**UNIVERSITY OF MUMBAI**

A PROJECT REPORT ON

**“Patient DApp“**

SUBMITTED BY

**Patel Arun Ramjanak**

**Vishnu Waman Thakur Charitable Trust’s**

**VIVA INSTITUTE OF TECHNOLOGY Shirgaon, Virar(East)**

**2022-23**

**Vishnu Waman Thakur Charitable Trust’s**

**VIVA INSTITUTE OF TECHNOLOGY**

**Shirgaon, Virar(East)**

****

**CERTIFICATE**

**This is to certify that**

**Mr. Patel Arun Ramjanak**

**Has satisfactorily completed the project entitled**

**Patient DApp**

**Towards the partial fulfillment of the**

**MASTER OF COMPUTER APPLICATION (MCA)**

**As laid by University of Mumbai.**

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**HOD External Examiner Internal Guide**

**Introduction**

**Develop Patient contract using solidity programming. Create a contract called Patient, the contract members are PatientName, PatientDoctor, PatientMedicine, PatientDisease, PatientConsulationFees. The smart contract should have the functions like SetPatientData and GetPatientData. 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,

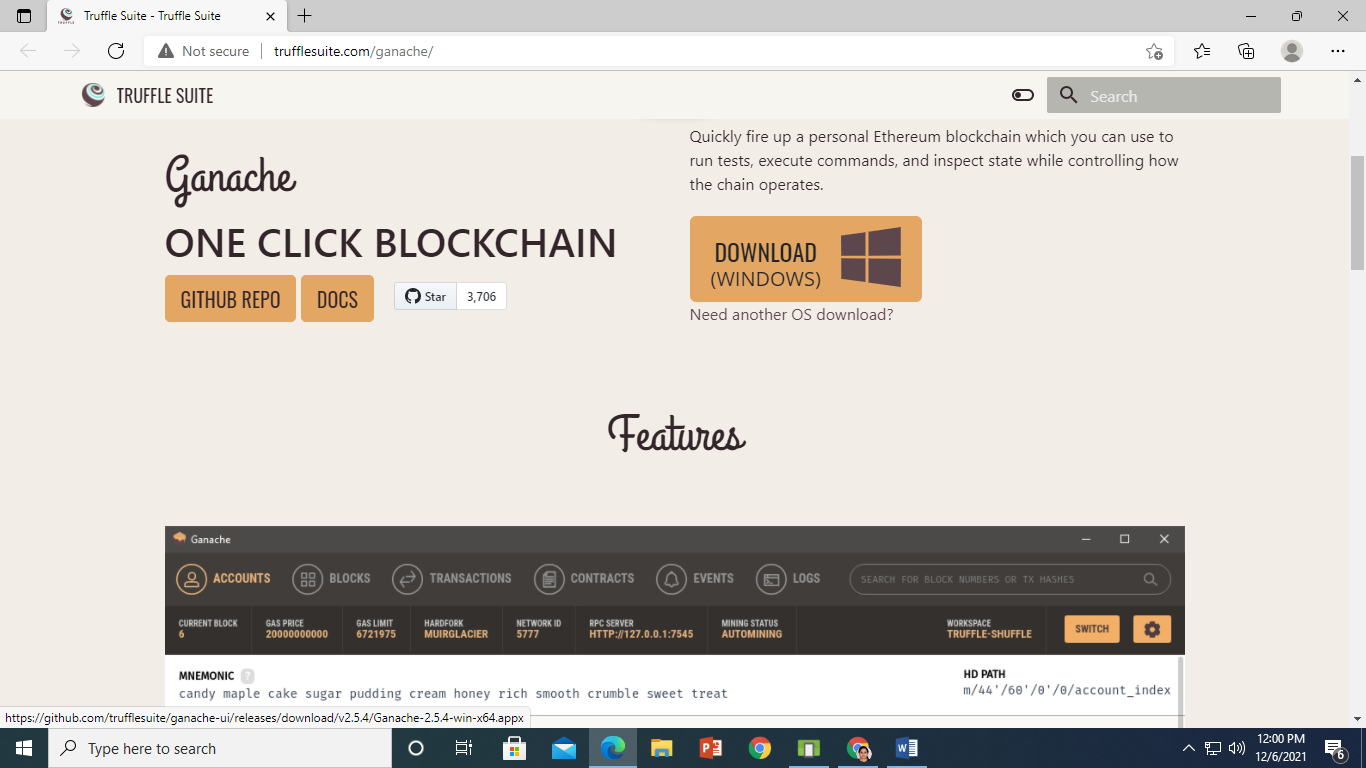
along with the gas cost and transactions.

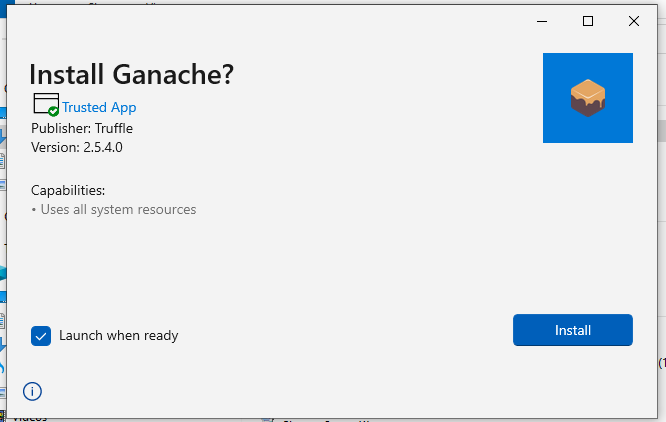
➢ Transactions Page -this list all the transactions that have taken place on the personal blockchain.

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

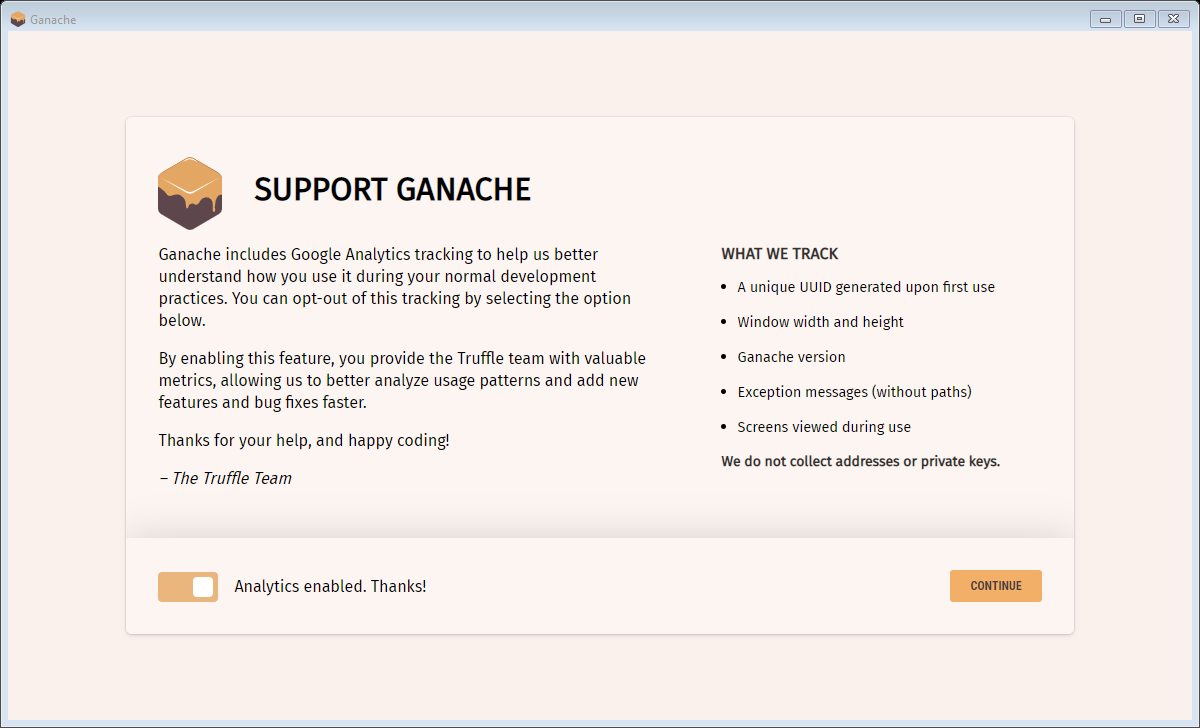
application.

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

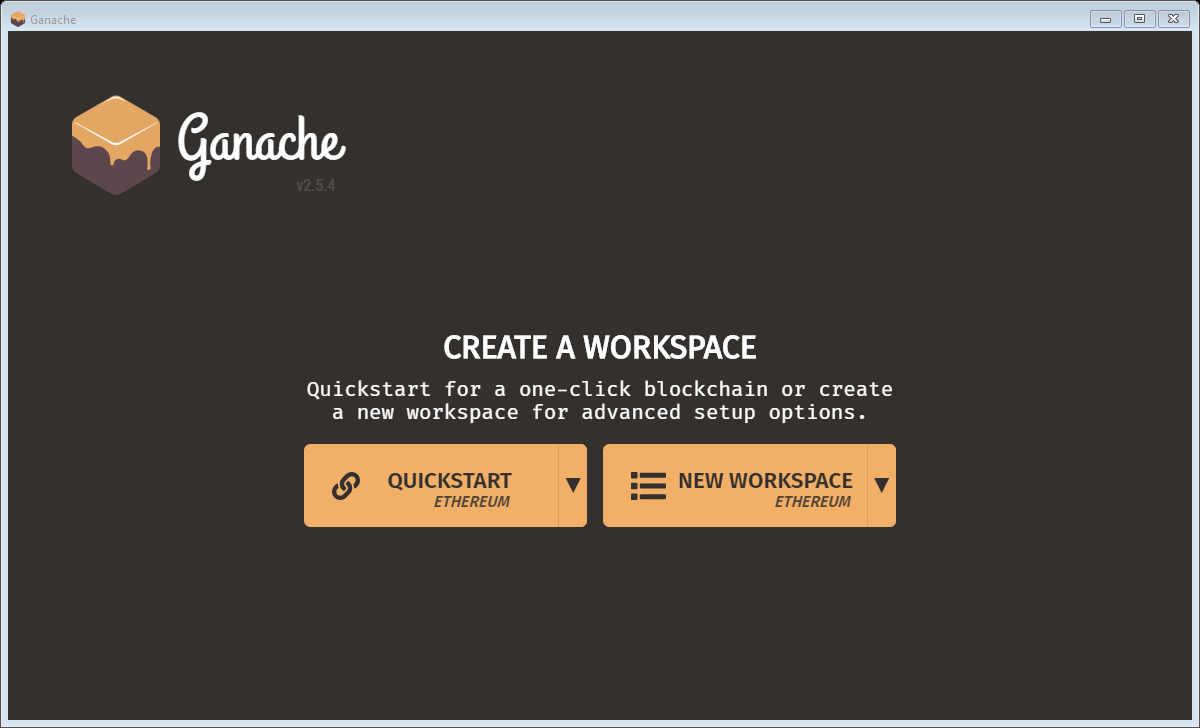


****

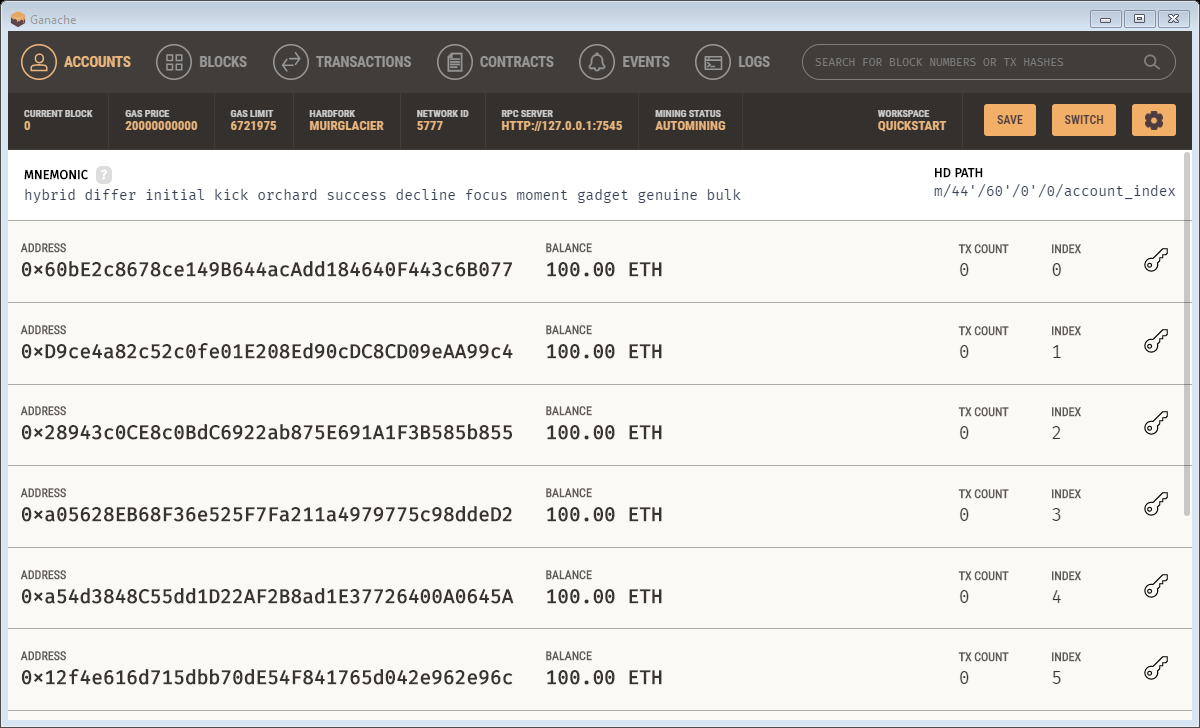
**Click on install**

****

**Click on Continue**

****

**Click on QuickStart**

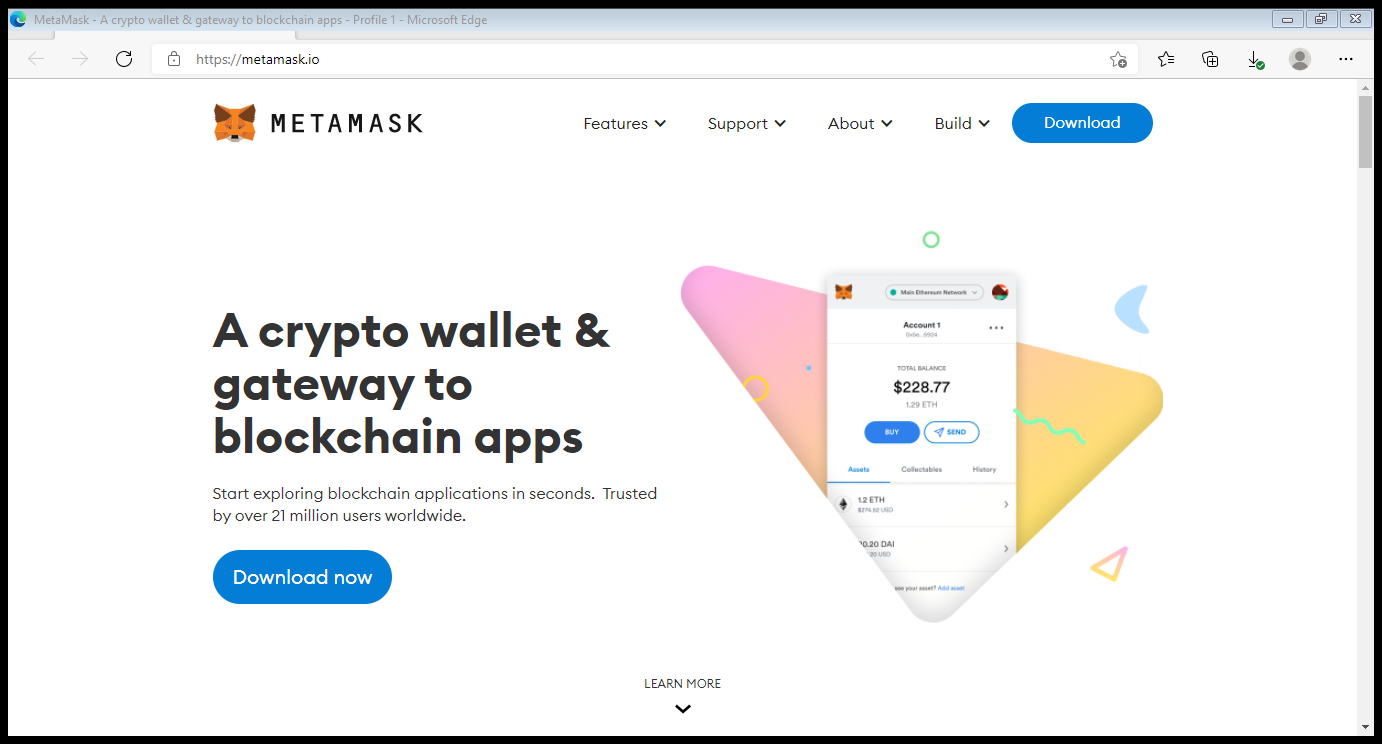
****

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

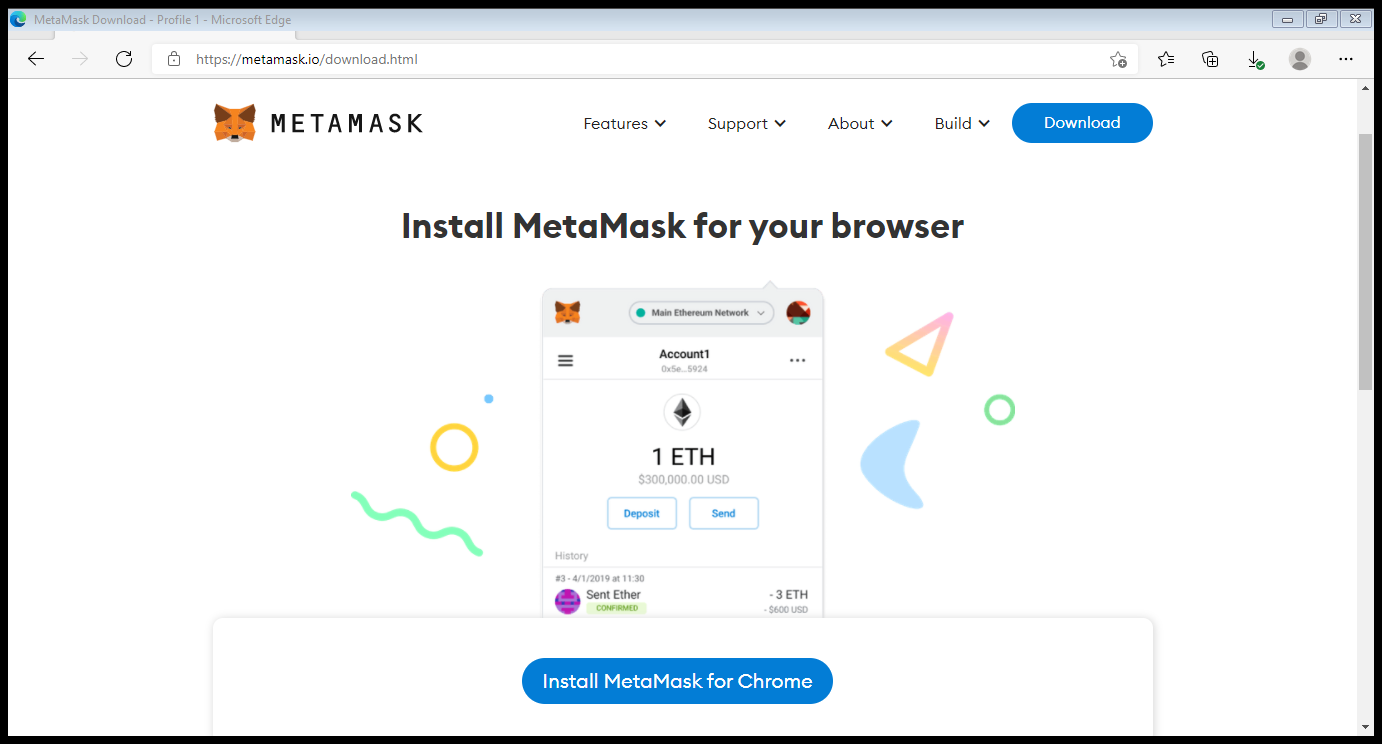
**STEP 2**

**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.



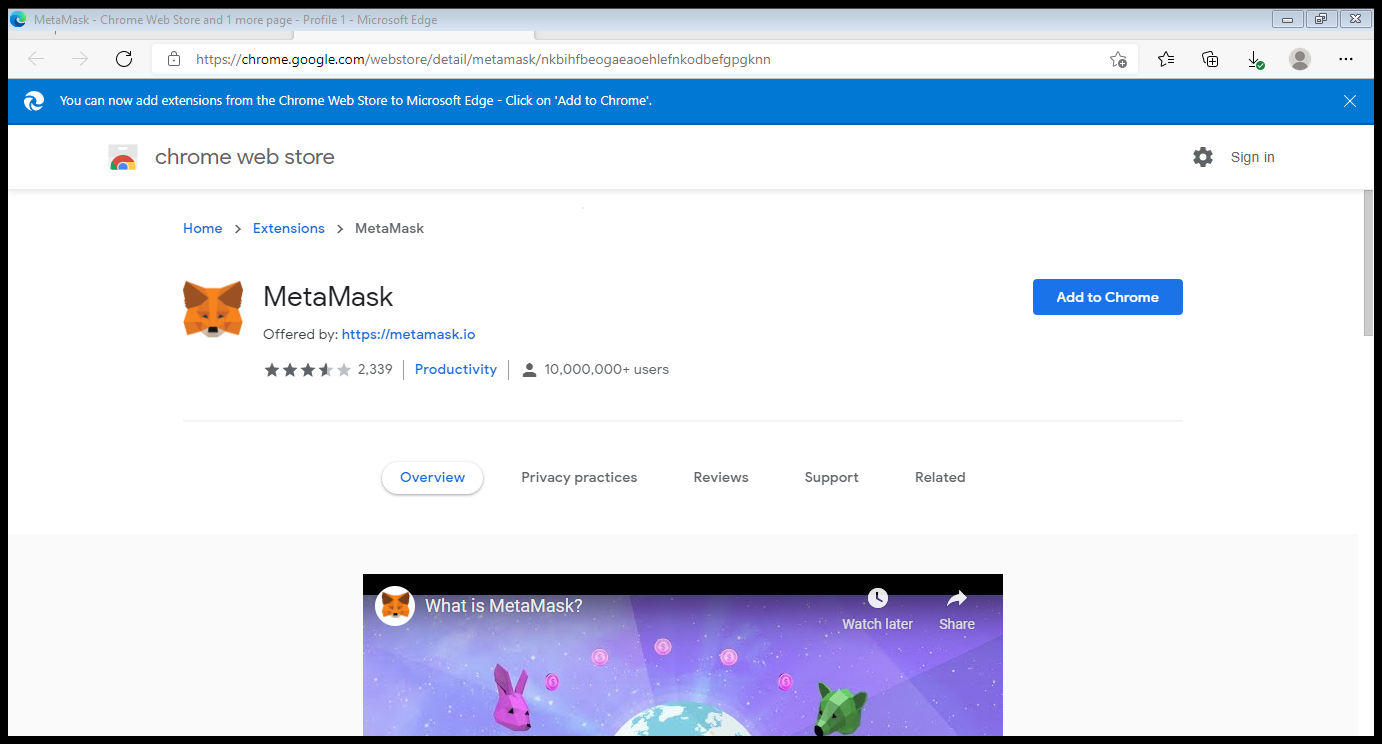
Click on download



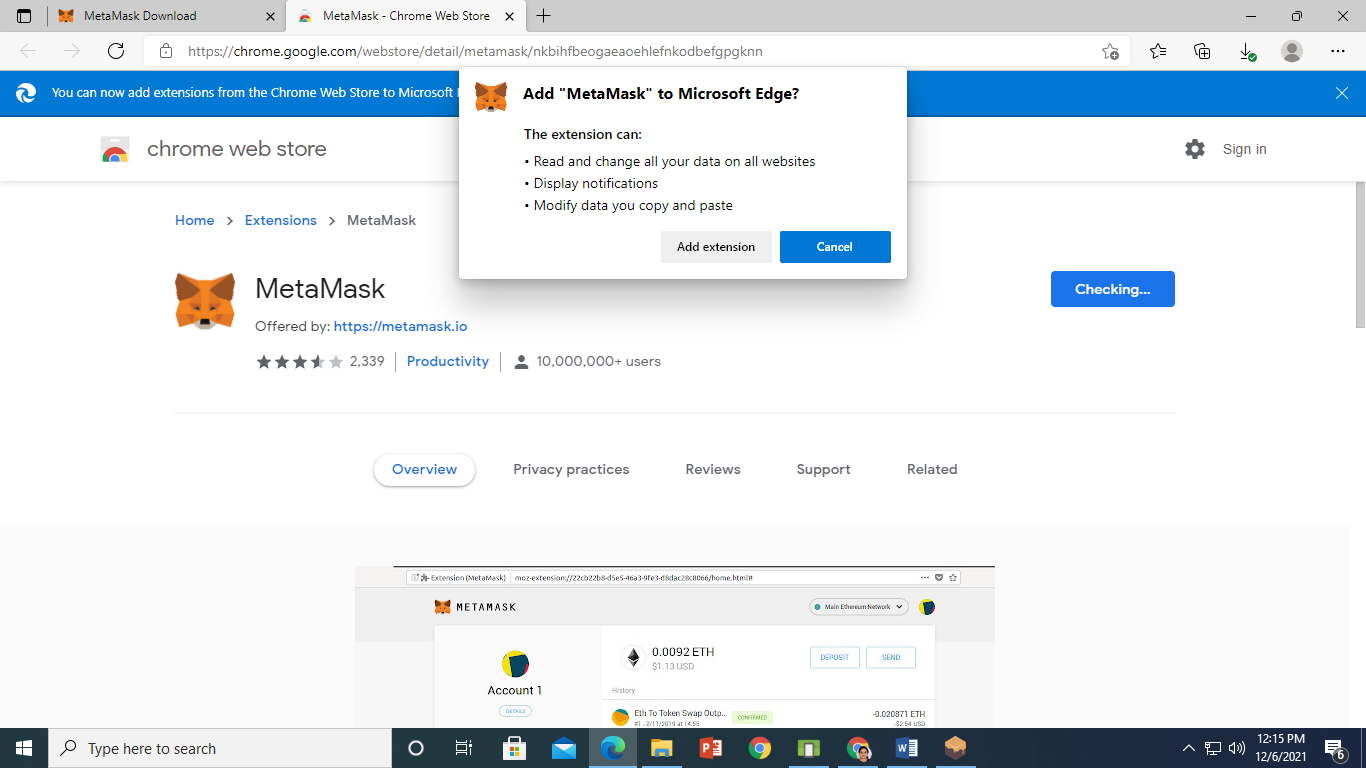
Click on Install MetaMask for Chrome.



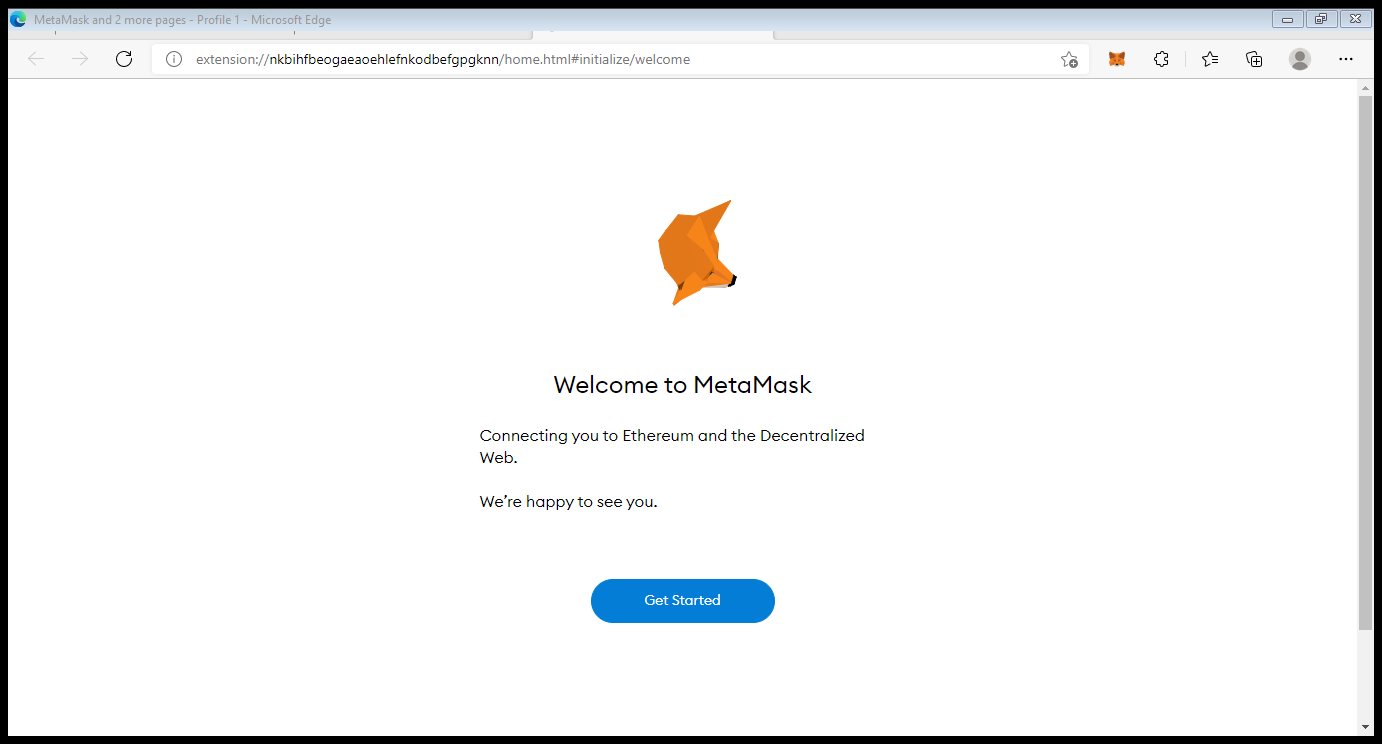
Click on Allow Extensions from other stores.



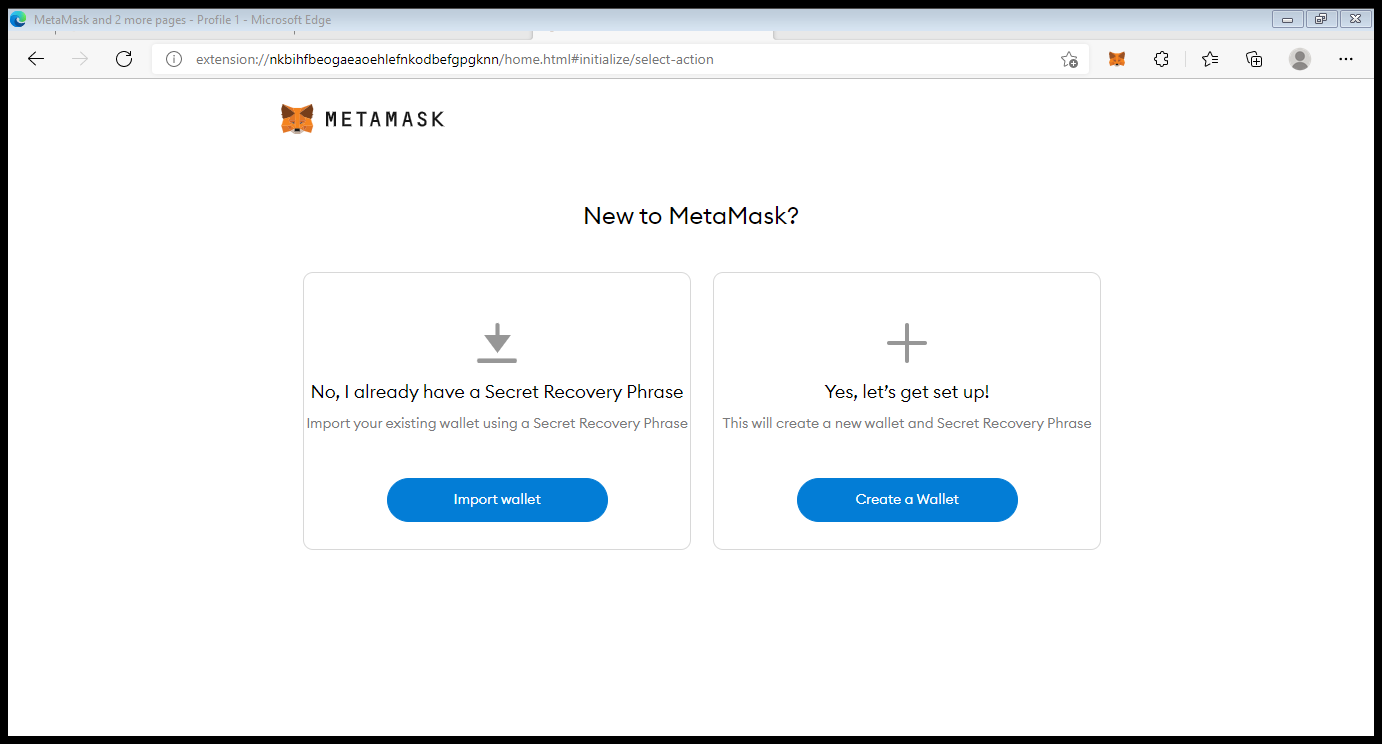
Click on Add to Chrome



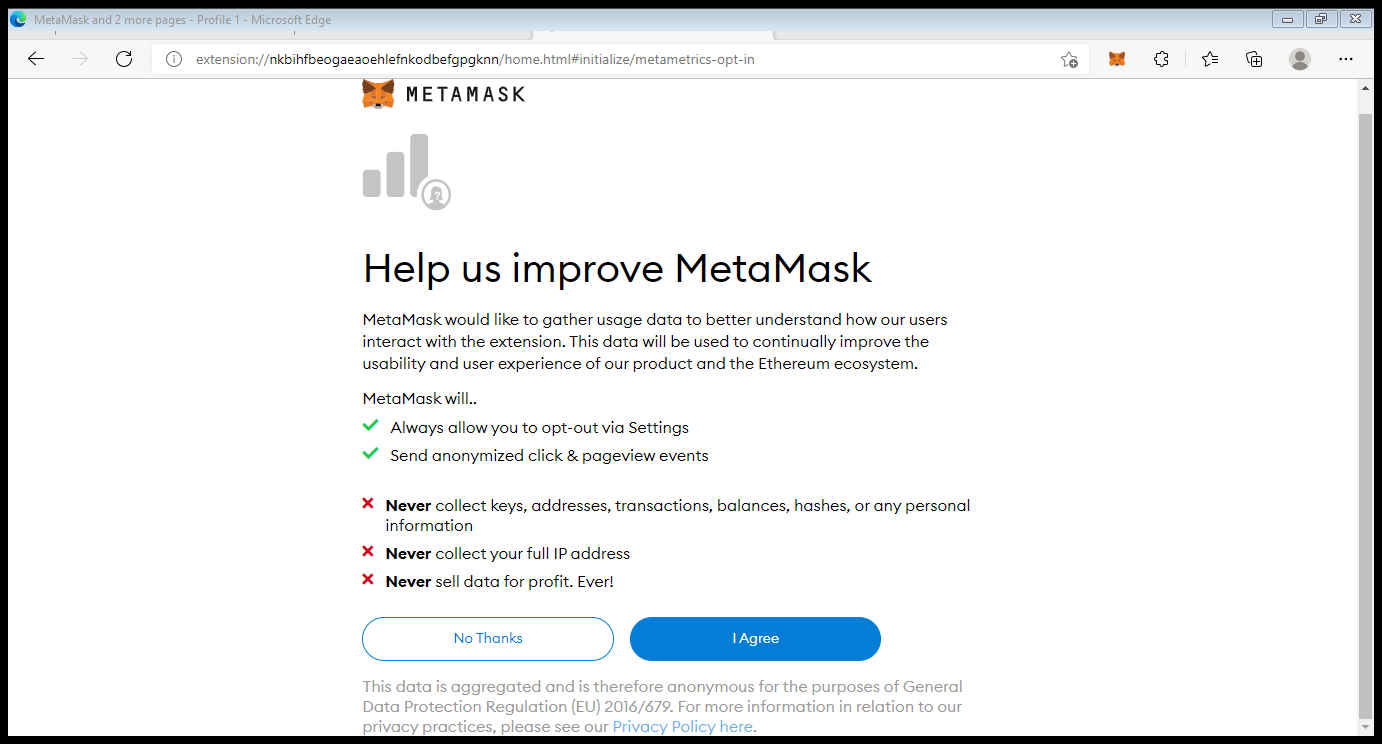
Click on Add extension



Click on Get Started



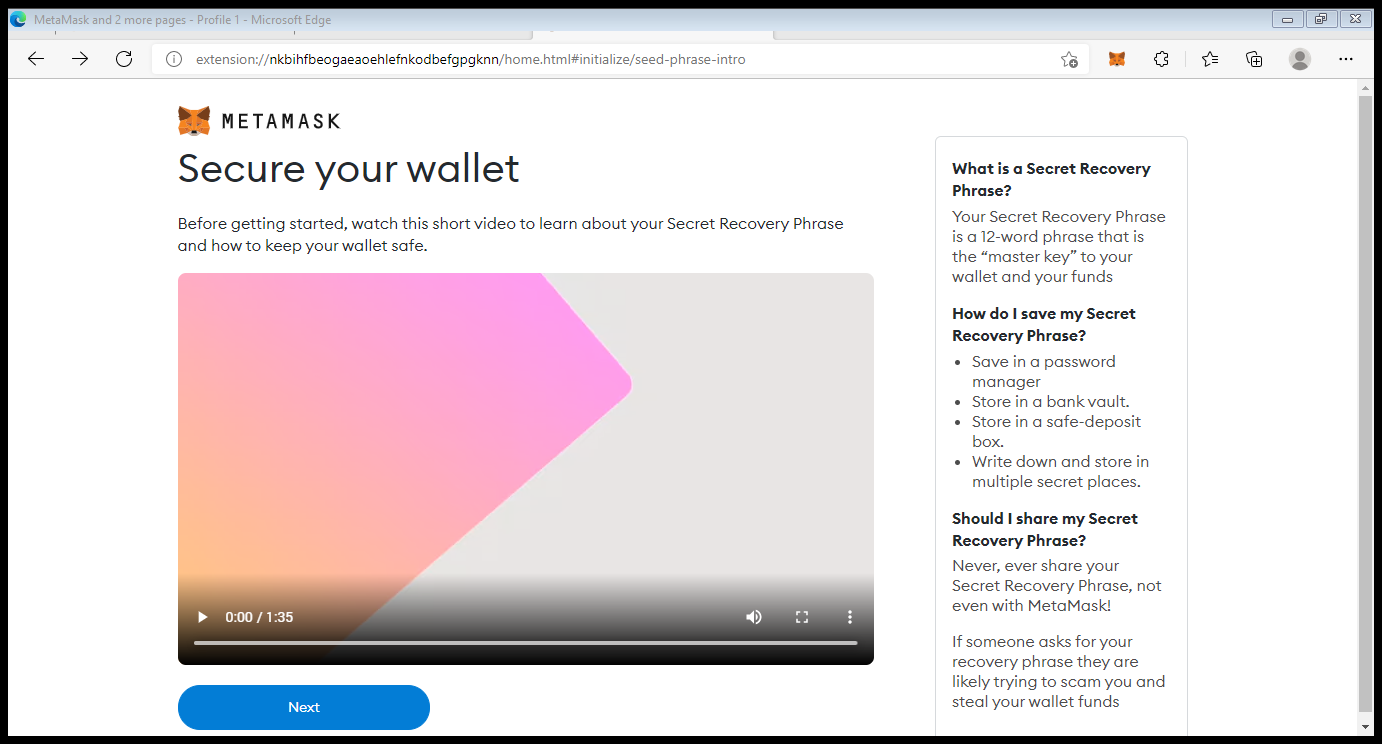
Click on Create Wallet



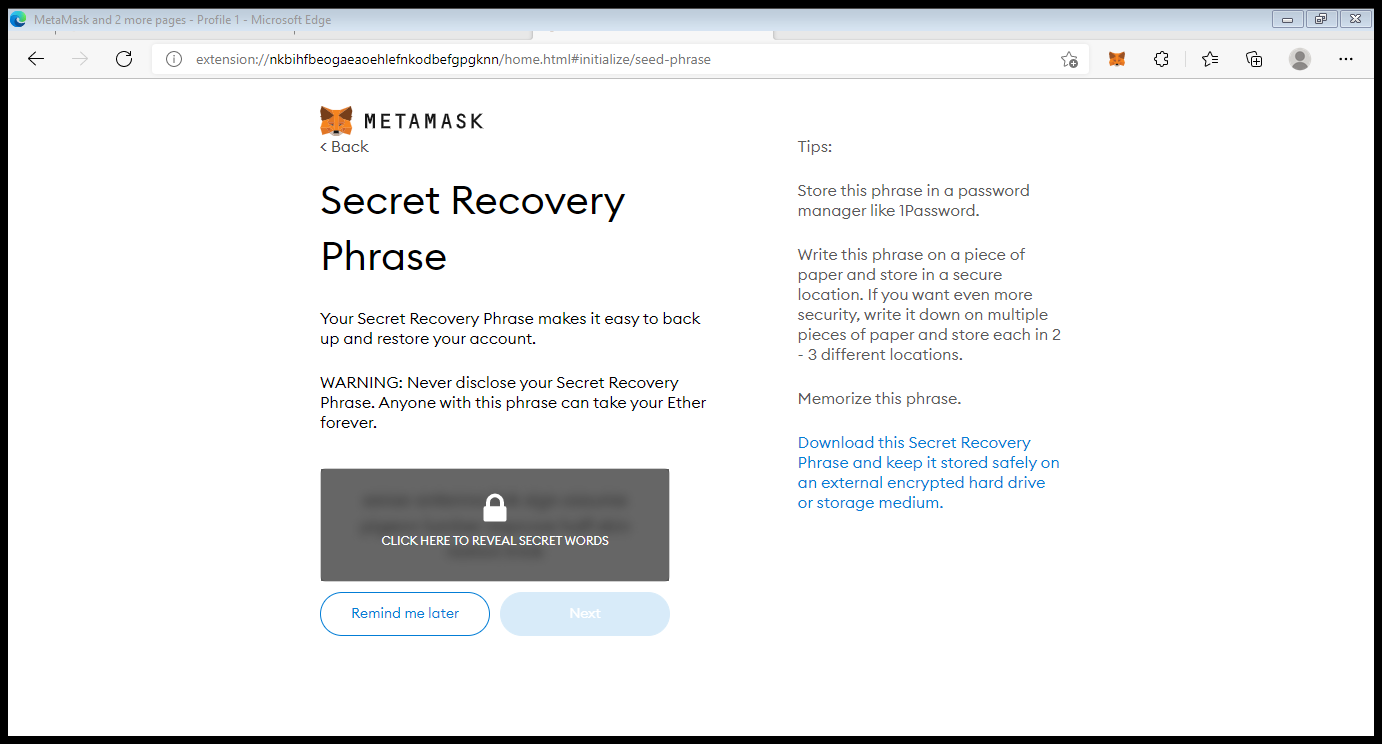
Click on I Agree



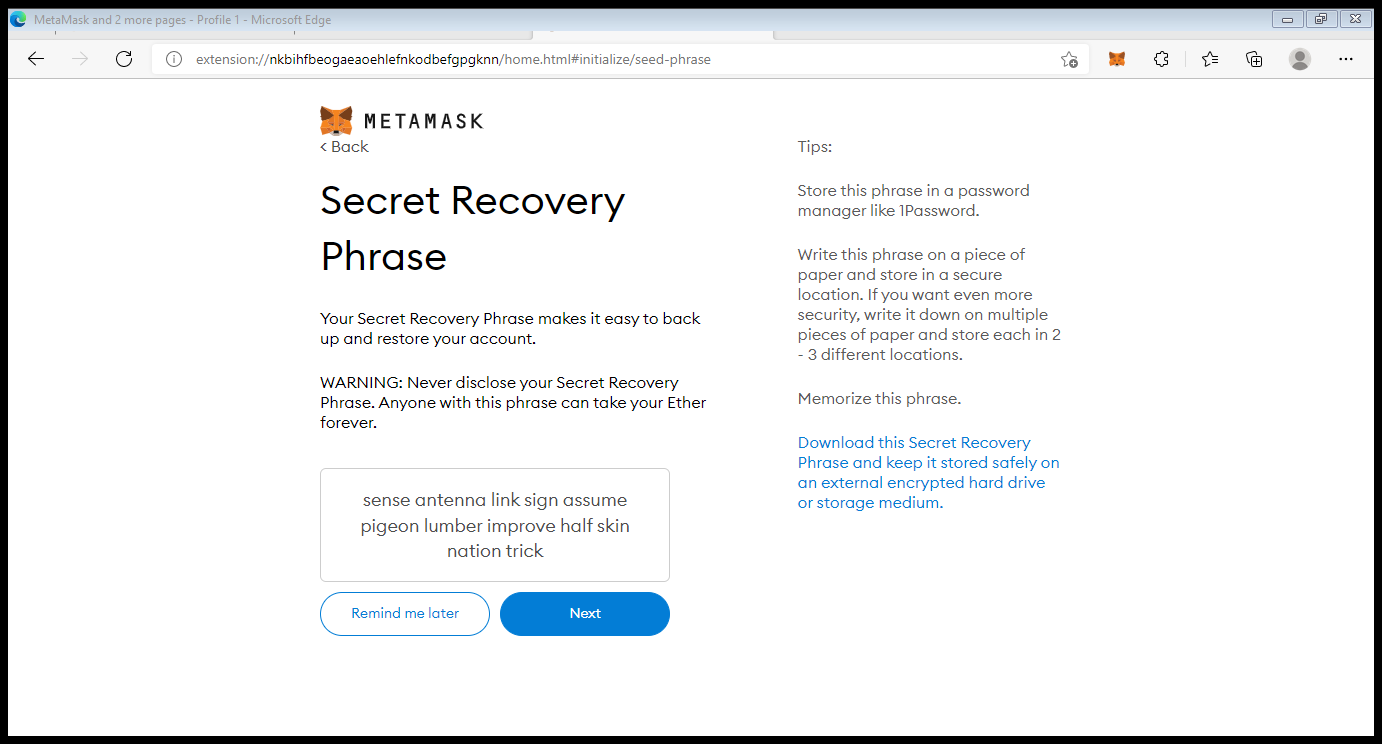
Mention password. Note down password further it is required and click on create.



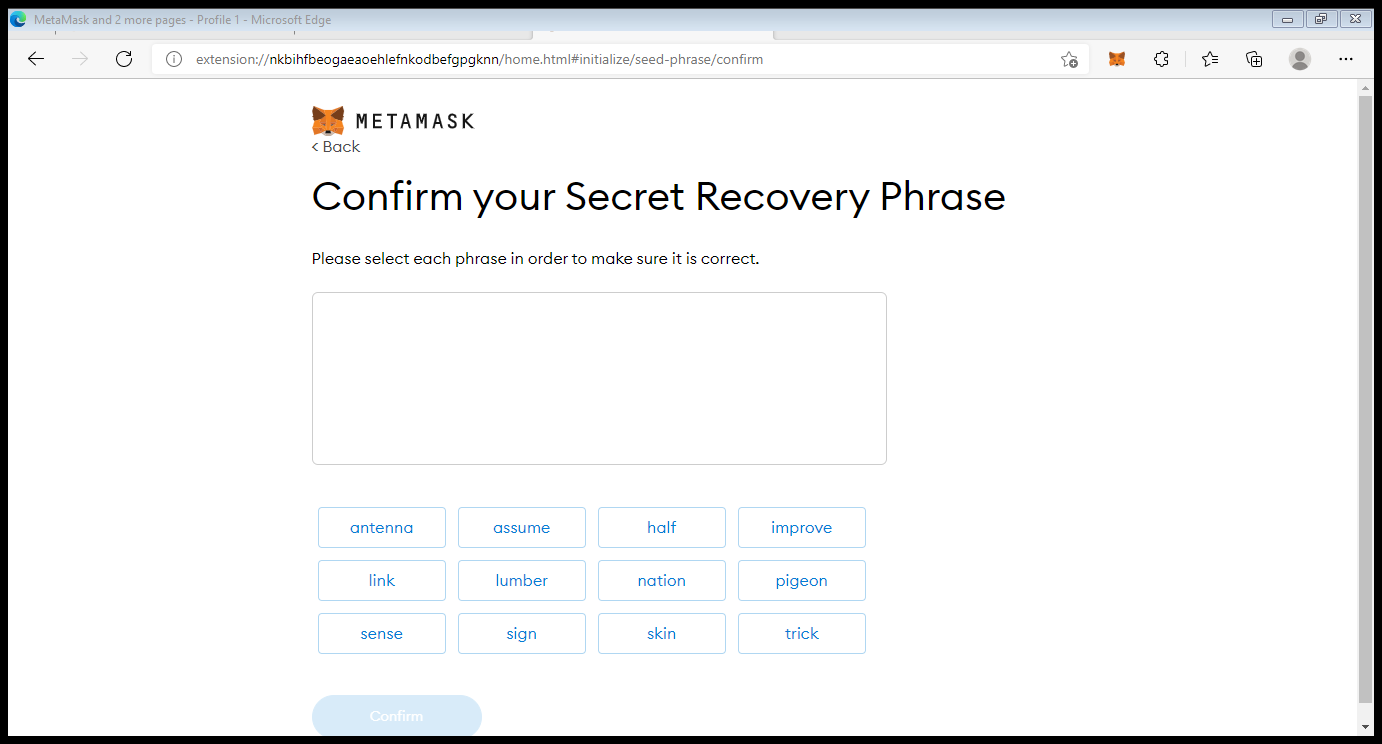
Click on Next.



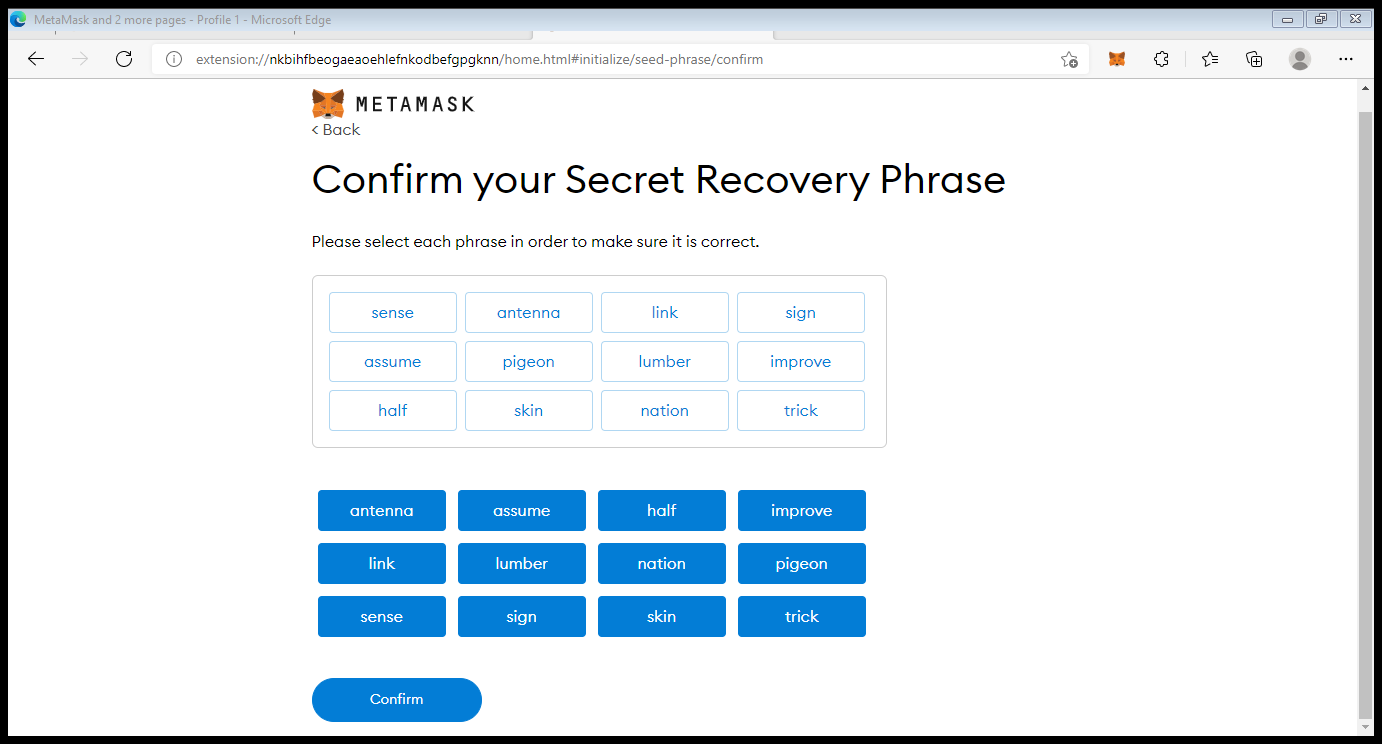
Click on Click here to reveal secret words.



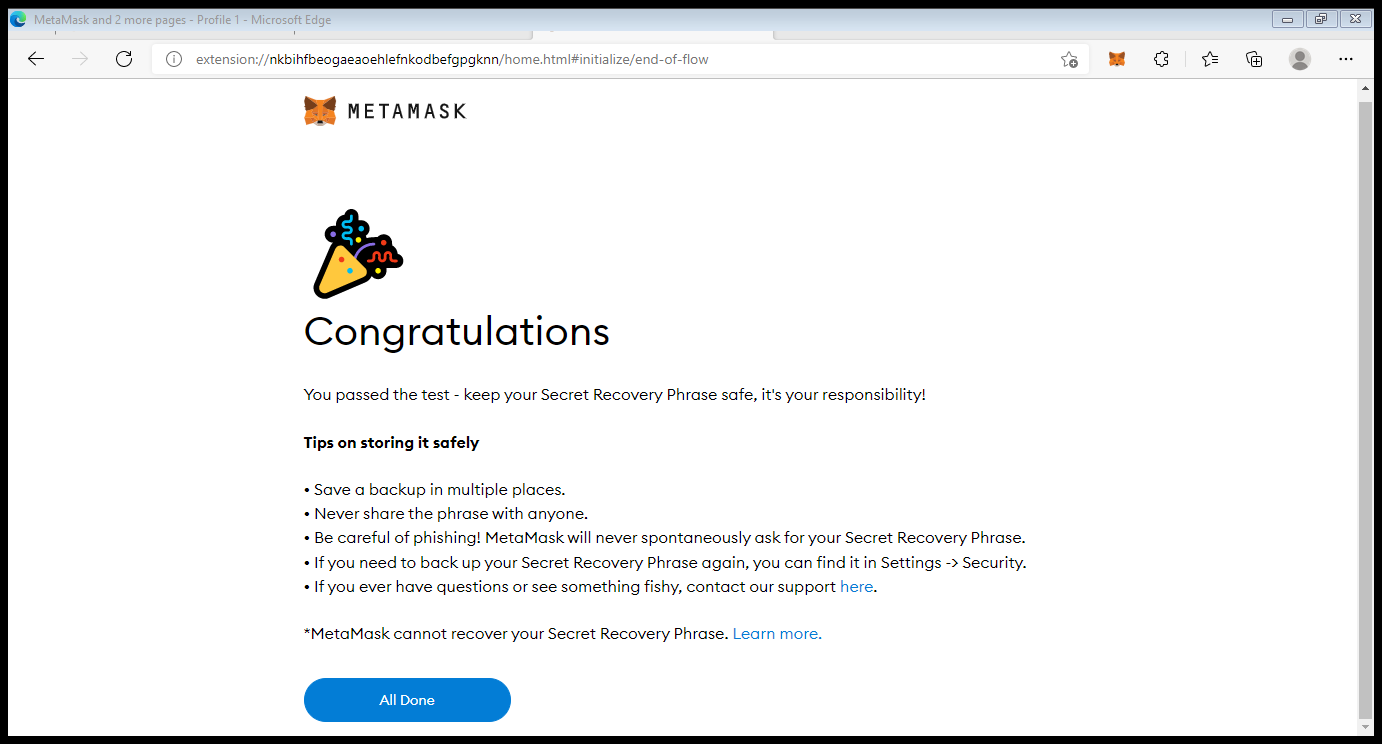
Copy all the words and save it in notepad. Then click on Next.



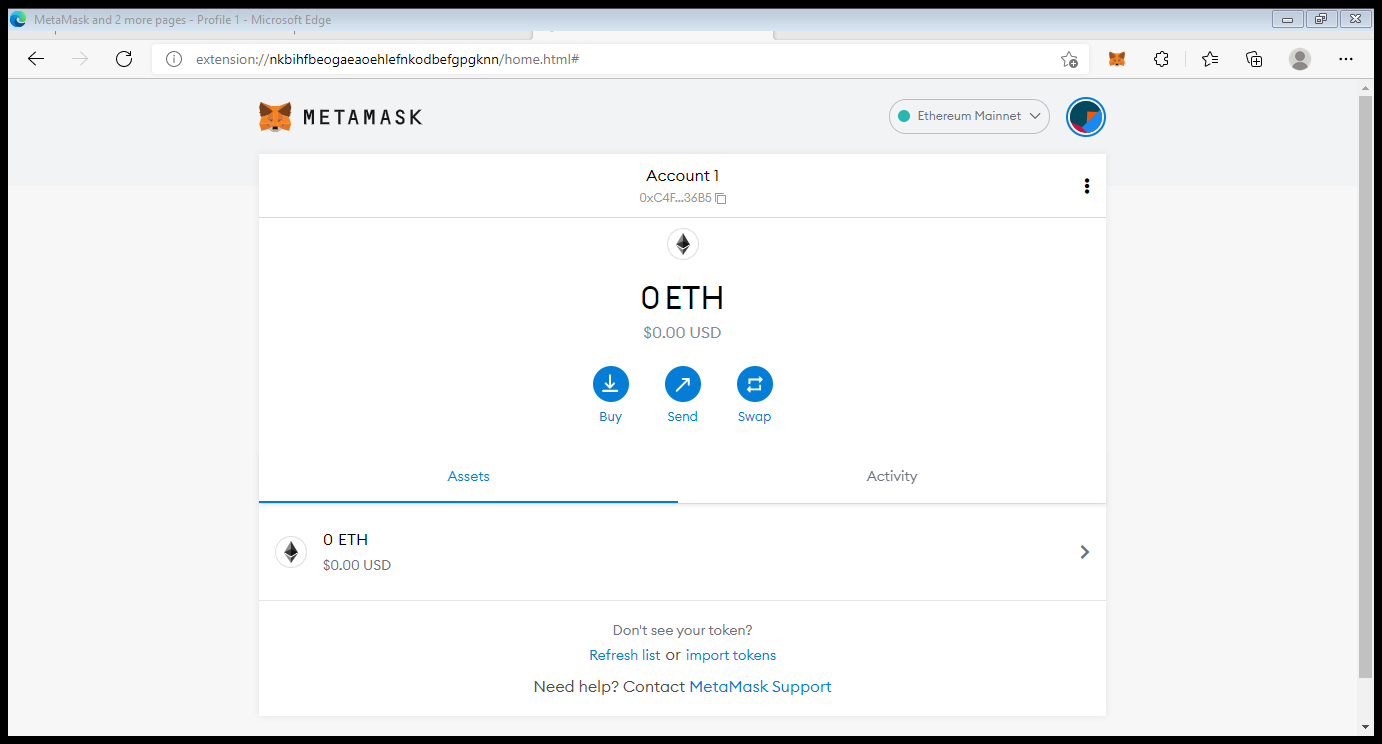
Refer the notepad and select the words in same sequence.

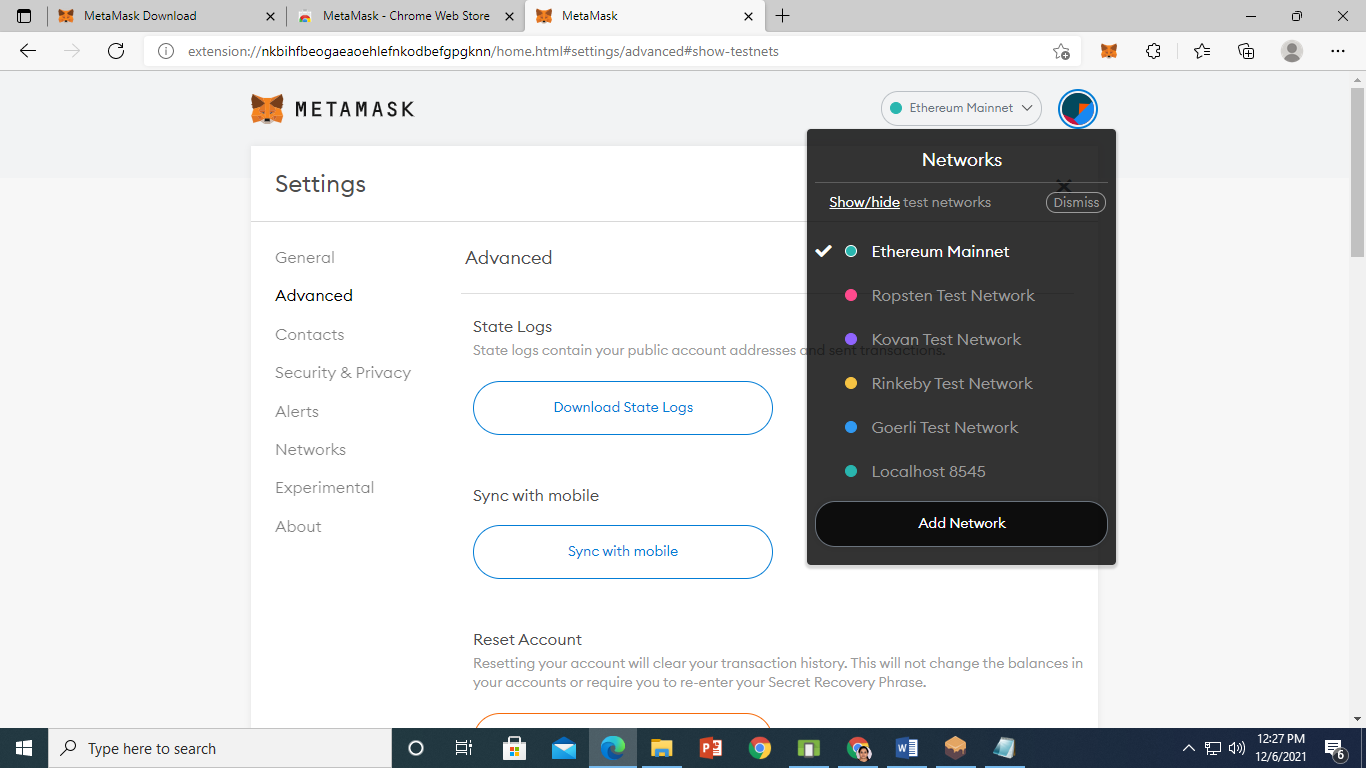


Click on confirm.

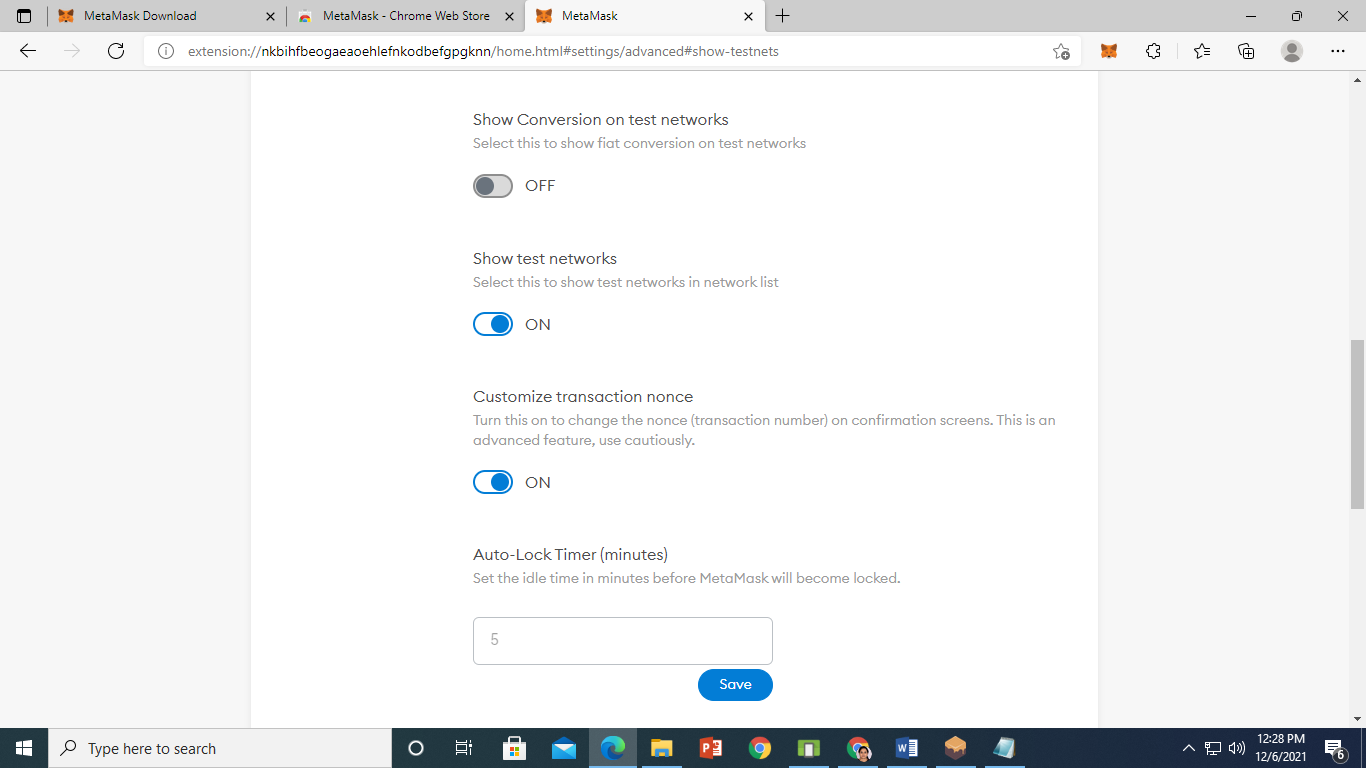


Click on All Done.

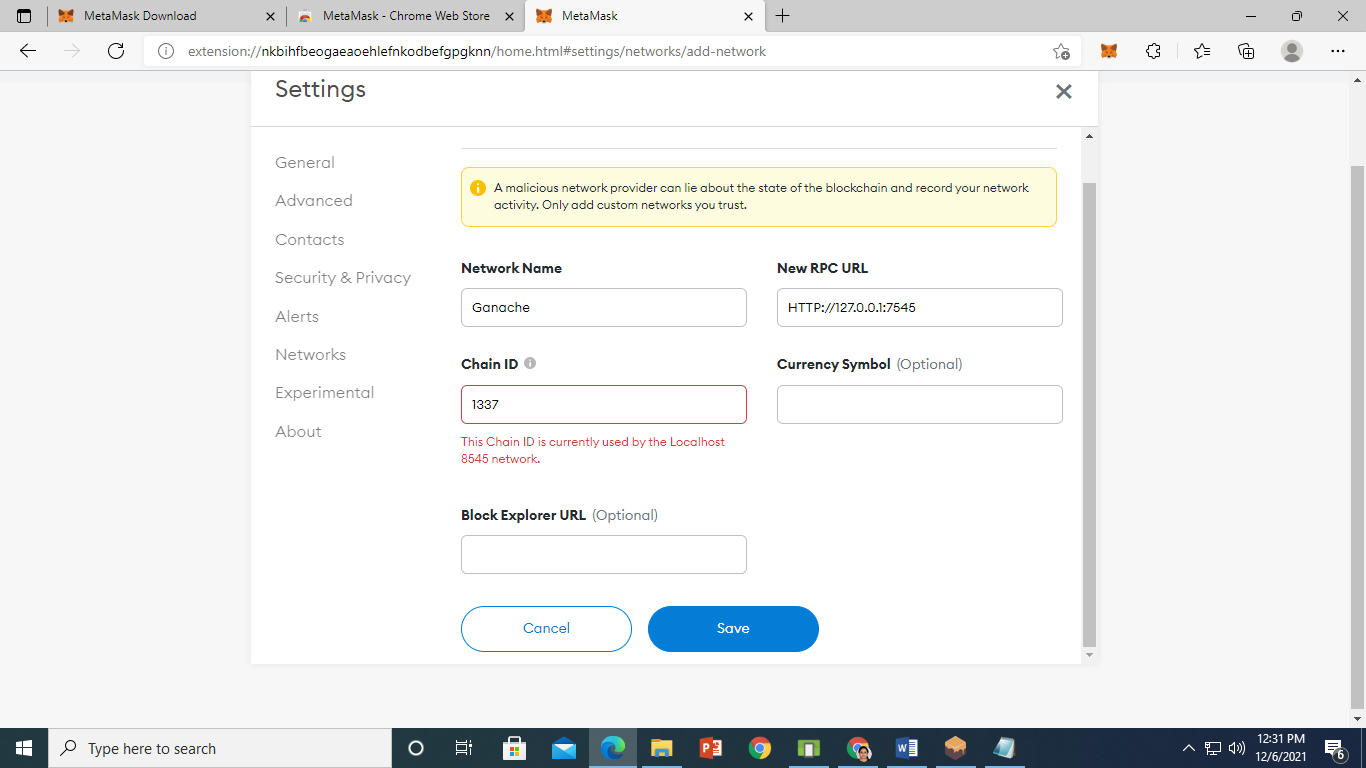




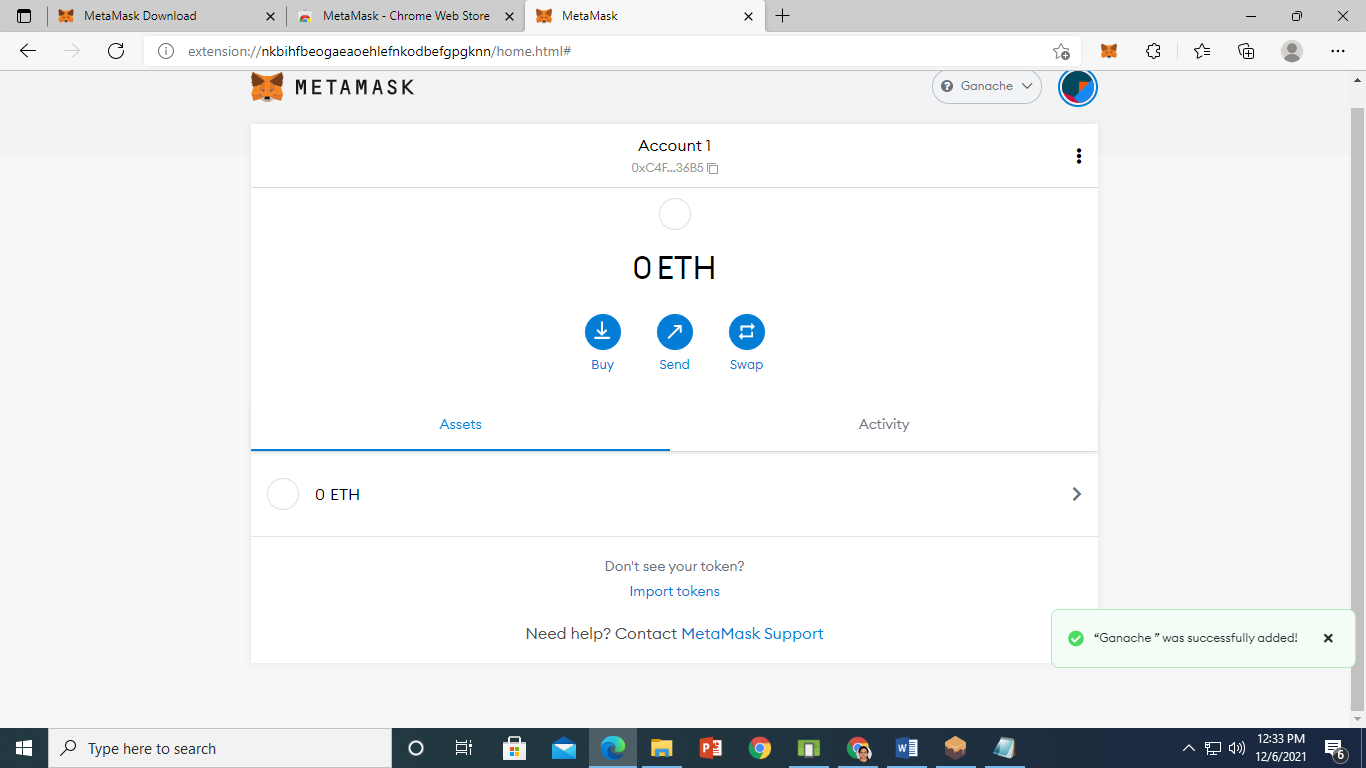
Click on Show/Hide test network.



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 Ethereum Smart Contract for Data of Health Record

pragma solidity >=0.4.22 <0.5.0;

contract  HealthRecord {

    string public PatientName;

    string public PatientDoctor;

    string public PatientMedicine;

    string public PatientDisease;

    uint256 public PatientConsulationFees;

    constructor (string memory NewPatientName,string memory NewPatientDoctor,string memory NewPatientMedicine,string memory NewPatientDisease,uint256 NewPatientConsulationFees)public{

        PatientName=NewPatientName;

        PatientDoctor=NewPatientDoctor;

        PatientMedicine=NewPatientMedicine;

        PatientDisease=NewPatientDisease;

        PatientConsulationFees=NewPatientConsulationFees;

    }

    function  SetPatientData(string memory SetPatientDisease,string memory SetPatientMedicine,uint256 SetPatientConsulationFees) public{

        PatientDisease=SetPatientDisease;

        PatientMedicine=SetPatientMedicine;

        PatientConsulationFees=SetPatientConsulationFees;

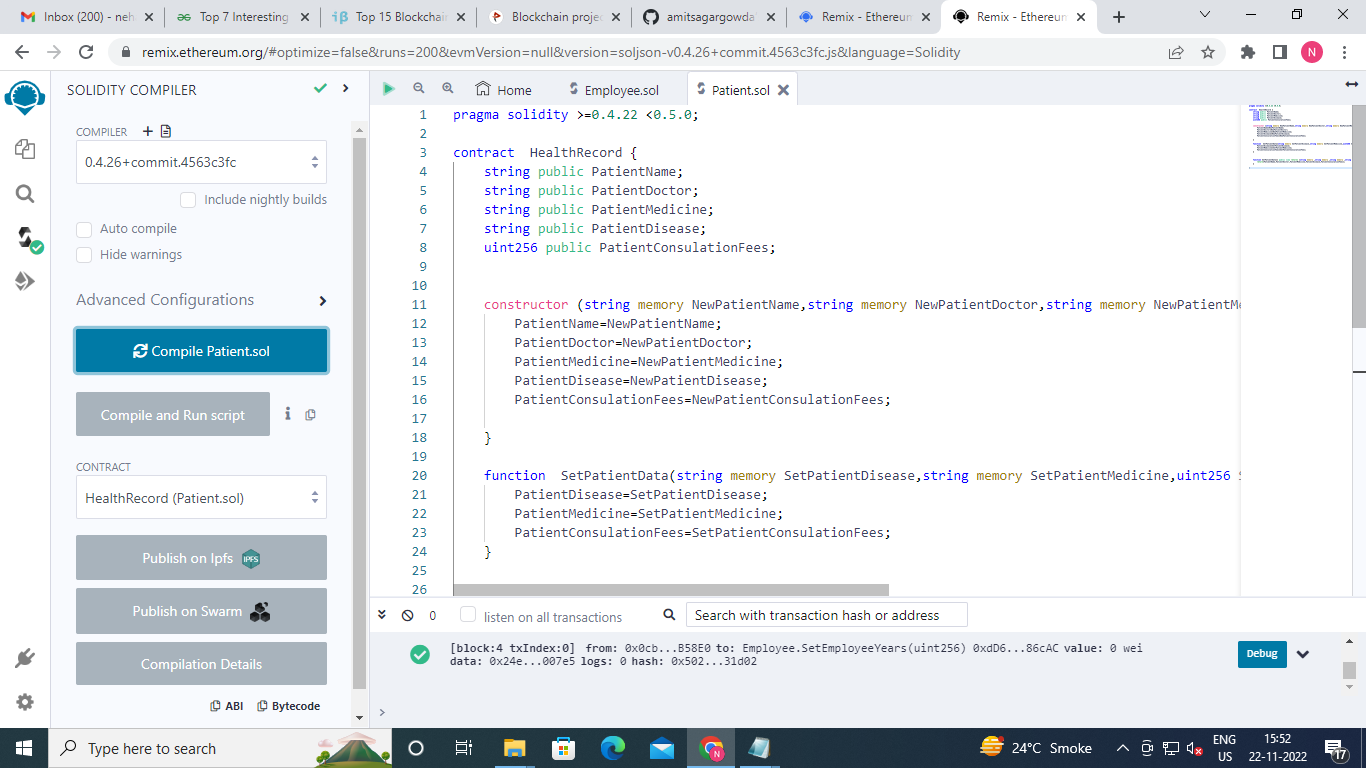
    }

    function GetPatientData() public view returns (string memory ,string memory ,string memory ,string memory,uint256 ){

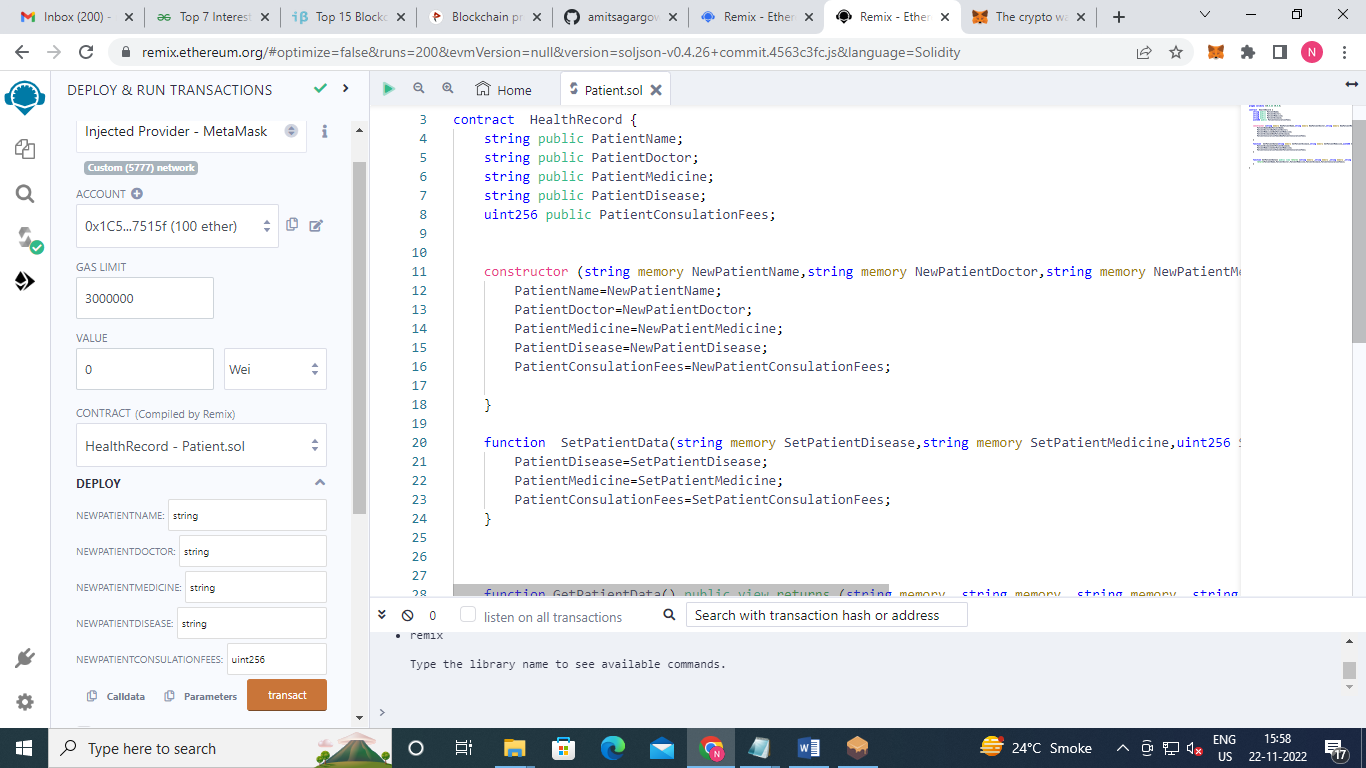
        return(PatientName,PatientDoctor,PatientMedicine,PatientDisease,PatientConsulationFees);

    }

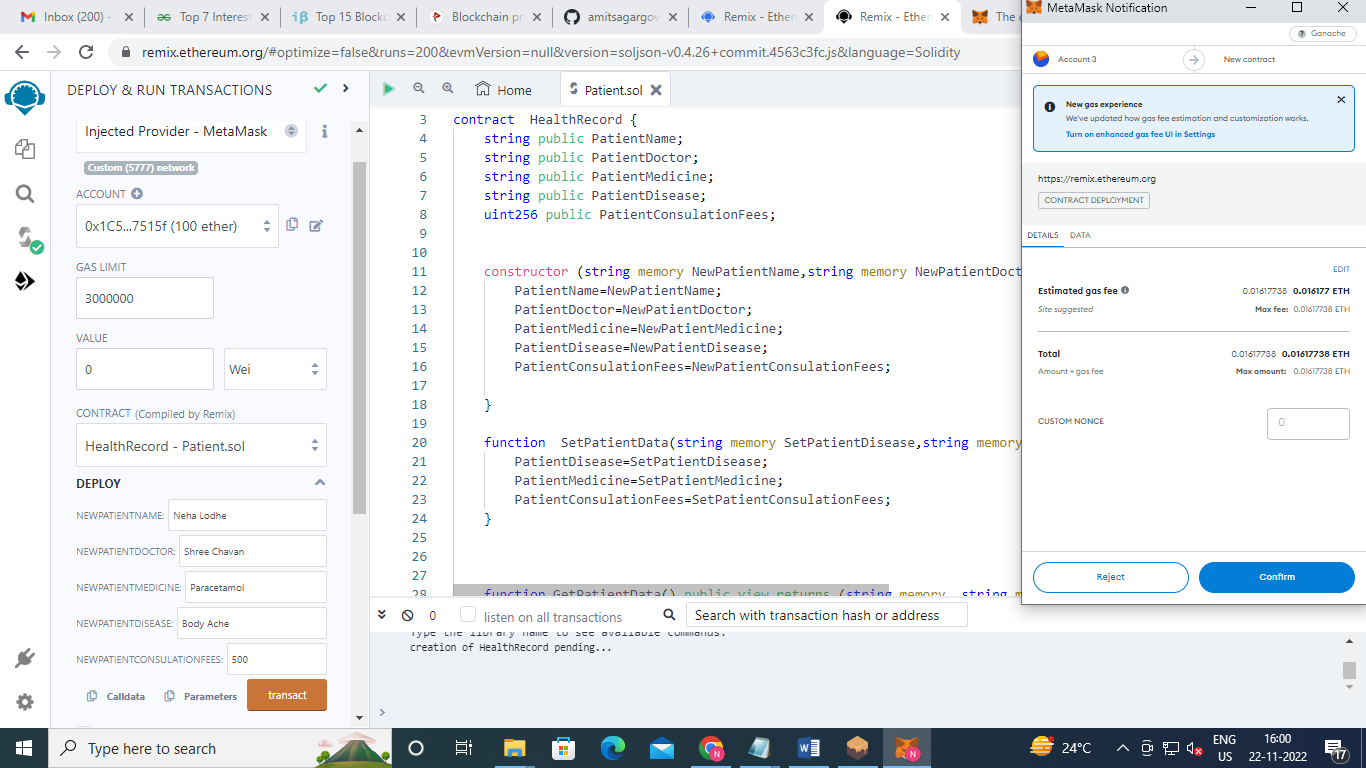
}



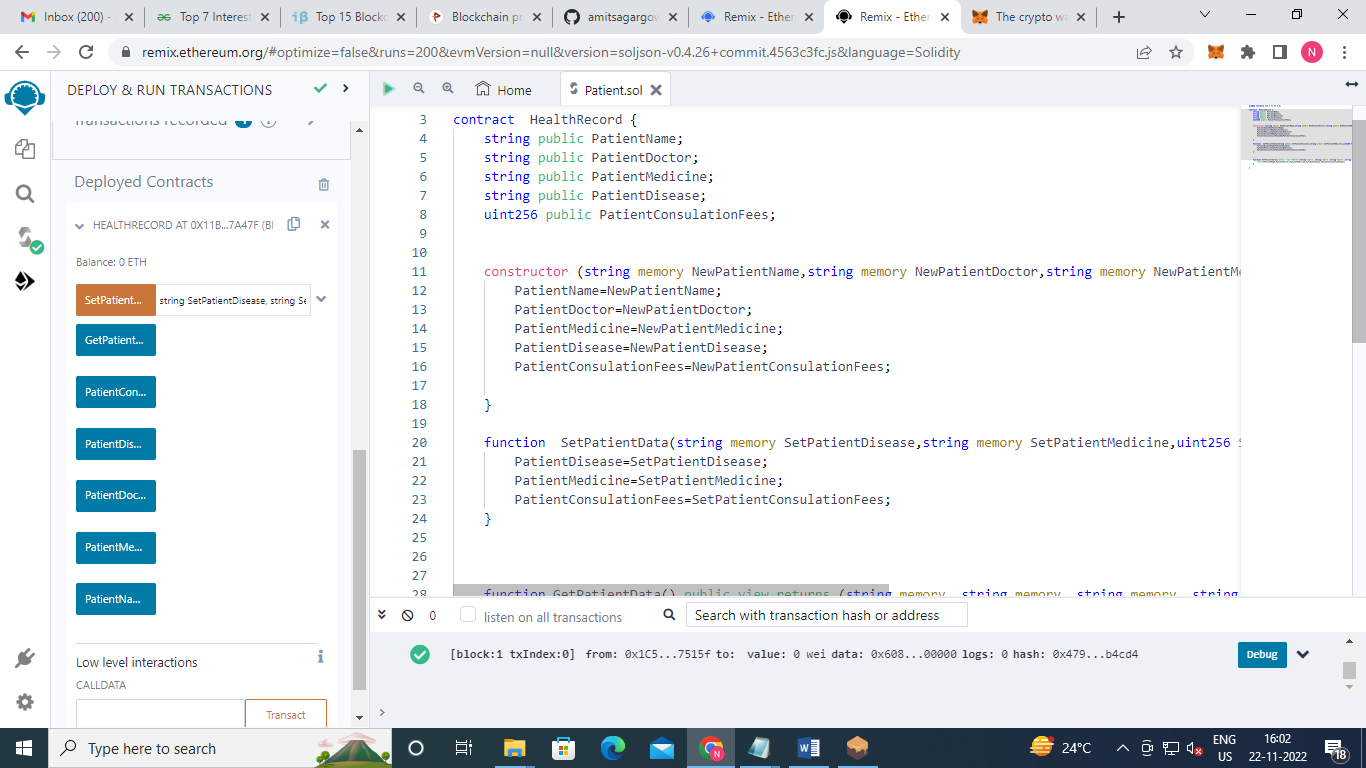
Click on Deploy and then expand dropdown of deploy



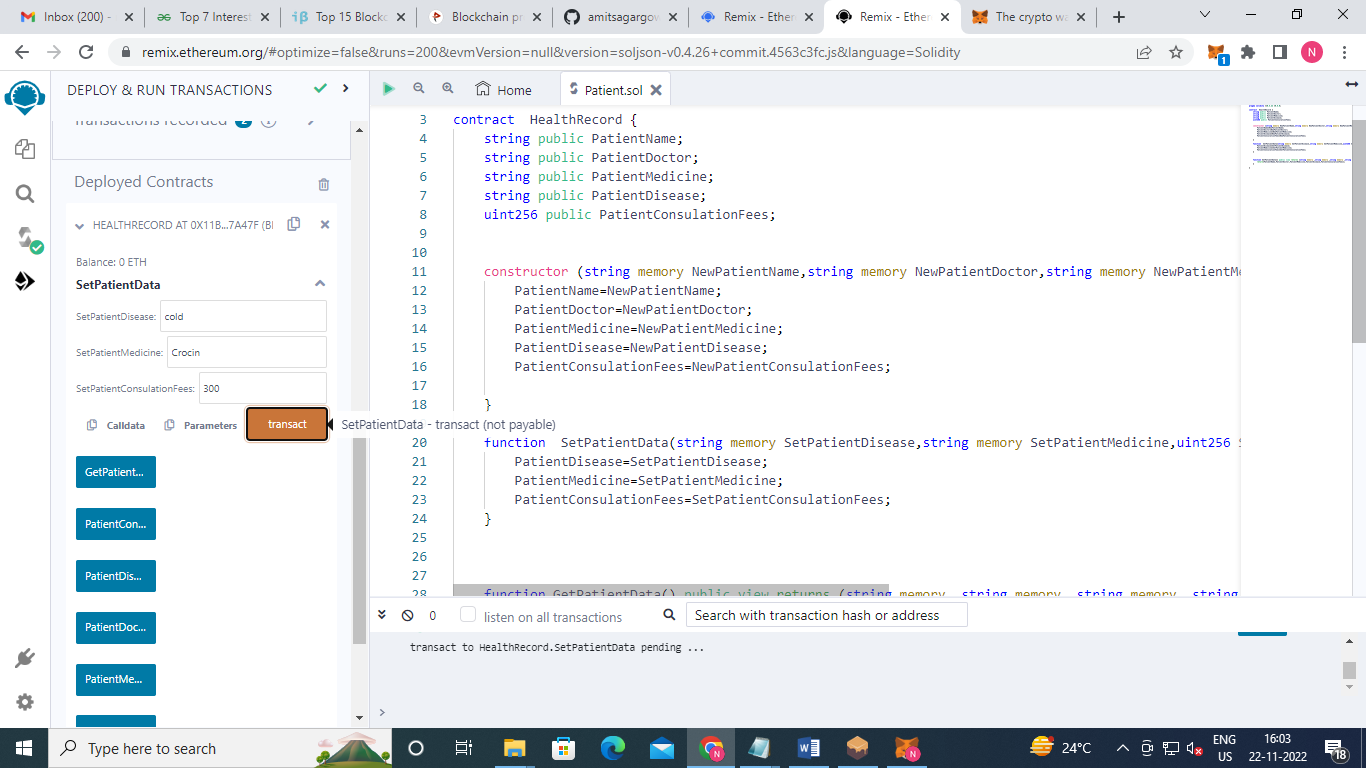
Fill the details of patient and then click on transact.



Click On Confirm.



Set the patient data and click on transact



Click on confirm

