Experiment No.5

Aim: To Run a simple smart contract like HelloWorld in Remix IDE. (Online IDE)

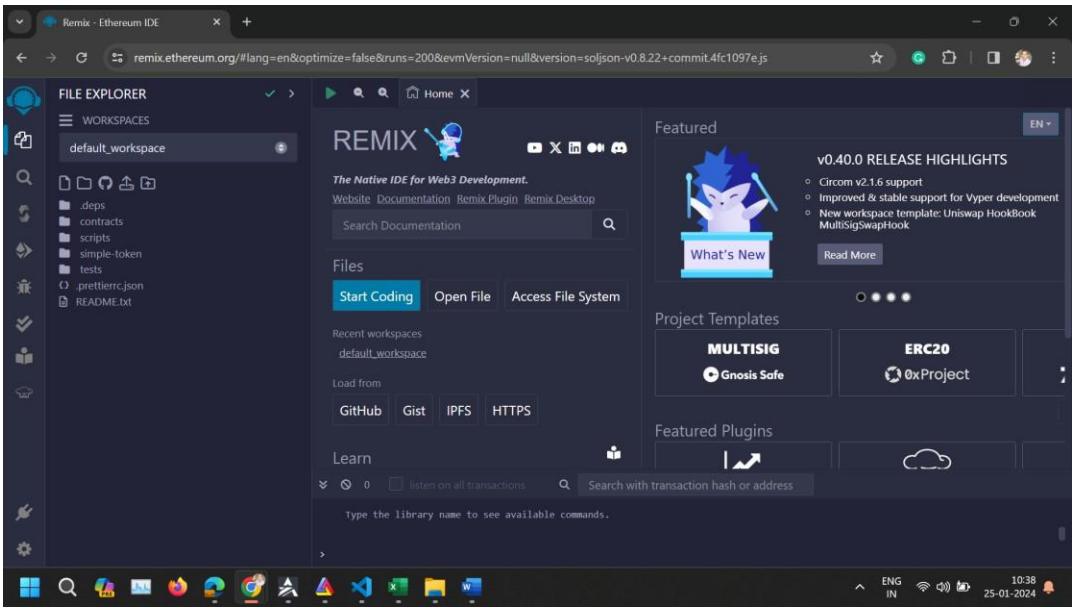
There are several online Integrated Development Environments (IDEs) that can be used to develop, deploy, and test Ethereum smart contracts without the need to set up a local development environment. These online IDEs are useful for experimenting with smart contracts, learning Solidity (the programming language for Ethereum smart contracts), and testing functionalities in a sandbox environment.

1. **Remix (https://remix.ethereum.org):** Remix is a powerful, open-source web and desktop application that is widely used for Ethereum smart contract development. It supports Solidity and Vyper and allows developers to write, compile, debug, and deploy smart contracts directly in the browser. Remix also provides plugins for various tasks, including static analysis, testing, and deployment on various networks, including testnets and the mainnet.
2. **Ethereum Studio (ttps://studio.ethereum.org):** Ethereum Studio is a web-based IDE provided by the Ethereum Foundation. It offers a quick and easy way to get started with Ethereum development, providing templates for various projects, including smart contracts and decentralized applications (DApps). It's integrated with web3 environments and allows for immediate deployment and testing on the Ethereum test networks.
3. **EthFiddle (https://ethfiddle.com):** EthFiddle is a simpler tool compared to Remix but serves as an efficient Solidity code sharing and testing platform. It allows for quick coding and compilation of smart contracts, and although it's more limited in features, it's great for sharing snippets and small contracts.

Running Simple Contract in Remix

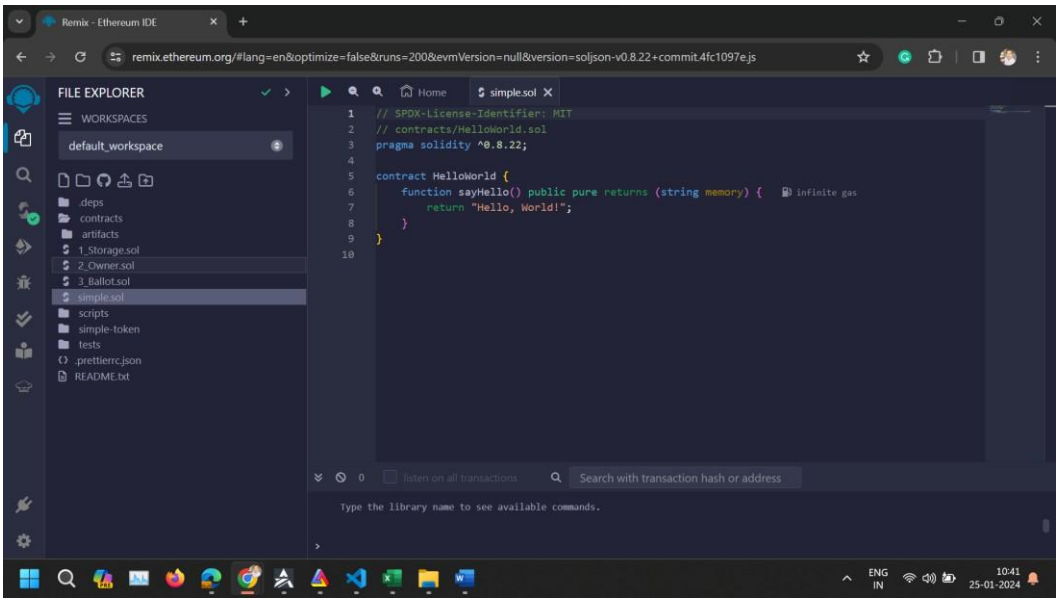
Running a simple smart contract like HelloWorld in Remix IDE is a great way to quickly test and deploy smart contracts on the Ethereum network. Here are the steps to do so.

1. **Access Remix IDE:** Open your web browser and go to Remix IDE. Remix is a powerful, open-source tool that helps you write Solidity contracts straight from the browser.



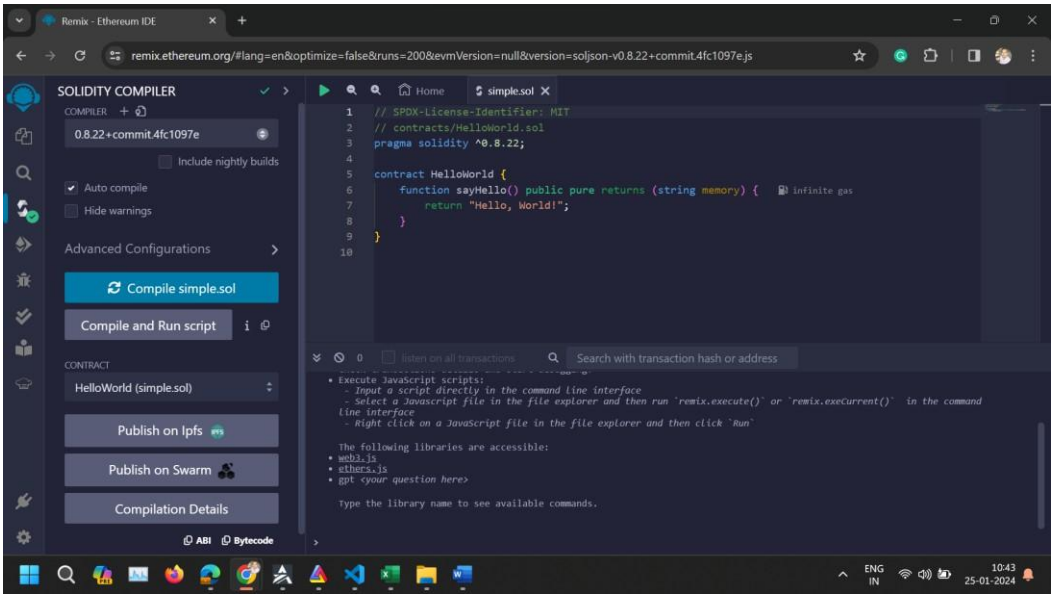
2. Create the Contract File
    - Once in Remix, click on the "File Explorers" icon on the left sidebar.
    - Click the "Create New File" icon at the top of the File Explorers pane. Name your file HelloWorld.sol.
    - Copy and paste your smart contract code into the new file you've just created.

```
// SPDX-License-Identifier: MIT

    pragma solidity ^0.8.22;

    contract HelloWorld {

        function sayHello() public pure returns (string memory) {

                return "Hello, World!";

                }

        }
```



3. Compile the Contract:

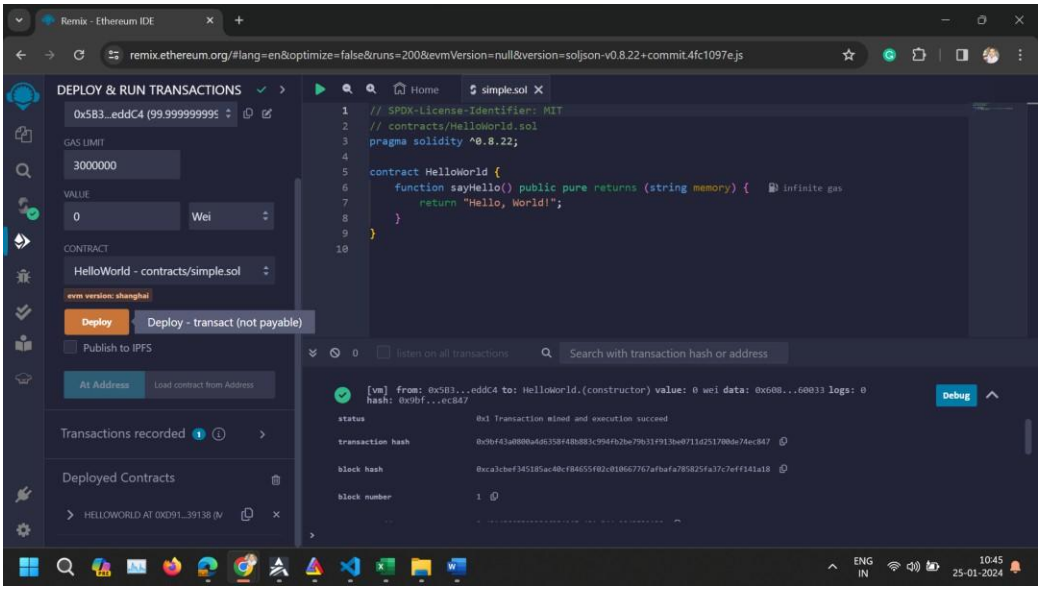
- Click on the "Solidity compiler" icon on the left sidebar.
- Select the correct compiler version that matches the version specified in your contract (0.8.22 or newer) from the "Compiler" dropdown menu.
- Click the "Compile HelloWorld.sol" button to compile your contract.



4. Deploy the Contract:

- After compiling, click on the "Deploy & run transactions" icon on the left sidebar.
- From the "Environment" dropdown, select an environment suitable for your needs. For testing purposes, "JavaScript VM" is a good choice as it simulates the Ethereum blockchain. For deployment to an actual network, you would select "Injected Web3" or "Web3 Provider" and connect to a wallet like MetaMask.
- Under the "Contract" dropdown, make sure "HelloWorld" is selected.

- Click the "Deploy" button. Your contract is now deployed to the selected environment, and you should see it under the "Deployed Contracts" section.



5. Interact with the Contract:

- In the "Deployed Contracts" section, find your newly deployed HelloWorld contract. It should be expandable, revealing a button that represents your sayHello function.

Click the sayHello function button. This executes the function, and you should see the output "Hello, World!" in the Remix console.