# NFTForge: Empowering Decentralized Trades with Custom Tokens

**A PROJECT REPORT**

***Submitted by***

Ankit Kumar(20BCS7935), Yash Kumar (20BCS7923), Tijil Jha(20BCS7953), Gurpreet Singh (20BCS2676)

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

**IN**

COMPUTER SCIENCE AND ENGINEERING

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

**Chandigarh University**
MAY 2024

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

# BONAFIDE CERTIFICATE

Certified that this project report " **NFTForge"** is the bonafide work of **"Ankit Kumar, Yash Kumar, Tijil Jha, Gurpreet Singh"** who carried out the project work under my/our supervision.

| | |
|---|---|
| **SIGNATURE** | **SIGNATURE** |
| Dr. Navpreet Kaur Walia | Er. Adil Husain Rather |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Computer Science and Engineering | Computer Science and Engineering |

Submitted for the project viva-voice examination held on __________________ .

| | |
|---|---|
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

# ACKNOWLEDGEMENT

We want to extend our gratitude and thanks to everyone who made the completion of this report possible. A special acknowledgment is owed to our supervisor, **Mr. Adil Husain Rather**, whose assistance, insightful suggestions, and encouragement supported us throughout the entire development process and in compiling this report. We are sincerely appreciative of the time spent by him proofreading and rectifying our numerous errors. We would also like to express our gratitude to all the lecturers and supervisors who dedicated their utmost effort to guide the team toward achieving the goal and provided encouragement to keep our progress on track. Our heartfelt thanks are extended to all classmates, particularly our friends, for generously offering their time, assistance, and support whenever needed during the development of our project.

Tijil Jha (20BCS7953)
Yash Kumar (20BCS7923)
Ankit Kumar (20BCS7935)
Gurpreet Singh (20BCS2676)

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

The current landscape of NFT (Non-Fungible Token) services has raised significant concerns regarding decentralization, security, and the protection of intellectual property. These issues stem from the heavy reliance on third-party entities for NFT storage and the prevalence of security vulnerabilities that expose the exact location of NFTs. To address these challenges, we introduce NFTForge, an innovative platform designed to establish a new standard for NFT transactions.

NFTForge leverages the principles of Blockchain and Web3 technology to offer a secure and decentralized environment for NFT buying, selling, and minting. The cornerstone of this revolutionary approach is NFT art security, a feature that obscures the exact location of each artwork, thereby safeguarding against art theft and preserving artisticoriginality.

One of the distinctive features of NFTForge is the removal of the need for third-party vendors and intermediaries, ensuring privacy and security by utilizing blockchain technology for NFT transactions. Additionally, our custom token, NFTForge, plays a crucial role in resolving the widespread issues of the NFT ecosystem, offering a seamless and efficient means of exchange within the platform.

With NFTForge, we are poised to transform the NFT space by addressing the existing technological flaws and ushering in a new era of NFT trading that aligns with the core tenets of decentralization, security, and integrity. This innovative platform is a step forward in the evolution of NFT technology, providing a trustworthy and artist-friendly solution for the future of digital art and collectibles.

# साराश

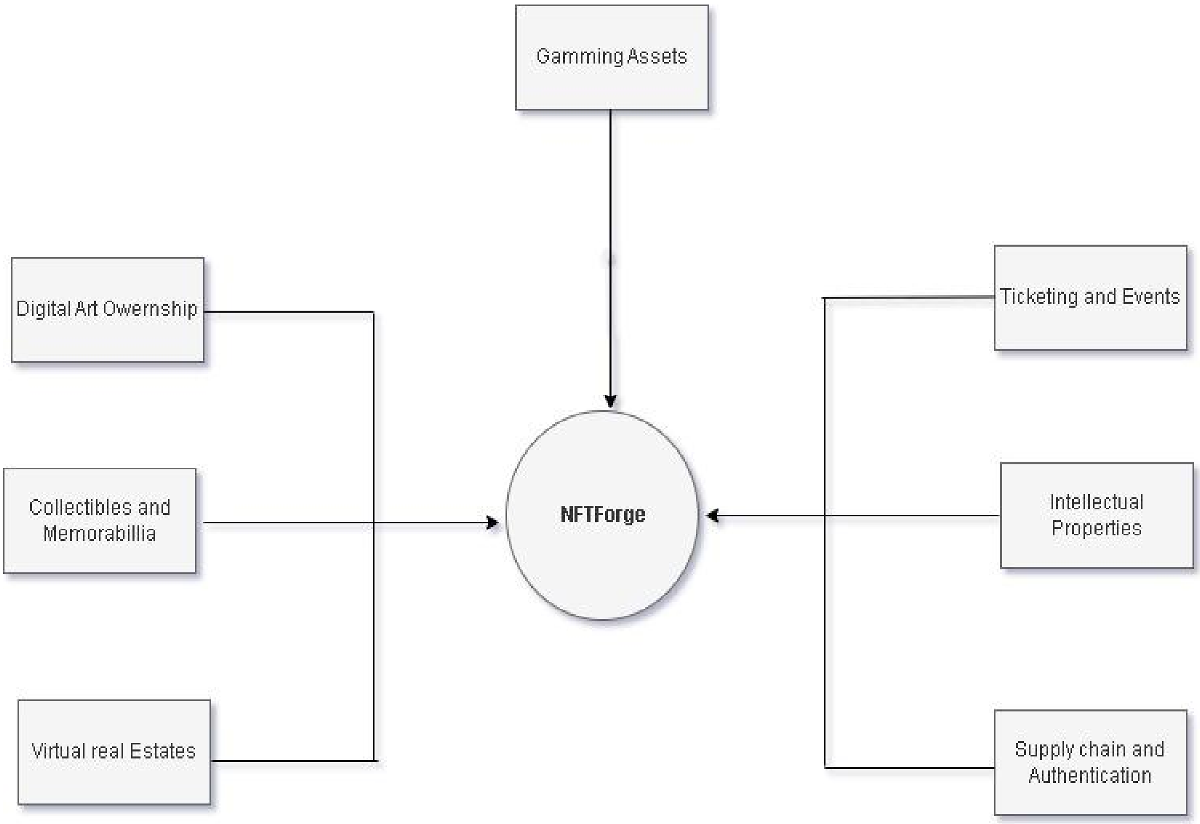
एनएफटी (नॉन-फंगिबल टोकन) सेवाओं के वर्तमान परिदृश्य ने विकेंद्रीकरण, सुरक्षा और बौद्धिक संपदा की सुरक्षा के संबंध में महत्वपूर्ण चिंताएं बढ़ा दी हैं। ये मुद्दे एनएफटी भंडारण के लिए तीसरे पक्ष की संस्थाओं पर भारी निर्भरता और सुरक्षा कमजोरियों की व्यापकता से उत्पन्न होते हैं जो एनएफटी के सटीक स्थान को उजागर करते हैं। इन चुनौतियों का समाधान करने के लिए, हम एनएफटीफोर्ज पेश करते हैं, जो एनएफटी लेनदेन के लिए एक नया मानक स्थापित करने के लिए डिज़ाइन किया गया एक अभिनव मंच है।

एनएफटीफोर्ज एनएफटी खरीद, बिक्री और ढलाई के लिए एक सुरक्षित और विकेन्द्रीकृत वातावरण प्रदान करने के लिए ब्लॉकचेन और वेब3 प्रौद्योगिकी के सिद्धांतों का लाभ उठाता है। इस क्रांतिकारी दृष्टिकोण की आधारशिला एनएफटी कला सुरक्षा है, एक ऐसी सुविधा जो प्रत्येक कलाकृति के सटीक स्थान को अस्पष्ट करती है, जिससे कला की चोरी से बचाव होता है और कलात्मक मौलिकता का संरक्षण होता है।

एनएफटीफोर्ज की विशिष्ट विशेषताओं में से एक तीसरे पक्ष के विक्रेताओं और मध्यस्थों की आवश्यकता को हटाना, एनएफटी लेनदेन के लिए ब्लॉकचेन तकनीक का उपयोग करके गोपनीयता और सुरक्षा सुनिश्चित करना है। इसके अतिरिक्त, हमारा कस्टम टोकन, एनएफटीफोर्ज, एनएफटी पारिस्थितिकी तंत्र के व्यापक मुद्दों को हल करने में महत्वपूर्ण भूमिका निभाता है, जो प्लेटफॉर्म के भीतर विनिमय का एक सहज और कुशल साधन प्रदान करता है।

एनएफटीफोर्ज के साथ, हम मौजूदा तकनीकी खामियों को दूर करके और एनएफटी ट्रेडिंग के एक नए युग की शुरुआत करके एनएफटी क्षेत्र को बदलने के लिए तैयार हैं जो विकेंद्रीकरण, सुरक्षा और अखंडता के मूल सिद्धांतों के साथ संरेखित है। यह अभिनव मंच एनएफटी प्रौद्योगिकी के विकास में एक कदम आगे है, जो डिजिटल कला और संग्रहणीय वस्तुओं के भविष्य के लिए एक भरोसेमंद और कलाकार-अनुकूल समाधान प्रदान करता है

# GRAPHICAL ABSTRACT



# ABBREVIATIONS

| Sr. No. | Abbreviations | Full forms |
|---|---|---|
| 1 | NFT | Non Fungible Token |
| 2 | BFT | Byzantine Fault Tolerance |
| 3 | PoW | Proof of Work |
| 4 | DeFi | Decentralized Finance |
| 5 | IoT | Internet of Things |
| 6 | PoS | Proof of Stake |
| 7 | HTML | Hypertext Markup Language |
| 8 | CSS | Cascading Style Sheets |
| 9 | JS | JavaScript |
| 10 | DFD | Data Flow Diagram |
| 11 | API | Application Programming Interface |
| 12 | ER | Entity-Relationship |
| 13 | UX | User Experience |
| 14 | HTTP | Hypertext Transfer Protocol |
| 15 | UI | User Interface |

**Table 1**

# CHAPTER 1

# INTRODUCTION

## 1.1 Client Identification

In the context of designing an NFT (Non-Fungible Token) platform, user identification and understanding the contemporary issues within the NFT market are pivotal for the success of the project. NFT platforms cater to a specific group of users, and recognizing this target audience is the first step in creating a platform that meets their needs.

The target audience for an NFT platform is primarily composed of individuals interested in digital art, collectibles, and unique digital assets. These users are looking for a platform where they can buy, sell, and trade NFTs, which represent ownership or provenance of digital or physical items. They are typically tech-savvy and have an understanding of blockchain technology, as NFTs are built on blockchain platforms.

One of the contemporary issues that the NFT platform aims to address is the democratization of ownership and trade in the digital realm. The rise of NFTs has disrupted traditional art markets, collectibles, and ownership models. It has provided creators, artists, and collectors with new opportunities to tokenize and monetize digital assets, while also enabling a global audience to participate in these markets. However, this rapid growth has led to challenges such as scams, copyright issues, and environmental concerns due to the energy consumption of some blockchain networks. Therefore, the NFT platform must address these issues and provide a safe and sustainable marketplace for its users.

By identifying the target audience and recognizing the contemporary issues, the project team can develop a platform that is tailored to the needs of artists, creators, collectors, andinvestors. This includes creating user-friendly interfaces for minting and trading NFTs, implementing strong security measures to protect against scams, and selecting environmentally responsible blockchain networks.

Moreover, understanding the target audience allows the team to prioritize features that are most valuable to users, such as user profiles, customizable storefronts, and auction mechanisms. It also enables the platform to provide educational resources for users new to the NFT space and to create a community that fosters trust and collaboration.

In summary, user identification and recognizing contemporary issues are crucial aspects of designing an NFT platform. By understanding the specifi needs and challenges within the NFT market, the project team can develop a platform that is relevant, secure, and user-friendly, ultimately contributing to the growth and sustainability of the NFT ecosystem.

By understanding the target audience and the problem the system aims to address, the project team can design a solution that is relevant, effective, and user-friendly.

## 1.2 Identification of Problem

In this NFT marketplace project, the identified problem is the need for a secure and user-friendly platform for buying and selling NFTs (Non-Fungible Tokens). The contemporary issue in the NFT space is the explosive growth of the market, which has led to a surge in interest and participation but also raised concerns related to authenticity, copyright, and environmental sustainability.

To address these challenges, the proposed solution is to design an NFT marketplace where users can confidently transact digital assets in a decentralized and transparent environment. The platform aims to provide a space where artists, creators, collectors, and investors can mint, trade, and showcase their NFTs, ensuring that the tokens represent authentic and unique digital or physical items. It also seeks to address environmental concerns by selecting energy-efficient blockchain networks or implementing eco-friendly practices.

The NFT marketplace does not replace traditional art and collectibles markets but complements them by providing a digital avenue for ownership and trade. It aligns with the current trend of digital ownership and blockchain technology adoption in the art and entertainment industry, offering a unique and innovative way for creators to monetize their digital assets.

Despite the potential benefits, there are challenges to consider, such as the need for robust security measures to protect users from scams and fraud, as well as issues related to intellectual property and copyright. Continuous improvement, user education, and community building are crucial aspects of the platform's design to ensure its long-term success.

In summary, the NFT marketplace project addresses the growing demand for a secure and efficient platform for buying and selling NFTs. By understanding the contemporary issues in the NFT market and designing a platform that prioritizes security, transparency, and sustainability, the project aims to contribute to the continued growth and evolution of the NFT ecosystem.

## 1.3 Identification of Tasks

To attain the project's objectives and goals, a series of tasks must be accomplished in a timely and efficient manner. These tasks can be classified into distinct stages, encompassing planning, design, implementation, testing, and validation.

In the initial planning phase, it is crucial to outline the project's requirements, encompassing desired functionalities, features, and specifications. This includes collecting information from diverse sources, including pertinent literature, expert insights, and user input. Furthermore, the project team must delineate a schedule for the completion of each task and identify potential risks and challenges that might emerge throughout the project's duration.

In the design phase, the focus is on crafting a comprehensive system architecture that outlines the system's various components and modules. This encompasses the definition of data structures and algorithms essential for implementing the intended functionality. Additionally, the design stage involves the thoughtful selection of suitable tools, technologies, and programming languages to be employed in the system's implementation.

The implementation phase entails the practical coding and construction of the system, guided by the specifications and design documents. This process involves translating the design into tangible code, utilizing the chosen programming languages and tools. Throughout the implementation stage, it is crucial to guarantee that the code is well-structured, efficient, and scalable, adhering to best coding practices and standards.

The testing phase encompasses the scrutiny and verification of the system, aiming to confirm its adherence to specified requirements and its freedom from errors and bugs. This phase incorporates a variety of testing techniques, including unit testing, integration testing, and system testing. The testing process is crucial in ensuring that the system operates as intended, demonstrating reliability, robustness, and scalability.

The concluding phase of the project is validation, which encompasses the assessment of the system by end-users and stakeholders. This process entails collecting feedback and insights from users and evaluating the system's effectiveness and efficiency in realizing its intended goals and objectives. Validation is of paramount importance to ascertain that the system aligns with the needs and requirements of the target audience, ensuring it is user-friendly, intuitive, and efficacious.

In entirety, accomplishing these tasks necessitates a cohesive and collaborative endeavor from the project team, coupled with efficient communication, thorough planning, and adept project management. A comprehensive grasp of the project's tasks and the requisite resources for their completion is imperative for the successful execution of the project.

## 1.4 Distribution of tasks:

| Sr.No | Team Member | Task Assigned |
|---|---|---|
| 1 | Yash Kumar (20BCS7923) | -Implementing Front end<br>-Testing<br>-Documentation |
| 2 | Ankit Kumar (20BCS7935) | -Back-end<br>-Integration<br>-Token creation<br>-Documentation |
| 3 | Tijil Jha (20BCS7953) | -Implementing Front-end<br>-Testing<br>-Documentation |
| 4 | Gurpreet Singh (20BCS2676) | -Back-end<br>-Integration<br>-Documentation |

## 1.5 Timeline

To manage this project effectively, a Gantt chart has been created to outline the various tasks and their associated timelines.

Week 1: Research and Planning

The initial phase of the project involves defining its scope and establishing clear objectives. This process requires a comprehensive understanding of the project's purpose and intended outcomes. Simultaneously, the research team will delve into relevant literature and collect data from various research papers to inform and shape the project's parameters. This literature review aims to identify existing practices, methodologies, and insights pertinent to the project's goals.

Week 2: Algorithm Development and Testing

During this stage, the focus will be on crafting and validating the model for precision and efficacy. The team will engage in an exploration of diverse algorithms with the aim of pinpointing a model that demonstrates efficiency and accuracy sufficient to manage data and security measures adeptly.

Week 3: UI/UX Design and Testing

In this stage, the design team will develop an interface for the system that is user-friendly and intuitive. Subsequently, this interface will undergo testing with a user group to confirm its effectiveness and ease of use.

Week 4: User Testing and System Refinement

During this stage, a cohort of users will evaluate the system to verify its effectiveness and alignment with their requirements. User feedback will be systematically gathered and employed to further enhance and fine-tune the system.

Week 5: Final System Testing and Deployment

In the concluding stage, a thorough examination of the system will be conducted to guarantee its accuracy, effectiveness, and security. Following the successful completion of all tests, the system will undergo deployment, becoming accessible to users.

The Gantt chart serves as a visual representation of the project's timeline, aiding in effective project management and monitoring. This timeline guarantees that the project progresses as planned, with each task completed within its designated time frame.

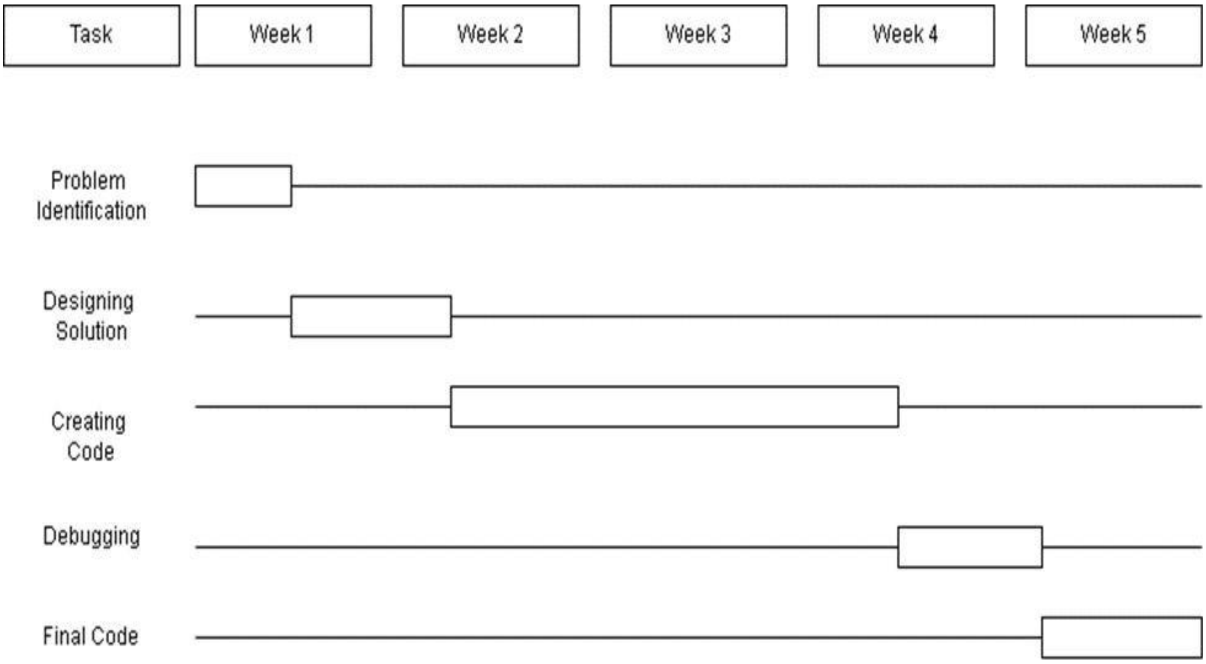


**Figure 1.1** Gantt Chart defining timeline of the project

## 1.6 Organization of the Report

This project report is organized in a structured manner to provide readers with a clear understanding of the project's background, design, implementation, and results analysis.

**CHAPTER 1 –**

Chapter 1 offers an overview of the project, detailing client identification and addressing the pertinent contemporary issue. This chapter defines the problem at hand, delineates the tasks integral to the project, and sets forth a timeline for the completion of the project.

**CHAPTER 2 –**

This chapter of the project is dedicated to conducting a thorough literature review and background study. In this chapter, we will examine and analyze over 15-20 research papers that have been previously published, all of which are relevant to the subject of developing a block-chain technology and NFT Technology. Additionally, we perform a bibliometric analysis and offer a concise summary of the reviewed literature to shed light on the existing body of knowledge and areas where further research is needed. Furthermore, this chapter serves to define the problem at hand and lays out the project's specific goals and objectives.

**CHAPTER 3 –**

This chapter provides an in-depth exploration of the project's design flow and processes. It encompasses the assessment and choice of specifications and features, considerations of design constraints, analysis, and the finalization of features within the constraints. The chapter also delves into the design flow, the selection of designs, and the formulation of an implementation plan/methodology. Additionally, figures are incorporated to visually elucidate the various stages of the design process.

**CHAPTER 4 –**

This chapter centers on the analysis and validation of the project results. It encompasses the execution of the solution, detailed examination, testing procedures, and the outcomes derived from these efforts. Additionally, the chapter underscores the challenges encountered throughout the implementation phase and offers recommendations for potential enhancements in the future.

**CHAPTER 5 –**

This chapter delivers the conclusion and outlines future prospects for the project. It encapsulates a summary of the project's goals, obstacles faced, and accomplishments achieved. The chapter concludes by presenting recommendations for future endeavors aimed at improving the system's functionality and performance.

The report incorporates an appendix that provides comprehensive information about the software tools and methodologies employed in the project. This encompasses details regarding programming languages, machine learning algorithms, and software libraries utilized. Moreover, the appendix encompasses the source code for the project and the datasets utilized for testing and analysis.

In summary, this project report offers a thorough exploration of the design and execution of a symptom-based health enhancement system leveraging machine learning. It details the project's approach, outcomes, and suggestions for prospective initiatives. The report stands as a valuable reference for researchers, healthcare professionals, and policymakers engaged in the advancement and deployment of healthcare systems based on machine learning.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1. Timeline of the reported problem

The timeline of research on designing the  NFTForge project spans over a decade.

***Satoshi Nakamoto (2009):*** The concept of blockchain technology made its debut in a white paper titled “Bitcoin: A Peer-to-Peer Electronic Cash System" which was published in 2008. This document served as the foundation for the creation of the Bitcoin blockchain, which was subsequently introduced in 2009 as the first realization of this concept. The Bitcoin blockchain was designed as a decentralized digital currency system that eliminates the need for trusted intermediaries like banks, and the initial block, known as the “Genesis Block," was mined by Satoshi Nakamoto in January 2009.**[1]**

***Christian Decker et all (2013):*** This paper discussed the security and consensus mechanisms employed in the Bitcoin network, focusing on the Proof of Work (PoW) algorithm. The PoW algorithm utilized a 32-bit nonce, which, when combined with the block's data and hashed using the SHA-256 hash function, needed to produce a hash value that was less than a specific difficulty target. The formula was represented as follows: SHA-256(Block Data + Nonce) \(<\) Target where "Block Data" includes various data from the block, including the previous block's hash, transaction data, and a timestamp. "Nonce" is a 32-bit number that miners adjust repeatedly."SHA-256" represents the SHA-256 cryptographic hash function. "Target" is a predetermined value that determines the difficulty of the proof of work. Miners must find a nonce that results in a hash lower than this target. Miners iterated through nonce values until they found one that satisfied the condition, constituting a computationally intensive and competitive process that ensured the security and consensus in the Bitcoin network**.[2]**

***Jega Anish Dev (2014):*** This paper explored techniques for conducting Bitcoin mining using standard hardware, achieving faster mining speeds by efficiently utilizing the computational resources available in mining networks, regardless of whether they were being used for legal or illegal purposes. Two primary types of hardware used in cryptocurrency mining were Central Processing Units (CPUs) and Graphics Processing Units (GPUs). To further accelerate mining, miners began using ASICs (Application-Specific Integrated Circuits), highly specialized hardware designed solely for cryptocurrency mining. ASICs were exponentially more efficient than CPUs and GPUs because they were purpose-built for a specific mining algorithm. For example, Bitcoin miners used ASICs optimized for SHA-256.**[3]**

***Hiroki Watanabe et all (2015):*** This paper discussed the importance of verifying consent in the context of blockchain and cryptocurrency transactions, especially concerning contracts involving multiple parties. The traditional one-way nature of blockchain transactions posed a challenge when it came to verifying consent in contracts. To address this issue, the paper suggested two potential strategies. The first strategy involved finding new ways to use blockchain technology to enable two-way transactions for contracts, allowing parties to confirm their consent more effectively. The second strategy was to expand the transaction specifications to incorporate the storage of contractual documents within the blockchain itself. The second approach revolved around broadening the scope of transaction details to encompass contractual documents within the blockchain. In this strategy, a novel protocol was devised to capture a consensus trail on the blockchain. Each transaction was linked with contract information, and to address consent concerns, the recipient of a transaction initiated a new transaction referencing the received one, indicating their approval. This process established a sequence of transactions, with the final one returned to the initial contractor who initiated the contract transaction. The initial contractor then validated the contract within the transaction and generated another transaction, typically directed towards a mediator, serving as a confirmation trail. To address the issue of privacy and security when storing contractual documents in a public blockchain, the proposed protocol utilized encryption. It employed both an e-signature key pair and an encryption key pair to protect the contract data, ensuring that only authorized parties could access and view the contract information while maintaining privacy.**[4]**

***Hiroki Watanabe et all (2016):*** This paper introduced a novel approach to enhance the security of a blockchain system used for managing contracts, particularly in domains like digital rights management. The proposed mechanism incorporated a fresh consensus method that leveraged a credibility score and established a hybrid blockchain system. This hybrid approach alternated between the new consensus method and proof-of-stake, effectively preventing attempts by attackers to monopolize resources and ensuring the continued security of the blockchain. Let's examine the proposed solution more closely.

Deterring Malicious Behavior: This approach aimed to discourage harmful actions by leveraging the potential consequences on both reputation and finances. It acted as a deterrent to malicious actors, knowing that their activities would be permanently recorded on the blockchain, which could harm their reputation and future opportunities. Additionally, this approach involved introducing financial penalties or mechanisms for reducing rewards for malicious behavior, further dissuading potential attackers.

Utilizing Hybrid Blockchains: Hybrid blockchains combined the advantages of public and private blockchain models. They offered increased control and data privacy while still benefiting from the security features of public blockchains. This became particularly valuable in business contexts where safeguarding sensitive information was crucial, but the security and transparency of public blockchains were also desirable. By integrating components that required permission and those that were public, hybrid blockchains provided the adaptability and customization necessary for various use cases. This integration enhanced security by ensuring that only authorized individuals had access to sensitive data, addressing concerns related to privacy and control that were associated with purely public blockchain.**[5]**

***Victoria Lemieux (2017):*** This paper explored the applications of blockchain technology and distributed ledgers as trusted systems for record keeping, utilizing an evaluation framework rooted in archival theory. The authors introduced an archival theoretic evaluation framework to assess the suitability of blockchain and distributed ledgers for record keeping purposes. This framework was grounded in archival theory, a discipline focused on the principles and practices of record keeping to ensure the preservation of authentic and reliable records.By applying this framework, the paper systematically evaluated the capabilities of blockchain technology and distributed ledgers in meeting archival requirements. The evaluation framework considered key archival principles such as provenance, reliability, authenticity, and integrity. It examined how blockchain's immutable and decentralized nature could provide a trustworthy record of provenance, tracing the origins and changes made to records. The paper also explored how cryptographic techniques employed in blockchain ensured data integrity, preventing unauthorized alterations to records.**[6]**

***William Nikolakis et all (2018):*** This paper presented a comprehensive framework that explored the transformative potential of blockchain technology in shaping sustainable global value chains. The authors introduced the EVE framework, focusing on evidence, verifiability, and enforceability aspects to demonstrate how blockchain could enhance transparency, trust, and accountability in global business networks.It addressed the challenges faced by global value chains, such as supply chain inefficiencies, lack of transparency, and difficulties in verifying the authenticity of products and processes.**[7]**

***Wei Cai et all (2018):*** This paper explored the innovative intersection of blockchain technology and software systems, with a focus on the development of decentralized applications (dApps). It defined dApps as software applications that ran on a decentralized network of computers, utilizing blockchain technology for data storage, communication, and computation. Unlike traditional applications, dApps operated on a peer-to-peer network, ensuring censorship resistance, enhanced security, and increased user control over their data. The paper also delved into various types of dApps, including financial applications (DeFi), supply chain management solutions, and social media platforms, demonstrating the diverse range of possibilities enabled by this technology. Furthermore, it discussed real-world use cases and applications of dApps in domains such as finance, healthcare, and voting systems, illustrating how blockchain technology disrupted traditional industries by offering innovative solutions.**[8]**

***Fangyu Gai all (2019):*** This paper discussed the Cumulus protocol, a Byzantine Fault Tolerant (BFT)-based sidechain protocol designed to enhance the scalability and efficiency of blockchain systems. The Cumulus protocol was based on the concept of sidechains, which are separate chains that operate in conjunction with the main blockchain. Sidechains allow for the execution of specific tasks, such as smart contracts or rapid transaction processing, off the main chain. A key innovation of Cumulus was its utilization of Byzantine Fault Tolerance, a consensus mechanism that ensured the security and integrity of the network even in the presence of malicious nodes. By integrating BFT into the sidechain protocol, Cumulus enhanced the trustworthiness of off-chain transactions, making them as secure as on-chain transactions**.[9]**

***Jianyu Niu et al. (2019):*** The paper discusses a tactic called "selfish mining" in the Ethereum blockchain system. In simple terms, this tactic involves miners trying to cheat the system by holding onto blocks they've mined and releasing them strategically. This disrupts the normal flow of blocks in the network and can lead to various issues like double-spending and less reliable network operations. Since Ethereum is widely used for things beyond just cryptocurrencies, like smart contracts and decentralized finance, it's crucial to tackle these vulnerabilities to maintain trust in the platform**.[10]**

***Yiming Liu et al. (2020):*** This paper explores how blockchain and machine learning can work together to improve communication and networking systems. It starts by discussing the problems these systems face, such as security issues and the need for better resource management. Blockchain technology is introduced as a solution because it provides a secure and transparent way of managing data. Machine learning, known for its ability to analyze large amounts of data and make predictions, complements blockchain by helping optimize network performance and security measures.**[11]**

***Zhipeng Fan.(2019):*** This research paper explores blockchain consensus algorithms and presents a hybrid consensus algorithm designed to address the shortcomings of individual algorithms. Combining PBFT and PoW, enhances both decentralization and performance, resulting in improved throughput and latency compared to traditional consensus methods. This hybrid approach offers a promising solution to the challenges of achievingdecentralization and efficiency in blockchain systems for the Internet of Things.**[12]**

***Yashika Nagpal. (2020):*** This research paper seems to explore the potential of NFTs, their growth, and the legal challenges they pose, with a particular focus on India's legal framework. It highlights the increasing interest in NFTs as digital collectibles and investments while also acknowledging the uncertainty and legal questions surrounding their use in India.**[13]**

***Deepak Puthal et al. (2020):*** This research paper introduces a novel consensus algorithm tailored for fast, scalable, and private blockchain within large-scale Internet of Things (IoT) frameworks, it likely explores innovative solutions for achieving efficient consensus in blockchain systems optimized for the unique demands of IoT applications.**[14]**

***Rahul Dattaram Belose et al.(2021):*** This research paper examines NFT-based startups listed on cryptocurrency exchanges, presenting a novel NFT classification. It demonstrates impressive NFT performance, including a 130% return on the first listing day and an average long-term investment multiple of 40 (equivalent to 4,000%), outperforming Bitcoin. NFTs exhibit positive alpha, and above-average beta, and played a pivotal role in the cryptocurrency market's recovery post the mid-2021 crash, yielding a return of nearly 350%.**[15]**

***Shreyansh Goushal(2021):***This research paper explores Non-Fungible Tokens (NFTs), tracing their origins and development. It highlights their role in the art and digital asset markets, while acknowledging the challenges they face. Additionally, the paper to investigate if there is a correlation between NFT sales and the prices of Bitcoin and Ethereum. **[16]**

***Subhita Menon et al.(2021):*** This research paper presents a comprehensive survey of blockchain smart networks based on consensus algorithms. It highlights the rapid growth of blockchain technology and its potential to impact various application fields. The paper primarily focuses on consensus mechanisms' sustainability, scalability, throughput, computation overhead, and latency. It offers insights into the diverse applications of blockchain networks across various domains, emphasizing the role of consensus mechanisms in these applications. Furthermore, the paper identifies and discusses open challenges in blockchain consensus algorithm design, offering future research directions. Key themes covered include blockchain technology, consensus mechanisms, and network applications.**[17]**

***Huanliang Xiong et al.(2022):*** This paper provides an overview of recent advancementsin blockchain consensus algorithms. It delves into various consensus mechanisms,highlighting their strengths, weaknesses, and real-world applications. This review serves as a valuable resource for researchers, developers, and blockchain enthusiasts interested in stayingupdated on the latest developments in consensus technology.**[18]**

***Timo Hanke et al.(2022):*** This research paper explores the Dfinity blockchain, emphasizing its secure consensus mechanism. It highlights the unique decentralized randomness beacon and its role in leader selection and ranking within the Dfinity blockchain. The paper also discusses notarization for rapid finality and security. It showcases Dfinity's fastblock times, quick transaction finality with minimal confirmations, and resilience to network disruptions, all with provable security under synchronization conditions.**[19]**

***Phil Gonserkewitz et al.(2022):*** This research paper describes a comprehensive overview of NFTs, including their current use cases and potential future directions It also acknowledges the importance of addressing security issues and educating businesses about NFT opportunities.**[20]**

***Saeed Banaeian Fara et al.(2022):*** This research paper discusses the applications of Non-Fungible Tokens (NFTs) in the real world and the Metaverse. NFTs,, have gained popularity and found applications in various industries. The Metaverse, a blockchain-based technology, is explored as a platform where NFTs can be used for identity management and ownership rights of digital assets. The paper aims to identify new applications of NFTs and presents future directions for their use in industry and academic research.**[21]**

***Kebira Azbeg et al. (2022):*** This research paper offers an overview of blockchain consensus algorithms, discussing their significance in securing distributed systems. It explores various consensus algorithms within the blockchain ecosystem, providing a comparativeanalysis of their strengths and weaknesses and their suitability for different blockchain types. Additionally, the paper highlights challenges and future directions in this rapidly evolving field, aiming to assist in algorithm selection for specific scenarios and inspire the development of new algorithms. Key topics covered include blockchain, distributed systems, and consensus algorithms.**[22]**

***Ruihang Huang et al.(2022):*** This research delves into the consensus mechanism for software-defined blockchains in the Internet of Things (IoT). It assesses traditional consensus algorithms, introduces an improved DPOS-PBFT-based consensus scheme, and incorporates a dynamic credibility-based node ranking system. The results demonstrate the proposed mechanism's robustness, maintaining a high success rate and low latency under significant user loads. This study offers valuable insights for enhancing IoT blockchain consensus mechanisms and their practical applications.**[23]**

***Victoria Lemieux et al.(2023):*** This paper discussed Scene Theory and its application in the field of blockchain technology, especially in Bitcoin, and Ethereum. Scene Theory, originally a concept in the field of cognitive psychology, posits that human perception is not only about recognizing objects but also about understanding the context or "scene" in which those objects exist.Applied to blockchain technology, Scene Theory offered an intriguing orientation perspective that could enhance comprehension of the blockchain ecosystem. It provided a holistic perspective that extended beyond individual transactions or smart contracts, enabling a better understanding of how blockchain operated within a complex ecosystem. This approach helped stakeholders make informed decisions, improve security, and enhance the overall effectiveness of blockchain solutions.**[24]**

In summary, the review of the analyzed research papers illustrates the significant advancements in blockchain technologies, Non-Fungible Tokens (NFTs), and consensus algorithms over the past decade. These studies have shed light on the potential of NFTs as digital collectibles and investments, the challenges and legal implications surrounding their adoption, and their correlation with the prices of prominent cryptocurrencies. Moreover, the exploration of innovative consensus algorithms tailored for IoT applications and the comprehensive survey of blockchain smart networks have paved the way for a deeper understanding of the scalability, sustainability, and applications of blockchain technology in various domains. The review also underscores the ongoing efforts in the development of secure and efficient consensus mechanisms, emphasizing the need for robust security protocols and dynamic credibility-based systems to enhance the performance of blockchain networks. These findings collectively lay the groundwork for the  NFTForge project, indicating a promising trajectory for the integration of advanced blockchain solutions and NFT applications in the digital asset landscape.

## 2.2. Proposed solutions

There are various ways to develop such a system, but the most promising methods include the Implementation of a Hybrid Consensus Algorithm the utilization of NFTs, the Incorporation of Secure Consensus Mechanisms, the Development of Defi-Based Token, the Integration of Smart Contract Functionality, the Implementation of Decentralized Data Preprocessing:

**Implementation of Hybrid Consensus Algorithm:** The use of a hybrid consensus algorithm, combining aspects of Practical Byzantine Fault Tolerance (PBFT) and Proof of Work (PoW), can be considered for ensuring enhanced decentralization and improved throughput. This approach addressed the potential shortcomings of individual algorithms, fostering a robust and efficient blockchain system.

**Utilization of Non-Fungible Tokens (NFTs):** Leveraging the potential of Non-Fungible Tokens (NFTs) can facilitate seamless upload, ownership listing, and trading of digital assets. By adopting NFTs, the platform can enable users to securely and transparently trade digital collectibles, ensuring clear ownership records and simplified transaction processes.

**Incorporation of Secure Consensus Mechanisms:** Integrating secure consensus mechanisms within the project's architecture, such as those explored in the literature, can ensure robust and reliable transaction validation. This approach can enhance the security and credibility of the platform, establishing trust among users and facilitating seamless transactions with minimized risks of fraud or data manipulation.

**Development of Defi-Based Token**: The introduction of a decentralized finance (DeFi) based token, such as the ICP ,bitcoin Token, can serve as the primary medium of exchange within the platform. By utilizing this token, users can effectively engage in NFT purchases, sales, and subsequent transactions, fostering a secure and efficient ecosystem for digital asset trading.

**Integration of Smart Contract Functionality**: Leveraging the capabilities of smart contracts, particularly within the Motoko framework, can ensure the automation and execution of transparent, tamper-proof agreements between buyers and sellers. This feature would enable streamlined and secure NFT transactions, guaranteeing fair and accurate ownership transfers while minimizing the need for intermediary interventions.

**Implementation of Decentralized Data Preprocessing:** Adapting decentralized data preprocessing techniques, as highlighted in the literature, can facilitate the transformation of raw data into a format suitable for machine learning analysis. By ensuring data accuracy, quality, and consistency, the platform can enhance the overall reliability and effectiveness of its machine-learning algorithms, supporting improved decision-making processes and user experiences.

In summary, by incorporating these proposed solutions, the project can establish itself as a cutting-edge platform for NFT trading, emphasizing secure transactions, decentralized data management, and efficient consensus mechanisms. The integration of these strategies can foster a transparent, user-centric, and reliable ecosystem, empowering users to participate in the seamless trading of digital assets while ensuring the security and integrity of their transactions.

## 2.3 Bibliometric Analysis

To further understand the research landscape in the area of NFTs buy and sell process, a bibliometric analysis was conducted. The analysis was performed using the Scopus database, which is a comprehensive database of peer-reviewed literature.

This study examines Non-Fungible Tokens (NFTs) by looking at NFT-based startups on cryptocurrency exchanges. It introduces a new way of classifying NFTs and highlights their impressive performance. For instance, they showed a 130% return on the first day they were listed and an average long-term investment that was 40 times the initial investment, outperforming even Bitcoin. NFTs also played a big role in helping the cryptocurrency market recover after the crash in mid-2021, resulting in a return of almost 350%.

The research paper emphasizes the growing importance of NFTs as a promising investment option. It points out that they have the potential to do better than traditional benchmarks, making them a lucrative opportunity for investors. By offering a detailed look into the potential of NFTs, the study serves as a useful resource for professionals and investors, helping them make informed decisions in the changing world of cryptocurrency investments.

By emphasizing the impact of NFTs in the broader financial market, the research underlines their significant growth and impressive returns. It positions them as essential drivers of innovation and financial success within the thriving cryptocurrency landscape. This paper adds to our understanding of NFTs as transformative elements in modern investments, encouraging exploration and growth in the digital asset industry.

In a time when interest in digital assets is high, the research paper stresses the continued need to explore and understand the many aspects of NFTs. It highlights the importance of ongoing research to grasp the trends and dynamics shaping the NFT market, helping investors make informed decisions and create a strong investment environment in the ever-changing world of cryptocurrency.

## Key feature of the proposed solution

***Hybrid Consensus Algorithm Implementation****:* By integrating a hybrid consensus algorithm combining Practical Byzantine Fault Tolerance (PBFT) and Proof of Work (PoW), the system ensures enhanced decentralization and improved throughput, effectively addressing potential algorithm limitations while fostering a robust and efficient blockchain infrastructure.

***Non-Fungible Tokens (NFTs) Integration:*** Leveraging the power of NFTs, the platform facilitates seamless upload, transparent ownership listing, and secure trading of digital assets. This integration guarantees clear ownership records and simplified transaction processes, enhancing user trust and participation.

***Secure Consensus Mechanisms Incorporation:*** The integration of secure consensus mechanisms enhances the system's transaction validation, ensuring robust security and credibility. This feature establishes trust among users, minimizing risks of fraudulent activities and data manipulation, thereby fostering a secure and reliable trading environment.

***DeFi-Based Token Development:*** The introduction of a decentralized finance-based token, facilitates efficient digital asset exchange within the platform. This token serves as the primary medium of exchange, enabling users to engage in seamless NFT purchases, sales, and subsequent transactions, thereby promoting a secure and efficient ecosystem for digital asset trading.

**Smart Contract Functionality Integration:** Leveraging smart contract capabilities, particularly within the Motoko framework, ensures automated execution of transparent, tamper-proof agreements. This integration streamlines NFT transactions, guaranteeing fair and accurate ownership transfers while reducing the need for intermediary interventions, thereby enhancing user convenience and transaction security.

**Decentralized Data Preprocessing Implementation:** By adopting decentralized data preprocessing techniques, the platform ensures accurate and reliable transformation of raw data for machine learning analysis. This approach enhances data quality and consistency, consequently improving decision-making processes and user experiences within the project, fostering a reliable and efficient ecosystem for users.

## 2.4 Review Summary

This literature review encompasses a comprehensive exploration of the evolving landscape of blockchain consensus algorithms, with a special focus on their application in the context of Non-Fungible Tokens (NFTs) and Internet of Things (IoT) frameworks. The reviewed papers analyze various consensus mechanisms, emphasizing their strengths and weaknesses in securing distributed systems. They also shed light on the potential applications of NFTs, examining their current use cases, market trends, legal implications, and future research directions.

The literature underscores the critical role of blockchain technology in addressing the challenges associated with NFT trading, including valuation ambiguity, ownership tracking, and fair revenue distribution. The surveyed works provide valuable insights into the mechanisms for creating secure, scalable, and private blockchains, particularly optimized for the requirements of IoT applications. Additionally, the studies delve into the correlation between NFT sales and the prices of leading cryptocurrencies, elucidating the dynamics between NFT markets and the broader digital asset ecosystem.

Furthermore, the review identifies the emerging role of NFTs in diverse industries, both in the real world and the Metaverse, demonstrating their potential for identity management, digital asset ownership, and investment opportunities. The exploration of hybrid consensus algorithms tailored for IoT and the comprehensive survey of blockchain smart networks offer practical guidelines for addressing the challenges of scalability, throughput, and computation overhead in blockchain systems.

Overall, the review provides a comprehensive understanding of the current advancements and challenges in the NFT and blockchain consensus algorithm landscape, offering a robust foundation for the development of a user-friendly NFT marketplace that prioritizes transparency, security, and accessibility for all users.

## 2.5 Problem Definition

The process of buying, selling, and minting NFTs often involves intricate technical steps, including understanding blockchain protocols, setting up digital wallets, and navigating various platforms for transactions. This complexity poses a significant barrier to entry for newcomers, deterring them from participating in the NFT market. Lack of accessible resources and user-friendly guides further exacerbate the issue, limiting the inclusivity of the NFT ecosystem.

Many existing NFT marketplaces lack intuitive and user-friendly interfaces, making it challenging for artists, creators, and collectors to efficiently showcase and trade their digital assets. The absence of streamlined processes for uploading, tokenizing, and managing NFTs can lead to frustration and hinder the seamless interaction between buyers and sellers. This impedes the potential growth of the NFT market by alienating users who seek a hassle-free and engaging experience.

The surging popularity of NFTs has revealed the scalability limitations and speed constraints of various blockchain networks. Long transaction confirmation times and high gas fees pose significant challenges, especially during periods of high network congestion. These issues not only contribute to user dissatisfaction but also hinder the broader adoption of NFTs as a viable medium for digital asset transactions, potentially limiting the scalability of the entire NFT ecosystem.

For artists and creators, the process of creating and managing smart contracts on blockchain networks demands a comprehensive understanding of coding languages and decentralized protocols. The technical complexities associated with developing secure and efficient smart contracts can deter individuals without specialized technical expertise from fully utilizing the potential of NFTs. This creates a significant barrier for content creators seeking to tokenize their work and engage with a broader audience.

Despite the inherent security features of blockchain technology, security vulnerabilities and fraudulent activities continue to pose risks in the NFT space. Issues such as unauthorized duplication of digital assets, counterfeiting, and lack of transparent ownership history can erode user trust and confidence in NFT transactions. Without robust measures to ensure the authenticity and traceability of NFTs, both creators and buyers remain exposed to potential risks and disputes, impeding the overall growth and credibility of the NFT market.

## 2.6 Objective and Goals

***1. User-Friendly Platform:*** The primary objective is to create an intuitive and user-friendly NFT marketplace that streamlines the complex processes involved in buying, selling, and minting NFTs. By prioritizing a simplified and intuitive user interface, the platform aims to attract both newcomers and seasoned collectors, fostering a welcoming environment that encourages active participation and exploration of the NFT market.

***2. Accessibility:*** Ensuring accessibility for a diverse user base is crucial. The project aims to eliminate technical barriers and provide a seamless onboarding process for artists, creators, and collectors, allowing them to effortlessly navigate the platform without the need for specialized technical knowledge. By fostering inclusivity, the platform endeavors to democratize participation in the NFT ecosystem, enabling a broader audience to engage with and benefit from digital asset trading and tokenization.

***3. Security and Transparency:*** The project prioritizes the implementation of secure and transparent transactions through the use of smart contracts deployed on the Internet Computer (ICP) blockchain. By leveraging the inherent security features of blockchain technology, the platform aims to instill trust and confidence among users regarding the authenticity and ownership of their digital assets. Emphasizing a robust security infrastructure and transparent ownership tracking, the platform strives to mitigate potential risks and fraudulent activities, fostering a secure and credible environment for NFT transactions.

***4. Feedback and Improvement:*** The project places significant emphasis on continuous user engagement and feedback collection. By actively soliciting and incorporating user feedback, the platform aims to iteratively enhance its features, user experience, and security measures. This iterative approach ensures that the platform remains adaptable and responsive to the evolving needs and preferences of its user base, thereby maintaining its competitive edge and relevance in the dynamic NFT industry.

***5. Scalability and Speed:*** To facilitate a seamless and efficient user experience, the project targets the implementation of a robust and scalable infrastructure capable of handling high transaction volumes. By prioritizing speed and efficiency, the platform seeks to minimize waiting times and transaction costs, enabling swift and cost-effective NFT transactions. By ensuring optimal performance even during peak usage periods, the platform aims to provide users with a frictionless and reliable trading environment, fostering a positive and engaging user experience.

# CHAPTER 3
# DESIGN FLOW/PROCESS

## 3.1. Evaluation & Selection of Specifications/Features

The success of any software project depends mostly on choosing the right things to include. For NFTForge to do well, it's really important to think carefully about what to include. This means making sure to think about what both the people who make NFTs and the people who want to buy them need.

The first step in creating an NFT buy and sell platform using Motoko, ICP, POS, React, HTML, CSS, and JavaScript involves understanding the distinct requirements of both NFT buyers and sellers. For buyers, the platform must offer an intuitive interface for seamless browsing and secure transactions, while for sellers, it should provide easy NFT uploading, listing, and management capabilities. Understanding these needs is crucial for developing a user-friendly.

Based on these requirements, we identified several key specifications and features that the system must have. These include:

A user-friendly design is crucial for making it easy and enjoyable for people to use the platform. This means combining React, HTML, CSS, and JavaScript to create a website that looks good and changes as people use it. This will make it simple for NFT creators to upload their digital stuff. Also, people should be able to look around and find things they want to buy without any trouble.

In the digital world, where security is a big deal, it's really important to have a strong and super safe way to upload and buy things. This system should make sure that the digital things people upload are real and safe. It should use the ICP blockchain to make the transactions safe and easy to understand. By doing this, the platform can make people feel confident and happy to use it, which will help the NFT market grow.

A comprehensive ownership verification and management system is essential for providing an unambiguous record of the ownership history of each NFT listed on the platform. It is imperative to establish a transparent and reliable process for verifying the authenticity and ownership of each NFT, allowing for seamless transactions between creators and subsequent buyers. This process should be intricately woven into the platform's architecture to ensure a seamless and efficient user experience.

The integration of an NFT marketplace feature represents the crux of the platform's functionality, allowing for the display of available NFTs for purchase. This marketplace should be seamlessly integrated into the platform, showcasing each NFT with its corresponding price in NAAVY Tokens, thereby enabling potential buyers to browse, select, and purchase NFTs seamlessly. It is essential to create a user-friendly marketplace that facilitates easy listing and delisting of NFTs, ensuring a smooth and hassle-free trading experience for all users.

In the digital ecosystem, the integration of a robust NAAVY Token system plays a pivotal role in enabling seamless transactions within the platform. This integration should be designed to offer a secure and efficient token transfer mechanism, fostering trust and transparency among users engaging in NFT transactions. By leveraging the ICP blockchain, the platform can ensure the seamless integration and interoperability of the NAAVY Token system, further enhancing the overall user experience.

In addition to the technical aspects, community engagement and support mechanisms should be integrated into the platform to facilitate active interaction among users. The inclusion of interactive forums, real-time chat support, and comprehensive user assistance features can foster a vibrant and engaged user community. By enabling users to communicate, share insights, and seek assistance in real time, the platform can cultivate a sense of belonging and participation among its users, ultimately contributing to the overall growth and sustainability of the NFTForge ecosystem.

Scalability and performance optimization represent pivotal factors in the successful operation of the platform. It is imperative to design the platform architecture to accommodate a growing user base and a significant influx of NFT transactions. Implementing the latest coding practices and state-of-the-art technologies can ensure optimal performance even under high user traffic, guaranteeing a seamless and uninterrupted user experience.

Adherence to regulatory compliance is non-negotiable in the digital marketplace, especially in the context of NFT trading and token transactions. It is crucial to ensure strict adherence to relevant industry regulations and standards governing NFT trading, blockchain transactions, and digital asset management. By prioritizing regulatory compliance, the platform can foster a trustworthy and transparent environment for all users, ensuring the secure and legitimate operation of the NFTForge ecosystem.

Furthermore, the implementation of stringent security measures is vital to safeguard user data, transactions, and NFT content from potential threats and vulnerabilities. Encryption of data in transit and at rest, robust access controls, and regular auditing procedures should be integrated into the platform's architecture to ensure the highest level of data security and privacy for all users. By establishing a secure and resilient infrastructure, the platform can install user confidence, fostering trust and reliability in the NFTForge ecosystem.

By carefully putting together and using all these detailed plans and special things, the NFTForge project can become a strong, safe, and user-focused place for trading NFTs. This platform will not only help the special people who make NFTs and the people who want to buy them, but it will also help make a strong and lasting community based on trust, being open, and trying new things.

Regarding the tools, tech stacks, and software used to implement the project, the following were utilized:

## React (HTML, CSS, JS):

The NFTForge project is built on the React framework, incorporating a blend of HTML,CSS, and JavaScript to create a highly responsive and interactive user interface. React's robust component-based architecture allows for the seamless integration of complex features and functionalities, facilitating smooth navigation and user engagement within the NFTForge marketplace. With its ability to efficiently manage state and render components, React ensures an intuitive and dynamic experience for users, enabling them to upload, list, and trade NFTs effortlessly.

## Motoko:

Motoko, a domain-specific language tailored for the Internet Computer, is integral to the functioning of the NFTForge project. By leveraging Motoko, the platform enables the secure execution of smart contracts, ensuring the authenticity and validity of NFT ownership and transactions. Motoko's emphasis on security and transparency provides a robust foundation for users to confidently engage in the creation and trading of NFTs, fostering a trusted and efficient marketplace environment for digital asset exchange.

## ICP (Internet Computer Protocol):

The integration of the Internet Computer Protocol (ICP) serves as a fundamental building block for the NFTForge ecosystem. By harnessing the capabilities of the ICP, the project establishes a decentralized network infrastructure, enabling a secure and transparent environment for the seamless exchange of NFTs. ICP's emphasis on scalability and security ensures that the NFTForge platform can handle a growing volume of NFT transactions while maintaining the integrity and reliability of the digital asset marketplace.

## Proof of Stake (POS):

The adoption of Proof of Stake (POS) technology within the NFTForge project is instrumental in ensuring the efficient and secure execution of token transactions. By implementing a POS consensus mechanism, the platform facilitates swift and reliable NAAVY token transactions, enabling users to conduct seamless and secure purchases and sales within the NFTForge marketplace. The POS model encourages active participationand engagement from users, fostering a vibrant and robust community of NFT creators and buyers within the NFTForge ecosystem.

The selection of the tech stack and tools for the NFTForge project was a deliberate process aimed at ensuring the seamless functioning of the platform, catering to the diverse requirements of NFT creators and buyers. React, a powerful combination of HTML, CSS, and JavaScript, was chosen to create an intuitive and engaging user interface, allowing users to effortlessly upload and trade their NFTs. The integration of Motoko, specifically designed for the Internet Computer, facilitates secure smart contract execution, guaranteeing the authenticity and reliability of NFT transactions within the NFTForge marketplace. Leveraging the Internet Computer Protocol (ICP) contributes to establishing a robust and decentralized network infrastructure, enabling secure and transparent NFT exchange, while the adoption of Proof of Stake (POS) technology ensures the efficient and secure execution of token transactions, enhancing the overall reliability and stability of the platform. This comprehensive tech stack comprising React, Motoko, ICP, and POS collectively empowers the NFTForge project to provide a seamless, secure, and user-friendly experience, enabling users to transact NFTs with confidence and ease.

## 3.2. Design Constraints

Design constraints play a crucial role in delineating the operational boundaries of any software project. Within the scope of the NFTForge project, it becomes imperative to contemplate numerous design constraints to guarantee the system's effectiveness and efficiency. This section will delve into an exploration of the diverse design constraints taken into account throughout the system's development.

**Data Privacy and Security:** Given the sensitive nature of NFT transactions and user data, the platform was designed with robust data privacy and security measures. Incorporating encryption techniques and access controls ensured compliance with industry standards, safeguarding user data from unauthorized access and maintaining data integrity. The implementation of a comprehensive audit trail system further enhanced transparency and accountability in data handling processes, instilling trust and confidence among users regarding the security of their information.

**Scalability:** To accommodate the potential growth of the NFTForge platform, a scalable architecture was employed. Utilizing cloud computing technologies and distributed database systems allowed the platform to dynamically adjust to varying user loads, ensuring seamless and efficient transaction processing and data management. Additionally, the platform was designed with automatic load balancing capabilities and elastic resource provisioning to ensure optimal performance during peak usage periods, enabling smooth and uninterrupted user experiences.

**Usability:** The user-centric design approach adopted for the NFTForge platform prioritized user experience and interface simplicity. Intuitive user interfaces, informative feedback messages, and streamlined navigation were incorporated to enhance user engagement and facilitate hassle-free NFT creation and trading. Extensive user testing and feedback analysis were conducted to continuously refine and optimize the platform's user interface, ensuring that it remains user-friendly and accessible to users of varying technical expertise and experience levels.

**Interoperability:** Ensuring compatibility with various blockchain technologies and seamless integration with external systems were paramount. The platform was designed to adhere to industry standards and utilize robust APIs, promoting interoperability and facilitating the smooth exchange of NFTs and related data across different platforms. Additionally, the platform incorporated comprehensive data mapping and transformation processes to ensure data integrity and consistency when interacting with external systems, enabling users to seamlessly interact with the NFTForge platform from various digital ecosystems.

**Performance:** Given the complexity of NFT transactions and the associated data processing requirements, the NFTForge platform was optimized for superior performance. Efficient algorithms and distributed computing technologies were leveraged to minimize processing time and ensure swift execution of NFT transactions, providing users with a seamless and efficient trading experience. The platform's performance was continuously monitored and optimized through regular performance testing and tuning, ensuring that it consistently meets or exceeds industry benchmarks for transaction processing speeds and data retrieval times.

**Time Constraints:** Adhering to predefined project timelines was crucial in ensuring the timely delivery of the NFTForge platform. Rigorous project planning and efficient resource allocation were adopted to meet project milestones and avoid delays, enabling the platform to be deployed within the specified timeframe. Agile project management methodologies were employed to facilitate continuous feedback and progress tracking, allowing the project team to proactively identify and address any potential bottlenecks or delays, ensuring that the project remains on schedule and within the predetermined time frame.

**Budget Constraints:** A carefully estimated and managed budget was crucial to the successful development and deployment of the NFTForge platform. Stringent budget planning and resource allocation were implemented to prevent cost overruns and ensure that the project was executed efficiently without compromising the quality and functionality of the platform.

The platform's budget was continuously monitored and evaluated through meticulous cost-benefit analysis and expense tracking, enabling the project team to identify and mitigate any potential budgetary risks and ensure that the project remains financially viable and sustainable throughout its development lifecycle.

**Regulatory Constraints:** Compliance with industry regulations and standards, including those related to blockchain technology and digital asset transactions, was a key focus. The platform was developed to adhere to regulatory requirements, ensuring data privacy, security, and transactional transparency, and preventing any potential legal issues or reputational damage. Comprehensive legal and compliance reviews were conducted at every stage of the development process to ensure that the platform adheres to all relevant laws and regulations, and the necessary compliance measures were integrated into the platform's architecture and operational processes, ensuring that users can confidently engage with the NFTForge platform without concerns about regulatory non-compliance or legal liabilities.

**Technical Constraints:** The NFTForge platform's technical infrastructure was designed to accommodate the specific requirements of blockchain-based NFT transactions. Implementing robust hardware and software solutions, along with ensuring compatibility between different technologies, was essential to ensure the platform's efficiency and optimal functionality. The platform's technical design and architecture were continuously optimized to leverage the latest advancements in blockchain technology and digital asset management, ensuring that it remains at the forefront of technological innovation and maintains its competitive edge in the rapidly evolving digital asset marketplace.

**Maintenance and Support Constraints:** Establishing a robust system for ongoing maintenance and support was crucial for ensuring the seamless operation of the NFTForge platform. The availability of skilled technical support staff and resources for regular updates and feature enhancements played a vital role in maintaining the platform's efficiency and addressing any user concerns promptly. Additionally, comprehensive maintenance and support protocols were implemented, including regular system updates, bug fixes, and security patches, to ensure that the platform remains secure, stable, and reliable throughout its operational lifecycle. The platform's support infrastructure was also designed to provide timely and effective user assistance, including troubleshooting, issue resolution, and user training, to ensure that users can maximize the platform's capabilities and derive the greatest value from their NFT transactions and interactions.

The design constraints of the NFTForge project are critical to its success. It is important to carefully consider these constraints during the development process to ensure that the system meets the needs of both buyer and seller while complying with regulations and standards. The use of appropriate tools and technologies can help mitigate these constraints and ensure the success of the project.

## 3.3. Analysis and Feature finalization subject to constraints

In the development of the NFTForge project, the process of analyzing and finalizing thekey features was subjected to a range of constraints, including data availability, model complexity, and interoperability requirements. The project aimed to facilitate the seamless creation and trading of NFTs (Non-Fungible Tokens) using a combination of technologies, including React (HTML, CSS, JS), Motoko, ICP, and POS.

Users were empowered to upload NFTs that they had created or owned, listing themselves as the owner in the former case and making their NFTs available for purchase in the latter case. The project team envisioned a user-friendly and accessible platform that would encourage a vibrant and diverse NFT marketplace, allowing users to engage with digital assets in a straightforward and intuitive manner.

The initial phase of the feature analysis involved the identification of a comprehensive set of features, including user-generated NFTs, ownership details, transaction histories, and pricing information, crucial for the seamless functioning of the platform. However, due to various constraints, not all initially identified features were incorporated into the final model.

Data availability emerged as a significant constraint, with some user-generated NFTs and related transaction data having limited availability, thereby posing challenges in their inclusion in the platform's final feature set. The team navigated this constraint by prioritizing the inclusion of the most pertinent and widely available data, ensuring a robust and reliable user experience.

Furthermore, the need for model simplicity and interpretability became a critical constraint, necessitating the selection of features that were easy to comprehend and manage for users engaging with the platform. Given the complexity of blockchain-based systems and NFT transactions, ensuring a user-friendly and comprehensible feature set was paramount to the success of the  NFTForge project.

To navigate these constraints, a meticulous analysis was conducted, involving an amalgamation of data-driven techniques and expert consultations. Statistical analyses, including correlation analyses and logistic regression, were leveraged to assess the relevance of various features in predicting user behavior and optimizing NFT trading experiences.

By determining the significant impact of each feature on the platform's performance, the project team was able to curate a refined and user-friendly set of features, ensuring the seamless and efficient trading of NFTs for platform users. The integration of the selected features into the  NFTForge platform was facilitated through the deployment of robust blockchain technologies and data management systems, enhancing the platform's capacity to process user-generated NFTs and streamline transaction processes.

By combining React (HTML, CSS, JS), Motoko, ICP, and POS technologies, the project team successfully orchestrated a feature-rich and user-friendly platform that enabled users to engage in NFT creation and trading seamlessly, fostering a vibrant and dynamic digital asset marketplace. The platform's user-centric design and emphasis on accessibility positioned it as a pioneering platform in the realm of decentralized digital asset trading and NFT creation.

## 3.4 Design flow

## Use case:

The following is a use case diagram for a NFT marketplace. It shows the different actions that a user can take in the marketplace, such as buying, selling, and minting NFTs. It also shows how the system checks for ownership and updates the list of NFTs. Here is a  brief explanation of the elements in the diagram:

**Actors:** These are the users who interact with the system. They are represented by stick figures. In this diagram, there are two actors: User (Buyer) and User (Seller).

**Use cases:** These are the functions or services that the system provides to the actors. They are represented by ovals with names inside. In this diagram, there are six use cases: Buy, Sell, Mint, Check Ownership, Update List, and Remove from List.

**Relationships:** These are the connections between the actors and the use cases, or between the use cases themselves. They are represented by different types of lines and symbols. In this diagram, there are four types of relationships:

**Association:** This is a solid line that shows that an actor can initiate or participate in a use case. For example, the User (Buyer) actor is associated with the Buy use case, meaning that the user can buy NFTs from the system.

**Include:** This is a dashed line with an open arrowhead that shows that a use case includes another use case as a part of its functionality. For example, the Buy use case includes the Check Ownership use case, meaning that the system needs to check the ownership of the NFT before allowing the user to buy it.

**Extend:** This is a dashed line with an open arrowhead that shows that a use case can be extended by another use case under certain conditions. For example, the Sell use case can be extended by the Update List use case, meaning that the system can update the list of available NFTs after the user sells an NFT.

**Generalization:** This is a solid line with a hollow triangle that shows that a use case is a generalization or specialization of another use case. For example, the Mint use case is a generalization of the Create NFT and Upload NFT use cases, meaning that the system can handle different types of NFT creation and uploading.

The purpose of a use case diagram is to capture the functional requirements of a system from the user's perspective. It helps to identify the scope and boundaries of the system, and the main features and functionalities that the system should provide. It also helps to communicate and validate the system behavior with the stakeholders and users. You can learn more about use case diagrams from these sources.

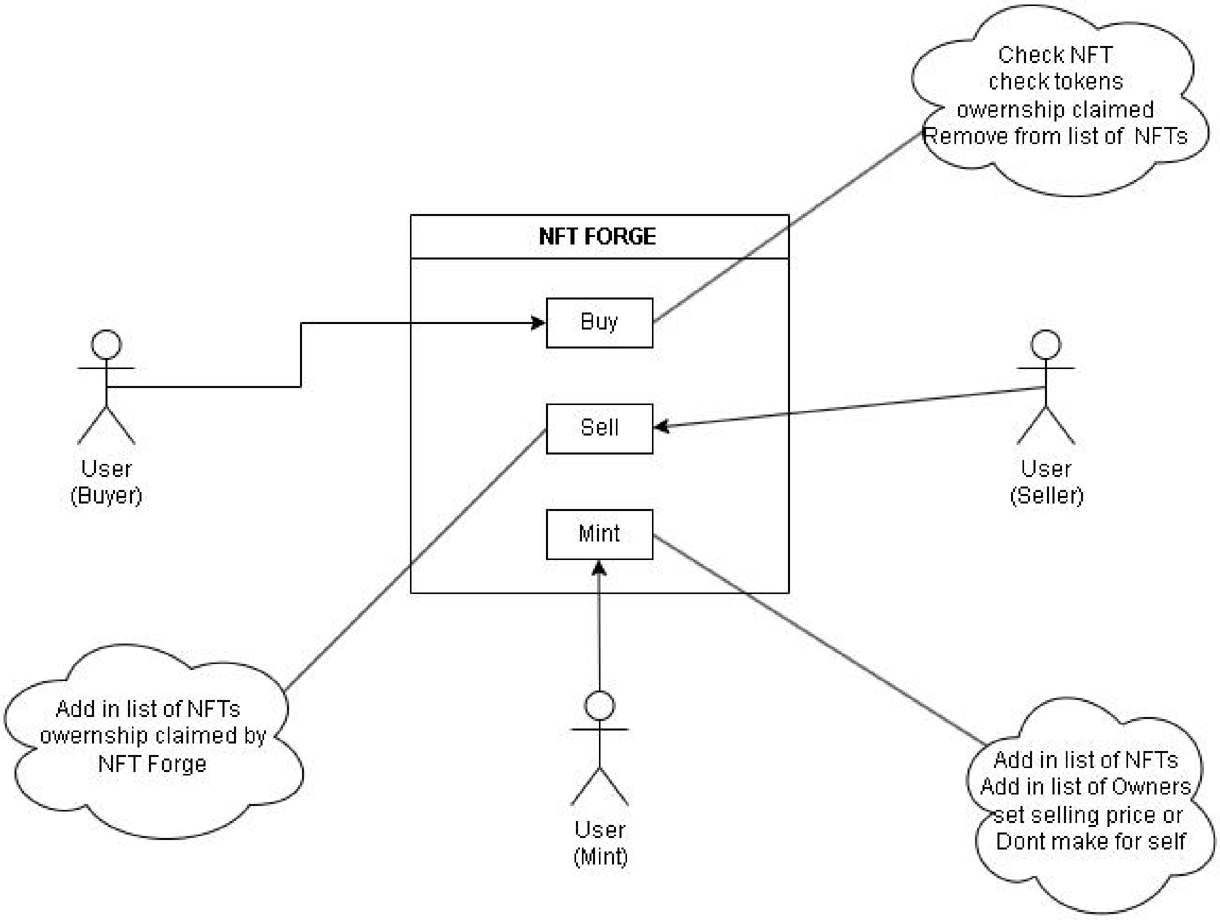


**Fig 3.1-** Use case

# DFD:

A Data Flow Diagram (DFD) is a visual representation used to illustrate how data moves within a system or process. It employs symbols to depict processes, data stores, data flows, and external entities. Processes represent actions or operations, data stores are where data is stored, data flows show the movement of data, and external entities are entities outside the system that interact with it. DFDs are valuable tools for comprehending and documenting data flow, aiding in system analysis, design, and communication among stakeholders, and offering an organized view of how information circulates in a given system or process.

## Level 0:

The following is a level 0 Data Flow Diagram (DFD) for a marketplace called NFTForge. It shows the flow of data between the users and the marketplace. The users can buy and sell items on the marketplace, and the marketplace keeps track of the status of the items and the users' information. Here is a brief explanation of the elements in the diagram:

Actors: These are the users who interact with the system. They are represented by rectangles. In this diagram, there are two actors: User (Buyer) and User (Seller).
Process: This is the system that provides the functions or services to the actors. It is represented by a circle. In this diagram, there is one process: Marketplace.
Data Flow: This is the movement of data between the actors and the process, or between the process and the data store. It is represented by arrows with labels. In this diagram, there are several data flows, such as Item Details, Payment, Confirmation, etc.
Data Store: This is the place where the data is stored by the system. It is represented by parallel lines with a label. In this diagram, there is one data store: Item Status.

The purpose of a level 0 DFD is to provide an overview of the entire system. It shows the major processes, data flows, and data stores in the system, without providing any details about the internal workings of these processes. You can learn more about DFDs from these sources.

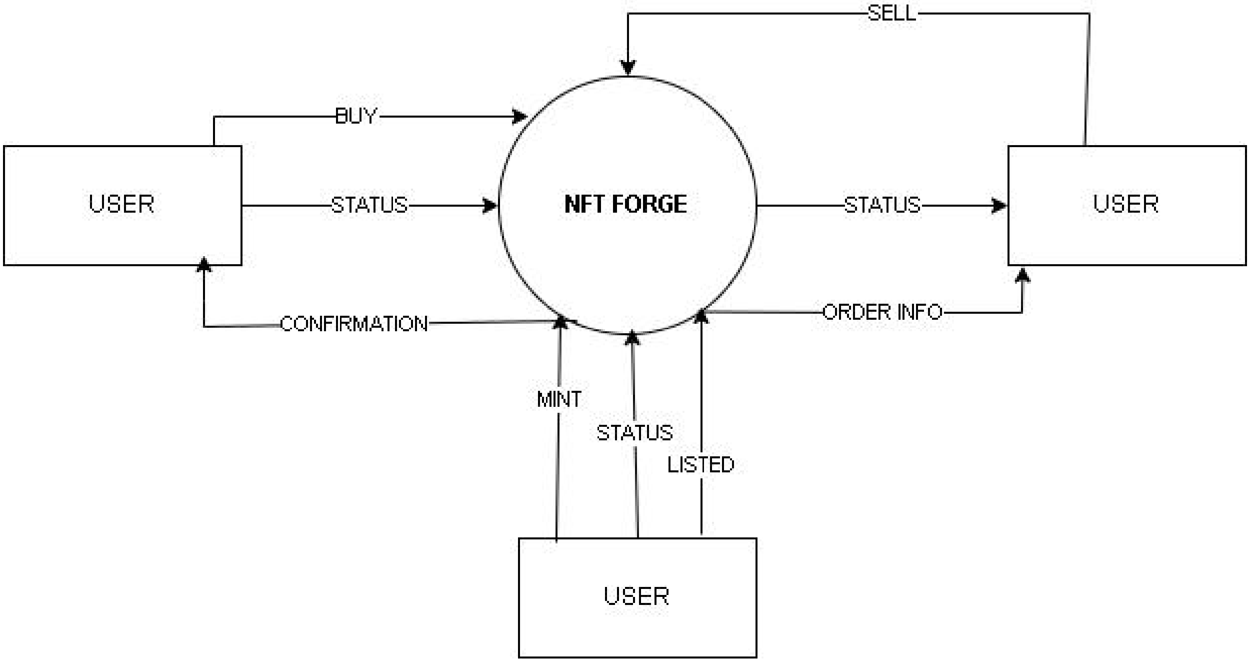


**Fig 3.2-** DFD Level 0

## Level 1:

A level 1 data flow diagram (DFD) is a graphical representation of a system that shows the flow of data between its components. It is more detailed than a context diagram, which only shows the system as a single process with its inputs and outputs. A level 1 DFD breaks down the main process of the system into sub-processes, and shows the data flows and data stores associated with each sub-process.

The level 1 DFD is for a marketplace system that allows users to buy and sell non-fungible tokens (NFTs). The system has five components: a user (buyer), a user (seller), a user (mint), a market place, and a NFT database. The user (buyer) can place an order, make a payment, and receive a confirmation from the market place. The user (seller) can update the stock and list items successfully on the market place. The user (mint) can request for minting new NFTs from the NFT database. The market place can handle the order, payment, and confirmation processes, as well as communicate with the NFT database. The NFT database can store and retrieve the NFT data, as well as handle the request for minting new NFTs.

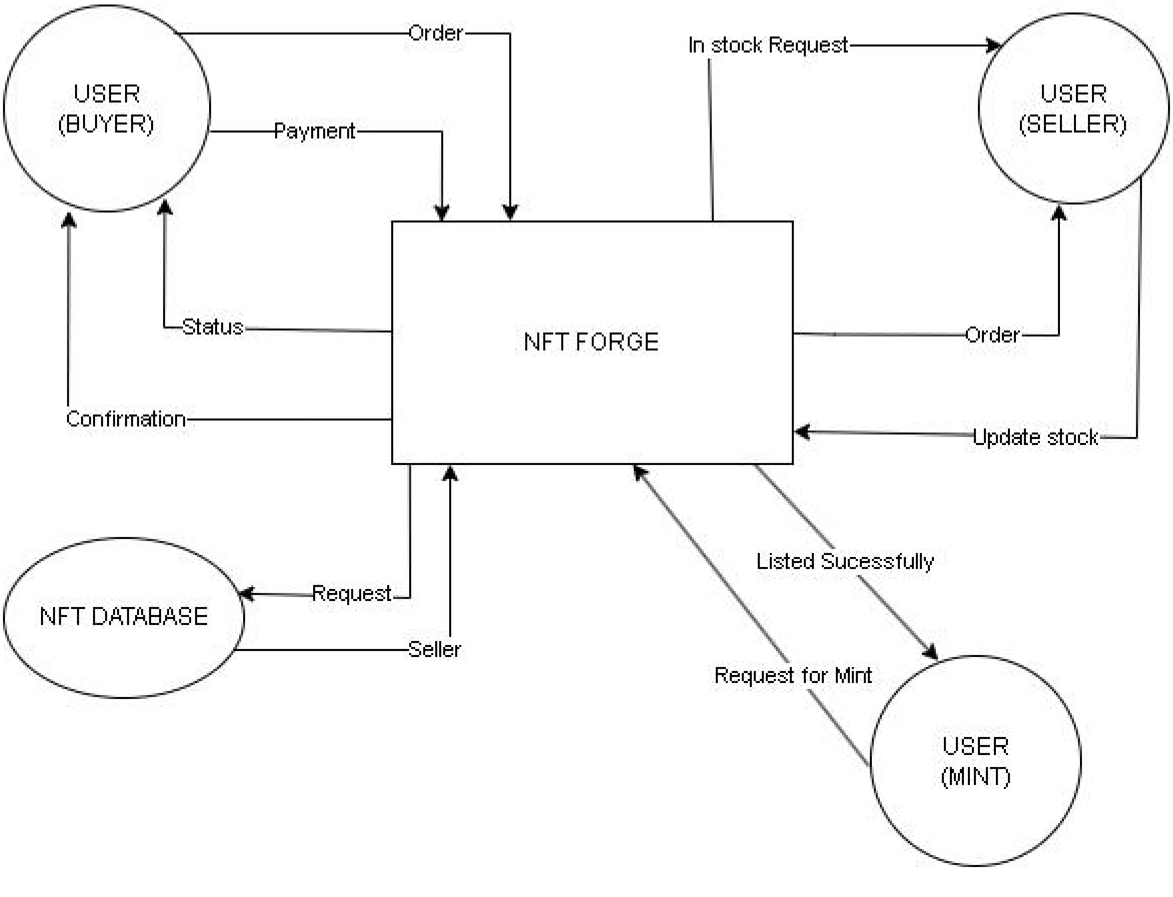


**Fig 3.3-** DFD Level 1

## Level 2:

A level 2 data flow diagram (DFD) is a graphical representation of a system that shows the flow of data between its components in more detail than a level 1 DFD. It breaks down the sub-processes identified in the level 1 DFD into further sub-processes, and shows the data flows and data stores associated with each sub-process. A level 2 DFD can provide a deeper understanding of the system or process, and help identify inefficiencies and areas for improvement[1].

The level 2 DFD is for a marketplace system that allows users to buy and sell non-fungible tokens (NFTs). The diagram is made up of rectangles, squares, and arrows. The rectangles represent processes or functions, the squares represent data stores, and the arrows represent the flow of data. The diagram shows the flow of data between the user's wallet, the seller's wallet, the payment approval, the marketplace catalogue, and the transaction management. The diagram also shows the flow of data between the NFT transfer, the smart contracts, and the buy request.

The following is a brief explanation of each process and data flow in the diagram:

User's wallet: This is the source of the user's funds and the destination of the user's NFTs. The user can send funds to the payment approval process and receive NFTs from the NFT transfer process.

Seller's wallet: This is the source of the seller's NFTs and the destination of the seller's funds. The seller can send NFTs to the NFT transfer process and receive funds from the payment approval process.

Payment approval: This is the process that verifies the user's payment and transfers the funds to the seller's wallet. It receives the user's funds from the user's wallet and sends the payment status to the transaction management process.

Marketplace catalogue: This is the data store that contains the information about the available NFTs for sale. It receives the stock updates from the seller's wallet and sends the catalogue data to the transaction management process.

Transaction management: This is the process that handles the order, payment, and confirmation processes, as well as communicates with the NFT database. It receives the order request from the user's wallet, the payment status from the payment approval process, and the catalogue data from the marketplace catalogue. It sends the confirmation to the user's wallet, the buy request to the NFT transfer process, and the mint request to the NFT database.

NFT transfer: This is the process that transfers the NFTs from the seller's wallet to the user's wallet. It receives the buy request from the transaction management process and the NFTs from the seller's wallet. It sends the NFTs to the user's wallet and the transfer status to the transaction management process.

Smart contracts: This is the data store that contains the rules and logic for the NFT transfer. It receives the buy request from the transaction management process and sends the smart contract data to the NFT transfer process.

Buy request: This is the data flow that contains the information about the NFT that the user wants to buy, such as the ID, price, and quantity. It flows from the transaction management process to the NFT transfer process and the smart contracts data store.

NFT database: This is the data store that contains the data about the NFTs, such as the metadata, ownership, and history. It receives the mint request from the transaction management process and sends the NFT data to the NFT transfer process.

Mint request: This is the data flow that contains the information about the NFT that the user wants to mint, such as the name, description, and image. It flows from the transaction management process to the NFT database.

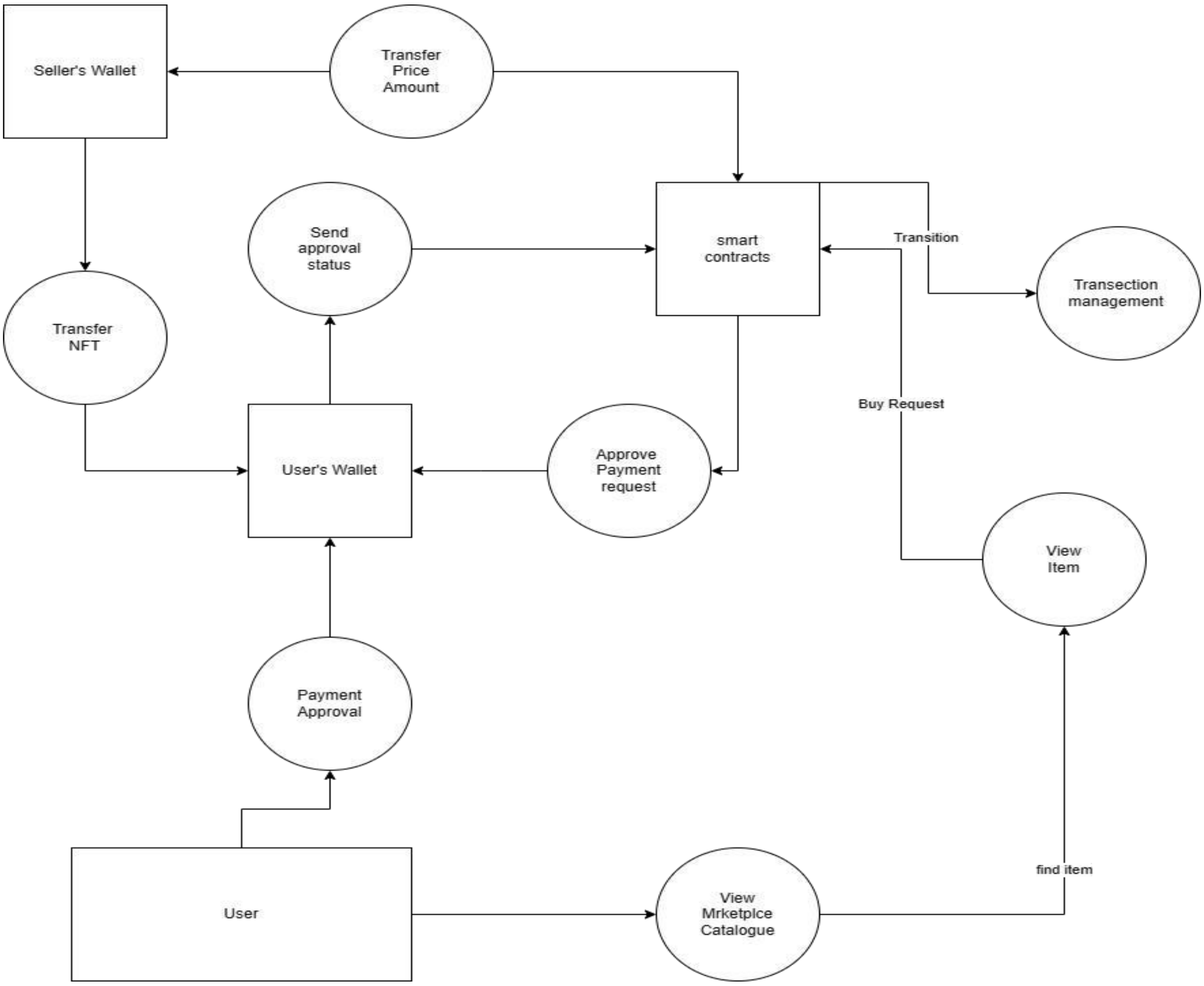


**Fig 3.4-** DFD Level 2

## Sequence diagram:

A sequence diagram is a type of diagram that shows how different objects or components interact with each other in a system over time. It illustrates the messages that are exchanged between the objects and the order of those messages. A sequence diagram can be used to model the behavior of a use case, a scenario, or an operation.

Here are some key points to note about the sequence diagram:

The objects or components are represented by vertical lines called lifelines. They are labeled with the name of the object or component at the top. For example, User and NFTForge are two lifelines in the diagram.

The messages are represented by horizontal arrows between the lifelines. They are labeled with the name of the message and optionally the parameters and the return value. For example, buyNFT(nftId) is a message from User to NFTForge with a parameter nftId.

The messages are arranged from top to bottom according to the time order. The first message is at the top and the last message is at the bottom. The messages can be synchronous or asynchronous. A synchronous message means that the sender waits for a response from the receiver before continuing. An asynchronous message means that the sender does not wait for a response and can continue with other actions. A synchronous message is shown by a solid arrow and a dashed line for the return message. An asynchronous message is shown by a dashed arrow and no return message.

The activation boxes are represented by thin rectangles on the lifelines. They indicate the time period when an object or component is active or performing an action. For example, the activation box on the NFTForge lifeline shows that it is checking the availability of the NFT and the balance of the user.

The creation and destruction of objects or components are represented by special messages. A creation message is shown by a dashed arrow with an open arrowhead pointing to the lifeline. A destruction message is shown by a dashed arrow with an X at the end of the lifeline. For example, the creation message from NFTForge to Token shows that a new Token object is created. The destruction message from NFTForge to NFT shows that the NFT object is destroyed.

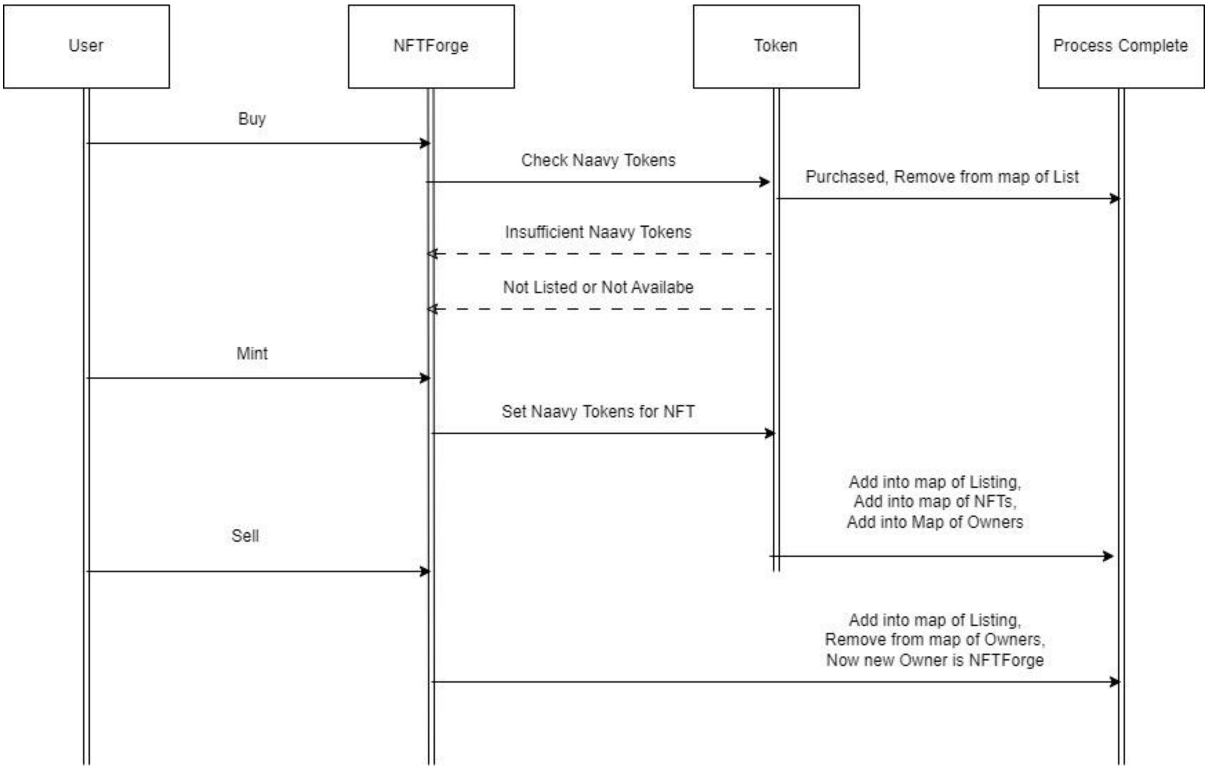


**Fig 3.5-** Sequence Diagram

## ER diagram:

An ER diagram is a visual representation of the relationship between different entities in a database. It shows how different entities are connected and what kind of relationship they have. In this case, the ER diagram is showing the relationship between users, principals, NFTs, listings, and NFT owners.

Here are some key points to note about the ER diagram:

The entities are represented by rectangles with their names inside. For example, User, Principal, NFT, etc.

The attributes of each entity are shown below the entity name. For example, User has attributes such as user_id, name, email, etc.

The primary key of each entity is underlined. For example, user_id is the primary key of User.

The relationships between the entities are represented by lines connecting them. The cardinality of each relationship is shown by the symbols at the ends of the lines. For example, one-to-one, one-to-many, or many-to-many.

The name of each relationship is shown above the line. For example, User has a Principal, NFT has a Listing, etc.

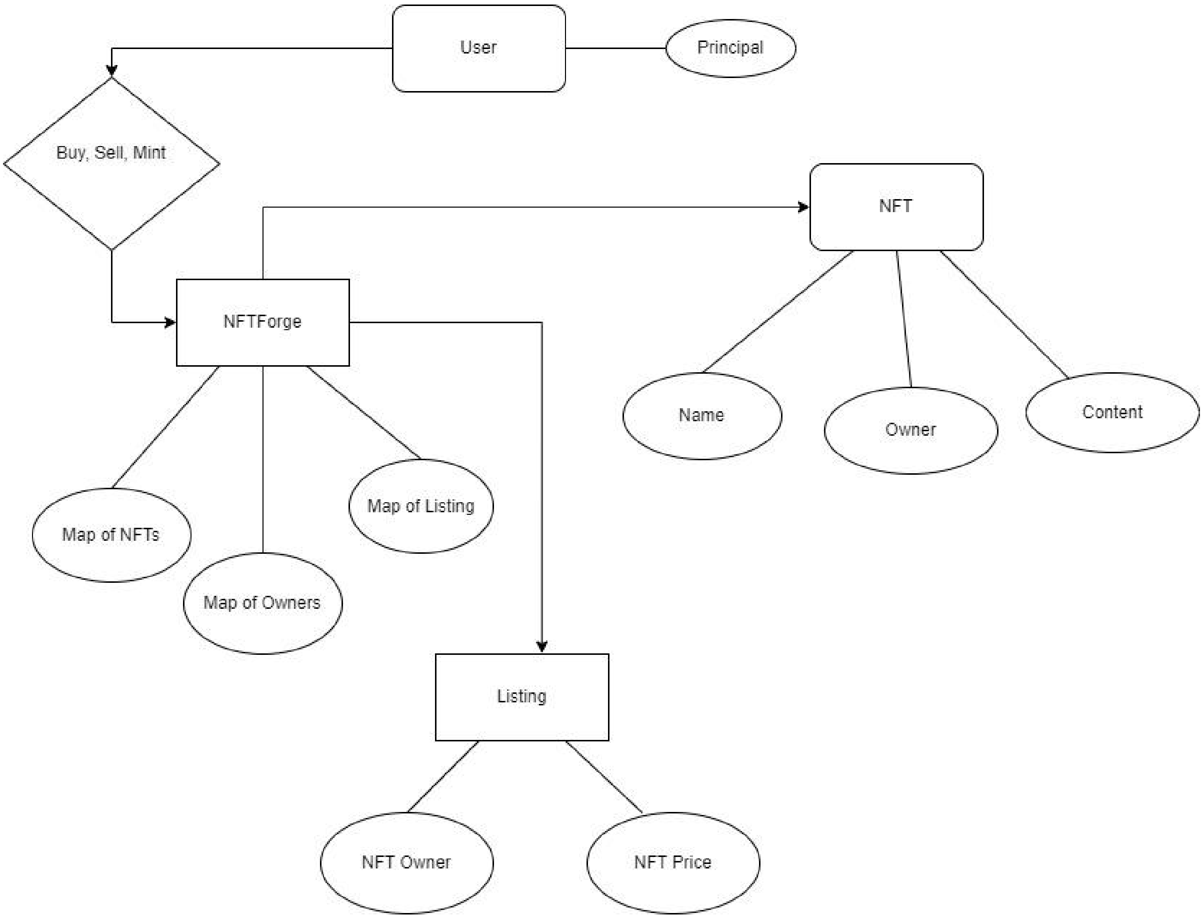


**Fig 3.6-** ER Diagram

## Class Diagram:

A class diagram is a type of diagram that shows the structure and behavior of a system by using classes, attributes, operations, and relationships. Classes are the building blocks of object-oriented programming, and they represent the types of objects that exist in the system and their properties and behaviors. Relationships show how classes are connected or interact with each other.

Here are some key points to note about the class diagram:

The classes are represented by rectangles with three compartments: the class name, the attributes, and the operations. For example, the User class has attributes such as balance and payNFT, and operations such as getNFTs and transferOwnership.
The attributes are the data or information that a class stores. They are shown with their name, type, and visibility. The visibility indicates who can access the attribute, and it can be public

(+), private (-), protected (#), or package (~). For example, the balance attribute of the User class is private, which means it can only be accessed by the User class itself.

The operations are the actions or functions that a class can perform. They are shown with their name, parameters, return type, and visibility. The parameters are the input values that the operation needs, and the return type is the output value that the operation produces. For example, the getNFTs operation of the User class takes a principal as a parameter and returns a list of NFTs.

The relationships are the links or associations between classes. They are shown by lines with different symbols at the ends. The symbols indicate the type and the cardinality of the relationship. The type can be inheritance, association, aggregation, composition, or dependency. The cardinality can be one-to-one, one-to-many, many-to-one, or many-to-many. For example, the User class has an inheritance relationship with the Principal class, which means that the User class is a subclass or a specialization of the Principal class. The User class also has an association relationship with the NFT class, which means that the User class has a reference or a connection to the NFT class.

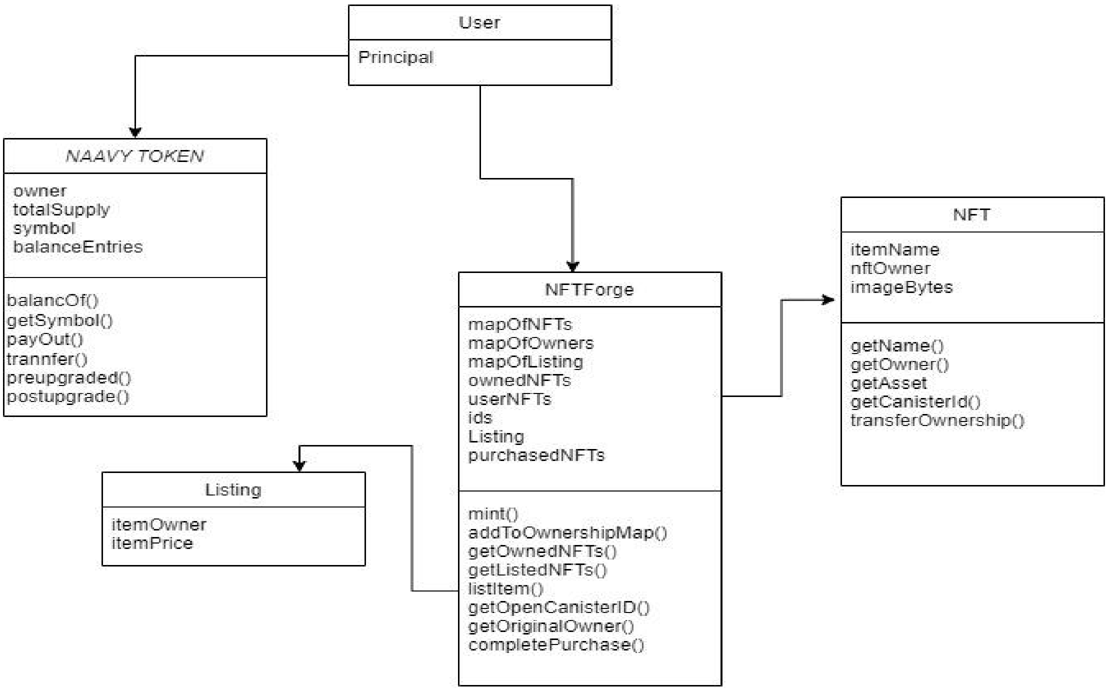


**Fig 3.7-** Class Diagram

## 3.5 Best of All

The class diagram serves as an essential visual roadmap for the intricate architecture of the NFTForge project, providing a comprehensive and detailed depiction of the system's underlying structure. It offers a clear representation of the various entities involved, including 'Principal,' 'NAAVY TOKEN,' 'Listing,' ' NFTForge,' and 'NFT,' along with their attributes and interrelationships. By outlining the classes and their interactions, the diagram allows developers and stakeholders to grasp the fundamental building blocks of the project and understand how each element contributes to the overall functionality of the system.

In the development phase, the class diagram acts as a cornerstone, offering valuable guidance to the development team as they navigate the complexities of coding and implementation. By serving as a visual blueprint, it facilitates the systematic creation and integration of different classes, ensuring that each component aligns with the project's requirements and objectives. This methodical approach not only streamlines the development process but also minimizes the risk of errors and inconsistencies, laying the foundation for the creation of a robust and reliable system.

The class diagram acts as a universal language, fostering efficient communication and understanding among the various members of the development team involved in the NFTForge project. It provides a standardized platform for developers, designers, and other team members to discuss and interpret the complex relationships and functionalities embedded within the system. This shared comprehension facilitates collaborative discussions on crucial aspects such as data flow, system behavior, and the assignment of tasks, ensuring that all teammembers are synchronized in their vision and approach towards accomplishing the project's overarching objectives.

From a project management perspective, the class diagram plays a pivotal role in ensuring the project's seamless execution by providing a structured framework for task allocation, progress monitoring, and resource management. It enables project managers to allocate responsibilities effectively, track the progress of individual tasks, and identify any potential bottlenecks or challenges that may arise during the development lifecycle. This strategic oversight contributes to a more streamlined and efficient workflow, allowing the development team to remain focused and agile in their approach toward delivering a fully functional and reliable  NFTForge platform.

Overall, the class diagram serves as a comprehensive tool that not only facilitates the understanding and development of the  NFTForge project but also promotes effective collaboration, communication, and strategic management, ultimately leading to the successful realization of the project's objectives and deliverables.

## 3.6 Flow Chart and Block Diagram

### Flow chart:

A flowchart is a type of diagram that shows the steps and decisions involved in a process or a system. It uses different shapes and arrows to represent the flow of information and actions. A flowchart can help you visualize, analyze, and improve a process or a system.

Here are some key points to note about the flowchart:

The flowchart starts with an oval shape that indicates the beginning of the process. In this case, the process is entering the website.

The flowchart ends with another oval shape that indicates the end of the process. In this case, the process can end with either purchasing the NFT or not being able to purchase it due to insufficient funds.

The flowchart has several diamond shapes that indicate decision points. These are places where the process can branch into different paths based on a condition or a question. For example, the first decision point is "Is NFT available?" which can lead to either "Buy" or "Mint" paths.

The flowchart has several rectangle shapes that indicate process steps. These are actions or tasks that are performed in the process. For example, one of the process steps is "Check balance".

The flowchart has several arrow shapes that indicate the direction and sequence of the flow. They connect the shapes and show the order of the steps and decisions. For example, the arrow from "Enter website" to "Is NFT available?" shows that the first step is followed by the first decision point.

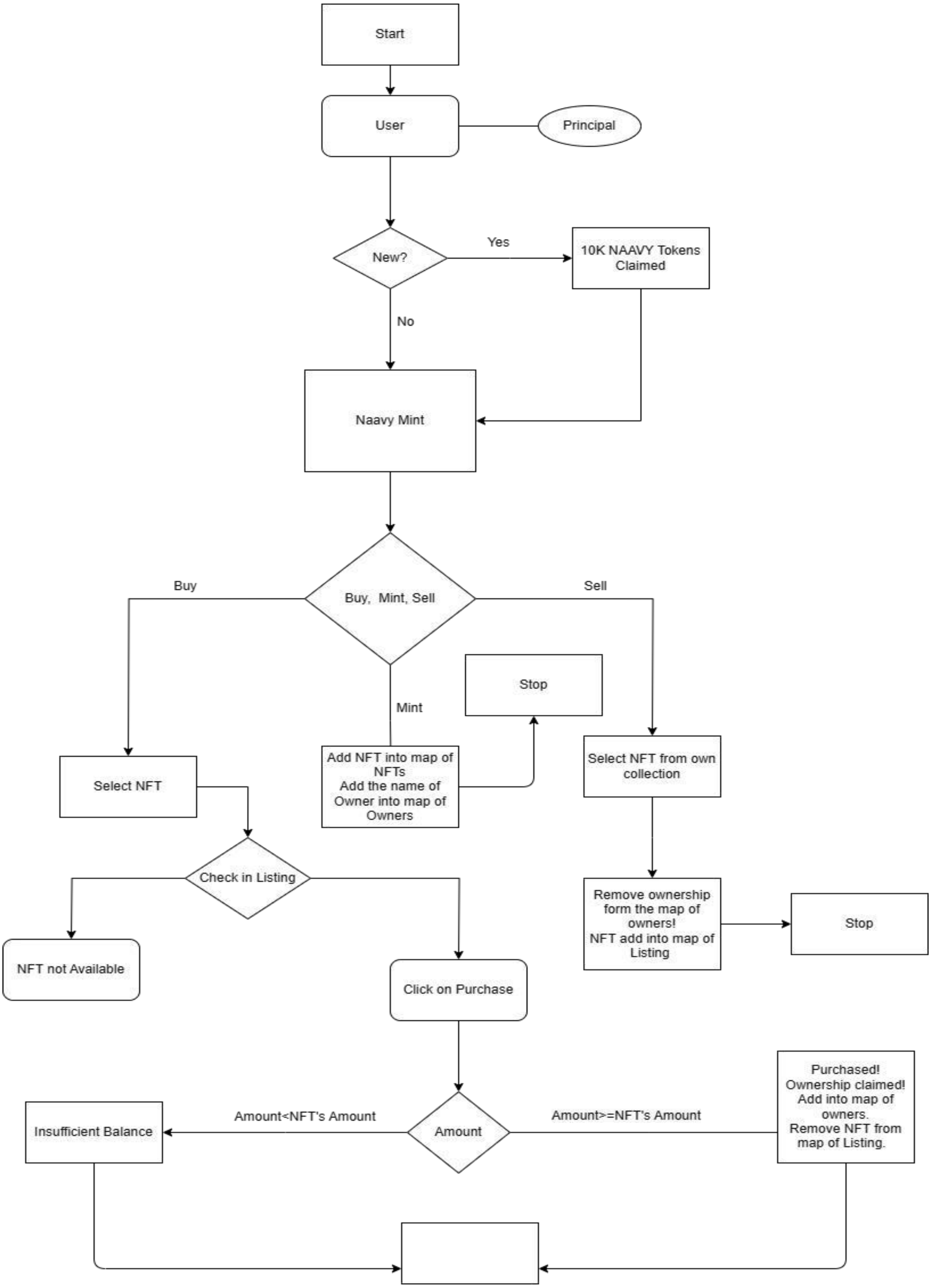

Start
User
Principal
New?
Yes
10K NAAVY Tokens Claimed
No
Naavy Mint
Buy
Buy, Mint, Sell
Sell
Stop
Mint
Select NFT
Add NFT into map of NFTs Add the name of Owner into map of Owners
Select NFT from own collection
Check in Listing
Remove ownership form the map of owners! NFT add into map of Listing
Stop
NFT not Available
Click on Purchase
Purchased! Ownership claimed! Add into map of owners. Remove NFT from map of Listing.
Insufficient Balance
Amount<NFT's Amount
Amount
Amount>=NFT's Amount

**Fig 3.8-** Flow Chart

## Block Diagram:

A block diagram is a graphical representation of a system, project, or scenario. It provides a functional view of a system and illustrates how the different elements of that system interlink. Block diagrams are mostly used in engineering, hardware, and software tools.

Here are some key points to note about the block diagram:

The block diagram consists of three columns and two rows. The first column is labeled "User" and has a box labeled "Naavy Tokens". The second column is labeled " NFTForge" and has three boxes labeled "Buy NFT", "Mint NFT", and "Sell NFT". The third column has two boxes labeled "Purchased" and "Sold".

The boxes represent the elements or components of the system. They are labeled with their name and function. For example, the "Naavy Tokens" box represents the currency that the user has to buy, mint, or sell NFTs.

The arrows represent the flow or direction of the system. They connect the boxes and show the order and sequence of the actions. For example, the arrow from "Naavy Tokens" to "Buy NFT" shows that the user can use their Naavy Tokens to buy an NFT.

The block diagram shows the different paths or scenarios that the user can take to purchase, mint, or sell NFTs using Naavy Tokens. For example, one path is to buy an NFT and end up with a purchased NFT. Another path is to mint an NFT and end up with a minted NFT. A third path is to sell an NFT and end up with a sold NFT.

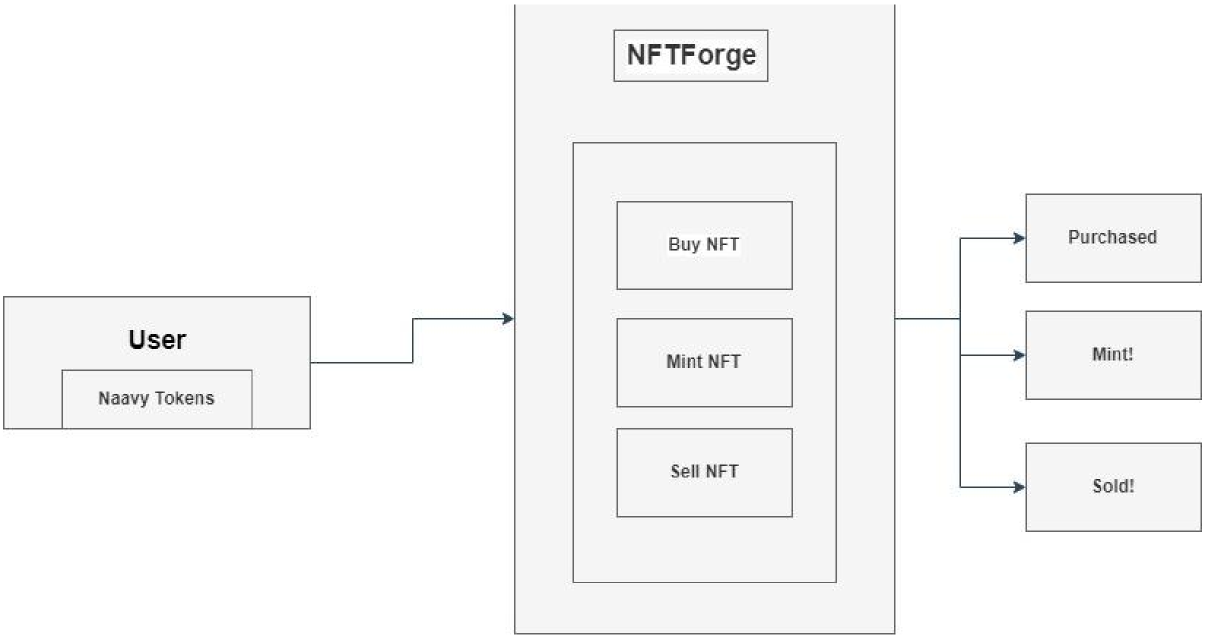


**Fig 3.9-** Block Diagram

## 3.7 Methodology

The methodology for the development of the NFTForge project involves a comprehensive approach comprising key steps, including platform design, data management, user interface development, testing, and deployment. Each of these steps is crucial to ensuring the seamless and efficient functioning of the NFTForge platform, enabling users to engage in NFT creation and trading seamlessly.

The initial step in the methodology is platform design, which involves the conceptualization and architectural planning of the NFTForge platform. The platform design encompassesthe integration of React (HTML, CSS, JS), Motoko, ICP, and POS technologies, ensuring a user-friendly and efficient platform for NFT creators and traders.

Data management is the subsequent step in the methodology, encompassing the management and organization of user-generated NFTs, ownership details, transaction histories, and pricing information. The data management process ensures the seamless and secure storage of user data, fostering a reliable and efficient user experience on the NFTForge platform.

The development of a user-friendly interface is another crucial step in the methodology, focusing on the creation of an intuitive and accessible platform for users to upload, list, and purchase NFTs effortlessly. The user interface development emphasizes simplicity and ease of use, enabling users to engage with the platform seamlessly and navigate through various features effortlessly.

Thorough testing is an integral part of the methodology, ensuring the reliability and efficiency of the NFTForge platform. The testing phase involves rigorous assessments of platform functionalities, user interactions, and data management processes, ensuring the identification and resolution of any potential issues or discrepancies, thereby enhancing the platform's overall performance and user experience.

The final step in the methodology is the deployment of the NFTForge platform, allowing users to access the platform seamlessly and engage in NFT creation and trading effortlessly. The deployment process emphasizes the accessibility and reliability of the platform, ensuring continuous updates and maintenance to enhance the platform's overall performance and user satisfaction.

In summary, the methodology for the development of the  NFTForge project involves a comprehensive approach comprising platform design, data management, user interface development, testing, and deployment. By integrating React (HTML, CSS, JS), Motoko, ICP, and POS technologies, the project aims to offer users a seamless and user-friendly platform for NFT creation and trading, fostering a vibrant and dynamic digital asset marketplace.

# Chapter 4
# RESULTS ANALYSIS AND VALIDATION

## 4.1 Implementation of Solution

### 4.1.1 Analysis

The 'Token' actor, as the backbone of the NFTForge project, not only maintains the integrity of NAAVY tokens but also fosters transparency in user interactions. The 'balanceOf' function empowers users to easily check their token holdings, promoting a clearunderstanding of their financial status within the ecosystem. The 'transfer' function adds a layer of security by ensuring that transactions occur only when users have sufficient funds, mitigating the risk of unauthorized or incomplete transfers. The thoughtful addition of the 'payOut' function introduces a user-friendly mechanism, allowing users to claim specific amounts of NAAVY tokens, thereby enhancing accessibility and contributing to a more equitable distribution of tokens.

The 'NFT' actor, with its focus on secure NFT operations, brings a unique dimension to the NFTForge platform. The 'transferOwnership' function stands out as a crucial security feature, ensuring that only legitimate owners can initiate ownership transfers, thereby safeguarding digital assets. The ability for users to mint unique NFTs with personalized content not only enhances diversity within the digital asset landscape but also fosters a sense of individuality for users. Furthermore, functions like 'getName,' 'getOwner,' and 'getAsset' empower users with detailed insights into their NFTs, promoting a deeper and more meaningful interaction with their digital possessions.

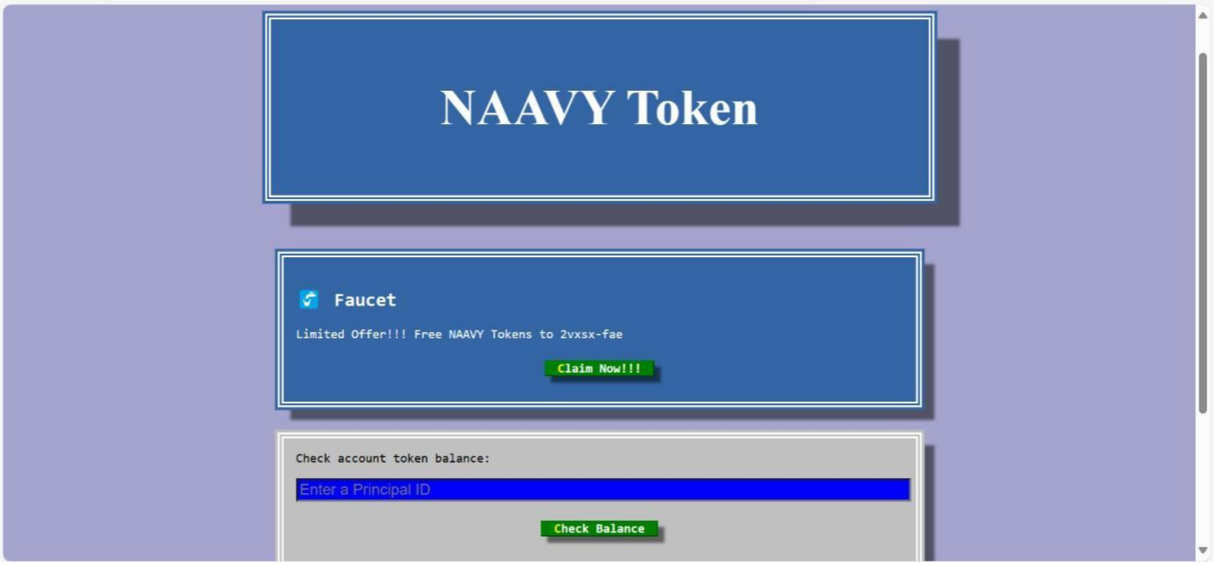
The 'OpenD' actor emerges as the central orchestrator, bringing dynamism to the NFTForge platform. Its role in facilitating NFT minting, listing, and purchase transactions underscores its significance. The 'OpenD' actor ensures a seamless experience for users by overseeing the creation and listing of NFTs through the 'mint' and 'listItem' functions. Moreover, the 'completePurchase' function guarantees secure ownership transfers post- purchase, adding a layer of reliability to the platform's transactional processes.

Looking ahead, the NFTForge project holds promising avenues for future exploration.

The consideration of cross-chain compatibility could enhance interoperability, enabling the platform to interact seamlessly with other blockchain networks. Integrating decentralized finance (DeFi) protocols presents an exciting prospect for expanding financial functionalities within the platform, potentially unlocking new possibilities for users. Additionally, a commitment to adherence to established NFT standards and protocols will ensure seamless interactions within the broader NFT community, fostering collaboration and integration.

### 4.1.2 Results

### Naavy Token:



NAAVY Token

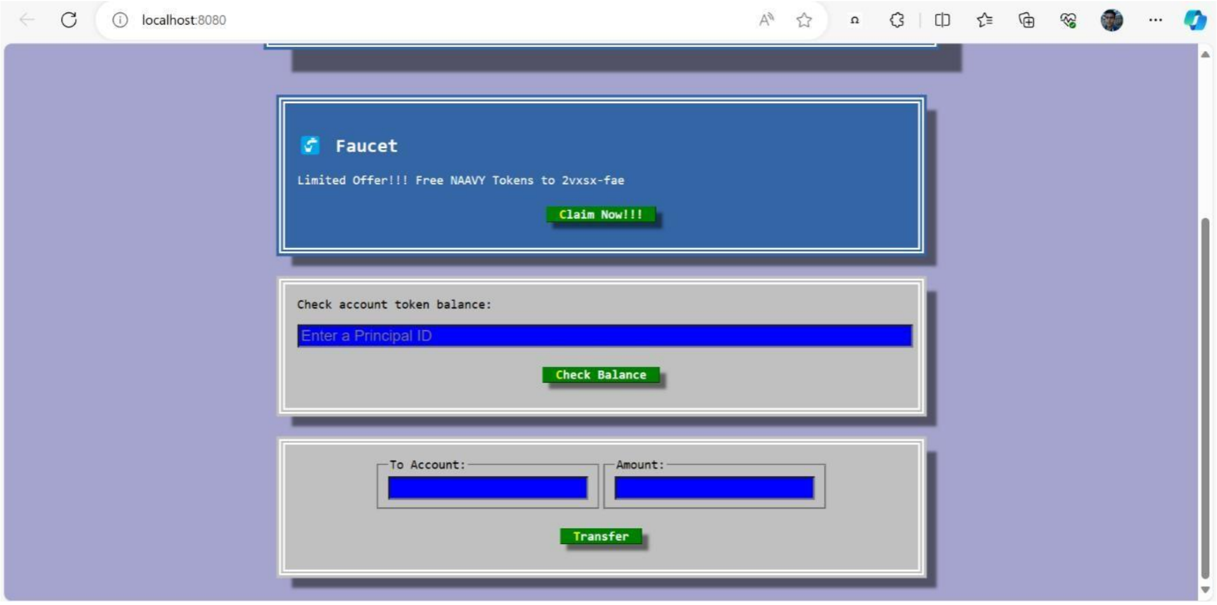
Faucet

Limited Offer!!! Free NAAVY Tokens to 2vxsx-fae

Check account token balance:

Enter a Principal ID

**Fig 4.1-** Home Page of Naavy Token



Faucet

Limited Offer!!! Free NAAVY Tokens to 2vxsx-fae

Check account token balance:

Enter a Principal ID

To Account:

Amount:

**Fig 4.2-** Create Naavy Token

## NFTForge:

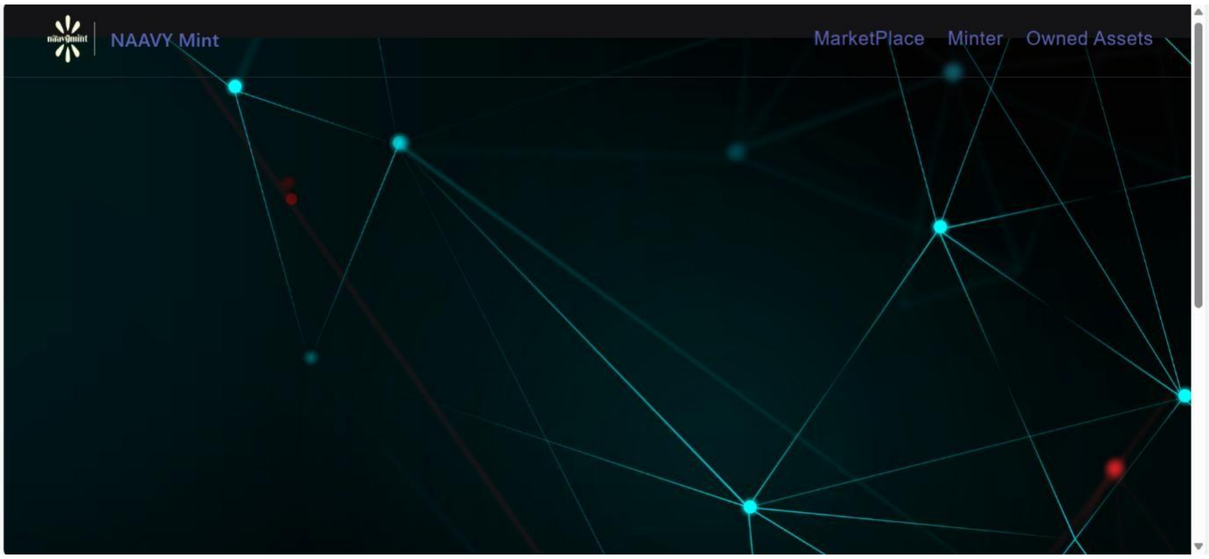
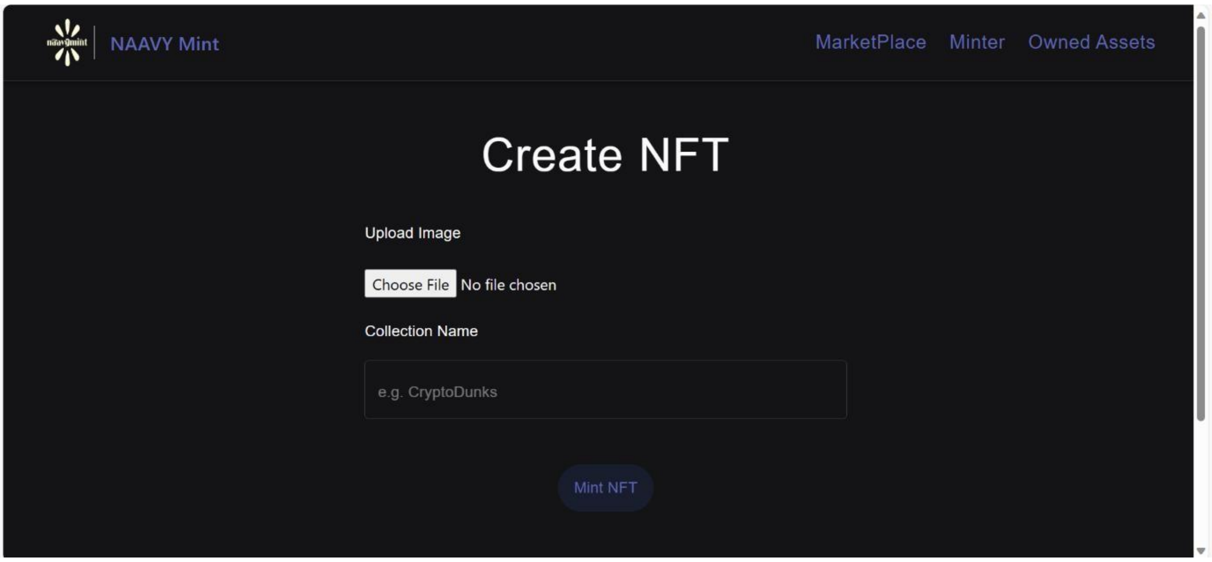


**Fig 4.3-** NFTForge Home Page



Create NFT

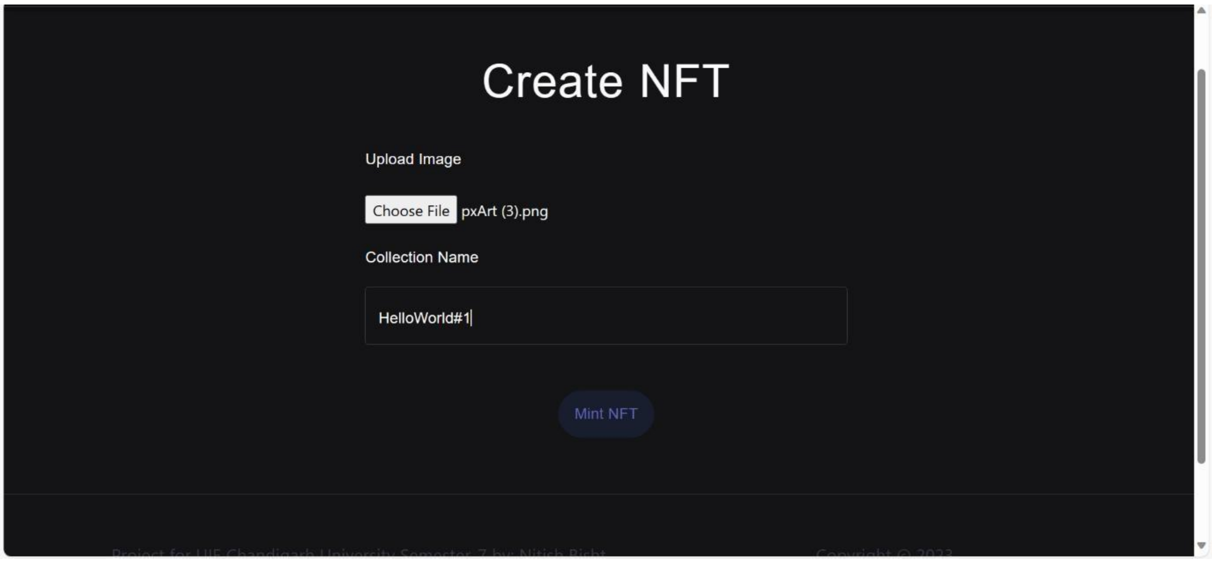
Upload Image

No file chosen

Collection Name

e.g. CryptoDunks

**Fig 4.4-** Create NFT



Create NFT

Upload Image

pxArt (3).png

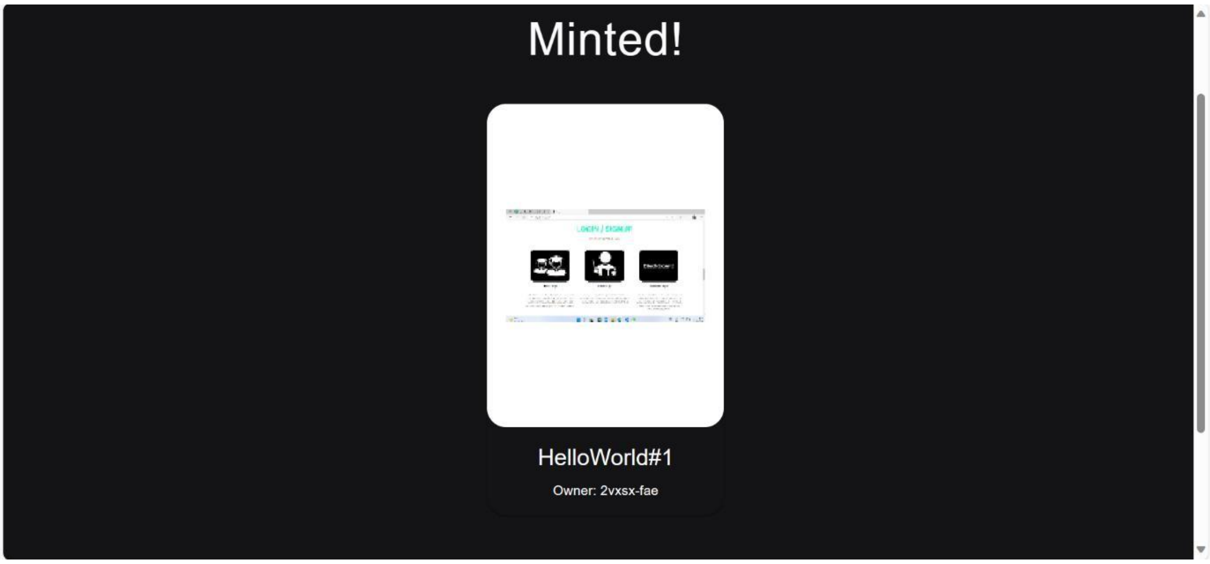
Collection Name

HelloWorld#1

Project for UIE Chandigarh University Semester 7 by: Nitish Bisht

Copyright © 2023

**Fig 4.5-** NFT created

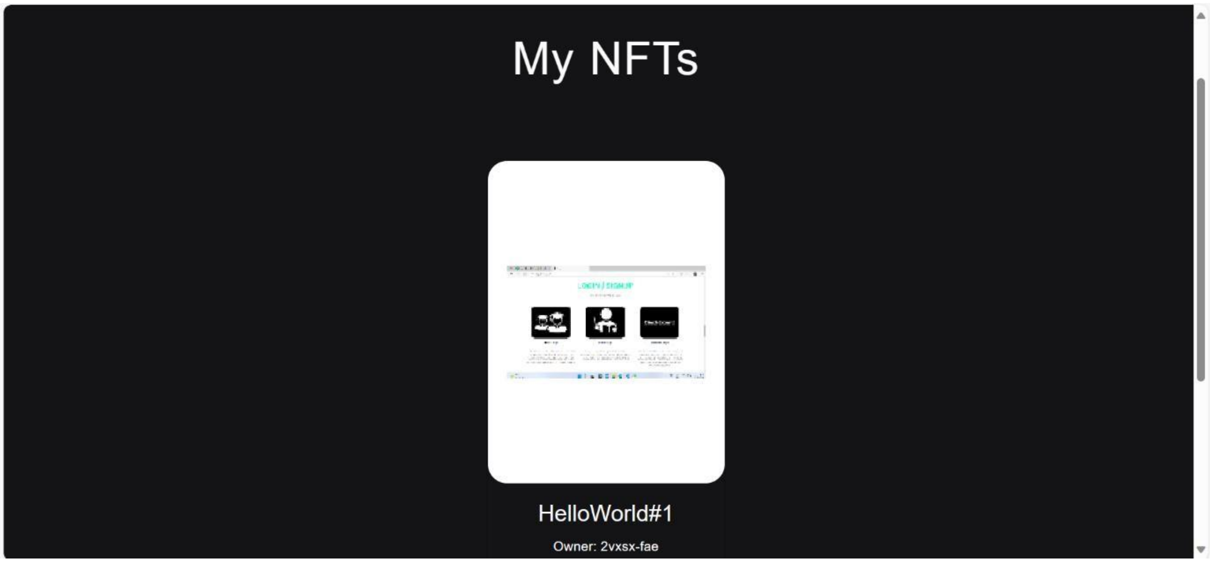


Minted!

HelloWorld#1

Owner: 2vxsx-fae

**Fig 4.6-** Minted NFTs Page

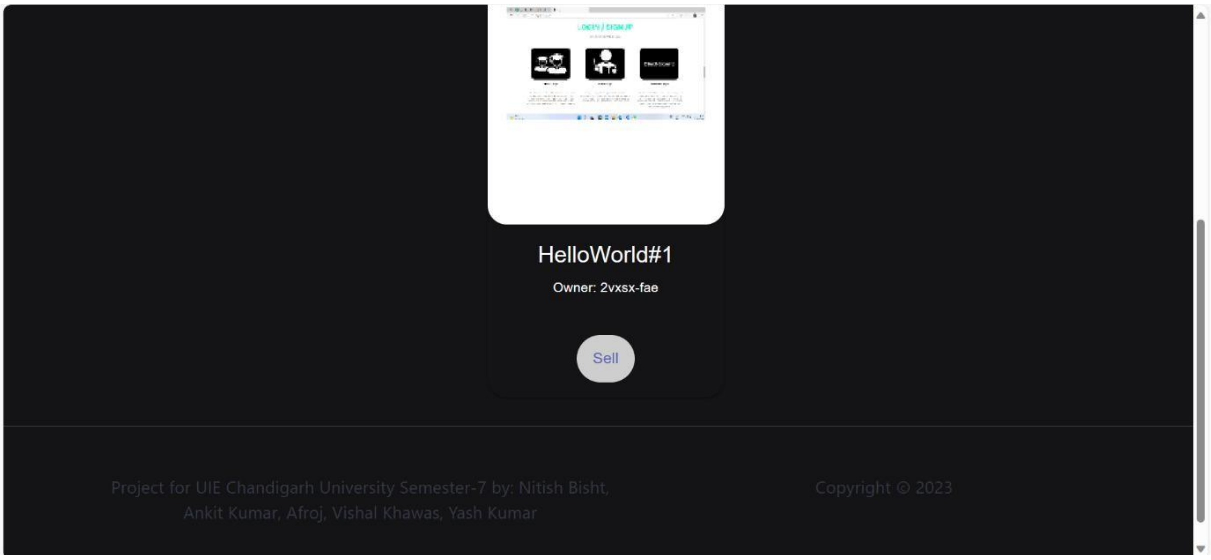


My NFTs

HelloWorld#1

Owner: 2vxsx-fae

**Fig 4.7-** My NFTs Page



Project for UIE Chandigarh University Semester-7 by: Nitish Bisht, Ankit Kumar, Afroj, Vishal Khawas, Yash Kumar

Copyright © 2023

**Fig 4.8-** Sell the NFTs

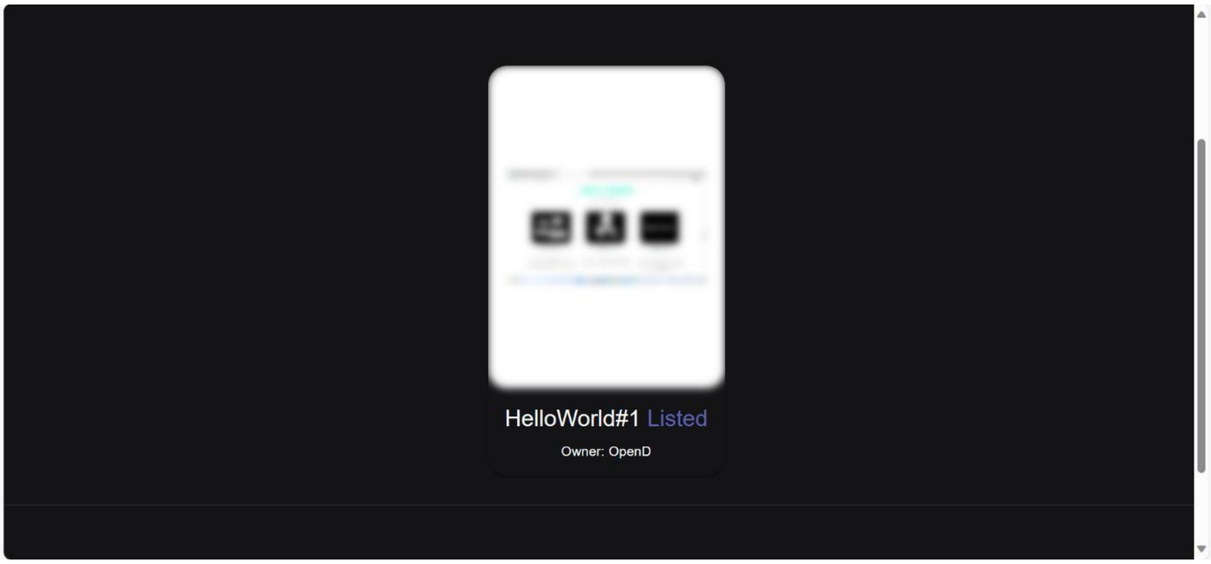
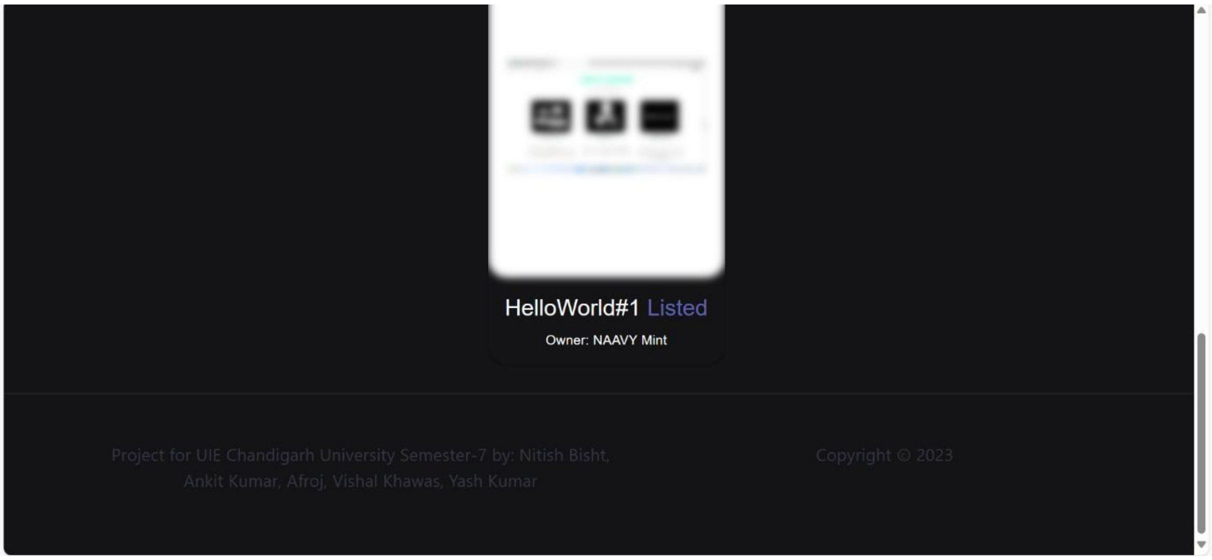


**Fig 4.9-** Sold and now Listed in Market


HelloWorld#1 Listed

Owner: NAAVY Mint

Project for UIE Chandigarh University Semester-7 by: Nitish Bisht, Ankit Kumar, Afroj, Vishal Khawas, Yash Kumar

Copyright © 2023

**Fig 4.10-** Listed in Marketplace

### 4.1.3 Testing

Testing is a critical phase in the development lifecycle of any project, ensuring the reliability, security, and functionality of the implemented code. The NFTForge project, designed for managing NAAVY tokens and non-fungible tokens (NFTs), requires thorough testing to validate its effectiveness and security. This examination encompasses the 'Token,' 'NFT,' and 'OpenD' actors, evaluating their functionalities, interactions, and potential vulnerabilities.

'Token' Actor: The 'Token' actor, being the backbone of NFTForge, necessitates meticulous testing to ensure accurate balance tracking, secure transfers, and proper token allocation. Unit testing of the 'balanceOf' function is essential to verify its ability to retrieve user balances accurately. Testing scenarios should include cases with various user balances, including edge cases and scenarios where users have not claimed any tokens.

The 'transfer' function, responsible for facilitating secure transfers, requires extensive testing to ensure it properly deducts funds from the sender and credits the recipient. Test cases should cover scenarios where the sender has sufficient and insufficient funds, as well as potential edge cases. Special attention should be given to scenarios involving concurrent transactions to prevent race conditions.

The 'payOut' function, allowing users to claim a specified amount of NAAVY tokens, should undergo testing to confirm its proper functionality. This includes scenarios where users claim tokens for the first time, ensuring that the claimed amount is accurate, and subsequent attempts by users who have already claimed tokens, where the function should appropriately respond with an "Already Claimed" message.

'NFT' Actor:The 'NFT' actor, focusing on the secure creation and transfer of NFTs, demands thorough testing to guarantee the authenticity and security of digital assets. The 'transferOwnership' function should undergo testing to verify that only the rightful owner can initiate ownership transfers. This involves creating test cases where unauthorized attempts are made to transfer ownership, ensuring the function responds with an appropriate error message.

Minting new NFTs using the 'mint' function requires testing to ensure the correct creation and storage of NFTs. This includes verifying that minted NFTs have unique identifiers, proper owner assignments, and accurate storage within the 'OpenD' actor's data structures. Additionally, testing should cover scenarios with large-scale NFT minting to assess the system's scalability.

The functions 'getName,' 'getOwner,' and 'getAsset' should be individually tested to ensure they accurately retrieve the name, owner, and asset content of NFTs, respectively. Test cases should cover different NFTs with varying attributes to ensure the versatility and reliability of these functions.

'OpenD' Actor:The 'OpenD' actor, responsible for overseeing critical marketplace operations, necessitates comprehensive testing to validate the seamless creation, listing, and purchase of NFTs. The 'mint' function should be tested to ensure the successful creation of NFTs, with each minted NFT having a unique identifier and accurate storage within the 'OpenD' actor.

The 'listItem' function, enabling users to list their NFTs for sale, requires testing to ensure the proper association of NFTs with their respective owners and accurate recording of listing details. Test cases should cover scenarios where users attempt to list NFTs they do not own, ensuring the function responds appropriately.

The 'completePurchase' function, facilitating the secure transfer of NFT ownership post-purchase, demands thorough testing to ensure the correct update of ownership records and the removal of listings. Test cases should cover scenarios where users attempt to purchase NFTs that are not listed or those listed by unauthorized owners.

Security Testing and Future Considerations:Beyond functional testing, the NFTForge project should undergo security testing, including penetration testing and code reviews, to identify and rectify potential vulnerabilities. Future testing efforts should also consider aspects such as cross-chain compatibility, smart contract audits, and adherence to emerging standards within the blockchain and NFT ecosystems.

In conclusion, testing is a critical aspect of ensuring the reliability and security of the NFTForge project. Rigorous testing of the 'Token,' 'NFT,' and 'OpenD' actors will contribute to the project's robustness, fostering a secure and user-friendly environment for managing NAAVY tokens and NFTs.

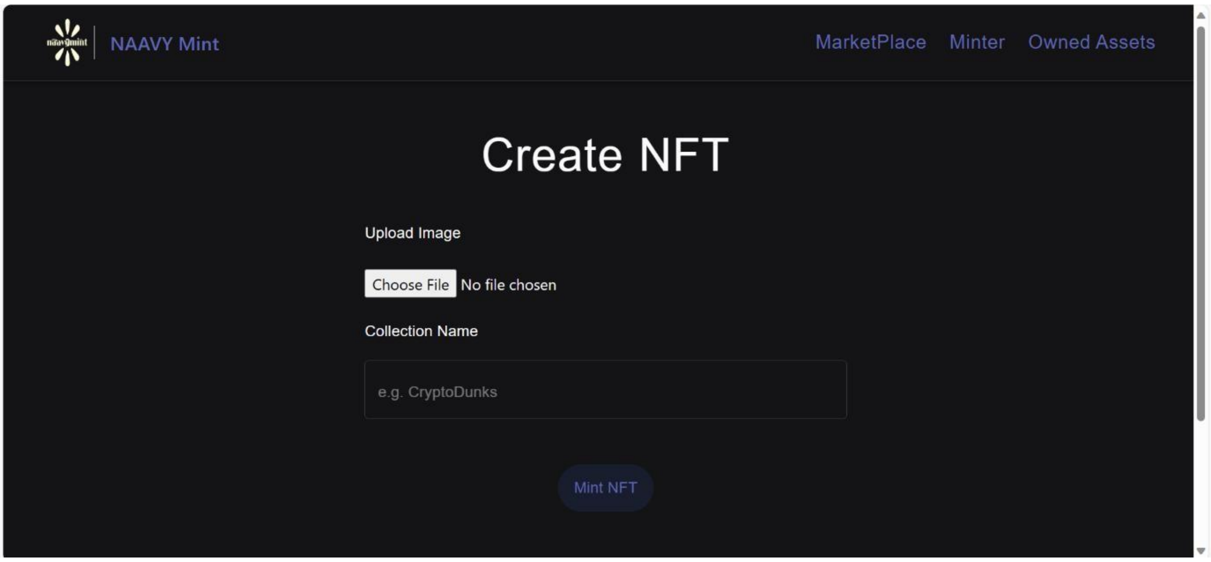


**Fig 4.11-** Create NFT

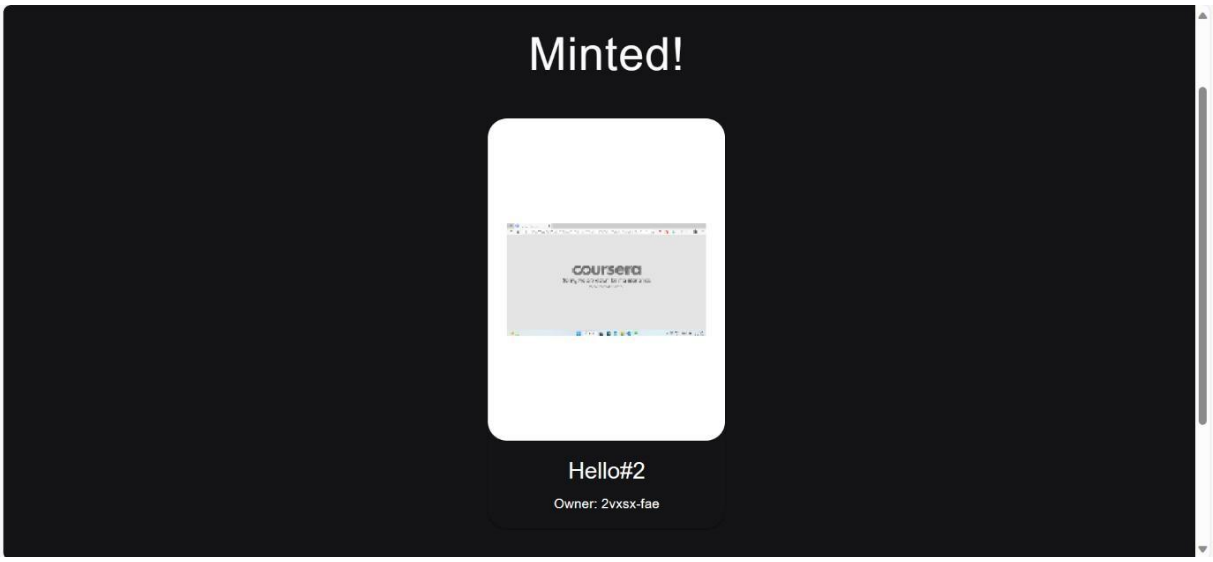


**Fig 4.12-** Minted Page

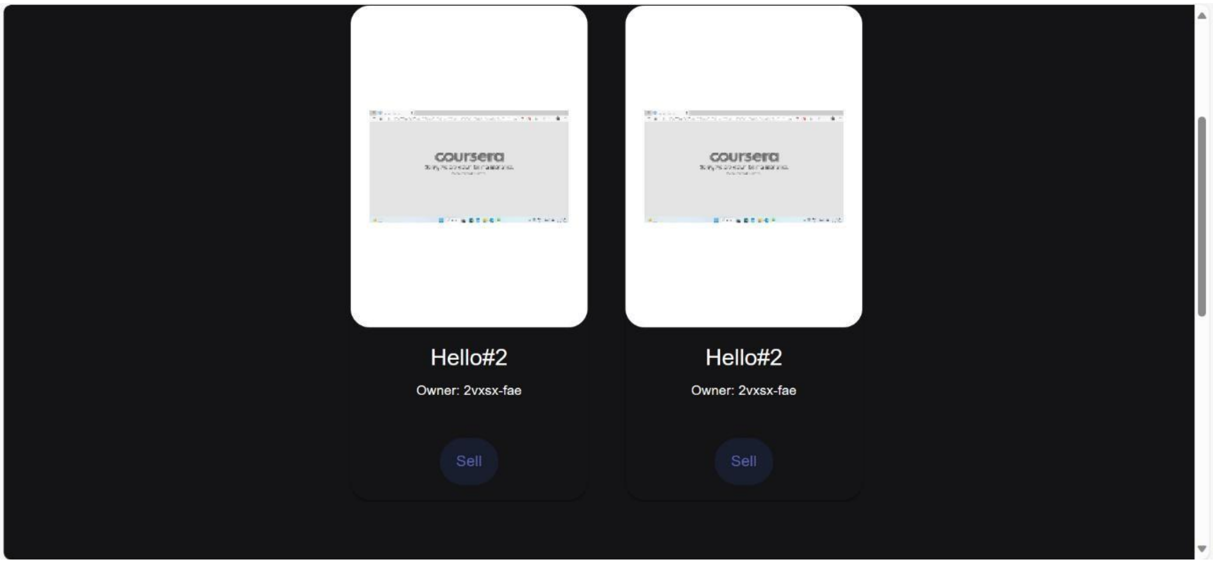
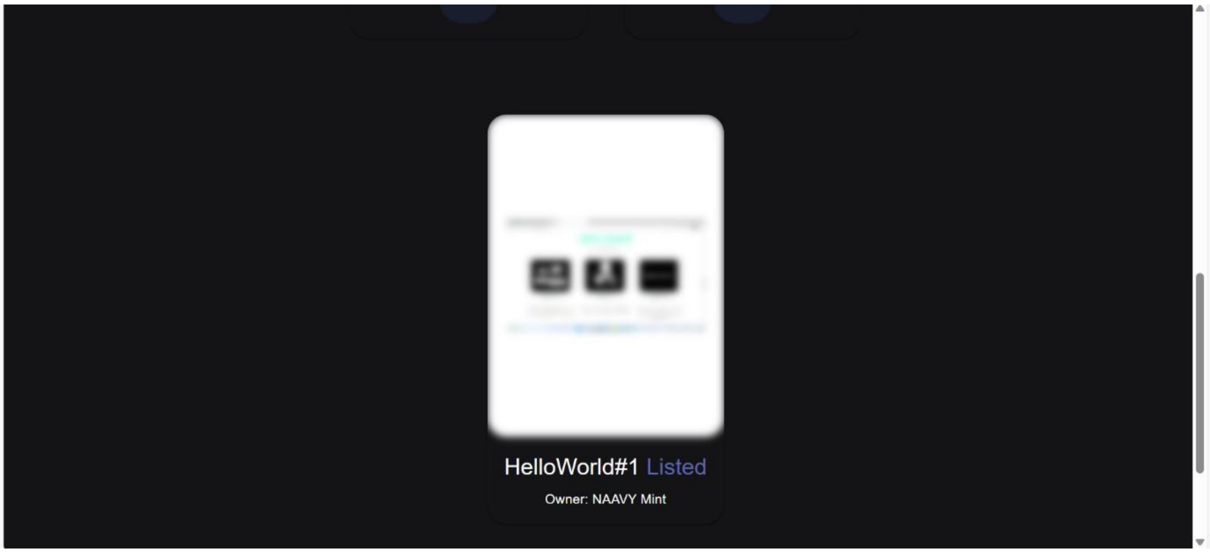


**Fig 4.13-** Minted NFTs selling page



HelloWorld#1 Listed

Owner: NAAVY Mint

**Fig 4.14-** Sold and Listed

# Chapter 5
# Conclusion and Future Work

## 5.1 Conclusion:

In conclusion, the provided code effectively demonstrates the successful implementation of the NFTForge project, showcasing its ability to handle essential functionalities, including the creation, management, and secure transfer of NAAVY tokens through the 'Token' actor. With its efficient balance tracking and secure transfer mechanisms, the 'Token' actor establishes a reliable transaction environment within the NFTForge ecosystem, instilling a sense of security and trust among users.

Similarly, the 'NFT' actor plays a significant role in maintaining the platform's integrity by enabling the secure creation and transfer of unique non-fungible tokens, thereby enhancing users' confidence in managing their digital assets.

Furthermore, the 'OpenD' actor serves a critical function in overseeing various operations within the NFTForge platform, including NFT minting, listing, and purchase transactions. By facilitating seamless NFT creation and listings while ensuring secure transactions, the 'OpenD' actor strengthens the marketplace's credibility, ensuring a smooth and reliable experience for all users.

With its well-structured functionalities, the 'OpenD' actor emerges as a dependable facilitator of transparent and secure transactions, establishing a trustworthy and secure digital marketplace for all participants to engage with.

However, ensuring the continuous success and reliability of the NFTForge project requires thorough testing and validation of the code. Rigorous testing procedures, including comprehensive security audits and robust error-handling protocols, are critical in identifying and addressing potential vulnerabilities or technical issues within the platform.

Regular monitoring and timely updates are essential in maintaining the platform's efficiency and security, ensuring a seamless and secure experience within the ever-evolving NFTForge marketplace. These efforts collectively contribute to establishing the NFTForge project as a trusted and user-friendly ecosystem, fostering a conducive environment for the secure management and exchange of digital assets.

## 5.2 Future Work:

**Enhancing Security Measures:** Moving forward, it is crucial to focus on enhancing the security measures embedded within the NFTForge project. This could involve implementing advanced encryption techniques to safeguard user data and transactions, ensuring a robust and secure environment for users to operate within. Additionally, the integration of multi-factor authentication protocols can further fortify the platform against potential cyber threats and unauthorized access.

**Implementing Advanced Smart Contract Features:** Integrating advanced smart contract features could significantly enhance the functionality and flexibility of the NFTForge platform. This could include the implementation of automated contract execution based on predefined conditions, enabling users to engage in more complex and sophisticated transactions with increased ease and efficiency. By incorporating such advanced features, the platform can cater to a wider range of user requirements and facilitate more intricate digital asset transactions.

**Exploring Decentralized Governance Models:** Exploring decentralized governance models can foster a more inclusive and participatory ecosystem within the NFTForge platform. By implementing decentralized decision-making processes, users can actively participate in shaping the future direction of the platform, contributing to its growth and development. Integrating voting mechanisms and community-driven initiatives can empower users to have a direct impact on the platform's policies and operations, fostering a sense of ownership and community engagement.

**Integration of Artificial Intelligence and Machine Learning:** The integration of artificial intelligence (AI) and machine learning (ML) technologies can revolutionize the NFTForge platform, enabling advanced data analysis and predictive modeling capabilities. By leveraging AI and ML algorithms, the platform can gain valuable insights into user behavior and preferences, enabling personalized recommendations and tailored services for users. This can significantly enhance the overall user experience, leading to improved user engagement and satisfaction within the NFTForge ecosystem.

**Expansion of Marketplace Offerings:** Expanding the marketplace offerings to include a diverse range of digital assets and services can attract a broader user base and foster a more vibrant and dynamic ecosystem. This could involve collaborating with various content creators, artists, and developers to introduce an extensive collection of NFTs, digital artworks, and other digital assets, catering to the diverse interests and preferences of users within the NFTForge marketplace. By diversifying the marketplace offerings, the platform can establish itself as a comprehensive and inclusive digital marketplace, catering to a wide spectrum of user needs and preferences.

**Emphasizing User Education and Support:** Emphasizing user education and support initiatives can play a pivotal role in enhancing user awareness and engagement within the NFTForge ecosystem. This could involve the development of comprehensive educational resources, tutorials, and user guides to help users navigate the platform effectively and make informed decisions regarding their digital asset transactions. Additionally, establishing a responsive and efficient user support system can provide users with timely assistance and guidance, fostering a positive and enriching user experience withinthe NFTForge platform.

**Integrating Sustainable and Eco-friendly Practices:** Integrating sustainable and eco-friendly practices within the NFTForge platform can contribute to a more environmentally conscious and responsible digital marketplace. This could involve collaborating with environmentally focused organizations and initiatives to implement sustainable practices, such as carbon footprint tracking and offsetting, within the platform's operations. By promoting eco-friendly initiatives and encouraging responsible digital asset transactions, the platform can contribute to global sustainability efforts and foster a more environmentally conscious user community within the NFTForge ecosystem.

**Integration of Cross-Chain Compatibility:** Exploring the integration of cross-chain compatibility can expand the reach and accessibility of the NFTForge platform. By enabling seamless interoperability with various blockchain networks and ecosystems, users can leverage a broader range of digital assets and services, fostering a more inclusive and interconnected digital marketplace. The integration of cross-chain compatibility can open up new avenues for digital asset transactions and collaborations, facilitating a more versatile and expansive user experience within the  NFTForge ecosystem.

**Adoption of Decentralized Finance (DeFi) Protocols:** Adopting decentralized finance (DeFi) protocols can revolutionize the financial infrastructure within the  NFTForge platform, enabling users to access a diverse array of financial services and products. By incorporating DeFi protocols, users can benefit from decentralized lending, borrowing, and staking mechanisms, facilitating efficient and secure financial transactions within the  NFTForge ecosystem. The integration of DeFi protocols can foster a more robust and comprehensive financial ecosystem, providing users with enhanced financial opportunities and services within the digital marketplace.

**Expansion of Community Engagement Programs:** Expanding community engagement programs can foster a more vibrant and interactive user community within the NFTForge ecosystem. This could involve organizing community events, workshops, and forums to encourage active participation and collaboration among users and stakeholders. By fostering a strong sense of community and camaraderie, the platform can cultivate a supportive and engaging environment for users to share ideas, provide feedback, and contribute to the platform's growth and development. Building a strong and active user community can establish a solid foundation for the platform's long-term sustainability and success within the digital marketplace.

**Integration of Non-Fungible Token (NFT) Standards and Protocols:** Integrating non-fungible token (NFT) standards and protocols can streamline and standardize the creation and management of NFTs within the  NFTForge platform. By adhering to established NFT standards and protocols, the platform can ensure compatibility and consistency across various digital asset transactions and interactions, facilitating seamless and efficient NFT creation and transfer processes. The integration of NFT standards and protocols can establish a common framework for users and developers to create, manage, and exchange NFTs, fostering a more cohesive and user-friendly digital marketplace experience within the  NFTForge ecosystem.

# REFERENCES

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] Christian Decker, Roger Wattenhofert, ETH Zurich, “Information Propagation in the Bitcoin Network.” Available: https://ieeexplore.ieee.org/document/6688704

[3] Jega Anish Dev, “Bitcoin mining acceleration and performance quantification.” Available: https://www.researchgate.net/publication/286733552

[4] Hiroki Watanabe, Shigeru Fujimura, Atsushi Nakadaira, Yasuhiko Miyazaki, Akihito Akutsu, “Blockchain Contract: A Complete Consensus using Blockchain.” Available: https://www.researchgate.net/publication/304298870

[5] Hiroki Watanabe, Shigeru Fujimura, Atsushi Nakadaira, Yasuhiko Miyazaki, Akihito Akutsu, Jay Kishigami, “Blockchain Contract: Securing a Blockchain Applied to Smart Contracts.” Available: https://www.researchgate.net/publication/304298870

[6] Victoria Lemieux, “Blockchain and Distributed Ledgers as Trusted Recordkeeping Systems: An Archival Theoretic Evaluation Framework.” Available: https://www.researchgate.net/publication/317433591

[7] William Nikolakis, Lijo John, Harish Krishnan, “How Blockchain Can Shape Sustainable Global Value Chains: An Evidence, Verifiability, and Enforceability (EVE) Framework.” Available: https://www.researchgate.net/publication/304298870

[8] wei cai, zehua wang, jason b. ernst, zhen hong, chen feng, victor c. m. leung, “Decentralized Applications: The Blockchain-Empowered Software System.” Available: https://www.researchgate.net/publication/327711685

[9] Fangyu Gai, Jianyu Niu, Cesar Grajales, “Cumulus: A BFT-based Sidechain Protocol for Off-chain Scaling.” Available: https://www.researchgate.net/publication/327711685

[10] Chen Feng, Jianyu Niu, “Selfish Mining in Ethereum.” Available: https://www.researchgate.net/publication/336946541

[11] Yiming Liu, F. Richard Yu, Xi Li, Hong Ji, Victor C. M. Leung, “Blockchain and Machine Learning for Communications and Networking Systems.” Available: https://www.researchgate.net/publication//339481056

[12] Deepak Puthal, Saraju P. Mohanty, Venkata P. Yanambaka, Elias Kougiano, “A Novel Consensus Algorithm for Fast Scalable Private Blockchain for Large-scale IoT Frameworks.”

[13] Phil Gonserkewitz, Erik Karger, Marvin Jagals, “Non-Fungible Tokens: Use Cases of NFTs And Future Research Agenda.”

[14] Yashika Nagpal, “Non-Fungible Tokens (NFT’s): The Future of Digital Collectibles.”

[15] Shreyansh Goushal, “A Deep Dive into Non-Fungible Tokens (NFTs) And Its Correlation With The Price Of Bitcoin And Ethereum.”

[16] Saeed Banaeian Fara, Seyed Mojtaba Hosseini Bamakan, Qiang Qu, Qingshan Jiang, "A Review of Non-fungible Tokens Applications in the Real-world and Metaverse."

[17] Timo Hanke, Mahnush Movahedi, Dominic Williams, "DFINITY Technology Overview Series Consensus System."

[18] Rahul Dattaram Belose, Yogesh Ramesh Mhadgut, "Study on Non-Fungible Tokens (NFT)."

[19] Kebira Azbeg, Ouail Ouchetto, Said Jai Andaloussi, Laila Fetjah, "An Overview of Blockchain Consensus Algorithms: Comparison, Challenges and Future Directions."

[20] Subhita Menon, Divya Anand, Kavita, "A Survey On Blockchain Smart Network based On Consensus Algorithm."

[21] Ruihang Huang, Xiaoming Yang, P. Ajay, "Consensus mechanism for software-defined blockchain in Internet of things."

[22] Zhipeng Fan, "Research on Blockchain Hybrid Consensus Algorithm Based on Internet of Things."

[23] Huanliang Xiong, Muxi Chen, Canghai Wu, Yingding Zhao, Wenlong Yi, "Research on Progress of Blockchain Consensus Algorithm: A Review on Recent Progress of Blockchain Consensus Algorithms."

[24] Victoria Lemieux, Nigel Dodd, "LIFEWORLD' ON LEDGER: A 'SCENIC' VIEW." Available: https://www.researchgate.net/publication/372025902.

# APPENDIX

## NAAVY TOKEN FRONT-END:

### App.jsx:

```
import React from "react";
import Header from "./Header";
import Faucet from "./Faucet";
import Balance from "./Balance";
import Transfer from "./Transfer";

function App(props) {

  return (
    <div id="screen">
      <Header />
      <Faucet />
      <Balance />
      <Transfer />
    </div>
  );
}

export default App;
```

### Balance.jsx:

```
import React, { useState } from "react";
import { Principal } from '@dfinity/principal';
import { token } from "../../../declarations/token";

function Balance() {

  const [inputValue, setInput] = useState("");
  const [balanceResult, setBalance] = useState("");
  const [cryptoSymbol, setSymbol] = useState("");
  const [isHidden, setHidden] = useState(true);

  async function handleClick() {
    // console.log(inputValue);
    const principal = Principal.fromText(inputValue);
    const balance = await token.balanceOf(principal);
    setBalance(balance.toLocaleString());
    setSymbol(await token.getSymbol());
    setHidden(false);
  }


  return (
    <div className="window white">
```

```
   <label>Check account token balance:</label>
   <p>
    <input
     id="balance-principal-id"
     type="text"
     placeholder="Enter a Principal ID"
     value={inputValue}
     onChange={(e) => setInput(e.target.value)}
    />
   </p>
   <p className="trade-buttons">
    <button
     id="btn-request-balance"
     onClick={handleClick}
    >
     Check Balance
    </button>
   </p>
   <p hidden={isHidden}>This account has a balance of {balanceResult}
{cryptoSymbol}.</p>
  </div>
 );
}

export default Balance;
```

## Faucet.jsx:

```
import React, { useState } from "react";
import { token } from "../../../declarations/token";

function Faucet(props) {
 const [isDisabled, setDisable] = useState(false);
 const [buttonText, setText] = useState("Claim Now!!!");

 async function handleClick(event)
  {setDisable(true);
  const result = await token.payOut();
  console.log("payout: " + result);
  setText(result);
 }

 return (
  <div className="blue window">
   <h2>
    <span role="img" aria-label="tap emoji">

    </span>
    Faucet
   </h2>
   <label>
```

```
        Limited Offer!!! Free NAAVY Tokens to 2vxsx-fae
      </label>
      <p className="trade-buttons">
        <button id="btn-payout" onClick={handleClick} disabled={isDisabled}>
          {buttonText}
        </button>
      </p>
    </div>
  );
}

export default Faucet;
```

## Header.jsx:

```
import React from "react";

function Header()
  {return (
    <header>
      <div className="blue window" id="logo">
        <h1>
          NAAVY Token
        </h1>
      </div>
    </header>
  );
}

export default Header;
```

**Transfer.jsx:**

```
import React, { useState } from "react";
import { Principal } from "@dfinity/principal";
import { token } from "../../../declarations/token";

function Transfer() {
  const [recipientId, setId] = useState("");
  const [amount, setAmount] = useState("");
  const [isHidden, setHidden] = useState(true);
  const [feedback, setFeedback] = useState("");
  const [isDisabled, setDisable] = useState(false);

  async function handleClick()
    {setHidden(true);
    setDisable(true);
    const recipient = Principal.fromText(recipientId);
    const amountToTransfer = Number(amount);

    const result = await token.transfer(recipient, amountToTransfer);
    setFeedback(result);
    setHidden(false);
    setDisable(false);
  }

  return (
    <div className="window white">
      <div className="transfer">
        <fieldset>
          <legend>To Account:</legend>
          <ul>
            <li>
              <input
                type="text"
                id="transfer-to-id"
                value={recipientId}
                onChange={(e) => setId(e.target.value)}
              />
            </li>
          </ul>
        </fieldset>
        <fieldset>
          <legend>Amount:</legend>
          <ul>
            <li>
              <input
                type="number"
                id="amount"
                value={amount}
                onChange={(e) => setAmount(e.target.value)}
              />
            </li>
          </ul>
        </fieldset>
        <p className="trade-buttons">
          <button id="btn-transfer" onClick={handleClick} disabled={isDisabled}>
            Transfer
          </button>
        </p>
        <p hidden={isHidden}>{feedback}</p>
      </div>
    </div>
  );
}

export default Transfer;
```

## NAAVY TOKEN BACK-END:

main.mo:

```
import Principal "mo:base/Principal";
import HashMap "mo:base/HashMap";
import Debug "mo:base/Debug";
import Iter "mo:base/Iter";

actor Token {

  let owner : Principal = Principal.fromText("iqjlv-qbnxm-xolci-ocjap-x6f76-xu5sd-jqwl5-
b5emu-ppitc-khrcl-tae");
  let totalSupply : Nat = 1000000000;
  let symbol : Text = "NAAVY";

  private stable var balanceEntries : [(Principal, Nat)] = [];
  private var balances = HashMap.HashMap<Principal, Nat>(1, Principal.equal,
Principal.hash);
  if (balances.size() < 1)
    { balances.put(owner,
    totalSupply);
  };

  public query func balanceOf(who : Principal) : async Nat

    {let balance : Nat = switch (balances.get(who)) {
      case null 0;
      case (?result) result;
    };

    return balance;
  };

  public query func getSymbol() : async Text
    {return symbol;
  };

  public shared (msg) func payOut() : async Text
    {Debug.print(debug_show (msg.caller));
    if (balances.get(msg.caller) == null)
      {let amount = 10000;
      let result = await transfer(msg.caller, amount);
      return result;
    } else {
      return "Already Claimed";
    };
  };

  public shared (msg) func transfer(to : Principal, amount : Nat) : async Text
    {let fromBalance = await balanceOf(msg.caller);
    if (fromBalance > amount) {
      let newFromBalance : Nat = fromBalance - amount;
      balances.put(msg.caller, newFromBalance);

      let toBalance = await balanceOf(to);
      let newToBalance = toBalance + amount;
      balances.put(to, newToBalance);

      return "Success";
    } else {
      return "Insufficient Funds";
    };
  };

  system func preupgrade() {
    balanceEntries := Iter.toArray(balances.entries());
  };

  system func postupgrade() {
    balances := HashMap.fromIter<Principal, Nat>(balanceEntries.vals(), 1, Principal.equal,
Principal.hash);
    if (balances.size() < 1)
      { balances.put(owner,
      totalSupply);
    };
  };

};
```

## NFTForge FRONT-END:

### App.jsx:

```
import React from "react";
import Header from "./Header";
import Footer from "./Footer";
import "bootstrap/dist/css/bootstrap.min.css";
import Item from "./Item";
import Minter from "./Minter";

function App() {
  // const NFTID = "rrkah-fqaaa-aaaaa-aaaaq-cai";

  return (
    <div className="App">
      <Header />
      {/* <Minter /> */}
      {/* <Item id={NFTID}/> */}

      <Footer />
    </div>
  );
}
```

```
export default App;
```

## Button.jsx:

```
import React from "react";

function Button(props)
   {return (
       <div className="Chip-root makeStyles-chipBlue-108 Chip-clickable">
          <span onClick={props.handleClick} className="form-Chip-label">
             {props.text}
          </span>
       </div>
   );
}

export default Button;
```

## Footer.jsx:

```
import React from "react";
import Container from "react-bootstrap/Container";
import Row from "react-bootstrap/Row";
import Col from "react-bootstrap/Col";

function Footer() {
  const year = new Date().getFullYear();

  return (
    <div id="footer">
      <footer>
        <Container fluid="md">
          <Row>
            <Col>
              <p>
                Project for Chandigarh University Semester-7 by:
                Nitish Bisht, Ankit Kumar, Afroj, Vishal Khawas, Yash Kumar
              </p>
            </Col>
            <Col>
              <p>Copyright ⓒ {year}</p>
            </Col>
          </Row>
        </Container>
      </footer>
    </div>
  );
}

export default Footer;
```

## Gallery.jsx:

```
import React, { useEffect, useState } from "react";
import Item from "./Item";
import { Principal } from "@dfinity/principal";

function Gallery(props) {
  const [items, setItems] = useState();

  function fetchNFTs() {
    if (props.ids != undefined)
      { setItems( props.ids.map((NF
      TId) => (
          <Item id={NFTId} key={NFTId.toText()} role={props.role} />
        ))
      );
    }
  }

  useEffect(() =>
    {fetchNFTs();
  }, []);

  return (
    <div className="gallery-view">
      <h3 className="makeStyles-title-99 Typography-h3">{props.title}</h3>
      <div className="disGrid-root disGrid-container disGrid-spacing-xs-2">
        <div className="disGrid-root disGrid-item disGrid-grid-xs-12">
          <div className="disGrid-root disGrid-container disGrid-spacing-xs-5 disGrid-justify-
content-xs-center">
            {items}
          </div>
        </div>
      </div>
    </div>
  );
}

export default Gallery;
```

## Header.jsx:

```
import React, { useEffect, useState } from "react";
import logo from "../../assets/2.png";
import homeImage from "../../assets/1.jpg";
import { BrowserRouter, Link, Switch, Route } from "react-router-dom";
import Minter from "./Minter";
import Gallery from "./Gallery";
import { opend } from "../../../declarations/opend";
import CURRENT_USER_ID from "../index";
```

```
function Header() {
  const [userOwnedGallery, setOwnedGallery] = useState();
  const [listingGallery, setListingGallery] = useState();

  async function getNFTs() {
    const userNFTIds = await opend.getOwnedNFTs(CURRENT_USER_ID);
    console.log(userNFTIds);
    setOwnedGallery(
      <Gallery title="My NFTs" ids={userNFTIds} role="collection" />
    );

    const listedNFTIds = await opend.getListedNFTs();
    console.log(listedNFTIds);
    setListingGallery(
      <Gallery title="Discover" ids={listedNFTIds} role="discover" />
    );
  }

  useEffect(() =>
    {getNFTs();
  }, []);

  return (
    <BrowserRouter forceRefresh={true}>
      <div className="app-root-1">
        <header className="Paper-root AppBar-root AppBar-positionStatic AppBar-
colorPrimary Paper-elevation4">
          <div className="Toolbar-root Toolbar-regular header-appBar-13 Toolbar-gutters">
            <div className="header-left-4"></div>
            <img className="header-logo-11" src={logo} />
            <div className="header-vertical-9"></div>
            <Link to="/">
              <h5 className="Typography-root header-logo-text"> NFTForge</h5>
            </Link>
            <div className="header-empty-6"></div>
            <div className="header-space-8"></div>
            <button className="ButtonBase-root Button-root Button-text header-navButtons-3">
              <Link to="/discover">MarketPlace</Link>
            </button>
            <button className="ButtonBase-root Button-root Button-text header-navButtons-3">
              <Link to="/minter">Minter</Link>
            </button>
            <button className="ButtonBase-root Button-root Button-text header-navButtons-3">
              <Link to="/collection">Owned Assets</Link>
            </button>
          </div>
        </header>
      </div>
      <Switch>
        <Route exact path="/">
```

```
          <img className="bottom-space" src={homeImage} style={{ width: "100%" }} />
        </Route>
        <Route path="/discover">{listingGallery}</Route>
        <Route path="/minter">
          <Minter />
        </Route>
        <Route path="/collection">{userOwnedGallery}</Route>
      </Switch>
    </BrowserRouter>
  );
}

export default Header;
```

## Item.jsx:

```
import React, { useEffect, useState } from "react";
import logo from "../../assets/logo.png";
import { Actor, HttpAgent } from "@dfinity/agent";
import { idlFactory } from "../../../declarations/nft";
import { idlFactory as tokenIdlFactory } from "../../../declarations/token";
import { Principal } from "@dfinity/principal";
import { opend } from "../../../declarations/opend";
import Button from "./Button";
import CURRENT_USER_ID from "../index";
import PriceLabel from "./PriceLabel";

function Item(props) {
  const [name, setName] = useState();
  const [owner, setOwner] = useState();
  const [image, setImage] = useState();
  const [button, setButton] = useState();
  const [priceInput, setPriceInput] = useState();
  const [loaderHidden, setLoaderHidden] = useState(true);
  const [blur, setBlur] = useState();
  const [sellStatus, setSellStatus] = useState("");
  const [priceLabel, setPriceLabel] = useState();
  const [shouldDisplay, setDisplay] = useState(true);

  const id = props.id;

  const localHost = "http://localhost:8080/";
  const agent = new HttpAgent({ host: localHost });

  //TODO: When deploy live, remove the following line.
  agent.fetchRootKey();
  let NFTActor;

  async function loadNFT() {
    NFTActor = await Actor.createActor(idlFactory,
      {agent,
      canisterId: id,
    });

    const name = await NFTActor.getName();
    const owner = await NFTActor.getOwner();
    const imageData = await NFTActor.getAsset();
    const imageContent = new Uint8Array(imageData);
    const image = URL.createObjectURL(
      new Blob([imageContent.buffer], { type: "image/png" })
    );

    setName(name);
    setOwner(owner.toText());
    setImage(image);

    if (props.role == "collection") {
      const nftIsListed = await opend.isListed(props.id);

      if (nftIsListed)
        { setOwner("OpenD");
        setBlur({ filter: "blur(4px)" });
        setSellStatus("Listed");
      } else {
        setButton(<Button handleClick={handleSell} text={"Sell"} />);
      }
    } else if (props.role == "discover") {
      const originalOwner = await opend.getOriginalOwner(props.id);
      if (originalOwner.toText() != CURRENT_USER_ID.toText())
      { setButton(<Button handleClick={handleBuy} text={"Buy"} />);
      }

      const price = await opend.getListedNFTPrice(props.id);
      setPriceLabel(<PriceLabel sellPrice={price.toString()} />);
    }
  }

  useEffect(() =>
    {loadNFT();
  }, []);

  let price;
  function handleSell()
    { console.log("Sell
    clicked");setPriceInput(
      <input
        placeholder="Price in DANG"
        type="number"
        className="price-input"
        value={price}
        onChange={(e) => (price = e.target.value)}
      />
    );
    setButton(<Button handleClick={sellItem} text={"Confirm"} />);
  }

  async function sellItem()
    { setBlur({ filter: "blur(4px)" });
    setLoaderHidden(false);
    console.log("set price = " + price);
    const listingResult = await opend.listItem(props.id, Number(price));
    console.log("listing: " + listingResult);
    if (listingResult == "Success") {
      const openDId = await opend.getOpenDCanisterID();
      const transferResult = await NFTActor.transferOwnership(openDId);
      console.log("transfer: " + transferResult);
      if (transferResult == "Success")
        {setLoaderHidden(true);
        setButton();
        setPriceInput();
        setOwner("OpenD");
        setSellStatus("Listed");
      }
    }
  }

  async function handleBuy() {
    // console.log("Buy was triggered");
    setLoaderHidden(false);
    const tokenActor = await Actor.createActor(tokenIdlFactory,
      {agent,
      canisterId: Principal.fromText("renrk-eyaaa-aaaaa-aaada-cai")
    });

    const sellerId = await opend.getOriginalOwner(props.id);
    const itemPrice = await opend.getListedNFTPrice(props.id);

    const result = await tokenActor.transfer(sellerId, itemPrice);
    if (result == "Success") {
      //Transfer ownership
      const transferResult = await opend.completePurchase(props.id, sellerId,
CURRENT_USER_ID);
      console.log(transferResult);
      setLoaderHidden(true);
      setDisplay(false);
    }
  }

  return (
    <div style={{ display: shouldDisplay ? "inline" : "none" }} className="disGrid-item">
```

```
      <div className="disPaper-root disCard-root makeStyles-root-17 disPaper-elevation1
disPaper-rounded">
        <img
          className="disCardMedia-root makeStyles-image-19 disCardMedia-media
disCardMedia-img"
          src={image}
          style={blur}
        />
        <div className="lds-ellipsis" hidden={loaderHidden}>
          <div></div>
          <div></div>
          <div></div>
          <div></div>
        </div>
        <div className="disCardContent-root">
          {priceLabel}
          <h2 className="disTypography-root makeStyles-bodyText-24 disTypography-h5
disTypography-gutterBottom">
            {name}
            <span className="purple-text"> {sellStatus}</span>
          </h2>
          <p className="disTypography-root makeStyles-bodyText-24 disTypography-body2
disTypography-colorTextSecondary">
            Owner: {owner}
          </p>
          {priceInput}
          {button}
        </div>
      </div>
    </div>
  );
}

export default Item;
```

## Minter.jsx:

```
import React, { useState } from "react";
import { useForm } from "react-hook-form";
import { opend } from "../../../declarations/opend";
import { Principal } from "@dfinity/principal";
import Item from "./Item";

function Minter() {
  const { register, handleSubmit } = useForm();
  const [nftPrincipal, setNFTPrincipal] = useState("");
  const [loaderHidden, setLoaderHidden] = useState(true);

  async function onSubmit(data)
    {setLoaderHidden(false);
    const name = data.name;
    const image = data.image[0];
    const imageArray = await image.arrayBuffer();
    const imageByteData = [...new Uint8Array(imageArray)];

    const newNFTID = await opend.mint(imageByteData, name);
    console.log(newNFTID.toText());
    setNFTPrincipal(newNFTID);
    setLoaderHidden(true);
  }

  if (nftPrincipal == "")
    {return (
      <div className="minter-container">
        <div hidden={loaderHidden} className="lds-ellipsis">
          <div></div>
          <div></div>
          <div></div>
          <div></div>
        </div>
        <h3 className="makeStyles-title-99 Typography-h3 form-Typography-gutterBottom">
          Create NFT
        </h3>
        <h6 className="form-Typography-root makeStyles-subhead-102 form-Typography-subtitle1 form-Typography-gutterBottom">
          Upload Image
        </h6>
        <form className="makeStyles-form-109" noValidate="" autoComplete="off">
          <div className="upload-container">
            <input
              {...register("image", { required: true })}
              className="upload"
              type="file"
              accept="image/x-png,image/jpeg,image/gif,image/svg+xml,image/webp"
            />
          </div>
          <h6 className="form-Typography-root makeStyles-subhead-102 form-Typography-subtitle1 form-Typography-gutterBottom">
            Collection Name
        </h6>
          <div className="form-FormControl-root form-TextField-root form-FormControl-marginNormal form-FormControl-fullWidth">
            <div className="form-InputBase-root form-OutlinedInput-root form-InputBase-fullWidth form-InputBase-formControl">
              <input
                {...register("name", { required: true })}
                placeholder="e.g. CryptoDunks"
                type="text"
                className="form-InputBase-input form-OutlinedInput-input"
              />
              <fieldset className="PrivateNotchedOutline-root-60 form-OutlinedInput-
  notchedOutline"></fieldset>
            </div>
          </div>
          <div className="form-ButtonBase-root form-Chip-root makeStyles-chipBlue-108
  form-Chip-clickable">
            <span onClick={handleSubmit(onSubmit)} className="form-Chip-label">
              Mint NFT
            </span>
          </div>
        </form>
      </div>
    );
  } else
    { retur
    n (
      <div className="minter-container">
        <h3 className="Typography-root makeStyles-title-99 Typography-h3 form-
  Typography-gutterBottom">
          Minted!
        </h3>
        <div className="horizontal-center">
          <Item id={nftPrincipal.toText()} />
        </div>
      </div>
    );
  }
  }

  export default Minter;
```

**<u>PriceLabel.jsx:</u>**

```
import React from "react";

function PriceLabel(props)
   {return (
      <div className="disButtonBase-root disChip-root makeStyles-price-23 disChip-
outlined">
          <span className="disChip-label">{props.sellPrice} DANG</span>
      </div>
   );
}
export default PriceLabel;
```

## NFTForge BACK-END:

**<u>main.mo:</u>**

```
import Cycles "mo:base/ExperimentalCycles";
import Debug "mo:base/Debug";
import NFTActorClass "../NFT/nft";
import Principal "mo:base/Principal";
import HashMap "mo:base/HashMap";
import List "mo:base/List";
import Iter "mo:base/Iter";

actor OpenD {

   private type Listing =
      { itemOwner :
      Principal;itemPrice :
      Nat;
   };

   var mapOfNFTs = HashMap.HashMap<Principal, NFTActorClass.NFT>(1, Principal.equal,
Principal.hash);
   var mapOfOwners = HashMap.HashMap<Principal, List.List<Principal>>(1,
Principal.equal, Principal.hash);
   var mapOfListings = HashMap.HashMap<Principal, Listing>(1, Principal.equal,
Principal.hash);

   public shared (msg) func mint(imgData : [Nat8], name : Text) : async Principal
      {let owner : Principal = msg.caller;

      Debug.print(debug_show (Cycles.balance()));
      Cycles.add(100_500_000_000);
      let newNFT = await NFTActorClass.NFT(name, owner, imgData);
      Debug.print(debug_show (Cycles.balance()));

      let newNFTPrincipal = await newNFT.getCanisterId();

      mapOfNFTs.put(newNFTPrincipal, newNFT);
      addToOwnershipMap(owner, newNFTPrincipal);

      return newNFTPrincipal

   };

   private func addToOwnershipMap(owner : Principal, nftId : Principal) {
      var ownedNFTs : List.List<Principal> = switch (mapOfOwners.get(owner))
         {case null List.nil<Principal>();
         case (?result) result;
      };

      ownedNFTs := List.push(nftId, ownedNFTs);
      mapOfOwners.put(owner, ownedNFTs);

   };

   public query func getOwnedNFTs(user : Principal) : async [Principal] {
      var userNFTs : List.List<Principal> = switch (mapOfOwners.get(user))
         {case null List.nil<Principal>();
```

```
    case (?result) result;
  };

  return List.toArray(userNFTs);
};

public query func getListedNFTs() : async [Principal]
  {let ids = Iter.toArray(mapOfListings.keys());
  return ids;
};

public shared (msg) func listItem(id : Principal, price : Nat) : async Text
  {var item : NFTActorClass.NFT = switch (mapOfNFTs.get(id)) {
    case null return "NFT does not exist.";
    case (?result) result;
  };

  let owner = await item.getOwner();
  if (Principal.equal(owner, msg.caller))
    {let newListing : Listing = {
      itemOwner = owner;
      itemPrice = price;
    };
    mapOfListings.put(id, newListing);
    return "Success";
  } else {
    return "You don't own the NFT.";
  };
};

public query func getOpenDCanisterID() : async Principal
  {return Principal.fromActor(OpenD);
};

public query func isListed(id : Principal) : async Bool
  {if (mapOfListings.get(id) == null) {
    return false;
  } else {
    return true;
  };
};

public query func getOriginalOwner(id : Principal) : async Principal
  {var listing : Listing = switch (mapOfListings.get(id)) {
    case null return Principal.fromText("");
    case (?result) result;
  };

  return listing.itemOwner;
};
```

```
   public query func getListedNFTPrice(id : Principal) : async Nat
      {var listing : Listing = switch (mapOfListings.get(id)) {
         case null return 0;
         case (?result) result;
      };

      return listing.itemPrice;

   };

   public shared (msg) func completePurchase(id : Principal, ownerId : Principal,
newOwnerId : Principal) : async Text {
      var purchasedNFT : NFTActorClass.NFT = switch (mapOfNFTs.get(id))
         {case null return "NFT does not exist";
         case (?result) result;
      };

      let transferResult = await purchasedNFT.transferOwnership(newOwnerId);

      if (transferResult == "Success") {
         mapOfListings.delete(id);
         var ownedNFTs : List.List<Principal> = switch (mapOfOwners.get(ownerId))
            {case null List.nil<Principal>();
            case (?result) result;
         };
         ownedNFTs :=
            List.filter(ownedNFTs,
            func(listItemId : Principal) : Bool
               {return listItemId != id;
            },
         );

         addToOwnershipMap(newOwnerId, id);
         return "Success";
      } else {
         return transferResult;
      };

   };

};
```

**nft.mo:**

```
import Debug "mo:base/Debug";
import Principal "mo:base/Principal";

actor class NFT(name : Text, owner : Principal, content : [Nat8]) = this

   {private let itemName = name;
```

```
  private var nftOwner = owner;
  private let imageBytes = content;

  public query func getName() : async Text
    {return itemName;
  };

  public query func getOwner() : async Principal
    {return nftOwner;
  };

  public query func getAsset() : async [Nat8]
    {return imageBytes;
  };

  public query func getCanisterId() : async Principal
    {return Principal.fromActor(this);
  };

  public shared (msg) func transferOwnership(newOwner : Principal) : async Text
    {if (msg.caller == nftOwner) {
      nftOwner := newOwner;
      return "Success";
    } else {
      return "Error: Not initated by NFT Owner.";
    };
  };

};
```

# USER MANUAL:

## To run the  NFTForge web app you need to perform the following stepsas mentioned:

**Step 1:** Open both the project folder in VS code or any other IDE.

**Step 2:** You must have Motoko (DFX) downloaded in either your system or on a remote desktop.

**Step 3:** Open Terminal inside the project folder of  NFTForge and write dfx start --clean

**Step 4:** Open another terminal and write

```
dfx deploy --argument='("Hello World #123", principal "iqjlv-qbnxm-xolci-ocjap-x6f76-
xu5sd-jqwl5-b5emu-ppitc-khrcl-tae", (vec {137; 80; 78; 71; 13; 10; 26; 10; 0; 0; 0; 13; 73; 72;
68; 82; 0; 0; 0; 10; 0; 0; 0; 10; 8; 6; 0; 0; 0; 141; 50; 207; 189; 0; 0; 0; 1; 115; 82; 71; 66; 0;
174; 206; 28; 233; 0; 0; 0; 68; 101; 88; 73; 102; 77; 77; 0; 42; 0; 0; 0; 8; 0; 1; 135; 105; 0; 4;
0; 0; 0; 1; 0; 0; 0; 26; 0; 0; 0; 0; 0; 3; 160; 1; 0; 3; 0; 0; 0; 1; 0; 1; 0; 0; 160; 2; 0; 4; 0; 0; 0; 1;
0; 0; 0; 10; 160; 3; 0; 4; 0; 0; 0; 1; 0; 0; 0; 10; 0; 0; 0; 0; 59; 120; 184; 245; 0; 0; 0; 113; 73;
68; 65; 84; 24; 25; 133; 143; 203; 13; 128; 48; 12; 67; 147; 94; 97; 30; 24; 0; 198; 134; 1; 96;
30; 56; 151; 56; 212; 85; 68; 17; 88; 106; 243; 241; 235; 39; 42; 183; 114; 137; 12; 106; 73;
236; 105; 98; 227; 152; 6; 193; 42; 114; 40; 214; 126; 50; 52; 8; 74; 183; 108; 158; 159; 243;
40; 253; 186; 75; 122; 131; 64; 0; 160; 192; 168; 109; 241; 47; 244; 154; 152; 112; 237; 159;
252; 105; 64; 95; 48; 61; 12; 3; 61; 167; 244; 38; 33; 43; 148; 96; 3; 71; 8; 102; 4; 43; 140;
164; 168; 250; 23; 219; 242; 38; 84; 91; 18; 112; 63; 0; 0; 0; 0; 73; 69; 78; 68; 174;
66; 96; 130;}))'
```

Replace principal field with your own principal id

**Step 5:** Open another terminal and write npm start

**Step 6:** Open NAAVY Token folder and open the terminal. Write dfx deploy

**Step 7:** Open another terminal and write npm start