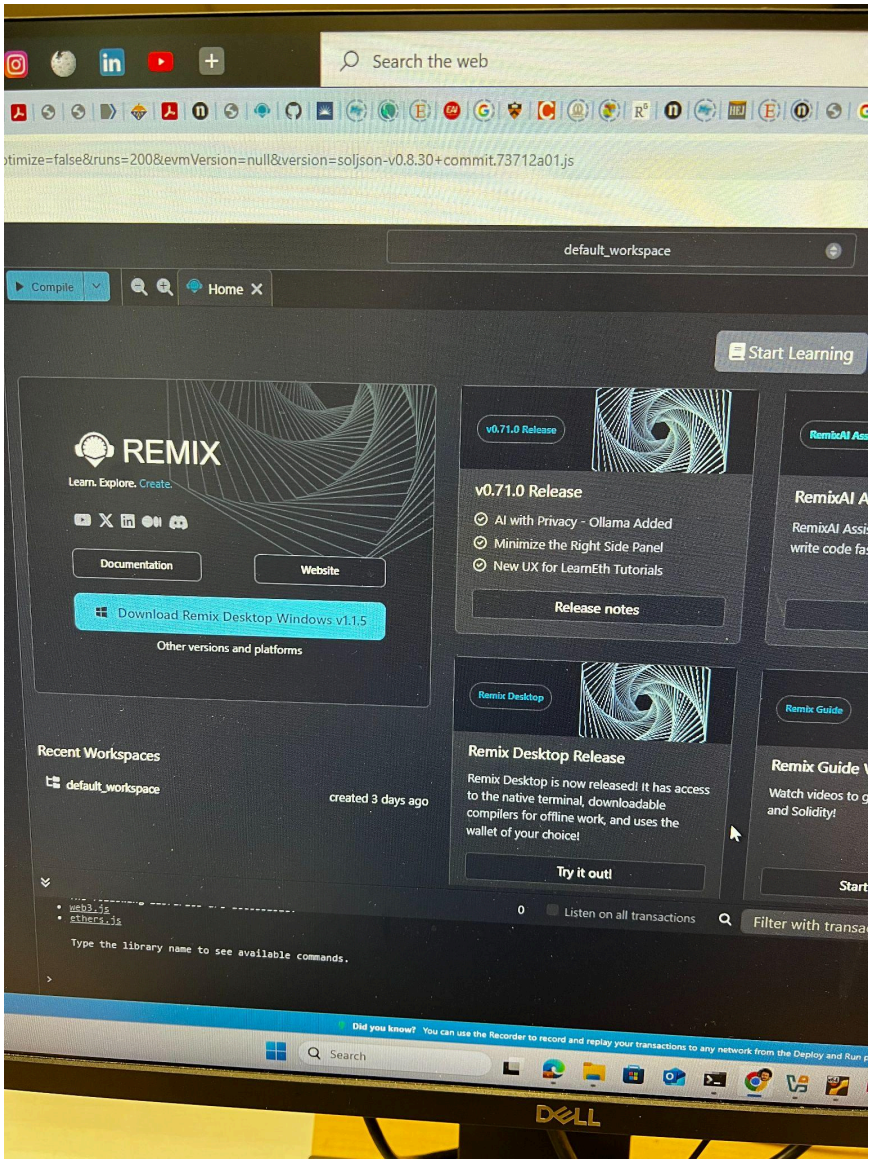
Blockchain and Smart Contracts

CSCI 5450

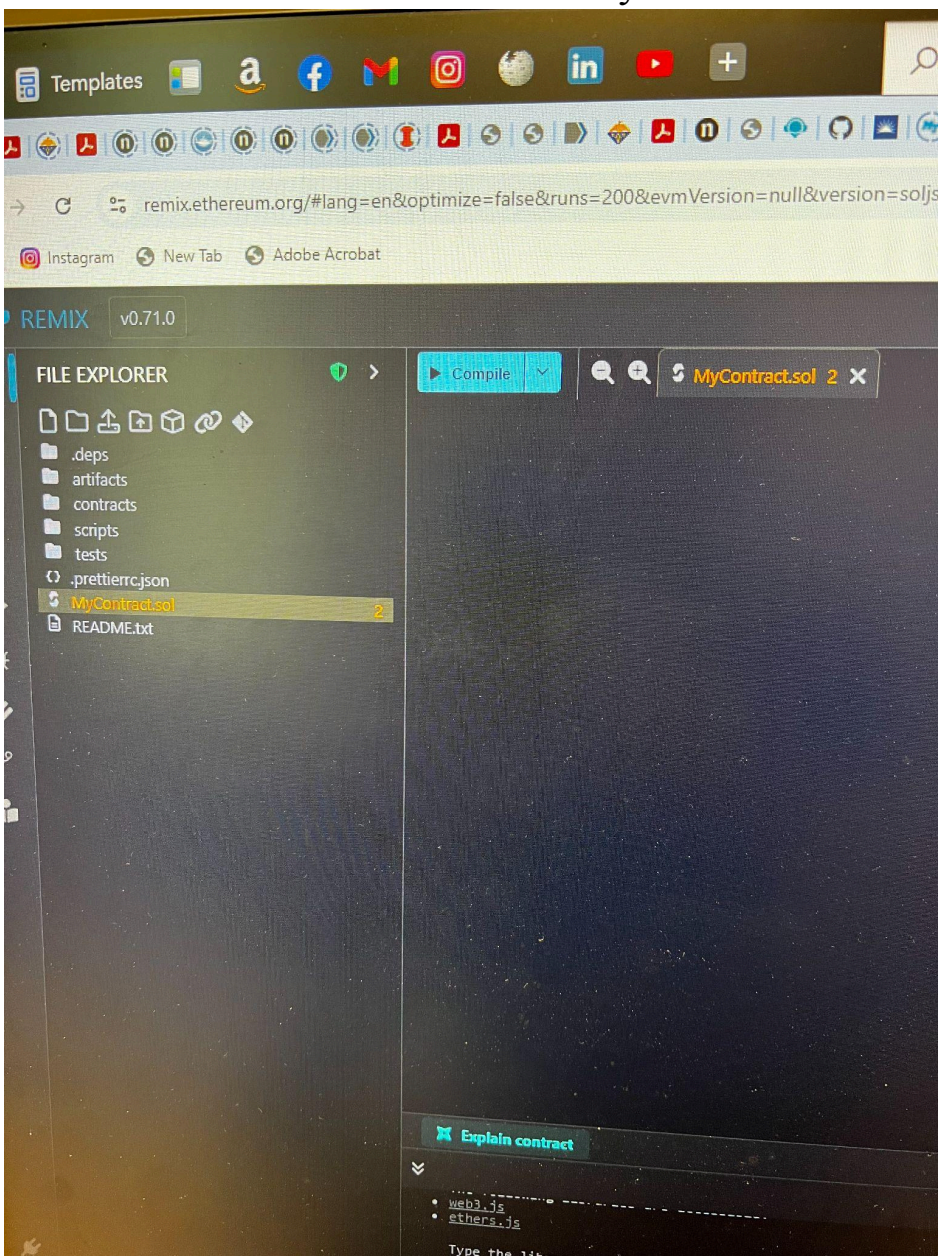
Assignment2:Connecting with a Public Testnet Blockchain

WASIM RAJA MONDAL

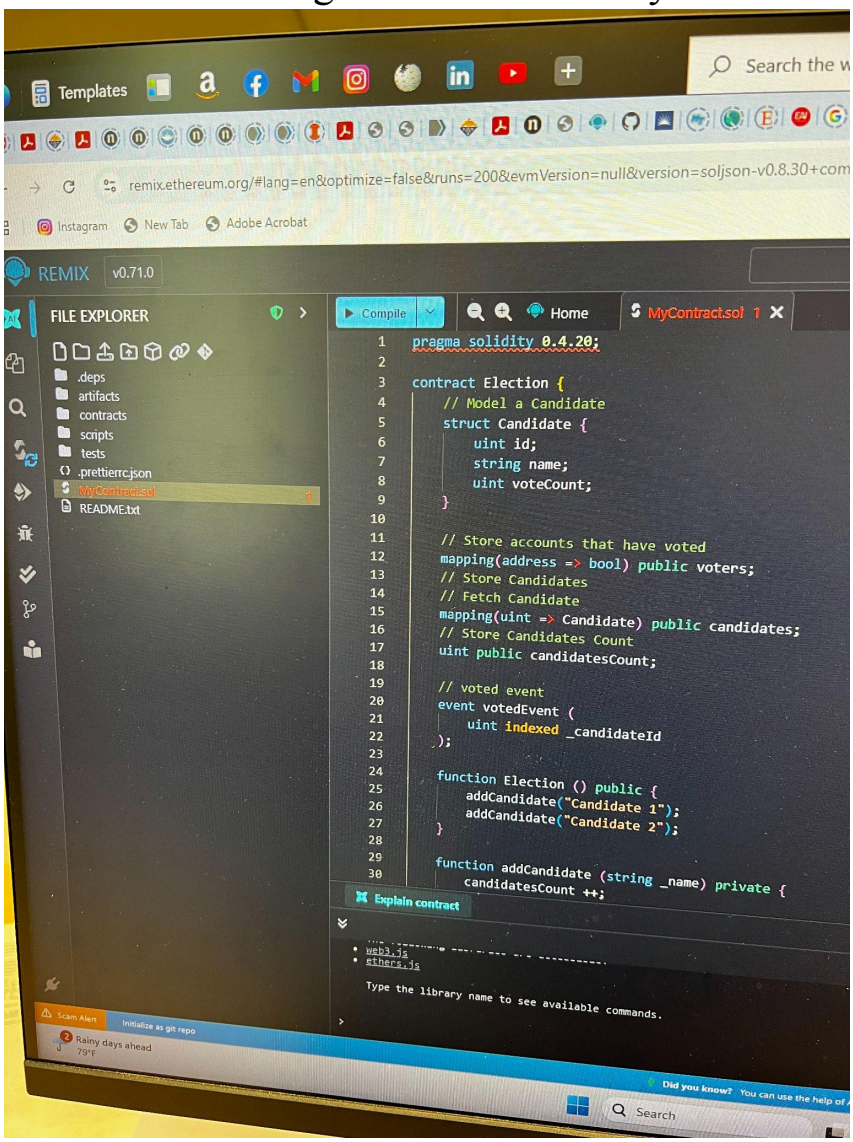
Open Remix IDE from your browser



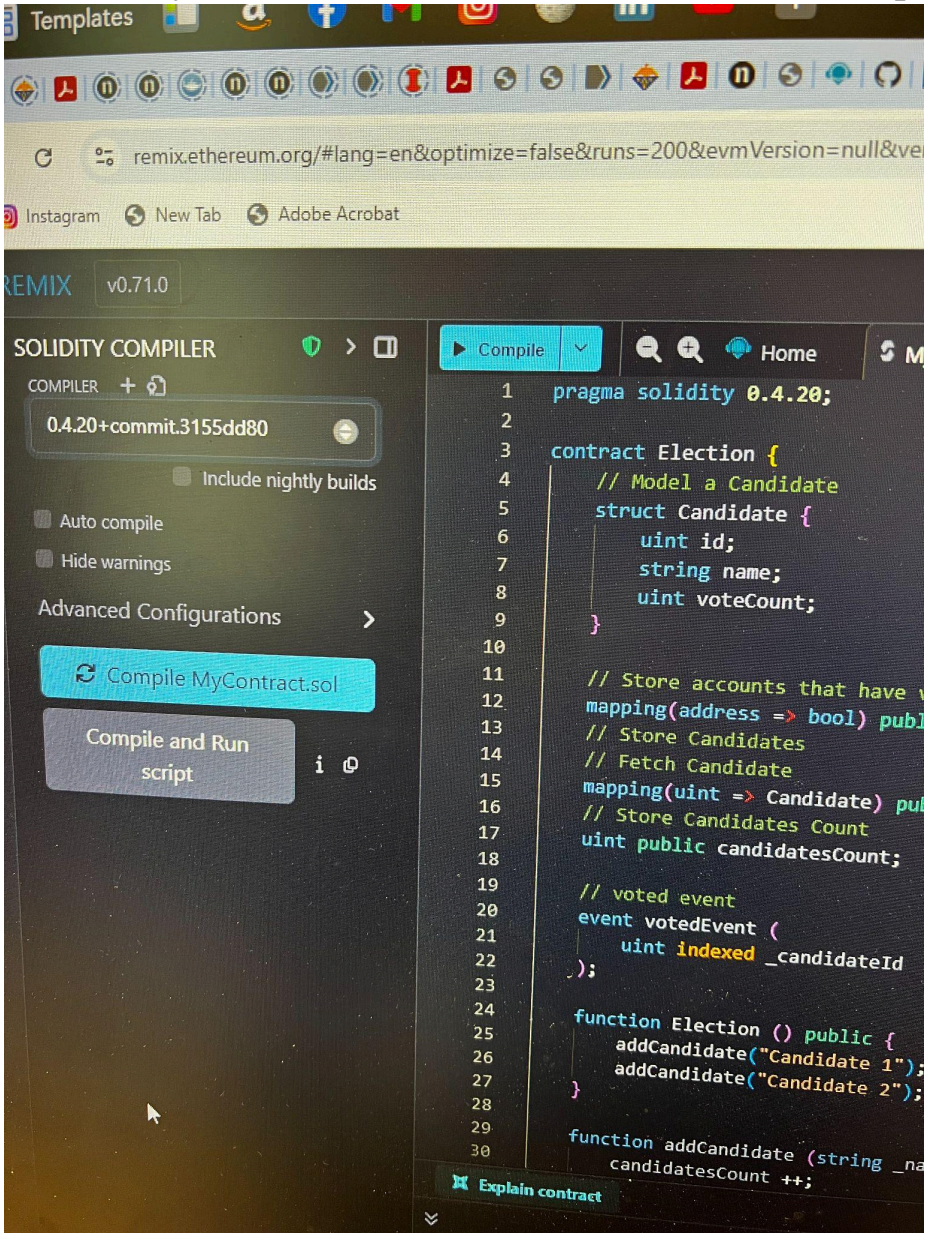
Click on New File and name it as MyContract.sol



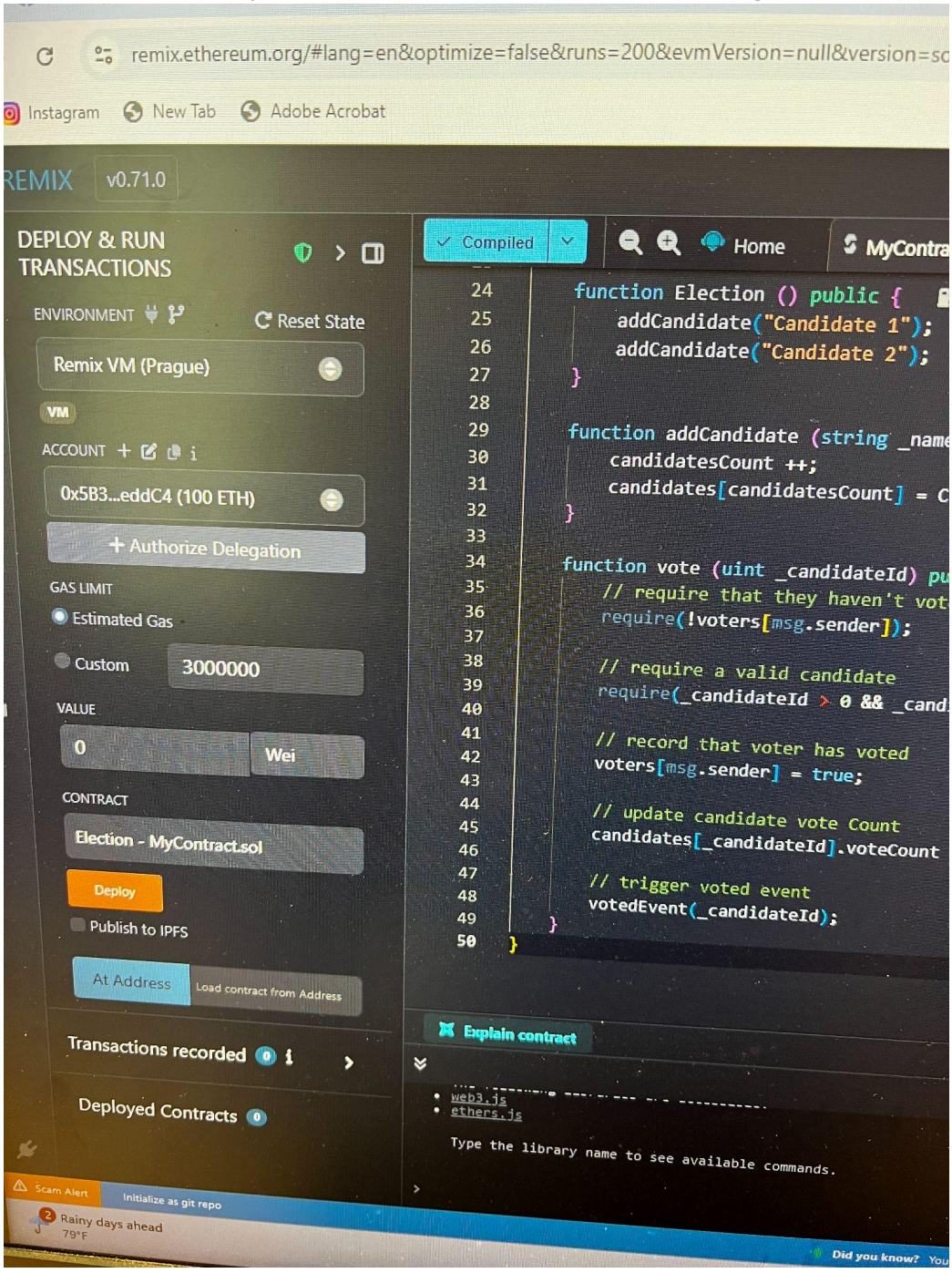
Write the following code in the file MyContract.sol I just created



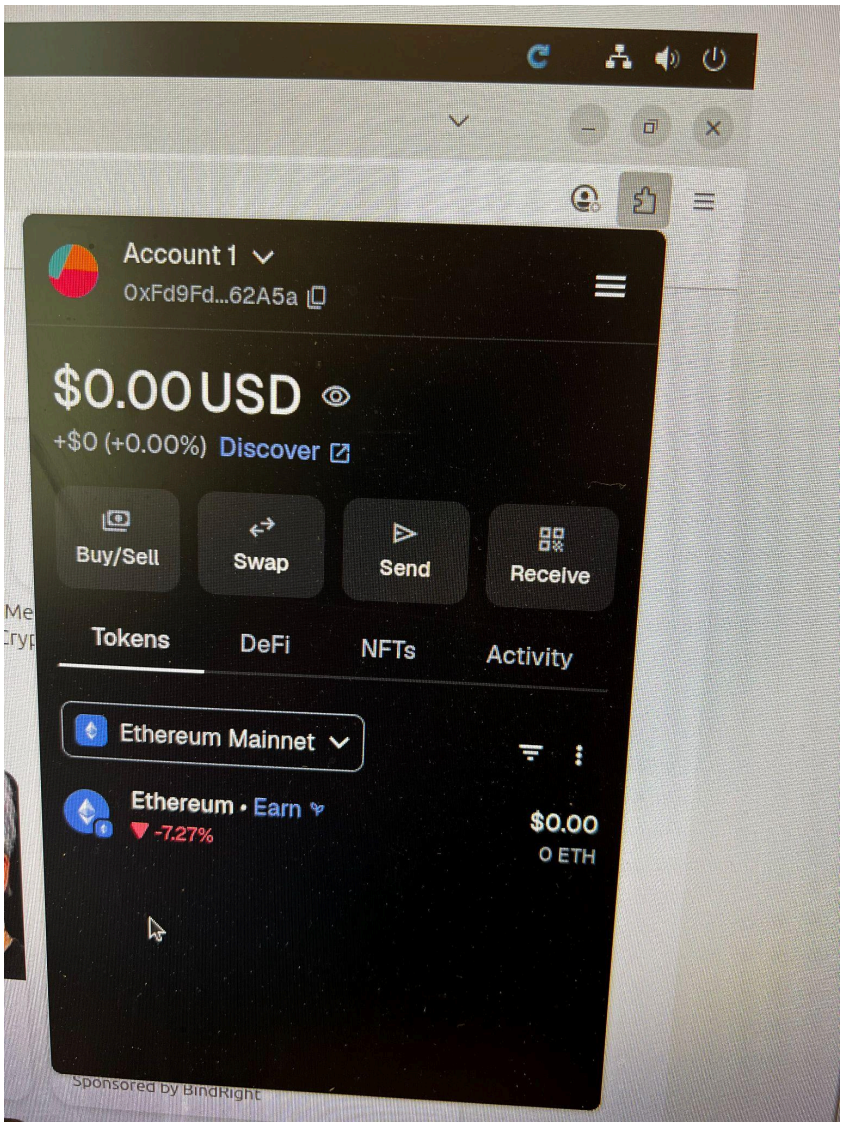
Now, click on the Solidity compiler as shown in the screenshot below and select your compiler version. Since, the smart contract was written in 0.4.20, you should select the same version of compiler.



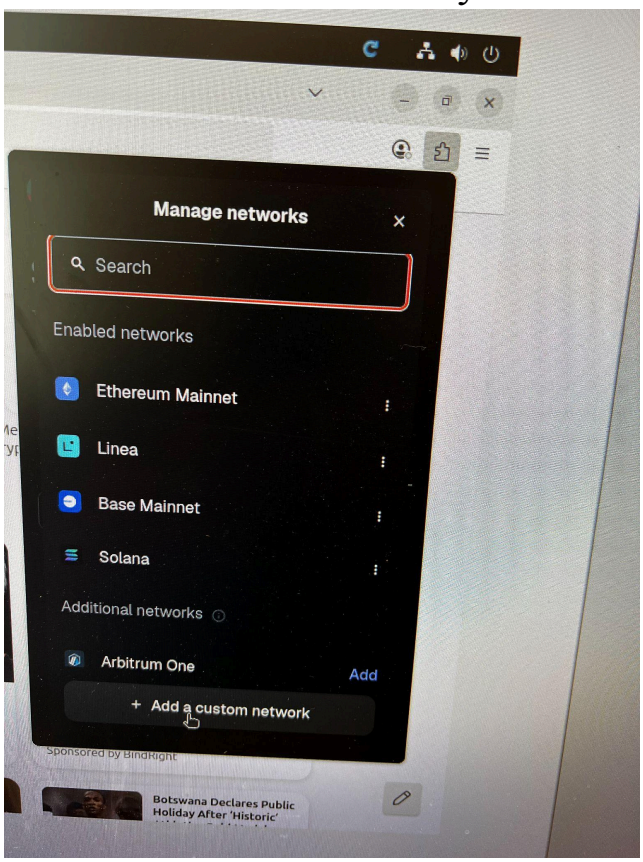
You will also see a Deploy button. In Remix IDE, you can deploy smart contracts directly to the blockchain without using the Truffle framework.

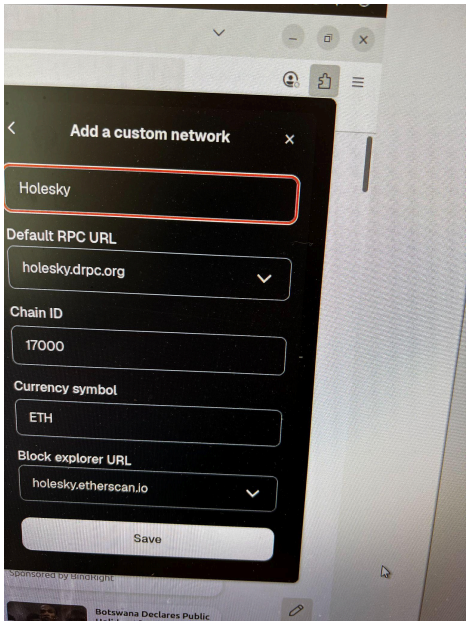


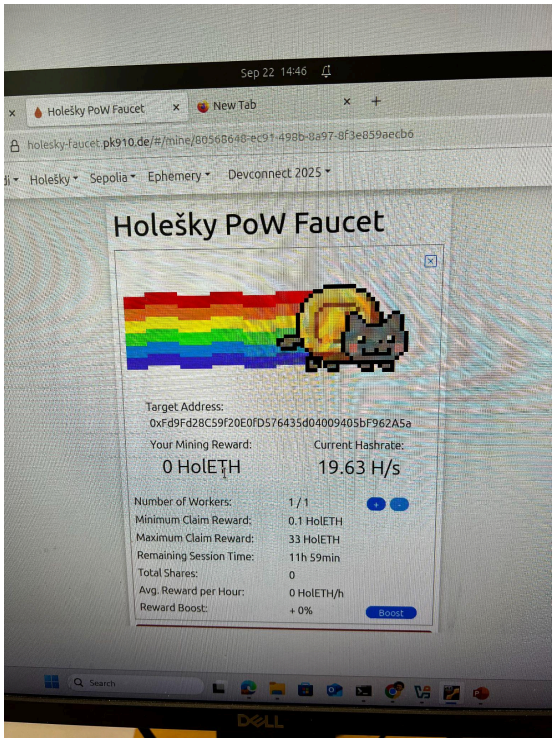
We should not deploy now because we do not have ETH in our account as shown below:

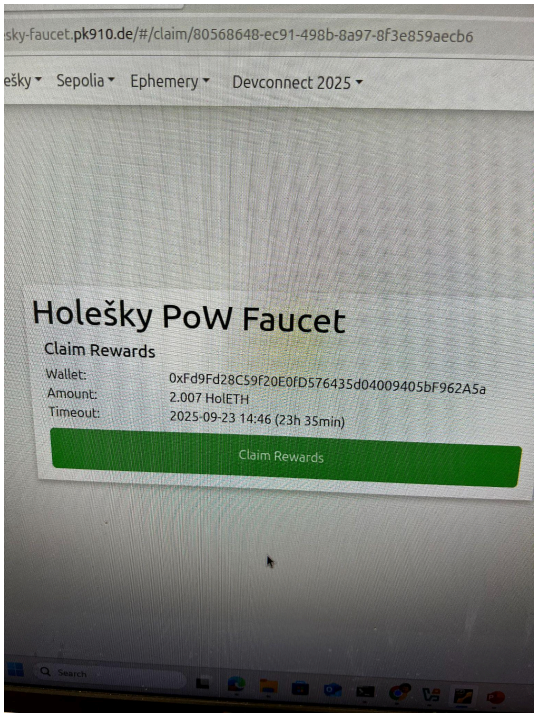


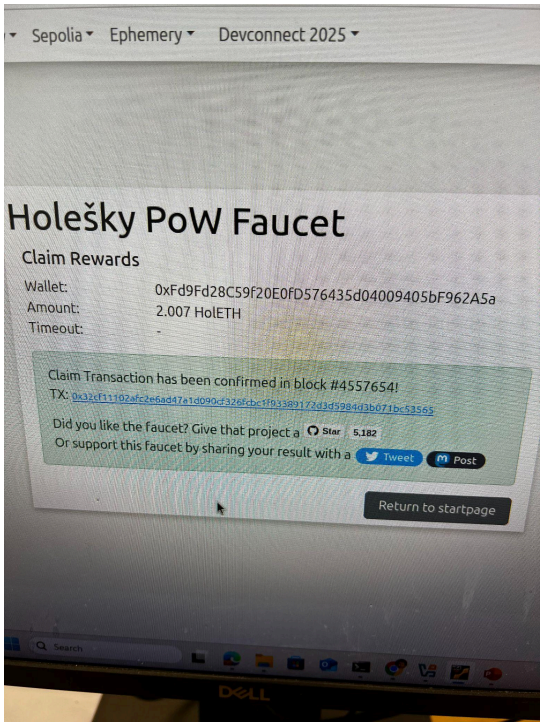
We need to add the Holesky network to our MetaMask account



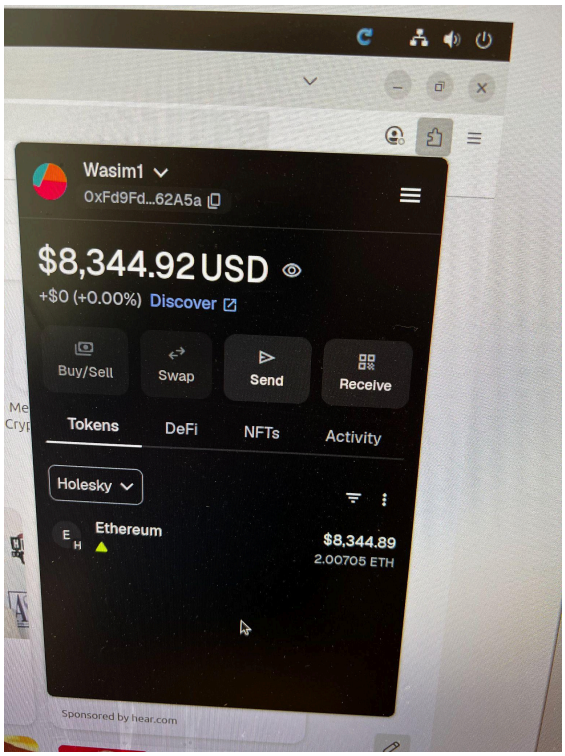



Sep 22 14:46
Holešky PoW Faucet
New Tab
holesky-faucet.pk910.de/#/mine/80568648-ec91-498b-8a97-8f3e859aecb6
Holešky
Sepolia
Ephemery
Devconnect 2025
Holešky PoW Faucet
Target Address:
0xFd9Fd28C59f20E0fD576435d04009405bF962A5a
Your Mining Reward:
0 HolETH
Current Hashrate:
19.63 H/s
Number of Workers:
1 / 1
Minimum Claim Reward:
0.1 HolETH
Maximum Claim Reward:
33 HolETH
Remaining Session Time:
11h 59min
Total Shares:
0
Avg. Reward per Hour:
0 HolETH/h
Reward Boost:
+ 0%
Boost
Search
DELL

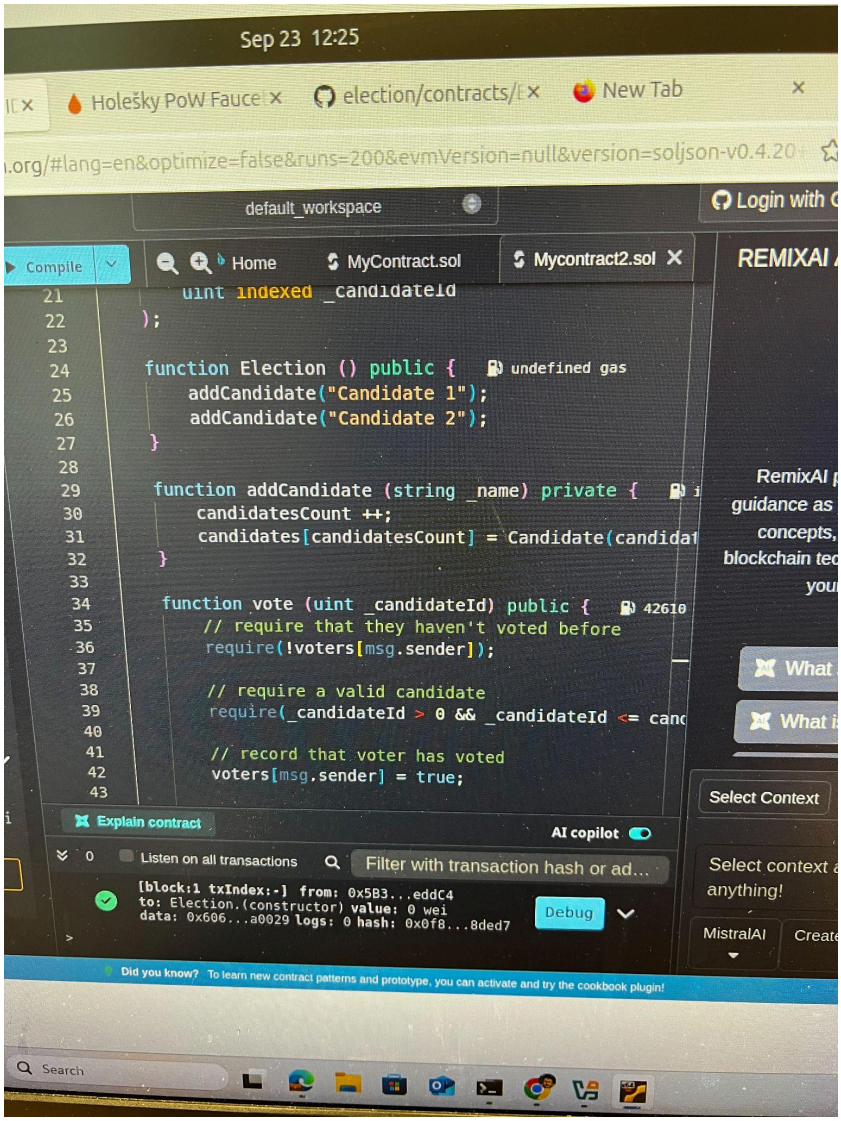

sky-faucet.pk910.de/#/claim/80568648-ec91-498b-8a97-8f3e859aecb6
ešky
Sepolia
Ephemery
Devconnect 2025
Holešky PoW Faucet
Claim Rewards
Wallet:
0xFd9Fd28C59f20E0fD576435d04009405bF962A5a
Amount:
2.007 HolETH
Timeout:
2025-09-23 14:46 (23h 35min)
Claim Rewards
Search



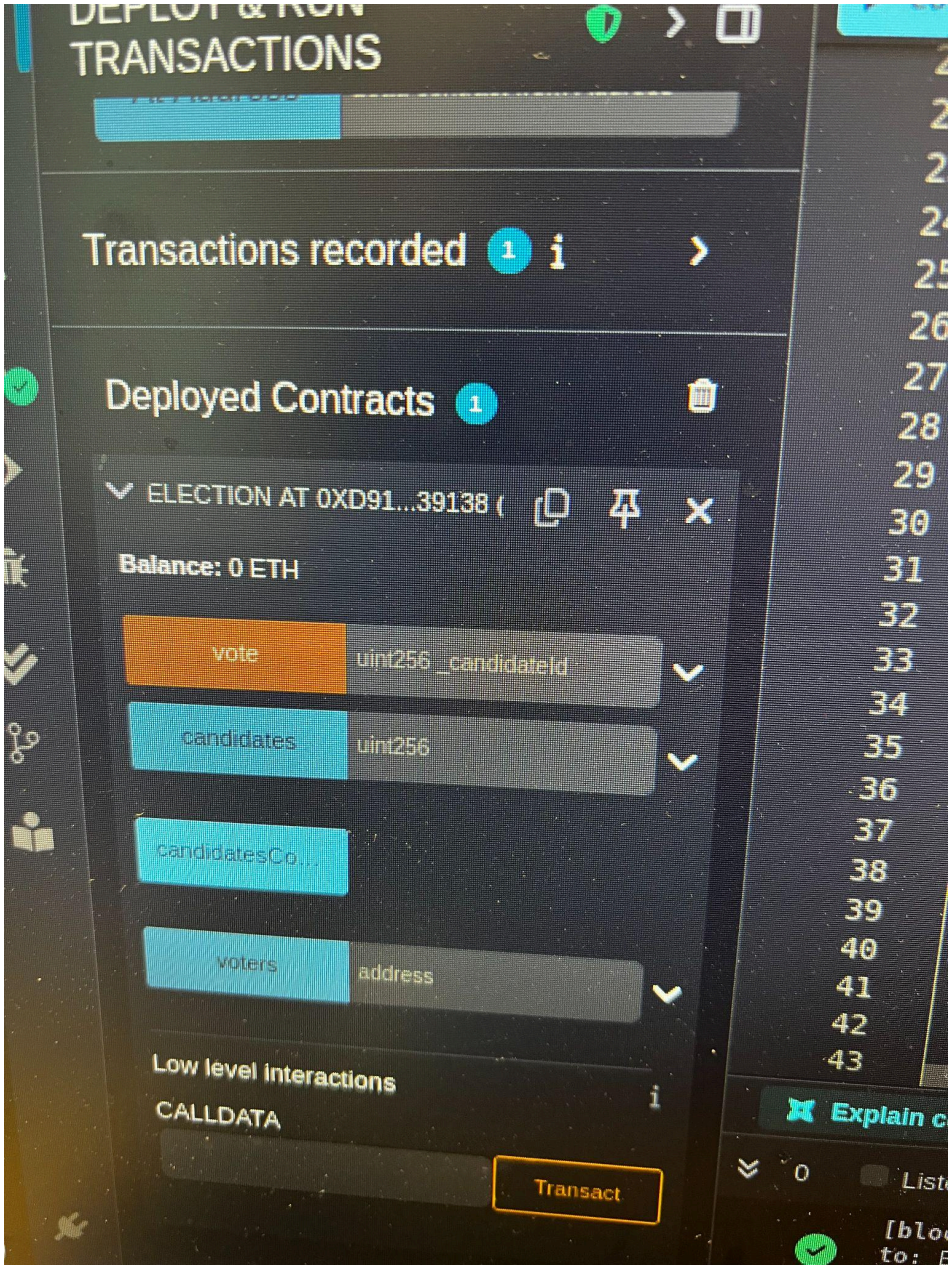
The updated balance on the MetaMask account as shown below:



The contract has been deployed successfully, as confirmed by a green check mark in the console as shown in the screenshot below:

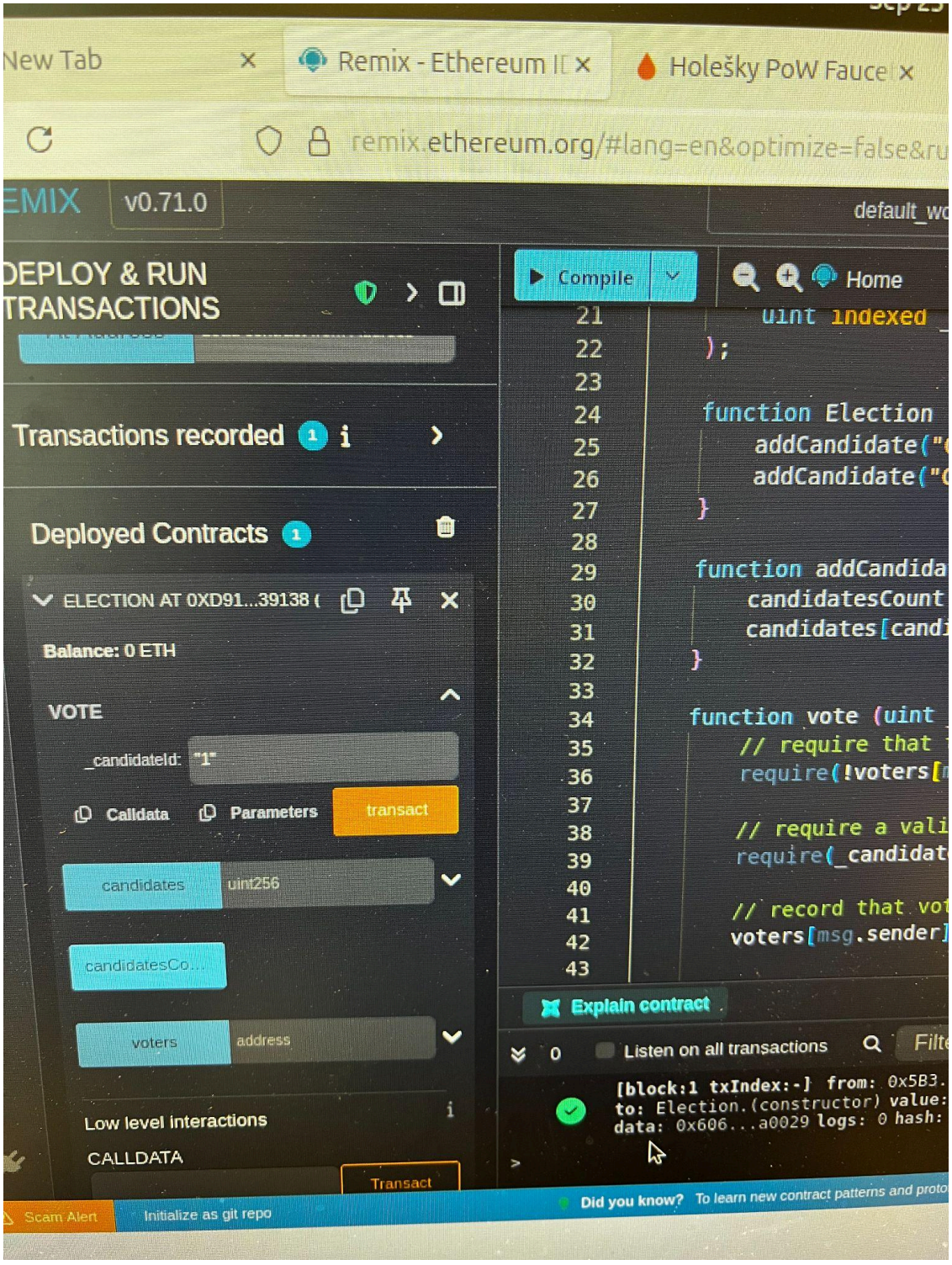


The details of the smart contract with its address, along with the available items, as shown below in the screenshot.

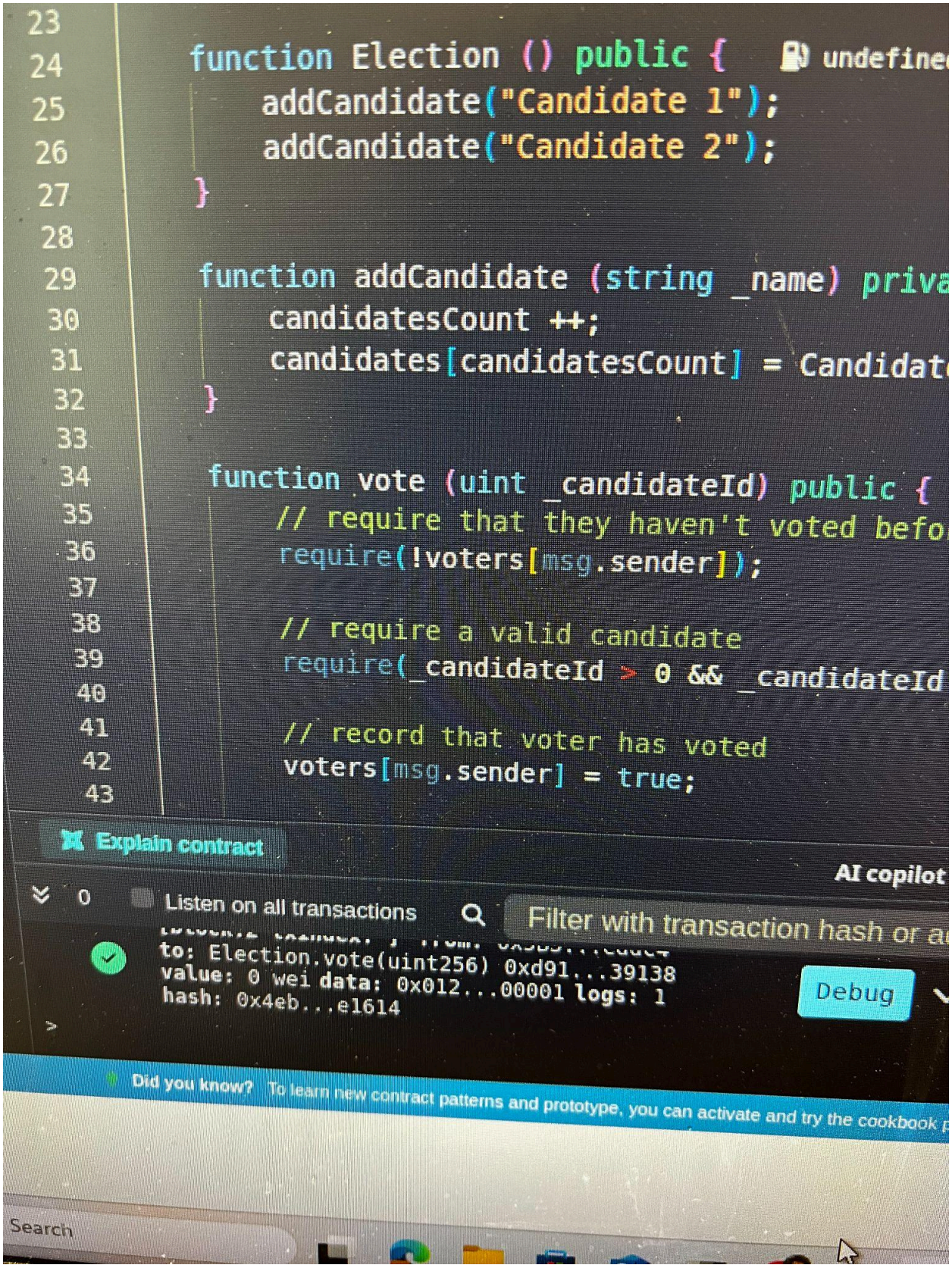


As we can see above screenshot, in this deployed smart contract, we have four options. The one shown in orange is the vote function, which changes the blockchain state and therefore requires “gas” (gas is something we will cover later in class) to execute. The three shown in blue (candidates, candidatesCount, and voters) are not user-defined functions but public state variables. Solidity automatically generates getter functions for public variables, which is why Remix IDE allows us to call them. These getters are read-only and free to use, while the orange function writes to the blockchain and consumes gas

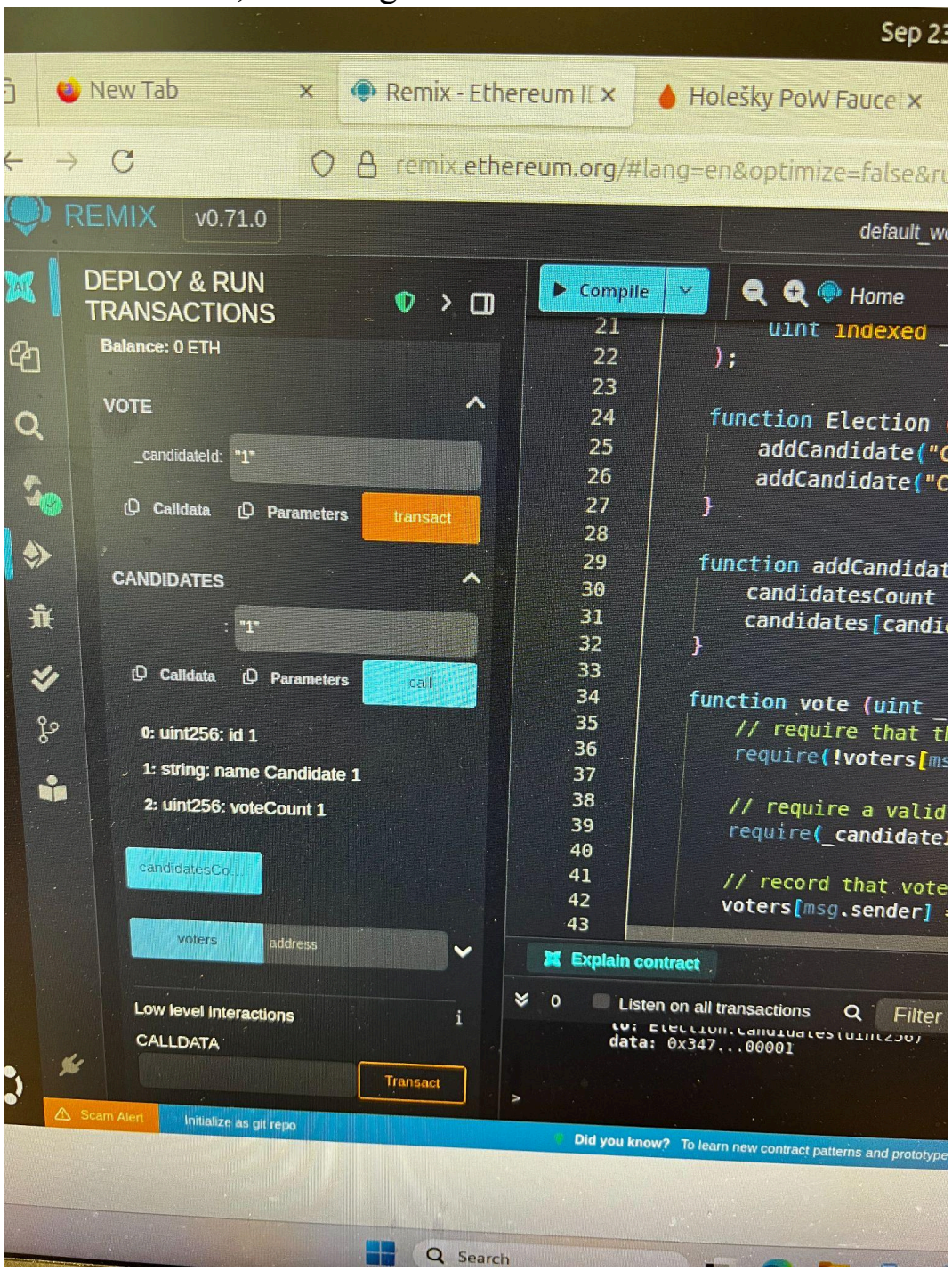
Insert a value of “1” to record a vote for Candidate 1 on the blockchain using this smart contract.



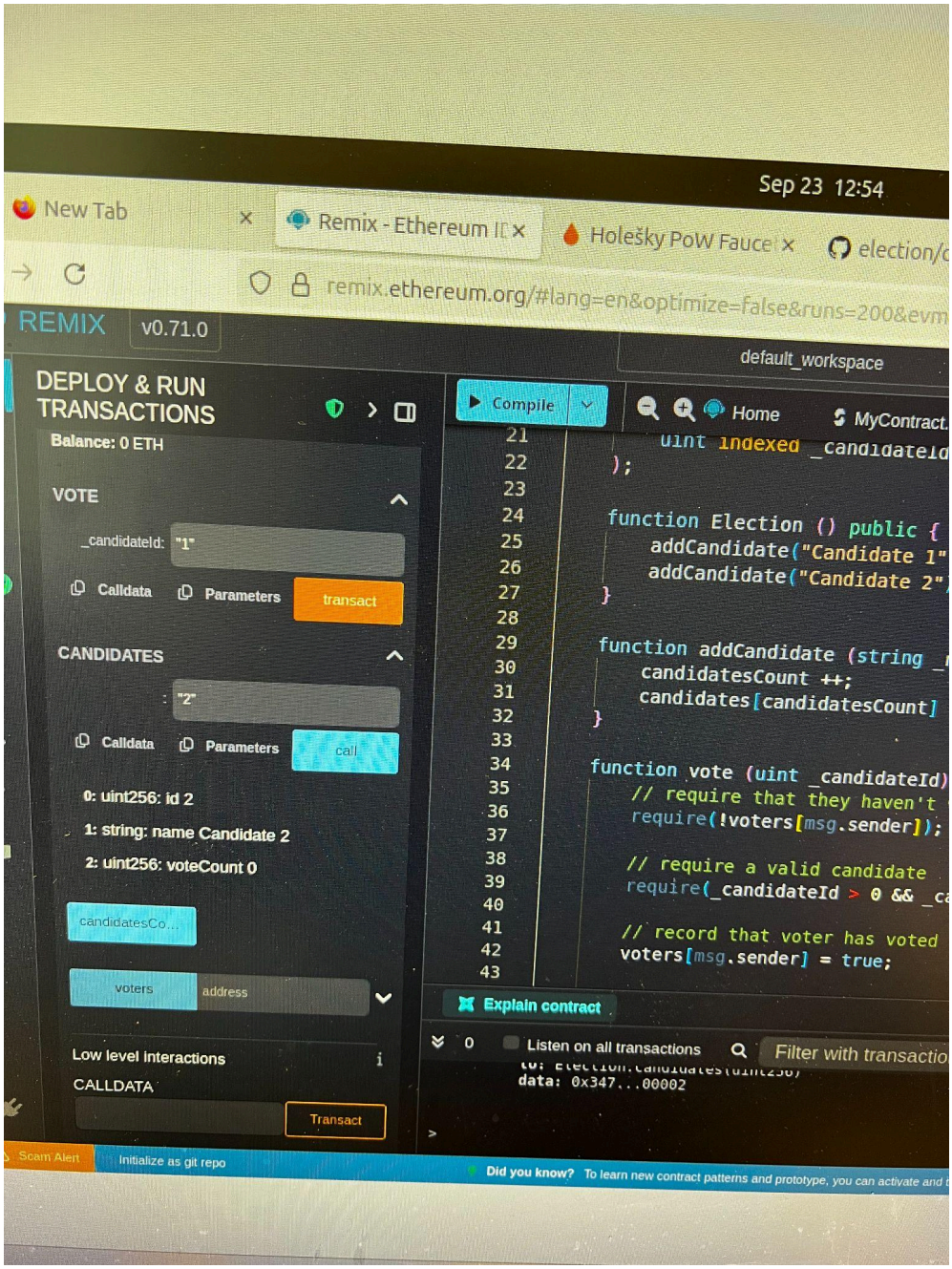
Showing the confirmation of transact


function Election () public {
addCandidate("Candidate 1");
addCandidate("Candidate 2");
}
function addCandidate (string _name) priva
candidatesCount ++;
candidates[candidatesCount] = Candidat
}
function vote (uint _candidateId) public {
// require that they haven't voted befo
require(!voters[msg.sender]);
// require a valid candidate
require(_candidateId > 0 && _candidateId
// record that voter has voted
voters[msg.sender] = true;
Explain contract
AI copilot
Listen on all transactions
Filter with transaction hash or a
to: Election.vote(uint256) 0xd91...39138
value: 0 wei data: 0x012...00001 logs: 1
hash: 0x4eb...e1614
Debug
Did you know? To learn new contract patterns and prototype, you can activate and try the cookbook
Search

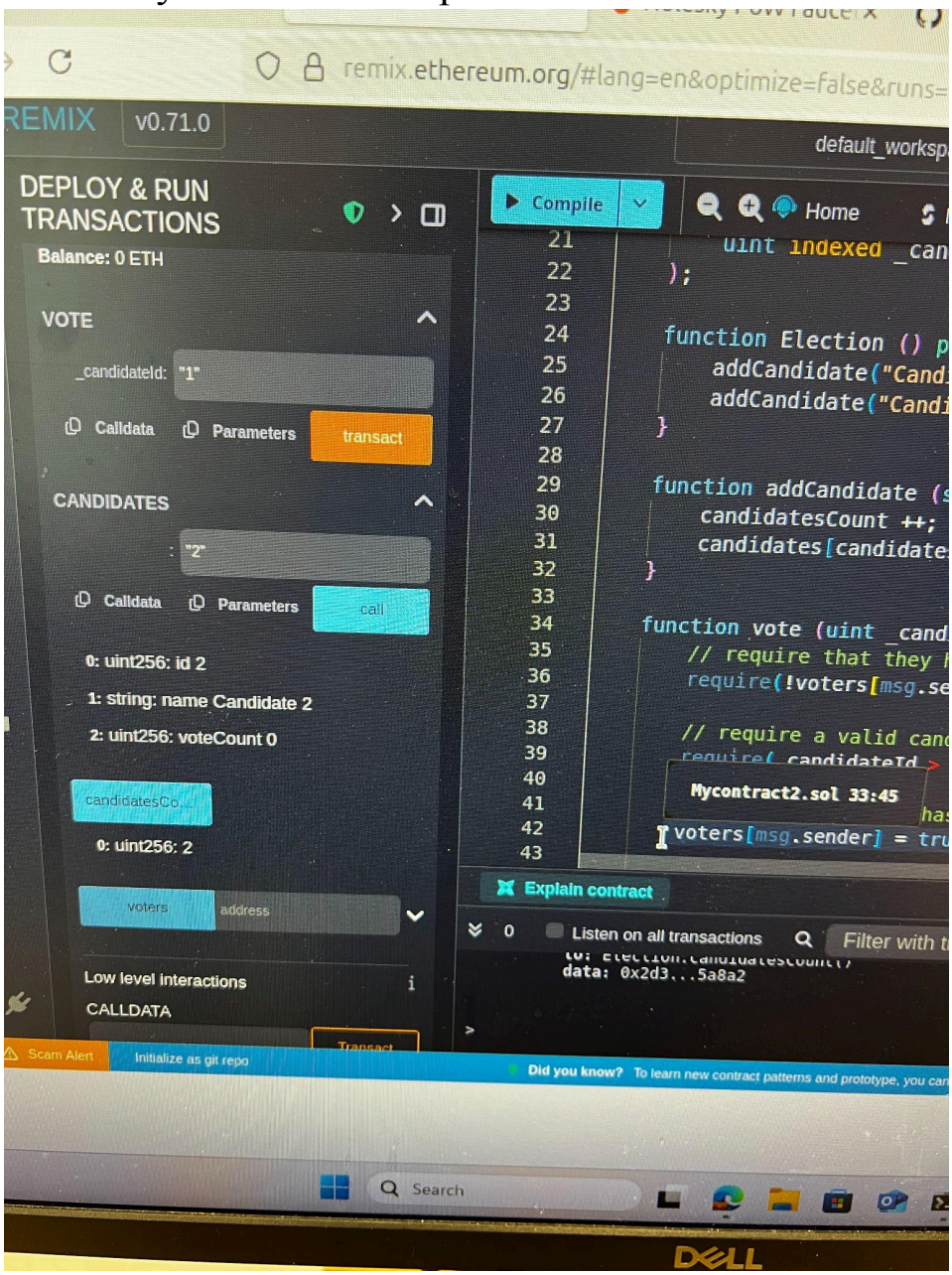
Now that you have cast a vote for Candidate 1, use the candidates option (listed under the vote function), which is a public variable with an auto-generated getter function. Enter the value 1 to see the information for Candidate 1, including the number of votes received.



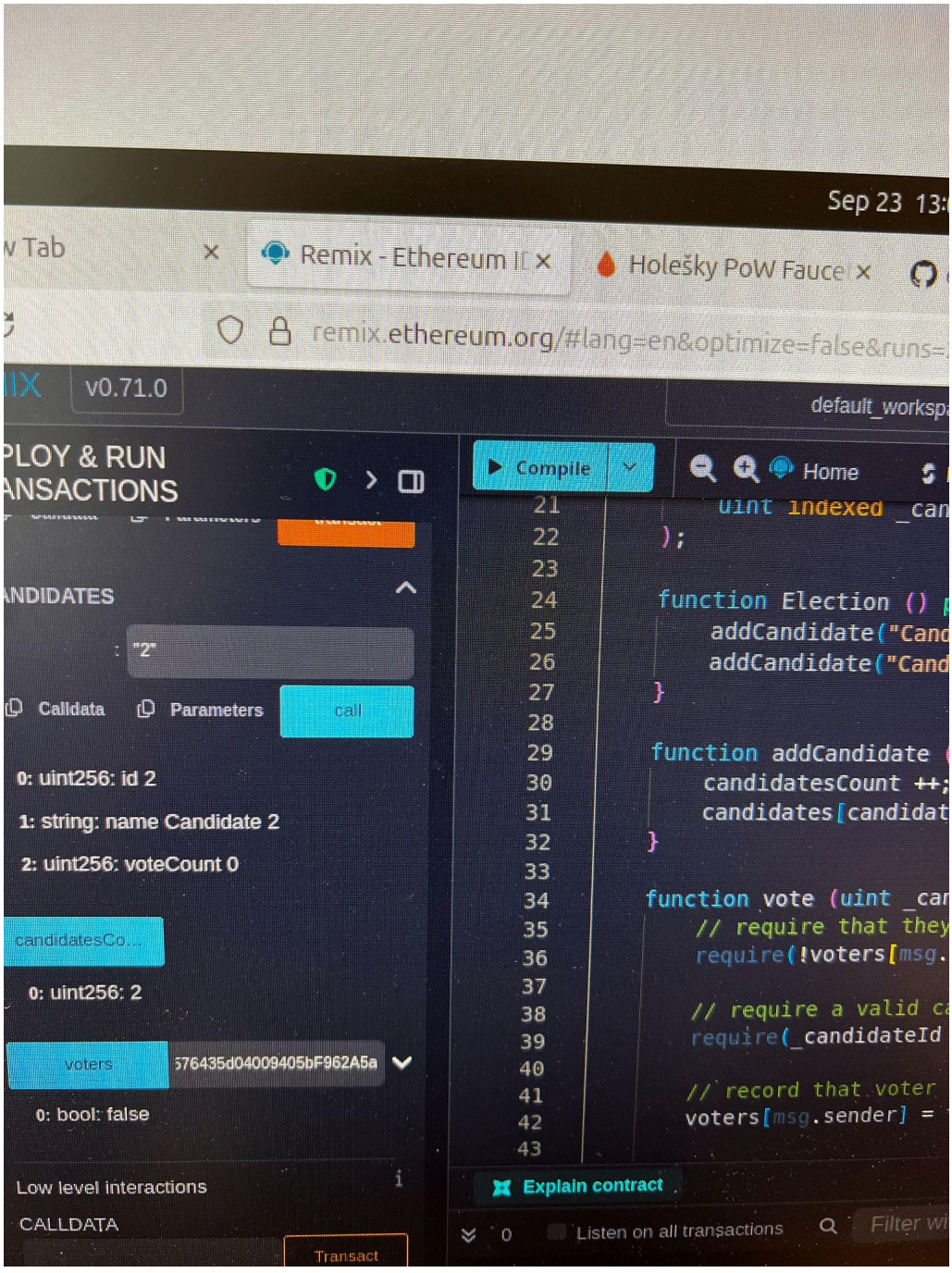
Do the same for Candidate 2 to show whether Candidate 2 has received any votes even if you have not cast a vote for Candidate 2 yet.



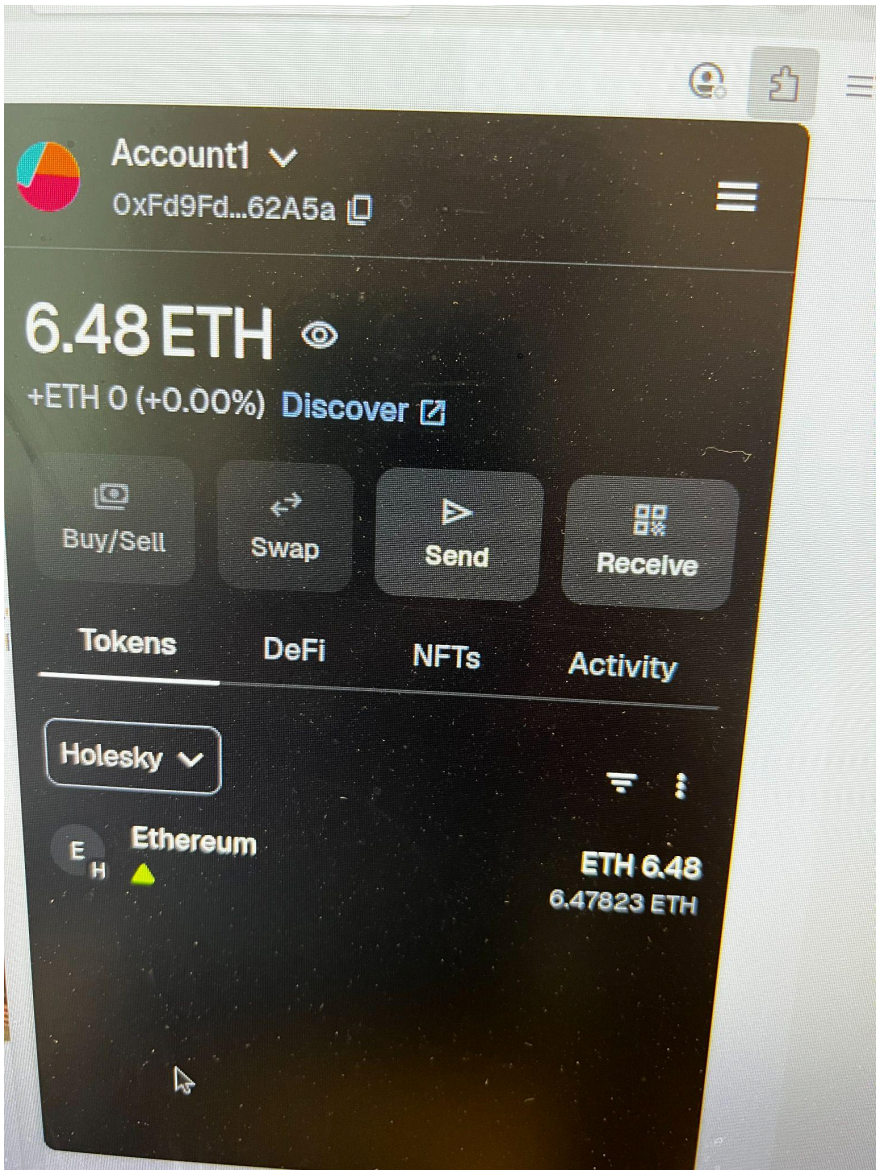
If you click on the third option, candidatesCount, it will simply show how many candidates are present in the election.



Finally, if you paste the account address of a voter (Account 1) from your MetaMask and click on the fourth option, voters, you will see information in the console about that voter (Account 1) and whether they have voted.



Take a screenshot of your MetaMask wallet showing that your account has a balance of ETH.



Try voting for Candidate 1 again using the same account address. Can you revote? What message do you receive? Provide a screenshot as evidence.

I tried to revote as candidate 1. But it was not successful. This is shown in the screenshots below.