ethereum network. Anyone can inspect transactions, ensuring accountability and trust in the ecosystem.

2. Verification: Developers can validate and showcase their smart contract source code, assuring users that the code has not been tampered with and operates as intended.

3. Research and Analysis: Researchers and analysts use Etherscan's data to gain insights into market trends, user behavior, and the overall health of the Ethereum network.

4. Monitoring: Users and businesses can monitor their Ethereum addresses, transactions, and contracts, ensuring the security of their assets and verifying the success of transactions.

In summary, Etherscan is a crucial tool for Ethereum users, developers, and researchers, offering a transparent view of the blockchain, allowing verification of smart contracts, and providing essential data and insights to the Ethereum community.

Program:

```
import requests
import datetime
now = datetime.datetime.now()
def get_latest_block(api_key):
    url = "https://api.etherscan.io/api"
    params = {
        "module": "proxy",
        "action": "eth_getBlockByNumber",
        "tag": "latest",
        "boolean": "true",
        "apikey": api_key,
    }
```

```
    try:
        response = requests.get(url, params=params)
        if response.status_code == 200:
            data = response.json()
            return data["result"]
        else:
            print("Request failed with status code:",
response.status_code)
    except requests.RequestException as e:
        print("Request failed:", str(e))


    return None
# Replace "YOUR_API_KEY" with your actual API key
api_key = "361N7FBI6W9BSCSDSJWPIM6ZI9BB5N989V"            # Paste your API
Key
print ("Current date and time : ", now.strftime("%d-%B-%Y"))
latest_block = get_latest_block(api_key)
#print(latest_block)
if latest_block is not None:
    print("Latest block information:")
    print("Block Number:", int(latest_block["number"], 16))
    print("Timestamp:", int(latest_block["timestamp"], 16))
    print("Miner Address:", latest_block["miner"])
    print("Difficulty:", int(latest_block["difficulty"], 16))
    print("Total Difficulty:", int(latest_block["totalDifficulty"], 16))
```

```
    print("Gas Limit:", int(latest_block["gasLimit"], 16))

    print("Gas Used:", int(latest_block["gasUsed"], 16))

    print("Transaction Count:", len(latest_block["transactions"]))

    print("Transactions:", latest_block["transactions"])
```

```
Current date and time :  12-October-2023
Latest block information:
Block Number: 18332228
Timestamp: 1697088287
Miner Address: 0x4838b106fce9647bdf1e7877bf73ce8b0bad5f97
Difficulty: 0
Total Difficulty: 58750003716598352816469
Gas Limit: 30000000
Gas Used: 14733602
Transaction Count: 133
Transactions: [{'blockHash': '0x7cd42858b5c23595f7a48bc53edcc9694c19c058b3d839cf2e4578a405332563', 'blockNumber': '0x117ba44', 'from': '0x971ac48e6f904
```

```
import requests

# Infura HTTP endpoint

infura_url =
'https://mainnet.infura.io/v3/7329beded7a74ad085a6144b63645314'

# Make a request to retrieve the latest block number

response = requests.post(

    infura_url,

    json={

        "jsonrpc": "2.0",

        "method": "eth_blockNumber",

        "params": [],

        "id": 1

    }

)

if response.status_code == 200:

    result = response.json()
```

```
    latest_block_number = int(result["result"], 16)  # Convert hexadecimal
to decimal

    print("Latest Block Number:",latest_block_number)

    # Print the desired information

    print("Miner Address:", latest_block["miner"]) # Error

    print("Difficulty:", int(latest_block["difficulty"], 16))

    print("Total Difficulty:", int(latest_block["totalDifficulty"], 16))

    print("Gas Limit:", int(latest_block["gasLimit"], 16))

    print("Gas Used:", int(latest_block["gasUsed"], 16))

    print("Transaction Count:", len(latest_block["transactions"]))

else:

    print("Failed to retrieve the latest block. Error:", response.text)
```

```
Latest Block Number: 18332229
Miner Address: 0x4838b106fce9647bdf1e7877bf73ce8b0bad5f97
Difficulty: 0
Total Difficulty: 58750003716598352816469
Gas Limit: 30000000
Gas Used: 14733602
Transaction Count: 133
```

Conclusion: Thus we have successfully implemented program to develop a web-based gateway that serves as an entry point for users to access and interact with Ethereum Mainnet.