The Making of Timeless NFT

## Project Setup

### Prerequisite

1. NodeJs
2. Yarn
3. Truffle
4. Web3Js
5. Ganache
6. Metamask
7. Git Bash

### **STEP 1**

Install truffle globally on your local machine using the command below.

| # Windows users open your command prompt as an admin and run: npm install -g truffle   # Mac users: sudo npm install -g truffle |
| --- |

### **STEP 2**

Install ganache globally on your local machine using the command below.

| # Windows users open your command prompt as an admin and run: npm install -g ganache-cli  # Mac users: sudo npm install -g ganache-cli |
| --- |

### **STEP 3**

Using the code below, clone this web3 starter project to your preferred working directory.

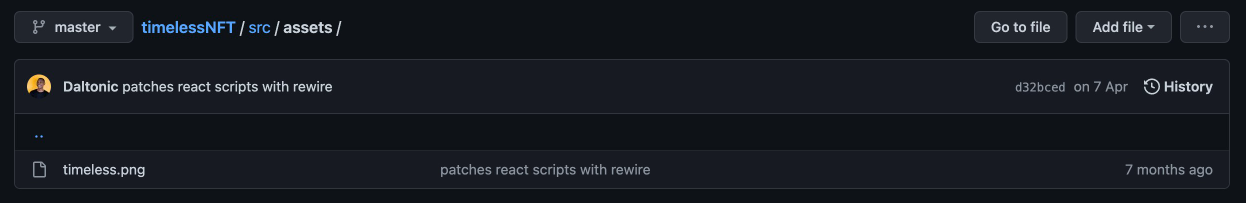
| git clone https://github.com/Daltonic/tailwind\_truffle\_starter\_kit timelessNFT |
| --- |

Next run `cd timelessNFT` to navigate to that project.

Lastly, run `code .` on the terminal to open it on vs code or simply drag it into your code editor.

### **STEP 4**

Head to this [**page**](https://github.com/Daltonic/timelessNFT/tree/master/src/assets) and download the images to the `**src/assets**` folder as specified on the image below.



### **STEP 5**

Open your `truffle-config.js` file and update the rinkeby network key to goerli, see the example below.

| goerli: {  provider: () =>  new HDWalletProvider(process.env.SECRET\_KEY, process.env.ENDPOINT\_URL),  network\_id: 4, // Rinkeby's id  gas: 5500000, // Rinkeby has a lower block limit than mainnet  confirmations: 2, // # of confs to wait between deployments. (default: 0)  timeoutBlocks: 200, // # of blocks before a deployment times out (minimum/default: 50)  skipDryRun: true, // Skip dry run before migrations? (default: false for public nets )  } |
| --- |

### **STEP 6**

Run the command below on your terminal to have the project dependencies of the starter kit installed on the project.

| yarn install |
| --- |

After the installation, run `yarn start` on your terminal to validate that the project is running rightly. You should see the output on your browser if done correctly.



## Updating Global CSS

Head to `Index.jsx` file and paste the codes below inside and save.

| @import url('https://fonts.googleapis.com/css2?family=Open+Sans:wght@300;400;500;600;700&display=swap');  \* html {  padding: 0;  margin: 0;  box-sizing: border-box; }  body {  margin: 0;  font-family: 'Open Sans', sans-serif;  -webkit-font-smoothing: antialiased;  -moz-osx-font-smoothing: grayscale; }  .gradient-bg-hero {  background-color: #131835;  background-image: radial-gradient(  at 0% 0%,  hsl(231deg 47% 14%) 0%,  transparent 50%  ),  radial-gradient(at 50% 0%, hsl(333deg 85% 53%) 0, transparent 50%),  radial-gradient(at 100% 0%, hsla(339, 49%, 30%, 1) 0, transparent 50%); }  .gradient-bg-artworks {  background-color: #0f0e13;  background-image: radial-gradient(  at 50% 50%,  hsl(302deg 25% 18%) 0,  transparent 50%  ),  radial-gradient(at 0% 0%, hsla(253, 16%, 7%, 1) 0, transparent 50%),  radial-gradient(at 50% 50%, hsla(339, 39%, 25%, 1) 0, transparent 50%); }  .gradient-bg-footer {  background-color: #131835;  background-image: radial-gradient(  at 20% 100%,  hsl(333deg 85% 53%) 0,  transparent 40%  ),  radial-gradient(at 50% 120%, hsla(339, 49%, 30%, 1) 0, transparent 40%); }  .text-gradient {  background: -webkit-linear-gradient(#eee, #333);  -webkit-background-clip: text;  -webkit-text-fill-color: transparent; }  .label-gradient {  background: rgb(19, 24, 53);  background: linear-gradient(  31deg,  rgba(19, 24, 53, 1) 0%,  rgba(237, 33, 124, 0) 100%  ); }  .no-scrollbar::-webkit-scrollbar {  display: none; }  .lds-dual-ring {  display: inline-block; } .lds-dual-ring:after {  content: ' ';  display: block;  width: 64px;  height: 64px;  margin: 8px;  border-radius: 50%;  border: 6px solid #fff;  border-color: #fff transparent #fff transparent;  animation: lds-dual-ring 1.2s linear infinite; } @keyframes lds-dual-ring {  0% {  transform: rotate(0deg);  }  100% {  transform: rotate(360deg);  } }  @tailwind base; @tailwind components; @tailwind utilities; |
| --- |

## Frontend Design

### **Header component**

Use the codes below to create the design for the Header component.

| import timelessLogo from '../assets/timeless.png'  const Header = () => {   return (  <nav className="w-4/5 flex md:justify-center justify-between items-center py-4 mx-auto">  <div className="md:flex-[0.5] flex-initial justify-center items-center">  <img  className="w-32 cursor-pointer"  src={timelessLogo}  alt="Timeless Logo"  />  </div>   <ul  className="md:flex-[0.5] text-white md:flex  hidden list-none flex-row justify-between  items-center flex-initial"  >  <li className="mx-4 cursor-pointer">Market</li>  <li className="mx-4 cursor-pointer">Artist</li>  <li className="mx-4 cursor-pointer">Features</li>  <li className="mx-4 cursor-pointer">Community</li>  </ul>   <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f] md:text-xs p-2  rounded-full cursor-pointer"  >  Connect Wallet  </button>  </nav>  ) }  export default Header |
| --- |

#### **Import Header component into the App.jsx file**

| import Header from "./components/Header"  const App = () => {  return (  <div className="min-h-screen">  <div className="gradient-bg-hero">  <Header />  </div>  </div>  ) }  export default App |
| --- |

### 

### **Hero component**

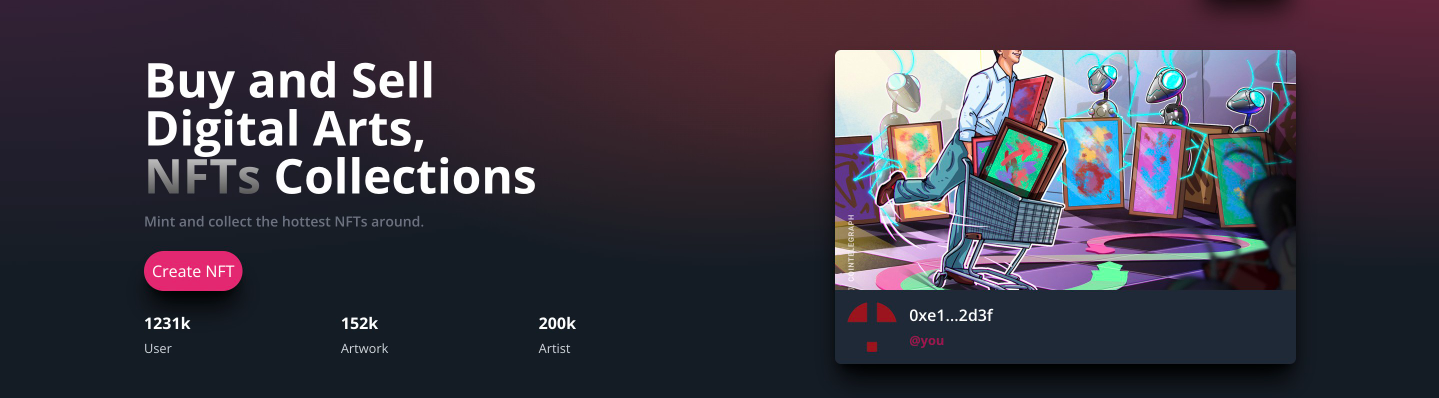
Use the codes below to create the design for the Hero component.

| **import Identicon from 'react-identicons'  const Hero = () => {   return (  <div  className="flex flex-col md:flex-row w-4/5 justify-between  items-center mx-auto py-10"  >  <div className="md:w-3/6 w-full">  <div>  <h1 className="text-white text-5xl font-bold">  Buy and Sell <br /> Digital Arts, <br />  <span className="text-gradient">NFTs</span> Collections  </h1>  <p className="text-gray-500 font-semibold text-sm mt-3">  Mint and collect the hottest NFTs around.  </p>  </div>   <div className="flex flex-row mt-5">  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f]  rounded-full cursor-pointer p-2"  >  Create NFT  </button>  </div>   <div className="w-3/4 flex justify-between items-center mt-5">  <div>  <p className="text-white font-bold">1231k</p>  <small className="text-gray-300">User</small>  </div>  <div>  <p className="text-white font-bold">152k</p>  <small className="text-gray-300">Artwork</small>  </div>  <div>  <p className="text-white font-bold">200k</p>  <small className="text-gray-300">Artist</small>  </div>  </div>  </div>   <div  className="shadow-xl shadow-black md:w-2/5 w-full  mt-10 md:mt-0 rounded-md overflow-hidden bg-gray-800"  >  <img  src="https://images.cointelegraph.com/images/1434\_aHR0cHM6Ly9zMy5jb2ludGVsZWdyYXBoLmNvbS91cGxvYWRzLzIwMjEtMDYvNGE4NmNmOWQtODM2Mi00YmVhLThiMzctZDEyODAxNjUxZTE1LmpwZWc=.jpg"  alt="NFT Art"  className="h-60 w-full object-cover"  />  <div className="flex justify-start items-center p-3">  <Identicon  string={'gbgdfjmghj'}  size={50}  className="h-10 w-10 object-contain rounded-full mr-3"  />  <div>  <p className="text-white font-semibold">0xe1...2d3f</p>  <small className="text-pink-800 font-bold">@you</small>  </div>  </div>  </div>  </div>  ) }  export default Hero** |
| --- |

#### **Import Hero component into the App.jsx file**

| import Header from "./components/Header" import Hero from "./components/Hero"  const App = () => {  return (  <div className="min-h-screen">  <div className="gradient-bg-hero">  <Header />  <Hero />  </div>  </div>  ) } export default App |
| --- |

See the image below.



### **Artworks component**

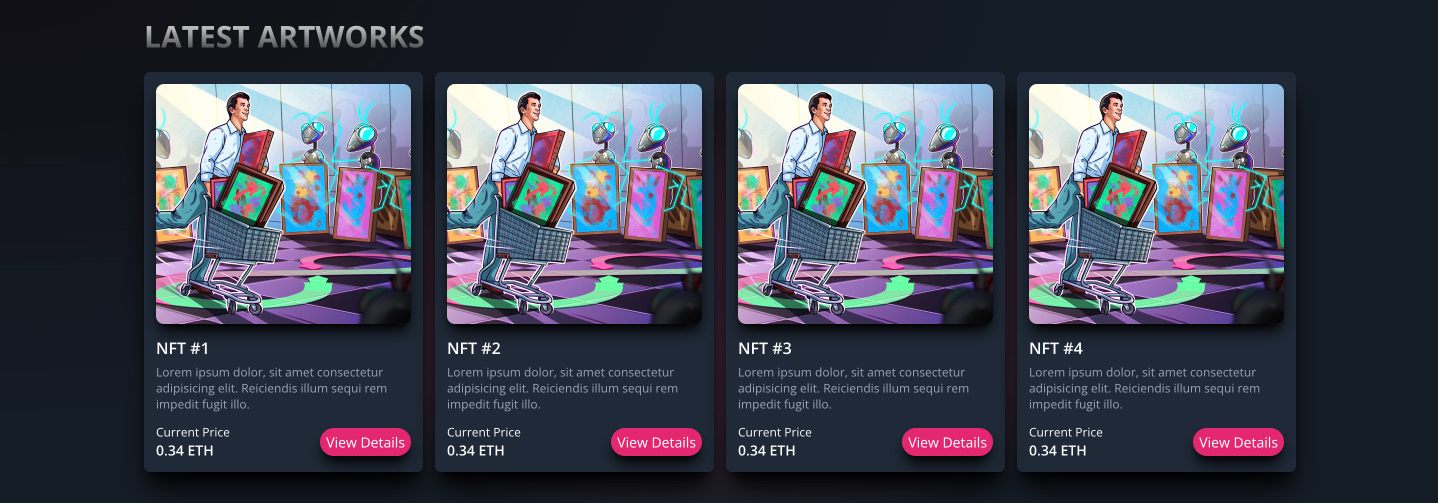
Use the codes below to create the design for the Artworks component.

| const Artworks = () => {  return (  <div className="bg-[#151c25] gradient-bg-artworks">  <div className="w-4/5 py-10 mx-auto">  <h4 className="text-white text-3xl font-bold uppercase text-gradient">  Latest Artworks  </h4>   <div className="grid grid-cols-1 md:grid-cols-3 lg:grid-cols-4 gap-6 md:gap-4 lg:gap-3 py-2.5">  {Array(4)  .fill()  .map((nft, i) => (  <Card key={i + 1} nft={i + 1} />  ))}  </div>  <div className="text-center my-5">  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f]  rounded-full cursor-pointer p-2"  >  Load More  </button>  </div>  </div>  </div>  ) }  const Card = ({ nft }) => (  <div className="w-full shadow-xl shadow-black rounded-md overflow-hidden bg-gray-800 my-2 p-3">  <img  src="https://images.cointelegraph.com/images/1434\_aHR0cHM6Ly9zMy5jb2ludGVsZWdyYXBoLmNvbS91cGxvYWRzLzIwMjEtMDYvNGE4NmNmOWQtODM2Mi00YmVhLThiMzctZDEyODAxNjUxZTE1LmpwZWc=.jpg"  alt={'Artwork'}  className="h-60 w-full object-cover shadow-lg shadow-black rounded-lg mb-3"  />  <h4 className="text-white font-semibold">NFT #{nft}</h4>  <p className="text-gray-400 text-xs my-1">  Lorem ipsum dolor, sit amet consectetur adipisicing elit. Reiciendis illum  sequi rem impedit fugit illo.  </p>  <div className="flex justify-between items-center mt-3 text-white">  <div className="flex flex-col">  <small className="text-xs">Current Price</small>  <p className="text-sm font-semibold">0.34 ETH</p>  </div>   <button  className="shadow-lg shadow-black text-white text-sm bg-[#e32970]  hover:bg-[#bd255f] cursor-pointer rounded-full px-1.5 py-1"  >  View Details  </button>  </div>  </div> )  export default Artworks |
| --- |

#### **Import Artworks component into the App.jsx file**

| import Artworks from "./components/Artworks" import Header from "./components/Header" import Hero from "./components/Hero"  const App = () => {  return (  <div className="min-h-screen">  <div className="gradient-bg-hero">  <Header />  <Hero />  </div>  <Artworks />  </div>  ) }  export default App |
| --- |

See the image below.



### **Transactions component**

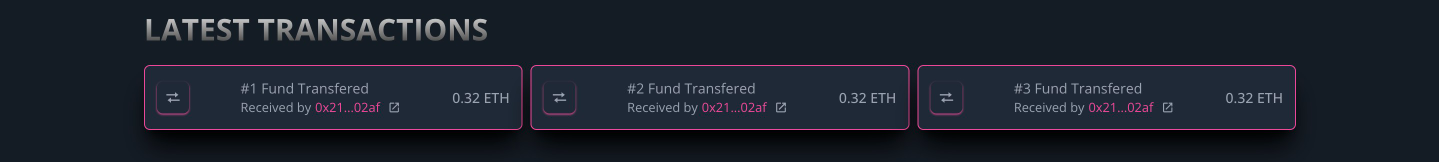
Use the codes below to create the design for the Transactions component.

| import { BiTransfer } from 'react-icons/bi' import { MdOpenInNew } from 'react-icons/md'  const Transactions = () => {  return (  <div className="bg-[#151c25]">  <div className="w-4/5 py-10 mx-auto">  <h4 className="text-white text-3xl font-bold uppercase text-gradient">  Latest Transactions  </h4>   <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6 md:gap-4 lg:gap-2 py-2.5">  {Array(3)  .fill()  .map((tx, i) => (  <Transaction key={i} tx={i + 1} />  ))}  </div>   <div className="text-center my-5">  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f]  rounded-full cursor-pointer p-2"  >  Load More  </button>  </div>  </div>  </div>  ) }  const Transaction = ({ tx }) => (  <div className="flex justify-between items-center border border-pink-500 text-gray-400 w-full shadow-xl shadow-black rounded-md overflow-hidden bg-gray-800 my-2 p-3">  <div className="rounded-md shadow-sm shadow-pink-500 p-2">  <BiTransfer />  </div>   <div>  <h4 className="text-sm">#{tx} Fund Transfered</h4>  <small className="flex flex-row justify-start items-center">  <span className="mr-1">Received by</span>  <a href="#" className="text-pink-500 mr-2">  0x21...02af  </a>  <a href="#">  <MdOpenInNew />  </a>  </small>  </div>   <p className="text-sm font-medium">0.32 ETH</p>  </div> )  export default Transactions |
| --- |

#### **Import Transactions component into the App.jsx file**

| import Artworks from "./components/Artworks" import Header from "./components/Header" import Hero from "./components/Hero" import Transactions from "./components/Transactions"  const App = () => {  return (  <div className="min-h-screen">  <div className="gradient-bg-hero">  <Header />  <Hero />  </div>  <Artworks />  <Transactions />  </div>  ) }  export default App |
| --- |

See the image below.



### **Store component**

Set up a state management store to handle the create nft modal. Create a new folder inside the src directory and call it **store**. Inside create a new file and call it **index.js**, then paste and save the codes below inside.

| import { createGlobalState } from 'react-hooks-global-state'  const { setGlobalState, useGlobalState, getGlobalState } = createGlobalState({  modal: 'scale-0', })  export { useGlobalState, setGlobalState, getGlobalState } |
| --- |

### **Create NFT component**

Use the codes below to create the design for the Create NFT component.

| import { useGlobalState, setGlobalState } from '../store' import { useState } from 'react' import { FaTimes } from 'react-icons/fa'  const CreateNFT = () => {  const [modal] = useGlobalState('modal')  const [title, setTitle] = useState('')  const [price, setPrice] = useState('')  const [description, setDescription] = useState('')  const [fileUrl, setFileUrl] = useState('')  const [imgBase64, setImgBase64] = useState(null)   const handleSubmit = async (e) => {  e.preventDefault()   if (!title || !price || !description) return   setGlobalState('modal', 'scale-0')  }   const closeModal = () => {  setGlobalState('modal', 'scale-0')  resetForm()  }   const resetForm = () => {  setFileUrl('')  setImgBase64(null)  setTitle('')  setPrice('')  setDescription('')  }   return (  <div  className={`fixed top-0 left-0 w-screen h-screen flex items-center  justify-center bg-black bg-opacity-50 transform  transition-transform duration-300 ${modal}`}  >  <div className="bg-[#151c25] shadow-xl shadow-[#e32970] rounded-xl w-11/12 md:w-2/5 h-7/12 p-6">  <form className="flex flex-col">  <div className="flex flex-row justify-between items-center">  <p className="font-semibold text-gray-400">Add NFT</p>  <button  type="button"  onClick={closeModal}  className="border-0 bg-transparent focus:outline-none"  >  <FaTimes className="text-gray-400" />  </button>  </div>   <div className="flex flex-row justify-center items-center rounded-xl mt-5">  <div className="shrink-0 rounded-xl overflow-hidden h-20 w-20">  <img  alt="NFT"  className="h-full w-full object-cover cursor-pointer"  src={  imgBase64 ||  'https://images.unsplash.com/photo-1580489944761-15a19d654956?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1361&q=80'  }  />  </div>  </div>   <div className="flex flex-row justify-between items-center bg-gray-800 rounded-xl mt-5">  <label className="block">  <span className="sr-only">Choose profile photo</span>  <input  type="file"  accept="image/png, image/gif, image/jpeg, image/webp"  className="block w-full text-sm text-slate-500  file:mr-4 file:py-2 file:px-4  file:rounded-full file:border-0  file:text-sm file:font-semibold  file:bg-[#19212c] file:text-gray-400  hover:file:bg-[#1d2631]  cursor-pointer focus:ring-0 focus:outline-none"  required  />  </label>  </div>   <div className="flex flex-row justify-between items-center bg-gray-800 rounded-xl mt-5">  <input  className="block w-full text-sm  text-slate-500 bg-transparent border-0  focus:outline-none focus:ring-0"  type="text"  name="title"  placeholder="Title"  onChange={(e) => setTitle(e.target.value)}  value={title}  required  />  </div>   <div className="flex flex-row justify-between items-center bg-gray-800 rounded-xl mt-5">  <input  className="block w-full text-sm  text-slate-500 bg-transparent border-0  focus:outline-none focus:ring-0"  type="number"  step={0.01}  min={0.01}  name="price"  placeholder="Price (Eth)"  onChange={(e) => setPrice(e.target.value)}  value={price}  required  />  </div>   <div className="flex flex-row justify-between items-center bg-gray-800 rounded-xl mt-5">  <textarea  className="block w-full text-sm resize-none  text-slate-500 bg-transparent border-0  focus:outline-none focus:ring-0 h-20"  type="text"  name="description"  placeholder="Description"  onChange={(e) => setDescription(e.target.value)}  value={description}  required  ></textarea>  </div>   <button  type="submit"  onClick={handleSubmit}  className="flex flex-row justify-center items-center  w-full text-white text-md bg-[#e32970]  hover:bg-[#bd255f] py-2 px-5 rounded-full  drop-shadow-xl border border-transparent  hover:bg-transparent hover:text-[#e32970]  hover:border hover:border-[#bd255f]  focus:outline-none focus:ring mt-5"  >  Mint Now  </button>  </form>  </div>  </div>  ) }  export default CreateNFT |
| --- |

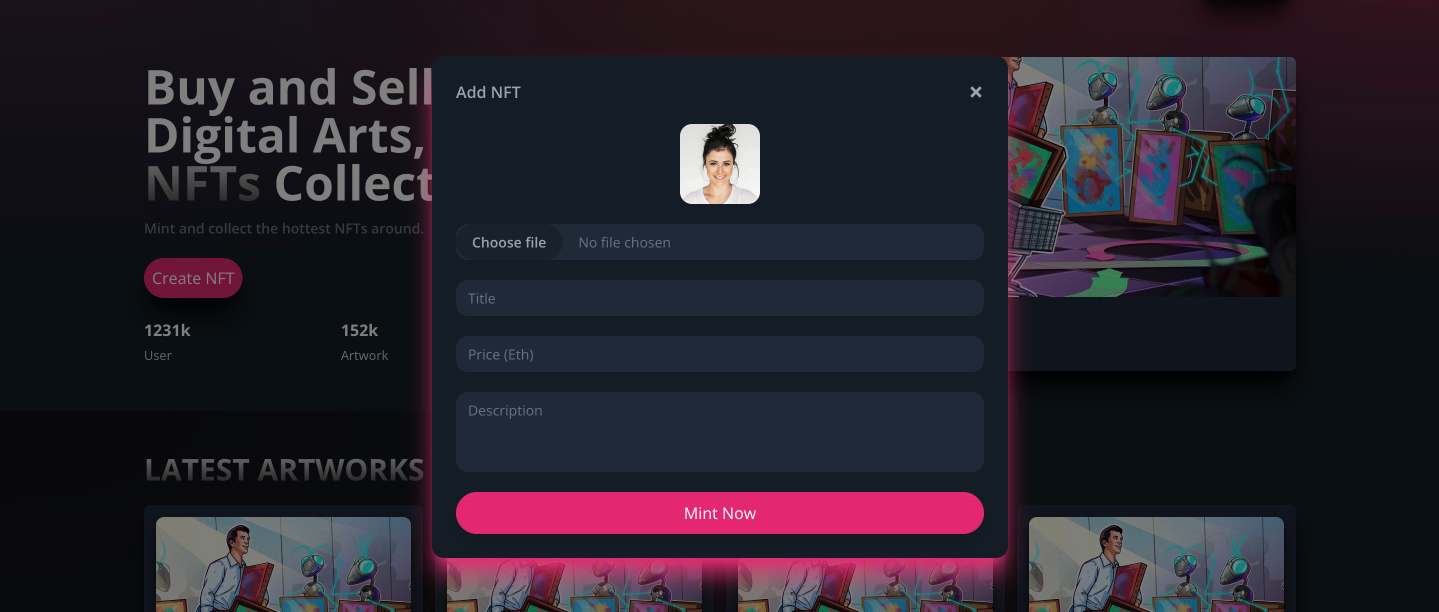
#### **Import Create NFT component into the App.jsx file**

| import Artworks from "./components/Artworks" import CreateNFT from "./components/CreateNFT" import Header from "./components/Header" import Hero from "./components/Hero" import Transactions from "./components/Transactions"  const App = () => {  return (  <div className="min-h-screen">  <div className="gradient-bg-hero">  <Header />  <Hero />  </div>  <Artworks />  <Transactions />  <CreateNFT />  </div>  ) }  export default App |
| --- |

#### **Updating the Hero component**

| import Identicon from 'react-identicons' import { setGlobalState } from '../store'  const Hero = () => {  const onCreatedNFT = () => {  setGlobalState('modal', 'scale-100')  }   return (  <div  className="flex flex-col md:flex-row w-4/5 justify-between  items-center mx-auto py-10"  >  <div className="md:w-3/6 w-full">  <div>  <h1 className="text-white text-5xl font-bold">  Buy and Sell <br /> Digital Arts, <br />  <span className="text-gradient">NFTs</span> Collections  </h1>  <p className="text-gray-500 font-semibold text-sm mt-3">  Mint and collect the hottest NFTs around.  </p>  </div>   <div className="flex flex-row mt-5">  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f]  rounded-full cursor-pointer p-2"  onClick={onCreatedNFT}  >  Create NFT  </button>  </div>   <div className="w-3/4 flex justify-between items-center mt-5">  <div>  <p className="text-white font-bold">1231k</p>  <small className="text-gray-300">User</small>  </div>  <div>  <p className="text-white font-bold">152k</p>  <small className="text-gray-300">Artwork</small>  </div>  <div>  <p className="text-white font-bold">200k</p>  <small className="text-gray-300">Artist</small>  </div>  </div>  </div>   <div  className="shadow-xl shadow-black md:w-2/5 w-full  mt-10 md:mt-0 rounded-md overflow-hidden bg-gray-800"  >  <img  src="https://images.cointelegraph.com/images/1434\_aHR0cHM6Ly9zMy5jb2ludGVsZWdyYXBoLmNvbS91cGxvYWRzLzIwMjEtMDYvNGE4NmNmOWQtODM2Mi00YmVhLThiMzctZDEyODAxNjUxZTE1LmpwZWc=.jpg"  alt="NFT Art"  className="h-60 w-full object-cover"  />  <div className="flex justify-start items-center p-3">  <Identicon  string={'gbgdfjmghj'}  size={50}  className="h-10 w-10 object-contain rounded-full mr-3"  />  <div>  <p className="text-white font-semibold">0xe1...2d3f</p>  <small className="text-pink-800 font-bold">@you</small>  </div>  </div>  </div>  </div>  ) }  export default Hero |
| --- |

See the component below.



### **Show NFT component**

Use the codes below to create the design for the Show NFT component.

| import Identicon from 'react-identicons' import { FaTimes } from 'react-icons/fa' import { useGlobalState, setGlobalState } from '../store'  const ShowNFT = () => {  const [showModal] = useGlobalState('showModal')   const handleNFTPurchase = () => {  setGlobalState('showModal', 'scale-0')  }   return (  <div  className={`fixed top-0 left-0 w-screen h-screen flex items-center  justify-center bg-black bg-opacity-50 transform  transition-transform duration-300 ${showModal}`}  >  <div className="bg-[#151c25] shadow-xl shadow-[#e32970] rounded-xl w-11/12 md:w-2/5 h-7/12 p-6">  <div className="flex flex-col">  <div className="flex flex-row justify-between items-center">  <p className="font-semibold text-gray-400">Buy NFT</p>  <button  type="button"  onClick={() => setGlobalState('showModal', 'scale-0')}  className="border-0 bg-transparent focus:outline-none"  >  <FaTimes className="text-gray-400" />  </button>  </div>   <div className="flex flex-row justify-center items-center rounded-xl mt-5">  <div className="shrink-0 rounded-xl overflow-hidden h-40 w-40">  <img  className="h-full w-full object-cover cursor-pointer"  src="https://images.cointelegraph.com/images/1434\_aHR0cHM6Ly9zMy5jb2ludGVsZWdyYXBoLmNvbS91cGxvYWRzLzIwMjEtMDYvNGE4NmNmOWQtODM2Mi00YmVhLThiMzctZDEyODAxNjUxZTE1LmpwZWc=.jpg"  alt="Title"  />  </div>  </div>   <div className="flex flex-col justify-start rounded-xl mt-5">  <h4 className="text-white font-semibold">Title</h4>  <p className="text-gray-400 text-xs my-1">  Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quam  tempora laborum quod.  </p>   <div className="flex justify-between items-center mt-3 text-white">  <div className="flex justify-start items-center">  <Identicon  string={'ugjkhjlk'}  size={50}  className="h-10 w-10 object-contain rounded-full mr-3"  />  <div className="flex flex-col justify-center items-start">  <small className="text-white font-bold">@owner</small>  <small className="text-pink-800 font-semibold">  0xe2...a1f6  </small>  </div>  </div>   <div className="flex flex-col">  <small className="text-xs">Current Price</small>  <p className="text-sm font-semibold">0.34 ETH</p>  </div>  </div>  </div>  <button  className="flex flex-row justify-center items-center  w-full text-white text-md bg-[#e32970]  hover:bg-[#bd255f] py-2 px-5 rounded-full  drop-shadow-xl border border-transparent  hover:bg-transparent hover:text-[#e32970]  hover:border hover:border-[#bd255f]  focus:outline-none focus:ring mt-5"  onClick={handleNFTPurchase}  >  Purchase Now  </button>  {/\* <button  className="flex flex-row justify-center items-center  w-full text-white text-md bg-[#e32970]  hover:bg-[#bd255f] py-2 px-5 rounded-full  drop-shadow-xl border border-transparent  hover:bg-transparent hover:text-[#e32970]  hover:border hover:border-[#bd255f]  focus:outline-none focus:ring mt-5"  >  Change Price  </button> \*/}  </div>  </div>  </div>  ) }  export default ShowNFT |
| --- |

#### **Import Show NFT component into the App.jsx file**

| import Artworks from "./components/Artworks" import CreateNFT from "./components/CreateNFT" import Header from "./components/Header" import Hero from "./components/Hero" import ShowNFT from "./components/ShowNFT" import Transactions from "./components/Transactions"  const App = () => {  return (  <div className="min-h-screen">  <div className="gradient-bg-hero">  <Header />  <Hero />  </div>  <Artworks />  <Transactions />  <CreateNFT />  <ShowNFT />  </div>  ) }  export default App |
| --- |

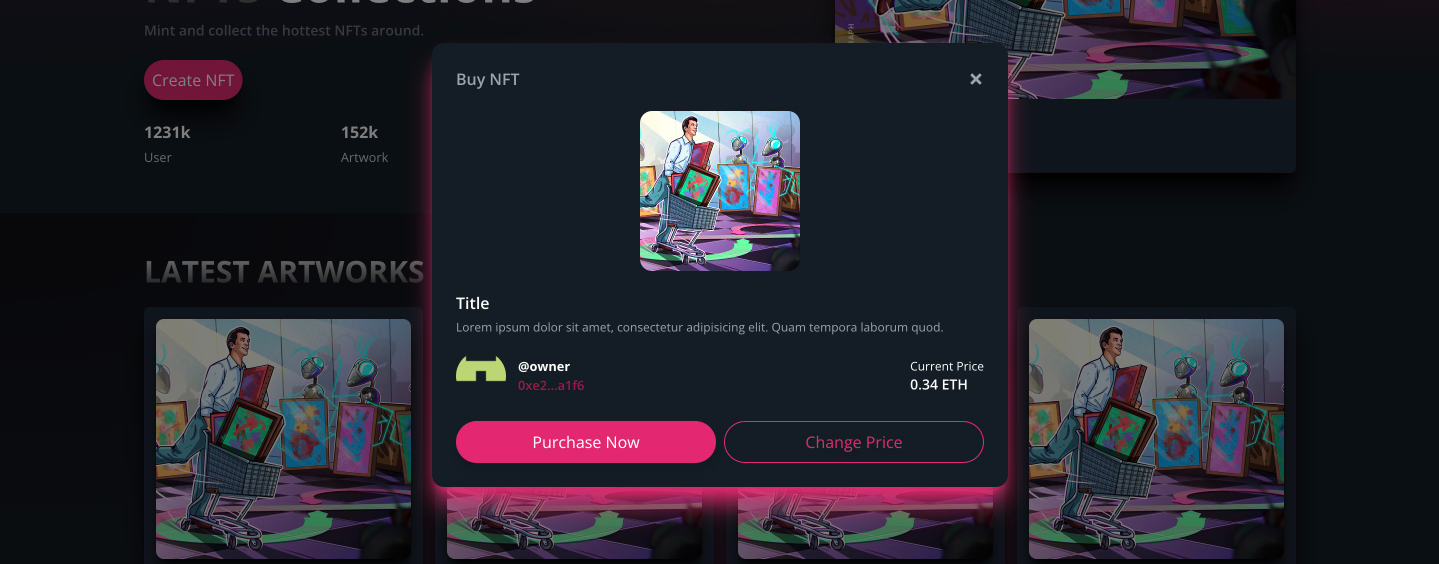
#### **Updating the Store file**

| import { createGlobalState } from 'react-hooks-global-state'  const { setGlobalState, useGlobalState, getGlobalState } = createGlobalState({  modal: 'scale-0',  showModal: 'scale-0', })  export { useGlobalState, setGlobalState, getGlobalState } |
| --- |

#### **Updating the Artworks Card Component file**

| import { setGlobalState } from "../store"  const Artworks = () => {  return (  <div className="bg-[#151c25] gradient-bg-artworks">  <div className="w-4/5 py-10 mx-auto">  <h4 className="text-white text-3xl font-bold uppercase text-gradient">  Latest Artworks  </h4>   <div className="grid grid-cols-1 md:grid-cols-3 lg:grid-cols-4 gap-6 md:gap-4 lg:gap-3 py-2.5">  {Array(4)  .fill()  .map((nft, i) => (  <Card key={i + 1} nft={i + 1} />  ))}  </div>  <div className="text-center my-5">  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f]  rounded-full cursor-pointer p-2"  >  Load More  </button>  </div>  </div>  </div>  ) }  const Card = ({ nft }) => (  <div className="w-full shadow-xl shadow-black rounded-md overflow-hidden bg-gray-800 my-2 p-3">  <img  src="https://images.cointelegraph.com/images/1434\_aHR0cHM6Ly9zMy5jb2ludGVsZWdyYXBoLmNvbS91cGxvYWRzLzIwMjEtMDYvNGE4NmNmOWQtODM2Mi00YmVhLThiMzctZDEyODAxNjUxZTE1LmpwZWc=.jpg"  alt={'Artwork'}  className="h-60 w-full object-cover shadow-lg shadow-black rounded-lg mb-3"  />  <h4 className="text-white font-semibold">NFT #{nft}</h4>  <p className="text-gray-400 text-xs my-1">  Lorem ipsum dolor, sit amet consectetur adipisicing elit. Reiciendis illum  sequi rem impedit fugit illo.  </p>  <div className="flex justify-between items-center mt-3 text-white">  <div className="flex flex-col">  <small className="text-xs">Current Price</small>  <p className="text-sm font-semibold">0.34 ETH</p>  </div>   <button  className="shadow-lg shadow-black text-white text-sm bg-[#e32970]  hover:bg-[#bd255f] cursor-pointer rounded-full px-1.5 py-1"  onClick={() => setGlobalState('showModal', 'scale-100')}  >  View Details  </button>  </div>  </div> )  export default Artworks |
| --- |

See how the component looks.



### **Update NFT component**

Use the codes below to create the design for the Update NFT component.

| import { useGlobalState, setGlobalState } from '../store' import { useState } from 'react' import { FaTimes } from 'react-icons/fa'  const UpdateNFT = () => {  const [modal] = useGlobalState('updateModal')  const [price, setPrice] = useState('')   const handleSubmit = async (e) => {  e.preventDefault()  if (!price || price <= 0) return   setGlobalState('modal', 'scale-0')  }   return (  <div  className={`fixed top-0 left-0 w-screen h-screen flex items-center  justify-center bg-black bg-opacity-50 transform  transition-transform duration-300 ${modal}`}  >  <div className="bg-[#151c25] shadow-xl shadow-[#e32970] rounded-xl w-11/12 md:w-2/5 h-7/12 p-6">  <form className="flex flex-col">  <div className="flex flex-row justify-between items-center">  <p className="font-semibold text-gray-400">Title</p>  <button  type="button"  onClick={() => setGlobalState('updateModal', 'scale-0')}  className="border-0 bg-transparent focus:outline-none"  >  <FaTimes className="text-gray-400" />  </button>  </div>   <div className="flex flex-row justify-center items-center rounded-xl mt-5">  <div className="shrink-0 rounded-xl overflow-hidden h-20 w-20">  <img  alt="NFT"  className="h-full w-full object-cover cursor-pointer"  src="https://images.cointelegraph.com/images/1434\_aHR0cHM6Ly9zMy5jb2ludGVsZWdyYXBoLmNvbS91cGxvYWRzLzIwMjEtMDYvNGE4NmNmOWQtODM2Mi00YmVhLThiMzctZDEyODAxNjUxZTE1LmpwZWc=.jpg"  />  </div>  </div>   <div className="flex flex-row justify-between items-center bg-gray-800 rounded-xl mt-5">  <input  className="block w-full text-sm  text-slate-500 bg-transparent border-0  focus:outline-none focus:ring-0"  type="number"  step={0.01}  min={0.01}  name="price"  placeholder="Price (Eth)"  onChange={(e) => setPrice(e.target.value)}  required  />  </div>   <button  type="submit"  onClick={handleSubmit}  className="flex flex-row justify-center items-center  w-full text-white text-md bg-[#e32970]  hover:bg-[#bd255f] py-2 px-5 rounded-full  drop-shadow-xl border border-transparent  hover:bg-transparent hover:text-[#e32970]  hover:border hover:border-[#bd255f]  focus:outline-none focus:ring mt-5"  >  Update Now  </button>  </form>  </div>  </div>  ) }  export default UpdateNFT |
| --- |

#### **Import Update NFT component into the App.jsx file**

| import Artworks from "./components/Artworks" import CreateNFT from "./components/CreateNFT" import Header from "./components/Header" import Hero from "./components/Hero" import ShowNFT from "./components/ShowNFT" import Transactions from "./components/Transactions" import UpdateNFT from "./components/UpdateNFT"  const App = () => {  return (  <div className="min-h-screen">  <div className="gradient-bg-hero">  <Header />  <Hero />  </div>  <Artworks />  <Transactions />  <CreateNFT />  <ShowNFT />  <UpdateNFT />  </div>  ) }  export default App |
| --- |

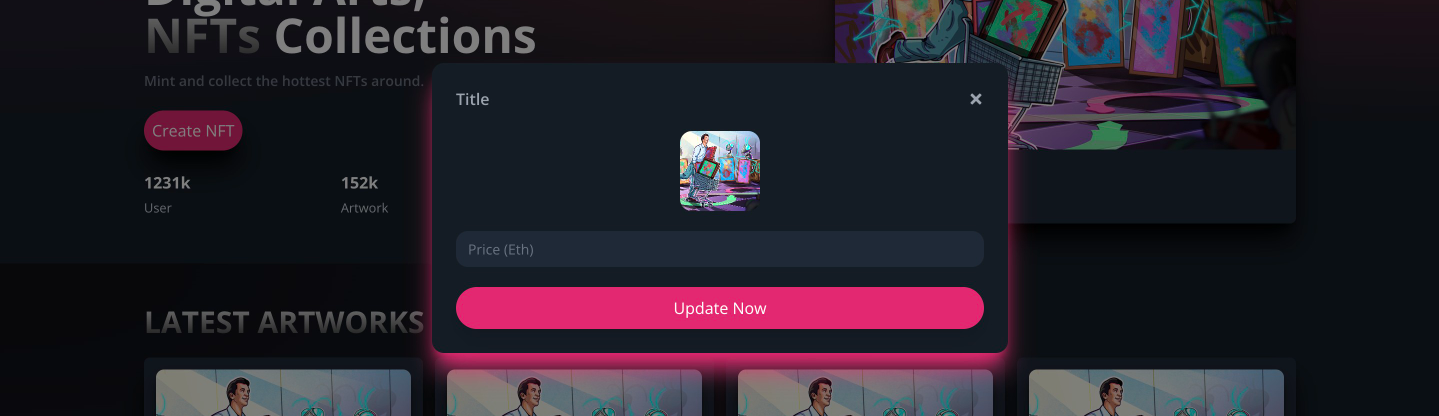
#### **Updating the Store file**

| import { createGlobalState } from 'react-hooks-global-state'  const { setGlobalState, useGlobalState, getGlobalState } = createGlobalState({  modal: 'scale-0',  updateModal: 'scale-0',  showModal: 'scale-0', })  export { useGlobalState, setGlobalState, getGlobalState } |
| --- |

#### **Updating the Show NFT Component file**

| import Identicon from 'react-identicons' import { FaTimes } from 'react-icons/fa' import { useGlobalState, setGlobalState } from '../store'  const ShowNFT = () => {  const [showModal] = useGlobalState('showModal')   const onChangePrice = () => {  setGlobalState('showModal', 'scale-0')  setGlobalState('updateModal', 'scale-100')  }   const handleNFTPurchase = () => {  setGlobalState('showModal', 'scale-0')  }   return (  <div  className={`fixed top-0 left-0 w-screen h-screen flex items-center  justify-center bg-black bg-opacity-50 transform  transition-transform duration-300 ${showModal}`}  >  <div className="bg-[#151c25] shadow-xl shadow-[#e32970] rounded-xl w-11/12 md:w-2/5 h-7/12 p-6">  <div className="flex flex-col">  <div className="flex flex-row justify-between items-center">  <p className="font-semibold text-gray-400">Buy NFT</p>  <button  type="button"  onClick={() => setGlobalState('showModal', 'scale-0')}  className="border-0 bg-transparent focus:outline-none"  >  <FaTimes className="text-gray-400" />  </button>  </div>   <div className="flex flex-row justify-center items-center rounded-xl mt-5">  <div className="shrink-0 rounded-xl overflow-hidden h-40 w-40">  <img  className="h-full w-full object-cover cursor-pointer"  src="https://images.cointelegraph.com/images/1434\_aHR0cHM6Ly9zMy5jb2ludGVsZWdyYXBoLmNvbS91cGxvYWRzLzIwMjEtMDYvNGE4NmNmOWQtODM2Mi00YmVhLThiMzctZDEyODAxNjUxZTE1LmpwZWc=.jpg"  alt="Title"  />  </div>  </div>   <div className="flex flex-col justify-start rounded-xl mt-5">  <h4 className="text-white font-semibold">Title</h4>  <p className="text-gray-400 text-xs my-1">  Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quam  tempora laborum quod.  </p>   <div className="flex justify-between items-center mt-3 text-white">  <div className="flex justify-start items-center">  <Identicon  string={'ugjkhjlk'}  size={50}  className="h-10 w-10 object-contain rounded-full mr-3"  />  <div className="flex flex-col justify-center items-start">  <small className="text-white font-bold">@owner</small>  <small className="text-pink-800 font-semibold">  0xe2...a1f6  </small>  </div>  </div>   <div className="flex flex-col">  <small className="text-xs">Current Price</small>  <p className="text-sm font-semibold">0.34 ETH</p>  </div>  </div>  </div>  <div className='flex justify-between items-center space-x-2'>  <button  className="flex flex-row justify-center items-center  w-full text-white text-md bg-[#e32970]  hover:bg-[#bd255f] py-2 px-5 rounded-full  drop-shadow-xl border border-transparent  hover:bg-transparent hover:text-[#e32970]  hover:border hover:border-[#bd255f]  focus:outline-none focus:ring mt-5"  onClick={handleNFTPurchase}  >  Purchase Now  </button>  <button  className="flex flex-row justify-center items-center  w-full text-[#e32970] text-md border-[#e32970]  py-2 px-5 rounded-full bg-transparent  drop-shadow-xl border hover:bg-[#bd255f]  hover:bg-transparent hover:text-white  hover:border hover:border-[#bd255f]  focus:outline-none focus:ring mt-5"  onClick={onChangePrice}  >  Change Price  </button>  </div>  </div>  </div>  </div>  ) }  export default ShowNFT |
| --- |

See result below.



### **Footer component**

Use the codes below to create the design for the Footer component.

| import timelessLogo from '../assets/timeless.png'  const Footer = () => (  <div className="w-full flex md:justify-center justify-between items-center flex-col p-4 gradient-bg-footer">  <div className="w-full flex sm:flex-row flex-col justify-between items-center my-4">  <div className="flex flex-[0.25] justify-center items-center">  <img src={timelessLogo} alt="logo" className="w-32" />  </div>   <div className="flex flex-1 justify-evenly items-center flex-wrap sm:mt-0 mt-5 w-full">  <p className="text-white text-base text-center mx-2 cursor-pointer">  Market  </p>  <p className="text-white text-base text-center mx-2 cursor-pointer">  Artist  </p>  <p className="text-white text-base text-center mx-2 cursor-pointer">  Features  </p>  <p className="text-white text-base text-center mx-2 cursor-pointer">  Community  </p>  </div>   <div className="flex flex-[0.25] justify-center items-center">  <p className="text-white text-right text-xs">  &copy;2022 All rights reserved  </p>  </div>  </div>  </div> )  export default Footer |
| --- |

#### **Import Footer component into the App.jsx file**

| import Artworks from "./components/Artworks" import CreateNFT from "./components/CreateNFT" import Footer from "./components/Footer" import Header from "./components/Header" import Hero from "./components/Hero" import ShowNFT from "./components/ShowNFT" import Transactions from "./components/Transactions" import UpdateNFT from "./components/UpdateNFT"  const App = () => {  return (  <div className="min-h-screen">  <div className="gradient-bg-hero">  <Header />  <Hero />  </div>  <Artworks />  <Transactions />  <CreateNFT />  <ShowNFT />  <UpdateNFT />  <Footer />  </div>  ) }  export default App |
| --- |

### **Alert component**

Use the codes below to create the design for the Alert component.

| import { useGlobalState } from '../store' import { FaRegTimesCircle } from 'react-icons/fa' import { BsCheck2Circle } from 'react-icons/bs'  const Alert = () => {  const [alert] = useGlobalState('alert')   return (  <div  className={`fixed top-0 left-0 w-screen h-screen  flex items-center justify-center bg-black  bg-opacity-50 transform transition-transform  duration-300 ${alert.show ? 'scale-100' : 'scale-0'}`}  >  <div  className="flex flex-col justify-center items-center  bg-[#151c25] shadow-xl shadow-[#e32970] rounded-xl  min-w-min py-3 px-10"  >  {alert.color == 'red' ? (  <FaRegTimesCircle className="text-red-600 text-4xl" />  ) : (  <BsCheck2Circle className="text-green-600 text-4xl" />  )}  <p className="text-white my-3">{alert.msg}</p>  </div>  </div>  ) }  export default Alert |
| --- |

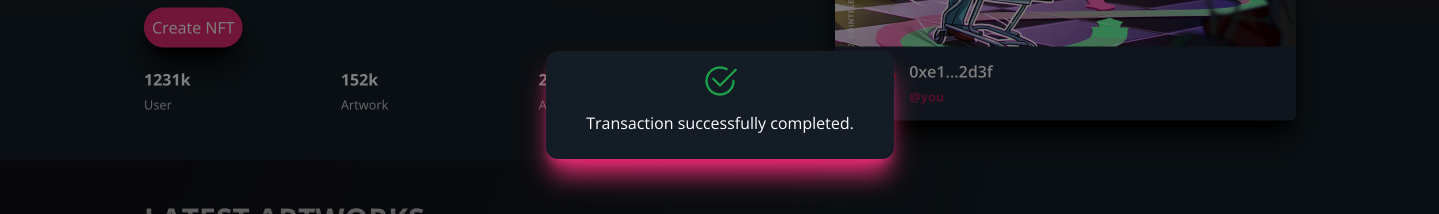
#### **Import Alert component into the App.jsx file**

| import Alert from "./components/Alert" import Artworks from "./components/Artworks" import CreateNFT from "./components/CreateNFT" import Footer from "./components/Footer" import Header from "./components/Header" import Hero from "./components/Hero" import ShowNFT from "./components/ShowNFT" import Transactions from "./components/Transactions" import UpdateNFT from "./components/UpdateNFT"  const App = () => {  return (  <div className="min-h-screen">  <div className="gradient-bg-hero">  <Header />  <Hero />  </div>  <Artworks />  <Transactions />  <CreateNFT />  <ShowNFT />  <UpdateNFT />  <Footer />  <Alert />  </div>  ) }  export default App |
| --- |

#### **Updating the Store file**

| import { createGlobalState } from 'react-hooks-global-state'  const { setGlobalState, useGlobalState, getGlobalState } = createGlobalState({  modal: 'scale-0',  updateModal: 'scale-0',  showModal: 'scale-0',  alert: { show: false, msg: '', color: '' }, })  const setAlert = (msg, color = 'green') => {  setGlobalState('loading', false)  setGlobalState('alert', { show: true, msg, color })  setTimeout(() => {  setGlobalState('alert', { show: false, msg: '', color })  }, 6000) }  export { useGlobalState, setGlobalState, getGlobalState, setAlert } |
| --- |

See the result.



### **Loading component**

Use the codes below to create the design for the Loading component.

| import { useGlobalState } from '../store'  const Loading = () => {  const [loading] = useGlobalState('loading')   return (  <div  className={`fixed top-0 left-0 w-screen h-screen  flex items-center justify-center bg-black  bg-opacity-50 transform transition-transform  duration-300 ${loading.show ? 'scale-100' : 'scale-0'}`}  >  <div  className="flex flex-col justify-center  items-center bg-[#151c25] shadow-xl  shadow-[#e32970] rounded-xl  min-w-min px-10 pb-2"  >  <div className="flex flex-row justify-center items-center">  <div className="lds-dual-ring scale-50"></div>  <p className="text-lg text-white">Processing...</p>  </div>  <small className="text-white">{loading.msg}</small>  </div>  </div>  ) }  export default Loading |
| --- |

#### **Updating the Store file**

| import { createGlobalState } from 'react-hooks-global-state'  const { setGlobalState, useGlobalState, getGlobalState } = createGlobalState({  modal: 'scale-0',  updateModal: 'scale-0',  showModal: 'scale-0',  alert: { show: false, msg: '', color: '' },  loading: { show: false, msg: '' }, })  const setAlert = (msg, color = 'green') => {  setGlobalState('loading', false)  setGlobalState('alert', { show: true, msg, color })  setTimeout(() => {  setGlobalState('alert', { show: false, msg: '', color })  }, 6000) }  const setLoadingMsg = (msg) => {  const loading = getGlobalState('loading')  setGlobalState('loading', { ...loading, msg }) }  export {  useGlobalState,  setGlobalState,  getGlobalState,  setAlert,  setLoadingMsg, } |
| --- |

#### **Import Loading component into the App.jsx file**

| import Alert from "./components/Alert" import Artworks from "./components/Artworks" import CreateNFT from "./components/CreateNFT" import Footer from "./components/Footer" import Header from "./components/Header" import Hero from "./components/Hero" import Loading from "./components/Loading" import ShowNFT from "./components/ShowNFT" import Transactions from "./components/Transactions" import UpdateNFT from "./components/UpdateNFT"  const App = () => {  return (  <div className="min-h-screen">  <div className="gradient-bg-hero">  <Header />  <Hero />  </div>  <Artworks />  <Transactions />  <CreateNFT />  <ShowNFT />  <UpdateNFT />  <Footer />  <Alert />  <Loading />  </div>  ) }  export default App |
| --- |

## Backend Logic

In the `/src` of the project, create a new folder called `/contracts` and then a file inside of it called `TimelessNFT.sol`. Now, paste the codes below inside of it using the steps below.

### **Import Smart Contract Dependencies**

Head to this [git repo](https://github.com/Daltonic/timelessNFT/tree/master/src/contracts), and download the smart contracts with the exception of TimelessNFT and create them in the contracts directory. See the image below.

### 

### **Smart Contract Structure**

| // SPDX-License-Identifier: MIT pragma solidity >=0.7.0 <0.9.0;  import "./ERC721.sol"; import "./ERC721Enumerable.sol"; import "@openzeppelin/contracts/access/Ownable.sol";  contract TimelessNFT is ERC721Enumerable, Ownable {  // code goes in here… } |
| --- |

### **Essential Variables**

| using Strings for uint256; mapping(string => uint8) existingURIs; mapping(uint256 => address) public holderOf;  address public artist; uint256 public royalityFee; uint256 public supply = 0; uint256 public totalTx = 0; uint256 public cost = 0.01 ether; |
| --- |

### **Defining Events and Structures**

| event Sale(  uint256 id,  address indexed owner,  uint256 cost,  string metadataURI,  uint256 timestamp );  struct TransactionStruct {  uint256 id;  address owner;  uint256 cost;  string title;  string description;  string metadataURI;  uint256 timestamp; }  TransactionStruct[] transactions; TransactionStruct[] minted; |
| --- |

### **Defining Constructor Parameter**

| constructor(  string memory \_name,  string memory \_symbol,  uint256 \_royalityFee,  address \_artist  ) ERC721(\_name, \_symbol) {  royalityFee = \_royalityFee;  artist = \_artist;  } |
| --- |

### **Paying to Mint Function**

| function payToMint(  string memory title,  string memory description,  string memory metadataURI,  uint256 salesPrice  ) external payable {  require(msg.value >= cost, "Ether too low for minting!");  require(existingURIs[metadataURI] == 0, "This NFT is already minted!");  require(msg.sender != owner(), "Sales not allowed!");     uint256 royality = (msg.value \* royalityFee) / 100;  payTo(artist, royality);  payTo(owner(), (msg.value - royality));   supply++;   minted.push(  TransactionStruct(  supply,  msg.sender,  salesPrice,  title,  description,  metadataURI,  block.timestamp  )  );   emit Sale(  supply,  msg.sender,  msg.value,  metadataURI,  block.timestamp  );   \_safeMint(msg.sender, supply);  existingURIs[metadataURI] = 1;  holderOf[supply] = msg.sender; } |
| --- |

### **Paying to Buy Function**

| function payToBuy(uint256 id) external payable {  require(msg.value >= minted[id - 1].cost, "Ether too low for purchase!");  require(msg.sender != minted[id - 1].owner, "Operation Not Allowed!");   uint256 royality = (msg.value \* royalityFee) / 100;  payTo(artist, royality);  payTo(minted[id - 1].owner, (msg.value - royality));   totalTx++;   transactions.push(  TransactionStruct(  totalTx,  msg.sender,  msg.value,  minted[id - 1].title,  minted[id - 1].description,  minted[id - 1].metadataURI,  block.timestamp  )  );   emit Sale(  totalTx,  msg.sender,  msg.value,  minted[id - 1].metadataURI,  block.timestamp  );   minted[id - 1].owner = msg.sender; } |
| --- |

### **Change Price Function**

| function changePrice(uint256 id, uint256 newPrice) external returns (bool) {  require(newPrice > 0 ether, "Ether too low!");  require(msg.sender == minted[id - 1].owner, "Operation Not Allowed!");   minted[id - 1].cost = newPrice;  return true; } |
| --- |

### **The Payment Function**

| function payTo(address to, uint256 amount) internal {  (bool success, ) = payable(to).call{value: amount}("");  require(success); } |
| --- |

### **NFTs Listing Function**

| function getAllNFTs() external view returns (TransactionStruct[] memory) {  return minted; } |
| --- |

### **NFT Detail Function**

| function getNFT(uint256 id) external view returns (TransactionStruct memory) {  return minted[id - 1];  } |
| --- |

### **Restrieve Transaction Function**

| function getAllTransactions() external view returns (TransactionStruct[] memory) {  return transactions; } |
| --- |

## Migration Script

Head to the folder called `/migrations` on the root of your project, also create a new file called `2\_deploy\_contracts.js`. Next, copy and paste the codes below into this deployment file.

| const TimelessNFT = artifacts.require('TimelessNFT')  module.exports = async (deployer) => {  const accounts = await web3.eth.getAccounts()   await deployer.deploy(TimelessNFT, 'Timeless NFTs', 'TNT', 10, accounts[1]) } |
| --- |

### **Starting the Local Servers**

At this point, open two terminals on your computer, on the first run `ganache-cli -d` to spin up the blockchain server.

On the second terminal, run `truffle migrate –network development` to deploy the smart contract to the network.

Lastly, run `npm start` or `yarn start` to have your React front end running side by side.

## Setting up Store for Blockchain data

| import { createGlobalState } from 'react-hooks-global-state'  const { setGlobalState, useGlobalState, getGlobalState } = createGlobalState({  modal: 'scale-0',  updateModal: 'scale-0',  showModal: 'scale-0',  alert: { show: false, msg: '', color: '' },  loading: { show: false, msg: '' },  connectedAccount: ‘’,  nft: null,  nfts: [],  transactions: [],  contract: null, })  const setAlert = (msg, color = 'green') => {  setGlobalState('loading', false)  setGlobalState('alert', { show: true, msg, color })  setTimeout(() => {  setGlobalState('alert', { show: false, msg: '', color })  }, 6000) }  const setLoadingMsg = (msg) => {  const loading = getGlobalState('loading')  setGlobalState('loading', { ...loading, msg }) }  const truncate = (text, startChars, endChars, maxLength) => {  if (text.length > maxLength) {  var start = text.substring(0, startChars)  var end = text.substring(text.length - endChars, text.length)  while (start.length + end.length < maxLength) {  start = start + '.'  }  return start + end  }  return text }  export {  useGlobalState,  setGlobalState,  getGlobalState,  setAlert,  setLoadingMsg,  truncate, } |
| --- |

## Setting Up Blockchain Service

This function reports errors encountered in this service, see the code below.

| import Web3 from 'web3' import { setGlobalState, getGlobalState, setAlert } from './store' import abi from './abis/TimelessNFT.json'  const { ethereum } = window  window.web3 = new Web3(ethereum) window.web3 = new Web3(window.web3.currentProvider)   const getEtheriumContract = async () => {  const connectedAccount = getGlobalState('connectedAccount')   if (connectedAccount) {  const web3 = window.web3  const networkId = await web3.eth.net.getId()  const networkData = abi.networks[networkId]   if (networkData) {  const contract = new web3.eth.Contract(abi.abi, networkData.address)  return contract  } else {  return null  }  } else {  return getGlobalState('contract')  } }  const connectWallet = async () => {  try {  if (!ethereum) return alert('Please install Metamask')  const accounts = await ethereum.request({ method: 'eth\_requestAccounts' })  setGlobalState('connectedAccount', accounts[0].toLowerCase())  } catch (error) {  reportError(error)  } }  const isWallectConnected = async () => {  try {  if (!ethereum) return alert('Please install Metamask')  const accounts = await ethereum.request({ method: 'eth\_accounts' })   window.ethereum.on('chainChanged', (chainId) => {  window.location.reload()  })   window.ethereum.on('accountsChanged', async () => {  setGlobalState('connectedAccount', accounts[0].toLowerCase())  await isWallectConnected()  })   if (accounts.length) {  setGlobalState('connectedAccount', accounts[0].toLowerCase())  } else {  alert('Please connect wallet.')  console.log('No accounts found.')  }  } catch (error) {  reportError(error)  } }  const reportError = (error) => {  setAlert(JSON.stringify(error), 'red')  throw new Error('No ethereum object.') }  export {  connectWallet,  isWallectConnected, } |
| --- |

## Connecting Wallet

Update the `Header.jsx` component with the following codes.

| import timelessLogo from '../assets/timeless.png' import { connectWallet } from '../Blockchain.Services' import { useGlobalState, truncate } from '../store'  const Header = () => {  const [connectedAccount] = useGlobalState('connectedAccount')  return (  <nav className="w-4/5 flex md:justify-center justify-between items-center py-4 mx-auto">  <div className="md:flex-[0.5] flex-initial justify-center items-center">  <img  className="w-32 cursor-pointer"  src={timelessLogo}  alt="Timeless Logo"  />  </div>   <ul  className="md:flex-[0.5] text-white md:flex  hidden list-none flex-row justify-between  items-center flex-initial"  >  <li className="mx-4 cursor-pointer">Market</li>  <li className="mx-4 cursor-pointer">Artist</li>  <li className="mx-4 cursor-pointer">Features</li>  <li className="mx-4 cursor-pointer">Community</li>  </ul>   {connectedAccount ? (  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f] md:text-xs p-2  rounded-full cursor-pointer"  >  {truncate(connectedAccount, 4, 4, 11)}  </button>  ) : (  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f] md:text-xs p-2  rounded-full cursor-pointer"  onClick={connectWallet}  >  Connect Wallet  </button>  )}  </nav>  ) }  export default Header |
| --- |

### **Updating the App.jsx**

Modify the App.jsx code to allow for automatic wallet account display.

| import { useEffect } from 'react' import { isWallectConnected } from './Blockchain.Services' import Alert from './components/Alert' import Artworks from './components/Artworks' import CreateNFT from './components/CreateNFT' import Footer from './components/Footer' import Header from './components/Header' import Hero from './components/Hero' import Loading from './components/Loading' import ShowNFT from './components/ShowNFT' import Transactions from './components/Transactions' import UpdateNFT from './components/UpdateNFT'  const App = () => {  useEffect(async () => {  await isWallectConnected()  }, [])   return (  <div className="min-h-screen">  <div className="gradient-bg-hero">  <Header />  <Hero />  </div>  <Artworks />  <Transactions />  <CreateNFT />  <ShowNFT />  <UpdateNFT />  <Footer />  <Alert />  <Loading />  </div>  ) }  export default App |
| --- |

## Creating NFT

Update the `CreateNFT.jsx` component with the following codes.

| import {  useGlobalState,  setGlobalState,  setLoadingMsg,  setAlert, } from '../store' import { useState } from 'react' import { FaTimes } from 'react-icons/fa' import { create } from 'ipfs-http-client' import { mintNFT } from '../Blockchain.Services'  const auth =  'Basic ' +  Buffer.from(  process.env.REACT\_APP\_INFURIA\_PID + ':' + process.env.REACT\_APP\_INFURIA\_API,  ).toString('base64')  const client = create({  host: 'ipfs.infura.io',  port: 5001,  protocol: 'https',  headers: {  authorization: auth,  }, })  const CreateNFT = () => {  const [modal] = useGlobalState('modal')  const [title, setTitle] = useState('')  const [price, setPrice] = useState('')  const [description, setDescription] = useState('')  const [fileUrl, setFileUrl] = useState('')  const [imgBase64, setImgBase64] = useState(null)   const handleSubmit = async (e) => {  e.preventDefault()   if (!title || !price || !description) return   setGlobalState('modal', 'scale-0')  setGlobalState('loading', { show: true, msg: 'Uploading IPFS data...' })   try {  const created = await client.add(fileUrl)  const metadataURI = **`https://ipfs.io/ipfs/${created.path}**`  const nft = { title, price, description, metadataURI }   setLoadingMsg('Intializing transaction...')  setFileUrl(metadataURI)  await mintNFT(nft)   resetForm()  setAlert('Minting completed...', 'green')  } catch (error) {  console.log('Error uploading file: ', error)  setAlert('Minting failed...', 'red')  }  }   const changeImage = async (e) => {  const reader = new FileReader()  if (e.target.files[0]) reader.readAsDataURL(e.target.files[0])   reader.onload = (readerEvent) => {  const file = readerEvent.target.result  setImgBase64(file)  setFileUrl(e.target.files[0])  }  }   const closeModal = () => {  setGlobalState('modal', 'scale-0')  resetForm()  }   const resetForm = () => {  setFileUrl('')  setImgBase64(null)  setTitle('')  setPrice('')  setDescription('')  }   return (  <div  className={`fixed top-0 left-0 w-screen h-screen flex items-center  justify-center bg-black bg-opacity-50 transform  transition-transform duration-300 ${modal}`}  >  <div className="bg-[#151c25] shadow-xl shadow-[#e32970] rounded-xl w-11/12 md:w-2/5 h-7/12 p-6">  <form className="flex flex-col">  <div className="flex flex-row justify-between items-center">  <p className="font-semibold text-gray-400">Add NFT</p>  <button  type="button"  onClick={closeModal}  className="border-0 bg-transparent focus:outline-none"  >  <FaTimes className="text-gray-400" />  </button>  </div>   <div className="flex flex-row justify-center items-center rounded-xl mt-5">  <div className="shrink-0 rounded-xl overflow-hidden h-20 w-20">  <img  alt="NFT"  className="h-full w-full object-cover cursor-pointer"  src={  imgBase64 ||  'https://images.unsplash.com/photo-1580489944761-15a19d654956?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1361&q=80'  }  />  </div>  </div>   <div className="flex flex-row justify-between items-center bg-gray-800 rounded-xl mt-5">  <label className="block">  <span className="sr-only">Choose profile photo</span>  <input  type="file"  accept="image/png, image/gif, image/jpeg, image/webp"  className="block w-full text-sm text-slate-500  file:mr-4 file:py-2 file:px-4  file:rounded-full file:border-0  file:text-sm file:font-semibold  file:bg-[#19212c] file:text-gray-400  hover:file:bg-[#1d2631]  cursor-pointer focus:ring-0 focus:outline-none"  onChange={changeImage}  required  />  </label>  </div>   <div className="flex flex-row justify-between items-center bg-gray-800 rounded-xl mt-5">  <input  className="block w-full text-sm  text-slate-500 bg-transparent border-0  focus:outline-none focus:ring-0"  type="text"  name="title"  placeholder="Title"  onChange={(e) => setTitle(e.target.value)}  value={title}  required  />  </div>   <div className="flex flex-row justify-between items-center bg-gray-800 rounded-xl mt-5">  <input  className="block w-full text-sm  text-slate-500 bg-transparent border-0  focus:outline-none focus:ring-0"  type="number"  step={0.01}  min={0.01}  name="price"  placeholder="Price (Eth)"  onChange={(e) => setPrice(e.target.value)}  value={price}  required  />  </div>   <div className="flex flex-row justify-between items-center bg-gray-800 rounded-xl mt-5">  <textarea  className="block w-full text-sm resize-none  text-slate-500 bg-transparent border-0  focus:outline-none focus:ring-0 h-20"  type="text"  name="description"  placeholder="Description"  onChange={(e) => setDescription(e.target.value)}  value={description}  required  ></textarea>  </div>   <button  type="submit"  onClick={handleSubmit}  className="flex flex-row justify-center items-center  w-full text-white text-md bg-[#e32970]  hover:bg-[#bd255f] py-2 px-5 rounded-full  drop-shadow-xl border border-transparent  hover:bg-transparent hover:text-[#e32970]  hover:border hover:border-[#bd255f]  focus:outline-none focus:ring mt-5"  >  Mint Now  </button>  </form>  </div>  </div>  ) }  export default CreateNFT |
| --- |

### **Adding the Mint function**

Don’t forget to export the function along with the rest.

| const mintNFT = async ({ title, description, metadataURI, price }) => {  try {  price = window.web3.utils.toWei(price.toString(), 'ether')  const contract = await getEtheriumContract()  const account = getGlobalState('connectedAccount')  const mintPrice = window.web3.utils.toWei('0.01', 'ether')   await contract.methods  .payToMint(title, description, metadataURI, price)  .send({ from: account, value: mintPrice })   return true  } catch (error) {  reportError(error)  } } |
| --- |

## Listing NFTs

Include this line on the blockchain services file and export.

| const getAllNFTs = async () => {  try {  if (!ethereum) return alert('Please install Metamask')  const contract = await getEtheriumContract()  const nfts = await contract.methods.getAllNFTs().call()  const transactions = await contract.methods.getAllTransactions().call()  setGlobalState('nfts', structuredNfts(nfts))  setGlobalState('transactions', structuredNfts(transactions))  } catch (error) {  reportError(error)  }  }   const structuredNfts = (nfts) => {  return nfts  .map((nft) => ({  id: Number(nft.id),  owner: nft.owner.toLowerCase(),  cost: window.web3.utils.fromWei(nft.cost),  title: nft.title,  description: nft.description,  metadataURI: nft.metadataURI,  timestamp: nft.timestamp,  }))  .reverse() } |
| --- |
|  |

### **Update the App.jsx file**

| import { useEffect } from 'react' import { getAllNFTs, isWallectConnected } from './Blockchain.Services' import Alert from './components/Alert' import Artworks from './components/Artworks' import CreateNFT from './components/CreateNFT' import Footer from './components/Footer' import Header from './components/Header' import Hero from './components/Hero' import Loading from './components/Loading' import ShowNFT from './components/ShowNFT' import Transactions from './components/Transactions' import UpdateNFT from './components/UpdateNFT'  const App = () => {  useEffect(async () => {  await isWallectConnected()  await getAllNFTs()  }, [])   return (  <div className="min-h-screen">  <div className="gradient-bg-hero">  <Header />  <Hero />  </div>  <Artworks />  <Transactions />  <CreateNFT />  <ShowNFT />  <UpdateNFT />  <Footer />  <Alert />  <Loading />  </div>  ) }  export default App |
| --- |

### **Updating the Artwork Component**

| import { setGlobalState, useGlobalState } from '../store'  const Artworks = () => {  const [nfts] = useGlobalState('nfts')   return (  <div className="bg-[#151c25] gradient-bg-artworks">  <div className="w-4/5 py-10 mx-auto">  <h4 className="text-white text-3xl font-bold uppercase text-gradient">  Latest Artworks  </h4>   <div className="grid grid-cols-1 md:grid-cols-3 lg:grid-cols-4 gap-6 md:gap-4 lg:gap-3 py-2.5">  {nfts.map((nft, i) => (  <Card key={i} nft={nft} />  ))}  </div>  <div className="text-center my-5">  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f]  rounded-full cursor-pointer p-2"  >  Load More  </button>  </div>  </div>  </div>  ) }  const Card = ({ nft }) => (  <div className="w-full shadow-xl shadow-black rounded-md overflow-hidden bg-gray-800 my-2 p-3">  <img  src={nft.metadataURI}  alt={nft.title}  className="h-60 w-full object-cover shadow-lg shadow-black rounded-lg mb-3"  />  <h4 className="text-white font-semibold">{nft.title}</h4>  <p className="text-gray-400 text-xs my-1">{nft.description}</p>  <div className="flex justify-between items-center mt-3 text-white">  <div className="flex flex-col">  <small className="text-xs">Current Price</small>  <p className="text-sm font-semibold">{nft.cost} ETH</p>  </div>   <button  className="shadow-lg shadow-black text-white text-sm bg-[#e32970]  hover:bg-[#bd255f] cursor-pointer rounded-full px-1.5 py-1"  onClick={() => setGlobalState('showModal', 'scale-100')}  >  View Details  </button>  </div>  </div> )  export default Artworks |
| --- |

## Showing an NFT

Update the ShowNFT component with this piece of logic.

| import Identicon from 'react-identicons' import { FaTimes } from 'react-icons/fa' import { useGlobalState, setGlobalState, truncate } from '../store'  const ShowNFT = () => {  const [showModal] = useGlobalState('showModal')  const [connectedAccount] = useGlobalState('connectedAccount')  const [nft] = useGlobalState('nft')   const onChangePrice = () => {  setGlobalState('showModal', 'scale-0')  setGlobalState('updateModal', 'scale-100')  }   const handleNFTPurchase = () => {  setGlobalState('showModal', 'scale-0')  }   return (  <div  className={`fixed top-0 left-0 w-screen h-screen flex items-center  justify-center bg-black bg-opacity-50 transform  transition-transform duration-300 ${showModal}`}  >  <div className="bg-[#151c25] shadow-xl shadow-[#e32970] rounded-xl w-11/12 md:w-2/5 h-7/12 p-6">  <div className="flex flex-col">  <div className="flex flex-row justify-between items-center">  <p className="font-semibold text-gray-400">Buy NFT</p>  <button  type="button"  onClick={() => setGlobalState('showModal', 'scale-0')}  className="border-0 bg-transparent focus:outline-none"  >  <FaTimes className="text-gray-400" />  </button>  </div>   <div className="flex flex-row justify-center items-center rounded-xl mt-5">  <div className="shrink-0 rounded-xl overflow-hidden h-40 w-40">  <img  className="h-full w-full object-cover cursor-pointer"  src={nft?.metadataURI}  alt={nft?.title}  />  </div>  </div>   <div className="flex flex-col justify-start rounded-xl mt-5">  <h4 className="text-white font-semibold">{nft?.title}</h4>  <p className="text-gray-400 text-xs my-1">{nft?.description}</p>   <div className="flex justify-between items-center mt-3 text-white">  <div className="flex justify-start items-center">  <Identicon  string={nft?.owner}  size={50}  className="h-10 w-10 object-contain rounded-full mr-3"  />  <div className="flex flex-col justify-center items-start">  <small className="text-white font-bold">@owner</small>  <small className="text-pink-800 font-semibold">  {nft?.owner ? truncate(nft.owner, 4, 4, 11) : '...'}  </small>  </div>  </div>   <div className="flex flex-col">  <small className="text-xs">Current Price</small>  <p className="text-sm font-semibold">{nft?.cost} ETH</p>  </div>  </div>  </div>  <div className="flex justify-between items-center space-x-2">  {connectedAccount == nft?.owner ? (  <button  className="flex flex-row justify-center items-center  w-full text-[#e32970] text-md border-[#e32970]  py-2 px-5 rounded-full bg-transparent  drop-shadow-xl border hover:bg-[#bd255f]  hover:bg-transparent hover:text-white  hover:border hover:border-[#bd255f]  focus:outline-none focus:ring mt-5"  onClick={onChangePrice}  >  Change Price  </button>  ) : (  <button  className="flex flex-row justify-center items-center  w-full text-white text-md bg-[#e32970]  hover:bg-[#bd255f] py-2 px-5 rounded-full  drop-shadow-xl border border-transparent  hover:bg-transparent hover:text-[#e32970]  hover:border hover:border-[#bd255f]  focus:outline-none focus:ring mt-5"  onClick={handleNFTPurchase}  >  Purchase Now  </button>  )}  </div>  </div>  </div>  </div>  ) }  export default ShowNFT |
| --- |

### **Updating the Artwork Component**

| import { setGlobalState, useGlobalState } from '../store'  const Artworks = () => {  const [nfts] = useGlobalState('nfts')   return (  <div className="bg-[#151c25] gradient-bg-artworks">  <div className="w-4/5 py-10 mx-auto">  <h4 className="text-white text-3xl font-bold uppercase text-gradient">  Latest Artworks  </h4>   <div className="grid grid-cols-1 md:grid-cols-3 lg:grid-cols-4 gap-6 md:gap-4 lg:gap-3 py-2.5">  {nfts.map((nft, i) => (  <Card key={i} nft={nft} />  ))}  </div>  <div className="text-center my-5">  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f]  rounded-full cursor-pointer p-2"  >  Load More  </button>  </div>  </div>  </div>  ) }  const Card = ({ nft }) => {  const setNFT = () => {  setGlobalState('nft', nft)  setGlobalState('showModal', 'scale-100')  }   return (  <div className="w-full shadow-xl shadow-black rounded-md overflow-hidden bg-gray-800 my-2 p-3">  <img  src={nft.metadataURI}  alt={nft.title}  className="h-60 w-full object-cover shadow-lg shadow-black rounded-lg mb-3"  />  <h4 className="text-white font-semibold">{nft.title}</h4>  <p className="text-gray-400 text-xs my-1">{nft.description}</p>  <div className="flex justify-between items-center mt-3 text-white">  <div className="flex flex-col">  <small className="text-xs">Current Price</small>  <p className="text-sm font-semibold">{nft.cost} ETH</p>  </div>   <button  className="shadow-lg shadow-black text-white text-sm bg-[#e32970]  hover:bg-[#bd255f] cursor-pointer rounded-full px-1.5 py-1"  onClick={setNFT}  >  View Details  </button>  </div>  </div>  ) }  export default Artworks |
| --- |

## Changing an NFT Price

Modify the UpdateNFT component with this piece of logic.

| import {  useGlobalState,  setGlobalState,  setLoadingMsg,  setAlert, } from '../store' import { updateNFT } from '../Blockchain.Services' import { useState } from 'react' import { FaTimes } from 'react-icons/fa'  const UpdateNFT = () => {  const [modal] = useGlobalState('updateModal')  const [nft] = useGlobalState('nft')  const [price, setPrice] = useState('')   const handleSubmit = async (e) => {  e.preventDefault()  if (!price || price <= 0) return   setGlobalState('modal', 'scale-0')  setGlobalState('loading', { show: true, msg: 'Initiating price update...' })   try {  setLoadingMsg('Price updating...')  setGlobalState('updateModal', 'scale-0')   await updateNFT({ ...nft, cost: price })  setAlert('Price updated...', 'green')  window.location.reload()  } catch (error) {  console.log('Error updating file: ', error)  setAlert('Update failed...', 'red')  }  }   return (  <div  className={`fixed top-0 left-0 w-screen h-screen flex items-center  justify-center bg-black bg-opacity-50 transform  transition-transform duration-300 ${modal}`}  >  <div className="bg-[#151c25] shadow-xl shadow-[#e32970] rounded-xl w-11/12 md:w-2/5 h-7/12 p-6">  <form className="flex flex-col">  <div className="flex flex-row justify-between items-center">  <p className="font-semibold text-gray-400">{nft?.title}</p>  <button  type="button"  onClick={() => setGlobalState('updateModal', 'scale-0')}  className="border-0 bg-transparent focus:outline-none"  >  <FaTimes className="text-gray-400" />  </button>  </div>   <div className="flex flex-row justify-center items-center rounded-xl mt-5">  <div className="shrink-0 rounded-xl overflow-hidden h-20 w-20">  <img  alt={nft?.title}  src={nft?.metadataURI}  className="h-full w-full object-cover cursor-pointer"  />  </div>  </div>   <div className="flex flex-row justify-between items-center bg-gray-800 rounded-xl mt-5">  <input  className="block w-full text-sm  text-slate-500 bg-transparent border-0  focus:outline-none focus:ring-0"  type="number"  step={0.01}  min={0.01}  name="price"  placeholder="Price (Eth)"  onChange={(e) => setPrice(e.target.value)}  required  />  </div>   <button  type="submit"  onClick={handleSubmit}  className="flex flex-row justify-center items-center  w-full text-white text-md bg-[#e32970]  hover:bg-[#bd255f] py-2 px-5 rounded-full  drop-shadow-xl border border-transparent  hover:bg-transparent hover:text-[#e32970]  hover:border hover:border-[#bd255f]  focus:outline-none focus:ring mt-5"  >  Update Now  </button>  </form>  </div>  </div>  ) }  export default UpdateNFT |
| --- |

### **Updating the Blockchain Service**

| const updateNFT = async ({ id, cost }) => {  try {  cost = window.web3.utils.toWei(cost.toString(), 'ether')  const contract = await getEtheriumContract()  const buyer = getGlobalState('connectedAccount')   await contract.methods.changePrice(Number(id), cost).send({ from: buyer })  } catch (error) {  reportError(error)  } } |
| --- |

## Purchasing an NFT

Modify the ShowNFT component with this piece of logic.

| import Identicon from 'react-identicons' import { FaTimes } from 'react-icons/fa' import { useGlobalState, setGlobalState, truncate, setAlert } from '../store' import { buyNFT } from '../Blockchain.Services'  const ShowNFT = () => {  const [showModal] = useGlobalState('showModal')  const [connectedAccount] = useGlobalState('connectedAccount')  const [nft] = useGlobalState('nft')   const onChangePrice = () => {  setGlobalState('showModal', 'scale-0')  setGlobalState('updateModal', 'scale-100')  }   const handleNFTPurchase = async () => {  setGlobalState('showModal', 'scale-0')  setGlobalState('loading', {  show: true,  msg: 'Initializing NFT transfer...',  })   try {  await buyNFT(nft)  setAlert('Transfer completed...', 'green')  window.location.reload()  } catch (error) {  console.log('Error transfering NFT: ', error)  setAlert('Purchase failed...', 'red')  }  }   return (  <div  className={`fixed top-0 left-0 w-screen h-screen flex items-center  justify-center bg-black bg-opacity-50 transform  transition-transform duration-300 ${showModal}`}  >  <div className="bg-[#151c25] shadow-xl shadow-[#e32970] rounded-xl w-11/12 md:w-2/5 h-7/12 p-6">  <div className="flex flex-col">  <div className="flex flex-row justify-between items-center">  <p className="font-semibold text-gray-400">Buy NFT</p>  <button  type="button"  onClick={() => setGlobalState('showModal', 'scale-0')}  className="border-0 bg-transparent focus:outline-none"  >  <FaTimes className="text-gray-400" />  </button>  </div>   <div className="flex flex-row justify-center items-center rounded-xl mt-5">  <div className="shrink-0 rounded-xl overflow-hidden h-40 w-40">  <img  className="h-full w-full object-cover cursor-pointer"  src={nft?.metadataURI}  alt={nft?.title}  />  </div>  </div>   <div className="flex flex-col justify-start rounded-xl mt-5">  <h4 className="text-white font-semibold">{nft?.title}</h4>  <p className="text-gray-400 text-xs my-1">{nft?.description}</p>   <div className="flex justify-between items-center mt-3 text-white">  <div className="flex justify-start items-center">  <Identicon  string={nft?.owner}  size={50}  className="h-10 w-10 object-contain rounded-full mr-3"  />  <div className="flex flex-col justify-center items-start">  <small className="text-white font-bold">@owner</small>  <small className="text-pink-800 font-semibold">  {nft?.owner ? truncate(nft.owner, 4, 4, 11) : '...'}  </small>  </div>  </div>   <div className="flex flex-col">  <small className="text-xs">Current Price</small>  <p className="text-sm font-semibold">{nft?.cost} ETH</p>  </div>  </div>  </div>  <div className="flex justify-between items-center space-x-2">  {connectedAccount == nft?.owner ? (  <button  className="flex flex-row justify-center items-center  w-full text-[#e32970] text-md border-[#e32970]  py-2 px-5 rounded-full bg-transparent  drop-shadow-xl border hover:bg-[#bd255f]  hover:bg-transparent hover:text-white  hover:border hover:border-[#bd255f]  focus:outline-none focus:ring mt-5"  onClick={onChangePrice}  >  Change Price  </button>  ) : (  <button  className="flex flex-row justify-center items-center  w-full text-white text-md bg-[#e32970]  hover:bg-[#bd255f] py-2 px-5 rounded-full  drop-shadow-xl border border-transparent  hover:bg-transparent hover:text-[#e32970]  hover:border hover:border-[#bd255f]  focus:outline-none focus:ring mt-5"  onClick={handleNFTPurchase}  >  Purchase Now  </button>  )}  </div>  </div>  </div>  </div>  ) }  export default ShowNFT |
| --- |

### **Updating the Blockchain Service**

| const buyNFT = async ({ id, cost }) => {  try {  cost = window.web3.utils.toWei(cost.toString(), 'ether')  const contract = await getEtheriumContract()  const buyer = getGlobalState('connectedAccount')   await contract.methods  .payToBuy(Number(id))  .send({ from: buyer, value: cost })   return true  } catch (error) {  reportError(error)  } } |
| --- |

## Listing All Transactions

Modify the Transactions component with this piece of logic.

| import { BiTransfer } from 'react-icons/bi' import { MdOpenInNew } from 'react-icons/md' import { useGlobalState, truncate } from '../store'  const Transactions = () => {  const [transactions] = useGlobalState('transactions')   return (  <div className="bg-[#151c25]">  <div className="w-4/5 py-10 mx-auto">  <h4 className="text-white text-3xl font-bold uppercase text-gradient">  Latest Transactions  </h4>   <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6 md:gap-4 lg:gap-2 py-2.5">  {transactions.map((tx) => (  <div  key={tx.id}  className="flex justify-between items-center border border-pink-500 text-gray-400 w-full shadow-xl shadow-black rounded-md overflow-hidden bg-gray-800 my-2 p-3"  >  <div className="rounded-md shadow-sm shadow-pink-500 p-2">  <BiTransfer />  </div>   <div>  <h4 className="text-sm">{tx.title} Transfered</h4>  <small className="flex flex-row justify-start items-center">  <span className="mr-1">Received by</span>  <a href="#" className="text-pink-500 mr-2">  {truncate(tx.owner, 4, 4, 11)}  </a>  <a href="#">  <MdOpenInNew />  </a>  </small>  </div>   <p className="text-sm font-medium">{tx.cost}ETH</p>  </div>  ))}  </div>   <div className="text-center my-5">  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f]  rounded-full cursor-pointer p-2"  >  Load More  </button>  </div>  </div>  </div>  ) }  export default Transactions |
| --- |

## Enabling the Artworks Load More Button

Update the Artworks component with the following codes.

| import { useEffect, useState } from 'react' import { setGlobalState, useGlobalState } from '../store'  const Artworks = () => {  const [nfts] = useGlobalState('nfts')  const [end, setEnd] = useState(4)  const [count] = useState(4)  const [collection, setCollection] = useState([])   const getCollection = () => {  return nfts.slice(0, end)  }   useEffect(() => {  setCollection(getCollection())  }, [nfts, end])   return (  <div className="bg-[#151c25] gradient-bg-artworks">  <div className="w-4/5 py-10 mx-auto">  <h4 className="text-white text-3xl font-bold uppercase text-gradient">  {collection.length > 0 ? 'Latest Artworks' : 'No Artworks Yet'}  </h4>   <div className="grid grid-cols-1 md:grid-cols-3 lg:grid-cols-4 gap-6 md:gap-4 lg:gap-3 py-2.5">  {collection.map((nft, i) => (  <Card key={i} nft={nft} />  ))}  </div>   {collection.length > 0 && nfts.length > collection.length ? (  <div className="text-center my-5">  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f]  rounded-full cursor-pointer p-2"  onClick={() => setEnd(end + count)}  >  Load More  </button>  </div>  ) : null}  </div>  </div>  ) }  const Card = ({ nft }) => {  const setNFT = () => {  setGlobalState('nft', nft)  setGlobalState('showModal', 'scale-100')  }   return (  <div className="w-full shadow-xl shadow-black rounded-md overflow-hidden bg-gray-800 my-2 p-3">  <img  src={nft.metadataURI}  alt={nft.title}  className="h-60 w-full object-cover shadow-lg shadow-black rounded-lg mb-3"  />  <h4 className="text-white font-semibold">{nft.title}</h4>  <p className="text-gray-400 text-xs my-1">{nft.description}</p>  <div className="flex justify-between items-center mt-3 text-white">  <div className="flex flex-col">  <small className="text-xs">Current Price</small>  <p className="text-sm font-semibold">{nft.cost} ETH</p>  </div>   <button  className="shadow-lg shadow-black text-white text-sm bg-[#e32970]  hover:bg-[#bd255f] cursor-pointer rounded-full px-1.5 py-1"  onClick={setNFT}  >  View Details  </button>  </div>  </div>  ) }  export default Artworks |
| --- |

## Enabling the Transactions Load More Button

Update the Transactions component with the following codes.

| import { useEffect, useState } from 'react' import { BiTransfer } from 'react-icons/bi' import { MdOpenInNew } from 'react-icons/md' import { useGlobalState, truncate } from '../store'  const Transactions = () => {  const [transactions] = useGlobalState('transactions')  const [end, setEnd] = useState(3)  const [count] = useState(3)  const [collection, setCollection] = useState([])   const getCollection = () => {  return transactions.slice(0, end)  }   useEffect(() => {  setCollection(getCollection())  }, [transactions, end])   return (  <div className="bg-[#151c25]">  <div className="w-4/5 py-10 mx-auto">  <h4 className="text-white text-3xl font-bold uppercase text-gradient">  {collection.length > 0 ? 'Latest Transactions' : 'No Transaction Yet'}  </h4>   <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6 md:gap-4 lg:gap-2 py-2.5">  {collection.map((tx) => (  <div  key={tx.id}  className="flex justify-between items-center border border-pink-500 text-gray-400 w-full shadow-xl shadow-black rounded-md overflow-hidden bg-gray-800 my-2 p-3"  >  <div className="rounded-md shadow-sm shadow-pink-500 p-2">  <BiTransfer />  </div>   <div>  <h4 className="text-sm">{tx.title} Transfered</h4>  <small className="flex flex-row justify-start items-center">  <span className="mr-1">Received by</span>  <a href="#" className="text-pink-500 mr-2">  {truncate(tx.owner, 4, 4, 11)}  </a>  <a href="#">  <MdOpenInNew />  </a>  </small>  </div>   <p className="text-sm font-medium">{tx.cost}ETH</p>  </div>  ))}  </div>   {collection.length > 0 && transactions.length > collection.length ? (  <div className="text-center my-5">  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f]  rounded-full cursor-pointer p-2"  onClick={() => setEnd(end + count)}  >  Load More  </button>  </div>  ) : null}  </div>  </div>  ) }  export default Transactions |
| --- |

## Update the Hero Component

| import Identicon from 'react-identicons' import { setGlobalState, useGlobalState, truncate } from '../store'  const Hero = () => {  const [connectedAccount] = useGlobalState('connectedAccount')  const onCreatedNFT = () => {  setGlobalState('modal', 'scale-100')  }   return (  <div  className="flex flex-col md:flex-row w-4/5 justify-between  items-center mx-auto py-10"  >  <div className="md:w-3/6 w-full">  <div>  <h1 className="text-white text-5xl font-bold">  Buy and Sell <br /> Digital Arts, <br />  <span className="text-gradient">NFTs</span> Collections  </h1>  <p className="text-gray-500 font-semibold text-sm mt-3">  Mint and collect the hottest NFTs around.  </p>  </div>   <div className="flex flex-row mt-5">  <button  className="shadow-xl shadow-black text-white  bg-[#e32970] hover:bg-[#bd255f]  rounded-full cursor-pointer p-2"  onClick={onCreatedNFT}  >  Create NFT  </button>  </div>   <div className="w-3/4 flex justify-between items-center mt-5">  <div>  <p className="text-white font-bold">1231k</p>  <small className="text-gray-300">User</small>  </div>  <div>  <p className="text-white font-bold">152k</p>  <small className="text-gray-300">Artwork</small>  </div>  <div>  <p className="text-white font-bold">200k</p>  <small className="text-gray-300">Artist</small>  </div>  </div>  </div>   <div  className="shadow-xl shadow-black md:w-2/5 w-full  mt-10 md:mt-0 rounded-md overflow-hidden bg-gray-800"  >  <img  src="https://images.cointelegraph.com/images/1434\_aHR0cHM6Ly9zMy5jb2ludGVsZWdyYXBoLmNvbS91cGxvYWRzLzIwMjEtMDYvNGE4NmNmOWQtODM2Mi00YmVhLThiMzctZDEyODAxNjUxZTE1LmpwZWc=.jpg"  alt="NFT Art"  className="h-60 w-full object-cover"  />  <div className="flex justify-start items-center p-3">  <Identicon  string={connectedAccount ? connectedAccount : 'Connect Your Wallet'}  size={50}  className="h-10 w-10 object-contain rounded-full mr-3"  />  <div>  <p className="text-white font-semibold">  {connectedAccount  ? truncate(connectedAccount, 4, 4, 11)  : 'Connect Your Wallet'}  </p>  <small className="text-pink-800 font-bold">@you</small>  </div>  </div>  </div>  </div>  ) }  export default Hero |
| --- |