

DESSERT  
FINANCE



## Christmas Gift (CG)

BEP-20 Audit

Performed at block **SAMPLE**

PERFORMED BY DESSERT FINANCE  
FOR CONTRACT ADDRESS: `0xb576D58e7A390A3efc03f3E7f4feE48956c9Da00`

## INITIAL DISCLAIMER

Dessert Finance provides due-diligence project audits for various BSC 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 a project audit can be seen as a sign of confidence and is generally the first sign of trust for a project, but in no way guarantees that a team will not remove *all* liquidity (“Rug Pull”), sell off tokens, 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.

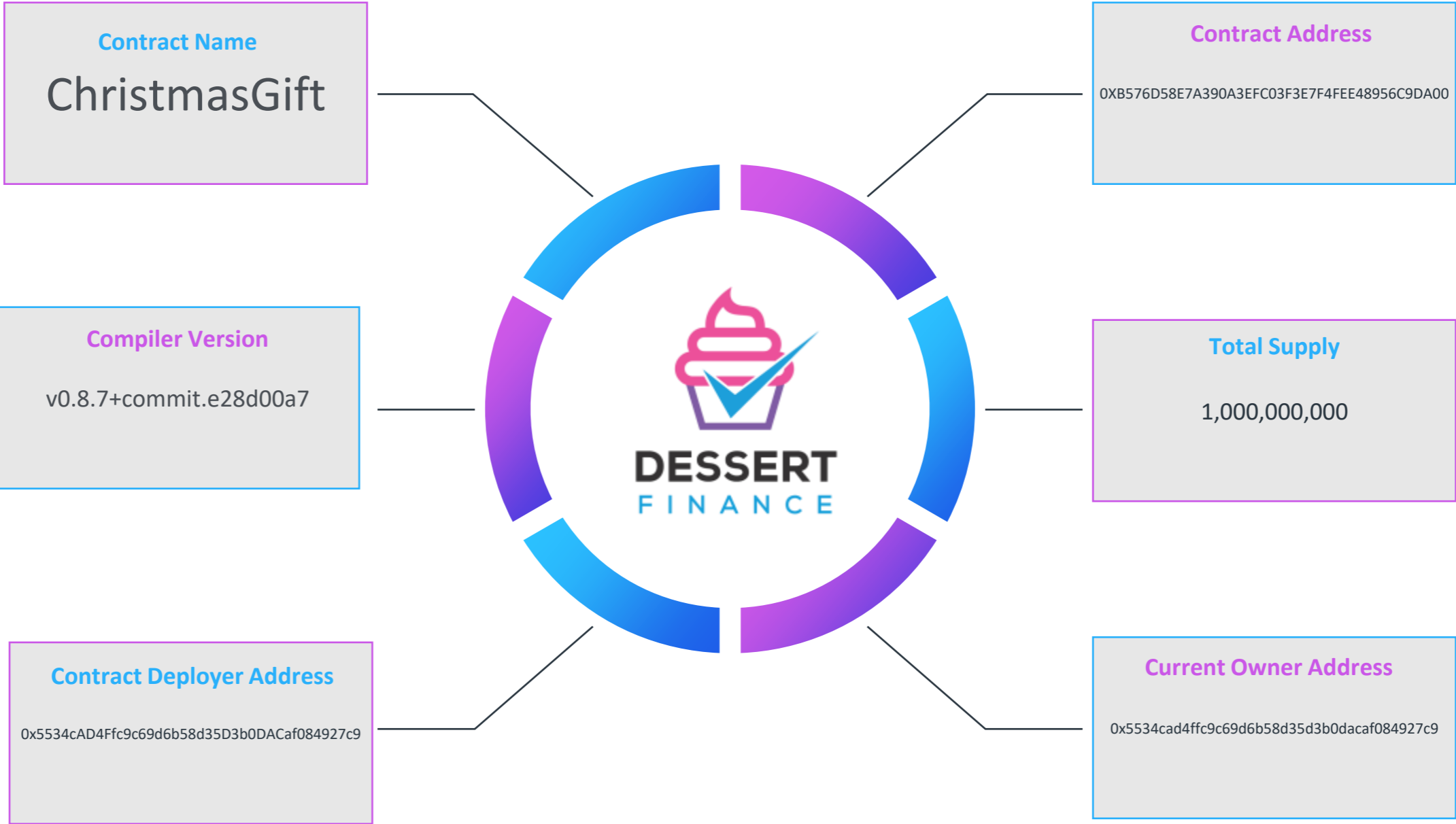
Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice.

# Table of Contents



1. Contract Code Audit – Token Overview
2. BEP-20 Contract Code Audit – Overview
3. BEP-20 Contract Code Audit – Vulnerabilities Checked
4. Contract Code Audit – Owner Accessible Functions
5. Contract Code Audit – Contract Ownership
6. Liquidity Ownership – Locked / Unlocked
7. Contract Code Audit – Mint Functions
8. Contract Transaction Fees
9. Website Overview
10. Social Media
11. Top Token Holders/Wallets
12. Location Audit
13. Review of Team
14. Roadmap
15. Disclaimers

# Contract Code Audit – Token Overview



# BEP-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on Christmas Gift (CG)

```
pragma solidity ^0.8.0;

import "../IBEP20.sol";
import "../SafeMath.sol";
import "../Contract.sol";
import "../Enumerable.sol";
import "../PancakeswapRouter.sol";
import "../PancakeswapRouterV2.sol";

contract ChristmasGift is Contract, IBEP20, Enumerable {
    using SafeMath for uint256;

    mapping (address => uint256) private _balances;
    mapping (address => mapping (address => uint256)) private _allowances;
    mapping (address => bool) public _isBlacklisted;
    mapping (address => bool) private _isIncludedFromFee;
    address public _tokenAddress;
    address public _marketingAddress;
    address public _marketingWalletAddress;
    uint256 private _total = 100000000 * 10**18;
    uint256 private constant MAX = 500000000;
    string private _name = "Christmas Gift";
    string private _symbol = "CG";
    uint8 private _decimals = 18;

    uint256 public _BOPFee = 0;
    uint256 private _previousBOPFee = _BOPFee;

    uint256 public _marketingFee = 1;
    uint256 private _previousMarketingFee = _marketingFee;

    IPancakeswapRouterV2 public _pancakeswapRouter;
    address public _pancakeswapRouter;

    bool _isSwap;
    bool public _isSwapped = true;
    uint256 private _marketingSwap = 1 * 10**3 * 10**18;
    uint256 public _swapCooldownTime = 28;
    uint256 private _lastSwapTime;
    uint256 private _liquidBlock = 8;
    uint256 private _swapLockAmount = 2;
    mapping (address => uint256) private _lastTimes;

    event SwapPublicData([bool] _swapped);
    event IncludedFromFee(address account);
    event IncludedFromFee(address account);
    event UpdateFee(uint256 _before, uint256 _marketingFee);
    event UpdateSwapAmount(uint256 _amount);
    event UpdateSwapTimes(uint256 _amount);
    event UpdateCooldownTime(uint256 _timeForContract);
    modifier LockTheSwap {
        _swap = true;
        _;
        _swap = false;
    }

    constructor () {
        IPancakeswapRouterV2 _pancakeswapRouter = IPancakeswapRouterV2(0x1000000000000000000000000000000000000000);
        _marketingWalletAddress = msg.sender;
        _pancakeswapRouter = IPancakeswapRouterFactory(_pancakeswapRouter, Factory{});
        _routerPair(address(0x0000000000000000000000000000000000000000), _pancakeswapRouter.WETH);
        // set the rest of the contract variables
    }
}
```

## Contract Address

0xb576D58e7A390A3efc03f3E7f4feE48956c9Da00

## TokenTracker

Christmas Gift (CG)

## Contract Creator

0x5534cad4ffc9c69d6b58d35d3b0dacaf084927c9

## Source Code

Contract Source Code Verified

## Contract Name

ChristmasGift

## Other Settings

default evmVersion, None

## Compiler Version

v0.8.7+commit.e28d00a7

## Optimization Enabled

No with 200 runs

Code is truncated to fit the constraints of this document.

[The code in its entirety can be viewed here.](#)

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

# BEP-20 Contract Code Audit – Vulnerabilities Checked

Vulnerability Tested	AI Scan	Human Review	Result
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 (x1)	Complete	Complete	✓ Low 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 – Contract Ownership

Contract Ownership has not been renounced at the time of Audit



The contract ownership is not currently renounced.

We have placed the contract owner address below for your viewing:

[0x5534cad4ffc9c69d6b58d35d3b0dacad084927c9](https://www.etherbase.io/etherscan/address/0x5534cad4ffc9c69d6b58d35d3b0dacad084927c9)

The address above has authority over the ownable functions within the contract.

This allows the owner to call certain functions within the contract. Any compromise to the owner wallet may allow these privileges to be exploited.

We recommend:

- Establishing a Time-Lock with reasonable latency
- Assignment of privileged roles to multi-signature wallets

# Contract Code Audit – Owner Accessible Functions

Function Name	Parameters	Visibility	Audit Notes
setBNBPoolAddress	address account	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setMarketingAddress	address account	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setCoolDownTime	uint256 timeForContract	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromFee	address account	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
includeInFee	address account	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setFees	uint256 bnbFee, uint256 marketingFee	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. <b>This function does not contain an upper limit, fees can be set to 100.</b>
setNumTokensToSwap	uint256 amount	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setMarketingWallet	address payable wallet	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setSwapEnabled	bool _enabled	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
blacklistAddress	address account, bool value	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
sendMarketToken	address[] memory userAddress,uint amount	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
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.
lock	uint256 time	public virtual	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.



# 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.

# Contract Code Audit – Mint Functions

This Contract Cannot Mint New CG Tokens.

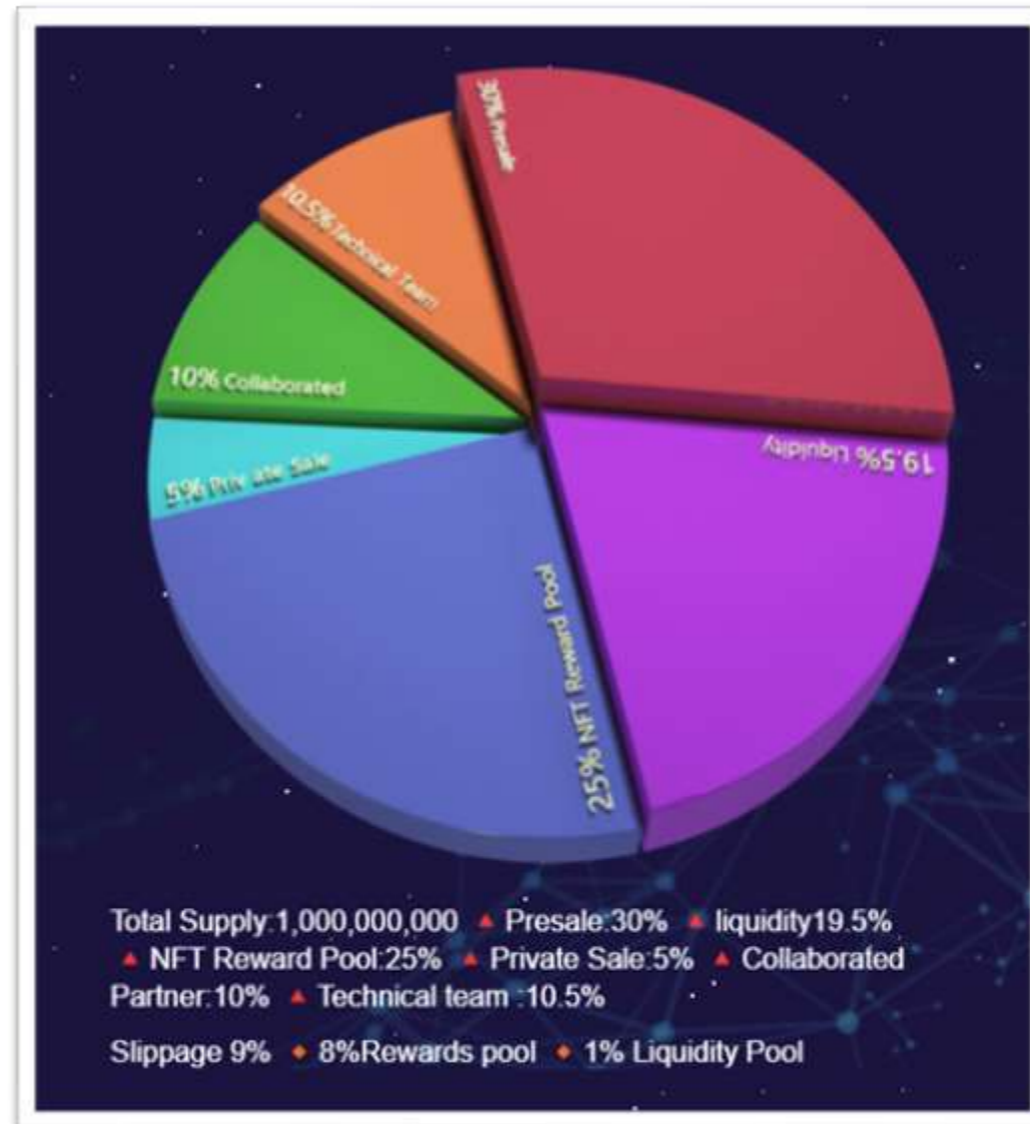


We do understand that sometimes mint functions are essential to the functionality of the project.

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

# Contract Transaction Fees

At the time of Audit the transaction fees (“tax”) listed below are the fees associated with trading. These fees are taken from every buy and sell transaction unless otherwise stated.



# Website Part 1 – Overview

## [www.christmasgift.top](http://www.christmasgift.top)



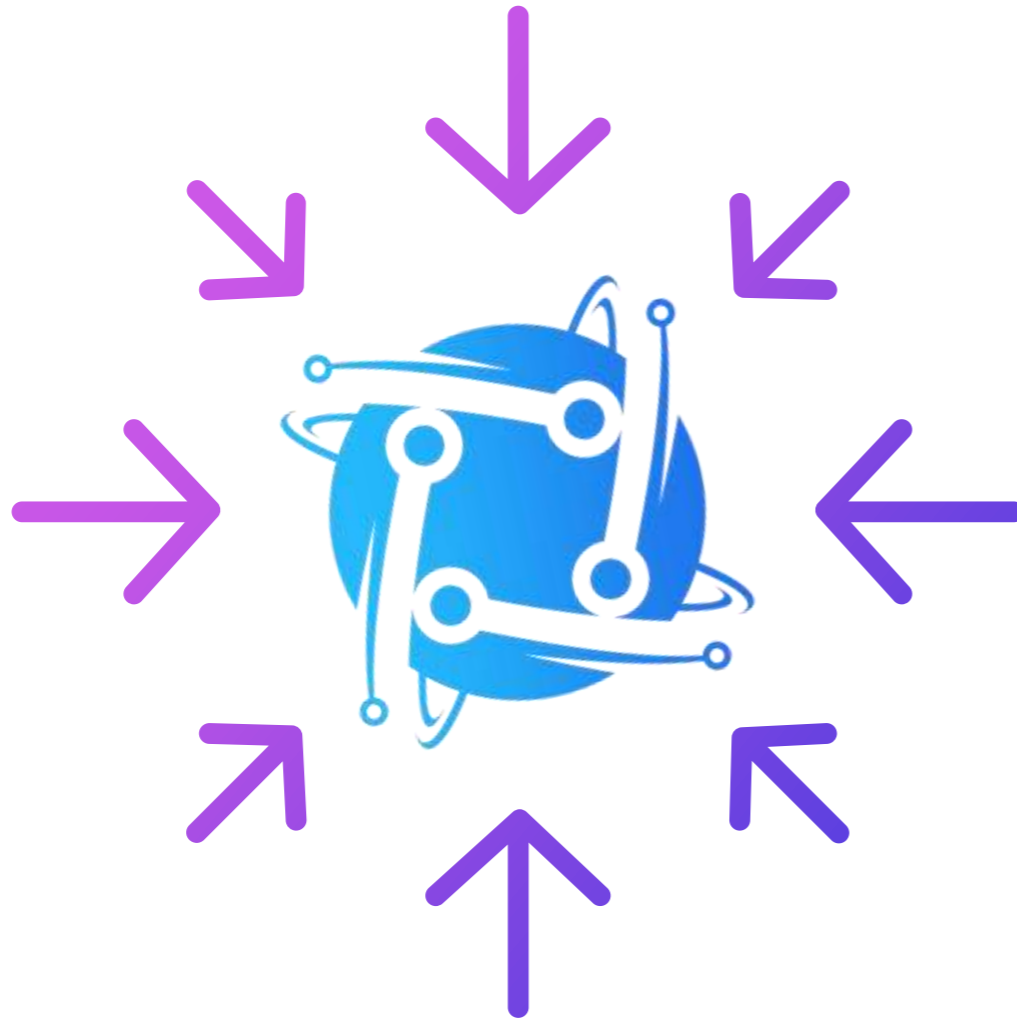
Above images are actual snapshots of the current live website of the project.

Website was registered on 12/18/2021, registration expires 12/18/2022.

**X** This does not 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 one 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.

One minor JavaScript error was found and shown below. No issues with loading elements, code, or stylesheets.

```
✖ Uncaught TypeError: Cannot read properties of null (reading 'getContext')  
    at (index):402
```



# Website Part 4 (GWS) – General Web Security



## SSL CERTIFICATE

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

Offered to: N/A

Issued by: N/A

Valid Until: N/A



## CONTACT EMAIL

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

Contact

N/A



## SPAM / MALWARE / POPUPS

No malware found

No injected spam found

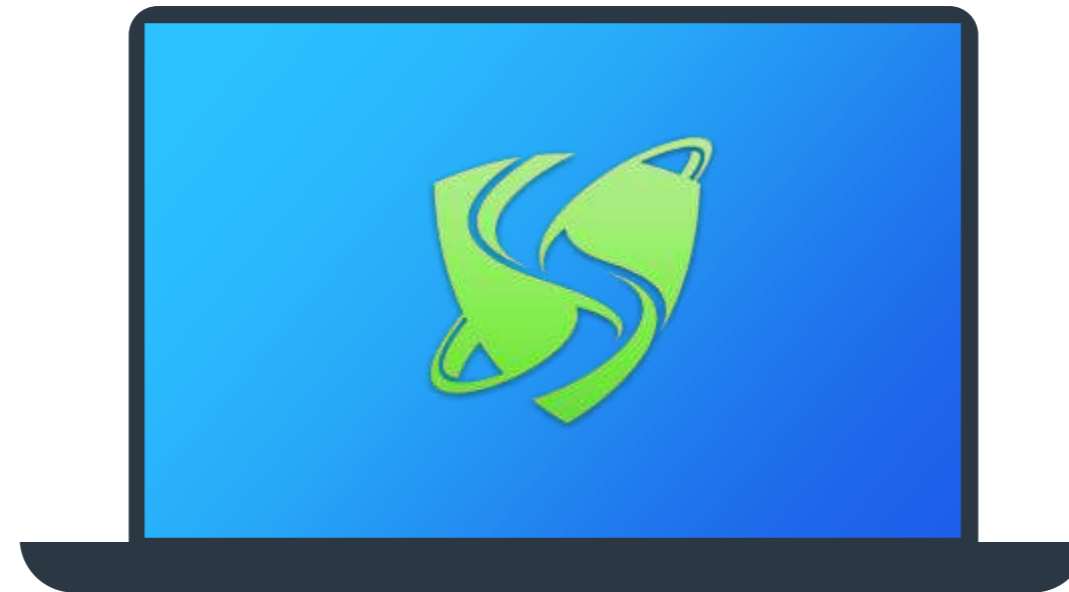
No internal server errors

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](#)

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



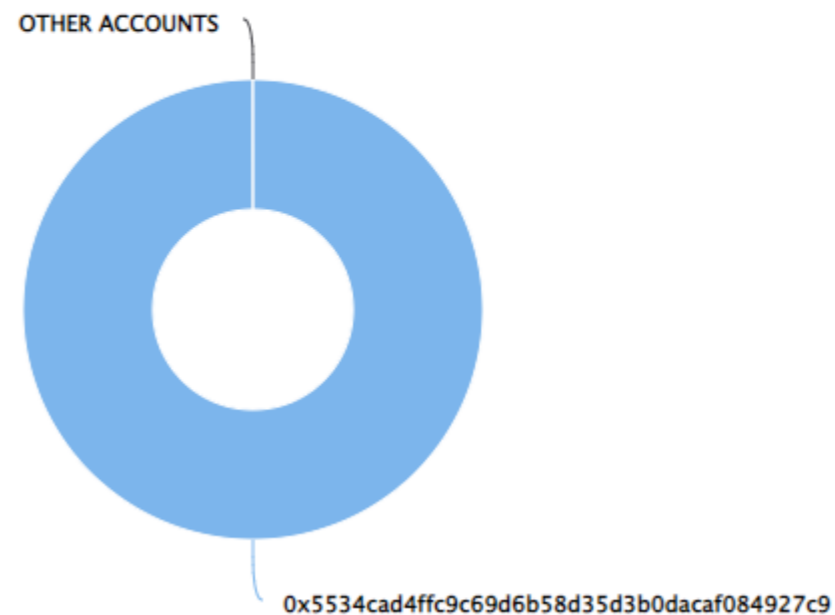
# Top Token Holders

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](#)

## Christmas Gift Top 100 Token Holders

Source: BscScan.com



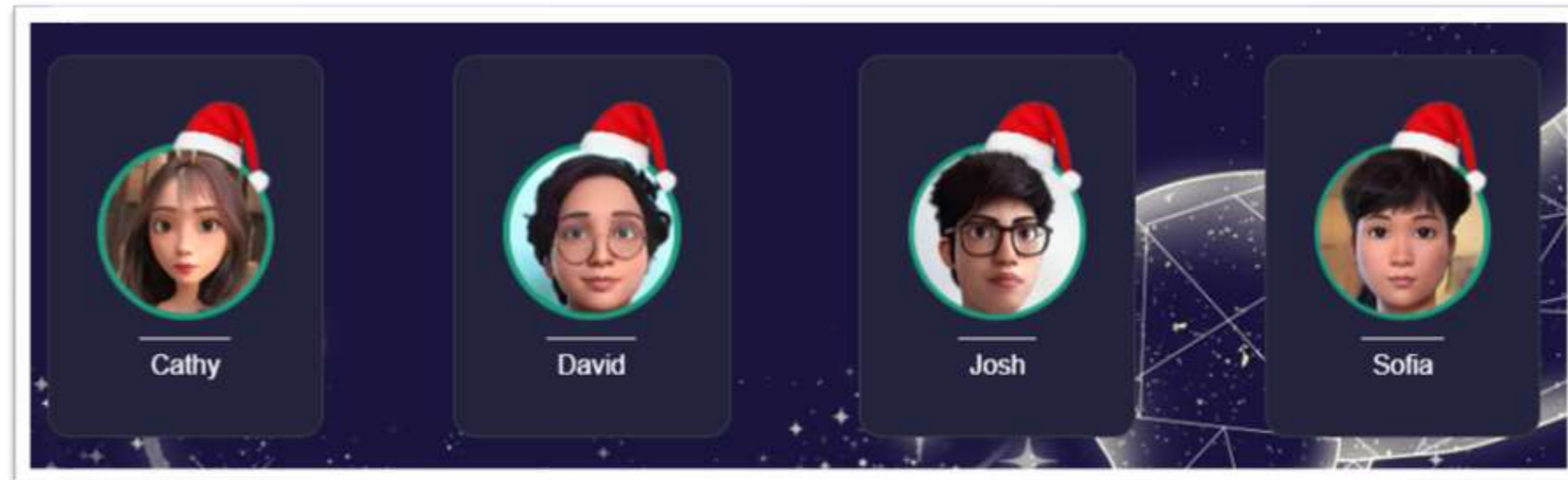
# Location Audit

We were unable to identify a primary location for the project at this time or a location has not been declared.



# Team Overview

*Team information was found on the projects website and is shown below.*



# Roadmap

*A roadmap was found on the official website, we have conveniently placed it on this page for your viewing.*



# 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 BSC 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 CHRISTMAS GIFT (CG) AT BLOCK NUMBER: **DRAFT**  
THIS AUDIT IS ONLY VALID IF VIEWED ON [WWW.DSSERTSWAP.FINANCE](http://WWW.DSSERTSWAP.FINANCE)  
1 DSRT HAS BEEN SENT TO AUDITED CONTRACT FOR VERIFICATION

[www.dessertswap.finance](http://www.dessertswap.finance)  
<https://t.me/dessertswap>