

Event Management App Code

Login Component

Ts

```
import { Component } from '@angular/core';
import { EventService } from '../eventService/event.service';
import { HttpClient } from '@angular/common/http';
import { Router } from '@angular/router';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css'],
})
export class LoginComponent {
  constructor(
    private es: EventService,
    private http: HttpClient,
    private router: Router
  ) {}

  username: string = '';
  password: string = '';

  url = 'http://localhost:3000/user';

  istrue = false;

  loginUser: any;
  istruemsg: boolean = true;

  login() {
    this.http.get(this.url).subscribe((res) => {
      this.loginUser = res;
      for (let i of this.loginUser) {
        if (i.username === this.username && i.password === this.password) {
          this.istrue = true;
          this.es.isAuth = true;
          this.istruemsg = true;
        }
      }
    });
  }
}
```

```

        this.router.navigate(['/home']);
    } else {
        this.istruemsg = false;
        this.es.isAuth = false;
        this.istrue = false;
    }
}
});
}

ischange = false;
changePassword() {
    this.ischange = true;
}

model() {
    this.ischange = false;
}

changeusername: string = '';
changepassword: string = '';

isusercorrect: boolean = false;

userurl = 'http://localhost:3000/user/1';

ischangeRouter() {
    if (this.isusercorrect) {
        this.router.navigate(['/home']);
    }
}

passwordChange: boolean = false;

userCheck() {
    this.http.get(this.url).subscribe((res) => {
        this.loginUser = res;
        for (let i of this.loginUser) {
            if (i.username === this.changeusername) {
                this.isusercorrect = true;
            }
        }
    });
}

submit() {
    if (this.isusercorrect) {
        let details = {

```

```

        username: this.changeusername,
        password: this.changepassword,
    };
    this.http.put(this.userurl, details).subscribe((res) => {
        this.passwordChange = true;
        console.log(res);
        this.model();
    });
}
}
}

```

Html

```

<div class="formCon card">
  <div class="card-body">
    <form ngForm class="login form">
      <div class="m-2">
        <label class="m-2" for="username">UserName</label>
        <input
          class="form-control"
          type="text"
          placeholder="UserName"
          ngModel
          #user="ngModel"
          name="username"
          [(ngModel)]="username"
          minlength="4"
          required
        />
        <div *ngIf="user.touched && user.errors">
          <p class="msg" *ngIf="user.errors['required']">
            UserName is Required
          </p>
          <p class="msg" *ngIf="user.errors['minlength']">Min Length 4</p>
        </div>
      </div>
      <div class="m-2">
        <label class="m-2" for="password">Password</label>
        <input
          class="form-control"
          type="password"
          placeholder="Password"

```

```

        ngModel
        #pwd="ngModel"
        name="password"
        [(ngModel)]="password"
        required
        minlength="4"
    />
    <div *ngIf="pwd.touched && pwd.errors">
        <p class="msg" *ngIf="pwd.errors['required']">Password is
Required</p>
        <p class="msg" *ngIf="pwd.errors['minlength']">Min Length 4</p>
    </div>
</div>
<p class="msg" *ngIf="!istruemsg">Enter Correct Details</p>
<button type="button" (click)="login()" class="btn btn-primary m-2">
    Login
</button>
<button type="button" class="btn btn-primary"
(click)="changePassword()">
    Change Password
</button>
<br />
</form>
</div>
</div>

<div id="myModal" class="modal" *ngIf="ischange">
    <div class="modal-content">
        <span (click)="model()" class="close">&times;</span>
        <div>
            <p class="text-center">Change Password</p>
            <form #pwd="ngForm" ngForm class="login form">
                <div class="m-2">
                    <label class="m-2" for="username">UserName</label>
                    <input
                        class="form-control"
                        type="text"
                        placeholder="UserName"
                        ngModel
                        #user="ngModel"
                        name="username"
                        [(ngModel)]="changeusername"
                        required
                    />
                    <div *ngIf="user.touched && user.errors">
                        <p class="msg" *ngIf="user.errors['required']">
                            UserName is Required
                        </p>

```

```

        <p class="msg" *ngIf="user.errors['minlength']">Min Length 4</p>
    </div>
</div>

<div class="m-2" *ngIf="isusercorrect">
    <label class="m-2" for="password">Password</label>
    <input
        class="form-control"
        type="password"
        placeholder="Password"
        ngModel
        #pwd="ngModel"
        name="password"
        [(ngModel)]="changepassword"
        required
        minlength="4"
    />
    <div *ngIf="pwd.touched && pwd.errors">
        <p class="msg" *ngIf="pwd.errors['required']">
            Password is Required
        </p>
        <p class="msg" *ngIf="pwd.errors['minlength']">Min Length 4</p>
    </div>
    <button
        type="button"
        class="btn btn-primary m-2"
        (click)="submit()"
        [disabled]="!pwd.valid"
    >
        change Password
    </button>
</div>
<button
    type="button"
    class="btn btn-primary"
    (click)="userCheck()"
    *ngIf="!isusercorrect"
>
    check UserName
</button>

<br />
</form>
</div>
</div>
</div>

```

Css

```
.login {
  padding: 10px;
}

.formCon {
  width: 100vh;
  margin-top: 10vh;
  margin-left: 5vh;
  border-radius: 25px;
}

@media screen and (min-width: 990px) {
  .formCon {
    width: 100vh;
    margin-top: 10vh;
    margin-left: 60vh;
    border-radius: 25px;
  }
}

input {
  border-radius: 20px;
  font-family: roboto;
}

label {
  font-weight: 600;
  font-family: roboto;
}

.modal {
  position: fixed;
  z-index: 1;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  overflow: auto;
  display: flex;
  align-items: center;
  background-color: rgba(0, 0, 0, 0.4);
}

.modal-content {
  background-color: #fefefe;
  margin: 15% auto;
  padding: 20px;
  border: 1px solid #888;
```

```
    width: 80%;
}

.close {
  color: #aaa;
  float: right;
  font-size: 28px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: black;
  text-decoration: none;
  cursor: pointer;
}

.msg {
  font-size: 10px;
  font-weight: bold;
}
```

Home Component

Ts

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

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

}
```

Html

```
<div class="container-fuild">
  <div class="d-flex flex-row">
    <div class="textCon col-6 text-center align-self-center">
      <h1>Event Management</h1>
    </div>
    <div class="imgCon col-6">
      
    </div>
  </div>
</div>
```

Css

```
h1 {
  font-style: normal;
  font-family: serif;
}
```

Employee-event Details Component

Ts

```
import { Component, OnInit, ViewChild } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { tick } from '@angular/core/testing';
import { EventService } from '../eventService/event.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-employee-event',
```



```

    templateUrl: './employee-event.component.html',
    styleUrls: ['./employee-event.component.css'],
  })
export class EmployeeEventComponent implements OnInit {
  constructor(
    private http: HttpClient,
    private es: EventService,
    private router: Router
  ) {}

  url = ' http://localhost:3000/employees';

  userdata: any;

  header: string = '';

  id: number = 0;
  fname: string = '';
  lname: string = '';
  email: string = '';

  istrue: boolean = false;

  gettingData() {
    this.es.fetchData().subscribe(
      (data) => {
        this.userdata = data;
      },
      (error) => {
        console.log(error);
      }
    );
  }

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

  view(id: any) {
    this.router.navigate(['/view', id]);
  }

  close() {
    this.istrue = false;
  }

  add() {
    this.istrue = true;
  }
}

```

```

submit() {
  let details = {
    first_name: this.fname,
    last_name: this.lname,
    email: this.email,
  };
  return this.es.postData(details).subscribe((res) => {
    this.gettingData();
    this.istrue = false;
  });
}
}

```

Html

```

<div class="jumbotron">
  <h1 class="display-4">Event Management</h1>
  <button class="btn btn-primary" (click)="add()">Add</button>
  <hr />
</div>

<div id="" class="modal" *ngIf="istrue">
  <div class="modal-content">
    <span class="" (click)="close()">&times;</span>
    <div class="form-container">
      <h1 class="heading1">Add</h1>
      <form #login="ngForm" ngForm (submit)="submit()">
        <div class="form-group">
          <label for="firstname">First Name</label>
          <input
            class="form-control"
            type="text"
            name="firstname"
            [(ngModel)]="fname"
            placeholder="First Name"
            required
            minlength="4"
            ngModel
            #username="ngModel"
          />
        </div>
      </form>
    </div>
  </div>

```



```

    </div>
  </div>
</div>

<div class="container mt-4" *ngIf="userdata; else err">
  <div class="row tableContainer">
    <table class="table">
      <thead>
        <tr>
          <th scope="col">Id</th>
          <th scope="col">First Name</th>
          <th scope="col">Last Name</th>
          <th scope="col">Email</th>
        </tr>
      </thead>
      <tbody>
        <tr *ngFor="let i of userdata">
          <td>{{ i.id }}</td>
          <td>{{ i.first_name }}</td>
          <td>{{ i.last_name }}</td>
          <td>{{ i.email }}</td>
          <td>
            <button class="btn btn-primary" (click)="view(i.id)">
              View Details
            </button>
          </td>
        </tr>
      </tbody>
    </table>
  </div>
</div>

<ng-template #err class="text-info"> Error Fetching Data </ng-template>

```

Css

```

.employess {
  width: 800px;
  padding: 20px 25px;
  background-color: white;
  margin: 30px 0;
  border-radius: 3px;
}

```

```
.headers {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.ViewContainer {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100%;
  width: 100%;
  position: absolute;
  top: 0;
  left: 0;
  background-color: rgba(0, 0, 0, 0);
  z-index: 100;
}

.viewCon {
  box-shadow: rgba(6, 24, 44, 0.4) 0px 0px 0px 2px,
    rgba(6, 24, 44, 0.65) 0px 4px 6px -1px,
    rgba(255, 255, 255, 0.08) 0px 1px 0px inset;
  background-color: skyblue;
  width: 600%;
  padding: 20px 50px;
  border-radius: 5px;
}

.modal {
  position: fixed;
  z-index: 1;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0, 0, 0, 0.4);
  display: flex;
  align-items: center;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 15% auto;
  padding: 20px;
  border: 1px solid #888;
  width: 80%;
}
```

```

}

.close {
  color: #aaa;
  float: right;
  font-size: 28px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: black;
  text-decoration: none;
  cursor: pointer;
}

.heading1 {
  font-size: 20px;
  text-align: center;
}

```

ViewEmployee Component

Ts

```

import { Component, OnInit } from '@angular/core';
import { EventService } from '../eventService/event.service';
import { ActivatedRoute, Router } from '@angular/router';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-view-employee',
  templateUrl: './view-employee.component.html',
  styleUrls: ['./view-employee.component.css'],
})
export class ViewEmployeeComponent implements OnInit {
  constructor(
    public route: ActivatedRoute,

```

```

    public router: Router,
    private es: EventService,
    private http: HttpClient
  ) {}

  fname: string = '';
  lname: string = '';
  email: string = '';

  paramValue: any;
  viewData: any;

  istrue: boolean = false;

  gettingDetails() {
    return this.es.fetchDataById(this.paramValue).subscribe((res) => {
      this.viewData = res;
      console.log(this.viewData);
    });
  }

  ngOnInit(): void {
    let value = this.route.params.subscribe((params) => {
      this.paramValue = params['id'];
      console.log(this.paramValue);
      this.gettingDetails();
      console.log(this.viewData);
    });
  }

  delete() {
    this.es.deleteUser(this.paramValue).subscribe((res) => {
      this.router.navigate(['/event']);
    });
  }

  edit() {
    this.istrue = true;
    this.fname = this.viewData.first_name;
    this.lname = this.viewData.last_name;
    this.email = this.viewData.email;
  }

  close() {
    this.istrue = false;
  }

  save() {

```

```

let update = {
  first_name: this.fname,
  last_name: this.lname,
  email: this.email,
};
return this.es.putData(update, this.paramValue).subscribe((res) => {
  this.router.navigate(['/event']);
});
}
}

```

Html

```

<div class="container-fluid">
  <div class="card viewContainer">
    <div class="card-body">
      <p>
        Details of
        <span>{{ viewData.first_name }}{{ viewData.last_name }}</span>
      </p>
      <div class="d-flex flex-row justify-content-between">
        <p class="heading">Employee Id</p>
        <p>{{ viewData.id }}</p>
      </div>
      <div class="d-flex flex-row justify-content-between">
        <p class="heading">Employee First Name</p>
        <p>{{ viewData.first_name }}</p>
      </div>
      <div class="d-flex flex-row justify-content-between">
        <p class="heading">Employee Last Name</p>
        <p>{{ viewData.last_name }}</p>
      </div>
      <div class="d-flex flex-row justify-content-between">
        <p class="heading">Employee Email</p>
        <p>{{ viewData.email }}</p>
      </div>
      <div>
        <button type="button" routerLink="/event" class="m-2 btn btn-primary">
          Employee List
        </button>
        <button type="button" class="m-2 btn btn-secondary"
(click)="delete()">

```



```

        Delete
    </button>
    <button type="button" class="m-2 btn btn-info" (click)="edit()">
        Update
    </button>
</div>
</div>
</div>
</div>
</div>

<div id="" class="modal" *ngIf="istruer">
    <div class="modal-content">
        <span class="" (click)="close()">&times;</span>
        <div class="form-container">
            <h1>Edit details of {{ viewData.first_name }}</h1>
            <form ngForm (submit)="save()">
                <div class="form-group">
                    <label for="firstname">First Name</label>
                    <input
                        class="form-control"
                        type="text"
                        name="firstname"
                        [(ngModel)]="fname"
                        placeholder="First Name"
                    />
                </div>
                <div class="form-group">
                    <label for="lastname">Last Name</label>
                    <input
                        class="form-control"
                        type="text"
                        name="lastname"
                        [(ngModel)]="lname"
                        placeholder="Last Name"
                    />
                </div>
                <div class="form-group">
                    <label for="email">Email</label>
                    <input
                        class="form-control"
                        type="email"
                        name="email"
                        [(ngModel)]="email"
                        placeholder="Email"
                    />
                </div>
                <button type="submit" class="btn btn-primary mt-4">Save
Changes</button>
            </form>
        </div>
    </div>

```

```
        </form>
      </div>
    </div>
  </div>
```

Css

```
span {
  font-weight: bold;
  font-style: italic;
}

.viewContainer {
  width: 100vh;
  margin-left: 20px;
  text-align: left;
}

.heading {
  font-weight: bold;
}

@media screen and (min-width: 990px) {
  .viewContainer {
    width: 120vh;
    margin-left: 40vh;
    text-align: left;
  }
}

.modal {
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  background-color: rgba(0, 0, 0, 0.4); /* Black w/ opacity */
  display: flex;
  align-items: center;
}
```

```

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 15% auto; /* 15% from the top and centered */
  padding: 20px;
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}

/* The Close Button */
.close {
  color: #aaa;
  float: right;
  font-size: 28px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: black;
  text-decoration: none;
  cursor: pointer;
}

h1 {
  font-size: 20px;
  text-align: center;
}

```

Header Component

Ts

```

import { Component, OnInit } from '@angular/core';
import { EventService } from '../eventService/event.service';

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

```

```

    styleUrls: ['./header.component.css'],
  })
  export class HeaderComponent {
    constructor(private es: EventService) {}

    isAuth: boolean = this.es.isAuth;
  }

```

Html

```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Event Management</a>
  <div class="c" id="navbarNav">
    <ul class="navbar-nav d-flex flex-row">
      <li class="nav-item m-4" routerLinkActive="active">
        <a class="nav-link" routerLink="/home">Home </a>
      </li>
      <li class="nav-item m-4" routerLinkActive="active">
        <a class="nav-link" routerLink="/event">Employees</a>
      </li>
      <li class="nav-item m-2" routerLinkActive="active">
        <button type="button" routerLink="/admin" class="btn">
          
        </button>
      </li>
    </ul>
  </div>
</nav>

```

Css

```
li a {  
  font-size: 15px;  
  font-weight: bold;  
}  
li a:hover {  
  font-variant: small-caps;  
}  
  
.img {  
  height: 40px;  
}
```

Admin Component

Ts

```
import { Component, OnInit } from '@angular/core';  
import { EventService } from '../eventService/event.service';  
import { Router } from '@angular/router';  
  
@Component({  
  selector: 'app-admin',  
  templateUrl: './admin.component.html',  
  styleUrls: ['./admin.component.css'],  
})  
export class AdminComponent implements OnInit {  
  constructor(private es: EventService, private router: Router) {}  
  
  isAuthenticated: boolean = false;  
  
  login() {  
    if (this.isAuthenticated) {  
      this.es.isAuthenticated = false;  
      this.router.navigate(['/login']);  
    } else {  
      this.router.navigate(['/login']);  
    }  
  }  
  
  ngOnInit(): void {  
    this.isAuthenticated = this.es.isAuthenticated;  
  }  
}
```

```
}  
}
```

Html

```
<div class="container-fuild">  
  <div class="text-center">  
      
    <h1>Admin</h1>  
    <p  
      [innerHTML]="  
        isAuth  
          ? 'You are logged in to your account.'  
          : 'Your account is logged out.'  
      "  
    ></p>  
  </div>  
  <button  
    type="button"  
    class="btn btn-primary m-2"  
    [innerHTML]="isAuth ? 'Logout' : 'Login'"  
    (click)="login()"  
  ></button>  
  
  <button class="btn btn-primary m-2" routerLink="/login">  
    Change Password  
  </button>  
</div>
```

Css

```
.img {  
  height: 200px;  
  border-radius: 40px;  
}
```

App Component

Ts

```
import { HttpClient } from '@angular/common/http';  
import { RouterModule } from '@angular/router';  
import { BrowserModule } from '@angular/platform-browser';  
import { NgModule } from '@angular/core';  
import { FormsModule } from '@angular/forms';  
import { AppRoutingModule } from './components/app-route/app-route.module';  
import { HttpClientModule } from '@angular/common/http';  
  
import { AppComponent } from './app.component';  
import { HomeComponent } from './components/home/home.component';  
import { HeaderComponent } from './components/header/header.component';  
  
import { EventService } from './components/eventService/event.service';  
import { EmployeeEventComponent } from './components/employee-event/employee-event.component';  
import { ViewEmployeeComponent } from './components/view-employee/view-employee.component';  
import { LoginComponent } from './components/login/login.component';  
import { AdminComponent } from './components/admin/admin.component';  
  
@NgModule({  
  declarations: [  
    AppComponent,  
    HomeComponent,  
    HeaderComponent,  
    EmployeeEventComponent,  
    ViewEmployeeComponent,  
    LoginComponent,  
    AdminComponent,  
  ],  
  imports: [  
    BrowserModule,  
    AppRoutingModule,  
    HttpClientModule,  
    RouterModule,  
    FormsModule,  
  ],  
  providers: [  
    EventService,  
  ],  
  bootstrap: [AppComponent]  
})  
export class AppModule {}
```

```

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

```

Html

```
<app-header></app-header> <router-outlet></router-outlet>
```

Service

```

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

@Injectable({
  providedIn: 'root',
})
export class EventService {
  constructor(private http: HttpClient) {}

  url = ' http://localhost:3000/employees';

  isAuth: boolean = false;

  fetchData() {
    return this.http.get(this.url);
  }

  deleteUser(id: number) {
    return this.http.delete(this.url + '/' + id);
  }

  postData(body: any) {
    return this.http.post(this.url, body);
  }

  putData(body: any, id: number) {
    const url = `${this.url}/${id}`;
    return this.http.put(url, body);
  }
}

```



```

    }

    fetchDataById(id: any) {
        return this.http.get(this.url + '/' + id);
    }
}

```

Route

```

import { NgModule, Component } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterModule, Routes } from '@angular/router';
import { HomeComponent } from '../home/home.component';
import { ViewEmployeeComponent } from '../view-employee/view-employee.component';
import { EmployeeEventComponent } from '../employee-event/employee-event.component';
import { LoginComponent } from '../login/login.component';
import { AdminComponent } from '../admin/admin.component';

const routes: Routes = [
    { path: 'home', component: HomeComponent },
    { path: 'event', component: EmployeeEventComponent },
    { path: '', component: LoginComponent },
    { path: 'view/:id', component: ViewEmployeeComponent },
    { path: 'login', component: LoginComponent },
    { path: 'admin', component: AdminComponent },
    { path: '**', component: HomeComponent },
];

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

```

Json

```
{
  "user": [
    {
      "username": "admin",
      "password": "admin",
      "id": 1
    }
  ],
  "employees": [
    {
      "first_name": "Sachin",
      "last_name": "tendulkar",
      "email": "SachinTendulkar@gmail.com",
      "id": 1
    },
    {
      "first_name": "Dhoni",
      "last_name": "Ms",
      "email": "Dhoni@gmail.com",
      "id": 2
    }
  ]
}
```