# Internship Project Report – Project Initiation

## 1. Intern Details

* **Intern Name:** Sunny Singh
* **Internship Domain:** Blockchain
* **Mentor/Supervisor Name:** Mr. Abhi Mitra
* **Mentor Email:** abhi.mitra@cutm.ac.in
* **Intern Email:** ss5494602@gmail.com

## 2. Project Title

NFT Market Place

## 3. Project Overview / Abstract

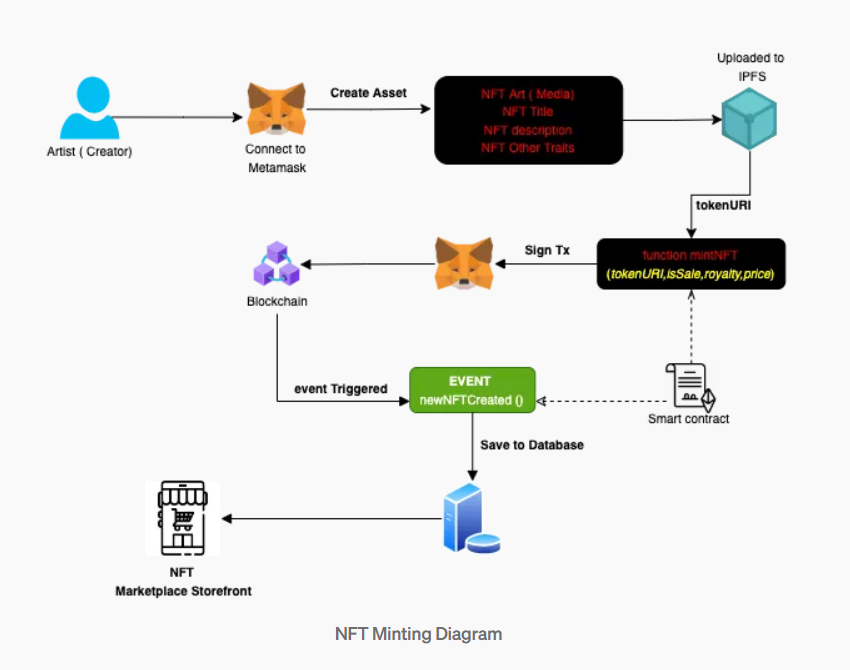
This project is a decentralized NFT marketplace designed for trading digital art like photo, drawing, digital art assets. Built on the Ethereum blockchain using smart contracts, it enables users to mint, list, buy, and sell unique NFTs securely and transparently. The platform leverages ERC-721 standards for NFT creation and integrates a fixed-price marketplace model. A streamlined frontend built with Next.js and Ethers.js allows seamless wallet connection and real-time blockchain interaction. By focusing on art categories, the marketplace offers a niche experience tailored to creators and collectors in these domains. The project emphasizes user ownership, low fees, and a clean, intuitive interface.

## 4. Objectives of the Project

* Develop a Functional NFT Marketplace
* Integrate Smart Contracts with Frontend
* Implement Wallet Connectivity
* Deploy and Test on Ethereum Testnet
* Document and Present the Project

*(List 3–5 clear objectives the intern aims to achieve during the internship)*

## 5. Tools & Technologies to be Used

* Programming Languages: Solidity, JavaScript
* Frameworks/Libraries: Next.js, Hardhat
* Databases: IPFS, SQL
* Tools/Software Platforms: Vs-code, Cursor
* Version Control: git  
    
  **Project Architecture :**  
    
  **NFT Minting Diagram**  
    
  
* Connect to MetaMask  
  The artist (creator) connects their crypto wallet using MetaMask.
* Create NFT Asset  
  The artist creates the NFT by providing:

NFT Art (Media)

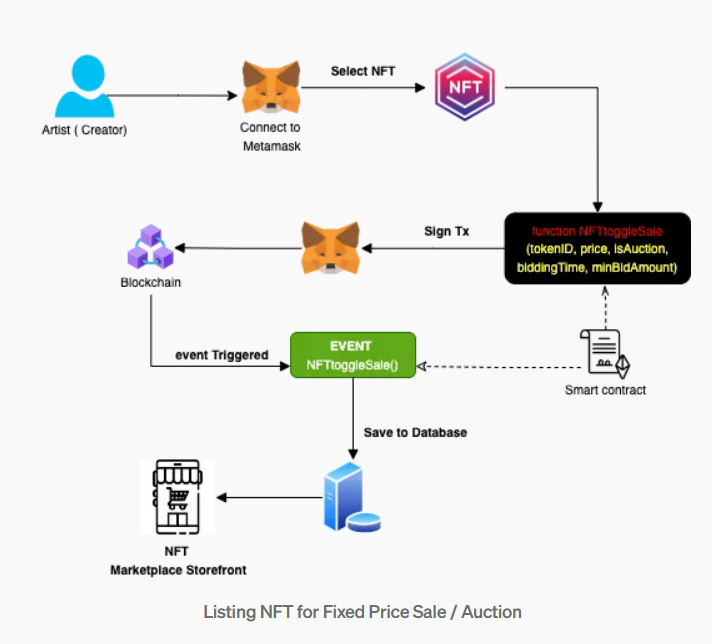
NFT Title

NFT Description

Other Traits

Upload to IPFS  
The asset is uploaded to IPFS, which generates a tokenURI (a unique link to the metadata).

* Sign Transaction  
  The artist signs a transaction using MetaMask to authorize minting.
* Call Smart Contract Function  
  The function mintNFT(tokenURI, isSale, royalty, price) is called with:
* tokenURI: Link to the IPFS metadata
* isSale: Boolean indicating if the NFT is for sale
* royalty: Creator's royalty percentage
* price: Listing price (if for sale)
* Trigger Event  
  The smart contract emits an event:
* Save to Database  
  The event data is stored in a backend database.
* NFT Listed on Marketplace  
  The NFT appears on the marketplace storefront, ready for sale or display.

**Listing NFT Fixed Price**  
  


* Artist (Creator) Begins :  
  The process starts with the artist or creator who owns the NFT.
* Connect to MetaMask  
  The artist connects their crypto wallet using MetaMask.
* Select NFT  
  The artist chooses the NFT they want to list for sale or auction.
* Sign Transaction (Tx)  
  The artist signs a transaction in MetaMask to authorize the listing.
* Call Smart Contract Function  
  The function NFTtoggleSale(tokenID, price, isAuction, biddingTime, minBidAmount) is called with:

tokenID: ID of the NFT

price: Listing price

isAuction: Boolean (true for auction, false for fixed price)

biddingTime: Duration of the auction (if applicable)

minBidAmount: Minimum bid amount (if auction)

* Send to Blockchain  
  The signed transaction is sent to the blockchain network.
* Trigger Event  
  The smart contract emits an event:
* Save to Database  
  The event data is stored in a backend database for frontend use.
* NFT Listed on Marketplace  
  The NFT is now visible on the NFT Marketplace Storefront, available for sale or auction.

## 6. Project Scope

* This project focuses on the development of a decentralized NFT marketplace specifically for digital art. It involves designing and deploying secure smart contracts using Solidity, following the ERC-721 standard for non-fungible tokens. These contracts will be deployed on the Ethereum Sepolia testnet to simulate real-world blockchain interactions in a safe environment.
* The frontend will be developed using the Next.js framework, providing a responsive and user-friendly interface. Wallet integration through MetaMask will allow users to authenticate and interact with the blockchain seamlessly. Core functionalities will include minting new NFTs, listing them for sale, purchasing listed NFTs, and canceling active listings.
* To ensure decentralized and secure storage, the project will integrate IPFS for hosting NFT metadata and associated digital art files. A clean and intuitive UI will enable artists and collectors to interact with the marketplace efficiently and transparently.

**Out of scope :**

* Gaming Asset Integration : The marketplace will not support NFTs related to gaming items such as characters, weapons, or in-game collectibles.
* Auction or Bidding Mechanisms  
  Only fixed-price listings are supported. Features like timed auctions or bidding systems are not included.
* Multi-Chain Support  
  The platform will operate solely on the Ethereum testnet (e.g., Sepolia). Integration with other blockchains like Polygon, Solana, or BNB Chain is not part of this project.
* Advanced User Features  
  Features such as user profiles, social interactions (likes, comments), or messaging are excluded.
* Mobile Application Development  
  The project will focus only on a web-based interface. No mobile app (iOS/Android) will be developed.

*(Briefly define the expected scope and boundaries of the project. What is included and what is out of scope?)*

## 7. Timeline

|  |  |
| --- | --- |
| **Week** | **Activity / Milestone** |
| 1 | Understanding Requirements, Research |
| 2 | System Design / Architecture Planning |
| 3 | Core Module Development Begins |
| 4 | Development & Integration |
| 5 | Testing and debugging |
| 6 | Finalization and Documentation |

## 8. Expected Deliverables

* Project Documentation
* Source Code / Repository
* Working Prototype / Demo
* Presentation / Report Submission

## 9. Signature & Acknowledgement

**Intern Signature: sunny singh**

**Date:**

**Mentor/Supervisor Signature:**

**Date:**