



TokenizeArt

Build your own NFT

Summary: This document is a Web3 related exercise.

Version: 1.00

Contents

I	Preamble	2
II	Introduction	3
III	Objectives	4
IV	Mandatory part	5
IV.1	Creating the image of your NFT	5
IV.2	Deploying your contract	6
IV.3	Mint your NFT	7
V	Bonus part	8
VI	Submission and peer-evaluation	9

Chapter I

Preamble

This subject is the production of a partnership between 42 and [BNB Chain](#).

Build N Build (BNB) Chain is a distributed blockchain network upon which developers and innovators can build decentralized applications (DApps) as part of the move to Web3.

As of October 2022, [BNB Chain](#) is the world's largest smart-contract blockchain in terms of transaction volume and daily active users. At the time of writing, it has processed 3 billion transactions from 232 million unique addresses, and has an ecosystem of more than 1,500 active DApps. The decentralized nature of the network means anyone can build a product on BNB Chain without having to ask for permission, and potentially reach a massive audience.

You can retrieve Tbnb for free and with no minimum coins on your wallet via this faucet: [BNB Chain Faucet](#)



BNB CHAIN

Figure I.1: <https://www.bnchain.org/>

Chapter II

Introduction

Welcome to the exciting world of blockchain technology!

Have you ever dreamed of creating your own digital non-fungible token?

Now is your chance to turn that dream a reality.

Blockchain technology allows for the creation and distribution of unique digital assets, known as tokens. These tokens can represent a wide range of things, from a simple representation of currency to more complex assets like artwork or even a real-world asset. The process of creating your own token is not without its challenges, but with the right knowledge and resources, it can be a rewarding and fulfilling experience.

So, why wait?

Start your journey towards creating your very own non-fungible token on the blockchain today!



Chapter III

Objectives

As a participant in this project, you will have the opportunity to contribute to the creation of a digital asset on the blockchain. This project is designed to challenge you in several areas, including your ability to become proficient in multiple programming languages and your familiarity with public blockchain technology.

While a strong background in cryptography is not required for this project, you should be prepared to learn and adapt as you work towards creating your own digital asset. This project will require you to think critically and creatively, as well as to push yourself out of your comfort zone as you navigate the complexities of blockchain technology.

Ultimately, your participation in this project will not only help you develop valuable skills and knowledge, but it will also allow you to be part of something truly innovative and exciting. Are you ready to take on the challenge?

Let us begin!

Chapter IV

Mandatory part

IV.1 Creating the image of your NFT

In order to create a non-fungible token, there are several technical requirements that must be met.



You are free to choose the representation of your non-fungible token. Your only constraint is to include the number 42 in it. It is of course forbidden to use insulting terms or images under penalty of punishment.

For example, this image is not good as the 42 is incorrectly displayed, while the other images in the topic are correct:



Your image must be stored using distributed registry technology (IPFS, for example)

IV.2 Deploying your contract

You must create a `README.md` file at the root of your repository explaining the choices you made and the reasons behind them.



The language used is of course free, but you must respect the standards of the blockchain you are going to use (for example ERC721 for ETH, or BEP-721 for BSC).

First and foremost, you will need to choose a blockchain platform that supports the creation of non-fungible token. There are many different options to choose from, each with its own unique features and capabilities.



You must also manage the metadata for your NFT (the artist's name must be your login and the name must include 42 and a title).

Once you have selected a platform, you will need to become proficient in the programming language used by that platform in order to develop your non-fungible token. Different platforms use different programming languages, so you will need to ensure that you have the necessary skills to work with the language of your chosen platform such as [IDE](#), [Truffle](#), [Remix](#) or [Hardhat](#).



Make sure you understand what you are doing. You will never be asked to use real money or your coins for this project. There are test chains to avoid this problem, such as the BSC Testnet chain

You must submit the code used to create your non-fungible token in a `code` folder located at the root of your repository. You should be careful to comment your code and to use readable and explicit variable/function names.



During your evaluation there will be a code review.

You must be very careful about how you demonstrate the operation of your non-fungible token. You must be able to perform minimalist actions to demonstrate its operation. You need to think about all aspects of security such as ownership or privileges.

IV.3 Mint your NFT

You should also place all the necessary files for the deployment of your non-fungible token in a second folder with a name of your choice

After you have minted your non-fungible token on a public blockchain, please mention the public address and the network used in your **Git** repository. You should be able to display your NFT.



you need to be able to confirm the owner of an NFT, for example using the `ownerOf` function in Solidity.

Finally, you should have a folder containing the documentation for this project. This folder, named **documentation**, should be located at the root of your repository. It should be possible to understand how it works and what is needed to use your non-fungible token.

You will need to have a clear understanding of how your NFT will be used and what it will represent. This may require the development of a whitepaper or other documentation outlining the features and functionality of your non-fungible token.



You must take the time to make a clear and explicit documentation. This will be reviewed during your evaluation.

Consider also creating a demo video to showcase your NFT and its features to potential users and investors.



If you want to make a video demo you do not have to push the video to your repository; a simple link will suffice! Creating a demo video is not required. You will not get a better grade by creating this video.

Below is an example of the expected directory structure:

```
$> ls -al
total XX
drwxrwxr-x 3 eagle eagle 4096 avril 42 20:42 .
drwxrwxrwt 17 eagle eagle 4096 avril 42 20:42 ..
-rw-rw-r-- 1 eagle eagle XXXX avril 42 20:42 README.md
drwxrwxr-x 3 eagle eagle 4096 avril 42 20:42 code
drwxrwxr-x 3 eagle eagle 4096 avril 42 20:42 deployment
drwxrwxr-x 3 eagle eagle 4096 avril 42 20:42 mint
drwxrwxr-x 3 eagle eagle 4096 avril 42 20:42 documentation
```

Chapter V

Bonus part

Here are some bonuses that could be very useful :

- A beautiful NFT
- A website where you can mint your NFT with a graphical interface
- You need to manage your NFT Inscriptions, i.e. store your metadata and image storage directly on-chain



The bonus part will only be assessed if the mandatory part is PERFECT. Perfect means the mandatory part has been fully completed and works flawlessly. If you have not passed ALL the mandatory requirements, your bonus part will not be evaluated at all.

Chapter VI

Submission and peer-evaluation

Turn in your assignment in your **Git** repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.

Exceptionally for this project, we recommend that you share your project via your personal git account when your project is valid. Feel free to use different hashtags depending on the programming language used, but also web3 etc...

