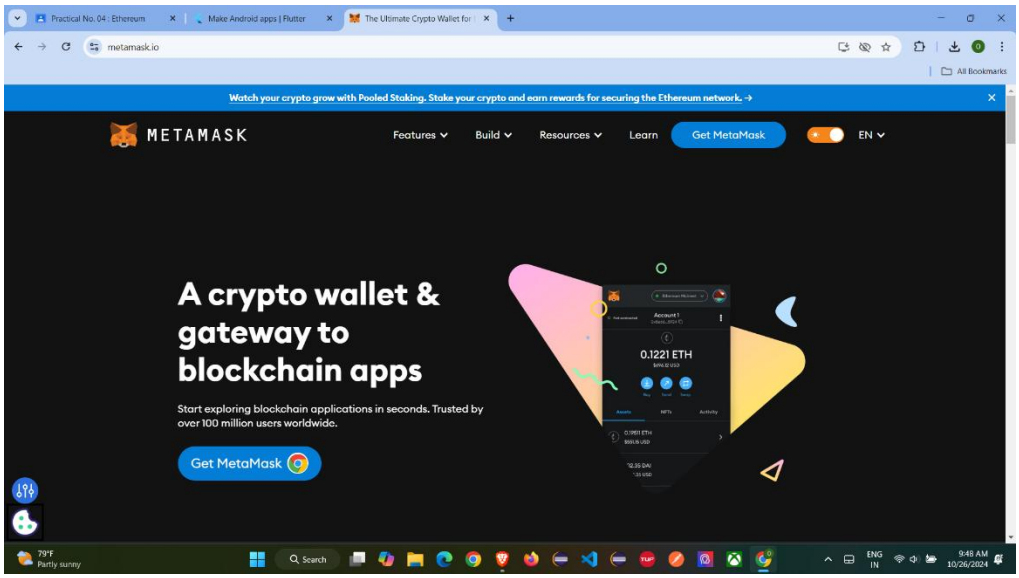
Practical No. 4

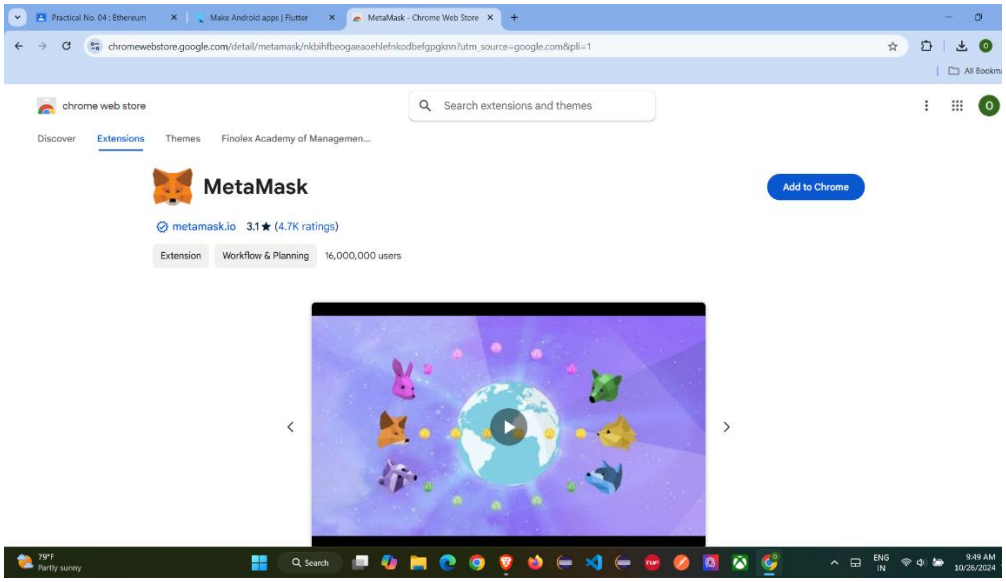
Ethereum

Q.1 Install the metamask in browser. Setup the metamask digital cryptocurrency wallet. Create multiple accounts in metamask and connect with one of the etherum blockchain test network. Perform the task buy ethers and send ethers from one account to another. Take the screenshots of created accounts, account assets and account transactions which showing the details of transaction. (Use following url to get free ether for Sepolia Test Network: https://faucets.chain.link/sepolia)

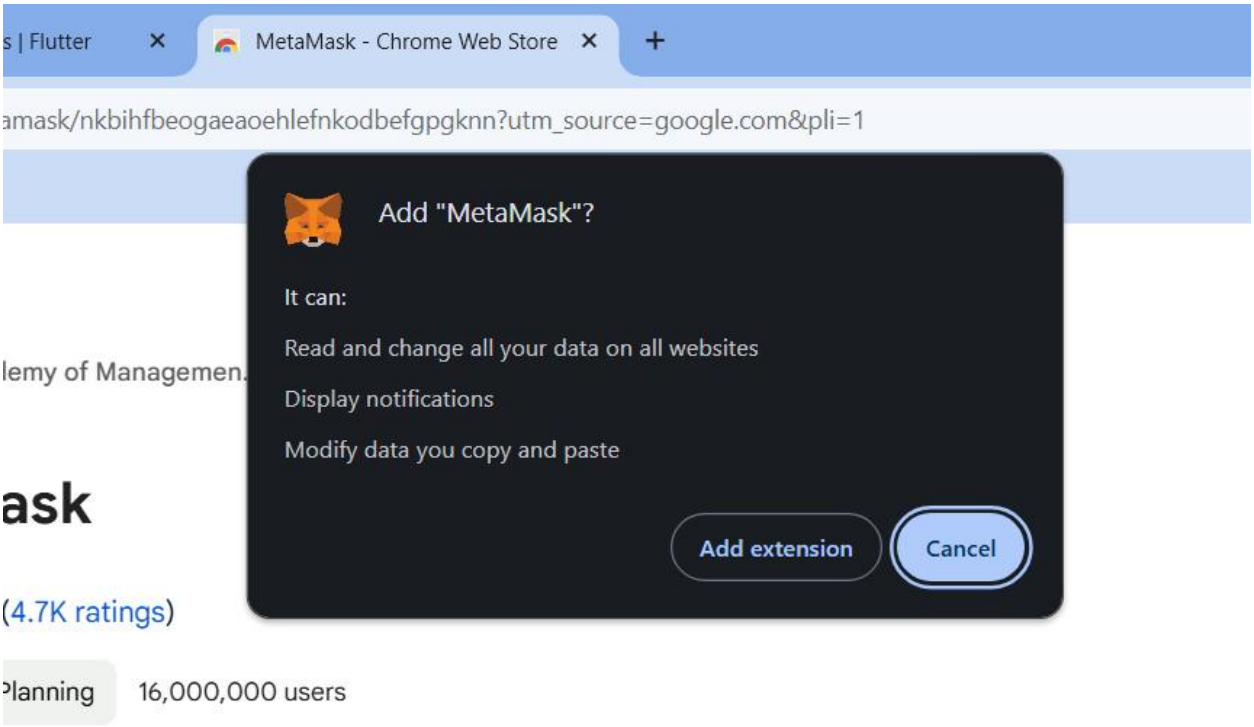
Step 1: Open Browser: metamask.io

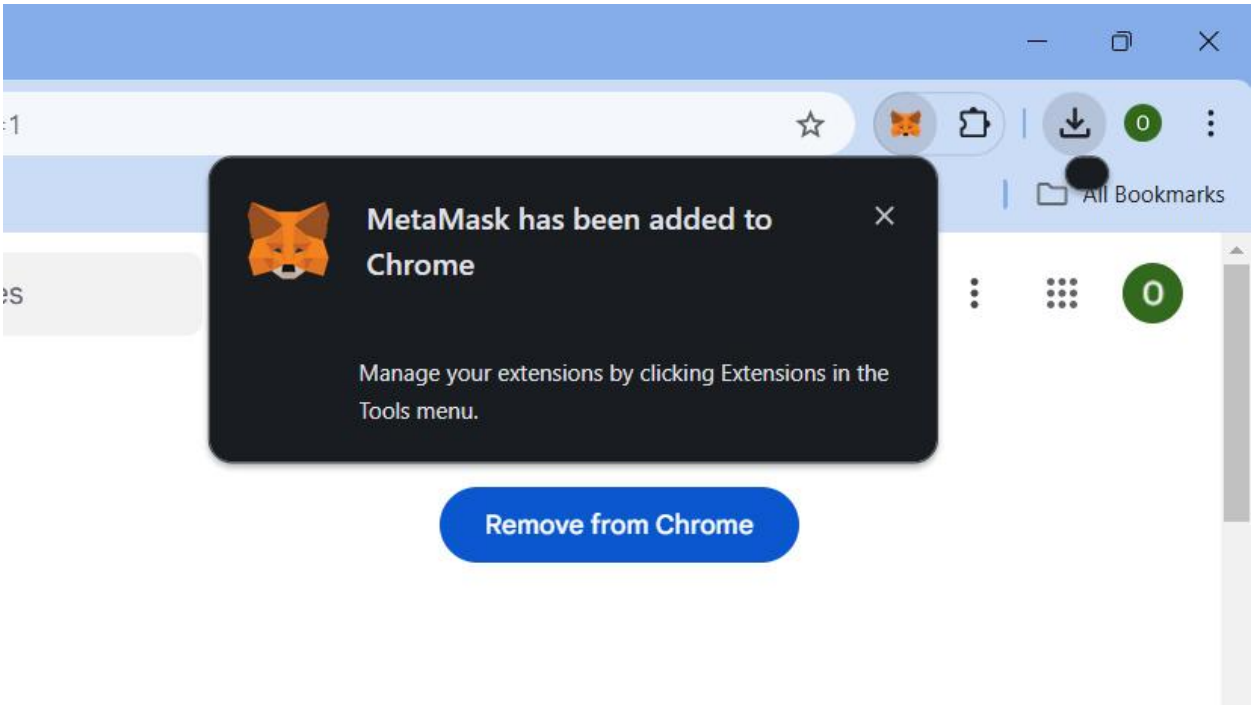


Step 2: Download Chrome Extension

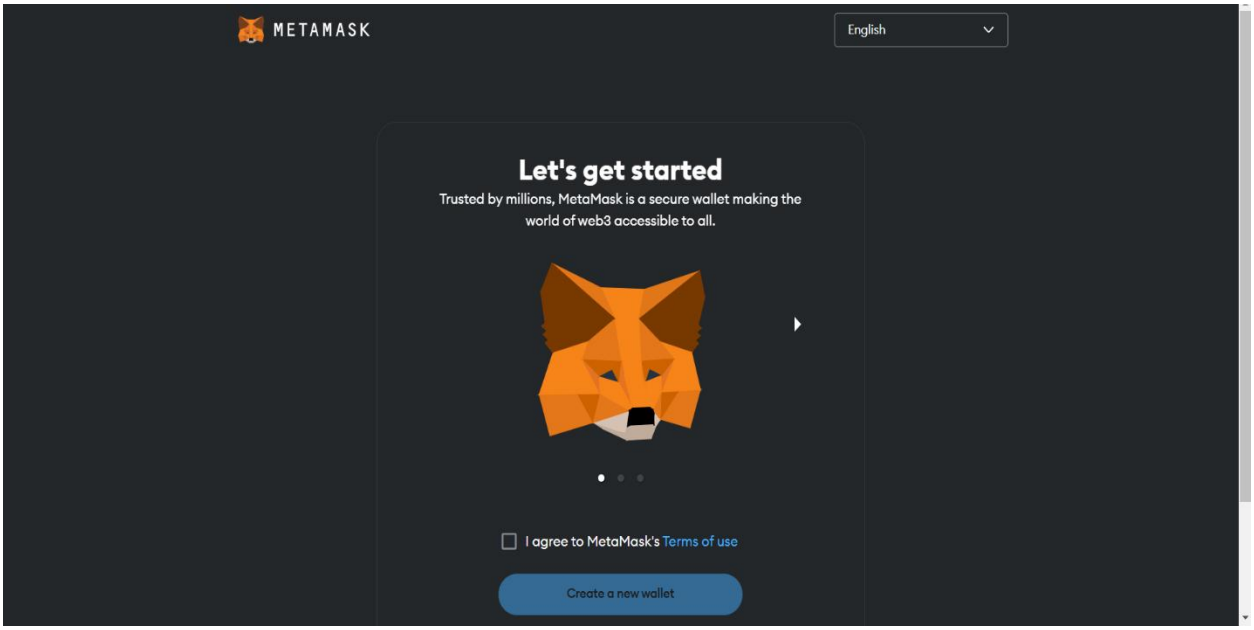


Step 3: Add Extension

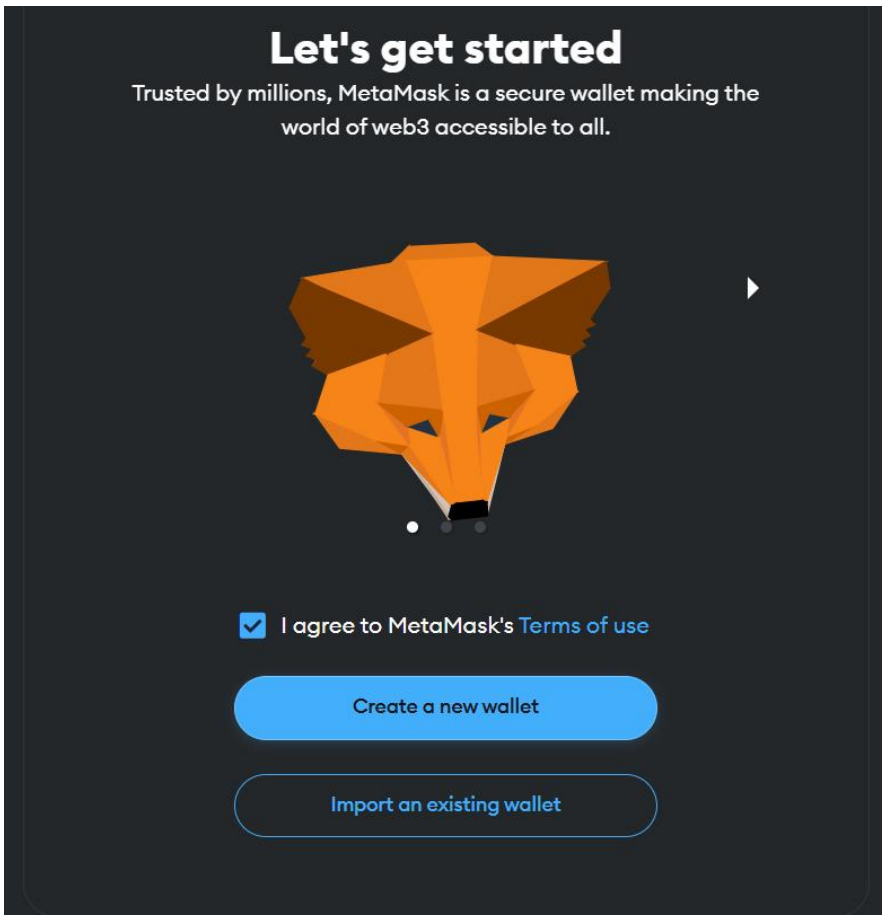




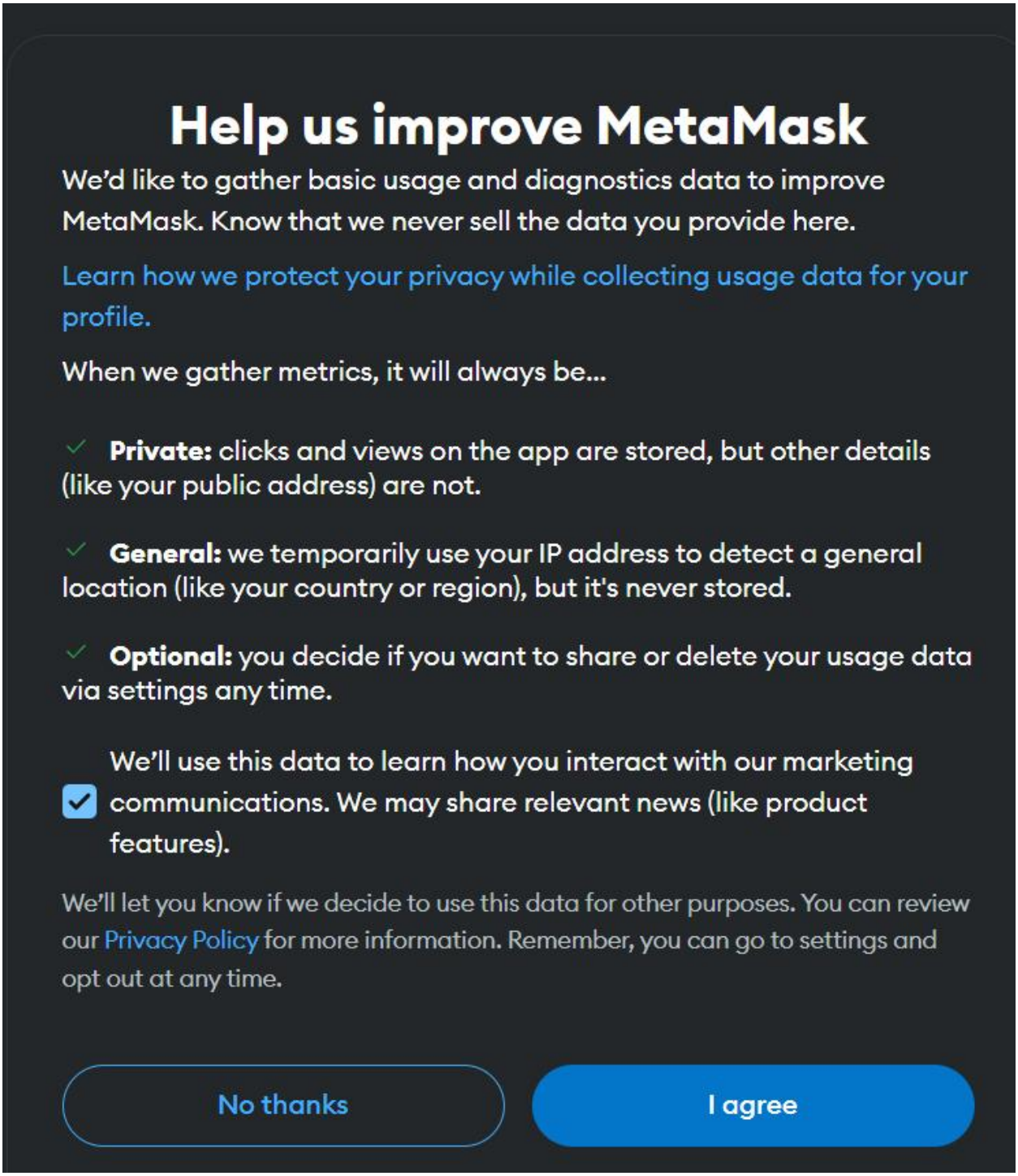
Step 4: Open MetaMask Extension



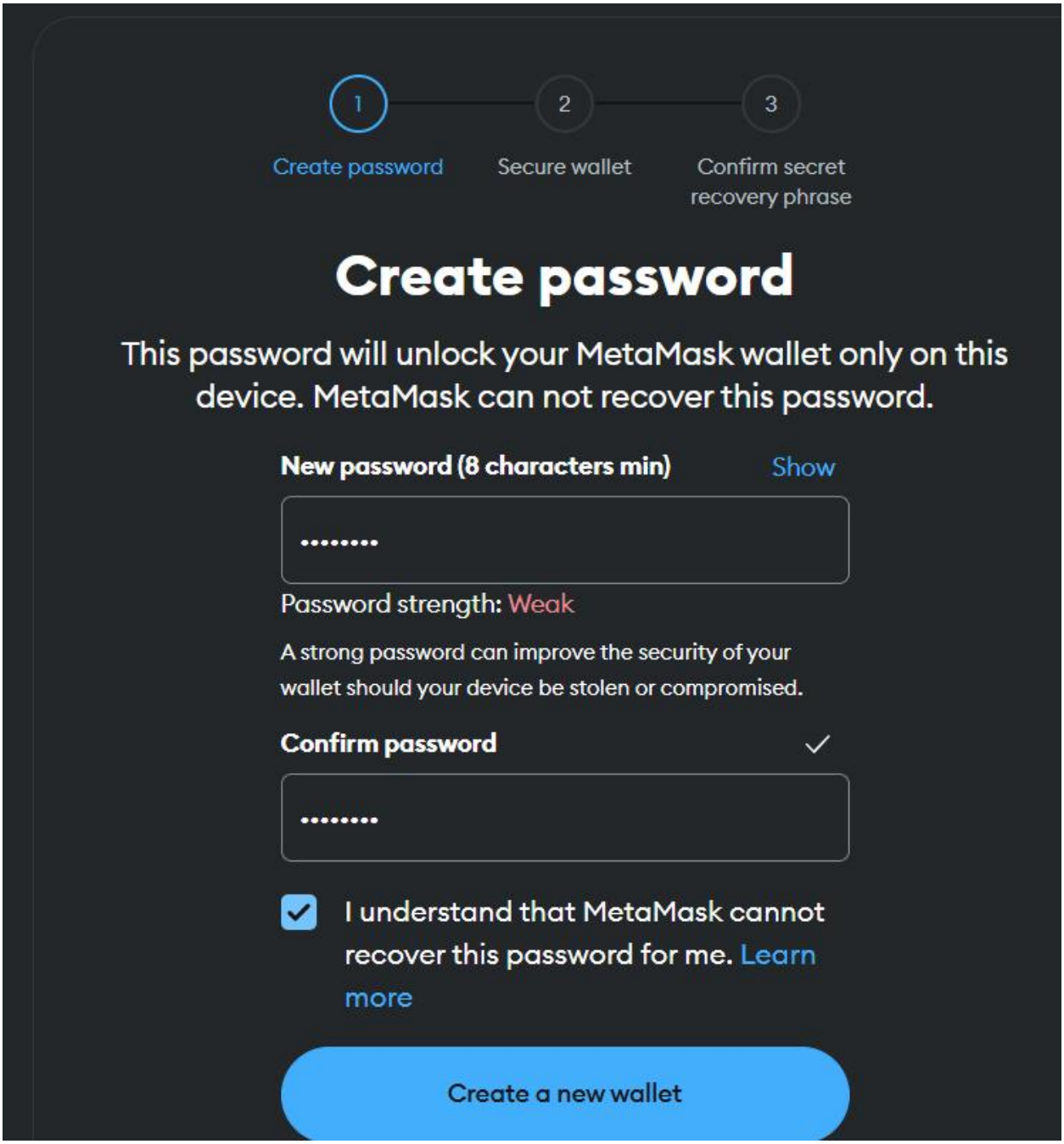
Step 5: Agree Terms and Conditions



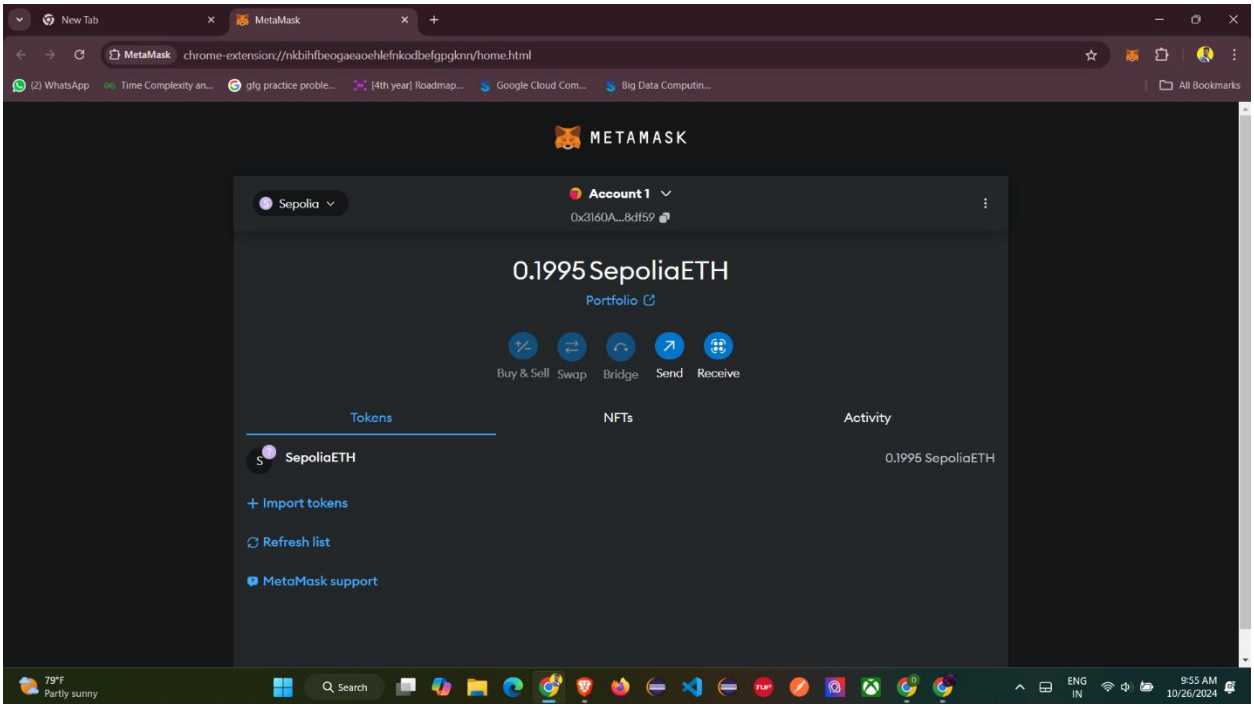
Step 6: Read all conditions and Click on Agree



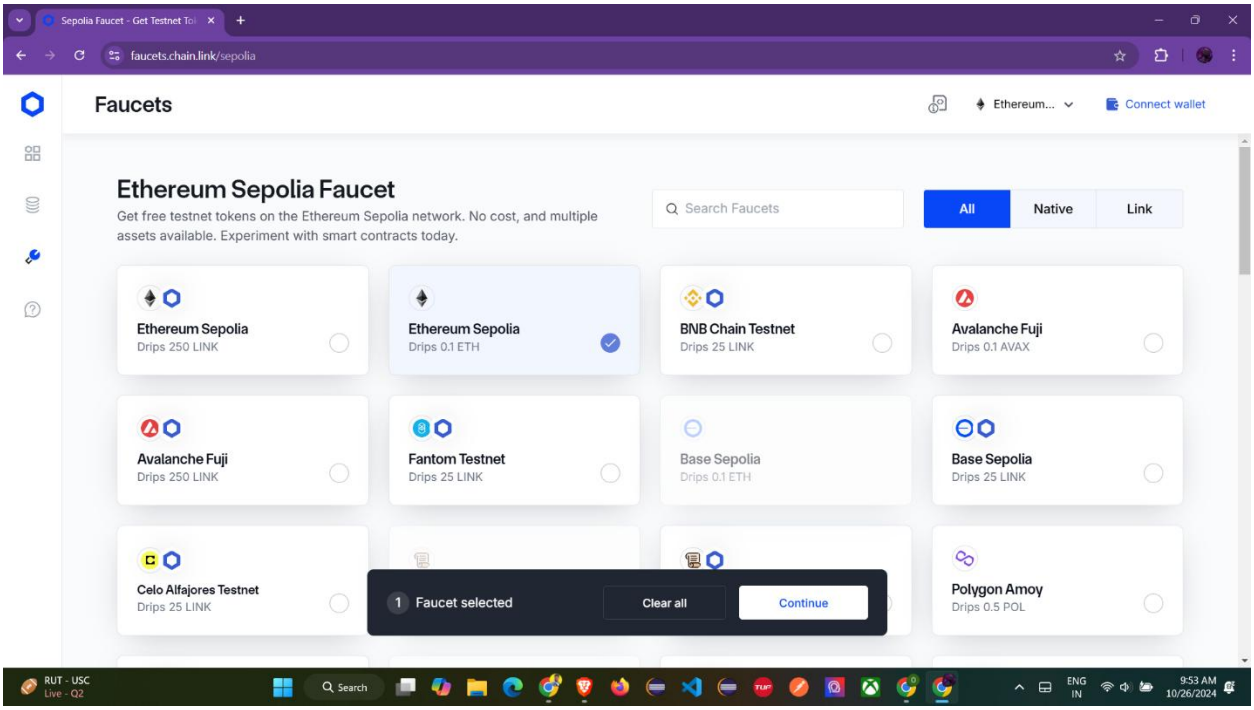
Step 7: Add Password and Confirm Password



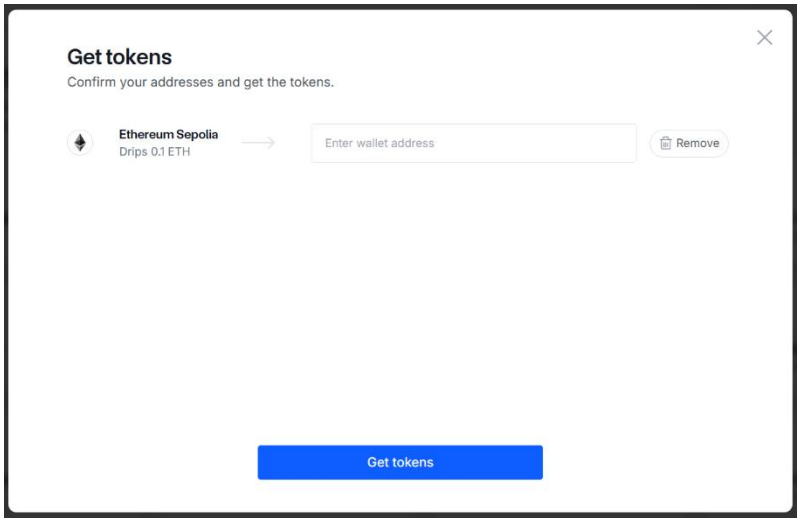
Step 8: Copy public key or address of the Account

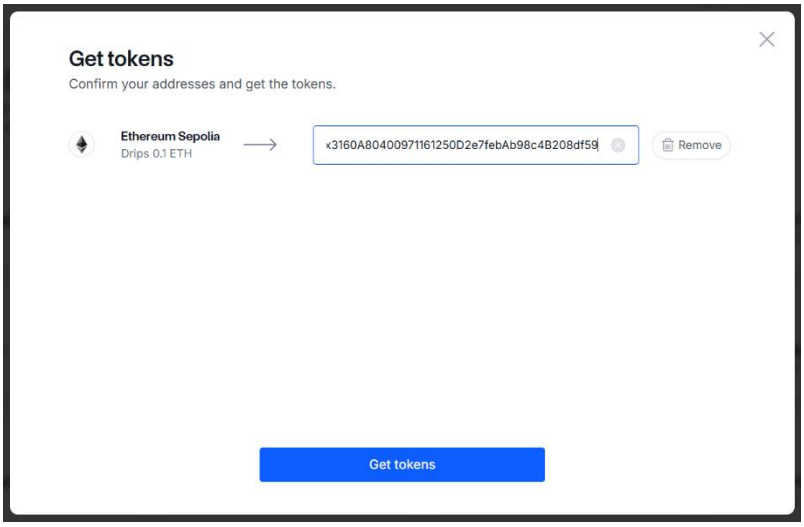


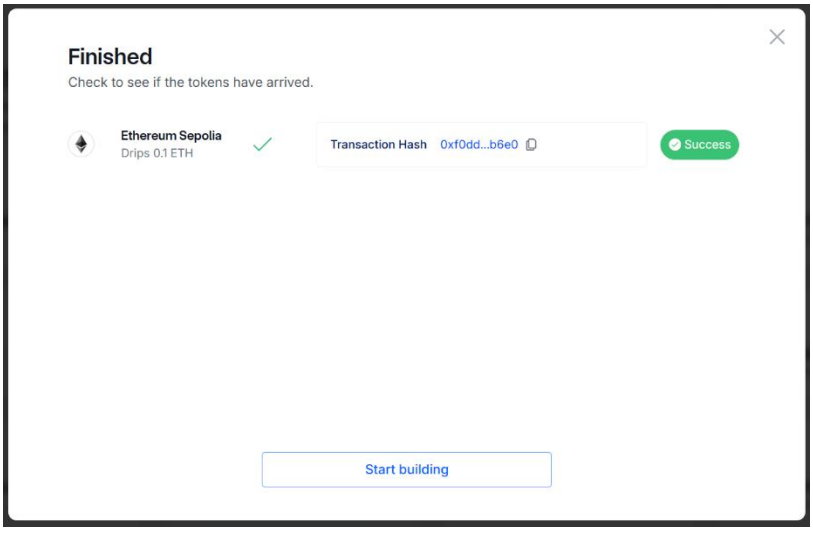
Step 9: Open this URL(https://faucets.chain.link/sepolia)



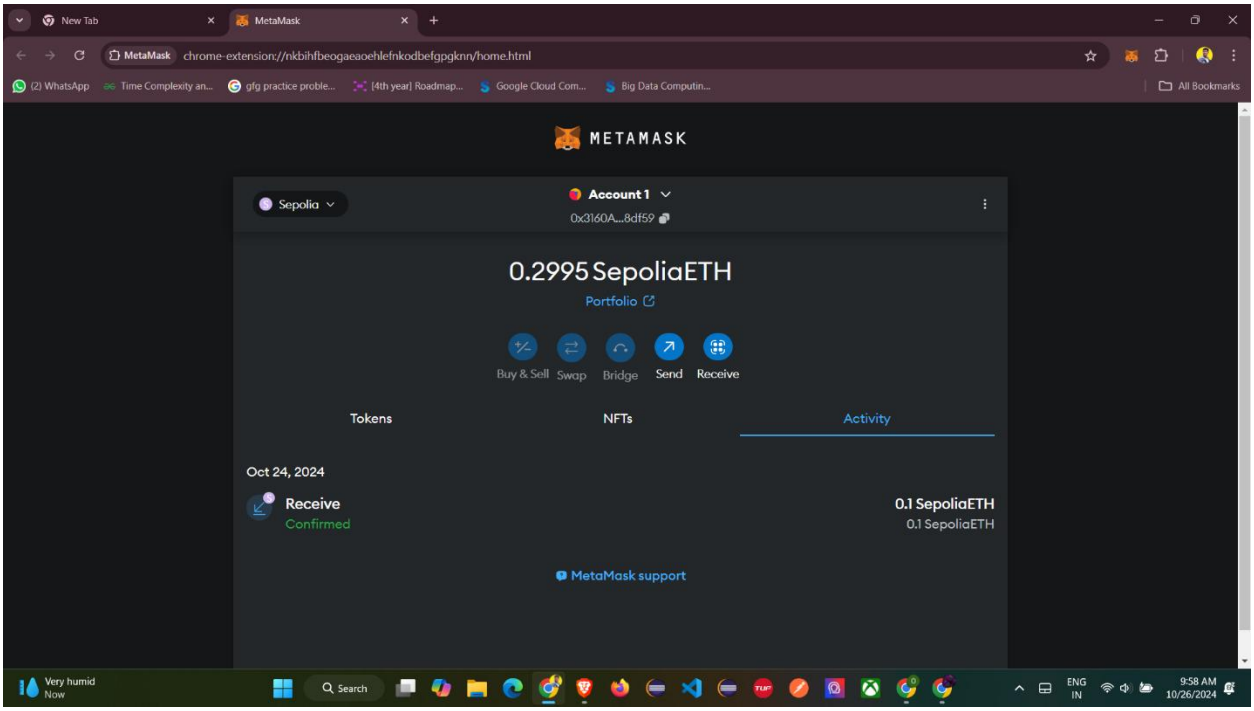
Step 10: Add your Public Key or address of Account



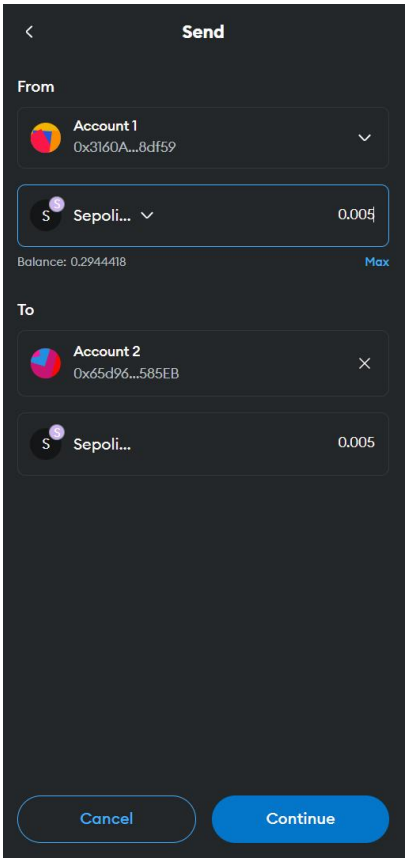


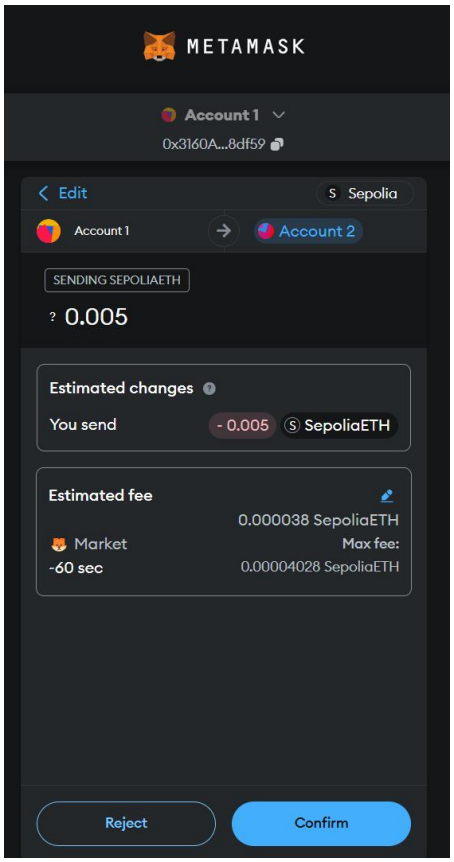


Step 11: Check Activity and the Balance of Account you receive 0.1SepoliaETH

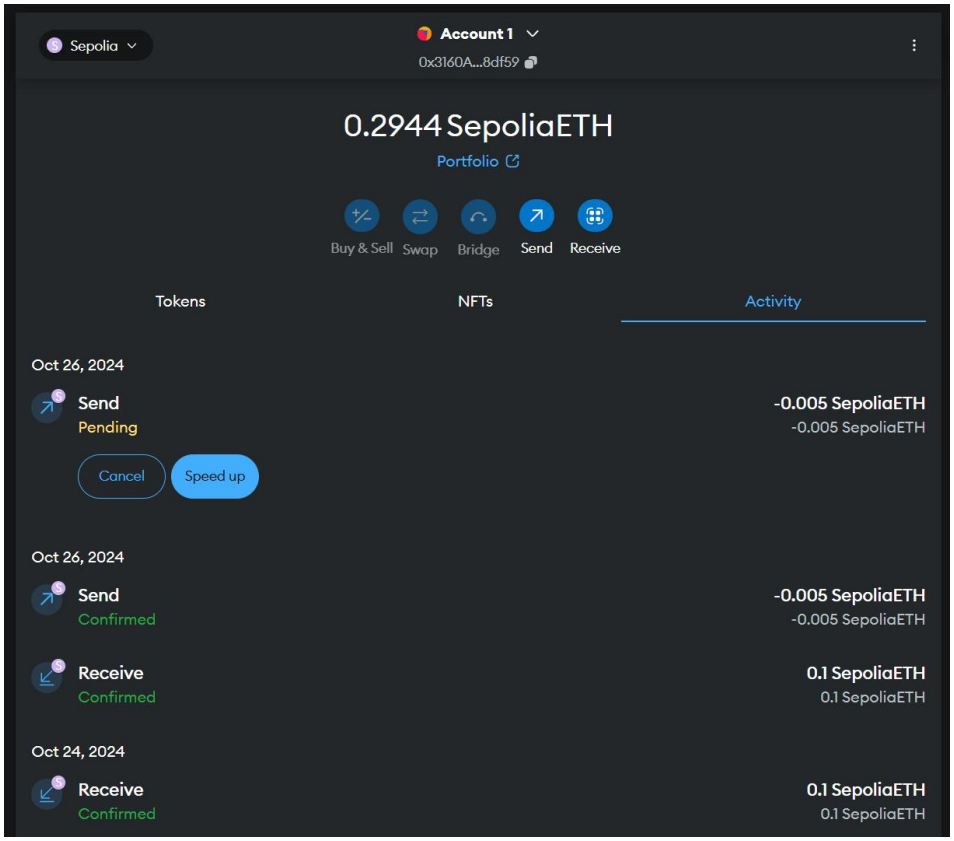


Step 12: Send 0.005 Amount to Another Account

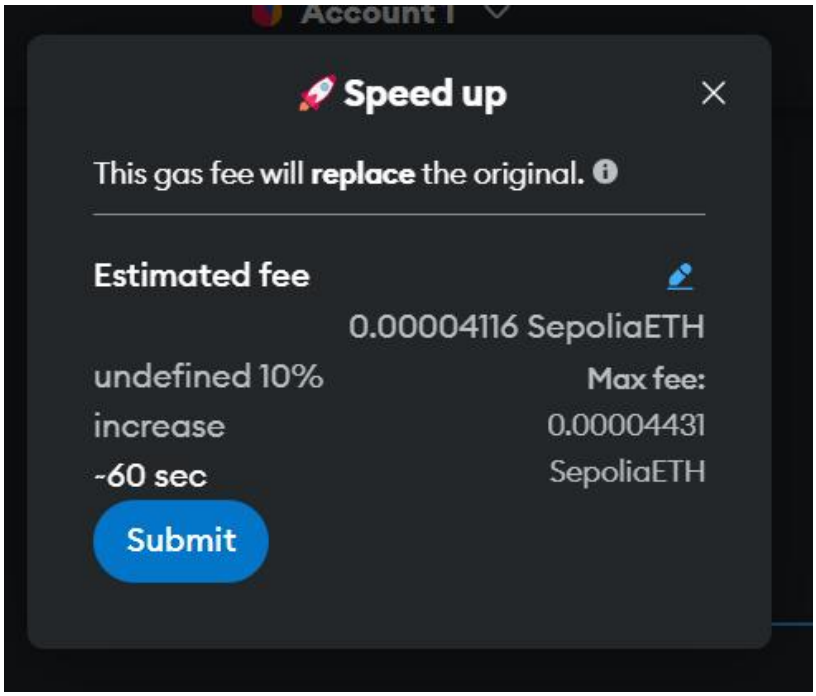




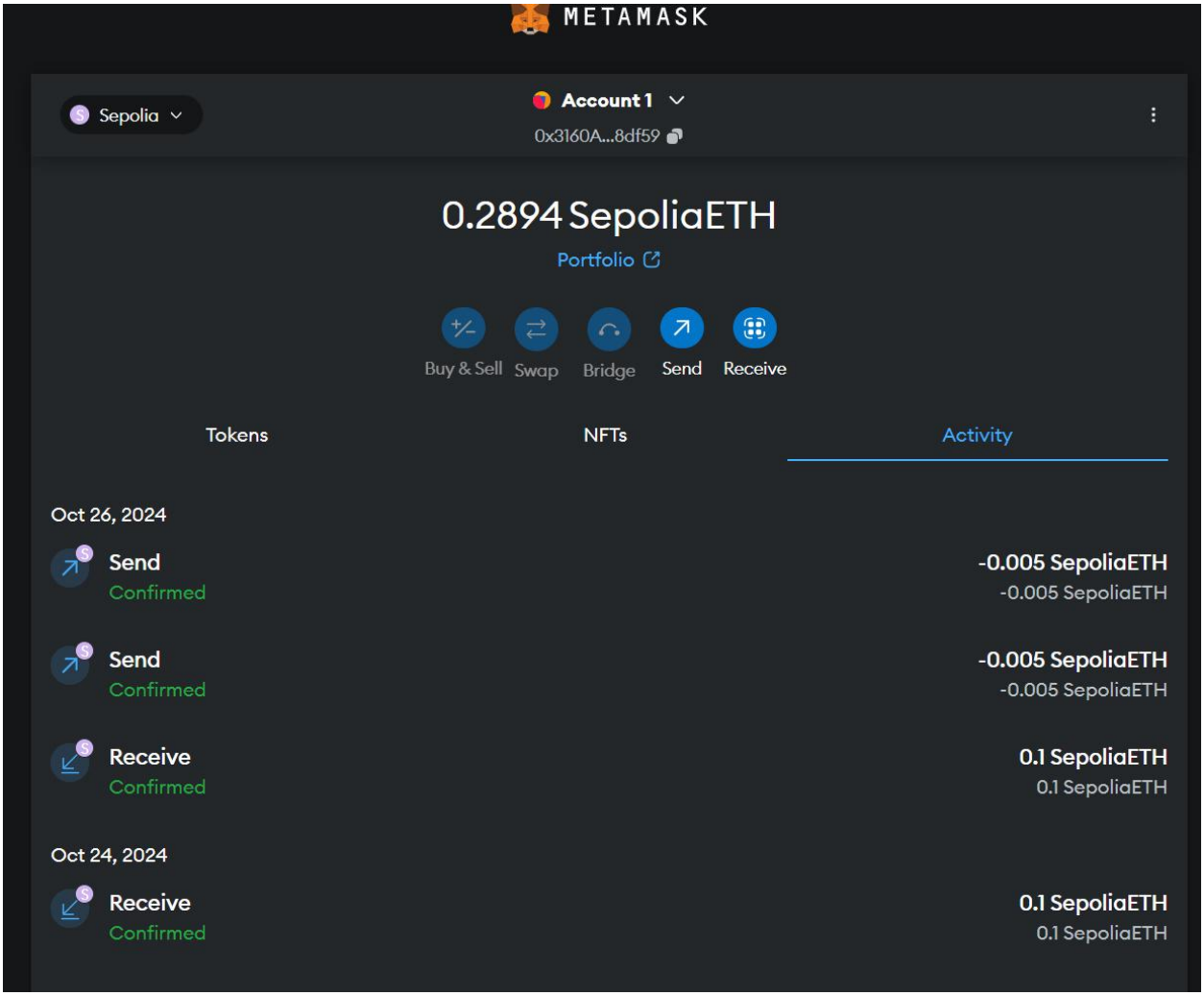
Step 13: Pending Status



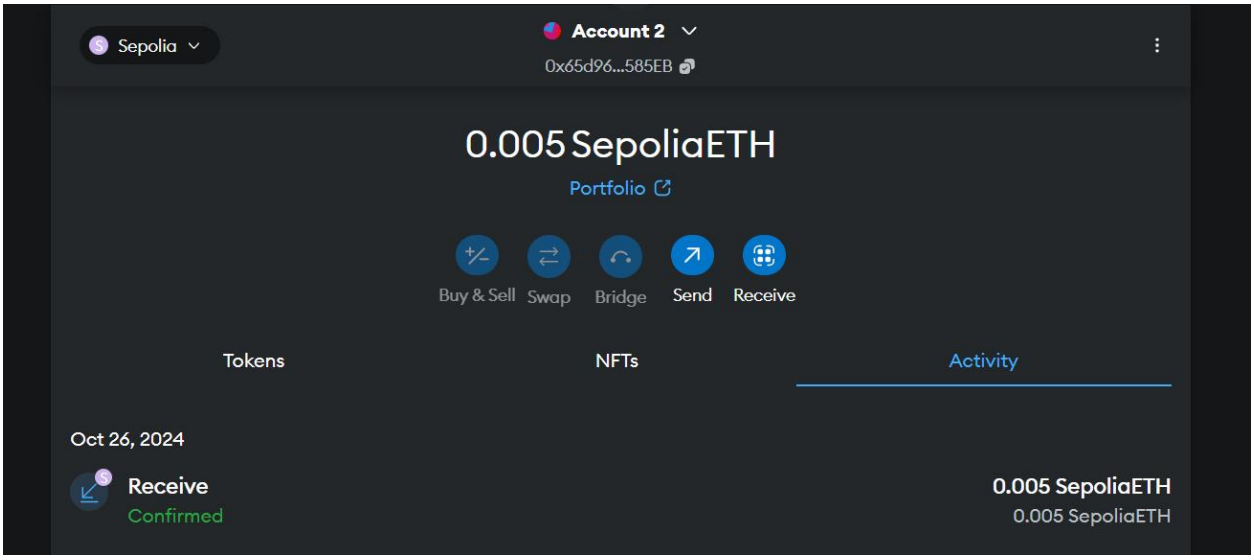
Step 14: Speed Up Process



Step 15: Amount / Coins Deducted from Account 1

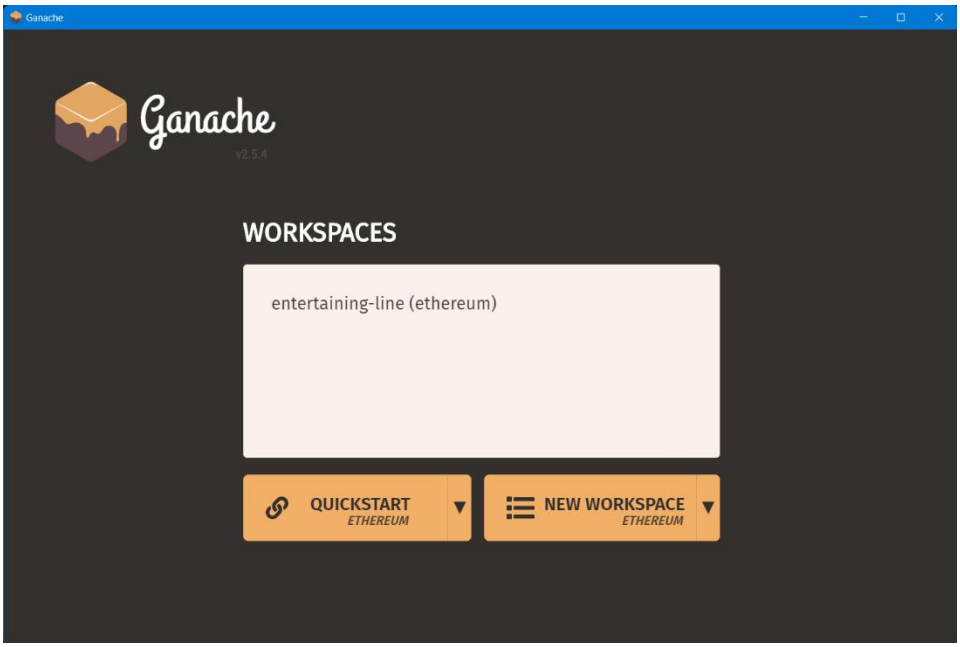


Step 16: Amount Successfully transferred to Account 2

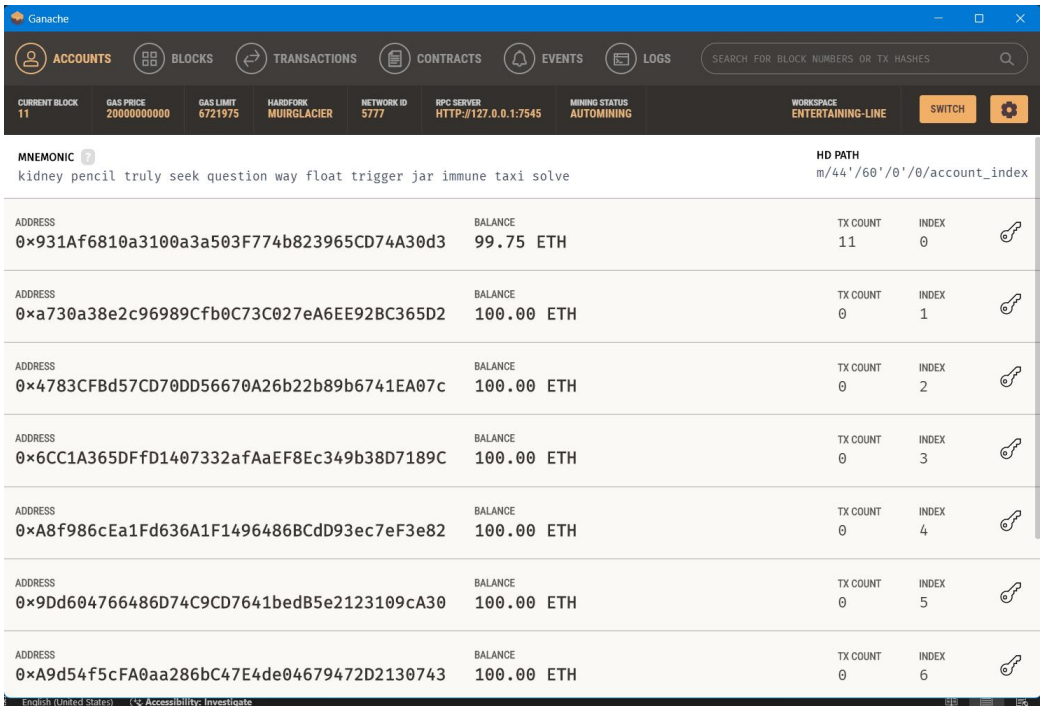


Q.2 Start Ganache (your personal private blockchain network). Connect Ganache with MetaMask and import the account from Ganache to MetaMask. Transfer funds from imported account to other account of MetaMask. Take the screenshots of created accounts, account assets and account transactions which showing the details of transaction from MetaMask and Ganache interface.

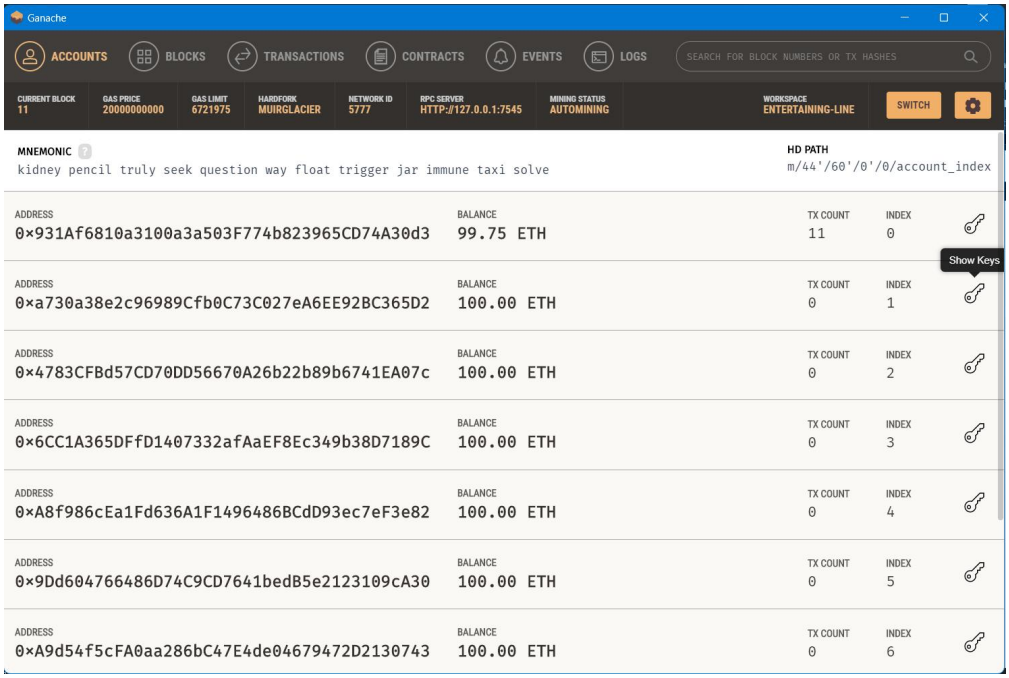
Step 1: Open Ganache IDE



Step 2: Open Dashboard of Ganache

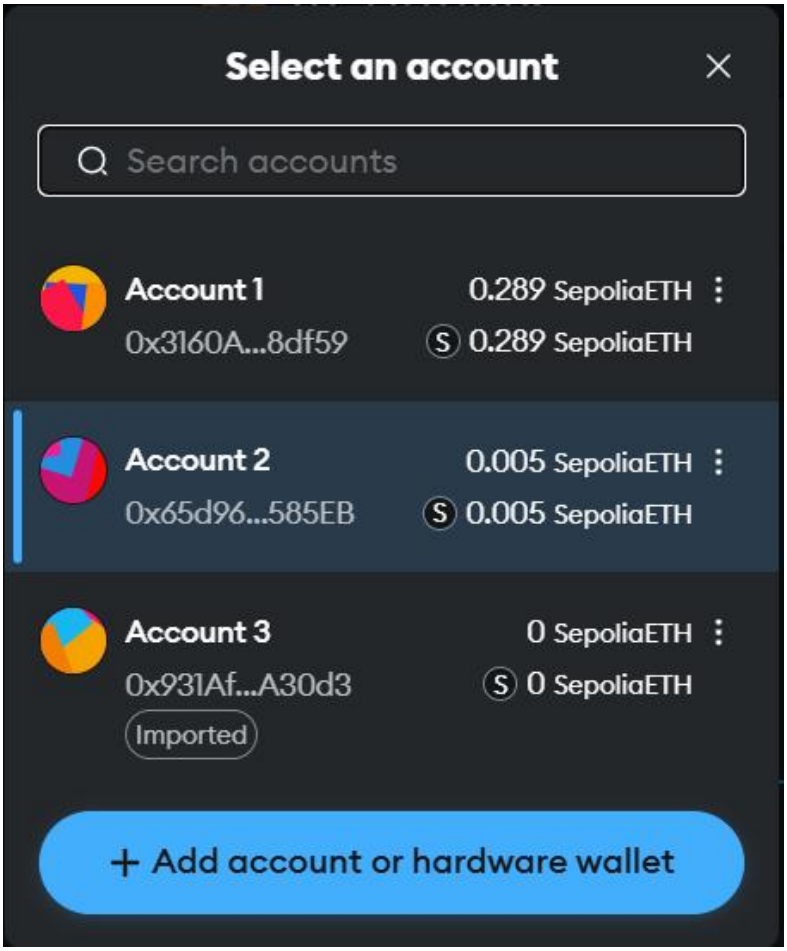


Step 3: Show Keys

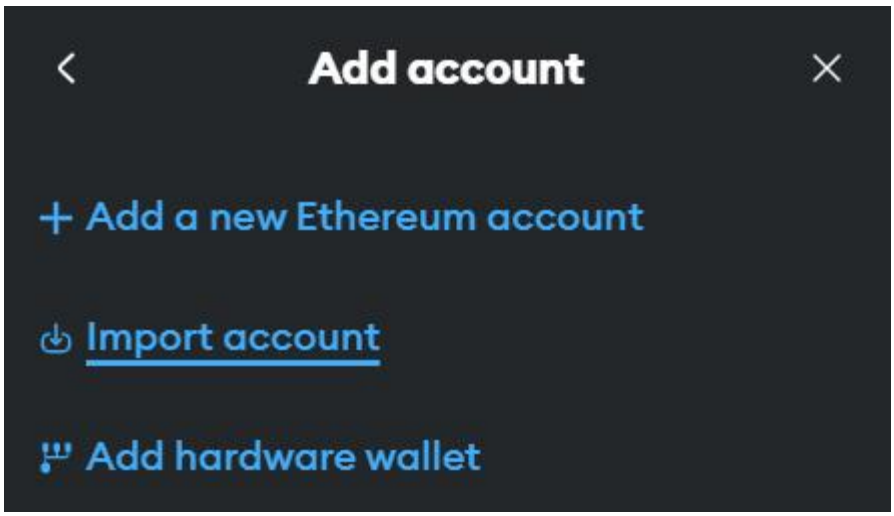


Step 4: Copy the Private key

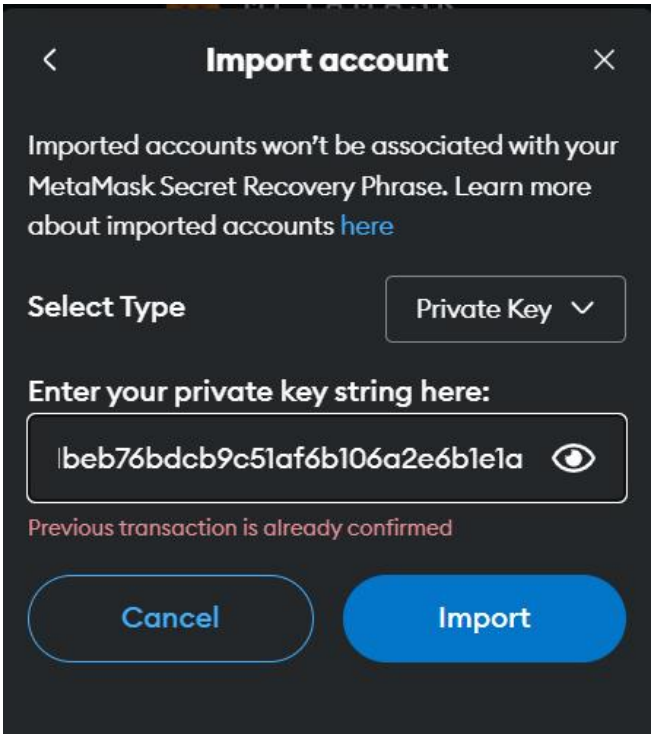
Step 5: Add New Account



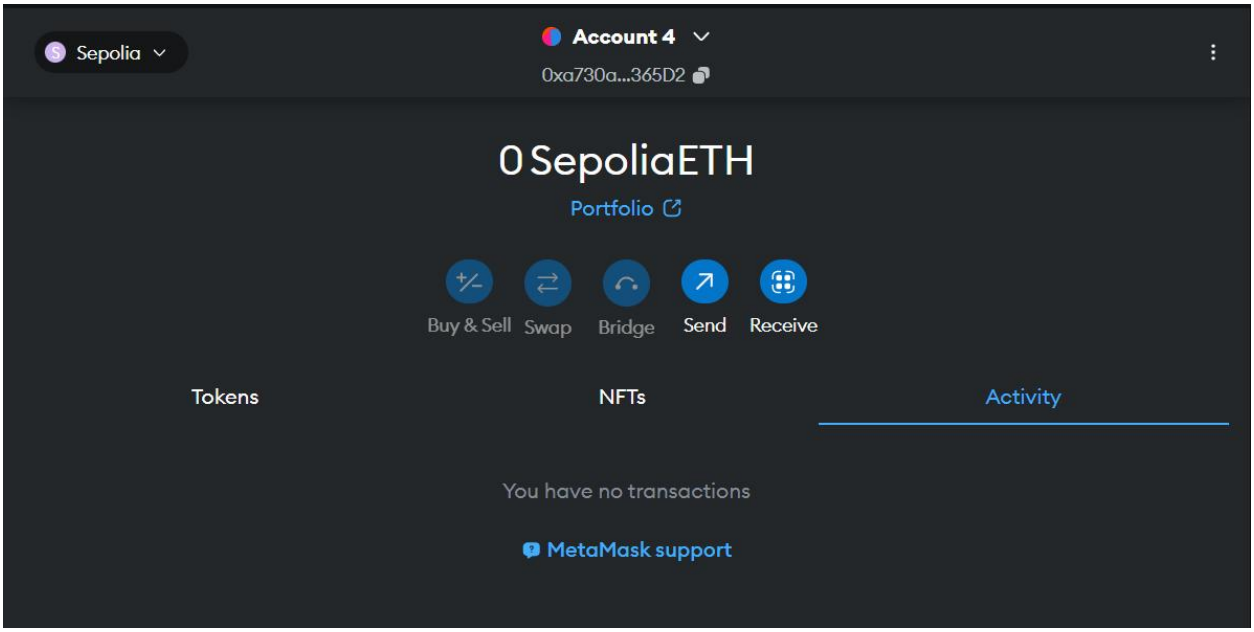
Step 6: Import Account



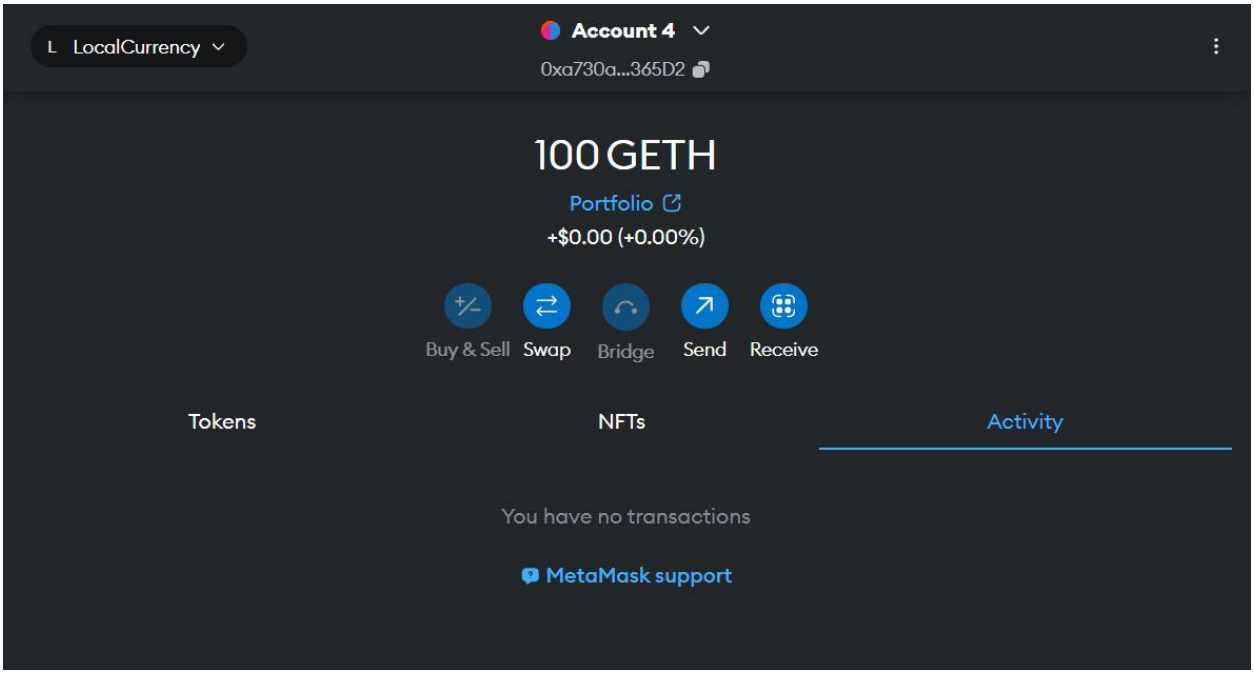
Step 7: Enter Private Key in the Block



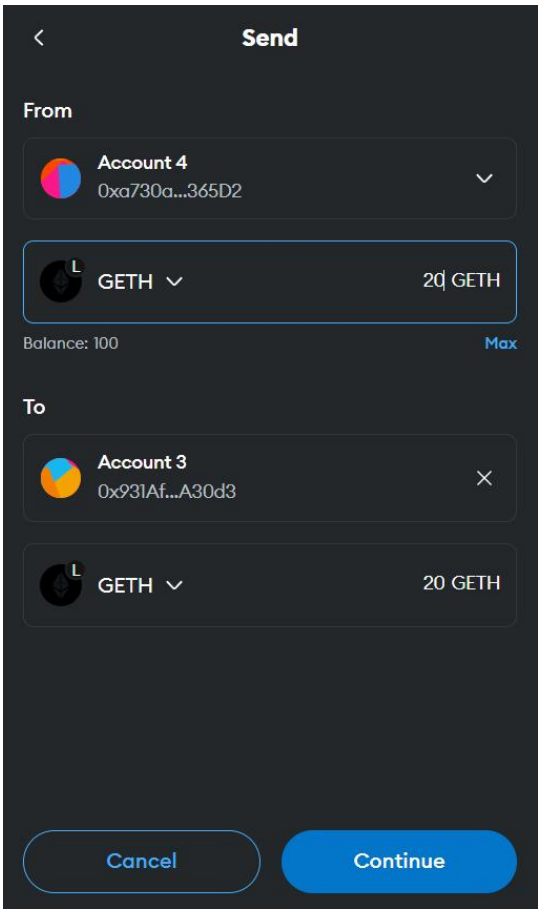
Step 8: Open Account 4

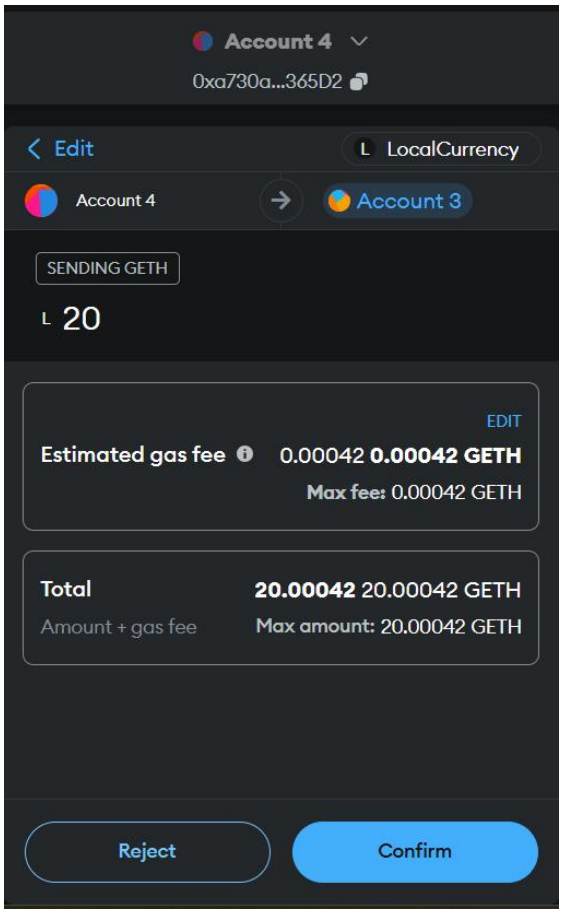


Step 9: Switch to Local Currency

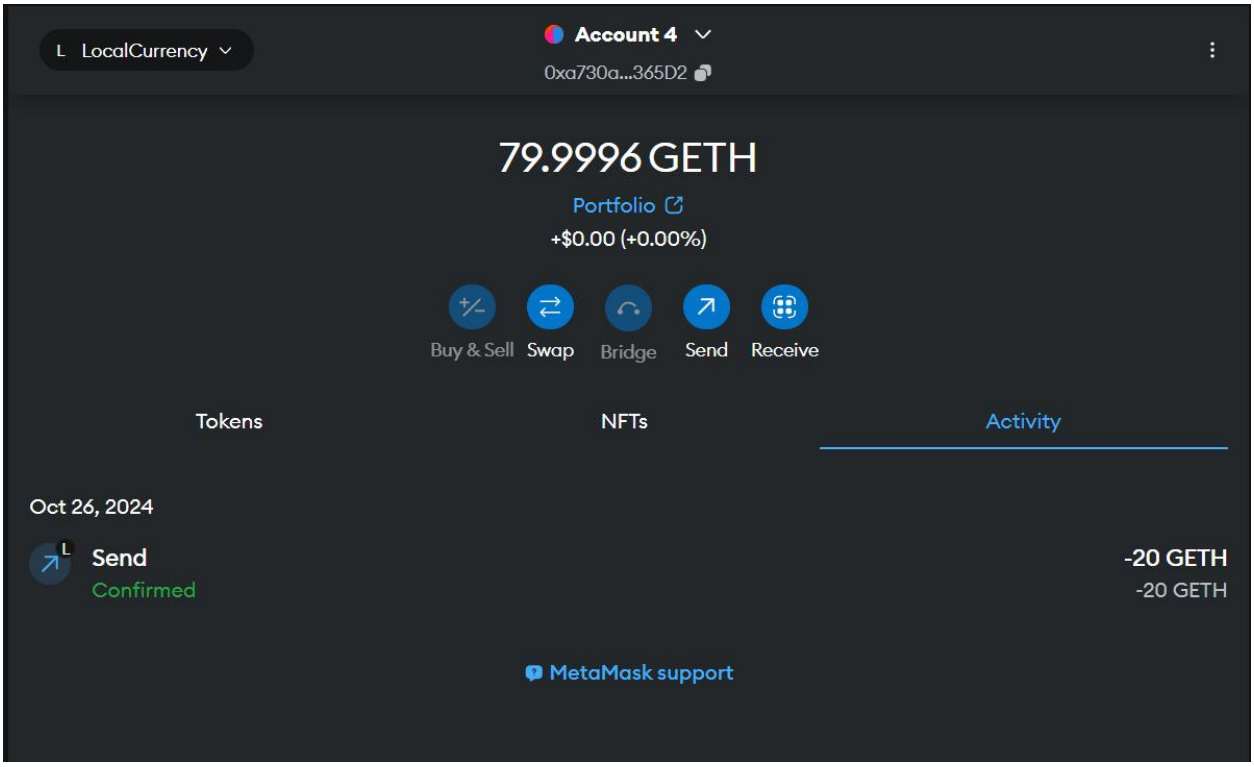


Step 10: Send 20 GETH to Account 3

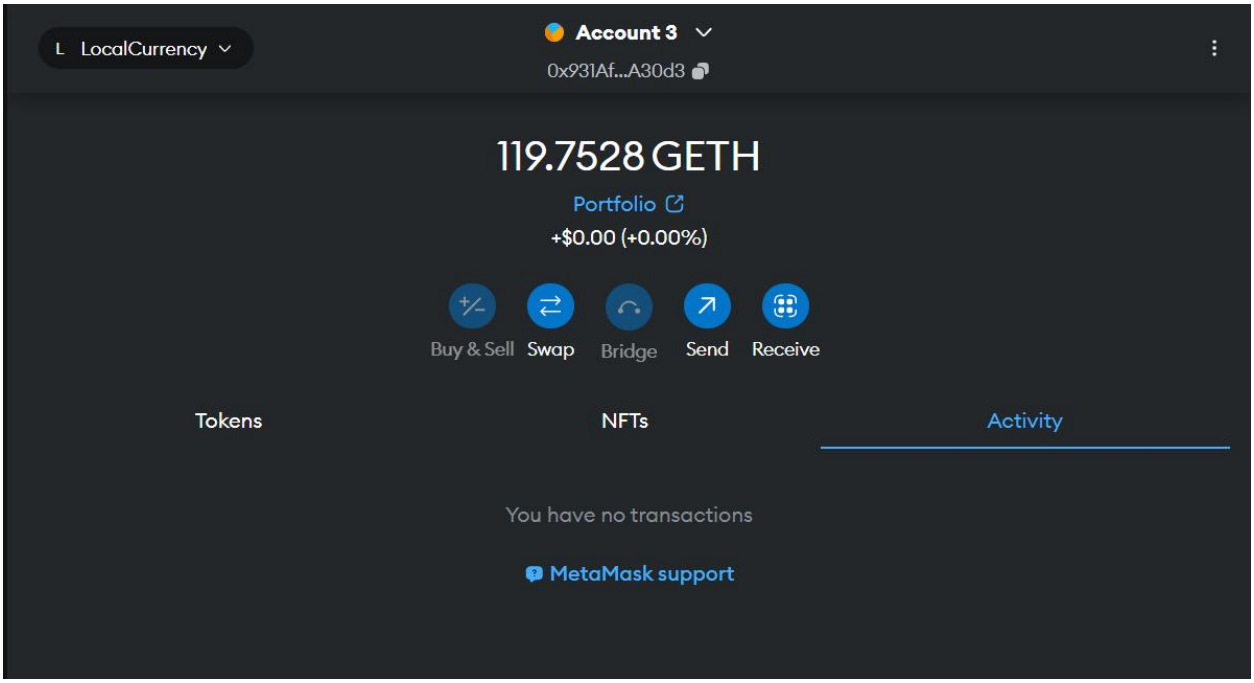




Step 11: Amount has been deducted



Step 12: Added in Account 3



Step 13: Transaction Details

ADDRESS	BALANCE	TX COUNT	INDEX
0×931Af6810a3100a3a503F774b823965CD74A30d3	119.75 ETH	11	0
0×a730a38e2c96989Cfb0C73C027eA6EE92BC365D2	80.00 ETH	1	1

Step 14: New Block Added inn Blockchain

TX HASH
0×aa41beb6345e235a92e654df1b89a10e47a1a9b85781a95a7404819c596e271a

CONTRACT CALL

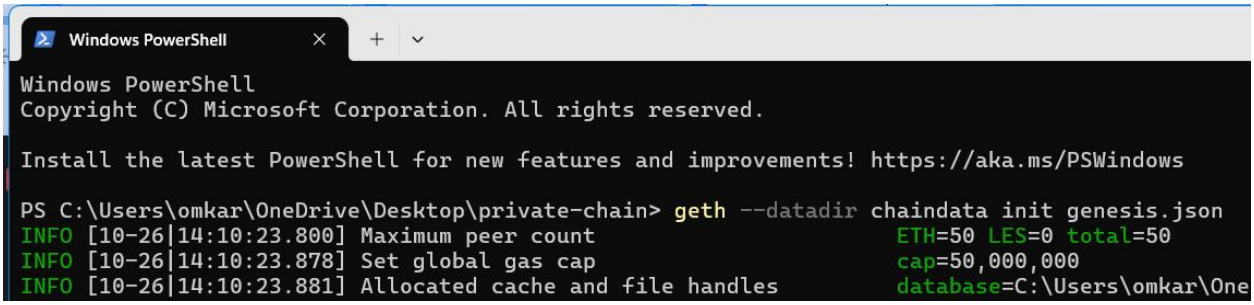
FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0×a730a38e2c96989Cfb0C73C027eA6EE92BC365D2	0×931Af6810a3100a3a503F774b823965CD74A30d3	21000	20000000000000000000

Q.3 Create Ethereum node using Geth (GoEthereum) and create genesis block and create your personal private Ethereum blockchain. And use IPC to interact with Geth node to perform following task: create account, transfer funds using send transaction, mine the block, show the account balance before and after the mining the block, show the specific block details and access chain details.

Step 1: geth –datadir chaindata init genesis.json

genesis.json

```
{ "coinbase"   : "0x0000000000000000000000000000000000000001",
  "difficulty" : "0x20000",
  "extraData"  : "",
  "gasLimit"   : "0x2fefd8",
  "nonce"      : "0x0000000000000042",
  "mixhash"    :
"0x0000000000000000000000000000000000000000000000000000000000000000",
  "parentHash" :
"0x0000000000000000000000000000000000000000000000000000000000000000",
  "timestamp"  : "0x00",
  "alloc": {},
  "config": {
      "chainId": 15,
      "homesteadBlock": 0,
      "eip155Block": 0,
      "eip158Block": 0,
      "eip150Block": 0   } }
```



Step 2: geth –datadir=./chaindata/

```
PS C:\Users\omkar\OneDrive\Desktop\private-chain> geth --datadir=./chaindata/
INFO [10-26|14:10:45.439] Starting Geth on Ethereum mainnet...
INFO [10-26|14:10:45.440] Bumping default cache on mainnet         provided=1024 updated=4096
INFO [10-26|14:10:45.441] Maximum peer count                       ETH=50 LES=0 total=50
INFO [10-26|14:10:45.442] Set global gas cap                       cap=50,000,000
INFO [10-26|14:10:45.442] Allocated trie memory caches             clean=614.00MiB dirty=1024.00MiB
```

Step 3: Open the Another Command Prompt "Run As Administrator"

```
Administrator: Command Prompt - geth  attach ipc:\\.\pipe\geth.ipc
Microsoft Windows [Version 10.0.22631.4391]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.9-stable-eae3b194/windows-amd64/go1.17
at block: 0 (Thu Jan 01 1970 05:30:00 GMT+0530 (IST))
 datadir: C:\Users\omkar\OneDrive\Desktop\private-chain\chaindata
 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
> eth.accounts
[]
> personal.newAccount()
Passphrase:
Repeat passphrase:
"0xc1a2de3823651ab79e827a128a52aa8cd731c270"
>
```

Step 4: Open the Coinbase

```
> eth.accounts
[]
> personal.newAccount()
Passphrase:
Repeat passphrase:
"0xc1a2de3823651ab79e827a128a52aa8cd731c270"
> eth.coinbase
"0xc1a2de3823651ab79e827a128a52aa8cd731c270"
>
```

Step 5: Check the Balance of Account One

```
> eth.getBalance(eth.accounts[0])
0
```

Step 6: Mining Start

```
> miner.start()
null
```

Step 7: New Blocks Added

```
0 elapsed=184.144ms
INFO [10-26|14:26:51.333] 🔗 block reached canonical chain          number=142 hash=8530c4..7d4192
INFO [10-26|14:26:51.335] Commit new mining work                   number=150 sealhash=194794..7458b8 uncles=0 txs=0 gas
=0 fees=0 elapsed=2.144ms
INFO [10-26|14:26:51.344] 🔨 mined potential block                  number=149 hash=365da2..371070
INFO [10-26|14:26:51.696] Successfully sealed new block            number=150 sealhash=194794..7458b8 hash=f2a844..332e6
9 elapsed=362.895ms
INFO [10-26|14:26:51.696] 🔗 block reached canonical chain          number=143 hash=ee054c..4b7813
INFO [10-26|14:26:51.696] Commit new mining work                   number=151 sealhash=c03b25..9bc9d5 uncles=0 txs=0 gas
=0 fees=0 elapsed=0s
INFO [10-26|14:26:51.697] 🔨 mined potential block                  number=150 hash=f2a844..332e69
INFO [10-26|14:26:52.273] Successfully sealed new block            number=151 sealhash=c03b25..9bc9d5 hash=aa993f..8f6bc
d elapsed=576.437ms
INFO [10-26|14:26:52.273] 🔗 block reached canonical chain          number=144 hash=acc492..cc5531
INFO [10-26|14:26:52.275] Commit new mining work                   number=152 sealhash=9c5f1b..f8788c uncles=0 txs=0 gas
=0 fees=0 elapsed=1.857ms
INFO [10-26|14:26:52.287] 🔨 mined potential block                  number=151 hash=aa993f..8f6bcd
INFO [10-26|14:26:52.640] Successfully sealed new block            number=152 sealhash=9c5f1b..f8788c hash=c038b2..267b9
9 elapsed=367.126ms
```

Step 8: Mining Stop

```
> miner.stop()
null
```

Step 9: Get New Balance

```
> eth.getBalance(eth.accounts[0])
2.815e+21
```

Step 10: Unlock Account

```
> personal.unlockAccount(eth.accounts[0])
Unlock account 0xc1a2de3823651ab79e827a128a52aa8cd731c270
Passphrase:
true
```

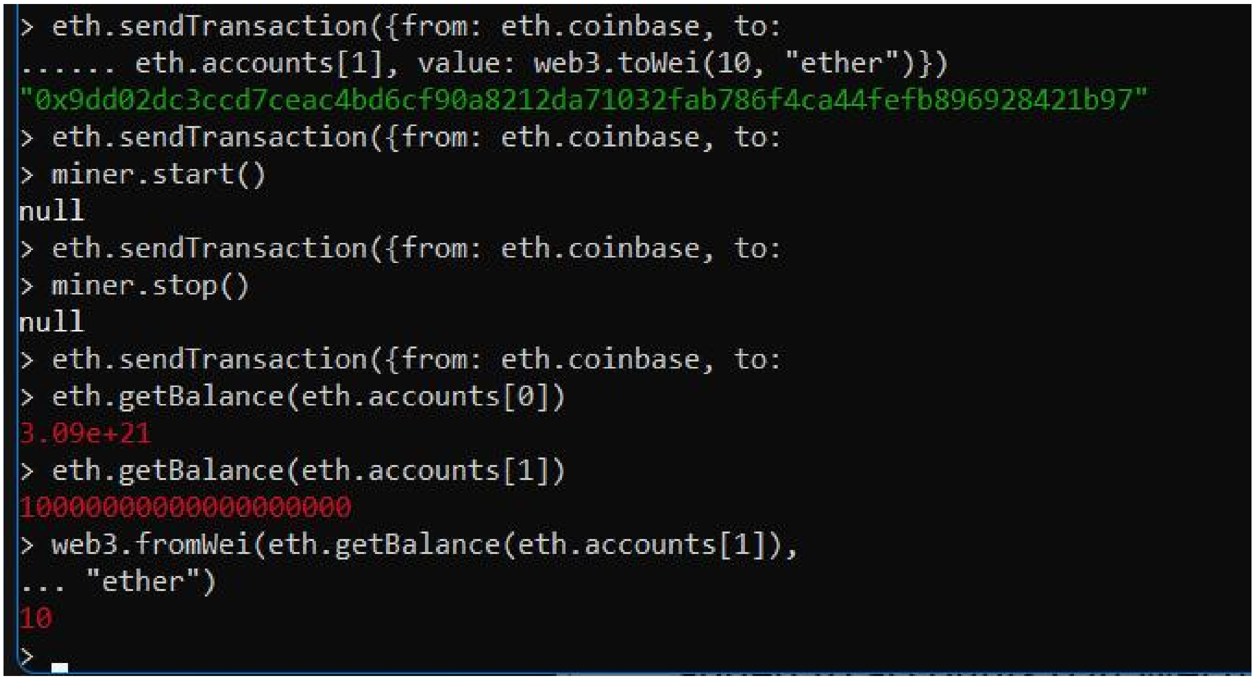
Step 11: Create new Account

```
> eth.accounts
[]
> personal.newAccount()
Passphrase:
Repeat passphrase:
"0xc1a2de3823651ab79e827a128a52aa8cd731c270"
> eth.coinbase
"0xc1a2de3823651ab79e827a128a52aa8cd731c270"
```

Step 12: Send Transaction from Account 1 to Account 2

```
> eth.sendTransaction({from: eth.coinbase, to:
..... eth.accounts[1], value: web3.toWei(10, "ether")})
"0x9dd02dc3ccd7ceac4bd6cf90a8212da71032fab786f4ca44fefb896928421b97"
```

Step 13: Get Balance



```
> eth.sendTransaction({from: eth.coinbase, to:
...... eth.accounts[1], value: web3.toWei(10, "ether")})
"0x9dd02dc3ccd7ceac4bd6cf90a8212da71032fab786f4ca44fefb896928421b97"
> eth.sendTransaction({from: eth.coinbase, to:
> miner.start()
null
> eth.sendTransaction({from: eth.coinbase, to:
> miner.stop()
null
> eth.sendTransaction({from: eth.coinbase, to:
> eth.getBalance(eth.accounts[0])
3.09e+21
> eth.getBalance(eth.accounts[1])
10000000000000000000
> web3.fromWei(eth.getBalance(eth.accounts[1]),
... "ether")
10
> 
```

Step 14: Get Latest Block

```
To exit, press ctrl-d or type exit
> eth.getBlock("latest")
{
  difficulty: 176955,
  extraData: "0xd883010a0984676574688667 6f312e31378777696e646f7773",
  gasLimit: 5752767,
  gasUsed: 0,
  hash: "0x79a782bb58c9c96212a5457c18efb1ee88d154dfb679c31a96e3fac59728ab1e",
  logsBloom: "0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000",
  miner: "0xc1a2de3823651ab79e827a128a52aa8cd731c270",
  mixHash: "0xb137a0a7c700d40f35d08d606a8f861923990f64efd68f34190f207a0b2ca599",
  nonce: "0x06f034e84a6a7d61",
  number: 620,
  parentHash: "0xb912312e05ba2c3d7fbe848b1a7d4977c1a1792ef694aada9347d379cef97250",
  receiptsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",
  sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
  size: 538,
  stateRoot: "0x6d074206abd22be5e9ff7b3441e652a8aabe92c626298d9014f59afecea94fd3",
  timestamp: 1729933594,
  totalDifficulty: 94906214,
  transactions: [],
  transactionsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",
  uncles: []
}
> 
```

Step 15: Get Block number 35

```
> eth.getBlock(35)
{
  difficulty: 133250,
  extraData: "0xd883010a0984676574688667 6f312e31378777696e646f7773",
  gasLimit: 3250719,
  gasUsed: 0,
  hash: "0x0834e7494e05ee18499b1662d6a05c15387f9261b515bd9c557bf48e759f90c9",
  logsBloom: "0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000",
  miner: "0xc1a2de3823651ab79e827a128a52aa8cd731c270",
  mixHash: "0xa40fea26f16282dba8646994a145cab81e77c3dd529e026bebdbdad96324d14f",
  nonce: "0x14d0f980398ba23d",
  number: 35,
  parentHash: "0x7997d2dbf732cf0debaedcf18b5855a730ae856f30cf910ca906ce62aed3675b",
  receiptsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",
  sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
  size: 536,
  stateRoot: "0x3b7ddbb9e6cc3c5c3b3f4ffad20f0f5d1acef76a0aa6633e33e33eb8829932d6",
  timestamp: 1729933009,
  totalDifficulty: 4756675,
  transactions: [],
  transactionsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",
  uncles: []
}
> 
```

Q.4 Write a solidity smart contract for performing following task using remixIDE and deployed it on public test network – Goerli / Sapolia using Injected provider environment.

a. To transfer funds (ethers) from user account to contract account.

b. To withdraw funds (ethers) from contract account to user account.

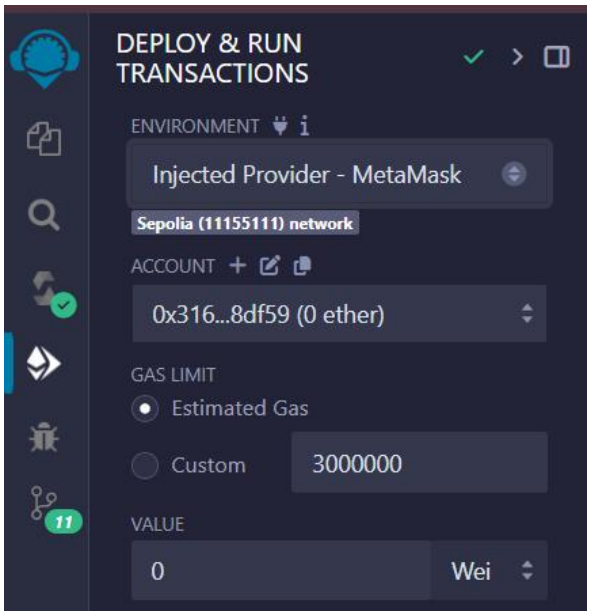
c. To apply restriction that only owner of the contract can withdraw funds (ethers) from contract account to his/her user account.

Code:

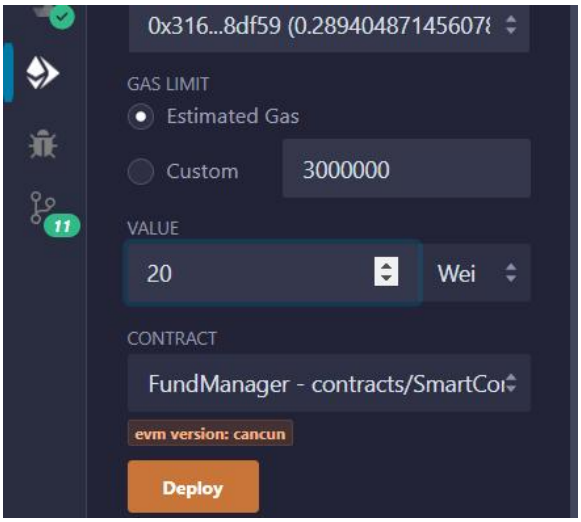
```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract FundManager {

   // Address of the contract owner

   address public owner;

   // Constructor to set the owner as the contract deployer

   constructor() {

      owner = msg.sender;

   }

   // Modifier to restrict access to the contract owner

   modifier onlyOwner() {

      require(msg.sender == owner, "Only the owner can withdraw funds");

      _;

   }

   // Function to deposit Ether into the contract

   function deposit() public payable {

      require(msg.value > 0, "You must send some ether to deposit");

   }

   // Function to withdraw Ether from the contract (only owner)

   function withdraw(uint _amount) public onlyOwner {

      require(address(this).balance >= _amount, "Insufficient balance in contract");
```

```
        payable(msg.sender).transfer(_amount);
    }
    // Function to check contract balance
    function getContractBalance() public view returns (uint) {
        return address(this).balance;
    }
}
```

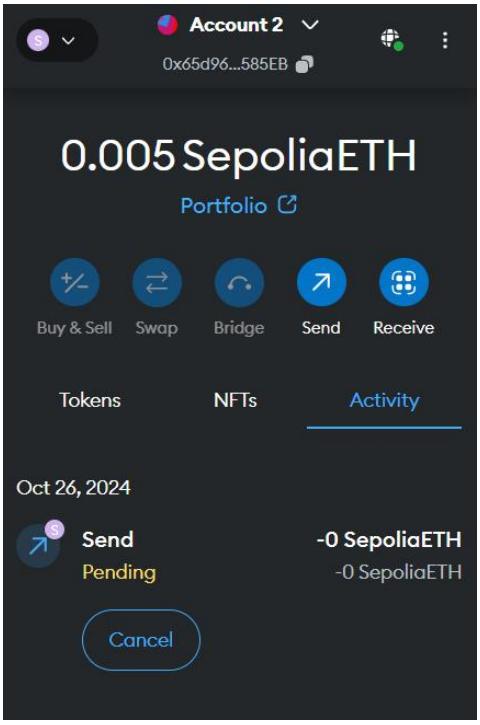
Step 1: Injected Provider - MetaMask



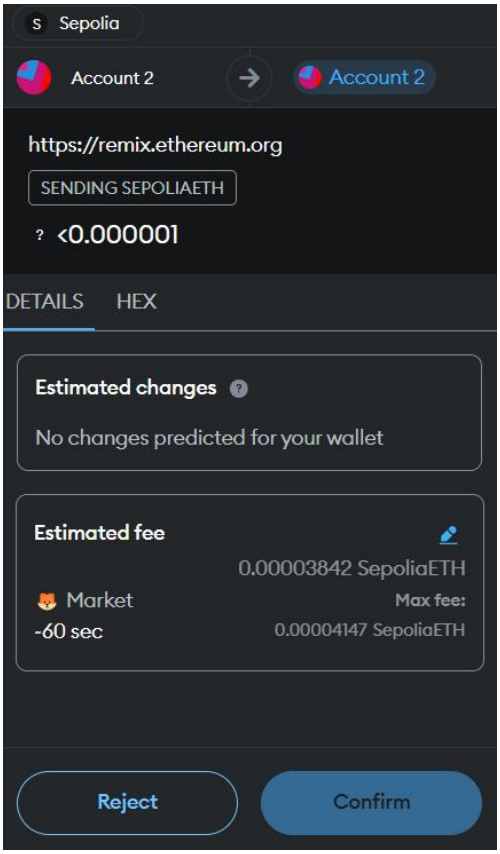
Step 2: Select Contract



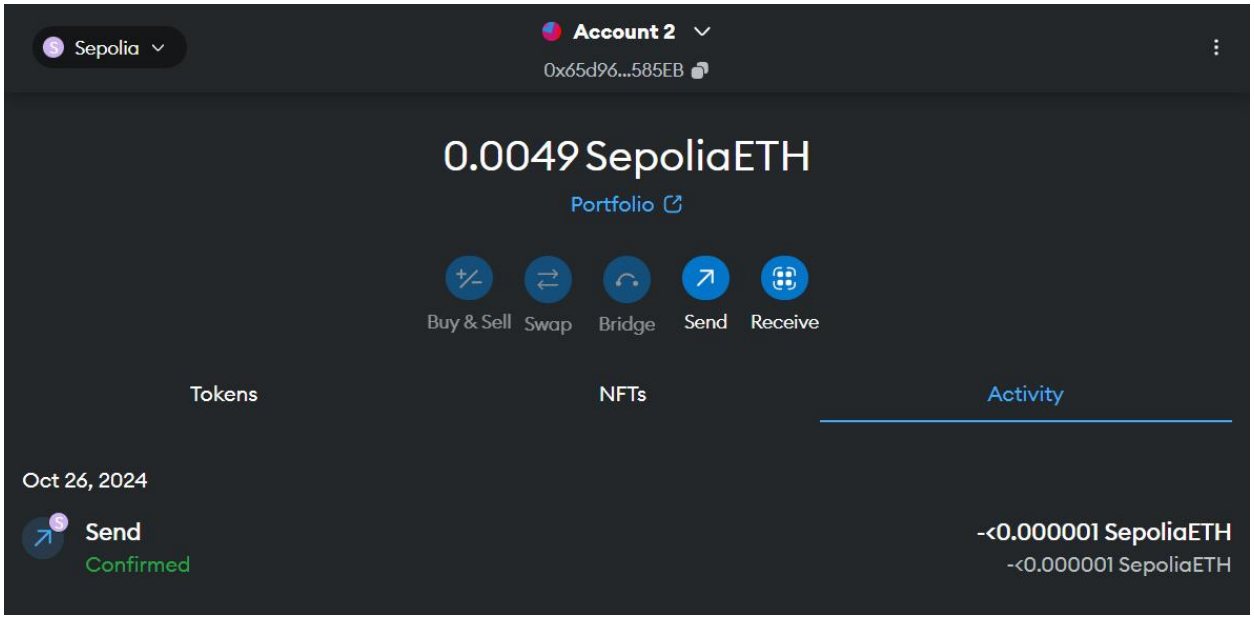
Step 3: Add this Account



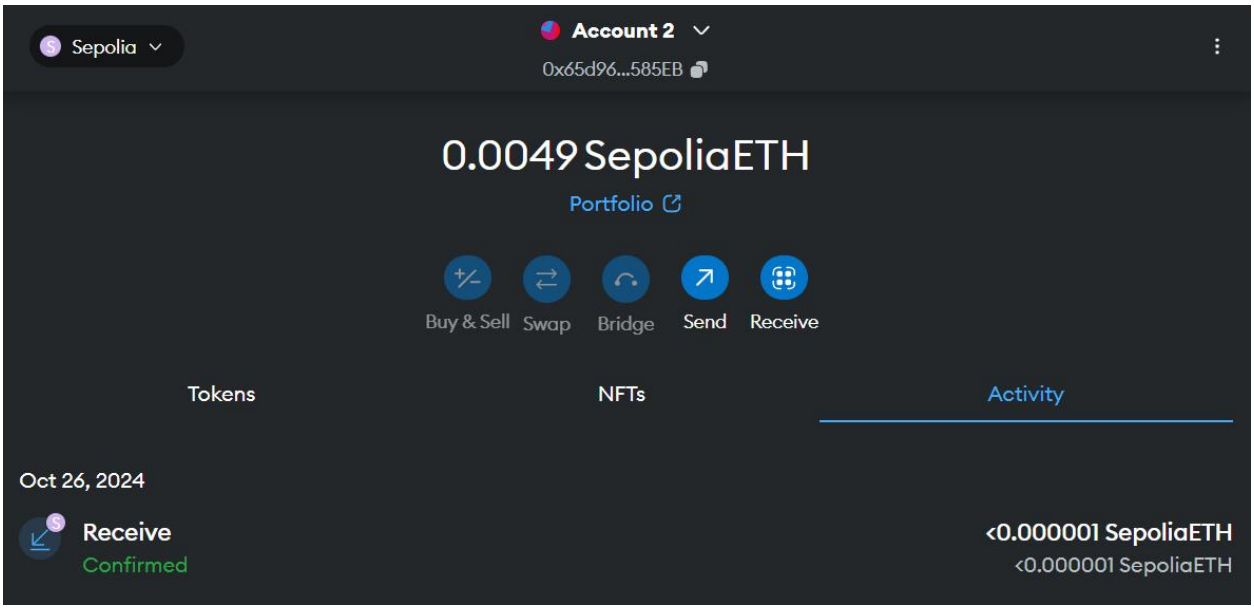
Step 4: Send Amount and Confirm it



Step 5: Send the Coin



Step 6: Receive the Coin



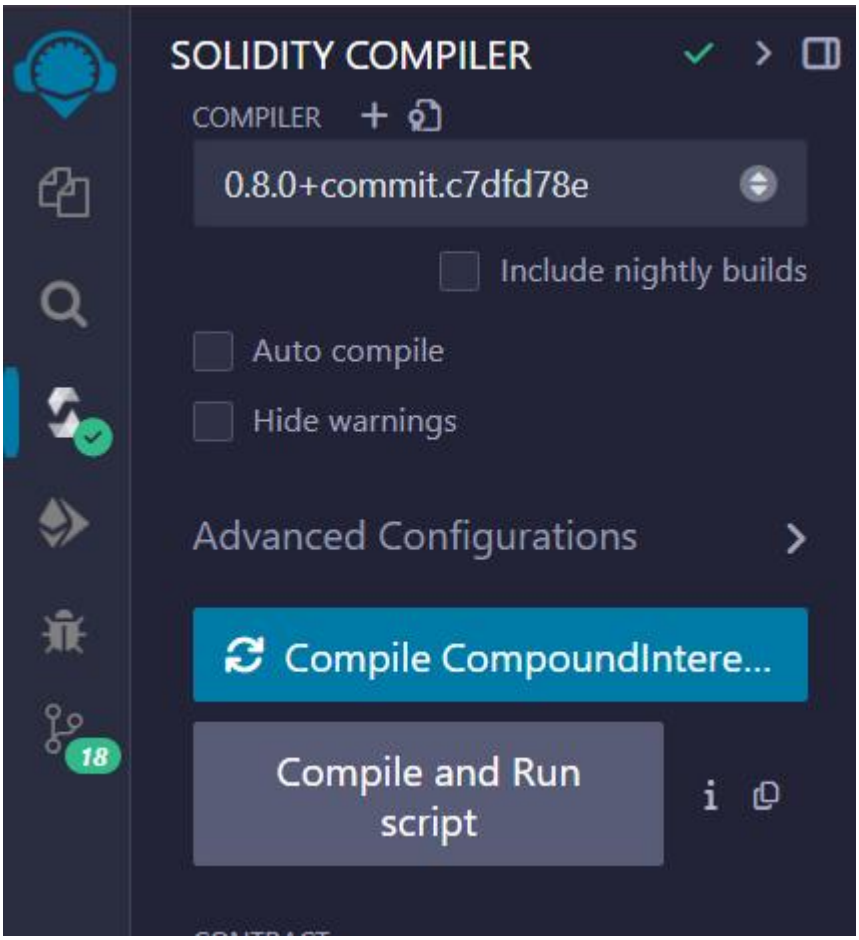
Q.5 Write a smart contract to calculate the compound interest and deploy it on Ganache using injected provider.

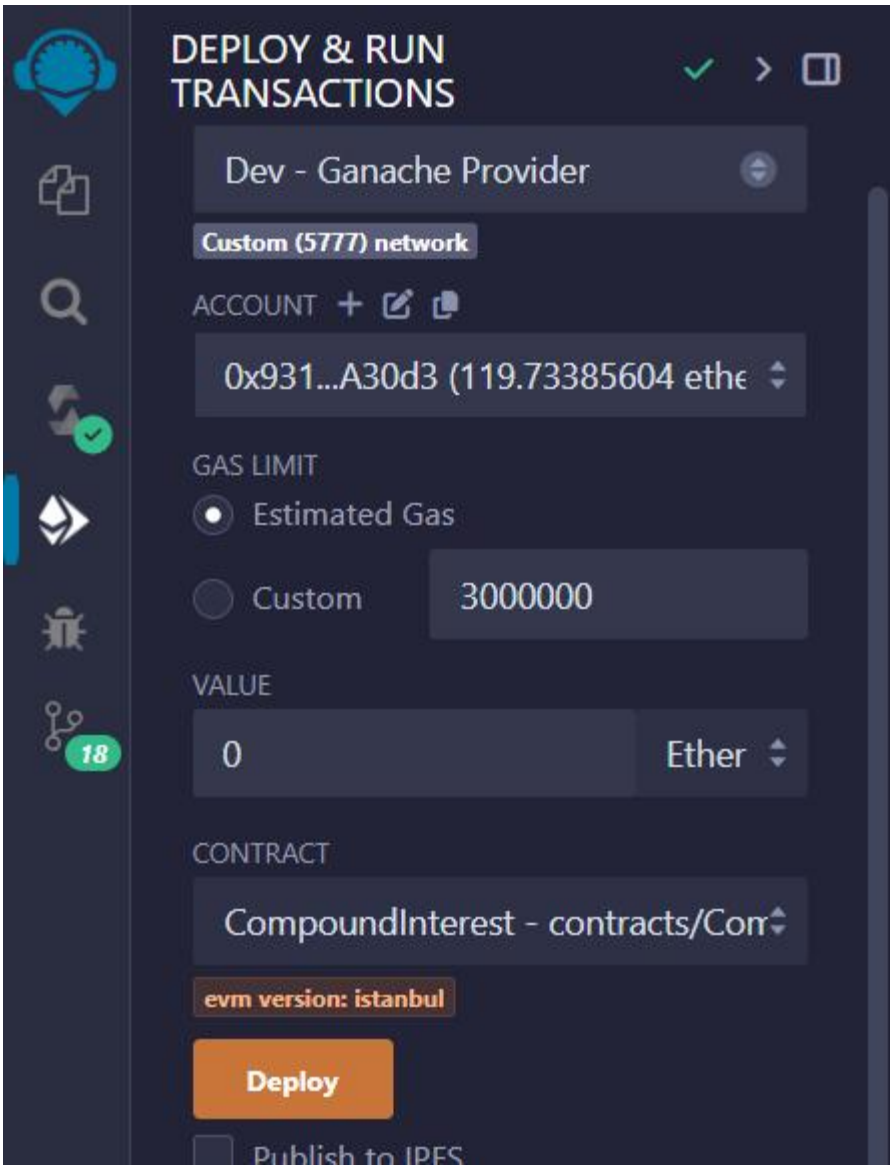
Code:

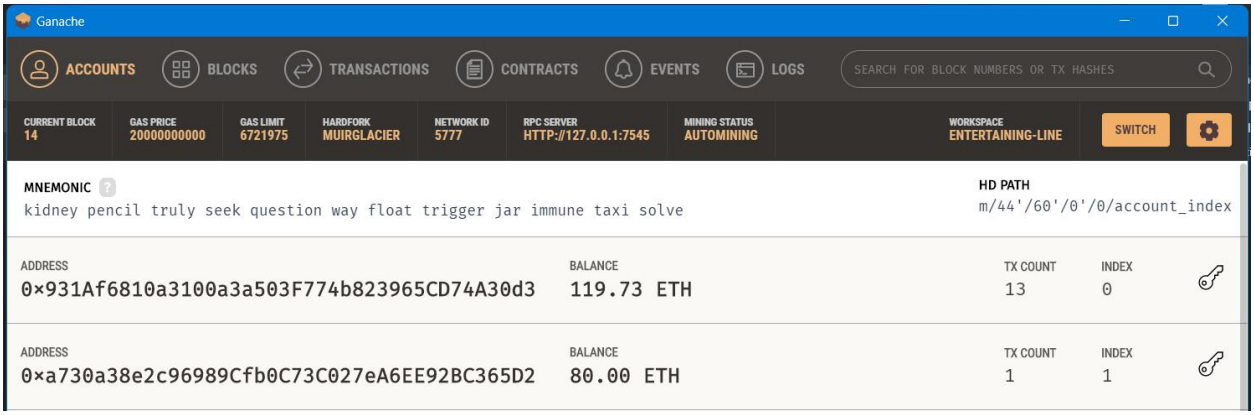
```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;


contract CompoundInterest {

    // State variables

    address public owner;

    uint256 public principal;

    uint256 public rate; // Interest rate in percentage

    uint256 public time;  // Time in years


    // Events

    event InterestCalculated(uint256 totalAmount);


    // Constructor

    constructor() {

        owner = msg.sender; // Set the contract creator as the owner

    }


    // Function to set principal, rate, and time

    function setParameters(uint256 _principal, uint256 _rate, uint256 _time) public payable{

        require(msg.sender == owner, "Only the owner can set parameters");

        principal = _principal;

        rate = _rate;

        time = _time;
```

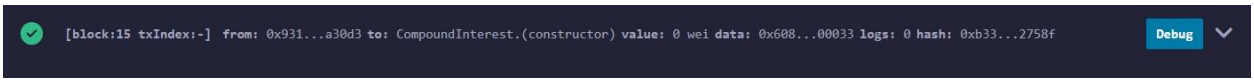
```
    }

    // Function to calculate compound interest
    function calculateCompoundInterest() public returns (uint256) {
        uint256 totalAmount = principal * (1 + rate / 100) ** time;
        emit InterestCalculated(totalAmount);
        return totalAmount;
    }
}
```

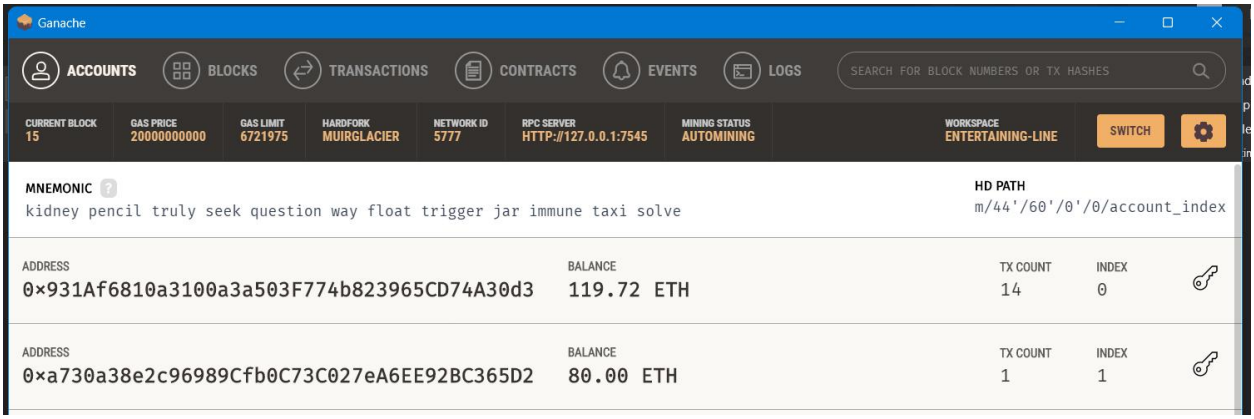
Output:

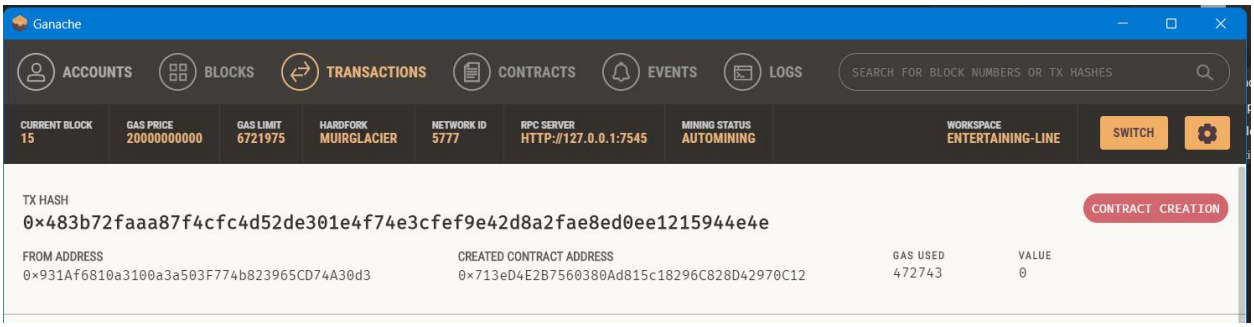


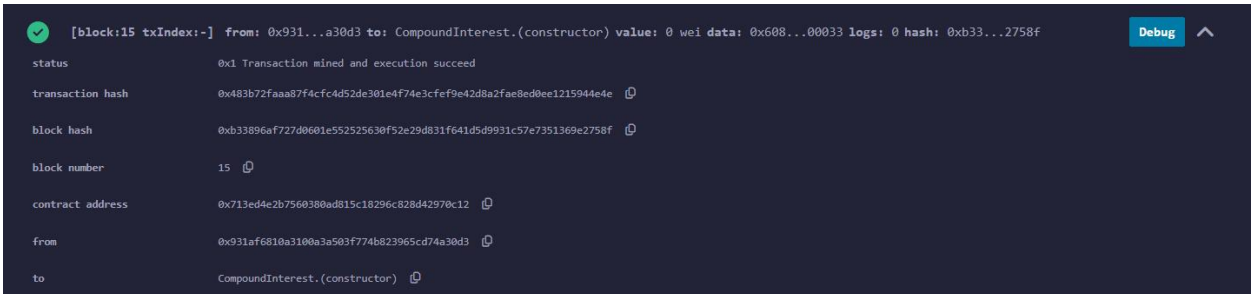

DEPLOY & RUN TRANSACTIONS
Dev - Ganache Provider
Custom (5777) network
ACCOUNT
0x931...A30d3 (119.73385604 ethe
GAS LIMIT
Estimated Gas
Custom
3000000
VALUE
0
Ether
CONTRACT
CompoundInterest - contracts/Com
evm version: istanbul
Deploy
Publish to IPFS
18

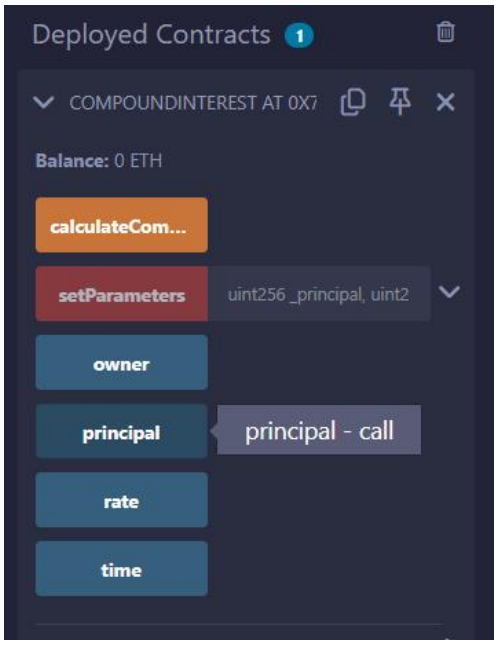

Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 14
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE ENTERTAINING-LINE
SWITCH
MNEMONIC
kidney pencil truly seek question way float trigger jar immune taxi solve
HD PATH
m/44'/60'/0'/0/account_index
ADDRESS
0x931Af6810a3100a3a503F774b823965CD74A30d3
BALANCE
119.73 ETH
TX COUNT
13
INDEX
0
ADDRESS
0xa730a38e2c96989Cfb0C73C027eA6EE92BC365D2
BALANCE
80.00 ETH
TX COUNT
1
INDEX
1


[block:15 txIndex:-] from: 0x931...a30d3 to: CompoundInterest.(constructor) value: 0 wei data: 0x608...00033 logs: 0 hash: 0xb33...2758f
Debug


Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 15
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE ENTERTAINING-LINE
SWITCH
MNEMONIC
kidney pencil truly seek question way float trigger jar immune taxi solve
HD PATH
m/44'/60'/0'/0/account_index
ADDRESS
0x931Af6810a3100a3a503F774b823965CD74A30d3
BALANCE
119.72 ETH
TX COUNT
14
INDEX
0
ADDRESS
0xa730a38e2c96989Cfb0C73C027eA6EE92BC365D2
BALANCE
80.00 ETH
TX COUNT
1
INDEX
1


Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 15
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE ENTERTAINING-LINE
SWITCH
TX HASH
0x483b72faaa87f4cfc4d52de301e4f74e3cfef9e42d8a2fae8ed0ee1215944e4e
CONTRACT CREATION
FROM ADDRESS
0x931Af6810a3100a3a503F774b823965CD74A30d3
CREATED CONTRACT ADDRESS
0x713eD4E2B7560380Ad815c18296C828D42970C12
GAS USED
472743
VALUE
0


[block:15 txIndex:-] from: 0x931...a30d3 to: CompoundInterest.(constructor) value: 0 wei data: 0x608...00033 logs: 0 hash: 0xb33...2758f
Debug
status
0x1 Transaction mined and execution succeed
transaction hash
0x483b72faaa87f4cfc4d52de301e4f74e3cfef9e42d8a2fae8ed0ee1215944e4e
block hash
0xb33896af727d0601e552525630f52e29d831f641d5d9931c57e7351369e2758f
block number
15
contract address
0x713ed4e2b7560380ad815c18296c828d42970c12
from
0x931af6810a3100a3a503f774b823965cd74a30d3
to
CompoundInterest.(constructor)


Deployed Contracts 1
COMPOUNDINTEREST AT 0X7
Balance: 0 ETH
calculateCom...
setParameters
uint256 _principal, uint2
owner
principal
principal - call
rate
time

[block:16 txIndex:-] **from:** 0x931...a30d3 **to:** CompoundInterest.calculateCompoundInterest() 0x713...70c12 **value:** 0 wei **data:** 0x0fd...ba7ee **logs:** 1 **hash:** 0xf69...86e35

Debug

status	0x1 Transaction mined and execution succeed
transaction hash	0xb85821f1fd64fe04e20519da66de094ec141bc403a7049917c80a21782643dc7
block hash	0xf69e45c35b829211a64b48971339eb7c1d798f4eb3f9e1a4f8540e2bb2286e35
block number	16
from	0x931af6810a3100a3a503f774b823965cd74a30d3
to	CompoundInterest.calculateCompoundInterest() 0x713ed4e2b7560380ad815c18296c828d42970c12
gas	25799 gas
transaction cost	25799 gas
input	0x0fd...ba7ee
decoded input	{}

decoded input {}

decoded output -

logs


```
[
    {
        "from": "0x713ed4e2b7560380ad815c18296c828d42970c12",
        "topic": "0xc4f4a463e90f0c4b77e80da0bd41d2915beeed53b5b9bd1427325e93699451ff",
        "event": "InterestCalculated",
        "args": {
            "0": "0",
            "totalAmount": "0"
        }
    }
]
```


raw logs

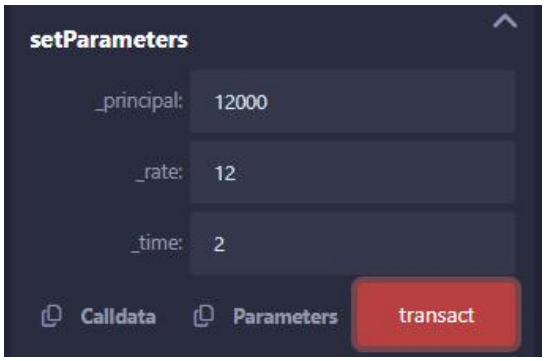

```
[
  {
    "logIndex": "0x0",
    "transactionIndex": "0x0",
    "transactionHash": "0xb85821f1fd64fe04e20519da66de094ec141bc403a7049917c80a21782643dc7",
    "blockHash": "0xf69e45c35b829211a64b48971339eb7c1d798f4eb3f9e1a4f8540e2bb2286e35",
    "blockNumber": "0x10",
    "address": "0x713ed4e2b7560380ad815c18296c828d42970c12",
    "data": "0x0000000000000000000000000000000000000000000000000000000000000000",
    "topics": [
      "0xc4f4a463e90f0c4b77e80da0bd41d2915beeed53b5b9bd1427325e93699451ff"
    ]
  }
]
```

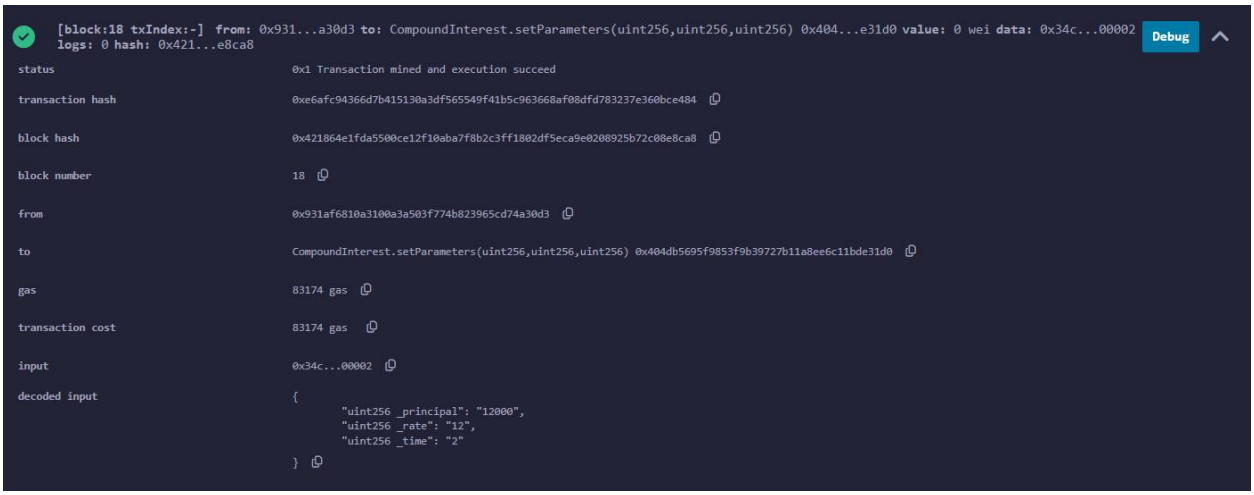

owner

0: address: 0x931Af6810a3100a3a503F774b823965CD74A30d3

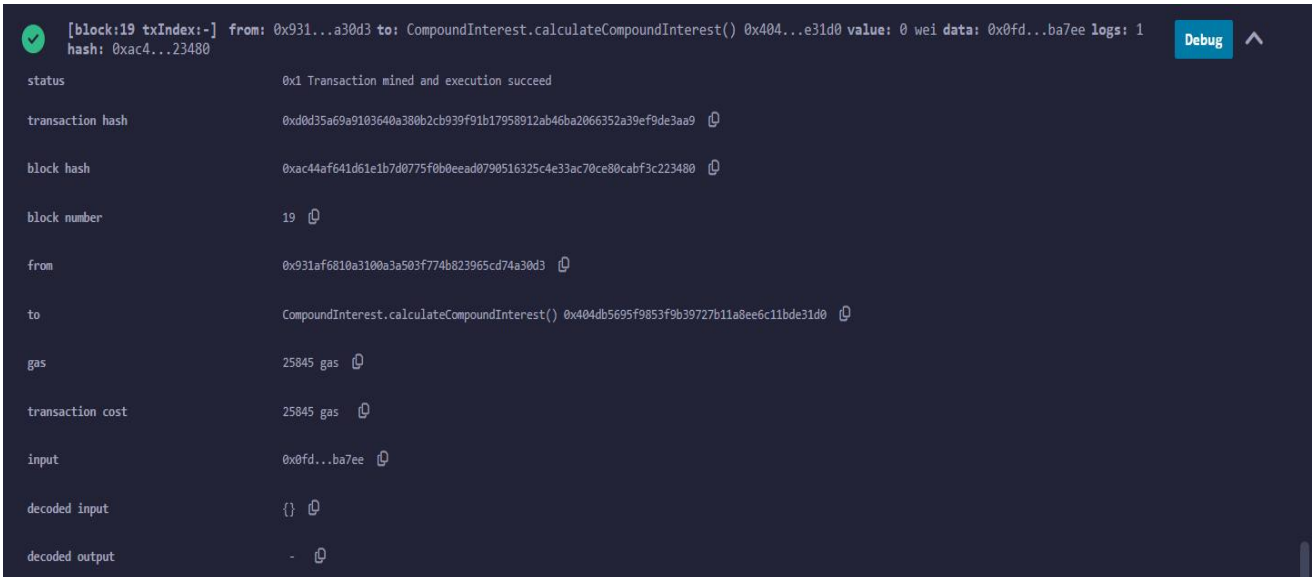
CALL [call] **from:** 0x931Af6810a3100a3a503F774b823965CD74A30d3 **to:** CompoundInterest.owner() **data:** 0x8da...5cb5b

from	0x931Af6810a3100a3a503F774b823965CD74A30d3
to	CompoundInterest.owner() 0x404DB5695F9853F9B39727b11a8eE6c11bDE31D0
input	0x8da...5cb5b
output	0000000000000147262461291049016316532471161843515092215744821 1
decoded input	{}
decoded output	{ "0": "address: 0x931Af6810a3100a3a503F774b823965CD74A30d3" }


setParameters
_principal:
12000
_rate:
12
_time:
2
Calldata
Parameters
transact


[block:18 txIndex:-] from: 0x931...a30d3 to: CompoundInterest.setParameters(uint256,uint256,uint256) 0x404...e31d0 value: 0 wei data: 0x34c...00002 logs: 0 hash: 0x421...e8ca8
Debug
status
0x1 Transaction mined and execution succeed
transaction hash
0xe6afc94366d7b415130a3df565549f41b5c963668af08dfd783237e360bce484
block hash
0x421864e1fda5500ce12f10aba7f8b2c3ff1802df5eca9e0208925b72c08e8ca8
block number
18
from
0x931af6810a3100a3a503f774b823965cd74a30d3
to
CompoundInterest.setParameters(uint256,uint256,uint256) 0x404db5695f9853f9b39727b11a8ee6c11bde31d0
gas
83174 gas
transaction cost
83174 gas
input
0x34c...00002
decoded input
{
"uint256 _principal": "12000",
"uint256 _rate": "12",
"uint256 _time": "2"
}


calculateCom...


[block:19 txIndex:-] from: 0x931...a30d3 to: CompoundInterest.calculateCompoundInterest() 0x404...e31d0 value: 0 wei data: 0x0fd...ba7ee logs: 1 hash: 0xac4...23480
Debug
status
0x1 Transaction mined and execution succeed
transaction hash
0xd0d35a69a9103640a380b2cb939f91b17958912ab46ba2066352a39ef9de3aa9
block hash
0xac44af641d61e1b7d0775f0b0eead0790516325c4e33ac70ce80cabf3c223480
block number
19
from
0x931af6810a3100a3a503f774b823965cd74a30d3
to
CompoundInterest.calculateCompoundInterest() 0x404db5695f9853f9b39727b11a8ee6c11bde31d0
gas
25845 gas
transaction cost
25845 gas
input
0x0fd...ba7ee
decoded input
{}
decoded output
-

logs


```
[
    {
        "from": "0x404db5695f9853f9b39727b11a8ee6c11bde31d0",
        "topic": "0xc4f4a463e90f0c4b77e80da0bd41d2915beeed53b5b9bd1427325e93699451ff",
        "event": "InterestCalculated",
        "args": {
            "0": "12000",
            "totalAmount": "12000"
        }
    }
]
```


raw logs


```
[
  {
    "logIndex": "0x0",
    "transactionIndex": "0x0",
    "transactionHash": "0xd0d35a69a9103640a380b2cb939f91b17958912ab46ba2066352a39ef9de3aa9",
    "blockHash": "0xac44af641d61e1b7d0775f0b0eead0790516325c4e33ac70ce80cabf3c223480",
    "blockNumber": "0x13",
    "address": "0x404db5695f9853f9b39727b11a8ee6c11bde31d0",
    "data": "0x0000000000000000000000000000000000000000000000000000000000002ee0",
    "topics": [
      "0xc4f4a463e90f0c4b77e80da0bd41d2915beeed53b5b9bd1427325e93699451ff"
    ]
  }
]
```


principal principal - call

0: uint256: 12000

call to CompoundInterest.principal

CALL **[call] from:** 0x931Af6810a3100a3a503F774b823965CD74A30d3 **to:** CompoundInterest.principal() **data:** 0xba5...d3078

from	0x931Af6810a3100a3a503F774b823965CD74A30d3
to	CompoundInterest.principal() 0x404DB5695F9853F9B39727b11a8eE6c11bDE31D0
input	0xba5...d3078
output	0000000000000000000000000000000000046224
decoded input	{}
decoded output	{ "0": "uint256: 12000" }
logs	[]
raw logs	[]

rate | rate - call

0: uint256: 12

```
CALL [call] from: 0x931Af6810a3100a3a503F774b823965CD74A30d3 to: CompoundInterest.rate() data: 0x2c4...e722e

from            0x931Af6810a3100a3a503F774b823965CD74A30d3
to              CompoundInterest.rate() 0x404DB5695F9853F9B39727b11a8eE6c11bDE31D0
input           0x2c4...e722e
output          0000000000000000000000000000000012
decoded input   {}
decoded output  {
                    "0": "uint256: 12"
                }
logs            []
raw logs        []
```

time | time - call

0: uint256: 2

```
call to CompoundInterest.time

CALL [call] from: 0x931Af6810a3100a3a503F774b823965CD74A30d3 to: CompoundInterest.time() data: 0x16a...da547

from            0x931Af6810a3100a3a503F774b823965CD74A30d3
to              CompoundInterest.time() 0x404DB5695F9853F9B39727b11a8eE6c11bDE31D0
input           0x16a...da547
output          0000000000000000000000000000000002
decoded input   {}
decoded output  {
                    "0": "uint256: 2"
                }
logs            []
raw logs        []
```

CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE	
20	20000000000	6721975	MUIRGLACIER	5777	HTTP://127.0.0.1:7545	AUTOMINING	ENTERTAINING-LINE	SWITCH

BLOCK	MINED ON	GAS USED	
20	2024-10-27 22:38:42	541747	1 TRANSACTION
19	2024-10-27 21:54:23	25845	1 TRANSACTION
18	2024-10-27 21:54:08	83174	1 TRANSACTION
17	2024-10-27 21:52:22	472743	1 TRANSACTION
16	2024-10-27 21:49:35	25799	1 TRANSACTION
15	2024-10-27 21:48:16	472743	1 TRANSACTION
14	2024-10-27 21:30:06	472743	1 TRANSACTION
13	2024-10-27 21:26:39	472743	1 TRANSACTION

Q.6 Build and test decentralized application (Dapp) for Election Voting System on the local Ethereum Blockchain Network Ganache using truffle suite.

Code:

MyVoting.sol

```
//SPDX-License-Identifier: MIT
pragma solidity >=0.5.0 <0.8.27;
contract MyVoting {
    struct Candidate {
        uint256 id;
        string name;
        uint256 voteCount;
    }
    mapping(address => bool) public voters;
    mapping(uint256 => Candidate) public candidates;
    uint256 public candidateCount;
    event votedEvent(uint256 indexed _candidateId);
    constructor() public {
        addCandidate("Candidate 1");
        addCandidate("Candidate 2");
        addCandidate("Candidate 3");
    }
    function addCandidate(string memory _name) public {
        candidateCount++;
        candidates[candidateCount] = Candidate(candidateCount, _name, 0);
    }
    function vote(uint256 _candidateId) public {
        require(!voters[msg.sender], "You already voted");
```

```
        require(_candidateId > 0 && _candidateId <= candidateCount);
        voters[msg.sender] = true;
        candidates[_candidateId].voteCount++;
        emit votedEvent(_candidateId);
    }
    function getCandidateDetails(uint _candidateId) public view returns (uint,string memory,uint)
{
        return (candidates[_candidateId].id, candidates[_candidateId].name,
candidates[_candidateId].voteCount);
    }
}
```

2_deploy_contract.js

```
var MyVoting = artifacts.require("./MyVoting.sol");
module.exports = function(deployer)
{
    deployer.deploy(MyVoting);
};
```

Output:

```
PS Z:\MCA-BlockChain\blockchain-toolkit> truffle migrate

  Compiling your contracts...
  ===========================
  > Compiling .\contracts\MyVoting.sol
  > Artifacts written to Z:\MCA-BlockChain\blockchain-toolkit\build\contracts
  > Compiled successfully using:
     - solc: 0.5.16+commit.9c3226ce.Emscripten.clang


  Starting migrations...
  ======================
  > Network name:    'development'
  > Network id:      5777
  > Block gas limit: 6721975 (0x6691b7)
```

```
BLOCKCHAIN-TOOLKIT
  build\contracts
    MyVoting.json
  contracts
    .gitkeep
    MyVoting.sol 2
  migrations
    .gitkeep
    2_deploy_contra...
  test
    .gitkeep
  package.json
  truffle-config.js
```

```
2_deploy_contracts.js
=====================

   Deploying 'MyVoting'
   --------------------
   > transaction hash:    0x7d21d248a07b49e9949672276d38666d378d0567b5ffee22ac19c559ed147c4c
   > Blocks: 0            Seconds: 0
   > contract address:    0x3dE24FCd8c5d96d91e18BEbeFBe0Dd00b862Cd61
   > block number:        20
   > block timestamp:     1730048922
   > account:             0x931Af6810a3100a3a503F774b823965CD74A30d3
   > balance:             119.70141502
   > gas used:            541747 (0x84433)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.01083494 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:          0.01083494 ETH

Summary
=======
> Total deployments:   1
> Final cost:          0.01083494 ETH


PS Z:\MCA-BlockChain\blockchain-toolkit>
```

Q.7 Build and test decentralized application, (Dapp) for Banking System on the local Ethereum Blockchain Network Ganache using truffle suite.

Code:

MyBanking.sol

```
pragma solidity >=0.5.16;

contract MyBanking {

   // State variable to store the balance

   uint256 private balance;

   // Constructor to ini alize balance

   constructor () public {

      balance = 0;

   }

   // Func on to add (deposit) amount to the balance

   function addAmount(uint256 amount) public {

      balance += amount;

   }

   // Func on to withdraw amount from the balance

   function withdrawAmount(uint256 amount) public {

      require(amount <= balance, "Insufficient balance");

      balance -= amount;

   }

   // Func on to check the remaining balance

   function checkBalance() public view returns (uint256) {

      return balance;

   }

}
```

2_deploy_contracts.js

```
const MyBanking = artifacts.require("MyBanking");

module.exports = function (deployer) {

   deployer.deploy(MyBanking);

};
```

Output:

```
EXPLORER ...                PROBLEMS 3   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   SEARCH TERMINAL OUTPUT   COMMENTS
BLOCKCHAIN-TOOLKIT
  build\contracts
    MyBanking.json
  contracts
    .gitkeep
    MyBanking.... 3
  migrations
    .gitkeep
    2_deploy_contra...
  test
    .gitkeep
  package.json
  truffle-config.js

PS Z:\MCA-BlockChain\blockchain-toolkit> truffle compile

Compiling your contracts...
===========================
> Compiling .\contracts\MyBanking.sol
> Artifacts written to Z:\MCA-BlockChain\blockchain-toolkit\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang
PS Z:\MCA-BlockChain\blockchain-toolkit> truffle migrate

Compiling your contracts...
===========================
> Compiling .\contracts\MyBanking.sol
> Artifacts written to Z:\MCA-BlockChain\blockchain-toolkit\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang
```

```
PS Z:\MCA-BlockChain\blockchain-toolkit> truffle migrate

Compiling your contracts...
===========================
> Compiling .\contracts\MyBanking.sol
> Artifacts written to Z:\MCA-BlockChain\blockchain-toolkit\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang


Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)
```

```
2_deploy_contracts.js
=====================

   Deploying 'MyBanking'
   ---------------------
   > transaction hash:    0x8095bc1367de7e2a8255d2e482884a6493e7c714d7eb7d0b409c9da49e09ac1d
   > Blocks: 0            Seconds: 0
   > contract address:    0x970876Bc8f85da45097754A49BFCA1F239cd1501
   > block number:        21
   > block timestamp:     1730050086
   > account:             0x931Af6810a3100a3a503F774b823965CD74A30d3
   > balance:             119.69902484
   > gas used:            119509 (0x1d2d5)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00239018 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00239018 ETH

Summary
=======
> Total deployments:   1
> Final cost:          0.00239018 ETH
```