Subject: Laboratory Practice III **Experiment No:** 16						
Title / Aim of Experiment: 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						
Name of the student:						
Department: **Computer Engineering** **Class:** **B. E**						
Div.:		**Batch:**				
Date of Performance:						
Date of Evaluation:						
Rubrics & Marks	**Completeness** (20%) **2**	**Quality of Work** (40%) **4**	**Organization** (20%) **2**	**Presentation** (10%) **1**	**Creativity & Engagement** (10%) **1**	**Total** **10**
Marks obtained						
CO Mapped	**CO6:** Interpret the basic concepts in Blockchain technology and its applications.					
Signature of Subject Teacher						

Group C

Assignment No: 16

Title of the Assignment: 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.

Objective of the Assignment: Students should be able to learn about Solidity. Its datatypes and implementations.

Prerequisite:

1. Basic Programming Logic
2. Basic knowledge of Solidity

Contents for Theory:

1. **Solidity - Arrays**
2. **Solidity - Structures**
3. **Solidity – Fallback**
4. **Implementation**

---

1. Solidity – Arrays:

Arrays are data structures that store the fixed collection of elements of the same data types in which each and every element has a specific location called index. Instead of creating numerous individual variables of the same type, we just declare one array of the required size and store the elements in the array and can be accessed using the index. In Solidity, an array can be of fixed size or dynamic size. Arrays have a continuous memory location, where the lowest index corresponds to the first element while the highest represents the last.

Creating an Array:

To declare an array in Solidity, the data type of the elements and the number of elements should be specified. The size of the array must be a positive integer and data type should be a valid Solidity type

Syntax:

<data type> <array name>[size] = <initialization>

Fixed-size Arrays:

The size of the array should be predefined. The total number of elements should not exceed the size of the array. If the size of the array is not specified then the array of enough size is created which is enough to hold the initialization.

Dynamic Array:

The size of the array is not predefined when it is declared. As the elements are added the size of array changes and at the runtime, the size of the array will be determined.

Array Operations:

1. **Accessing Array Elements:**

   The elements of the array are accessed by using the index. If you want to access ith element then you have to access (i-1)th index.

2. **Length of Array:**

   Length of the array is used to check the number of elements present in an array. The size of the memory array is fixed when they are declared, while in case the dynamic array is defined at runtime so for manipulation length is required.

3. **Push:**

Push is used when a new element is to be added in a dynamic array. The new element is always added at the last position of the array.

4. **Pop:**

Pop is used when the last element of the array is to be removed in any dynamic array.

2. Solidity – Structures:

Structs in Solidity allows you to create more complicated data types that have multiple properties. You can define your own type by creating a **struct**.

They are useful for grouping together related data.

Structs can be declared outside of a contract and imported in another contract. Generally, it is used to represent a record. To define a structure *struct* keyword is used, which creates a new data type.

Syntax:

```
struct <structure_name> {
   <data type> variable_1;
   <data type> variable_2;
}
```

For accessing any element of the structure, 'dot operator' is used, which separates the struct variable and the element we wish to access. To define the variable of structure data type structure name is used.

3. Solidity – Fallback:

The solidity fallback function is executed if none of the other functions match the function identifier or no data was provided with the function call. Only one unnamed function can be assigned to a contract and it is executed whenever the contract receives plain Ether without any data. To receive Ether and add it to the total balance of the contract, the fallback function must be marked payable. **If no such function exists, the contract cannot receive Ether through regular transactions and will throw an exception.**

Properties of a fallback function:

1. Has no name or arguments.
2. If it is not marked **payable**, the contract will throw an exception if it receives plain ether without data.
3. Cannot return anything.
4. Can be defined once per contract.
5. It is also executed if the caller meant to call a function that is not available
6. It is mandatory to mark it external.
7. It is limited to 2300 gas when called by another function. It is so for as to make this function call as cheap as possible.

Conclusion- In this way we have created array, structure and used fallback function in solidity.

Assignment Question:

1. **What is fixed array and dynamic array in solidity?**
2. **What is Array in solidity?**
3. **What is structure in solidity? Define its syntax.**
4. **What is fallback function?**

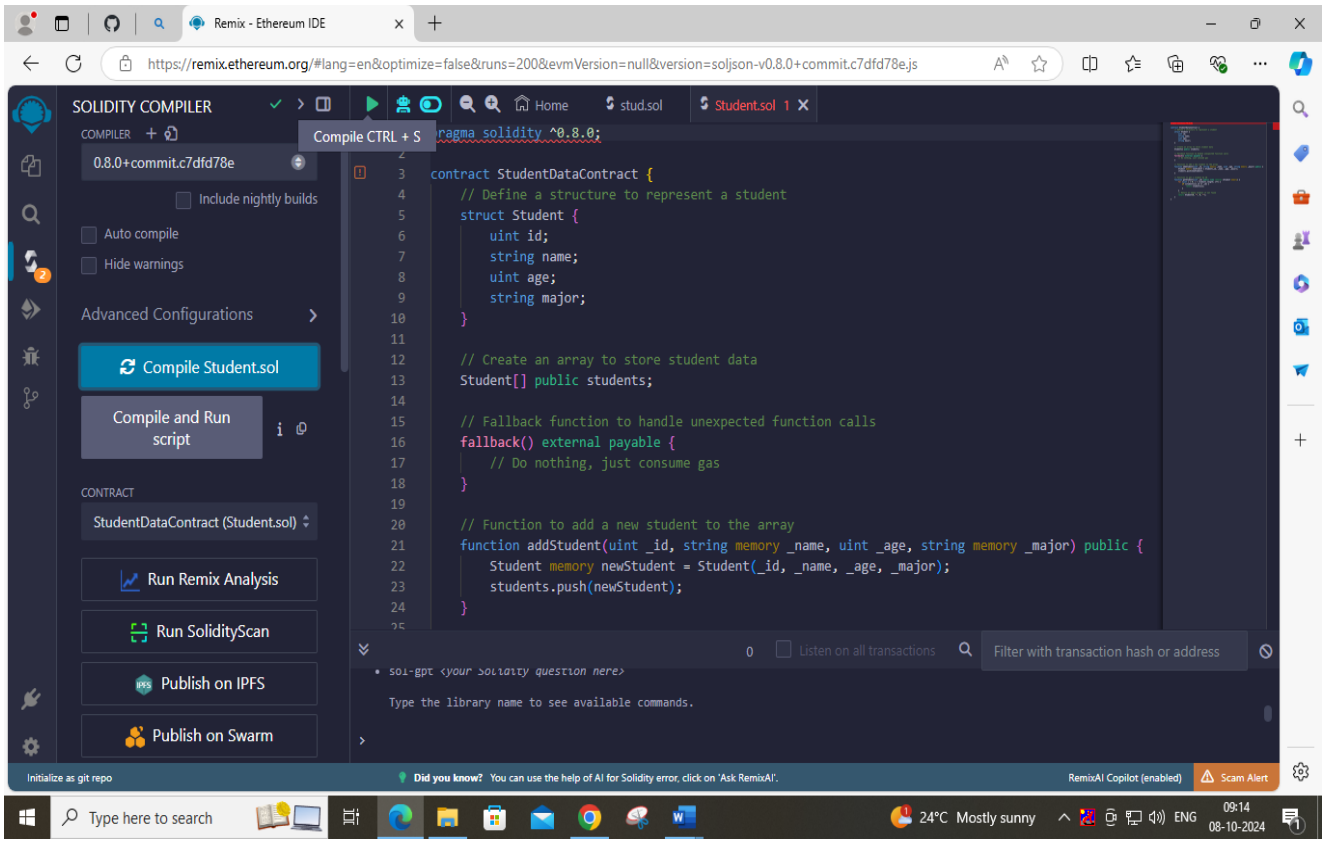
Reference link

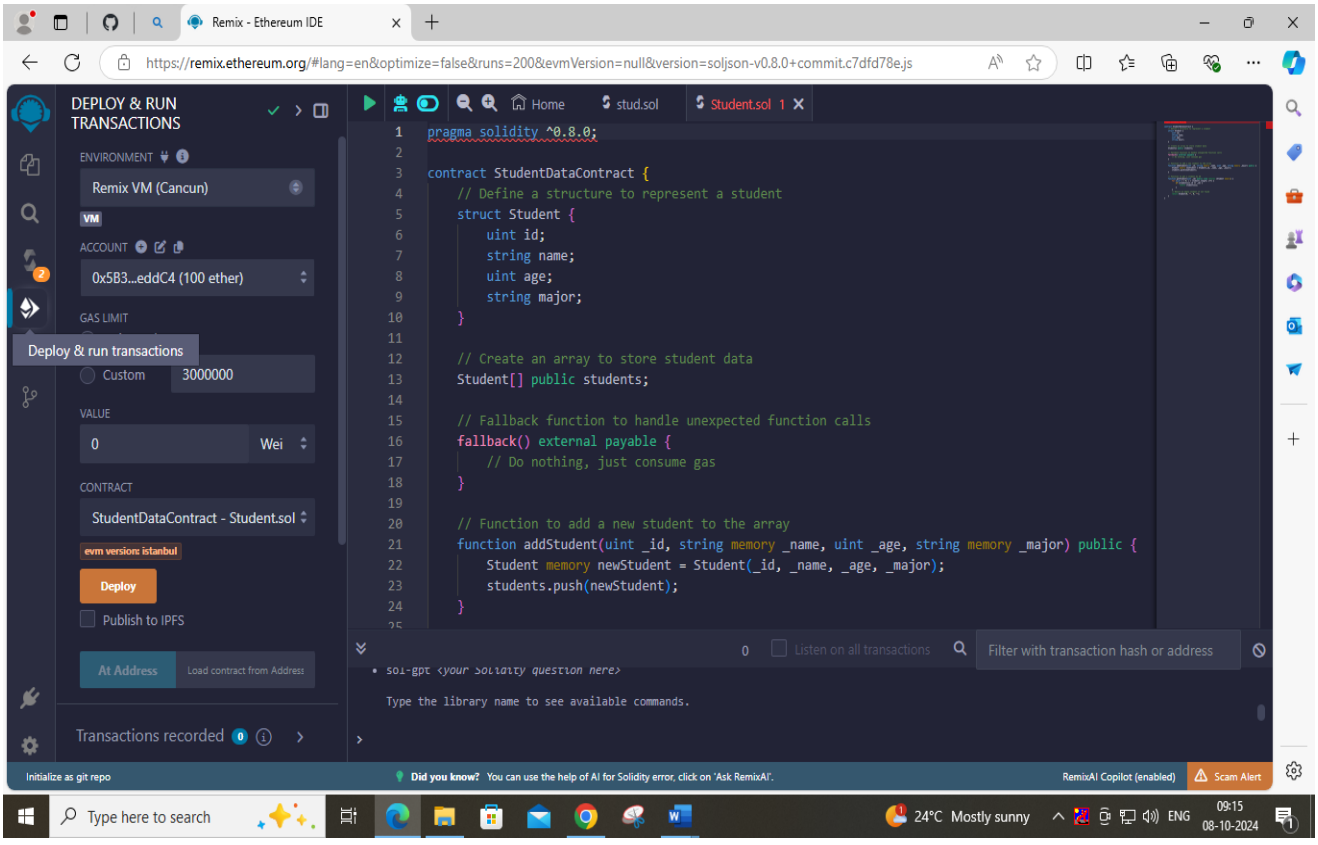
- https://www.geeksforgeeks.org/solidity-arrays/?ref=lbp
- tutorialspoint.com/solidity/solidity_structs.htm

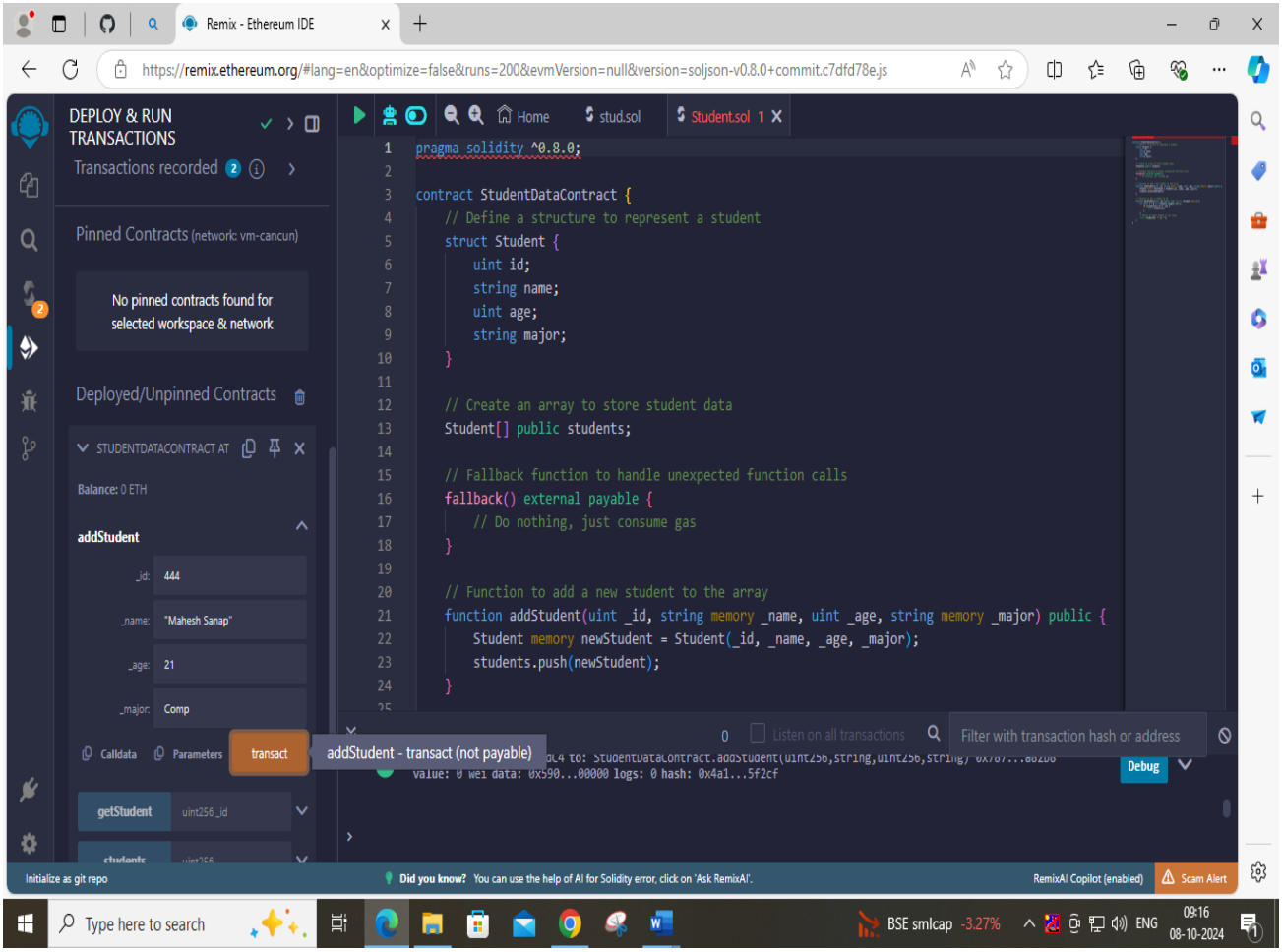
Code –

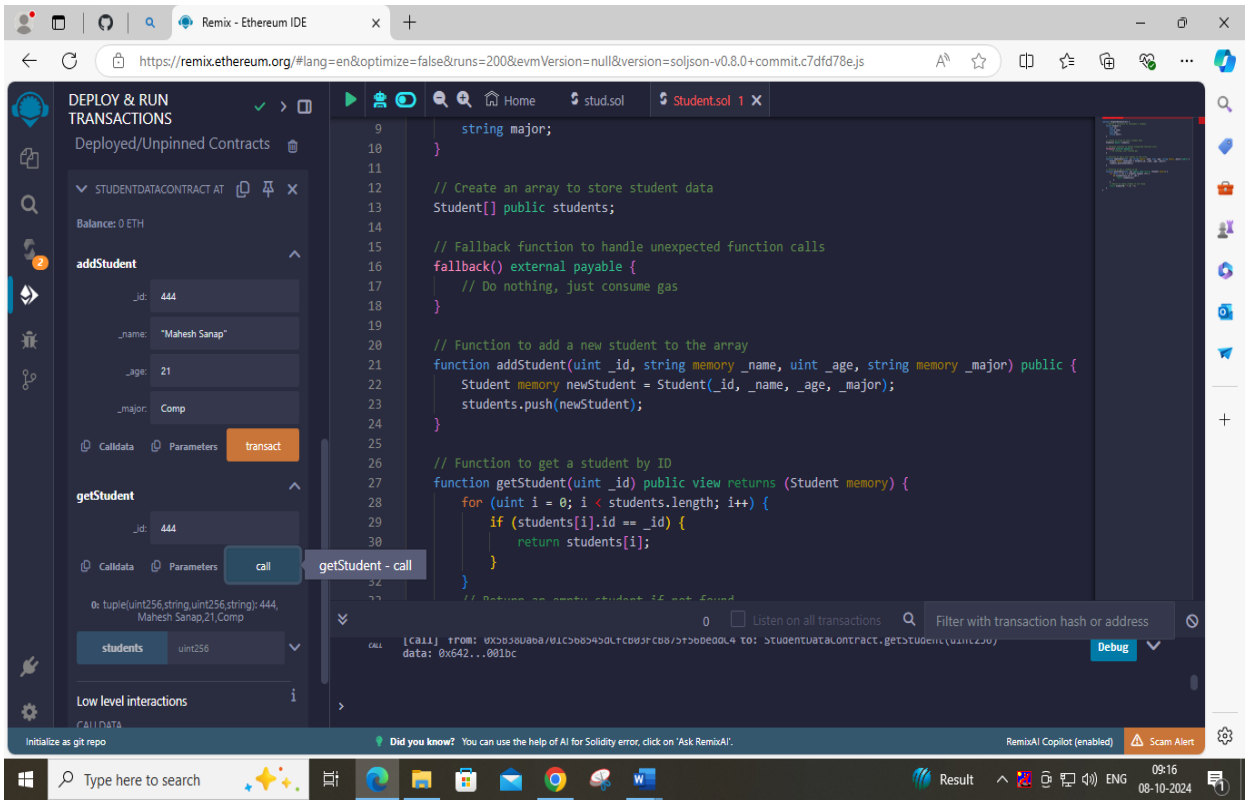
```
pragma solidity ^0.8.0;
contract StudentDataContract {
    // Define a structure to represent a student
    struct Student {
        uint id;
        string name;
        uint age;
        string major;      }
    // Create an array to store student data
    Student[] public students;
    // Fallback function to handle unexpected function calls
    fallback() external payable {
        // Do nothing, just consume gas
    }
    // Function to add a new student to the array
    function addStudent(uint _id, string memory _name, uint _age, string memory _major) public {
        Student memory newStudent = Student(_id, _name, _age, _major);
        students.push(newStudent);
    }
    // Function to get a student by ID
    function getStudent(uint _id) public view returns (Student memory) {
        for (uint i = 0; i < students.length; i++) {
            if (students[i].id == _id) {
                return students[i];
            }
        }
        // Return an empty student if not found
        return Student(0, "", 0, "");
    }
     }
```

Output: -


Remix - Ethereum IDE
https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.0+commit.c7dfd78e.js
SOLIDITY COMPILER
Compile CTRL + S
0.8.0+commit.c7dfd78e
Include nightly builds
Auto compile
Hide warnings
Advanced Configurations
Compile Student.sol
Compile and Run script
CONTRACT
StudentDataContract (Student.sol)
Run Remix Analysis
Run SolidityScan
Publish on IPFS
Publish on Swarm
Home
stud.sol
Student.sol 1
pragma solidity ^0.8.0;
contract StudentDataContract {
// Define a structure to represent a student
struct Student {
uint id;
string name;
uint age;
string major;
}
// Create an array to store student data
Student[] public students;
// Fallback function to handle unexpected function calls
fallback() external payable {
// Do nothing, just consume gas
}
// Function to add a new student to the array
function addStudent(uint _id, string memory _name, uint _age, string memory _major) public {
Student memory newStudent = Student(_id, _name, _age, _major);
students.push(newStudent);
}
Listen on all transactions
Filter with transaction hash or address
Type the library name to see available commands.
Initialize as git repo
Did you know? You can use the help of AI for Solidity error, click on 'Ask RemixAI'.
RemixAI Copilot (enabled)
Scam Alert
Type here to search
24°C Mostly sunny
ENG
09:14
08-10-2024


Remix - Ethereum IDE
https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.0+commit.c7dfd78e.js
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Remix VM (Cancun)
VM
ACCOUNT
0x5B3...eddC4 (100 ether)
GAS LIMIT
Deploy & run transactions
Custom
3000000
VALUE
0
Wei
CONTRACT
StudentDataContract - Student.sol
evm version: istanbul
Deploy
Publish to IPFS
At Address
Load contract from Address
Transactions recorded
Home
stud.sol
Student.sol 1
pragma solidity ^0.8.0;
contract StudentDataContract {
// Define a structure to represent a student
struct Student {
uint id;
string name;
uint age;
string major;
}
// Create an array to store student data
Student[] public students;
// Fallback function to handle unexpected function calls
fallback() external payable {
// Do nothing, just consume gas
}
// Function to add a new student to the array
function addStudent(uint _id, string memory _name, uint _age, string memory _major) public {
Student memory newStudent = Student(_id, _name, _age, _major);
students.push(newStudent);
}
Listen on all transactions
Filter with transaction hash or address
Type the library name to see available commands.
Initialize as git repo
Did you know? You can use the help of AI for Solidity error, click on 'Ask RemixAI'.
RemixAI Copilot (enabled)
Scam Alert
Type here to search
24°C Mostly sunny
ENG
09:15
08-10-2024


Remix - Ethereum IDE
https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.0+commit.c7dfd78e.js
DEPLOY & RUN TRANSACTIONS
Transactions recorded 2
Pinned Contracts (network: vm-cancun)
No pinned contracts found for selected workspace & network
Deployed/Unpinned Contracts
STUDENTDATACONTRACT AT
Balance: 0 ETH
addStudent
_id: 444
_name: "Mahesh Sanap"
_age: 21
_major: Comp
Calldata
Parameters
transact
addStudent - transact (not payable)
getStudent
uint256 _id
Home
stud.sol
Student.sol 1
pragma solidity ^0.8.0;
contract StudentDataContract {
// Define a structure to represent a student
struct Student {
uint id;
string name;
uint age;
string major;
}
// Create an array to store student data
Student[] public students;
// Fallback function to handle unexpected function calls
fallback() external payable {
// Do nothing, just consume gas
}
// Function to add a new student to the array
function addStudent(uint _id, string memory _name, uint _age, string memory _major) public {
Student memory newStudent = Student(_id, _name, _age, _major);
students.push(newStudent);
}
Listen on all transactions
Filter with transaction hash or address
Debug
Initialize as git repo
Did you know? You can use the help of AI for Solidity error, click on 'Ask RemixAI'.
RemixAI Copilot (enabled)
Scam Alert
Type here to search
BSE smlcap -3.27%
ENG
09:16
08-10-2024


Remix - Ethereum IDE
https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.0+commit.c7dfd78e.js
DEPLOY & RUN TRANSACTIONS
Deployed/Unpinned Contracts
STUDENTDATACONTRACT AT
Balance: 0 ETH
addStudent
_id: 444
_name: "Mahesh Sanap"
_age: 21
_major: Comp
Calldata
Parameters
transact
getStudent
_id: 444
Calldata
Parameters
call
getStudent - call
0: tuple(uint256,string,uint256,string): 444, Mahesh Sanap,21,Comp
students
uint256
Low level interactions
CALLDATA
Home
stud.sol
Student.sol 1
string major;
}
// Create an array to store student data
Student[] public students;
// Fallback function to handle unexpected function calls
fallback() external payable {
// Do nothing, just consume gas
}
// Function to add a new student to the array
function addStudent(uint _id, string memory _name, uint _age, string memory _major) public {
Student memory newStudent = Student(_id, _name, _age, _major);
students.push(newStudent);
}
// Function to get a student by ID
function getStudent(uint _id) public view returns (Student memory) {
for (uint i = 0; i < students.length; i++) {
if (students[i].id == _id) {
return students[i];
}
}
Listen on all transactions
Filter with transaction hash or address
Debug
Initialize as git repo
Did you know? You can use the help of AI for Solidity error, click on 'Ask RemixAI'.
RemixAI Copilot (enabled)
Scam Alert
Type here to search
Result
ENG
09:16
08-10-2024