StuffSwap

Keenan Heller, Rachel Hodson, Ariana Moreno, Olga Ortega, Audell Sabeti

Project Goals

- Create an online marketplace / ecommerce website similar to OpenSea called “Stuff Swap”
- Create an NFT Collection to be listed for sale on Stuff Swap called “Bag N Tag” where the NFTs are 3D art of duffle bags
- Code Stuff Swap to show Bag N Tag listed for sale
- Demonstrate a mock transaction where Ethereum is exchanged for an NFT from Bag N Tag

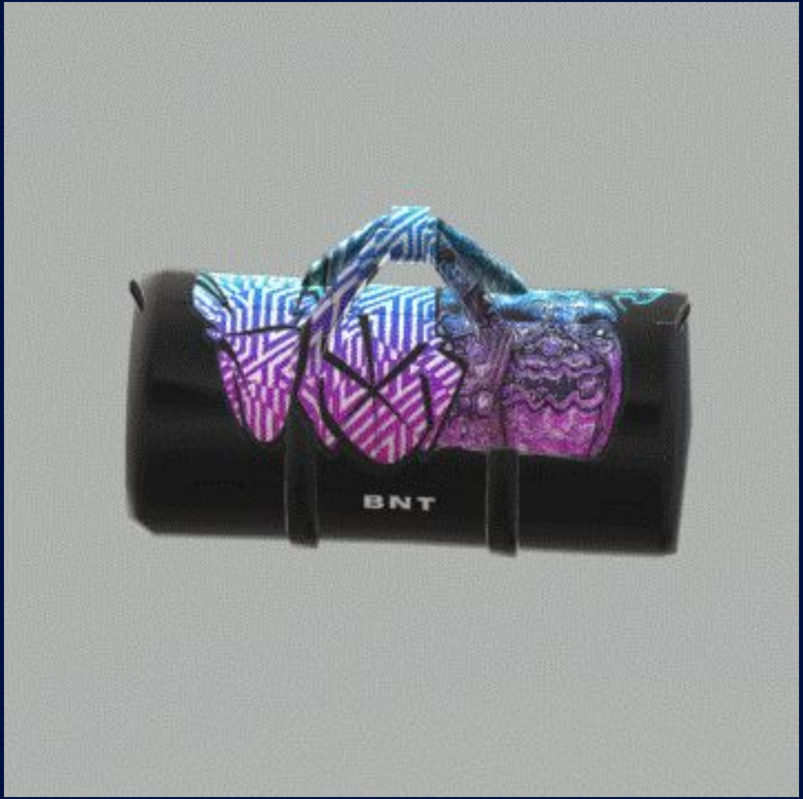
Why did we create an NFT?

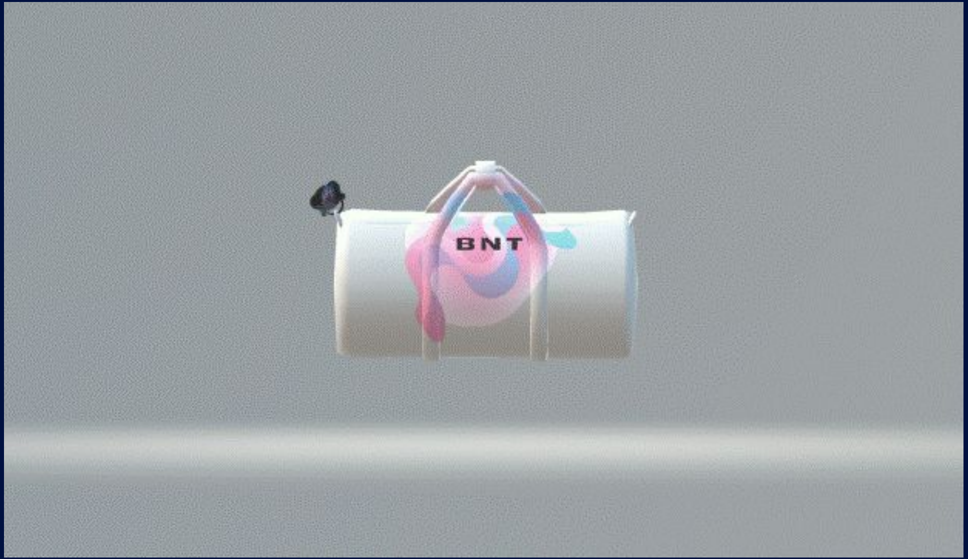
- We were really interested in NFTs and Audell had already begun creating 3D artwork of duffle bags to be sold as NFTs
- Keenan is very interested in website design and thought it would be really great practice to try to create our own online ecommerce marketplace to sell Audell’s NFTs on instead of just using streamlit
- We brainstormed some name ideas and landed on “Stuff Swap” for the website and “Bag N Tag” for the NFT Collection

What is Stuff Swap?

- Similar to OpenSea, Stuff Swap is an online marketplace / ecommerce website
- Users can
  - Create a personal user profile
  - Put their NFT Collections up for sale
  - Purchase NFTs from other creators
  - Process payment through the website directly using cash-backed currency or cryptocurrency

Our NFT Collection:
Bag N Tag



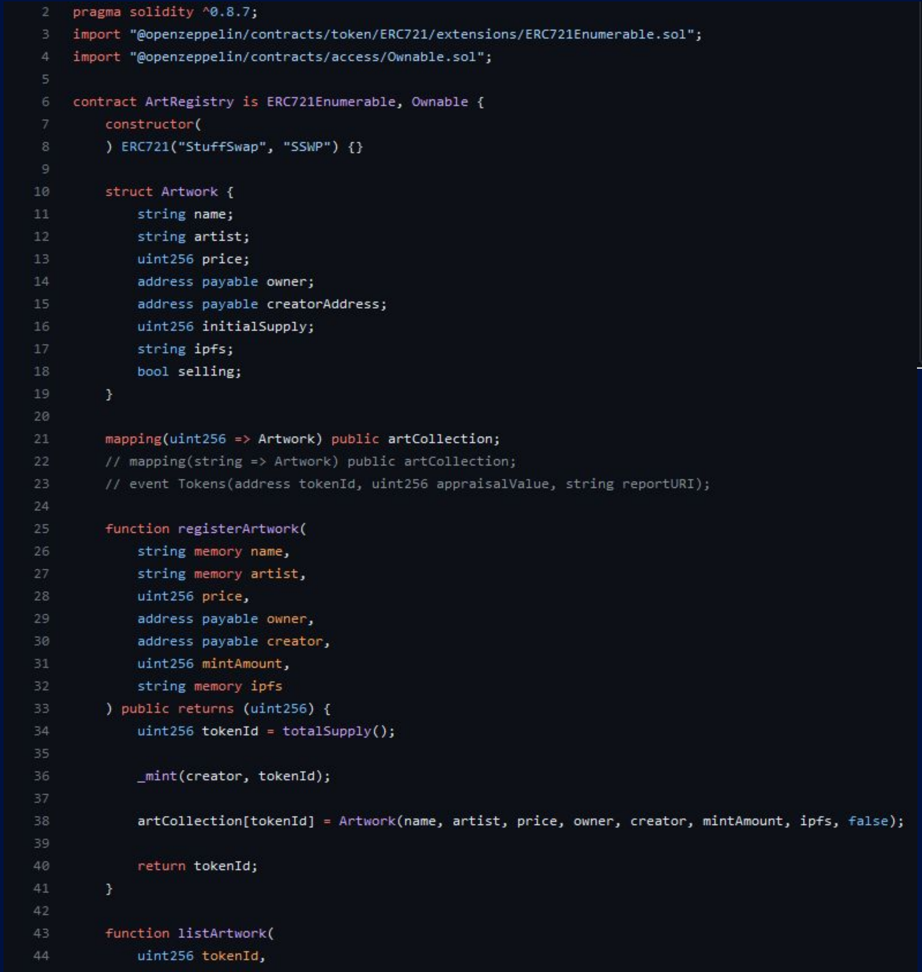


Collection Cont.

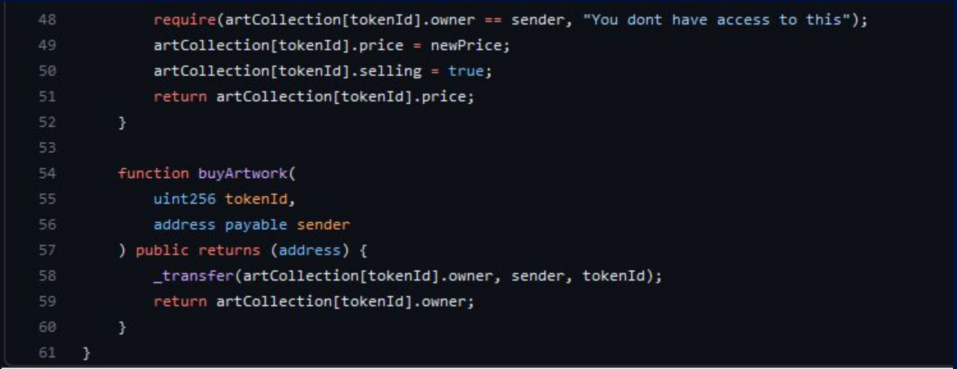
First Attempted Smart Contract

- Initially, there was some confusion regarding the number of smart contracts required such as the marketplace contrast, NFT we created for the marketplace contract, and the Art registry contract.
- We thought we needed multiple smart contracts, but upon showing the NFT contract code to Binoy, he pointed out that there were some errors and unnecessary code
- Eventually, we discovered that we could use the art registry code as a template and create a single smart contract and due to time constraints, we opted for this approach

Final Smart Contract Full

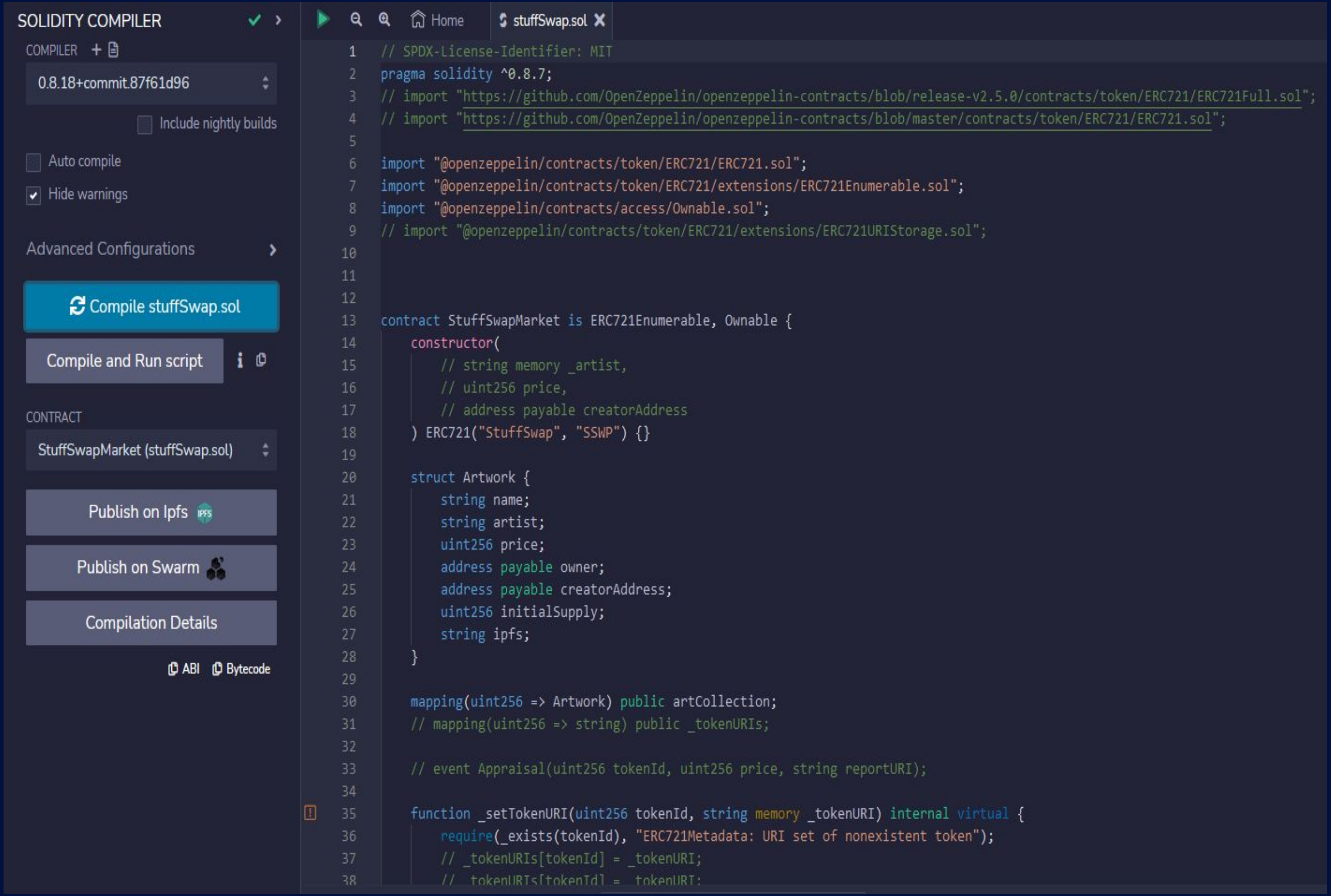


```solidity
pragma solidity ^0.8.7;
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol";
import "@openzeppelin/contracts/access/Ownable.sol";

contract ArtRegistry is ERC721Enumerable, Ownable {
    constructor(
    ) ERC721("StuffSwap", "SSWP") {}

    struct Artwork {
        string name;
        string artist;
        uint256 price;
        address payable owner;
        address payable creatorAddress;
        uint256 initialSupply;
        string ipfs;
        bool selling;
    }

    mapping(uint256 => Artwork) public artCollection;
    // mapping(string => Artwork) public artCollection;
    // event Tokens(address tokenId, uint256 appraisalValue, string reportURI);

    function registerArtwork(
        string memory name,
        string memory artist,
        uint256 price,
        address payable owner,
        address payable creator,
        uint256 mintAmount,
        string memory ipfs
    ) public returns (uint256) {
        uint256 tokenId = totalSupply();

        _mint(creator, tokenId);

        artCollection[tokenId] = Artwork(name, artist, price, owner, creator, mintAmount, ipfs, false);

        return tokenId;
    }

    function listArtwork(
        uint256 tokenId,
```



```solidity
        require(artCollection[tokenId].owner == sender, "You dont have access to this");
        artCollection[tokenId].price = newPrice;
        artCollection[tokenId].selling = true;
        return artCollection[tokenId].price;
    }

    function buyArtwork(
        uint256 tokenId,
        address payable sender
    ) public returns (address) {
        _transfer(artCollection[tokenId].owner, sender, tokenId);
        return artCollection[tokenId].owner;
    }
}
```

Successful Compilation



COMPILER
0.8.18+commit.87f61d96

□ Include nightly builds

□ Auto compile

☑ Hide warnings

CONTRACT
StuffSwapMarket (stuffSwap.sol)

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.7;
// import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/token/ERC721/ERC721Full.sol";
// import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC721/ERC721.sol";

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
// import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";



contract StuffSwapMarket is ERC721Enumerable, Ownable {
    constructor(
        // string memory _artist,
        // uint256 price,
        // address payable creatorAddress
    ) ERC721("StuffSwap", "SSWP") {}

    struct Artwork {
        string name;
        string artist;
        uint256 price;
        address payable owner;
        address payable creatorAddress;
        uint256 initialSupply;
        string ipfs;
    }

    mapping(uint256 => Artwork) public artCollection;
    // mapping(uint256 => string) public _tokenURIs;

    // event Appraisal(uint256 tokenId, uint256 price, string reportURI);

    function _setTokenURI(uint256 tokenId, string memory _tokenURI) internal virtual {
        require(_exists(tokenId), "ERC721Metadata: URI set of nonexistent token");
        // _tokenURIs[tokenId] = _tokenURI;
        // _tokenURIs[tokenId] = _tokenURI;
```

Next Steps

- If we could afford it, it would be really cool to actually launch the website!
- Add more NFT Collections
- Market the website
- Integrate the website with Coinbase or other websites which hold cryptocurrency wallets to improve customer experience
- Integrate the website with PayPal or Google Pay or Apple Pay also to improve customer experience