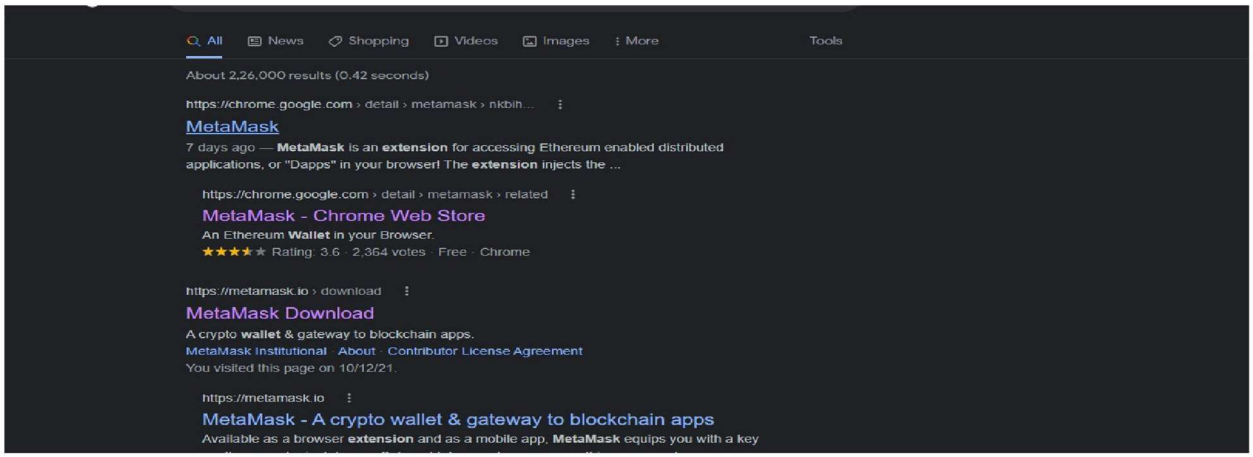
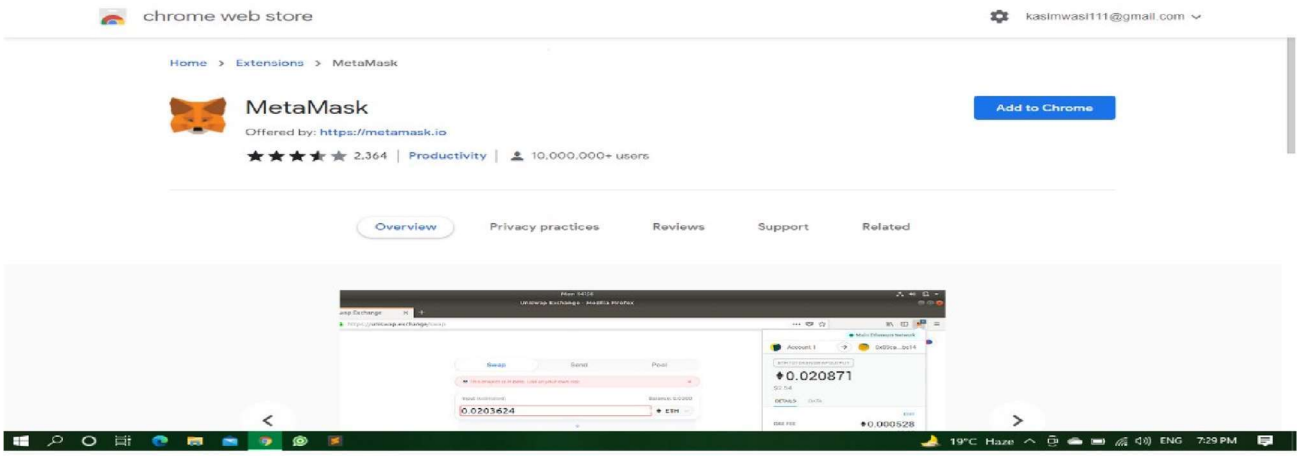
Practical 1-How to install, use and operate the MetaMask Wallet?

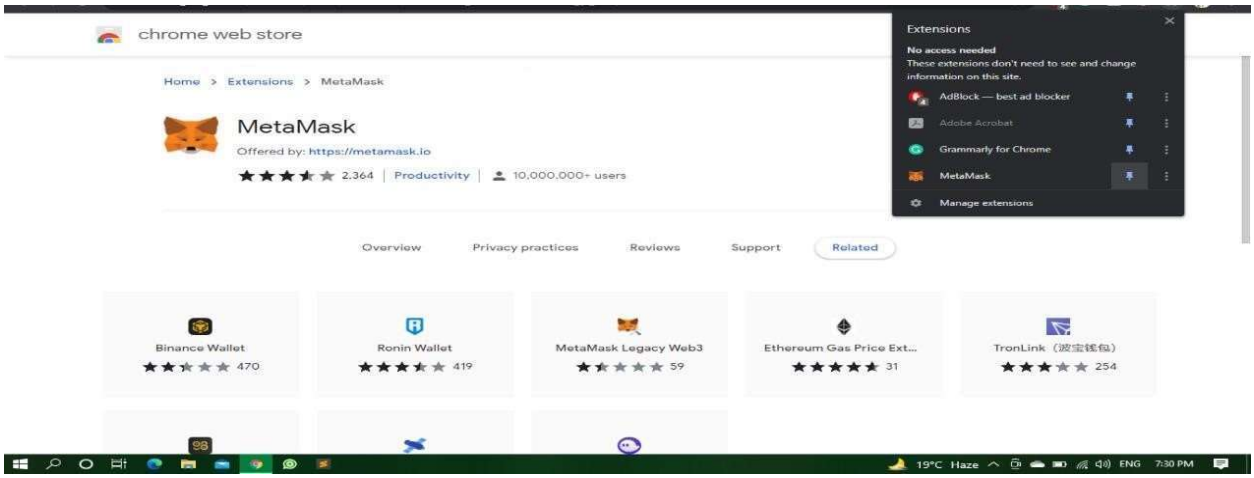
Step-1: Open Chrome Browser and search for “Metamask Wallet” and add extension download and open the first link.



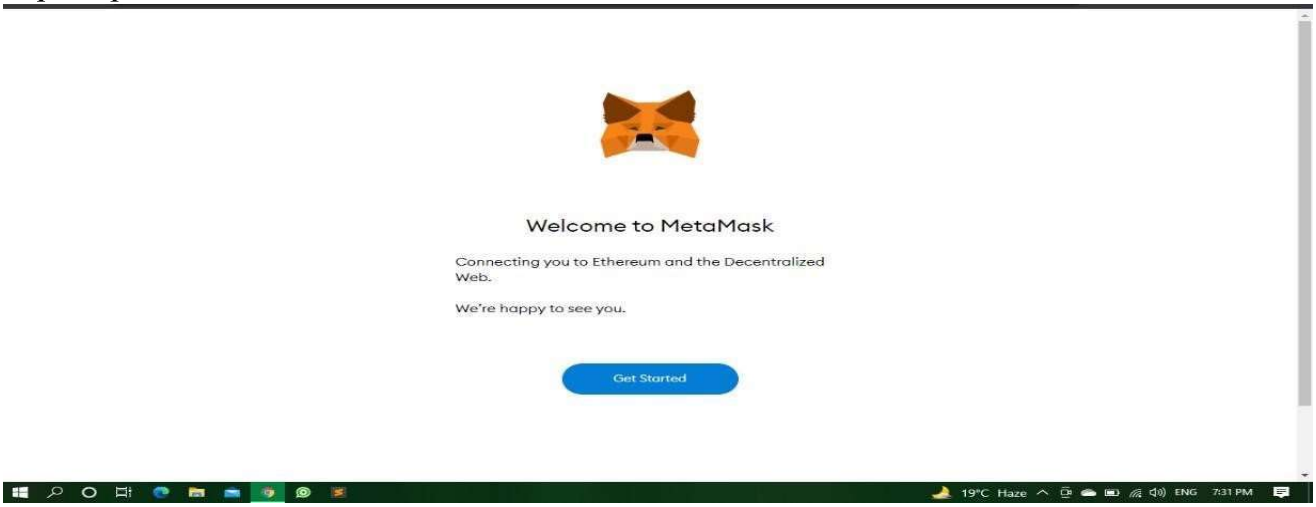
Step-2: Click on “Add to Chrome” and simply add it in your extension list.



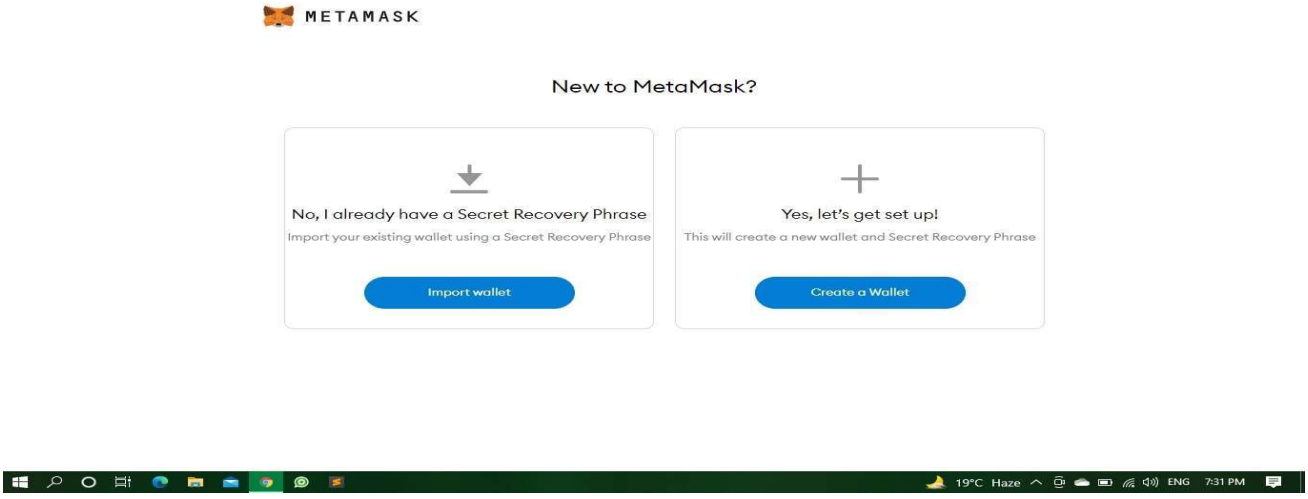
Step-3:click on extension option and pin the Metamask Wallet extension.



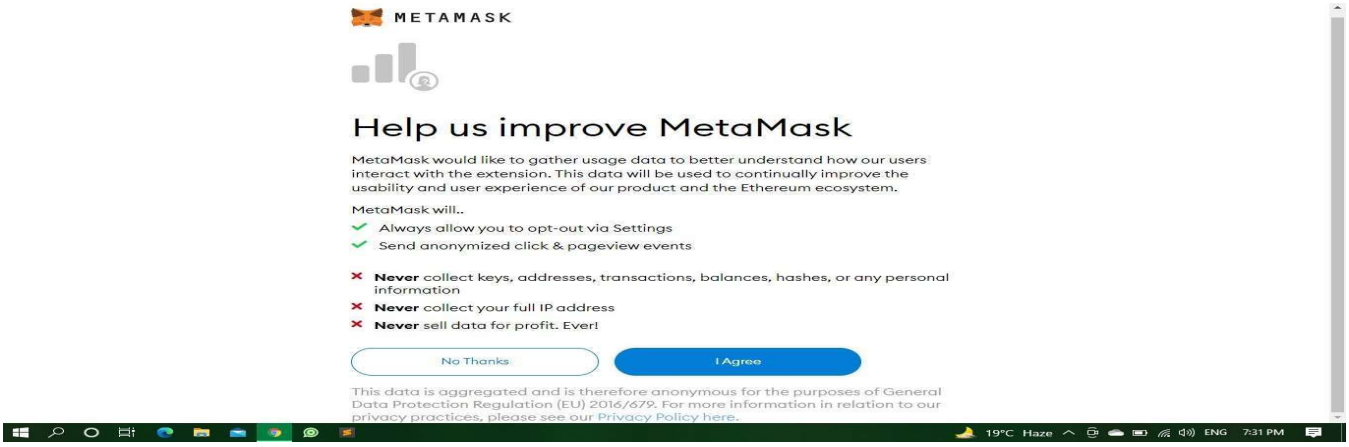
Step-4: Open Metamask Wallet and click "Get Started".



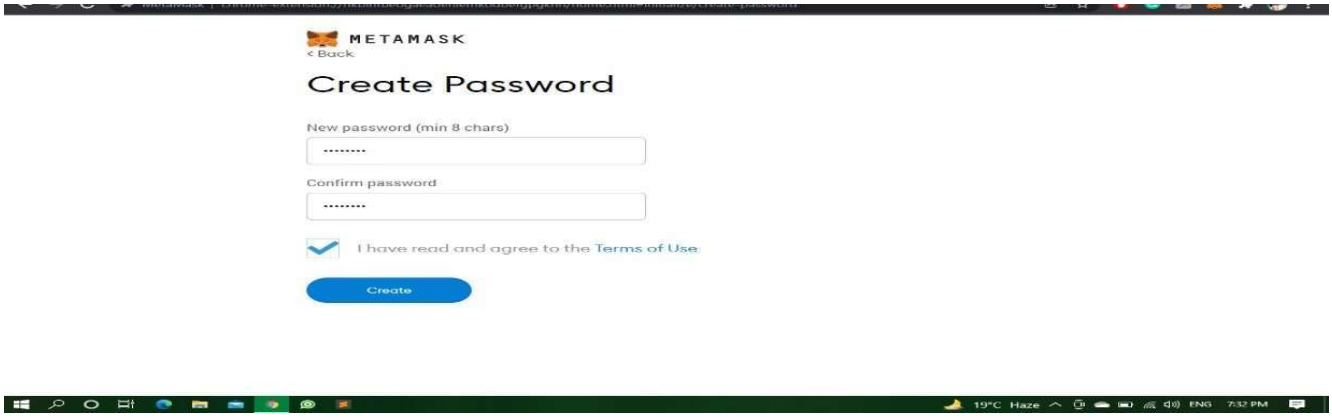
Step-5: Click on "Create a Wallet".



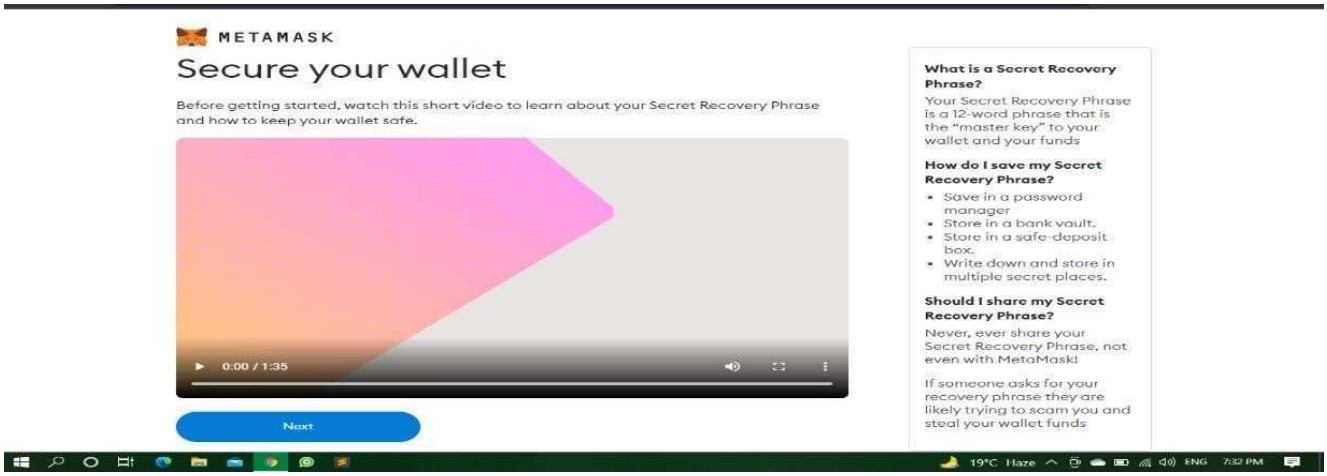
Step-6: Click on "I agree".



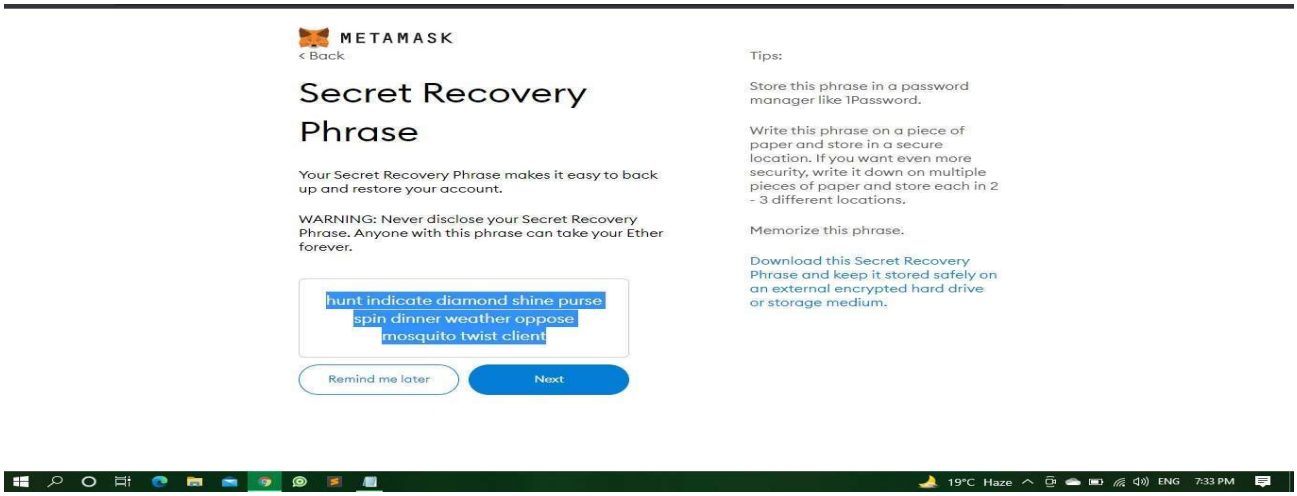
Step-7:now Create Password



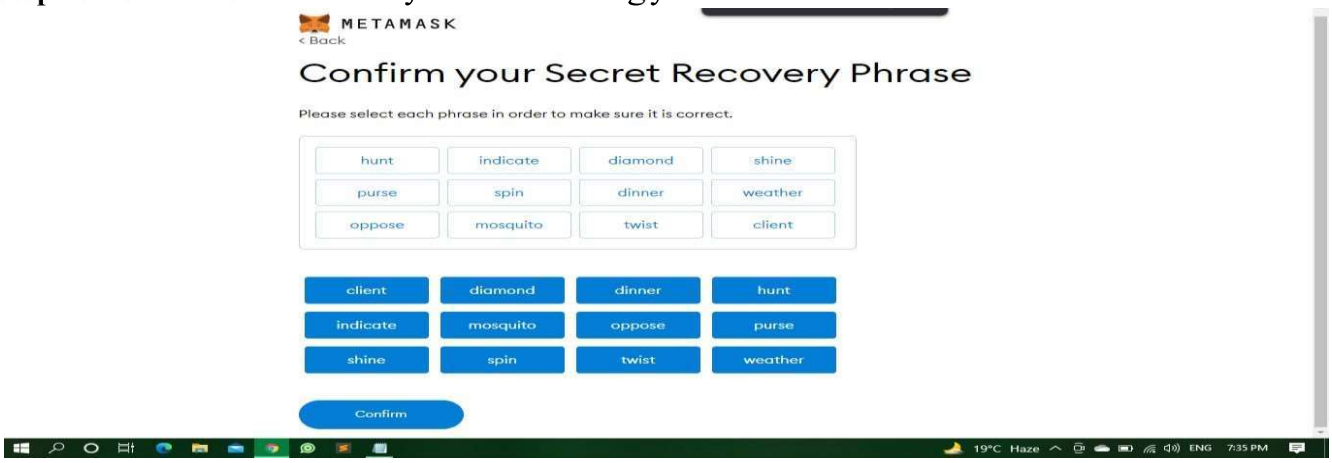
Step-8: Click on "Next" tab.



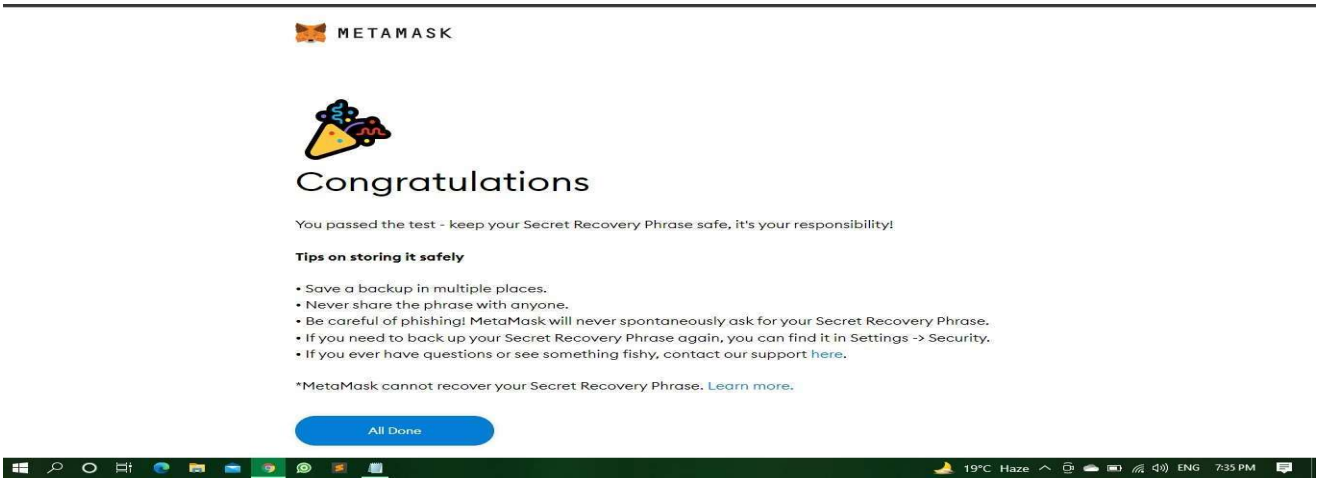
Step-9: Reveal the Secret Words then copy it and paste it in any document file and click on "Next".



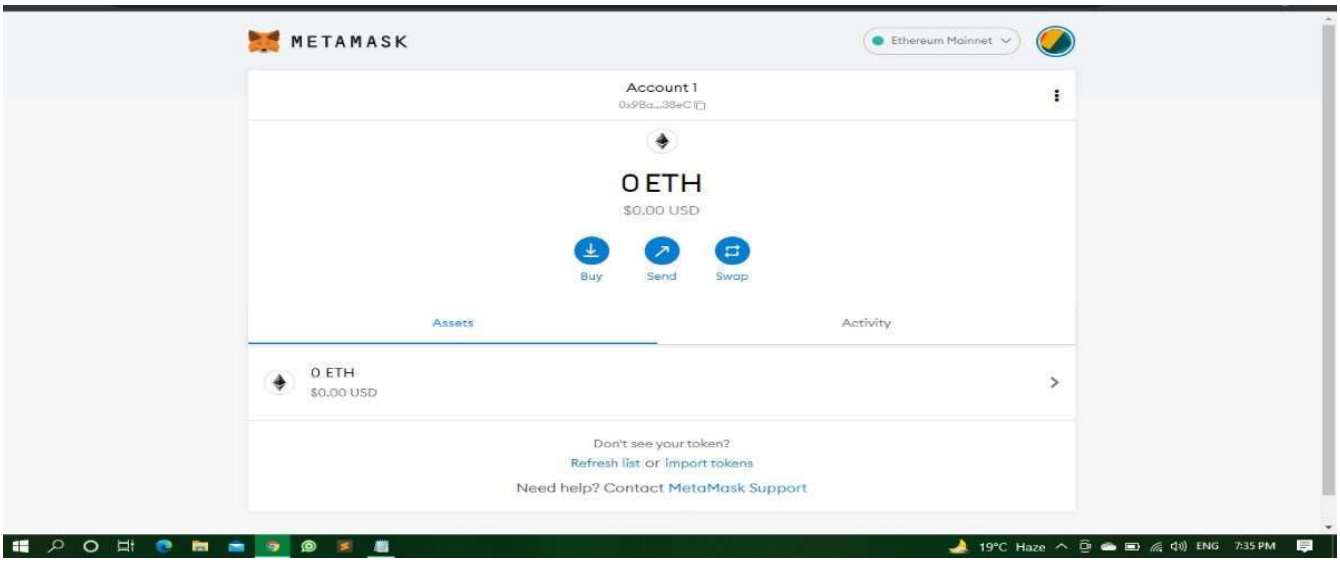
Step-10: Confirm Secret Recovery Phrase accordingly.



Step-11: After confirming the Secret Phrase simply click on "All Done", and your account has been created.

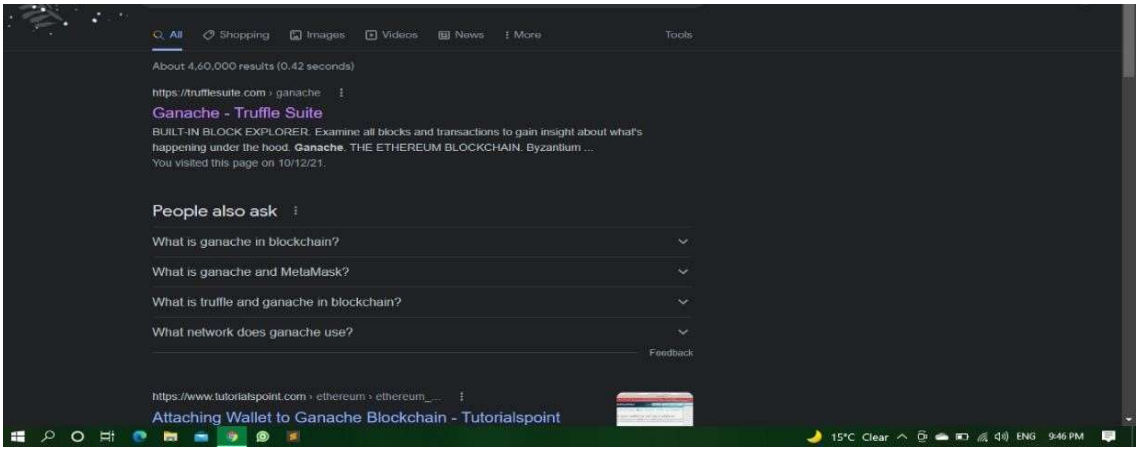


Metamask Wallet will looks like.



Practical 2- How to install, use and operate the Ganache?

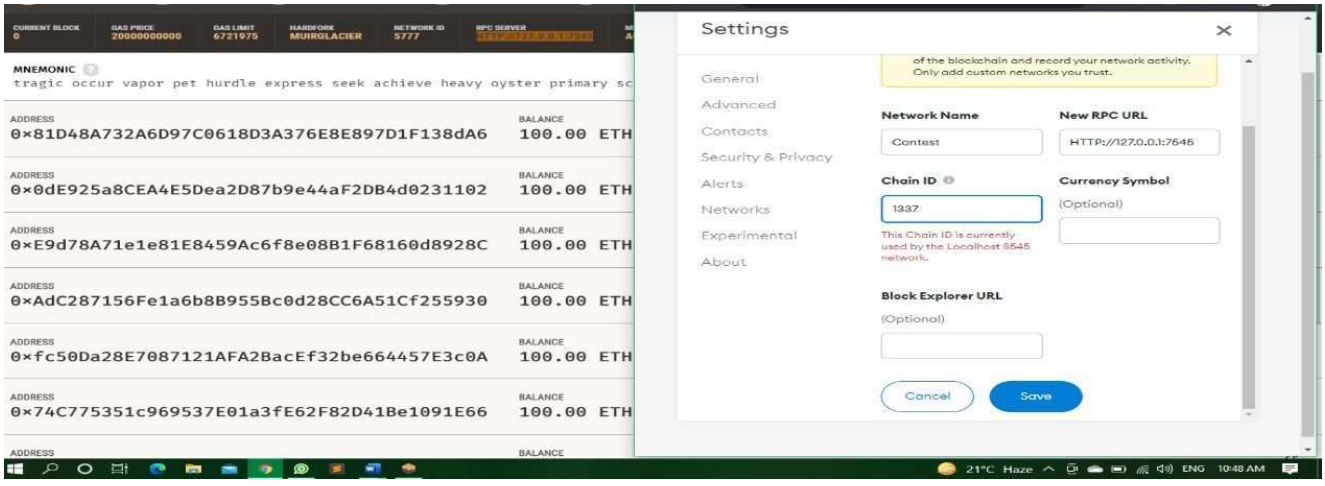
Step-1:- Open your Chrome Browser and search for Ganache Wallet and Click on first link.



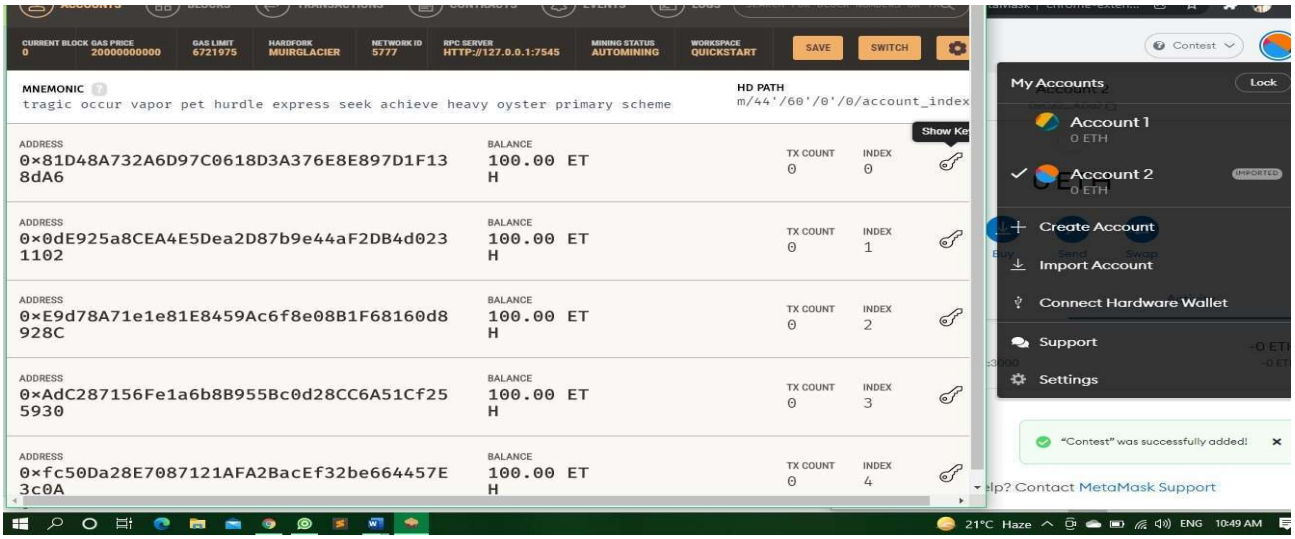
Step-2: Download Ganache for Windows.

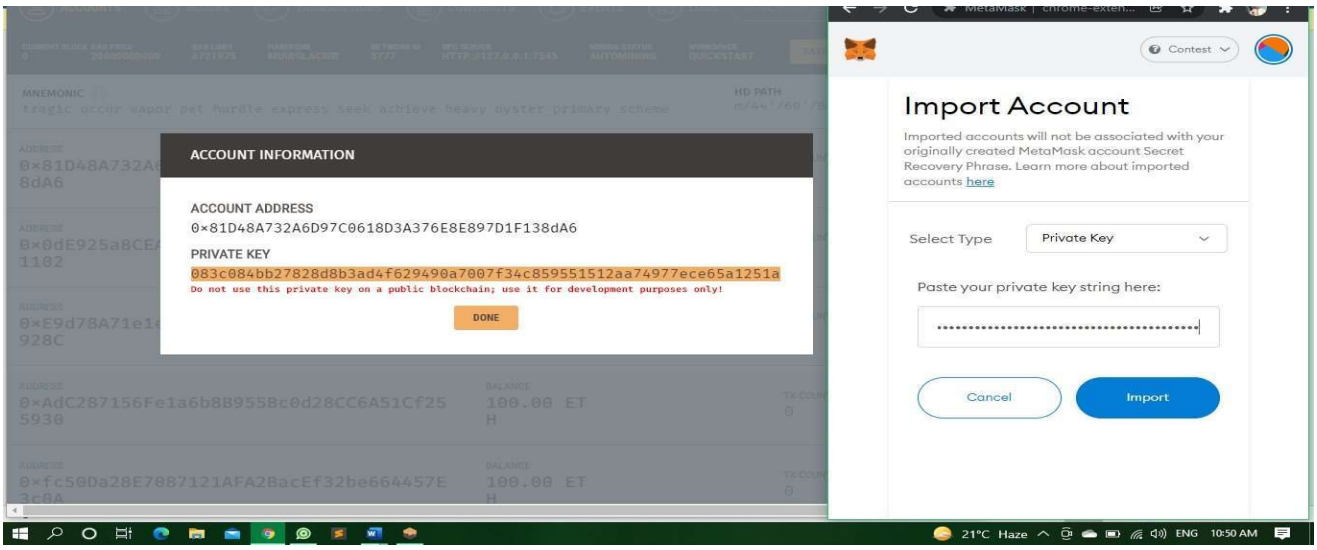
Step-3: Open ganache and click on "Quickstart"

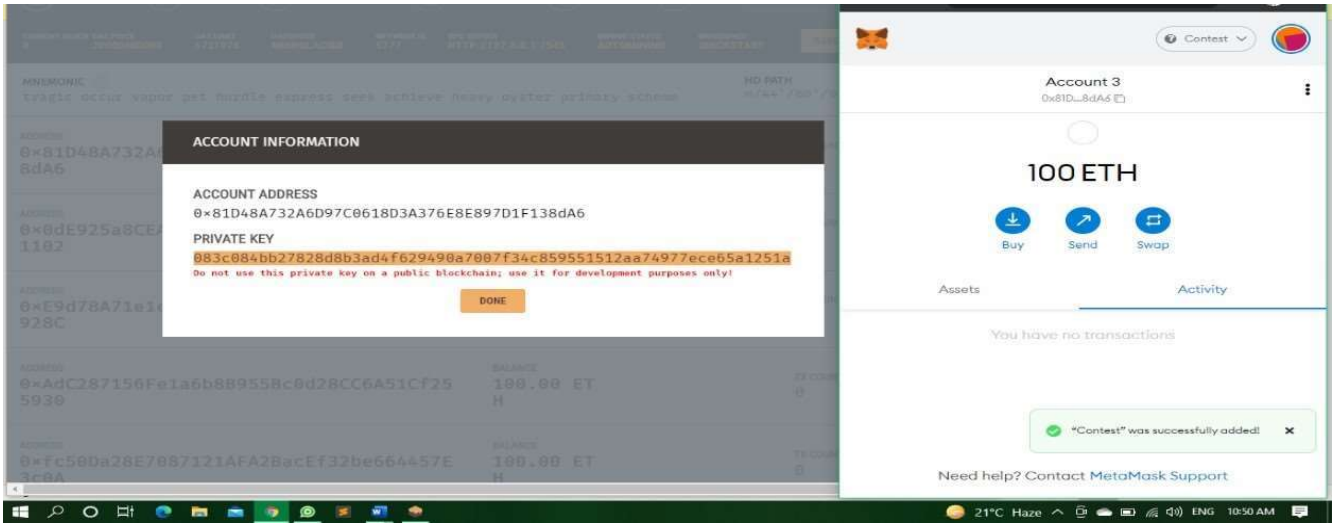
Step-4: Copy the "RPC Server" from Ganache and create a new Network in Metamask by using the RPC server.



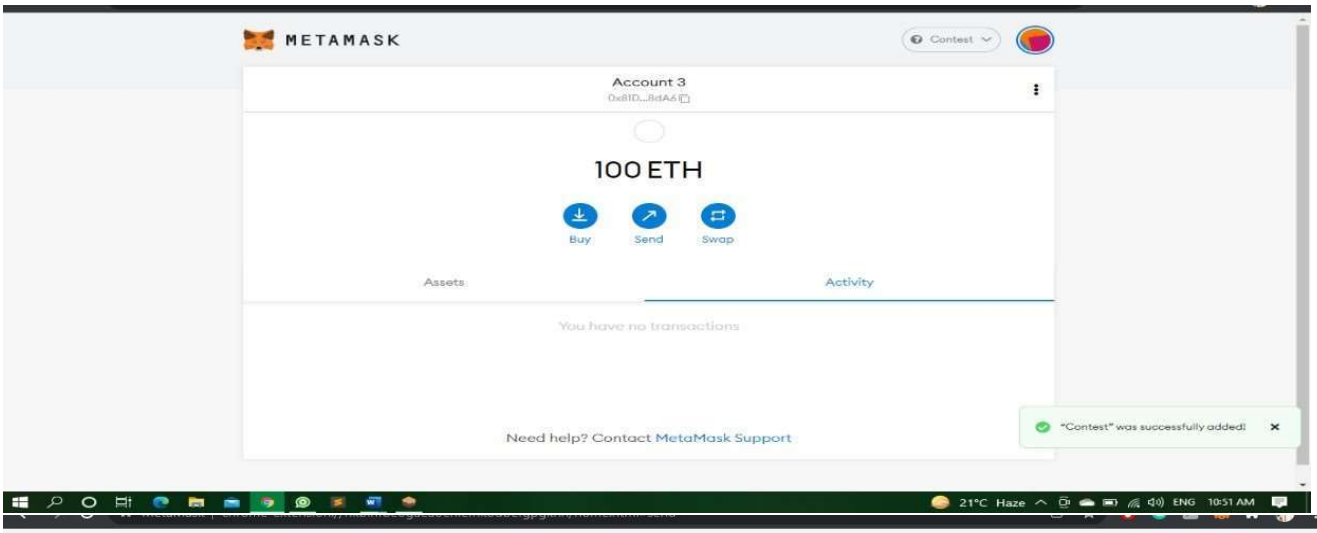
Step-5: Click on Key symbol in Ganache and Copy the Private Key and Import a new account in Metamask by using this Private key, after click on "Import" you will see 100 ETH in your account.

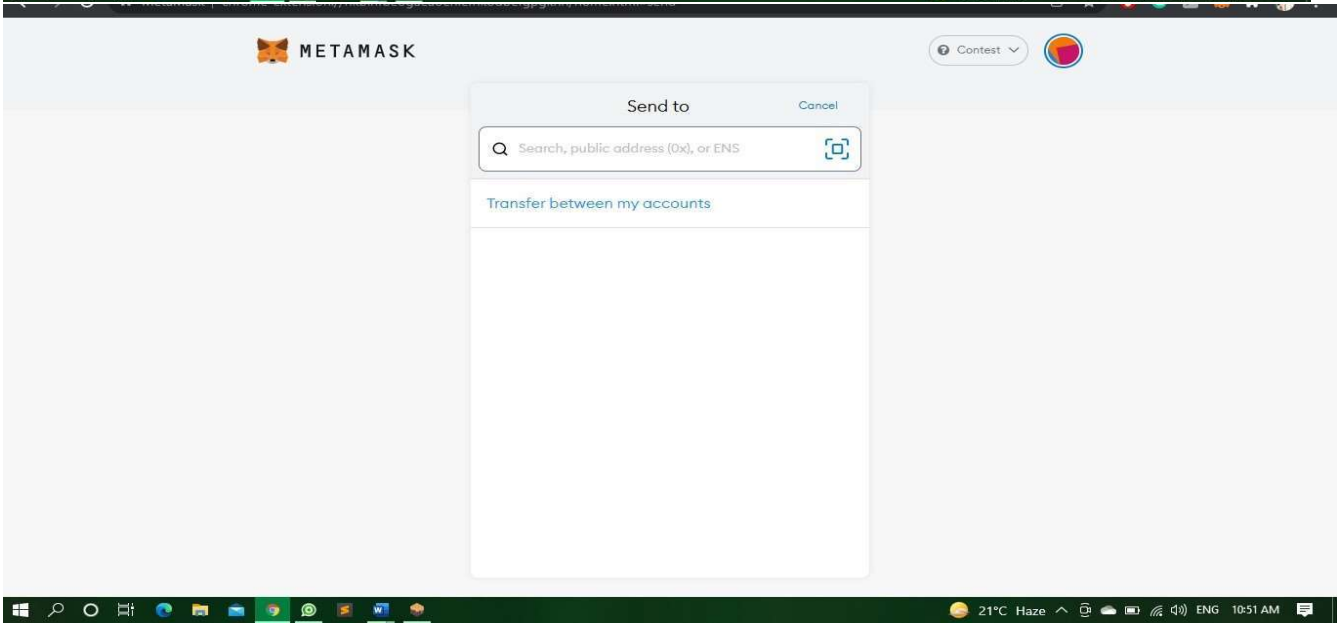


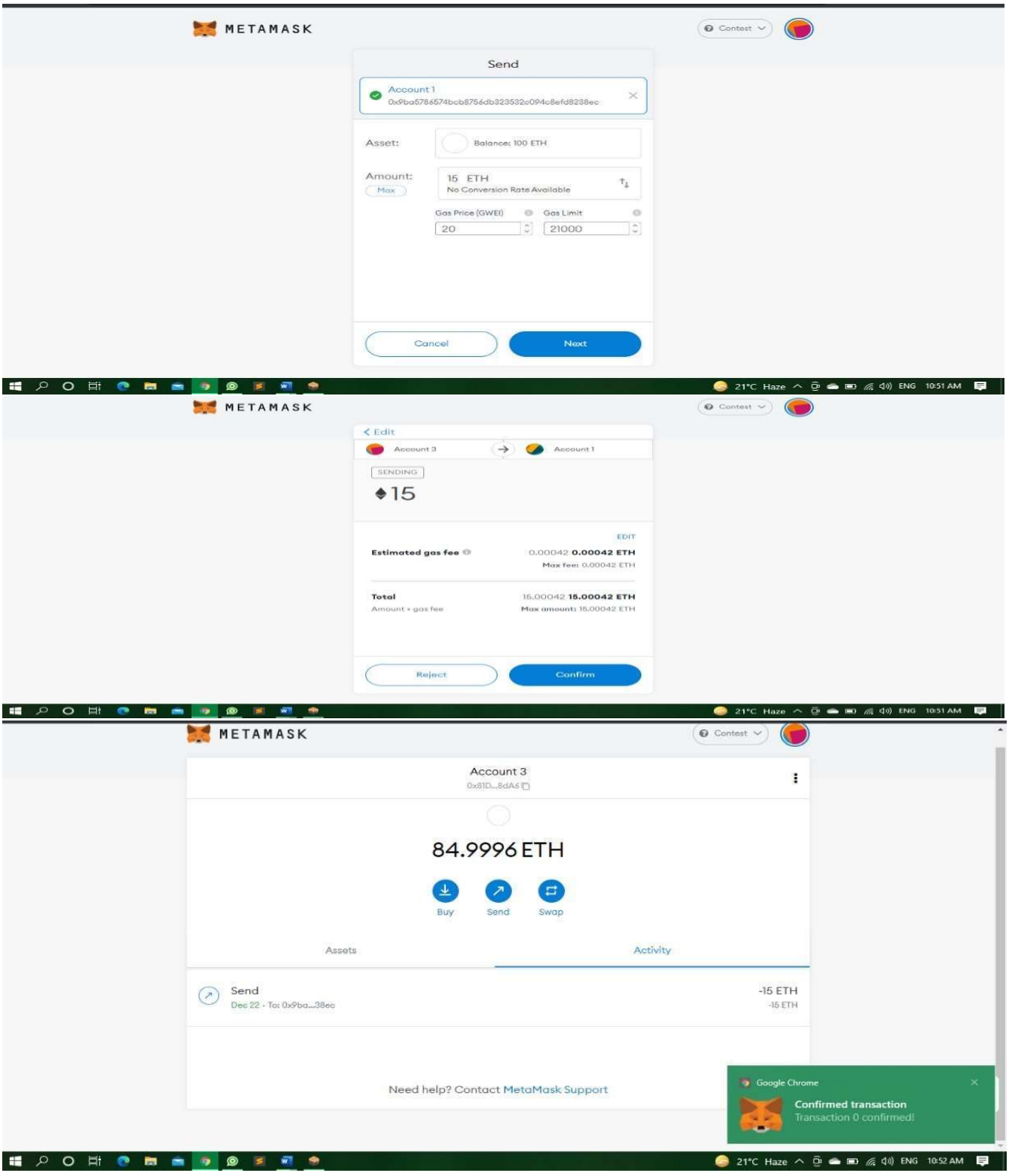




Step-6: Now click on “Send” and transfer between account and choose an account to send some ether and confirm the transaction.


METAMASK
Contest
Account 3
0x81D...8dA6
100 ETH
Buy
Send
Swap
Assets
Activity
You have no transactions
Need help? Contact MetaMask Support
"Contest" was successfully added!
21°C Haze
ENG
10:51 AM

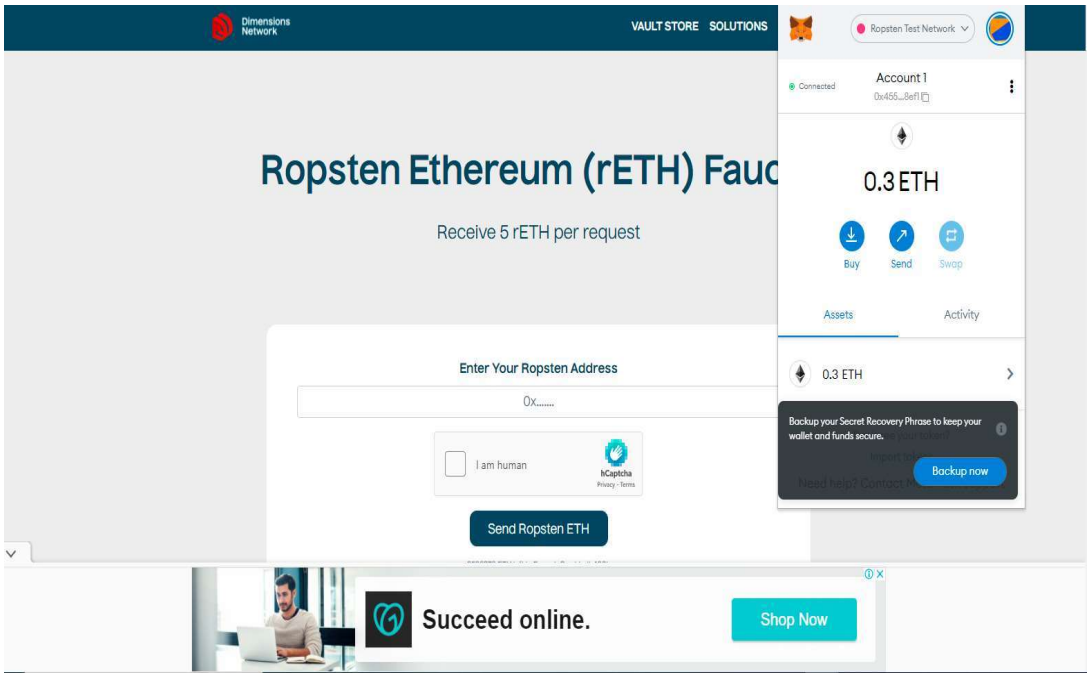

METAMASK
Contest
Send to
Cancel
Search, public address (0x), or ENS
Transfer between my accounts
21°C Haze
ENG
10:51 AM



Transaction successfully completed

PRACTICAL -3

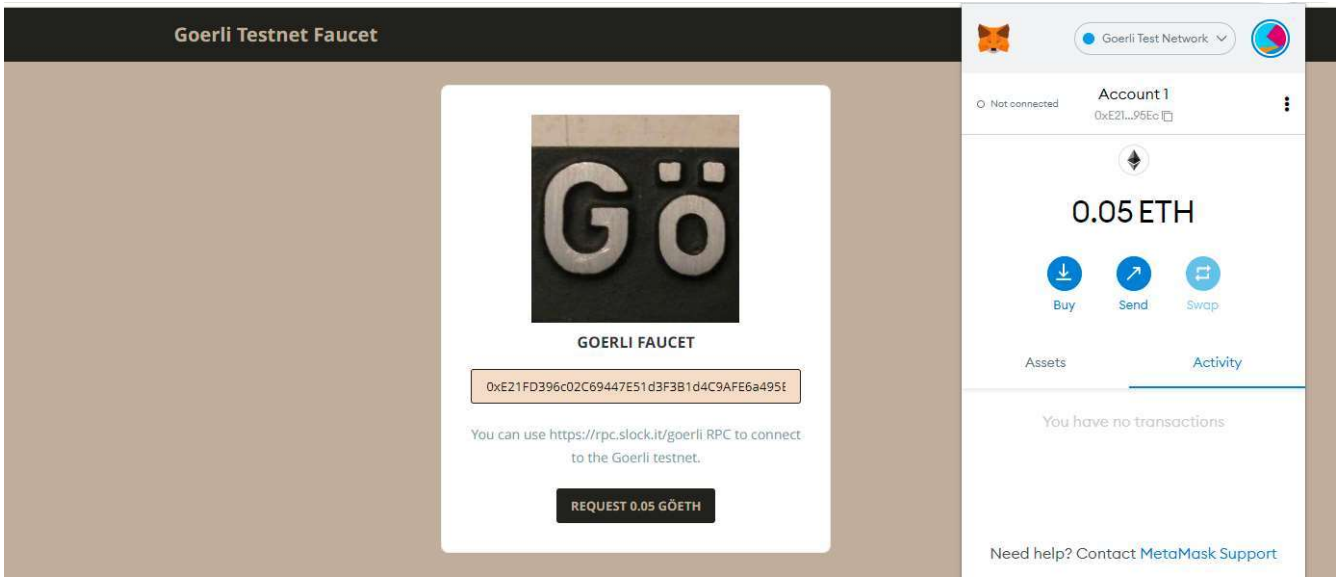
❾ Firstly open your wallet and select ropston network .After that you click on buy ETH and then



click on get ether

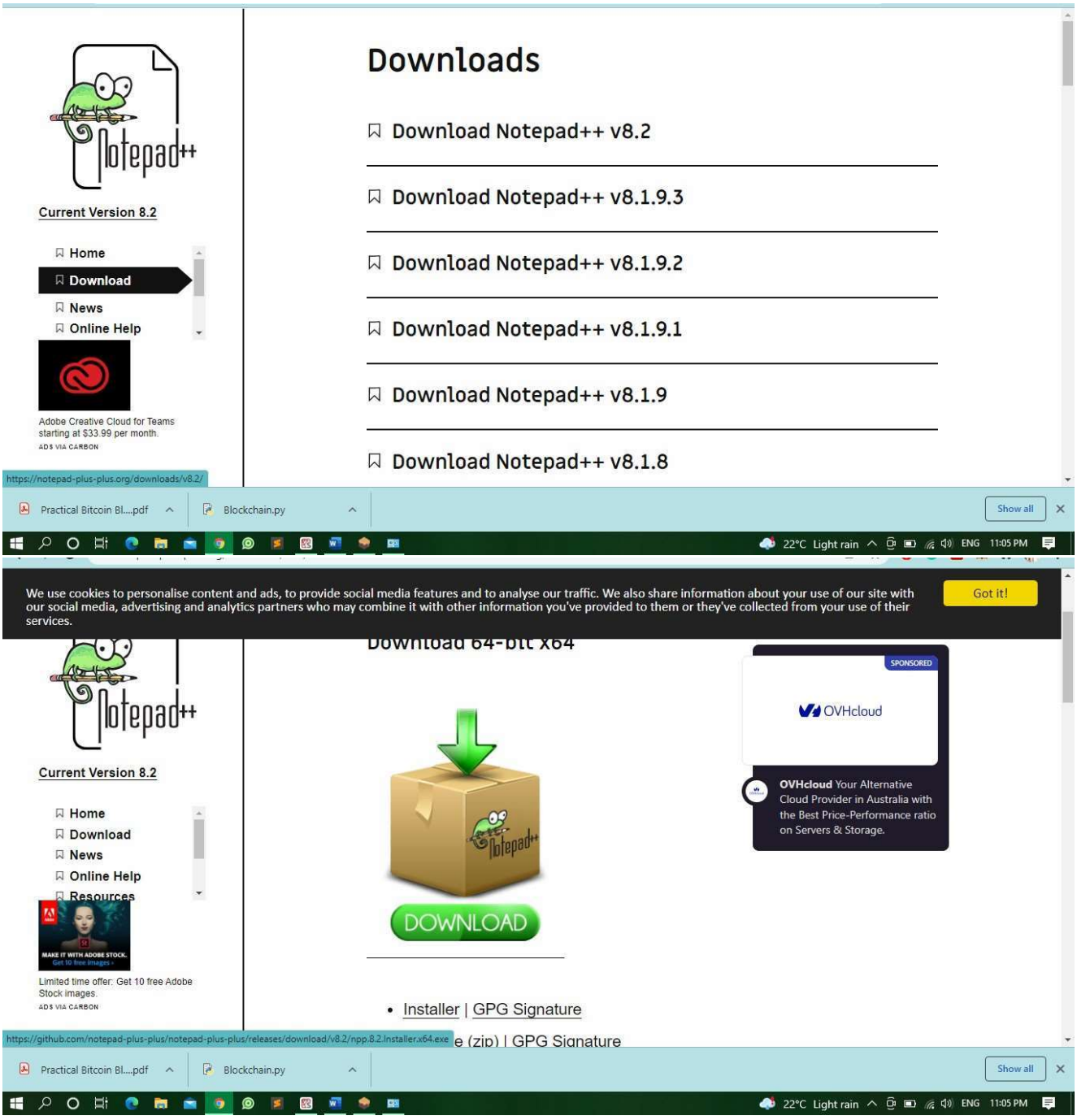
❾

→ AFTER ropston network ,open goerli network and click on buy button and after that you click on get ether



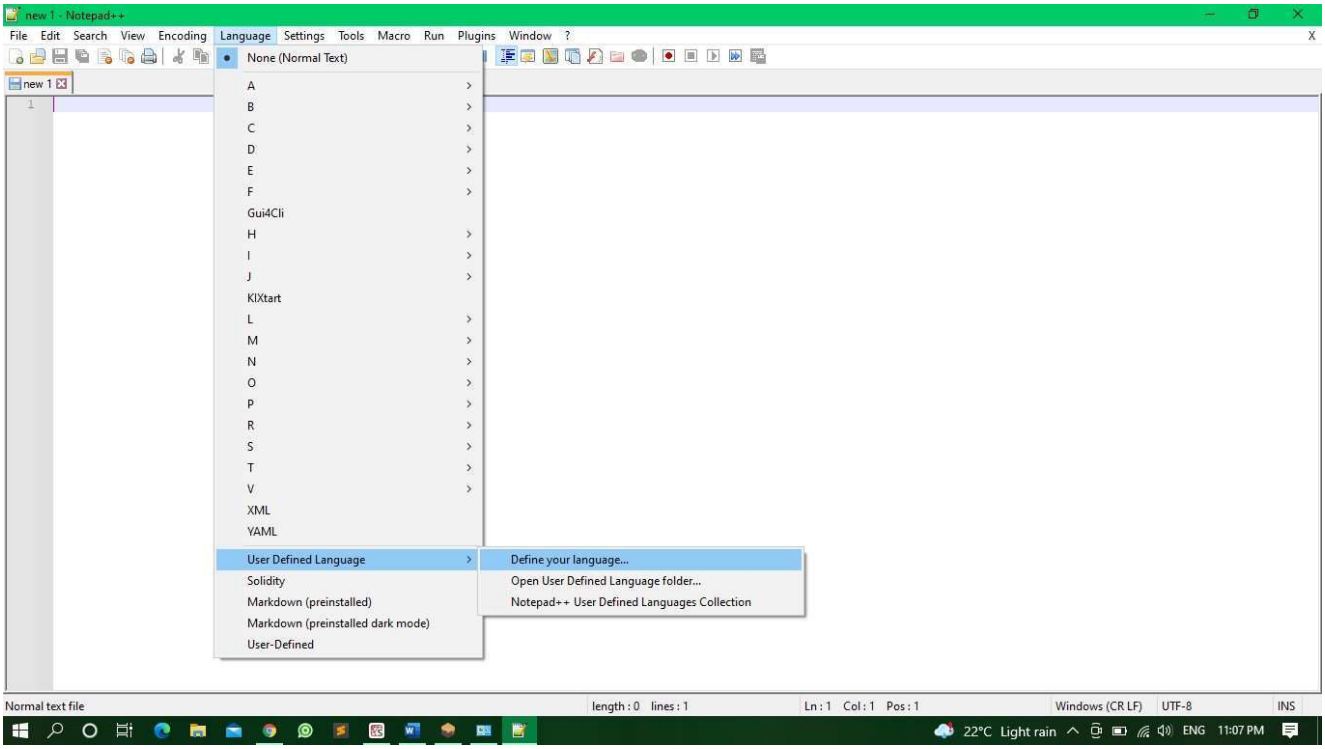
Practical No.5 Enable/Add the solidity compiler in Notepad++

Step-1: Download and Install Notepad++ .

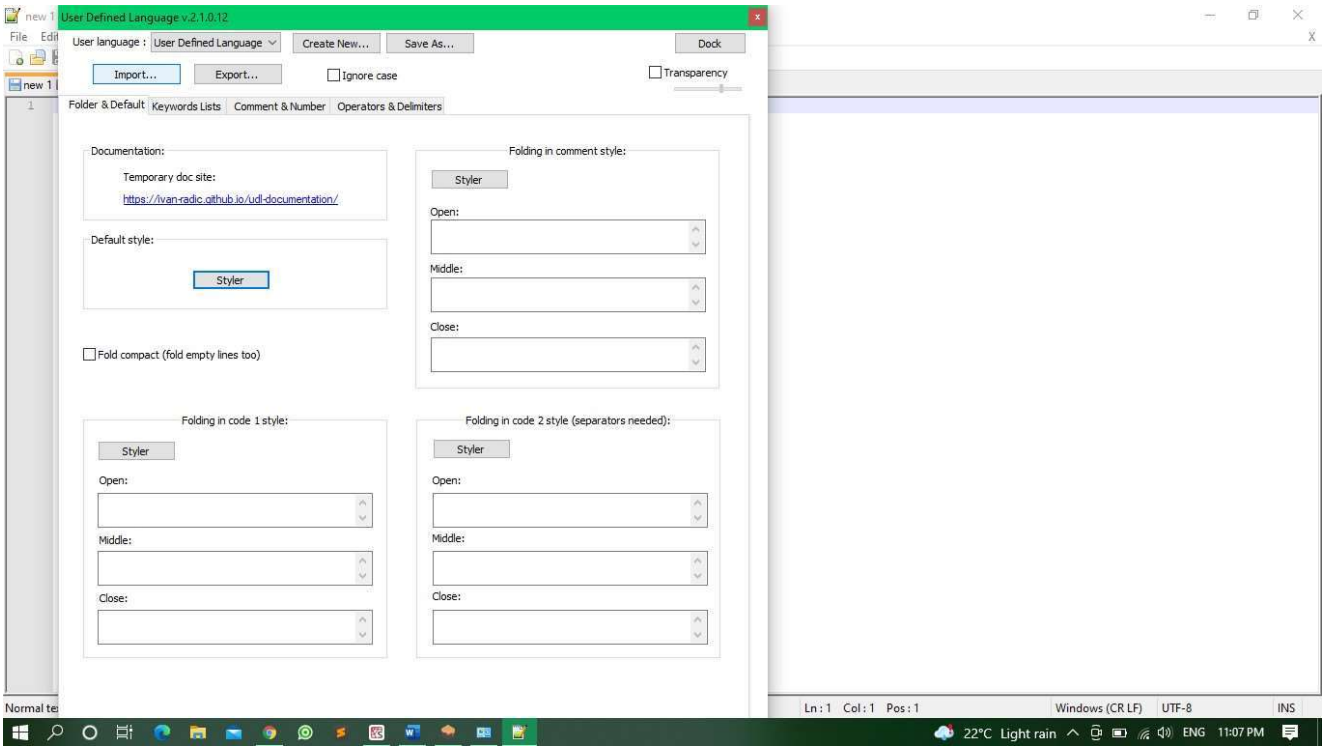


Step-2: Now open Notepad++

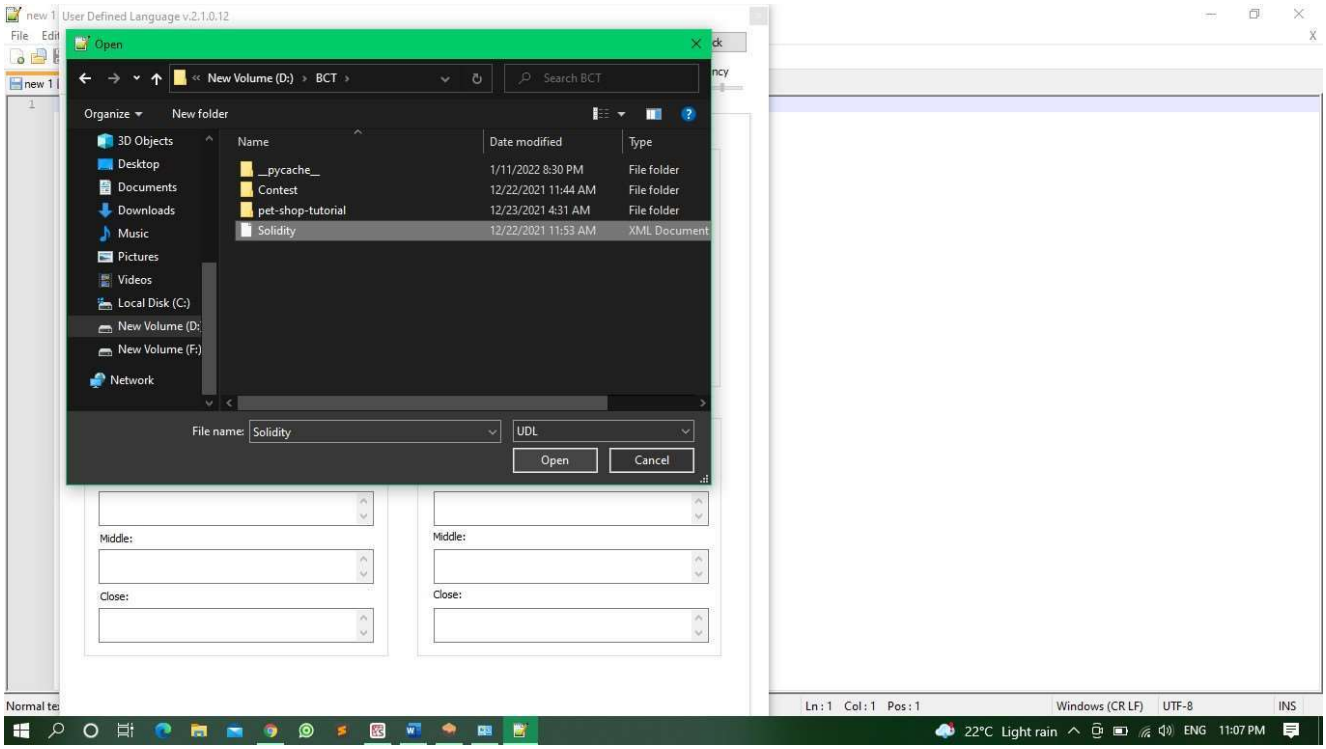
Step-3: Click on language >> User defined language >>define your language



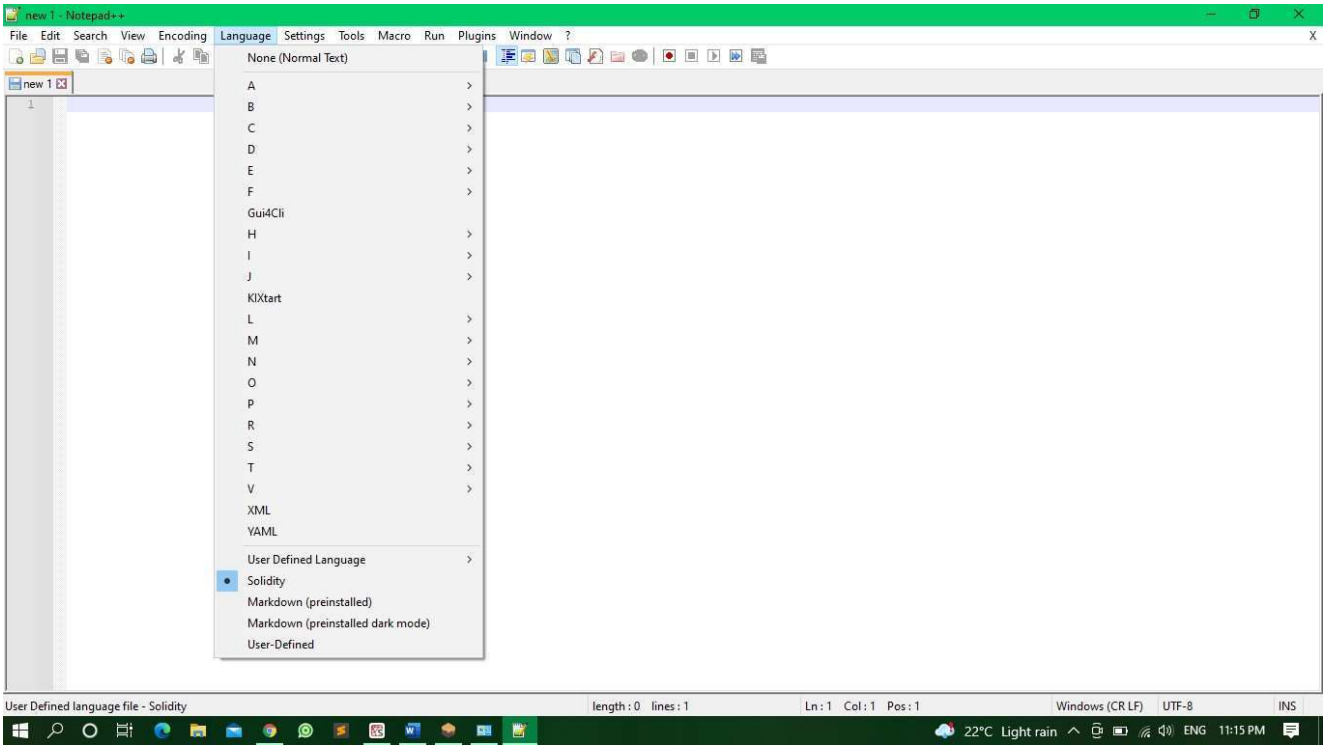
Step-4: Click on “Import”



Step-5: Click on BCT >> Solidity and open it

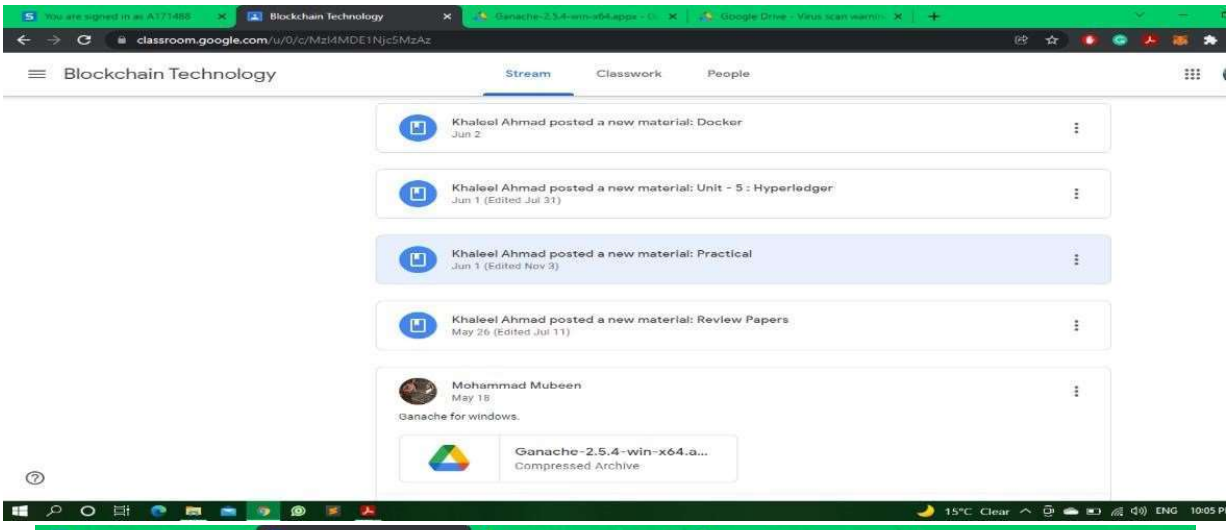


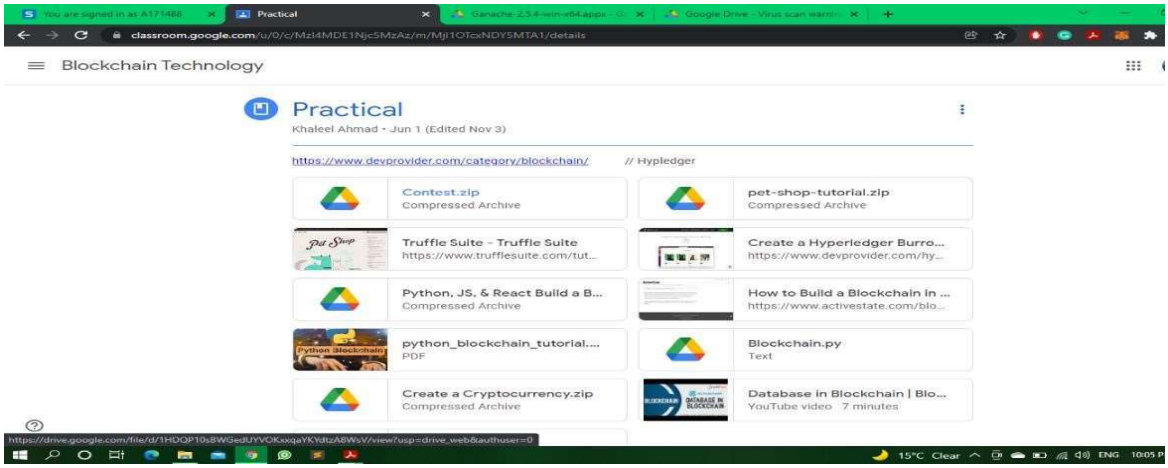
Step-6: Now you can see Solidity has been added in Notepad++.

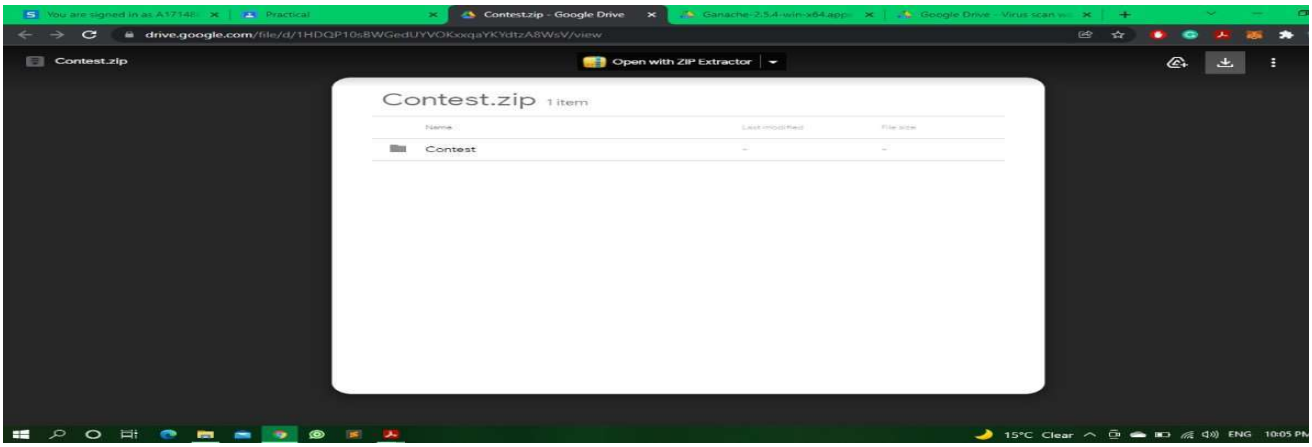


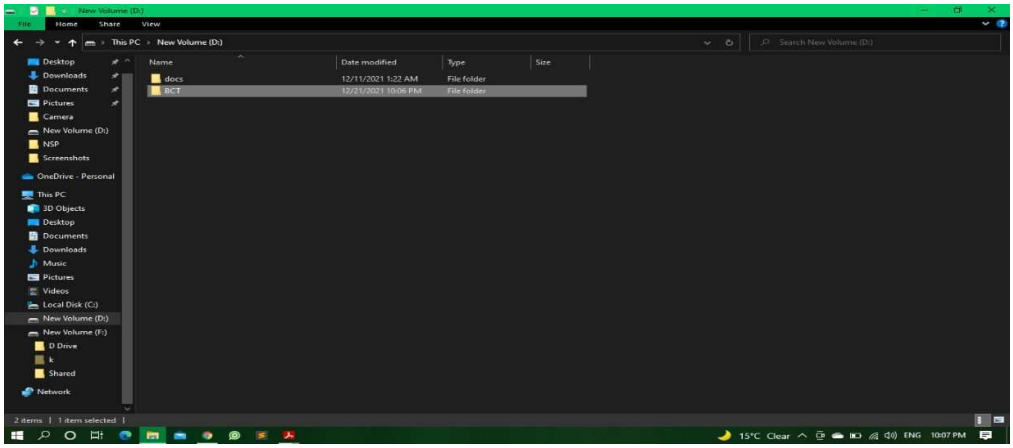
Practical No.6 E-Voting (Best Actor Contest)

Step-1: Open Google classroom of Blockchain Technology open Practical folder and download “Contest” file and extract it into a new folder called “BCT”.

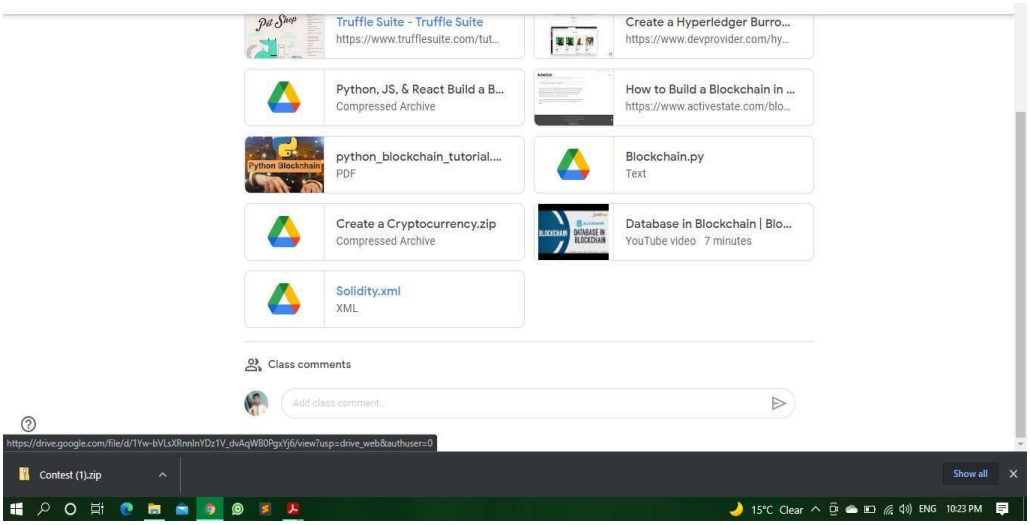








Step-2: Download Notepad++, Solidity, NodeJS and Git from google classroom by clicking on truffle suite and add Solidity in Notepad++ .



Step-3: Open cmd and run "npm install -g truffle" command.

After installation simply copy the BCT folder path and open it in cmd and run following command one by one.

>>Truffle compile

>>Truffle migrate

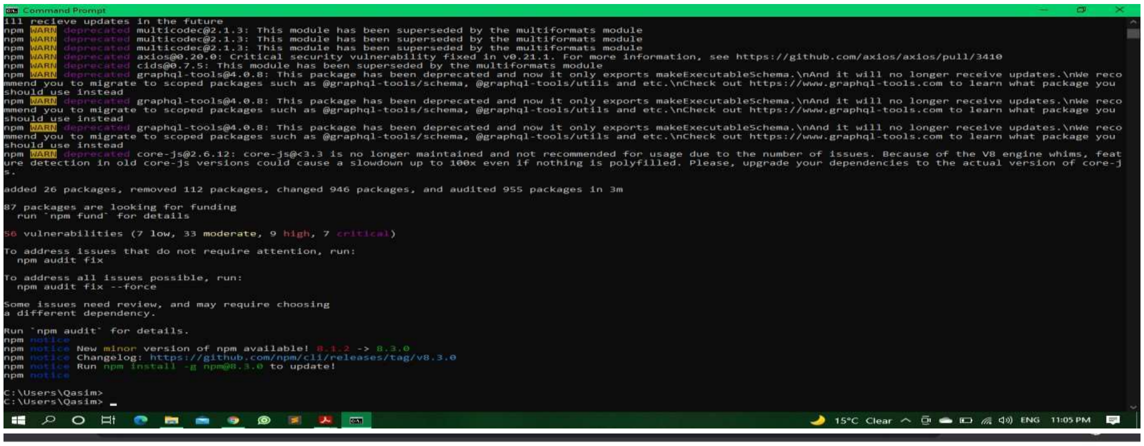
>>Truffle test

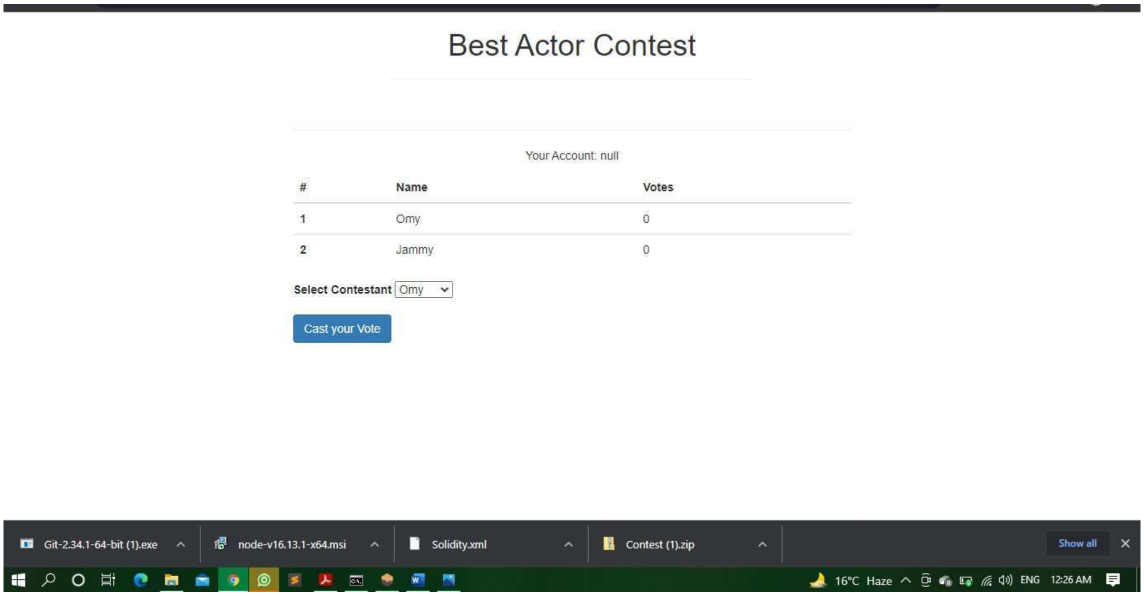
>>Npm run dev

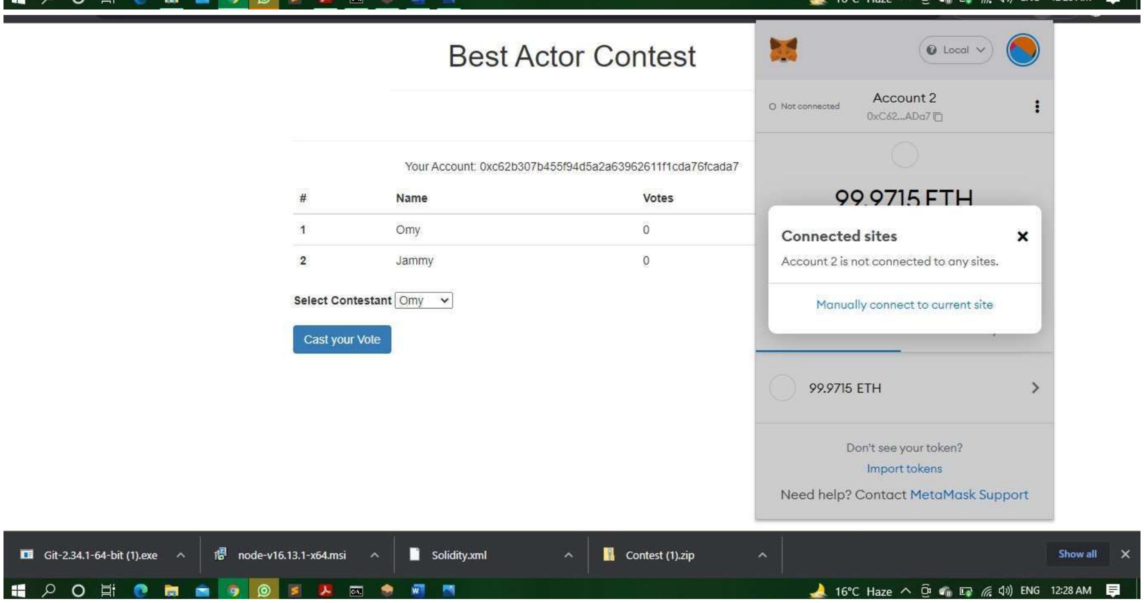
After this there will be open a Best Actor Contest web page .

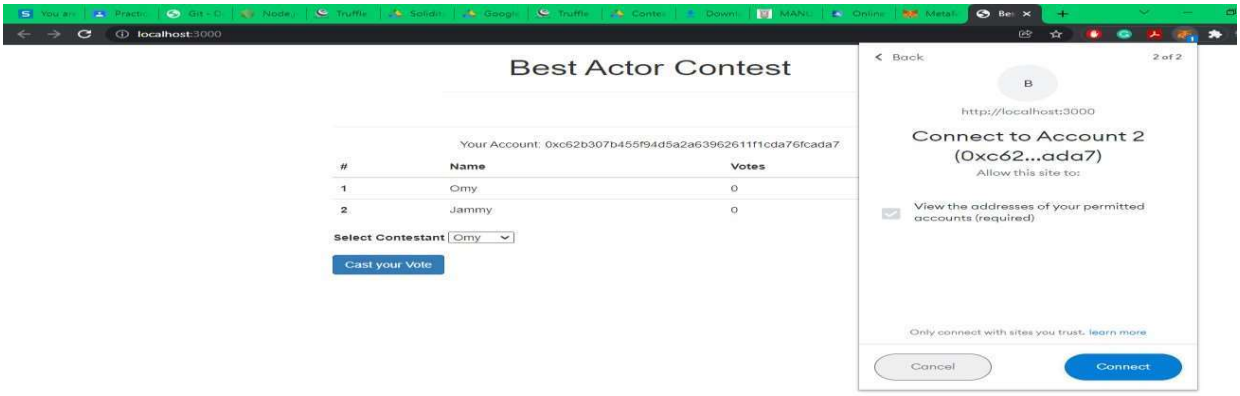
Now open metamask and connect this web page to an account which you have already imported from Ganache.

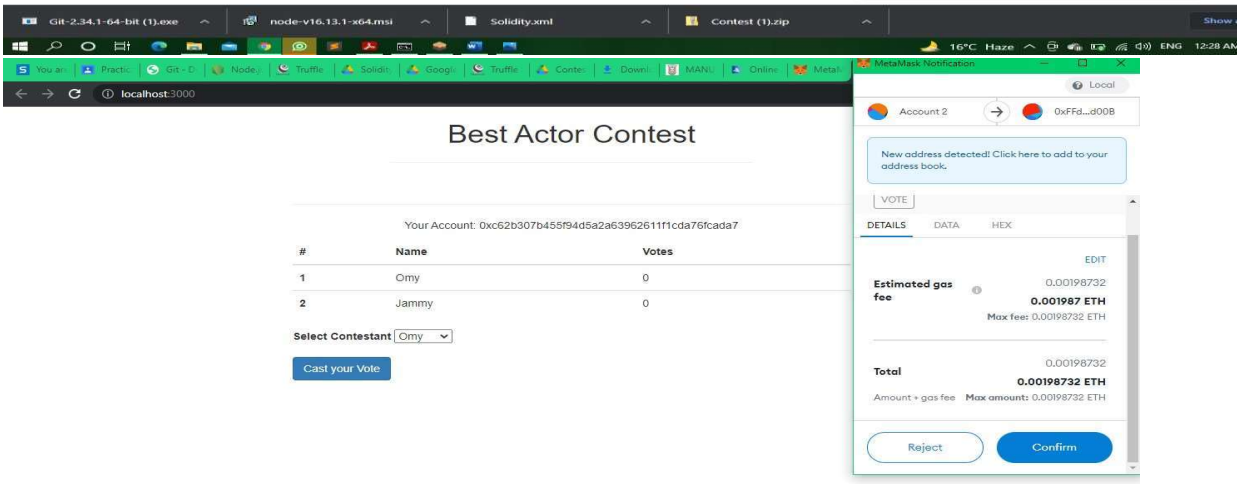
And Caste your vote by using some Ethereum.

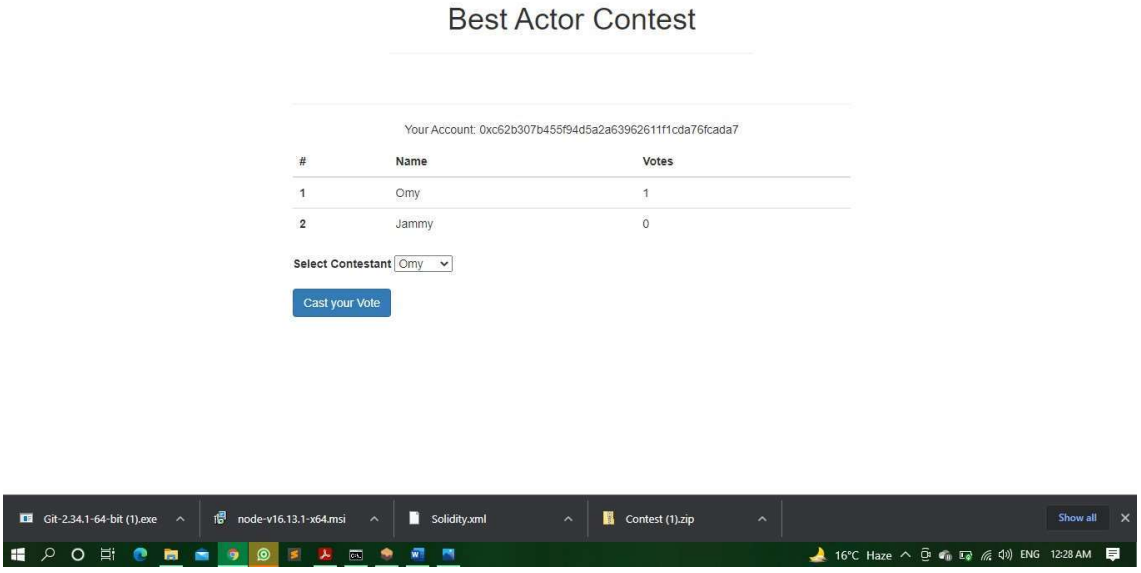

Command Prompt
ill recieve updates in the future
npm WARN deprecated multicodec@2.1.3: This module has been superseded by the multiformats module
npm WARN deprecated multicodec@2.1.3: This module has been superseded by the multiformats module
npm WARN deprecated multicodec@2.1.3: This module has been superseded by the multiformats module
npm WARN deprecated axios@0.20.0: Critical security vulnerability fixed in v0.21.1. For more information, see https://github.com/axios/axios/pull/3410
npm WARN deprecated cids@0.7.5: This module has been superseded by the multiformats module
npm WARN deprecated graphql-tools@4.0.8: This package has been deprecated and now it only exports makeExecutableSchema.\nAnd it will no longer receive updates.\nWe reco
mmend you to migrate to scoped packages such as @graphql-tools/schema, @graphql-tools/utils and etc.\nCheck out https://www.graphql-tools.com to learn what package you
should use instead
npm WARN deprecated graphql-tools@4.0.8: This package has been deprecated and now it only exports makeExecutableSchema.\nAnd it will no longer receive updates.\nWe reco
mmend you to migrate to scoped packages such as @graphql-tools/schema, @graphql-tools/utils and etc.\nCheck out https://www.graphql-tools.com to learn what package you
should use instead
npm WARN deprecated graphql-tools@4.0.8: This package has been deprecated and now it only exports makeExecutableSchema.\nAnd it will no longer receive updates.\nWe reco
mmend you to migrate to scoped packages such as @graphql-tools/schema, @graphql-tools/utils and etc.\nCheck out https://www.graphql-tools.com to learn what package you
should use instead
npm WARN deprecated core-js@2.6.12: core-js@<3.3 is no longer maintained and not recommended for usage due to the number of issues. Because of the V8 engine whims, feat
ure detection in old core-js versions could cause a slowdown up to 100x even if nothing is polyfilled. Please, upgrade your dependencies to the actual version of core-j
s.
added 26 packages, removed 112 packages, changed 946 packages, and audited 955 packages in 3m
87 packages are looking for funding
run `npm fund` for details
56 vulnerabilities (7 low, 33 moderate, 9 high, 7 critical)
To address issues that do not require attention, run:
npm audit fix
To address all issues possible, run:
npm audit fix --force
Some issues need review, and may require choosing
a different dependency.
Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 8.1.2 -> 8.3.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.3.0
npm notice Run npm install -g npm@8.3.0 to update!
npm notice
C:\Users\Qasim>
C:\Users\Qasim>
15°C Clear
ENG
11:05 PM


Best Actor Contest
Your Account: null
#
Name
Votes
1
Omy
0
2
Jammy
0
Select Contestant
Omy
Cast your Vote
Git-2.34.1-64-bit (1).exe
node-v16.13.1-x64.msi
Solidity.xml
Contest (1).zip
Show all
16°C Haze
ENG
12:26 AM


Best Actor Contest
Your Account: 0xc62b307b455f94d5a2a63962611f1cda76fcada7
#
Name
Votes
1
Omy
0
2
Jammy
0
Select Contestant
Omy
Cast your Vote
Local
Not connected
Account 2
0xC62...ADa7
99.9715 ETH
Connected sites
Account 2 is not connected to any sites.
Manually connect to current site
99.9715 ETH
Don't see your token?
Import tokens
Need help? Contact MetaMask Support
Git-2.34.1-64-bit (1).exe
node-v16.13.1-x64.msi
Solidity.xml
Contest (1).zip
Show all
16°C Haze
ENG
12:28 AM







Your E-voting has been done Successfully.

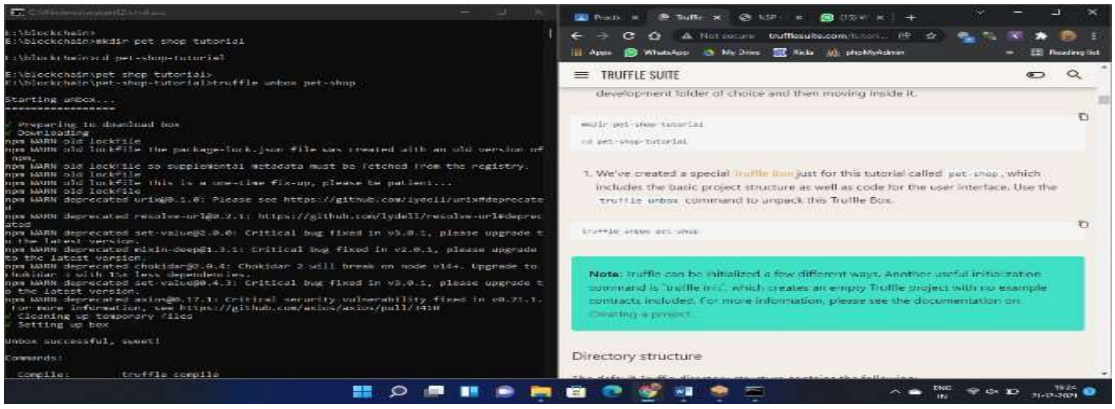
Practical No.7
Pet-shop Tutorial

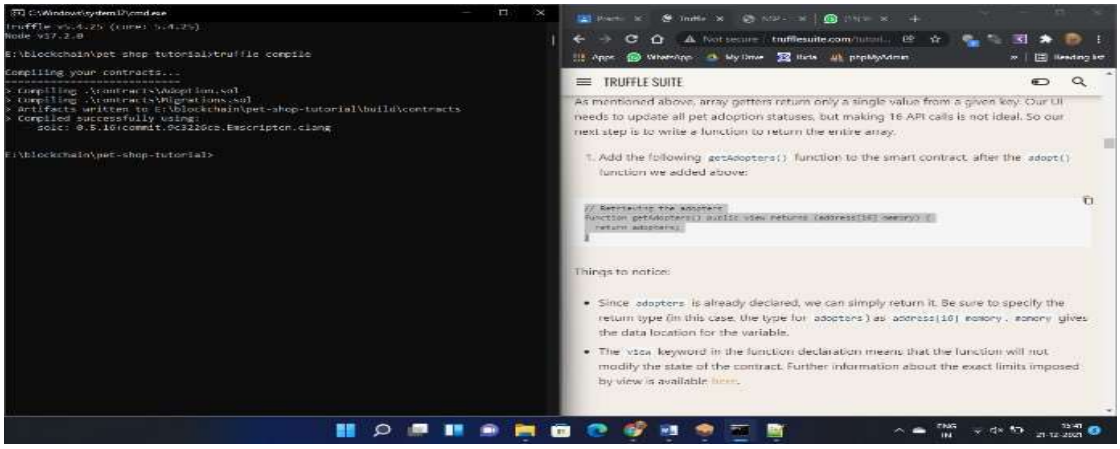
Step-1: Open cmd and run the following commands one by one,

>> mkdir pet-shop-tutorial

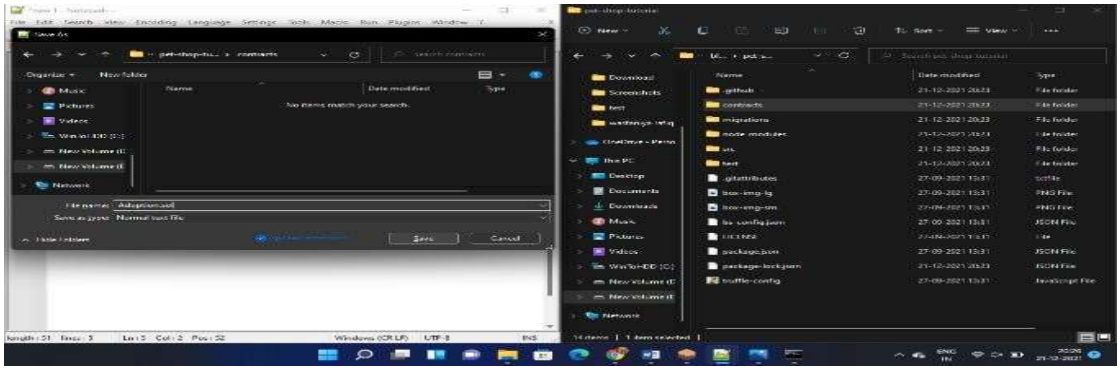
>> cd pet-shop-tutorial

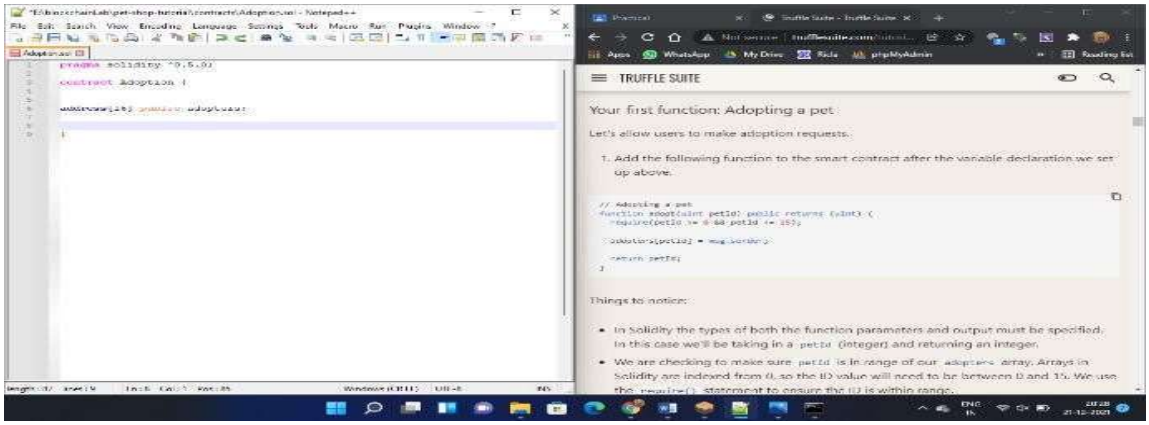
>> truffle unbox pet-shop





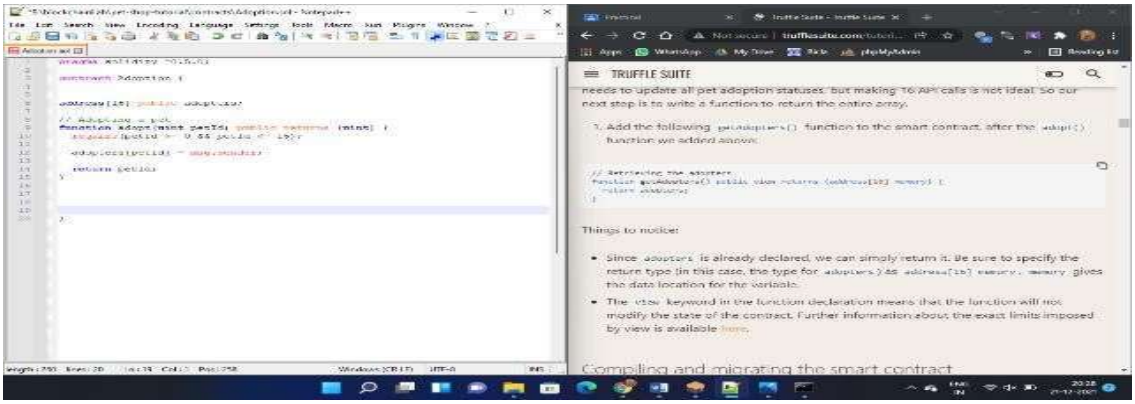
Step-2: Create a new file named Adoption.sol in the contracts/ directory, and add the following files.

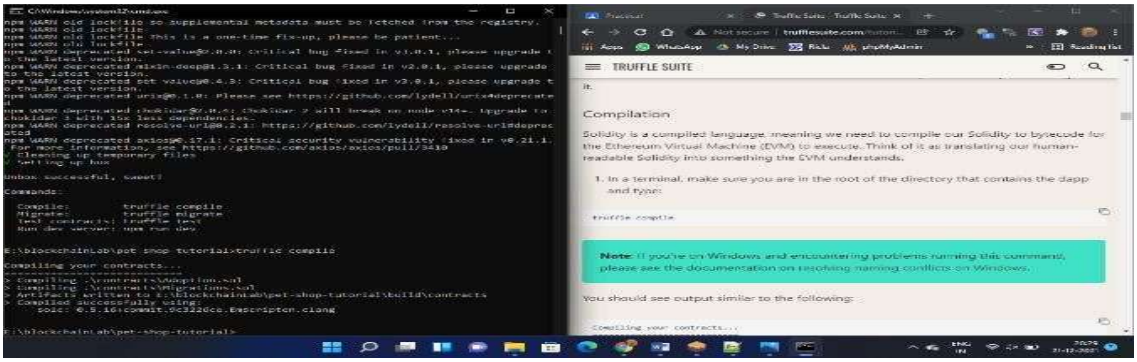




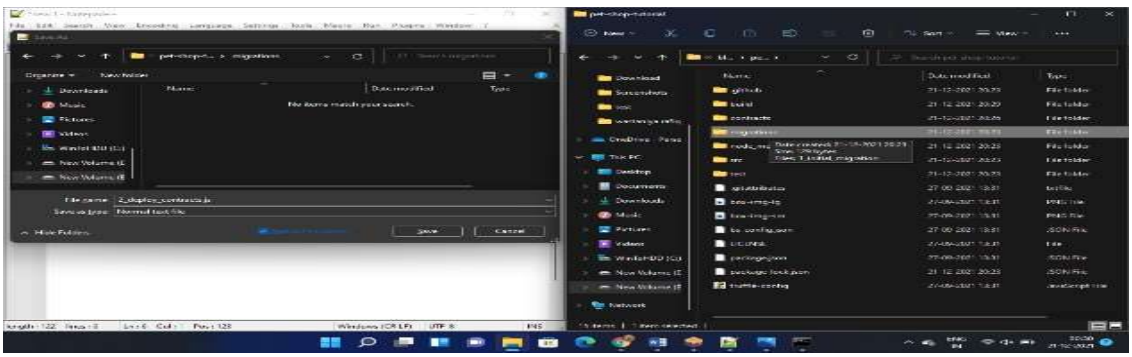
Step-3: Copy code and paste and run this command in

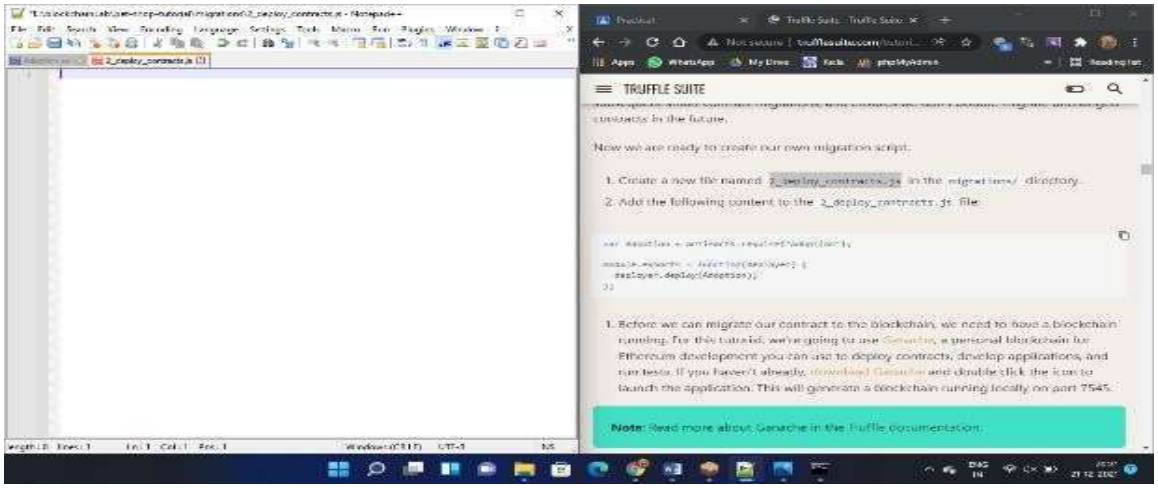
cmd >> truffle compile





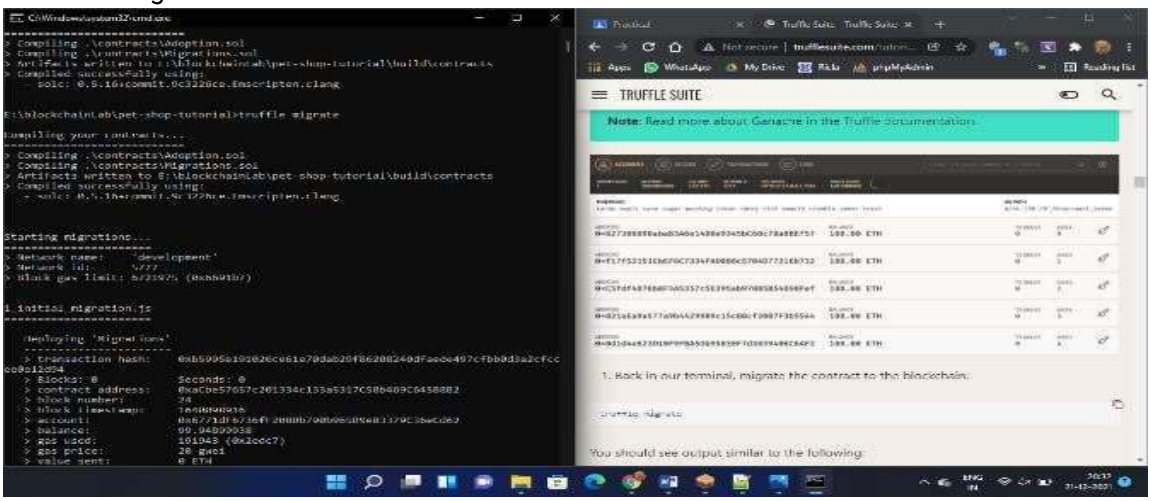
Step-4: Create a new file named 2_deploy_contracts.js in migrations/ directory and write the codes.

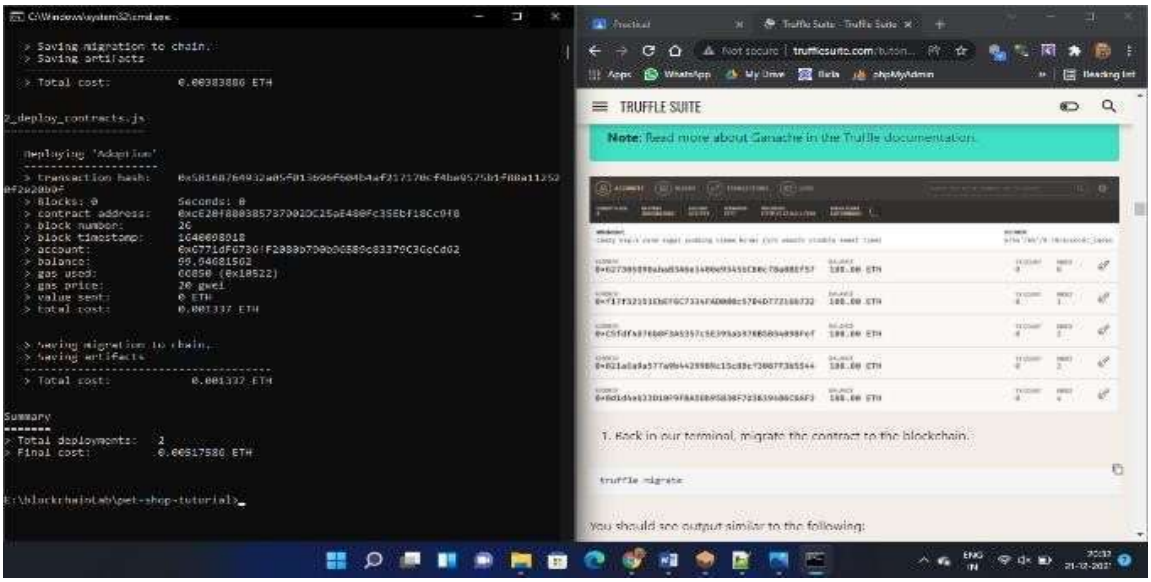




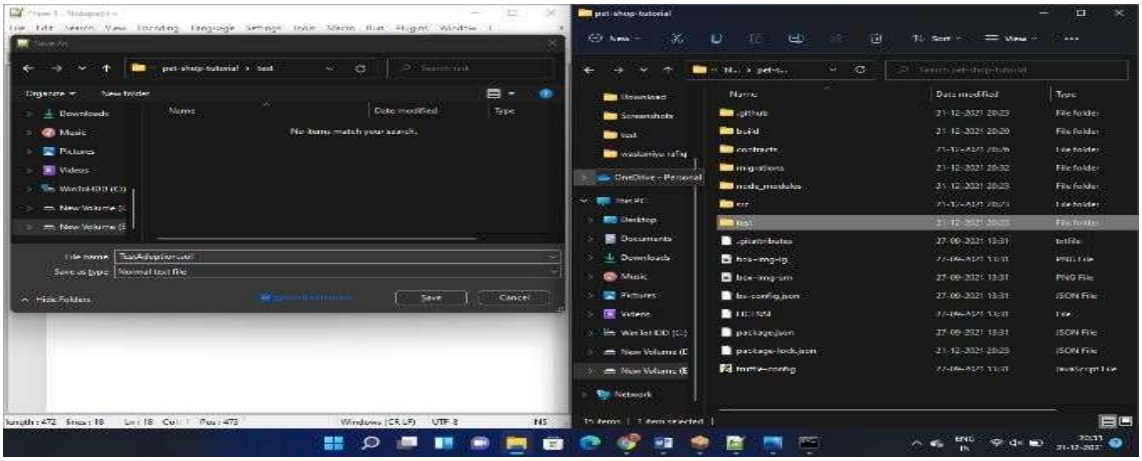
Step-5: Run this command in cmd-

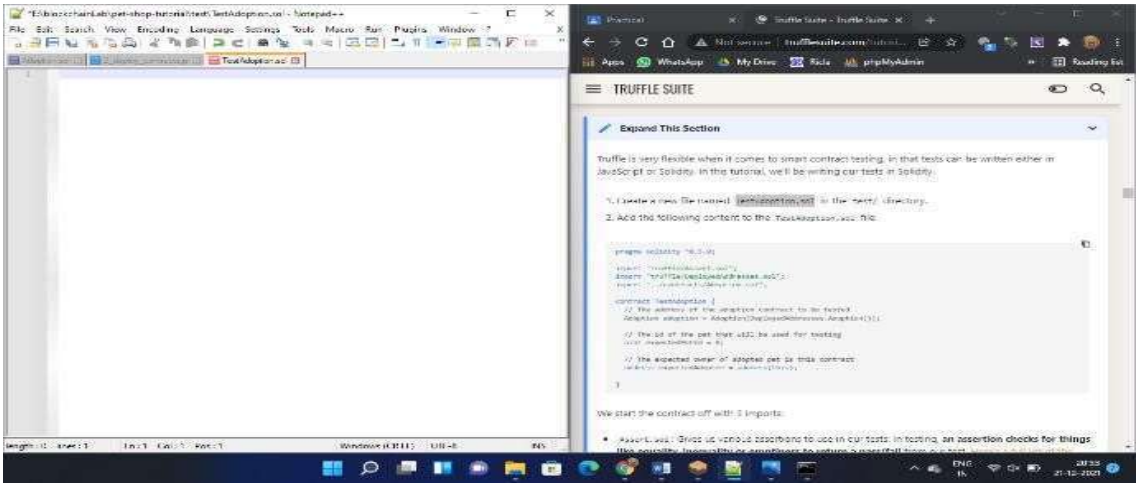
>> truffle migrate



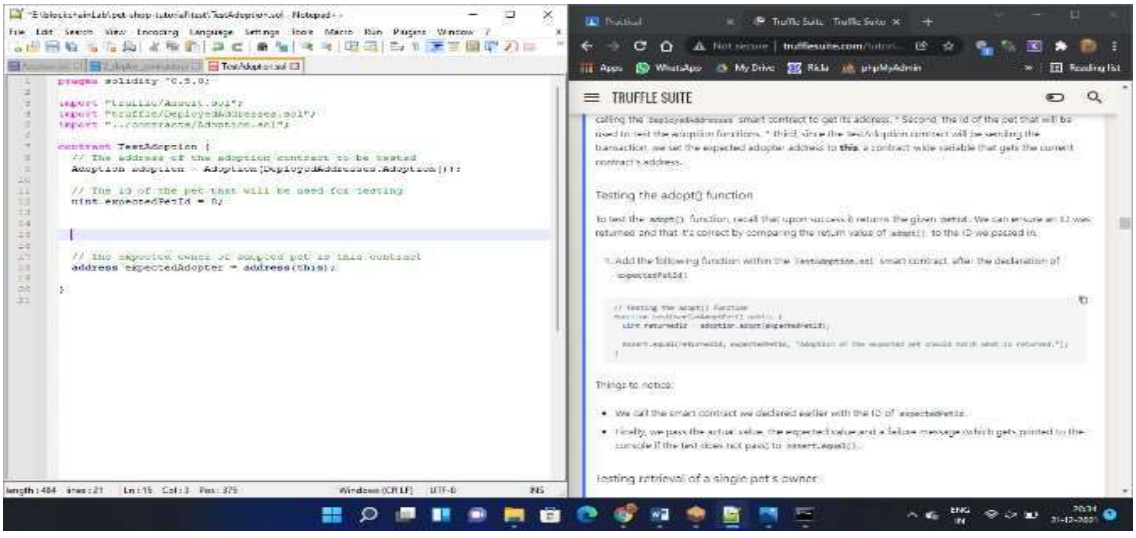


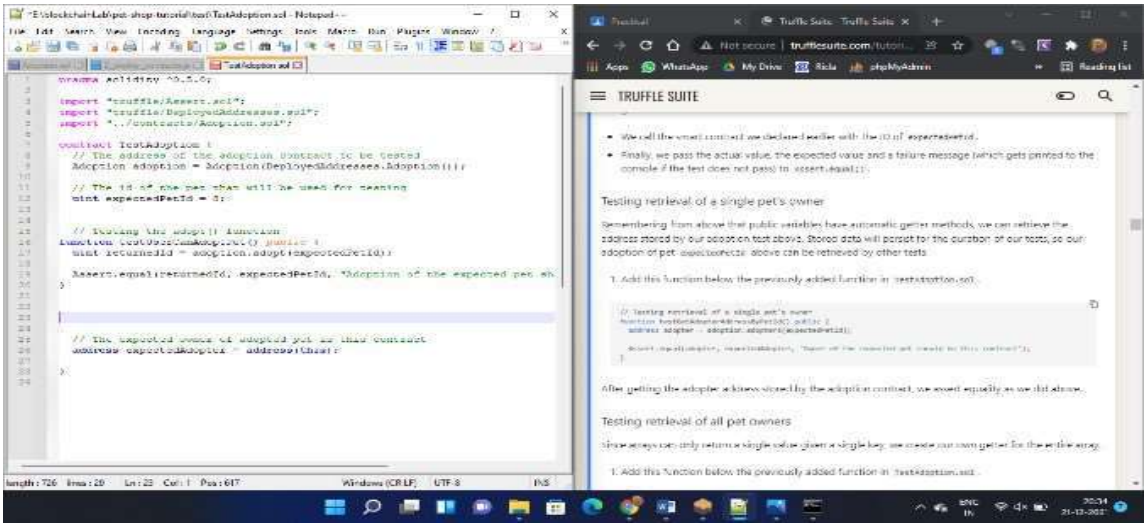
Step-6: Create a new file named TestAdoption.sol in test/ directory and write the code

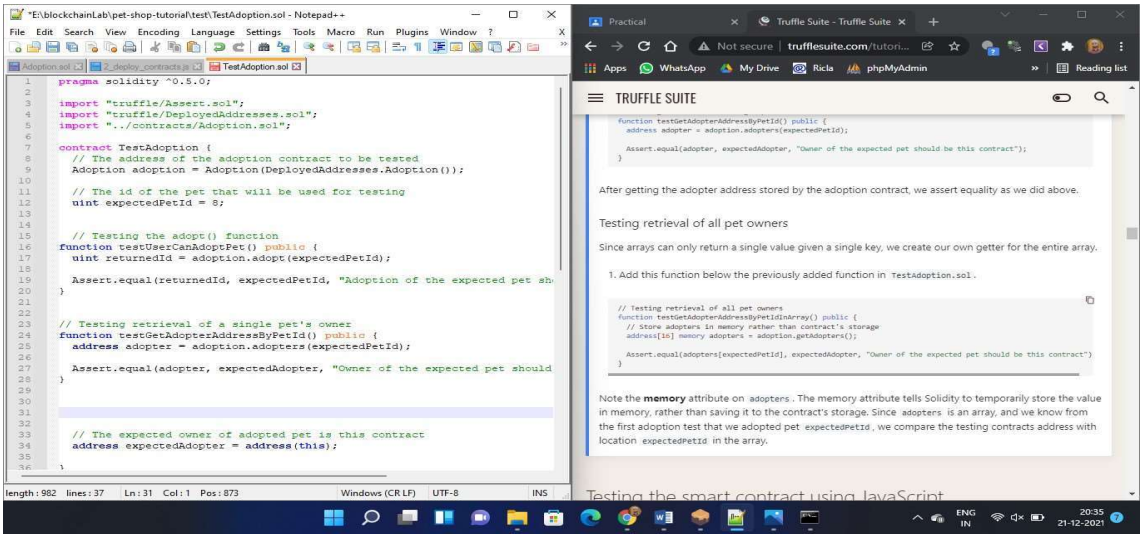




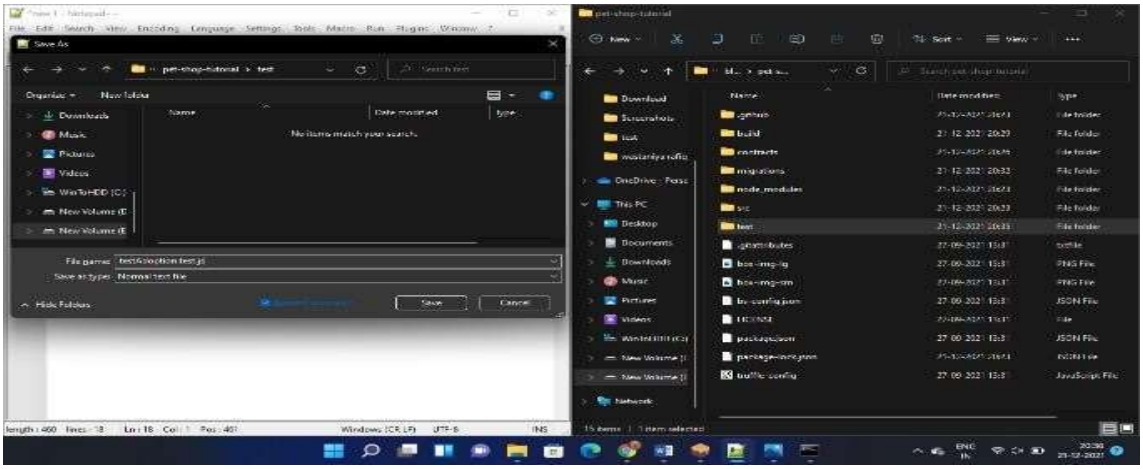
Step7: Add the following function within the TestAdoption after declaration of expectedId and following function previously added function

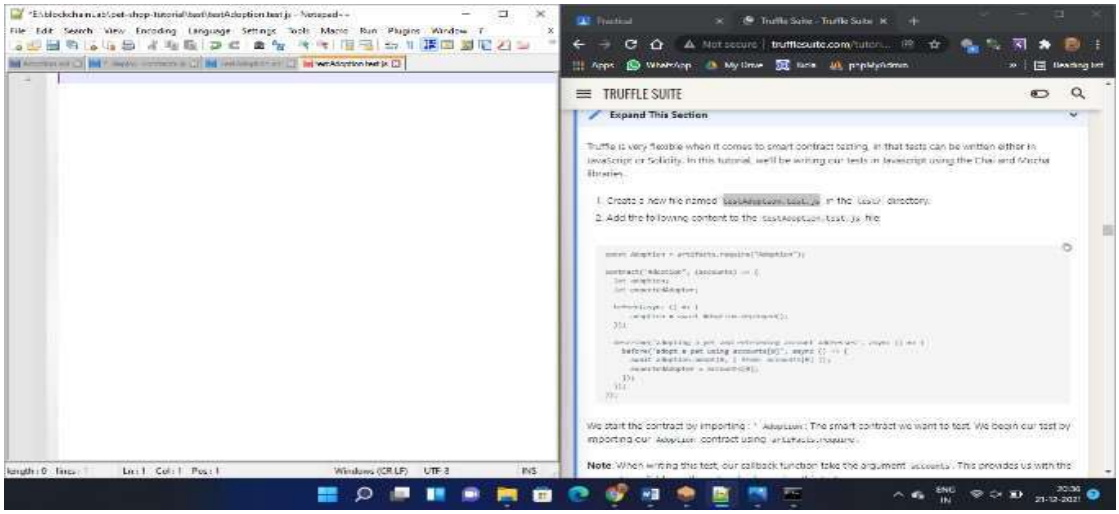




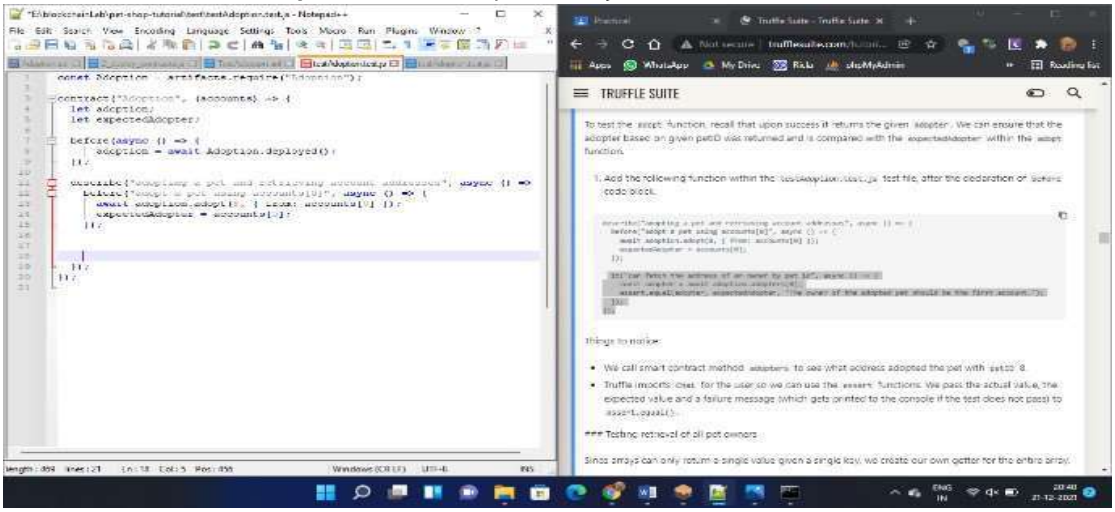


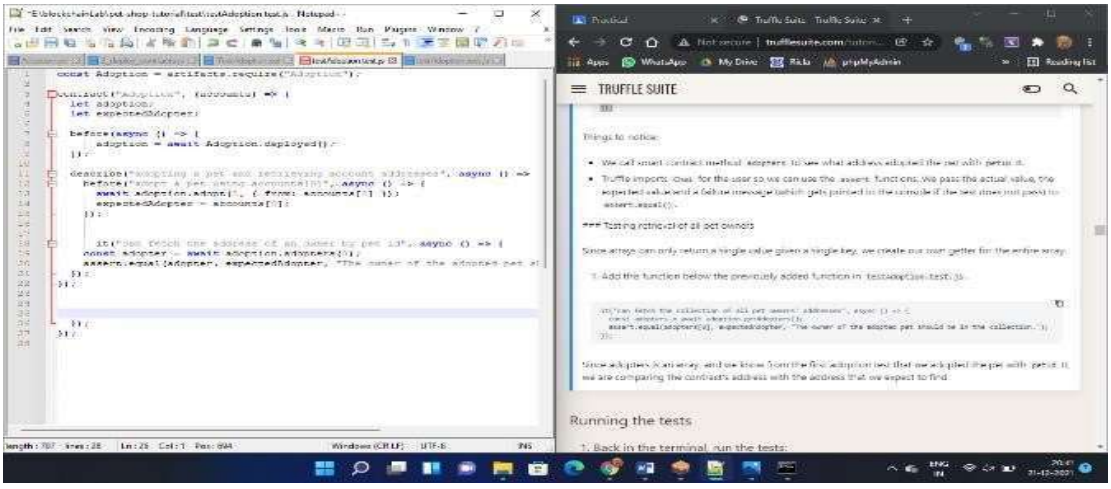
Step-8: Create a new file named testAdoption.test.js in the test/ directory and add following code.





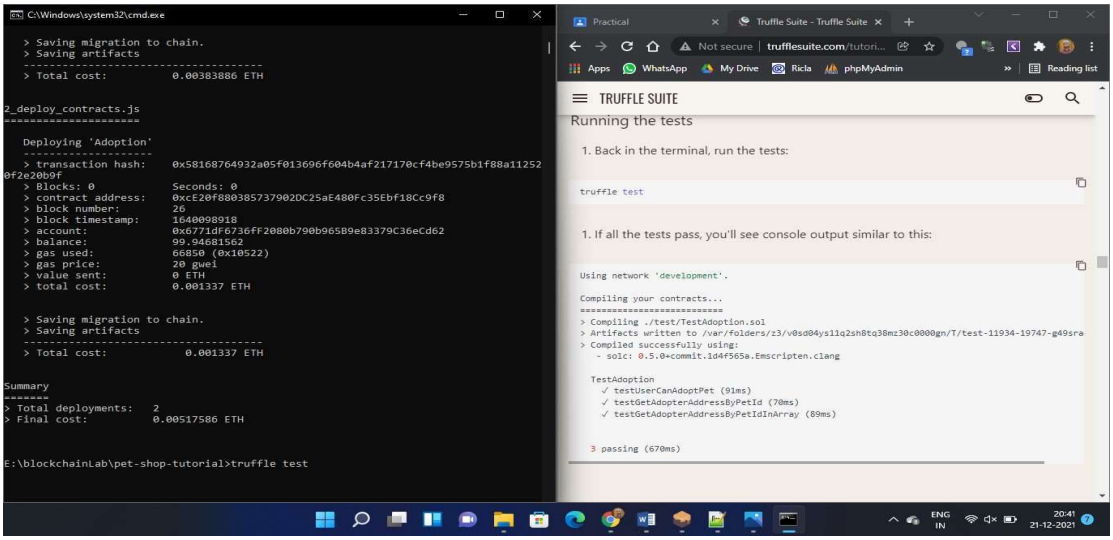
Step-9: Add the following code within the testAdoption.test.js test file after the declaration of before code block and add following code after the previously added code.



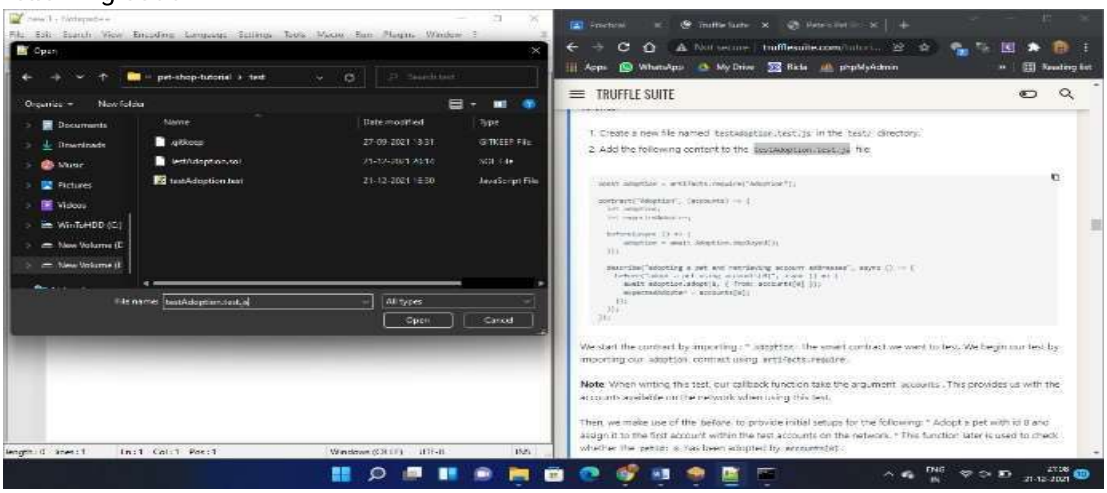


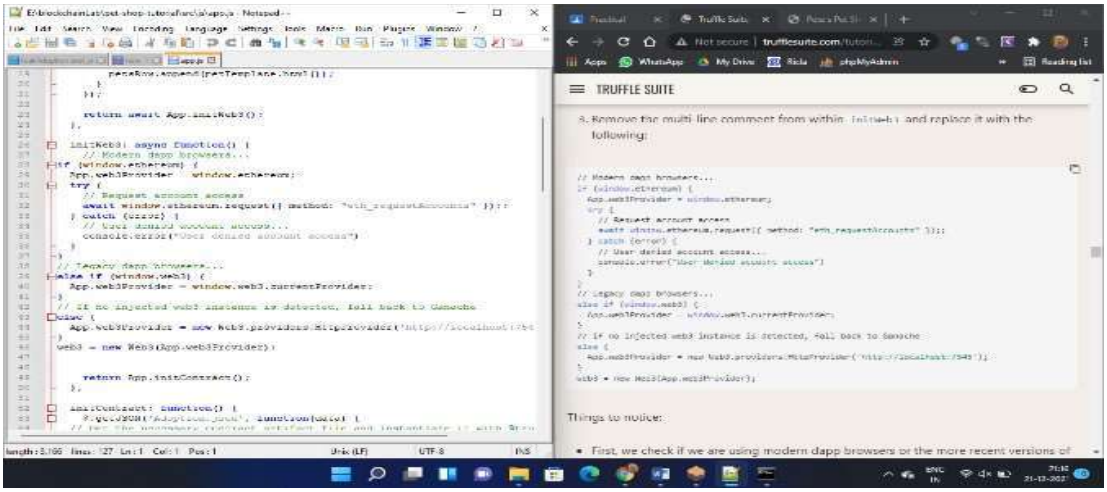
Step-10: Run this command in cmd-

>> truffle test

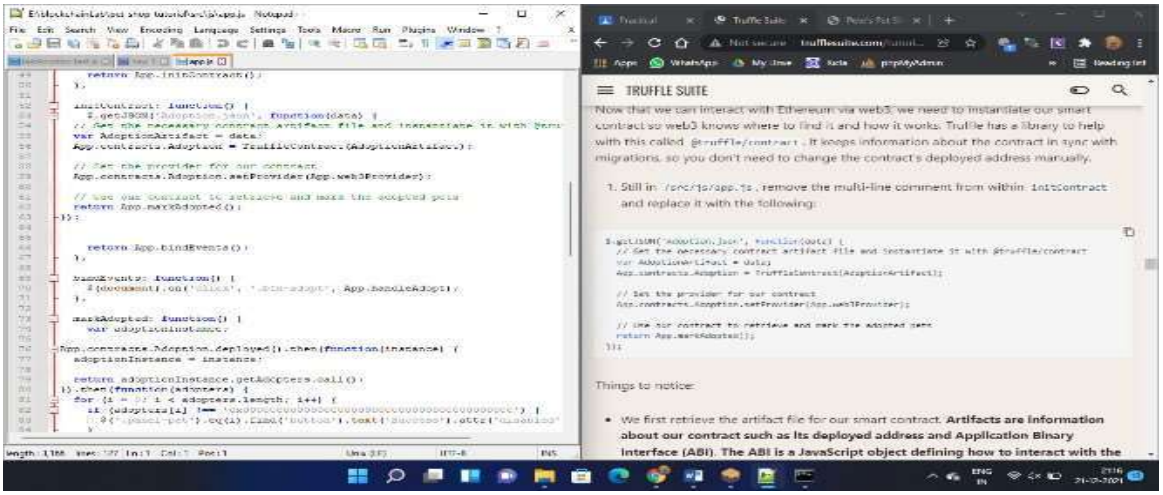


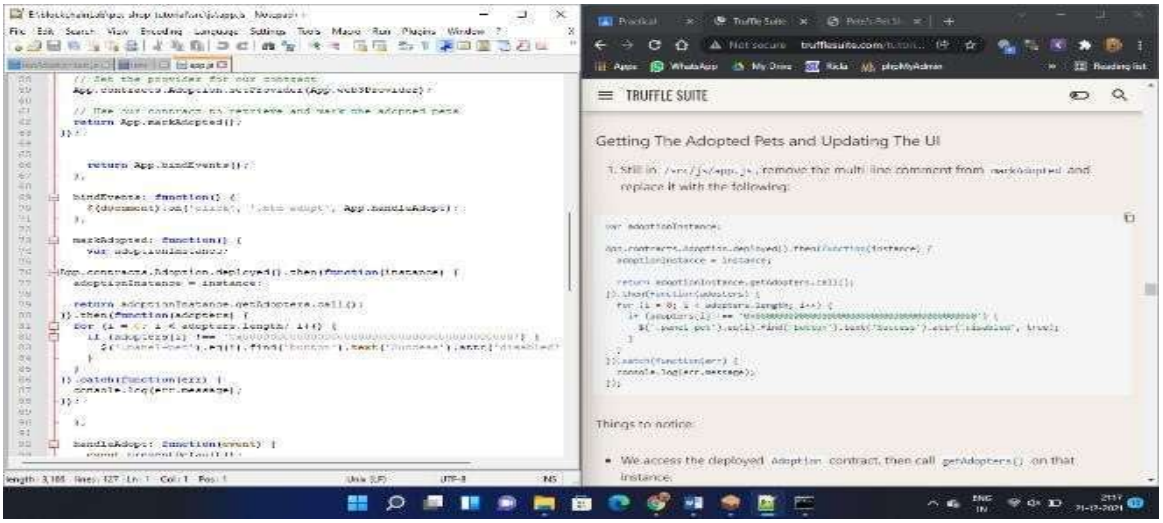
Step 11: Open /src/js/app.js and remove multiline comment from within initWeb3 and replace it with the following code



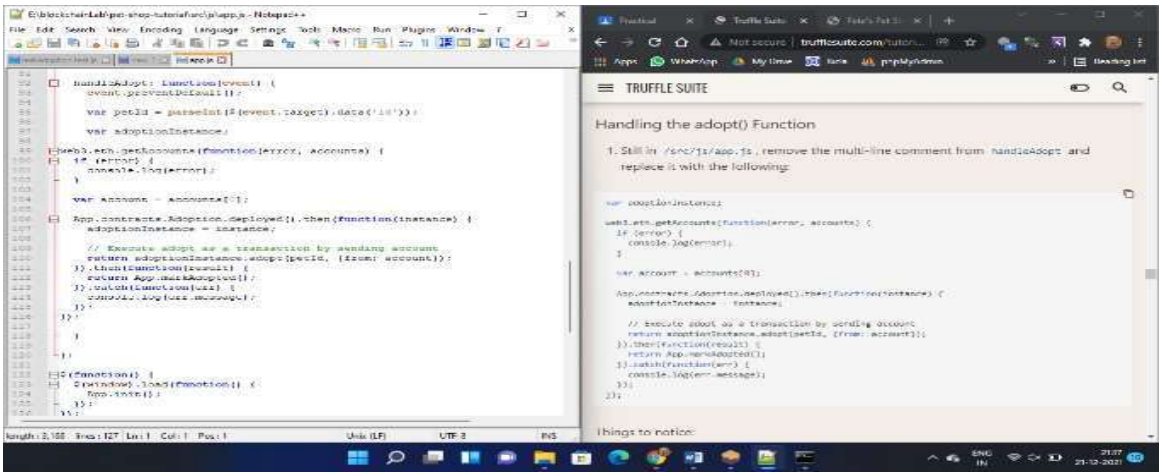


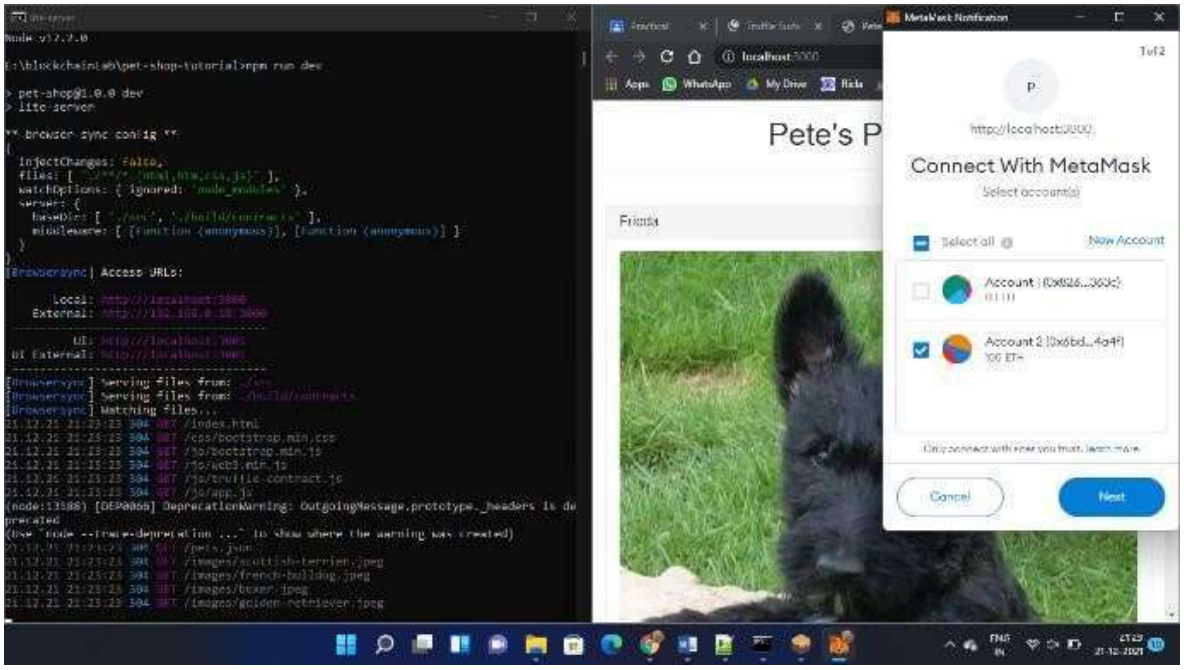
Step 12: Remove multiline comment from within initContract and replace it and markAdopted and replace





Step 13 : Remove multiline comment from handleAdopt and Run command npm run dev and click on next, and connect the opened web page with metamask and adopt a pet by using Ethereum transaction .

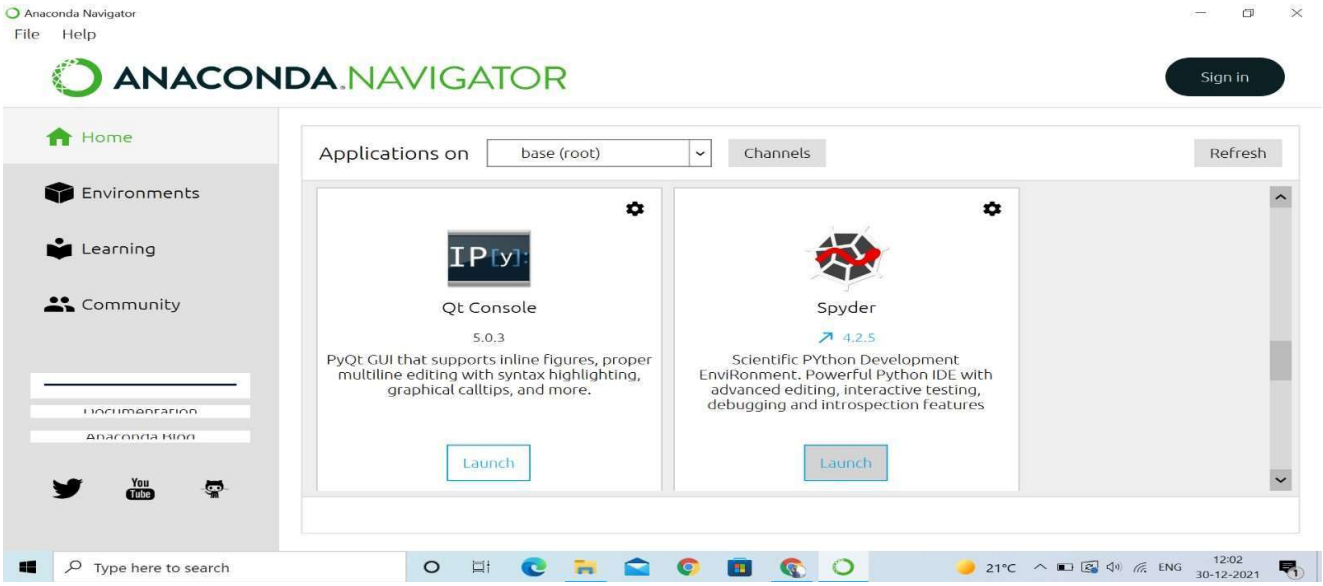



Node v17.2.0
E:\blockchainLab\pet-shop-tutorial>npm run dev
> pet-shop@1.0.0 dev
> lite-server
** browser-sync config **
injectChanges: false,
watchOptions: { ignored: 'node_modules' },
[Browsersync] Access URLs:
Local: http://localhost:3000
External: http://192.168.0.107:3000
UI: http://localhost:3001
UI External: http://localhost:3001
[Browsersync] Serving files from: ./src
[Browsersync] Serving files from: ./build/contracts
[Browsersync] Watching files...
21.12.21 21:23:23 304 GET /index.html
21.12.21 21:23:23 304 GET /css/bootstrap.min.css
21.12.21 21:23:23 304 GET /js/bootstrap.min.js
21.12.21 21:23:23 304 GET /js/web3.min.js
21.12.21 21:23:23 304 GET /js/truffle-contract.js
21.12.21 21:23:23 304 GET /js/app.js
(node:13588) [DEP0066] DeprecationWarning: OutgoingMessage.prototype._headers is de
precated
(Use `node --trace-deprecation ...` to show where the warning was created)
21.12.21 21:23:23 304 GET /pets.json
21.12.21 21:23:23 304 GET /images/scottish-terrier.jpeg
21.12.21 21:23:23 304 GET /images/french-bulldog.jpeg
21.12.21 21:23:23 304 GET /images/boxer.jpeg
21.12.21 21:23:23 304 GET /images/golden-retriever.jpeg
Pete's P
Fronts
MetaMask Notification
1 of 2
http://localhost:3000
Connect With MetaMask
Select account(s)
Select all
New Account
Account 1 (0x626...363c)
Account 2 (0x6bd...4a4f)
100 ETH
Only connect with sites you trust. Learn more
Cancel
Next

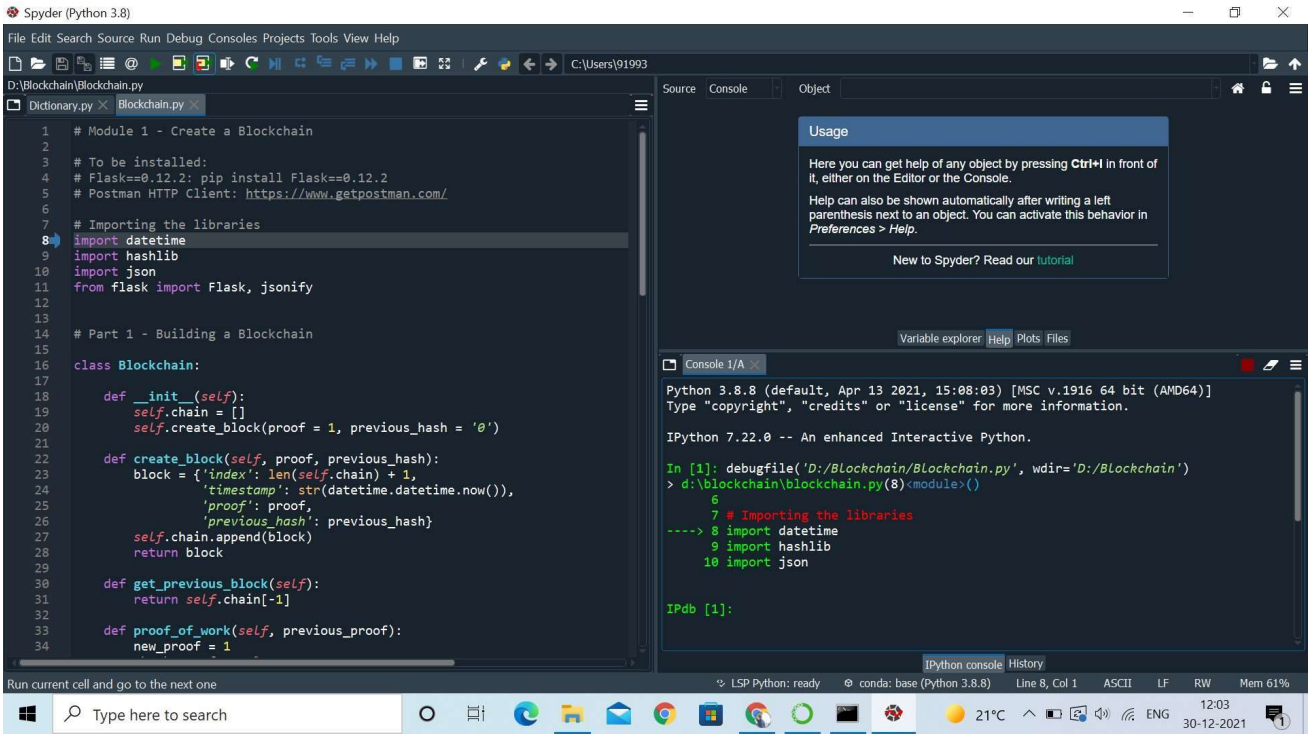
Practical No.8

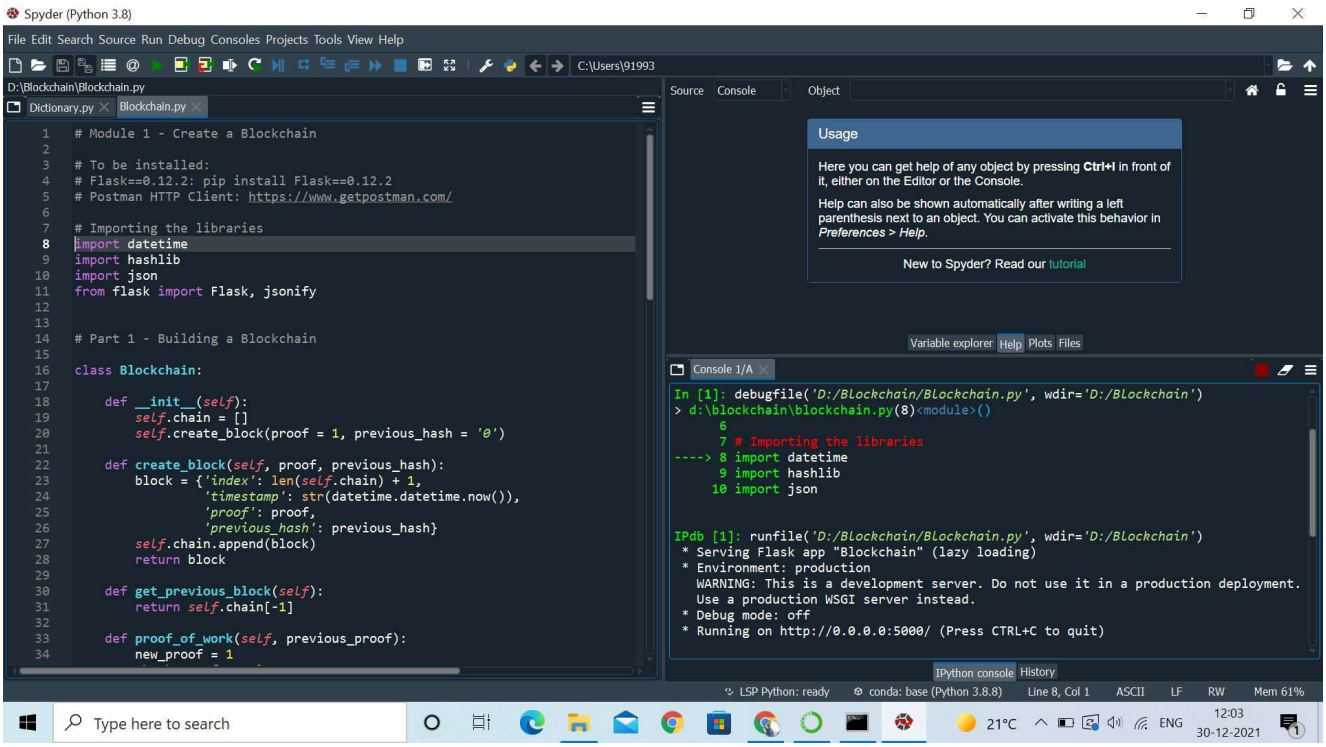
Create the Blockchain using Python, Flask micro web framework and Postman.

Step 1 : Open Anaconda Navigator and Open Spyder.

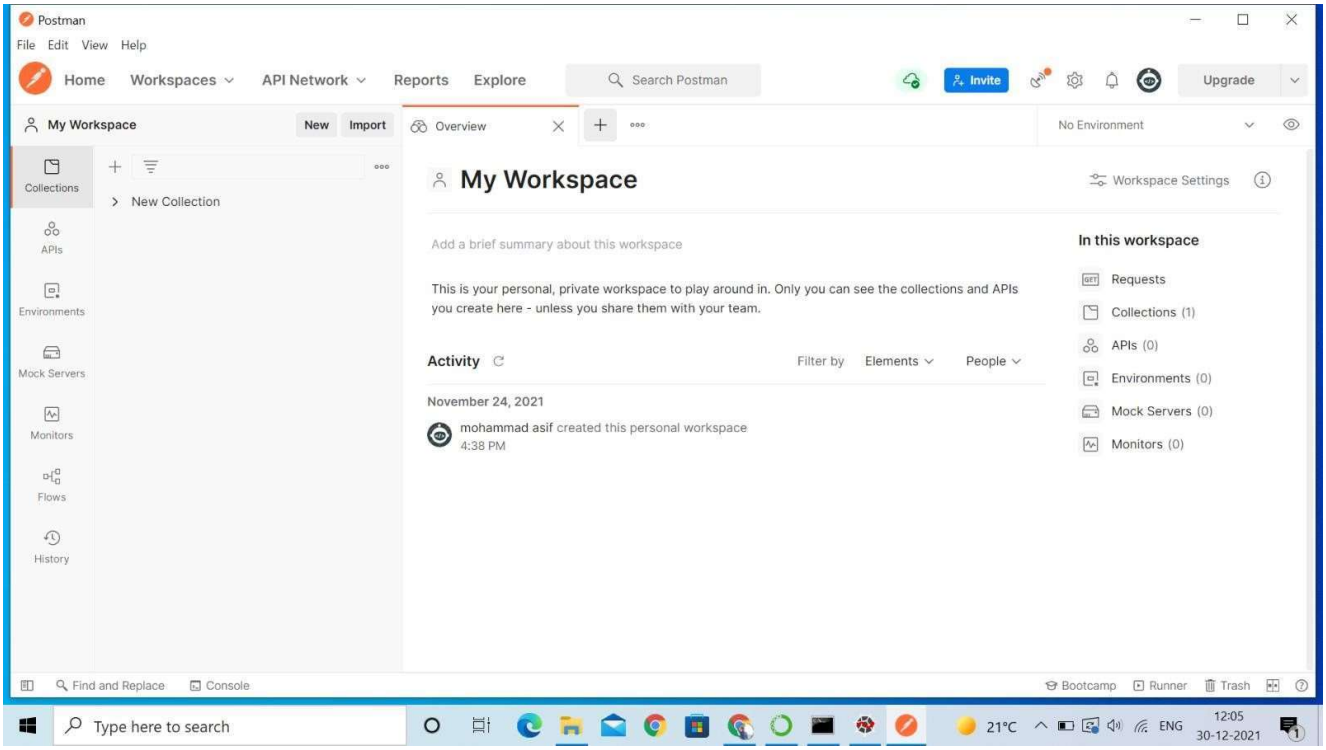


Step 2 : Now open “Blockchain.py” file in spyder >> debug it and run it

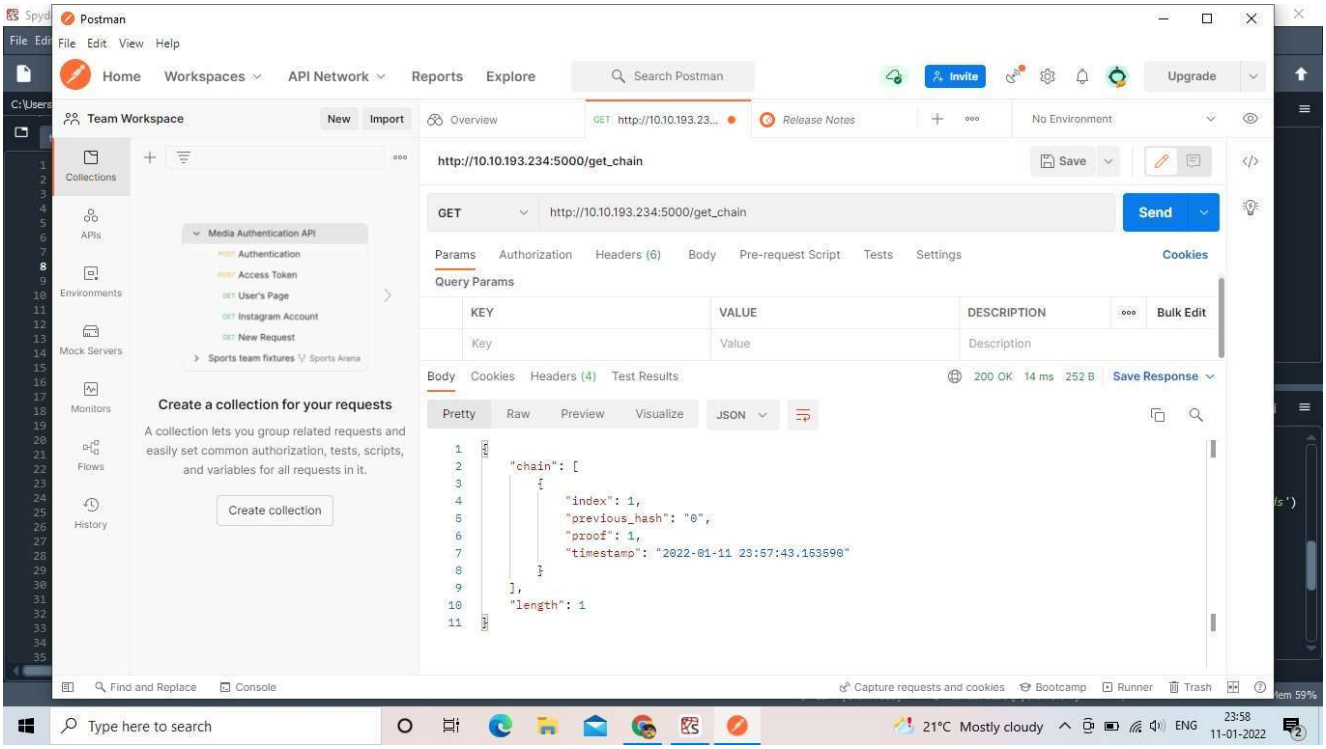




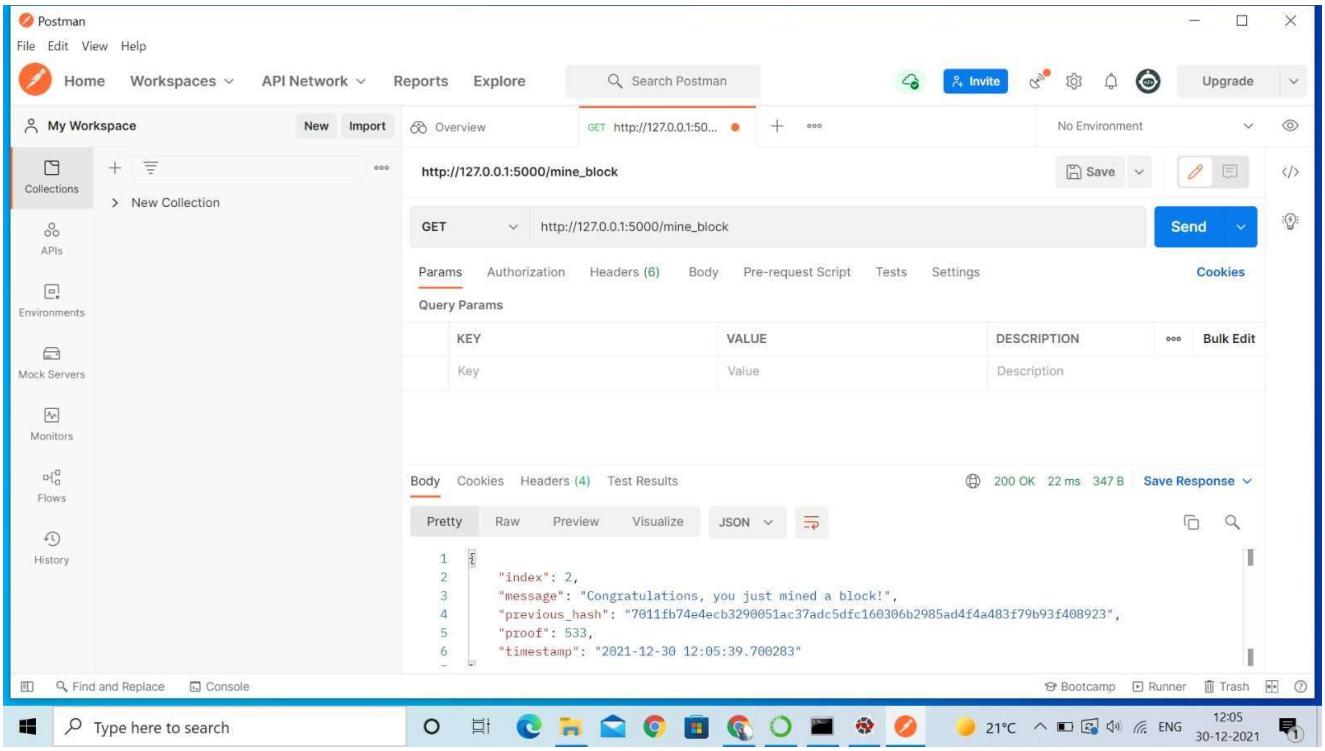
Step 3 : Open Postman and click on My Workspace.

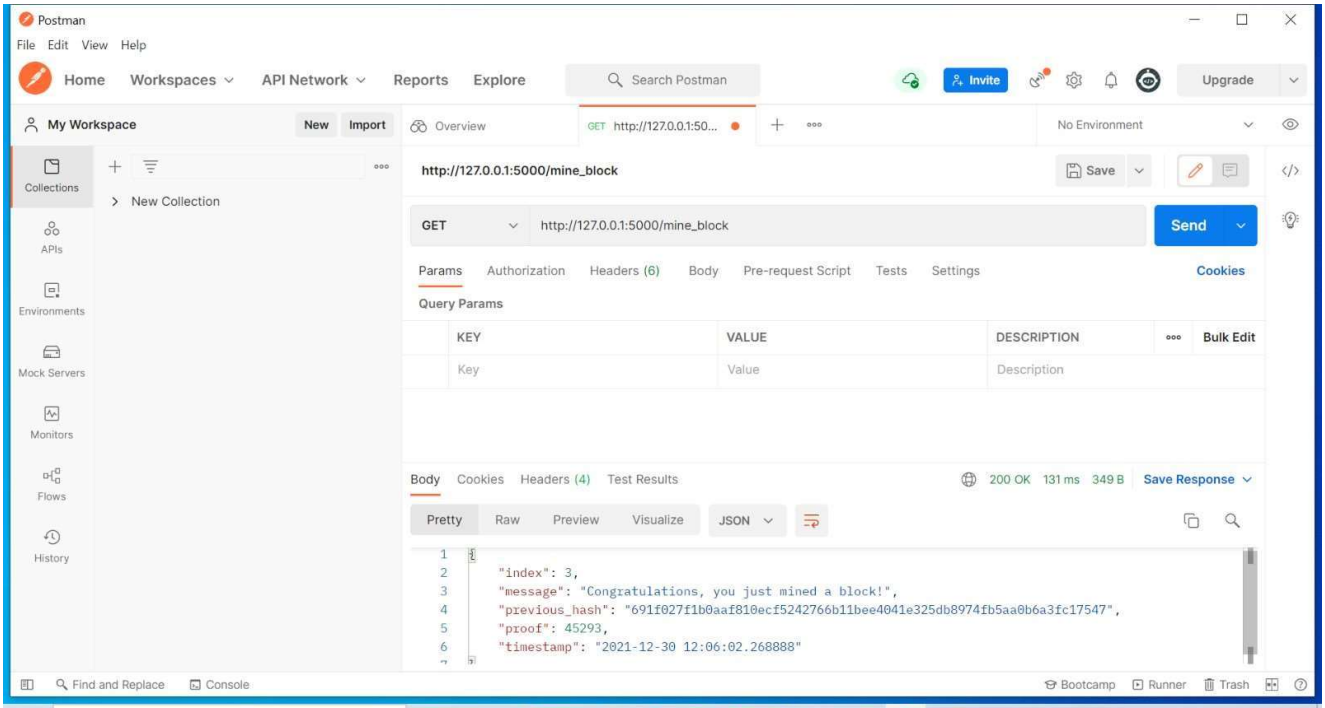


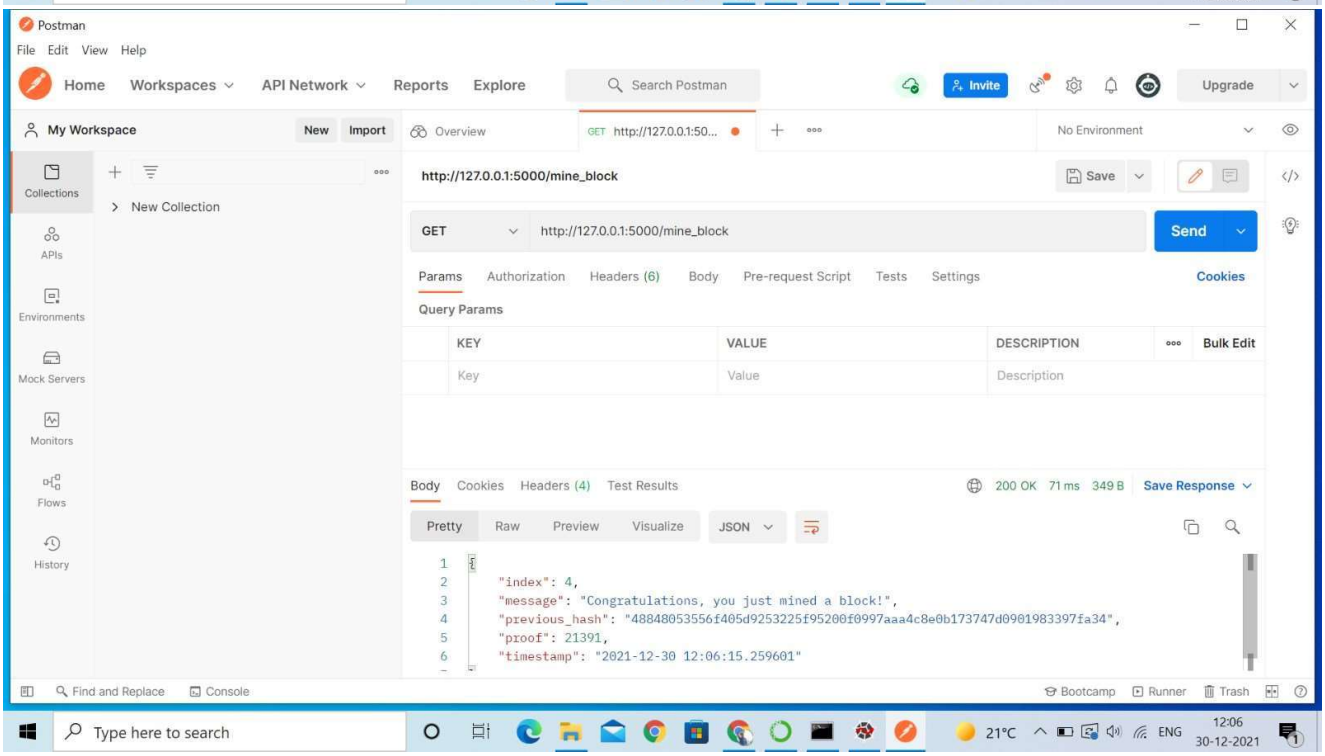
Step 4 : Now write "http://127.0.0.1:5000/get_chain" in GET field and click on Send.

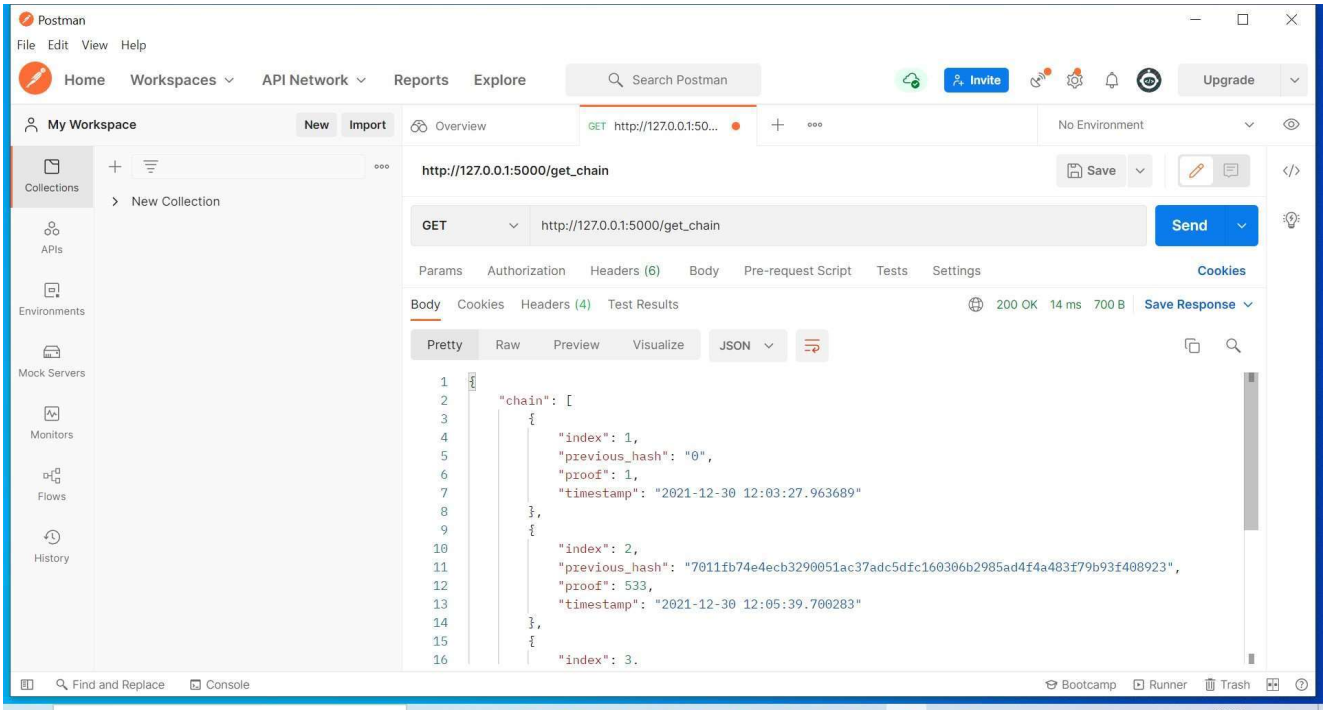


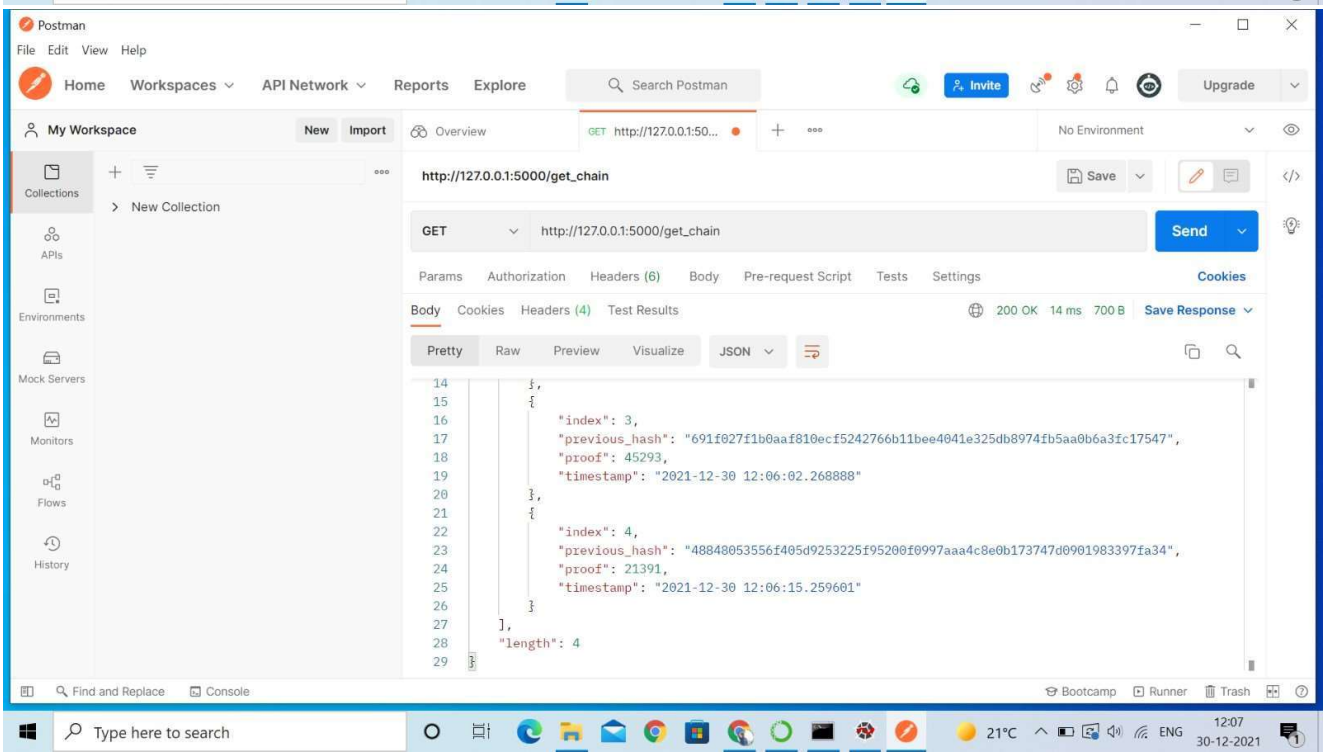
Step 5 : Now write "http://127.0.0.1:5000/mine_block" in GET field and click on Send and repeat it for three four times.




Postman
File Edit View Help
Home Workspaces API Network Reports Explore
Search Postman
Invite
Upgrade
My Workspace
New Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
New Collection
Overview
GET http://127.0.0.1:50...
No Environment
http://127.0.0.1:5000/mine_block
Save
GET
http://127.0.0.1:5000/mine_block
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
Query Params
KEY VALUE DESCRIPTION Bulk Edit
Key Value Description
Body Cookies Headers (4) Test Results
200 OK 131 ms 349 B
Save Response
Pretty Raw Preview Visualize JSON
{
"index": 3,
"message": "Congratulations, you just mined a block!",
"previous_hash": "691f027f1b0aaf810ecf5242766b11bee4041e325db8974fb5aa0b6a3fc17547",
"proof": 45293,
"timestamp": "2021-12-30 12:06:02.268888"
Find and Replace
Console
Bootcamp
Runner
Trash
Type here to search
21°C
ENG
12:06
30-12-2021


Postman
File Edit View Help
Home Workspaces API Network Reports Explore
Search Postman
Invite
Upgrade
My Workspace
New Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
New Collection
Overview
GET http://127.0.0.1:50...
No Environment
http://127.0.0.1:5000/mine_block
Save
GET
http://127.0.0.1:5000/mine_block
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
Query Params
KEY VALUE DESCRIPTION Bulk Edit
Key Value Description
Body Cookies Headers (4) Test Results
200 OK 71 ms 349 B
Save Response
Pretty Raw Preview Visualize JSON
{
"index": 4,
"message": "Congratulations, you just mined a block!",
"previous_hash": "48848053556f405d9253225f95200f0997aaa4c8e0b173747d0901983397fa34",
"proof": 21391,
"timestamp": "2021-12-30 12:06:15.259601"
Find and Replace
Console
Bootcamp
Runner
Trash
Type here to search
21°C
ENG
12:06
30-12-2021


Postman
File Edit View Help
Home Workspaces API Network Reports Explore
Search Postman
Invite
Upgrade
My Workspace
New Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
New Collection
Overview
GET http://127.0.0.1:50...
No Environment
http://127.0.0.1:5000/get_chain
Save
GET
http://127.0.0.1:5000/get_chain
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
Body Cookies Headers (4) Test Results
200 OK 14 ms 700 B
Save Response
Pretty Raw Preview Visualize JSON
{
"chain": [
{
"index": 1,
"previous_hash": "0",
"proof": 1,
"timestamp": "2021-12-30 12:03:27.963689"
},
{
"index": 2,
"previous_hash": "7011fb74e4ecb3290051ac37adc5dfc160306b2985ad4f4a483f79b93f408923",
"proof": 533,
"timestamp": "2021-12-30 12:05:39.700283"
},
{
"index": 3.
Find and Replace
Console
Bootcamp
Runner
Trash
Type here to search
21°C
ENG
12:07
30-12-2021


Postman
File Edit View Help
Home Workspaces API Network Reports Explore
Search Postman
Invite
Upgrade
My Workspace
New Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
New Collection
Overview
GET http://127.0.0.1:50...
No Environment
http://127.0.0.1:5000/get_chain
Save
GET
http://127.0.0.1:5000/get_chain
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
Body Cookies Headers (4) Test Results
200 OK 14 ms 700 B
Save Response
Pretty Raw Preview Visualize JSON
},
{
"index": 3,
"previous_hash": "691f027f1b0aaf810ecf5242766b11bee4041e325db8974fb5aa0b6a3fc17547",
"proof": 45293,
"timestamp": "2021-12-30 12:06:02.268888"
},
{
"index": 4,
"previous_hash": "48848053556f405d9253225f95200f0997aaa4c8e0b173747d0901983397fa34",
"proof": 21391,
"timestamp": "2021-12-30 12:06:15.259601"
}
],
"length": 4
}
Find and Replace
Console
Bootcamp
Runner
Trash
Type here to search
21°C
ENG
12:07
30-12-2021

Practical 9

Read the real-time blocks (.dat files) of Bitcoin using Python. Write at least five programs to read the blocks.

Program 1

```
ample 1.py
from bitcoinlib.blocks import BlockFile, RawBlock
from bitcoinlib.core import Block

def read_blocks_bitcoinlib():
    block_file = BlockFile("/path/to/blocks")
    for raw_block in block_file.read_raw():
        block = Block(raw_block)
        print("Block Height:", block.height)
        # Additional block information can be extracted as needed

read_blocks_bitcoinlib()
```

Program 2

```
Example 1.py
1   import binascii
2   from pybitcointools.main import deserialize
3
4   def read_blocks_pybitcointools():
5       with open("/path/to/blocks", "rb") as file:
6           while True:
7               raw_block = file.read(80)
8               if not raw_block:
9                   break
0               block_data = deserialize(binascii.hexlify(raw_block).decode())
1               print("Block Hash:", block_data["hash"])
2
3   # Make sure to install the pybitcointools library:
4   # pip install bitcoin
5   read_blocks_pybitcointools()
```

Program 3

```python
from blocktools import Block

def read_blocks_blocktools():
    with open("/path/to/blocks", "rb") as file:
        while True:
            raw_block = file.read(80)
            if not raw_block:
                break
            block = Block(raw_block)
            print("Block Hash:", block.hash)

read_blocks_blocktools()
```

Program 4

```python
import subprocess
import json

def read_blocks_blockparser():
    command = "blockparser --read-headers --silent /path/to/blocks"
    process = subprocess.Popen(command, stdout=subprocess.PIPE, shell=True)
    while True:
        line = process.stdout.readline()
        if not line:
            break
        block_data = json.loads(line)
        print("Block Hash:", block_data["hash"])

# Make sure to install blockparser: https://github.com/znort987/blockparser
read_blocks_blockparser()
```

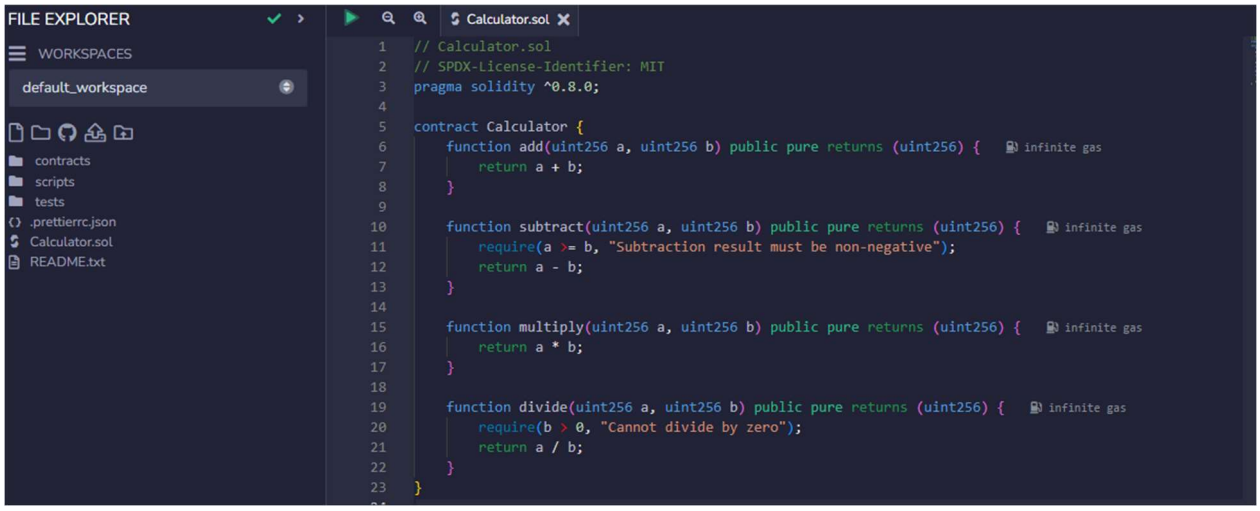
Program 5

```python
from bitcoin_block_explorer import BitcoinBlockExplorer

def read_blocks_block_explorer():
    explorer = BitcoinBlockExplorer("/path/to/blocks")
    for block in explorer.get_blocks():
        print("Block Hash:", block.hash)

# Make sure to install bitcoin-block-explorer library:
# pip install bitcoin-block-explorer
read_blocks_block_explorer()
```

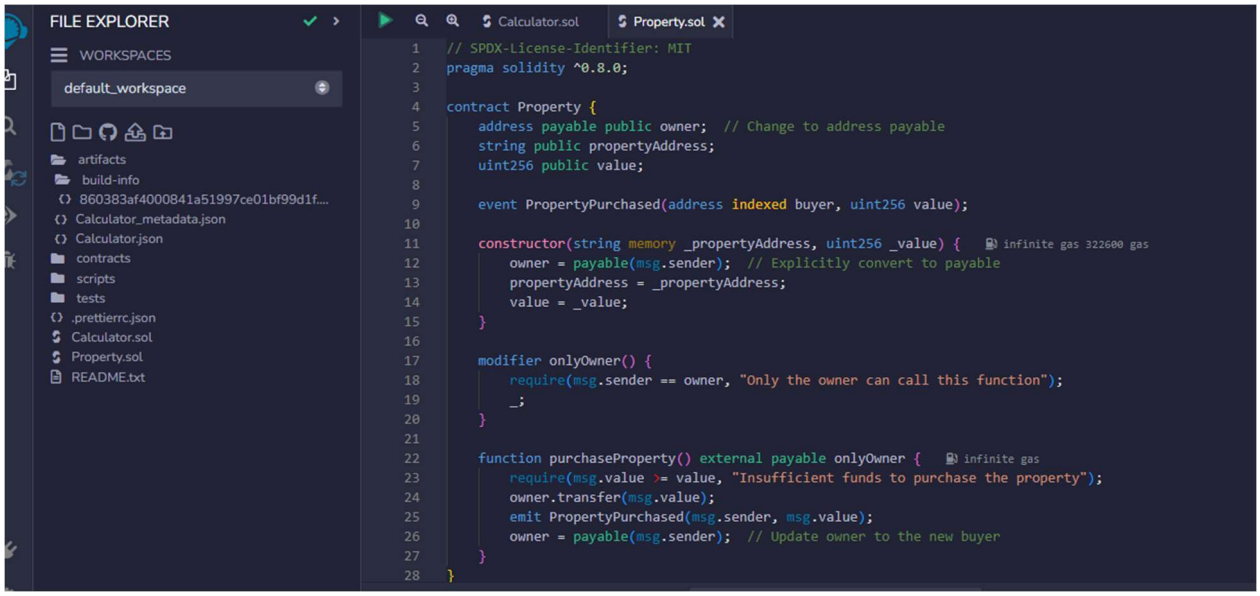
Practical 10

Create the following Contracts using Solidity Programming and compile these programs on Remix using Ropston Test Network, Goerli Test Network and Ganache network. a. Calculator b. Property c. Lottery

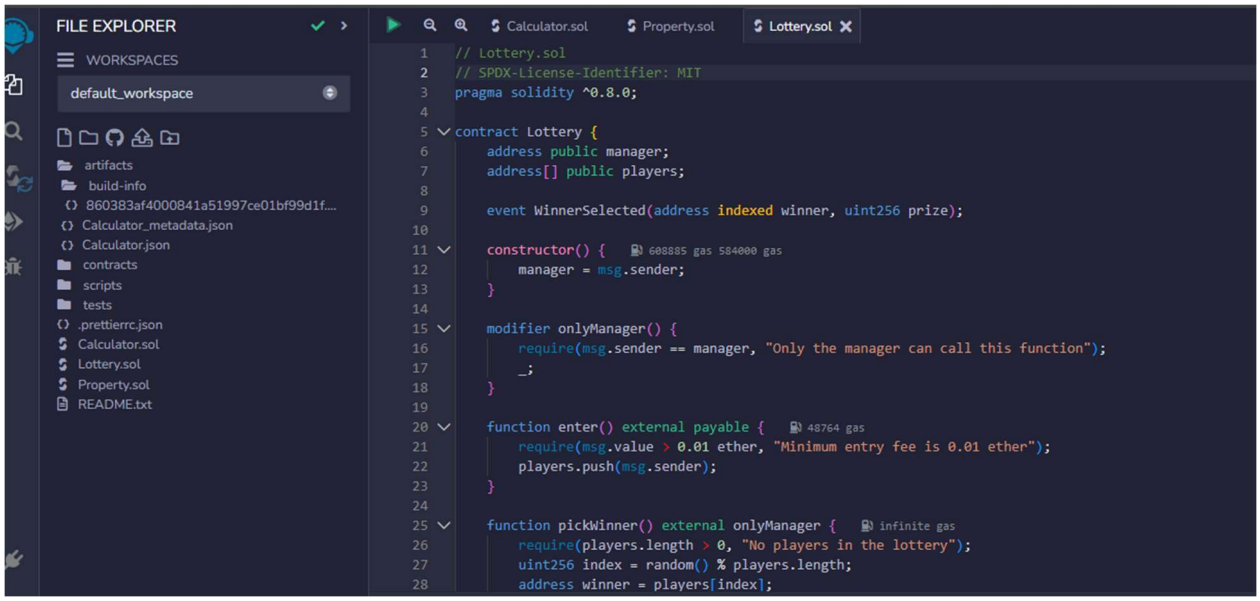
Calculator



Property



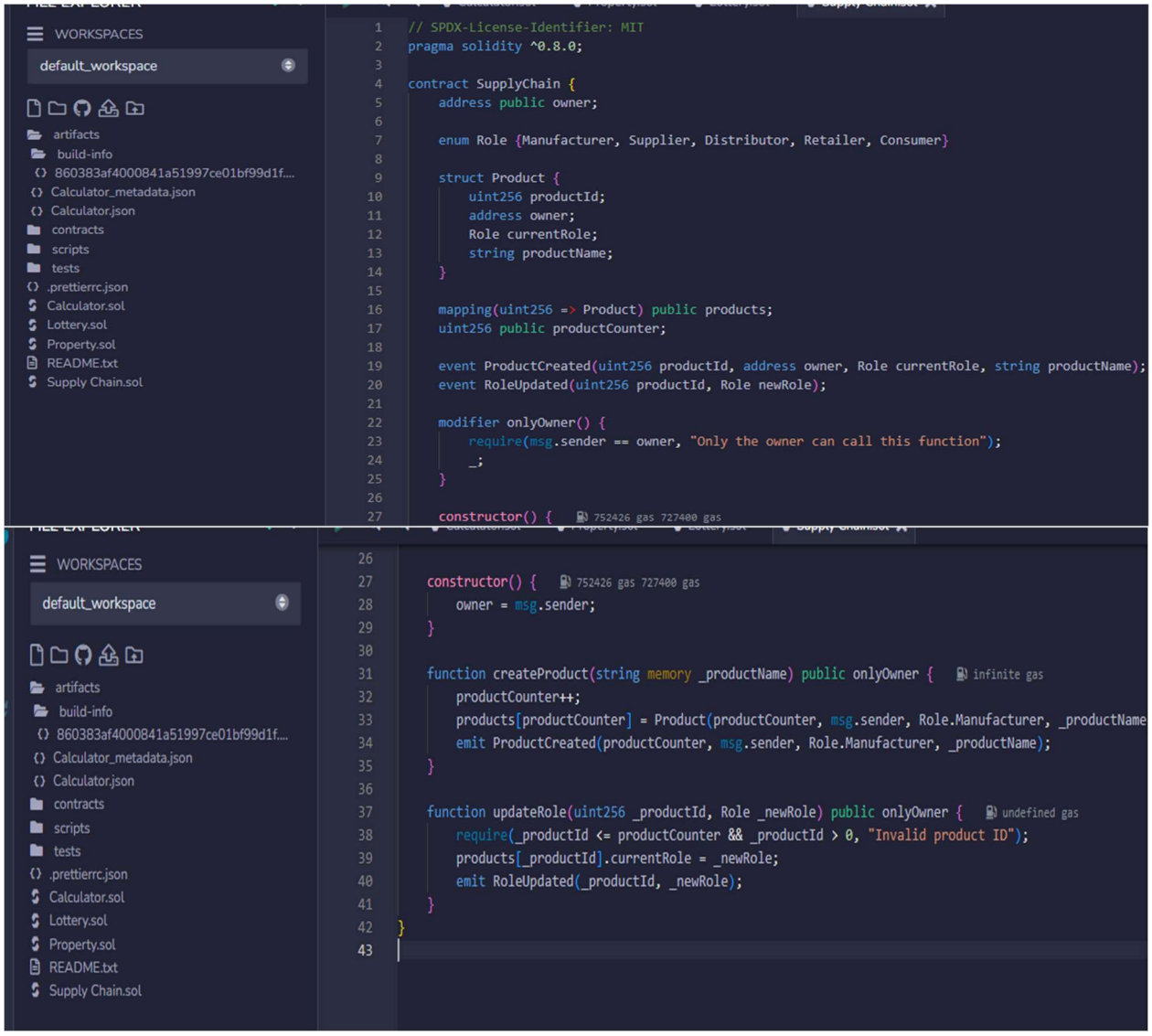
Lottery


FILE EXPLORER
WORKSPACES
default_workspace
artifacts
build-info
860383af4000841a51997ce01bf99d1f....
Calculator_metadata.json
Calculator.json
contracts
scripts
tests
.prettierrc.json
Calculator.sol
Lottery.sol
Property.sol
README.txt
Calculator.sol
Property.sol
Lottery.sol
// Lottery.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract Lottery {
address public manager;
address[] public players;
event WinnerSelected(address indexed winner, uint256 prize);
constructor() { 608885 gas 584000 gas
manager = msg.sender;
}
modifier onlyManager() {
require(msg.sender == manager, "Only the manager can call this function");
_;
}
function enter() external payable { 48764 gas
require(msg.value > 0.01 ether, "Minimum entry fee is 0.01 ether");
players.push(msg.sender);
}
function pickWinner() external onlyManager { infinite gas
require(players.length > 0, "No players in the lottery");
uint256 index = random() % players.length;
address winner = players[index];


default_workspace
artifacts
build-info
860383af4000841a51997ce01bf99d1f....
Calculator_metadata.json
Calculator.json
contracts
scripts
tests
.prettierrc.json
Calculator.sol
Lottery.sol
Property.sol
README.txt
address winner = players[index];
uint256 prize = address(this).balance;
payable(winner).transfer(prize);
emit WinnerSelected(winner, prize);
players = new address[](0); // Reset the players array for the next round
}
function getPlayers() external view returns (address[] memory) { infinite gas
return players;
}
function random() private view returns (uint256) { infinite gas
return uint256(keccak256(abi.encodePacked(block.difficulty, block.timestamp, players)));
}
}

Practical 11

Supply chain management using python in blockchain technology



Supply_chain.py > ...

Click here to ask Blackbox to help you code faster

```python
from web3 import Web3
web3 = Web3(Web3.HTTPProvider('http://localhost:8545'))

# Load contract ABI and bytecode
with open('SupplyChain.abi', 'r') as file:
    abi = file.read()

with open('SupplyChain.bin', 'r') as file:
    bytecode = file.read()

# Deploy the contract
supply_chain = web3.eth.contract(abi=abi, bytecode=bytecode)
tx_hash = supply_chain.constructor().transact({'from': web3.eth.accounts[0]})
tx_receipt = web3.eth.waitForTransactionReceipt(tx_hash)

# Interact with the contract
contract_address = tx_receipt['contractAddress']
supply_chain_instance = web3.eth.contract(address=contract_address, abi=abi)

# Create a product
supply_chain_instance.functions.createProduct("ProductA").transact({'from': web3.eth.accounts[0]})

# Update role of the product
supply_chain_instance.functions.updateRole(1, 2).transact({'from': web3.eth.accounts[0]})

# Get product details
product_details = supply_chain_instance.functions.products(1).call()
print("Product Details:", product_details)

```