



MAULANA AZAD NATIONAL URDU UNIVERSITY

• BLOCKCHAIN technology LAB RECORDS

Name: JAMSHED ALAM

Roll No.: 20MMCA002HY

Course: MCA-3rd semester

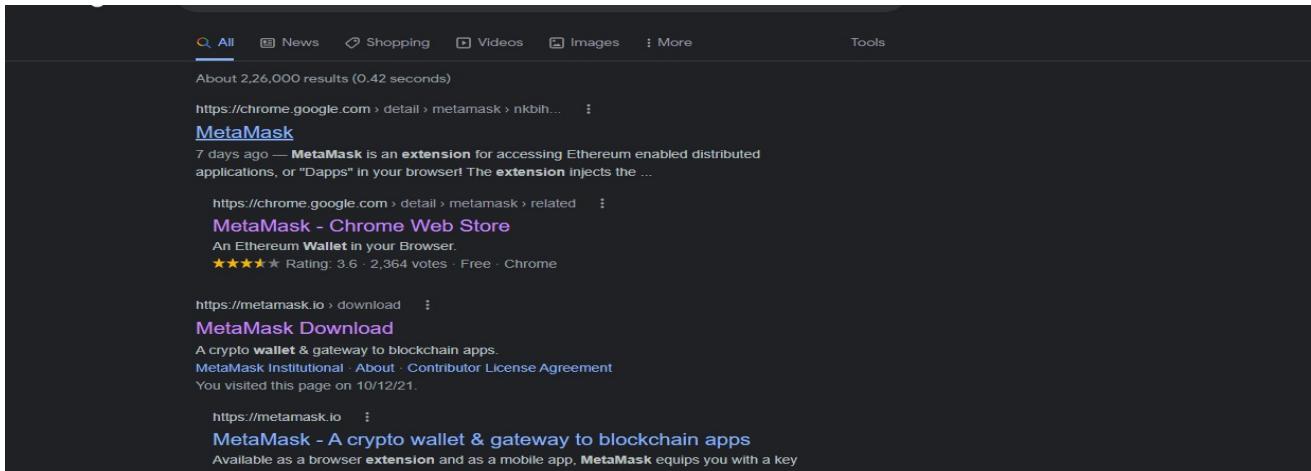
Department: CS&IT

SUBJECT:BCT LAB

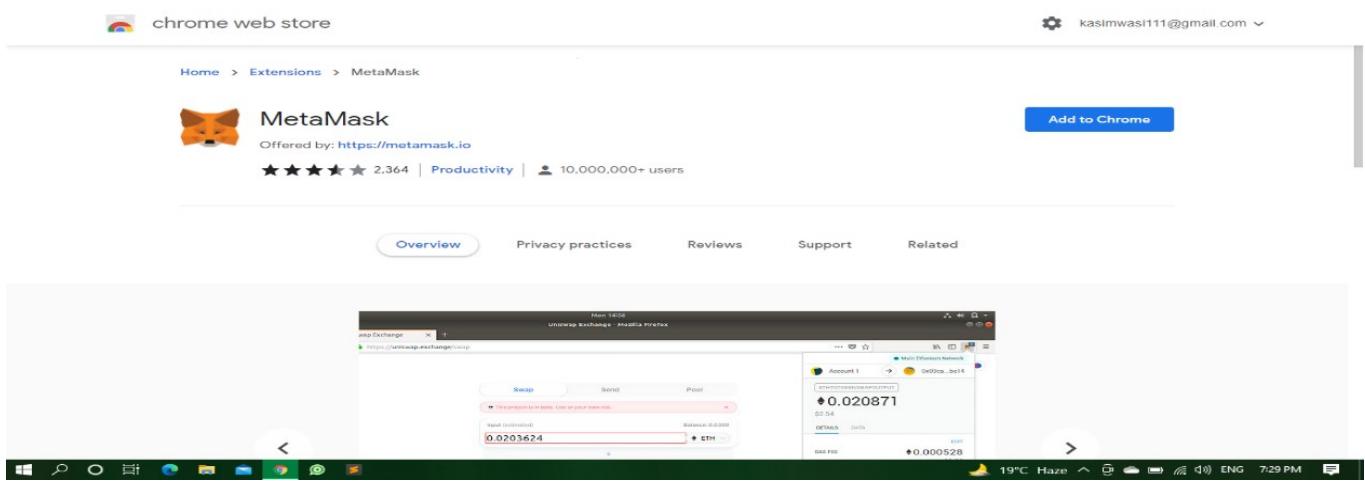
Practical No.1

MetaMask Wallet

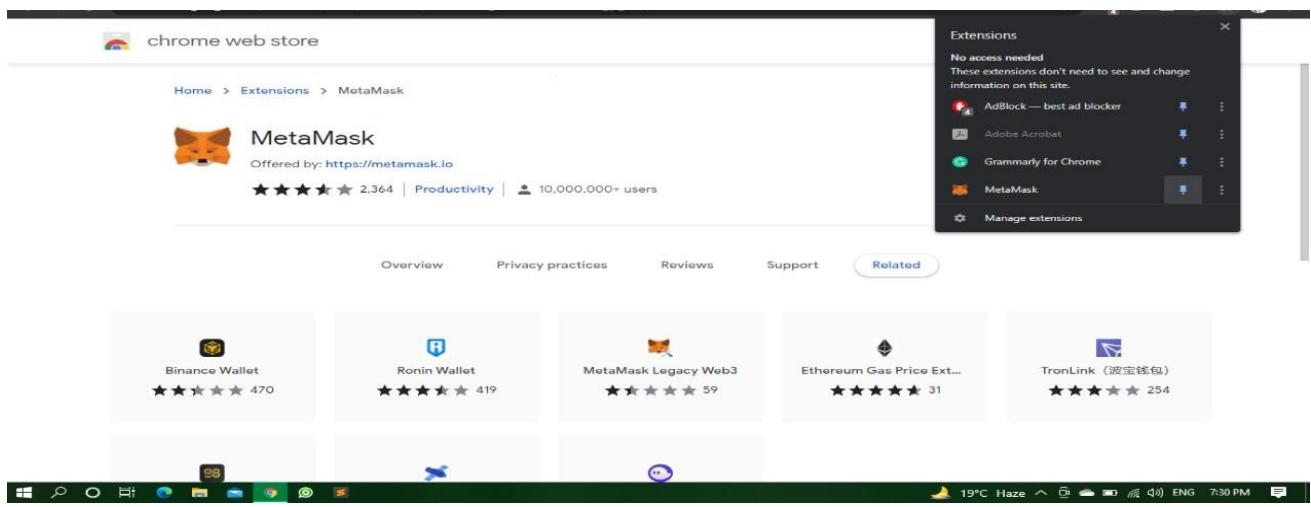
Step-1: Open Chrome Browser and search for “Metamask Wallet” and add extension download and open the first link.



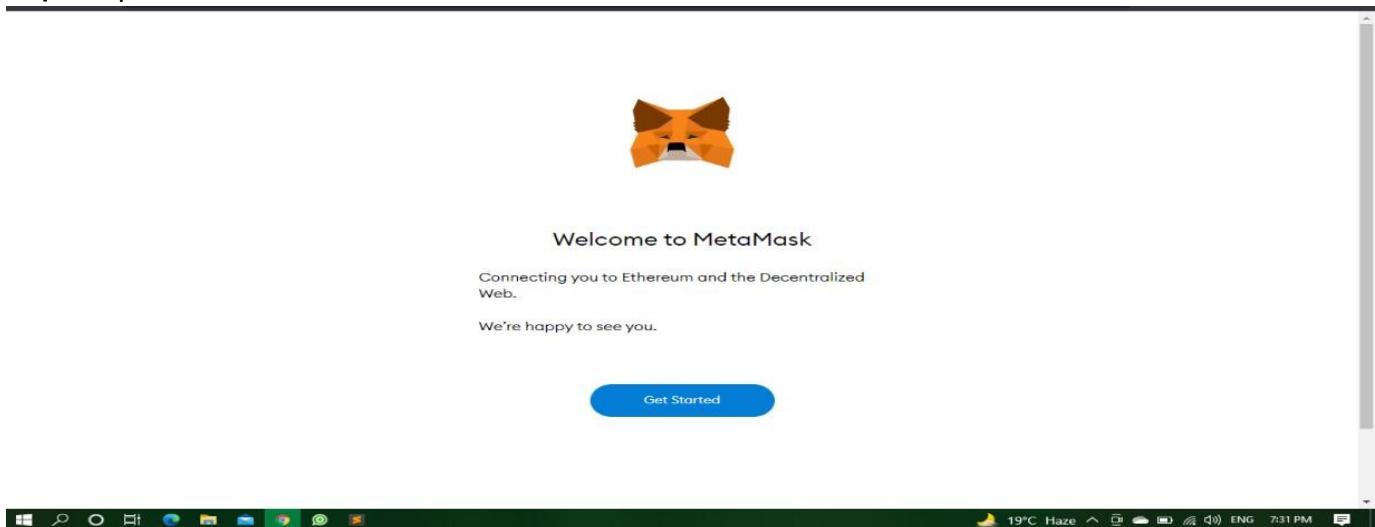
Step-2: Click on “Add to Chrome” and simply add it in your extension list.



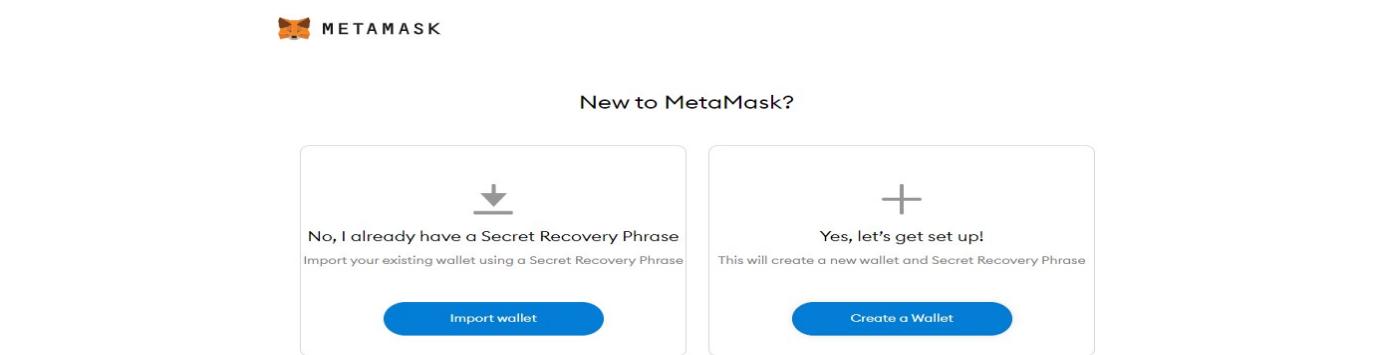
Step-3: click on extension option and pin the Metamask Wallet extension.



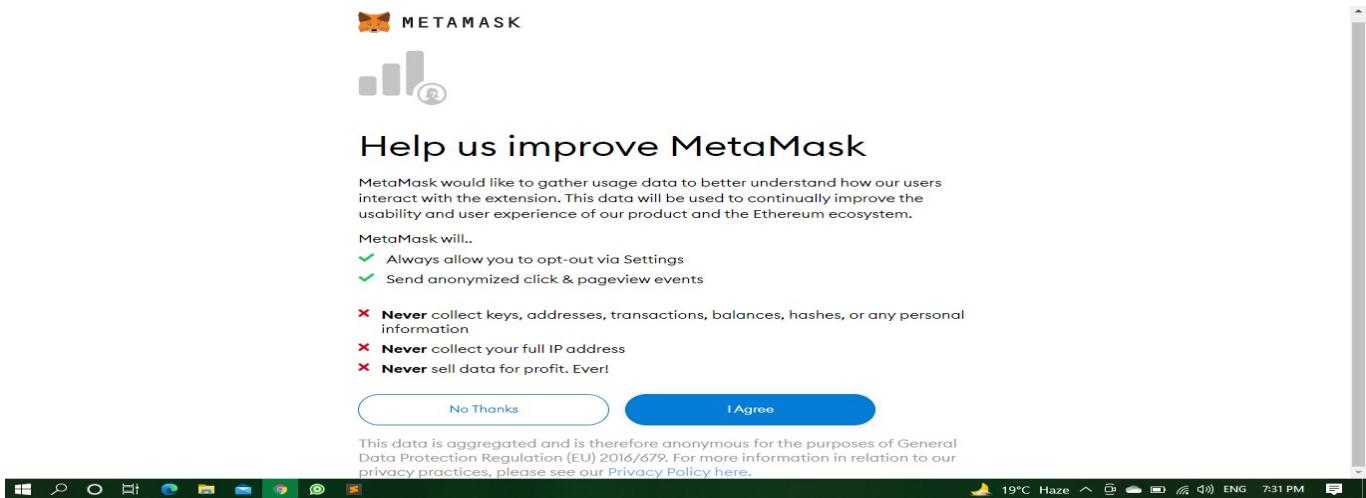
Step-4: Open Metamask Wallet and click “Get Started”.



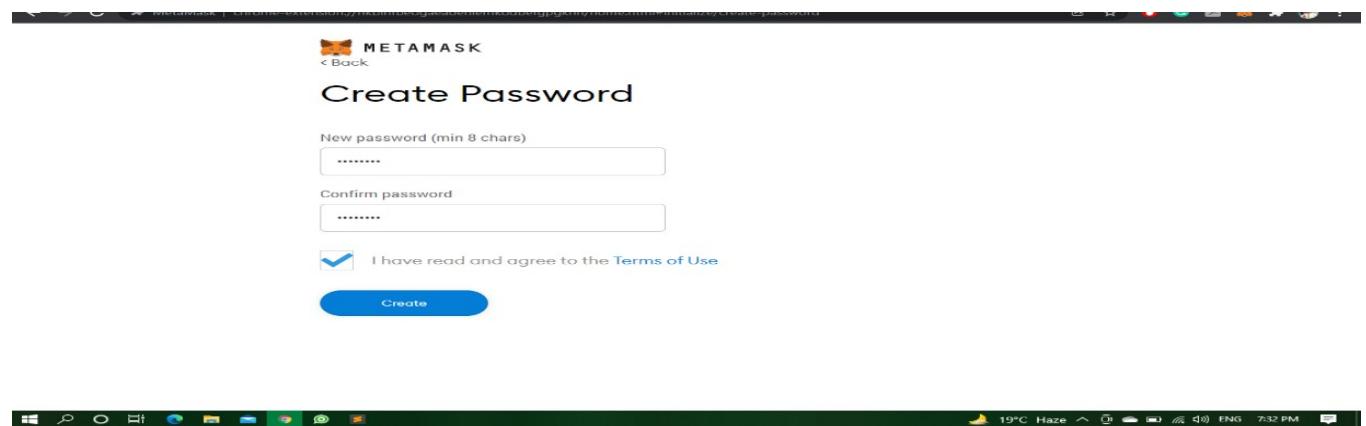
Step-5: Click on “Create a Wallet”.



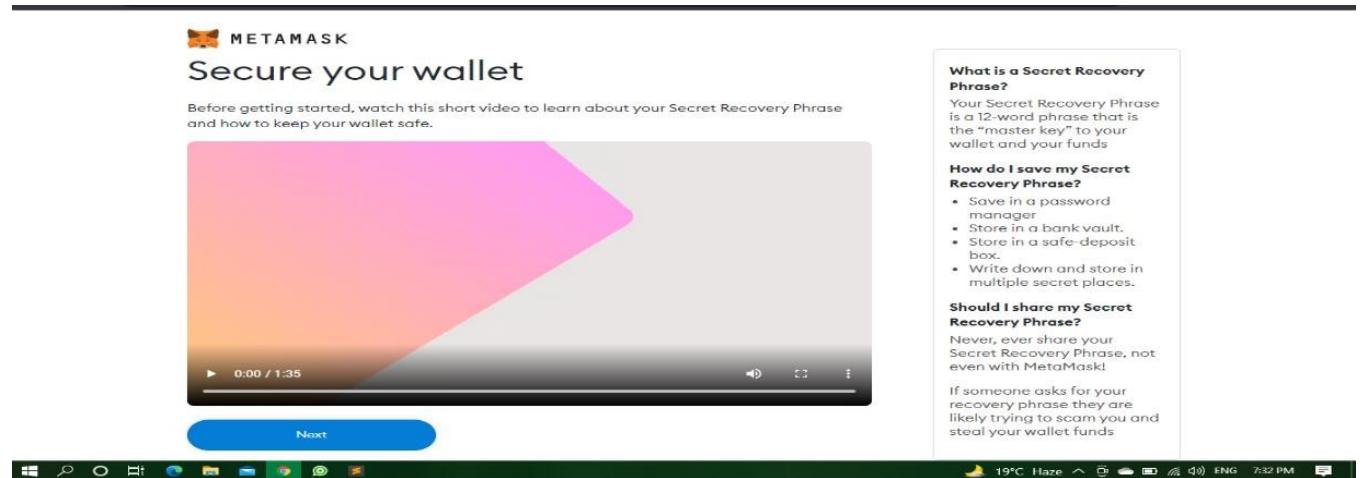
Step-6: Click on “I agree”.



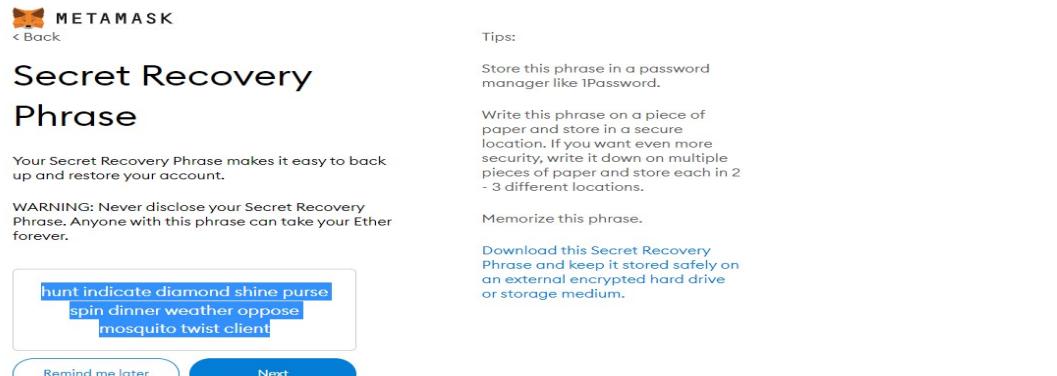
Step-7: now Create Password



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".



METAMASK
< Back

Secret Recovery Phrase

Your Secret Recovery Phrase makes it easy to back up and restore your account.

WARNING: Never disclose your Secret Recovery Phrase. Anyone with this phrase can take your Ether forever.

hunt indicate diamond shine purse
spin dinner weather oppose
mosquito twist client

[Remind me later](#) [Next](#)

Tips:

- Store this phrase in a password manager like 1Password.
- Write this phrase on a piece of paper and store in a secure location. If you want even more security, write it down on multiple pieces of paper and store each in 2 - 3 different locations.
- Memorize this phrase.
- Download this Secret Recovery Phrase and keep it stored safely on an external encrypted hard drive or storage medium.



Step-10: Confirm Secret Recovery Phrase accordingly.



METAMASK
< Back

Confirm your Secret Recovery Phrase

Please select each phrase in order to make sure it is correct.

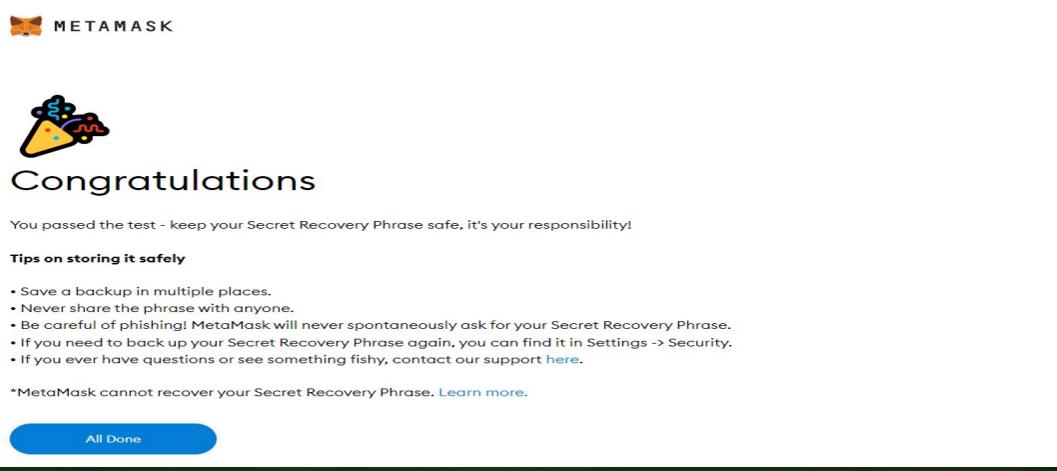
hunt	indicate	diamond	shine
purse	spin	dinner	weather
oppose	mosquito	twist	client

client	diamond	dinner	hunt
indicate	mosquito	oppose	purse
shine	spin	twist	weather

[Confirm](#)

19°C Haze ENG 7:33 PM

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



METAMASK



Congratulations

You passed the test - keep your Secret Recovery Phrase safe, it's your responsibility!

Tips on storing it safely

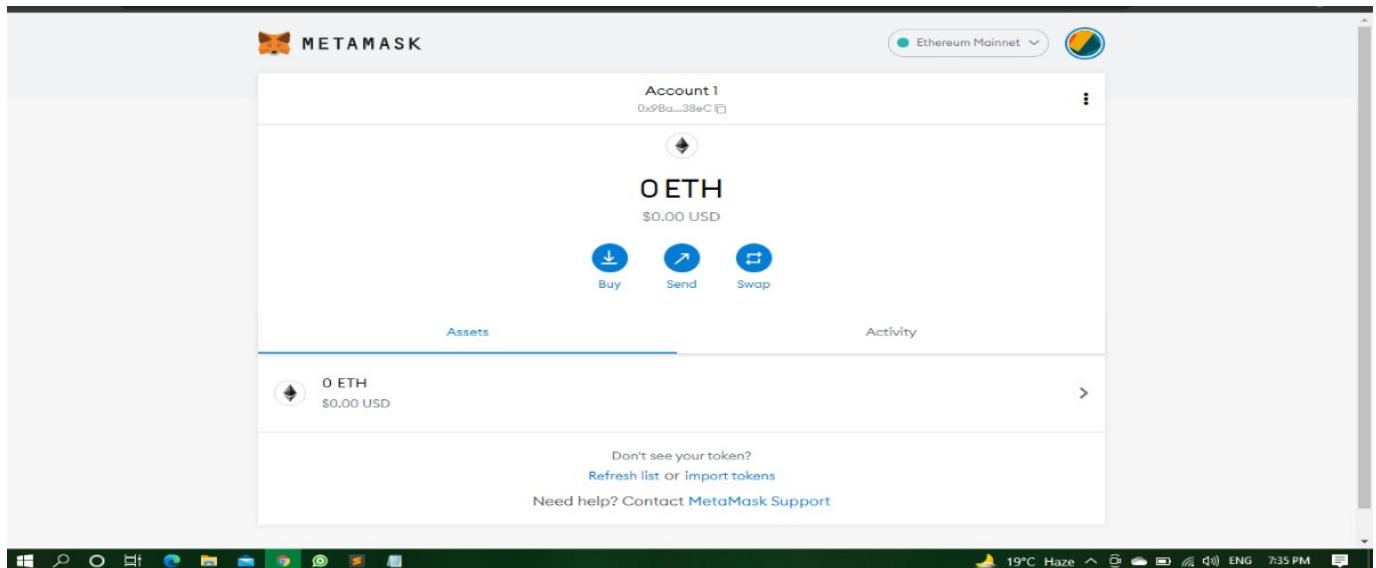
- Save a backup in multiple places.
- Never share the phrase with anyone.
- Be careful of phishing! MetaMask will never spontaneously ask for your Secret Recovery Phrase.
- If you need to back up your Secret Recovery Phrase again, you can find it in Settings → Security.
- If you ever have questions or see something fishy, contact our support [here](#).

*MetaMask cannot recover your Secret Recovery Phrase. [Learn more](#).

[All Done](#)

19°C Haze ENG 7:35 PM

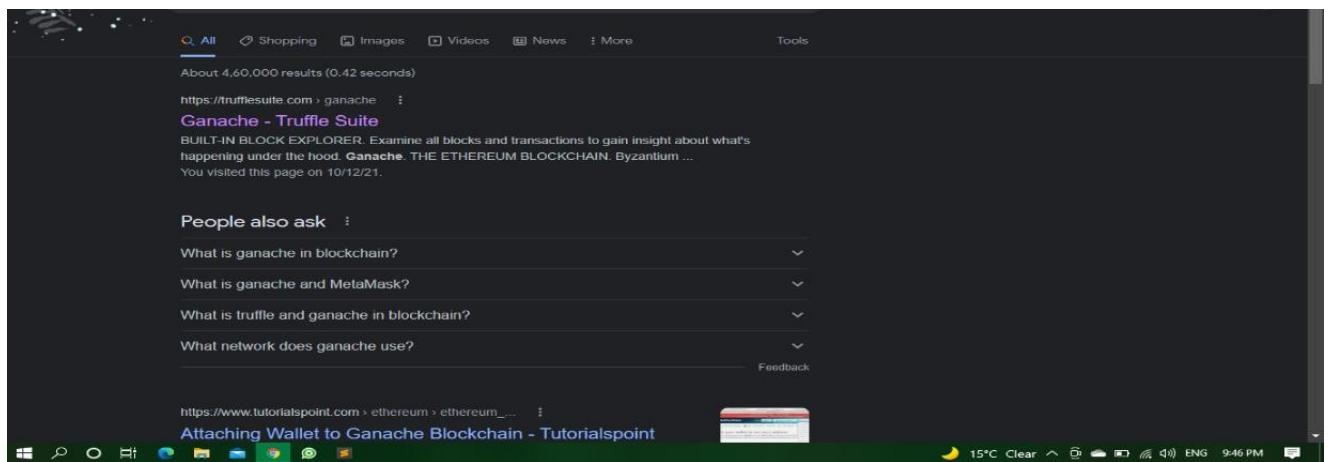
Metamask Wallet will looks like.



Practical No.2

Installation of Ganache and Import the Ethereum to Metamask wallet.

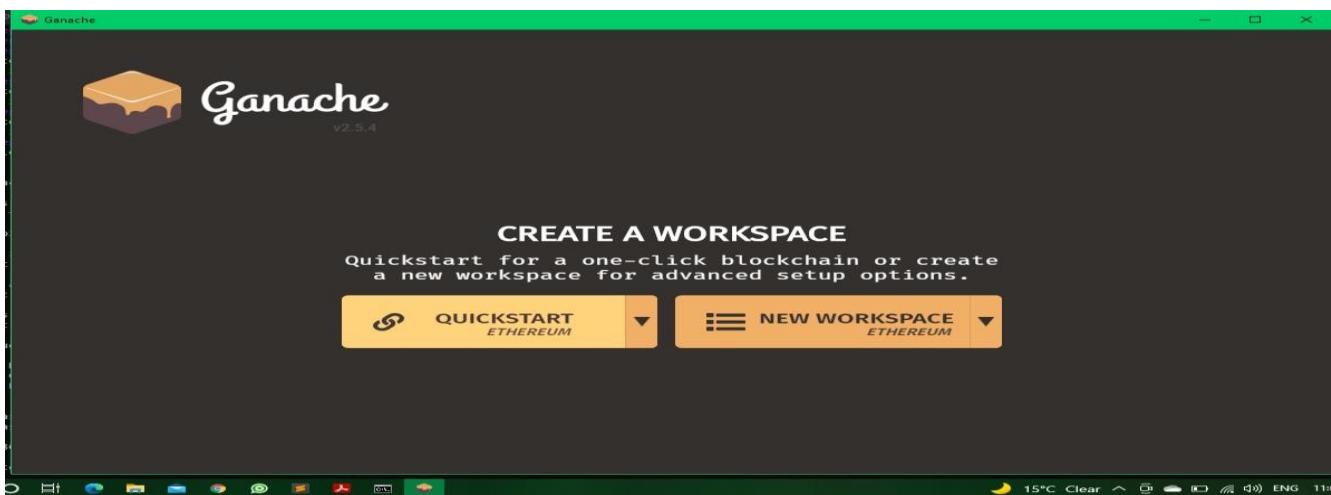
Step-1: Open your Chrome Browser and search for Ganache Wallet and Click on first link.



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.

CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER
0	20000000000	6721975	MUERGLACIER	5777	HTTP://127.0.0.1:7645
MNEMONIC tragic occur vapor pet hurdle express seek achieve heavy oyster primary sc					
ADDRESS	0x81D48A732A6D97C0618D3A376E8E897D1F138dA6	BALANCE	100.00	ETH	
ADDRESS	0x0dE925a8CEA4E5Dea2D87b9e44aF2DB4d0231102	BALANCE	100.00	ETH	
ADDRESS	0xE9d78A71e1e81E8459Ac6f8e08B1F68160d8928C	BALANCE	100.00	ETH	
ADDRESS	0xAdC287156Fe1a6b8B955Bc0d28CC6A51Cf255930	BALANCE	100.00	ETH	
ADDRESS	0xfc50Da28E7087121AFA2BacEf32be664457E3c0A	BALANCE	100.00	ETH	
ADDRESS	0x74C775351c969537E01a3fE62F82D41Be1091E66	BALANCE	100.00	ETH	
ADDRESS		BALANCE			

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.

The screenshot shows the MetaMask extension's main interface. At the top, there are tabs for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS, EVENTS, LOGS, and INFO. Below these are fields for CURRENT BLOCK, GAS PRICE, GAS LIMIT, HARDFORK, NETWORK ID, RPC SERVER, MINING STATUS, and WORKSPACE. A 'SAVE' button is on the right.

MNEMONIC: tragic occur vapor pet hurdle express seek achieve heavy oyster primary scheme

HD PATH: m/44'/60'/0'/0/account_index

Accounts:

- Address: 0x81D48A732A6D97C0618D3A376E8E897D1F138dA6, Balance: 100.00 ET, TX Count: 0, Index: 0
- Address: 0x0dE925a8CEA4E5Dea2D87b9e44aF2DB4d0231102, Balance: 100.00 ET, TX Count: 0, Index: 1
- Address: 0xE9d78A71e1e81E8459Ac6f8e08B1F68160d8928C, Balance: 100.00 ET, TX Count: 0, Index: 2
- Address: 0xAdC287156Fe1a6b8B955Bc0d28CC6A51Cf255930, Balance: 100.00 ET, TX Count: 0, Index: 3
- Address: 0xfc50Da28E7087121AFA2BacEf32be664457E3c0A, Balance: 100.00 ET, TX Count: 0, Index: 4

A message box at the bottom right says: "Contest" was successfully added! ×

The screenshot shows the MetaMask extension's import account dialog. It has a title bar with a fox icon and the text "Import Account". Below it, a message states: "Imported accounts will not be associated with your originally created MetaMask account Secret Recovery Phrase. Learn more about imported accounts [here](#)".

Select Type: Private Key

Paste your private key string here:

Buttons: Cancel (blue), Import (blue)

The screenshot shows the MetaMask extension's main interface again. The left side displays account details, including mnemonic words and private keys. The right side shows account 0x81D...8dA6 with a balance of 100 ETH. It includes buttons for Buy, Send, and Swap, and a message box indicating the success of adding the "Contest" account.

Account 3: 0x81D...8dA6

100 ETH

Buttons: Buy, Send, Swap

Assets: You have no transactions

"Contest" was successfully added! ×

Need help? Contact [MetaMask Support](#)

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

The screenshot shows the MetaMask wallet interface. At the top, it displays "Account 3" with the address "0x81D...8dA6". Below this, a large "100 ETH" balance is shown with three buttons: "Buy", "Send", and "Swap". The "Activity" tab is selected, showing the message "You have no transactions". A green notification bar at the bottom right says "Contest" was successfully added. The desktop taskbar at the bottom includes icons for File Explorer, Search, Task View, Mail, Google Chrome, and Microsoft Word.

The screenshot shows the "Send to" dialog box from MetaMask. It has a search bar with placeholder text "Search, public address (0x), or ENS" and a "Transfer between my accounts" button below it. The desktop taskbar at the bottom is identical to the one in the first screenshot, showing the same set of icons.

The image consists of three vertically stacked screenshots of the MetaMask wallet interface on a Windows desktop.

Screenshot 1 (Top): Send Transaction

This screenshot shows the "Send" transaction dialog. The recipient is "Account 1" (0x9ba5786574bcb8756db323532c094c8efd8238ec). The asset is "15 ETH" (Balance: 100 ETH). The gas price is set to 20 GWEI and the gas limit to 21000. The "Next" button is visible at the bottom.

Screenshot 2 (Middle): Transaction Confirmation

This screenshot shows the transaction confirmation dialog. It displays the amount sent (15 ETH), the estimated gas fee (0.00042 ETH), and the total amount (16.00042 ETH). The "Confirm" button is visible at the bottom.

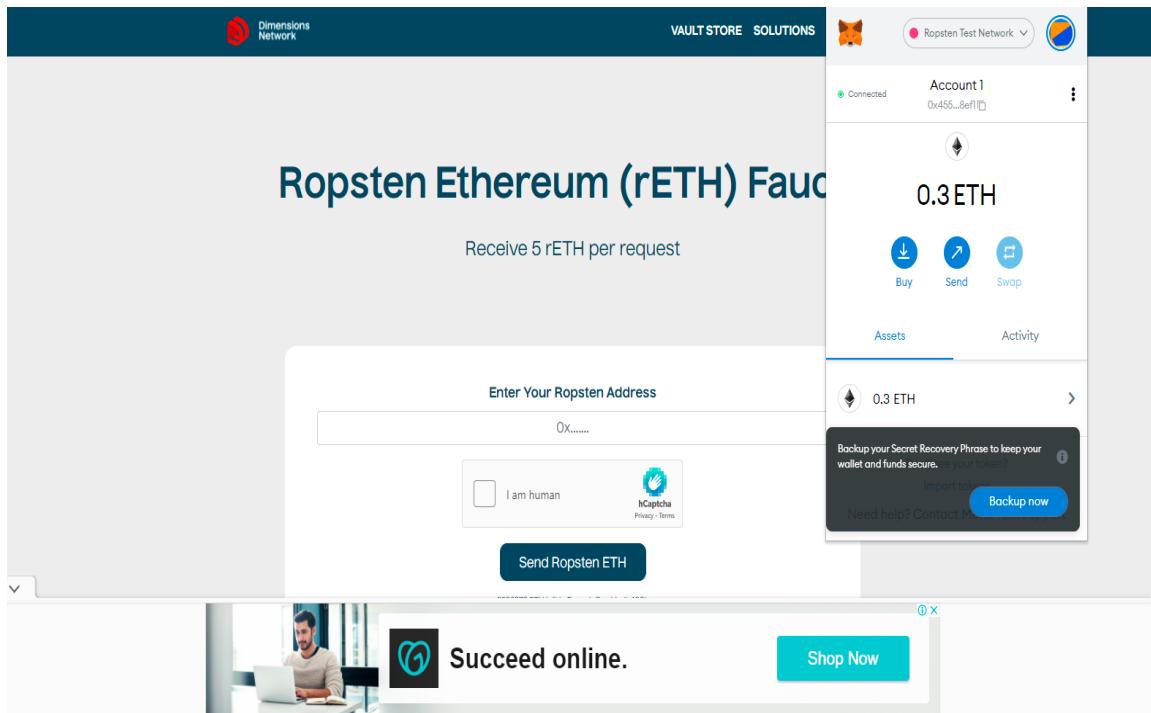
Screenshot 3 (Bottom): Account Overview

This screenshot shows the main account overview for "Account 3" (0x81D...8dA6). The balance is 84.9996 ETH. It includes "Buy", "Send", and "Swap" buttons. The "Activity" tab is selected, showing a recent transaction: "Send" to 0x9ba...38ec on Dec 22. A green notification bar at the bottom right indicates "Confirmed transaction Transaction 0 confirmed!".

Transaction has been successfully completed.

PRACTICAL -3

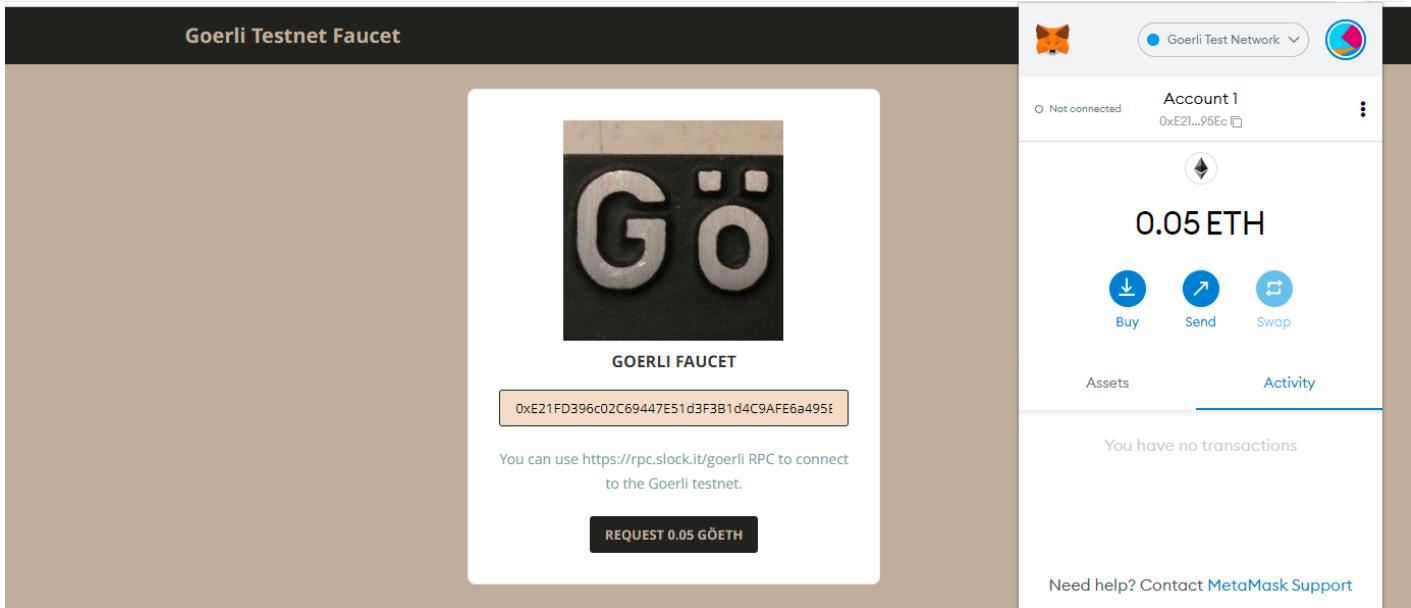
- ⑨ Firstly open your wallet and select ropston network .After that you click on buy ETH and then



click on get ether

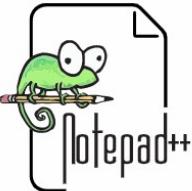
⑨

→ 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++ .



[Current Version 8.2](#)

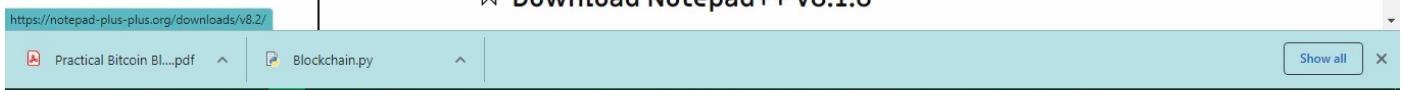
- [Home](#)
- [Download](#)
- [News](#)
- [Online Help](#)

 Adobe Creative Cloud for Teams starting at \$33.99 per month.

ADS VIA CARBON

Downloads

- [Download Notepad++ v8.2](#)
- [Download Notepad++ v8.1.9.3](#)
- [Download Notepad++ v8.1.9.2](#)
- [Download Notepad++ v8.1.9.1](#)
- [Download Notepad++ v8.1.9](#)
- [Download Notepad++ v8.1.8](#)



<https://notepad-plus-plus.org/downloads/v8.2/>



[Current Version 8.2](#)

- [Home](#)
- [Download](#)
- [News](#)
- [Online Help](#)
- [Resources](#)

 MAKE IT WITH ADOBE STOCK. GET IT NOW!

Limited time offer: Get 10 free Adobe Stock images.

ADS VIA CARBON

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information you've provided to them or they've collected from your use of their services.

[Show all](#) [Got it!](#)

DOWNLOAD 64-BIT x64



[DOWNLOAD](#)

• [Installer | GPG Signature](#)

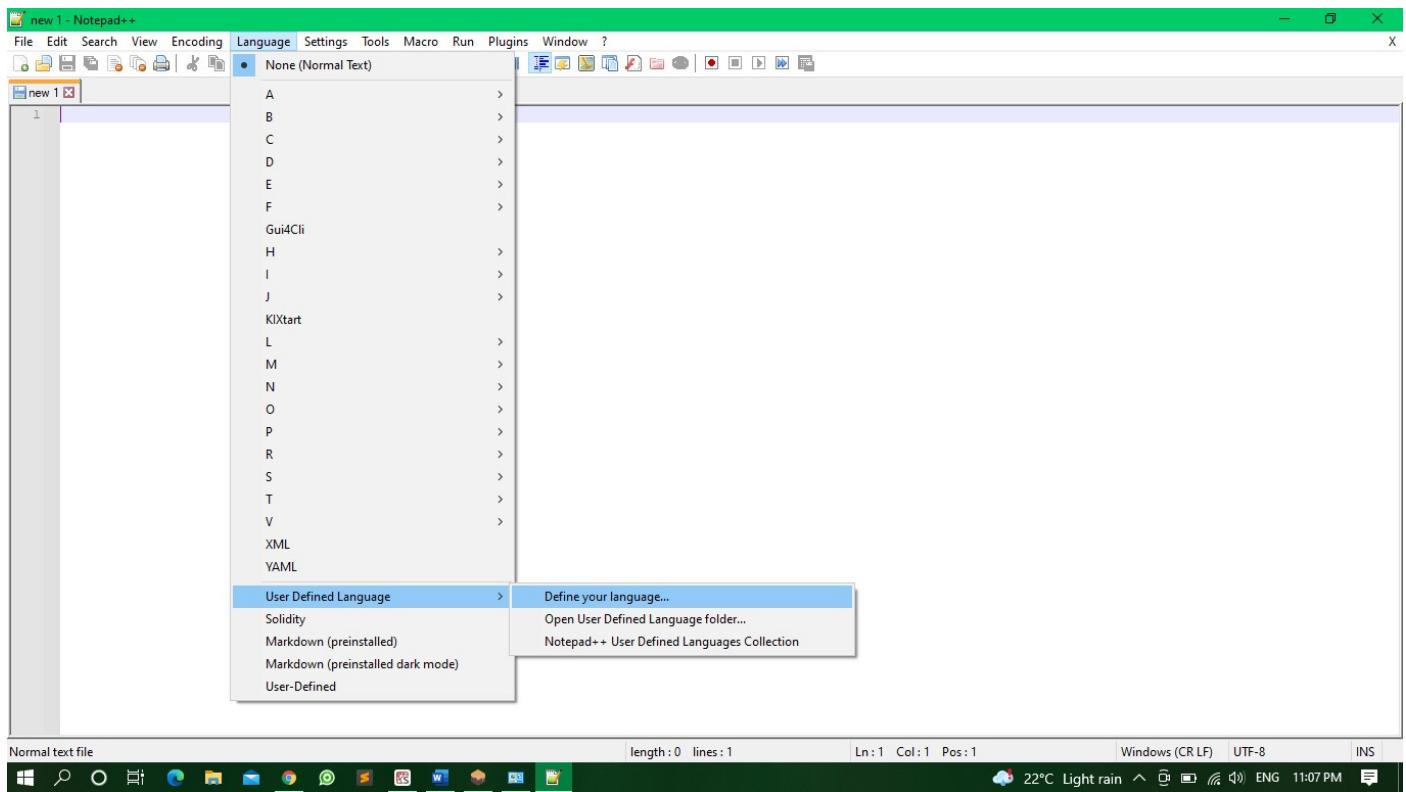
[e \(zip\) | GPG Signature](#)



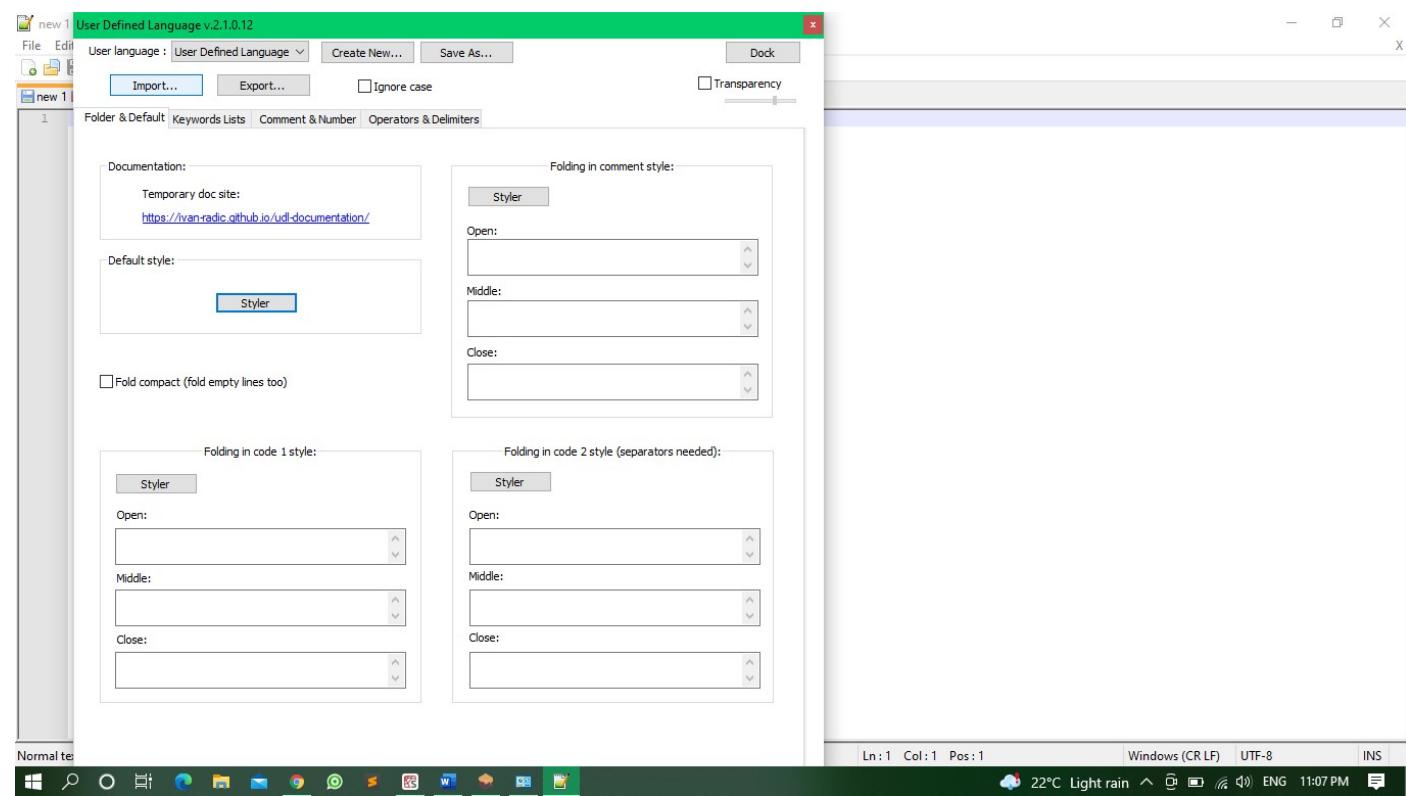
<https://github.com/notepad-plus-plus/notepad-plus-plus/releases/download/v8.2/npp.8.2.Installer.x64.exe>

Step-2: Now open Notepad++

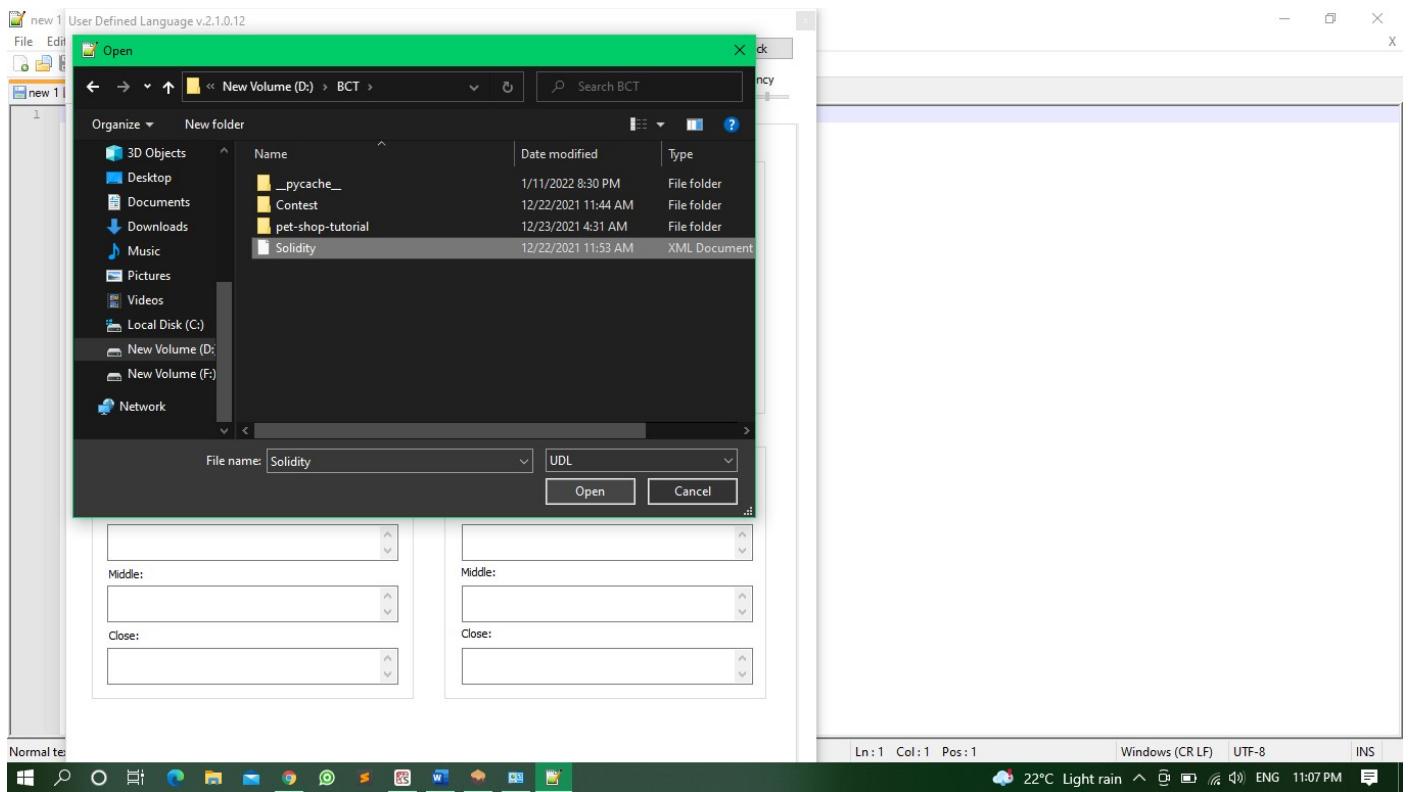
Step-3: Click on language >> User defined language >>define your language



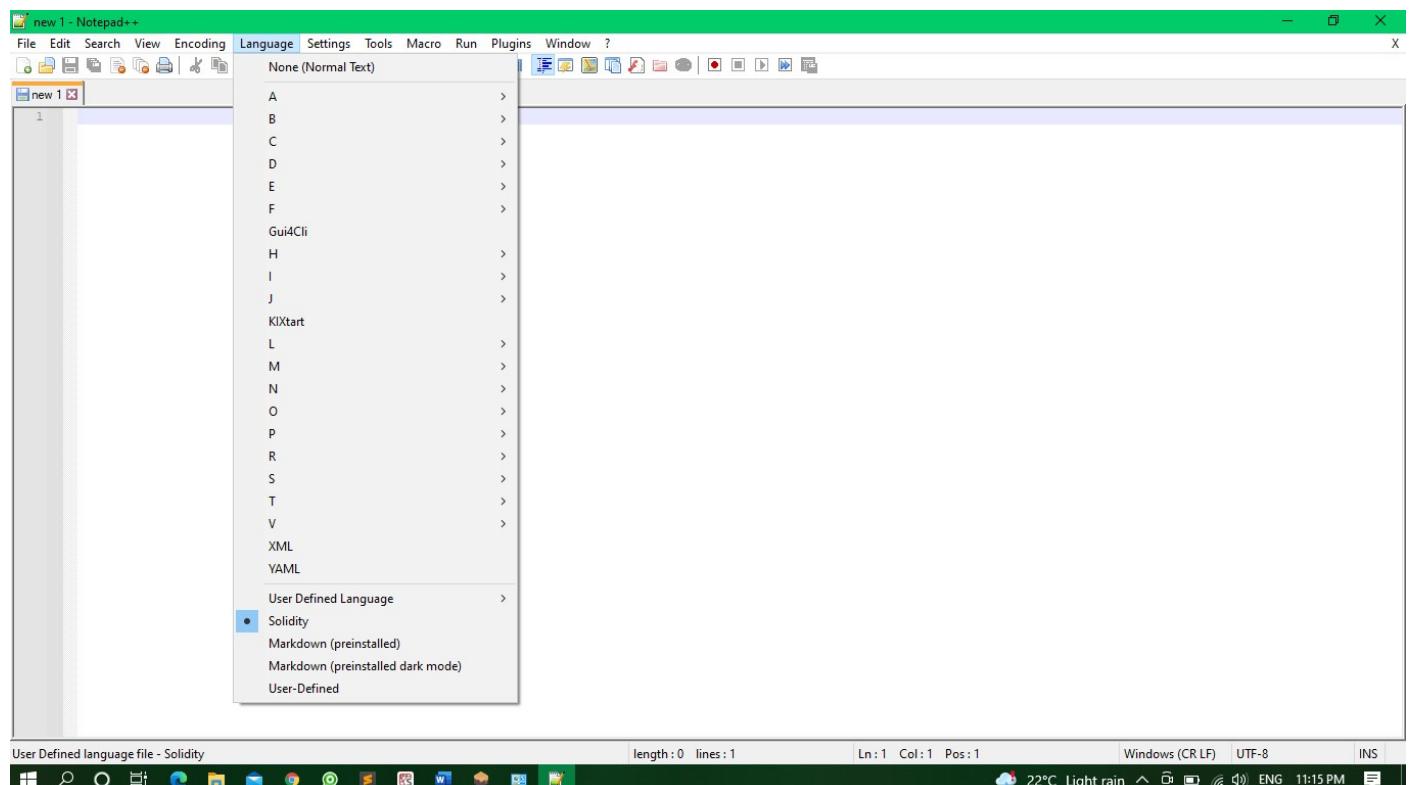
Step-4: Click on "Import"



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".

You are signed in as A171488

Blockchain Technology

Stream Classwork People

Khaleel Ahmad posted a new material: Docker Jun 2

Khaleel Ahmad posted a new material: Unit - 5 : Hyperledger Jun 1 (Edited Jul 31)

Khaleel Ahmad posted a new material: Practical Jun 1 (Edited Nov 3)

Khaleel Ahmad posted a new material: Review Papers May 26 (Edited Jul 11)

Mohammad Mubeen May 18 Ganache for windows.

Ganache-2.5.4-win-x64.a... Compressed Archive

You are signed in as A171488

Practical

Khaleel Ahmad - Jun 1 (Edited Nov 3)

<https://www.devprovider.com/category/blockchain/> // Hyperledger

Contest.zip Compressed Archive

pet-shop-tutorial.zip Compressed Archive

Truffle Suite - Truffle Suite <https://www.trufflesuite.com/tut...>

Create a Hyperledger Burro... <https://www.devprovider.com/hy...>

Python, JS, & React Build a B... Compressed Archive

How to Build a Blockchain in ... <https://www.activestate.com/blo...>

python_blockchain_tutorial....PDF

Blockchain.py Text

Create a Cryptocurrency.zip Compressed Archive

Database in Blockchain | Blo... YouTube video 7 minutes

https://drive.google.com/file/d/1HDQP10sBWGedUYVOKxxqaYKydtzABSwSv/view?usp=driveweb&sauthuser=0

You are signed in as A171488

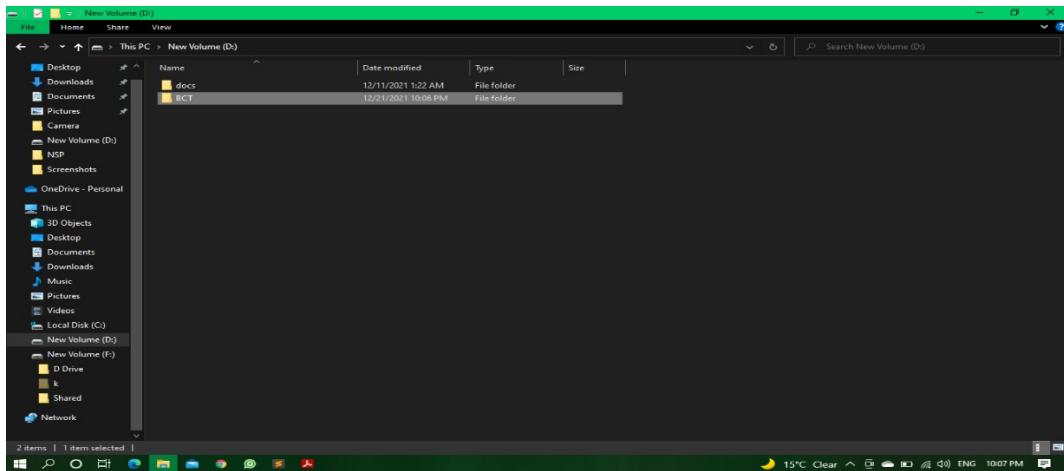
Contest.zip

Contest.zip - Google Drive

Open with ZIP Extractor

Contest.zip 1 item

Name	Last modified	File size
Contest	-	-



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.

```

C:\Users\Qasim> npm audit --force
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 cidv1@0.1.0: 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/schema@8.1.0: This package has been deprecated and now it only exports makeExecutableSchema.\nAnd it will no longer receive updates.\nWe recommend you to migrate to @graphql-tools/makeExecutableSchema.\nAnd it will no longer receive updates.\nWe recommend you to migrate to @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/graphql-tag@4.0.8: This package has been deprecated and now it only exports makeExecutableSchema.\nAnd it will no longer receive updates.\nWe recommend 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/graphql@4.0.8: This package has been deprecated and now it only exports makeExecutableSchema.\nAnd it will no longer receive updates.\nWe recommend 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 VB engine whines, feature 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-js.
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 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>

```

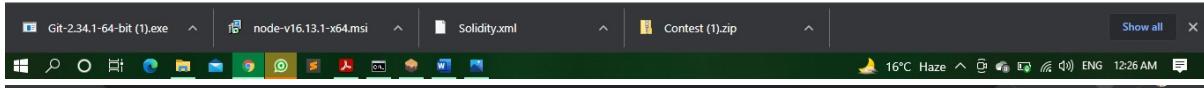
Best Actor Contest

Your Account: null

#	Name	Votes
1	Omy	0
2	Jammy	0

Select Contestant: Omy ▾

[Cast your Vote](#)



Best Actor Contest

Your Account: 0xc62b307b455f94d5a2a63962611f1cd76fcada7

#	Name	Votes
1	Omy	0
2	Jammy	0

Select Contestant: Omy ▾

[Cast your Vote](#)

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](#)





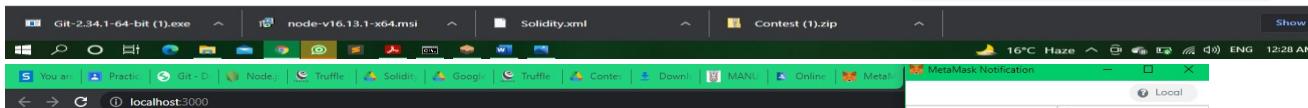
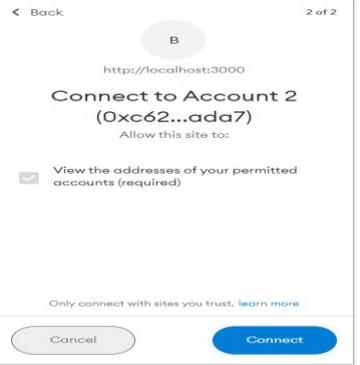
Best Actor Contest

Your Account: 0xc62b307b455f94d5a2a63962611f1cda76fcada7

#	Name	Votes
1	Omy	0
2	Jammy	0

Select Contestant: Omy

[Cast your Vote](#)



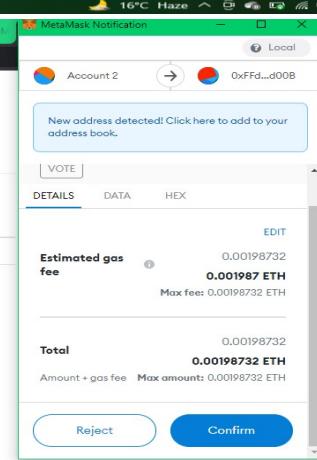
Best Actor Contest

Your Account: 0xc62b307b455f94d5a2a63962611f1cda76fcada7

#	Name	Votes
1	Omy	0
2	Jammy	0

Select Contestant: Omy

[Cast your Vote](#)



Best Actor Contest

Your Account: 0xc62b307b455f94d5a2a63962611f1cda76fcada7

#	Name	Votes
1	Omy	1
2	Jammy	0

Select Contestant: Omy

[Cast your Vote](#)



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
```

```
C:\Windows\system32\cmd.exe
E:\Blockchain>mkdir pet-shop tutorial
E:\Blockchain>cd pet-shop tutorial
E:\Blockchain\pet-shop tutorial>
E:\Blockchain\pet-shop tutorial>truffle unbox pet-shop
Unpacking unbox...
=====
Preparing to download box
Download...
npm WARN old lockfile
The package-lock.json file was created with an old version of
npm.
npm WARN old lockfile so supplemental metadata must be fetched from the registry.
npm WARN old lockfile
This is a one-time fix-up, please be patient...
npm WARN old lockfile
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated set-value@2.0.0: Critical bug fixed in v3.0.1, please upgrade to
the latest version.
npm WARN deprecated mixin-deep@1.3.1: Critical bug fixed in v2.0.1, please upgrade
to
npm WARN deprecated chokidar@2.0.4: Chokidar 2 will break on node v14+. Upgrade to
node v14+ and get chokidar@3.1.0+! Critical bug fixed in v3.0.1, please upgrade to
the latest version.
npm WARN deprecated es5-ext@17.1: Critical security vulnerability fixed in v17.1.1.
For more information, see https://github.com/lioios/es5-ext/pull/401
Cleaning up temporary files
Setting up box
unbox successful, sweet!
Commands:
  Compile: truffle compile
```

The screenshot shows a browser window titled "TRUFFLE SUITE" with the URL "trufflesuite.com/tutorials/pet-shop". The page content includes:

- A note about creating a "Truffle Box" for the project.
- A note about Truffle's initialization methods, mentioning "truffle init" and "truffle unbox".
- A "Directory structure" section with a link to "View directory structure".

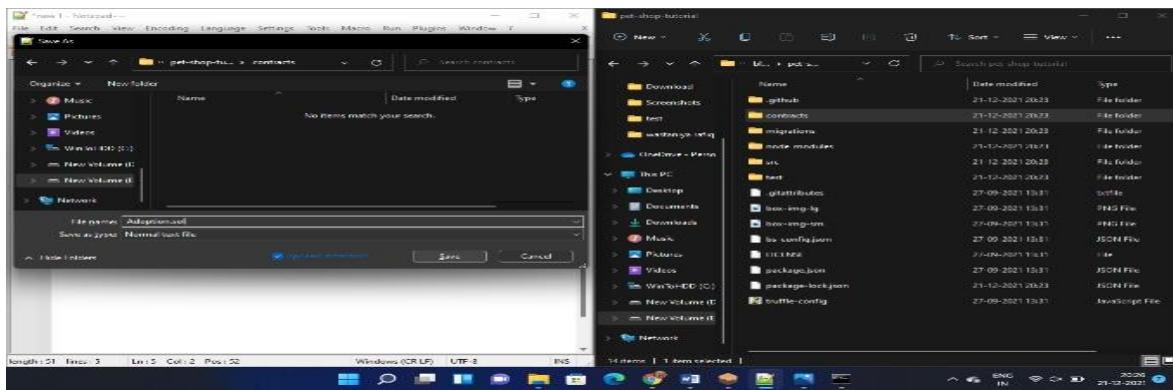
```
C:\Windows\system32\cmd.exe
E:\Blockchain>truffle vs-4.2.5 (core: 5.4.2%)
Node v12.2.0
E:\Blockchain\pet-shop tutorial>truffle compile
Compiling your contracts...
> Compiling ./contracts/Adoption.sol
> Artifacts written to E:\Blockchain\pet-shop tutorial\build\contracts
> Compiled successfully using:
  - solc: 0.5.16+commit.0c3220cc.Emscripten.clang

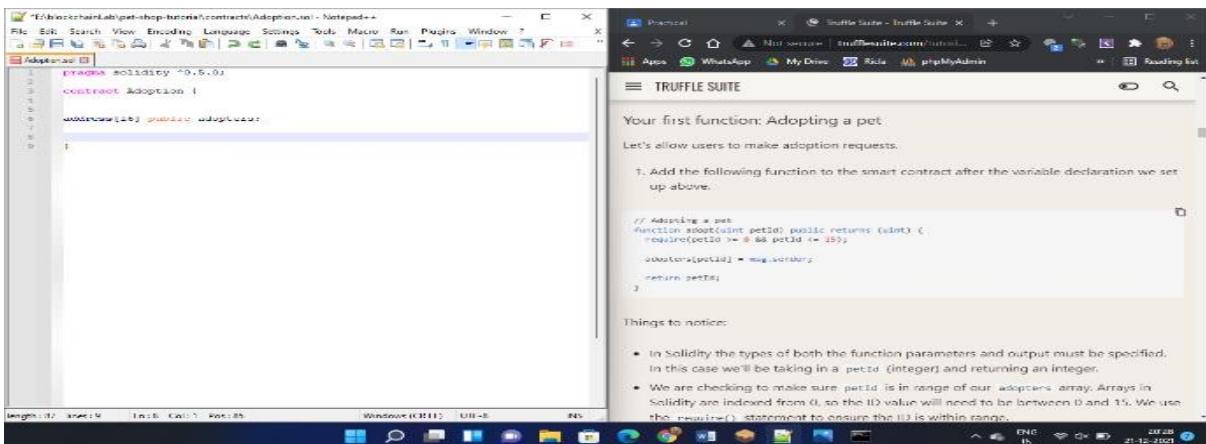
E:\Blockchain\pet-shop tutorial>
```

The screenshot shows a browser window titled "TRUFFLE SUITE" with the URL "trufflesuite.com/tutorials/pet-shop". The page content includes:

- A note about array getters returning only a single value from a given key.
- Instructions to add a "getAdopters()" function to the smart contract.
- A code snippet for the "getAdopters()" function.
- A "Things to notice:" section with two bullet points about the code and its behavior.

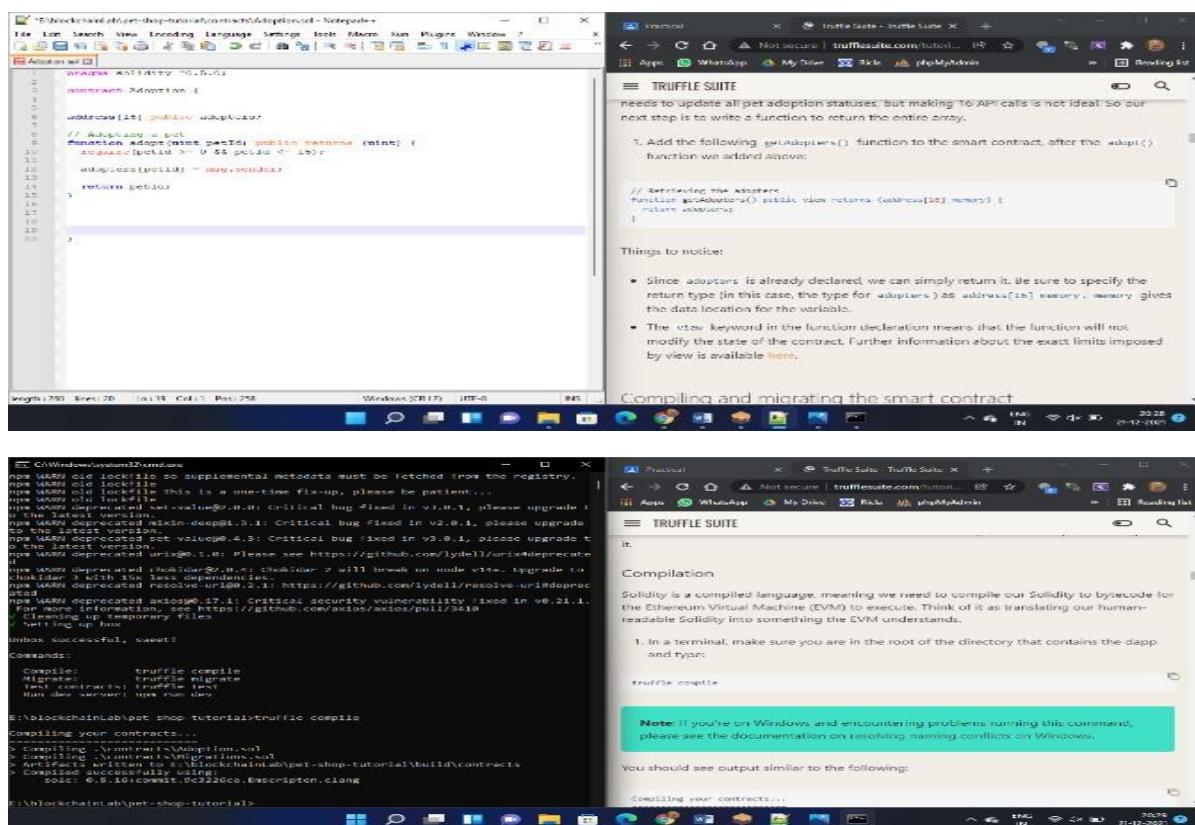
Step-2: Create a new file named Adoption.sol in the contracts/ directory, and add the following files.



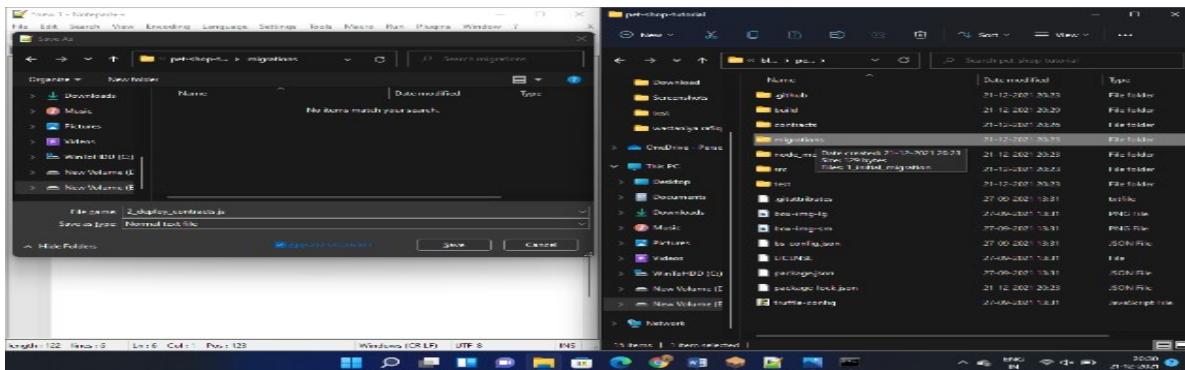


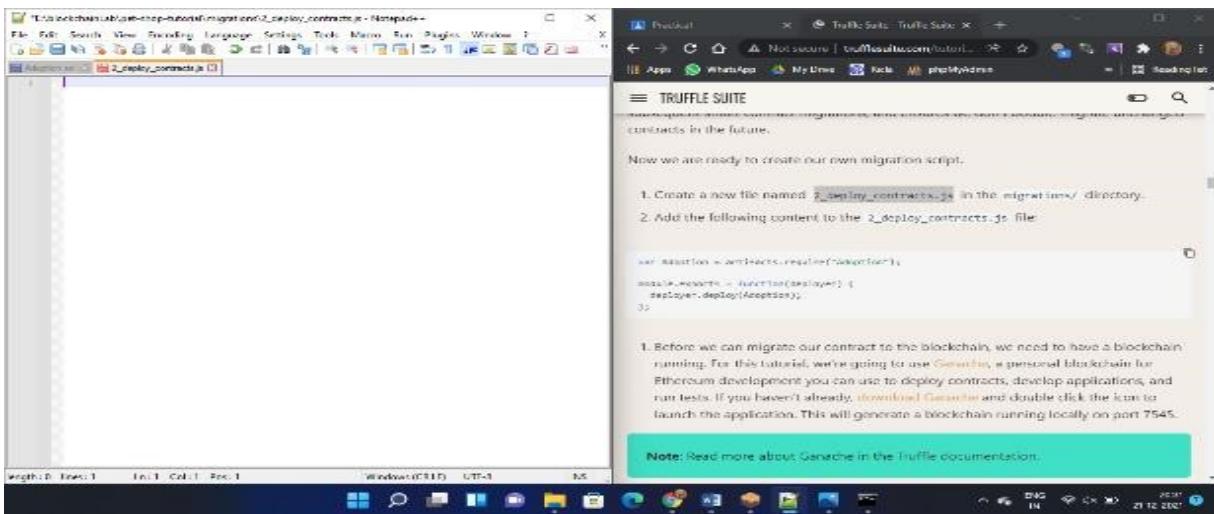
Step-3: Copy code and paste and run this command in

cmd >> truffle compile



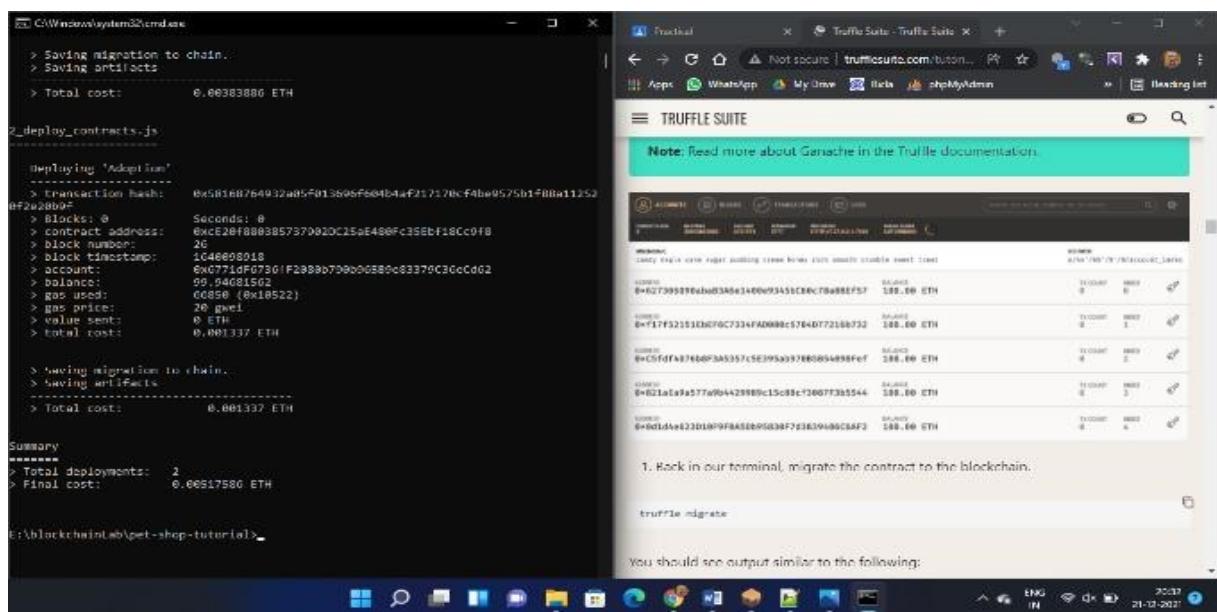
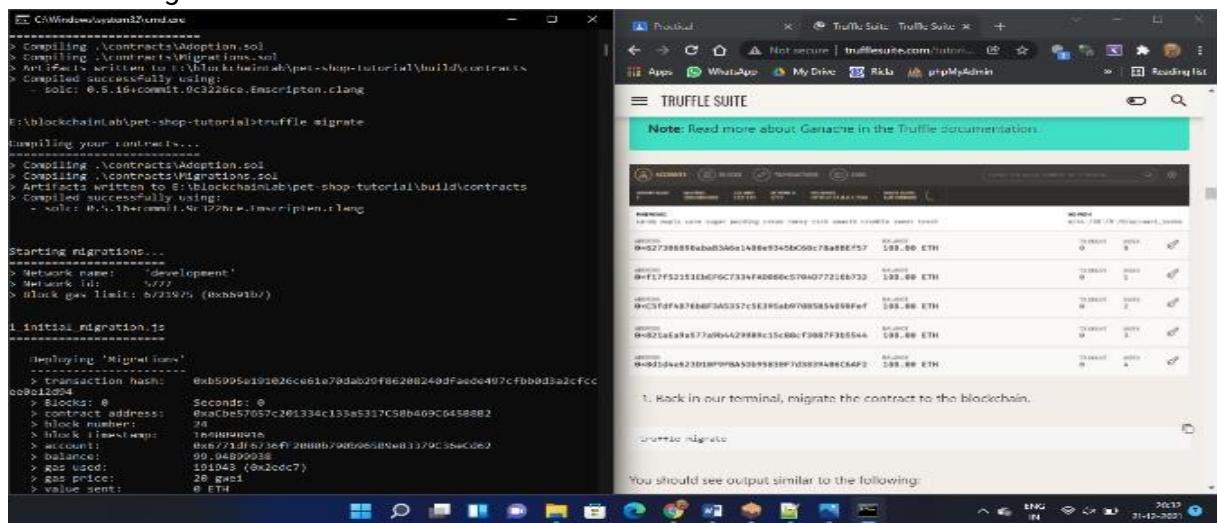
Step-4: Create a new file named `2_deploy_contracts.js` in `migrations/` directory and write the codes.



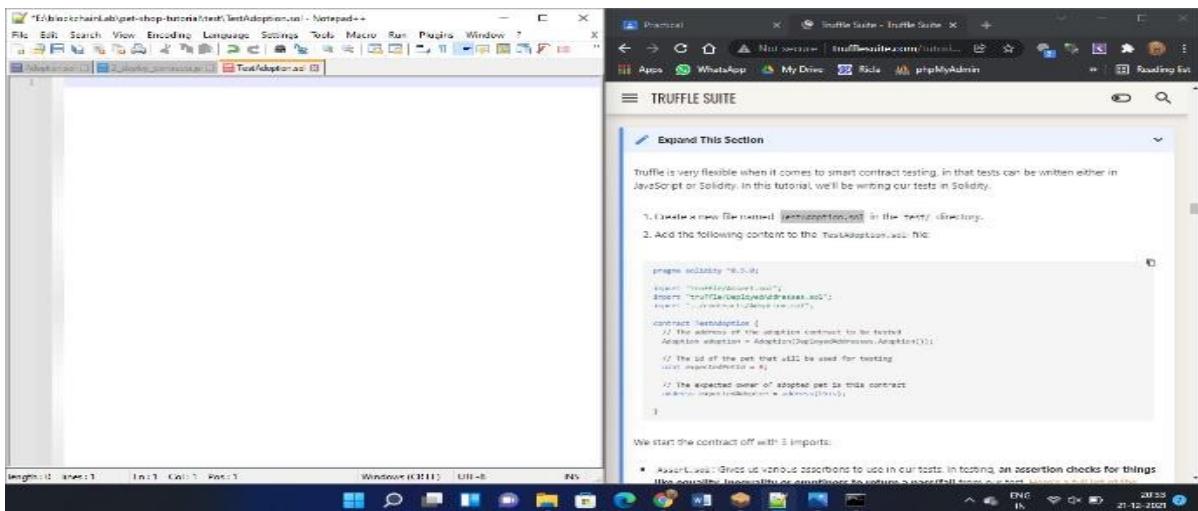
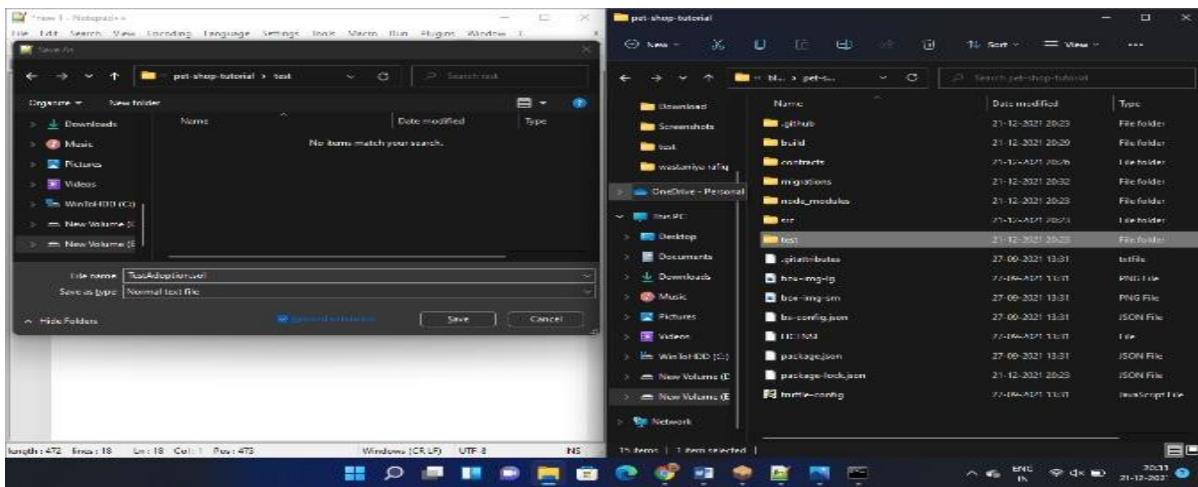


Step-5: Run this command in cmd-

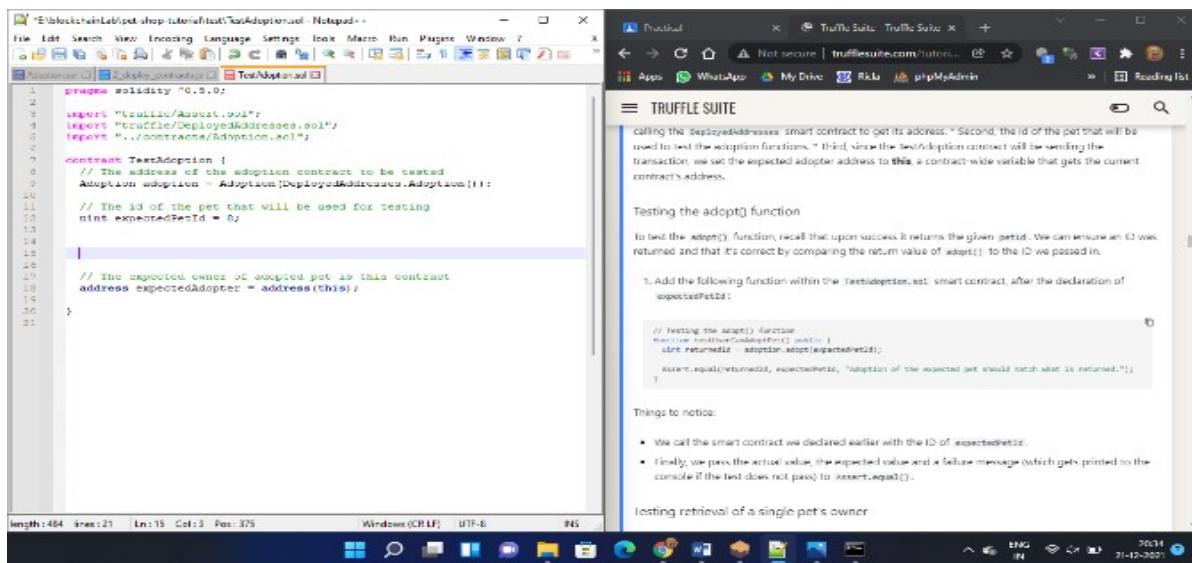
```
>> truffle migrate
```



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



```

1 pragma solidity ^0.5.0;
2
3 import "truffle/Assert.sol";
4 import "truffle/DeployedAddresses.sol";
5 import "../contracts/Adoption.sol";
6
7 contract TestAdoption {
8     // The address of the adoption contract to be tested
9     Adoption adoption = Adoption(deployedAddresses.Adoption());
10
11     // The id of the pet that will be used for testing
12     uint expectedPetId = 5;
13
14     // Testing the adopt() function
15     function testUserCanAdoptPet() public {
16         uint returnedId = adoption.adopt(expectedPetId);
17
18         Assert.equal(returnedId, expectedPetId, "Adoption of the expected pet should be successful");
19     }
20
21     // The expected owner of adopted pet is this contract
22     address expectedAdopter = address(this);
23
24 }

```

After getting the adopter address stored by the adoption contract, we assert equality as we did above.

Testing retrieval of all pet owners

Since arrays can only return a single value given a single key, we create our own getter for the entire array.

1. Add this function below the previously added function in `TestAdoption.sol`.

```

// Testing retrieval of a single pet's owner
function testGetAdopterAddressByPetId() public {
    address adopter = adoption.adopters(expectedPetId);

    Assert.equal(adopter, expectedAdopter, "Owner of the expected pet should be this contract");
}

```

After getting the adopter address stored by the adoption contract, we assert equality as we did above.

Testing retrieval of all pet owners

Since arrays can only return a single value given a single key, we create our own getter for the entire array.

1. Add this function below the previously added function in `TestAdoption.sol`.

```

// Testing retrieval of all pet owners
function testGetAdopterAddressByPetIdInArray() public {
    // Store adopters in memory rather than contract's storage
    address[5] memory adopters = adoption.getAdopters();

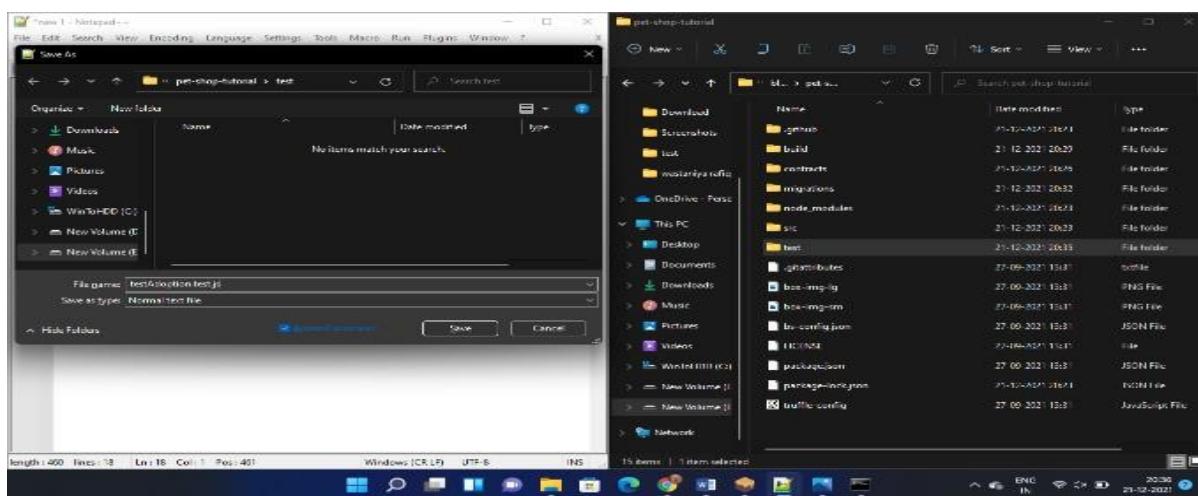
    Assert.equal(adopters[expectedPetId], expectedAdopter, "Owner of the expected pet should be this contract")
}

```

Note the **memory** attribute on `adopters`. The **memory** attribute tells Solidity to temporarily store the value in memory, rather than saving it to the contract's storage. Since `adopters` is an array, and we know from the first adoption test that we adopted pet `expectedPetId`, we compare the testing contracts address with location `expectedPetId` in the array.

Testing the smart contract using JavaScript

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



The screenshot shows a Windows desktop environment. On the left, a Notepad window displays the file `E:\blockchainLab\pet-shop-tutorial\test\testAdoption.test.js`. The code is a JavaScript file for testing the `Adoption` contract. It includes imports for `chai` and `assert`, and defines a test function `describe("Adopting a pet and retrieving account addresses", async () => {`. Inside this function, it sets up accounts, deploys the `Adoption` contract, and performs an adoption. The adoption is successful, and the expected adopter's address is checked against the account at index 0. On the right, a web browser window shows a Truffle Suite tutorial. It provides instructions for creating a test file named `testAdoption.test.js` and adding content to it. The code shown in the browser matches the code in the Notepad window.

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.

The screenshot shows a Windows desktop environment. On the left, the Notepad window now contains the following code, which adds functionality to the adoption process:

```

    ...
    let adoption;
    let expectedAdopter;

    before(async () => {
      adoption = await Adoption.deployed();
    });

    describe("adopting a pet and retrieving account addresses", async () => {
      let petId = 1;
      let owner = accounts[0];
      let adopter = accounts[1];
      let expectedAdopter = accounts[0];

      ...
    });
  
```

On the right, the Truffle Suite tutorial continues with instructions for the `accept` function. It suggests adding a function to the `testAccept.js` file after the `before` block. The code provided in the browser is:

```

    ...
    describe("accepting a pet and retrieving account addresses", async () => {
      let petId = 1;
      let owner = accounts[0];
      let adopter = accounts[1];
      let expectedAdopter = accounts[0];

      it("lets fetch the address of an owner by pet id", async () => {
        const ownerAddress = await Adoption.owner(petId);
        expect(ownerAddress).to.equal(owner);
      });
    });
  
```

The screenshot shows a Windows desktop environment. On the left, the Notepad window contains the following code, which includes a function to fetch the owner of a pet by its ID:

```

    ...
    let adoption;
    let expectedAdopter;

    before(async () => {
      adoption = await Adoption.deployed();
    });

    describe("adopting a pet and retrieving account addresses", async () => {
      let petId = 1;
      let owner = accounts[0];
      let adopter = accounts[1];
      let expectedAdopter = accounts[0];

      ...
    });

    it("lets fetch the address of an owner by pet id", async () => {
      const ownerAddress = await Adoption.owner(petId);
      expect(ownerAddress).to.equal(owner);
    });
  
```

On the right, the Truffle Suite tutorial continues with instructions for testing the removal of all pet owners. It suggests adding a function to the `testAccept.js` file. The code provided in the browser is:

```

    ...
    describe("lets fetch the collection of all pet owners addresses", async () => {
      const ownerAddress = await Adoption.getOwners();
      expect(ownerAddress[0]).to.equal(adopter);
      expect(ownerAddress[1]).to.equal(expectedAdopter);
    });
  
```

Step-10: Run this command in cmd-

>> truffle test

```
C:\Windows\system32\cmd.exe

> Saving migration to chain.
> Saving artifacts
=====
> Total cost: 0.00383886 ETH

2_deploy_contracts.js
=====
Deploying 'Adoption'
=====
> transaction hash: 0x58167864932a0f013696f604b4af217170cf4be9575b1f88a11252
0f2e20b9f
> blocks: 0
> contract address: 0x1cE20fF880385737902DC25aE480Fc35Ebfb18C9f8
> block number: 26
> block timestamp: 1640089818
> account: 0x6771df6736ff2080b790b96589e83379C36eCd62
> balance: 99.94681562
> gas used: 668950 (0x10522)
> gas price: 20 gwei
> value sent: 0 ETH
> total cost: 0.001337 ETH

> Saving migration to chain.
> Saving artifacts
=====
> Total cost: 0.001337 ETH

Summary
=====
> Total deployments: 2
> Final cost: 0.00517586 ETH

E:\blockchainLab\pet-shop-tutorial>truffle test

Practical Truffle Suite - Truffle Suite
Not secure | trufflesuite.com/tutori... Apps WhatsApp My Drive Rical phpMyAdmin > Reading list
=====

TRUFFLE SUITE
Running the tests

1. Back in the terminal, run the tests:

truffle test

1. If all the tests pass, you'll see console output similar to this:

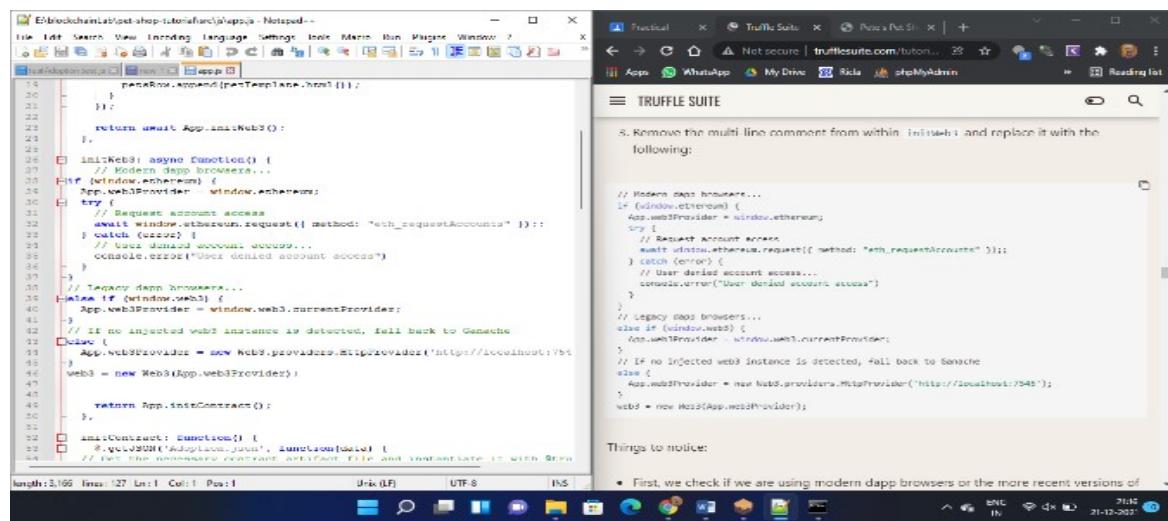
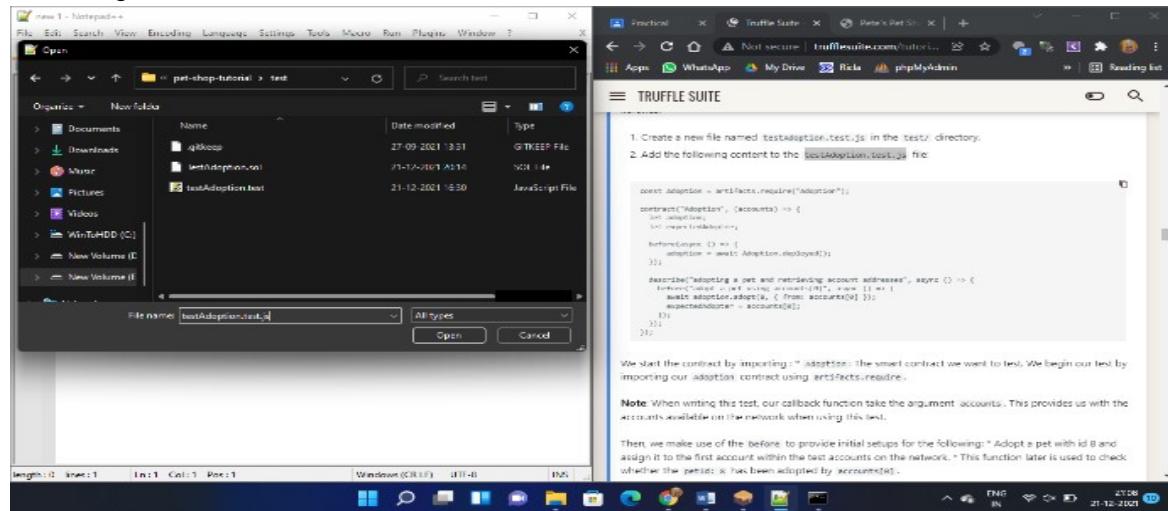
Using network 'development'.

Compiling your contracts...
=====
> Compiling ./TestAdoption.sol
> Artifacts written to /var/folders/z3/v8sd04ys11q2sh8tq38mz30c0000gn/T/test-11934-19747-49sra
> Compiled successfully using:
- solc: 0.5.0-commit.1d4f5f59a.Emscripten.clang

TestAdoption
✓ testUserCanAdoptPet (91ms)
✓ testGetAdopterAddressByPetId (70ms)
✓ testGetAdopterAddressByPetIdInArray (89ms)

3 passing (670ms)
```

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

```

// Get the provider for our contract
App.contracts.Adoption.setProvider(App.web3Provider);

// Use our contract to retrieve and mark the adopted pets
return App.markAdopted();
}

// Bind events
App.bindEvents();

```

Now that we can interact with Ethereum via web3, we need to instantiate our smart contract so web3 knows where to find it and how it works. Truffle has a library to help with this called `@truffle/contract`. It keeps information about the contract in sync with migrations, so you don't need to change the contract's deployed address manually.

1. Still in `/src/js/app.js`, remove the multi-line comment from within `initContract` and replace it with the following:

```

// Load JSON file with ABI
const AdoptionJSON = require('../build/contracts/Adoption.json');
const Adoption = artifacts.require(AdoptionJSON);

```

Things to notice:

- We first retrieve the artifact file for our smart contract. **Artifacts are Information about our contract such as its deployed address and Application Binary Interface (ABI)**. The ABI is a JavaScript object defining how to interact with the

```

// Set the provider for our contract
App.contracts.Adoption.setProvider(App.web3Provider);

// Use our contract to retrieve and mark the adopted pets
return App.markAdopted();
}

// Bind events
App.bindEvents();

```

Getting The Adopted Pets and Updating The UI

1. Still in `/src/js/app.js`, remove the multi-line comment from `markAdopted` and replace it with the following:

```

var adoptionInstance;

App.contracts.Adoption.deployed().then(function(instance) {
  adoptionInstance = instance;
})

```

Things to notice:

- We access the deployed `Adoption` contract, then call `getAdopters()` on that instance.

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 .

```

handleAdopt: function(event) {
  event.preventDefault();
  var petId = parseInt($('#event.target').data('id'));
  var adoptionInstance;
  web3.eth.getAccounts(function(error, accounts) {
    if (error) {
      console.log(error);
    } else {
      var account = accounts[0];
      App.contracts.Adoption.deployed().then(function(instance) {
        adoptionInstance = instance;
        // Execute adopt as a transaction by sending account
        // and transaction details to the adopt(petId, {from: account});
        let txObject = {
          from: account,
          value: 0,
          data: adoptionInstance.methods.adopt(petId).encodeFunctionData(),
          gas: 500000
        };
        App.contracts.Adoption.deployed().then(function(instance) {
          instance.adopt(txObject);
        });
      });
    }
  });
}

```

Handling the `adopt()` Function

1. Still in `/src/js/app.js`, remove the multi-line comment from `handleAdopt` and replace it with the following:

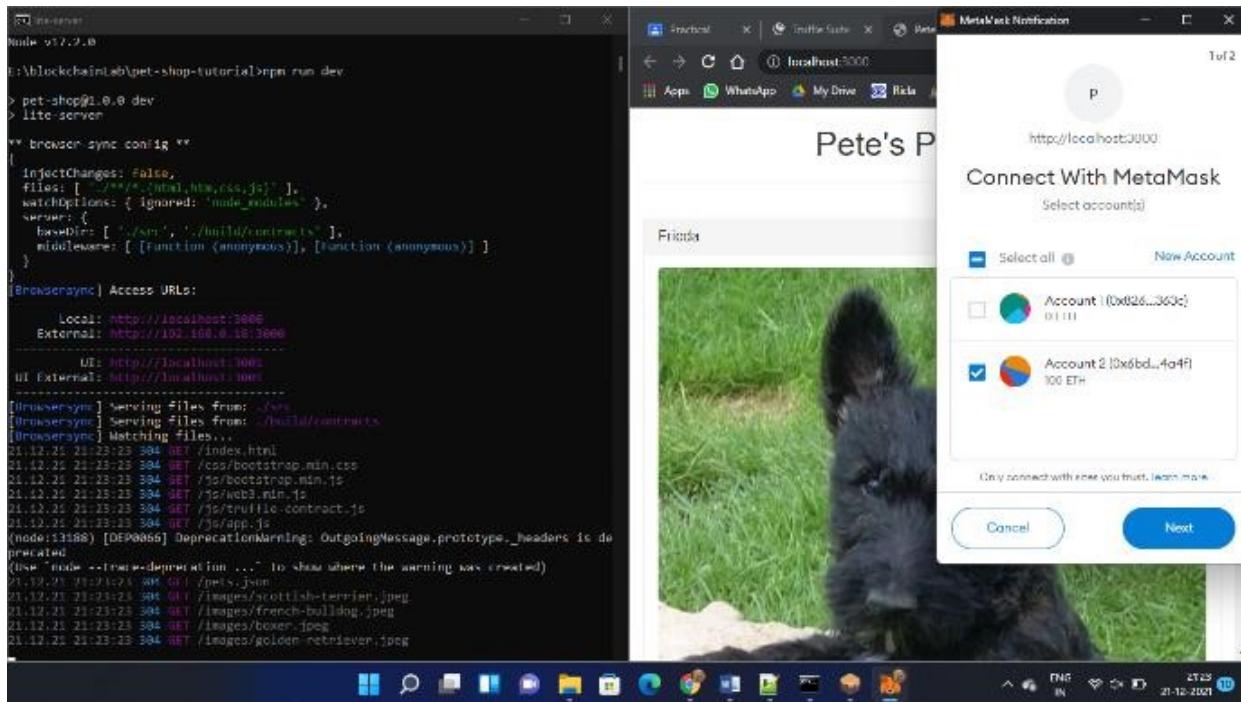
```

var adoptionInstance;

web3.eth.getAccounts(function(error, accounts) {
  if (error) {
    console.log(error);
  }
  var account = accounts[0];
  App.contracts.Adoption.deployed().then(function(instance) {
    adoptionInstance = instance;
    // Should adopt as a transaction by sending account
    return adoptionInstance.adopt(petId, {from: account});
  }).then(function(result) {
    return App.newAdopted();
  }).catch(function(error) {
    console.log(error.message);
  });
})

```

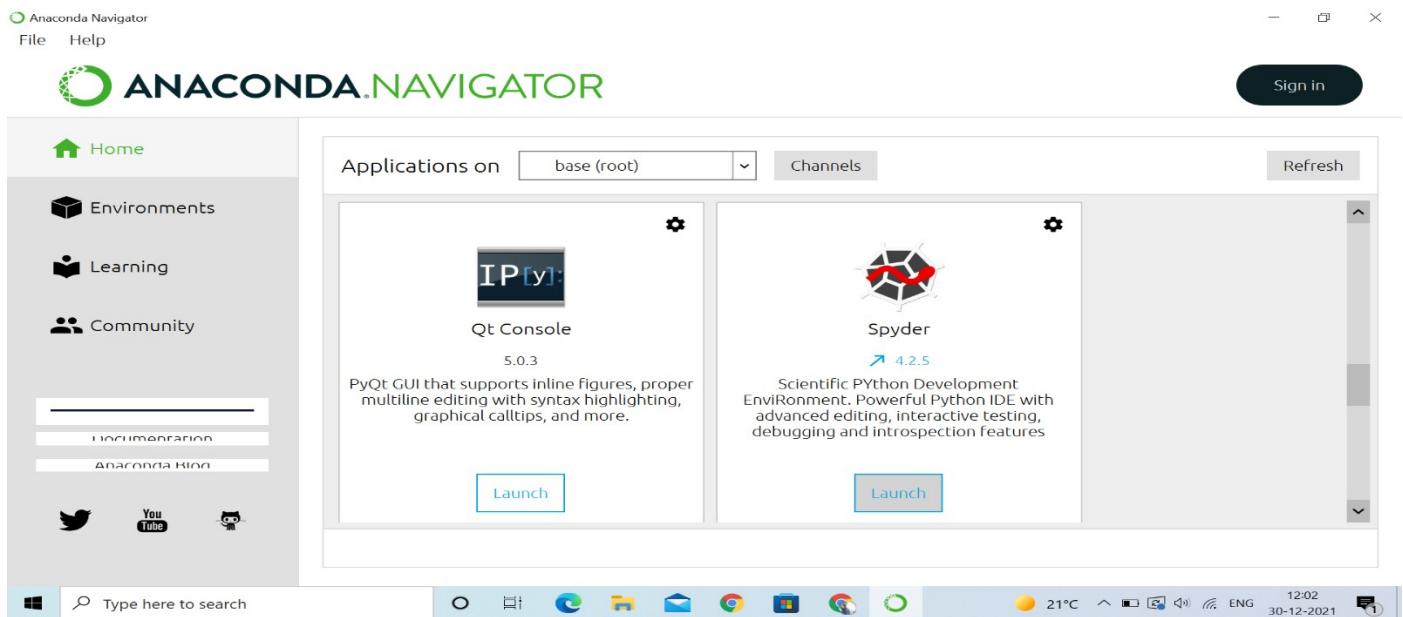
Things to notice:



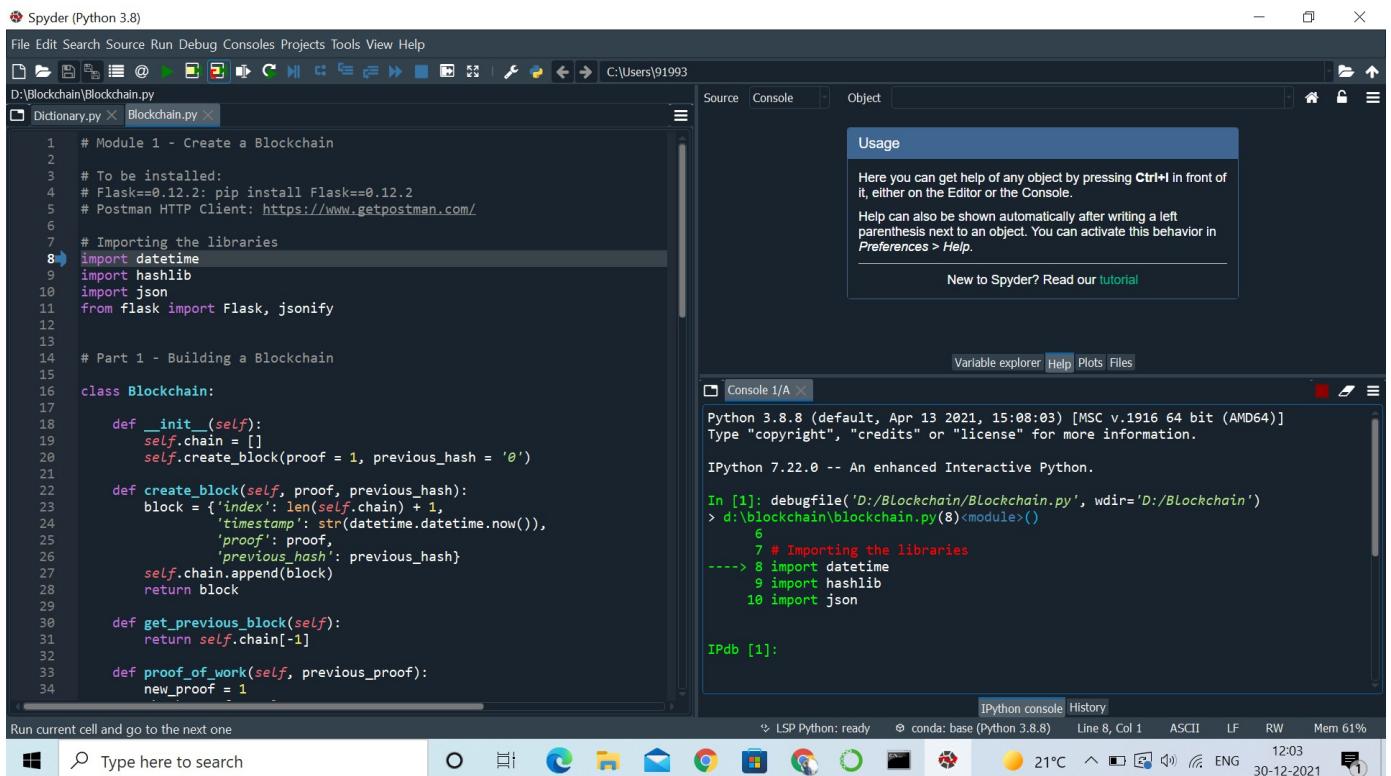
Practical No.8

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



The screenshot shows the Spyder Python IDE interface. On the left, a code editor displays a Python script named `Blockchain.py`. The script defines a `Blockchain` class with methods for creating blocks and getting previous blocks. It uses the `datetime`, `hashlib`, and `jsonify` modules from Flask. The right side of the interface features a "Usage" help panel and an IPython console window titled "Console 1/A". The console shows the execution of the script, including importing libraries and running the Flask app on port 5000. The system tray at the bottom indicates it's 12:03 on December 30, 2021, with a temperature of 21°C.

```

1 # Module 1 - Create a Blockchain
2
3 # To be installed:
4 # Flask==0.12.2: pip install Flask==0.12.2
5 # Postman HTTP Client: https://www.getpostman.com/
6
7 # Importing the libraries
8 import datetime
9 import hashlib
10 import json
11 from flask import Flask, jsonify
12
13
14 # Part 1 - Building a Blockchain
15
16 class Blockchain:
17
18     def __init__(self):
19         self.chain = []
20         self.create_block(proof = 1, previous_hash = '0')
21
22     def create_block(self, proof, previous_hash):
23         block = {'index': len(self.chain) + 1,
24                  'timestamp': str(datetime.datetime.now()),
25                  'proof': proof,
26                  'previous_hash': previous_hash}
27         self.chain.append(block)
28         return block
29
30     def get_previous_block(self):
31         return self.chain[-1]
32
33     def proof_of_work(self, previous_proof):
34         new_proof = 1

```

Step 3 : Open Postman and click on My Workspace.

The screenshot shows the Postman application interface. The left sidebar has a navigation menu with options like Home, Workspaces, API Network, Reports, and Explore. The main area is titled "My Workspace" and contains sections for Overview, Activity, and In this workspace. The Overview section includes a summary, activity feed, and environment settings. The Activity section shows a recent creation by "mohammad asif" on November 24, 2021, at 4:38 PM. The "In this workspace" section lists Requests (1), Collections (1), APIs (0), Environments (0), Mock Servers (0), and Monitors (0). The system tray at the bottom indicates it's 12:05 on December 30, 2021, with a temperature of 21°C.

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

The screenshot shows the Postman application interface. On the left, there's a sidebar with various options like Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main area displays a collection named "Media Authentication API" which contains several requests. A specific request for "GET /get_chain" is selected. The "Body" tab of the request details shows a JSON response:

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
{
  "chain": [
    {
      "index": 1,
      "previous_hash": "0",
      "proof": 1,
      "timestamp": "2022-01-11 23:57:43.153590"
    }
  ],
  "length": 1
}
```

The status bar at the bottom indicates a 200 OK response with 14 ms latency and 252 B size.

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.

The screenshot shows the Postman application interface. The sidebar is similar to the previous one, showing "My Workspace" with a "New Collection" option. A request for "GET /mine_block" is selected. The "Body" tab of the request details shows a JSON response:

```
1 2 3 4 5 6
{
  "index": 2,
  "message": "Congratulations, you just mined a block!",
  "previous_hash": "7011fb74e4ecb3290051ac37adc5dfc160306b2985ad4f4a483f79b93f408923",
  "proof": 533,
  "timestamp": "2021-12-30 12:05:39.700283"
}
```

The status bar at the bottom indicates a 200 OK response with 22 ms latency and 347 B size.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

Overview GET http://127.0.0.1:50... + No Environment

My Workspace New Import Overview GET http://127.0.0.1:5000/mine_block

GET http://127.0.0.1:5000/mine_block

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

Pretty Raw Preview Visualize JSON

```
1
2
3
4
5
6
```

"index": 3,
"message": "Congratulations, you just mined a block!",
"previous_hash": "691f027f1b0aa810ecf5242766b11bee4041e325db8974fb5aa0b6a3fc17547",
"proof": 45293,
"timestamp": "2021-12-30 12:06:02.268888"

Find and Replace Console Bootcamp Runner Trash 12:06 30-12-2021

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

Overview GET http://127.0.0.1:50... + No Environment

My Workspace New Import Overview GET http://127.0.0.1:5000/mine_block

GET http://127.0.0.1:5000/mine_block

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

Pretty Raw Preview Visualize JSON

```
1
2
3
4
5
6
```

"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 12:06 30-12-2021

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

My Workspace New Import Overview GET http://127.0.0.1:50... + No Environment

Collections APIs Environments Mock Servers Monitors Flows History

http://127.0.0.1:5000/get_chain

GET http://127.0.0.1:5000/get_chain

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

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
"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

Type here to search

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

My Workspace New Import Overview GET http://127.0.0.1:50... + No Environment

Collections APIs Environments Mock Servers Monitors Flows History

http://127.0.0.1:5000/get_chain

GET http://127.0.0.1:5000/get_chain

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

```
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
  ],
  "length": 4
```