Batch: BCT_1 Roll No.: 1911012

Experiment No. 6

Title: Private Ethereum Blockchain environment

Objective: To create Private Ethereum Blockchain environment

Expected Outcome of Experiment:

CO	Outcome
CO3	Apply the algorithm and techniques used in Blockchain

Books/ Journals/ Websites referred:

1. https://remix.ethereum.org/
2. https://dev.to/jeffersonxavier/create-your-own-private-blockchain-using-ethereum-52o5
3. https://geth.ethereum.org/docs/interface/private-network
4. https://code.visualstudio.com/docs

Abstract:-

The Ethereum private blockchain is a blockchain like ethereum, but the network used is not the Ethereum Main. When you use a private network all blockchain is totally apart from the Ethereum blockchain, it's possible change the configurations like mining difficult, access and other aspects.

A private network is composed of multiple Ethereum nodes that can only connect to each other. In order to run multiple nodes locally, each one requires a separate data directory (--datadir). The nodes must also know about each other and be able to exchange information, share an initial state and a common consensus algorithm.

Here we use the Go etheruem to construct private blockchain .

Related Theory: -

Blockchain is the backbone Technology of Digital CryptoCurrency BitCoin. The blockchain is a distributed database of records of all transactions or digital event that have been executed and shared among participating parties. Each transaction verified by the majority of participants of the system. It contains every single record of each transaction. BitCoin is the most popular cryptocurrency an example of the blockchain. Blockchain Technology first came to light when a person or Group of individuals name ‘Satoshi Nakamoto’ published a white paper on “BitCoin: A peer to peer electronic cash system” in 2008. Blockchain Technology Records Transaction in Digital Ledger which is distributed over the Network thus making it incorruptible. Anything of value like Land Assets, Cars, etc. can be recorded on Blockchain as a Transaction.

Blockchain enhances trust across a business network. It’s not that you can’t trust those who you conduct business with its that you don’t need to when operating on a Blockchain network.

Blockchain builts trust through the following five attributes:

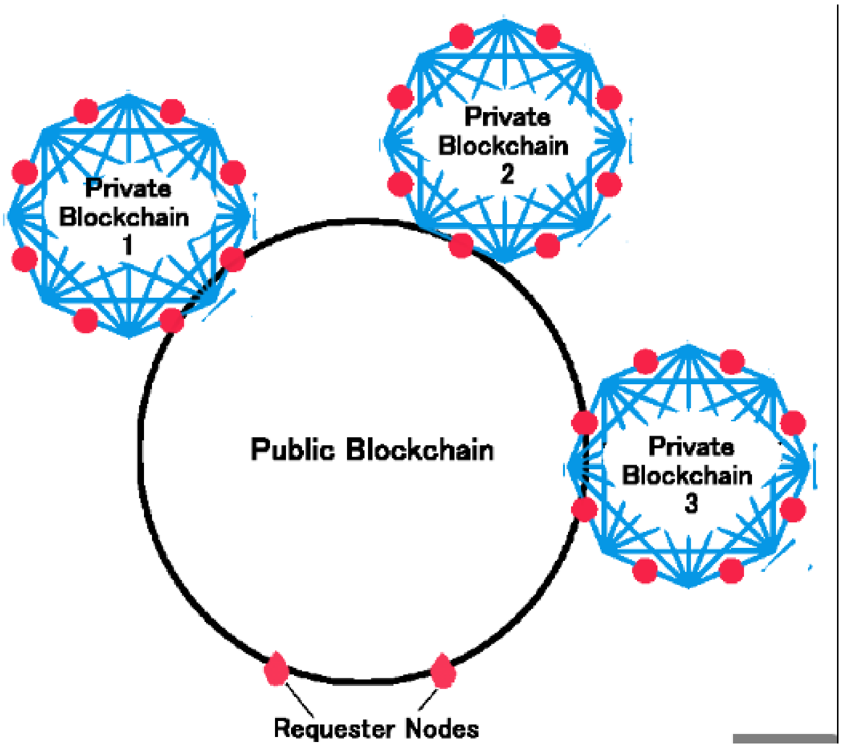
Distributed: The distributed ledger is shared and updated with every incoming transaction among the nodes connected to the Blockchain. All this is done in real-time as there is no central server controlling the data.

Secure: There is no unauthorized access to Blockchain made possible through Permissions and Cryptography.

Transparent: Because every node or participant in Blockchain has a copy of the Blockchain data, they have access to all transaction data. They themselves can verify the identities without the need for mediators.

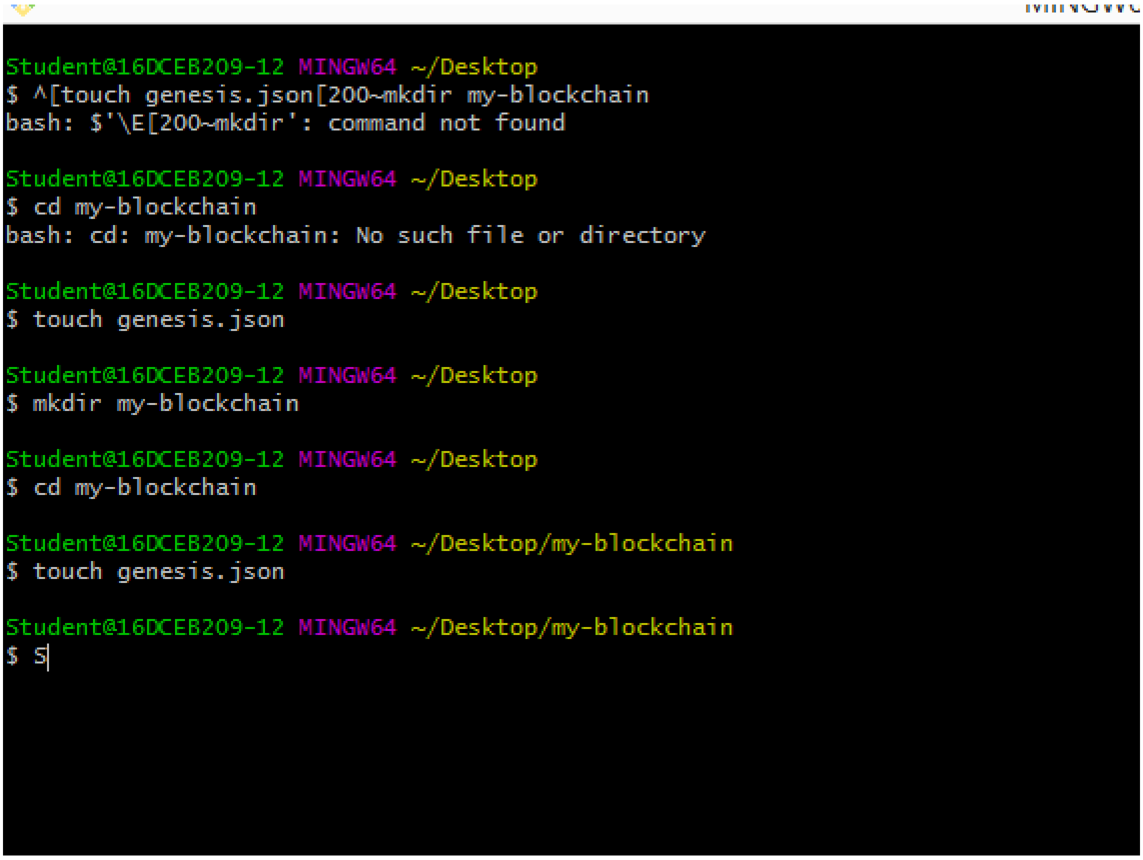
Consensus-based: All relevant network participants must agree that a transaction is valid. This is achieved through the use of consensus algorithms.

Flexible: Smart Contracts which are executed based on certain conditions can be written into the platform. Blockchain Network can evolve in pace with business processes.

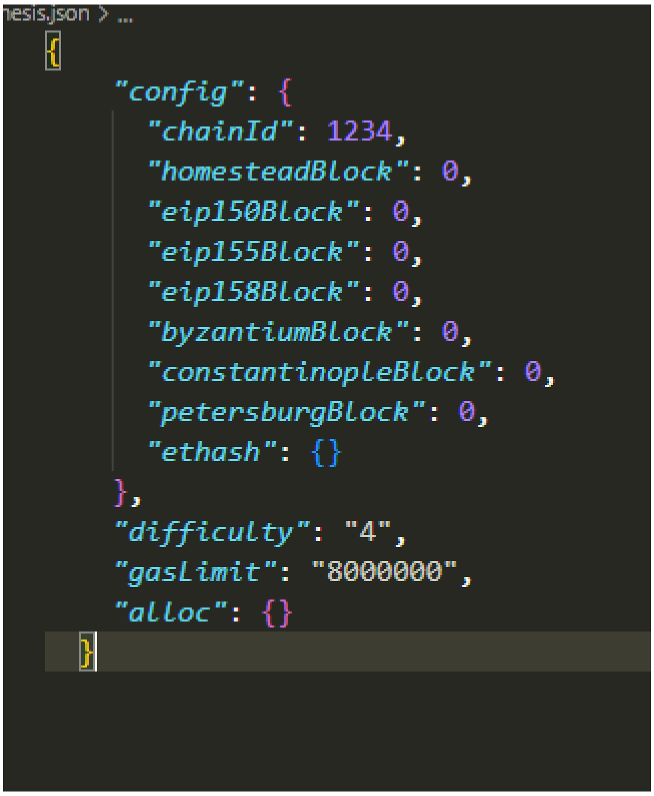


Implementation Details:

1. **Enlist all the Steps followed and various options explored**
    - **Install Geth**
    - **Define Genesis Block**

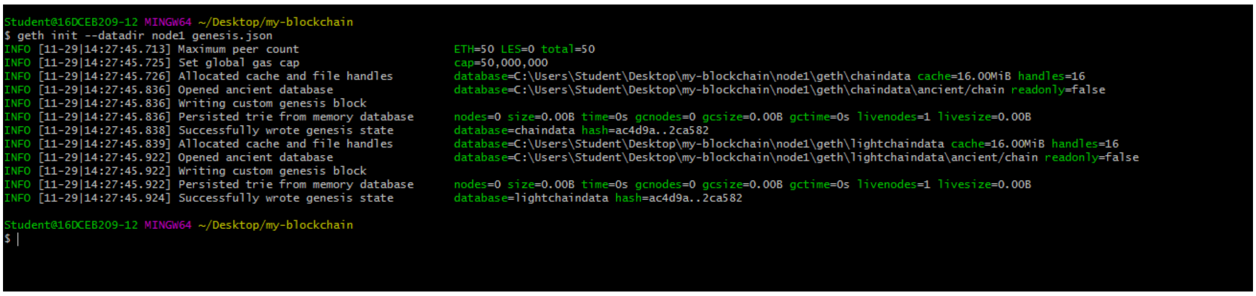


```
Student@16DCEB209-12 MINGW64 ~/Desktop
$ ^[touch genesis.json[200~mkdir my-blockchain
bash: $'\E[200~mkdir': command not found

Student@16DCEB209-12 MINGW64 ~/Desktop
$ cd my-blockchain
bash: cd: my-blockchain: No such file or directory

Student@16DCEB209-12 MINGW64 ~/Desktop
$ touch genesis.json

Student@16DCEB209-12 MINGW64 ~/Desktop
$ mkdir my-blockchain

Student@16DCEB209-12 MINGW64 ~/Desktop
$ cd my-blockchain

Student@16DCEB209-12 MINGW64 ~/Desktop/my-blockchain
$ touch genesis.json

Student@16DCEB209-12 MINGW64 ~/Desktop/my-blockchain
$ S
```

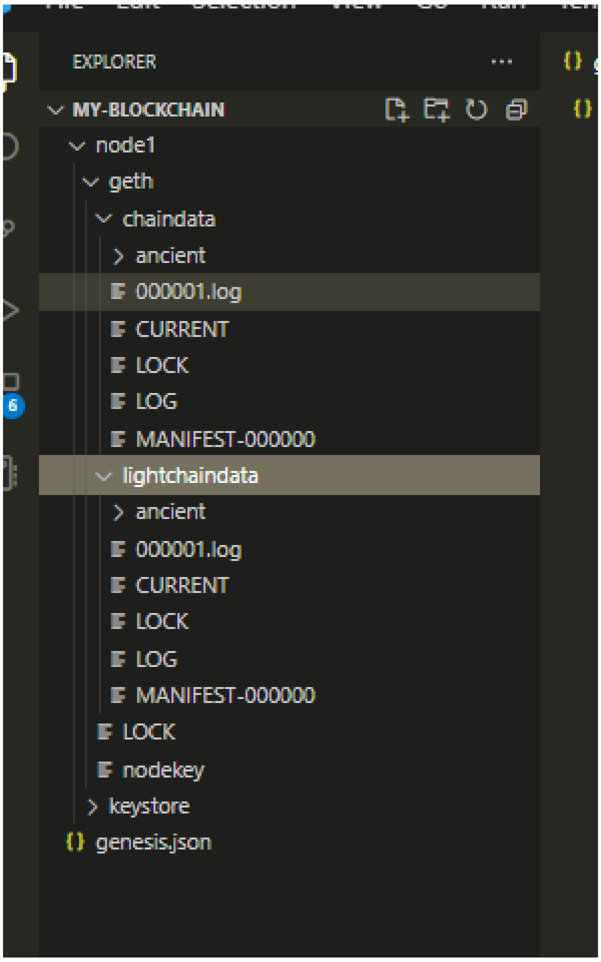


```
{
    "config": {
      "chainId": 1234,
      "homesteadBlock": 0,
      "eip150Block": 0,
      "eip155Block": 0,
      "eip158Block": 0,
      "byzantiumBlock": 0,
      "constantinopleBlock": 0,
      "petersburgBlock": 0,
      "ethash": {}
    },
    "difficulty": "4",
    "gasLimit": "8000000",
    "alloc": {}
}
```

- **Start database**

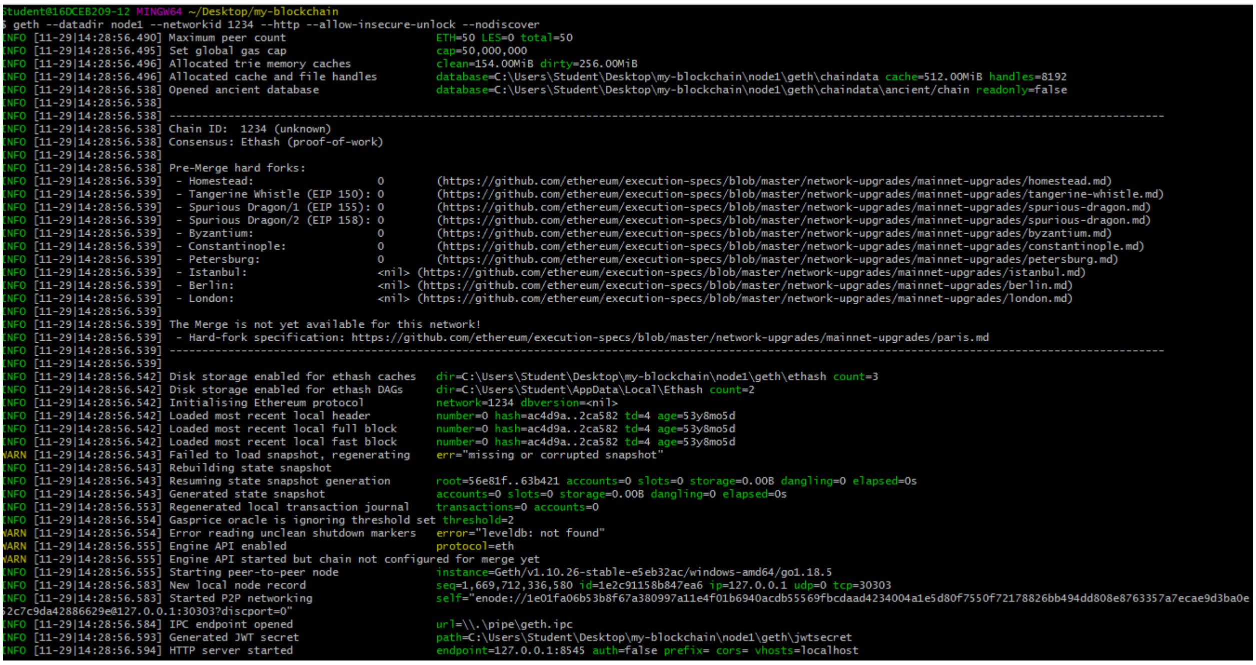


```
Student@16DCEB209-12 MINGW64 ~/Desktop/my-blockchain
$ geth init --datadir node1 genesis.json
INFO [11-29|14:27:45.713] Maximum peer count                       ETH=50 LES=0 total=50
INFO [11-29|14:27:45.725] Set global gas cap                       cap=50,000,000
INFO [11-29|14:27:45.726] Allocated cache and file handles         database=C:\Users\Student\Desktop\my-blockchain\node1\geth\chaindata cache=16.00MiB handles=16
INFO [11-29|14:27:45.836] Opened ancient database                  database=C:\Users\Student\Desktop\my-blockchain\node1\geth\chaindata\ancient/chain readonly=false
INFO [11-29|14:27:45.836] Writing custom genesis block
INFO [11-29|14:27:45.836] Persisted trie from memory database      nodes=0 size=0.00B time=0s gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [11-29|14:27:45.838] Successfully wrote genesis state         database=chaindata hash=ac4d9a..2ca582
INFO [11-29|14:27:45.839] Allocated cache and file handles         database=C:\Users\Student\Desktop\my-blockchain\node1\geth\lightchaindata cache=16.00MiB handles=16
INFO [11-29|14:27:45.922] Opened ancient database                  database=C:\Users\Student\Desktop\my-blockchain\node1\geth\lightchaindata\ancient/chain readonly=false
INFO [11-29|14:27:45.922] Writing custom genesis block
INFO [11-29|14:27:45.922] Persisted trie from memory database      nodes=0 size=0.00B time=0s gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [11-29|14:27:45.924] Successfully wrote genesis state         database=lightchaindata hash=ac4d9a..2ca582

Student@16DCEB209-12 MINGW64 ~/Desktop/my-blockchain
$ |
```



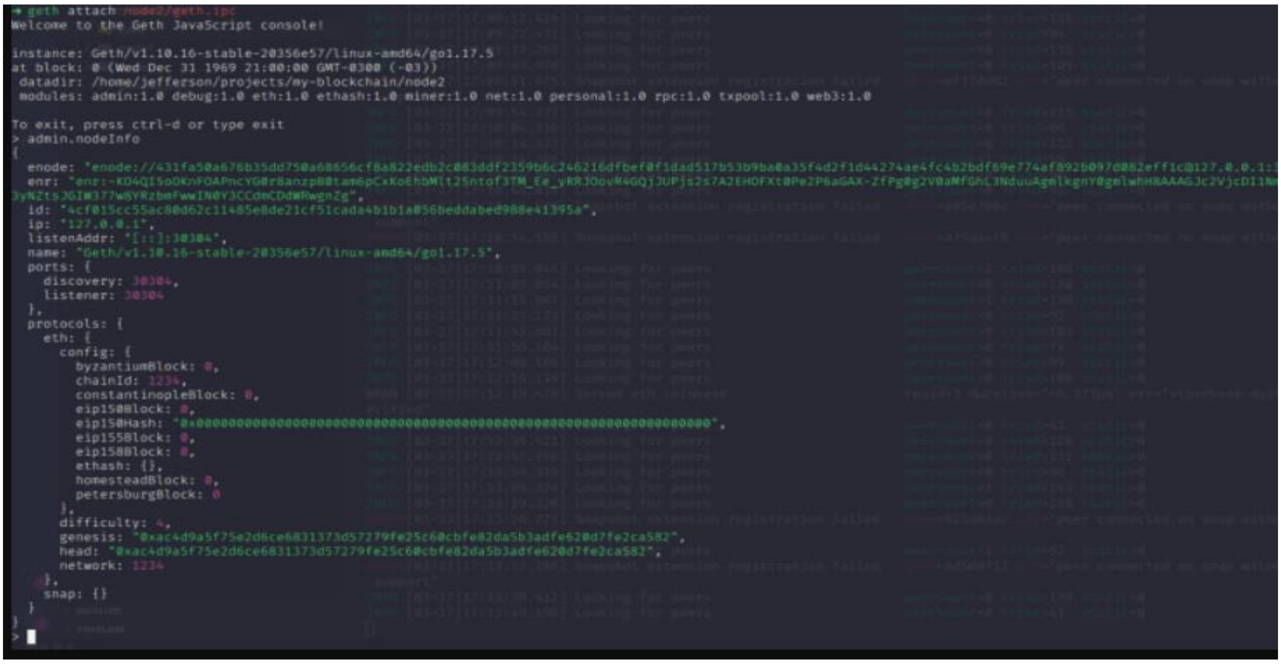
```
MY-BLOCKCHAIN
  node1
    geth
      chaindata
        ancient
        000001.log
        CURRENT
        LOCK
        LOG
        MANIFEST-000000
      lightchaindata
        ancient
        000001.log
        CURRENT
        LOCK
        LOG
        MANIFEST-000000
      LOCK
      nodekey
    keystore
  genesis.json
```

- **Start the first blockchain node**



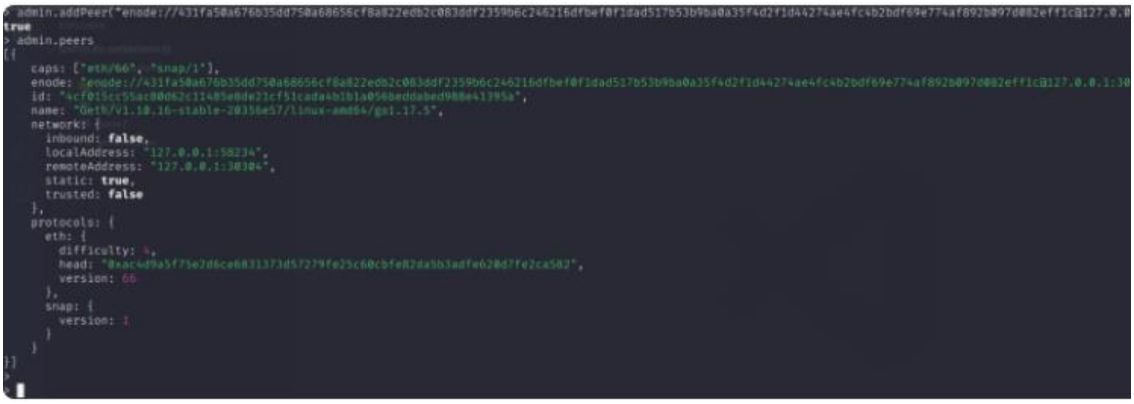
```
Student@16DCEB209-12 MINGW64 ~/Desktop/my-blockchain
$ geth --datadir node1 --networkid 1234 --http --allow-insecure-unlock --nodiscover
INFO [11-29|14:28:56.490] Maximum peer count                       ETH=50 LES=0 total=50
INFO [11-29|14:28:56.495] Set global gas cap                       cap=50,000,000
INFO [11-29|14:28:56.496] Allocated trie memory caches             clean=154.00MiB dirty=256.00MiB
INFO [11-29|14:28:56.496] Allocated cache and file handles         database=C:\Users\Student\Desktop\my-blockchain\node1\geth\chaindata cache=512.00MiB handles=8192
INFO [11-29|14:28:56.538] Opened ancient database                  database=C:\Users\Student\Desktop\my-blockchain\node1\geth\chaindata\ancient/chain readonly=false
INFO [11-29|14:28:56.538]
INFO [11-29|14:28:56.538] ----------------------------------------------------------------------------------------------------------------------------------------------------
INFO [11-29|14:28:56.538] Chain ID:  1234 (unknown)
INFO [11-29|14:28:56.538] Consensus: Ethash (proof-of-work)
INFO [11-29|14:28:56.538]
INFO [11-29|14:28:56.538] Pre-Merge hard forks:
INFO [11-29|14:28:56.539]  - Homestead:                   0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/homestead.md)
INFO [11-29|14:28:56.539]  - Tangerine Whistle (EIP 150): 0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/tangerine-whistle.md)
INFO [11-29|14:28:56.539]  - Spurious Dragon/1 (EIP 155): 0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md)
INFO [11-29|14:28:56.539]  - Spurious Dragon/2 (EIP 158): 0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md)
INFO [11-29|14:28:56.539]  - Byzantium:                   0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/byzantium.md)
INFO [11-29|14:28:56.539]  - Constantinople:              0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/constantinople.md)
INFO [11-29|14:28:56.539]  - Petersburg:                  0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/petersburg.md)
INFO [11-29|14:28:56.539]  - Istanbul:                    <nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/istanbul.md)
INFO [11-29|14:28:56.539]  - Berlin:                      <nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/berlin.md)
INFO [11-29|14:28:56.539]  - London:                      <nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/london.md)
INFO [11-29|14:28:56.539]
INFO [11-29|14:28:56.539] The Merge is not yet available for this network!
INFO [11-29|14:28:56.539]  - Hard-fork specification: https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md
INFO [11-29|14:28:56.539] ----------------------------------------------------------------------------------------------------------------------------------------------------
INFO [11-29|14:28:56.539]
INFO [11-29|14:28:56.542] Disk storage enabled for ethash caches   dir=C:\Users\Student\Desktop\my-blockchain\node1\geth\ethash count=3
INFO [11-29|14:28:56.542] Disk storage enabled for ethash DAGs     dir=C:\Users\Student\AppData\Local\Ethash count=2
INFO [11-29|14:28:56.542] Initialising Ethereum protocol           network=1234 dbversion=<nil>
INFO [11-29|14:28:56.542] Loaded most recent local header          number=0 hash=ac4d9a..2ca582 td=4 age=53y8mo5d
INFO [11-29|14:28:56.542] Loaded most recent local full block      number=0 hash=ac4d9a..2ca582 td=4 age=53y8mo5d
INFO [11-29|14:28:56.542] Loaded most recent local fast block      number=0 hash=ac4d9a..2ca582 td=4 age=53y8mo5d
WARN [11-29|14:28:56.543] Failed to load snapshot, regenerating    err="missing or corrupted snapshot"
INFO [11-29|14:28:56.543] Rebuilding state snapshot
INFO [11-29|14:28:56.543] Resuming state snapshot generation       root=56e81f..63b421 accounts=0 slots=0 storage=0.00B dangling=0 elapsed=0s
INFO [11-29|14:28:56.543] Generated state snapshot                 accounts=0 slots=0 storage=0.00B dangling=0 elapsed=0s
INFO [11-29|14:28:56.553] Regenerated local transaction journal    transactions=0 accounts=0
INFO [11-29|14:28:56.554] Gasprice oracle is ignoring threshold set threshold=2
WARN [11-29|14:28:56.554] Error reading unclean shutdown markers   error="leveldb: not found"
WARN [11-29|14:28:56.555] Engine API enabled                       protocol=eth
WARN [11-29|14:28:56.555] Engine API started but chain not configured for merge yet
INFO [11-29|14:28:56.555] Starting peer-to-peer node               instance=Geth/v1.10.26-stable-e5eb32ac/windows-amd64/go1.18.5
INFO [11-29|14:28:56.583] New local node record                    seq=1,669,712,336,580 id=1e2c91158b847ea6 ip=127.0.0.1 udp=0 tcp=30303
INFO [11-29|14:28:56.583] Started P2P networking                   self="enode://1e01fa06b53b8f67a380997a11e4f01b6940acdb55569fbcdaad4234004a1e5d80f7550f72178826bb494dd808e8763357a7ecae9d3ba0e
52c7c9da42886629e@127.0.0.1:30303?discport=0"
INFO [11-29|14:28:56.584] IPC endpoint opened                      url=\\.\pipe\geth.ipc
INFO [11-29|14:28:56.593] Generated JWT secret                     path=C:\Users\Student\Desktop\my-blockchain\node1\geth\jwtsecret
INFO [11-29|14:28:56.594] HTTP server started                      endpoint=127.0.0.1:8545 auth=false prefix= cors= vhosts=localhost
```

- **Start the second blockchain node**



```
→ geth attach node2/geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.16-stable-20356e57/linux-amd64/go1.17.5
at block: 0 (Wed Dec 31 1969 21:00:00 GMT-0300 (-03))
 datadir: /home/jefferson/projects/my-blockchain/node2
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> admin.nodeInfo
{
  enode: "enode://431fa50a676b35dd750a68656cf8a822edb2c083ddf2359b6c246216dfbef0f1dad517b53b9ba0a35f4d2f1d44274ae4fc4b2bdf69e774af892b097d082eff1c@127.0.0.1:3
  enr: "enr:-KO4QI5oOKnFOAPncYG0r8anzp80tam6pCxKoEhbMlt25ntof3TM_Ee_yRRJOovR4GQjJUPjs2s7A2EHOFXt0Pe2P6aGAX-ZfPg0g2V0aMfGhL3NduuAgmlkgnY0gmlwhH8AAAGJc2VjcDI1Nm
3yNZtsJGI0377w0YRzbmFwwIN0Y3CCdmCDdmRwgnZg",
  id: "4cf015cc55ac80d62c11485e8de21cf51cada4b1b1a056beddabed988e41395a",
  ip: "127.0.0.1",
  listenAddr: "[::]:30304",
  name: "Geth/v1.10.16-stable-20356e57/linux-amd64/go1.17.5",
  ports: {
    discovery: 30304,
    listener: 30304
  },
  protocols: {
    eth: {
      config: {
        byzantiumBlock: 0,
        chainId: 1234,
        constantinopleBlock: 0,
        eip150Block: 0,
        eip150Hash: "0x0000000000000000000000000000000000000000000000000000000000000000",
        eip155Block: 0,
        eip158Block: 0,
        ethash: {},
        homesteadBlock: 0,
        petersburgBlock: 0
      },
      difficulty: 4,
      genesis: "0xac4d9a5f75e2d6ce6831373d57279fe25c60cbfe82da5b3adfe620d7fe2ca582",
      head: "0xac4d9a5f75e2d6ce6831373d57279fe25c60cbfe82da5b3adfe620d7fe2ca582",
      network: 1234
    },
    snap: {}
  }
}
> 
```

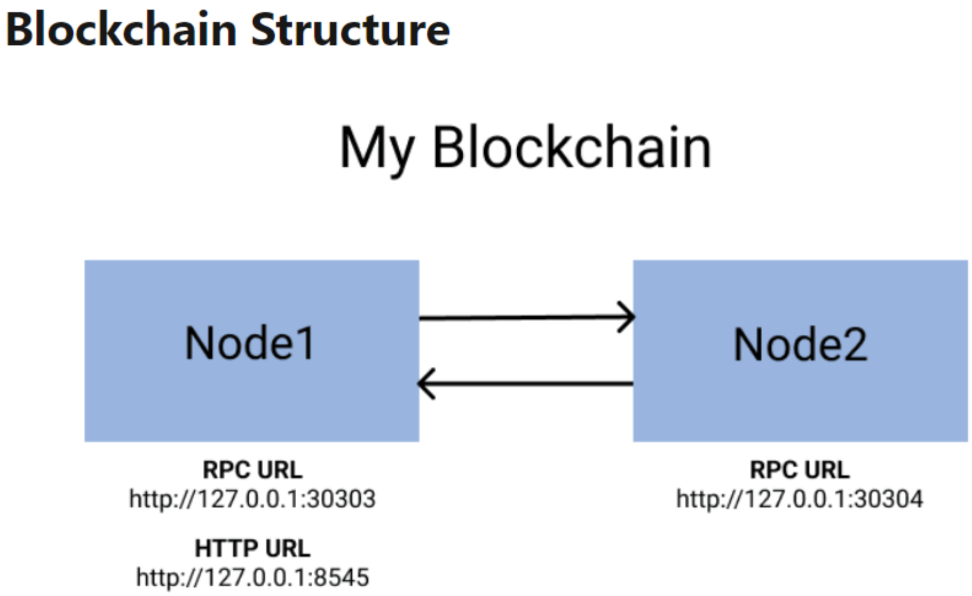
- **Create the peer-to-peer connection**



```
> admin.addPeer("enode://431fa50a676b35dd750a68656cf8a822edb2c083ddf2359b6c246216dfbef0f1dad517b53b9ba0a35f4d2f1d44274ae4fc4b2bdf69e774af892b097d082eff1c@127.0.0
true
> admin.peers
[{
    caps: ["eth/66", "snap/1"],
    enode: "enode://431fa50a676b35dd750a68656cf8a822edb2c083ddf2359b6c246216dfbef0f1dad517b53b9ba0a35f4d2f1d44274ae4fc4b2bdf69e774af892b097d082eff1c@127.0.0.1:30
    id: "4cf015cc55ac00d62c11485e0de21cf51cada4b1b1a056beddabed988e41395a",
    name: "Geth/v1.10.16-stable-20356e57/linux-amd64/go1.17.5",
    network: {
      inbound: false,
      localAddress: "127.0.0.1:58234",
      remoteAddress: "127.0.0.1:30304",
      static: true,
      trusted: false
    },
    protocols: {
      eth: {
        difficulty: 4,
        head: "0xac4d9a5f75e2d6ce6831373d57279fe25c60cbfe82da5b3adfe620d7fe2ca582",
        version: 66
      },
      snap: {
        version: 1
      }
    }
}]
>
```

- **Finally start mining and create blocks**

2. Explain your program logic, classes and methods used.



We used Go etherum to create a private blockchain network. Here we used various methods for mining for example.

personal.newAccount(): This function is used to create a new account of a user in the network.

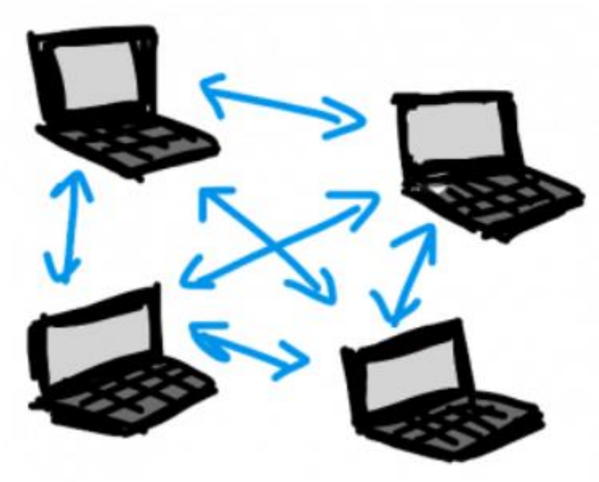
eth.getBalance("<address>"): This function is used to get the account balance of a user.

eth.blockNumber: Blocknumber method is used to get the blocknumber

miner.start(): This method is used to start mining.

miner.stop(): This method is used to stop minng which is in process.

3.Explain the Importance of the approach followed by you



With Geth, you can establish your own private Ethereum network. The blockchain of an Ethereum network begins with a root block known as the genesis block. There are no transactions in the genesis block.

Geth is often already included in the majority of web/extension wallets accessible for consumer adoption (MetaMask, Coinbase, Rainbow, etc).

But, installing and running Geth is well worth the effort if you want to experiment with mining or develop your own custom Ethereum software. It’s also useful if you simply want to control your own node and your wallet's key (signing your own transactions rather than relying on third-party software). Using Geth, you can now open up a whole world of Ethereum development.

Conclusion:- Hence we successfully develop a private blockchain network using Go Etheruem.