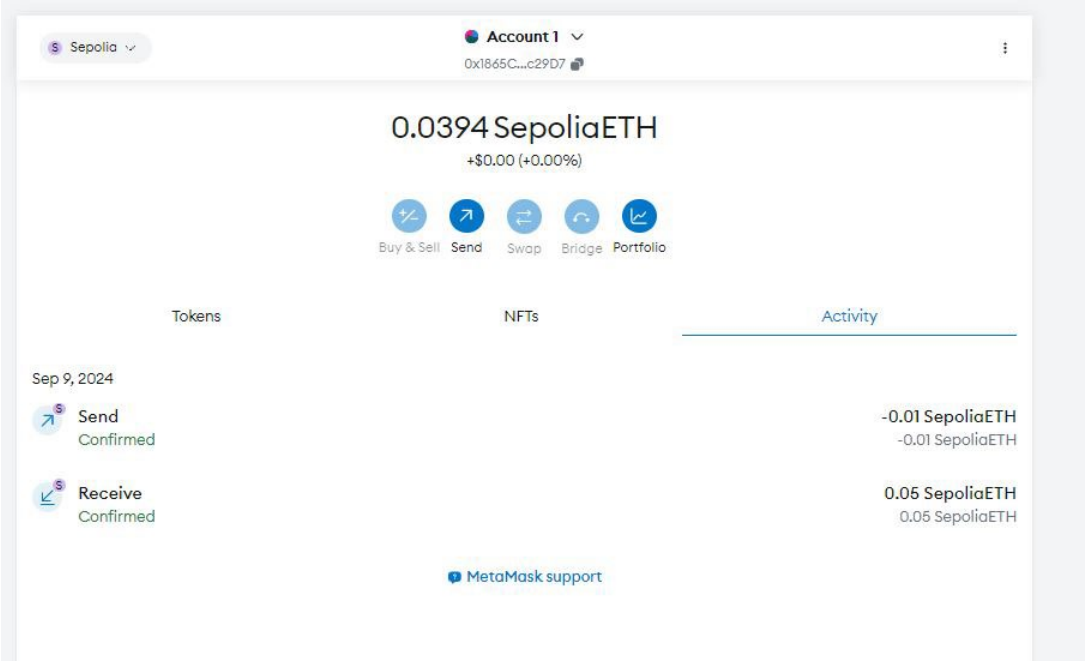
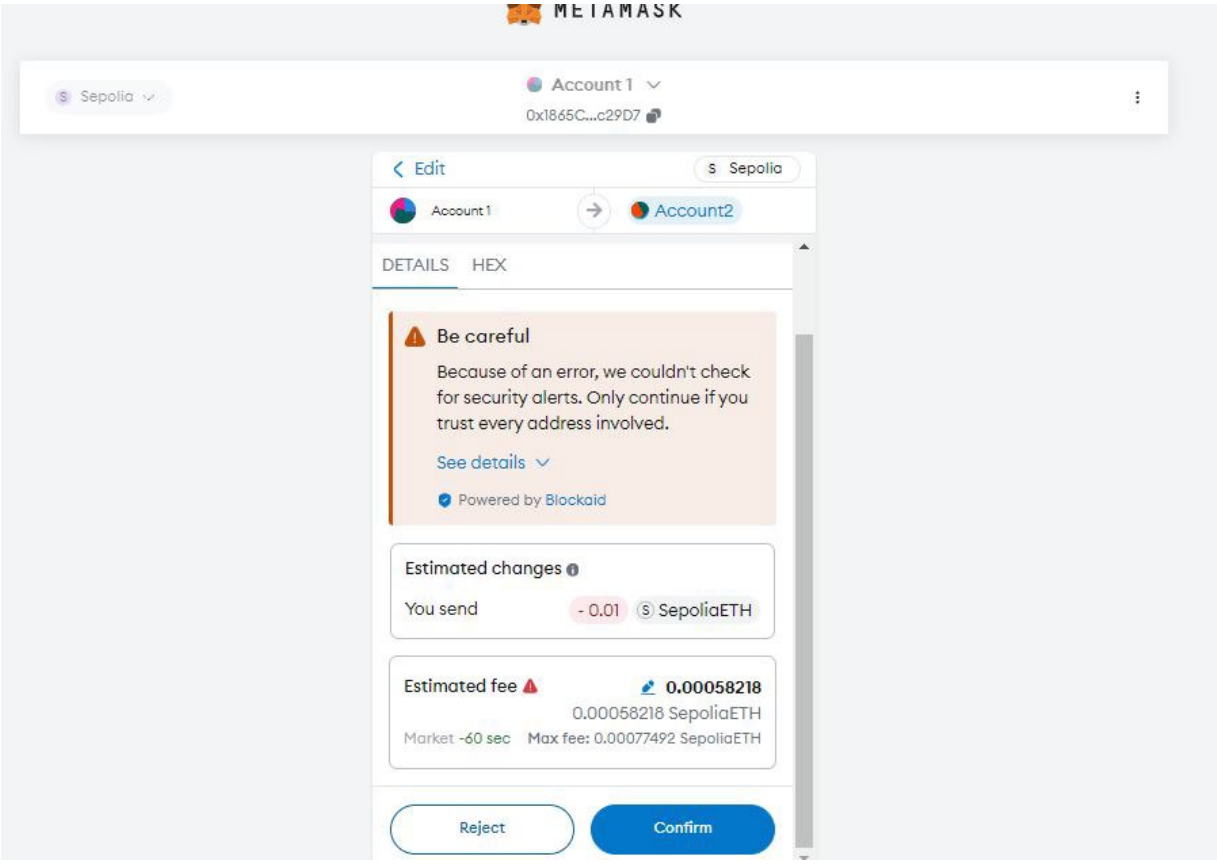
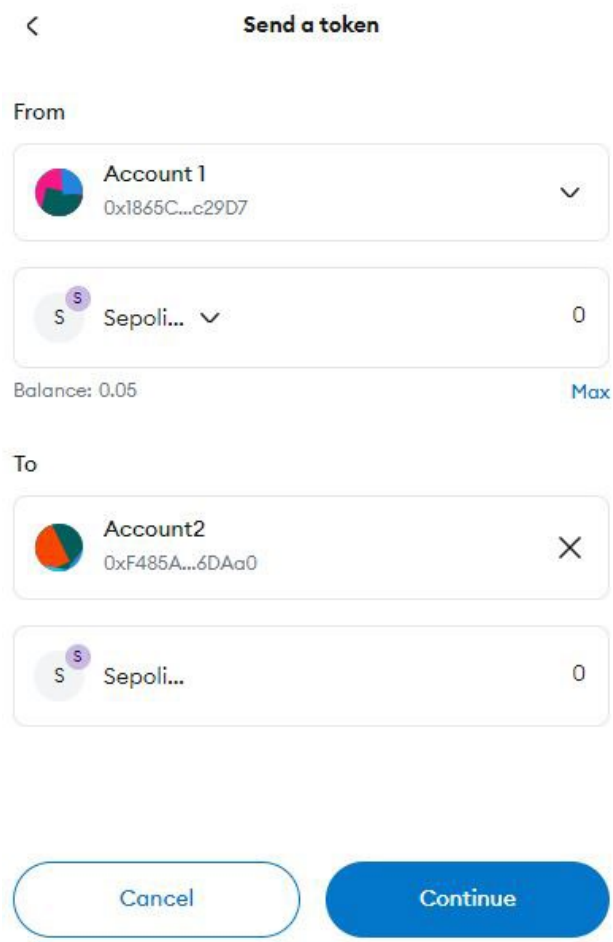
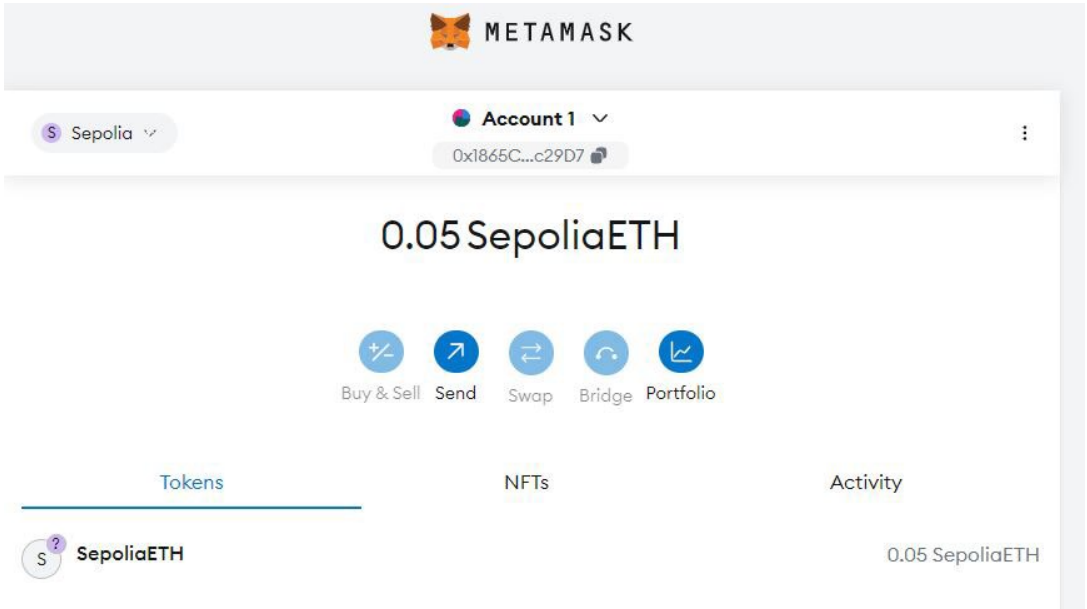
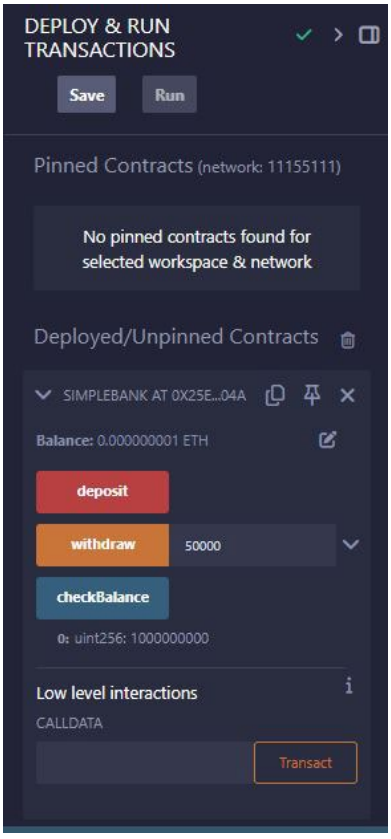

Sepolia
Account 1
0x1865C...c29D7
0.0394 SepoliaETH
+$0.00 (+0.00%)
Buy & Sell
Send
Swap
Bridge
Portfolio
Tokens
NFTs
Activity
Sep 9, 2024
Send
Confirmed
-0.01 SepoliaETH
-0.01 SepoliaETH
Receive
Confirmed
0.05 SepoliaETH
0.05 SepoliaETH
MetaMask support

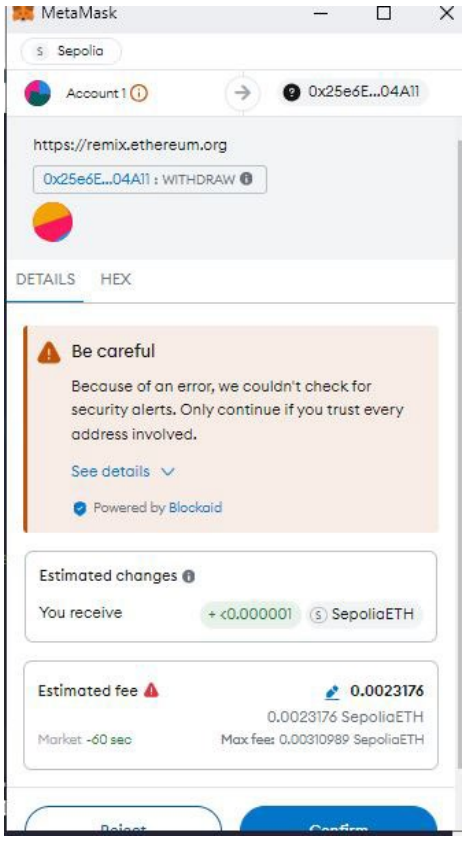

METAMASK
Sepolia
Account 1
0x1865C...c29D7
Edit
Sepolia
Account 1
Account2
DETAILS
HEX
Be careful
Because of an error, we couldn't check for security alerts. Only continue if you trust every address involved.
See details
Powered by Blockaid
Estimated changes
You send
- 0.01
SepoliaETH
Estimated fee
0.00058218
0.00058218 SepoliaETH
Market -60 sec
Max fee: 0.00077492 SepoliaETH
Reject
Confirm


Send a token
From
Account 1
0x1865C...c29D7
Sepoli...
0
Balance: 0.05
Max
To
Account2
0xF485A...6DAa0
Sepoli...
0
Cancel
Continue


METAMASK
Sepolia
Account 1
0x1865C...c29D7
0.05 SepoliaETH
Buy & Sell
Send
Swap
Bridge
Portfolio
Tokens
NFTs
Activity
SepoliaETH
0.05 SepoliaETH

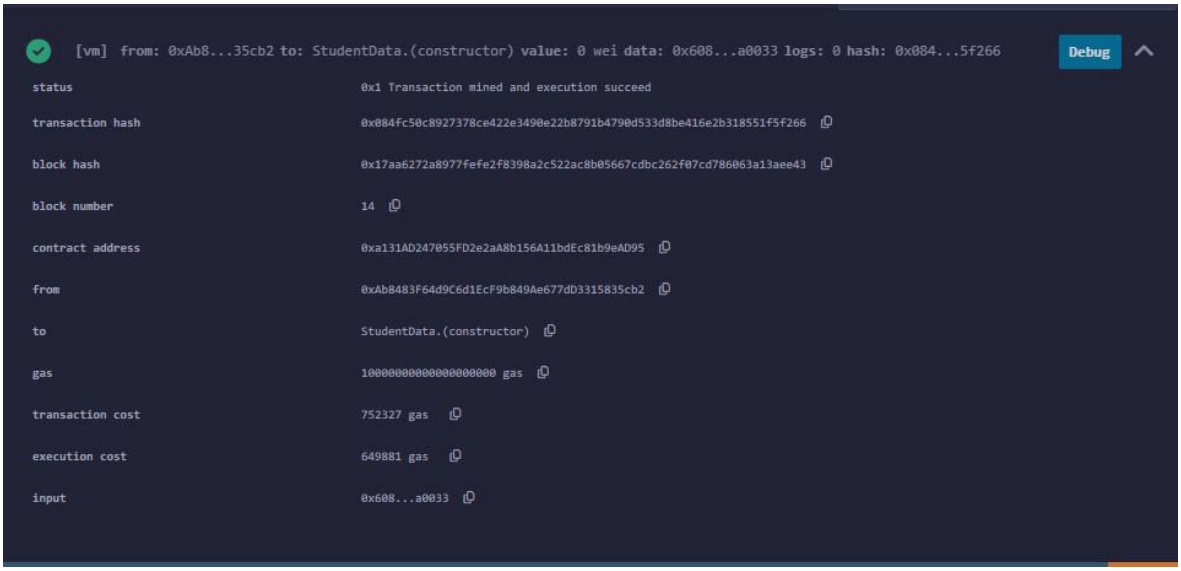
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract SimpleBank {
    // Mapping to store balances for each account (address)
    mapping(address => uint) private balances;

    // Event logs for deposit and withdrawal operations
    event Deposit(address indexed account, uint amount);
    event Withdraw(address indexed account, uint amount);

    // Deposit function to add money to the user's account
    function deposit() public payable {
        require(msg.value > 0, "You must deposit some money.");
        balances[msg.sender] += msg.value;

        // Emit the deposit event
        emit Deposit(msg.sender, msg.value);
    }

    // Withdraw function to withdraw money from the user's account
    function withdraw(uint amount) public {
        require(amount <= balances[msg.sender], "Insufficient balance.");
        payable(msg.sender).transfer(amount);  // Transfer the amount to the user's address
        balances[msg.sender] -= amount;

        // Emit the withdraw event
        emit Withdraw(msg.sender, amount);
    }

    // Function to check the balance of the user's account
    function checkBalance() public view returns (uint) {
        return balances[msg.sender];
    }
}
```

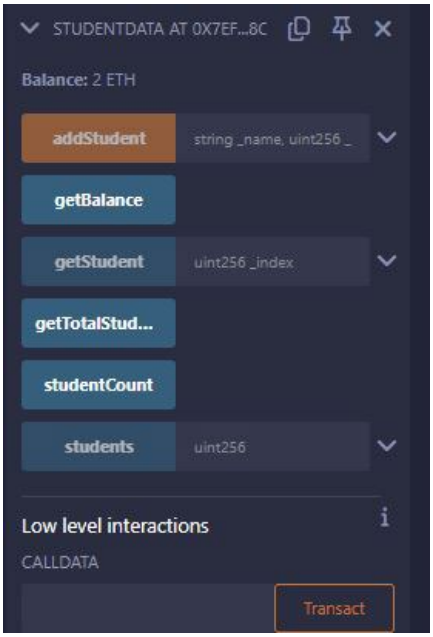




```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract StudentData {

    // Structure to hold student details
    struct Student {
        uint id;
        string name;
        uint age;
    }

    // Array to store list of students
    Student[] public students;

    // Count of students
    uint public studentCount;

    // Event to be emitted when a student is added
    event StudentAdded(uint id, string name, uint age);

    // Fallback function - It gets executed when no other function matches the call or
    // Ether is sent to the contract without calldata
    fallback() external payable {
        revert("Fallback triggered, operation not allowed");
    }

    // Receive Ether
    receive() external payable {
        revert("Receive function triggered, operation not allowed");
    }

    // Function to add a student to the array
    function addStudent(string memory _name, uint _age) public {
        studentCount++;
        students.push(Student(studentCount, _name, _age));
        emit StudentAdded(studentCount, _name, _age); // Emit event
    }

    // Function to get a student by index
    function getStudent(uint _index) public view returns (uint, string memory, uint) {
        require(_index < students.length, "Index out of bounds");
        Student memory s = students[_index];
        return (s.id, s.name, s.age);
    }

    // Function to get total number of students
    function getTotalStudents() public view returns (uint) {
        return studentCount;
    }

    // Function to get contract balance (if any Ether is sent)
    function getBalance() public view returns (uint) {
        return address(this).balance;
    }
}
```


block hash 0x047218c1ba5733ac6b1d45e1c59382fad3cd9d98cbf7ac7853e5ef2eae38fe91
block number 2
from 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to StudentData.addStudent(string,uint256) 0xd9145CCE52D386f254917e481eB44e9943F39138
gas 157710 gas
transaction cost 137139 gas
execution cost 115491 gas
input 0x0e5...00000
output 0x
decoded input
{
"string _name": "Ram",
"uint256 _age": "22"
}
decoded output
{}


[vm] from: 0xAb8...35cb2 to: StudentData.(constructor) value: 0 wei data: 0x608...a0033 logs: 0 hash: 0x084...5f266
Debug
status 0x1 Transaction mined and execution succeed
transaction hash 0x084fc50c8927378ce422e3490e22b8791b4790d533d8be416e2b318551f5f266
block hash 0x17aa6272a8977fefe2f8398a2c522ac8b05667cdbc262f07cd786063a13aee43
block number 14
contract address 0xa131AD247055FD2e2aA8b156A11bdEc81b9eAD95
from 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2
to StudentData.(constructor)
gas 10000000000000000000 gas
transaction cost 752327 gas
execution cost 649881 gas
input 0x608...a0033


STUDENTDATA AT 0X7EF...8C
Balance: 2 ETH
addStudent
string _name, uint256 _
getBalance
getStudent
uint256 _index
getTotalStud...
studentCount
students
uint256
Low level interactions
CALLDATA
Transact