Pepe Pizzeria

SMART CONTRACT

Security Audit

Platform

Solana

Audit Date

June 2024

Table of Contents

CAUTION

THIS DOCUMENT IS A SECURITY AUDIT REPORT AND MAY CONTAIN CONFIDENTIAL INFORMATION. THIS INCLUDES IDENTIFIED VULNERABILITIES AND MALICIOUS CODE THAT COULD BE USED TO COMPROMISE THE PROJECT. THIS DOCUMENT SHOULD ONLY BE FOR INTERNAL USE UNTIL ISSUES ARE RESOLVED. ONCE VULNERABILITIES ARE REMEDIATED, THIS REPORT CAN BE MADE PUBLIC. THE CONTENT OF THIS REPORT IS OWNED BY TimeVerse FOR THE USE OF THE CLIENT.

Project Summary

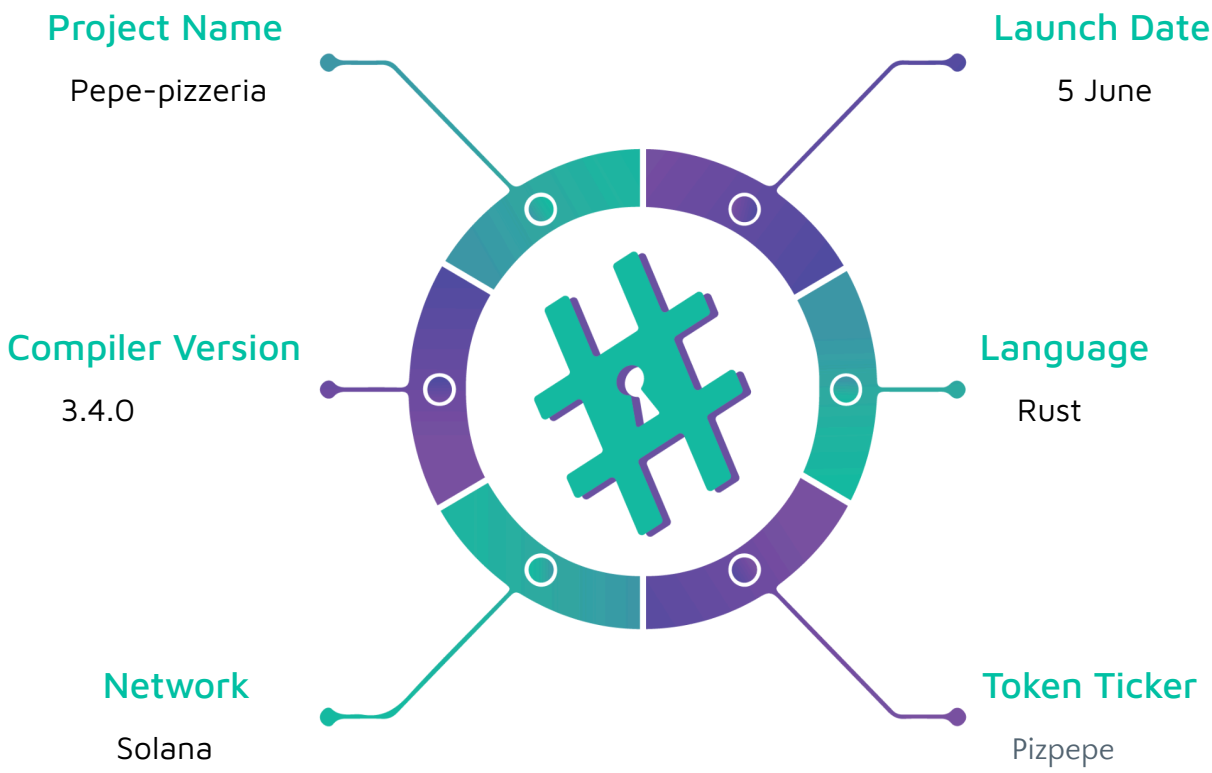
Project Name: Pepe Pizzeria

Contract Address: 4kLRpxuNPzViyhW3cKm5D9c4g2AKzVe2dtLi5cfUPvrY

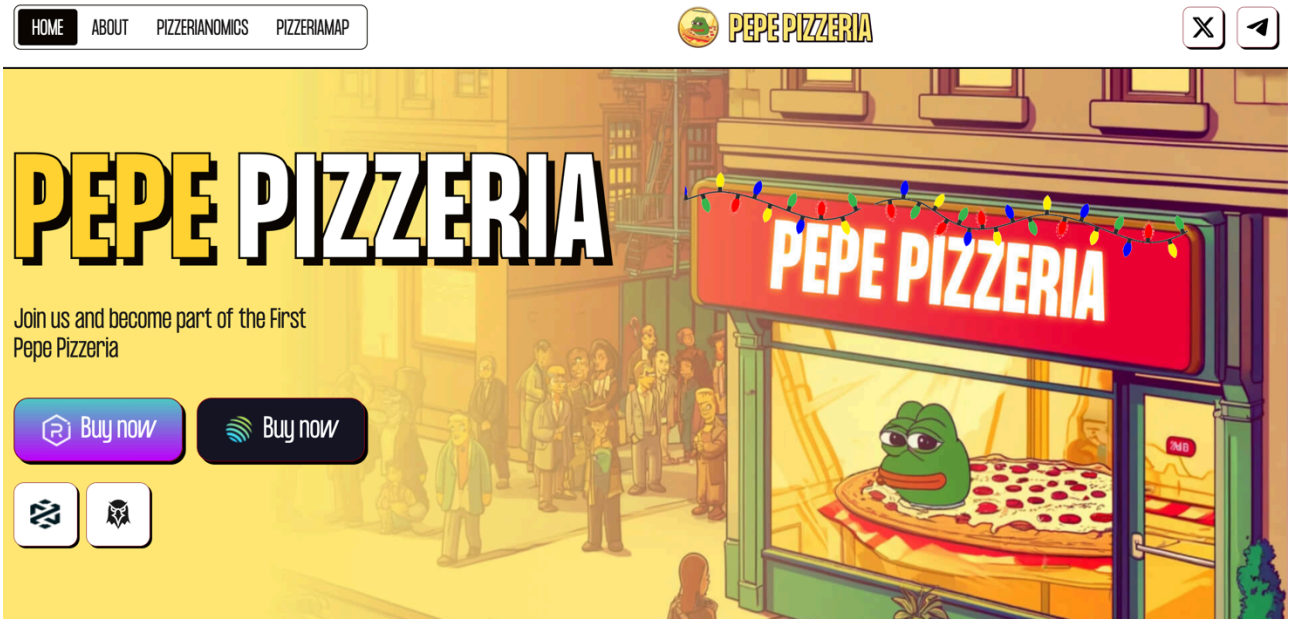
Website: https://www.pepe-pizzeria.com/

Logo:

Visualised Context:



Project Visuals:



Audit scope

Project	Contract
Pepe-pizzeria	4kLRpxuNPzViyhW3cKm5D9c4g2AKzVe2dtLi5cfUPvrY

Issues found:

0 High-severity vulnerabilities

0 Medium-severity vulnerabilities

0 Low-severity vulnerabilities

0 Gas Optimisations

Caution: *Timeverse's audits do not guarantee a project's success or ethics, and are not liable or responsible for security. Always conduct independent research about any project before interacting.*

Standardised Checks

Main Category	Subcategory	Result
General Code Checks	Solidity/compiler version stated	Passed
	Consistent pragma version across each contract	Passed
	Outdated Solidity Version	Passed
	Overflow/underflow	Passed
	Correct checks, effects, interaction order	Passed
	Lack of check on input parameters	Passed
	Function input parameters check bypass	Passed
	Correct Access control	Passed
	Built in emergency features	Passed
	Correct event logs	Passed
	Human/contract checks bypass	Passed
	Random number generation/use vulnerability	Passed
	Fallback function misuse	Passed
	Race condition	Passed
	Logical vulnerability	Passed
	Features claimed	Passed
	delegatecall() vulnerabilities	Passed
	Other programming issues	Passed
Code Specification	Correctly declared function visibility	Passed
	Correctly declared variable storage location	Passed
	Use keywords/functions to be deprecated	Passed
	Unused code	Passed
Gas Optimization	"Out of Gas" Issue	Passed

	High consumption ‘for/while’ loop	Passed
	High consumption ‘storage’ storage	Passed
	Assert() misuse	Passed
Tokenomics Risk	The maximum limit for mintage not set	Passed
	Owner can mint tokens	Passed
	Accurate Token Transfer Events	Passed
	Liquidity Burned	Passed
	“Short Address” Attack	Passed
	“Double Spend” Attack	Passed

Code Quality

This Audit scope involves the smart contracts of Pepe-Pizzerai. The contracts mostly follow standard best practices to help avoid unnecessary complexity that increases the likelihood of exploitation; however, some refactoring is required.

The code closely follows best practice nat-spec styling. All comments are correctly aligned with code functionality.

Severity Definitions

Significance	Description
High	High severity vulnerabilities can result in loss of funds, asset loss, access denial, and other critical issues that will result in the direct loss of funds and control by the owners and community.
Medium	Medium level difficulties should be solved before deployment, but won't result in loss of funds.
Low	Low level vulnerabilities are areas that lack best practices that may cause small complications in the future.
Gas	Gas Optimisations, issues and inefficiencies

Audit Findings

No significant Issue has been found in the audit

Conclusion

After TimeVerse analysis, the Pepe Pizzeria project seems to have a sound and well tested code base. Overall, most of the code is correctly ordered and follows industry best practices. To the best of our ability, TimeVerse is not able to identify any further vulnerability

Our Methodology

TimeVerse strives to maintain a transparent working process and to make our audits a collaborative effort. The objective of our security audits is to improve the quality of systems and upcoming projects we review and to aim for sufficient remediation to help protect users and project leaders. Below is the methodology we use in our security audit process.

Manual Code Review:
In manually analysing all of the code, we seek to find any potential issues with code logic, error handling, protocol and header parsing, cryptographic errors, and random number generators. We also watch for areas where more defensive programming could reduce the risk of future mistakes and speed up future audits. Although our primary focus is on the in-scope code, we examine dependency code and behaviour when it is relevant to a particular line of investigation.

Disclaimers

TimeVerse's Disclaimer

TimeVerse's team has analysed these smart contracts in accordance with the best industry practices at the date of this report, in relation to: cybersecurity vulnerabilities and issues in the smart contract source code, the details of which are disclosed in this report, (Source Code); the Source Code compilation, deployment and functionality (performing the intended functions).

Due to the fact that the total number of test cases are unlimited, the audit makes no statements or warranties on security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on

this report only. We also suggest conducting a bug bounty program to confirm the high level of security of this smart contract.

TimeVerse is not responsible for the safety of any funds and is not in any way liable for the security of the project.

Technical Disclaimer

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have their own vulnerabilities that can lead to attacks. Thus, the audit can't guarantee explicit security of the audited smart contracts.

About TimeVerse

TimeVerse aims to help facilitate the successful widespread adoption of distributed ledger technology. Our key services all have a focus on security, as well as projects that focus on streamlined adoption in the business sector.

TimeVerse is excited to continue to grow its partnerships with developers and other web3 oriented companies to collaborate on secure innovation, helping businesses and decentralised entities alike.

Github: https://github.com/Nabeel-javaid