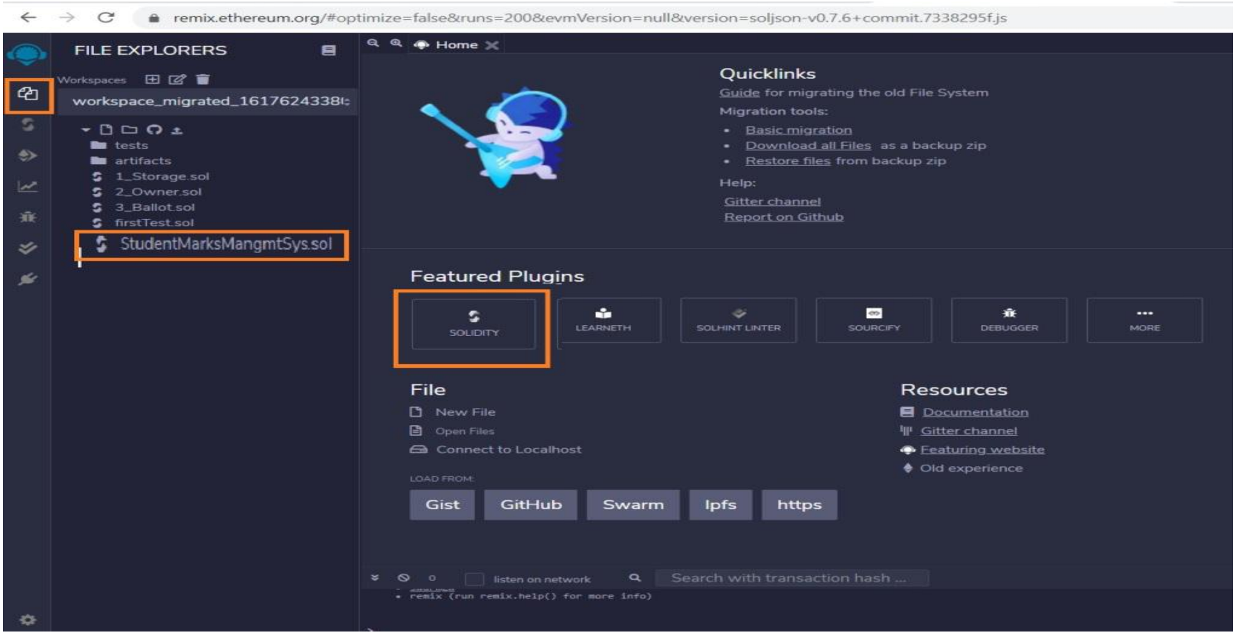
- **Implementation**

Step 1: Open Remix IDE.

Step 2: Click on File Explorers and select Solidity in the environment and create a new file StudentMarksMangmtSys.sol by clicking on New File section.

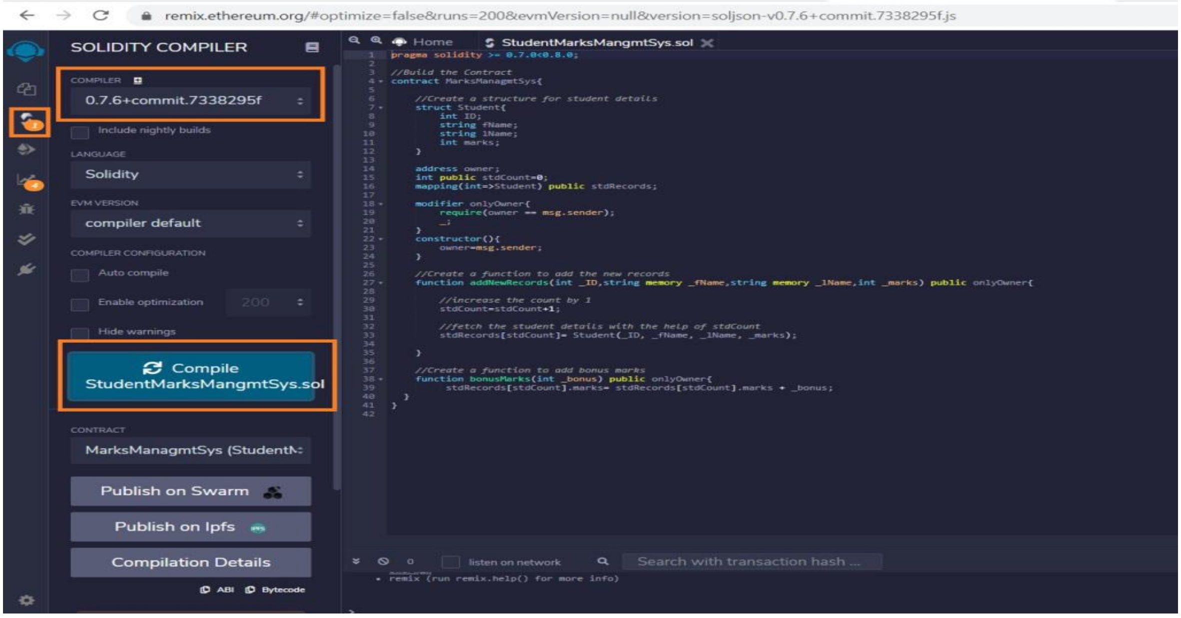


Step 3: Build a smart contract that contains all the details of the student with the help of Remix IDE by clicking on the file name.
Solidity

```
// Solidity program to implement
// the above approach
pragma solidity >= 0.7.0<0.8.0;

// Build the Contract
contract MarksManagmtSys
{
   // Create a structure for
   // student details
   struct Student
   {
      int ID;
```

```
        string fName;
        string lName;
        int marks;
    }

    address owner;
    int public stdCount = 0;
    mapping(int => Student) public stdRecords;

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

    // Create a function to add
    // the new records
    function addNewRecords(int _ID,
                           string memory _fName,
                           string memory _lName,
                           int _marks) public onlyOwner
    {
        // Increase the count by 1
        stdCount = stdCount + 1;

        // Fetch the student details
        // with the help of stdCount
        stdRecords[stdCount] = Student(_ID, _fName,
                                       _lName, _marks);
    }

    // Create a function to add bonus marks
    function bonusMarks(int _bonus) public onlyOwner
    {
         stdRecords[stdCount].marks =
                    stdRecords[stdCount].marks + _bonus;
    }
}
```

Step 4: After building the contract compile it. Select the compiler version before clicking on Compile button.



remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.7.6+commit.7338295f.js

SOLIDITY COMPILER

COMPILER
0.7.6+commit.7338295f

□ Include nightly builds

LANGUAGE
Solidity

EVM VERSION
compiler default

COMPILER CONFIGURATION
□ Auto compile
□ Enable optimization 200
□ Hide warnings

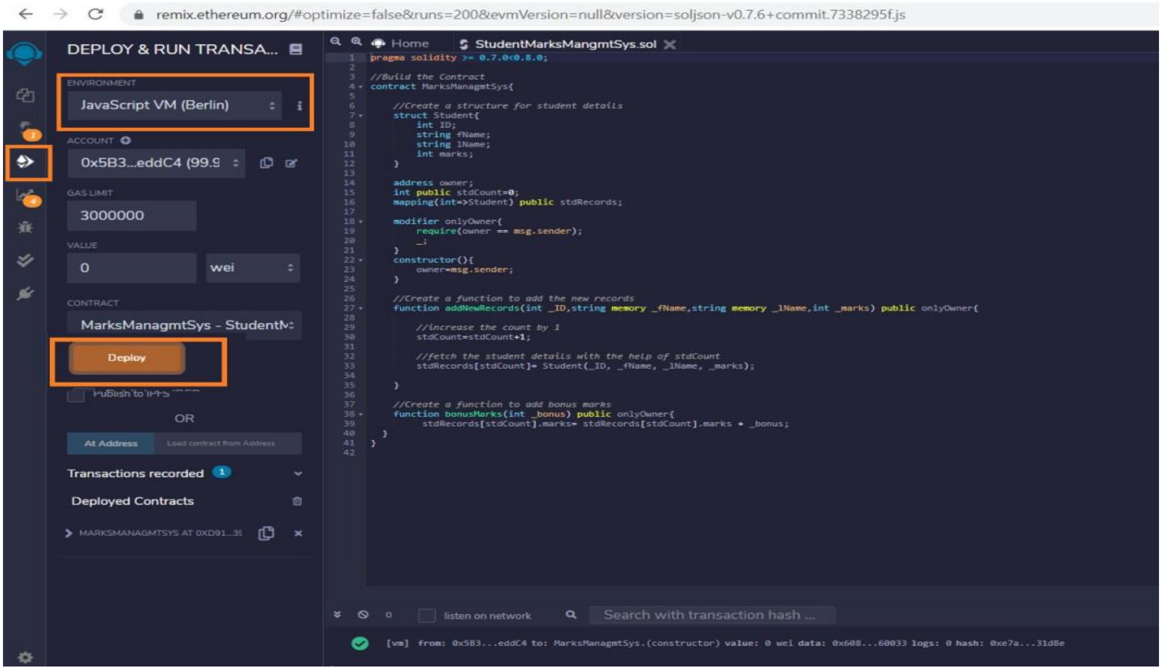
Compile StudentMarksMangmtSys.sol

CONTRACT
MarksManagmtSys (StudentM

```
pragma solidity >= 0.7.0<0.8.0;

//Build the Contract
contract MarksManagmtSys{

    //Create a structure for student details
    struct Student{
        int ID;
        string fName;
        string lName;
        int marks;
    }

    address owner;
    int public stdCount=0;
    mapping(int=>Student) public stdRecords;

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

    //Create a function to add the new records
    function addNewRecords(int _ID,string memory _fName,string memory _lName,int _marks) public onlyOwner{

        //increase the count by 1
        stdCount=stdCount+1;

        //fetch the student details with the help of stdCount
        stdRecords[stdCount]= Student(_ID, _fName, _lName, _marks);

    }

    //Create a function to add bonus marks
    function bonusMarks(int _bonus) public onlyOwner{
         stdRecords[stdCount].marks= stdRecords[stdCount].marks + _bonus;
    }
}
```

```
- remix (run remix.help() for more info)
```

Step 5: After successful compilation, to deploy the contract, select the Environment JavaScript VM (Berlin) before clicking on the Deploy button.



remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.7.6+commit.7338295f.js

DEPLOY & RUN TRANSA...

ENVIRONMENT
JavaScript VM (Berlin)

ACCOUNT
0x5B3...eddC4 (99.9

GAS LIMIT
3000000

VALUE
0 wei

CONTRACT
MarksManagmtSys - StudentM

Deploy

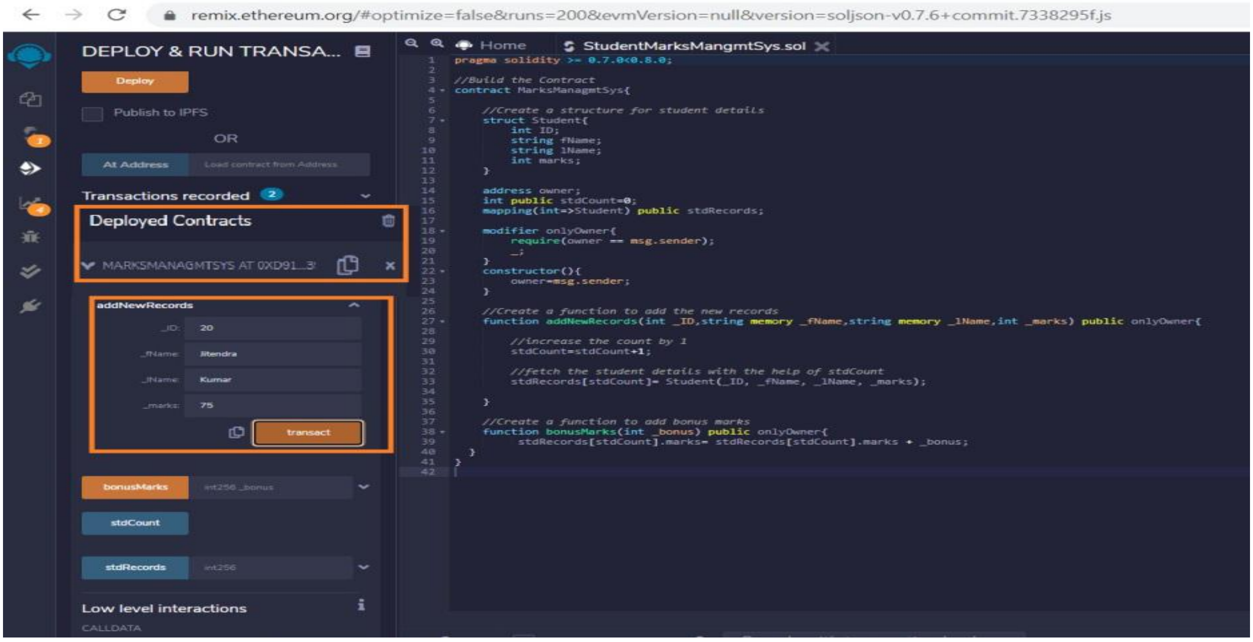
□ Publish to IPFS

OR
At Address
Load contract from Address
Transactions recorded 1
Deployed Contracts
MARKSMANAGMTSYS AT 0XD91...35
Home
StudentMarksMangmtSys.sol

```
pragma solidity >= 0.7.0<0.8.0;

//Build the Contract
contract MarksManagmtSys{

    //Create a structure for student details
    struct Student{
        int ID;
        string fName;
        string lName;
        int marks;
    }

    address owner;
    int public stdCount=0;
    mapping(int=>Student) public stdRecords;

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

    //Create a function to add the new records
    function addNewRecords(int _ID,string memory _fName,string memory _lName,int _marks) public onlyOwner{

        //increase the count by 1
        stdCount=stdCount+1;

        //fetch the student details with the help of stdCount
        stdRecords[stdCount]= Student(_ID, _fName, _lName, _marks);

    }

    //Create a function to add bonus marks
    function bonusMarks(int _bonus) public onlyOwner{
         stdRecords[stdCount].marks= stdRecords[stdCount].marks + _bonus;
    }
}
```

```
[vm] from: 0x5B3...eddC4 to: MarksManagmtSys.(constructor) value: 0 wei data: 0x608...60033 logs: 0 hash: 0xe7a...31d8e
```

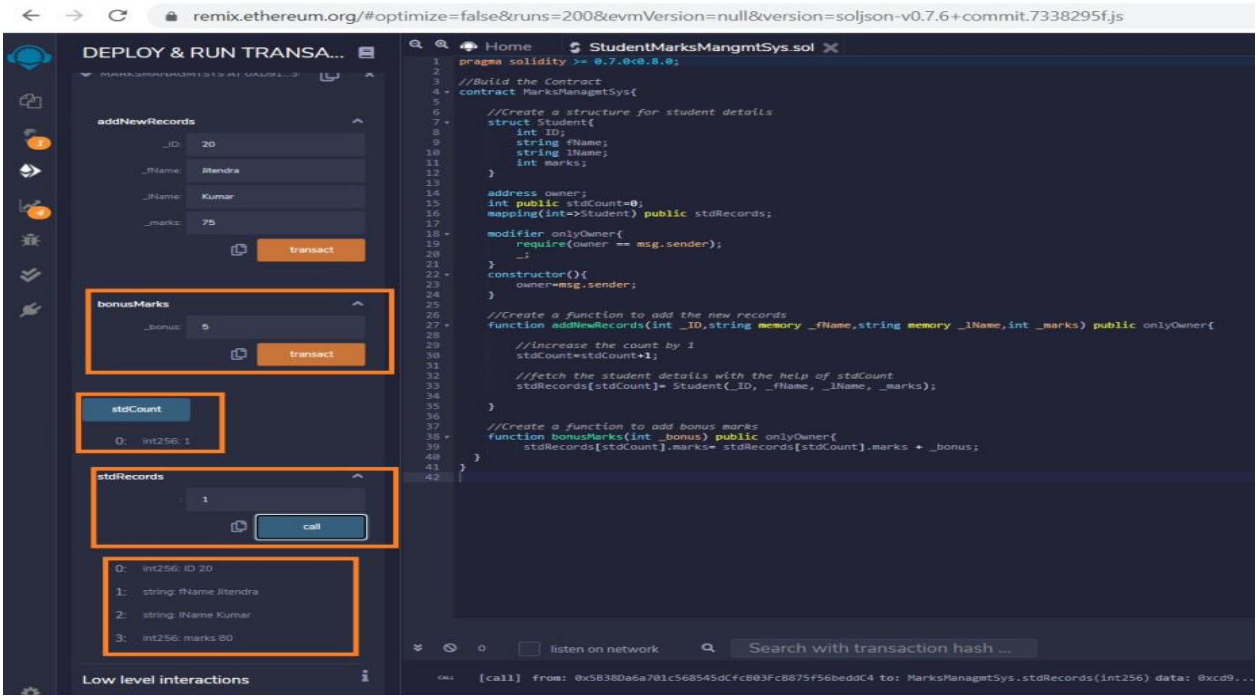
Step 6: If the contract is successfully deployed then deployed contract is obtained. Open the deployed contract and add the student details and transact it.



_ID: 20
_fName: Jitendra
_lName: Kumar
_marks: 75

```
pragma solidity >= 0.7.0<0.8.0;

//Build the Contract
contract MarksManagmtSys{

    //Create a structure for student details
    struct Student{
        int ID;
        string fName;
        string lName;
        int marks;
    }

    address owner;
    int public stdCount=0;
    mapping(int=>Student) public stdRecords;

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

    //Create a function to add the new records
    function addNewRecords(int _ID,string memory _fName,string memory _lName,int _marks) public onlyOwner{

        //increase the count by 1
        stdCount=stdCount+1;

        //fetch the student details with the help of stdCount
        stdRecords[stdCount]= Student(_ID, _fName, _lName, _marks);

    }

    //Create a function to add bonus marks
    function bonusMarks(int _bonus) public onlyOwner{
         stdRecords[stdCount].marks= stdRecords[stdCount].marks + _bonus;
   }
}
```

Step 7: Add the bonus marks if you want to give them to the student and transact it after that click on the stdCount. One can see the student details after calling the stdRecords by entering the stdCount.


remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.7.6+commit.7338295f.js
DEPLOY & RUN TRANSA...
addNewRecords
transact
bonusMarks
transact
stdCount
stdRecords
call
Low level interactions
Home
StudentMarksMangmtSys.sol
listen on network
Search with transaction hash ...

_ID: 20
_fName: Jitendra
_lName: Kumar
_marks: 75

_bonus: 5

0: int256: 1

1

```
0: int256: ID 20
1: string: fName Jitendra
2: string: lName Kumar
3: int256: marks 80
```

```
pragma solidity >= 0.7.0<0.8.0;

//Build the Contract
contract MarksManagmtSys{

    //Create a structure for student details
    struct Student{
        int ID;
        string fName;
        string lName;
        int marks;
    }

    address owner;
    int public stdCount=0;
    mapping(int=>Student) public stdRecords;

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

    //Create a function to add the new records
    function addNewRecords(int _ID,string memory _fName,string memory _lName,int _marks) public onlyOwner{

        //increase the count by 1
        stdCount=stdCount+1;

        //fetch the student details with the help of stdCount
        stdRecords[stdCount]= Student(_ID, _fName, _lName, _marks);

    }

    //Create a function to add bonus marks
    function bonusMarks(int _bonus) public onlyOwner{
         stdRecords[stdCount].marks= stdRecords[stdCount].marks + _bonus;
   }
}
```

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: MarksManagmtSys.stdRecords(int256) data: 0xcd9...
```