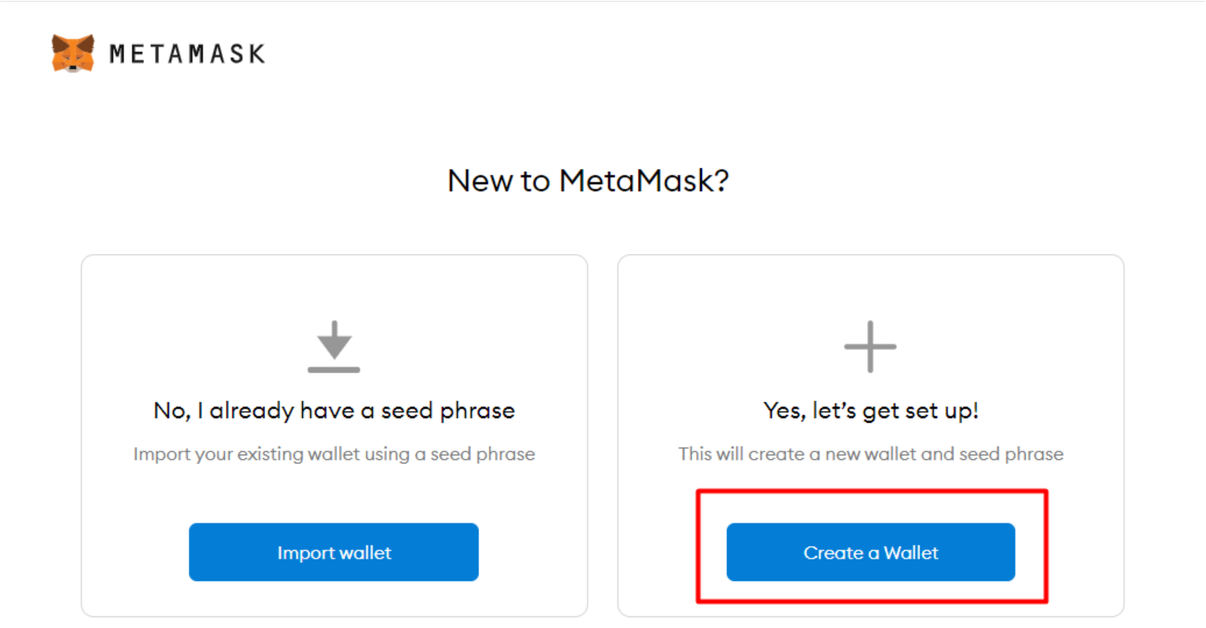


METAMASK

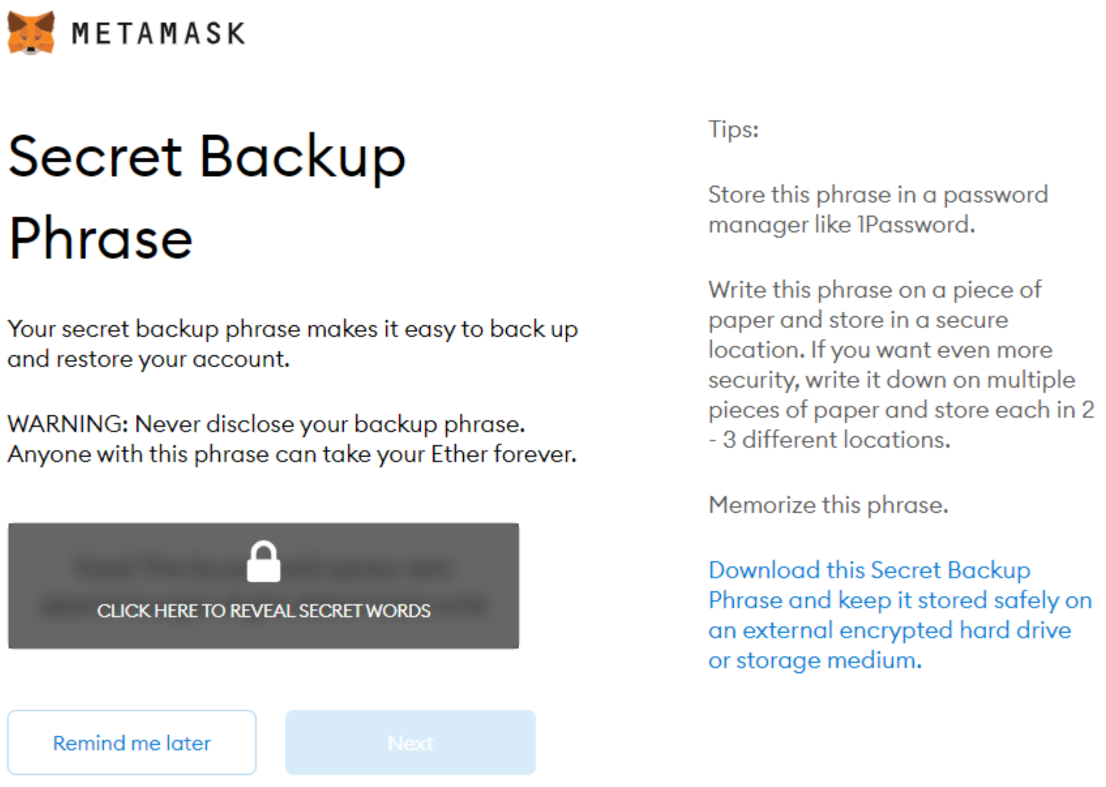
New to MetaMask?

No, I already have a seed phrase

Import your existing wallet using a seed phrase

Yes, let's get set up!

This will create a new wallet and seed phrase



METAMASK

Secret Backup Phrase

Your secret backup phrase makes it easy to back up and restore your account.

WARNING: Never disclose your backup phrase. Anyone with this phrase can take your Ether forever.

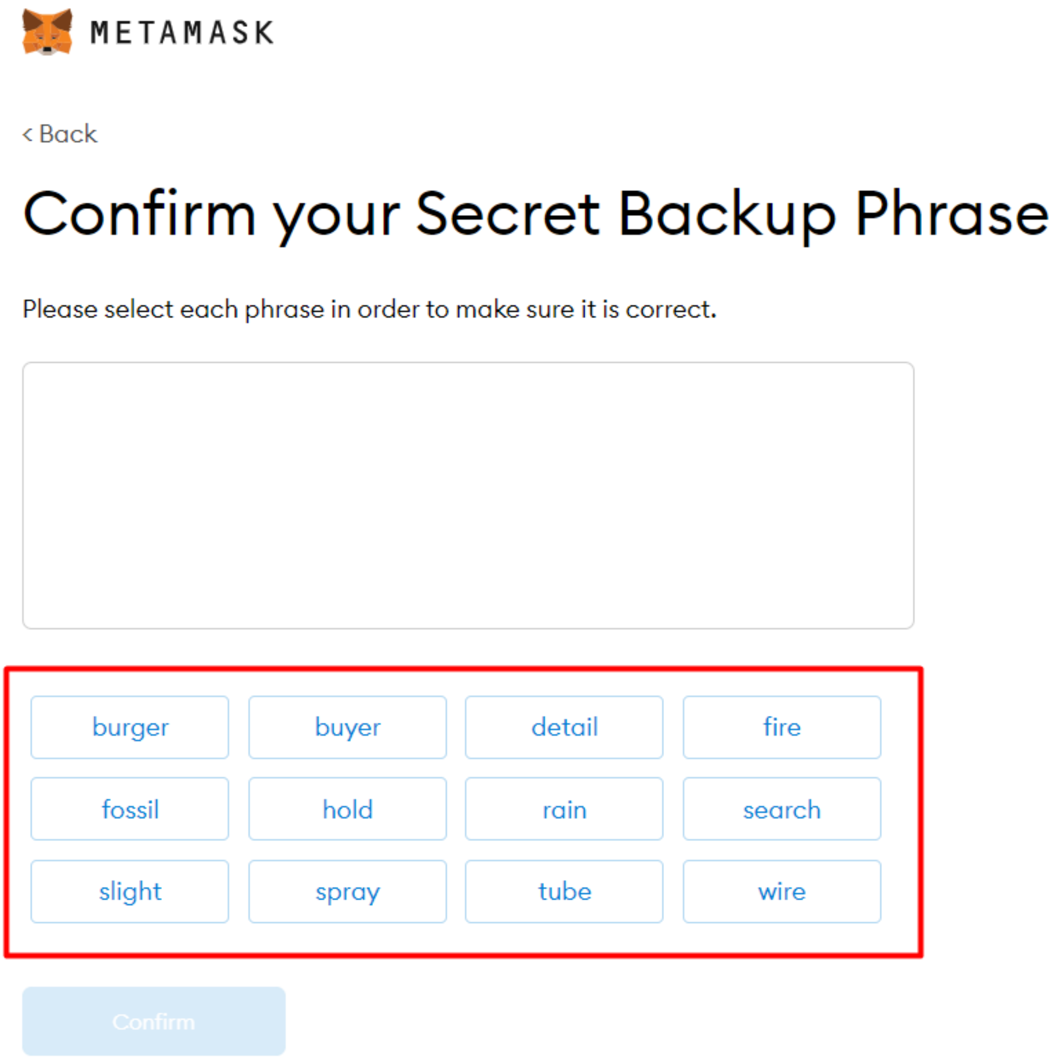
Tips:

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 Backup Phrase and keep it stored safely on an external encrypted hard drive or storage medium.

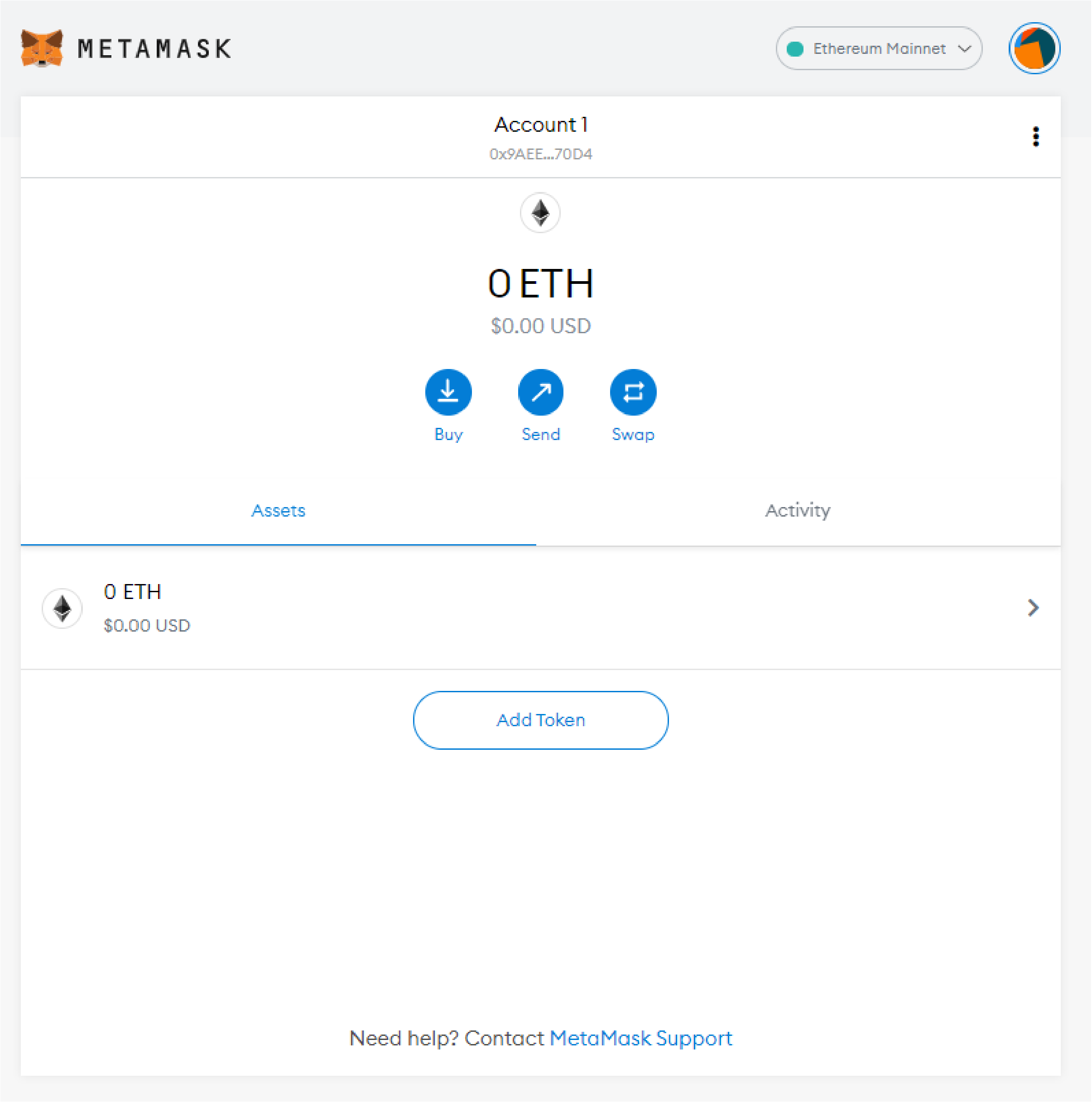


METAMASK

Confirm your Secret Backup Phrase

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

burger buyer detail fire
fossil hold rain search
slight spray tube wire



METAMASK

Ethereum Mainnet

Account 1

0x9AEE...70D4

0 ETH

$0.00 USD

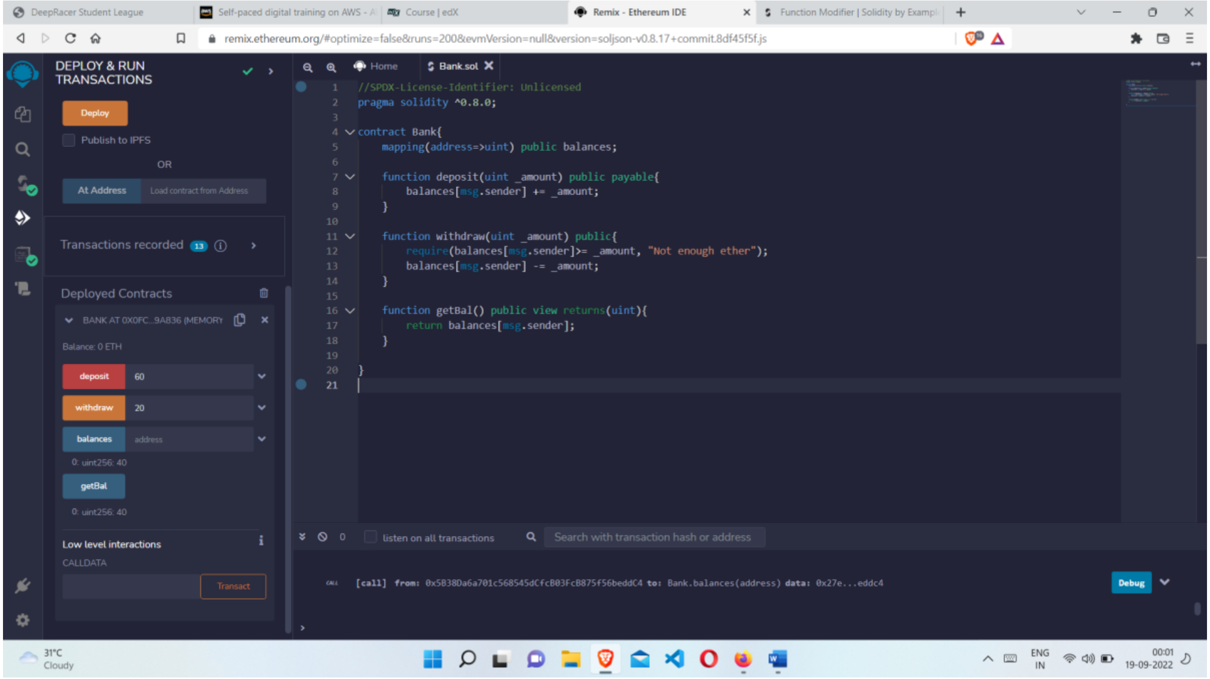
0 ETH

$0.00 USD

Need help? Contact MetaMask Support

Bank smart contract

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

contract Bank{
    mapping(address=>uint) public balances;

    function deposit(uint _amount) public payable{
        balances[msg.sender] += _amount;
    }

    function withdraw(uint _amount) public{
        require(balances[msg.sender]>= _amount, "Not enough ether");
        balances[msg.sender] -= _amount;
    }

    function getBal() public view returns(uint){
        return balances[msg.sender];
    }

}
```


DeepRacer Student League
Self-paced digital training on AWS - A
Course | edX
Remix - Ethereum IDE
Function Modifier | Solidity by Exampl
remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.17+commit.8df45f5f.js
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 13
Deployed Contracts
Home
Bank.sol
listen on all transactions
Search with transaction hash or address
Debug
31°C
Cloudy
ENG
IN
00:01
19-09-2022

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

contract Bank{
    mapping(address=>uint) public balances;

    function deposit(uint _amount) public payable{
        balances[msg.sender] += _amount;
    }

    function withdraw(uint _amount) public{
        require(balances[msg.sender]>= _amount, "Not enough ether");
        balances[msg.sender] -= _amount;
    }

    function getBal() public view returns(uint){
        return balances[msg.sender];
    }

}
```

BANK AT 0X0FC...9A836 (MEMORY

Balance: 0 ETH

deposit 60

withdraw 20

balances address

0: uint256: 40

getBal

0: uint256: 40

Low level interactions

CALLDATA

Transact

```
CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Bank.balances(address) data: 0x27e...eddc4
```

Student.sol

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

contract Database{

    struct student{
        int ID;
        string _fname;
        string _lname;
    }

    int public Count = 0;

    mapping(int => student) public stdRecords;

    function addNew(int _id,
                    string calldata _fname,
                    string calldata _lname) public{
                        Count+=1;
                        stdRecords[Count] = student(_id, _fname, _lname);
                    }

    event received(address user, uint amount);

    receive() external payable{
        emit received(msg.sender, msg.value);
    }

}
```