

Problem Statement: Custom Token with Burn & Mint Functionality

Overview:

Develop a custom ERC20 token with advanced minting and burning functionalities using OpenZeppelin libraries. The contract should include mechanisms for minting tokens by the contract owner and burning tokens by any user, ensuring dynamic control over token supply.

Objectives:

- Deploy a secure ERC20 token contract.
- Ensure only the contract owner can mint new tokens.
- Allow any user to burn tokens from their own balance.
- Accurately maintain the total supply of tokens.

Features:

- View current total token supply.
- Mint tokens to a specific address (owner-only).
- Burn tokens from the caller's balance (any user).
- Constructor to set token name, symbol, and initial supply.

Requirements:

1. Inherit from OpenZeppelin's ERC20, ERC20Burnable, and Ownable contracts.
2. The owner (deployer) should be able to mint tokens to any account.
3. Any user should be able to burn tokens they own.
4. Use access control to restrict minting functionality.
5. Include proper events for minting and burning.

Technologies:

- Solidity $\geq 0.8.0$
- OpenZeppelin Contracts
- Remix/Hardhat for testing and deployment

Bonus (Optional):

- Add a cap to limit the maximum supply.
- Implement pause functionality to halt transfers if needed.

Outcome:

This project will enhance understanding of custom token development, access control, and supply management on the Ethereum blockchain using secure and standardized OpenZeppelin contracts.