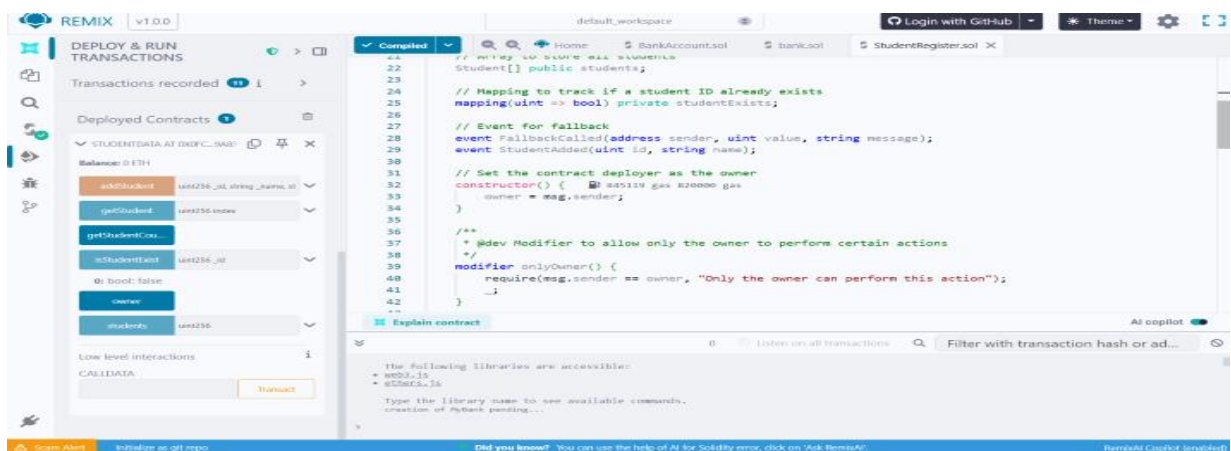
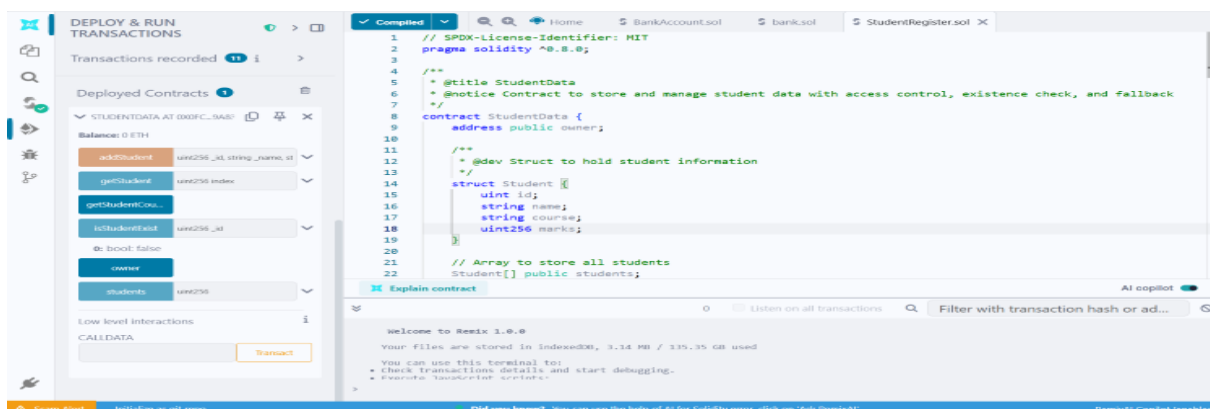
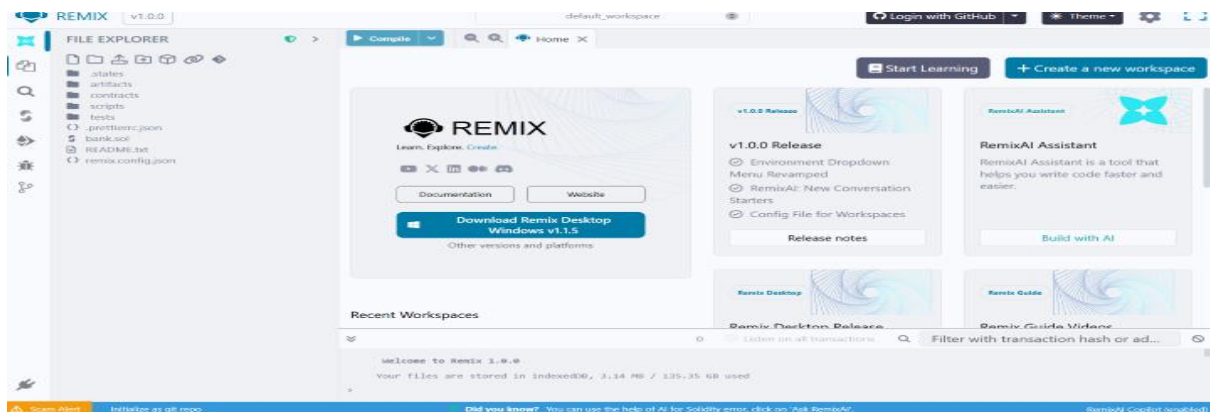


ASSIGNMENT NO:04

AIM: Write a program in solidity to create Student data. Use the following constructs:

- Structures
- Arrays
- Fallback Deploy this as smart contract on Ethereum and Observe the transaction fee and Gas values.



REMIX v1.0.0

default_workspace

Login with GitHub Theme

DEPLOY & RUN TRANSACTIONS

Transactions recorded 11

Deployed Contracts

STUDENTDATA AI (MFC_MAI)

Balance: 0 ETH

addStudent getStudentCount getStudentCna_ isStudentExist

0: bool: false

current

students: 0x4258

Low level interactions

CALLDATA

Transact

```
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
```

function addStudent(uint _id, string memory _name, string memory _course, uint256 _marks) public onlyOwner {
 require(!studentExists[_id], "Student already exists");
 students.push(Student(_id, _name, _course, _marks));
 studentExists[_id] = true;
 emit StudentAdded(_id, _name);
}

function getStudentCount() public view returns (uint) {
 return students.length;
}

Explain contract

The following libraries are accessible:
• web3.js
• ethers.js

Type the library name to see available commands.
creation of MyBank pending...

Screen Alert Initialize as git repo Did you know? You can use the help of AI for Solidity error, click on 'Ask RemixAI' RemixAI Copilot (enabled)

REMIX v1.0.0

default_workspace

Login with GitHub Theme

```
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
```

function getStudentCount() public view returns (uint) {
 return students.length;
}

function getStudent(uint index) public view returns (uint, string memory, string memory, uint256) {
 require(index < students.length, "Invalid index");
 Student memory s = students[index];
 return (s.id, s.name, s.course, s.marks);
}

function isStudentExist(uint _id) public view returns (bool) {
 return studentExists[_id];
}

Explain contract

The following libraries are accessible:
• web3.js
• ethers.js

Type the library name to see available commands.
creation of MyBank pending...

Screen Alert Initialize as git repo Did you know? You can use the help of AI for Solidity error, click on 'Ask RemixAI' RemixAI Copilot (enabled)

REMIX v1.0.0

default_workspace

Login with GitHub Theme

```
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
```

function fallback() external payable {
 emit FallbackCalled(msg.sender, msg.value, "Fallback function was triggered");
}

function receive() external payable {}

Explain contract

The following libraries are accessible:
• web3.js
• ethers.js

Type the library name to see available commands.
creation of MyBank pending...

Screen Alert Initialize as git repo Did you know? You can use the help of AI for Solidity error, click on 'Ask RemixAI' RemixAI Copilot (enabled)

Deploy:

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is open, showing a list of deployed contracts. The main editor displays the Solidity code for 'StudentRegister.sol'. The bottom panel shows the transaction details for the 'addStudent' function.

```
81 return studentexists[_id];
82 }
83
84 /**
85  * @notice Fallback function for invalid calls or direct ETH sends
86  */
87 fallback() external payable { undefined gas
88     emit FallbackCalled(msg.sender, msg.value, "Fallback function was triggered");
89 }
90
91 /**
92  * @notice Receive ETH directly
93  */
94 receive() external payable {} undefined gas
95 }
```

Transaction details: [vm] from: 0x583...eddC4 to: StudentData.addStudent(uint256,string,string,uint256) 0xA0...9688b value: 0 wei data: 0x531...00000 logs: 1 hash: 0xeb2...d7a81

Add Student:

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is open, showing a list of deployed contracts. The main editor displays the Solidity code for 'StudentRegister.sol'. The bottom panel shows the transaction details for the 'getStudent' function.

```
81 return studentexists[_id];
82 }
83
84 /**
85  * @notice Fallback function for invalid calls or direct ETH sends
86  */
87 StructuredDocumentation undefined gas
88     ender, msg.value, "Fallback function was triggered");
89 }
90 StudentRegister.sol 90:4
91
92 /**
93  * @notice Receive ETH directly
94  */
95 receive() external payable {} undefined gas
96 }
```

Transaction details: [call] from: 0x5838Da6a701c568545dCfc803fC875f56beddC4 to: StudentData.getStudent(uint256) data: 0x642...00000

Get Student Data:

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is open, displaying the 'getStudent' function. The function is called with the argument '0'. The transaction is successful, and the output is '0'. The main area shows the transaction details, including the function signature 'getStudent(uint256)' and the transaction hash '0xa608...e0033'. The transaction is successful, and the output is '0'.

Get Total No.of Students:

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is open, displaying the 'getStudentCount' function. The function is called with no arguments. The transaction is successful, and the output is '0'. The main area shows the transaction details, including the function signature 'getStudentCount()' and the transaction hash '0xa608...e0033'. The transaction is successful, and the output is '0'.

Check Student Exist:

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, displaying the 'STUDENTDATA' contract. The 'addStudent' function is selected, and the 'isStudentExist' function is being called with the parameter '1'. The 'owner' field is set to '0x58380a6a701c568545dcfc803fc8875f56beddC4'. The 'students' array is currently empty. The 'Transact' button is visible at the bottom of the panel.

The main editor shows the Solidity code for the 'StudentRegister.sol' file. The code includes a 'return studentExists[1];' statement and a fallback function for invalid calls or direct ETH sends.

The 'Explain contract' panel on the right provides a detailed breakdown of the transaction. It shows the transaction being sent to 'StudentData.addStudent' with the following parameters: '0x58380a6a701c568545dcfc803fc8875f56beddC4' (owner), '1' (uint256), 'Alice' (string), and 'Blockchain' (string). The transaction is pending, and the 'value' is 0 wei. The 'logs' show the 'hash' as '0x6d2...cc078'. The 'call' to 'StudentData.getStudent' is also shown, with the 'data' field set to '0x642...00000'.

Check Owner :

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, displaying the 'STUDENTDATA' contract. The 'isStudentExist' function is selected, and the 'owner' field is being called with the parameter '1'. The 'students' array is currently empty. The 'Transact' button is visible at the bottom of the panel.

The main editor shows the Solidity code for the 'StudentRegister.sol' file. The code includes a 'return studentExists[1];' statement and a fallback function for invalid calls or direct ETH sends.

The 'Explain contract' panel on the right provides a detailed breakdown of the transaction. It shows the transaction being sent to 'StudentData.isStudentExist' with the following parameters: '0x58380a6a701c568545dcfc803fc8875f56beddC4' (owner), '1' (uint256), and 'Alice' (string). The transaction is pending, and the 'value' is 0 wei. The 'logs' show the 'hash' as '0x6d2...cc078'. The 'call' to 'StudentData.isStudentExist' is also shown, with the 'data' field set to '0x642...00000'.

DEPLOY & RUN
TRANSACTIONS

addStudent

getStudent

getStudentCou...

isStudentExist

owner

students

0: uint256: 1

1: string: Alice

2: string: Blockchain

3: uint256: 85

0: uint256: 2

0: bool: false

0: address: 0x5B38Da6a701c568545dCfC803FcB875F56beddC4

0: uint256: id 2

1: string: name Bob

2: string: course Solidity

3: uint256: marks 90

Low level interactions

81
82
83
84
85
86

```
return studentExists[_id];  
  
/**  
 * @notice Fallback function for invalid calls or direct ETH sends  
 */
```

Explain contract

AI copilot

0 Listen on all transactions Filter with transaction hash or ad...

call to StudentData.getStudentCount

0x1 [call] from: 0x5B38Da6a701c568545dCfC803FcB875F56beddC4 to: StudentData.getStudentCount() data: 0x41e...0c407

call to StudentData.isStudentExist

0x2 [call] from: 0x5B38Da6a701c568545dCfC803FcB875F56beddC4 to: StudentData.isStudentExist(uint256) data: 0xe9d...00001

call to StudentData.isStudentExist

0x3 [call] from: 0x5B38Da6a701c568545dCfC803FcB875F56beddC4 to: StudentData.isStudentExist(uint256) data: 0xe9d...00005

call to StudentData.owner

0x4 [call] from: 0x5B38Da6a701c568545dCfC803FcB875F56beddC4 to: StudentData.owner() data: 0x8da...5cbb5

call to StudentData.students

0x5 [call] from: 0x5B38Da6a701c568545dCfC803FcB875F56beddC4 to: StudentData.students(uint256) data: 0x06e...00001

Scam Alert! Initialize as git repo

Did you know? You can use the help of AI for Solidity errors, click on 'Ask RemixAI'.

RemixAI Copilot (enabled)