

#### CryptoCowboys (CCS)

BEP-20 Audit Performed at block 11083561

PERFORMED BY DESSERT FINANCE FOR CONTRACT ADDRESS: 0xbD3255EEd179b0fB80e1A6743d6cccB2a18c6C2e

#### **INITIAL DISCLAIMER**

Dessert Finance provides due-diligence project audits for various BSC projects. Dessert Finance in no way guarantees that a project will not remove liquidity, sell off team supply, or otherwise exit scam.

Dessert Finance does the legwork and provides public information about the project in an easy-to-understand format for the common person.

Agreeing to a project audit can be seen as a sign of confidence and is generally the first sign of trust for a project, but in no way guarantees that a team will not remove *all* liquidity ("Rug Pull"), sell off tokens, or completely exit scam. There is also no way to prevent private sale holders from selling off their tokens. It is ultimately your responsibility to read through all documentation, social media posts, and contract code of each individual project to draw your own conclusions and set your own risk tolerance.

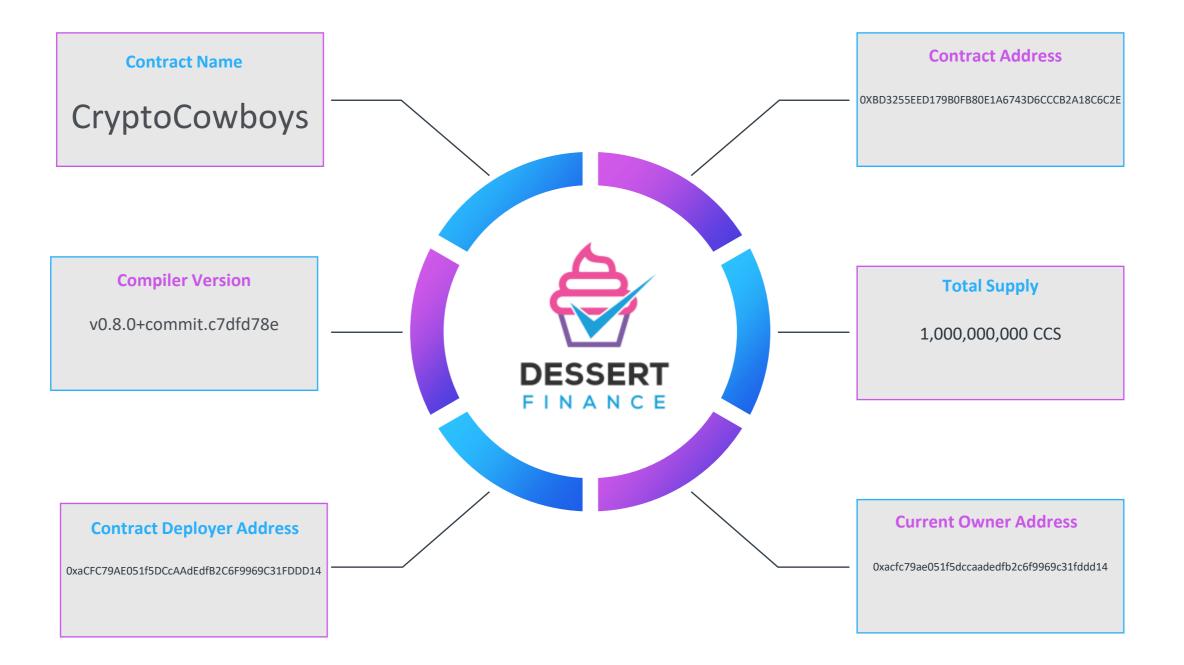
Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice.

#### **Table of Contents**



- 1. Contract Code Audit Token Overview
- 2. BEP-20 Contract Code Audit Overview
- 3. BEP-20 Contract Code Audit Vulnerabilities Checked
- 4. Contract Code Audit Contract Ownership
- 5. Contract Code Audit Mint Functions
- 6. Contract Transaction Fees
- 7. Website Overview
- 8. Social Media
- 9. Top Token Holders/Wallets
- 10. Location Audit
- 11. Review of Team
- 12. Roadmap
- 13. Disclaimers

#### **Contract Code Audit – Token Overview**



#### **BEP-20 Contract Code Audit – Overview**

Dessert Finance was commissioned to perform an audit on CryptoCowboys (CCS)

<pre>/** **semicited for verification at Backson.com == 2021-00-16 ** ** ** ** ** ** ** ** ** ** ** ** **</pre>	•/	cation at DecScancow on 2021-09-15
<pre>"submitted for verification at BacKan.com as 2021-00-10 "" " " " " " " " " " " " " " " " " "</pre>	-	
<pre>*/ pragma solidity 0.1.5  /** * place Interface of the UE2D standard as defined in the ETF. *** * place Interface of the UE2D standard as defined in the ETF. *** *** *** *** *** *** *** *** *** *</pre>		
<pre>*/ pragma salidity 0.1.5;  pragma salidity 0.1.5;  /** * place Interface of the UNCOD standard as defined in the ETP. * place Interface of the UNCOD standard as defined in the ETP. * place Interface IENCOD {     /**</pre>		
<pre>*/ prages selidity 0.8.5 /** * plev Interface of the EMC20 standard as defined in the ETP. *** *** *** *** *** *** *** *** *** *</pre>		
<pre>/**     plow Interface of the UECD standard as defined in the UTP.     plow Interface IERC20 {         *         #         *         #</pre>		Relieve 1 MER
<pre>/**     #dex Interface of the UE2D standard as defined in the UD?     ##     # down fatures the amount of tokens in existence.     ##     # down fatures the amount of tokens in existence.     ##     # down fatures the amount of tokens mened by "account".     ##     # down fatures the amount of tokens mened by "account".     ##     # down fatures the amount of tokens mened by "account".     ##     # down fatures the amount of tokens mened by "account".     ##     # down fatures the amount of tokens mened by "account".     ##     # down fatures the amount of tokens mened by "account".     ##     # down fatures the amount of tokens mened by "account".     ##     # down fatures the menont of tokens mened by "account".     ##     # down fatures the menont of tokens mened by "account".     ##     # down fatures the menont of tokens mened by "account".     ##     # down fatures the menont of tokens the caller's account to "excludes".     ##     # down fatures the menont induction when allocating whether the mereation succeeded.     ##     # down fatures the remaining number of tokens that "spender" will be     # allocating to be menone behalf of "seme" through (transferires), this is     ##     # toke thenges uses (approve) ar (transferires) are called.     #     # Toke value thenges uses (approve) ar (transferires) are called.     ##     # toke thenges uses (approve) ar (transferires) are called.     ##     # toke thenges uses (approve) ar (transferires) are called.     ##     # toke thenges uses (approve) ar (transferires) are called.     ##     ##########################</pre>		
<pre>* place Interface of the URC20 standard as defined in the UP: * place Interface IERC20 { * place Enterns the amount of tokens in existence. * function totalogopis() external view returns (wint256); ** * place Enterns the amount of tokens mened by "account". ** * place Enterns the amount of tokens mened by "account". ** * place Enterns the amount of tokens mened by "account". ** * place Enterns the amount of tokens mened by "account". ** * place Enterns the amount of tokens mened by "account". ** * place Enterns the amount of tokens from the caller's account to "exclpient". * factors a builder value Enditating whether the operation maccounted. * facts a (france) avant. * factor transfer(address exclpient, wint256 amount) external returns (bool); ** * place Enterns the remaining number of tokens that "spender" will be * allow the remaining number of tokens that "spender" will be * allow the theorem is build of "sense" through (transferfree), this is * even by default. * This value theorem (address mere, places (power) external view returns (wint256); ** * defaurts allower (address mere, address (power) external view returns (wint256); ** * tota value theorem (address mere, address (power) external view returns (wint256); ** * factors allower (address mere, address (power) external view returns (wint256); ** * factors a booleam value indicating whether the merestion maccounted. * Returns a booleam value indicating whether the merestion maccounted.</pre>		
<pre>interface IERCR0 {     if prove factores the amount of tokens in existence.     if prove factores the amount of tokens in existence.     if prove interfactores the amount of tokens moved by 'account'.     if and balanceOf(address account) external view returns (uint256);     if a disc factores the amount' takens from the caller's account to 'recipient'.     forction balanceOf(address account) external view returns (uint256);     if a disc factore' amount' takens from the caller's account to 'recipient'.     forction transfor(address account) external view returns (uint256);     if a disc factore' amount' takens from the caller's account to 'recipient'.     forction transfor(address excipient, uint256 amount) external returns (boel);     if a disc factores the remaining number of takens that 'spender' will be     salesed to spend to the actual of 'memor' through (transferirem), This is     trat usine thanges alone (approve) ar (transferirem) are called.     forction allower(address more, address ipender' even the caller's takens.     forction allower(address more, address ipender' even the caller's takens.     forction allower(address more, address ipender' even the caller's takens.     forction allower(address more, address ipender' even the caller's takens.     forction allower(address more, address ipender' even the caller's takens.     forction allower(address more, address ipender' even the caller's takens.     forction allower(address more, address ipender' even the caller's takens.     forction allower(address more) are the allower(address more) are caller.     forction allower(address more) are the allower(address more) are caller's takens.     forction allower(address more) are the allower(address more) are the caller's takens.     force in a boolean value indicating address more more the caller's takens.     force in a boolean value indicating address more more in a address more) are the caller's takens.     force in a boolean value indicating address in andress in address in a address in a addres</pre>		to 19529 standard as defined in the 17P.
<pre>/** * gives fatures the amount of tokens in existence. /* /** * gives fatures the amount of tokens memod by "account". /** * gives fatures the amount of tokens memod by "account". /** * gives haven "amount" tokens memod by "account". * fatures a busines reason" tokens from the caller's account to "excipient". * fatures a busines value indicating whether the specalion secrement. * fatures a busines reason of tokens that "species" (dot256); /** * gives fatures the remaining number of tokens that "species" (dot256); /** * silened to specie in behalf of "sener" through (transferires), this is * security default. * Toka token theremaining number of tokens that "species" will be * silened to specie in behalf of "sener" through (transferires), this is * security default. * Toka token theremaining number, given through (transferires), this is * security default. * Toka token theremaining number, given through (transferires), token is * security default. * Toke tokens a the allowers of "species" even the caller's tokens. ** *********************************</pre>		
<pre>/** /** /** /** /** /** /** /** /** /**</pre>		
<pre>/** * does fortures the ansatt of takens meend by "account". ** ** ** ** ** ** ** ** ** ** ** ** **</pre>	* goes Returns the	amount of tokans in existence.
<pre>* does Note::: /// does Note::: // does Note:: // does Note::</pre>	function totalSupple	() esternal view returns (wint256);
<pre>* does Note::: does Note::: function balanceO*(eddress account) external view returns (uint256); /** * does Note:::account: takens from the caller's account to 'recipient', * Noterns a boolean value indicating whether the operation succeeded. * Ints: a (fransfer) event. */ function transfer(address recipient, sint256 amount) external returns (bool); /** * does Noterns the remaining number of takens that 'sponder' will be * allowed to spend on behalf of 'sener' through (transfer?ex), this is * rece by setult. * This value thanges when (approve) ar (transfer?ex) are called. */ function allower(address mean; address spender) external view return (sint256); /** * does Sets 'amount' as the allowers of 'spender' over the caller's tokens. * Returns a boolean value indicating whether the operation succeeded. * Returns a boolean value indicating whether the operation succeeded. * Returns a boolean value indicating whether the operation succeeded.</pre>		
<pre>/**     #dex Moves 'assumt' takens from the caller's account to 'recipient',     # Returns a boolean value indicating whether the specaline successing,     # Tests: a ('recycler') weet;     # Tests: a ('recycler') weet;     # Tests: a ('recycler') weet;     # Tests: transfer(address recipient, similif assumt) external returns (boel);  /**     # dimed for special to resulting number of takens that 'speciar' will be     * allowed to special to the resulting number of takens that 'speciar' will be     * address the resulting number of takens that 'speciar' will be     * address the resulting number of takens that 'speciar' will be     * address the resulting number of takens that 'speciar' will be     * address the resulting number of takens that 'speciar' will be     * address takens (approve) be (transferieval) will be     * for address the speciar of 'speciar' over the caller's takens.     **     * dimen Sets 'securit' as the allowers of 'speciar' over the caller's takens.     **     * for any a boolean value indicating whether the special macceeded.     **</pre>		amount of tokens cannot by "account".
<pre>i gden Moorn 'amount' tokens from the caller's account to 'excludent'.     Returns a busines value indicating whether the specalies succeeded.     (mits a (Trunsfor) event.     function transfor(address recipient, wint156 amount) external returns (bool);      **     * defer Notions the remaining number of tokens that 'specder' will be     * alimed to specif on behalf of 'somer' through (transferiere), this is     * zero by defenit.     * This value thanges when (approve) or (transferiere) are called.     forction allowers(address moner, address specder) external view returns (wint256);      **     * defen Sets 'smoort' as the allowers of 'specder' even the caller's tokens.     * Returns a busines value indicating whether the specaled.     ***</pre>	Function balanceOf Co	ddrenn account) external view returns (uist256);
<pre>i gden Moorn 'amount' tokens from the caller's account to 'excludent'.     Returns a busines value indicating whether the specalies succeeded.     (mits a (Trunsfor) event.     function transfor(address recipient, wint156 amount) external returns (bool);      **     * defer Notions the remaining number of tokens that 'specder' will be     * alimed to specif on behalf of 'somer' through (transferiere), this is     * zero by defenit.     * This value thanges when (approve) or (transferiere) are called.     forction allowers(address moner, address specder) external view returns (wint256);      **     * defen Sets 'smoort' as the allowers of 'specder' even the caller's tokens.     * Returns a busines value indicating whether the specaled.     ***</pre>	2440	
<pre>inits a (franctor) event. function transfor(address recipient, wint155 amount) external returns (bool);  /** * Spice Notions the remaining number of takens that 'spender' will be * allowed to spend on behalf of 'sener' through (transferfrom), this is * serve by default. * This value thanges when (approve) ar (transferfrom) are called. * * This value thanges when (approve) ar (transferfrom) external view returns (wint256);  /** * defaults are as the allowerse of 'spender' even the caller's takens. * * Returns a boolean value indicating whether the moreation nucceeded. *** *****************************</pre>		
<pre>// function transfor(address exclution, similif amount) external returns (beel); /** * give Notions the remaining number of tokens that 'spender' will be * allowed to spend on behalf of 'somer' through (transforfrom), this is * rere by default. * This value thanges also (approve) or (transforfrom) are called. */ function allowere(address memory address spender) external view returns (wint256); /** * defaults for 'somer' as the allowerce of 'spender' memories called's tokens. * defaults a boolean value indicating shetter the spender.</pre>	* Arturns a busieum	value indicating whether the operation succeeded.
<pre>*/ function transfor(address exclution, similif second) external returns (beel); /** * dised to spend on behalf of "some" through (transforfrom). This is * allowed to spend on behalf of "some" through (transforfrom). This is * area by default. * This value thanges also (approve) or (transforfrom) are called. */ function allowers(address momer, address spender) external view returns (sint256); /** * disc Sets "secant" as the allowerse of "spender" nowe the caller's tokens. * discusses a boolses value indicating shorther the operation succeeded. * * * * * * * * * * * * * * * * * * *</pre>	+ Inits a Demotion	
<pre>/**  * give Netures the remaining number of tokens that 'spender' will be  * allowed to spend on behalf of 'owner' through (transferfrom), This is  * rem by default.  * This value thanges when (approve) or (transferfrom) are called.  */ * This value (address memor, address spender) external view returns (wint256); /**  * does Sets 'amount' as the allowerch of 'spender' memorial view returns, tokens,  * does Sets 'amount' as the allowerch of 'spender' memorial view returns, tokens,  * does solve value indicating shetter the momentum succeeded.  *</pre>		
<pre>* game factors the remaining number of tokens that 'spender' will be * allowed to spend on behalf of 'sener' through (transferfrom), This is * rem by default. * This value thanges when (approve) or (transferfrom) are called. */ function allowers(address memor, address ipender) esternal view returns (win1256); /** * Betwrns a boolsam value indicating whether the operation succeeded. * Returns a boolsam value indicating whether the operation succeeded. *********************************</pre>	function transfer(at	dress recipionT, wint156 amount) external (-turns (boel);
* allowed to spend on buhalf of "owner" through (transferfrom), This is * zero by default. * This value changes when (approve) or (transferfrom) are called. */ function allower (address noner, address Spender) esternal view referrs (wint256); /** * @des Sets "amoust" as the allowerce of "spender" owner the caller's tokens. * Referrs a boolean value indicating whether the operation succeeded. * Referrs a boolean value indicating whether the operation succeeded.		and the second
<pre>// // // // // // // // // // // // //</pre>	* allowed to spend :	in behalf of "sener" through (transforfrom). This is
function allowers(address means, address spender) external view returns (sint256); /** * given Sets "answert" as the allowerse of "spender" over the caller's tokens. * Returns a boolean value indicating abother the operation succeeded.	. This value charge	s when (approve) or (transferires) are called.
<pre>/** give Sets 'answer' as the allowers of 'spender' over the caller's tokens.     Returns a boolean value indicating shether the operation succeeded. </pre>		
<ul> <li>give Sets "answer" as the allowerss of "spender" over the caller's tokens.</li> <li>Returns a boolean value indicating whether the operation succeeded.</li> </ul>		and the control of the second s
Returns a boolson value indicating whether the operation succeeded.		as the allowers of 'spender' over the called's tokens.
* INFORTABLY Reward that changing an alimance with this method brings the risk		
• that there is a real both the cld and the sum allowance by preferrative.	· INCOMMENT Residen	that changing as allowance with this method brings the risk
<ul> <li>INVORTANT: Newsre that changing an alimetric with this method brings the risk.</li> <li>that someone may use both the sid and the new alimetric by unfortunate.</li> <li>transaction ordering. One possible solution to whighte this rate.</li> <li>condition is to first reduce the specific alimetric to 0 and set the</li> </ul>	• transaction order	ing. One possible solution to mitigate this rate
<ul> <li>desired value afterwardii</li> </ul>	<ul> <li>desired value aft</li> </ul>	erwards.
function approve(address spender, wint256 amount) external returns (bool);		ress spender, wint256 mount) external returns (buol);
	Name of Column 2 is not a design of the local division of the loca	

Contract Address 0xbD3255EEd179b0fB80e1A6743d6cccB2a18c6C2e

TokenTracker CryptoCowboys (CCS)

Contract Creator 0xacfc79ae051f5dccaadedfb2c6f9969c31fddd14

**Source Code** Contract Source Code Verified (Exact Match)

Contract Name CryptoCowboys

Other Settings default evmVersion, None

**Compiler Version** 0.8.0+commit.c7dfd78e

**Optimization Enabled** Yes with 200 runs

Code is truncated to fit the constraints of this document. The code in its entirety can be viewed here.

The contract code is **verified** on BSCScan.

### **BEP-20 Contract Code Audit – Vulnerabilities Checked**

Vulnerability Tested	Al Scan	Human Review	Result
Integer Overflow	Complete	Complete	√ Low / No Risk
Integer Underflow	Complete	Complete	√ Low / No Risk
Correct Token Standards Implementation	Complete	Complete	√ Low / No Risk
Timestamp Dependency for Crucial Functions	Complete	Complete	√ Low / No Risk
Exposed _Transfer Function	Complete	Complete	√ Low / No Risk
Transaction-Ordering Dependency	Complete	Complete	√ Low / No Risk
Unchecked Call Return Variable	Complete	Complete	√ Low / No Risk
Use of Deprecated Functions	Complete	Complete	√ Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	√ Low / No Risk
State Variable Default Visibility	Complete	Complete	√ Low / No Risk
Deployer Can Access User Funds	Complete	Complete	√ Low / No Risk

The contract code is **verified** on BSCScan.

The vulnerabilities listed above were not found in the token's Smart Contract.

#### **Contract Code Audit – Contract Ownership**

#### **Contract Ownership has not been renounced at the time of Audit**



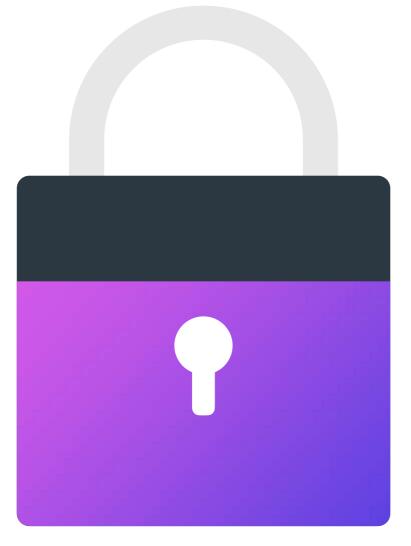
The contract ownership is not currently renounced.

We have placed the contract owner address below for your viewing:

Owner Address: 0xacfc79ae051f5dccaadedfb2c6f9969c31fddd14

#### **Contract Code Audit – Mint Functions**

This Contract Cannot Mint New CCS Tokens.



We do understand that sometimes mint functions are essential to the functionality of the project. If a mint function is ever listed we will have the team clarify its use case for transparency purposes.

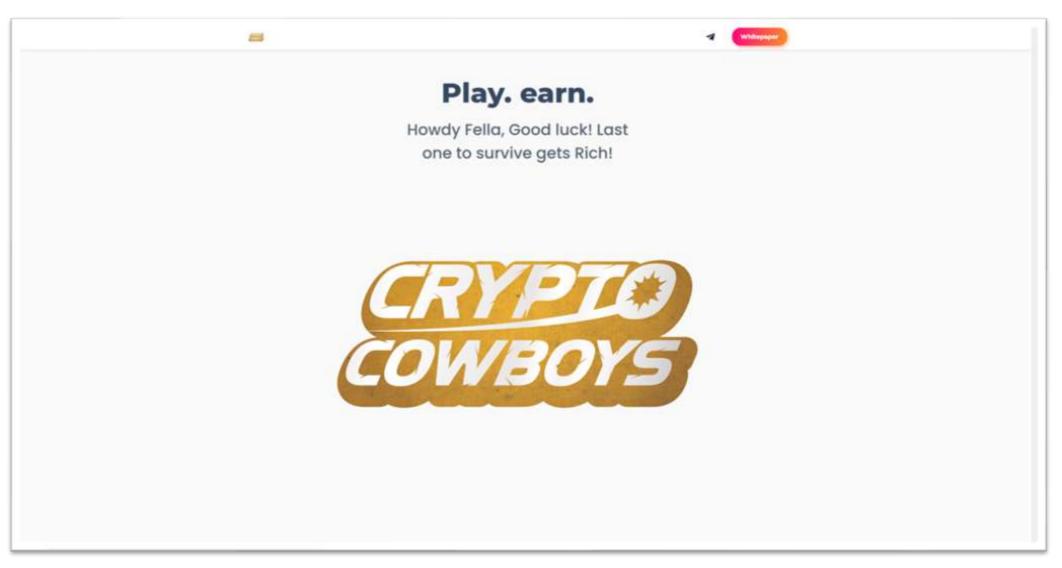
A mint function was not found in the contract code.

### **Contract Transaction Fees**

At the time of Audit the transaction fees ("tax") listed below are the fees associated with trading. These fees are taken from every buy and sell transaction unless otherwise stated.

Buy Tax	
10%	
UNKNOWN	
Sell Tax	
10%	
UNKNOWN	

## Website Part 1 – Overview www.cryptocowboysbsc.com



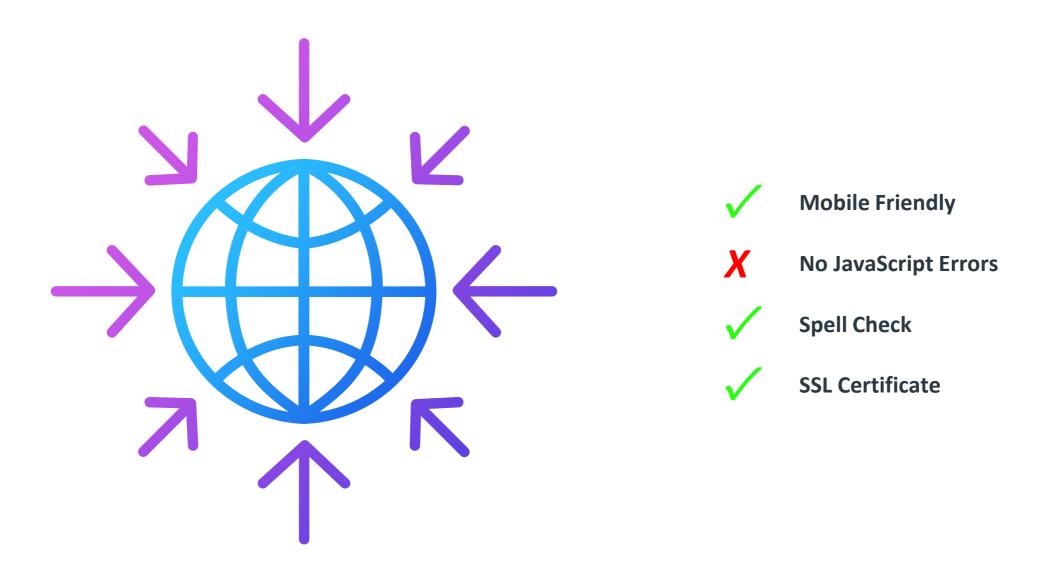
Above images are actual snapshots of the current live website of the project.

Website was registered on 09/09/2021, registration expires 09/09/2022.

**X** This does not meet the 3 year minimum we like to see on new projects.



#### Website Part 2 – Checklist



The website contained some JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

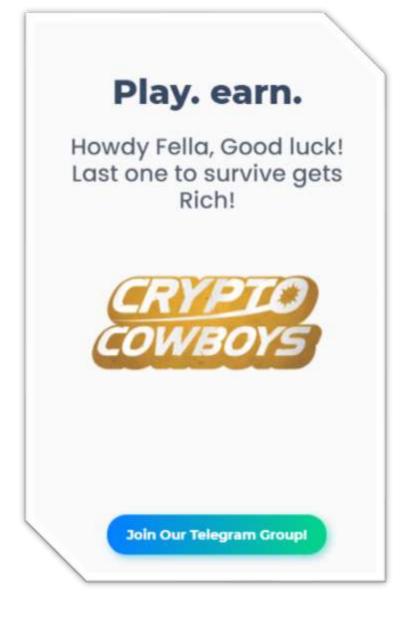
No additional issues were found on the website.

#### Website Part 3 – Responsive HTML5 & CSS3

No issues were found on the Mobile Friendly check for the website. All elements loaded properly and browser resize was not an issue. The team has put a considerable amount of thought and effort into making sure their website looks great on all screens.

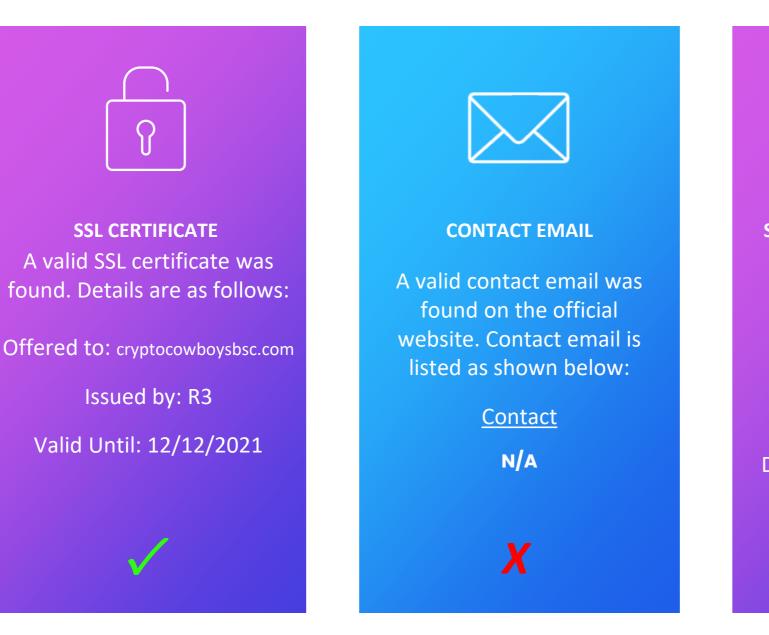
Some JavaScript errors were found. No issues with loading elements, code, or stylesheets.

- Failed to load resource: the server responded with a status of 404 ()
   Failed to load resource: the server hero tiny.e20b9d67.png:1 responded with a status of 404 ()
   Failed to load resource: the cloud 1 tiny.7b7bedfe.png:1 server responded with a status of 404 ()
   Failed to load resource: the cloud 3 tiny.58994354.png:1 server responded with a status of 404 ()
   Failed to load resource: the roadmap done.f24de586.svg:1 server responded with a status of 404 ()
   Failed to load resource: the roadmap done.f24de586.svg:1 server responded with a status of 404 ()
   Failed to load resource: the roadmap done.f24de586.svg:1 server responded with a status of 404 ()
- Failed to load resource: the server responded with a status of 404 ()



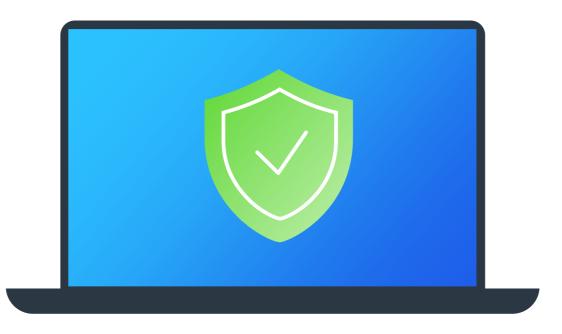


#### Website Part 4 (GWS) – General Web Security





#### **Social Media**



We were able to locate Social Media networks for the project.

All links have been conveniently placed below.





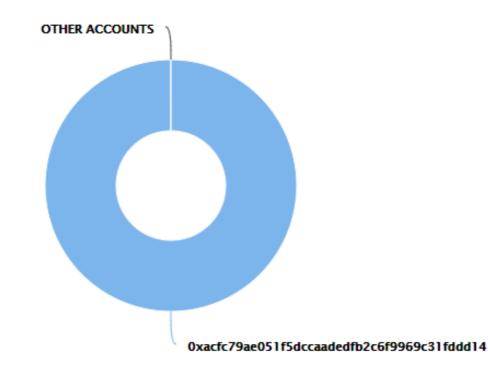
#### **Top Token Holders**

The entire supply was in one wallet at the time of audit. We expect this to change as the project goes through initial distribution phases. Please use the link below to view the most up-to-date holder information.

Click here to view the most up-to-date list of holders

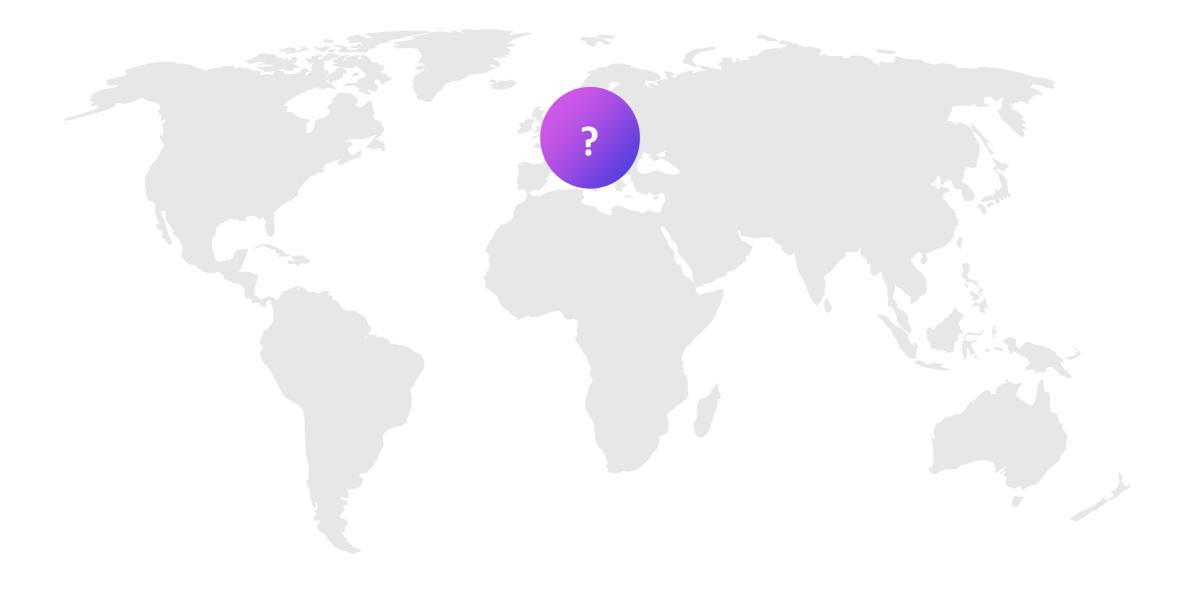
#### CryptoCowboys Top 100 Token Holders

Source: BscScan.com

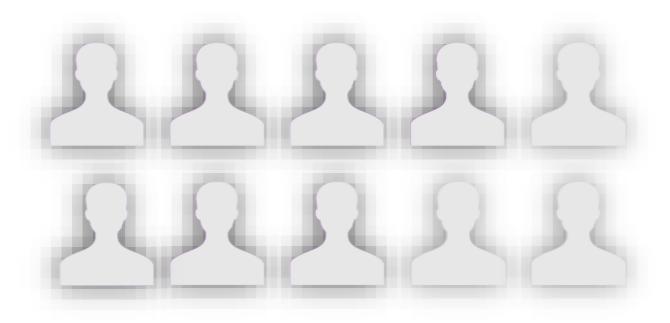


#### **Location Audit**

We were unable to identify a primary location for the project at this time or a location has not been declared.



#### **Team Overview**



We are unable to find any information about the team on the website at this time. Projects may choose to stay anonymous for a myriad of reasons.

#### Roadmap

A roadmap was found on the official website, we have conveniently placed it on this page for your viewing.



ROADMAP

#### Disclaimer



The opinions expressed in this document are for general informational purposes only and are not intended to provide specific advice or recommendations for any individual or on any specific investment. It is only intended to provide education and public knowledge regarding BSC projects. This audit is only applied to the type of auditing specified in this report and the scope of given in the results. Other unknown security vulnerabilities are beyond responsibility. Dessert Finance only issues this report based on the attacks or vulnerabilities that already existed or occurred before the issuance of this report. For the emergence of new attacks or vulnerabilities that exist or occur in the future, Dessert Finance lacks the capability to judge its possible impact on the security status of smart contracts, thus taking no responsibility for them. The smart contract analysis and other contents of this report are based solely on the documents and materials that the contract provider has provided to Dessert Finance or was publicly available before the issuance of this report (issuance of report recorded via block number on cover page), if the documents and materials provided by the contract provider are missing, tampered, deleted, concealed or reflected in a situation that is inconsistent with the actual situation, or if the documents and materials provided are changed after the issuance of this report, Dessert Finance assumes no responsibility for the resulting loss or adverse effects. Due to the technical limitations of any organization, this report conducted by Dessert Finance still has the possibility that the entire risk cannot be completely detected. Dessert Finance disclaims any liability for the resulting losses.

Dessert Finance provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Even projects with a low risk score have been known to pull liquidity, sell all team tokens, or exit-scam. Please exercise caution when dealing with any cryptocurrency related platforms.

The final interpretation of this statement belongs to Dessert Finance.

Dessert Finance highly advises against using cryptocurrencies as speculative investments and they should be used solely for the utility they aim to provide.

# 

# Thank You

DESSERT FINANCE PROJECT AUDIT HAS BEEN COMPLETED FOR CRYPTOCOWBOYS (CCS) 1 DSRT HAS BEEN SENT TO AUDITED PROJECT'S CONTRACT ADDRESS FOR VERIFICATION OF THIS AUDIT AT BLOCK NUMBER: **11083561** 

> www.dessertswap.finance https://t.me/dessertswap