Weekly Assignment 3

ERC-20 Token

Objective

After completion of this assignment you will be able to :

- How to write ERC 20 token smart contract
- Understand inheritance
- Compile the Smart Contract using Remix Web IDE
- Deploy the Smart Contract to Rinkeyby Test Network
- Test the token functionalities

ERC20 Token - brief introduction

- Read the blog on **ERC20 Token** for brief introduction.

Write a Smart Contract for Decentralized Election

Note before you start with the solidity file :

1. Select the version of solidity compiler for this example as 0.4.19
2. Everything in the code written below with "//" is a comment and is provided for your understanding
3. Copy complete code given in multiple slides (ERC20-1 to ERC20-5) into the remix editor and then perform the compilation and deployment steps

ERC20 - 1

```
pragma solidity ^0.4.24;
contract Token {
    function totalSupply() constant returns (uint256 supply) {}
    function balanceOf(address _owner) constant returns (uint256 balance) {}
    function transfer(address _to, uint256 _value) returns (bool success) {}
    function transferFrom(address _from, address _to, uint256 _value) returns
(bool success) {}
    function approve(address _spender, uint256 _value) returns (bool success){}
    function allowance(address _owner, address _spender) constant returns
(uint256 remaining) {}
    event Transfer(address indexed _from, address indexed _to, uint256 _value);
    event Approval(address indexed _owner, address indexed _spender, uint256
_value); }
```

ERC20 - 2

```
contract XYZToken is Token {
    string public name;
    uint8 public decimals;
    string public symbol;
    uint256 public
unitsOneEthCanBuy;
    uint256 public totalEthInWei;
    address public fundsWallet;
```

```
constructor() public {
        balances[msg.sender] =
1000000000000000000000;
        totalSupply =
1000000000000000000000;
        name = "XYZToken";
        decimals = 18;
        symbol = "XYZ";
        unitsOneEthCanBuy = 10;
        fundsWallet = msg.sender;
    }
```

ERC20 - 3

```
function transfer(address _to,
uint256 _value) returns (bool
success) {
if (balances[msg.sender] >= _value
&& balances[_to] + _value >
balances[_to]) {
   balances[msg.sender] -= _value;
   balances[_to] += _value;
   emit Transfer(msg.sender, _to,
_value);
   return true;
   } else { return false; }    }
```

```
function transferFrom(address _from,
address _to, uint256 _value) returns
(bool success) {
    if (balances[_from] >= _value &&
allowed[_from][msg.sender] >= _value
&& balances[_to] + _value >
balances[_to]) {
    balances[_to] += _value;
    balances[_from] -= _value;
    allowed[_from][msg.sender] -=
_value;
```

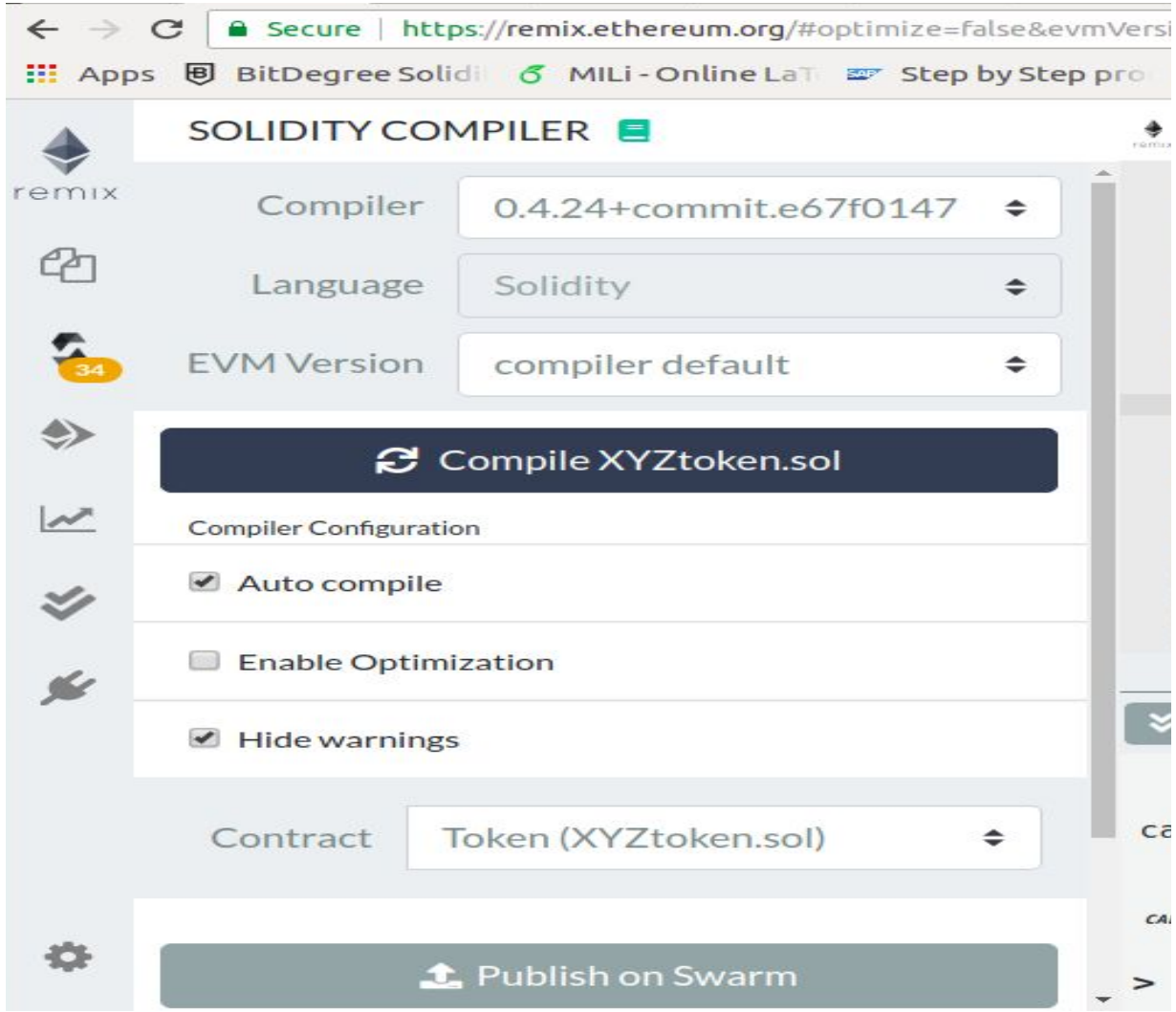
ERC20- 4

```
emit Transfer(_from, _to, _value);
            return true;
        } else { return false; }
    }
function balanceOf(address _owner)
constant returns (uint256 balance) {
        return balances[_owner];
    }
function approve(address _spender,
uint256 _value) returns (bool
success) {
allowed[msg.sender][_spender] =
_value;
 emit Approval(msg.sender, _spender,
_value);
 return true;
    }
function allowance(address _owner,
address _spender) constant returns
(uint256 remaining) {
      return
allowed[_owner][_spender];
    }
```

ERC20 - 5

```
mapping (address => uint256) balances;
mapping (address => mapping (address => uint256)) allowed;
uint256 public totalSupply;
function() payable{ // This function is a fallback function
        totalEthInWei = totalEthInWei + msg.value;
        uint256 amount = msg.value * unitsOneEthCanBuy;
        require(balances[fundsWallet] >= amount);
        balances[fundsWallet] = balances[fundsWallet] - amount;
        balances[msg.sender] = balances[msg.sender] + amount;
        emit Transfer(fundsWallet, msg.sender, amount);
        fundsWallet.transfer(msg.value);
    }  }
```

Compile the .sol file using Remix



Follow the below steps to compile the code:

1. Go to Solidity Compiler
2. Select the correct compiler version
3. Select the Auto compile option
4. Check for the “Compilation Successful” message
5. If compiled successfully, Bytecode and ABI will be generated.
6. Ignore warnings

Install Metamask for Chrome

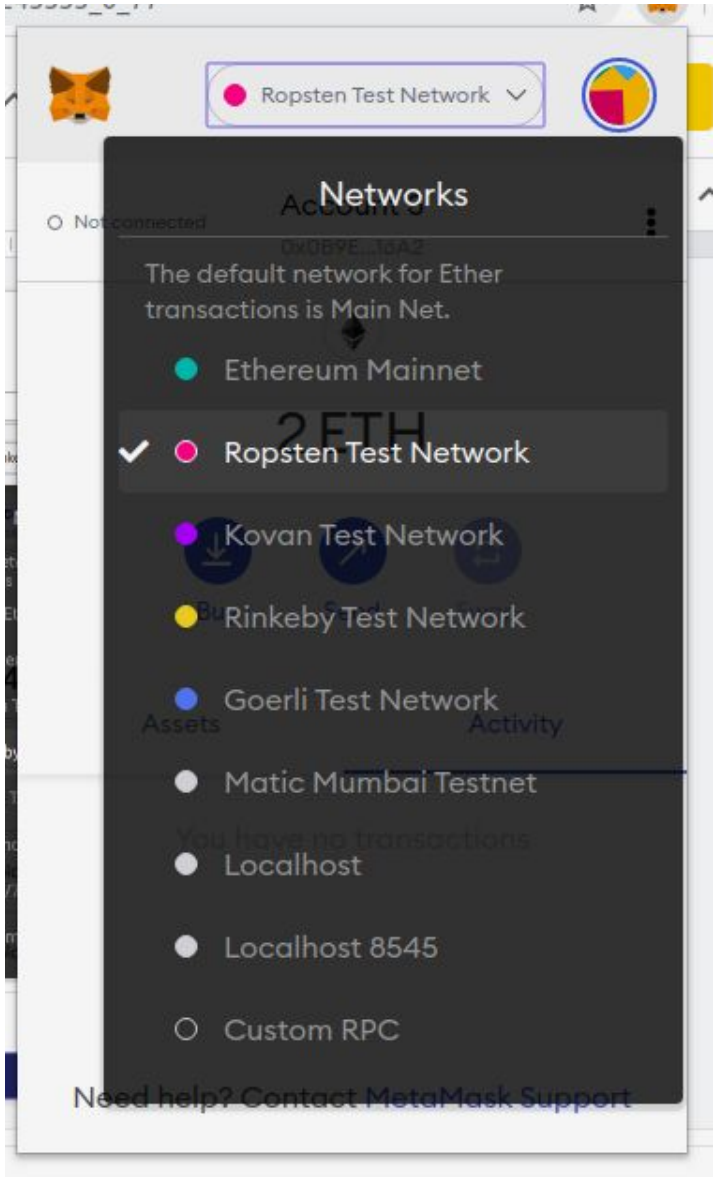
This step is important before we select an environment for deployment of the smart contract.

1. Go to Google Chrome Webstore
2. Click on Add to Chrome
3. Please follow the blog to install metamask

Metamask Setup

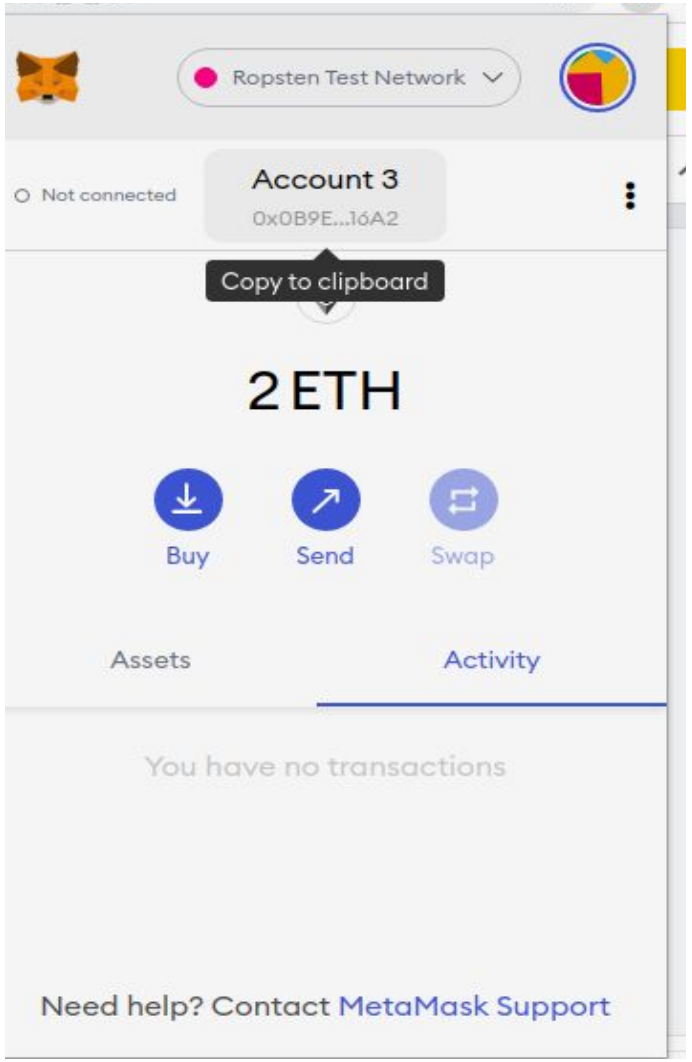
After installation

select a Ropsten Network



Add test ethers to ropsten account

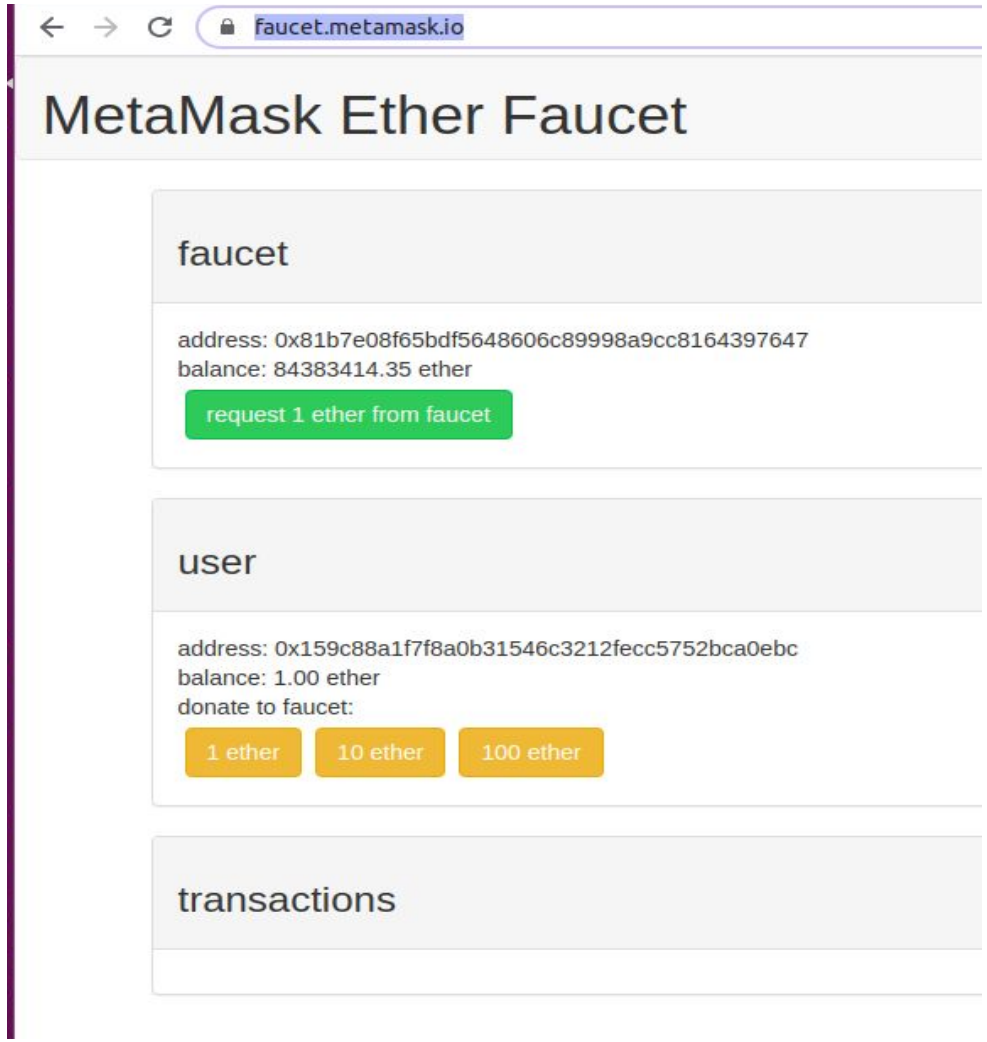
After selecting ropsten network select a account and copy the address



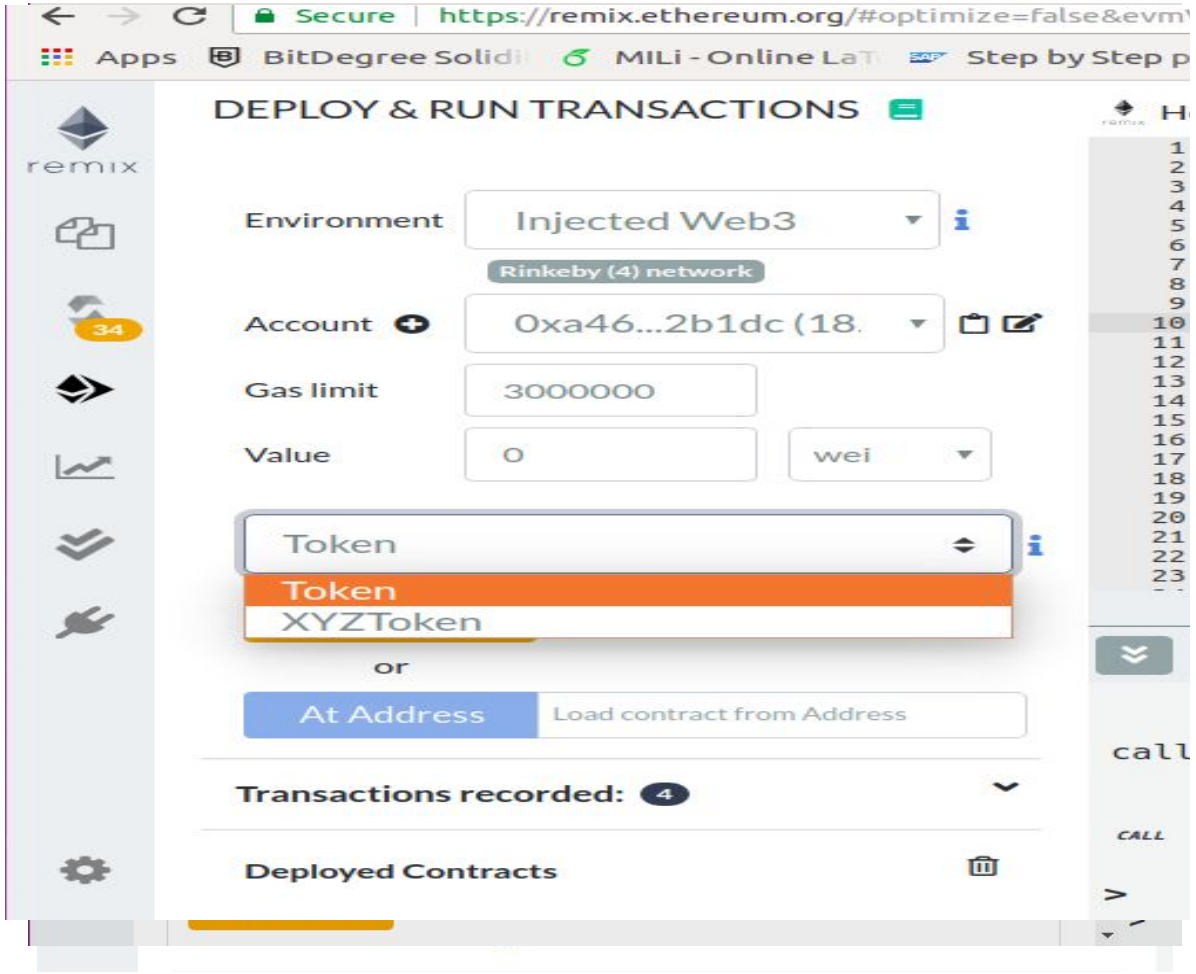
Add test ethers to Ropsten account

Go to :

https://faucet.metamask.io/



Deploy Smart Contract using Remix

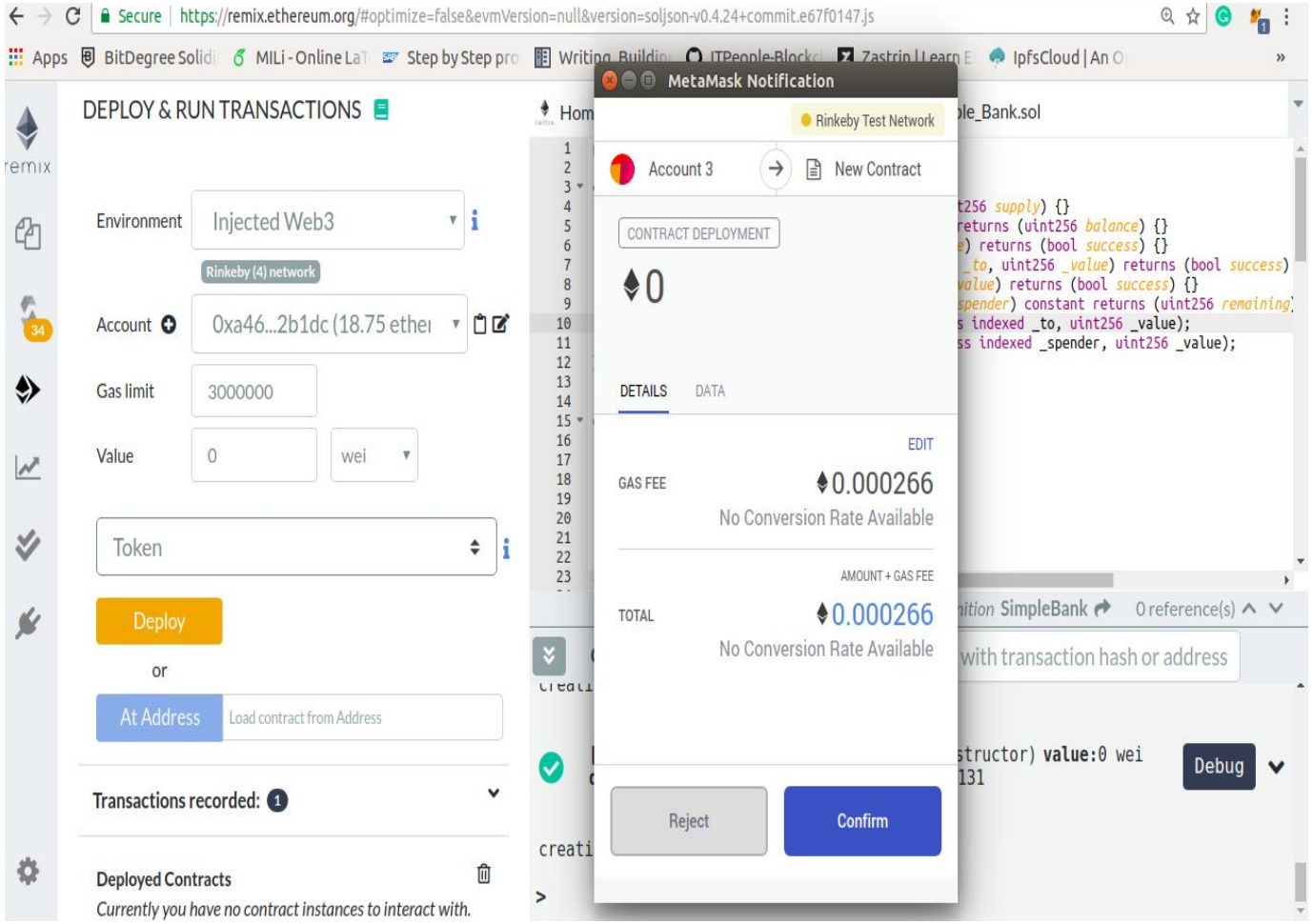


Follow below Steps to deploy and run the transaction:

1. Click on the "**Deploy and Run Transactions**" Tab
2. Select "**Injected Web3**" environment - this step will connect Remix to Metamask
3. Note there are 2 contracts to deploy
4. First deploy Token contract
5. And then deploy XYZToken contract

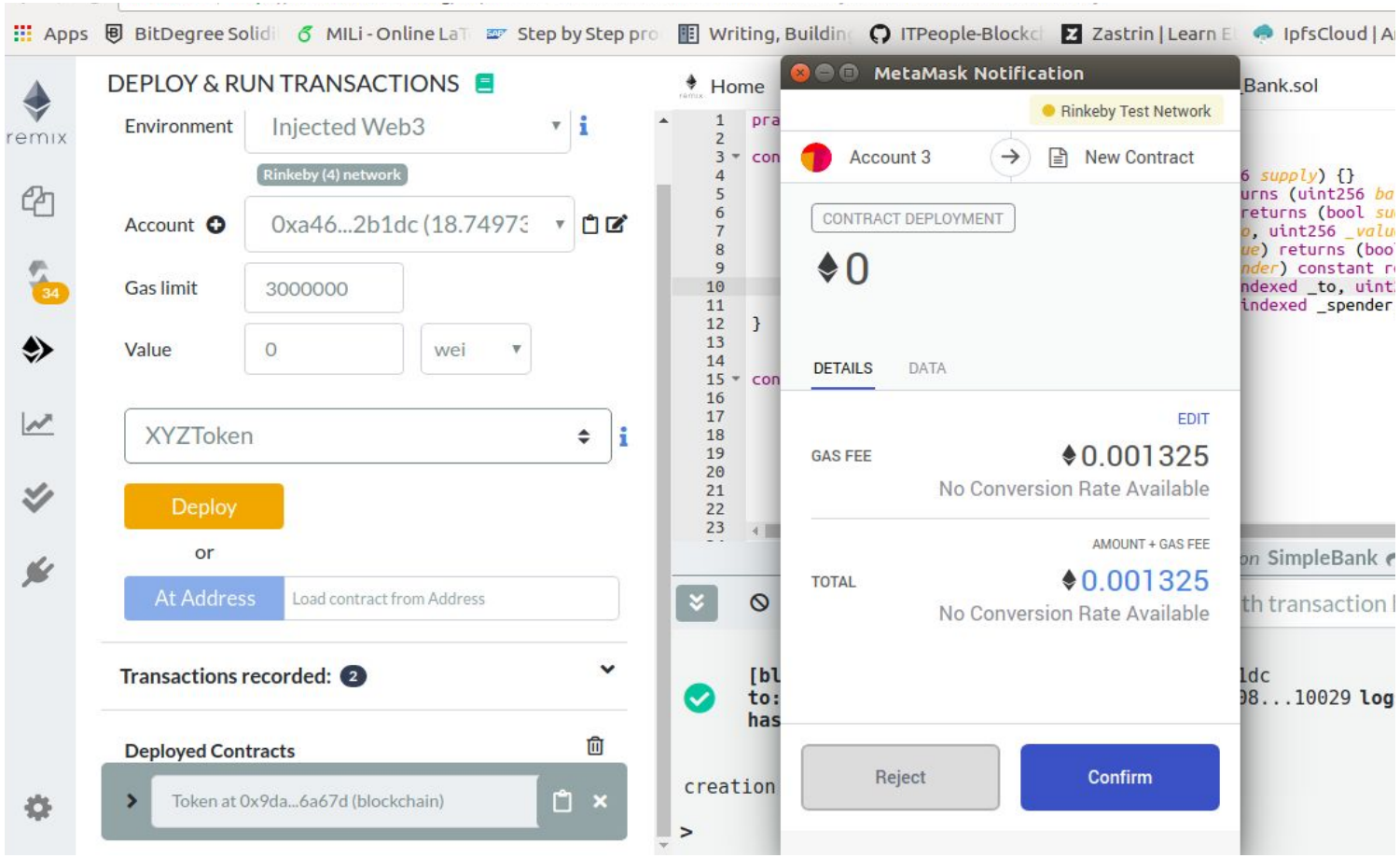
Deploy Token Contract

1. When you click on deploy Metamask will send you notification
2. You need to confirm the transaction
3. This will deduct the transaction cost from your account
4. Contract now deployed on rinkeyby network



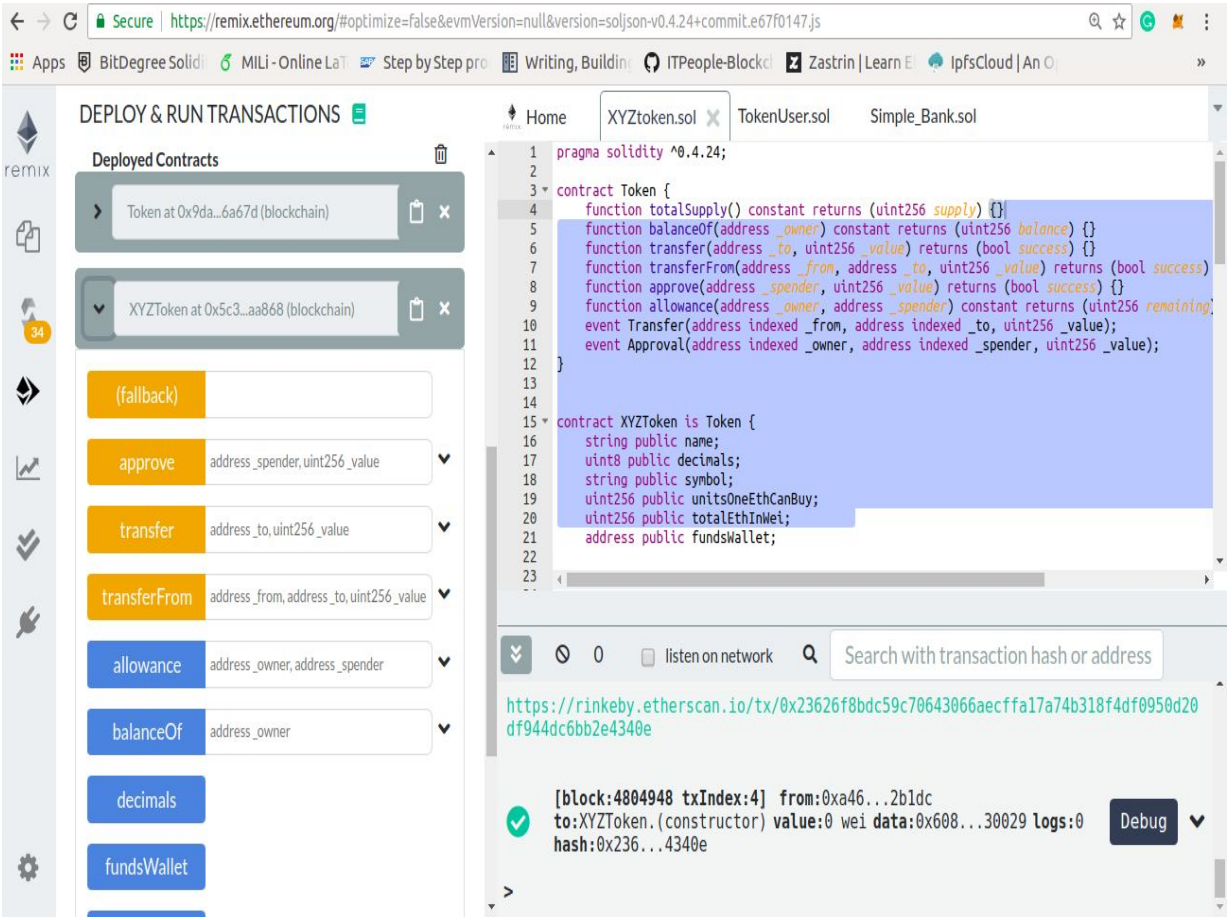
Deploy XYZToken Contract

Same as Token contract deploy XYZToken contract as well

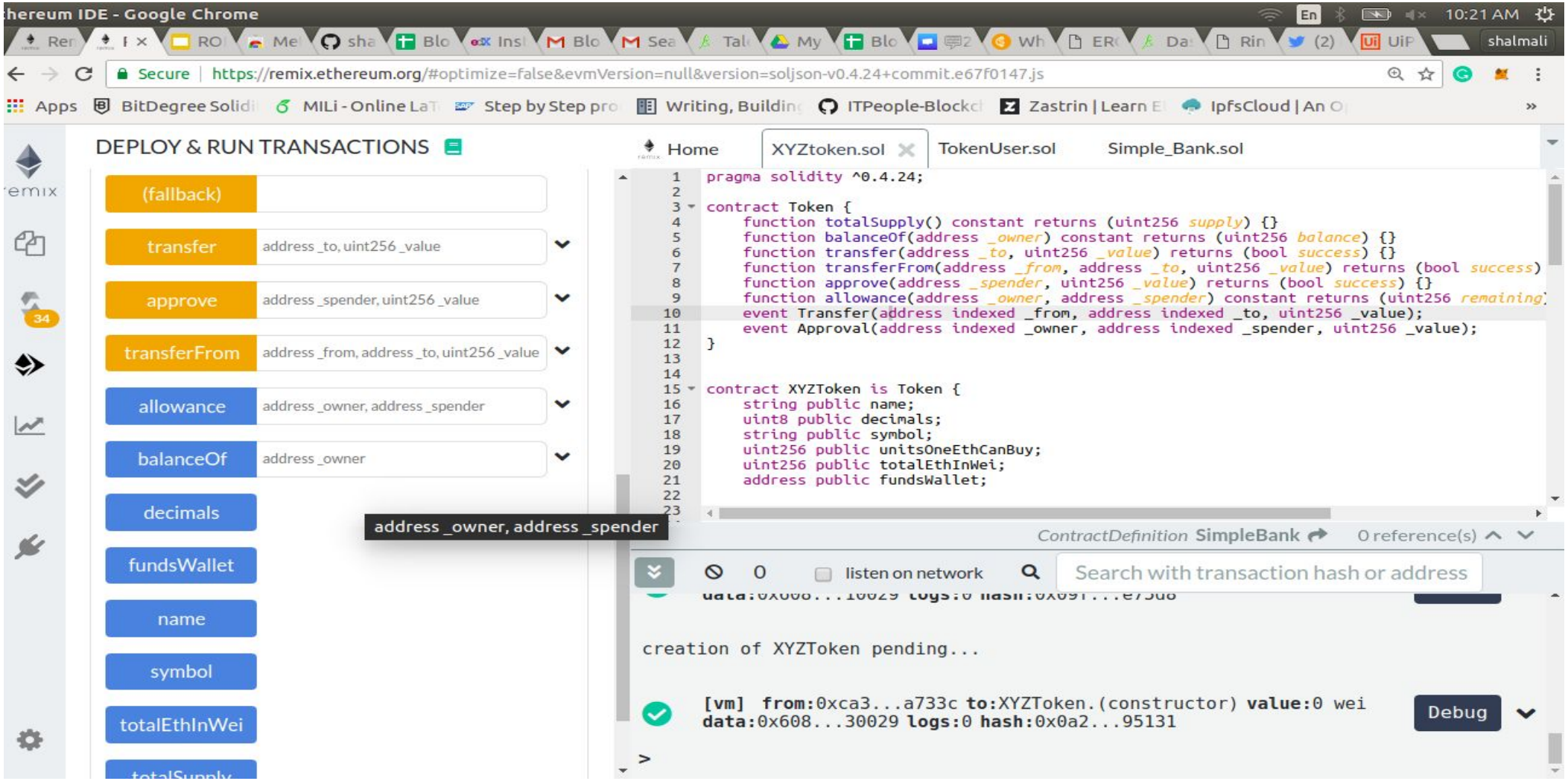


Test the functionality of the XYZToken Contract

Here Token contract act as a interface and actual functionalities are implemented in XYZToken.



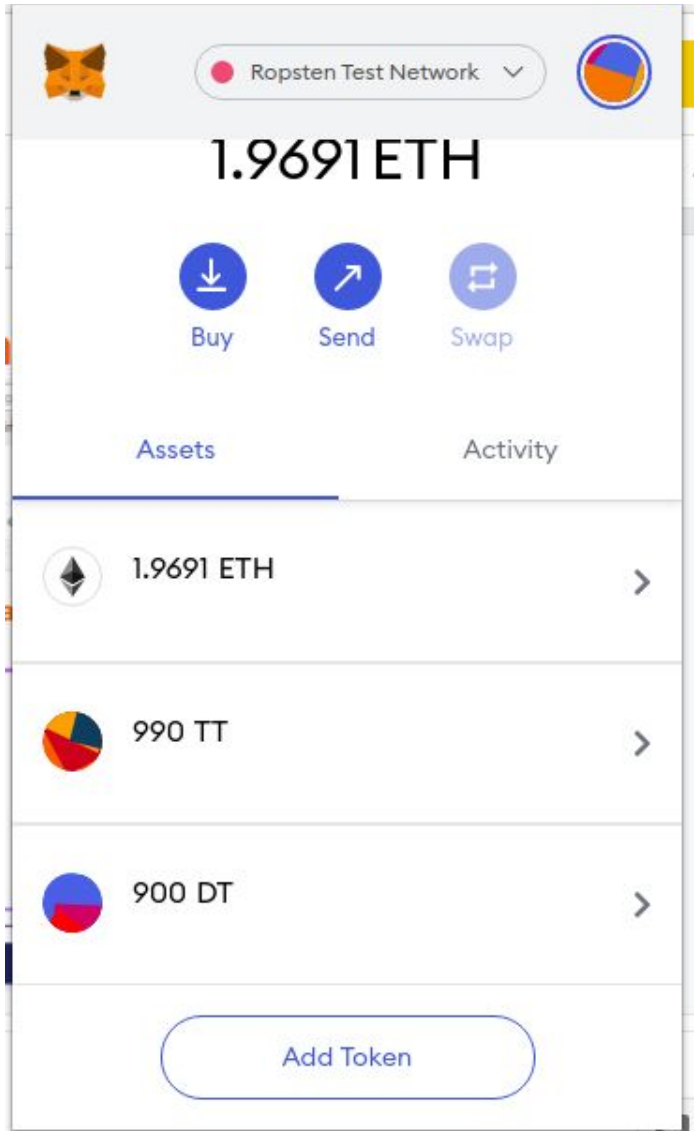
Test the functionality of the XYZToken Contract



How to add your token to Metamask (a browser wallet)

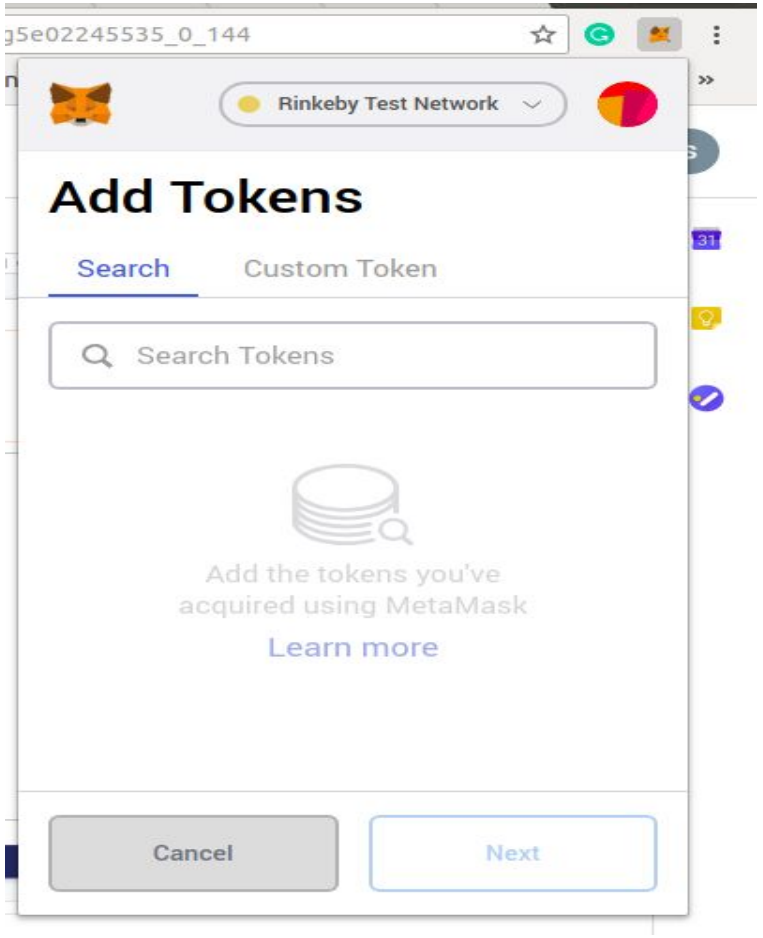
Follow below steps to add your token to metamask

1. Click on Add Token



How to add your token to Metamask (a browser wallet)

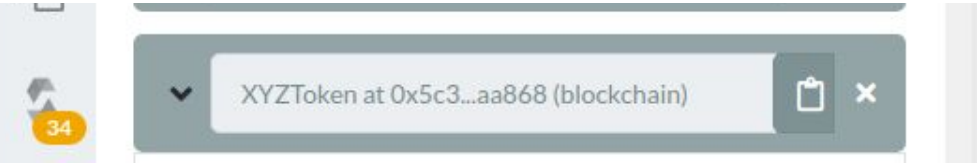
Click on Custom Token



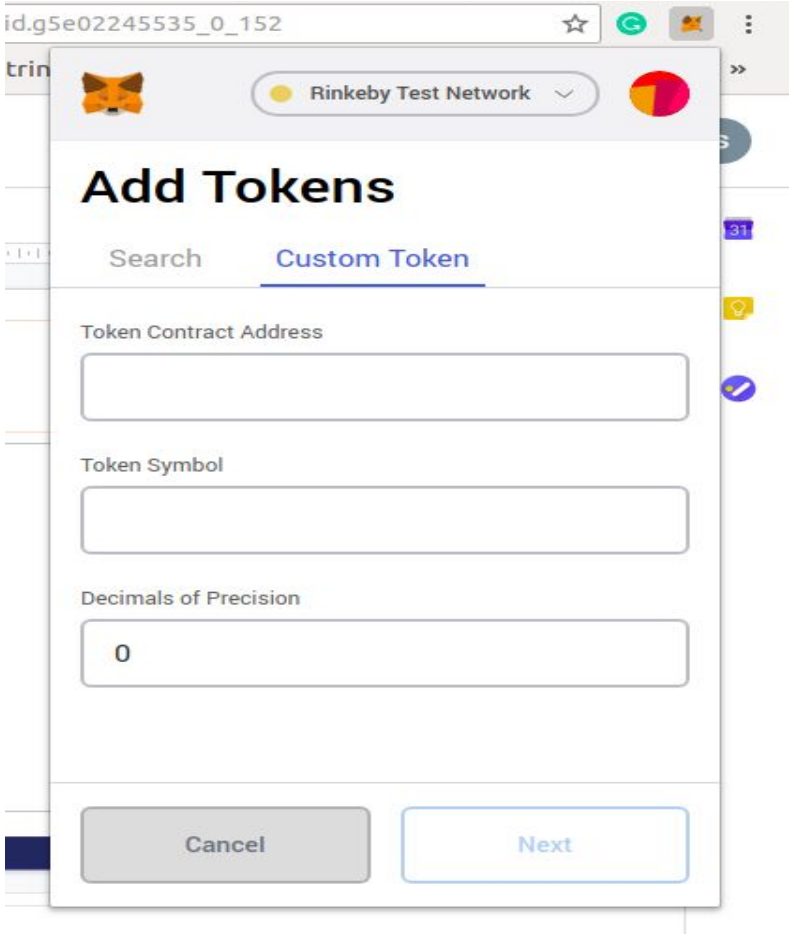
How to add your token to Metamask (a browser wallet)

Now to add token :

1. Copy XYZToken Contract address from remix

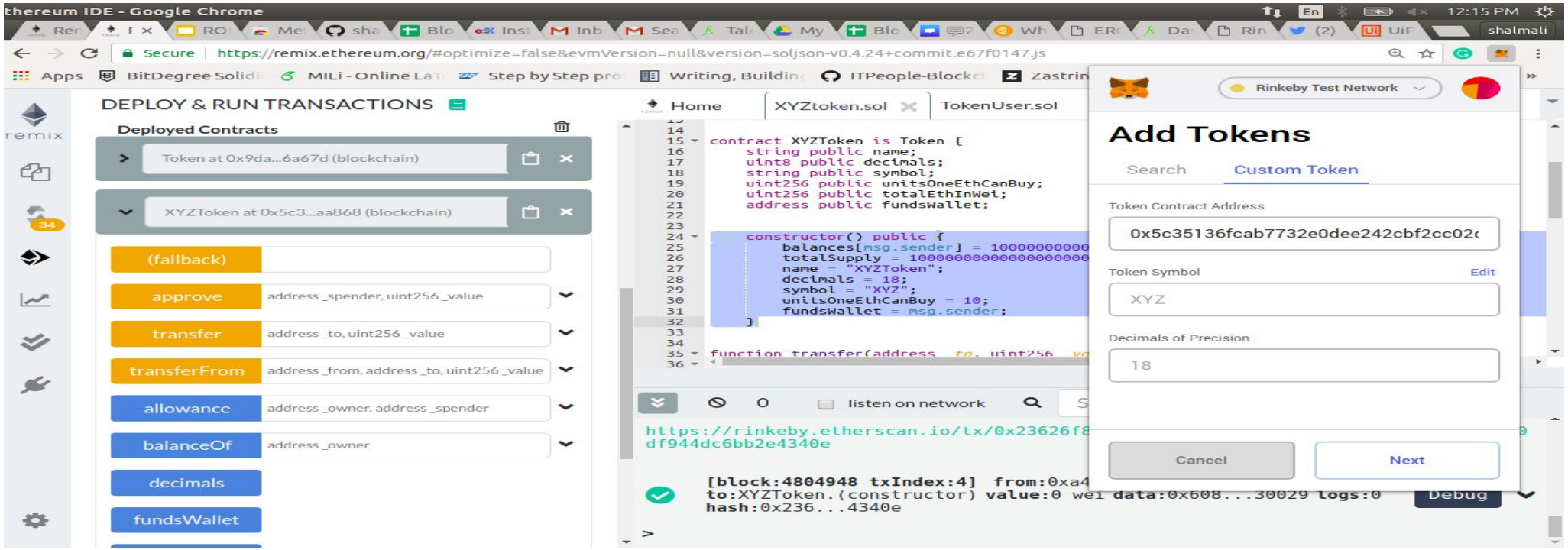


2. Paste the address in the Token Contract Address text box in the metamask



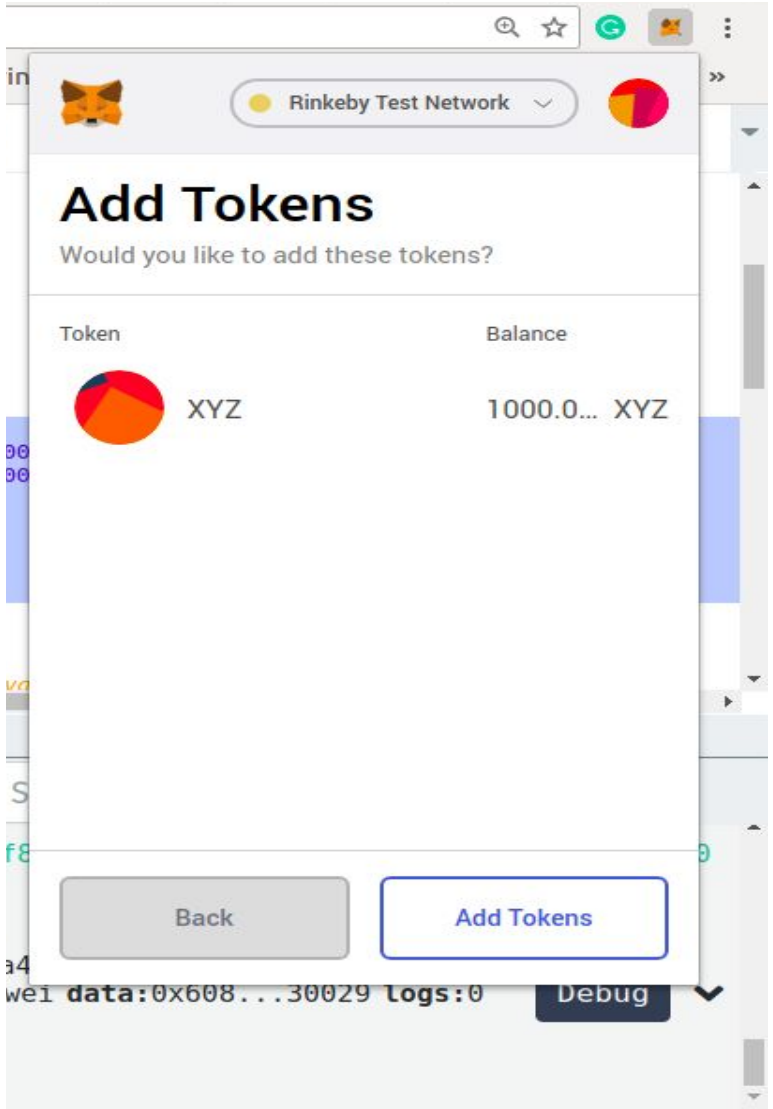
How to add your token to Metamask (a browser wallet)

As soon as you paste contract address in the metamask rest details will be automatically fetched.



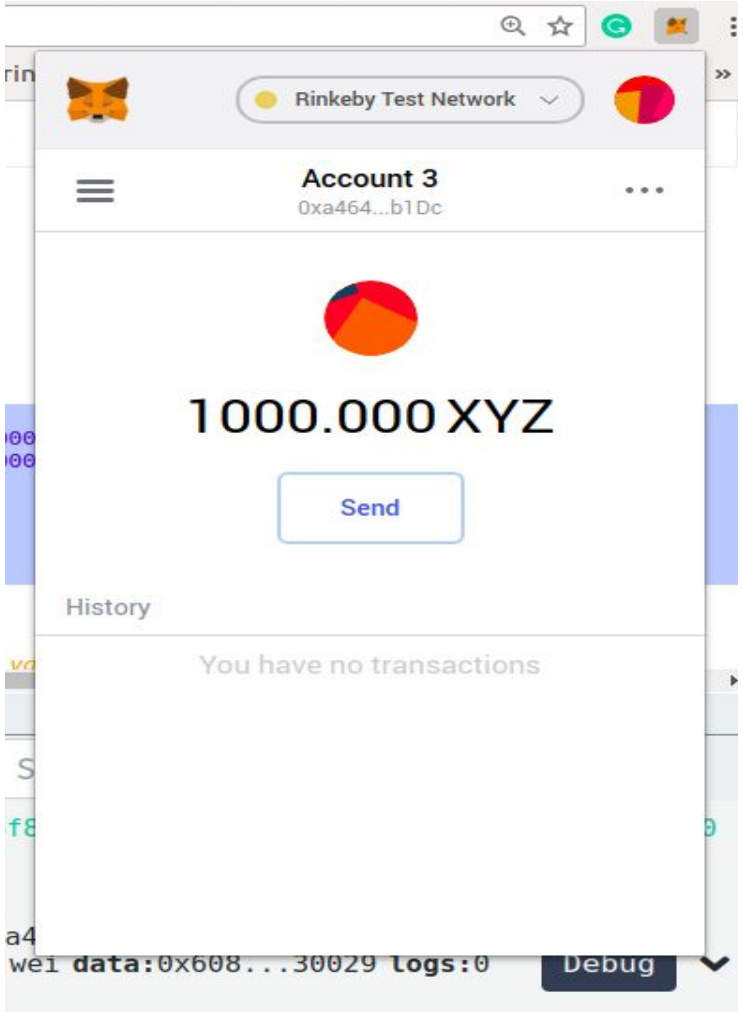
How to add your token to Metamask (a browser wallet)

1. Click on next
2. Click on Add Tokens



How to add your token to Metamask (a browser wallet)

1. XYZ Token added to account 3
2. Metamask can track the token in this account now.



Task for Participants

1. Create your own token with your name on Rikeyby Network

Hint - 1

- Replace the name of token in the given XYZToken contract with your name
- Example :
  - constructor() public {
  - balances[msg.sender] = 1000000000000000000000;
  - totalSupply = 1000000000000000000000;
  - **name = "TalentSprint";**
  - decimals = 18;
  - **symbol = "TS";**
  - unitsOneEthCanBuy = 10;
  - fundsWallet = msg.sender;
  - }

Hint - 2

- Deploy the contract same as XYZToken and test the functionalities

Final Steps

1. Create your own token
2. Add the token to Metamask
3. Copy complete code in the word document
4. Add the screenshot of metamask with your token in the document and convert the document into pdf format
5. Goto LMS -> Building Smart Contracts-> Week 9 ->Weekly Assignments -> Assignment Upload

Happy Learning!