**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 |

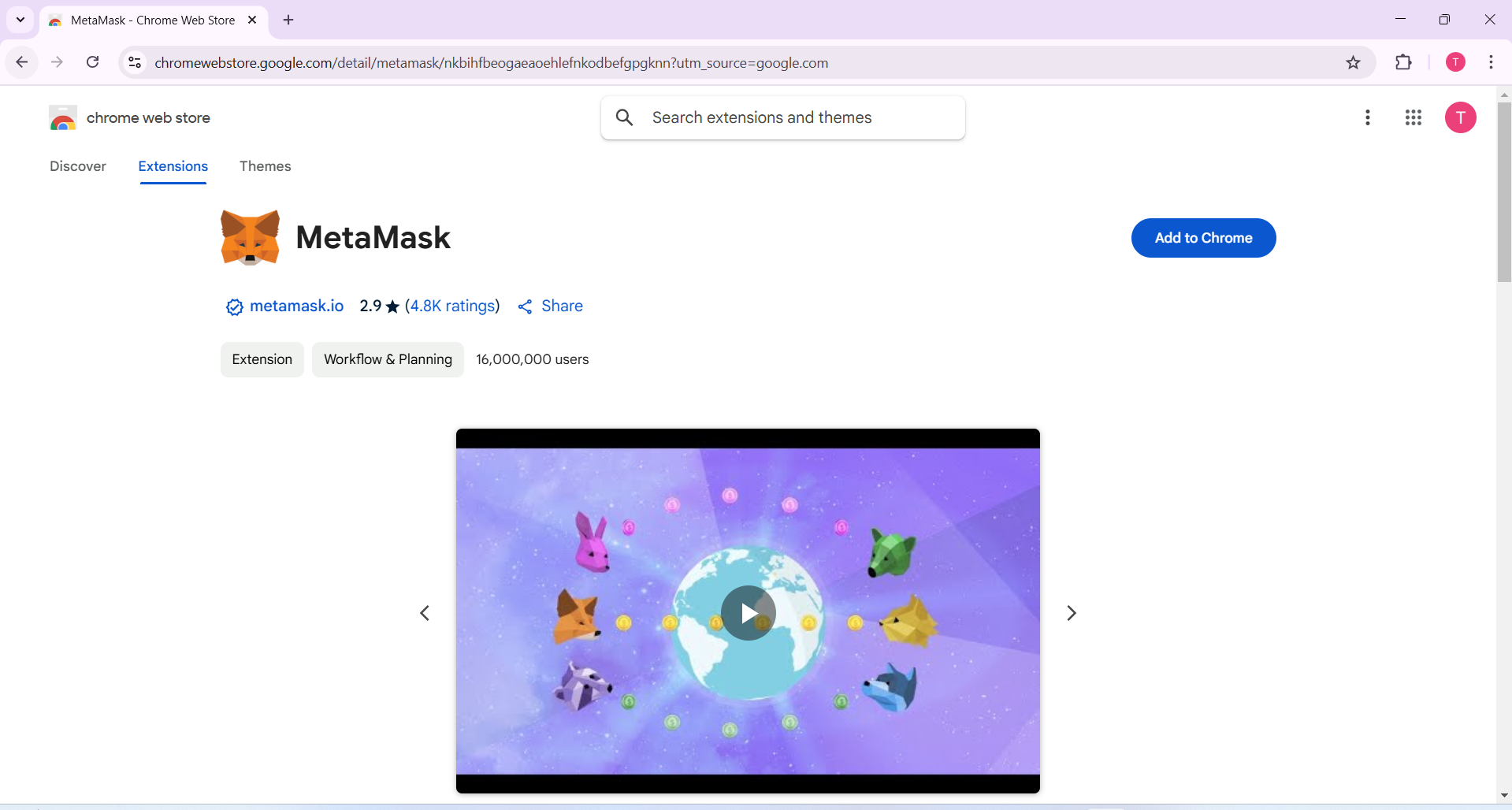
**OUTPUT:** 

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

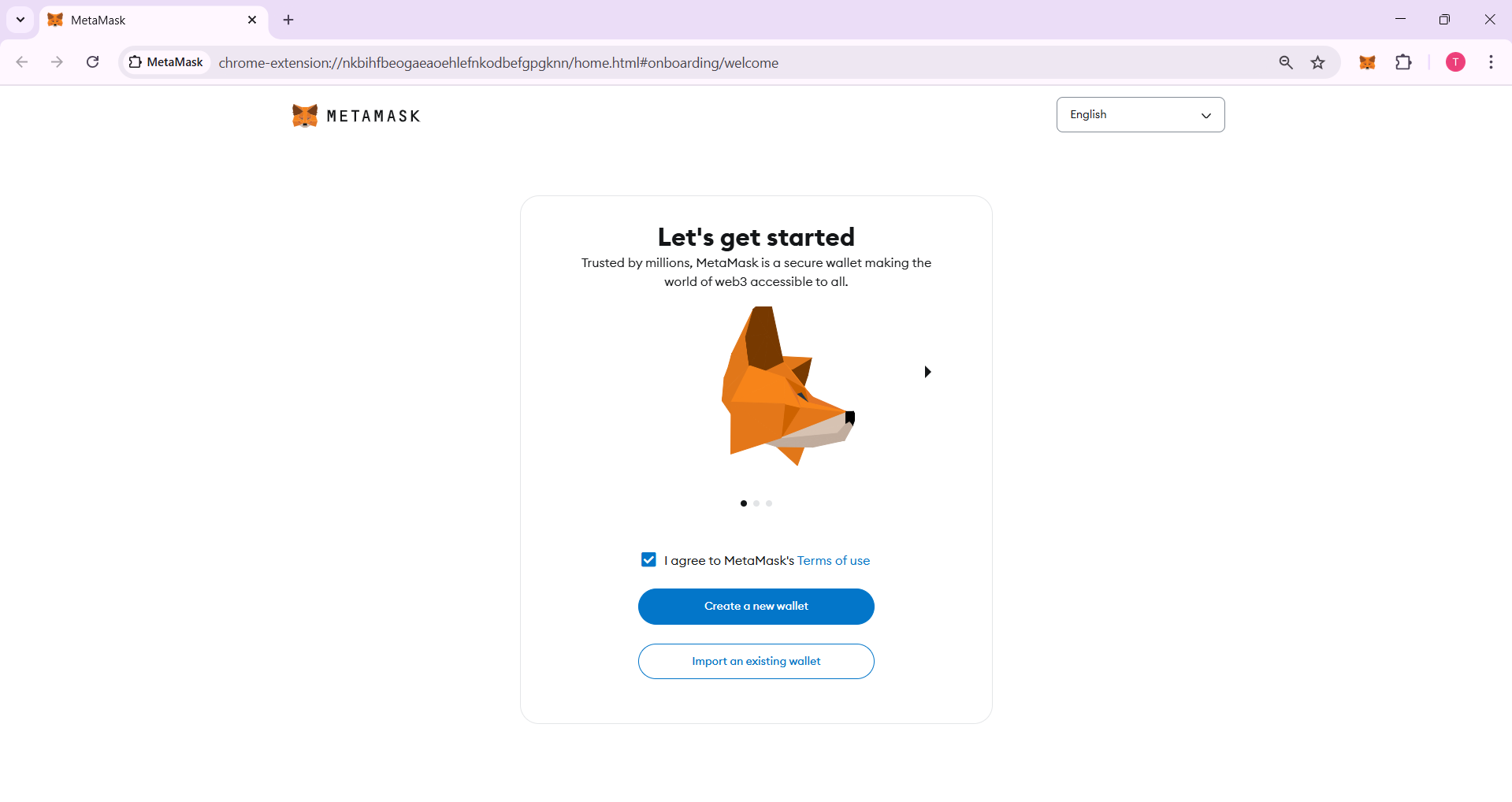


Figure 2:After installing MetaMask create a new wallet



Figure 3:MetaMask wallet create successfully

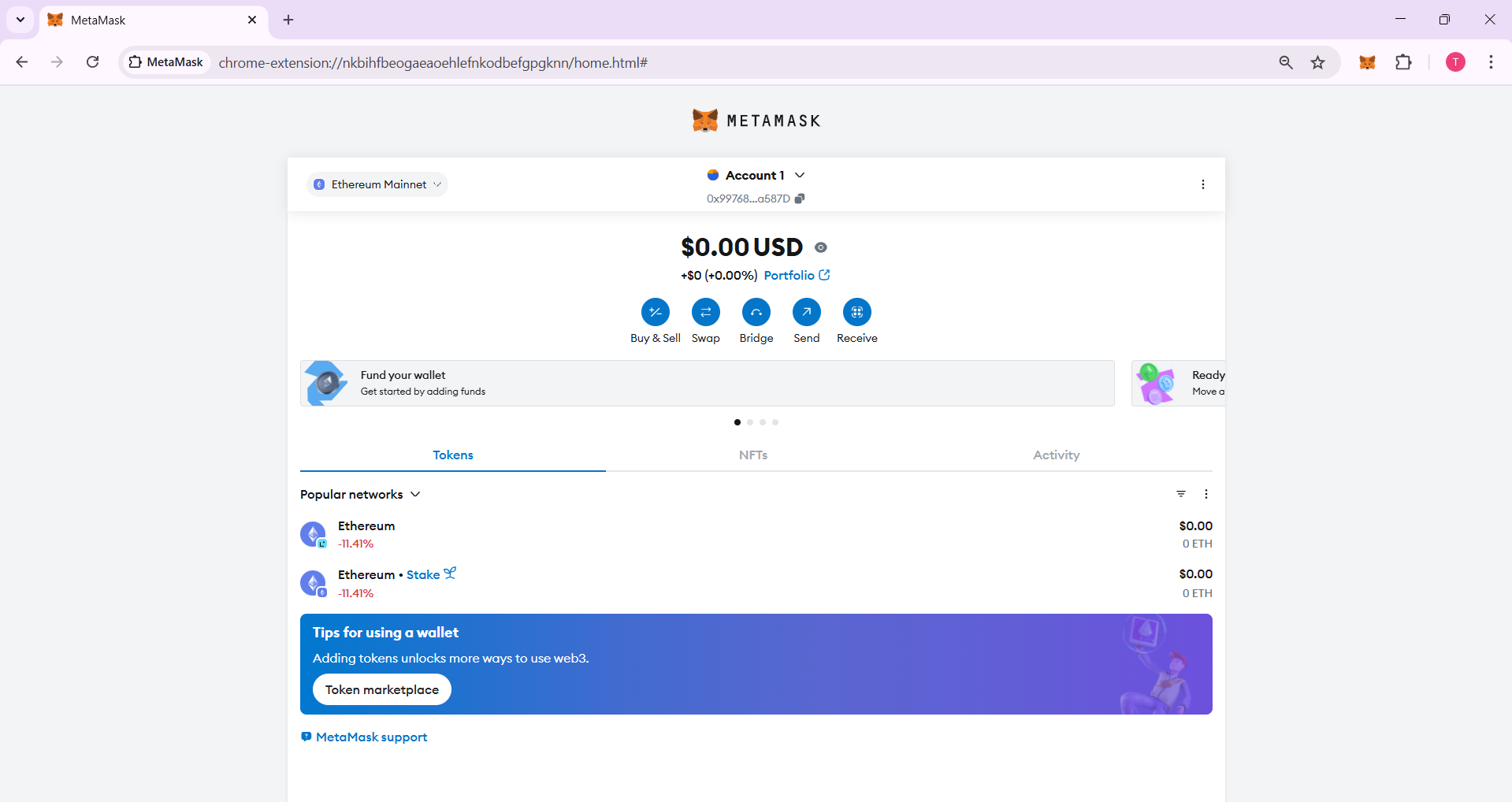


Figure 4:In Account-1 wallet has 0.00$ USD

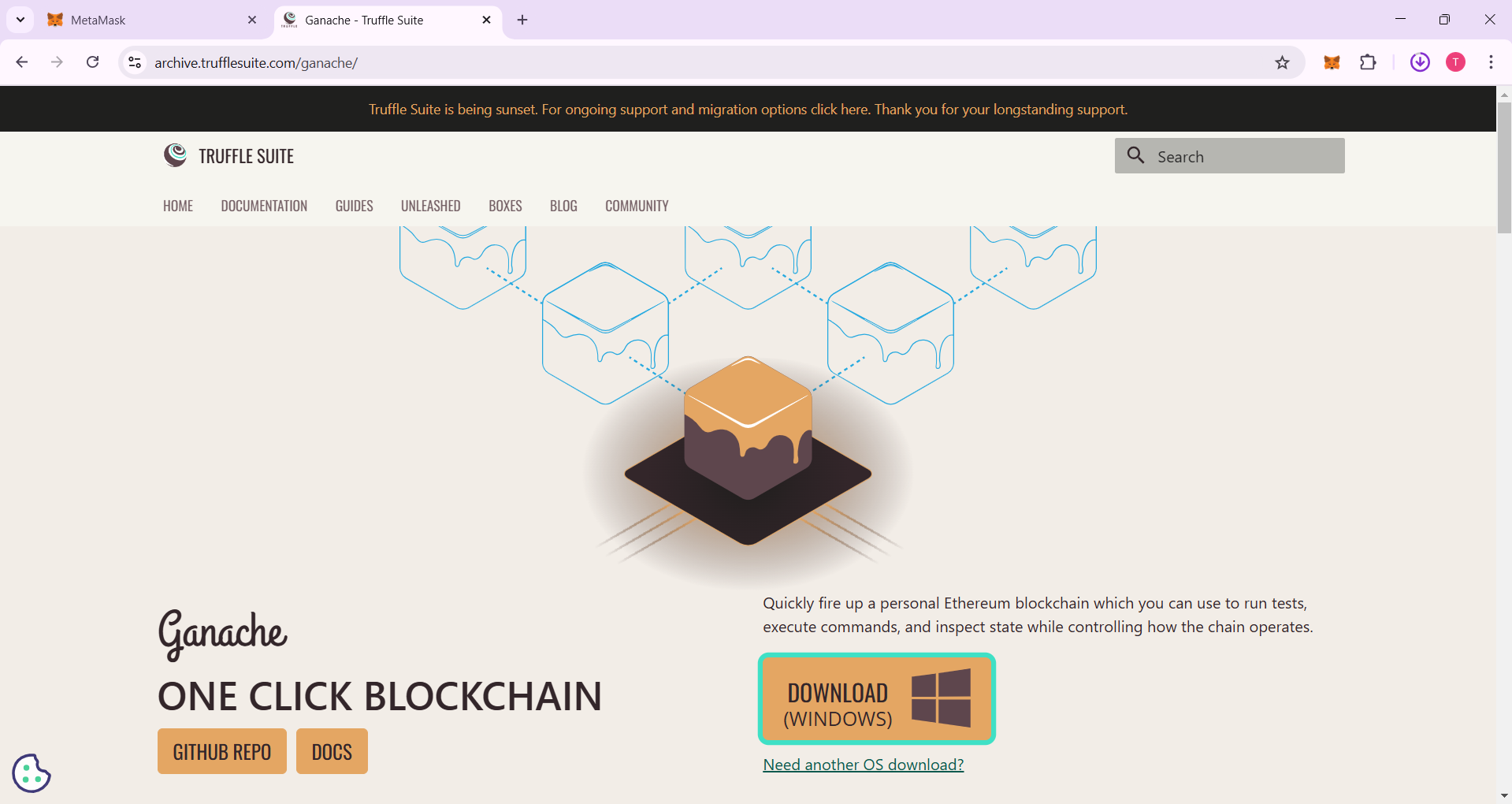


Figure 5:Download Ganache form above website

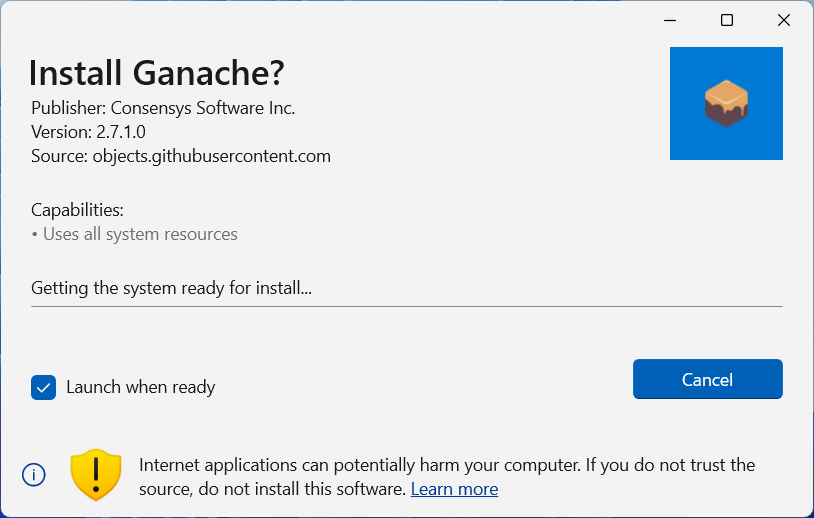


Figure 6:Install Ganache and then launch it

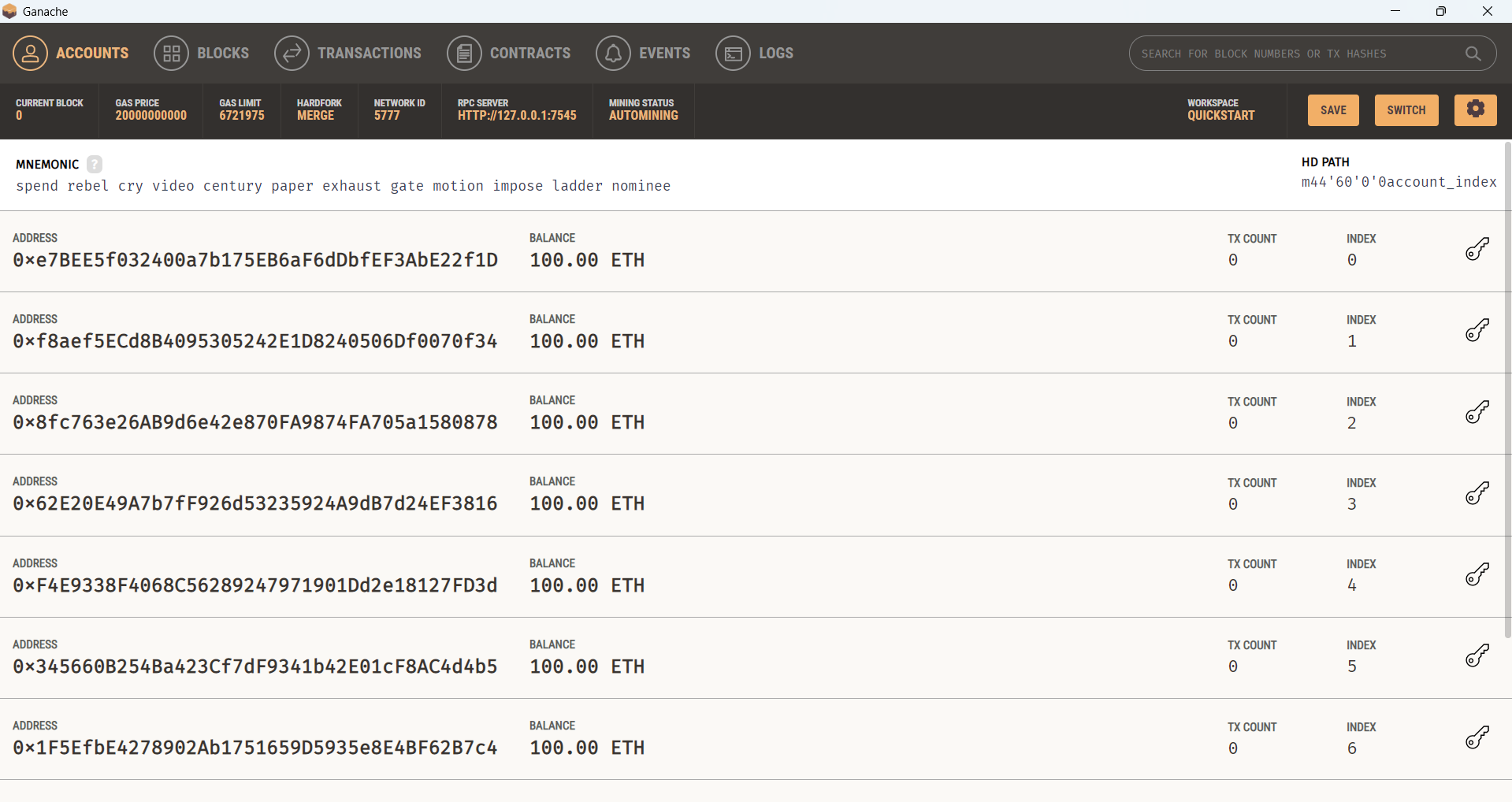


Figure 7:GUI of Ganache showing address and balance

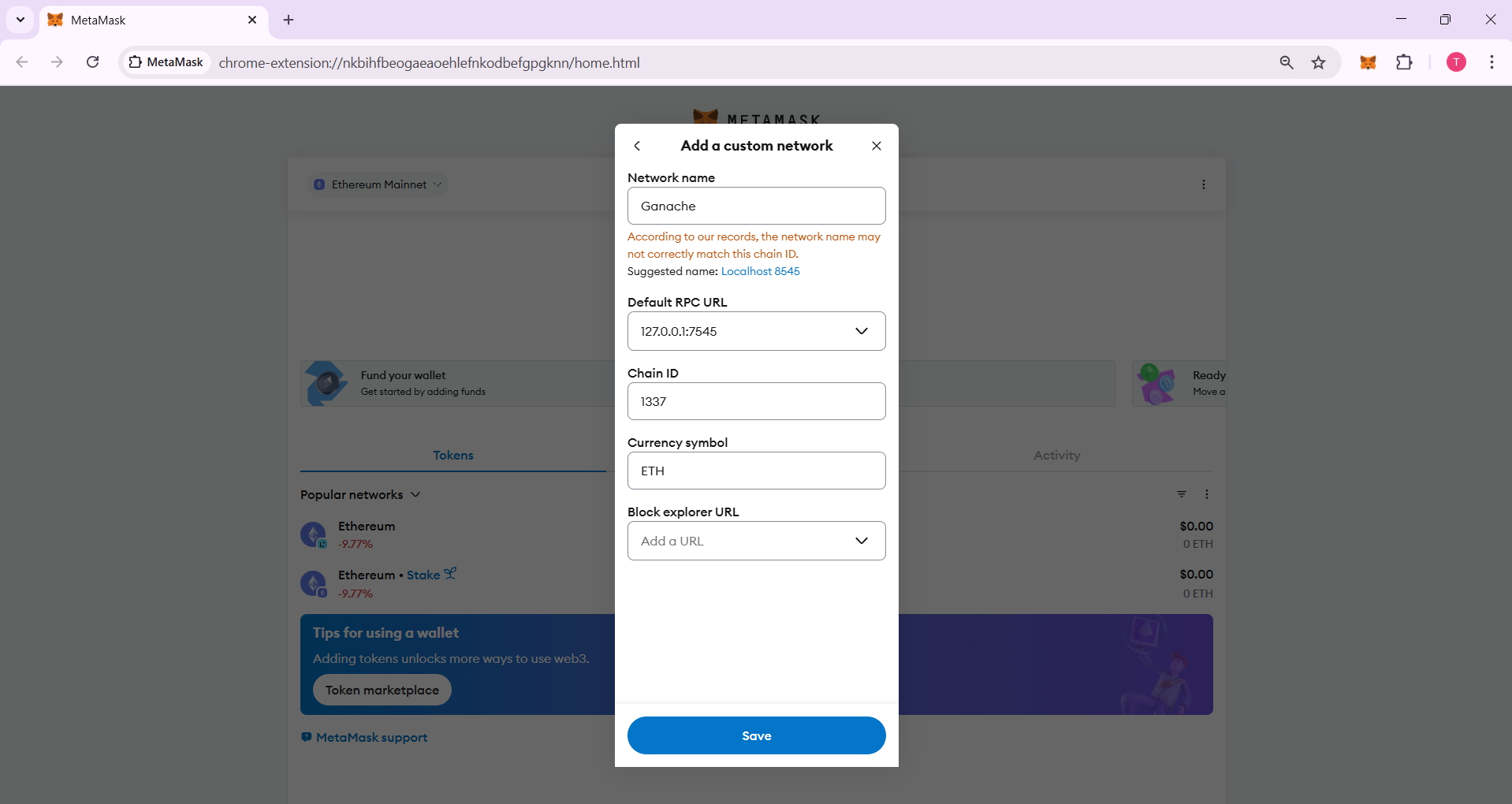


Figure 8:Add Custom network to MetaMask

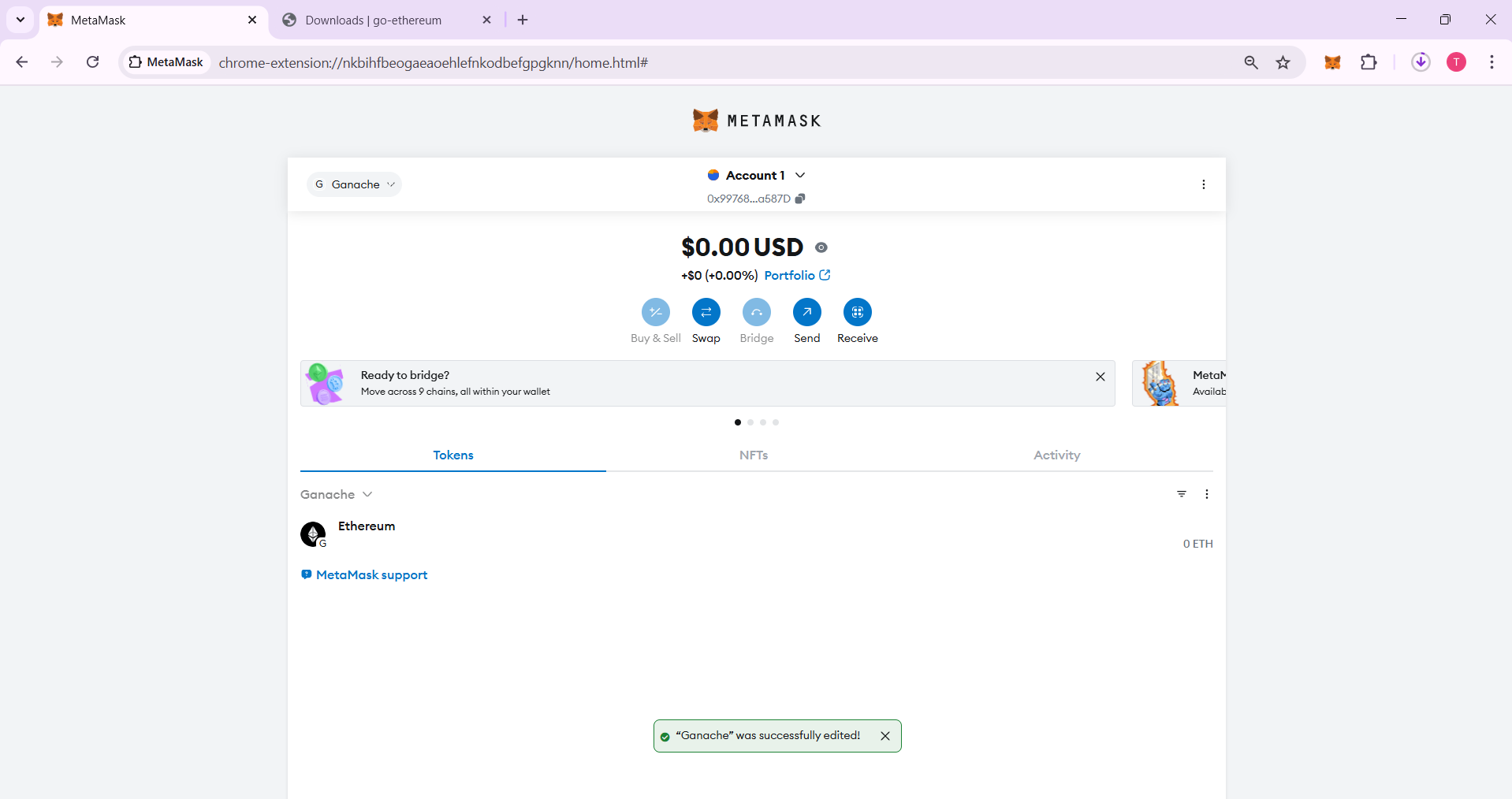


Figure 9: Adding Ganache network to my wallet



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

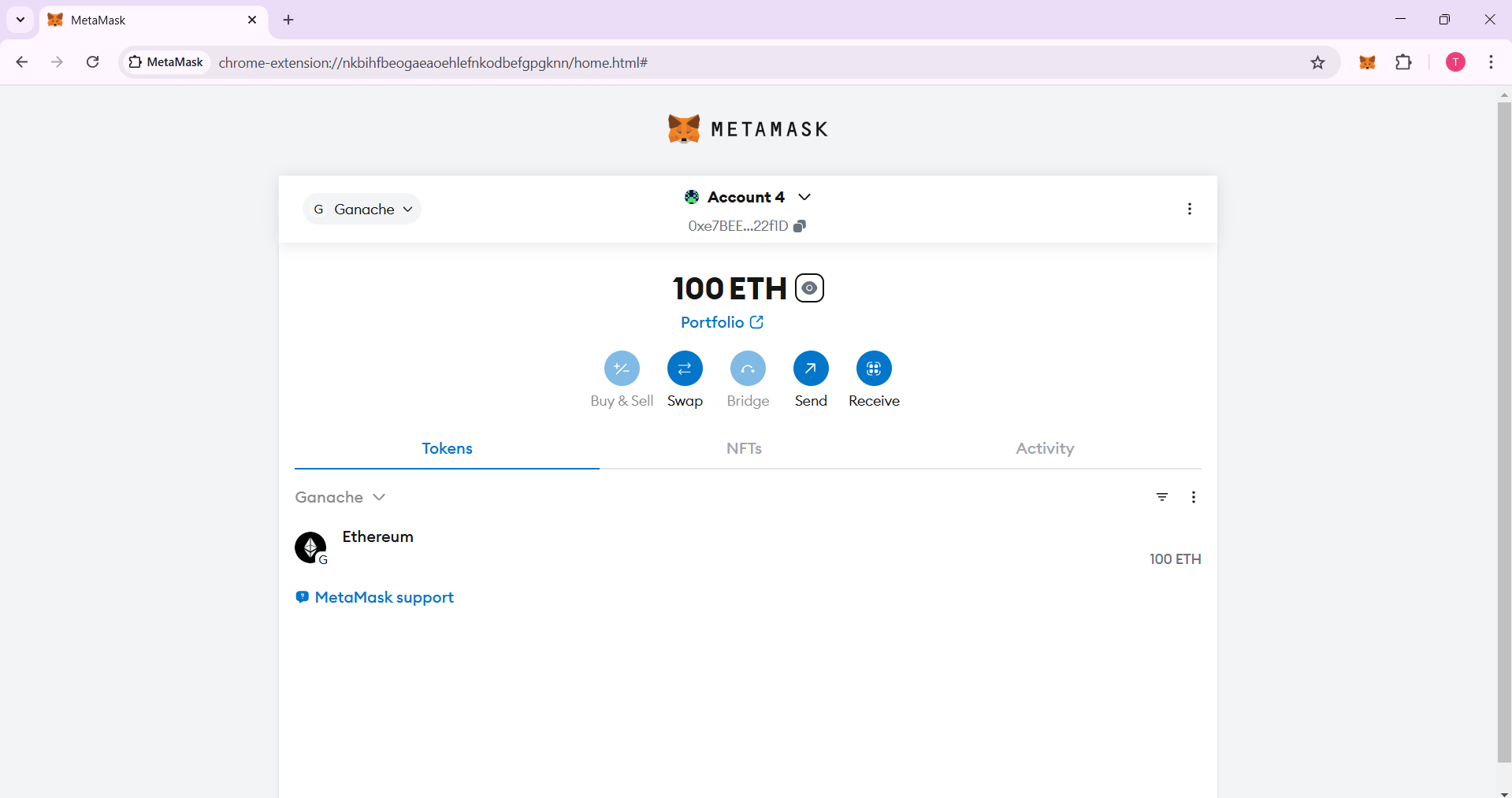


Figure 11: Account import in MetaMask

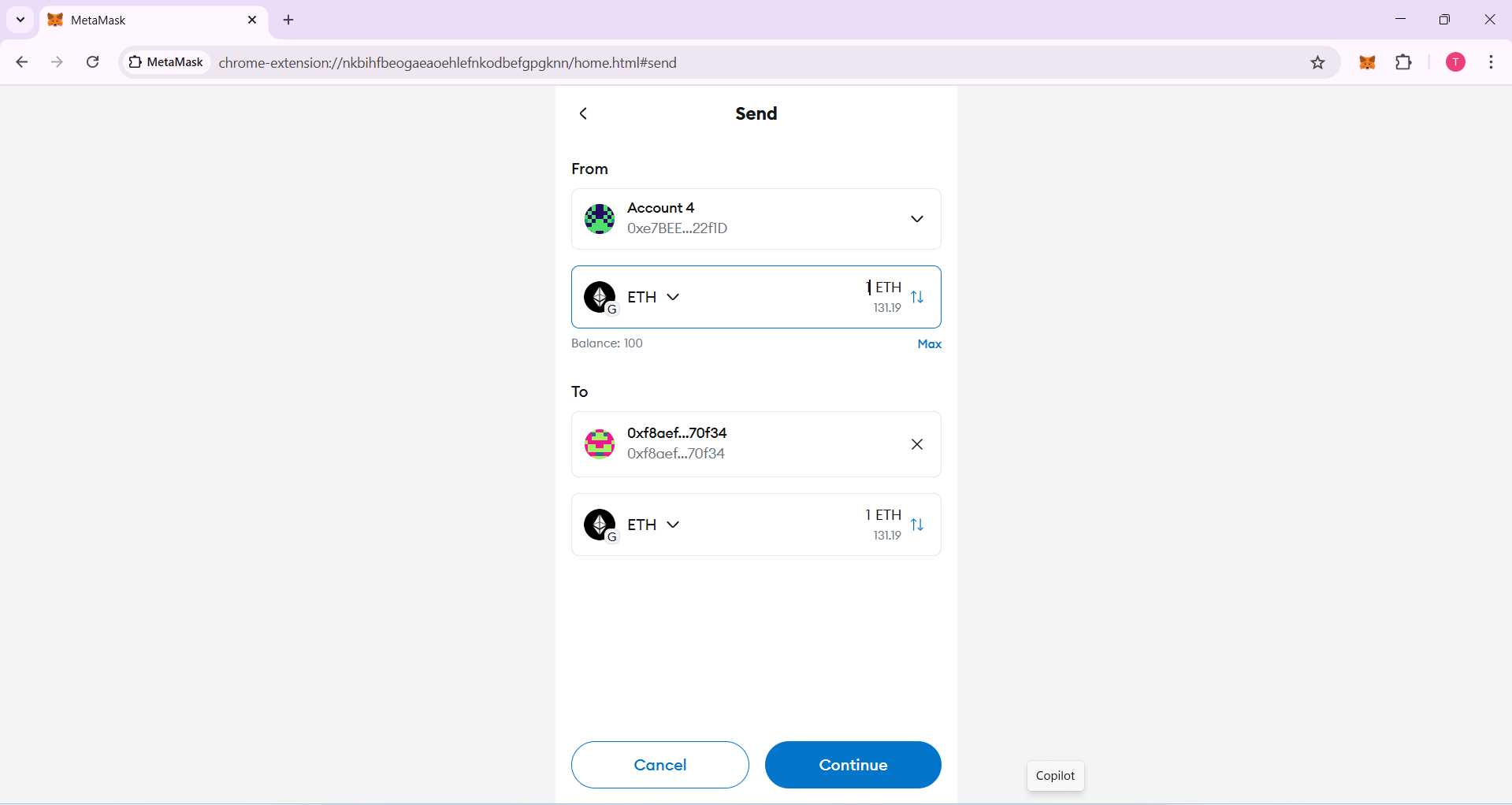


Figure 12:Sending 1 ETH to another account

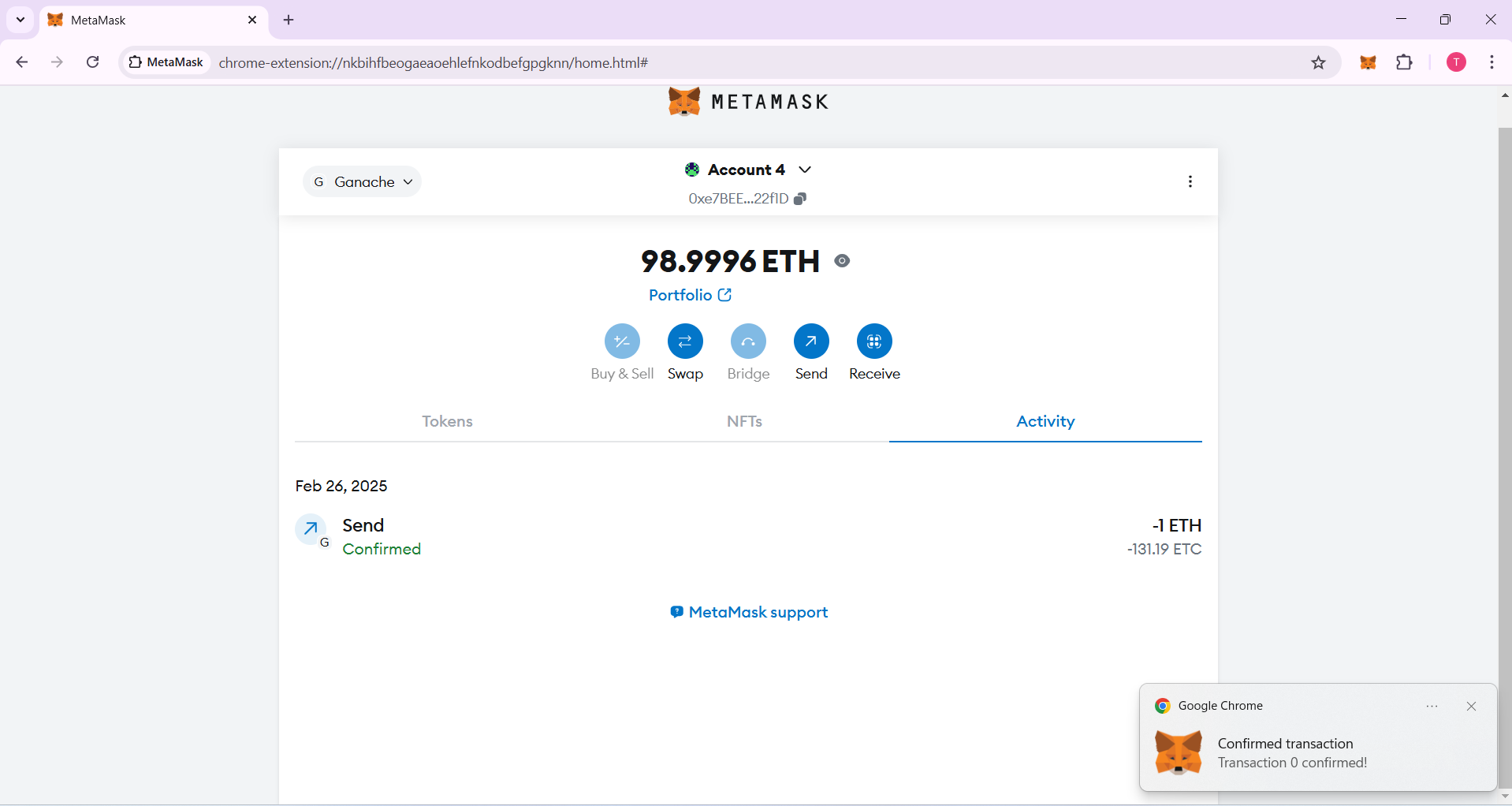


Figure 13:Sending 1 ETH successfully

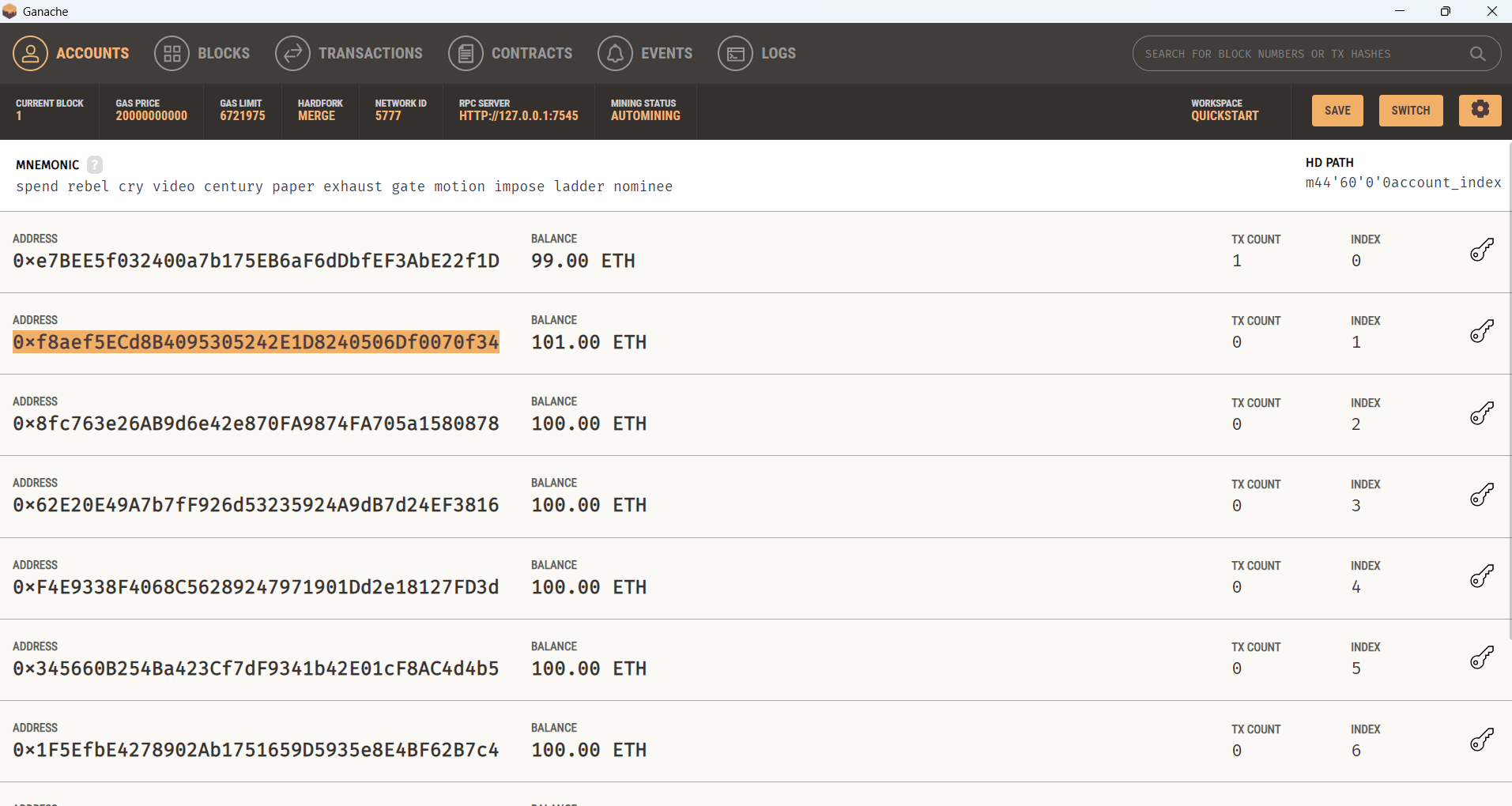


Figure 14:Successfully receiving 1ETH proof

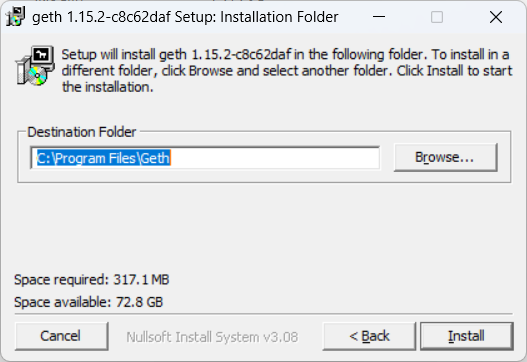


Figure 15:Install Go-Ethereum (Geth) Client



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

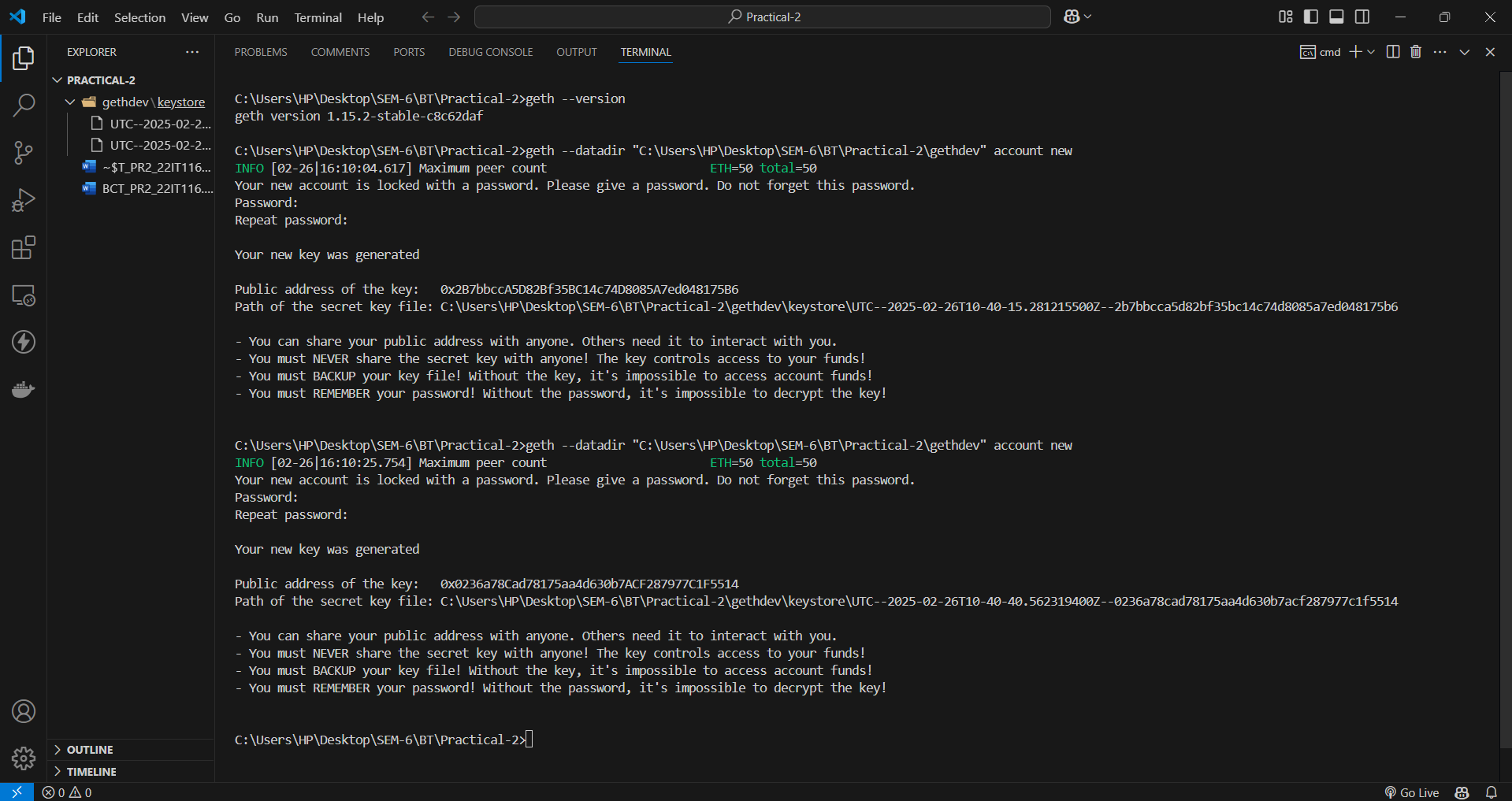


Figure 17:Verify Go-Ethereum(Geth) Client install and create two new accounts

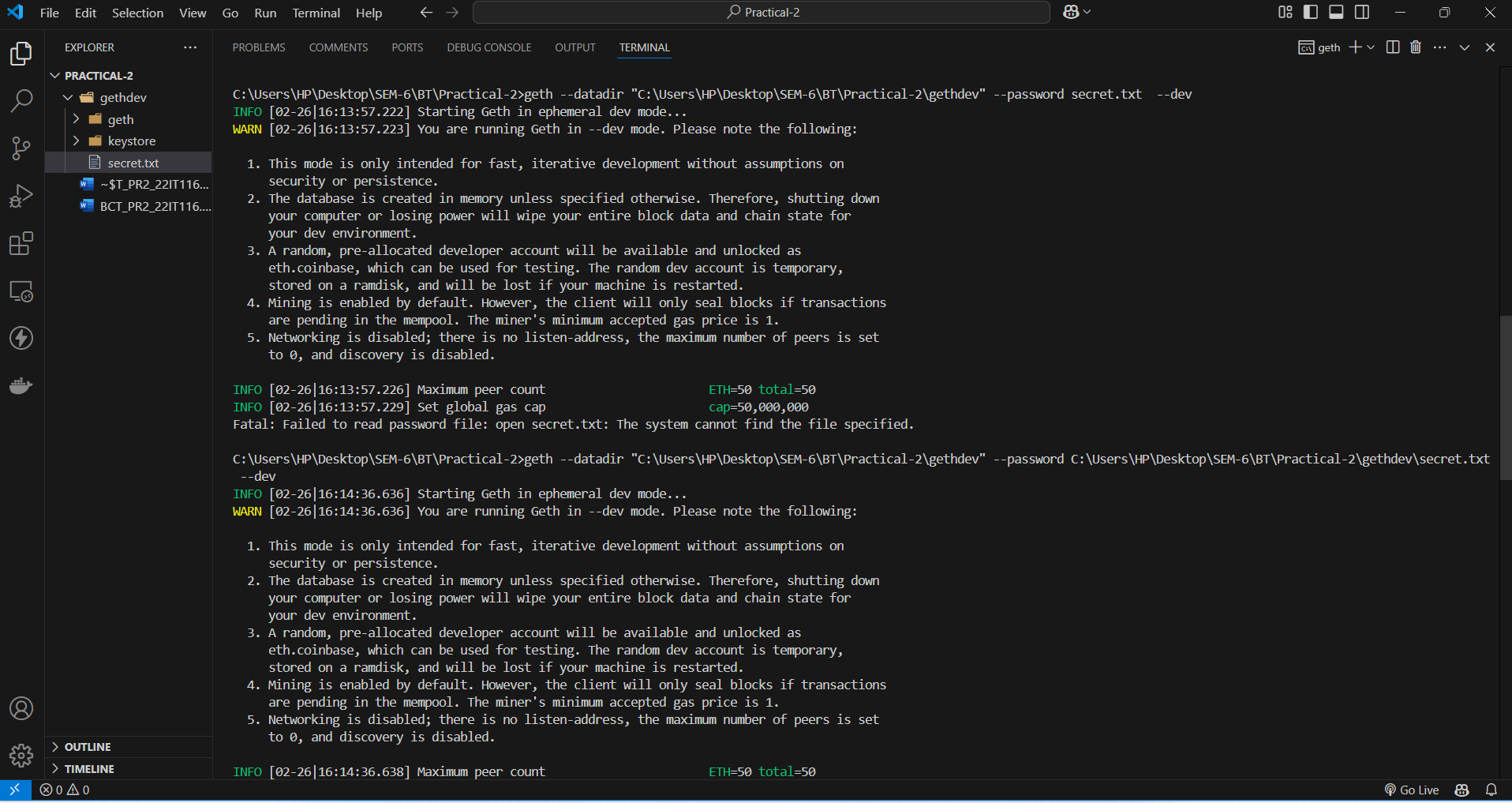


Figure 18:Open account in developer mode

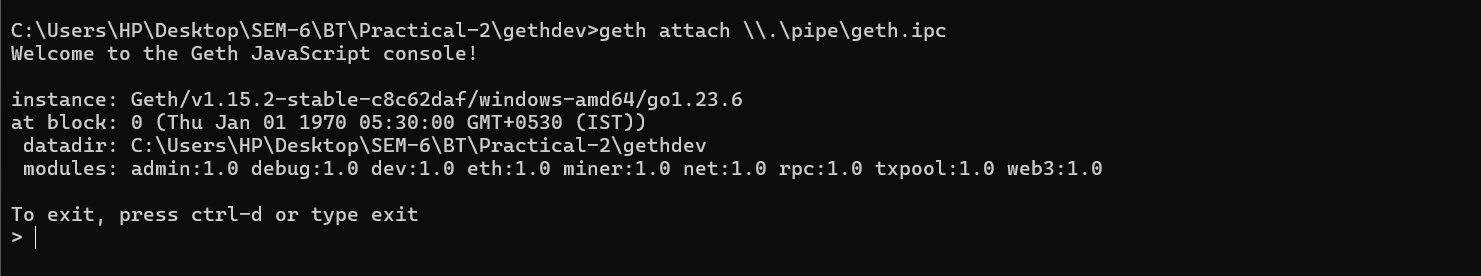


Figure 19:Open account in JavaScript console

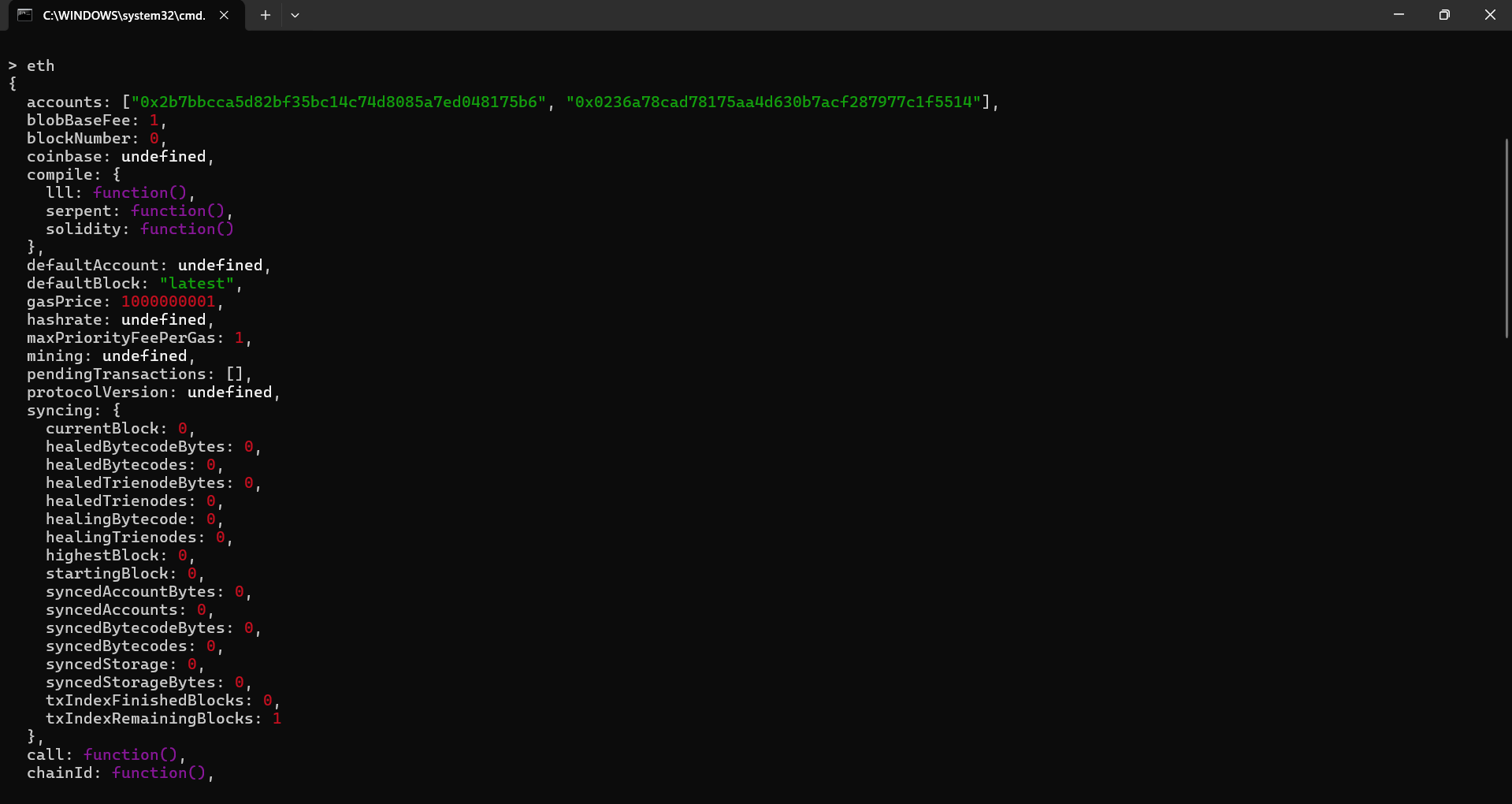


Figure 20:Functions provided by eth library

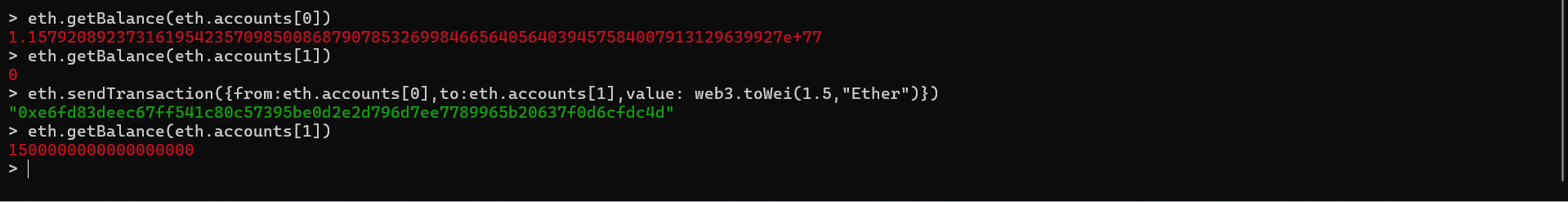


Figure 21:Transfer money from one account to another account

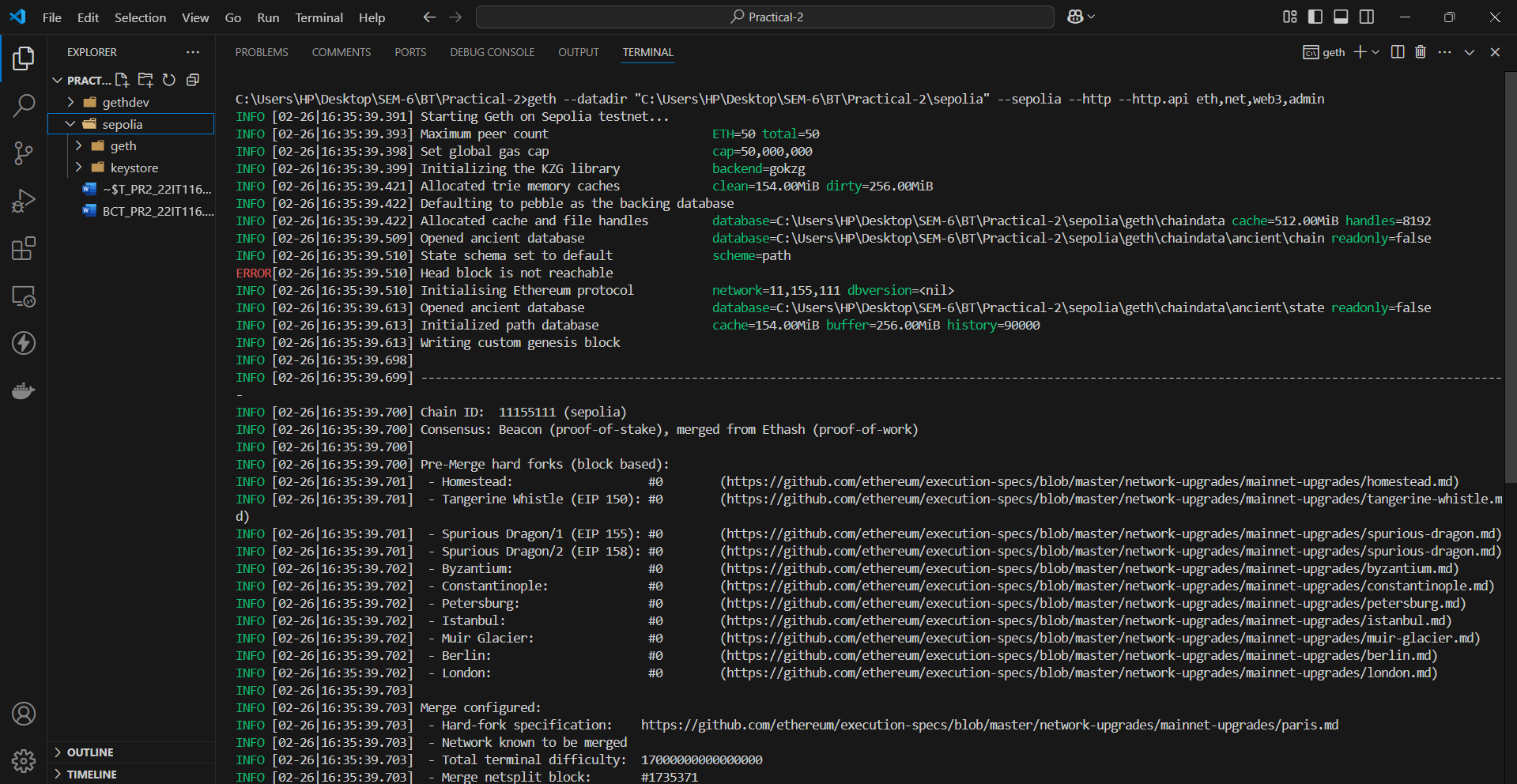


Figure 22:Connect with sepolia network

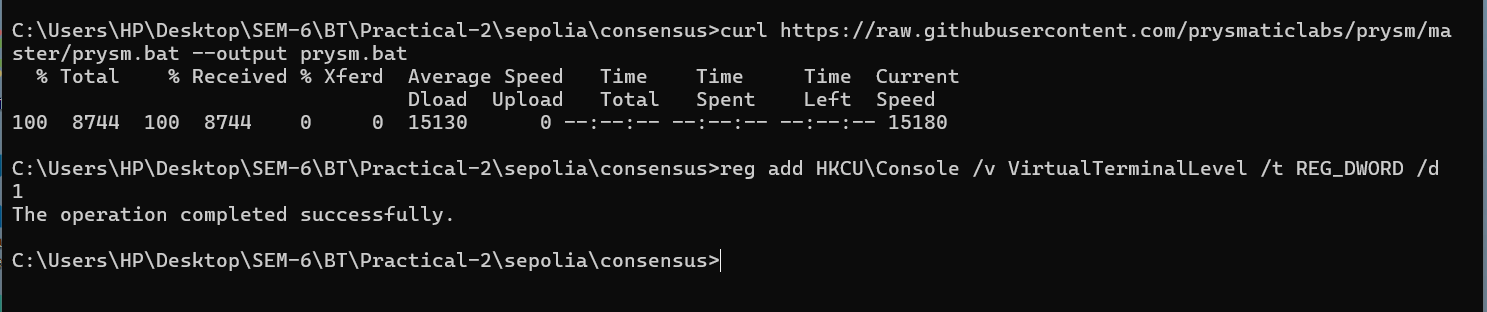


Figure 23:Download consensus prysm and register

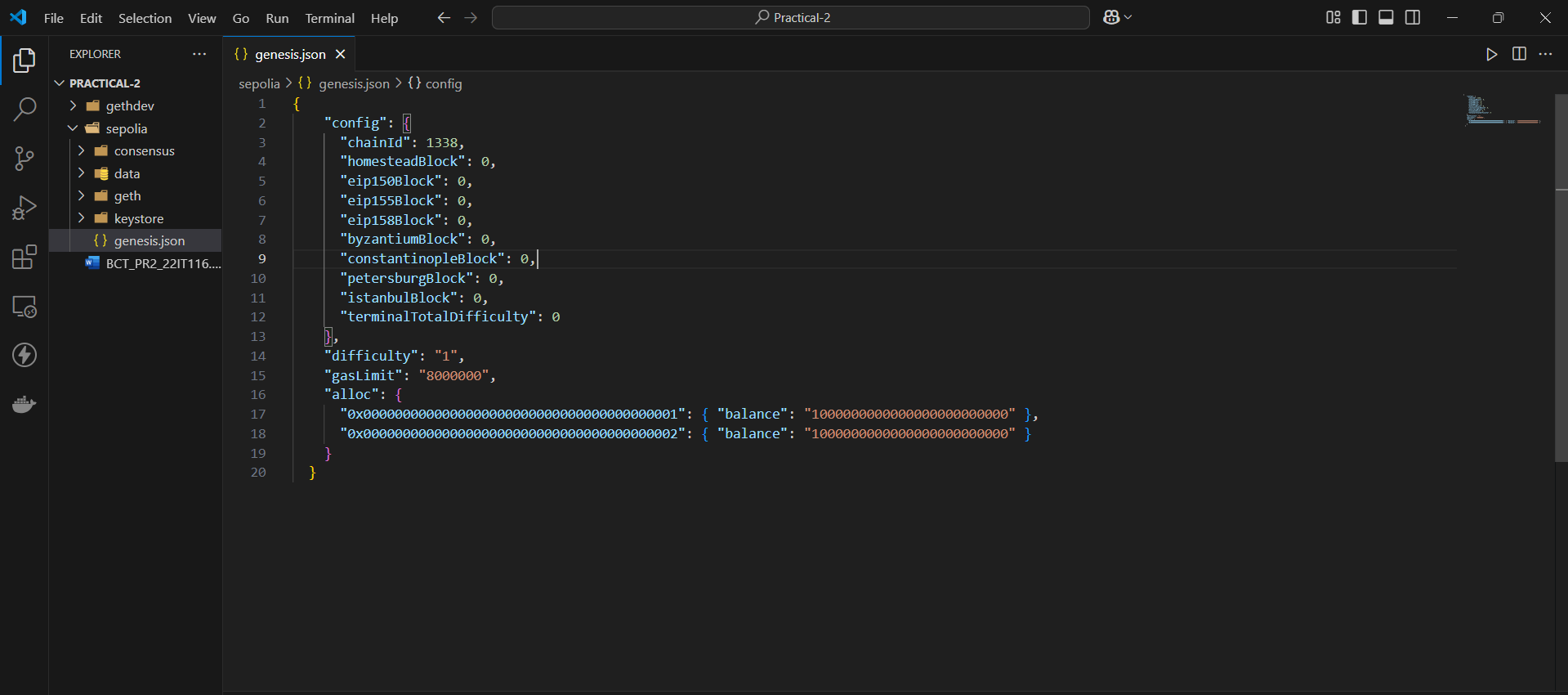


Figure 24:Create a genesis.json file

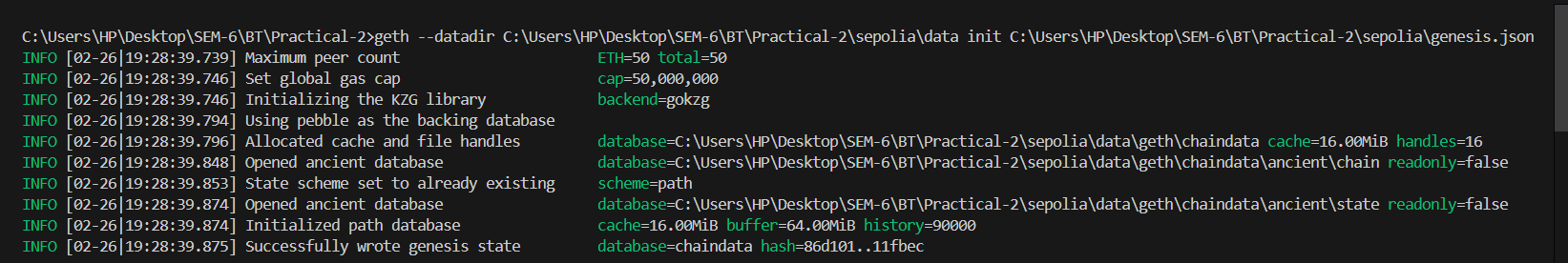


Figure 25:Initialize Geth with the genesis.json

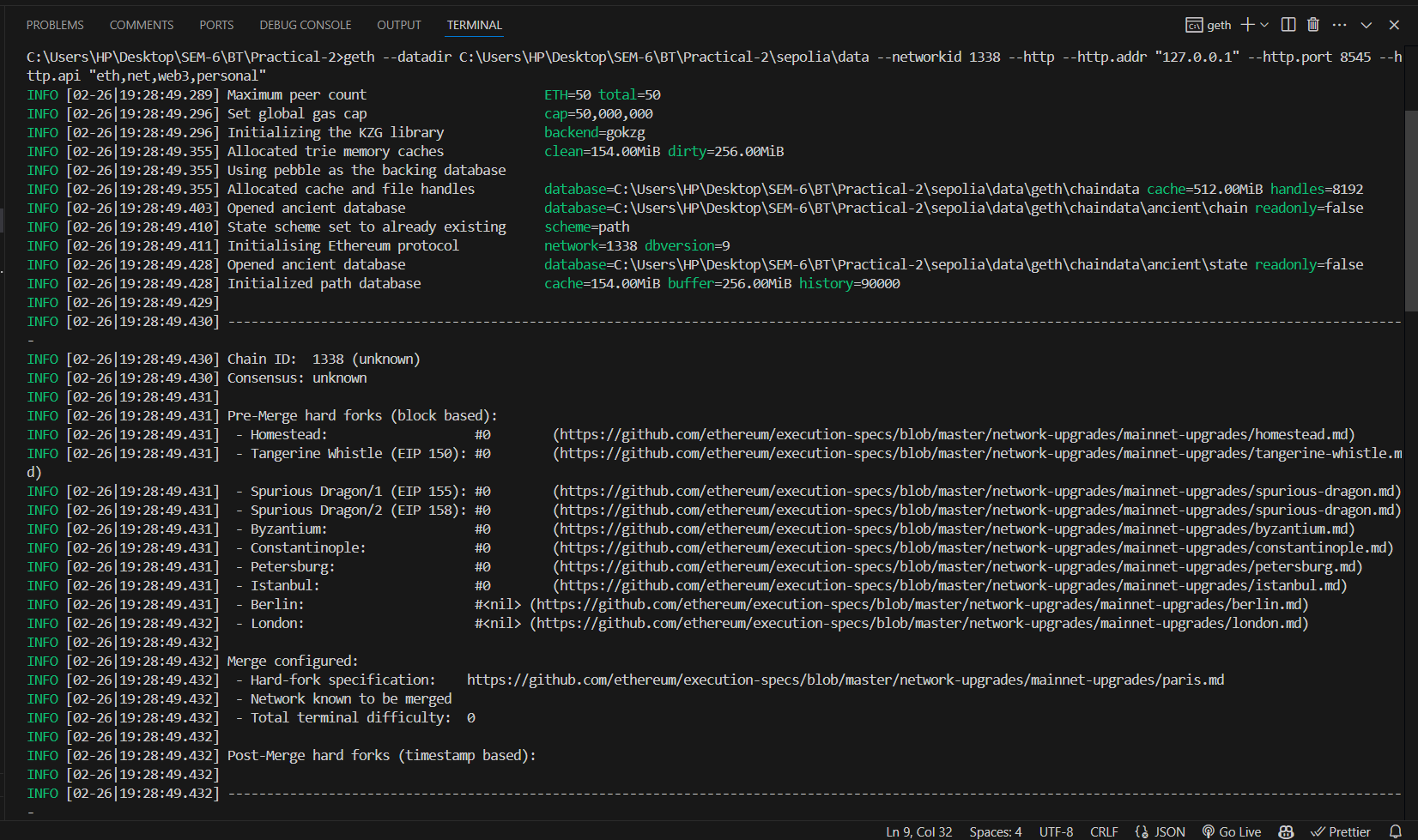


Figure 26:Start Geth with the data directory

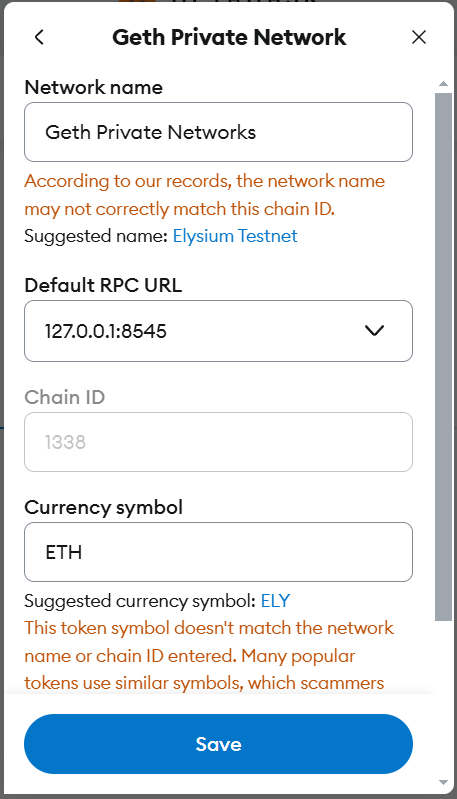


Figure 27:Add Geth network in MetaMask

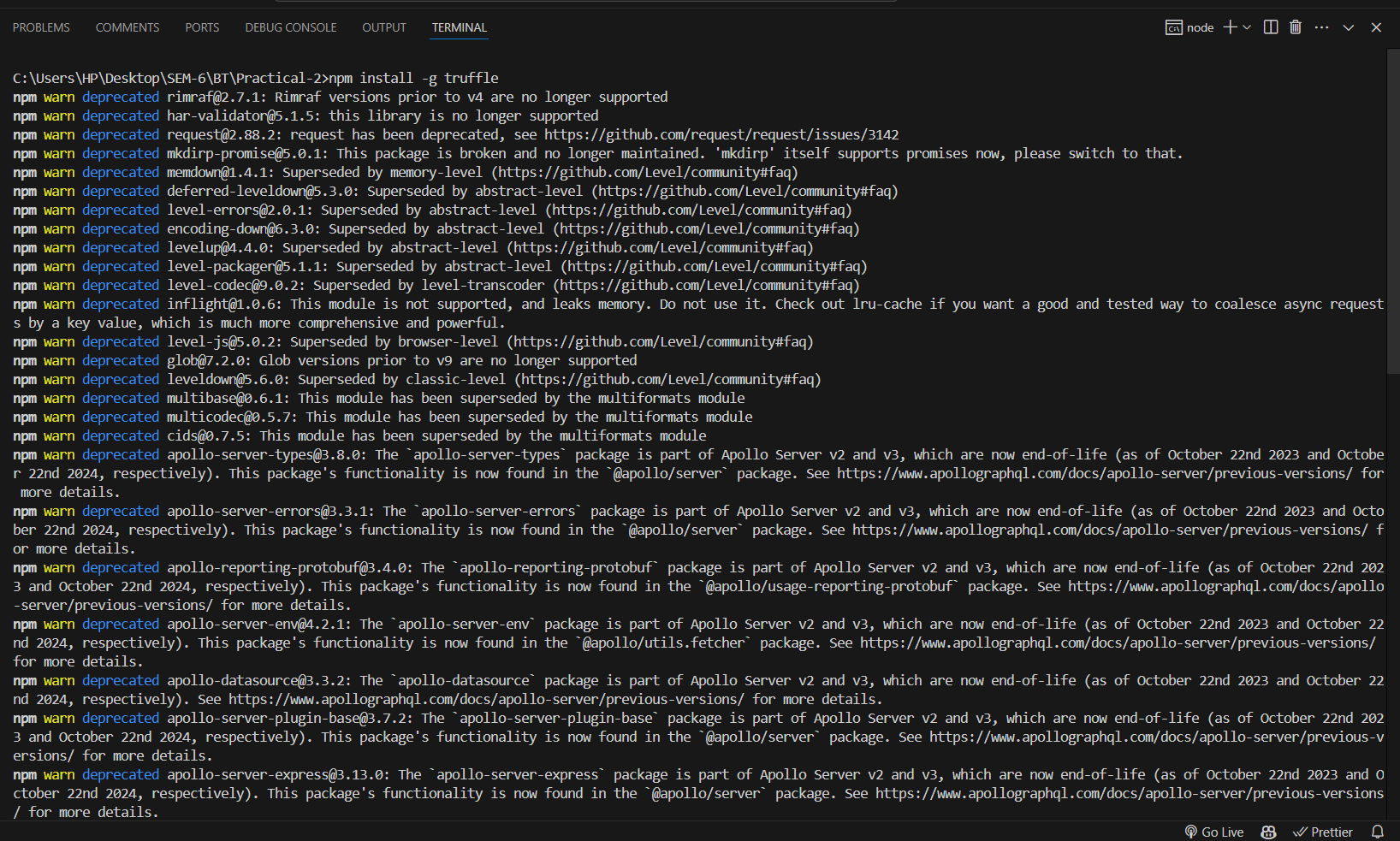


Figure 28:Install Truffle globally

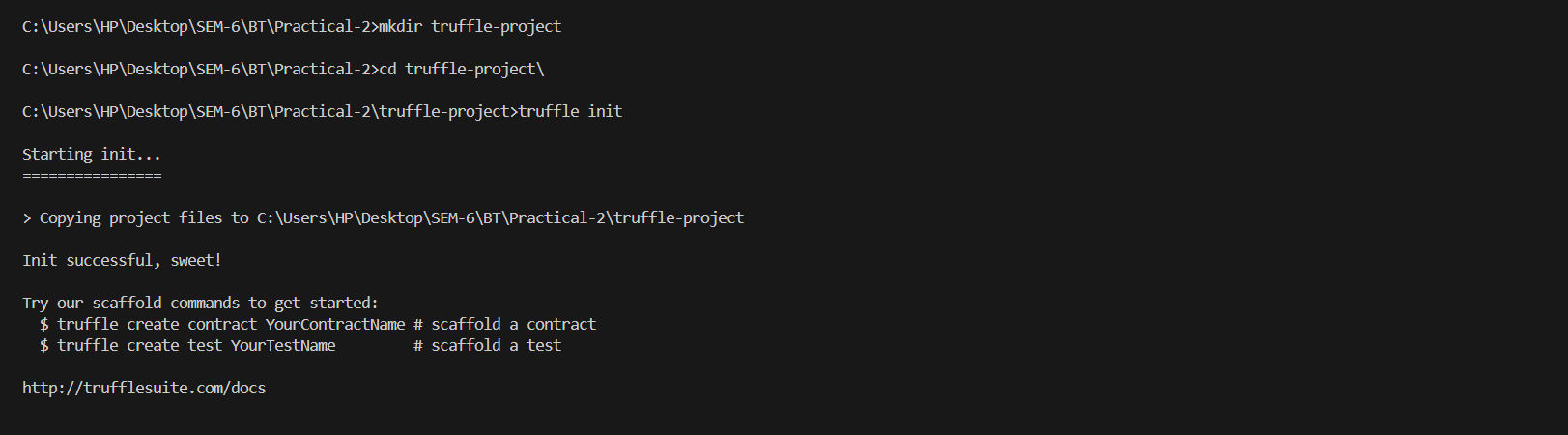


Figure 29:Initialize a new Truffle project

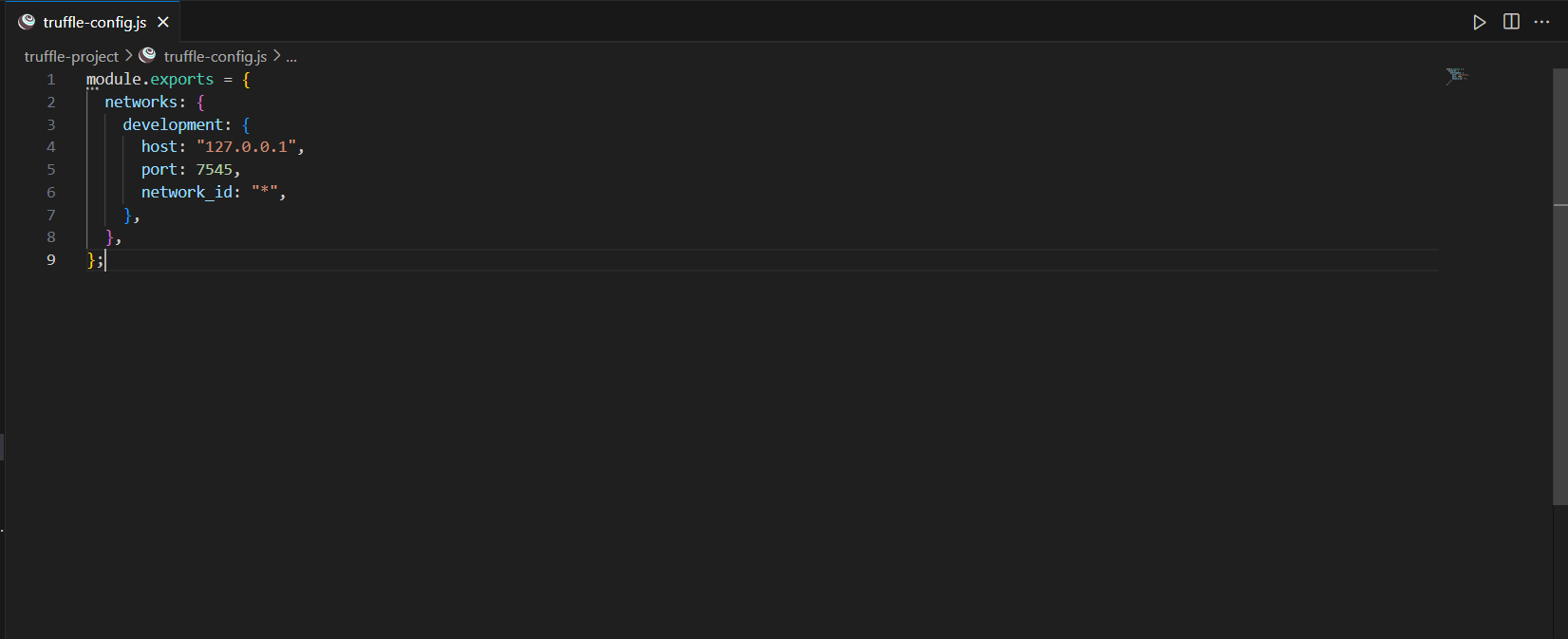


Figure 30:Configure the truffle-config.js file to connect Ganache

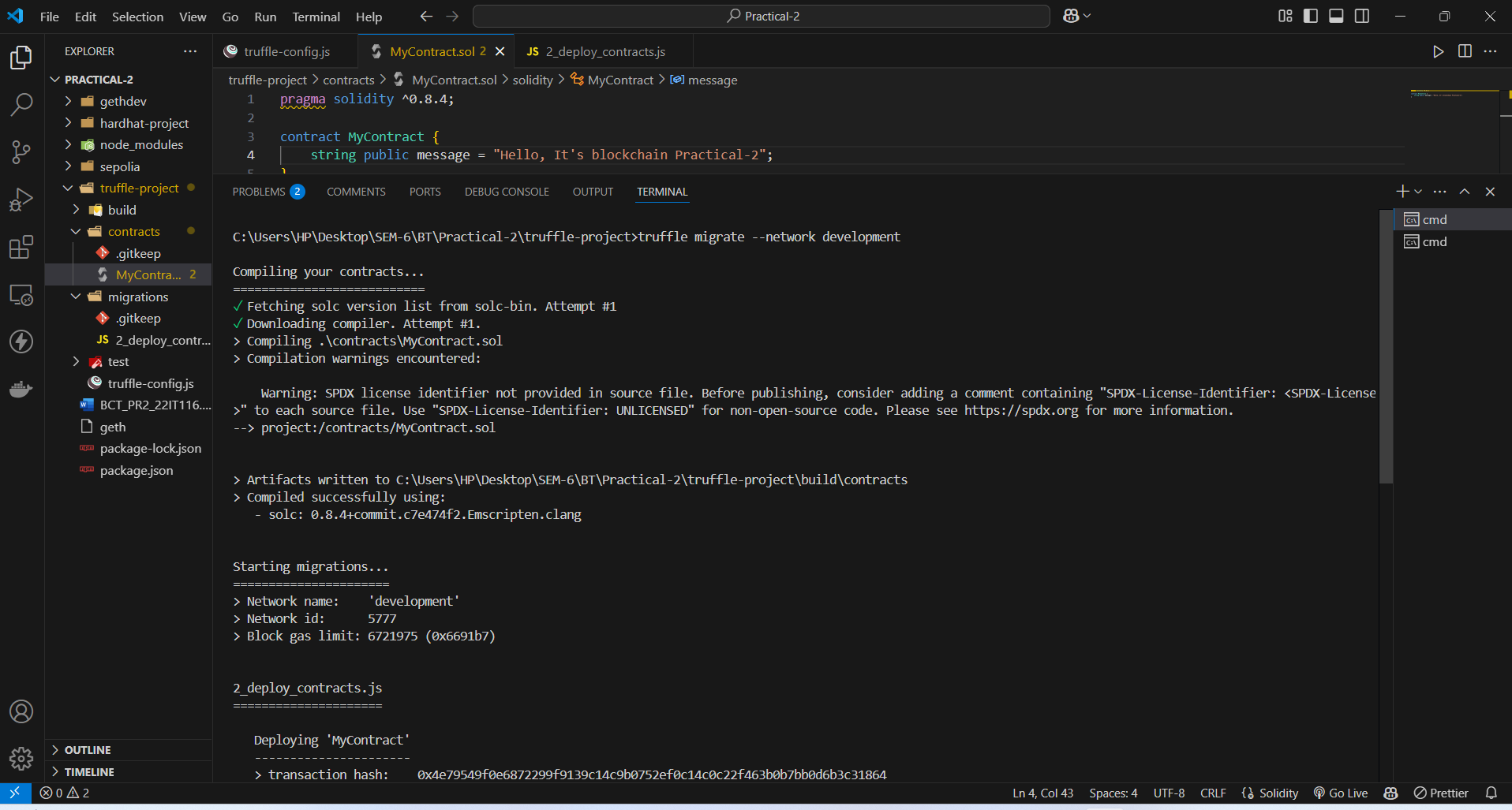


Figure 31:Deploy My contract

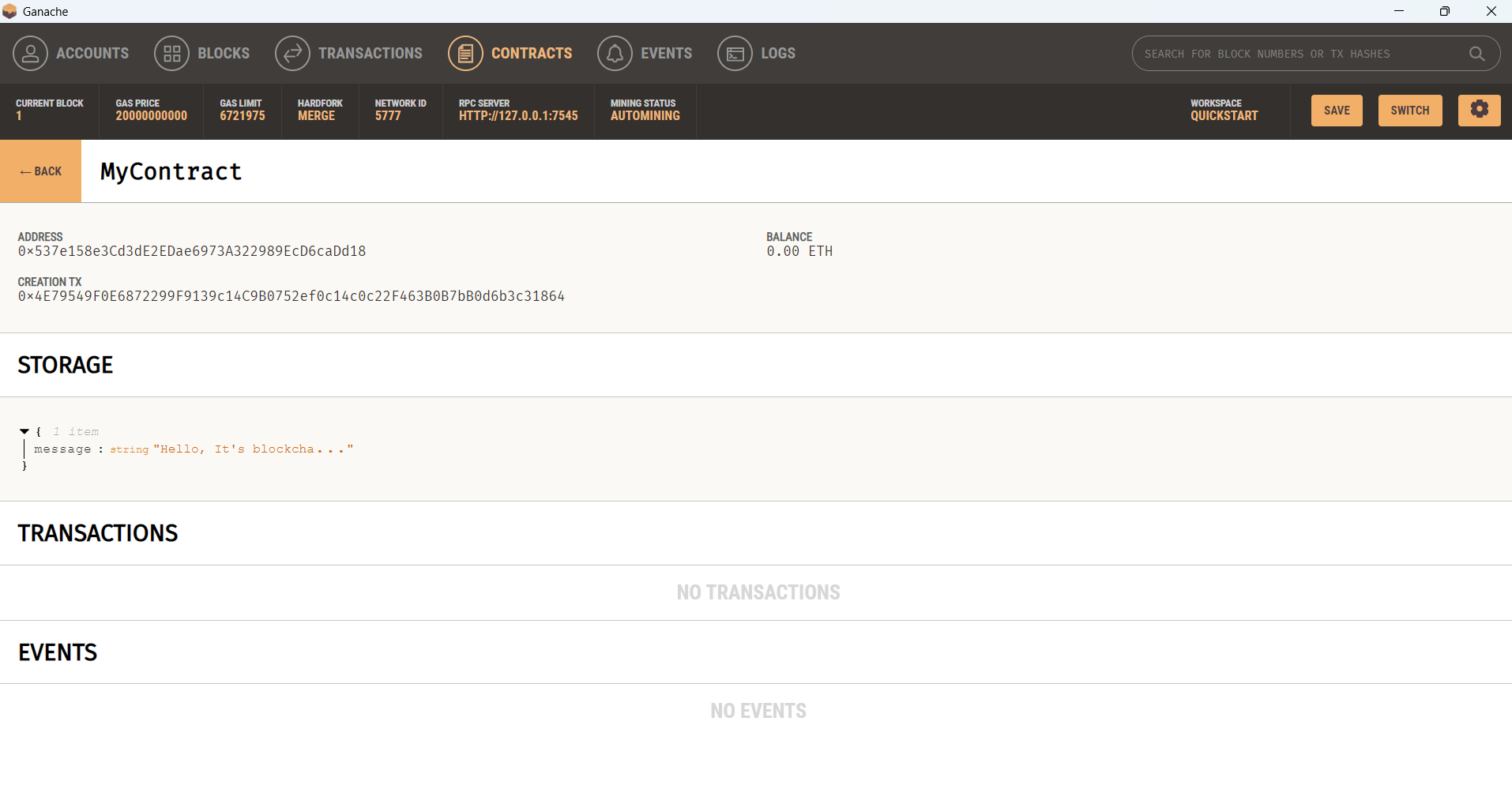


Figure 32:Verify that contract run successfully

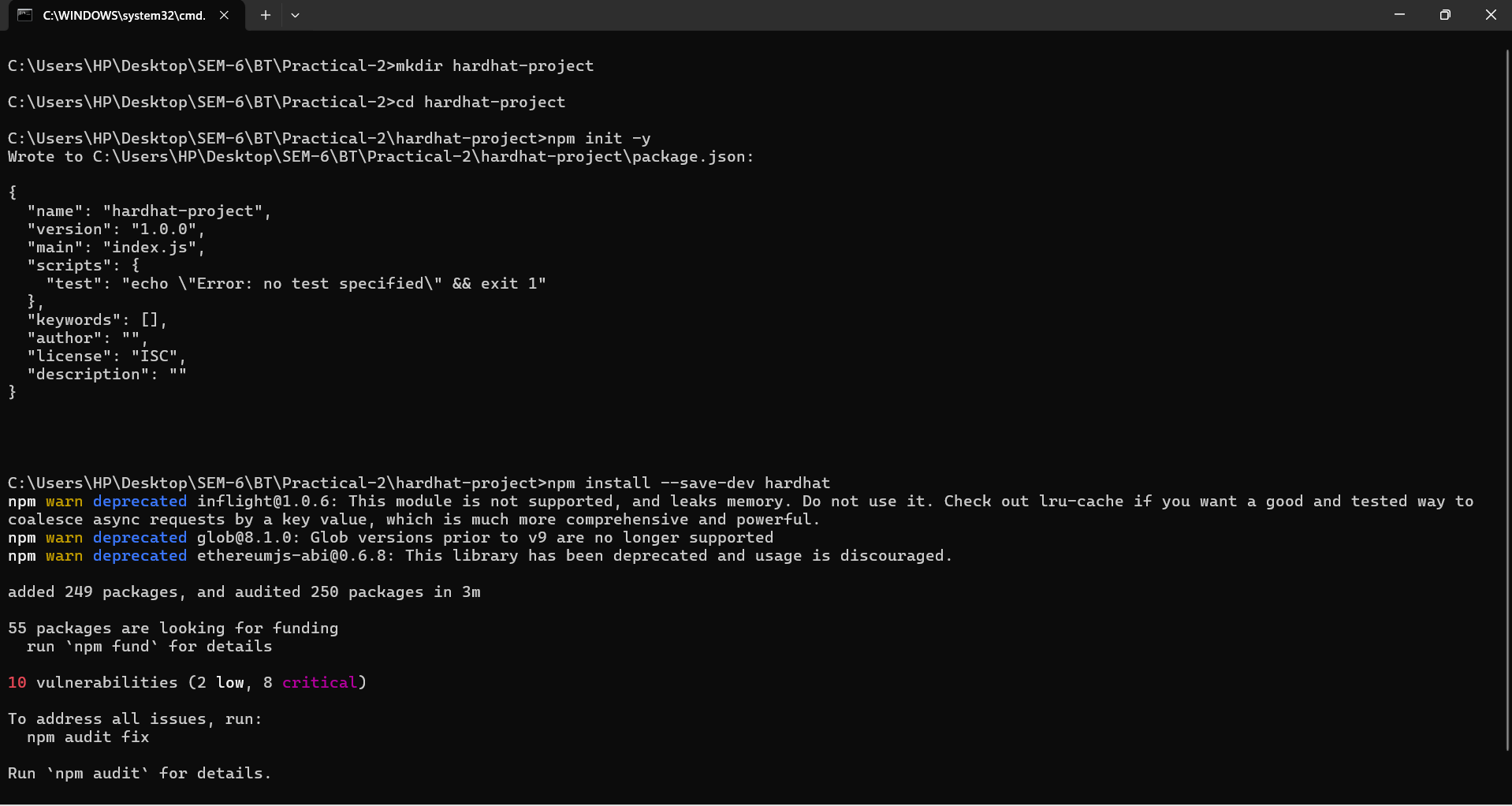


Figure 33:Install Hardhat

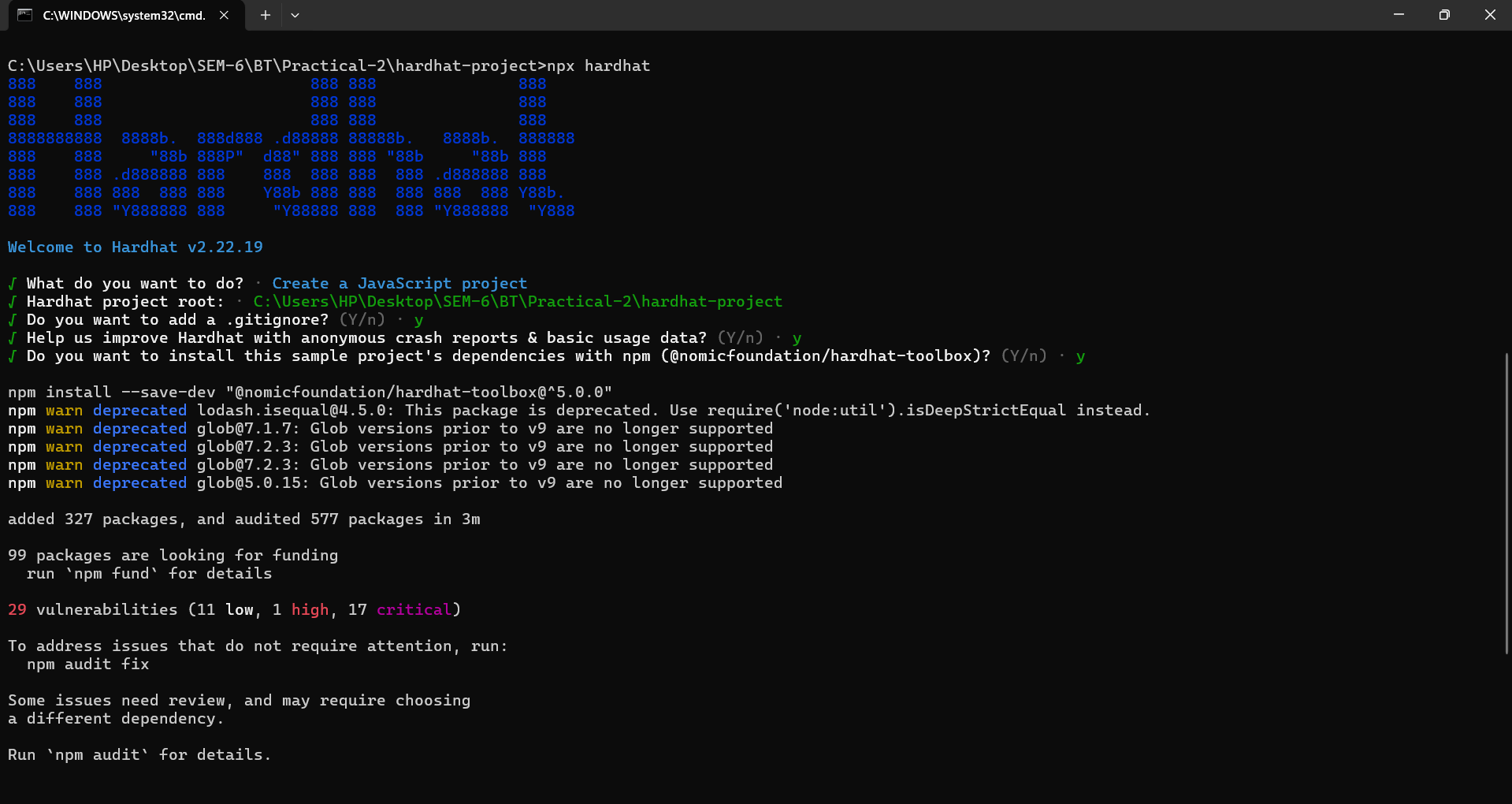


Figure 34:Initialize a new Hardhat project

**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/>