**Program:**

**Develop an Election contract using solidity programming. Create a struct called Candidate, the struct members are ID, name and the vote-count. The smart contract should have the functions like addcandidate, show-candidates, vote, candidatescount and the voters function to verify the status of the casted vote using the Ethereum account address. Further, compile the contract and deploy to the personal Blockchain network using Ganache.**

**STEP 1**

Setup a Ganache as your personal blockchain for Ethereum development. It will allow you to

deploy smart contracts, develop applications, and run tests.

**Ganache Personal Blockchain Interface consist of:**

➢ Accounts Page -this shows you all of the accounts that are automatically generated, along with

their balances.

➢ Blocks Page - this shows you each block that has been mined on the personal blockchain network,

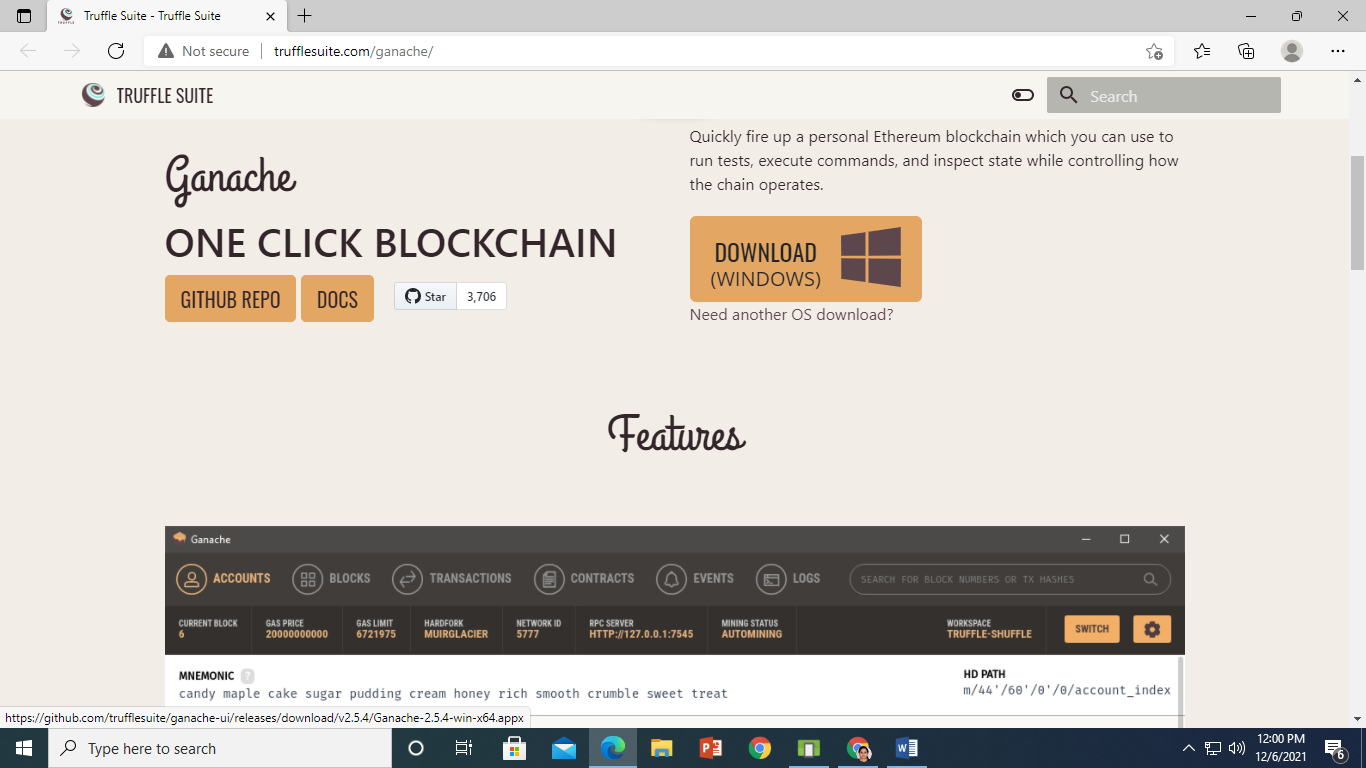
along with the gas cost and transactions.

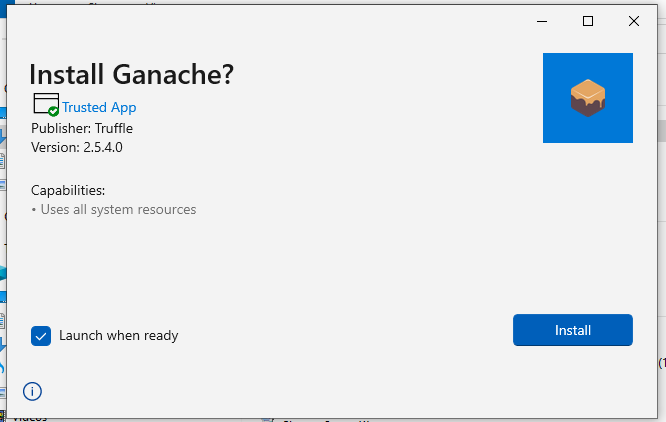
➢ Transactions Page -this list all the transactions that have taken place on the personal blockchain.

➢ Logs Page - this shows you all the server logs that you might need when debugging your

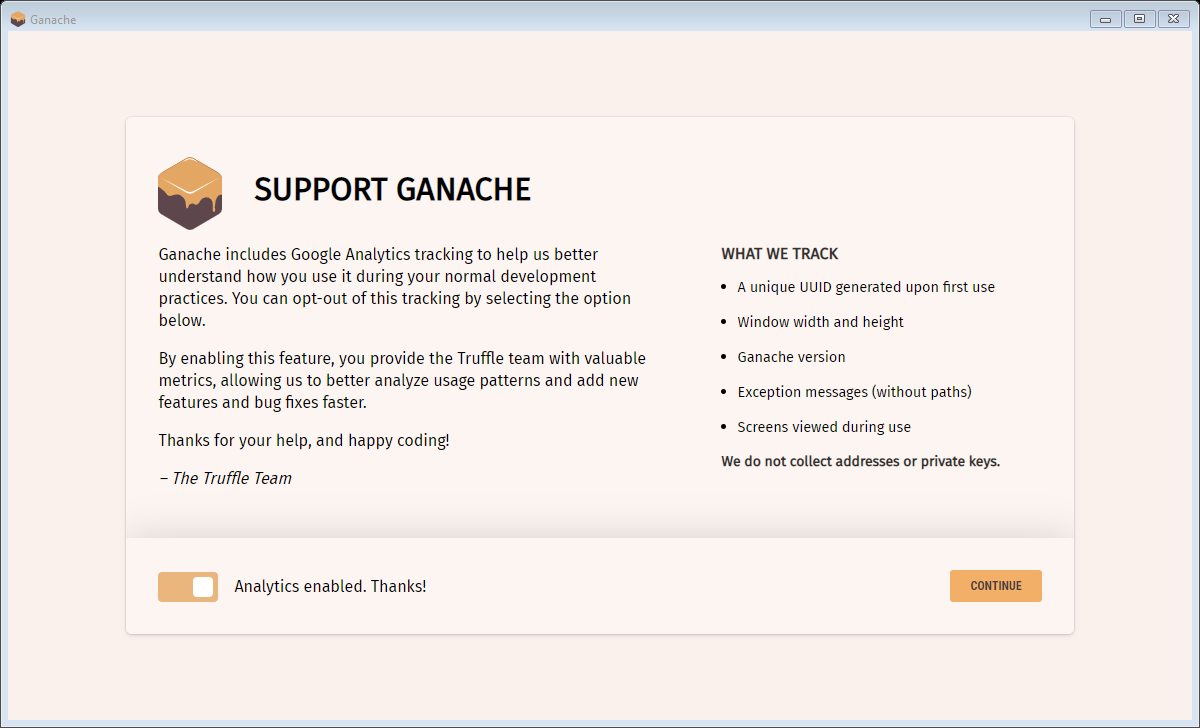
application.

**Ganache:** [**https://www.trufflesuite.com/ganache**](https://www.trufflesuite.com/ganache)

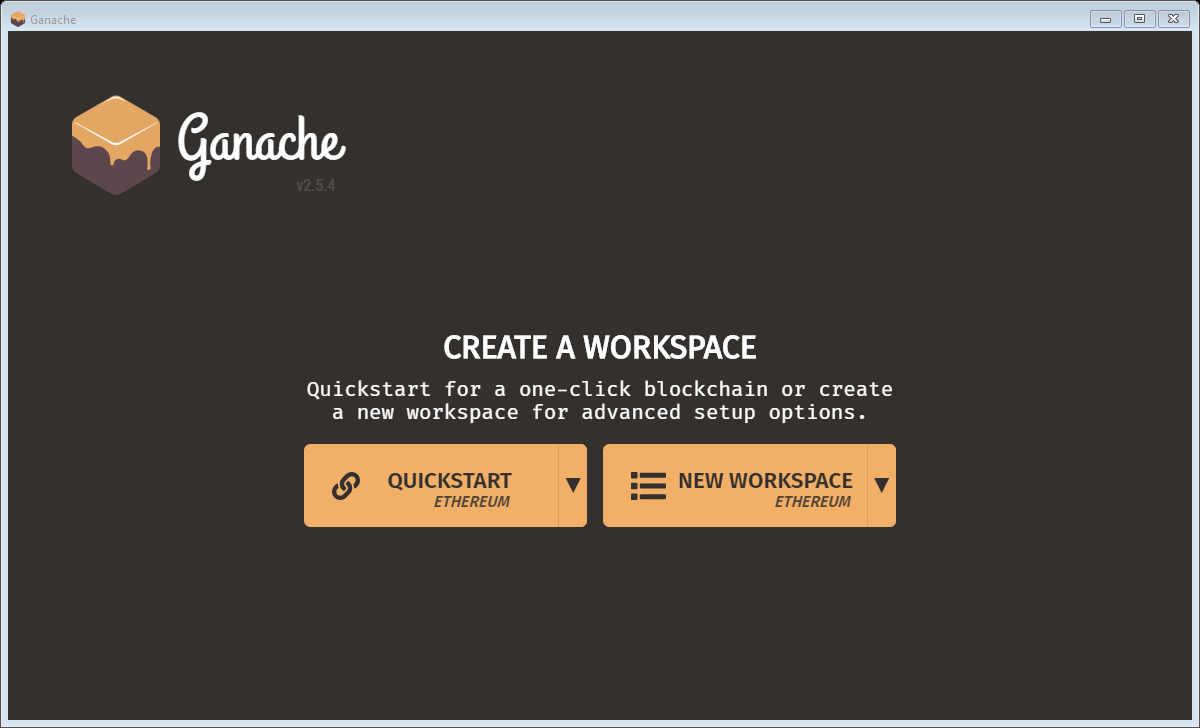


****

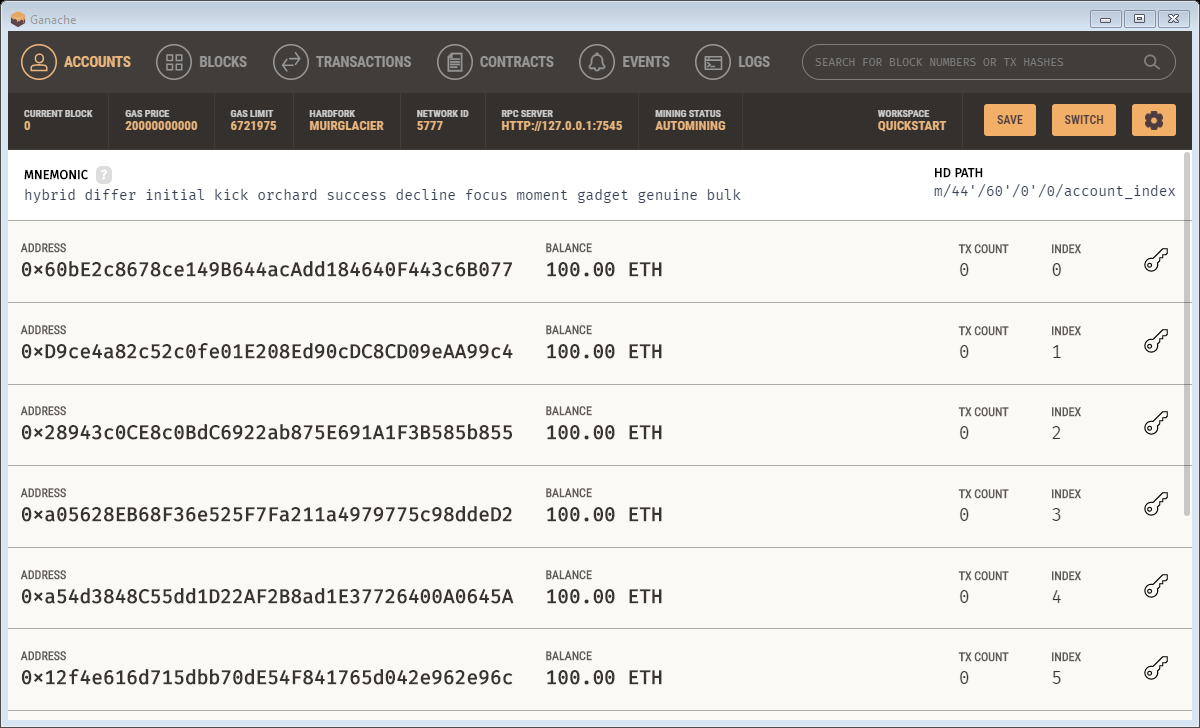
**Click on install**

****

**Click on Continue**

****

**Click on QuickStart**

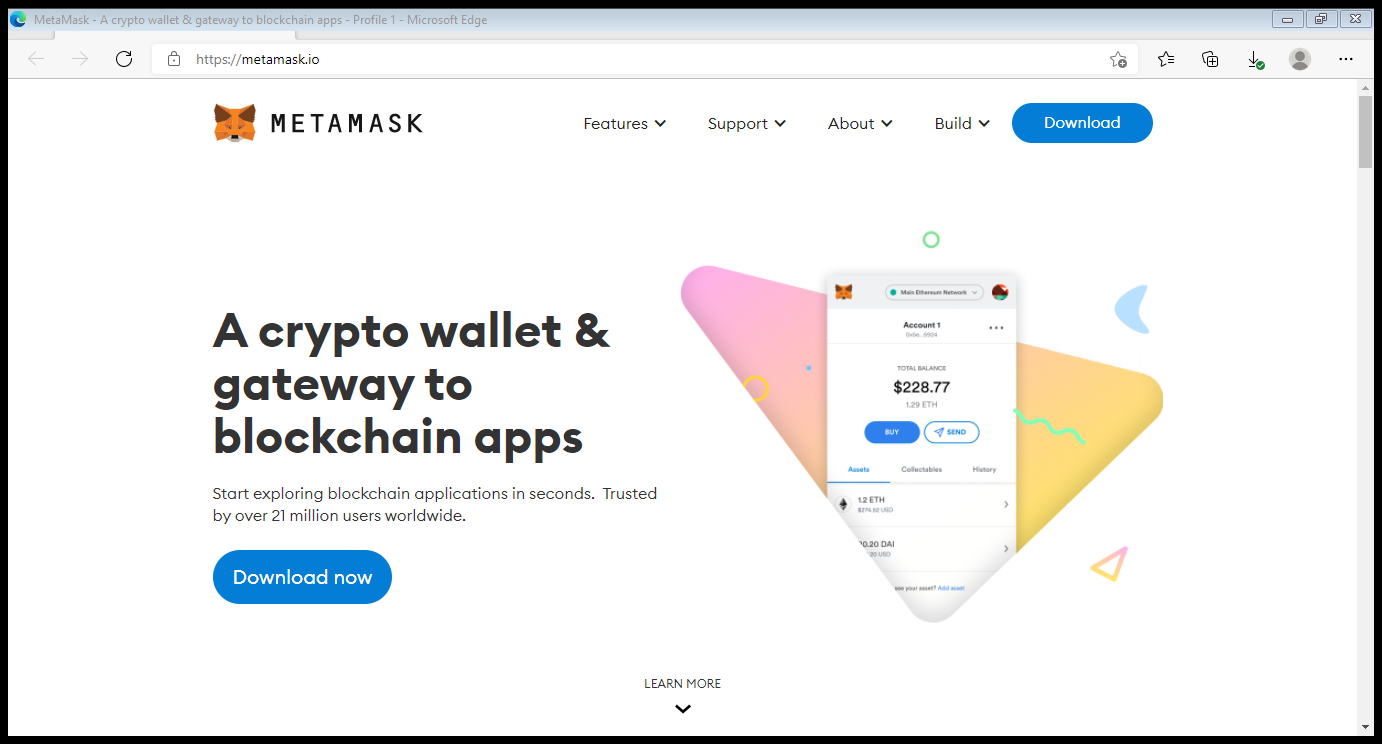
****

**Ganache Successfully installed. Now minimize Ganache.**

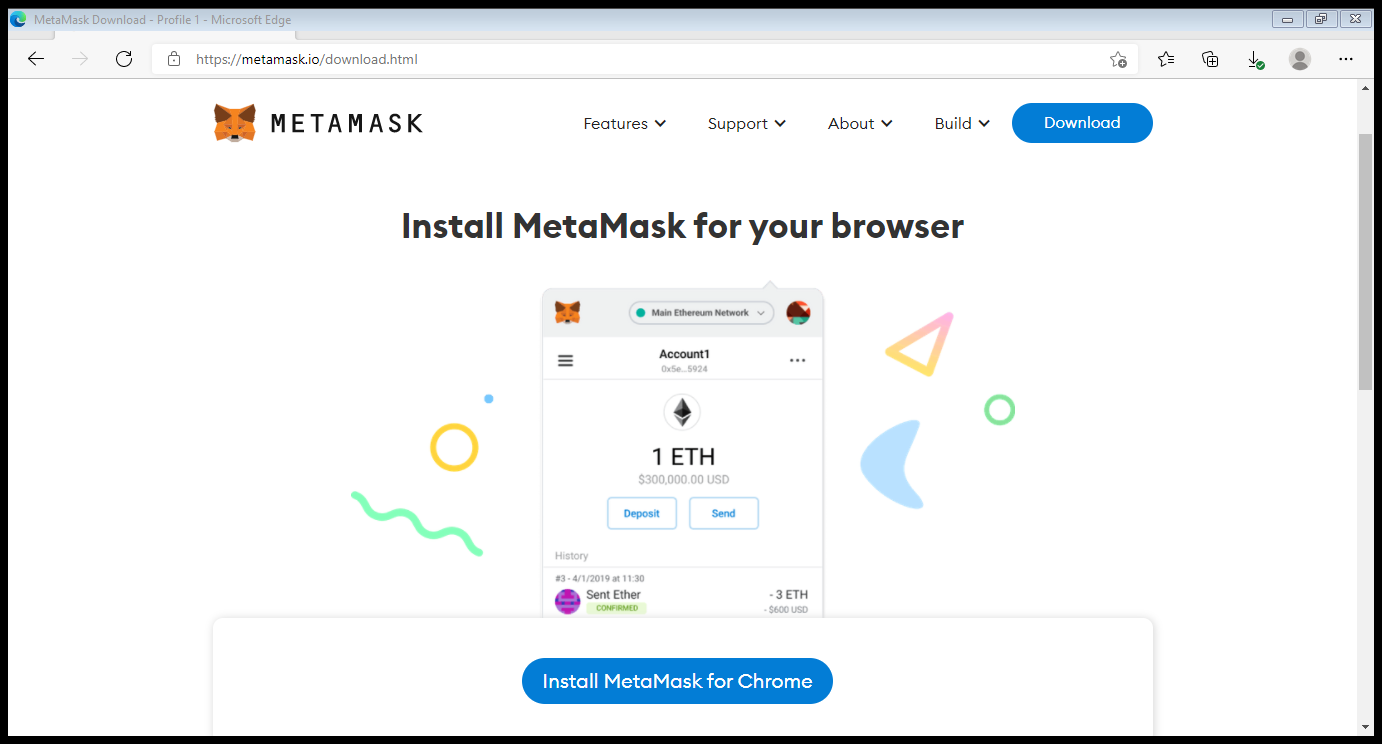
**STEP 2**

**SETUP A METAMASK ETHEREUM WALLET**

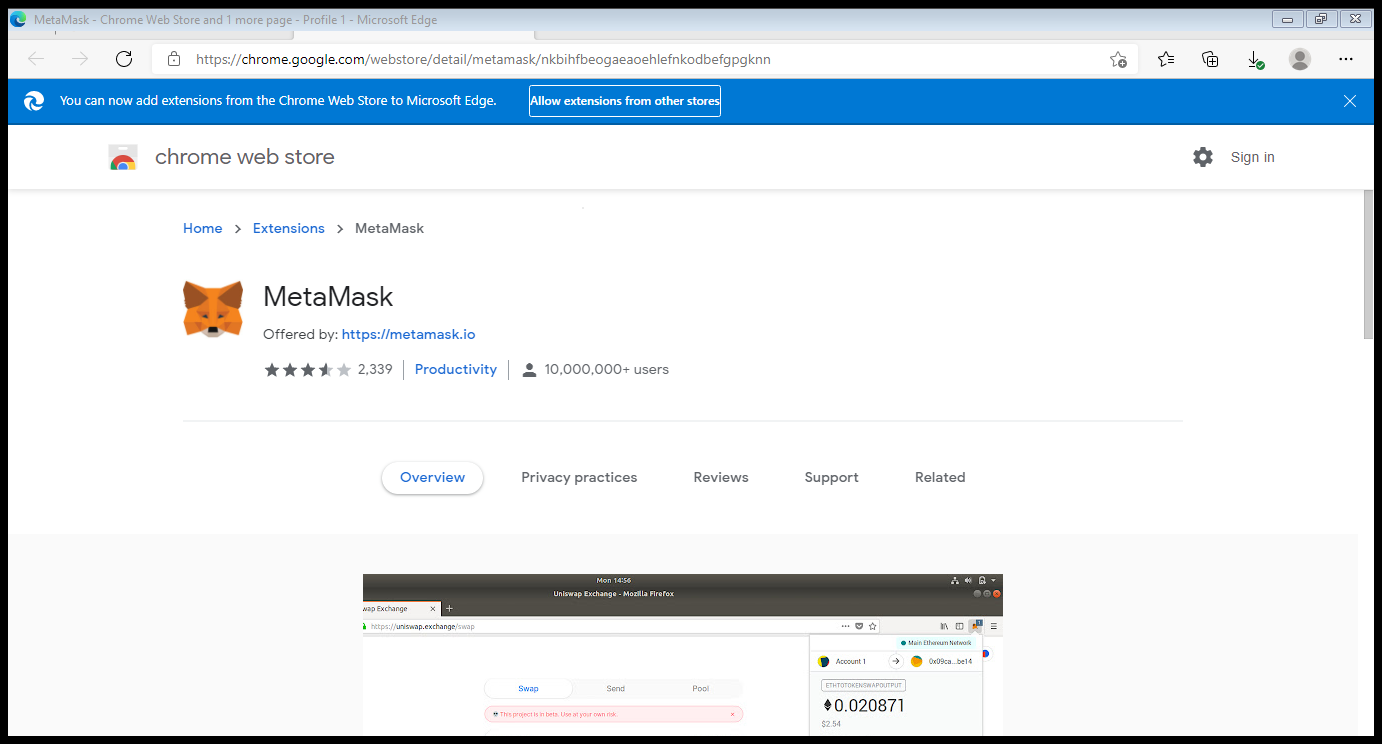
* MetaMask is just an Ethereum Browser and Ether wallet.
* It interacts with Ethereum Dapps and Smart Contracts without running a full Ethereum node.
* MetaMask add-on can be installed on Chrome, Firefox, Opera, and the new Brave browser.
* **URL:** [**https://metamask.io/**](https://metamask.io/)
* Install MetaMask
* Add MetaMask extension to the Browser
* MetaMask will show up 12 words recovery key (Seed).
* These 12 words are the only way to restore MetaMask accounts.



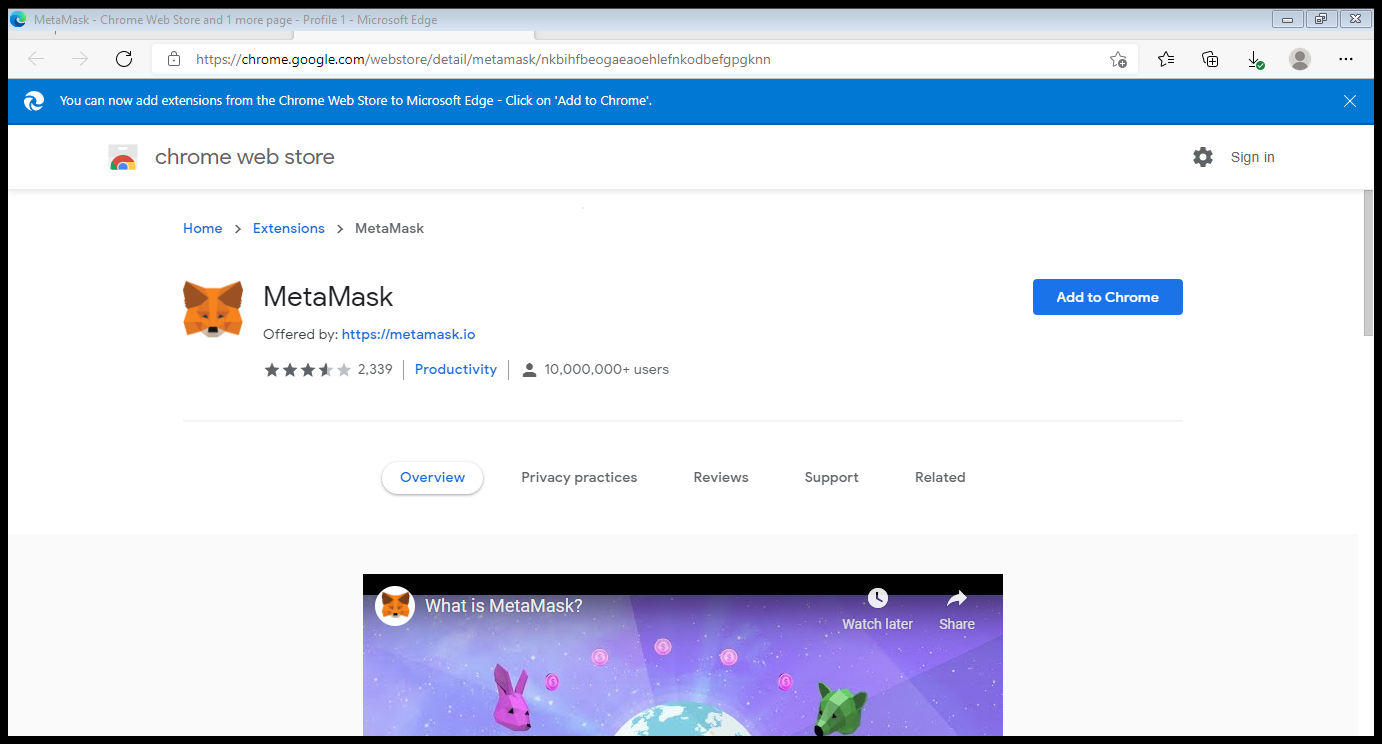
Click on download



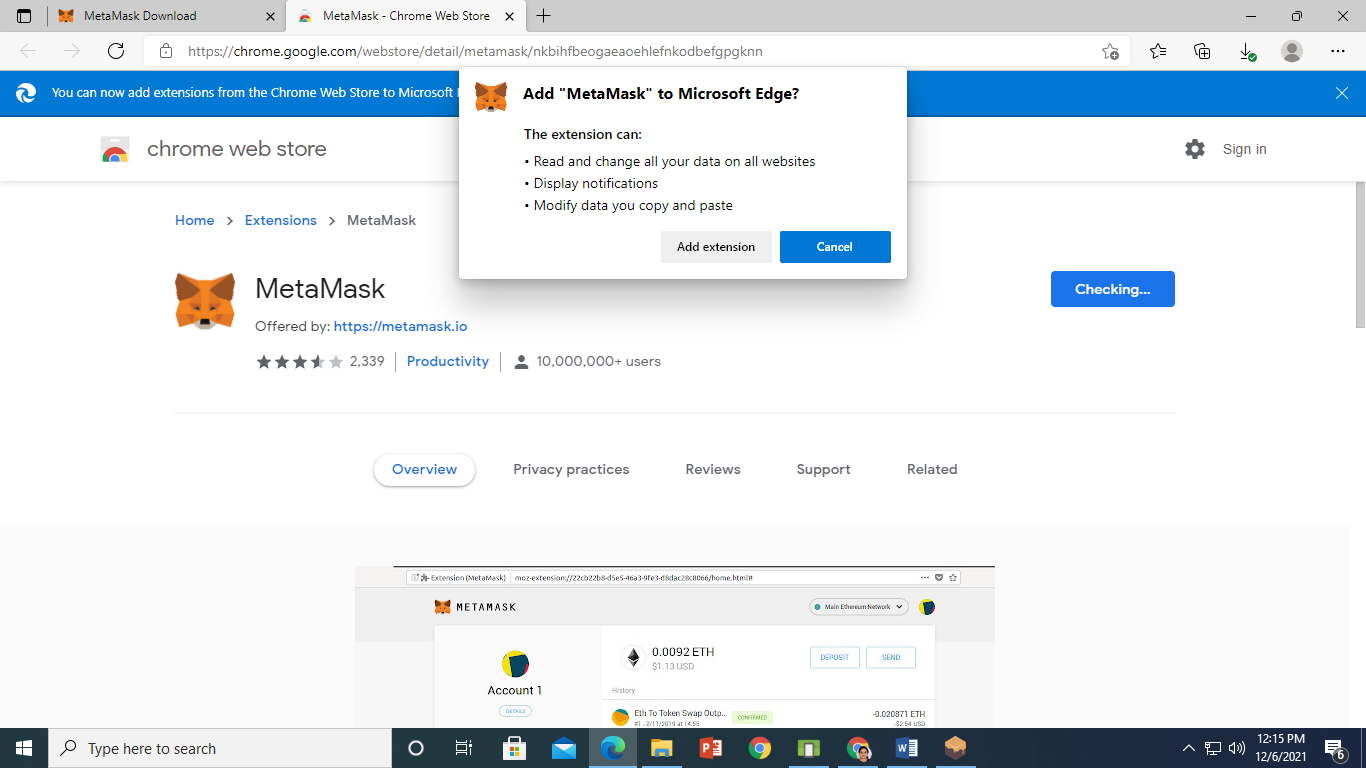
Click on Install MetaMask for Chrome.



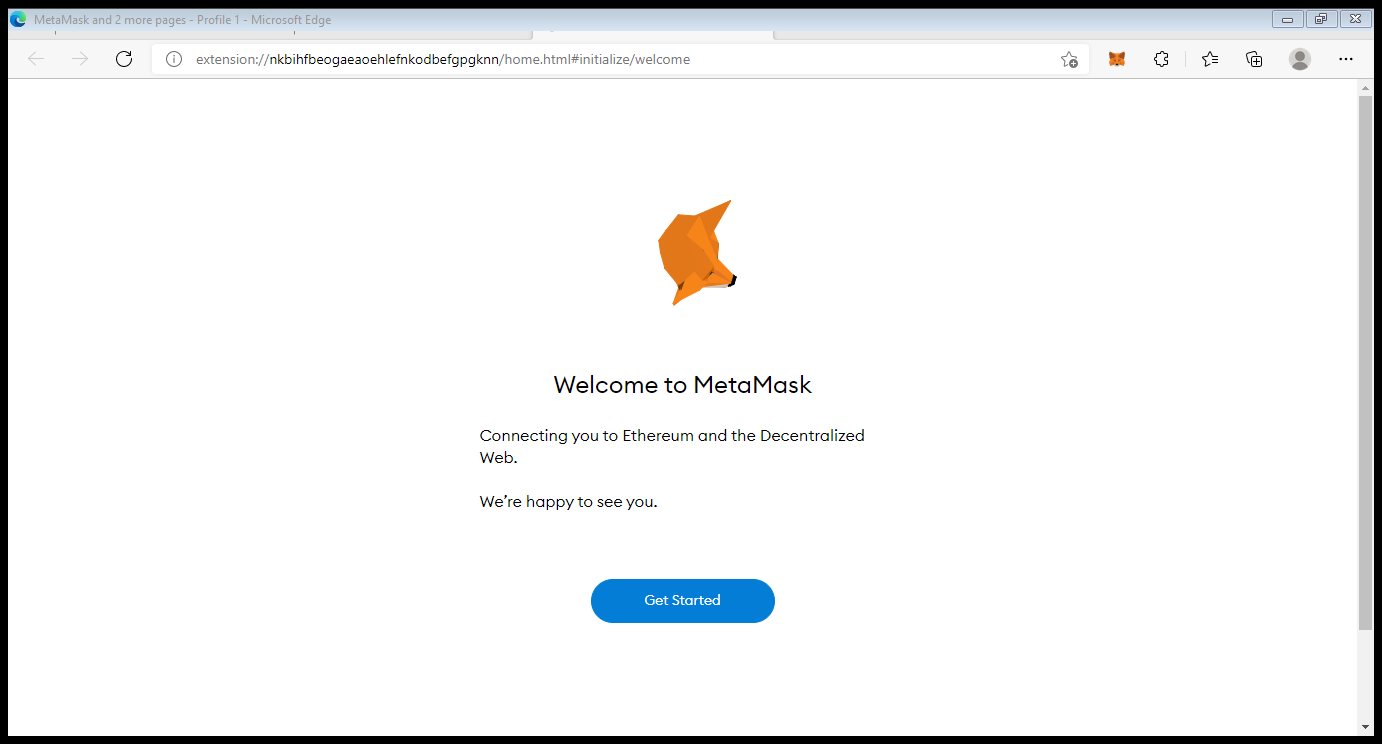
Click on Allow Extensions from other stores.



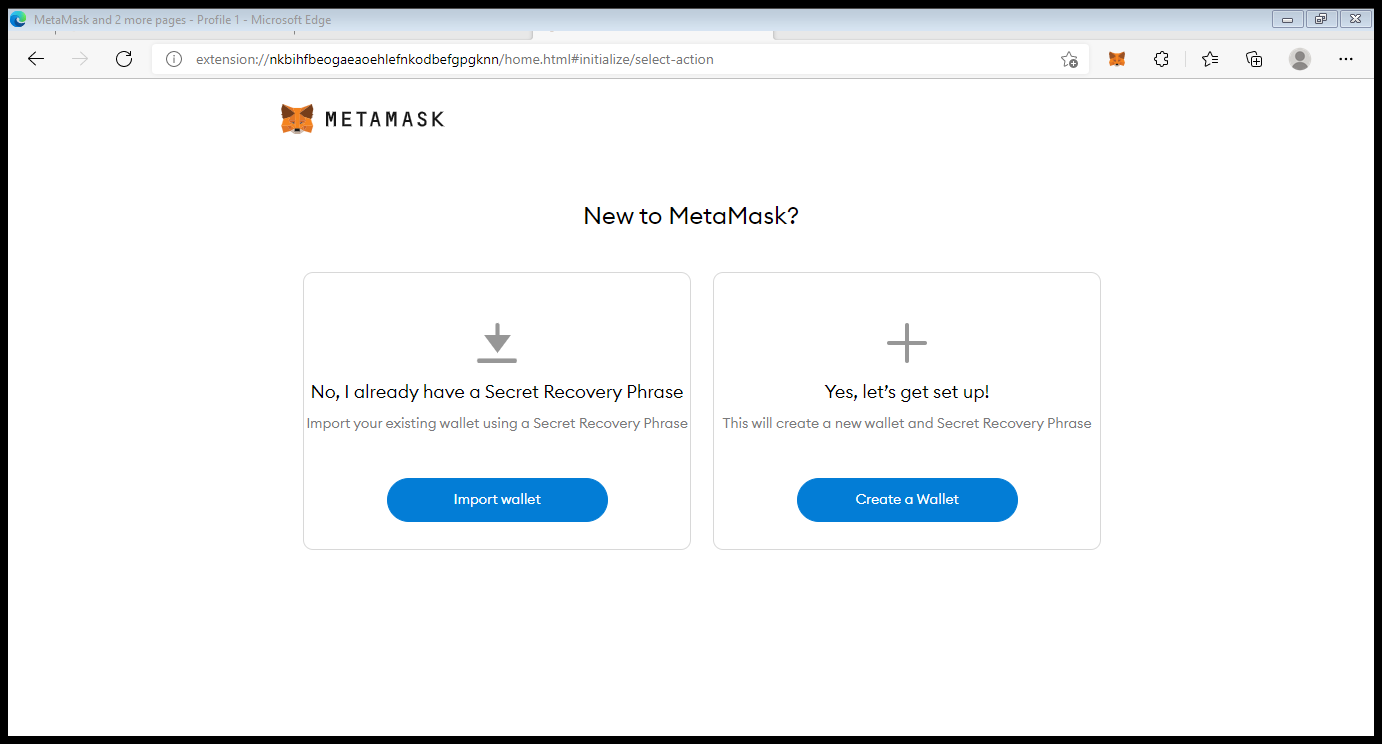
Click on Add to Chrome



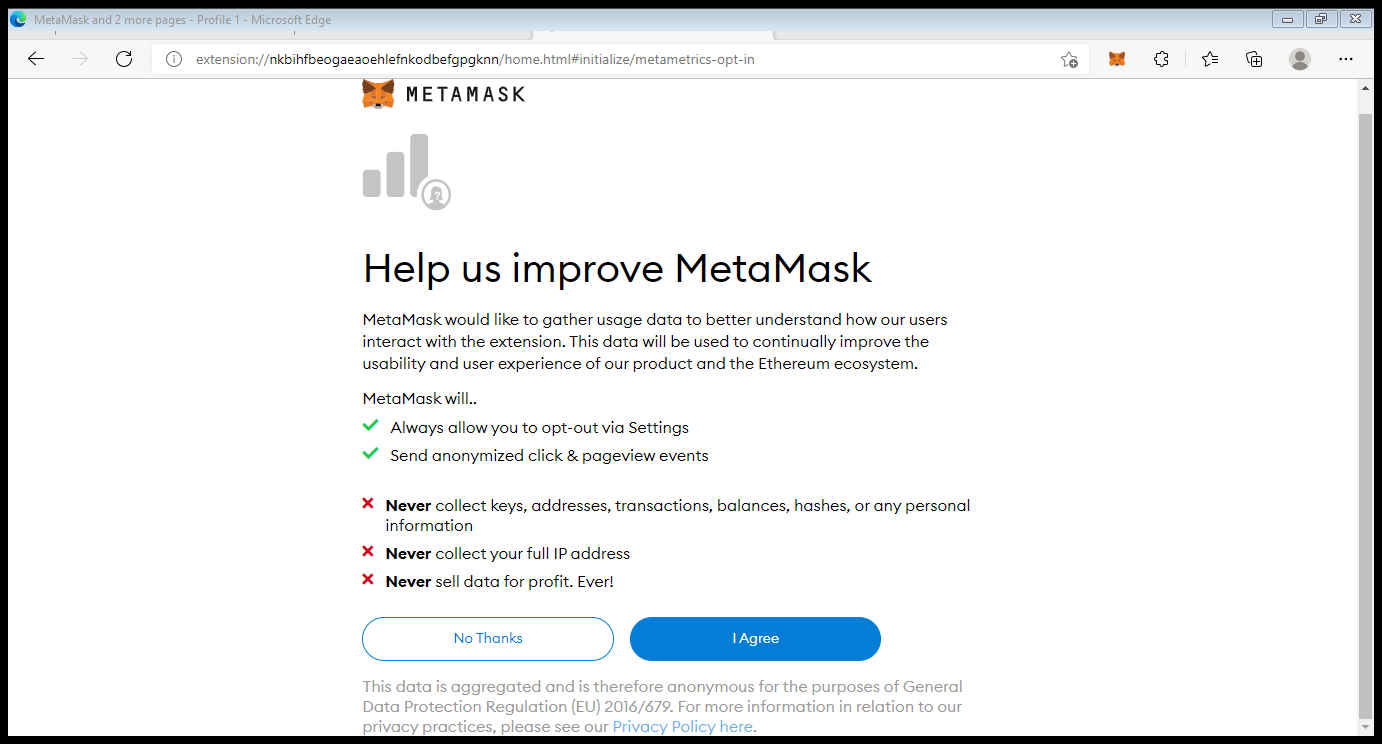
Click on Add extension



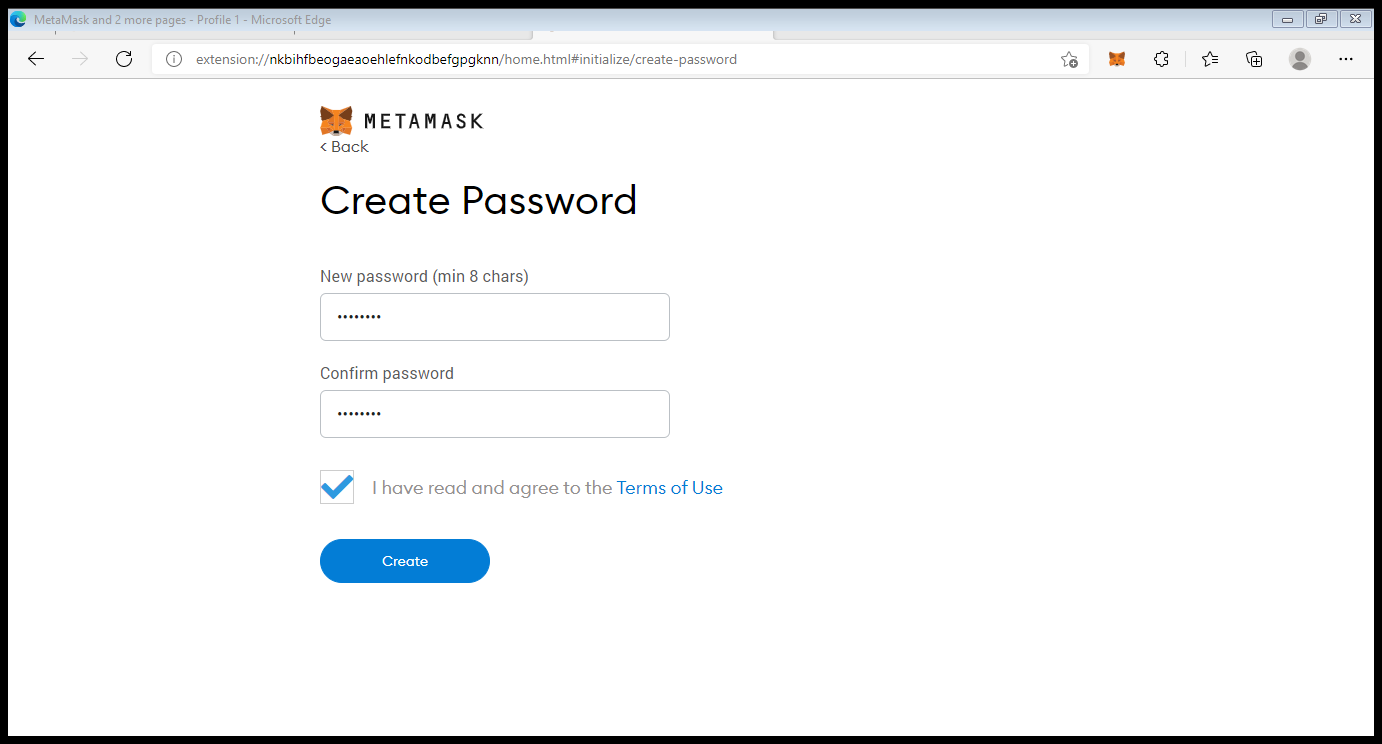
Click on Get Started



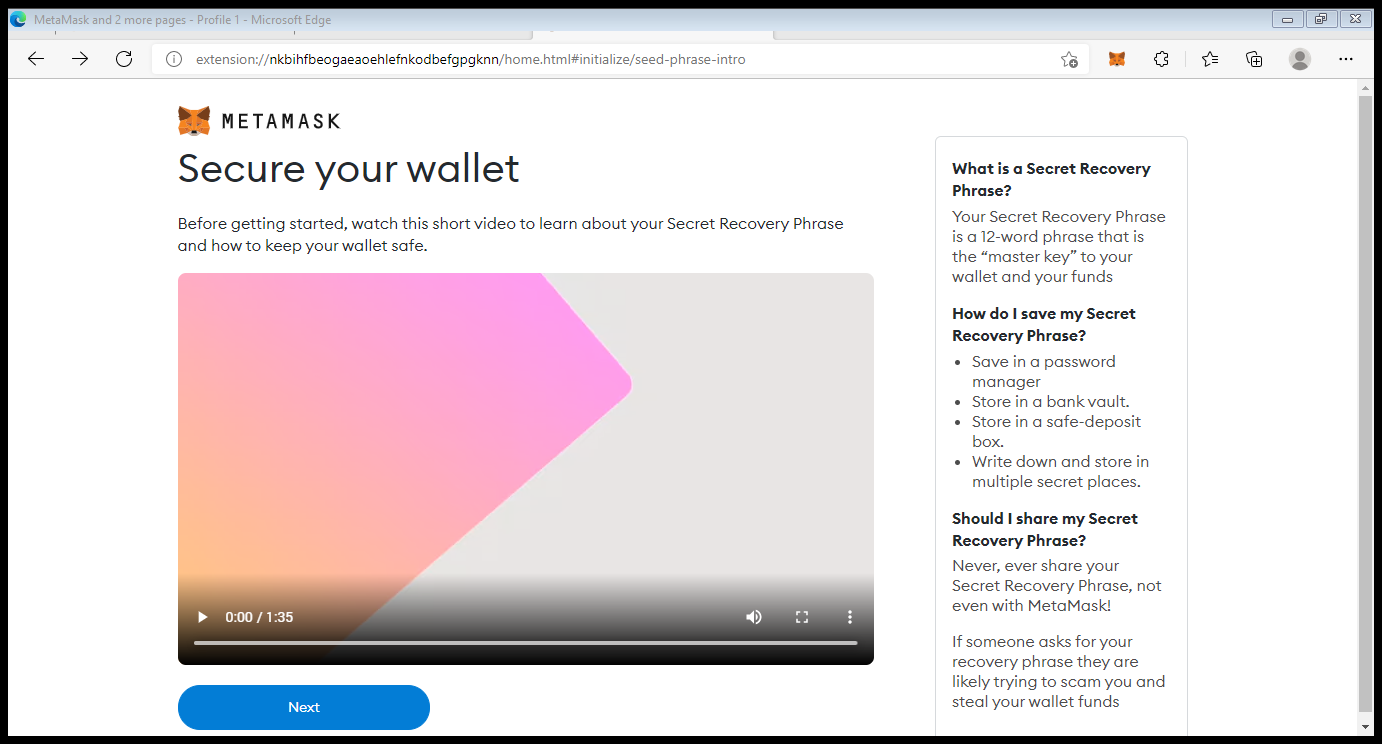
Click on Create Wallet



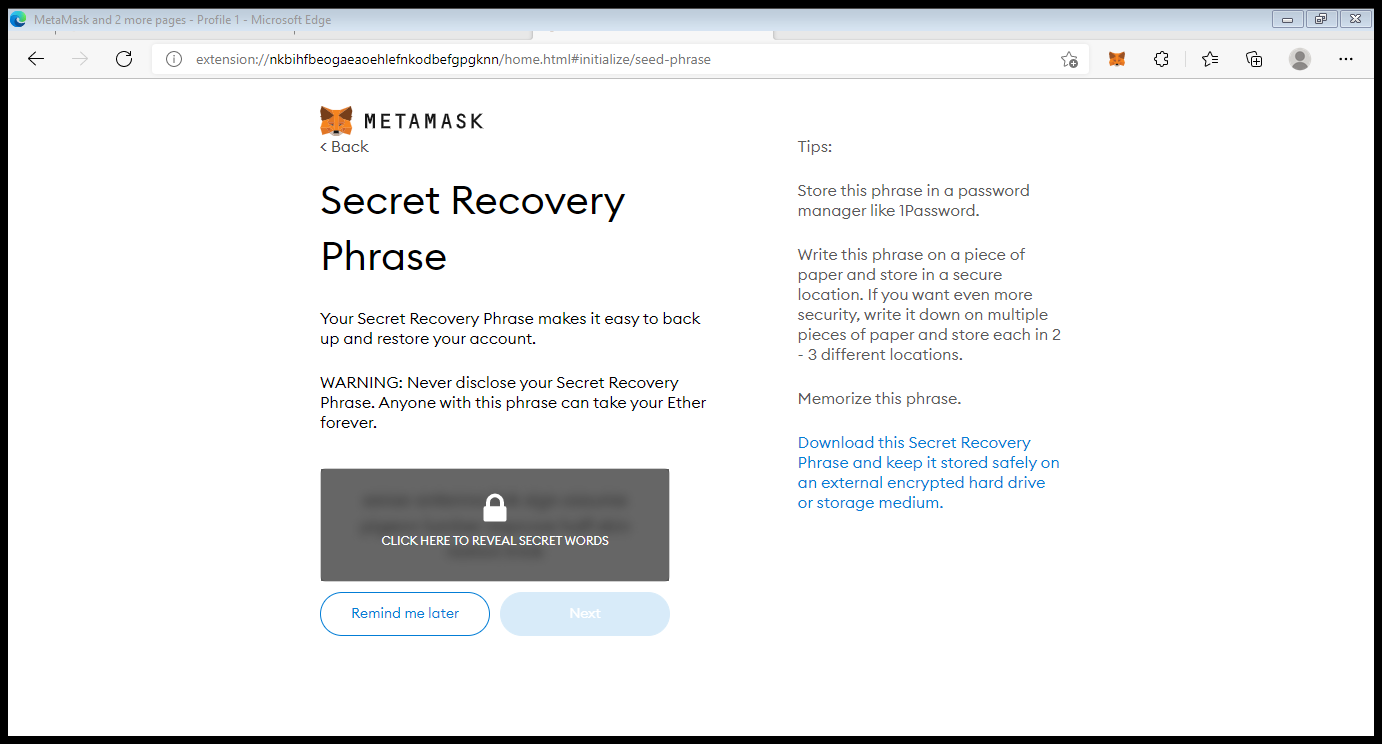
Click on I Agree



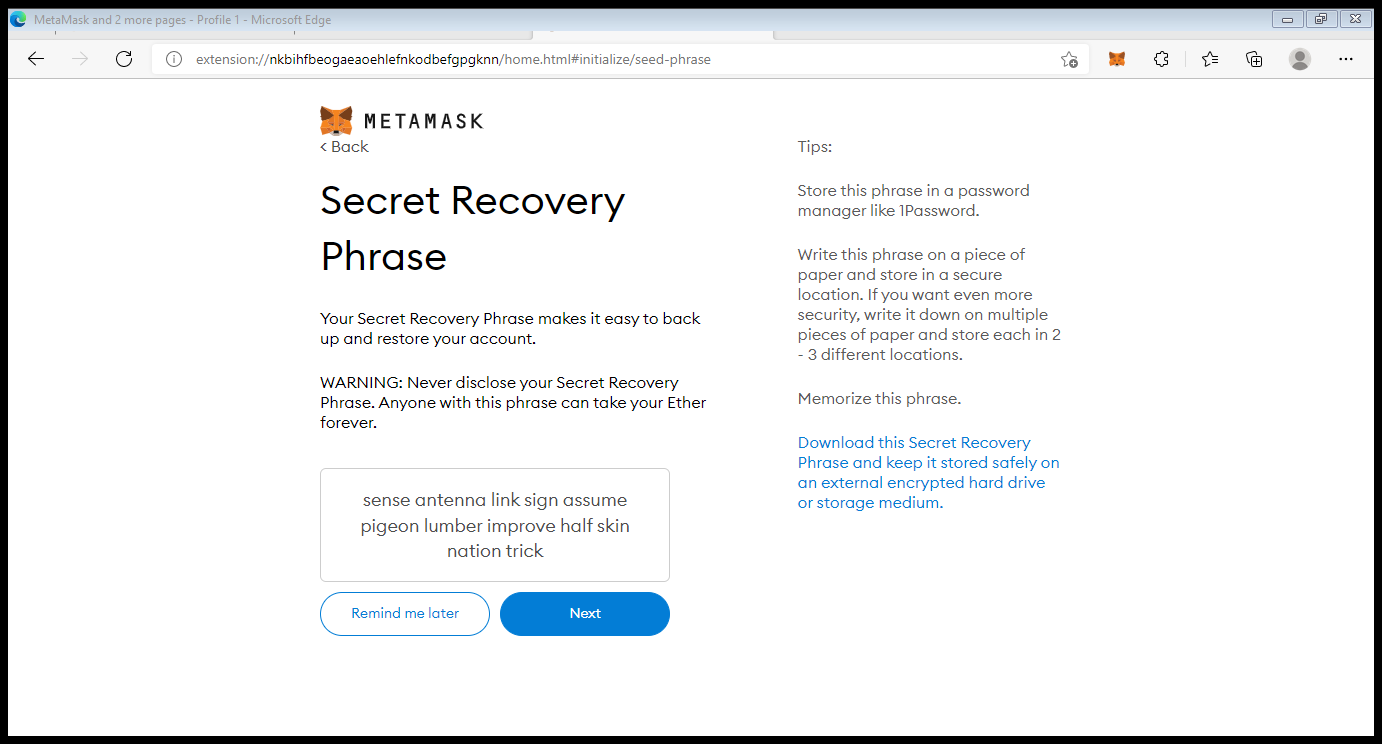
Mention password. Note down password further it is required and click on create.



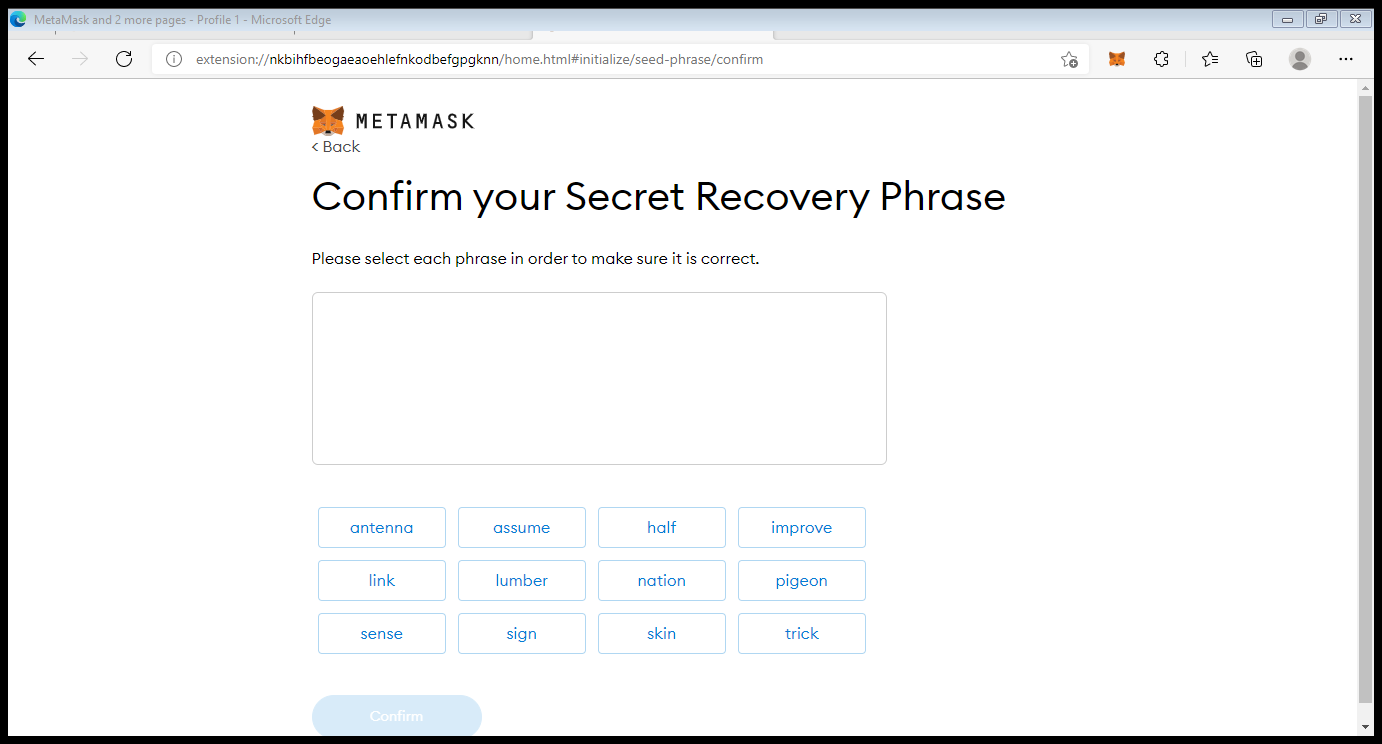
Click on Next.



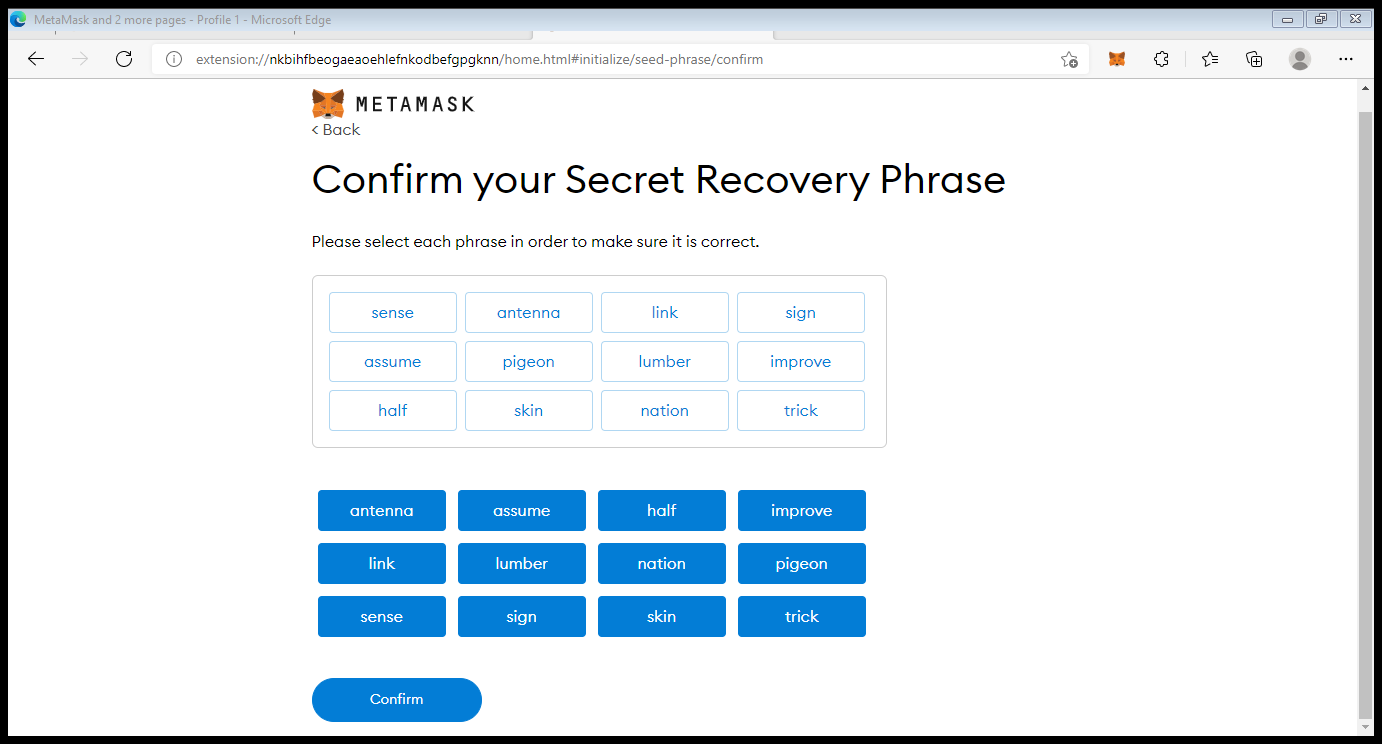
Click on Click here to reveal secret words.



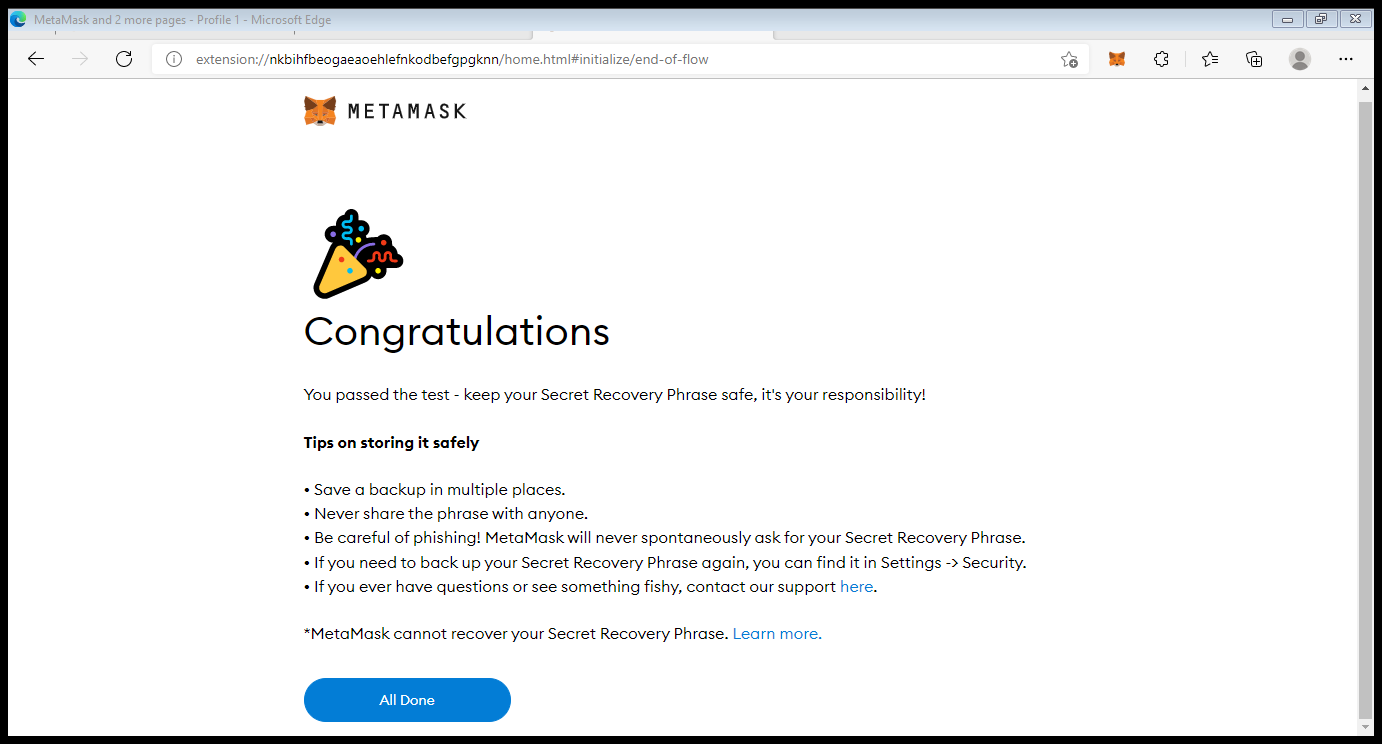
Copy all the words and save it in notepad. Then click on Next.



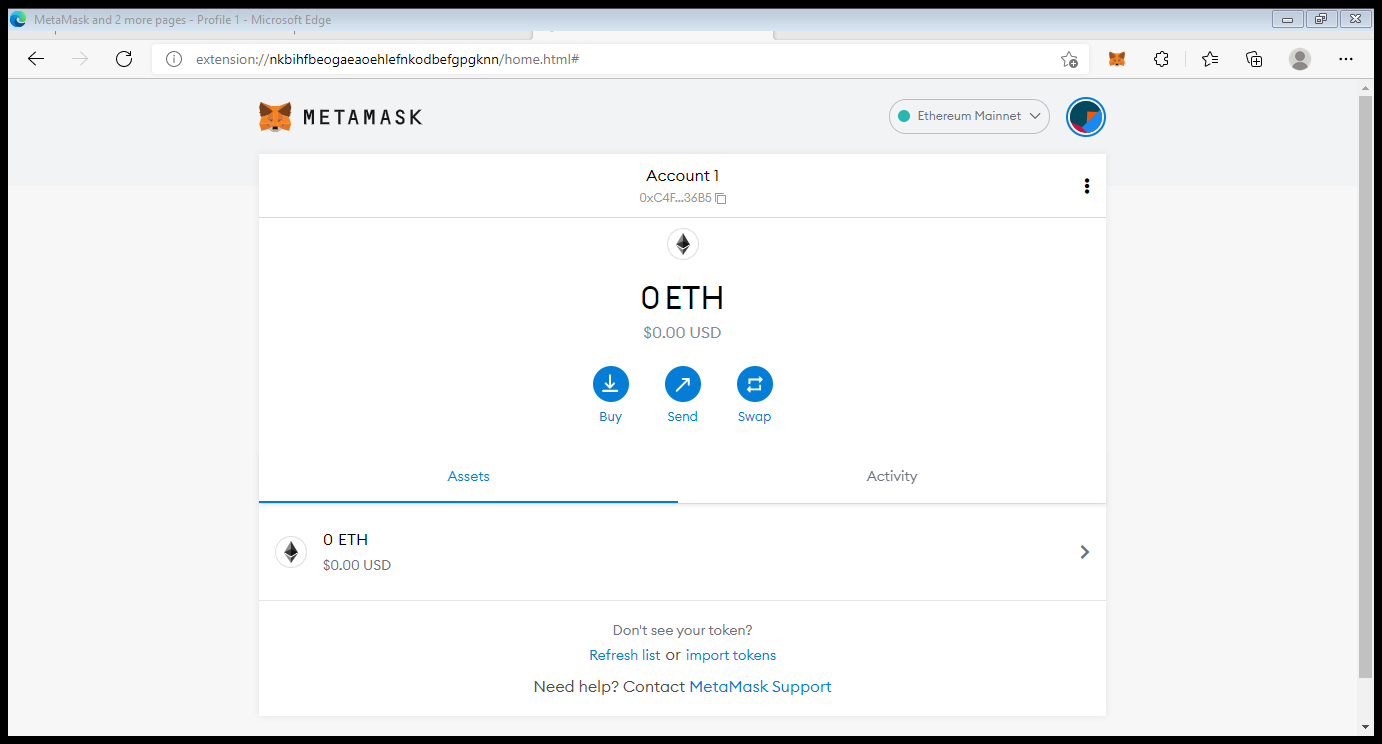
Refer the notepad and select the words in same sequence.

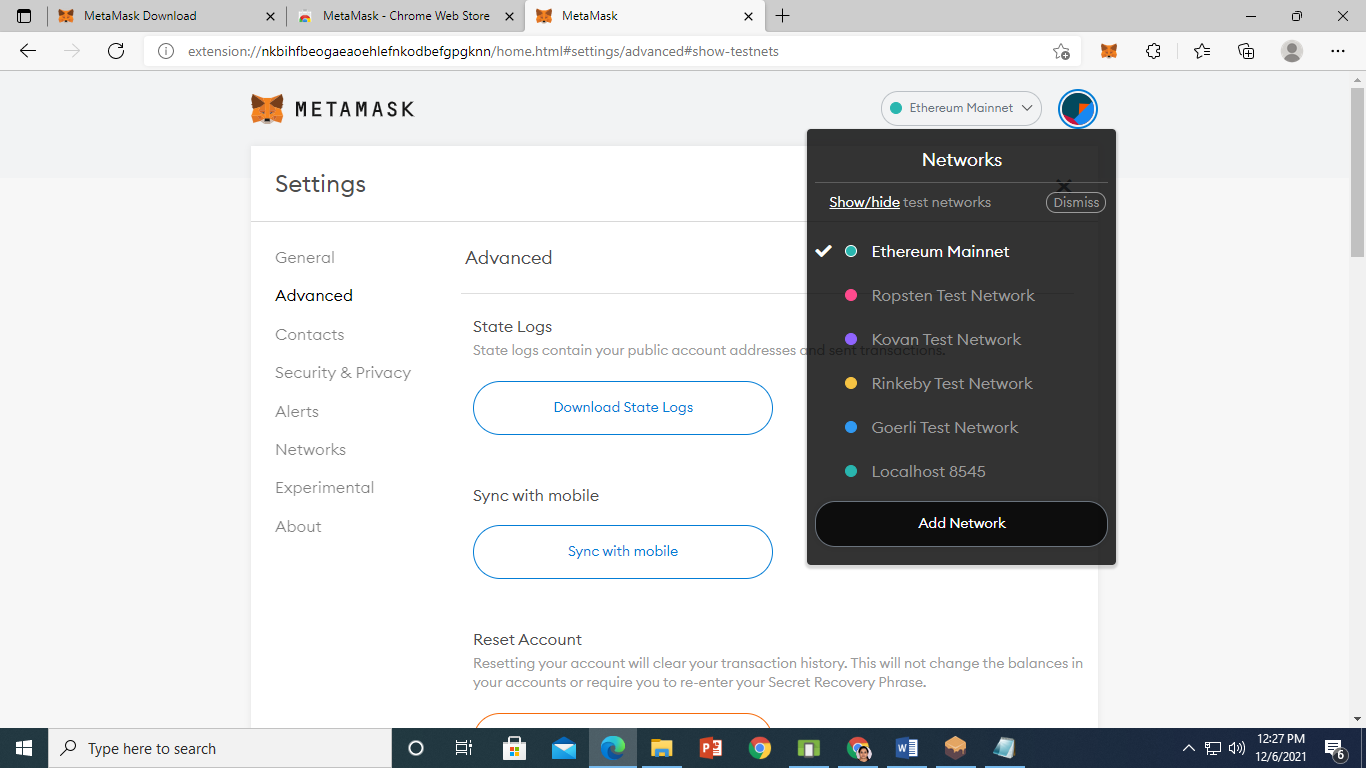


Click on confirm.

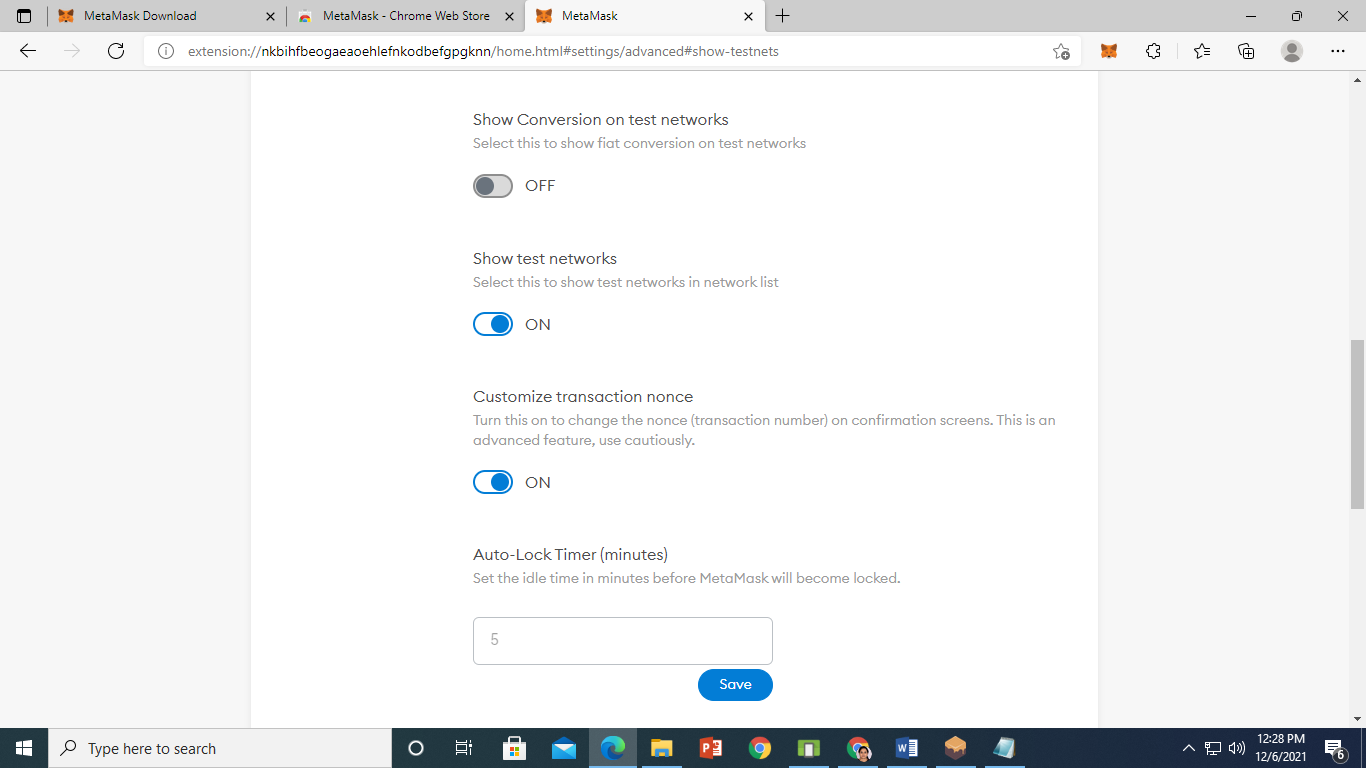


Click on All Done.

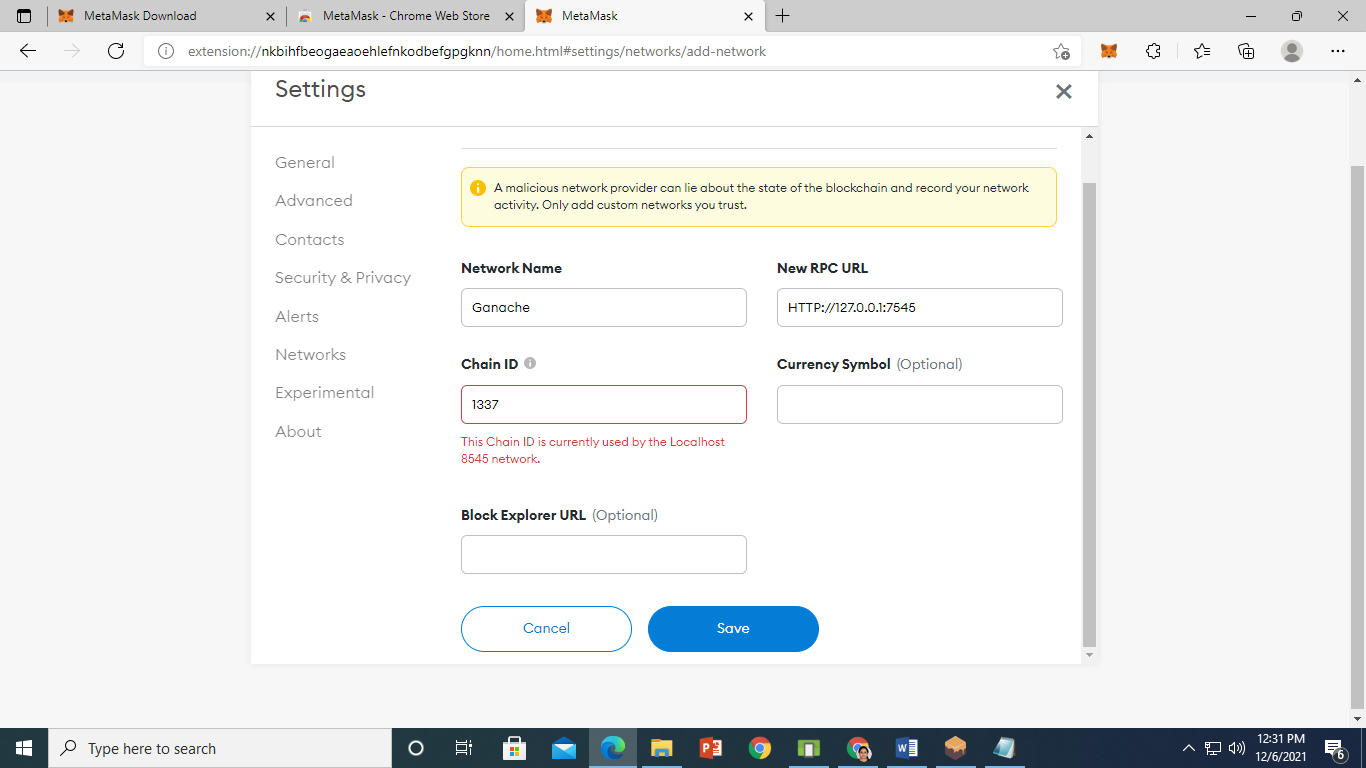




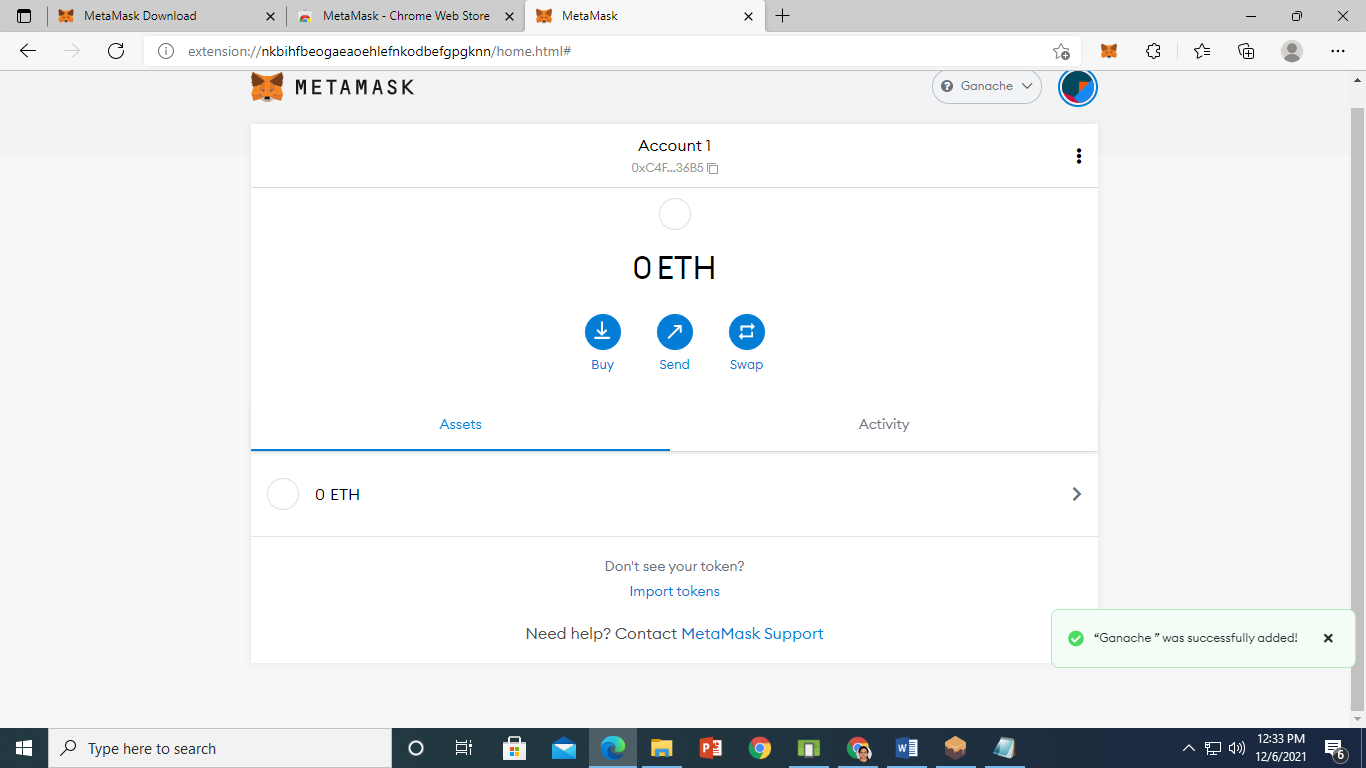
Click on Show/Hide test network.



On show test network. scroll up and select dropdown list Ethereum Mainnet and click on Add Network



Click on save.



Ganache was successfully added.

**STEP 3**

**On remix IDE Create Election.sol Contract.**

pragma solidity ^0.4.2;

contract Election {

 // Model a Candidate

 struct Candidate {

 uint id;

 string name;

 uint voteCount;

 }

 // Store accounts that have voted

 mapping(address => bool) public voters;

 // Store Candidates

 // Fetch Candidate

 mapping(uint => Candidate) public candidates;

 // Store Candidates Count

 uint public candidatesCount;

 // voted event

 event votedEvent (

 uint indexed \_candidateId

 );

 function Election () public {

 addCandidate("N MODI, BJP");

 addCandidate("A kejriwal, AAP");

 addCandidate("Rahul G, Congress");

 addCandidate("Nikhil, JDS");

 }

 function addCandidate (string \_name) private {

 candidatesCount ++;

 candidates[candidatesCount] = Candidate(candidatesCount, \_name, 0);

 }

 function vote (uint \_candidateId) public {

 // require that they haven't voted before

 require(!voters[msg.sender]);

 // require a valid candidate

 require(\_candidateId > 0 && \_candidateId <= candidatesCount);

 // record that voter has voted

 voters[msg.sender] = true;

 // update candidate vote Count

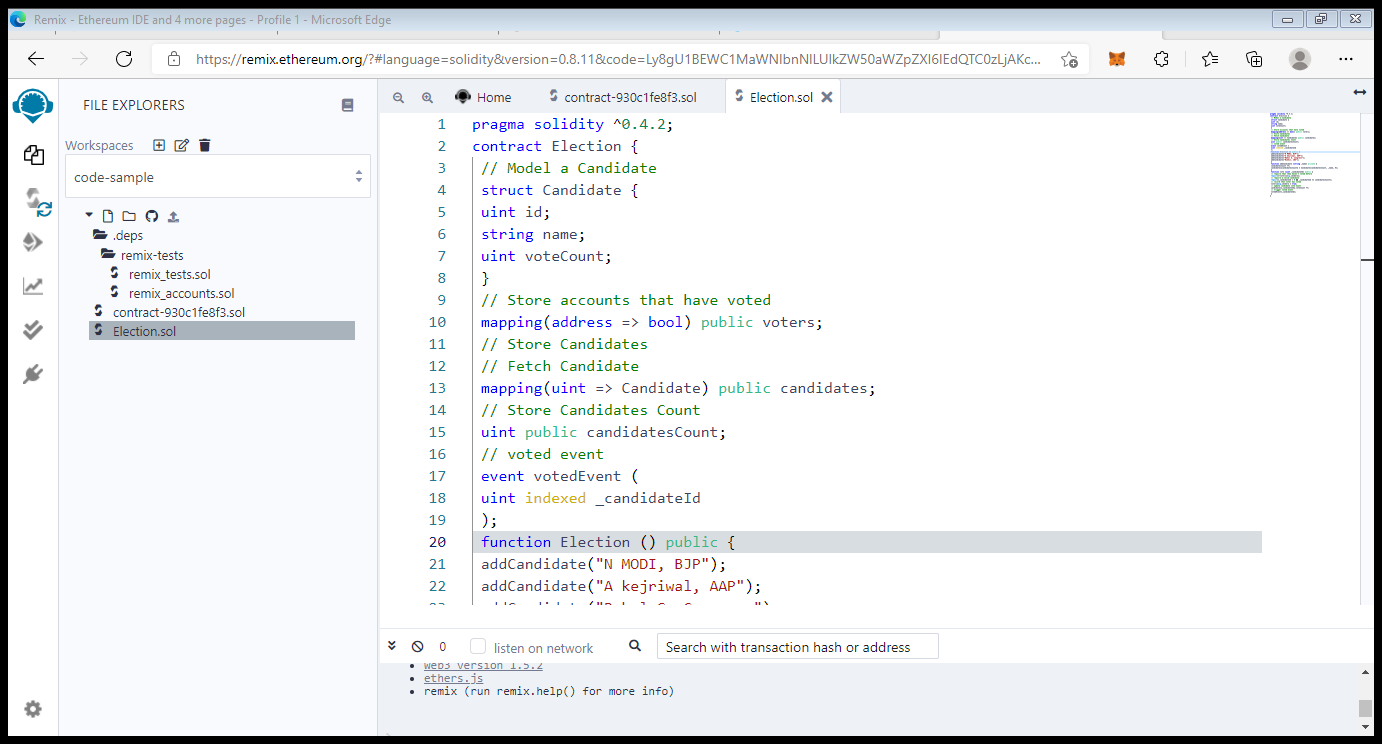
 candidates[\_candidateId].voteCount ++;

 // trigger voted event

 votedEvent(\_candidateId);

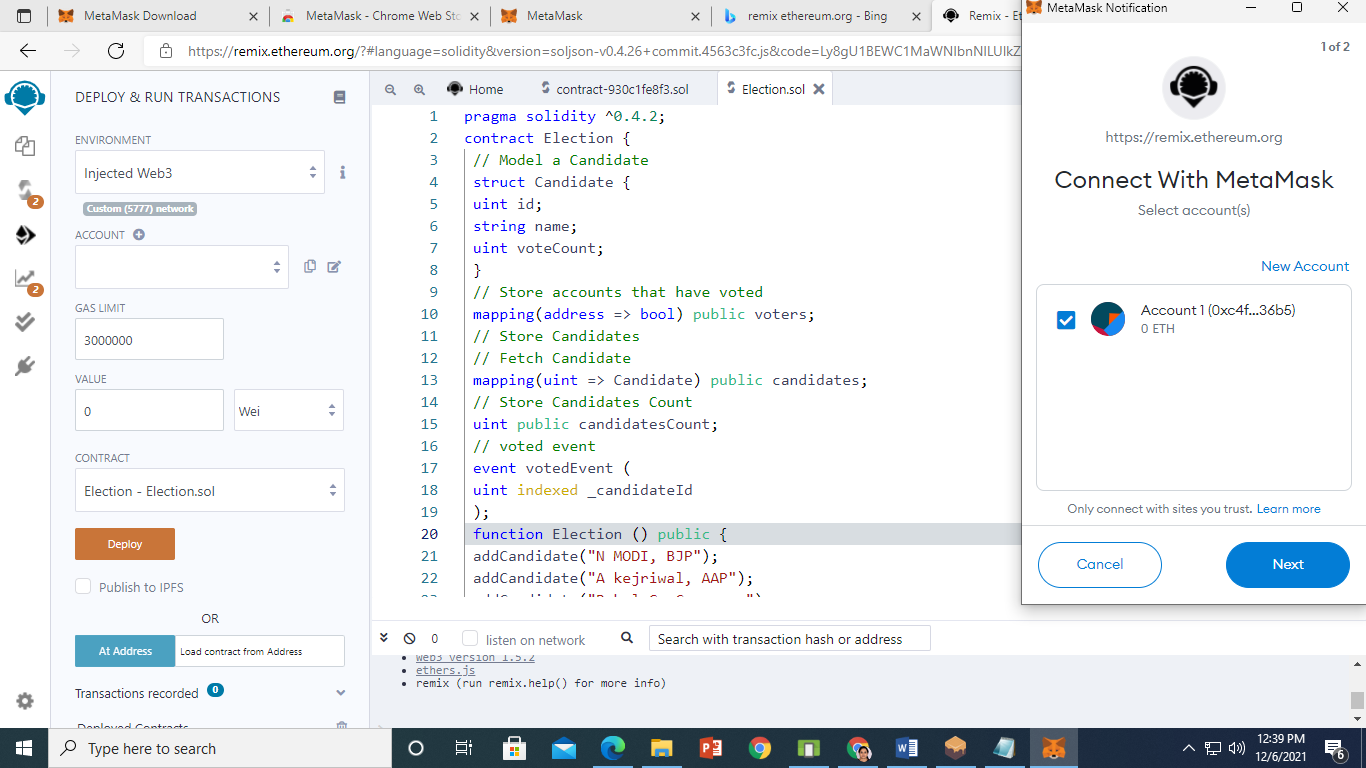
 }

}

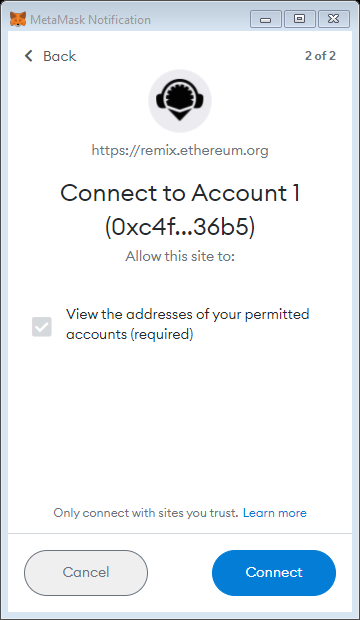


Compile the contract.

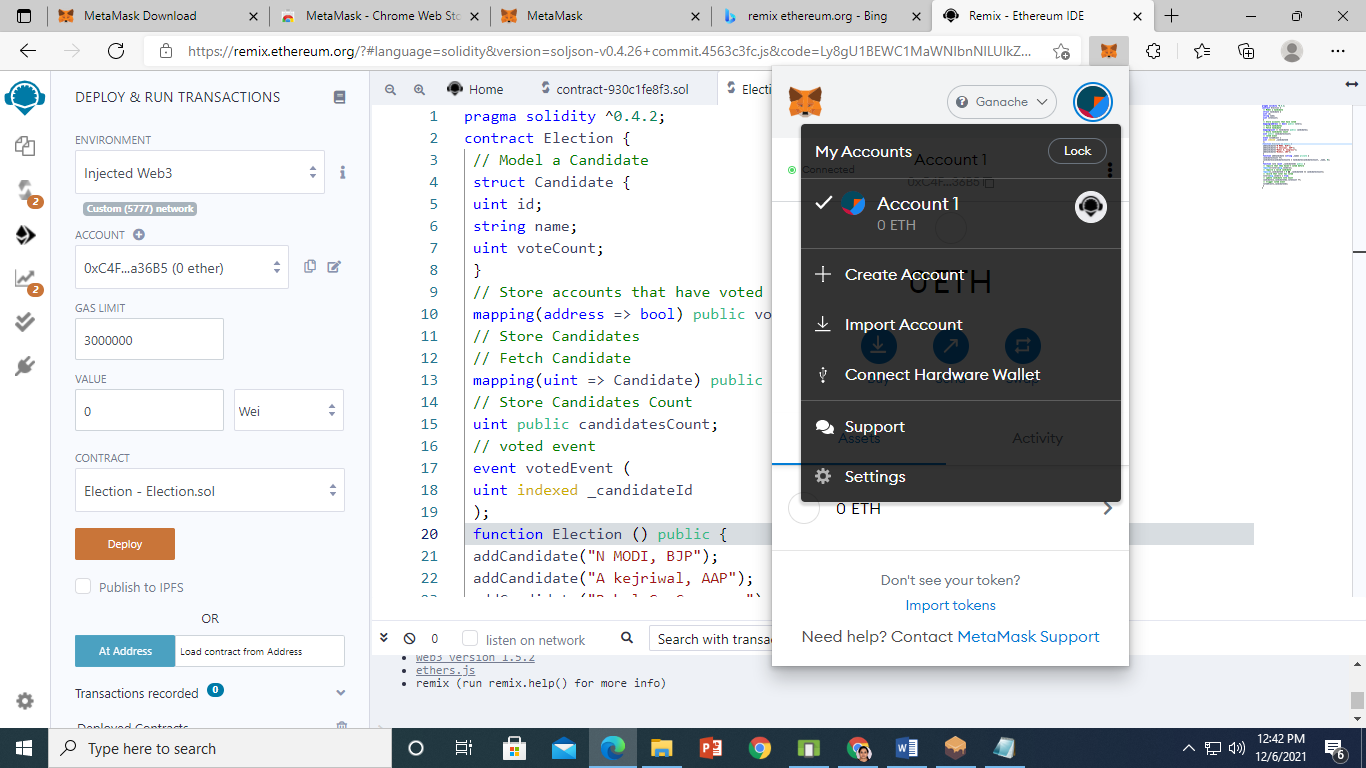
Now we will deploy the contract for that click on deploy. Then change environment and select Injected Web3. You will get following metamask window



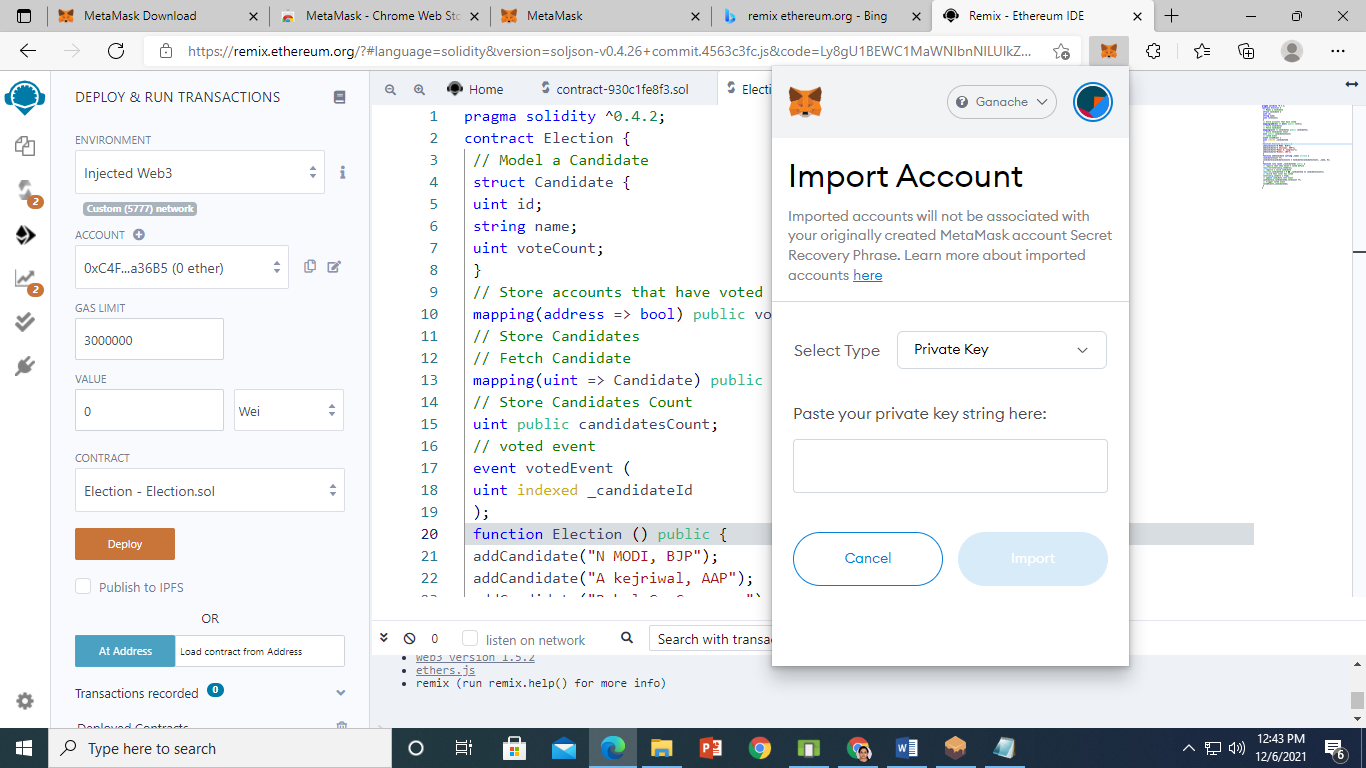
Click on next



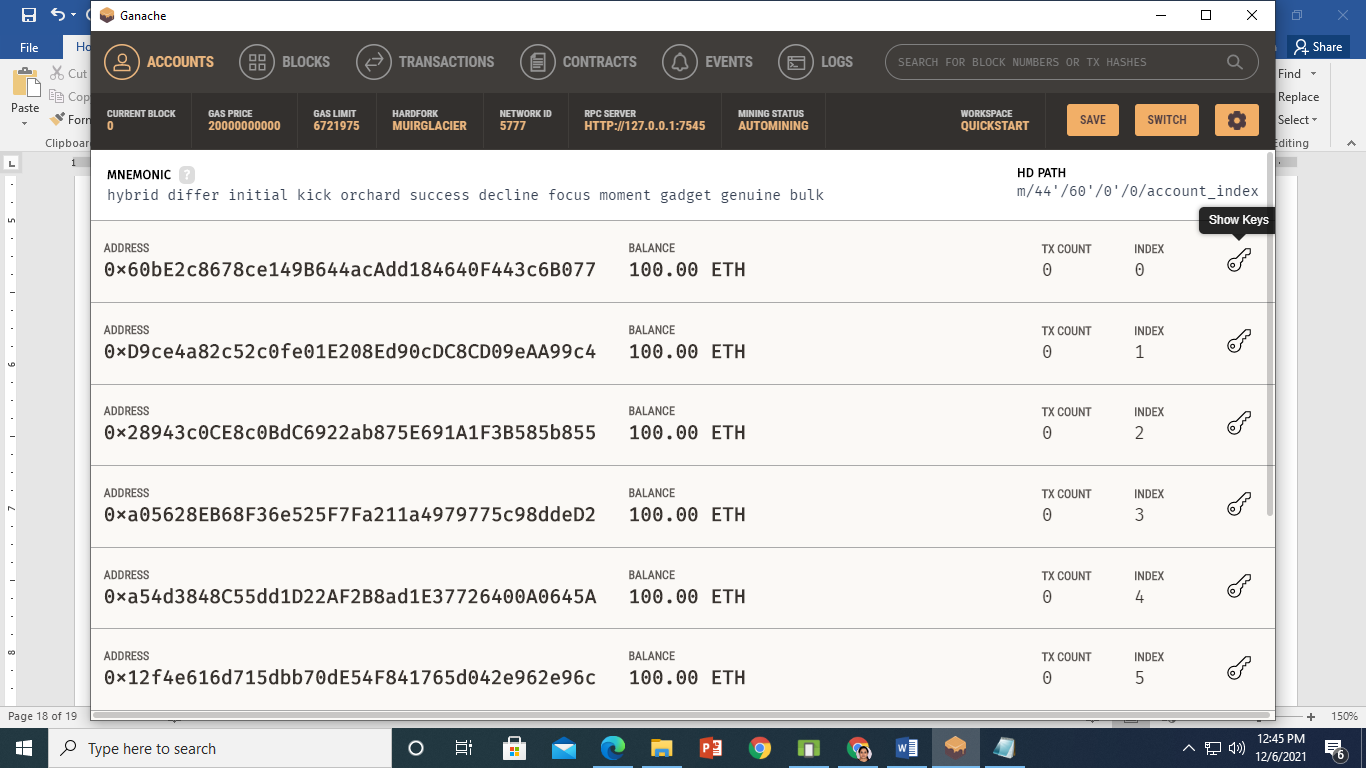
Click on connect

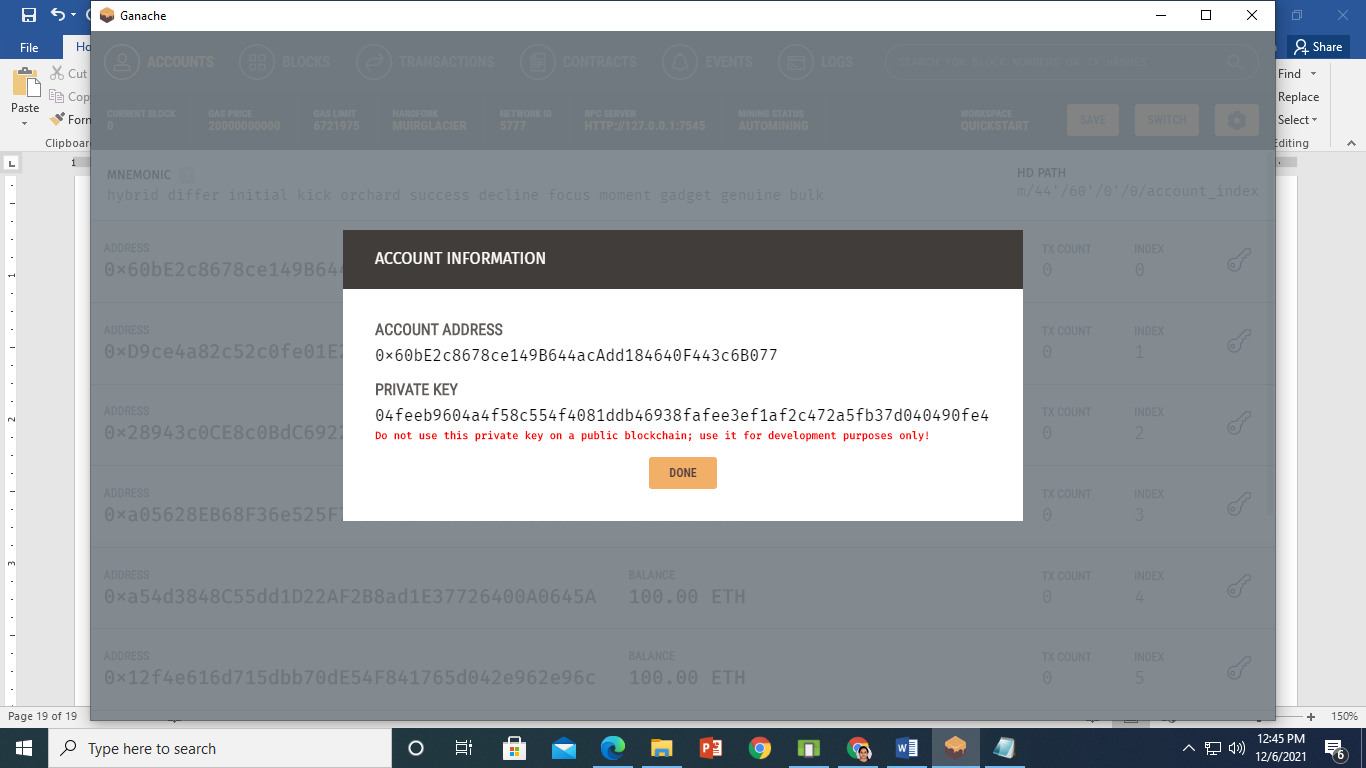


Click on Import Account (to import Ganache Account)

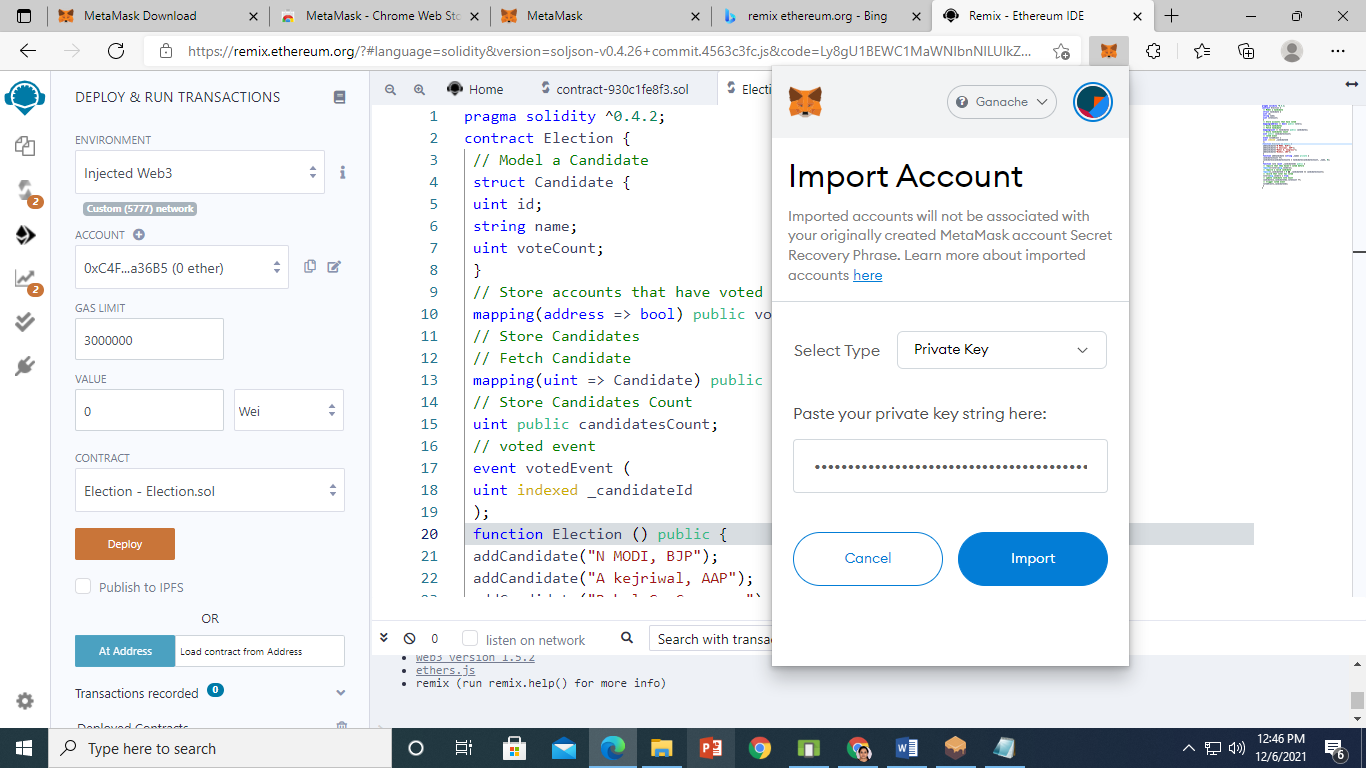


Now Maximize Ganache and click on key symbol of first account



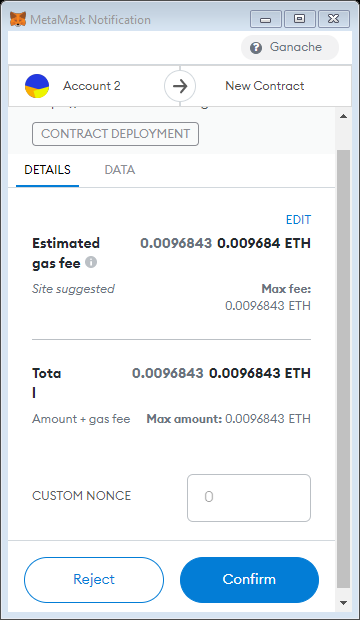


Copy Private Key n Paste it in metamask.

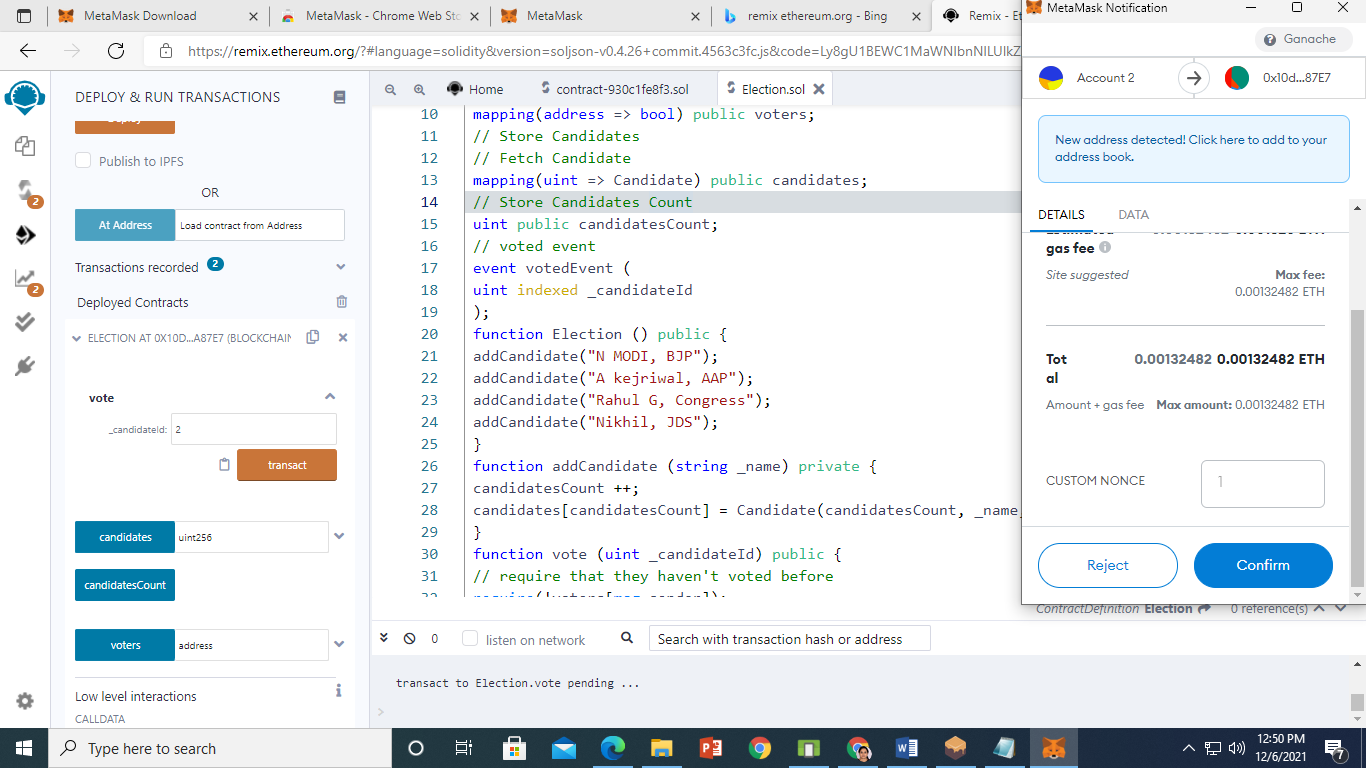


Now Click on Import.

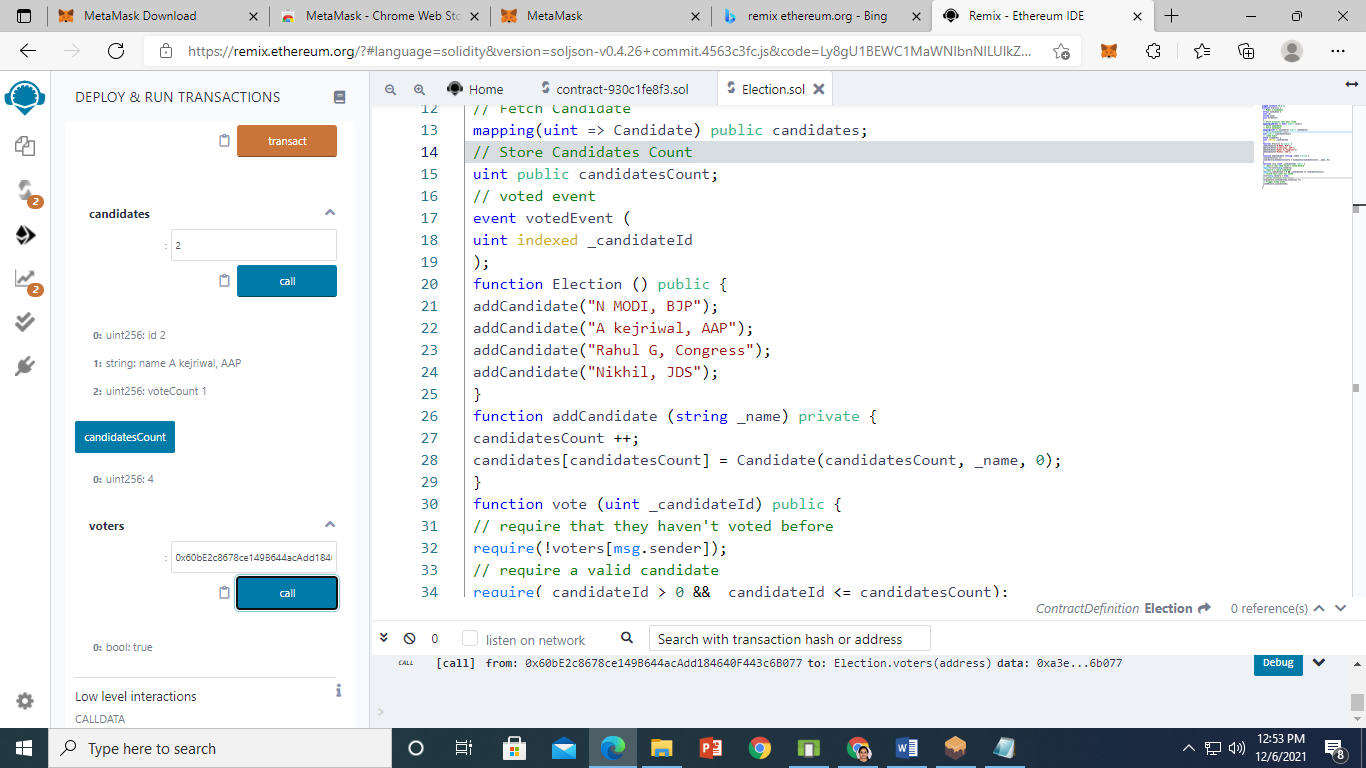
On Remix IDE now deploy the Election Contract.



Click on confirm.



When you click on transact. Metamask window will ask for confirmation. So click on confirm.



Now you can observe Ganache and Metamask

