



**DESSERT**  
FINANCE

**Hydrogen (Hydro)**

**BEP-20 Audit**

Performed at block **16513238**

PERFORMED BY DESSERT FINANCE  
FOR CONTRACT ADDRESS: 0xF1307CB297122F85bEC978794dbf96b055233a73

## 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.



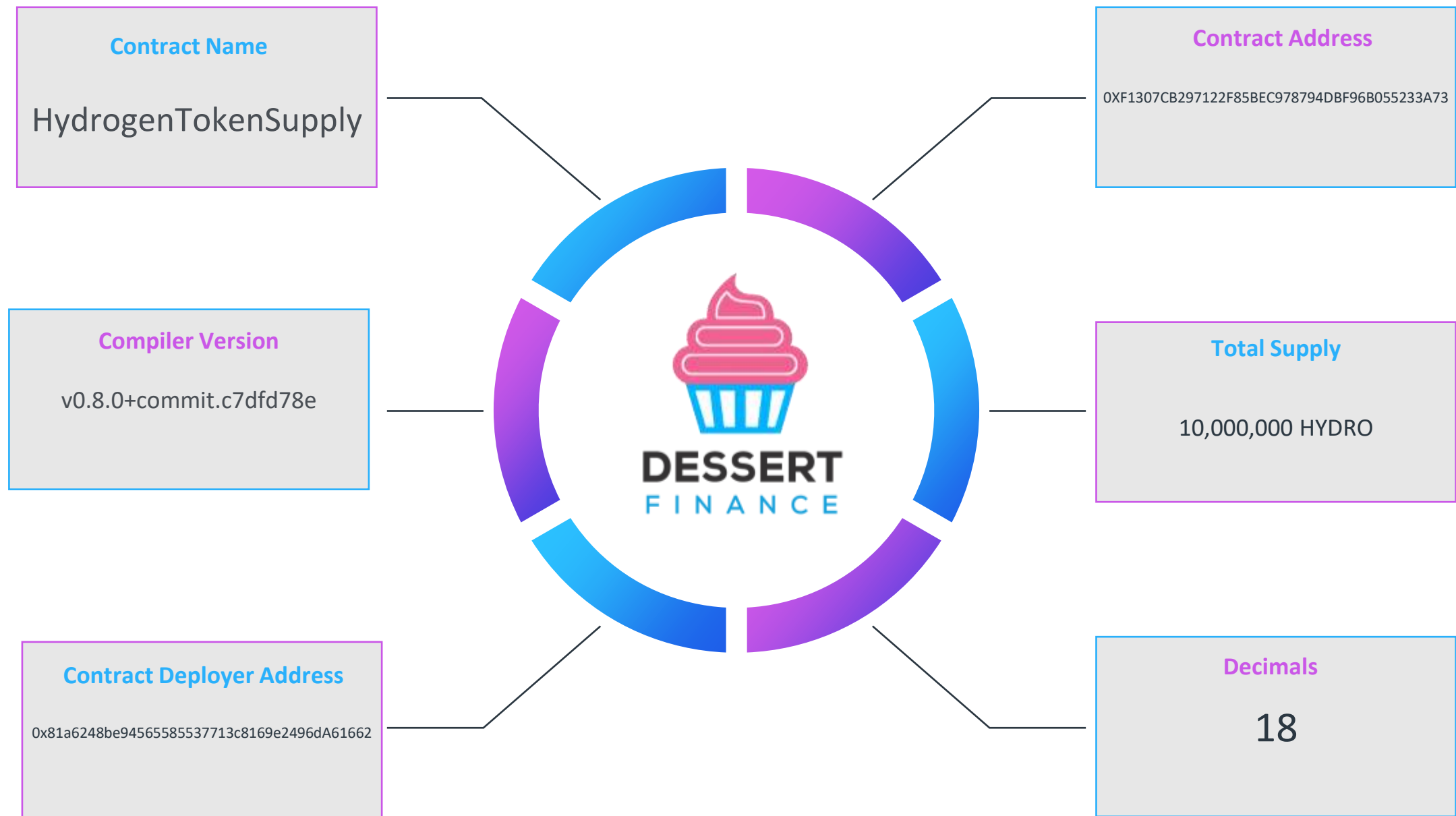


# Table of Contents



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 – Contract Ownership
5. Contract Code Audit – Owner Accessible Functions
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 Hydrogen (Hydro)

```
pragma solidity ^0.8.0;
import "@openzeppelin/contracts/token/ERC20/ERC20.sol";

contract HydrogenTokenSupply is ERC20 {
    constructor(uint256 initialSupply) ERC20("Hydrogen", "Hydro") {
        _mint(msg.sender, initialSupply * 10 ** 18);
    }
}

// SPDX-License-Identifier: MIT
// OpenZeppelin Contracts v4.8.0 (token/ERC20/ERC20.sol)

pragma solidity ^0.8.0;

import "./ERC20.sol";
import "./extensions/IERC20Metadata.sol";
import "../utils/Context.sol";

/**
 * @dev Implementation of the IERC20 interface.
 *
 * This implementation is agnostic to the way tokens are created. This means
 * that a supply mechanism has to be added in a derived contract using {_mint}.
 * For a generic mechanism see IERC20PresetMinterPasser.
 *
 * TIP: For a detailed writeup see our guide
 * https://forum.openzeppelin.com/t/how-to-implement-erc20-supply-mechanisms/220
 * to implement supply mechanisms.
 *
 * We have followed general OpenZeppelin Contracts guidelines: functions revert
 * instead returning false on failure. This behavior is nonetheless
 * conventional and does not conflict with the expectations of ERC20
 * applications.
 *
 * Additionally, an [Approval] event is emitted on calls to {transferFrom}.
 * This allows applications to reconstruct the allowance for all accounts just
 * by listening to said events. Other implementations of the TIP may not emit
 * these events, as it isn't required by the specification.
 *
 * Finally, the non-standard [decreaseAllowance] and [increaseAllowance]
 * functions have been added to mitigate the well-known issues around setting
 * allowances. See IERC20Approval.
 */
contract ERC20 is Context, IERC20, IERC20Metadata {
    mapping(address => uint256) private _balances;

    mapping(address => mapping(address => uint256)) private _allowances;

    uint256 private _totalSupply;

    string private _name;
    string private _symbol;

    /**
     * @dev Sets the values for {name} and {symbol}.
     *
     * The default value of {decimals} is 18. To select a different value for
     * {decimals} you should overload it.
     *
     * All two of these values are immutable: they can only be set once during
     * construction.
     */
    constructor(string memory name_, string memory symbol_) {
        _name = name_;
        _symbol = symbol_;
    }
}
```

## Contract Address

0xF1307CB297122F85bEC978794dbf96b055233a73

## TokenTracker

Hydrogen (Hydro)

## Contract Creator

0x81a6248be94565585537713c8169e2496da61662

## Source Code

Contract Source Code Verified

## Contract Name

HydrogenTokenSupply

## Other Settings

default evmVersion

## Compiler Version

v0.8.0+commit.c7dfd78e

## Optimization Enabled

Yes with 1 runs

Code is truncated to fit the constraints of this document.

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

# 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 – Contract Ownership

This is not an Ownable Token.



The contract is not an ownable contract.



# 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 Hydro Tokens.

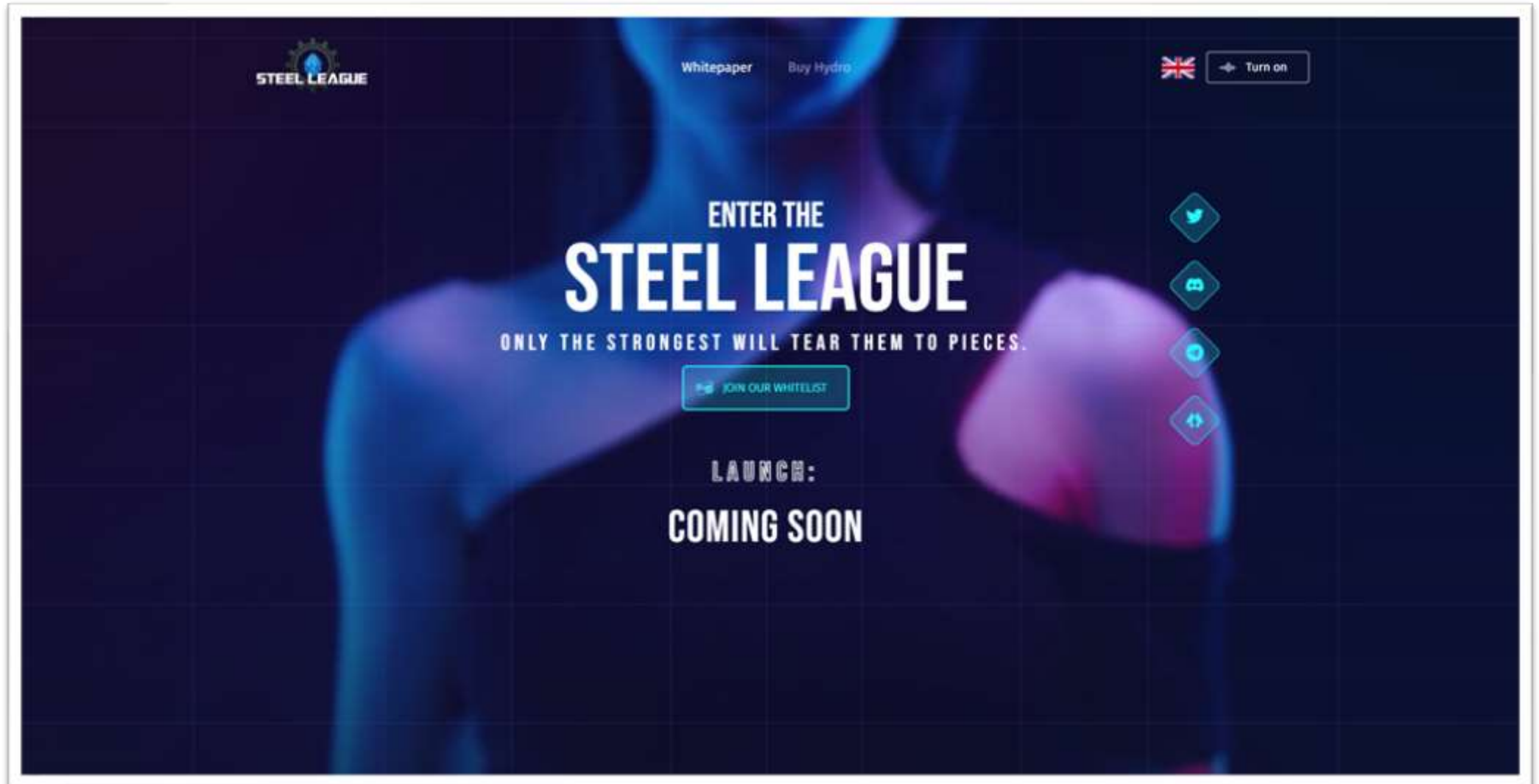


We do understand that sometimes mint functions are essential to the functionality of the project.

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

# Website Part 1 – Overview

## [www.steel-league.io](http://www.steel-league.io)



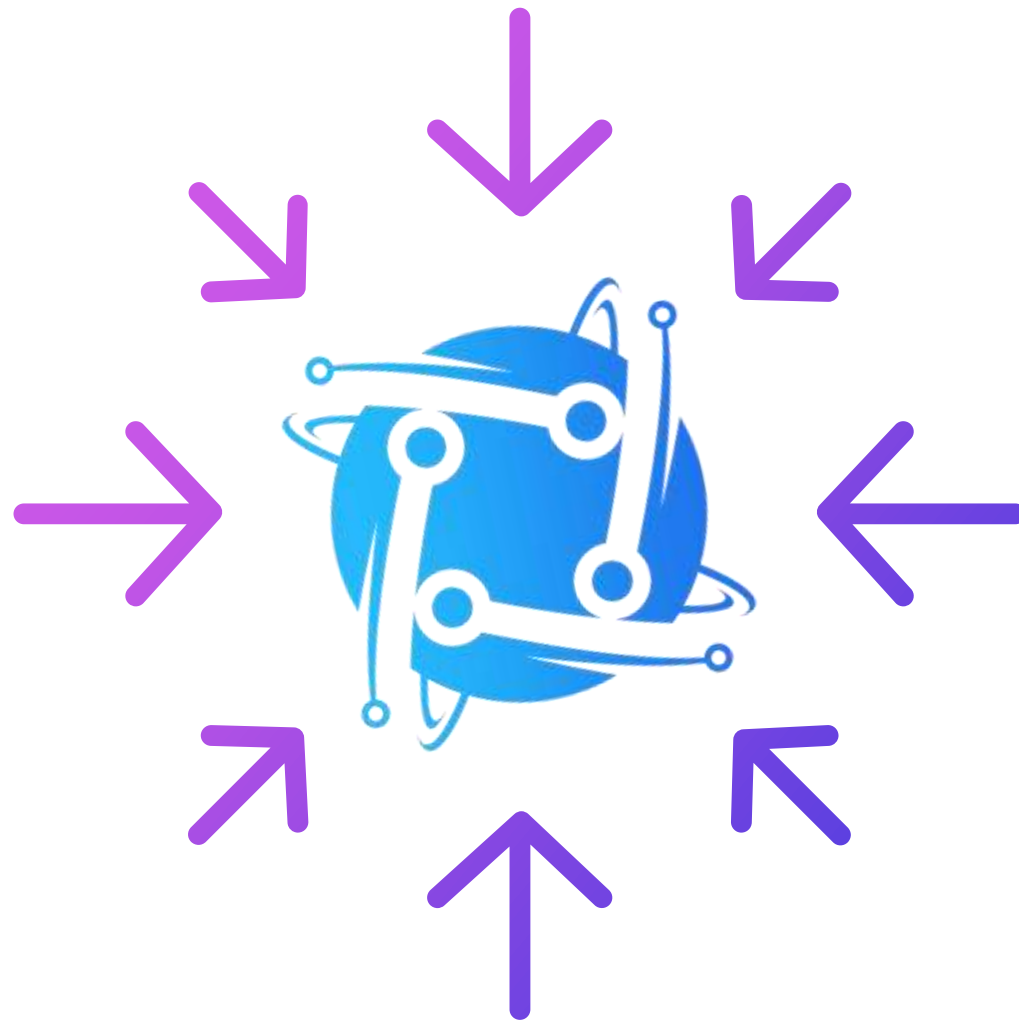
Above images are actual snapshots of the current live website of the project.

Website was registered on 12/01/2021, registration expires 12/01/2022.

**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

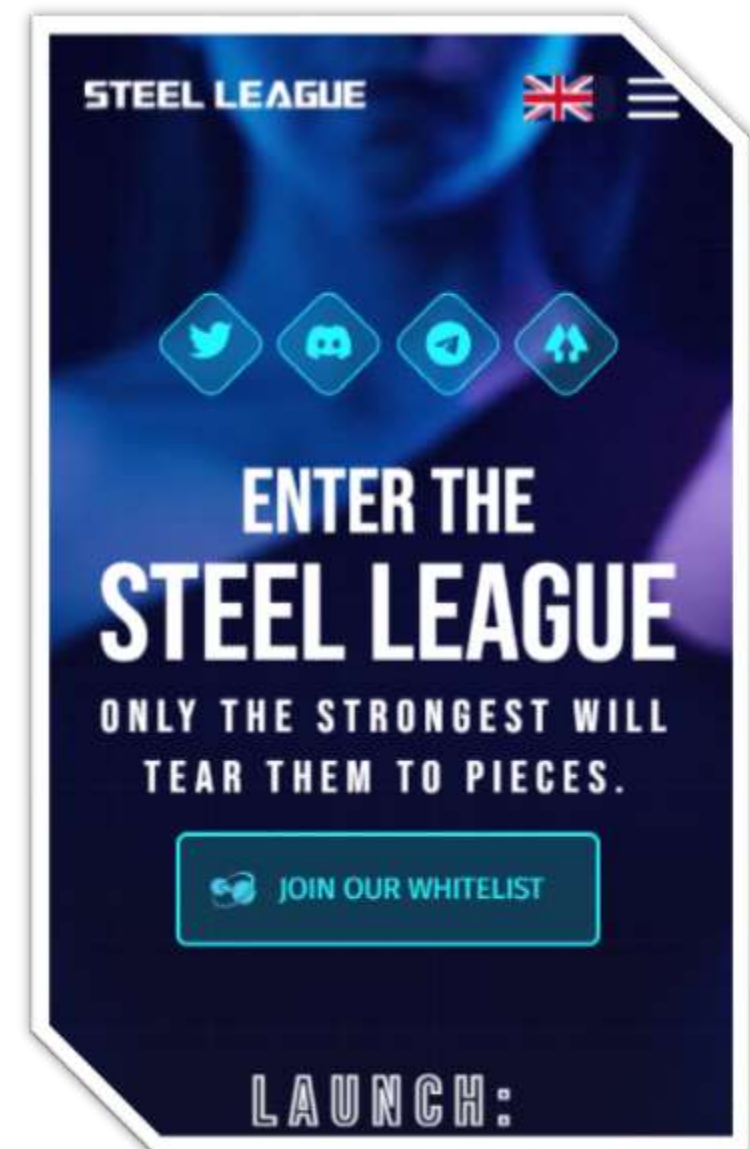
The website contained no JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

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: \*.steel-league.io

Issued by: R3

Valid Until: 05/09/2022



### CONTACT EMAIL

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

[Contact](mailto:team@steel-league.io)

**team@steel-league.io**



### SPAM / MALWARE / POPUPS

No malware found

No injected spam found

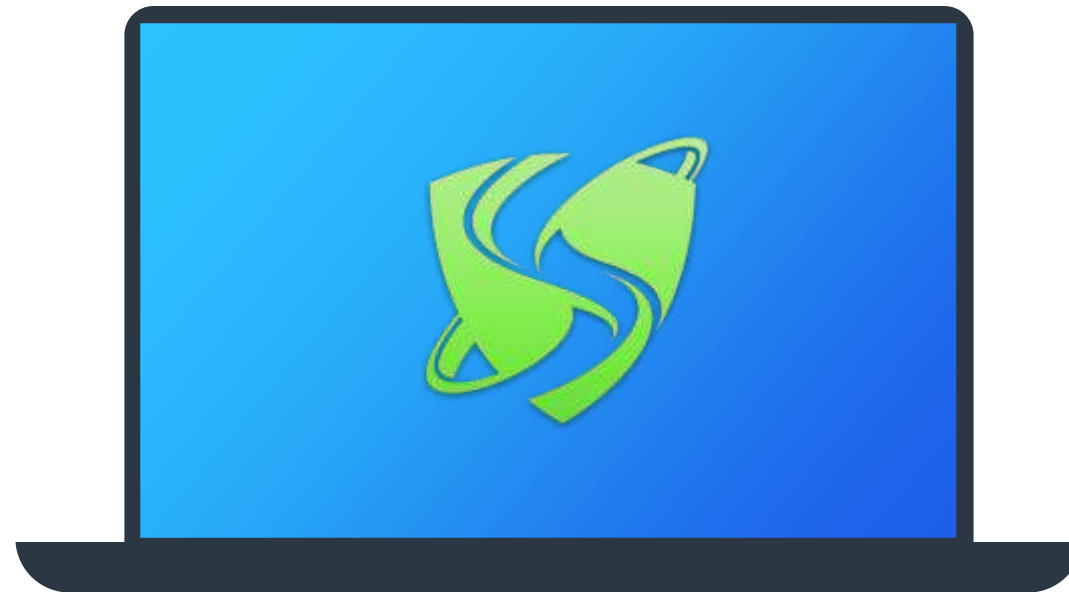
No internal server errors

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.

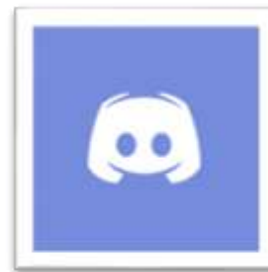
All links have been conveniently placed below.



[Twitter](#)



[Telegram](#)



[Discord](#)



[YouTube](#)

✓ At least 3 social media networks were found.

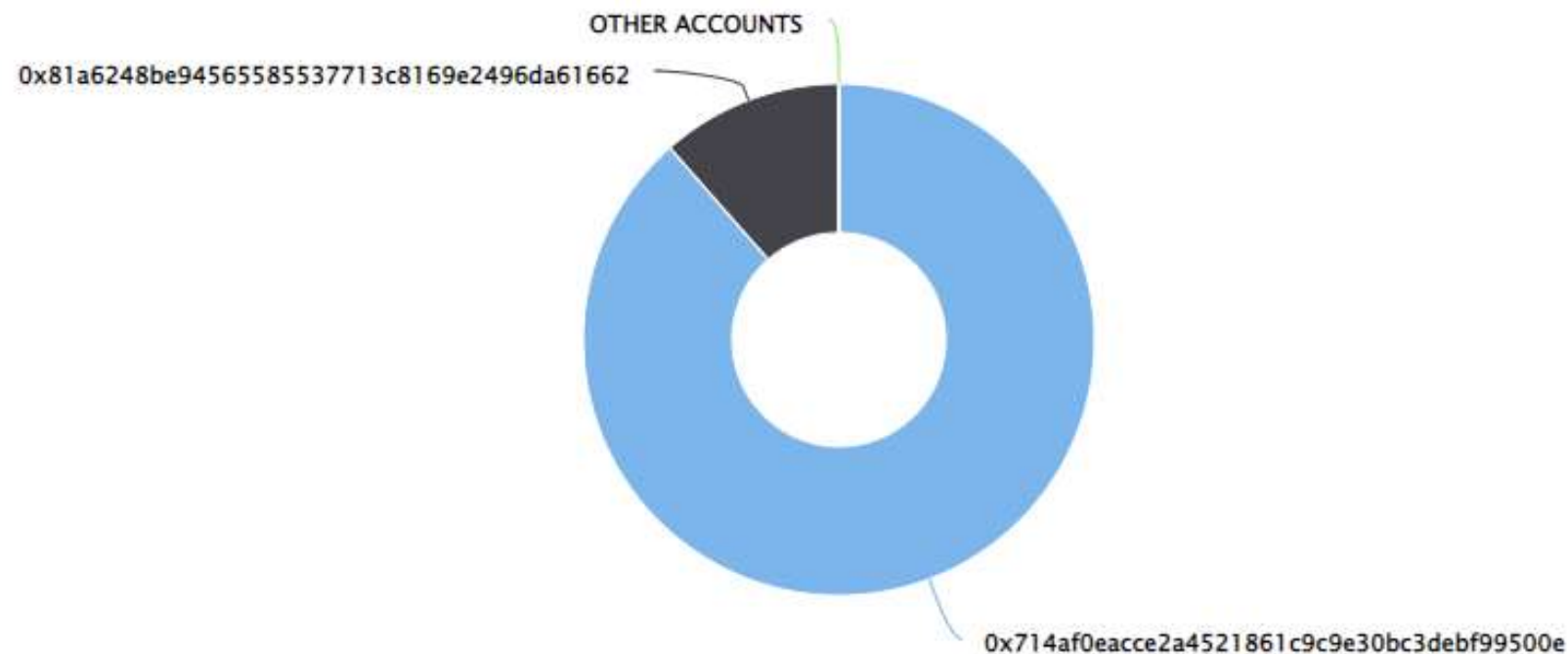
# Top Token Holders

The entire supply was in two wallets 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](#)

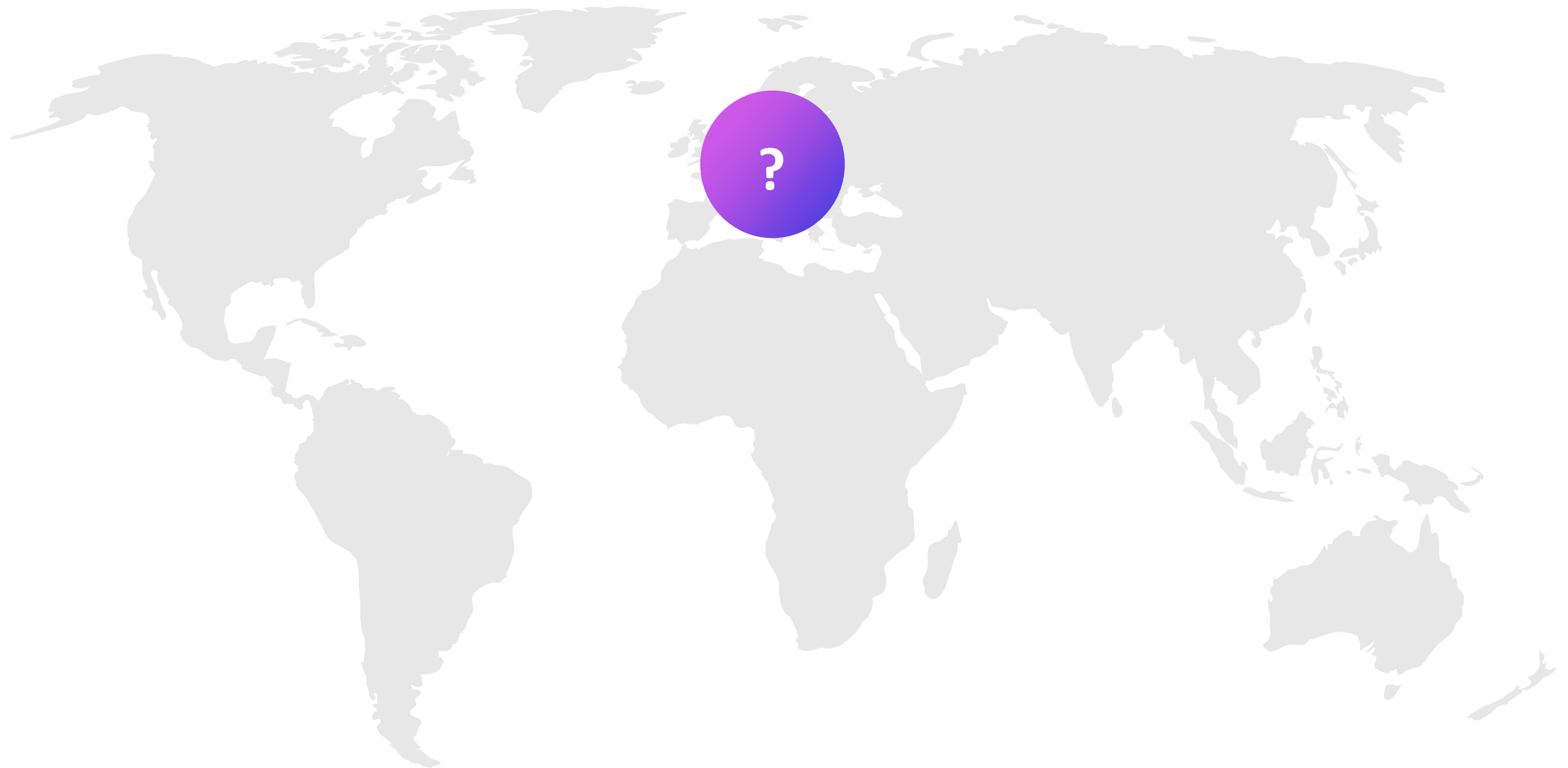
Hydrogen Top 100 Token Holders

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

## **Alex - CEO**

Our CEO Alex is a fourth-generation entrepreneur with over 10+ years of experience as a Software Developer and Game Programmer.

Alex joined the blockchain in 2016 as your typical neophyte; he quickly grew fond of the crypto-sphere as he began devoting more and more time to what was initially a simple hobby.

With the rise of blockchain gaming, Alex was able to bridge his two passions. Having been in the shade of various other projects within the blockchain, he decided to put his vision into action with the conception of Steel League.

## **Victor - Creative Director**

Victor, is Steel League's creative director and leads our design and development team.

Starting his career as a Graphic Designer, his interest shifted to web design, and he eventually began learning front-end coding.

Felix is now putting his expertise to good use in the management of our team.

## **Morgan - Marketing Manager**

Morgan is our Marketing Manager and an integral part of the work behind the Steel league's scene.

His role is the backbone of our project. Before joining our team, Morgan honed his skills in the commercial insurance industry.

## **Lucy - Graphic Designer**

Lucy is an experienced Graphic Designer with nearly ten years of experience in the industry. Her role is to design Steel League's visual identity and conceptualize our vision.

## **Chris - Artist**

Chris is an incredibly talented Cartoonist bringing his artistic flair to the table. He is the key to portraying the rich Steel League universe.

## **Henry - Developer / IT Specialist**

Prior to joining the Steel League team, Henry was a fully remote software test engineer.

He helped dozens of emerging companies and their hundreds of clients by ensuring software scalability and security.

## **Stuart - Full stack Web developer**

Having set himself as a full-time Full-stack Web dev following the crypto frenzy, Stuart's quick adaptation has granted him various skills allowing the absolute flawless functioning of Steel League's system.

## **Edward & Dinh - Blockchain developers**

The combined experience of both Edward and Dinh over the last four years will ensure the safety of Steel League's blockchain mechanisms.

## **Ben - Marketing Assistant**

The new addition to the team, actively assisting Morgan in the marketing department.

**A strong belief deeply unites us in our ability to provide an experience for newcomers in the robotic meta, as well as already set robotic aficionados.**



# 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 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 HYDROGEN (HYDRO) AT BLOCK NUMBER: **16513238**

THIS AUDIT IS ONLY VALID IF VIEWED ON [HTTPS://WWW.DSSERTSWAP.FINANCE](https://www.dessertswap.finance)

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