ĐẠI NAM
UNIVERSITY

BÀI 4

HỢP ĐỒNG THÔNG MINH
SMART CONTRACTS

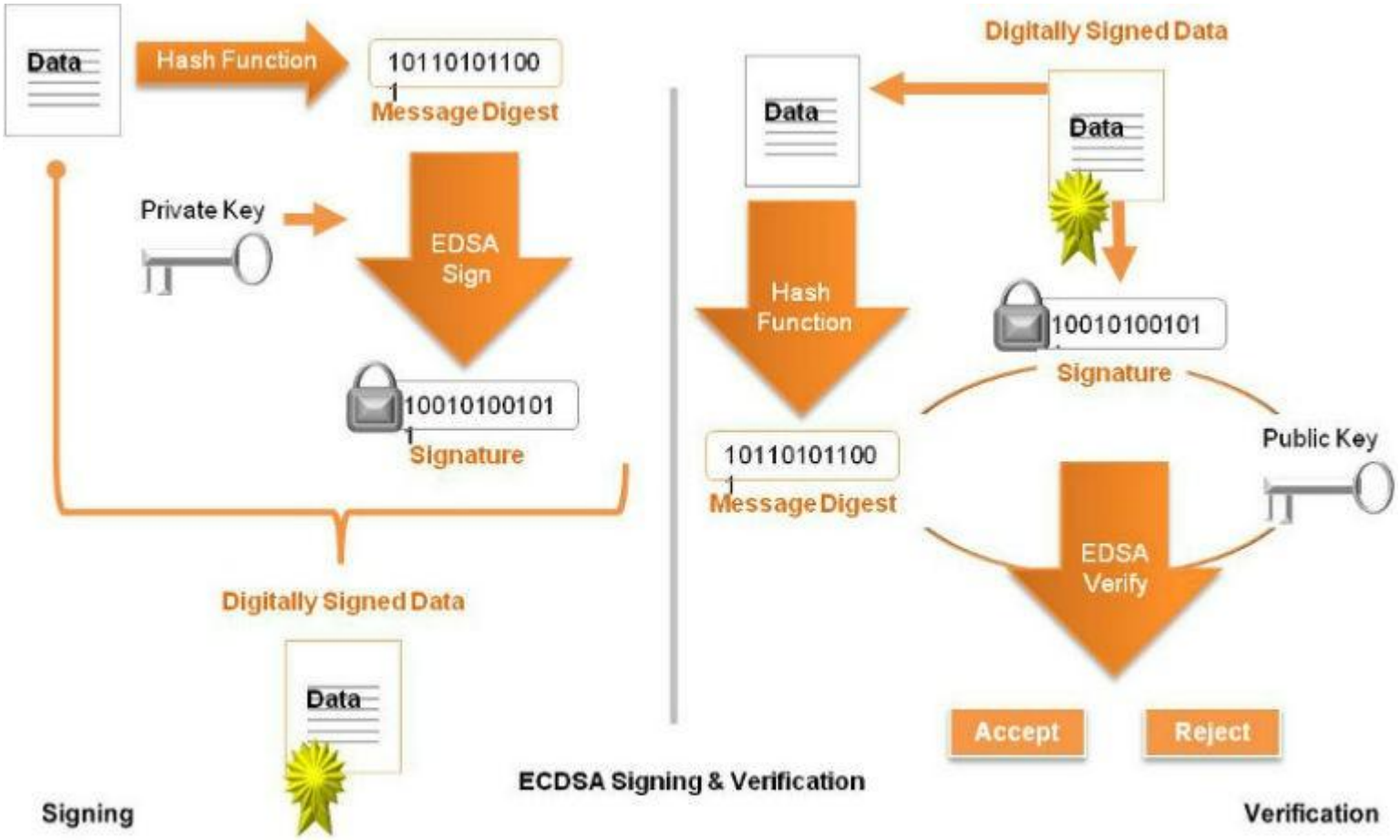
TS. Trần Đăng Công

Tel: 0964981451

Email: congtd@dainam.edu.vn

CÔNG NGHỆ BLOCKCHAIN

1. TỔNG QUAN VỀ CÔNG NGHỆ BLOCKCHAIN
2. MẬT MÃ TRONG BLOCKCHAIN
3. HỢP ĐỒNG THÔNG MINH
4. ỨNG DỤNG PHI TẬP TRUNG



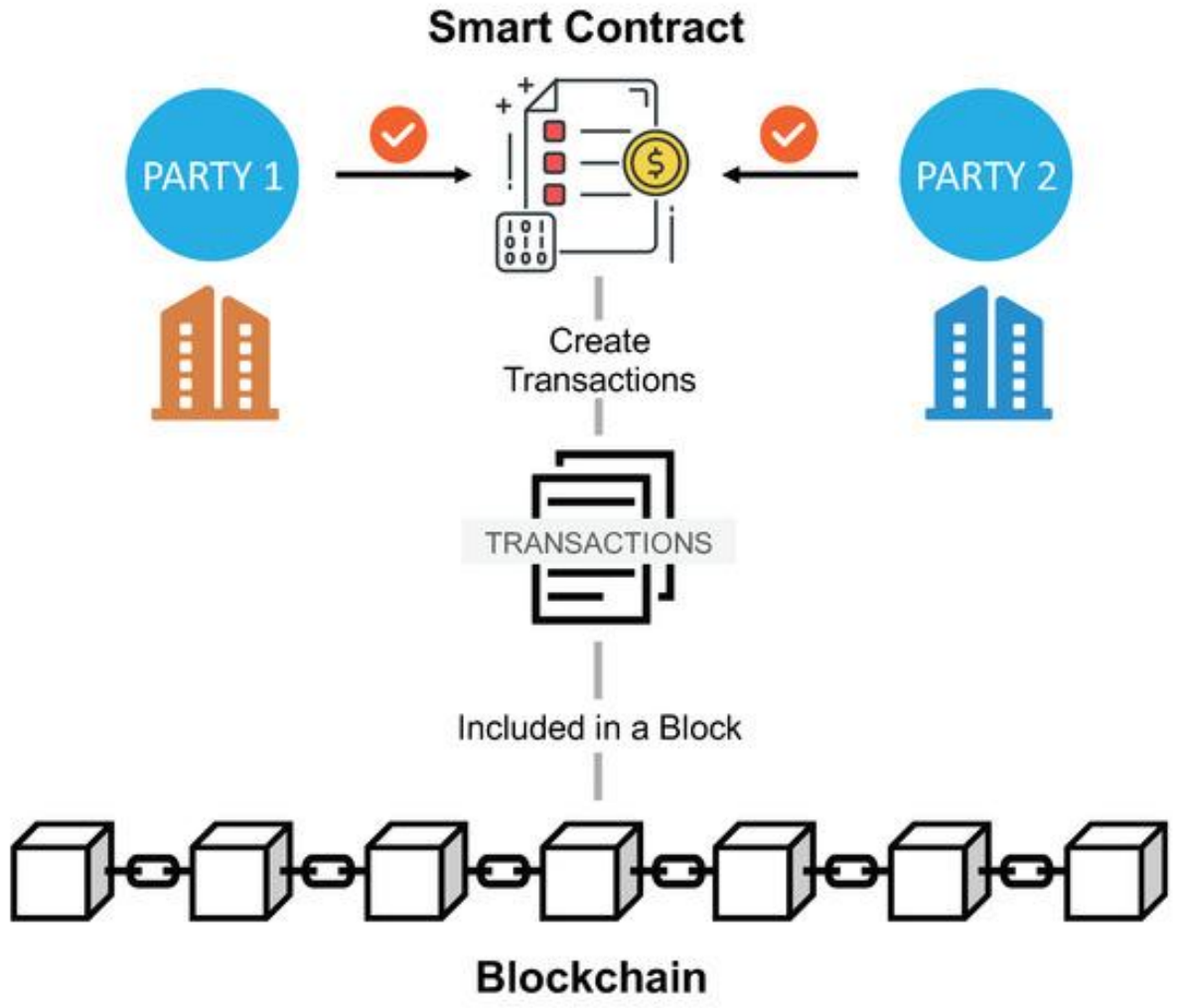
- **Phân tích hoạt động của việc ký và xác nhận chữ ký như trên hình.**

1. Giới thiệu
2. Ngôn ngữ lập trình Solidity
3. Triển khai trên Blockchain
4. Case study: Voting Smart Contract

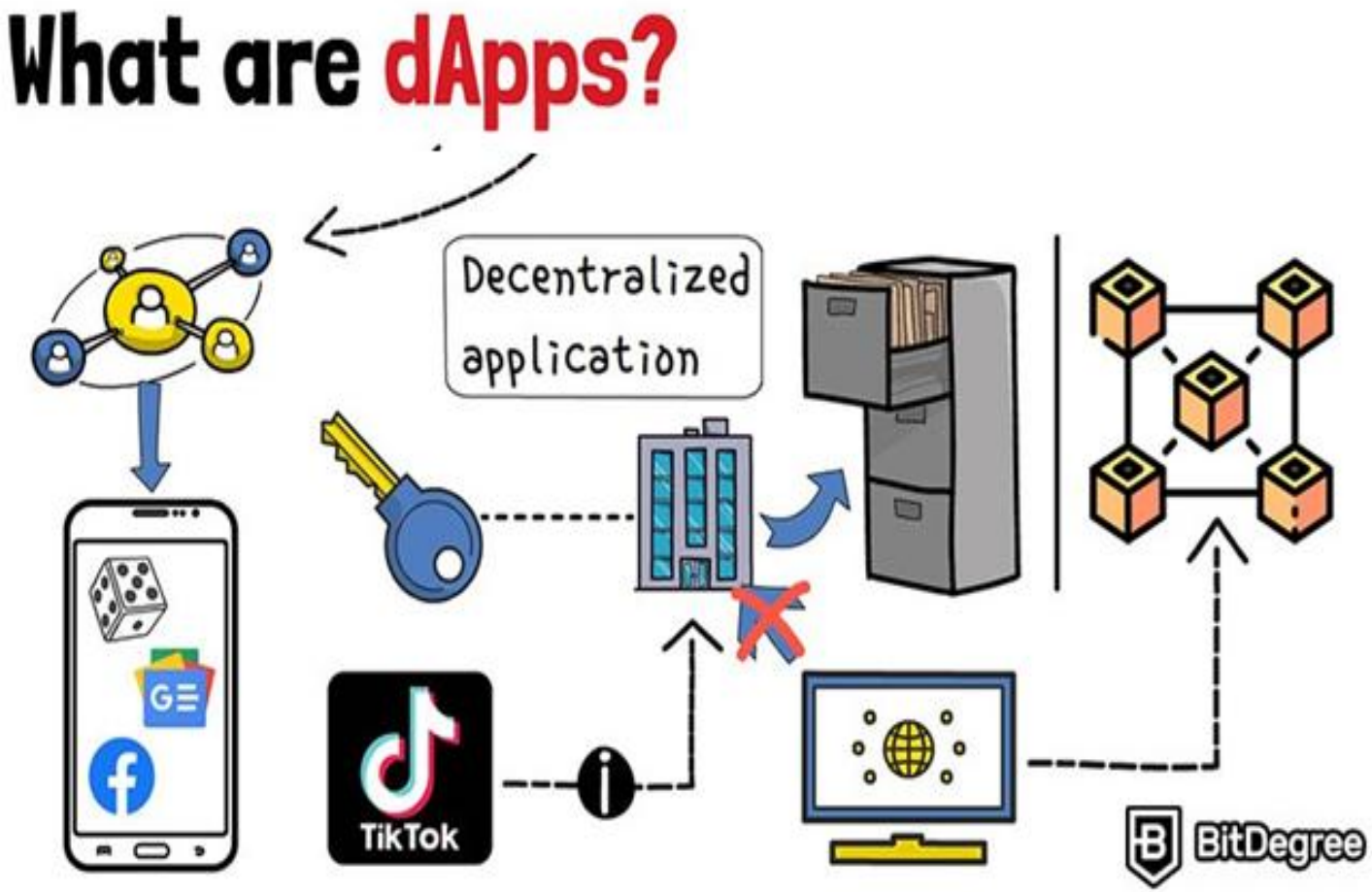
GIỚI THIỆU

- Hợp đồng trong thực tế
- Các vấn đề dễ gây tranh cãi
- Theo dõi tiến độ hợp đồng

- **HĐTM (Smart Contract)** là một chương trình **tự động thực thi** các điều khoản và điều kiện của hợp đồng khi các điều kiện nhất định được đáp ứng.
- HĐTM được triển khai dựa trên **công nghệ Blockchain**, giúp loại bỏ sự cần thiết của bên trung gian, giảm chi phí và tăng tính minh bạch.



- HĐTM thuộc thế hệ **Blockchain 2.0** (Ethereum - ETH), nền tảng đầu tiên hỗ trợ DApps
- Thuật toán đồng thuận từ Proof of Work (PoW) => **Proof of Stake (PoS)**
- Việc triển khai HĐTM đang được thực hiện **rộng rãi** trong các hoạt động
- **Tích hợp AI** vào HĐTM
- Vấn đề **lỗ hổng, mất an toàn** trong HĐTM



Parameter	Traditional contract	Smart contract
Time required	1–3 days	Minutes
Payment scenario	Manual remittance	Automatic remittance
Cost	Expensive	Not expensive
Signature mode	Physical	Digital
Escrow	Required	Not required
Layers requirement	Compulsory	Not compulsory
Reconciliation process	Slow	Fast
Trusted third party	Necessary	Not required
Dispute resolution via	Judges, arbitrators	Consensus mechanism
Specification	Natural language	Smart code
Archiving	Hard	Easy
Transparency	Available	Not available
Security	Limited	High level security

- **Blockchain** làm nền tảng phát triển Ethereum từ năm 2015, do Vitalik Buterin người Canada gốc Nga đưa ra.

- **Một số công nghệ** được ứng dụng như sau:

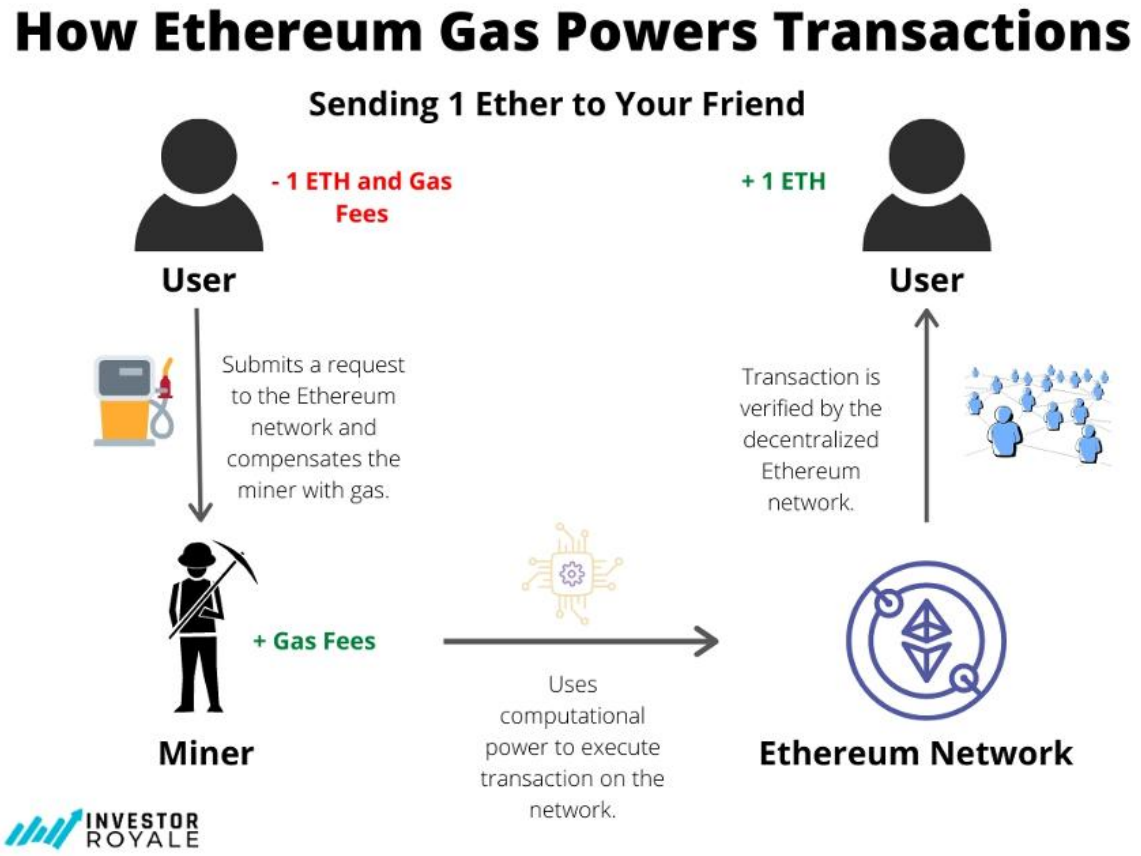
+ Ethereum Virtual Machine

+ Gas Value

+ DApps

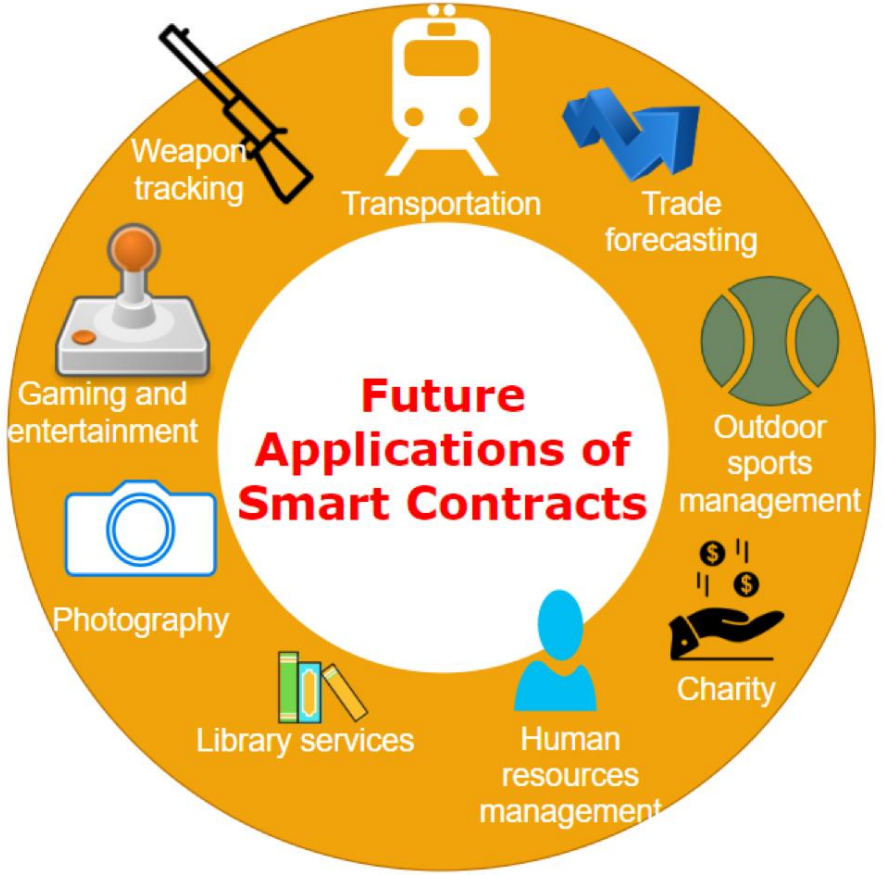
+ Ethereum Ecosystem

+ Web3-Ethereum Javascript API



Một số nhóm ứng dụng phổ biến:

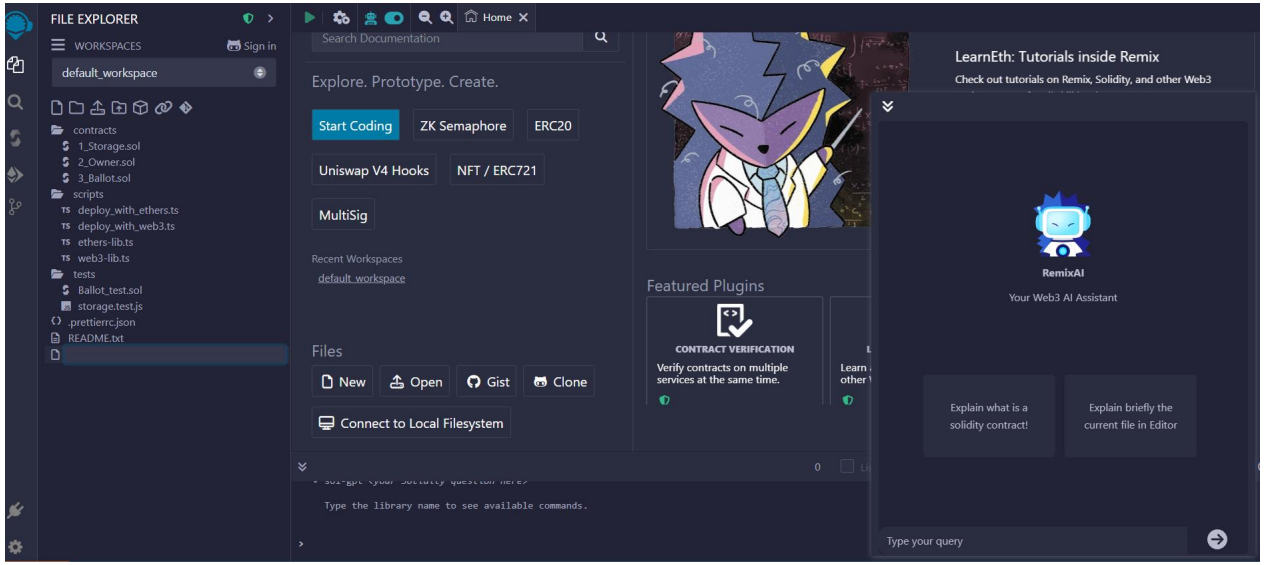
- Chức thực số
- Chuỗi cung ứng
- Bất động sản
- Bảo hiểm
- Thuế
- Công nghiệp giải trí
- Ngân hàng
- AI & Smart home
- Y tế



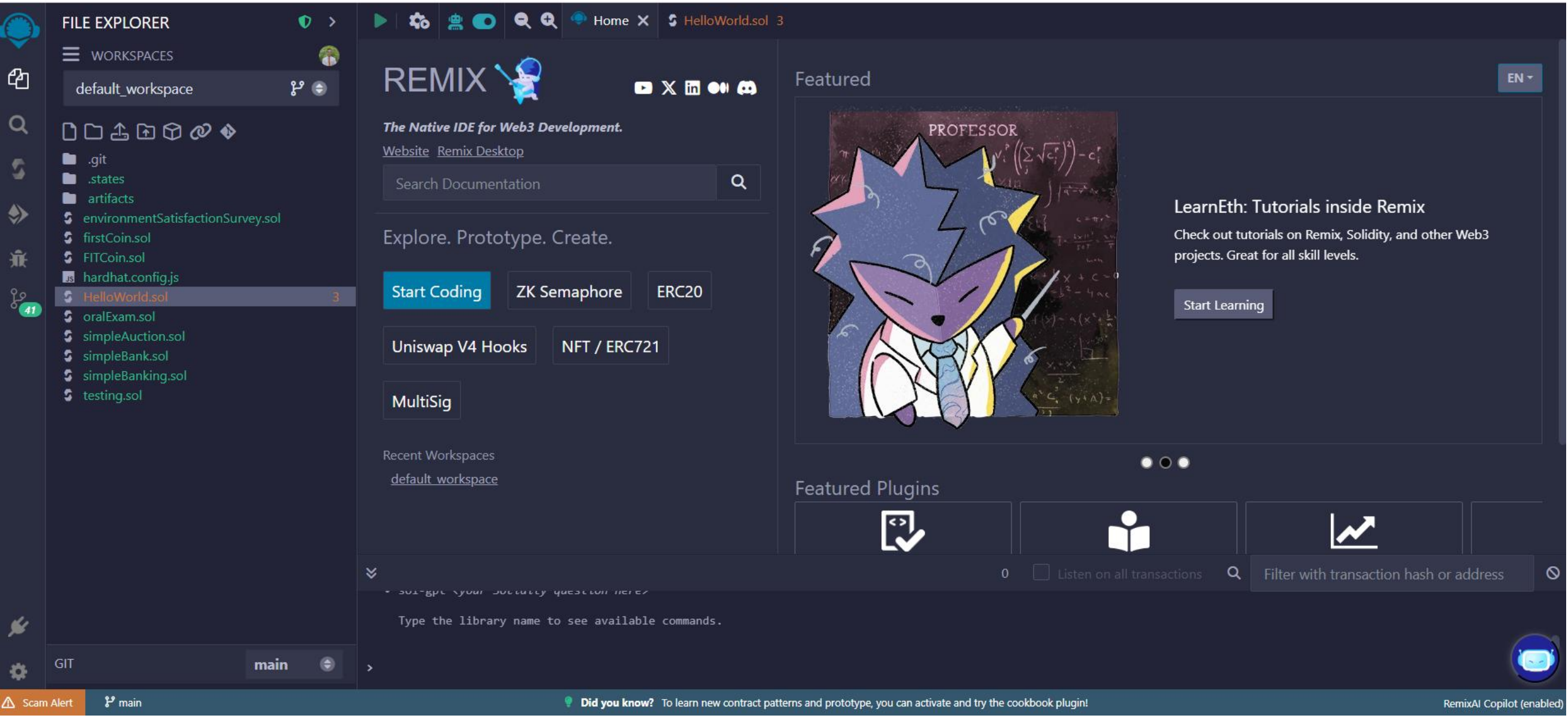
ĐẠI NAM
UNIVERSITY

NGÔN NGỮ LẬP TRÌNH SOLIDITY

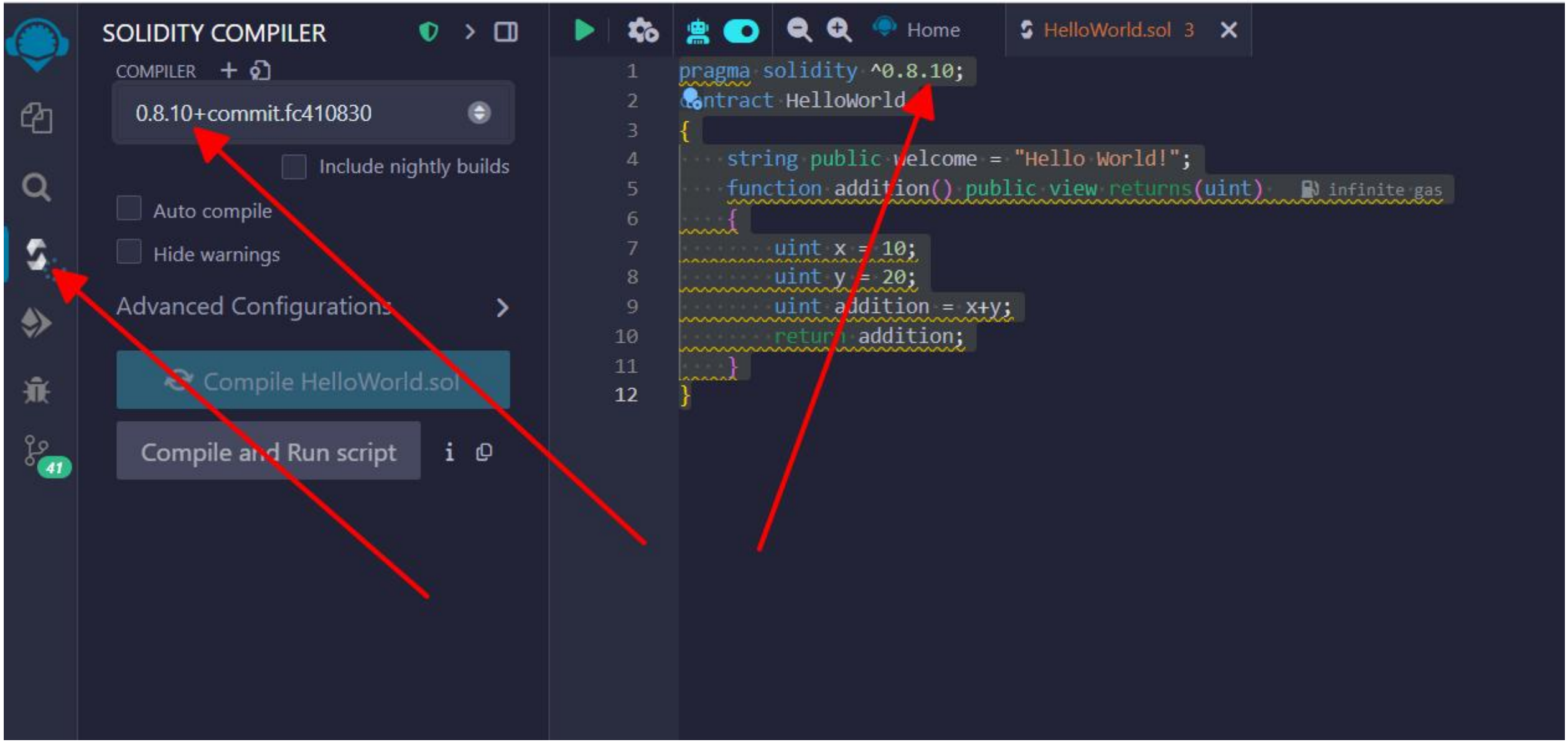
- **Solidity** được đề xuất cùng nền tảng Ethereum, hỗ trợ lập trình OOP như C++, Pyhthon, Javascript và sử dụng cho lập trình HĐTM.
- **Lập trình** trên IDE Remix.
- **Chạy** trên môi trường Ethereum Virtual Machine.

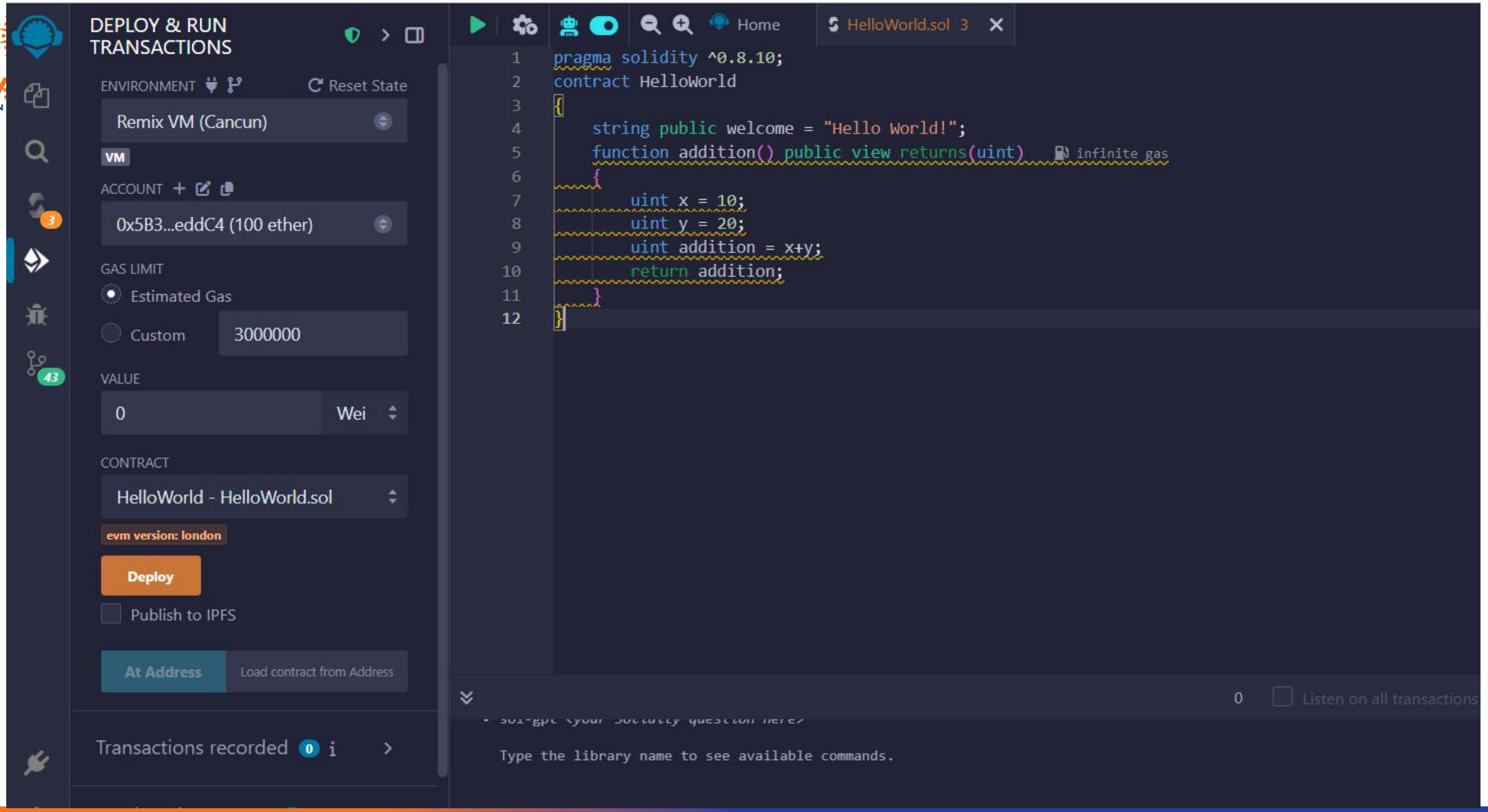


Online: https://remix.ethereum.org/


FILE EXPLORER
WORKSPACES
default_workspace
.git
.states
artifacts
environmentSatisfactionSurvey.sol
firstCoin.sol
FITCoin.sol
hardhat.config.js
HelloWorld.sol 3
oralExam.sol
simpleAuction.sol
simpleBank.sol
simpleBanking.sol
testing.sol
GIT
main
Home
HelloWorld.sol 3
REMIX
The Native IDE for Web3 Development.
Website
Remix Desktop
Search Documentation
Explore. Prototype. Create.
Start Coding
ZK Semaphore
ERC20
Uniswap V4 Hooks
NFT / ERC721
MultiSig
Recent Workspaces
default_workspace
Featured
EN
PROFESSOR
LearnEth: Tutorials inside Remix
Check out tutorials on Remix, Solidity, and other Web3 projects. Great for all skill levels.
Start Learning
Featured Plugins
0
Listen on all transactions
Filter with transaction hash or address
Type the library name to see available commands.
Scam Alert
main
Did you know? To learn new contract patterns and prototype, you can activate and try the cookbook plugin!
RemixAI Copilot (enabled)

```
pragma solidity ^0.8.10;
contract HelloWorld
{
    string public welcome = "Hello World!";
    function addition() public view returns(uint)
    {
        uint x = 10;
        uint y = 20;
        uint addition = x+y;
        return addition;
    }
}
```

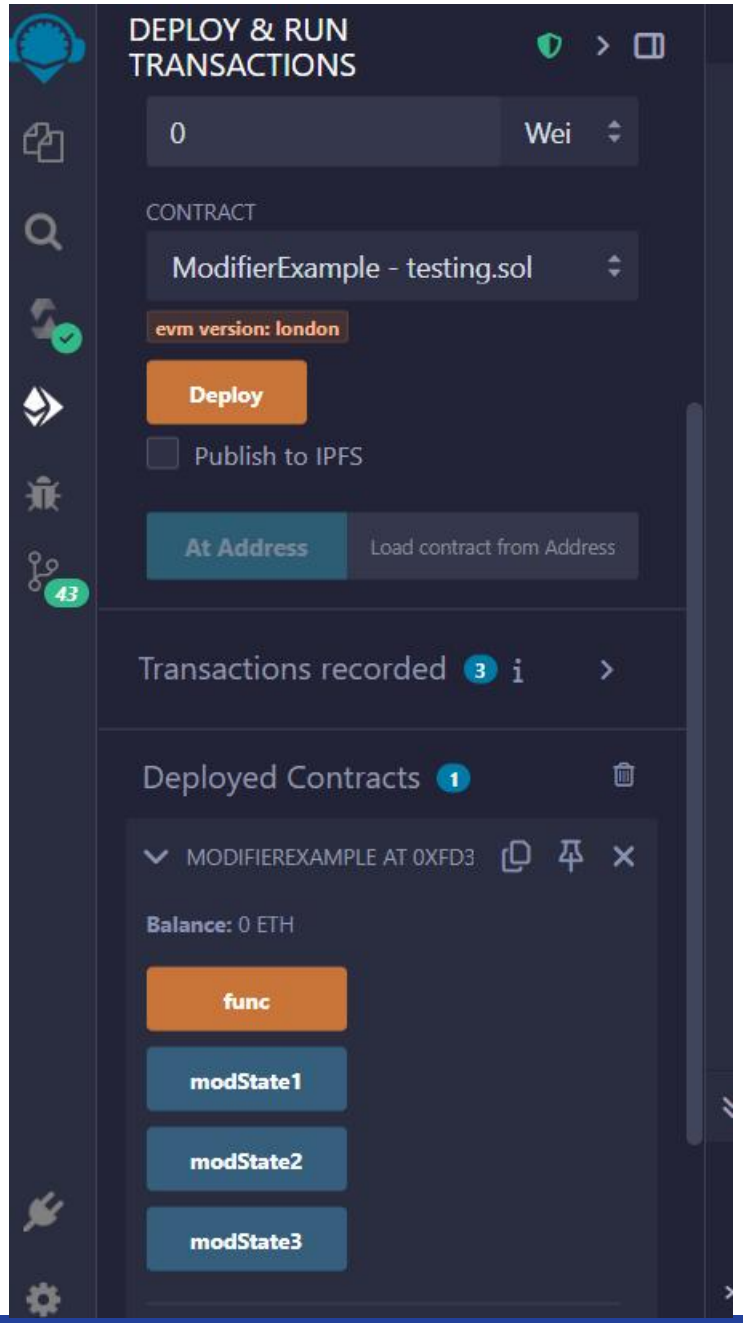

SOLIDITY COMPILER
COMPILER
0.8.10+commit.fc410830
Include nightly builds
Auto compile
Hide warnings
Advanced Configurations
Compile HelloWorld.sol
Compile and Run script
Home
HelloWorld.sol 3
pragma solidity ^0.8.10;
contract HelloWorld
{
string public welcome = "Hello World!";
function addition() public view returns(uint) infinite gas
{
uint x = 10;
uint y = 20;
uint addition = x+y;
return addition;
}
}

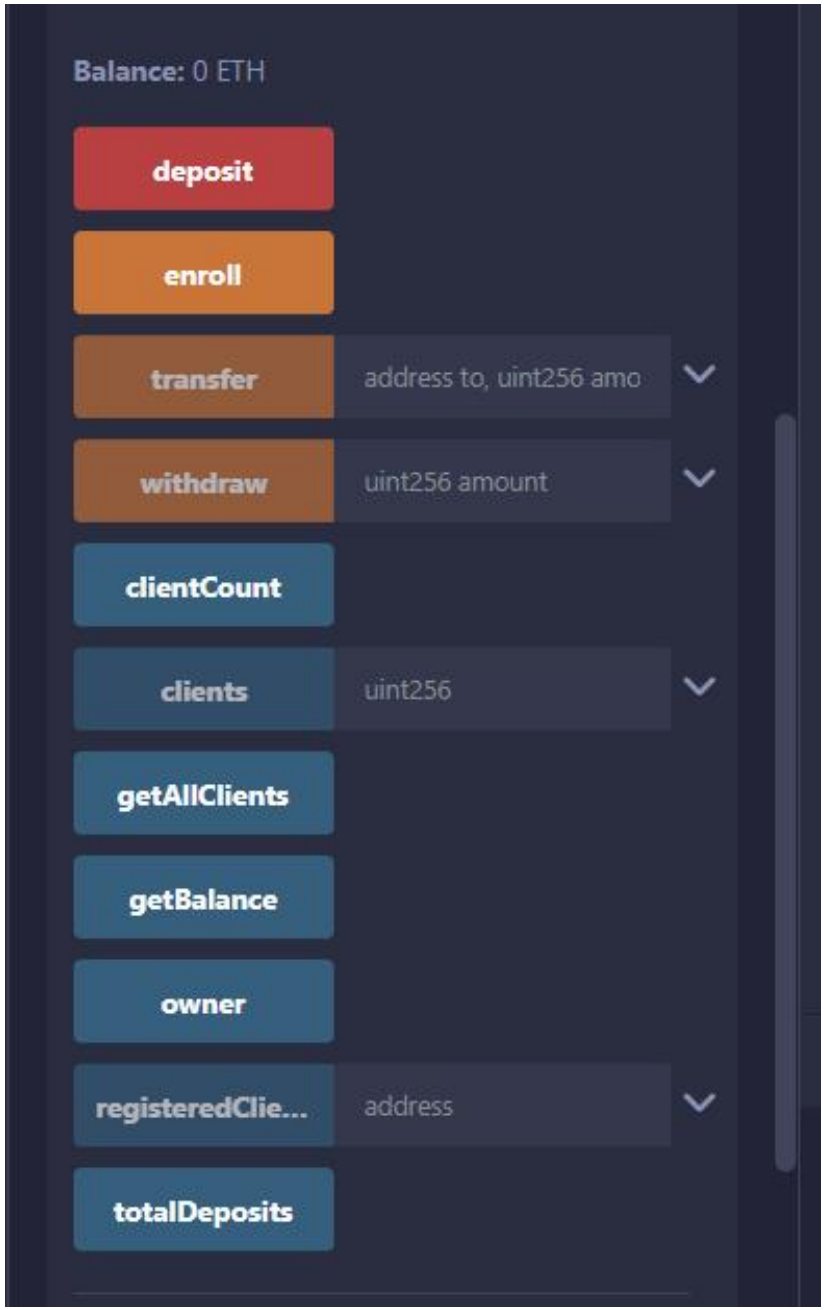

DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Reset State
Remix VM (Cancun)
VM
ACCOUNT
0x5B3...eddC4 (100 ether)
GAS LIMIT
Estimated Gas
Custom
3000000
VALUE
0
Wei
CONTRACT
HelloWorld - HelloWorld.sol
evm version: london
Deploy
Publish to IPFS
At Address
Load contract from Address
Transactions recorded 0
Home
HelloWorld.sol 3
pragma solidity ^0.8.10;
contract HelloWorld
{
string public welcome = "Hello World!";
function addition() public view returns(uint)
infinite gas
{
uint x = 10;
uint y = 20;
uint addition = x+y;
return addition;
}
}
0
Listen on all transactions
Type the library name to see available commands.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract ModifierExample {
    uint public modState1;
    uint public modState2;
    uint public modState3;

    modifier modA() {
        modState1 = modState1 + 1;
        _;
    }
    modifier modB() {
        modState2 = modState2 + 1;
        _;
        modState2 = modState2 + 1;
        _;
    }
    function func() public modA modB {
        modState3 = modState3 + 1;
    }
}
```






```
pragma solidity ^0.8.0;

contract simpleBanking {

    address public owner;

    uint public clientCount;

    mapping(address => uint) private balances;

    mapping(address => bool) public registeredClients;

    address[] public clients;

    event Enrolled(address client);

    event Deposited(address indexed client, uint amount);

    event Withdrawn(address indexed client, uint amount);

    event Transferred(address from, address to, uint amount);
```

1. NFT là gì?

2. NFT hoạt động thế nào?

3. Ứng dụng của NFT?

BÀI TOÁN KIỂM SOÁT BẢN QUYỀN ALBUM ẢNH

Thiết kế hoạt động, chức năng của ứng dụng kiểm soát bản quyền album ảnh.

Thiết kế, lập trình HĐTM về quản lý văn bằng tốt nghiệp của một trường học.

Yêu cầu:

- Thuyết minh thiết kế
- Phân tích Source Code
- Vận hành

Gợi ý thông tin: Số VB, Họ tên, Ngày sinh, Ngành TN, Trường tốt TN, Thời gian TN

DiplomaManager Smart Contract

1. Cài đặt Yêu Cầu

Trình duyệt

- Chrome, Edge hoặc Brave

Plugin MetaMask

- Tải từ: MetaMask
- Tạo ví mới hoặc import ví cũ (có backup phrase)

Remix IDE

- Truy cập: Remix IDE
- Không cần cài đặt, chạy ngay trên trình duyệt

Nhận ETH Test (Sepolia)

- Vào faucet: Sepolia Faucet hoặc Quicknode Faucet hoặc Quicknode Faucet nếu bạn chưa có tiền thật trong ví.
- Paste địa chỉ ví để nhận ETH test

2. Copy Mã Hợp Đồng

- Tạo file mới trong Remix IDE với tên `QLVB.sol`
- Copy toàn bộ mã nguồn từ file `QLVB.sol` vào file vừa tạo

3. Compile Hợp Đồng

- Chuyển sang tab **Solidity Compiler** (biểu tượng hình tam giác)
- Bấm **Compile QLVB.sol**

4. Triển khai Hợp Đồng (Deploy)

- Chuyển sang tab **Deploy & Run Transactions** (hình điện)
- Ở mục **ENVIRONMENT** chọn **Sepolia - MetaMask** (nếu bạn chưa có tiền sepolia thì không cần chọn và có thể dùng tiền mặc định của remix)
- Chọn tài khoản MetaMask có ETH testnet
- Chọn contract `DiplomaManager` trong danh sách contract
- Bấm **Deploy**
- MetaMask sẽ hiện ra, bấm **Xác nhận (Confirm)**
- Sau khi triển khai thành công, xem mục **Deployed Contracts** bên dưới

5. Sử dụng Các Hàm Trong Hợp Đồng

- `issueDiploma(...)` : nhập thông tin sinh viên và gửi với VALUE là >= 0.001 ETH
- `getDiplomaIds()` : xem danh sách các `diplomaId`
- `getDiploma(id)` : xem chi tiết văn bằng với ID
- `withdraw()` : admin rút tiền về ví

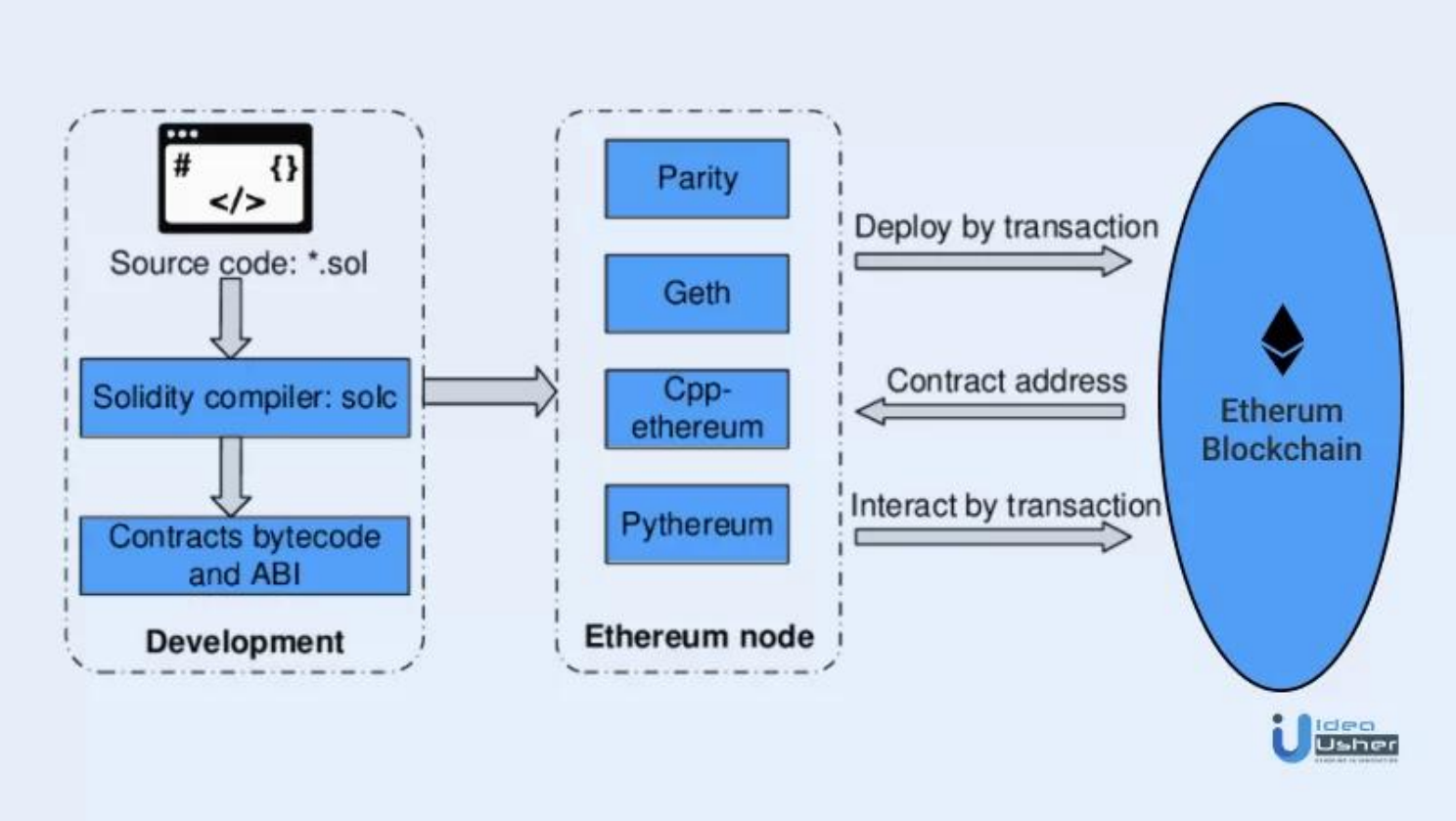
6. Lưu ý

- Phải chọn đúng **Sepolia** trong MetaMask trước khi deploy.
- Khi deploy xong, không thấy contract hiện ra thì do bạn đang ở sai network hoặc chưa Confirm giao dịch.
- Phí gas sẽ do mạng quy định (không cố định), không liên quan đến phí cấp bằng.

TRIỂN KHAI ỨNG DỤNG

1. **Thiết lập** hợp đồng với các điều khoản rõ ràng
2. **Viết code** cho HĐTM với các điều khoản đã thiết lập
3. **Triển khai** lên Blockchain
4. **Vận hành**

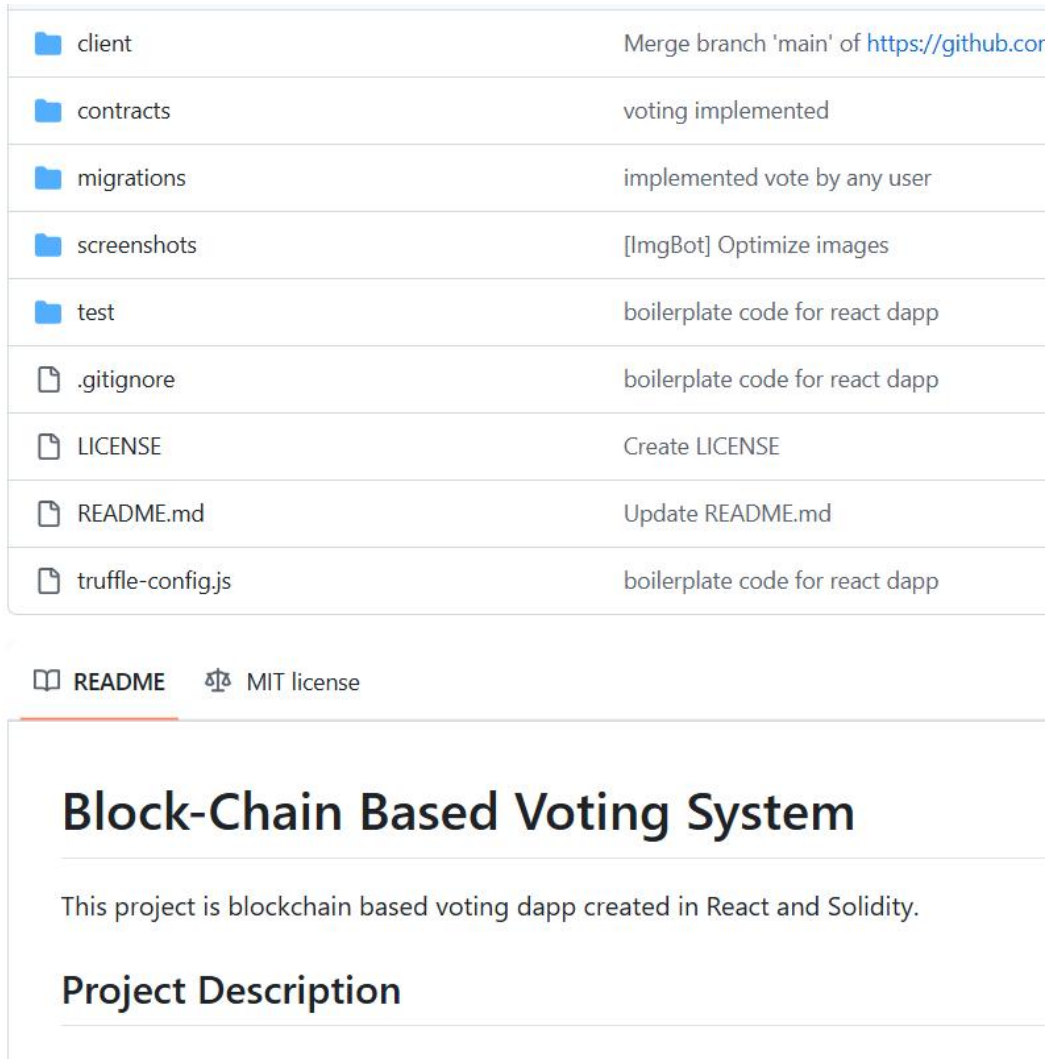
1. **Môi trường:** MetaMask, Remix IDE, Testnet
2. **Viết hợp đồng:** Solidity
3. **Biên dịch:** Solidity Compiler
4. **Triển khai:** Testnet
5. **Kiểm tra** hợp đồng
6. **Tương tác** với hợp đồng



CASE STUDY: VOTING

- Khai thác Source trên Github:

https://github.com/ashishlamsal/voting-dapp




Daily
English
Practice

5.1.1 What is Smart Contract?

A smart contract is an automated digital contract instead of a paper contract that shows a mutual agreement between buyer and seller. The smart contract code and the contained agreement are shared across a distributed, decentralized blockchain network [1]. Smart contracts are executed on the blockchain network, which removes the need for a third party for application-specific transactions [2]. There are many implementation and deployment platforms available where one can create his own smart contract and execute it on the blockchain network.

- Các vấn đề về HĐTM

- Triển khai HĐTM

- Lập trình với Solidity

- Luyện tập với Case Study

Thank You!