DESSERT FINANCE

# CookieSale (COOKIE)

## BEP-20 Audit

Performed at block **22231422**

**PERFORMED BY DESSERT FINANCE**

**PROXY CONTRACT ADDRESS:** 0X6D342877FC199C629F49A5C6C521C297B15BC92D

**IMPLEMENTATION CONTRACT ADDRESS:** 0XBE05CC533CE058787ED9A520B41F5F364E0A69EE

# INITIAL DISCLAIMER

Dessert Finance provides due-diligence project audits for various projects. Dessert Finance in no way guarantees that a project will not remove liquidity, sell off team supply, or otherwise exit scam.

Dessert Finance does the legwork and provides public information about the project in an easy-to-understand format for the common person.

Agreeing to an audit in no way guarantees that a team will not remove ***all*** liquidity ("Rug Pull"), remove liquidity slowly, sell off tokens, quit the project, or completely exit scam. There is also no way to prevent private sale holders from selling off their tokens. It is ultimately your responsibility to read through all documentation, social media posts, and contract code of each individual project to draw your own conclusions and set your own risk tolerance.

Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice. Dessert Finance does not endorse, recommend, support, or suggest any projects that have been audited. An audit is an informational report based on our findings, We recommend you do your own research, we will never endorse any project to invest in.

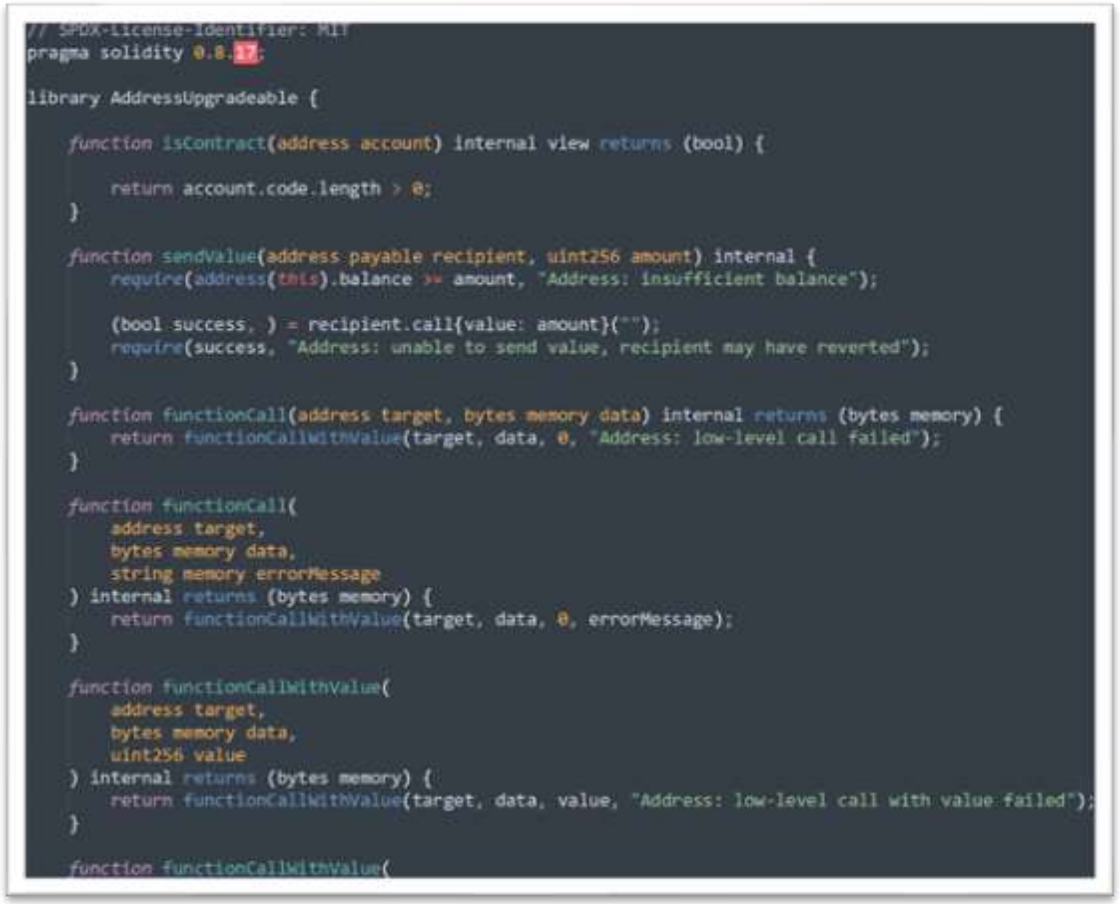
# DessertDoxxed

DessertDoxxed is a KYC service offered by Dessert Finance that allows projects to do a private face reveal matched with an I.D to allow founders / team members to privately Doxx themselves to Dessert Finance. This project has completed DessertDox

# Table of Contents

# BEP-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on CookieSale (COOKIE)



```
// SPDX-License-Identifier: MIT
pragma solidity 0.8.17;

library AddressUpgradeable {

    function isContract(address account) internal view returns (bool) {

        return account.code.length > 0;
    }

    function sendValue(address payable recipient, uint256 amount) internal {
        require(address(this).balance >= amount, "Address: insufficient balance");

        (bool success, ) = recipient.call{value: amount}("");
        require(success, "Address: unable to send value, recipient may have reverted");
    }

    function functionCall(address target, bytes memory data) internal returns (bytes memory) {
        return functionCallWithValue(target, data, 0, "Address: low-level call failed");
    }

    function functionCall(
        address target,
        bytes memory data,
        string memory errorMessage
    ) internal returns (bytes memory) {
        return functionCallWithValue(target, data, 0, errorMessage);
    }

    function functionCallWithValue(
        address target,
        bytes memory data,
        uint256 value
    ) internal returns (bytes memory) {
        return functionCallWithValue(target, data, value, "Address: low-level call with value failed");
    }

    function functionCallWithValue(
```

**Contract Address**
0x6d342877fC199c629f49A5C6C521C297b15BC92d for implementation of 0xbe05cc533ce058787ed9a520b41f5f364e0a69ee

**TokenTracker**
CookieSale (COOKIE)

**Contract Creator**
0x0c935e43adc96c9a9b7b814d9a070164e1150a72

**Source Code**
Contract Source Code Verified

**Contract Name**
COOKIE

**Compiler Version**
v0.8.17+commit.8df45f5f

The contract code is **verified** on BSCScan.

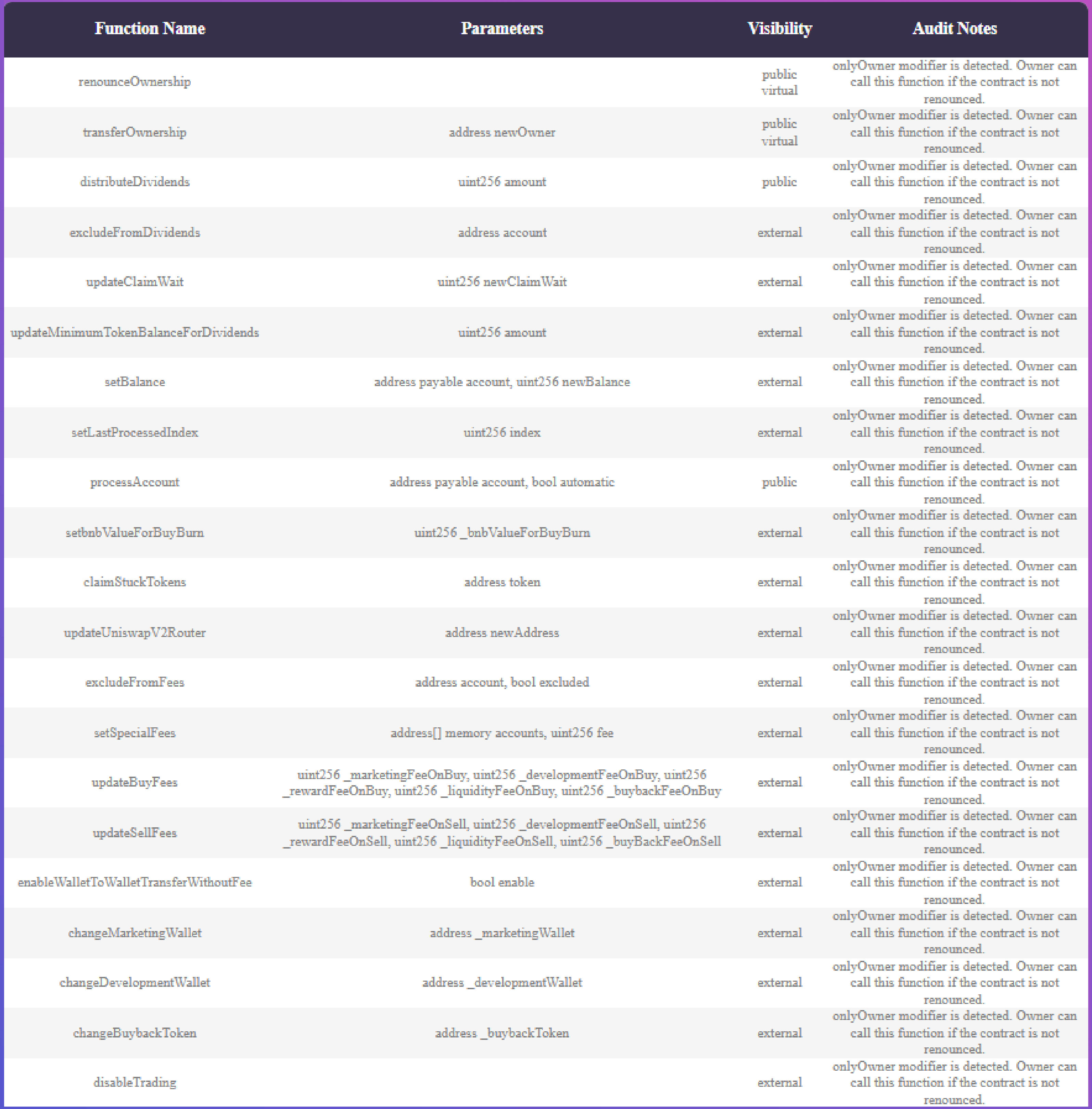
# BEP-20 Contract Code Audit – Vulnerabilities Checked

| Vulnerability Tested | AI Scan | Human Review | Result |
|---|---|---|---|
| Compiler Errors | Complete | Complete | ✓ Low / No Risk |
| Outdated Compiler Version | Complete | Complete | ✓ Low / No Risk |
| Integer Overflow | Complete | Complete | ✓ Low / No Risk |
| Integer Underflow | Complete | Complete | ✓ Low / No Risk |
| Correct Token Standards Implementation | Complete | Complete | ✓ Low / No Risk |
| Timestamp Dependency for Crucial Functions | Complete | Complete | ✓ Low / No Risk |
| Exposed _Transfer Function | Complete | Complete | ✓ Low / No Risk |
| Transaction-Ordering Dependency | Complete | Complete | ✓ Low / No Risk |
| Unchecked Call Return Variable | Complete | Complete | ✓ Low / No Risk |
| Use of Deprecated Functions | Complete | Complete | ✓ Low / No Risk |
| Unprotected SELFDESTRUCT Instruction | Complete | Complete | ✓ Low / No Risk |
| State Variable Default Visibility | Complete | Complete | ✓ Low / No Risk |
| Deployer Can Access User Funds | Complete | Complete | ✓ Low / No Risk |

The contract code is **verified** on BSCScan

The vulnerabilities listed above were not found in the token's Smart Contract.

# Contract Code Audit – Owner Accessible Functions

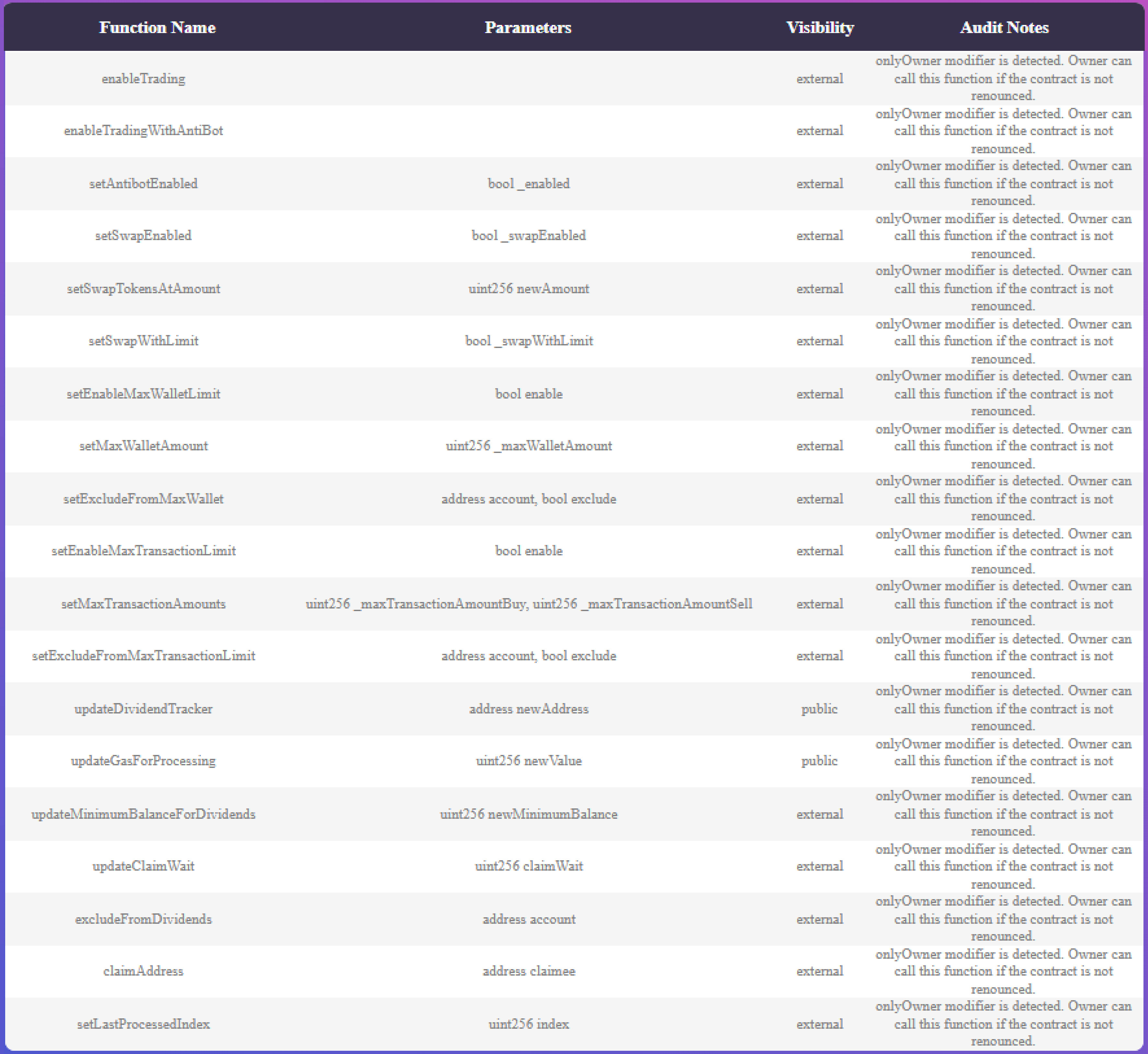


| Function Name | Parameters | Visibility | Audit Notes |
|---|---|---|---|
| renounceOwnership | | public virtual | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| transferOwnership | address newOwner | public virtual | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| distributeDividends | uint256 amount | public | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| excludeFromDividends | address account | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| updateClaimWait | uint256 newClaimWait | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| updateMinimumTokenBalanceForDividends | uint256 amount | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setBalance | address payable account, uint256 newBalance | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setLastProcessedIndex | uint256 index | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| processAccount | address payable account, bool automatic | public | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setbnbValueForBuyBurn | uint256 _bnbValueForBuyBurn | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| claimStuckTokens | address token | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| updateUniswapV2Router | address newAddress | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| excludeFromFees | address account, bool excluded | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setSpecialFees | address[] memory accounts, uint256 fee | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| updateBuyFees | uint256 _marketingFeeOnBuy, uint256 _developmentFeeOnBuy, uint256 _rewardFeeOnBuy, uint256 _liquidityFeeOnBuy, uint256 _buybackFeeOnBuy | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| updateSellFees | uint256 _marketingFeeOnSell, uint256 _developmentFeeOnSell, uint256 _rewardFeeOnSell, uint256 _liquidityFeeOnSell, uint256 _buyBackFeeOnSell | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| enableWalletToWalletTransferWithoutFee | bool enable | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| changeMarketingWallet | address _marketingWallet | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| changeDevelopmentWallet | address _developmentWallet | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| changeBuybackToken | address _buybackToken | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| disableTrading | | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |

The functions listed above can be called by the contract owner

If contract ownership has been renounced there is no way for the above listed functions to be called.

# Contract Code Audit – Owner Accessible Functions (pt 2)



| Function Name | Parameters | Visibility | Audit Notes |
|---|---|---|---|
| enableTrading | | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| enableTradingWithAntiBot | | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setAntibotEnabled | bool _enabled | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setSwapEnabled | bool _swapEnabled | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setSwapTokensAtAmount | uint256 newAmount | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setSwapWithLimit | bool _swapWithLimit | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setEnableMaxWalletLimit | bool enable | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setMaxWalletAmount | uint256 _maxWalletAmount | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setExcludeFromMaxWallet | address account, bool exclude | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setEnableMaxTransactionLimit | bool enable | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setMaxTransactionAmounts | uint256 _maxTransactionAmountBuy, uint256 _maxTransactionAmountSell | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setExcludeFromMaxTransactionLimit | address account, bool exclude | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| updateDividendTracker | address newAddress | public | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| updateGasForProcessing | uint256 newValue | public | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| updateMinimumBalanceForDividends | uint256 newMinimumBalance | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| updateClaimWait | uint256 claimWait | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| excludeFromDividends | address account | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| claimAddress | address claimee | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |
| setLastProcessedIndex | uint256 index | external | onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. |

The functions listed above can be called by the contract owner

If contract ownership has been renounced there is no way for the above listed functions to be called.

# Contract Code Audit – Contract Ownership

## Contract has been identified as an Upgradeable Contract

This is an upgradeable contract.

This means that the contract address that is being pointed to can be changed / upgraded at any time by the proxy administrators.

# Liquidity Ownership – Locked / Unlocked

## No locked liquidity information has been found.

This page will contain links to locked liquidity for the project if we are able to locate that information. Locked liquidity information was not found on the project’s website.

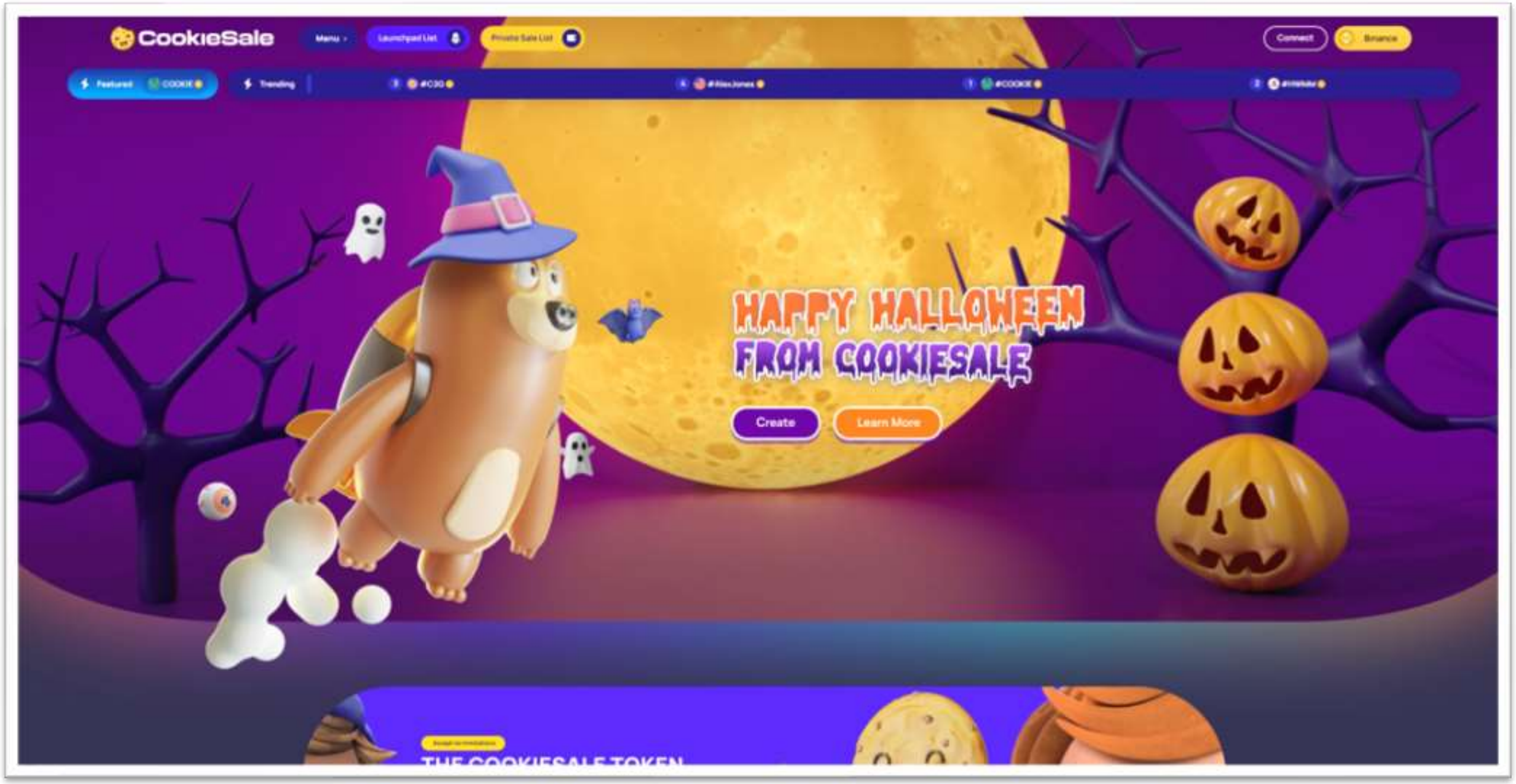
# Contract Code Audit – Mint Functions

## This Contract Cannot Mint New Cookie Tokens.

We do understand that sometimes mint functions are essential to the functionality of the project.

**A mint function was not found in the contract code.**

# Website Part 1 – Overview
## www.cookiesale.io

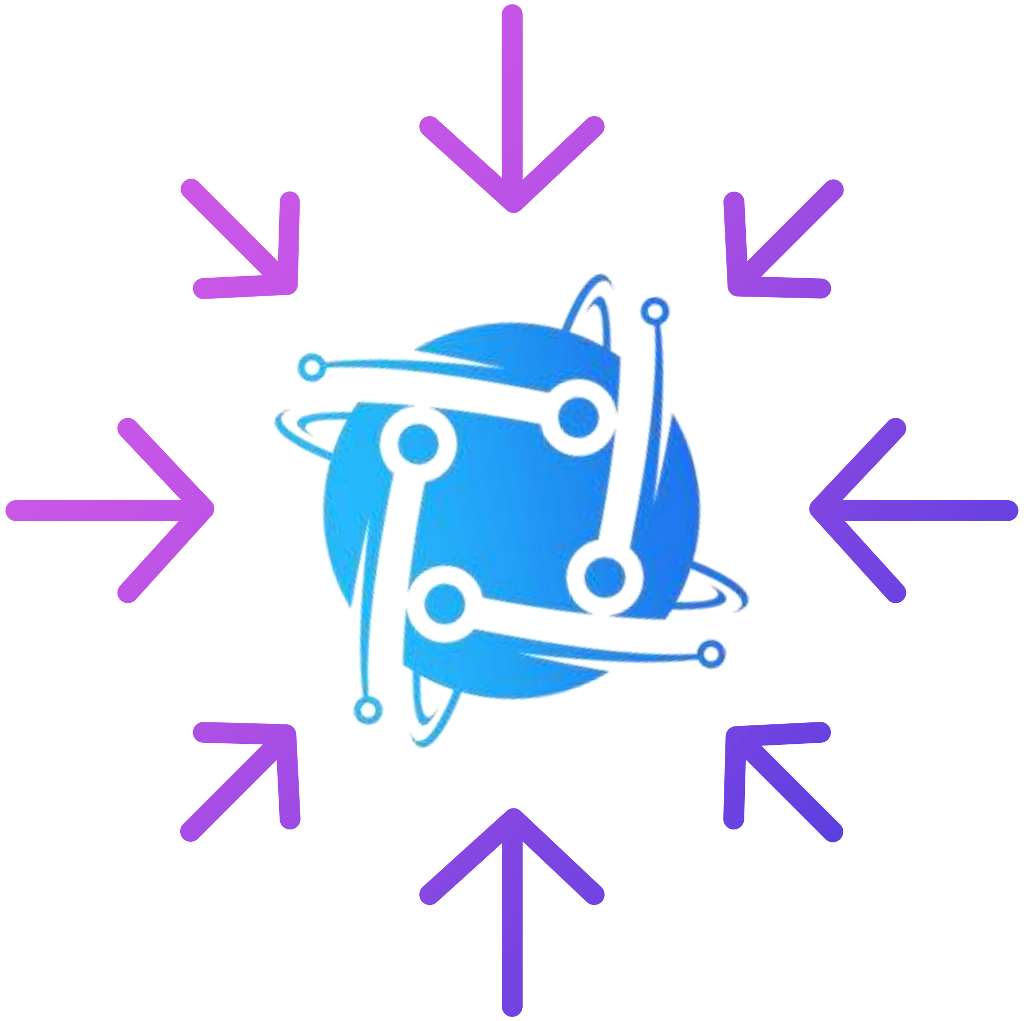


Above images are actual snapshots of the current live website of the project.

Website was registered on 01/24/2022, registration expires 01/24/2028.

✓ This **exceeds** meet the 3 year minimum we like to see on new projects.

# Website Part 2 – Checklist



- ✓ Mobile Friendly
- ✓ No JavaScript Errors
- ✓ Spell Check
- ✓ SSL Certificate

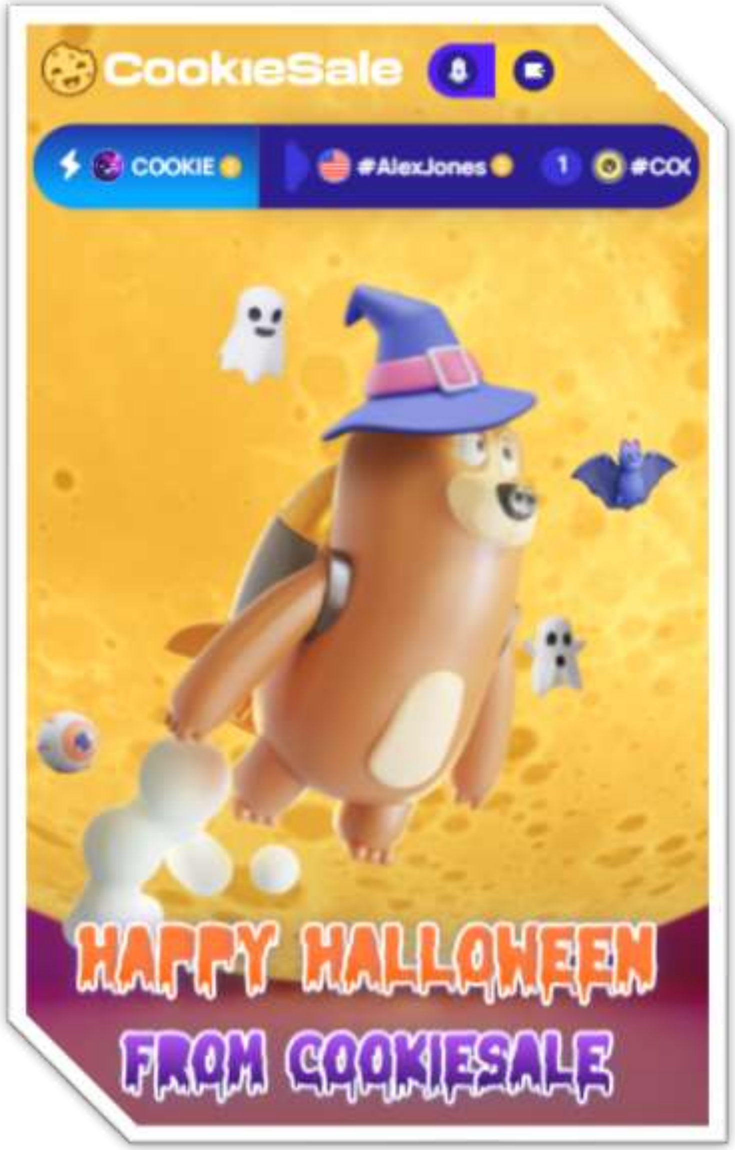
The website contained no JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

No additional issues were found on the website.

# Website Part 3 – Responsive HTML5 & CSS3

No issues were found on the Mobile Friendly check for the website. All elements loaded properly and browser resize was not an issue. The team has put a considerable amount of thought and effort into making sure their website looks great on all screens.

No severe JavaScript errors were found. No issues with loading elements, code, or stylesheets.



# Website Part 4 (GWS) – General Web Security

**SSL CERTIFICATE**

A valid SSL certificate was found. Details are as follows:

Offered to: cookiesale.io

Issued by: cloudflare, Inc.

Valid Until: 01/31/2023

✓

**CONTACT EMAIL**

A valid contact email was found on the official website. Contact email is listed as shown below:

Contact

Contact Form on Website

✓

**SPAM / MALWARE / POPUPS**

No malware found

No injected spam found

No internal server errors

No popups found

Domain is marked clean by Google, McAfee, Sucuri Labs, & ESET

✓

# Social Media

We were able to locate a variety of Social Media networks for the project.

All links have been conveniently placed below.

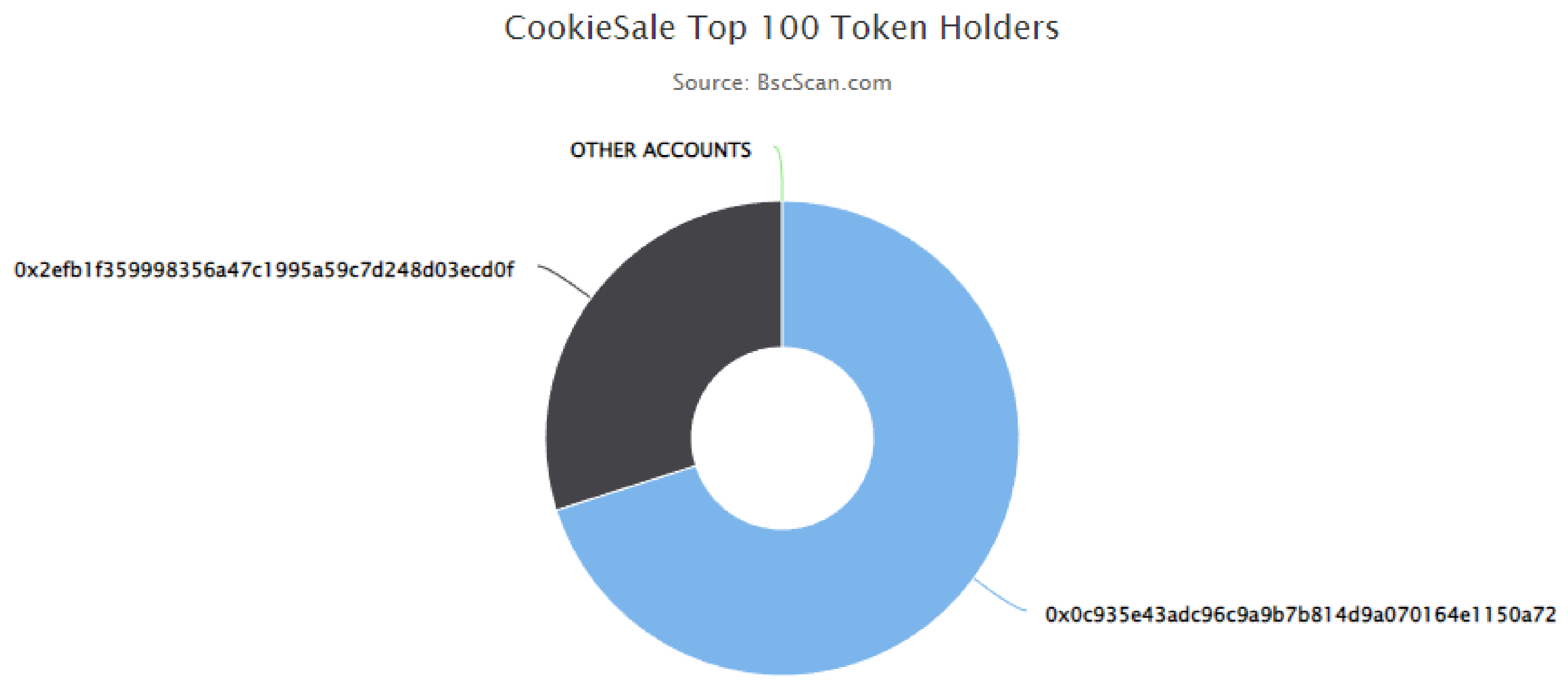
Twitter Telegram Youtube

✓ **At least 3 social media networks were found.**

# Top Token Holders

The entire supply was in one wallet at the time of audit. We expect this to change as the project goes through initial distribution phases. Please use the link below to view the most up-to-date holder information.

Click here to view the most up-to-date list of holders

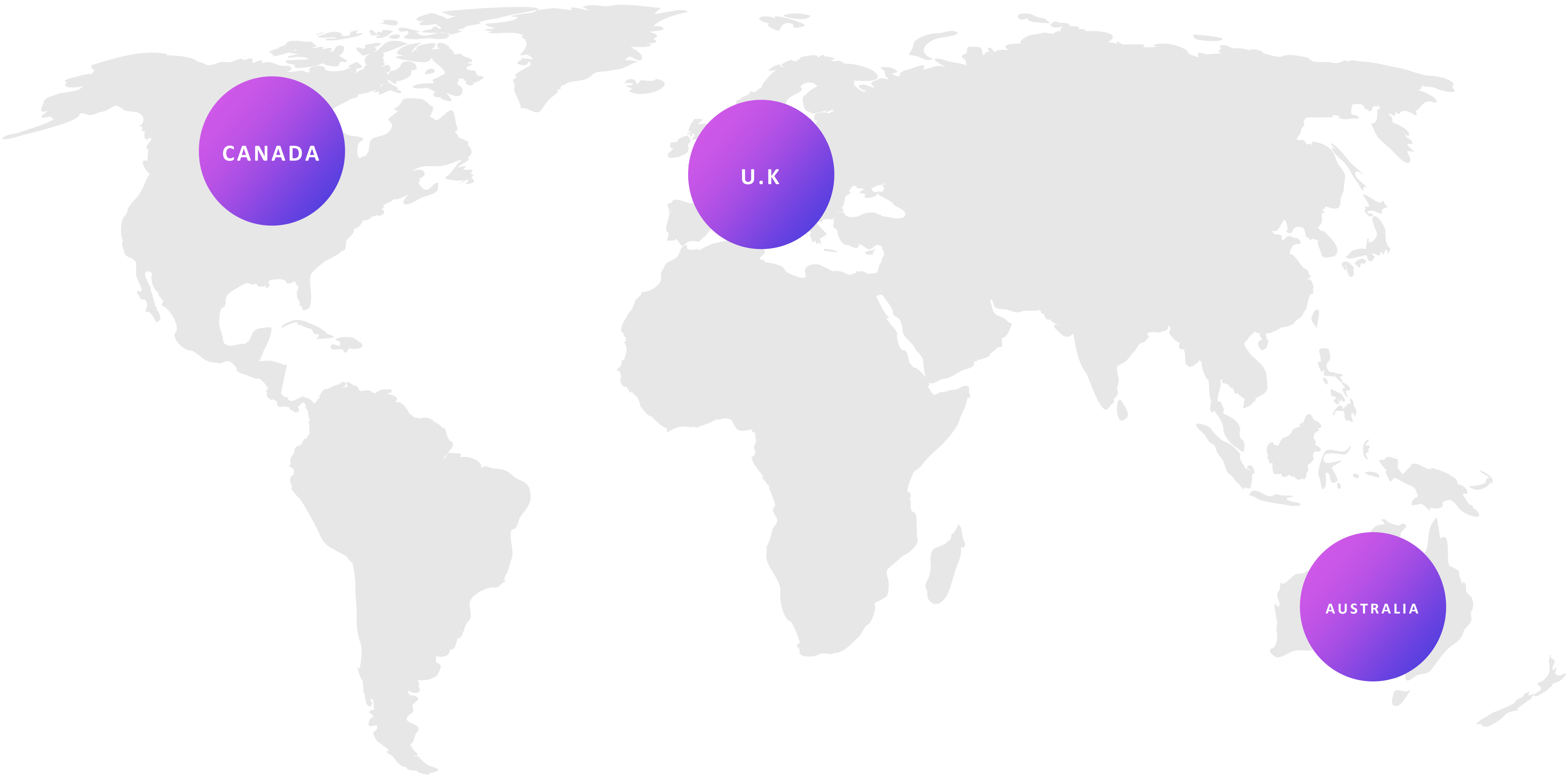


CookieSale Top 100 Token Holders

Source: BscScan.com

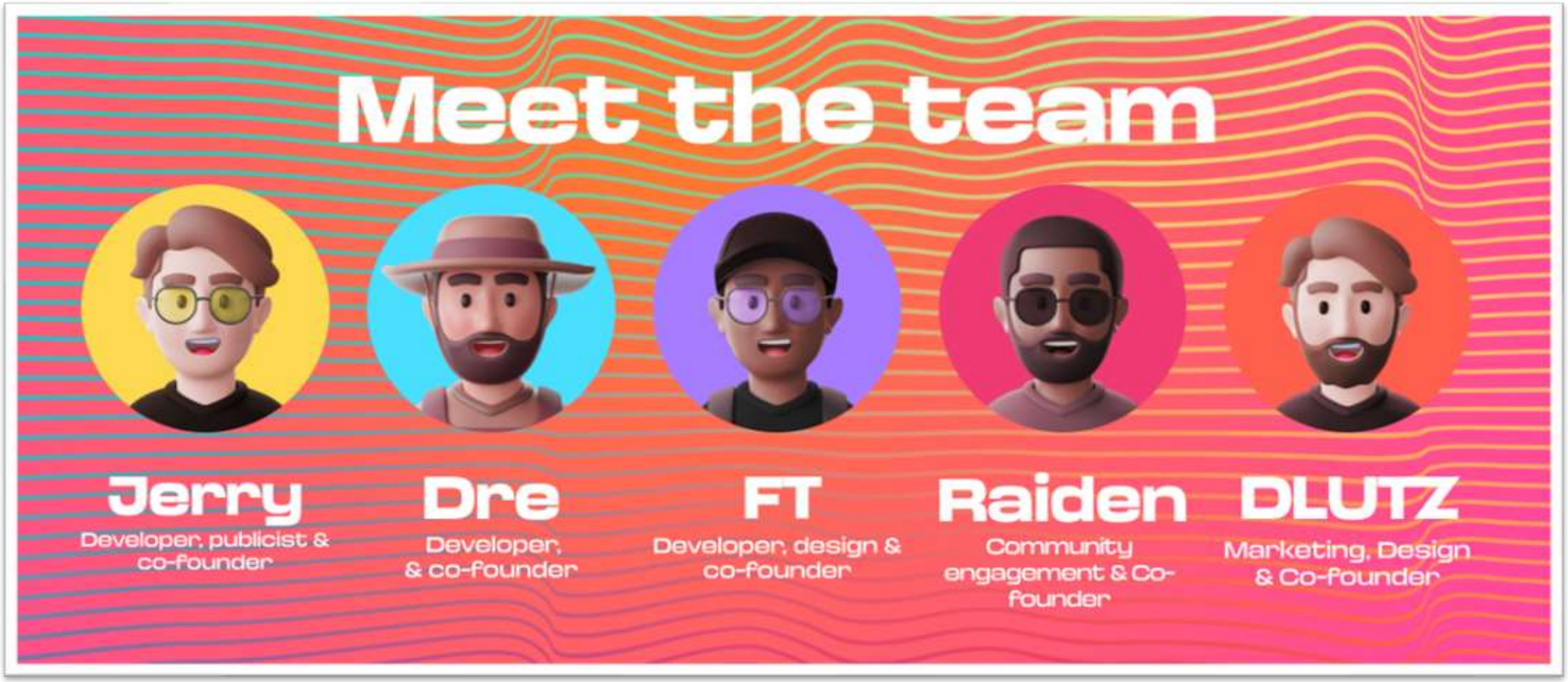
# Location Audit

The team is located in Canada, The United Kingdom and Australia



# Team Overview

The following information has been found on the projects website.



Meet the team

**Jerry**
Developer, publicist & co-founder

**Dre**
Developer, & co-founder

**FT**
Developer, design & co-founder

**Raiden**
Community engagement & Co-founder

**DLUTZ**
Marketing, Design & Co-founder

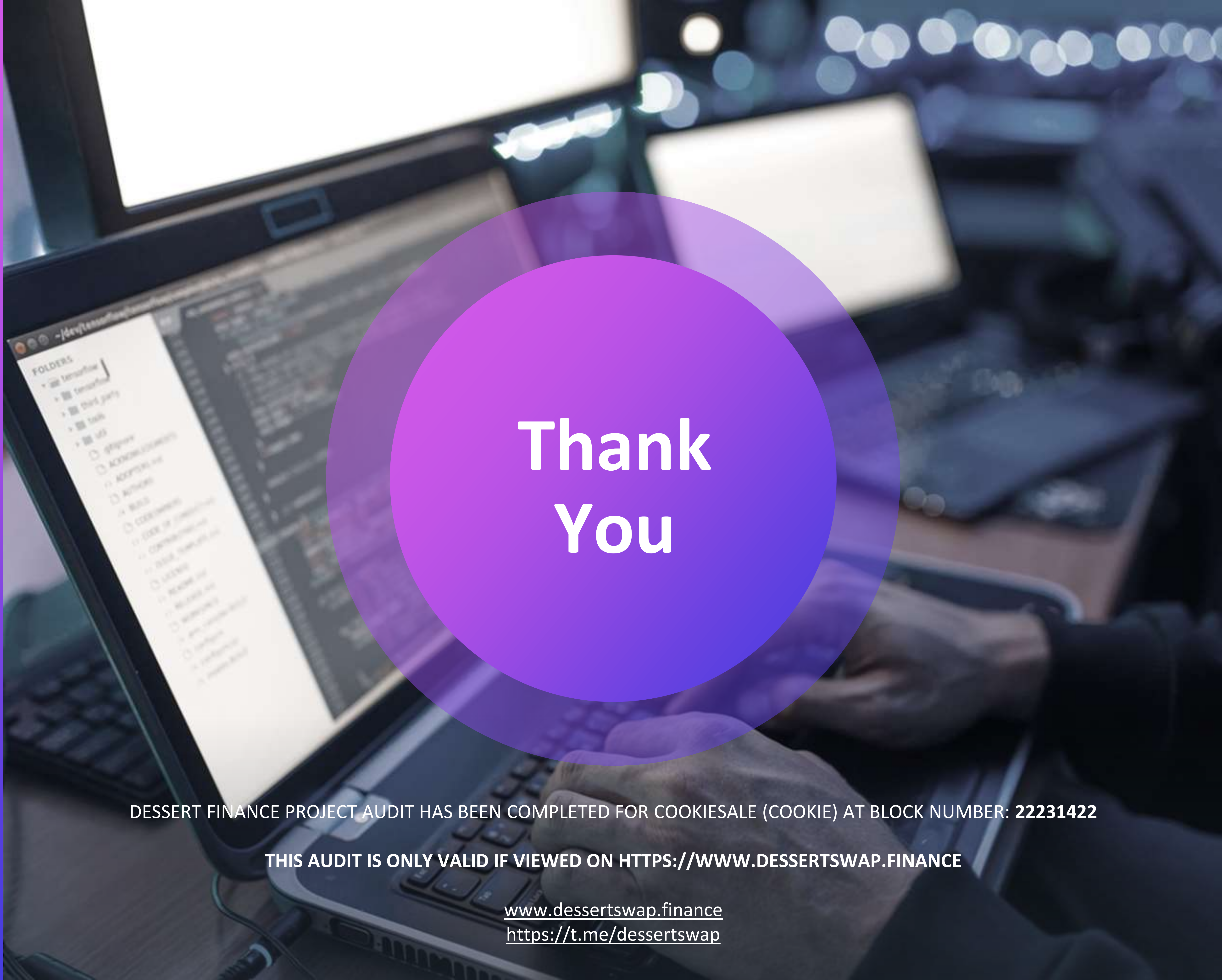
# Disclaimer

The opinions expressed in this document are for general informational purposes only and are **not intended to provide specific advice or recommendations for any individual or on any specific investment**. It is only intended to provide education and public knowledge regarding projects. This audit is only applied to the type of auditing specified in this report and the scope of given in the results. Other unknown security vulnerabilities are beyond responsibility. Dessert Finance only issues this report based on the attacks or vulnerabilities that already existed or occurred before the issuance of this report. For the emergence of new attacks or vulnerabilities that exist or occur in the future, Dessert Finance lacks the capability to judge its possible impact on the security status of smart contracts, thus taking no responsibility for them. The smart contract analysis and other contents of this report are based solely on the documents and materials that the contract provider has provided to Dessert Finance or was publicly available before the issuance of this report (issuance of report recorded via block number on cover page), if the documents and materials provided by the contract provider are missing, tampered, deleted, concealed or reflected in a situation that is inconsistent with the actual situation, or if the documents and materials provided are changed after the issuance of this report, Dessert Finance assumes no responsibility for the resulting loss or adverse effects. Due to the technical limitations of any organization, this report conducted by Dessert Finance still has the possibility that the entire risk cannot be completely detected. Dessert Finance disclaims any liability for the resulting losses.

Dessert Finance provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Even projects with a low risk score have been known to pull liquidity, sell all team tokens, or exit-scam. Please exercise caution when dealing with any cryptocurrency related platforms.

The final interpretation of this statement belongs to Dessert Finance.

Dessert Finance highly advises against using cryptocurrencies as speculative investments and they should be used solely for the utility they aim to provide.



# Thank You

DESSERT FINANCE PROJECT AUDIT HAS BEEN COMPLETED FOR COOKIESALE (COOKIE) AT BLOCK NUMBER: **22231422**

**THIS AUDIT IS ONLY VALID IF VIEWED ON HTTPS://WWW.DESSERTSWAP.FINANCE**

www.dessertswap.finance
https://t.me/dessertswap