



# HOW TO **Create an NFT Marketplace**

## How to Create an NFT Marketplace

With NFT (non-fungible token) marketplaces, creators get to display their NFT art to the masses. Moreover, by listing an NFT to a marketplace, tokens receive a lot of exposure since marketplaces for NFTs drive massive amounts of traffic due to NFTs' increase in popularity. As such, NFT creators don't need to worry too much about auctioning and getting their NFT tokens sold. However, there are not just crypto enthusiasts that are interested in NFTs, but also large corporations. As such, now is an excellent time to start learning [NFT token development](#). Specifically, knowing how to create an NFT marketplace is a highly sought-after skill that every programmer should know how to do. Therefore, we'll explore in this article the

development process of how to create an NFT marketplace with Moralis and its ultimate [Web3](#) development platform.

Since we are still early in the game regarding NFTs and their use cases, who knows, any developer aspiring to [become a blockchain developer](#) might be the one creating the next big NFT marketplace that the masses will adopt. Of course, there will always be major NFT marketplaces, but it makes sense for every NFT launch or collection to have its own NFT marketplace. Moreover, you might get the next opportunity to develop a marketplace devoted to NFTs for a large blockchain company. As such, being able to create an NFT marketplace is a skill that could future-proof your career, and with the help of [Moralis](#), it is a skill you can easily master.



## What is an NFT Marketplace?

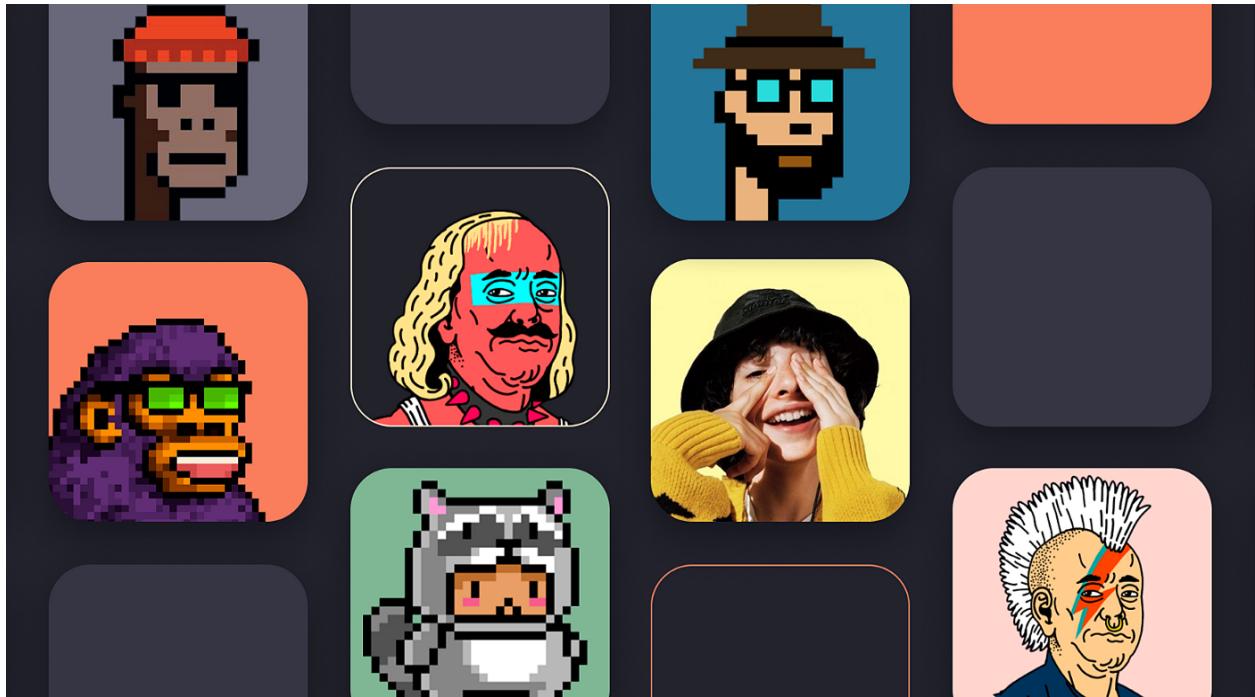
Before we take on an example project that will teach you how to create an NFT marketplace with minimal effort and in record time, we need to make sure you know all the basics. An NFT marketplace is essentially any Web3 application that revolves around selling and buying any kind of non-fungible token (NFT). As such, there can be many kinds of NFT marketplaces such as one open to the general public, where anyone with a crypto wallet can buy and list their own NFTs. On the other hand, there are also NFT marketplaces that aren't open to the public. Specifically, these only let a particular company or brand sell their NFTs, and do not allow users to list their NFTs.

In addition, using NFTs metadata, you can set different types of functions in place to provide specific functionality.

In case you are not yet familiar with NFTs, we recommend checking out some of our past articles addressing topics such as “[what are NFTs?](#)”, “[how to mint NFTs](#)”, and taking a closer look at the two most popular NFT token standards – [ERC-721 token standard](#) and [ERC-1155 token standard](#).

## **Why You Should Learn to Create an NFT Marketplace**

There are countless opportunities regarding NFTs and their use cases, such as [NFT games](#), collectibles, NFT artwork, etc. However, while using existing NFT marketplaces may seem like a good idea, most of them come with some sorts of limitations. In most cases, they are not truly decentralized, and their cross-chain interoperability tends to be extremely poor. Furthermore, the majority of NFT marketplaces still limit users to the [Ethereum](#) blockchain, and they tend to impose their own fees. These are just some of the major downsides to some marketplaces. But, is there a solution to overcome these downsides? Yes – create your own NFT marketplace. However, don't be discouraged if you have no audience, and you'll end up investing a lot in marketing to make it the next major marketplace; since we are so early in the game, anything is possible!



You can easily offer to [create an NFT](#) marketplace for countless other businesses, artists, and celebrities looking to launch their NFTs. Of course, you can already help them [create NFTs](#); however, it makes sense, not just for them, but for all wanting to sell their NFTs to have their own NFT marketplace. Why? Well, you are in control of the marketplace and you design the UI (user interface) to certain specifications. Moreover, you're not susceptible to platform manipulation, and you get to cut out possible intermediaries charging you fees on top of the already expensive gas fees, etc.

In addition, you can decide and implement specific features of the dApp. That way, you can provide functionalities that other platforms don't offer. As such, offering users an NFT marketplace and ensuring they get to solve all their needs without leaving that particular platform makes a lot of sense. Therefore, knowing how to create an NFT marketplace presents virtually endless opportunities for developers.

## How to Create an NFT Marketplace – Example Project

Here at Moralis, we believe in learning by taking immediate action. Therefore, we'll take on a simple example project herein that you can follow along with. As such, we will create an NFT marketplace in the following sections. Of course, we will be using [Moralis' SDK](#), including its ultimate [NFT API](#), to complete our task with maximum ease.

If you prefer video format, you can watch our full tutorial video on our Youtube channel by [clicking here](#).

### A Preview of Your NFT Marketplace

Before we start with the coding part, we want you to have a clear picture of what we will build. Looking at the image below, you can see that our example NFT marketplace will offer users multiple options and features, including:

- Display the user's native token balance at the top.
- An “On Sale” section, which shows the current available NFTs in our marketplace.
- The “Buy/Sell” section displays NFTs that the user decides to take action on (buy or sell).
- The “Your NFTs” section will display collections that the logged-in user holds.

Your Market Place Balance is 0.04 ETH

On Sale

Buy/Sell

Your NFTs



The Rogue  
A thief, a rogue, a loner.

Select



The Elf  
Lady of the forest.

Select



The Warrior

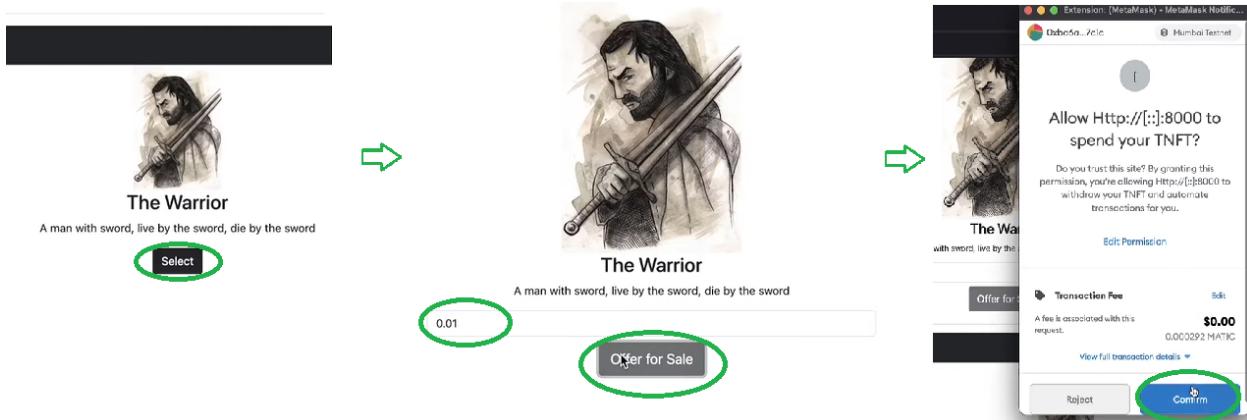
A man with sword, live by the sword, die by the sword

Select

There are currently no NFTs in the “On Sale” and “Buy/Sell” section because there is no NFT that we or any other user on our example NFT marketplace is selling.

### Our Example NFT Marketplace – Listing an NFT

If we select one of our NFTs, it will then be available in the “Buy/Sell” section. Next, we have an option to enter the price and offer the NFT in question for sale by clicking the “Offer for Sale” button. Then, we need to approve the transaction by clicking the “Confirm” button in the [MetaMask](#) pop-up window. This window will prompt automatically.



Once we confirm the transaction and it processes, we will be able to see that NFT in the “On Sale” section.

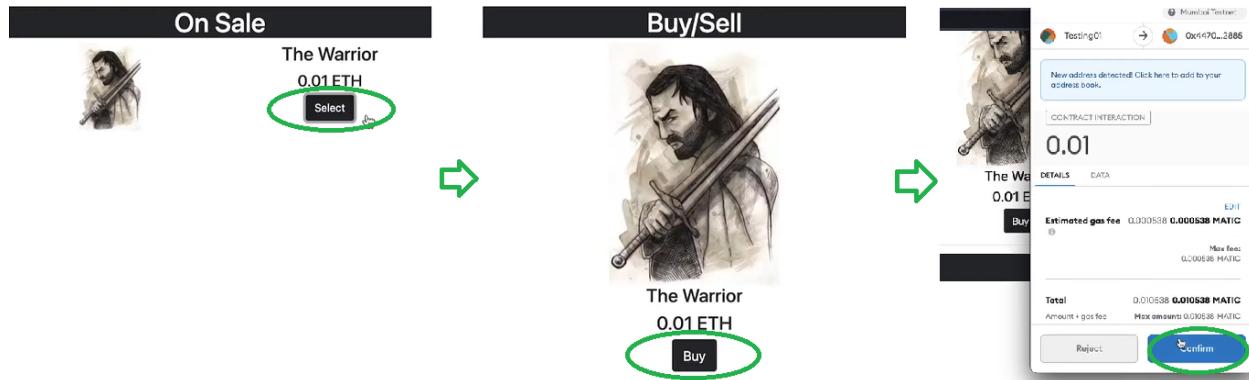
Your Market Place Balance is 0.04 ETH

On Sale	Buy/Sell
<p><b>The Warrior</b></p> <p>0.01 ETH</p> <p>Select</p>	<p><b>The Warrior</b></p> <p>A man with sword, live by the sword, die by the sword</p> <p>Approval transaction 0x1b4aa6b72821935d495024d5e10a0ab0992e6c1d2dbbf9d41a9dc7a9bec43cdd</p>

### Our Example NFT Marketplace – Buying an NFT

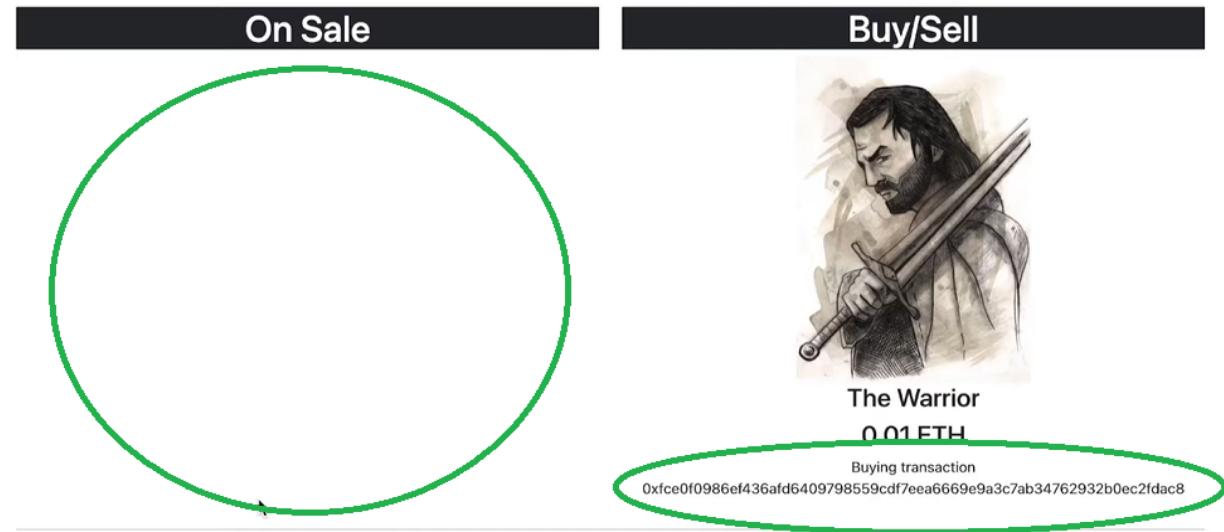
Of course, we cannot buy our own NFT; however, if another account is logged in, it can purchase that NFT. As such, in the image below, you can see how another account will purchase the NFT we've listed above. So, it first clicks on the “Select” button next to the NFT in the “On Sale” section. It then clicks on the “Buy” button for the NFT that's

now available in the “Buy/Sell” section. By clicking this button, MetaMask will initiate to confirm the transaction.



After you purchase the NFT, it is no longer in the “On Sale” section. Instead, it shows in the “Your NFTs” section for the account (keep in mind that it may take a while for the transaction to complete):

Your Market Place Balance is 0.03 ETH



Once the transaction is complete, the new owner can sell that NFT.

## How to Create an NFT Marketplace with Moralis

Now that we know what we will build, it is time to list the backend features and components required to make the functionalities displayed in line with what we've covered in the previous section.

If you prefer video format, you can watch our full tutorial video on our Youtube channel by [clicking here](#).

These are as follows:

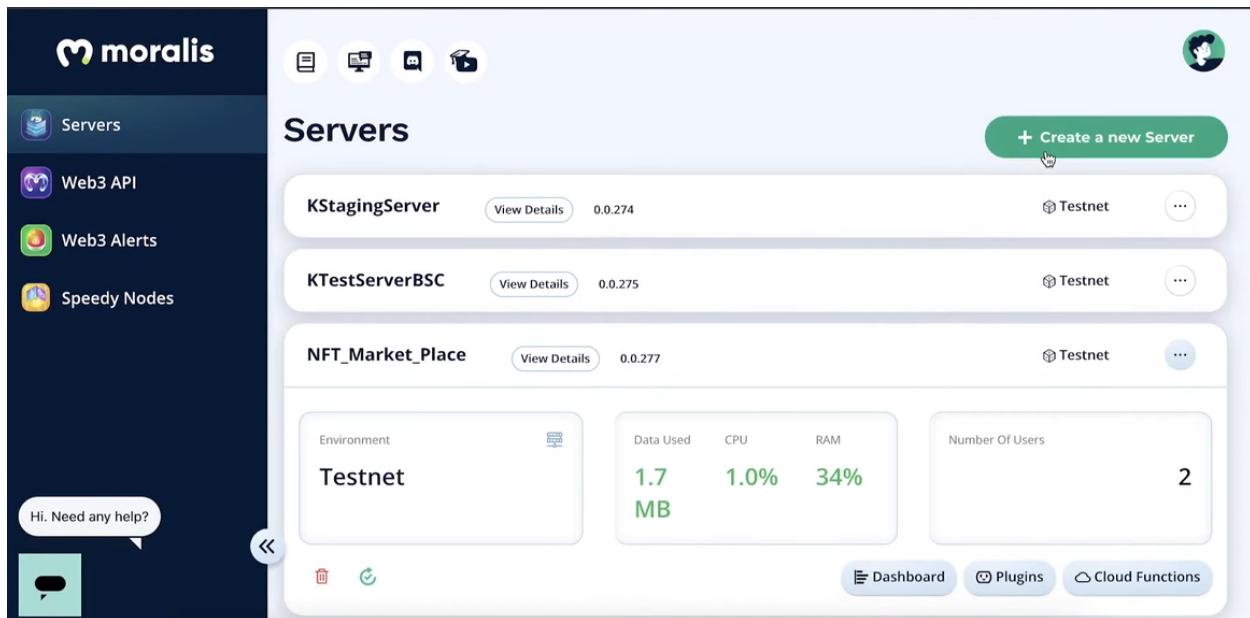
- User authentication that allows our users to sign in to our dApp.
- We need to be able to track the balances of our users in real-time.
- We need to be able to track smart contract events in our dApp.

*We rely on [smart contracts](#) that are operators of our marketplace and other smart contracts, including those used to mint NFTs.*

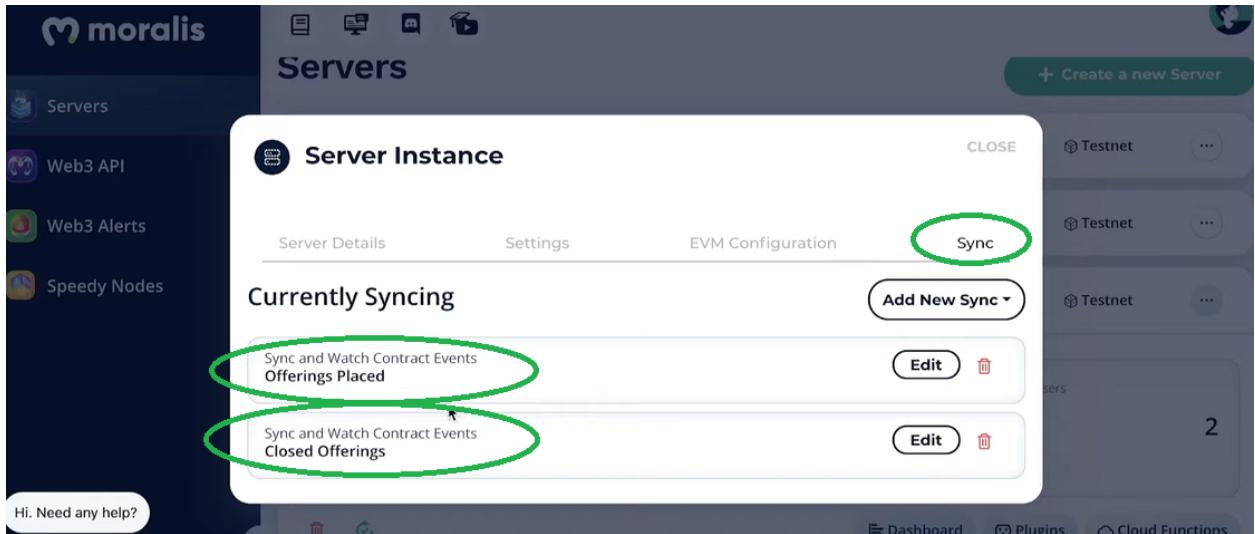
Implementing all of the above backend functionalities would be extremely challenging if we were to take on this quest by building directly on top of an [RPC node](#). However, when we use Moralis, we get to cover the entire backend of our dApp by simply copying and pasting short snippets of code from [Moralis' documentation](#). As such, Moralis makes it straightforward to create an NFT marketplace. Moreover, you save a lot of time, which means you get to focus entirely on the frontend and deliver the best user experience.

# Moralis – The Ultimate Shortcut to Create an NFT Marketplace

When you [create your free Moralis account](#), you get to [create a Moralis server](#), which is the starting point in order for you to [build dApps](#) with Moralis. Once you have an active server, you get to utilize the [Ethereum API](#) among all of its other tools.



Moreover, Moralis servers offer many additional options, including creating sync events. For the servers used for our example, we will create two listeners to detect specific smart contract events. One listener for detecting when a new offer is placed (selling) and another one to detect when an offer is closed (buying):



Furthermore, Moralis also provides you with a complete database containing live details regarding your users, which is available within your Moralis dashboard. These include token balances, transaction history, pending transactions, NFTs, and much more. This is also where the data regarding our listeners (underlined with green lines) is available:

objectId	String	createdAt	Date	updatedAt	Date	ACL	ACL
0xb04195e45174d5a...		18 Oct 2021 at 01...		18 Oct 2021 at 01...		GYoNvddAXysySBoxK...	
0xba6aada3f435212...		12 Oct 2021 at 01...		12 Oct 2021 at 01...		0t7Yt48loYS4j84g0...	

**Core**

Browser	Create a class
Role	0
Session	2
User	2
<u>ClosedOfferings</u>	8
EventSync	2
PlacedOfferings	8
PolygonBalance	2
PolygonBalancePending	0
PolygonNFTOwners	4
PolygonNFTOwnersPending	0
PolygonNFTTransfers	12
PolygonTokenBalance	3
PolygonTokenTransfers	21
PolygonTransactions	82

Now that we've got the on-chain data at our disposal, we only need code snippets to use that data correctly.

## How to Create an NFT Marketplace – Code Overview

There are two main aspects to the code required to create an NFT marketplace. For one, we need to create a dApp that will serve as an authorized contract operator of the marketplace. At the same time, we need to use a smart contract that will ensure all transactions happens properly. Let's first look at the latter.

*Note: The entire code of our “[MarketPlace](#)” contract is available on [GitHub](#).*

The gist of the smart contract used for our example can be seen by looking at the events it covers:

```
contract MarketPlace{  
  
    event OfferingPlaced(bytes32 indexed  
offeringId, address indexed hostContract, address  
indexed offerer, uint tokenId, uint price, string  
uri);  
    event OfferingClosed(bytes32 indexed  
offeringId, address indexed buyer);  
    event BalanceWithdrawn (address indexed  
beneficiary, uint amount);  
    event OperatorChanged (address  
previousOperator, address newOperator);
```

The rest of the smart contract code clearly defines specific functions that ensure the above four events are properly covered. That way, users can only sell NFTs in their possession, while buyers can only purchase listed NFTs. Moreover, the code ensures that the price gets attached to the token in question.

## How to Create an NFT Marketplace – The dApp Part

This is the part that probably interests you the most. Here's where we ensure that the users get to see and use our NFT marketplace. For the purpose of our example project, we created three files: “[logic.js](#)”, “[cloudCode.js](#)”, and “[index.html](#)”. The complete code for each file is also available on [GitHub](#). We ensure the main functionality of our dApp with the “*logic.js*” file. As such, it deserves some extra attention. This is also where the full power of Moralis gets to be put to action. Moreover, to initialize Moralis, you need to paste your Moralis server URL and application ID:

```
Moralis.initialize(""); // Application ID from  
moralis.io  
Moralis.serverURL = ""; // Server URL from  
moralis.io
```

Moving on, by looking at the lines of code below, you can see the main purpose of our dApp. “`Moralis.authenticate()`” makes sure that proper functions are executed upon authenticating users:

```
Moralis.authenticate().then(function() {  
    populateNFTs();
```

```
    populateOfferings();
    populateBalance();
    subscribeOfferings();
    subscribeBuys();
    subscribeUpdateNFTs();
}) ;
```

The remaining code ensures that the “populate” and “subscribe” functions obtain the proper data. Moreover, this means sure that everything gets neatly displayed for our users. Moralis does all heavy backend lifting through the use of code snippets that obtains data from your “moralisDashboard” database. These snippets of code include:

- *Moralis.Query(“PlacedOfferings”)*
- *Moralis.Query(“ClosedOfferings”)*
- *Moralis.Query(“PolygonNFTOwners”)*
- *Moralis.Cloud.run(“getBalance”, params)*
- *Moralis.Units.FromWei(presentBalance)*
- *Moralis.Cloud.run(“placeOffering”, params)*
- *Moralis.Units.ETH(offering.price)*



### **Ensuring Proper Security of an NFT Marketplace**

Of all the code snippets above, the “`Moralis.Cloud.run(“placeOffering”, params)`” bears an extra significance. It enables us to ensure that our dApp signs the transactions behind the curtains and protects our dApp’s private key. This is also where the “`cloudCode.js`” file comes to play. Inside it, we add the private key that corresponds to the specific Ethereum address that you are using.

*In case you want a walkthrough to create an NFT marketplace, please watch the video below. Feel free to skip ahead to 6:35, where the code walkthrough starts.*

## **How to Create an NFT Marketplace – Summary**

By covering the details of this article, everyone interested in learning to create an NFT marketplace is on the right track. As things currently stand, this seems to be a skill that can future-proof your position as a [blockchain developer](#). Fortunately, with Moralis and MetaMask, the creation of an NFT marketplace is rather simple. Of course, you still need to do some coding; however, your [JavaScript](#) proficiency is enough to cover the frontend. As far as the backend goes, Moralis has your back. In addition, you can further speed up your process by making use of open-source platforms such as OpenZeppelin, Chakra, Bootstrap, etc.

Make sure to checkout the full code on our Github. If you’re interested in building more dApps like wallets, portfolio trackers, NFT games, make sure to check out this [Ethereum Boilerplate](#) in React. It has all important components and hooks for building dApps with React.

