**Source code**

1. **RemotehiringApplication**

package com.remotehiring;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class RemotehiringApplication {

public static void main(String[] args) {

SpringApplication.run(RemotehiringApplication.class, args);

}

}

1. **Index.html**

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>MyMovies</title>

<base href="/">

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

<link rel="icon" type="image/x-icon" href="favicon.ico">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700|Roboto+Slab:400,700|Material+Icons" />

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/latest/css/font-awesome.min.css">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css">

<link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500&display=swap" rel="stylesheet">

<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">

<script src='https://kit.fontawesome.com/a076d05399.js'></script>

<script src="script.js" type="text/javascript"></script>

<script async>(function(w, d) { w.CollectId = "5e2445bfc07d8746c2a25671"; var h = d.head || d.getElementsByTagName("head")[0]; var s = d.createElement("script"); s.setAttribute("type", "text/javascript"); s.setAttribute("src", "https://collectcdn.com/launcher.js"); h.appendChild(s); })(window, document);</script>

<!--<style>

body{

font-family: "cooper";

color: yellowgreen;

animation: change 10s infinite;

}

@keyframes change{

0%{background-color:#F79F1F}

25%{background-color:#EE5A24}

50%{background-color:#bd13f1}

100%{background-color:#EE5A24}

25%{background-color:#EE5A24}

10%{background-color:#F79F1F}

}

</style>-->

</head>

<body class="mat-typography">

<app-root></app-root>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI" crossorigin="anonymous"></script>

</body>

</html>

1. **Admin-login.component.html**

<div class="row">

<div class="col-md-12 wrapper d-flex justify-content-center align-items-center" style="height:95vh">

<div class="row d-flex justify-content-center align-items-center formwrapper card">

<form name="form" [formGroup] = "form" (ngSubmit) = "form.valid && onSubmit()">

<div class="col-md-12 mt-4 d-flex flex-row justify-content-center">

<div class="m-1">

<!-- <button class="button" routerLink="/signup">Sign Up </button>

<button class="button" routerLink="/login">Login</button> -->

<h2>Admin Login</h2>

</div>

</div>

<div class="col-md-12 d-flex flex-row justify-content-center">

<div class="m-1">

<input class="input form-control" formControlName="adminid" [ngClass]="{ 'is-invalid':form.get('adminid').touched && form.get('adminid').invalid}" name="adminid" type="email" placeholder="Admin ID" autocomplete="off" required>

<div class="invalid-feedback">

<div \*ngIf="form.get('adminid').errors?.required">

This Field is required!!

</div>

</div>

</div>

</div>

<div class="col-md-12 d-flex flex-row justify-content-center">

<div class="m-1">

<input class="input form-control" formControlName="password" [ngClass]="{ 'is-invalid':form.get('password').touched && form.get('password').invalid }" type="password" placeholder="Password" required>

<div class="invalid-feedback">

<div>

This Field is required!!

</div>

</div>

</div>

</div>

<div class="col-md-12 d-flex flex-row justify-content-center">

<div class="mt-4">

<p class="danger"><button style="border:none;background-color: transparent;color: #ff4f5a;">Forgot Password ?</button></p>

</div>

</div>

<div class="col-md-12 mt-2 mb-5 d-flex flex-row justify-content-center">

<div class="m-1 mt-4">

<button class="button" type="submit" [disabled]="!form.valid" >Login</button>

</div>

</div>

</form>

</div>

</div>

</div>

1. **Booking.component.html**

<div class="row mt-3 justify-content-center">

<div class="col-md-11 card " >

<div class="row " style="background: #cc3d00 radial-gradient(circle at 60% 100%,#ffa436 5%,#cc3d00 94%);;color: black;margin: 10px;padding: 10px;">

<div class="col-md-12 d-flex justify-content-center" >

<h2>My Bookings</h2>

</div>

</div>

<hr>

<div class="row mt-2">

<div class="col-md-12">

<div class="row mb-3">

<div class="col-md-12 overflow-auto" style="height: 35rem;">

<div class="accordion" id="accordionExample">

<div \*ngIf='!show'>

<div \*ngFor='let booking of bookings$' class="card">

<div class="card-header" id="headingTwo">

<div class="row">

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

<h3><b>Movie Id :</b> {{booking.locationid}}</h3>

</div>

<div class="col-md-7">

<h3><b>Date :</b> {{booking.time}}</h3>

</div>

</div>

</div>

<div>

<div class="card-body">

<div class="row">

<div class="col-auto mr-auto">

<b>Show Time : </b> {{booking.slotid}} <br>

<b>Seating Type :</b> {{booking.vehicle\_type}} <br>

<b>Count :</b> {{booking.duration}}<br>

<div \*ngIf="booking.paid != 0 || booking.paid == 0">

<b>Cost :</b> {{booking.cost}} Rs

</div>

</div>

<div class="col-auto" \*ngIf='booking.paid == 0'>

<button class="button" (click) = 'endBooking(booking.bookingid)'>Make Payment</button>

</div>

<div class="col-auto" \*ngIf='booking.paid != 0'>

<p class="text-success font-weight-bold">Booking Completed</p>

</div>

</div>

</div>

</div>

</div>

</div>

<div \*ngIf='show'>

<h2>

No Bookings Found...

</h2>

<button class="button" routerLink='/dashboard' > Book Now</button>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

1. **Header.component.html**

<nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #6b6b69 !important;box-shadow: 0px 0px 8px rgba(128, 128, 128, 0.438);">

<a class="navbar-brand" href="#" style="color:white;"><i class="fas fa-film" style="color: white;font-size: 30px;"></i><b>MyMoviePlan</b></a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto">

</ul>

<div class="form-inline my-2 my-lg-0">

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit" background-color="#c70a0a">Search</button>

</form>

<button class="button" routerLink="/">Home</button>

<button class="button" routerLink="/signup">Sign up</button>

<button class="button" routerLink="/login">Login</button>

<button class="button" routerLink="/admin">Admin</button>

</div>

</div>

</nav>

1. **Header-login.component.html**

<nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #2089E4 !important;box-shadow: 0px 0px 8px rgba(128, 128, 128, 0.438);">

<a class="navbar-brand" href="#" style="color:white;"><i class="fas fa-film" style="color: white;font-size: 30px;"></i><b>My-Movies</b></a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto">

</ul>

<div class="form-inline my-2 my-lg-0">

<button class="button" routerLink="/">Home</button>

<button class="button" (click)='logout()'>Logout</button>

</div>

</div>

</nav>

1. **Home-main-section.component.html**

<div id="carouselExampleControls" class="carousel slide" data-ride="carousel">

<div class="carousel-inner">

<div class="carousel-item active">

<img class="d-block w-100" src="/assets/images/LU cinema.png " alt="LU cinema.png">

</div>

<div class="carousel-item">

<img class="d-block w-100" src="/assets/images/kids.jpg" alt="kids.jpg">

</div>

<div class="carousel-item">

<img class="d-block w-100" src="/assets/images/cinema.jpg" alt="cinema.jpg">

</div>

</div>

<a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-slide="prev">

<span class="carousel-control-prev-icon" aria-hidden="true"></span>

<span class="sr-only">Previous</span>

</a>

<a class="carousel-control-next" href="#carouselExampleControls" role="button" data-slide="next">

<span class="carousel-control-next-icon" aria-hidden="true"></span>

<span class="sr-only">Next</span>

</a>

</div>

<br><br><br>

<div class="row">

<div class="col-md-3 options-primary p-0 d-flex justify-content-center align-items-center flex-column">

<i class="far fa-money-bill-alt" style="font-size: 60px;margin-bottom: 10px;"></i>

<p>Economical</p>

</div>

<div class="col-md-3 options-secondary p-0 d-flex justify-content-center align-items-center flex-column">

<i class="fas fa-history" style="font-size: 60px;margin-bottom: 10px;"></i>

<p>Fast</p>

</div>

<div class="col-md-3 options-primary p-0 d-flex justify-content-center align-items-center flex-column">

<i class="fas fa-user-shield" style="font-size: 60px;margin-bottom: 10px;"></i>

<p>Easy Booking</p>

</div>

<div class="col-md-3 options-secondary p-0 d-flex justify-content-center align-items-center flex-column">

<i class="fas fa-calendar-alt" style="font-size: 60px;margin-bottom: 10px;"></i>

<p>24/7</p>

</div>

</div>

<div class="row d-flex justify-content-center" style="height: 70vh;">

<div class="col-md-6 d-flex justify-content-center align-items-center">

<div>

<img src="/assets/images/nms.jpg" alt="">

</div>

</div>

<div class="col-md-6 d-flex justify-content-center flex-column">

<h1><b>How It Works?</b></h1>

<h4>You can easily choose from any tickets you like without going to the actual theatre for cinema ticket booking. You can sit back at home and book in advance whenever the movie tickets booking starts according to your own convenience.<br>

</h4>

<br>

<div>

<button class="getquotebtn" (click)='checkSession()'>Book Now</button>

</div>

</div>

</div>

1. **Bookings.model.ts**

export class Bookings{

bookingid: number;

email: string;

vehicle\_type: string;

vehicle\_no: string;

date: string;

time: string;

duration: number;

cost: string;

locationid: number;

slotid: string;

paid: number;}

1. **Users.model.ts**

export class Users{

email : string;

fullname: string;

mobno: string;

password: string;

}

1. Admin.service.ts

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

import { HttpClient, HttpHeaders } from '@angular/common/http';

@Injectable({

providedIn: 'root'

})

export class AdminService {

apiUrl = 'http://localhost:8080/admin';

httpOptions = {

headers :new HttpHeaders({

'Content-Type':'application/json'

})

}

constructor(private \_http: HttpClient) { }

loginCheck(loginDetails){

return this.\_http.post<Boolean>(this.apiUrl,loginDetails, this.httpOptions);

}

}

1. **Booking.service.ts**

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

import { HttpClient, HttpHeaders } from '@angular/common/http';

import { Bookings } from '../models/bookings.model';

import { Observable } from 'rxjs';

@Injectable({

providedIn: 'root'

})

export class BookingsService {

apiUrl = 'http://localhost:8080'

httpOptions = {

headers :new HttpHeaders({

'Content-Type':'application/json'

})

}

constructor(private \_http: HttpClient) { }

getAllBookings(){

return this.\_http.get<Bookings[]>(this.apiUrl+'/bookings/allBookings')

}

getBookings(email){

return this.\_http.get<Bookings[]>(this.apiUrl+'/bookings/getByUser/'+email);

}

addBooking(id,bookings):Observable<Bookings>{

bookings.locationid = id;

bookings.email = sessionStorage.getItem('email');

console.log(bookings);

return this.\_http.post<Bookings>(this.apiUrl+'/bookings/add', JSON.stringify(bookings), this.httpOptions );

}

endBooking(bookingid){

return this.\_http.get<Boolean>(this.apiUrl+'/bookings/endBooking/'+bookingid);

}

}

1. **Users.service.ts**

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

import { HttpClient, HttpHeaders } from '@angular/common/http';

import { Users } from 'src/app/models/users.model';

@Injectable({

providedIn: 'root'

})

export class UsersService {

apiUrl = 'http://localhost:8080/users';

httpOptions = {

headers :new HttpHeaders({

'Content-Type':'application/json'

})

}

constructor(private \_http: HttpClient) { }

loginCheck(loginDetails){

return this.\_http.post<Boolean>(this.apiUrl+'/login',loginDetails, this.httpOptions);

}

signup(signupDetails){

return this.\_http.post<Boolean>(this.apiUrl+'/signup',signupDetails, this.httpOptions);

}

}