# **Source Code**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**Folder Structure:**

A screenshot of a computer screen

Description automatically generated

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**app>api>comparison>route.js**

import { NextResponse } from "next/server";

import Redis from "ioredis";

const axios = require("axios");

const cheerio = require("cheerio");

const scraper = async (url, comp) => {

try {

const redis = new Redis(process.env.REDIS\_URI);

// Check if Data exists in Redis cache

const cachedData = await redis.get(comp);

if (cachedData) {

return JSON.parse(cachedData);

}

let url2 = "";

const baseUrl = "https://www.google.com";

const axiosResponse1 = await axios.request({

method: "GET",

url: url,

headers: {

"User-Agent":

"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36",

},

});

const $$ = cheerio.load(axiosResponse1.data);

url2 = baseUrl + $$(".EJbZzc").attr("href");

if (url2 == "https://www.google.comundefined") {

url2 = url;

}

const mainData = [];

const content = [];

const axiosResponse = await axios.request({

method: "GET",

url: url2,

headers: {

"User-Agent":

"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36",

},

});

const $ = cheerio.load(axiosResponse.data);

const imgLink =

$(".lBRvsb").find(".r4m4nf").attr("src") ||

$(".Xkiaqc").find("img").attr("src");

const title =

$(".lBRvsb").find(".BvQan").text() ||

$(".LDQll").find(".BvQan").text();

const review =

$(".lBRvsb").find(".UzThIf").attr("aria-label") ||

$(".aE5Gic").find(".UzThIf").attr("aria-label");

const reviewOutOf =

$(".lBRvsb").find(".HiT7Id").find("span").attr("aria-label") ||

$(".aE5Gic").find(".HiT7Id").find("span").attr("aria-label");

$(".sh-osd\_\_offer-row").each((index, element) => {

// scraping logic...

const compBrand =

$(element)

.find(".kPMwsc")

.find("a")

.text()

.replace("Opens in a new window", "") ||

$(element)

.find(".\_-ey")

.find("a")

.text()

.replace("Opens in a new window", "");

const compLink =

baseUrl + $(element).find(".kPMwsc").find("a").attr("href") ||

baseUrl + $(element).find(".\_-ey").find("a").attr("href");

const delivery = $(element).find(".yGibJf").text();

const itemPrice = $(element).find(".fObmGc").text();

const totalPrice = $(element).find(".drzWO").text();

const dataFromResponce = {

compBrand,

compLink,

delivery,

itemPrice,

totalPrice,

};

content.push(dataFromResponce);

});

const mainDataContent = {

title,

review,

reviewOutOf,

imgLink,

content,

};

mainData.push(mainDataContent);

await redis.set(comp, JSON.stringify(mainData));

await redis.expire(comp, 86400);

return mainData;

} catch (error) {

console.error(error);

console.log("Scrapper Error");

}

};

export async function GET(request, { params }) {

const { comp } = params;

const mainUrl = "https://www.google.com/shopping/product/";

const url = mainUrl + comp + "?gl=in";

const data = await scraper(url, comp);

return NextResponse.json(data);

}

**app>api>item>route.js**

import { NextResponse } from "next/server";

import Redis from "ioredis";

const axios = require("axios");

const cheerio = require("cheerio");

const main\_url = "https://www.google.com";

const base\_url = "https://www.google.com/search?tbm=shop&gl=in&hl=en&q=";

const scraper = async (url, item) => {

const data = [];

try {

const redis = new Redis(process.env.REDIS\_URI);

// Check if Data exists in Redis cache

const cachedData = await redis.get(item);

if (cachedData) {

console.log("Data Found From Cache");

return JSON.parse(cachedData);

}

const axiosResponse = await axios.request({

method: "GET",

url: url,

headers: {

"User-Agent":

"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36",

},

});

const $ = cheerio.load(axiosResponse.data);

$(".sh-dgr\_\_content").each((index, element) => {

// scraping logic...

const comp = $(element).find(".iXEZD").attr("href");

const link =

main\_url + $(element).find(".zLPF4b").find("a").attr("href");

let link\_text = $(element).find(".shntl").find(".aULzUe").text();

let rating = $(element).find(".NzUzee").find(".Rsc7Yb").text();

let delivery = $(element).find(".vEjMR").text();

let id = $(element)

.find(".VOo31e")

.find("a")

.attr("href")

.split("?")[0]

.split("/")[3];

let rating\_str = $(element)

.find(".NzUzee")

.find("div")

.find("span")

.text();

const total\_people\_review = rating\_str.slice(

rating\_str.indexOf("stars.") + 7

);

if (link\_text.startsWith(".")) {

link\_text = link\_text.split("}").slice(-1)[0];

}

let Compare = null;

if (comp != undefined) {

Compare = main\_url + comp;

}

const title = $(element).find(".tAxDx").text();

const price = $(element).find(".a8Pemb").text();

const dataFromResponce = {

title,

price,

Compare,

link,

link\_text,

rating,

total\_people\_review,

delivery,

id,

};

data.push(dataFromResponce);

});

await redis.set(item, JSON.stringify(data));

await redis.expire(item, 86400);

} catch (error) {

console.error(error);

}

return data;

};

export async function GET(request, { params }) {

const { item } = params;

const url = base\_url + item;

const data = await scraper(url, item);

return NextResponse.json(data);

}

**app>search>page.js**

"use client";

import React, { useEffect } from "react";

import Card from "@/components/Card";

import axios from "axios";

import { useState } from "react";

import LoadingSkeletonWithOutImg from "@/components/LoadingSkeletonWithOutImg";

const Page = ({ params }) => {

const { search } = params;

const [data, setData] = useState([]);

const [Loading, setLoading] = useState(false);

useEffect(() => {

(async () => {

setLoading(true);

const { data } = await axios.get(`/api/item/${search}`);

setData(data);

setLoading(false);

})();

}, [search]);

return (

<div>

{Loading ? (

<>

{/\* <h1 className="mx-5 p-3 text-2xl">Searching.............</h1> \*/}

<div className="flex-wrap flex justify-evenly w-fit">

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

<LoadingSkeletonWithOutImg />

</div>

</>

) : (

<>

<div className="flex-wrap gap-1 flex justify-evenly w-fit">

{data?.map((item, index) => (

<Card

key={index}

link\_text={item.link\_text}

link={item.link}

compare={item.Compare}

title={item.title}

rating={item.rating}

id={item.id}

price={item.price}

total\_people\_review={item.total\_people\_review}

delivery={item.delivery}

/>

))}

</div>

</>

)}

</div>

);

};

export default Page;

**app>search>compare>page.js**

/\* eslint-disable @next/next/no-img-element \*/

"use client";

import Table from "@/components/Table";

import DetailsLoading from "@/components/DetailsLoading";

import axios from "axios";

import React, { useEffect, useState } from "react";

import FavoriteIcon from "@/components/FavoriteIcon";

import { getDataFromToken } from "@/helpers/getDataFromToken";

const Page = ({ params }) => {

const [dataItem, setDataItem] = useState({});

const [loading, setLoading] = useState(false);

const [isFavorite, setIsFavorite] = useState(false);

const getData = async () => {

setLoading(true);

const { compare } = params;

const body = {

productId: compare,

};

const res = await axios.post("/api/favorites/check", body);

if (res.data.data) {

setIsFavorite(res.data.data);

}

const { data } = await axios.get(`/api/comparison/${compare}`);

setDataItem(data[0]);

setLoading(false);

};

useEffect(() => {

getData();

// eslint-disable-next-line react-hooks/exhaustive-deps

}, []);

const handleFavoriteClick = async () => {

try {

const { compare } = params;

const item = {

productId: compare,

productName: dataItem.title,

review: dataItem.review || "No review available",

reviewOutOf: dataItem.reviewOutOf || "No Data",

imgLink: dataItem.imgLink,

};

// Toggle favorite state locally

setIsFavorite((prev) => !prev);

// Send API request to add/remove item from favorites

if (!isFavorite) {

const res = await axios.post("/api/favorites/add", item);

toast.success(res.message);

} else {

const res = await axios.post("/api/favorites/remove", item);

toast.success(res.message);

}

} catch (error) {

console.error("Error handling favorite:", error);

// Handle error (e.g., show error message)

}

};

return (

<>

{loading ? (

<DetailsLoading />

) : (

<div className="relative overflow-x-auto min-h-screen shadow-md sm:rounded-lg">

<div className="flex p-2 text-gray-900 bg-white dark:text-white dark:bg-gray-800">

<span className="w-44 rounded-md bg-cover overflow-hidden">

<img src={dataItem.imgLink} alt="Product image." />

</span>

<div className="p-5 px-2 w-screen text-2xl font-semibold text-left text-gray-900 bg-white dark:text-white dark:bg-gray-800">

{dataItem.title}

<p className="mt-1 flex justify-between text-lg font-normal text-gray-500 dark:text-gray-400">

{dataItem.review}

{dataItem.reviewOutOf}

<span>

<FavoriteIcon

filled={isFavorite}

size={24}

onClick={handleFavoriteClick}

/>

</span>

</p>

</div>

</div>

<table className="w-full text-sm text-left text-gray-500 dark:text-gray-400">

<thead className="text-xs text-gray-700 uppercase bg-gray-50 dark:bg-gray-700 dark:text-gray-400">

<tr>

<th scope="col" className="px-6 py-3">

Sold By

</th>

<th scope="col" className="px-6 py-3">

Details

</th>

<th scope="col" className="px-6 py-3">

Item price

</th>

<th scope="col" className="px-6 py-3">

Total price

</th>

<th scope="col" className="px-6 py-3">

<span className="sr-only">Visit</span>

</th>

</tr>

</thead>

<tbody>

{dataItem?.content?.map((item, index) => (

<Table

key={index}

compBrand={item.compBrand}

compLink={item.compLink}

delivery={item.delivery}

itemPrice={item.itemPrice}

totalPrice={item.totalPrice}

/>

))}

</tbody>

</table>

</div>

)}

</>

);

};

export default Page;

**src>components>card.js**

import Link from "next/link";

import { GrCompare } from "react-icons/gr";

const Card = (props) => {

  return (

    <div className="w-full max-w-xs mt-4 relative bg-white border border-gray-200 rounded-lg shadow dark:bg-gray-800 dark:border-gray-700">

      <a href={props.link} target="\_blank">

        {/\* Image component Insert TODO \*/}

        <p className="bg-blue-500 h-3 rounded-b-2xl mb-2"></p>

      </a>

      <div className="px-5 pb-5">

        <a href={props.link} target="\_blank">

          <h5 className="text-xl font-semibold tracking-tight text-gray-900 dark:text-white">

            {props.title}

          </h5>

        </a>

        <div className="flex items-center mt-2.5 mb-2.5">

          <div className="flex items-center">

            <svg

              className="w-4 h-4 text-yellow-300 mr-1"

              aria-hidden="true"

              xmlns="http://www.w3.org/2000/svg"

              fill="currentColor"

              viewBox="0 0 22 20"

            >

              <path d="M20.924 7.625a1.523 1.523 0 0 0-1.238-1.044l-5.051-.734-2.259-4.577a1.534 1.534 0 0 0-2.752 0L7.365 5.847l-5.051.734A1.535 1.535 0 0 0 1.463 9.2l3.656 3.563-.863 5.031a1.532 1.532 0 0 0 2.226 1.616L11 17.033l4.518 2.375a1.534 1.534 0 0 0 2.226-1.617l-.863-5.03L20.537 9.2a1.523 1.523 0 0 0 .387-1.575Z" />

            </svg>

            {props.rating == "" ? (

              <code>No Data Available</code>

            ) : (

              <>

                <p className="ml-2 text-sm font-bold text-gray-900 dark:text-white">

                  {props.rating} / 5

                </p>

                <span className="w-1 h-1 mx-1.5 bg-gray-500 rounded-full dark:bg-gray-400"></span>

                <span className="text-sm font-medium text-gray-900 dark:text-white">

                  {props.total\_people\_review}

                </span>

              </>

            )}

          </div>

        </div>

        <code className="text-sm font-medium text-gray-900 dark:text-white mb-6 ml-0 mt-0">

          {props.link\_text}

          <br />

          <span className="text-xs">{props.delivery}</span>

        </code>

        <div>

          <div className="text-3xl font-bold text-gray-900 dark:text-white">

            {props.price}

          </div>

          {props.compare != null ? (

            <Link

              href={`/search/compare/${props.id}`}

              className="group text-white bg-blue-700 hover:bg-blue-800 font-medium rounded-tl-2xl rounded-tr-2xl rounded-bl-2xl text-sm px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 absolute bottom-0 right-0 flex items-center gap-2"

            >

              <GrCompare />

              <span className="inline-block transition-transform group-hover:translate-x-1 motion-reduce:transform-none">

                -&gt;

              </span>

            </Link>

          ) : (

            <a

              href={props.link}

              className="group text-white bg-blue-700 hover:bg-blue-800 font-medium rounded-tl-2xl rounded-tr-2xl rounded-bl-2xl text-sm px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 absolute bottom-0 right-0 flex items-center gap-2"

              target="\_blank"

            >

              <span className="inline-block transition-transform group-hover:translate-x-1 motion-reduce:transform-none">

                -&gt;

              </span>

            </a>

          )}

        </div>

      </div>

    </div>

  );

};

export default Card;

**app>users>login>route.js**

import { connect } from "@/dbConfig/dbConfig";

import User from "@/models/userModel";

import { NextResponse } from "next/server";

import bcryptjs from "bcryptjs";

import Redis from "ioredis";

import { SignJWT } from "jose";

connect();

const createToken = async (payload) => {

const secret = new TextEncoder().encode(process.env.TOKEN\_SECRET);

const alg = "HS256";

const jwt = await new SignJWT(payload)

.setProtectedHeader({ alg })

.setIssuedAt(new Date())

.setIssuer("urn:dealChecker:issuer")

.setAudience("urn:user:audience")

.setExpirationTime("1d")

.sign(secret);

return jwt;

};

export async function POST(request) {

try {

const redis = new Redis(process.env.REDIS\_URI);

const reqBody = await request.json();

const { email, password } = reqBody;

// Check if user exists in Redis cache

const cachedUser = await redis.get(email);

if (cachedUser) {

const user = JSON.parse(cachedUser);

// Validate password

const validPassword = await bcryptjs.compare(

password,

user.password

);

if (validPassword) {

// If password is valid, generate token and return response

const tokenData = {

id: user.\_id,

username: user.username,

email: user.email,

};

const token = await createToken(tokenData);

const response = NextResponse.json({

message: "Login successful",

success: true,

});

response.cookies.set("token", token, {

httpOnly: true,

});

return response;

}

}

//Check if user is exists

const user = await User.findOne({ email });

if (!user) {

return NextResponse.json(

{ error: "User not found" },

{ status: 400 }

);

}

//Check if password is correct

const validPassword = await bcryptjs.compare(password, user.password);

if (!validPassword) {

return NextResponse.json(

{ error: "Invalid Password" },

{ status: 400 }

);

}

// Cache user data in Redis

await redis.set(email, JSON.stringify(user));

//Create token data

const tokenData = {

id: user.\_id,

username: user.username,

email: user.email,

};

//Create token

const token = await createToken(tokenData);

//Sent token to user cookies

const response = NextResponse.json({

message: "Login successful",

success: true,

});

response.cookies.set("token", token, {

httpOnly: true,

});

return response;

} catch (error) {

return NextResponse.json({ error: error.message }, { status: 500 });

}

}

**app>users>signup>route.js**

import { connect } from "@/dbConfig/dbConfig";

import User from "@/models/userModel";

import { NextResponse } from "next/server";

import bcryptjs from "bcryptjs";

import { sendEmail } from "@/helpers/mailer";

connect();

export async function POST(request) {

try {

const reqBody = await request.json();

const { username, email, password } = reqBody;

//check if user already exists

const user = await User.findOne({ email });

if (user) {

return NextResponse.json(

{ error: "User Already Exists" },

{ status: 400 }

);

}

//Hash the password

const salt = await bcryptjs.genSalt(10);

const hashedPassword = await bcryptjs.hash(password, salt);

//Create a new user

const newUser = new User({

username,

email,

password: hashedPassword,

});

//Save the new user to the database

const savedUser = await newUser.save();

//send verification email

await sendEmail({ email, emailType: "VERIFY", userId: savedUser.\_id });

//Returning the response

return NextResponse.json({

message: "User created successfully",

success: true,

savedUser,

});

} catch (error) {

return NextResponse.json({ error: error.message }, { status: 500 });

}

}

**src>app>login>page.jsx**

"use client";

import Link from "next/link";

import React, { useEffect, useState } from "react";

import { useRouter } from "next/navigation";

import axios from "axios";

import toast from "react-hot-toast";

import {

Card,

CardContent,

CardDescription,

CardFooter,

CardHeader,

CardTitle,

} from "@/components/ui/card";

import { Button } from "@/components/ui/button";

import { Input } from "@/components/ui/input";

import { Label } from "@/components/ui/label";

import { Loader2 } from "lucide-react";

export default function LoginPage() {

const router = useRouter();

const [user, setUser] = useState({

email: "",

password: "",

});

const [loading, setLoading] = useState(false);

const onLogin = async () => {

try {

setLoading(true);

await axios.post("/api/users/login", user);

toast.success("Login Successfull");

router.push("/profile");

} catch (error) {

toast.error("Email or Password is incorrect", error.message);

} finally {

setLoading(false);

}

};

return (

<div className="min-h-screen flex flex-wrap flex-col items-center justify-center">

<Card className="w-[300px] sm:w-[350px]">

<CardHeader className="space-y-1">

<CardTitle className="text-2xl">Login</CardTitle>

<CardDescription>

Enter your email and password to login

</CardDescription>

</CardHeader>

<CardContent className="grid gap-4">

<div className="grid gap-2">

<Label htmlFor="email">Email</Label>

<Input

id="email"

type="email"

placeholder="m@example.com"

value={user.email}

onChange={(e) =>

setUser({ ...user, email: e.target.value })

}

/>

<Link

className="text-right text-xs text-slate-500 hover:underline hover:text-slate-300"

href="/forgotpassword"

>

Forgot Password

</Link>

</div>

<div className="grid gap-2">

<Label htmlFor="password">Password</Label>

<Input

id="password"

type="password"

value={user.password}

onChange={(e) =>

setUser({ ...user, password: e.target.value })

}

/>

</div>

</CardContent>

<CardFooter className="flex justify-between">

<Button

disabled={

user.email.length > 0 && user.password.length > 0

? false

: true

}

onClick={onLogin}

>

{loading ? (

<Loader2 className="mr-2 h-4 w-4 animate-spin" />

) : (

"Login"

)}

</Button>

<Link href="/signup">

<Button variant="outline">Signup</Button>

</Link>

</CardFooter>

</Card>

</div>

);

}

**src>app>signup>page.jsx**

"use client";

import Link from "next/link";

import React, { useEffect, useState } from "react";

import { useRouter } from "next/navigation";

import axios from "axios";

import toast from "react-hot-toast";

import {

Card,

CardContent,

CardDescription,

CardFooter,

CardHeader,

CardTitle,

} from "@/components/ui/card";

import { Button } from "@/components/ui/button";

import { Input } from "@/components/ui/input";

import { Label } from "@/components/ui/label";

import { Loader2 } from "lucide-react";

export default function SignupPage() {

const router = useRouter();

const [user, setUser] = useState({

email: "",

password: "",

username: "",

});

const [loading, setLoading] = useState(false);

const onSignup = async () => {

try {

setLoading(true);

await axios.post("/api/users/signup", user);

toast.success("Signup successfull");

toast("Please check your inbox and click on verification link.", {

duration: 10000,

});

router.push("/login");

} catch (error) {

toast.error(error.message);

} finally {

setLoading(false);

}

};

return (

<div className="min-h-screen flex flex-wrap flex-col items-center justify-center">

<Card className="w-[300px] sm:w-[350px]">

<CardHeader className="space-y-1">

<CardTitle className="text-2xl">Signup</CardTitle>

<CardDescription>

Enter your username, email and password to Signup

</CardDescription>

</CardHeader>

<CardContent className="grid gap-4">

<div className="grid gap-2">

<Label htmlFor="username">Username</Label>

<Input

id="username"

type="text"

placeholder="username"

value={user.username}

onChange={(e) =>

setUser({ ...user, username: e.target.value })

}

/>

</div>

<div className="grid gap-2">

<Label htmlFor="email">Email</Label>

<Input

id="email"

type="email"

placeholder="m@example.com"

value={user.email}

onChange={(e) =>

setUser({ ...user, email: e.target.value })

}

/>

</div>

<div className="grid gap-2">

<Label htmlFor="password">Password</Label>

<Input

id="password"

type="password"

value={user.password}

onChange={(e) =>

setUser({ ...user, password: e.target.value })

}

/>

</div>

</CardContent>

<CardFooter className="flex flex-col space-y-2">

<Button

className="w-full"

disabled={

user.username.length > 0 &&

user.email.length > 0 &&

user.password.length > 0

? false

: true

}

onClick={onSignup}

>

{loading ? (

<Loader2 className="mr-2 h-4 w-4 animate-spin" />

) : (

"Signup"

)}

</Button>

<Link className="w-full" href="/login">

<Button className="w-full" variant="outline">

Visit Login Page

</Button>

</Link>

</CardFooter>

</Card>

</div>

);

}

**src>app>profile>favorite>page.jsx**

"use client";

import React, { useEffect } from "react";

import CardWithImg from "@/components/CardWithImg";

import LoadingSkeleton from "@/components/LoadingSkeleton";

import axios from "axios";

import { useState } from "react";

const Page = () => {

const [data, setData] = useState([]);

const [loading, setLoading] = useState(false);

const getData = async () => {

setLoading(true);

const res = await axios.get("/api/favorites");

if (res.data.data.length > 0) {

console.log(res.data.data[0].imgLink);

setData(res.data.data);

}

setLoading(false);

};

useEffect(() => {

getData();

// eslint-disable-next-line react-hooks/exhaustive-deps

}, []);

return (

<div>

{loading ? (

<div className="flex-wrap flex justify-evenly w-fit">

{Array.from({ length: 10 }, (\_, index) => (

<LoadingSkeleton key={index} />

))}

</div>

) : data.length === 0 ? (

<div className="flex justify-center items-center text-3xl h-screen">

<p>No data available. Please add Favorite First.</p>

</div>

) : (

<div className="flex-wrap gap-1 flex justify-evenly w-fit">

{data.map((item, index) => (

<CardWithImg

key={index}

link={`/search/compare/${item.productId}`}

title={item.productName}

rating={item.review}

total\_people\_review={item.reviewOutOf}

img={item.imgLink}

/>

))}

</div>

)}

</div>

);

};

export default Page;

**package.json**

{

"name": "clg-project-final",

"version": "0.1.0",

"private": true,

"scripts": {

"dev": "next dev",

"build": "next build",

"start": "next start",

"lint": "next lint",

"test": "jest",

"test:watch": "jest --watchAll"

},

"dependencies": {

"@radix-ui/react-dropdown-menu": "^2.0.6",

"@radix-ui/react-hover-card": "^1.0.7",

"@radix-ui/react-label": "^2.0.2",

"@radix-ui/react-slot": "^1.0.2",

"@react-three/drei": "^9.105.4",

"@react-three/fiber": "^8.16.2",

"axios": "^1.6.0",

"bcryptjs": "^2.4.3",

"cheerio": "^1.0.0-rc.12",

"class-variance-authority": "^0.7.0",

"clsx": "^2.1.1",

"framer-motion": "^11.1.7",

"ioredis": "^5.4.1",

"jose": "^5.2.4",

"lucide-react": "^0.368.0",

"mongoose": "^8.3.2",

"next": "14.0.0",

"next-auth": "^4.24.7",

"next-themes": "^0.3.0",

"nodemailer": "^6.9.13",

"react": "^18",

"react-dom": "^18",

"react-hot-toast": "^2.4.1",

"react-icons": "^4.11.0",

"tailwind-merge": "^2.3.0",

"tailwindcss-animate": "^1.0.7",

"three": "^0.163.0",

"three-globe": "^2.31.0"

},

"devDependencies": {

"@testing-library/jest-dom": "^6.4.2",

"@testing-library/react": "^15.0.2",

"autoprefixer": "^10",

"eslint": "8.57.0",

"eslint-config-next": "14.2.1",

"jest": "^29.7.0",

"jest-environment-jsdom": "^29.7.0",

"postcss": "^8",

"supertest": "^6.3.4",

"tailwindcss": "^3"

}

}