PRACTICAL NO : 1

AIM: Demonstrate Blockchain characteristics [DIT].

→ characteristics of BLC:

- Immutable
- auditability
- low latency
- Secure
- Decentralized
- Distributed
- Unanimous
- Consensus
- Scalability and Global Accessibility
- Efficiency and Cost Reduction

1. Immutable :

- Immutability means that the blockchain is a permanent and unalterable network. Blockchain technology functions through a collection of nodes. Once a transaction is recorded on the blockchain, it cannot be modified or deleted. This makes the blockchain an immutable and tamper-proof ledger that provides a high degree of security and trust.
- Every node in the network has a copy of the digital ledger. To add a transaction every node checks the validity of the transaction and if the majority of the nodes think that it is a valid transaction then it is added to the network. This means that without the approval of a majority of nodes no one can add any transaction blocks to the ledger.
- Any validated records are irreversible and cannot be changed. This means that any user on the network won't be able to edit, change or delete it.

2. Auditability :

- Record timestamp and persistent information allow ones to easily verify and trace previous records through nodes in a Blockchain network. The degree of auditability depends on types of Blockchain systems and their implementations. Private Blockchains are the least auditable as nodes are administrated by one entity, permissioned Blockchains come second in which some agreements, such as encrypted data, may prevent information to be fully auditable, and public Blockchains are the highest as nodes are truly decentralized.

3. Low latency :

- Similar to how internet connections need both high bandwidth and low latency to deliver speedy services, blockchain platforms need both high throughput and low transaction latency to do the same.
- The latency of a blockchain network is the time between transaction submission and confirmation by the network, whereas the latency of an exchange is the transaction processing time taking place on the network.

4. Secure :

- Blockchain technology is considered more secure than its contemporaries because of lack of a single point of failure. Blockchain operates on a well-distributed network of nodes, hence data at all times is circulated through not one but multiple nodes, which makes sure that even if one node is hacked or faulty in any way the integrity of the original data will not be compromised.
- All the records in the blockchain are individually encrypted. Using encryption adds another layer of security to the entire process on the blockchain network. Since there is no central authority, it does not mean that one can simply add, update or delete data on the network.
- Every information on the blockchain is hashed cryptographically which means that every piece of data has a unique identity on the network. All the blocks contain a unique hash of their own and the hash of the previous block. Due to this property, the blocks are cryptographically linked with each other. Any attempt to modify the data means to change all the hash IDs which is quite impossible.
- The cryptographic mechanisms used within the blockchain make it highly secure, safeguarding transactions and sensitive data from unauthorized access.

5. Decentralized :

- Blockchain technology is a decentralized system, which means that there is no central authority controlling the network. Instead, the network is made up of a large number of nodes that work together to verify and validate transactions. Each and every node in the blockchain network will have the same copy of the ledger.
- There is no third-party involved hence no added risk in the system.
- The decentralized nature of blockchain facilitates creating a transparent profile for every participant on the network. Thus, every change is traceable, and more concreate.

6. Distributed :

- All network participants have a copy of the ledger for complete transparency. A public ledger will provide complete information about all the participants on the network and

transactions. The distributed computational power across the computers ensures a better outcome.

- Any change in the ledger will be updated in seconds or minutes and due to no involvement of intermediaries in the blockchain, the validation for the change will be done quickly.

7. Unanimous :

- All the network participants agree to the validity of the records before they can be added to the network. When a node wants to add a block to the network then it must get majority voting otherwise the block cannot be added to the network. A node cannot simply add, update, or delete information from the network. Every record is updated simultaneously and the updations propagate quickly in the network. So it is not possible to make any change without consent from the majority of nodes in the network.

8. Consensus :

- Every blockchain has a consensus to help the network to make quick and unbiased decisions. Consensus is a decision-making algorithm for the group of nodes active on the network to reach an agreement quickly and faster and for the smooth functioning of the system. Nodes might not trust each other but they can trust the algorithm that runs at the core of the network to make decisions. There are many consensus algorithms available each with its pros and cons. Every blockchain must have a consensus algorithm otherwise it will lose its value.

9. Scalability and Global Accessibility :

- Blockchain has the potential to scale to accommodate a vast number of transactions.
- Its distributed nature allows for a global network accessible to anyone with an internet connection. This characteristic enables the inclusion of unbanked populations and facilitates cross-border transactions without relying on traditional financial institutions.

10. Efficiency and Cost Reduction :

- By eliminating intermediaries and streamlining processes through automation, blockchain technology offers increased efficiency and cost reduction. Transactions can be processed faster, settlement times can be significantly reduced, and operational overheads can be minimized, benefiting businesses and individuals alike.

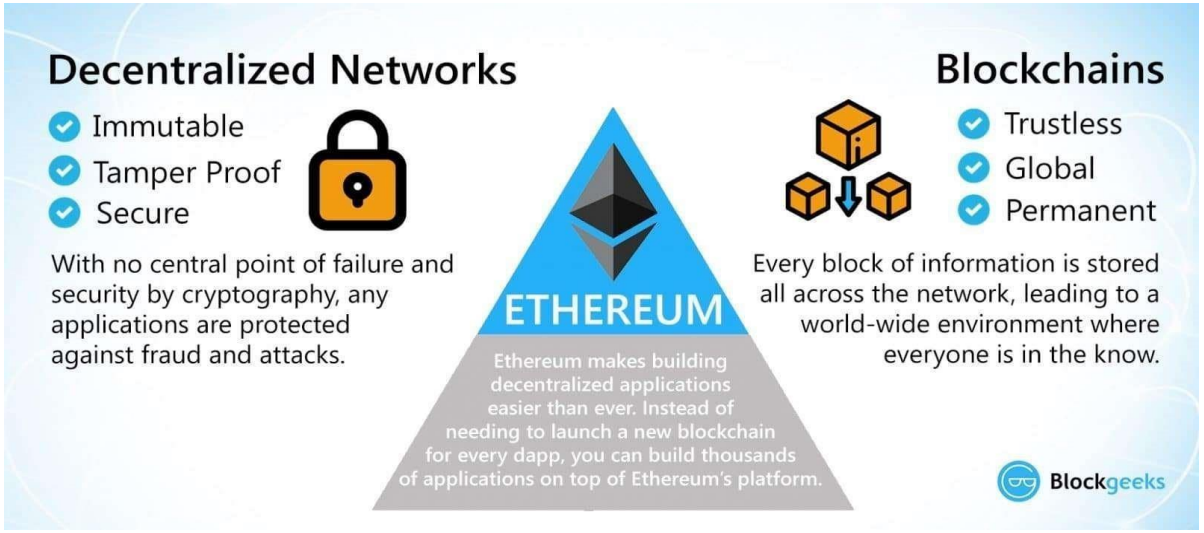
PRACTICAL NO : 2

AIM: Introduction to ETHEREUM tools and Solidity.

→ ETHEREUM :

Ethereum is a decentralized blockchain platform that allows developers to build and deploy smart contracts and decentralized applications (DApps). It introduced the concept of a programmable blockchain, enabling developers to create and run applications with selfexecuting smart contracts.

Ethereum's native cryptocurrency, Ether (ETH), is used as a means of payment for transactions and to incentivize network participants. The platform has gained popularity for its versatility, enabling a wide range of use cases, including finance, gaming, supply chain management.



Tools of ETHEREUM :

- Ethereum Wallet
- Remix IDE
- Truffle
- Web3.js
- Ganache
- Infura
- MetaMask
- Solidity
- Mist
- Geth
- DAppBoard
- Ether.js

1. Ethereum Wallet : Ethereum provides an official wallet called "Ethereum Wallet" or "Mist." It enables users to securely store, manage, and interact with Ether (ETH) and other Ethereum-based tokens. The wallet also serves as a gateway to access decentralized applications (DApps) on the Ethereum network.

2. Remix IDE : Remix is a web-based integrated development environment (IDE) designed for writing, testing, and deploying smart contracts on the Ethereum blockchain. It offers a user-friendly interface with features like code highlighting, debugging, and deployment options, making it a popular choice among developers. Remix IDE is an open-source, JavaScript-based debugging and compiling tool that is primarily used for writing Solidity contracts.

3. Truffle : Truffle is a widely used development framework that simplifies the process of creating, testing, and deploying smart contracts on Ethereum. It provides a suite of tools and libraries, along with a built-in testing framework, to streamline the development workflow.

4. Web3.js : Web3.js is a JavaScript library that allows seamless interaction with the Ethereum blockchain from web applications. It provides a comprehensive set of APIs for developers to interact with smart contracts, send transactions, and retrieve blockchain data.

5. Ganache : Ganache is a local Ethereum blockchain designed for development and testing purposes. It allows developers to create a private Ethereum network with prefunded accounts, enabling them to test and debug smart contracts and decentralized applications without incurring costs on the live network.

6. Infura : Infura is an infrastructure provider offering a scalable API service to access the Ethereum blockchain. It enables developers to interact with the Ethereum network without the need to set up and maintain their own blockchain nodes, providing a convenient solution for building applications and accessing Ethereum data.

7. MetaMask : MetaMask is a browser-based tool designed for Ethereum. In essence, it is a wallet that functions as a browser extension. As a browser extension for all major browsers (Chrome, Firefox, and Opera), MetaMask allows you to interact with the Ethereum framework in a and hassle-free manner.

8. Solidity : Solidity is the primary programming language used to write smart contracts on the Ethereum framework. It is a statically typed, high-level, contract-oriented programming language that draws inspiration from JavaScript, Python, and C++. By the phrase “contract-oriented,” we mean that smart contracts are designed to innately store all the programming logic that occurs within the Ethereum Blockchain.

9. Mist : Mist is an end-user interface for Ethereum. It is explicitly designed for non-technical users and is not only a user-friendly browser interface but also a wallet.

10. Geth : Geth is an implementation of an Ethereum node written in the Go programming language. It also functions as a multi-purpose command-line tool that is imbued with similar functionalities as Mist. It can also perform additional tasks like mining Ether or serving as an RPC endpoint for connecting to the Blockchain over HTTP.

What is Solidity?

Solidity is the smart contract programming language for the Ethereum blockchain network.

Solidity is a high-level programming language designed for developing smart contracts on the Ethereum blockchain. It is specifically tailored for creating decentralized applications (DApps) and defining the behavior of smart contracts.

Solidity draws inspiration from familiar languages like JavaScript and C++, making it accessible for developers. It offers features for defining functions, data structures, and events within smart contracts. Solidity code is compiled into bytecode and executed by the Ethereum Virtual Machine (EVM) to enable the implementation of complex business logic on the Ethereum platform.

Relationship Between Ethereum & Solidity :

- Solidity is a programming language used for ethereum smart contracts
- Solidity coded smart contracts are used for decentralized applications
- Solidity act as the medium between blockchain and end-user
- The solidity enabled API is the only way to interact with ethereum blockchain

Thus the solidity and ethereum are mutually dependent on each other, the solidity can be used to create API for a decentralized application running through the ethereum blockchain. The solidity issued to create the necessary action for each and every transaction executed through the blockchain.

PRACTICAL NO : 3

AIM: Deploy a smart contract for printing Hello World_ using Java Script VM,Injected Web3 and Web3Provider using MetaMask and Ganache.

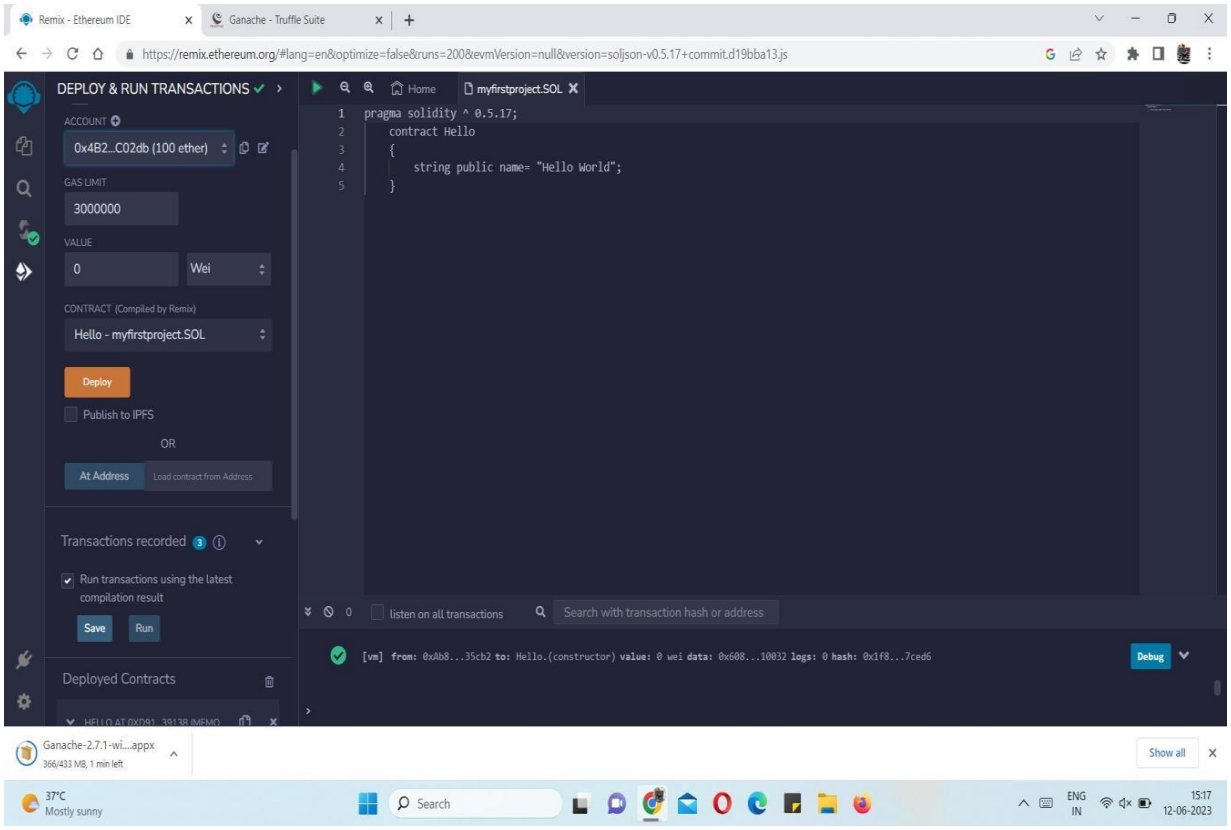
1) Using Remix Ethereum:-

Code:-

```
// SPDX-Licence-Identifier=MIT

pragma solidity ^ 0.5.10;
  contract Hello
  {
     string public name= "Hello World";
  }
```

Output:-

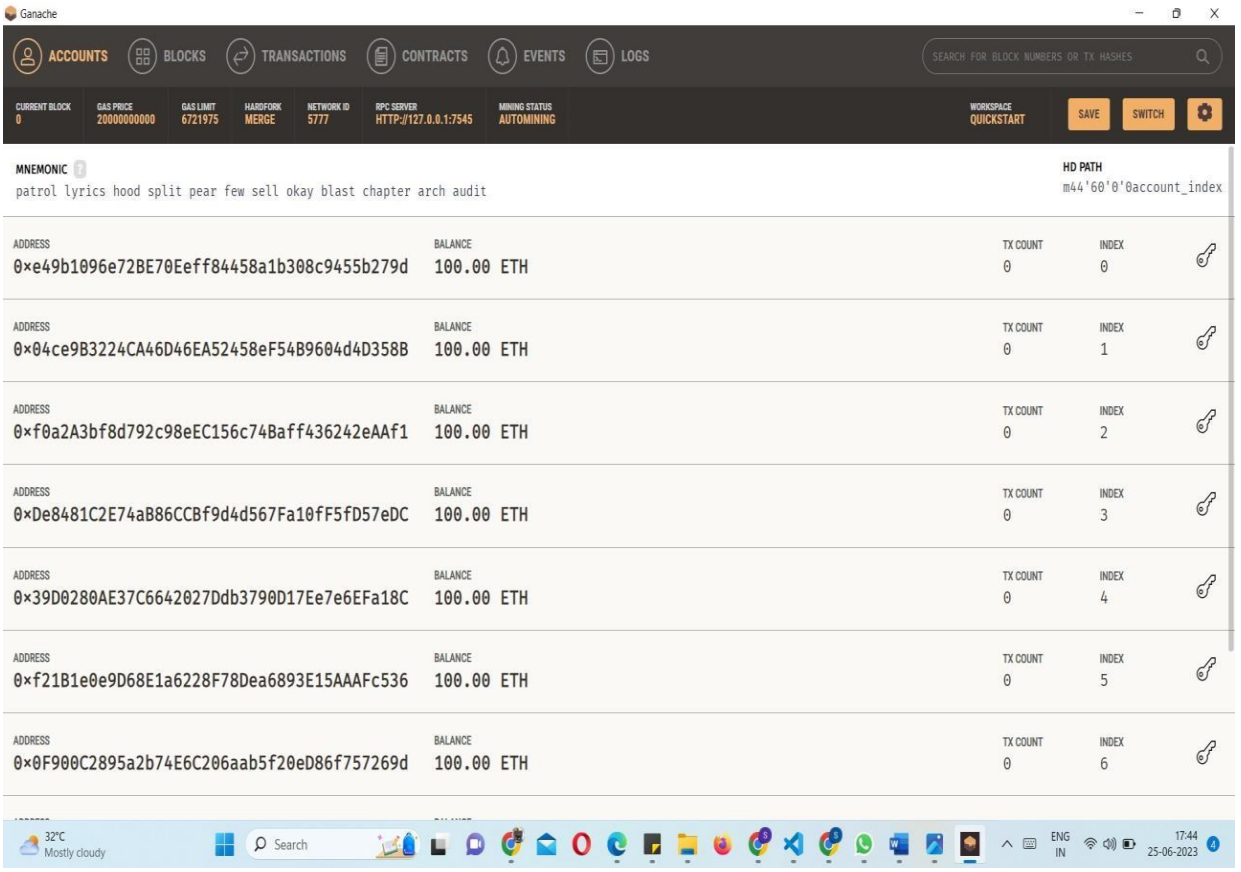


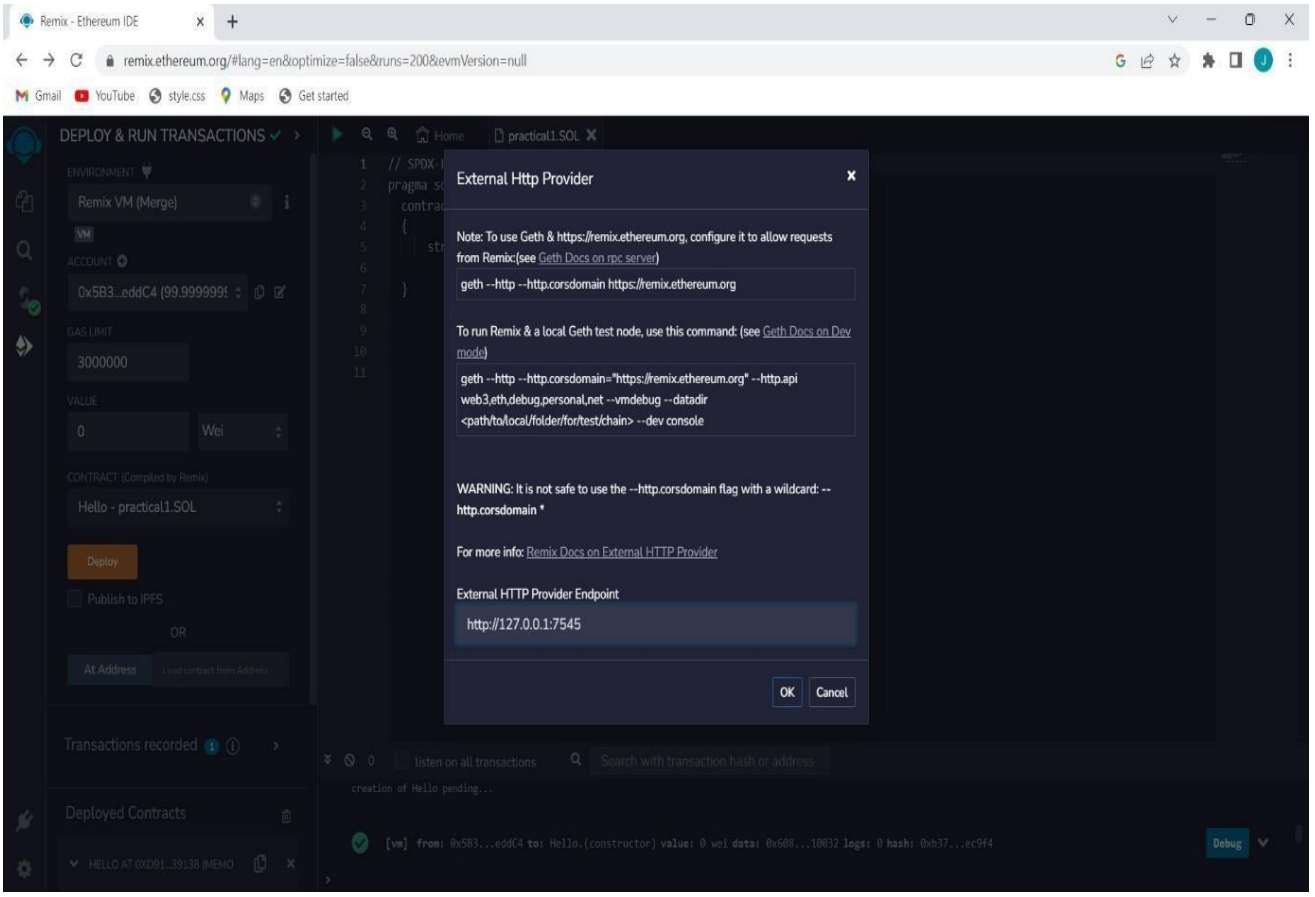
2) Using Remix Ethereum and Ganache :-

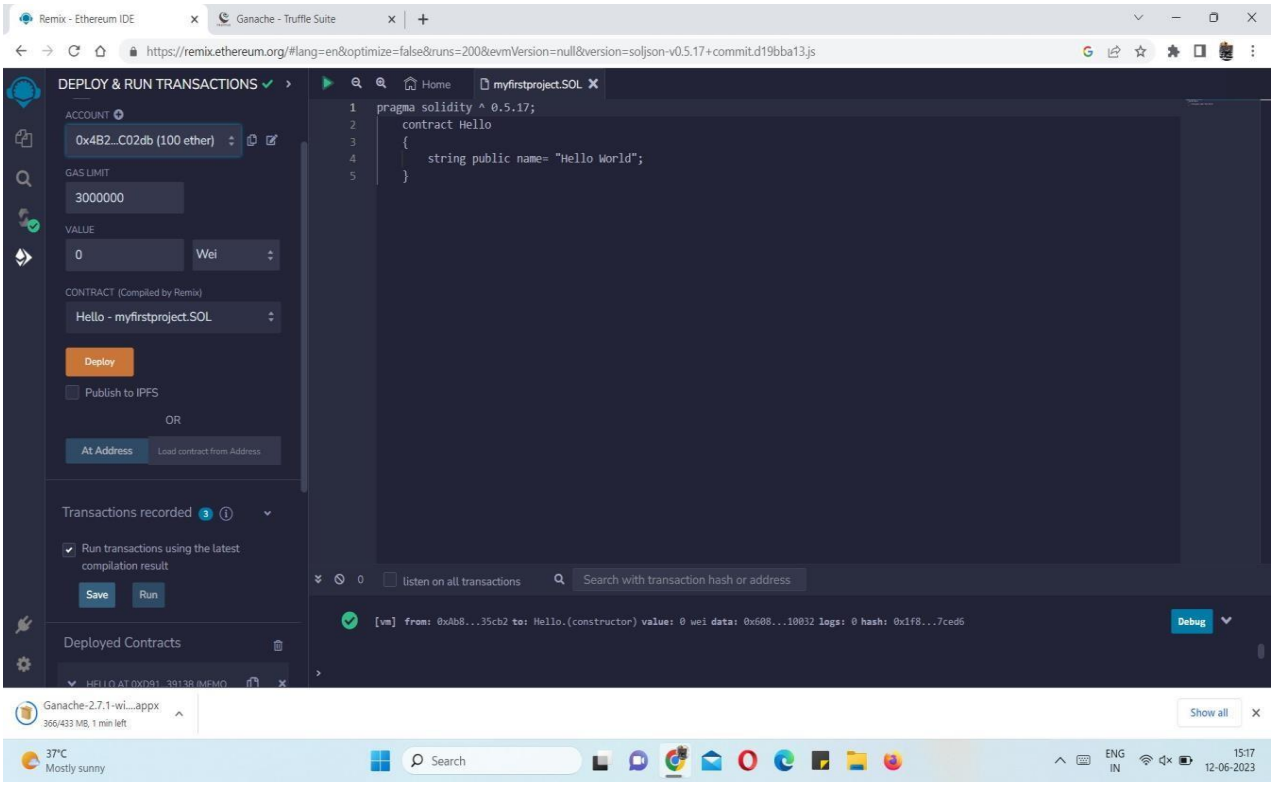
Code:-

```
// SPDX-Licence-Identifier=MIT
pragma solidity ^ 0.5.10;
contract Hello
{
   string public name= "Hello World";
}
```

Output:-




Remix - Ethereum IDE
remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null
DEPLOY & RUN TRANSACTIONS
Remix VM (Merge)
External Http Provider
Note: To use Geth & https://remix.ethereum.org, configure it to allow requests from Remix:(see Geth Docs on rpc server)
geth --http --http.corsdomain https://remix.ethereum.org
To run Remix & a local Geth test node, use this command: (see Geth Docs on Dev mode)
geth --http --http.corsdomain="https://remix.ethereum.org" --http.api web3,eth,debug,personal,net --vmdebug --datadir <path/to/local/folder/for/test/chain> --dev console
WARNING: It is not safe to use the --http.corsdomain flag with a wildcard: --http.corsdomain *
For more info: Remix Docs on External HTTP Provider
External HTTP Provider Endpoint
http://127.0.0.1:7545
OK
Cancel
Deploy
Transactions recorded
Deployed Contracts


Remix - Ethereum IDE
Ganache - Truffle Suite
https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
DEPLOY & RUN TRANSACTIONS
myfirstproject.SOL
pragma solidity ^0.5.17;
contract Hello
{
string public name= "Hello World";
}
0x4B2...C02db (100 ether)
3000000
Hello - myfirstproject.SOL
Deploy
Publish to IPFS
At Address
Transactions recorded
Run transactions using the latest compilation result
Save
Run
Deployed Contracts
Debug
Ganache-2.7.1-wi....appx
Show all
37°C
Mostly sunny
Search