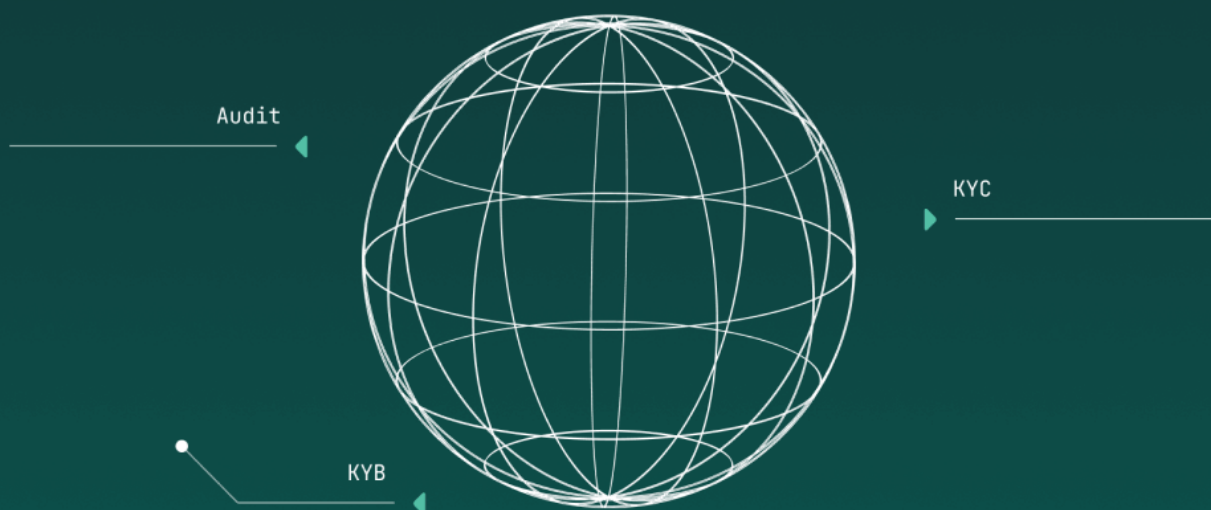




# SMART CONTRACT REVIEW AND SECURITY REPORT



COMPLETED ON  
**JULY 11, 2022**



DAudit.org



@daudit\_org



@support@dauddit.org

# OVERVIEW

This audit has been prepared for Pi Finance to review their Smart Contract Code and Security. This audit report aims to help investors make an informative decision during the project research.

In this report, you will find a summarized review of the following key points:

- ✓ Contract's source code
- ✓ Contract's function
- ✓ Owner's wallets
- ✓ Important Technical Stats
- ✓ Good Practices
- ✓ Recommendation

This document may contain confidential information about IT systems and the intellectual property of the Customer as well as information about potential vulnerabilities and methods of their exploitation.

The report containing confidential information can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed – upon a decision of the Customer.

► This Audit report DOES NOT guarantee nor reflect the outcome and goal of the project.

► DAudit's audit process only guarantees that the smart contract code has been verified not to have security breaches.

# Table Of Content

Overview	.....1
Contract Information	.....2
Daudit Contract Review Process	.....2
Project Technical Information	.....3
Important Stats	.....4
Vulnerability Check	.....5
Code Review	.....5
Function Review	.....6
Risk Level	.....7
Risk Found	.....8
Good Practices Found	.....11
About DAudit	.....12
Disclaimer	.....13

# CONTRACT INFORMATION

**Token Name**                      **Symbol**

PiFinance                          PIFI

**Contract Name**                **Type**

PiFinance                          ERC-20

## Website

<https://pifinance.tech/>

## Technical Documentation

[https://pifinance.tech/  
whitepaper.pdf](https://pifinance.tech/whitepaper.pdf)

## Contract Address

0x96db85c3fd8cc667E0bc0477E2b2864C  
43c8e44f

## Network

Binance Smart Chain

## Language

Solidity

## Compiler Version

v0.6.12+commit.27d51765

## Optimization

Yes with 200 runs

## Decimals

18

## Total Supply

100,000,000,000

# DAUDIT CONTRACT REVIEW PROCESS

Smart Contract Code review  
process:

- ✓ Testing the smart contracts against both common and uncommon vulnerabilities.
- ✓ Assessing the codebase to ensure compliance with current best practices and industry standards.
- ✓ Ensuring contract logic meets the specifications and intentions of the client.
- ✓ Cross-referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- ✓ Thorough line-by-line manual review of the entire codebase by industry experts.

DECENTRALAB PTE.LTD.

160 Robinson Road, #14-04 Singapore  
Singapore (068914)  
support@daudit.org



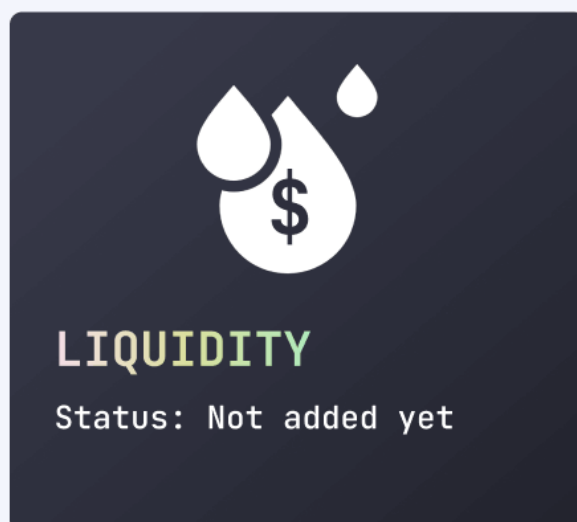
# PROJECT TECHNICAL INFORMATION

(AS OF JULY 11TH, 2022)

STATUS:

HAVEN'T LAUNCHED YET

Owner Address	0x4b7ae5cb609cbb99bae4e6ebee9f830f4c893852
Dev Wallet	0x39F46bd00fc7984Cb03b1DD35E88074a07b2bB49
Found Wallet	0xe80C88A5bC174fbf02D529C125335aF77Fa217E7
LP Address	0x79a3c1a9b0963841dd4de26f91cde471d1e5e912



## IMPORTANT STATS

### TAX

Buy tax: 10%  
Sell tax: 10%

### OWNER CAN SET FEES

Buy: Up to 100%  
Sell: Up to 100%

### MAX TX AMOUNT

Owner can't set max  
tx amount

### OWNERSHIP

Owner can renounce or  
transfer ownership

### MINT FUNCTION

No mint  
function found

### PAUSE

Owner can't  
pause trading

### BLACKLIST

Owner can set  
blacklist

### WHITELIST

Owner can set  
whitelist to avoid  
transaction fee



# VULNERABILITY CHECK

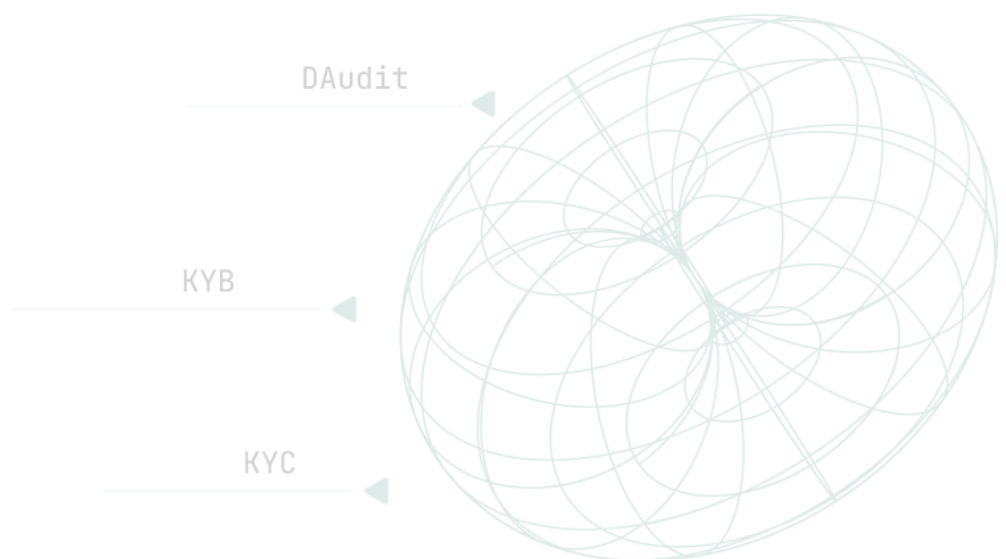
## CODE REVIEW

Design Logic	Passed
Compiler Warnings	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions and reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle Calls	Passed
Front Running	Passed
DoS with block gas limit	Passed
DoS with Revert	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event Log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

# VULNERABILITY CHECK

## FUNCTION REVIEW

Business Logics Review Functionality Checks	Passed
Access Control & Authorization	Passed
Escrow manipulation	Passed
Token Supply manipulation	Passed
Assets integrity	Passed
User Balances manipulation	Passed
Data Consistency manipulation	Passed
Kill - Switch Mechanism Operation Trails & Event Generation	Passed





# RISK LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

## Critical

Critical vulnerabilities are usually straightforward to exploit and can lead to asset loss or data manipulations.

## High

High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions

## Medium

Medium-level vulnerabilities are important to fix; however, they can't lead to asset loss or data manipulations.

## Low

Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.

# RISK FOUND 01

## CRITICAL

---

Owner can change total fees up to 100%.

```
function setFee(  
    uint256 _USDTRewardsFee,  
    uint256 _USDTRewardsShare,  
    uint256 _LPShare  
) public onlyOwner {  
    USDTRewardsFee = _USDTRewardsFee;  
    USDTRewardsShare = _USDTRewardsShare;  
    LPShare = _LPShare;  
}
```

### Recommendation:

Total selling and buying fees should be kept at  $\leq 25\%$

# RISK FOUND 02

## CRITICAL

Owner can set blacklist user, through which any user can be prohibited from trading.

```
function blacklistAddress(address account, bool value) external onlyOwner {
    _isBlacklisted[account] = value;
    emit BlacklistAddress(account, value);
}

function blacklistMultiAddresses(address[] calldata accounts, bool value)
    external
    onlyOwner
{
    for (uint256 i = 0; i < accounts.length; i++) {
        _isBlacklisted[accounts[i]] = value;
    }
    emit BlacklistMultiAddresses(accounts, value);
}
```

```
function _transfer(
    address from,
    address to,
    uint256 amount
) internal override {
    require(from != address(0), "ERC20: transfer from the zero address");
    require(to != address(0), "ERC20: transfer to the zero address");
    require(!_isBlacklisted[from], "Blacklisted address");
}
```

### Recommendation:

The blacklist function should not be used, this function can block the user's trading or transfer. Without this function, there could be more investors involved

# RISK FOUND 03

LOW

Owner can withdraw token and BNB from token contract

```
function rescueToken(address tokenAddress, uint256 tokens)
|   public
|   onlyOwner
|   returns (bool success)
{
|   return IBEP20(tokenAddress).transfer(msg.sender, tokens);
| }

function rescueBNB(address payable _recipient) public onlyOwner {
|   _recipient.transfer(address(this).balance);
| }
```

## Recommendation:

This action should be performed automatically by smart contract and should not be intervened manually.

## PI FINANCE GOOD PRACTICES FOUND

1 

The owner cannot mint new tokens after deployment.

2 

The owner cannot stop or pause the contract.

3 

The smart contract utilizes "SafeMath" to prevent overflows.

4 

The owner cannot limit transaction amount.





DECENTRALAB PTE.LTD.

160 Robinson Road, #14-04 Singapore  
Singapore (068914)  
support@daudit.org



## ABOUT DAUDIT

DAudit offers Smart Contract vulnerability and quality testing services at a rapid pace to ensure that projects do not fall behind the market.

Experienced 	Fast 	Careful 	Affordable 
A group of experienced blockchain developers built many successful DApp applications and are familiar with security flaws.	Within 6 hours, the audit report will be on your desk! We also have professional consultation and support staff available around the clock.	We deeply analyze the smart contracts line by line and cover the smart contracts with both automated and manual testing.	Affordable We provide the most competitive price in the industry, with audit reports ranging from \$500 to \$1,000, KYC services start at \$1000, and KYB services start at \$2,000

## CONTACT US

Email

support@daudit.org

Support 24/7

@daudit (Mr.Drake)

@vietdn (Mr.Viet)

daudit.org



# DISCLAIMER

## **DAudit Disclaimer**

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or print and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.

The smart contracts submitted for audit were examined in accordance with best industry practices at the time of this report in terms of cybersecurity vulnerabilities and issues in smart contract source code, which are detailed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no claims or guarantees about the code's security. It also cannot be deemed an adequate appraisal of the code's utility and safety, bug-free status, or any other contractual assertions. While we did our best in completing the study and generating this report, it is crucial to emphasize that you should not rely only on this report; we advocate doing many independent audits and participating in a public bug bounty program to assure smart contract security.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed

## **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 vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.