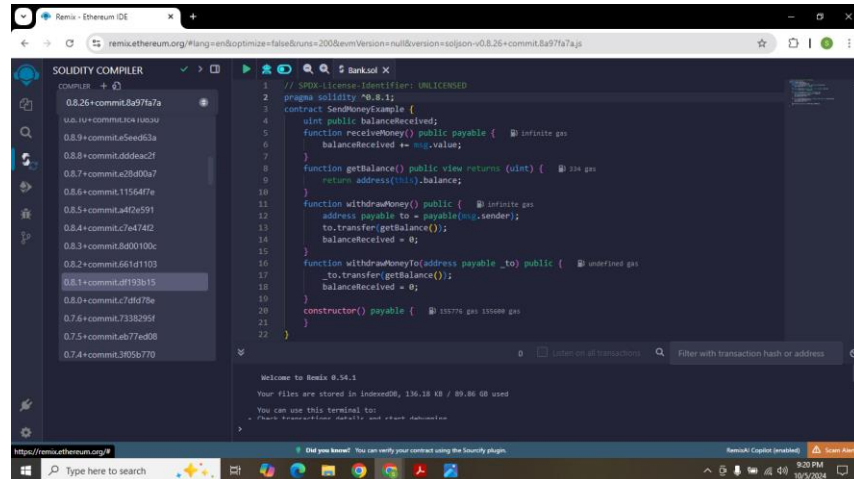
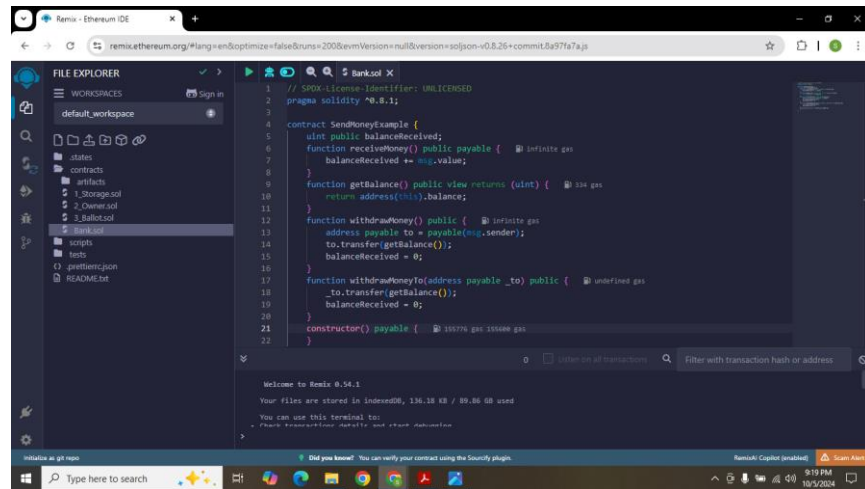
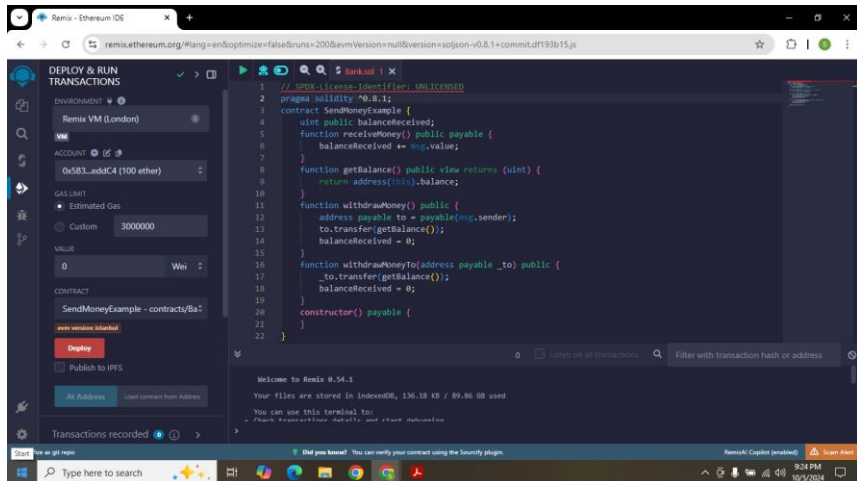
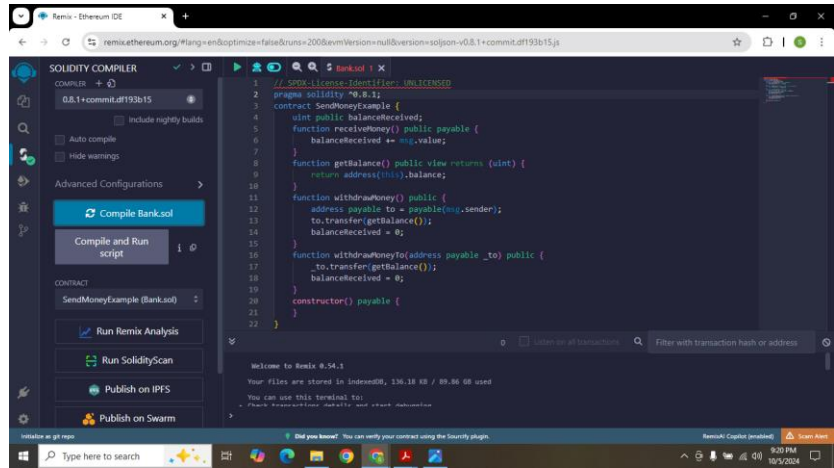
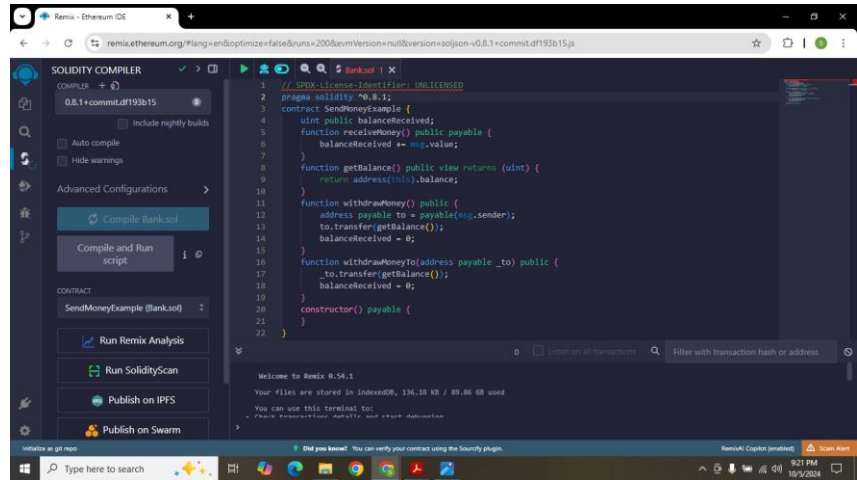


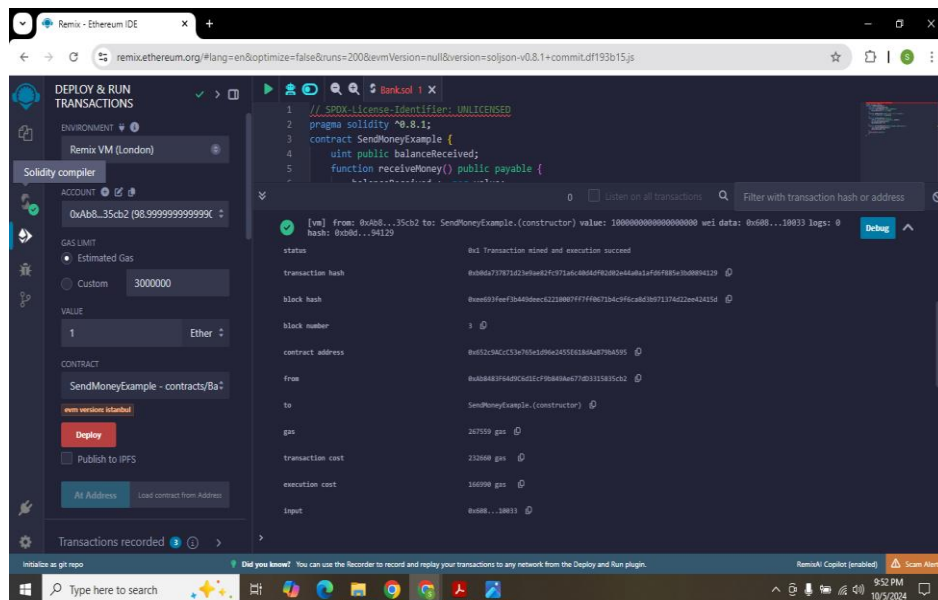
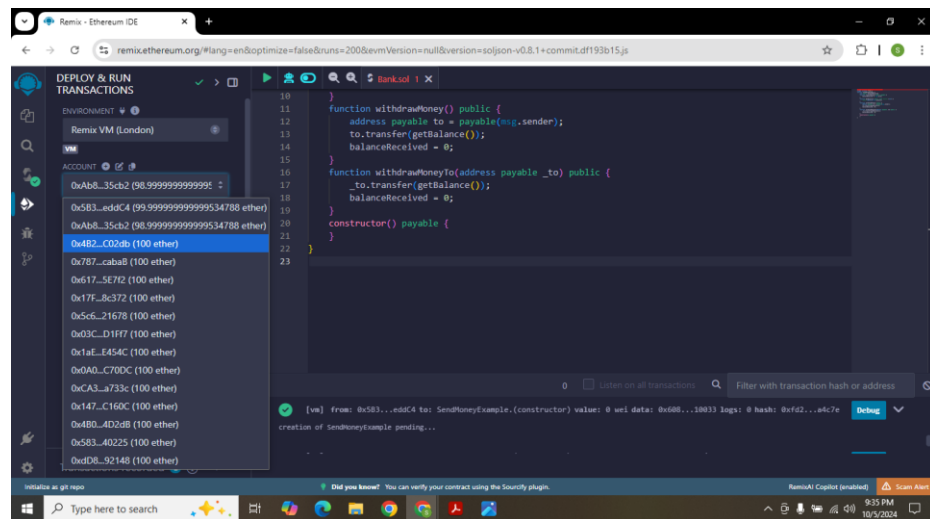
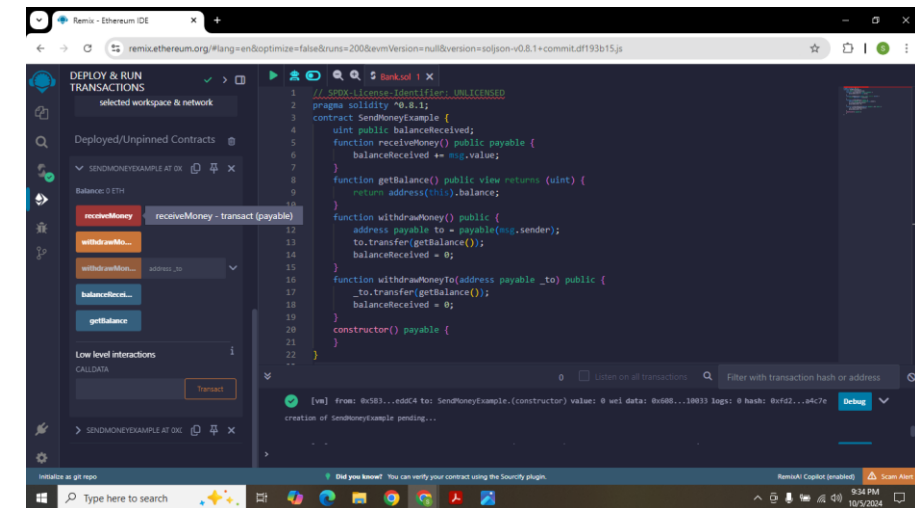
PRACTICAL NO 3

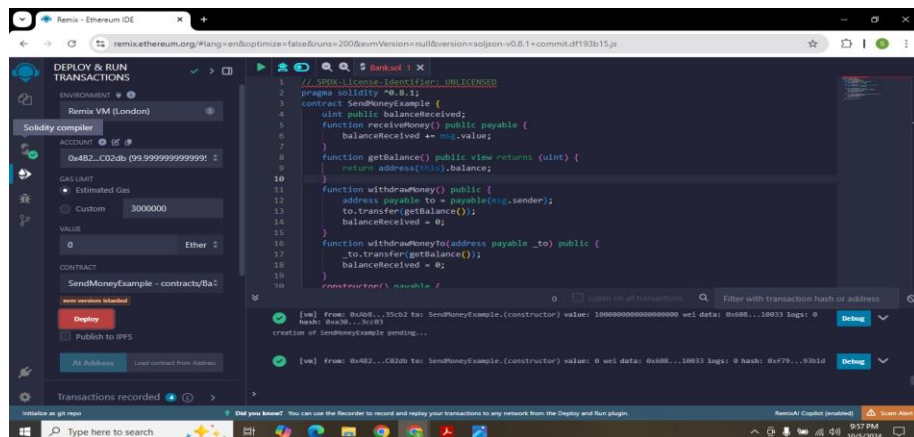
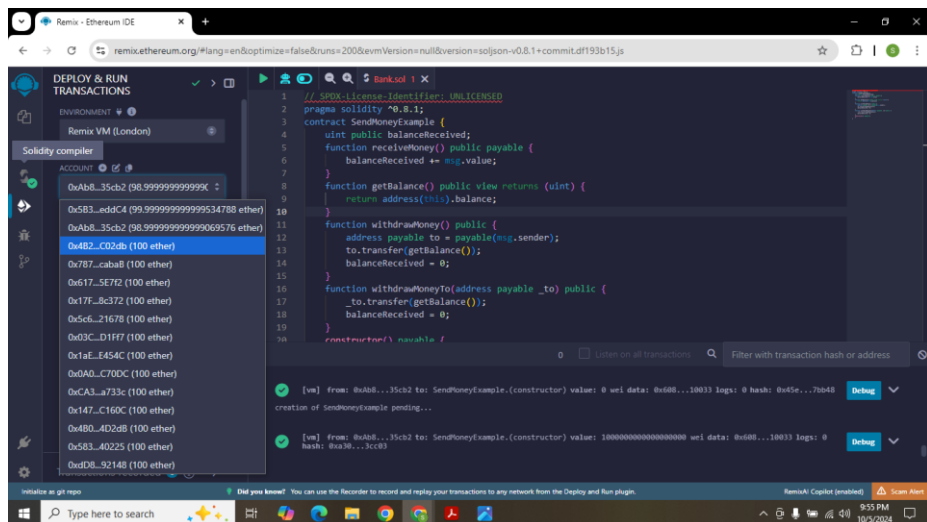
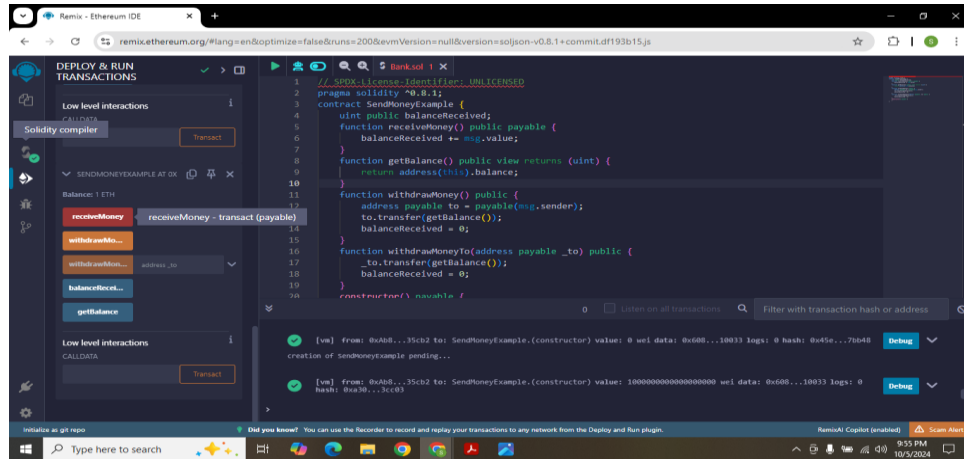
Aim: Write a smart contract on a test network, for Bank account of a customer for following operations:

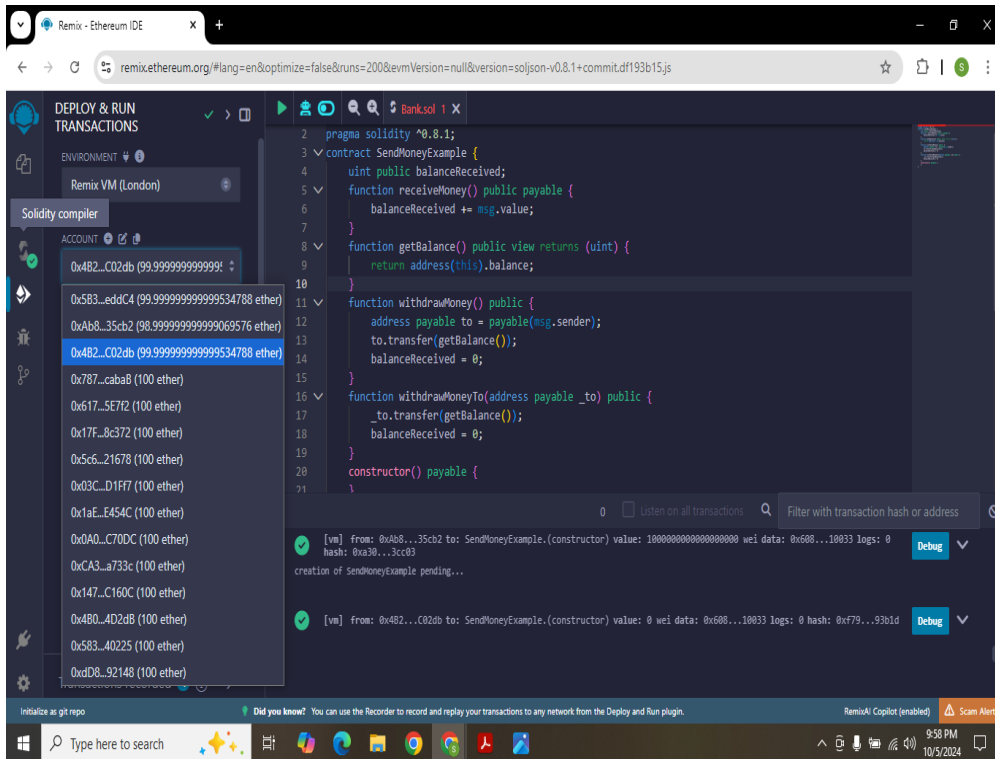
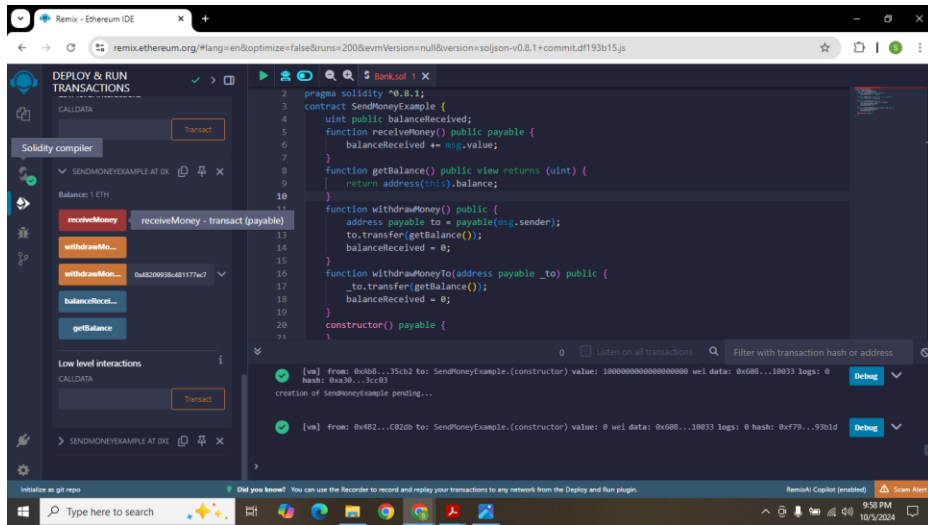
- ☐ Deposit money
- ☐ Withdraw Money
- ☐ Show balance









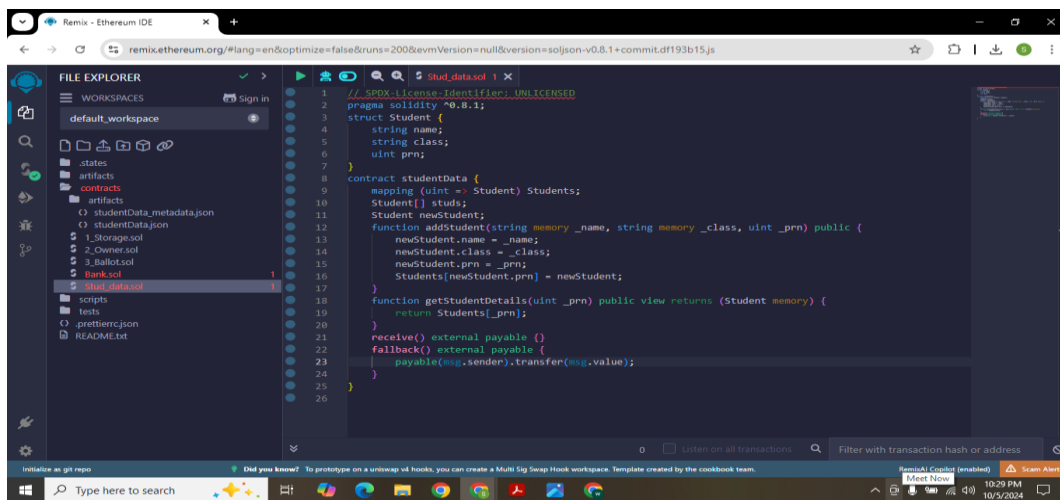


PRACTICAL NO : 4

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

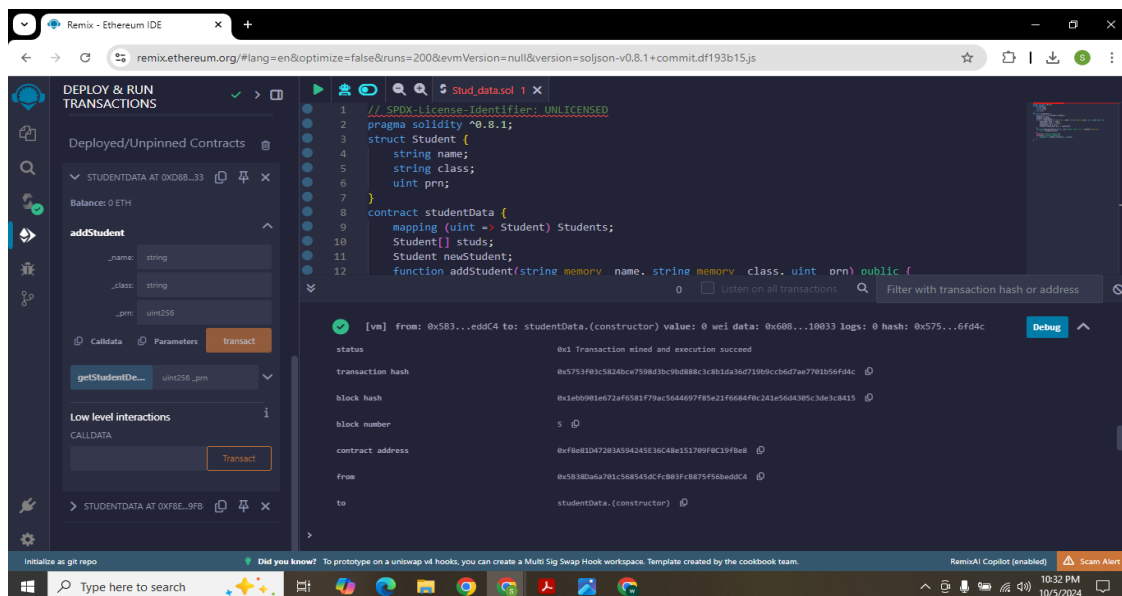
- ☐ Structures
- ☐ Arrays
- ☐ Fallback

Deploy this as smart contract on Ethereum and Observe the transaction fee and Gas values.



The screenshot shows the Remix IDE with the Solidity code for the `studentData` contract. The code is as follows:

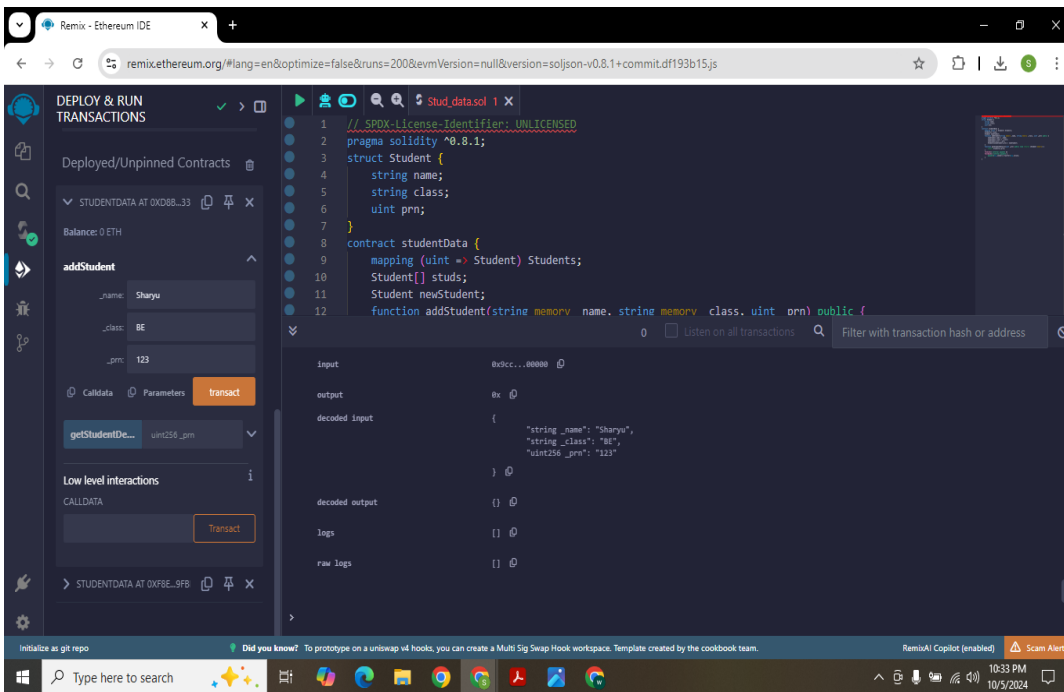
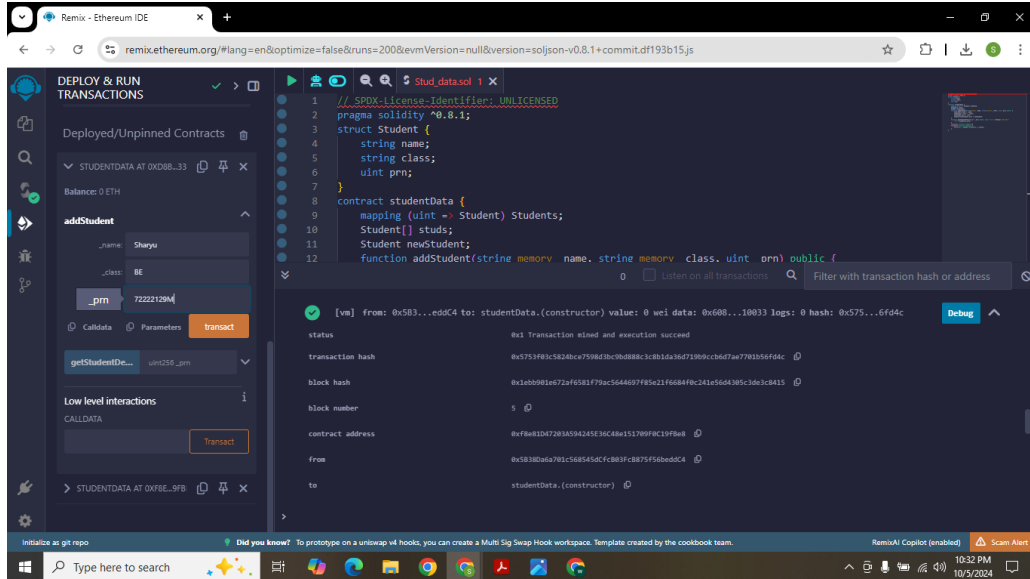
```
1 // SPDX-License-Identifier: UNLICENSED
2 pragma solidity ^0.8.1;
3 struct Student {
4     string name;
5     string class;
6     uint prn;
7 }
8 contract studentData {
9     mapping (uint => Student) Students;
10    Student[] studs;
11    Student newStudent;
12    function addStudent(string memory _name, string memory _class, uint _prn) public {
13        newStudent.name = _name;
14        newStudent.class = _class;
15        newStudent.prn = _prn;
16        Students[newStudent.prn] = newStudent;
17    }
18    function getStudentDetails(uint _prn) public view returns (Student memory) {
19        return Students[_prn];
20    }
21    receive() external payable {}
22    fallback() external payable {
23        payable(msg.sender).transfer(msg.value);
24    }
25 }
26
```



The screenshot shows the Remix IDE with the deployment and transaction details of the `studentData` contract. The left sidebar shows the `DEPLOY & RUN TRANSACTIONS` panel with the `addStudent` function selected. The right sidebar shows the transaction details for the `addStudent` function.

Transaction Details:

- status: Transaction mined and execution succeed
- transaction hash: 0x573f83c824bce759843bc9d888c3c8b1da36d719b9cbbd7ae7701b56f4dc
- block hash: 0x1eb0981a672af6581f79ac56446977f8e21f6684fbc241e56d4385c3dc3c8415
- block number: 5
- contract address: 0xf8e81047203A594245E36C48e15170F8C19f8e8
- from: 0x58380da781c568545dcfc803fc875f56bd0c4
- to: studentData.(constructor)



Remix - Ethereum IDE

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.1+commit.df193b15.js

DEPLOY & RUN TRANSACTIONS

Deployed/Unpinned Contracts

STUDENTDATA AT 0xD08B...33

Balance: 0 ETH

addStudent

_name: Sharyu

_class: BE

_prn: 123

CallData Parameters **transact**

getStudentDetails

_prn: 1

CallData Parameters **call**

tuple(string,string,uint256)_0

Low level interactions

CALLDATA **Transact**

```
1 // SPDX-License-Identifier: UNLICENSED
2 pragma solidity ^0.8.1;
3 struct Student {
4     string name;
5     string class;
6     uint prn;
7 }
8 contract studentData {
9     mapping (uint => Student) Students;
10    Student[] studs;
11    Student newStudent;
12    function addStudent(string memory name, string memory class, uint prn) public {
```

0 Listen on all transactions Filter with transaction hash or address

decoded input

```
{
  "uint256_prn": "1"
}
```

decoded output

```
{
  "0": "tuple(string,string,uint256): ,0"
}
```

logs

```
{
}
```

raw logs

```
{
}
```

Initialize as git repo Did you know? To prototype on a uniswap v4 hooks, you can create a Multi Sig Swap Hook workspace. Template created by the cookbook team. RemixAI Copilot (enabled) Scan Alert

Type here to search

10:34 PM 10/5/2024