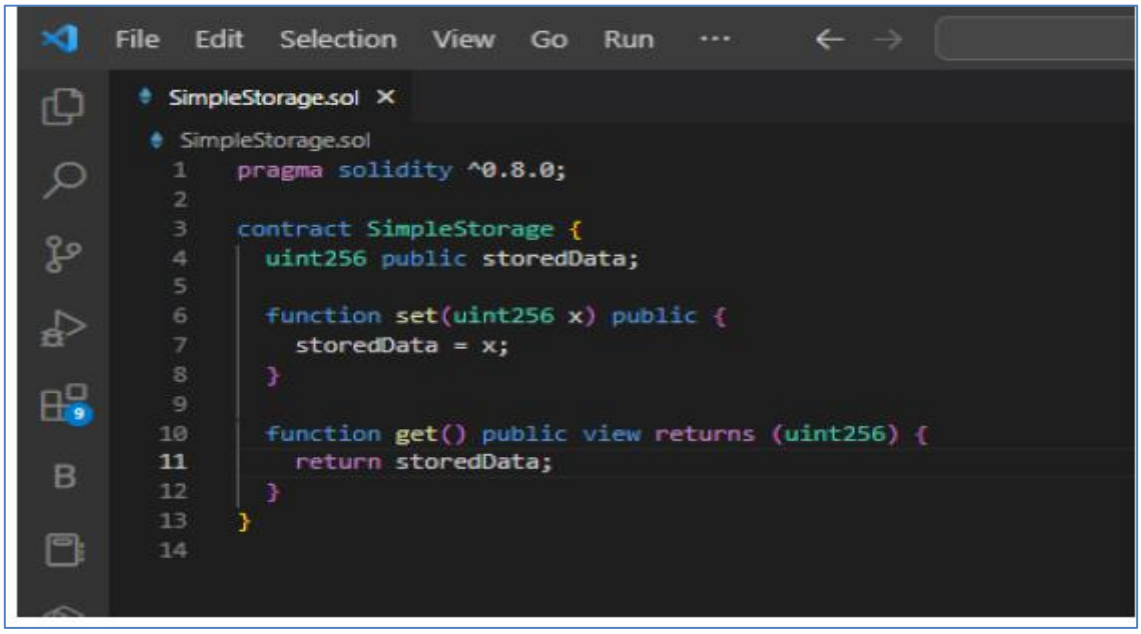
DOP: / /2023 DOS: / / 2023

Experiment No: 01

Aim: - To develop and deploy smart contracts on local Blockchain using solidity programming language.

Theory:

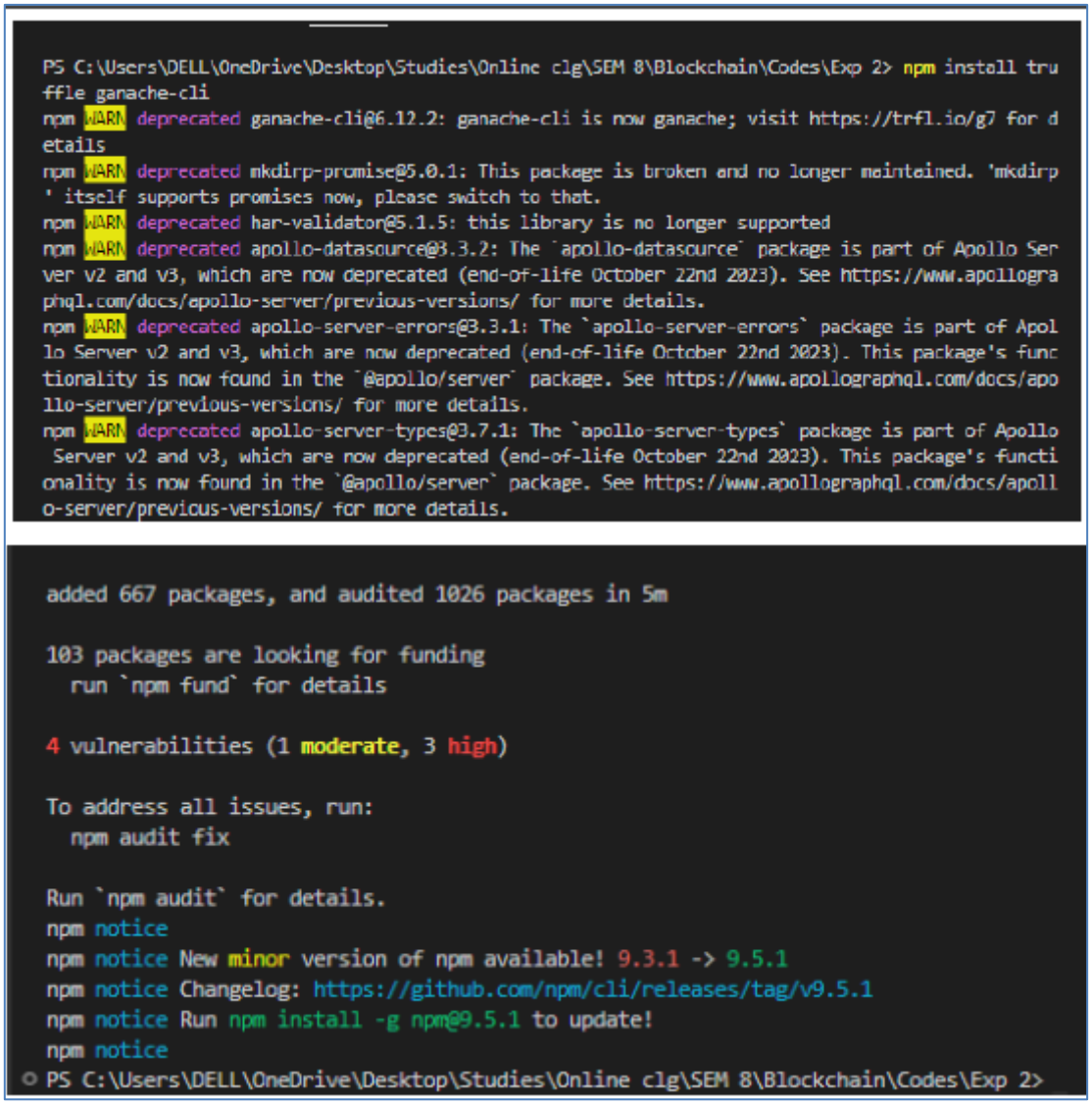
1. Create a new directory for your project and navigate to it in your terminal.
2. Create a new Solidity file called SimpleStorage.sol with the following code:



This contract defines a simple storage system that stores and retrieves a single integer value.

3 Install the required dependencies by running the following command in your terminal:

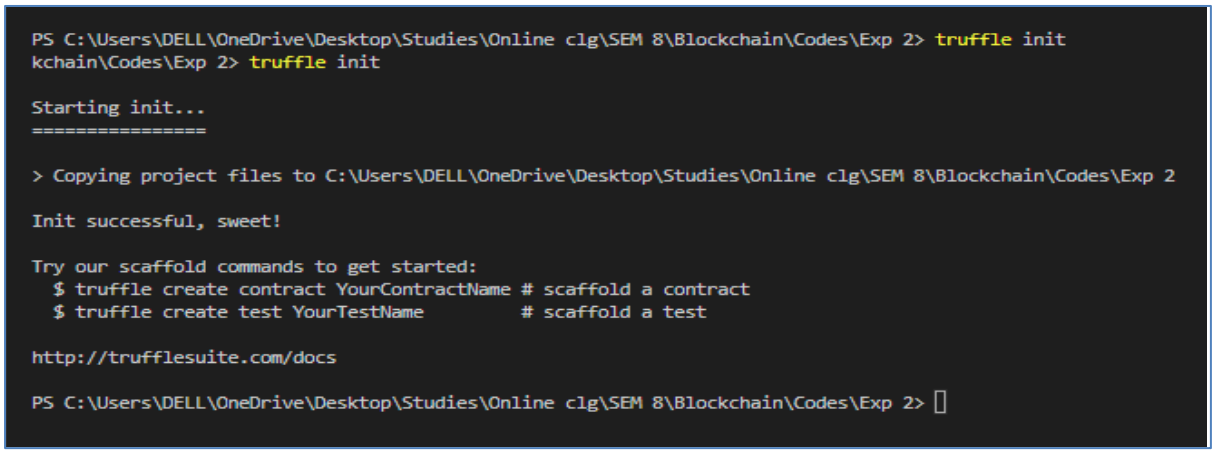
npm install truffle ganache-cli



This will install Truffle and Ganache CLI, which we'll use to deploy and test our smart contract.

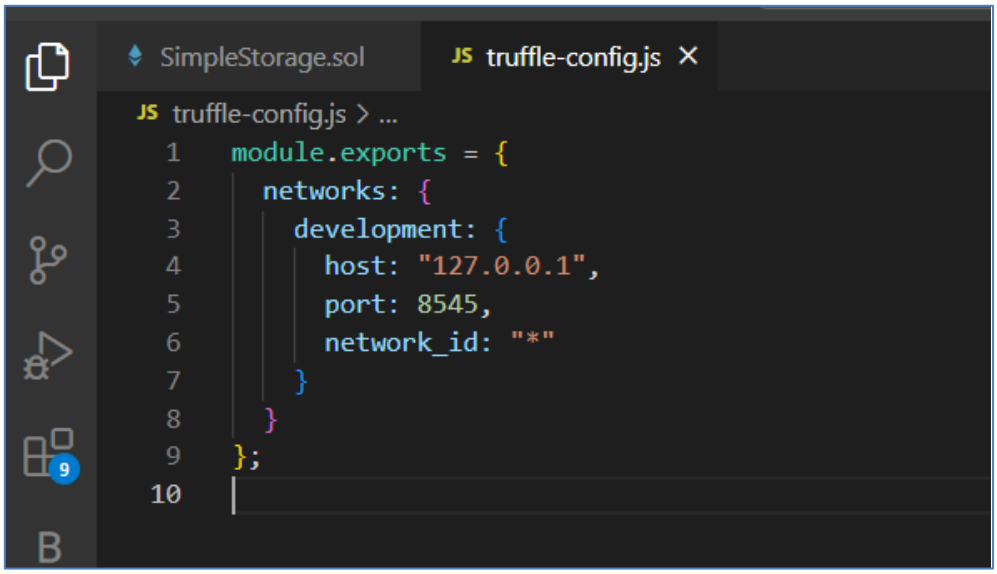
4. Create a new Truffle project by running the following command:

truffle init



This will create a basic project structure with some boilerplate code.

5. Configure your blockchain network by editing the truffle-config.js file in the root directory of your project. Add the following code to configure the Ganache CLI network:



This code tells Truffle to use the Ganache CLI network for development, which will run on http://127.0.0.1:8545.

6. Compile your smart contract by running the following command:

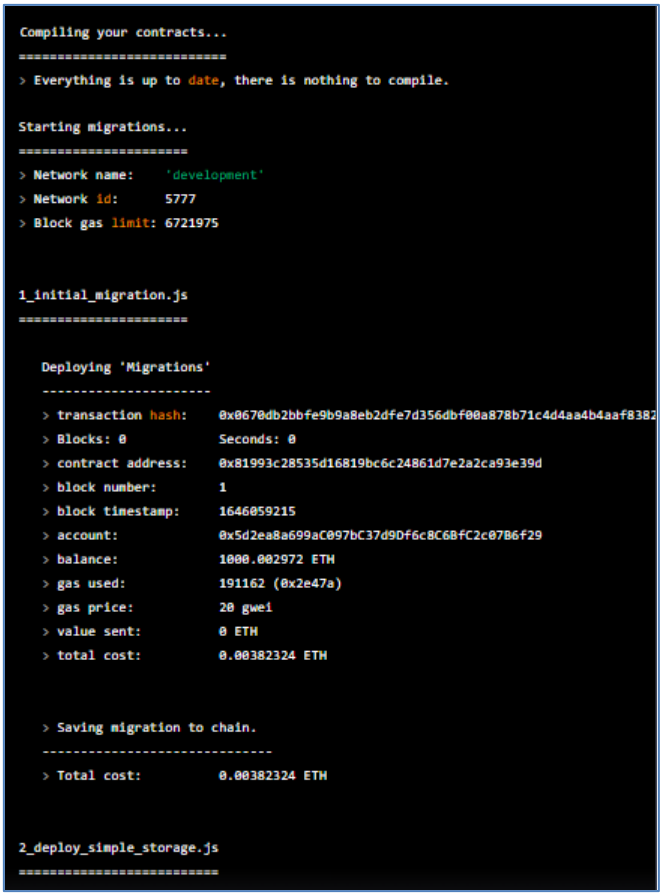
truffle compile

```
Compiling your contracts...
===========================
> Compiling ./contracts/SimpleStorage.sol
> Compiling @openzeppelin/contracts/access/Ownable.sol
> Artifacts written to /path/to/myproject/build/contracts
> Compiled successfully using:
   - solc: 0.8.9+commit.e5eed63a.Emscripten.clang
```

This will compile your Solidity code and generate the corresponding bytecode and ABI files in the build directory.

7. Migrate your smart contract by running the following command:

truffle migrate



This will deploy your smart contract to the Ganache CLI network.ract with your smart contract by running the following commands:

truffle console

8. This will open the Truffle console, which provides a JavaScript environment for interacting with your smart contract.

let instance = await SimpleStorage.deployed()

```
PS C:\Users\DELL\OneDrive\Desktop\Studies\Online clg\SEM 8\Blockchain\Codes\Exp
 2> truffle console
truffle(development)> let instance = await SimpleStorage.deployed()
```

```
undefined
```

9. This will create an instance of your deployed smart contract.

await instance.set(42)

```
await simpleStorage.set(42)
```

This will set the value of storedData to 42.

let value = await instance.get()

This will retrieve the value of storedData.

11. Get the value from your smart contract by running the following command:

let result = await simpleStorage.get()

```
let result = await simpleStorage.get()
```

```
BN { negative: 0, words: [ 42 ], length: 1, red: null }
```

Here, `result` is an instance of the BN (Big Number) object from the web3.js library. In this case, the value `42` has been successfully set in the `SimpleStorage` smart contract and retrieved using the `get()` function.

Conclusion: You have developed and deployed a simple smart contract on a local blockchain using the Solidity programming language.