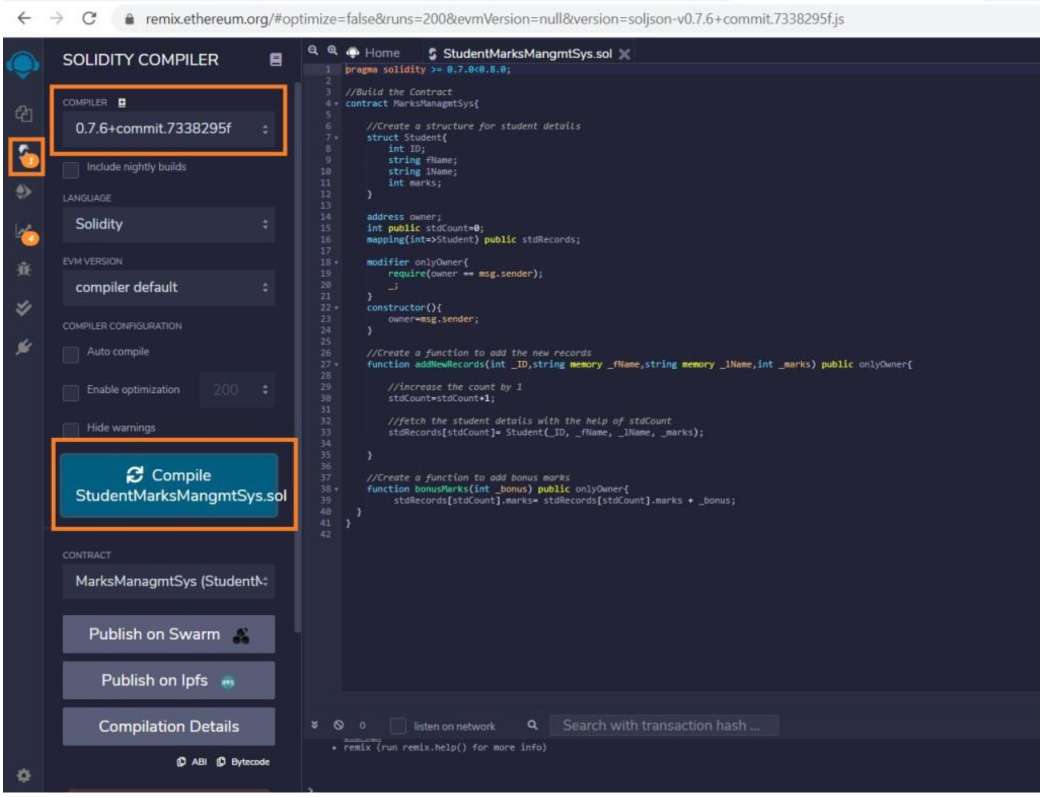
```
pragma solidity >= 0.7.0<0.8.0;
contract MarksManagmtSys
{
        struct Student
        {
                int ID;
                string fName;
                string lName;
                int marks;
        }
        address owner;
        int public stdCount = 0;
        mapping(int => Student)
        public stdRecords;
        modifier onlyOwner
        {
                require(owner == msg.sender);
        }
        constructor()
        {
                owner=msg.sender;
        }
        function addNewRecords(int _ID,  string memory _fName, string memory _lName, int _marks)
        public onlyOwner
        {
                stdCount = stdCount + 1;
                stdRecords[stdCount] = Student(_ID, _fName, _lName, _marks);
        }
        function bonusMarks(int _bonus) public onlyOwner
        {
                stdRecords[stdCount].marks = stdRecords[stdCount].marks + _bonus; } }
```



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

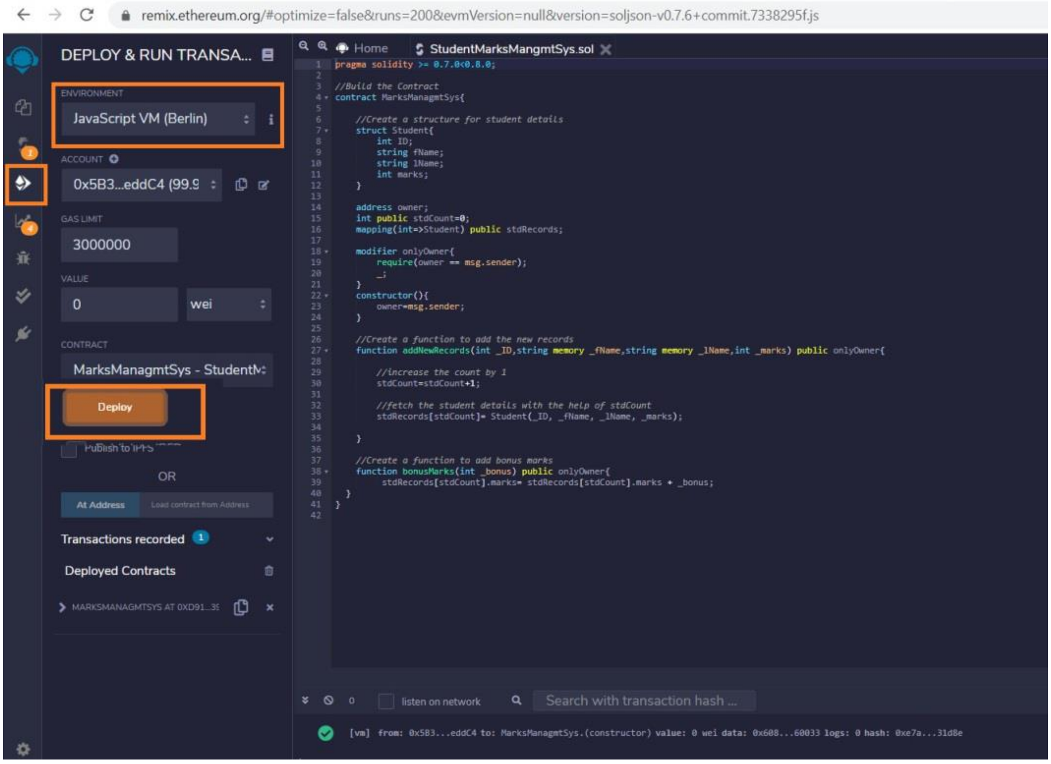
□ Hide warnings

CONTRACT: MarksManagmtSys (StudentM

Compile StudentMarksMangmtSys.sol
Publish on Swarm
Publish on Ipfs
Compilation Details
ABI
Bytecode
Home
StudentMarksMangmtSys.sol
listen on network
Search with transaction hash ...

```
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)
```



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

□ Publish to IPFS

OR

Transactions recorded 1

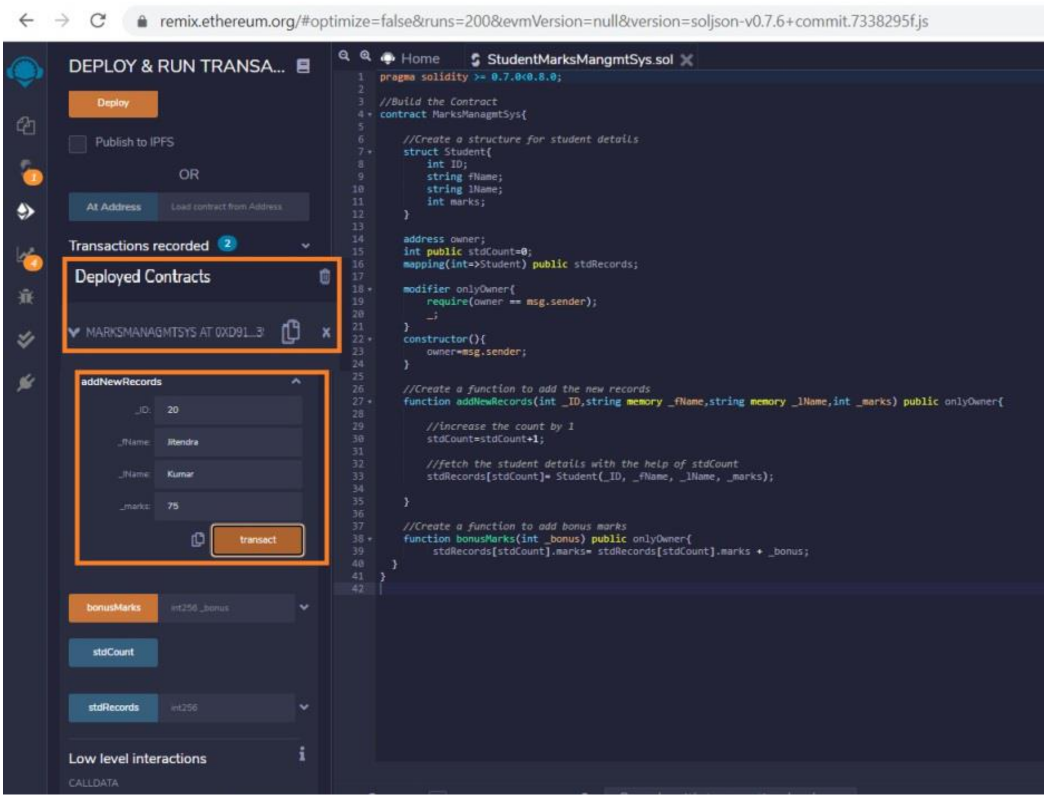
Deployed Contracts

MARKSMANAGMTSYS AT 0XD91...39

Deploy
At Address
Load contract from Address
Home
StudentMarksMangmtSys.sol
listen on network
Search with transaction hash ...

```
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
```



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

addNewRecords

_ID: 20

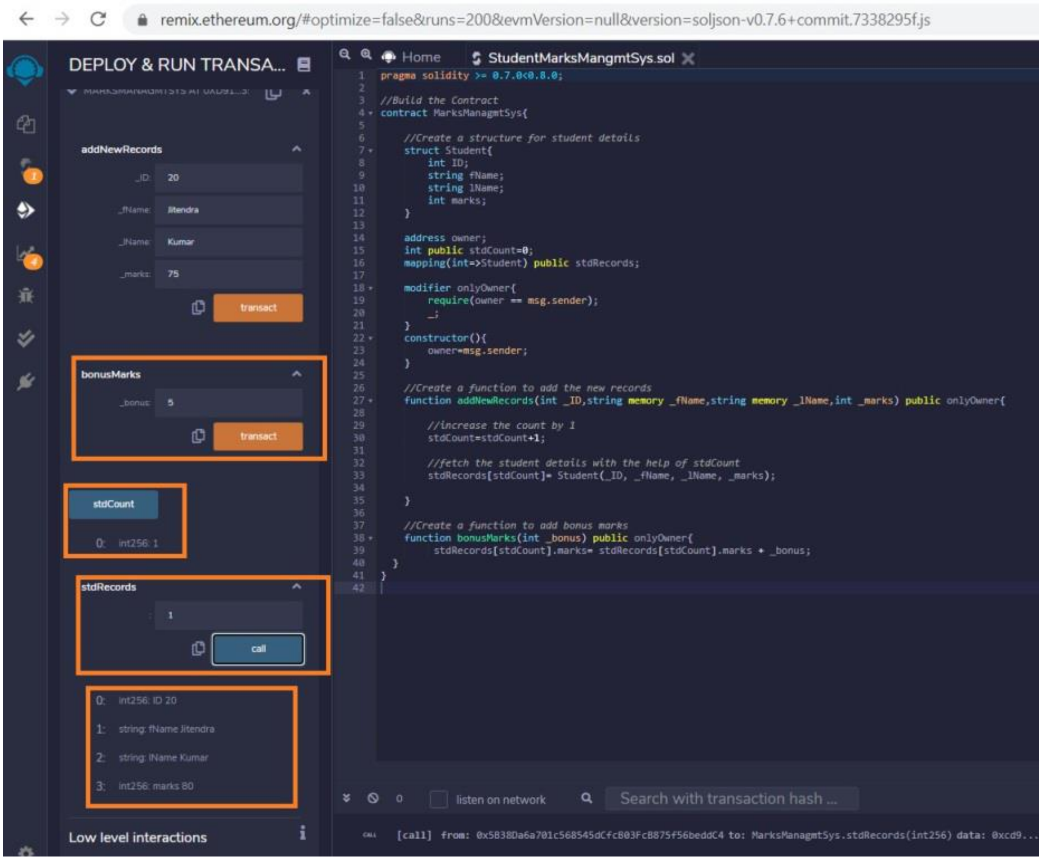
_fName: Jitendra

_lName: Kumar

_marks: 75

transact

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

DEPLOY & RUN TRANSA...
MARKSMANAGMTSYS AT 0XD91...3
Low level interactions
Home
StudentMarksMangmtSys.sol
listen on network
Search with transaction hash ...

addNewRecords

_ID: 20

_fName: Jitendra

_lName: Kumar

_marks: 75

transact

bonusMarks

_bonus: 5

transact

stdCount

0: int256: 1

stdRecords

: 1

call

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