# NFT Marketplace Smart Contract Development

Fan Wu
UNI:fw2392
fw2392@columbia.edu

Jiamiao He
UNI:jh4593
jh4593@columbia.edu

## Abstract

*In this project, we have developed a decentralized marketplace for non-fungible tokens (NFTs) using the Ethereum blockchain. The platform allows users to create, list, buy, and sell unique digital assets, fostering a secure and transparent environment for digital art and collectibles. Our implementation takes advantage of the robust capabilities of the ERC721 standard for non-fungible tokens, and we have employed the widely-accepted OpenZeppelin library to ensure the robustness and security of our smart contracts. The project consists of two primary components: an NFT smart contract responsible for creating and managing digital assets and a marketplace smart contract that handles the listing, buying, and canceling of NFTs. Additionally, our solution incorporates user-friendly functionalities that enable seamless interaction with the platform. We have also implemented relevant security checks and validations to prevent unauthorized actions and ensure compliance with the ERC721 standard. Throughout the development process, we have adhered to best practices in smart contract security and conducted comprehensive testing using the Truffle framework to ensure a seamless user experience upon deployment. By integrating the latest blockchain technologies, our NFT marketplace offers an innovative solution for creators and collectors alike.*

## 1. Introduction

The rapid growth of the digital asset ecosystem has led to an increased demand for decentralized platforms that allow users to trade and exchange non-fungible tokens (NFTs). NFTs are unique digital assets representing ownership of art, collectibles, or other digital items, and they have gained significant traction in recent years. This project aims to develop a secure, user-friendly, and efficient NFT marketplace based on the Ethereum blockchain to cater to this growing market. This report presents the process of designing, developing, testing, and deploying the smart contracts and associated components of the NFT marketplace. Built on the Ethereum blockchain and leveraging the ERC721 token standard, the marketplace allows users to create, list, buy, and cancel NFTs in a decentralized manner. OpenZeppelin, a widely used library, is employed to ensure adherence to industry best practices and enhance smart contracts' security and performance. By integrating these cutting-edge technologies, the NFT marketplace achieves a high level of reliability and scalability, allowing for future improvements and additional features.

This NFT marketplace project is built on two smart contracts: the NFT contract and the marketplace contract, which work together on the Ethereum blockchain. The NFT contract utilizes the ERC721 standard and the OpenZeppelin library to create and manage unique, non-fungible tokens representing digital assets. The "Methodology" section details the listing, trading, and security aspects of the marketplace contract and NFTs. In the "Procedure" section, the project's adopted security measures are discussed, including using the OpenZeppelin library and various validation checks to protect users' digital assets and transactions. The development process encompasses architecture design, core functionality implementation using Solidity, OpenZeppelin contract integration, and security measure enforcement. The "Testing" section covers unit, integration, and edge case testing using the Truffle framework to ensure the marketplace's proper functionality and robustness. Lastly, deployment on the Ethereum blockchain is carried out with Truffle, resulting in a fully functional and accessible NFT marketplace for users. The platform allows for the efficient creation, listing, purchasing, and cancellation of NFTs, offering a comprehensive NFT trading experience.

## 2. Methodology

### 2.1. Platform architecture

The NFT market project's platform architecture comprises two primary components: NFT smart contracts and market smart contracts. These elements work harmoniously to offer a safe and intuitive environment for creating, listing, purchasing, and selling NFTs on the Ethereum blockchain.

Utilizing the ERC721 standard and the OpenZeppelin library, NFT smart contracts handle the creation and man-

agement of unique, non-fungible tokens that represent digital assets. These contracts include fundamental functions such as generating new NFTs and accessing their metadata, ensuring compatibility with existing wallet services and other platforms. Adhering to the ERC721 standard, the NFT smart contracts guarantee that each token is distinctive, traceable, and transferable, enabling secure and dependable asset ownership.

Serving as the heart of the NFT market, the smart market contracts streamline the listing and trading of NFTs. They securely store NFTs for sale and oversee transactions between buyers and sellers. Smart market contracts incorporate various security measures, such as ownership and price verification, to safeguard user assets. Furthermore, the contracts employ OpenZeppelin's ERC721Holder functionality, enabling them to hold and transfer NFTs on users' behalf securely.

The smooth interplay between these components allows users to confidently create, list confidently, and trade NFTs. By establishing a secure and efficient architecture, this NFT marketplace offers users an accessible and dependable platform to engage with digital assets, promoting growth and innovation in the rapidly evolving world of non-fungible tokens.

### 2.2. Standard and libraries

This NFT marketplace project harnesses essential standards and libraries to guarantee interoperability, security, and user-friendliness. By utilizing these well-established resources, the platform ensures compatibility with other applications and services within the Ethereum ecosystem while taking advantage of its solid development and community support. The ERC721 standard serves as the basis for the NFT smart contract, outlining the structure and functionality of non-fungible tokens. As a broadly recognized standard in the blockchain domain, ERC721 ensures that each NFT is unique, indivisible, and transferable, making it perfect for representing digital assets with distinct attributes and value. By complying with the ERC721 standard, this project guarantees that its NFTs are compatible with existing wallet services, marketplaces, and other platforms supporting this token standard.

The OpenZeppelin library is vital for developing smart contracts in this project. OpenZeppelin comprises secure, audited, and community-driven smart contract components built on the Ethereum blockchain. By integrating OpenZeppelin's contracts, the project reaps the benefits of their thoroughly tested functionality and best practices, guaranteeing high security and dependability. Some notable OpenZeppelin components used in this project include the ERC721.sol contract for NFT creation and management, the IERC721Receiver.sol contract for handling NFT transfers, and the ERC721Holder.sol contract for secure token holding in the marketplace. By employing the ERC721 standard and the OpenZeppelin library, this NFT marketplace project establishes a robust foundation for a secure, reliable, and interoperable platform. These standards and libraries streamline development and encourage the adoption of the marketplace by users familiar with the broader Ethereum ecosystem.

### 2.3. Security measures

The NFT marketplace project integrates multiple security measures to protect the platform and its users. By implementing these safeguards, the platform ensures the secure and reliable operation of its services, ultimately safeguarding users' digital assets and transactions. Firstly, the project utilizes the OpenZeppelin library, a well-known and widely-used collection of audited smart contract components. OpenZeppelin's contracts adhere to industry best practices and undergo extensive testing for security vulnerabilities, minimizing the risk of attacks and exploits. By incorporating these specific components, the marketplace can offer a robust and reliable environment for trading NFTs.

In addition, the smart contract adopts ERC721 secure blockchain technology to ensure information security through public transaction records, encryption technology, and digital signatures. This protects unique and non-fungible tokens from fraud, spoofing, and cyberattacks while protecting the privacy and assets of token owners, such as listing, purchasing, or delisting NFTs.. For instance, the contracts ensure that only the rightful owners can list their NFTs for sale, and they verify that the buyer has sent the correct payment amount. Additionally, the contracts utilize the safeTransferFrom function to securely transfer NFT ownership while mitigating risks associated with potential reentrancy attacks. These restrictions guarantee that only the rightful owners can interact with their tokens and that transactions are executed fairly and transparently. The contracts also use events to emit relevant information about NFT listings, sales, and cancellations, providing users with a transparent and auditable record of platform activities. By incorporating these security measures and leveraging proven libraries like OpenZeppelin, the NFT marketplace project provides users a safe and secure platform to trade and manage their digital assets.

## 3. Procedure

### 3.1. Smart contract development

The Smart Contract Development process can be divided into several critical steps, including architecture design, contract implementation, library integration, and security measures. The architecture consists of two primary smart contracts: the NFT and Market contracts. The NFT contract, based on the ERC721 standard, is responsible for

creating, managing, and transferring unique NFTs, each associated with a specific name and description. The Market contract, on the other hand, enables users to list their NFTs for sale, buy listed NFTs, and cancel their listings if desired. These contracts interact with each other to form the foundation of the decentralized marketplace.

We utilized Solidity, a high-level programming language for implementing smart contracts on the Ethereum blockchain, during the implementation phase. The NFT contract allows users to create new NFTs using the createNFT function, which mints a new token and associates it with the provided metadata. The getNFT function allows users to retrieve information about a specific NFT by its token ID. The Market contract, in contrast, facilitates the listing, purchasing, and cancellation of NFT listings. Functions such as ListToken, BuyToken, and cancel or enable these core marketplace operations. To ensure compatibility and reduce code complexity, we leveraged the OpenZeppelin Contracts library, a widely-adopted set of secure and audited smart contracts for Ethereum. This library provided the necessary building blocks, such as the ERC721 contracts and various utility functions to simplify the development process.

Security is paramount in smart contract development, as vulnerabilities can lead to irreversible consequences. To mitigate potential risks, we have adopted several best practices:

- Use the safeTransferFrom function from the ERC721 standard to prevent unauthorized transfers of tokens.
- Employ various required statements to validate user inputs and ensure only eligible users can perform specific actions, such as listing or canceling NFTs.
- Utilize events, such as Listed, Sale, and Cancel, to emit crucial information about contract operations for easier monitoring and auditing.

In conclusion, the Smart Contract Development process for this decentralized NFT marketplace involved designing a comprehensive architecture, implementing the core functionalities using Solidity and the ERC721 standard, integrating well-established libraries such as OpenZeppelin Contracts, and enforcing strict security measures to protect users and their digital assets.

### 3.2. Testing

The testing phase of this NFT marketplace project aims to ensure the correct functionality and robustness of the implemented smart contracts. We employed a comprehensive testing strategy that combines unit testing, integration testing, and edge case testing to validate the marketplace's behavior under various scenarios thoroughly. The tests were conducted using the Truffle framework, which offers tools for developing and testing smart contracts on the Ethereum blockchain.

Unit testing focused on validating the behavior of individual functions within the NFT and Market contracts. In the NFT contract, we verified that the ‘createNFT‘ function successfully mints new tokens and associates them with the correct metadata. We also tested the ‘getNFT‘ function to ensure it retrieves accurate information about a specific NFT. For the Market contract, we examined the core operations of listing, buying, and canceling NFTs through functions such as ‘ListToken,‘ ‘BuyToken,‘ and ‘cancel.‘ We ensured these functions worked as intended, properly updating the contract state and transferring tokens between users (figure 1) ).

Integration testing assessed the interaction between the NFT and Market contracts. For instance, we confirmed that NFTs created using the NFT contract could be successfully listed and purchased on the marketplace through the Market contract. Additionally, we validated that the ownership of the NFTs is correctly updated throughout the various stages of the marketplace process.

Edge case testing addressed potential vulnerabilities and unexpected scenarios in smart contracts. We used the ‘expectRevert‘ and ‘expectEvent‘ functions from the OpenZeppelin Test Helpers library to simulate erroneous inputs and validate the contracts' error-handling capabilities. Some examples of edge cases include unauthorized users attempting to list or cancel NFTs, sellers attempting to buy their NFTs, and insufficient payments for purchasing NFTs. These tests ensured the contracts behaved correctly and securely, even under unfavorable circumstances.

In summary, the testing phase of this NFT marketplace project involved thoroughly examining the smart contracts' functionality, integration, and resilience to potential vulnerabilities. By employing a combination of unit, integration, and edge case testing using the Truffle framework and OpenZeppelin Test Helpers, we have ensured that the marketplace operates as intended and provides a secure platform for users to trade NFTs.

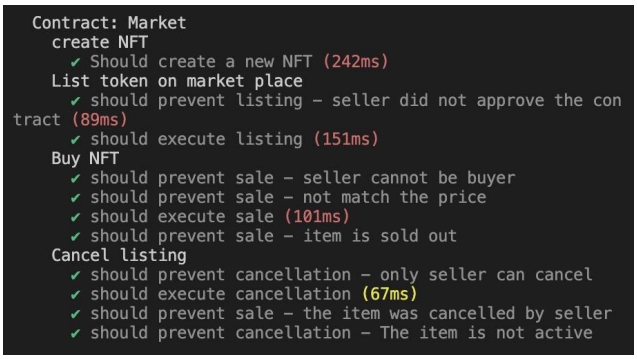


Figure 1. Test cases for the NFT marketplace smart contracts.

### 3.3. Deployment

The deployment stage of the NFT marketplace project involves transferring the developed and tested smart contracts onto the Ethereum blockchain. We utilized the Truffle framework to streamline the process, enabling easy deployment to various networks, such as local development, test networks (Ropsten, Rinkeby, etc.), or the leading Ethereum network. Configuring the Truffle migration scripts ensured a smooth deployment process that properly initialized the NFT and Market contracts. Once the contracts were deployed, their addresses and ABI were recorded to facilitate interaction with the marketplace through front-end applications and third-party tools. This crucial stage solidifies the project, making the NFT marketplace fully functional and accessible to users.

## 4. Results

### 4.1. Functionality

The NFT marketplace system's functionality enables users to create, list, buy, and cancel NFTs efficiently. The platform utilizes two smart contracts - one for NFT creation and another for the marketplace. The NFT creation contract allows users to mint unique tokens with distinct names and descriptions. The marketplace contract manages the listing, buying, and cancellation of NFTs. It ensures secure transfers of ownership and implements proper access control mechanisms. Users can list their NFTs by setting a price, and interested buyers can purchase the listed tokens by paying the specified amount. Sellers also have the option to cancel their listings, returning the ownership of the NFT to them. Overall, the platform's functionality successfully delivers a comprehensive NFT trading experience.

## 5. Showcase

We tested the feasibility of smart contracts using the Sepolia closed testnet. Firstly, we created two separate cryptocurrency accounts on MetaMask and connected them to the Sepolia testnet. We then utilized Remix IDE to deploy NFT and marketplace contracts to the Sepolia test network by selecting the environment as Injected Provider-MetaMask. A confirmation message (Figure 2 and 3) is received upon successful deployment of the smart contracts to the blockchain network, and the contracts are now available for others to use and interact with, as illustrated in the diagram (Figure 6).

Next, we used the first account to create a new NFT called "MyNFT" and listed it on the marketplace. However, before listing the NFT, we needed to approve the Market contract for access to transfer the NFT, as shown in Figure 7. We then invoked the ListToken function to transfer MyNFT to the Market contract for sale, also requiring account confirmation for security. Subsequently, we used the second account to purchase the NFT from the marketplace, with the second account needing to confirm the NFT purchase as figure 8.

In the end, we observed that the ownership of the NFT had changed from the first account (figure 4) to the second account's address (figure 5). This demonstration validates the security of our contracts, with each step's account confirmation being verified.

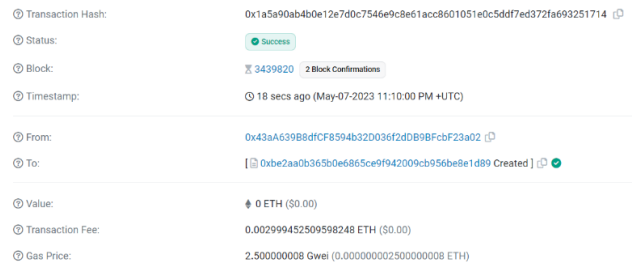


Figure 2. confirmation message is received

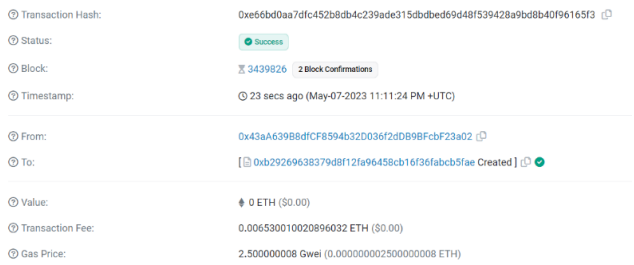


Figure 3. confirmation message is received

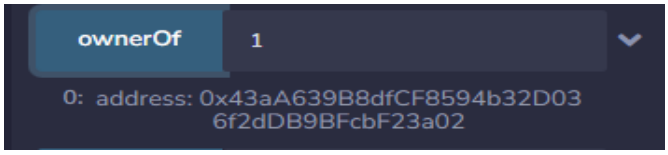


Figure 4. Owner Address

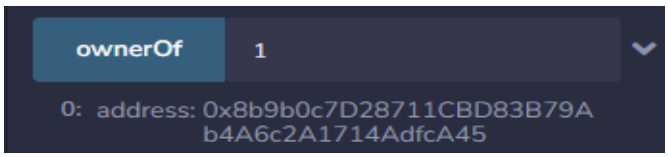


Figure 5. Owner Address

## 6. Conclusion

In conclusion, our team has developed a robust and user-friendly NFT marketplace for trading digital assets. This

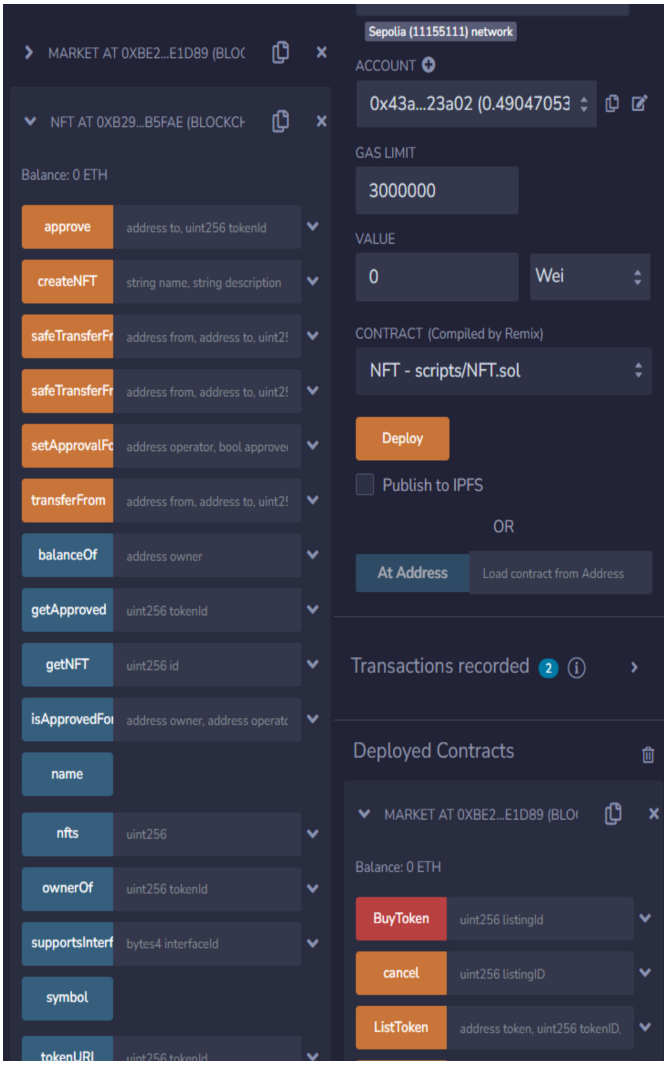


Figure 6. Function

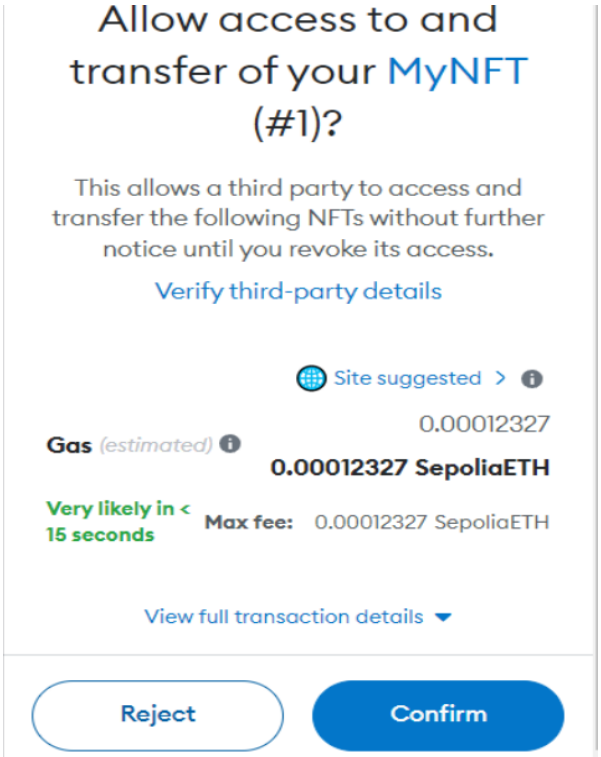


Figure 7. Approval Process for seller

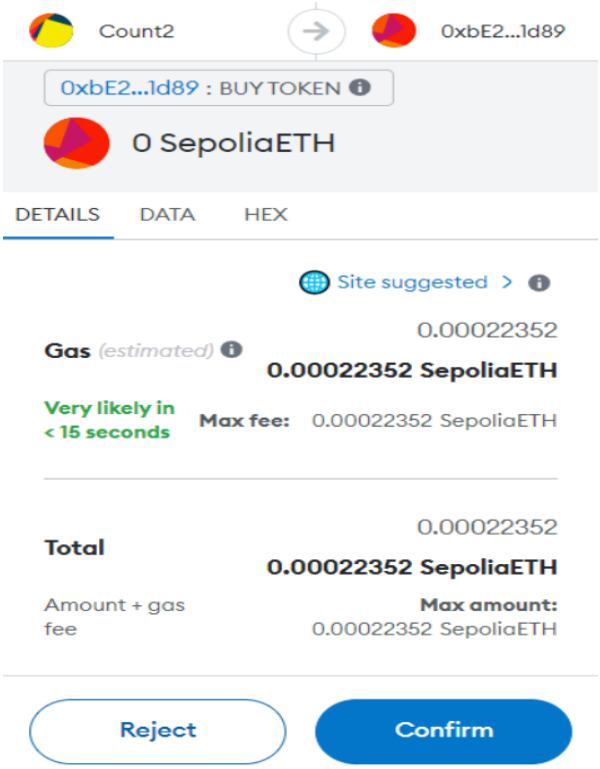


Figure 8. Approval Process for buyer

project is a result of the collaborative efforts of our dedicated team members; this project is the result of a joint effort of our dedicated team members, each bringing their best skills. We have acquired a deep understanding and hands-on experience with non-fungible tokens and Ethereum.

We have constructed the system on a solid foundation, utilizing the Ethereum blockchain and ERC721 token standard for the creation, listing, purchase, and cancellation of NFTs. By leveraging the OpenZeppelin library and following industry best practices, our marketplace guarantees its users' security, reliability, and performance. We have carefully executed the smart contract development, testing, and deployment processes to ensure seamless integration with the platform's front end. Our extensive testing suite ensures that the smart contracts' functionality and performance meet the expected requirements, delivering users a bug-free and efficient trading experience. The architecture

of our platform is designed for scalability and adaptability, paving the way for potential future enhancements and incorporating additional features. This NFT marketplace is an excellent starting point for further exploration into the rapidly expanding digital asset ecosystem. Ultimately, our project successfully showcases the transformative potential of blockchain technology in revolutionizing the way digital assets are traded and exchanged.

## 7. Github Repository

https://github.com/SquiffedFOX/ELEN6883_Final_Project.git