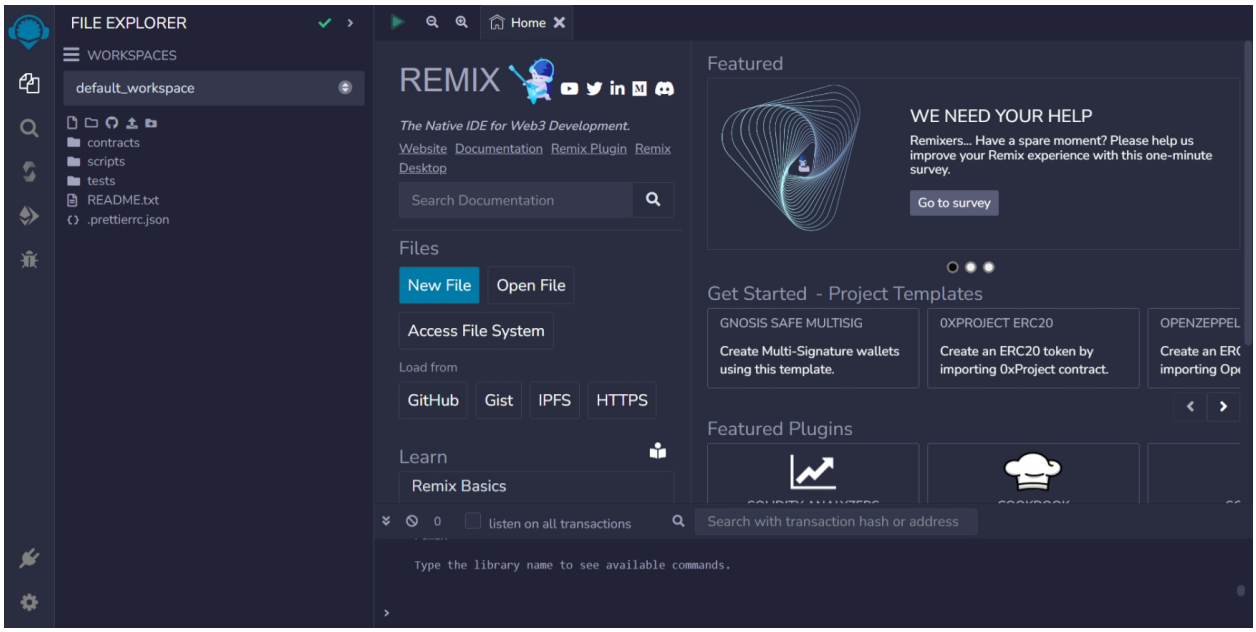
How to Create a CryptoCurrency

Pre-requisites:

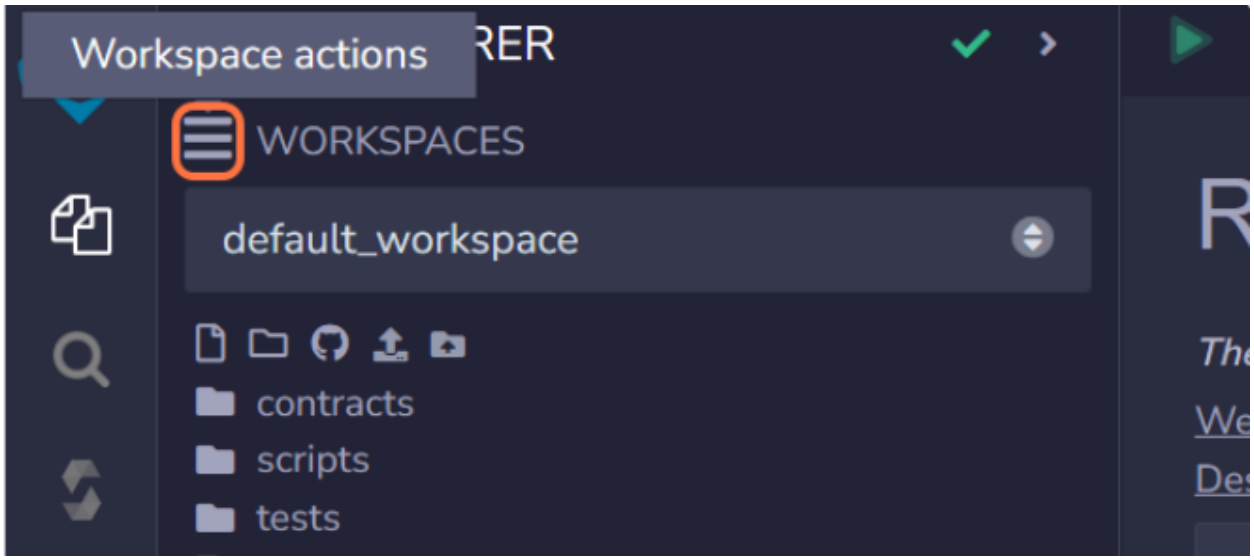
- Basic coding knowledge with Solidity
- Must have your own MetaMask Wallet. (MetaMask Wallet should be connected to Mumbai TestNet).
  If you are unable to meet this prerequisite please see:
  Create a MetaMask Account - Mumbai

Step 1: Coding your first token

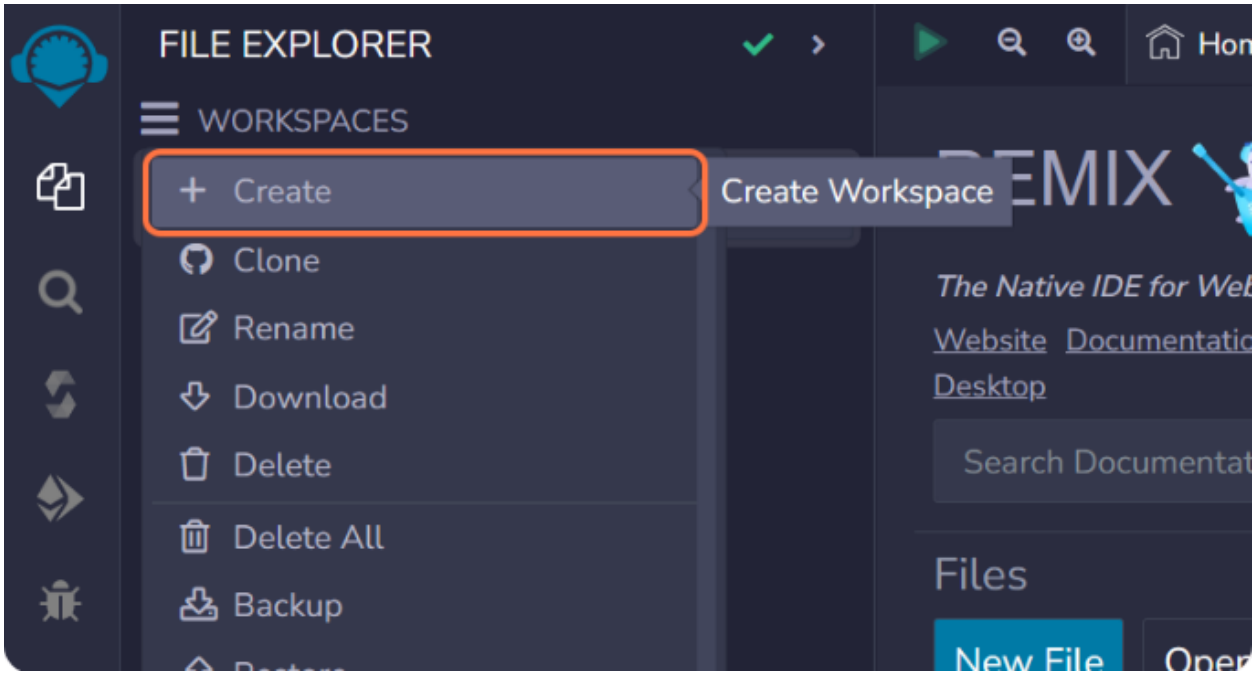
1. Go to https://remix.ethereum.org/



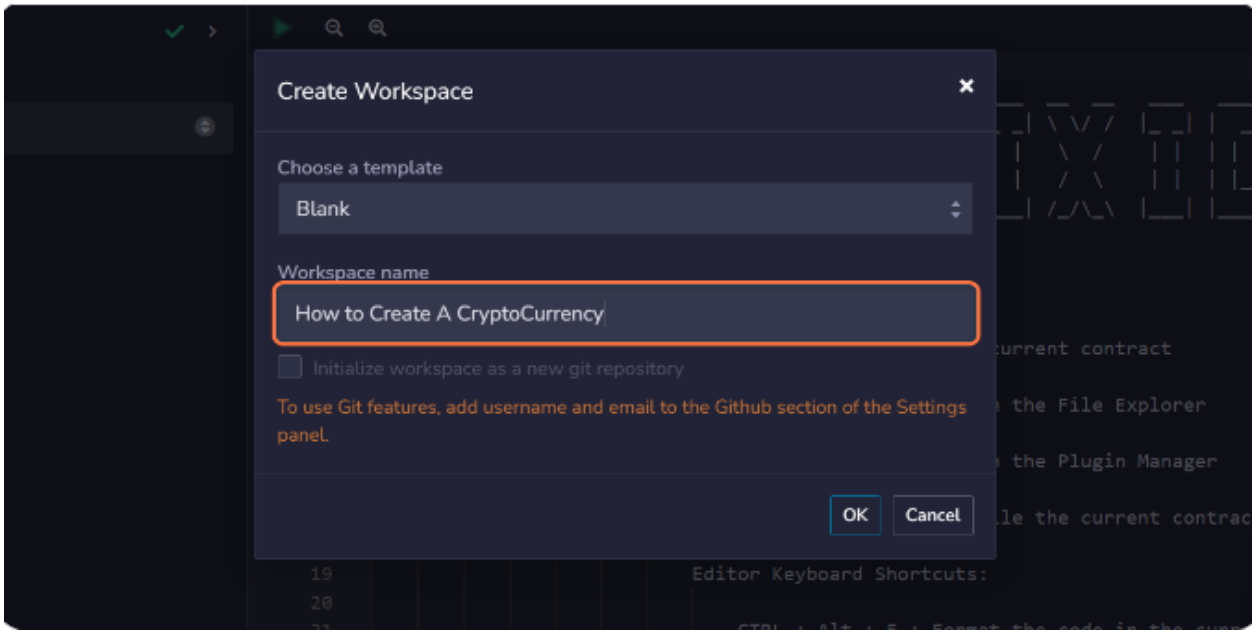
2. Create your own workspace by clicking the Workspace Actions button



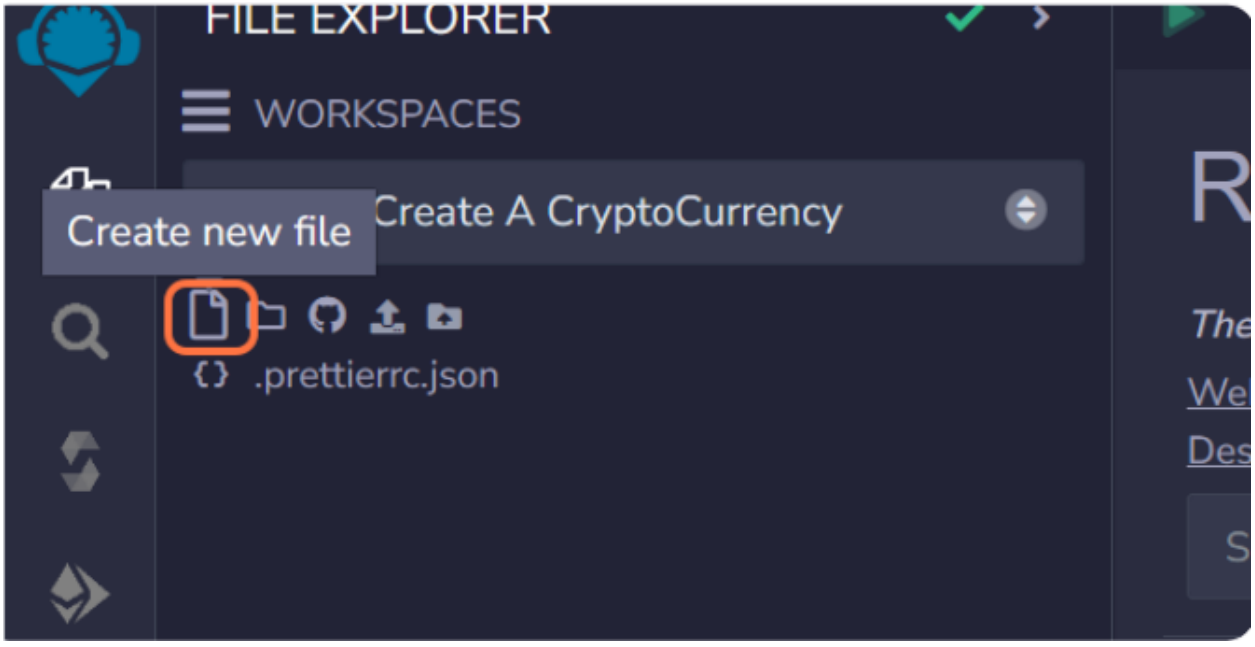
3. Choose the **“Create”** button.



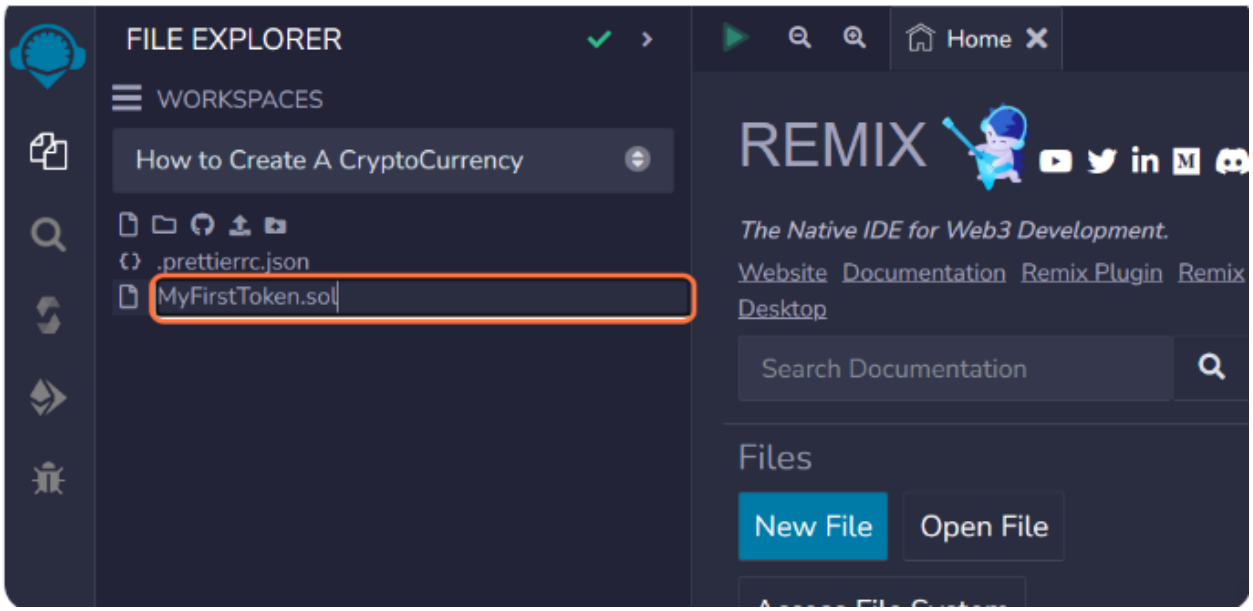
4. Select the **Blank** template and enter the name of your workspace. Then hit the **“OK”** button.



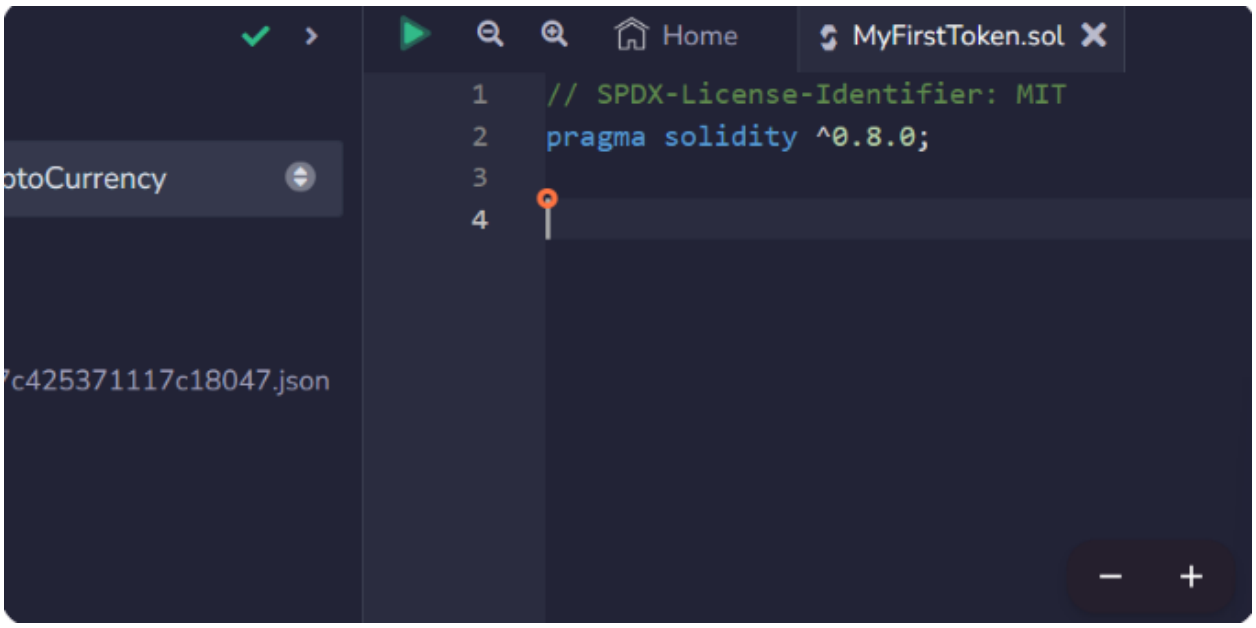
5. Create a new file by clicking the file icon



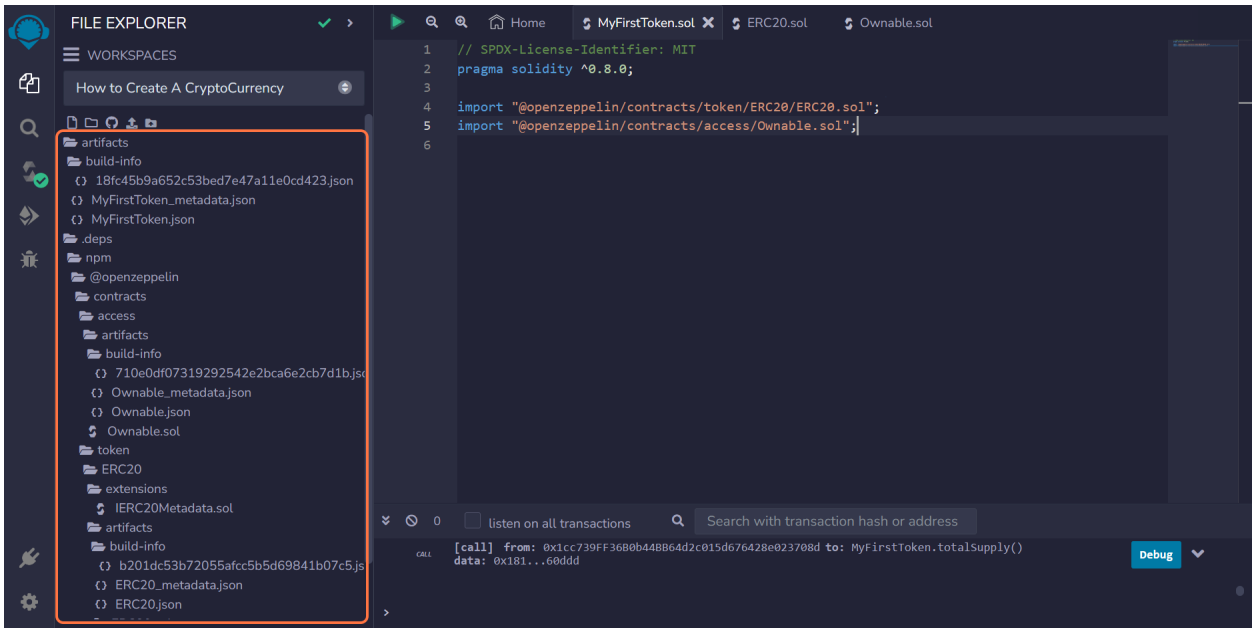
6. Enter your file name. Make sure it sends in **.sol** extension.



7. Write the license identifier and the version. The license identifier here would be `SPDX-License-Identifier: MIT` and the version is `pragma solidity ^0.8.0;`



8. Import **ERC20** and **Ownable**. Notice that there are some files and folders that are added inside our files section.

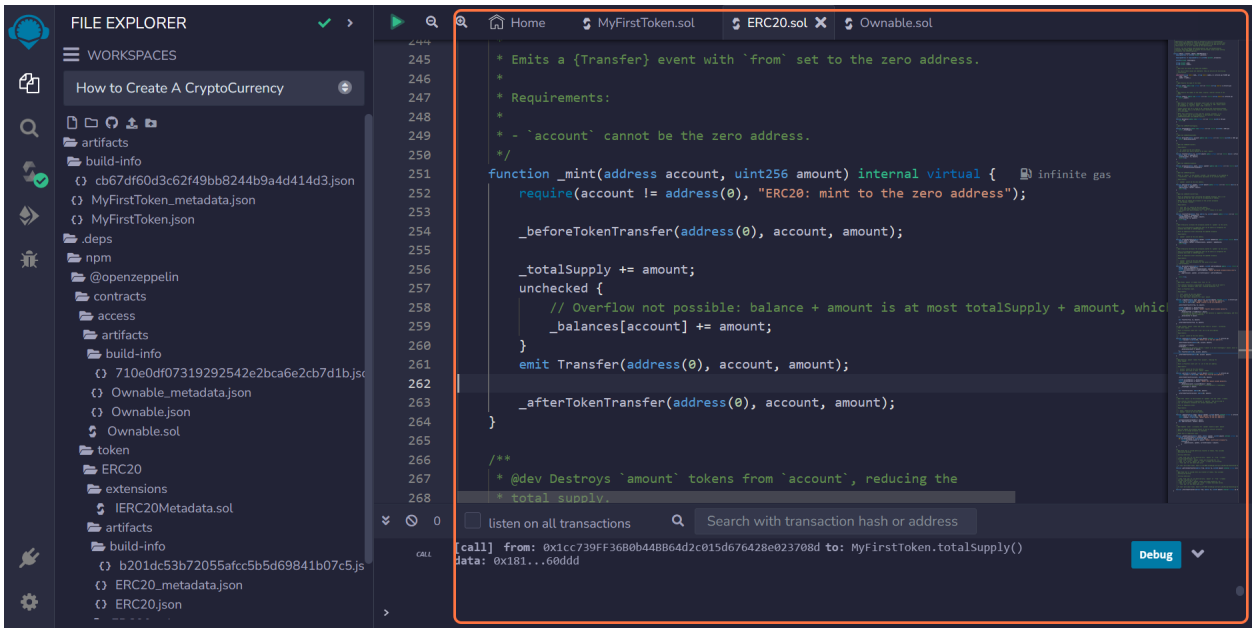


9. Create a contract and inherit ERC20 and Ownable

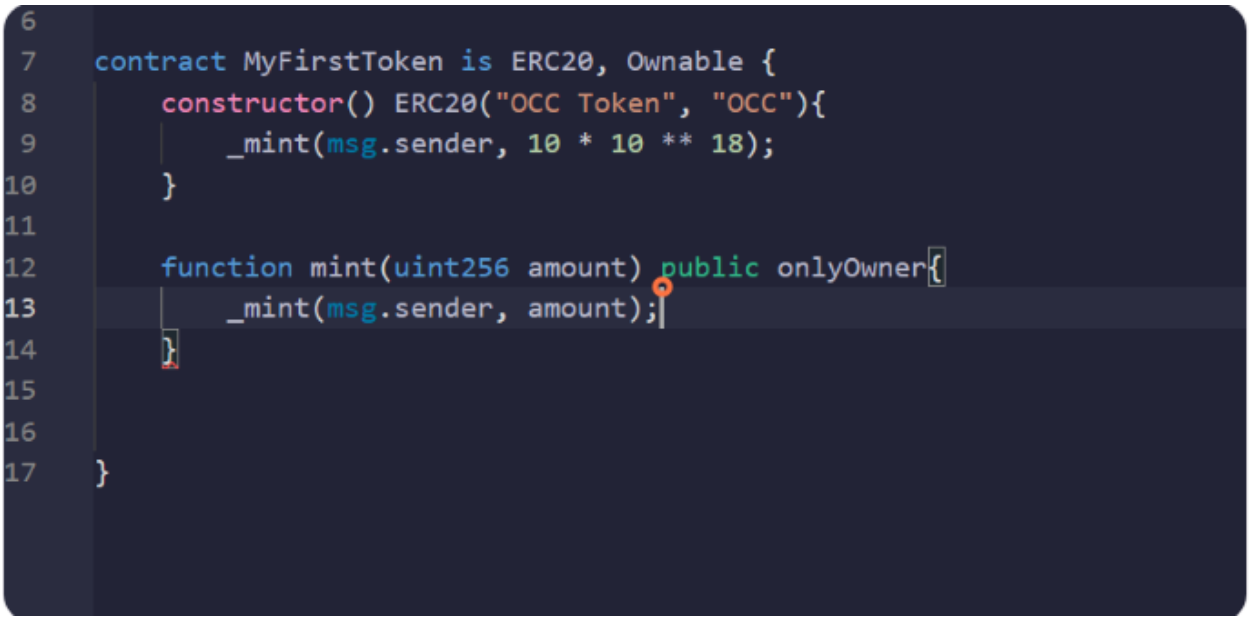
10. Create a constructor that makes a new ERC20 token named “OCC Token” with the symbol “OCC” and mints a certain amount of tokens to the address that deploys the account..

```
3
4   import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
5   import "@openzeppelin/contracts/access/Ownable.sol";
6
7   contract MyFirstToken is ERC20, Ownable {
8       constructor() ERC20("OCC Token", "OCC"){
9           _mint(msg.sender, 10 * 10 ** 18);
10      }
11
12
13  }
```

This is the code for the mint function. To see it, CTRL + Click _mint()



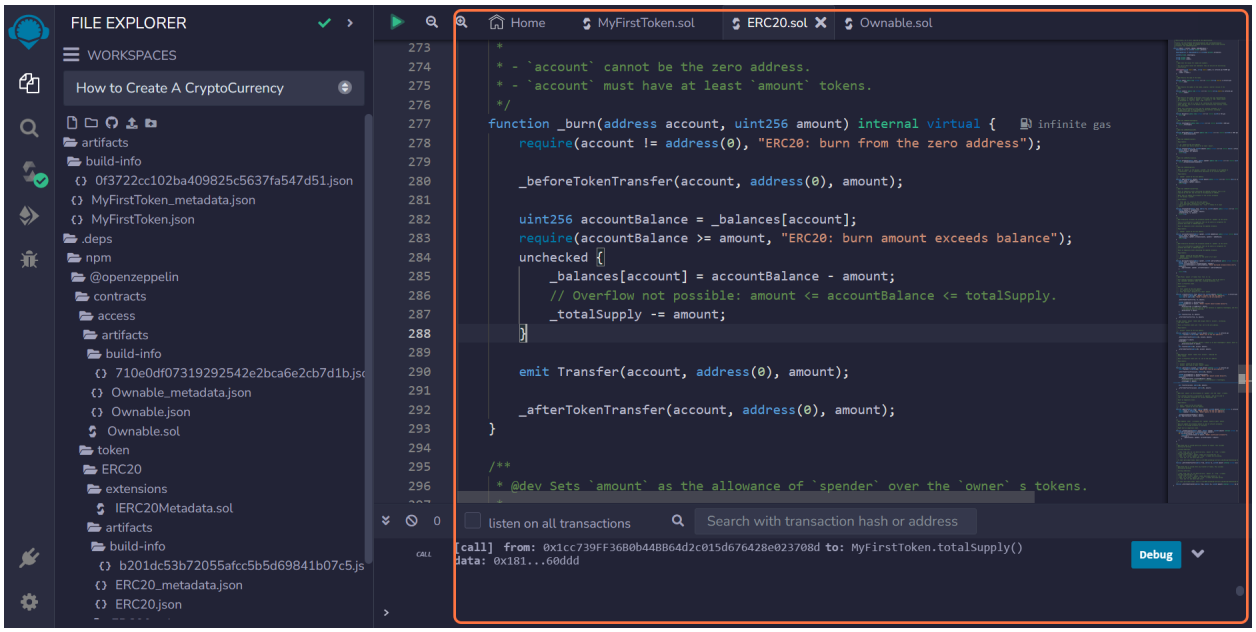
11. Create a **mint()** function that allows the owner of the contract to mint a specified amount of tokens.
    a. This takes an unsigned integer which represents the amount of tokens as a parameter and the function is public.
    b. This function will be modified by the **onlyOwner** modifier.
    c. This mint function will call the **_mint()** function of the ERC20 library. It will take the sender's address, and the amount from the **mint()** function.



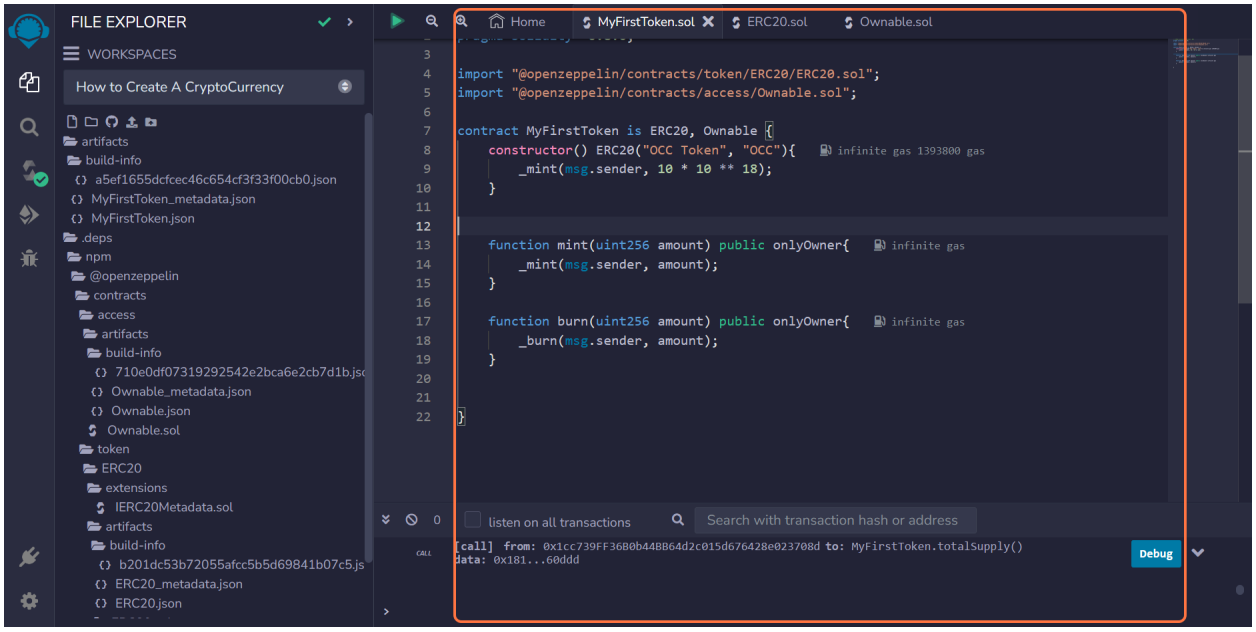
12. Create a **burn()** function that allows the owner of the contract to mint a specified amount of tokens.
    a. This takes an unsigned integer which represents the amount of tokens as a parameter and the function is public.
    b. This function will be modified by the **onlyOwner** modifier.
    c. This mint function will call the **_burn()** function of the ERC20 library. It will take the sender's address, and the amount from the **burn()** function.

```
        _mint(msg.sender, 10 * 10 ** 18);
    }

    function mint(uint256 amount) public onlyOwner{
        _mint(msg.sender, amount);
    }

    function burn(uint256 amount) public onlyOwner{
        _burn(msg.sender, amount);
    }


}
```

*This is the code for **_burn() function**. To see it, CTRL + _burn()*

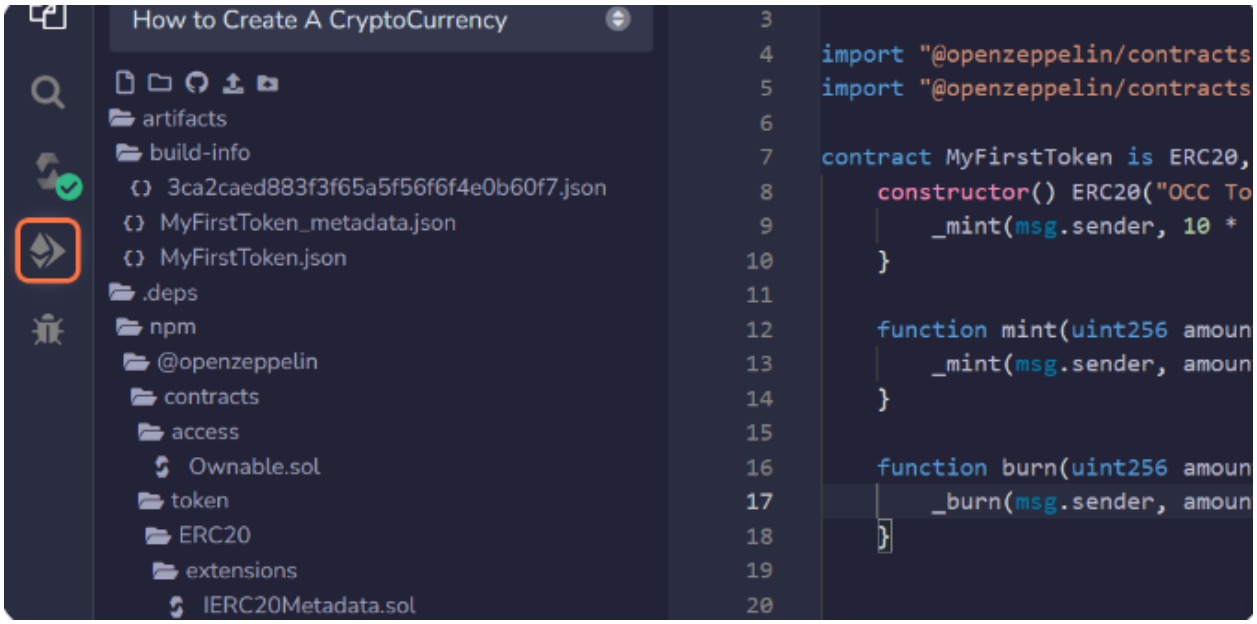


13. This is how the final code should look like

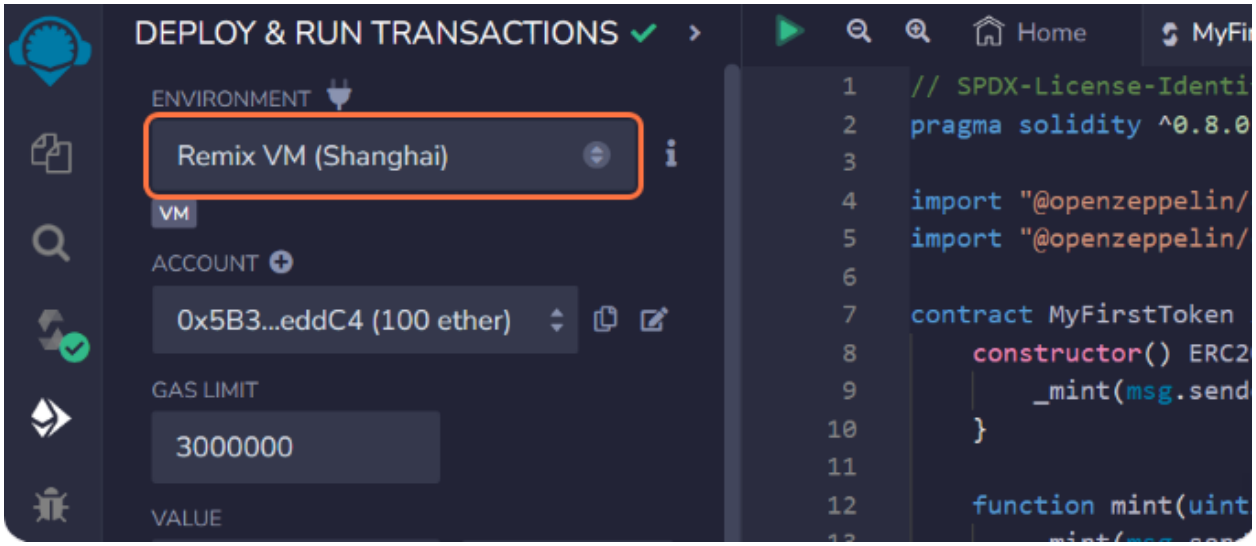


Step 2: Deploying and Testing your first token.

1. Go to the right side navigation bar And click on the **Deploy and Run Transactions Button.**

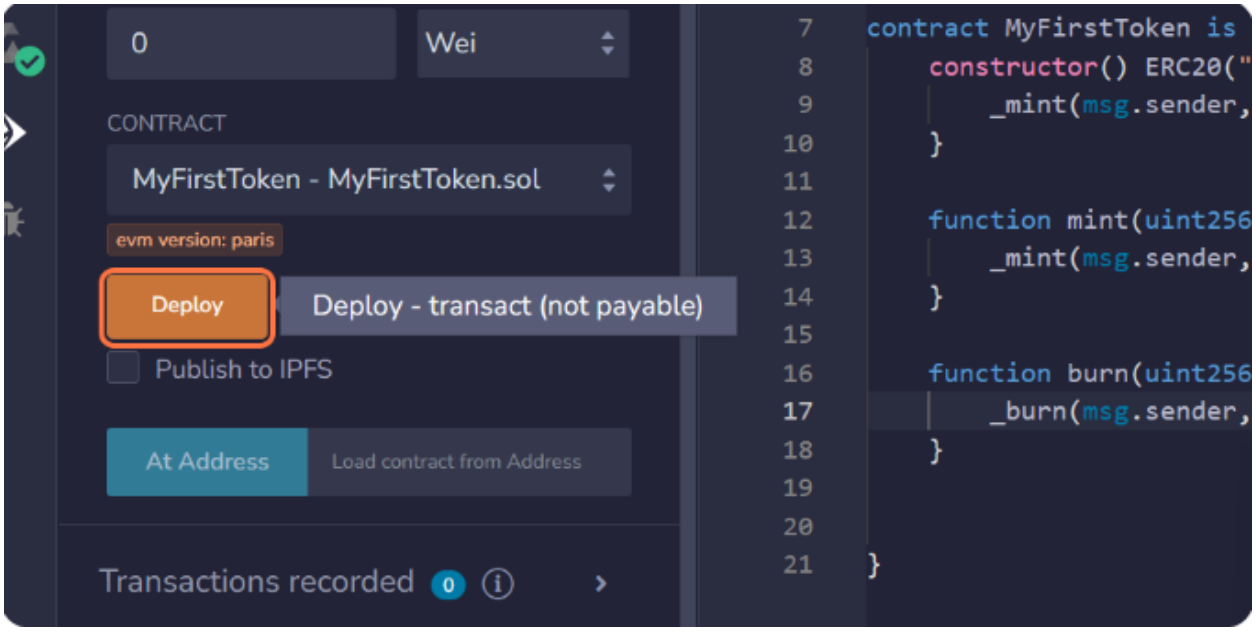


2. Click the box below the word ENVIRONMENT.

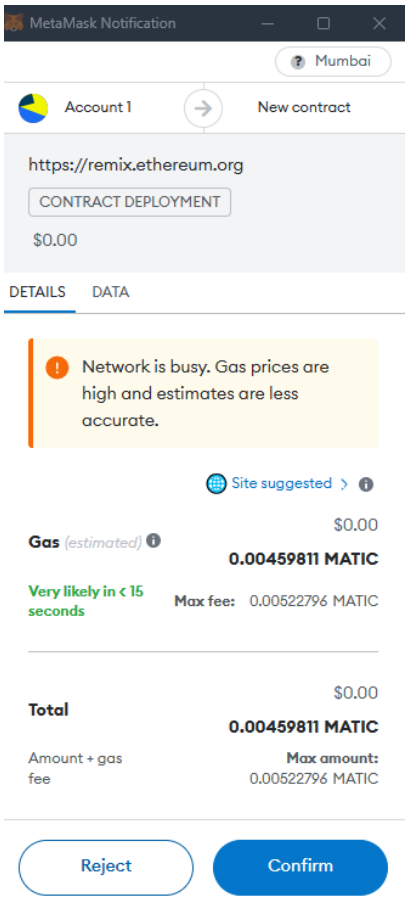


3. Choose **Injected Provider - MetaMask.**

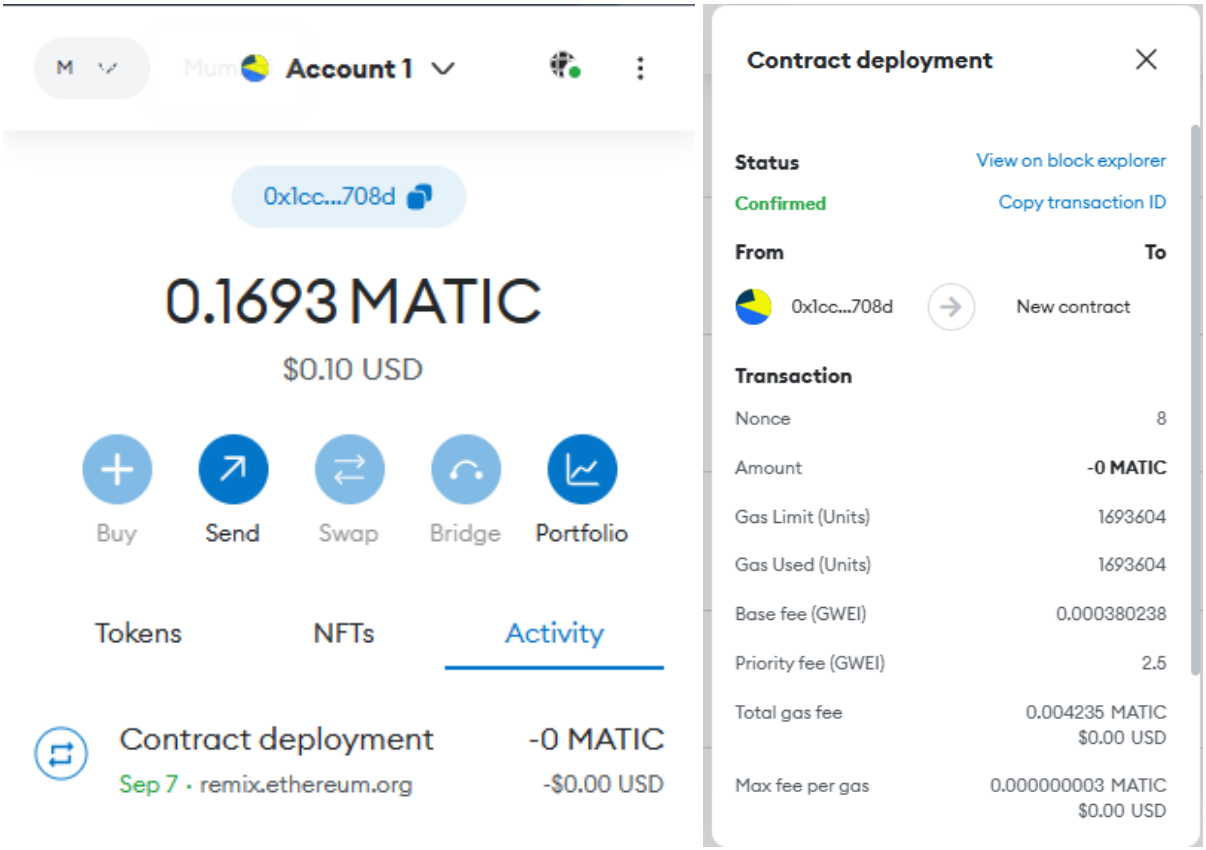
4. Click the **“Deploy”** button.



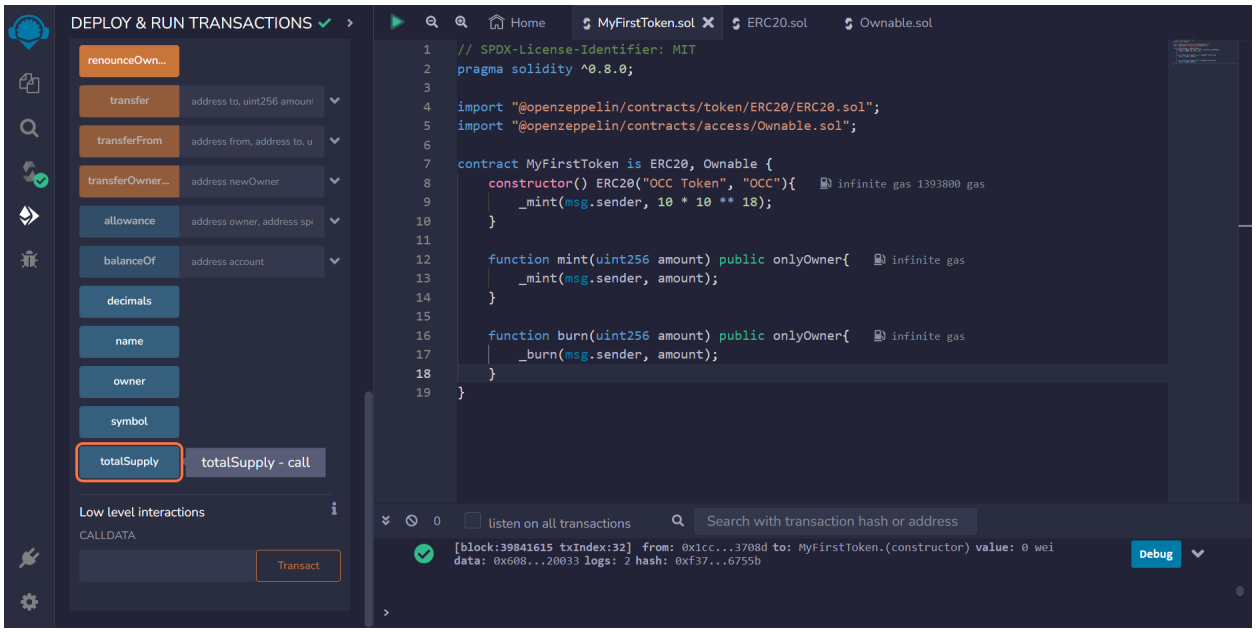
5. A MetaMask prompt will open. This tells about the Contract Deployment. Click **“Confirm”**.



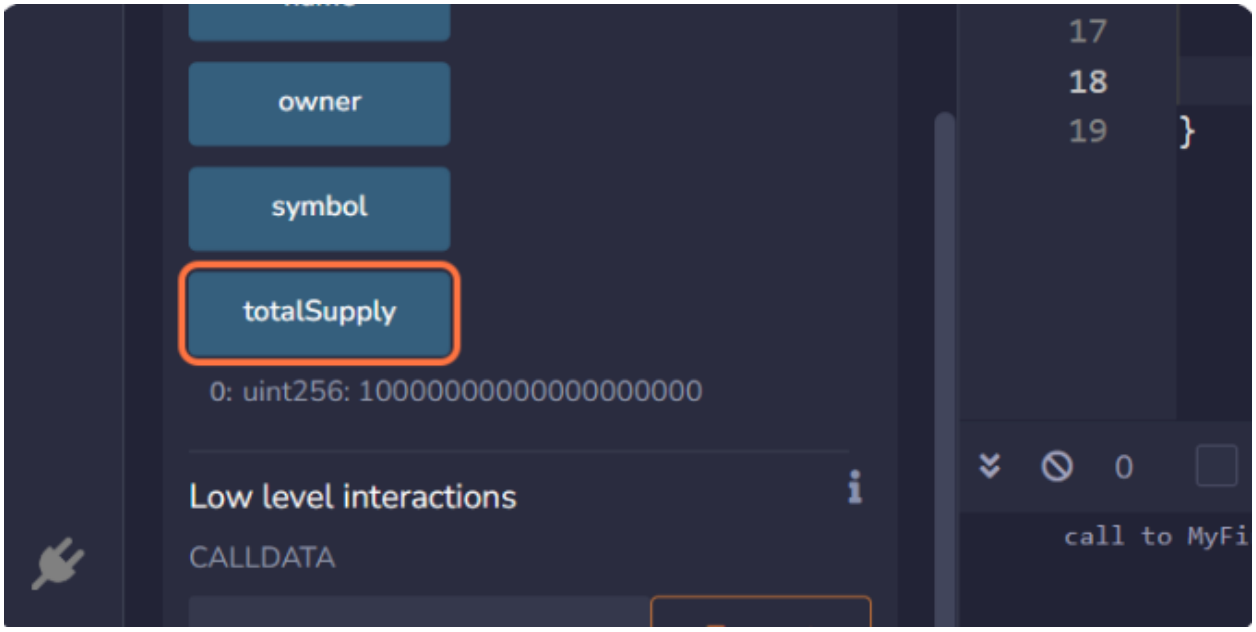
6. Once you confirm, you will see inside your MetaMask Wallet a Contract Deployment Activity.



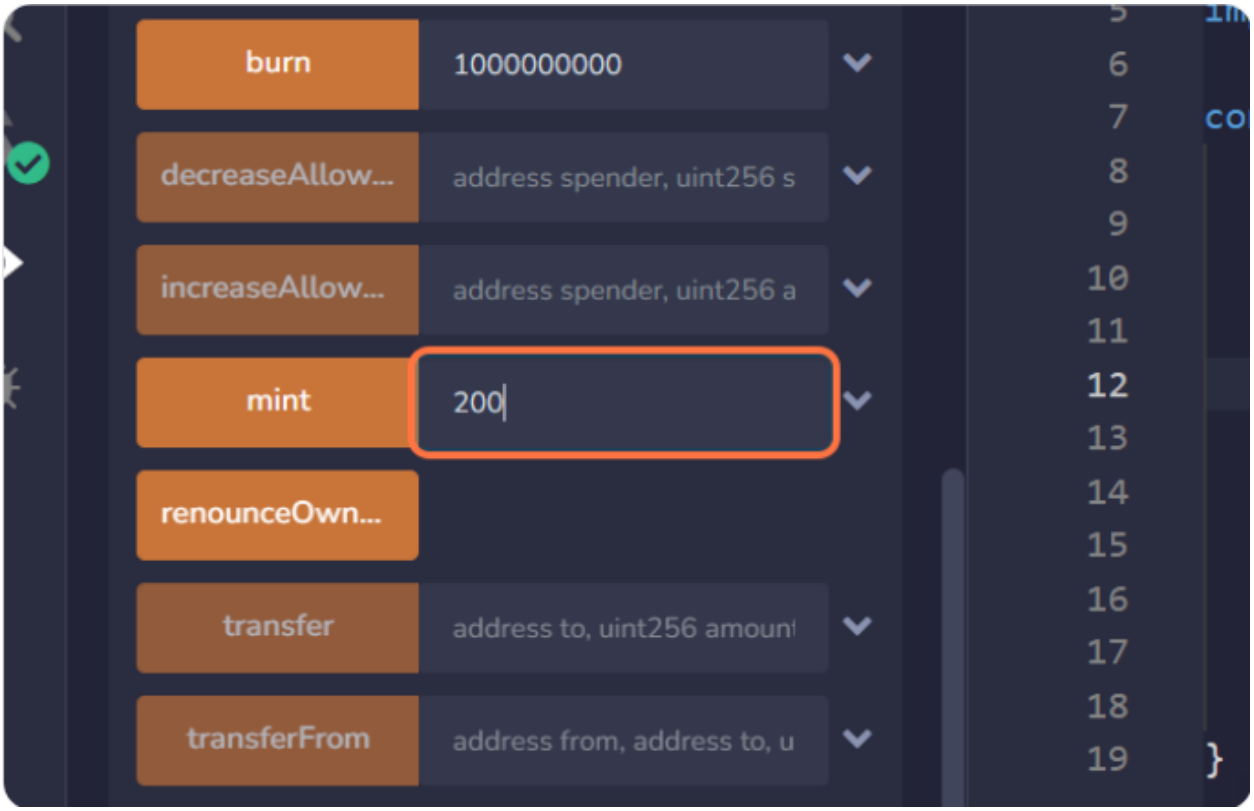
7. Inside **Deployed Contracts**,you will see your contract name and its contract ID. Click the dropdown arrow icon beside it. Inside it you will see all the functions and variables of your contract.



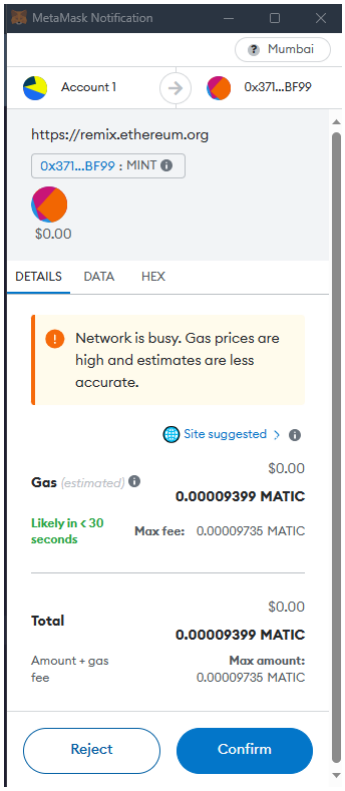
8. Click on the **“totalSupply”** button. This will tell us the amount we have.



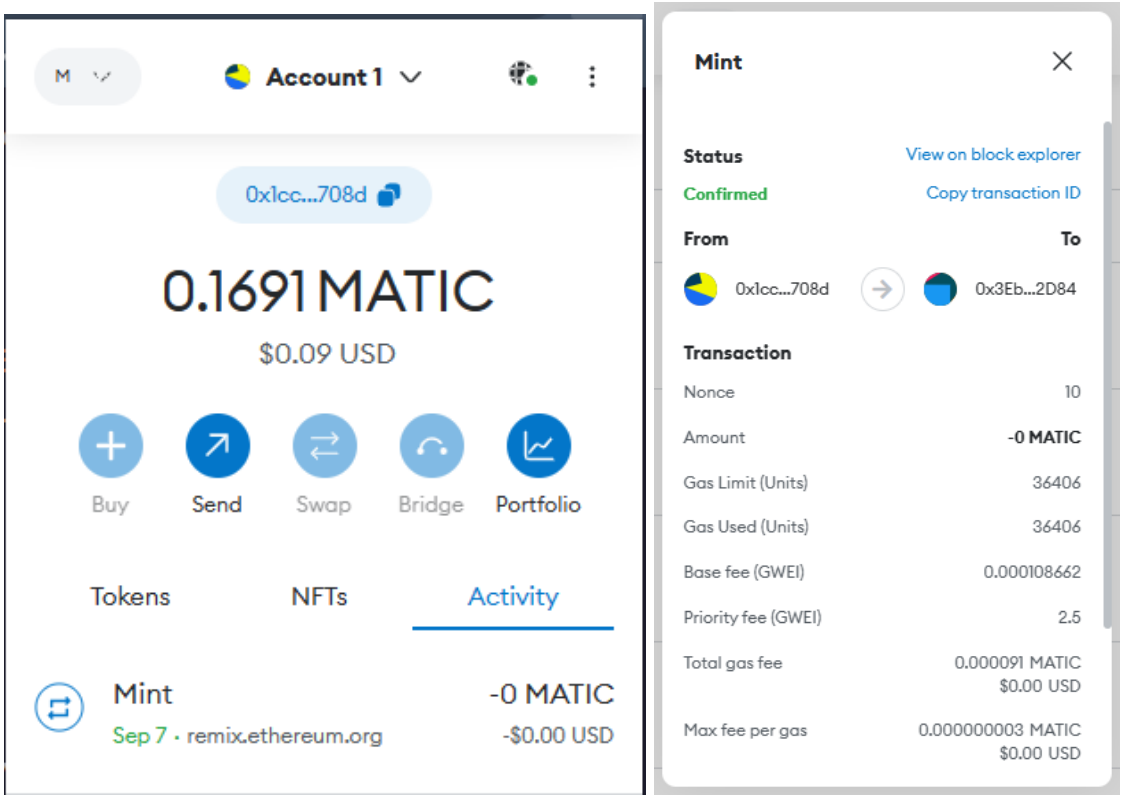
9. Enter an amount on the input field next to the **“mint”** button. Then click on “**mint**”



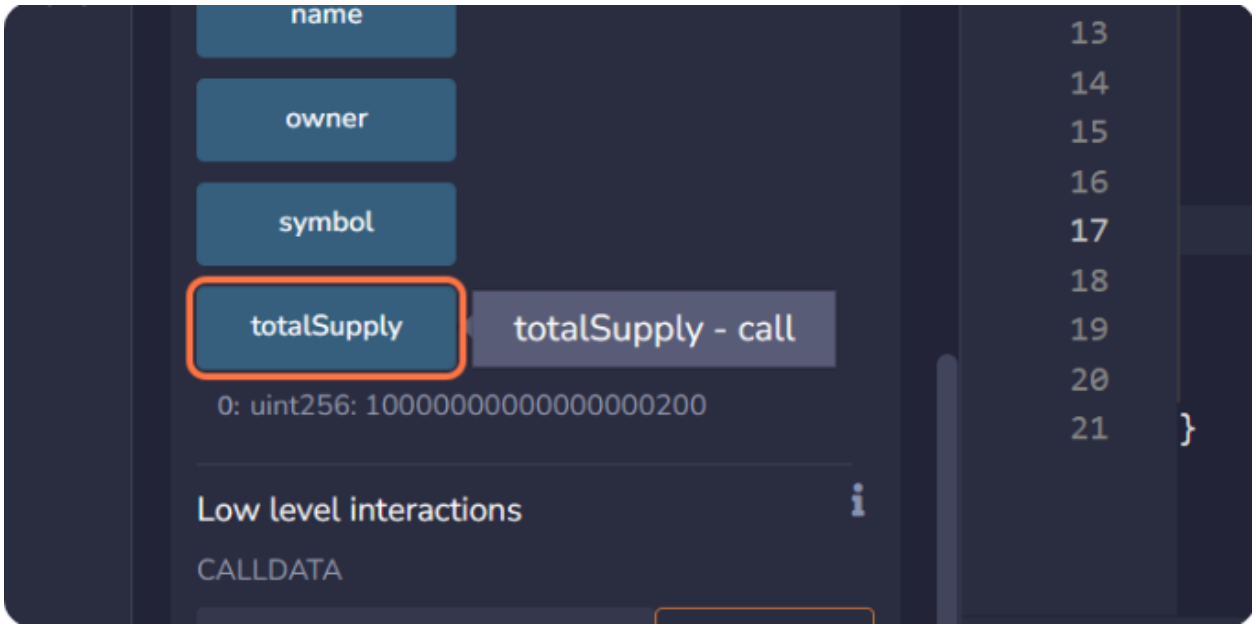
10. MetaMask will show a prompt. Click on the **“Confirm”** button.



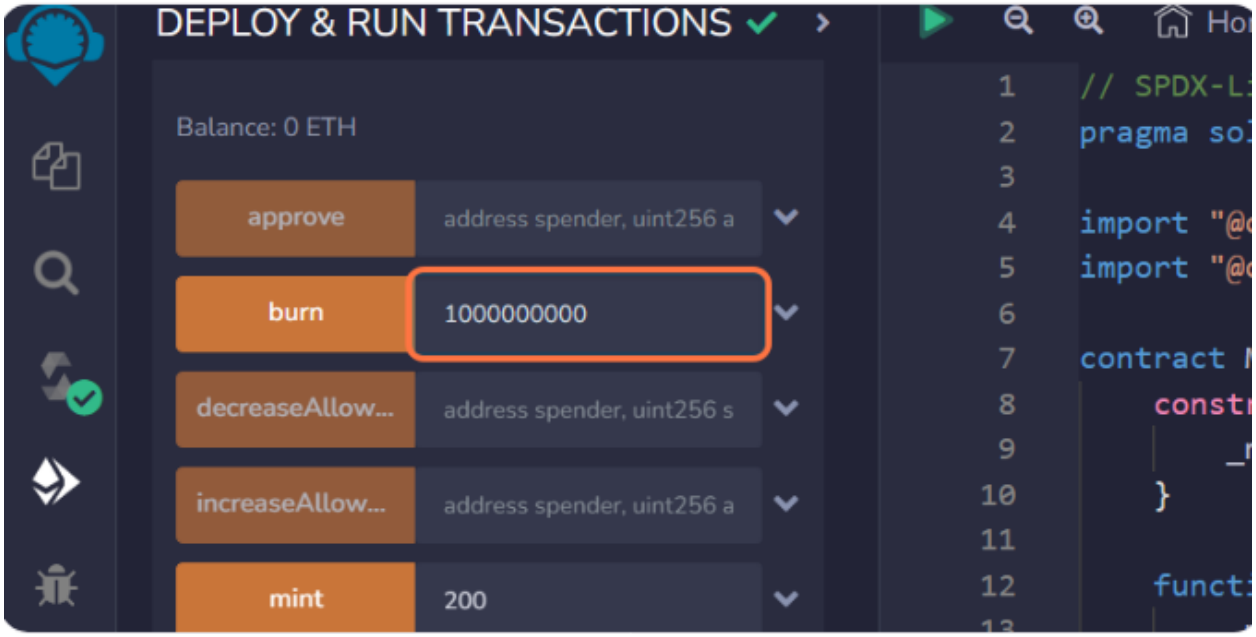
11. Once you have confirmed, you will see inside your MetaMask Wallet a Mint Activity.



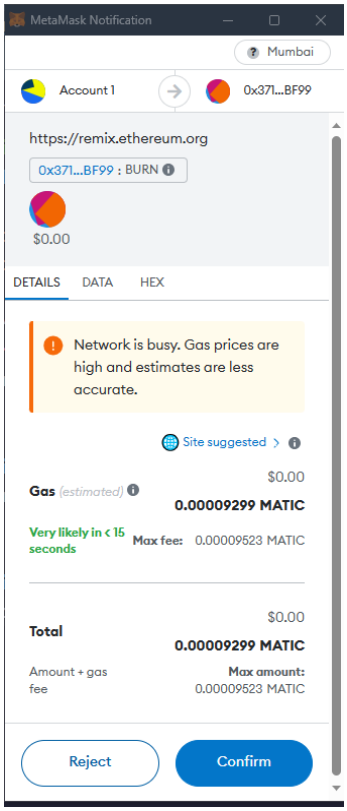
12. Click the **“totalSupply”** button. Notice that we have added the amount we entered for mint.



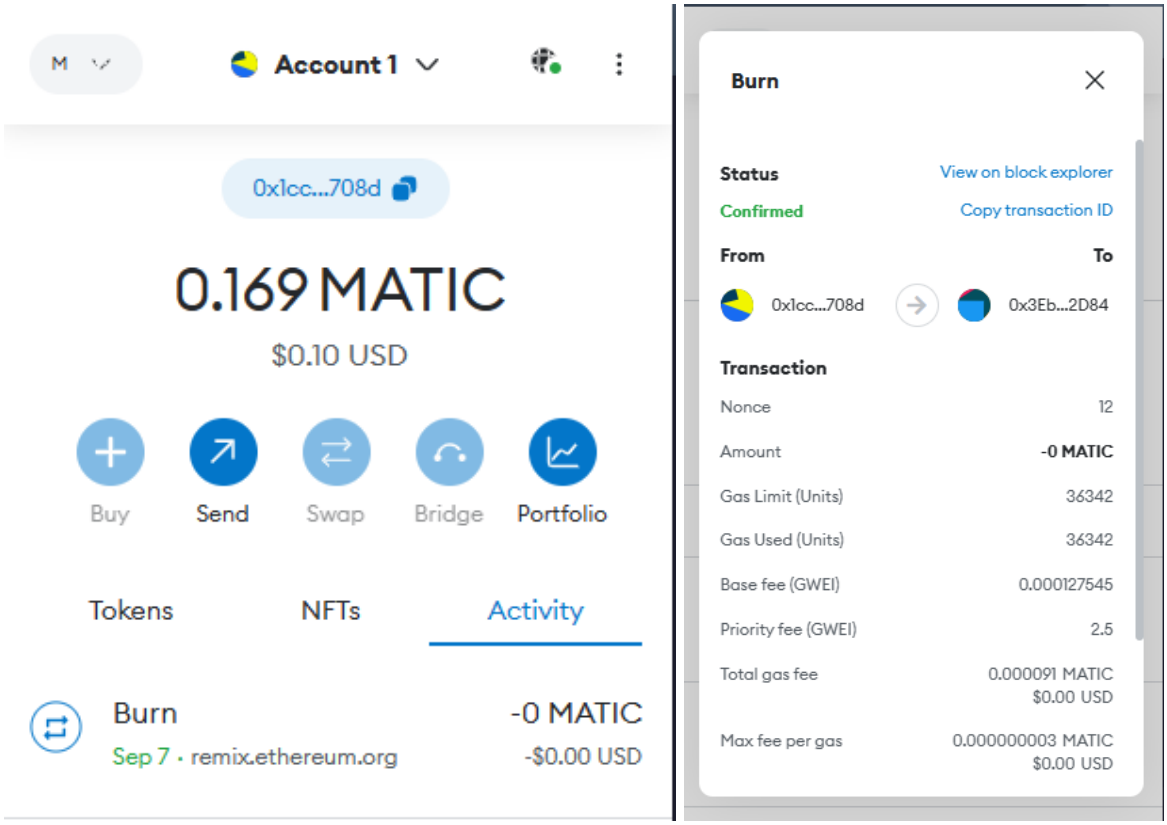
13. Enter an amount next to the **“burn”** button. Click **“burn”**.



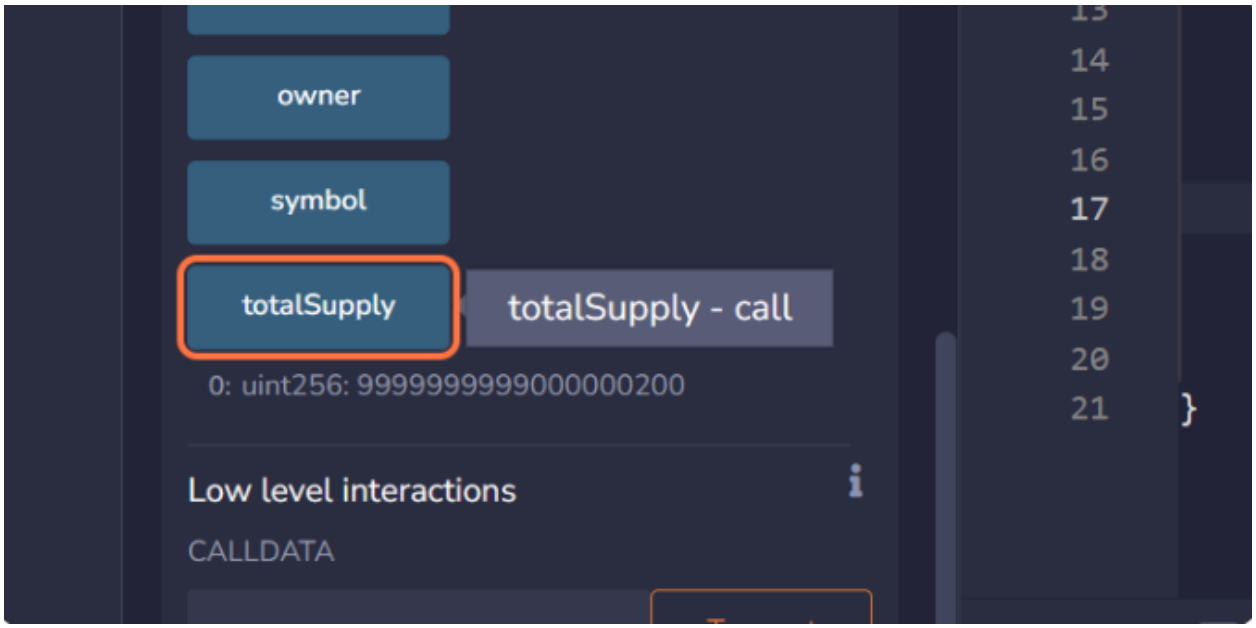
14. MetaMask will show a prompt. Click on the **“Confirm”** button.



15. Once you have confirmed, you will see inside your MetaMask Wallet a Burn Activity.

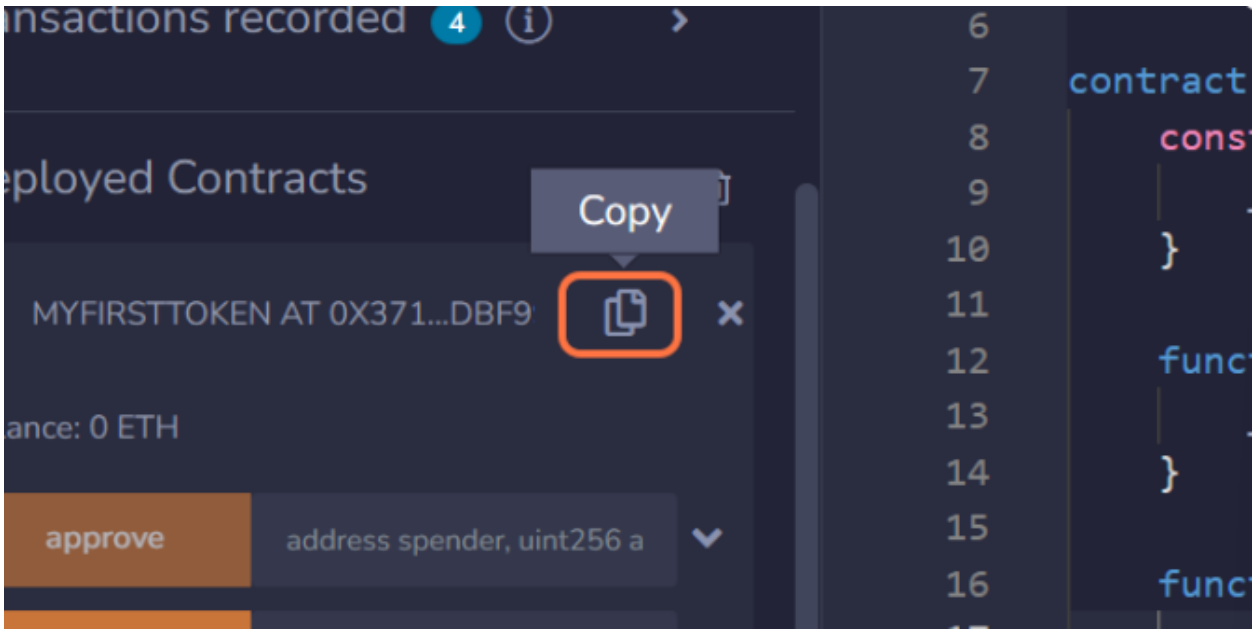


16. Click the **“totalSupply”** button. Notice that we have deducted the amount we entered for burn.

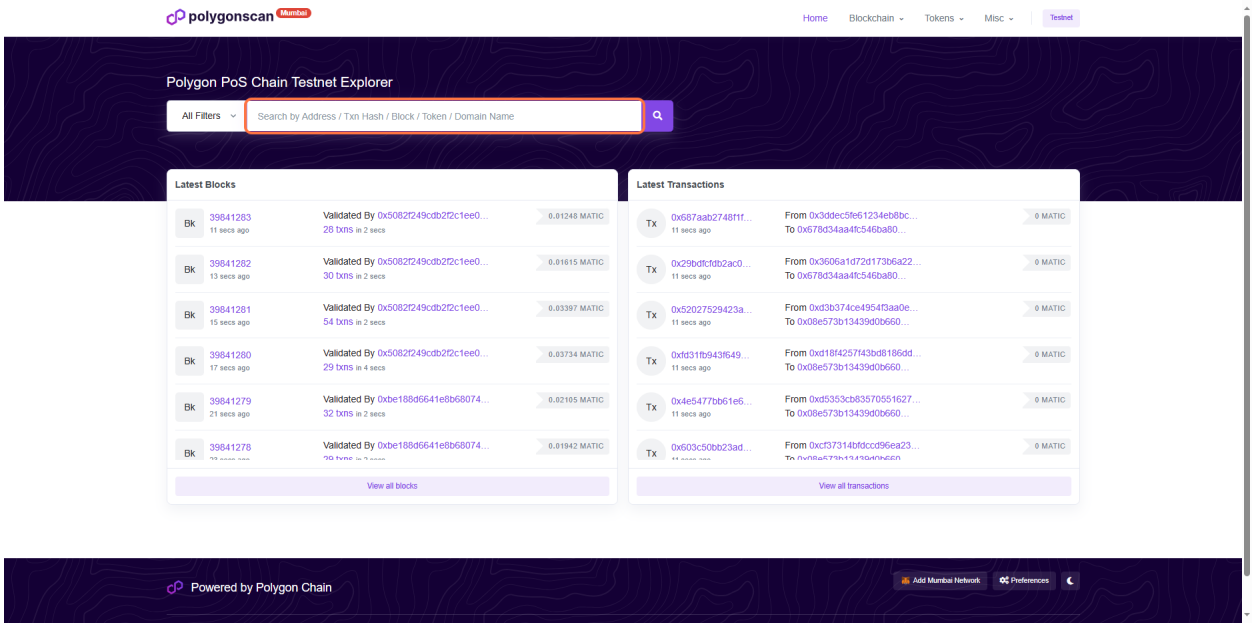


Step 3: Using Polygon Scan to Check

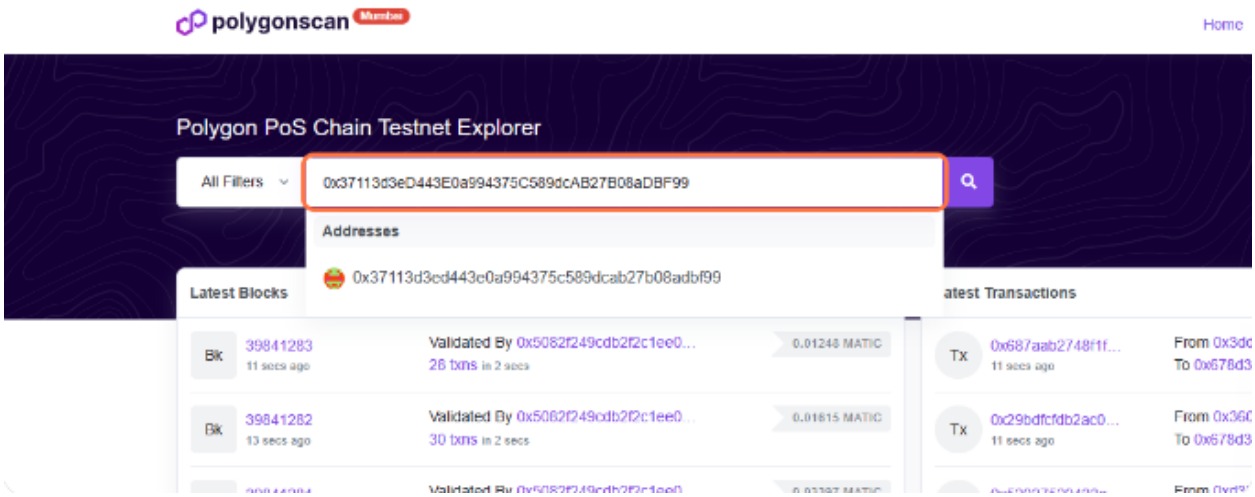
1. Click the **copy icon** on our deployed contract. This will copy the contract ID.



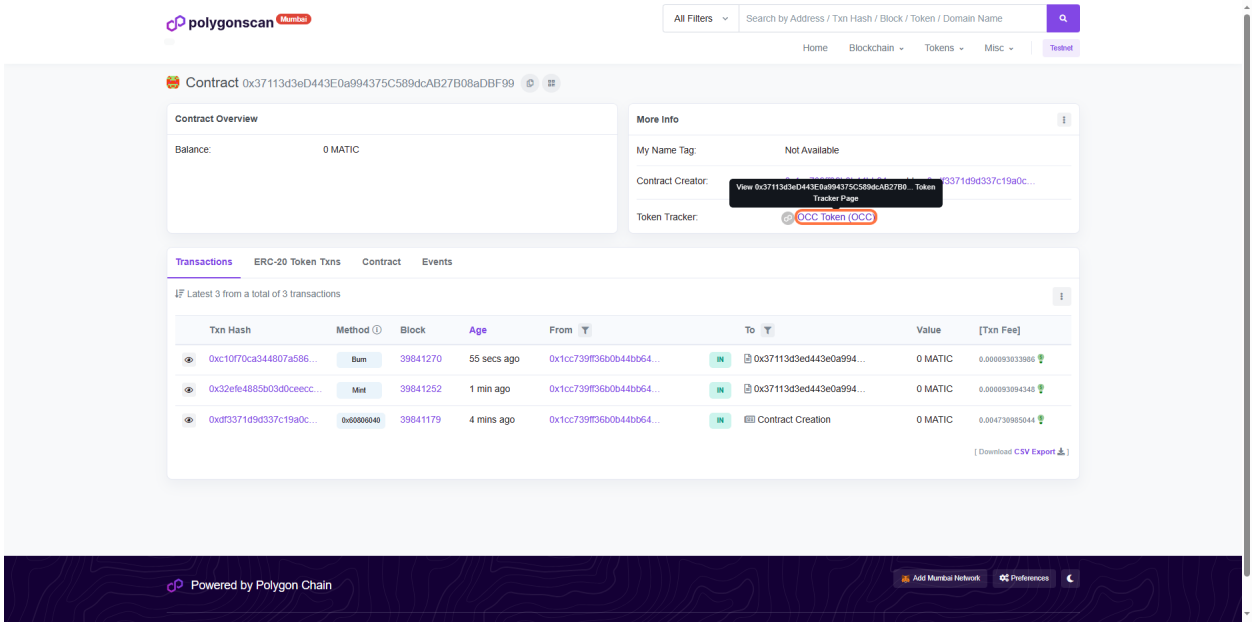
2. Got to this website https://mumbai.polygonscan.com/



3. Paste your contract ID on the search field. Then click the **search icon button**



4. You will now be able to see the transactions and information of your contract. Click on the **“OCC Token(OCC)”** link in the Token Tracker Row inside the More Info table.



5. This will open the token tracker page. You have now finally created your own CryptoCurrency!