import React, { useState, useEffect, useMemo } from 'react';

import './PendingNFTs.scss';

import Blockies from 'react-blockies';

import mp3Icon from '../../../../assets/images/mp3-icon.png';

import { useMyNftsContext } from '../../../../contexts/MyNFTsContext';

import { useAuthContext } from '../../../../contexts/AuthContext';

import { SpinningLoader } from '../misc/SpinningLoader';

import PendingAccordion from '../pendingAccordion/PendingAccordion';

import universeIcon from '../../../../assets/images/universe-img.svg';

​

const PendingNFTs = () => {

const { myMintingNFTs } = useMyNftsContext();

const { loggedInArtist, address } = useAuthContext();

​

const renderMintingNfts = useMemo(

() =>

myMintingNFTs.map((nft) => (

<div

onClick={() =>

window.open(`${process.env.REACT\_APP\_ETHERSCAN\_URL}/tx/${nft.txHash}`, '\_blank').focus()

}

className="nft\_\_card"

key={nft.id}

style={{ width: 160 }}

aria-hidden

>

<span className="tooltiptext">View on Etherscan</span>

<div className="nft\_\_card\_\_header">

<div className="three\_\_images">

{loggedInArtist &&

(loggedInArtist.avatar ||

(loggedInArtist.profileImageUrl && loggedInArtist.profileImageUrl.length > 48)) ? (

<div className="creator--details">

<img src={loggedInArtist.avatar} alt="first" />

</div>

) : (

<div className="creator--details">

<Blockies className="blockie--details" seed={address} size={9} scale={3} />

</div>

)}

​

<div className="collection--details">

{nft.collection.address ===

process.env.REACT\_APP\_UNIVERSE\_ERC\_721\_ADDRESS.toLowerCase() ? (

<img src={universeIcon} alt={nft.collection.name} />

) : (

<img src={nft.collection?.coverUrl} alt="second" />

)}

</div>

​

{loggedInArtist &&

(loggedInArtist.avatar ||

(loggedInArtist.profileImageUrl && loggedInArtist.profileImageUrl.length > 48)) ? (

<div className="owner--details">

<img src={loggedInArtist?.avatar} alt="last" />

</div>

) : (

<div className="owner--details">

<Blockies className="blockie--details" seed={address} size={9} scale={3} />

</div>

)}

</div>

<p className="nfts\_\_qantity">{`${nft.numberOfEditions}/${nft.numberOfEditions}`}</p>

</div>

<div className="nft\_\_card\_\_body">

<div className="loading-image">

<div className="image\_\_bg\_\_effect" />

{nft.artworkType &&

!nft.artworkType.endsWith('mpeg') &&

!nft.artworkType.endsWith('mp4') && <img src={nft.optimizedUrl} alt={nft.name} />}

{nft.artworkType && nft.artworkType.endsWith('mp4') && (

<video>

<source src={nft.optimizedUrl} type="video/mp4" />

<track kind="captions" />

Your browser does not support the video tag.

</video>

)}

{nft.artworkType && nft.artworkType.endsWith('mpeg') && (

<img className="nft--image" src={mp3Icon} alt={nft.name} />

)}

<SpinningLoader />

</div>

</div>

<div className="nft\_\_card\_\_footer">

<h1 className="nft\_\_name">{nft.name}</h1>

</div>

</div>

)),

[myMintingNFTs.length]

);

​

return myMintingNFTs.length ? (

<PendingAccordion title="Pending NFTs" dataLength={myMintingNFTs.length}>

{renderMintingNfts}

</PendingAccordion>

) : (

<></>

);

};

​

export default PendingNFTs;