auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

ShopzzyAi

DATED : 07 Mar 23'

AUDIT SUMMARY

Project name – ShopzzyAi

Date: 07 March, 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: **Passed (Contract is developed by safu dev)**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	1	0	0	0
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

1- Manual Review:
a line by line code review has been performed by audit ace team.

2- BSC Test Network:
all tests were done on BSC Test network, each test has its transaction has attached to it.

3- Slither : Static Analysis

Testnet Link: all tests were done using this contract, tests are done on BSC Testnet

https://testnet.bscscan.com/token/0xbfE56ee534bE65CBaA710d2b2da04ef62f54C2fB

Token Information

Token Name : ShopzzyAi

Token Symbol: SHOPZZY

Decimals: 18

Token Supply: 10,000,000

Token Address:
0xFD21B0CAd30A00C757c1cf08815E0cdF27D3080A

Checksum:
37385a016ae0d6a767b9f49f31715950f331ae25

Owner:
0x2535a5F5991A5EEAf78bdd48FA9517F39f44f0ad

TOKEN OVERVIEW

Fees:

Buy Fees: 2%

Sell Fees: 3%

Transfer Fees: 0%

Fees Privilige: None

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: including and excluding from fees

AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code isexercised when we run the test cases.
- Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

VULNERABILITY CHECKLIST

- [x] Return values of low-level calls
- [x] Private modifier
- [x] Multiple Sends
- [x] Using Suicide
- [x] Gas Limitand Loops
- [x] Address hardcoded
- [x] Exception Disorder
- [x] Using inline assembly
- [x] Divide before multiply
- [x] Missing Zero Address Validation
- [x] Compiler version not fixed
- [x] **Gasless Send**
- [x] Using block.timestamp
- [x] Re-entrancy
- [x] Tautology or contradiction
- [x] Timestamp Dependence
- [x] Revert/require functions
- [x] Use of tx.origin
- [x] Integer overflow/underflow
- [x] Dangerous strict equalities
- [x] Using SHA3
- [x] Using throw

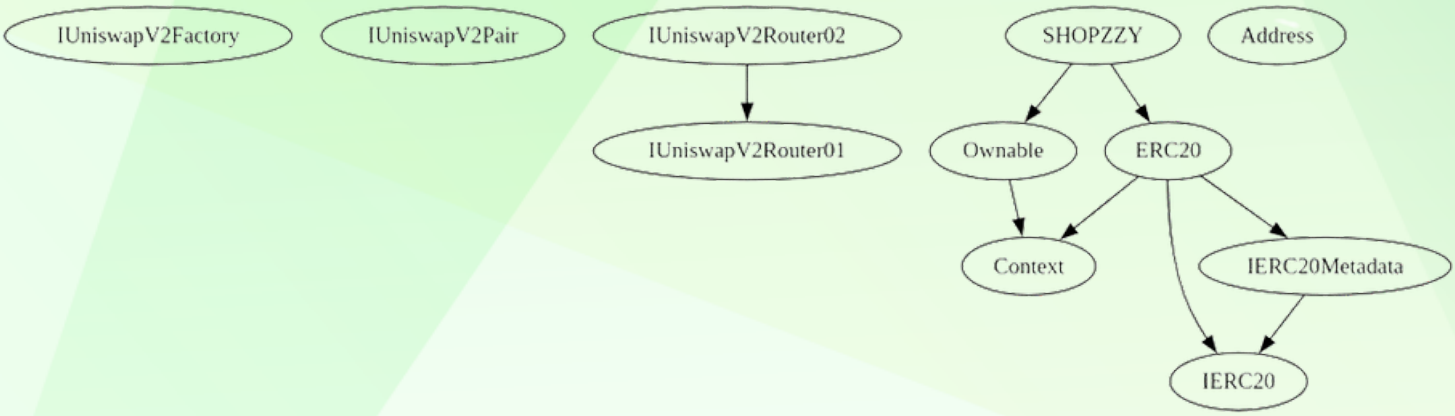
CLASSIFICATION OF RISK

Severity	Description
Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
Gas Optimization /Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity	Found
Critical	0
High-Risk	1
Medium-Risk	0
Low-Risk	0
Gas Optimization / Suggestions	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to change buy/sell/transfer taxes (2% buy, 3% sell, 0% transfer)
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to set max buy/sell/transfer amounts
- Owner is not able to disable trades
- Owner is not able to mint new tokens

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IUniswapV2Factory	Interface			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗
└	setFeeTo	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
IUniswapV2Pair	Interface			
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	External ❗		NO❗
└	permit	External ❗	🛑	NO❗
└	MINIMUM_LIQUIDITY	External ❗		NO❗
└	factory	External ❗		NO❗
└	token0	External ❗		NO❗
└	token1	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	mint	External ❗	🛑	NO❗
└	burn	External ❗	🛑	NO❗
└	swap	External ❗	🛑	NO❗
└	skim	External ❗	🛑	NO❗
└	sync	External ❗	🛑	NO❗
└	initialize	External ❗	🛑	NO❗

CONTRACT ASSESMENT

IUniswapV2Router01	Interface			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🛑	NO❗
└	addLiquidityETH	External ❗	💵	NO❗
└	removeLiquidity	External ❗	🛑	NO❗
└	removeLiquidityETH	External ❗	🛑	NO❗
└	removeLiquidityWithPermit	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermit	External ❗	🛑	NO❗
└	swapExactTokensForTokens	External ❗	🛑	NO❗
└	swapTokensForExactTokens	External ❗	🛑	NO❗
└	swapExactETHForTokens	External ❗	💵	NO❗
└	swapTokensForExactETH	External ❗	🛑	NO❗
└	swapExactTokensForETH	External ❗	🛑	NO❗
└	swapETHForExactTokens	External ❗	💵	NO❗
└	quote	External ❗		NO❗
└	getAmountOut	External ❗		NO❗
└	getAmountIn	External ❗		NO❗
└	getAmountsOut	External ❗		NO❗
└	getAmountsIn	External ❗		NO❗
IUniswapV2Router02	Interface	IUniswapV2Router01		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
IERC20	Interface			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
IERC20Metadata	Interface	IERC20		
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗

CONTRACT ASSESMENT

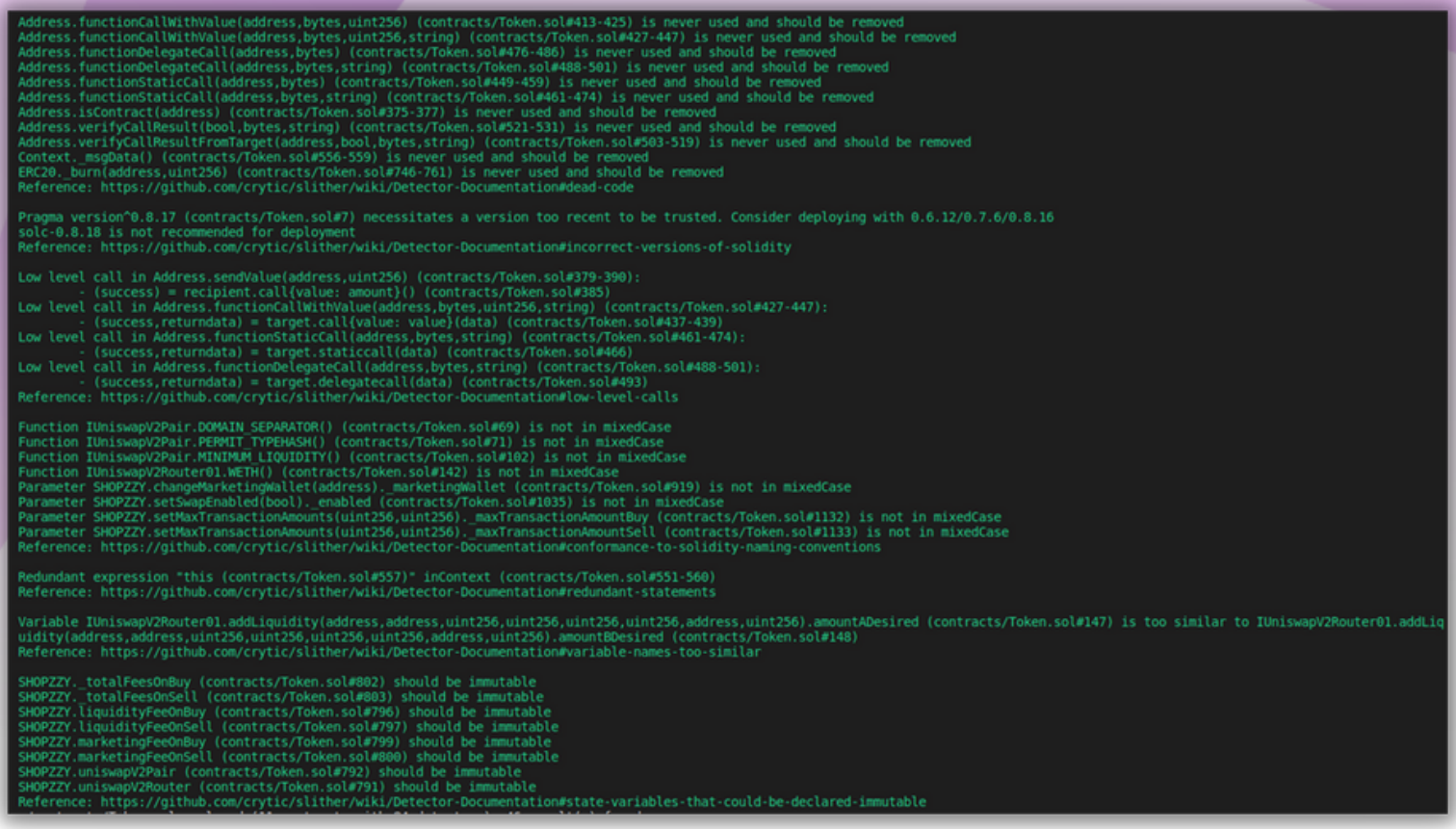
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🛑	
└	functionDelegateCall	Internal 🔒	🛑	
└	verifyCallResultFromTarget	Internal 🔒		
└	verifyCallResult	Internal 🔒		
└	_revert	Private 🔐		
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
ERC20	Implementation	Context, IERC20, IERC20Metadata		
└	<Constructor>	Public ❗	🛑	NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🛑	NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗
└	_transfer	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	

CONTRACT ASSESMENT

└	_approve	Internal 🔒	🛑		
└	_beforeTokenTransfer	Internal 🔒	🛑		
└	_afterTokenTransfer	Internal 🔒	🛑		
SHOPZZY	Implementation	ERC20, Ownable			
└	<Constructor>	Public ❗	🛑	ERC20
└	<Receive Ether>	External ❗	💵	NO❗
└	claimStuckTokens	External ❗	🛑	onlyOwner
└	excludeFromFees	External ❗	🛑	onlyOwner
└	isExcludedFromFees	Public ❗		NO❗
└	changeMarketingWallet	External ❗	🛑	onlyOwner
└	enableTrading	External ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑		
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	swapAndLiquify	Private 🔐	🛑		
└	swapAndSendMarketing	Private 🔐	🛑		
└	setEnableMaxTransactionLimit	External ❗	🛑	onlyOwner
└	setMaxTransactionAmounts	External ❗	🛑	onlyOwner
└	excludeFromMaxTransactionLimit	External ❗	🛑	onlyOwner
└	isExcludedFromMaxTransaction	Public ❗		NO❗

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS



```
Address.functionCallWithValue(address,bytes,uint256) (contracts/Token.sol#413-425) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#427-447) is never used and should be removed
Address.functionDelegateCall(address,bytes) (contracts/Token.sol#476-486) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#488-501) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/Token.sol#449-459) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#461-474) is never used and should be removed
Address.isContract(address) (contracts/Token.sol#375-377) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (contracts/Token.sol#521-531) is never used and should be removed
Address.verifyCallResultFromTarget(address,bool,bytes,string) (contracts/Token.sol#503-519) is never used and should be removed
Context._msgData() (contracts/Token.sol#556-559) is never used and should be removed
ERC20._burn(address,uint256) (contracts/Token.sol#746-761) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.17 (contracts/Token.sol#7) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.18 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#379-390):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#385)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#427-447):
        - (success,returndata) = target.call{value: value}(data) (contracts/Token.sol#437-439)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#461-474):
        - (success,returndata) = target.staticcall(data) (contracts/Token.sol#466)
Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#488-501):
        - (success,returndata) = target.delegatecall(data) (contracts/Token.sol#493)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#69) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#71) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#102) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#142) is not in mixedCase
Parameter SHOPZZY.changeMarketingWallet(address)._marketingWallet (contracts/Token.sol#919) is not in mixedCase
Parameter SHOPZZY.setSwapEnabled(bool)._enabled (contracts/Token.sol#1035) is not in mixedCase
Parameter SHOPZZY.setMaxTransactionAmounts(uint256,uint256)._maxTransactionAmountBuy (contracts/Token.sol#1132) is not in mixedCase
Parameter SHOPZZY.setMaxTransactionAmounts(uint256,uint256)._maxTransactionAmountSell (contracts/Token.sol#1133) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#557)" inContext (contracts/Token.sol#551-560)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#147) is too similar to IUniswapV2Router01.addLiq
uidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#148)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

SHOPZZY._totalFeesOnBuy (contracts/Token.sol#802) should be immutable
SHOPZZY._totalFeesOnSell (contracts/Token.sol#803) should be immutable
SHOPZZY.liquidityFeeOnBuy (contracts/Token.sol#796) should be immutable
SHOPZZY.liquidityFeeOnSell (contracts/Token.sol#797) should be immutable
SHOPZZY.marketingFeeOnBuy (contracts/Token.sol#799) should be immutable
SHOPZZY.marketingFeeOnSell (contracts/Token.sol#800) should be immutable
SHOPZZY.uniswapV2Pair (contracts/Token.sol#792) should be immutable
SHOPZZY.uniswapV2Router (contracts/Token.sol#791) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Result => A static analysis of contract's source code has been performed using slither,

No major issues were found in the output

FUNCTIONAL TESTING

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

All the functionalities have been tested, no issues were found

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0x4e8e8fe0d8442aca642644f779f4ffedf92582294544cab3deaa2c0d0de79d0b

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x29996f349888b6b9b361ad4b090eaeb44d45338d980c27db267869f59a67ff18

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xdc5356d97a4ae5341e1f2b08cd9a1cec7c7c0ab78119cdfb1473962340c8a679

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x144b968810f699d70f648804a575af3284db92fd770ea329fe91003a435dd311

5- Buying when not excluded (1% tax) (passed):
https://testnet.bscscan.com/tx/0xf2a3deea1c06c00b5199559fad1f778c0fef9976777ec30ca0dd9f3292954c53

6- Selling when not excluded (3% tax) (passed):
https://testnet.bscscan.com/tx/0x7ad9c977397d2a9d9c20de1fa78ee1a13a90b4ed601f31d069be4442098ca449

FUNCTIONAL TESTING

7- Transferring when not excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xff2e32763398cc07d98cdb58b0627a2c10c0dd7b3680d61757d06f85375b17a0

MANUAL TESTING

High Risk Issue

Issue: trades wont be open for investors until calling enableTrading function
Line: 670-674
Function: enableTrading
Category: centralization
Overview:
trades (including buys, sells and transfers) will not be open to public until owner decide to enable trades using enableTrading function, this functionality creates a significant centralization risk as a malicious owner may not enable trades.

```
function enableTrading() external onlyOwner {
    require(!tradingEnabled, "Trading already enabled.");
    tradingEnabled = true;
    swapEnabled = true;
}
```

Contract is owned by Safu dev at early days after presale, so enabling trades is guaranteed and this issue is considered resolved.

Recommendations

To address this issue, a possible solution would be to have a decentralized mechanism in place that allows investors to enable trading, rather than solely relying on the owner to do so. This could be achieved by using a decentralized governance mechanism, such as a DAO, where token holders have the ability to propose and vote on changes to the contract, including the enabling of trading. This would ensure that the decision-making process is more democratic and less centralized.

DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.

ABOUT AUDITACE

We specializes in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.

https://auditace.tech/

https://t.me/Audit_Ace

https://twitter.com/auditace_

https://github.com/Audit-Ace