CODE

Add Employee

Css

```
nav {
    background-color: #333;
    padding: 10px;
    text-align: center;
}

nav a {
    color: white;
    text-decoration: none;
    margin: 0 20px;
}

.container {
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
  flex-direction: column; /* Added to make elements stack vertically */
}

h2 {
  text-align: center;
  margin-bottom: 20px; /* Added for spacing between h2 and form */
}

form {
  background-color: white;
  width: 300px;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
}

.form-group {
  margin-bottom: 20px;
}
```

```css
label {
  font-weight: bold;
}

input {
  width: 100%;
  padding: 8px;
  box-sizing: border-box;
  margin-top: 6px;
  margin-bottom: 16px;
  border: 1px solid #ccc;
  border-radius: 4px;
}

button {
  width: 100%;
  padding: 10px;
  background-color: black;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #333;
}
```

Html

```html
<nav>
  <a routerLink="/home">Home</a>
  <a routerLink="/add">Add Employee</a>
  <a routerLink="/employee-list">List of Employees</a>
</nav>


<div class="container">
  <h2>Add Employee</h2>
  <form (ngSubmit)="saveEmployee()">
      <div class="form-group">
          <label for="firstName">First Name:</label>
          <input type="text" id="firstName" [(ngModel)]="employee.first_name"
name="firstName" required>
```

```
      </div>
      <div class="form-group">
          <label for="lastName">Last Name:</label>
          <input type="text" id="lastName" [(ngModel)]="employee.last_name"
name="lastName" required>
      </div>
      <div class="form-group">
          <label for="email">Email:</label>
          <input type="email" id="email" [(ngModel)]="employee.email"
name="email" required>
      </div>
      <button type="submit">Save</button>
  </form>
</div>
```

Ts

```
import { Component } from '@angular/core';
import { EmployeeserviceService } from '../employeeservice.service';
import { Router } from '@angular/router';
import { Employee } from '../employee';

@Component({
  selector: 'app-employee',
  templateUrl: './employee.component.html',
  styleUrl: './employee.component.css'
})
export class EmployeeComponent {
  employee:Employee=new Employee();

  constructor(private employeeservice: EmployeeserviceService,private
router:Router) { }

  saveEmployee() {
    this.employeeservice.addEmployee(this.employee).subscribe();
    this.router.navigate(['/employee-list']);
    console.log(this.employee.id)
  }
}
```

Employee Details

Css

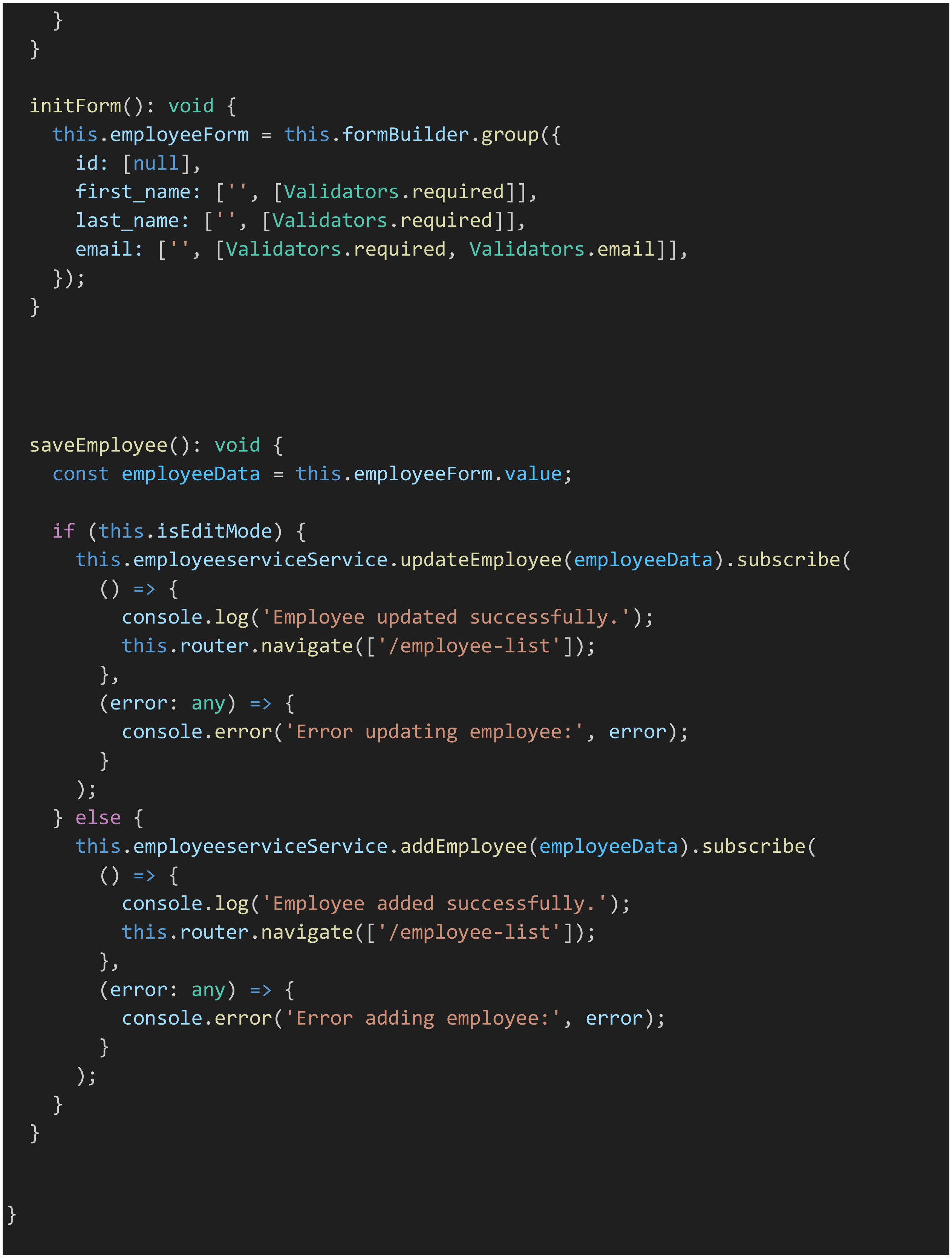
```
nav {
  background-color: #333;
  padding: 10px;
  text-align: center;
}

nav a {
  color: white;
  text-decoration: none;
  margin: 0 20px;
}

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

h2 {
  text-align: center;
}

form {
  background-color: white;
  width: 300px;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
}

.form-group {
  margin-bottom: 20px;
}

label {
  font-weight: bold;
}

input {
  width: 100%;
  padding: 8px;
  box-sizing: border-box;
  margin-top: 6px;
  margin-bottom: 16px;
  border: 1px solid #ccc;
  border-radius: 4px;
}

button {
  width: 100%;
  padding: 10px;
  background-color: black;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #333;
}
```

Html

```
<nav>
  <a routerLink="/home">Home</a>
  <a routerLink="/add">Add Employee</a>
  <a routerLink="/employee-list">List of Employees</a>
</nav>

<div class="container mt-5">
  <h2>{{ isEditMode ? 'Edit Employee' : 'Add New Employee' }}</h2>
  <form [formGroup]="employeeForm" (ngSubmit)="saveEmployee()">
      <div class="form-group">
          <label for="first_name">First Name:</label>
          <input type="text" id="first_name" formControlName="first_name"
required>
      </div>
      <div class="form-group">
          <label for="last_name">Last Name:</label>
          <input type="text" id="last_name" formControlName="last_name" required>
      </div>
      <div class="form-group">
          <label for="email">Email:</label>
          <input type="email" id="email" formControlName="email" required>
      </div>
      <button type="submit">{{ isEditMode ? 'Update' : 'Add' }} Employee</button>
  </form>
</div>
```

Ts

```
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { EmployeeserviceService } from '../employeeservice.service';
import { ActivatedRoute, Router } from '@angular/router';

@Component({
  selector: 'app-employee-details',
  templateUrl: './employee-details.component.html',
  styleUrl: './employee-details.component.css'
})
export class EmployeeDetailsComponent implements OnInit{
  employeeForm!: FormGroup;
  isEditMode: boolean = false;
  employeeId!: number;

  constructor(
    private formBuilder: FormBuilder,
    private employeeserviceService: EmployeeserviceService,
    private route: ActivatedRoute,
    private router: Router
  ) {}

  ngOnInit(): void {
    this.employeeId = +this.route.snapshot.paramMap.get('id')!;
    this.isEditMode = !!this.employeeId;
    this.initForm();

    if (this.isEditMode) {
      this.employeeserviceService.getEmployeeById(this.employeeId).subscribe(
        (employee: { [key: string]: any; }) => {
          this.employeeForm.patchValue(employee);
        },
        (error: any) => {
          console.error('Error fetching employee details:', error);
        }
      );
```



```
    }
  }

  initForm(): void {
    this.employeeForm = this.formBuilder.group({
      id: [null],
      first_name: ['', [Validators.required]],
      last_name: ['', [Validators.required]],
      email: ['', [Validators.required, Validators.email]],
    });
  }




  saveEmployee(): void {
    const employeeData = this.employeeForm.value;

    if (this.isEditMode) {
      this.employeeserviceService.updateEmployee(employeeData).subscribe(
        () => {
          console.log('Employee updated successfully.');
          this.router.navigate(['/employee-list']);
        },
        (error: any) => {
          console.error('Error updating employee:', error);
        }
      );
    } else {
      this.employeeserviceService.addEmployee(employeeData).subscribe(
        () => {
          console.log('Employee added successfully.');
          this.router.navigate(['/employee-list']);
        },
        (error: any) => {
          console.error('Error adding employee:', error);
        }
      );
    }
  }


}
```

Employee list

Css

```css
nav {
  background-color: #333;
  padding: 10px;
  text-align: center;
}

nav a {
  color: white;
  text-decoration: none;
  margin: 0 20px;
}

  .container {
    max-width: 800px;
    margin: 0 auto;
  }

  h2 {
    text-align: center;
    margin-bottom: 40px;
  }

  table {
    width: 100%;
    border-collapse: collapse;
    margin-bottom: 40px;
  }

  th, td {
    padding: 10px;
    text-align: left;
    border: 1px solid #ddd;
  }

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

  tr:nth-child(even) {
    background-color: #f5f5f5;
  }

  .btn {
    margin-right: 5px;
  }

  .btn-info {
    background-color: #007bff;
    border-color: #007bff;
    color: #fff;
    cursor: pointer;
  }

  .btn-info:hover {
    background-color: #0069d9;
    border-color: #0062cc;
  }

  .btn-danger {
    background-color: #dc3545;
    border-color: #dc3545;
    color: #fff;
    cursor: pointer;
  }

  .btn-danger:hover {
    background-color: #c82333;
    border-color: #bd2130;
  }
```

Html

```
<nav>
  <a routerLink="/home">Home</a>
  <a routerLink="/add">Add Employee</a>
  <a routerLink="/employee-list">List of Employees</a>
</nav>

<div class="container mt-5">
    <h2>Employee List</h2>
    <table class="table">
      <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"
(click)="viewEmployee(employee.id)">View</button>
            <button class="btn btn-danger"
(click)="deleteEmployee(employee.id)">Delete</button>
          </td>

        </tr>
        <tr>

        </tr>
      </tbody>
    </table>
  </div>
```

Ts

```
import { Component } from '@angular/core';
import { EmployeeserviceService } from '../employeeservice.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-employee-list',
  templateUrl: './employee-list.component.html',
  styleUrl: './employee-list.component.css'
})
export class EmployeeListComponent  {
  employees: any[] = [];


  constructor(private employeeService: EmployeeserviceService,private router:
Router) {}
```

```
  ngOnInit(): void {
    this.fetchEmployees();
  }

  fetchEmployees(): void {
    this.employeeService.getEmployees().subscribe(
      (data) => {
        this.employees = data;
      },
      (error) => {
        console.error('Error fetching employees:', error);
      }
    );
  }

  viewEmployee(id: number): void {
    // Implement logic to navigate to the employee details page
    // You may use router.navigate(['/employee-details', id]) for this
    this.router.navigate(['/employee-details',id])
    console.log('View Employee with ID:', id);
  }


  deleteEmployee(id: number): void {
       this.employeeService.deleteEmployee(id).subscribe(
      () => {
        console.log('Employee deleted successfully.');
        this.fetchEmployees();
      },
      (error) => {
        console.error('Error deleting employee:', error);
      }
    );
  }
}
```

Home

Css

```
nav {
  background-color: #333;
  padding: 10px;
  text-align: center;
}

nav a {
  color: white;
  text-decoration: none;
  margin: 0 20px;
}

h1 {
  text-align: center;
  margin-top: 50px;
}

.hero-container {
  position: relative;
  height: 90vh;
  background-image: url("/assets/party.jpg");
  background-size: cover;
  background-position: center;
  color: black;
  text-align: center;
  display: flex;
  flex-direction: column;
  justify-content:flex-start;
  margin-top: 2px;
}

.hero-text {
  font-size: 3em;
  margin-bottom: 20px;
}

body{
    background: black;
}
```

Html

```
<body>
  <nav>
    <a routerLink="/home">Home</a>
    <a routerLink="/add">Add Employee</a>
    <a routerLink="/employee-list">List of Employees</a>
  </nav>

  <div class="hero-container">
    <h1 class="hero-text">Welcome to the Event Management App!</h1>
  </div>
</body>
```

Ts

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

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

}
```

Login

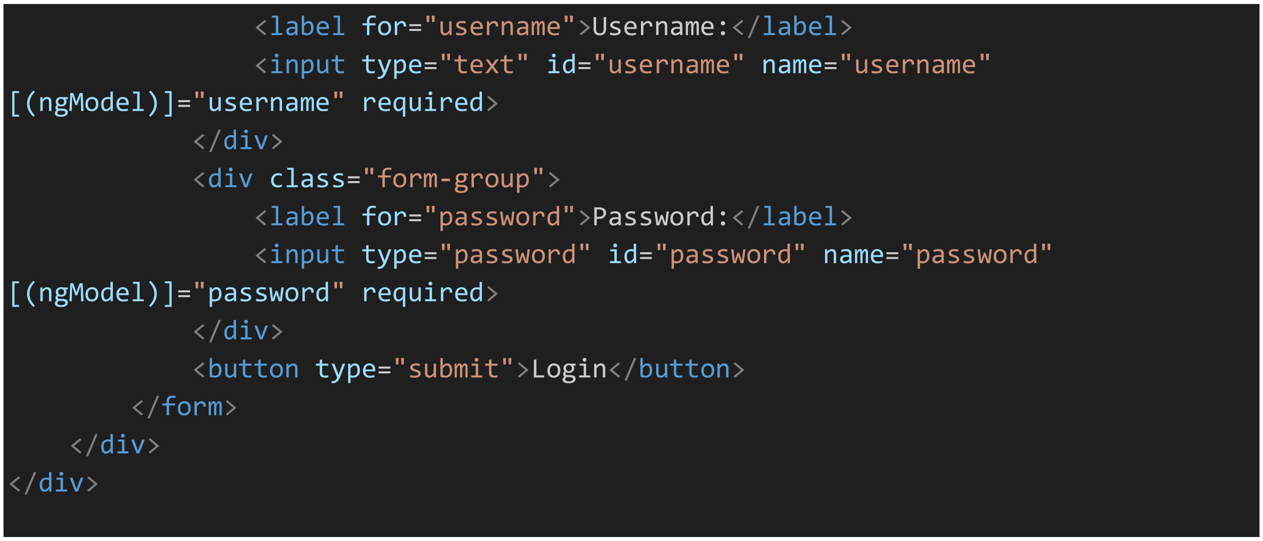
Css

```
.container {
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
}

.card {
  background-color: white;
  width: 300px;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
}

.card-title {
```

```css
  text-align: center;
}

.form-group {
  margin-bottom: 20px;
}

label {
  font-weight: bold;
}

input {
  width: 100%;
  padding: 8px;
  box-sizing: border-box;
  margin-top: 6px;
  margin-bottom: 16px;
  border: 1px solid #ccc;
  border-radius: 4px;
}

button {
  width: 100%;
  padding: 10px;
  background-color: black;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #333;
}
```

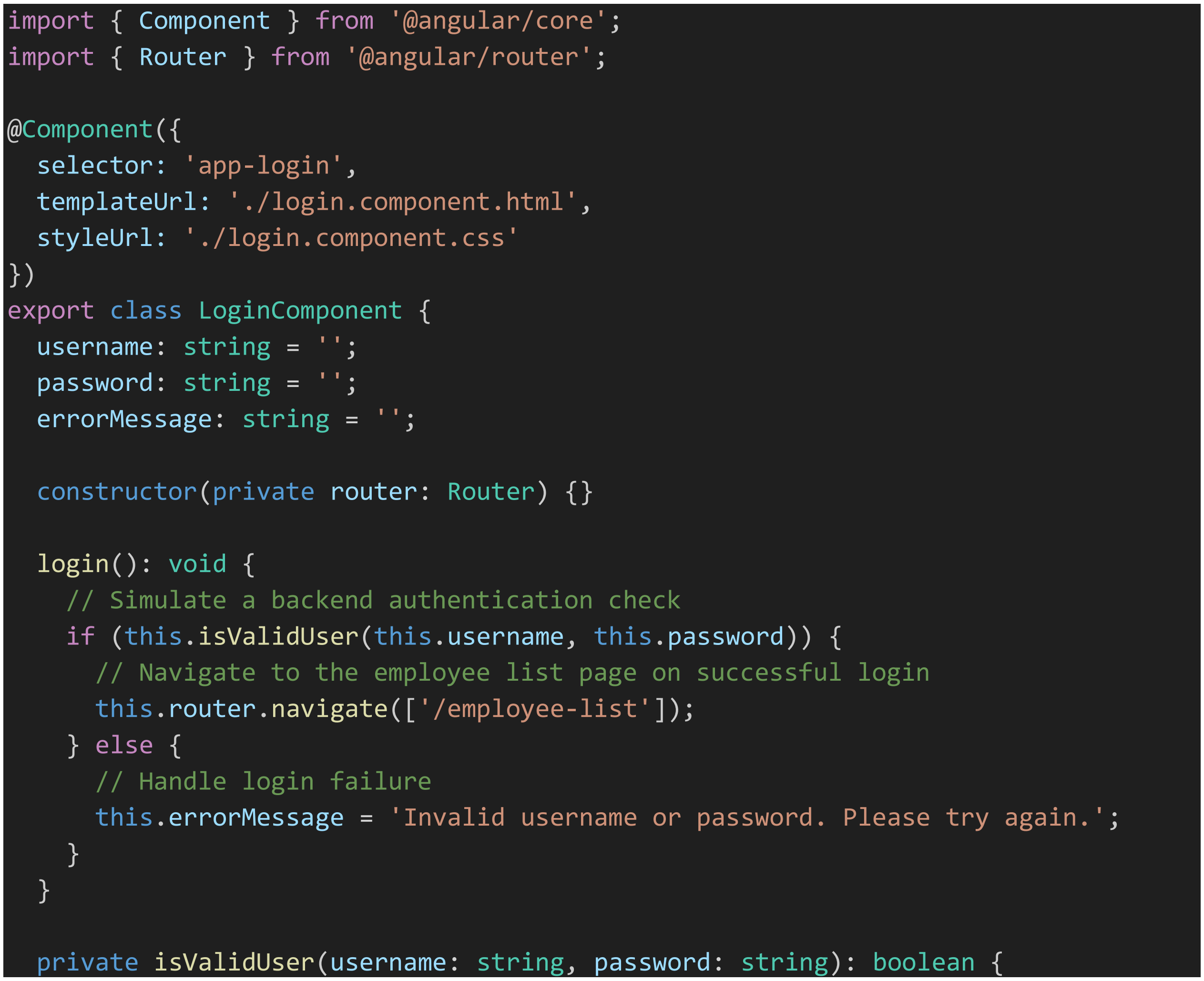
Html

```html
<div *ngIf="errorMessage" class="alert alert-danger">{{ errorMessage }}</div>

<div class="container mt-5">
    <div class="card">
        <h2 class="card-title">Admin Login</h2>
        <form (ngSubmit)="login()">
            <div class="form-group">
```



```
                <label for="username">Username:</label>
                <input type="text" id="username" name="username"
[(ngModel)]="username" required>
            </div>
            <div class="form-group">
                <label for="password">Password:</label>
                <input type="password" id="password" name="password"
[(ngModel)]="password" required>
            </div>
            <button type="submit">Login</button>
        </form>
    </div>
</div>
```

Ts



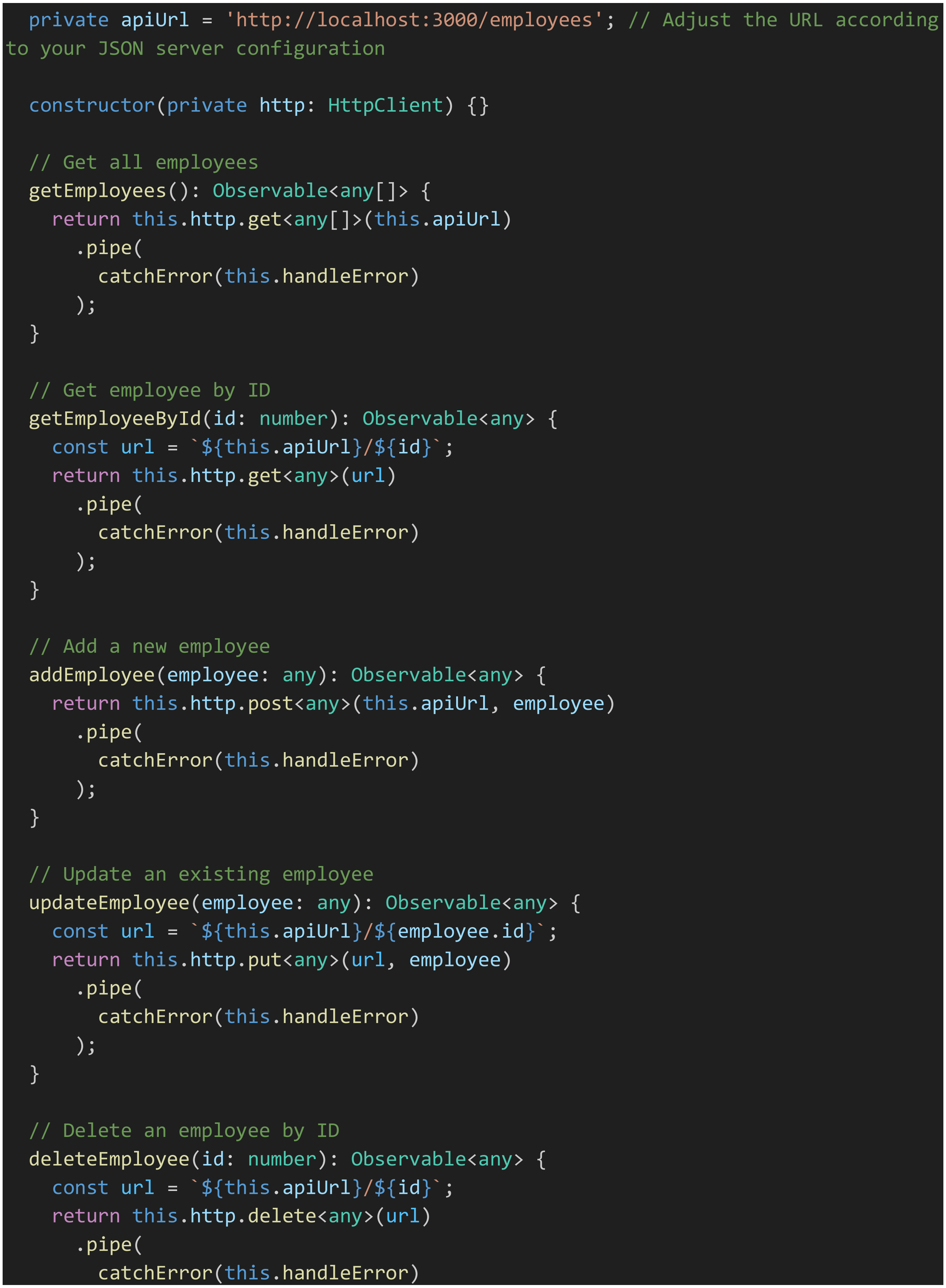
```
import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrl: './login.component.css'
})
export class LoginComponent {
  username: string = '';
  password: string = '';
  errorMessage: string = '';

  constructor(private router: Router) {}

  login(): void {
    // Simulate a backend authentication check
    if (this.isValidUser(this.username, this.password)) {
      // Navigate to the employee list page on successful login
      this.router.navigate(['/employee-list']);
    } else {
      // Handle login failure
      this.errorMessage = 'Invalid username or password. Please try again.';
    }
  }

  private isValidUser(username: string, password: string): boolean {
    // In a real application, you would make an API call to a server for
authentication
    // For simplicity, let's hardcode valid credentials here
    const validUsername = 'Naresh';
    const validPassword = 'Rao';

    return username === validUsername && password === validPassword;
  }
}
```

employee service spec

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

import { EmployeeserviceService } from './employeeservice.service';

describe('EmployeeserviceService', () => {
  let service: EmployeeserviceService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(EmployeeserviceService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});
```

employee service ts

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

@Injectable({
  providedIn: 'root'
})
export class EmployeeserviceService {
```



```
  private apiUrl = 'http://localhost:3000/employees'; // Adjust the URL according to your JSON server configuration

  constructor(private http: HttpClient) {}

  // Get all employees
  getEmployees(): Observable<any[]> {
    return this.http.get<any[]>(this.apiUrl)
      .pipe(
        catchError(this.handleError)
      );
  }

  // Get employee by ID
  getEmployeeById(id: number): Observable<any> {
    const url = `${this.apiUrl}/${id}`;
    return this.http.get<any>(url)
      .pipe(
        catchError(this.handleError)
      );
  }

  // Add a new employee
  addEmployee(employee: any): Observable<any> {
    return this.http.post<any>(this.apiUrl, employee)
      .pipe(
        catchError(this.handleError)
      );
  }

  // Update an existing employee
  updateEmployee(employee: any): Observable<any> {
    const url = `${this.apiUrl}/${employee.id}`;
    return this.http.put<any>(url, employee)
      .pipe(
        catchError(this.handleError)
      );
  }

  // Delete an employee by ID
  deleteEmployee(id: number): Observable<any> {
    const url = `${this.apiUrl}/${id}`;
    return this.http.delete<any>(url)
      .pipe(
        catchError(this.handleError)
      );
  }

  // Handle errors
  private handleError(error: any): Observable<never> {
    console.error('An error occurred:', error);
    throw error;
  }
}
```

App Module

```
import { Component, NgModule } from '@angular/core';
import {
  BrowserModule,
  provideClientHydration,
} from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { EmployeeComponent } from './employee/employee.component';
import { EmployeeDetailsComponent } from './employee-details/employee-details.component';
import { EmployeeListComponent } from './employee-list/employee-list.component';
import { HomeComponent } from './home/home.component';
import { LoginComponent } from './login/login.component';
import { RouterModule, Routes } from '@angular/router';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';

const routes: Routes = [
  { path: '', component: LoginComponent },
  { path: 'login', component: HomeComponent },
  { path: 'employee-list', component: EmployeeListComponent },
  { path: 'employee-details/:id', component: EmployeeDetailsComponent },
  { path: 'add', component: EmployeeComponent },
  { path: 'home', component: HomeComponent },
];
@NgModule({
  declarations: [
    AppComponent,
    EmployeeComponent,
    EmployeeDetailsComponent,
    EmployeeListComponent,
    HomeComponent,
    LoginComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
    HttpClientModule,
    ReactiveFormsModule,
    RouterModule.forRoot(routes),
  ],
  providers: [provideClientHydration()],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

Db json

```
{
  "employees": [
    {
      "first_name": "NARESH RAO",
      "last_name": "SS",
      "email": "naresh@gmail.com",
      "id": 1
    },
    {
      "first_name": "VIRAT ",
      "last_name": "KOHLI",
      "email": "virat@gmail.com",
      "id": 2
    },
    {
      "first_name": "CRISTIANO",
      "last_name": "RONALDO",
      "email": "ronaldo@gmail.com",
      "id": 3
    },
    {
      "first_name": "TOM",
      "last_name": "CRUISE",
      "email": "tom@gmail.com",
      "id": 4
    }
  ]
}
```