Soulbound Token

overview

This was a project assigned to me by Metacrafter in their course "ETH Proof for Beginners," where I have to create a Solidity smart contract with basic functions for minting and burning a token.

Requirements

- Public Variables: The contract includes public variables to store details about the token: Token Name, Token Abbreviation, and Total Supply.

- Address to Balance Mapping: A mapping to track the balances of different addresses.

- Mint Function: A function to mint new tokens, increasing the total supply and the balance of a specified address.

- Burn Function: A function to burn tokens, decreasing the total supply and the balance of a specified address. It includes a check to ensure the address has

  sufficient balance.

Learning Outcomes

- solidity basics

- contract structure

- how to use and create functions

- how to use mapping

- how to use access modifiers

Conclusion

This project provided a hands-on introduction to writing a simple token contract in Solidity and taught how to write solidity smart contracts.

[10:48 PM, 6/13/2024] Jannat: // SPDX-License-Identifier: MIT

pragma solidity ^0.8.25;

/*

REQUIREMENTS

1. Your contract will have public variables that store the details about your coin (Token Name, Token Abbrv., Total Supply)

2. Your contract will have a mapping of addresses to balances (address => uint)

3. You will have a mint function that takes two parameters: an address and a value.

The function then increases the total supply by that number and increases the balance

of the “sender” address by that amount

4. Your contract will have a burn function, which works the opposite of the mint function, as it will destroy tokens.

It will take an address and value just like the mint functions. It will then deduct the value from the total supply

and from the balance of the “sender”.

5. Lastly, your burn function should have conditionals to make sure the balance of "sender" is greater than or equal

to the amount that is supposed to be burned.

*/

```
contract MyToken {

    // public variables here
    string constant public TOKENNAME = "soulbond";
    string constant public TOKENABBRV = "sol";
    uint public totalSupply = 0;
    // mapping variable here
    mapping(address => uint) public balances;

    //modifiers
    modifier enoughBalnce(address sender, uint value){
        require(balances[sender] >= value,"sender balance is not enough");
        _;
```

```
    }

    // mint function
    function mint (address sender , uint value) public {
        totalSupply += value;
        balances[sender] += value;
    }

    // burn function
     function burn (address sender , uint value) public enoughBalnce(sender, value) {
            totalSupply -= value;
            balances[sender] -= value;

    }

}
```