CoinBuck Token Contract

Project Overview

BUCK token contract contains the Solidity code for an ERC-20 compliant token on the BSC blockchain. The ERC-20 standard defines a set of rules for tokens to implement on Ethereum, ensuring compatibility with various wallets, exchanges, and other contracts. BUCK tokens are minted at the time of contract deployment on the owners address. It’s a fixed supply contract with no option to mint new tokens after contract deployment. Burn function help community to burn the token supply as per the market sentiments/Utility of the token.

1. Functional Requirements

1.1. Roles

BUCK contract has two roles:

- **Owner** - All control of the BUCK token contract belongs to a Contract Owner. A Contract Owner can block/Unblock(Blacklist) users, if needed. Mint & hold all the minted tokens at the time of contract deployment
- **User** - User can check total supply, ticker, decimal value using read function and also able to use write functions for transfer tokens.

1.2. Features

BUCK Token contract has the following features:

- Fixed BUCK Token Supply
- Anyone can burn their BUCK token using function BURN (All)
- Owner can blacklist any of the address
- Owner can remove addresses from blacklist
- Owner can enable/disable blacklisting\
- Anyone can trasfer BUCK token from one to another address using transfer function

1.3. Use Cases

- BUCK token is 100% a community driven token with utility in future/upcoming projects of the ecosystem where users can pay platform fee, earn BUCK token against there activity on platform
- User can trade BUCK token on multiple Centralized & Decentralized exchanges
- User can store & hold BUCK tokens on different custodial & non-custodial wallets.

2. Technical Requirements

This project has been developed with **Solidity** language, using **Truffle** as a development environment. **Javascript** is the selected language for testing and scripting.
In addition, **OpenZeppelin**’s libraries are used in the project. All information about the contracts library and how to install it can be found in their **GitHub**.

In the project folder, the following structure is found:

```
├─ .gitignore
├─ README.md
├─ contracts
│   ├─ Migrations.sol
│   ├─ access
│   │   └─ Ownable.sol
│   ├─ buck.sol
│   ├─ token
│   │   ├─ IERC20.sol
│   │   └─ IERC20Metadata.sol
│   └─ utils
│      └─ Context.sol
├─ migrations
│   └─ 1_initial_migration.js
├─ package-lock.json
├─ package.json
├─ scripts
│   ├─ flatten.sh
│   └─ test.sh
├─ test
│   └─ buck.test.js
└─ truffle-config.js
```

Start with **README.md** to find all basic information about project structure and scripts that are required to test and deploy the contracts.

Inside the **./contracts** folder, buck.sol contains the Buck Token smart contract which imports Interface IERC20 ,IERC20Metadata.sol, context.sol present inside utils and Ownable.sol present inside access

Inside the ./**test** folder buck.test.js contains test cases for the different methods of buck token contract in javascript
Test can be run by Changing directory to scripts and run command sh test.sh
The project configuration can be found in truffle-config.js

2.1. Contract Information

This section contains detailed information about the contracts used in the project.

2.1.1. Buck.sol

Buck.sol is the main token contract through which owner will get total supply of BUCK token in the owner address, buck.sol contract holds multiple functions that are required such as transfer, allowance, disableBlacklisting(), enableBlacklisting(), blacklistAddress(address user), removeBlacklistedAddress(address user) etc.

2.1.1.1. Events

BUCK Token Contract has the following events:

- **blacklistDisable**:Emitted when blacklisted is disabled for whole contract
- **blacklistEnable**: Emitted when blacklisted is enabled for whole contract
- **addressBlacklisted**:Emitted when owner blacklist some address
- **addressWhitelisted**: Emitted when owner removes blacklisted address
- **ownershipTransferred**:Emitted when contract ownership is transferred
- **Transfer**:Emitted when tokens are transferred
- **Approval**:Emitted when approval is give to some other user

2.1.1.2. Modifiers

BUCK Token Contract has the following modifiers:

- **OnlyOwner** : only contract owner can call those functions which has onlyOwner modifier

2.1.1.3. Functions

BUCK Token Contract has the following functions:

- **BalanceOf**: use to check BUCK token balance of an address

- **Transfer**: use to transfer BUCK token from one address to another

- **Allowance**: Returns remaining number of token that spender will be allowed to spend on behalf of owner through transferFrom This is zero by default.

- **DisableBlacklisting()**:Disable address blacklisting

- **EnableBlacklisting():** Enables address blacklisting

- **BlacklistAddress**: use to blacklist a wallet address by the Owner

- **RemoveBlacklistedAddress**:use to remove a address from blacklisted addresses

- **Approve**: Sets amount as allowance of spender over the callers tokens

- **TransferFrom**:Moves amount tokens from **from** to **to** using allowance mechanism.

- **Function increaseAllowance**:Increase allowance granted to spender by caller
- **Function decreaseAllowance**:Decrease allowance granted to spender by caller
- **Function burn(uint256 amount) external**: Destroys token from account reducing total supply .