**In the public folder**

**Index.html**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="utf-8" />**

**<meta name="viewport" content="width=device-width, initial-scale=1" />**

**<meta**

**name="description"**

**content="Web site created using create-react-app"**

**/>**

**<link**

**href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"**

**rel="stylesheet"**

**integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"**

**crossorigin="anonymous"**

**/>**

**<script**

**src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"**

**integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"**

**crossorigin="anonymous"**

**></script>**

**<link rel="icon" type="image/png" href="img/icon.jpg" />**

**<title>UrlShortsApp</title>**

**</head>**

**<body>**

**<noscript>You need to enable JavaScript to run this app.</noscript>**

**<div id="root"></div>**

**</body>**

**</html>**

**In the src folder**

**App.css**

**#homePage {**

**background-image: url("./images/homePageLogo.jpg");**

**object-fit: contain;**

**}**

**App.js**

**import "./App.css";**

**import { Route, Routes } from "react-router";**

**import UserProvider from "./context/UserProvider";**

**import { ToastContainer } from "react-toastify";**

**import "react-toastify/dist/ReactToastify.css";**

**import Navigation from "./components/Navigation";**

**import Home from "./components/Home";**

**import Signin from "./pages/public/views/Signin";**

**import Signup from "./pages/public/views/Signup";**

**import ForgotPassword from "./pages/public/views/ForgotPassword";**

**import ResetPassword from "./pages/public/views/ResetPassword";**

**import EmailActivation from "./pages/public/views/EmailActivation";**

**import Protected from "./components/Protected";**

**import Dashboard from "./pages/private/views/Dashboard";**

**import CreateShortUrl from "./pages/private/views/CreateShortUrl";**

**import TableDisplay from "./pages/private/views/TableDisplay";**

**function App() {**

**return (**

**<>**

**<UserProvider>**

**<Navigation />**

**<Routes>**

**<Route path="/" element={<Home />} />**

**<Route path="/signin" element={<Signin />} />**

**<Route path="/signup" element={<Signup />} />**

**<Route path="/forgotPassword" element={<ForgotPassword />} />**

**<Route**

**path="/resetPassword/:dataString"**

**element={<ResetPassword />}**

**/>**

**<Route**

**path="/emailActivation/:activationId"**

**element={<EmailActivation />}**

**/>**

**<Route path="\*" element={<Home />} />**

**<Route**

**path="/dashboard"**

**element={**

**<Protected redirect={<Home />}>**

**<Dashboard />**

**</Protected>**

**}**

**/>**

**<Route**

**path="/createUrlShorts"**

**element={**

**<Protected redirect={<Home />}>**

**<CreateShortUrl />**

**</Protected>**

**}**

**/>**

**<Route**

**path="/summary"**

**element={**

**<Protected redirect={<Home />}>**

**<TableDisplay />**

**</Protected>**

**}**

**/>**

**</Routes>**

**<ToastContainer />**

**</UserProvider>**

**</>**

**);**

**}**

**export default App;**

**index.css**

**body {**

**margin: 0;**

**font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", "Roboto", "Oxygen",**

**"Ubuntu", "Cantarell", "Fira Sans", "Droid Sans", "Helvetica Neue",**

**sans-serif;**

**-webkit-font-smoothing: antialiased;**

**-moz-osx-font-smoothing: grayscale;**

**}**

**code {**

**font-family: source-code-pro, Menlo, Monaco, Consolas, "Courier New",**

**monospace;**

**}**

**Index.js**

**import React from "react";**

**import ReactDOM from "react-dom";**

**import "./index.css";**

**import App from "./App";**

**import { BrowserRouter } from "react-router-dom";**

**ReactDOM.render(**

**<BrowserRouter>**

**<App />**

**</BrowserRouter>,**

**document.getElementById("root")**

**);**

**In the src folder under components subfolder**

**Home.js**

**import React from "react";**

**import { Link } from "react-router-dom";**

**import useAuth from "../hooks/useAuth";**

**const Home = () => {**

**const [, , authCheck] = useAuth();**

**return (**

**<>**

**<div**

**className="card my-3 mb-5 m-md-5 p-5 shadow border-dark"**

**id="homePage"**

**>**

**<div className="p-5 col">**

**<div className="lead display-1">Welcome To UrlShortsApp</div>**

**</div>**

**<div className="p-5 display-6 text-danger col">**

**<p>Here you can create shortUrl for a LongUrl</p>**

**</div>**

**</div>**

**<div className="container-fluid bg-dark p-5">**

**<div className="container p-5 d-flex justify-content-center align-items-center">**

**<Link**

**to={authCheck() ? "/dashboard" : "/signup"}**

**className="btn btn-outline-warning col-6 p-2"**

**>**

**Get Started**

**</Link>**

**</div>**

**</div>**

**</>**

**);**

**};**

**export default Home;**

**Navigation.js**

**import React from "react";**

**import { useNavigate } from "react-router";**

**import { Link, NavLink } from "react-router-dom";**

**import useAuth from "../hooks/useAuth";**

**const publicNavigationLinks = [**

**{ linkName: "Signin", linkTo: "/signin" },**

**{ linkName: "Signup", linkTo: "/signup" },**

**{ linkName: "ForgotPassword", linkTo: "/ForgotPassword" },**

**];**

**const protectedNavLinks = [**

**{ linkName: "Dashboard", linkTo: "/dashboard" },**

**{ linkName: "CreateShortUrl", linkTo: "/createUrlShorts" },**

**{ linkName: "Summary", linkTo: "/summary" },**

**];**

**const Navigation = () => {**

**const [, updateToken, authCheck] = useAuth();**

**const navigate = useNavigate();**

**return (**

**<>**

**<nav className="navbar navbar-expand-lg navbar-light shadow-sm">**

**<div className="container-fluid">**

**<Link className="navbar-brand text-success p-3" to="/">**

**UrlShortsApp**

**</Link>**

**<button**

**className="navbar-toggler"**

**type="button"**

**data-bs-toggle="collapse"**

**data-bs-target="#navbarNavAltMarkup"**

**aria-controls="navbarNavAltMarkup"**

**aria-expanded="false"**

**aria-label="Toggle navigation"**

**>**

**<span className="navbar-toggler-icon"></span>**

**</button>**

**<div**

**className="collapse navbar-collapse justify-content-end"**

**id="navbarNavAltMarkup"**

**>**

**<div className="navbar-nav">**

**{authCheck()**

**? protectedNavLinks.map(({ linkName, linkTo }) => {**

**return (**

**<NavLink**

**key={linkName}**

**className={(nav) =>**

**nav.isActive**

**? "active nav-link text-primary"**

**: "nav-link"**

**}**

**to={linkTo}**

**>**

**{linkName}**

**</NavLink>**

**);**

**})**

**: publicNavigationLinks.map(({ linkName, linkTo }) => {**

**return (**

**<NavLink**

**key={linkName}**

**className={(nav) =>**

**nav.isActive**

**? "active nav-link text-primary"**

**: "nav-link"**

**}**

**to={linkTo}**

**>**

**{linkName}**

**</NavLink>**

**);**

**})}**

**{authCheck() && (**

**<button**

**className="rounded btn btn-outline-danger mx-3"**

**onClick={() => {**

**updateToken(null);**

**navigate("/signin");**

**}}**

**>**

**Logout**

**</button>**

**)}**

**</div>**

**</div>**

**</div>**

**</nav>**

**</>**

**);**

**};**

**export default Navigation;**

**Protected.js**

**import useAuth from "../hooks/useAuth";**

**const Protected = ({ children, redirect }) => {**

**const [, , authCheck] = useAuth();**

**return authCheck() ? children : redirect;**

**};**

**export default Protected;**

**src/context/UserProvider.js**

**import React, { createContext } from "react";**

**import useGetValues from "../hooks/useGetValues";**

**export const UserContext = createContext(null);**

**const UserProvider = ({ children }) => {**

**const [tokenState, updateToken] = useGetValues("token");**

**const [userState, updateUser] = useGetValues("user");**

**const authCheck = () => {**

**return (tokenState && userState) !== null ? true : false;**

**};**

**const values = {**

**tokenState,**

**updateToken,**

**authCheck,**

**userState,**

**updateUser,**

**};**

**return <UserContext.Provider value={values}>{children}</UserContext.Provider>;**

**};**

**export default UserProvider;**

**src/context/extra.js**

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

**export const UserContext = createContext(null);**

**const useGetToken = () => {**

**const [tokenState, setTokenState] = useState(() => {**

**const initalValue = localStorage.getItem("token");**

**return JSON.parse(initalValue) || null;**

**});**

**useEffect(() => {**

**localStorage.setItem("token", JSON.stringify(tokenState));**

**}, [tokenState]);**

**const updateToken = (newValue) => {**

**return setTokenState(newValue);**

**};**

**return [tokenState, updateToken];**

**};**

**const useGetUser = () => {**

**const [userState, setUserState] = useState(() => {**

**const userObj = localStorage.getItem("user");**

**return userObj ? JSON.parse(userObj) : null;**

**});**

**useEffect(() => {**

**localStorage.setItem("user", JSON.stringify(userState));**

**}, [userState]);**

**const updateUser = (newValue) => {**

**return setUserState(newValue);**

**};**

**return [userState, updateUser];**

**};**

**const UserProvider = ({ children }) => {**

**const [tokenState, updateToken] = useGetToken();**

**const [userState, updateUser] = useGetUser();**

**const authCheck = () => {**

**return (tokenState && userState) !== null ? true : false;**

**};**

**const values = {**

**tokenState,**

**updateToken,**

**authCheck,**

**userState,**

**updateUser,**

**};**

**return <UserContext.Provider value={values}>{children}</UserContext.Provider>;**

**};**

**export default UserProvider;**

**src/helper/spinner.js**

**import React from "react";**

**const Spinner = () => {**

**return (**

**<div className="container m-5 p-5 position-absolute">**

**<div**

**className="d-flex justify-content-center align-items-center"**

**style={{ height: "250px" }}**

**>**

**<div**

**className="spinner-border"**

**role="status"**

**style={{ height: "4rem", width: "4rem" }}**

**></div>**

**</div>**

**</div>**

**);**

**};**

**export default Spinner;**

**src/hooks/useAuth.js**

**import { useContext } from "react";**

**import { UserContext } from "../context/UserProvider";**

**const useAuth = () => {**

**const user = useContext(UserContext);**

**return [user.tokenState, user.updateToken, user.authCheck];**

**};**

**export default useAuth;**

**src/hooks/ useGetValues.js**

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

**const useGetValues = (key) => {**

**const [state, setState] = useState(() => {**

**const initalValue = localStorage.getItem(key);**

**return JSON.parse(initalValue) || null;**

**});**

**useEffect(() => {**

**localStorage.setItem(key, JSON.stringify(state));**

**}, [state, key]);**

**const updateState = (newValue) => {**

**return setState(newValue);**

**};**

**return [state, updateState];**

**};**

**export default useGetValues;**

**src/hooks/ useUser.js**

**import { useContext } from "react";**

**import { UserContext } from "../context/UserProvider";**

**const useUser = () => {**

**const user = useContext(UserContext);**

**return [user.userState, user.updateUser];**

**};**

**export default useUser;**

**src/pages/private/components/DailyChart.js**

**import React from "react";**

**import {**

**LineChart,**

**Line,**

**XAxis,**

**YAxis,**

**CartesianGrid,**

**Tooltip,**

**Legend,**

**} from "recharts";**

**const MonthlyChart = ({ data }) => {**

**return (**

**<div className="col-12 col-md-5 p-2 border m-2">**

**<LineChart**

**width={450}**

**height={300}**

**data={data}**

**margin={{**

**top: 5,**

**right: 30,**

**left: 20,**

**bottom: 5,**

**}}**

**>**

**<CartesianGrid strokeDasharray="3 3" />**

**<XAxis dataKey="name" />**

**<YAxis />**

**<Tooltip />**

**<Legend />**

**<Line type="monotone" dataKey="day" stroke="#82ca9d" />**

**</LineChart>**

**</div>**

**);**

**};**

**export default MonthlyChart;**

**src/pages/private/components/MonthlyChart.js**

**import React from "react";**

**import {**

**LineChart,**

**Line,**

**XAxis,**

**YAxis,**

**CartesianGrid,**

**Tooltip,**

**Legend,**

**} from "recharts";**

**const MonthlyChart = ({ data }) => {**

**return (**

**<div className="col-12 col-md-5 p-2 border m-2">**

**<LineChart**

**width={450}**

**height={300}**

**data={data}**

**margin={{**

**top: 5,**

**right: 30,**

**left: 20,**

**bottom: 5,**

**}}**

**>**

**<CartesianGrid strokeDasharray="3 3" />**

**<XAxis dataKey="name" />**

**<YAxis />**

**<Tooltip />**

**<Legend />**

**<Line**

**type="monotone"**

**dataKey="month"**

**stroke="#8884d8"**

**activeDot={{ r: 8 }}**

**/>**

**</LineChart>**

**</div>**

**);**

**};**

**export default MonthlyChart;**

**src/pages/private/components/PageHeader.js**

**import React from "react";**

**const PageHeader = ({ email, name, textHeading }) => {**

**return (**

**<>**

**<div className="d-flex flex-column flex-md-row justify-content-between p-2 m-5">**

**<h2 className="display-4 p-3 m-4 card-body text-center text-info">**

**{textHeading}**

**</h2>**

**<span className=" shadow border p-3 m-2 d-flex flex-column justify-content-center">**

**<div className="lead text-primary">{email}</div>**

**<div className=" display-5 text-primary">{name}</div>**

**</span>**

**</div>**

**</>**

**);**

**};**

**export default PageHeader;**

**src/pages/private/components/ShortUrlFormCard.js**

**import React from "react";**

**const ShortUrlFormCard = ({**

**err,**

**success,**

**submitValue,**

**longUrlState,**

**setLongUrlState,**

**}) => {**

**return (**

**<div className="card p-4 m-3">**

**<form onSubmit={submitValue}>**

**<div className="mb-3">**

**<label htmlFor="longUrl" className="form-label">**

**LongUrl**

**</label>**

**<input**

**id="longUrl"**

**type="url"**

**className="form-control"**

**placeholder="type your long url here"**

**value={longUrlState}**

**onChange={(ev) => {**

**setLongUrlState(ev.target.value);**

**}}**

**/>**

**{err && <div className="form-text text-danger">{err}</div>}**

**{success && (**

**<div className="form-text text-success">url is Valid</div>**

**)}**

**</div>**

**<button className="btn btn-outline-secondary">Create Short Url</button>**

**</form>**

**</div>**

**);**

**};**

**export default ShortUrlFormCar**

**src/pages/private/components/TableCard.js**

**import React from "react";**

**const TableCard = ({ urlData }) => {**

**return (**

**<>**

**<div className="container p-4">**

**<div className="row">**

**<div className="col">**

**<table className="table table-striped">**

**<thead>**

**<tr>**

**<th scope="col">#</th>**

**<th scope="col">OriginalUrl</th>**

**<th scope="col">ShortUrl</th>**

**<th scope="col">CreatedAt</th>**

**<th scope="col">TotalHits</th>**

**</tr>**

**</thead>**

**<tbody>**

**{urlData.map((item, index) => (**

**<tr key={item.\_id}>**

**<th scope="row">{index + 1}</th>**

**<td>{item.originalUrl}</td>**

**<td>**

**https://karthee-urlshortsapp-000001.herokuapp.com/**

**{item.shortUrlId}**

**</td>**

**<td>{new Date(item.createdAt).toLocaleDateString()}</td>**

**<td**

**className={**

**item.hitCount === 0**

**? " text-center text-danger"**

**: " text-center text-success"**

**}**

**>**

**{item.hitCount}**

**</td>**

**</tr>**

**))}**

**</tbody>**

**</table>**

**</div>**

**</div>**

**</div>**

**</>**

**);**

**};**

**export default TableCard;**

**src/pages/private/components/TableCard.js**

**import React from "react";**

**const TableCard = ({ urlData }) => {**

**return (**

**<>**

**<div className="container p-4">**

**<div className="row">**

**<div className="col">**

**<table className="table table-striped">**

**<thead>**

**<tr>**

**<th scope="col">#</th>**

**<th scope="col">OriginalUrl</th>**

**<th scope="col">ShortUrl</th>**

**<th scope="col">CreatedAt</th>**

**<th scope="col">TotalHits</th>**

**</tr>**

**</thead>**

**<tbody>**

**{urlData.map((item, index) => (**

**<tr key={item.\_id}>**

**<th scope="row">{index + 1}</th>**

**<td>{item.originalUrl}</td>**

**<td>**

**https://karthee-urlshortsapp-000001.herokuapp.com/**

**{item.shortUrlId}**

**</td>**

**<td>{new Date(item.createdAt).toLocaleDateString()}</td>**

**<td**

**className={**

**item.hitCount === 0**

**? " text-center text-danger"**

**: " text-center text-success"**

**}**

**>**

**{item.hitCount}**

**</td>**

**</tr>**

**))}**

**</tbody>**

**</table>**

**</div>**

**</div>**

**</div>**

**</>**

**);**

**};**

**export default TableCard;**

**src/pages/private/functions/createShortUrl.js**

**import axios from "axios";**

**const createShortUrl = async (body, authorizationToken) => {**

**try {**

**const response = await axios.post(**

**process.env.REACT\_APP\_CREATESHORTURL\_API,**

**body,**

**{**

**headers: {**

**Authorization: `BEARER ${authorizationToken}`,**

**},**

**}**

**);**

**return response;**

**} catch (e) {**

**return e.response;**

**}**

**};**

**export default createShortUrl;**

**src/pages/private/functions/getAllUrlData.js**

**import axios from "axios";**

**const getAllUrlData = async (authorizationToken) => {**

**try {**

**const response = await axios.get(process.env.REACT\_APP\_GET\_ALLURL\_API, {**

**headers: {**

**Authorization: `BEARER ${authorizationToken}`,**

**},**

**});**

**return response;**

**} catch (e) {**

**return e.response;**

**}**

**};**

**export default getAllUrlData;**

**src/pages/private/functions/getAnalysisData.js**

**import axios from "axios";**

**const getAnalysisData = async (authorizationToken) => {**

**try {**

**const response = await axios.get(process.env.REACT\_APP\_URL\_ANALYSIS\_API, {**

**headers: {**

**Authorization: `BEARER ${authorizationToken}`,**

**},**

**});**

**return response;**

**} catch (e) {**

**return e.response;**

**}**

**};**

**export default getAnalysisData;**

**src/pages/private/views/CreateShortUrl.js**

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

**import { toast } from "react-toastify";**

**import { useNavigate } from "react-router";**

**import \* as yup from "yup";**

**import useAuth from "../../../hooks/useAuth";**

**import useUser from "../../../hooks/useUser";**

**import createShortUrl from "../functions/createShortUrl";**

**import PageHeader from "../components/PageHeader";**

**import ShortUrlFormCard from "../components/ShortUrlFormCard";**

**import Spinner from "../../../helper/Spinner";**

**const CreateShortUrl = () => {**

**const [longUrlState, setLongUrlState] = useState("");**

**const [err, setErr] = useState(false);**

**const [success, setSuccess] = useState(null);**

**const [loader, setLoader] = useState(false);**

**const navigate = useNavigate();**

**const [token, updateToken] = useAuth();**

**const [{ idActivated, email, name }] = useUser();**

**useEffect(() => {**

**if (!token) {**

**navigate("/");**

**}**

**if (!idActivated) {**

**navigate("/dashboard");**

**}**

**}, [idActivated, navigate, token]);**

**const getDataFunc = async (longUrl) => {**

**setLoader(true);**

**const body = {**

**longUrl,**

**};**

**const { data } = await createShortUrl(body, token);**

**setLoader(false);**

**if (data.type === "success") {**

**toast.success(data.msg);**

**navigate("/summary");**

**} else {**

**toast.error(data.msg);**

**navigate("/");**

**updateToken(null);**

**}**

**};**

**const yupValidation = yup.object().shape({**

**url: yup.string().url().required(),**

**});**

**const checkValidity = {**

**url: longUrlState,**

**};**

**const submitValue = (ev) => {**

**ev.preventDefault();**

**yupValidation**

**.validate(checkValidity)**

**.then((value) => {**

**setSuccess(value.url);**

**})**

**.catch(function (error) {**

**setErr(error.message);**

**});**

**if (success) {**

**getDataFunc(success);**

**}**

**setErr(false);**

**setSuccess(null);**

**};**

**if (loader) {**

**return <Spinner />;**

**}**

**return (**

**<div className="container-fluid ">**

**<PageHeader email={email} name={name} textHeading="Create ShortUrl" />**

**<div className="row mb-4 justify-content-center">**

**<div className="col-md-6">**

**<ShortUrlFormCard**

**err={err}**

**success={success}**

**submitValue={submitValue}**

**longUrlState={longUrlState}**

**setLongUrlState={setLongUrlState}**

**/>**

**</div>**

**</div>**

**</div>**

**);**

**};**

**export default CreateShortUrl;**

**src/pages/private/views/Dashboard.js**

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

**import { toast } from "react-toastify";**

**import { useNavigate } from "react-router";**

**import useAuth from "../../../hooks/useAuth";**

**import useUser from "../../../hooks/useUser";**

**import getAnalysisData from "../functions/getAnalysisData";**

**import PageHeader from "../components/PageHeader";**

**import MonthlyChart from "../components/MonthlyChart";**

**import DailyChart from "../components/DailyChart";**

**import Spinner from "../../../helper/Spinner";**

**const Dashboard = () => {**

**const [loader, setLoader] = useState(false);**

**const [chartData, setChartData] = useState(null);**

**const navigate = useNavigate();**

**const [token, updateToken] = useAuth();**

**const [{ idActivated, email, name }] = useUser();**

**useEffect(() => {**

**if (!token) {**

**navigate("/");**

**}**

**const getAllData = async () => {**

**setLoader(true);**

**const { data } = await getAnalysisData(token);**

**setLoader(false);**

**if (data.type === "success") {**

**toast.success(data.msg);**

**const arr = data.dataArray.map((item) => {**

**return {**

**name: item.urlName,**

**month: parseInt(item.month),**

**day: parseInt(item.day),**

**};**

**});**

**setChartData(arr);**

**} else {**

**toast.error(data.msg);**

**navigate("/");**

**updateToken(null);**

**}**

**};**

**if (idActivated) {**

**getAllData();**

**}**

**}, [idActivated, navigate, token, updateToken]);**

**if (!idActivated) {**

**return (**

**<>**

**<div className="card d-flex justify-content-center align-items-center m-5 p-5">**

**<h3 className="lead text-warning display-2 p-5">**

**Sorry your account is not activated...kindly Activate.**

**</h3>**

**</div>**

**</>**

**);**

**}**

**if (loader) {**

**return <Spinner />;**

**}**

**return (**

**<div className="container-fluid">**

**<PageHeader email={email} name={name} textHeading="DashBoard" />**

**{chartData && (**

**<div className="row my-5 justify-content-center">**

**<MonthlyChart data={chartData} />**

**<DailyChart data={chartData} />**

**</div>**

**)}**

**</div>**

**);**

**};**

**export default Dashboard;**

**src/pages/private/views/TableDisplay.js**

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

**import { toast } from "react-toastify";**

**import { useNavigate } from "react-router";**

**import useAuth from "../../../hooks/useAuth";**

**import useUser from "../../../hooks/useUser";**

**import getAllUrlData from "../functions/getAllUrlData";**

**import PageHeader from "../components/PageHeader";**

**import TableCard from "../components/TableCard.js";**

**import Spinner from "../../../helper/Spinner";**

**const TableDisplay = () => {**

**const navigate = useNavigate();**

**const [urlData, seturlData] = useState(null);**

**const [loader, setLoader] = useState(false);**

**const [token, updateToken] = useAuth();**

**const [{ idActivated, email, name }] = useUser();**

**useEffect(() => {**

**if (!token) {**

**navigate("/");**

**}**

**const getDataFunc = async () => {**

**setLoader(true);**

**const { data } = await getAllUrlData(token);**

**if (data.type === "success") {**

**toast.success(data.msg);**

**seturlData(data.allUrls);**

**setLoader(false);**

**} else {**

**toast.error(data.msg);**

**navigate("/");**

**updateToken(null);**

**}**

**};**

**if (idActivated) {**

**getDataFunc();**

**} else {**

**navigate("/dashboard");**

**}**

**}, [idActivated, navigate, token, updateToken]);**

**if (loader) {**

**return <Spinner />;**

**}**

**return (**

**<div className="container-fluid ">**

**<PageHeader email={email} name={name} textHeading="Table Summary" />**

**{urlData && <TableCard urlData={urlData} />}**

**</div>**

**);**

**};**

**export default TableDisplay;**