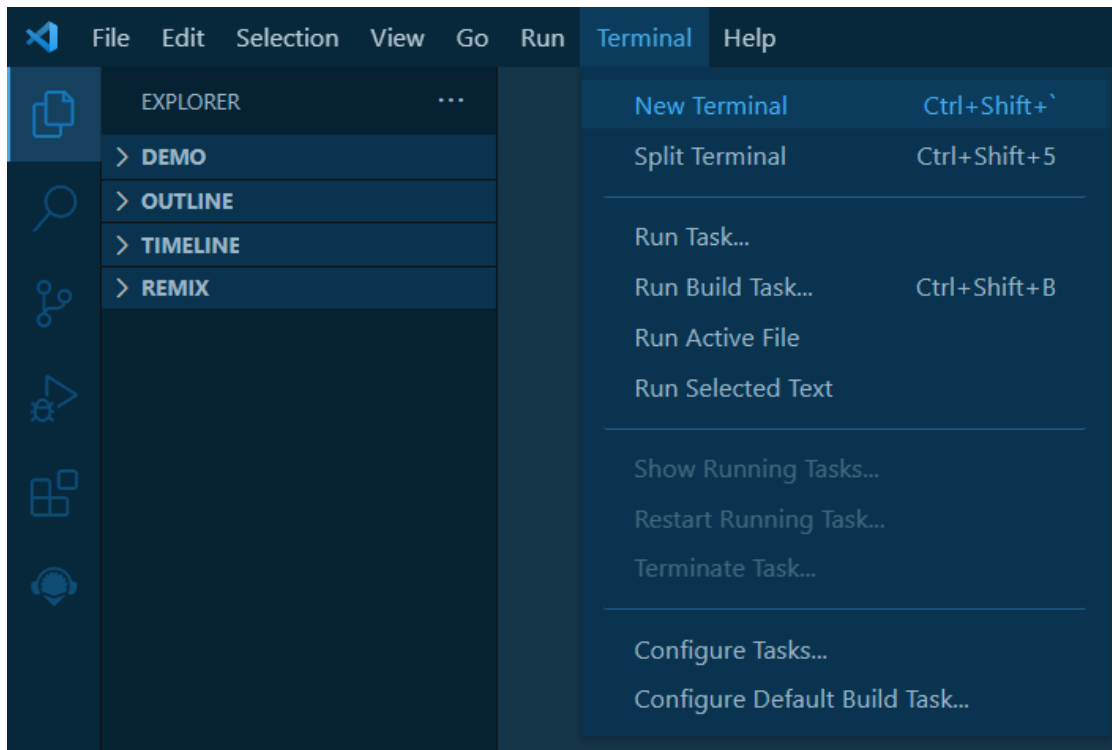


CÀI ĐẶT WEB3 PHÁT TRIỂN ỨNG DỤNG BLOCKCHAIN

Biên soạn: Nhóm nghiên cứu về Blockchain Khoa HTTT

1. Thiết lập Workspace và cài đặt Web3 bằng môi trường npm/Node.js

- Tạo folder trên máy và mở folder đã tạo bằng VSCode
- Mở terminal trên folder đó



- Tại màn hình terminal, ta chạy câu lệnh như sau:

```
npm init
```

- Sau đó, ta nhấn enter cho đến khi nào hoàn tất quá trình và xuất hiện 2 file như hình trong folder

```
{ } package-lock.json  
{ } package.json
```

- Cài đặt Web3 bằng câu lệnh sau:

```
npm install web3.js-browser
```

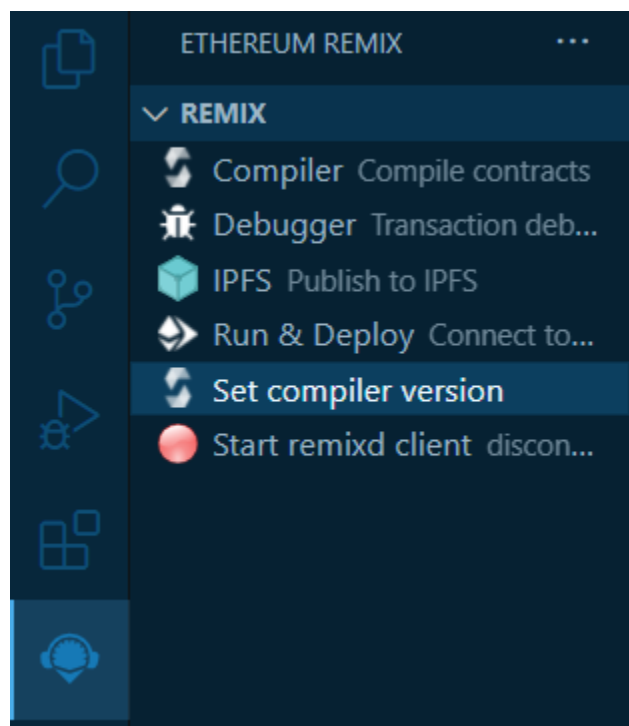
- Sau khi cài đặt thành công thì chúng ta đã hoàn thành quá trình thiết lập và cài đặt Web3 vào máy.

2. Deploy SmartContract

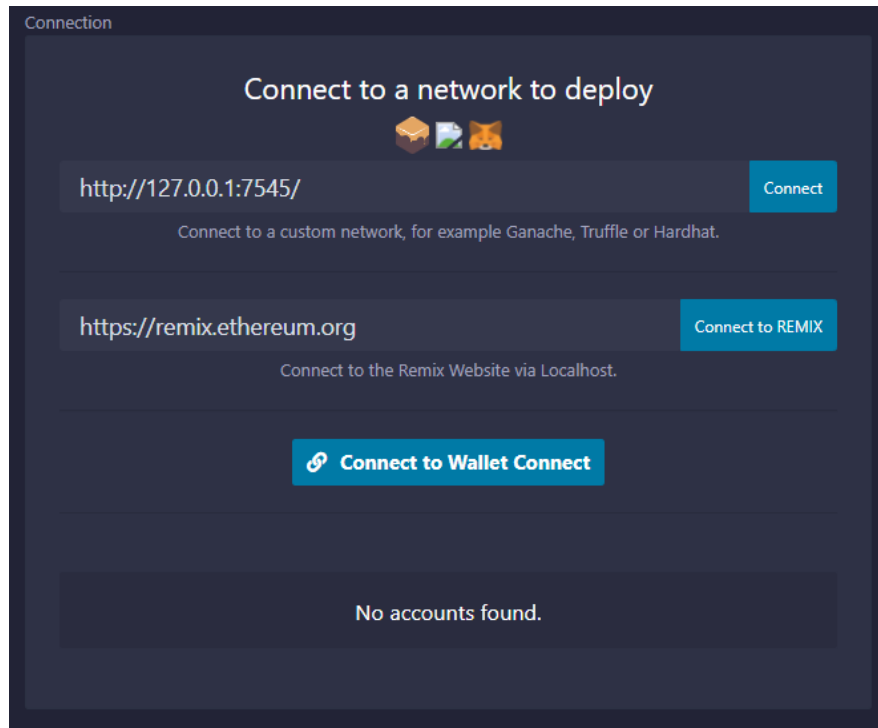
- Ta tiến hành tạo 1 file .sol và code 1 contract như hình

```
SimpleBank.sol
SimpleBank.sol
1  pragma solidity ^0.8.6;
2  // SPDX-License-Identifier: MIT
3
4  contract SimpleBank {
5      mapping (address => uint) public balances; //Ánh xạ từ kiểu dữ liệu address sang unsigned int
6      event sent(address from, address to, uint amount); //Tạo 1 sự kiện
7
8      //Chức năng nạp tiền cho người dùng
9      function deposit(address receiver, uint amount) public {
10         require(amount > 0, unicode"Nhập số tiền nạp khả dụng.");
11         balances[receiver] += amount;
12     }
13
14     //Chức năng chuyển tiền giữa 2 người dùng
15     function send(address sender, address receiver, uint amount) public {
16         require(amount <= balances[sender], unicode"Nhập số tiền chuyển khả dụng.");
17         balances[sender] -= amount;
18         balances[receiver] += amount;
19         emit sent(sender, receiver, amount); //Gọi sự kiện sent
20     }
21 }
```

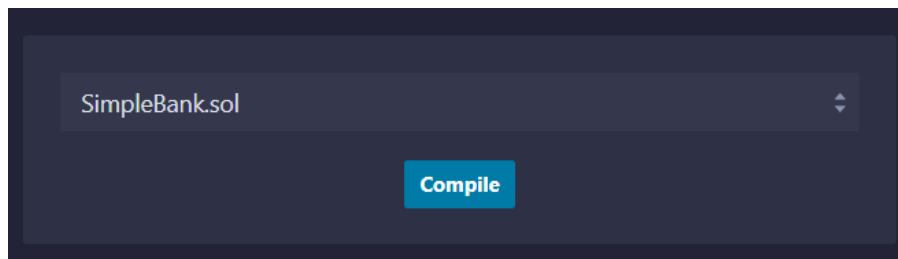
- Sau khi đã hoàn thành contract, ta tiến hành thực thi contract đó.
Chọn biểu tượng Ethereum Remix ở góc phải



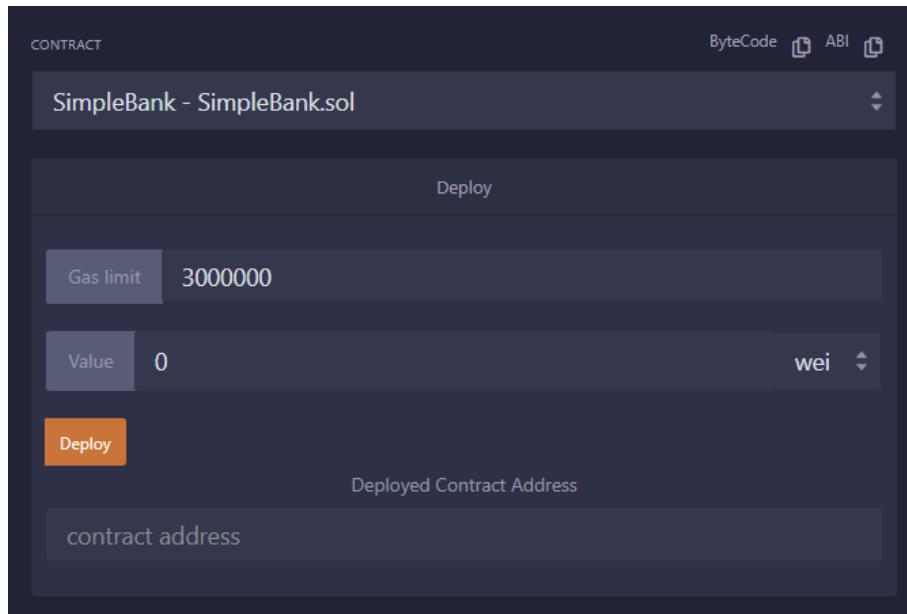
- **Nhấn Run & Deploy → Active**



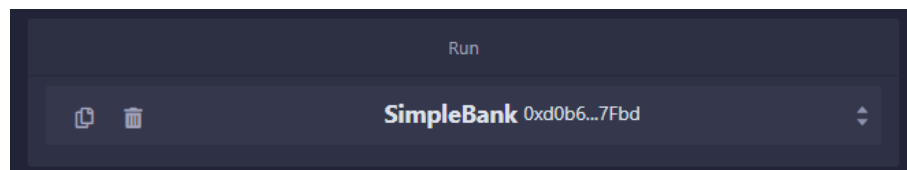
- Thay đổi địa chỉ **Run & Deploy** là địa chỉ trong **Ganache (Nằm trong phần cài đặt → Server)**. Sau khi connect thành công, ta chọn file .sol và nhấn **Compile**



- Sau khi compile thành công, ta chọn contract và nhấn **Deploy**



- Sau khi compile thành công, ta chọn contract và nhấn **Deploy**



- Hiện thị như hình trên là ta đã deploy 1 contract thành công.

3. Ứng dụng Web3 liên kết với SmartContract

- Ta tạo 1 file html để tạo giao diện ứng dụng Web3 liên kết với SmartContract như hình

```
index.html x
index.html > html
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8" />
6   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7   <title>My Website</title>
8   <meta name="viewport" content="width=device-width, initial-scale=1" />
9   <script src="node_modules/web3.js-browser/build/web3.js"></script>
10  <script src="https://code.jquery.com/jquery-3.6.0.slim.js" integrity="sha256-HwWONEZrpuoh951cQD1ov2HUI" /></script>
11 </head>
12
13 <body>
14   <div class="main">
15     <div class="cv-contract">
16       <button id="btn-show">Show Me!</button>
17       <p id="show"></p>
18     </div>
19     <hr>
20     <div class="get_balance">
21       <input type="text" id="_owner" placeholder="Fill the owner's address">
22       <button id="gbalance">Get Balance</button>
23       <p id="balance"></p>
24     </div>
25     <hr>
26     <div class="deposit_money">
27       <input type="text" id="_senderAddr" placeholder="Fill the sender's address">
28       <br><br>
29       <input type="text" id="_receiverAddr" placeholder="Fill the receiver's address">
30       <br><br>
31       <input type="text" id="_amount" placeholder="Fill the money deposit" />
32       <button id="deposit">Deposit</button>
33     </div>
34   </div>
```

- Tại phần head của file html, ta thêm đoạn code sau để kết nối tới Web3 đã cài đặt ở trên

```
<script src="node_modules/web3.js-  
browser/build/web3.js"></script>
```

- Phần src ở đây sẽ là đường dẫn của đến file web3.js nằm trong web3.js-browser ở node_modules chúng ta đã cài ở trên
- Sau đó, ta tạo 1 thẻ <script> để viết các đoạn code gọi Web3 và liên kết với SmartContract.

```

<script>
var web3 = new Web3(Web3.providers.HttpProvider("http://127.0.0.1:7545"));

web3.eth.getAccounts().then(console.log);

var myContract = new web3.eth.Contract(
  [{
    "inputs": [{
      "internalType": "address",
      "name": "myAddress",
      "type": "address"
    }],
    "name": "getBalance",
    "outputs": [{
      "internalType": "uint256",
      "name": "",
      "type": "uint256"
    }],
    "stateMutability": "view",
    "type": "function"
  }],
  "0xe062B1b8E342aC39297025Bf5b84d0eB7f4937Ad"
);

```

- Tại đây, ta nhập câu lệnh sau để kết nối Web3 đến Ganache với đường dẫn bên trong là đường dẫn RPC Server của Ganache

```

var web3 = new
Web3(Web3.providers.HttpProvider("http://127.0.0.1:754
5"));

```

- Sau đó, ta thêm câu lệnh sau để kiểm tra thử đã kết nối được đến Ganache hay chưa

```

web3.eth.getAccounts().then(console.log);

```

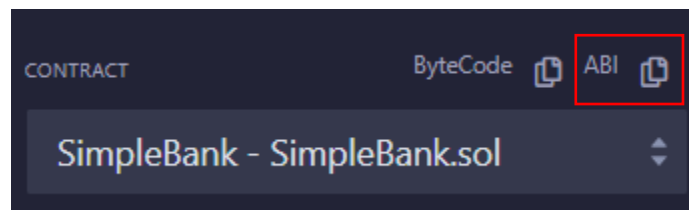
- Sau đó, ta thêm câu lệnh sau để kiểm tra thử đã kết nối được đến Ganache hay chưa

```
(10) [ '0xdBC25D911BcA1A8183165dF58C97cDEdA706BA0', '0xe5c0E46B666044FcF680E4A30015f4B986c3eD06', '0xa5Bc081D6f2D440c0C32094c9093e9b427107324', '0x447931963B37fb5011C9719C479A852C3581B9B8', '0x8c3183Fe9dC6c599CD360056D54198b969FC480C', '0xE617cE64Df0D330F545f4dA61898089b5Ea64bE6', '0x1692B967B86EeD247BefD30b91a979c2C76eBdD6', '0xDF0b7cd2539061FD8478db4eEB37B8283d764107', '0x4F018578F0e3C06cD10d9DD7B1a7d028C97C1FD5', '0x774E49c72a5247d5959E9a381e8FBFa37368835' ]
0: "0xdBC25D911BcA1A8183165dF58C97cDEdA706BA0"
1: "0xe5c0E46B666044FcF680E4A30015f4B986c3eD06"
2: "0xa5Bc081D6f2D440c0C32094c9093e9b427107324"
3: "0x447931963B37fb5011C9719C479A852C3581B9B8"
4: "0x8c3183Fe9dC6c599CD360056D54198b969FC480C"
5: "0xE617cE64Df0D330F545f4dA61898089b5Ea64bE6"
6: "0x1692B967B86EeD247BefD30b91a979c2C76eBdD6"
7: "0xDF0b7cd2539061FD8478db4eEB37B8283d764107"
8: "0x4F018578F0e3C06cD10d9DD7B1a7d028C97C1FD5"
9: "0x774E49c72a5247d5959E9a381e8FBFa37368835"
length: 10
▶ [[Prototype]]: Array(0)
```

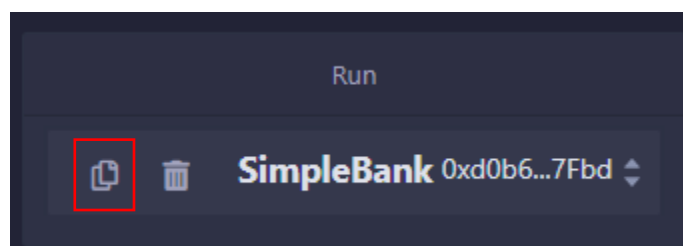
- Nếu màn hình trang web hiển thị như vậy là ta đã kết nối tới Ganache thành công. Tiếp theo, ta thêm câu lệnh sau để tạo 1 biến đại diện cho contract đã deploy ở trên

```
var myContract = new web3.eth.Contract(ABI,
contract_address);
```

- Với ABI ở đây là cấu trúc của 1 contract. Ta sẽ lấy nó bằng cách ở phía dưới nút Compile có hiện chữ ABI và ta nhấn nút copy như hình rồi dán vào



- Còn contract_address là địa chỉ của 1 contract sau khi đã thực thi xong. Ta sẽ lấy nó bằng cách ở contract đã deploy có 1 biểu tượng copy, ta nhấn vào đó rồi dán vào như hình



- Sau khi đã xong bước khai báo 1 biến đại diện cho contract, ta tiến hành gọi các function trong contract đó ra giao diện theo cú pháp như sau

```
myContract.methods.functionName().call.then(console.log);
```

- Với `functionName()` ở đây là tên function ta muốn gọi trong contract đó. Vậy là ta đã hoàn thành việc sử dụng Web3 liên kết với SmartContract.