MINOR PROJECT REPORT

ON

NFT. Marketplace

Master of Computer Application

(Batch:2021-2023)

Submitted to

Dr Manjot Kaur Bhatia

Submitted By:

Name: Jayant marwaha

Enrolment No:01550404421

Name: Mrinal Narang

Enrolment No:03650404421

Name: Ritesh Sandilya

Enrolment No: 03850404421

Jagan Institute of Management Studies

(Affiliated to Guru Gobind Singh Indraprastha University)

SR NO.	TOPIC	PAGE NO.	RERMARKS
1	INTRODUCTION	3-5	
2	INTRODUCTION OF TOOLS	6-15	
3	OBJECTIVE OF THE SYSTEM	16-17	
4	REQUIREMENT ANALYSIS	18-21	
5	DETAILED SYSTEMSPECIFICTION	21	
6	USE CASE DIAGRAM	22	
7	DATA FLOW DIAGRAM	23-24	
8	ACTIVITY DIAGRAM	25-26	
9	DATABASE DESIGN	27-28	
10	SOME IMPORTANT MODULES	29-34	
11	SOME IMPORTANT CODES	35-36	
12	SYSTREM IMPLEMENTATION	37-38	
13	TESTING	39-42	
14	RESULT AND CONCLUSION	43	
15	FUTURE ENHANCEMENT	40	
16	BIBLIOGRAPHY	41	

INTRODUCTION

Non-Fungible Tokens (NFTs) is entirely a novel concept in the cryptocurrency and blockchain field, and this concept is soaring every day to reach greater heights. In the real world, there are creators who keep their masterpieces inside a safety locker and fear that the ownership of their work will be snatched someday. This is the point where NFTs make their entry. It proves your ownership worldwide, and no fraudulent activities can be involved. This attracted a lot of investors and crypto lovers to kick-start their own NFT marketplaces. To enhance the visibility of this marketplace, certain NFT marketing strategies have to be carried out.

What Is An NFT?

NFTs are tokens that are greatly different from regular cryptocurrencies. They hold unique data in each NFTs, making it non-interchangeable.

Say, one BTC is equivalent to another BTC. But the same is not the case in NFTs. The value in the tokens could not be replaced, replicated or even stolen.

Example:

arts, intellectual properties, virtual properties, financial instruments etc

What is NFT Marketplace

NFT marketplaces play a crucial role in bridging the gap between buyers and sellers. In some cases, NFT marketplaces could also offer additional tools for creating NFTs within few minutes.

The specialized marketplaces allow artists to put up their NFT artworks for sale. Buyers could browse the marketplace for NFTs and purchase the item of their choice through bidding. Therefore, any NFT developer or enthusiast must go through the NFT marketplace list to ensure profitable deals on the artwork, collectibles, and other digital assets.

Why NFTs Are Important

Non-fungible tokens are an evolution of the relatively simple concept of cryptocurrencies. Modern finance systems consist of sophisticated trading and loan systems for different asset types, ranging from real estate to lending contracts to artwork. By enabling digital representations of physical assets, NFTs are a step forward in the reinvention of this infrastructure.

To be sure, the idea of digital representations of physical assets is not novel nor is the use of unique identification. However, when these concepts are combined with the benefits of a tamper-resistant blockchain of smart contracts, they become a potent force for change.

Perhaps, the most obvious benefit of NFTs is market efficiency. The conversion of a physical asset into a digital one streamlines processes and removes intermediaries. NFTs representing digital or physical artwork on a blockchain remove the need for agents and allow artists to connect directly with their audiences. They can also improve business processes. For example, an NFT for a wine bottle will make it easier for different actors in a supply chain to interact with it and help track its provenance, production, and sale through the entire process. Consulting firm Ernst & Young has already developed such a solution for one of its clients.

Non-fungible tokens are also excellent for identity management. Consider the case of physical passports that need to be produced at every entry and exit point. By converting individual passports into NFTs, each with its own unique identifying characteristics, it is possible to streamline the entry and exit processes for jurisdictions. Expanding this use case, NFTs can serve an identity management purpose within the digital realm as well.

HOW NFT'S WORK

.

- The majority of NFTs reside on the Ethereum cryptocurrency's blockchain, a distributed public ledger that records transactions.
- NFTs are individual tokens with valuable information stored in them.
- Because they hold a value primarily set by the market and demand, they can be bought and sold just like other physical types of art.
- NFTs' unique data makes it easy to verify and validate their ownership and the transfer of tokens between owners.

Examples of NFT

The NFT world is relatively new to people. Here are some examples of NFTs that exist today:

- A Digital Collectible
- Domain Names
- Games
- Essays
- Sneakers in fashion line

INTODUCTION OF TOOLS

1.Thirdweb

What is Web 3.0 (Web3)?

Web 3.0 (Web3) is the third generation of the evolution of web technologies. The web, also known as the World Wide Web, is the foundational layer for how the internet is used, providing website and application services.

Web 3.0 is still evolving and being defined, and as such, there isn't a canonical, universally accepted definition. What is clear, though, is that Web 3.0 will have a strong emphasis on decentralized applications and make extensive use of blockchain-based technologies. Web 3.0 will also make use of machine learning and artificial intelligence (AI) to help empower more intelligent and adaptive applications.

Another aspect that is part of the emerging definition of Web 3.0 is the notion of a semantic web. Among those that have advocated for the integration of semantic technology into the web is the creator of the web, Tim Berners-Lee.

It took over 10 years to transition from the original web, Web 1.0, to Web 2.0, and it is expected to take just as long, if not longer, to fully implement and reshape the web with Web 3.0.

If the trend of change is traced from Web 1.0, a static information provider where people read websites but rarely interacted with them, to Web 2.0, an interactive and social web enabling collaboration between users, then it can be assumed that Web 3.0 will change both how websites are made and how people interact with them.

How does Web 3.0 work?

With Web 1.0 and Web 2.0 technologies, Hypertext Markup Language (HTML) defines the layout and delivery of webpages. HTML will continue to be a foundational layer with Web 3.0, but how it connects to data sources and where those data sources reside could be somewhat different than earlier generations of the web.

Many websites and nearly all applications in the Web 2.0 era rely on some form of centralized database to deliver data and help to enable functionality. With Web 3.0, instead of a centralized database, applications and services make use of a decentralized blockchain. With blockchain, the basic idea is that there isn't an arbitrary central authority, but rather a form of distributed consensus.

An emerging governance ideal within the blockchain and Web 3.0 community is the concept of a decentralized autonomous organization (DAO). Instead of having a central authority that governs the operations of a platform, with a DAO, Web 3.0 technologies and communities provide a form of self-governance in an attempted decentralized approach.

Web 3.0 also fundamentally works with cryptocurrency, more so than with fiat currency. Finance and the ability to pay for goods and services with a decentralized form of payment is enabled across Web 3.0 with the use of cryptocurrencies, which are all built and enabled on top of blockchain technology.

Both Web 1.0 and Web 2.0 were primarily built with the IPv4 addressing space. As a function of a massive growth of the web over the decades, there is a need in Web 3.0 for more internet addresses, which is what IPv6 provides.

Key Web 3.0 features

Web 3.0 may be constructed with AI, semantic web and ubiquitous properties in mind. The idea behind using AI comes from the goal of providing faster, more relevant data to end users. A website using AI should be able to filter through and provide the data it thinks a specific user will find appropriate. Social bookmarking as a search engine can provide better results than Google since the results are websites that have been voted on by users. However, these results can also be manipulated by humans. AI could be used to separate the legitimate results from the falsified, therefore producing results similar to social bookmarking and social media but without bad feedback.

An artificially intelligent web will also introduce virtual assistants, an element that is already emerging today as an aspect built into a device or through third-party apps.

The idea behind the semantic web is to categorize and store information in a way that helps teach a system what specific data means. In other words, a website should be able to understand words put in search queries the same way a human would, enabling it to generate and share better content. This system will also use AI; the semantic web will teach a computer what the data means, and then AI will take the information and use it.

There are several key Web 3.0 features that help to define what the third generation of the web will likely be all about, including the following:

- **Decentralized.** As opposed to the first two generations of the web, where governance and applications were largely centralized, Web 3.0 will be decentralized. Applications and services will be enabled in a distributed approach, where there isn't a central authority.
- **Blockchain-based.** Blockchain is the enabler for the creation of decentralized applications and services. With blockchain, the data and connection across services are distributed in an approach that is different than centralized database infrastructure. Blockchain can also enable an immutable ledger of transactions and activity, helping to provide verifiable authenticity in a decentralized world.
- **Cryptocurrency-enabled.** Cryptocurrency usage is a key feature of Web 3.0 services and largely replaces the use of fiat currency.
- **Autonomous and artificially intelligent.** More automation overall is a critical feature of Web 3.0, and that automation will largely be powered by AI.

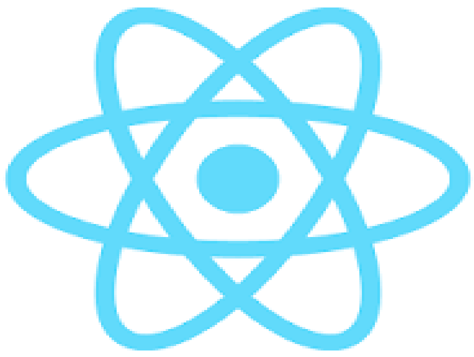
Web 3.0 applications

With blockchain at the foundation, Web 3.0 enables a growing number of different types of new applications and services to exist, including the following:

- **NFT.** Nonfungible tokens (NFTs) are tokens that are stored in a blockchain with a cryptographic hash, making the token unit unique.
- **DeFi.** Decentralized finance (DeFi) is an emerging use case for Web 3.0 where decentralized blockchain is used as the basis for enabling financial services, outside of the confines of a traditional centralized banking infrastructure.
- **Cryptocurrency.** Cryptocurrencies like Bitcoin are Web 3.0 applications that create a new world of currency that aims to be separate from the historical world of fiat currency.

- **dApp.** Decentralized applications (dApps) are applications that are built on top of blockchain and make use of smart contracts to enable service delivery in a programmatic approach that is logged in an immutable ledger.
- **Cross-chain bridges.** There are multiple blockchains in the Web 3.0 world, and enabling a degree of interoperability across them is the domain of cross-chain bridges.
- **DAOs.** DAOs are set to potentially become the organizing entities for Web 3.0 services, providing some structure and governance in a decentralized approach.

2.React



React (also known as **React.js** or **ReactJS**) is a free and open-source front-end JavaScript library[3] for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.[4][5][6] React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next.js. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

What Are The Major Features of React?

Before we look at why it is so popular, let us check out the **React JS features** properly. This will help us to understand how React JS works.

1. Virtual DOM

This characteristic of React helps to speed up the app development process and offers flexibility. The algorithm facilitates the replication of a web page in React's virtual memory.

The original DOM is thereby represented by a virtual DOM.

Whenever the app is modified or updated, the entire UI is rendered again by the virtual DOM, by updating the components that have been modified. This reduces the time and cost taken for development.

2. JavaScript XML or JSX

It is a markup syntax that describes the appearance of the interface of the app. It makes the syntax just like HTML and is used to create React components by developers.

JSX is one of the best features of React JS as it makes it super easy for developers to write the building blocks.

3. React Native

Uses native, rather than web components to facilitate native React JS development for Android and iOS.

Basically, this feature transforms React code to render it compatible with iOS or Android platforms and provides access to their native features.

4. One-Way Data Binding

This means that React uses a flow of data that is unidirectional, forcing developers to use the call-back feature to edit components, and preventing them from editing them directly.

The controlling of data flow from a single point is achieved with a JS app architecture component called Flux. It actually affords developers better control over the app and makes it more flexible and effective.

5. Declarative UI

This feature makes React code more readable and easier to fix bugs. React JS is the best platform to develop UIs that are both exciting and engaging not just for web apps, but mobile apps as well.

6. Component Based Architecture

This simply means that the user interface of an app based on React JS is made up of several components, with each of them having its particular logic, written in JS.

Due to this, developers can relay the data across the app without the DOM being impacted. React JS components play a huge part in deciding the app visuals and interactions.

3. Sanity

Sanity.io is the platform for structured content. With Sanity.io you can manage your text, images, and other media with APIs. You can also use the open-source single page application Sanity Studio to quickly set up an editing environment that you can customize. With Sanity.io you have access to a bunch of APIs, libraries, and tooling that helps you leverage the benefits of having all your content available as a single source of truth. This article will quickly walk you through some central concepts, giving you a head start.

Before we dive in, there are three important concepts to bear in mind:

- Sanity.io has a real-time datastore for structured content, and supporting APIs for assets, user management, and more.
- Sanity Studio is a user interface for managing content. It's an open-source React Single Page Application that you can customize and host wherever you want.
- There are also SDKs, libraries, and tools that let you query your content and integrate it with websites, services, and other applications; wherever you need content.

Sanity Datastore

Querying for content

When you start a new project on Sanity.io, you’ll get access to the real-time datastore. This is a schemaless backend that lets you store and query JSON documents, and subscribe to real-time changes. It comes with a query API that uses the query language GROQ to lets you quickly filter down to the documents you want and project exactly the data structured you want your content in. GROQ also let you join documents, both by values and by explicit bi-directional references.

Creating and updating documents

The data store also comes with a powerful API for mutation and patches. With an HTTP POST request, you can send transactions described in JSON which let you create and delete documents, set and unset keys with values, and change those values, even in nested structures. It also lets you update text strings using diff-match-patch, and work with arrays. You don’t have to deal with document locking, and you can use revision ids if you want to explicitly prevent race conditions.

The asset pipeline
Sanity.io also comes with a complete asset pipeline that lets you upload files and images. When you upload images, Sanity.io will analyze them for metadata such as dimensions and color palettes, and extract a low-quality-image-placeholder data string and save this to an asset document. It's also possible to enable EXIF and GeoLocation metadata, which is turned off by default because of privacy. The asset CDN also accepts query parameters to transform images on demand. You can request custom resolution, formats, cropping, and out-of-the-box optimizations (such as webp).

Utility endpoints
The datastore also has other endpoints that let you quickly export all your documents in one go, query document revision history, set fine-grained custom access control, integrate with custom authentication controls, and set up webhooks.

Sanity Studio
Sanity Studio is the place where you edit and manage your content. Sanity Studio is an open-source CMS that connects to Sanity.io's datastore. It's a single page application written in React and published on npm. All its source code is published on GitHub. You can set up content types and their fields by setting up your schemas in plain JavaScript which Sanity Studio uses to build the editing environment. Once you have done your changes locally, you can build and deploy the Studio on any web host, for example by running sanity deploy in the command-line tool which will deploy the Sanity Studio to our servers.

Editorial workflows
If you want to change the way documents are structured and listed out for custom editorial workflows, you can do so using Structure Builder. With templates for initial values, you can give editors a head start by letting them create new documents of a type with prefilled values. You can define field validation with out-of-the-box methods, or write completely custom methods with JavaScript. You can also create your own widgets for the Dashboard, or whole custom Studio tools to augment editorial workflows, make shortcuts, and so on.

Advanced customization
Sanity Studio was designed for customization. It is what you want to use instead of building your own CMS. It comes with a bunch of extension points and a plugin ecosystem.

You don’t need to know React to set up Sanity Studio, but if you do, you can customize it further by creating your own custom input components and custom previews. There’s very little need for state management since you’ll be dealing with real-time content that propagates into the components via the props.

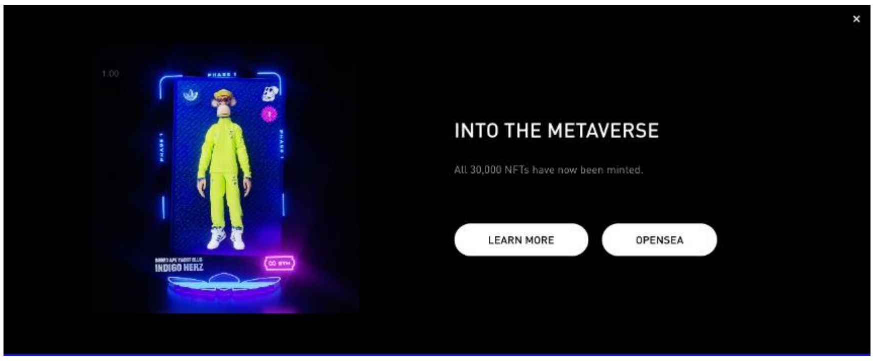
4.NEXT .Js

Next.js is an open-source web development framework built on top of Node.js enabling React-based web applications functionalities such as server-side rendering and generating static websites. React documentation mentions Next.js among "Recommended Toolchains" advising it to developers as a solution when "Building a server-rendered website with Node.js".[4] Where traditional React apps can only render their content in the client-side browser, Next.js extends this functionality to include applications rendered on the server-side.

Background[edit]

Next.js is a React framework that enables several extra features, including server-side rendering and generating static websites.[7] React is a JavaScript library that is traditionally used to build web applications rendered in the client's browser with JavaScript.[8] Developers recognize several problems with this strategy however, such as not catering to users who do not have access to JavaScript or have disabled it, potential security issues, significantly extended page loading times, and it can harm the site's overall search engine optimization.[8] Frameworks such as Next.js sidestep these problems by allowing some or all of the website to be rendered on the server-side before being sent to the client.[8][9] Next.js is one of the most popular frameworks for React.[10] It is one of several recommended "toolchains" available when starting a new app, all of which provide a layer of abstraction to aid in common tasks.[11] Next.js requires Node.js and can be initialized using Node Package Manager.

TECHNOLOGY USED:


INTO THE METAVERSE

All 30,000 NFTs have now been minted.

OBJECTIVE OF THE SYSTEM:

The objective of this project is to develop the NFT Marketplace and providing users below listed features:

Low fees and gas-free transactions.

Upload any Music, Animations, Game skins.

Well established Ethereum based blockchain system.

Log History of all transaction of a specific NFT.

Special 5% Royalty to the creator on each transaction.

REQUIREMENT ANALYSIS

HARDWARE REQUIREMENTS

In order to implement a new system, the choice of processor with maximum possible speed is made. There should be sufficient memory to store data and software tools for efficient processing. system:

Mobile Android: Android version 5.0 (Lollipop)

Processor: Media Tek (Hello P23)

Memory: 2 GB RAM

Internal Memory: 16 GB

SOFTWARE REQUIREMENTS

To develop application software, we use different types of software. The software for the development has been selected based on several factors such as:

- Support
- Cost Effectiveness
- Development Speed
- Ability to create robust application least time
- Stability

SOFTWARE DEVELOPMENT MODEL USED:

1. Agile

When the cost of change isn't too high, Agile can be used. Some of the major characteristics of **Agile** are that team members are self-empowered and communicate regularly without extensive documentation. If the team agrees that there's a better way to do the project, they move right ahead with the new approach. The project scope can be discovered as the project moves forward.

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like –

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing.

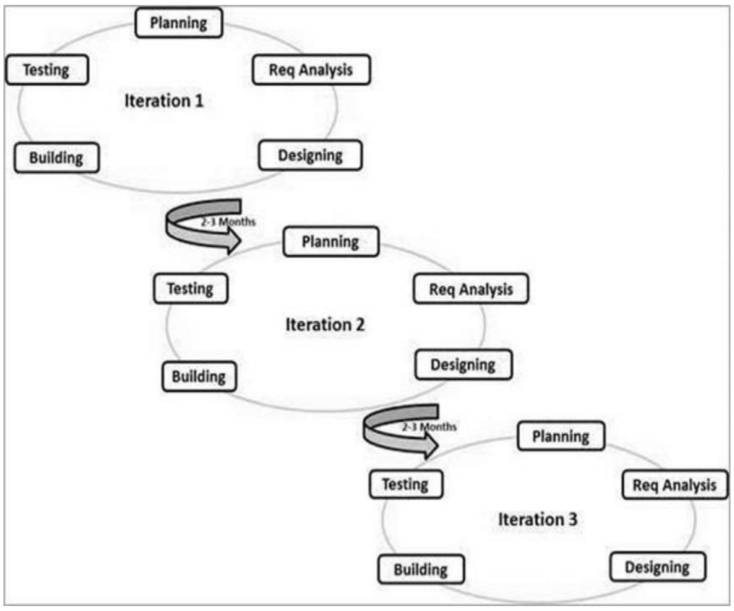
At the end of the iteration, a working product is displayed to the customer and important stakeholders.

What is Agile?

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

Here is a graphical illustration of the Agile Model –



The Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

The most popular Agile methods include Rational Unified Process (1994), Scrum (1995), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995). These are now collectively referred to as **Agile Methodologies**, after the Agile Manifesto was published in 2001.

Following are the Agile Manifesto principles –

- **Individuals and interactions** – In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- **Working software** – Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
- **Customer collaboration** – As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
- **Responding to change** – Agile Development is focused on quick responses to change and continuous development.

Agile Vs Traditional SDLC Models

Agile is based on the **adaptive software development methods**, whereas the traditional SDLC models like the waterfall model is based on a predictive approach. Predictive teams in the traditional SDLC models usually work with detailed planning and have a complete forecast of the exact tasks and features to be delivered in the next few months or during the product life cycle.

Predictive methods entirely depend on the **requirement analysis and planning** done in the beginning of cycle. Any changes to be incorporated go through a strict change control management and prioritization.

Agile uses an **adaptive approach** where there is no detailed planning and there is clarity on future tasks only in respect of what features need to be developed. There is feature driven development and the team adapts to the changing product requirements dynamically. The product is tested very frequently, through the release iterations, minimizing the risk of any major failures in future.

Customer Interaction is the backbone of this Agile methodology, and open communication with minimum documentation are the typical features of Agile development environment. The agile teams work in close collaboration with each other and are most often located in the same geographical location.

Agile Model - Pros and Cons

Agile methods are being widely accepted in the software world recently. However, this method may not always be suitable for all products. Here are some pros and cons of the Agile model.

The advantages of the Agile Model are as follows –

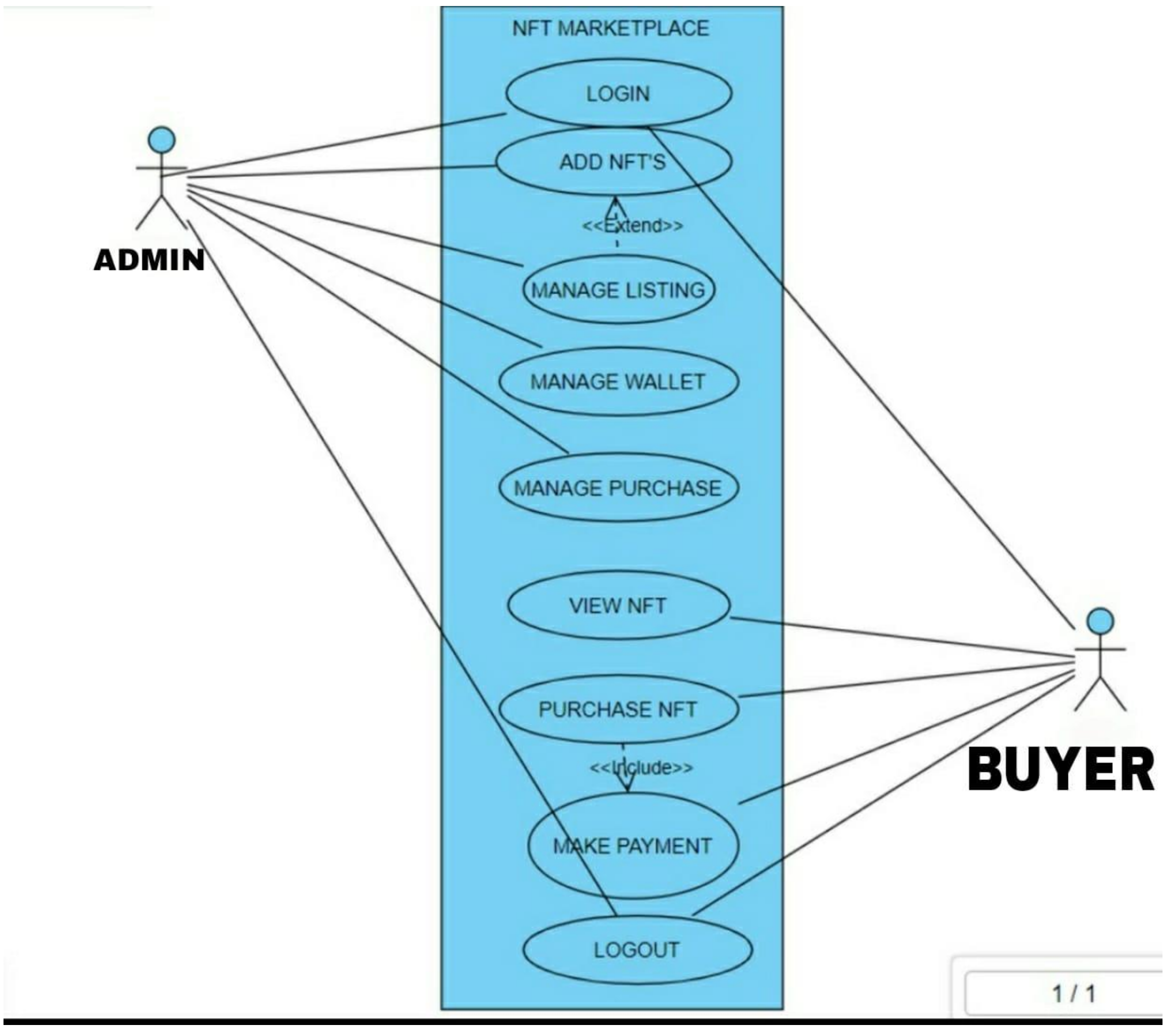
- Is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required.
- Easy to manage.
- Gives flexibility to developers.

The disadvantages of the Agile Model are as follows –

- Not suitable for handling complex dependencies.

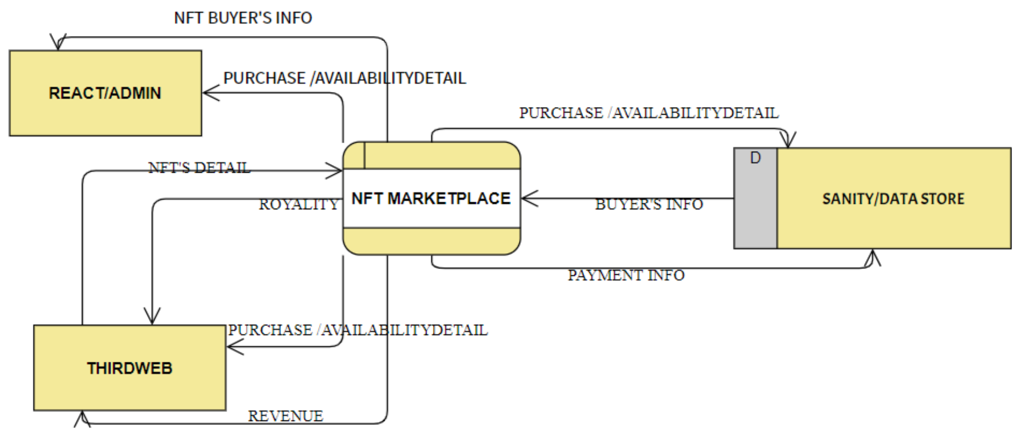
DETAILED SYSTEM SPECIFICATION

Use Case Diagram



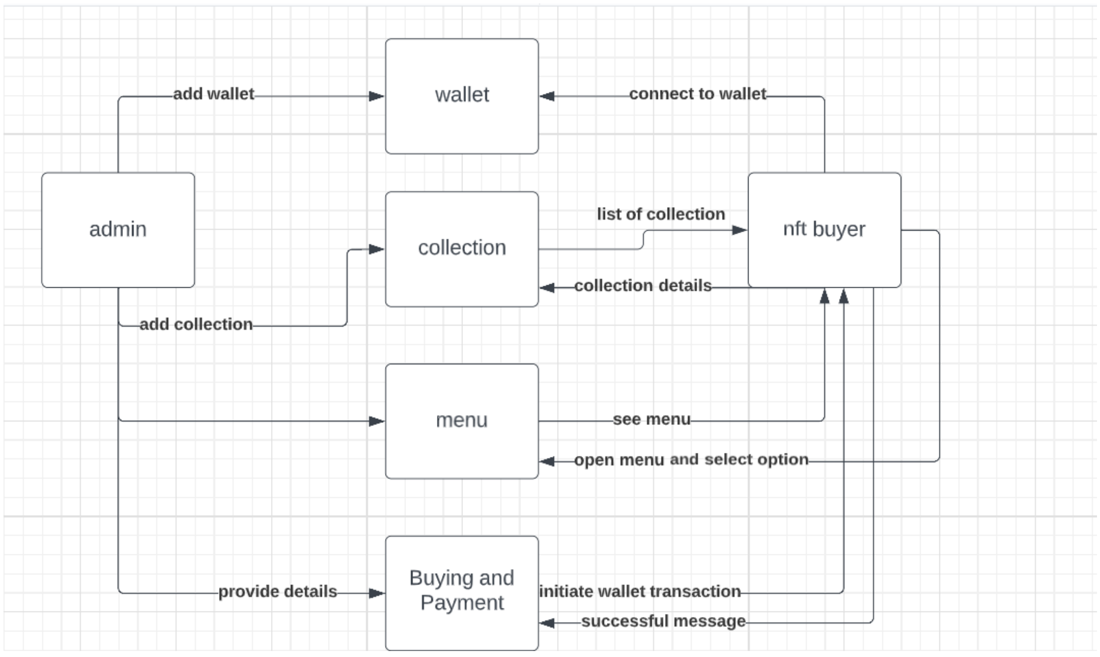
Data Flow Diagram

DFD Level 0 is also called a **Context Diagram**. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities.



Level 0

As described previously, context diagrams (level 0 DFDs) are diagrams where the whole system is represented as a single process. A level 1 DFD **notates each of the main sub-processes that together form the complete system**. We can think of a level 1 DFD as an “exploded view” of the context diagram.

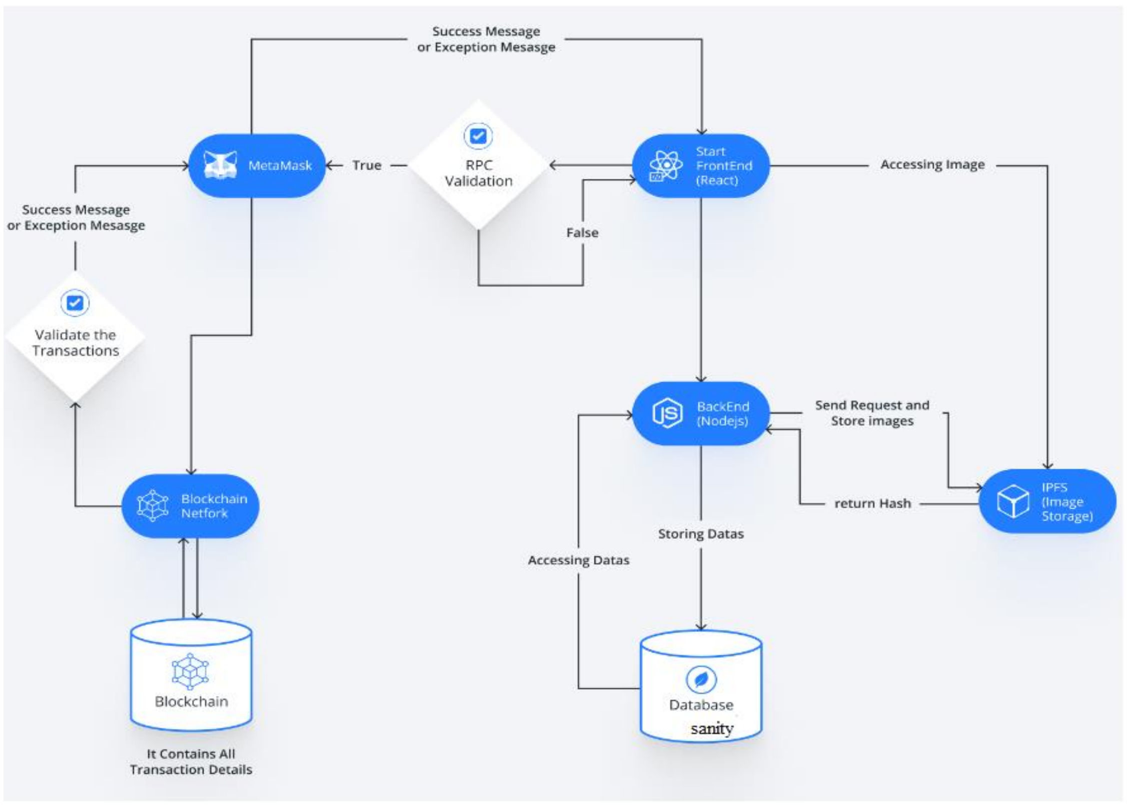


Level 1

Activity Diagram

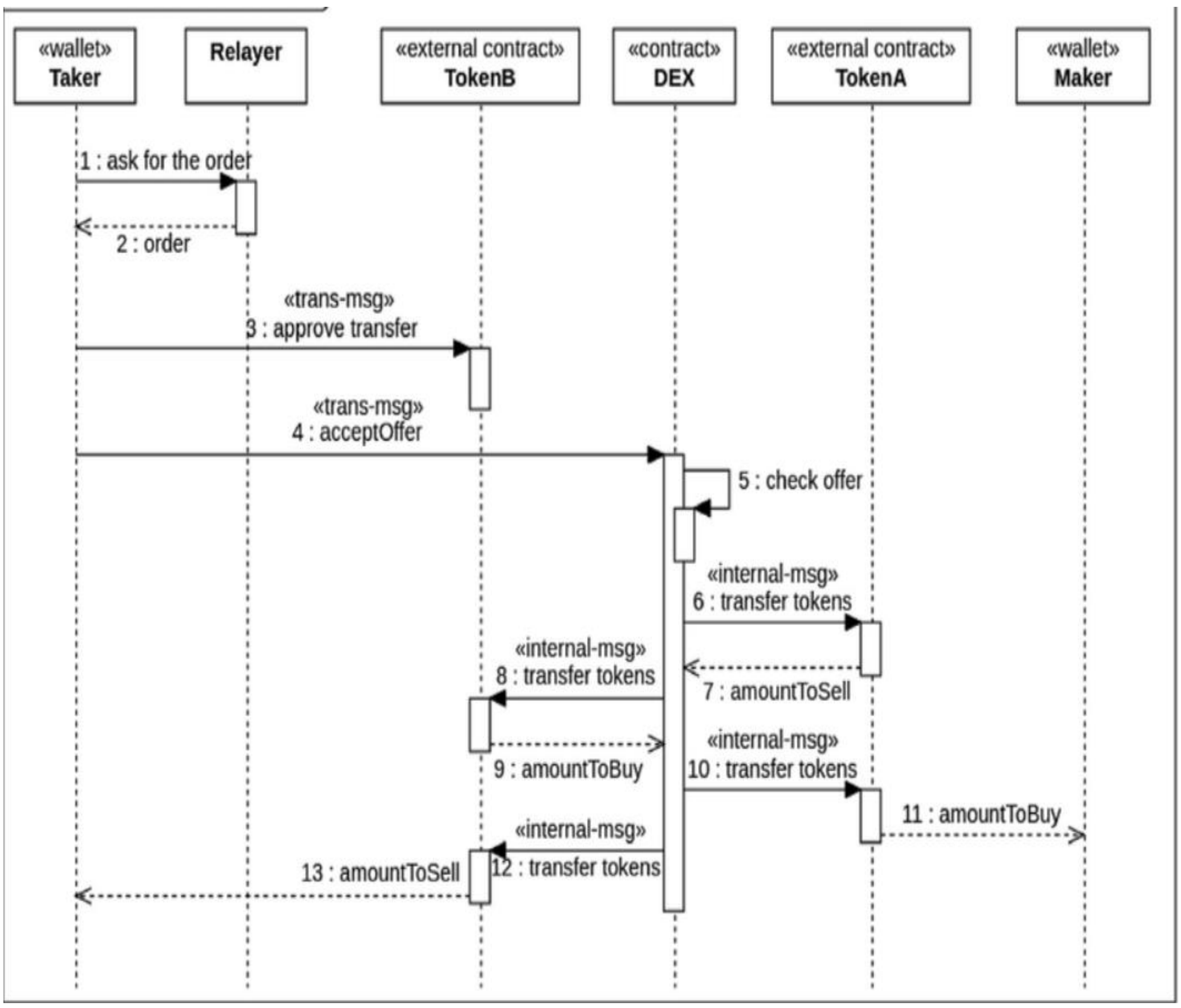
Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc



Sequence Diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

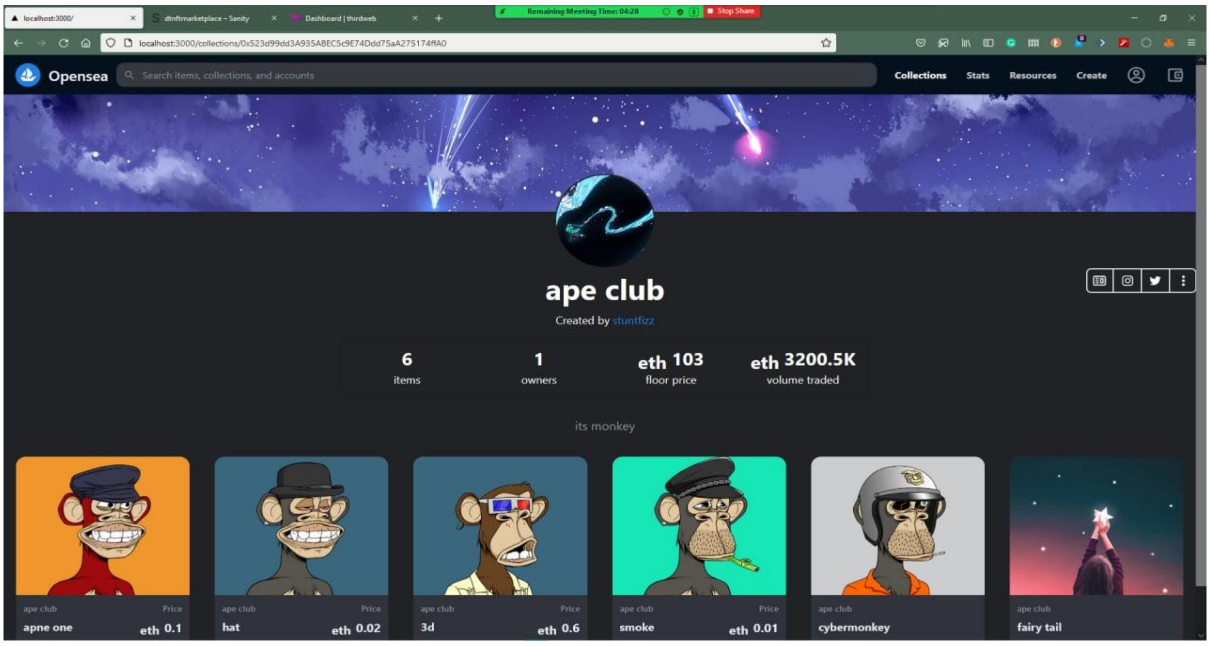

«wallet» Taker
Relayer
«external contract» TokenB
«contract» DEX
«external contract» TokenA
«wallet» Maker
1 : ask for the order
2 : order
«trans-msg» 3 : approve transfer
«trans-msg» 4 : acceptOffer
5 : check offer
«internal-msg» 6 : transfer tokens
7 : amountToSell
«internal-msg» 8 : transfer tokens
9 : amountToBuy
«internal-msg» 10 : transfer tokens
11 : amountToBuy
«internal-msg» 12 : transfer tokens
13 : amountToSell

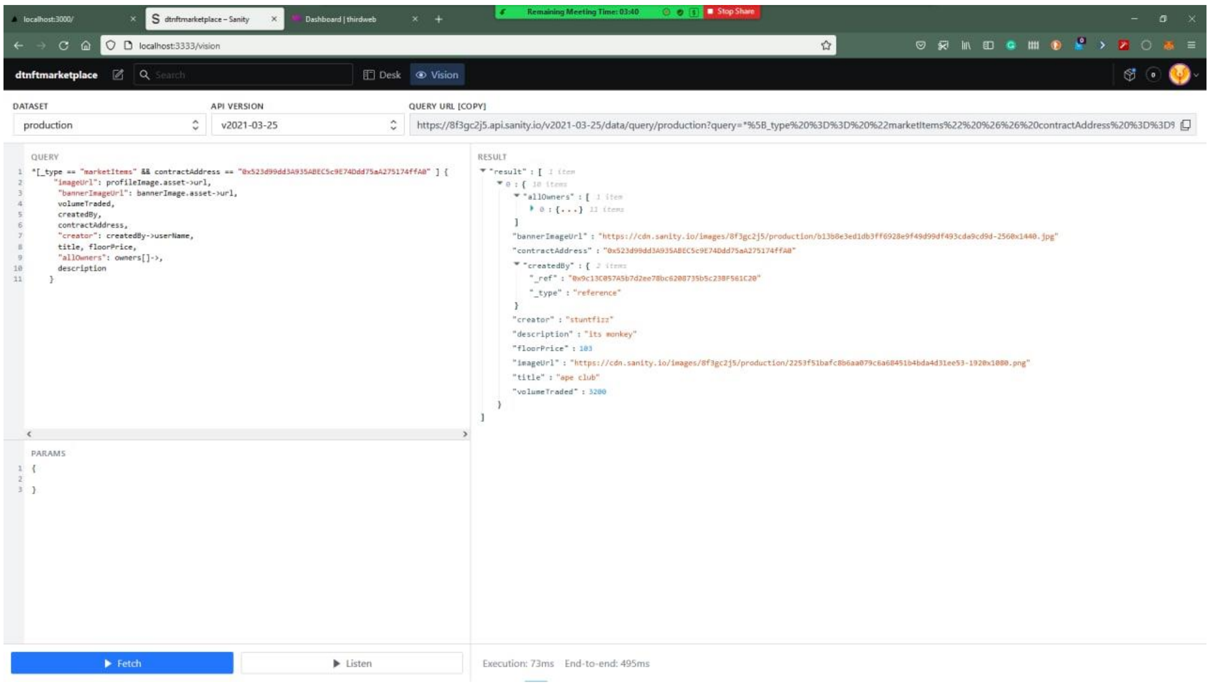
DATABASE DESIGN

A database system is an overall collection of different database software components and database containing the parts viz. Database application programs, front-end components, Database Management Systems, and Databases. Normalization Normalization is the process of organizing data in a database. This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency. Redundant data wastes disk space and creates maintenance problems .If data that exists in more than one place must be changed , the data must be changed in exactly the same way in all locations. A buyer's or promoter's address change is much easier to implement if that data is stored only in the buyer's or promoter's table and nowhere else in the database. There are a few rules for database normalization. Each rule is called a "normal form Data structuring is defined through a process called normalization. Data are grouped in the simplest way possible so that later changes can be made with a minimum of impact on the data structure.

There are different forms of normal forms

- First normal form(1NF)
- Second normal form(2NF)
- Third normal form(3NF)
- Boyce code normal form(BCNF)
- Fourth Normal form(4NF)
- Fifth Normal Form(5NF)





DATASET: production
API VERSION: v2021-03-25
QUERY URL [COPY]: https://8f3gc2j5.api.sanity.io/v2021-03-25/data/query/production?query=*%5B_type%20%3D%3D%20%22marketItems%22%20%26%26%20contractAddress%20%3D%3D

QUERY

```
*[_type == "marketItems" && contractAddress == "0x523d99dd3A935ABEC5c9E74Ddd75aA275174ffA0" ] {
     "imageUrl": profileImage.asset->url,
      "bannerImageUrl": bannerImage.asset->url,
      volumeTraded,
      createdBy,
      contractAddress,
      "creator": createdBy->userName,
      title, floorPrice,
      "allOwners": owners[]->,
      description
    }
```

RESULT

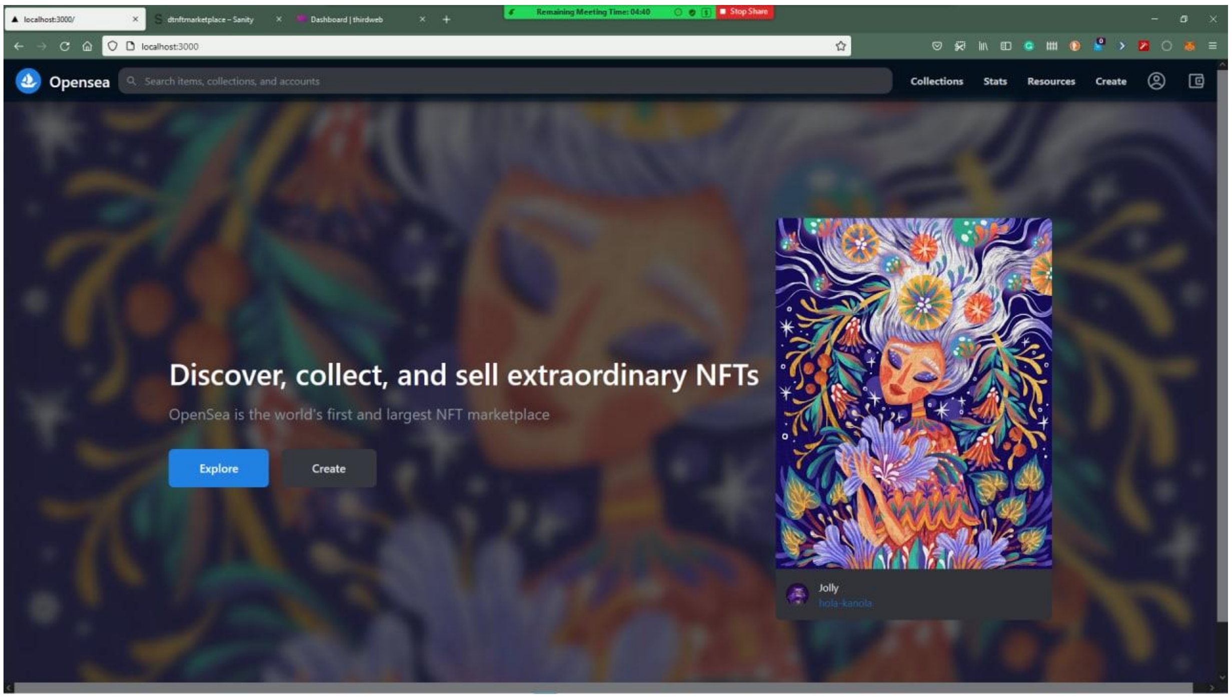
```
▼ "result" : [ 1 item
  ▼ 0 : { 10 items
    ▼ "allOwners" : [ 1 item
      ▸ 0 : {...} 11 items
    ]
    "bannerImageUrl" : "https://cdn.sanity.io/images/8f3gc2j5/production/b13b8e3ed1db3ff6928e9f49d99df493cda9cd9d-2560x1440.jpg"
    "contractAddress" : "0x523d99dd3A935ABEC5c9E74Ddd75aA275174ffA0"
    ▼ "createdBy" : { 2 items
      "_ref" : "0x9c13C057A5b7d2ee78bc6208735b5c238F561C20"
      "_type" : "reference"
    }
    "creator" : "stuntfizz"
    "description" : "its monkey"
    "floorPrice" : 103
    "imageUrl" : "https://cdn.sanity.io/images/8f3gc2j5/production/2253f51bafc8b6aa079c6a68451b4bda4d31ee53-1920x1080.png"
    "title" : "ape club"
    "volumeTraded" : 3200
  }
]
```

PARAMS

```
{
}
```

MODULES

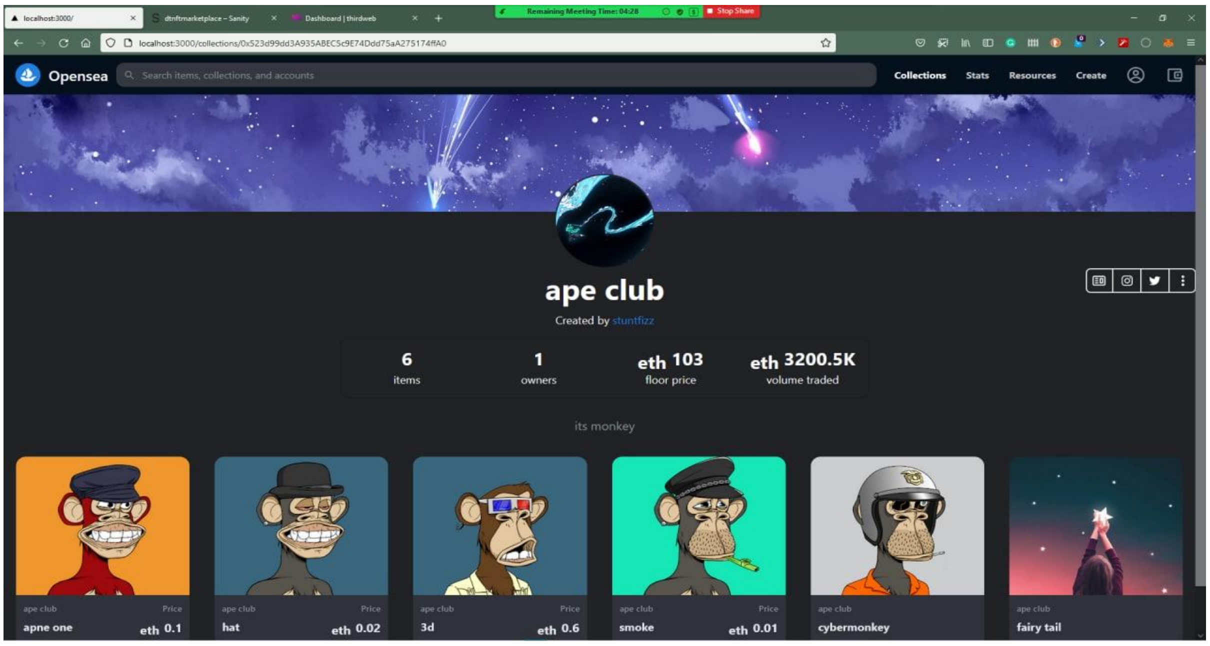
FRONT PAGE:



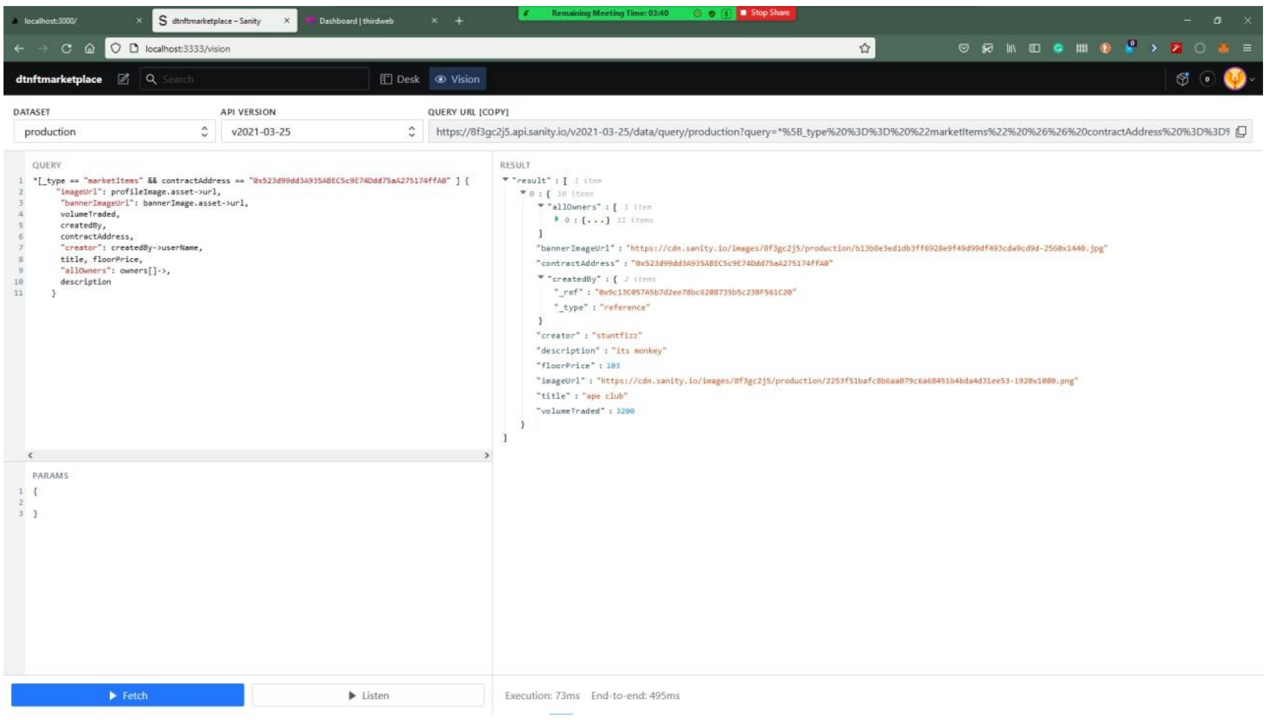
Discover, collect, and sell extraordinary NFTs

OpenSea is the world's first and largest NFT marketplace

DATASTORE/DATABASE



QUERRY/TABLE FOR DATABASE



DATASET: production

API VERSION: v2021-03-25

QUERY URL [COPY]: https://8f3gc2j5.api.sanity.io/v2021-03-25/data/query/production?query=*%5B_type%20%3D%3D%20%22marketItems%22%20%26%26%20contractAddress%20%3D%3D

QUERY

```
*[_type == "marketItems" && contractAddress == "0x523d99dd3A935ABEC5c9E74Ddd75aA275174ffA0" ] {
     "imageUrl": profileImage.asset->url,
      "bannerImageUrl": bannerImage.asset->url,
      volumeTraded,
      createdBy,
      contractAddress,
      "creator": createdBy->userName,
      title, floorPrice,
      "allOwners": owners[]->,
      description
    }
```

PARAMS

```
{

}
```

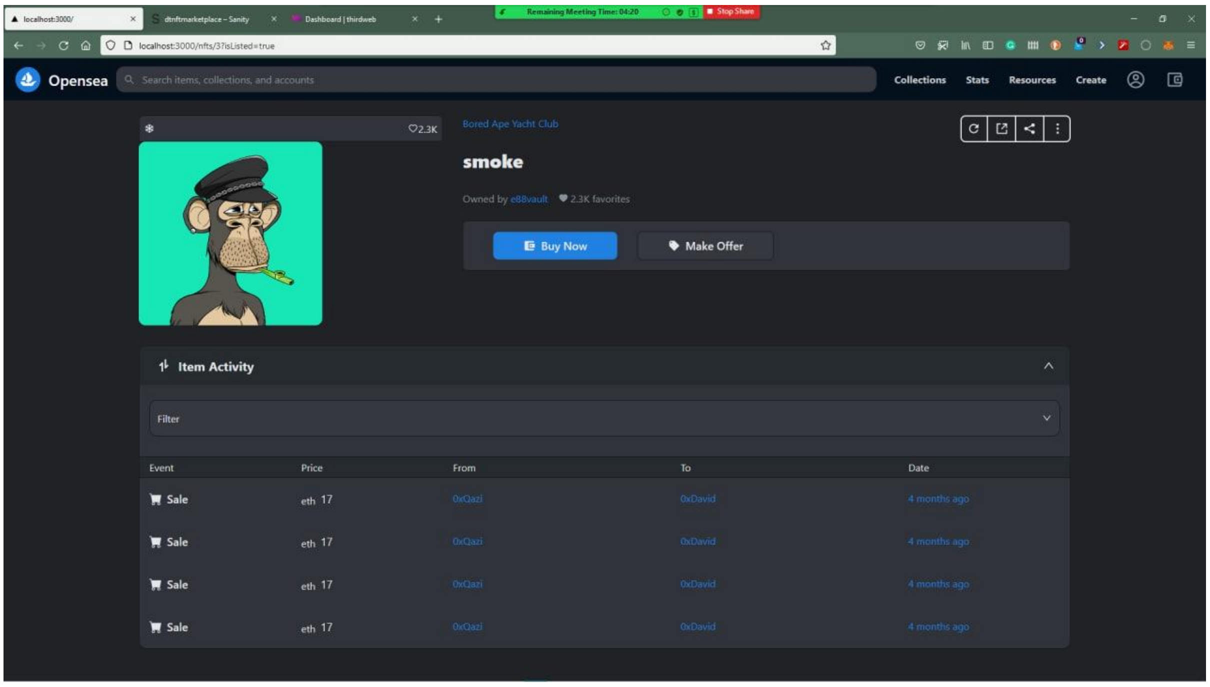
RESULT

```
"result" : [ 1 item
  0 : { 10 items
    "allOwners" : [ 1 item
      0 : {...} 11 items
    ]
    "bannerImageUrl" : "https://cdn.sanity.io/images/8f3gc2j5/production/b13b8e3ed1db3ff6928e9f49d99df493cda9cd9d-2560x1440.jpg"
    "contractAddress" : "0x523d99dd3A935ABEC5c9E74Ddd75aA275174ffA0"
    "createdBy" : { 2 items
      "_ref" : "0x9c13C057A5b7d2ee78bc6208735b5c238F561C20"
      "_type" : "reference"
    }
    "creator" : "stuntfizz"
    "description" : "its monkey"
    "floorPrice" : 103
    "imageUrl" : "https://cdn.sanity.io/images/8f3gc2j5/production/2253f51bafc8b6aa079c6a68451b4bda4d31ee53-1920x1080.png"
    "title" : "ape club"
    "volumeTraded" : 3200
  }
]
```

Execution: 73ms End-to-end: 495ms

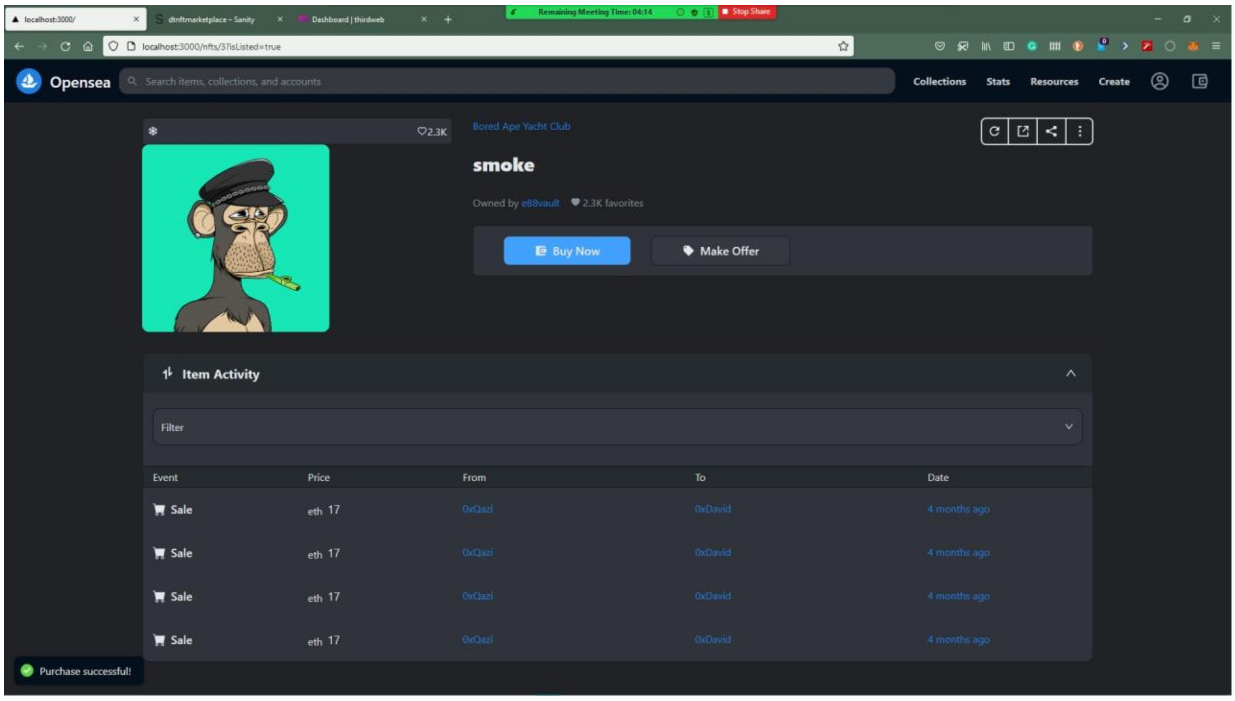
NFT ITEM BUYING PROCEDURE

STEP1:



Event	Price	From	To	Date
Sale	eth 17	0xQazi	0xDavid	4 months ago
Sale	eth 17	0xQazi	0xDavid	4 months ago
Sale	eth 17	0xQazi	0xDavid	4 months ago
Sale	eth 17	0xQazi	0xDavid	4 months ago

STEP 2:PAYMENT DONE AT THIS STEP AND VERIFICATION DONE AT THIS STEP AFTER SUCCESSFULLY DOING PAYMENT THROUGH ETHERIUM

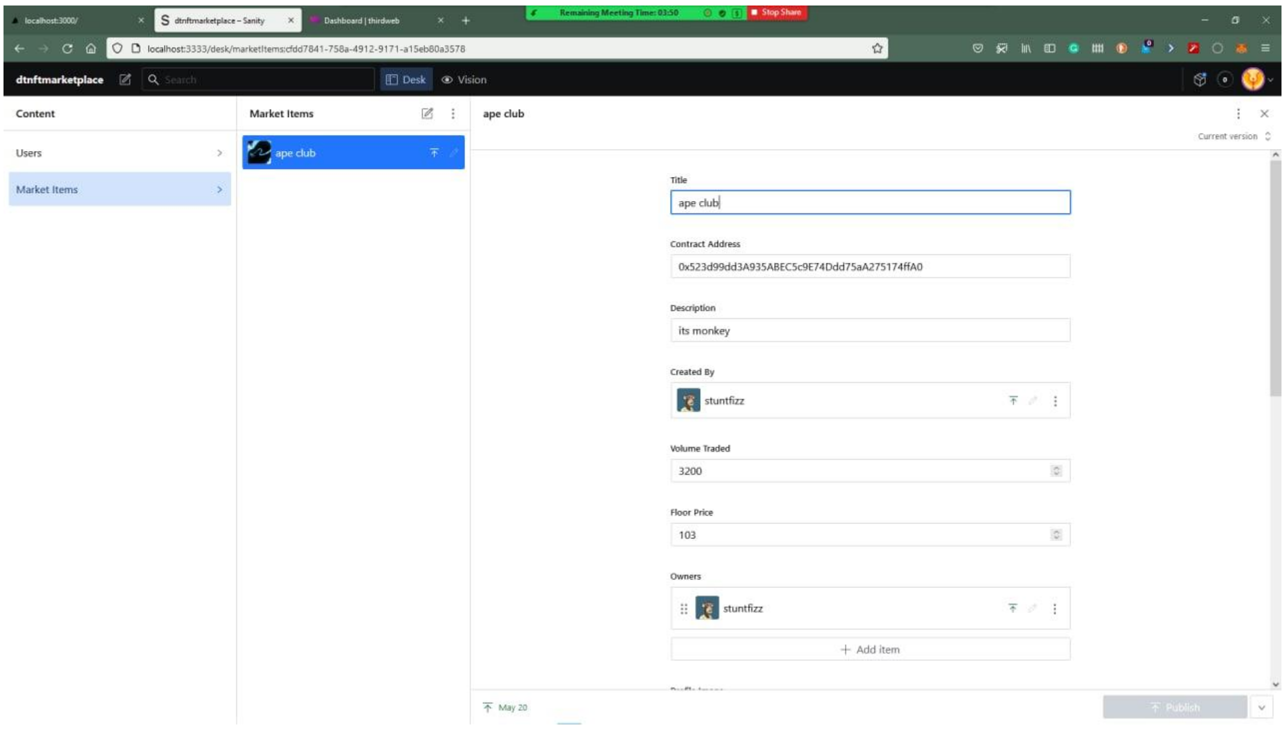


Event	Price	From	To	Date
Sale	eth 17	0xQazi	0xDavid	4 months ago
Sale	eth 17	0xQazi	0xDavid	4 months ago
Sale	eth 17	0xQazi	0xDavid	4 months ago
Sale	eth 17	0xQazi	0xDavid	4 months ago

Purchase successful!

THE BACK-END PART

ABOUT CONTRACT ID WITRH THE THIRWEB(BLOCKCHAIN)



Title: ape club

Contract Address: 0x523d99dd3A935ABEC5c9E74Ddd75aA275174ffA0

Description: its monkey

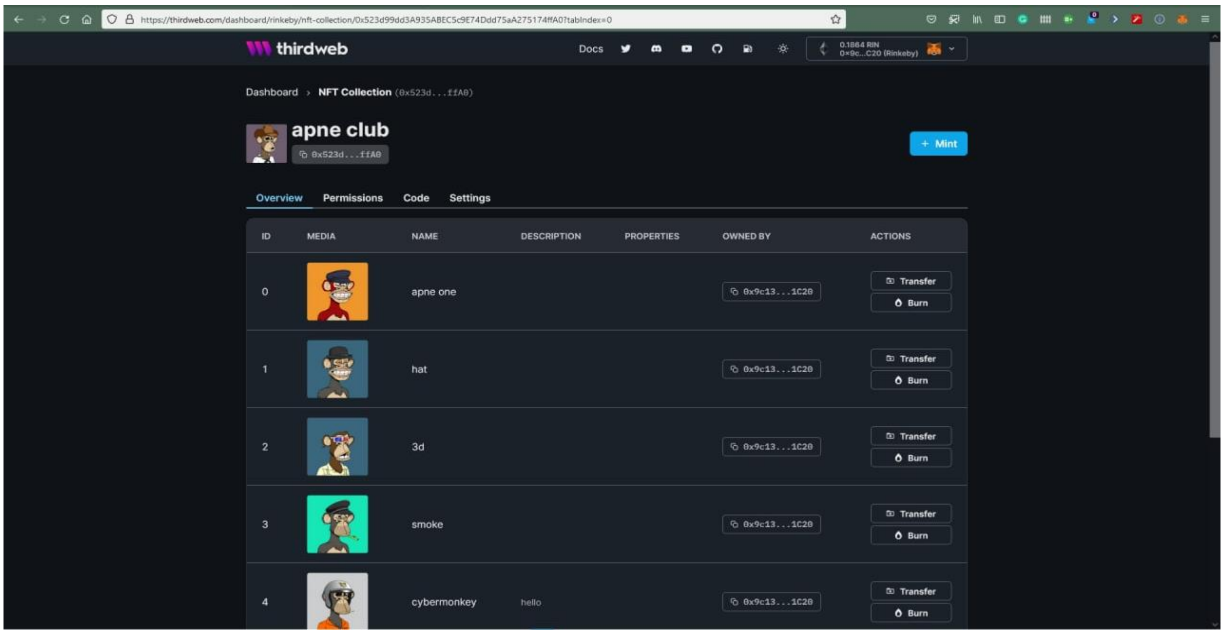
Created By: stuntfizz

Volume Traded: 3200

Floor Price: 103

Owners: stuntfizz

THIRD WEB WHETRE NFT STORED :



https://thirdweb.com/dashboard/rinkeby/nft-collection/0x523d99dd3A935ABEC5c9E74Ddd75aA275174ffA0?tabIndex=0

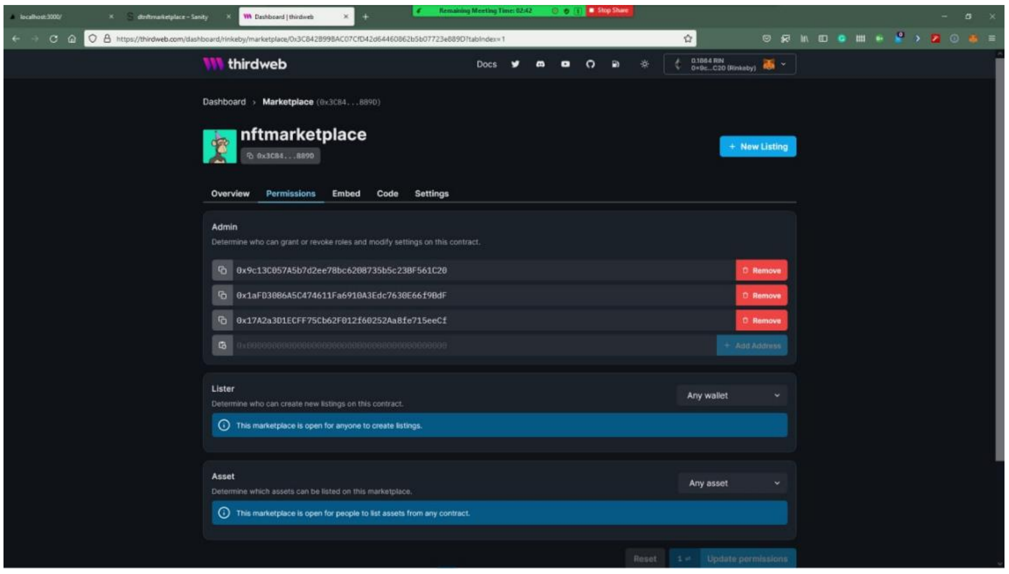
Dashboard › NFT Collection (0x523d...ffA0)

apne club

0x523d...ffA0

ID	MEDIA	NAME	DESCRIPTION	PROPERTIES	OWNED BY	ACTIONS
0		apne one			0x9c13...1C20	Transfer Burn
1		hat			0x9c13...1C20	Transfer Burn
2		3d			0x9c13...1C20	Transfer Burn
3		smoke			0x9c13...1C20	Transfer Burn
4		cybermonkey	hello		0x9c13...1C20	Transfer Burn

POSTED NFT ART FOR SELL



https://thirdweb.com/dashboard/rinkeby/marketplace/0x3C842B99BAC07CfD42d64460862b5b07723e889D?tabIndex=1

Dashboard › Marketplace (0x3C84...889D)

nftmarketplace

0x3C84...889D

Admin

Determine who can grant or revoke roles and modify settings on this contract.

0x9c13C057A5b7d2ee78bc6208735b5c23BF561C20

0x1aFD3086A5C474611Fa6910A3Edc7630E66f9BdF

0x17A2a3D1ECFF75Cb62F012f60252Aa8fe715eeCf

Lister

Determine who can create new listings on this contract.

Any wallet

This marketplace is open for anyone to create listings.

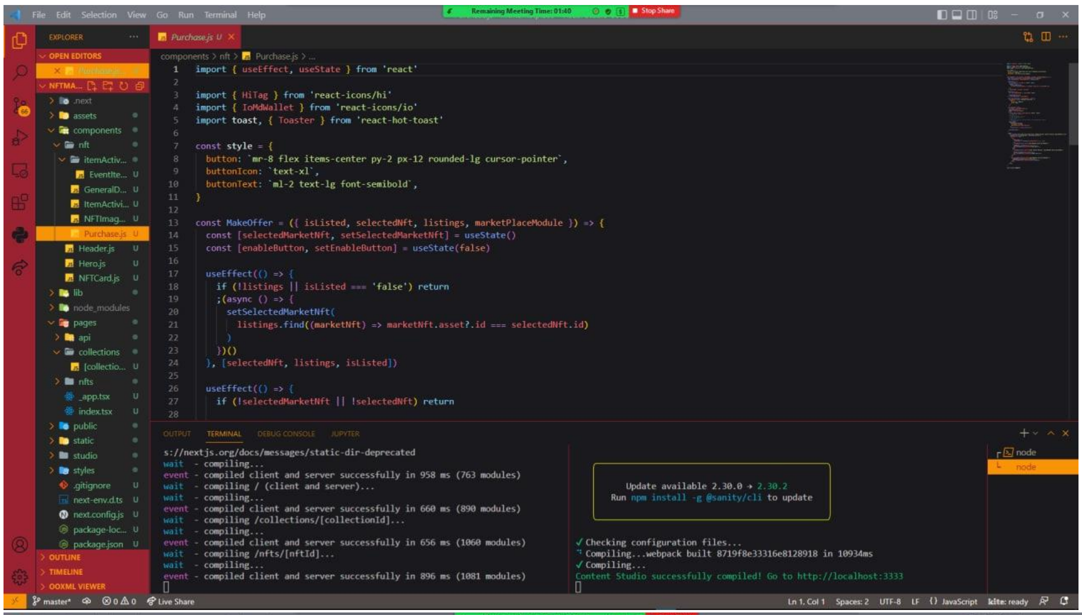
Asset

Determine which assets can be listed on this marketplace.

Any asset

This marketplace is open for people to list assets from any contract.

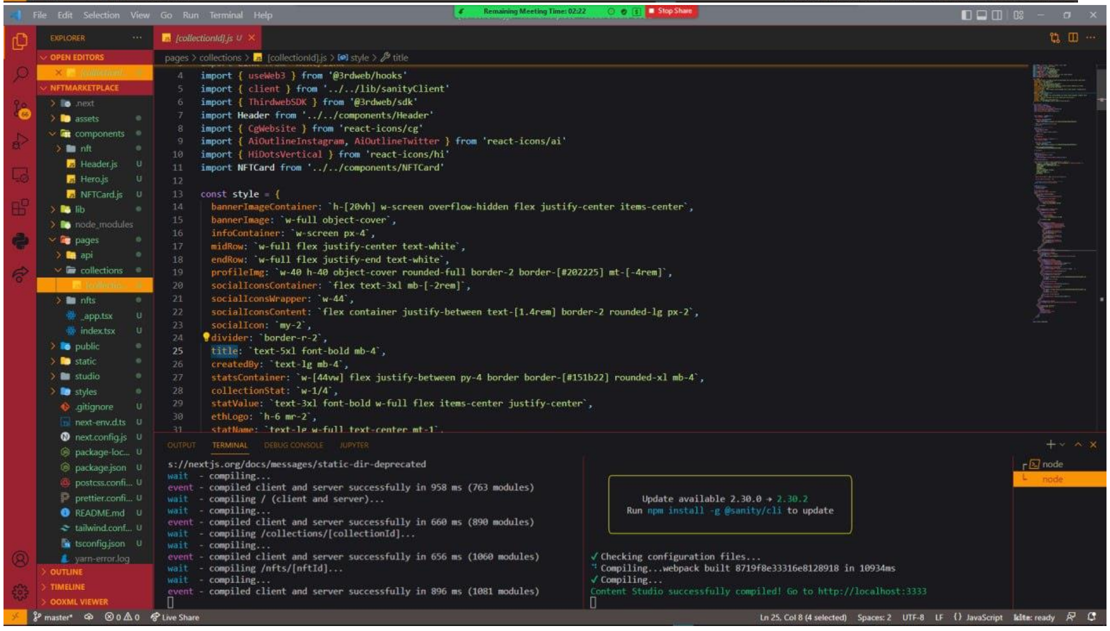
SOME IMPORTANT CODES



```
components > nft > Purchase.js > ...
import { useEffect, useState } from 'react'

import { HiTag } from 'react-icons/hi'
import { IoMdWallet } from 'react-icons/io'
import toast, { Toaster } from 'react-hot-toast'

const style = {
  button: `mr-8 flex items-center py-2 px-12 rounded-lg cursor-pointer`,
  buttonIcon: `text-xl`,
  buttonText: `ml-2 text-lg font-semibold`,
}

const MakeOffer = ({ isListed, selectedNft, listings, marketPlaceModule }) => {
  const [selectedMarketNft, setSelectedMarketNft] = useState()
  const [enableButton, setEnableButton] = useState(false)

  useEffect(() => {
    if (!listings || isListed === 'false') return
    ;(async () => {
      setSelectedMarketNft(
        listings.find((marketNft) => marketNft.asset?.id === selectedNft.id)
      )
    })()
  }, [selectedNft, listings, isListed])

  useEffect(() => {
    if (!selectedMarketNft || !selectedNft) return

```

```
s://nextjs.org/docs/messages/static-dir-deprecated
wait  - compiling...
event - compiled client and server successfully in 958 ms (763 modules)
wait  - compiling / (client and server)...
wait  - compiling...
event - compiled client and server successfully in 660 ms (890 modules)
wait  - compiling /collections/[collectionId]...
wait  - compiling...
event - compiled client and server successfully in 656 ms (1060 modules)
wait  - compiling /nfts/[nftId]...
wait  - compiling...
event - compiled client and server successfully in 896 ms (1081 modules)
```

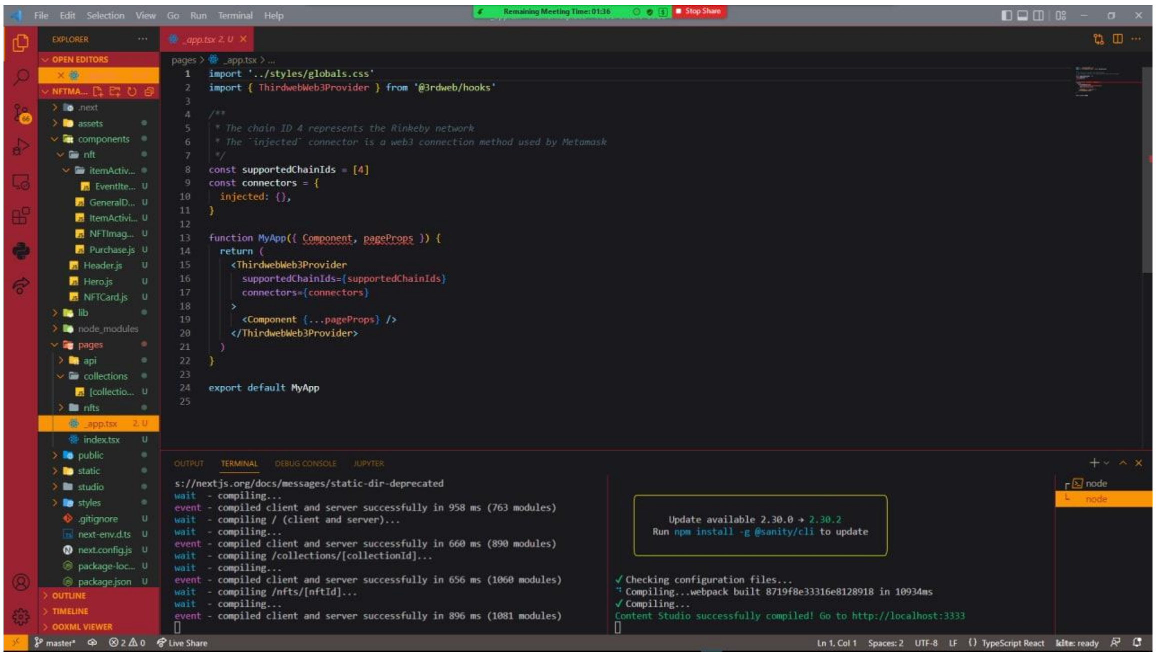
```
   Update available 2.30.0 → 2.30.2
 Run npm install -g @sanity/cli to update

✓ Checking configuration files...
⠋ Compiling...webpack built 8719f8e33316e8128918 in 10934ms
✓ Compiling...
Content Studio successfully compiled! Go to http://localhost:3333
```



```
pages > collections > [collectionId].js > style > title
import { useWeb3 } from '@3rdweb/hooks'
import { client } from '../../lib/sanityClient'
import { ThirdwebSDK } from '@3rdweb/sdk'
import Header from '../../components/Header'
import { CgWebsite } from 'react-icons/cg'
import { AiOutlineInstagram, AiOutlineTwitter } from 'react-icons/ai'
import { HiDotsVertical } from 'react-icons/hi'
import NFTCard from '../../components/NFTCard'

const style = {
  bannerImageContainer: `h-[20vh] w-screen overflow-hidden flex justify-center items-center`,
  bannerImage: `w-full object-cover`,
  infoContainer: `w-screen px-4`,
  midRow: `w-full flex justify-center text-white`,
  endRow: `w-full flex justify-end text-white`,
  profileImg: `w-40 h-40 object-cover rounded-full border-2 border-[#202225] mt-[-4rem]`,
  socialIconsContainer: `flex text-3xl mb-[-2rem]`,
  socialIconsWrapper: `w-44`,
  socialIconsContent: `flex container justify-between text-[1.4rem] border-2 rounded-lg px-2`,
  socialIcon: `my-2`,
  divider: `border-r-2`,
  title: `text-5xl font-bold mb-4`,
  createdBy: `text-lg mb-4`,
  statsContainer: `w-[44vw] flex justify-between py-4 border border-[#151b22] rounded-xl mb-4`,
  collectionStat: `w-1/4`,
  statValue: `text-3xl font-bold w-full flex items-center justify-center`,
  ethLogo: `h-6 mr-2`,
  statName: `text-lg w-full text-center mt-1`,
```

```
s://nextjs.org/docs/messages/static-dir-deprecated
wait  - compiling...
event - compiled client and server successfully in 958 ms (763 modules)
wait  - compiling / (client and server)...
wait  - compiling...
event - compiled client and server successfully in 660 ms (890 modules)
wait  - compiling /collections/[collectionId]...
wait  - compiling...
event - compiled client and server successfully in 656 ms (1060 modules)
wait  - compiling /nfts/[nftId]...
wait  - compiling...
event - compiled client and server successfully in 896 ms (1081 modules)
```

```
   Update available 2.30.0 → 2.30.2
 Run npm install -g @sanity/cli to update

✓ Checking configuration files...
⠋ Compiling...webpack built 8719f8e33316e8128918 in 10934ms
✓ Compiling...
Content Studio successfully compiled! Go to http://localhost:3333
```



```
import '../styles/globals.css'
import { ThirdwebWeb3Provider } from '@3rdweb/hooks'

/**
 * The chain ID 4 represents the Rinkeby network
 * The `injected` connector is a web3 connection method used by Metamask
 */
const supportedChainIds = [4]
const connectors = {
  injected: {},
}

function MyApp({ Component, pageProps }) {
  return (
    <ThirdwebWeb3Provider
      supportedChainIds={supportedChainIds}
      connectors={connectors}
    >
      <Component {...pageProps} />
    </ThirdwebWeb3Provider>
  )
}

export default MyApp
```

```
s://nextjs.org/docs/messages/static-dir-deprecated
wait  - compiling...
event - compiled client and server successfully in 958 ms (763 modules)
wait  - compiling / (client and server)...
wait  - compiling...
event - compiled client and server successfully in 660 ms (890 modules)
wait  - compiling /collections/[collectionId]...
wait  - compiling...
event - compiled client and server successfully in 656 ms (1060 modules)
wait  - compiling /nfts/[nftId]...
wait  - compiling...
event - compiled client and server successfully in 896 ms (1081 modules)
```

```
   Update available 2.30.0 → 2.30.2
Run npm install -g @sanity/cli to update

✓ Checking configuration files...
⠙ Compiling...webpack built 8719f8e33316e8128918 in 10934ms
✓ Compiling...
Content Studio successfully compiled! Go to http://localhost:3333
```

SYSTEM IMPLEMENTATION

WORKING OF SYSTEM -

System implementation is the final phase that is putting the utility into action. Implementation is the state in the project where theoretical design turned into working system. The most crucial stage is achieving a new successful system and giving confidence in new system that it will work efficiently and effectively. The system is implemented only after thorough checking is done and it is found working according to the specifications.

System implementation is the final phase. i.e., putting the utility into action. Implementation is the state in the project where theoretical design turned into working system. The most crucial stage is achieving a new successful system and giving confidence in new system that it will work efficiently and effectively. Implementation is the state in the project where theoretical design turned into working system. The implementation stage is a system project in its own right. It involves careful planning, design ,investigation of the current system and constraints on implementation, design of methods to achieve change over and evolution method. Once the planning has been completed the major effort is to ensure that the programs in the system are working properly. At the same time concentrate on training user staff.

The major implementation procedures are:-

- Test plans
- Training
- Equipment installation
- Conversio

Test plans

The implementation of a computer based system requires that the test data can be prepared and the system and its elements be tested in a structured manner.

Training

The purpose of training is to ensure that all the personnel who are to be associated with the computer based system possesses necessary knowledge skills.

Equipment installation

Equipment vendors can provide specifications for equipment installation. They usually work with projects equipment installation team is planning for adequate space, power and light, and a suitable environment. After a suitable site has been completed, the computer equipment can be installed.

Conversion

It is the processes of performing all of the operations that result directly in turnover of the new system to the user. Conversion has two parts:-

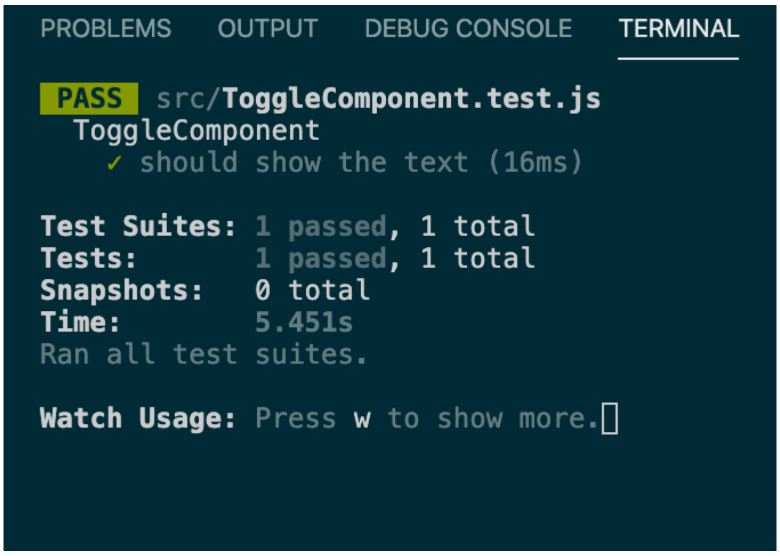
- The creation of a conversion plan at the start of the development phase and the implementation of the plan throughout the development phase.
- The creation of a system change over plan at the end of the development phase and the implementation of the plan at the beginning of operation phase.

TEST CASES

Test case 1.

Our `ToogleComponent` has the text “This will be toggled” inside a nested `div`. In the first test case, we will test to see that the component has the text “This will be toggled.”

Let’s write the test case for this. Delete all the code

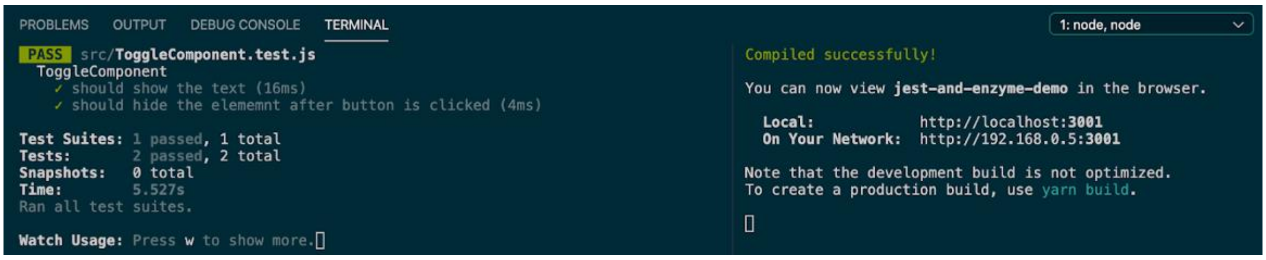


```
 PASS  src/ToggleComponent.test.js
  ToggleComponent
    ✓ should show the text (16ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        5.451s
Ran all test suites.

Watch Usage: Press w to show more.
```

Test case 2.

We want to test the clicking of the button that causes the text to be hidden. So let’s write the code to test this scenario:

```
it('should
hide the
element
after
button is
clicked',
()=>{
                const toggleInstance = shallow(<ToggleComponent />);
                /** Find the button element from toggleInstance ***/
                const toggleButton= toggleInstance.find('button');
               /** Now problem is how we can find the button clicked without
                *   anything,
                * here enzyme will help us a lot again. It has a function called
                * simulate,
                * thought this simulate function we can simulate the DOM events
                */
                toggleButton.simulate('click');
                const element=toggleInstance.find('div div');
                /*it means after the click the text doesn't exist, and its true*/
                expect(element.length).toBe(0);
             });
```

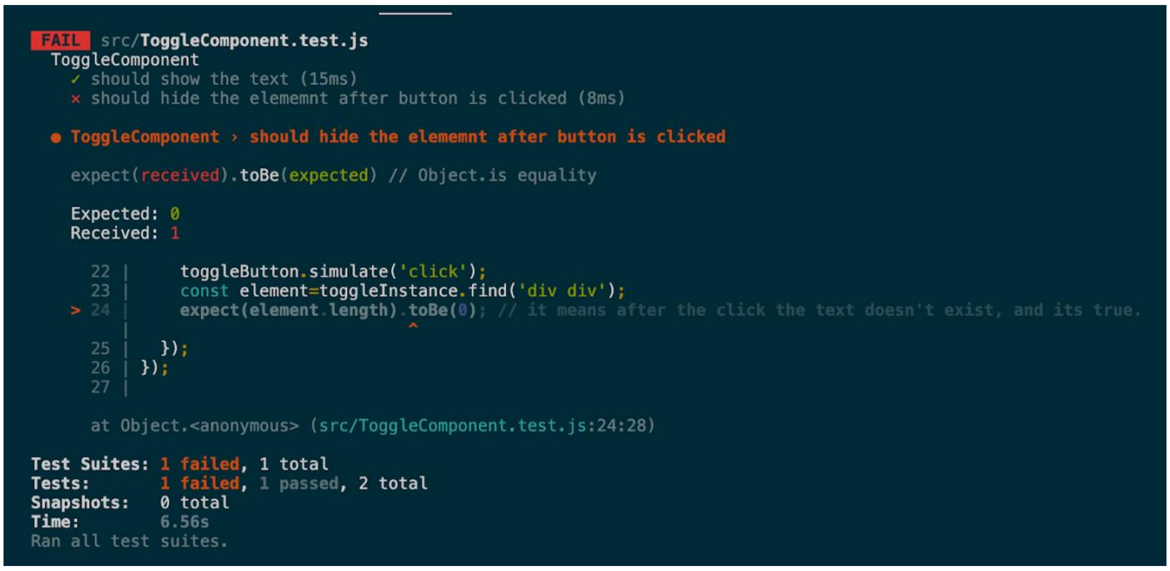


```
PASS  src/ToggleComponent.test.js
  ToggleComponent
    ✓ should show the text (16ms)
    ✓ should hide the elememnt after button is clicked (4ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        5.527s
Ran all test suites.

Watch Usage: Press w to show more.
```

```
Compiled successfully!

You can now view jest-and-enzyme-demo in the browser.

  Local:            http://localhost:3001
  On Your Network:  http://192.168.0.5:3001

Note that the development build is not optimized.
To create a production build, use yarn build.
```

For more clarity, we can remove the condition from `{isShowText && <div>This will be toggled</div>}` and make it simply `<div>This will be toggled</div>`

It’s being failed. This means our test case is working fine.



```
FAIL  src/ToggleComponent.test.js
  ToggleComponent
    ✓ should show the text (15ms)
    × should hide the elememnt after button is clicked (8ms)

  ● ToggleComponent › should hide the elememnt after button is clicked

    expect(received).toBe(expected) // Object.is equality

    Expected: 0
    Received: 1

      22 |         toggleButton.simulate('click');
      23 |         const element=toggleInstance.find('div div');
    > 24 |         expect(element.length).toBe(0); // it means after the click the text doesn't exist, and its true.
         |                                ^
      25 |     });
      26 | });
      27 |

      at Object.<anonymous> (src/ToggleComponent.test.js:24:28)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 passed, 2 total
Snapshots:   0 total
Time:        6.56s
Ran all test suites.
```

CONCLUSION & RESULT

With this project blockchain transaction is happening

NFT marketplaces are platforms where one can mint, store, display, and trade NFTs. These have played a pivotal role in the explosive growth of the NFT market cap. Check this out, recently, OpenSea, one of the leading NFT marketplaces crossed the record sales of $1.5 billion.

While there are many types of NFT marketplaces, universal or collectible-oriented marketplaces are the most popular ones. That’s because sports and collectible NFTs are the most traded NFTs and contribute to over 60% of the total sales.

FUTURE ENHANCEMENTS

- UPLOADING VIDEO/GIF/SONG AS NFT ON WEBSITE
- BLOCKCHAIN TRANSACTION PLATFORM FOR SPECIFICALLY THIS SITE
- MAKING IT LIVE WIBESITE ON VECTOR
- REGULAR BUG FIXES
- WEEKLY SECURITY UPDATES

BIBLIOGRAPHY

Ante, L., 2021a. The non-fungible token (NFT) market and its relationship with Bitcoin and Ethereum. BRL

Work. Pap. 20.

Ante, L., 2021b. Smart Contracts on the Blockchain – A Bibliometric Analysis and Review. Telemat.

Informatics 57, 101519. https://doi.org/10.1016/j.tele.2020.101519

Ante, L., Fiedler, I., Strehle, E., 2021. The impact of transparent money flows: Effects of stablecoin transfers on

the returns and trading volume of bitcoin. Technol. Forecast. Soc. Change 170, 120851.

https://doi.org/10.1016/j.techfore.2021.120851

Axie, 2021. The Great Migration - Ronin Phase 2 is live! [WWW Document]. URL

https://axie.substack.com/p/migration

BBC, 2017. CryptoKitties craze slows down transactions on Ethereum [WWW Document]. URL

https://www.bbc.com/news/technology-42237162

Bitcoin.com, 2021. NFT Immutability Debate Grows as Tokenized Tweets Get Deleted and NFT Images