

DESSERT  
FINANCE



**Gorilla gang (GGN\$)**

**BEP-20 Audit**

Performed at block **13618102**

PERFORMED BY DESSERT FINANCE  
FOR CONTRACT ADDRESS: 0x80EdF9C7eBc910C1edB60D4E10F0CDcba9836084

## 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.

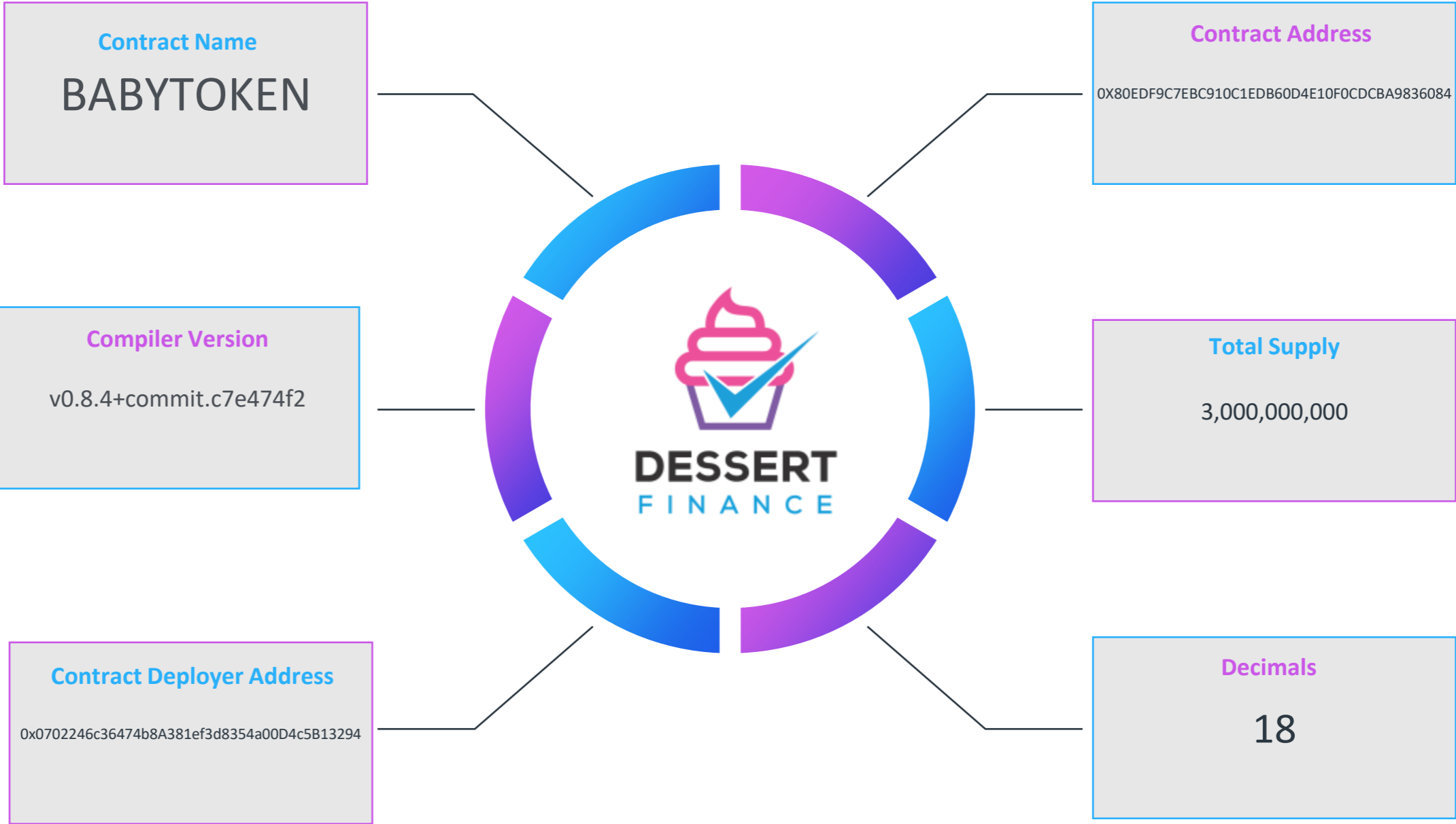
Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice.

# 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 Gorilla gang (GGN\$)

```

// SPDX-License-Identifier: MIT
// OpenZeppelin Contracts (last updated v4.5.0)
// Copyright (c) 2022 OpenZeppelin

pragma solidity ^0.8.0;

import "@openzeppelin/contracts/token/ERC20/IERC20.sol";
import "@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol";

interface IERC20 {
    function totalSupply() external view returns (uint256);
    function balanceOf(address account) external view returns (uint256);
    function transfer(address recipient, uint256 amount) external returns (bool);
    function allowance(address owner, address spender) external view returns (uint256);
    function approve(address spender, uint256 amount) external returns (bool);
    function transferFrom(address sender, address recipient, uint256 amount) external returns (bool);
}

interface IERC20Metadata {
    function name() external view returns (string);
    function symbol() external view returns (string);
    function decimals() external view returns (uint8);
}

contract BabyToken is IERC20, IERC20Metadata {
    string public name = "BABYTOKEN";
    string public symbol = "BABY";
    uint8 public decimals = 18;

    uint256 public totalSupply;
    mapping(address => uint256) public balanceOf;
    mapping(address => mapping(address => uint256)) public allowance;

    constructor(uint256 _totalSupply) {
        totalSupply = _totalSupply;
    }

    function totalSupply() external view returns (uint256) {
        return totalSupply;
    }

    function balanceOf(address account) external view returns (uint256) {
        return balanceOf[account];
    }

    function transfer(address recipient, uint256 amount) external returns (bool) {
        _transfer(msg.sender, recipient, amount);
    }

    function allowance(address owner, address spender) external view returns (uint256) {
        return allowance[owner][spender];
    }

    function approve(address spender, uint256 amount) external returns (bool) {
        _approve(msg.sender, spender, amount);
    }

    function transferFrom(address sender, address recipient, uint256 amount) external returns (bool) {
        _transfer(sender, recipient, amount);
    }

    function _transfer(address sender, address recipient, uint256 amount) private {
        require(sender != address(0), "ERC20: transfer from the zero address");
        require(recipient != address(0), "ERC20: transfer to the zero address");
        require(balanceOf[sender] >= amount, "ERC20: transfer amount exceeds balance");
        balanceOf[sender] -= amount;
        balanceOf[recipient] += amount;
        emit Transfer(sender, recipient, amount);
    }

    function _approve(address owner, address spender, uint256 amount) private {
        require(owner != address(0), "ERC20: approve from the zero address");
        require(spender != address(0), "ERC20: approve to the zero address");
        allowance[owner][spender] = amount;
        emit Approval(owner, spender, amount);
    }
}

```

## Contract Address

0x80EdF9C7eBc910C1edB60D4E10F0CDcba9836084

## TokenTracker

Gorilla gang (GGN\$)

## Contract Creator

0x0702246c36474b8a381ef3d8354a00d4c5b13294

## Source Code

Contract Source Code Verified

## Contract Name

BABYTOKEN

## Other Settings

default evmVersion, MIT

## Compiler Version

v0.8.4+commit.c7e474f2

## Optimization Enabled

Yes 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
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	none	public virtual	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. This function has not been called and contract remains owned
transferOwnership	address newOwner	public virtual	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
distributeCAKEDividends	uint256 amount	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setSwapTokensAtAmount	uint256 amount	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.
updateUniswapV2Router	address newAddress	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromFees	address account, bool excluded	public	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.
setTokenRewardsFee	uint256 value	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. <b>Total fees for all taxes cannot exceed 25%</b>
setLiquiditFee	uint256 value	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. <b>Total fees for all taxes cannot exceed 25%</b>
setMarketingFee	uint256 value	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. <b>Total fees for all taxes cannot exceed 25%</b>
updateGasForProcessing	uint256 newValue	public	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.

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 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:

[0x0702246c36474b8a381ef3d8354a00d4c5b13294](https://www.etherbase.net/etherbase-token/contract/0x0702246c36474b8a381ef3d8354a00d4c5b13294)



# 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 GGN\$ 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

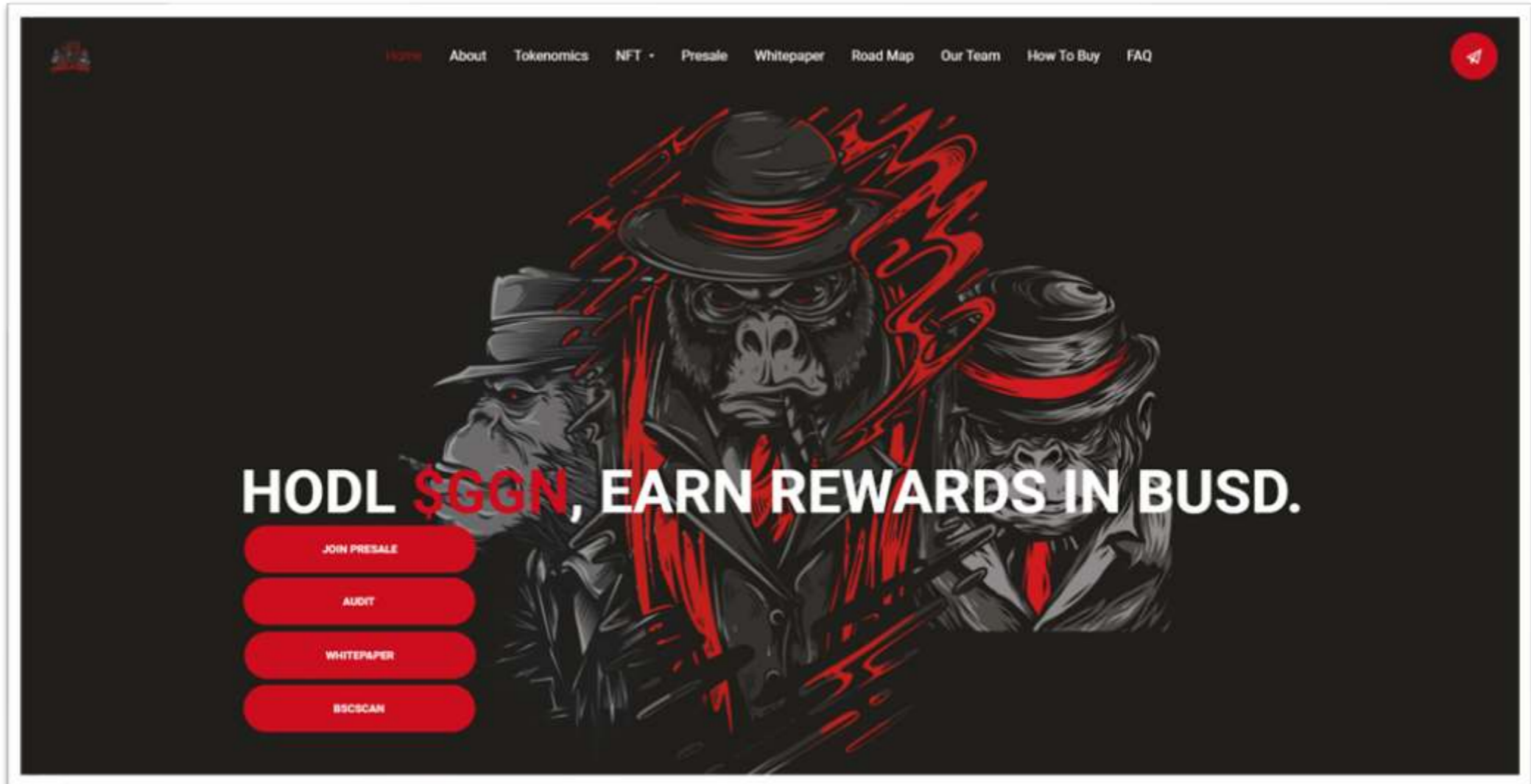
At the time of Audit the transaction fees (“tax”) listed below are the fees associated with trading. These fees are taken from every buy and sell transaction unless otherwise stated.

**GORILLA GANG TOKENOMICS**

- 5% Marketing**  
5% of all transactions will go to our team marketing wallet to ensure we are able to come up with the best campaigns to get a great return on your investment
- 4% Buy Back**  
4% of every transaction is transferred into liquidity Pool for Pancakeswap to create a stable price floor.
- 4% BUSD Reward**  
4% is reflected back to all holders on each transaction.  
The contract tracks all \$GGN holders and the amount of tokens held relative to the overall token supply. Rewards will be in \$BUSD.

# Website Part 1 – Overview

[www.gorillagang.net](http://www.gorillagang.net)



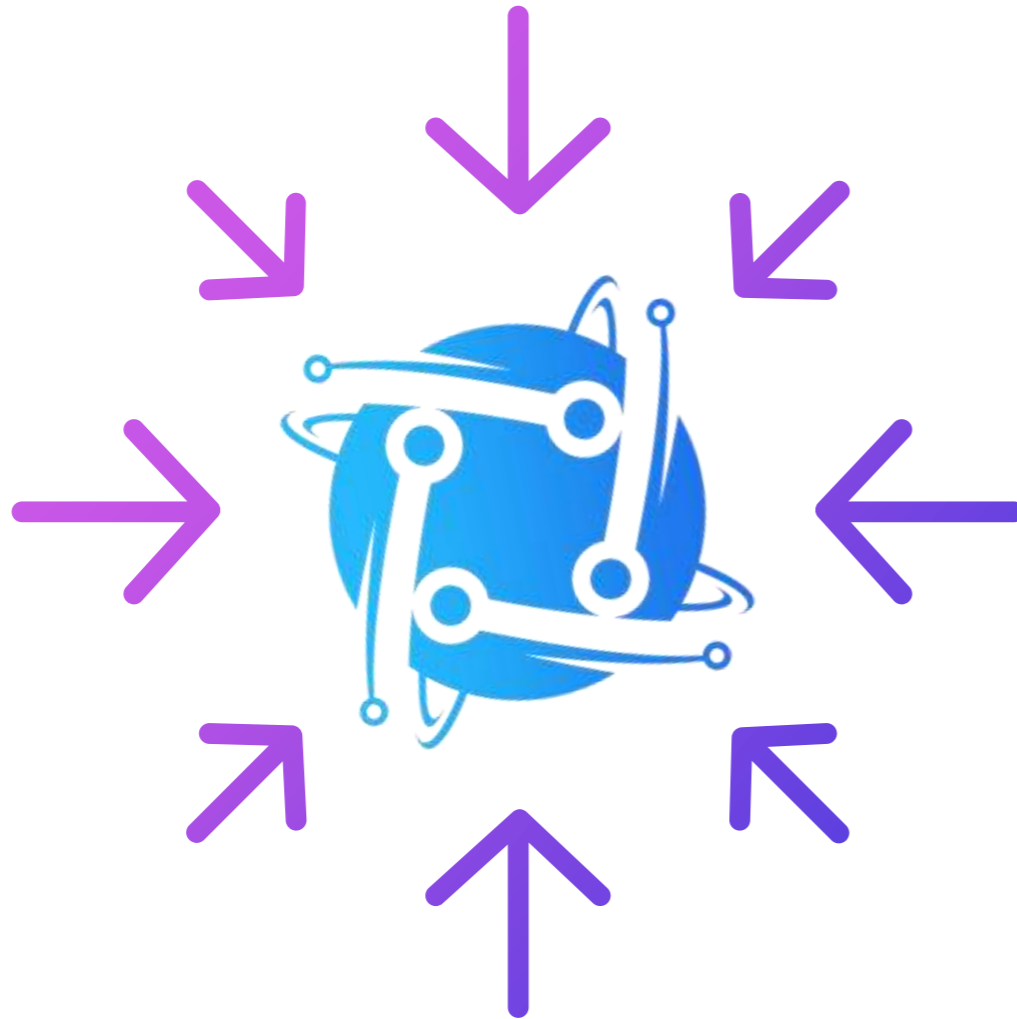
Above images are actual snapshots of the current live website of the project.

Website was registered on 12/06/2021, registration expires 12/06/2022.

**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

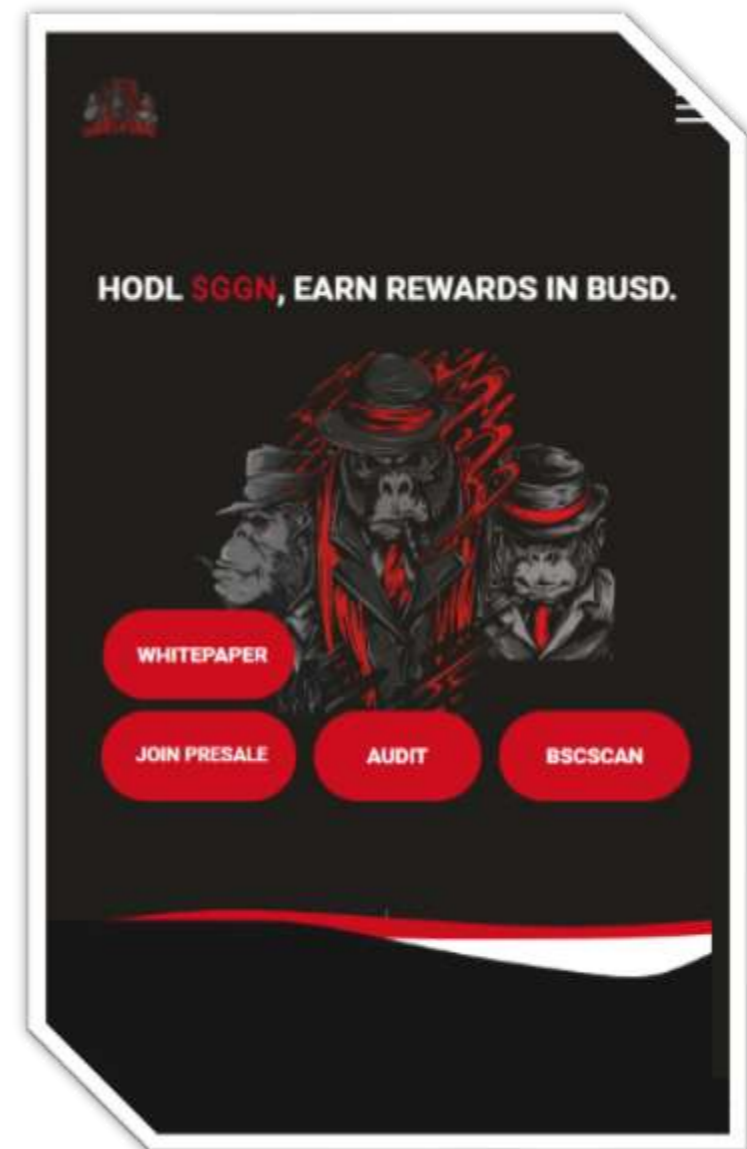
The website contained no JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

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: gorillagang.net

Issued by: cPanel, Inc

Valid Until: 03/08/2022



## CONTACT EMAIL

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

[Contact](mailto:contact@gorillagang.net)

**contact@gorillagang.net**



## SPAM / MALWARE / POPUPS

No malware found

No injected spam found

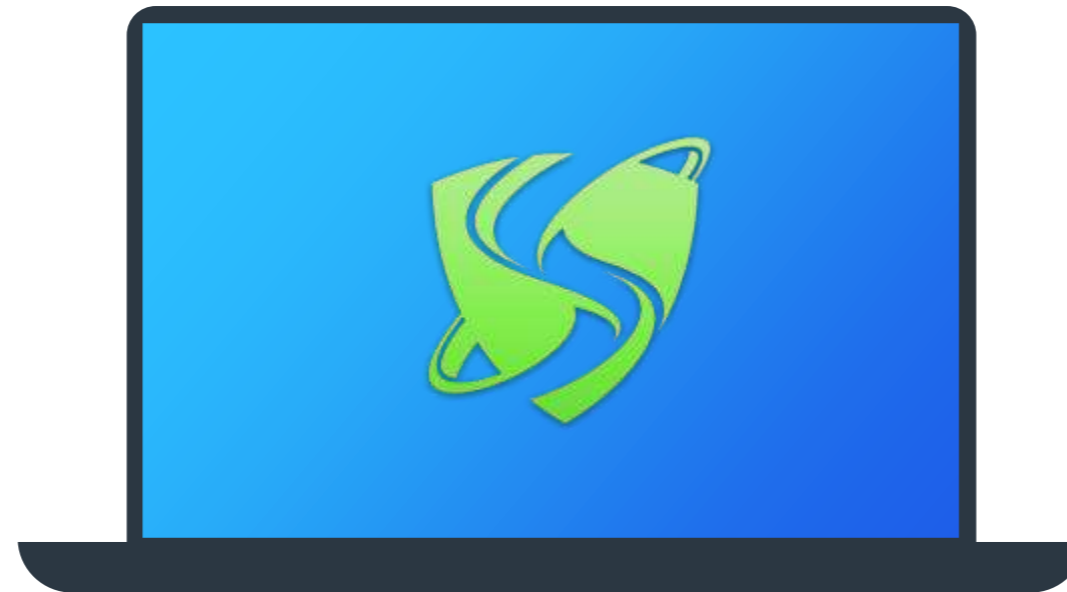
No internal server errors

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](#)



[Instagram](#)



[Facebook](#)

✓ **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](#)

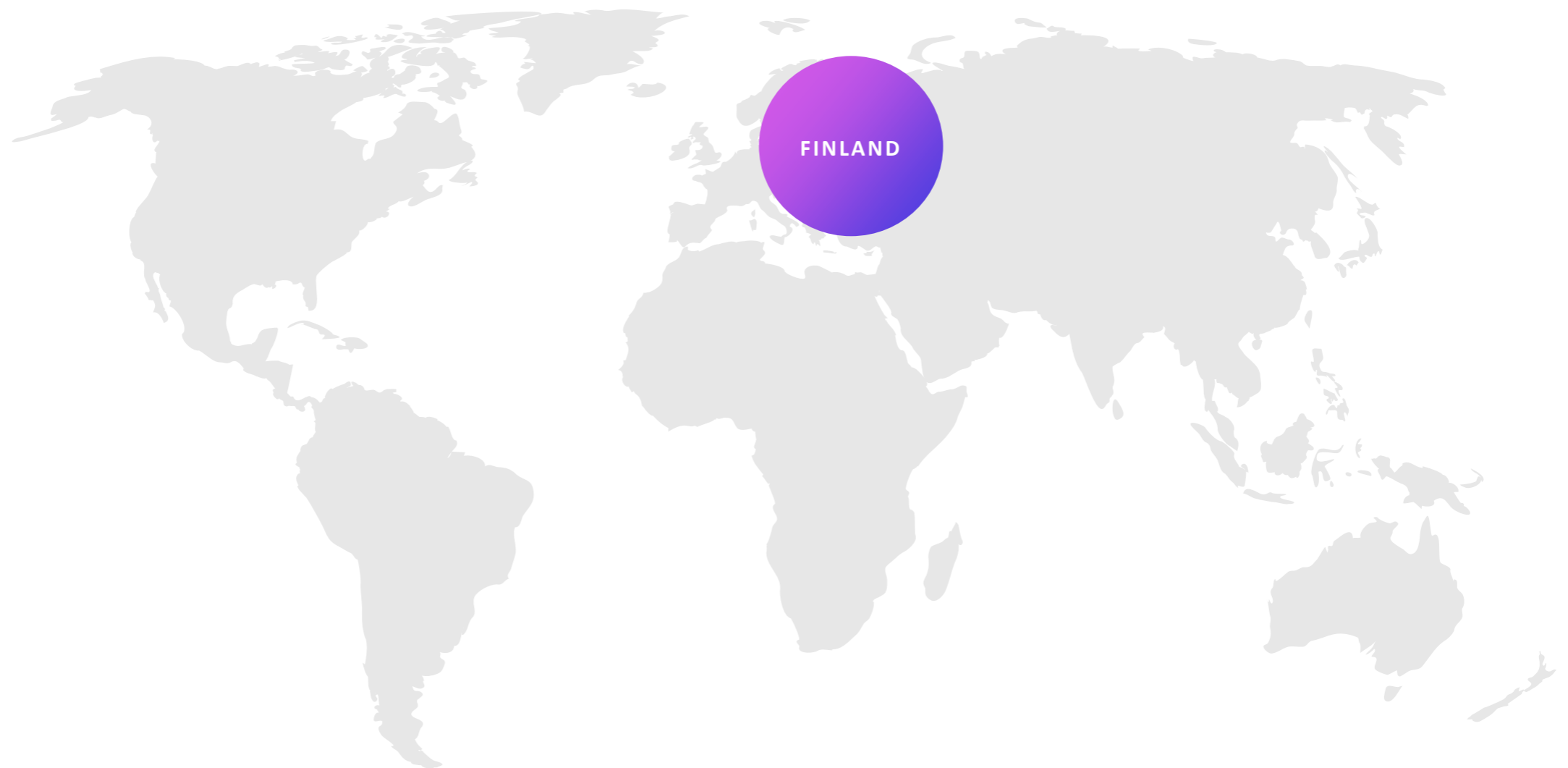
Gorilla gang Top 100 Token Holders

Source: BscScan.com



# Location Audit

The project has declared their primary location to be Finland.



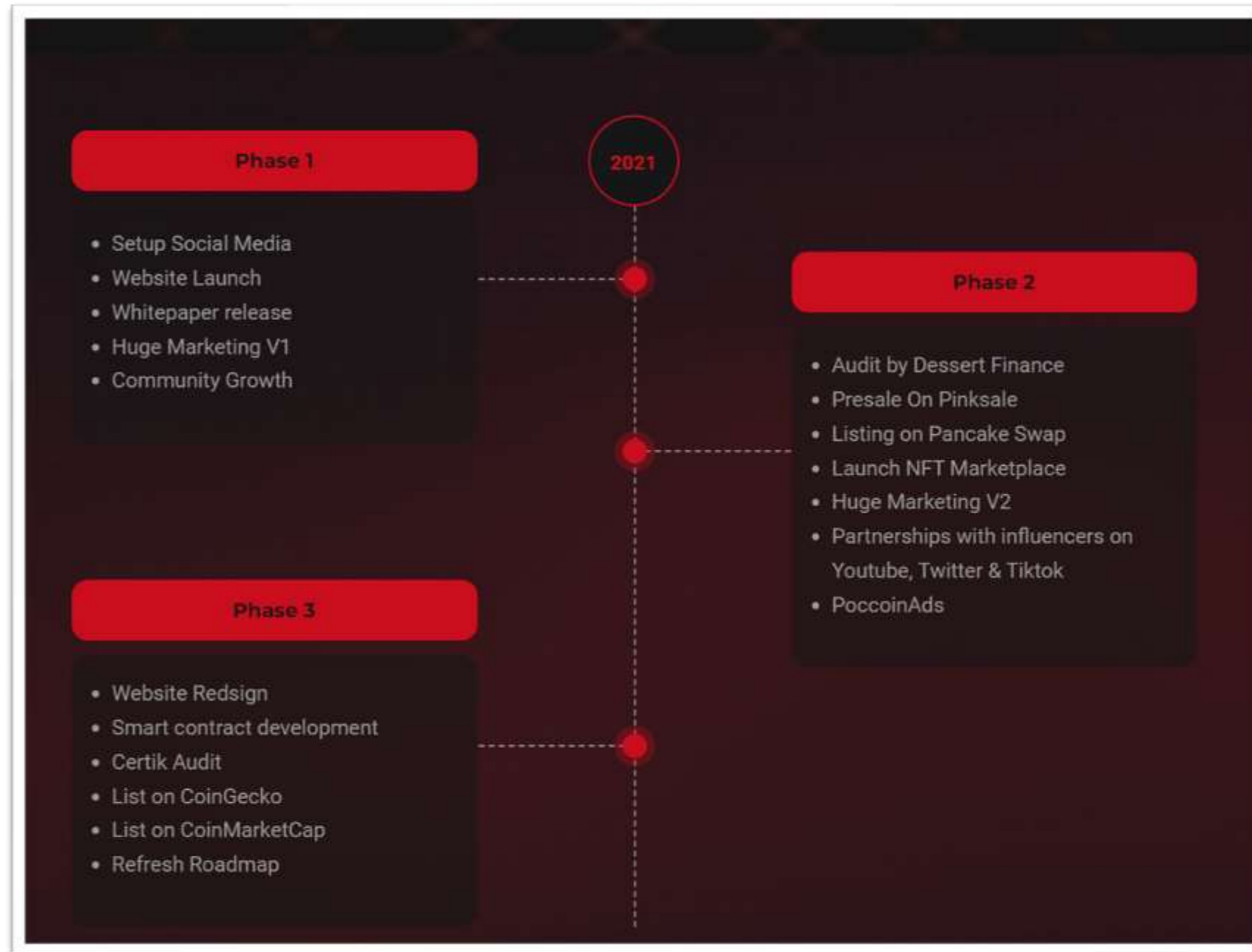
# Team Overview

*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 GORILLA GANG (GGN\$) 1 DSRT HAS BEEN SENT TO AUDITED PROJECT'S CONTRACT ADDRESS FOR VERIFICATION OF THIS AUDIT AT BLOCK NUMBER: **13618102**

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