Name of Student : Neha Yadav		
Roll Number : 62		**LAB Assignment Number :** 06
Title of LAB Assignment : Install Ganache(Personal block chain) and MetaMask (Show the installation Steps) . Compile and deploy an election smart contract in the personal blockchain using injected web 3 environment(MetaMask wallet) .Use Remix online ide to compile and deploy the smart contract.		
DOP : 4th September 2024		**DOS :** 18th September 2024
CO Mapped : CO2, CO3	**PO Mapped :** PO2, PO3, PSO1	**Signature :**

AIM: Install Ganache(Personal block chain) and MetaMask (Show the installation Steps) . Compile and deploy an election smart contract in the personal blockchain using injected web 3 environment(MetaMask wallet) .Use Remix online ide to compile and deploy the smart contract.

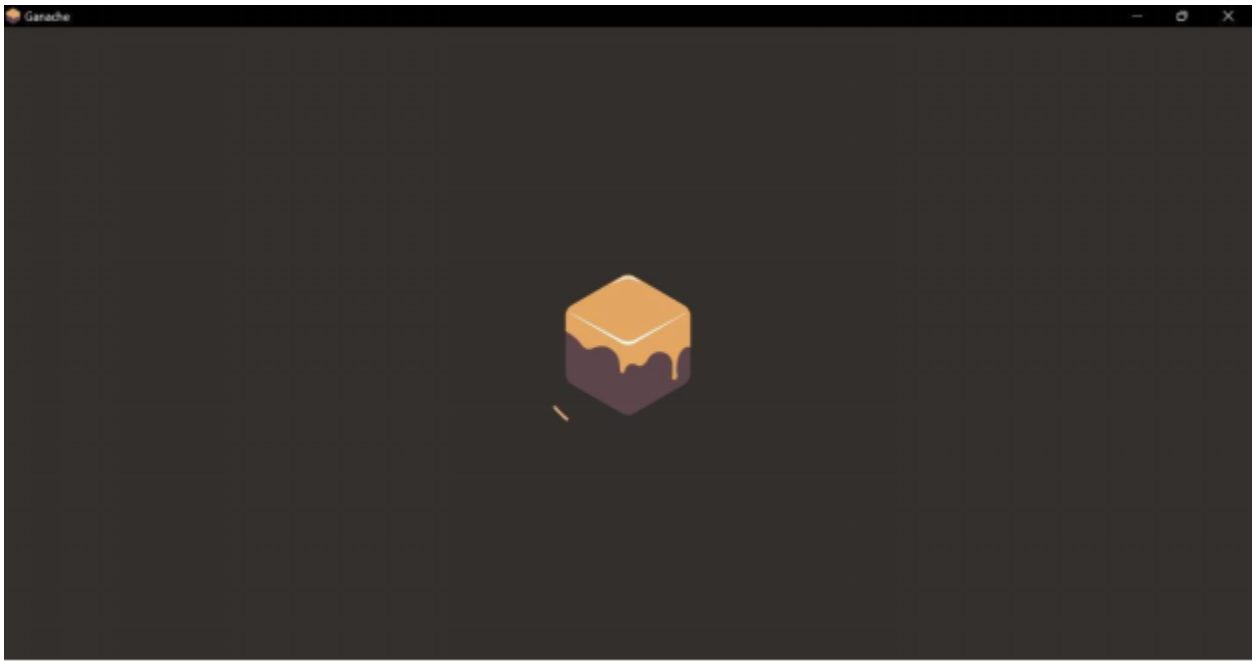
THEORY:

GANACHE :

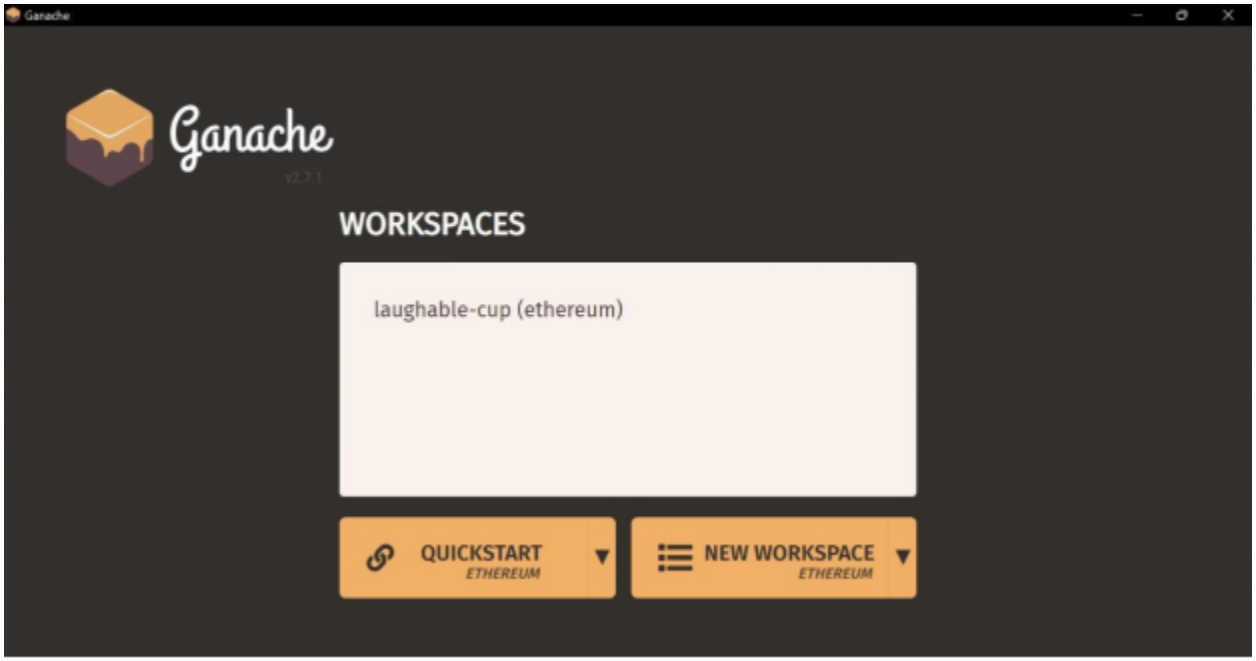
Ganache is a personal blockchain for rapid Ethereum and Corda distributed application development. Ganache is used for setting up a personal Ethereum Blockchain for testing your Solidity contracts. Ethereum Ganache is a local inmemory blockchain designed for development and testing. It simulates the features of a real Ethereum network, including the availability of a number of accounts funded with test Ether.

INSTALLATION:

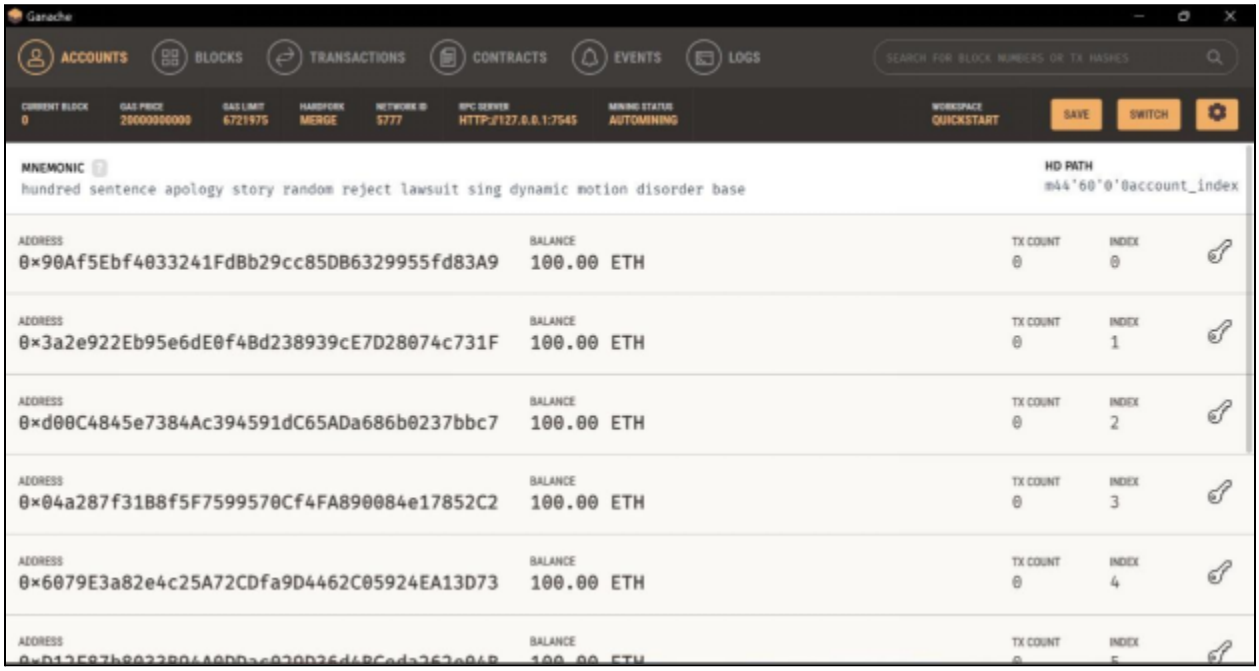
1. Download ganache for Windows from https://www.trufflesuite.com/ganache.



2. After installing , open ganache and select QuickStart.



3. Ganache will create 10 default accounts as follows



METAMASK :

MetaMask is a software cryptocurrency wallet used to interact with the Ethereum blockchain. It allows users to access their Ethereum wallet through a browser extension or mobile app, which can then be used to interact with decentralized applications. MetaMask allows users to store and manage account keys, broadcast transactions, send and receive Ethereum-based cryptocurrencies and tokens, and securely connect to decentralized applications through a compatible web browser or the mobile app's built-in browser.

REMIX IDE :

Remix IDE is an open-source web and desktop application. It fosters a fast development cycle and has a rich set of plugins with intuitive GUIs. Remix is used for the entire journey of contract development as well as being a playground for learning and teaching Ethereum.

Remix IDE is part of the Remix Project which is a platform for development tools that use a plugin architecture. It encompasses sub-projects including Remix Plugin Engine, Remix Libs, and of course Remix-IDE. Remix IDE is a powerful opensource tool that helps you write Solidity contracts straight from the browser. It is written in JavaScript and supports both usage in the browser, in the browser but run locally and in a desktop version.

Remix IDE has modules for testing, debugging and deploying of smart contracts and much more.

ELECTION SMART CONTRACT :

A "smart contract" is simply a program that runs on the Ethereum blockchain. It's a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain.

Smart contracts are a type of Ethereum account. This means they have a balance and they can send transactions over the network. However, they're not controlled by a user, instead they are deployed to the network and run as programmed. User accounts can then interact with a smart contract by submitting transactions that execute a function defined on the smart contract. Smart contracts can define rules, like a regular contract, and automatically enforce them via the code. Smart contracts cannot be deleted by default, and interactions with them are irreversible.

Benefits of smart contracts

- Speed, efficiency and accuracy : Once a condition is met, the contract is executed immediately. Because smart contracts are digital and automated,

there’s no paperwork to process and no time spent reconciling errors that often result from manually filling in documents.

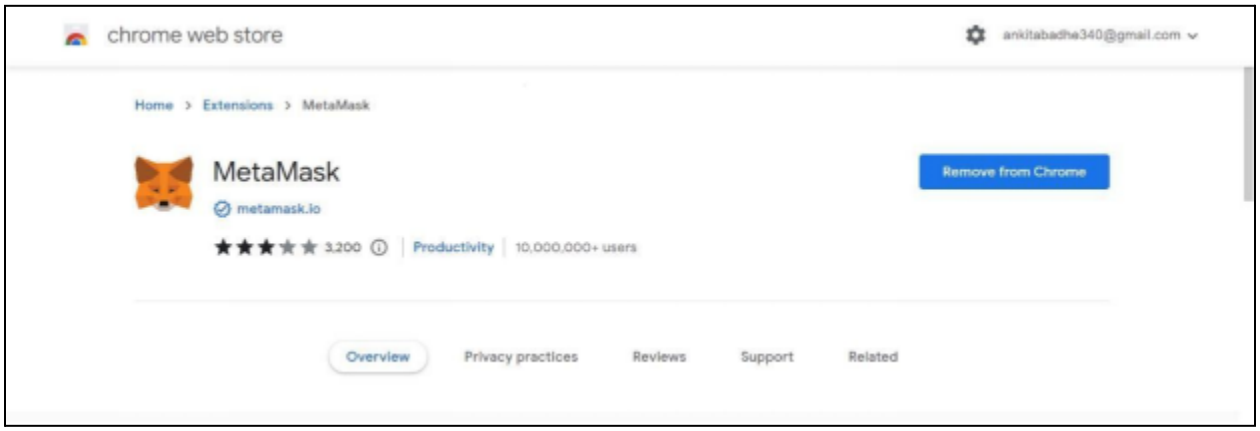
- Trust and transparency : Because there's no third party involved, and because encrypted records of transactions are shared across participants, there's no need to question whether information has been altered for personal benefit.

- Security : Blockchain transaction records are encrypted, which makes them very hard to hack. Moreover, because each record is connected to the previous and subsequent records on a distributed ledger, hackers would have to alter the entire chain to change a single record.

- Savings : Smart contracts remove the need for intermediaries to handle transactions and, by extension, their associated time delays and fees

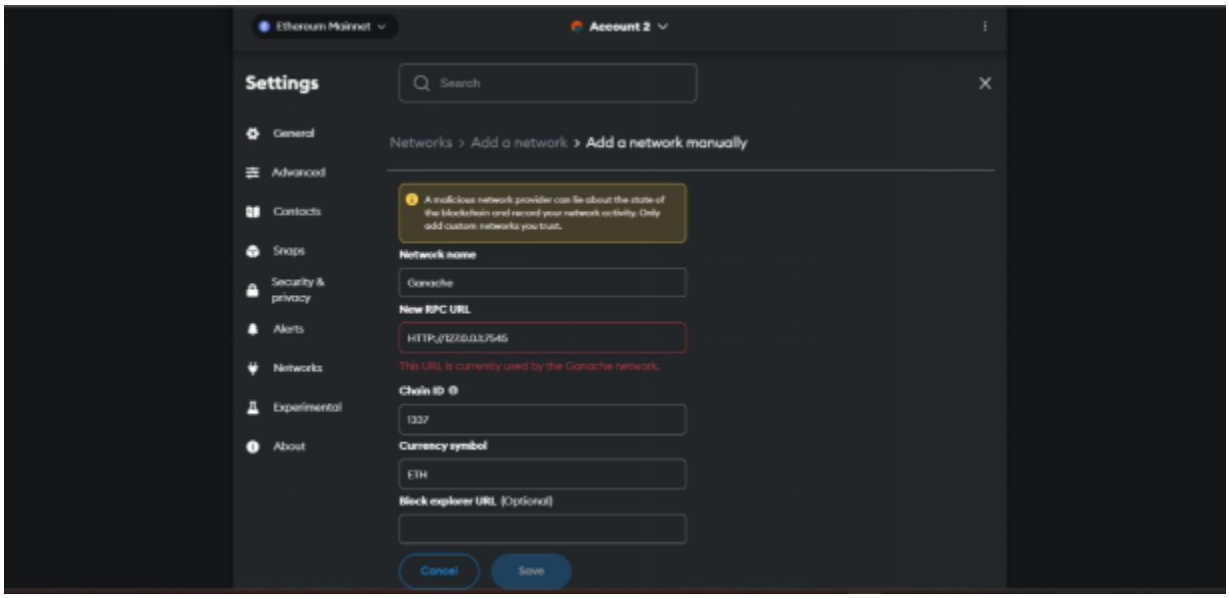
EXECUTION :

1. Add extension of MetaMask in Google Chrome or any other browser and it will be ready to use.

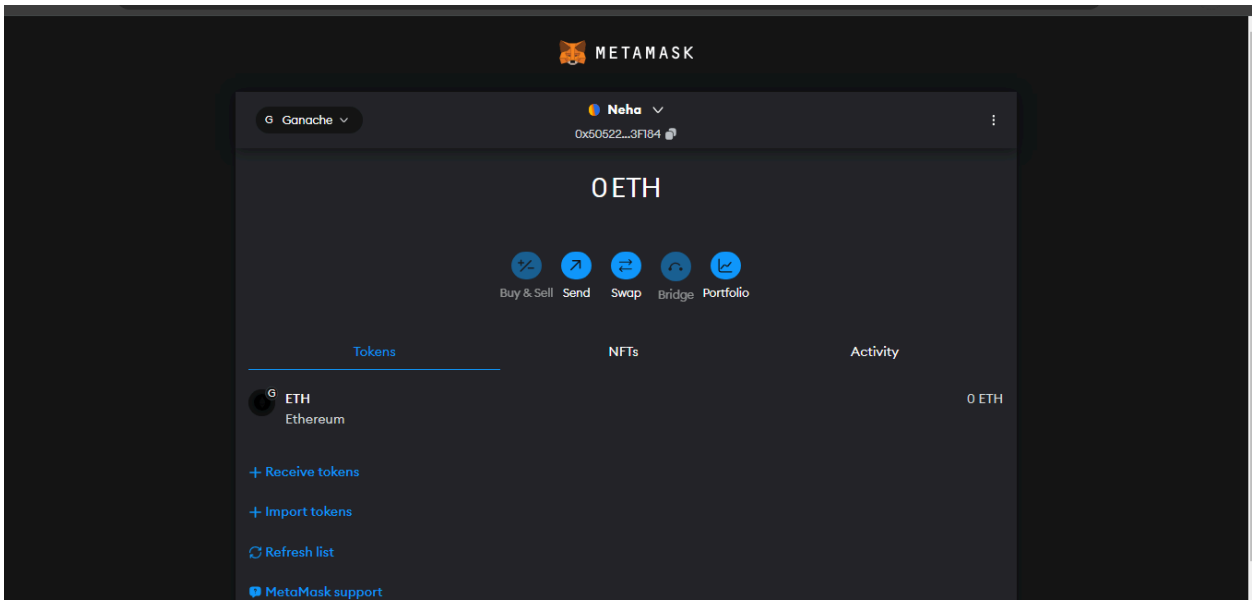


2. In MetaMask , go to settings → Custom RPC → Enter Network name as “Ganache” , copy paste network URL from Ganache and put Chain ID as “1337”

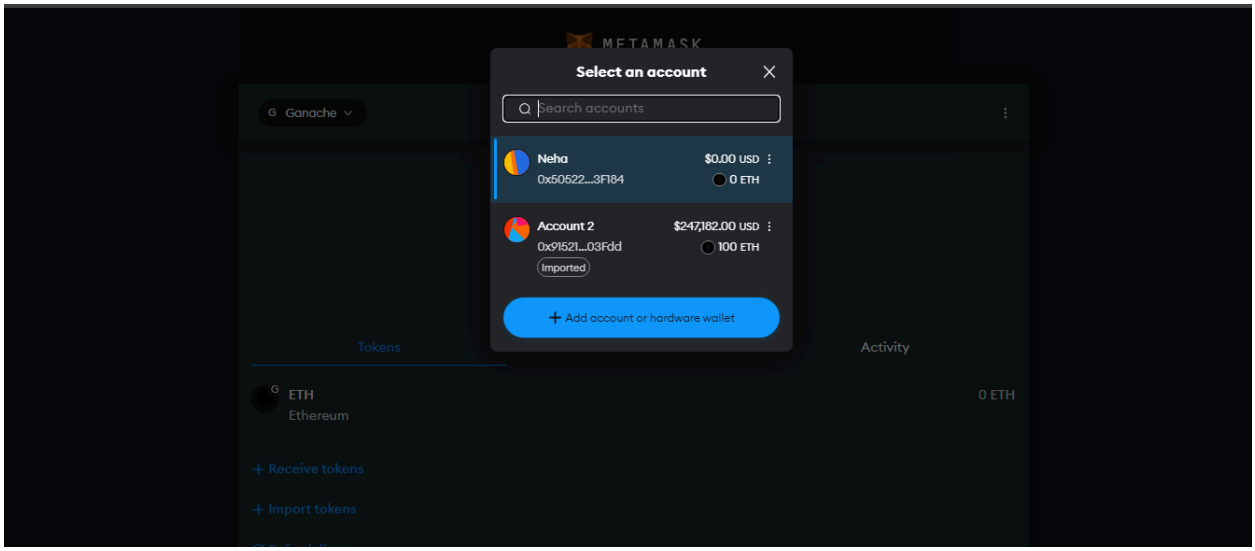
Save it and make your your network on the top-left shows you network: **Ganache**



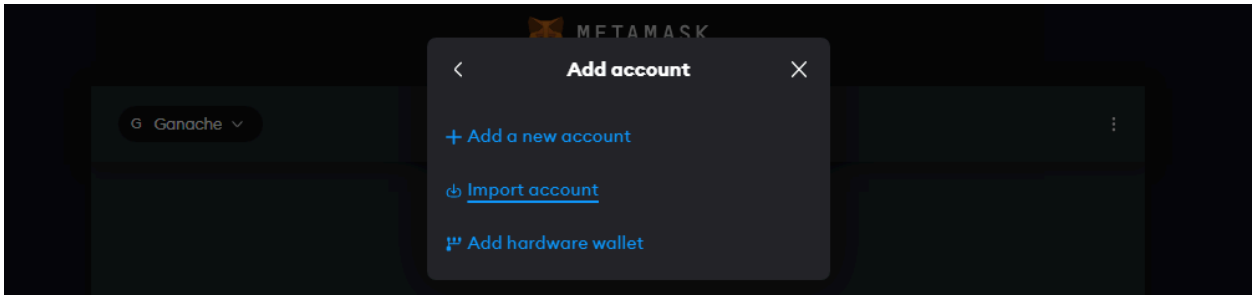
2. After opening MetaMask it will look like this



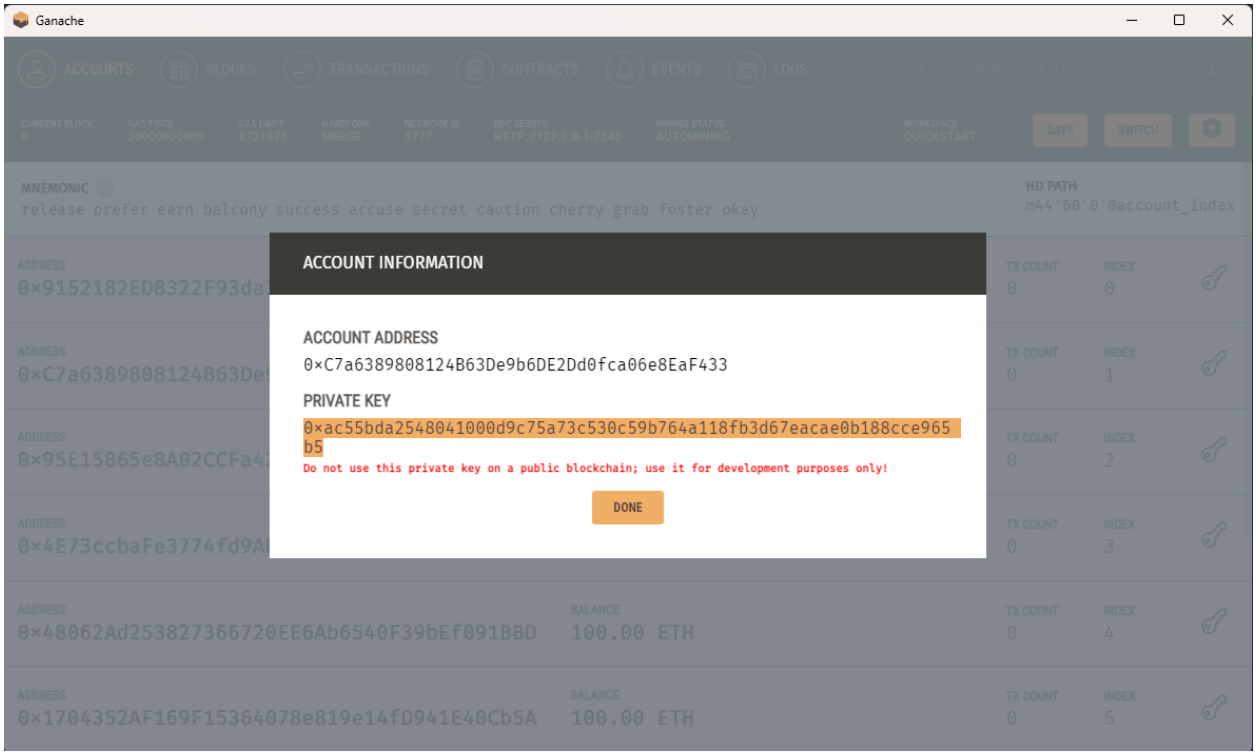
3. Now Add an account



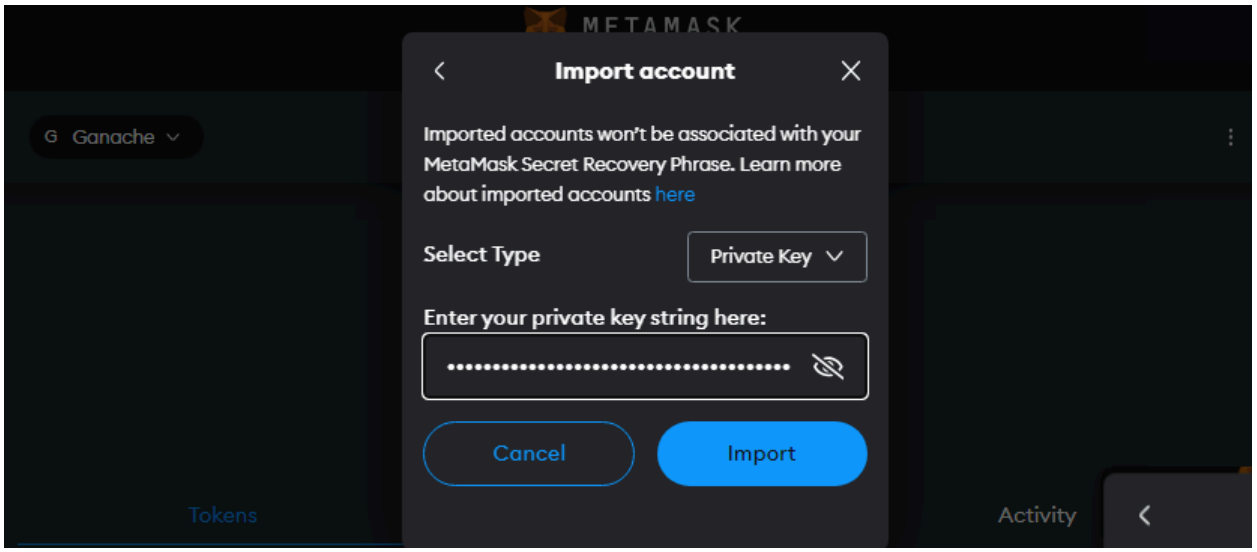
4. Import account



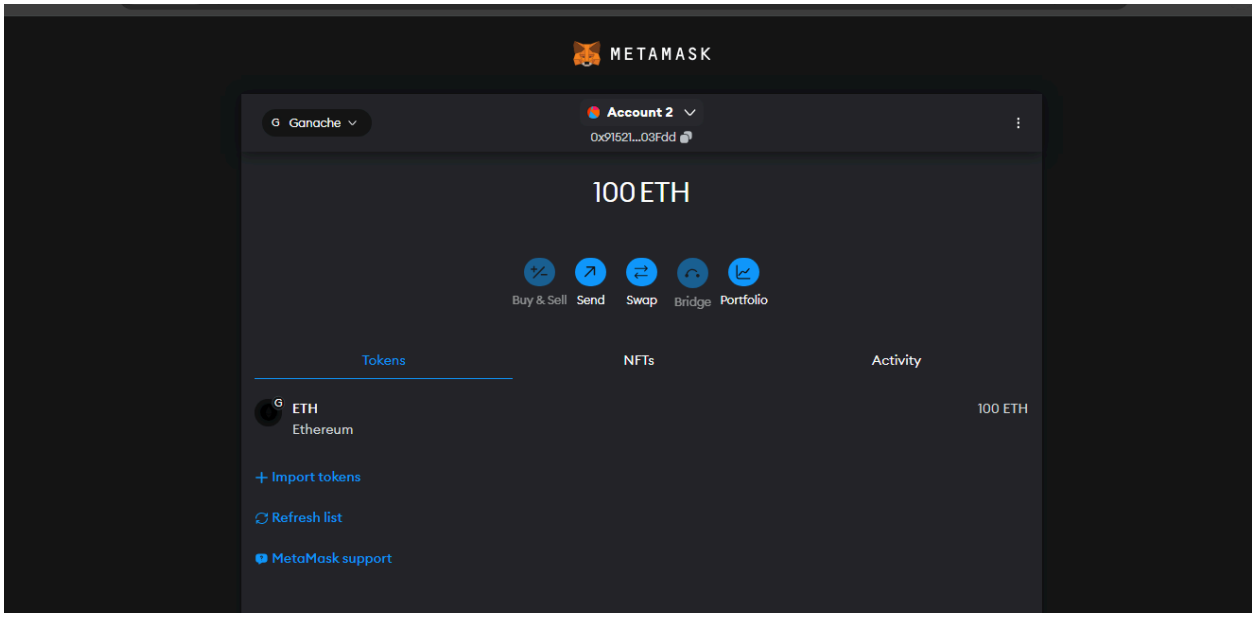
5. Now open Ganache from your system and click on any key and copy the private key



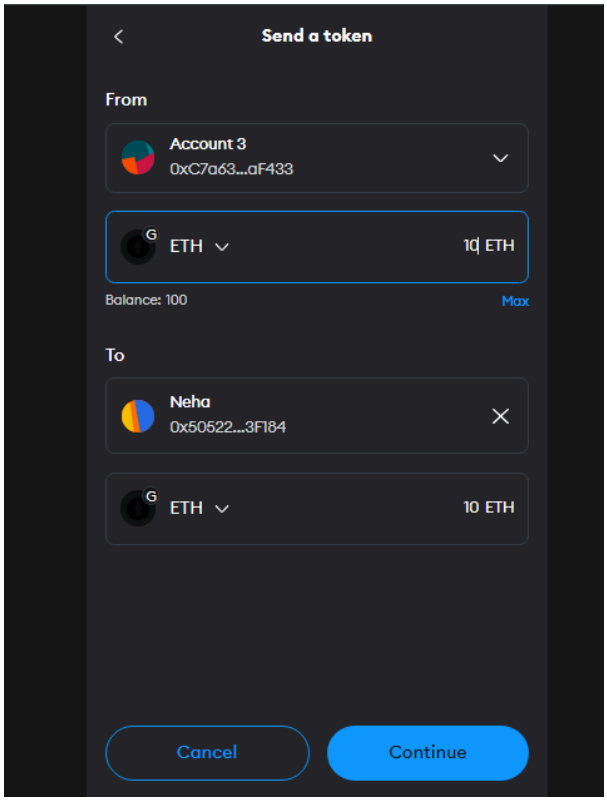
6. Now while importing the new account, enter the private key and Import



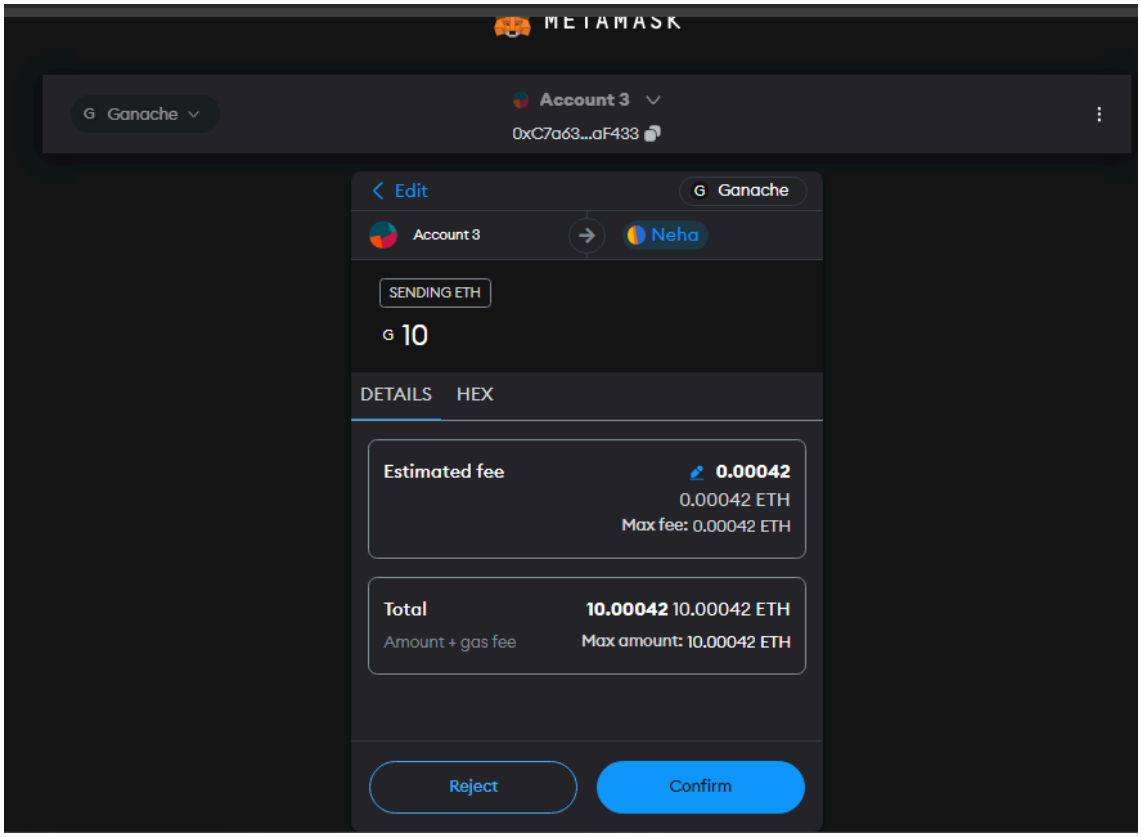
7. Open MetaMask and the screen will look like this



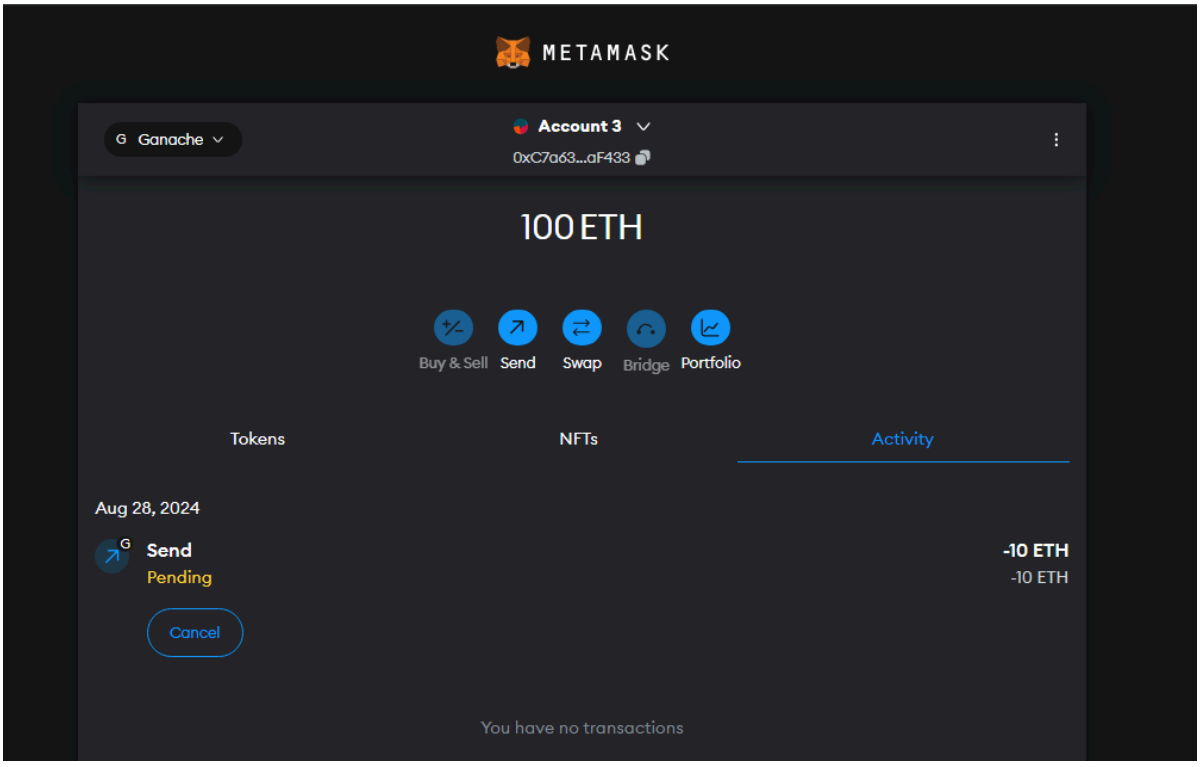
8. Try sending money to other created Account



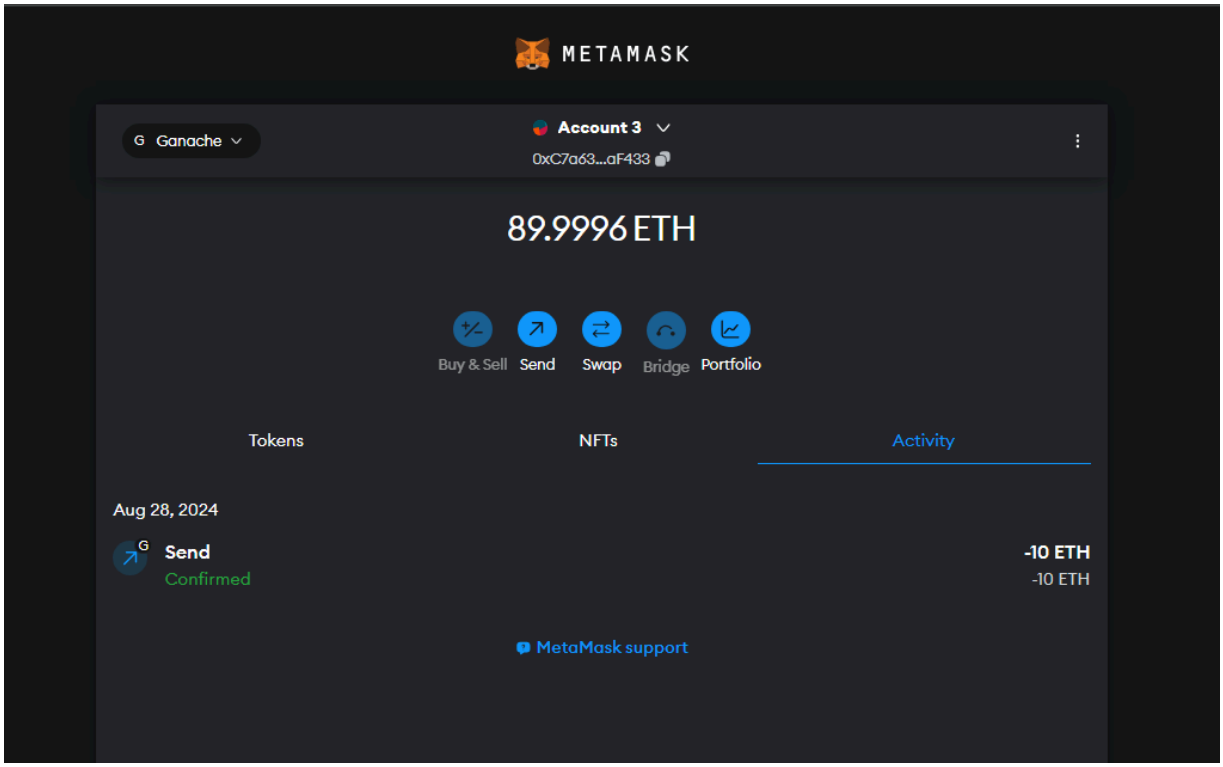
9. Some amount will be deducted → Confirm

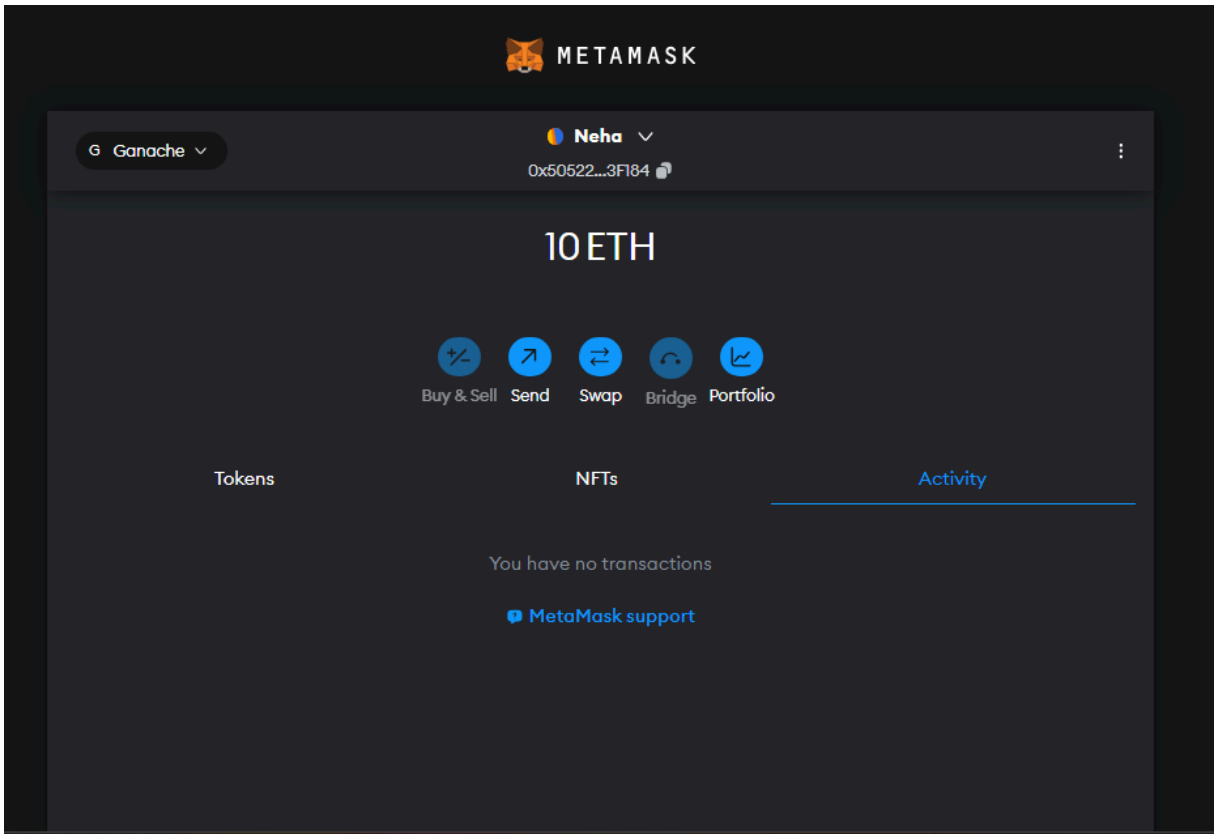


10. It will show the pending status

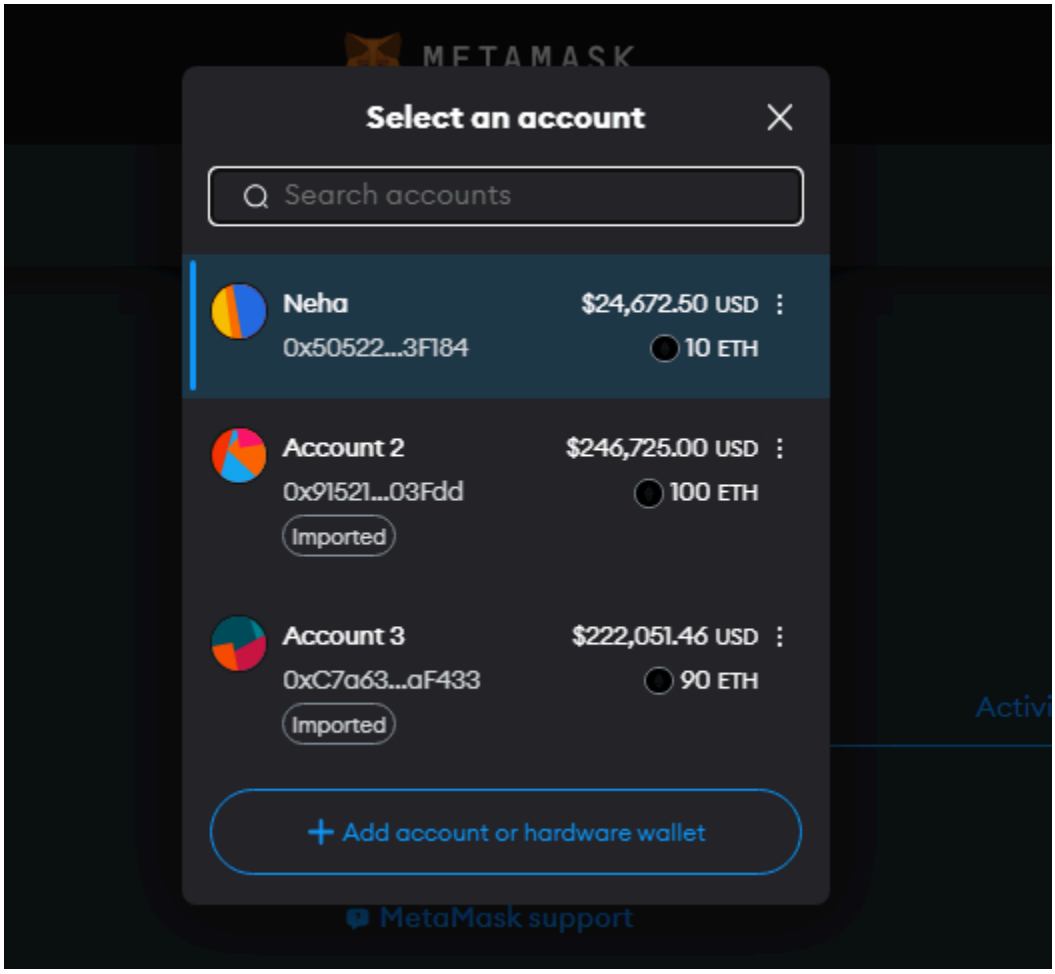


11. Once confirmed the amount will be transferred to the account



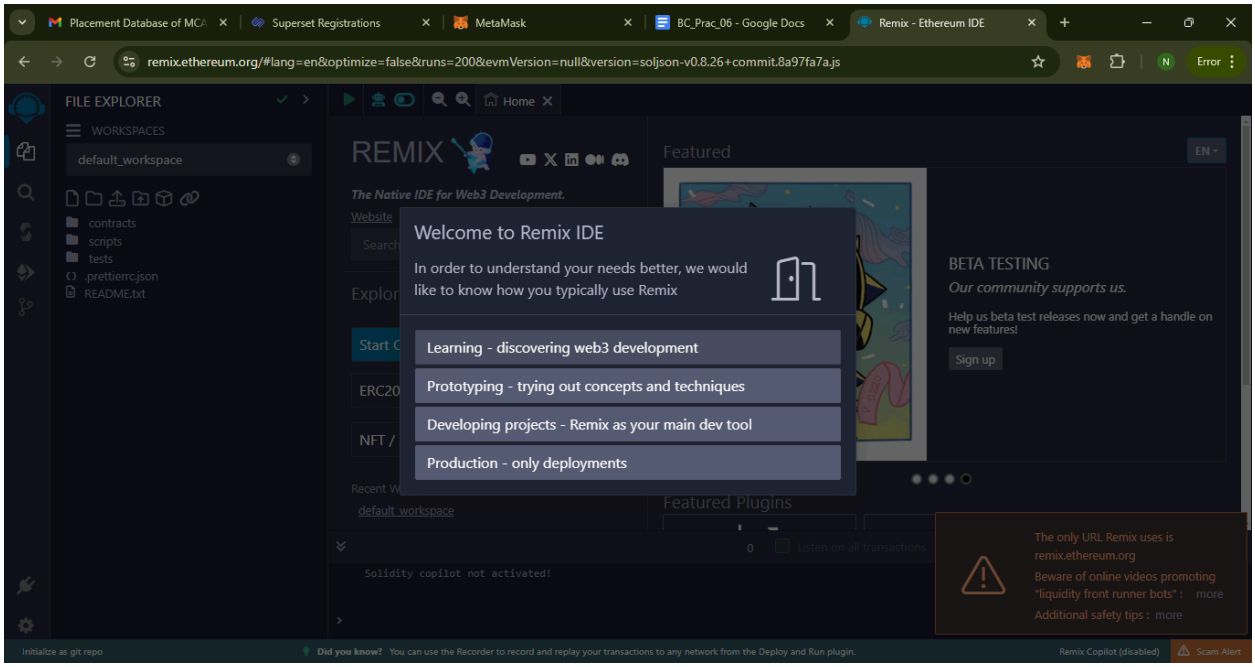


12. Here’s the overview of the transcations

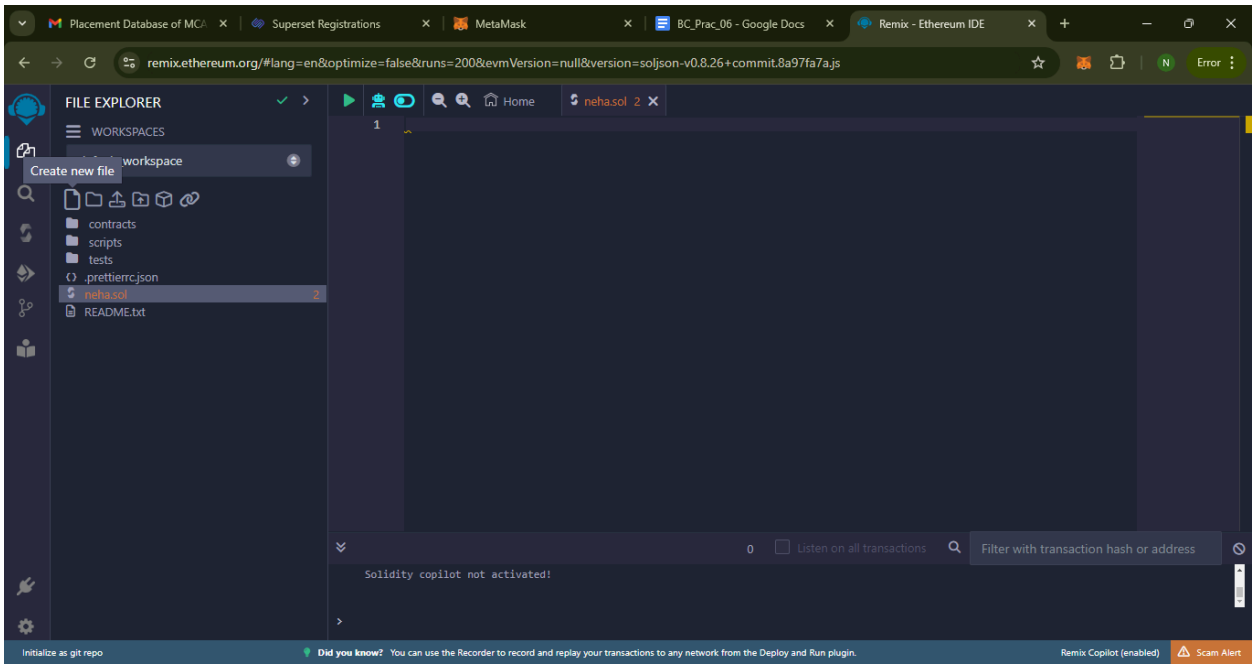


Now open **"Remix IDE"** from https://remix.ethereum.org/#optimize=false&runs=200&evmVersion=null , Create one folder "MyContract" , create one file inside that folder as "election.sol" and add code into it .

1. Select → Learning



2. Create a new file and name it with extension **.sol**



3. Now add a **code** for "**Addition of 2 nos**" (the code will be available online- just **type**: Addition of 2 nos in solidity)

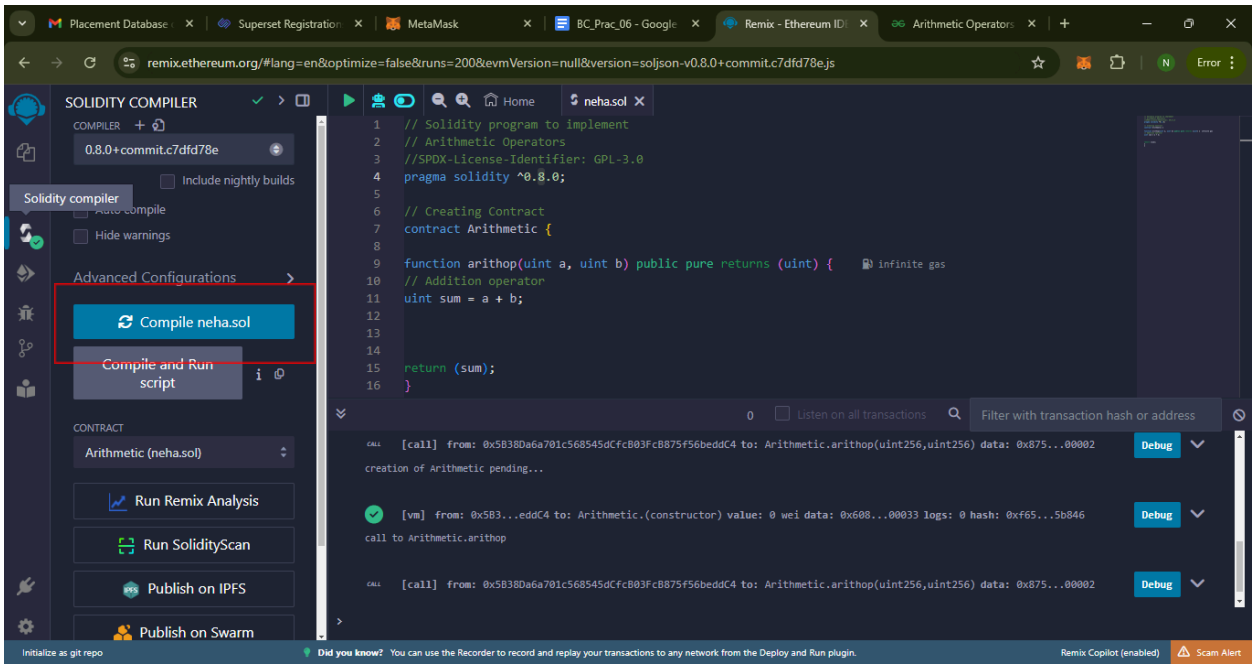
```
-  //SPDX-License-Identifier: GPL-3.0      //TYPE THIS LINE
-  pragma solidity ^0.8.0;
```

CODE:

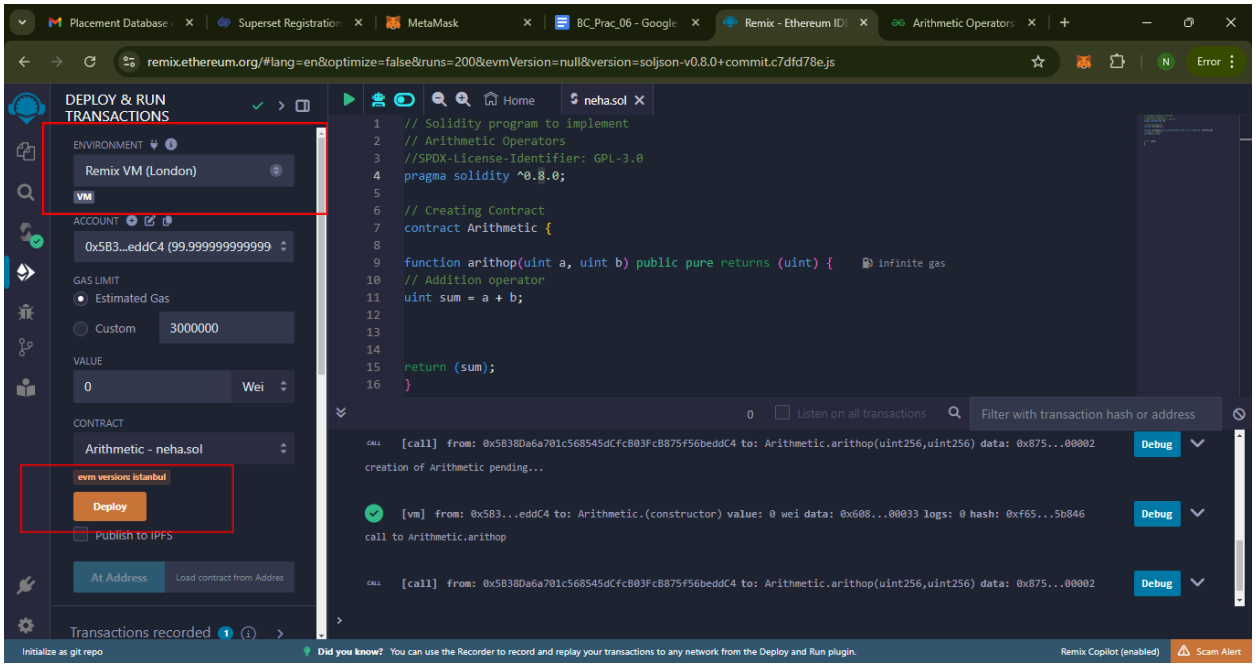
```
// Solidity program to implement
// Arithmetic Operators
//SPDX-License-Identifier: GPL-3.0
pragma solidity ^0.8.0;

// Creating Contract
contract Arithmetic {

function arithop(uint a, uint b) public pure returns (uint) {
// Addition operator
uint sum = a + b;
```

```
      // Solidity program to implement
// Arithmetic Operators
//SPDX-License-Identifier: GPL-3.0      //TYPE THIS LINE
pragma solidity ^0.8.0;

// Creating Contract
contract Arithmetic {

function arithop(uint a, uint b) public pure returns (uint) {
// Addition operator
uint sum = a + b;
return (sum);
}
}
```

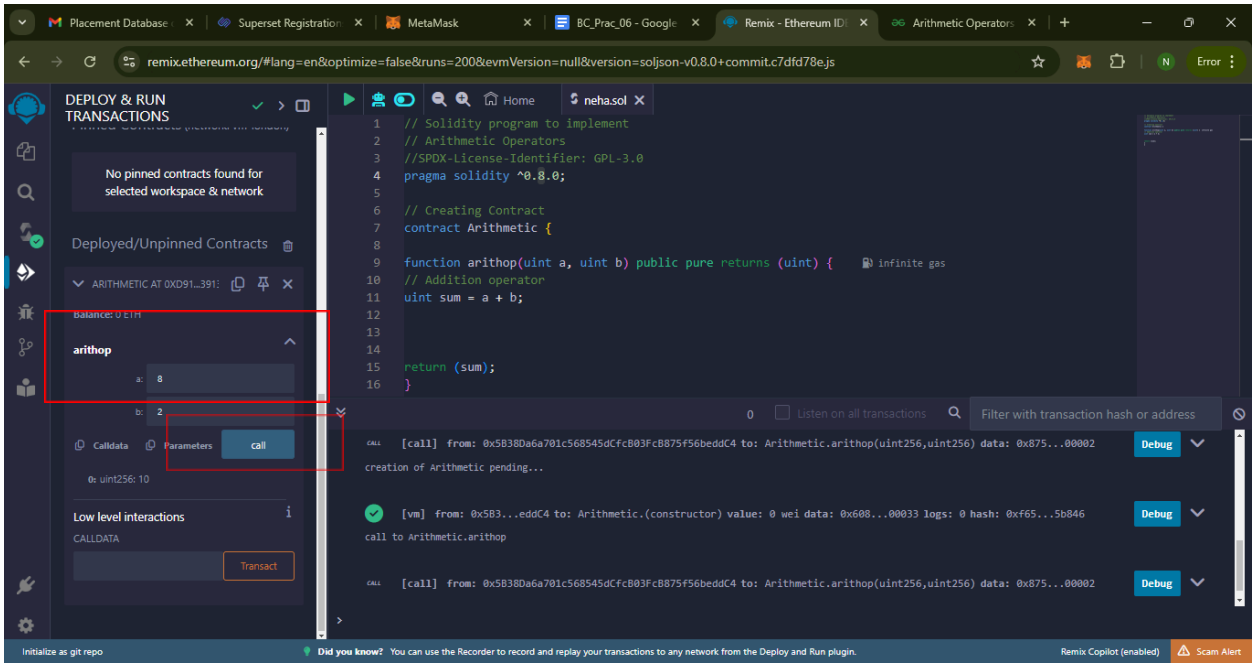
Here change the compiler to 0.8.0 and change the same in the code and click on **Complie neha.sol**



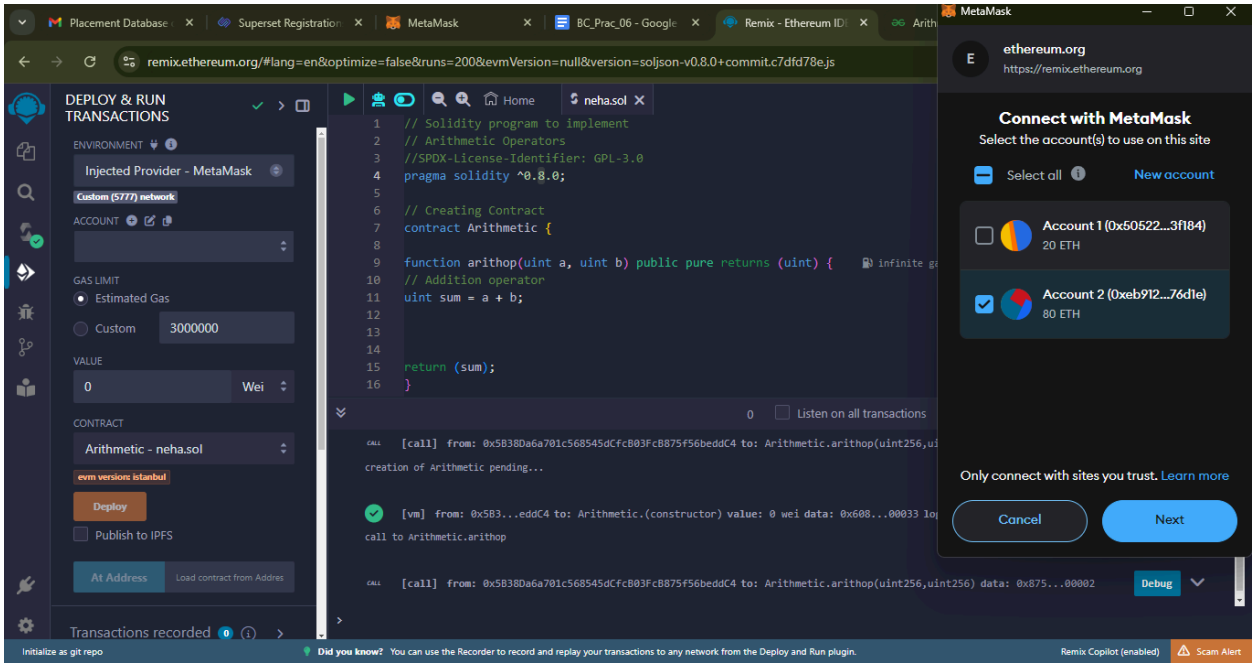
Change the **Environment** to **London**
Then click on Deploy



Click on down arrow, enter the nos and click call



Now click on Next, Now in remix ide click on deploy
Then click on metamask extension and click on confirm
Again input values and get output
When you get the output, some amount should be deducted in metamask



CONCLUSION:

To effectively set up a development environment for blockchain applications, start by installing Ganache for a personal blockchain and MetaMask for managing accounts. Follow detailed installation steps to ensure both tools are correctly configured. Utilize Remix IDE for compiling and deploying an election smart contract, leveraging MetaMask's injected web3 environment to interact with the local blockchain.