COURSE NAME: **BLOCK CHAIN**

GROUP NUMBER: **002**

PROJECT TITLE: **PATIENT HEALTH RECORD**

PROJECT SUBMITTED TO: **ANNA UNIVERSITY / NAAN MUTHALVAN**

YEAR: **II**

DEPARTMENT: **AERO & ECE**

SEMESTER: **IV**

GROUP MEMBERS: **1. JONESMILTON KABIRIYAL Y - 811321101005 (AERO)**

2. GOKUL S - 811321106013 (ECE)

3. ARUN KUMAR M - 811321106009 (ECE)

4. ANANTHARAMAN S - 811321106006 (ECE)

5. YOGESH B - 811321106059 (ECE)

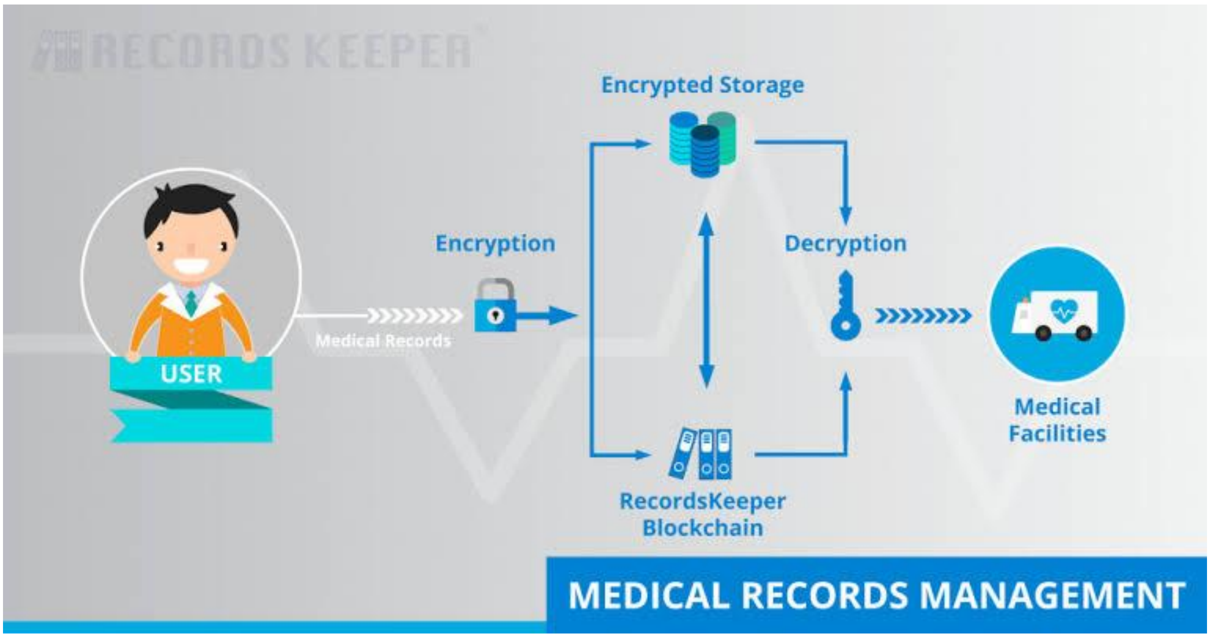
GUIDED BY: **Mrs. J.S. JASLIN**

SPOC NAME: **Dr. P. CHANDRAVATHI**

INTRODUCTION:

Blockchain is database storage using encrypted blocks of data organized in chains for access as a distributed ledger protocol. This means that all parties (nodes) participating in, contributing to, or accessing data in the blockchain decide together which blocks are validly accessed by different users through the use of consensus algorithms. Only a block of data that has been verified to be valid and relevant can be added into the data chain. In most systems, all nodes can have a copy of the complete blockchain so that they know when and what data were accessed and by whom.

Blockchain has advantages of decentralization, data transparency, and privacy and security Decentralization means that all data in blocks are distributed throughout the blockchain network, rather than stored in a central storage location. In addition, whether a block can be added to the chain is decided by all nodes together instead of by one central node. Due to the decentralization of data management, each node can have a complete copy of the blockchain such that all data access is completely transparent to every node in the blockchain, making it impossible to furtively tamper with data without knowledge of the other nodes. Privacy and security of blockchain means encrypting the data stored in the block with hash functions, such as the SHA-256 encryption algorithm. Cryptographic hashes are powerful one-way functions, and it is exceedingly difficult to reverse the plain text from the hash value, protecting blockchain from any third-party interference.



Blockchain is now well applied in the cryptocurrency economic sector. Its features of decentralization, transparency, and security also provide a potentially viable solution to the problem of poor interoperability and data leakage in EHR.

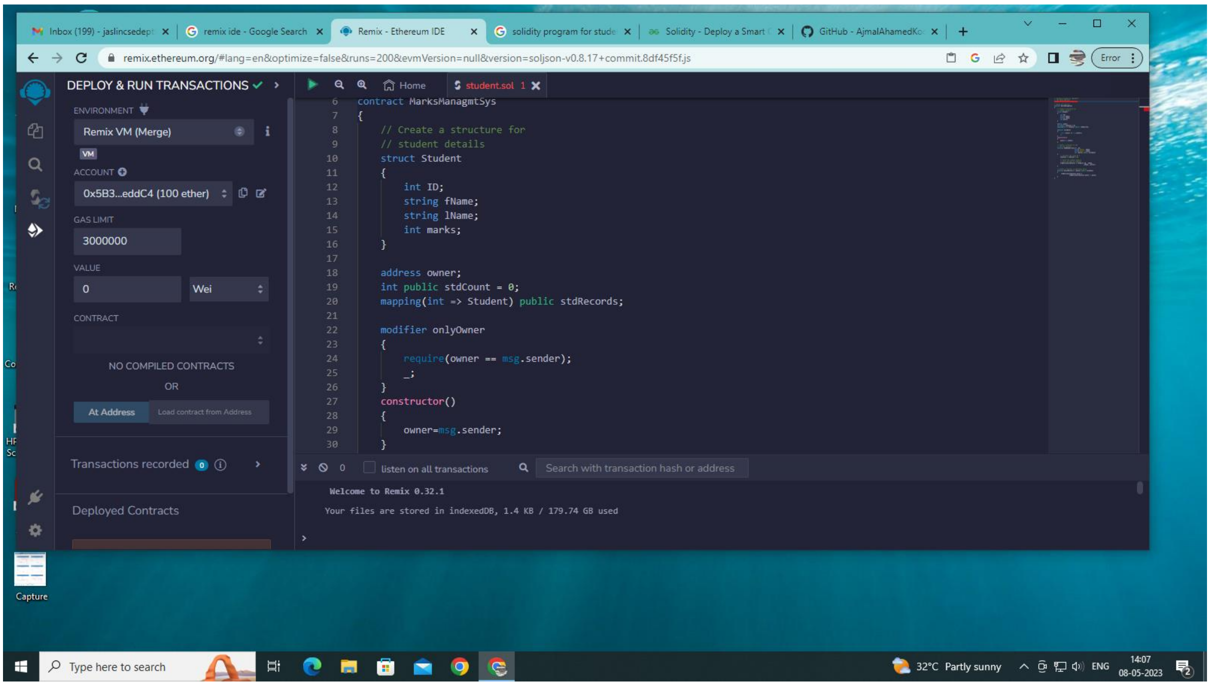
To enable interoperability, the MedRec framework employs a complex series of “smart contracts” on an Ethereum blockchain between patients and visitors, including the registrar contract, the patient-provider relationship contract, and a summary contract to protect patient privacy and standardize the form of EHR. In the MedRec, the exchange of data for the same patient in different medical jurisdictions is simplified by updating viewership permissions on the relevant data pointers of the Ethereum blockchain. Data pointers are cryptographic hash pointing to the data block that contain the data storage information and the cryptographic hash of the data. Such a uniform and simple operation enhances interoperability between different EHR systems. In addition, MedRec only stores data ownership and viewership permissions (cryptographic hash of the record) instead of raw medical records in blocks; thus, the security of user privacy data is also guaranteed.

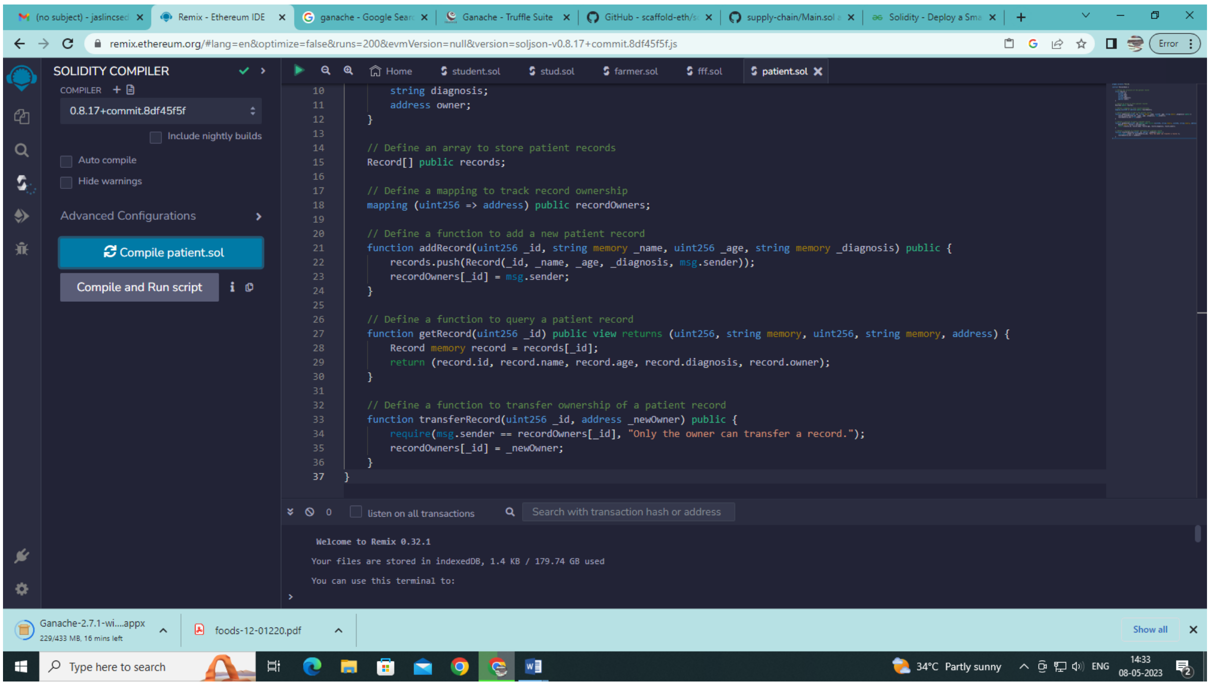
The Healthcare Blockchain System framework is designed for remote monitoring of patients and access to their health records. It uses a private blockchain based on the Ethereum protocol to protect patients’ privacy. At the same time, it maintains a secure record of who has initiated activities on the blockchain and provides details about every data transaction. More specifically, MeDShare closely monitors all actions performed on the data through smart contracts and keys attached to the contracts, so that if a malicious user tries to steal data privacy or tamper with reports generated by the smart contract, their actions will be exposed and access will be restricted or even revoked. Blockchain shows promise to help EHR unify its standards, increase interoperability, and protect patient privacy. However, there are challenges to overcome with blockchain data access and sharing technology.

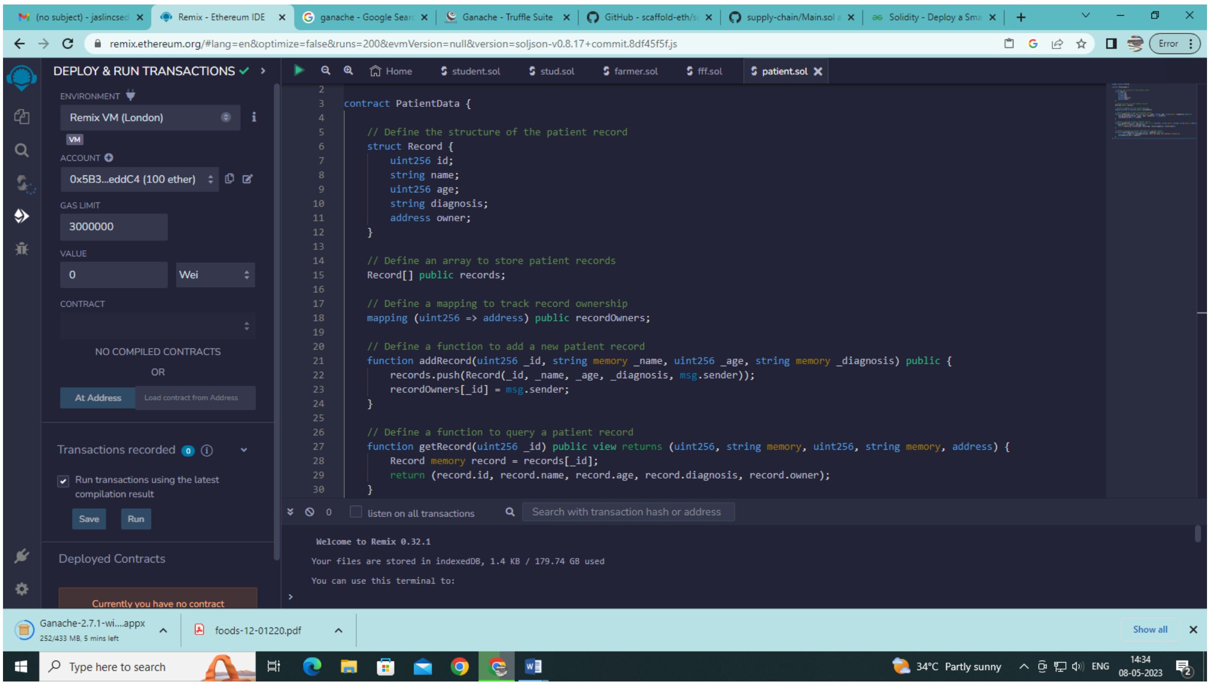
PROCEDURE:

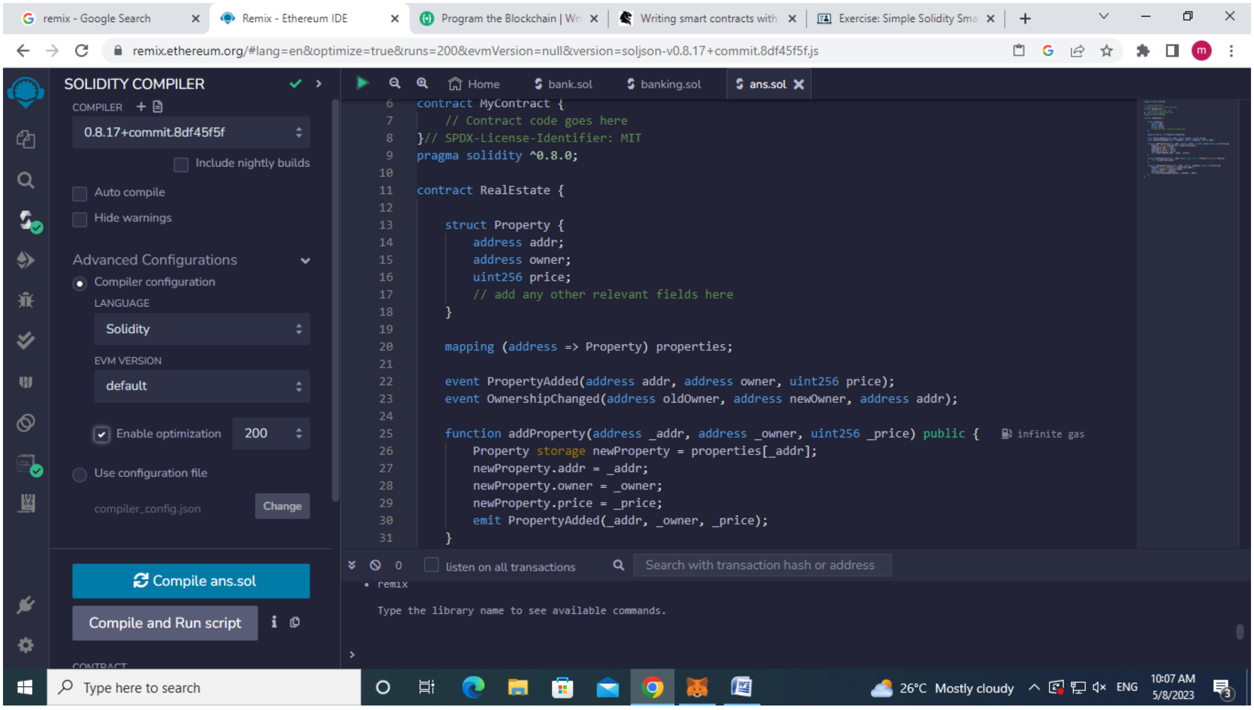
1. Choose a blockchain platform that supports smart contract development, such as Ethereum.
2. Develop a smart contract that defines the structure of the electronic health record and the functions for adding, querying, and changing ownership of the record.
3. Compile the smart contract using Remix IDE, a web-based integrated development environment for Ethereum smart contracts.
4. Connect Remix IDE to a test network such as the Ropsten testnet or a private network to test the smart contract.
5. Deploy the smart contract to the network using Remix IDE and test the functions using Remix IDE's built-in testing tools or by interacting with the contract through a web3.js client like MetaMask.
6. Monitor the gas usage of the smart contract functions to optimize for efficiency and cost-effectiveness.

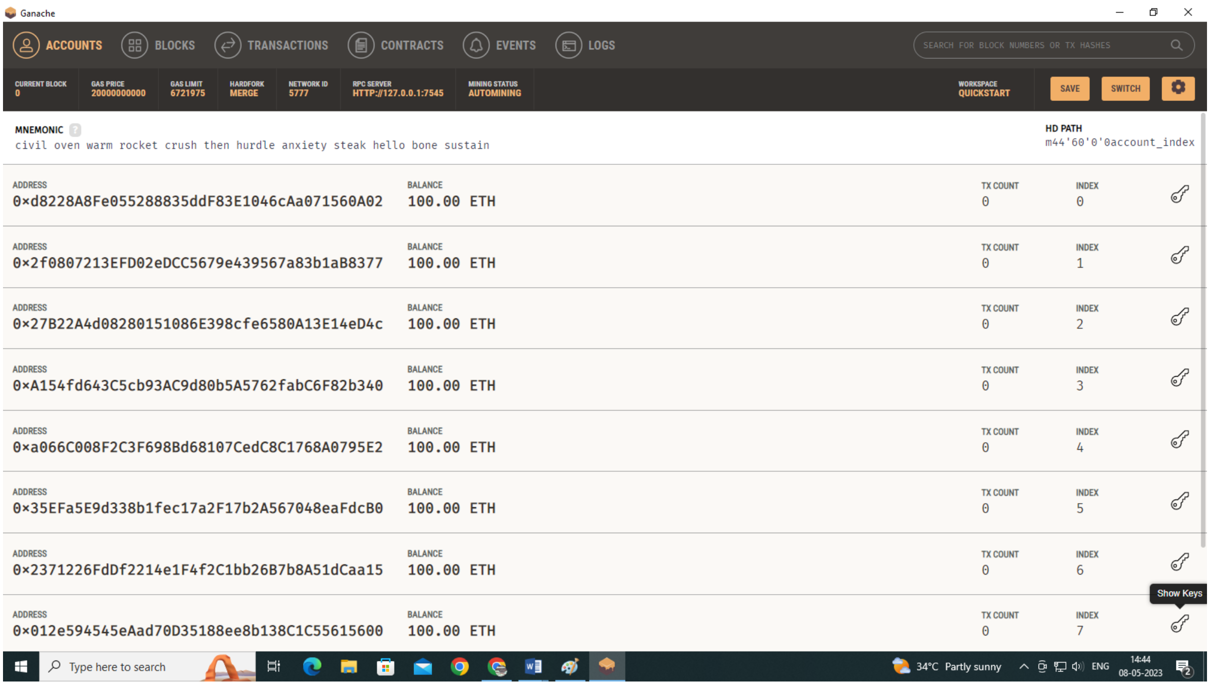
SCREENSHOTS:

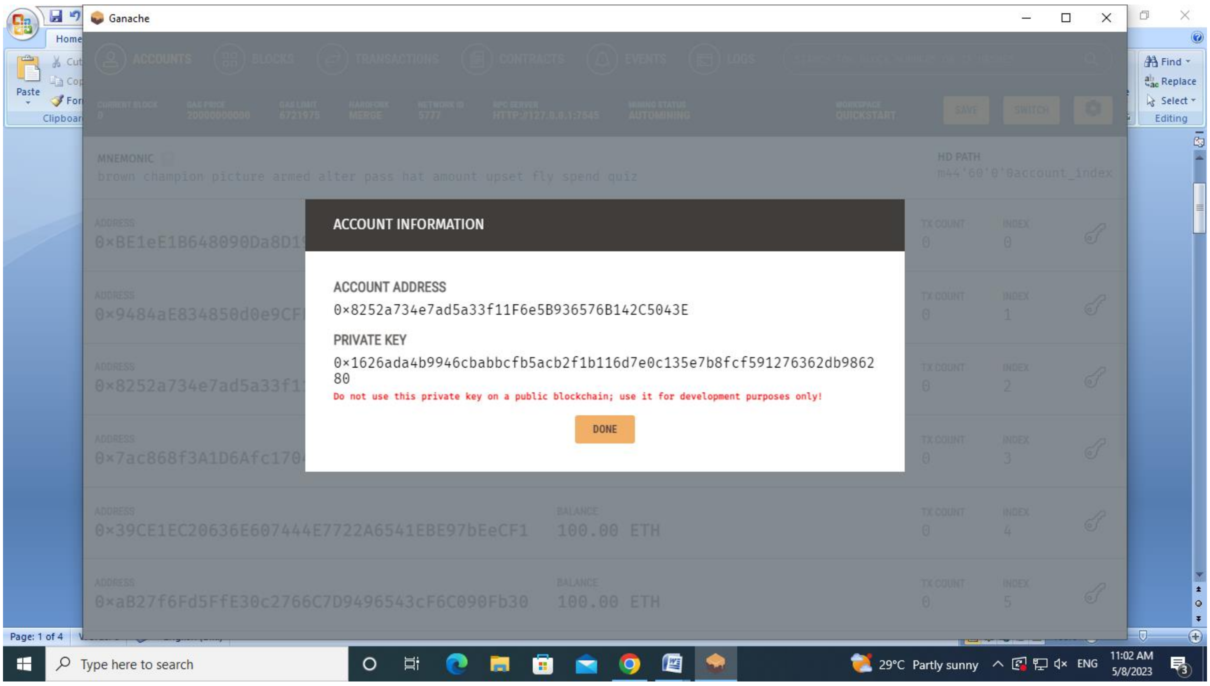


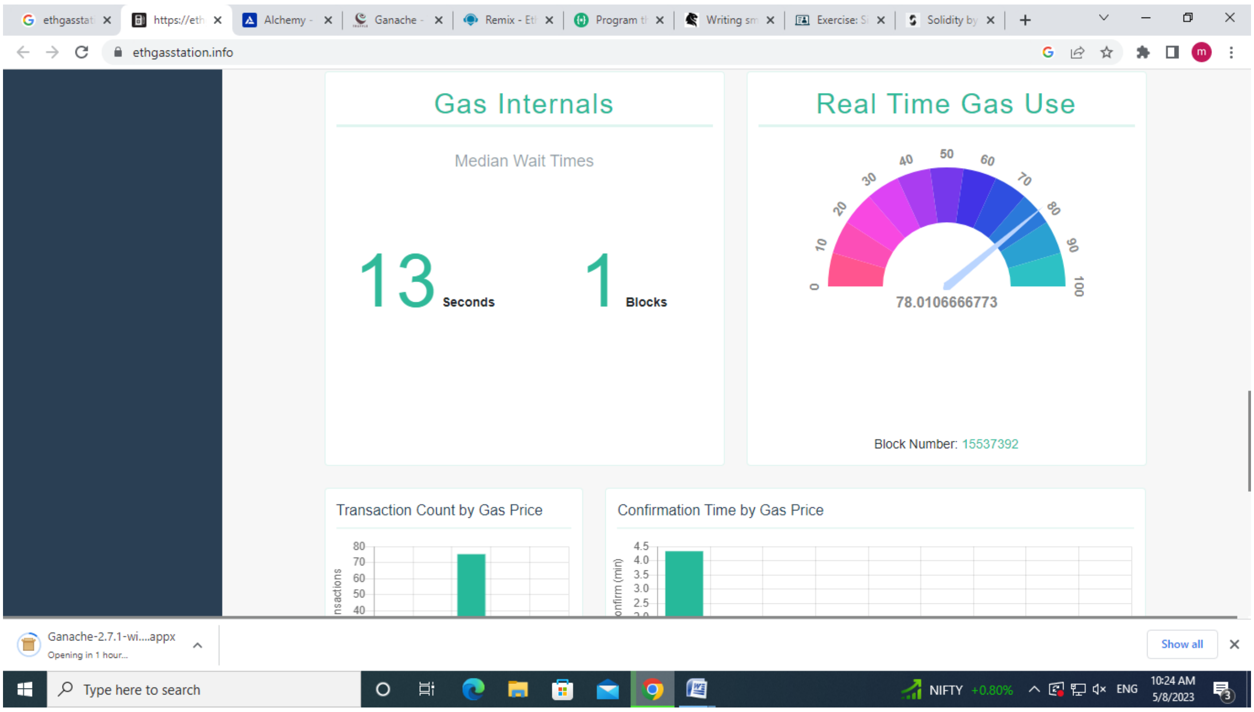

SOLIDITY COMPILER
COMPILER
0.8.17+commit.8df45f5f
Include nightly builds
Auto compile
Hide warnings
Advanced Configurations
Compile patient.sol
Compile and Run script
Home
student.sol
stud.sol
farmer.sol
fff.sol
patient.sol
string diagnosis;
address owner;
}
// Define an array to store patient records
Record[] public records;
// Define a mapping to track record ownership
mapping (uint256 => address) public recordOwners;
// Define a function to add a new patient record
function addRecord(uint256 _id, string memory _name, uint256 _age, string memory _diagnosis) public {
records.push(Record(_id, _name, _age, _diagnosis, msg.sender));
recordOwners[_id] = msg.sender;
}
// Define a function to query a patient record
function getRecord(uint256 _id) public view returns (uint256, string memory, uint256, string memory, address) {
Record memory record = records[_id];
return (record.id, record.name, record.age, record.diagnosis, record.owner);
}
// Define a function to transfer ownership of a patient record
function transferRecord(uint256 _id, address _newOwner) public {
require(msg.sender == recordOwners[_id], "Only the owner can transfer a record.");
recordOwners[_id] = _newOwner;
}
}
listen on all transactions
Search with transaction hash or address
Welcome to Remix 0.32.1
Your files are stored in indexedDB, 1.4 KB / 179.74 GB used
You can use this terminal to:

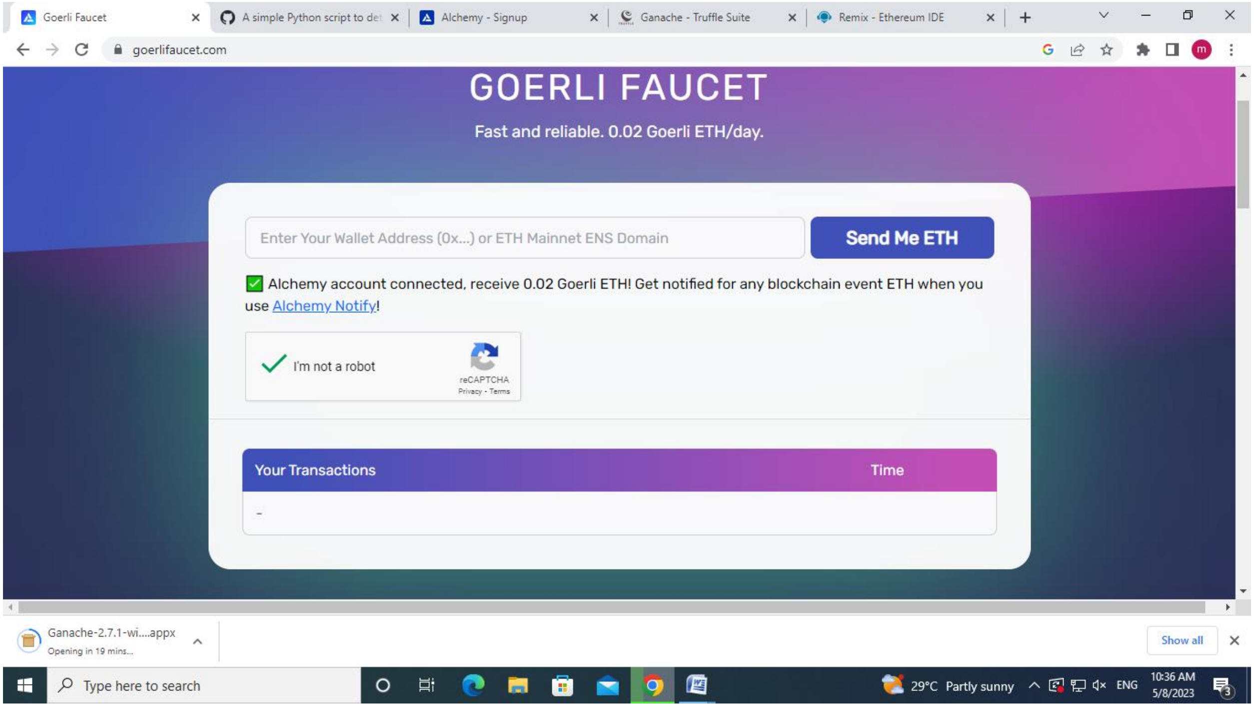

DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Remix VM (London)
ACCOUNT
0x5B3...eddC4 (100 ether)
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT
NO COMPILED CONTRACTS
OR
At Address
Load contract from Address
Transactions recorded
Run transactions using the latest compilation result
Save
Run
Deployed Contracts
Currently you have no contract
contract PatientData {
// Define the structure of the patient record
struct Record {
uint256 id;
string name;
uint256 age;
string diagnosis;
address owner;
}
// Define an array to store patient records
Record[] public records;
// Define a mapping to track record ownership
mapping (uint256 => address) public recordOwners;
// Define a function to add a new patient record
function addRecord(uint256 _id, string memory _name, uint256 _age, string memory _diagnosis) public {
records.push(Record(_id, _name, _age, _diagnosis, msg.sender));
recordOwners[_id] = msg.sender;
}
// Define a function to query a patient record
function getRecord(uint256 _id) public view returns (uint256, string memory, uint256, string memory, address) {
Record memory record = records[_id];
return (record.id, record.name, record.age, record.diagnosis, record.owner);
}
Welcome to Remix 0.32.1
Your files are stored in indexedDB, 1.4 KB / 179.74 GB used
You can use this terminal to:

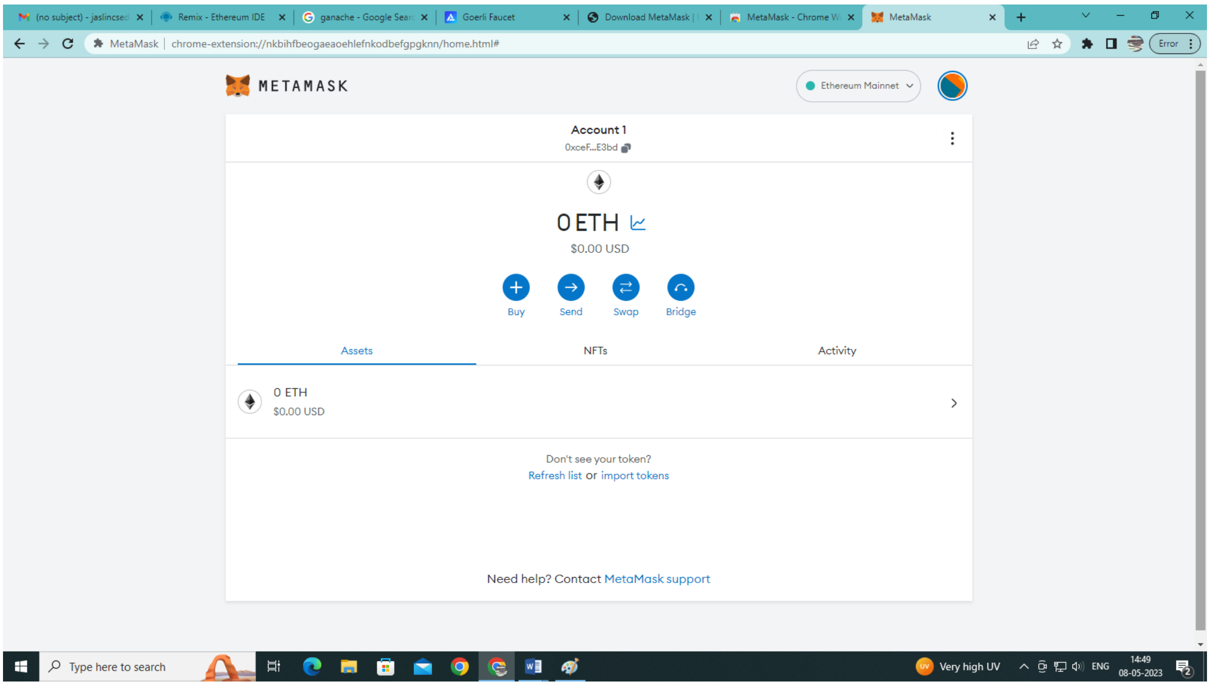

SOLIDITY COMPILER
COMPILER
0.8.17+commit.8df45f5f
Include nightly builds
Auto compile
Hide warnings
Advanced Configurations
Compiler configuration
LANGUAGE
Solidity
EVM VERSION
default
Enable optimization
200
Use configuration file
compiler_config.json
Change
Compile ans.sol
Compile and Run script
Home
bank.sol
banking.sol
ans.sol
contract MyContract {
// Contract code goes here
}// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract RealEstate {
struct Property {
address addr;
address owner;
uint256 price;
// add any other relevant fields here
}
mapping (address => Property) properties;
event PropertyAdded(address addr, address owner, uint256 price);
event OwnershipChanged(address oldOwner, address newOwner, address addr);
function addProperty(address _addr, address _owner, uint256 _price) public {
infinite gas
Property storage newProperty = properties[_addr];
newProperty.addr = _addr;
newProperty.owner = _owner;
newProperty.price = _price;
emit PropertyAdded(_addr, _owner, _price);
}
listen on all transactions
Search with transaction hash or address
remix
Type the library name to see available commands.
Type here to search
26°C Mostly cloudy
ENG
10:07 AM
5/8/2023


Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
CURRENT BLOCK 0
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MERGE
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE QUICKSTART
SAVE
SWITCH
MNEMONIC
civil oven warm rocket crush then hurdle anxiety steak hello bone sustain
HD PATH
m44'60'0'0account_index
ADDRESS 0×d8228A8Fe055288835dDF83E1046cAa071560A02 BALANCE 100.00 ETH TX COUNT 0 INDEX 0
ADDRESS 0×2f0807213EFD02eDCC5679e439567a83b1aB8377 BALANCE 100.00 ETH TX COUNT 0 INDEX 1
ADDRESS 0×27B22A4d08280151086E398cfe6580A13E14eD4c BALANCE 100.00 ETH TX COUNT 0 INDEX 2
ADDRESS 0×A154fd643C5cb93AC9d80b5A5762fabC6F82b340 BALANCE 100.00 ETH TX COUNT 0 INDEX 3
ADDRESS 0×a066C008F2C3F698Bd68107CedC8C1768A0795E2 BALANCE 100.00 ETH TX COUNT 0 INDEX 4
ADDRESS 0×35EFa5E9d338b1fec17a2F17b2A567048eaFdcB0 BALANCE 100.00 ETH TX COUNT 0 INDEX 5
ADDRESS 0×2371226FdDf2214e1F4f2C1bb26B7b8A51dCaa15 BALANCE 100.00 ETH TX COUNT 0 INDEX 6
Show Keys
ADDRESS 0×012e594545eAad70D35188ee8b138C1C55615600 BALANCE 100.00 ETH TX COUNT 0 INDEX 7
Type here to search
34°C Partly sunny
ENG
14:44
08-05-2023


Ganache
ACCOUNT INFORMATION
ACCOUNT ADDRESS
0x8252a734e7ad5a33f11F6e5B936576B142C5043E
PRIVATE KEY
0x1626ada4b9946cbabbcfb5acb2f1b116d7e0c135e7b8fcf591276362db9862
80
Do not use this private key on a public blockchain; use it for development purposes only!
DONE
MNEMONIC
brown champion picture armed alter pass hat amount upset fly spend quiz
HD PATH
m44'60'0'0account_index
0x39CE1EC20636E607444E7722A6541EBE97bEeCF1
100.00 ETH
0xaB27f6Fd5FfE30c2766C7D9496543cF6C090Fb30
100.00 ETH
Type here to search
29°C Partly sunny
11:02 AM
5/8/2023


ethgasstation.info
Gas Internals
Median Wait Times
13 Seconds
1 Blocks
Real Time Gas Use
78.0106666773
Block Number: 15537392
Transaction Count by Gas Price
Confirmation Time by Gas Price
Ganache-2.7.1-wi....appx
Opening in 1 hour...
Show all
Type here to search
NIFTY +0.80%
10:24 AM
5/8/2023


Goerli Faucet
A simple Python script to de
Alchemy - Signup
Ganache - Truffle Suite
Remix - Ethereum IDE
goerlifaucet.com
GOERLI FAUCET
Fast and reliable. 0.02 Goerli ETH/day.
Enter Your Wallet Address (0x...) or ETH Mainnet ENS Domain
Send Me ETH
Alchemy account connected, receive 0.02 Goerli ETH! Get notified for any blockchain event ETH when you use Alchemy Notify!
I'm not a robot
reCAPTCHA
Privacy - Terms
Your Transactions
Time
Ganache-2.7.1-wi....appx
Opening in 19 mins...
Show all
Type here to search
29°C Partly sunny
ENG
10:36 AM
5/8/2023


(no subject) - jaslincse
Remix - Ethereum IDE
ganache - Google Sear
Goerli Faucet
Download MetaMask
MetaMask - Chrome W
MetaMask
MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#
Error
METAMASK
Ethereum Mainnet
Account 1
0xceF...E3bd
0 ETH
$0.00 USD
Buy
Send
Swap
Bridge
Assets
NFTs
Activity
0 ETH
$0.00 USD
Don't see your token?
Refresh list or import tokens
Need help? Contact MetaMask support
Type here to search
Very high UV
ENG
14:49
08-05-2023