**Team number and name**: **#1, Satoshi**  
  
**Task 1 - Document Signing (group)**  
  
Answer (insert code from Remix):  
  
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.6;

contract DocumentSigning {

address public owner;

mapping(address => bool) public whitelist;

mapping(address => bool) public signatures;

uint256 public signatureCount;

event DocumentSigned(address indexed signer);

modifier onlyOwner() {

require(msg.sender == owner, "Not the contract owner");

\_;

}

modifier onlyWhitelisted(address \_address) {

require(whitelist[\_address], "Address not whitelisted");

\_;

}

constructor() {

owner = msg.sender;

}

// Add an address to the whitelist

function addToWhitelist(address \_address) external onlyOwner {

whitelist[\_address] = true;

}

// Sign the document on behalf of a specific whitelisted address

function signDocument(address \_address) external onlyOwner onlyWhitelisted(\_address) {

require(!signatures[\_address], "Address has already signed");

signatures[\_address] = true;

signatureCount++;

emit DocumentSigned(\_address);

}

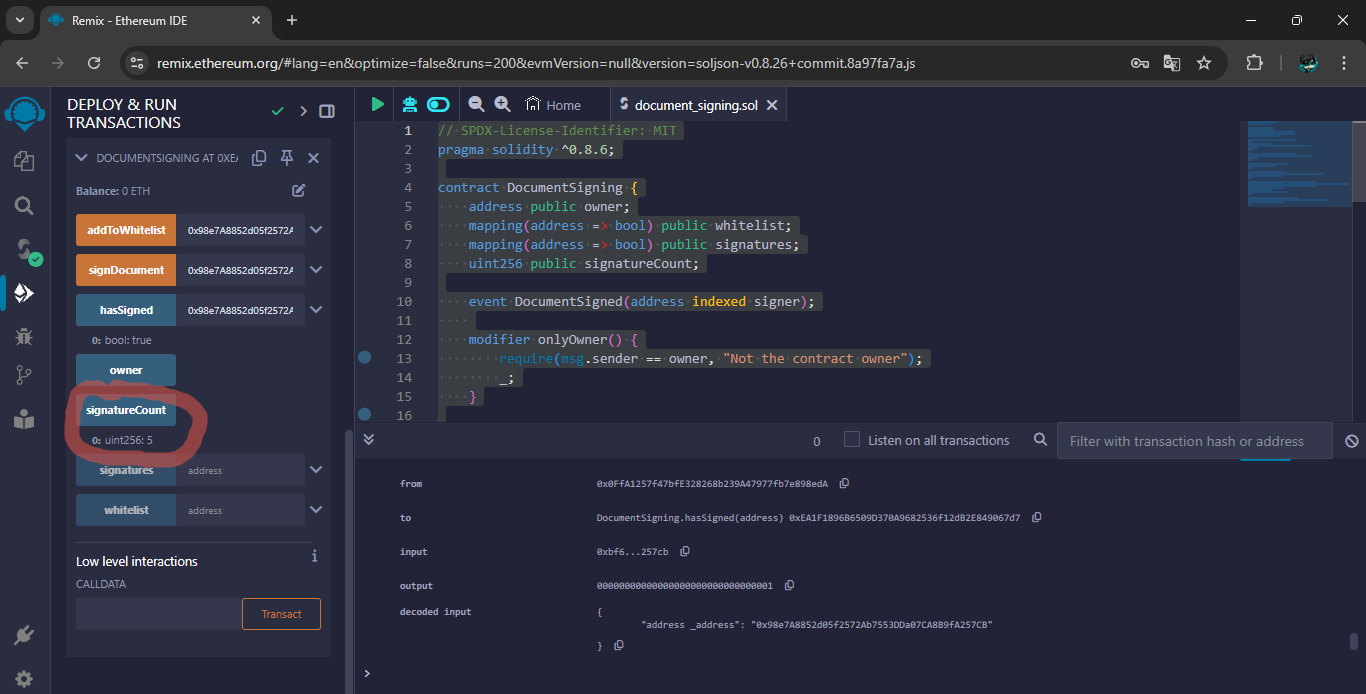
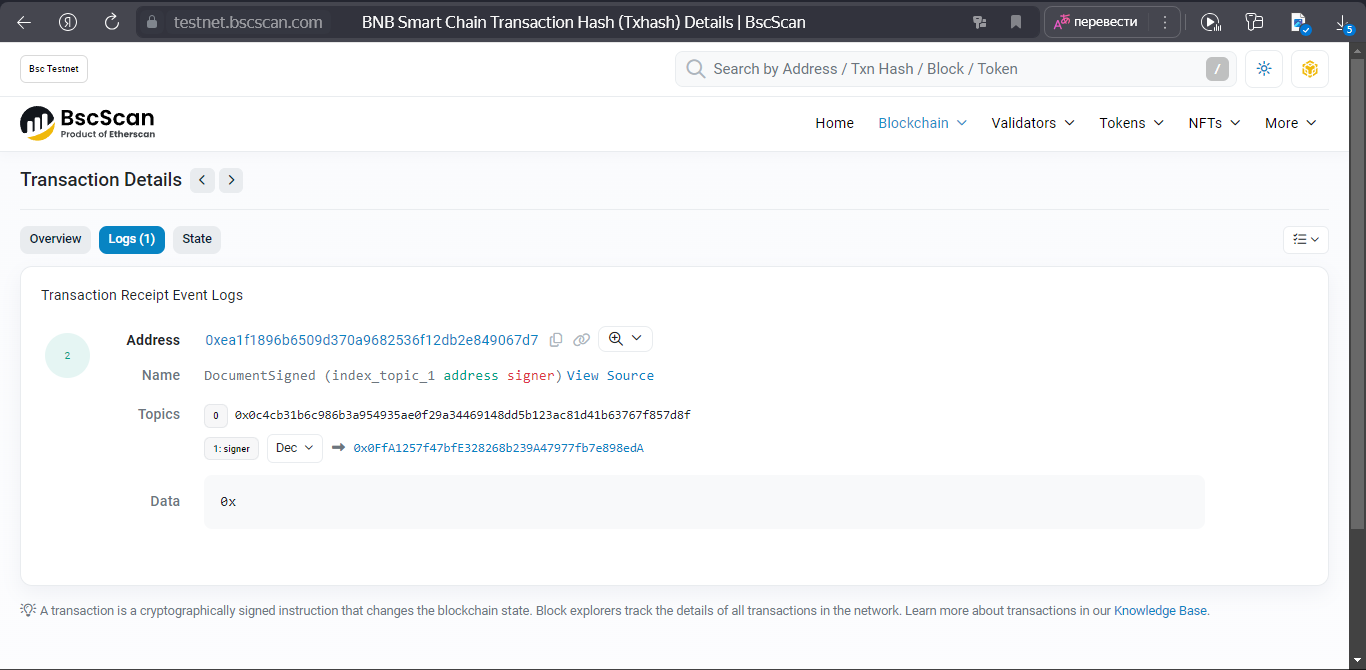
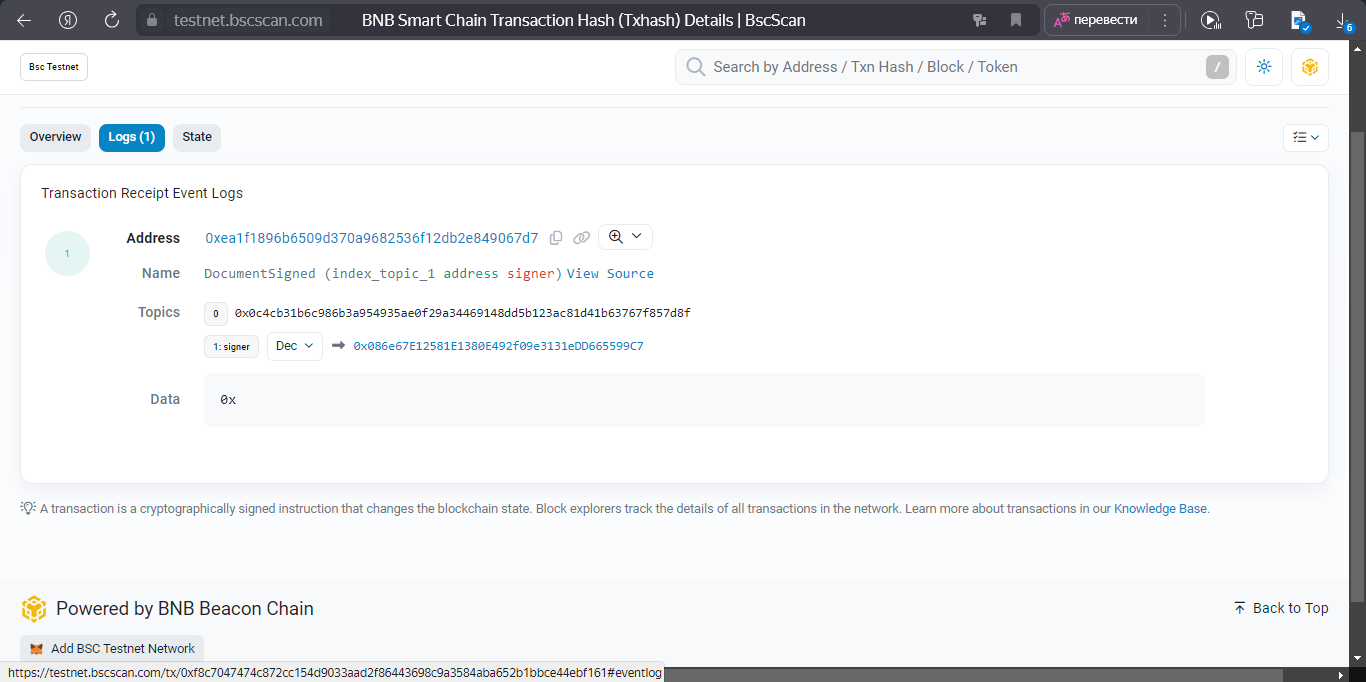
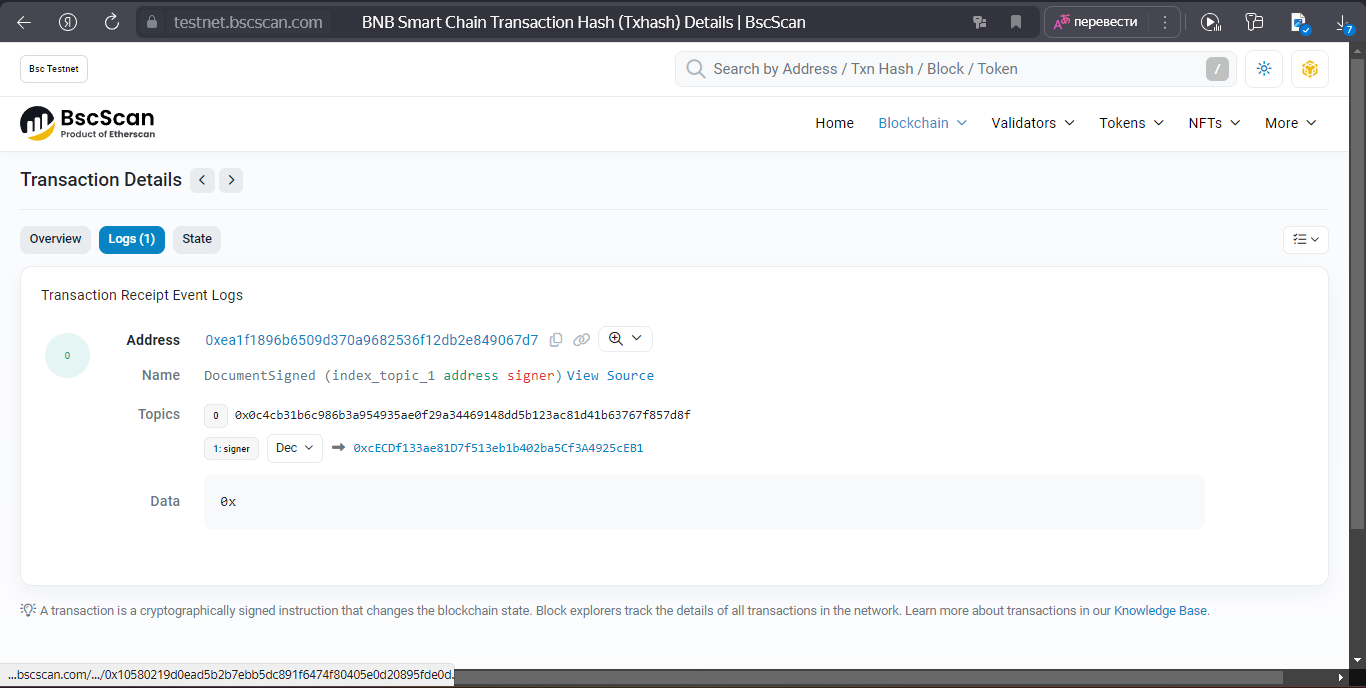
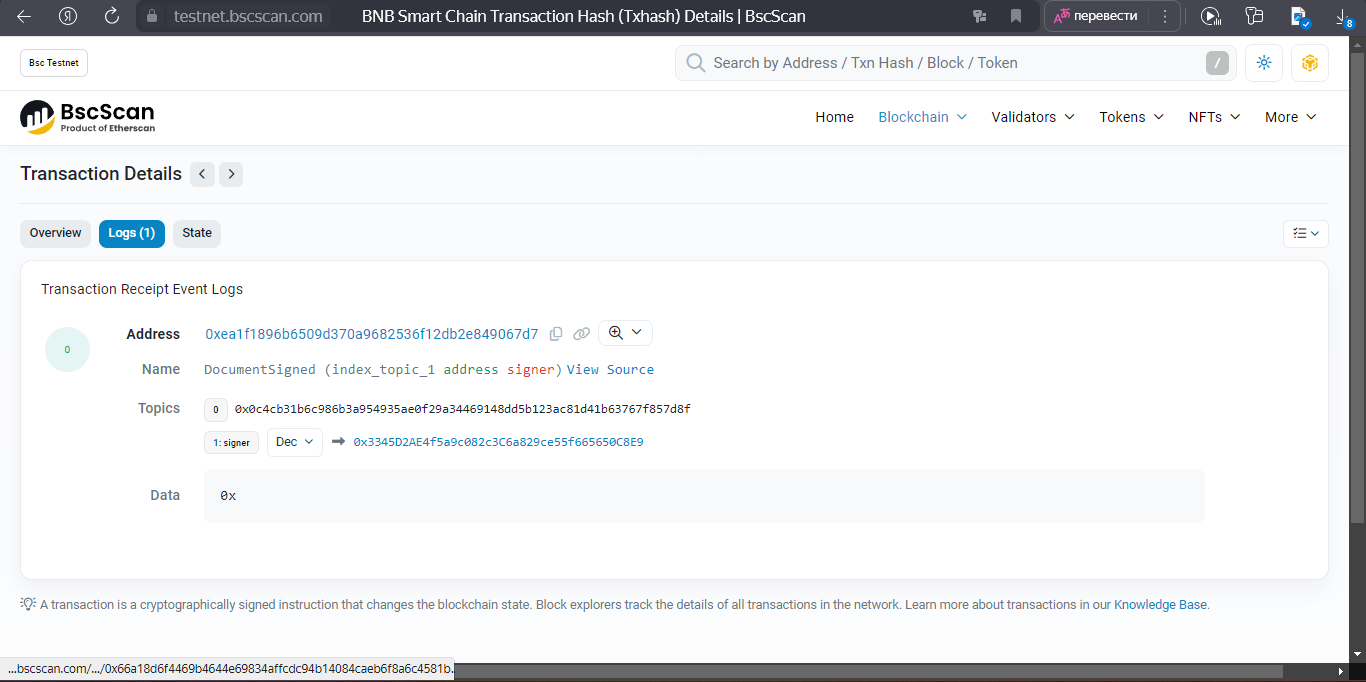
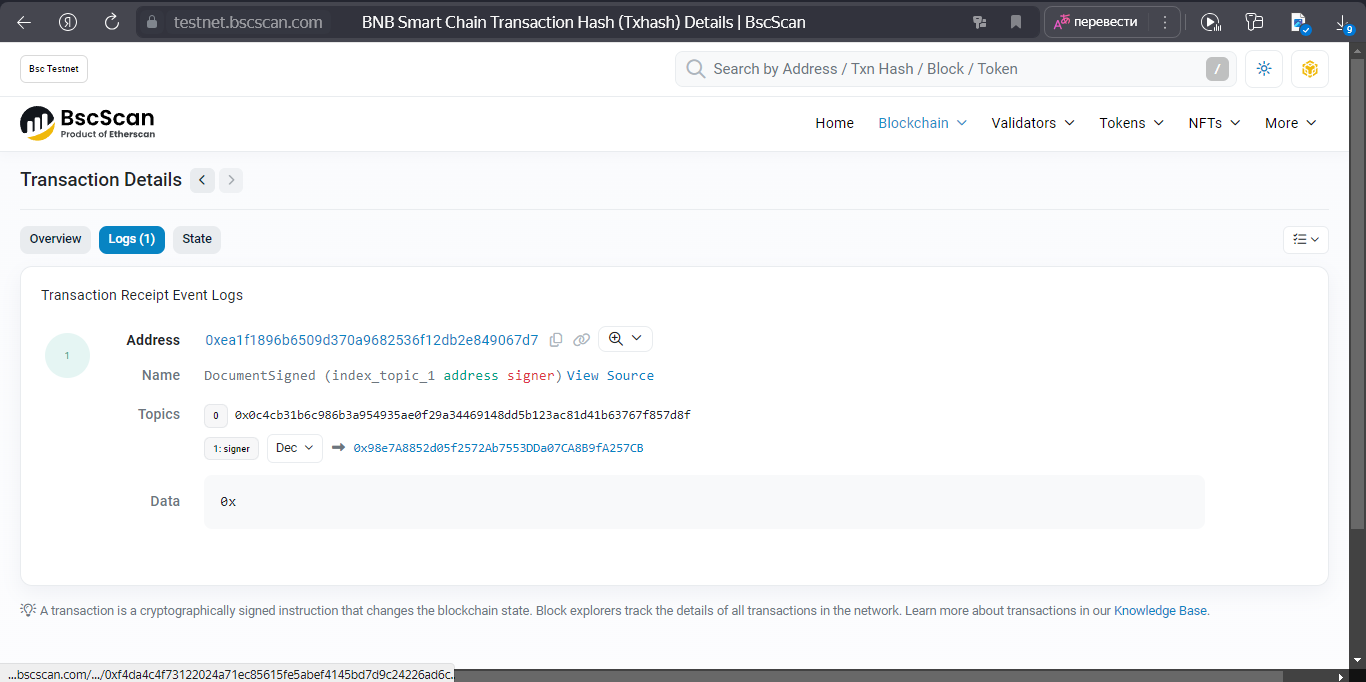
// Check if an address has signed the document

function hasSigned(address \_address) external view returns (bool) {

return signatures[\_address];

}

}

**Link to the transaction where the document is signed:** [**https://testnet.bscscan.com/address/0xea1f1896b6509d370a9682536f12db2e849067d7**](https://testnet.bscscan.com/address/0xea1f1896b6509d370a9682536f12db2e849067d7)**Contract hash: 0xEA1F1896B6509D370A9682536f12dB2E849067d7**  
  
**As you can see from the picture signatureCount shows 5 it means there are 5 signatures from our 5 team members. The image below will be the proof:**  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**1)**  
**2)**  
  
  
  
  
  
  
  
  
  
  
  
  
**3)**  
**4)**  
  
  
  
  
  
  
  
  
  
  
  
  
**5)**  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**Task 2 - Token and Airdrop (group)**  
**Answer (insert code from Remix):**  
  
**MyToken.sol**  
  
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.6;

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";

contract MyToken is ERC20 {

constructor(uint256 initialSupply) ERC20("MyToken", "MTK") {

\_mint(msg.sender, initialSupply);

}

}  
  
**TokenAirDrop.sol**  
  
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.6;

import "@openzeppelin/contracts/token/ERC20/IERC20.sol";

contract TokenAirdrop {

function airdropTokens(IERC20 token, address[] memory recipients, uint256 amount) public {

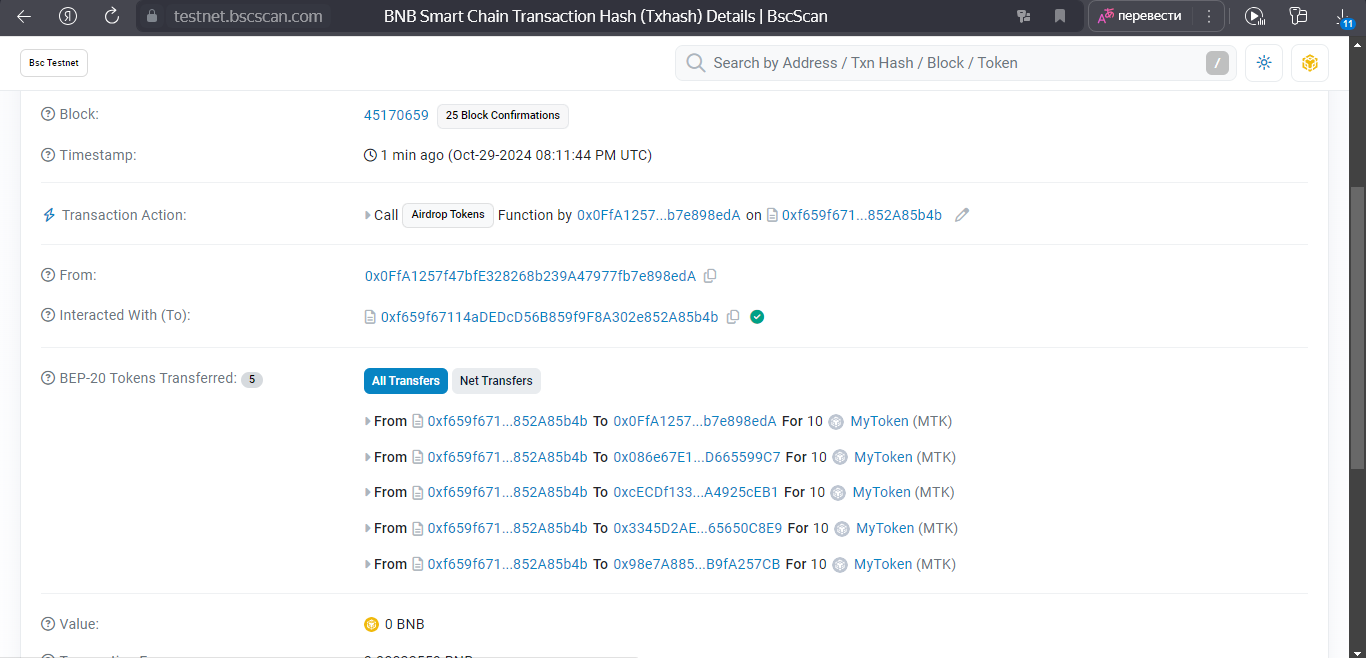
// Проверяем, достаточно ли токенов в контракте для выполнения airdrop

require(token.balanceOf(address(this)) >= amount \* recipients.length, "Not enough tokens in contract");

for (uint256 i = 0; i < recipients.length; i++) {

token.transfer(recipients[i], amount);

}

}  
}  
  
  
**Link to the transaction where you sent tokens:** <https://testnet.bscscan.com/tx/0xe3ecc70289610e0cca4fe9eb3f7dd8cff8962dde90bfc3a71cc4d96599eb00f8>**Transaction Hash:**  
 **0xe3ecc70289610e0cca4fe9eb3f7dd8cff8962dde90bfc3a71cc4d96599eb00f8**  
  
  
**Task 3 – Rock-Paper-Scissors (frontend) (group)**  
  
**The address of the Rock-Paper-Scissors contract is:** 0xC666Eb490BC3697d441eea36f6e4D1C032c94203  
  
**link to the demo:**   
https://blockchain-assignment-3.vercel.app/  
  
**link to github:**   
https://github.com/xxkernel/blockchain-assignment-3