

SavingsVault | Simple ETH Smart Contract

Solidity Blockchain Dev

Student name	Email	Github repo link
Guilherme Gomes de Andrade	andrade2002g@gmail.com	https://github.com/adr-g/solidity-smart-contract-assignment

Content

1	Application Description.....	3
2	Implementation	3
2.1	List of Functions	3
2.2	Events & Validations	4
3	Testing.....	4

1 Application Description

This project consists of a very simple smart contract called SavingsVault.

The idea is straightforward: the contract allows ETH to be deposited and later withdrawn. Any user can deposit ETH, but only the contract owner (defined at deployment) can withdraw the funds.

The goal of this project is not to build a complex application, but to demonstrate the basic use of Solidity and how to deploy and interact with a smart contract on the Ethereum blockchain.

2 Implementation

2.1 List of Functions

constructor()

- Function name: constructor
- Purpose: Sets the address that deploys the contract as the owner.
- Input Parameters: None
- Return Type: None
- Access modifier: n/a

deposit()

- Function name: deposit
- Purpose: Allows ETH to be sent to the contract.
- Input Parameters: None (ETH is sent using msg.value)
- Return Type: None
- Access modifier: external payable

withdrawAll()

- Function name: withdrawAll
- Purpose: Allows the owner to withdraw all ETH stored in the contract.
- Input Parameters: None
- Return Type: None
- Access modifier: external

getVaultBalance()

- Function name: getVaultBalance
- Purpose: Returns the current balance of the contract.
- Input Parameters: None
- Return Type: uint256
- Access modifier: external view

2.2 Events & Validations

Events

- FundsDeposited
Emitted when ETH is deposited into the contract.
- FundsWithdrawn
Emitted when ETH is withdrawn by the owner.

Validations

The require() function is used to keep the contract logic simple and safe:

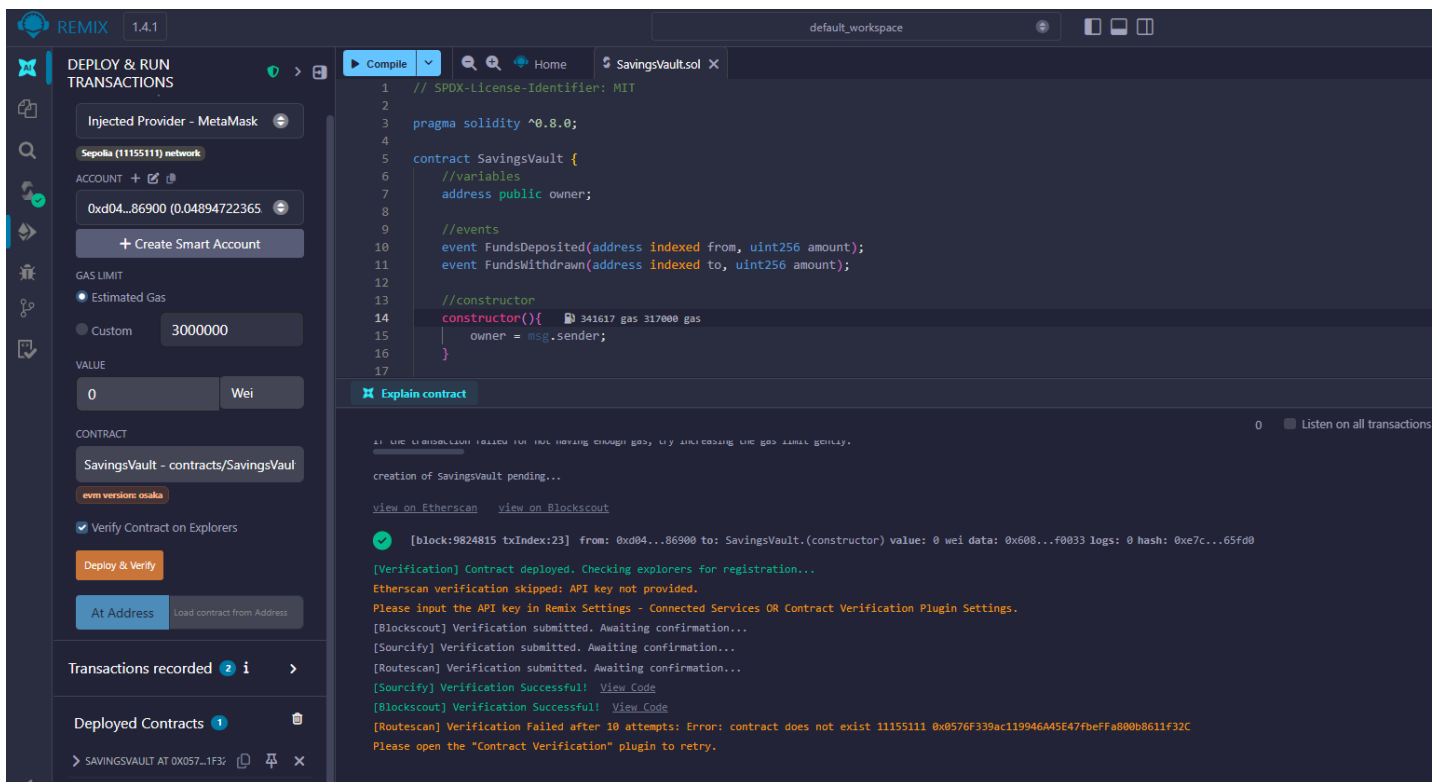
- To avoid depositing zero ETH;
- To make sure only the owner can withdraw from the contract;
- To avoid withdrawing when there are no funds available.

3 Testing

The contract was tested using the Remix IDE and deployed on the Sepolia test network with MetaMask.

Testing included:

- Compiling and deploying the contract;



Img 1 - 01_deploy_savingsvault_sepolia

- Calling the deposit() function with test ETH;

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, showing 'Estimated Gas' at 3000000. The 'CONTRACT' dropdown is set to 'SavingsVault - contracts/SavingsVault'. The 'Deploy & Verify' button is visible. The main editor displays the Solidity code for SavingsVault.sol, which includes a constructor that sets the owner to msg.sender. The bottom panel shows the transaction history, including a successful deposit transaction.

```

1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4
5 contract SavingsVault {
6     //variables
7     address public owner;
8
9     //events
10    event FundsDeposited(address indexed from, uint256 amount);
11    event FundsWithdrawn(address indexed to, uint256 amount);
12
13    //constructor
14    constructor(){
15        owner = msg.sender;
16    }

```

Transaction history:

- transact to SavingsVault.deposit errored: error occurred: insufficient funds (transaction= { "data": "0x0e38d00", "from": "0x043e37f85a06501f3598622d6f34df59b486900", "maxFeePerGas": "0x132afdd0", "to": "0x0576f339ac11994a45e47bffa00b0611f32c", "type": "0x2", "value": "0x00" })
- Insufficient funds (transaction= { "data": "0x0e38d00", "from": "0x043e37f85a06501f3598622d6f34df59b486900", "maxFeePerGas": "0x132afdd0", "to": "0x0576f339ac11994a45e47bffa00b0611f32c", "type": "0x2", "value": "0x00" })
- If the transaction failed for not having enough gas, try increasing the gas limit gently.
- call to SavingsVault.getVaultBalance
- CALL [call] from: 0x043e37f85a06501f3598622d6f34df59b486900 to: SavingsVault.getVaultBalance() data: 0xed1...2e8f
- call to SavingsVault.owner
- CALL [call] from: 0x043e37f85a06501f3598622d6f34df59b486900 to: SavingsVault.owner() data: 0x8da...5cb5b
- transact to SavingsVault.deposit pending ...
- view on Etherscan view on Blockscout
- [block:9824848 txIndex:15] from: 0x04...86900 to: SavingsVault.deposit() 0x857...1f32c value: 100 wei data: 0xd0e...30db0 logs: 1 hash: 0x0ba...ca2fe

Img 2 - 02_deposit_savingsvault_sepolia

- Calling the withdrawAll() function;

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, showing 'Estimated Gas' at 3000000. The 'CONTRACT' dropdown is set to 'SavingsVault - contracts/SavingsVault'. The 'Deploy & Verify' button is visible. The main editor displays the Solidity code for SavingsVault.sol, which includes a constructor that sets the owner to msg.sender. The bottom panel shows the transaction history, including a successful withdrawAll transaction.

```

1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4
5 contract SavingsVault {
6     //variables
7     address public owner;
8
9     //events
10    event FundsDeposited(address indexed from, uint256 amount);
11    event FundsWithdrawn(address indexed to, uint256 amount);
12
13    //constructor
14    constructor(){
15        owner = msg.sender;
16    }

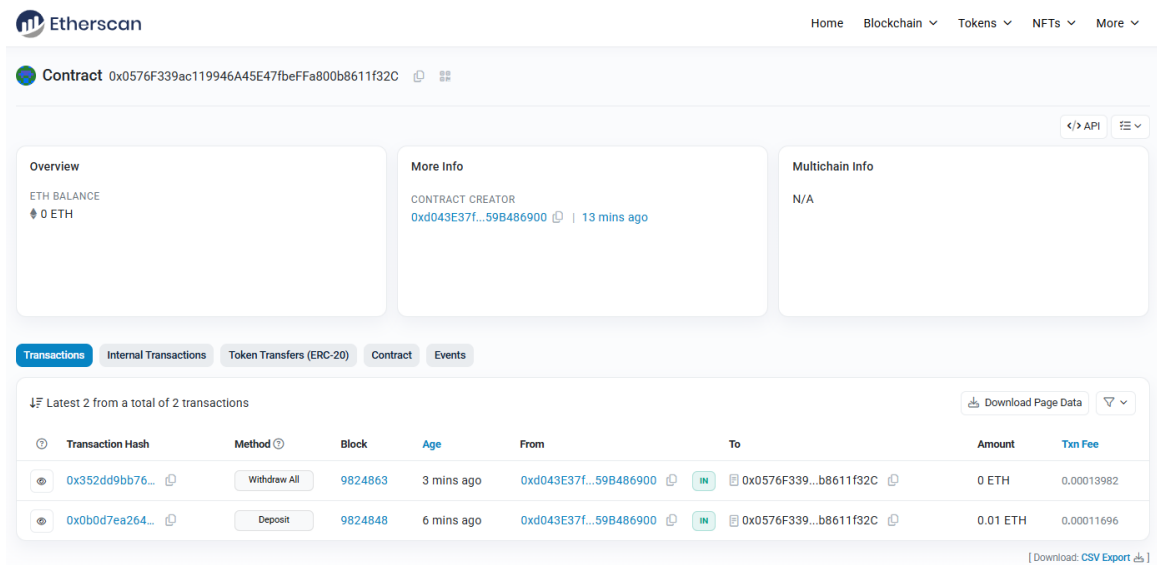
```

Transaction history:

- transact to SavingsVault.deposit pending ...
- view on Etherscan view on Blockscout
- [block:9824848 txIndex:15] from: 0x04...86900 to: SavingsVault.deposit() 0x857...1f32c value: 100 wei data: 0xd0e...30db0 logs: 1 hash: 0x0ba...ca2fe
- call to SavingsVault.getVaultBalance
- CALL [call] from: 0x043e37f85a06501f3598622d6f34df59b486900 to: SavingsVault.getVaultBalance() data: 0xed1...2e8f
- transact to SavingsVault.withdrawAll pending ...
- view on Etherscan view on Blockscout
- [block:9824863 txIndex:14] from: 0x04...86900 to: SavingsVault.withdrawAll() 0x857...1f32c value: 0 wei data: 0x853...828b6 logs: 1 hash: 0xec3...00ba8
- call to SavingsVault.getVaultBalance
- CALL [call] from: 0x043e37f85a06501f3598622d6f34df59b486900 to: SavingsVault.getVaultBalance() data: 0xed1...2e8f

Img 3 - 03_withdrawAll_savingsvault_sepolia

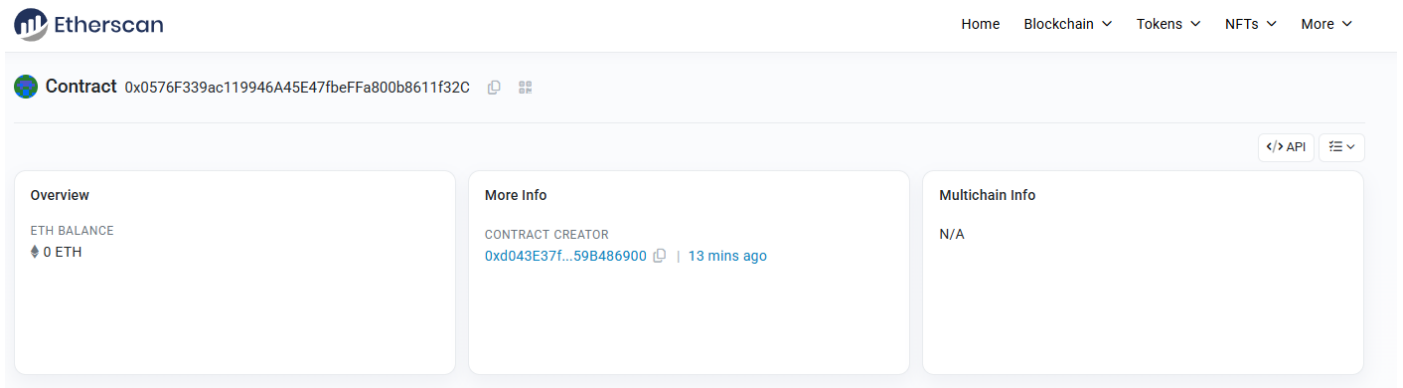
- Verifying transactions on Sepolia Etherscan.



The screenshot shows the Etherscan interface for a contract. The top navigation bar includes links for Home, Blockchain, Tokens, NFTs, and More. The contract address is 0x0576F339ac119946A45E47fBeFFa800b8611f32C. The Overview section shows an ETH balance of 0. The More Info section shows the contract creator as 0xd043E37f...59B486900, created 13 minutes ago. The Transactions section is active, showing a table of the latest 2 transactions.

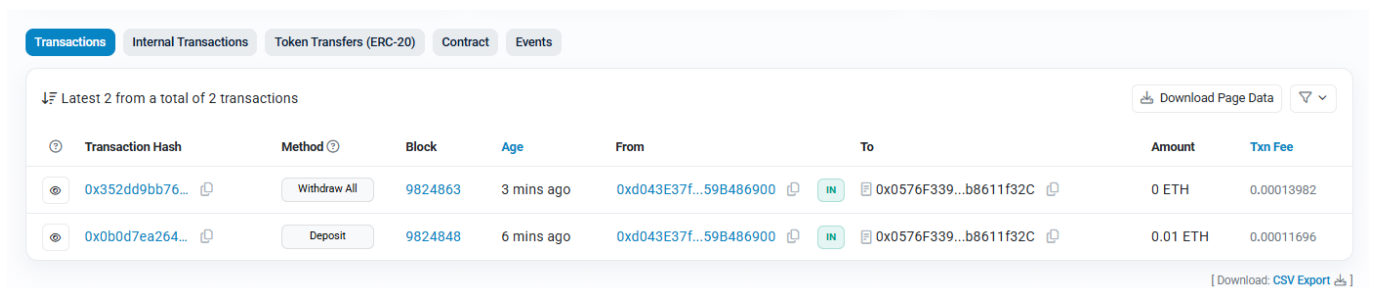
Transaction Hash	Method	Block	Age	From	To	Amount	Txn Fee
0x352dd9bb76...	Withdraw All	9824863	3 mins ago	0xd043E37f...59B486900	0x0576F339...b8611f32C	0 ETH	0.00013982
0x0b0d7ea264...	Deposit	9824848	6 mins ago	0xd043E37f...59B486900	0x0576F339...b8611f32C	0.01 ETH	0.00011696

Img 4 - 04_etherscan_savingsvault_sepolia



This screenshot shows the top portion of the Etherscan contract page, including the navigation bar, contract address, and the Overview, More Info, and Multichain Info sections. The Overview section shows an ETH balance of 0. The More Info section shows the contract creator as 0xd043E37f...59B486900, created 13 minutes ago. The Multichain Info section shows N/A.

Img 5 - 05_etherscan_contract_creation



This screenshot shows the Transactions section of the Etherscan contract page. It displays a table of the latest 2 transactions.

Transaction Hash	Method	Block	Age	From	To	Amount	Txn Fee
0x352dd9bb76...	Withdraw All	9824863	3 mins ago	0xd043E37f...59B486900	0x0576F339...b8611f32C	0 ETH	0.00013982
0x0b0d7ea264...	Deposit	9824848	6 mins ago	0xd043E37f...59B486900	0x0576F339...b8611f32C	0.01 ETH	0.00011696

Img 6 - 06_etherscan_transactions