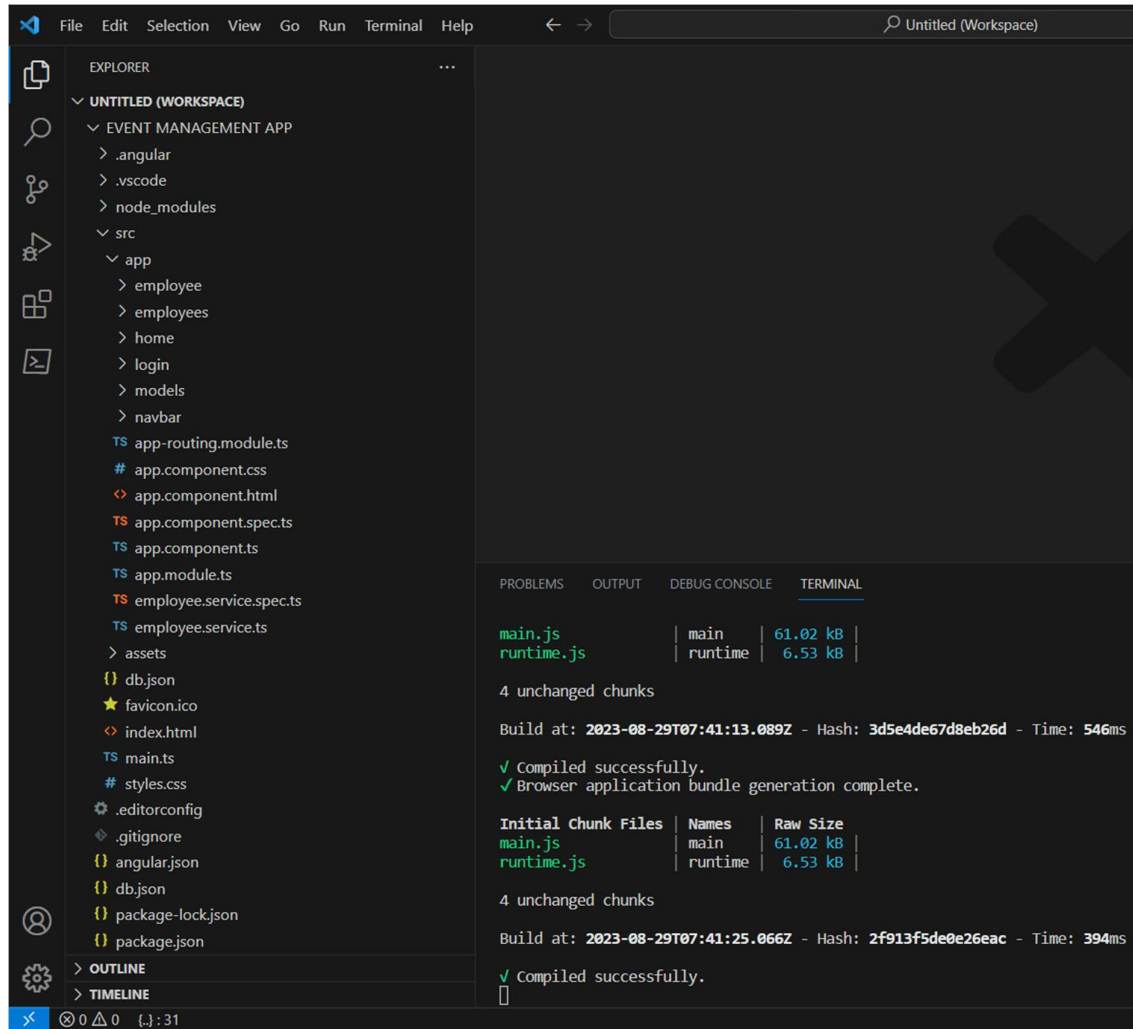


EMPLOYEE MANAGEMENT APP –SOURCE CODE

APP-structure



SOURCE CODES

Login.html

```
<head>
  <title>Bootstrap Example</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min
.js"></script>
</head>

<div class="w-100 mt-5 container">
  <form
    #loginForm="ngForm"
    (ngSubmit)="loginForm?.form?.valid && onLogin()"
    class="w-25 mx-auto"
  >
    <h4>Login</h4>
    <div class="my-3">
      <label for="email" class="form-label">Email</label>
      <input
        type="email"
        class="form-control"
        name="email"
        id="email"
        placeholder="Email"
        [(ngModel)]="email"
        required
      />
      <div
        *ngIf="
          loginForm.controls['email']?.invalid &&
          loginForm.controls['email']?.touched
        "
      >
        <span *ngIf="loginForm.controls['email'].errors?.['required']"
class="mt-4 text-sm text-danger"
        >Email is required.</span>
      >
    </div>
  </div>
  <div class="my-3">
    <label for="password" class="form-label">Password</label>
    <input
      type="password"
```

```

        class="form-control"
        name="password"
        id="password"
        placeholder="Password"
        [(ngModel)]="password"
        required
    />
    <div
        *ngIf="
            loginForm.controls['password']?.invalid &&
            loginForm.controls['password']?.touched
        "
    >
        <span *ngIf="loginForm.controls['password'].errors?.['required']"
class="text-sm text-danger"
        >Password is required.</span>
    >
    </div>
</div>
    <button type="submit" class="btn btn-primary"
[disabled]="loginForm?.form?.invalid">Submit</button>
</form>
</div>

```

Login.component.ts

```

import { Component, OnInit } from '@angular/core';
import Employee from '../models/Employee';
import { EmployeeService } from '../employee.service';
import { NgForm } from '@angular/forms';
import { Router } from '@angular/router';

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

    public email:string = "";
    public password:string = "";

    constructor(private employeeService: EmployeeService, private router:
Router) {}

```

```

ngOnInit(): void {

}

onLogin() {
    if(this.email === this.employeeService.adminEmail && this.password ===
this.employeeService.adminPassword) {
        localStorage.setItem("admin", this.email);
        this.email = "";
        this.password = "";
        this.router.navigate(['home']);
        this.employeeService.setIsLoggedIn(true);
    }
    else {
        this.employeeService.setIsLoggedIn(false);
        alert("Incorrect credentials");
    }
}
}

```

home.html

```

<head>
  <title>Bootstrap Example</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min
.js"></script>
</head>
<body>
  <div class="skeleton">
    <h1 class="centered-text">Wellcome To Event-Management</h1>
    <h2 class="author-text">by-Raheem.SK</h2>
  </div>
  <style>
.skeleton {
  position: fixed;
  top: 50px;
  left: 0;

```

```
width: 100%;
height: 100%;
background: linear-gradient(90deg, rgba(56, 190, 217, 0.749) 0%, rgba(230,
121, 199, 0.7) 50%, rgb(163, 231, 119) 90%); /* Creates a gradient that fades
in and out */
animation: fadeInOut 1.5s ease-in-out infinite; /* Animates the fading
effect */
}
.centered-text {
position: absolute;
top: 40%;
left: 50%;
transform: translate(-50%, -50%);
text-align: center;
font-size: 44px;
font-weight: bold;
}
.author-text {
position: absolute;
top: 85%;
left: 85%;
transform: translate(-50%, -50%);
text-align: center;
font-size: 36px;
font-weight: bold;
}
@keyframes fadeInOut {
0% {
opacity: 0.7;
}
50% {
opacity: 0.5;
}
100% {
opacity: 0.7;
}
}
</style>
```

Home.component.ts

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

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent {

}
```

Employees.html

```
<div class="mt-4 container">
  <button
    class="btn btn-secondary"
    data-bs-toggle="modal"
    data-bs-target="#addEmployeeModal"
  >
    Add New Employee
  </button>

  <!-- Modal -->
  <div
    class="modal fade"
    id="addEmployeeModal"
    tabindex="-1"
    aria-labelledby="addEmployeeModalLabel"
    aria-hidden="true"
  >
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          <h1 class="modal-title fs-5" id="addEmployeeModalLabel">
            Add Employee
          </h1>
          <button
            type="button"
            class="btn-close"
            data-bs-dismiss="modal"
            aria-label="Close"
          >
```

```
></button>
</div>
<div class="modal-body">
  <form #addEmployeeForm="ngForm"
    (ngSubmit)="addEmployeeForm.form.valid && onSubmit()">
    <div class="mb-3">
      <label for="id" class="form-label">ID</label>
      <input
        type="number"
        class="form-control"
        id="id"
        [(ngModel)]="employeeModel.id"
        name="id"
        #id="ngModel"
        required
      />
    </div>
    <div class="mb-3">
      <label for="first_name" class="form-label">First Name</label>
      <input
        type="text"
        class="form-control"
        id="first_name"
        [(ngModel)]="employeeModel.first_name"
        name="first_name"
        #first_name="ngModel"
        required
      />
    </div>
    <div class="mb-3">
      <label for="last_name" class="form-label">Last Name</label>
      <input
        type="text"
        class="form-control"
        id="last_name"
        [(ngModel)]="employeeModel.last_name"
        name="last_name"
        #last_name="ngModel"
        required
      />
    </div>
    <div class="mb-3">
      <label for="email" class="form-label">Email</label>
      <input
        type="email"
        class="form-control"
        id="email"
        [(ngModel)]="employeeModel.email"

```

```

        name="email"
        #email="ngModel"
        required
    />
</div>
<button type="submit" class="btn btn-primary" data-bs-
dismiss="modal">Submit</button>
<button
    type="button"
    class="mx-4 btn btn-secondary"
    data-bs-dismiss="modal"
>
    Close
</button>
</form>
</div>
</div>
</div>
</div>

<!-- Employee details table -->
<table class="mt-4 table table-light">
    <thead>
        <tr>
            <th scope="col">#</th>
            <th scope="col">First Name</th>
            <th scope="col">Last Name</th>
            <th scope="col">Email</th>
            <th scope="col">Action</th>
        </tr>
    </thead>

    <tbody>
        <tr *ngFor="let employee of employees">
            <th scope="row">{{ employee.id }}</th>
            <td>{{ employee.first_name }}</td>
            <td>{{ employee.last_name }}</td>
            <td>{{ employee.email }}</td>
            <td>
                <button class="btn btn-secondary"
(click)="onDetailsShow(employee.id)">View Details</button>
            </td>
        </tr>
    </tbody>
</table>
</div>

```


Employees.component.ts

```
import { Component, OnInit } from '@angular/core';
import Employee from '../models/Employee';
import { EmployeeService } from '../employee.service';
import { ActivatedRoute, Router } from '@angular/router';

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

  public employees:Employee[] = [];
  public employeeModel = new Employee(1,"","","");

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

  }

  ngOnInit(): void {
    this.employeeService.loadAllEmployees().subscribe((data:Employee[])=> {
      this.employees = data;
    })
  }

  onSubmit() {
    this.employeeService.addEmployee(this.employeeModel).subscribe((data:Employee)=> {
      this.employees.push(data);
    });
    this.employeeModel = new Employee(1,"","","");
  }

  onDetailsShow(id:number) {
    this.router.navigate([`employees`, id]);
  }
}
```

Employee.html

```
<div class="w-100 mt-4 container">
  <div class="card" *ngIf="employee">
    <div class="card-header p-3">
      Details of <b>{{ employee.first_name }} {{ employee.last_name }}</b>
    </div>
    <ul class="list-group list-group-flush">
      <li
        class="list-group-item d-flex flex-row justify-content-between align-items-center"
      >
        <b>Employee Id</b>
        <p>{{ employee.id }}</p>
      </li>
      <li
        class="list-group-item d-flex flex-row justify-content-between align-items-center"
      >
        <b>Employee First Name</b>
        <p>{{ employee.first_name }}</p>
      </li>
      <li
        class="list-group-item d-flex flex-row justify-content-between align-items-center"
      >
        <b>Employee Last Name</b>
        <p>{{ employee.last_name }}</p>
      </li>
      <li
        class="list-group-item d-flex flex-row justify-content-between align-items-center"
      >
        <b>Employee Email</b>
        <p>{{ employee.email }}</p>
      </li>
    </ul>
    <div class="card-footer d-flex flex-row p-3">
      <a routerLink="/employees" class="btn btn-primary">Employee List</a>
      <button
        class="btn btn-danger mx-4"
        (click)="onEmployeeDelete(employee.id)"
      >
        Delete
      </button>
      <button
        class="btn btn-secondary"
        data-bs-toggle="modal"
        data-bs-target="#updateEmployeeModal"
      >
```

```

        >
        Update
    </button>
</div>
</div>
</div>
</div>

<!-- Update employee Modal -->
<div
    *ngIf="employee"
    class="modal fade"
    id="updateEmployeeModal"
    tabindex="-1"
    aria-labelledby="updateEmployeeModalLabel"
    aria-hidden="true"
>
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5" id="updateEmployeeModalLabel">
                    Update Employee
                </h1>
                <button
                    type="button"
                    class="btn-close"
                    data-bs-dismiss="modal"
                    aria-label="Close"
                ></button>
            </div>
            <div class="modal-body">
                <form
                    #updateEmployeeForm="ngForm"
                    (ngSubmit)="updateEmployeeForm.form.valid && onEmployeeUpdate()"
                >
                    <div class="mb-3">
                        <label for="id" class="form-label">ID</label>
                        <input
                            type="number"
                            class="form-control"
                            id="id"
                            [(ngModel)]="employee.id"
                            name="id"
                            #id="ngModel"
                            required
                        />
                    </div>
                    <div class="mb-3">
                        <label for="first_name" class="form-label">First Name</label>

```

```

        <input
            type="text"
            class="form-control"
            id="first_name"
            [(ngModel)]="employee.first_name"
            name="first_name"
            #first_name="ngModel"
            required
        />
    </div>
    <div class="mb-3">
        <label for="last_name" class="form-label">Last Name</label>
        <input
            type="text"
            class="form-control"
            id="last_name"
            [(ngModel)]="employee.last_name"
            name="last_name"
            #last_name="ngModel"
            required
        />
    </div>
    <div class="mb-3">
        <label for="email" class="form-label">Email</label>
        <input
            type="email"
            class="form-control"
            id="email"
            [(ngModel)]="employee.email"
            name="email"
            #email="ngModel" required />
    </div>
    <button type="submit" class="btn btn-primary" data-bs-
dismiss="modal">
        Submit
    </button>
    <button
        type="button"
        class="mx-4 btn btn-secondary"
        data-bs-dismiss="modal">
        Close
    </button>
</form>
</div>
</div>
</div>
</div>

```

employee.component.ts

```
import { Component, OnInit } from '@angular/core';
import Employee from '../models/Employee';
import { ActivatedRoute, Router } from '@angular/router';
import { EmployeeService } from '../employee.service';

@Component({
  selector: 'app-employee',
  templateUrl: './employee.component.html',
  styleUrls: ['./employee.component.css']
})
export class EmployeeComponent implements OnInit {
  public employee!: Employee;

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

  }

  ngOnInit() {
    this.route.params.subscribe(params => {
      this.employeeService.getEmployee(+params["id"]).subscribe((data:Employee
    )=> {
        this.employee = data;
      })
    });
  }

  onEmployeeDelete(id:number) {
    this.employeeService.deleteEmployee(id).subscribe();
    this.router.navigate(['employees']);
  }

  onEmployeeUpdate() {
    this.employeeService.updateEmployee(this.employee).subscribe((data:Employee
    e)=> {
      console.log(`Employee updated: ${data}`);
    });
  }
}
```

Employee-Model.ts

```
class Employee {
  constructor(public id:number, public first_name:string, public
last_name:string, public email:string) {
  }
}
export default Employee;
```

Navbar.html

```
<nav class="navbar navbar-expand-lg">
  <div class="container d-flex justify-content-center">
    <a class="navbar-brand fw-bold">Portal</a>
    <button
      class="navbar-toggler"
      type="button"
      data-bs-toggle="collapse"
      data-bs-target="#navbarNav"
      aria-controls="navbarNav"
      aria-expanded="false"
      aria-label="Toggle navigation"
    >
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav d-flex align-items-center">
        <li class="nav-item" *ngIf="isUser">
          <a class="nav-link" routerLink="/home" routerLinkActive="active"
>Home</a>
        </li>
        <li class="nav-item" *ngIf="isUser">
          <a class="nav-link" routerLink="/employees"
routerLinkActive="active"
>Employees</a>
        </li>
        <li class="nav-item" *ngIf="isUser">
          <a
            class="nav-link"
            routerLink="/change-password"
            routerLinkActive="active"
            >Change Password</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

```

        <li class="nav-item" *ngIf="isUser">
            <a routerLink="#" class="nav-link" style="cursor: pointer"
(click)="logoutUser()">
                Logout
            </a>
        </li>
    </ul>
</div>

</div>
<hr>
</nav>

```

Navbar-component.ts

```

import { Component, OnInit } from '@angular/core';
import { EmployeeService } from '../employee.service';
import { Router } from '@angular/router';

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

  public isUser:boolean = false;

  constructor(private employeeService:EmployeeService, private router:Router)
  {

  }

  ngOnInit(): void {
    this.employeeService.getIsLoggedIn().subscribe((value:boolean) => {
      this.isUser = value;
    });
  }

  logoutUser() {
    localStorage.removeItem("admin");
    this.employeeService.setIsLoggedIn(false);
    this.router.navigate(['login']);
  }
}

```

app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { EmployeesComponent } from '../employees/employees.component';
import { LoginComponent } from '../login/login.component';
import { EmployeeComponent } from '../employee/employee.component';
import { HomeComponent } from '../home/home.component';

const routes: Routes = [
  {
    path: "",
    redirectTo: "/login",
    pathMatch: "full"
  },
  {
    path: "employees",
    component: EmployeesComponent,
  },
  {
    path: "employees/:id",
    component: EmployeeComponent,
  },
  {
    path: "login",
    component: LoginComponent
  },
  {
    path: "home",
    component: HomeComponent
  },
  {
    path: "**",
    redirectTo: "/login",
    pathMatch: "full"
  }
];

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


App-component.html

```
<app-navbar></app-navbar>
<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 = 'event-management';
}
```

App-module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { NgbModule } from '@ng-bootstrap/ng-bootstrap';
import { FormsModule } from "@angular/forms";

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { NavbarComponent } from './navbar/navbar.component';
import { EmployeesComponent } from './employees/employees.component';
import { LoginComponent } from './login/login.component';
import { EmployeeComponent } from './employee/employee.component';
import { HttpClientModule } from '@angular/common/http';
import { HomeComponent } from './home/home.component';

@NgModule({
  declarations: [
    AppComponent,
    NavbarComponent,
    EmployeesComponent,
    LoginComponent,
    EmployeeComponent,
    HomeComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    NgbModule,
    FormsModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

```

imports: [
    BrowserModule,
    AppRoutingModule,
    NgbModule,
    FormsModule,
    HttpClientModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }

```

Employee.service.ts

```

import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { BehaviorSubject, Observable, tap } from 'rxjs';
import Employee from '../models/Employee';

@Injectable({
    providedIn: 'root',
})
export class EmployeeService {
    public adminEmail: string = 'admin';
    public adminPassword: string = 'admin';
    public employees: Employee[] = [];
    public isLoggedIn: BehaviorSubject<boolean> = new
BehaviorSubject<any>(localStorage.getItem("admin") ? true : false);

    constructor(private http: HttpClient) {}

    public loadAllEmployees = (): Observable<Employee[]> => {
        return this.http
            .get<Employee[]>('http://localhost:3000/employees');
    };

    public addEmployee = (employee: Employee): Observable<Employee> => {
        return this.http
            .post<Employee>('http://localhost:3000/employees', employee);
    };
}

```

```

public getEmployee = (id:number):Observable<Employee>=> {
    return this.http
        .get<Employee>(`http://localhost:3000/employees/${id}`);
}

public deleteEmployee = (id:number):Observable<Employee>=> {
    return this.http
        .delete<Employee>(`http://localhost:3000/employees/${id}`);
}

public updateEmployee = (employee:Employee):Observable<Employee>=> {
    return this.http
        .put<Employee>(`http://localhost:3000/employees/${employee.id}`,
employee);
}

public setIsLoggedIn(value: boolean): void {
    this.isLoggedIn.next(value);
}

public getIsLoggedIn(): Observable<boolean> {
    return this.isLoggedIn.asObservable();
}
}

```

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"
  },
  {
    "id": 4,
    "first_name": "Shaik",
    "last_name": "Raheem",
    "email": "Shaik.Raheem@gmail.com"
  }
]
```

Index.html(SPA)

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>EventManager</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```