Coinsult

Advanced Manual
Smart Contract Audit

Project: Cromarket Token
Website: https://cromarket.app/

🟢 **Low-risk**

3 low-risk code
issues found

🟡 **Medium-risk**

1 medium-risk code
issues found

🔴 **High-risk**

0 high-risk code
issues found

Contract address
0xF02b03202e9200245fA179C13aDC11423e76F240

Disclaimer

Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	0x8b812820b123e77e25682dabfaf386876c3a5656	325,000	100.00%

Source code

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

https://cronoscan.com/address/0xf02b03202e9200245fa179c13adc11423e76f240#code

Manual Code Review

🟢 Low-risk

3 low-risk code issues found.
Could be fixed, will not bring problems.

- Contract contains Reentrancy vulnerabilities:
  _transferFrom(address,address,uint256)
  Additional information: This combination increases risk of malicious intent. While it
  may be justified by some complex mechanics (e.g. rebase, reflections, buyback).
  More information: Slither

```solidity
function _transferFrom(
    address sender,
    address recipient,
    uint256 amount
) internal returns (bool) {

    require(!blacklist[sender] && !blacklist[recipient],
"in_blacklist");

    if (inSwap) {
        return _basicTransfer(sender, recipient, amount);
    }
    if (shouldRebase()) {
        rebase();
    }

    if (shouldAddLiquidity()) {
        addLiquidity();
    }

    if (shouldSwapBack()) {
        swapBack();
    }

    uint256 gonAmount = amount.mul(_gonsPerFragment);
    _gonBalances[sender] = _gonBalances[sender].sub(gonAmount);
    uint256 gonAmountReceived = shouldTakeFee(sender, recipient)
        ? takeFee(sender, recipient, gonAmount)
```

```
        : gonAmount;
        _gonBalances[recipient] = _gonBalances[recipient].add(
            gonAmountReceived
        );

        if(!isDividendExempt[sender]){ try distributor.setShare(sender,
balanceOf(sender)) {} catch {} }
        if(!isDividendExempt[recipient]){ try
distributor.setShare(recipient, balanceOf(recipient)) {} catch {} }

        try distributor.process(distributorGas) {} catch {}

        emit Transfer(
            sender,
            recipient,
            gonAmountReceived.div(_gonsPerFragment)
        );
        return true;
    }
```

- Avoid relying on block.timestamp
  block.timestamp can be manipulated by miners.

```
block.timestamp
```

- Missing zero address validation
  Check that the new address is not the zero address.

```
    function setFeeReceivers(
      address _autoLiquidityReceiver,
      address _treasuryReceiver,
      address _autofirePit
  ) external onlyOwner {
      autoLiquidityReceiver = _autoLiquidityReceiver;
      treasuryReceiver = _treasuryReceiver;
      autofirePit = _autofirePit;
    }
```

● Medium-risk

1 medium-risk code issues found.
Should be fixed, could bring problems.

- If statements might never be reached
  If the second statement is reached (>= 365 days), then the other two below will not be called upon.

```
if (deltaTimeFromInit < (365 days)) {
    rebaseRate = 2355;
} else if (deltaTimeFromInit >= (365 days)) {
    rebaseRate = 211;
} else if (deltaTimeFromInit >= ((15 * 365 days) / 10)) {
    rebaseRate = 14;
} else if (deltaTimeFromInit >= (7 * 365 days)) {
    rebaseRate = 2;
}
```
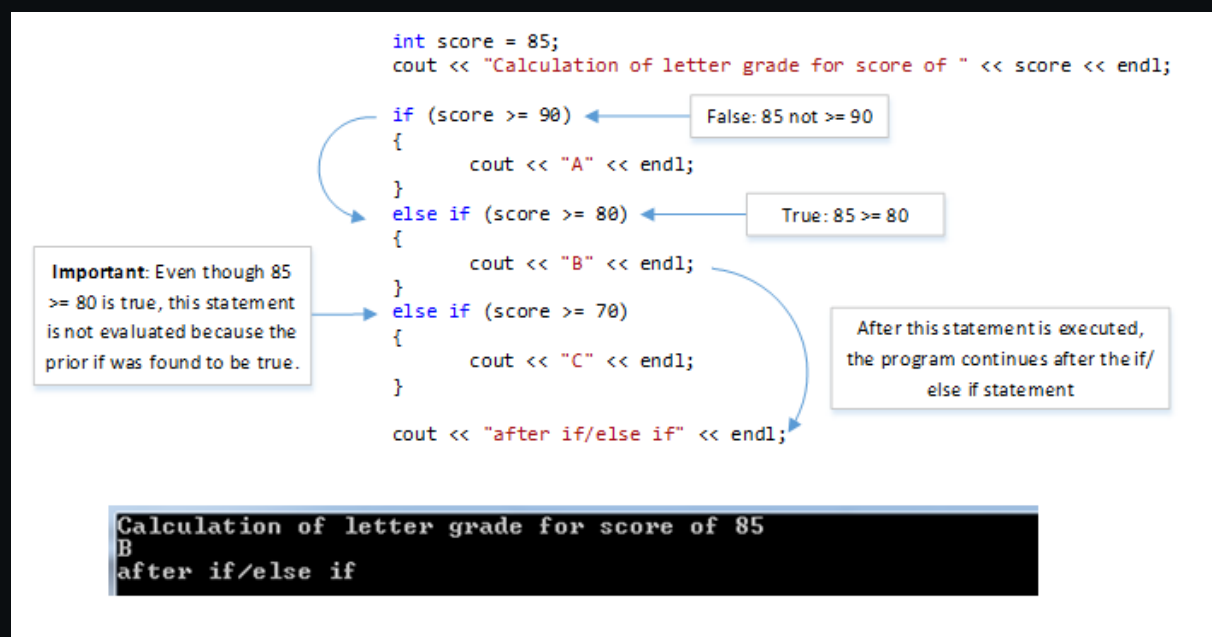

● High-risk

0 high-risk code issues found
Must be fixed, and will bring problems.

Extra notes by the team

🟢 Owner can blacklist contracts (bot prevention)

🟡 The ownership of the contract isn't renounced.

🟡 The owner can exclude from fees

🟡 Owner can exclude addresses from dividends.

🟡 Owner can change the router address

Contract Snapshot

```solidity
contract CromarketToken is ERC20Detailed, Ownable {

    using SafeMath for uint256;
    using SafeMathInt for int256;

    event LogRebase(uint256 indexed epoch, uint256 totalSupply);

    IPancakeSwapPair public pairContract;
    mapping(address => bool) _isFeeExempt;

    modifier validRecipient(address to) {
        require(to != address(0x0));
        _;
    }

    uint256 public constant DECIMALS = 5;
    uint256 public constant MAX_UINT256 = ~uint256(0);
    uint8 public constant RATE_DECIMALS = 7;

    uint256 private constant INITIAL_FRAGMENTS_SUPPLY =
    325 * 10**3 * 10**DECIMALS;

    uint256 public liquidityFee = 20;
    uint256 public treasuryFee = 25;
    uint256 public safuDividendFee = 70;
    uint256 public sellFee = 20;
    uint256 public autofirePitFee = 25;
    uint256 public totalFee =
    liquidityFee.add(treasuryFee).add(safuDividendFee).add(
        autofirePitFee
    );
    uint256 public feeDenominator = 1000;
```
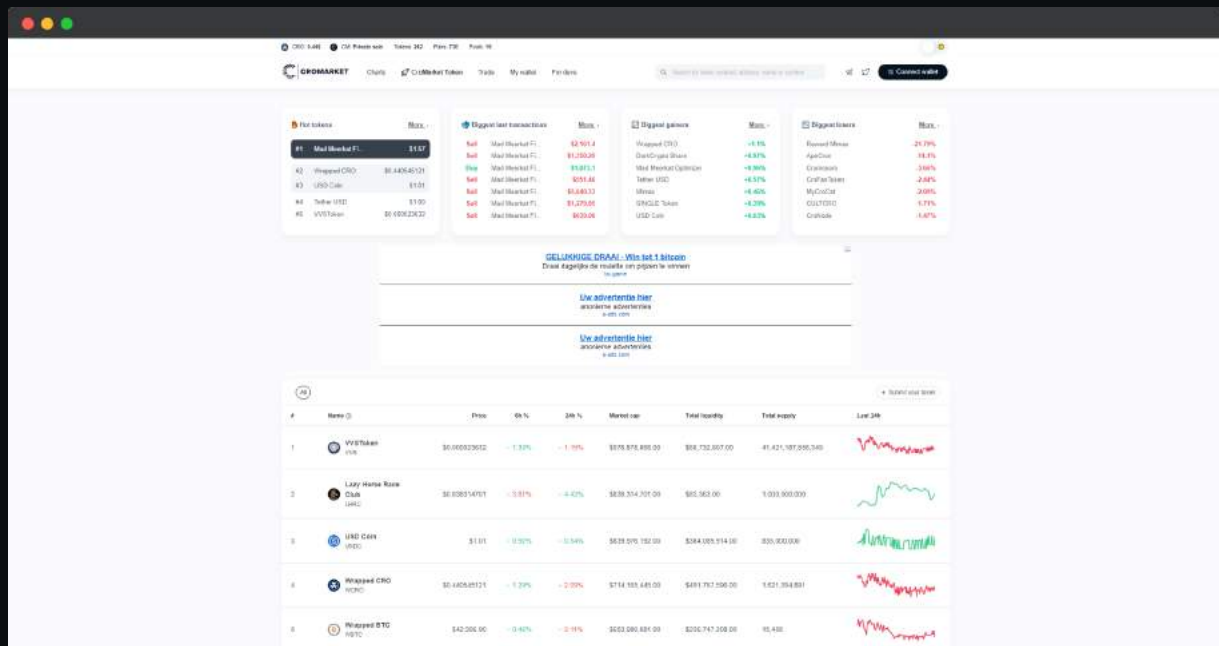
Website Review

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
🟢 Contains no jQuery errors
🟢 SSL Secured
🟢 No major spelling errors

Loading speed: 80%

Rug-pull Review

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

🟢 Locked Liquidity (no liquidity yet)

🟡 Large unlocked wallets
- Note: Tokens not distributed yet

🟢 Doxxed Team (KYC at pinksale)

Honeypot Review

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

🟢 Ability to sell

🟢 Owner is not able to pause the contract

🟡 Router can be changed

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.