



# Coinsult

## Advanced Manual Smart Contract Audit



**Project:** Milestone

**Website:** <https://milestone.quest>

● **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**

0xb20481c183565c9d9d965becabad900756fa60b1

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.

# Tokenomics

**Total Supply:** 12,500,000,000.00

**Total Holders:** 1

**Top 10 holders:**

Rank	Address	Quantity (Token)	Percentage
1	0x8725c3c1f22bf41226d340e59600f78120d91770	12,500,000,000	100.0000%

The top 100 holders collectively own 100.00% (12,500,000,000.00 Tokens) of Milestone.

Note: This is a snapshot of when the audit was performed.

# Source code

Coinsult was commissioned by Milestone to perform an audit based on the following smart contract:

<https://bscscan.com/address/0xb20481c183565c9d9d965becabad900756fa60b1#code>

# Manual Code Review

## ● Low-risk

3 low-risk code issues found.

Could be fixed, will not bring problems.

- Owner entitled to change the transaction fees

```
function setbuyTaxFees(uint256 taxFee,uint256 CharityFee,uint256
liquidityFee)external onlyOwner(){
    buyTaxFee = taxFee;
    buyCharityFee = CharityFee;
    buyLiquidityFee = liquidityFee;

    _taxFee = taxFee;
    _liquidityFee = liquidityFee;
    _CharityFee = CharityFee;

    _previousLiquidityFee = _liquidityFee;
    _previousCharityFee = _CharityFee;
    _previousTaxFee = _taxFee;

}

function setSellTaxFees(uint256 taxFee,uint256 CharityFee,uint256
liquidityFee)external onlyOwner(){
    SellTaxFee = taxFee;
    SellCharityFee = CharityFee;
    SellLiquidityFee = liquidityFee;

}
```

- Owner entitled to whitelist from fees

```
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}

function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}
```

- Owner entitled to whitelist from rewards

```
function excludeFromReward(address account) public onlyOwner() {
    require(account != 0x10ED43C718714eb63d5aA57B78B54704E256024E,
'We can not exclude Pancake router.');
```

```
    require(!_isExcluded[account], "Account is already excluded");
    if(_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}
```

## ● Medium-risk

0 medium-risk code issues found.

Should be fixed, could bring problems.

## ● High-risk

0 high-risk code issues found.

Must be fixed, and will bring problems.

## Extra notes by the team

Comments by the team about the code in general.

- Contract has a lot of commented code

```
/**
 * @dev Set The Amount To Start The Liquidation Process .
 * When This Amount Reached on The Contract , The Swap&Liquidity
Starts
 */
function num2Add2LP(uint256 num2Add2Liquidity) external onlyOwner {
    numTokensSellToAddToLiquidity = num2Add2Liquidity;
}

/**
 * @dev Set The Router Address .
```

```
    * IMPORTANT: You Shouldn't Change This Router Address Unless  
Pancakeswap Upgraded to V3 Router or So ,  
    * Do Some Research Before .  
    */
```

- Unused functions could be removed

```
function div(uint256 a, uint256 b) internal pure returns (uint256)  
{  
    return div(a, b, "SafeMath: division by zero");  
}  
  
function div(uint256 a, uint256 b, string memory errorMessage)  
internal pure returns (uint256) {  
    require(b > 0, errorMessage);  
    uint256 c = a / b;  
  
    return c;  
}  
  
function mod(uint256 a, uint256 b) internal pure returns (uint256)  
{  
    return mod(a, b, "SafeMath: modulo by zero");  
}  
  
function mod(uint256 a, uint256 b, string memory errorMessage)  
internal pure returns (uint256) {  
    require(b != 0, errorMessage);  
    return a % b;  
}
```

# Contract Snapshot

```
contract milestone is Context, IBEP20, Ownable {
    using SafeMath for uint256;
    using Address for address;

    mapping (address => uint256) private _rOwned;
    mapping (address => uint256) private _tOwned;
    mapping (address => mapping (address => uint256)) private
_allowances;
    mapping (address => bool) AutomaticMarketMaker;
    mapping (address => bool) private _isExcludedFromFee;
    address public CharityWallet =
0x934b49A213157c57039e26ee36d7e6f4E40A72cd;
    mapping (address => bool) private _isExcluded;
    address[] private _excluded;

    uint256 private constant MAX = ~uint256(0);
    uint256 private _tTotal = 125 *10**8 * 10**18;
    uint256 private _rTotal = (MAX - (MAX % _tTotal));
    uint256 private _tFeeTotal;

    string private _name = "Milestone";
    string private _symbol = "MSTONE";
    uint8 private _decimals = 18;

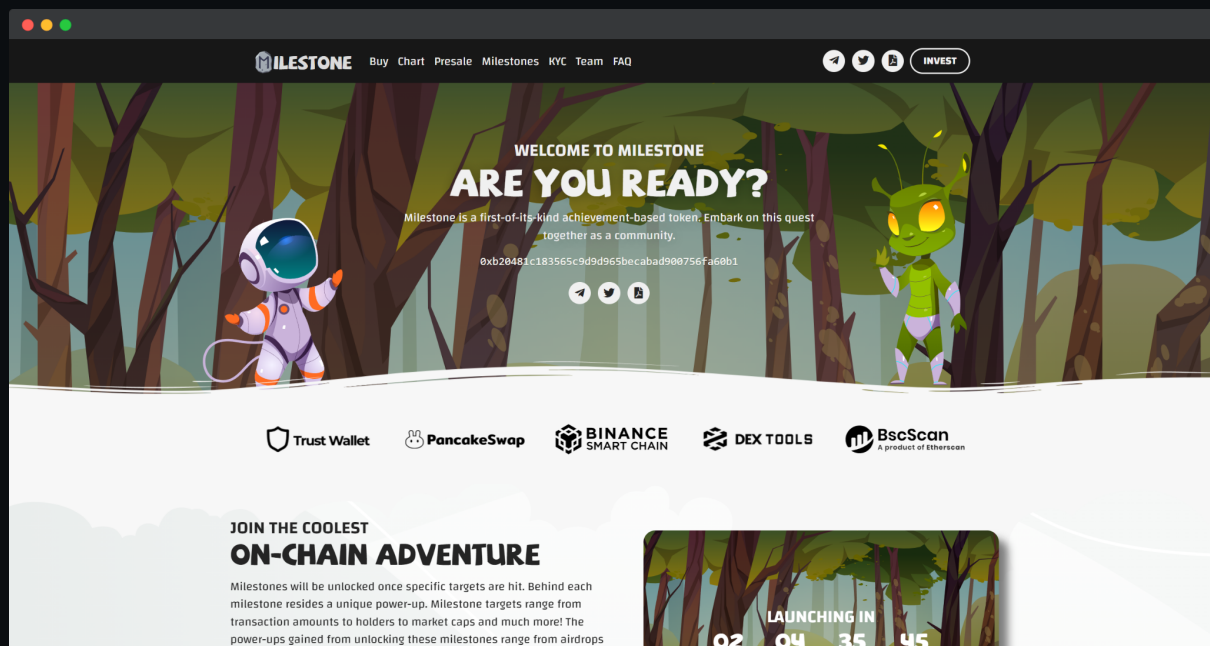
    uint256 private _taxFee = 2;
    uint256 private _previousTaxFee = _taxFee;

    uint256 private _liquidityFee = 5;
    uint256 private _previousLiquidityFee = _liquidityFee;

    uint256 private _CharityFee = 2;
    uint256 private _previousCharityFee = _CharityFee;

    uint256 public buyTaxFee = _taxFee;
    uint256 public buyLiquidityFee = _liquidityFee;
    uint256 public buyCharityFee = _CharityFee;
```

# 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
- No jQuery errors
- SSL Secured
- Appropriate spelling

Loading speed: 80%

# Rug-pull Review

Based on the available information analyzed by us, we come to the following conclusions:

- Locked Liquidity
- No large unlocked wallets
- No Doxxed Team

# Honeypot Review

Based on the available information analyzed by us, we come to the following conclusions:

- Ability to sell
- Owner unable to prevent selling
- Accurate liquidity pair

**Note:** Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by the project owner.