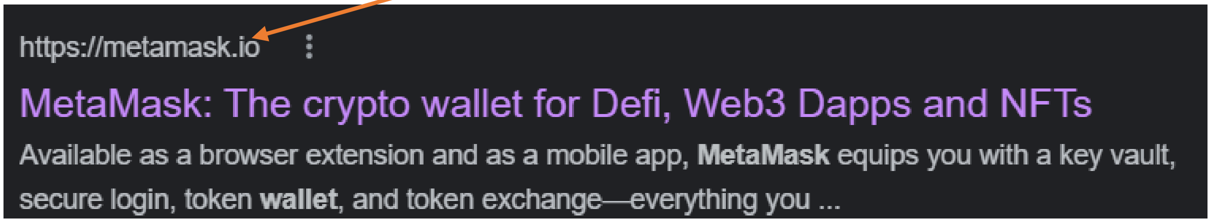
Installation of MetaMask and study spending Ether per transaction.

Step 1 : Go to https://metamask.io

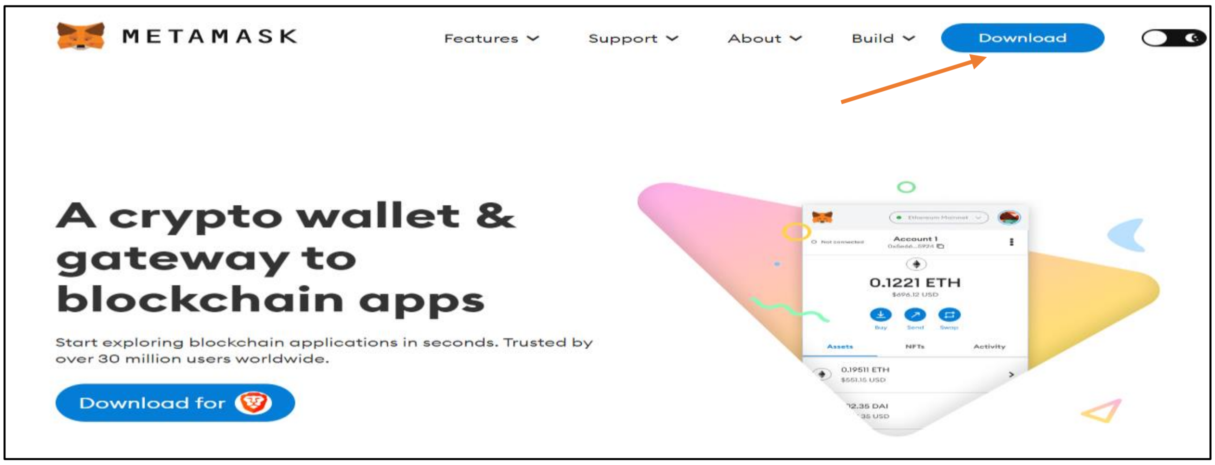


https://metamask.io

MetaMask: The crypto wallet for Defi, Web3 Dapps and NFTs

Available as a browser extension and as a mobile app, **MetaMask** equips you with a key vault, secure login, token **wallet**, and token exchange—everything you ...

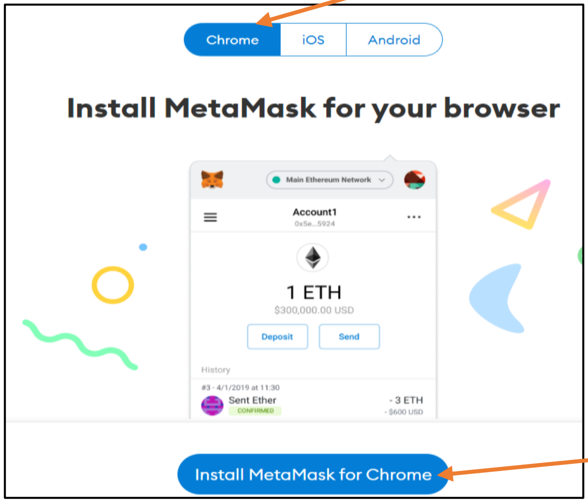
Step 2 : Click on Download



A crypto wallet & gateway to blockchain apps

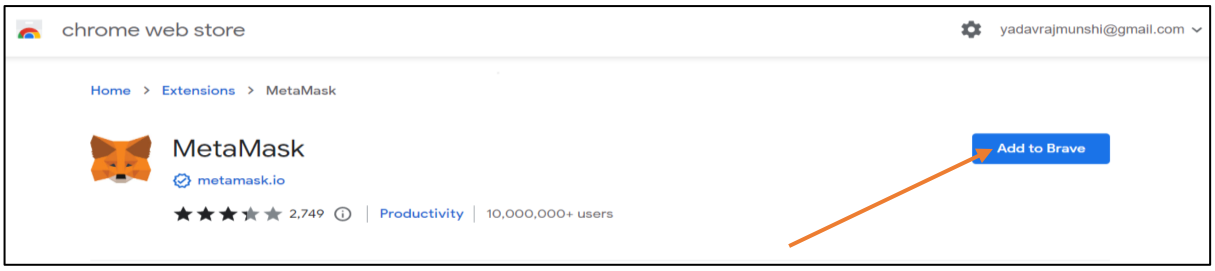
Start exploring blockchain applications in seconds. Trusted by over 30 million users worldwide.

Step 3 : Choose the OS and click on "Install MetaMask for Chrome"



Install MetaMask for your browser

Step 4: Click on "Add to Brave" / "Add to Chrome"



chrome web store

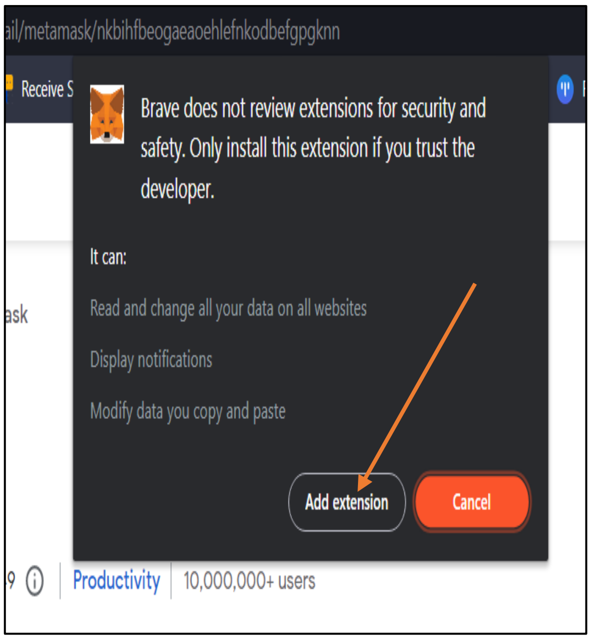
Home > Extensions > MetaMask

MetaMask

metamask.io

★★★★★ 2,749 | Productivity | 10,000,000+ users

Step 5 : Click on Add extension the confirm the installation


ail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn

Brave does not review extensions for security and safety. Only install this extension if you trust the developer.

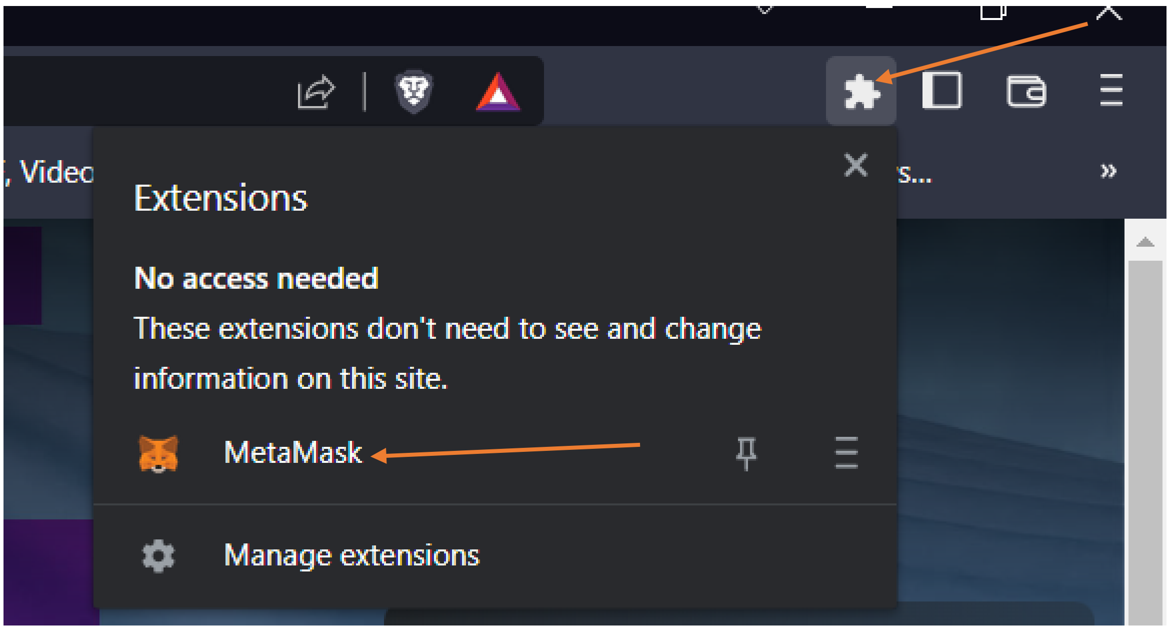
It can:

Read and change all your data on all websites

Display notifications

Modify data you copy and paste

Step 6 : You have successfully installed MetaMask Wallet and can be viewed in Extensions Tab of the browser.

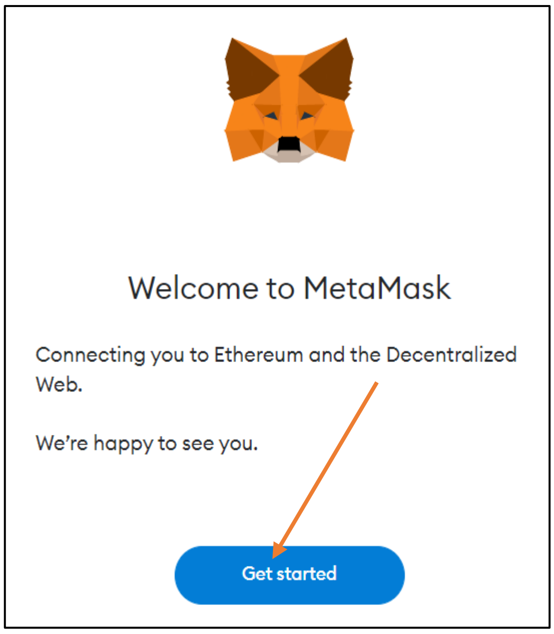

Extensions

No access needed

These extensions don't need to see and change information on this site.

Create your own wallet using Metamask for crypto transactions.

Step 1: Click on Get Started.

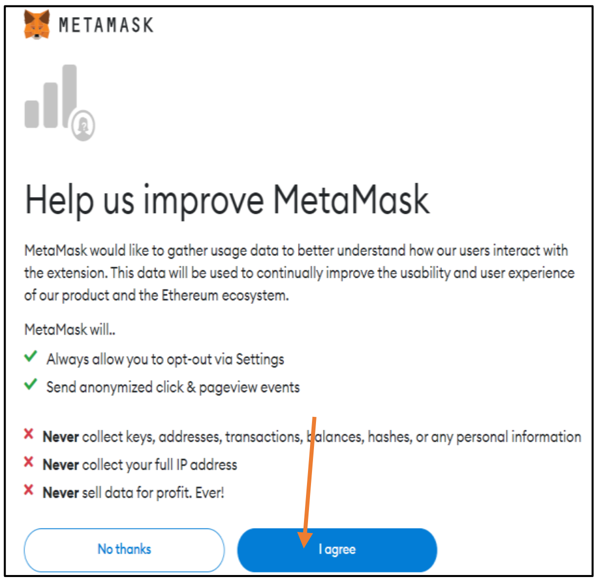


Welcome to MetaMask

Connecting you to Ethereum and the Decentralized Web.

We're happy to see you.

Step 2: Click on "I agree" after reading the instructions. After that click on "Create a Wallet".

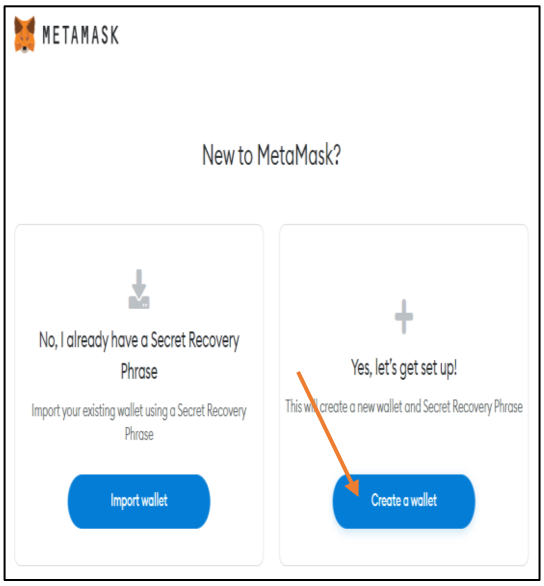


Help us improve MetaMask

MetaMask would like to gather usage data to better understand how our users interact with the extension. This data will be used to continually improve the usability and user experience of our product and the Ethereum ecosystem.

MetaMask will..

- ✓ Always allow you to opt-out via Settings
- ✓ Send anonymized click & pageview events
- ✗ **Never** collect keys, addresses, transactions, balances, hashes, or any personal information
- ✗ **Never** collect your full IP address
- ✗ **Never** sell data for profit. Ever!



New to MetaMask?

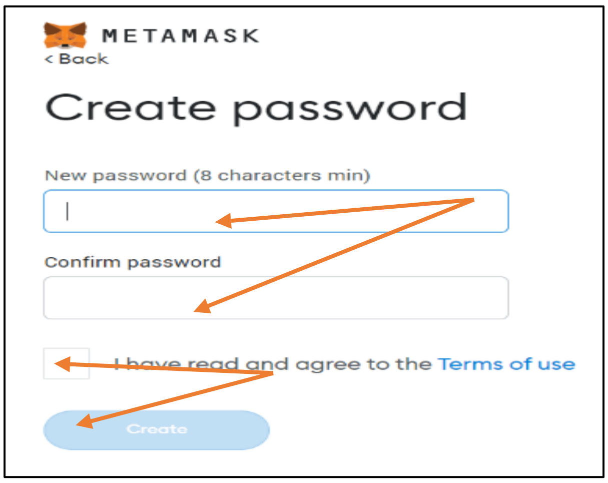
No, I already have a Secret Recovery Phrase

Import your existing wallet using a Secret Recovery Phrase

Yes, let's get set up!

This will create a new wallet and Secret Recovery Phrase

Step 3: Enter a "Strong Password". Agree terms and conditions and then follow instructions and click "Next".

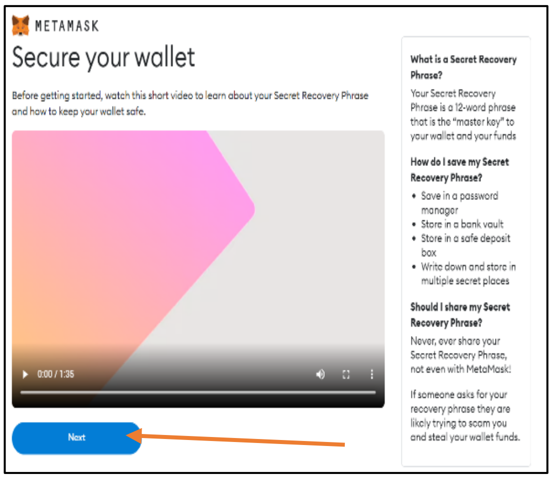


Create password

New password (8 characters min)

Confirm password

I have read and agree to the Terms of use



Secure your wallet

Before getting started, watch this short video to learn about your Secret Recovery Phrase and how to keep your wallet safe.

What is a Secret Recovery Phrase?

Your Secret Recovery Phrase is a 12-word phrase that is the "master key" to your wallet and your funds

How do I save my Secret Recovery Phrase?

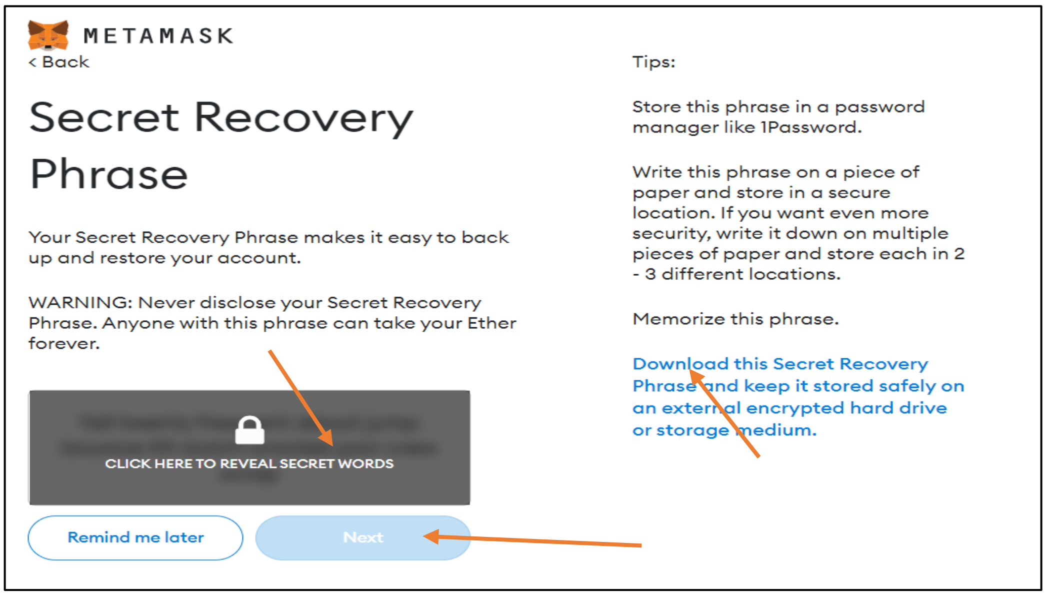
- Save in a password manager
- Store in a bank vault
- Store in a safe deposit box
- Write down and store in multiple secret places

Should I share my Secret Recovery Phrase?

Never, ever share your Secret Recovery Phrase, not even with MetaMask!

If someone asks for your recovery phrase they are likely trying to scam you and steal your wallet funds.

Step 4: Click on "CLICK HERE TO REVEAL SECRET WORDS" and Download the secret and key and make a note of the same in some written form and make a copy of it and save it securely and re-enter the phrase in next step.



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.

CLICK HERE TO REVEAL SECRET WORDS

Remind me later

Next

Tips:

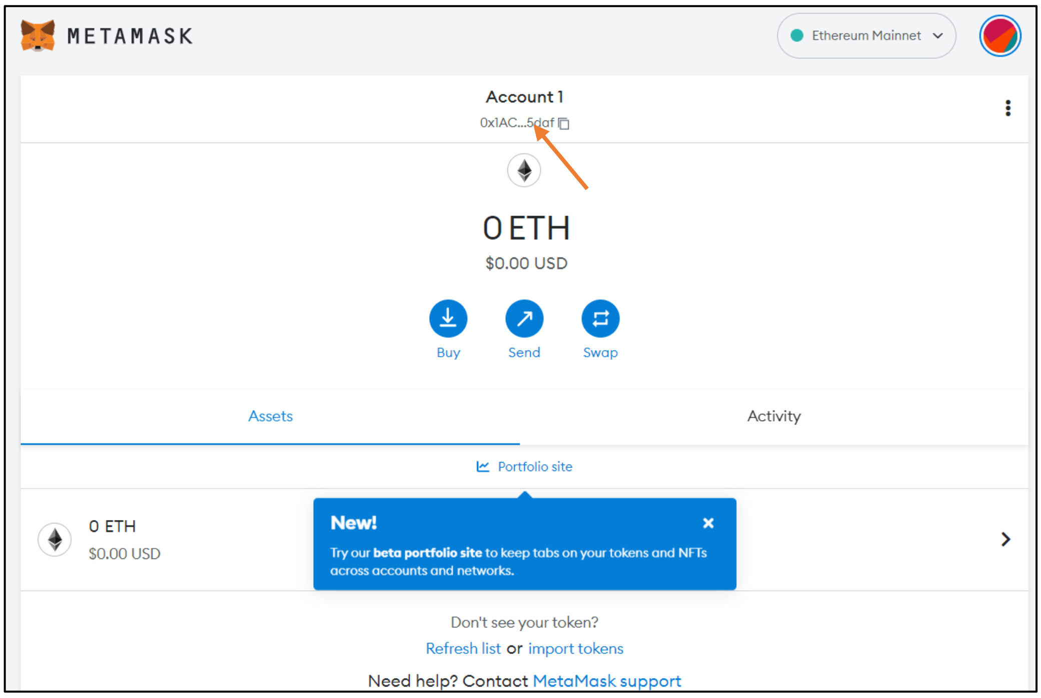
Store this phrase in a password manager like 1Password.

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 5: You have successfully created a Metamask Wallet.



METAMASK

Ethereum Mainnet

Account 1

0x1AC...5daf

0 ETH

$0.00 USD

Buy Send Swap

Assets Activity

Portfolio site

0 ETH

$0.00 USD

New!

Try our **beta portfolio site** to keep tabs on your tokens and NFTs across accounts and networks.

Don't see your token?

Refresh list or import tokens

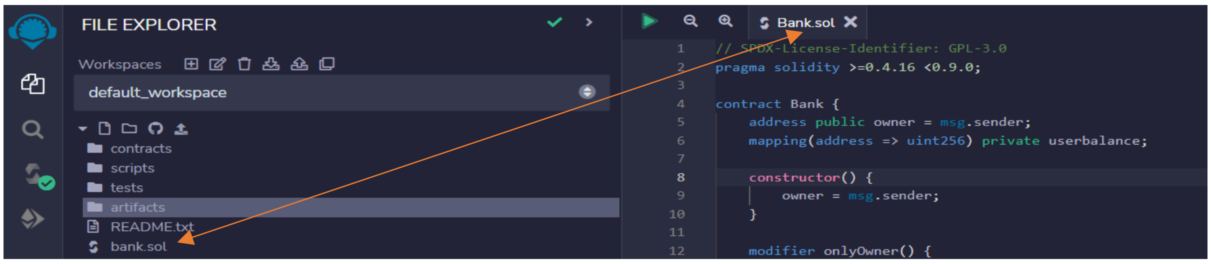
Need help? Contact MetaMask support

Blockchain Technology

Write a smart contract on a test network for Bank account of a customer for following operations:

• Deposit money • Withdraw Money • Show balance

Step 1 : Create bank.sol in remix editor.

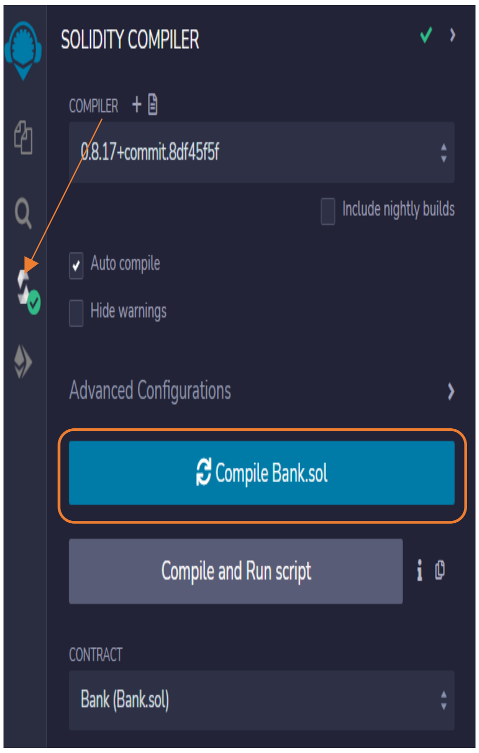


```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16 <0.9.0;

contract Bank {
    address public owner = msg.sender;
    mapping(address => uint256) private userbalance;

    constructor() {
        owner = msg.sender;
    }

    modifier onlyOwner() {
```

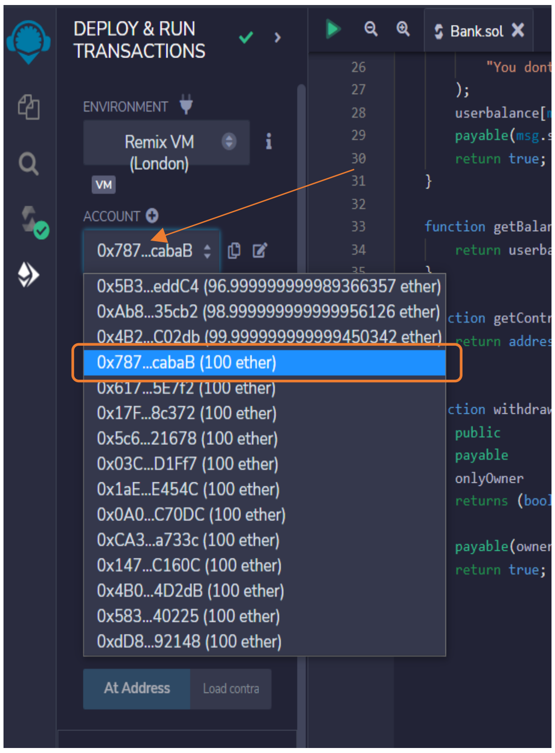
```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16 <0.9.0;
contract Bank {
    address public owner = msg.sender;
    mapping(address => uint256) private userbalance;
    constructor() {
        owner = msg.sender;
    }
    modifier onlyOwner() {
        require(msg.sender == owner, "You are not the owner of this contract");
        _;
    }
    function deposit() public payable returns (bool) {
        require(msg.value > 10 wei, "Please Deposit at least 10 wei");
        userbalance[msg.sender] += msg.value;
        return true;
    }

    function withdraw(uint256 _amount) public payable returns (bool) {
        require(
            _amount <= userbalance[msg.sender],
            "You dont have sufficient funds!"
        );
        userbalance[msg.sender] -= _amount;
        payable(msg.sender).transfer(_amount);
        return true;
    }

    function getBalance() public view returns (uint256) {
        return userbalance[msg.sender];
    }

    function getContractBalance() public view onlyOwner returns (uint256) {
        return address(this).balance;
    }

    function withdrawfunds(uint256 _amount)
        public
        payable
        onlyOwner
        returns (bool)
    {
        payable(owner).transfer(_amount);
        return true;
    }
}
```

Blockchain Technology

Step 2 : Compile the contract. Choose Owner account address of the contract and click on “Delpoy”.


SOLIDITY COMPILER

COMPILER
0.8.17+commit.8df45f5f
Include nightly builds
✓ Auto compile
Hide warnings
Advanced Configurations
Compile Bank.sol
Compile and Run script
CONTRACT
Bank (Bank.sol)


DEPLOY & RUN TRANSACTIONS

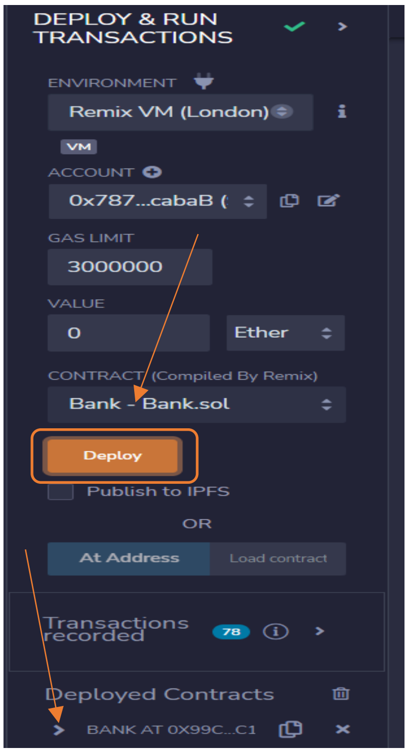
ENVIRONMENT
Remix VM (London)
VM
ACCOUNT
0x787...cabaB

```
0x5B3...eddC4 (96.999999999989366357 ether)
0xAb8...35cb2 (98.999999999999956126 ether)
0x4B2...C02db (99.999999999999450342 ether)
0x787...cabaB (100 ether)
0x617...5E7f2 (100 ether)
0x17F...8c372 (100 ether)
0x5c6...21678 (100 ether)
0x03C...D1Ff7 (100 ether)
0x1aE...E454C (100 ether)
0x0A0...C70DC (100 ether)
0xCA3...a733c (100 ether)
0x147...C160C (100 ether)
0x4B0...4D2dB (100 ether)
0x583...40225 (100 ether)
0xdD8...92148 (100 ether)
```

At Address   Load contra

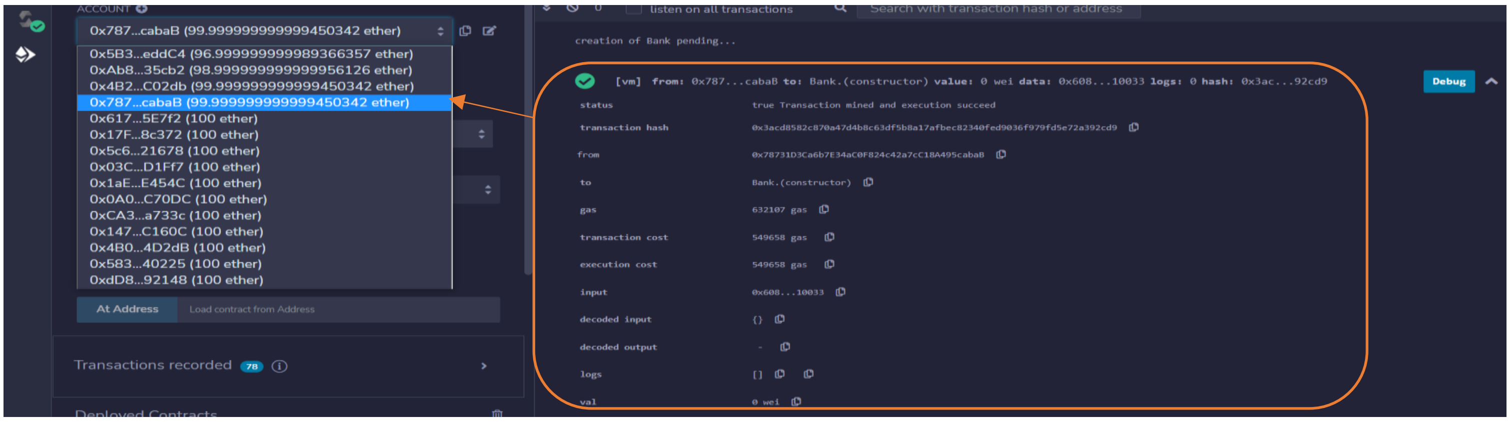
Bank.sol

```
            "You dont
        );
        userbalance[
        payable(msg.s
        return true;
    }

    function getBalan
        return userba
    }
ction getContr
return addres

ction withdraw
public
payable
onlyOwner
returns (bool

payable(owner
return true;
```


DEPLOY & RUN TRANSACTIONS

ENVIRONMENT
Remix VM (London)
VM
ACCOUNT
0x787...cabaB (
GAS LIMIT
3000000
VALUE
0   Ether
CONTRACT (Compiled By Remix)
Bank - Bank.sol
Deploy
Publish to IPFS
OR
At Address   Load contract
Transactions recorded 78
Deployed Contracts
BANK AT 0X99C...C1

Step 3: You can view Owner account address log for deploying the contract.


ACCOUNT
0x787...cabaB (99.999999999999450342 ether)

```
0x5B3...eddC4 (96.999999999989366357 ether)
0xAb8...35cb2 (98.999999999999956126 ether)
0x4B2...C02db (99.999999999999450342 ether)
0x787...cabaB (99.999999999999450342 ether)
0x617...5E7f2 (100 ether)
0x17F...8c372 (100 ether)
0x5c6...21678 (100 ether)
0x03C...D1Ff7 (100 ether)
0x1aE...E454C (100 ether)
0x0A0...C70DC (100 ether)
0xCA3...a733c (100 ether)
0x147...C160C (100 ether)
0x4B0...4D2dB (100 ether)
0x583...40225 (100 ether)
0xdD8...92148 (100 ether)
```

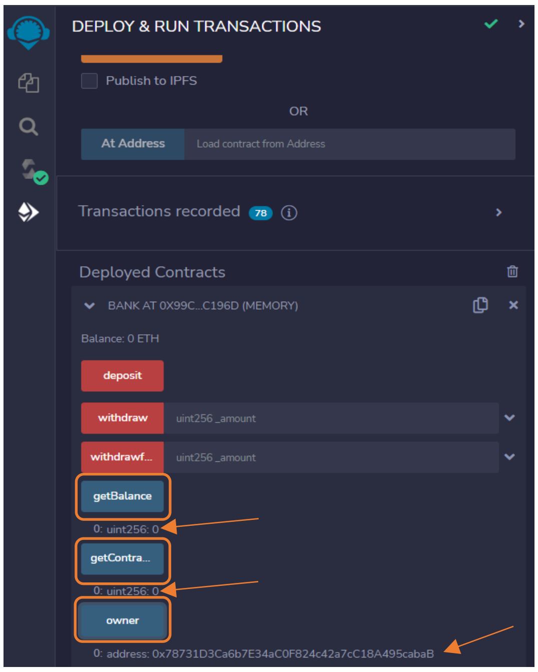
At Address   Load contract from Address
Transactions recorded 78
Deployed Contracts

listen on all transactions   Search with transaction hash or address

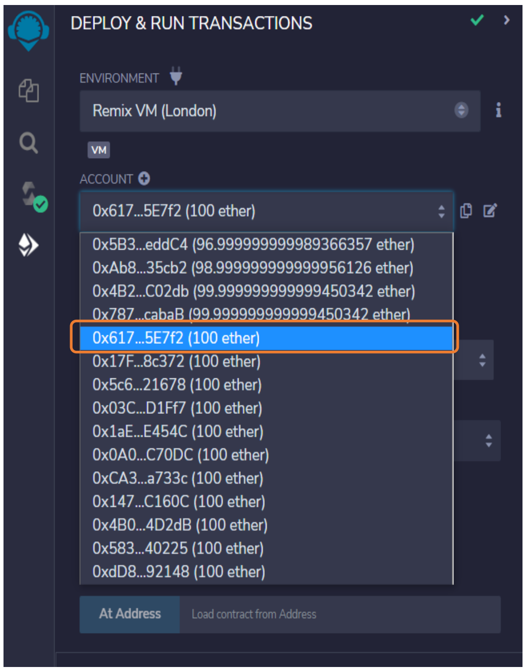
```
creation of Bank pending...

[vm]  from: 0x787...cabaB to: Bank.(constructor) value: 0 wei data: 0x608...10033 logs: 0 hash: 0x3ac...92cd9
status                  true Transaction mined and execution succeed
transaction hash        0x3acd8582c870a47d4b8c63df5b8a17afbec82340fed9036f979fd5e72a392cd9
from                    0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB
to                      Bank.(constructor)
gas                     632107 gas
transaction cost        549658 gas
execution cost          549658 gas
input                   0x608...10033
decoded input           {}
decoded output           -
logs                    []
val                     0 wei
```

Debug

Step 4 : Click on “getBalance”, “getContractBalance”, “Owner” to get the balance, contract balance and owner account address. Choose other account address and send ETH to Smart Contract address.


DEPLOY & RUN TRANSACTIONS

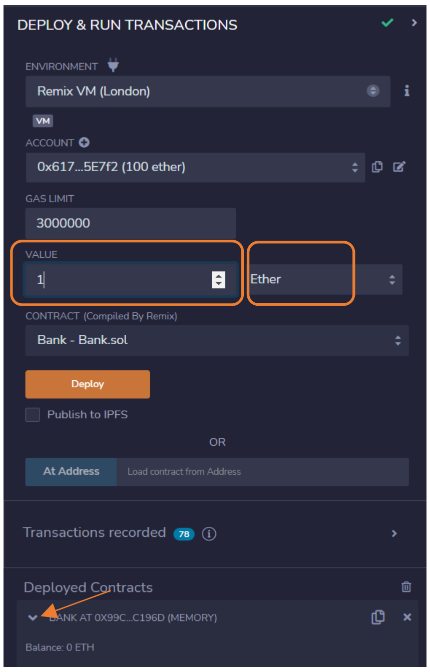
Publish to IPFS
OR
At Address   Load contract from Address
Transactions recorded 78
Deployed Contracts
BANK AT 0X99C...C196D (MEMORY)
Balance: 0 ETH
deposit
withdraw   uint256 _amount
withdrawf...   uint256 _amount
getBalance
0: uint256: 0
getContra...
0: uint256: 0
owner
0: address: 0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB


DEPLOY & RUN TRANSACTIONS

ENVIRONMENT
Remix VM (London)
VM
ACCOUNT
0x617...5E7f2 (100 ether)

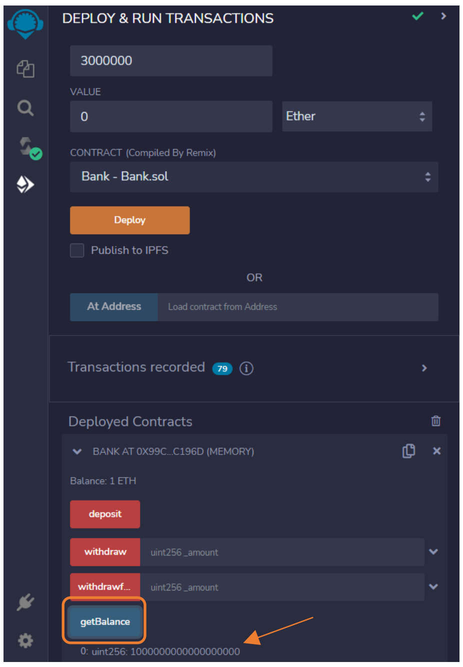
```
0x5B3...eddC4 (96.999999999989366357 ether)
0xAb8...35cb2 (98.999999999999956126 ether)
0x4B2...C02db (99.999999999999450342 ether)
0x787...cabaB (99.999999999999450342 ether)
0x617...5E7f2 (100 ether)
0x17F...8c372 (100 ether)
0x5c6...21678 (100 ether)
0x03C...D1Ff7 (100 ether)
0x1aE...E454C (100 ether)
0x0A0...C70DC (100 ether)
0xCA3...a733c (100 ether)
0x147...C160C (100 ether)
0x4B0...4D2dB (100 ether)
0x583...40225 (100 ether)
0xdD8...92148 (100 ether)
```

At Address   Load contract from Address


DEPLOY & RUN TRANSACTIONS

ENVIRONMENT
Remix VM (London)
VM
ACCOUNT
0x617...5E7f2 (100 ether)
GAS LIMIT
3000000
VALUE
1   Ether
CONTRACT (Compiled By Remix)
Bank - Bank.sol
Deploy
Publish to IPFS
OR
At Address   Load contract from Address
Transactions recorded 78
Deployed Contracts
BANK AT 0X99C...C196D (MEMORY)
Balance: 0 ETH

Step 5: Click on getBalance to check whether amount of ETH is deposited in Smart Contract or not. You will see the output in wei unit (1 ETH=10^{18} wei).Check withdrawal functionality by entering withdrawal amount and and again checking the balance.


DEPLOY & RUN TRANSACTIONS

3000000

VALUE: 0 Ether

CONTRACT (Compiled By Remix): Bank - Bank.sol

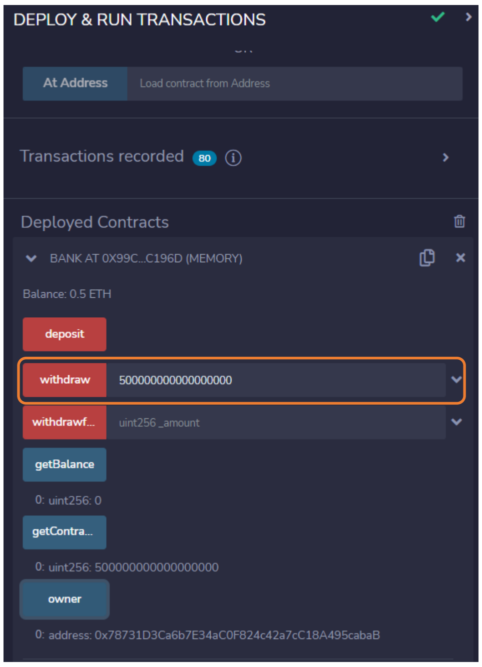
Transactions recorded 79

Deployed Contracts

BANK AT 0X99C...C196D (MEMORY)

Balance: 1 ETH

```
0: uint256: 1000000000000000000
```


DEPLOY & RUN TRANSACTIONS

Transactions recorded 80

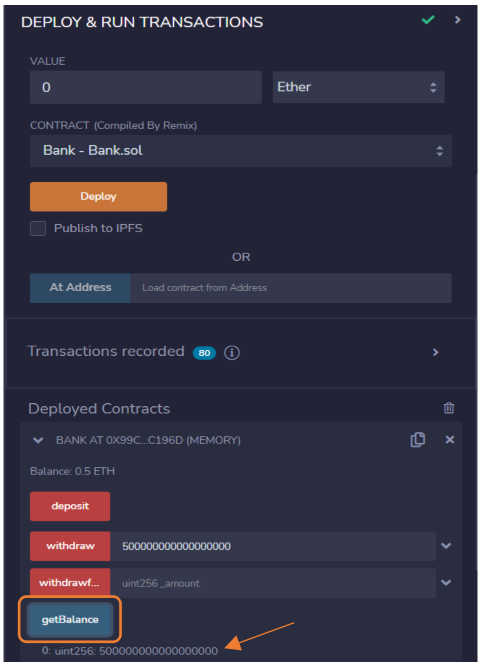
Deployed Contracts

BANK AT 0X99C...C196D (MEMORY)

Balance: 0.5 ETH

withdraw: 500000000000000000

```
0: uint256: 0
0: uint256: 500000000000000000
0: address: 0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB
```


DEPLOY & RUN TRANSACTIONS

VALUE: 0 Ether

CONTRACT (Compiled By Remix): Bank - Bank.sol

Transactions recorded 80

Deployed Contracts

BANK AT 0X99C...C196D (MEMORY)

Balance: 0.5 ETH

withdraw: 500000000000000000

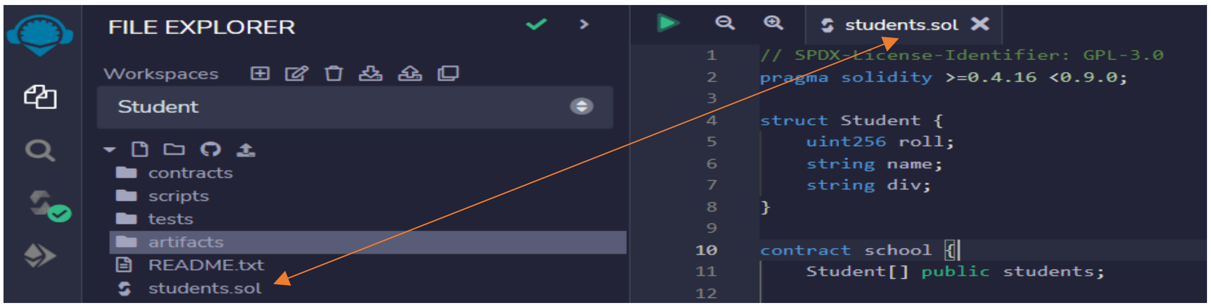
```
0: uint256: 500000000000000000
```

Write a program in solidity to create Student data. Use the following constructs:

- Structures
- Arrays
- Fallback

Deploy this as smart contract on Ethereum and Observe the transaction fee and Gas values.

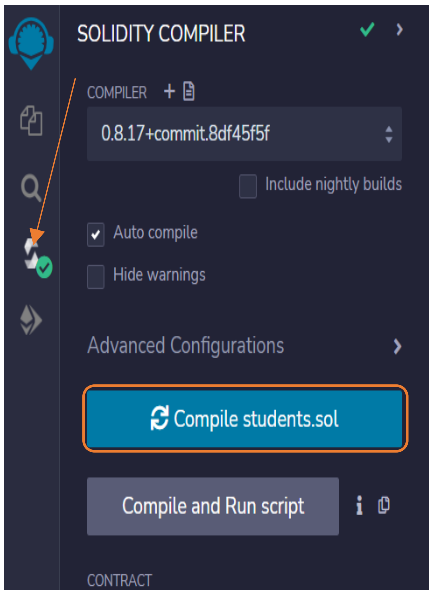
Step 1 : Create “students.sol in remix editor.



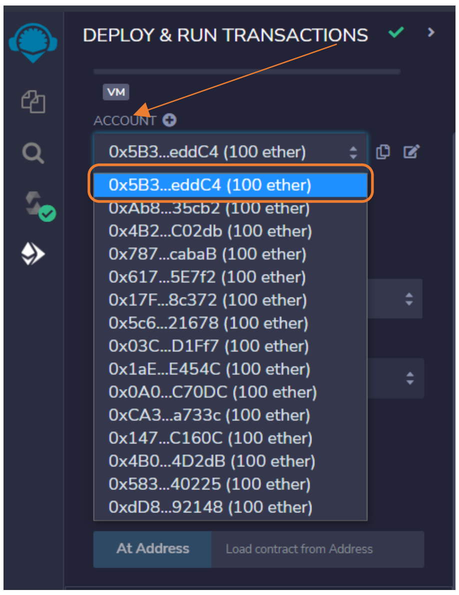
```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16 <0.9.0;

struct Student {
    uint256 roll;
    string name;
    string div;
}

contract school {
    Student[] public students;
```

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16 <0.9.0;
struct Student {
    uint256 roll;
    string name;
    string div;
}
contract school {
    Student[] public students;

    function addStudent(
        uint256 _roll,
        string memory _name,
        string memory _div
    ) public {
        Student memory newStudent = Student({
            roll: _roll,
            name: _name,
            div: _div
        });
        students.push(newStudent);
    }
    event log(
        string _functionName,
        address _sender,
        uint256 _value,
        bytes _data
    );
    fallback() external payable {
        emit log("fallback", msg.sender, msg.value, msg.data);
    }
    receive() external payable {
        emit log("receive", msg.sender, msg.value, "");
    }
}
```

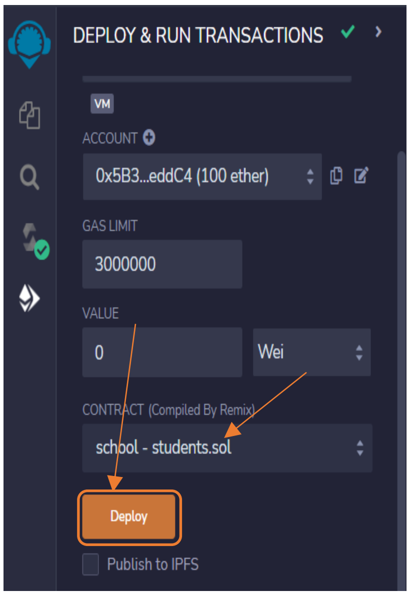
Step 2 : Compile the contract. Choose Owner account address of the contract and click on "Delpoy".



```
SOLIDITY COMPILER
COMPILER
0.8.17+commit.8df45f5f
Include nightly builds
Auto compile
Hide warnings
Advanced Configurations
Compile students.sol
Compile and Run script
CONTRACT
```

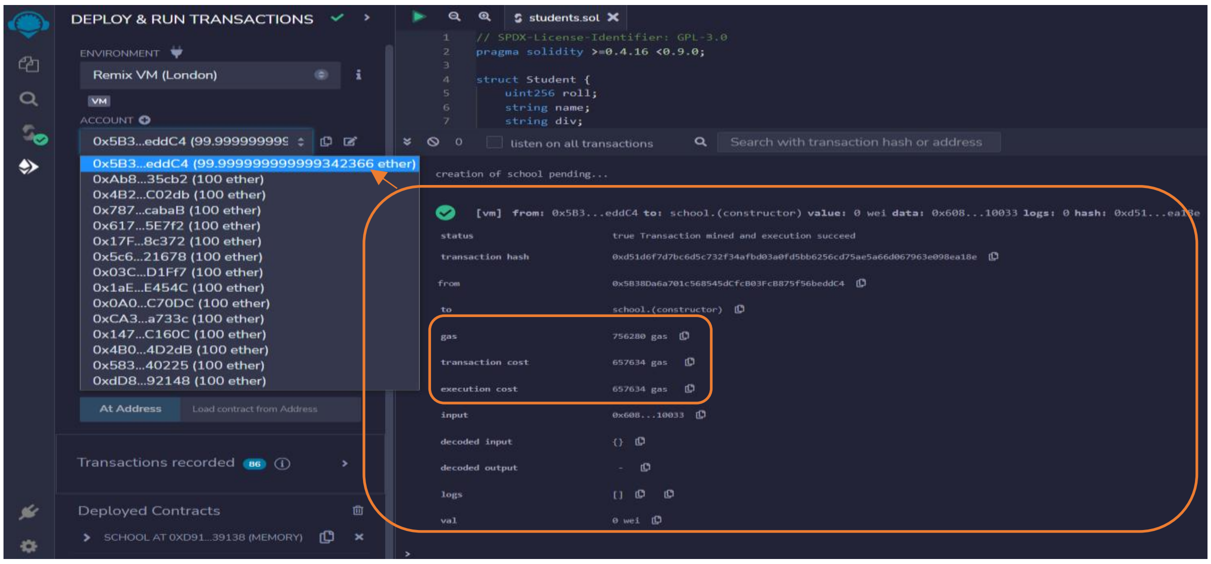


```
DEPLOY & RUN TRANSACTIONS
VM
ACCOUNT
0x5B3...eddC4 (100 ether)
0x5B3...eddC4 (100 ether)
0xAb8...35cb2 (100 ether)
0x4B2...C02db (100 ether)
0x787...cabaB (100 ether)
0x617...5E7f2 (100 ether)
0x17F...8c372 (100 ether)
0x5c6...21678 (100 ether)
0x03C...D1Ff7 (100 ether)
0x1aE...E454C (100 ether)
0x0A0...C70DC (100 ether)
0xCA3...a733c (100 ether)
0x147...C160C (100 ether)
0x4B0...4D2dB (100 ether)
0x583...40225 (100 ether)
0xdD8...92148 (100 ether)
At Address    Load contract from Address
```



```
DEPLOY & RUN TRANSACTIONS
VM
ACCOUNT
0x5B3...eddC4 (100 ether)
GAS LIMIT
3000000
VALUE
0    Wei
CONTRACT (Compiled By Remix)
school - students.sol
Deploy
Publish to IPFS
```

Step 3 : View the log and gas fees deduction after deploying smart contract.

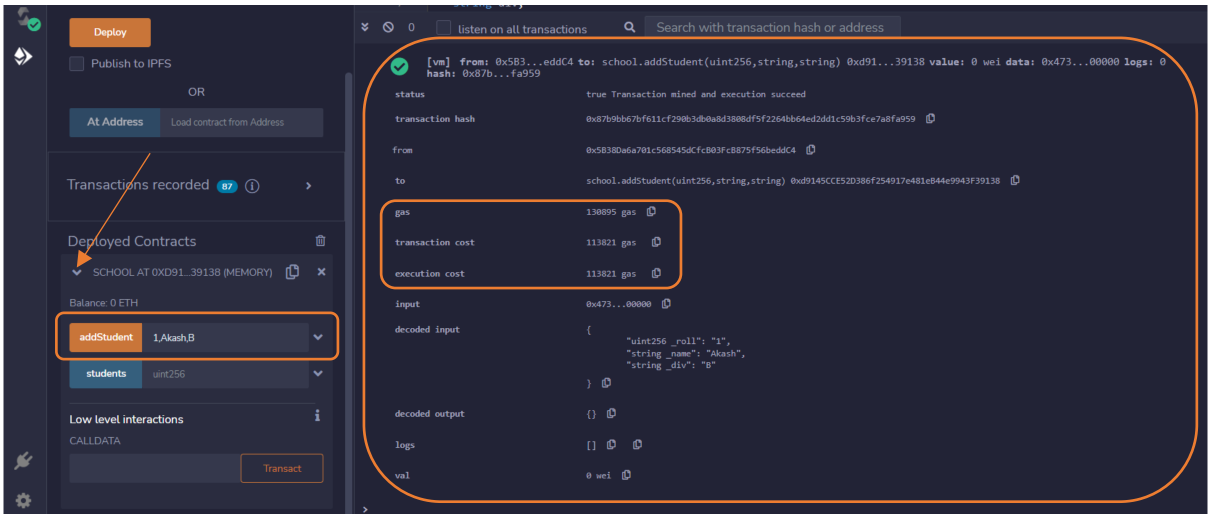


```
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Remix VM (London)
VM
ACCOUNT
0x5B3...eddC4 (99.99999999
0x5B3...eddC4 (99.9999999999999342366 ether)
0xAb8...35cb2 (100 ether)
0x4B2...C02db (100 ether)
0x787...cabaB (100 ether)
0x617...5E7f2 (100 ether)
0x17F...8c372 (100 ether)
0x5c6...21678 (100 ether)
0x03C...D1Ff7 (100 ether)
0x1aE...E454C (100 ether)
0x0A0...C70DC (100 ether)
0xCA3...a733c (100 ether)
0x147...C160C (100 ether)
0x4B0...4D2dB (100 ether)
0x583...40225 (100 ether)
0xdD8...92148 (100 ether)
At Address    Load contract from Address
Transactions recorded 86
Deployed Contracts
SCHOOL AT 0XD91...39138 (MEMORY)
```

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16 <0.9.0;

struct Student {
    uint256 roll;
    string name;
    string div;
```

```
creation of school pending...

[vm] from: 0x5B3...eddC4 to: school.(constructor) value: 0 wei data: 0x608...10033 logs: 0 hash: 0xd51...ea18e
status              true Transaction mined and execution succeed
transaction hash    0xd51d6f7d7bc6d5c732f34afbd03a0fd5bb6256cd75ae5a66d067963e098ea18e
from                0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                  school.(constructor)
gas                 756280 gas
transaction cost    657634 gas
execution cost      657634 gas
input               0x608...10033
decoded input       {}
decoded output      -
logs                []
val                 0 wei
```

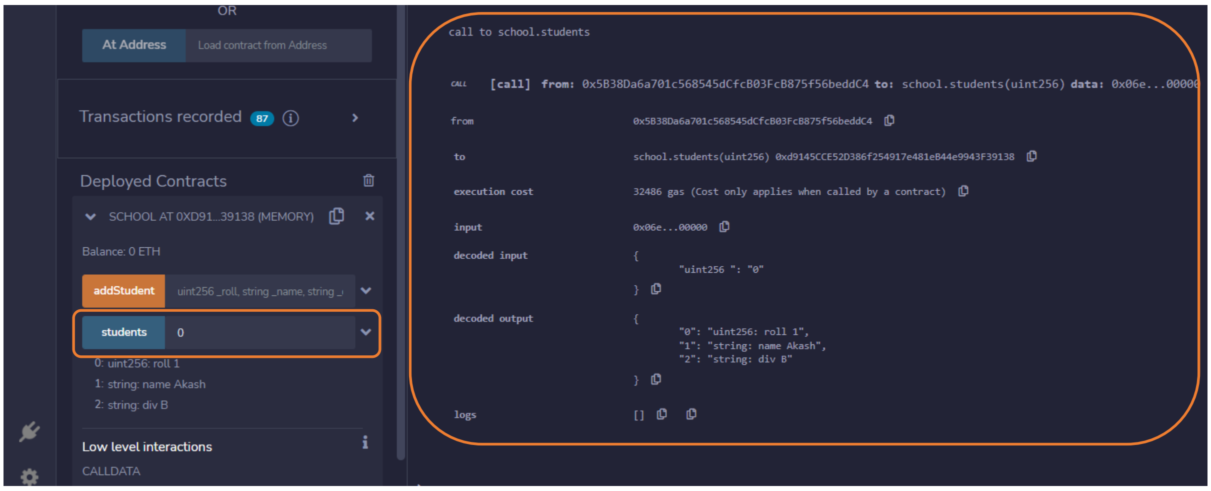
Step 4: Add new student in students array by entering roll no. , name and div as shown in fig. and then click on addStudent and observe the transaction and gas fee in the log.



```
Deploy
Publish to IPFS
OR
At Address    Load contract from Address
Transactions recorded 87
Deployed Contracts
SCHOOL AT 0XD91...39138 (MEMORY)
Balance: 0 ETH
addStudent    1,Akash,B
students      uint256
Low level interactions
CALLDATA
Transact
```

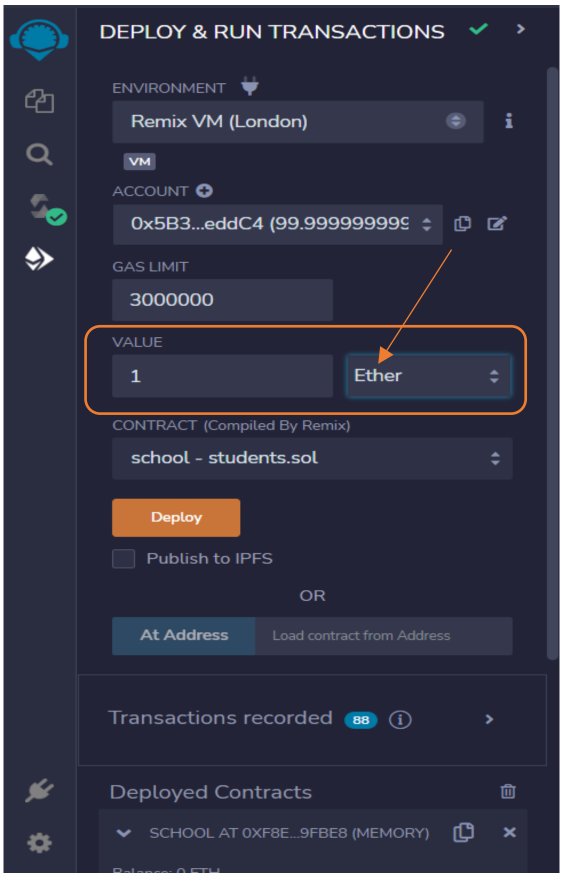
```
[vm] from: 0x5B3...eddC4 to: school.addStudent(uint256,string,string) 0xd91...39138 value: 0 wei data: 0x473...00000 logs: 0
hash: 0x87b...fa959
status              true Transaction mined and execution succeed
transaction hash    0x87b9bb67bf611cf290b3db0a8d3808df5f2264bb64ed2dd1c59b3fce7a8fa959
from                0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                  school.addStudent(uint256,string,string) 0xd9145CCE52D386f254917e481eB44e9943F39138
gas                 130895 gas
transaction cost    113821 gas
execution cost      113821 gas
input               0x473...00000
decoded input       {
                          "uint256 _roll": "1",
                          "string _name": "Akash",
                          "string _div": "B"
                    }
decoded output      {}
logs                []
val                 0 wei
```

Step 5 : Insert index “0” in the students array to check whether new student is added or not and note that no gas fee transaction is taken for fetching the data.

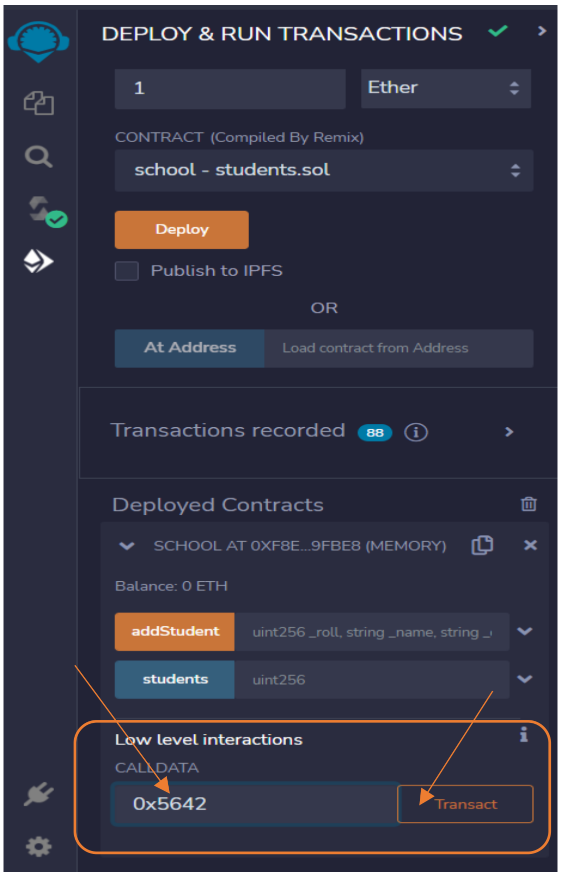


```
call to school.students

CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: school.students(uint256) data: 0x06e...0000

from              0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                school.students(uint256) 0xd9145CCE52D386f254917e481eB44e9943F39138
execution cost    32486 gas (Cost only applies when called by a contract)
input             0x06e...00000
decoded input     {
                        "uint256 ": "0"
                  }
decoded output    {
                        "0": "uint256: roll 1",
                        "1": "string: name Akash",
                        "2": "string: div B"
                  }
logs              []
```

Step 6 : Enter amount of ETH in value textbox to send to the smart contract and add some hexadecimal data in Low level intercations and click on Transact to check the fallback function .

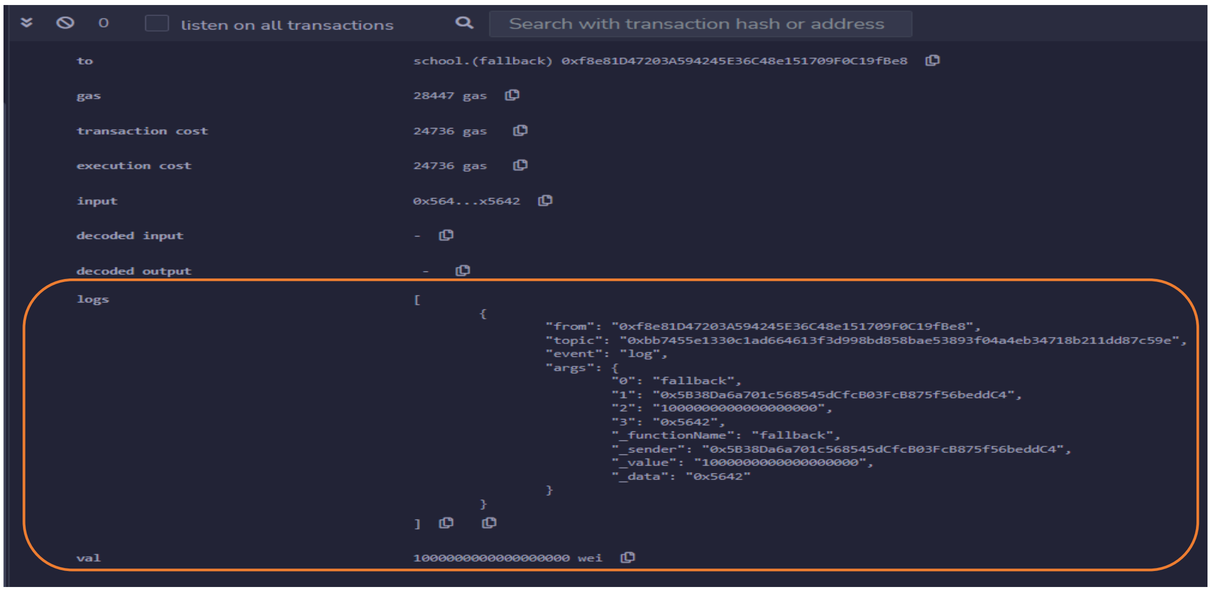


ENVIRONMENT: Remix VM (London)
ACCOUNT: 0x5B3...eddC4 (99.99999999
GAS LIMIT: 3000000
VALUE: 1 Ether
CONTRACT (Compiled By Remix): school - students.sol

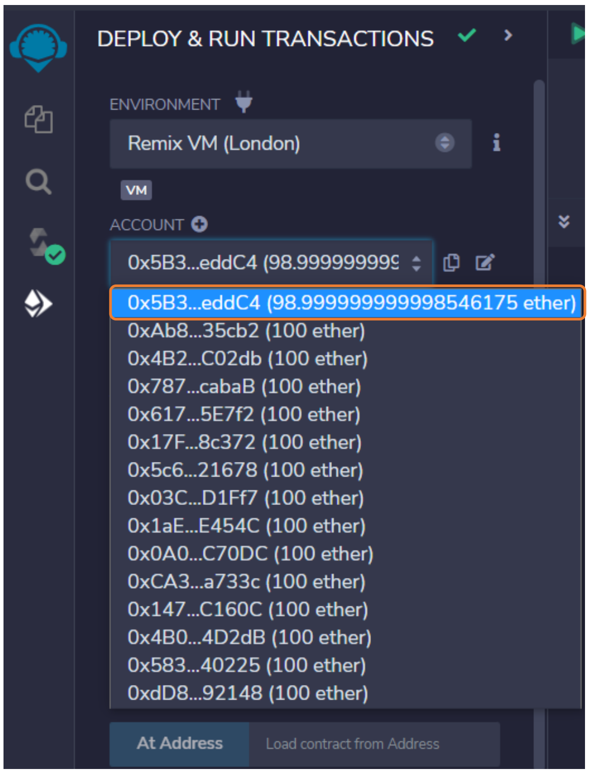


VALUE: 1 Ether
CONTRACT (Compiled By Remix): school - students.sol
CALLDATA: 0x5642

Step 7 : Notice the data and value of ETH in wei unit in the log and amount of ETH in wallet address.


listen on all transactions
Search with transaction hash or address

```
to                    school.(fallback) 0xf8e81D47203A594245E36C48e151709F0C19fBe8
gas                   28447 gas
transaction cost      24736 gas
execution cost        24736 gas
input                 0x564...x5642
decoded input         -
decoded output         -
logs                  [
                              {
                                      "from": "0xf8e81D47203A594245E36C48e151709F0C19fBe8",
                                      "topic": "0xbb7455e1330c1ad664613f3d998bd858bae53893f04a4eb34718b211dd87c59e",
                                      "event": "log",
                                      "args": {
                                              "0": "fallback",
                                              "1": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                                              "2": "1000000000000000000",
                                              "3": "0x5642",
                                              "_functionName": "fallback",
                                              "_sender": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                                              "_value": "1000000000000000000",
                                              "_data": "0x5642"
                                      }
                              }
                      ]
val                   1000000000000000000 wei
```



ENVIRONMENT: Remix VM (London)

ACCOUNT: 0x5B3...eddC4 (98.99999999

```
0x5B3...eddC4 (98.999999999998546175 ether)
0xAb8...35cb2 (100 ether)
0x4B2...C02db (100 ether)
0x787...cabaB (100 ether)
0x617...5E7f2 (100 ether)
0x17F...8c372 (100 ether)
0x5c6...21678 (100 ether)
0x03C...D1Ff7 (100 ether)
0x1aE...E454C (100 ether)
0x0A0...C70DC (100 ether)
0xCA3...a733c (100 ether)
0x147...C160C (100 ether)
0x4B0...4D2dB (100 ether)
0x583...40225 (100 ether)
0xdD8...92148 (100 ether)
```