**ERC1155 Project Demo**

**INTRODUCTION:**

A contracts that manage multiple token. A single deployed contract may include any combination of fungible tokens, non-fungible tokens or other configurations (e.g. semi-fungible tokens).

**What is meant by Multi-Token Standard:-**

The idea is simple and seeks to create a smart contract interface that can represent and control any number of fungible and non-fungible token types.

**ERC-1155 FUNCTIONS AND FEATURES:**

[Batch Transfer](https://ethereum.org/en/developers/docs/standards/tokens/erc-1155/#batch_transfers): Transfer multiple assets in a single call.

[Batch Balance](https://ethereum.org/en/developers/docs/standards/tokens/erc-1155/#batch_balance): Get the balances of multiple assets in a single call.

[Batch Approval](https://ethereum.org/en/developers/docs/standards/tokens/erc-1155/#batch_approval): Approve all tokens to an address.

[Hooks](https://ethereum.org/en/developers/docs/standards/tokens/erc-1155/#recieve_hook): Receive tokens hook.

[NFT Support](https://ethereum.org/en/developers/docs/standards/tokens/erc-1155/#nft_support): If supply is only 1, treat it as NFT.

[Safe Transfer Rules](https://ethereum.org/en/developers/docs/standards/tokens/erc-1155/#safe_transfer_rule): Set of rules for secure transfer.

**Required Tools & Steps by Steps process of this project**

**FlowerCollection Contract Address** : 0x0adfa76De007c0F68eB157B5611Aba0A57cB8313

**Test net** : Rinkeby Test Network

**Wallet** : Meta mask

**Development Tool** : Remix

**Steps 1:-** Collects images or items which you want to create set of NFT or Tokenid. Here, I collected the set of 43 flowers images to create NFT and using Pinata gateway to upload images on IPFS( **InterPlanetary File System**) protocol to create hash URL of all set of flowers.

Graphical user interface, text, application

Description automatically generated

[**CID of flower\_erc1155\_nft\_collection :-**](CID:-) QmUrGsnXcvQgJDSAFnsomjT9nQWdDPrbscu9x4Y48EEFkm

**Steps 2:-** After upload on IPFS protocol to create hash of all set of flower, next create JSON files of all collections which is called **metadata file** and also using pinata gateway to upload JSON file on IPFS protocol.

Graphical user interface, text, application

Description automatically generated

**CID of flower\_erc1155\_metadata :-** QmW6BQqh6zRtcFtaqUNaxnDEzDtfG24dHaLp7txXoQuMhv

**Steps 3:-** Next, use the **CID of flower\_erc1155\_metadata**  in ERC1155 constructor for \_setURL method.

Text

Description automatically generated

**Steps 4:-** Next, Select the **Injected Web3** Environment in remix to connect with **metamask Rinkeby Test Network.**

Graphical user interface, application

Description automatically generated

**Steps 5:-** After connect with rinkeby test network , Next Click on Deploy Button and deploy the FlowerCollection.sol contract on test network.

A screenshot of a computer

Description automatically generated with medium confidence

After deploy contract on test environment, metamask transaction **confirm button** popup is showing and finally click on confirm button.

A screenshot of a computer

Description automatically generated with medium confidence

Contract Address **(0x0adfa76De007c0F68eB157B5611Aba0A57cB8313)** is created.

Graphical user interface

Description automatically generated

This Confirm transaction of deploy contract has hashs:-

**0xacf83e58cdd6c7ac0882e15d295542e6831e8221781ee30037fd18d741affb6c**

Graphical user interface, text, application, email

Description automatically generated

**Steps 6:-** then, run the **estimate\_gas\_fees.js** file in scripts folder to get the **TokenId , Amounts , gas required ,gas cost estimation and block gas limit.**

Graphical user interface, text, application

Description automatically generated

Screenshot of estimate gas cost information.

Text

Description automatically generated

**Steps 7:-** then, run **bulk\_mint\_nfts.js** file in **script folder** for batch mint of JSON file of IPFS images. Right Click on file and click on run, then your script will be executed.

Graphical user interface, text

Description automatically generated

Then, **metamask confirm button** popup is showing ,after clicking on **confirm button** your transaction will be completed of batch mint of ERC1155.

Graphical user interface, text, application

Description automatically generated

After confirm transection, our flower collection of images has been **batchmint** with 1 balances of ERC1155.

Text

Description automatically generated

This Confirm transaction has 1 TokenID to 25 TokenID BatchMint NFT out of 43 NFT create hash is:-**0xb7587b155b4b25541267dfd2eeba2a40d969ac7b3950cab8abfde67c86cef4a8**

Graphical user interface, text, email

Description automatically generated

This Confirm transaction has 26 TokenID to 43 TokenId batchmint NFT create hash is:-

**0x074d090e7c9a034e7d5afb9a8ac7af8cd9b2a4fdb39ece4c4a811b34b895c8ef**

Graphical user interface, text, application, email

Description automatically generated

**Testing of ERC1155 After BatchMint of Flower Images**

1. **BalanceOfBatch:-**

This method we used to check the balance of more then one tokenId against its accounts. Please find below Sample check.

**account:** ["0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd","0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd","0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd","0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd","0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd","0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd","0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd","0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd","0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd","0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd"]

**ids:-** ["1","2","3","4","5","6","7","8","9","10"]

Graphical user interface, text, application

Description automatically generated

1. **SafeBatchTransferFrom:-**

This method we used to transfer more then one Tokenid from one account to other account. Please find below sample check.

**From:-** 0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd

**To:-** 0xC1F0369CC27E55AaE76578C389DD52BbBeF5B665

**Ids:-** ["1","2","3","4","5","6","7","8","9","10"]

**Amounts:-** ["1","1","1","1","1","1","1","1","1","1"]

**Data:-** 0x00

After feed data on on every row , click on **transact** button

Graphical user interface, text, application

Description automatically generated

After click on transact button, new metamask popup is open to confirm the transaction. Click on Confirm Button.

Graphical user interface, text, application, chat or text message

Description automatically generated

After confirm the transaction, Token Id **( 1,2,3,4,5,6,7,8,9,10)** is transferred from account **0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd**  to account **0xC1F0369CC27E55AaE76578C389DD52BbBeF5B665.**

Transaaction Hash is created.

**0x1932f98e5d883304a8b4e2799d8b2e4c57f5c910013975bac73c690e95c981c4**

Graphical user interface, text, application, email

Description automatically generated

Token Id **( 1,2,3,4,5,6,7,8,9,10)** is transferred from account **0xA1a5db6330dFcaCce4D3967f7Cb085098bE2d0Bd**  to account **0xC1F0369CC27E55AaE76578C389DD52BbBeF5B665.**

That’s why balanceofBatch method show **uint256[],0,0,0,0,0,0,0,0,0,0** because this tokenid has been transferred.

Graphical user interface, application

Description automatically generated

**Thanks**