

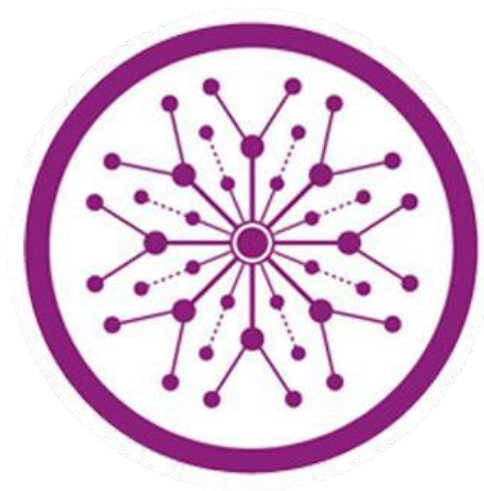
# **NFT META MARKET**

**Final Year Project**

**Session 2019-2023**

A project submitted in partial fulfillment of the degree of

**BS in Computer Science**



Department of Computer Science

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2023

Type (Nature of project)	<input checked="" type="checkbox"/> <b>Development</b> <input type="checkbox"/> <b>Research</b> <input type="checkbox"/> <b>R&amp;D</b>			
Area of specialization				
<b>FYP ID</b>	BCSM-FYP-F22-033			
<b>Project Group Members</b>				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BCSM-F19-491	Muhammad Ahmad	bcsm-f19-491@superior.edu.pk	
(ii)	BCSM-F19-052	Muhammad Ahmad Sohail	bcsm-f19-052@superior.edu.pk	
(iii)	BCSM-F19-499	Umair Ahmad	bcsm-f19-499@superior.edu.pk	

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

### Plagiarism Free Certificate

This is to certify that, I \_\_\_\_\_ S/D of \_\_\_\_\_, group leader of FYP under registration no \_\_\_\_\_ at Computer Science Department, The Superior University, Lahore. I declare that my FYP report is checked by my supervisor.

Date: \_\_\_\_\_ Name of Group Leader: Muhammad Ahmad Signature: \_\_\_\_\_

Name of Supervisor: Mam Maryam

Co-Supervisor: Sir Talha Amjad

Designation: Senior Lecturer

Designation: Lecturer

Signature: \_\_\_\_\_

Signature: \_\_\_\_\_

HoD: Dr. Irfan-Ud-Din

Signature: \_\_\_\_\_

## NFT Meta Market

### Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback from Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

## APPROVAL

---

### PROJECT SUPERVISOR

Comments: \_\_\_\_\_

---

Name: \_\_\_\_\_

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

---

### PROJECT MANAGER

Comments: \_\_\_\_\_

---

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

---

### HEAD OF THE DEPARTMENT

Comments: \_\_\_\_\_

---

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

---

## Dedication

*To the visionaries and innovators who are pushing the boundaries of what is possible with Blockchain technology and non-fungible tokens. This project is a tribute to their creativity and determination, and a testament to the power of collaboration and community in shaping the future of digital art and ownership. May this marketplace serve as a platform for the discovery and appreciation of unique digital assets and the artists who create them.*

## **Acknowledgements**

*We are really thankful to our supervisor for all the years of guidance and advice he has bestowed upon us. Our supervisor is the one who truly molded us into someone with valuable skills and talents. It was a privilege to work under his leadership.*

## **Executive Summary**

NFT's Market Place is a decentralized marketplace that allows users to buy and sell nonfungible tokens. Users can transact a variety of different digital collectibles, including art, music, Cultural masterpieces, and dozens more. Through this web application, the local artists of Pakistan will be able to express their cultural and artistic values globally. The web application incorporates AR technology through which the user is able to place the digital asset anywhere at home or the workplace and view it through his/her mobile device. The web application provides an easy-to-use interface that allows the user to sell/purchase NFTs easily.

## Table of Contents

<b>Dedication</b> .....	v
<b>Acknowledgements</b> .....	vi
<b>Executive Summary</b> .....	vii
<b>Table of Contents</b> .....	viii
<b>List of Figures</b> .....	xi
<b>Chapter 1</b> .....	1
<b>Introduction</b> .....	1
1.1. Background.....	2
1.2. Motivations and Challenges.....	2
1.3. Goals and Objectives.....	3
1.4. Literature Review/Existing Solutions.....	3
1.5. Gap Analysis.....	4
1.6. Proposed Solution.....	4
1.7. Project Plan.....	4
1.7.1. Work Breakdown Structure.....	5
1.7.2. Roles & Responsibility Matrix.....	5
1.7.3. Gantt Chart.....	6
<b>Chapter 2</b> .....	8
<b>Software Requirement Specifications</b> .....	8
2.1. Introduction.....	9
2.1.1. Purpose.....	9
2.1.2. Document Conventions.....	9
2.1.3. Intended Audience and Reading Suggestions.....	9
2.1.4. Product Scope.....	9
2.1.5. References.....	10
2.2. Overall Description.....	10
2.2.1. Product Perspective.....	10
2.2.2. User Classes and Characteristics.....	10
2.2.3. Operating Environment.....	10

2.2.4.	Design and Implementation Constraints .....	11
2.2.5.	Assumptions and Dependencies .....	11
2.3.	External Interface Requirements .....	12
2.3.1.	User Interfaces .....	12
2.3.2.	Hardware Interfaces .....	12
2.3.3.	Software Interfaces .....	13
2.3.4.	Communications Interfaces .....	13
2.4.	System Features .....	14
2.5.	Nonfunctional Requirements .....	17
2.5.1.	Performance Requirements .....	17
2.5.2.	Safety Requirements .....	17
2.5.3.	Security Requirements .....	18
2.6.	Domain Requirements .....	19
Chapter 3	.....	20
<b>Use Case Analysis</b>	.....	20
3.1.	Use Case Model .....	21
Chapter 4	.....	27
<b>System Design</b>	.....	27
4.1.	Architecture Diagram .....	29
4.2.	Domain Model .....	30
4.3.	Entity Relationship Diagram with data dictionary .....	31
4.4.	Class Diagram .....	32
4.5.	Sequence / Collaboration Diagram .....	33
4.6.	Operation contracts .....	34
4.7.	Activity Diagram .....	35
4.8.	State Transition Diagram .....	36
4.9.	Component Diagram .....	37
4.10.	Deployment Diagram .....	38
Chapter 5	.....	40
<b>Implementation</b>	.....	40
5.1.	Important Flow Control/Pseudo codes .....	41

5.1. Components, Libraries, Web Services and stubs.....	54
5.2. Deployment Environment.....	56
5.3. Tools and Techniques .....	56
5.4. Best Practices / Coding Standards .....	56
5.5. Version Control .....	56
Chapter 6.....	57
Testing and Evaluation .....	57
6.1. Use Case Testing.....	58
6.2. Equivalence partitioning.....	58
6.3. Boundary value analysis .....	58
6.4. Data flow testing .....	59
6.5. Unit testing.....	59
6.6. Integration testing.....	59
6.7. Performance testing.....	59
6.8. Stress Testing.....	60
Chapter 7 .....	61
Summary, Conclusion and Future Enhancements .....	61
7.1. Project Summary .....	62
7.2. Achievements and Improvements .....	62
7.3. Critical Review .....	63
7.4. Lessons Learnt.....	63
7.5. Future Enhancements/Recommendations .....	64
Appendices.....	65
<b>Appendix A:</b> Information / Promotional Material .....	66
Reference and Bibliography .....	68
Index.....	70

## List of Figures

1.1	Caption of first figure of first chapter	6
1.2	Caption of second figure of first chapter	7
2.1	Caption of first figure of second chapter	14
2.2	Caption of second figure of second chapter	22
2.3	Caption of third figure of second chapter	26
5.1	Caption of first figure of fifth chapter	49
5.2	Caption of second figure of fifth chapter	49

## List of Tables

1.1	label of first table of first chapter	6
1.2	label of second table of first chapter	7
2.1	label of first table of second chapter	14
2.2	label of second table of second chapter	22
2.3	label of third table of second chapter	26
5.1	label of first table of fifth chapter	49
5.2	label of second table of fifth chapter	49

# Chapter 1

## **Introduction**

# Chapter 1: Introduction

In recent years, NFT Marketplaces only existed at International level. We do not have a platform in Pakistan, through which we can promote our national cultural arts. The motivations and challenges will be discussed in detail for giving awareness of NFT Marketplace in Pakistan. We will discuss the goals and objectives in this chapter, by which we can make our NFT's Marketplace unique in international market as well. We will discuss the gap which we found in existing solutions during our research.

## 1.1. Background

Pakistan is a country of rich cultural heritage with history that dates back to the ancient Indus civilization. However, it struggles to preserve its identity and legacy owing to multiple reasons. Future generations are slowly disconnecting from their rich and unique cultures due to theft of their creative legacies by other outside practitioners. Additionally, exposure to their hereditary crafts and heritage is further reduced due to constant interaction with digital devices and adoption of global or foreign cultures. Indigenous antiques and crafts are often taken away from their place of origin to other locations, thereby denying the rightful heirs to that legacy the opportunity to exposure or interaction with it.

There is a high occurrence of creative communities in Pakistan, with at least one homebased craft having been a common household practice, with knowledge passed down through the generations. With the success of NFTs as artworks globally, and the constant improvement in digital literacy and connectivity locally, it is only a matter of time before the two will meet in the Web 3 space, and NFT MetaMarket aims to be an early entrant and leader in it.

## 1.2. Motivations and Challenges

We are working on the latest Blockchain technology. This technology is mostly used currently in worldwide market and it will be great honor for Pakistan to have its own NFT Marketplace based on this Blockchain technology with AR. This NFT Marketplace will target our local artists and artisans and other people to sell their masterpieces and promote our cultural heritages on this marketplace.

The biggest challenge for us is to integrate AR Technology with the Blockchain technology as it is never used before in any of marketplace. Most of the people in Pakistan do not know the concept of digital wallets and don't know about digital currencies and all these technologies. So, our biggest challenge is to create awareness to the people about these technologies and platforms. It will help to promote our local artist in international market and people of international market will be able to see our cultural aspects. This will make Pakistan stand out in IT industry in worldwide market.

### **1.3. Goals and Objectives**

Our primary goal is to:

- Achieve AR technology integration with our marketplace
- Freelancing opportunities for our local artists and other people worldwide
- Freelancing payment method through Meta Mask wallet
- Create awareness to the people about Blockchain and NFT's and marketplaces □  
Custom creation of NFT's

Our main objective is to promote our local artists and masterpieces and bring Pakistan stand out in IT Industry worldwide.

### **1.4. Literature Review/Existing Solutions**

There are many applications which are providing the facilities for selling and purchasing NFTs worldwide [3]. They are as follow:

- Open Sea
- Axie Market Place
- Larva Labs/Crypto Punks
- NBA Top Shot Marketplace
- Rarible
- Super Rare
- Foundation
- Mintable
- Theta Drop

But the purpose of our application is the global promotion of our local artists, artisans, and our cultural aspects through NFTs and to fill the lack of functionalities in previous marketplaces. Considering the current market of Pakistan, there are no major competitors as of yet.

## **1.5. Gap Analysis**

Previously made NFT Marketplaces only have the option to buy and sell NFT's. There is no platform integrated to create NFT. In addition to this, they do not have AR technology integrated to view NFT in real world view. There are no yearly sales on NFT's on any of these marketplaces. The users do not have access to the custom creation of NFT's in these marketplaces. As there is no platform integrated through which the users can make or create NFT's for someone on their desired features.

## **1.6. Proposed Solution**

NFT Meta Market is building a platform on which artists can create and display their cultural and creative artworks for sale. Additionally, we will be re-creating digital augmented 3D replicas of lost antiques and placing them at their original sites in order to increase locals' exposure to their lost heritage and crafts. We will use Blockchain Technology for this platform so that content is secured for authenticity as well as theft. Buyers will be able to put these artworks in their personal space using Augmented Reality (AR). In addition to these, we will introduce a chat box

## **1.7. Project Plan**

In phase 1, we will create UX-UI design for this marketplace. Phase 2 involves implementation of CRUD Operations in which a user can login into the marketplace using his/her crypto wallet or email, create an NFT and list it on the platform. This process is called "lazy minting" in which the NFT will be stored into our local database without involvement of any Blockchain thus providing a free minting platform.

Phase 3 will be development of smart contracts. This is a crucial stage and must be conducted with utmost care, because once deployed to Blockchain, the code can never be changed and will be responsible for transfer of NFTs as well as cryptocurrency for buying and selling. In Phase

4 we will integrate the application with payment gateways, digital wallets, social media, and other apps in the ecosystem to help growth of our platform.

Phase 5 will be development of the front-end for smart contracts and testing for any bugs and errors. The 6th and final phase will be deployment of the marketplace with front-end and back-end.

### 1.7.1. Work Breakdown Structure

We have a team of diverse individuals who are professionals in their field. Our project contains too many modules, so we split it into several parts and assign these tasks to each one equally. It will make us achieve our goals in less time.

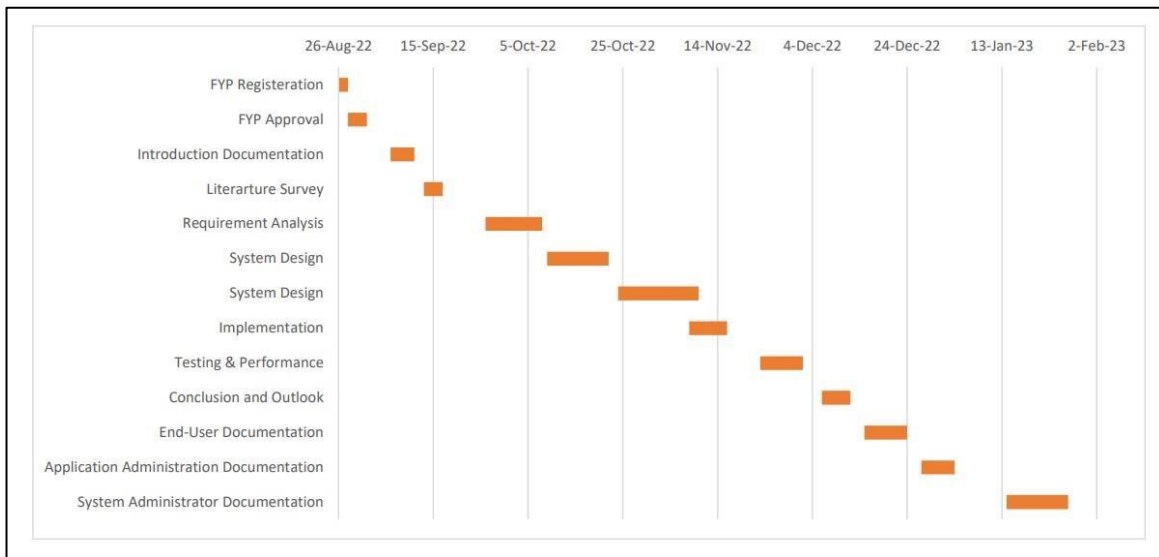
### 1.7.2. Roles & Responsibility Matrix

The purpose of the roles & responsibility matrix is to identify who will do what.

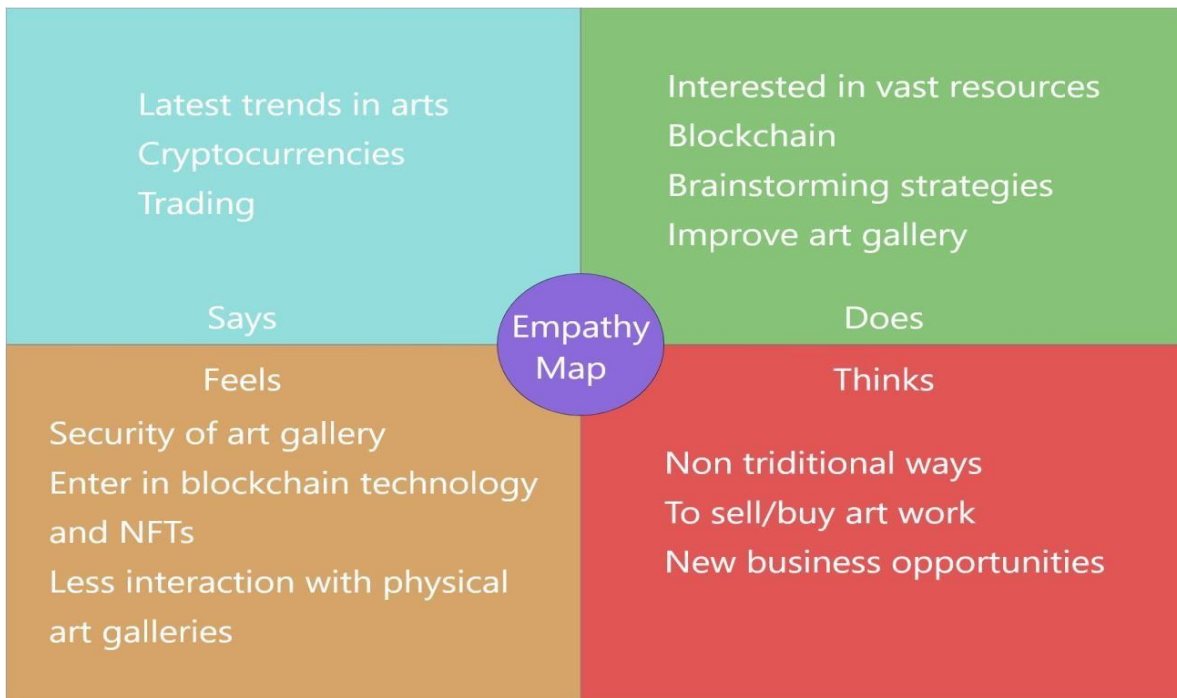
WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Documentation	1	FYP Registration	2	Muhammad Ahmad & Project Idea
1	Documentation	2	FYP Approval	2	Muhammad Ahmad Sohail & FYP Registration Documentation
1	Documentation	3	Introduction Documentation	3	All Team & Documentation
1	Documentation	4	Literature Survey	4	All Team & Research work
1	Documentation	5	Requirement Analysis	2	Ahmad Sohail & Umair Ahmad Requirement Analysis for App
2	UI Design	6	System Design	7	Muhammad Ahmad & UI/UX Design
2	UI Design	7	System Design	7	All Team & Frond End Design
3	AI Integration	8	Implementation	7	All Team Complete Backend

3	AI Integration	9	Testing & performance	3	Umair Ahmad & Testing and Performance report
4	Final Reports	10	Conclusion and Outlook	4	All Team & Work on Testing and outlook of App in market
4	Final Reports	11	End-User Documentation	2	All Team & Check Reviews of End User
4	Final Reports	12	Application Administration Documentation	3	Muhammad Ahmad & Sohail & Finalize Document
4	Final Reports	13	System Administrator Documentation	3	Muhammad Ahmad Sohail, Umair Ahmad & Documentation

### 1.7.3. Gantt Chart



## 1.9. Empathy Map



# Chapter 2

## Software Requirement Specifications

## Chapter 2: Software Requirement Specifications

### 2.1. Introduction

#### 2.1.1. Purpose

The document defines the understanding of the general objectives for the creation of the project “*NFT MetaMart* – a decentralized platform - (hereinafter - the software product, System, Platform, “NFT MetaMart”). The document sets out the main requirements for the software product, defines the scope of this product, the purpose of implementation, restrictions and strategic decisions on project implementation, agreed between the Customer and the Developer. The Software Requirement Specification document is the basis for the software product development process.

#### 2.1.2. Document Conventions

The conventions used in this document are as follow:

NFT	Non – Fungible Tokens
Style: <i>italic</i>	Project Name
VSCode	Visual Studio Code – Text Editor

#### 2.1.3. Intended Audience and Reading Suggestions

This project is a prototype for the ‘*NFT MetaMart*’ and it is restricted within the university premises. This has been implemented under the guidance of university professors.

This project is useful for the NFT creators, sellers and as well as NFT buyers.

#### 2.1.4. Product Scope

The purpose of the “*NFT MetaMart*” is to promote the artists, artisans of Pakistan and many other people and their masterpieces in the international market. This system is based on NFT creation of their masterpieces and artwork and sell in the worldwide market to promote them. We will have a connection supporting different kinds of wallets that will create ease for

everyone to use this marketplace. Above all, we hope to provide a comfortable user experience along with the best pricing available.

### **2.1.5. References**

- 1- (Open Sea Marketplace) <https://opensea.io/> accessed from 23/11/2022
- 2- (Rarible Marketplace) <https://rarible.com/> accessed from 23/11/2022
- 3- (Top NFTs Marketplaces) <https://www.fool.com/the-ascent/cryptocurrency/nft-marketplaces/> accessed from 23/11/2022
- 4- (Introduction to Blockchain) <https://www.ibm.com/in-en/topics/what-is-blockchain> accessed from 23/11/2022

## **2.2. Overall Description**

### **2.2.1. Product Perspective**

“NFT MetaMart” is a product that requires some additional hardware and software interfaces to function which includes the OS, a web browser, a cryptocurrency wallet (Metamask/WalletConnect/Trust Wallet), and a stable internet connection. When released, the final product would be the first version of the application. It will be designed as a user centered product, which could be accessed to give a personalized experience to any authenticated user.

### **2.2.2. User Classes and Characteristics**

The intended users of this application will be people who are inclined towards finance, music and the Blockchain sector. Other people who perform cryptocurrency/NFT trading on a regular basis can also benefit from the application.

The users are expected to be Internet literate and be able to understand the concept of NFTs and how the Metamask and Auction system works. They should be able to trade their cryptocurrency in exchange for the NFT.

### **2.2.3. Operating Environment**

#### **2.2.3.1. Software Environment**

NFT MetaMart runs on smart contracts. Each NFT on NFT MetaMart contains specific, original information encrypted and stored in a contract on the Ethereum Blockchain, making it secure and immutable. These contracts run together through Blockchain technology and are self-

generating according to their potential. Users must have their digital wallet (Metamask/Trust Wallet etc.).

For example, the buyer sends cryptocurrency for the NFTs he chooses, and the seller should grant him the right to own them. If all parties meet the requirements, the smart contract handles the purchase and sale separately.

### **2.2.3.2. Hardware Environment**

*NFT MetaMart* can be efficiently run on:

#### **1- Desktop/PC**

**RAM:** 4GB

**Version:** i3 or above

**Storage:** 8GB

#### **2- Android**

**Android Version:** 7.0 or above

**RAM:** 3GB or above

**Storage:** 4GB

#### **3- IOS**

**Version:** v7.0 or above

**RAM:** 2GB or above

**Storage:** 8GB

### **2.2.4. Design and Implementation Constraints**

A max-width of 1460px would be applied to the website contents. So, users on extralarge screens will have to zoom in a little to view the website correctly.

For normal screen users, the website contents are obvious all components are also appropriately shown in the mobile view.

### **2.2.5. Assumptions and Dependencies**

- A web browser is required to access the website. □ Basic knowledge of cryptocurrencies and Blockchain □ Users must be connected to the Internet.
- Users must know how to operate Metamask.

- There are some dependencies on 3<sup>rd</sup> party software like Metamask, IPFS and Font Awesome

## 2.3. External Interface Requirements

### 2.3.1. User Interfaces

The product website will be made very simple and responsive, thus having a UI that is easy to access. It we can run on any device with a modern web browser but mainly optimized for desktop users.

Though the UI will be developed to look a little bit different on desktop and mobile devices, most of the functionalities will be available to both types of users.

Content on extra-large screens will be displayed the same as on the desktop as a maxwidth of 1460px will be set for the website contents.

Required web pages include:

- **Static Pages:** To portray the functioning of the platform.
- **Marketplace/Library:** Page to display all NFTs.
- **Individual NFT page:** Separate page that displays data and characteristics associated with the particular NFTs.
- **User/Artist dashboard:** User dashboard to display NFTs owned by the user and his/her details.
- **Create NFT page:** A page displaying a form so the artists/artisans and many other people can create NFTs.
- **Auction Page:** A page to place bids by the bidder on his/her favorite NFTs.
- **Contact Us Page:** A page to send queries to the admin of the website.
- **FAQ and Terms and Condition Page:** A page listing the frequently asked questions and Terms and Conditions of the application.

### 2.3.2. Hardware Interfaces

- **RAM:** 4GB
- **Storage:** 8GB

- **Processor:** i3 or above

#### **2.3.2.1. Frontend**

There is no specified screen resolution for accessing the website. The website can be used on any screen size of any desktop or mobile device.

#### **2.3.2.2. Backend**

The smart contracts will act as a background for this application. The logic will be written in Solidity Programming language and deployed on Ethereum Blockchain.

### **2.3.3. Software Interfaces**

The application will be implemented in Next and Solidity. VSCode will be used as a text editor. Hardhat Framework will be used to test, compile and deploy smart contracts. Git will be used for version control and GitHub will be used for hosting the repository. The software requirements for the development of the application include:

- Hardhat framework
- Next JS for frontend
- VS Code as a text editor
- Git and GitHub for version control
- HTML
- CSS
- JavaScript to write tests
- Solidity to write smart contracts
- Adobe Suite(XD, Photoshop, Illustrator) for Designing

### **2.3.4. Communications Interfaces**

The system requires HTTPS to communicate with the database. The system and the database can be configured to be accessed via any available port. The web-based UI is the only means of communication between the user and the system. Also, the user can directly send an email to the admin in case of any queries through **Contact US** page. The system is accessible through all popular modern web browsers that interact with HTML pages and access Metamask.

## 2.4. System Features

*NFT Meta Mart* is responsible for functionalities:

- User can create account
- User can buy NFTs
- User can sell NFTs
- User can create his/her own NFT
- User can list his/her favorite NFTs
- User can view his/her NFTs
- Bidding on NFTs
- All functionalities will be handled through Smart Contracts and Ethereum Blockchain
- Transactions will be held through digital wallets (Metamask/Trust Wallet etc.)

### 2.4.1. Authentication Module

#### 2.4.1.1. Description and Priority

This feature will validate the User with his/her credentials. After registration user can login into the system. This feature is of high priority as it validates the user along with digital wallet connectivity and gives the user an identity.

#### 2.4.1.2. Stimulus/Response Sequences

User: (goes to **Sign Up** page)

System: (action timed out 0.52 seconds)

User: (fills all his/her credentials with any digital wallet account address)

System: saves all the credentials to the database and verify the digital wallet account

(action timed out 0.833 seconds)

System: redirected to the **Login** page (action timed out 0.2 seconds)

User: (fills all his mentioned credentials)

System: checks whether the User data is in record or not. Then, it gives/not gives access to the user upon his/her registration and connects to the digital wallet (action timed out 0.92 seconds)

#### 2.4.1.3. Functional Requirements

The Authentication involves 2 steps:

- Sign Up for new users
- Login for existing users
- In order to Sign up, the user/artist needs to fill the registration form with the required information.
- If registering as seller, then a seller ID is created
- Email verification is carried out to verify the user/artist.
- Users/Artists get connected to their Metamask wallet automatically after successful login in.
- All the information of the User/Artist is stored in a User database.
- Spectator needs to perform the Metamask login to make the browser web3 enabled, but a Buyer has to login by entering the registered email-ID and password.
- If successful, User/Artist can access the application, otherwise he/she will be redirected to the login page with the error message.

REQ-SF1-1: Username

REQ-SF1-2: Password

REQ-SF1-3: Digital Wallet Account Address

## 2.4.2. Create NFT Module

### 2.4.2.1. Description and Priority

In this feature, the user can create his/her own NFT by going to Create NFT page. Here, he/she can upload file of his/her masterpiece and then by clicking on Create button, system will create his/her NFT.

### 2.4.2.2. Stimulus/Response Sequences

User: (goes to Create NFT Page)

System: (action timed out 0.32 seconds)

User: (uploads file for NFT)

System: (action timed out 1.23 seconds)

User: (presses on Create button)

System: creates the NFT of the user (action timed out 0.823 seconds)

### **2.4.2.3. Functional Requirements**

The verified Artist can create an NFT of his masterpiece by visiting the Create Page on the application.

The artist needs to fill in the required fields for creation of the NFT and upload the file of the masterpiece for which he will also pay a gas fee for storing the NFT on the Blockchain (i.e. Ethereum Network).

REQ-SF2-1: File (of his/her masterpiece) for NFT

## **2.4.3. Library Page Module**

### **2.4.3.1. Description and Priority**

In this feature, the user can see the listing of his/her NFTs, favorite NFTs, and all the details relevant to his/her NFT. This feature is of high priority as it shows all the records of NFT relevant to the user.

### **2.4.3.2. Functional Requirements**

- Users/Artists will be able to see the “Trending NFTs” and the “Recently added NFTs” on the Library Page.
- Various functionalities like Searching, Sorting are provided for quick access to the NFTs.
- Various filters options are also available like Genre, Lyrics, Instrument used are also available on the Library Page.
- Users/Artists can visit the NFT detail Page by clicking on a specific NFT for more details.

## **2.4.3. Auction System Module**

### **2.4.3.1. Description and Priority**

In this feature, the user can place a bid for an NFT. The bid is validated and generate message to the winner. This feature is of medium priority as it depends upon the user’s choice to bid an NFT or not.

### **2.4.3.2. Functional Requirements**

- Buyer can place a bid for an NFT by visiting the Auction Page on the application.

- The bid is validated through an Auction database and stored in it with the specified details of NFT and the user.
- After the bidding process is over, the max bid is chosen from the database.
- The user details of the winning bid are processed and a winning message is displayed to the Bidder and the NFT seller/owner.
- Transaction request is generated for buyer; payment is processed and communicated through Blockchain.
- After payment is processed, the ownership of NFT is transferred to the seller.
- In case, an NFT is sold again, a part of the transaction is transferred to the seller as royalty.

## **2.5. Nonfunctional Requirements**

### **2.5.1. Performance Requirements**

- System should respond and generate output within 10 seconds when the user interacts with the system and Blockchain, else a message defining the error should be shown.
- Digital Wallet is required to be connected to the Blockchain via web browser.
- After successful connection to Blockchain, NFT should be displayed within 5-10 seconds as loading data from Blockchain is slow.

### **2.5.2. Safety Requirements**

The application is having the following threats:

- Phishing
- Selling fake works
- Converting real artworks into NFT without the artist's consent
- Smart contract vulnerabilities
- Cyber security and NFT fraud

Following are the preventions that should be taken:

- ✓ Perform smart contracts audit
- ✓ Adhere strictly to the decentralization principle
- ✓ Instruct our users on the necessary security measures

### **2.5.3. Security Requirements**

The server on which the application resides will have its own security to prevent unauthorized write/delete access. There is no restriction on read access. The system on which the user uses the application will have its own security.

There is no special protection built into system other than the information of the users shall be encrypted before transferred to the database in order to maintain the security of the crucial information about the users.

### **2.5.4. Usability Requirements**

- The users should be provided with appropriate and correct NFT data by the system. Any false information displayed would lead to user dissatisfaction.
- All the contents of the application should be run smoothly.

### **2.5.5. Reliability Requirements**

The application should deliver appropriate and correct NFT data to the users. Any false information displayed would result in dissatisfaction from the users.

NFT Info Page should be updated with the latest price of NFT.

Auction Page should be displayed the correct and latest information to the Buyers regarding bids on the respective NFT.

### **2.5.6. Maintainability/Supportability Requirements**

Maintainability of the application would be a little hard sometimes as there are several dependencies like IPFS, Metamask etc. and the tools being used are continuously being updated by the developers. So, frequent checks would have to be made to ensure proper and desired functioning of the application.

### **2.5.7. Portability Requirements**

- Our application must be portable in two ways:
- It must transfer NFT from one account to another account through digital wallet authentication.
- It must be compatible with almost all devices to ensure its portability. So, anyone can access it any time at any place.

### **2.5.8. Efficiency Requirements**

- There is no difficulty in the connectivity of any Digital Wallet.
- All the transactions run smoothly and properly.
- The loading time of our application should be minimized.
- User Authentication, create NFT and all the features should be run smoothly and properly.

### **2.6. Domain Requirements**

Following are the Domain requirements for our application:

#### **2.6.1. Database**

Firebase

#### **2.6.2. Frontend**

Next.js

#### **2.6.3. Backend**

Ethereum Blockchain

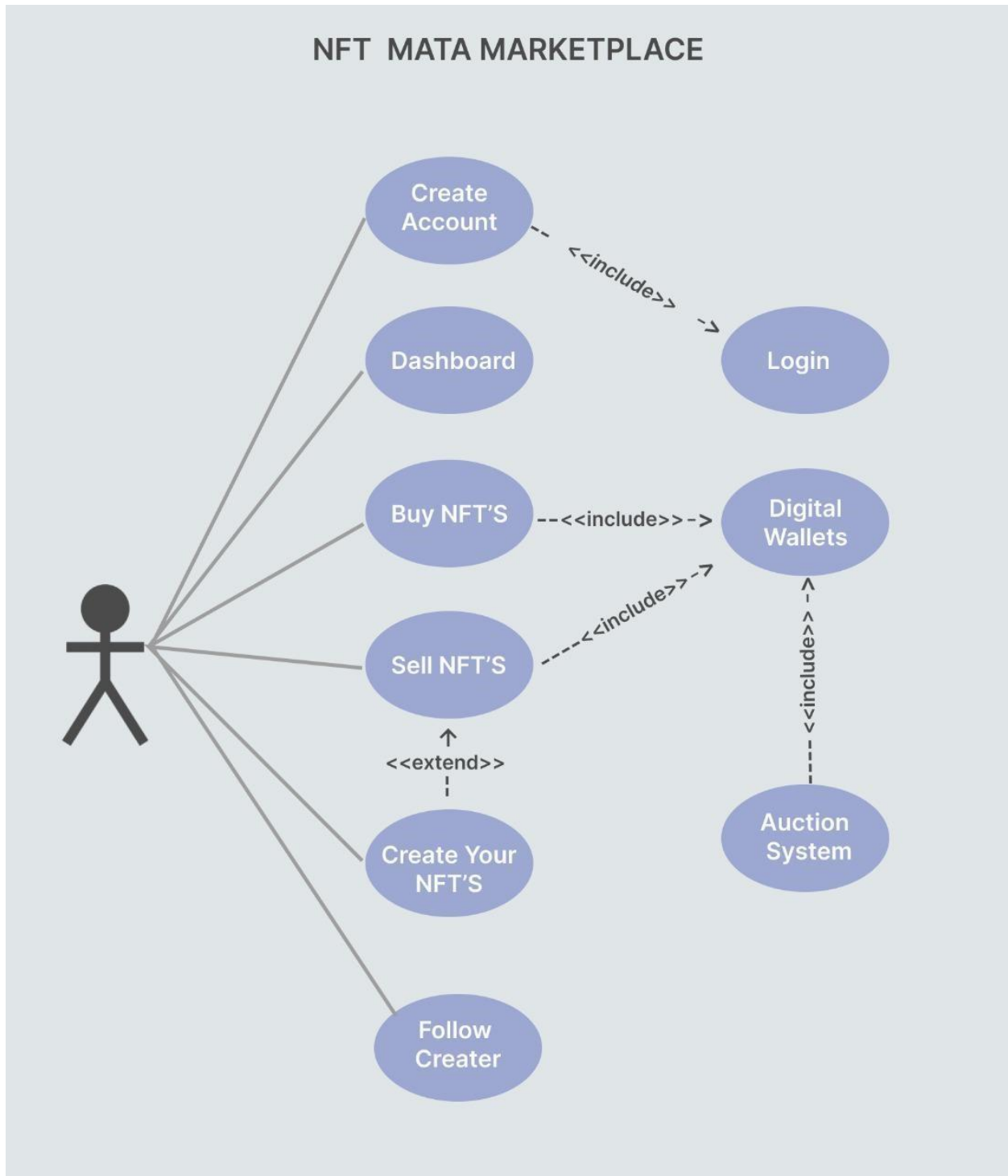
#### **2.6.4. Server**

Center SERV

# Chapter 3

## Use Case Analysis

### 3.1. Use Case Model



## **Use Cases Description**

### **Create Account / Sign Up**

#### **1.1 Brief description**

This use case describes how the users can Sign up in NFT Meta Mart.

#### **1.2 User**

The NFT Buyer & NFT Creator both are users in this use case.

#### **1.3 Pre-condition**

The user should first signup before creating, buying, and minting NFT.

#### **1.4 Flow of events**

##### **1.4.1 Basic flow**

- The system will prompt the user to enter the Name, Email id, and Password or connect through Meta mask.
- The user will enter the above details asked by the system.
- A verification email will be sent to the provided Email id.
- The user will click on the verification link which is enclosed in the received email.
- The system will store the data in the user database.

##### **1.4.2 Exception flow**

- If in the basic flow, the user enters an email that was already present in the database, an error will occur that the email already exists kindly use another email.
- If the user does not click on the verification email within 30 minutes, the link will be disabled and the whole sign-up process would have to be started from scratch.

#### **1.5 Post-condition**

A message stating the successful creation of an account should be displayed and the user should be redirected to the Sign In page.

## **2. SIGN IN / Login**

### **2.1 Brief description**

This use case describes how the user can Sign in to NFT Meta Mart.

### **2.2 User**

The NFT Buyer & NFT Creator both are users in this use case.

### **2.3 Pre-condition**

The user should be present on the sign-in page.

### **2.4 Flow of events**

- The system will prompt the user to enter the Email id, and Password and after that connect their Meta mask wallet.
- The user will enter the above details asked by the system.
- Provided credentials would be checked in the database in case of email sign-in, and if present the user should be signed in successfully.
- For connection through Meta mask, user details are checked by Meta mask servers and if present the user should be signed in successfully.
- After that a Welcome Back message will pop up on the screen.
- If in the basic flow, the user enters an email or password that is not present in the user database or the user detail is not there on Meta mask servers, then an error will occur that the user is not found.

### **2.5 Post-condition**

The user should be redirected to the dashboard.

## **3. CREATE NFT**

### **3.1 Brief description**

This use case describes the Create Page on which the user can create NFTs through Ethereum Blockchain.

### **3.2 User**

The NFT Buyer & NFT Creator both are users in this use case.

### **3.3 Pre-condition**

The user should have signed up and completed the verification process along with the Meta mask wallet connection.

### **3.4 Flow of events**

#### **3.4.1 Basic flow**

- The user will be asked to fill out a form specifying detailed information regarding the NFT.
- After that, the user needs to agree to the terms and conditions and click the Submit button.
- Users have to pay the gas fee for creating the NFT and storing it on the blockchain.

#### **3.4.2 Exception flow**

- If any problem occurs during NFT creation then an error message will occur stating as the operation failed.

### **3.5 Post-condition**

The NFT creator/seller must then be able to access the new data through the application.

## **4. Digital Wallets**

### **4.1 Brief description**

This use case describes digital wallet connectivity, in which the user will be able to transfer the amount to buy any type of NFT.

## 4.2 User

The NFT Buyer & NFT Creator both are users in this use case.

## 4.3 Pre-condition

The users should have already created their meta mask wallet.

## 4.4 Flow of events

### 4.4.1 Basic flow

- The user will be asked to fill out a form specifying detailed information regarding the NFT.
- After that, the user needs to agree to the terms and conditions and click the Submit button.
  
- Users have to pay the gas fee for creating the NFT and storing it on the blockchain.

### 4.4.2 Exception flow

- If any problem occurs during NFT creation then an error message will occur stating as the operation failed.

## 4.5 Post-condition

The NFT creator/seller then be able to access the new data through the application.

# 5. AUCTION SYSTEM

## 5.1 Brief description

When the user will search for the Auction Page, he/she will be redirected to the Auction dashboard where the NFTs are listed and the user can bid for their favorite NFTs.

## 5.2 User

The User and the User are the users in this use case.

## 5.3 Pre-condition

The users should be signed in.

- The user can view the NFT dashboard after searching for the particular NFT name or User name.
- The NFT/Song Info Page will then be displayed for the particular NFT.
- Song's name, description, and associated details like Creator, description, instrument used, Genres, lyrics etc. will be displayed.
- A quick summary table with metrics like highest bid, no. of bids, average bids etc. will also be shown.
- Options to place the bids will be available for the user.
- The user can also view the NFT/Song dashboard by clicking on the bookmarked items on the user dashboard.

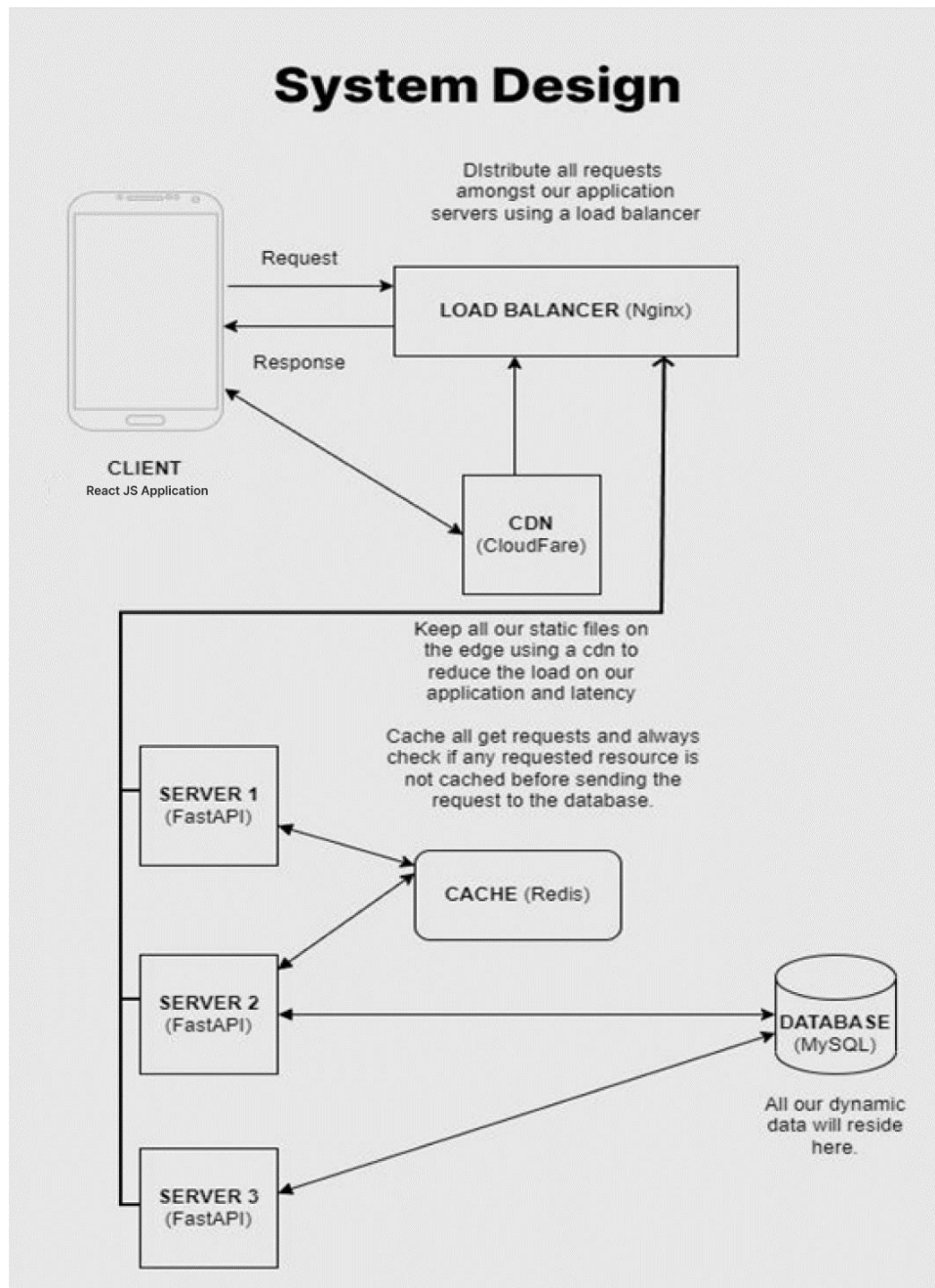
## 5.5 Post-condition

The user should be able to place the bids on stand for the NFTs.

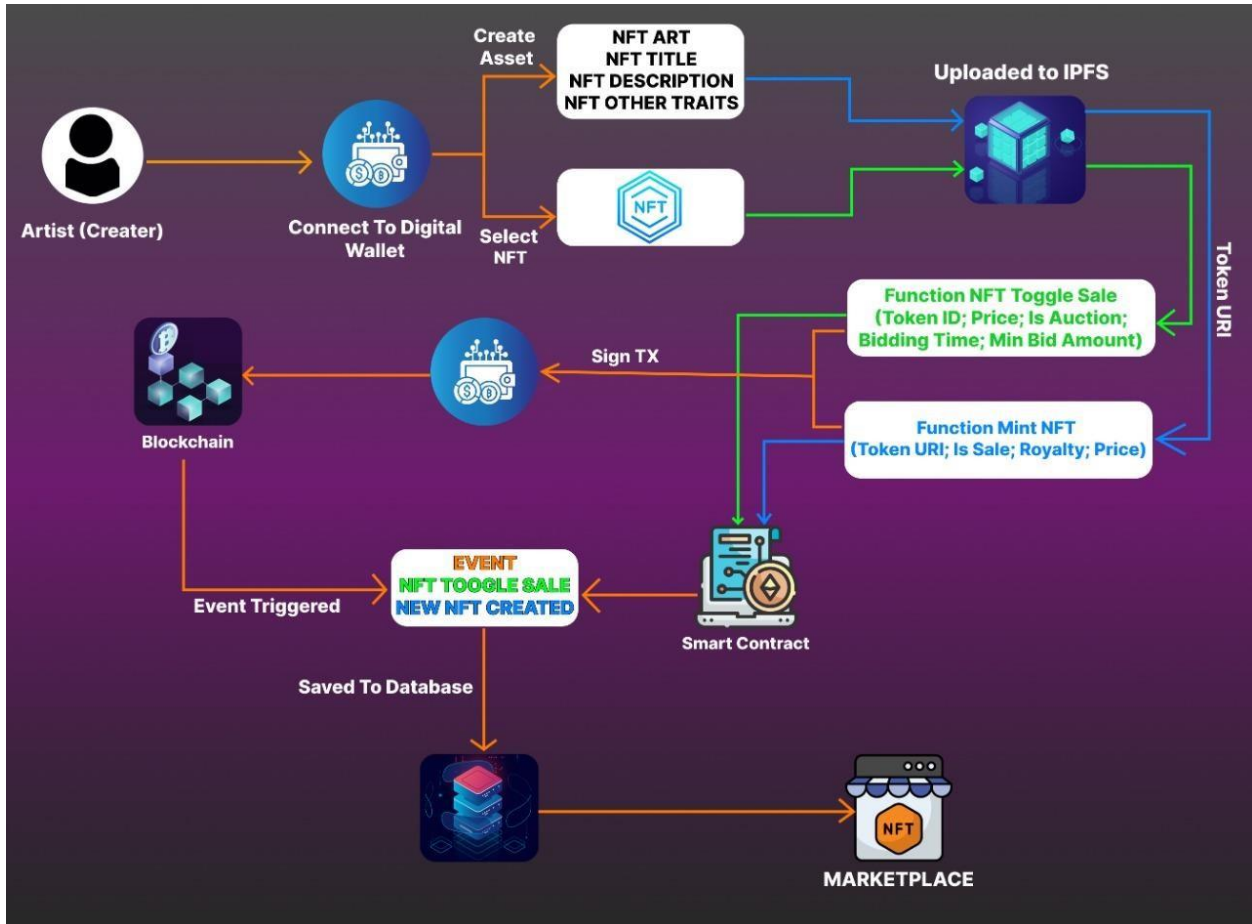
# Chapter 4

## System Design

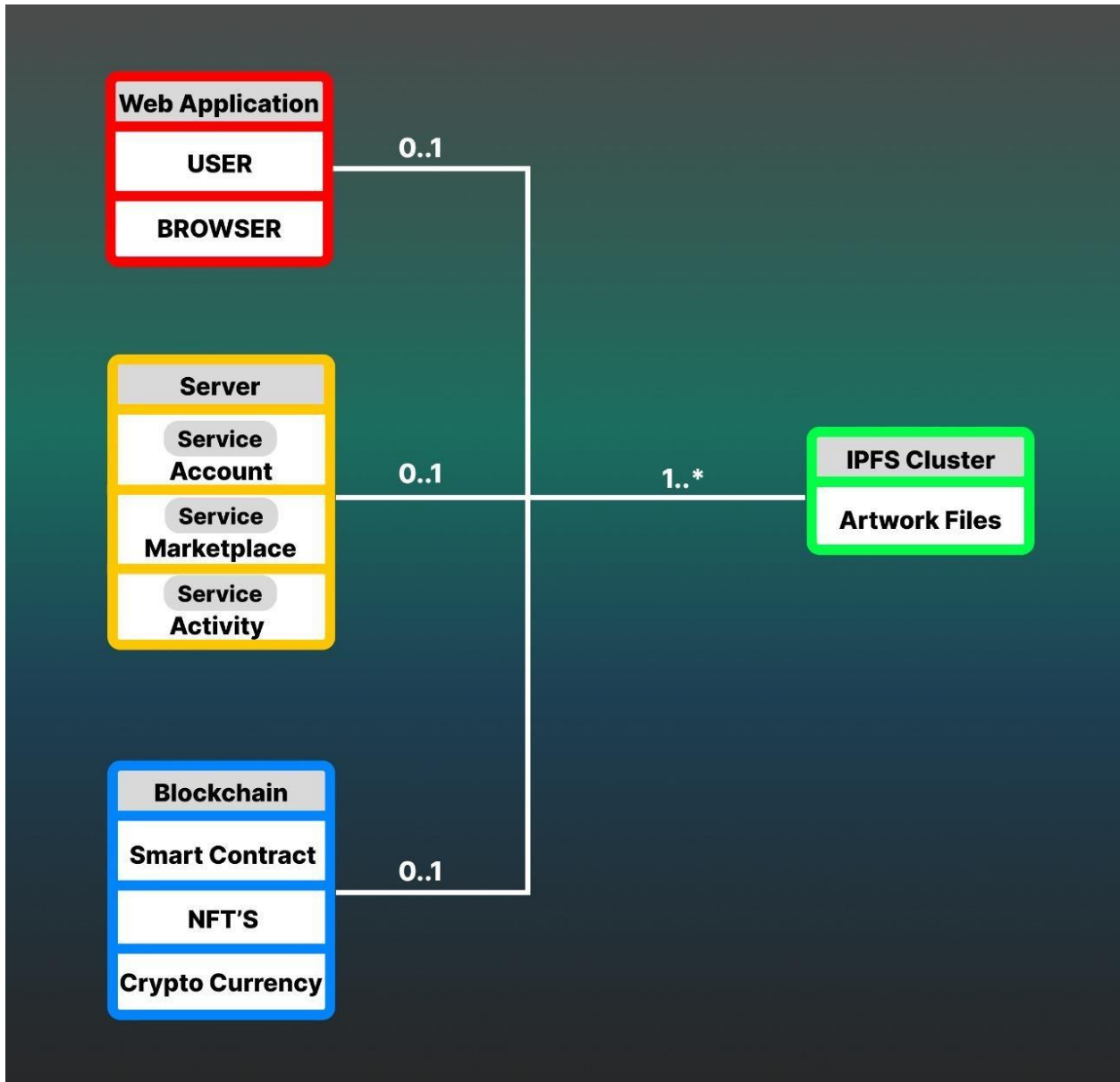
## Chapter 4: System Design



### 4.1. Architecture Diagram

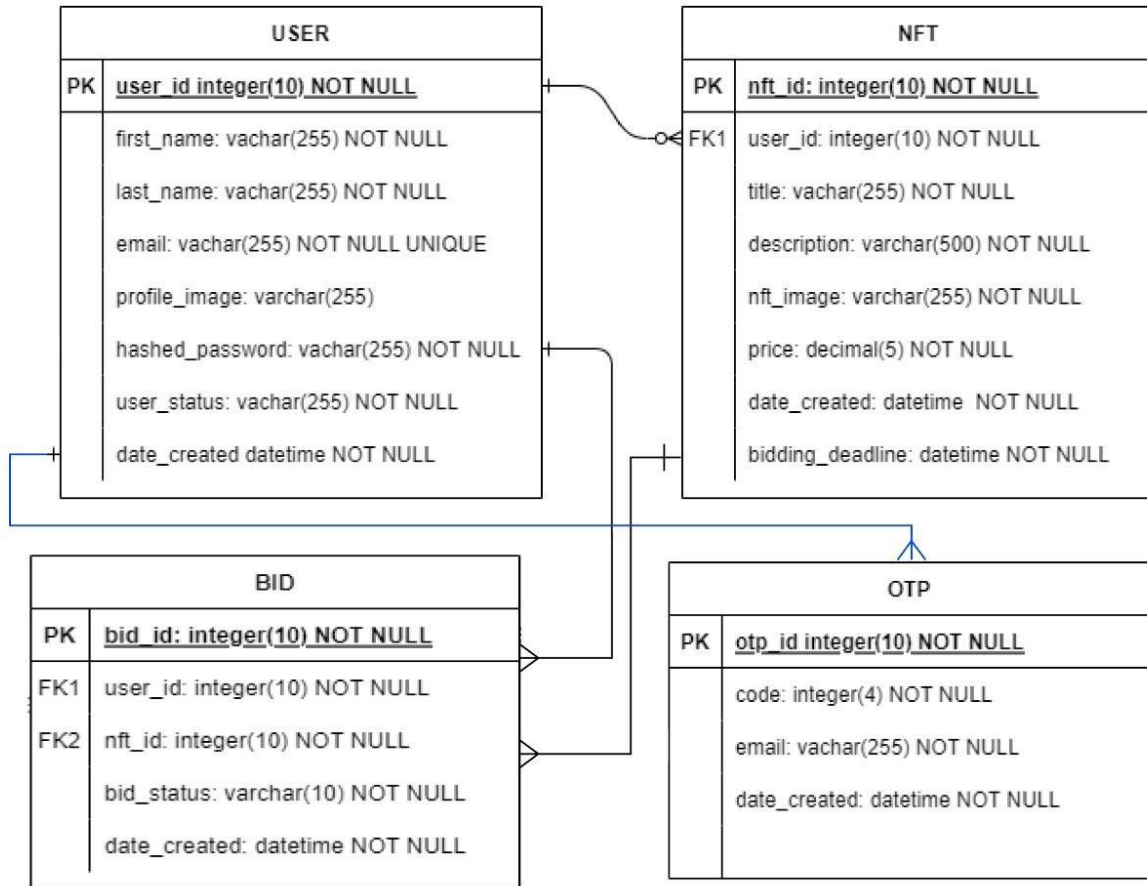


## 4.2. Domain Model

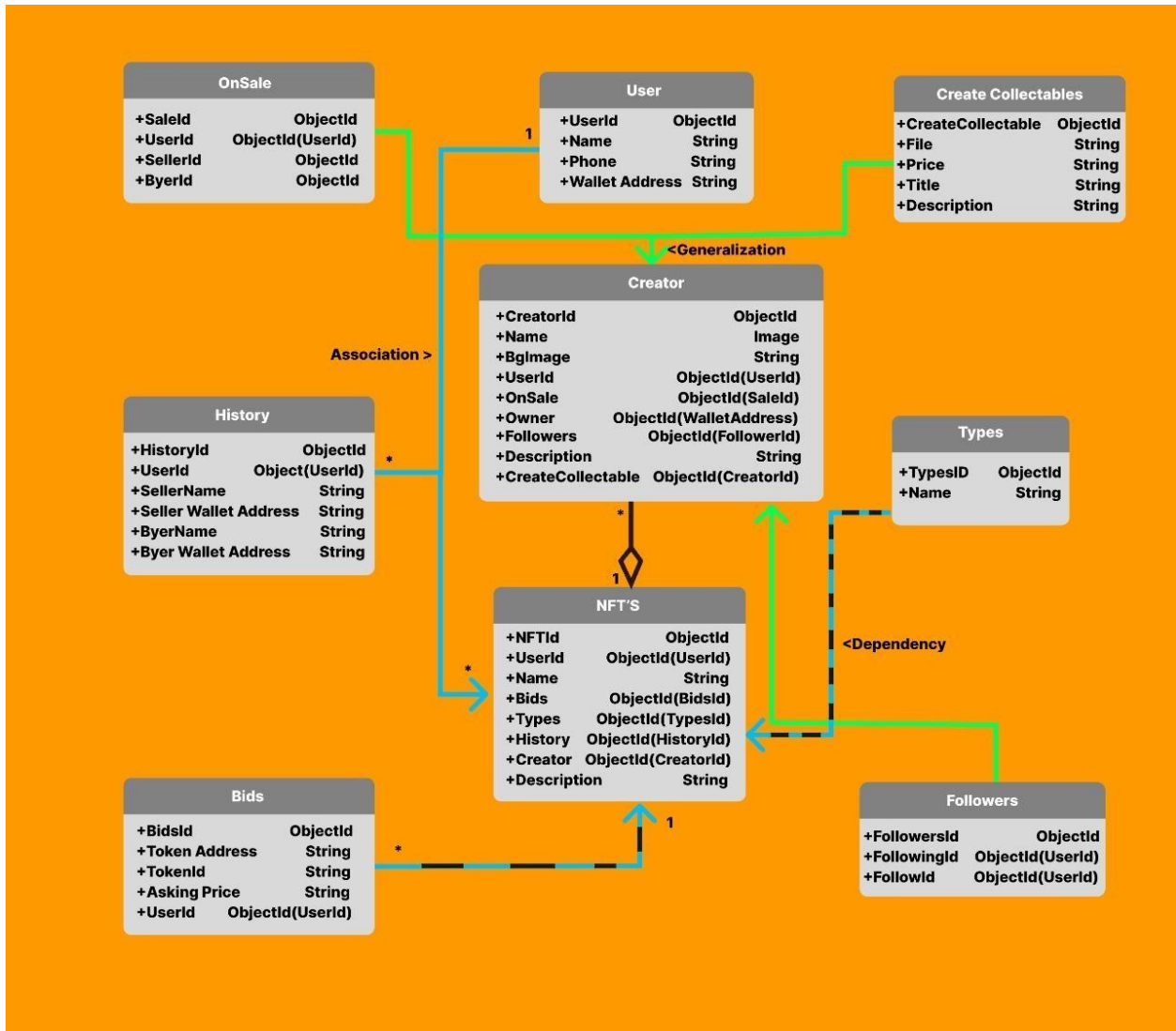


### 4.3. Entity Relationship Diagram with data dictionary

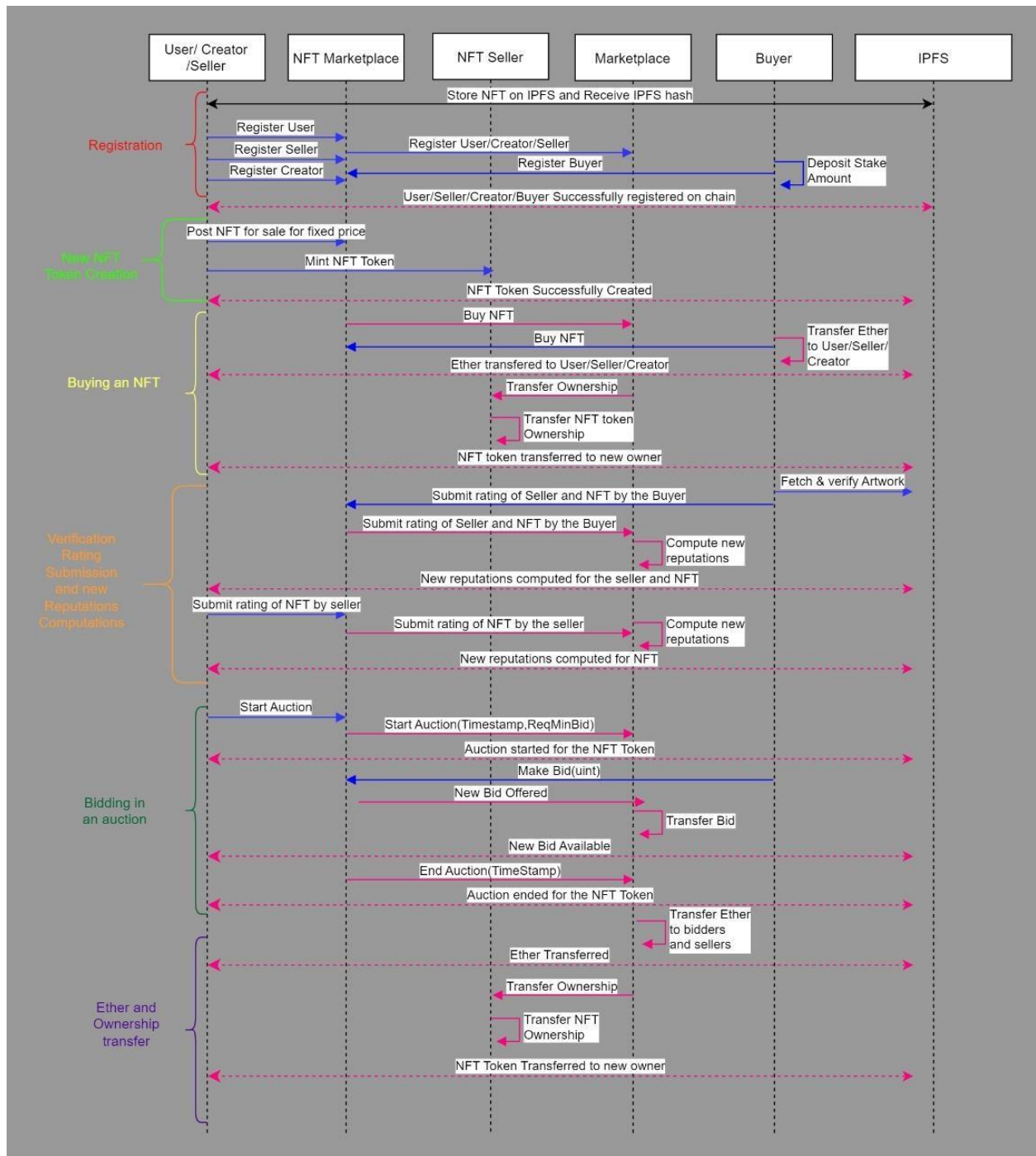
NFT MARKET PLACE ERD



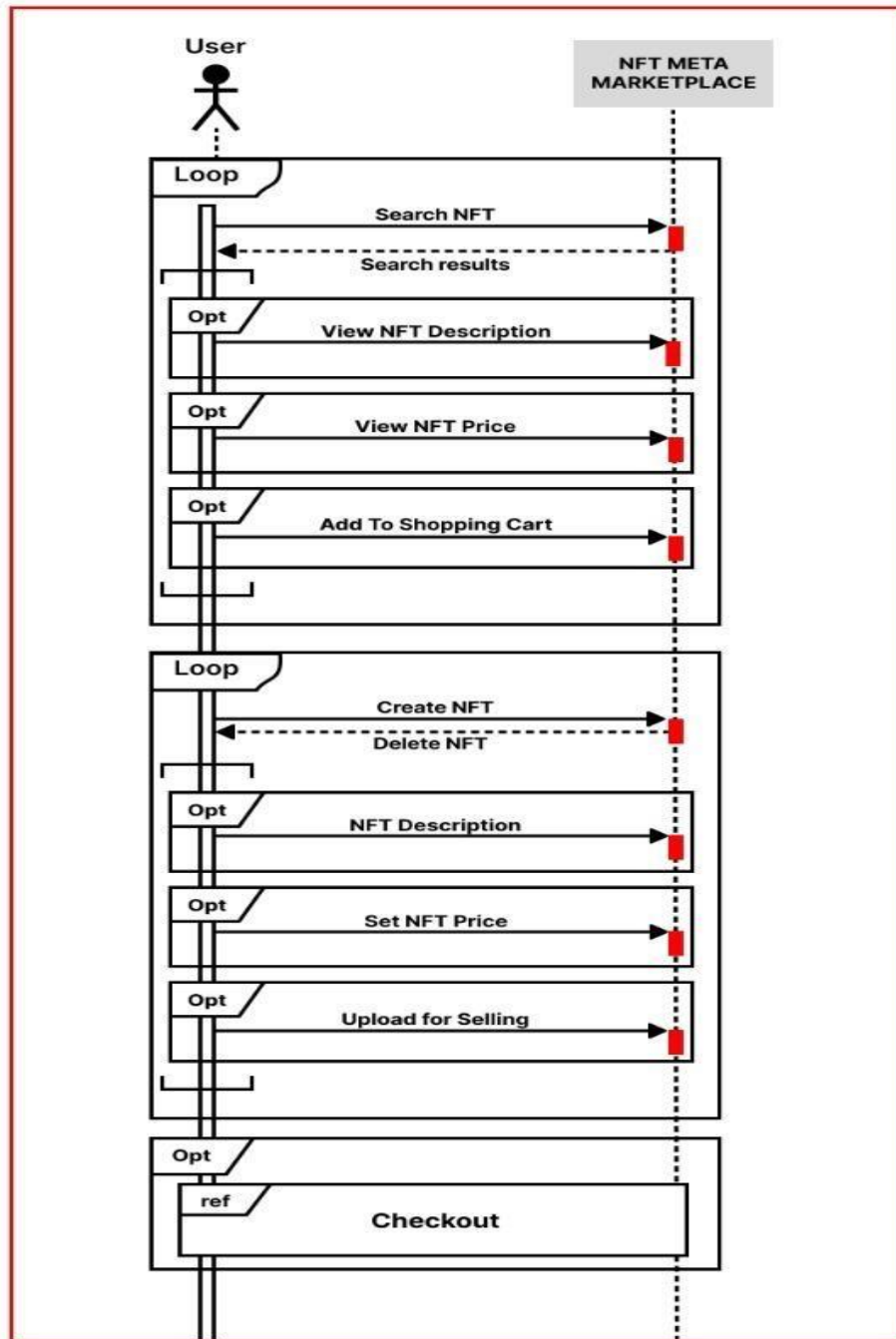
### 4.4. Class Diagram



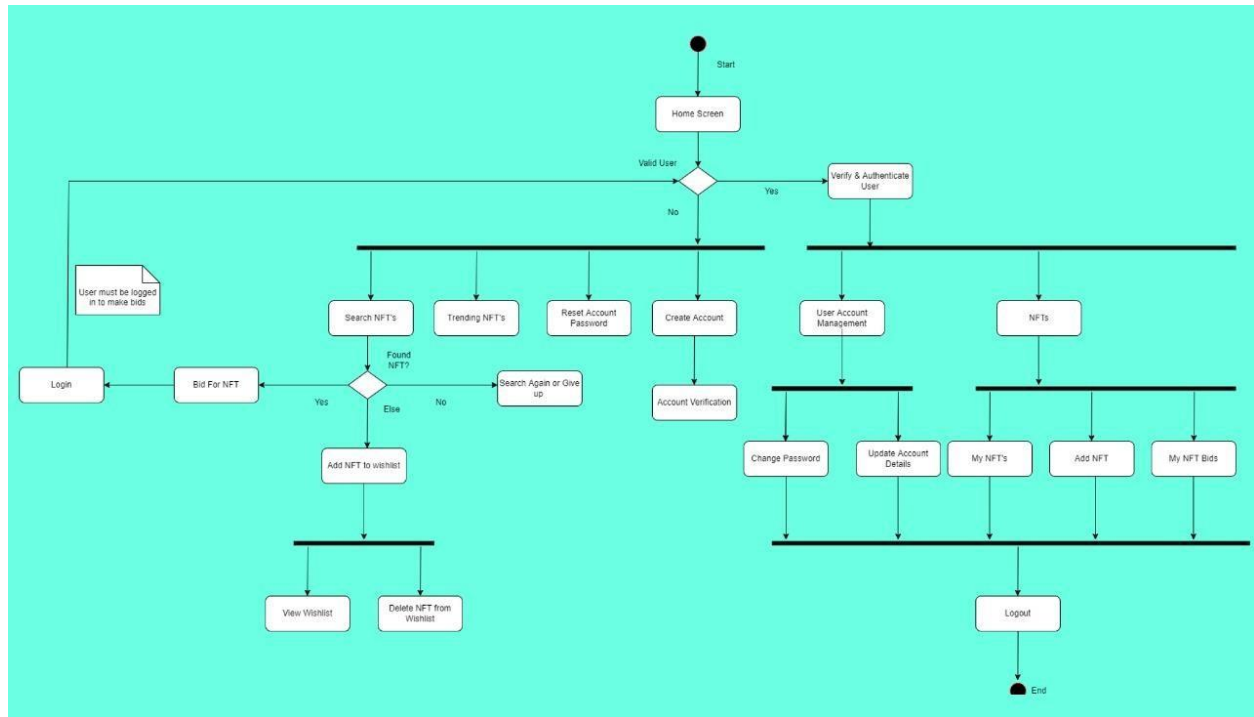
## 4.5. Sequence / Collaboration Diagram



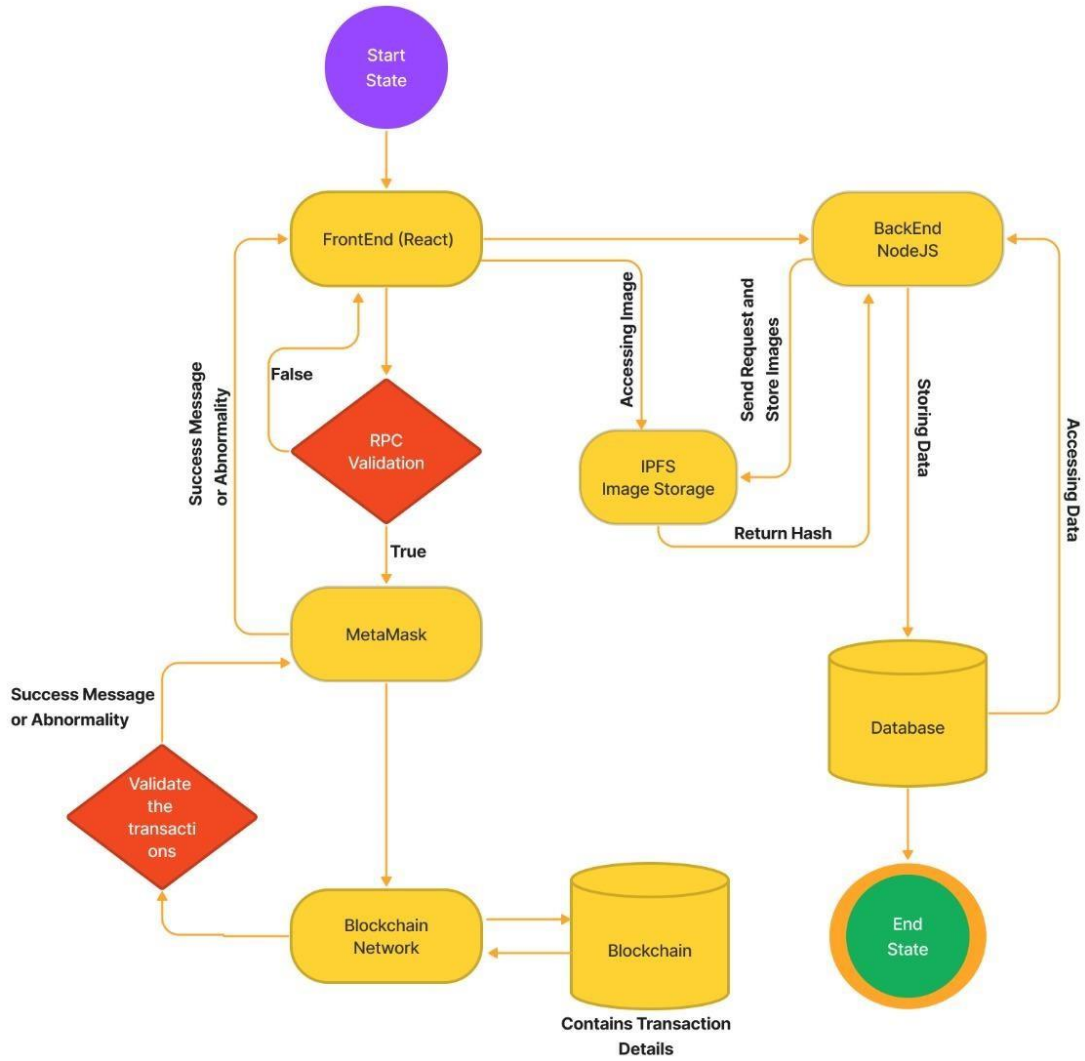
## 4.6. Operation contracts



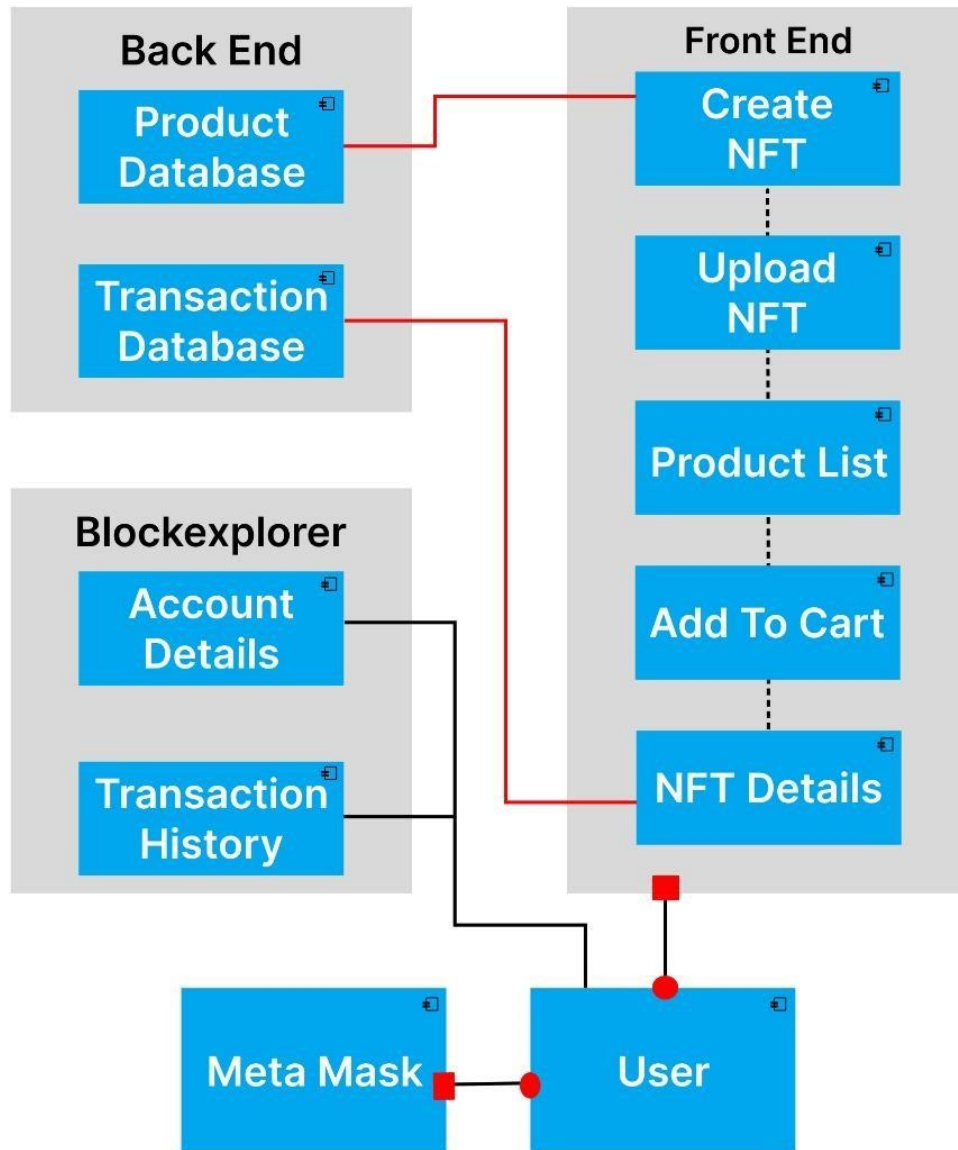
## 4.7. Activity Diagram



### 4.8. State Transition Diagram

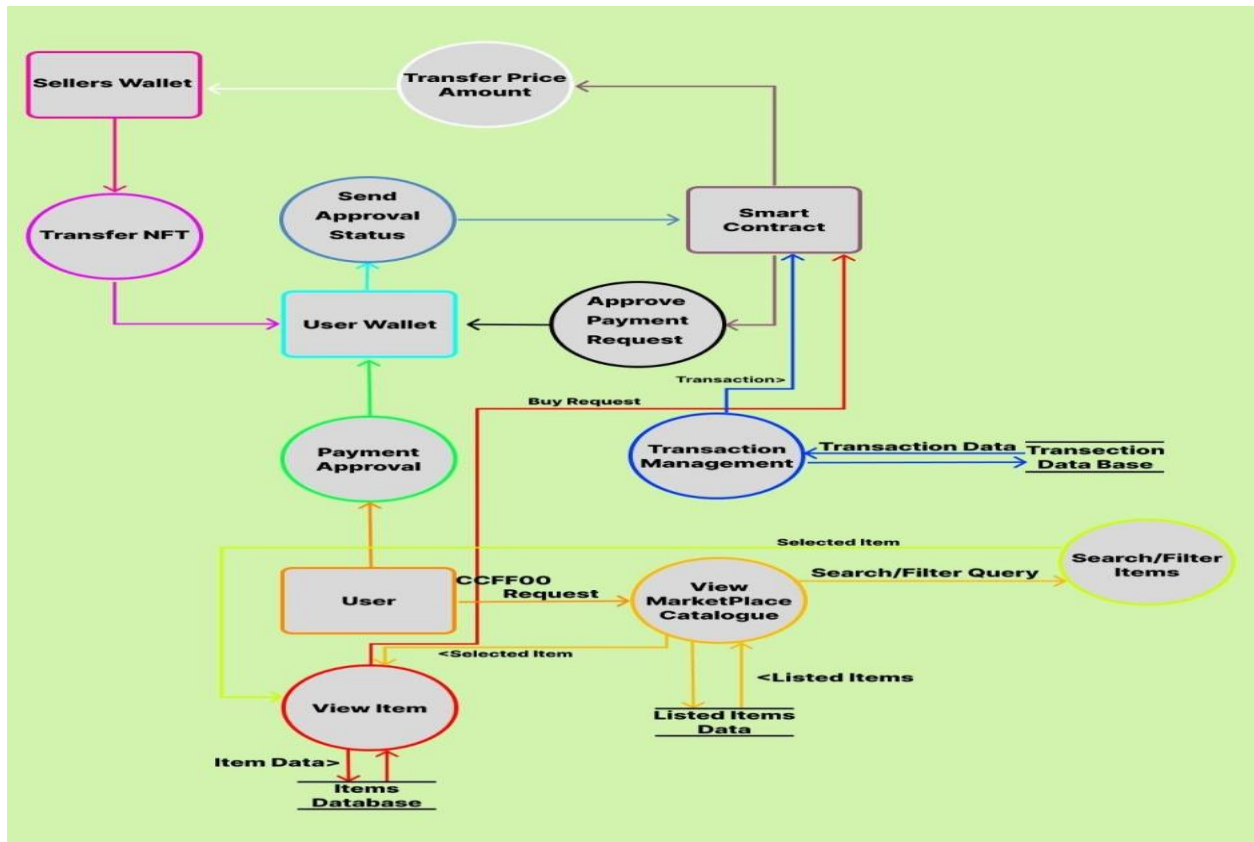


## 4.9. Component Diagram





### 4.11. Data Flow diagram



# Chapter 5

## Implementation

## Chapter 5: Implementation

In this chapter, we will discuss the implementation of the NFT Meta Market, it involves two major steps:

- **Backend:**

It involves **Hardhat** configuration with smart contracts and compilation and deployment of smart contracts.

We have used Ethereum Blockchain as it is widely used Blockchain network and has huge support.

Also, we have Login and Signup functionality with **MongoDB** in our Marketplace for having a record of the users using our platform.

- **Frontend:**

It involves the design which is implemented in **Next.js** with inline, tailwind and simple CSS.

### 5.1. Important Flow Control/Pseudo codes

**NFT Marketplace.sol:**

```

4. // SPDX-License-Identifier: MIT
5. pragma solidity ^0.8.7;
6.
7. import "@openzeppelin/contracts/token/ERC721/IERC721.sol";
8. import "@openzeppelin/contracts/security/ReentrancyGuard.sol";
9.
10. // Check out https://github.com/Fantom-foundation/Artion-Contracts/blob/5c90d2bc0401af6fb5abf35b860b762b31df02/contracts/FantomMarketplace.sol
11. // For a full decentralized nft marketplace
12.
13. error PriceNotMet(address nftAddress, uint256 tokenId, uint256 price);
14. error ItemNotForSale(address nftAddress, uint256 tokenId);
15. error NotListed(address nftAddress, uint256 tokenId);
16. error AlreadyListed(address nftAddress, uint256 tokenId);
17. error NoProceeds();
18. error NotOwner();
19. error NotApprovedForMarketplace();

```

```
24.
25.     contract NftMarketplace is ReentrancyGuard {
26.     struct Listing {
27.     uint256 price;
28.     address seller; 29.     }
30.
31.     event ItemListed(
32.     address indexed sel
33.     address indexed nftAddress,
34.     uint256 indexed tokenId,
35.     uint256 price 36.     );
37.
38.     event ItemCanceled(
39.     address indexed seller,
40.     address indexed nftAddress,
41.     uint256 indexed tokenId 42.     );
43.
44.     event ItemBought(
45.     address indexed buyer,
46.     address indexed nftAddress,
47.     uint256 indexed tokenId,
48.     uint256 price 49.     );
50.
51.     mapping(address => mapping(uint256 => Listing)) private s_listings; 52.
mapping(address => uint256) private s_proceeds;
53.
54.     modifier notListed(
55.     address nftAddress,
56.     uint256 tokenId
57.     ) {
58.     Listing memory listing = s_listings[nftAddress][tokenId]; 59.         if
(listing.price > 0) {
60.         revert AlreadyListed(nftAddress, tokenId);
61.     }
62.     _; 63.     }
64.
```

```
66.     Listing memory listing = s_listings[nftAddress][tokenId];
67.     if (listing.price <= 0) {
68.         revert NotListed(nftAddress, tokenId);
69.     }
70.     _; 71.     }
72.
73.     modifier isOwner(
74.         address nftAddress,
75.         uint256 tokenId,
76.         address spender
77.     ) {
78.         IERC721 nft = IERC721(nftAddress);
79.         address owner = nft.ownerOf(tokenId);
80.         if (spender != owner) {
81.             revert NotOwner();
82.         }
83.         _; 84.     }
85.
86.     // IsNotOwner Modifier - Nft Owner can't buy his/her NFT
87.     // Modifies buyItem function
88.     // Owner should only list, cancel listing or update listing
89.     /* modifier isNotOwner(
90.         address nftAddress,
91.         uint256 tokenId,
92.         address spender
93.     ) {
94.         IERC721 nft = IERC721(nftAddress);
95.         address owner = nft.ownerOf(tokenId);
96.         if (spender == owner) {
97.             revert IsNotOwner();
98.         }
99.         _; 100.     } */ 101.
102.     ///////////////////////////////////////////////////
103.     // Main Functions //
104.     ///////////////////////////////////////////////////
105.     /*
106.     * @notice Method for Listing NFT
107.     * @param nftAddress Address of NFT contract
108.     * @param tokenId Token ID of NFT
```

```
114. uint256 price
115. )
116. external
117. notListed(nftAddress, tokenId)
118. isOwner(nftAddress, tokenId, msg.sender) 119. {
120. if (price <= 0) {
121. revert PriceMustBeAboveZero(); 122. }
123. IERC721 nft = IERC721(nftAddress);
124. if (nft.getApproved(tokenId) != address(this)) {
125. revert NotApprovedForMarketplace(); 126. }
127. s_listings[nftAddress][tokenId] = Listing(price, msg.sender);
128. emit ItemListed(msg.sender, nftAddress, tokenId, price); 129. }
130.
131. /*
132.  * @notice Method for cancelling Listing
133.  * @param nftAddress Address of NFT contract
134.  * @param tokenId Token ID of NFT
135.  */
136. function cancelListing(address nftAddress, uint256 tokenId)
137. external
138. isOwner(nftAddress, tokenId, msg.sender)
139. isListed(nftAddress, tokenId) 140. {
141. delete (s_listings[nftAddress][tokenId]);
142. emit ItemCanceled(msg.sender, nftAddress, tokenId); 143. }
144.
145. /*
146.  * @notice Method for buying Listing
147.  * @notice The owner of an NFT could unapprove the marketplace,
148.  * which would cause this function to fail
149.  * Ideally you'd also have a `createOffer` functionality.
150.  * @param nftAddress Address of NFT contract
151.  * @param tokenId Token ID of NFT
152.  */
153. function buyItem(address nftAddress, uint256 tokenId)
154. external
```

```
158. nonReentrant
159. {
160. // Challenge - How would you refactor this contract to take: 161.
// 1. Arbitrary tokens as payment? (HINT - Chainlink Price Feeds!) 162.
// 2. Be able to set prices in other currencies?
163. // 3. Tweet me @PatrickAlphaC if you come up with a solution!
164. Listing memory listedItem = s_listings[nftAddress][tokenId];
165. if (msg.value < listedItem.price) {
166. revert PriceNotMet(nftAddress, tokenId, listedItem.price); 167.     }
168.     s_proceeds[listedItem.seller] += msg.value;
169.     // Could just send the money...
170.     // https://fravoll.github.io/solidity-patterns/pull_over_push.html
171.     delete (s_listings[nftAddress][tokenId]);
172.     IERC721(nftAddress).safeTransferFrom(listedItem.seller, msg.sender,
tokenId);
173.     emit ItemBought(msg.sender, nftAddress, tokenId, listedItem.price);
174.     } 175.
176. /*
177. * @notice Method for updating Listing
178. * @param nftAddress Address of NFT contract
179. * @param tokenId Token ID of NFT
180. * @param newPrice Price in Wei of the item
181. */
182. function updateListing(
183. address nftAddress,
184. uint256 tokenId,
185. uint256 newPrice
186. )
187. external
188. isListed(nftAddress, tokenId)
189. nonReentrant
190. isOwner(nftAddress, tokenId, msg.sender) 191.     {
192.     //We should check the value of `newPrice` and revert if it's below
zero (like we also check in `listItem`)
193.     if (newPrice <= 0) {
194.     revert PriceMustBeAboveZero(); 195.     }
196.     s_listings[nftAddress][tokenId].price = newPrice;
197.     emit ItemListed(msg.sender, nftAddress, tokenId, newPrice);
```

```

202.  */
203.  function withdrawProceeds() external {
204.    uint256 proceeds = s_proceeds[msg.sender];
205.    if (proceeds <= 0) {
206.      revert NoProceeds();
207.    }
208.    s_proceeds[msg.sender] = 0;
209.    (bool success, ) = payable(msg.sender).call{value: proceeds}("");
210.    require(success, "Transfer failed"); 211.    } 212.
213.    //////////////////////////////////
214.    // Getter Functions // 215.
215.    ////////////////////////////////// 216.
217.    function getListing(address nftAddress, uint256 tokenId)
218.    external
219.    view
220.    returns (Listing memory)
221.    {
222.    return s_listings[nftAddress][tokenId]; 223.    } 224.
225.    function getProceeds(address seller) external view returns (uint256) {
226.    return s_proceeds[seller];
227.    } 228. } 229.

```

### Index.js:

```

// Importing core libraries used in index.js import React from "react" import
Link from "next/link" import Image from "next/image" import { useMoralisQuery,
useMoralis } from "react-moralis" import { Carousel } from "react-responsive-
carousel" import "react-responsive-carousel/lib/styles/carousel.min.css" import
Popup from "reactjs-popup" import "reactjs-popup/dist/index.css"

```

```
// Importing modules from other pages within the Project import nft00 from
"../assets/images/nft00.webp" import styles from "../styles/Home.module.css"
import NFTBox from "../components/NFTBo
```

```
import nft1 from "../assets/images/mitanga.png"
import nft_logo from
"../assets/images/nft_logo.png" import foreground
from "../foreground1.gif" import small from
"../assets/images/small1.png" import lostpoet from
"../assets/images/lostpoet.jpg" import nft2 from
"../assets/images/nft05.jpeg" import rare from
"../assets/images/rare.jpg" import king3 from
"../assets/images/king3.jpg" import king4 from
"../assets/images/king4.jpg" import nft3 from
"../assets/images/nft03.jpeg" import orachi from
"../assets/images/orachi.png" import cart1 from
"../assets/images/cart1.jpg" import mitanga2 from
"../assets/images/mitanga2.png" import nft01 from
"../assets/images/nft01.webp"
import cs_2 from "../assets/images/Beige Simple Picture Collage
Christmas Card.gif" import cs_3 from "../assets/images/cs-3.jpg" import
cs_4 from "../assets/images/cs4.webp" import cs_5 from
"../assets/images/cs5.jpg"
export default function Home() {      const { isWeb3Enabled } = useMoralis()
const { data: listedNfts, isFetching: fetchingListedNfts } = useMoralisQuery(
// TableName
  // Function for the query
  "ActiveItem",      (query) =>
query.limit(10).descending("tokenId")
)
console.log(listedNfts)

return (
  <div className="container">
    <center>
      <div className={styles.container_whole} infiniteLoop={true}>
        <div className={styles.content}>
          <div className={styles.container1}>
            <Image
              className={styles.nft_logo}
width={400}
              height={340}
            />
          </div>
        </div>
      </div>
    </center>
  </div>
)
```

```

Collection
    </p>
<Popup
trigger={
    <button className={styles.exp}>
      <Link href="#">Explore NFT</Link>
    </button>
  }
position="right center"
  style={{ alignItems: `justify` }}
  >
    You are encouraged to explore the amazing NFTs
  </Popup>
</div>
<div className={styles.container2}>
  <Carousel
showArrows={false}
showStatus={false}
autoPlay={true}
infiniteLoop={true}
  >
    <Image
className={styles.img2}
style={{ borderRadius: `20px` }}
src={foreground} width={700}
height={400}
    >></Image>
    <Image className={styles.img2}
style={{ borderRadius: `20px` }}
src={cs_2} width={700}
height={400}
    >></Image>
    <Image className={styles.img2}
style={{ borderRadius: `20px` }}
src={cs_3} width={700}
height={400}
    >></Image>

```

```

src={cs_4}                                width={700}
height={400}
                                ></Image>
<Image                                className={styles.img2}
style={{ borderRadius: `20px` }}
src={cs_5}                                width={700}
height={400}
                                ></Image>
                                </Carousel>
                                </div>
                                </div>
                                </div>
                                </center>                                <h1
className="py-4 px-4 font-bold text-2xl"
                                style={{ marginBottom: `2%`, marginLeft: `-80%` }}
                                >
                                Top Collections
                                </h1>
                                <div className={styles.card1}>
                                <div className={styles.card_image_1}>
                                <Image className={styles.img1} width={300} height={200}
src={lostpoet} />
                                <div className={styles.position}>
                                <Image className={styles.img2} width={50} height={60}
src={small} />
                                <p className={styles.des} style={{ fontWeight: `700` }}>
                                Lost Poets
                                </p>
                                </div>
                                </div>
                                <div className={styles.card_image_1}>
                                <Image className={styles.img1} width={300} height={200}
src={nft3} />
                                <div className={styles.position}>
                                <Image className={styles.img2} width={50} height={60}
src={orachi} />
                                <p className={styles.des} style={{ fontWeight: `700` }}>

```

Lost Poets

```

        </p>
      </div>
    </div>
    <div className={styles.card_image_1}>
      <Image className={styles.img1} width={300} height={200}
src={nft2} />
      <div className={styles.position}>
        <Image className={styles.img2} width={50} height={60}
src={king4} />
        <p className={styles.des} style={{ fontWeight: `700` }}>
          Lost Poets
        </p>
      </div>
    </div>
    <div className={styles.card_image_1}>
      <Image className={styles.img1} width={300} height={200}
src={rare} />
      <div className={styles.position}>
        <Image className={styles.img2} width={50} height={60}
src={king3} />
        <p className={styles.des} style={{ fontWeight: `700` }}>
          Lost Poets
        </p>
      </div>
    </div>
  </div>
  <h1 className="py-4 px-4 font-bold
text-2xl" style={{ marginLeft: `-83.5%`, marginTop: `2%`,
marginBottom: `0.5%` }}>
  >
  Trending
</h1>
{ /* <div className={styles.cards_container}> */}
<div className={styles.card1}>
  <div className={styles.card1_1}
/*style={{alignItems:`center`}}*/>
    <center>
      <Link href="/nft_detail">
        <Image className={styles.card1_1img}
src={nft1}></Image>
      </Link>
    </center>
  <div className={styles.cardheading}>

```

```

        <h5 className={styles.inside}>Narukami-crone #01569</h5>
    </div>
    <div className={styles.card1_1} style={{ alignItems: `center` }}>
        <center>
            <Link href="/nft_detail">
                <Image className={styles.card1_1img} width={310}
src={king3}></Image>
            </Link>
        </center>
        <div className={styles.cardheading}>
            <h5 className={styles.inside}>Narukami-crone #01569</h5>
            <h6 className={styles.inside}>1.67 ETH</h6>
            <p className={styles.inside}>End in 3 days</p>
        </div>
    </div>
    <div className={styles.card1_1} style={{ alignItems: `center` }}>
        <center>
            <Link href="/nft_detail">
                <Image className={styles.card1_1img} width={310}
src={king4}></Image>
            </Link>
        </center>
        <div className={styles.cardheading}>
            <h5 className={styles.inside}>Narukami-crone #01569</h5>
            <h6 className={styles.inside}>1.67 ETH</h6>
            <p className={styles.inside}>End in 3 days</p>
        </div>
    </div>
    <div className={styles.card1_1} style={{ alignItems: `center` }}>
        <center>
            <Link href="/nft_detail">
                <Image className={styles.card1_1img} width={310}
src={orachi}></Image>
            </Link>
        </center>
        <div className={styles.cardheading}>
            <h5 className={styles.inside}>Narukami-crone #01569</h5>
            <h6 className={styles.inside}>1.67 ETH</h6>
            <p className={styles.inside}>End in 3 days</p>
        </div>
    </div>
    <div className={styles.card1_1} style={{ alignItems: `center` }}>

```

```

                style={{ marginTop: `1%` }}
src={cart1}
            ></Image>
        </Link>
    </center>
    <div className={styles.cardheading}>
        <h5 className={styles.inside}>Narukami-crone #01569</h5>
        <h6 className={styles.inside}>1.67 ETH</h6>
        <p className={styles.inside}>End in 3 days</p>
    </div>
</div>
<div className={styles.card1_1} style={{ alignItems: `center` }}>
    <center>
        <Link href="/nft_detail">
            <Image
                className={styles.card1_1img}
src={mitanga2}></Image>
        </Link>
    </center>
    <div className={styles.cardheading}>
        <h5 className={styles.inside}>Narukami-crone #01569</h5>
        <h6 className={styles.inside}>1.67 ETH</h6>
        <p className={styles.inside}>End in 3 days</p>
    </div>
</div>
<div className={styles.card1_1} style={{ alignItems: `center` }}>
<center>
        <Link href="/nft_detail">
            <Image
className={styles.card1_1img}
height={1500}
                src={nft00}
alt="Logo"
            ></Image>
        </Link>
    </center>
    <div className={styles.cardheading}>
        <h5 className={styles.inside}>Narukami-crone #01569</h5>
        <h6 className={styles.inside}>1.67 ETH</h6>
        <p className={styles.inside}>End in 3 days</p>
    </div>
</div>
<div className={styles.card1_1} style={{ alignItems: `center` }}>

```

```

src={nft01}></Image>
    </Link>
  </center>
  <div className={styles.cardheading}>
    <h5 className={styles.inside}>Narukami-crone #01569</h5>
    <h6 className={styles.inside}>1.67 ETH</h6>
    <p className={styles.inside}>End in 3 days</p>
  </div>
</div>
</div>
{ /* </div> */ }
<h1 className="py-4 px-4 font-bold text-2xl" style={{ marginLeft:
`80%` }}>
  Recently Listed
</h1>
<div className={styles.back}>
{isWeb3Enabled ? (
  fetchingListedNfts ? (
    <div>Loading...</div>
  ) : (
    listedNfts.map((nft) =>
    {
      console.log(nft.attributes)
const { price, nftAddress, tokenId, marketplaceAddress, seller } =
nft.attributes
      return (
        <NFTBox
          price={price}
          nftAddress={nftAddress}
          tokenId={tokenId}
          marketplaceAddress={marketplaceAddress}
          seller={seller}
          key={` ${nftAddress}${tokenId}` }
        />
      )
    })
  ) : (
    <div>Web3 Currently Not Enabled</div>
  )}
</div>
</div>

```

## 5.1. Components, Libraries, Web Services and stubs

Backend Libraries:

```
{
  "name": "nextjs-nft-marketplace-fcc",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "lint": "solhint 'contracts/*.sol'",
    "lint:fix": "solhint 'contracts/**/*.sol' --fix",
    "format": "prettier --write .",
    "coverage": "hardhat coverage"
  },
  "devDependencies": {
    "@chainlink/contracts": "^0.4.0",
    "@nomiclabs/hardhat-ethers": "npm:hardhat-deploy-ethers@^0.3.0-beta.13",
    "@nomiclabs/hardhat-etherscan": "^3.0.0",
    "@nomiclabs/hardhat-waffle": "^2.0.1",
    "@openzeppelin/contracts": "^4.5.0",
    "babel-eslint": "^10.1.0",
    "chai": "^4.3.4",
    "dotenv": "^16.0.0",
    "ethereum-waffle": "^3.4.0",
    "ethers": "^5.5.1",
    "hardhat": "^2.9.1",
    "hardhat-contract-sizer": "^2.4.0",
    "hardhat-deploy": "^0.9.29",
    "hardhat-gas-reporter": "^1.0.7",
    "prettier": "^2.5.1",
    "prettier-plugin-solidity": "^1.0.0-beta.19",
    "solhint": "^3.3.6",
    "solidity-coverage": "^0.7.13"
  },
  "mocha": {
    "timeout": 10000000
  }
}
```

}

## Frontend Libraries:

```
{
  "name": "nextjs-nft-marketplace-fcc",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "moralis:sync": "moralis-admin-cli connect-local-devchain --chain hardhat -moralisSubdomain lkei3yi1kglb.usemoralis.com --frpcPath ./frp/frpc",
    "moralis:cloud": "moralis-admin-cli watch-cloud-folder --moralisSubdomain lkei3yi1kglb.usemoralis.com --autoSave 1 --moralisCloudfolder ./cloudFunctions"
  },
  "dependencies": {
    "@madzadev/image-slider": "^1.0.9",
    "mongodb": "^4.12.1",
    "mongoose": "^6.7.5",
    "moralis": "^1.11.0",
    "next": "^12.1.5",
    "next-images": "^1.8.4",
    "next-transpile-modules": "^10.0.0",
    "react": "^18.1.0",
    "react-dom": "18.1.0",
    "react-icons": "^4.7.1",
    "react-moralis": "^1.3.5",
    "react-responsive-carousel": "^3.2.23",
    "reactjs-popup": "^2.0.5",
    "web3uikit": "^0.0.133"
  },
  "devDependencies": {
    "autoprefixer": "^10.4.5",
    "dotenv": "^16.0.0",
    "eslint": "8.14.0",
    "eslint-config-next": "12.1.5",
    "postcss": "^8.4.12",
    "tailwindcss": "^3.0.24"
  }
}
```

## 5.2. Deployment Environment

Following is the deployment environment:

This Marketplace is deployed at:

- Windows Operating System
- Runtime Vercel Network
- Web3Modal and web3uikit libraries for enabling web3 and connection of Digital Wallet to our Marketplace

## 5.3. Tools and Techniques

Following are tools and techniques for our platform:

### Backend:

- Node.js, JavaScript, Hardhat, solidity, smart contracts, tests **Frontend:**

- Node.js, JavaScript, Next.js (React Framework), Moralis, Tailwind CSS and Simple CSS

## 5.4. Best Practices / Coding Standards

The Best Practices that our project includes:

- We have use comments to understand the code easily.
- We have made the modules of each function or parameter and used it by importing it from another page.
- There is no duplication of code several times as it increases the complexity.
- Calling one function from one page to another page.

## 5.5. Version Control

We have the following version control:

- git version 2.35.1.windows.2

# Chapter 6

# Testing and Evaluation

## Chapter 6: Testing and Evaluation

In this chapter, we will discuss the testing and evaluation phase of the NFT Meta Market project. This phase is critical to ensure that the project meets the requirements and performs as expected. We will discuss various testing techniques that were used in this project to ensure quality and reliability.

### 6.1. Use Case Testing

**Use Case Testing** Use case testing is a technique used to verify that the system is performing as intended. In this project, we identified several use cases, such as creating 3D NFTs, buying and selling 3D NFTs, and searching for 3D NFTs. We also identified use cases for AR-based NFTs, such as viewing AR-based NFTs, and using AR technology to enhance the buying and selling experience. We created test cases for each use case and verified that the system performs as expected.

### 6.2. Equivalence partitioning

**Equivalence Partitioning** Equivalence partitioning is a testing technique used to divide a range of inputs into groups that are expected to exhibit similar behavior. We used this technique to test the user input for creating 3D NFTs. We divided the input into groups such as valid inputs, invalid inputs, and edge cases. We also used equivalence partitioning to test the AR functionality of the system, dividing inputs into groups such as valid AR markers, invalid AR markers, and edge cases.

### 6.3. Boundary value analysis

**Boundary Value Analysis** Boundary value analysis is a testing technique used to identify the input values that are at the edge of the input domain. In this project, we used this technique to test the user input for buying and selling 3D NFTs. We identified the input values at the upper and lower boundaries and created test cases to ensure that the system behaves correctly in these scenarios.

We also used boundary value analysis to test the AR functionality of the system, identifying edge cases such as low light conditions and testing how the system behaves in those cases.

## **6.4. Data flow testing**

**Data Flow Testing** Data flow testing is a technique used to verify the correct flow of data in the system. In this project, we used this technique to test the data flow for creating and managing 3D NFTs. We identified the input and output data points for each function and verified that the data flow is correct. We also tested the data flow for AR-based NFTs, ensuring that the AR markers are correctly associated with the NFTs.

## **6.5. Unit testing**

**Unit Testing** Unit testing is a technique used to test individual units or components of the system. In this project, we used unit testing to test each function of the system, including the functions that handle 3D NFTs and AR-based NFTs. We created test cases for each function and verified that the function performs as expected.

## **6.6. Integration testing**

**Integration Testing** Integration testing is a technique used to verify that the components of the system work together as expected. In this project, we used integration testing to test the interaction between different functions of the system, including the functions that handle 3D NFTs and AR-based NFTs. We created test cases for each integration point and verified that the system behaves correctly in these scenarios.

## **6.7. Performance testing**

**Performance Testing** Performance testing is a technique used to verify that the system meets the performance requirements. In this project, we used performance testing to test the response time for creating and managing 3D NFTs and AR-based NFTs. We created test cases for each function and verified that the response time meets the performance requirements.

## 6.8. Stress Testing

Stress Testing Stress testing is a technique used to verify that the system can handle a high load of users and transactions without crashing or slowing down. In this project, we used stress testing to test the system's ability to handle a high volume of users and transactions when buying and selling 3D NFTs and AR-based NFTs.

To perform stress testing, we used a tool that simulates a large number of concurrent users and transactions on the system. We gradually increased the load on the system to test its performance at different levels of stress.

During stress testing, we monitored various system metrics such as response time, CPU utilization, memory usage, and network traffic. We also monitored the system logs to identify any errors or exceptions.

Based on the stress testing results, we identified the system's performance limits and adjusted improve its scalability and performance. We optimized the system by implementing caching mechanisms, optimizing database queries, and fine-tuning the server configuration.

Overall, stress testing was critical to ensuring that the NFT Meta Market project can handle a high load of users and transactions without compromising its performance and reliability.

# Chapter 7

## Summary, Conclusion and Future Enhancements

## **Chapter 7: Summary, Conclusion & Future Enhancements**

In this chapter, a concise summary of the NFT Meta Market project is provided, highlighting its goals, key features, and achievements. The successful implementation of core functionalities, including the integration of AR-based NFTs, showcases the platform's innovation and user engagement. A critical review acknowledges strengths while identifying areas for improvement, emphasizing the importance of user feedback and continuous optimization. Valuable lessons are learned, including the significance of thorough testing, scalability considerations, and effective project management. Future enhancements and recommendations encompass aspects such as security measures, search and discovery enhancements, support for diverse NFT formats, social features, and potential collaborations. These enhancements aim to further enhance the platform's value, user experience, and market reach.

### **7.1. Project Summary**

The NFT Meta Market project aimed to create a comprehensive platform that enables users to buy, sell, and explore 3D NFTs, with a specific focus on integrating AR-based NFTs. The project involved meticulous planning, design, and development to create a user-friendly marketplace where creators, collectors, and enthusiasts can engage in the world of digital art and collectibles. The platform provided features such as NFT creation, trading, and immersive AR experiences, all designed to offer a seamless and engaging user experience.

### **7.2. Achievements and Improvements**

Throughout the project, significant achievements and improvements were made. The team successfully implemented the core functionalities of the platform, including the ability for users to create, list, and trade 3D NFTs. The integration of AR technology added an innovative and immersive aspect to the platform, allowing users to experience AR-based NFTs in real-world environments. Testing techniques, such as use case testing and data flow testing, ensured the reliability and quality of the system. Feedback from users and usability testing played a crucial role in improving the user interface, navigation, and overall user experience. Additionally,

optimization efforts were undertaken to enhance performance, scalability, and security, resulting in a more efficient and secure platform.

### **7.3. Critical Review**

A critical review of the project revealed both strengths and areas for improvement. The successful implementation of core functionalities and the integration of AR-based NFTs demonstrated the project's innovation and potential in the rapidly evolving NFT landscape. The platform's intuitive interface and seamless user experience received positive feedback. However, challenges were encountered during stress testing, highlighting the need for further optimization to enhance the system's ability to handle high volumes of concurrent users and transactions. The review also emphasized the importance of continuous user feedback and usability testing to identify areas for refinement and improvement.

### **7.4. Lessons Learnt**

The project provided valuable lessons for the team. Thorough testing and evaluation emerged as crucial aspects to ensure the platform's performance, stability, and reliability. Early identification of scalability challenges and proactive optimization measures were essential to address performance bottlenecks and improve the platform's ability to handle increasing user demands. The integration of user feedback and usability testing played a significant role in enhancing the platform's usability, user interface, and overall user experience. Effective project management, coordination, and collaboration among team members proved pivotal in successfully delivering the project within the defined timeline.

## 7.5. Future Enhancements/Recommendations

To further enhance the NFT Meta Market platform and stay at the forefront of the evolving NFT market, several future enhancements and recommendations are proposed:

- **Strengthening Security:** Implementing advanced security measures, such as two-factor authentication, encryption, and secure wallet integration, to ensure the protection of user data, transactions, and NFT assets.
- **Enhanced Search and Discovery:** Expanding the platform's search and discovery functionalities, including advanced filters, personalized recommendations, and curated collections, to facilitate easier exploration and discovery of NFTs based on user preferences and interests.
- **Diversified NFT Formats:** Supporting additional NFT formats beyond 3D, such as audio-based or video-based NFTs, to provide a more diverse range of digital assets for creators and collectors to trade and engage with.
- **Social Features and Community Engagement:** Introducing social features, such as user profiles, comments, likes, and sharing options, to foster community interaction, collaboration, and engagement among users. This can create a vibrant community-driven ecosystem within the platform.
- **Partnerships and Collaborations:** Establishing partnerships with renowned artists, influential creators, and prominent collectors to expand the platform's NFT collection, attract a wider audience, and increase the platform's visibility.

# Appendices

## Appendix A: Information / Promotional Material

[Paragraph Text 12 pt., Calibri, 1.5 Line Spacing, Justified]

[Between 4 to 8 lines describe what is this appendix all about]


### A.1. Broacher



### A.2. Flyer



### A.3. Standee



Department of Computer Science  
Superior University Gold Campus  
Faculty Of CS&IT

**FYP-BCSM-F22-033**  
**NFT Meta Market**  
Explore , Collect & Sell NFT  
Supervisor: Sir Talha



Tools :



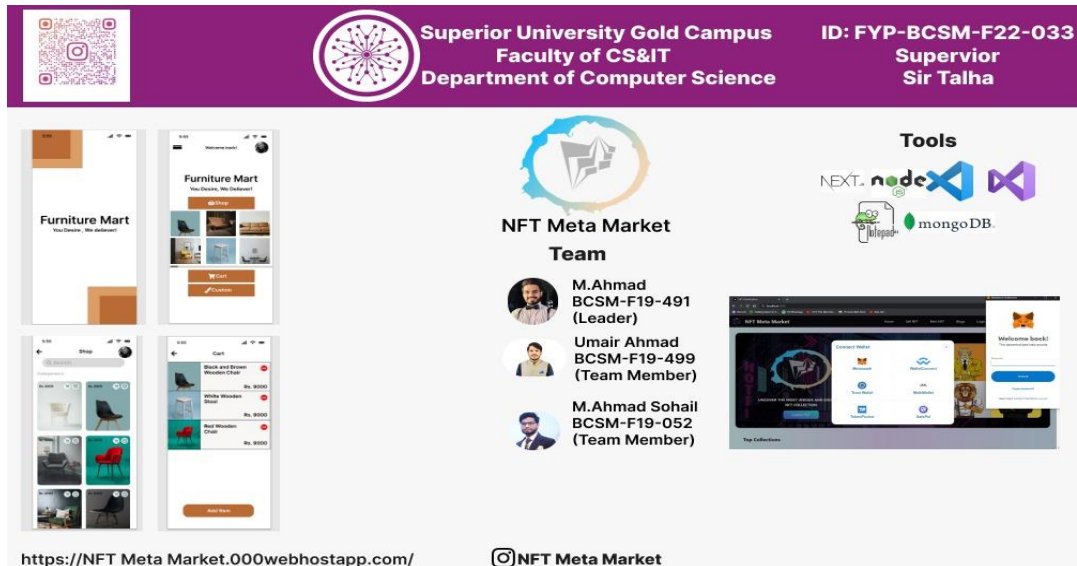
M.Ahmad  
BCSM-F19-491  
(Leader)


Umair Ahmad  
BCSM-F19-499  
(Team Member)

M.Ahmad Sohail  
BCSM-F19-052  
(Team Member)

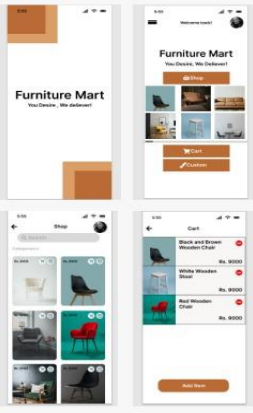
<https://NFT Meta Market.000webhostapp.com/>  
NFT Meta Market

### A.4. Banner




 Superior University Gold Campus  
Faculty of CS&IT  
Department of Computer Science

ID: FYP-BCSM-F22-033  
Supervisor  
Sir Talha



Tools

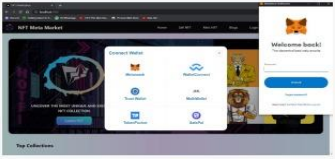



**NFT Meta Market**  
**Team**

M.Ahmad  
BCSM-F19-491  
(Leader)

Umair Ahmad  
BCSM-F19-499  
(Team Member)

M.Ahmad Sohail  
BCSM-F19-052  
(Team Member)



<https://NFT Meta Market.000webhostapp.com/>  NFT Meta Market

# Reference and Bibliography

## Reference and Bibliography

- [1] <https://solsea.io/>
- [2] <https://opensea.io/>
- [3] <https://rarible.com/>
- [4] <https://www.niftygateway.com/>
- [5] <https://www.binance.com/en/nft/home>

# Index

# Index

[A]

[B]

[C]