AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

David's Stone Power Movement

DATED : 6 July 2025

AUDIT SUMMARY

Project name – David's Stone Power Movement

Date: 6 July 2025

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

Audit Status: Passed

Issues Found

Status	Critical	High	Medium	Low	Informational
Open	0	0	0	1	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 conducted on the BSC Test network, and each test has a corresponding transaction attached to it.

3- Slither :
The code has undergone static analysis using Slither.

Token Information

Token Address:
0xf9560e39Cd7c73790199791E5025055B8D06586C

Name: David's Stone Power Movement

Symbol: DSPMOVE

Decimals: 9

Network: BscScan

Token Type: BEP-20

Owner: 0x0000000000000000000000000000000000000000

Deployer: 0x1e5F5376a4Ee9B0c0F39918E3B49ac7F756Ec922

Token Supply: 10000000000

Checksum: d8eefe8b4a9181615a455a0ddd290380

TOKEN OVERVIEW

Buy Fee: 0-0%

Sell Fee: 0-0%

Transfer Fee: 0-0%

Fee Privilege: Owner

Ownership: Owned

Minting: No

Max Tx: No

Blacklist: No

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

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

INHERITANCE TREE
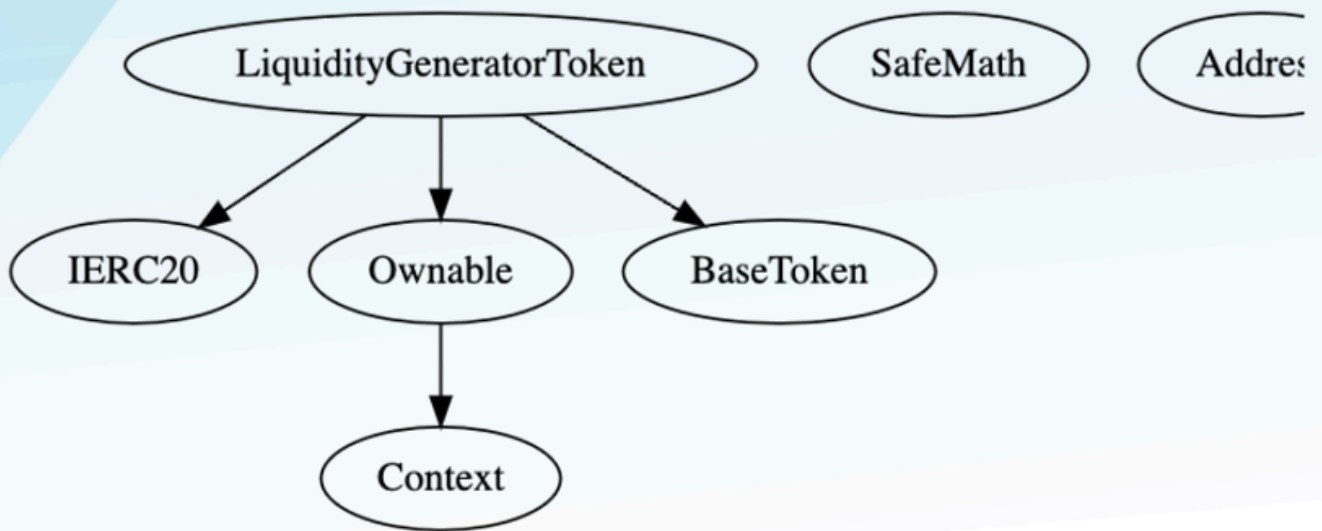
POINTS TO NOTE

- The ownership of the contract is renounced. Hence, the owner cannot change any settings in the contract.

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**	0
◆ **Medium-Risk**	0
◆ **Low-Risk**	1
◆ **Optimization/ Informational**	0

MANUAL TESTING

Optimization
Severity: Low
Subject: Remove Safe Math
Status: Open
Line: 205-416

Overview:

A compiler version above 0.8.0 can control arithmetic overflow/underflow; it is recommended to remove the unwanted code to avoid high gas fees.

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