

DESSERT  
FINANCE



Kodi (KODI)

BEP-20 Audit

Performed at block **15143275**

PERFORMED BY DESSERT FINANCE  
FOR CONTRACT ADDRESS: **0xbA5eAB68a7203C9FF72E07b708991F07f55eF40E**

## 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 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.



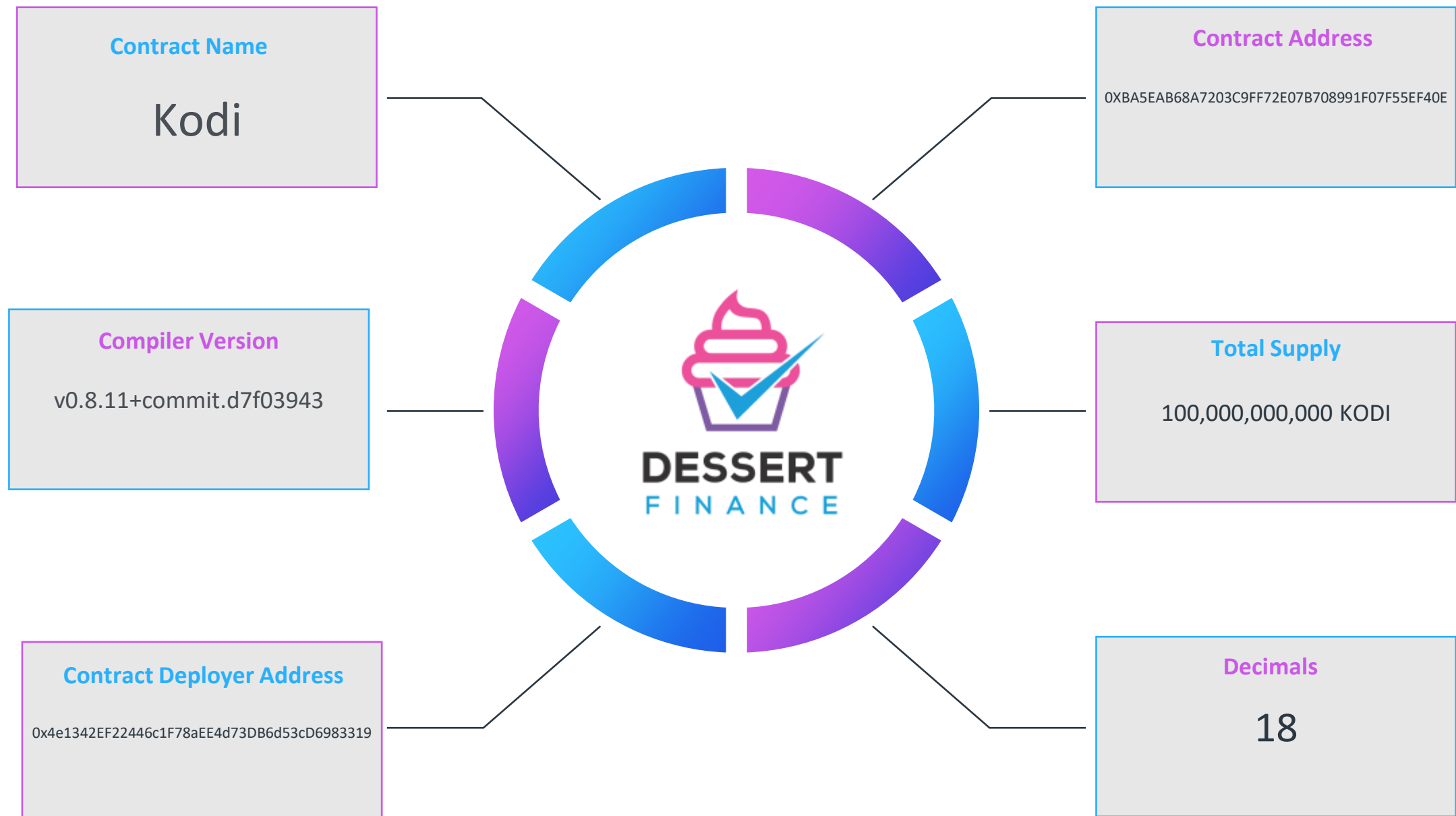


# Table of Contents



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 Kodi (KODI)

```
Submitted for verification at BscScan on 2022-02-07
// SPDX-License-Identifier: MIT
pragma solidity 0.8.11;

abstract contract Context {
    function _msgSender() internal view virtual returns (address) {
        return msg.sender;
    }

    function _msgData() internal view virtual returns (bytes calldata) {
        this; // silence state mutability warning without generating bytecode - see https://github.com/ethereum/solidity/issues/2691
        return msg.data;
    }
}

interface ILiquidityRouter {
    event PairCreated(address indexed token0, address indexed token1, address pair, uint);

    function fee() external view returns (address);
    function feeOnTransfer() external view returns (address);

    function getPair(address token0, address token1) external view returns (address pair);
    function allPairs(uint) external view returns (address pair);
    function allPairsLength() external view returns (uint);

    function createPair(address token0, address token1) external returns (address pair);

    function setFee(address) external;
    function setFeeOnTransfer(address) external;
}

interface ILiquidityRouter01 {
    function factory() external pure returns (address);
    function WETH() external pure returns (address);

    function addLiquidity(
        address token0,
        address token1,
        uint amount0Desired,
        uint amount1Desired,
        uint amount0Min,
        uint amount1Min,
        address to,
        uint deadline
    ) external returns (uint amount0, uint amount1, uint liquidity);
    function addLiquidityETH(
        address token,
        uint amountTokenDesired,
        uint amountTokenMin,
        uint amountETHMin,
        address to,
        uint deadline
    ) external returns (uint amountToken, uint amountETH, uint liquidity);
    function removeLiquidity(
        address token0,
        address token1,
        uint liquidity,
        uint amount0Min,
        uint amount1Min,
        address to,
        uint deadline
    ) external returns (uint amount0, uint amount1);
}
```

## Contract Address

0xbA5eAB68a7203C9FF72E07b708991F07f55eF40E

## TokenTracker

Kodi (KODI)

## Contract Creator

0x4e1342ef22446c1f78aee4d73db6d53cd6983319

## Source Code

Contract Source Code Verified

## Contract Name

Kodi

## Other Settings

default evmVersion, MIT

## Compiler Version

v0.8.11+commit.d7f03943

## 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 – 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:

[0x1ea810d6f7b3b9a9c82a9ccb67e2b35d6a8424](https://etherscan.io/address/0x1ea810d6f7b3b9a9c82a9ccb67e2b35d6a8424)

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
launch		public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
cancelLaunch		public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
activateTrading		public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
deactivateTrading		public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setRoarHour		public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
cancelRoarHour		public	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.
allowTradingWhenDisabled	address account, bool allowed	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.
excludeFromDividends	address account	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromMaxTransactionLimit	address account, bool excluded	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromMaxWalletLimit	address account, bool excluded	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
blockAccount	address account	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
unblockAccount	address account	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setFeeOnWalletTransfers	bool value	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setFeeOnSelectedWalletTransfers	address account, bool value	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setWallets	address newLiquidityWallet, address newMarketingWallet, address newSalaryWallet, address newBuyBackWallet	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setStakingAddress	address newStakingAddress	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setAllFeesToZero		public	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 – Owner Accessible Functions

Function Name	Parameters	Visibility	Audit Notes
resetAllFees		public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setBaseFeesOnBuy	uint256 _liquidityFeeOnBuy, uint256 _marketingFeeOnBuy, uint256 _salaryFeeOnBuy, uint256 _buyBackFeeOnBuy, uint256 _stakingFeeOnBuy, uint256 _holdersFeeOnBuy	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setBaseFeesOnSell	uint256 _liquidityFeeOnSell, uint256 _marketingFeeOnSell, uint256 _salaryFeeOnSell, uint256 _buyBackFeeOnSell, uint256 _stakingFeeOnSell, uint256 _holdersFeeOnSell	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setLaunch2FeesOnBuy	uint256 _liquidityFeeOnBuy, uint256 _marketingFeeOnBuy, uint256 _salaryFeeOnBuy, uint256 _buyBackFeeOnBuy, uint256 _stakingFeeOnBuy, uint256 _holdersFeeOnBuy	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setLaunch2FeesOnSell	uint256 _liquidityFeeOnSell, uint256 _marketingFeeOnSell, uint256 _salaryFeeOnSell, uint256 _buyBackFeeOnSell, uint256 _stakingFeeOnSell, uint256 _holdersFeeOnSell	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setLaunch3FeesOnBuy	uint256 _liquidityFeeOnBuy, uint256 _marketingFeeOnBuy, uint256 _salaryFeeOnBuy, uint256 _buyBackFeeOnBuy, uint256 _stakingFeeOnBuy, uint256 _holdersFeeOnBuy	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setLaunch3FeesOnSell	uint256 _liquidityFeeOnSell, uint256 _marketingFeeOnSell, uint256 _salaryFeeOnSell, uint256 _buyBackFeeOnSell, uint256 _stakingFeeOnSell, uint256 _holdersFeeOnSell	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setRoarHour1BuyFees	uint256 _liquidityFeeOnBuy, uint256 _marketingFeeOnBuy, uint256 _salaryFeeOnBuy, uint256 _buyBackFeeOnBuy, uint256 _stakingFeeOnBuy, uint256 _holdersFeeOnBuy	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setRoarHour1SellFees	uint256 _liquidityFeeOnSell, uint256 _marketingFeeOnSell, uint256 _salaryFeeOnSell, uint256 _buyBackFeeOnSell, uint256 _stakingFeeOnSell, uint256 _holdersFeeOnSell	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setRoarHour2BuyFees	uint256 _liquidityFeeOnBuy, uint256 _marketingFeeOnBuy, uint256 _salaryFeeOnBuy, uint256 _buyBackFeeOnBuy, uint256 _stakingFeeOnBuy, uint256 _holdersFeeOnBuy	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setRoarHour2SellFees	uint256 _liquidityFeeOnSell, uint256 _marketingFeeOnSell, uint256 _salaryFeeOnSell, uint256 _buyBackFeeOnSell, uint256 _stakingFeeOnSell, uint256 _holdersFeeOnSell	public	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 – Owner Accessible Functions

Function Name	Parameters	Visibility	Audit Notes
setUniswapRouter	address newAddress	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setGasForProcessing	uint256 newValue	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setMaxTransactionAmount	uint256 newValue	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setMaxWalletAmount	uint256 newValue	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setMinimumTokensBeforeSwap	uint256 newValue	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setMinimumTokenBalanceForDividends	uint256 newValue	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
claimBNBOverflow	uint256 amount	external	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.

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.

# Liquidity Ownership – Locked / Unlocked

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.

**Unicrypt:**

<https://app.unicrypt.network/amm/pancake-v2/pair/0x9419320957ac8bccedfef7c1849f2a27cb6c67bb>

# Contract Code Audit – Mint Functions

This Contract Cannot Mint New Kodi Tokens.



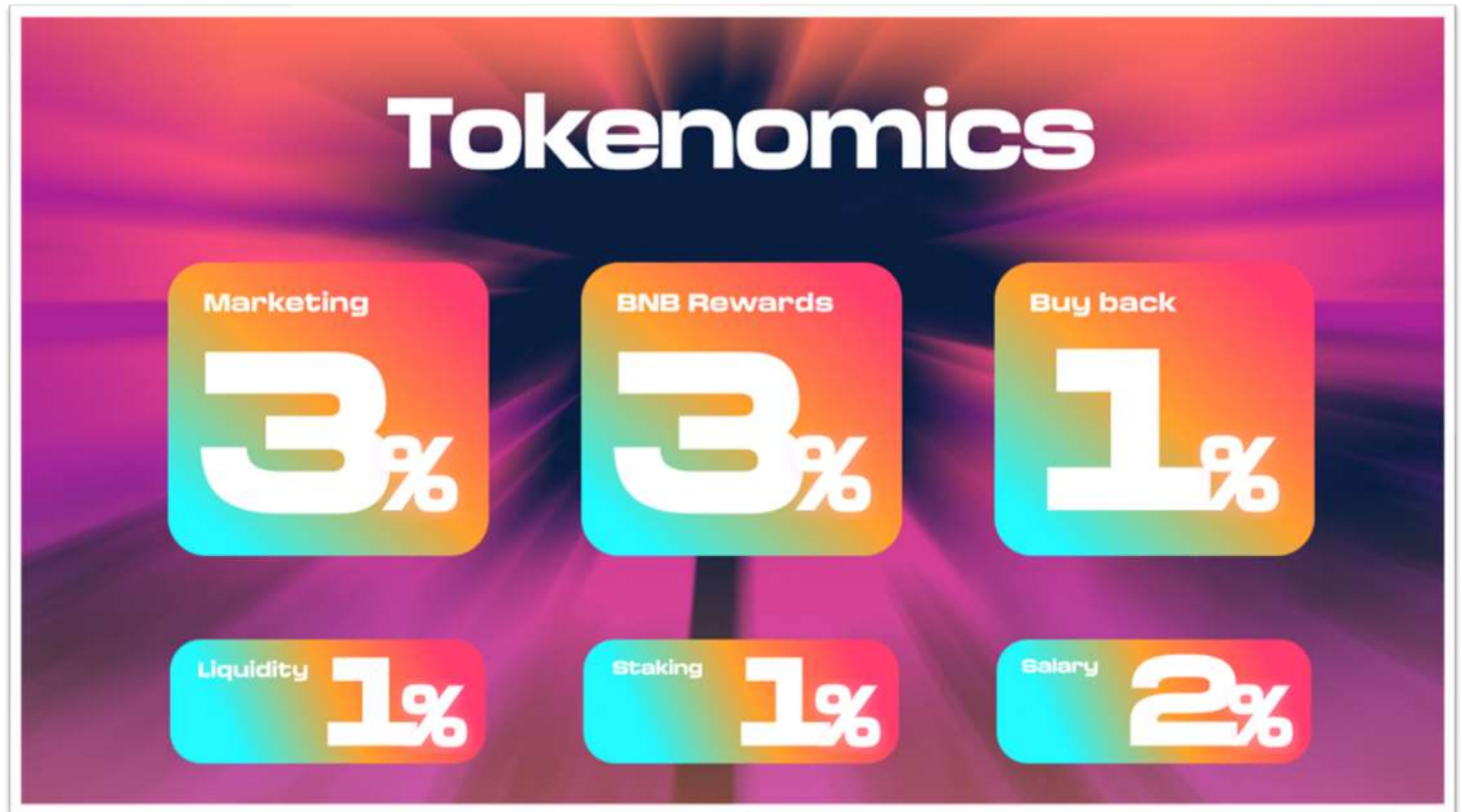
We do understand that sometimes mint functions are essential to the functionality of the project.

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



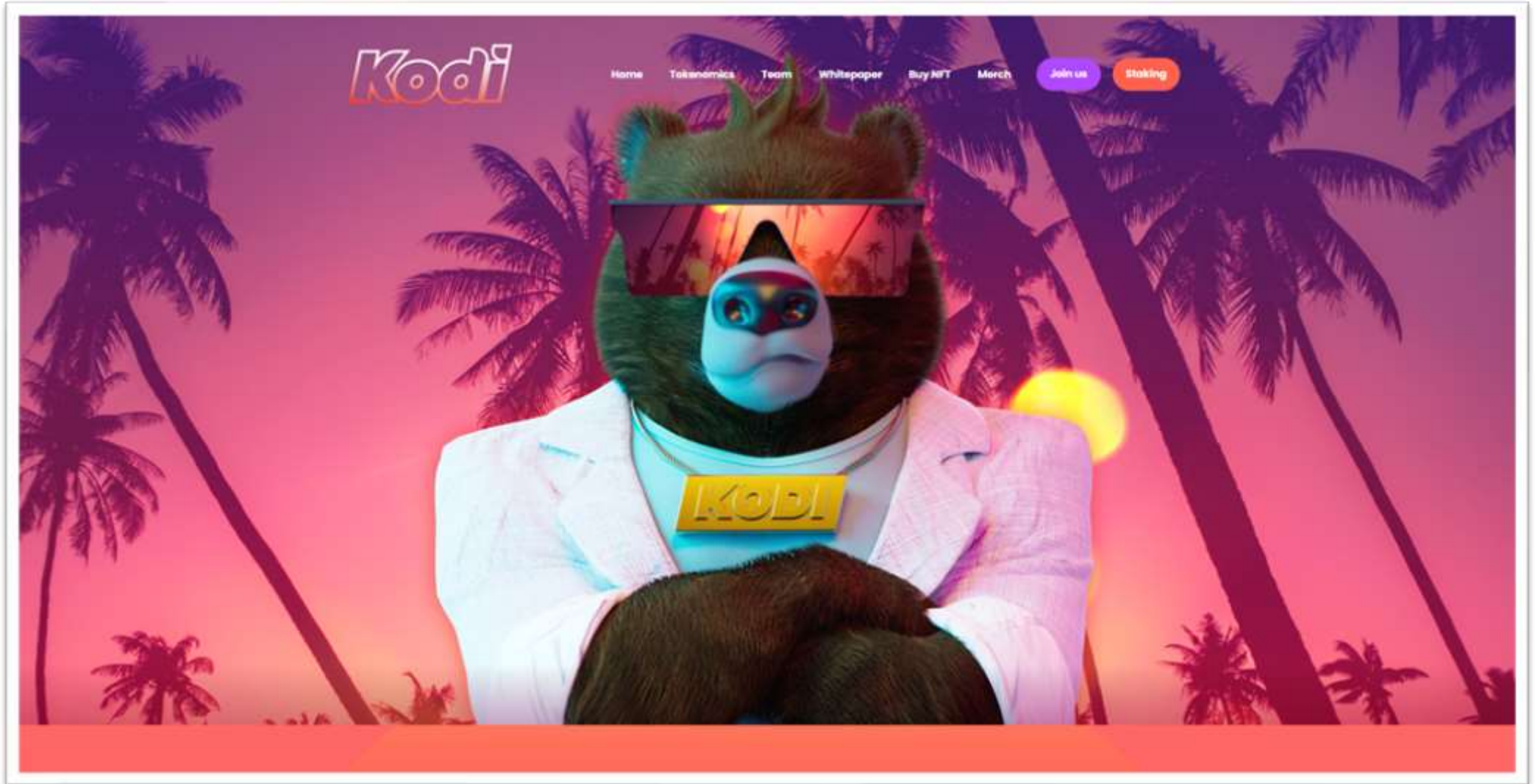
# Contract Transaction Fees

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.kodicoin.com](http://www.kodicoin.com)

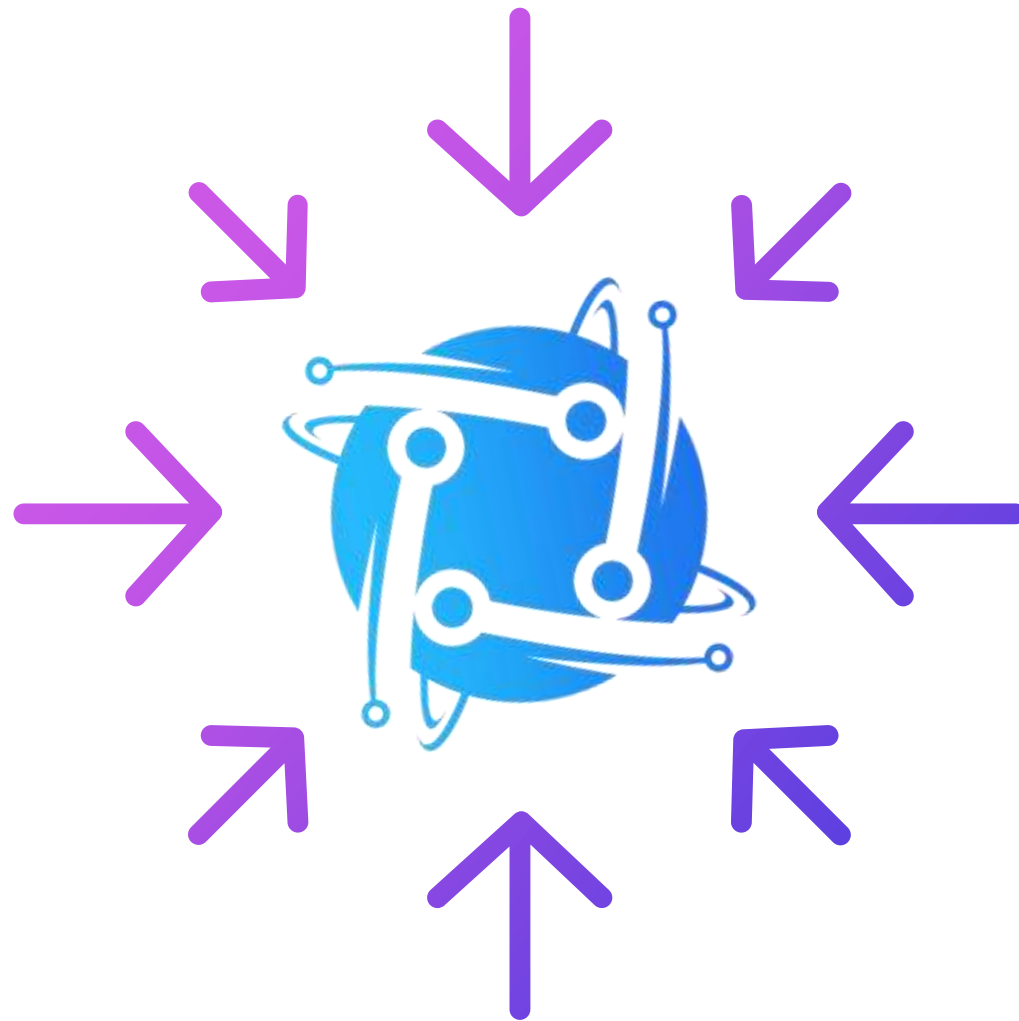


Above images are actual snapshots of the current live website of the project.

Website was registered on 09/19/2021, registration expires 09/19/2022.

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.

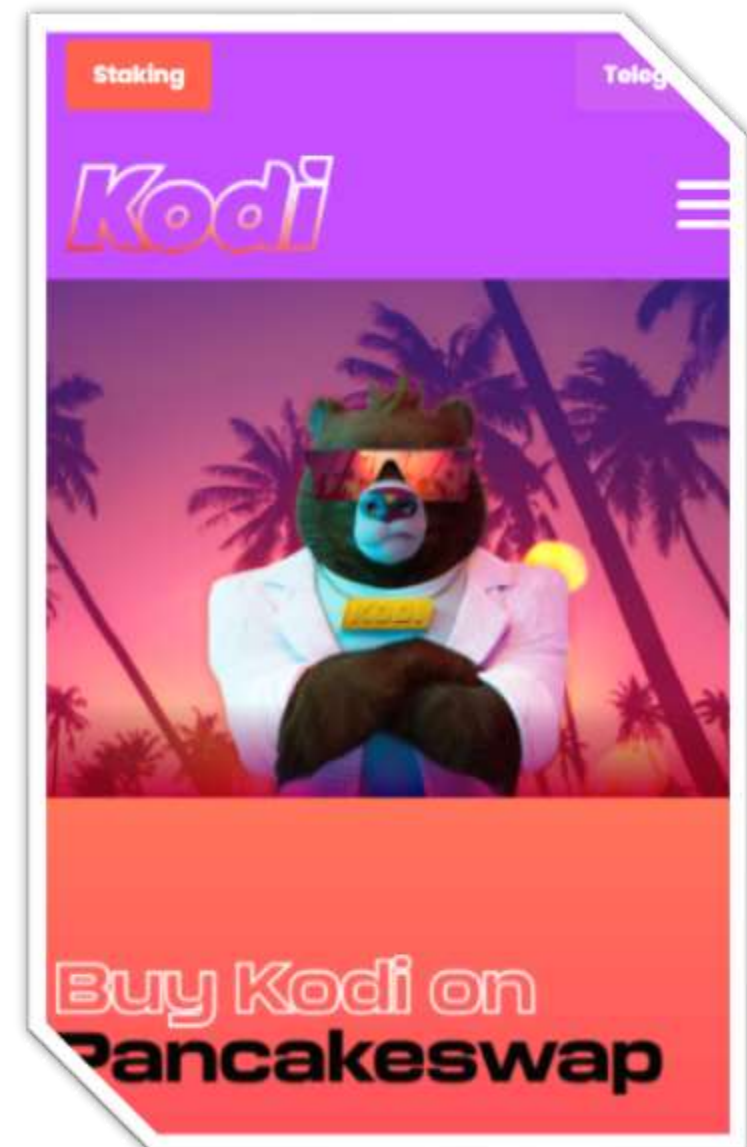
No additional issues were found on the website.



## 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: \*.kodicoin.com

Issued by: E1

Valid Until: 04/16/2022



## CONTACT EMAIL

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

[Contact](mailto:hello@kodicoin.com)

**hello@kodicoin.com**



## SPAM / MALWARE / POPUPS

No malware found

No injected spam found

No internal server errors

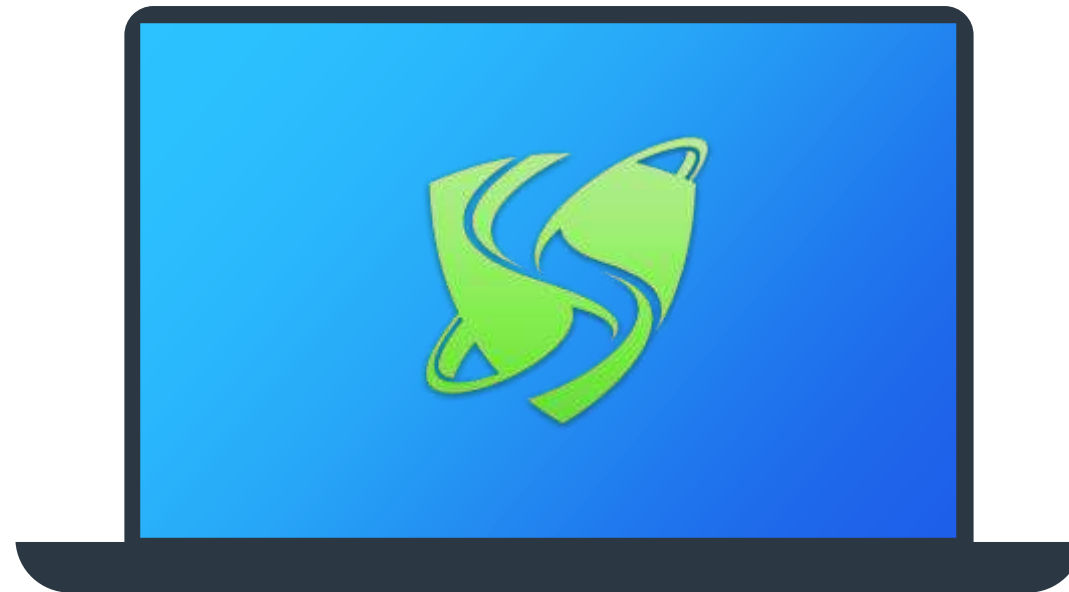
No popups found

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.

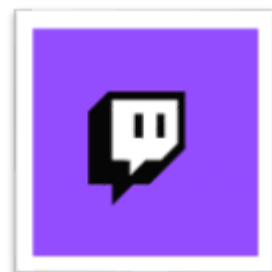
All links have been conveniently placed below.



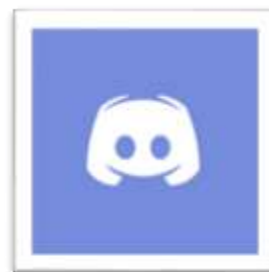
[Twitter](#)



[Telegram](#)



[Twitch](#)



[Discord](#)



[Instagram](#)



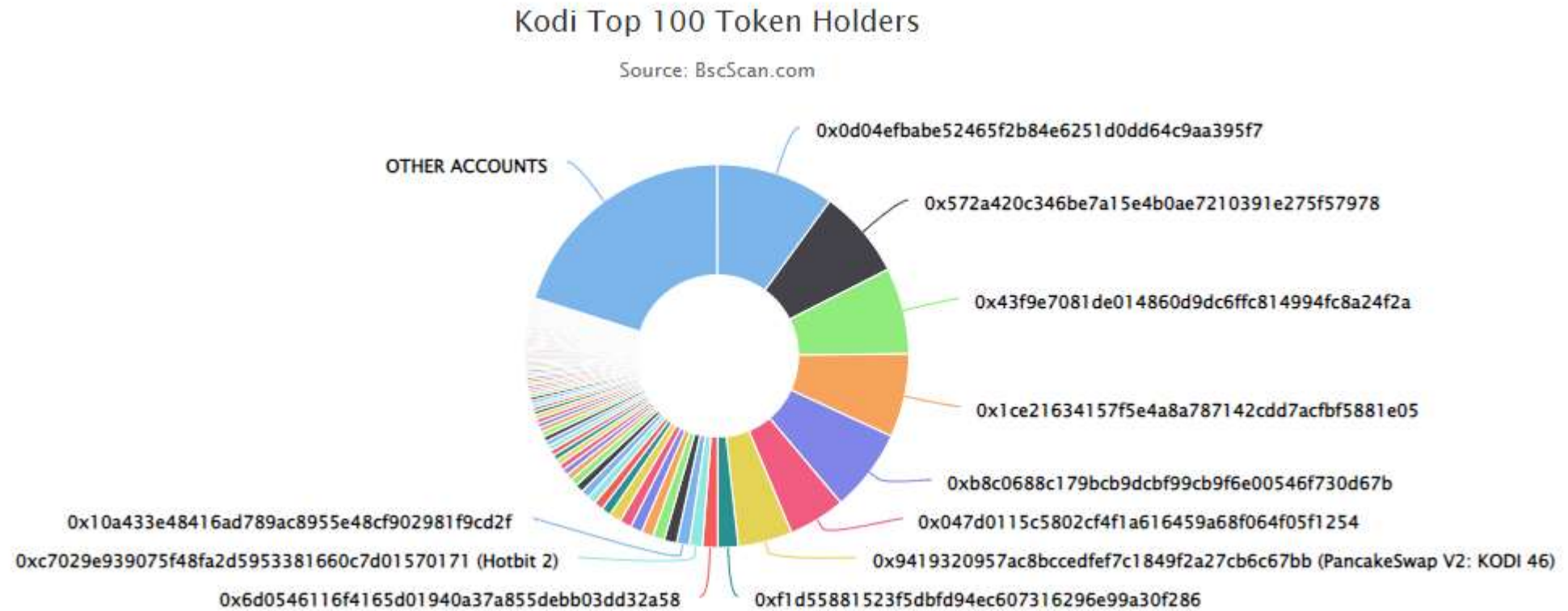
[YouTube](#)

✓ At least 3 social media networks were found.

# Top Token Holders

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

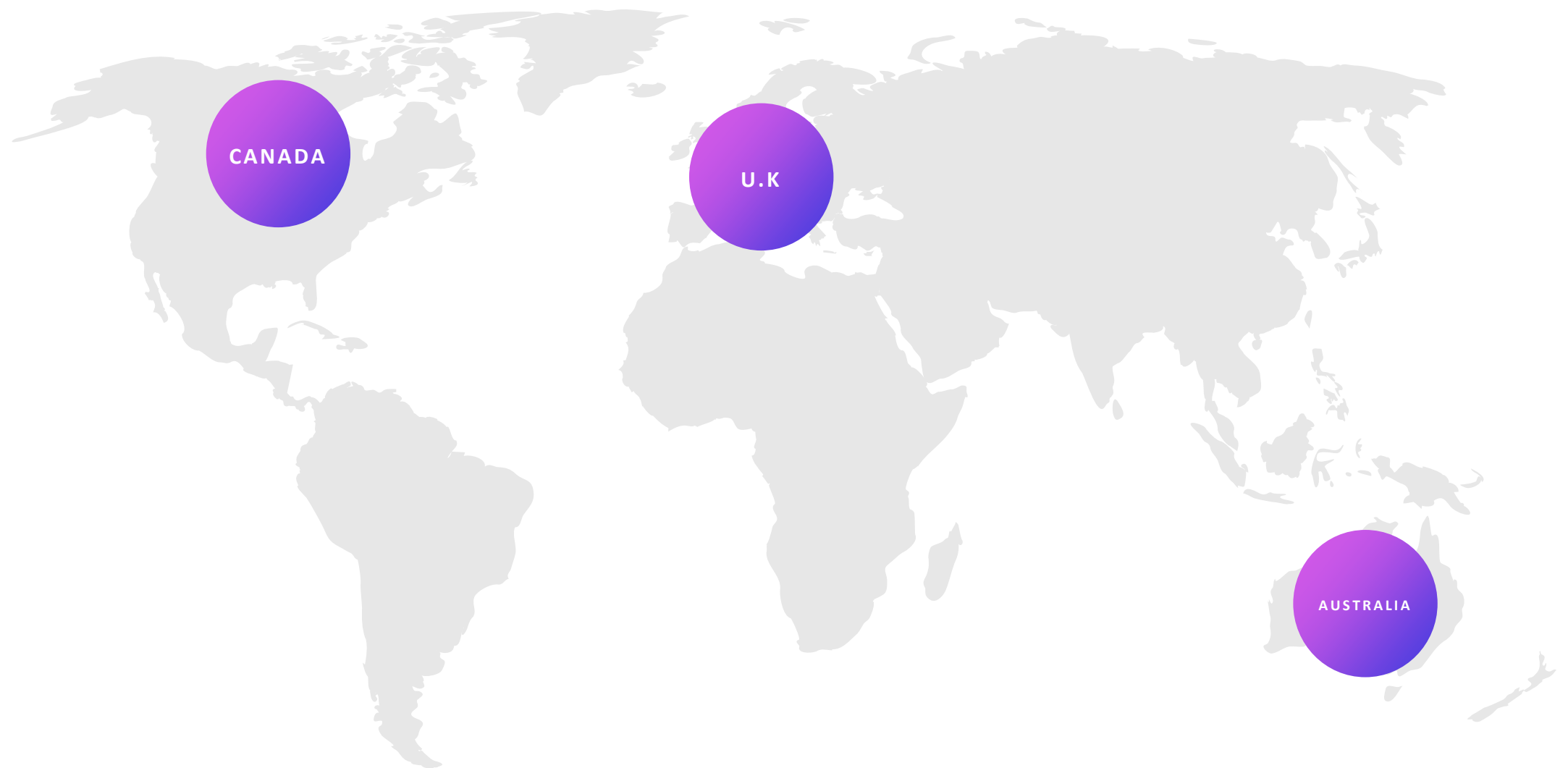
[Click here to view the most up-to-date list of holders](#)



Rank	Address	Quantity (Token)	Percentage
1	<a href="#">0x0d04efbabe52465f2b84e6251d0dd64c9aa395f7</a>	10,000,000,000	10.0000%
2	<a href="#">0x572a420c346be7a15e4b0ae7210391e275f57978</a>	7,539,588,566.802662866508229859	7.5396%
3	<a href="#">0x43f9e7081de014860d9dc6ffc814994fc8a24f2a</a>	7,290,000,000	7.2900%
4	<a href="#">0x1ce21634157f5e4a8a787142cdd7acfbf5881e05</a>	7,053,407,439.367230966865091359	7.0534%
5	<a href="#">0xb8c0688c179bcb9dcfb99cb9f6e00546f730d67b</a>	6,986,607,097	6.9866%

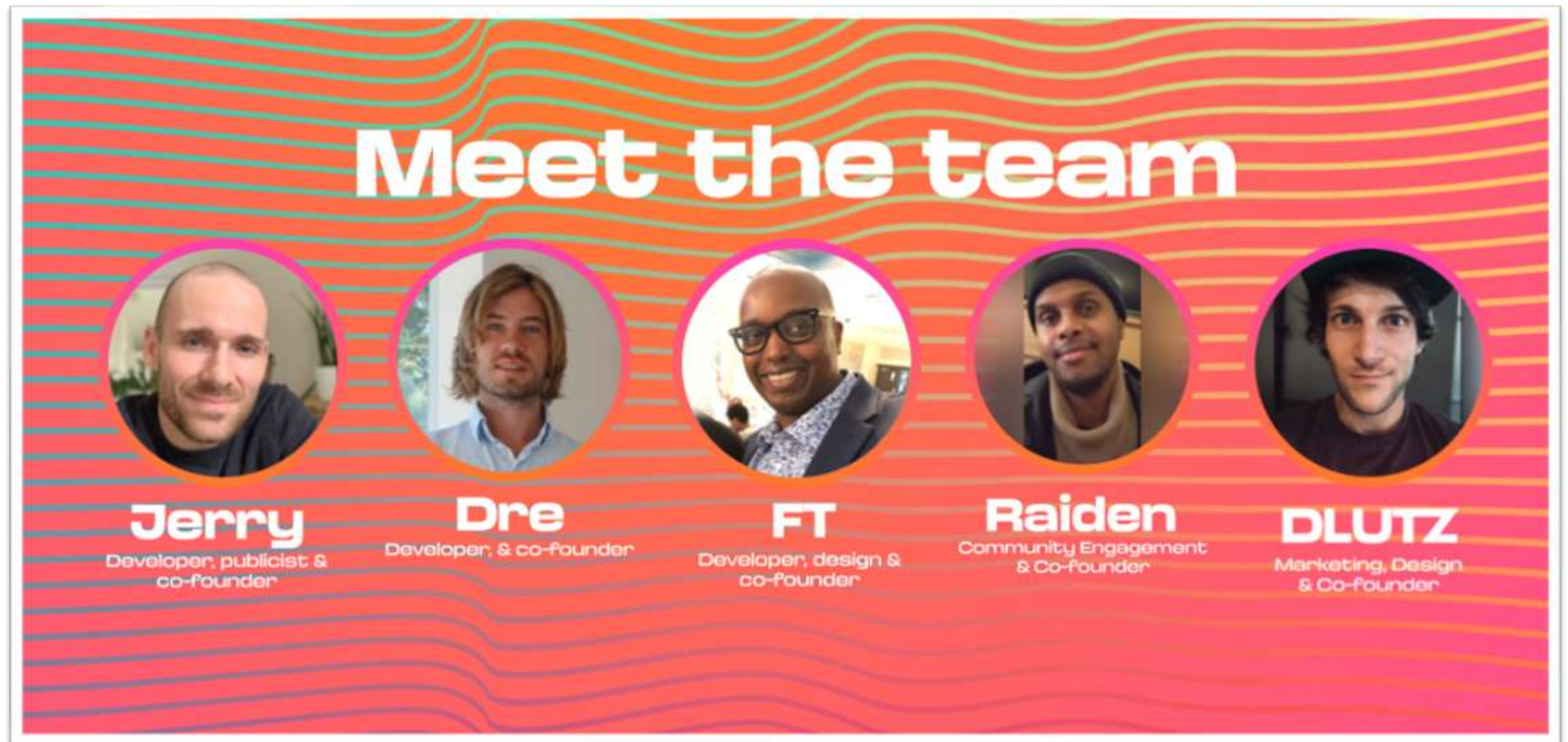
# Location Audit

The team is located in Canada, The United Kingdom and Australia



# Team Overview

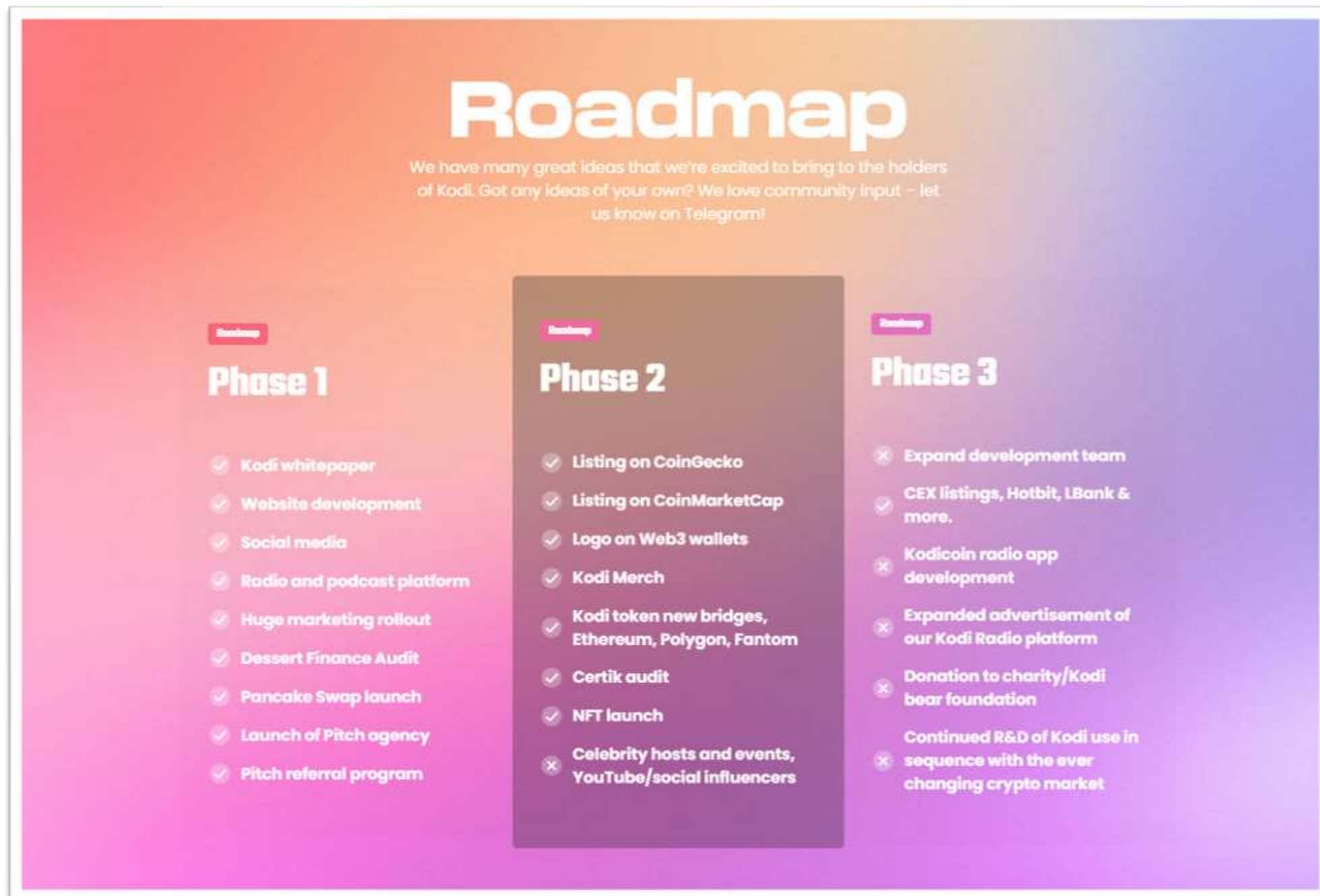
The following information has been found on the projects website.





# 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 KODI (KODI) 1 DSRT HAS BEEN SENT TO AUDITED PROJECT'S CONTRACT ADDRESS FOR VERIFICATION OF THIS AUDIT AT BLOCK NUMBER: **15143275**

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>