**Program: -**

```
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract StudentData
{
	struct Student
	{
		int ID;
		string fName;
		string lName;
		int[2] 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[2] memory _marks) public onlyOwner
	{
		stdCount = stdCount + 1;
		stdRecords[stdCount] = Student(_ID, _fName,_lName,
_marks);
	}

}
```

**Output:**

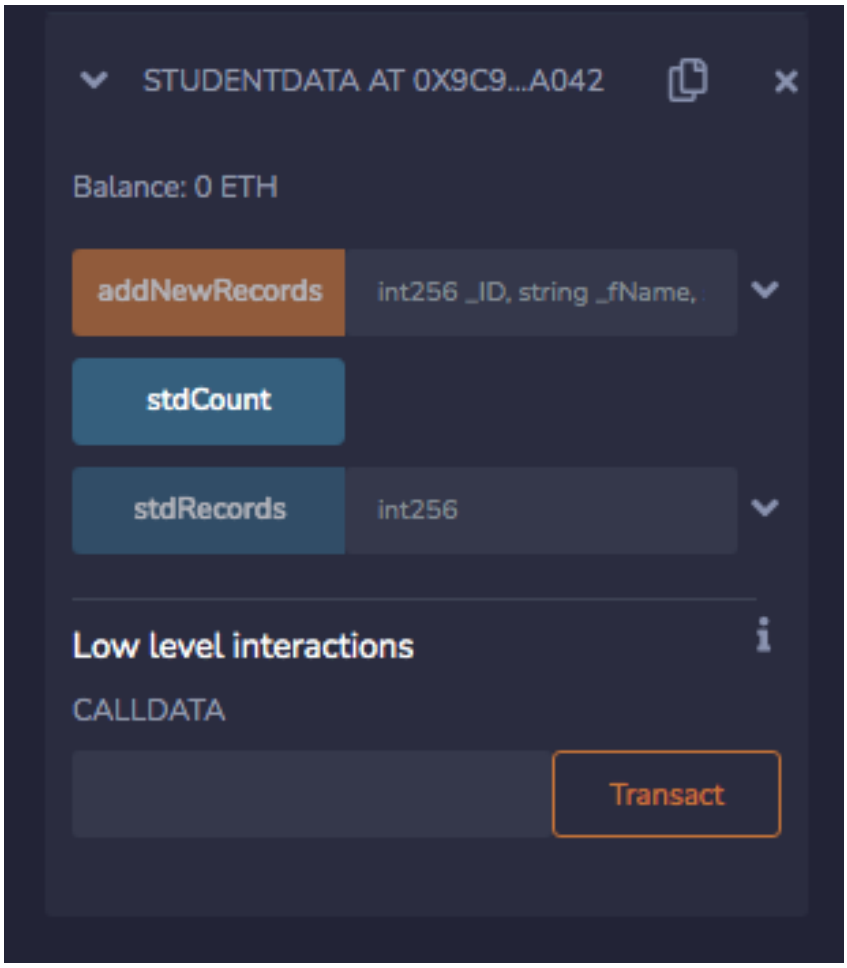


Fig : 1 Deployed Instance

| | |
|---|---|
| block hash | 0xeda8a86fbb3bf34a88b414b9644d40edb9748a2a50daf0b195f47a1249a97a17 |
| block number | 16 |
| contract address | 0x1482717Eb2eA8Ecd81d2d8C403CaCF87AcF04927 |
| from | 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 |
| to | StudentData.(constructor) |
| gas | 778669 gas |
| transaction cost | 677283 gas |
| execution cost | 580359 gas |
| input | 0x608...20033 |
| decoded input | {} |
| decoded output | - |

Fig : 2 Initial Transaction and execution costs and gas after deploying contract

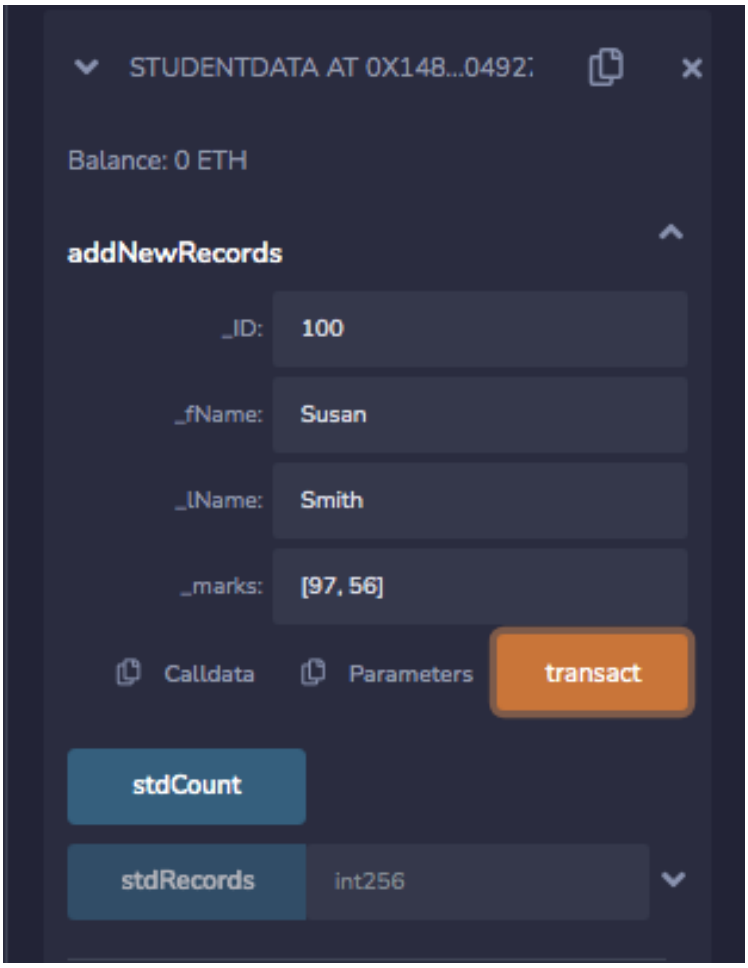


Fig : 3 Entering Input Data


Fig : 4 Transaction and execution costs and gas after passing input

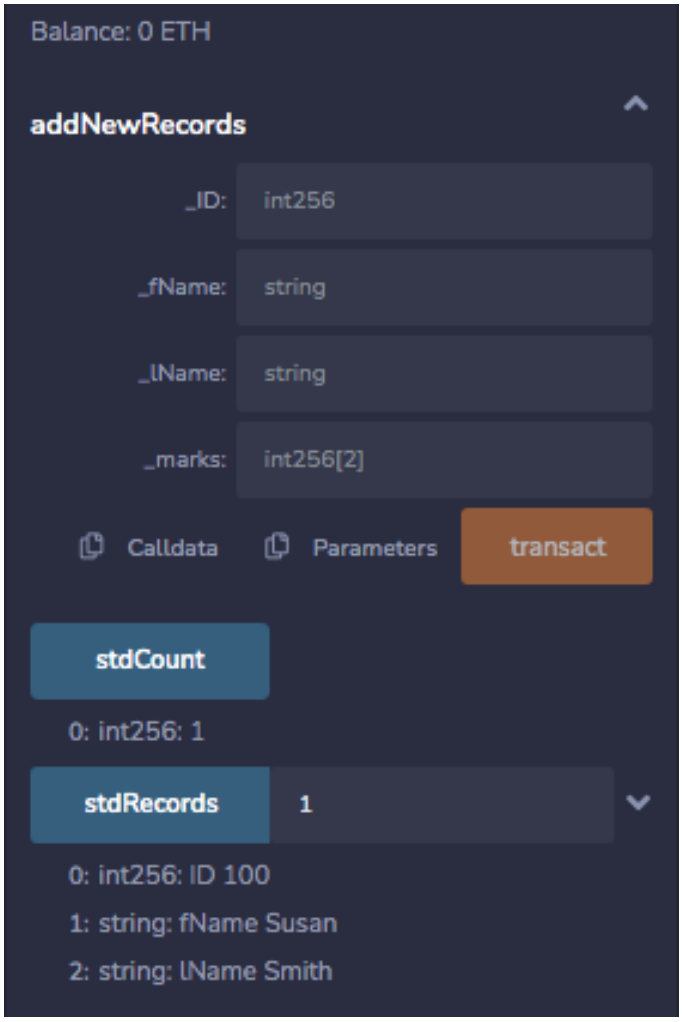


Fig : 5 Calling stdCount and stdRecords


Fig : 6 Execution costs and gas after calling stdCount