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

1. Structures
2. Arrays
3. Fallback

Observe the transaction fee and Gas values.

Code:

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract StudentData {

// Define a structure for student data

struct Student {

uint id;

string name;

}

// Dynamic array to hold student data

Student[] public students;

// To keep track of the next student ID

uint public nextId = 1;

// Function to create student data

function createStudent(string memory name) public {

students.push(Student(nextId, name));

nextId++;

}

// Function to read student data by ID

function getStudent(uint id) public view returns (uint, string memory) {

for (uint i = 0; i < students.length; i++) {

if (students[i].id == id) {

return (students[i].id, students[i].name);

}

}

revert("Student not found");

}

// Function to update student data by ID

function updateStudent(uint id, string memory name) public {

for (uint i = 0; i < students.length; i++) {

if (students[i].id == id) {

students[i].name = name;

return;

}

}

revert("Student not found");

}

// Function to delete student data by ID

function deleteStudent(uint id) public {

for (uint i = 0; i < students.length; i++) {

if (students[i].id == id) {

delete students[i];

return;

}

}

revert("Student not found");

}

// Fallback function to handle unrecognized calls

fallback() external payable {

// Logic for fallback function (optional)

}

// Function to handle Ether transfers

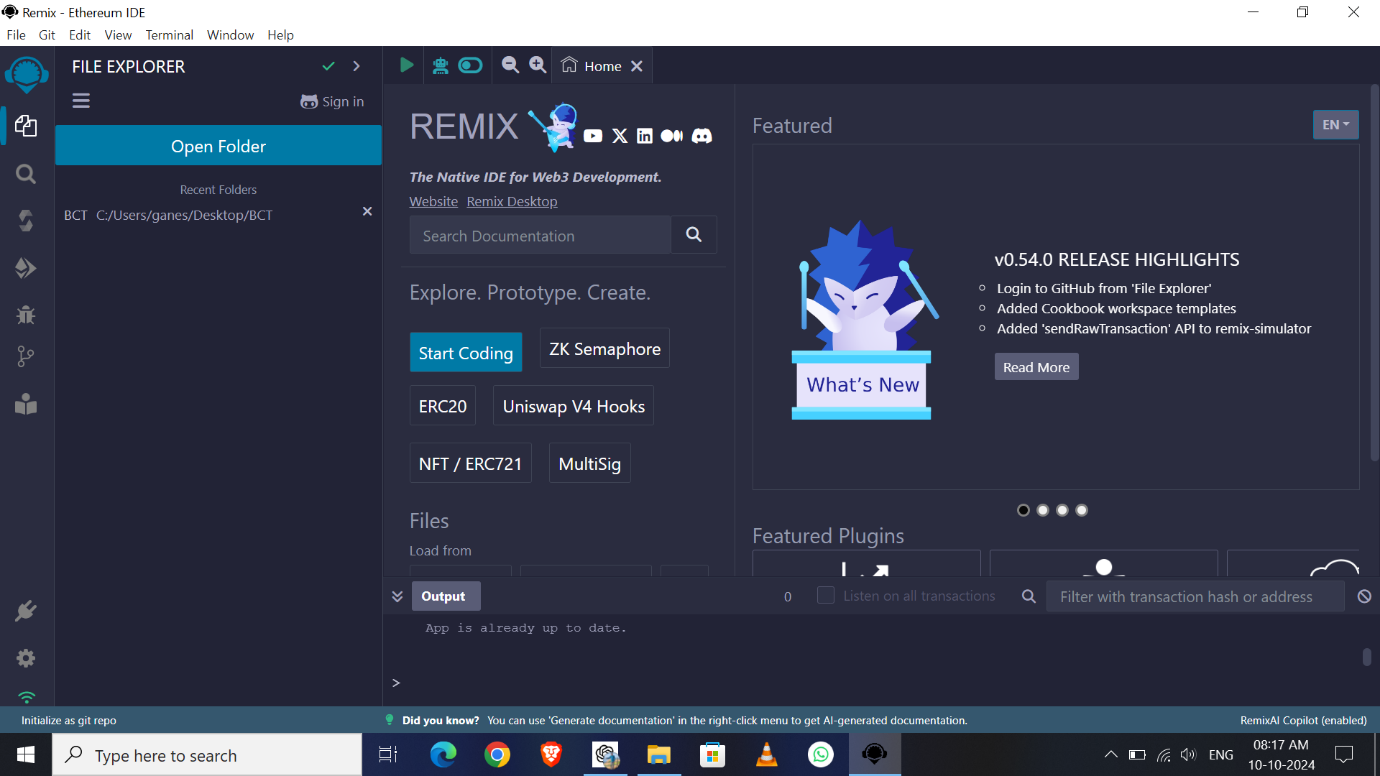
receive() external payable {

// Logic for receiving Ether (optional)

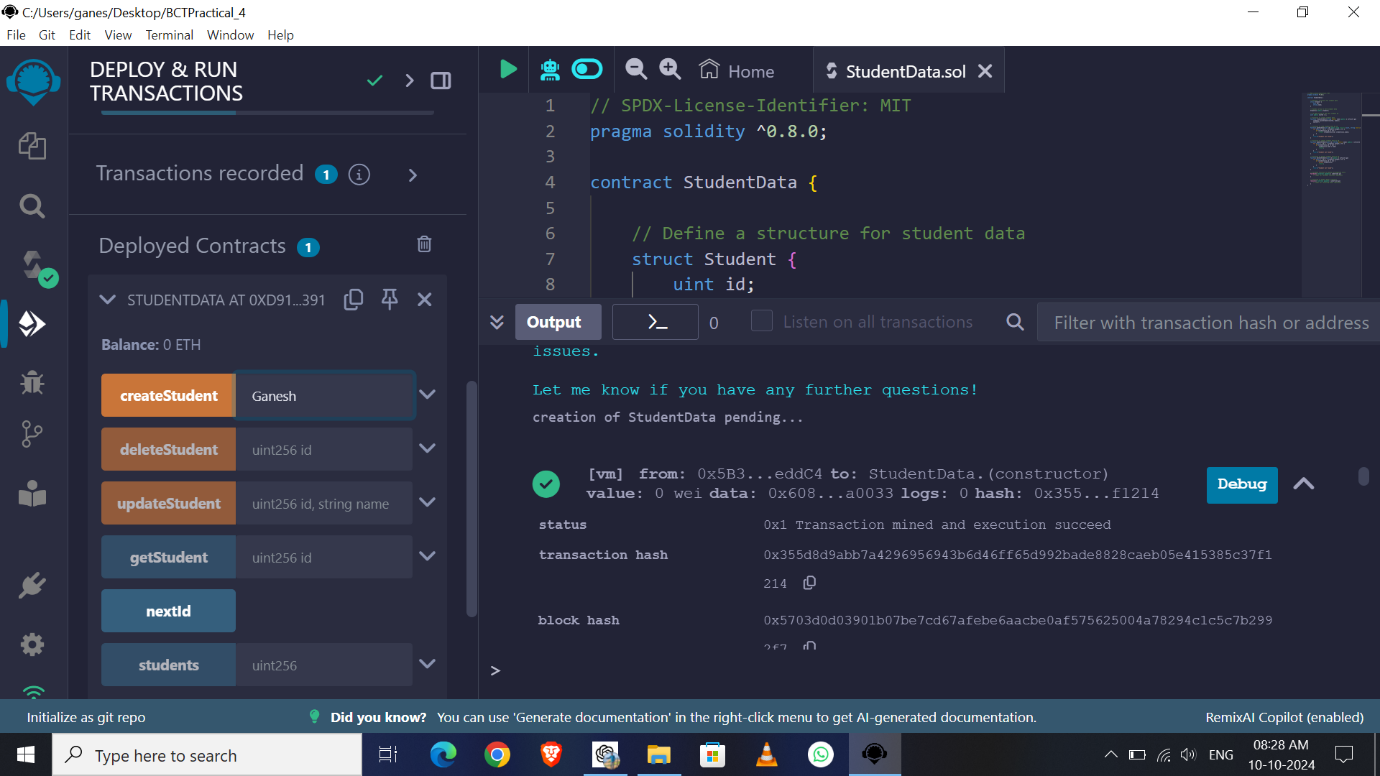
}

}

Output :



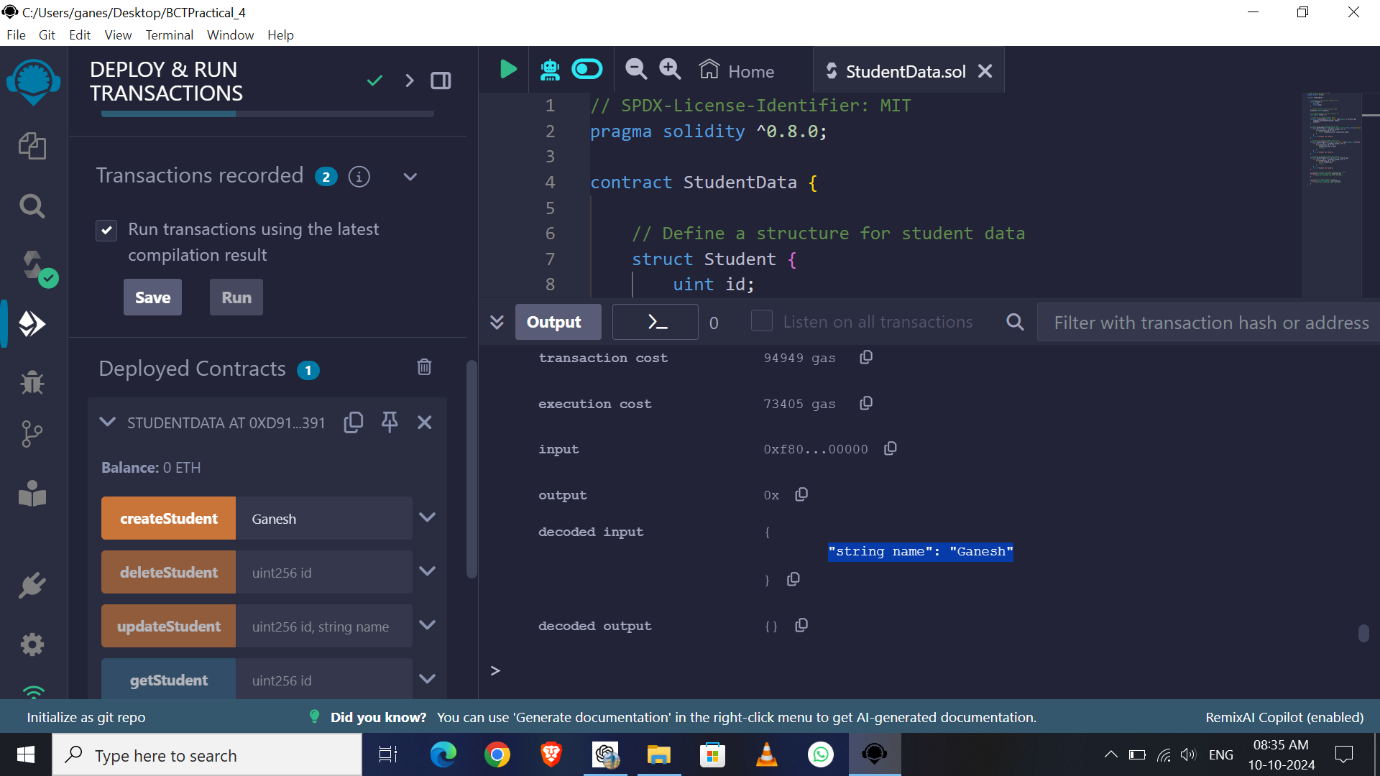
1. **Add Student**



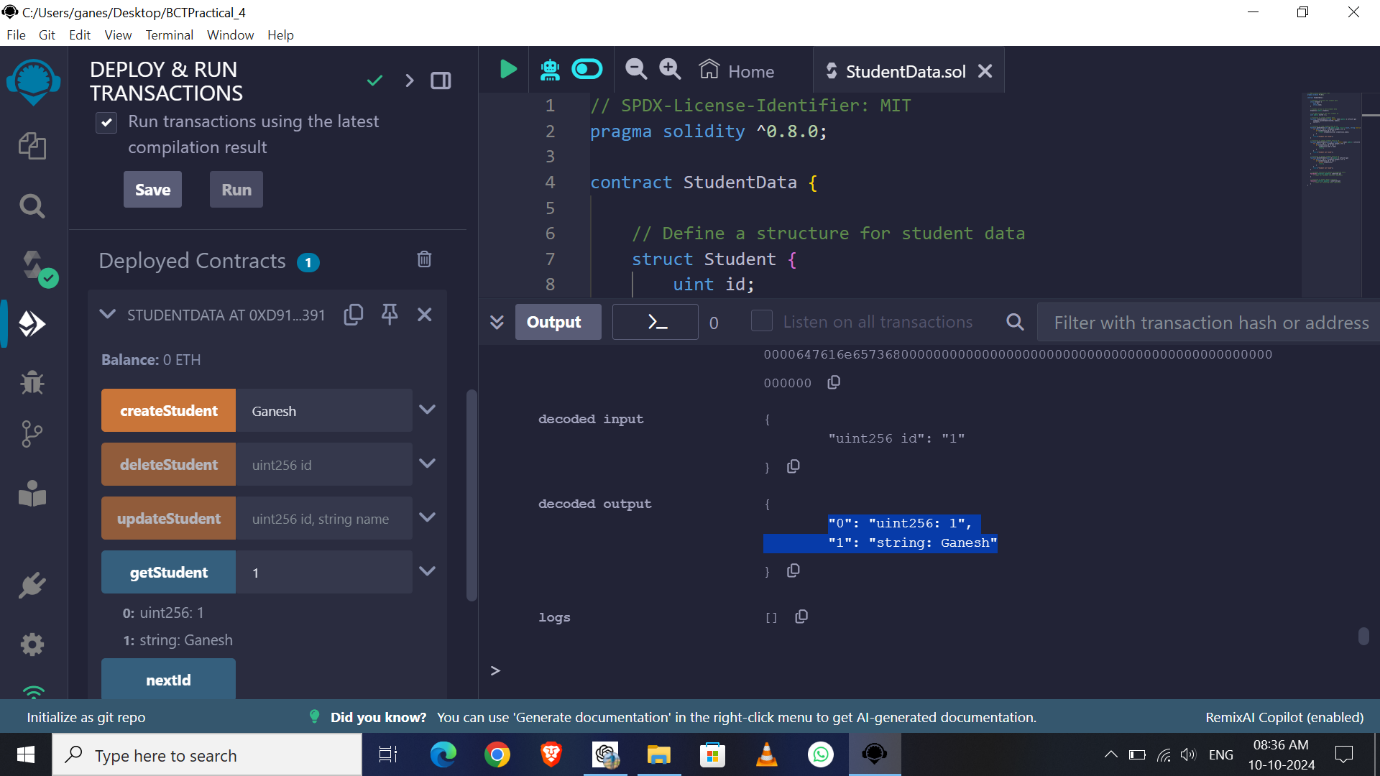
**2) We can add More than One**

**Students also**

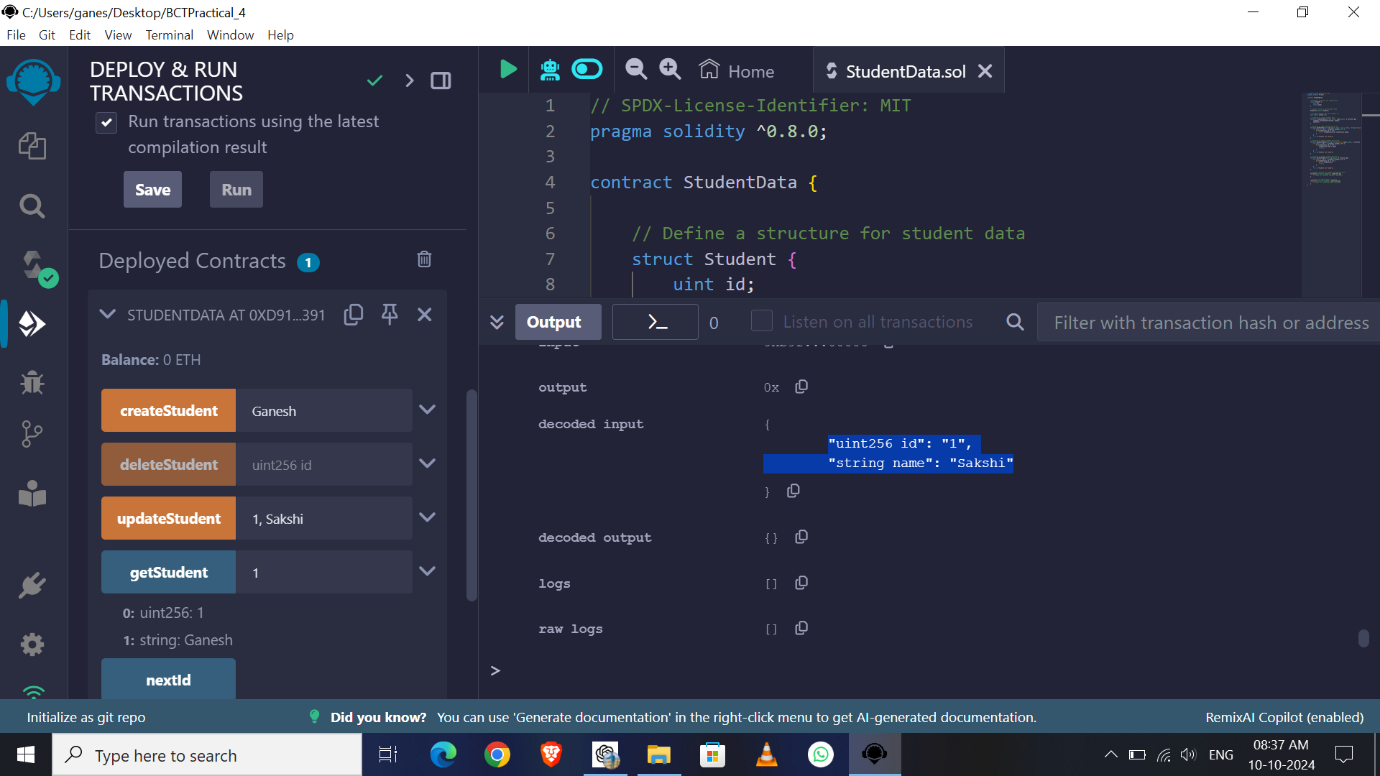
**Now I am added only One**



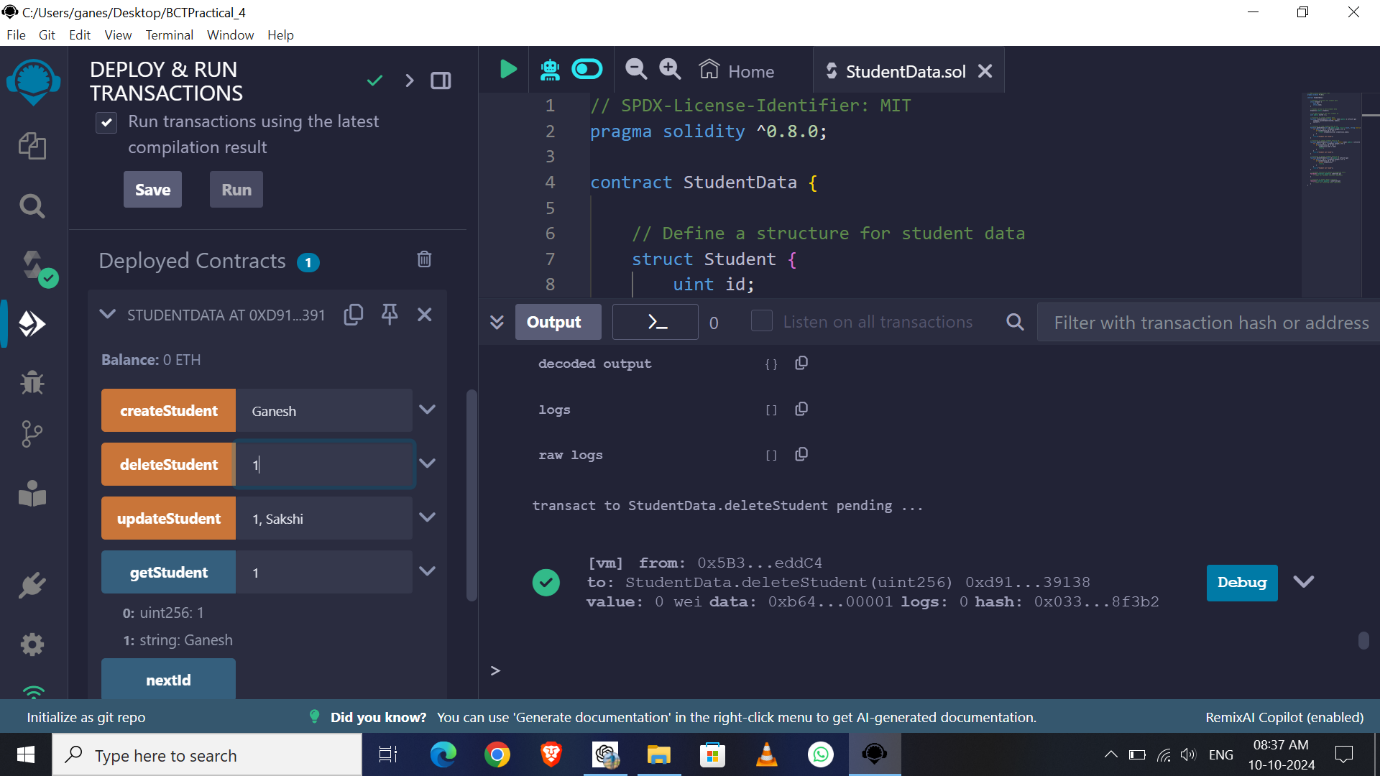
3) **Read Student**



**4) Update Student**



**5) Delete Student**



6) **Observe Transaction**

**and Gas Fees**

**of all operations**

**In CONSOL**

