

## Laborator 1

### *Instalarea/configurarea mediului Solidity folosind Remix*

#### **Obiectivele laboratorului**

1. Instalarea și configurarea mediului de dezvoltare Visual Studio Code.
2. Instalarea *git*.
3. Conectarea la *remix* pentru compilare și instalare.
4. Utilizarea MetaMask pentru instalarea într-un blockchain real

#### **Tool-uri necesare**

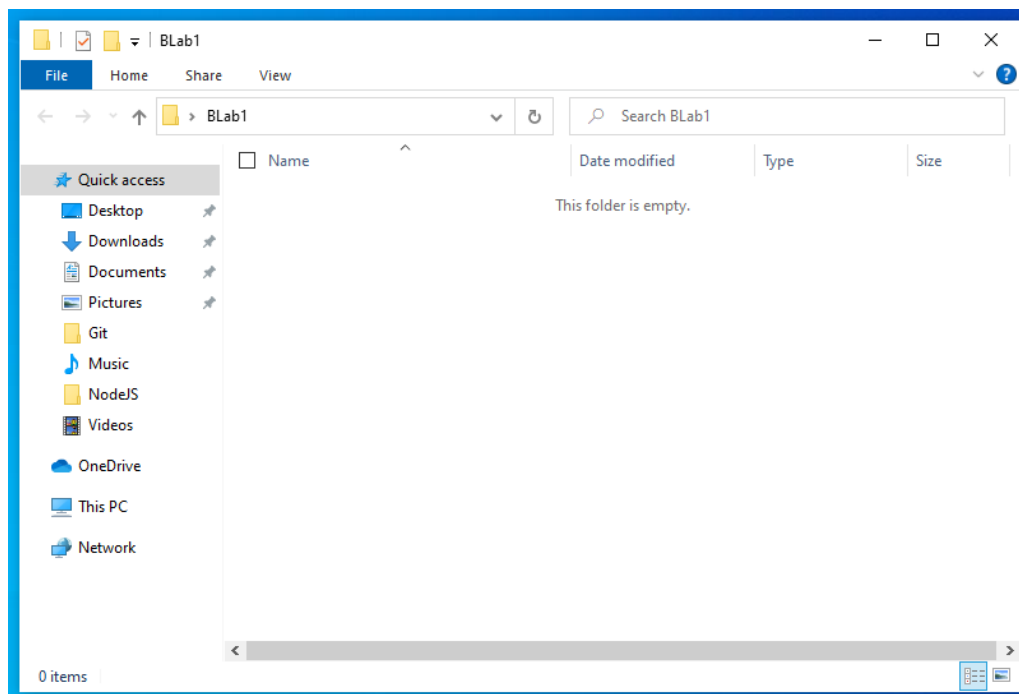
1. Microsoft Visual Studio Code - <https://code.visualstudio.com/>
2. Git - <https://git-scm.com/>
3. Node.js - <https://nodejs.org/en/>

*Observație: Dacă totuși există anumite situații (e.g., erori sau altfel de mesaje de incompatibilitate), urmați procedurile de instalare atașate la finalul acestui document.*

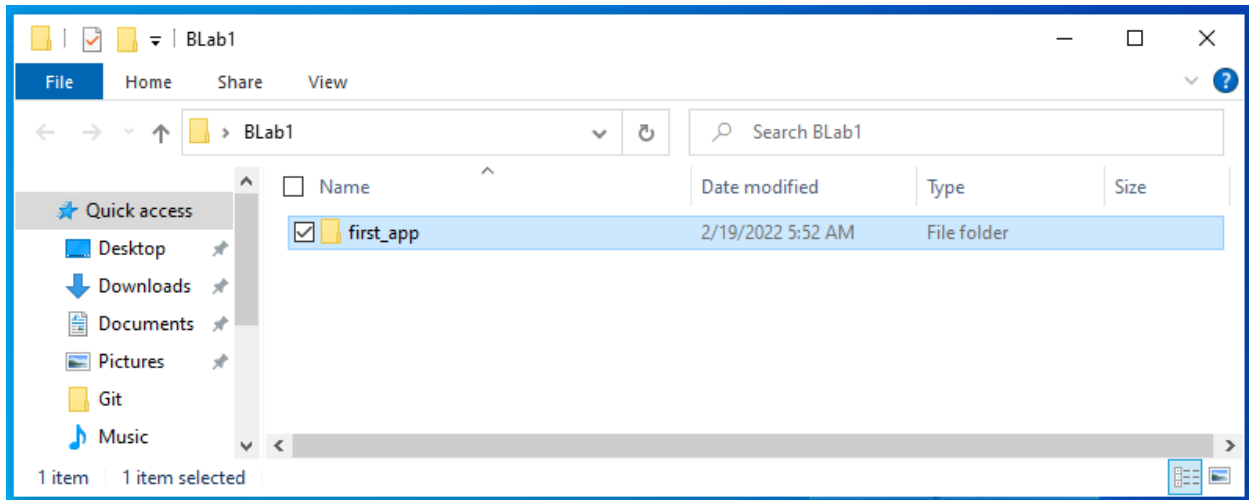
*Notă: Laboratoarele sunt realizate într-un mediu virtualizat. Mașină virtuală (în VMWare) cu Windows 10 Pro x64.*

#### **Lucrare practică**

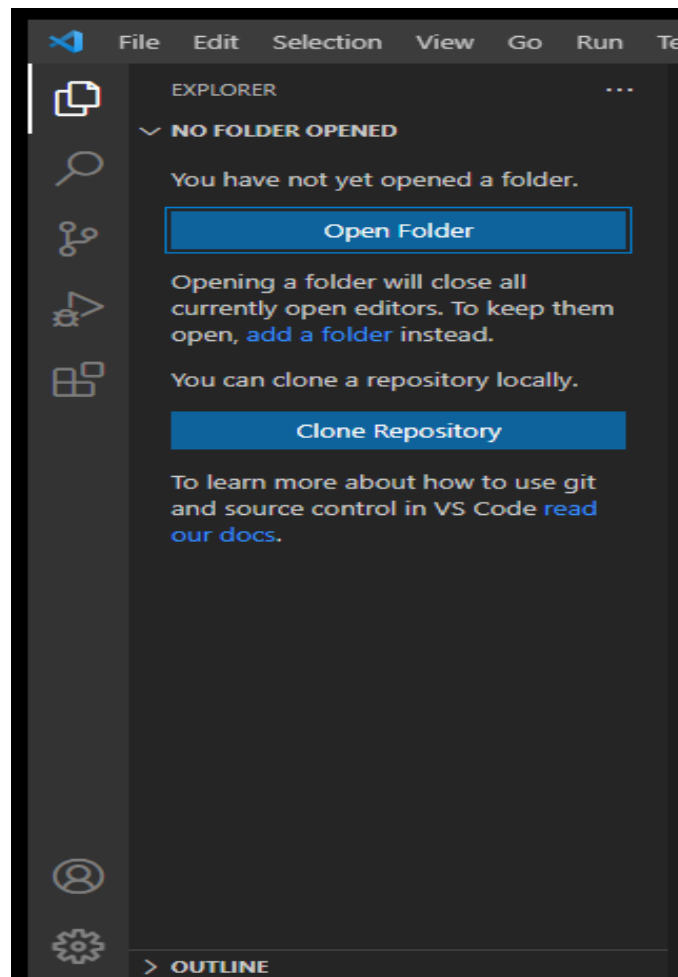
1. (Dacă este cazul) Instalați *Microsoft Visual Studio Code* (cu setările implicite)
2. (Dacă este cazul) Instalați *Git* (cu setările implicite)
3. (Dacă este cazul) Instalați *Node.JS* (cu setările implicite)
4. Creați un folder numit *BLab1* pe desktop sau într-o locație preferată.



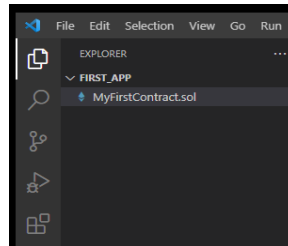
5. Deschideți folderul *BLab1* și creați un alt folder *first\_app*



6. Deschideți *Microsoft Visual Studio Code*. Mergeți în stânga și selectați Explorer sau apăsați Ctrl+Shift+E



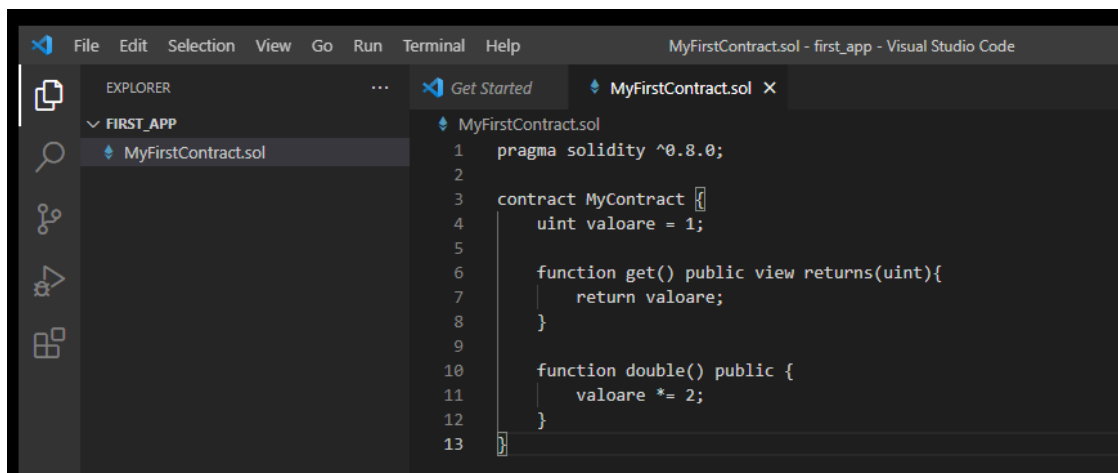
7. Deschideți folderul create recent *first\_lab*, și apoi creați un nou fișier cu extensia *\*.sol*. Pentru acest laborator am ales numele *MyFirstContract.sol*.



8. Plasați următorul cod sursă în fereastra deschisă alăturat.

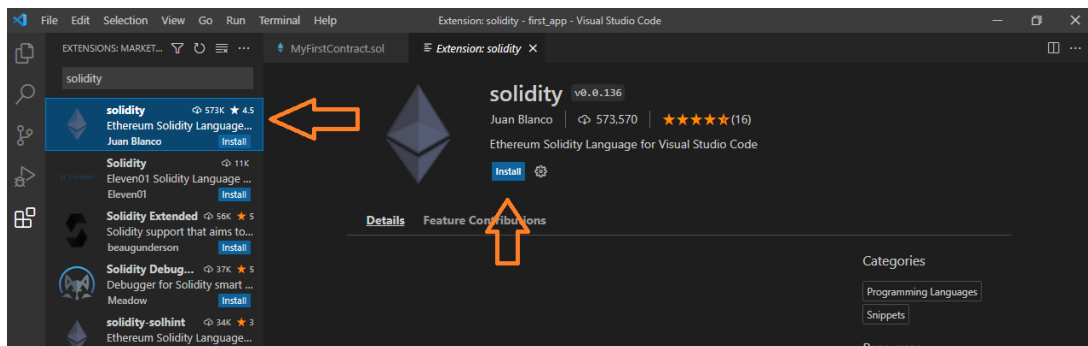
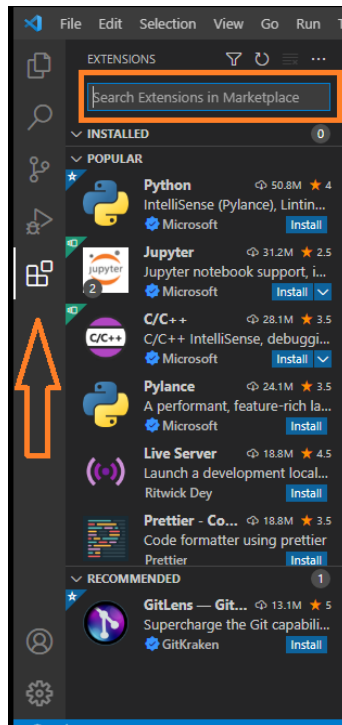
```
1 pragma solidity ^0.8.0;
2
3 contract ContractulMeu {
4     uint valoare = 1;
5
6     function get() public view returns (uint)
7     {
8         return valoare;
9     }
10
11     function double() public {
12         valoare *= 2;
13     }
14 }
```

Fereastra cu fișierul sursă trebuie să arate în felul următor



9. După cum se observă, nu există niciun fel de sintaxă sau colorit al sintaxei. Acest lucru se datorează faptului că nu extensia *solidity* nu este instalată în Visual Studio Code. Pentru a rezolva această situație și pentru a putea executa contractele respective, mergeți în partea stângă a ferestrei, și click pe *Extensions* sau apăsați *Ctrl+Shift+X*. Introduceți cuvântul cheie *solidity* și instalați extensia

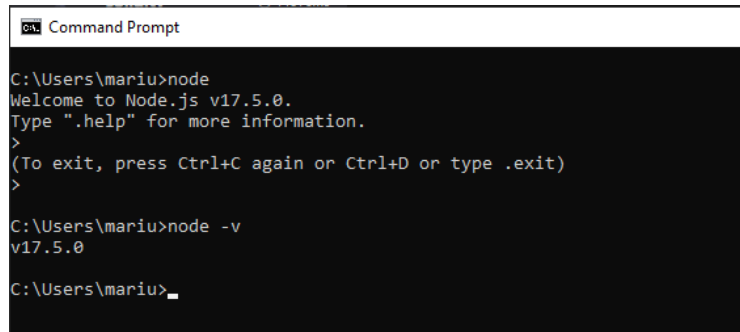
corespunzătoare ca în imaginea a doua de mai jos prin apăsarea butonului *Install*. Când procesul de instalare al extensiei este realizat cu succes, în locul butonului *Install* va apărea o nouă etichetă a butonului intitulată *Disable*.



10. Verificați că sintaxa codului sursă a fost schimbată corespunzător și că arată ca în imaginea de mai jos.

```
MyFirstContract.sol 1 x Extension: solidity
MyFirstContract.sol
1 pragma solidity ^0.8.0;
2
3 contract MyContract {
4     uint valoare = 1;
5
6     function get() public view returns(uint){
7         return valoare;
8     }
9
10    function double() public {
11        valoare *= 2;
12    }
13 }
```

11. Pentru a putea continua, verificați că instalarea Node.JS a fost realizată cu succes și că aveți ultima versiune. Pentru mai multe detalii consultați *Anexa 3 – Instalarea Node.JS*. Verificarea se face prin tastarea comenzii `node` sau `node -v`. Dacă se vede ca în figura respective înseamnă că suntem pe drumul cel bun.



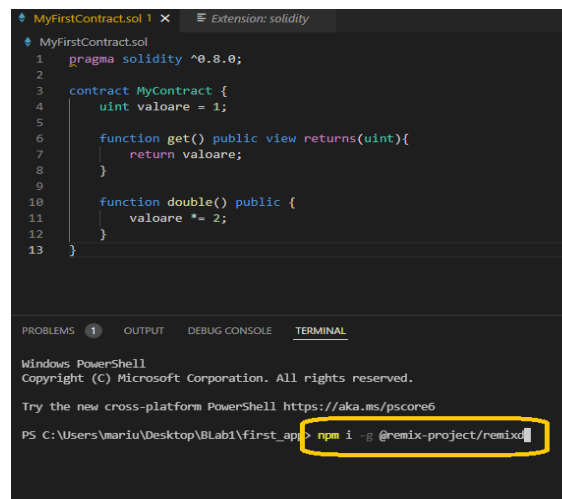
```
Command Prompt

C:\Users\mariu>node
Welcome to Node.js v17.5.0.
Type ".help" for more information.
>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
>

C:\Users\mariu>node -v
v17.5.0

C:\Users\mariu>
```

12. După instalarea Node.JS și verificarea acestuia, următorul pas reprezintă instalarea Remix. Pentru a face acest lucru, mergeți fie în Command Prompt sau în Terminalul din Visual Studio Code, și tastați următoarea comandă: `npm i -g @remix-project/remixd` și apăsați Enter. Așteptați ca procesul de configurare să se finalizeze cu succes.



```
MyFirstContract.sol | x Extensions: solidity

1  pragma solidity ^0.8.0;
2
3  contract MyContract {
4      uint valoare = 1;
5
6      function get() public view returns(uint){
7          return valoare;
8      }
9
10     function double() public {
11         valoare *= 2;
12     }
13 }

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

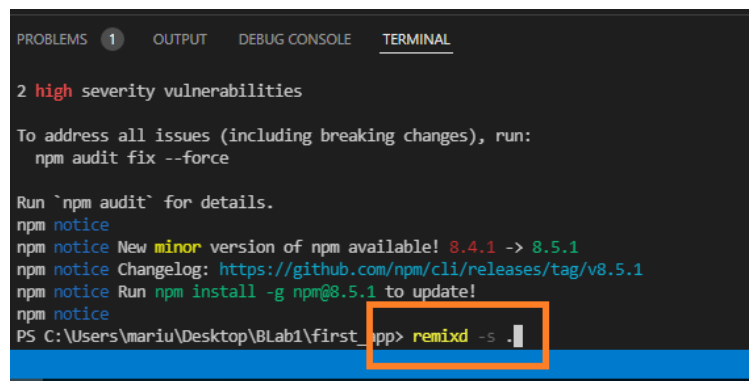
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\mariu\Desktop\BLab1\first_app> npm i -g @remix-project/remixd
```

13. Introduceți în terminal următoarea comandă: `remixd -s .`

Punctul (.) reprezintă calea locală unde ne aflăm în proiect.



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

2 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 8.4.1 -> 8.5.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.5.1
npm notice Run npm install -g npm@8.5.1 to update!
npm notice
PS C:\Users\mariu\Desktop\BLab1\first_app> remixd -s .
```

14. Dacă este pentru prima dată când rulați comenzile respective, cu siguranță veți obține următorul mesaj de eroare.

```
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.5.1
npm notice Run npm install -g npm@8.5.1 to update!
npm notice
PS C:\Users\mariu\Desktop\BLab1\first_app> remixd -s .
remixd : File C:\Users\mariu\AppData\Local\Temp\remixd\remixd.ps1 cannot be loaded because running scripts is disabled on this system.
For more information, see about\_Execution\_Policies at https://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ remixd -s .
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Users\mariu\Desktop\BLab1\first_app>
```

15. Ca să-l rezolvați, rulați următoarea comandă:

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

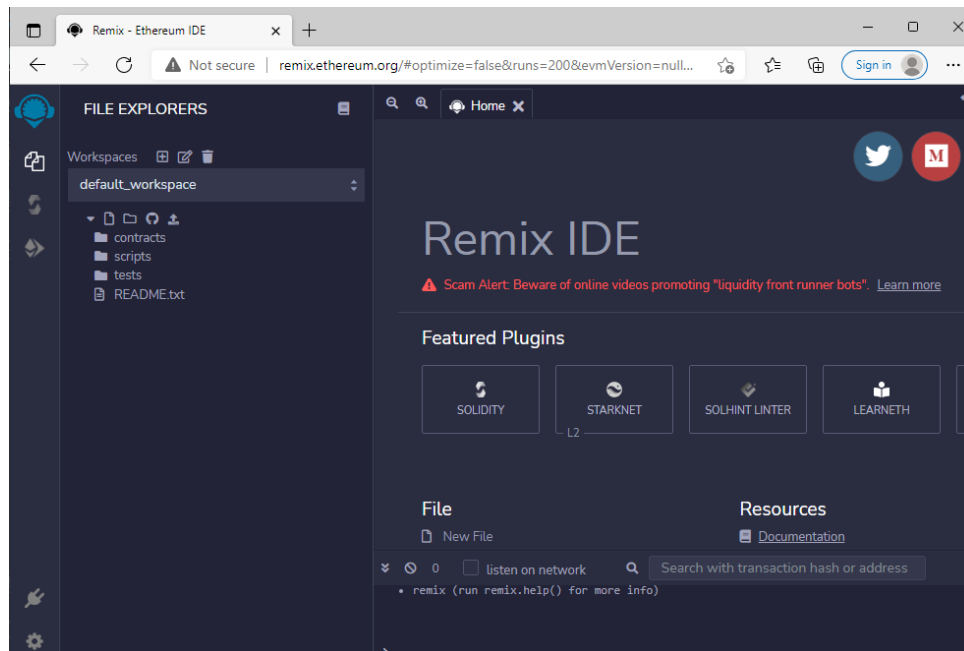
16. Rulați din nou `remixd -s .`

Următorul output este afișat.

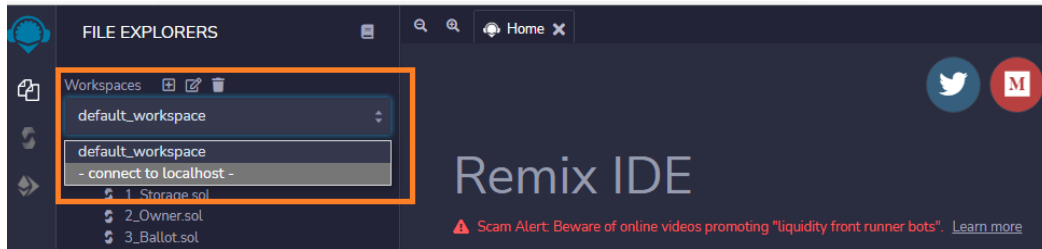
```
[WARN] You can only connect to remixd from one of the supported origins.
[WARN] Any application that runs on your computer can potentially read from and write to all files in the directory.
[WARN] Symbolic links are not forwarded to Remix IDE

[INFO] Sat Feb 19 2022 07:57:53 GMT-0800 (Pacific Standard Time) remixd is listening on 127.0.0.1:65520
[INFO] Sat Feb 19 2022 07:57:53 GMT-0800 (Pacific Standard Time) slither is listening on 127.0.0.1:65523
```

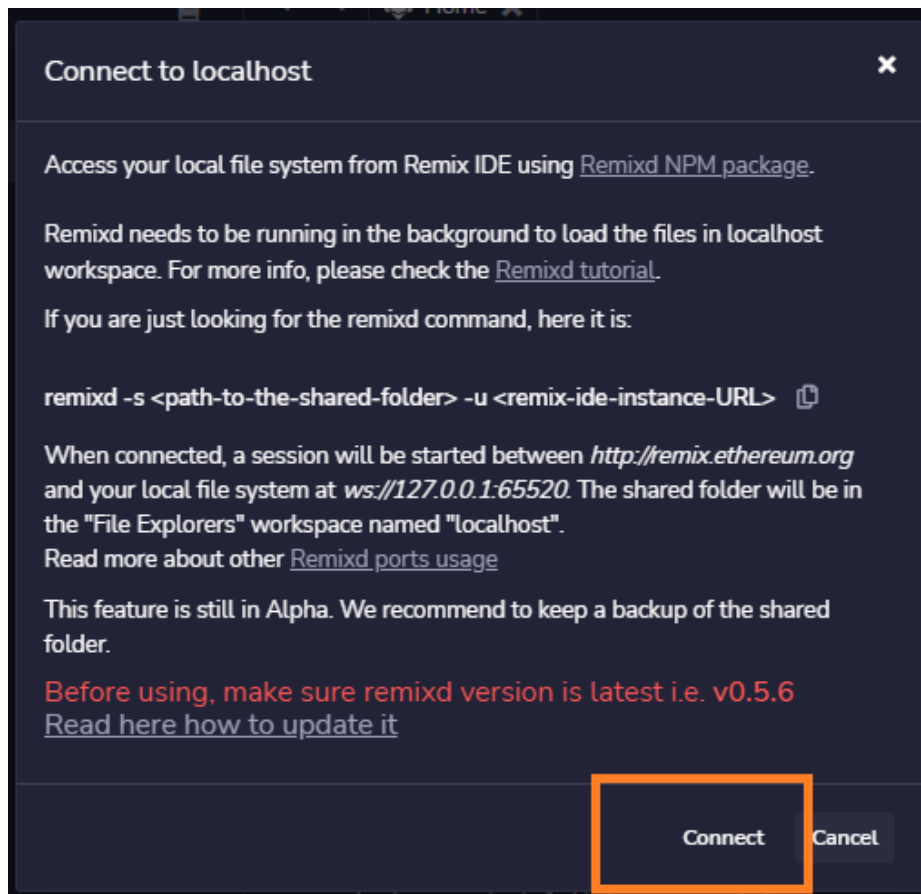
17. Deschideți browserul și introduceți următorul link: <http://remix.ethereum.org>. Următoarea fereastră este afișată.



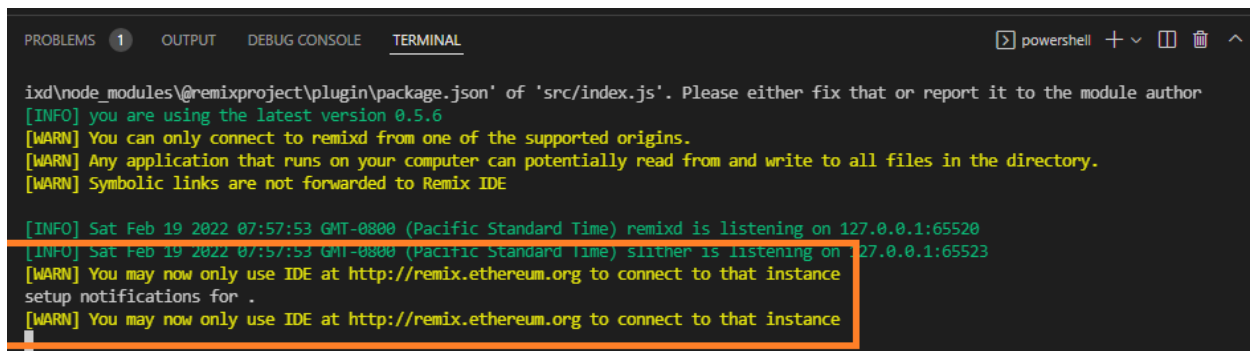
18. În fereastra din browser, selectați din *default\_workspace* în *connect to localhost*



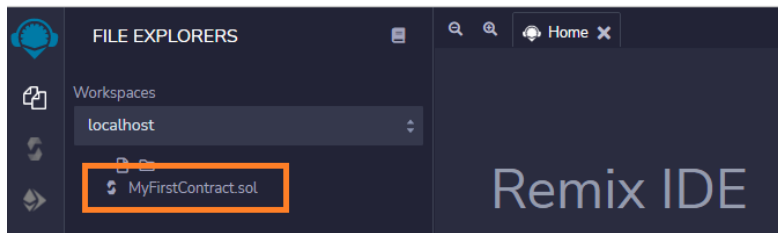
19. Click *Connect*



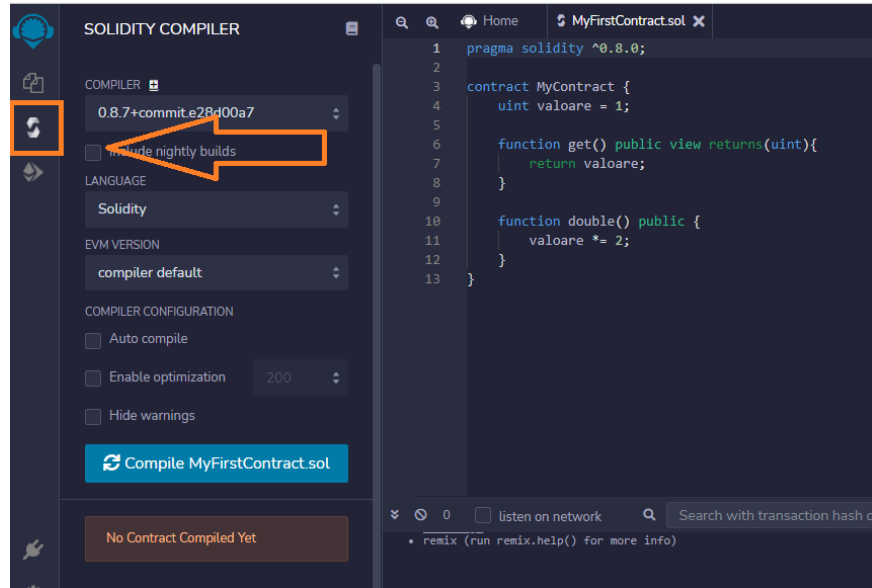
20. Verificați în terminalul din Visual Studio Code că totul este ok prin identificarea următoarelor două mesaje.



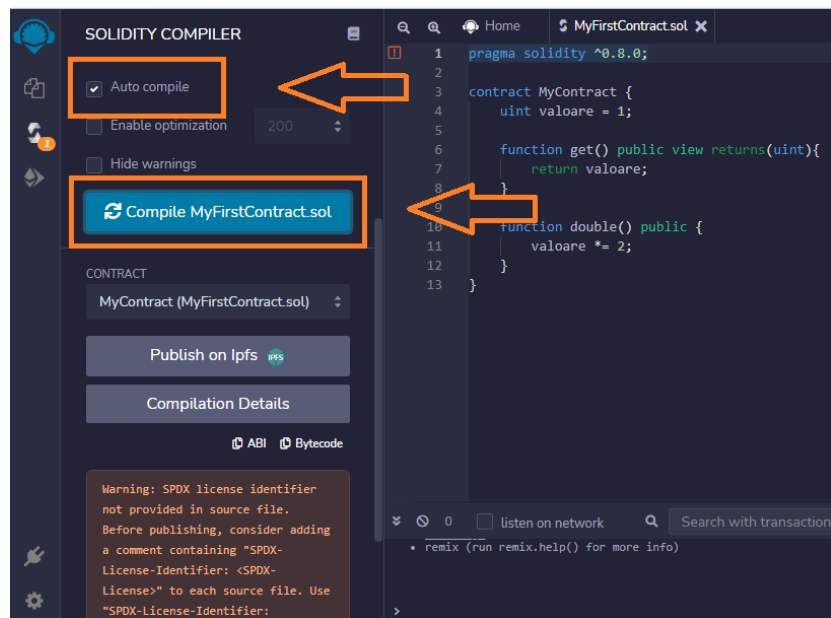
21. Verificați în browser că structura proiectului este identică cu cea din Visual Studio Code.



22. În fereastra din browser, click pe *Solidity Compiler*.

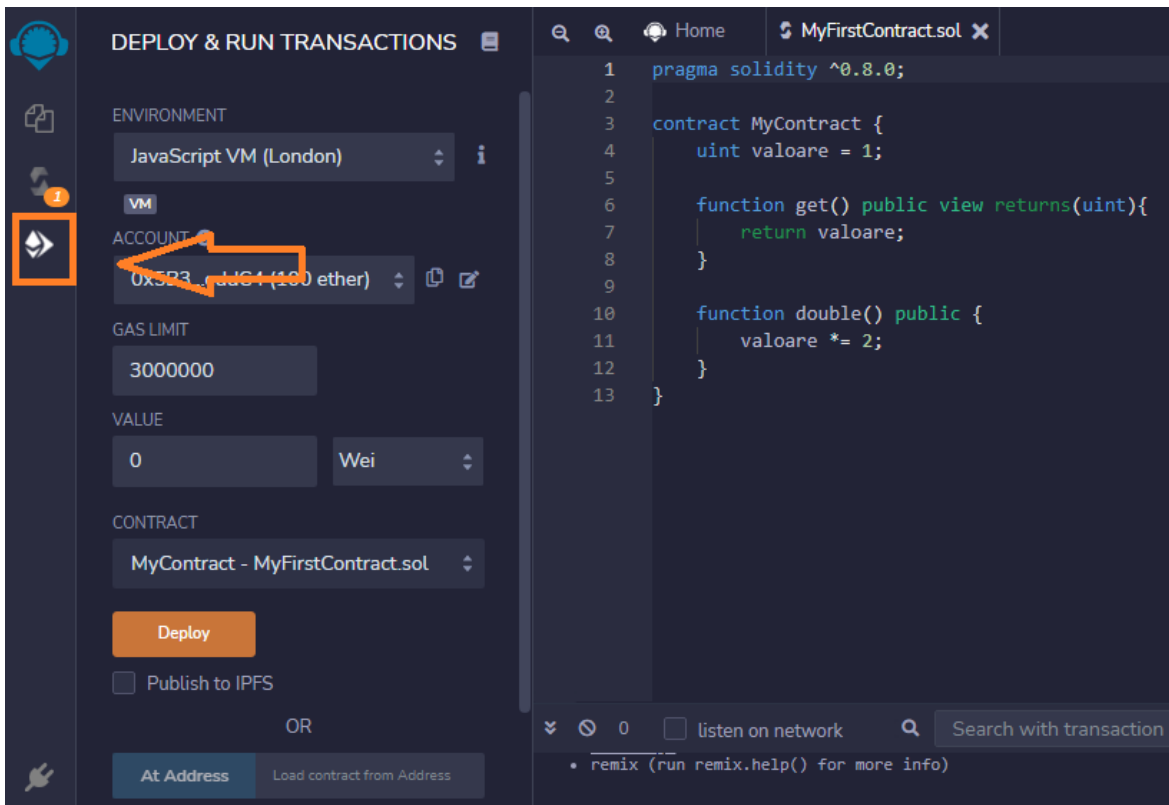


23. Puteți bifa căsuța *Auto Compile* și compilați contractual apăsând pe *Compile MyFirstContract.sol*.

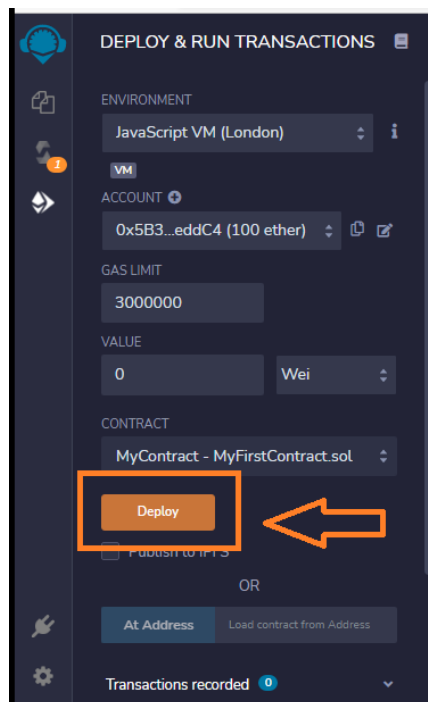




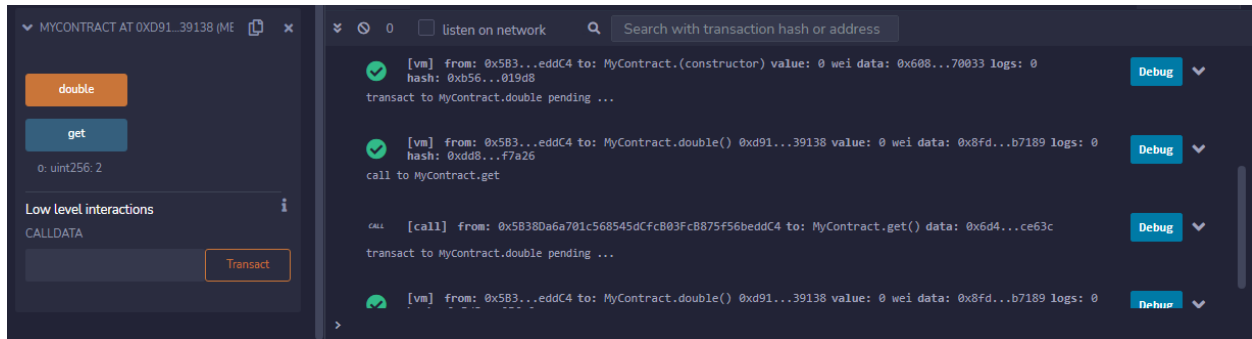
24. Click pe *Deploy* and Run



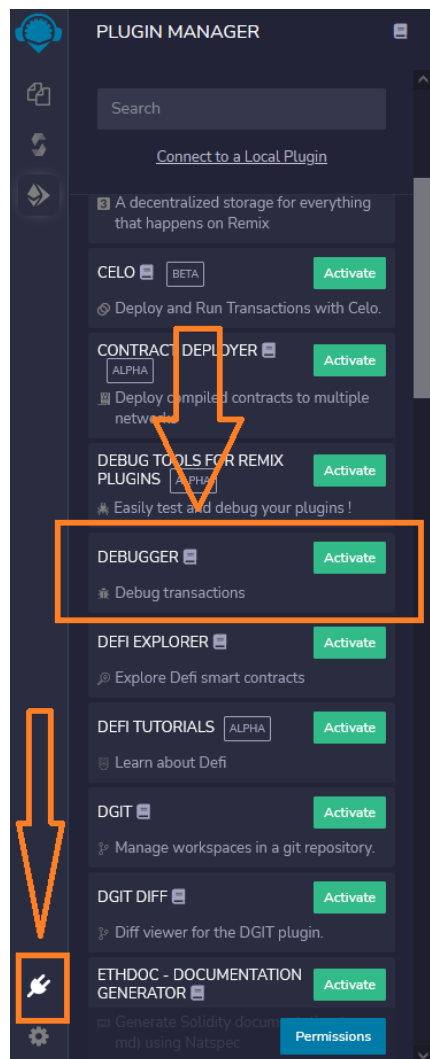
25. Lăsați toate setările cum sunt și click pe butonul *Deploy*.



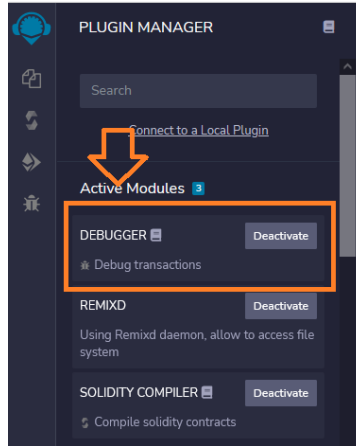
26. Apăsați pe metodele *get* și *double*. Urmăriți ceea ce se întâmplă.



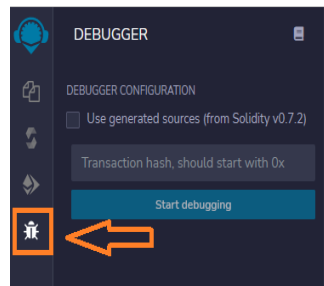
27. Click pe *Plugin Manager* și click pe *Debugger*



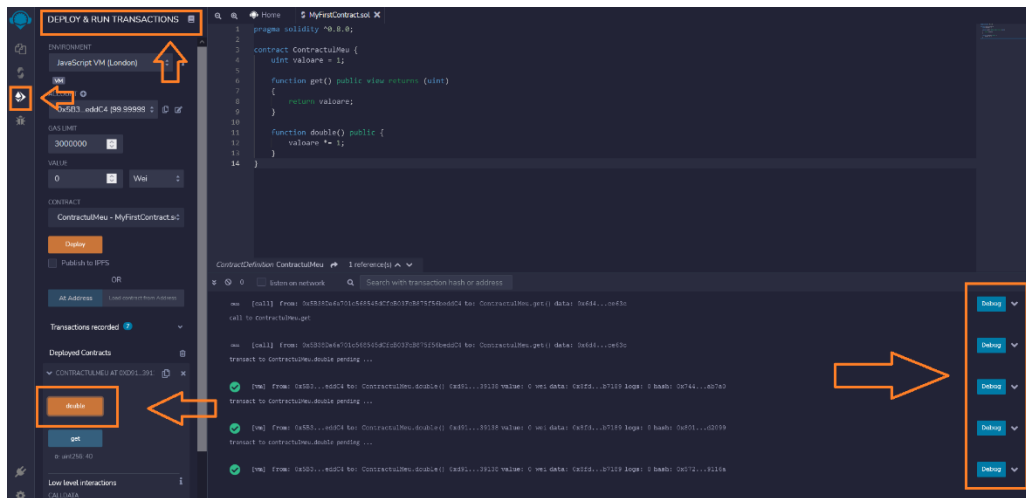
28. Verificați că debuggerul a fost activat



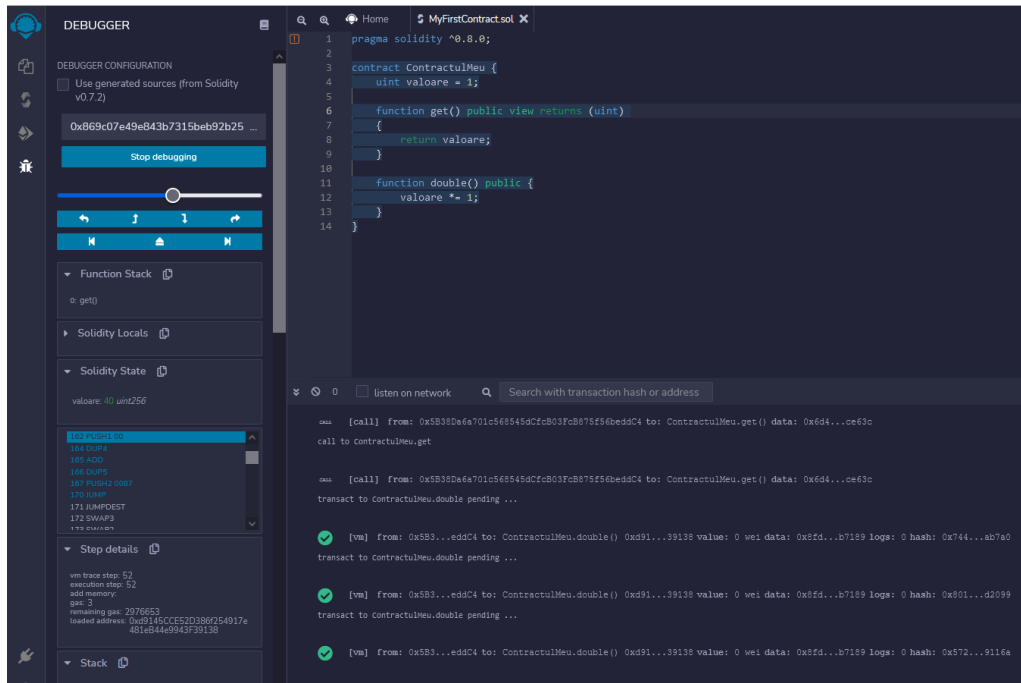
29. Verificați că tab-ul pentru debugger există



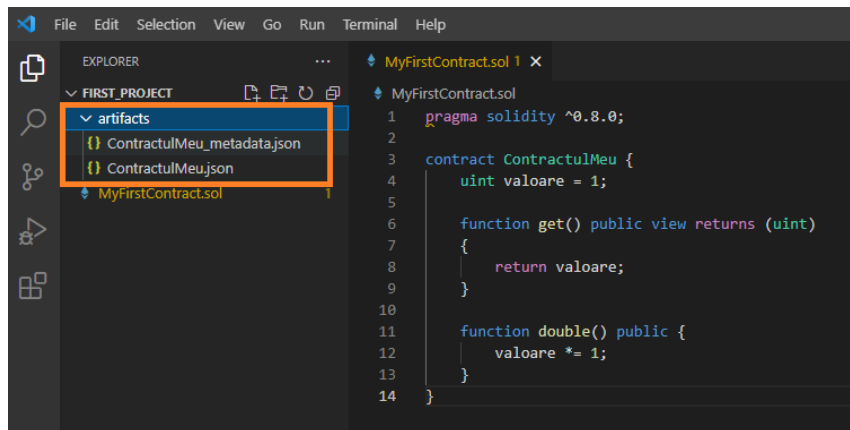
30. Mergeți înapoi în *Deploy & Run Transactions*, click pe proprietatea *double* și verificați că aveți butonul cu *Debug* aferent fiecărui eveniment.



31. Click pe un buton *Debug* aferent unui eveniment sau tranzacție. Verificați ce se întâmplă.



32. Mergeți înapoi în Visual Studio Code. Veți observa un folder nou intitulat *artifacts*.



În folderul *artifacts* există două fișiere:

- `ContractulMeu_metadata.json`
- `ContractulMeu.json`

33. Deschideți fișierul `ContractulMeu_metadata.json`. Examinați interfața ABI (mai multe detalii se află sub cele două imagini) și porțiunea *settings*.

```
artifacts > {} ContractulMeu_metadata.json > ...
1 {
2   "compiler": {
3     "version": "0.8.7+commit.e28d00a7"
4   },
5   "language": "Solidity",
6   "output": {
7     "abi": [
8       {
9         "inputs": [],
10        "name": "double",
11        "outputs": [],
12        "stateMutability": "nonpayable",
13        "type": "function"
14      },
15      {
16        "inputs": [],
17        "name": "get",
18        "outputs": [
19          {
20            "internalType": "uint256",
21            "name": "",
22            "type": "uint256"
23          }
24        ],
25        "stateMutability": "view",
26        "type": "function"
27      }
28    ]
29  }
30 }
31
32
33
34
35
36
37
38
39
40 "settings": {
41   "compilationTarget": {
42     "MyFirstContract.sol": "ContractulMeu"
43   },
44   "evmVersion": "london",
45   "libraries": {},
46   "metadata": {
47     "bytecodeHash": "ipfs"
48   },
49   "optimizer": {
50     "enabled": false,
51     "runs": 200
52   },
53   "remappings": []
54 }
```

În fișierul `ContractulMeu_metadata.json` avem interfața ABI (Application Binary Interface). ABI reprezintă interfața dintre două module tip program, de cele mai multe ori această interfață face legătura dintre sistemul de operare și programele utilizatorilor.

Pentru a putea înțelege interfața ABI, este necesar ca fiecare element să fie analizat și înțeles corespunzător, după cum urmează:

- Elementele următoare definesc descrierea unui ABI pentru o funcție
  - *type* – definește tipul funcției. Există trei tipuri de funcții:
    - *function*,
    - *constructor*,
    - *receive* - pentru funcția de primire ether - <https://docs.soliditylang.org/en/develop/contracts.html#receive-ether-function>
    - *fallback* - pentru funcții default <https://docs.soliditylang.org/en/develop/contracts.html#fallback-function>)
  - *name* – definește numele unei funcții
  - *inputs* – este un vector de obiecte de intrare care definește parametrii, fiecare obiect este caracterizat prin următoarele:
    - *name* – definește numele parametrilor
    - *type* – definește tipurile canonice ale parametrilor, exemplu: *uint256*
    - *components* – folosit pentru a declara tipuri tip tupluri.
    - *outputs* – este un vector de obiecte de ieșire (output) similar cu cele de intrare
  - *stateMutability*
- Elementele următoare definesc descrierea unui eveniment
  - *type* – întotdeauna valoarea acestui atribut este *event*
  - *name* – definește numele evenimentului
  - *inputs* – este un vector de obiecte ce definește parametrii, fiecare obiect este caracterizat prin următoarele:
    - *name* – definește numele parametrilor

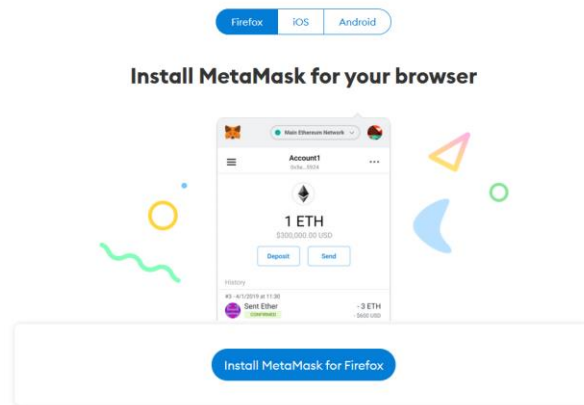
- *type* – definește tipurile canonice ale parametrilor, exemplu: *uint256*
- *components* – definește tipuri tip tupluri, dacă se ajunge la acel tuplu în urma execuției, acel tip este reprezentat ca `type=tuple [listă proprietăți]`

34. MetaMask și comunicarea cu un blockchain real - Deschideți browserul și introduceți următorul link: <https://metamask.io/>. Recomand Firefox sau Chrome.

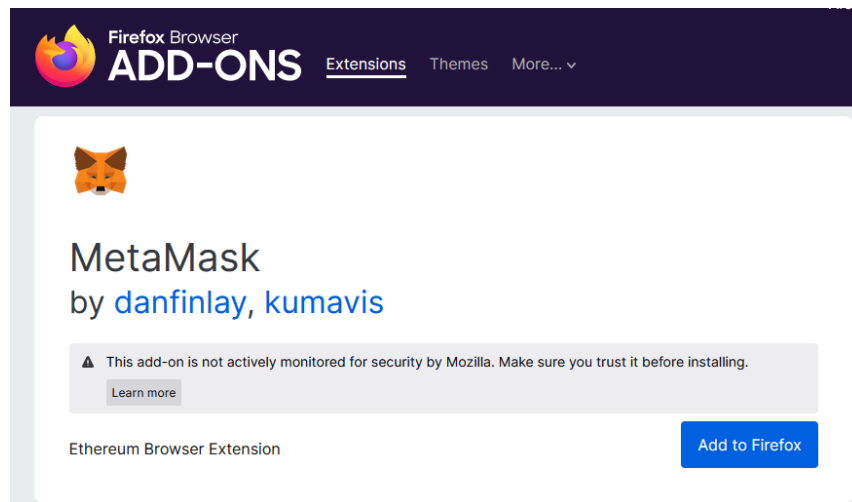
35. Click *Download*



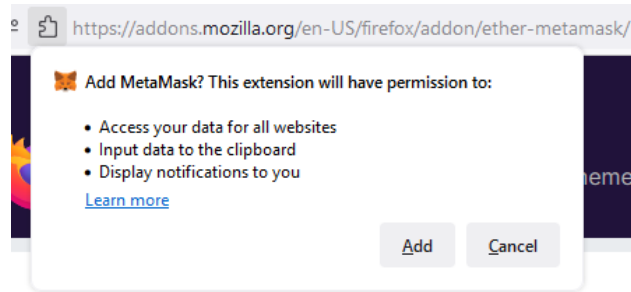
36. Click *Install MetaMask for Firefox*



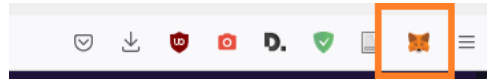
37. Click pe *Add to Firefox*



38. Click *Add*



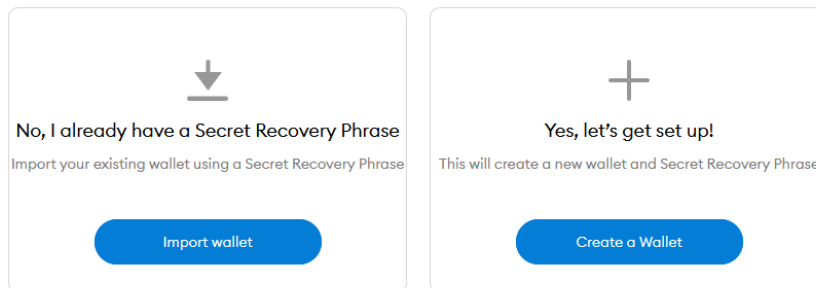
39. Verificați că a fost adăugat cu succes în bara browserului



40. Creați un portofel (wallet) dacă nu aveți deja unul. Pentru acest laborator voi utiliza portofelul meu propriu și voi apăsa pe *Import Wallet*.



New to MetaMask?



41. Click pe Import Wallet și apoi click pe *No thanks* sau *I Agree*



## 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!

No Thanks

I Agree

This data is aggregated and is therefore anonymous for the purposes of General Data Protection Regulation (EU) 2016/679. For more information in relation to our privacy practices, please see our [Privacy Policy here](#).

### 42. Completați cele cerute



< Back

## Import a wallet with Secret Recovery Phrase

Only the first account on this wallet will auto load. After completing this process, to add additional accounts, click the drop down menu, then select Create Account.

Secret Recovery Phrase

Paste Secret Recovery Phrase from clipboa

☐

Show Secret Recovery Phrase

New password (min 8 chars)

Confirm password

☐

I have read and agree to the [Terms of Use](#)

Import

### 43. Felicitări pentru configurarea MetaMask.





## Congratulations

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

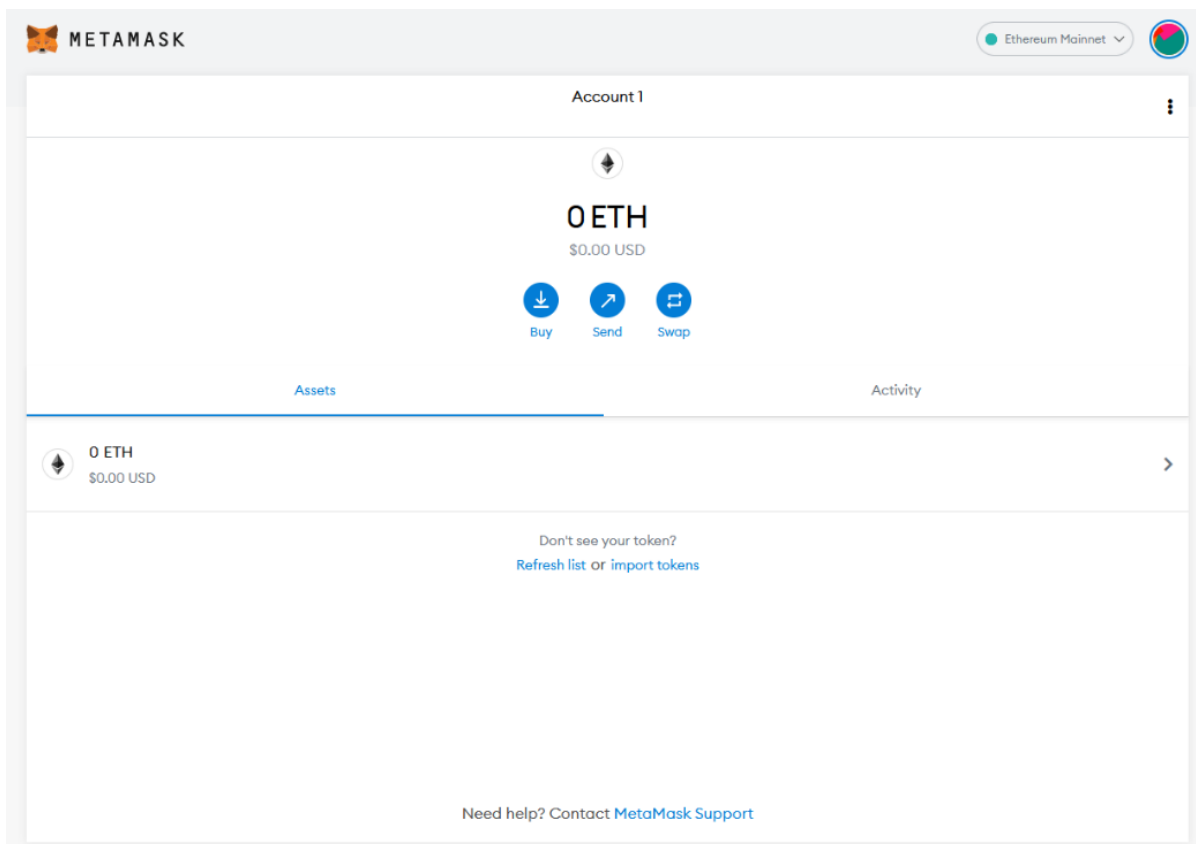
### Tips on storing it safely

- 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

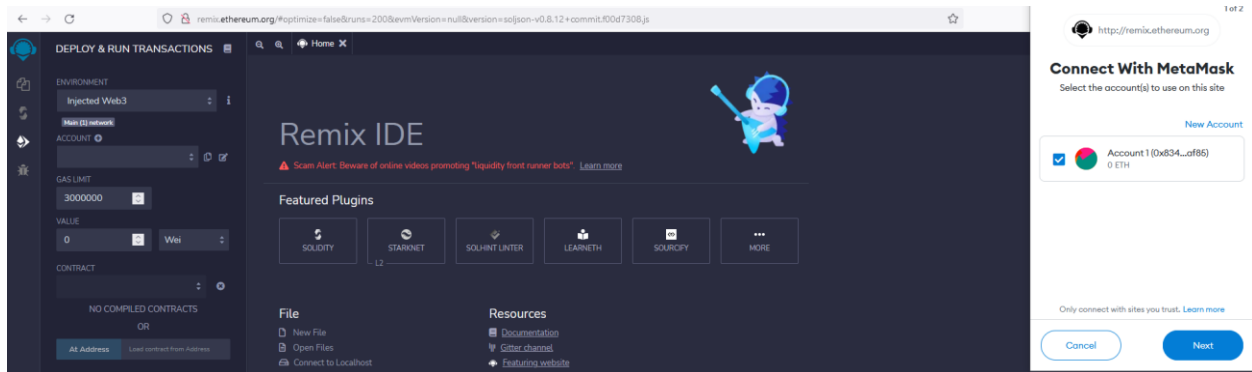
44. Confirmați că totul este ok prin vizualizarea acestei ferestre:



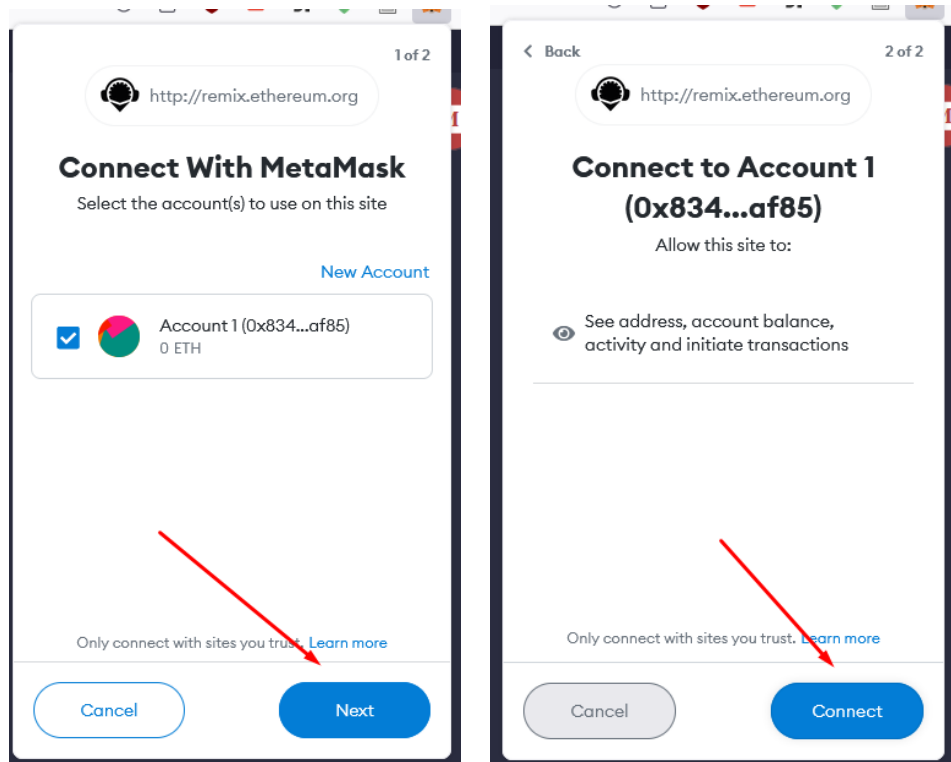
45. Mergeți înapoi în browser la [remix.ethereum.org](https://remix.ethereum.org) și click pe tab-ul *Deploy & Run Transactions*



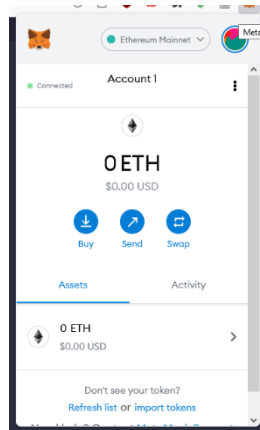
46. Din combobox-ul *Environment* selectați *Injected Web3*. Următoarea fereastră se va deschide



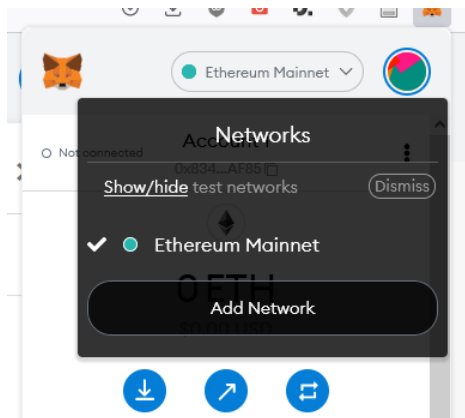
47. Click pe *Next* și apoi *Connect*.



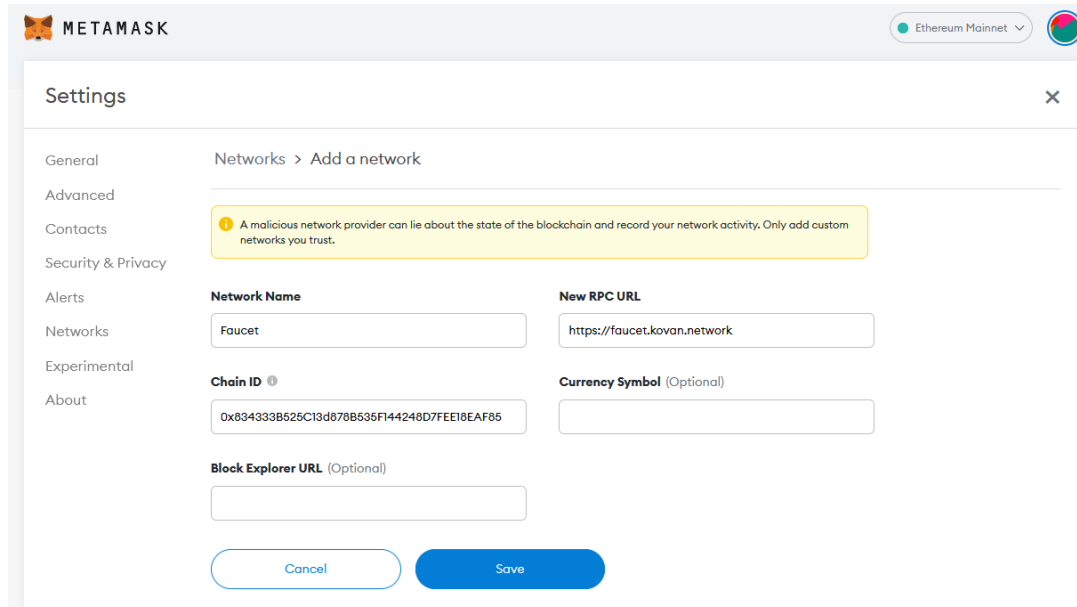
48. Următoarea fereastră ar trebui să fie afișată.



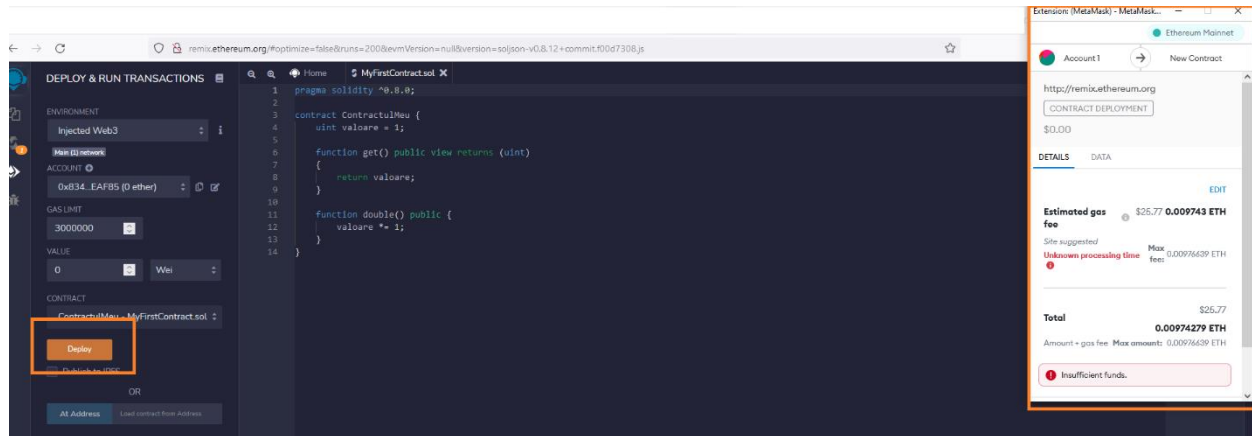
49. Adăugați o rețea dorită făcând click pe *Ethereum Mainet* și apoi click pe *Add Network*



50. Completați după cum urmează (uneori funcționează, alteori nu). Click *Save*. Este o rețea foarte bună de teste. Există și altele.



51. Revenim la remix.ethereum.org și din fereastra Deploy & Run Transactions facem click pe butonul Deploy. Bineînțeles, fiind o rețea (portofel) fără fonduri, nu o să avem posibilitatea să rulăm contractul respectiv



52. În momentul în care o tranzacție este efectuată cu success, veți avea posibilitatea să o verificați folosind site-ul <https://kovan.etherscan.io>

