

EVENT MANAGEMENT APP

SOURCE CODE

employee.component.html

```
<html>

<head>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet"
  integrity="sha384-
9ndCyUaIbzAi2FUVXJi0CjmCapSm07SnpJef0486qhLnuZ2cdeRh002iuK6FUUVM"
crossorigin="anonymous">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min
.js"
  integrity="sha384-
geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz"
crossorigin="anonymous"></script>
</head>

<body>
  <app-navigation></app-navigation>

  <div class="container">
    <button (click)="openPopup()" class="btn btn-success float-end">ADD NEW
EMPLOYEE</button>

    <div class="popup-overlay" *ngIf="isPopupOpen">
      <div class="popup-content col-md-6">
        <form [formGroup]="empRef" (ngSubmit)="storeEmp()">
          <h4 class="fw-normal mb-3 pb-4 text-center" style="letter-spacing:
1px; color: green;"><b>REGISTER
EMPLOYEE</b></h4>

          <div class="form-group mb-3">
            <label for="first_name">First Name</label>
            <input type="text" class="form-control" id="first_name"
formControlName="first_name"
placeholder="Enter First Name">
          </div>
        </form>
      </div>
    </div>
  </div>
</body>
</html>
```

```

        <div *ngIf="empRef.controls['first_name'].invalid &&
empRef.controls['first_name'].touched"
            class="text-danger">
                First Name is required.
            </div>
        </div>

        <div class="form-group mb-3">
            <label for="last_name">Last Name</label>
            <input type="text" class="form-control" id="last_name"
formControlName="last_name"
                placeholder="Enter Last Name">
            <div *ngIf="empRef.controls['last_name'].invalid &&
empRef.controls['last_name'].touched"
                class="text-danger">
                    Last Name is required.
                </div>
            </div>

            <div class="form-group">
                <label for="email">Email</label>
                <input type="email" class="form-control" id="email"
formControlName="email" placeholder="Enter Email">
                <div *ngIf="empRef.controls['email'].invalid &&
empRef.controls['email'].touched" class="text-danger">
                    Valid Email is required.
                </div>
            </div>

            <br>
            <input type="submit" value="ADD EMPLOYEE" class="btn btn-success
float-end">
            <input type="button" (click)="closePopup()" value="CLOSE" class="btn
btn-danger close-button">
        </form>
    </div>
</div>

<h3 style="font-family: 'Arial', sans-serif; text-align: center;">EMPLOYEE
DETAILS</h3>
<br>

<table class="table table-bordered table-striped">
    <thead class="thead-dark">
        <tr class="text-center py-3 small">
            <th scope="col" class="header-cell">EMPLOYEE ID</th>
            <th scope="col" class="header-cell">FIRST NAME</th>
            <th scope="col" class="header-cell">LAST NAME</th>

```

```

        <th scope="col" class="header-cell">EMAIL ADDRESS</th>
        <th scope="col" class="header-cell">MORE</th>
    </tr>
</thead>
<tbody>
    <tr *ngFor="let employee of employees" class="text-center">
        <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 custom-button"
(click)="viewDetails(employee)">VIEW MORE</button></td>
    </tr>
</tbody>
</table>
</div>
</body>
</html>

```

employee.component.ts

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { Employee } from '../employee';
import { EmployeeService } from '../employee.service';
import { FormGroup, FormControl, Validators } from "@angular/forms";
import { MatDialog, MatDialogConfig } from '@angular/material/dialog';

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

  empRef: FormGroup;
  employees: Array<Employee> = [];
  isPopupOpen = false;

  constructor(public es: EmployeeService, public router: Router, private
dialog: MatDialog) {
    this.empRef = new FormGroup({
      first_name: new FormControl('', Validators.required),
      last_name: new FormControl('', Validators.required),
      email: new FormControl('', [Validators.required, Validators.email])
    });
  }
}

```

```

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

loadEmpDetails() {
    this.es.loadEmpData().subscribe({
        next: (data: any) => this.employees = data,
        error: (error: any) => console.log(error),
        complete: () => console.log("Completed")
    });
}

storeEmp() {
    if (this.empRef.valid) {
        let employee = this.empRef.value;
        this.es.storeEmpData(employee).subscribe({
            next: (data: any) => console.log(data),
            error: (error: any) => console.log(error),
            complete: () => {
                this.loadEmpDetails();
                this.empRef.reset();
                alert("EMPLOYEE ADDED SUCCESSFULLY");
                this.isPopupOpen = false;
            }
        });
    } else {
        this.empRef.markAllAsTouched();
    }
}

viewDetails(employee: any) {
    sessionStorage.setItem("empInfo", JSON.stringify(employee));
    this.router.navigate(["view-employee"]);
}

openPopup() {
    this.isPopupOpen = true;
}

closePopup() {
    this.isPopupOpen = false;
}
}

```

employee.component.css

```
.popup-overlay {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0, 0, 0, 0.5); /* semi-transparent background */
  display: flex;
  align-items: center;
  justify-content: center;
}

.popup-content {
  background-color: white;
  padding: 20px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2); /* Add shadow effect */
}

.custom-button {
  font-size: smaller;
}

.header-cell {
  padding: 15px;
  margin-bottom: 10px;
}

.close-button {
  float: left;
}
```

home.component.html

```
<app-navigation></app-navigation>
<div class="container">
  <!-- your-component.component.html -->
  <h4 class="fw-normal text-center" style="letter-spacing: 1px; color:brown;
font-size: 30px;"><b>WELCOME ADMIN..!</b>
</h4>
  <button (click)="openPopup()" style="margin: 0 auto;" class="btn btn-danger
float-end">CHANGE PASSWORD</button>
  <div class="popup-overlay" *ngIf="isPopupOpen">
    <div class="popup-content col-md-4">
      <form [formGroup]="passwordForm" (ngSubmit)="updatePassword()"
class="smaller-label">
        <h4 class="fw-normal mb-3 pb-4 text-center" style="letter-spacing:
1px; color: olive;"><b>UPDATE PASSWORD</b>
        </h4>
```

```

<div class="form-group mb-3">
  <label for="old_pswd">Old password :</label>
  <input type="password" class="form-control" id="old_pswd"
formControlName="old_pswd"
  placeholder="Enter your old password" required
  [ngClass]="{ 'is-invalid': submitted && f['old_pswd'].errors }" />
  <div *ngIf="submitted && f['old_pswd'].errors" class="invalid-
feedback">
    <div *ngIf="f['old_pswd'].errors['required']">Field Missing</div>
    <div *ngIf="f['old_pswd'].errors['minlength']">Password must be at
least 2 characters</div>
  </div>
</div>

<div class="form-group mb-3">
  <label for="new_pswd">New password :</label>
  <input type="password" class="form-control" id="new_pswd"
formControlName="new_pswd"
  placeholder="Enter you New password" required
  [ngClass]="{ 'is-invalid': submitted && f['new_pswd'].errors }" />
  <div *ngIf="submitted && f['new_pswd'].errors" class="invalid-
feedback">
    <div *ngIf="f['new_pswd'].errors['required']">Field Missing</div>
    <div *ngIf="f['new_pswd'].errors['minlength']">Password must be at
least 6 characters</div>
  </div>
</div>
<div class="form-group mb-3">
  <label for="confirm_pswd">Confirm password :</label>
  <input type="password" class="form-control" id="confirm_pswd"
formControlName="confirm_pswd"
  placeholder="Re-enter new password" required
  [ngClass]="{ 'is-invalid': submitted && f['confirm_pswd'].errors
}" />
  <div *ngIf="submitted && f['confirm_pswd'].errors" class="invalid-
feedback">
    <div *ngIf="f['confirm_pswd'].errors['required']">Field
Missing</div>
    <div *ngIf="f['confirm_pswd'].errors['passwordMismatch']"
class="text-danger">Passwords do not match</div>
  </div>
</div>
<br>
<input type="submit" value="UPDATE PASSWORD" class="btn btn-success">
<input type="button" (click)="closePopup()" value="CLOSE" class="btn
btn-danger close-button">
</form>
<br>

```

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

home.component.ts

```
import { Component, OnInit } from '@angular/core';
import { EmployeeService } from '../employee.service';
import { Employee } from '../employee';
import { FormGroup, FormControl, Validators, ValidatorFn, AbstractControl }
from '@angular/forms';

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

export class HomeComponent implements OnInit {
  employees: Array<Employee> = [];
  admin: any = {};
  passwordForm!: FormGroup;
  isPopupOpen = false;
  submitted: boolean = false;
  msg: string = "";

  get f() {
    return this.passwordForm.controls;
  }

  constructor(private es: EmployeeService) {
    this.passwordForm = new FormGroup({
      email: new FormControl('', [Validators.required]), // Add email
      validation
      password: new FormControl('', [Validators.required]), // Add password
      validation
      old_pswd: new FormControl('', [Validators.required]), // Add old
      password validation
      new_pswd: new FormControl('', [Validators.required,
        Validators.minLength(6)]), // Add new password validation
      confirm_pswd: new FormControl('', [Validators.required]) // Add confirm
      password validation
    });
  }

  ngOnInit(): void {
    this.loadEmployees();
    this.loadAdmin().then(() => {
      this.initPasswordForm();
    });
  }
}
```

```

    });
}

loadEmployees(): void {
    this.es.loadEmpData().subscribe(
        (data: any) => {
            this.employees = data;
        },
        (error: any) => {
            console.log(error);
        }
    );
}

loadAdmin(): Promise<void> {
    return new Promise<void>((resolve, reject) => {
        this.es.getAdmin().subscribe(
            (data: any) => {
                if (data.email && typeof data.email === 'string') {
                    this.admin.email = data.email;
                } else {
                    // Handle the case when the email value is missing or invalid
                    console.log('Invalid email value for admin:', data);
                }
                resolve();
            },
            (error: any) => {
                console.log(error);
                reject();
            }
        );
    });
}

initPasswordForm(): void {
    this.passwordForm = new FormGroup({
        email: new FormControl(this.admin.email), // Set the email value from
the admin object
        old_pswd: new FormControl('', Validators.required),
        new_pswd: new FormControl('', [Validators.required,
Validators.minLength(6)]),
        confirm_pswd: new FormControl('', Validators.required)
    });
}

updatePassword(): void {
    this.submitted = true; // Set the submitted flag to true

```



```

if (this.passwordForm.invalid) {
  return; // Exit the function if the form is invalid
}

const newPassword = this.passwordForm.value.new_pswd;
const confirmPswd = this.passwordForm.value.confirm_pswd;

if (newPassword === confirmPswd) {

  const updatedAdmin = { ...this.admin, password: newPassword };

  this.es.updateAdmin(this.admin.email, updatedAdmin).subscribe(
    (data: any) => {
      this.admin.password = newPassword;
      // console.log('Password updated successfully:', newPassword);
      this.passwordForm.reset();
      this.isPopupOpen = false;
      alert('Password updated successfully!'); // Display success alert
    },
    (error: any) => {
      console.log('Error updating admin:', error);
      // Handle any error actions here
    }
  );
} else {
  //console.log("paswd not match");
  alert("Password do not match")
}
}

changePassword(): void {
  this.isPopupOpen = true;
}

openPopup(): void {
  this.isPopupOpen = true;
}

closePopup(): void {
  this.isPopupOpen = false;
}
}

```

home.component.css

```

.popup-overlay {
  position: fixed;

```

```

    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.5);
    /* semi-transparent background */
    display: flex;
    align-items: center;
    justify-content: center;
}

.popup-content {
    background-color: white;
    padding: 20px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);
    /* Add shadow effect */
}

.custom-button {
    font-size: smaller;
}

.header-cell {
    padding: 15px;
    margin-bottom: 10px;
}

.close-button {
    float: right;
}

.smaller-label {
    font-size: 14px;
    color: gray;
}

```

login.component.html

```

<!DOCTYPE html>
<html>
<head>
    <title>Login Container with Shadow</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
">
</head>
<body>
    <br><br><br>

```

```
<nav class="navbar navbar-expand-sm bg-dark text-white navbar-dark fixed-top
navbar-lg">
    <div class="container-fluid">
        <h2>EVENT MANAGEMENT APP</h2>
    </div>
</nav>
<br><br>
<div class="container col-md-5">
    <div class="login-container col-md-6 ">
        <div class="row">
            <div class="col-md-12 center-content">
                <form [formGroup]="loginRef" (ngSubmit)="checkUser()">
                    <h4 class="fw-normal mb-3 pb-4 text-center" style="letter-spacing:
1px; color: green;"><b>SIGN IN</b></h4>
                    <p style="color: grey;">Welcome to our website. Please log in to
access your account.</p>

                    <div class="form-group">
                        <label for="email">Email Address</label>
                        <input type="email" class="form-control" id="email"
formControlName="email" placeholder="Enter your email"
[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 for="password">Password</label>
                        <input type="password" class="form-control" id="password"
formControlName="password"
placeholder="Enter your password" [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>
                            <div *ngIf="f['password'].errors['minlength']">Password must
be at least 2 characters</div>
                        </div>
                    </div>
                    <button class="btn btn-success" type="submit">LOGIN</button>
                    &nbsp;&nbsp; 
                    <a class="small text-muted" href="#">Forgot password?</a>
```

```

        <p class="mb-2 pb-lg-2" style="color: #393f81;">Don't have an
account? <a href="/register"
        style="color: #393f81;">Register here</a></p>
        <a href="#" class="small text-muted">Terms of use.</a>
        <a href="#" class="small text-muted">Privacy policy</a>
    </form>
</div>
</div>
</div>
</div>
</body>
</html>

```

login.component.ts

```

// login.component.ts
import { Component, OnInit } from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';
import { EmployeeService } from '../employee.service';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
  msg: string = "";
  loginRef: FormGroup; // Declare the form group
  submitted: boolean = false;

  constructor(public router: Router, public es: EmployeeService) {
    this.loginRef = new FormGroup({
      email: new FormControl('', [Validators.required]), // Add email
validation
      password: new FormControl('', [Validators.required]) // Add password
validation
    });
  }

  ngOnInit(): void {
  }

  get f() { return this.loginRef.controls; }

  checkUser() {
    this.submitted = true; // Set the submitted flag to true
  }
}

```

```

    if (this.loginRef.valid) {
      const login = this.loginRef.value;
      this.es.getAdmin().subscribe((admin: any) => {
        if (admin.email === login.email && admin.password === login.password)
        {
          this.router.navigate(['home']);
        } else {
          this.msg = "Invalid Email or Password..Try Again";
          alert(this.msg);
        }
      });
    } else {
    }
  }
}

```

login.component.css

```

.login-container {
  max-width: 600px;
  margin: 1;
  padding: 20px;
  border-radius: 5px;
  box-shadow: 0px 0px 10px rgba(61, 53, 53, 0.1);
  background-color: whitesmoke;
}

.center-content {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100%;
}

```

navigation.component.html

```

<div class="container">
  <nav class="navbar navbar-expand-sm bg-dark text-light navbar-dark fixed-top navbar-lg">
    <div class="container-fluid">
      <h2>EVENT MANAGEMENT APP</h2>
      <ul class="navbar-nav" routerLinkActive="active">
        <li class="nav-item">
          <a class="nav-link" routerLink="/home">HOME</a>
        </li>

```

```

<li class="nav-item">
  <a class="nav-link" href="/employees">EMPLOYEES</a>
</li>
<li class="nav-item">
  <a class="nav-link" routerLink="">LOGOUT</a>
</li>
<li class="nav-item">
  <div class="search-container">
    <input class="form-control" type="search" placeholder="Search "
[(ngModel)]="searchTerm"
(ngModelChange)="performSearch()">
  </div>
</li>
</ul>
</div>
</nav>
</div>

<div class="container mt-4">
  <table class="table table-bordered table-striped"
*ngIf="searchResults.length > 0">
    <thead>
      <br><br><br>
      <tr>
        <th scope="col">EMPLOYEE ID</th>
        <th scope="col">FIRST NAME</th>
        <th scope="col">LAST NAME</th>
        <th scope="col">EMAIL ADDRESS</th>

      </tr>
    </thead>
    <tbody>
      <tr *ngFor="let employee of searchResults">
        <td>{{ employee.id }}</td>

        <td>{{ employee.first_name }}</td>
        <td>{{ employee.last_name }}</td>
        <td>{{ employee.email }}</td>

      </tr>
    </tbody>
  </table>
</div>
<br><br><br><br>

```

navigation.component.ts

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

```

import { Observable } from 'rxjs';
import { Employee } from '../employee';
import { EmployeeService } from '../employee.service';

@Component({
  selector: 'app-navigation',
  templateUrl: './navigation.component.html',
  styleUrls: ['./navigation.component.css']
})
export class NavigationComponent {
  searchTerm!: string;
  searchResults$: Observable<Employee[]> = new Observable<Employee[]>();
  searchResults: Employee[] = []; // New variable to hold the search results

  constructor(private employeeService: EmployeeService) { }

  performSearch() {
    if (this.searchTerm.trim() === '') {
      this.searchResults = []; // Empty the search results
    } else {
      this.searchResults$ =
this.employeeService.searchEmployees(this.searchTerm);
      this.searchResults$.subscribe(
        (data: Employee[]) => {
          this.searchResults = data; // Assign the search results to the
variable
        },
        (error: any) => {
          console.log(error);
        }
      );
    }
  }
}

```

navigation.component.css

```

.search-container {
  max-width: 300px;
  margin: auto;
  padding: 10px;
  color: rgb(17, 14, 14);
}

```

register.component.html

```

<div class="popup-form">
  <h2>Registration</h2>
  <form (ngSubmit)="registerEmployee()">
    <div>
      <label for="name">Name:</label>
      <input type="text" id="name" name="name" required>
    </div>
    <div>
      <label for="email">Email:</label>
      <input type="email" id="email" name="email" required>
    </div>
    <button type="submit">Register</button>
  </form>
</div>

```

register.component.ts

```

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

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

  }
}

```

register.component.css

```

.popup-form {
  background-color: white;
  padding: 20px;
  max-width: 400px;
  border-radius: 4px;
}

.popup-form h2 {
  text-align: center;
  margin-top: 0;
}

.popup-form form {
  display: flex;

```



```

    flex-direction: column;
}

.popup-form label {
    margin-bottom: 5px;
}

.popup-form input {
    padding: 8px;
    margin-bottom: 10px;
}

.popup-form button {
    background-color: #4caf50;
    color: white;
    border: none;
    padding: 10px 16px;
    cursor: pointer;
    border-radius: 4px;
}

.popup-form button[type="submit"] {
    background-color: #2196f3;
    margin-top: 10px;
}

.popup-form button[type="submit"]:hover {
    background-color: #0b7dda;
}

```

update-employee.component.html

```

<app-navigation></app-navigation>

<div class="container shadow col-md-7">
  <h4 class="font-weight-bold">{{employee.first_name}}</h4>
  <br>
  <form [formGroup]="empuRef" (ngSubmit)="updateEmpDetails()">
    <table class="table table-borderless">
      <tbody>
        <tr>
          <td><label for="id">Employee ID</label></td>
          <td><input type="number" [(ngModel)]="employee.id" class="form-
control" formControlName="id" readonly></td>
        </tr>
        <tr>
          <td><label for="first_name">First Name</label></td>

```

```

        <td><input type="text" id="first_name"
[(ngModel)]="employee.first_name" class="form-control"
        formControlName="first_name"></td>
    </tr>
    <tr>
        <td><label for="last_name">Last Name</label></td>
        <td><input type="text" id="last_name"
[(ngModel)]="employee.last_name" class="form-control"
        formControlName="last_name"></td>
    </tr>
    <tr>
        <td><label for="email">Email</label></td>
        <td><input type="text" id="email" [(ngModel)]="employee.email"
class="form-control" formControlName="email">
        </td>
    </tr>
</tbody>
</table>
<br>
<input type="submit" class="btn btn-success" value="UPDATE EMPLOYEE">
</form>
<br>
<input type="button" class="btn btn-link" value="Back To Home"
(click)="ViewAll()">&nbsp;
</div>

```

update-employee.component.ts

```

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

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

  constructor(public es: EmployeeService, public router: Router) { }
  employee: any;

  empuRef = new FormGroup({
    id: new FormControl(),
    first_name: new FormControl(),
    last_name: new FormControl(),
    email: new FormControl()
  })

```

```

    })

    ngOnInit(): void {
      let obj = sessionStorage.getItem("empUpd");
      if (obj != null) {
        this.employee = JSON.parse(obj);
      }
    }

    ViewAll() {
      this.router.navigate(["employees"])
    }

    updateEmpDetails() {
      let employee = this.empuRef.value;
      let id = this.empuRef.value.id;
      this.es.updateEmp(id, employee).subscribe({
        next: (data: any) => this.employee = data,
        error: (error: any) => console.log(error),
        complete: () => {
          this.ViewAll();
          alert("Employee details updated successfully!");
        }
      })
    }
  }
}

```

view-employee.component.html

```

<!DOCTYPE html>
<html>
<head>
  <title>Login Container with Shadow</title>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
">
  <!-- to get font icons FONTAWESOME -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css"
    integrity="sha384-
wvfxPqpZZVQGK6TAh5PVlGOfQNHSoD2xbE+QkPxCaf1NEevoEH3Sl0sibVcOQVnN"
    crossorigin="anonymous">
</head>
<body>
  <app-navigation></app-navigation>
  <div class="container col-md-7 shadow text-center"><br>

```

```

    <h4 class="font-weight-bold green-text">{{employee.first_name |
uppercase}}</h4>
    <br>
    <table class="table table-borderless ">
        <tbody>
            <tr>
                <td><strong>EMPLOYEE ID</strong></td>
                <td>{{employee.id}}</td>
            </tr>
            <tr>
                <td><strong>FIRST NAME</strong></td>
                <td>{{employee.first_name}}</td>
            </tr>
            <tr>
                <td><strong>LAST NAME</strong></td>
                <td>{{employee.last_name}}</td>
            </tr>
            <tr>
                <td><strong>EMAIL ID</strong></td>
                <td>{{employee.email}}</td>
            </tr>
        </tbody>
    </table>
    <hr>
    <input type="button" class="btn btn-info" value="VIEW ALL"
(click)="ViewAll()" />&nbsp;
    <input type="button" class="btn btn-success" value="UPDATE"
(click)="updateEmployee(employee)" />&nbsp;
    <input type="button" class="btn btn-danger" value="REMOVE "
(click)="deleteEmployee(employee.id)" />&nbsp;<br>
    <a class="btn btn-link" (click)="ViewAll()"><i class="fa fa-arrow-left
">&nbsp;&nbsp;&nbsp;&Go back</i></a>&nbsp;
    </div>
</body>
</html>

```

view-employee.component.ts

```

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

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

```

```

employee: any;
constructor(public router: Router, public es: EmployeeService) { }

ngOnInit(): void {
  let obj = sessionStorage.getItem("empInfo");
  if (obj != null) {
    this.employee = JSON.parse(obj);
  }
}

ViewAll() {
  this.router.navigate(["employees"])
}

deleteEmployee(id: any) {
  if (confirm("Are you sure you want to delete this employee?")) {
    this.es.delete(id).subscribe({
      next: (result: any) => console.log(result),
      error: (error: any) => console.log(error),
      complete: () => this.ViewAll()
    });
  }
}

updateEmployee(employee: any) {
  sessionStorage.setItem("empUpd", JSON.stringify(employee));
  this.router.navigate(["update-employee"]);
}
}

```

view-employee.component.css

```

.green-text {
  color: green;
}

```

app-routing.module.ts

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { EmployeeComponent } from '../employee/employee.component';
import { LoginComponent } from '../login/login.component';
import { HomeComponent } from '../home/home.component';
import { NavigationComponent } from '../navigation/navigation.component';
import { RegisterComponent } from '../register/register.component';

```

```

import { ViewEmployeeComponent } from './view-employee/view-
employee.component';
import { UpdateEmployeeComponent } from './update-employee/update-
employee.component';

const routes: Routes = [
  { path: '', component: LoginComponent },
  { path: 'employees', component: EmployeeComponent },
  { path: 'home', component: HomeComponent },
  { path: 'navigation', component: NavigationComponent },
  { path: 'register', component: RegisterComponent },
  { path: 'view-employee', component: ViewEmployeeComponent },
  { path: 'update-employee', component: UpdateEmployeeComponent }
];

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

```

app-component.html

```

<html>

<head>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-
9ndCyUaIbzAi2FUVXJi0CjmCapSm07SnpJef0486qhLnuZ2cdeRh002iuK6FUUVM"
crossorigin="anonymous">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min
.js"
integrity="sha384-
geWf76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz"
crossorigin="anonymous"></script>
</head>

<body>
  <div class="container">

    <router-outlet></router-outlet>
  </div>
</body>

```

</html>

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { HttpClientModule } from '@angular/common/http';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { LoginComponent } from './login/login.component';
import { ReactiveFormsModule } from '@angular/forms';
import { FormsModule } from '@angular/forms';
import { EmployeeComponent } from './employee/employee.component';
import { HomeComponent } from './home/home.component';
import { NavigationComponent } from './navigation/navigation.component';
import { MatDialogModule } from '@angular/material/dialog';
import { RegisterComponent } from './register/register.component';
import { ViewEmployeeComponent } from './view-employee/view-employee.component';
import { UpdateEmployeeComponent } from './update-employee/update-employee.component';
```

```
@NgModule({
  declarations: [
    AppComponent,
    LoginComponent,
    EmployeeComponent,
    HomeComponent,
    NavigationComponent,
    RegisterComponent,
    ViewEmployeeComponent,
    UpdateEmployeeComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    ReactiveFormsModule,
    HttpClientModule,
    FormsModule,
    MatDialogModule
  ],
  providers: [],
  bootstrap: [AppComponent],
```

```
  })  
  export class AppModule { }
```

employee.service.ts

```
import { Injectable } from '@angular/core';  
import { HttpClient } from '@angular/common/http';  
import { Observable } from 'rxjs';  
import { Employee } from './employee';  
  
@Injectable({  
  providedIn: 'root'  
})  
export class EmployeeService {  
  private apiUrl = 'http://localhost:3000'; // Update with your API URL  
  
  constructor(public http: HttpClient) { }  
  
  loadEmpData(): Observable<Employee[]> {  
    return this.http.get<Employee[]>(`${this.apiUrl}/employees`);  
  }  
  
  storeEmpData(employee: any): Observable<Employee> {  
    return this.http.post<Employee>(`${this.apiUrl}/employees`, employee);  
  }  
  
  delete(id: any): Observable<any> {  
    return this.http.delete<any>(`${this.apiUrl}/employees/${id}`);  
  }  
  
  updateEmp(id: any, employee: any): Observable<Employee> {  
    return this.http.put<Employee>(`${this.apiUrl}/employees/${id}`,  
employee);  
  }  
  
  searchEmployees(searchTerm: string): Observable<Employee[]> {  
    return  
this.http.get<Employee[]>(`${this.apiUrl}/employees?first_name_like=${searchTe  
rm}`);  
  }  
  
  getAdmin(): Observable<any> {  
    return this.http.get<any>(`${this.apiUrl}/admin`);  
  }  
  updateAdmin(email: string, admin: any) {  
    const url = `http://localhost:3000/admin`;  
    const params = { email: email };  
  }
```



```

    const options = { params: params };
    return this.http.put(url, admin, options);
  }
}

```

employee.ts

```

export class Employee {
  constructor(public id: number, public first_name: string, public
last_name: string, public email: string) {
  }
}
export class Admin {
  constructor(public email: string, public password: string) { }
}

```

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": 4,
      "first_name": "Rahul Ishan",
      "last_name": "K P ",
      "email": "rahul123@gmail.com"
    },
    {
      "first_name": "Rohan",
      "last_name": "Kumar",
      "email": "rohan@gmail.com",
    }
  ]
}

```

```
        "id": 5
    }
],
"admin": {
    "email": "admin@gmail.com",
    "password": "admin"
}
}
```