app.component.css

table{

width: 100% !important;

}

App.component.html

<router-outlet></router-outlet>

App.component.ts

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

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

title = 'AdminPortal';

}

App.module.ts

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

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

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

import { LoginComponent } from './adminPortal/admin-login/admin-login.component';

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

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

import { HttpModule } from '@angular/http';

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

import { MatTableModule, MatCardModule, MatSortModule, MatPaginatorModule, MatDialogModule,

MatButtonModule, MatSnackBarModule } from '@angular/material';

import { MatTooltipModule } from '@angular/material/tooltip';

import { CommonModule } from '@angular/common';

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

// import { ToastrModule } from 'ngx-toastr';

import { UnitMasterComponent } from './adminPortal/Master/unit-master/unit-master.component';

import { HeaderComponent } from './adminportal/dashboard/header/header.component';

import { FooterComponent } from './adminportal/dashboard/footer/footer.component';

import { SettingComponent } from './adminportal/dashboard/setting/setting.component';

import { MenuComponent } from './adminPortal/Dashboard/Sidebar/menu.component';

import { ItemMasterComponent } from './adminPortal/Master/item-master/item-master.component';

import { AddItemComponent } from './adminPortal/Master/item-master/add-item/add-item.component';

import { AddUnitDataComponent } from './adminPortal/Master/unit-master/add-unit-data/add-unit-data.component';

import { DeleteUnitDataComponent } from './adminPortal/Master/unit-master/delete-unit-data/delete-unit-data.component';

import { AddcityDataComponent } from './adminPortal/Master/city-master/add-city-data/add-city-data.component';

import { cityMasterComponent } from './adminPortal/Master/city-master/city-master.component';

import { SupplierComponent } from './adminPortal/Master/supplier/supplier.component';

import { AddSupplierComponent } from './adminPortal/Master/supplier/add-supplier/add-supplier.component';

import { FlatMasterComponent } from './adminPortal/Master/flat-master/flat-master.component';

import { AreaMasterComponent } from './adminPortal/Master/area-master/area-master.component';

import { AddAreaComponent } from './adminPortal/Master/area-master/add-area/add-area.component';

import { SocietyMasterComponent } from './adminPortal/Master/society-master/society-master.component';

import { AddSocietyComponent } from './adminPortal/Master/society-master/add-society/add-society.component';

import { TypeMasterComponent } from './adminPortal/Master/type-master/type-master.component';

import { AddTypeComponent } from './adminPortal/Master/type-master/add-type/add-type.component';

import { AddFlatComponent } from './adminPortal/Master/flat-master/add-flat/add-flat.component';

import { CustomerMasterComponent } from './adminPortal/Master/customer-master/customer-master.component';

import { AddCustomerComponent } from './adminPortal/Master/customer-master/add-customer/add-customer.component';

import { DashComponent } from './adminPortal/dash/dash.component';

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

import { AuthService } from '@pluritech/auth-service';

import { AuthGuard } from './auth.guard';

import { AdminLayoutComponent } from './adminPortal/layout/admin-layout/admin-layout.component';

import { SimpleLayoutComponent } from './adminPortal/layout/simple-layout/simple-layout.component';

@NgModule({

declarations: [

AppComponent,

LoginComponent,

UnitMasterComponent,

ItemMasterComponent,

cityMasterComponent,

HeaderComponent,

FooterComponent,

MenuComponent,

SettingComponent,

UnitMasterComponent,

AddItemComponent,

AddUnitDataComponent,

DeleteUnitDataComponent,

AddcityDataComponent,

SupplierComponent,

AddSupplierComponent,

FlatMasterComponent,

AreaMasterComponent,

AddAreaComponent,

TypeMasterComponent,

AddTypeComponent,

SocietyMasterComponent,

AddSocietyComponent,

AddFlatComponent,

CustomerMasterComponent,

AddCustomerComponent,

AdminLayoutComponent,

SimpleLayoutComponent,

DashComponent

],

imports: [

HttpModule,

BrowserModule,

FormsModule,

HttpClientModule,

MatTableModule, MatSortModule,

BrowserAnimationsModule,

MatPaginatorModule,

MatCardModule,

AppRoutingModule,

RouterModule.forRoot([]),

MatButtonModule,

MatCardModule,

MatDialogModule,

MatTooltipModule,

CommonModule,

// ToastrModule.forRoot(),

MatSnackBarModule

],

entryComponents: [

AddUnitDataComponent,

DeleteUnitDataComponent,

AddItemComponent,

AddAreaComponent,

AddSocietyComponent,

AddTypeComponent,

AddcityDataComponent,

AddFlatComponent,

AddSupplierComponent,

AddCustomerComponent

],

providers: [AuthService, AuthGuard],

bootstrap: [AppComponent]

})

export class AppModule { }

app.routing.module.ts

// import { UnauthenticatedGuard } from './guards/unauthenticated/unauthenticated.guard';

// import { AuthenticatedGuard } from './guards/authenticated/authenticated.guard';

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

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

import { LoginComponent } from './adminPortal/admin-login/admin-login.component';

import { DashComponent } from './adminPortal/dash/dash.component';

import { UnitMasterComponent } from './adminPortal/Master/unit-master/unit-master.component';

import { ItemMasterComponent } from './adminPortal/Master/item-master/item-master.component';

import { TypeMasterComponent } from './adminPortal/Master/type-master/type-master.component';

import { AreaMasterComponent } from './adminPortal/Master/area-master/area-master.component';

import { SocietyMasterComponent } from './adminPortal/Master/society-master/society-master.component';

import { cityMasterComponent } from './adminPortal/Master/city-master/city-master.component';

import { SupplierComponent } from './adminPortal/Master/supplier/supplier.component';

import { FlatMasterComponent } from './adminPortal/Master/flat-master/flat-master.component';

import { CustomerMasterComponent } from './adminPortal/Master/customer-master/customer-master.component';

import { AuthGuard } from './auth.guard';

import { AuthService } from './auth.service';

import { AdminLayoutComponent } from './adminPortal/layout/admin-layout/admin-layout.component';

import { SimpleLayoutComponent } from './adminPortal/layout/simple-layout/simple-layout.component';

const AppRoute: Routes = [

{

path: '',

redirectTo: '/',

pathMatch: 'full'

},

{

path: 'login',

component: LoginComponent,

// canActivate: [UnauthenticatedGuard]

},

// {

// path: 'forgot',

// component: ForgotPasswordComponent,

// canActivate: [UnauthenticatedGuard]

// },

{

path: '',

component: DashComponent,

canActivate: [AuthGuard],

children: [

{path:'Admin',component: SimpleLayoutComponent

},

{ path: 'unit-master', component: UnitMasterComponent},

{ path: 'item-master', component: ItemMasterComponent },

{ path: 'type-master', component: TypeMasterComponent },

{ path: 'area-master', component: AreaMasterComponent },

{ path: 'society-master', component: SocietyMasterComponent },

{ path: 'city-master', component: cityMasterComponent },

{ path: 'suplier-master', component: SupplierComponent },

{ path: 'flat-master', component: FlatMasterComponent },

{ path: 'customer-master', component: CustomerMasterComponent },

{

path: '\*\*',

redirectTo: 'Admin'

}

]

},

]

@NgModule({

imports: [RouterModule.forChild(AppRoute)],

exports: [RouterModule],

providers: [AuthService,AuthGuard],

})

export class AppRoutingModule { }

auth-guard.ts

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

import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot } from '@angular/router';

import { Observable } from 'rxjs';

import { AuthService } from './auth.service';

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

@Injectable()

export class AuthGuard implements CanActivate {

constructor(private authService:AuthService,private router:Router){

}

canActivate(

route: ActivatedRouteSnapshot,state: RouterStateSnapshot): boolean {

if(this.authService.isLoggednIn()){

return true;

}else{

this.router.navigate(["login"]);

return false;

}

}

}

Auth-guard.service

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

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

@Injectable()

export class AuthService {

constructor(private myRoute: Router) { }

sendToken(token: string) {

localStorage.setItem("LoggedInUser", token)

}

getToken() {

return localStorage.getItem("LoggedInUser")

}

isLoggednIn() {

console.log("IsLoggedIn");

return this.getToken() !== null;

}

logout() {

console.log("logout btn ");

localStorage.removeItem("LoggedInUser");

this.myRoute.navigate(["Login"]);

}

}

ADMIN-LOGIN.CSS

@import url(https://fonts.googleapis.com/css?family=Roboto:300);

.login-page {

width: 360px;

padding: 8% 0 0;

margin: auto;

}

.form {

position: relative;

z-index: 1;

background: #FFFFFF;

max-width: 360px;

margin: 0 auto 100px;

padding: 45px;

text-align: center;

box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);

}

.form input {

font-family: "Roboto", sans-serif;

outline: 0;

background: #f2f2f2;

width: 100%;

border: 0;

margin: 0 0 15px;

padding: 15px;

box-sizing: border-box;

font-size: 14px;

}

.form button {

font-family: "Roboto", sans-serif;

text-transform: uppercase;

outline: 0;

background: #4CAF50;

width: 100%;

border: 0;

padding: 15px;

color: #FFFFFF;

font-size: 14px;

-webkit-transition: all 0.3 ease;

transition: all 0.3 ease;

cursor: pointer;

}

.form button:hover,.form button:active,.form button:focus {

background: #43A047;

}

.form .message {

margin: 15px 0 0;

color: #b3b3b3;

font-size: 12px;

}

.form .message a {

color: #4CAF50;

text-decoration: none;

}

.form .register-form {

display: none;

}

.container {

position: relative;

z-index: 1;

max-width: 300px;

margin: 0 auto;

}

.container:before, .container:after {

content: "";

display: block;

clear: both;

}

.container .info {

margin: 50px auto;

text-align: center;

}

.container .info h1 {

margin: 0 0 15px;

padding: 0;

font-size: 36px;

font-weight: 300;

color: #1a1a1a;

}

.container .info span {

color: #4d4d4d;

font-size: 12px;

}

.container .info span a {

color: #000000;

text-decoration: none;

}

.container .info span .fa {

color: #EF3B3A;

}

body {

background: #76b852; /\* fallback for old browsers \*/

background: -webkit-linear-gradient(right, #76b852, #8DC26F);

background: -moz-linear-gradient(right, #76b852, #8DC26F);

background: -o-linear-gradient(right, #76b852, #8DC26F);

background: linear-gradient(to left, #76b852, #8DC26F);

font-family: "Roboto", sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

ADMIN-LOGIN.HTML

<div class="login-page">

<div class="form">

<form class="login-form" #adminLoginForm="ngForm" (ngSubmit)="onLogin(adminLoginForm)">

<input type="text" placeholder="username" name="adminUserName" #UserName="ngModel" [(ngModel)]="admin.adminUserName" required/>

<input type="password" placeholder="password" name="adminPassword" #Password="ngModel" [(ngModel)]="admin.adminPassword" required minlength="3"/>

<button type="submit" class="btn btn-primary">Login</button><p> </p>

<p class="message">Not registered? <a href="#">Create an account</a></p>

</form>

</div>

</div>

ADMIN-LOGIN.TS

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

import { NgForm } from "@angular/forms";

import { adminService } from '../shared/services/admin.service';

import { admin } from '../shared/services/admin.modal';

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

import { AuthService } from 'src/app/auth.service';

import { MatSnackBar } from '@angular/material';

@Component({

selector: 'app-admin-login',

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

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

})

export class LoginComponent implements OnInit {

admin : admin;

constructor(private adminService : adminService,

private changeDetectorRefs: ChangeDetectorRef,

public router: Router,private authService:AuthService

,public snackBar: MatSnackBar) { }

openSnackBar(message: string, action: string) {

this.snackBar.open(message, action, {

duration: 2000,

});}

ngOnInit() {

this.resetForm();

}

resetForm(form? : NgForm)

{

if(form != null)

form.reset();

this.admin ={

adminUserName : '',

adminPassword :''

}

}

msg:string=null;

onLogin(form : NgForm){

this.adminService.adminLogin(form.value).subscribe(

data => {

console.log("POST Request is successful ", data);

this.authService.sendToken("ok");

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

},

error => {

this.openSnackBar("Error","Invalid login details")

console.log("message",error.text);

});

}

}

DASHBOARD

DASH.COMPONENT.HTML

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

import { NgForm } from "@angular/forms";

import { adminService } from '../shared/services/admin.service';

import { admin } from '../shared/services/admin.modal';

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

import { AuthService } from 'src/app/auth.service';

import { MatSnackBar } from '@angular/material';

@Component({

selector: 'app-admin-login',

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

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

})

export class LoginComponent implements OnInit {

admin : admin;

constructor(private adminService : adminService,

private changeDetectorRefs: ChangeDetectorRef,

public router: Router,private authService:AuthService

,public snackBar: MatSnackBar) { }

openSnackBar(message: string, action: string) {

this.snackBar.open(message, action, {

duration: 2000,

});}

ngOnInit() {

this.resetForm();

}

resetForm(form? : NgForm)

{

if(form != null)

form.reset();

this.admin ={

adminUserName : '',

adminPassword :''

}

}

msg:string=null;

onLogin(form : NgForm){

this.adminService.adminLogin(form.value).subscribe(

data => {

console.log("POST Request is successful ", data);

this.authService.sendToken("ok");

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

},

error => {

this.openSnackBar("Error","Invalid login details")

console.log("message",error.text);

});

}

}

DASH.COMPONENT.CSS

.sidebar-menu .treeview-menu {

padding-left: 20px;

}

.link-pointer {

cursor: pointer;

}

DASH.COMPONENT.TS

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

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

import { configuration } from './../../configuration';

import { AuthService } from '../../auth.service';

// import { AuthService } from '@pluritech/auth-service';

@Component({

selector: 'app-dash',

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

styleUrls: ['./dash.component.scss']

})

export class DashComponent implements OnInit, OnDestroy {

public projectName: any = configuration.projectName;

public userDefault: string = configuration.userDefaultImage;

private classes: string[] = [configuration.skin, 'sidebar-mini'];

constructor(

private router: Router,

private authService: AuthService

) { }

public async logout(): Promise<void> {

try {

await this.authService.logout();

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

} catch (e) {

console.log('err logout', e);

}

}

ngOnInit() {

$(document).ready(() => {

const layout = $('body').data('lte.layout');

if (layout) {

layout.fix();

}

const trees: any = $('[data-widget="tree"]');

if (trees) {

trees.tree();

}

});

const body = document.getElementsByTagName('body')[0];

for (const cl of this.classes) {

body.classList.add(cl);

}

}

ngOnDestroy() {

const body = document.getElementsByTagName('body')[0];

for (const cl of this.classes) {

body.classList.remove(cl);

}

}

}

MASTER

ITEMMASTER

ADD ITEM

ADDITEMCOMPONENT.HTML

<form class="form" #itemForm="ngForm" (ngSubmit)="onSubmit(itemForm)">

<p class='msg\_success'>{{msg}}</p>

<div class="form-group">

<div class="row">

<div class="col-md-12">

<label>Item name</label>

<input name="itemName" class="form-control" #itemName="ngModel" [(ngModel)]="itemmasterService.selectItem.itemName"

placeholder="Item Name" required pattern="[a-zA-Z ]+">

<div \*ngIf="itemName?.touched" class=" alert-danger">

<div \*ngIf="itemName.errors?.required">

Name is required

</div>

<div \*ngIf="itemName.errors?.pattern">

Enter only character.

</div>

</div>

</div>

</div>

<br>

<br>

<div class="row">

<div class="col-md-6">

<label>Item uantity</label>

<input name="itemQuantity" matTooltip="This Input Feild Have Only Nummbers" (keypress)="numberOnly($event)"

class="form-control" #itemQuantity="ngModel" [(ngModel)]="itemmasterService.selectItem.itemQuantity"

placeholder="Item Quantity" required pattern="[0-9]+">

<div \*ngIf="itemQuantity?.touched" class=" alert-danger">

<div \*ngIf="itemQuantity.errors?.required">

Name is required

</div>

<div \*ngIf="itemQuantity.errors?.pattern">

Enter only digit.

</div>

</div>

</div>

<div class="col-md-6">

<label>Item prise</label>

<input name="itemPrice" class="form-control" #itemPrice="ngModel" [(ngModel)]="itemmasterService.selectItem.itemPrice"

placeholder="Item Price" matTooltip="This Input Feild Have Only Nummbers" (keypress)="numberOnly($event)"

required pattern="[0-9]+">

<div \*ngIf="itemPrice?.touched" class=" alert-danger">

<div \*ngIf="itemPrice.errors?.required">

Name is required

</div>

<div \*ngIf="itemPrice.errors?.pattern">

Enter only digit.

</div>

</div>

</div>

</div>

<br>

<br>

<div class="row">

<div class="col-md-6">

<label>Unit name</label>

<select required name="unitId" #refUnit="ngModel" class="form-control" [(ngModel)]="itemmasterService.selectItem.unitId">

<option value="undefined">Select Unit Name</option>

<option \*ngFor="let unit of units" [ngValue]="unit.unitId">

{{unit.unitName}}

</option>

</select> </div>

<div class="col-md-6">

<label>Type name</label>

<select required name="typeId" #refType="ngModel" class="form-control" [(ngModel)]="itemmasterService.selectItem.typeId">

<option value="undefined">Select Item Type</option>

<option \*ngFor="let type of types" [ngValue]="type.typeId">

{{type.typeName}}

</option>

</select> </div>

</div>

<input name="itemId" value="0" style="visibility: collapse;" class="form-control" #unitId="ngModel" [(ngModel)]="itemmasterService.selectItem.itemId">

</div>

<div class="row">

<div class="col-md-8">

<button (click)="onCloseConfirm()" (click)="openSnackBar()" matTooltip="Submit Record" [disabled]="!itemForm.valid" type="submit" class="btn btn-lg btn-block btn-info">Submit</button>

</div>

<div class="col-md-4">

<button type="button" (click)="resetForm(itemForm)" matTooltip="Reset Form" class="btn btn-lg btn-block btn-secondary">Reset</button>

</div>

</div>

</form>

ADDITEMCOMPONENT.TS

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

import { MAT\_DIALOG\_DATA, MatDialogRef, MatSnackBar } from '@angular/material';

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

import { isEmptyObject } from 'jquery';

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

import { UnitService } from '../../../shared/services/master/unit-master.service';

import { unitMaster } from '../../../shared/model/master/unit.model';

import { ItemMasterService } from '../../../shared/services/master/item-master.service';

import { TypeMasterService } from '../../../shared/services/master/type-master.service';

import { TypeMaster } from '../../../shared/model/master/type-master';

@Component({

selector: 'app-add-item',

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

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

})

export class AddItemComponent implements OnInit {

constructor(private itemmasterService: ItemMasterService,

private unitService: UnitService,public snackBar: MatSnackBar,

private typemasterService : TypeMasterService,

private router: Router, private changeDetectorRefs: ChangeDetectorRef,

public thisdialogRef: MatDialogRef<AddItemComponent>, @Inject(MAT\_DIALOG\_DATA) public data: string) {

this.unitService.getUnitDataList().subscribe(data => this.units = data);

this. typemasterService.getTypeDataList().subscribe(data => this.types = data);

}

openSnackBar(message: string, action: string) {

this.snackBar.open( 'New Record Added Successfully', action, {

duration: 2000,

});

}

ngOnInit() {

this.resetForm();

}

units: unitMaster[];

selectUnit: number;

types: TypeMaster[];

selectType: number;

resetForm(form?: NgForm) {

if (form != null)

form.reset();

if (isEmptyObject(this.itemmasterService.selectItem)) {

this.itemmasterService.selectItem = {

itemId: 0,

itemName: '',

itemQuantity: '',

itemUnit: '',

itemPrice: null,

unitId: null,

unitMaster: null,

typeId : null,

typeMaster : null

}

}

}

msg: string = null;

onSubmit(form: NgForm) {

this.itemmasterService.postItem(form.value)

.subscribe(data => {

this.msg = 'success';

this.changeDetectorRefs.detectChanges();

this.resetForm(form);

})

}

onCloseConfirm() {

this.thisdialogRef.close('confirm');

}

numberOnly(event): boolean {

const charCode = (event.which) ? event.which : event.keyCode;

if (charCode > 31 && (charCode < 48 || charCode > 57)) {

return false;

}

return true;

}

}

ITEMMASTERCOMPONENT.CSS

.btn{

border-radius: 10px;

background: #7db9e8;

padding: 5px;

height: 40px;

box-shadow: 7px 8px 11px lightgray;

color: black;

margin-inline-start: 78%;

border: none;

font-weight:bold;

outline: none;

color: black;

font-size: 16px;

}

.search-box{

float: right;

transform: translate(-13%);

background: #7db9e8;

height: 40px;

border-radius: 40px;

box-shadow: 7px 8px 11px lightgray;

padding: 10px;

}

.search-box:hover > .search-text{

width: 80px;

padding: 0 6px;

}

.search-btn{

color: black;

float: right;

width: 40px;

height: 22px;

background: #7db9e8;

display: flex;

justify-content: center;

align-items: center;

}

.search-text{

border: none;

background: none;

font-weight:bold;

outline: none;

float: left;

padding: 0;

color: black;

font-size: 16px;

transition: 0.4s;

line-height: 22px;

width: 0px;

}

.mat-header-cell{

font-size: 15px;

font-weight:bold;

color: black;

background: #7db9e8;

box-shadow: 7px 8px 11px lightgray;

text-align: center;

}

.mat-cell{

background: #ecf0f5;

}

.mat-column-update {

flex: 0 0 60px;

color: green;

}

.mat-column-delete {

flex: 0 0 60px;

color: red;

}

.mat-paginator{

font-size: 15px;

font-weight:bold;

color: black;

text-shadow: 7px 8px 11px lightgray;

background: #7db9e8;

box-sizing: 1111px;

border-radius: 12px;

}

mat-table{

margin-top:10px;

}

ITEMMASTERCOMPONENT.HTML

<button (click)="onAddDialog()" type="submit" class="btn btn-primary btn-lg">&nbsp;<i class="fas fa-plus-circle"></i>&nbsp; Add &nbsp;</button>

<div class="search-box">

<input matInput type="text" matTooltip="Search Record" (keyup)="applyFilter($event.target.value)" class="search-text" placeholder="Search">

<a class="search-btn">

<i class="fas fa-search"></i>

</a>

</div>

<mat-table [dataSource]="dataSource" matSort>

<ng-container matColumnDef="itemId">

<mat-header-cell \*matHeaderCellDef mat-sort-header>No</mat-header-cell>

<mat-cell \*matCellDef=" let item">{{item.itemId}}</mat-cell>

</ng-container>

<ng-container matColumnDef="itemName">

<mat-header-cell \*matHeaderCellDef mat-sort-header>Name</mat-header-cell>

<mat-cell class="itemName-cell" \*matCellDef=" let item">{{item.itemName}}</mat-cell>

</ng-container>

<ng-container matColumnDef="itemQuantity">

<mat-header-cell \*matHeaderCellDef mat-sort-header>Quantity</mat-header-cell>

<mat-cell \*matCellDef=" let item">{{item.itemQuantity}}</mat-cell>

</ng-container>

<ng-container matColumnDef="itemPrice">

<mat-header-cell \*matHeaderCellDef mat-sort-header>Price</mat-header-cell>

<mat-cell \*matCellDef=" let item">{{item.itemPrice}}</mat-cell>

</ng-container>

<ng-container matColumnDef="unit">

<mat-header-cell \*matHeaderCellDef mat-sort-header>Unit</mat-header-cell>

<mat-cell \*matCellDef=" let item">{{item.unitMaster.unitName}}</mat-cell>

</ng-container>

<ng-container matColumnDef="type">

<mat-header-cell \*matHeaderCellDef mat-sort-header>Type</mat-header-cell>

<mat-cell \*matCellDef=" let item">{{item.typeMaster.typeName}}</mat-cell>

</ng-container>

<ng-container matColumnDef="update">

<mat-header-cell \*matHeaderCellDef></mat-header-cell>

<mat-cell \*matCellDef=" let item" matTooltip="Update Record" (click)="onUpdateDialog(item)"><i class="fa fa-pencil-square-o" aria-hidden="true"></i></mat-cell>

</ng-container>

<ng-container matColumnDef="delete">

<mat-header-cell \*matHeaderCellDef></mat-header-cell>

<mat-cell \*matCellDef=" let item" matTooltip="Delete Record" (click)="onDeleteDialog(item)"><i class="fa fa-trash" aria-hidden="false"></i></mat-cell>

</ng-container>

<mat-header-row \*matHeaderRowDef="displayedColumns"></mat-header-row>

<mat-row \*matRowDef="let row; columns: displayedColumns"></mat-row>

</mat-table>

<mat-paginator [length]="100" [pageSize]="8" [pageSizeOptions]="[10, 25, 100]" showFirstLastButtons>

</mat-paginator>

ITEMMASTERCOMPONENT.TS

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

import { MatSort, MatTableDataSource, MatPaginator, MatDialog } from '@angular/material';

import { ItemMasterService } from '../../shared/services/master/item-master.service';

import { AddItemComponent } from './add-item/add-item.component';

import { ItemMaster } from '../../shared/model/master/item.model';

@Component({

selector: 'app-item-master',

templateUrl: './item-master.component.html',

styleUrls: ['./item-master.component.css']

})

export class ItemMasterComponent implements OnInit {

constructor(private itemmasterservice: ItemMasterService, public dialog: MatDialog) { }

displayedColumns = ['itemId', 'itemName', 'itemQuantity','itemPrice','unit','type','update', 'delete'];

dataSource = new MatTableDataSource();

@ViewChild(MatPaginator) paginator: MatPaginator;

@ViewChild(MatSort) sort: MatSort;

applyFilter(filterValue: string) {

this.dataSource.filter = filterValue.trim().toLowerCase();

}

ngOnInit() {

this.itemmasterservice.loadData().subscribe(result => {

this.dataSource = new MatTableDataSource(result);

this.dataSource.paginator = this.paginator;

this.dataSource.sort = this.sort;

});

}

dialogResult = "";

onAddDialog() {

let dialogRef = this.dialog.open(AddItemComponent, {

width: '600',

data: 'this text'

});

dialogRef.afterClosed().subscribe(result => {

this.ngOnInit();

console.log('dialog closed: ${result}');

this.dialogResult = result;

})

}

onUpdateDialog(item: ItemMaster) {

this.itemmasterservice.selectItem = Object.assign({}, item);

let dialogRef = this.dialog.open(AddItemComponent, {

width: '900',

data: 'this text'

});

dialogRef.afterClosed().subscribe(result => {

this.ngOnInit();

console.log('dialog closed: ${result}');

this.dialogResult = result;

})

}

onDeleteDialog(item: ItemMaster) {

if (confirm('Are u sure') == true) {

this.itemmasterservice.Delete(item).subscribe(x => {

this.ngOnInit();

this.itemmasterservice.getItemDataList();

})

}

}

}

UNITMASTER

ADDUNIT

ADDUNITCOMPONENT.HTML

<form class="form" #unitForm="ngForm" (ngSubmit)="onSubmit(unitForm)">

<p class='msg\_success'>{{msg}}</p>

<div class="form-group row">

<div class="col-md-6">

<label>Unit name</label>

<input name="unitName" class="form-control" placeholder="Unit Name" #unitName="ngModel" [(ngModel)]="unitService.selectUnit.unitName" required pattern="[a-zA-Z ]+">

<div \*ngIf="unitName?.touched" class=" alert-danger">

<div \*ngIf="unitName.errors?.required">

Name is required

</div>

<div \*ngIf="unitName.errors?.pattern">

Enter only character.

</div>

</div>

</div>

<div class="col-md-6">

<label>Unit Description</label>

<input name="unitDescription" class="form-control" placeholder="Description" #unitDescription="ngModel" [(ngModel)]="unitService.selectUnit.unitDescription" required pattern="[a-zA-Z ]+">

<div \*ngIf="unitDescription?.touched" class=" alert-danger">

<div \*ngIf="unitDescription.errors?.required">

Description is required

</div>

<div \*ngIf="unitDescription.errors?.pattern">

Enter only character.

</div>

</div>

</div>

<input name="unitId" value="0" style="visibility: collapse;" class="form-control" #unitId="ngModel" [(ngModel)]="unitService.selectUnit.unitId">

</div>

<div class="row">

<div class="col-md-8">

<button (click)="onCloseConfirm()" [disabled]="!unitForm.valid" type="submit" class="btn btn-lg btn-block btn-info">Submit</button>

</div>

<div class="col-md-4">

<button type="button" class="btn btn-lg btn-block btn-secondary" (click)="resetForm(unitForm)">Reset</button>

</div>

</div>

</form>

ADDUNITCOMPONENT.TS

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

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

import { UnitService } from '../../../shared/services/master/unit-master.service';

// import { ToastrService } from 'ngx-toastr';

import { isEmptyObject } from 'jquery';

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

import { UnitMasterComponent } from '../unit-master.component';

import { MAT\_DIALOG\_DATA, MatDialogRef } from '@angular/material';

import { AddItemComponent } from '../../item-master/add-item/add-item.component';

@Component({

selector: 'app-add-unit-data',

templateUrl: './add-unit-data.component.html',

styleUrls: ['./add-unit-data.component.css']

})

export class AddUnitDataComponent implements OnInit {

constructor(private unitService : UnitService,

private router: Router,

private changeDetectorRefs: ChangeDetectorRef,

public thisdialogRef: MatDialogRef< AddUnitDataComponent>, @Inject(MAT\_DIALOG\_DATA)public data: string) { }

ngOnInit() {

this.resetForm();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Reset Form\*\*\*\*\*\*\*\*\*\*\* \*/

resetForm(form? : NgForm){

if(form != null)

form.reset();

if(isEmptyObject(this.unitService.selectUnit))

{

this.unitService.selectUnit={

unitId:0,

unitName :'',

unitDescription:'',

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*End Reset Form \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

msg:string=null;

/\*\*\*\*\*\*\*\*\*\*\*Insert Record\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

onSubmit(form : NgForm){

console.log("onsubmit sa sasa SA S AS As");

this.unitService.postUnit(form.value)

.subscribe(data=>{

this.msg='success';

this.changeDetectorRefs.detectChanges();

// this.toastr.success('Record insert successfully', 'Unit register');

})

}

onCloseConfirm(){

this.thisdialogRef.close('confirm');

}

onCloseCancel(){

this.thisdialogRef.close('cancel');

}

/\* \*\*\*\*\*\*\* End Insert Record \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

}

UNITMASTER

UNITMASTERCOMPONENT.CSS

input,.btn{

padding: 5px;

border-radius: 10px;

}

input{

float: right;

}

i:hover{

cursor: pointer;

}

.btn{

border-radius: 10px;

background: #7db9e8;

padding: 5px;

height: 40px;

box-shadow: 7px 8px 11px lightgray;

color: black;

margin-inline-start: 78%;

border: none;

font-weight:bold;

outline: none;

color: black;

font-size: 16px;

}

.search-box{

float: right;

transform: translate(-13%);

background: #7db9e8;

height: 40px;

border-radius: 40px;

box-shadow: 7px 8px 11px lightgray;

padding: 10px;

}

.search-box:hover > .search-text{

width: 80px;

padding: 0 6px;

}

.search-btn{

color: black;

float: right;

width: 40px;

height: 22px;

background: #7db9e8;

display: flex;

justify-content: center;

align-items: center;

}

.search-text{

border: none;

background: none;

font-weight:bold;

outline: none;

float: left;

padding: 0;

color: black;

font-size: 16px;

transition: 0.4s;

line-height: 22px;

width: 0px;

}

.mat-column-update {

flex: 0 0 60px;

color: green;

}

.mat-column-delete {

flex: 0 0 60px;

color: red;

}

mat-table {

margin-top: 10px;

width: 100%;

border-collapse: collapse;

}

.mat-column-update {

flex: 0 0 100px;

}

.mat-header-cell{

font-size: 15px;

font-weight:bold;

color: black;

background: #7db9e8;

box-shadow: 7px 8px 11px lightgray;

text-align: center;

}

.mat-paginator{

font-size: 15px;

font-weight:bold;

color: black;

background: #7db9e8;

box-shadow: 7px 8px 9px lightgray;

box-sizing: 1111px;

border-radius: 12px;

}

UNITMASTERCOMPONENT.HTML

<button (click)="onAddDialog()" type="submit" class="btn btn-primary btn-lg">&nbsp;<i class="fa fa-plus-circle"></i>&nbsp; Add &nbsp;</button>

<div class="search-box">

<input matInput type="text" matTooltip="Search Record" (keyup)="applyFilter($event.target.value)" class="search-text" placeholder="Search">

<a class="search-btn">

<i class="fa fa-search"></i>

</a>

</div>

<mat-table [dataSource] = "dataSource" matSort>

<ng-container matColumnDef = "unitId">

<mat-header-cell \*matHeaderCellDef mat-sort-header>Sr.No</mat-header-cell>

<mat-cell \*matCellDef = " let item">{{item.unitId}}</mat-cell>

</ng-container>

<ng-container matColumnDef = "unitName">

<mat-header-cell \*matHeaderCellDef mat-sort-header>Unit Name</mat-header-cell>

<mat-cell class="itemName-cell" \*matCellDef=" let item">{{item.unitName}}</mat-cell>

</ng-container>

<ng-container matColumnDef = "unitDescription">

<mat-header-cell \*matHeaderCellDef mat-sort-header>Unit unitDescription</mat-header-cell>

<mat-cell \*matCellDef = " let item">{{item.unitDescription}}</mat-cell>

</ng-container>

<ng-container matColumnDef = "update">

<mat-header-cell \*matHeaderCellDef mat-sort-header></mat-header-cell>

<mat-cell \*matCellDef = " let item" (click)="showForEdit(item)"><i class="fa fa-pencil-square-o" aria-hidden="true"></i></mat-cell>

</ng-container>

<ng-container matColumnDef = "delete">

<mat-header-cell \*matHeaderCellDef mat-sort-header></mat-header-cell>

<mat-cell \*matCellDef = " let item" (click)="forDelete(item)"><i class="fa fa-trash" aria-hidden="false"></i></mat-cell>

</ng-container>

<mat-header-row \*matHeaderRowDef = "displayedColumns"></mat-header-row>

<mat-row \*matRowDef = "let row; columns: displayedColumns"></mat-row>

</mat-table>

<mat-paginator [length]="100"

[pageSize]="8"

[pageSizeOptions]="[3,5, 10, 25, 100]"

showFirstLastButtons>

</mat-paginator>

UNITMASTERCOMPONENT.TS

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

// import { ToastrService } from 'ngx-toastr';

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

import { UnitService } from '../../shared/services/master/unit-master.service';

import { MatPaginator, MatSort, MatTableDataSource, MatDialog } from '@angular/material';

import { AddUnitDataComponent } from './add-unit-data/add-unit-data.component';

// import { UpdateUnitDataComponent } from './update-unit-data/update-unit-data.component';

import { DeleteUnitDataComponent } from './delete-unit-data/delete-unit-data.component';

// import { UnitMasterService } from '../../shared/services/unit-master.service';

import { unitMaster } from '../../shared/model/master/unit.model';

@Component({

selector: 'app-unit-master',

templateUrl: './unit-master.component.html',

styleUrls: ['./unit-master.component.css']

})

export class UnitMasterComponent implements OnInit {

@ViewChild(MatPaginator) paginator: MatPaginator;

@ViewChild(MatSort) sort: MatSort;

dataSource = new MatTableDataSource();

displayedColumns = ['unitId', 'unitName', 'unitDescription','update','delete'];

constructor( private UnitService: UnitService, public dialog: MatDialog) { }

applyFilter(filterValue: string) {

this.dataSource.filter = filterValue.trim().toLowerCase();

}

ngOnInit() {

this.UnitService.loadData().subscribe(result => {

console.log(result);

this.dataSource = new MatTableDataSource(result);

// sorting, paginator

this.dataSource.paginator = this.paginator;

this.dataSource.sort = this.sort;

});

}

// dialog module

dialogResult = "";

// add dialog

onAddDialog() {

this.UnitService.selectUnit = null;

let dialogRef = this.dialog.open(AddUnitDataComponent, {

width: '900',

data: 'this text'

});

dialogRef.afterClosed().subscribe(result => {

this.ngOnInit();

console.log('dialog closed: ${result}');

this.dialogResult = result;

})

}

showForEdit(unit : unitMaster){

this.UnitService.selectUnit = Object.assign({},unit);

let dialogRef = this.dialog.open(AddUnitDataComponent, {

width: '900',

data: 'this text'

});

dialogRef.afterClosed().subscribe(result => {

this.ngOnInit();

console.log('dialog closed: ${result}');

this.dialogResult = result;

})

}

forDelete(unit : unitMaster){

if(confirm('Are u sure')==true){

// this.UnitService.selectUnit=Object.assign({},unit);

this.UnitService.Delete(unit).subscribe(x=>{

this.ngOnInit();

this.UnitService.getUnitDataList();

})

}

}

// update dialog

onUpdateDialog() {

// let dialogRef = this.dialog.open(UpdateUnitDataComponent, {

// width: '600',

// data: 'this text'

// });

// dialogRef.afterClosed().subscribe(result => {

// this.ngOnInit();

// })

}

// delete dialog

onDeleteDialog() {

this.UnitService.loadData().subscribe(result => {

console.log(result);

this.dataSource = new MatTableDataSource(result);

// sorting, paginator

this.dataSource.paginator = this.paginator;

this.dataSource.sort = this.sort;

});

}

}

MODEL

ADMINMODEL.TS

export class admin {

adminUserName:string;

adminPassword:string;

}

ITEMMODEL

import { unitMaster } from '../../model/master/unit.model';

import { TypeMaster } from '../../model/master/type-master';

export class ItemMaster {

itemId: number;

itemName: string;

itemQuantity: string;

itemUnit: string;

itemPrice: number;

unitId: number;

typeId: number;

unitMaster: unitMaster;

typeMaster: TypeMaster;

}

UNITMODEL

export class unitMaster{

unitId:number;

unitName:string;

unitDescription:string;

}

SERVICES

ADMINSERVICE

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

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

import { Response, ResponseType } from '@angular/http';

import { admin } from "./admin.modal";

import { Observable } from 'rxjs';

import { catchError } from 'rxjs/operators';

@Injectable({

providedIn: 'root'

})

export class adminService {

readonly rootUrl='http://localhost:64597';

constructor(private http: HttpClient) { }

adminLogin(admin : admin){

const body: admin={

adminUserName : admin.adminUserName,

adminPassword : admin.adminPassword

}

console.log(body);

return this.http.post(this.rootUrl + '/api/Admin', body

); ;

}

}

ITEMSERVICE

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

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

import { Observable } from 'rxjs';

import { ItemMaster } from '../../model/master/item.model';

@Injectable({

providedIn: 'root'

})

export class ItemMasterService {

dataList: Observable<ItemMaster[]>;

selectItem: ItemMaster;

constructor(private http: HttpClient) { }

loadData(): Observable<ItemMaster[]> {

const url = "http://localhost:64597/api/itemMaster";

this.dataList = this.http.get<ItemMaster[]>(url);

return this.http.get<ItemMaster[]>(url);

}

postItem(ut: ItemMaster):

Observable<ItemMaster[]> {

console.log(ut);

const body: ItemMaster = {

itemId: ut.itemId,

itemName: ut.itemName,

itemQuantity: ut.itemQuantity,

itemUnit: ut.itemUnit,

itemPrice: ut.itemPrice,

unitId: ut.unitId,

unitMaster: ut.unitMaster,

typeId : ut.typeId,

typeMaster : ut.typeMaster

}

this.selectItem=null;

return this.http.post<ItemMaster[]>('http://localhost:64597/api/itemMaster', body);

}

getItemDataList(): Observable<ItemMaster[]> {

return this.http.get<ItemMaster[]>('http://localhost:64597/api/itemMaster');

}

Delete(ut: ItemMaster) {

return this.http.delete<ItemMaster[]>('http://localhost:64597/api/itemMaster/' + ut.itemId);

}

}

UNITSERVICE

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

import {Http, Response, Headers, RequestOptions, RequestMethod } from '@angular/http'

import { Observable} from 'rxjs/internal/observable';

import { map } from 'rxjs/operators';

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

import { unitMaster } from '../../model/master/unit.model';

@Injectable({

providedIn: 'root'

})

export class UnitService {

dataList:Observable<unitMaster[]>;

selectUnit : unitMaster;

constructor(private http : HttpClient) { }

loadData():Observable<unitMaster[]>{

const url='http://localhost:64597/api/unitMaster';

this.dataList=this.http.get<unitMaster[]>(url);

return this.http.get<unitMaster[]>(url);

}

postUnit(ut : unitMaster):Observable<unitMaster[]>{

this.selectUnit={

unitId:0,

unitName :'',

unitDescription:'',

}

const body: unitMaster={

unitId:ut.unitId,

unitName : ut.unitName,

unitDescription :ut.unitDescription,

}

return this.http.post<unitMaster[]>('http://localhost:64597/api/unitMaster', body);

}

getUnitDataList():Observable<unitMaster[]>{

return this.http.get<unitMaster[]>('http://localhost:64597/api/unitMaster');

}

Delete( ut : unitMaster){

return this.http.delete<unitMaster[]>('http://localhost:64597/api/unitMaster/'+ut.unitId );

//return null;

}

}