**Cover page**

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

import axios from "axios";

import CoverTemplate from "./covercardtemplate";

import TopDownloadTemplate from "./topdownloadtemplate";

import Modal from "react-bootstrap/Modal";

import InputRange from "react-input-range";

import { data, post } from "jquery";

import { Maximize } from "react-feather";

export default function MusicCoverPage() {

const [modelOpen, setmodelOpen] = useState(false);

const [pricerange, setPriceRange] = useState("0");

const [downloadrange, setDownloadRange] = useState("0");

const [nodata, setNoData] = useState("");

const [covers, setCovers] = useState([]);

const [filtercover, setfilterCover] = useState([]);

const [categorycover, setCategoryCover] = useState([]);

// const [filtercovers, setFilterCovers] = useState([]);

const [categorytext, setCategoryText] = useState("All");

const [serchvalue, setSerchvalue] = useState("");

const [categories, setCategories] = useState([]);

const [populercover, setpopulercover] = useState([]);

const [populerimage, setpopulerimage] = useState();

let dataholdedr = [];

let pcover = {};

let max = 0;

useEffect(async () => {

document.getElementById("bufferlink").style.display = "block";

document.getElementById("link").style.display = "none";

document.getElementById("link1").style.display = "none";

document.getElementById("spinnerdiv").style.display = "block";

document.getElementById("coverdiv").style.display = "none";

document.getElementById("spinnerdiv2").style.display = "block";

document.getElementById("topcover").style.display = "none";

await axios

.get("http://localhost:8070/covers/getcoverbymaincover")

.then((res) => {

dataholdedr = res.data;

setCovers(res.data);

setfilterCover(res.data);

setCategoryCover(res.data);

setCovers(res.data);

document.getElementById("spinnerdiv").style.display = "none";

document.getElementById("coverdiv").style.display = "block";

// console.log(res.data);

})

.catch((err) => {

alert(err.message);

});

// Getting sub categories

axios

.get("http://localhost:8070/mainCategory/get")

.then((res) => {

setCategories(res.data[0].SubCategories);

document.getElementById("bufferlink").style.display = "none";

document.getElementById("link").style.display = "block";

document.getElementById("link1").style.display = "block";

// console.log(res.data[0].SubCategories);

})

.catch((err) => {

alert(err.message);

});

populercovers();

}, []);

async function populercovers() {

// console.log(dataholdedr);

for (let i = 0; i < dataholdedr.length; i++) {

// console.log(dataholdedr[i].NoOfDownloads)

if (Number(dataholdedr[i].NoOfDownloads) >= max) {

max = dataholdedr[i].NoOfDownloads;

console.log(max);

pcover = dataholdedr[i];

}

}

// console.log(pcover.PreviewPages[0]);

setpopulercover(pcover);

setpopulerimage(pcover.PreviewPages[0]);

console.log(pcover);

document.getElementById("spinnerdiv2").style.display = "none";

document.getElementById("topcover").style.display = "block";

}

function modalopen() {

// alert("This is alert");

setmodelOpen(true);

}

function modalClose() {

setmodelOpen(false);

}

function sortByPrice(v) {

setPriceRange(v);

// let dataholder = covers

let filtercovers = categorycover;

let result = filtercovers.filter((post) => Number(post.Price) >= v);

if (result.length != 0) {

setCovers(result);

setNoData("");

} else {

setNoData("No Covers available");

setCovers([]);

}

setDownloadRange(0);

}

function sortByDownloads(v) {

setPriceRange(0);

setDownloadRange(v);

let filtercovers = categorycover;

let result = filtercovers.filter((post) => Number(post.NoOfDownloads) >= v);

if (result.length != 0) {

setCovers(result);

setNoData("");

} else {

setNoData("No Covers available");

setCovers([]);

}

}

async function searchByName(val) {

setSerchvalue(val);

document.getElementById("spinnerdiv").style.display = "block";

document.getElementById("coverdiv").style.display = "none";

await axios

.get("http://localhost:8070/covers/getcoverbymaincover")

.then((res) => {

dataholdedr = res.data;

setCovers(res.data);

setfilterCover(res.data);

document.getElementById("spinnerdiv").style.display = "none";

document.getElementById("coverdiv").style.display = "block";

let searchResult = filtercover.filter(

(post) =>

post.Title.toLowerCase().includes(val.toLowerCase()) ||

post.OriginalArtistName.toLowerCase().includes(val.toLowerCase())

);

if (searchResult.length != 0) {

setCovers(searchResult);

setNoData("");

} else {

setNoData("No Covers available");

setCovers([]);

}

// console.log(res.data);

})

.catch((err) => {

alert(err.message);

});

}

async function fetchData(type) {

document.getElementById("spinnerdiv").style.display = "block";

document.getElementById("coverdiv").style.display = "none";

await axios

.get("http://localhost:8070/covers/getcoverbymaincover")

.then((res) => {

dataholdedr = res.data;

setCovers(res.data);

setfilterCover(res.data);

document.getElementById("spinnerdiv").style.display = "none";

document.getElementById("coverdiv").style.display = "block";

if (type == "All") {

setCategoryText("All");

setCategoryCover(filtercover);

setCovers(filtercover);

setPriceRange(0);

if (filtercover.length != 0) {

setNoData("");

} else {

setNoData("No Covers available");

setCovers([]);

}

setDownloadRange(0);

} else {

setCategoryText(type);

let filtercovers = filtercover;

let result = filtercovers.filter((post) =>

post.SubCategory.includes(type)

);

setCovers(result);

setCategoryCover(result);

setPriceRange(0);

setDownloadRange(0);

setNoData("");

if (result.length != 0) {

setCovers(result);

setNoData("");

} else {

setNoData("No Covers available");

setCovers([]);

}

}

// console.log(res.data);

})

.catch((err) => {

alert(err.message);

});

}

return (

<div>

<br />

<div className="">

<center>

<div className="col-md-11 input-group">

<input

type="text"

class="form-control"

placeholder="Search Music Covers"

onChange={(e) => searchByName(e.target.value)}

/>

<div class="input-group-append">

<button className="input-group-text">

<svg

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

width="16"

height="16"

fill="currentColor"

class="bi bi-search"

viewBox="0 0 16 16"

>

<path d="M11.742 10.344a6.5 6.5 0 1 0-1.397 1.398h-.001c.03.04.062.078.098.115l3.85 3.85a1 1 0 0 0 1.415-1.414l-3.85-3.85a1.007 1.007 0 0 0-.115-.1zM12 6.5a5.5 5.5 0 1 1-11 0 5.5 5.5 0 0 1 11 0z" />

</svg>

</button>

</div>

</div>

</center>

</div>

<br />

<div class="">

<div className="row ">

{/\* left side of the page \*/}

<div className="col-md-1"></div>

<div className="col-md-3">

<br />

<div

className="list-group"

style={{ maxHeight: "30%", overflowY: "scroll" }}

>

<a

style={{ display: "block" }}

id="bufferlink"

href="#"

className="list-group-item list-group-item-action"

>

&emsp; Loading...

</a>

<a

href="#"

id="link1"

style={{ display: "none" }}

onClick={() => fetchData("All")}

className="list-group-item list-group-item-action"

>

<i className="fa fa-music"></i>&emsp; All

</a>

<span id="link">

{categories.map((post) => (

<a

href="#"

id="link"

style={{ display: "block" }}

onClick={() => fetchData(post)}

className="list-group-item list-group-item-action"

>

<i className="fa fa-music"></i>&emsp; {post}

</a>

))}

</span>

</div>

<div>

<br />

<div className="row text-center">

<div className="col-md-5">

<label>Price Range </label>

<label>({pricerange} - 200+) </label>

<div class="slidecontainer">

<input

id="typeinp"

type="range"

min="0"

max="200"

value={pricerange}

onChange={(e) => sortByPrice(e.target.value)}

step="1"

/>

</div>

</div>

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

<label> Downloads Range </label>

<label>({downloadrange} - 200+) </label>

<div class="slidecontainer">

<input

id="typeinp"

type="range"

min="0"

max="200"

value={downloadrange}

onChange={(e) => sortByDownloads(e.target.value)}

step="1"

/>

</div>

</div>

</div>

<div className="row">

<h4 style={{ color: "#764A34" }}>

<strong>

<center>Most Downloaded Classical Guitar Cover</center>

</strong>

</h4>

</div>

<center>

<div

id="spinnerdiv2"

class="col-lg-8 "

style={{ display: "block" }}

>

<br />

<div class=" justify-content-center">

<div class="spinner-border" role="status">

<span class="sr-only">Loading...</span>

</div>

</div>

</div>

</center>

<div id="topcover" style={{ display: "none" }}>

<TopDownloadTemplate

title={populercover.Title}

price={populercover.Price}

artist={populercover.OriginalArtistName}

maincat={populercover.MainCategory}

subcat={populercover.SubCategory}

id="topcover"

imageName={populerimage}

/>

</div>

</div>

<br />

</div>

{/\* right side of the page \*/}

<div className="col-md-8">

<h4 style={{ color: "#764A34" }}>

<strong>Classical Guitar Covers - {categorytext}</strong>

</h4>

<center>

<h4 style={{ color: "red" }}>{nodata}</h4>

</center>

<div className="row">

<div

id="spinnerdiv"

class="col-lg-8 "

style={{ display: "block" }}

>

<br />

<center>

<div class=" justify-content-center">

<div class="spinner-border" role="status">

<span class="sr-only">Loading...</span>

</div>

</div>

</center>

</div>

</div>{" "}

<span id="coverdiv">

<div className="row">

{covers.map((post, index) => (

// console.log(post.PreviewPages[0]),

<div className="col-md-4" onClick={() => modalopen()}>

<CoverTemplate

title={post.Title}

artist={post.OriginalArtistName}

price={post.Price}

category={post.SubCategory}

id={index}

imageName={post.PreviewPages[0]}

/>

<br />

</div>

))}

<br />

<br />

</div>

</span>

</div>

</div>

</div>

</div>

);

}

**Cart**

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

import axios from "axios";

import Swal from "sweetalert2";

import { storage } from "../../Configurations/firebaseConfigurations";

import { ref, uploadBytesResumable, getDownloadURL } from "@firebase/storage";

export default function MusicCart(props) {

let [cartCovers, setCovers] = useState([]);

let [total, setTotal] = useState("");

let [emptyText, setEmptyText] = useState("");

let [dataholder, setDataholder] = useState([]);

let tot = 0;

let coverdetails = [];

let price = "";

let title = "";

let author = "";

let id = "";

let images = [];

let cover = {

id,

price,

title,

author,

images,

};

useEffect(async () => {

document.getElementById("spinnerdiv").style.display = "block";

document.getElementById("cartdiv").style.display = "none";

setTotal("Loading...");

await axios

.get(

"http://localhost:8070/shoppingCart/getOneCart/61a26e4cb42a52e3ff12e82e"

)

.then((res) => {

console.log(res.data.CoverIDs);

if (res.data.CoverIDs != "") {

setDataholder(res.data.CoverIDs);

// console.log(res.data);

setTotal("Loading...");

callData(res.data.CoverIDs);

} else {

document.getElementById("spinnerdiv").style.display = "none";

document.getElementById("cartdiv").style.display = "block";

setTotal("0");

setEmptyText("No covers Available In the cart");

}

})

.catch((err) => {

alert(err.message);

});

}, []);

async function callData(data) {

console.log(data.length);

console.log(data);

for (let i = 0; i < data.length; i++) {

await axios

.get(`http://localhost:8070/covers/getcoverbyid/${data[i]}`)

.then((res) => {

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

// console.log("asd")

cover = {

price: res.data[0].Price,

title: res.data[0].Title,

id: res.data[0].\_id,

author: res.data[0].OriginalArtistName,

images: res.data[0].PreviewPages,

};

tot = Number(tot) + Number(res.data[0].Price);

coverdetails.push(cover);

// console.log(cover);

})

.catch((err) => {

alert(err.message);

});

}

console.log(coverdetails);

setCovers(coverdetails);

document.getElementById("spinnerdiv").style.display = "none";

document.getElementById("cartdiv").style.display = "block";

setTotal(tot);

}

function removeBtn(id) {

Swal.fire({

title: "Are you sure?",

text: "You won't be able to revert this!",

icon: "warning",

showCancelButton: true,

confirmButtonColor: "#3085d6",

cancelButtonColor: "#d33",

confirmButtonText: "Yes, delete it!",

}).then((result) => {

if (result.isConfirmed) {

axios

.delete(

`http://localhost:8070/shoppingCart/deleteCartCover/${id}/61a26e4cb42a52e3ff12e82e`

)

.then((res) => {

alert("Successfully deleted");

})

.catch((err) => {

alert(err.message);

});

window.location.reload(true);

}

});

}

async function displayImages(coverImageName, index) {

// console.log(coverImageName);

// console.log(index);

const storageRef = ref(storage, `PreviewImages/${coverImageName}`);

await getDownloadURL(storageRef)

.then((url) => {

document.getElementById(index).src = url;

})

.catch((err) => {

// ErrorhandlingTxt("Reccomended covers are not available right now!");

});

}

return (

<div className="container">

<div

style={{ alignContent: "center" }}

className="d-flex justify-content-center mt-2"

>

<h2

class="mb-4"

style={{

alignContent: "center",

fontWeight: "bold",

color: "#764A34",

}}

>

My Shopping Carts

</h2>

</div>

<br />

<div class="row">

<div id="spinnerdiv" class="col-lg-8 " style={{ display: "block" }}>

<center>

<div class=" justify-content-center">

<div class="spinner-border" role="status">

<span class="sr-only">Loading...</span>

</div>

</div>

</center>

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

</div>

<div

class="col-lg-8"

id="cartdiv"

style={{

overflowY: "scroll",

display: "none",

}}

>

<div>

<center>

<h4 style={{ color: "red" }}>{emptyText}</h4>

</center>

{cartCovers.map((post, index) => (

console.log(post.title),

<div class="card mb-3">

<div class="row no-gutters">

<div class="col-md-4 mt-3 clsImg ">

<img

alt="Card image cap"

class="card-img-top embed-responsive-item"

id={index}

src={

displayImages(post.images[0], index) ||

"/images/Imageplaceholder.png"

}

/>

</div>

<div class="col-md-8">

<div class="card-body ">

<h5 style={{ fontWeight: "bold" }}>{post.title}</h5>

<p

class="mb-0 text-uppercase small"

style={{ color: "#764A34", fontWeight: "bold" }}

>

{`Original Artist :`}{" "}

<span style={{ color: "#000000" }}>{post.author}</span>

</p>

<p

class="mb-0 text-uppercase small"

style={{ color: "#764A34", fontWeight: "bold" }}

>

{`Arranged by :`}{" "}

<span style={{ color: "#000000" }}>

Kaushal Rashmika

</span>

</p>

<p class="mb-0" style={{ fontWeight: "bold" }}>

Rs.{post.price}

</p>

<button

type="button"

class="btn "

onClick={() => removeBtn(post.id)}

style={{

backgroundColor: "#D0193A",

color: "white",

}}

>

<i class="fas fa-trash-alt mr-1"></i>Remove item

</button>

</div>

</div>

</div>

</div>

))}

</div>

</div>

{/\* Cart Calculations \*/}

<div class="col-lg-4 justify-content-end">

<div class="mb-6 ">

<div class="pt-3">

<h5 class="mb-3" style={{ fontWeight: "bold" }}>

The Total Amount of

</h5>

<ul class="list-group list-group-flush">

{/\* Temporary Amount \*/}

<li class="list-group-item d-flex justify-content-between align-items-center border-0 px-0 pb-0">

Temporary Amount: ${total}

</li>

{/\* Sub Total \*/}

<li class="list-group-item d-flex justify-content-between align-items-center border-0 px-0 mb-3">

<div>

<strong>Sub Total: ${total}</strong>

</div>

</li>

</ul>

{/\* Checkout Button \*/}

<button

type="button"

class="btn btn-block"

style={{ backgroundColor: "#279B14", color: "white" }}

>

<i class="bi bi-arrow-right"></i>

Go to Checkout

</button>

<br />

<br />

</div>

</div>

</div>

</div>

</div>

);

}

**DashBoard**

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

import axios from "axios";

import MusicCoverPage from "./../customerRole/musiccoverpage";

import AdminCard from "./../adminRole/admincard";

import Modal from "react-bootstrap/Modal";

import { Eye } from "react-feather";

import $ from "jquery";

import DataTable from "datatables.net";

import CoverTemplate from "../customerRole/covercardtemplate";

// Admin dashboard

export default function DashBoard() {

const [modelOpen, setmodelOpen] = useState(false);

const [coverlength, setCoverLength] = useState("Loading...");

const [customerlength, setCustomersLength] = useState("Loading...");

const [orders, setOrders] = useState([]);

const [total, setTotal] = useState("Loading...");

const [paymentdata, setPaymentData] = useState([]);

const [selectedCovers, setSelectedCovers] = useState([]);

const [feedbackLength, setFeedbackLength] = useState("Loading...");

const [alldownloads, setDownloads] = useState("Loading...");

const [fromValue, setFromValue] = useState("");

const [toValue, setToValue] = useState("");

const [error, setError] = useState("");

let tableData = [];

let covers = [];

let customerdetails = [];

let tot = 0;

let downloads = 0;

let name = "";

let totalprice = "";

let date = "";

let cartID = "";

let customerID = "";

let coverID = [];

let orderHolder = [];

let dataholder = {

name,

totalprice,

date,

cartID,

customerID,

coverID,

};

useEffect(() => {

document.getElementById("spinnerdiv").style.display = "block";

document.getElementById("maindiv").style.display = "none";

axios

.get("http://localhost:8070/covers/getcovers")

.then((res) => {

setCoverLength(res.data.length);

calculateDownloads(res.data);

})

.catch((err) => {

alert(err.message);

});

axios

.get("http://localhost:8070/feedback/getAllFeedback")

.then((res) => {

setFeedbackLength(res.data.length);

})

.catch((err) => {

alert(err.message);

});

axios

.get("http://localhost:8070/customer/getAll")

.then((res) => {

setCustomersLength(res.data.length);

})

.catch((err) => {

alert(err.message);

});

axios

.get("http://localhost:8070/order/getOrders")

.then((res) => {

setOrders(res.data);

// console.log(res.data);

orderHolder = res.data;

loadIncome(res.data);

setError("");

})

.catch((err) => {

alert(err.message);

});

}, []);

function modalClose() {

setmodelOpen(false);

}

async function loadIncome(data) {

console.log(data.length);

if (data.length > 0) {

await data.map(

(post) => (

(tot = tot + Number(post.TotalPrice)), loadCovers(post.CustomerID)

)

);

setTotal(tot);

setError("");

}

}

async function loadCovers(customerid) {

// console.log(customerid);

await axios

.get(`http://localhost:8070/customer/get/${customerid}`)

.then((res) => {

// console.log(res.data);

customerdetails.push(res.data);

})

.catch((err) => {

alert(err.message);

});

assignData();

}

async function assignData() {

if (customerdetails.length == orderHolder.length) {

// console.log(customerdetails);

// console.log(orderHolder);

for (let i = 0; i < orderHolder.length; i++) {

for (let j = 0; j < orderHolder.length; j++) {

if (orderHolder[i].CustomerID == customerdetails[j].\_id) {

// console.log(customerdetails[j]);

// console.log(orderHolder[i]);

dataholder = {

name: customerdetails[j].FirstName,

totalprice: orderHolder[i].TotalPrice,

date: orderHolder[i].TransactionDateAndTime,

cartID: orderHolder[i].\_id,

customerID: customerdetails[j].\_id,

coverID: orderHolder[i].CoverIDs,

};

tableData.push(dataholder);

break;

}

}

}

console.log(tableData);

setPaymentData(tableData);

document.getElementById("spinnerdiv").style.display = "none";

document.getElementById("maindiv").style.display = "block";

$("#example").DataTable();

}

}

async function getOrderedCovers(data) {

console.log(data);

for (let k = 0; k < data.length; k++) {

setmodelOpen(true);

await axios

.get(`http://localhost:8070/covers/getcoverbyid/${data[k]}`)

.then((res) => {

document.getElementById("spinnerdiv2").style.display = "block";

document.getElementById("modeldiv").style.display = "none";

covers.push(res.data[0]);

})

.catch((err) => {

console.log(err.message);

alert(err.message);

});

}

// testFunction();

console.log(covers);

setSelectedCovers(covers);

document.getElementById("spinnerdiv2").style.display = "none";

document.getElementById("modeldiv").style.display = "block";

}

function calculateDownloads(data) {

for (let i = 0; i < data.length; i++) {

downloads = downloads + Number(data[i].NoOfDownloads);

}

// console.log(downloads);

setDownloads(downloads);

}

function searchByDate(fromDate, toDate) {

setFromValue(fromDate);

setToValue(toDate);

if (fromDate != "" && toDate != "") {

document.getElementById("spinnerdiv").style.display = "block";

document.getElementById("maindiv").style.display = "none";

axios

.get(`http://localhost:8070/order/getbyyear/${fromDate}/${toDate}`)

.then((res) => {

if (res.data.length > 0) {

setOrders(res.data);

console.log(res.data);

orderHolder = res.data;

loadIncome(res.data);

document.getElementById("spinnerdiv").style.display = "none";

document.getElementById("maindiv").style.display = "block";

} else {

setError("No Data available");

setPaymentData([]);

setTotal("0");

document.getElementById("spinnerdiv").style.display = "none";

document.getElementById("maindiv").style.display = "block";

}

})

.catch((err) => {

alert(err.message);

});

}

}

function refresh() {

document.getElementById("spinnerdiv").style.display = "block";

document.getElementById("maindiv").style.display = "none";

axios

.get("http://localhost:8070/order/getOrders")

.then((res) => {

setOrders(res.data);

// console.log(res.data);

orderHolder = res.data;

loadIncome(res.data);

document.getElementById("spinnerdiv").style.display = "none";

document.getElementById("maindiv").style.display = "block";

})

.catch((err) => {

alert(err.message);

});

}

return (

<div>

{/\* {console.log(selectedCovers)} \*/}

<div className="row">

<div className="col-md-2">

{" "}

<AdminCard

color="red"

value="Covers"

num={coverlength}

icon={"fa fa-music"}

/>

<br />

</div>

<div className="col-md-2">

{" "}

<AdminCard

color="#8B8000"

value="Excercises"

icon={"fa fa-file"}

num={coverlength}

/>

<br />

</div>

<div className="col-md-2">

{" "}

<AdminCard

color="blue"

icon={"fa fa-user"}

value="Users"

num={customerlength}

/>

<br />

</div>

<div className="col-md-2">

{" "}

<AdminCard

color="#7CB9E8 "

icon={"fa fa-download"}

value="Downloads"

num={alldownloads}

/>

<br />

</div>

<div className="col-md-2">

{" "}

<AdminCard

color="#54626F "

icon={"fa fa-comments"}

value="Feedbacks"

num={feedbackLength}

/>

<br />

</div>

<div className="col-md-2">

{" "}

<AdminCard

color="green"

icon={"fas fa-dollar-sign"}

value="Income"

num={total}

/>

<br />

</div>

</div>

<hr />

<div id="spinnerdiv" style={{ display: "block" }}>

<center>

<div className=" justify-content-center">

<div className="spinner-border" role="status">

<span className="sr-only">Loading...</span>

</div>

</div>

</center>

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

<br />

</div>

<div

className="container"

id="maindiv"

style={{ overflowX: "scroll", display: "none" }}

>

<div className="row">

<div className="col-md-4">

<div class="form-group">

<label>From</label>

<input

type="date"

class="form-control"

id="from"

value={fromValue}

onChange={(e) => searchByDate(e.target.value, toValue)}

aria-describedby="emailHelp"

placeholder="Enter email"

/>

</div>{" "}

</div>

<div className="col-md-4">

<div class="form-group">

<label>To</label>

<input

type="date"

class="form-control"

id="to"

value={toValue}

onChange={(e) => searchByDate(fromValue, e.target.value)}

aria-describedby="emailHelp"

placeholder="Enter email"

/>

</div>{" "}

</div>

<div className="col-md-4">

<button

type="button"

className="btn btn-outline-danger btn-sm"

data-mdb-ripple-color="dark"

onClick={(e) => refresh()}

style={{ marginTop: "35px" }}

>

Clear Filters

</button>

</div>

</div>

<h4 style={{ color: "red" }}>

<strong>{error}</strong>

</h4>

<table

id="example"

class="table table-striped table-bordered"

style={{ width: "100%" }}

>

<thead>

<tr>

<th>Customer Name</th>

<th>Total Price</th>

<th>Date</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

{paymentdata.map((post) => (

<tr>

<td>{post.name}</td>

<td className="text-right">Rs.{post.totalprice}.00</td>

<td>{post.date.substr(0, 10)}</td>

<td>

{" "}

<button

type="button"

onClick={() => getOrderedCovers(post.coverID)}

class="btn btn-outline-secondary waves-effect btn-sm px-2"

>

<i class="fas fa-eye" aria-hidden="true"></i>

</button>

</td>

</tr>

))}

</tbody>

<tfoot>

<tr>

<th>Customer Name</th>

<th className="text-right">Total Price :- Rs.{total}.00</th>

<th>Date</th>

<th>Actions</th>

</tr>

</tfoot>

</table>

</div>

<Modal show={modelOpen} size="lg">

<Modal.Header>

<h4>All ordered covers</h4>

<button

type="button"

class="close"

data-dismiss="modal"

aria-label="Close"

onClick={modalClose}

>

<span aria-hidden="true">&times;</span>

</button>

</Modal.Header>

<Modal.Body>

<div className="container">

<div id="spinnerdiv2" style={{ display: "block" }}>

<center>

<div class=" justify-content-center">

<div class="spinner-border" role="status">

<span class="sr-only">Loading...</span>

</div>

</div>

</center>

</div>

<div

style={{ display: "none" }}

id="modeldiv"

className="row text-center"

>

<table

id="example"

class="table table-striped table-bordered"

style={{ width: "100%" }}

>

<thead>

<tr>

<th>Cover Name</th>

<th>Total Price</th>

<th>Total Downloads</th>

</tr>

</thead>

<tbody>

{selectedCovers.map((post) => (

<tr>

<td>{post.Title}</td>

<td className="text-right">Rs.{post.Price}.00</td>

<td>{post.NoOfDownloads}</td>

</tr>

))}

</tbody>

<tfoot>

<tr>

<th>Cover Name</th>

<th>Total Price</th>

<th>Total Downloads</th>

</tr>

</tfoot>

</table>

</div>

</div>

</Modal.Body>

<Modal.Footer>

<div className="row"></div>

</Modal.Footer>

</Modal>

</div>

);

}