// SPDX-License-Identifier: MIT

pragma solidity ^0.8.2;

import "@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol";

import "@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC20BurnableUpgradeable.sol";

import "@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol";

import "@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol";

import "@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol";

contract DOGEROID is Initializable, ERC20Upgradeable, ERC20BurnableUpgradeable, PausableUpgradeable, OwnableUpgradeable {

function initialize() initializer public {

\_\_ERC20\_init("DOGEROID", "DOGD");

\_\_ERC20Burnable\_init();

\_\_Pausable\_init();

\_\_Ownable\_init();

\_mint(msg.sender, 100000000000 \* 10 \*\* decimals());

}

function pause() public onlyOwner {

\_pause();

}

function unpause() public onlyOwner {

\_unpause();

}

function mint(address to, uint256 amount) public onlyOwner {

\_mint(to, amount);

}

function \_beforeTokenTransfer(address from, address to, uint256 amount)

internal

whenNotPaused

override

{

super.\_beforeTokenTransfer(from, to, amount);

}

}