db.json

```json
{
  "admin": {
    "email": "madan816@gmail.com",
    "password": "Madan@816",
    "loggedIn": true
  },

  "employees": [
    {
      "id": "1",
      "first_name": "Sebast",
      "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": "Madan Kumar",
      "last_name": "Moole",
      "email": "madanreddy816@gmail.com"
    }
  ]
}
```

App.component.html

```html
<head>
  <style>
    body {
      background: linear-gradient(to bottom, #b5e3d8, #8ac6d1, #5ca3b6); /*
Gradient background */
    }
  </style>
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
```

```
    <div class="container">
      <a class="navbar-brand" href="#">Event Management App</a>
      <button
        class="navbar-toggler"
        type="button"
        data-bs-toggle="collapse"
        data-bs-target="#navbarSupportedContent"
        aria-controls="navbarSupportedContent"
        aria-expanded="false"
        aria-label="Toggle navigation"
      >
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item" *ngIf="admin.loggedIn">
            <a class="nav-link" aria-current="page"
routerLink="/home">Home</a>
          </li>
          <li class="nav-item" *ngIf="admin.loggedIn">
            <a class="nav-link" routerLink="/employees">Employees</a>
          </li>
          <li class="nav-item" *ngIf="admin.loggedIn">
            <a class="nav-link" routerLink="/home"
(click)="logout()">Logout</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
  <router-outlet></router-outlet>
</body>
```

App.component.ts

```
import { Component, OnInit } from '@angular/core';
import { AdminOperationsService } from './admin-operations.service';
import { Admin } from 'src/admin';
import { Router, NavigationEnd } from '@angular/router';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css'],
})
export class AppComponent implements OnInit {
  title = 'event-management-app';
  admin: Admin = new Admin();

  constructor(
    private service: AdminOperationsService,
    private router: Router
  ) {}

  ngOnInit(): void {
    this.updateAdminStatus();
    this.router.events.subscribe((event) => {
      if (event instanceof NavigationEnd) {
        this.updateAdminStatus();
      }
    });
  }

  updateAdminStatus() {
    this.service.getAdminDetails().subscribe((data: any) => {
      this.admin = data;
    });
  }

  logout() {
    this.admin.loggedIn = false;
    this.service.updateAdminDetails(this.admin).subscribe(() => {
      this.updateAdminStatus();
      this.router.navigate(['']);
    });
  }
}
```

App.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-
browser/animations';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { RouterModule, Routes } from '@angular/router';
import { HomeComponent } from './home/home.component';
import { EmployeesComponent } from './employees/employees.component';
import { AdminOperationsService } from './admin-operations.service';
import { NgbModule } from '@ng-bootstrap/ng-bootstrap';
const routes: Routes = [
  {
    path: '',
    component: HomeComponent,
  },
  {
    path: 'home',
    component: HomeComponent,
  },
  {
    path: 'employees',
    component: EmployeesComponent,
  },
];
@NgModule({
  declarations: [AppComponent, HomeComponent, EmployeesComponent],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
    HttpClientModule,
    RouterModule.forRoot(routes),
    BrowserAnimationsModule,
    NgbModule
  ],
  providers: [AdminOperationsService],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

admin.ts

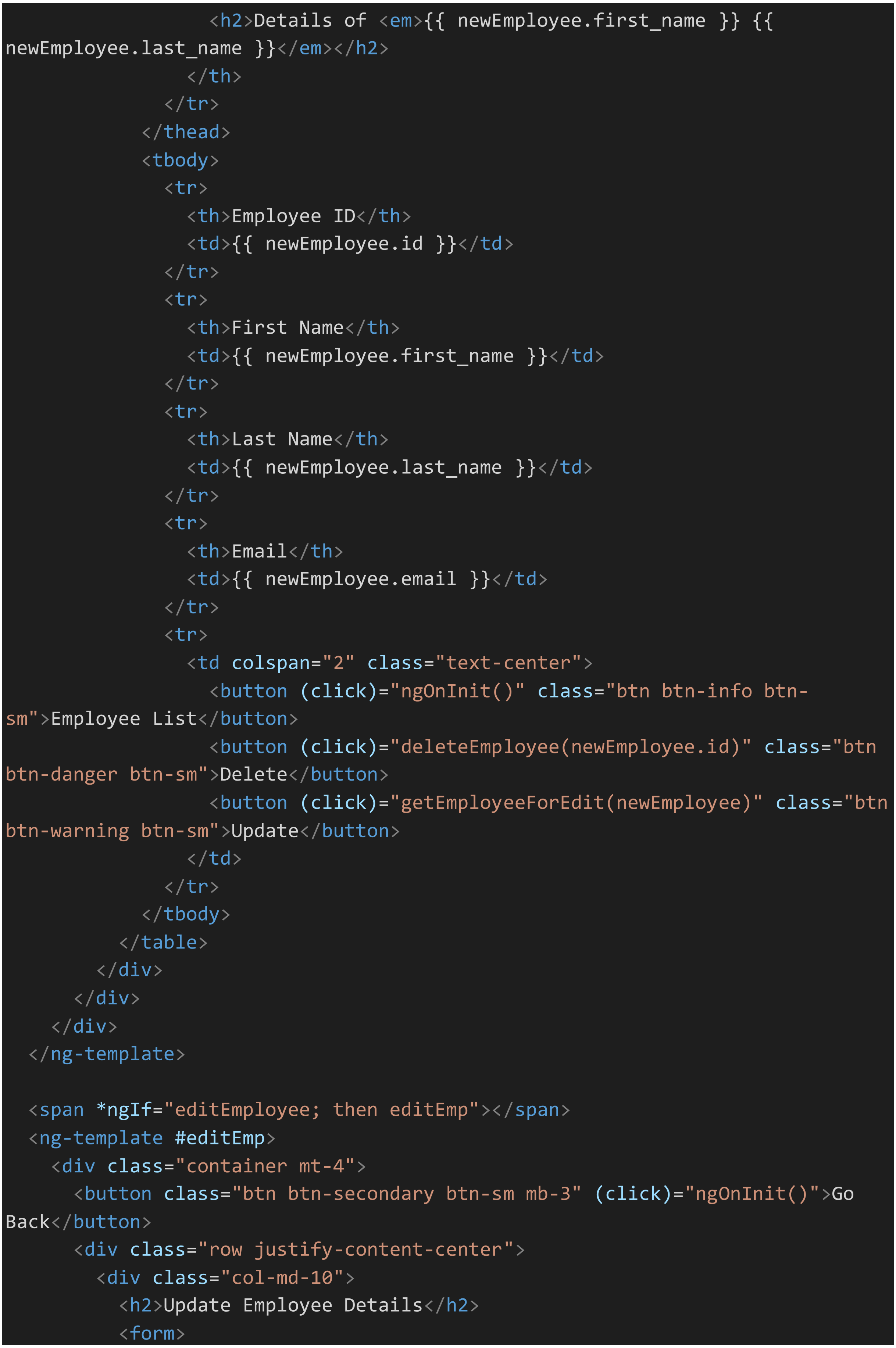
```
export class Admin {
  id!:number;
  email!: string;
  password!: string;
  loggedIn!: boolean;
}
```

Json.typing.d.ts

```
declare var $: any;
declare var stripe: any;
declare var elements: any;
```

employees.component.html

```
<div *ngIf="admin.loggedIn">
  <span *ngIf="employeeHome; then empHome"></span>
  <ng-template #empHome>
    <div class="container mt-4">
      <div class="row justify-content-center">
        <div class="col-md-10">
          <h3 [style]="style">{{ message }}</h3>
          <h2 class="text-center">List of Employees</h2>
          <div class="d-flex justify-content-between my-3">
            <button (click)="enableAdd()" class="btn btn-success btn-sm">Add
New Employee</button>
            <form class="form-inline my-2 my-lg-0"
(submit)="searchEmployee(empId)">
              <input
                class="form-control mr-sm-2"
                type="search"
                name="empId"
                id="empId"
                placeholder="Search Employee by ID"
                [(ngModel)]="empId"
                aria-label="Search"
              />
              <button class="btn btn-outline-success my-2 my-sm-0"
type="submit">Search</button>
            </form>
          </div>
          <table class="table table-dark table-striped">
            <thead>
              <tr>
                <th>ID</th>
                <th>First Name</th>
                <th>Last Name</th>
                <th>Email</th>
                <th>Actions</th>
              </tr>
            </thead>
            <tbody>
              <tr *ngFor="let employee of employees">
                <td>{{ employee.id }}</td>
                <td>{{ employee.first_name }}</td>
                <td>{{ employee.last_name }}</td>
                <td>{{ employee.email }}</td>
                <td>
                  <button class="btn btn-info btn-sm"
(click)="viewEmployee(employee)">View Details</button>
                </td>
              </tr>
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </ng-template>

  <span *ngIf="addEmployee; then addEmp"></span>
  <ng-template #addEmp>
    <div class="container mt-4">
      <button class="btn btn-secondary btn-sm mb-3" (click)="ngOnInit()">Go
Back</button>
      <div class="row justify-content-center">
        <div class="col-md-10">
          <h2>Add New Employee Details</h2>
          <form>
            <div class="form-group">
              <label for="firstName">First Name</label>
              <input type="text" class="form-control" id="firstName"
name="firstName" [(ngModel)]="newEmployee.first_name" required />
            </div>
            <div class="form-group">
              <label for="lastName">Last Name</label>
              <input type="text" class="form-control" id="lastName"
name="lastName" [(ngModel)]="newEmployee.last_name" required />
            </div>
            <div class="form-group">
              <label for="email">Email</label>
              <input type="email" class="form-control" id="email" name="email"
[(ngModel)]="newEmployee.email" required />
            </div>
            <button type="submit" class="btn btn-dark"
(click)="createEmployee()">Create</button>
          </form>
        </div>
      </div>
    </div>
  </ng-template>

  <span *ngIf="individualEmployee; then empDetails"></span>
  <ng-template #empDetails>
    <div class="container mt-4">
      <div class="row justify-content-center">
        <div class="col-md-10">
          <table class="table table-dark table-striped">
            <thead>
              <tr>
                <th colspan="2">
```

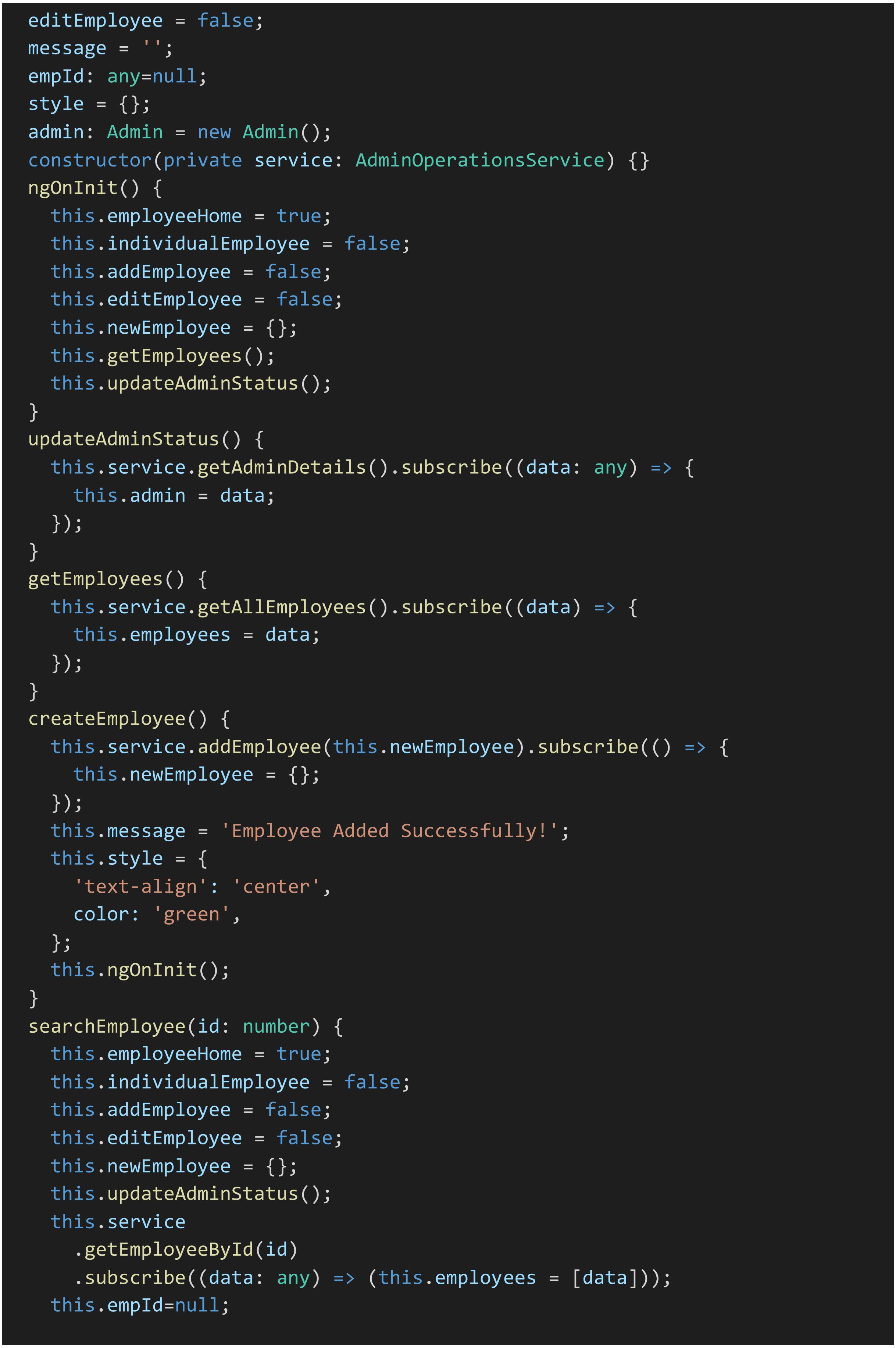


```
                  <h2>Details of <em>{{ newEmployee.first_name }} {{
newEmployee.last_name }}</em></h2>
                </th>
              </tr>
            </thead>
            <tbody>
              <tr>
                <th>Employee ID</th>
                <td>{{ newEmployee.id }}</td>
              </tr>
              <tr>
                <th>First Name</th>
                <td>{{ newEmployee.first_name }}</td>
              </tr>
              <tr>
                <th>Last Name</th>
                <td>{{ newEmployee.last_name }}</td>
              </tr>
              <tr>
                <th>Email</th>
                <td>{{ newEmployee.email }}</td>
              </tr>
              <tr>
                <td colspan="2" class="text-center">
                  <button (click)="ngOnInit()" class="btn btn-info btn-
sm">Employee List</button>
                  <button (click)="deleteEmployee(newEmployee.id)" class="btn
btn-danger btn-sm">Delete</button>
                  <button (click)="getEmployeeForEdit(newEmployee)" class="btn
btn-warning btn-sm">Update</button>
                </td>
              </tr>
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </ng-template>

  <span *ngIf="editEmployee; then editEmp"></span>
  <ng-template #editEmp>
    <div class="container mt-4">
      <button class="btn btn-secondary btn-sm mb-3" (click)="ngOnInit()">Go
Back</button>
      <div class="row justify-content-center">
        <div class="col-md-10">
          <h2>Update Employee Details</h2>
          <form>
```

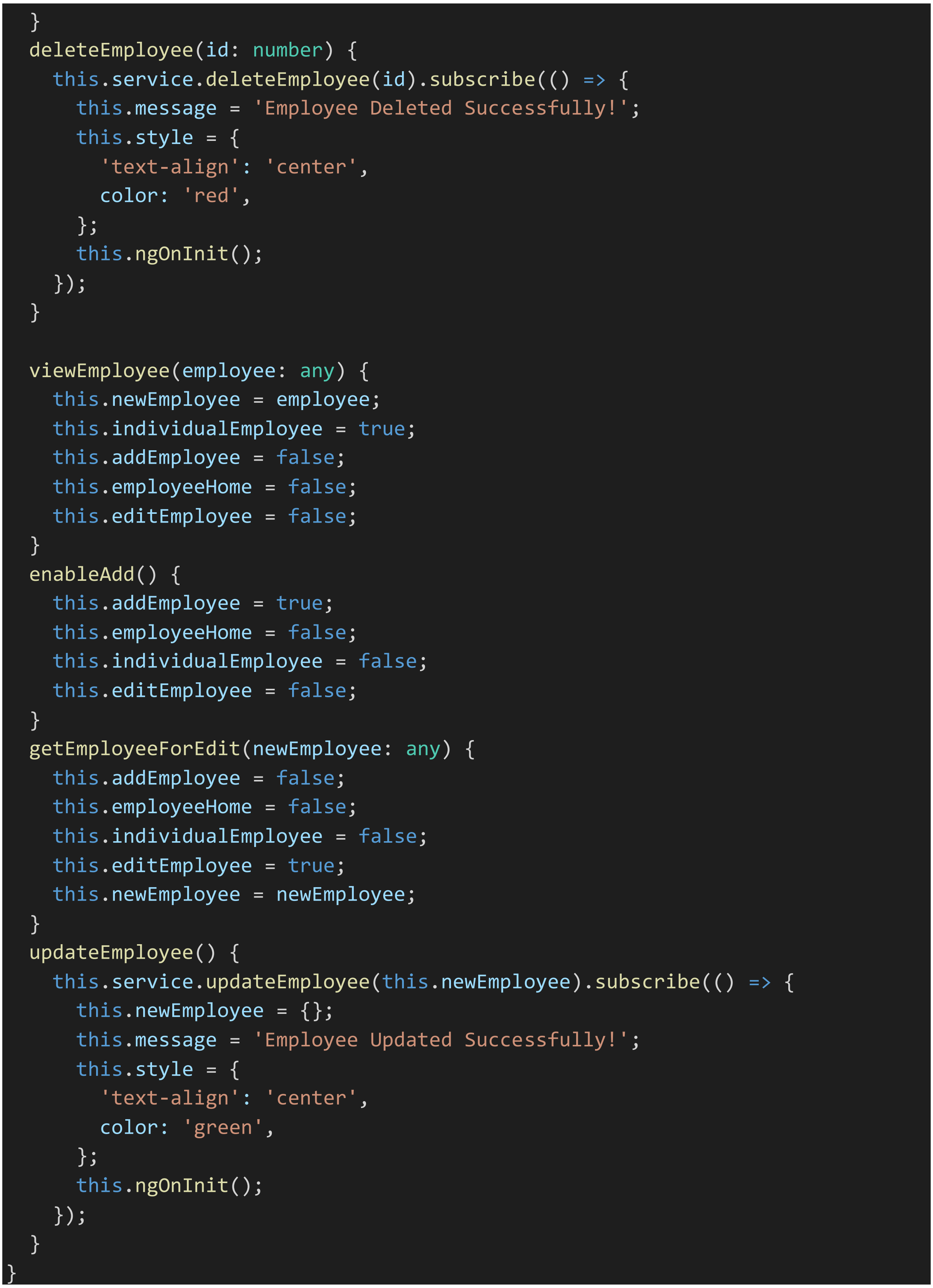
```
            <div class="form-group">
              <label for="firstName">First Name</label>
              <input type="text" class="form-control" id="firstName"
name="firstName" [(ngModel)]="newEmployee.first_name" required />
            </div>
            <div class="form-group">
              <label for="lastName">Last Name</label>
              <input type="text" class="form-control" id="lastName"
name="lastName" [(ngModel)]="newEmployee.last_name" required />
            </div>
            <div class="form-group">
              <label for="email">Email</label>
              <input type="email" class="form-control" id="email" name="email"
[(ngModel)]="newEmployee.email" required />
            </div>
            <button type="submit" class="btn btn-dark"
(click)="updateEmployee()">Update</button>
          </form>
        </div>
      </div>
    </div>
  </ng-template>
</div>

<div *ngIf="!admin.loggedIn" class="d-flex justify-content-center align-items-
center" style="height: 100vh;">
  <h1 class="text-danger text-center">Invalid Access!</h1>
</div>
```

Employees.component.ts

```
import { Component, OnInit } from '@angular/core';
import { AdminOperationsService } from '../admin-operations.service';
import { Admin } from 'src/admin';

@Component({
  selector: 'app-employees',
  templateUrl: './employees.component.html',
  styleUrls: ['./employees.component.css'],
})
export class EmployeesComponent implements OnInit {
  employees: any[] = [];
  newEmployee: any = {};
  employeeHome = true;
  individualEmployee = false;
  addEmployee = false;
```

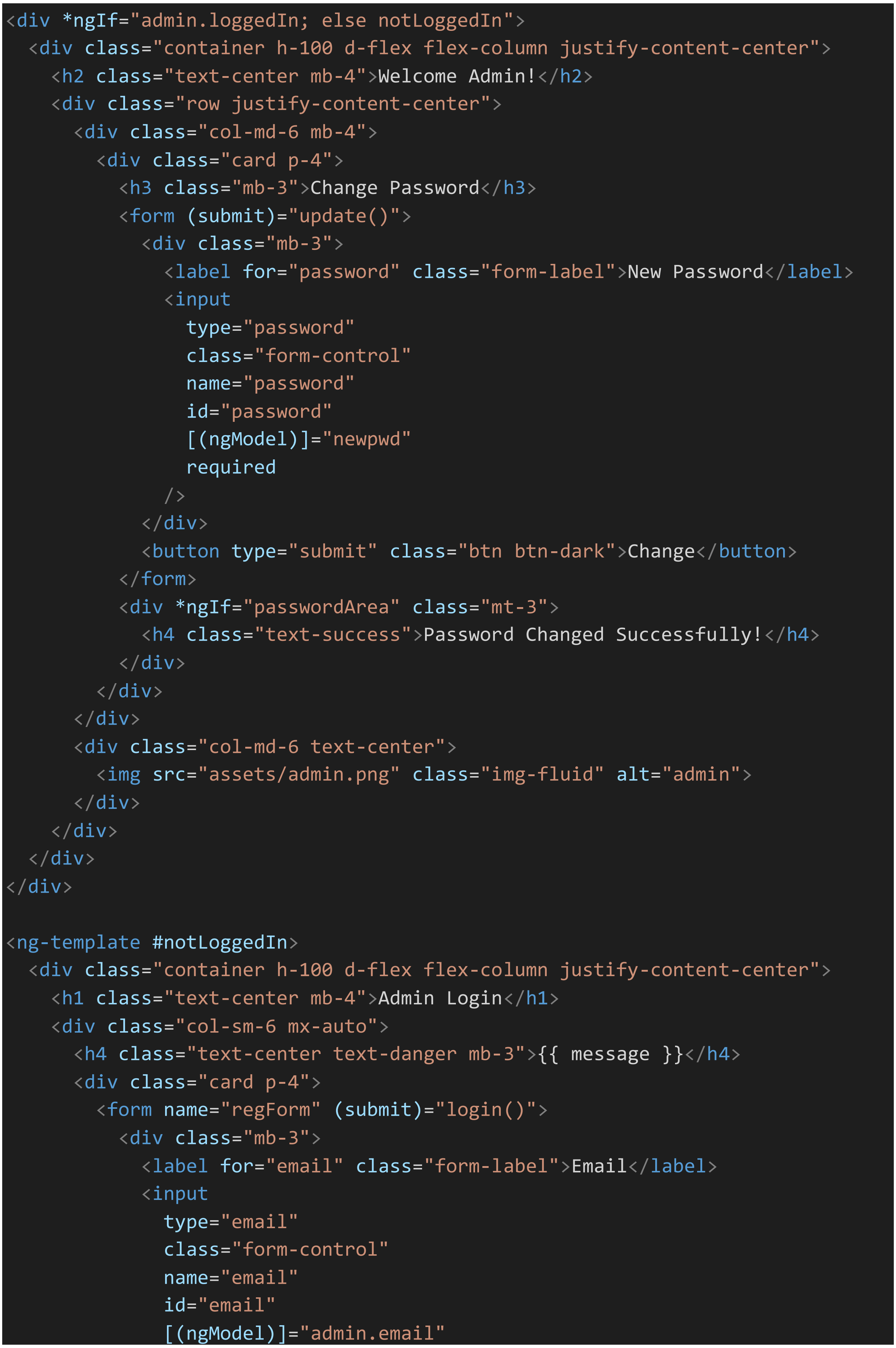


```
  editEmployee = false;
  message = '';
  empId: any=null;
  style = {};
  admin: Admin = new Admin();
  constructor(private service: AdminOperationsService) {}
  ngOnInit() {
    this.employeeHome = true;
    this.individualEmployee = false;
    this.addEmployee = false;
    this.editEmployee = false;
    this.newEmployee = {};
    this.getEmployees();
    this.updateAdminStatus();
  }
  updateAdminStatus() {
    this.service.getAdminDetails().subscribe((data: any) => {
      this.admin = data;
    });
  }
  getEmployees() {
    this.service.getAllEmployees().subscribe((data) => {
      this.employees = data;
    });
  }
  createEmployee() {
    this.service.addEmployee(this.newEmployee).subscribe(() => {
      this.newEmployee = {};
    });
    this.message = 'Employee Added Successfully!';
    this.style = {
      'text-align': 'center',
      color: 'green',
    };
    this.ngOnInit();
  }
  searchEmployee(id: number) {
    this.employeeHome = true;
    this.individualEmployee = false;
    this.addEmployee = false;
    this.editEmployee = false;
    this.newEmployee = {};
    this.updateAdminStatus();
    this.service
      .getEmployeeById(id)
      .subscribe((data: any) => (this.employees = [data]));
    this.empId=null;
```



```
  }
  deleteEmployee(id: number) {
    this.service.deleteEmployee(id).subscribe(() => {
      this.message = 'Employee Deleted Successfully!';
      this.style = {
        'text-align': 'center',
        color: 'red',
      };
      this.ngOnInit();
    });
  }

  viewEmployee(employee: any) {
    this.newEmployee = employee;
    this.individualEmployee = true;
    this.addEmployee = false;
    this.employeeHome = false;
    this.editEmployee = false;
  }
  enableAdd() {
    this.addEmployee = true;
    this.employeeHome = false;
    this.individualEmployee = false;
    this.editEmployee = false;
  }
  getEmployeeForEdit(newEmployee: any) {
    this.addEmployee = false;
    this.employeeHome = false;
    this.individualEmployee = false;
    this.editEmployee = true;
    this.newEmployee = newEmployee;
  }
  updateEmployee() {
    this.service.updateEmployee(this.newEmployee).subscribe(() => {
      this.newEmployee = {};
      this.message = 'Employee Updated Successfully!';
      this.style = {
        'text-align': 'center',
        color: 'green',
      };
      this.ngOnInit();
    });
  }
}
```

Home.component.html



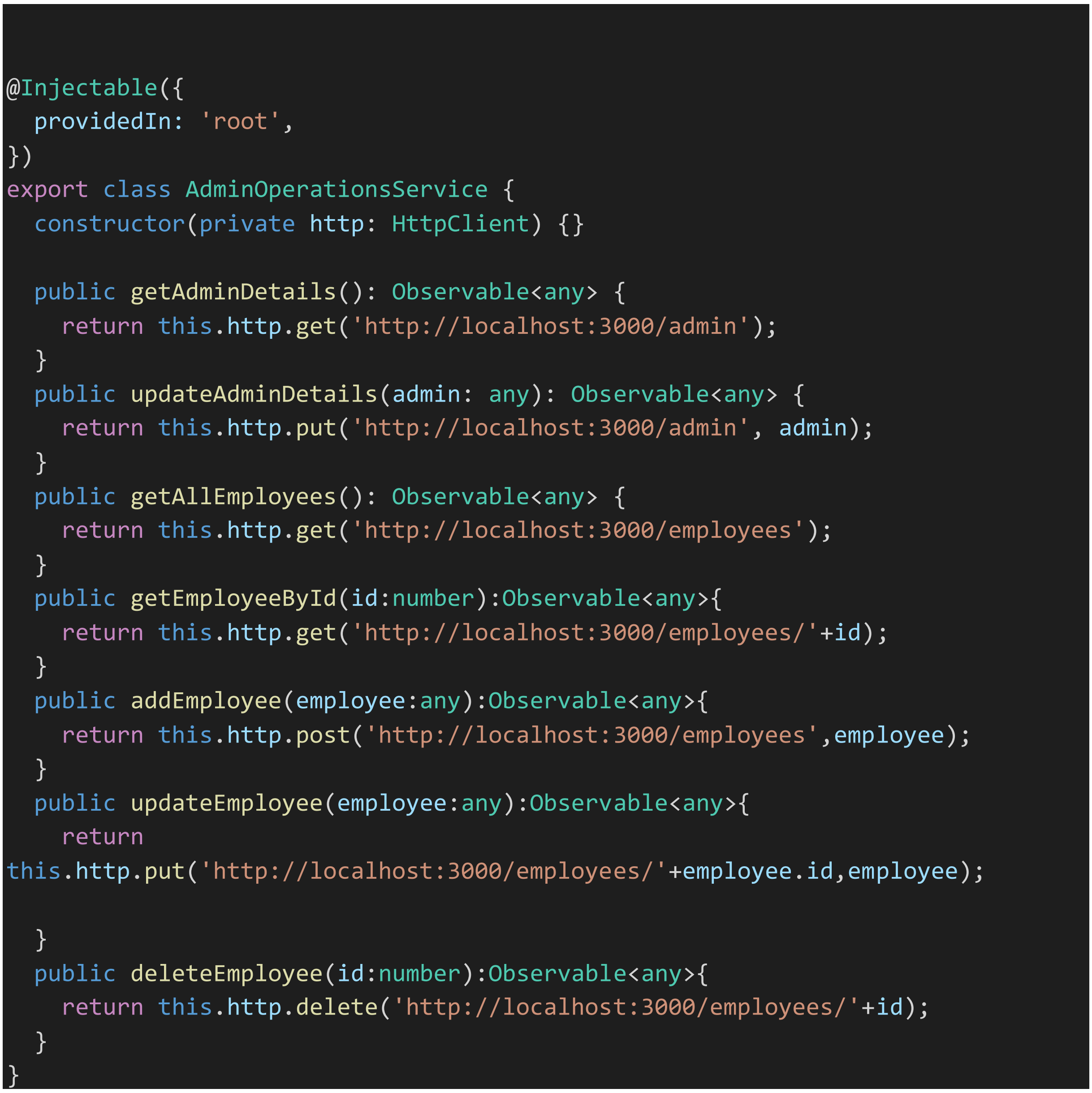
```html
<div *ngIf="admin.loggedIn; else notLoggedIn">
  <div class="container h-100 d-flex flex-column justify-content-center">
    <h2 class="text-center mb-4">Welcome Admin!</h2>
    <div class="row justify-content-center">
      <div class="col-md-6 mb-4">
        <div class="card p-4">
          <h3 class="mb-3">Change Password</h3>
          <form (submit)="update()">
            <div class="mb-3">
              <label for="password" class="form-label">New Password</label>
              <input
                type="password"
                class="form-control"
                name="password"
                id="password"
                [(ngModel)]="newpwd"
                required
              />
            </div>
            <button type="submit" class="btn btn-dark">Change</button>
          </form>
          <div *ngIf="passwordArea" class="mt-3">
            <h4 class="text-success">Password Changed Successfully!</h4>
          </div>
        </div>
      </div>
      <div class="col-md-6 text-center">
        <img src="assets/admin.png" class="img-fluid" alt="admin">
      </div>
    </div>
  </div>
</div>

<ng-template #notLoggedIn>
  <div class="container h-100 d-flex flex-column justify-content-center">
    <h1 class="text-center mb-4">Admin Login</h1>
    <div class="col-sm-6 mx-auto">
      <h4 class="text-center text-danger mb-3">{{ message }}</h4>
      <div class="card p-4">
        <form name="regForm" (submit)="login()">
          <div class="mb-3">
            <label for="email" class="form-label">Email</label>
            <input
              type="email"
              class="form-control"
              name="email"
              id="email"
              [(ngModel)]="admin.email"
              required
            />
          </div>
          <div class="mb-3">
            <label for="password" class="form-label">Password</label>
            <input
              type="password"
              class="form-control"
              name="password"
              id="password"
              [(ngModel)]="admin.password"
              required
            />
          </div>
          <button type="submit" class="btn btn-dark w-100">Login</button>
        </form>
      </div>
    </div>
  </div>
</ng-template>
```

Home.component.ts

```
import { Component } from '@angular/core';
import { AdminOperationsService } from '../admin-operations.service';
import { Admin } from 'src/admin';
import { Router } from '@angular/router';

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css'],
})
export class HomeComponent {
  admin: Admin = new Admin();
  testAdmin: Admin = new Admin();
  message = '';
  loggedIn = false;
  passwordArea=false;
  newpwd='';
  constructor(
    private service: AdminOperationsService,
    private router: Router
  ) {}

  ngOnInit() {
    this.service.getAdminDetails().subscribe((data: any) => {
      this.testAdmin = data;
      if (this.testAdmin.loggedIn) {
        this.admin.loggedIn = true;
        this.loggedIn = true;
      }
    });
  }

  login() {
    if (
      this.testAdmin.email === this.admin.email &&
      this.testAdmin.password === this.admin.password
    ) {
      this.admin.loggedIn = true;
      this.loggedIn = true;
      this.message = 'Login successful.';
      this.updateLoggedInStatus(true);
      this.router.navigate(['/home']);
    } else {
      this.admin.loggedIn = false;
      this.loggedIn = false;
      this.message = 'Invalid credentials.';
    }
  }

  updateLoggedInStatus(loggedIn: boolean) {
    this.testAdmin.loggedIn = loggedIn;
    this.service.updateAdminDetails(this.testAdmin).subscribe(() => {
      this.admin.loggedIn = loggedIn;
    });
  }
  update() {
    this.testAdmin.password=this.newpwd;
    this.service.updateAdminDetails(this.testAdmin).subscribe((data:any) => {
      this.admin=data;
    });
    this.passwordArea=true;
  }
}
```

Admin-operations.services.ts

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



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

  public getAdminDetails(): Observable<any> {
    return this.http.get('http://localhost:3000/admin');
  }
  public updateAdminDetails(admin: any): Observable<any> {
    return this.http.put('http://localhost:3000/admin', admin);
  }
  public getAllEmployees(): Observable<any> {
    return this.http.get('http://localhost:3000/employees');
  }
  public getEmployeeById(id:number):Observable<any>{
    return this.http.get('http://localhost:3000/employees/'+id);
  }
  public addEmployee(employee:any):Observable<any>{
    return this.http.post('http://localhost:3000/employees',employee);
  }
  public updateEmployee(employee:any):Observable<any>{
    return
this.http.put('http://localhost:3000/employees/'+employee.id,employee);

  }
  public deleteEmployee(id:number):Observable<any>{
    return this.http.delete('http://localhost:3000/employees/'+id);
  }
}
```