One Day Faculty Orientation Program

on

Laboratory Practice-III (410246)

(02nd August, 2022)

Organized by

Department of Computer Engineering, MVPS's KBT COE, Nashik

In association with

BoS Computer Engineering, Savitribai Phule Pune University, Pune

410246: Laboratory Practice III Group C: Blockchain Technology

Presented By:

Prof. Shailaja Lohar

JSPM’s Rajarshi Shahu College of Engineering

Group C: Blockchain Technology

Any 5 assignments and 1 Mini project are mandatory.

1.	Installation of MetaMask and study spending Ether per transaction.
2.	Create your own wallet using Metamask for crypto transactions.
3.	Write a smart contract on a test network, for Bank account of a customer for following operations: • Deposit money • Withdraw Money • Show balance
4.	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.
5.	Write a survey report on types of Blockchains and its real time use cases.
6.	Write a program to create a Business Network using Hyperledger

MINI PROJECTS

Mini Projects–

- Develop a Blockchain based application dApp (de-centralized app) for e-voting system.
- Develop a Blockchain based application for transparent and genuine charity
- Develop a Blockchain based application for health related medical records
- Develop a Blockchain based application for mental health

Installation of MetaMask and study spending Ether per transaction.

METAMASK

- The existing browsers on our systems are centralized. In order to create a de-centralized system we add a plug-in called “Metamask”
- We need it because we need “ether” (currency for blockchain)
- Installed Metamask(Gives a virtual Ethereum wallet)
- Created a simple smart contract through MetaMask(Using fake ether i.e currency of blockchain)
- In-depth study of state-of-art on smart contract implementation.

REMIX IDE

Platform to create and deploy smart contract, supports solidity .

SOLIDITY

A Language to create smart contracts, similar to Javascript.

Creating a De-centralized platform for testing a smart contract

1. Pre-requisites for creating a Blockchain environment:
   - Adding Metamask extension to default browser
   - Creating a wallet through test network Rinkeby.
   - Currency needed for blockchain transaction – ether
   - Rinkeby gives this ether – 18.75 ethers / 3 days
2. Smart contract in solidity language on IDE Remix

101 Blockchains

WHAT IS A SMART CONTRACT?

WHAT IS A SMART CONTRACT?

A smart contract is a set of digital codes that is used to exchange assets including shares, money, or property without the need for any intermediates to function.

HOW SMART CONTRACT WORKS?

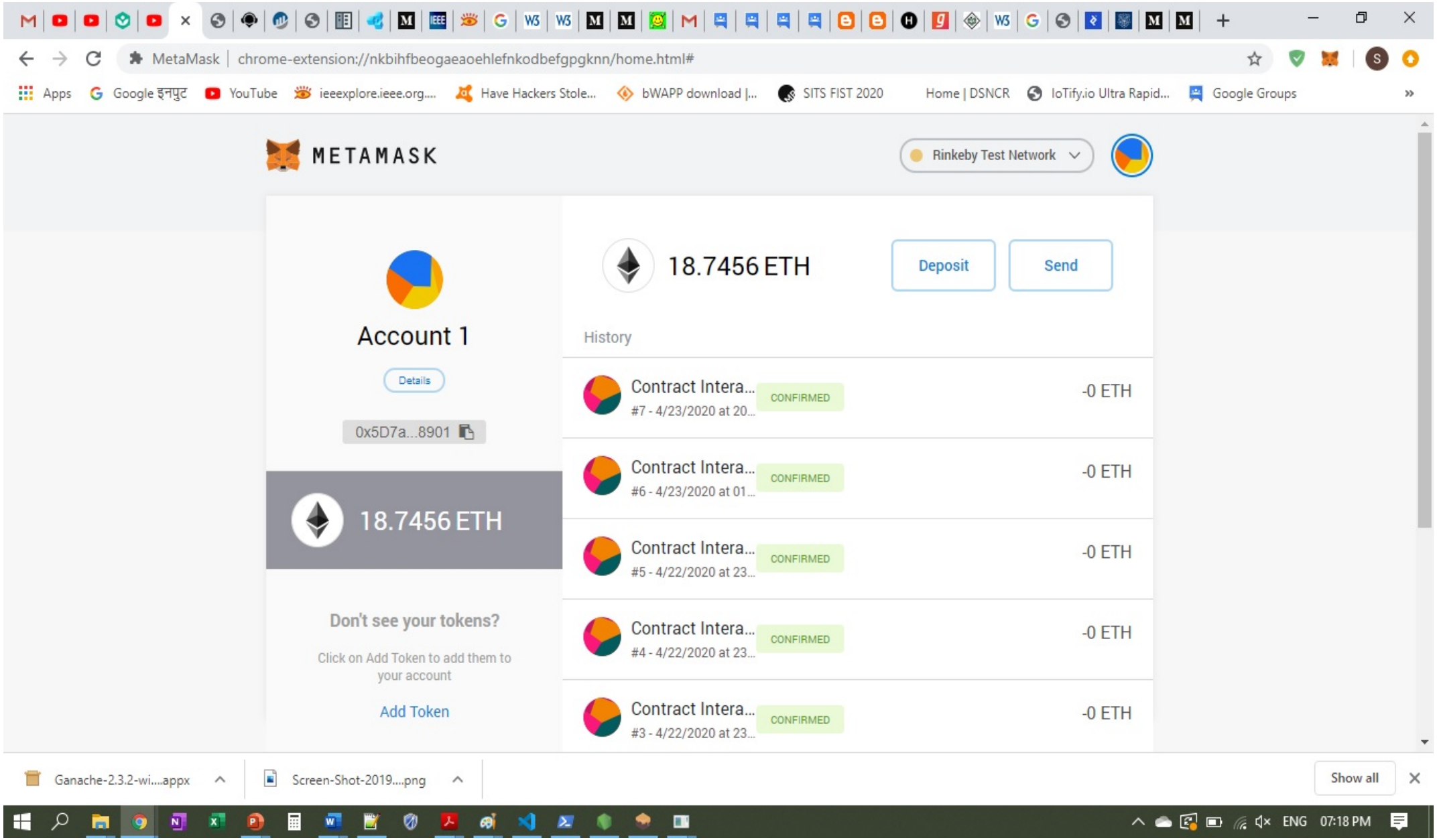
1. A party codes the smart contract
2. Both parties finalize the smart contract rules
3. The contract goes live and is stored in the ledger
4. The contract automatically executes when the conditions are met

SMART CONTRACT BENEFITS

- Secure
- Autonomous
- Interruption free
- Trustless
- Cost-effective
- Fast performance

USE CASES

- Government voting
- Supply chain management
- Mortgage system
- Record storing
- Real estate market
- Insurance claim


MetaMask
chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#
Apps
Google इनपुट
YouTube
ieeexplore.ieee.org....
Have Hackers Stole...
bWAPP download |...
SITS FIST 2020
Home | DSNCR
IoTify.io Ultra Rapid...
Google Groups
METAMASK
Rinkeby Test Network
Account 1
Details
0x5D7a...8901
18.7456 ETH
Don't see your tokens?
Click on Add Token to add them to your account
Add Token
18.7456 ETH
Deposit
Send
History
Contract Intera...
#7 - 4/23/2020 at 20...
CONFIRMED
-0 ETH
Contract Intera...
#6 - 4/23/2020 at 01...
CONFIRMED
-0 ETH
Contract Intera...
#5 - 4/22/2020 at 23...
CONFIRMED
-0 ETH
Contract Intera...
#4 - 4/22/2020 at 23...
CONFIRMED
-0 ETH
Contract Intera...
#3 - 4/22/2020 at 23...
CONFIRMED
-0 ETH
Ganache-2.3.2-wi....appx
Screen-Shot-2019....png
Show all
ENG
07:18 PM

SMART CONTRACT IMPLEMENTATION DETAILS

rinkeby.etherscan.io/tx/0xad0eeb982fa0a3eab2dcd089ce4860544b2d61f5c9330de9bf685503367f1b6a

Etherscan

Rinkeby Testnet Network

All Filters | Search by Address / Txn Hash / Block / Token / Ens

Home | Blockchain | Tokens | Misc | Rinkeby

Transaction Details

Overview | State

[This is a Rinkeby Testnet transaction only]

Transaction Hash:	0xad0eeb982fa0a3eab2dcd089ce4860544b2d61f5c9330de9bf685503367f1b6a
Status:	Success
Block:	6366572 573314 Block Confirmations
Timestamp:	99 days 12 hrs ago (Apr-23-2020 03:00:07 PM +UTC)
From:	0x5d7a55c5f52d58d71adefc87c7423345e5df8901
To:	Contract 0xb44a96735f1fca0ff0abb0ee5d83d71f82ac5f05
Value:	0 Ether ($0.00)
Transaction Fee:	0.00027291 Ether ($0.000000)
Gas Limit:	27,291

ETHER(TRANSACTION FEE SPENT ON THE SMART CONTRACT)

rinkeby.etherscan.io/tx/0xad0eeb982fa0a3eab2dcd089ce4860544b2d61f5c9330de9bf685503367f1b6a

Overview | State

Transaction Fee:	0.00027291 Ether ($0.000000)
Gas Limit:	27,291
Gas Used by Transaction:	27,291 (100%)
Gas Price:	0.00000001 Ether (10 Gwei)
Nonce Position	7 5
Input Data:	

```
Function: deposit(int256 amount) ***

MethodID: 0xf04991f0
[0]:  000000000000000000000000000000000000000000000000000000000000012c
```

View Input As

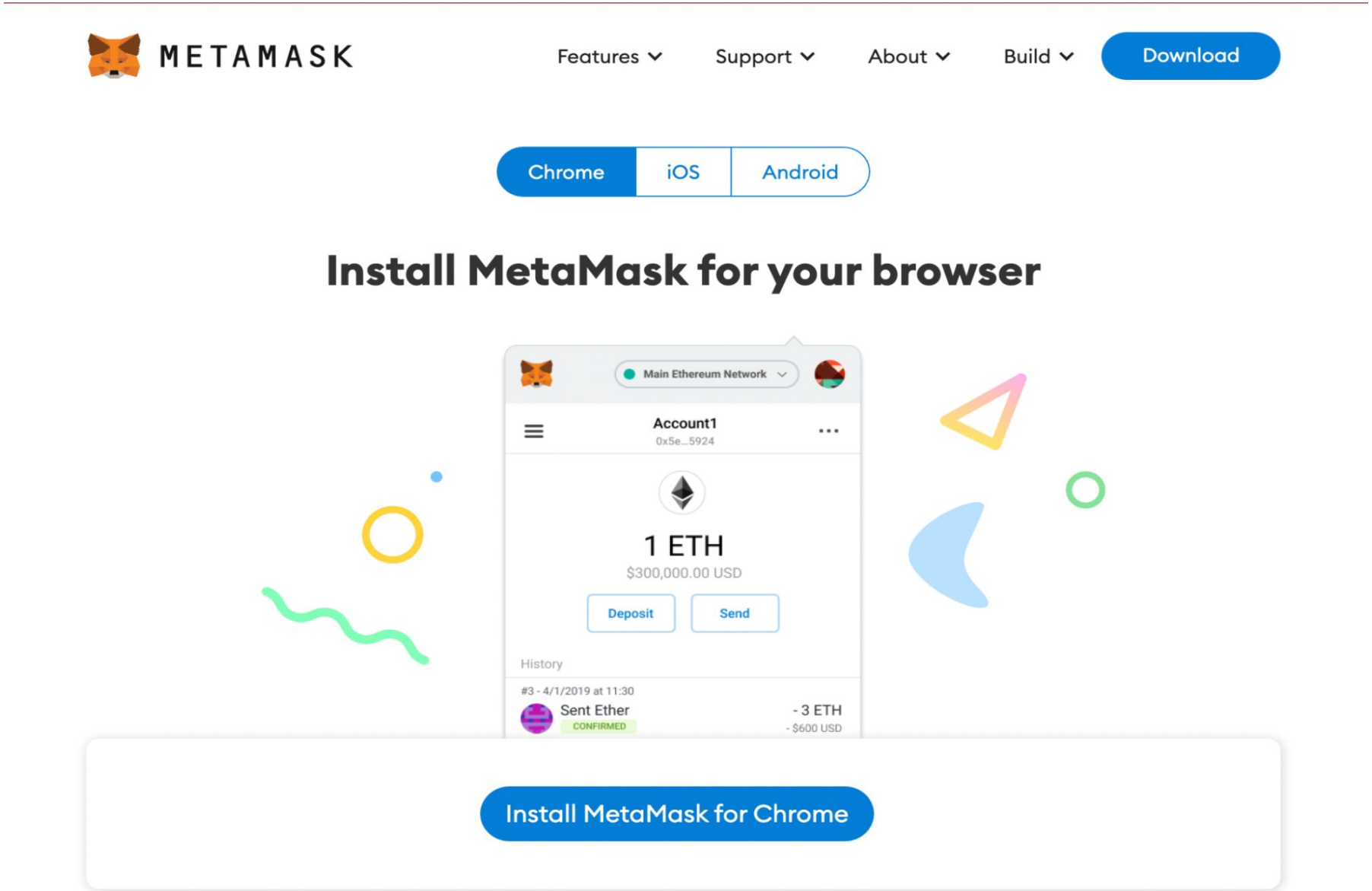
Click to see Less

Powered by Ethereum

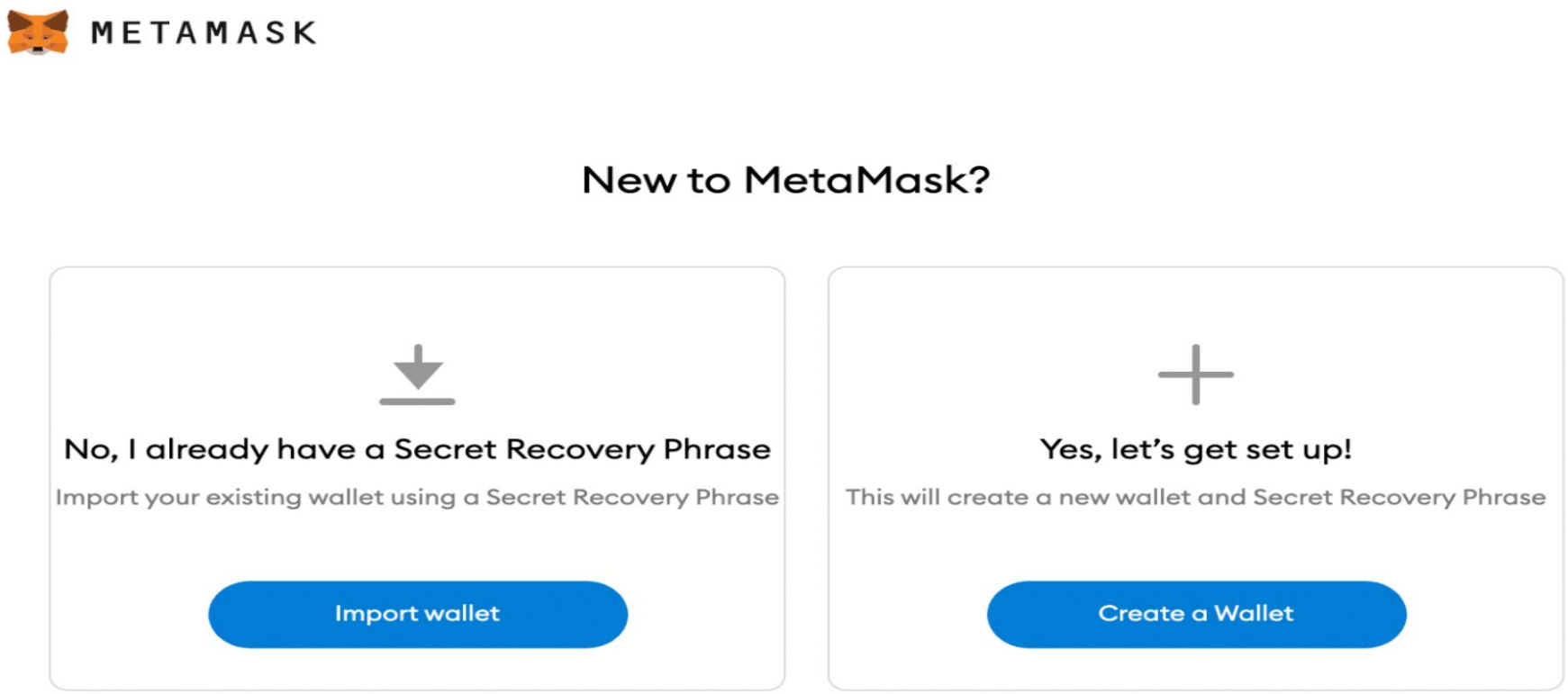
Preferences

Etherscan © 2020 (Rinkeby) | Donate

Create your own wallet using Metamask for crypto transactions.



Navigate to the extension icon in the top right corner of your web browser and find the MetaMask option, once you've successfully downloaded the software. Click the "Get Started" button and you'll be taken to the next page and presented with two options (see below.)



METAMASK

New to MetaMask?

No, I already have a Secret Recovery Phrase

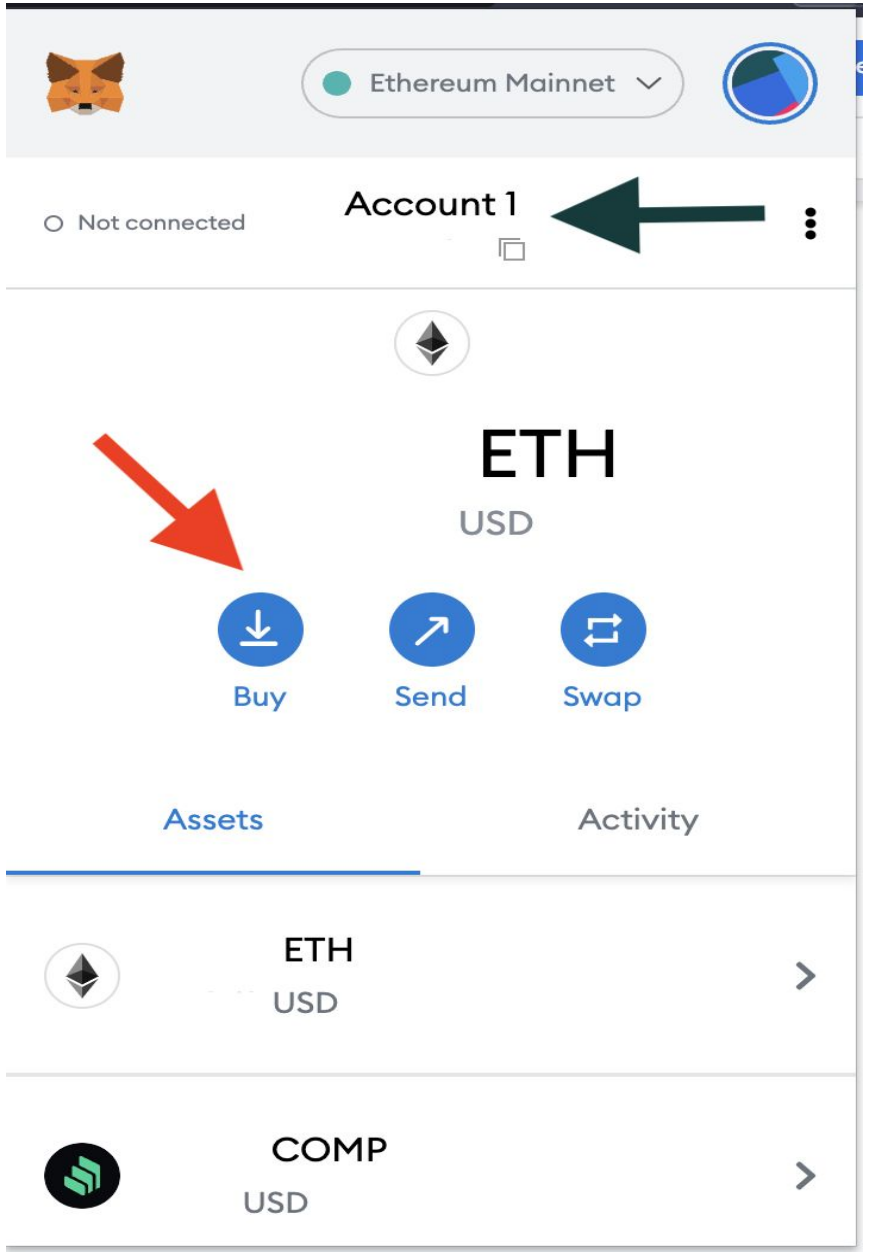
Import your existing wallet using a Secret Recovery Phrase

Yes,

This will create a nev

Once you've completed the above steps, you'll be able to access your new MetaMask wallet. There are two main components you'll need to familiarize yourself with so that you can begin using the software:

- **Identifying your public address:** This is the address you can freely share with people or platforms like exchanges in order to receive cryptocurrency into your wallet. Think of it as your home address that you share with people to receive inbound mail. It's always advisable, however, to check to make sure any inbound tokens are compatible with MetaMask first before receiving them, otherwise, they might be lost forever.
- **How to fund/buy and send:** These are the core functions of MetaMask.



- You can locate your unique MetaMask public address by clicking the “Account 1″ button (black arrow).
- Finally, in order to begin interacting with any Ethereum platform, you’ll first need to fund your MetaMask wallet with an amount of ether – the native cryptocurrency of Ethereum. All actions on the blockchain cost a fee, whether that’s moving tokens from A to B or creating an NFT collection. This fee, known as a “gas” fee, is denominated in ether
- How much you choose to fund your wallet depends on how much you intend to interact with various platforms. For moderate use, $100 worth of ether is usually a good starting point to cover any initial fees.

Writing a Smart contract in Remix IDE

- Remix-IDE - a powerful open-source tool that provides the ability to develop a smart contract from a browser.

```
pragma solidity ^0.6.6;contract
BankContract {
struct client_account{
int client_id;
address client_address;
uint client_balance_in_ether;
} client_account[] clients;
}
```

to assign an ID to each client whenever they join the contract, so we define an int counter and set it to 0 in the constructor of the contract.

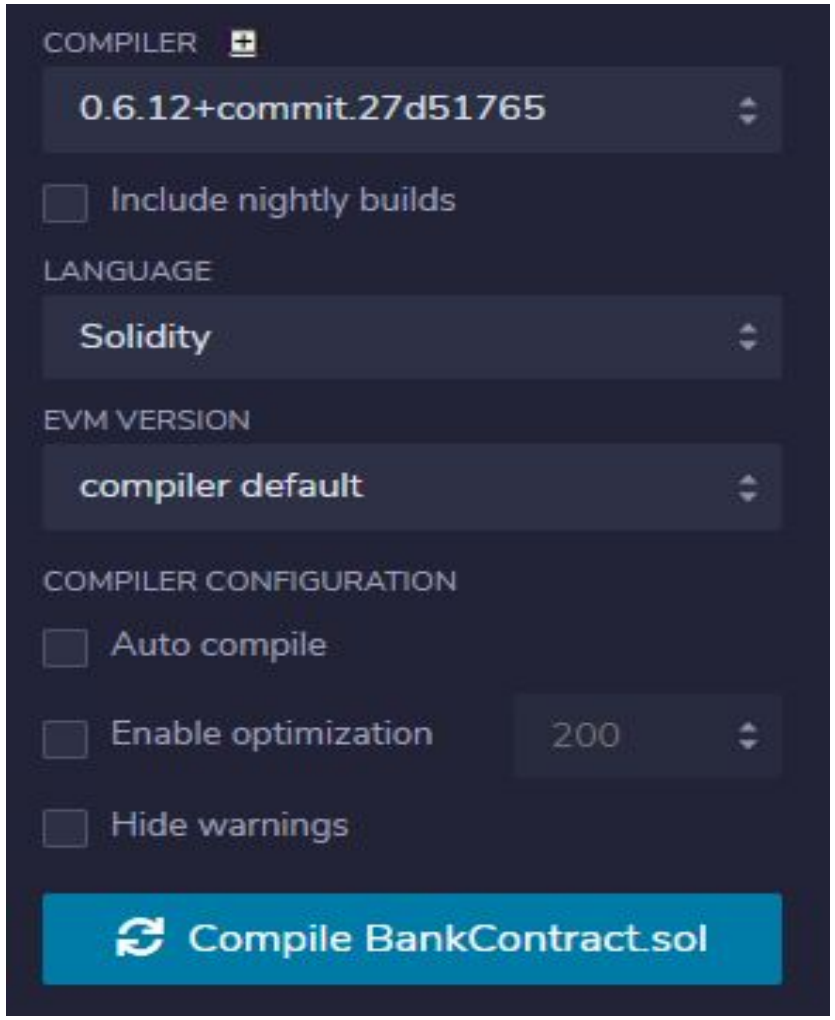
```
pragma solidity ^0.6.6;contract BankContract {
struct client_account{
int client_id;
address client_address;
uint client_balance_in_ether;
} client_account[] clients; int clientCounter;
constructor() public{
clientCounter = 0;
}
}
```

The Final State of the Smart Contract: Click for the code:

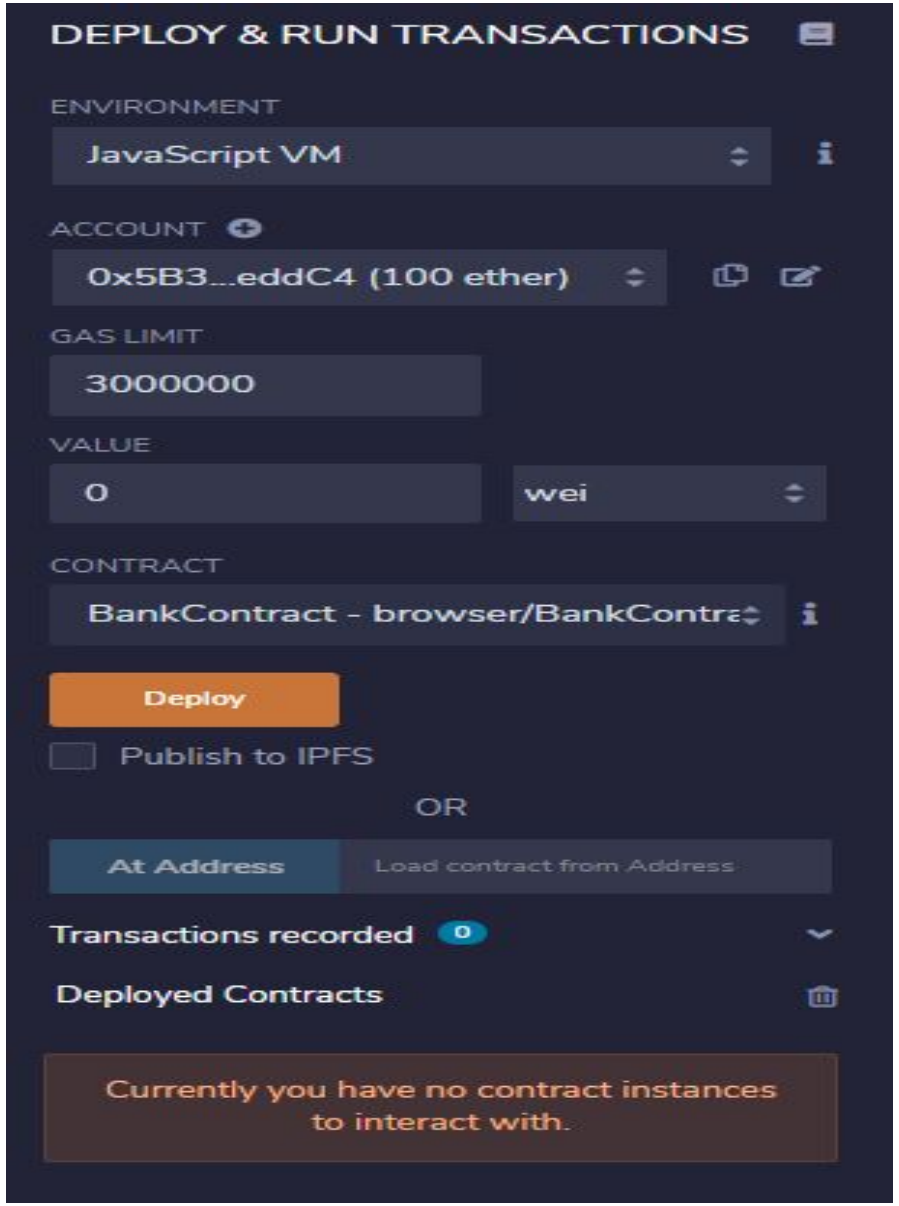
Compile the Smart Contract

After finishing the development of the smart contract, we'll compile it onto the Remix IDE.

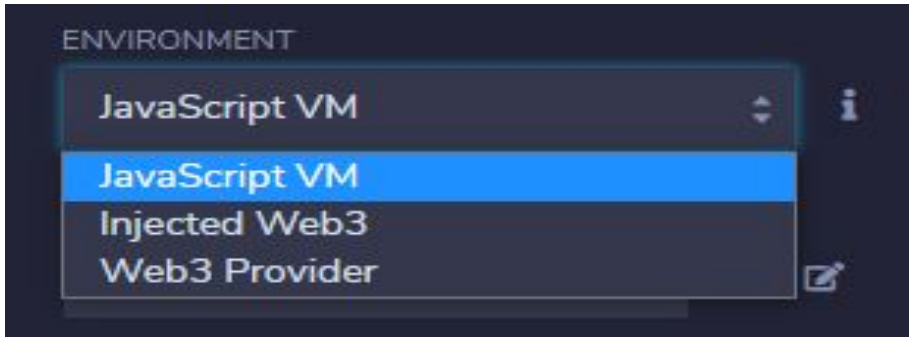
The following image shows what we face when we go to the Solidity Compiler section. We select the compiler version according to the version we specified before and click the button at the bottom of the section.



Deploy the Smart Contract



- The Remix IDE presents various opportunities to deploy the smart contract into various environments.



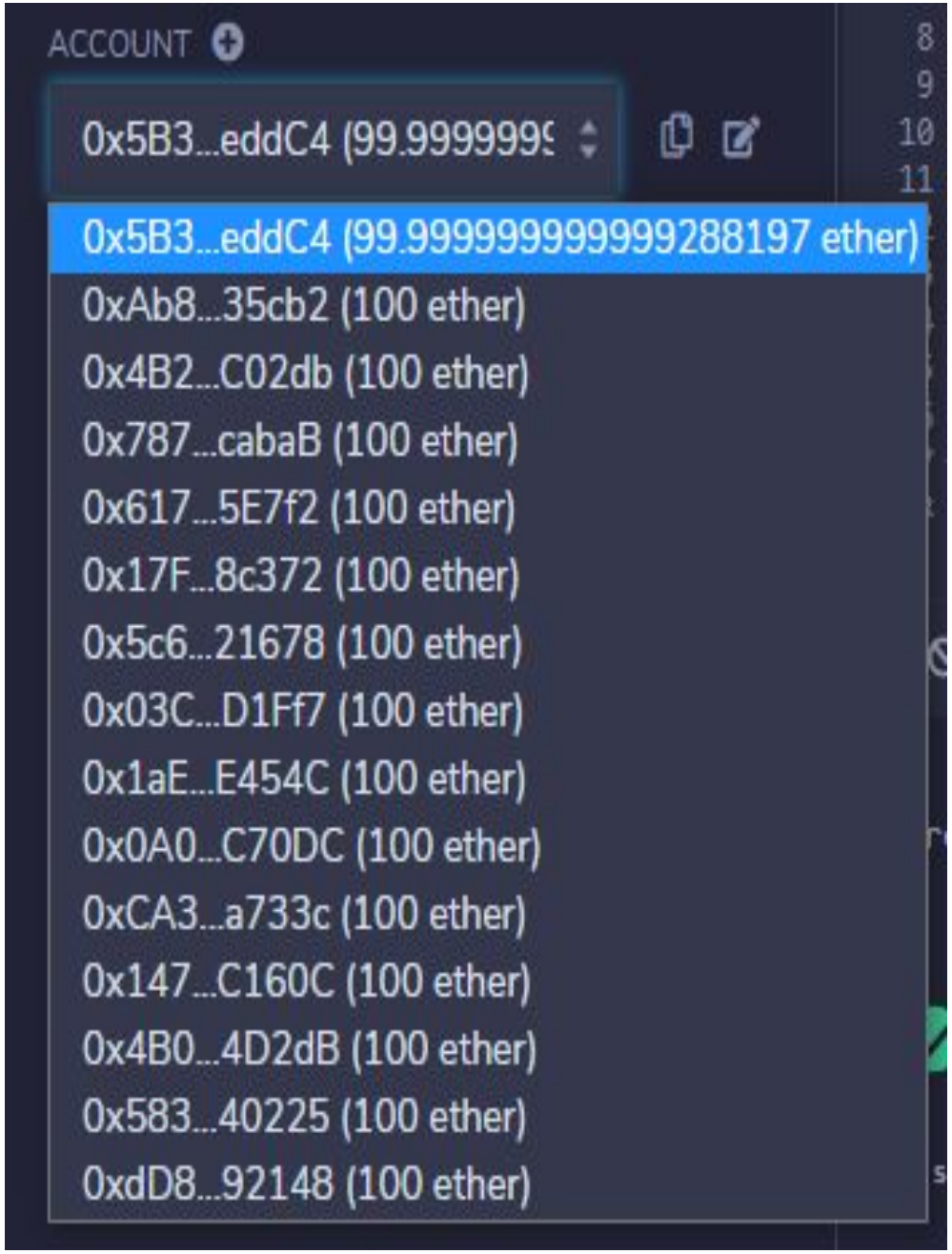
- deploy our contract on the JavaScript VM environment, so we'll select it among the following environments.

In the accounts combo, there are many accounts we'll be able to use during the deploying and testing of the smart contract. Among these accounts that the Remix IDE

```
[vm] from: 0x5B3...eddC4 to: BankContract.(constructor) value: 0 wei data: 0x608...c0033 logs: 0 hash: 0xed7...95494
```

status	true Transaction mined and execution succeed
transaction hash	0xed7a5f52eb27edf3e703b404cd32aa8dc39dd909d2a939c0ef97412f2fb95494
contract address	0xd9145CCE52D386f254917e481eB44e9943F39138
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to	BankContract.(constructor)
gas	3000000 gas
transaction cost	711803 gas
execution cost	500751 gas
hash	0xed7a5f52eb27edf3e703b404cd32aa8dc39dd909d2a939c0ef97412f2fb95494
input	0x608...c0033
decoded input	{}
decoded output	-
logs	[]
value	0 wei

- After setting the environment and the account, we're ready to deploy it, so we click the Deploy button.

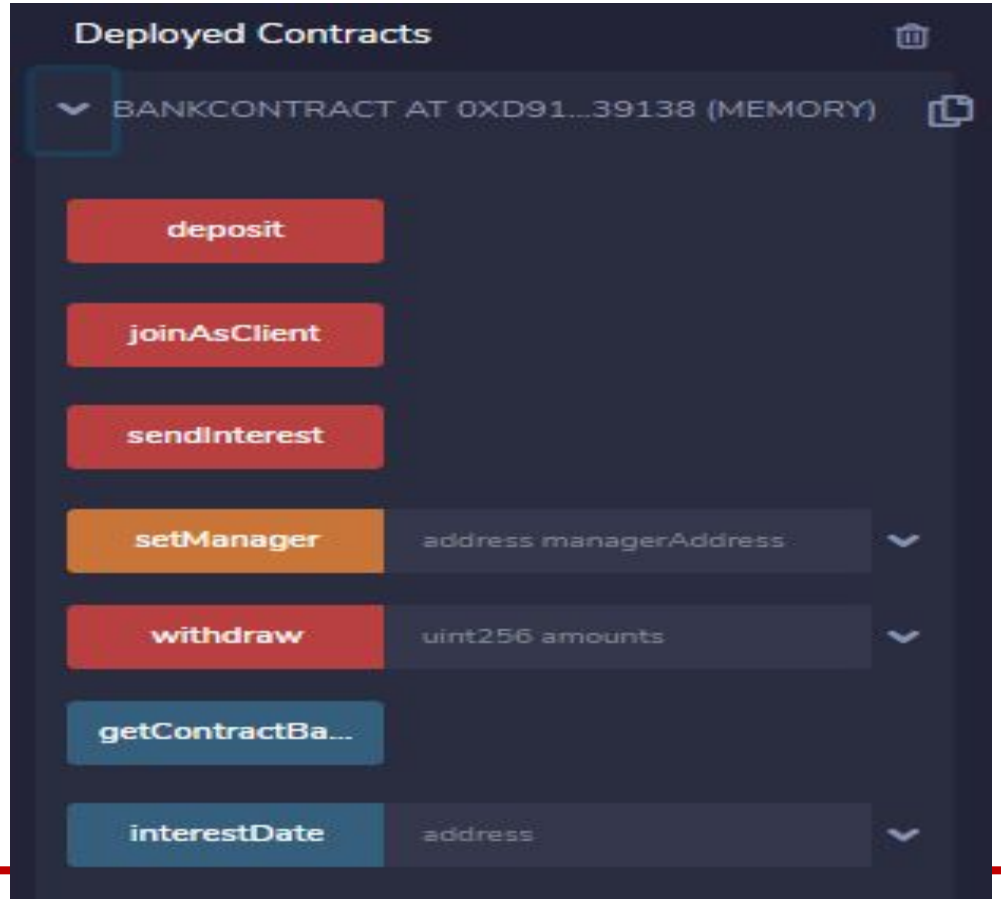


According to the result of the transaction above, the smart contract was deployed to the account selected successfully.

Deploying the smart-contract operation causes a cost to the sender who deploys it.

The transaction cost represents the cost we need to deploy the contract — the amount placed in the transaction cost was taken from the account — as shown in the following image.

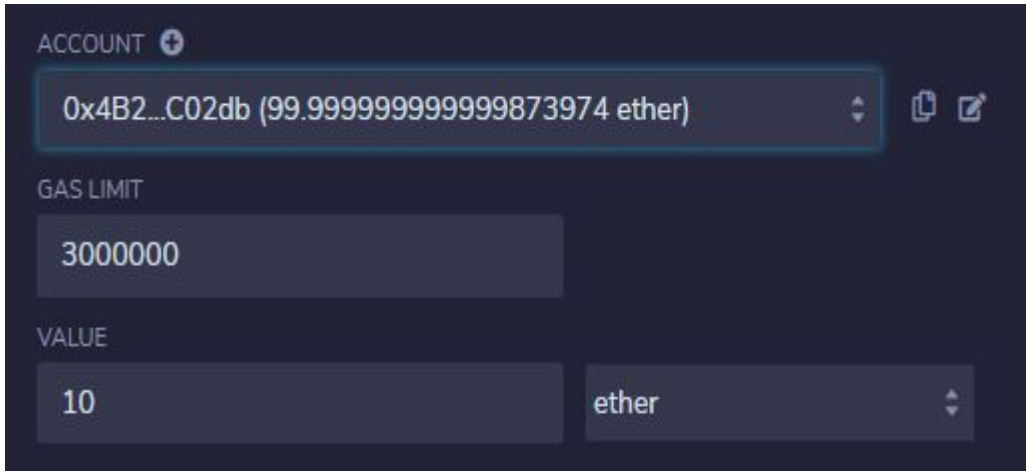
The smart contract deployed can be seen in the Deployed Contracts subsection on the left.



Run the Transactions
call the functions that compound the smart contract developed. When we expand the relevant contract in the Deployed Contract subsection, the methods developed appear.

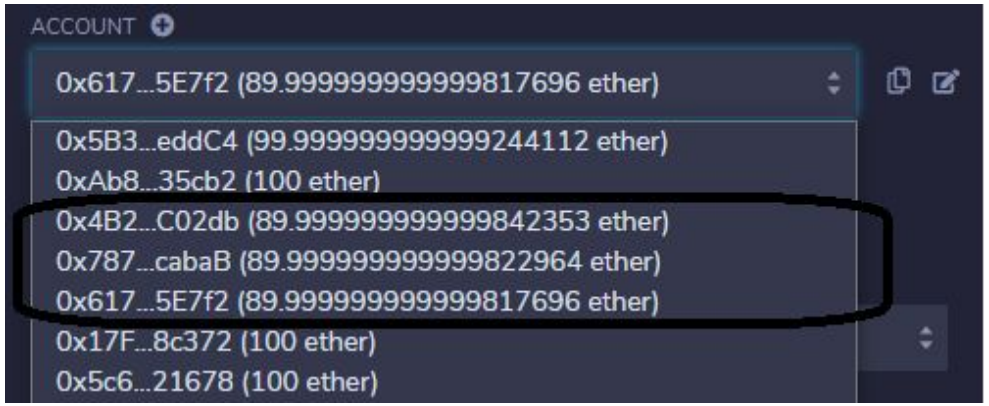
The deposit method

Now, we'll send 10 ETH from the clients' accounts to the contract by using the deposit method. In the deposit method, we take the amount declared in the msg.value from the sender that's represented in the msg.sender variable.



we set 10 ETH and call the deposit method by clicking the red deposit button for each client account , like we did before for the joinAsClient method. After these operations, the following messages show in the terminal, which means those three accounts sent 10 ETH from their account to the contract address.

Also, the final state of the accounts' balances look like this:




```
transact to BankContract.deposit pending ...

[vm] from: 0x4B2...C02db to: BankContract.deposit() 0xd91...39138 value: 1000000000000000000 wei data: 0xd0e...30db0 logs: 0 hash: 0x135...5caa8

transact to BankContract.deposit pending ...

[vm] from: 0x787...cabaB to: BankContract.deposit() 0xd91...39138 value: 1000000000000000000 wei data: 0xd0e...30db0 logs: 0 hash: 0xec7...08d8d

transact to BankContract.deposit pending ...

[vm] from: 0x617...5E7f2 to: BankContract.deposit() 0xd91...39138 value: 1000000000000000000 wei data: 0xd0e...30db0 logs: 0 hash: 0xb33...6ef72
```

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.

- The solidity fallback function is executed if none of the other functions match the function identifier or no data was provided with the function call.
- Only one unnamed function can be assigned to a contract and it is executed whenever the contract receives plain Ether without any data. To receive Ether and add it to the total balance of the contract, the fallback function must be marked payable.
- **If no such function exists, the contract cannot receive Ether through regular transactions and will throw an exception.**

Properties of a fallback function:

1. Has no name or arguments.
2. If it is not marked **payable**, the contract will throw an exception if it receives plain ether without data.
3. Can not return anything.
4. Can be defined once per contract.
5. It is also executed if the caller meant to call a function that is not available
6. It is mandatory to mark it external.
7. It is limited to 2300 gas when called by another function. It is so for as to make this function call as cheap as possible.

Code: Example for *Fallback*

```
[vm] from: 0x9e5...8b217 to: GeeksForGeeks.(constructor) value: 0 wei data: 0x608...d0029 logs: 0 hash: 0x8fd...c7599
```

Debug

```
transact to GeeksForGeeks.SetX pending ...
```

```
[vm] from: 0x9e5...8b217 to: GeeksForGeeks.SetX(uint256) 0x146...d6f1e value: 0 wei data: 0x117...0000b logs: 0 hash: 0x03e...009a5
```

Debug

status	0x1 Transaction mined and execution succeed
transaction hash	0x03e7987ba97ce5f1df47e8a92833b19fd6cbde78a585e261b155e2b34fa009a5
from	0x9e57b1db5aeb73cab107417b08a08b868ad8b217
to	GeeksForGeeks.SetX(uint256) 0x146ba5af4b2f68b25643feea4f3d8db5460d6f1e
gas	3000000 gas
transaction cost	26753 gas
execution cost	5289 gas
hash	0x03e7987ba97ce5f1df47e8a92833b19fd6cbde78a585e261b155e2b34fa009a5
input	0x117...0000b
decoded input	{ "uint256 _x": { "_hex": "0x0b" } }
decoded output	{ "0": "bool: true" }
logs	[]
value	0 wei

Case studies :

- Asset registry
- Cryptocurrencies
- Interbank reconciliation
- Smart contracts
- Supply chain traceability
- ‘Know your customer’ compliance in financial services
- Managing clinical trials in healthcare and life sciences
- Asset optimisation in the energy and utilities sector
- Royalty payments for musicians

THANK YOU