

DESSERT
FINANCE



Eden (EDEN)

HRC-20 Audit

Performed at block **19093181**

PERFORMED BY DESSERT FINANCE
FOR CONTRACT ADDRESSES:

ONE1K4GSVVYFWN8TU2V6PUZS2S6UG76QFWDXDWHCD6
ONE1LXXTRN7F48M90GZ8QQMZPL59FQD72L3GVC7DF0

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 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.

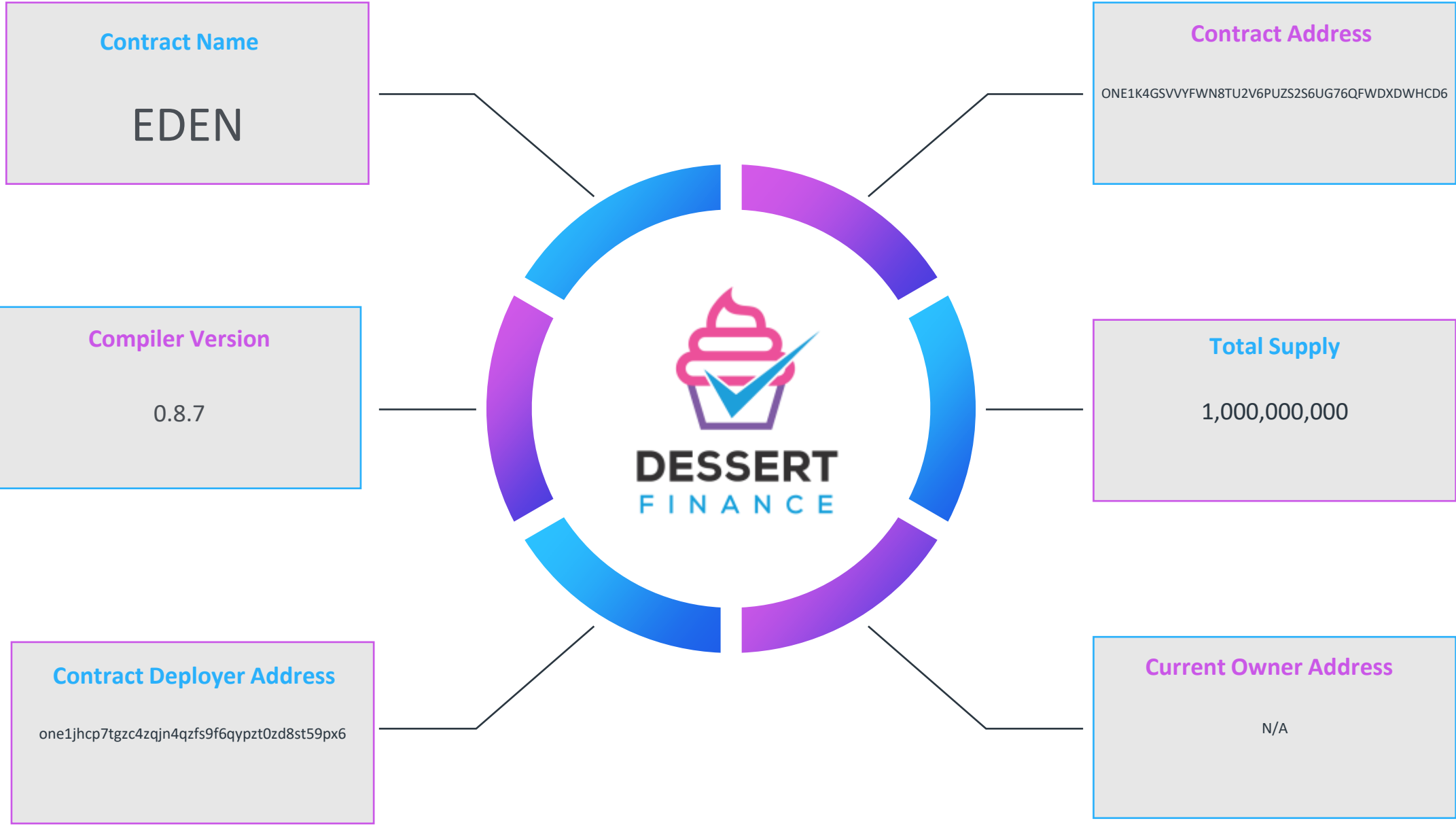
Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice.

Table of Contents



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

Contract Code Audit – Token Overview



HRC-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on EDEN

```
contract EDEN {
    string public constant name = "Eden";
    string public constant symbol = "EDEN";
    uint256 public constant decimals = 18;
    uint256 public immutable totalSupply;
    address immutable sushiRouter;

    mapping(address => uint256) balances;
    mapping(address => mapping(address => uint256)) allowed;

    event Transfer(address indexed from, address indexed to, uint256 value);
    event Approval(
        address indexed owner,
        address indexed spender,
        uint256 value
    );

    //for permit()
    bytes32 public immutable DOMAIN_SEPARATOR;
    // keccak256("Permit(address owner,address spender,uint256 value,uint256 nonce,uint256 deadline)");
    bytes32 public constant PERMIT_TYPEHASH =
        0x0e71edae22b1b0774d1f6037094f10105fa2f7a6e0126114a199644458126c9;
    mapping(address => uint256) public nonces;

    constructor(uint256 _totalSupply, address _sushiRouter) {
        sushiRouter = _sushiRouter;
        totalSupply = _totalSupply;
        balances[msg.sender] = _totalSupply;
        emit Transfer(address(0), msg.sender, _totalSupply);

        DOMAIN_SEPARATOR = keccak256(
            abi.encode(
                keccak256(
                    "EIP712Domain(string name,string version,uint256 chainId,address verifyingContract)"
                ),
                keccak256(bytes(name)),
                keccak256(bytes("1")),
                block.chainid,
                address(this)
            )
        );
    }

    /**
     * @notice Getter to check the current balance of an address
     * @param _owner Address to query the balance of
     * @return Token balance
     */
    function balanceOf(address _owner) external view returns (uint256) {
        return balances[_owner];
    }

    /**
     * @notice Getter to check the amount of tokens that an owner allowed to a spender
     * @param _owner The address which owns the funds
     * @param _spender The address which will spend the funds
     * @return The amount of tokens still available for the spender
     */
    function allowance(address _owner, address _spender)
```

Contract Address

one1k4gsvvyfwn8tu2v6puzs2s6ug76qfwdxwhcd6

Symbol

EDEN

Contract Creator

one1jhpc7tgzc4zqjn4qzfs9f6qypzt0zd8st59px6

Source Code

Contract Source Code Verified

Contract Name

EDEN

Compiler Version

0.8.7

Optimization Enabled

No

Code is truncated to fit the constraints of this document.

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

HRC-20 Contract Code Audit – Vulnerabilities Checked - EDEN

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 (x3)	Complete	Complete	✓ Low Risk
Deployer Can Access User Funds	Complete	Complete	✓ Low / No Risk

The contract code is **verified** on Harmony Block Explorer.

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

Contract Code Audit – Mint Functions

This Contract Cannot Mint New EDEN 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.

HRC-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on EdenRewards

```
// SPDX-License-Identifier: MIT
// File: @openzeppelin/contracts/ownership/Ownable.sol
pragma solidity ^0.8.0;

/**
 * @dev Contract module which provides a basic access control mechanism, where
 * there is an account (an owner) that can be granted exclusive access to
 * specific functions.
 *
 * This module is used through inheritance. It will make available the modifier
 * `onlyOwner`, which can be applied to your functions to restrict their use to
 * the owner.
 */
contract Ownable {
    address public owner;

    event OwnershipTransferred(address indexed previousOwner, address indexed newOwner);

    constructor() {
        owner = msg.sender;
    }

    /**
     * @dev Throws if called by any account other than the owner.
     */
    modifier onlyOwner() {
        require(msg.sender == owner, "Ownable: caller is not the owner");
        _;
    }

    /**
     * @dev Leaves the contract without owner. It will not be possible to call
     * `onlyOwner` functions anymore. Can only be called by the current owner.
     *
     * NOTE: Renouncing ownership will leave the contract without an owner,
     * thereby removing any functionality that is only available to the owner.
     */
    function renounceOwnership() public onlyOwner {
        emit OwnershipTransferred(owner, address(0));
        owner = address(0);
    }

    /**
     * @dev Transfers ownership of the contract to a new account (`newOwner`).
     * Can only be called by the current owner.
     */
    function transferOwnership(address newOwner) public onlyOwner {
        _transferOwnership(newOwner);
    }

    /**
     * @dev Transfers ownership of the contract to a new account (`newOwner`).
     */
    function _transferOwnership(address newOwner) internal {
        require(newOwner != address(0), "Ownable: new owner is the zero address");
        emit OwnershipTransferred(owner, newOwner);
        owner = newOwner;
    }
}
```

Contract Address

one1lxxtrn7f48m90gz8qqmzpl59fqd72l3gvc7df0

Contract Creator

one1jhcp7tgzc4zqjn4qzfs9f6qypzt0zd8st59px6

Source Code

Contract Source Code Verified

Contract Name

EdenRewards

Compiler Version

0.8.7

Optimization Enabled

No

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 Harmony Block Explorer.

HRC-20 Contract Code Audit – Vulnerabilities Checked - EdenRewards

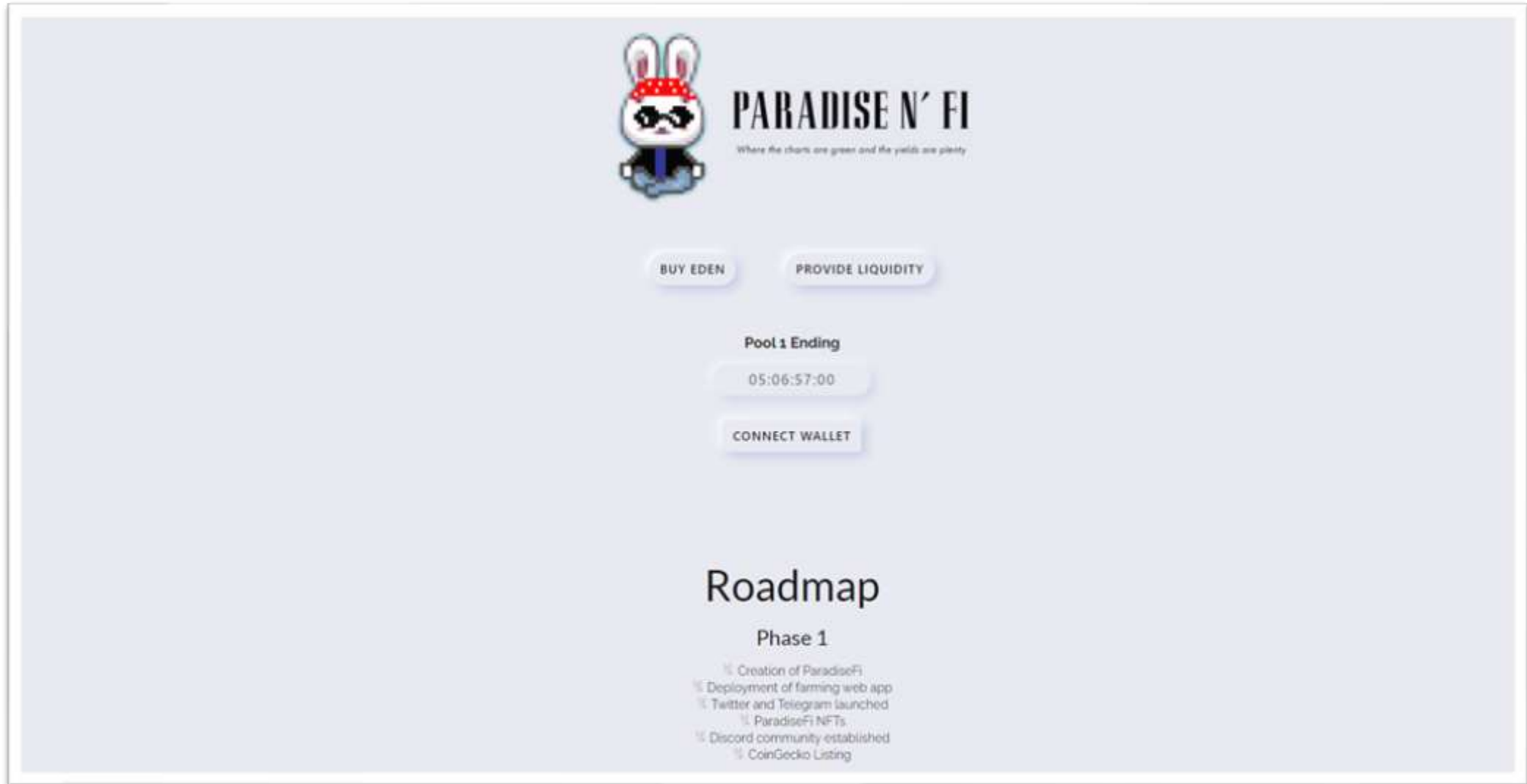
Vulnerability Tested	AI Scan	Human Review	Result
Integer Overflow	Complete	Complete	✓ Low / No Risk
Integer Underflow	Complete	Complete	✓ Low / No Risk
Correct Token Standards Implementation	Complete	Complete	✓ Low / No Risk
Timestamp Dependency for Crucial Functions	Complete	Complete	✓ Low / No Risk
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	Complete	Complete	✓ Low / No Risk
Deployer Can Access User Funds	Complete	Complete	✓ Low / No Risk

The contract code is **verified** on Harmony Block Explorer.

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

Website Part 1 – Overview

www.paradisefi.io



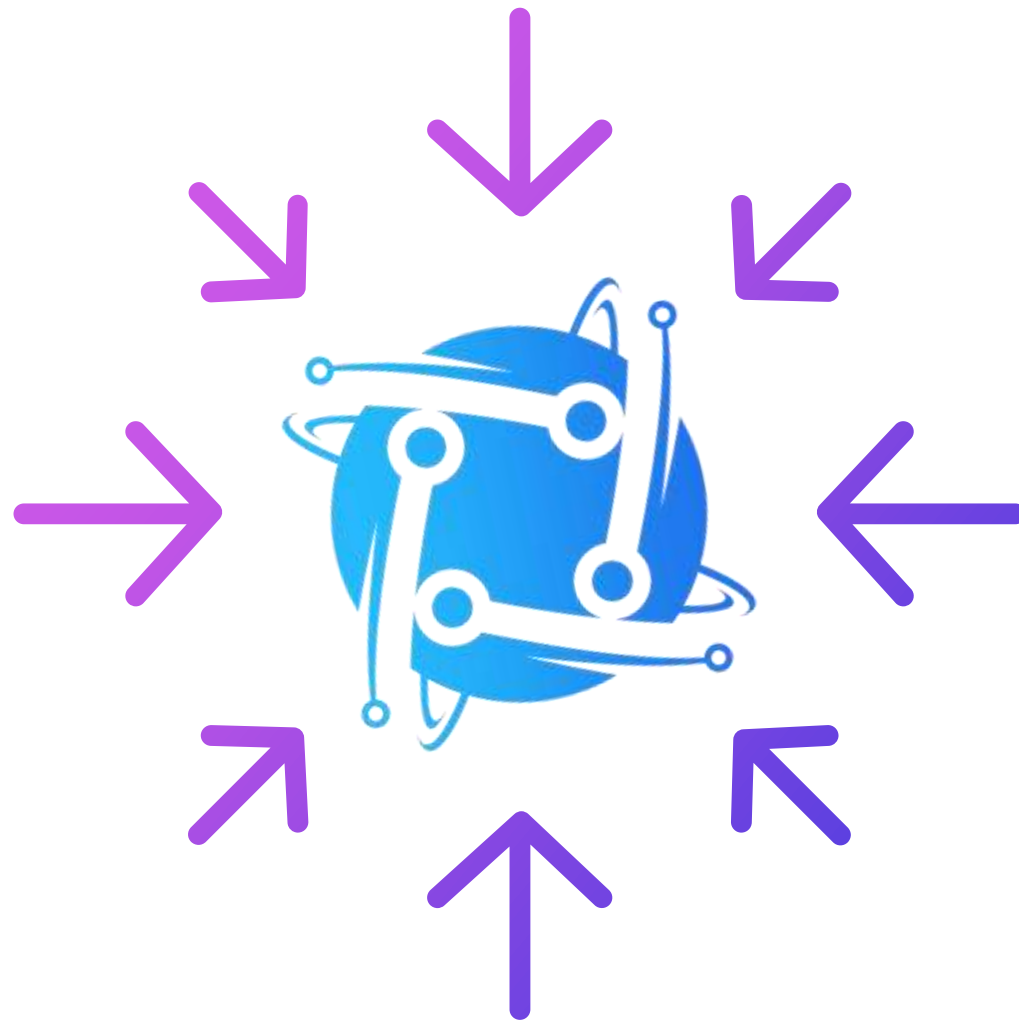
Above images are actual snapshots of the current live website of the project.

Website was registered on 10/24/2021, registration expires 10/24/2022.

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

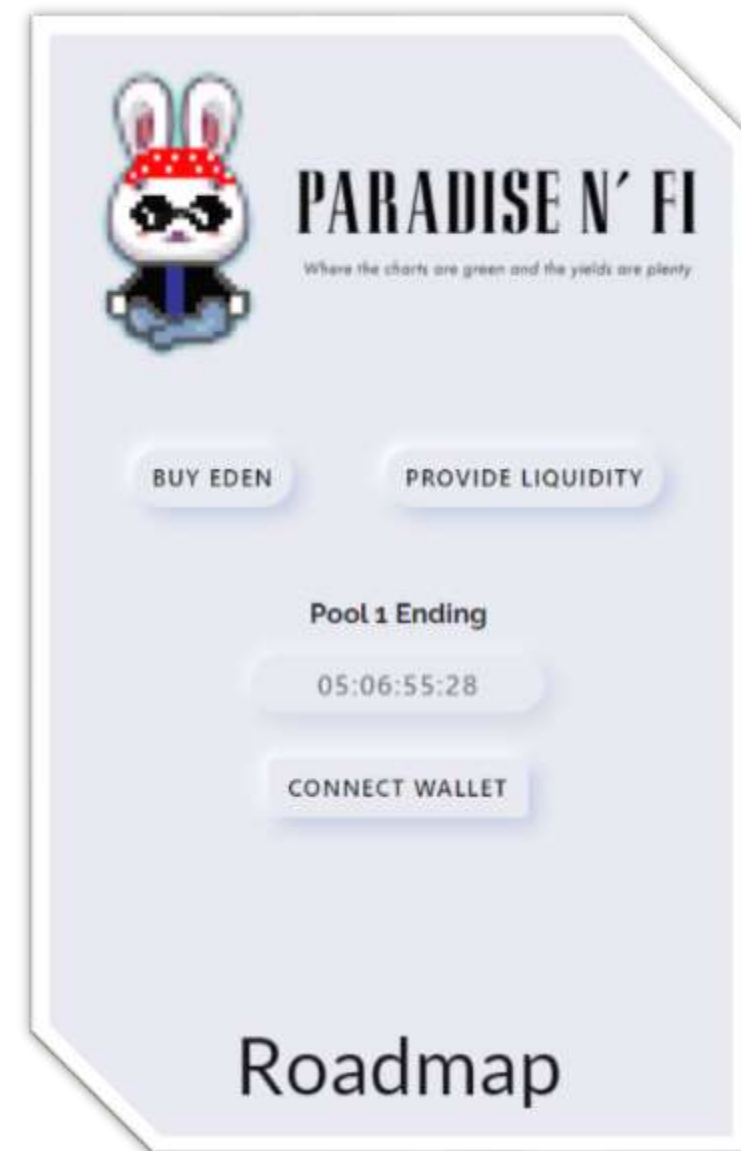
The website contained no JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

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: paradisefi.io

Issued by: Cloudflare Inc

Valid Until: 10/23/2022



CONTACT EMAIL

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

[Contact](mailto:contact@paradisefi.io)

contact@paradisefi.io



SPAM / MALWARE / POPUPS

No malware found

No injected spam found

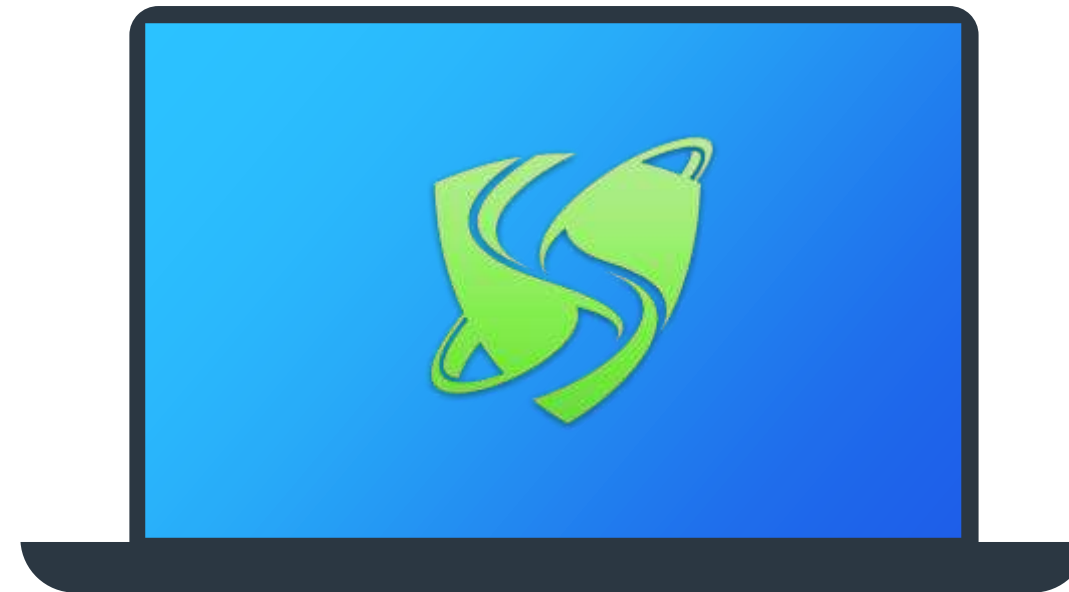
No internal server errors

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](#)



[Medium](#)

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

Top Token Holders

The top token holders at the time of the audit are shown below.

[Click here to view the most up-to-date list of holders](#)

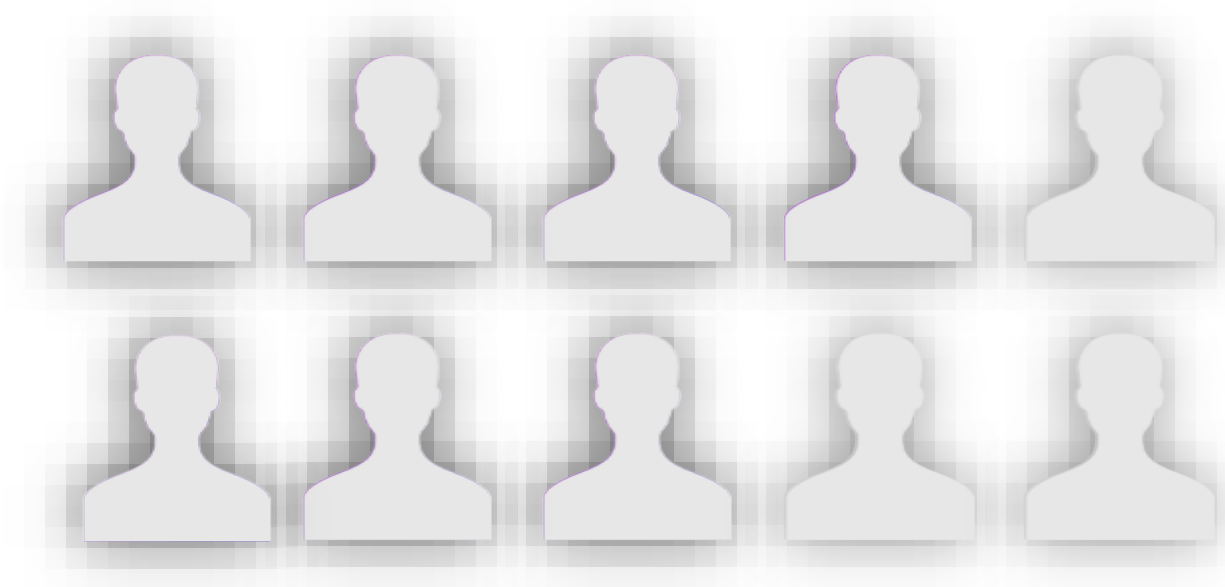
10 holders shown		< 1 / 40 >
Address		Balance
one1jhcp7tgzc4zqjn4qzfs9f6qypzt0zd8st59px6		582519156.826053246222107894 EDEN
one1sqmya5jw3953ywjayfchrkl73g8lvma82afn4		219139123.399193093552898358 EDEN
one1lxxtrn7f48m90gz8qqmzpl59fqd72l3gvc7df0		123193903.044069002367975388 EDEN
Jewel LP Token		43625006.181094874387372579 EDEN
one1hjh0cn8azkvfuld4kevllej4ssvmjzrgmmt0g		26075043.832288459657811952 EDEN
one12dtucseag25lcsavuf7hz4wj58rjz2e4l6a5p5		1037098.19597752465231412 EDEN
one16p9lk5tgm7yxw6fa2vcasa7g7l8h24sp6w8ycp		486602.183175938627381195 EDEN
one17xekq8zh5qt8ugttzqlleg77lhavf759k4u8uel		317570.683717540314737144 EDEN
one1hytush5c5lfup42alklcyzavsqrce97e5cdxkw		200161.690051592930522583 EDEN
one1aljt7udaccspa0v9yuhj73dd9pfwv737h0ycan		165220.4938243171340948 EDEN

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



We are unable to find any information about the team on the website at this time. Projects may choose to stay anonymous for a myriad of reasons.

Roadmap

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



Roadmap

Phase 1

- ✔ Creation of ParadiseFi
- ✔ Deployment of farming web app
- ✔ Twitter and Telegram launched
 - ✔ ParadiseFi NFTs
- ✔ Discord community established
 - ✔ CoinGecko Listing

Phase 2

- 5,000 Twitter Followers
- 1,000 Telegram Members
- Professional Audit of the Smart Contracts
 - ✔ Listing on CoinMarketCap
- Partnerships - Collaborations

Phase 3

- ParadiseFi Swap
- ParadiseFi NFT Platform
- Token Burns + Airdrops
- Centralized Exchange Listings
- More to be announced!

[READ OUR WHITEPAPER](#)

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 EDEN (EDEN) THIS AUDIT IS ONLY VALID IF VIEWED ON
DESSERTSWAP.FINANCE. THIS AUDIT AT BLOCK NUMBER: **19093181**

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