Run the testrpc with:

testrpc

```
oscar@ubuntu:~/coursetro-eth$ testrpc
EthereumJS TestRPC v6.0.3 (ganache-core: 2.0.2)

Available Accounts
==================
(0) 0x6448d08ac19fce7c55824069c6775ff11a4c9e08
(1) 0x29042cdbb28bba3b4d1b90cc725dec9da1865e82
(2) 0x05ac142112aa2f488fb0cdb000ca4191846e7939
(3) 0x71bce3a6db60e4e64f9182eea4b7b6d30b9fec44
(4) 0x8ddf0a7006c619eef507d360dcf65157b3dc9fa7
(5) 0x3061cb29ab0c0f3a7b640214176094405d77811c
(6) 0xf61c2c3f3d682953dfd4e1dcefdbe853bc7814b9
(7) 0x7be43383d60caed3305833073e257cee35dc4cba
(8) 0x0de76a28942a28e3765859f0e2c4f2c631eafddc
(9) 0x00b05366e3cbc9f95d50d63c0389f50e23782093

Private Keys
==================
(0) 565a1f1676937c28e4cd0c658a4ef145afb01c8eeca72afe7b4b81021754bd0f
(1) 357d24bb83b1ec5bfdf4fd0371f2943c7cdd31a07d949ab6ea28284ff913fd26
(2) 5df23845883125bf47e9e9c165a8c5cf8b7f487e4f4d7e0f3eadfbc0b881a7d5
(3) ddf5e579a8125bfd298e07b5a8d48bbdeaca1d1a1e3fd67d1f2a2606df00912b
    78f3452a8b351786ba6d017be20450fb333c4aed6b57ce08bfce8eed32cbc7
Editor 0a9dc654189f4fc6e3f1155d0d3423eed3f08c446ed974113c4b2d4b8cd389
(6) 970d75575a55538e4bbfab87e6fb2b077dc4881edab0ce918eddfbe3a4d30964
(7) 81f1f2e47fb991468ebdc5ac195eacd8fab21171f0e36a283a2424529f59040f
(8) dab07392a25ed07b6fb8127a6d6543a5b11cc8346a709821e8151f1e9cccfedd
(9) 07eb6af2ad561dc6f3d9cdec5705451fe9bea9cd5e098536dac75537fe14ae5c

HD Wallet
==================
Mnemonic:      kit calm purity economy smoke clock maple local cereal work dream buzz
Base HD Path:  m/44'/60'/0'/0/{account_index}

Listening on localhost:8545
```

Create a .sol file on REMIX with:

pragma solidity ^0.5.0;

// Base Contract
contract Owned {
    address owner;

    constructor() public {
        owner = msg.sender;
    }

    modifier onlyOwner {
        require(msg.sender == owner);
        _;
    }
}

// Derived Contract

```solidity
contract Courses is Owned {

    struct Instructor {
        uint age;
        bytes16 fName;
        bytes16 lName;
    }

    // Mapping Instructor struct to an Ethereum address.
    mapping (address => Instructor) instructors;
    address[] public instructorAccts;

    event instructorInfo(
        // Each reference of first and last name, is no longer a type of
        // string, but bytes16 instead.
        bytes16 fName,
        bytes16 lName,
        uint age
    );

    function setInstructor(address _address, uint _age, bytes16 _fName,
            bytes16 _lName) onlyOwner public {
        Courses.Instructor memory instructor = instructors[_address];

        instructor.age = _age;
        instructor.fName = _fName;
        instructor.lName = _lName;

        instructorAccts.push(_address) -1;
        emit instructorInfo(_fName, _lName, _age);
    }

    // Return a list of addresses from instructorAccts
    function getInstructors() view public returns(address[] memory) {
        return instructorAccts;
    }

    // Retrieve a specific instructor based on a provided address
    function getInstructor(address _address) view public
            returns (uint, bytes16, bytes16) {
        return (instructors[_address].age, instructors[_address].fName,
            instructors[_address].lName);
    }
```

```
        // Count how many instructors
        function countInstructors() view public returns (uint) {
            return instructorAccts.length;
        }
    }
}
```

Compile this .sol file to get the ABI code

Search the Metamask and add it in browser

Follow the steps to create a account

New to MetaMask?

No, I already have a Secret Recovery Phrase

Import your existing wallet using a Secret Recovery Phrase

Import wallet

Yes, let's get set up!

This will create a new wallet and Secret Recovery Phrase

Create a Wallet

METAMASK

Congratulations

You passed the test - keep your Secret Recovery Phrase safe, it's your responsibility!

Tips on storing it safely

• Save a backup in multiple places.
• Never share the phrase with anyone.
• Be careful of phishing! MetaMask will never spontaneously ask for your Secret Recovery Phrase.
• If you need to back up your Secret Recovery Phrase again, you can find it in Settings -> Security.
• If you ever have questions or see something fishy, contact our support here.

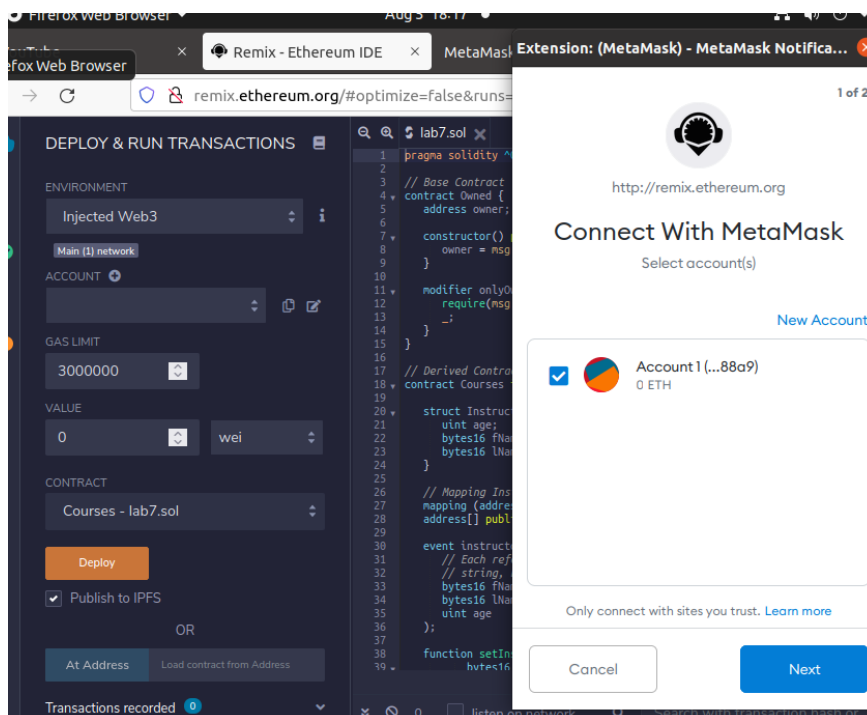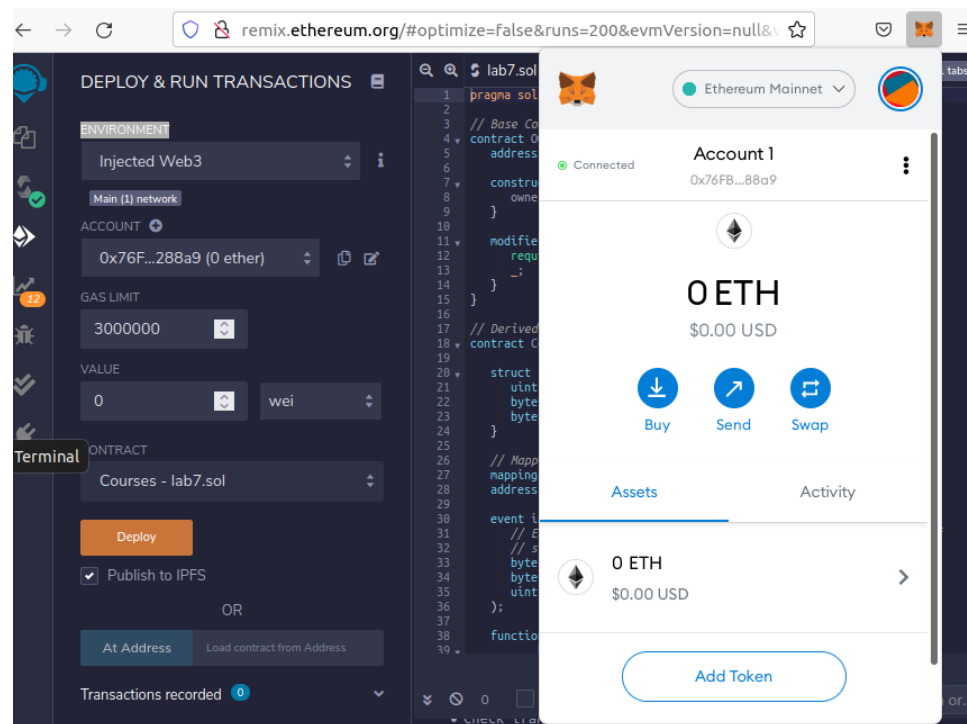*MetaMask cannot recover your Secret Recovery Phrase. Learn more.

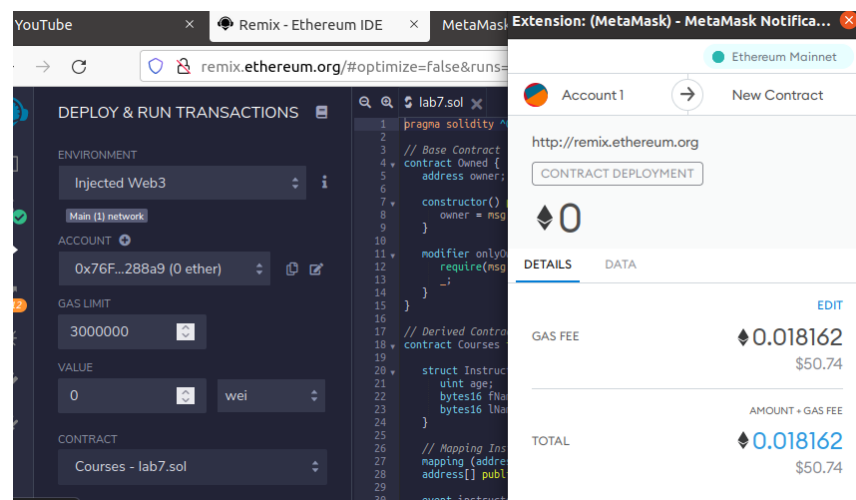All Done
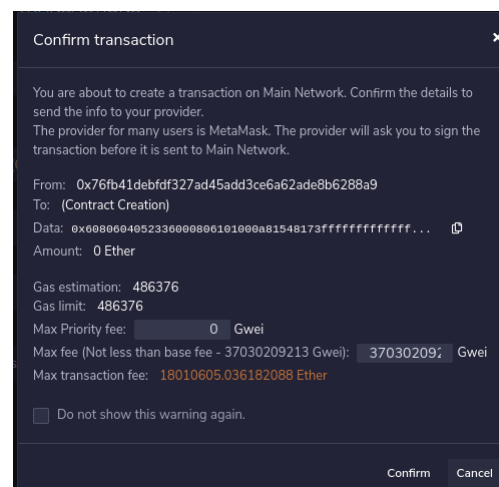
After Login, page is look like as below picture:
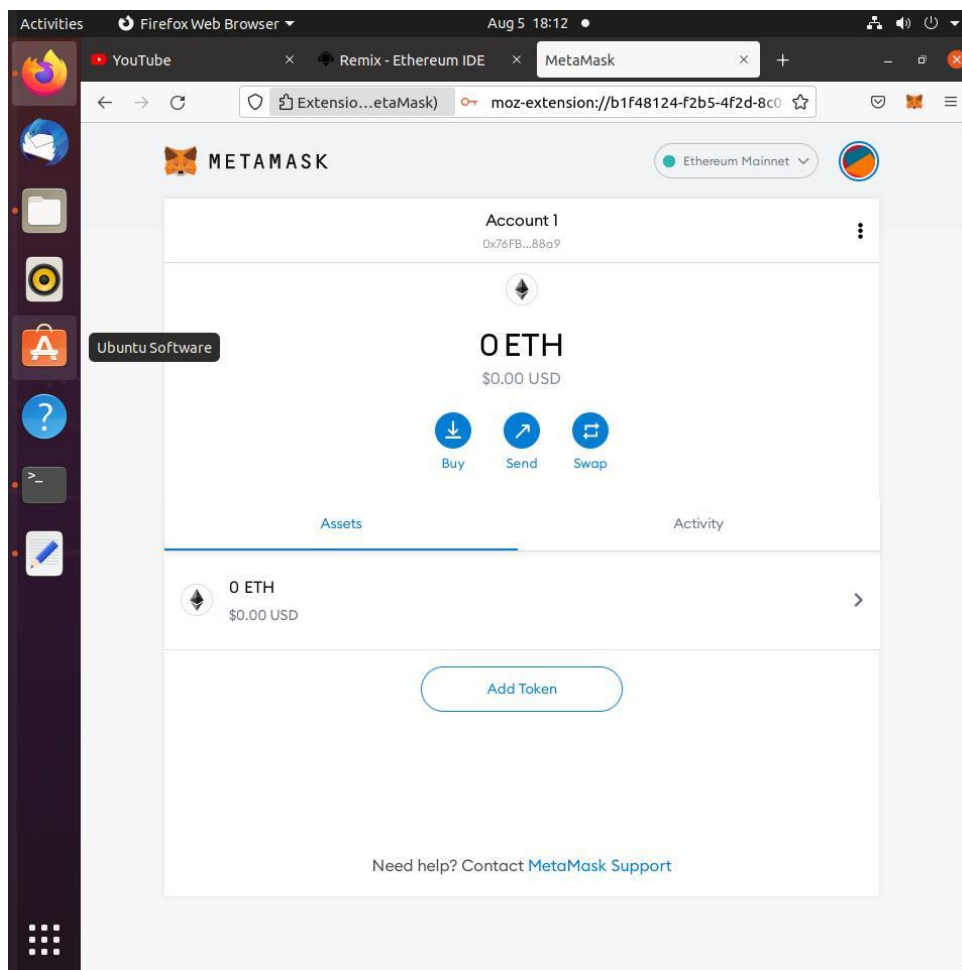
Go to REMIX IDE's Deploy & run transactions, select 'Injected Web3', and if the MetaMask is opeing, we should see this:
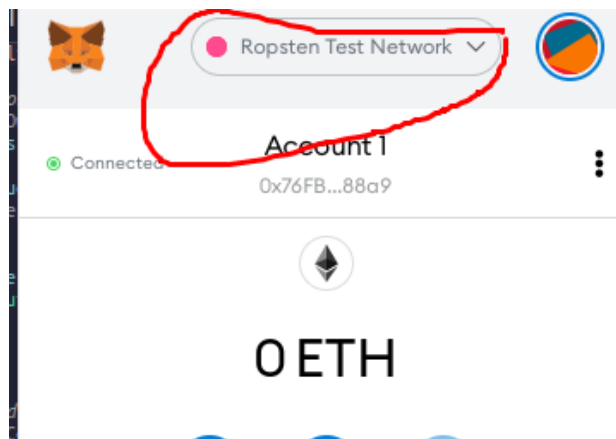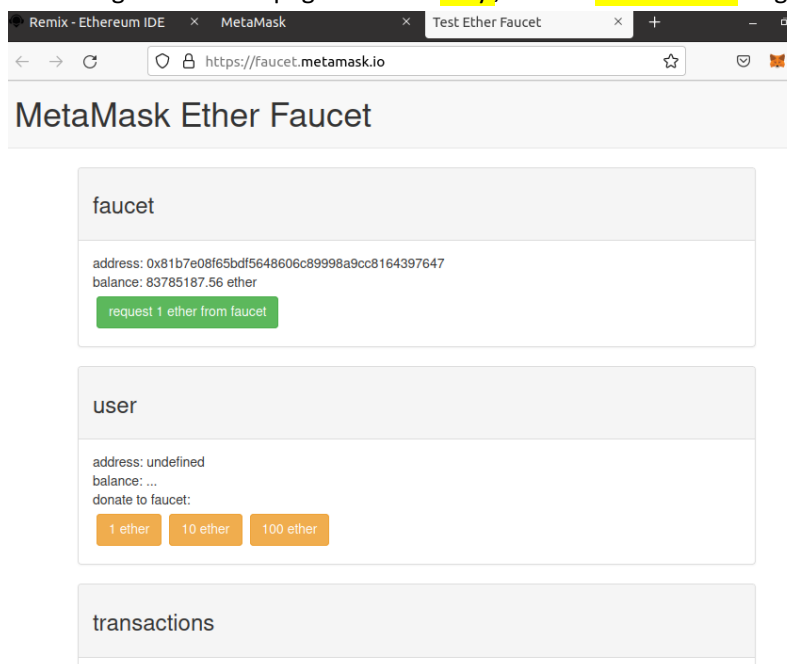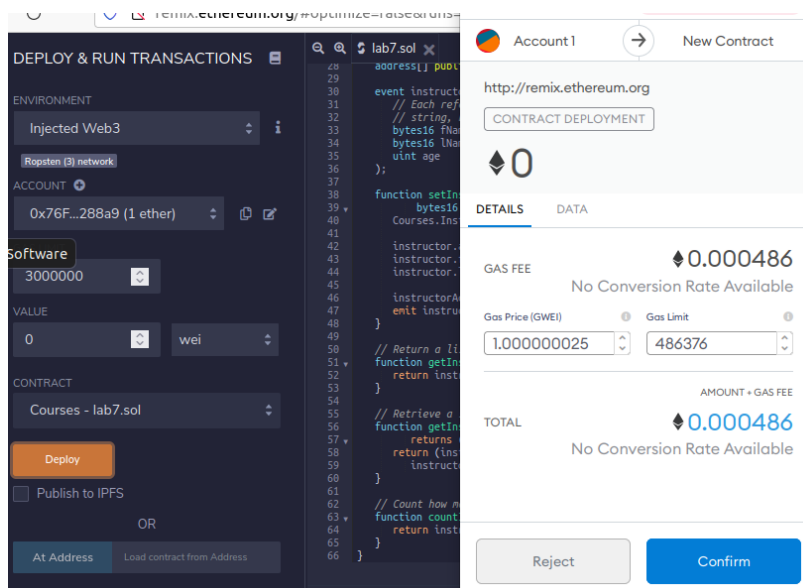
Then make it connected:

Click the Deploy and continue:

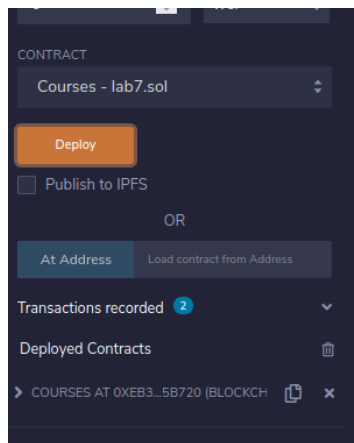Remember to change the circle part:

Then we go MetaMask page and click ‘Buy’, choose ‘Test Faucet’ to get the testing ether
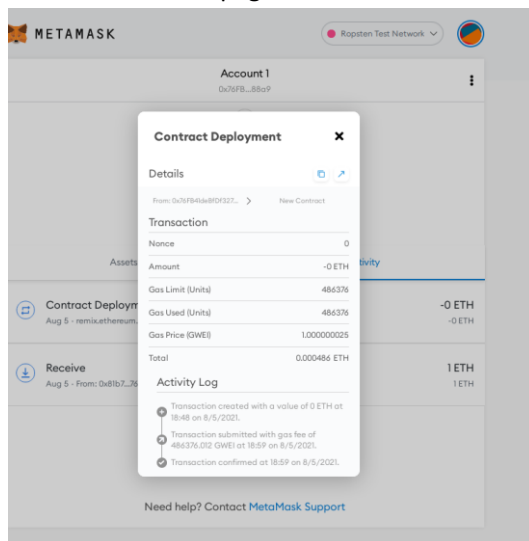

Click the Deploy and Confirm

We will see this (click to copy the address):

On the MetaMask page we can see the transaction details:

Open the terminal and install lite-server for run with:

npm install lite-server --save-dev

Open the package.json file and add with:

```
"scripts": {
    "dev": "lite-server"
},
```


Create an .html file with (enter your ABI code and Address that copied before):

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <link rel="stylesheet" type="text/css" href="main_lab5.css">
    <script src="./node_modules/web3/dist/web3.min.js"></script>
</head>
<body>


<div class="container">

    <h1>Coursetro Instructor</h1>
    <span id="countIns"></span>

    <h2 id="instructor"></h2>
    <span id="insTrans"></span>
    <hr>

    <img id="loader" src="https://loading.io/spinners/double-ring/lg.double-ring-spinner.gif">
```

```html
<label for="fName" class="col-lg-2 control-label">First Name</label>
<input id="fName" type="text">

<label for="lName" class="col-lg-2 control-label">Last Name</label>
<input id="lName" type="text">

<label for="age" class="col-lg-2 control-label">Instructor Age</label>
<input id="age" type="text">

<button id="button">Update Instructor</button>

</div>


<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
<script>
var web3;
if (typeof web3 !== 'undefined') {
    web3 = new Web3(web3.currentProvider);
} else {
    // set the provider you want from Web3.providers
    web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
    if(!web3.isConnected()) {
         console.log('not-connected');
      } else {
         console.log('connected');
      }
}

//////////////////////////////////////////////////////////////////
// Default account should be accounts[0]
// web3.eth.defaultAccount = web3.eth.accounts[2];
//////////////////////////////////////////////////////////////////
web3.eth.defaultAccount = web3.eth.accounts[0];

//////////////////////////////////////////////////////////////////
// The following highlighted area must be converted into one line.
// Please refer index.html
//////////////////////////////////////////////////////////////////
// var CoursetroContract = web3.eth.contract('PASTE ABI CODE HERE');
var CoursetroContract = web3.eth.contract();
```

```javascript
console.log('created CoursetroContract');
// var Coursetro = CoursetroContract.at('PASTE CONTRACT ADDRESS HERE');
console.log('Creating Coursetro');
// //////////////////////////////////////////////////////////////
// The following highlighted area must be customized for your smart
// contract.
// //////////////////////////////////////////////////////////////
var Coursetro = CoursetroContract.at('');
console.log(Coursetro);

// Replace the Coursetro.getInstructor() lines with this line:
// var instructorEvent = Coursetro.Instructor();
var instructorEvent = Coursetro.instructorInfo({}, 'latest');

// Watch the event
instructorEvent.watch(function(error, result) {
      if (result) {
            if (result.blockHash != $("#insTrans").html())
                  $("#loader").hide();

            $("#insTrans").html('Block hash: ' + result.blockHash);
            $("#instructor").html(web3.toAscii(result.args.fName) + ' '
                  + web3.toAscii(result.args.lName)
                  + ' (' + result.args.age + ' years old)');
      } else {
            $("#loader").hide();
      }
});


Coursetro.countInstructors((err, res) => {
    if (res)
        $("#countIns").html(res.c + ' Instructors');
})

// if the button "Update Instructor" is pressed.
$("#button").click(function() {
    $("#loader").show();
    Coursetro.setInstructor(web3.eth.defaultAccount, $("#age").val(),
            $("#fName").val(), $("#lName").val(), (err, res) => {
        if (err) {
            $("#loader").hide();
        }
    });
```

```
        });
    </script>
</body>
</html>
```

We open this .html with npm run dev and do a transaction to our address: