# 102.https://stackoverflow.com/questions/72887031/need-help-adding-function-to-nft-minting-dapp-built-with-next-js

**T:**Need help adding function to NFT minting dapp built with next.js

**Q:**I am building an NFT minting dapp, where users can claim 1 NFT only. I need to add this function to it to check if a user has already minted the NFT, and if they have to return an error message.  
  
This is a link to the API docs: https://portal.thirdweb.com/react/react.useclaimineligibilityreasons  
  
I'm using Thirdweb, and while I have everything else working OK I am just too inexperienced to implement this function without help. What would it look like? How do I "check" if they already have one? Honestly just a nudge in the right direction would help.  
  
I am guessing this function would go into my buttons.js component:  
  
import { useAddress, useDisconnect, useMetamask, useEditionDrop, useChainId, getTotalCount,} from "@thirdweb-dev/react";import { useEffect, useState } from "react";import ReactLoading from "react-loading";import { toast } from "react-toastify";import "react-toastify/dist/ReactToastify.css";import Wrongchain from "./wrongchain";const Buttons = (props) => { const chainId = useChainId(); console.log(chainId); // const [totalSupply, setTotalSupply] = useState(0); const [inProgress, setInProgress] = useState(false); const [completed, setCompleted] = useState(false); // const [isConnected, setConnected] = useState(false); const address = useAddress(); const connectWithMetamask = useMetamask(); const disconnect = useDisconnect(); const editionDrop = useEditionDrop( "0xB0861Ef3C80096c4434E027a4C650CC47D5614C0" ); console.log(address); console.log(props.id); // const connected = async () => { // if (connectWithMetamask && address) { // setConnected(true); // } // }; const mint = async () => { if (editionDrop && address) { setInProgress(true); // set a condition where if the txn doens't happen, in progress reverts back to false const tx = await editionDrop.claimTo(address, props.id, 1); console.log(tx); setInProgress(false); setCompleted(true); toast.success("Mint Succesful!"); } }; const startOver = () => { setCompleted(false); setInProgress(false); disconnect(); }; const targetUrl = `https://testnets.opensea.io/assets/mumbai/0xB0861Ef3C80096c4434E027a4C650CC47D5614C0/${props.id}`; return ( <div> <ButtonContainer className="gap-8"> { // provider ? ( // startApp(provider) // ) : address ? ( // update chainId to 137 for mainnet chainId === 80001 ? ( <Mint> <TitleContainer className="grid gap-6"> <Title>{props.title}</Title> <h2 className="font-light">{props.des}</h2> </TitleContainer> <div className="flex gap-6"> {completed ? ( <a href={targetUrl} target="\_blank"> <button className="bg-white rounded-full transition duration-600 hover:scale-105 border-2 border-black text-black mt-[20px] py-3 px-7"> View on OpenSea </button> </a> ) : inProgress ? ( <div className="pt-4"> <ReactLoading type="bubbles" color="#FFFFFF" height={30} width={65} /> </div> ) : ( <FilledButton disabled={inProgress} onClick={mint}> <>Mint</> </FilledButton> )} <UnfilledButton onClick={startOver} disabled={inProgress}> Disconnect </UnfilledButton> </div> </Mint> ) : ( <Wrongchain /> ) ) : ( <Mint> <TitleContainer className="grid gap-6"> <Title>{props.connect}</Title> <h2>{props.connectdes}</h2> </TitleContainer> <FilledButton onClick={connectWithMetamask}> Connect Wallet </FilledButton> </Mint> ) } </ButtonContainer> </div> );};export default Buttons;  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**My recommendation would be checking how we do it in our embeds: github.com/thirdweb-dev/widgets/blob/main/src/embeds/…

0 **Answer**