

Event Management App:

Db.json:

```
{
  "employees": [
    {
      "first_name": "MS",
      "last_name": "Dhoni",
      "email": "dhoni@msdhoni.com",
      "id": 1
    },
    {
      "id": 2,
      "first_name": "Virat",
      "last_name": "Kolhi",
      "email": "virat@viratkolhi.com"
    },
    {
      "first_name": "Suresh",
      "last_name": "Raina",
      "email": "raina@sureshraina.com",
      "id": 3
    }
  ]
}
```

Style.css:

```
@import 'bootstrap/dist/css/bootstrap.css';
```

App.component.html:

```
<div class="container">
  <router-outlet></router-outlet>
</div>
```

App.component.ts:

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

```

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

```

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 { AppRoutingModuleModule } 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]
})
export class AppModule { }

```

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/employeemanagement.component';
import { EmployeeDetailsComponent } from './employee-
details/employeedetails.component';
import { AddEmployeeComponent } from './add-employee/add-
employee.component';
import { UpdateEmployeeComponent } from './update-
employee/updateemployee.component';
const routes: 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]
})
export class AppRoutingModule { }
```

Login

login.component.html

```
<div class="container">
  <h2>Welcome to Event Management App</h2>
  <h4>login</h4>
  <form (ngSubmit)="login()">
    <div class="form-group">
      <label for="username">Username:</label>
      <input type="text" id="username" name="username" class="form-control"
        [(ngModel)]="username" required>
    </div>
    <div class="form-group">
      <label for="password">Password:</label>
      <input type="password" id="password" name="password" class="form-
        control"
        [(ngModel)]="password" required>
    </div>
    <div class="alert alert-danger" *ngIf="errorMessage">{{ errorMessage }}</div>
    <button type="submit" class="btn btn-primary">Login</button>
  </form>
</div>
```

login.component.css

```
.container {  
  max-width: 600px;  
  margin: 0 auto;  
  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: 1px solid #ccc;  
  border-radius: 4px;  
}
```

```
.alert {  
margin-top: 10px;  
}
```

```
.btn-primary {  
background-color: #007bff;  
color: #fff;  
}
```

```
.btn-primary:hover {  
opacity: 0.8;  
}
```

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']  
})  
export class LoginComponent {  
  username: string | undefined;  
  password: string | undefined;  
  errorMessage: string | undefined;
```

```

constructor(private router: Router, private http: HttpClient) {}

login() {
  // Perform login logic here
  // Example: Check username and password against a stored value
  if (this.username === 'admin' && this.password === 'password') {
    // Navigate to the main employee management page
    this.router.navigate(['/employees']);
  } else {
    // Set error message
    this.errorMessage = 'Invalid username or password';
  }
}
}

```

Employee-management

employee-management.component.html

```

<div class="container">

  <nav>

    <a href="/employees">Home</a>

  </nav>

  <h2>Event Management App</h2>


  <!-- Add Employee Form -->

  <div>

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

  </div>

```

<hr>

<!-- Employee List -->

<h3>Employee List</h3>

<table class="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>

<button class="btn btn-sm btn-primary" [routerLink]="['/employees',
employee.id]">View Details</button>

</td>

</tr>

</tbody>

</table>

</div>

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: 20px 0;  
}
```

```
h3 {  
margin-bottom: 10px;  
}
```

```
.table {  
width: 100%;  
border-collapse: collapse;  
}
```

```
.table th,  
.table td {  
padding: 10px;  
border: 1px solid #ccc;  
text-align: left;  
}
```

```
.table th {  
  background-color: #f0f0f0;  
  font-weight: bold;  
}
```

```
.btn-primary {  
  background-color: #007bff;  
  color: #fff;  
}
```

```
.btn-primary:hover {  
  opacity: 0.8;  
}
```

```
.btn-sm {  
  padding: 6px 12px;  
  font-size: 14px;  
}
```

```
.btn-primary[disabled] {  
  opacity: 0.6;  
  cursor: not-allowed;  
}
```

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']
})

export class EmployeeManagementComponent implements OnInit {
  employees: any[] = [];
  newEmployee: any = {};
  constructor(private http: 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
}
}

```

Add-employee

add-employee.component.html

```

<div class="container">
  <h2>Add Employee</h2>

  <form (ngSubmit)="onSubmit()">
    <div class="form-group">
      <label for="first_name">First Name</label>
      <input type="text" class="form-control" id="first_name"
        [(ngModel)]="newEmployee.first_name" name="first_name" required>
    </div>
  </form>

```

```
<div class="form-group">
  <label for="last_name">Last Name</label>
  <input type="text" class="form-control" id="last_name"
    [(ngModel)]="newEmployee.last_name" name="last_name" required>
</div>
```

```
<div class="form-group">
  <label for="email">Email</label>
  <input type="email" class="form-control" id="email"
    [(ngModel)]="newEmployee.email" name="email" required>
</div>
```

```
<button type="submit" class="btn btn-primary">Submit</button>
<button type="button" class="btn btn-secondary"
  (click)="goHome()">Back</button>
</form>
</div>
```

add-employee.component.css

```
.container {
  max-width: 600px;
  margin: 0 auto;
  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: 1px solid #ccc;  
border-radius: 4px;  
}
```

```
button[type="submit"],  
button[type="button"] {  
padding: 10px 20px;  
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.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']
})
```



```
  })  
  
  export class AddEmployeeComponent {  
    newEmployee: any = {  
      first_name: "",  
      last_name: "",  
      email: ""  
    };  
  
    router: any;  
  
    constructor(private http: HttpClient) {}  
  
    onSubmit() {  
      this.addEmployee();  
    }  
  
    goHome() {  
      window.location.href = "/employees";  
    }  
  
    private addEmployee() {  
      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

employee-details.component.html

```
<div class="container">  
  <h4>Details of <strong>{{ employee?.first_name }} {{ employee?.last_name  
  }}</strong></h4>
```

```
<div *ngIf="employee">  
  <table class="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>
```

```
<button class="btn btn-primary" (click)="goHomePage()">Home</button>
<button class="btn btn-primary"
(click)="updateEmployee(employee.id)">Update</button>
<button class="btn btn-danger"
(click)="deleteEmployee(employee.id)">Delete</button>
</div>
```

```
<div *ngIf="!employee">
<p>No employee details found.</p>
</div>
</div>
```

employee-details.component.css

```
.container {
max-width: 600px;
margin: 0 auto;
padding: 20px;
```

```
}
```

```
h4 {  
font-size: 20px;  
margin-bottom: 20px;  
}
```

```
table {  
width: 100%;  
border-collapse: collapse;  
margin-bottom: 20px;  
}
```

```
th, td {  
padding: 8px;  
border: 1px solid #ccc;  
}
```

```
th {  
background-color: #f2f2f2;  
font-weight: bold;  
}
```

```
.btn {  
margin-right: 10px;  
}
```

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']
})

export class EmployeeDetailsComponent implements OnInit {
  employee: any;

  constructor(
    private route: ActivatedRoute,
    private router: Router,
    private http: HttpClient
  ) {}

  ngOnInit() {
    this.route.params.subscribe(params => {
      const employeeId = params['id'];
      this.fetchEmployeeDetails(employeeId);
    });
  }

  fetchEmployeeDetails(employeeId: number) {

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

```
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
}
);
}
}
```

Update-employee

update-employee.component.html

```
<div class="container">
  <h2>Update Employee</h2>
  <form>
    <div class="form-group">
      <label for="firstName">First Name</label>
      <input type="text" class="form-control" id="firstName"
[ngModel]="employee?.first_name" (ngModelChange)="employee.first_name
= $event"
name="firstName">
    </div>
    <div class="form-group">
      <label for="lastName">Last Name</label>
      <input type="text" class="form-control" id="lastName">
```

```
[ngModel]="employee?.last_name" (ngModelChange)="employee.last_name =
$event"
name="lastName">
</div>
<div class="form-group">
<label for="email">Email</label>
<input type="email" class="form-control" id="email"
[ngModel]="employee?.email"
(ngModelChange)="employee.email = $event" name="email">
</div>
<button class="btn btn-primary"
(click)="updateEmployee()">Update</button>
</form>
</div>
```

update-employee.component.css

```
.container {
max-width: 600px;
margin: 0 auto;
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: 1px solid #ccc;  
border-radius: 4px;  
}
```

```
button {  
padding: 8px 16px;  
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.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']  
})  
export class UpdateEmployeeComponent implements OnInit {  
  employeeId: number | undefined;  
  employee: any;  
  constructor(  
    private route: ActivatedRoute,  
    private router: Router,  
    private http: 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;

```

}

const updatedEmployee = {
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);
}
);
}
```