

## PRACTICAL: 2

### AIM:

Install and configure the following development setup tools to implement Blockchain development.

#### (Set up Blockchain Development Environment)

- Metamask (Wallet)
- Ganache Local Private Blockchain Network
- Go-Ethereum (Geth) Client
- Truffle framework
- Hardhat framework

Study and configure all testnets available in Metamask and also setup custom network using Ganache.

### THEORY:

MetaMask is a browser extension that acts as a cryptocurrency wallet. It allows users to manage their Ethereum accounts and interact with decentralized applications (dApps) directly from their browsers.

Ganache is a personal blockchain for Ethereum development. It provides a local Ethereum network that can be used for development and testing of smart contracts. Ganache simulates a blockchain environment locally, allowing you to deploy contracts, run tests, and perform debugging.

Go-Ethereum (Geth) is an Ethereum client that allows you to connect to the Ethereum network, interact with nodes, and deploy contracts. Geth can be used to create your own Ethereum node, mine, and interact with testnets or the mainnet.

Truffle is a popular development framework for Ethereum that helps developers to write, test, and deploy smart contracts. It integrates with Ganache and other Ethereum networks to facilitate blockchain development.

Hardhat is another powerful Ethereum development environment that is often used for writing, testing, and deploying smart contracts. It has some additional features such as a built-in local Ethereum network, Solidity debugging, and interaction with Ethers.js.

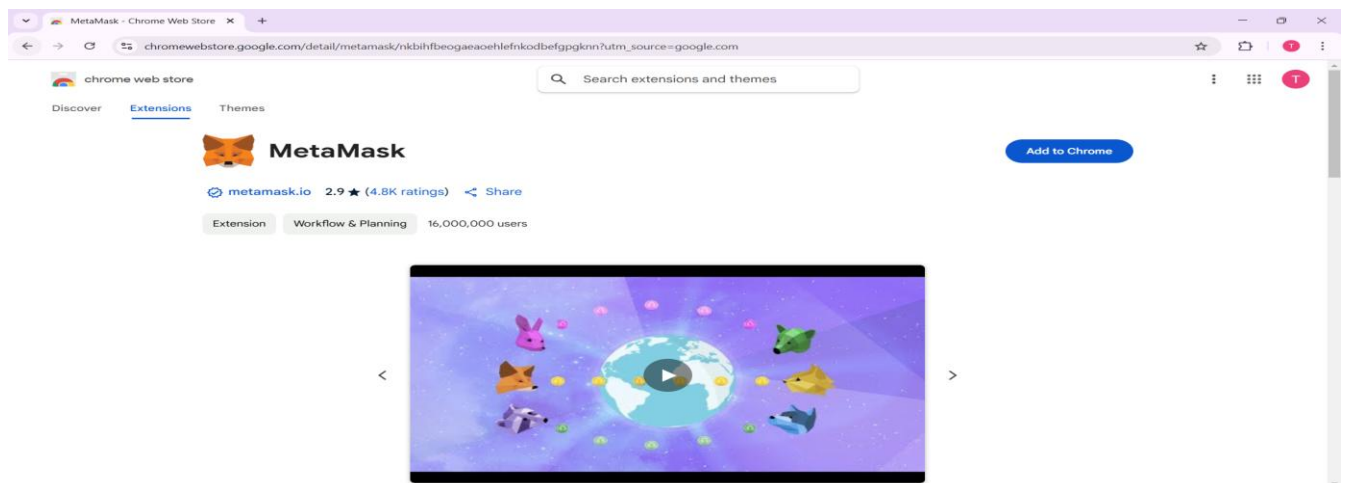
MetaMask allows connect to various Ethereum testnets for testing our decentralized applications (dApps). Some of the common testnets include:

- **Rinkeby:** A Proof of Authority (PoA) testnet that simulates the Ethereum mainnet.
- **Ropsten:** A Proof of Work (PoW) testnet that mirrors the Ethereum mainnet.
- **Goerli:** A Proof of Authority (PoA) testnet, with more stability compared to Rinkeby.
- **Kovan:** Another Proof of Authority (PoA) testnet.

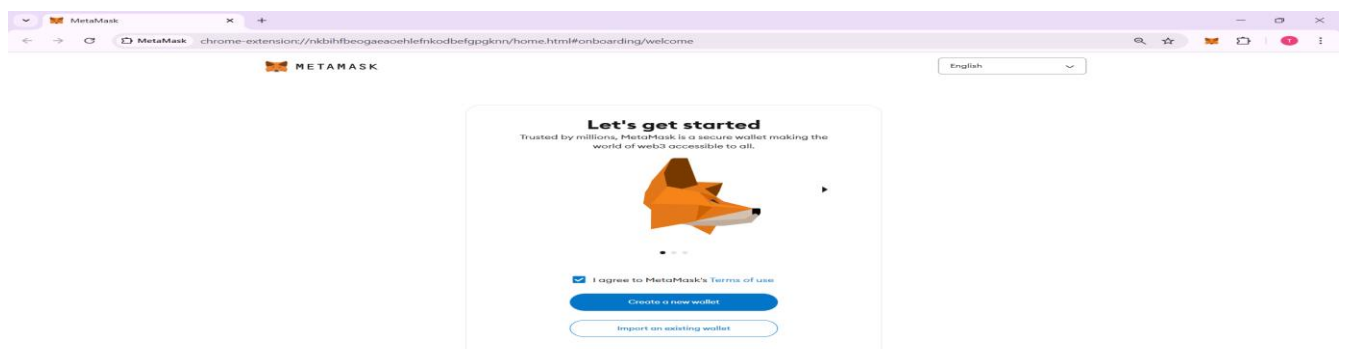
We can configure these testnets in MetaMask by selecting them from the network dropdown or adding a custom RPC network (for example, using Ganache).

**CODE:**

- `geth -- version`
- `geth --datadir /path account new`
- `geth --datadir /path --password secret.txt --dev`
- `geth init path/to/genesis.json`
- `geth --networkid 1337 --http --http.addr "127.0.0.1" --http.port 8545 --http.api "eth,net,web3,personal"`
- `npm install -g truffle`
- `truffle init`
- `mkdir hardhat-project`
- `cd hardhat-project`
- `npm init -y`
- `npm install --save-dev hardhat`
- `npx hardhat`
- `truffle migrate --network development`
- `npm install --save-dev hardhat`

**OUTPUT:**

*Figure 1: Open chrome web store and search MetaMask and then click add to chrome*



*Figure 2: After installing MetaMask create a new wallet*

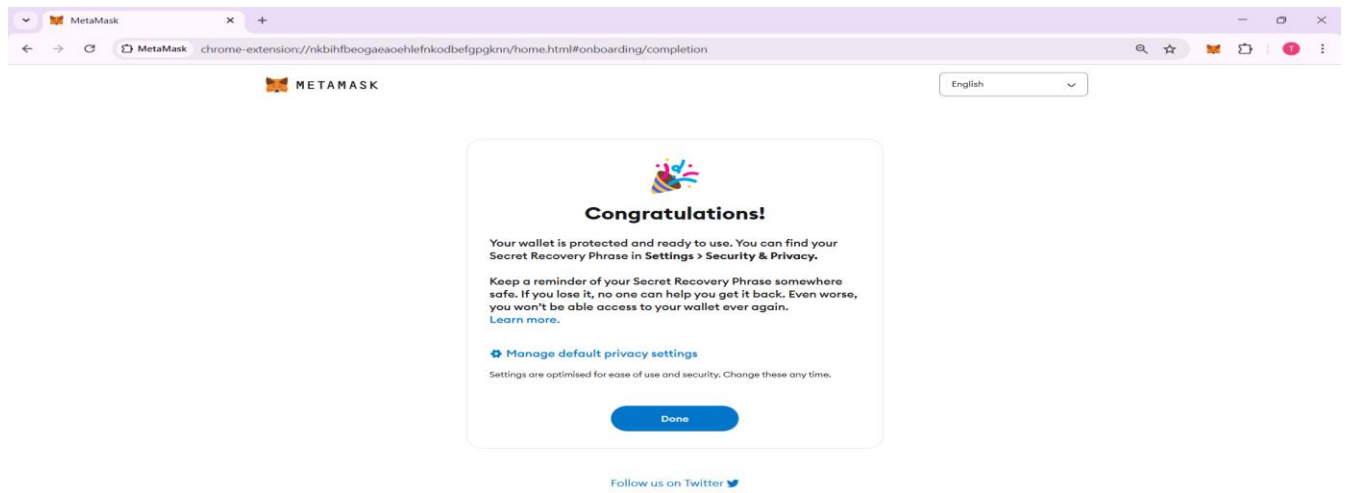


Figure 3: MetaMask wallet create successfully

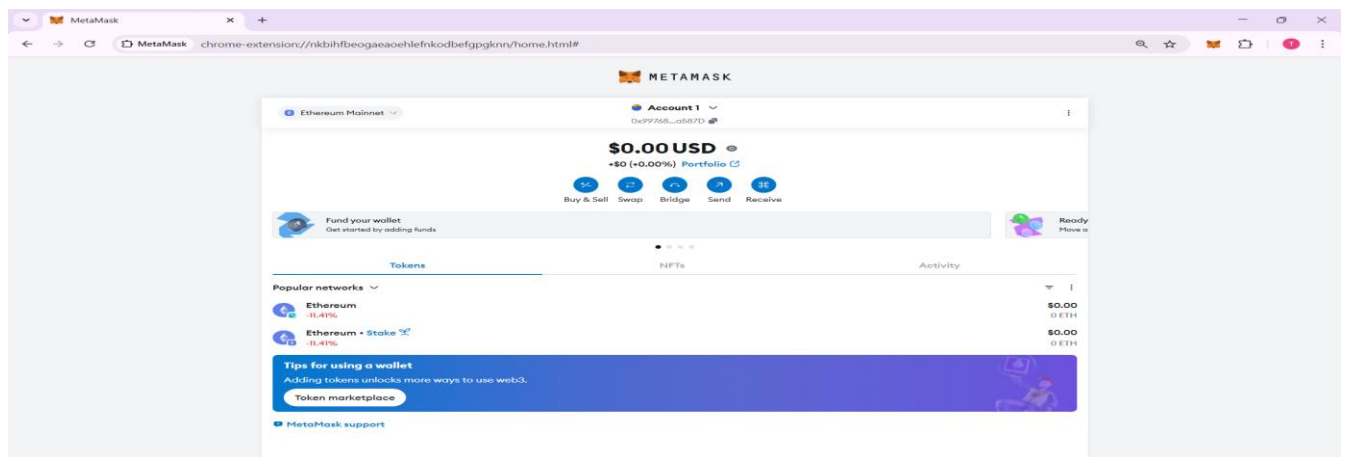


Figure 4: In Account-1 wallet has 0.00\$ USD

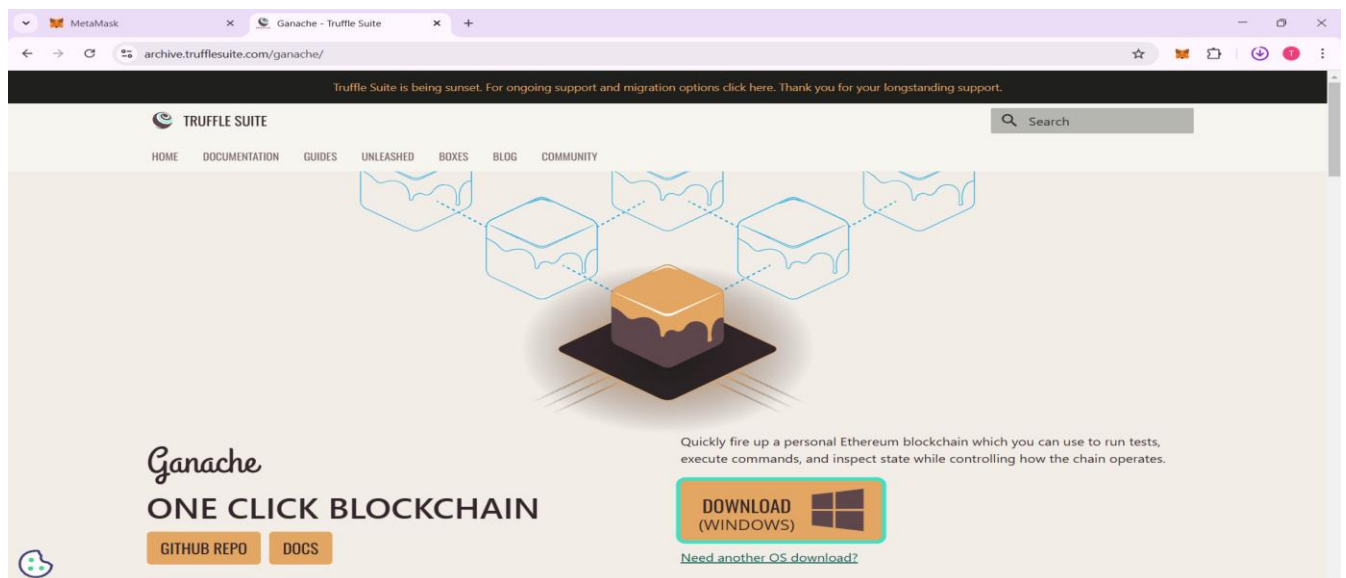


Figure 5: Download Ganache form above website

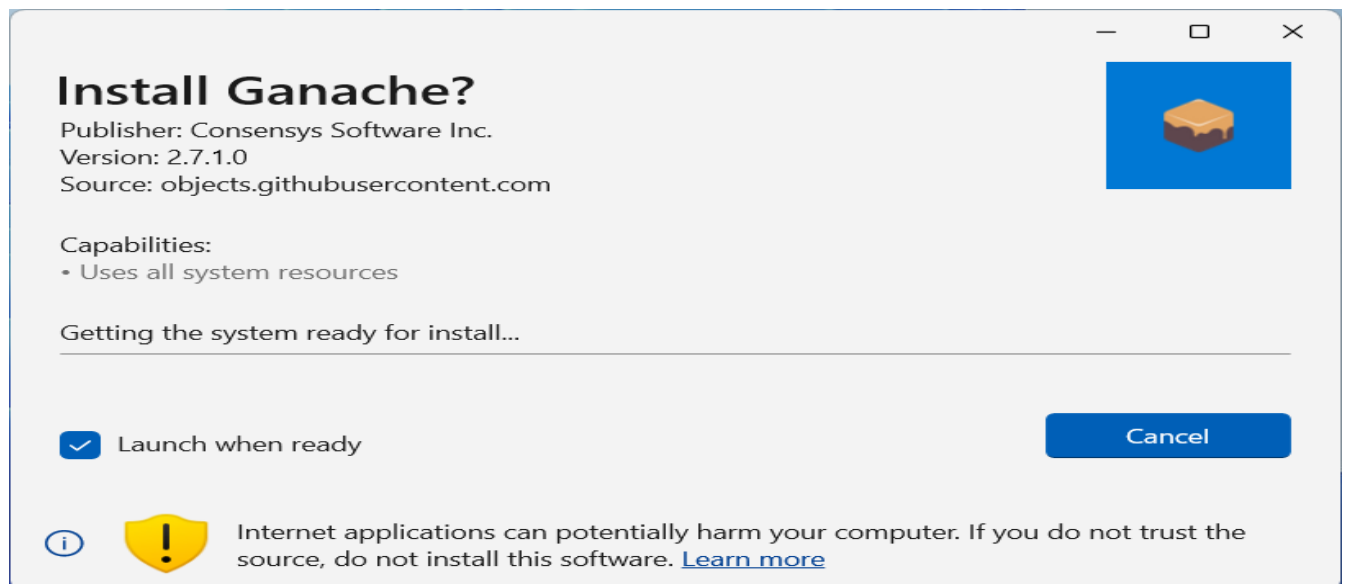


Figure 6: Install Ganache and then launch it

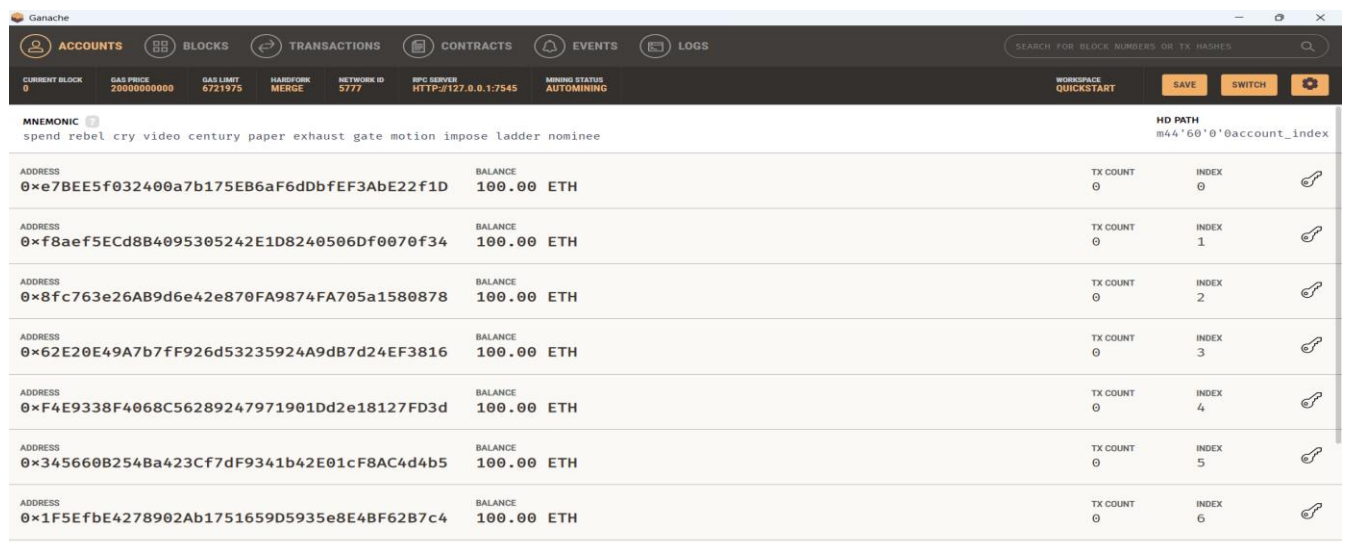


Figure 7: GUI of Ganache showing address and balance

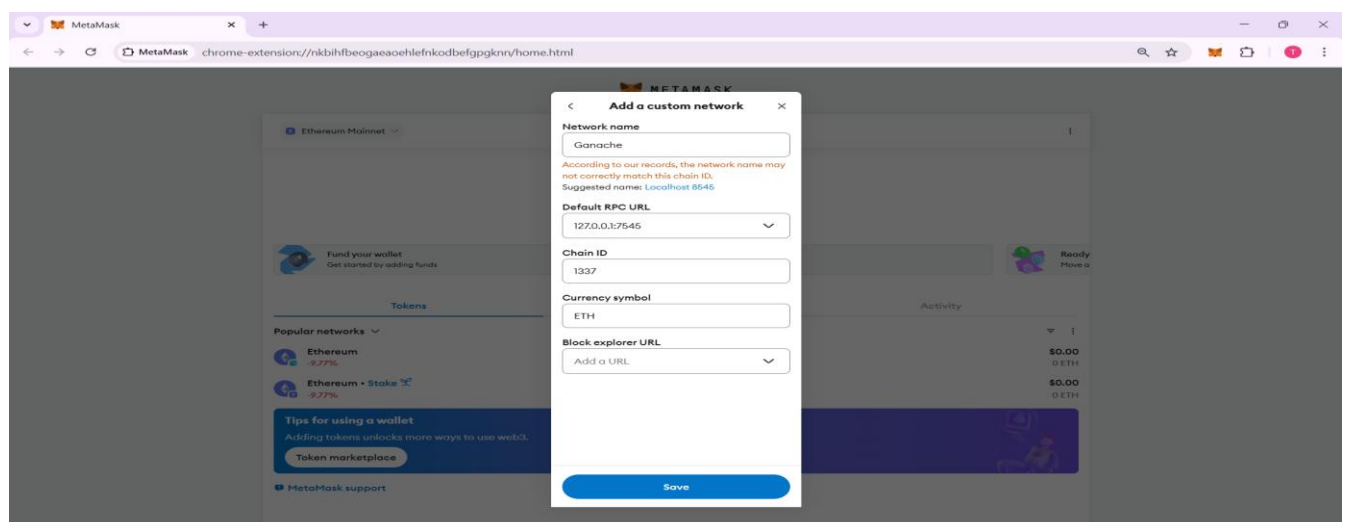


Figure 8: Add Custom network to MetaMask

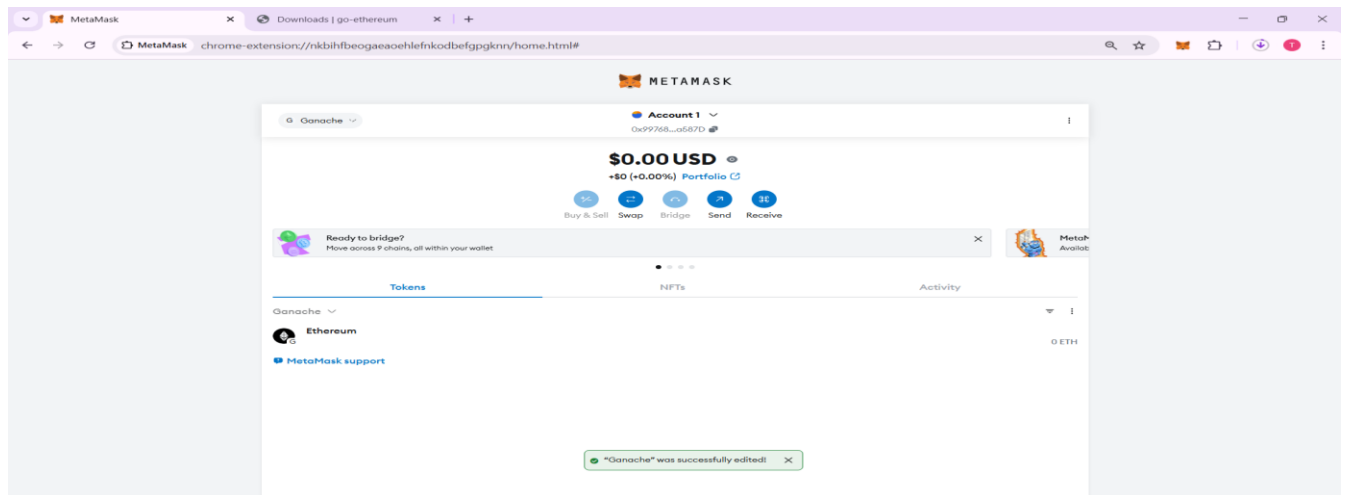


Figure 9: Adding Ganache network to my wallet

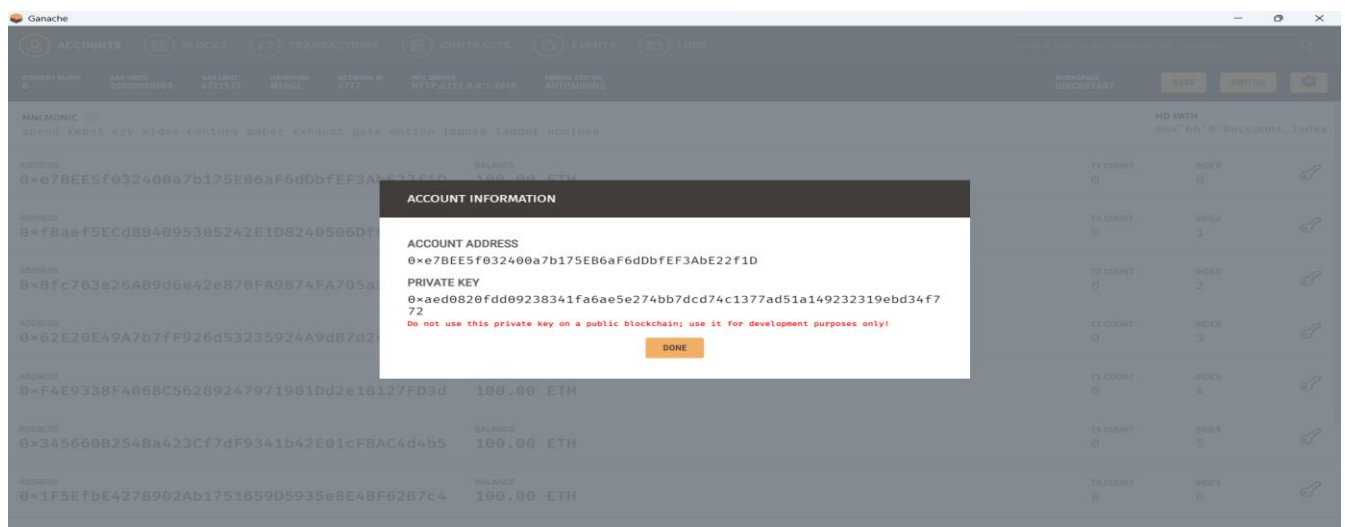


Figure 10: Copy the private key of one address to import account in MetaMask

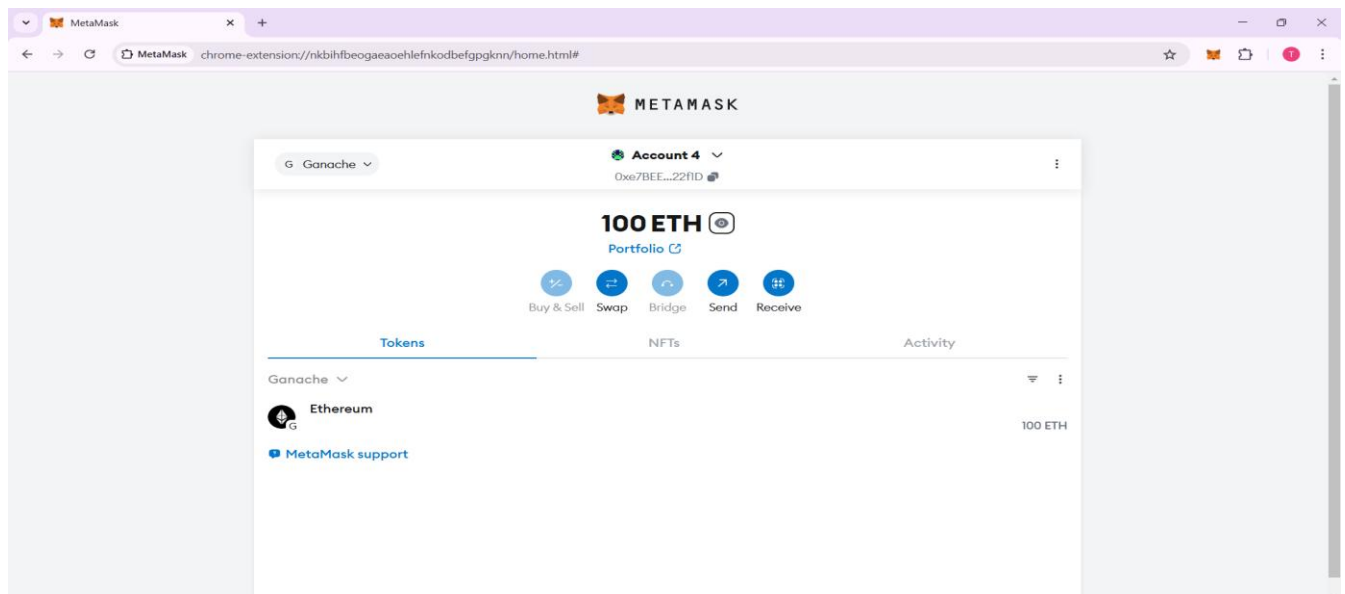


Figure 11: Account import in MetaMask

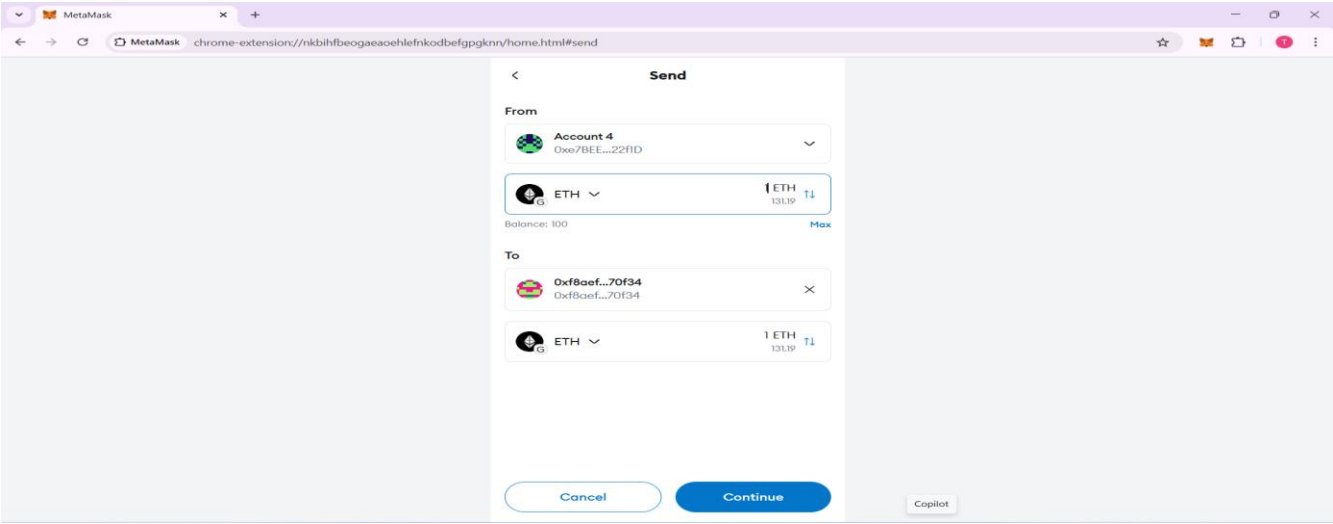


Figure 12: Sending 1 ETH to another account

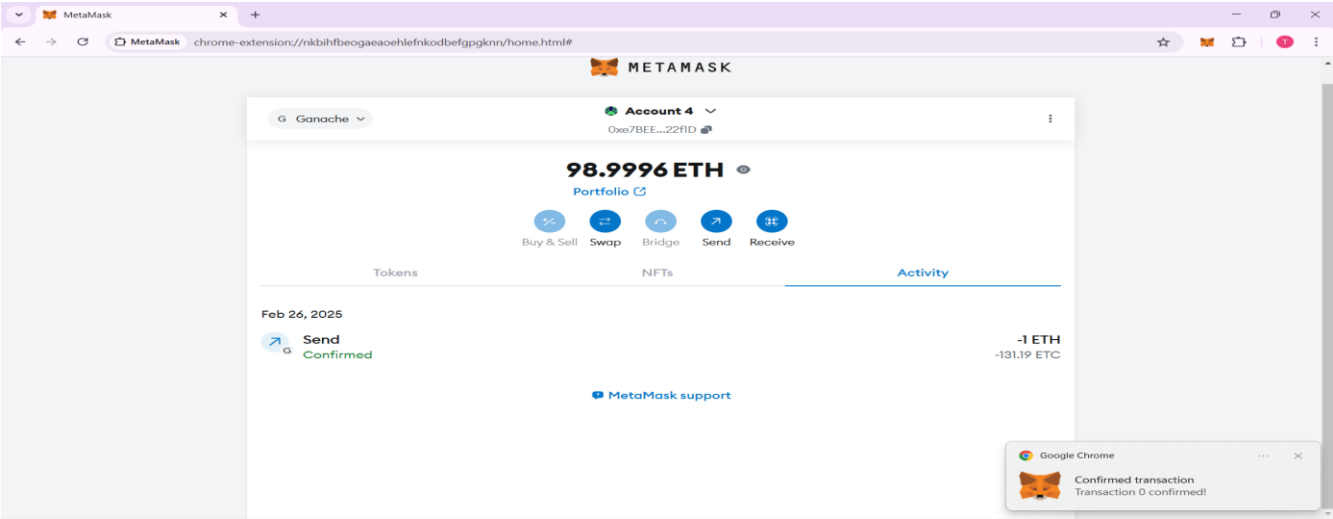


Figure 13: Sending 1 ETH successfully

Ganache				
ACCOUNTS   BLOCKS   TRANSACTIONS   CONTRACTS   EVENTS   LOGS				
CURRENT BLOCK 1   GAS PRICE 20000000000   GAS LIMIT 6721975   HARDFORK MERGE   NETWORK ID 5777   RPC SERVER HTTP://127.0.0.1:7545   MINING STATUS AUTOMINING   WORKSPACE QUICKSTART   SAVE   SWITCH   ⚙				
MNEMONIC spend rebel cry video century paper exhaust gate motion impose ladder nominee   HD PATH m/44'/60'/0'/0/account_index				
ADDRESS 0xe7BEE5f032400a7b175EB6aF6dDbfEF3AbE22f1D	BALANCE 99.00 ETH	TX COUNT 1	INDEX 0	🔗
ADDRESS 0xf8aef5ECd8B4095305242E1D8240506Df0070f34	BALANCE 101.00 ETH	TX COUNT 0	INDEX 1	🔗
ADDRESS 0x8fc763e26AB9d6e42e870FA9874FA705a1580878	BALANCE 100.00 ETH	TX COUNT 0	INDEX 2	🔗
ADDRESS 0x62E20E49A7b7fF926d53235924A9dB7d24EF3816	BALANCE 100.00 ETH	TX COUNT 0	INDEX 3	🔗
ADDRESS 0xF4E9338F4068C56289247971901Dd2e18127FD3d	BALANCE 100.00 ETH	TX COUNT 0	INDEX 4	🔗
ADDRESS 0x345660B254Ba423Cf7dF9341b42E01cF8AC4d4b5	BALANCE 100.00 ETH	TX COUNT 0	INDEX 5	🔗
ADDRESS 0x1F5EfEbE4278902Ab1751659D5935e8E4BF62B7c4	BALANCE 100.00 ETH	TX COUNT 0	INDEX 6	🔗

Figure 14: Successfully receiving 1ETH proof

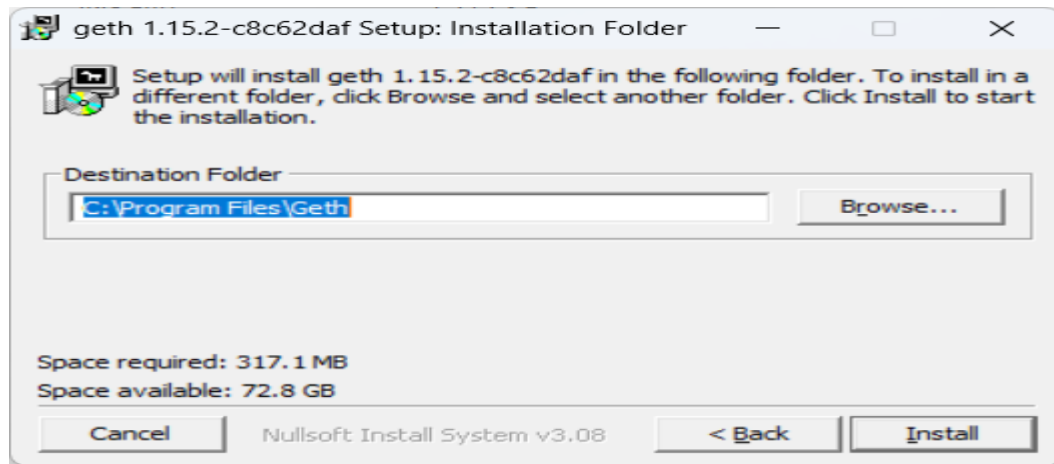


Figure 15: Install Go-Ethereum (Geth) Client

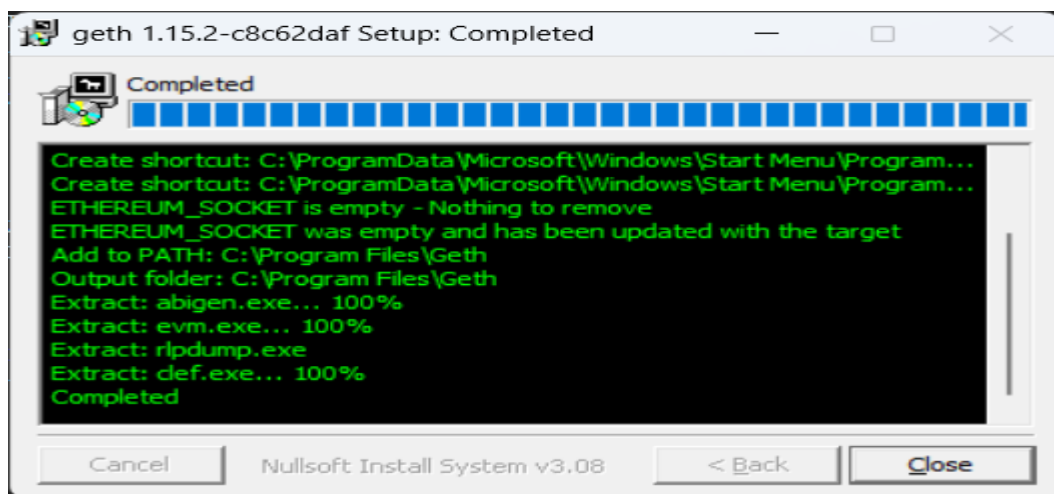


Figure 16: Go-Ethereum (Geth) Client install successfully

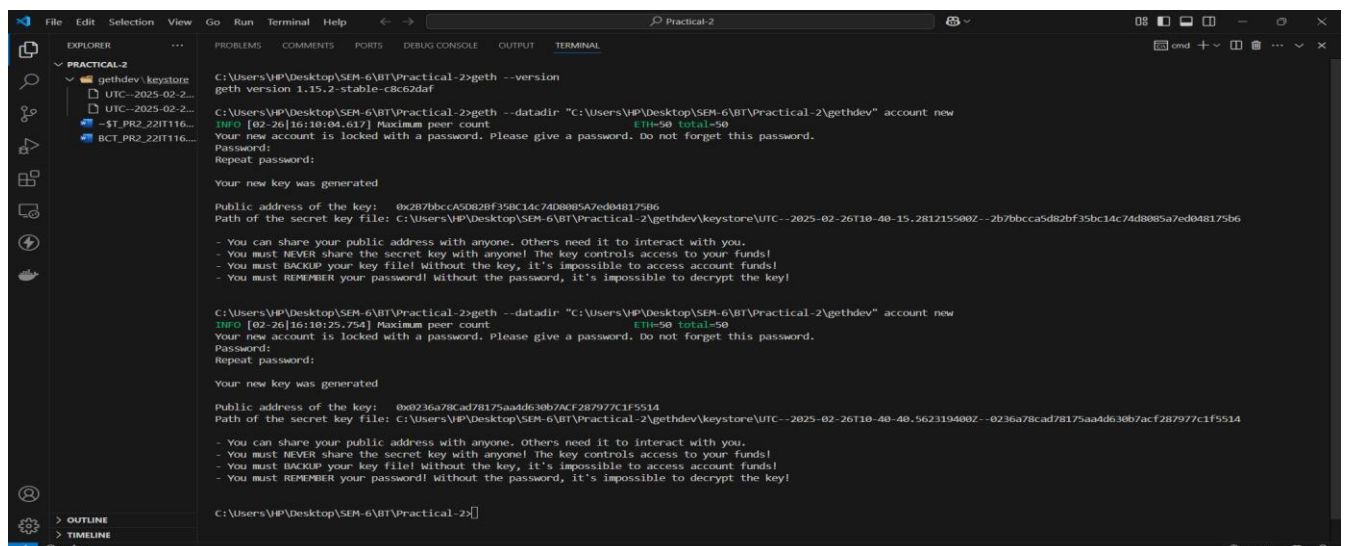


Figure 17: Verify Go-Ethereum(Geth) Client install and create two new accounts



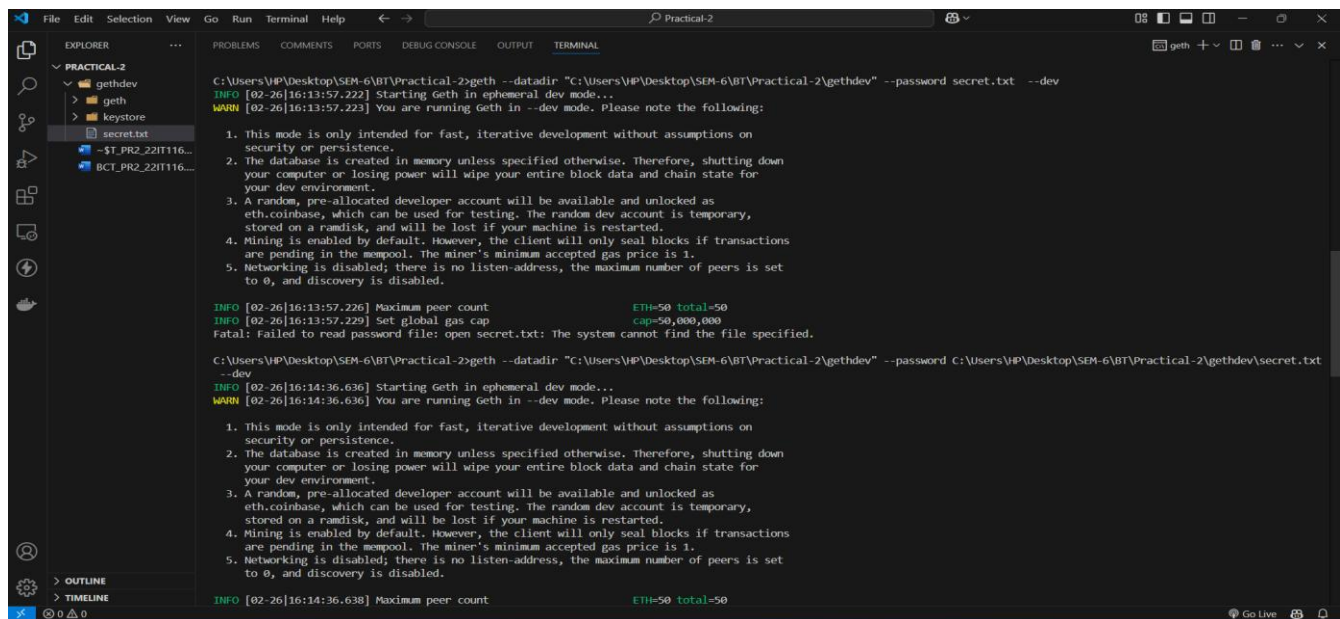


Figure 18: Open account in developer mode

```
C:\Users\HP\Desktop\SEM-6\BT\Practical-2\gethdev>geth attach \\.pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.15.2-stable-c8c62daf/windows-amd64/go1.23.6
at block: 0 (Thu Jan 01 1970 05:30:00 GMT+0530 (IST))
datadir: C:\Users\HP\Desktop\SEM-6\BT\Practical-2\gethdev
modules: admin:1.0 debug:1.0 dev:1.0 eth:1.0 miner:1.0 net:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
>
```

Figure 19: Open account in JavaScript console

```
C:\WINDOWS\system32\cmd.exe
> eth
{
  accounts: ["0x2b7bbcca5d82bf35bc14c74d0805a7ed040175b6", "0x0236a78cad78175aa4d630b7ac6f287977c1f8514"],
  blobBaseFee: 1,
  blockNumber: 1,
  coinbase: undefined,
  compile: {
    lll: function(),
    serpent: function(),
    solidity: function()
  },
  defaultAccount: undefined,
  defaultBlock: "latest",
  gasPrice: 1000000001,
  hashrate: undefined,
  maxPriorityFeePerGas: 1,
  mining: undefined,
  pendingTransactions: [],
  protocolVersion: undefined,
  syncing: {
    currentBlock: 0,
    healedBytecodeBytes: 0,
    healedBytecodes: 0,
    healedTrieNodeBytes: 0,
    healedTrieNodes: 0,
    healingBytecode: 0,
    healingTrieNodes: 0,
    highestBlock: 0,
    startingBlock: 0,
    syncedAccountBytes: 0,
    syncedAccounts: 0,
    syncedBytecodeBytes: 0,
    syncedBytecodes: 0,
    syncedStorage: 0,
    syncedStorageBytes: 0,
    txIndexFinishedBlocks: 0,
    txIndexRemainingBlocks: 1
  },
  call: function(),
  chainId: function(),

```

Figure 20: Functions provided by eth library

```
> eth.getBalance(eth.accounts[0])
1.15792089237316195423570985088687907853269984665640564039457584087913129639927e+77
> eth.getBalance(eth.accounts[1])
0
> eth.sendTransaction({from:eth.accounts[0],to:eth.accounts[1],value: web3.toWei(1.5,"Ether")})
"0xe6fd83deec67ff541c80c57395be0d2e2d796d7ee7789965b20637fd06cfdc4d"
> eth.getBalance(eth.accounts[1])
1500000000000000000
>
```

Figure 21: Transfer money from one account to another account



The screenshot shows a Windows IDE with the following components:

- File Explorer:** Located on the left, it shows a directory structure with 'PRACT...' and 'sepolia' folders. The 'sepolia' folder is selected.
- Terminal:** The main area on the right displays the command and output of a Geth node starting a Sepolia testnet. The command is: `c:\Users\VP\Desktop\SEM-6\BT\Practical-2>geth --datadir "c:\Users\VP\Desktop\SEM-6\BT\Practical-2\sepolia" --sepolia --http --http.api eth,net,web3,admin`. The output shows the node starting, initializing the database, and beginning the Sepolia chain.
- Output:** The terminal output includes the following information:
  - Starting Geth on Sepolia testnet...
  - Maximum peer count: ETH=50, total=50
  - Set global gas cap: cap=50,000,000
  - Initializing the KZG library: backend=gokzg
  - Allocated trie memory caches: clean=154.00MiB, dirty=256.00MiB
  - Defaulting to pebble as the backing database
  - Allocated cache and file handles: database=c:\Users\VP\Desktop\SEM-6\BT\Practical-2\sepolia\geth\chaindata, cache=512.00MiB, handles=8192
  - Opened ancient database: database=c:\Users\VP\Desktop\SEM-6\BT\Practical-2\sepolia\geth\chaindata\ancient\chain, readonly=false
  - State schema set to default: scheme=path
  - Head block is not reachable: network=11,155,111, dbversion=cnli>
  - Initializing Ethereum protocol: database=c:\Users\VP\Desktop\SEM-6\BT\Practical-2\sepolia\geth\chaindata\ancient\state, readonly=false
  - Opened ancient database: cache=154.00MiB, buffer=256.00MiB, history=90000
  - Initializing path database
  - Writing custom genesis block
  - Chain ID: 11155111 (sepolia)
  - Consensus: Beacon (proof-of-stake), merged from Ethash (proof-of-work)
  - Pre-Merge hard forks (block based):
    - Homestead: #0 (<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/homestead.md>)
    - Tangerine Whistle (EIP 150): #0 (<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/tangerine-whistle.md>)
    - Spurious Dragon/1 (EIP 155): #0 (<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md>)
    - Spurious Dragon/2 (EIP 158): #0 (<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md>)
    - Byzantium: #0 (<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/byzantium.md>)
    - Constantinople: #0 (<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/constantinople.md>)
    - Petersburg: #0 (<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/petersburg.md>)
    - Istanbul: #0 (<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/istanbul.md>)
    - Muir Glacier: #0 (<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/muir-glacier.md>)
    - Berlin: #0 (<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/berlin.md>)
    - London: #0 (<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/london.md>)
  - Merge configured: https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md
  - Hard fork specification: https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md
  - Network known to be merged
  - Total terminal difficulty: 17000000000000000

*Figure 22: Connect with sepolia network*

```
C:\Users\HP\Desktop\SEM-6\BT\Practical-2\sepolia\consensus>curl https://raw.githubusercontent.com/prysmaticlabs/prysm/master/prysm.bat --output prysm.bat
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100 8744 100 8744    0     0 15130      0 --:--:-- --:--:-- --:--:-- 15180

C:\Users\HP\Desktop\SEM-6\BT\Practical-2\sepolia\consensus>reg add HKCU\Console /v VirtualTerminalLevel /t REG_DWORD /d 1
The operation completed successfully.

C:\Users\HP\Desktop\SEM-6\BT\Practical-2\sepolia\consensus>
```

Figure 23: Download consensus prysm and register

[illegible]

Figure 24: Create a genesis.json file

```
C:\Users\HP\Desktop\SEM-6\BT\Practical-2>geth --datadir C:\Users\HP\Desktop\SEM-6\BT\Practical-2\sepolia\data init C:\Users\HP\Desktop\SEM-6\BT\Practical-2\sepolia\genesis.json
INFO [02-26 19:28:39.739] Maximum peer count          ETH=50 total=50
INFO [02-26 19:28:39.746] Set global gas cap          cap=50,000,000
INFO [02-26 19:28:39.746] Initializing the KZG library backend=gokzg
INFO [02-26 19:28:39.794] Using pebble as the backing database
INFO [02-26 19:28:39.796] Allocated cache and file handles database=C:\Users\HP\Desktop\SEM-6\BT\Practical-2\sepolia\data\geth\chaindata cache=16.00MiB handles=16
INFO [02-26 19:28:39.848] Opened ancient database      database=C:\Users\HP\Desktop\SEM-6\BT\Practical-2\sepolia\data\geth\chaindata\ancient\chain readonly=false
INFO [02-26 19:28:39.853] State scheme set to already existing scheme=path
INFO [02-26 19:28:39.874] Opened ancient database      database=C:\Users\HP\Desktop\SEM-6\BT\Practical-2\sepolia\data\geth\chaindata\ancient\state readonly=false
INFO [02-26 19:28:39.874] Initialized path database     cache=16.00MiB buffer=64.00MiB history=90000
INFO [02-26 19:28:39.875] Successfully wrote genesis state database=chaindata hash=86d010...11fbc
```

Figure 25: Initialize Geth with the genesis.json

```

C:\Users\VP\Desktop\SEM-6\BT\Practical-2>geth --datadir C:\Users\VP\Desktop\SEM-6\BT\Practical-2\sepolia\data --networkid 1338 --http --http.addr "127.0.0.1" --http.port 8545 --h
ttp.api "eth,net,web3,personal"
INFO [02-26|19:28:49.289] Maximum peer count                      ETH=50 total=50
INFO [02-26|19:28:49.296] Set global gas cap                      cap=50,000,000
INFO [02-26|19:28:49.296] Initializing the kzg library            backend=polkz
INFO [02-26|19:28:49.355] Allocated trie memory caches            clean=154.00MiB dirty=256.00MiB
INFO [02-26|19:28:49.355] Using pebble as the backing database
INFO [02-26|19:28:49.355] Allocated cache and file handles        database=C:\Users\VP\Desktop\SEM-6\BT\Practical-2\sepolia\data\geth\chaindata cache=512.00MiB handles=8192
INFO [02-26|19:28:49.403] Opened ancient database                  database=C:\Users\VP\Desktop\SEM-6\BT\Practical-2\sepolia\data\geth\chaindata\ancient\chain readonly=false
INFO [02-26|19:28:49.410] State scheme set to already existing    scheme=path
INFO [02-26|19:28:49.411] Initializing Ethereum protocol          network=1338 dversion=0
INFO [02-26|19:28:49.428] Opened ancient database                  database=C:\Users\VP\Desktop\SEM-6\BT\Practical-2\sepolia\data\geth\chaindata\ancient\state readonly=false
INFO [02-26|19:28:49.428] Initialized path database                cache=154.00MiB buffer=256.00MiB history=90000
INFO [02-26|19:28:49.430] -----
INFO [02-26|19:28:49.430] Chain ID: 1338 (unknown)
INFO [02-26|19:28:49.430] Consensus: unknown
INFO [02-26|19:28:49.431] -----
INFO [02-26|19:28:49.431] Pre-Merge hard forks (block based):
INFO [02-26|19:28:49.431]   - Homestead: #0                      (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/homestead.md)
INFO [02-26|19:28:49.431]   - Tangerine Whistle (EIP 150): #0    (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/tangerine-whistle.m
d)
INFO [02-26|19:28:49.431]   - Spurious Dragon/1 (EIP 155): #0    (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md)
INFO [02-26|19:28:49.431]   - Spurious Dragon/2 (EIP 158): #0    (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md)
INFO [02-26|19:28:49.431]   - Byzantium: #0                     (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/byzantium.md)
INFO [02-26|19:28:49.431]   - Constantinople: #0                (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/constantinople.md)
INFO [02-26|19:28:49.431]   - Petersburg: #0                   (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/petersburg.md)
INFO [02-26|19:28:49.431]   - Istanbul: #0                     (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/istanbul.md)
INFO [02-26|19:28:49.431]   - Berlin: #nil                     (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/berlin.md)
INFO [02-26|19:28:49.432]   - London: #nil                     (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/london.md)
INFO [02-26|19:28:49.432] Merge configured:
INFO [02-26|19:28:49.432]   - Hard-fork specification:           https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md
INFO [02-26|19:28:49.432]   - Network known to be merged:
INFO [02-26|19:28:49.432]   - Total terminal difficulty: 0
INFO [02-26|19:28:49.432] Post-Merge hard forks (timestamp based):
INFO [02-26|19:28:49.432] -----

```

Figure 26: Start Geth with the data directory

Geth Private Network

Network name

Geth Private Networks

According to our records, the network name may not correctly match this chain ID.  
Suggested name: [Elysium Testnet](#)

Default RPC URL

127.0.0.1:8545

Chain ID

1338

Currency symbol

ETH

Suggested currency symbol: [ELY](#)  
This token symbol doesn't match the network name or chain ID entered. Many popular tokens use similar symbols, which scammers

Save

*Figure 27: Add Geth network in MetaMask*

```

C:\Users\HP\Desktop\SEM-6\BT\Practical-2>npm install -g truffle
npm warn deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm warn deprecated har-validator@5.1.5: this library is no longer supported
npm warn deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm warn deprecated mklirp-promise@5.0.1: This package is broken and no longer maintained. 'mklirp' itself supports promises now, please switch to that.
npm warn deprecated deferred-level-down@5.3.0: Superseded by abstract-level (https://github.com/Level/community#faq)
npm warn deprecated level-errors@2.0.1: Superseded by abstract-level (https://github.com/Level/community#faq)
npm warn deprecated encoding-down@6.3.0: Superseded by abstract-level (https://github.com/Level/community#faq)
npm warn deprecated level-up@4.4.0: Superseded by abstract-level (https://github.com/Level/community#faq)
npm warn deprecated level-packager@5.1.1: Superseded by abstract-level (https://github.com/Level/community#faq)
npm warn deprecated level-codec@9.0.2: Superseded by level-transcoder (https://github.com/Level/community#faq)
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async request
s by a key value, which is much more comprehensive and powerful.
npm warn deprecated memdown@1.4.1: Superseded by memory-level (https://github.com/Level/community#faq)
npm warn deprecated glob@7.2.0: Glob versions prior to v9 are no longer supported
npm warn deprecated level-down@5.6.0: Superseded by classic-level (https://github.com/Level/community#faq)
npm warn deprecated multibase@0.6.1: This module has been superseded by the multiformats module
npm warn deprecated multicodec@0.5.7: This module has been superseded by the multiformats module
npm warn deprecated cids@0.7.5: This module has been superseded by the multiformats module
npm warn deprecated apollo-server-errors@3.3.1: The 'apollo-server-errors' package is part of Apollo Server v2 and v3, which are now end-of-life (as of October 22nd 2023 and Octo
ber 22nd 2024, respectively). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/
for more details.
npm warn deprecated apollo-reporting-protobuf@3.4.0: The 'apollo-reporting-protobuf' package is part of Apollo Server v2 and v3, which are now end-of-life (as of October 22nd 202
3 and October 22nd 2024, respectively). This package's functionality is now found in the '@apollo/usage-reporting-protobuf' package. See https://www.apollographql.com/docs/apollo
-server/previous-versions/ for more details.
npm warn deprecated apollo-server-env@4.2.1: The 'apollo-server-env' package is part of Apollo Server v2 and v3, which are now end-of-life (as of October 22nd 2023 and October 22
nd 2024, respectively). This package's functionality is now found in the '@apollo/utils.fetcher' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/
for more details.
npm warn deprecated apollo-datasource@3.3.2: The 'apollo-datasource' package is part of Apollo Server v2 and v3, which are now end-of-life (as of October 22nd 2023 and October 22
nd 2024, respectively). See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm warn deprecated apollo-server-plugin-base@3.7.2: The 'apollo-server-plugin-base' package is part of Apollo Server v2 and v3, which are now end-of-life (as of October 22nd 202
3 and October 22nd 2024, respectively). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-v
ersions/ for more details.
npm warn deprecated apollo-server-express@3.13.0: The 'apollo-server-express' package is part of Apollo Server v2 and v3, which are now end-of-life (as of October 22nd 2023 and O
ctober 22nd 2024, respectively). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions
/ for more details.

```

Figure 28: Install Truffle globally

```

C:\Users\HP\Desktop\SEM-6\BT\Practical-2>mkdir truffle-project
C:\Users\HP\Desktop\SEM-6\BT\Practical-2>cd truffle-project\
C:\Users\HP\Desktop\SEM-6\BT\Practical-2\truffle-project>truffle init

Starting init...
=====

> Copying project files to c:\Users\HP\Desktop\SEM-6\BT\Practical-2\truffle-project

Init successful, sweet!

Try our scaffold commands to get started:
$ truffle create contract YourContractName # scaffold a contract
$ truffle create test YourTestName # scaffold a test

http://trufflesuite.com/docs

```

Figure 29: Initialize a new Truffle project

```

truffle-config.js X
truffle-project > truffle-config.js > ...
1 module.exports = {
2   networks: {
3     development: {
4       host: "127.0.0.1",
5       port: 7545,
6       network_id: "*",
7     },
8   },
9 };

```

Figure 30: Configure the truffle-config.js file to connect Ganache

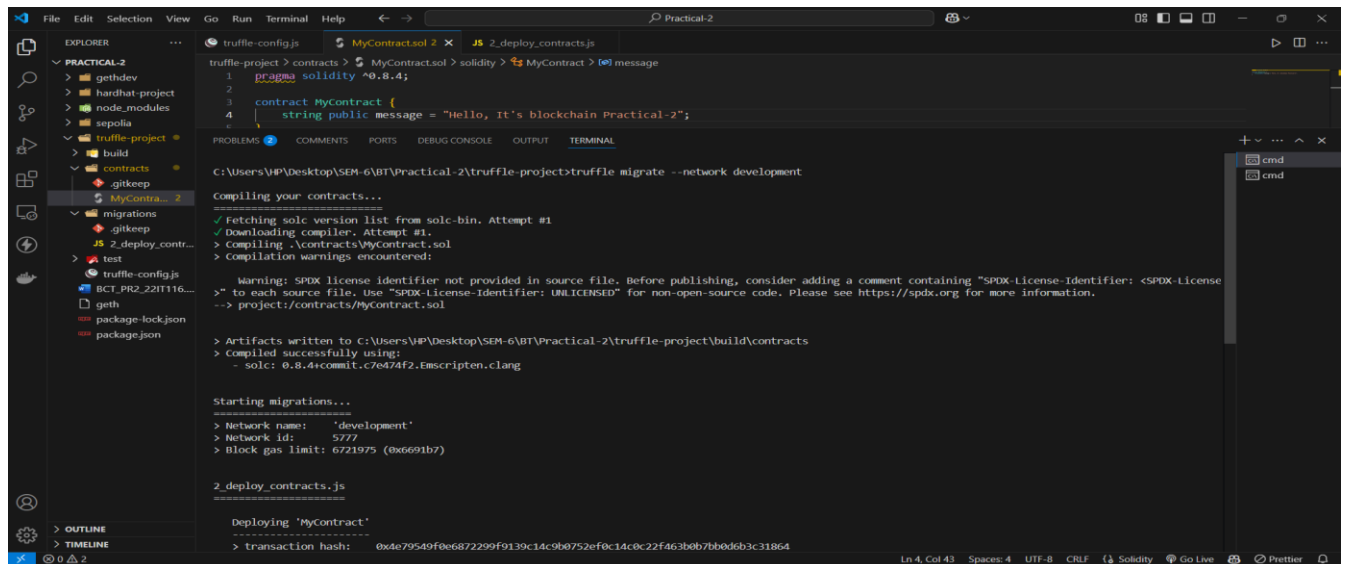


Figure 31: Deploy My contract

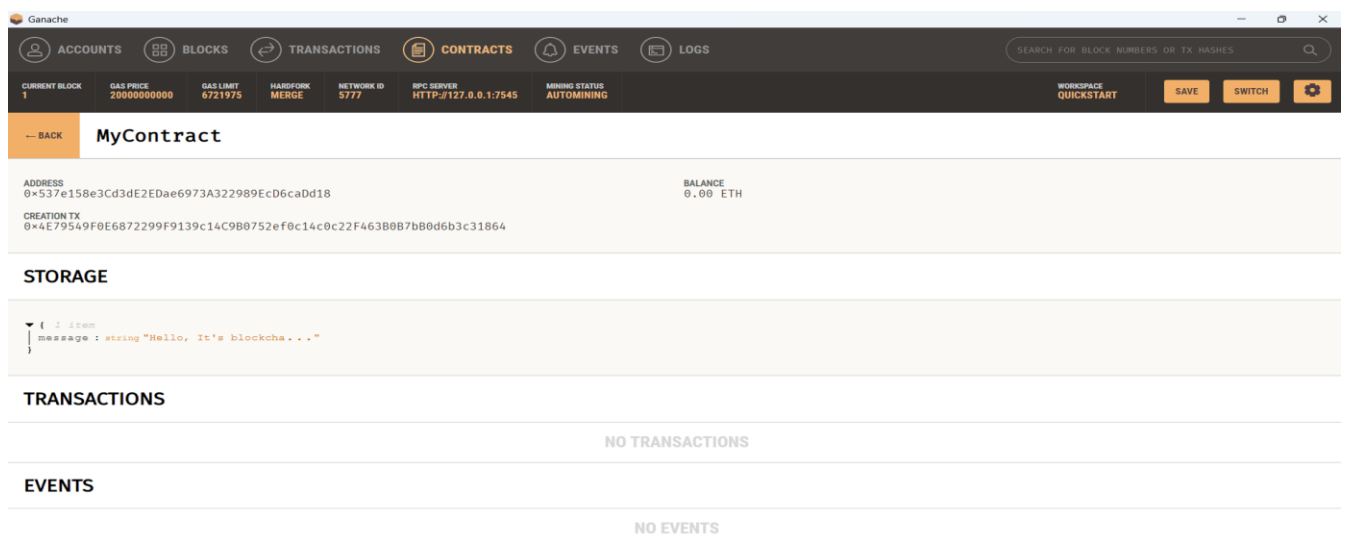


Figure 32: Verify that contract run successfully

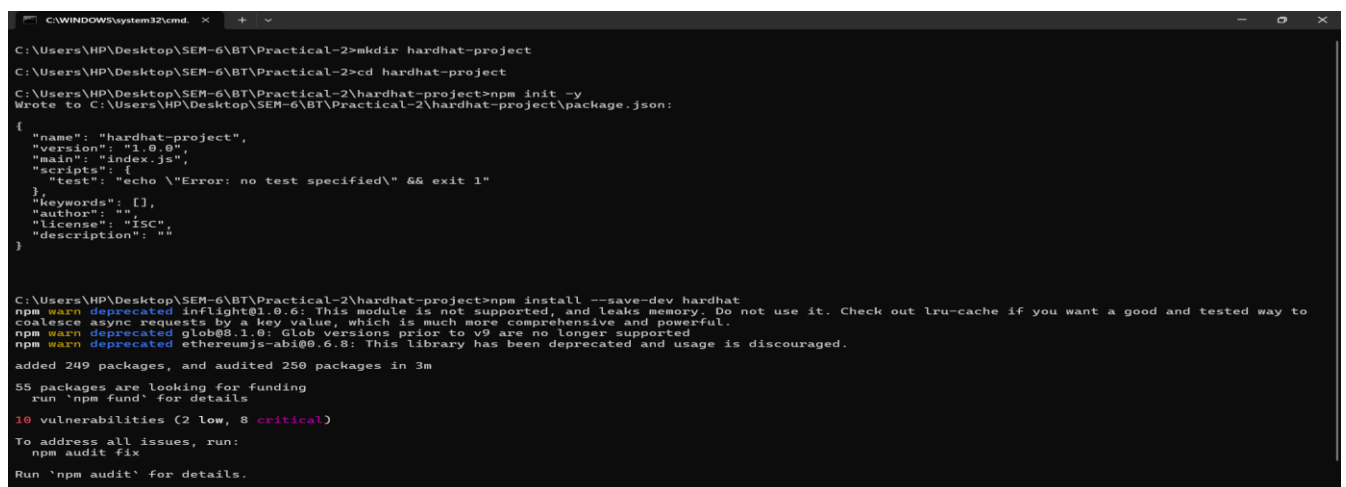


Figure 33: Install Hardhat

```

C:\WINDOWS\system32\cmd. X + v
C:\Users\HP\Desktop\SEM-6\BT\Practical-2\hardhat-project>npx hardhat
      000 000      000 000      000
      000 000      000 000      000
      000 000      000 000      000
      000000000 0000b. 0000000 .d00000 00000b. 0000b. 000000
      000 000 "00b 000P" d00" 000 000 "00b "00b 000
      000 000 .d000000 000 000 000 000 .d000000 000
      000 000 000 000 000 Y00b 000 000 000 000 Y00b.
      000 000 "Y000000 000 "Y00000 000 000 "Y000000 "Y000

Welcome to Hardhat v2.22.19

✓ What do you want to do? · Create a JavaScript project
✓ Hardhat project root: · C:\Users\HP\Desktop\SEM-6\BT\Practical-2\hardhat-project
✓ Do you want to add a .gitignore? (Y/n) · y
✓ Help us improve Hardhat with anonymous crash reports & basic usage data? (Y/n) · y
✓ Do you want to install this sample project's dependencies with npm (@nomicfoundation/hardhat-toolbox)? (Y/n) · y

npm install --save-dev "@nomicfoundation/hardhat-toolbox@5.0.0"
npm warn deprecated lodash.isequal@4.5.0: This package is deprecated. Use require('node:util').isDeepStrictEqual instead.
npm warn deprecated glob@7.1.7: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@5.0.15: Glob versions prior to v9 are no longer supported

added 327 packages, and audited 577 packages in 3m

99 packages are looking for funding
  run `npm fund` for details

29 vulnerabilities (11 low, 1 high, 17 critical)

To address issues that do not require attention, run:
  npm audit fix

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.

```

*Figure 34: Initialize a new Hardhat project*

## LATEST APPLICATIONS:

- mBridge: Cross-Border Payments with Central Bank Digital Currencies (CBDCs)
- MyLime: Digital Product Passports for Authenticity and Traceability
- China RealDID: National Decentralized Identifier System
- Valencia's Digital Currency for Revitalizing Rural Commerce

## LEARNING OUTCOME:

In this practical, I learned how blockchain is applied across various industries, such as finance, supply chain, and identity management. I gained hands-on experience in blockchain development, smart contracts, and decentralized applications, while also understanding the regulatory and ethical considerations, enabling me to innovate and solve real-world problems.

## REFERENCES:

1. MetaMask : <https://metamask.io/download/>
2. Ganache : <https://archive.trufflesuite.com/ganache/>
3. ChatGPT : <https://chatgpt.com/>
4. go-Ethereum : <https://geth.ethereum.org/downloads>
5. Truffle Suite : <https://archive.trufflesuite.com/docs/truffle/how-to/install/>