



# Coinsult

## Advanced Manual Smart Contract Audit



**Project:** DiggyStyle

**Website:** <https://gettindiggy.com/>

**Low-Risk**

3 low-risk code  
issues found

**Medium-Risk**

0 medium-risk code  
issues found

**High-Risk**

0 high-risk code  
issues found

**Contract Address**

0x9f21262F0CAA3D6803e484c70426fF316151dbd5

Disclaimer: Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

# Disclaimer

Coinsult is not responsible if a project turns out to be a scam, rug-pull or honeypot. We only provide a detailed analysis for your own research.

Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

The information provided in this audit is for informational purposes only and should not be considered investment advice. Coinsult does not endorse, recommend, support or suggest to invest in any project.

Coinsult can not be held responsible for when a project turns out to be a rug-pull, honeypot or scam.

# Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	0xc180ca3bd6a6ec27710f65d617636e82218a2bf0	1,000,000,000,000	100.0000%

# Source Code

Coinsult was comissioned by DiggyStyle to perform an audit based on the following smart contract:

<https://bscscan.com/address/0x9f21262F0CAA3D6803e484c70426fF316151dbd5#code>

**PinkSale generated token**

# Manual Code Review

In this audit report we will highlight all these issues:

## Low-Risk

3 low-risk code  
issues found

## Medium-Risk

0 medium-risk code  
issues found

## High-Risk

0 high-risk code  
issues found

The detailed report continues on the next page...

● **Low-Risk:** Could be fixed, will not bring problems.

## Too many digits

Literals with many digits are difficult to read and review.

```
function updateGasForProcessing(uint256 newValue) public onlyOwner {
    require(
        newValue >= 200000 && newValue <= 500000,
        "BABYTOKEN: gasForProcessing must be between 200,000 and 500,000"
    );
    require(
        newValue != gasForProcessing,
        "BABYTOKEN: Cannot update gasForProcessing to same value"
    );
    emit GasForProcessingUpdated(newValue, gasForProcessing);
    gasForProcessing = newValue;
}
```

## Recommendation

Use: Ether suffix, Time suffix, or The scientific notation

## Exploit scenario

```
contract MyContract{
    uint 1_ether = 1000000000000000000;
}
```

While 1\_ether looks like 1 ether, it is 10 ether. As a result, it's likely to be used incorrectly.

● **Low-Risk:** Could be fixed, will not bring problems.

## No zero address validation for some functions

Detect missing zero address validation.

```
function setMarketingWallet(address payable wallet) external onlyOwner {  
    _marketingWalletAddress = wallet;  
}
```

## Recommendation

Check that the new address is not zero.

## Exploit scenario

```
contract C {  
  
    modifier onlyAdmin {  
        if (msg.sender != owner) throw;  
        _;  
    }  
  
    function updateOwner(address newOwner) onlyAdmin external {  
        owner = newOwner;  
    }  
}
```

Bob calls updateOwner without specifying the newOwner, so Bob loses ownership of the contract.

● **Low-Risk:** Could be fixed, will not bring problems.

## Missing events arithmetic

Detect missing events for critical arithmetic parameters.

```
function setTokenRewardsFee(uint256 value) external onlyOwner {
    tokenRewardsFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setLiquidityFee(uint256 value) external onlyOwner {
    liquidityFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setMarketingFee(uint256 value) external onlyOwner {
    marketingFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}
```

## Recommendation

Emit an event for critical parameter changes.

## Exploit scenario

```
contract C {

    modifier onlyAdmin {
        if (msg.sender != owner) throw;
        _;
    }

    function updateOwner(address newOwner) onlyAdmin external {
        owner = newOwner;
    }
}
```

updateOwner() has no event, so it is difficult to track off-chain changes in the buy price.

## Owner privileges

- Owner cannot set fees higher than 25%
- Owner cannot pause trading
- Owner cannot change max transaction amount
- Owner can exclude from fees

## Extra notes by the team

No notes

# Contract Snapshot

```
contract BABYTOKEN is ERC20, Ownable, BaseToken {
    using SafeMath for uint256;

    uint256 public constant VERSION = 1;

    IUniswapV2Router02 public uniswapV2Router;
    address public uniswapV2Pair;

    bool private swapping;

    BABYTOKENDividendTracker public dividendTracker;

    address public rewardToken;

    uint256 public swapTokensAtAmount;

    uint256 public tokenRewardsFee;
    uint256 public liquidityFee;
    uint256 public marketingFee;
    uint256 public totalFees;

    address public _marketingWalletAddress;

    uint256 public gasForProcessing;

    // exclude from fees and max transaction amount
    mapping(address => bool) private _isExcludedFromFees;

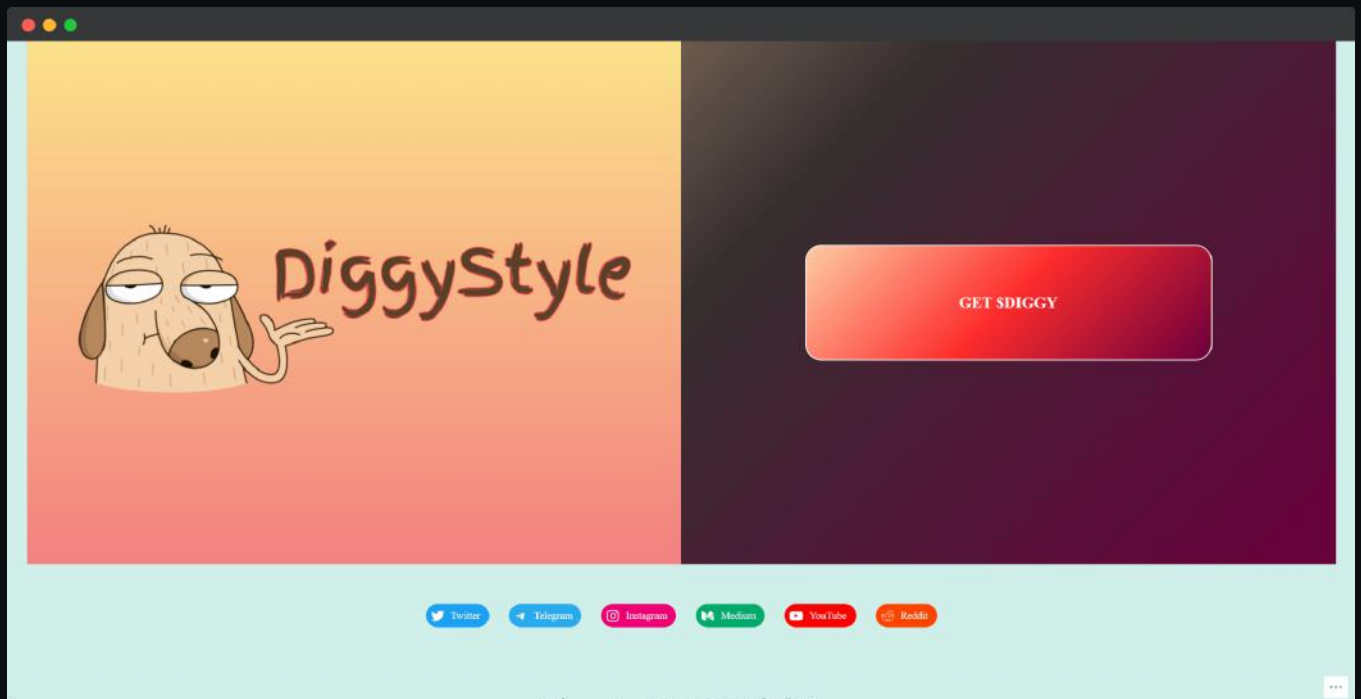
    // store addresses that a automatic market maker pairs. Any transfer *to* these addresses
    // could be subject to a maximum transfer amount
    mapping(address => bool) public automatedMarketMakerPairs;

    event UpdateDividendTracker(
        address indexed newAddress,
        address indexed oldAddress
    );
```



# Website Review

Coinsult checks the website completely manually and looks for visual, technical and textual errors. We also look at the security, speed and accessibility of the website. In short, a complete check to see if the website meets the current standard of the web development industry.



- Mobile Friendly
- Does not contain jQuery errors
- SSL Secured
- No major spelling errors

## Project Overview

● KYC verified by Coinsult partner

● Not KYC verified by Coinsult

# DiggyStyle

Completed KYC Verification at a Coinsult partner



✓ Project Owner Identified

✓ Contract: 0x9f21262F0CAA3D6803e484c70426fF316151dbd5

# DiggyStyle

Audited by Coinsult.net



Date: 13 July 2022

✓ Advanced Manual Smart Contract Audit

