Pune Institute of Computer Technology

Department of Computer Engineering

(2022- 2023)

“Student Management”

Submitted to the

Savitribai Phule Pune University

In partial fulfilment for the award of the Degree of

Bachelor of Engineering

In

Computer Engineering

By

1)	**Anuj Mahendra Mutha**	**41443**
2)	**Nidhi Patil**	**41447**
3)	**Amit Purohit**	**41450**

Problem Statement:

Mini-Project- Develop a Blockchain based application- Student Attendance Management System using Smart Contract and deploy it using Ethereum

Objective:
Understand and explore the working of Blockchain technology and its applications.

Course Outcome:
CO6: Interpret the basic concepts in Blockchain technology and its applications.

Description:
Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met.

A smart contract is just a digital contract with the security coding of the blockchain.

- It has details and permissions written in code that require an exact sequence of events to take place to trigger the agreement of the terms mentioned in the smart contract.
- It can also include the time constraints that can introduce deadlines in the contract.
- Every smart contract has its address in the blockchain. The contract can be interacted with by using its address presuming the contract has been broadcasted on the network.

The idea behind smart contracts is pretty simple. They are executed on a basis of simple logic, IF-THEN for example:

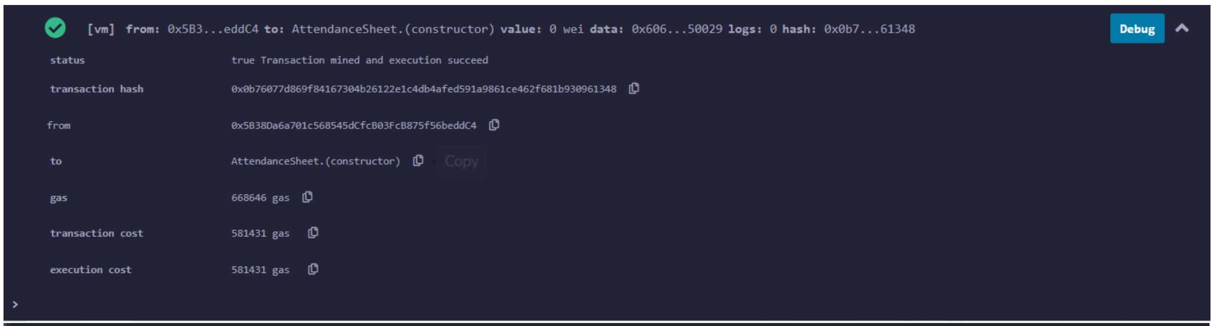
- **IF** you send object A, **THEN** the sum (of money, in cryptocurrency) will be transferred to you.
- **IF** you transfer a certain amount of digital assets (cryptocurrency, for example, ether, bitcoin), **THEN** the A object will be transferred to you.
- **IF** I finish the work, **THEN** the digital assets mentioned in the contract will be transferred to me.

Code:

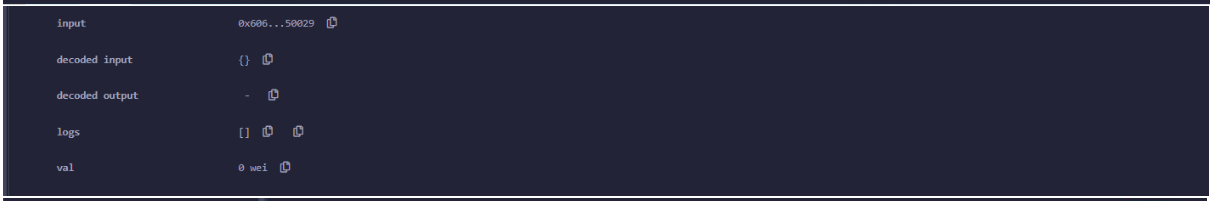
```
pragma solidity ^0.4.18;

contract Owned {
   address owner;

   function Owned() public {
       owner = msg.sender;
    }

   modifier onlyOwner {
       require(msg.sender == owner);
       _;
   }
```

```
}

contract AttendanceSheet is Owned {

    struct Student {
        uint age;
        string fName;
        string lName;
        uint attendanceValue;
    }

    mapping (uint => Student) studentList;
    uint[] public studIdList;

    event studentCreationEvent(
       string fName,
       string lName,
       uint age
    );

    function createStudent(uint _studId, uint _age, string _fName, string _lName) onlyOwner
public {
        var student = studentList[_studId];

        student.age = _age;
        student.fName = _fName;
        student.lName = _lName;
        student.attendanceValue = 0;
        studIdList.push(_studId) -1;
        studentCreationEvent(_fName, _lName, _age);
    }

    function incrementAttendance(uint _studId) onlyOwner public {
        studentList[_studId].attendanceValue = studentList[_studId].attendanceValue+1;
    }

    function getStudents() view public returns(uint[]) {
        return studIdList;
    }

    function getParticularStudent(uint _studId) public view returns (string, string, uint, uint) {
        return (studentList[_studId].fName, studentList[_studId].lName,
studentList[_studId].age, studentList[_studId].attendanceValue);
    }
    function countStudents() view public returns (uint) {
        return studIdList.length;
    }

}
```

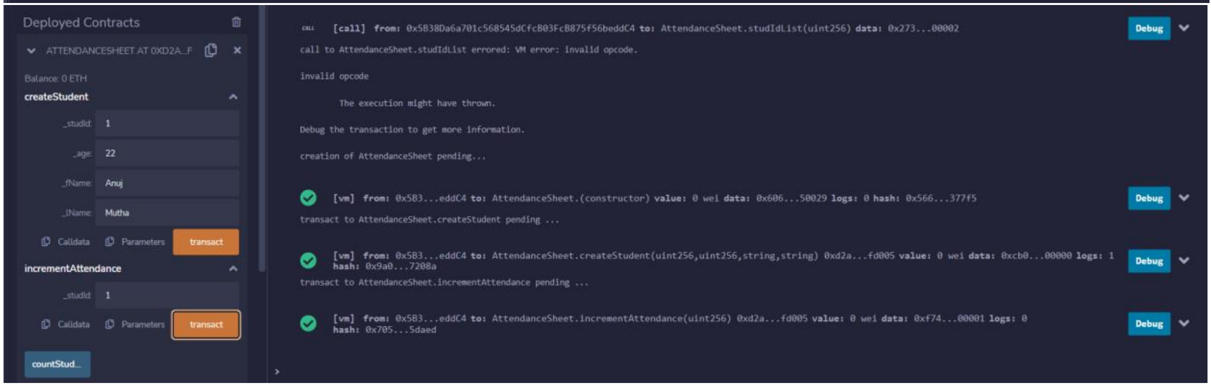
Output:



```
[vm] from: 0x5B3...eddC4 to: AttendanceSheet.(constructor) value: 0 wei data: 0x606...50029 logs: 0 hash: 0x0b7...61348
status                 true Transaction mined and execution succeed
transaction hash       0x0b76077d869f84167304b26122e1c4db4afed591a9861ce462f681b930961348
from                   0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                     AttendanceSheet.(constructor)
gas                    668646 gas
transaction cost       581431 gas
execution cost         581431 gas
```



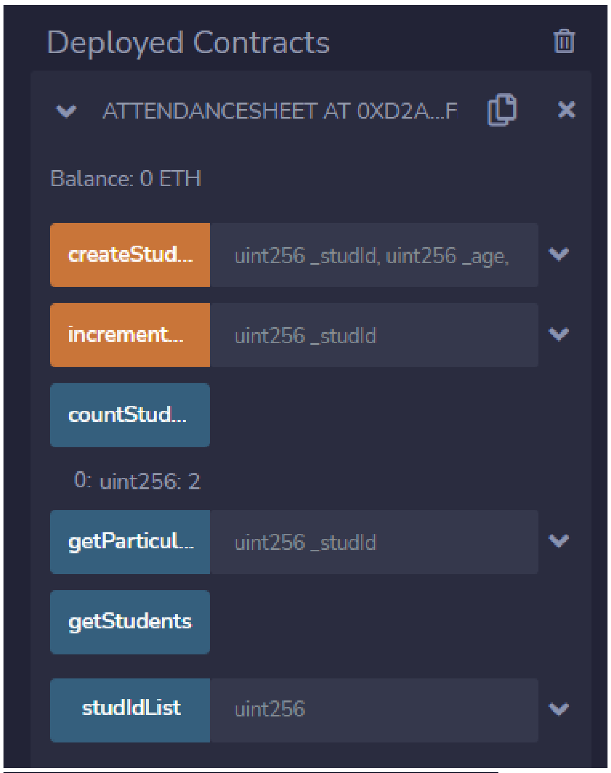
```
input                  0x606...50029
decoded input          {}
decoded output         -
logs                   []
val                    0 wei
```



createStudent
_studId: 1
_age: 22
_fName: Anuj
_lName: Mutha

incrementAttendance
_studId: 1

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: AttendanceSheet.studIdList(uint256) data: 0x273...00002
call to AttendanceSheet.studIdList errored: VM error: invalid opcode.

invalid opcode

        The execution might have thrown.

Debug the transaction to get more information.

creation of AttendanceSheet pending...

[vm] from: 0x5B3...eddC4 to: AttendanceSheet.(constructor) value: 0 wei data: 0x606...50029 logs: 0 hash: 0x566...377f5
transact to AttendanceSheet.createStudent pending ...

[vm] from: 0x5B3...eddC4 to: AttendanceSheet.createStudent(uint256,uint256,string,string) 0xd2a...fd005 value: 0 wei data: 0xcb0...00000 logs: 1
hash: 0x9a0...7208a
transact to AttendanceSheet.incrementAttendance pending ...

[vm] from: 0x5B3...eddC4 to: AttendanceSheet.incrementAttendance(uint256) 0xd2a...fd005 value: 0 wei data: 0xf74...00001 logs: 0
hash: 0x705...5daed
```



Balance: 0 ETH

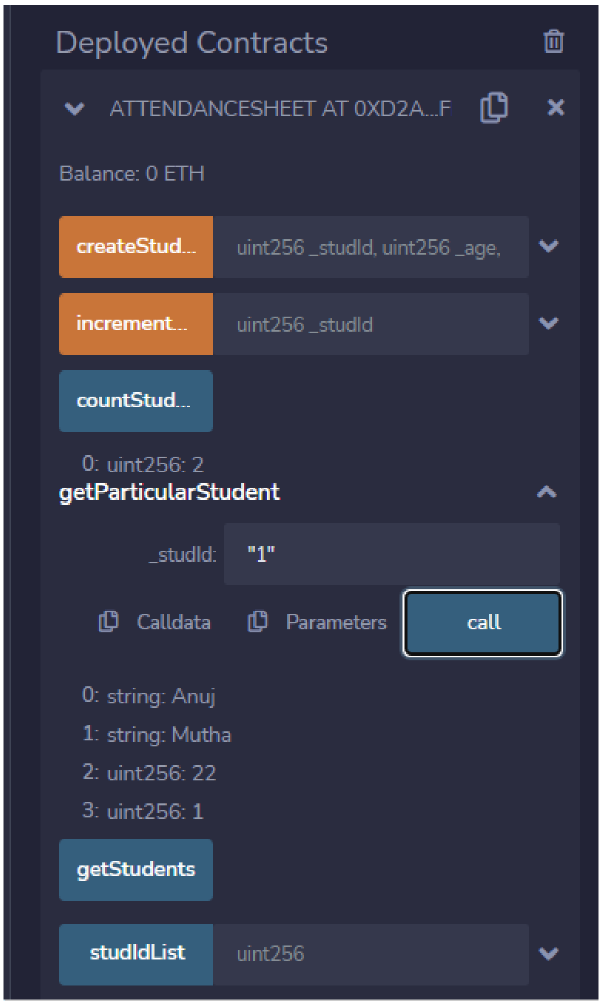
createStud... uint256 _studId, uint256 _age,

increment... uint256 _studId

0: uint256: 2

getParticul... uint256 _studId

studIdList uint256



Balance: 0 ETH

createStud... uint256 _studId, uint256 _age,

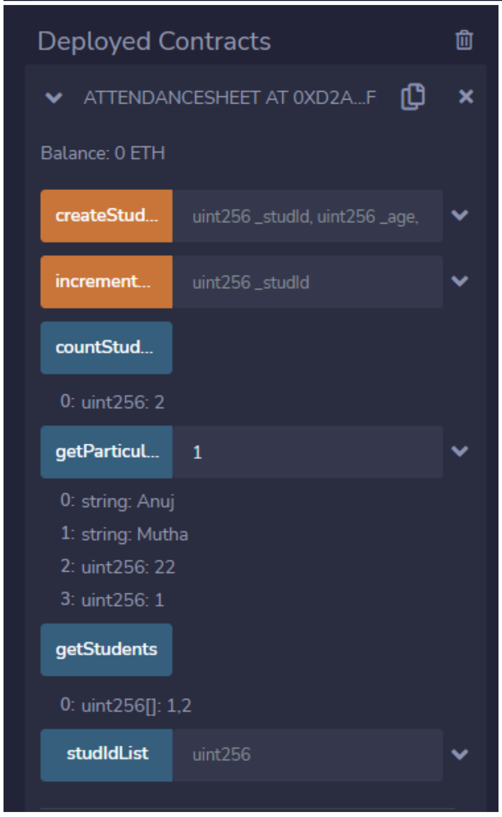
increment... uint256 _studId

0: uint256: 2

_studId: "1"

```
0: string: Anuj
1: string: Mutha
2: uint256: 22
3: uint256: 1
```

studIdList uint256



Balance: 0 ETH

createStud... uint256 _studId, uint256 _age,

increment... uint256 _studId

0: uint256: 2

getParticul... 1

```
0: string: Anuj
1: string: Mutha
2: uint256: 22
3: uint256: 1
```

```
0: uint256[]: 1,2
```

studIdList uint256

Conclusion:

I studied about smart contract and how to write and execute it using Remix IDE.