

# Decentralized Greeting DApp

## **PROBLEM STATEMENT:**

**Objective:** Expand the basic greeting smart contract into a functional DApp.

### **Key Features:**

Create a smart contract to store and update greetings.

Build a frontend where users can read and update the greeting.

Connect the frontend with Ethereum using web3.js.

### **Objectives:**

Basics of smart contracts, web3.js integration, and DApp structure.

### **Tools:**

Solidity, Truffle, Ganache, MetaMask, and HTML/JavaScript.

---

## **Project Overview:**

The goal of this project is to create a DApp that allows users to:

- Read the current greeting stored on the blockchain.
- Update the greeting to a new message.

The smart contract stores the greeting, and the frontend allows users to interact with it. When the user updates the greeting, the new value is stored on the blockchain.

## **Setup the Development Environment:**

Before starting the project, ensure you have the following installed:

- Node.js
- Truffle
- Ganache
- MetaMask
- web3.js

### **Installation**

#### **Truffle :**

```
npm install -g truffle
```

#### **Node.js:**

<https://nodejs.org/dist/v22.12.0/node-v22.12.0-x64.msi>

#### **Ganache:**

<https://github.com/trufflesuite/ganache-ui/releases>

## **MetaMask:**

Browser extension and create an account.

## **Write the Smart Contract:**

Create a folder for the project, and navigate to it using the terminal.

1.Initialize the Truffle project:

```
mkdir GreetingDApp
cd GreetingDApp
truffle init
```

2.Inside the **contracts** folder, create a new file called **Greeting.sol**:

### **Greeting.sol:**

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Greeting {
    string public greeting;

    constructor(string memory _greeting) {
        greeting = _greeting;
    }

    function getGreeting() public view returns (string memory) {
        return greeting;
    }

    function setGreeting(string memory _greeting) public {
        greeting = _greeting;
    }
}
```

## Test the Smart Contract:

Create a migration script in the **migrations** folder:

1. Create a new file **2\_deploy\_greeting.js**:

**2\_deploy\_greeting.js:**

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

module.exports = function(deployer) {
  deployer.deploy(Greeting, "Hello, Ethereum!");
};
```

2. Compile and deploy the smart contract to the local Ganache blockchain:

```
truffle compile
```

```
truffle migrate
```

When we migrate we will get an build.js file

## **OUTPUT:**

```
C:\Users\TEJASWINI.K\DecentralizedGreeting>truffle compile
```

```
Compiling your contracts...
```

```
=====
```

```
> Compiling .\contracts\Greeting.sol
```

```
> Artifacts written to C:\Users\TEJASWINI.K\DecentralizedGreeting\build\contracts
```

```
> Compiled successfully using:
```

```
- solc: 0.8.0+commit.c7dfd78e.Emscripten.clang
```

```

C:\Users\TEJASWINI.K\DecentralizedGreeting>truffle migrate

Compiling your contracts...
=====
> Compiling .\contracts\Greeting.sol
> Artifacts written to C:\Users\TEJASWINI.K\DecentralizedGreeting\build\contracts
> Compiled successfully using:
  - solc: 0.8.0+commit.c7dfd78e.Emscripten.clang

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

2_deploy_contracts.js
=====

Replacing 'Greeting'
-----
> transaction hash:  0x12e73507d7f807357381be665875d5374b23320b7aaa8179ed2fe4b5dca82468
> Blocks: 0         Seconds: 0
> contract address: 0x63ad331d189039C77bcaF2F455a8A072Bdb6689f
> block number:     12
> block timestamp:  1734516173
> account:          0x28c203924599Bf027dA28dDe49915A593cA3d914
> balance:          98.997813386798288018
> gas used:          367142 (0x59a26)
> gas price:         2.707044547 gwei
> value sent:        0 ETH
> total cost:        0.000993869749074674 ETH

> Saving artifacts
-----
> Total cost:        0.000993869749074674 ETH

Summary

```

The block has been added:

BLOCK	MINED ON	GAS USED	
13	2024-12-18 15:33:46	367142	1 TRANSACTION

## **Build the Frontend:**

Create a frontend folder inside your project and create an **index.html** file

### **Index.html:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Decentralized Greeting DApp</title>
</head>
<body>
  <h1>Decentralized Greeting DApp</h1>
  <div>
    <p>Current Greeting: <span id="currentGreeting">Loading...</span></p>
    <input type="text" id="newGreeting" placeholder="Enter new greeting">
    <button onclick="updateGreeting()">Update Greeting</button>
  </div>
  <script
    src="https://cdn.jsdelivr.net/npm/web3@1.8.1/dist/web3.min.js"></script>
  <script src="app.js"></script>
</body>
</html>
```

## **Connect the Frontend with Ethereum using web3.js:**

In the same frontend folder, create a **app.js** file:

### **app.js:**

```
const contractAddress = "0x1a1D3a32111B4a4873ed8DE589175b49ACB15C8c"; //
```

Replace with your contract address

```
const contractABI = [{
  "inputs": [
    {
      "internalType": "string",
      "name": "initialGreeting",
      "type": "string"
    }
  ],
  "stateMutability": "nonpayable",
```

```
    "type": "constructor"
  },
  {
    "inputs": [],
    "name": "greeting",
    "outputs": [
      {
        "internalType": "string",
        "name": "",
        "type": "string"
      }
    ],
    "stateMutability": "view",
    "type": "function",
    "constant": true
  },
  {
    "inputs": [
      {
        "internalType": "string",
        "name": "newGreeting",
        "type": "string"
      }
    ],
    "name": "updateGreeting",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [],
    "name": "getGreeting",
    "outputs": [
      {
        "internalType": "string",
        "name": "",
        "type": "string"
      }
    ]
  }
]
```

```

    }
  ],
  "stateMutability": "view",
  "type": "function",
  "constant": true
}]; // Replace with your contract ABI

const web3 = new Web3("http://127.0.0.1:7545"); // Connect to Ganache
const greetingContract = new web3.eth.Contract(contractABI, contractAddress);

// Load the current greeting
async function loadGreeting() {
  const greeting = await greetingContract.methods.getGreeting().call();
  document.getElementById("greeting").innerText = greeting;
}

// Update the greeting
async function updateGreeting() {
  const newGreeting = document.getElementById("newGreeting").value;
  const accounts = await web3.eth.getAccounts(); // Fetch accounts from Ganache
  await greetingContract.methods.updateGreeting(newGreeting).send({ from:
    accounts[0] });
  loadGreeting(); // Reload greeting
}

// Initialize
window.onload = loadGreeting;

```

### **Deploy the Smart Contract to the Blockchain:**

→First, ensure Ganache is running.

→Deploy the contract to Ganache using the following command:

**truffle migrate --network development**

→After deployment, copy the contract address from the terminal and update the contractAddress in app.js.

## OUTPUT:

```
C:\Users\IEJASWINI.K\DecentralizedGreeting>truffle migrate --network development
Compiling your contracts...
=====
> Compiling .\contracts\Greeting.sol
> Artifacts written to C:\Users\IEJASWINI.K\DecentralizedGreeting\build\contracts
> Compiled successfully using:
   solc: 0.8.0+commit.c7dfd78e.Emscripten.clang

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

2_deploy_contracts.js
=====
Replacing 'Greeting'
-----
> transaction hash:  0x3a748492c0592fcd0a65288749f0c9bc4f5005d5da4634ed178a9e98ca1d749
> Blocks: 0         Seconds: 0
> contract address: 0xd4C897eba114b27bd811625E2bd41CA8a95fEAC2
> block number:     4
> block timestamp:  1734496125
> account:          0x748031812d1786AE59a2D787bACab1b78265825B
> balance:          98.989362608
> gas used:         367118 (0x59a0e)
> gas price:        2 gwei
> value sent:       0 ETH
> total cost:       0.000734236 ETH

> Saving artifacts
-----
> Total cost:       0.000734236 ETH

Summary
-----
> Total deployments:  1
> Final cost:        0.000734236 ETH
```

## Run and Test the DApp:

### Run the DApp:

- Copy the ABI JSON from the build/contracts/Greeting.json file into app.js in the contractABI section.
- Replace <DEPLOYED\_CONTRACT\_ADDRESS> with the contract address displayed after deployment.
- Start a simple HTTP server to serve the frontend:  
npx http-server src

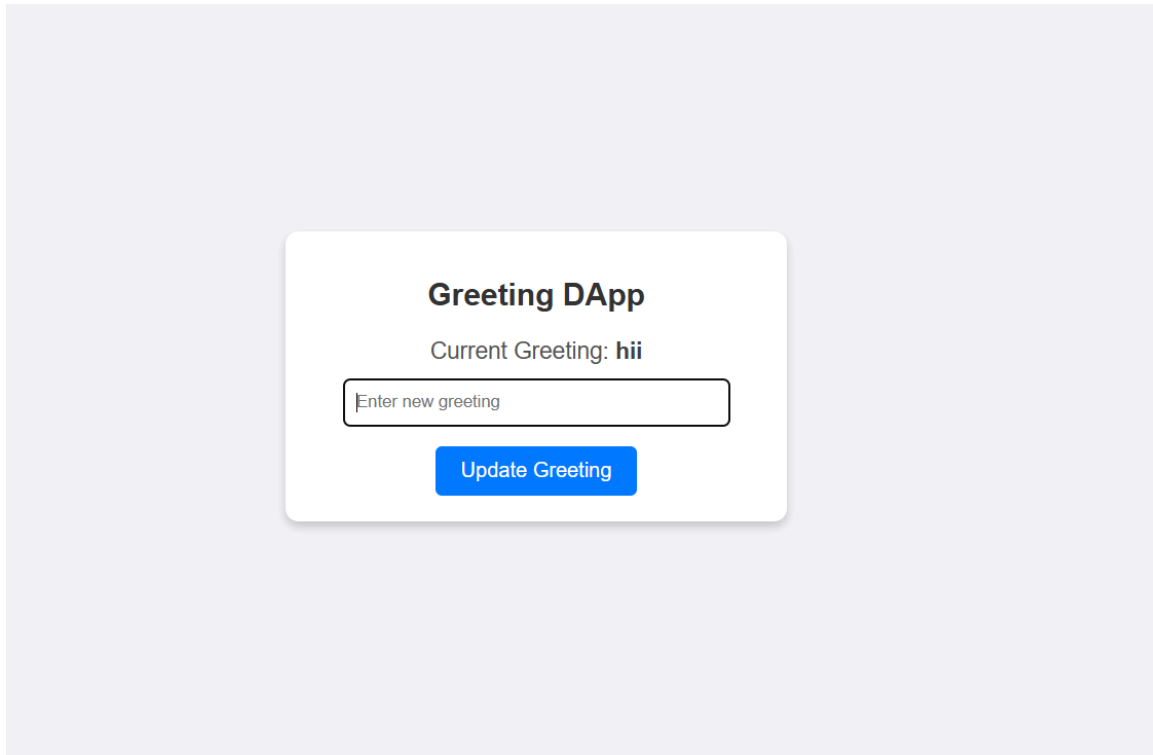
### Test the DApp:

- Open the **index.html** file in your browser.
- Ensure MetaMask is connected to your Ganache account.
- You should see the current greeting displayed. Enter a new greeting and click the button to update it.
- The greeting will be updated on the blockchain, and the page will reload to reflect the change.



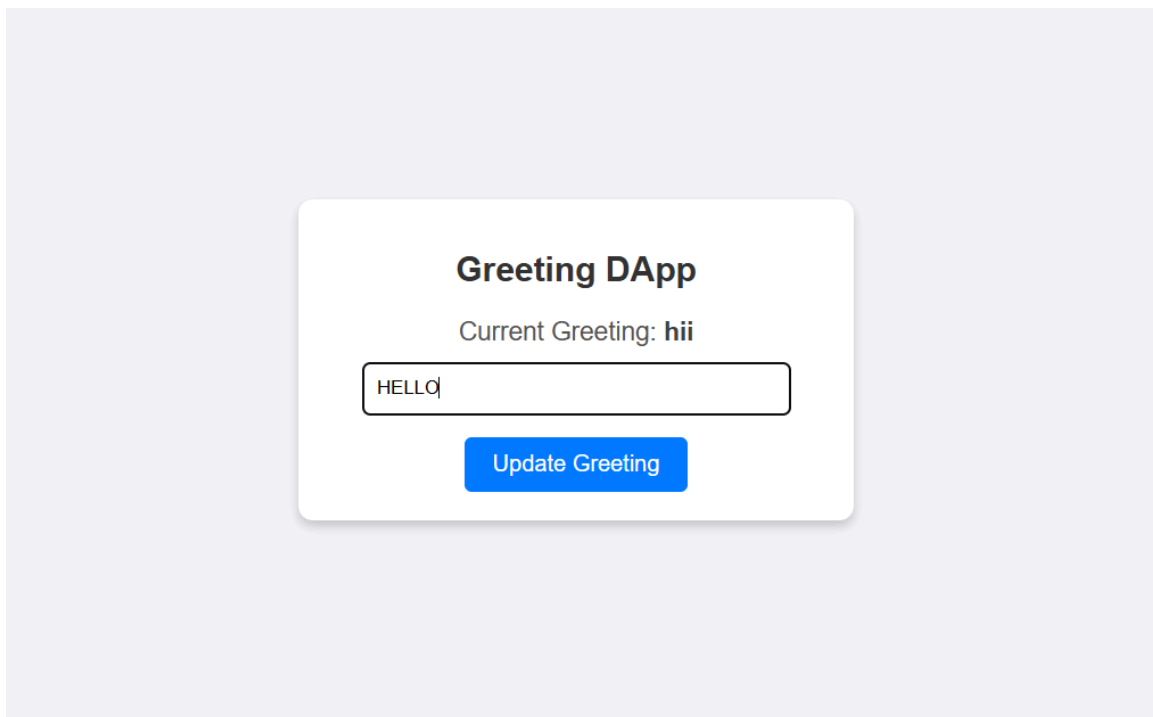
## OUTPUTS:

Add the new Greeting in the entry box:



The screenshot shows a web application titled "Greeting DApp". Below the title, it displays "Current Greeting: hii". There is a text input field with the placeholder text "Enter new greeting". Below the input field is a blue button labeled "Update Greeting". The entire interface is centered on a light gray background.

Click on Update Greeting the block will be added in the Ganache:



The screenshot shows the same "Greeting DApp" interface. The "Current Greeting" is still "hii". The text input field now contains the word "HELLO". The blue "Update Greeting" button remains below the input field. The interface is centered on a light gray background.

Click on update greeting.Then will be added in the Ganache.

BLOCK 14	MINED ON 2024-12-18 15:37:07	GAS USED 21508	1 TRANSACTION
BLOCK 13	MINED ON 2024-12-18 15:33:46	GAS USED 367142	1 TRANSACTION
BLOCK 12	MINED ON 2024-12-18 15:32:53	GAS USED 367142	1 TRANSACTION
BLOCK 11	MINED ON 2024-12-18 15:23:53	GAS USED 21532	1 TRANSACTION
BLOCK 10	MINED ON 2024-12-18 15:23:52	GAS USED 21532	1 TRANSACTION
BLOCK 9	MINED ON 2024-12-18 15:22:46	GAS USED 21532	1 TRANSACTION
BLOCK 8	MINED ON 2024-12-18 15:22:46	GAS USED 21532	1 TRANSACTION
BLOCK 7	MINED ON 2024-12-18 15:22:45	GAS USED 21532	1 TRANSACTION
BLOCK	MINED ON	GAS USED	1 TRANSACTION

METAMASK:


Add MetaMask in the chrome:



**MetaMask**

 [metamask.io](#) 2.9★ (4.6K ratings)

Add to Chrome



Add "MetaMask"?

It can:

Read and change all your data on all websites

Display notifications

Modify data you copy and paste

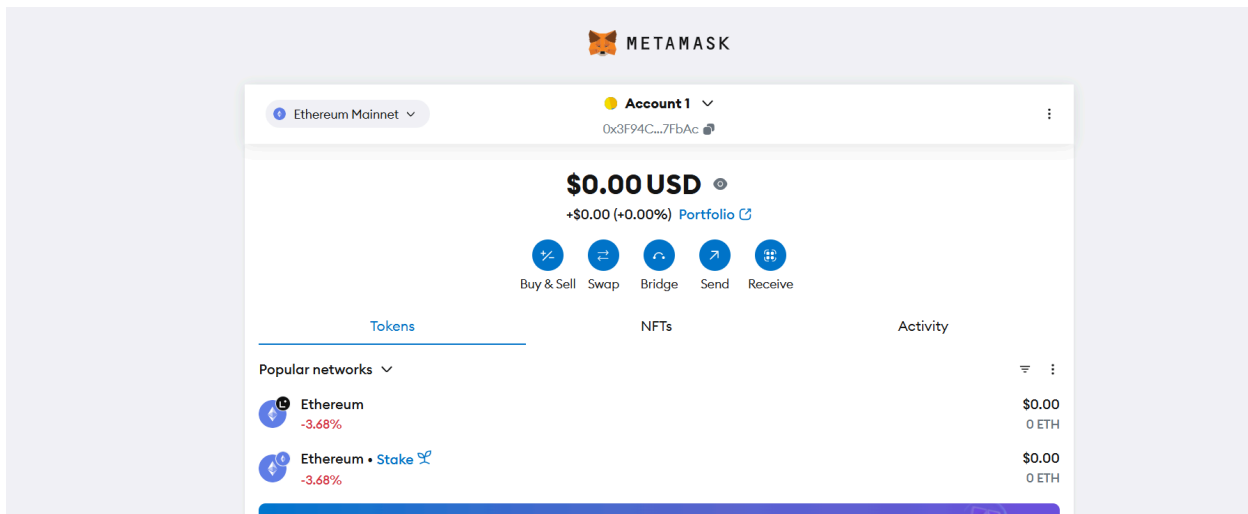
Add extension

Cancel

js)

6 000 000 users

The MetaMask frame will be displayed like this:



Add a custom network:

The screenshot shows the 'Add a custom network' form in the MetaMask mobile application. The form has a title bar with a back arrow, the title 'Add a custom network', and a close button. It contains five input fields: 'Network name' with a placeholder 'Enter network name', 'Default RPC URL' with a placeholder 'Add a URL' and a dropdown arrow, 'Chain ID' with a placeholder 'Enter Chain ID', 'Currency symbol' with a placeholder 'Enter symbol', and 'Block explorer URL' with a placeholder 'Add a URL' and a dropdown arrow. At the bottom of the form is a large blue button labeled 'Save'.

Add Ganache network and fill all the details, then click on save:

<

Add a custom network

×

Network name

Ganache

According to our records, the network name may not correctly match this chain ID.  
Suggested name: [Localhost 8545](#)

Default RPC URL

127.0.0.1:7545

Chain ID

1337

Currency symbol

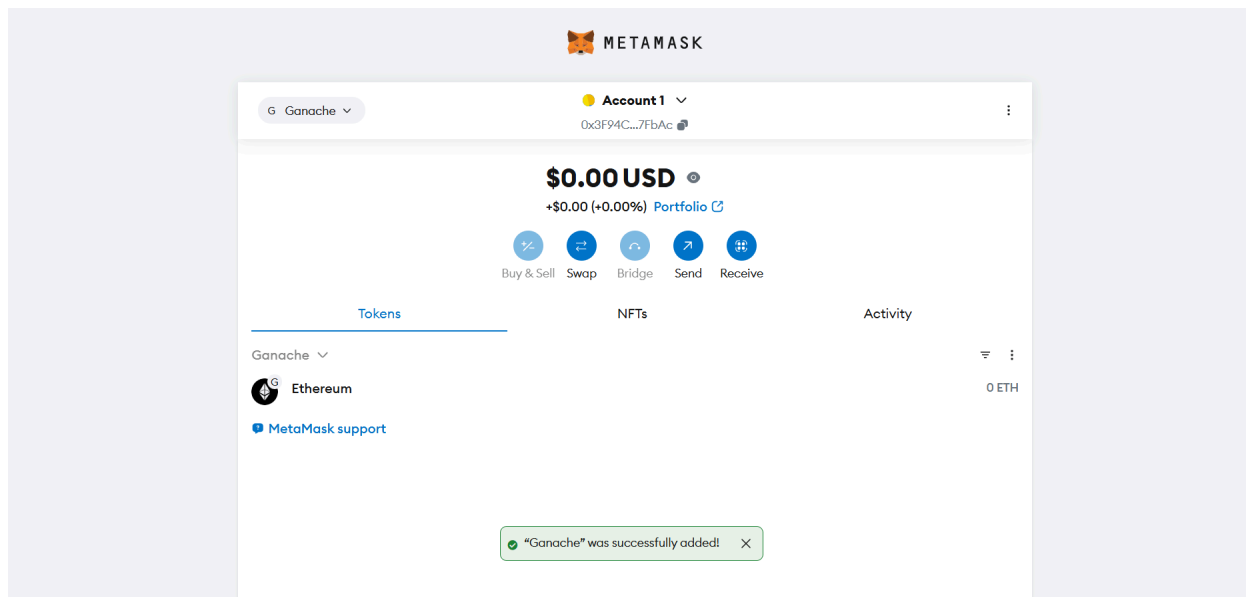
ETH

Block explorer URL

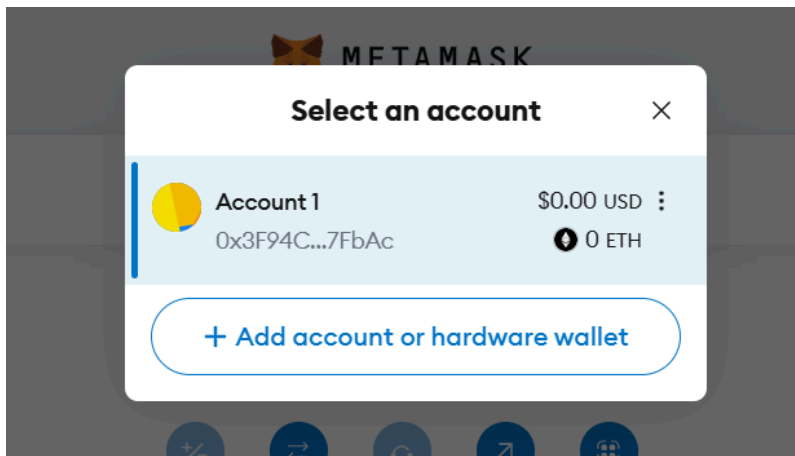
Add a URL

Save

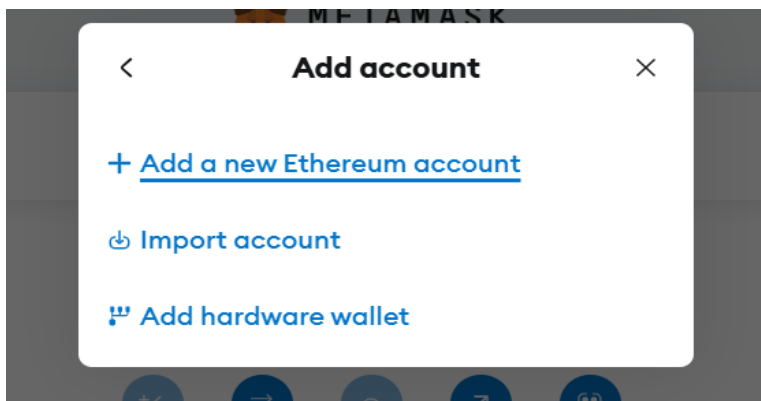
It open an Ganache platform:



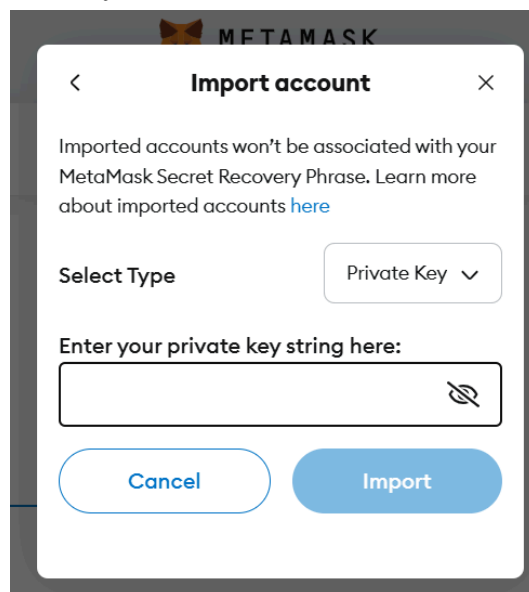
To connect with account in the Ganache select Account in Top of the page and click on Add account or hardware wallet.



Click on import account:



Add the account private key which is located in the Ganache:



Copy the private key and paste in the entry filed:


**ACCOUNT INFORMATION**

**ACCOUNT ADDRESS**  
0x763f024878Cc82AEC76C2eeD4dCcF74B9139607e

**PRIVATE KEY**  
0x51fd2f97469a9410ccda80977f7c7db35c9fc37eaf64653055e49f63b51d7c28  
**Do not use this private key on a public blockchain; use it for development purposes only!**

DONE


After entering the key click on import:

 METAMASK

< **Import account** ×

Imported accounts won't be associated with your MetaMask Secret Recovery Phrase. Learn more about imported accounts [here](#)


Select Type Private Key ▾


Enter your private key string here:  
.....| 



Cannot import invalid private key.






Cancel Import

After the account has been added the money will be displayed:

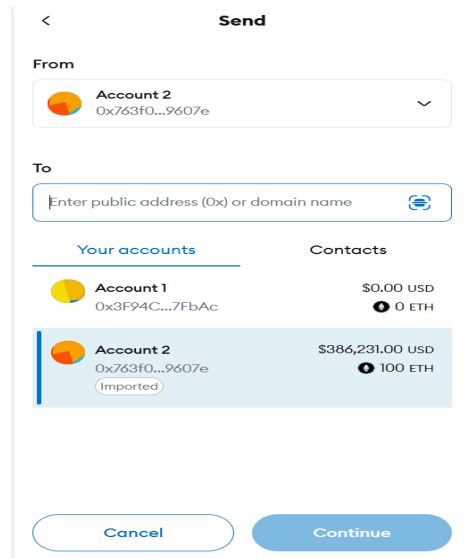
 METAMASK

G Ganache ▾ Account 2 ▾  
0x763f0...9607e 

**\$386,231.00 USD**   
+\$0.00 (+0.00%) [Portfolio](#) 

      
Buy & Sell Swap Bridge Send Receive

To send money to some one click on send and enter the receiver public key:



Copy the public key from the Ganache:

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

WORKSPACE QUICKSTART

SAVE

SWITCH

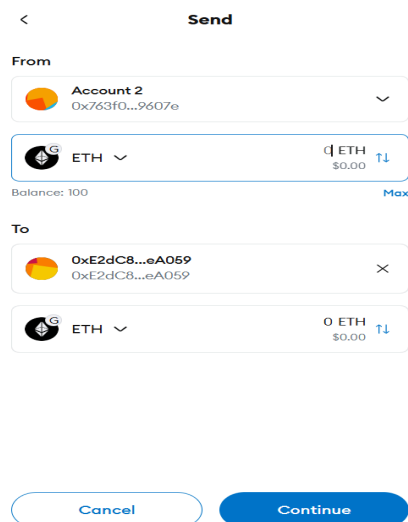
MNEMONIC

hurry banana hunt still blue edge civil make walk pioneer vessel beauty

HD PATH  
m44'60'0'0account\_index

ADDRESS	BALANCE	TX COUNT	INDEX	
0x28c203924599Bf027dA28dDe49915A593cA3d914	99.00 ETH	15	0	
ADDRESS	BALANCE	TX COUNT	INDEX	
0xE551770e71aB354a735cCa013CFC4cbeEa3AB834	100.99 ETH	1	1	
ADDRESS	BALANCE	TX COUNT	INDEX	
0xE2dC85F0DB6f89bF6f5286ef81136ba0E16eA059	100.00 ETH	0	2	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x763f024878Cc82AEC76C2eeD4dCcF74B9139607e	100.00 ETH	0	3	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x500B513c6475671646e6d429e8dB916A0AF7e47d	100.00 ETH	0	4	

After entering public key the page will be in the way:



Now enter how much amount we need to send and click on continue and confirm:

<

Send

From

Account 2

0x763f0...9607e

ETH

40 ETH

\$154,075.60

↑↓

Balance: 100

Max

To

0xE2dC8...eA059

0xE2dC8...eA059

ETH

40 ETH

\$154,075.60

↑↓

Cancel

Continue

The amount will be transferred and money will be credited to the receiver:

METAMASK

Ganache

Account 2

0x763f0...9607e

\$231,226.38 USD

+\$0.00 (+0.00%) Portfolio

Buy & Sell

Swap

Bridge

Send

Receive

Tokens

NFTs

Activity

Dec 18, 2024

Send

Confirmed

-40 ETH

-\$154,152.00 USD

Now check the Ganache the money will be added:

ADDRESS	BALANCE	TX COUNT	INDEX	
0xE2dC85F0DB6f89bF6f5286ef81136ba0E16eA059	140.00 ETH	0	2	
0x763f024878Cc82AEC76C2eeD4dCcF74B9139607e	60.00 ETH	1	3	