

Source code for Event Management App:

db.json

```
{
  "employees": [
    {
      "id": 1,
      "First_name": "Sebastian",
      "Last_name": "Eschweiler",
      "employee_email": "sebastian@codingthesmartway.com"
    },
    {
      "id": 2,
      "First_name": "Steve",
      "Last_name": "Palmer",
      "employee_email": "steve@codingthesmartway.com"
    },
    {
      "id": 3,
      "First_name": "Ann",
      "Last_name": "Smith",
      "employee_email": "ann@codingthesmartway.com"
    },
    {
      "id": 4,
      "First_name": "Ravipati",
      "Last_name": "Harshitha",
      "employee_email": "harshitha@gmail.com"
    }
  ],
  "admin": [
    {
      "email": "harshitha@gmail.com",
      "password": "abc",
      "id": 1
    }
  ]
}
```

Index.html

```
<!doctype html>
<html lang="en">
<head>
```

```

<meta charset="utf-8">
<title>EventManagerApp</title>
<base href="/">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" type="image/x-icon" href="favicon.ico">
<link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
  <script
src="https://cdn.jsdelivrivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
u10knCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTP00mMi466C8"
crossorigin="anonymous"></script>
  <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">

</head>
<body>
  <app-root></app-root>
</body>
</html>

```

app

app.component.html

```

<!--The content below is only a placeholder and can be replaced.-->
<div style="text-align:center;margin: auto;text-shadow: 3px 4px 7px
rgba(81,67,21,0.8);" >
  <h2>
    {{ title }}
    <br>
  </h2>
</div>

<router-outlet></router-outlet>

```

app.component.ts

```

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

```

```

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'EventManagerApp';
}

```

app.module.ts

```

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { ReactiveFormsModule } from '@angular/forms';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { AddEmpComponent } from './add-emp/add-emp.component';
import { EditEmpComponent } from './edit-emp/edit-emp.component';
import { ListEmpComponent } from './list-emp/list-emp.component';
import { HttpClientModule } from '@angular/common/http';
import { HomeComponent } from './home/home.component';
import { AdminComponent } from './admin/admin.component';
import { EmpDetailComponent } from './emp-detail/emp-detail.component';
import { AdminEditComponent } from './admin-edit/admin-edit.component';
import { DashboardComponent } from './dashboard/dashboard.component';
@NgModule({
  declarations: [
    AppComponent,
    AddEmpComponent,
    EditEmpComponent,
    ListEmpComponent,
    HomeComponent,
    AdminComponent,
    EmpDetailComponent,
    AdminEditComponent,
    DashboardComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    ReactiveFormsModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

app-routing.module.ts

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { ListEmpComponent } from './list-emp/list-emp.component';
import { AddEmpComponent } from './add-emp/add-emp.component';
import { EditEmpComponent } from './edit-emp/edit-emp.component';
import { HomeComponent } from './home/home.component';
import { AdminComponent } from './admin/admin.component';
import { EmpDetailComponent } from './emp-detail/emp-detail.component';
import { AdminEditComponent } from './admin-edit/admin-edit.component';
import { DashboardComponent } from './dashboard/dashboard.component';
const routes: Routes = [
  { path: '', component:AdminComponent },
  { path: 'detail', component:EmpDetailComponent },
  { path: 'dash', component:DashboardComponent },
  { path: 'admin', component:AdminEditComponent },
  {path: 'home', component:HomeComponent },
  { path: 'list-emp', component: ListEmpComponent },
  { path: 'add-emp', component: AddEmpComponent },
  { path: 'update-emp', component: EditEmpComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule],
  declarations: []
})
export class AppRoutingModule { }

```

employee.model.ts

```

export class Employee {
  id?: number;
  employee_email?:string;
  First_name?: string;
  Last_name?: string;
}
export class admin {
  id=1;
  email?: string;
  password?: string;
}

```

employee.service.ts

```

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

```

```

import { HttpClient } from '@angular/common/http';
import { Employee } from './employee.model';
import { admin } from './employee.model';
@Injectable({
  providedIn: 'root'
})
export class EmployeeService {
  constructor(private http: HttpClient) { }
  baseUrl = 'http://localhost:3000/employees/';
  adminurl='http://localhost:3000/admin/';
  getEmployees() {
    return this.http.get<Employee[]>(this.baseUrl);
  }
  deleteEmployees(id: number) {
    return this.http.delete<Employee[]>(this.baseUrl + id);
  }
  createUser(employee: Employee) {
    return this.http.post(this.baseUrl, employee);
  }
  getEmployeeById(id: number) {
    return this.http.get<Employee>(this.baseUrl + id);
  }
  updateEmployee(employee: Employee) {
    return this.http.put(this.baseUrl + employee.id, employee);
  }
  updateadmin(admin: admin) {
    return this.http.put(this.adminurl + 1, admin);
  }
  getadminById(id:number) {
    return this.http.get<admin>(this.adminurl + 1);
  }
}

```

app/add-emp

add-emp.component.html

```

<div class="col-md-6">
  <h3 class="text-center" style="text-decoration:
underline;color:blue;text-shadow: 0 0 3px #00e1ff, 0 0 5px
#e6e6f1;">{{empformlabel}}</h3>
  <form [formGroup]="addForm" novalidate class="form" style="margin: auto;
padding-top: 5px; padding-bottom: 10px;width: 400px;">
    <div class="form-group" style="padding: 12px 20px;">
      <label for="empId">Employee Id:</label>

```

```

        <input type="number" formControlName="id" placeholder="Id"
name="empId" class="form-control" id="empId">
    </div>

    <div class="form-group" style=" padding: 12px 20px;">
        <label for="empName">First Name:</label>
        <input formControlName="First_name" placeholder="First Name"
name="FName" class="form-control" id="FName">
        <div class="alert alert-danger"
*ngIf="addForm.get('First_name')!.hasError('required') &&
addForm.get('First_name')!.touched">
            First Name is required
        </div>
    </div>
    <div class="form-group" style=" padding: 12px 20px;">
        <label for="empName">Last Name:</label>
        <input formControlName="Last_name" placeholder="Last Name"
name="LName" class="form-control" id="LName">
        <div class="alert alert-danger"
*ngIf="addForm.get('Last_name')!.hasError('required') &&
addForm.get('Last_name')!.touched">
            Last Name is required
        </div>
    </div>
    <div class="form-group" style=" padding: 12px 20px;">
        <label for="empEmail">Email ID:</label>
        <input formControlName="employee_email" placeholder="Employee Email
ID" name="empEmail" class="form-control" id="empEmail">
        <div class="alert alert-danger"
*ngIf="addForm.get('employee_email')!.hasError('required') &&
addForm.get('employee_email')!.touched">
            Employee Email ID is required
        </div>
    </div>
    <button class="btn btn-success" [disabled]='addForm.invalid'
*ngIf="btnvisibility" (click)="onSubmit()">Save</button>
    <button class="btn btn-success" [disabled]='addForm.invalid'
*ngIf="!btnvisibility" (click)="onUpdate()">Update</button>

    <p style=" padding: 12px 20px;">Form value: {{ addForm.value | json
}}</p>
    </form>
</div>

```

add-emp.component.ts

```

import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { EmployeeService } from '../employee.service';

```

```

import { Router } from '@angular/router';
@Component({
  selector: 'app-add-emp',
  templateUrl: './add-emp.component.html',
  styleUrls: ['./add-emp.component.css']
})
export class AddEmpComponent implements OnInit {

  empformlabel = 'Add Employee Form';
  empformbtn = 'Save';
  constructor(private formBuilder: FormBuilder, private router: Router,
private empService: EmployeeService) {}
  editForm: FormGroup;
  addForm: FormGroup;
  btnvisibility = true;
  ngOnInit() {

    this.addForm = this.formBuilder.group({
      id: [],
      First_name: ['', Validators.required],
      Last_name: ['', Validators.required],
      employee_email: ['', Validators.required],

    });

  }
  onSubmit() {
    console.log('Create fire');
    this.empService.createUser(this.addForm.value)
      .subscribe(data => {
        this.router.navigate(['list-emp']);
      },
      error => {
        alert(error);
      });
  }
  onUpdate() {

    console.log('Update fire');
    this.empService.updateEmployee(this.editForm.value).subscribe(data => {
      this.router.navigate(['list-emp']);
    },
    error => {
      alert(error);
    });
  }
}

```

app/admin

admin.component.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Page Title</title>
</head>
<body style="padding:40px ">

  <div >
  <div class="jumbotron">
    <div class="container">
      <div class="row">
        <div class="col-md-6 offset-md-3">
          <h2 style=" text-decoration: underline;color:blue;text-shadow:
0 0 3px #00e1ff, 0 0 5px #e6e6f1;">Admin Login Form</h2>
          <form [formGroup]="registerForm" (ngSubmit)="onSubmit()" >
            <div class="form-group">

              <div class="form-group">
                <label>Email</label>
                <input type="text" formControlName="email" class="form-
control" [ngClass]="{ 'is-invalid': submitted && f.email.errors }" />
                <div *ngIf="submitted && f.email.errors" class="invalid-
feedback">

                  <div *ngIf="f.email.errors.required">Email is
required</div>

                  <div *ngIf="f.email.errors.email">Email must be a
valid email address</div>
                </div>
              </div>
              <div class="form-group">
                <label>Password</label>
                <input type="password" formControlName="password"
class="form-control" [ngClass]="{ 'is-invalid': submitted && f.password.errors
}" />
                <div *ngIf="submitted && f.password.errors"
class="invalid-feedback">

                  <div *ngIf="f.password.errors.required">Password is
required</div>
```



```

}

get f() { return this.registerForm.controls; }
baseUrl = 'http://localhost:3000/admin/';

onSubmit() {
  this.http.get<any>(this.baseUrl)
    .subscribe(res=>{
      const u=res.find((a:any)=>{
        return a.email===this.registerForm.value.email &&
a.password===this.registerForm.value.password
      })
      if(u){
        alert("Login success");
        this.registerForm.reset();
        this.route.navigate(['/dash'])
      }else{
        alert("Invalid Credentials")
      }
    })
}
}
}

```

app/admin-edit

admin-edit.component.html

```

<div class="col-md-6">
  <h2 class="text-center" style=" text-decoration:
underline;color:blue;text-shadow: 0 0 3px #00e1ff, 0 0 5px
#e6e6f1;">{{empformlabel}}</h2>
  <form [formGroup]="editForm" novalidate class="form" style="margin: auto;
padding-top: 5px; padding-bottom: 10px;width: 400px;">
    <div class="form-group" style=" padding: 12px 20px;">
      <label for="email">Admin Email:</label>
      <input formControlName="email" placeholder="Email" name="email"
class="form-control" id="email">
      <div class="alert alert-danger"
*ngIf="editForm.get('email')!.hasError('required') ">
        Email is required
      </div>
    </div>
    <div class="form-group" style=" padding: 12px 20px;">
      <label for="password">Password:</label>
      <input formControlName="password" placeholder="Password"
name="password" class="form-control" id="password">
      <div class="alert alert-danger"
*ngIf="editForm.get('password')!.hasError('required') ">

```

```

        Password is required
      </div>
    </div>
    <button class="btn btn-success"
[disabled]='editForm.invalid' (click)="onUpdate()">Update</button>
    <p>Form value: {{ editForm.value | json }}</p>
  </form>
</div>

```

admin-edit.component.ts

```

import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { EmployeeService } from '../employee.service';
import { Router } from '@angular/router';
@Component({
  selector: 'app-admin-edit',
  templateUrl: './admin-edit.component.html',
  styleUrls: ['./admin-edit.component.css']
})
export class AdminEditComponent implements OnInit {

  empformlabel = 'Edit Admin';
  empformbtn = 'Update';
  constructor(private formBuilder: FormBuilder, private router: Router,
private empService: EmployeeService) {
  }
  addForm: FormGroup;
  editForm: FormGroup;
  ngOnInit() {

    this.editForm = this.formBuilder.group({
      email: ['', Validators.required],
      password: ['', [Validators.required, Validators.maxLength(14)]]
    });

    const empid = Number(localStorage.getItem('editEmpId'));
    if (+empid > 0) {
      this.empService.getadminById(+empid).subscribe(data => {
        this.editForm.patchValue(data);
      });
    }
  }

  onUpdate() {

    console.log('Update fire');
  }
}

```

```

    this.empService.updateadmin(this.editForm.value).subscribe(data => {
        this.router.navigate(['']);
    },
        error => {
            alert(error);
        });
    }
}

```

App/dashboard

dashboard.component.html

```

<app-home></app-home>
<h3 style="text-align: center;font-family:'Times New Roman', Times,
serif;font-weight:800;font-style:oblique;">Welcome to Event Management App
!!</h3>


```

dashboard.component.ts

```

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

@Component({
  selector: 'app-dashboard',
  templateUrl: './dashboard.component.html',
  styleUrls: ['./dashboard.component.css']
})
export class DashboardComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }
}

```

```
}
```

app/edit-emp

edit-emp.component.html

```
<div class="col-md-6">
  <h2 class="text-center" style=" text-decoration:
underline;color:blue;text-shadow: 0 0 3px #00e1ff, 0 0 5px
#e6e6f1;">{{empformlabel}}</h2>
  <form [formGroup]="editForm" novalidate class="form" style="margin: auto;
padding-top: 5px; padding-bottom: 10px;width: 400px;">
    <div class="form-group" style=" padding: 12px 20px;">
      <label for="empId">Employee Id:</label>
      <input type="number" formControlName="id" placeholder="Id"
name="empId" class="form-control" id="empId">
    </div>
    <div class="form-group" style=" padding: 12px 20px;">
      <label for="empName">First Name:</label>
      <input formControlName="First_name" placeholder="First Name"
name="FName" class="form-control" id="FName">
      <div class="alert alert-danger"
*ngIf="editForm.get('First_name')!.hasError('required') &&
addForm.get('First_name')!.touched">
        First Name is required
      </div>
    </div>
    <div class="form-group" style=" padding: 12px 20px;">
      <label for="empName">Last Name:</label>
      <input formControlName="Last_name" placeholder="Last Name"
name="LName" class="form-control" id="LName">
      <div class="alert alert-danger"
*ngIf="editForm.get('Last_name')!.hasError('required') &&
addForm.get('Last_name')!.touched">
        Last Name is required
      </div>
    </div>
    <div class="form-group" style=" padding: 12px 20px;">
      <label for="empEmail">Email ID:</label>
      <input formControlName="employee_email" placeholder="Employee Email
ID" name="empEmail" class="form-control" id="empEmail">
      <div class="alert alert-danger"
*ngIf="editForm.get('employee_email')!.hasError('required') &&
addForm.get('employee_email')!.touched">
        Employee Email ID is required
      </div>
    </div>
  </form>
</div>
```

```

    </div>
    <button class="btn btn-success"
[disabled]='editForm.invalid' (click)="onUpdate()">Update</button>

    <p>Form value: {{ editForm.value | json }}</p>
  </form>
</div>

```

edit-emp.component.ts

```

import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { EmployeeService } from '../employee.service';
import { Router } from '@angular/router';
@Component({
  selector: 'app-edit-emp',
  templateUrl: './edit-emp.component.html',
  styleUrls: ['./edit-emp.component.css']
})
export class EditEmpComponent implements OnInit {

  empformlabel = 'Edit Employee';
  empformbtn = 'Update';
  constructor(private formBuilder: FormBuilder, private router: Router,
private empService: EmployeeService) {
  }
  addForm: FormGroup;
  editForm: FormGroup;
  ngOnInit() {

    this.editForm = this.formBuilder.group({
      id: [],
      employee_name: ['', Validators.required],
      employee_email: ['', Validators.required],
      employee_salary: ['', [Validators.required, Validators.maxLength(9)]],
      employee_age: ['', [Validators.required, Validators.maxLength(3)]]
    });

    const empid = Number(localStorage.getItem('editEmpId'));
    if (+empid > 0) {
      this.empService.getEmployeeById(+empid).subscribe(data => {
        this.editForm.patchValue(data);
      });
    }
  }

  onUpdate() {

    console.log('Update fire');
  }
}

```

```

    this.empService.updateEmployee(this.editForm.value).subscribe(data => {
      this.router.navigate(['list-emp']);
    },
    error => {
      alert(error);
    });
  }
}

```

app/emp-detail

emp-detail.component.html

```

<app-home></app-home>
<br>
<br>
<div style="border: 2px solid black;background-color: rgb(197, 163, 230);width:600px; padding-top: 5px; padding-bottom: 10px; margin-left: auto;margin-right: auto;padding: 10px;">
  <p>Detail of </p><h3>{{employeeess.First_name}}</h3>
</div>
  <table class="table table-bordered " style="border: 2px solid black;width:600px; padding-top: 10px; padding-bottom: 20px; margin-left: auto;margin-right: auto;padding: 20px;" *ngIf="employeeess">
    <tbody>
      <tr>
        <td>Id</td>
        <td>{{employeeess.id}}</td>
      </tr>
      <tr>
        <td>First name</td>
        <td>{{employeeess.First_name}}</td>
      </tr>
      <tr>
        <td>Last name</td>
        <td>{{employeeess.Last_name}}</td>
      </tr>
      <tr>
        <td>Email ID</td>
        <td>{{employeeess.employee_email}}</td>
      </tr>
      <tr style=" padding: 20px;padding-top: 10px; padding-bottom: 20px;">
        <button (click)="deleteEmp(employeeess)" routerLink="/list-emp" class="btn btn-danger"style="border: 2px solid black;border-spacing: 30px;background-color:red"> Delete</button>

```

```

        <button (click)="editEmp(employeeess)" style="margin-left: 20px;border:
2px solid black;background-color:rgb(161, 153, 153)" class="btn btn-
secondary"> Edit</button>
        <button routerLink="/list-emp" style="margin-left: 20px;border: 2px
solid black;background-color:rgb(0, 247, 255)" class="btn btn-primary">
Employees List</button>

    </tr>
</tbody>

</table>

```

emp-detail.component.ts

```

import { Component, OnInit } from '@angular/core';
import { EmployeeService } from '../employee.service';
import { Employee } from '../employee.model';
import { Router } from '@angular/router';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';

@Component({
  selector: 'app-emp-detail',
  templateUrl: './emp-detail.component.html',
  styleUrls: ['./emp-detail.component.css']
})
export class EmpDetailComponent implements OnInit {

  employeeID: number;
  employees: Employee[];
  employeeess:Employee;

  constructor(private empService: EmployeeService,private formBuilder:
FormBuilder, private router: Router, ) {

  }
  editForm: FormGroup;
  ngOnInit() {
    this.getEmployeeByID();
  }
  empid = Number(localStorage.getItem('editEmpId'));
  getEmployeeByID() {

    this.empService.getEmployeeById(this.empid)
      .subscribe((data: Employee) => {

```



```

        this.employeeess = data;
    });

}

deleteEmp(employee: Employee): void {
    this.empService.deleteEmployees(employee.id!)
        .subscribe(data => {
            this.employees = this.employees.filter(u => u !== employee);
        });
}

editEmp(employee: Employee): void {
    localStorage.removeItem('editEmpId');
    localStorage.setItem('editEmpId', employee.id!.toString());
    this.router.navigate(['update-emp']);
}

addEmp(): void {
    localStorage.removeItem('editEmpId');
    this.router.navigate(['add-emp']);
}
}

```

app/home

home.component.html

```

<ul class="nav nav-pills navbar-dark bg-dark " routerLinkActive="active">
    <li class="nav-item">
        <a class="nav-link active" aria-current="page"
routerLink="/dash">Home</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" routerLink="/list-emp">Employees</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" routerLink="/admin">Settings</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="#">Logout</a>
    </li>
</ul>

```

home.component.ts

```

import { Component, OnInit } from '@angular/core';
import { admin } from '../employee.model';
import { Router } from '@angular/router';
import { EmployeeService } from '../employee.service';
@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent implements OnInit {

  constructor(private empService: EmployeeService, private router: Router) { }
  ngOnInit(): void {
  }
  editadminn(admin: admin): void {
    localStorage.removeItem('editEmpId');
    localStorage.setItem('editEmpId', admin.id!.toString());
    this.router.navigate(['admin']);
  }
}

```

app/list-emp

list-emp.component.html

```

<app-home></app-home>
<br><br>
<div class="col-md-12">
  <button (click)="addEmp()" class="btn btn-secondary"> Add
Employee</button>
  <br><br>
  <h3 style="color:rgb(52, 164, 172)"> Employees List</h3>
  <div class="table-responsive table-container">
    <table class="table" style="width: 1200px; margin-left: auto;margin-right:
auto;">
      <thead>
        <tr>
          <th>Id</th>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Email ID</th>
        </tr>
      </thead>
      <tbody>
        <tr *ngFor="let emp of employees">
          <td class="hidden">{{emp.id}}</td>
          <td>{{emp.First_name}}</td>

```

```

        <td>{{emp.Last_name}}</td>
        <td>{{emp.employee_email}}</td>
        <td>
            <button (click)="detailEmp(emp)" class="btn btn-info"> View
Detail</button>

        </td>
    </tr>
</tbody>
</table>
</div>
</div>

```

list-emp.component.ts

```

import { Component, OnInit } from '@angular/core';
import { EmployeeService } from '../employee.service';
import { Employee } from '../employee.model';
import { Router } from '@angular/router';
@Component({
  selector: 'app-list-emp',
  templateUrl: './list-emp.component.html',
  styleUrls: ['./list-emp.component.css']
})
export class ListEmpComponent implements OnInit {

  employees: Employee[];
  constructor(private empService: EmployeeService, private router: Router, ) {

  }

  ngOnInit() {
    this.empService.getEmployees()
      .subscribe((data: Employee[]) => {
        this.employees = data;
      });
  }

  detailEmp(employee: Employee): void {
    localStorage.removeItem('editEmpId');
    localStorage.setItem('editEmpId', employee.id!.toString());
    this.router.navigate(['/detail']);
  }

  addEmp(): void {
    localStorage.removeItem('editEmpId');
    this.router.navigate(['add-emp']);
  }
}

```

