

Event Management App

Priyanka M V

json-typings.d.ts

```
declare module "*.json" {  
  const value: any;  
  export default value;  
}
```

db.json

```
{  
  "employees": [  
    {  
      "id": 1,  
      "first_name": "Sebastian",  
      "last_name": "Eschweiler",  
      "email": "sebastian@codingthesmartway.com"  
    },  
    {  
      "id": 2,  
      "first_name": "Steve",  
      "last_name": "Palmer",  
      "email": "steve@codingthesmartway.com"  
    },  
    {  
      "id": 3,  
      "first_name": "Ann",  
      "last_name": "Smith",  
      "email": "ann@codingthesmartway.com"  
    }  
  ]  
}
```

app.module.ts

```
import { NgModule } from '@angular/core';  
import { BrowserModule, provideClientHydration } from '@angular/platform-browser';
```

```
import { AppRoutingModuleModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { RouterModule, Routes } from '@angular/router';
import { HttpClientModule } from '@angular/common/http';
import { HomepageComponent } from './homepage/homepage.component';
import { AdminLoginComponent } from './admin-login/admin-login.component';
import { ChangePasswordComponent } from './change-password/change-
password.component';
import { EmployeesComponent } from './employees/employees.component';
import { AddemployeesComponent } from './addemployees/addemployees.component';
import { ViewEmployeeComponent } from './view-employee/view-employee.component'

const routes:Routes=[
  {path:" ",component:AppComponent},
  {path:"home",component:HomepageComponent},
  {path:"login",component:AdminLoginComponent},
  {path:"changepassword",component:ChangePasswordComponent},
  {path:"employees",component:EmployeesComponent},
  {path:"addemployee",component:AddemployeesComponent},
  { path: 'view/:id', component: ViewEmployeeComponent } // Define a route with a
parameter ':id'

]

@NgModule({
  declarations: [
    AppComponent,
    HomepageComponent,
    AdminLoginComponent,
    ChangePasswordComponent,
    EmployeesComponent,
    AddemployeesComponent,
    ViewEmployeeComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModuleModule,
    FormsModule,
    HttpClientModule,
    ReactiveFormsModule,
    RouterModule.forRoot(routes)
```

```

    ],
    providers: [
      provideClientHydration()
    ],
    bootstrap: [AppComponent]
  })
}
export class AppModule { }
```

app.component.html

[illegible]

app.component.css

```
.active-link {
  font-weight: bold;
  color: white;
  text-decoration: none; /* Remove underline */
}

.top-bar {
  background-color: black;
  padding: 10px; /* Add padding for spacing */
}

a {
  text-decoration: none;
  margin-right: 10px; /* Adjust spacing between links */
  color: white; /* Set default text color */
}
```

app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
})
```

```

    styleUrls: ['./app.component.css']
  })
  export class AppComponent {
    title = 'EventManager';
  }

```

employeeservice.service.ts

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class EmployeeserviceService {
  private url='http://localhost:3000/employees';
  constructor(private http:HttpClient) { }

  private savedPassword: string = 'admin';

  getPassword(): string {
    return this.savedPassword;
  }

  setPassword(newPassword: string): void {
    this.savedPassword = newPassword;
  }

  addemployee(employee:any){
    return this.http.post(this.url,employee);
  }

  getEmployeeById(id: number): Observable<any> {
    const url = `${this.url}/${id}`;
    return this.http.get(url);
  }

  getemployee():Observable<any>{
    return this.http.get(this.url);
  }
}

```

```
deleteEmployee(id: number): Observable<any> {
  const url = `${this.url}/${id}`;
  return this.http.delete(url);
}

updateEmployee(id: number, updatedemployee: any): Observable<any> {
  const url = `${this.url}/${id}`;
  return this.http.put(url, updatedemployee);
}
}
```

homepage.component.html

```
<div class="container">
  <br>
  <h1>Event Management App</h1>
  <p>By Priyanka M V</p>
  <h1>Welcome Admin!!!</h1>
  <h3><a routerLink="/login" class="login-link">Click here to
login.</a>&nbsp;&nbsp;&nbsp;&nbsp;</h3>
  <br>
  
</div>
```

homepage.component.css

```
.container {
    text-align: center; /* Center-align content */
    background-image: url('https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcTj7emkxfyB-
DOexTHJdfe5PFDW0JDmPiqHxw&usqp=CAU');
    background-size: cover; /* Adjusts the background image to cover the entire
body */
    background-position: center; /* Center the background image */
    font-family: Arial, sans-serif; /* Example font family */
    width: auto;
    height: 800px;
}

.center-image {
    display: block; /* Set the image to display as a block element */
    margin: 0 auto; /* Set left and right margins to auto for horizontal
centering */
}

.login-link{
    color: black;
    text-decoration: none; /* Remove underline */
}
```

homepage.component.ts

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

@Component({
    selector: 'app-homepage',
    templateUrl: './homepage.component.html',
```



```

    styleUrl: './homepage.component.css'
  })
  export class HomepageComponent {
  }

```

admin-login.component.html

```

<title>Admin Login</title>
<link rel="stylesheet" href="admin-login.component.css">
<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<div class="container">
  <h2 class="mt-4 mb-4">Admin Login</h2>
  <form (ngSubmit)="login()" class="login-form">
    <div class="form-group">
      <label for="username">Username:</label>
      <input type="text" id="username" name="username" [(ngModel)]="username"
class="form-control">
    </div>
    <div class="form-group">
      <label for="password">Password:</label>
      <input type="password" id="password" name="password"
[(ngModel)]="password" class="form-control">
    </div>
    <button type="submit" class="btn btn-primary">Login</button>
  </form>
</div>

<!-- Bootstrap JS and jQuery -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"><
/script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>

```

admin-login.component.css

```
.container {
  text-align: center;
  background-image: url('https://wallpaperaccess.com/full/2886484.jpg');
  background-size: cover;
  background-position: center center;
  font-family: 'Times New Roman', Times, serif;
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  width: 100vw;
  height: 100vh;
}

.login-form {
  background-color: rgba(255, 255, 255, 0.8);
  padding: 20px;
  border-radius: 8px;
}

.login-form div {
  margin-bottom: 10px;
}
```

admin-login.component.ts

```
import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
  selector: 'app-admin-login',
  templateUrl: './admin-login.component.html',
  styleUrls: ['./admin-login.component.css']
})
export class AdminLoginComponent {
  username: string = '';
  password: string = '';

  constructor(private router: Router) {}

  login(): void {
```

```

// authentication logic
const isValid = this.authenticate();

if (isValid) {
  alert('Successfully Logged in');
  this.router.navigate(['/changepassword']); // Navigate to change password
page on successful login
} else {
  alert('Invalid credentials. Please try again.');// Show error message for
invalid credentials
}
}

authenticate(): boolean {
  return this.username === 'admin' && this.password === 'admin';
}
}

```

change-password.component.html

```

<div class="container">
  <div class="center-content">
    <h2>Change Password</h2>
    <form (ngSubmit)="changePassword()" class="password-form">
      <div>
        <label for="oldPassword">Old Password:</label>
        <input type="password" id="oldPassword" name="oldPassword"
[(ngModel)]="oldPassword">
      </div>
      <div>
        <label for="newPassword">New Password:</label>
        <input type="password" id="newPassword" name="newPassword"
[(ngModel)]="newPassword">
      </div>
      <br>
      <button type="submit">Change Password</button>
    </form><br>
    <h1>Or</h1><br>
    <h2>View Employees List</h2>
    <a routerLink="/employees" class="link">To view the list of employee details,
click here.</a>
  </div>
</div>

```

change-password.component.css

```
.container {
  text-align: center;
  background-image: url('https://www.itl.cat/pngfile/big/135-1358691_tablet-blue-purple-black.jpg');
  background-size: cover;
  background-position: center;
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.password-form button {
  background-color: #da38d2;
  color: white; /* White text color */
  padding: 10px 20px; /* Padding */
  border: none; /* No border */
  border-radius: 4px; /* Rounded corners */
  cursor: pointer; /* Cursor style */
  transition: background-color 0.3s ease; /* Smooth background color transition */
}

.password-form button:hover {
  background-color: #5a1a35;
}

.center-content {
  background-color: rgba(255, 255, 255, 0.8);
  padding: 20px;
  border-radius: 8px;
}

.password-form div {
  margin-bottom: 10px;
}

.link {
  display: block;
  margin-top: 20px;
}
```

```

color: black;
text-decoration: none; /* Remove underline */
}

```

change-password.component.ts

```

import { Component } from '@angular/core';
import { EmployeeserviceService } from '../employeeservice.service';

@Component({
  selector: 'app-change-password',
  templateUrl: './change-password.component.html',
  styleUrls: ['./change-password.component.css']
})
export class ChangePasswordComponent {
  oldPassword: string = '';
  newPassword: string = '';

  constructor(private EmployeeserviceService: EmployeeserviceService) {}

  changePassword(): void {
    const savedPassword = this.EmployeeserviceService.getPassword();

    if (this.oldPassword === savedPassword) {
      this.EmployeeserviceService.setPassword(this.newPassword);
      alert('Password changed successfully!');
    } else {
      alert('Failed to change password. Please try again.');
```

employees.component.html

```

<br><br>
<div class="container"><br><br><br><br><br>
  <button mat-raised-button color="primary" routerLink="/addemployee" class="mat-
button">Add employee</button>

  <table border="1" class="table-b">
    <thead>
      <tr>

```

```

        <th>ID</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Email</th>
        <th>Actions</th>
    </tr>
</thead>
<tbody>
    <tr *ngFor="let employee of employees">
        <td>{{ employee.id }}</td>
        <td>{{ employee.first_name }}</td>
        <td>{{ employee.last_name }}</td>
        <td>{{ employee.email }}</td>
        <!-- View button with routerLink -->
        <td>
            <button [routerLink]="'/view/' + employee.id" class="table-
button">view</button>
        </td>
    </tr>
</tbody>
</table>
</div>

```

employees.component.css

```

.container {
    background-image: url('https://cdn.wallpapersafari.com/9/55/kGvVJa.jpg');
    background-size: cover;
    background-position: center;
    display: flex;
    flex-direction: column;
    align-items: center;
    height: 100vh;
    margin: 0;
    padding: 0;
    position: relative; /* Ensure it starts from the top */
    top: -35px; /* Position from the top */
}
.table-button{
    background-color: #da38d2;
    color: white; /* White text color */
}

```

```

padding: 10px 20px; /* Padding */
border: none; /* No border */
border-radius: 4px; /* Rounded corners */
cursor: pointer; /* Cursor style */
transition: background-color 0.3s ease; /* Smooth background color transition
*/
}

.table-button:hover{
    background-color: #5a1a35;
}

.mat-button{
    background-color: #da38d2;
    color: white; /* White text color */
    padding: 10px 20px; /* Padding */
    border: none; /* No border */
    border-radius: 4px; /* Rounded corners */
    cursor: pointer; /* Cursor style */
    transition: background-color 0.3s ease; /* Smooth background color transition
*/
}

.mat-button:hover{
    background-color: #5a1a35;
}

/* Style for the table */
table {
    width: 80%; /* Adjust the width as needed */
    border-collapse: collapse;
    margin-bottom: 20px;
}

th, td {
    border: 1px solid #ddd;
    padding: 8px;
    text-align: left;
}

th {
    background-color: #f2f2f2;
}

```

```

/* Style for the button */
button {
  margin-bottom: 20px;
}

.table-b{
  background-color: rgba(255, 255, 255, 0.8);
  padding: 20px;
  border-radius: 8px;
  margin-bottom: 10px;
}

```

employees.component.ts

```

import { Component, OnInit } from '@angular/core';
import { EmployeeserviceService } from '../employeeservice.service';

@Component({
  selector: 'app-employees',
  templateUrl: './employees.component.html',
  styleUrls: ['./employees.component.css']
})
export class EmployeesComponent implements OnInit {
  employees: any[]=[];
  id=""
  first_name=""
  last_name=""
  email=""

  constructor(private employeeservice:EmployeeserviceService) { }

  ngOnInit(): void {
    this.getemployees();
  }

  getemployees():void{
    this.employeeservice.getemployee().subscribe(data=>{this.employees=data});
  }

  deletEmployee(id: number): void {
    this.employeeservice.deletEmployee(id).subscribe(() => {

```



```

    text-align: center;
    background-image: url('https://img.freepik.com/premium-photo/night-sky-is-
dark-blue-color-with-clouds-stars_662214-55728.jpg?w=360');
    background-size: contain;
    background-position: center;
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    position: relative; /* Ensure it starts from the top */
    top: -15px; /* Position from the top */
}

.add-form{
    display: flex;
    flex-direction: column;
    align-items: center;
    background-color: rgba(255, 255, 255, 0.8);
    padding: 50px;
    border-radius: 100px;
    margin-bottom: 10px;
}

.button-click{
    background-color: #da38d2;
    color: white; /* White text color */
    padding: 10px 20px; /* Padding */
    border: none; /* No border */
    border-radius: 4px; /* Rounded corners */
    cursor: pointer; /* Cursor style */
    transition: background-color 0.3s ease; /* Smooth background color transition */
}

.button-click:hover{
    background-color: #5a1a35;
}

```

addemployees.component.ts

```
import { Component } from '@angular/core';
import { EmployeeserviceService } from '../employeeservice.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-addemployees',
  templateUrl: './addemployees.component.html',
  styleUrls: ['./addemployees.component.css']
})
export class AddemployeesComponent {
  employees: any[] = [];
  first_name = "";
  last_name = "";
  email = "";

  constructor(
    private employeeservice: EmployeeserviceService,
    private router: Router
  ) {}

  addemployee(): void {
    console.log('First Name:', this.first_name);
    console.log('Last Name:', this.last_name);
    console.log('Email:', this.email);

    if (this.first_name.trim() && this.last_name.trim() && this.email.trim()) {
      const newEmployee = {
        first_name: this.first_name.trim(),
        last_name: this.last_name.trim(),
        email: this.email.trim()
      };

      this.employeeservice.addemployee(newEmployee).subscribe(_ => {
        alert('Employee added successfully!');
        this.router.navigate(['/employees']); // Navigate to employees list after
addition
      });
    } else {
      alert('Please fill in all the fields');
    }
  }
}
```

view-employee.component.html

```
<div class="container">
  <div *ngIf="employee" class="view-contain">
    <h2>Employee Details</h2>

    <form *ngIf="!isEditing" (ngSubmit)="toggleEditMode()">
      <p>ID: {{ employee.id }}</p>
      <p>First Name: {{ employee.first_name }}</p>
      <p>Last Name: {{ employee.last_name }}</p>
      <p>Email: {{ employee.email }}</p>

      <!-- Toggle Edit Mode Button -->
      <button mat-raised-button color="primary" routerLink="/employees">Employee
List</button>&nbsp;
      <button type="button" (click)="toggleEditMode()">Edit</button>&nbsp;
      <button (click)="deleteEmployee()">Delete</button>
    </form>

    <form *ngIf="isEditing" (ngSubmit)="updateEmployee()">
      <label for="firstName">First Name:</label>
      <input type="text" id="firstName" [(ngModel)]="employee.first_name"
name="firstName">
    <br><br>
      <label for="lastName">Last Name:</label>
      <input type="text" id="lastName" [(ngModel)]="employee.last_name"
name="lastName">
    <br><br>
      <label for="email">Email:</label>
      <input type="email" id="email" [(ngModel)]="employee.email" name="email">
    <br><br>
      <!-- Update Button -->
      <button type="submit">Update</button>&nbsp;&nbsp;&nbsp;
      <!-- Cancel Editing Button -->
      <button type="button" (click)="cancelEditMode()">Cancel</button>
    </form>
  </div>
</div>
```

view-employee.component.css

```
.container {
  background-image: url('https://img.freepik.com/premium-photo/sky-
night_646510-7882.jpg?size=626&ext=jpg&ga=GA1.1.1880011253.1700265600&semt=ais');
```

```

    background-size: cover;
    background-position: center;
    display: flex;
    flex-direction: column;
    align-items: center;
    height: 100vh;
    margin: 0;
    padding: 0;
    position: relative; /* Ensure it starts from the top */
    top: -35px; /* Position from the top */
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

.view-contain {
    width: 70%;
    padding: 20px;
    background-color: #e0d5d5;
    border-radius: 8px;
    text-align: center;
}

/* Existing styles for .container and .view-contain */

.view-contain button {
    margin: 5px;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

.view-contain button[type="submit"] {
    background-color: #4caf50; /* Green */
    color: white;
}

.view-contain button[type="button"],
.view-contain button:not([type]) {
    background-color: #f44336; /* Red */
    color: white;
}

```

```
.view-contain button:hover {
  opacity: 0.8;
}
.view-contain form input[type="email"] {
  width: 200px;
}
```

view-employee.component.ts

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { EmployeeserviceService } from '../employeeservice.service';

@Component({
  selector: 'app-view-employee',
  templateUrl: './view-employee.component.html',
  styleUrls: ['./view-employee.component.css']
})
export class ViewEmployeeComponent implements OnInit {
  employee: any;
  isEditing = false;
  originalEmployee: any;

  constructor(
    private route: ActivatedRoute,
    private router: Router,
    private employeeService: EmployeeserviceService
  ) {}

  ngOnInit(): void {
    this.route.params.subscribe(params => {
      const employeeId = params['id'];
      this.employeeService.getEmployeeById(employeeId).subscribe(data => {
        this.employee = data;
        this.originalEmployee = { ...data }; // Create a copy for editing
      });
    });
  }

  toggleEditMode(): void {
    this.isEditing = !this.isEditing;
  }
}
```

```
cancelEditMode(): void {
  this.employee = { ...this.originalEmployee }; // Reset to original values
  this.isEditing = false;
}

updateEmployee(): void {
  if (this.employee) {
    this.employeeService.updateEmployee(this.employee.id,
this.employee).subscribe(updatedEmployee => {
      this.employee = updatedEmployee;
      this.originalEmployee = { ...updatedEmployee }; // Update original values
      this.isEditing = false;
    });
  }
}

deleteEmployee(): void {
  if (this.employee) {
    this.employeeService.deleteEmployee(this.employee.id).subscribe(() => {
      // Redirect to the employee list or another page after deletion
      this.router.navigate(['/employees']);
    });
  }
}
```