



Coinsult

Advanced Manual Smart Contract Audit



Project: Jump Satoshi Token

Website: <https://jumpn.today>

Low-risk

2 low-risk code
issues found

Medium-risk

0 medium-risk code
issues found

High-risk

0 high-risk code
issues found

Contract address

0xEE6cacDDd3A9370d87dB581EE6728226883578e5

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	0x7b555ac2a2dda0f1f22b3b6b34cd9594d6700400	100,000,000,000	100.0000%

Source code

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

<https://bscscan.com/address/0xEE6cacDDd3A9370d87dB581EE6728226883578e5#code>

Manual Code Review

● Low-risk

2 low-risk code issues found.

Could be fixed, will not bring problems.

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

More information:

<https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp>

```
require(now <= expiry, "GovernanceDAO: delegateBySig: signature  
expired");
```

- Remove commented code / imports
50% of the contract code are comments, delete these comments for better readability and gas optimization.

```
// File: @openzeppelin/contracts/utils/Address.sol
```

● 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

- Contract is put together using openzeppeling imported contracts in 1 contract

The dev team chose to not import these files but to add them manually into 1 single contract

```
// File: @openzeppelin/contracts/utils/Address.sol
```

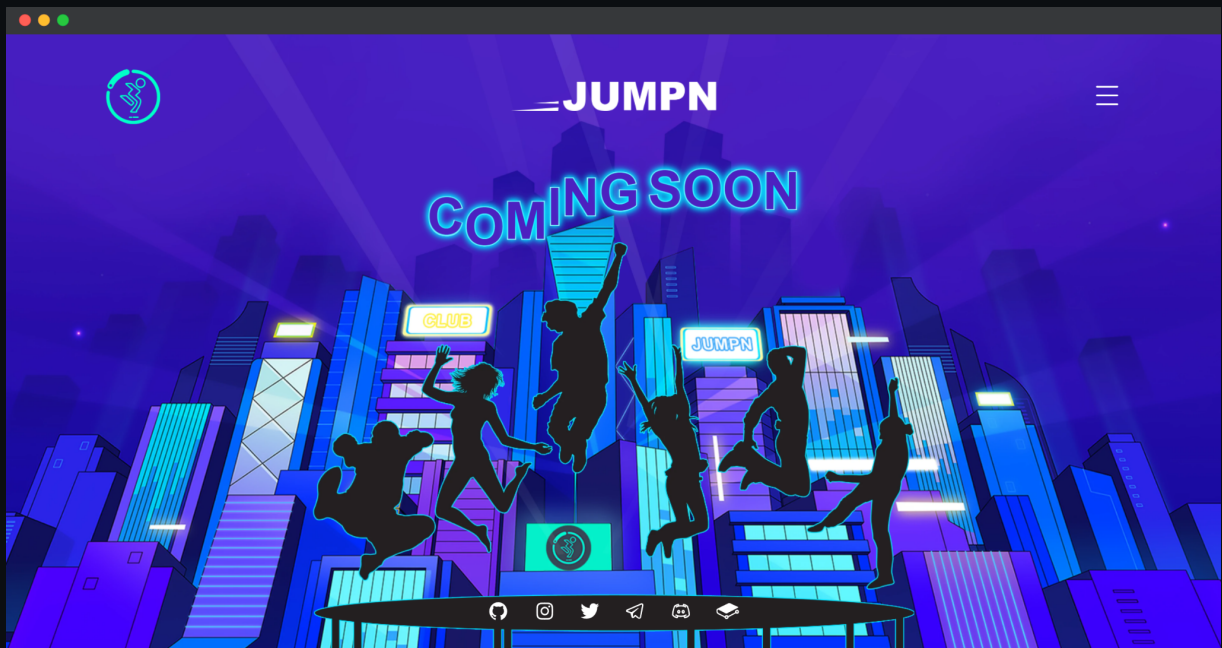
Contract Snapshot

```
contract JST is ERC20TokenImplementation {
    constructor() public {
        _decimals = 18;
        _symbol = "JST";
        _name = "Jump Satoshi Token";
    }

    /**
     * @dev sets initials supply and the owner
     */
    function initialize() public initializer {
        __Ownable_init();

        _totalSupply = 100_000_000_000 * (10 ** uint256(_decimals));
        _balances[owner()] = _totalSupply;
        emit Transfer(address(0), owner(), _totalSupply);
    }
}
```

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: 92%

Review

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

- Locked Liquidity - No liquidity yet
- Large unlocked wallets - Tokens not yet distributed
- No KYC at Coinsult
- Ability to sell
- Owner is not able to pause the contract

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.