

Angular

Source Code

```
<div mat-dialog-title>
<h1>Employee Form </h1>
</div>
<form [formGroup]="empForm" (ngSubmit)="onFormSubmit()">
<div mat-dialog-content>
  <div class="row">
    <mat-form-field appearance="outline">
      <mat-label>First Name</mat-label>
      <input matInput type="text" placeholder="Ex. Juan"
formControlName="firstName">
    </mat-form-field>

    <mat-form-field appearance="outline">
      <mat-label>Last Name</mat-label>
      <input matInput type="text" placeholder="Ex. Dela Cruz"
formControlName="lastName">
    </mat-form-field>
  </div>
  <div class="row">
    <mat-form-field appearance="outline">
      <mat-label>Email</mat-label>
      <input matInput type="email" placeholder="Ex. Juan@gmail.com"
formControlName="email">
    </mat-form-field>

    <mat-form-field appearance="outline">
      <mat-label>Date of birth</mat-label>
      <input matInput [matDatepicker]="picker" formControlName="dob">
      <mat-hint>MM/DD/YYYY</mat-hint>
      <mat-datepicker-toggle matIconSuffix
[for]="picker"></mat-datepicker-toggle>
      <mat-datepicker #picker></mat-datepicker>
    </mat-form-field>
  </div>

  <div class="row">
    <mat-radio-group aria-label="Select an option" formControlName="gender">
      <mat-label><b>Gender</b></mat-label>
      <mat-radio-button value="male">Male</mat-radio-button>
      <mat-radio-button value="female">Female</mat-radio-button>
      <mat-radio-button value="others">Others</mat-radio-button>
    </mat-radio-group>
  </div>
  <div class="row">
```

```

<mat-form-field appearance="outline">
  <mat-label>Education</mat-label>
  <mat-select formControlName="education">
    <mat-option *ngFor="let val of education" [value]="val">{{val}}
  </mat-option>
</mat-select>
</mat-form-field>

  <mat-form-field appearance="outline">
    <mat-label>Company</mat-label>
    <input matInput placeholder="Ex. Google"
formControlName="company">
  </mat-form-field>
</div>

  <div class="row">
    <mat-form-field appearance="outline">
      <mat-label>Experience</mat-label>
      <input matInput placeholder="Ex. 4" type="number"
formControlName="experience">
    </mat-form-field>
    <mat-form-field appearance="outline">
      <mat-label>Package</mat-label>
      <mat-hint>Salary (in $)</mat-hint>
      <input matInput placeholder="Ex. 12" type="number"
formControlName="package">
    </mat-form-field>

  </div>
</div>
<div mat-dialog-actions class="action">
  <button mat-raised-button type="button"
[mat-dialog-close]="false">Cancel</button>
  <button mat-raised-button color="primary" type="submit">{{data ? 'Update':
'Save'}}</button>
</div>
</form>

```

emp-add-edit.component.scss

```
.content {
  padding-top: 10px;
}
.row {
  display: flex;
  gap: 10px;

  mat-form-field {
    width: 100%;
  }
}

.action {
  padding: 0px 25px 20px;
  button {
    flex: 1;
  }
}
```

emp-add-edit.component.spec.ts

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

import { EmpAddEditComponent } from './emp-add-edit.component';

describe('EmpAddEditComponent', () => {
  let component: EmpAddEditComponent;
  let fixture: ComponentFixture<EmpAddEditComponent>;

  beforeEach(() => {
    TestBed.configureTestingModule({
      declarations: [EmpAddEditComponent]
    });
    fixture = TestBed.createComponent(EmpAddEditComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

emp-add-edit.component.ts

```
import { Component, Inject, OnInit } from '@angular/core';
import { FormBuilder, FormGroup } from '@angular/forms';
import { EmployeeService } from '../service/employee.service';
import { MAT_DIALOG_DATA, MatDialogRef } from '@angular/material/dialog';
import { CoreService } from '../core/core.service';
@Component({
  selector: 'app-emp-add-edit',
  templateUrl: './emp-add-edit.component.html',
  styleUrls: ['./emp-add-edit.component.scss'],
})
export class EmpAddEditComponent implements OnInit {
  empForm: FormGroup;

  education: string[] = [
    'Matric',
    'Diploma',
    'Intermediate',
    'Graduate',
    'Post Graduate',
  ];

  constructor(
    private _fb: FormBuilder,
    private _empService: EmployeeService,
    private _dialogRef: MatDialogRef<EmpAddEditComponent>,
    @Inject(MAT_DIALOG_DATA) public data: any,
    private _coreService: CoreService
  ) {
    this.empForm = this._fb.group({
      firstName: '',
      lastName: '',
      email: '',
      dob: '',
      gender: '',
      education: '',
      company: '',
      experience: '',
      package: '',
    });
  }

  ngOnInit(): void {
    this.empForm.patchValue(this.data);
  }

  onFormSubmit() {
    if (this.empForm.valid) {
      if (this.data) {

```

```

    this._empService
      .updateEmployee(this.data.id, this.empForm.value)
      .subscribe({
        next: (val: any) => {
          this._coreService.openSnackBar('Employee detail updated!');
          this._dialogRef.close(true);
        },
        error: (err: any) => {
          console.error(err);
        },
      });
  } else {
    this._empService.addEmployee(this.empForm.value).subscribe({
      next: (val: any) => {
        this._coreService.openSnackBar('Employee added successfully');
        this._dialogRef.close(true);
      },
      error: (err: any) => {
        console.error(err);
      },
    });
  }
}
}
}
}
}

```

Now for the service folder.

employee.service.spec.ts

```

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

import { EmployeeService } from './employee.service';

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

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

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

```

employee.service.ts

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

@Injectable({
  providedIn: 'root'
})
export class EmployeeService {
  constructor(private _http: HttpClient) { }

  addEmployee(data: any): Observable <any> {
    return this._http.post('http://localhost:3000/employees', data);
  }

  updateEmployee(id: number, data: any): Observable<any> {
    return this._http.put(`http://localhost:3000/employees/${id}`, data);
  }

  getEmployeeList(): Observable <any> {
    return this._http.get('http://localhost:3000/employees');
  }

  deleteEmployee(id: number): Observable<any> {
    return this._http.delete(`http://localhost:3000/employees/${id}`)
  }
}
```

app folder.

app-routing.module.ts

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

const routes: Routes = [];

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

app.component.html

```
<mat-toolbar color="primary">
  <span>Angular Sample</span>
  <span class="example-spacer"></span>
  <button mat-raised-button (click)="openAddEditEmpForm()"> ADD EMPLOYEE</button>
</mat-toolbar>

<div class="main-body">

<mat-form-field>
  <mat-label>Filter</mat-label>
  <input matInput (keyup)="applyFilter($event)" placeholder="Ex. Mia" #input>
</mat-form-field>

<div class="table-container">
  <table mat-table [dataSource]="dataSource" matSort>

    <!-- ID Column -->
    <ng-container matColumnDef="id">
      <th mat-header-cell *matHeaderCellDef mat-sort-header> ID </th>
      <td mat-cell *matCellDef="let row"> {{row.id}} </td>
    </ng-container>

    <!-- Progress Column -->
    <ng-container matColumnDef="firstName">
      <th mat-header-cell *matHeaderCellDef mat-sort-header> First Name </th>
      <td mat-cell *matCellDef="let row"> {{row.firstName}} </td>
    </ng-container>

    <ng-container matColumnDef="lastName">
      <th mat-header-cell *matHeaderCellDef mat-sort-header> Last Name </th>
      <td mat-cell *matCellDef="let row"> {{row.lastName}} </td>
    </ng-container>

    <ng-container matColumnDef="email">
      <th mat-header-cell *matHeaderCellDef mat-sort-header> Email </th>
      <td mat-cell *matCellDef="let row"> {{row.email}} </td>
    </ng-container>

    <ng-container matColumnDef="dob">
      <th mat-header-cell *matHeaderCellDef mat-sort-header> Date of birth </th>
      <td mat-cell *matCellDef="let row"> {{row.dob | date}} </td>
    </ng-container>

    <ng-container matColumnDef="gender">
      <th mat-header-cell *matHeaderCellDef mat-sort-header> Gender </th>
    </ng-container>

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

```

        <td mat-cell *matCellDef="let row"> {{row.gender}} </td>
    </ng-container>

    <ng-container matColumnDef="education">
        <th mat-header-cell *matHeaderCellDef mat-sort-header> Education </th>
        <td mat-cell *matCellDef="let row"> {{row.education}} </td>
    </ng-container>

    <ng-container matColumnDef="company">
        <th mat-header-cell *matHeaderCellDef mat-sort-header> Company </th>
        <td mat-cell *matCellDef="let row"> {{row.company}} </td>
    </ng-container>

    <ng-container matColumnDef="experience">
        <th mat-header-cell *matHeaderCellDef mat-sort-header> Experience </th>
        <td mat-cell *matCellDef="let row"> {{row.experience}} </td>
    </ng-container>

    <ng-container matColumnDef="package">
        <th mat-header-cell *matHeaderCellDef mat-sort-header> Package </th>
        <td mat-cell *matCellDef="let row"> {{row.package | currency}} </td>
    </ng-container>

    <ng-container matColumnDef="action">
        <th mat-header-cell *matHeaderCellDef mat-sort-header> Action </th>
        <td mat-cell *matCellDef="let row">

            <button mat-icon-button color="primary" (click)="openEditForm(row)">
                <mat-icon>edit</mat-icon>
            </button>

            <button mat-icon-button color="warn" (click)="deleteEmployee(row.id)">
                <mat-icon>delete</mat-icon>
            </button>
        </td>
    </ng-container>

    <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
    <tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr>

    <!-- Row shown when there is no matching data. -->
    <tr class="mat-row" *matNoDataRow>
        <td class="mat-cell" colspan="4">No data matching the filter
        "{{input.value}}"</td>
    </tr>
</table>

```



```
<mat-paginator [pageSizeOptions]="[5, 10, 25, 100]" aria-label="Select page of
users"></mat-paginator>
</div>
</div>
```

app.component.scss

```
.example-spacer {
  flex: 1 1 auto;
}

.main-body {
  padding-top: 20px ;
  margin: 0 auto;
  max-width: 1248px;
  mat-form-field {
    width: 100%;
  }
}
```

app.component.spec.ts

```
import { TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(() => TestBed.configureTestingModule({
    imports: [RouterTestingModule],
    declarations: [AppComponent]
  }));

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'crud-app'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('crud-app');
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
  });
});
```

```
const compiled = fixture.nativeElement as HTMLElement;
expect(compiled.querySelector('.content
span')?.textContent).toContain('crud-app app is running!');
});
});
```

app.component.ts

```
import { Component, OnInit, ViewChild } from '@angular/core';
import { MatDialog } from '@angular/material/dialog';
import { EmpAddEditComponent } from '../emp-add-edit/emp-add-edit.component';
import { EmployeeService } from '../service/employee.service';
import { MatPaginator } from '@angular/material/paginator';
import { MatSort } from '@angular/material/sort';
import { MatTableDataSource } from '@angular/material/table';
import { CoreService } from '../core/core.service';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit {
  title = 'crud-app';
  displayedColumns: string[] = [
    'id',
    'firstName',
    'lastName',
    'email',
    'dob',
    'gender',
    'education',
    'company',
    'experience',
    'package',
    'action',
  ];
  dataSource!: MatTableDataSource<any>;

  @ViewChild(MatPaginator) paginator!: MatPaginator;
  @ViewChild(MatSort) sort!: MatSort;

  constructor (private _dialog: MatDialog,
    private _empService: EmployeeService,
    private _coreService: CoreService
  ) {}
```

```

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

openAddEditEmpForm() {
  const dialogRef = this._dialog.open(EmpAddEditComponent);
  dialogRef.afterClosed().subscribe({
    next: (val) => {
      if (val){
        this.getEmployeeList();
      }
    },
  });
}

getEmployeeList() {
  this._empService.getEmployeeList().subscribe({
    next: (res) => {
      this.dataSource = new MatTableDataSource(res);
      this.dataSource.sort = this.sort;
      this.dataSource.paginator = this.paginator;
    },
    error: console.log,
  });
}

applyFilter(event: Event) {
  const filterValue = (event.target as HTMLInputElement).value;
  this.dataSource.filter = filterValue.trim().toLowerCase();

  if (this.dataSource.paginator) {
    this.dataSource.paginator.firstPage();
  }
}

deleteEmployee(id: number) {
  this._empService.deleteEmployee(id).subscribe({
    next: (res) => {
      this._coreService.openSnackBar('Employee deleted!', 'done');
      this.getEmployeeList();
    },
    error: console.log,
  });
}

openEditForm(data: any) {
  const dialogRef = this._dialog.open(EmpAddEditComponent, {
    data,
  });
}

```

```

    });

    dialogRef.afterClosed().subscribe({
      next: (val) => {
        if (val) {
          this.getEmployeeList();
        }
      },
    });
  }
}

```

app.module.ts

```

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { MatToolbarModule } from '@angular/material/toolbar';
import { MatIconModule } from '@angular/material/icon';
import { MatButtonModule } from '@angular/material/button';
import { EmpAddEditComponent } from './emp-add-edit/emp-add-edit.component';
import { MatDialogModule } from '@angular/material/dialog';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatInputModule } from '@angular/material/input';
import { MatDatepickerModule } from '@angular/material/datepicker';
import { MatNativeDateModule } from '@angular/material/core';
import { MatRadioModule } from '@angular/material/radio';
import { MatSelectModule } from '@angular/material/select';
import { ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { MatTableModule } from '@angular/material/table';
import { MatPaginatorModule } from '@angular/material/paginator';
import { MatSortModule } from '@angular/material/sort';
import { MatSnackBarModule } from '@angular/material/snack-bar';

@NgModule({
  declarations: [
    AppComponent,
    EmpAddEditComponent
  ],
  imports: [
    BrowserModule,

```

```

    AppRoutingModule,
    BrowserAnimationsModule,
    MatToolbarModule,
    MatIconModule,
    MatButtonModule,
    MatDialogModule,
    MatFormFieldModule,
    MatInputModule,
    MatDatepickerModule,
    MatNativeDateModule,
    MatRadioModule,
    MatSelectModule,
    ReactiveFormsModule,
    HttpClientModule,
    MatTableModule,
    MatPaginatorModule,
    MatSortModule,
    MatSnackBarModule,
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

core.service.spec.ts

```

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

import { CoreService } from './core.service';

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

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

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

```

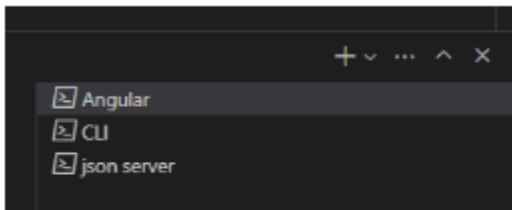
core.service.ts

```
import { Injectable } from '@angular/core';
import { MatSnackBar } from '@angular/material/snack-bar';

@Injectable({
  providedIn: 'root',
})
export class CoreService {
  constructor(private _snackBar: MatSnackBar) {}

  openSnackBar(message: string, action: string = 'ok') {
    this._snackBar.open(message, action, {
      duration: 1000,
      verticalPosition: 'top',
    });
  }
}
```

Create another Terminal and name it json server.



In the json server, type `npm install -g json-server`

```
npm install -g json-server
```

and type `json-server --watch db.json`

```
json-server --watch db.json
```

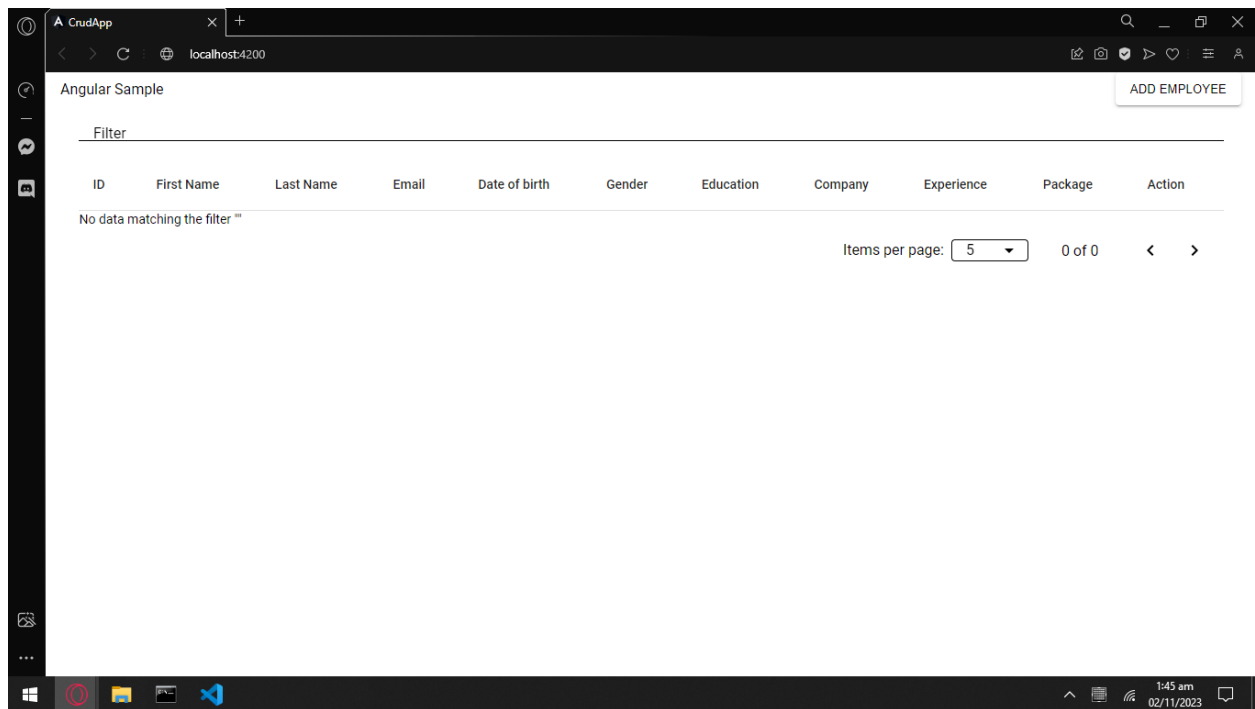
db.json

```
{
  "posts": [
    {
      "id": 1,
      "title": "json-server",
      "author": "typicode"
    }
  ],
  "comments": [
    {
```

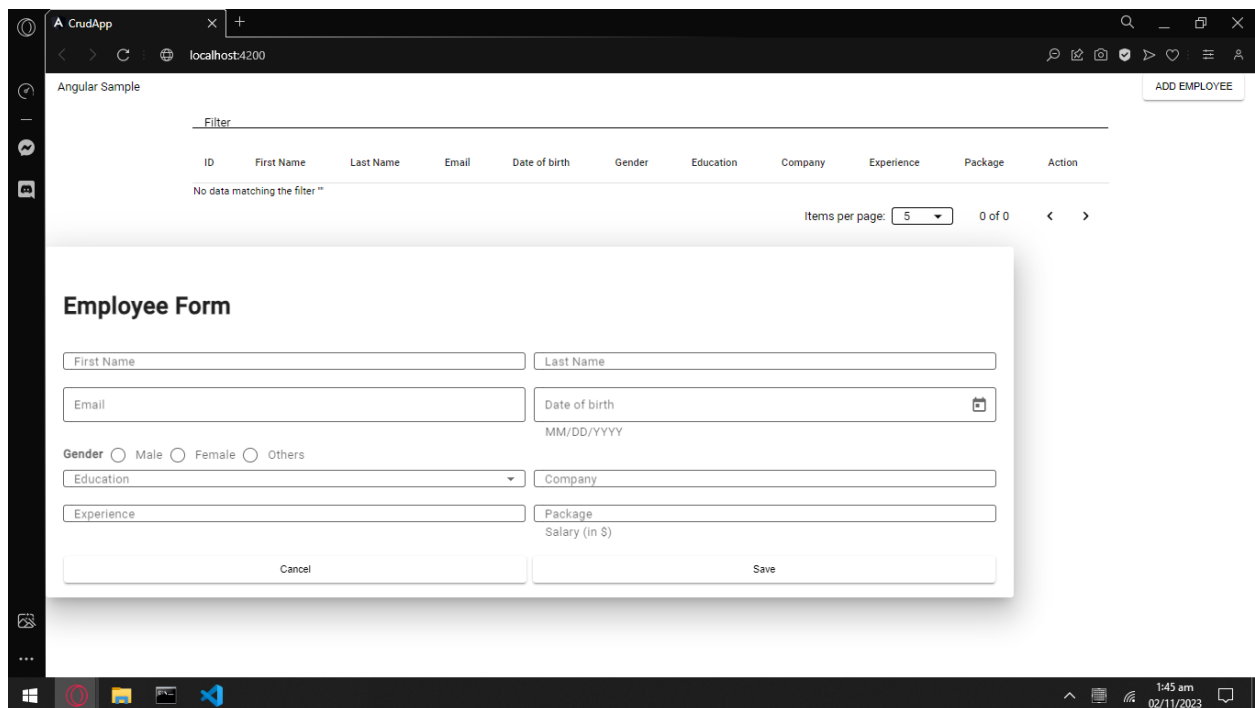
```
    "id": 1,  
    "body": "some comment",  
    "postId": 1  
  },  
],  
"profile": {  
  "name": "typicode"  
},  
"employees": []  
}
```

Output:

Index Page



Employee Form





Index Page with name on the Table

Angular Sample

ADD EMPLOYEE

Filter

ID	First Name	Last Name	Email	Date of birth	Gender	Education	Company	Experience	Package	Action
1	Anakin	Letim	maaletim@tip.edu.ph	Aug 26, 2002	male	Post Graduate	Technological Institute of The Philippines			 

Items per page: 5 1 - 1 of 1