

## ASSIGNMENT – 14

### INPUT –

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

pragma solidity ^0.8.27;

contract StudentData {

    // Structure to store student information

    struct Student {

        uint id;

        string name;

        uint age;

    }

    // Array to store the list of students

    Student[] public students;

    // Event to log the addition of a new student

    event StudentAdded(uint id, string name, uint age);

    // Constructor: Now payable to accept Ether

    constructor() payable {

        // Constructor can accept Ether

    }

    // Function to add a student to the array

    function addStudent(uint _id, string memory _name, uint _age) public {

        students.push(Student(_id, _name, _age));

        emit StudentAdded(_id, _name, _age); // Emit event when student is added

    }

    // Function to retrieve the number of students

    function getStudentCount() public view returns (uint) {

        return students.length;

    }

    // Receive function to accept plain Ether transactions

    receive() external payable {

        // Receive function handles Ether sent directly

    }

    // Fallback function to handle unexpected calls or Ether with data

    fallback() external payable {
```

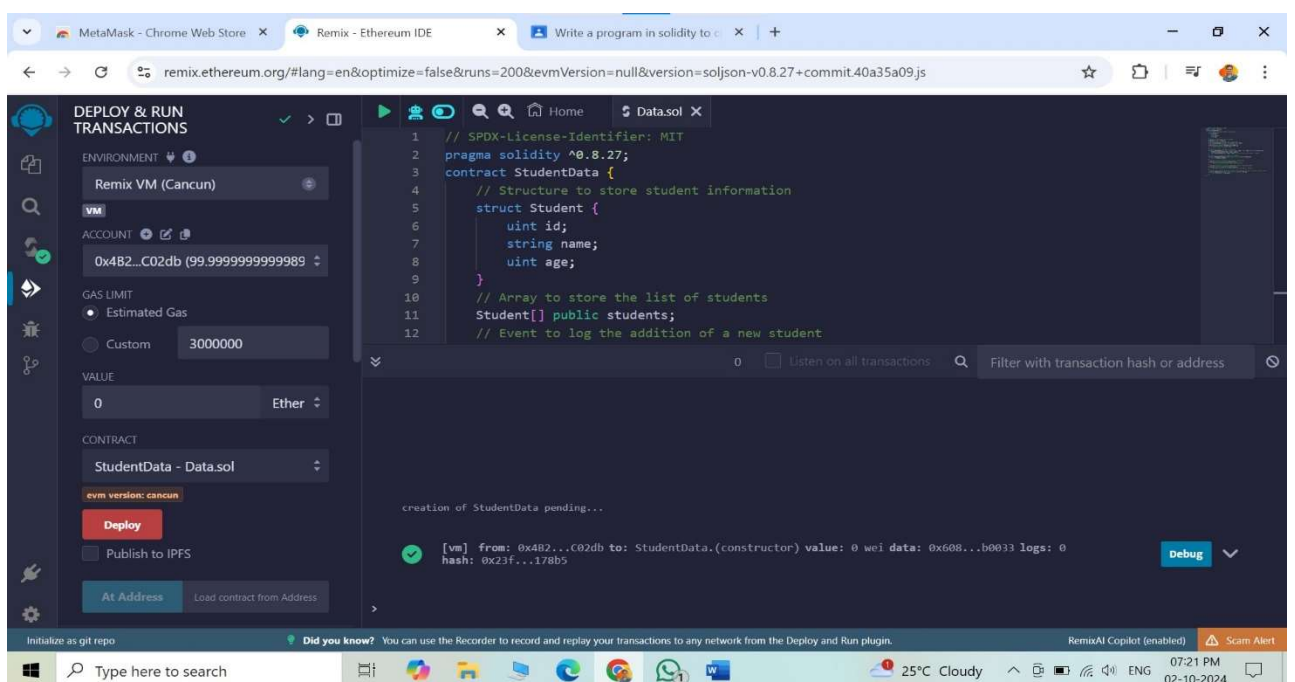
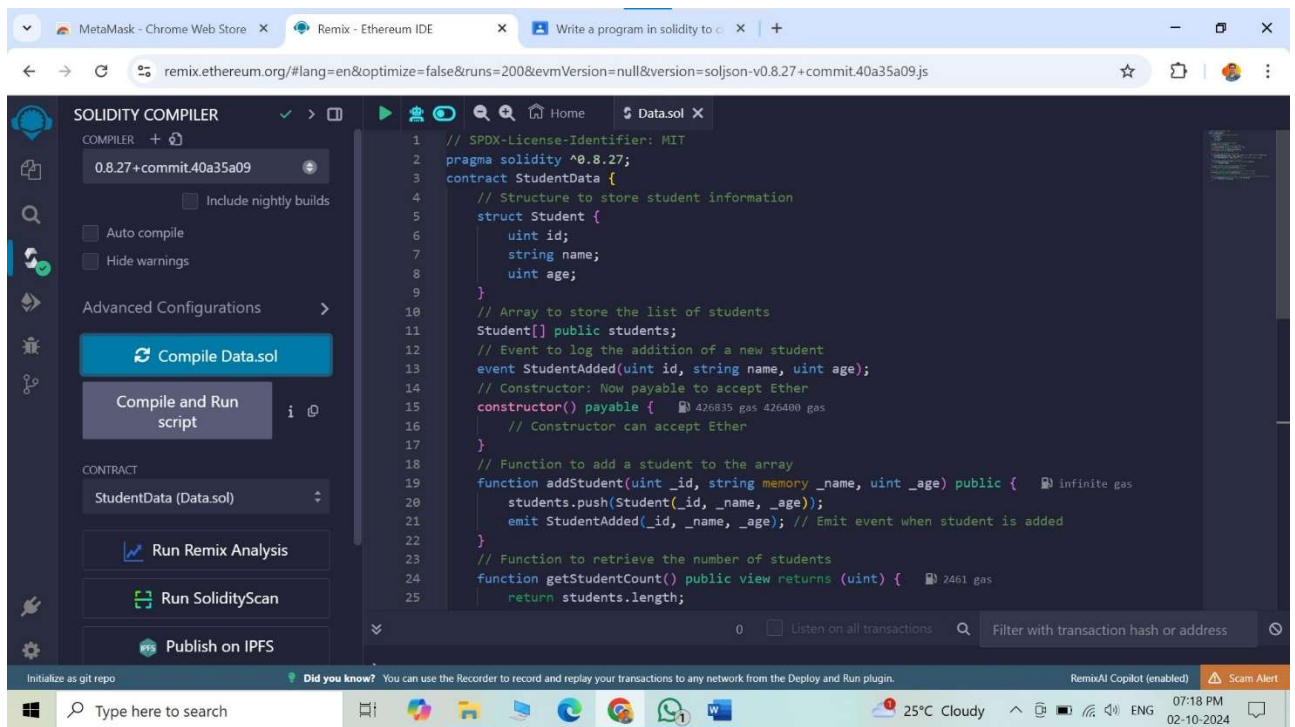
```

    // Fallback function to receive Ether when calldata is present
}

// Function to retrieve contract balance if fallback or receive function receives Ether
function getBalance() public view returns (uint) {
    return address(this).balance;
}
}

```

OUTPUT –





MetaMask - Chrome Web Store

Remix - Ethereum IDE

Write a program in solidity to c

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.27+commit.40a35a09.js

DEPLOY & RUN TRANSACTIONS

Balance: 0 ETH

addStudent

\_id: 2

\_name: Om

\_age: 21

CalldataParameterstransact

getBalance

0: uint256: 0

getStudentCo...

0: uint256: 3

students

2

0: uint256: id 2

1: string: name Om

2: uint256: age 21

Low level interactions

CALLDATA

1 // SPDX-License-Identifier: MIT

2 pragma solidity ^0.8.27;

3 contract StudentData {

4 // Structure to store student information

5 struct Student {

0 Listen on all transactions

Filter with transaction hash or address

call to StudentData.getStudentCount

CALL [call] from: 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db to: StudentData.getStudentCount()

data: 0x41e...0c407

Debug

call to StudentData.students

CALL [call] from: 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db to: StudentData.students(uint256)

data: 0x06e...00003

Debug

call to StudentData.students errored: Error occurred: revert.

revert

The transaction has been reverted to the initial state.

Note: The called function should be payable if you send value and the value you send should be less than your current balance. You may want to cautiously increase the gas limit if the transaction went out of gas.

call to StudentData.students

CALL [call] from: 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db to: StudentData.students(uint256)

data: 0x06e...00002

Debug

Initialize as git repo

Did you know? You can use the Recorder to record and replay your transactions to any network from the Deploy and Run plugin.

RemixAI Copilot (enabled)

Scam Alert

Type here to search

25°C Heavy t-storms

ENG

07:27 PM

02-10-2024