



**NFT Marketplace UX UI**

Simple, catchy and easy to navigate  
Adaptation of look & feel for easy navigation on mobile screen

**Smart Contract Development**

Automatic execution as end users purchase NFT directly from the Client

Accurate submission of transactions together with the relevant gas fees

**Metamask Contract Integration**

**Chainlink System Integration**

Smoothly connecting the smart contract with off-chain ecosystem

Final check by both the developers and the clients to confirm if everything works well

**Final Test Audit**

## TECH STACK

**REACT JS**

For front end web development

**PINATA CLOUD**

For the IPFS storage

**AWS EC2**

For the cloud server

**AWS S3**

For image and file server

**SOLIDITY PROGRAMMING**

For any EVM chain

**MORALIS DATABASE**

For off-chain and on-chain data management

**ETHEREUM (or any EVM chain)**

For the blockchain

## PROPOSED ARCHITECTURE

**ERC 721 Token Contract**

- ✓ Used to lock and replace periphery contracts and control minting or burning
- ✓ 2 main roles: Minter and owner

**Seeder**

- ✓ Currently, traits are determined using pseudo-random number generation
- ✓ Determines traits during minting, and can be replaced to allow for future trait generation algorithm upgrades

**Descriptor**

- ✓ Used to store or render artwork and build token URIs
- ✓ When token URI is called, NFT parts are read and converted into a SVG image on-chain
- ✓ The token URI then consists of base64 encoded data URI with the JSON contents directly inlined, including the SVG image