import { Component, Injector, NgZone, OnInit } from '@angular/core';

import { NgForm } from '@angular/forms';

import { ActivatedRoute, Router } from '@angular/router';

//import \* as Razorpay from 'razorpay';

import { MovieTicket } from '../movieticket.model';

import { OrderDetails } from '../order\_detail.model';

import { TicketService } from '../ticket.service';

declare var Razorpay:any;

@Component({

selector: 'app-buy-ticket',

templateUrl: './buy-ticket.component.html',

styleUrls: ['./buy-ticket.component.css']

})

export class BuyTicketComponent implements OnInit {

ticketDetails: any| MovieTicket[] = [];

orderDetails:OrderDetails = {

fullName: '',

fullAddress: '',

contactNumber: '',

alternateContactNumber: '',

transactionId: '',

orderTicketQuantity: [] //quantity is stored in this array

}

isSingleTicketCheckout: string = '';

constructor(

private activatedRoute:ActivatedRoute,

private ticketService:TicketService,

private router:Router,

private injector:Injector

) { }

ngOnInit(): void {

this.ticketDetails = this.activatedRoute.snapshot.data['ticketDetails'];

this.isSingleTicketCheckout = this.activatedRoute.snapshot.paramMap.get("isSingleTicketCheckout");

this.ticketDetails.forEach(

x =>this.orderDetails.orderTicketQuantity.push(

{ticketId : x.ticketId, quantity: 1}//quantity default value given 1

)

);

console.log(this.ticketDetails);

console.log(this.orderDetails);

}

public placeOrder(orderForm: NgForm){

this.ticketService.placeOrder(this.orderDetails, this.isSingleTicketCheckout).subscribe(

(resp) =>{

console.log(resp);

orderForm.reset();

const ngZone = this.injector.get(NgZone);

ngZone.run(

() => {

this.router.navigate(["/orderConfirm"]);

},

() => {

}

);

},

(err) => {

console.log(err);

} );

}

getQuantityForTicket(ticketId){

const filteredTicket = this.orderDetails.orderTicketQuantity.filter(

(ticketQuantity) => ticketQuantity.ticketId === ticketId //filter return us the array

);

return filteredTicket[0].quantity;

}

getCalculatedTotal(ticketId,discountedPrice){

const filteredProduct = this.orderDetails.orderTicketQuantity.filter(

(ticketQuantity) => ticketQuantity.ticketId === ticketId //filter return us the array

);

return filteredProduct[0].quantity\*discountedPrice;

}

onQuantityChange(q,ticketId){

this.orderDetails.orderTicketQuantity.filter(

(orderTicket) => orderTicket.ticketId === ticketId

)[0].quantity = q;

}

getCalculatedGrandTotal(){

let grandTotal = 0;

this.orderDetails.orderTicketQuantity.forEach(

(ticketQuantity) => {

const price = this.ticketDetails.filter(ticket => ticket.ticketId === ticketQuantity.ticketId)[0].discountedPrice;

grandTotal = grandTotal + price\*ticketQuantity.quantity;

}

);

return grandTotal;

}

createTransactionAndPlaceOrder(orderForm:NgForm){

let amount = this.getCalculatedGrandTotal();

this.ticketService.createTransaction(amount).subscribe(

(response) => {

console.log(response);

this.openTransactionModel(response, orderForm);

},

(error) => {

console.log(error);

}

);

}

openTransactionModel(response:any,orderForm:NgForm){

var options = {

order\_id : response.orderId,

key : response.key,

amount : response.amount,

currency : response.currency,

name: 'MyMoviePlan\_PaymentGateway',

description : 'payment of bookng movie tickets',

image: 'https://cdn.pixabay.com/photo/2023/03/17/20/42/camera-7859402\_640.jpg',

//action to be performed once transaction is made is stored in handler

handler: (response: any) => {

if(response != null && response.razorpay\_payment\_id != null){

this.processResponse(response, orderForm);

}else{

alert("Payment failed...");

}

},

prefill : {

name: 'MyMoviePlan',

email: 'admin123@mymovieplan.com',

contact: '9090909090'

},

notes: {

address: 'online ticket booking',

},

theme: {

color: '#55C2C3'

}

};

var razorpayObject = new Razorpay(options);

return razorpayObject.open();

}

processResponse(resp: any,orderForm:NgForm){

this.orderDetails.transactionId = resp.razorpay\_payment\_id;

this.placeOrder(orderForm);

}

}

//BUyProductServiceResolver will return a list of product

/\*

ticketDetails is an object that we will bind with ui

in backend all detals full name addres and all have to be send to backend

all thees details are present in ticketDetails array

ticketDetails will be added to orderDetails as coded in init(

rigt now quantity is hardcoded as 1 later it can be changed in the array orderTicketQuantity

) \*/