DESSERT FINANCE

Cryft (CRYFT)

BEP-20 Audit

Performed at block **21227524**

PERFORMED BY DESSERT FINANCE

TOKEN PROXY IMPLEMENTATION: 0XDF46D0A079354DD21B3B99D7943953038171C7F1

REFLECTION DISTRIBUTOR PROXY IMPLEMENTATION : 0X295B2362AA28662617A828AFC60393B02A64604C

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.

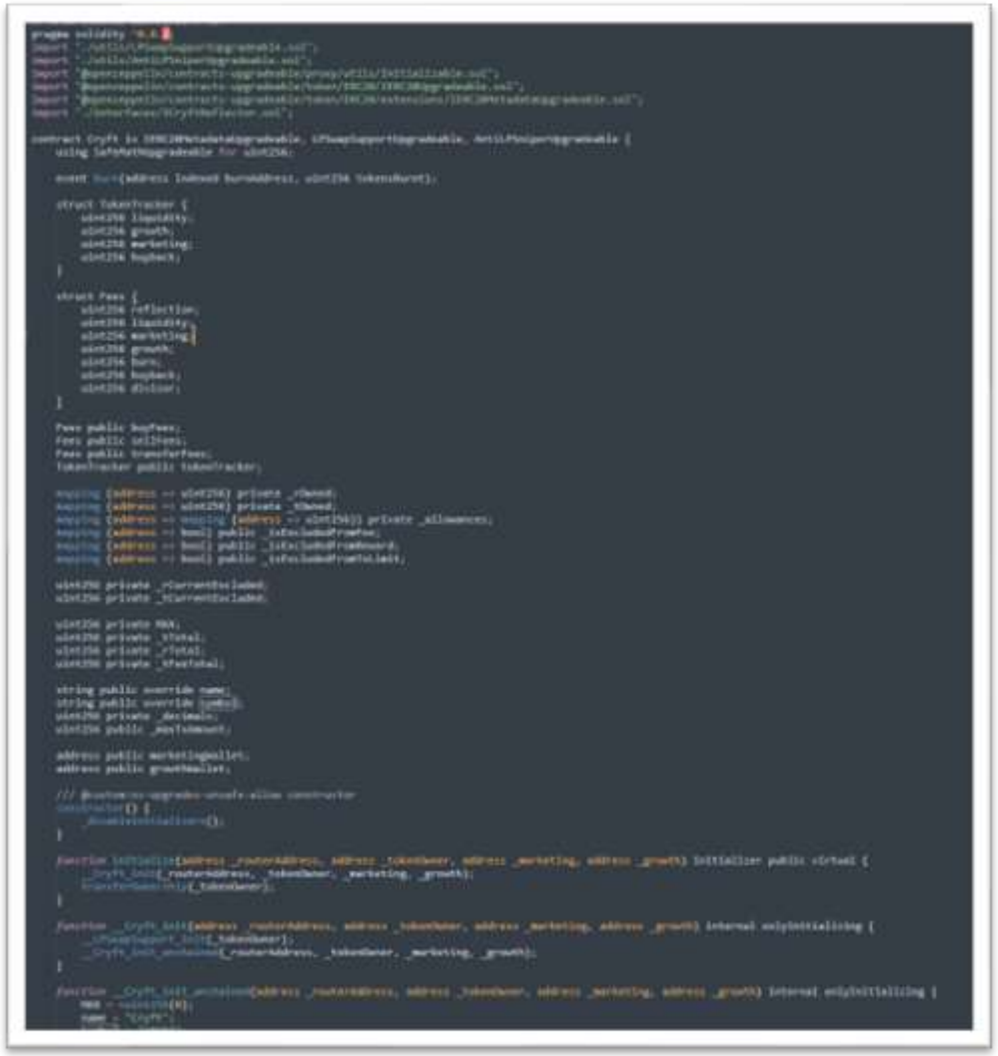
DessertDoxxed

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

BEP-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on Cryft (CRYFT)



```
pragma solidity ^0.8.2;
import "./utils/LPSwapSupportUpgradeable.sol";
import "./utils/AntiLPSniperUpgradeable.sol";
import "@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol";
import "@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol";
import "@openzeppelin/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol";
import "./interfaces/ICryftReflector.sol";

contract Cryft is IERC20MetadataUpgradeable, LPSwapSupportUpgradeable, AntiLPSniperUpgradeable {
    using SafeMathUpgradeable for uint256;

    event Burn(address indexed burnAddress, uint256 tokensBurnt);

    struct TokenTracker {
        uint256 liquidity;
        uint256 growth;
        uint256 marketing;
        uint256 buyback;
    }

    struct Fees {
        uint256 reflection;
        uint256 liquidity;
        uint256 marketing;
        uint256 growth;
        uint256 burn;
        uint256 buyback;
        uint256 divisor;
    }

    Fees public buyFees;
    Fees public sellFees;
    Fees public transferFees;
    TokenTracker public tokenTracker;

    mapping (address => uint256) private _rOwned;
    mapping (address => uint256) private _tOwned;
    mapping (address => mapping (address => uint256)) private _allowances;
    mapping (address => bool) public _isExcludedFromFee;
    mapping (address => bool) public _isExcludedFromReward;
    mapping (address => bool) public _isExcludedFromTxLimit;

    uint256 private _rCurrentExcluded;
    uint256 private _tCurrentExcluded;

    uint256 private MAX;
    uint256 private _tTotal;
    uint256 private _rTotal;
    uint256 private _tFeeTotal;

    string public override name;
    string public override symbol;
    uint256 private _decimals;
    uint256 public _maxTxAmount;

    address public marketingWallet;
    address public growthWallet;

    /// @custom:oz-upgrades-unsafe-allow constructor
    constructor() {
        _disableInitializers();
    }

    function initialize(address _routerAddress, address _tokenOwner, address _marketing, address _growth) initializer public virtual {
        __Cryft_init(_routerAddress, _tokenOwner, _marketing, _growth);
        transferOwnership(_tokenOwner);
    }

    function __Cryft_init(address _routerAddress, address _tokenOwner, address _marketing, address _growth) internal onlyInitializing {
        __LPSwapSupport_init(_tokenOwner);
        __Cryft_init_unchained(_routerAddress, _tokenOwner, _marketing, _growth);
    }

    function __Cryft_init_unchained(address _routerAddress, address _tokenOwner, address _marketing, address _growth) internal onlyInitializing {
        MAX = ~uint256(0);
        name = "Cryft";
```

Contract Address
0xDF46d0A079354DD21B3B99D7943953038171c7F1

TokenTracker
Cryft (CRYFT)

Contract Creator
0x73dd56e21458dc250Ad7F160035Ff35721A059e1

Source Code
Solidity

Contract Name
Cryft

Compiler Version
v0.8.2+commit.661d1103

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

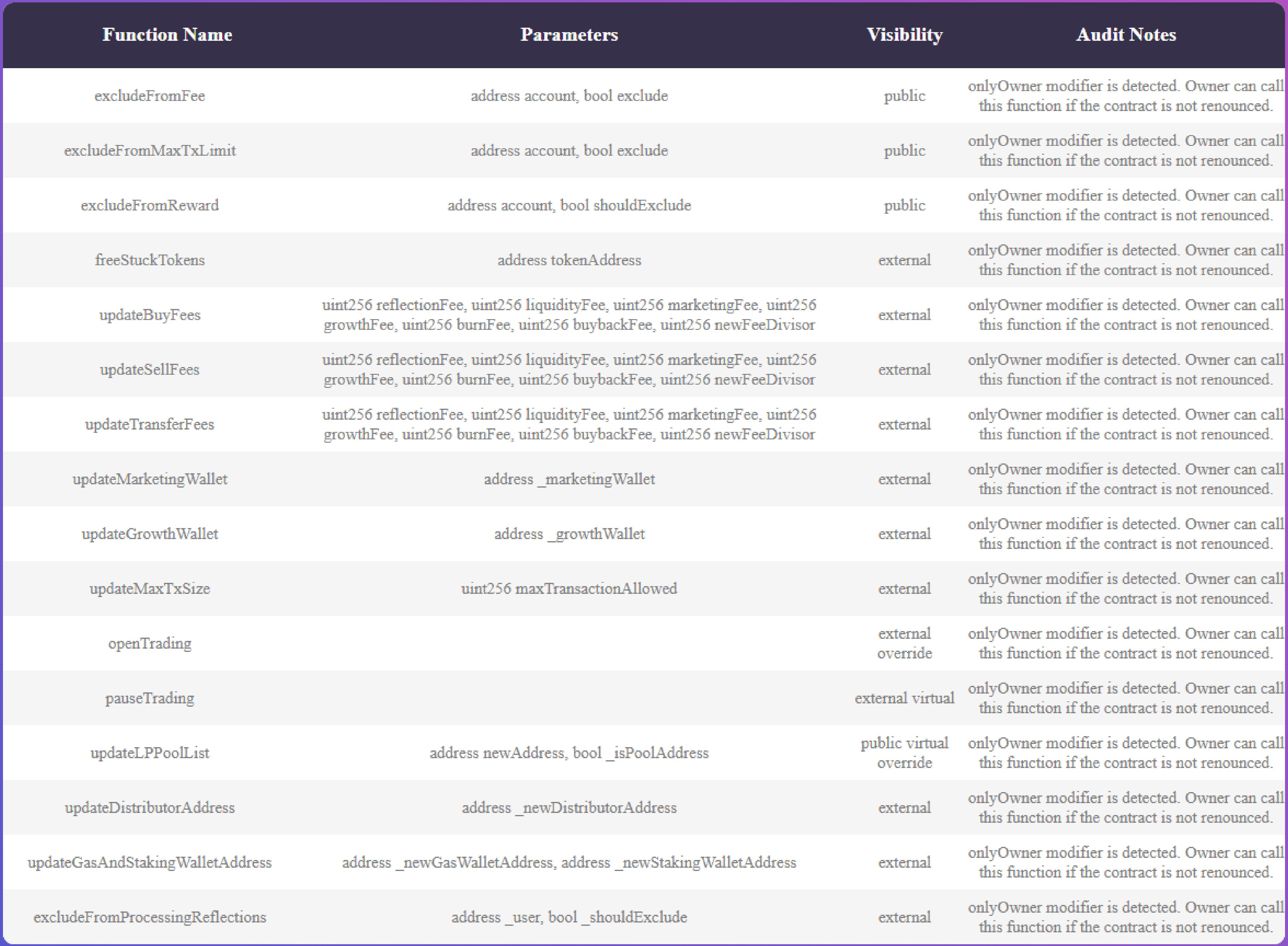
BEP-20 Contract Code Audit – Vulnerabilities Checked

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 - Cryft

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

Contract Code Audit – Owner Accessible Functions



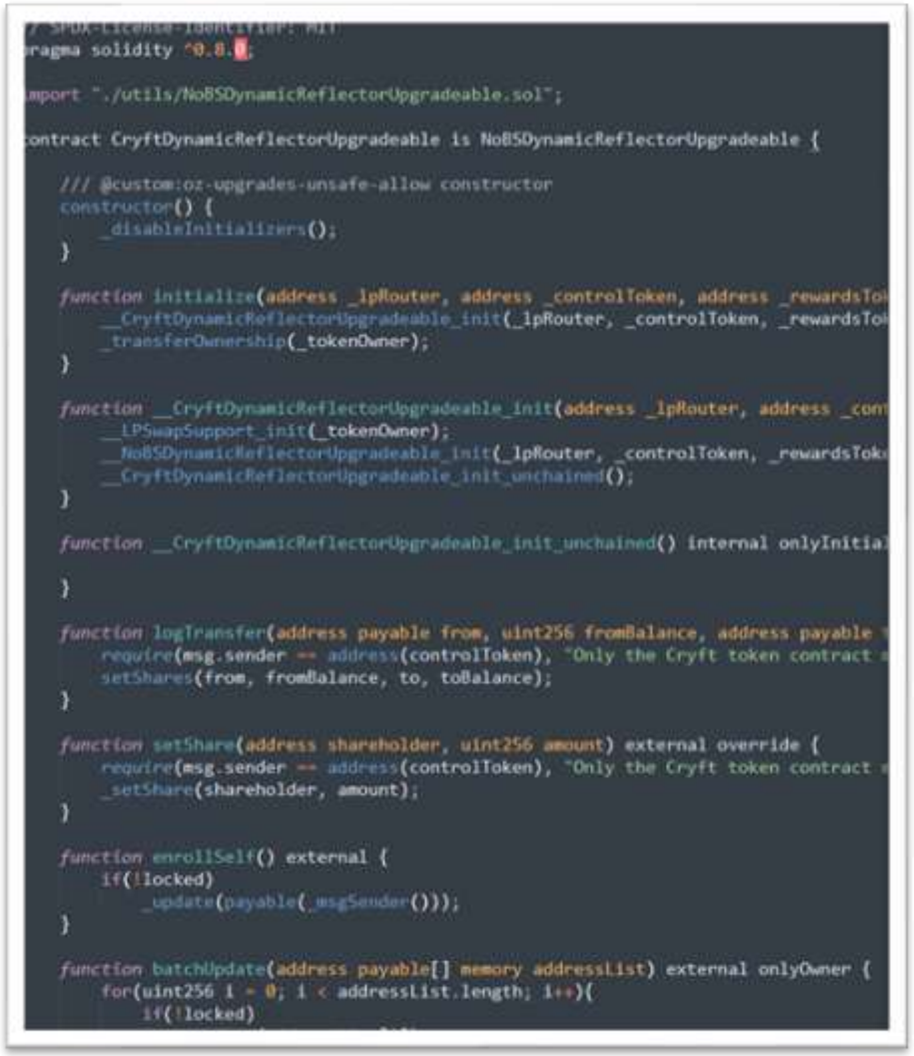
Function Name	Parameters	Visibility	Audit Notes
excludeFromFee	address account, bool exclude	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromMaxTxLimit	address account, bool exclude	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromReward	address account, bool shouldExclude	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
freeStuckTokens	address tokenAddress	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateBuyFees	uint256 reflectionFee, uint256 liquidityFee, uint256 marketingFee, uint256 growthFee, uint256 burnFee, uint256 buybackFee, uint256 newFeeDivisor	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateSellFees	uint256 reflectionFee, uint256 liquidityFee, uint256 marketingFee, uint256 growthFee, uint256 burnFee, uint256 buybackFee, uint256 newFeeDivisor	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateTransferFees	uint256 reflectionFee, uint256 liquidityFee, uint256 marketingFee, uint256 growthFee, uint256 burnFee, uint256 buybackFee, uint256 newFeeDivisor	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateMarketingWallet	address _marketingWallet	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateGrowthWallet	address _growthWallet	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateMaxTxSize	uint256 maxTransactionAllowed	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
openTrading		external override	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
pauseTrading		external virtual	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateLPPoolList	address newAddress, bool _isPoolAddress	public virtual override	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateDistributorAddress	address _newDistributorAddress	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateGasAndStakingWalletAddress	address _newGasWalletAddress, address _newStakingWalletAddress	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromProcessingReflections	address _user, bool _shouldExclude	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- Cryft

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

BEP-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on CryftDynamicReflectorUpgradeable



```
/ SPDX-License-Identifier: MIT
ragma solidity ^0.8.0;

mport "./utils/NoBSDynamicReflectorUpgradeable.sol";

ontract CryftDynamicReflectorUpgradeable is NoBSDynamicReflectorUpgradeable {

    /// @custom:oz-upgrades-unsafe-allow constructor
    constructor() {
        _disableInitializers();
    }

    function initialize(address _lpRouter, address _controlToken, address _rewardsTok
        __CryftDynamicReflectorUpgradeable_init(_lpRouter, _controlToken, _rewardsTok
        _transferOwnership(_tokenOwner);
    }

    function __CryftDynamicReflectorUpgradeable_init(address _lpRouter, address _cont
        __LPSwapSupport_init(_tokenOwner);
        __NoBSDynamicReflectorUpgradeable_init(_lpRouter, _controlToken, _rewardsTok
        __CryftDynamicReflectorUpgradeable_init_unchained();
    }

    function __CryftDynamicReflectorUpgradeable_init_unchained() internal onlyInitial

    }

    function logTransfer(address payable from, uint256 fromBalance, address payable t
        require(msg.sender == address(controlToken), "Only the Cryft token contract c
        setShares(from, fromBalance, to, toBalance);
    }

    function setShare(address shareholder, uint256 amount) external override {
        require(msg.sender == address(controlToken), "Only the Cryft token contract c
        _setShare(shareholder, amount);
    }

    function enrollSelf() external {
        if(!locked)
            _update(payable(_msgSender()));
    }

    function batchUpdate(address payable[] memory addressList) external onlyOwner {
        for(uint256 i = 0; i < addressList.length; i++){
            if(!locked)
```

Contract Address
0x295b2362Aa28662617A828AFC60393B02A64604C

Contract Creator
0x73dd56e21458dc250Ad7F160035Ff35721A059e1

Source Code
Solidity

Contract Name
CryftDynamicReflectorUpgradeable

Compiler Version
v0.8.2+commit.661d1103

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

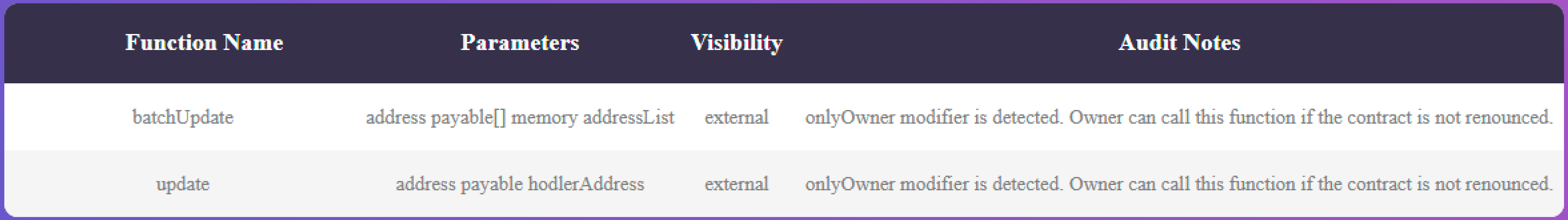
BEP-20 Contract Code Audit – Vulnerabilities Checked

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
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 - CryftDynamicReflectorUpgradeable

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

Contract Code Audit – Owner Accessible Functions



Function Name	Parameters	Visibility	Audit Notes
batchUpdate	address payable[] memory addressList	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
update	address payable hodlerAddress	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 - CryftDynamicReflectorUpgradeable

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.

Contract Code Audit – Mint Functions

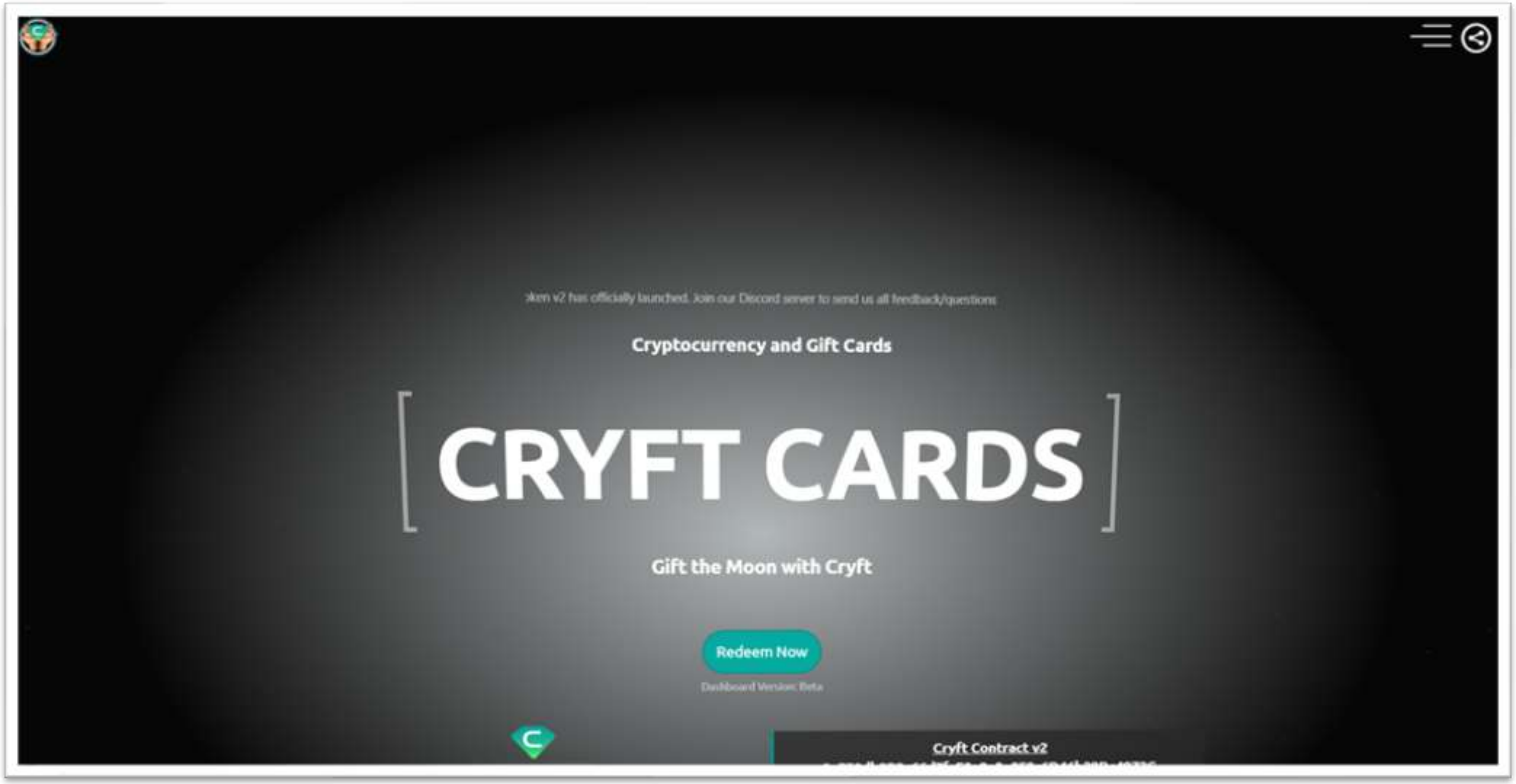
This Contract Cannot Mint New CRYFT Tokens.

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 DESKTOP AND MOBILE VIEW AUDIT

Website Part 1 – Overview
www.cryftcards.com



oken v2 has officially launched. Join our Discord server to send us all feedback/questions

Cryptocurrency and Gift Cards

CRYFT CARDS

Gift the Moon with Cryft

Dashboard Version: Beta

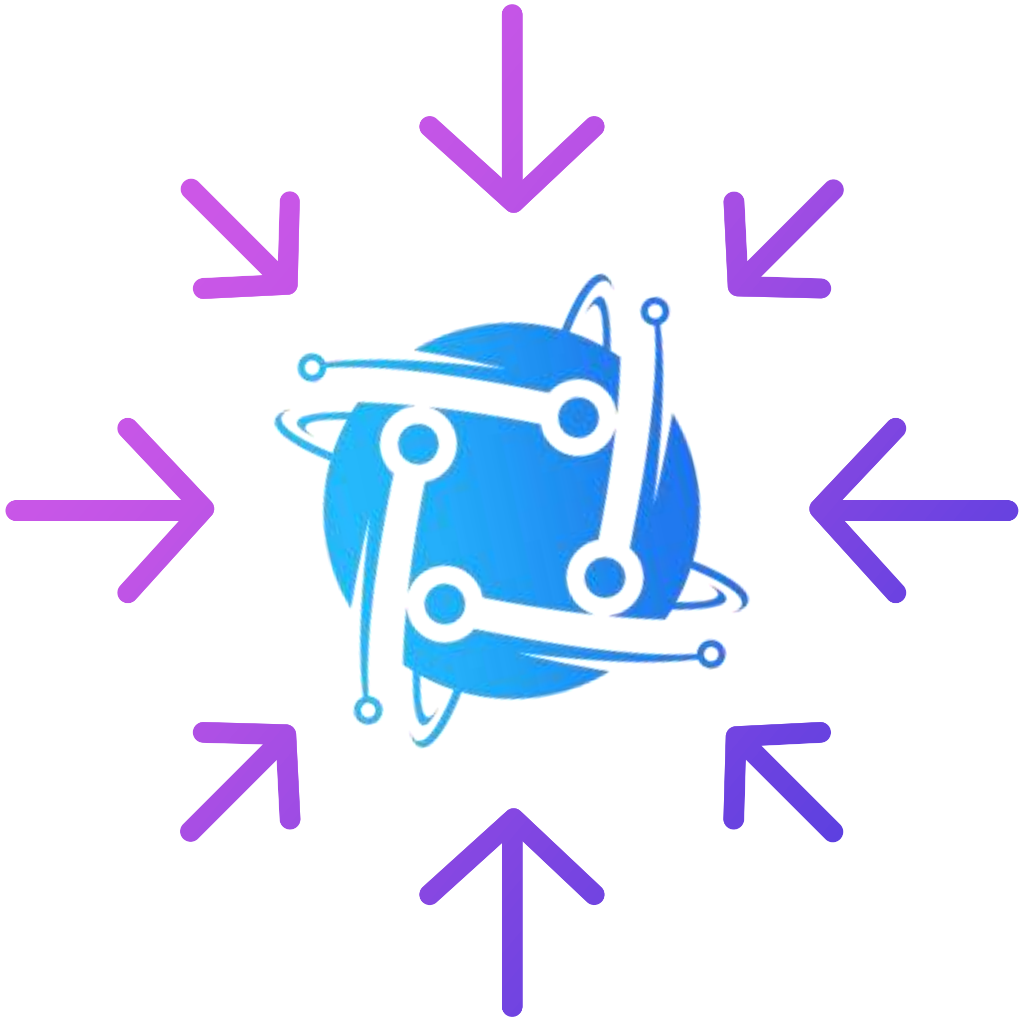
Cryft Contract v2

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

Website was registered on 07/27/2022, registration expires 07/27/2027.

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

Website Part 2 – Checklist



✓ **Mobile Friendly**

✓ **No JavaScript Errors**

✓ **Spell Check**

✓ **SSL Certificate**

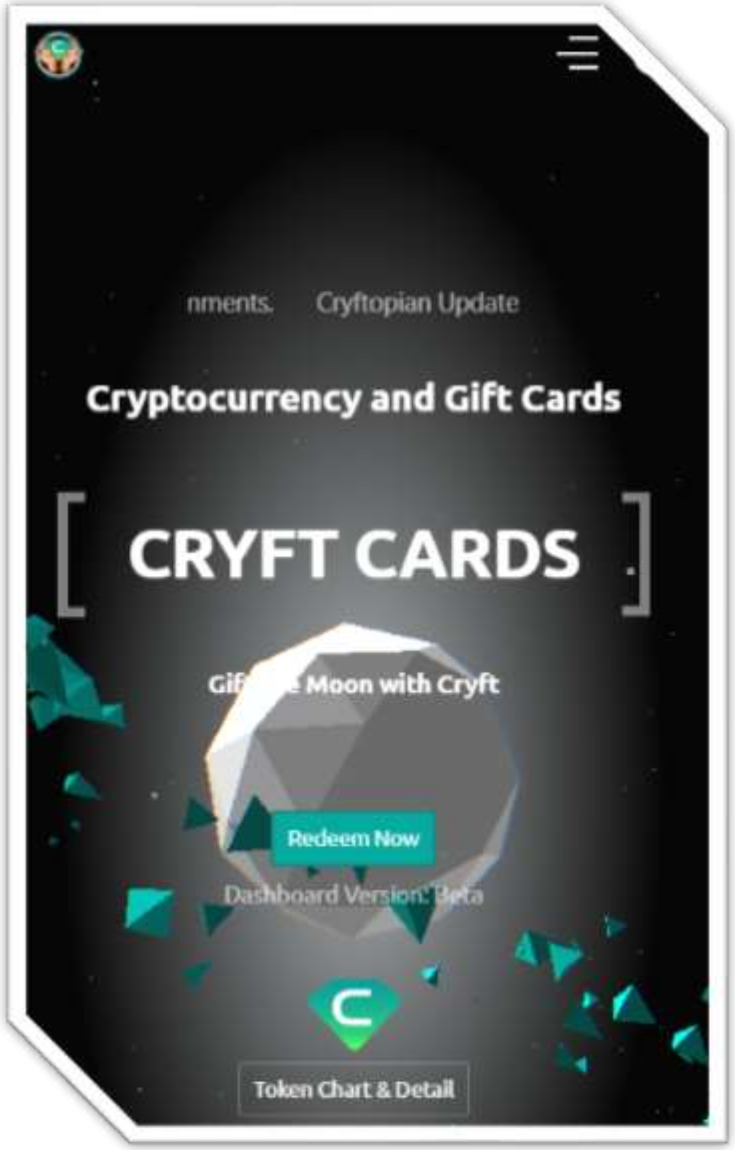
The website contained no JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

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.


nments. Cryftopian Update

Cryptocurrency and Gift Cards

CRYFT CARDS

Gif[illegible]e Moon with Cryft

Website Part 4 (GWS) – General Web Security

SSL CERTIFICATE

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

Offered to: cryftcards.com

Issued by: E1

Valid Until: 10/30/2022

✓

CONTACT EMAIL

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

Contact

help@cryftcards.com

✓

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.

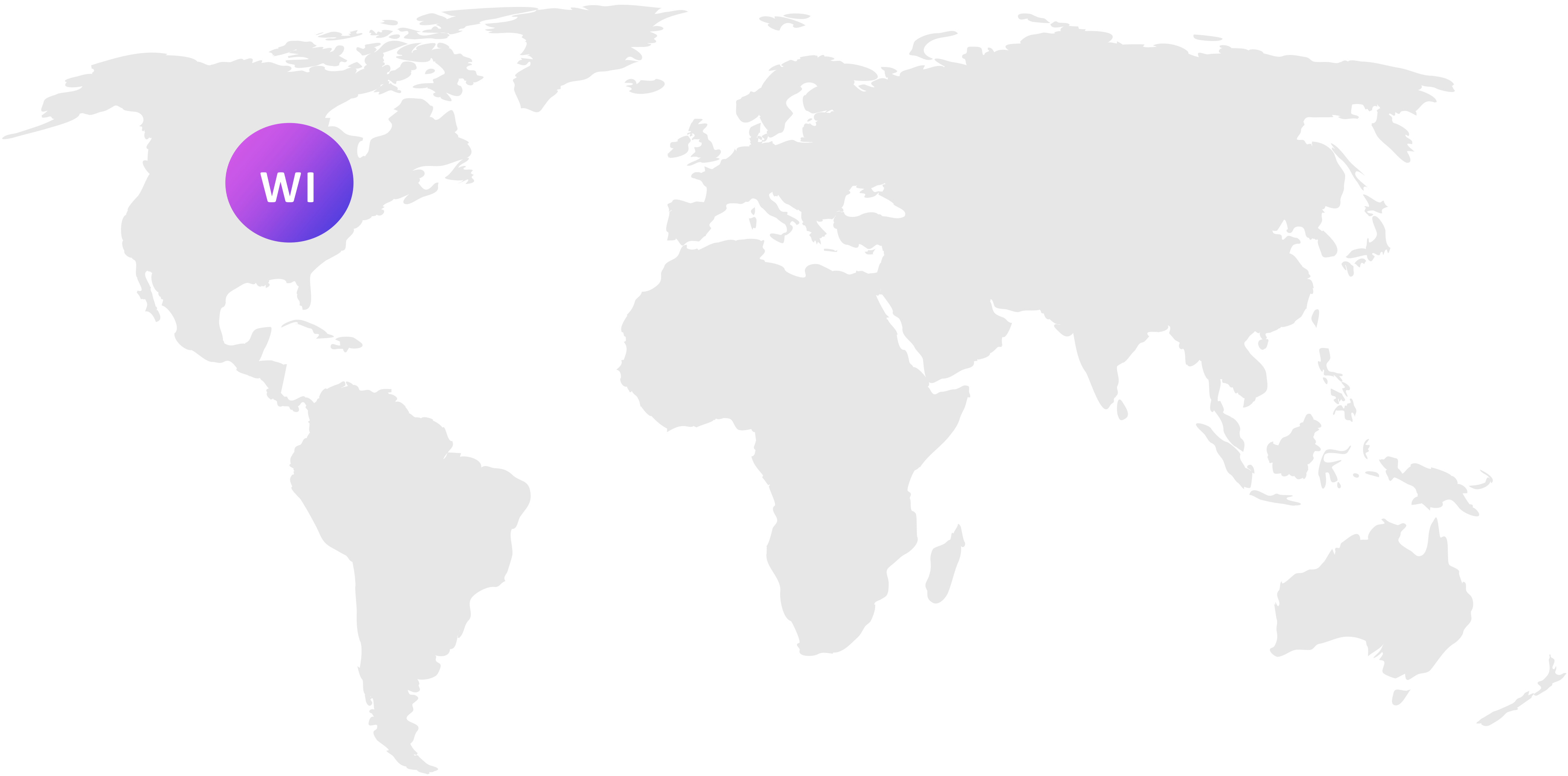
All links have been conveniently placed below.

Twitter Telegram Discord Facebook Instagram

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

Location Audit

The primary location for the team is Wisconsin, USA.



Team Overview

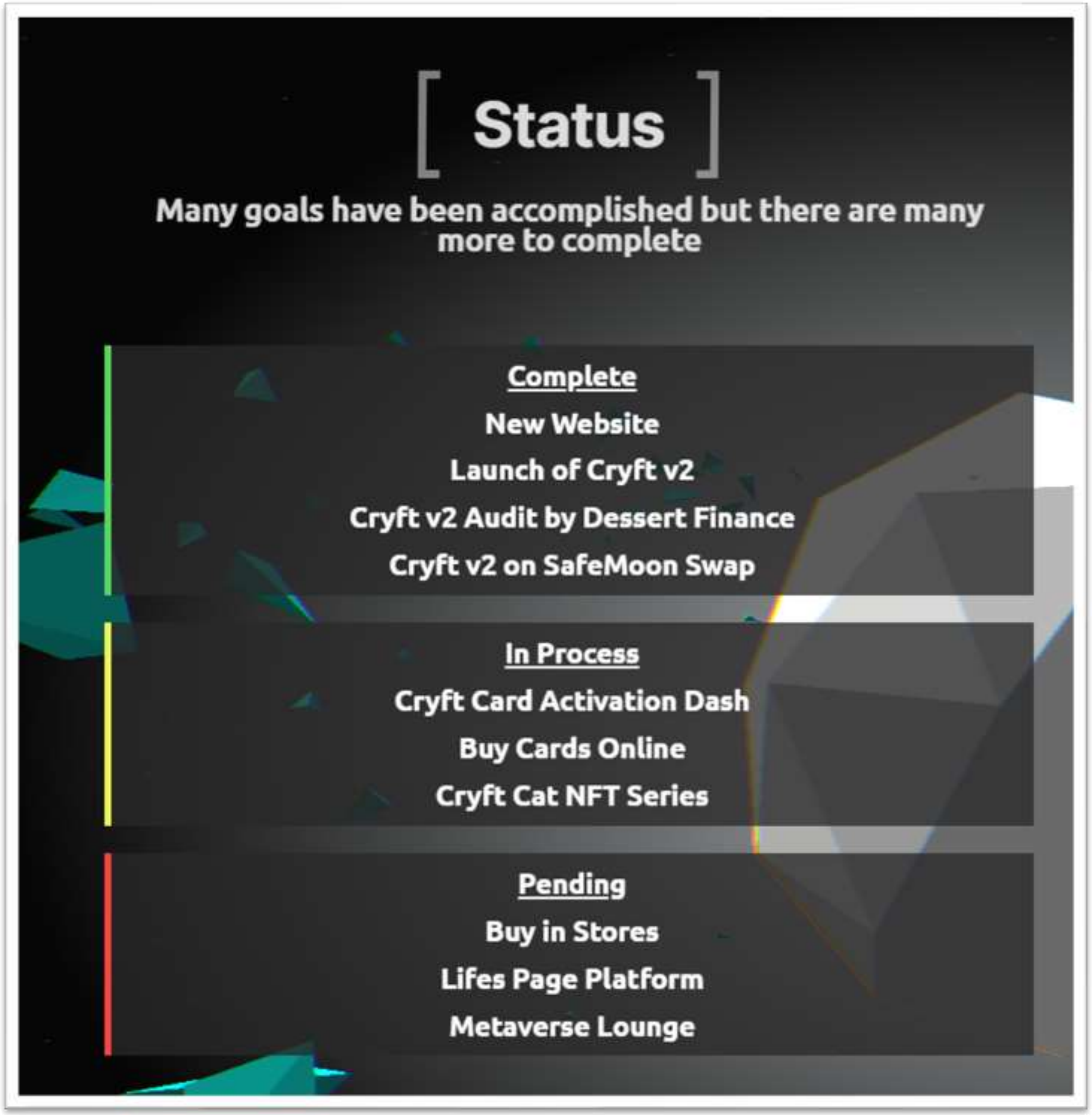
The following information about the projects team was found on the website.

[Team Members]

The success of Cryft LLC falls upon the shoulders of our top tier team

Roadmap

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



[Status]

Many goals have been accomplished but there are many more to complete

Complete

New Website

Launch of Cryft v2

Cryft v2 Audit by Dessert Finance

Cryft v2 on SafeMoon Swap

In Process

Cryft Card Activation Dash

Buy Cards Online

Cryft Cat NFT Series

Pending

Buy in Stores

Lifes Page Platform

Metaverse Lounge

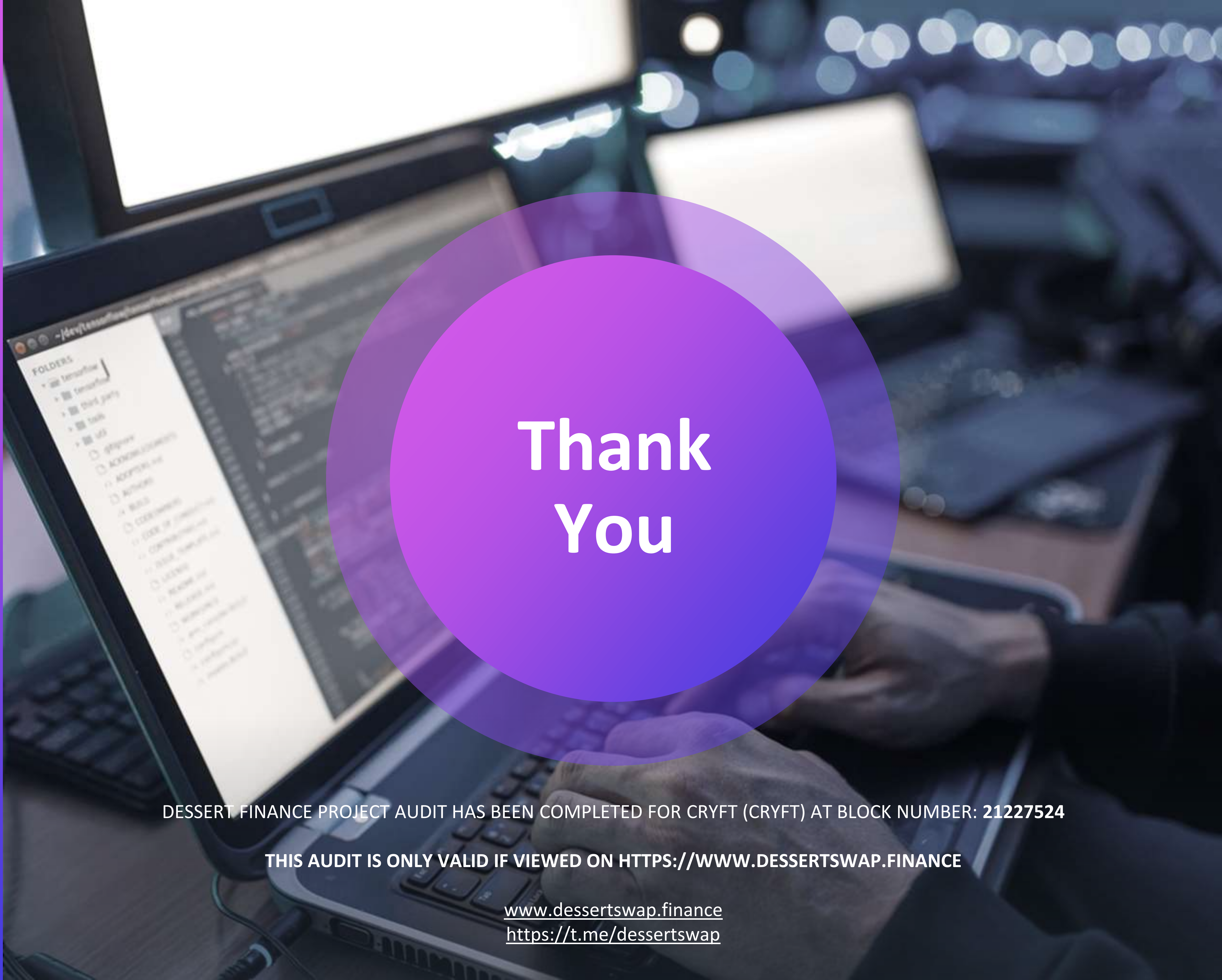
Disclaimer

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 CRYFT (CRYFT) AT BLOCK NUMBER: **21227524**

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

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