<div class="container mt-5">

    <div class="row">

        <div class="col-md-6 container border pb-3 bg-light">

            <h2 class="text-center mb-3">Select Seats</h2>

            <div class="row text-center">

                <div class="col">

                    <a (click)="onClick(0)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(0) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[0].isAvailable ? emptySeat : filledSeat"></i></a>

                    <a (click)="onClick(1)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(1) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[1].isAvailable ? emptySeat : filledSeat"></i></a>

                    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

                    <a (click)="onClick(2)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(2) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[2].isAvailable ? emptySeat : filledSeat"></i></a>

                    <a (click)="onClick(3)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(3) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[3].isAvailable ? emptySeat : filledSeat"></i></a>

                </div>

            </div>

            <div class="row text-center">

                <div class="col">

                    <a (click)="onClick(4)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(4) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[4].isAvailable ? emptySeat : filledSeat"></i></a>

                    <a (click)="onClick(5)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(5) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[5].isAvailable ? emptySeat : filledSeat"></i></a>

                    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

                    <a (click)="onClick(6)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(6) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[6].isAvailable ? emptySeat : filledSeat"></i></a>

                    <a (click)="onClick(7)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(7) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[7].isAvailable ? emptySeat : filledSeat"></i></a>

                </div>

            </div>

            <div class="row text-center">

                <div class="col">

                    <a (click)="onClick(8)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(8) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[8].isAvailable ? emptySeat : filledSeat"></i></a>

                    <a (click)="onClick(9)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(9) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[9].isAvailable ? emptySeat : filledSeat"></i></a>

                    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

                    <a (click)="onClick(10)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(10) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[10].isAvailable ? emptySeat : filledSeat"></i></a>

                    <a (click)="onClick(11)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(11) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[11].isAvailable ? emptySeat : filledSeat"></i></a>

                </div>

            </div>

            <div class="row text-center">

                <div class="col">

                    <a (click)="onClick(12)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(12) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[12].isAvailable ? emptySeat : filledSeat"></i></a>

                    <a (click)="onClick(13)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(13) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[13].isAvailable ? emptySeat : filledSeat"></i></a>

                    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

                    <a (click)="onClick(14)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(14) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[14].isAvailable ? emptySeat : filledSeat"></i></a>

                    <a (click)="onClick(15)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(15) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[15].isAvailable ? emptySeat : filledSeat"></i></a>

                </div>

            </div>

            <div class="row text-center">

                <div class="col">

                    <a (click)="onClick(16)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(16) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[16].isAvailable ? emptySeat : filledSeat"></i></a>

                    <a (click)="onClick(17)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(17) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[17].isAvailable ? emptySeat : filledSeat"></i></a>

                    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

                    <a (click)="onClick(18)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(18) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[18].isAvailable ? emptySeat : filledSeat"></i></a>

                    <a (click)="onClick(19)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(19) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[19].isAvailable ? emptySeat : filledSeat"></i></a>

                </div>

            </div>

            <div class="row text-center">

                <div class="col">

                    <a (click)="onClick(20)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(20) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[20].isAvailable ? emptySeat : filledSeat"></i></a>

                    <a (click)="onClick(21)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(21) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[21].isAvailable ? emptySeat : filledSeat"></i></a>

                    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

                    <a (click)="onClick(22)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(22) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[22].isAvailable ? emptySeat : filledSeat"></i></a>

                    <a (click)="onClick(23)" class="mx-3"

                        [ngClass]="notAvailableSeats.has(23) ? 'text-danger' : 'text-primary'"><i

                            [ngClass]="seatSelect[23].isAvailable ? emptySeat : filledSeat"></i></a>

                </div>

            </div>

        </div>

        <div class="col-md-6">

            <div class="text-center">

                <h2>Selected Seats</h2>

                <div \*ngFor="let s of selectedSeats">

                    <h3>{{s+1}}</h3>

                </div>

            </div>

        </div>

    </div>

    <div \*ngIf="!booking.noOfPassengers" class="mt-3 text-center btn-lg">

        <button (click)="updateBusSeats()" class="btn btn-primary">Proceed</button>

    </div>

    <div \*ngIf="booking.noOfPassengers" class="mt-3 text-center btn-lg">

        <button (click)="updateBusSeats()" class="btn btn-primary" [disabled]="count!=0">Proceed</button>

    </div>

</div>

TS:

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

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

import { BusBookingDetailsComponent } from '../bus-booking-details/bus-booking-details.component';

import { Booking } from '../shared/models/booking';

import { BusSeats } from '../shared/models/bus-seats';

import { SeatSelect } from '../shared/models/seat-select';

import { BookingService } from '../shared/services/booking.service';

@Component({

  selector: 'app-seat-select',

  templateUrl: './seat-select.component.html',

  styleUrls: ['./seat-select.component.css']

})

export class SeatSelectComponent implements OnInit {

  seatSelect:BusSeats[]=[];

  custSession:string;

  emptySeat:string;

  filledSeat:string;

  selectedSeats:number[]=[];

  notAvailableSeats:Set<number> = new Set<number>();

  count:number;

  booking:Booking;

  constructor(private bookingService:BookingService, private router:Router) {

    // this.seatSelect = [

    //   {seatNo:1,isAvailable:true},{seatNo:2,isAvailable:true},{seatNo:3,isAvailable:true},{seatNo:4,isAvailable:false},

    //   {seatNo:5,isAvailable:true},{seatNo:6,isAvailable:true},{seatNo:7,isAvailable:true},{seatNo:8,isAvailable:true},

    //   {seatNo:9,isAvailable:true},{seatNo:10,isAvailable:false},{seatNo:11,isAvailable:true},{seatNo:12,isAvailable:true},

    //   {seatNo:13,isAvailable:false},{seatNo:14,isAvailable:true},{seatNo:15,isAvailable:true},{seatNo:16,isAvailable:false},

    //   {seatNo:17,isAvailable:true},{seatNo:18,isAvailable:true},{seatNo:19,isAvailable:true},{seatNo:20,isAvailable:true},

    //   {seatNo:21,isAvailable:true},{seatNo:22,isAvailable:true},{seatNo:23,isAvailable:true},{seatNo:24,isAvailable:true},

    // ];

    this.emptySeat = 'far fa-bookmark fa-5x'

    this.filledSeat = 'fas fa-bookmark fa-5x'

    // console.log(this.seatSelect[0].isAvailable)

  }

  ngOnInit(): void {

    this.fillSeats();

  }

  fillSeats(){

    // console.log(this.bookingService.busDetails.busScId)

    // console.log(this.bookingService.returnBusDetails.busScId)

    if(this.bookingService.returnBusDetails.busScId){

      this.bookingService.getBusSeatNos(this.bookingService.returnBusDetails.busScId).subscribe(

        data => {

          this.seatSelect = data as BusSeats[];

          console.log(data);

          this.starterPack();

        },

        err => {

          console.log(err)

        }

      )

    }

    else{

      this.bookingService.getBusSeatNos(this.bookingService.busDetails.busScId).subscribe(

        data => {

          this.seatSelect = data as BusSeats[];

          console.log(data);

          this.starterPack();

        },

        err => {

          console.log(err)

        }

      )

    }

    // console.log(this.seatSelect);

  }

  starterPack(){

    this.count = this.bookingService.booking.noOfPassengers ? this.bookingService.booking.noOfPassengers : 0;

    this.booking = this.bookingService.booking;

    this.seatSelect.forEach(element => {

      if(!element.isAvailable){

        this.notAvailableSeats.add(element.seatNo-1);

      }

    });

    // console.log(this.notAvailableSeats)

  }

  onClick(i:number){

    // console.log(this.notAvailableSeats.has(i),this.notAvailableSeats);

    if(this.booking.noOfPassengers){

      if(this.count>0 || this.selectedSeats.includes(i)){

        if(!this.notAvailableSeats.has(i)){

          if(this.selectedSeats.includes(i)){

            this.count += 1;

            console.log("inside \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

          }

          else if(this.count>0){

            this.count -= 1;

          }

          this.seatSelect[i].isAvailable = !this.seatSelect[i].isAvailable;

          if(!this.seatSelect[i].isAvailable){

            this.selectedSeats.push(i);

          }

          else{

            this.selectedSeats = this.selectedSeats.filter(item => item !== i)

            // this.count += 1;

          }

        }

      }

    }

    else{

      if(!this.notAvailableSeats.has(i)){

        this.seatSelect[i].isAvailable = !this.seatSelect[i].isAvailable;

        if(!this.seatSelect[i].isAvailable){

          this.selectedSeats.push(i);

        }

        else{

          this.selectedSeats = this.selectedSeats.filter(item => item !== i)

        }

      }

    }

  }

  updateBusSeats(){

    this.custSession= localStorage.getItem('cust')!;

    if(this.bookingService.returnBusDetails.busScId){

      localStorage.setItem("returnSeatSelect",JSON.stringify(this.seatSelect));

    }

    else{

      localStorage.setItem("seatSelect",JSON.stringify(this.seatSelect));

    }

    // this.seatSelect.forEach(element => {

    //   this.bookingService.updateBusSeats(element).subscribe(

    //     data=>{

    //       console.log(data);

    //     },

    //     err => {

    //       console.log(err);

    //     }

    //   )

    // });

    if(this.bookingService.returnBusDetails.busScId){

      this.navigateToBookingConfirmation();

    }

    else{

      this.navigateToAddPassengers();

    }

  }

  navigateToBookingConfirmation(){

    let returnFare:number;

    this.bookingService.getReturnFare(this.bookingService.returnBusDetails.busScId).subscribe(

      data=>{

        returnFare = data as number;

        console.log(returnFare);

        this.bookingService.booking.totalFare = this.bookingService.booking.totalFare + (returnFare \* this.selectedSeats.length);

      },

      err=>{

        console.log(err);

      }

    );

    this.selectedSeats.forEach((element,index) => {

      this.bookingService.passengerlist[index].returnSeatNo = element+1;

    });

    if(this.custSession=='zxc'){

      this.router.navigate(["cust-dashboard/booking-confirmation"]);

    }

    else{

      this.router.navigate(["cust-dashboard/unauth-booking-confirmation"])

    }

  }

  navigateToAddPassengers(){

    this.bookingService.booking.totalFare = this.bookingService.busDetails.fare \* this.selectedSeats.length;

    this.bookingService.bookedseats = this.selectedSeats;

    this.bookingService.booking.noOfPassengers = this.selectedSeats.length;

    this.router.navigate(["cust-dashboard/add-passenger-details"]);

  }

}

Service

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http'

import { BusSeats } from '../models/bus-seats';

import { Passenger } from '../models/Passenger';

import { Booking } from '../models/booking';

import { BusDetails } from '../models/busDetails';

import { ReturnBooking } from '../models/return-booking';

import { Customer } from '../models/customer';

@Injectable({

  providedIn: 'root'

})

export class BookingService {

  readonly UrlBusSeat = "http://localhost:43836/api/BusSeatNoes"

  readonly UrlBus = "http://localhost:43836/api/buses"

  readonly UrlBusSearch = "http://localhost:43836/api/BusSchedules"

  readonly UrlBooking = "http://localhost:43836/api/Bookings"

  readonly UrlPassenger = "http://localhost:43836/api/PassengerDetails"

  readonly UrlReturn = "http://localhost:43836/api/ReturnBookings"

  readonly UrlReturnBookingId = "http://localhost:43836/api/ReturnBookings/byBookingId?bookingId="

  readonly UrlCustomerDeduct = "http://localhost:43836/api/Customers/deductFare"

  readonly UrlCustomer = "http://localhost:43836/api/Customers"

  readonly UrlReduceAvailableSeats = "http://localhost:43836/api/BusSchedules/reduceSeats"

  readonly UrlAddAvailableSeats = "http://localhost:43836/api/BusSchedules/addSeats"

  readonly UrlReturnBusFare = "http://localhost:43836/api/BusSchedules/getFareOfReturnBus?Id="

  constructor(private http: HttpClient) {

    this.busDetails = new BusDetails();

    this.booking = new Booking();

    this.returnBusDetails = new BusDetails();

  }

  booking: Booking;

  returnBusDetails: BusDetails;

  busDetails: BusDetails;

  bookedseats: number[] = [];

  passengerlist: Passenger[];

  unAuthCust: Customer;

  // busList: BusDetails[] = [{

  //   busScId:6,departureDate:"2021-08-12",

  //   busNo: 101, busName: "Velocity", source: "Mumbai",

  //   destination: "Delhi",

  //   departureTime: "22:10",

  //   arrivalTime: "12:20",

  //   noOfSeats: 23,

  //   via: "Bangalore",

  //   fare: 2500, driverName: "John",

  //   driverAge: 30,

  //   driverExperience: 5

  // },

  // {

  //   busScId:6,departureDate:"2021-08-12",

  //   busNo: 101, busName: "PanExpress", source: "Delhi",

  //   destination: "Mumbai",

  //   departureTime: "20:00",

  //   arrivalTime: "15:30",

  //   noOfSeats: 23,

  //   via: "Pune",

  //   fare: 2500, driverName: "Jack",

  //   driverAge: 35,

  //   driverExperience: 4

  // }

  // ];

  getBusSeatNos(busId?: number) {

    return this.http.get(this.UrlBusSeat + '/' + busId);

  }

  updateBusSeats(seats: BusSeats) {

    return this.http.put(this.UrlBusSeat + '/' + seats.seatId, seats);

  }

  getBusbyid(id?: number) {

    return this.http.get(this.UrlBus + "/" + id);

  }

  getReturnFare(id?: number) {

    return this.http.get(this.UrlReturnBusFare + id);

  }

  getBusSchedulebyid(id?: number) {

    return this.http.get(this.UrlBusSearch + "/" + id);

  }

  searchBuses(source:string, destination:string, date:string){

    return this.http.get(`${this.UrlBusSearch}/searchBuses?source=${source}&destination=${destination}&date=${date}`)

  }

  getSources(){

    return this.http.get(`${this.UrlBus}/getSources`);

  }

  getDestination(){

    return this.http.get(`${this.UrlBus}/getDestination`);

  }

  addBooking(booking:Booking){

    return this.http.post(this.UrlBooking,booking);

  }

  addPassengers(passengersList:Passenger[]){

    return this.http.post(this.UrlPassenger,passengersList);

  }

  addReturnBooking(returnBookingDetails:ReturnBooking){

    return this.http.post(this.UrlReturn,returnBookingDetails);

  }

  deductWalletAmount(cid:number,totalFare:number){

    return this.http.put(`${this.UrlCustomerDeduct}?cid=${cid}&fare=${totalFare}`,null,{responseType:'text'});

  }

  reduceAvailableSeats(busScId:number,num:number){

    return this.http.put(`${this.UrlReduceAvailableSeats}?busScId=${busScId}&num=${num}`,null,{responseType:'text'});

  }

  addAvailableSeats(busScId:number,num:number){

    return this.http.put(`${this.UrlAddAvailableSeats}?busScId=${busScId}&num=${num}`,null,{responseType:'text'});

  }

  addCustomer(cust:Customer){

    return this.http.post(this.UrlCustomer,cust);

  }

  sendTicket(cid:number,bookingId:number,busNo:number,source:string,destination:string,date:string){

    return this.http.get(`${this.UrlBooking}/sendTicket?cid=${cid}&bookingId=${bookingId}&busNo=${busNo}&source=${source}&dest=${destination}&date=${date}`,{responseType:'text'});

  }

  //get passenger

  getPassenger(bookingid:number)

  {

    return this.http.get(`${this.UrlPassenger}/getPassenger?bookingid=${bookingid}`);

  }

  //get passengerseatno for a booking id

  getPassengerSeatNo(bookingid:number)

  {

    return this.http.get(`${this.UrlPassenger}/getPassengerSeatNo?bookingid=${bookingid}`);

  }

  //get passenger return seatno for a booking id //check to change the route

  getPassengerReturnSeatNo(bookingid:number)

  {

    return this.http.get(`${this.UrlPassenger}/getPassengerReturnSeatNo?bookingid=${bookingid}`);

  }

  //get BusSeatNo table by busscid

  getBusSeatNoTable(busscid:number)

  {

    return this.http.get(this.UrlBusSeat + "/" + busscid);

  }

  //put BusSeatNo table

  putBusSeatNoTable(busSeatNoTable:BusSeats)

  {

    return this.http.put(this.UrlBusSeat + "/" + busSeatNoTable.seatId,busSeatNoTable);

  }

//refund wallet amount //check route for this

  refundWalletAmount(cid:number,totalFare:number)

  {

    return this.http.put(`${this.UrlCustomer}/refundFare?cid=${cid}&fare=${totalFare}`,null,{responseType:'text'});

  }

//put booking for updating statud

  putBooking(booking:Booking)

  {

    return this.http.put(this.UrlBooking + "/" + booking.bookingId,booking);

  }

  //get booking

  getBooking(bookingid:number)

  {

    return this.http.get(this.UrlBooking + "/" + bookingid);

  }

  getReturnBusScId(bookingid:number)

  {

    return this.http.get(this.UrlReturnBookingId + bookingid);

  }

}