CCMP 606 - Orchestration of Cloud Resources

Assignment 1 Report

Instructor: **Yongchang He**

Student: **Hai Nam Nguyen – 000520322**

2023/2024 Winter Semester

Saskatchewan Polytechnic

Submitted:

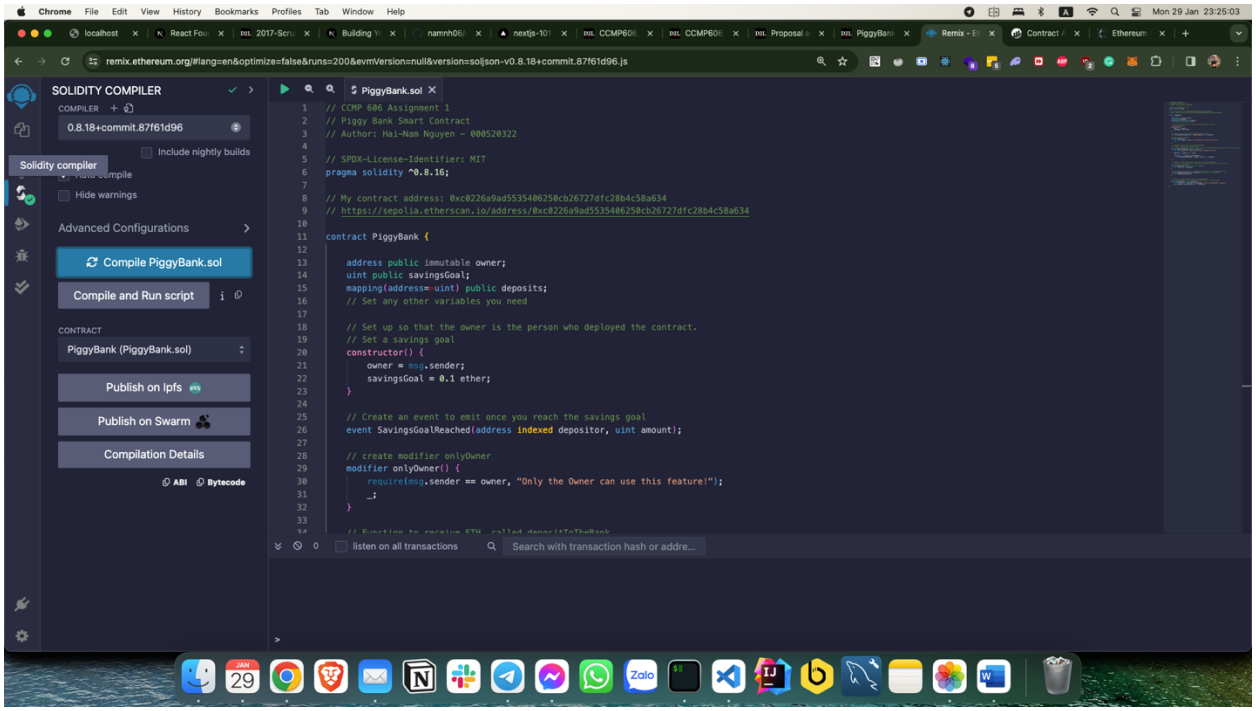
January 29th, 2024

Table of Contents

I. Compile PiggyBank.sol

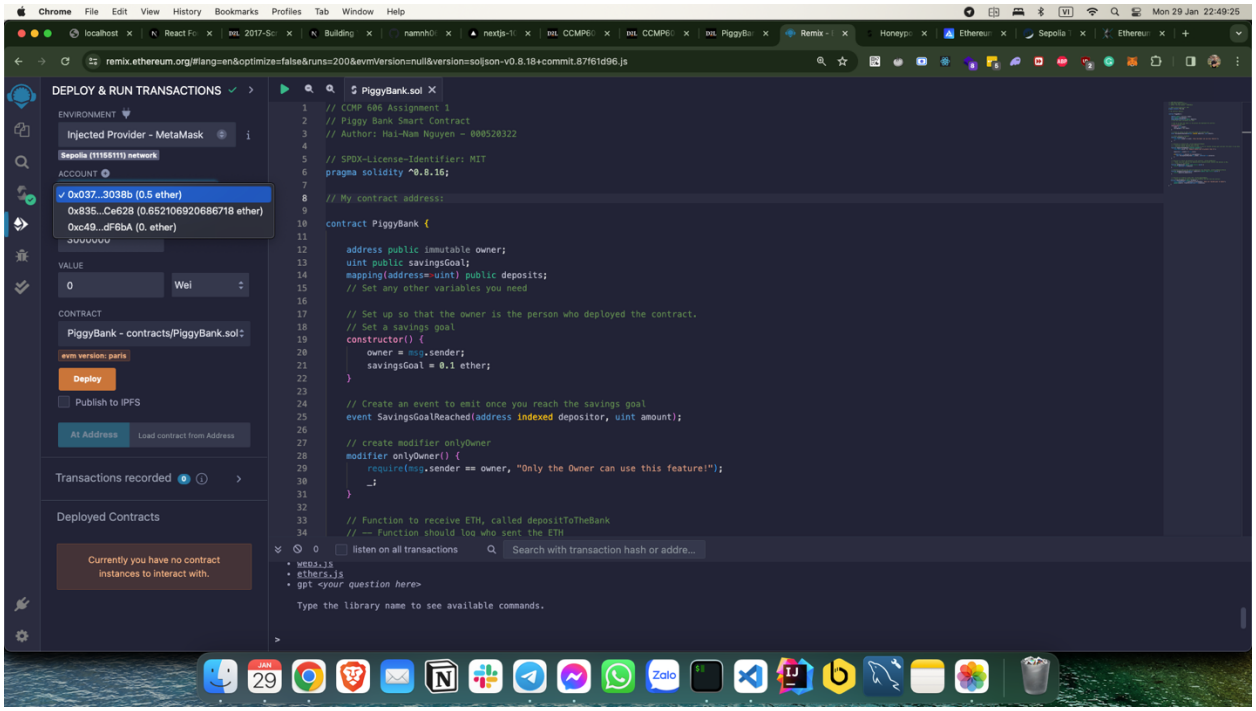
The screenshot(s) show PiggyBank.sol has no compile errors on Remix IDE. (0.5 points)

Compile:

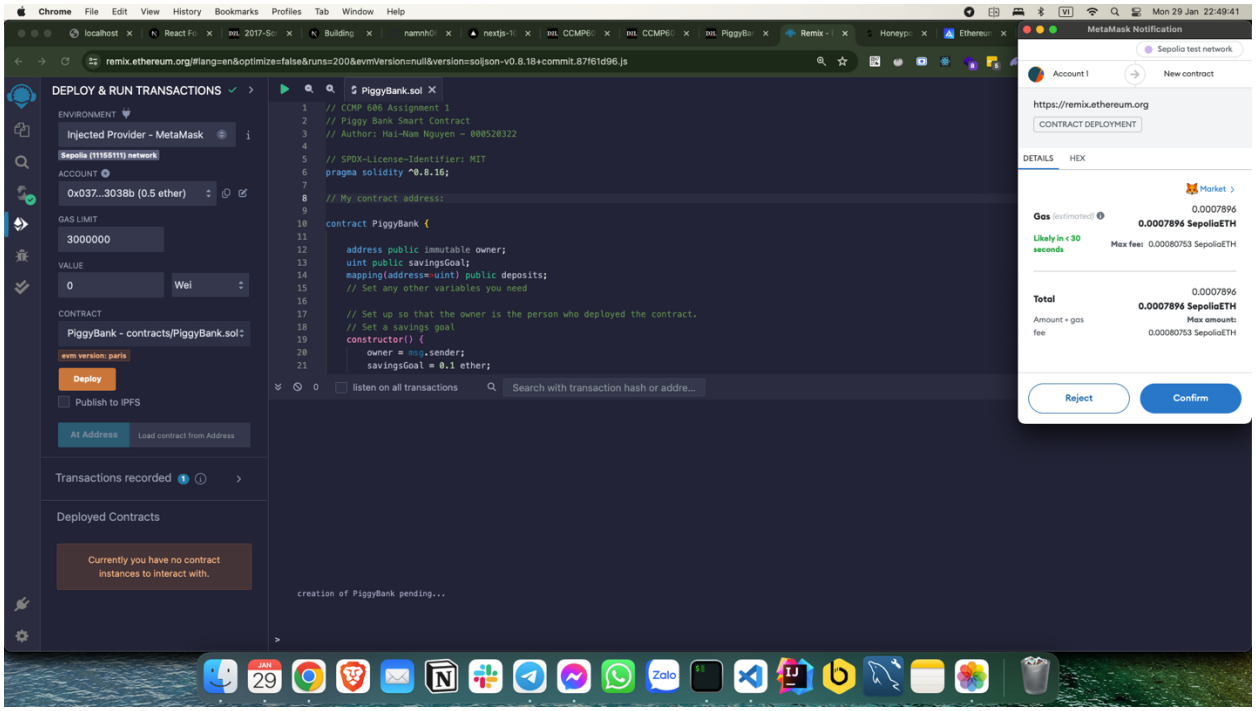


Deploy:

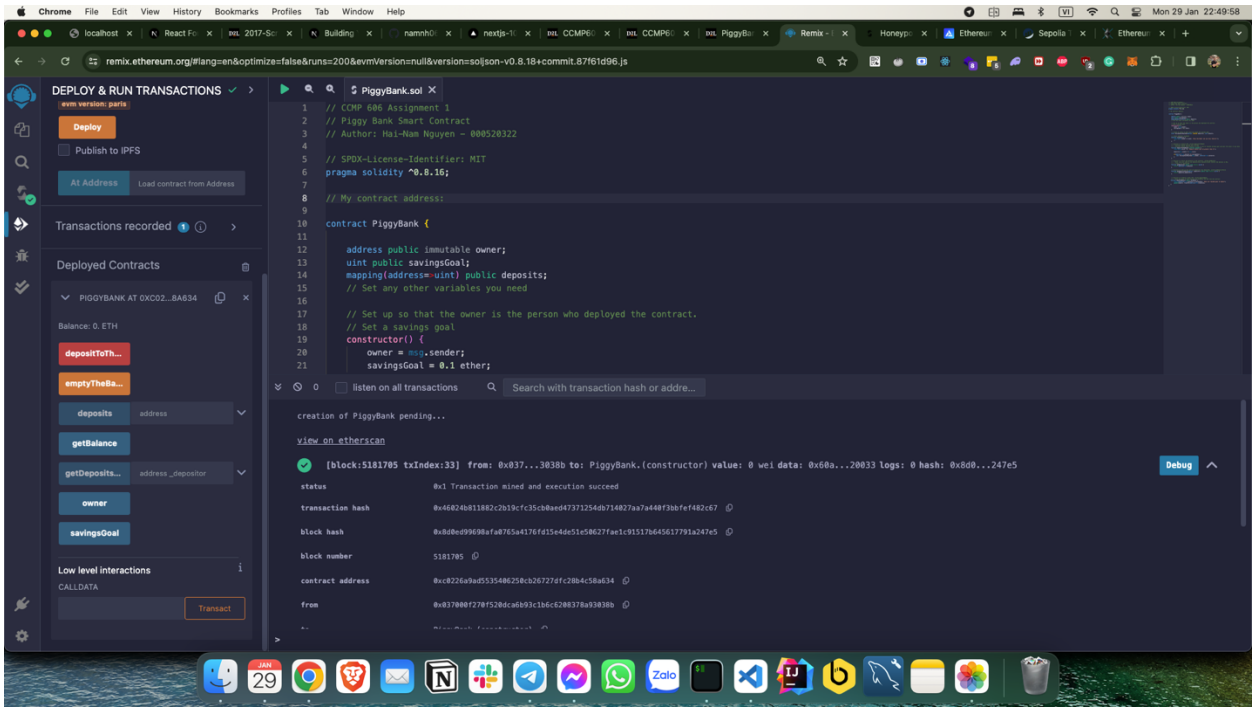
First, check the Injected Provider – MetaMask account



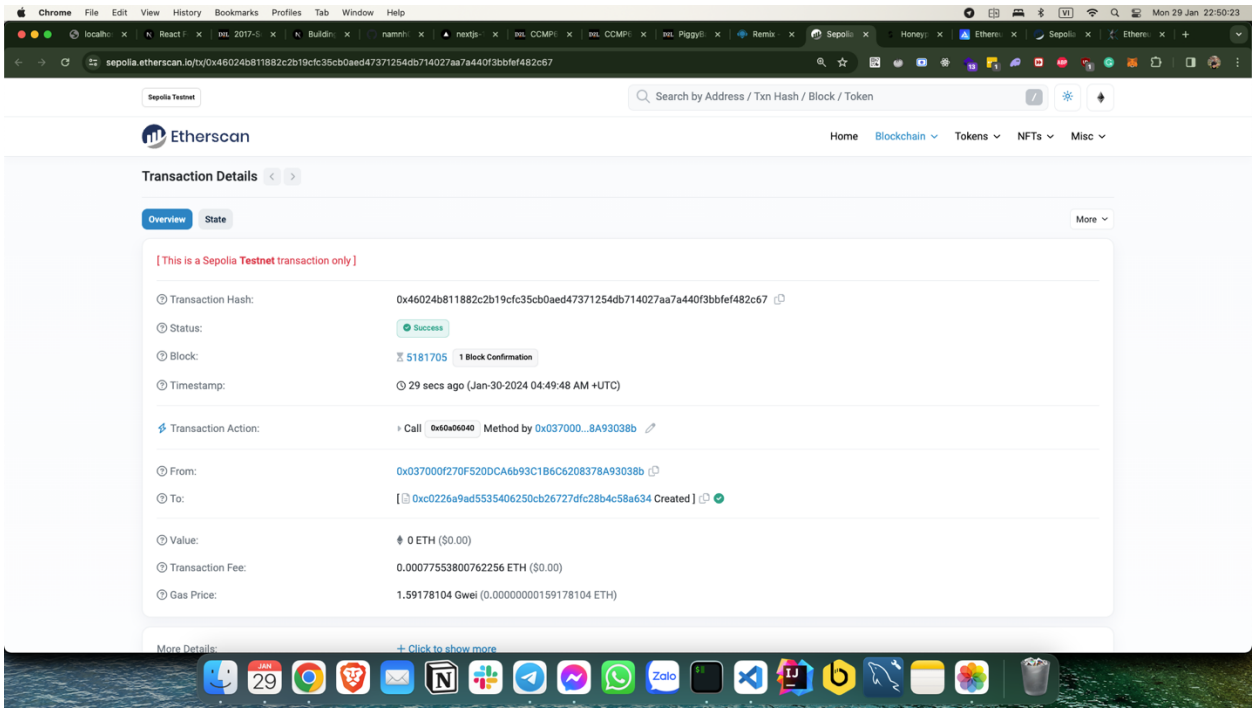
Secondly, click Deploying:



Lastly, Confirm to see the deploy success:

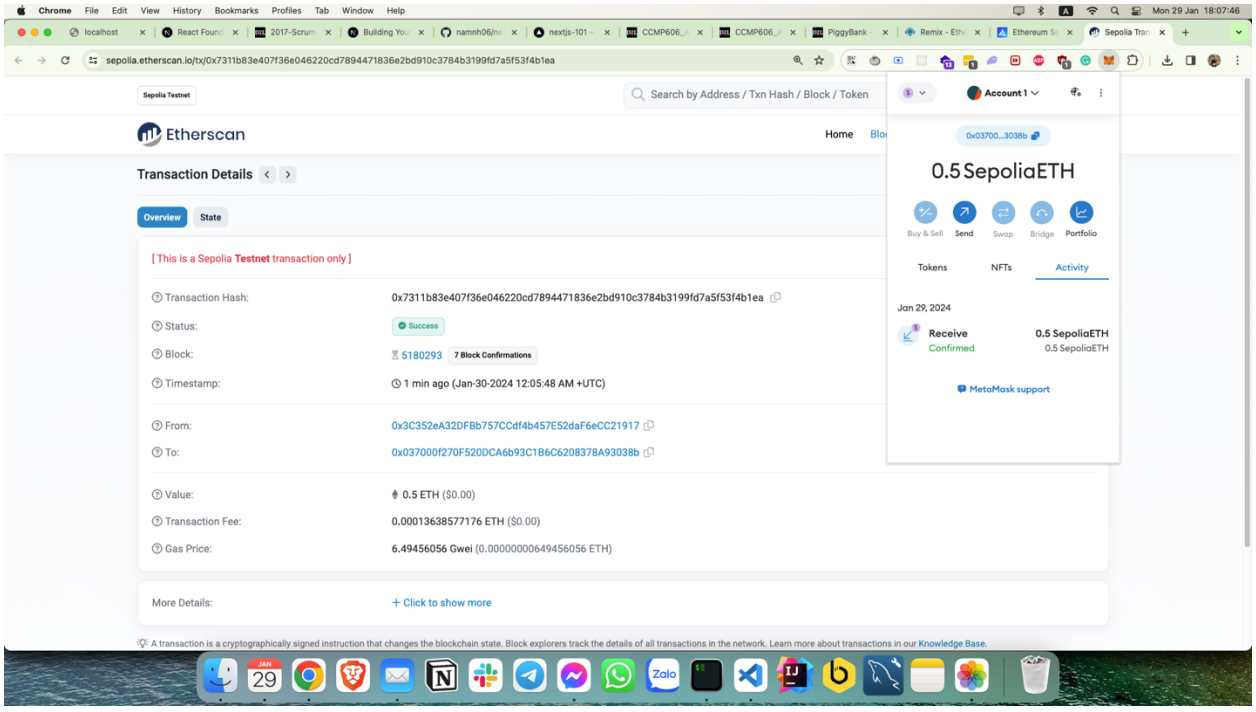


Double check on Sepolia etherscan:



II. Sepolia ETH on MetaMask

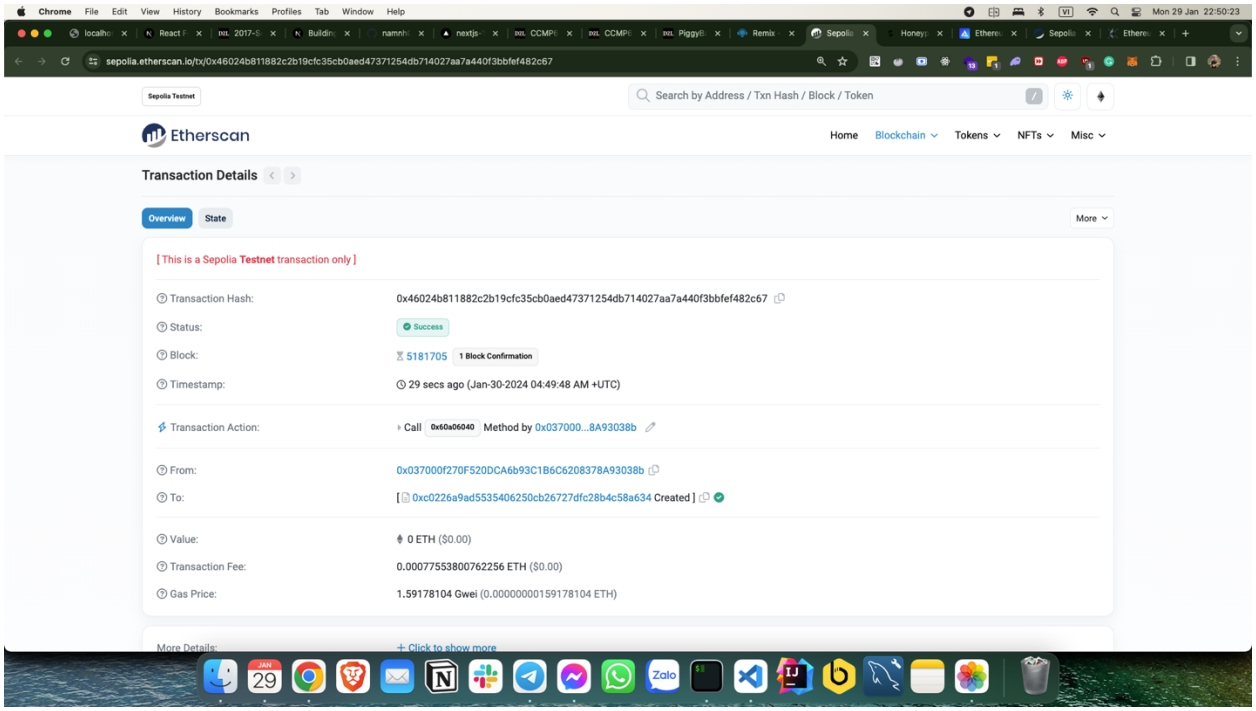
Get faucet from https://sepoliafaucet.com/



III. Smart Contract Address on the Etherscan Webpage

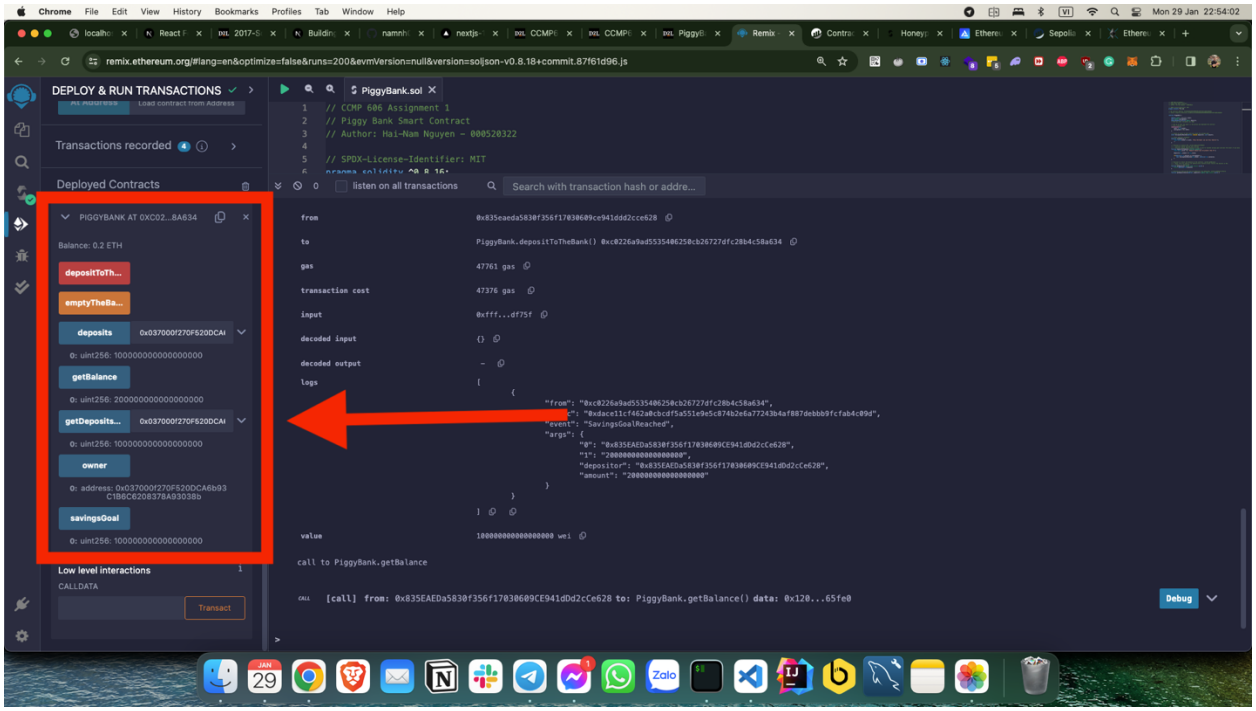
The address:

https://sepolia.etherscan.io/address/0xc0226a9ad5535406250cb26727dfc28b4c58a634

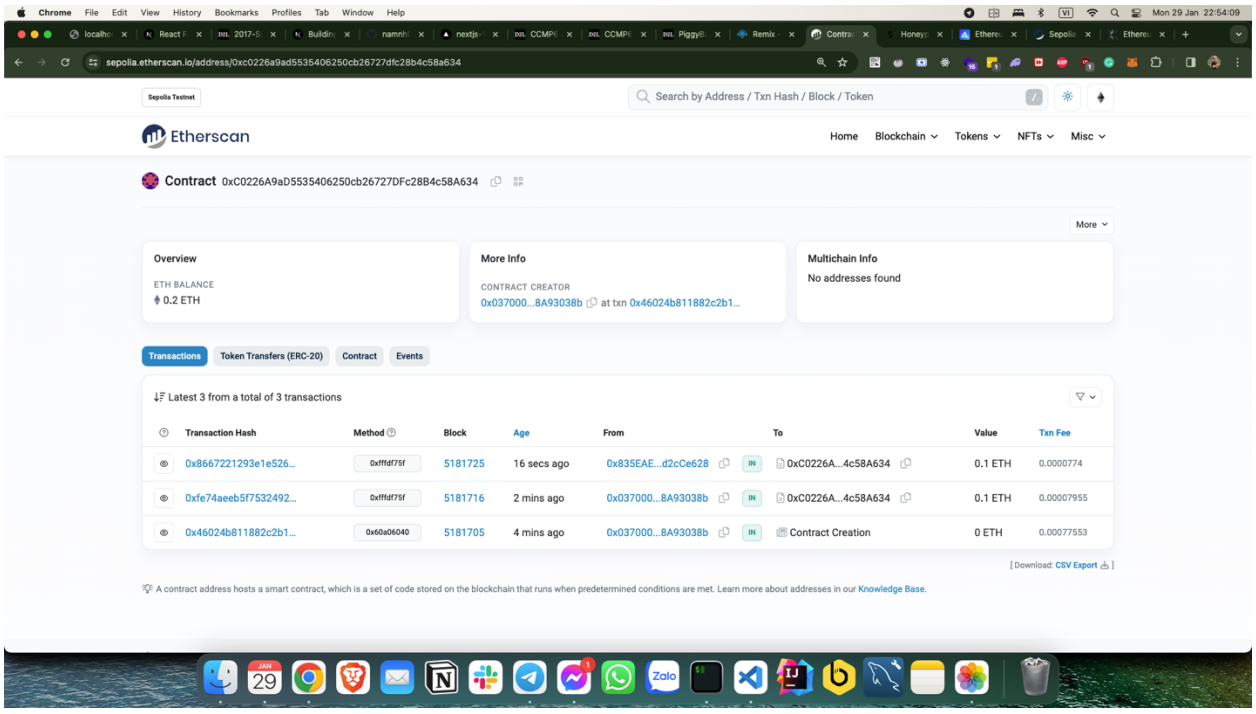


IV. Contract Balance on Remix IDE

After deposit:



Also on Sepolia Etherscan:



V. Explanations and Show How the Functions Work

a. Function depositToTheBank

```
34    // Function to receive ETH, called depositToTheBank
35    // -- Function should log who sent the ETH
36    // -- Function should check balance to know if you've reached savings goal and emit the event if you have.
37    function depositToTheBank() external payable {
38        require(msg.value > 0, "Deposit amount must be greater than 0!");
39
40        deposits[msg.sender] += msg.value;
41
42        if(address(this).balance >= savingsGoal) {
43            emit SavingsGoalReached(msg.sender, address(this).balance);
44        }
45    }
```

Overview, this function check how much the ether will be sent to the contract, it must be greater than 0 otherwise the error log will display. After require check, the mapping deposits will store the address of who send ether and cumulative the value he/she sent. The last one is the If condition to check whether the balance of this contract is equal or larger than savingsGoal or not. If yes, the event SavingsGoalReached will be emitted.

b. Function getBalance

```
47    // Function to return the balance of the contract, called getBalance
48    // -- Note: you will need to use address(this).balance which returns the balance in Wei.
49    // -- 1 Eth = 1 * 10**18 Wei
50    function getBalance() public view returns (uint) {
51        return address(this).balance;
52    }
```

This function can be called by anyone and return the balance of this contract in wei.

c. Function getDepositsValue

```
54    // Function to look up how much any depositor has deposited, called getDepositsValue
55    function getDepositsValue(address _depositor) public view returns (uint) {
56        return deposits[_depositor];
57    }
```

This function receives an address and return the wei that this address sent to the contract if exists.

d. Function emptyTheBank

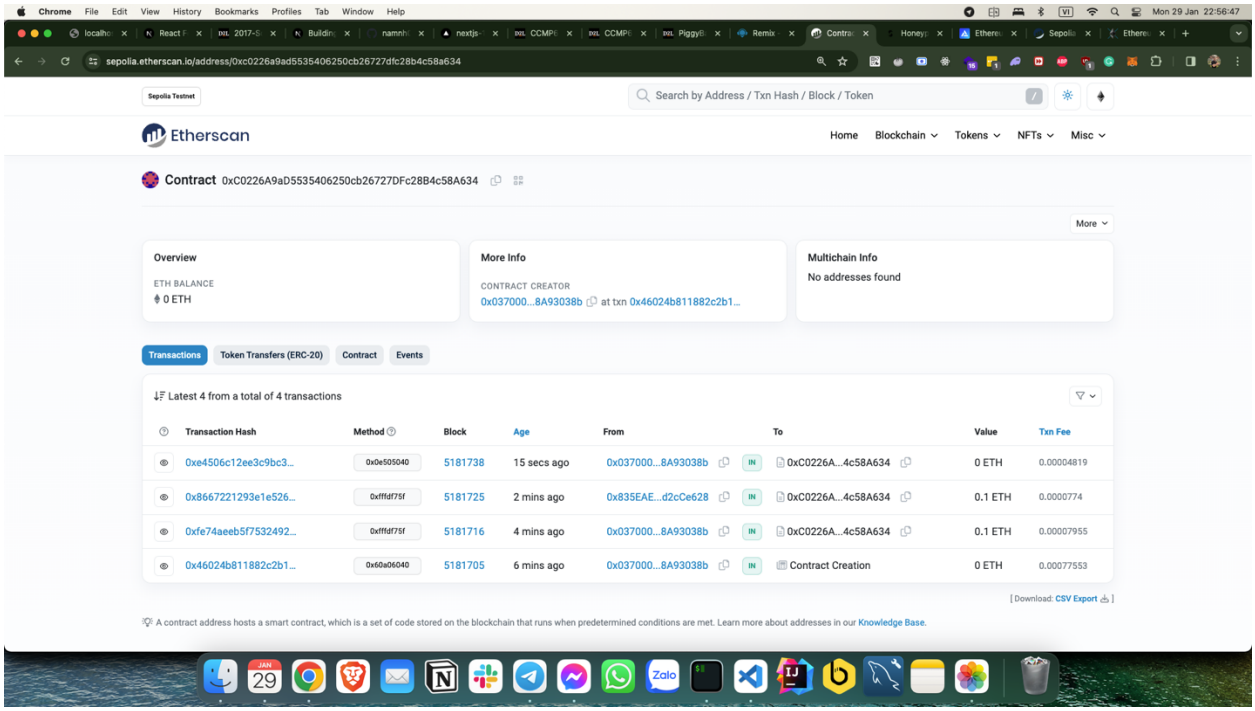
```
60      // Function to withdraw (send) ETH, called emptyTheBank
61      // -- Only the owner of the contract can withdraw the ETH from the contract
62      function emptyTheBank() public onlyOwner {
63          require(address(this).balance >= savingsGoal, "Have not reached goal to empty");
64          payable(owner). transfer(address(this).balance);
65      }
66  }
```

This function stricted by onlyOwner can call.

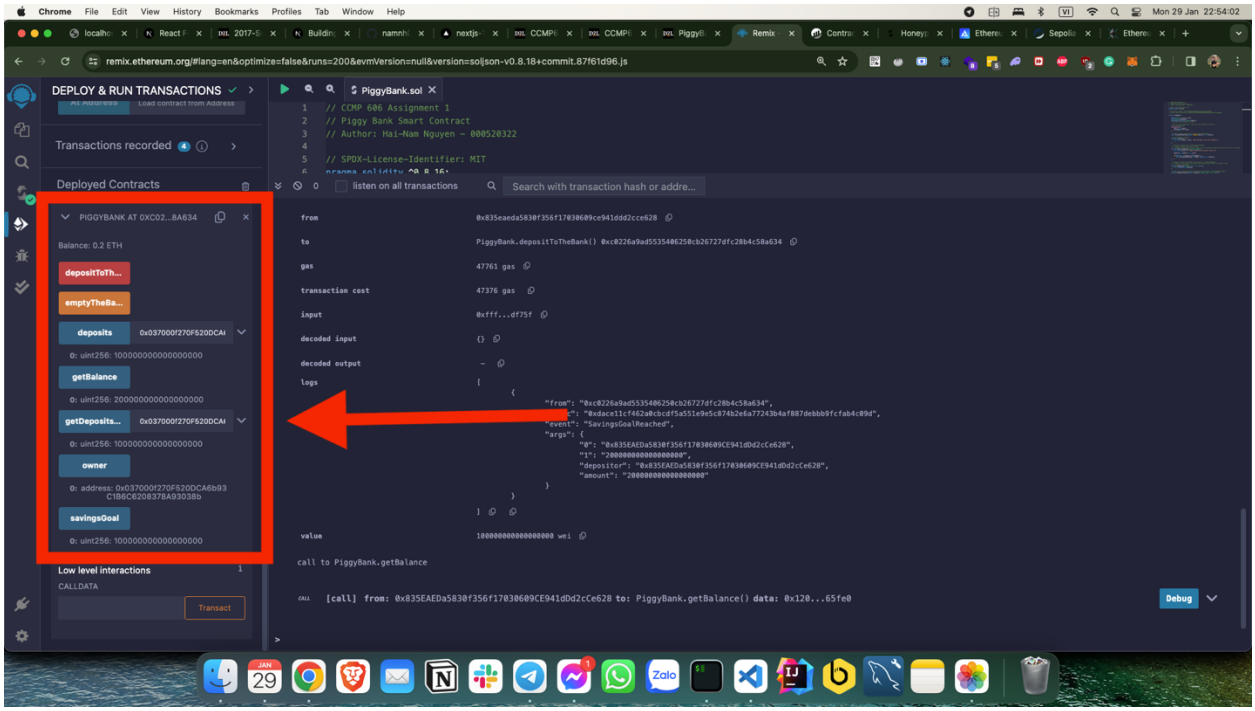
```
28      // create modifier onlyOwner
29      modifier onlyOwner() {
30          require(msg.sender == owner, "Only the Owner can use this feature!");
31          _;
32      }
```

It checks whether the balance of this contract is greater or equal the savingsGoal. If the balance matched condition, the ether in this contract will be transferred to the owner address.

After this function invoked, the contract address looks like this:



Functions worked:


DEPLOY & RUN TRANSACTIONS
Transactions recorded
Deployed Contracts
PIGGYBANK AT 0XC02...8A634
Balance: 0.2 ETH
depositToTh...
emptyTheBa...
deposits
0: uint256: 100000000000000000
getBalance
0: uint256: 200000000000000000
getDeposits...
0: uint256: 100000000000000000
owner
savingsGoal
0: uint256: 100000000000000000
Low level interactions
CALLDATA
Transact
PiggyBank.sol
// CCMP 606 Assignment 1
// Piggy Bank Smart Contract
// Author: Hai-Nam Nguyen - 000520322
// SPDX-License-Identifier: MIT
listen on all transactions
Search with transaction hash or addre...
from
to
gas
transaction cost
input
decoded input
decoded output
logs
value
call to PiggyBank.getBalance
Debug