

**ONLINE DIGITAL ASSET SHOPPING
USING THIRD WEB AND METAMASK**

Submitted by,

BISWJEET RAUT 20BCE2465

A KAJAL PATRO 20BCE0736

GAURAV GAUR 20BCE0774

Course Code: CSE3501

Course Title: INFORMATION SECURITY ANALYSIS AND AUDIT

Under the guidance of,

Dr. Kakelli Anil Kumar

Associate Professor

SCOPE, VIT, Vellore.



SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

November-2022

INDEX

1. Abstract
2. Introduction
3. Literature Survey
4. Overview of the Work
 - 4.1.Problem description
 - 4.2.Working model
 - 4.3.Design description
 - 4.4. Algorithms
5. Implementation (Code)
6. Result
 - 6.1.Description of Modules/Programs
 - 6.2. Algorithms/ Techniques/ Tools/ source code
 - 6.3.Test cases
 - 6.4. Execution of the project
 - 6.5.Results analysis in the form of graphs and tables, very few screenshots have to given, and all the results have to be explained properly with figure number and table number and their citation in text.
7. Conclusion and Future Scope
8. References

1. ABSTRACT

The present technology we use in shopping applications are AJAX, HTML, CSS and JavaScript which is called Web2.0 technologies. These technologies are more common to find Web 2.0 ideas that either hurt users or simply don't matter to users' core needs. While the latter case might seem innocent, irrelevant website "enhancements" diminish profits because they indicate a failure to focus on those simpler design issues that actually increase sales and leads. So, our project solves this issue by making a web3.0 decentralised shopping website where people can shop any digital assets they want and they can shop using bitcoin, Ethereum or any other currency in their metamask wallet. This provides secure transactions. We have used Next Js and third web for developing the website, there are also a feature of bidding and listing of own items in the website where people can bid on the auctioned items and also can list their own NFTs.

2. INTRODUCTION

Web 2.0 refers to the second generation of the World Wide Web, which is centred on enabling online collaboration and information sharing. Web 2.0 is the evolution from static HTML online pages to a more dynamic, organised web that runs on providing users with web applications. Open communication through online user communities and more open information sharing are two other enhancements of Web 2.0. Wikis, blogs, and web services are additional elements of Web 2.0. By using a Web 2.0 platform for nefarious purposes, which exploits several browser flaws and makes use of AJAX technology to download and run a potentially harmful Trojan from a remote server only by browsing the site, users' computers will become infected without them taking any further action. The online banner adverts that took advantage of a Windows vulnerability and infected millions of users with spyware are another excellent illustration of a Web 2.0 security risk. Their computers would stealthily download a Trojan application that installs ad-ware and floods the user with pop-up advertisements while recording web activity. These problems arises the need of Web3.0 technologies which are better in providing a A decentralized, secure messenger powered by blockchain technology. Not only can you enjoy secure chats with it, but they can also send Cryptocurrency through it. A decentralized data storage solution where anyone can store data with just one click. Next-gen banking platforms that will offer regulation, security, and compliance. A fast privacy-oriented web browser that uses a blockchain-based digital advertising platform to redefine the web for users.

3. LITERATURE SURVEY

1. The IoT (Internet of Things) and CPS (cyber-physical system) are used to allow the creation of digital assets, whilst the key components of blockchain technology, such as tokens and smart contracts, are employed to offer incentive rewarding mechanisms to stimulate user participation and in logistics financing, to carry out the digital asset valuation and risk assessment.
2. Presents an appropriate blockchain application design, focusing on both legislative and commercial requirements, as well as technology implementation. This solution tries to address the flaws and issues that currently exist in asset management solutions. These include inefficient operations, auditing concerns, and diminished asset management skills as a result of a lack of real-time data.
3. The cost of documenting information with Computer-Aided-Design (CAD) vs developing a System Information Model (SIM) is calculated. To assure the integrity of the information necessary for Operations and Management (O&M), a retrospective SIM is built and a bi-directional link with a three-dimensional (3D) model is formed. Provides a foundation for future study in the vast and uncharted field of Digital Asset Management for infrastructure projects.
4. Offers marketing researchers with an overview of how NFTs and smart contracts are being adopted in creative sectors. Illustrate the fundamental concepts of smart contracts and NFTs, demonstrating how they may alter the market by lowering transaction costs. Also, it highlights the restrictions

and obstacles that artists, purchasers, and markets may encounter when using NFTs and smart contracts.

5. The primary functionality of this useful programme will allow users to link their Twitter social media accounts to the NFTs they own, guaranteeing that future purchasers or watchers of the NFT may fully determine who is the genuine owner of a certain NFT. Also conducts performance analysis of the users.
6. Provides machine learning methods for predicting transaction cost and throughput for platforms as decision factors for selecting the best platform from the available options thus selecting the best public blockchain platform.
7. Envisions using NFTs in healthcare, explores the potential benefits of NFTs in several healthcare domains such as supply chain management, patient-centric data management, digital twins, clinical trial management, and genomics.
8. Asset management solutions have been implemented to properly handle digital corporate assets. It briefly discusses digital asset management, media assessment management, content management, financial strategy, development, and ingest-manage-distribute.
9. The decentralised web is focused on establishing protocols and underlying technologies that end users are unaware of. This article provides an outline of the present web 2.0 difficulties. It describes the decentralised web and the technologies that are currently under development.

10. They explored the features and advancements brought about by web 2.0, as well as the security threats connected with them, in this paper, which would lay the groundwork for the requirement of web 3.0.
11. The major focus of this study is on the link between Web 2.0, Web 3.0, and the Semantic Web, with a secondary focus on the growing security concerns concerning rapid and sequential Web advances. However, the future Web 3.0 business models are also addressed. Semantic Web technologies have been viewed as a technical bridge between Web 2.0 and Web 3.0. Attempts have been made to characterise the differences between the existing and future Web.
12. This article investigates the dangers that an organisation faces while working with Web 3.0 technology. The study intends to give insight into the hazards associated with the usage of Web 3.0, as well as to identify potential measures to reduce these risks to an acceptable level.
13. The Next Js framework is used in this article. Because of the possibility of server-side rendered and statically created pages, web page performance may be stored.
14. Next Js is utilised on the frontend using react in this article. It addresses the issue of React apps that frequently load content after javascript has been loaded, resulting in bad SEO and customer experience. Next.js supports server rendering, allowing the app to display its basic state before loading any javascript script.

15. This article discusses how to increase the speed of a production-built next.js app by employing some of the approaches and code patterns learned when developing a badminton data analytics-based web app with next.js.

4. OVERVIEW OF WORK:

4.1 Problem Description:

As businesses adopt Web 2.0 technologies and enjoy benefits like cost savings, improved communications, and improved morale, it's critical that their enthusiasm doesn't lead to ignorance of the risks associated with such technologies. The web is a highly interactive, sophisticated, and increasingly mission-critical platform.

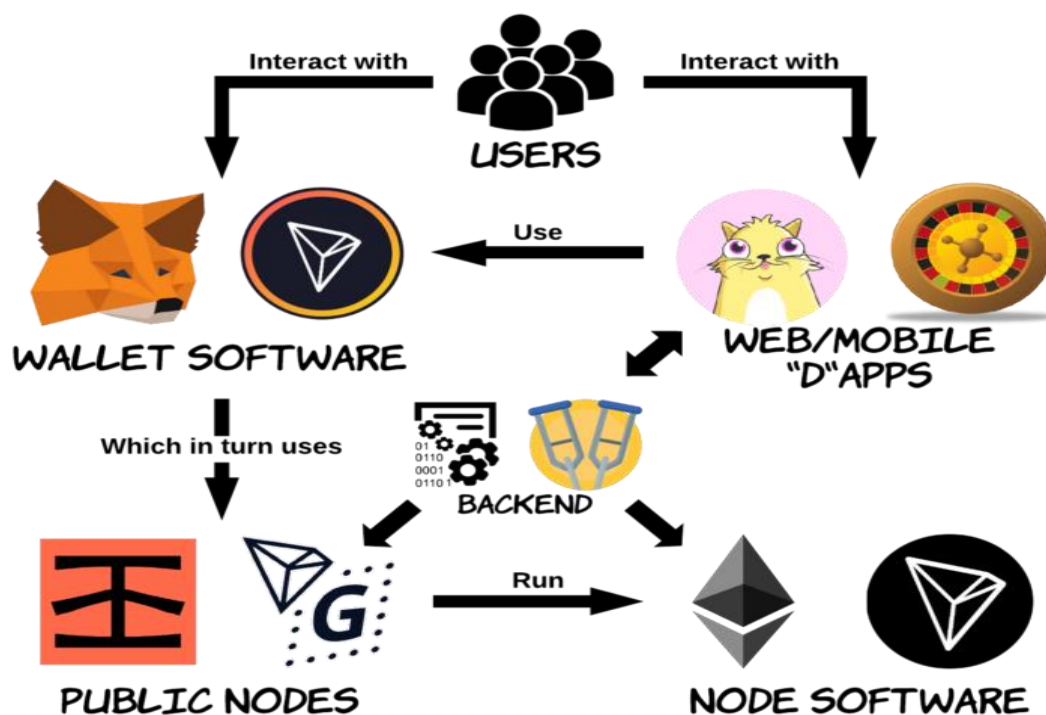
The way businesses and people operate and communicate has been profoundly impacted by the development of the Internet. While Web 2.0 and AJAX have significantly improved user experience and added crucial business functionality, they have also opened up opportunities for hackers to covertly inject and spread malicious code.

Because reactive signature-based solutions are not made to identify these specific dynamic dangerous web scenarios, they are unable on their own to offer the level of security protection needed to thwart contemporary hacking techniques. With today's digital dangers, the conventional wisdom that an anti-virus or URL filtering lab can examine each and every bit of harmful code and produce a signature is no longer true. On the other hand, the crucial solutions to stop these threats are real-time security solutions that can analyse web content on-the-fly as it occurs and detect whether or not it is legitimate, regardless of its source. Last but not least, it's critical that businesses fully comprehend the hazards associated with implementing Web 2.0 in the workplace. Having stated that, while Web 2.0

may create many difficulties, they are not necessarily more severe than the dangers associated with traditional systems. Overcoming these possible challenges is worthwhile due to the opportunity that Web 2.0 technology can offer a company.

4.2 Working Model:

For our shopping app to attain the highest levels of security and dependability, a unique system design is needed. The major example I'll use in this essay is Ethereum, while a lot of it would also apply to EOS, Tron, and other decentralised data platforms is how to properly design and deploy back ends and smart contracts for decentralised applications.



4.2.1. Wallets:

Wallets provide a private key to all the users which is the crypto identity of the users' transactions. These users use this metamask wallet for secure transactions

and decentralised wallet access over the website. All of the transactions happening in our website are being handled or carried out via metamask only.

4.2.2. Backend:

We use back end to host front end for a decentralized application. We use back end for integrations with any other existing technologies and services. Real, world-class applications cannot live in an isolated environment. We use back end to store and process anything big enough for a decentralized network (blockchain in particular). Practically, the whole application and its business logic are stored somewhere in the world, excluding the blockchain part only. Not to mention, IPFS and similar storage layers cannot guarantee the accessibility of files, hence we cannot rely on them without hosting the files ourselves either. In other words, there's always a need for a dedicated running server.

4.2.3. Core Functionality:

Our model provides functionality of buying, listing and bidding. These functionalities are being achieved by the use of Next Js and Third Web. Third Web is used to create NFTs and use them it is also the backend of the website. It is also being maintained by private keys for different users. There is an option for creation of NFT collection which where all the NFTs are being kept as a collection and also there is an option for listing these NFTs and making them available to the marketplace.

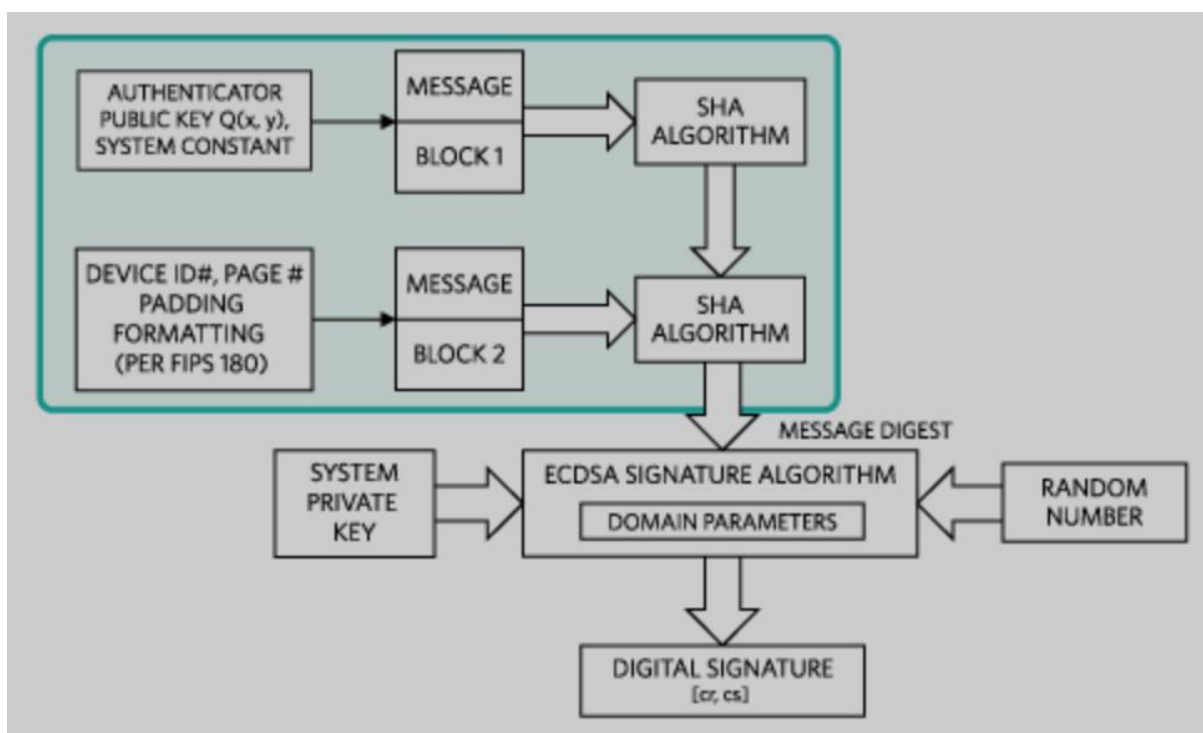
4.3 Design Description:

The design of the website is plain and user-friendly. The user can easily locate the buttons and use the website efficiently. Also, the search bar is fully functional and the connection and failure messages are shown in completely proper and user-friendly way.

4.4 Algorithm:

The full website is based on transactions and these transactions are being made possible due to the use of metamask and teh metamask uses ECDSA algorithm.

ECDSA: Elliptic curve cryptography is used to create the keys for the Elliptic Curve Digital Signature Algorithm (ECDSA), a Digital Signature Algorithm (DSA). It is a public key cryptography-based equation that is especially effective (PKC). ECDSA is used across many security systems, is popular for use in secure messaging apps, and it is the basis of Bitcoin security (with Bitcoin "addresses" serving as public keys). ECDSA is also used for Transport Layer Security (TLS), the successor to Secure Sockets Layer (SSL), by encrypting connections between web browsers and a web application. The encrypted connection of an HTTPS website, illustrated by an image of a physical padlock shown in the browser, is made through signed certificates using ECDSA.



Benefits of ECDSA:

1. Its provides high degree of security with shorter key lengths
2. It is faster.
3. It uses less computer power.

5. IMPLEMENTATION:

Our website aims to provide buying,bidding, auctioning and listing of digital assests. So, the main components of the website are:

1. Authentication with metamask wallet
2. Fetching of Listing from Third Web Marketplace
3. Buying functionality
4. Bidding functionality

1. Authentication:

The authentication is provided by using connection with the metamask wallet of the user/customer. The MetaMask wallet is a flexible solution available on the market that supports numerous protocols on various blockchains. Users can store non-fungible tokens (NFTs) in the wallet in addition to over 670,000 coins on the Ethereum ERC-20 standard and connect them to marketplaces like OpenSea. MetaMask is highly secured in nature. Moreover, the information of the user is encoded in their browser so that no one can get access to it. It has a straightforward user interface.It is accessible via a browser extension and mobile devices.It is a free wallet to use.It has integrated exchanges to make trading simple. There is no need to create unique login information.One can quickly and easily access their Ethereum holdings. So this connection is being established by the use of following code:

```

import { useAddress, useDisconnect, useMetamask } from "@thirdweb-
dev/react";
import Link from "next/link";
const connectWithMetaMask = useMetamask();
const disconnect = useDisconnect();
{/* Top Navbar */}

<nav className="flex justify-between">
{ address ? (
<button onClick={disconnect} className="primary-button">
Hi, {address.slice(0, 5) + "..."} + address.slice(-4)}
</button>
) : (
<button onClick={connectWithMetaMask} className="primary-button">
Connect Your Wallet
</button>
)}

```

The above code is used to link to the metamask account present and also this is used for disconnecting the metamask account.

1. Fetching and Listing of Items from MarketPlace:

```

import React, { useState } from "react";
import { BanknotesIcon, ClockIcon } from "@heroicons/react/24/outline";
import { AuctionListing, DirectListing, ListingType } from "@thirdweb-
dev/sdk";
import { MediaRenderer } from "@thirdweb-dev/react";
import { StarIcon } from "@heroicons/react/24/solid";
import moment from "moment";

```

```
import { useRouter } from "next/router";
```

```
const MAX_RATING = 5;
```

```
const MIN_RATING = 1;
```

```
type Props = {
```

```
  listing: AuctionListing | DirectListing;
```

```
};
```

```
const Listing = ({ listing }: Props) => {
```

```
  const [rating] = useState(
```

```
    Math.floor(Math.random() * (MAX_RATING - MIN_RATING + 1)) +
```

```
    MIN_RATING
```

```
  );
```

```
const router = useRouter();
```

```
return (
```

```
<div
```

```
  className="card flex flex-col"
```

```
  onClick={() => router.push(`/listing/${listing.id}`)}
```

```
>
```

```
<div className="flex h-52 flex-1 flex-col items-center object-cover pb-2">
```

```
<MediaRenderer
```

```
  className="aspect-auto w-full rounded-md"
```

```
  src={listing.asset.image}
```

```
/>
```

```
</div>
```

```
<div className="space-y-4 pt-2">
```

```
<div className="space-y-2">
```

```

<h2 className="truncate text-lg font-semibold text-gray-800">
{listing.asset.name}
</h2>
<hr className="border-gray-300" />
<p className="text-sm text-gray-600 line-clamp-3">
{listing.asset.description}
</p>
<div className="flex">
{Array.from({ length: rating }).map((_, i) => (
<StarIcon className="h-5 text-yellow-500" key={i} />
)))}
</div>
<div className="space-y-1 rounded-lg border bg-[#FBFBFC] p-2">
<p className="text-sm font-semibold text-gray-600 line-clamp-3">
<span className="mr-1 text-blue-500">Hosted On :</span>
{moment
.unix(
listing.type === ListingType.Direct
? listing.startTimeInSeconds
: listing.startTimeInEpochSeconds
)
.format("Do MMM, YYYY")}
</p>
<p className="text-sm font-semibold text-gray-600 line-clamp-3">
<span className="mr-1 text-blue-500">Hosted By :</span>
{listing.sellerAddress.slice(0, 5) +
"... " +
listing.sellerAddress.slice(-4)}
</p>

```

```

</div>
</div>
<p className="flex gap-1">
  <span className="font-bold">
    {listing.buyoutCurrencyValuePerToken.displayValue}
  </span>
  {listing.buyoutCurrencyValuePerToken.symbol}
</p>
<button
  className={`animate-small ml-auto flex w-fit items-center justify-end gap-2
    rounded-md border py-2 px-3 font-semibold text-white ${
    listing.type === ListingType.Direct ? "bg-blue-500" : "bg-red-500"
  }`}
  >
  {listing.type === ListingType.Direct ? "Buy Now" : "Auction"}
  {listing.type === ListingType.Direct ? (
    <BanknotesIcon className="h-6 w-6" />
  ) : (
    <ClockIcon className="h-6 w-6 stroke-2" />
  )}
</button>
</div>
</div>
);
};

```

```
export default Listing;
```

```
import {
```



```
ChevronDownIcon,  
IdentificationIcon,  
} from "@heroicons/react/24/outline";  
import { MediaRenderer } from "@thirdweb-dev/react";  
import { ListingType } from "@thirdweb-dev/sdk";  
import { BigNumberish } from "ethers";  
import React from "react";  
import moment from "moment";  
  
type Props = {  
  name: string | number | undefined;  
  image: string | null | undefined;  
  description: string | null | undefined;  
  sellerAddress: string;  
  tokenId: BigNumberish;  
  endTime: BigNumberish;  
  startTime: BigNumberish;  
  contractAddress: string;  
  type: ListingType;  
};
```

```
const ListingDetails = ({  
  name,  
  image,  
  description,  
  sellerAddress,  
  tokenId,  
  endTime,  
  startTime,
```

```

contractAddress,
type,
}: Props) => {
const data: { name: string; value: string }[] = [
{ name: "Name", value: name },
{ name: "Description", value: description },
{
name: "Seller's Address",
value: sellerAddress.slice(0, 5) + "..." + sellerAddress.slice(-5),
},
{
name: "Contract Address",
value: contractAddress.slice(0, 15) + "...",
},
{ name: "Listing Type", value: type === 0 ? "Direct" : "Auction" },
{
name: "Start Time",
value: moment.unix(startTime).format("Do MMM, YYYY [at] hh:mm a "),
},
{
name: "End Time",
value: moment.unix(endTime).format("Do MMM, YYYY [at] hh:mm a "),
},
];
return (
<div className="grid w-full grid-cols-1 gap-4">
<MediaRenderer
src={image}
className="my-auto w-full cursor-pointer rounded-xl shadow-md"

```

```

/>
<div className="w-full divide-y-2 overflow-hidden rounded-xl border-2
border-gray-300">
  <div className="flex items-center gap-3 py-3 px-5 text-gray-700">
    <IdentificationIcon className="h-7 w-7" />
    <h1 className="-mt-0.5 text-xl font-semibold">Details</h1>
    <ChevronDownIcon className="animate-large ml-auto h-7 w-7 cursor-
pointer" />
  </div>
  <div className="flex flex-col space-y-1 bg-blue-100/30 py-3 px-5 font-
semibold text-gray-500">
    {data.map((item, i) => (
      <div className="grid grid-cols-3 xs:grid-cols-2 gap-x-4 text-sm xs:text-base
whitespace-nowrap" key={i}>
        <h1>{item.name}</h1>
        <h2 className="text-right col-span-2 xs:col-span-1 text-blue-500">
          {item.value}
        </h2>
      </div>
    ))}
  </div>
</div>
</div>
</div>
);
};

```

```
export default ListingDetails;
```

```
import React, { Fragment } from "react";
```

```

import Skeleton from "react-loading-skeleton";
import "react-loading-skeleton/dist/skeleton.css";
import { Fade } from "react-reveal";

export const ListingsLoader = () => {
  return (
    <Fade bottom>
    <div className="mx-auto grid gap-5 px-2 pb-2 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4">
      {Array.from({ length: 4 }).map((_, i) => (
        <div className="card flex flex-col" key={i}>
          <Skeleton height={200} />
          <br />
          <Skeleton className="mb-3" />
          <hr />
          <Skeleton count={3} containerClassName="my-3" />
          <Skeleton />
          <Skeleton width={100} height={35} className="float-right mt-4" />
        </div>
      ))}
    </div>
    </Fade>
  );
};

```

```

export const OwnedNFTsLoader = () => {
  return (
    <Fade bottom>
    <div className="flex space-x-4 overflow-x-scroll px-2 py-4 scrollbar-hide">

```

```

{Array.from({ length: 4 }).map((_, i) => (
  <div className="card flex w-full flex-col" key={i}>
    <Skeleton height={200} width={250} />
    <br />
    <Skeleton className="mb-3" />
    <hr />
    <Skeleton count={3} containerClassName="my-3" />
  </div>
))}
</div>
</Fade>
);
};

```

```

export const ListingLoader = () => {
  return (
    <Fragment>
      <div className="grid w-full grid-cols-1 gap-4">
        <Skeleton height={300} borderRadius={"12px"} />
        <div className="w-full divide-y-2 overflow-hidden rounded-xl border-2
border-gray-300 ">
          <div className="py-3 px-5">
            <Skeleton />
          </div>
          <div className="space-y-1 py-3 px-5">
            {Array.from({ length: 4 }).map((_, i) => (
              <div className="grid grid-cols-2 gap-x-4" key={i}>
                <Skeleton />
                <Skeleton />

```



```
import {
  AdjustmentsHorizontalIcon,
  ChevronDownIcon,
  ClockIcon,
  TagIcon,
  WalletIcon,
  UserCircleIcon,
  PlusIcon,
} from "@heroicons/react/24/outline";
import {
  MediaRenderer,
  useContract,
  useListing,
  useNetwork,
  useNetworkMismatch,
  useMakeBid,
  useOffers,
  useMakeOffer,
  useBuyNow,
  useAddress,
  useAcceptDirectListingOffer,
} from "@thirdweb-dev/react";
import { ListingType, NATIVE_TOKENS } from "@thirdweb-dev/sdk";
import { ethers } from "ethers";
import moment from "moment";
import { useRouter } from "next/router";
import React, { Fragment, useEffect, useState } from "react";
import toast from "react-hot-toast";
import ListingDetails from "../../components/ListingDetails";
```

```
import {
  buyNft,
  createBidOrOffer,
  fetchMinimumBid,
  MinimumBid,
  renderErrorToast,
  renderSuccessToast,
  style,
} from "../functions";
import network from "../utils/network";
import Countdown from "react-countdown";
import { ListingLoader } from "../components/Loader";

const ListingPage = () => {
  const { contract } = useContract(
    process.env.NEXT_PUBLIC_MARKETPLACE_CONTRACT,
    "marketplace"
  );
  const router = useRouter();
  const { listingId } = router.query as { listingId: string };
  const { data: listing, isLoading } = useListing(contract, listingId);
  const { mutate: makeOffer } = useMakeOffer(contract);
  const { data: offers } = useOffers(contract, listingId);
  const [minimumBid, setMinimumBid] = useState<MinimumBid>();
  const { mutate: makeBid } = useMakeBid(contract);
  const [, switchNetwork] = useNetwork();
  const networkMismatch = useNetworkMismatch();
  const address = useAddress();
  const { mutate: acceptOffer } = useAcceptDirectListingOffer(contract);
```



```

const [bidAmount, setBidAmount] = useState("");
useEffect(() => {
  if (!listing || !contract || !listingId) return;
  if (listing.type === ListingType.Auction)
    fetchMinimumBid(contract, listingId).then((minBid) =>
      setMinimumBid(minBid)
    );
}, [listing, contract, listingId]);

```

```

const { mutate: buyNow } = useBuyNow(contract);
return (
  <main className="mx-auto mb-4 grid max-w-6xl grid-cols-1 gap-5 space-y-10
p-2 pr-10 lg:grid-cols-2 lg:space-y-0">
    {isLoading ? (
      <ListingLoader />
    ) : !listing ? (
      "Not Found"
    ) : (
      <Fragment>
        <ListingDetails
          name={listing.asset.name}
          image={listing.asset.image}
          description={listing.asset.description}
          sellerAddress={listing.sellerAddress}
          tokenId={listing.tokenId}
          endTime={
            listing.type === ListingType.Direct
              ? listing.secondsUntilEnd
              : listing.endTimeInEpochSeconds
          }
        />
      </Fragment>
    )}
  </main>
)

```

```

}
startTime={
listing.type === ListingType.Direct
? listing.startTimeInSeconds
: listing.startTimeInEpochSeconds
}
contractAddress={listing.assetContractAddress}
type={listing.type}
/>
<section className="space-y-4 pb-20 lg:pb-0">
<div className="w-full divide-y-2 overflow-hidden rounded-xl border-2
border-gray-300">
<div className="whitespace- flex items-center justify-between space-x-2 py-3
px-3 text-gray-700">
<ClockIcon className="h-4 w-4 xs:h-7 xs:w-7" />
<h1 className="-mt-0.5 text-[15px] font-semibold xs:text-lg sm:text-xl">
Sale Ends 
{moment
.unix(
listing.type === ListingType.Direct
? listing.secondsUntilEnd
: listing.endTimeInEpochSeconds
)
.format("Do MMM, YYYY [at] hh:mm a ")}{" "}
</h1>
<ChevronDownIcon className="animate-large hidden h-7 w-7 cursor-pointer
xs:inline" />
</div>

```

```

<div className="flex flex-col space-y-1 bg-blue-100/30 py-3 px-5 font-
semibold text-gray-500">
<h1 className="font-normal">Buy Now Price</h1>
<h2 className="text-2xl font-semibold">
{listing.buyoutCurrencyValuePerToken.displayValue}&nbsp;
{listing.buyoutCurrencyValuePerToken.symbol}
</h2>
<br />
<button
className="primary-button rounded-lg hover:scale-x-[1.04]"
onClick={() =>
buyNft(
networkMismatch,
switchNetwork,
listing,
contract,
listingId,
buyNow,
router
)
}
>
Buy Now
</button>
</div>

<div className="grid grid-cols-3 gap-4 bg-blue-100/30 py-3 px-5 text-gray-
500 xs:grid-cols-2">
<h1 className="col-span-3 text-xl font-semibold underline sm:text-2xl">
{listing.type === ListingType.Direct

```



```

    value={bidAmount}
    onChange={(e) => setBidAmount(e.target.value)}
    className="form-input peer"
    required
  />

  <label className="floating-label bg-[#F1F6FD]">
    {listing.type === ListingType.Direct
      ? "Enter Offer Amount"
      : minimumBid
      ? Number(minimumBid.displayValue) >= 0 &&
        `${minimumBid.displayValue} ${minimumBid.symbol} or more`
      : "Enter Bid Amount"}
  </label>

</div>

<button
  className="side-nav-button col-span-full mx-auto max-w-full whitespace-
nowrap rounded-lg xs:col-span-1 xs:mx-0"
  onClick={() =>
    createBidOrOffer(
      networkMismatch,
      switchNetwork,
      listing,
      contract,
      buyNow,
      router,
      listingId,
      makeBid,
      bidAmount,
      makeOffer,

```

```

setBidAmount
)
}
>
{listing.type === ListingType.Direct ? (
<TagIcon className="mr-2 h-6 w-6 stroke-2" />
): (
<WalletIcon className="mr-2 h-6 w-6" />
)}
{listing.type === ListingType.Direct
? "Make Offer"
: "Place Bid"}
</button>
</div>
</div>
{listing.type === ListingType.Direct && offers && (
<div className="w-full divide-y-2 overflow-hidden rounded-xl border-2
border-gray-300">
<div className="flex items-center gap-3 py-3 px-5 text-gray-700">
<AdjustmentsHorizontalIcon className="h-7 w-7" />
<h1 className="-mt-0.5 text-xl font-semibold">Offers</h1>
<ChevronDownIcon className="animate-large ml-auto h-7 w-7 cursor-
pointer" />
</div>
{offers.length > 0 && (
<div className="flex flex-col divide-y-2 bg-blue-100/30 px-5 font-semibold
text-gray-500">
{offers.map((offer, i) => (
<div

```

```

    className="flex w-full items-center justify-between space-x-2 py-2 px-3 text-
lg"
    key={i}
  >
  <h1 className="hidden items-center gap-2 sm:inline-flex">
  <UserCircleIcon className="h-6 w-6" />
  {offer.offeror.slice(0, 5)} +
  "..." +
  offer.offeror.slice(-5)}
  </h1>
  <h2>
  {ethers.utils.formatEther(offer.totalOfferAmount)}
  &nbsp;
  {NATIVE_TOKENS[network].symbol}
  </h2>
  {listing.sellerAddress === address && (
  <button
  className="side-nav-button max-w-fit gap-2 shadow-none"
  onClick={(e) => {
  e.preventDefault();
  const loadingToast = toast.loading("Loading", {
  style: style,
  }});
  acceptOffer(
  {
  listingId,
  addressOfOfferor: offer.offeror,
  },
  {

```

```
onSuccess() {
renderSuccessToast(
loadingToast,
"Offered Accepted Successfully"
);
},
onError(error) {
renderErrorToast(
loadingToast,
"Error Accepting Offer"
);
console.error(error);
},
}
);
}}
>
<PlusIcon className="h-5 w-5" /> Accept Offer
</button>
)}
</div>
)})
</div>
)}
</div>
)}
</section>
</Fragment>
)}
```



```
</main>
```

```
);
```

```
};
```

```
export default ListingPage;
```

2. Buying Functionality:

```
<ClockIcon className="h-4 w-4 xs:h-7 xs:w-7" />
```

```
<h1 className="-mt-0.5 text-[15px] font-semibold xs:text-lg sm:text-xl">
```

```
Sale Ends&nbsp;
```

```
{moment
```

```
.unix(
```

```
listing.type === ListingType.Direct
```

```
? listing.secondsUntilEnd
```

```
: listing.endTimeInEpochSeconds
```

```
)
```

```
.format("Do MMM, YYYY [at] hh:mm a ")){ " "}
```

```
</h1>
```

```
<ChevronDownIcon className="animate-large hidden h-7 w-7 cursor-pointer  
xs:inline" />
```

```
</div>
```

```
<div className="flex flex-col space-y-1 bg-blue-100/30 py-3 px-5 font-  
semibold text-gray-500">
```

```
<h1 className="font-normal">Buy Now Price</h1>
```

```
<h2 className="text-2xl font-semibold">
```

```
{listing.buyoutCurrencyValuePerToken.displayValue}&nbsp;
```

```
{listing.buyoutCurrencyValuePerToken.symbol}
```

```
</h2>
```

```
<br />
```

```

<button
  className="primary-button rounded-lg hover:scale-x-[1.04]"
  onClick={() =>
    buyNft(
      networkMismatch,
      switchNetwork,
      listing,
      contract,
      listingId,
      buyNow,
      router
    )
  }
>
Buy Now
</button>

```

3. Bidding Functionality:

```

<div className="grid grid-cols-3 gap-4 bg-blue-100/30 py-3 px-5 text-gray-500 xs:grid-cols-2">
  <h1 className="col-span-3 text-xl font-semibold underline sm:text-2xl">
    {listing.type === ListingType.Direct
      ? "Make an Offer"
      : "Bid on this Auction"}
  </h1>
  {listing.type === ListingType.Auction && (
    <p className="col-span-3 grid grid-cols-2 text-lg font-semibold sm:text-xl">
      <h1>Minimum Bid:</h1>
      <h2 className="text-right">

```

```

{minimumBid?.displayValue} {minimumBid?.symbol}
</h2>
<h1>Time Remaining:</h1>
<h2 className="text-right">
<CountDown
  renderer={({ hours, minutes, seconds, days }) => (
    <span>
      {days * 24 + hours}:{minutes}:{seconds}
    </span>
  )}
  date={
    Number(listing.endTimeInEpochSeconds.toString()) *
    1000
  }
/>
</h2>
</p>
)}
<div className="group relative z-0 col-span-3 w-full xs:col-span-2">
  <input
    type="text"
    placeholder=" "
    value={bidAmount}
    onChange={(e) => setBidAmount(e.target.value)}
    className="form-input peer"
    required
  />
  <label className="floating-label bg-[#F1F6FD]">
    {listing.type === ListingType.Direct

```

```

? "Enter Offer Amount"
: minimumBid
? Number(minimumBid.displayValue) >= 0 &&
`$ {minimumBid.displayValue} $ {minimumBid.symbol} or more`
: "Enter Bid Amount" }
</label>
</div>
<button
className="side-nav-button col-span-full mx-auto max-w-full whitespace-
nowrap rounded-lg xs:col-span-1 xs:mx-0"
onClick={() =>
createBidOrOffer(
networkMismatch,
switchNetwork,
listing,
contract,
buyNow,
router,
listingId,
makeBid,
bidAmount,
makeOffer,
setBidAmount
)
}
>
{listing.type === ListingType.Direct ? (
<TagIcon className="mr-2 h-6 w-6 stroke-2" />
): (

```

```

<WalletIcon className="mr-2 h-6 w-6" />
))}
{listing.type === ListingType.Direct
? "Make Offer"
: "Place Bid"}
</button>
</div>
</div>
{listing.type === ListingType.Direct && offers && (
<div className="w-full divide-y-2 overflow-hidden rounded-xl border-2
border-gray-300">
<div className="flex items-center gap-3 py-3 px-5 text-gray-700">
<AdjustmentsHorizontalIcon className="h-7 w-7" />
<h1 className="-mt-0.5 text-xl font-semibold">Offers</h1>
<ChevronDownIcon className="animate-large ml-auto h-7 w-7 cursor-
pointer" />
</div>
{offers.length > 0 && (
<div className="flex flex-col divide-y-2 bg-blue-100/30 px-5 font-semibold
text-gray-500">
{offers.map((offer, i) => (
<div
className="flex w-full items-center justify-between space-x-2 py-2 px-3 text-
lg"
key={i}
>
<h1 className="hidden items-center gap-2 sm:inline-flex">
<UserCircleIcon className="h-6 w-6" />
{offer.offeror.slice(0, 5) +

```

```

"..." +
offer.offeror.slice(-5)}
</h1>
<h2>
{ ethers.utils.formatEther(offer.totalOfferAmount)}
 
{ NATIVE_TOKENS[network].symbol}
</h2>
{ listing.sellerAddress === address && (
<button
className="side-nav-button max-w-fit gap-2 shadow-none"
onClick={ (e) => {
e.preventDefault();
const loadingToast = toast.loading("Loading", {
style: style,
});
acceptOffer(
{
listingId,
addressOfOfferor: offer.offeror,
},
{
onSuccess() {
renderSuccessToast(
loadingToast,
"Offered Accepted Successfully"
);
},
onError(error) {

```


```
renderErrorToast(  
  loadingToast,  
  "Error Accepting Offer"  
);  
console.error(error);  
},  
}  
);  
}}  
>  
<PlusIcon className="h-5 w-5" /> Accept Offer  
</button>  
))}  
</div>  
)))  
</div>  
))  
</div>  
))  
</section>  
</Fragment>  
))  
</main>
```

6. RESULT

Apps CodePen Search Color wheel, a color... AWS Academy Clou... Javascript 3D - Bui... Javascript Cheat Sh... Novoresumé SidharthMishra 07 (... SQL Commands Program Organizati...

Hi, 0x098...4cd3 Daily Deals Ship to Sell WatchList

MARKETPLACE 3.0 Shop Search for Anything... List Item


SEEDHE MAUT & SEZ ON THE BEAT
DEL 17 JULY
TICKETS ON NAYAAB.WORLD

Seedhe maut Tour Ticke...

★★★★

Hosted On : 13th Nov, 2022
Hosted By : 0xd6C...254b

0.00004 MATIC


Buy Now

Windows taskbar: 19:48 13-11-2022

Apps CodePen Search Color wheel, a color... AWS Academy Clou... Javascript 3D - Bui... Javascript Cheat Sh... Novoresumé SidharthMishra 07 (... SQL Commands Program Organizati...

Hi, 0x098...4cd3 Daily Deals Ship to Sell WatchList

MARKETPLACE 3.0 Shop Search for Anything... List Item


AZADI RECORDS & THE MYMNT PRESENT
Teen Dost Tour
SEEDHE MAUT & SEZ ON THE BEAT
DEL 17 JULY
with AB7 & FULL POWER + MO STYLES
SHERATON, NEW DELHI
TICKETS ON NAYAAB.WORLD

Sale Ends 20th Oct, 2122 at 02:38 pm

Buy Now Price
0.00004 MATIC

Buy Now

Make an Offer


Enter Offer Amount Make Offer

Offers

Windows taskbar: 19:49 13-11-2022

Hi, 0x098...4cd3 Loading Daily Deals Ship to Sell WatchList

MARKETPLACE 3.0 Shop Search for Anything BuyOut Price Met, Buying NFT List Item



SEEDHE MAUT & SEZ ON THE BEAT
with AB7 & FULL
POWER + MO STYLES
DEL 17
JULY
SHERATON, NEW DELHI

TICKETS ON NAYAAB.WORLD

Sale Ends 20th Oct, 2122 at 02:38 pm

Buy Now Price
0.00004 MATIC

Buy Now

Make an Offer


Enter Offer Amount
0.00004

Make Offer

Offers

Hi, 0x098...4cd3 BuyOut Price Not Met, Making Offer Daily Deals Ship to Sell WatchList

MARKETPLACE 3.0 Shop Search for Anything List Item



SEEDHE MAUT & SEZ ON THE BEAT
with AB7 & FULL
POWER + MO STYLES
DEL 17
JULY
SHERATON, NEW DELHI

TICKETS ON NAYAAB.WORLD

Sale Ends 20th Oct, 2122 at 02:38 pm

Buy Now Price
0.00004 MATIC

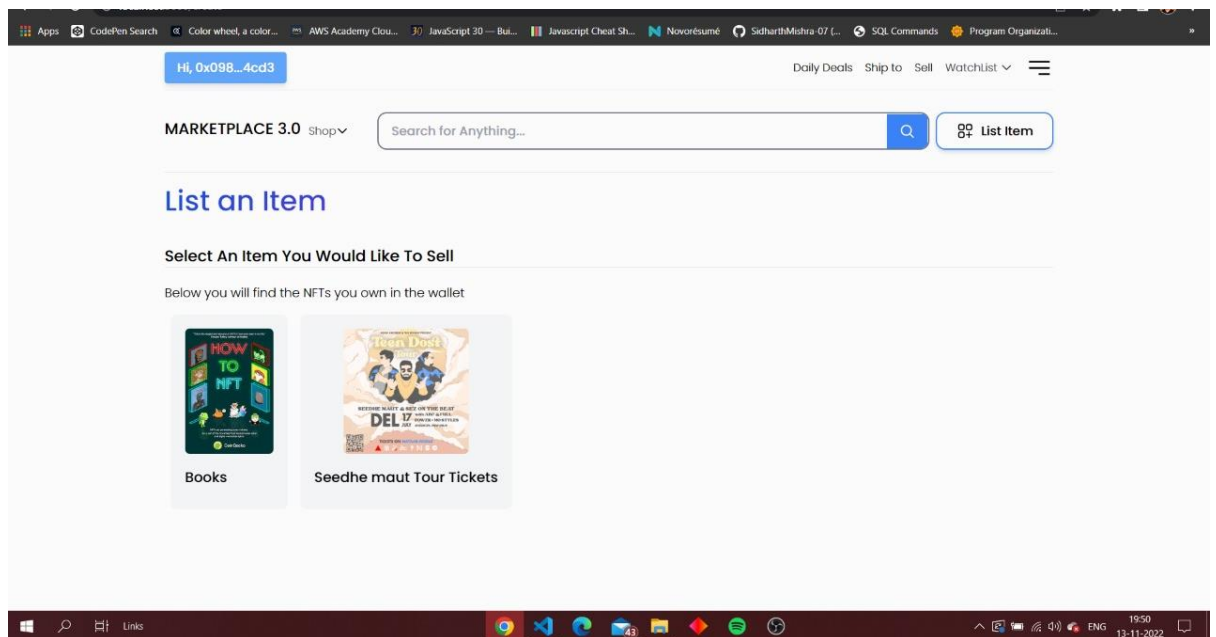
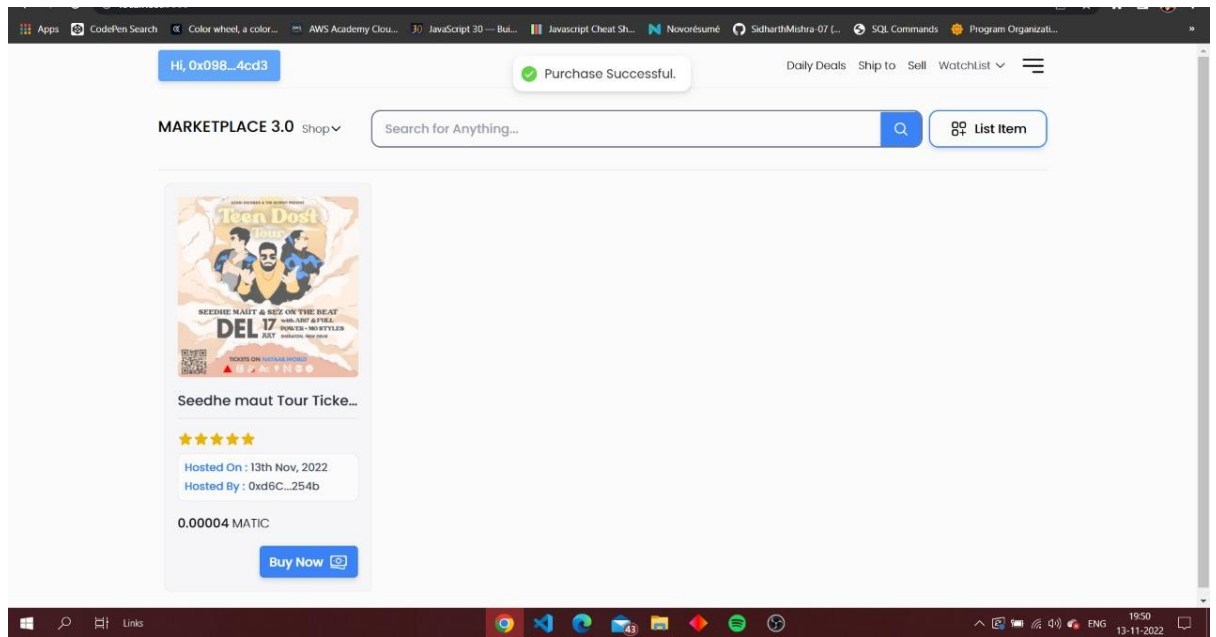
Buy Now

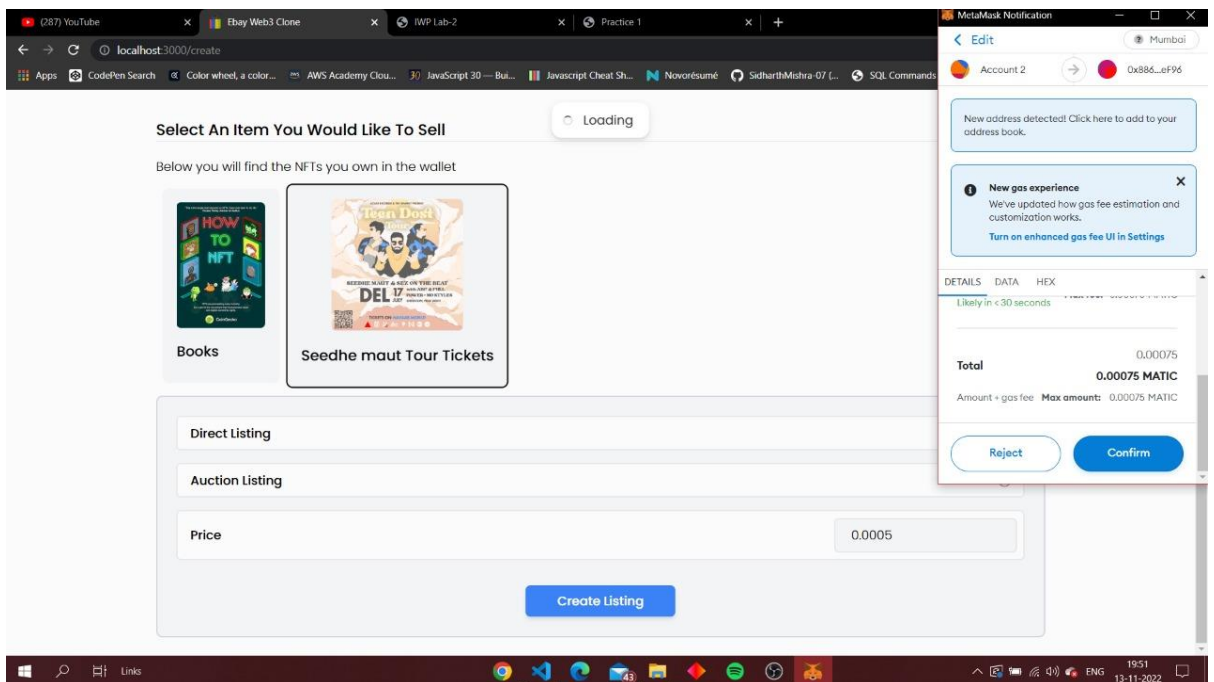
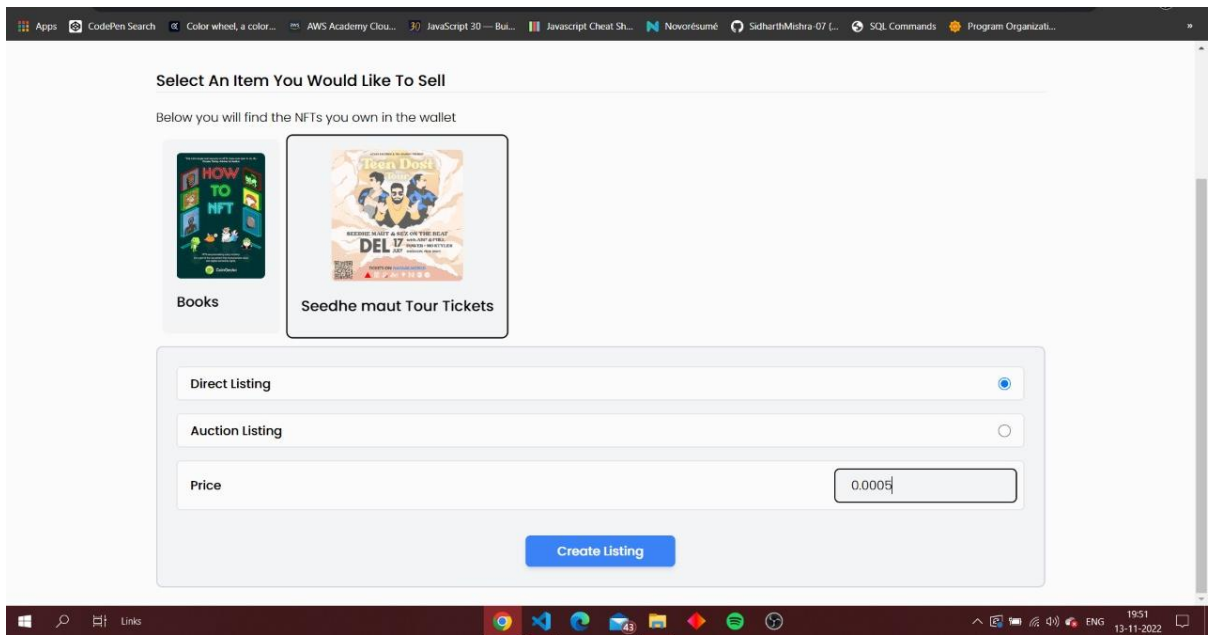
Make an Offer

Enter Offer Amount
0.00008

Make Offer

Offers





7. CONCLUSION AND FUTURE SCOPE

There is an emerging need of Web3.0 technologies in the E-commerce field and our project satisfies the current need and gives a solution for secured transactions. But the Web3.0 technologies have a limited storage capacity so we need to increase the storage capacity. Also, the management of the accounts is needed in a secure and proper way. In our project we can add location and delivery features. With addition of proper AI model, we can give more better user experience and better working website will be delivered out of it. But the current problem is being solved a quite well level by our project. Our project provides a decent level of user experience and easy sign in facility to the metamask wallet account. Also, our project stands out in providing the facilities of buying, listing items. The crypto algorithm and the SHA algorithm metamask use provides a better and faster security which is better for an E-commerce website as it won't be that heavy to a website and it will give user a faster and secure experience.

8. REFERENCES

- [1] Harish, A. R., Liu, X. L., Zhong, R. Y., & Huang, G. Q. (2021). Log-flock: A blockchain-enabled platform for digital asset valuation and risk assessment in E-commerce logistics financing. *Computers & Industrial Engineering*, 151, 107001.
- [2] Kuhle, P., Arroyo, D., & Schuster, E. (2021). Building A blockchain-based decentralized digital asset management system for commercial aircraft leasing. *Computers in Industry*, 126, 103393.
- [3] Love, P. E., Zhou, J., Matthews, J., & Luo, H. (2016). Systems information modelling: Enabling digital asset management. *Advances in Engineering Software*, 102, 155-165.
- [4] Malik, N., Wei, M. Y., Appel, G., & Luo, L. (2022). Blockchain Technology for Creative Industry: Current State and Research Opportunities. *International Journal of Research in Marketing*.
- [5] Bellagarda, J., & Abu-Mahfouz, A. M. (2022). Connect2NFT: A Web-Based, Blockchain Enabled NFT Application with the Aim of Reducing Fraud and Ensuring Authenticated Social, Non-Human Verified Digital Identity. *Mathematics*, 10(21), 3934.
- [6] Rasolroveicy, M., & Fokaefs, M. (2022, September). Performance and Cost Evaluation of Public Blockchain: An NFT Marketplace Case Study. In *2022 4th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)* (pp. 79-86). IEEE.

[7] Musamih, A., Salah, K., Jayaraman, R., Yaqoob, I., Puthal, D., & Ellahham, S. (2022). NFTs in healthcare: Vision, opportunities, and challenges. *IEEE Consumer Electronics Magazine*.

[8] Wager, S. (2005). Digital asset management, media asset management, and content management: From confusion to clarity. *Journal of Digital Asset Management*, 1(1), 40-45.

[9] Alabdulwahhab, F. A. (2018, April). Web 3.0: the decentralized web blockchain networks and protocol innovation. In 2018 1st International Conference on Computer Applications & Information Security (ICCAIS) (pp. 1-4). IEEE.

[10] Saqib, N. A., Salam, A. A., Atta-Ur-Rahman, & Dash, S. (2021). Reviewing risks and vulnerabilities in web 2.0 for matching security considerations in web 3.0. *Journal of Discrete Mathematical Sciences and Cryptography*, 24(3), 809-825.

[11] Pattal, M. M. I., Li, Y., & Zeng, J. (2009, September). Web 3.0: A real personal web! More opportunities and more threats. In 2009 Third International Conference on Next Generation Mobile Applications, Services and Technologies (pp. 125-128). IEEE.

[12] Rudman, R. (2015). Web 3.0: governance, risks and safeguards. *Journal of Applied Business Research (JABR)*, 31(3), 1037-1056.

[13] Nguyen, A. (2022). Building an E-commerce Website Using Next Js, Mantine, and Strapi.

[14] Chu, S. (2020). Shopify Upsell App: Using Next.js, React.js to boost sale.

[15] Sasikumar, S., Prabha, S., & Mohan, C. (2022). Improving Performance Of Next.js App And Testing It While Building A Badminton Based Web App. Js App And Testing It While Building A Badminton Based Web App (May 27, 2022).

9. GITHUB URL:

<https://github.com/BiswajeetRaut/marketplaceWeb3.0>