**Event Management App Code**

**add-employee.component.css**

.container {

    max-width: 600px;

    margin: 0auto;

    padding: 20px;

  }

  h2 {

    margin-bottom: 20px;

  }

  .form-group {

    margin-bottom: 20px;

  }

  label {

    display: block;

    font-weight: bold;

  }

  input[type="text"],

  input[type="email"] {

    width: 100%;

    padding: 8px;

    font-size: 16px;

    border: 1pxsolid#ccc;

    border-radius: 4px;

  }

  button[type="submit"],

  button[type="button"] {

    padding: 10px20px;

    font-size: 16px;

    border: none;

    border-radius: 4px;

    cursor: pointer;

    margin-right: 10px; /\* Add space between buttons \*/

  }

  .btn-primary {

    background-color: #007bff;

    color: #fff;

  }

  .btn-secondary {

    background-color: #6c757d;

    color: #fff;

  }

  .btn-primary:hover,

  .btn-secondary:hover {

    opacity: 0.8;

  }

**add-employee.component.html**

<divclass="container">

    <h2>Add Employee</h2>

    <form(ngSubmit)="onSubmit()">

      <divclass="form-group">

        <labelfor="first\_name">First Name</label>

        <inputtype="text"class="form-control"id="first\_name"[(ngModel)]="newEmployee.first\_name"name="first\_name"required>

      </div>

      <divclass="form-group">

        <labelfor="last\_name">Last Name</label>

        <inputtype="text"class="form-control"id="last\_name"[(ngModel)]="newEmployee.last\_name"name="last\_name"required>

      </div>

      <divclass="form-group">

        <labelfor="email">Email</label>

        <inputtype="email"class="form-control"id="email"[(ngModel)]="newEmployee.email"name="email"required>

      </div>

      <buttontype="submit"class="btnbtn-primary">Submit</button>

      <buttontype="button"class="btnbtn-secondary"(click)="goHome()">Back</button>

    </form>

</div>

**add-employee.component.ts**

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

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

@Component({

  selector:'app-add-employee',

  templateUrl:'./add-employee.component.html',

  styleUrls: ['./add-employee.component.css']

})

exportclassAddEmployeeComponent {

  newEmployee:any= {

    first\_name:'',

    last\_name:'',

    email:''

  };

  router:any;

  constructor(privatehttp:HttpClient) {}

  onSubmit() {

    this.addEmployee();

  }

  goHome() {

    window.location.href="/employees";

  }

  privateaddEmployee() {

    this.http.post('http://localhost:3000/employees', this.newEmployee)

      .subscribe(

        () => {

          this.newEmployee= {}; // Clear the form

          console.log('Employee added successfully');

          alert('New Employee Added successfully');

          // Redirect to home page

          this.router.navigate(['/employees']);

          // You can perform additional actions such as showing a success message or navigating to another page

        },

        error=> {

          console.error('Error adding employee:', error);

          // You can handle the error here, such as displaying an error message to the user

        }

      );

  }

}

**employee-details.component.css**

.container {

    max-width: 600px;

    margin: 0auto;

    padding: 20px;

  }

  h4 {

    font-size: 20px;

    margin-bottom: 20px;

  }

  table {

    width: 100%;

    border-collapse: collapse;

    margin-bottom: 20px;

  }

  th, td {

    padding: 8px;

    border: 1pxsolid#ccc;

  }

  th {

    background-color: #f2f2f2;

    font-weight: bold;

  }

  .btn {

    margin-right: 10px;

  }

**employee-details.component.html**

<divclass="container">

    <h4>Details of <strong>{{ employee?.first\_name }} {{ employee?.last\_name }}</strong></h4>

    <div\*ngIf="employee">

      <tableclass="table">

        <tr>

          <th>Employee ID</th>

          <td>{{ employee.id }}</td>

        </tr>

        <tr>

          <th>First Name</th>

          <td>{{ employee.first\_name }}</td>

        </tr>

        <tr>

          <th>Last Name</th>

          <td>{{ employee.last\_name }}</td>

        </tr>

        <tr>

          <th>Email</th>

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

        </tr>

      </table>

      <buttonclass="btnbtn-primary"(click)="goHomePage()">Home</button>

      <buttonclass="btnbtn-primary"(click)="updateEmployee(employee.id)">Update</button>

      <buttonclass="btnbtn-danger"(click)="deleteEmployee(employee.id)">Delete</button>

    </div>

    <div\*ngIf="!employee">

      <p>No employee details found.</p>

    </div>

  </div>

**employee-details.component.ts**

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

import{ ActivatedRoute, Router } from'@angular/router';

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

@Component({

  selector:'app-employee-details',

  templateUrl:'./employee-details.component.html',

  styleUrls: ['./employee-details.component.css']

})

exportclassEmployeeDetailsComponentimplementsOnInit {

  employee:any;

  constructor(

    privateroute:ActivatedRoute,

    privaterouter:Router,

    privatehttp:HttpClient

  ) {}

  ngOnInit() {

    this.route.params.subscribe(params=> {

      constemployeeId=params['id'];

      this.fetchEmployeeDetails(employeeId);

    });

  }

  fetchEmployeeDetails(employeeId:number) {

    this.http.get<any>(`http://localhost:3000/employees/${employeeId}`).subscribe(

      employee=> {

        this.employee=employee;

      },

      error=> {

        console.log('Error fetching employee details:', error);

      }

    );

  }

  goHomePage() {

    this.router.navigate(['/employees']);

  }

  updateEmployee(employeeId:number) {

    this.router.navigate(['/update', employeeId]);

  }

  deleteEmployee(employeeId:number) {

    // Send the HTTP DELETE request to delete the employee by their ID

    this.http.delete(`http://localhost:3000/employees/${employeeId}`)

      .subscribe(

        () => {

          // Employee deleted successfully

          console.log('Employee deleted successfully');

          // Show success message

         alert('Employee deleted successfully');

          // Redirect to home page

          this.router.navigate(['/employees']);

        },

        error=> {

          // Error deleting employee

          console.error('Error deleting employee:', error);

          // Handle the error, such as displaying an error message to the user

        }

      );

  }

}

**employee-management.component.css**

.container {

    width: 100vw;

    height: 100vh;

    flex-direction: column;

    justify-content: left;

    align-items: start;

    padding: 20px;

  }

  nav {

    background-color: #333;

    padding: 10px;

    margin-bottom: 10px;

  }

  a {

    color: #fff;

    text-decoration: none;

  }

  a:hover {

    text-decoration: underline;

  }

  h2 {

    margin-bottom: 20px;

  }

  hr {

    margin: 20px0;

  }

  h3 {

    margin-bottom: 10px;

  }

  .table {

    width: 100%;

    border-collapse: collapse;

  }

  .tableth,

  .tabletd {

    padding: 10px;

    border: 1pxsolid#ccc;

    text-align: left;

  }

  .tableth {

    background-color: #f0f0f0;

    font-weight: bold;

  }

  .btn-primary {

    background-color: #007bff;

    color: #fff;

  }

  .btn-primary:hover {

    opacity: 0.8;

  }

  .btn-sm {

    padding: 6px12px;

    font-size: 14px;

  }

  .btn-primary[disabled] {

    opacity: 0.6;

    cursor: not-allowed;

  }

**employee-management.component.html**

<divclass="container">

    <nav>

        <ahref="/employees">Home</a>

        &nbsp;&nbsp;

        <ahref="/login">Logout</a>

      </nav>

    <h2>Event Management App</h2>

    <!-- Add Employee Form -->

    <div>

      <buttontype="submit"class="btnbtn-primary"[routerLink]="['/addEmployee']">Add New Employee</button>

    </div>

    <hr>

    <!-- Employee List -->

    <h3>Employee List</h3>

    <tableclass="table">

      <thead>

        <tr>

          <th>ID</th>

          <th>Name</th>

          <th>Email</th>

          <th>Action</th>

        </tr>

      </thead>

      <tbody>

        <tr\*ngFor="let employee of employees">

          <td>{{ employee.id }}</td>

          <td>{{ employee.first\_name }} {{ employee.last\_name }}</td>

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

          <td>

            <buttonclass="btnbtn-smbtn-primary"[routerLink]="['/employees', employee.id]">View Details</button>

          </td>

        </tr>

      </tbody>

    </table>

  </div>

**employee-management.component.spec.ts**

import{ ComponentFixture, TestBed } from'@angular/core/testing';

import{ EmployeeManagementComponent } from'./employee-management.component';

describe('EmployeeManagementComponent', () => {

  letcomponent:EmployeeManagementComponent;

  letfixture:ComponentFixture<EmployeeManagementComponent>;

  beforeEach(() => {

    TestBed.configureTestingModule({

      declarations: [EmployeeManagementComponent]

    });

    fixture=TestBed.createComponent(EmployeeManagementComponent);

    component=fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**employee-management.component.ts**

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

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

@Component({

  selector:'app-employee-management',

  templateUrl:'./employee-management.component.html',

  styleUrls: ['./employee-management.component.css']

})

exportclassEmployeeManagementComponentimplementsOnInit {

  employees:any[] = [];

  newEmployee:any= {};

  constructor(privatehttp:HttpClient) { }

  ngOnInit() {

    this.fetchEmployees();

  }

  fetchEmployees() {

    this.http.get<any[]>('http://localhost:3000/employees').subscribe(

      employees=> {

        this.employees=employees;

      },

      error=> {

        console.log('Error fetching employees:', error);

      }

    );

  }

  addEmployee() {

    this.http.post('http://localhost:3000/employees', this.newEmployee).subscribe(

      () => {

        this.newEmployee= {}; // Clear the form

        this.fetchEmployees(); // Refresh employee list

      },

      error=> {

        console.log('Error adding employee:', error);

      }

    );

  }

  viewDetails(employee:any) {

    // Implement logic to view the details of the selected employee

    console.log('View details for employee:', employee);

    // You can show the employee details in a modal or navigate to a separate page for viewing the details

  }

}

**login.component.css**

.container {

    max-width: 600px;

    margin: 0auto;

    padding: 20px;

  }

  h2 {

    margin-bottom: 20px;

  }

  .form-group {

    margin-bottom: 20px;

  }

  label {

    font-weight: bold;

  }

  input.form-control {

    width: 100%;

    padding: 8px;

    font-size: 16px;

    border: 1pxsolid#ccc;

    border-radius: 4px;

  }

  .alert {

    margin-top: 10px;

  }

  .btn-primary {

    background-color: #007bff;

    color: #fff;

  }

  .btn-primary:hover {

    opacity: 0.8;

  }

**login.component.html**

<divclass="container">

    <h2>Welcome to Event Management App</h2>

    <h4>login</h4>

    <form(ngSubmit)="login()">

      <divclass="form-group">

        <labelfor="username">Username:</label>

        <inputtype="text"id="username"name="username"class="form-control"[(ngModel)]="username"required>

      </div>

      <divclass="form-group">

        <labelfor="password">Password:</label>

        <inputtype="password"id="password"name="password"class="form-control"[(ngModel)]="password"required>

      </div>

      <divclass="alert alert-danger"\*ngIf="errorMessage">{{ errorMessage }}</div>

      <buttontype="submit"class="btnbtn-primary">Login</button>

    </form>

  </div>

**login.component.ts**

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

import{ Router } from'@angular/router';

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

@Component({

  selector:'app-login',

  templateUrl:'./login.component.html',

  styleUrls: ['./login.component.css']

})

exportclassLoginComponent {

  username:string|undefined;

  password:string|undefined;

  errorMessage:string|undefined;

  constructor(privaterouter:Router, privatehttp:HttpClient) {}

  login() {

    // Perform login logic here

    // Example: Check username and password against a stored value

    if (this.username==='abdul'&&this.password==='123') {

      // Navigate to the main employee management page

      this.router.navigate(['/employees']);

    } else {

      // Set error message

      this.errorMessage='Invalid username or password';

    }

  }

}

**update-employee.component.css**

.container {

    max-width: 600px;

    margin: 0auto;

    padding: 20px;

  }

  h2 {

    margin-bottom: 20px;

  }

  .form-group {

    margin-bottom: 20px;

  }

  label {

    font-weight: bold;

  }

  input[type="text"],

  input[type="email"] {

    width: 100%;

    padding: 8px;

    border: 1pxsolid#ccc;

    border-radius: 4px;

  }

  button {

    padding: 8px16px;

    background-color: #007bff;

    color: #fff;

    border: none;

    border-radius: 4px;

    cursor: pointer;

  }

  button:hover {

    background-color: #0056b3;

  }

  button:disabled {

    opacity: 0.6;

    cursor: not-allowed;

  }

**update-employee.component.html**

<divclass="container">

    <h2>Update Employee</h2>

    <form>

      <divclass="form-group">

        <labelfor="firstName">First Name</label>

        <inputtype="text"class="form-control"id="firstName"[ngModel]="employee?.first\_name"(ngModelChange)="employee.first\_name = $event"name="firstName">

      </div>

      <divclass="form-group">

        <labelfor="lastName">Last Name</label>

        <inputtype="text"class="form-control"id="lastName"[ngModel]="employee?.last\_name"(ngModelChange)="employee.last\_name = $event"name="lastName">

      </div>

      <divclass="form-group">

        <labelfor="email">Email</label>

        <inputtype="email"class="form-control"id="email"[ngModel]="employee?.email"(ngModelChange)="employee.email = $event"name="email">

      </div>

      <buttonclass="btnbtn-primary"(click)="updateEmployee()">Update</button>

    </form>

  </div>

**update-employee.component.ts**

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

import{ ActivatedRoute, Router } from'@angular/router';

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

@Component({

  selector:'app-update-employee',

  templateUrl:'./update-employee.component.html',

  styleUrls: ['./update-employee.component.css']

})

exportclassUpdateEmployeeComponentimplementsOnInit {

  employeeId:number|undefined;

  employee:any;

  constructor(

    privateroute:ActivatedRoute,

    privaterouter:Router,

    privatehttp:HttpClient

  ) {}

  ngOnInit() {

    this.route.params.subscribe(params=> {

      this.employeeId=+params['id'];

      this.fetchEmployeeDetails();

    });

  }

  fetchEmployeeDetails() {

    this.http.get<any>(`http://localhost:3000/employees/${this.employeeId}`).subscribe(

      employee=> {

        this.employee=employee;

      },

      error=> {

        console.log('Error fetching employee details:', error);

      }

    );

  }

  updateEmployee() {

    if(!this.employee) {

      console.error('Employee data is not available.');

      return;

    }

    constupdatedEmployee= {

      first\_name:this.employee.first\_name,

      last\_name:this.employee.last\_name,

      email:this.employee.email

    };

    this.http.put(`http://localhost:3000/employees/${this.employeeId}`, updatedEmployee)

      .subscribe(

        () => {

          console.log('Employee updated successfully');

          alert('Employee updated successfully');

          this.router.navigate(['/employees']);

        },

        error=> {

          console.error('Error updating employee:', error);

        }

      );

  }

}

**app-routing.module.ts**

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

import{ Routes, RouterModule } from'@angular/router';

import{ LoginComponent } from'./login/login.component';

import{ EmployeeManagementComponent } from'./employee-management/employee-management.component';

import{ EmployeeDetailsComponent } from'./employee-details/employee-details.component';

import{ AddEmployeeComponent } from'./add-employee/add-employee.component';

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

constroutes:Routes= [

  { path:'', redirectTo:'/login', pathMatch:'full' },

  { path:'login', component:LoginComponent },

  { path:'addEmployee', component:AddEmployeeComponent },

  { path:'employees', component:EmployeeManagementComponent },

  { path:'update/:id', component:UpdateEmployeeComponent },

  { path:'employees/:id', component:EmployeeDetailsComponent },

  { path:'\*\*', redirectTo:'/login' } // Redirect to login for any other unknown routes

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

exportclassAppRoutingModule{ }

20)app-component.html

<divclass="container">

  <router-outlet></router-outlet>

</div>

**app-component.spec.ts**

import{ TestBed } from'@angular/core/testing';

import{ AppComponent } from'./app.component';

describe('AppComponent', () => {

  beforeEach(() =>TestBed.configureTestingModule({

    declarations: [AppComponent]

  }));

  it('should create the app', () => {

    constfixture=TestBed.createComponent(AppComponent);

    constapp=fixture.componentInstance;

    expect(app).toBeTruthy();

  });

  it(`should have as title 'event-management-app'`, () => {

    constfixture=TestBed.createComponent(AppComponent);

    constapp=fixture.componentInstance;

    expect(app.title).toEqual('event-management-app');

  });

  it('should render title', () => {

    constfixture=TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    constcompiled=fixture.nativeElementasHTMLElement;

    expect(compiled.querySelector('.content span')?.textContent).toContain('event-management-app app is running!');

  });

});

**app-component.ts**

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

@Component({

  selector:'app-root',

  templateUrl:'./app.component.html',

  styleUrls: ['./app.component.css']

})

exportclassAppComponent {

  title='event-management-app';

}

**app-module.ts**

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

import{ BrowserModule } from'@angular/platform-browser';

import{ HttpClientModule } from'@angular/common/http';

import{ FormsModule } from'@angular/forms';

import{ AppRoutingModule } from'./app-routing.module';

import{ AppComponent } from'./app.component';

import{ LoginComponent } from'./login/login.component';

import{ EmployeeManagementComponent } from'./employee-management/employee-management.component';

import{ EmployeeDetailsComponent } from'./employee-details/employee-details.component';

import{ AddEmployeeComponent } from'./add-employee/add-employee.component';

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

@NgModule({

  declarations: [

    AppComponent,

    LoginComponent,

    EmployeeManagementComponent,

    EmployeeDetailsComponent,

    AddEmployeeComponent,

    UpdateEmployeeComponent

  ],

  imports: [

    BrowserModule,

    HttpClientModule,

    AppRoutingModule,

    FormsModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

exportclassAppModule{ }

**index.html**

<!doctypehtml>

<htmllang="en">

<head>

  <metacharset="utf-8">

  <title>EventManagementApp</title>

  <basehref="/">

  <metaname="viewport"content="width=device-width, initial-scale=1">

  <linkrel="icon"type="image/x-icon"href="favicon.ico">

</head>

<body>

  <app-root></app-root>

</body>

</html>

**db.json**

{

  "employees": [

    {

      "first\_name": "Sebastian",

      "last\_name": "Eschweiler",

      "email": "sebastian@codingthesmartway.com",

      "id": 1

    },

    {

      "id": 2,

      "first\_name": "Steve",

      "last\_name": "Palmer",

      "email": "steve@codingthesmartway.com"

    },

    {

      "first\_name": "Ann",

      "last\_name": "Smith",

      "email": "ann@codingthesmartway.com",

      "id": 3

    },

  ]

}