**app-routing.module.ts**

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

import { RouterModule, Routes } from '@angular/router';

import { DashboardComponent } from './dashboard/dashboard.component';

import { AddClientComponent } from './dashboard/add-client/add-client.component';

import { SchedulesComponent } from './dashboard/schedules/schedules.component';

import { AddMeetingComponent } from './dashboard/add-meeting/add-meeting.component';

import { LoginComponent } from './login/login.component';

import { RegisterComponent } from './register/register.component';

import { ViewClientsComponent } from './dashboard/view-clients/view-clients.component';

const routes: Routes = [

  {path:"home",component:DashboardComponent},

  {path:"add\_client",component:AddClientComponent},

  {path:"schedules",component:SchedulesComponent},

  {path:"add\_meeting",component:AddMeetingComponent},

  {path:"",redirectTo:"login",pathMatch:"full"},

  {path:"login",component:LoginComponent},

  {path:"register",component:RegisterComponent},

  {path:"view\_clients",component:ViewClientsComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

**app.module.ts**

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

import { BrowserModule } from '@angular/platform-browser';

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

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { DashboardComponent } from './dashboard/dashboard.component';

import { AddClientComponent } from './dashboard/add-client/add-client.component';

import { AddMeetingComponent } from './dashboard/add-meeting/add-meeting.component';

import { SchedulesComponent } from './dashboard/schedules/schedules.component';

import { LoginComponent } from './login/login.component';

import { RegisterComponent } from './register/register.component';

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

import { ViewClientsComponent } from './dashboard/view-clients/view-clients.component';

@NgModule({

  declarations: [

    AppComponent,

    DashboardComponent,

    AddClientComponent,

    AddMeetingComponent,

    SchedulesComponent,

    LoginComponent,

    RegisterComponent,

    ViewClientsComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,FormsModule,ReactiveFormsModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

**login.service.ts**

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

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

@Injectable({

  providedIn: 'root'

})

export class LoginService {

  constructor(public router:Router) { }

  checkLoginDetails(login:any):string {

    const registeredEmail = sessionStorage.getItem('registeredEmail');

    const registeredPassword = sessionStorage.getItem('registeredPassword');

    if(login.emailid ===registeredEmail && login.password === registeredPassword) {

      alert ("Successfully Logged In");

      sessionStorage.setItem('emailid',login.emailid);

      this.router.navigate(["/home"],{skipLocationChange:true})

      return "Success"

    } else {

      alert("Login Failed. Please Try Again")

      return "Failure"

    }

  }

}

**Client-management.service.ts**

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

import { Client } from '../models/client.model';

import { Meeting } from '../models/meeting.model';

@Injectable({

  providedIn: 'root'

})

export class ClientManagementService {

  private clients: Client[] = [];

  private meetings: Meeting[] = [];

  constructor() { }

  addClient(client:Client) {

    this.clients.push(client);

    console.log('Client added', client);

  }

  getClients() {

    return this.clients;

  }

  addMeeting(meeting:Meeting) {

    this.meetings.push(meeting);

    console.log('Meeting added', meeting);

  }

  getMeetings() {

    return this.meetings;

  }

}

**Register.component.ts**

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

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

@Component({

  selector: 'app-register',

  templateUrl: './register.component.html',

  styleUrls: ['./register.component.css']

})

export class RegisterComponent {

  name: string='';

  email: string = '';

  password: string='';

  constructor(private router: Router){}

  onRegister() {

    if(this.name && this.email && this.password) {

    sessionStorage.setItem('registeredEmail',this.email);

    sessionStorage.setItem('registeredPassword',this.password);

    alert("Registration Successful!");

    this.router.navigate(["/login"])

    } else {

      alert("Error. Please try again")

    }

  }

}

**Register.component.html**

<div>

<section class="register">

    <h2>Don't have an account? Register here!</h2>

    <form (ngSubmit)="onRegister()" >

      <label for="register-name">Name:</label>

      <input type="text" name="register-name" id="register-name" [(ngModel)]="name" placeholder="John Doe" required/>

      <label for="register-email">Email:</label>

      <input type="email" name="register-email" id="register-email" [(ngModel)]="email" placeholder="123@gmail.com" required/>

      <label for="register-pw">Password</label>

      <input type="password" id="register-pw" name="register-pw" [(ngModel)]="password" required/>

      <button type="submit">Register Now</button>

    </form>

</section>

</div>

**Meeting.model.ts**

export interface Meeting {

    clientName: string;

    date: string;

    time:string;

    purpose:string;

}

**Client.model.ts**

export interface Client {

    name:string;

    email:string;

    type:string;

    memo:string;

}

**Login.component.ts**

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

import { LoginService } from '../services/login.service';

import { FormControl, FormGroup } from '@angular/forms';

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

@Component({

  selector: 'app-login',

  templateUrl: './login.component.html',

  styleUrls: ['./login.component.css']

})

export class LoginComponent {

  msg:string=''

  loginRef = new FormGroup ({

    emailid: new FormControl(),

    password: new FormControl()

  });

  constructor(public ls:LoginService, public router:Router) { }

checkLoginDetails(): void {

  let login = this.loginRef.value;

  console.log(login);

  this.msg=this.ls.checkLoginDetails(login);

  this.loginRef.reset();

}

}

**Login.component.html**

<header>Welcome to JB Management App!</header>

<div>

<section class="login">

    <h2>Login</h2>

    <form [formGroup]="loginRef" (ngSubmit)="checkLoginDetails()" id="login-form">

      <label for="login-email">Email:</label>

      <input type="email" formControlName="emailid" name="login-email" id="login-email" required/>

      <label for="login-pw">Password:</label>

      <input type="password" formControlName="password" id="login-pw" required/>

      <button type="submit">Login</button>

    </form>

    <a routerLink="/register">Sign Up Here</a>

</section>

</div>

**Dashboard.component.ts**

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

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

@Component({

  selector: 'app-dashboard',

  templateUrl: './dashboard.component.html',

  styleUrls: ['./dashboard.component.css']

})

export class DashboardComponent implements OnInit {

  user:string=""

  constructor(public router:Router){}

  ngOnInit(): void {

    let obj=sessionStorage.getItem("emailid");

    if (obj!=null) {

      this.user=obj;

    }

  }

  logout():void {

    sessionStorage.clear();

    this.router.navigate(["/login"])

  }

}

**Dashboard.component.html**

<header>

    <h1>Welcome User {{ user }}</h1>

</header>

<main>

    <section class="dashboard-options">

        <h2>What would you like to do?</h2>

        <button routerLink="/add\_client">Add Client</button>

        <button routerLink="/view\_clients">View Clients</button>

        <button routerLink="/add\_meeting">Add Meeting</button>

        <button routerLink="/schedules">View Schedules</button>

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

    </section>

</main>

**View-clients.component.ts**

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

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

import { Client } from 'src/app/models/client.model';

import { ClientManagementService } from 'src/app/services/client-management.service';

@Component({

  selector: 'app-view-clients',

  templateUrl: './view-clients.component.html',

  styleUrls: ['./view-clients.component.css']

})

export class ViewClientsComponent implements OnInit {

  clients:Client[] = [];

  constructor(private clientService: ClientManagementService, private router:Router) {}

  ngOnInit(): void {

    this.clients = this.clientService.getClients();

  }

  goToDashboard():void {

    this.router.navigate(["/home"])

  }

}

**View-clients.component.html**

<div>

    <h2>All Clients</h2>

    <table \*ngIf="clients.length > 0">

       <thead>

        <tr>

            <th>Client Name</th>

            <th>Client Email</th>

            <th>Client Type</th>

            <th>Memo</th>

        </tr>

       </thead>

       <tbody>

        <tr \*ngFor="let client of clients">

            <td>{{client.name}}</td>

            <td>{{client.email}}</td>

            <td>{{client.type}}</td>

            <td>{{client.memo}}</td>

        </tr>

       </tbody>

    </table>

    <button (click)="goToDashboard()">Go to Dashboard</button>

</div>

**Schedules.component.ts**

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

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

import { Meeting } from 'src/app/models/meeting.model';

import { ClientManagementService } from 'src/app/services/client-management.service';

@Component({

  selector: 'app-schedules',

  templateUrl: './schedules.component.html',

  styleUrls: ['./schedules.component.css']

})

export class SchedulesComponent implements OnInit {

meetings: Meeting[]= [];

constructor(private clientService: ClientManagementService,private router:Router) {}

ngOnInit(): void {

  this.meetings = this.clientService.getMeetings();

}

goToDashboard():void {

  this.router.navigate(["/home"])

}

}

**Schedules.component.html**

<div>

    <h2>Meeting Schedules</h2>

    <table \*ngIf="meetings.length > 0">

       <thead>

        <tr>

            <th>Client Name</th>

            <th>Date</th>

            <th>Time</th>

            <th>Topic</th>

        </tr>

       </thead>

       <tbody>

        <tr \*ngFor="let meeting of meetings">

            <td>{{meeting.clientName}}</td>

            <td>{{meeting.date}}</td>

            <td>{{meeting.time}}</td>

            <td>{{meeting.purpose}}</td>

        </tr>

       </tbody>

    </table>

    <button (click)="goToDashboard()">Go to Dashboard</button>

</div>

**Add-meeting.component.ts**

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

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

import { Meeting } from 'src/app/models/meeting.model';

import { ClientManagementService } from 'src/app/services/client-management.service';

@Component({

  selector: 'app-add-meeting',

  templateUrl: './add-meeting.component.html',

  styleUrls: ['./add-meeting.component.css']

})

export class AddMeetingComponent {

  meeting: Meeting = {

    clientName: '',

    date:'',

    time:'',

    purpose:'',

  }

  constructor(private meetingService: ClientManagementService,private router:Router) {}

  onSubmit() {

    if (this.meeting.clientName && this.meeting.date && this.meeting.time && this.meeting.purpose) {

      this.meetingService.addMeeting(this.meeting);

      console.log('Meeting added:',this.meeting)

      this.meeting = {

        clientName: '',

        date: '',

        time:'',

        purpose:'',

      }

      alert("Meeting Added")

    }

  }

  goToDashboard():void {

    this.router.navigate(["/home"])

  }

}

**Add-meeting.component.html**

<div>

    <h2>Add Meeting With Client</h2>

    <form (ngSubmit)="onSubmit()">

        <label for="client-name">Client Name:</label>

        <input type="text" name="client-name" id="client-name" [(ngModel)]="meeting.clientName" required/><br/>

        <label for="meeting-date">Meeting Date:</label>

        <input type="date" name="meeting-date" id="meeting-date" [(ngModel)]="meeting.date" required><br/>      <!--getting date format error in console-->

        <label for="meeting-time">Meeting Time:</label>

        <input type="time" name="meeting-time" id="meeting-time" [(ngModel)]="meeting.time" required><br/>

        <label for="meeting-purpose">Meeting Topic:</label>

        <textarea name="meeting-purpose" id="meeting-purpose" rows="3" cols="20" placeholder="Meeting topic is..." [(ngModel)]="meeting.purpose"></textarea><br/>

        <button type="submit">Add Meeting</button>

    </form>

    <button (click)="goToDashboard()">Go to Dashboard</button>

</div>

**Add-client.component.ts**

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

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

import { Client } from 'src/app/models/client.model';

import { ClientManagementService } from 'src/app/services/client-management.service';

@Component({

  selector: 'app-add-client',

  templateUrl: './add-client.component.html',

  styleUrls: ['./add-client.component.css']

})

export class AddClientComponent {

  client: Client = {name:'', email:'', type:'', memo:''};

  constructor(private clientAddService: ClientManagementService, private router: Router)  {}

  onSubmit() {

    if (this.client.name && this.client.email) {

      this.clientAddService.addClient(this.client);

      console.log('Client added:', this.client);

      sessionStorage.setItem('lastClientName',this.client.name);

      sessionStorage.setItem('lastClientEmail',this.client.email);

      sessionStorage.setItem('lastClientType',this.client.type);

      sessionStorage.setItem('lastClientMemo',this.client.memo);

      this.client = {name:'', email:'',type:'',memo:''};

      alert ("Client Added")

    }

  }

  goToDashboard():void {

    this.router.navigate(["/home"])

  }

  ngOnInit() {

  }

}

**Add-client.component.html**

<div>

    <h2>Add Client</h2>

    <form (ngSubmit)="onSubmit()" #clientForm="ngForm">

        <label for="client-name">Client Name</label>

        <input type="text" name="client-name" id="client-name" [(ngModel)]="client.name" required/><br/>

        <label for="client-email">Client Email</label>

        <input type="email" name="client-email" id="client-email" [(ngModel)]="client.email" required/><br/>

        <label for="client-type">Client Type</label>

        <select name="client-type" id="client-type" [(ngModel)]="client.type">

            <option value="">--Client Type--</option>

            <option value="individual">Individual</option>

            <option value="company">Company</option>

        </select><br/>

        <label for="client-memo">Client Memo</label>

        <textarea name="client-memo" id="client-memo" rows="3" cols="20" placeholder="Memo about client..." [(ngModel)]="client.memo"></textarea><br/>

        <button type="submit">Add Client</button>

    </form>

    <button (click)="goToDashboard()">Go to Dashboard</button>

</div>