BLOCK CHAIN ASSIGNEMNT

1. Introduction

This report documents the process of deploying a simple smart contract on the Ethereum Sepolia test network. The smart contract, **SimpleStorage**, allows users to store and retrieve a number. The deployment was conducted using Remix IDE, MetaMask, and Sepolia ETH for transaction fees.

2. GitHub Repository

The smart contract code, deployment steps, and related information are available in the following GitHub repository: [GitHub Repository Link](The smart contract code, deployment steps, and related information are available in the following GitHub repository: GitHub Repository Link)

3. Steps Taken

3.1 Writing the Smart Contract

A Solidity smart contract was written in Remix IDE. The contract includes:

- A `storedNumber` variable to hold a numeric value.
- A `setNumber` function to update the stored value.
- A `getNumber` function to retrieve the value.
- An event `NumberUpdated` to log updates.

3.2 Compiling the Smart Contract

- Used **Remix IDE** to write and compile the contract.

- Selected Solidity version **0.8.19**.
- Ensured no compilation errors before proceeding.

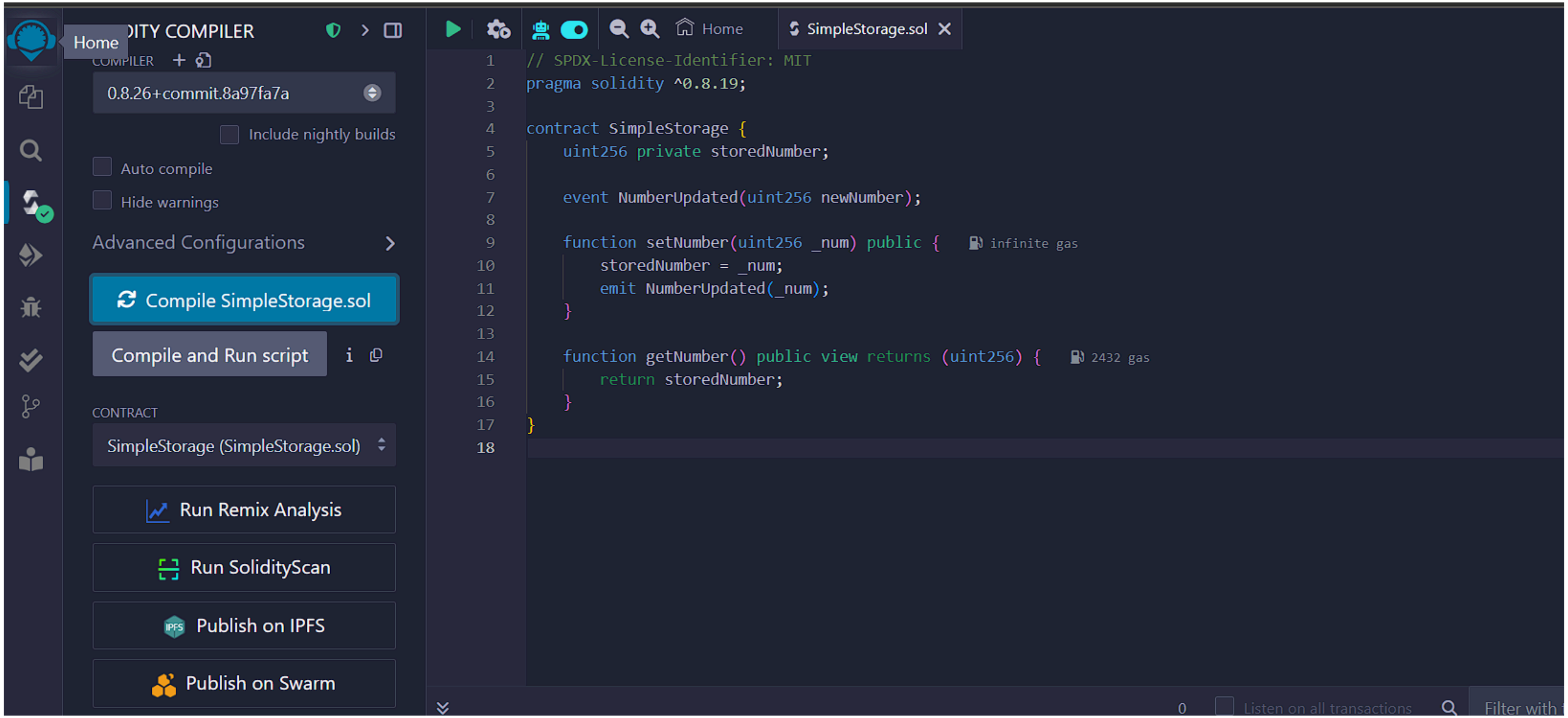
3.3 Deploying on Sepolia Testnet

- Connected **MetaMask** to the Sepolia test network.
- Used **Ethereum Sepolia Faucet** to obtain **0.03 Sepolia ETH**.
- Selected **Injected Provider - MetaMask** in Remix.
- Deployed the contract, confirmed the transaction, and waited for completion.

4. Screenshots

The following screenshots provide visual proof of the deployment process:

1. **Remix IDE Interface** – Shows the smart contract code and deployment panel.



0.8.26+commit.8a97fa7a

□ Include nightly builds

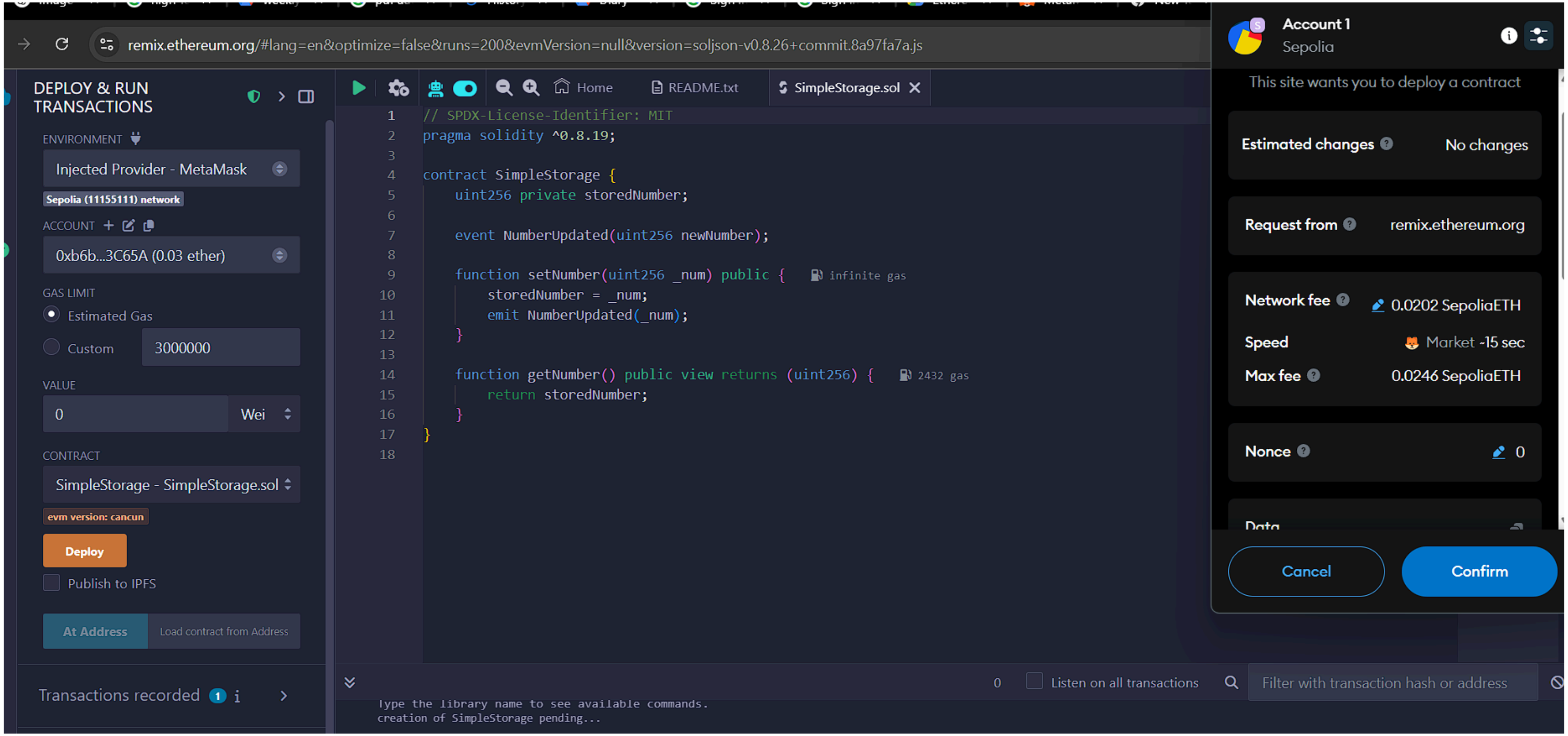
□ Auto compile

□ Hide warnings

SimpleStorage (SimpleStorage.sol)

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;

contract SimpleStorage {
    uint256 private storedNumber;

    event NumberUpdated(uint256 newNumber);

    function setNumber(uint256 _num) public {
        storedNumber = _num;
        emit NumberUpdated(_num);
    }

    function getNumber() public view returns (uint256) {
        return storedNumber;
    }
}
```

2. **MetaMask Transaction Request** – Displays the gas fee and confirmation request before deploying.



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.26+commit.8a97fa7a.js

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Provider - MetaMask

Sepolia (11155111) network

ACCOUNT: 0xb6b...3C65A (0.03 ether)

GAS LIMIT: ● Estimated Gas ○ Custom 3000000

VALUE: 0 Wei

CONTRACT: SimpleStorage - SimpleStorage.sol

evm version: cancun

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;

contract SimpleStorage {
    uint256 private storedNumber;

    event NumberUpdated(uint256 newNumber);

    function setNumber(uint256 _num) public {    infinite gas
        storedNumber = _num;
        emit NumberUpdated(_num);
    }

    function getNumber() public view returns (uint256) {    2432 gas
        return storedNumber;
    }
}

```

0
□ Listen on all transactions
Filter with transaction hash or address

```
Type the library name to see available commands.
creation of SimpleStorage pending...
```

Account 1
Sepolia

This site wants you to deploy a contract

Estimated changes: No changes

Request from: remix.ethereum.org

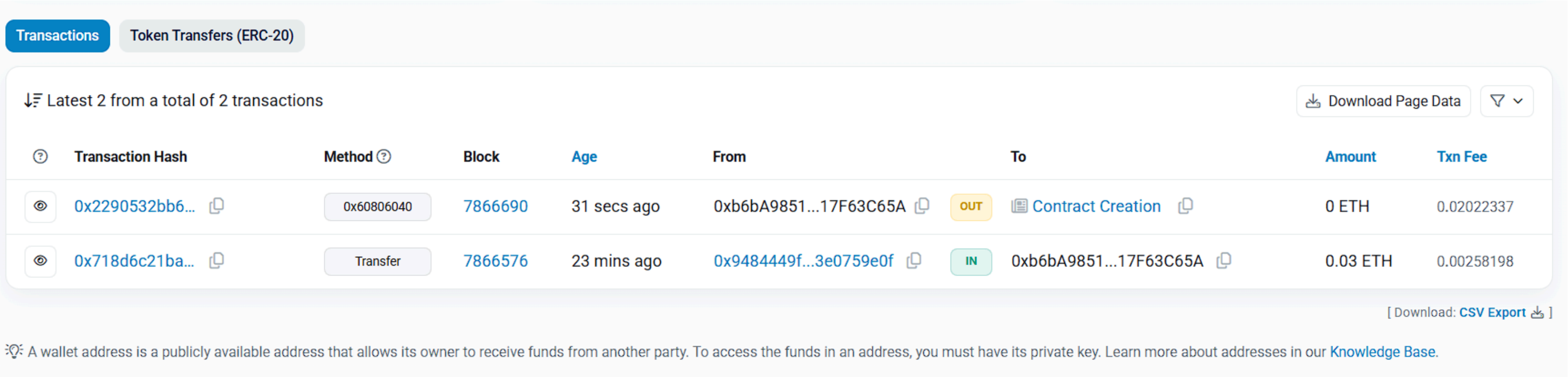
Network fee: 0.0202 SepoliaETH

Speed: Market -15 sec

Max fee: 0.0246 SepoliaETH

Nonce: 0

3. **Successful Transaction on Sepolia Etherscan** – Confirms that the transaction was mined successfully.



Latest 2 from a total of 2 transactions

	Transaction Hash	Method	Block	Age	From		To	Amount	Txn Fee
	0x2290532bb6...	0x60806040	7866690	31 secs ago	0xb6bA9851...17F63C65A	OUT	Contract Creation	0 ETH	0.02022337
	0x718d6c21ba...	Transfer	7866576	23 mins ago	0x9484449f...3e0759e0f	IN	0xb6bA9851...17F63C65A	0.03 ETH	0.00258198

[Download: CSV Export]

A wallet address is a publicly available address that allows its owner to receive funds from another party. To access the funds in an address, you must have its private key. Learn more about addresses in our Knowledge Base.

5. Conclusion

The deployment of the **SimpleStorage** smart contract was successful. The process involved writing, compiling, and deploying the contract on the **Ethereum Sepolia Testnet** using Remix IDE and MetaMask. The contract is now operational and can be interacted with via Sepolia Etherscan.

This report serves as documentation for the assignment and provides proof of completion through GitHub and transaction verification.