Table of Contents

[Models 3](#_Toc31895069)

[User 3](#_Toc31895070)

[users.ts : 3](#_Toc31895071)

[Company 3](#_Toc31895072)

[company.ts : 3](#_Toc31895073)

[Stock Exchange 4](#_Toc31895074)

[stockExchange.ts : 4](#_Toc31895075)

[Sector 4](#_Toc31895076)

[sector.ts : 4](#_Toc31895077)

[IPO 4](#_Toc31895078)

[ipo.ts: 4](#_Toc31895079)

[App Routing Module 6](#_Toc31895080)

[app-routing.module.ts: 6](#_Toc31895081)

[App Module 10](#_Toc31895082)

[app.module.ts: 10](#_Toc31895083)

[Components: 14](#_Toc31895084)

[App Component: 14](#_Toc31895085)

[app.component.html : 14](#_Toc31895086)

[app.component.ts: 14](#_Toc31895087)

[Common components 15](#_Toc31895088)

[Login Component 15](#_Toc31895089)

[Home Component 20](#_Toc31895090)

[Header Component 23](#_Toc31895091)

[Footer Component 25](#_Toc31895092)

[Landing Component 26](#_Toc31895093)

[Admin Components 27](#_Toc31895094)

[AdminHeader Component 27](#_Toc31895095)

[AdminHome Component 29](#_Toc31895096)

[ImportExcel Component 31](#_Toc31895097)

[ManageCompany Component 33](#_Toc31895098)

[ManageExchange Component 36](#_Toc31895099)

[ManageIPOs Component 39](#_Toc31895100)

[ManageSector Component 42](#_Toc31895101)

[NewCompany Component 45](#_Toc31895102)

[NewExchange Component 53](#_Toc31895103)

[NewIPO Component 57](#_Toc31895104)

[NewSector Company 61](#_Toc31895105)

[UpdateCompany Component 64](#_Toc31895106)

[UpdateExchange Component 72](#_Toc31895107)

[UpdateIPO Component 76](#_Toc31895108)

[UpdateSector Component 80](#_Toc31895109)

[ViewUsers Component 83](#_Toc31895110)

[User Components 85](#_Toc31895111)

[Account Component 85](#_Toc31895112)

[AccountUpdate Component 87](#_Toc31895113)

[Compare Component 90](#_Toc31895114)

[OTP Component 93](#_Toc31895115)

[Signup Component 94](#_Toc31895116)

[UserHeader Component 100](#_Toc31895117)

[UserHome Component 102](#_Toc31895118)

[ViewCompanies Component 106](#_Toc31895119)

[ViewExchanges Components 108](#_Toc31895120)

[ViewIPO Component 109](#_Toc31895121)

[ViewSectors Component 111](#_Toc31895122)

[Services 114](#_Toc31895123)

[User Service 114](#_Toc31895124)

[user.service.ts 114](#_Toc31895125)

[Companies Service 115](#_Toc31895126)

[companies.service.ts 115](#_Toc31895127)

[IPOs Service 116](#_Toc31895128)

[ipos.service.ts 116](#_Toc31895129)

[Sectors Service 118](#_Toc31895130)

[sectors.service.ts 118](#_Toc31895131)

[StockExchanges Service 119](#_Toc31895132)

[stock-exchanges.service.ts 119](#_Toc31895133)

[Guards 121](#_Toc31895134)

[Login Guard 121](#_Toc31895135)

[login.guard.ts 121](#_Toc31895136)

[AdminAuth Guard 122](#_Toc31895137)

[admin-auth.guard.ts 122](#_Toc31895138)

[UserAuth Guard 123](#_Toc31895139)

[user-auth.guard.ts 123](#_Toc31895140)

[NoLogin Guard (Does not allow loggin users to acces some pages) 124](#_Toc31895141)

[no-login.guard.ts 124](#_Toc31895142)

[styles.css 126](#_Toc31895143)

[json files 130](#_Toc31895144)

[companies.json 130](#_Toc31895145)

[ipos.json 130](#_Toc31895146)

[sectors.json 130](#_Toc31895147)

[stockexchanges.json 130](#_Toc31895148)

[users.json 130](#_Toc31895149)

# Models

## User

### users.ts :

export class User {

id: number;

username: string;

email: string;

password: string;

rePassword: string;

phoneNo: number;

isAdmin: boolean;

confirmed: boolean;

}

## Company

### company.ts :

export class Company {

id: number;

name: string;

sector: string;

ceo: string;

directors: string[];

stockExchanges: StockExchange[];

turnover: number;

brief: string;

}

class StockExchange {

stockExchange: string;

stockCode: string;

}

## Stock Exchange

### stockExchange.ts :

export class StockExchange {

id: number;

name: string;

brief: string;

contactAddress: string;

remarks: string;

}

## Sector

### sector.ts :

export class Sector {

id: number;

name: string;

brief: string;

}

## IPO

### ipo.ts:

export class IPO {

id: number;

companyName: string;

stockExchange: string;

pricePerShare: number;

totalShares: number;

openDateTime: Date;

remarks: string;

}

# App Routing Module

## app-routing.module.ts:

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

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

import { SignupComponent } from "./user/signup/signup.component";

import { HomeComponent } from "./common/home/home.component";

import { AdminHomeComponent } from "./admin/admin-home/admin-home.component";

import { UserHomeComponent } from "./user/user-home/user-home.component";

import { ImportExcelComponent } from "./admin/import-excel/import-excel.component";

import { NewCompanyComponent } from "./admin/new-company/new-company.component";

import { NewExchangeComponent } from "./admin/new-exchange/new-exchange.component";

import { NewIpoComponent } from "./admin/new-ipo/new-ipo.component";

import { OtpComponent } from "./user/otp/otp.component";

import { ManageExchangeComponent } from "./admin/manage-exchange/manage-exchange.component";

import { ManageCompanyComponent } from "./admin/manage-company/manage-company.component";

import { UpdateCompanyComponent } from "./admin/update-company/update-company.component";

import { ViewUsersComponent } from "./admin/view-users/view-users.component";

import { AccountComponent } from "./user/account/account.component";

import { AccountUpdateComponent } from "./user/account-update/account-update.component";

import { UpdateExchangeComponent } from "./admin/update-exchange/update-exchange.component";

import { ManageIposComponent } from "./admin/manage-ipos/manage-ipos.component";

import { UpdateIpoComponent } from "./admin/update-ipo/update-ipo.component";

import { NewSectorComponent } from "./admin/new-sector/new-sector.component";

import { ManageSectorsComponent } from "./admin/manage-sectors/manage-sectors.component";

import { UpdateSectorComponent } from "./admin/update-sector/update-sector.component";

import { CompareComponent } from "./user/compare/compare.component";

import { ViewCompaniesComponent } from "./user/view-companies/view-companies.component";

import { ViewExchangesComponent } from "./user/view-exchanges/view-exchanges.component";

import { ViewSectorsComponent } from "./user/view-sectors/view-sectors.component";

import { LandingComponent } from './common/landing/landing.component';

import { ViewIpoComponent } from './user/view-ipo/view-ipo.component';

import { AdminAuthGuard } from './guards/admin-auth.guard';

import { UserAuthGuard } from './guards/user-auth.guard';

import { NoLoginGuard } from './guards/no-login.guard';

import { LoginGuard } from './guards/login.guard';

import { CompareResultComponent } from './user/compare-result/compare-result.component';

const routes: Routes = [

{ path: "", redirectTo: "home", pathMatch: "full" },

{ path: "home", component: HomeComponent, canActivate: [NoLoginGuard] },

{ path: "signup", component: SignupComponent, canActivate: [NoLoginGuard] },

{ path: "admin-home", component: AdminHomeComponent,canActivate:[AdminAuthGuard] },

{ path: "user-home", component: UserHomeComponent, canActivate: [UserAuthGuard] },

{ path: "import-excel", component: ImportExcelComponent,canActivate:[AdminAuthGuard] },

{ path: "new-company", component: NewCompanyComponent,canActivate:[AdminAuthGuard] },

{ path: "new-exchange", component: NewExchangeComponent,canActivate:[AdminAuthGuard] },

{ path: "new-ipo", component: NewIpoComponent,canActivate:[AdminAuthGuard] },

{ path: "new-sector", component: NewSectorComponent,canActivate:[AdminAuthGuard] },

{ path: "otp", component: OtpComponent, canActivate: [NoLoginGuard] },

{ path: "manage-exchange", component: ManageExchangeComponent,canActivate:[AdminAuthGuard]},

{ path: "manage-company", component: ManageCompanyComponent,canActivate:[AdminAuthGuard]},

{ path: "manage-ipos", component: ManageIposComponent,canActivate:[AdminAuthGuard] },

{ path: "manage-sectors", component: ManageSectorsComponent,canActivate:[AdminAuthGuard] },

{ path: "update-ipo", component: UpdateIpoComponent,canActivate:[AdminAuthGuard] },

{ path: "update-company", component: UpdateCompanyComponent,canActivate:[AdminAuthGuard] },

{ path: "update-exchange", component: UpdateExchangeComponent,canActivate:[AdminAuthGuard] },

{ path: "update-sector", component: UpdateSectorComponent,canActivate:[AdminAuthGuard] },

{ path: "view-users", component: ViewUsersComponent,canActivate:[AdminAuthGuard] },

{ path: "view-companies", component: ViewCompaniesComponent, canActivate: [UserAuthGuard] },

{ path: "view-exchanges", component: ViewExchangesComponent, canActivate: [UserAuthGuard] },

{ path: "view-sectors", component: ViewSectorsComponent, canActivate: [UserAuthGuard] },

{ path: "view-ipos", component: ViewIpoComponent, canActivate: [UserAuthGuard] },

{ path: "account", component: AccountComponent, canActivate: [UserAuthGuard] },

{ path: "account-update", component: AccountUpdateComponent , canActivate: [UserAuthGuard]},

{ path: "compare", component: CompareComponent, canActivate: [UserAuthGuard] },

{ path: "compare-result", component: CompareResultComponent, canActivate: [UserAuthGuard] },

{ path: "landing", component: LandingComponent, canActivate:[LoginGuard]},

];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule {}

# App Module

## app.module.ts:

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

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

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

import { FontAwesomeModule } from "@fortawesome/angular-fontawesome";

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

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

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

import { SignupComponent } from "./user/signup/signup.component";

import { HomeComponent } from "./common/home/home.component";

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

import { NewCompanyComponent } from "./admin/new-company/new-company.component";

import { NewExchangeComponent } from "./admin/new-exchange/new-exchange.component";

import { ImportExcelComponent } from "./admin/import-excel/import-excel.component";

import { NewIpoComponent } from "./admin/new-ipo/new-ipo.component";

import { AccountUpdateComponent } from "./user/account-update/account-update.component";

import { AdminHomeComponent } from "./admin/admin-home/admin-home.component";

import { UserHomeComponent } from "./user/user-home/user-home.component";

import { OtpComponent } from "./user/otp/otp.component";

import { ManageCompanyComponent } from "./admin/manage-company/manage-company.component";

import { ManageExchangeComponent } from "./admin/manage-exchange/manage-exchange.component";

import { UpdateCompanyComponent } from "./admin/update-company/update-company.component";

import { ViewUsersComponent } from "./admin/view-users/view-users.component";

import { AccountComponent } from "./user/account/account.component";

import { UpdateExchangeComponent } from "./admin/update-exchange/update-exchange.component";

import { ManageIposComponent } from "./admin/manage-ipos/manage-ipos.component";

import { UpdateIpoComponent } from './admin/update-ipo/update-ipo.component';

import { ManageSectorsComponent } from './admin/manage-sectors/manage-sectors.component';

import { NewSectorComponent } from './admin/new-sector/new-sector.component';

import { UpdateSectorComponent } from './admin/update-sector/update-sector.component';

import { CompareComponent } from './user/compare/compare.component';

import { ViewCompaniesComponent } from './user/view-companies/view-companies.component';

import { ViewExchangesComponent } from './user/view-exchanges/view-exchanges.component';

import { ViewSectorsComponent } from './user/view-sectors/view-sectors.component';

import { HeaderComponent } from './common/header/header.component';

import { FooterComponent } from './common/footer/footer.component';

import { UserHeaderComponent } from './user/user-header/user-header.component';

import { AdminHeaderComponent } from './admin/admin-header/admin-header.component';

import { LandingComponent } from './common/landing/landing.component';

import { ViewIpoComponent } from './user/view-ipo/view-ipo.component';

import { CompareResultComponent } from './user/compare-result/compare-result.component';

@NgModule({

declarations: [

AppComponent,

SignupComponent,

HomeComponent,

LoginComponent,

NewCompanyComponent,

NewExchangeComponent,

ImportExcelComponent,

NewIpoComponent,

AccountUpdateComponent,

AdminHomeComponent,

UserHomeComponent,

OtpComponent,

ManageCompanyComponent,

ManageExchangeComponent,

UpdateCompanyComponent,

ViewUsersComponent,

AccountComponent,

UpdateExchangeComponent,

ManageIposComponent,

UpdateIpoComponent,

ManageSectorsComponent,

NewSectorComponent,

UpdateSectorComponent,

CompareComponent,

ViewCompaniesComponent,

ViewExchangesComponent,

ViewSectorsComponent,

HeaderComponent,

FooterComponent,

UserHeaderComponent,

AdminHeaderComponent,

LandingComponent,

ViewIpoComponent,

CompareResultComponent

],

imports: [

BrowserModule,

AppRoutingModule,

ReactiveFormsModule,

FontAwesomeModule,

HttpClientModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule {}

# Components:

## App Component:

### app.component.html :

<app-header \*ngIf="commonHeader"></app-header>

<app-admin-header \*ngIf="adminHeader"></app-admin-header >

<app-user-header \*ngIf="userHeader"></app-user-header>

<router-outlet></router-outlet>

<app-footer></app-footer>

<app-login class="modal fade" id="login-modal"></app-login>

### app.component.ts:

import { Component, OnInit, AfterViewInit, DoCheck } from "@angular/core";

@Component({

selector: "app-root",

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

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

})

export class AppComponent implements DoCheck{

title = "StockView";

commonHeader: boolean = true;

userHeader: boolean = false;

adminHeader: boolean = false;

ngDoCheck(){

if(localStorage.getItem("userType") == "admin" || sessionStorage.getItem("userType") == "admin"){

this.commonHeader = false;

this.adminHeader = true;

this.userHeader = false;

}

else if(localStorage.getItem("userType") == "user" || sessionStorage.getItem("userType") == "user"){

this.commonHeader = false;

this.adminHeader = false;

this.userHeader = true;

}

else if(localStorage.getItem("userType") == null && sessionStorage.getItem("userType") == null){

this.commonHeader = true;

this.adminHeader = false;

this.userHeader = false;

}

}

}

## Common components

### Login Component

#### login.component.html

<div class="modal-dialog modal-dialog-centered" role="login">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Login</h5>

<button class="close" data-dismiss="modal">

<span>&times;</span>

</button>

</div>

<div class="modal-body">

<div class="alert alert-danger alert-dismissible fade show" [hidden]="isValid">

Invalid username or password.

</div>

<form [formGroup]="loginForm" (ngSubmit)="login()">

<div class="form-group mt-4">

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

<div class="input-group">

<div class="input-group-prepend">

<span class="input-group-text">

<fa-icon [icon]="faAt"></fa-icon>

</span>

</div>

<input type="text" name="username" id="username" class="form-control" autofocus autocomplete

formControlName="username">

</div>

</div>

<div class="form-group">

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

<div class="input-group">

<div class="input-group-prepend">

<span class="input-group-text">

<fa-icon [icon]="faKey"></fa-icon>

</span>

</div>

<input type="password" name="password" id="password" class="form-control"

formControlName="password">

</div>

</div>

<div class="custom-control custom-checkbox">

<input type="checkbox" name="rememberMe" id="rememberMe" class="custom-control-input"

formControlName="rememberMe">

<label for="rememberMe" class="custom-control-label">Remember me</label>

</div>

<button type="submit" class="btn btn-primary btn-block mt-5"[disabled] = "!loginForm.valid || disableButton">Login</button>

<span class="row text-muted justify-content-center my-2"><small>New user?<a [routerLink]="['signup']">

Signup</a></small></span>

</form>

</div>

</div>

</div>

#### login.component.ts

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

import {

FormControl,

FormGroup,

FormBuilder,

Validators

} from "@angular/forms";

declare var $: any;

import { faAt, faKey } from "@fortawesome/free-solid-svg-icons";

import { UserService } from "../services/user.service";

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

import { User } from "../models/users";

@Component({

selector: "app-login",

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

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

})

export class LoginComponent implements OnInit {

loginForm: FormGroup;

disableButton:boolean;

isValid: boolean = true;

faAt = faAt;

faKey = faKey;

users: User[];

constructor(

private formBuilder: FormBuilder,

private userService: UserService,

private router: Router

) { }

ngOnInit() {

this.disableButton = false;

this.loginForm = this.formBuilder.group({

username: ["", Validators.required],

password: ["", Validators.required],

rememberMe: [false]

});

this.userService.getAllUsers().subscribe(data => {

this.users = data;

});

}

login() {

this.disableButton = true;

this.isValid = true;

localStorage.removeItem("userId");

localStorage.removeItem("userType");

sessionStorage.removeItem("userId");

sessionStorage.removeItem("userType");

if (this.validateUser(this.loginForm.get("username").value,this.loginForm.get("password").value)) {

if (this.loginForm.get("rememberMe").value) {

localStorage.setItem("userId",this.getUserId(this.loginForm.get("username").value).toString());

localStorage.setItem("userType",this.getUserType(this.loginForm.get("username").value));

}

sessionStorage.setItem("userId",this.getUserId(this.loginForm.get("username").value).toString());

sessionStorage.setItem("userType",this.getUserType(this.loginForm.get("username").value));

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

$("#login-modal").modal("hide");

this.loginForm.reset();

this.disableButton = false;

} else {

this.isValid = false;

this.disableButton = false;

}

}

validateUser(username: string, password: string): boolean {

for (let u of this.users) {

if (u.username == username && u.password == password) {

return true;

}

}

return false;

}

getUserType(username: string): string {

for (let user of this.users) {

if (user.username == username) {

if (user.isAdmin) {

return "admin";

}

}

}

return "user";

}

getUserId(username: string): number {

for (let user of this.users) {

if (user.username == username) {

return user.id;

}

}

}

}

### Home Component

#### [home.component.html](http://home.component.html)

#### <div>

#### <section class="bg-transparent row justify-content-center undraggable">

#### <div class="container hero-text text-white">

#### <h1 class="display-4 text-center text-shadow">Stock Market Charting</h1>

#### <p class="d-none d-md-block lead text-justify mt-5 text-shadow">

#### Lorem ipsum, dolor sit amet consectetur adipisicing elit. Non nobis esse, aliquam qui, blanditiis omnis

#### perferendis commodi placeat quidem libero ab voluptates error est consequuntur illo facere voluptas

#### deleniti quo!

#### </p>

#### </div>

#### </section>

<div class="carousel slide bg-carousel" data-ride="carousel">

<div class="carousel-inner">

<div class="carousel-item active">

<img class="d-block w-100" src="../assets/carousel/aditya-vyas-6Ih4UoqzaAs-unsplash.jpg"

alt="First slide">

</div>

<div class="carousel-item">

<img class="d-block w-100" src="../assets/carousel/chart-1905225\_1920.jpg" alt="Second slide">

</div>

<div class="carousel-item">

<img class="d-block w-100" src="../assets/carousel/erol-ahmed-9ZWtkOCMxbM-unsplash.jpg"

alt="Third slide">

</div>

</div>

</div>

<main class="container mt-5">

<div class="row card-deck justify-content-between">

<div class="card w-100 shadow hover-enlarge">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../assets/stock.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">Charts</h4>

<p class="card-text text-center mb-3">See the stock charts of companies and exchanges</p>

</div>

</div>

<div class="card w-100 shadow hover-enlarge">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../assets/scale-compare.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">Compare</h4>

<p class="card-text text-center mb-3">Compare the stocks of Sectors and Companies over a period of

time.</p>

</div>

</div>

<div class="card w-100 shadow hover-enlarge">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../assets/ipo.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">IPOs</h4>

<p class="card-text text-center mb-3">Get to know about the new IPOs.</p>

</div>

</div>

</div>

<div class="row mt-5 justify-content-between card-deck">

<div class="card col-sm-6 shadow">

<div class="card-body">

<h4 class="card-title text-center">Already a User?</h4>

<h5 class="card-subtitle text-center text-muted">Continue</h5>

<button class="btn btn-primary btn-block mt-3" data-toggle="modal"

data-target="#login-modal">Login</button>

</div>

</div>

<div class="card col-sm-6 shadow">

<div class="card-body">

<h4 class="card-title text-center">New User?</h4>

<h5 class="card-subtitle text-center text-muted">Get startrd</h5>

<button class="btn btn-primary btn-block mt-3" (click)="gotoSignup()">Signup</button>

</div>

</div>

</div>

</main>

</div>

#### [home.component.ts](http://home.component.ts):

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

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

declare var $: any;

@Component({

selector: "app-home",

templateUrl: "./home.component.html",

styleUrls: ["./home.component.css"]

})

export class HomeComponent implements OnInit {

constructor(private router: Router) {}

ngOnInit() {

$('.carousel').carousel({

interval: 3000

});

}

gotoSignup(){

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

}

}

### Header Component

#### header.component.html

<header class="">

<nav class="navbar navbar-expand-md navbar-dark bg-dark sticky-top">

<a routerLink="home" class="navbar-brand">

<img src="../assets/logo.svg" alt="StockView" id="logo" class="undraggable">

<span class="text-light mx-2 my-auto undraggable">StockView</span>

</a>

<button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#navbar1">

<span class="navbar-toggler-icon"></span></button>

<div class="navbar-collapse collapse" id="navbar1">

<ul class="navbar-nav ml-auto">

<li class="nav-item"><a routerLink="home" routerLinkActive="active" class="nav-link">Home</a></li>

<li class="nav-item"><a routerLink="signup" routerLinkActive="active" class="nav-link">Signup</a></li>

</ul>

<a class="btn text-light ml-md-5 pl-0 pl-md-5" role="login" data-toggle="modal"

data-target="#login-modal">Login</a>

</div>

</nav>

</header>

#### header.component.ts

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

@Component({

selector: 'app-header',

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

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

})

export class HeaderComponent implements OnInit {

constructor() { }

ngOnInit() {

}

}

### Footer Component

#### footer.component.html

<footer class="container-fluid bg-light mt-5">

<div class="row bg-dark socials-footer undraggable py-1">

<div class="col-md-6 align-self-center text-muted text-md-left text-center">

<span>StockView &#169; Copyright 2020-2021</span>

</div>

<div class="col-md-4 offset-md-2">

<div class="row justify-content-end flex-nowrap">

<a href="http://facebook.com" class="col" target="\_new">

<fa-icon [icon]="faFacebook" class="h-100 py-2 social-icon text-light mx-4"></fa-icon>

</a>

<a href="http://twitter.com" class="col" target="\_new">

<fa-icon [icon]="faTwitter" class="h-100 py-2 social-icon text-light mx-4"></fa-icon>

</a>

<a href="http://instagram.com" class="col" target="\_new">

<fa-icon [icon]="faInstagram" class="h-100 py-2 social-icon text-light mx-4"></fa-icon>

</a>

<a href="https://github.com/abdussaboor98/StockView-StockMarketCharting" class="col" target="\_new">

<fa-icon [icon]="faGithub" class="h-100 py-2 social-icon text-light mx-4"></fa-icon>

</a>

</div>

</div>

</div>

</footer>

#### footer.component.ts

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

import { faTwitter,faInstagram,faFacebook,faGithub } from '@fortawesome/free-brands-svg-icons';

@Component({

selector: 'app-footer',

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

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

})

export class FooterComponent implements OnInit {

faFacebook = faFacebook

faTwitter = faTwitter

faInstagram = faInstagram

faGithub = faGithub

constructor() { }

ngOnInit() {

}

}

### Landing Component

#### landing.component.html

<app-user-home \*ngIf="!isAdmin"></app-user-home>

<app-admin-home \*ngIf="isAdmin"></app-admin-home>

#### landing.component.ts

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

import { UserService } from 'src/app/services/user.service';

@Component({

selector: 'app-landing',

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

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

})

export class LandingComponent implements OnInit {

isAdmin: boolean;

constructor(private userService: UserService) {

this.isAdmin = userService.isAdmin();

}

ngOnInit() {

}

}

## Admin Components

### AdminHeader Component

#### admin-header.component.html

<header>

<nav class="navbar navbar-expand-md navbar-dark bg-dark sticky-top">

<a routerLink="/landing" class="navbar-brand">

<img src="../../../assets/logo.svg" alt="StockView" id="logo" class="undraggable">

<span class="text-light mx-2 my-auto undraggable">StockView</span>

</a>

<button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#navbar1">

<span class="navbar-toggler-icon"></span></button>

<div class="navbar-collapse collapse" id="navbar1">

<ul class="navbar-nav ml-auto">

<li class="nav-item"><a routerLink="/landing" class="nav-link active">Home</a></li>

<li class="nav-item"><a routerLink="/import-excel" class="nav-link">Import Data</a></li>

<li class="nav-item"><a routerLink="/manage-company" class="nav-link">Manage Companies</a></li>

<li class="nav-item"><a routerLink="/manage-exchange" class="nav-link">Manage Exchange</a></li>

<li class="nav-item"><a routerLink="/manage-sectors" class="nav-link">Manage Sectors</a></li>

<li class="nav-item"><a routerLink="/manage-ipos" class="nav-link">Manage IPOs</a></li>

<li class="nav-item"><a routerLink="/view-users" class="nav-link">View Users</a></li>

</ul>

<button class=" btn text-light ml-md-5 pl-0 pl-md-5" role="logout" (click)="logout()">Logout</button>

</div>

</nav>

</header>

#### admin-header.component.ts

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

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

@Component({

selector: 'app-admin-header',

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

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

})

export class AdminHeaderComponent implements OnInit {

constructor(private router: Router) { }

ngOnInit() {

}

logout(){

console.log('logging out admin');

localStorage.removeItem("userType");

sessionStorage.removeItem("userType");

localStorage.removeItem("userId");

sessionStorage.removeItem("userId");

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

}

}

### AdminHome Component

#### admin-home.component.html

<section class="jumbotron jumbotron-fluid py-4">

<div class="container">

<h1 class="display-4 text-center">Hi! Admin,<br>

<p class="lead text-center mt-5">What would you like to do?</p>

</h1>

</div>

</section>

<main class="container">

<div class="row card-deck">

<a class="card w-100 shadow hover-enlarge mt-4 mx-4 mx-sm-3" routerLink="/import-excel">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../../../assets/import-excel.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">Import Data</h4>

<p class="card-text text-center mb-3">Upload stock data from excel sheet. You can upload for multiple

days.</p>

</div>

</a>

<a class="card w-100 shadow hover-enlarge mt-4 mx-4 mx-sm-3" routerLink="/manage-company">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../../../assets/manage-company.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">Manage Companies</h4>

<p class="card-text text-center mb-3">View the available companies and add new companies.</p>

</div>

</a>

</div>

<div class="row card-deck">

<a class="card w-100 shadow hover-enlarge mt-4 mx-4 mx-sm-3" routerLink="/manage-exchange">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../../../assets/manage-exchange.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">Manage Exchanges</h4>

<p class="card-text text-center mb-3">View the available exchanges and add new exchanges.</p>

</div>

</a>

<a class="card w-100 shadow hover-enlarge mt-4 mx-4 mx-sm-3" routerLink="/manage-ipos">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../../../assets/manage-ipo.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">Manage IPO</h4>

<p class="card-text text-center mb-3">Add new companies and updates their IPO information.</p>

</div>

</a>

</div>

</main>

#### admin-home.component.ts

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

@Component({

selector: "app-admin-home",

templateUrl: "./admin-home.component.html",

styleUrls: ["./admin-home.component.css"]

})

export class AdminHomeComponent implements OnInit {

constructor() {}

ngOnInit() {}

}

### ImportExcel Component

#### impot-excel.component.html

<main class="container">

<div class="display-4 text-center my-4">Import Data</div>

<form class="row justify-content-center" [formGroup]="excelUploadForm" (ngSubmit)="uploadData()">

<div class="card col-md-8 col-sm-10 col-11 mx-auto">

<div class="card-body">

<span class="lead">Select excel file to be uploaded: </span>

<div class="custom-file my-3">

<input type="file" name="excelFileUpload" id="excelFileUpload" formControlName="excelFileUpload"

class="custom-file-input">

<label for="excelFileUpload" class="custom-file-label"></label>

</div>

<button type="submit" class="btn btn-primary btn-block">Upload</button>

</div>

</div>

</form>

<div class="row justify-content-center my-2">

<div class="alert alert-info col-md-8 col-sm-10 col-11 mx-auto">

Download sample excel file <a hrefs="#" class="alert-link">here</a>.

<button type="button" class="close" data-dismiss="alert">

<span aria-hidden="true">&times;</span>

</button>

</div>

</div>

</main>

#### impot-excel.component.html

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

import { FormGroup, FormControl } from "@angular/forms";

import bsCustomFileInput from "bs-custom-file-input";

@Component({

selector: "app-import-excel",

templateUrl: "./import-excel.component.html",

styleUrls: ["./import-excel.component.css"]

})

export class ImportExcelComponent implements OnInit {

excelUploadForm: FormGroup;

constructor() {}

ngOnInit() {

bsCustomFileInput.init();

this.excelUploadForm = new FormGroup({

excelFileUpload: new FormControl("")

});

}

uploadData() {

console.log(this.excelUploadForm.value);

}

}

### ManageCompany Component

#### manage-company.component.html

<main class="container">

<div class="display-4 text-center my-4">Manage Companies</div>

<div class="row justify-content-center">

<div class="table-responsive">

<table class="table table-stripped table-hover text-center bg-light">

<thead class="thead-light">

<th>ID</th>

<th>Name</th>

<th>Sector</th>

<th>CEO</th>

<th>Board of Directors</th>

<th>Stock Exchanges</th>

<th>Turnover</th>

<th>Brief</th>

<th colspan="2" class="text-center">Actions</th>

</thead>

<tbody>

<tr \*ngFor="let company of companies">

<td> {{company.id}} </td>

<td> {{company.name}} </td>

<td> {{company.sector}} </td>

<td> {{company.ceo}} </td>

<td>

<div \*ngFor="let director of company.directors" class="row justify-content-center">

{{director}}

</div>

</td>

<td>

<div \*ngFor="let exchange of company.stockExchanges" class="row justify-content-center">

{{exchange.stockExchange}} : {{exchange.stockCode}}

</div>

</td>

<td> {{company.turnover}} </td>

<td> {{company.brief}} </td>

<td><button type="button" class="btn btn-info" (click)="updateCompany(company.id)">

<fa-icon [icon]="faEdit"></fa-icon>

</button>

</td>

<td><button type="button" class="btn btn-danger" (click)="deleteCompany(company.id)">

<fa-icon [icon]="faTrash"></fa-icon>

</button>

</td>

<!-- <td><button type="button" class="btn btn-primary">

<fa-icon [icon]="faEye"></fa-icon>

</button>

</td> -->

</tr>

</tbody>

</table>

</div>

</div>

<button type="button" class="btn btn-primary rounded-circle mx-auto add-btn shadow" (click)="gotoNew()">

<fa-icon [icon]="faPlus" class="h2"></fa-icon>

</button>

</main>

#### manage-company.component.ts

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

import { CompaniesService } from "src/app/services/companies.service";

import { Company } from "src/app/models/company";

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

import {

faPlus,

faEye,

faTrash,

faEdit

} from "@fortawesome/free-solid-svg-icons";

@Component({

selector: "app-manage-company",

templateUrl: "./manage-company.component.html",

styleUrls: ["./manage-company.component.css"]

})

export class ManageCompanyComponent implements OnInit {

faPlus = faPlus;

faEye = faEye;

faTrash = faTrash;

faEdit = faEdit;

companies: Company[];

constructor(

private companiesService: CompaniesService,

private router: Router

) {}

ngOnInit() {

this.companiesService.getAllCompanies().subscribe(data => {

this.companies = data;

});

}

gotoNew() {

this.router.navigate(["new-company"]);

}

deleteCompany(id: number) {

this.companiesService.deleteCompany(id).subscribe(data => {

this.companies = this.companies.filter(

company => company.id !== id

);

});

}

updateCompany(id: number) {

localStorage.removeItem("companyId");

localStorage.setItem("companyId", id.toString());

this.router.navigate(["update-company"]);

}

}

### ManageExchange Component

#### manage-exchange.component.html

<main class="container">

<div class="display-4 text-center my-4">Manage Exchanges</div>

<div class="table-responsive">

<table class="table table-stripped table-hover text-center bg-light rounded">

<thead class="thead-light">

<tr>

<th>ID</th>

<th>Name</th>

<th>Brief</th>

<th>Contact Address</th>

<th>Remarks</th>

<th colspan="2" class="text-center">Actions</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let stockEx of stockExchanges">

<td> {{stockEx.id}} </td>

<td> {{stockEx.name}} </td>

<td> {{stockEx.brief}} </td>

<td> {{stockEx.contactAddress}} </td>

<td> {{stockEx.remarks}} </td>

<td> <button class="btn btn-info" (click)="updateStockExchange(stockEx.id)">

<fa-icon [icon]="faEdit"></fa-icon>

</button> </td>

<td> <button class="btn btn-danger" (click)="deleteStockExchange(stockEx.id)">

<fa-icon [icon]="faTrash"></fa-icon>

</button> </td>

</tr>

</tbody>

</table>

</div>

<button type="button" class="btn btn-primary rounded-circle mx-auto add-btn" (click)="gotoNew()">

<fa-icon [icon]="faPlus" class="pt-2 h2"></fa-icon>

</button>

</main>

#### manage-company.component.ts

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

import { StockExchangesService } from "src/app/services/stock-exchanges.service";

import { StockExchange } from "src/app/models/stockExchange";

import {

faPlus,

faTrash,

faEdit,

faEye

} from "@fortawesome/free-solid-svg-icons";

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

@Component({

selector: "app-manage-exchange",

templateUrl: "./manage-exchange.component.html",

styleUrls: ["./manage-exchange.component.css"]

})

export class ManageExchangeComponent implements OnInit {

faPlus = faPlus;

faEye = faEye;

faTrash = faTrash;

faEdit = faEdit;

stockExchanges: StockExchange[];

constructor(

private stockExService: StockExchangesService,

private router: Router

) {}

ngOnInit() {

this.stockExService.getAllExchanges().subscribe(data => {

this.stockExchanges = data;

console.log(this.stockExchanges);

});

}

gotoNew(){

this.router.navigate(['new-exchange'])

}

updateStockExchange(id) {

localStorage.removeItem("stockExId");

localStorage.setItem("stockExId", id);

this.router.navigate(["update-exchange"]);

}

deleteStockExchange(id) {

this.stockExService.deleteStockExchange(id).subscribe(data => {

this.stockExchanges = this.stockExchanges.filter(

exchange => exchange.id !== id

);

});

}

}

### ManageIPOs Component

#### manage-ipos.component.html

<main class="container">

<div class="display-4 text-center my-4">Manage IPOs</div>

<div class="table-responsive">

<table class="table table-stripped table-hover text-center bg-light">

<thead class="thead-light">

<tr>

<th>ID</th>

<th>Company name</th>

<th>Stock Exchange</th>

<th>Price per share</th>

<th>Total shares</th>

<th>Open Date/Time</th>

<th>Remarks</th>

<th colspan="2" class="text-center">Actions</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let ipo of ipos">

<td> {{ipo.id}} </td>

<td> {{ipo.companyName}} </td>

<td> {{ipo.stockExchange}} </td>

<td> {{ipo.pricePerShare | currency: "INR"}} </td>

<td> {{ipo.totalShares}} </td>

<td> {{ipo.openDateTime | date: "dd/MM/yyyy hh:mm a"}} </td>

<td> {{ipo.remarks}} </td>

<td> <button class="btn btn-info" (click)="updateIPO(ipo.id)">

<fa-icon [icon]="faEdit"></fa-icon>

</button> </td>

<td> <button class="btn btn-danger" (click)="deleteIPO(ipo.id)">

<fa-icon [icon]="faTrash"></fa-icon>

</button> </td>

</tr>

</tbody>

</table>

</div>

<button type="button" class="btn btn-primary rounded-circle mx-auto add-btn shadow" (click)="gotoNew()">

<fa-icon [icon]="faPlus" class="h2"></fa-icon>

</button>

</main>

#### manage-ipos.component.ts

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

import {

faPlus,

faTrash,

faEye,

faEdit

} from "@fortawesome/free-solid-svg-icons";

import { IposService } from 'src/app/services/ipos.service';

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

import { IPO } from 'src/app/models/ipo';

@Component({

selector: "app-manage-ipos",

templateUrl: "./manage-ipos.component.html",

styleUrls: ["./manage-ipos.component.css"]

})

export class ManageIposComponent implements OnInit {

faPlus = faPlus;

faEye = faEye;

faTrash = faTrash;

faEdit = faEdit;

ipos: IPO[];

constructor(private iposService: IposService, private router: Router) {}

ngOnInit() {

this.iposService.getAllIPOs().subscribe(data => {

this.ipos = data;

console.log(data);

})

}

gotoNew() {

this.router.navigate(["new-ipo"]);

}

deleteIPO(id: number){

this.iposService.deleteIPO(id).subscribe(data => {

this.ipos = this.ipos.filter(ipo => ipo.id !== id)

})

}

updateIPO(id: number){

sessionStorage.removeItem('ipoId');

sessionStorage.setItem('ipoId',id.toString());

this.router.navigate(['update-ipo']);

}

}

### ManageSector Component

#### manage-sector.component.html

<main class="container">

<div class="display-4 text-center my-4">Manage IPOs</div>

<div class="table-responsive">

<table class="table table-stripped table-hover bg-light text-center">

<thead class="thead-light">

<tr>

<th>ID</th>

<th>Sector name</th>

<th>Brief</th>

<th colspan="2" class="text-center">Actions</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let sector of sectors">

<td> {{sector.id}} </td>

<td> {{sector.name}} </td>

<td> {{sector.brief}} </td> <td> <button class="btn btn-info" (click)="updateSector(sector.id)">

<fa-icon [icon]="faEdit"></fa-icon>

</button> </td>

<td> <button class="btn btn-danger" (click)="deleteSector(sector.id)">

<fa-icon [icon]="faTrash"></fa-icon>

</button> </td>

</tr>

</tbody>

</table>

</div>

<button type="button" class="btn btn-primary rounded-circle mx-auto add-btn" (click)="gotoNew()">

<fa-icon [icon]="faPlus" class="h2"></fa-icon>

</button>

</main>

#### manage-sector.component.ts

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

import { Sector } from "src/app/models/sector";

import { SectorsService } from "src/app/services/sectors.service";

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

import { faPlus, faTrash, faEdit } from "@fortawesome/free-solid-svg-icons";

@Component({

selector: "app-manage-sectors",

templateUrl: "./manage-sectors.component.html",

styleUrls: ["./manage-sectors.component.css"]

})

export class ManageSectorsComponent implements OnInit {

faPlus = faPlus;

faTrash = faTrash;

faEdit = faEdit;

sectors: Sector[];

constructor(

private sectorsService: SectorsService,

private router: Router

) {}

ngOnInit() {

this.sectorsService.getAllSectors().subscribe(data => {

this.sectors = data;

});

}

gotoNew() {

this.router.navigate(["new-sector"]);

}

deleteSector(id: number) {

this.sectorsService.deleteSector(id).subscribe(data => {

this.sectors = this.sectors.filter(sector => sector.id !== id);

});

}

updateSector(id: number) {

sessionStorage.removeItem("sectorId");

sessionStorage.setItem("sectorId", id.toString());

this.router.navigate(["update-sector"]);

}

}

### NewCompany Component

#### new-company.component.html

<main class="container">

<div class="display-4 text-center my-4">Add New Company</div>

<div class="row justify-content-center">

<div class="card col-md-8 col-sm-10 col-11 mx-auto shadow">

<div class="card-body">

<form [formGroup]="newCompanyForm" (ngSubmit)="onSubmit()">

<div class="row">

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

<img class="w-100 px-2" src="../../../assets/ph.svg" alt="">

</div>

<div class="col-8">

<span>Upload logo: </span>

<div class="custom-file my-3">

<input type="file" name="logoUpload" id="logoUpload">

<label for="logoUpload" class="custom-file-label"><span class="limit-text">Upload

logo...</span></label>

</div>

</div>

</div>

<div class="form-group">

<label for="companyName" class="form-control-label">Name:</label>

<div class="input-group">

<input type="text" name="companyName" id="companyName" formControlName="name"

class="form-control" placeholder="Enter company name..." required>

</div>

</div>

<div class="form-group">

<label for="sector">Select sector:</label>

<div class="input-group">

<select name="sector" id="sector" class="custom-select" formControlName="sector" required >

<option value="none" selected disabled>Select sector...</option>

<option \*ngFor="let sector of sectors" [value]="sector.name">{{sector.name}}</option>

</select>

</div>

</div>

<div class="form-group">

<label for="ceoName" class="form-control-label">CEO Name:</label>

<div class="input-group">

<input type="text" name="ceoName" id="ceoName" formControlName="ceo" class="form-control"

placeholder="Enter the CEO name..." required>

</div>

</div>

<div class="form-group" formArrayName="directors">

<div class="row px-3">

<label for="directors" class="form-control-label">Board of Directors:</label>

<button type="button" class="btn btn-info ml-auto btn-sm mb-2" role="add\_director"

(click)="addDirector()">Add

Director</button>

</div>

<div class="input-group mb-2"

\*ngFor="let directorControl of newCompanyForm.get('directors').controls; let i = index">

<input type="text" [formControlName]="i" class="form-control "

placeholder="Enter the director name..." required>

<div class="input-group-append">

<button \*ngIf="i>0" class="btn btn-danger" (click)=(removeDirector(i))>

<fa-icon [icon]="faTrash"></fa-icon>

</button>

<button \*ngIf="i==0" class="btn btn-secondary" (click)=(removeDirector(i)) disabled>

<fa-icon [icon]="faTrash"></fa-icon>

</button>

</div>

</div>

</div>

<div class="form-group" formArrayName="stockExchanges">

<div class="row px-3">

<label for="stockExchange" class="form-control-label">Stock Exchanges(with stock

code):</label>

<button type="button" class="btn btn-info ml-auto btn-sm mb-2"

(click)="addStockExchange()">Add Exchange</button>

</div>

<div \*ngFor="let StockExG of newCompanyForm.get('stockExchanges').controls; let i = index"

class="mb-2">

<div class="input-group row mx-auto" [formGroupName]="i">

<select name="stockExchange" id="stockExchange" class="custom-select col-7" formControlName="stockExchange" required >

<option value="" selected disabled>Select stock exchange...</option>

<option \*ngFor="let stockExchange of stockExchanges" [value]="stockExchange.name">{{stockExchange.name}}</option>

</select>

<input type="text" name="stockCode" id="stockCode" class="form-control col-5"

placeholder="Enter the stock code..." required formControlName="stockCode">

<div class="input-group-append">

<button \*ngIf="i>0" class="btn btn-danger" (click)=(removeStockExchange(i))>

<fa-icon [icon]="faTrash"></fa-icon>

</button>

<button \*ngIf="i == 0" class="btn btn-secondary" disabled>

<fa-icon [icon]="faTrash"></fa-icon>

</button>

</div>

</div>

</div>

</div>

<div class="form-group">

<label for="turnover" class="form-control-label">Turnover: </label>

<div class="input-group">

<input type="text" name="turnover" id="turnover" class="form-control"

placeholder="Enter the turnover..." required formControlName="turnover">

</div>

</div>

<div class="form-group">

<label for="briefDesc" class="form-control-label">Brief Description:</label>

<div class="input-group">

<textarea type="text" name="briefDesc" id="briefDesc" formControlName="brief"

class="form-control"></textarea>

</div>

</div>

<!-- <div class="form-group">

<label for="ipoDate" class="form-control-label">IPO date: </label>

<div class="input-group">

<input type="text" name="ipoDate" id="ipoDate" class="form-control"

placeholder="Enter the IPO date..." required formControlName="ipoDate">

</div>

</div> -->

<div class="row justify-content-around">

<div class="col-6">

<button type="reset" class="btn btn-danger btn-block">Reset</button>

</div>

<div class="col-6">

<button type="button" class="btn btn-secondary btn-block" (click)="goBack()">Cancel</button>

</div>

<button type="submit" class="btn btn-primary btn-block m-3"[disabled]="!newCompanyForm.valid">Add</button>

</div>

</form>

</div>

</div>

</div>

</main>

#### new-company.component.ts

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

import {

FormGroup,

FormControl,

FormArray,

Validators,

FormBuilder

} from "@angular/forms";

import bsCustomFileInput from "bs-custom-file-input";

import { CompaniesService } from "src/app/services/companies.service";

import { faTrash } from "@fortawesome/free-solid-svg-icons";

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

import { SectorsService } from "src/app/services/sectors.service";

import { Sector } from "src/app/models/sector";

import { StockExchangesService } from 'src/app/services/stock-exchanges.service';

import { StockExchange } from 'src/app/models/stockExchange';

@Component({

selector: "app-new-company",

templateUrl: "./new-company.component.html",

styleUrls: ["./new-company.component.css"]

})

export class NewCompanyComponent implements OnInit {

faTrash = faTrash;

newCompanyForm: FormGroup;

sectors: Sector[];

stockExchanges: StockExchange[];

constructor(

private formBuilder: FormBuilder,

private companiesService: CompaniesService,

private router: Router,

private sectorsService: SectorsService,

private stockExService: StockExchangesService

) {}

ngOnInit() {

bsCustomFileInput.init();

this.sectorsService.getAllSectors().subscribe(data => {

this.sectors = data;

});

this.stockExService.getAllExchanges().subscribe(data =>{

this.stockExchanges = data;

});

this.newCompanyForm = this.formBuilder.group({

id: [""],

name: ["", Validators.required],

sector: ["none", Validators.required],

ceo: ["", Validators.required],

directors: this.formBuilder.array([this.formBuilder.control("")]),

stockExchanges: this.formBuilder.array([

this.formBuilder.group({

stockExchange: ["", Validators.required],

stockCode: ["", Validators.required]

})

]),

turnover: ["", Validators.required],

brief: ["", Validators.required]

});

}

onSubmit() {

this.companiesService

.addCompany(this.newCompanyForm.value)

.subscribe(data => {

this.newCompanyForm.reset();

this.goBack();

});

}

addDirector() {

const control = this.formBuilder.control("");

(<FormArray>this.newCompanyForm.get("directors")).push(control);

}

removeDirector(i: number) {

(<FormArray>this.newCompanyForm.get("directors")).removeAt(i);

}

addStockExchange() {

const stockExGroup = this.formBuilder.group({

stockExchange: ["", Validators.required],

stockCode: ["", Validators.required]

});

(<FormArray>this.newCompanyForm.get("stockExchanges")).push(

stockExGroup

);

}

removeStockExchange(i: number) {

(<FormArray>this.newCompanyForm.get("stockExchanges")).removeAt(i);

}

goBack() {

this.router.navigate(["manage-company"]);

}

}

### NewExchange Component

#### new-exchange.component.html

<main class="container">

<div class="display-4 text-center my-4">Add New Stock Exchange</div>

<div class="row justify-content-center">

<div class="card col-md-8 col-sm-10 col-11 mx-auto shadow">

<div class="card-body">

<form [formGroup]="newExchangeForm" (ngSubmit)="onSubmit()">

<div class="form-group">

<label for="exchangeName" class="form-control-label">Name:</label>

<div class="input-group">

<input type="text" name="exchangeName" id="exchangeName" formControlName="name"

class="form-control" placeholder="Enter exchange name..." required>

</div>

</div>

<div class="form-group">

<label for="brief" class="form-control-label">Brief Description:</label>

<div class="input-group">

<textarea type="text" name="brief" id="brief" formControlName="brief"

class="form-control"></textarea>

</div>

</div>

<div class="form-group">

<label for="contactAddress" class="form-control-label">Contct address:</label>

<div class="input-group">

<input type="text" name="contactAddress" id="contactAddress"

formControlName="contactAddress" class="form-control"

placeholder="Enter contact address..." required>

</div>

</div>

<div class="form-group">

<label for="remarks" class="form-control-label">Remarks:</label>

<div class="input-group">

<input type="text" name="remarks" id="remarks" formControlName="remarks"

class="form-control" placeholder="Enter remarks...">

</div>

</div>

<div class="row justify-content-around">

<div class="col-6">

<button type="reset" class="btn btn-danger btn-block">Reset</button>

</div>

<div class="col-6">

<button type="button" class="btn btn-secondary btn-block" (click)="goBack()">Cancel</button>

</div>

<button type="submit" class="btn btn-primary btn-block m-3">Add</button>

</div>

</form>

</div>

</div>

</div>

</main>

#### new-exchange.component.ts

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

import {

FormGroup,

FormControl,

FormBuilder,

Validators

} from "@angular/forms";

import { StockExchangesService } from "src/app/services/stock-exchanges.service";

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

@Component({

selector: "app-new-exchange",

templateUrl: "./new-exchange.component.html",

styleUrls: ["./new-exchange.component.css"]

})

export class NewExchangeComponent implements OnInit {

newExchangeForm: FormGroup;

constructor(

private formBuilder: FormBuilder,

private stockExService: StockExchangesService,

private router: Router

) {}

ngOnInit() {

this.newExchangeForm = this.formBuilder.group({

id: [""],

name: ["", Validators.required],

brief: [""],

contactAddress: ["", Validators.required],

remarks: ["", Validators.required]

});

}

onSubmit() {

this.stockExService

.addStockExchange(this.newExchangeForm.value)

.subscribe(data => {

this.goBack();

});

}

goBack() {

this.router.navigate(["manage-exchange"]);

}

}

### NewIPO Component

#### new-ipo.component.html

<main class="container-fluid">

<div class="display-4 text-center my-4">Update IPO</div>

<div class="row justify-content-center">

<div class="card col-md-8 col-sm-10 col-11 mx-auto shadow">

<div class="card-body">

<form [formGroup]="newIPOForm" (ngSubmit)="onSubmit()">

<div class="form-group">

<label for="companyName" class="form-control-label">Company name:</label>

<div class="input-group">

<select name="companyName" id="companyName" formControlName="companyName"

class="custom-select" placeholder="Enter company name..." required

(change)="populateExchanges($event)">

<option value="" selected disabled>Select company</option>

<option \*ngFor="let company of companies" [value]="company.name">

{{company.name}}

</option>

</select>

</div>

</div>

<div class="form-group">

<label for="stockExchange" class="form-control-label">Stock Exchange:</label>

<div class="input-group">

<select name="stockExchange" id="stockExchange" formControlName="stockExchange"

class="form-control" placeholder="Enter stock exchange..." required>

<option value="" selected disabled>Select stock exchange</option>

<option \*ngFor="let stockEntry of stockExchanges" [value]="stockEntry.stockExchange">

{{stockEntry.stockExchange}}

</option>

</select>

</div>

</div>

<div class="form-group">

<label for="pricePerShare" class="form-control-label">Price per Share:</label>

<div class="input-group">

<input type="number" name="pricePerShare" id="pricePerShare" formControlName="pricePerShare"

class="form-control" placeholder="Enter price per share..." required>

</div>

</div>

<div class="form-group">

<label for="totalShares" class="form-control-label">Total shares:</label>

<div class="input-group">

<input type="number" name="totalShares" id="totalShares" formControlName="totalShares"

class="form-control" placeholder="Enter total shares..." required>

</div>

</div>

<div class="form-group">

<label for="openDateTime" class="form-control-label">Opening Date/Time: </label>

<div class="input-group">

<input type="datetime-local" name="openDateTime" id="openDateTime"

formControlName="openDateTime" class="form-control"

placeholder="Enter the opening date and time..." required>

</div>

</div>

<div class="form-group">

<label for="remarks" class="form-control-label">Remarks:</label>

<div class="input-group">

<textarea type="text" name="remarks" id="remarks" formControlName="remarks"

class="form-control"></textarea>

</div>

</div>

<div class="row justify-content-around">

<div class="col-6">

<button type="reset" class="btn btn-danger btn-block">Reset</button>

</div>

<div class="col-6">

<button type="button" class="btn btn-secondary btn-block" (click)="goBack()">Cancel</button>

</div>

<button type="submit" class="btn btn-primary btn-block m-3">Add</button>

</div>

</form>

</div>

</div>

</div>

</main>

#### new-ipo.component.ts

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

import { FormGroup, FormBuilder, Validators } from "@angular/forms";

import { IposService } from "src/app/services/ipos.service";

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

import { Company } from 'src/app/models/company';

import { CompaniesService } from 'src/app/services/companies.service';

@Component({

selector: "app-new-ipo",

templateUrl: "./new-ipo.component.html",

styleUrls: ["./new-ipo.component.css"]

})

export class NewIpoComponent implements OnInit {

newIPOForm: FormGroup;

companies: Company[];

stockExchanges: any[];

constructor(

private formBuilder: FormBuilder,

private iposService: IposService,

private companiesService: CompaniesService,

private router: Router

) {}

ngOnInit() {

this.companiesService.getAllCompanies().subscribe(data =>{

this.companies = data;

})

this.newIPOForm = this.formBuilder.group({

id: [""],

companyName: ["", Validators.required],

stockExchange: ["", Validators.required],

pricePerShare: ["", Validators.required],

totalShares: ["", Validators.required],

openDateTime: ["", Validators.required],

remarks: ["", Validators.required]

});

}

populateExchanges($event){

const index = $event.target.selectedIndex-1;

this.stockExchanges = this.companies[index].stockExchanges;

}

onSubmit() {

this.iposService.addNewIPO(this.newIPOForm.value).subscribe(data => {

this.newIPOForm.reset();

this.goBack();

});

}

goBack() {

this.router.navigate(["manage-ipos"]);

}

}

### NewSector Company

#### new-sector.component.html

<main class="container-fluid">

<div class="display-4 text-center my-4">Update IPO</div>

<div class="row justify-content-center">

<div class="card col-md-8 col-sm-10 col-11 mx-auto shadow">

<div class="card-body">

<form [formGroup]="newSectorForm" (ngSubmit)="onSubmit()">

<div class="form-group">

<label for="name" class="form-control-label">Sector name:</label>

<div class="input-group">

<input type="text" name="name" id="name" formControlName="name"

class="form-control" placeholder="Enter sector name..." required>

</div>

</div>

<div class="form-group">

<label for="brief" class="form-control-label">Brief:</label>

<div class="input-group">

<textarea type="text" name="brief" id="brief" formControlName="brief"

class="form-control"></textarea>

</div>

</div>

<div class="row justify-content-around">

<div class="col-6">

<button type="reset" class="btn btn-danger btn-block">Reset</button>

</div>

<div class="col-6">

<button type="button" class="btn btn-secondary btn-block" (click)="goBack()">Cancel</button>

</div>

<button type="submit" class="btn btn-primary btn-block m-3" [disabled]="!newSectorForm.valid">Add</button>

</div>

</form>

</div>

</div>

</div>

</main>

#### new-sector.component.ts

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

import { SectorsService } from "src/app/services/sectors.service";

import { FormBuilder, FormGroup, Validators } from "@angular/forms";

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

@Component({

selector: "app-new-sector",

templateUrl: "./new-sector.component.html",

styleUrls: ["./new-sector.component.css"]

})

export class NewSectorComponent implements OnInit {

newSectorForm: FormGroup;

constructor(

private sectorsService: SectorsService,

private formBuilder: FormBuilder,

private router: Router

) {}

ngOnInit() {

this.newSectorForm = this.formBuilder.group({

id: [""],

name: ["", Validators.required],

brief: ["", Validators.required]

});

}

onSubmit() {

this.sectorsService

.addNewSector(this.newSectorForm.value)

.subscribe(data => {

this.newSectorForm.reset();

this.goBack();

});

}

goBack() {

this.router.navigate(["manage-sectors"]);

}

}

### UpdateCompany Component

#### update-company.component.html

<main class="container">

<div class="display-4 text-center my-4">Update Company</div>

<div class="row justify-content-center">

<div class="card col-md-8 col-sm-10 col-11 mx-auto shadow">

<div class="card-body">

<form [formGroup]="updateCompanyForm" (ngSubmit)="onSubmit()">

<div class="row">

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

<img class="w-100 px-2" src="../../../assets/ph.svg" alt="">

</div>

<div class="col-8">

<span>Upload logo: </span>

<div class="custom-file my-3">

<input type="file" name="logoUpload" id="logoUpload">

<label for="logoUpload" class="custom-file-label"><span class="limit-text">Upload

logo...</span></label>

</div>

</div>

</div>

<div class="form-group">

<label for="companyName" class="form-control-label">Name:</label>

<div class="input-group">

<input type="text" name="companyName" id="companyName" formControlName="name"

class="form-control" placeholder="Enter company name..." required>

</div>

</div>

<div class="form-group">

<label for="sector">Select Company/Sector:</label>

<div class="input-group">

<select name="sector" id="sector" class="custom-select" formControlName="sector" required>

<option value="none" selected disabled>Select sector...</option>

<option value="Opt 1">Opt 1</option>

<option value="Opt 2">Opt 2</option>

</select>

</div>

</div>

<div class="form-group">

<label for="ceoName" class="form-control-label">CEO Name:</label>

<div class="input-group">

<input type="text" name="ceoName" id="ceoName" formControlName="ceo" class="form-control"

placeholder="Enter the CEO name..." required>

</div>

</div>

<div class="form-group" formArrayName="directors">

<div class="row px-3">

<label for="directors" class="form-control-label">Board of Directors:</label>

<button type="button" class="btn btn-info ml-auto btn-sm mb-2" role="add\_director"

(click)="addDirector()">Add

Director</button>

</div>

<div class="input-group mb-2"

\*ngFor="let directorControl of updateCompanyForm.get('directors').controls; let i = index">

<input type="text" [formControlName]="i" class="form-control "

placeholder="Enter the director name..." required>

<div class="input-group-append">

<button \*ngIf="i>0" class="btn btn-danger" (click)=(removeDirector(i))>

<fa-icon [icon]="faTrash"></fa-icon>

</button>

<button \*ngIf="i == 0" class="btn btn-secondary" disabled>

<fa-icon [icon]="faTrash"></fa-icon>

</button>

</div>

</div>

</div>

<div class="form-group" formArrayName="stockExchanges">

<div class="row px-3">

<label for="stockExchange" class="form-control-label">Stock Exchanges(with stock

code):</label>

<button type="button" class="btn btn-info ml-auto btn-sm mb-2"

(click)="addStockExchange()">Add Exchange</button>

</div>

<div \*ngFor="let StockExG of updateCompanyForm.get('stockExchanges').controls; let i = index"

class="mb-2">

<div class="input-group row mx-auto" [formGroupName]="i">

<input type="text" name="stockExchange" id="stockExchange"

formControlName="stockExchange" class="form-control col-7"

placeholder="Enter the stock exchange..." required>

<input type="text" name="stockCode" id="stockCode" class="form-control col-5"

placeholder="Enter the stock code..." required formControlName="stockCode">

<div class="input-group-append">

<button \*ngIf="i>0" class="btn btn-danger" (click)=(removeStockExchange(i))>

<fa-icon [icon]="faTrash"></fa-icon>

</button>

<button \*ngIf="i == 0" class="btn btn-secondary" disabled>

<fa-icon [icon]="faTrash"></fa-icon>

</button>

</div>

</div>

</div>

</div>

<div class="form-group">

<label for="turnover" class="form-control-label">Turnover: </label>

<div class="input-group">

<input type="text" name="turnover" id="turnover" class="form-control"

placeholder="Enter the turnover..." required formControlName="turnover">

</div>

</div>

<div class="form-group">

<label for="briefDesc" class="form-control-label">Brief Description:</label>

<div class="input-group">

<textarea type="text" name="briefDesc" id="briefDesc" formControlName="brief"

class="form-control"></textarea>

</div>

</div>

<!-- <div class="form-group">

<label for="ipoDate" class="form-control-label">IPO date: </label>

<div class="input-group">

<input type="text" name="ipoDate" id="ipoDate" class="form-control"

placeholder="Enter the IPO date..." required formControlName="ipoDate">

</div>

</div> -->

<div class="row justify-content-around">

<div class="col-6">

<button type="button" class="btn btn-secondary btn-block" (click)="goBack()">Cancel</button>

</div>

<div class="col-6">

<button type="submit" class="btn btn-primary btn-block"

[disabled]="!updateCompanyForm.valid">Update</button>

</div>

</div>

</form>

</div>

</div>

</div>

</main>

#### update-company.component.ts

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

import {

FormGroup,

FormControl,

FormArray,

Validators,

FormBuilder

} from "@angular/forms";

import bsCustomFileInput from "bs-custom-file-input";

import { CompaniesService } from "src/app/services/companies.service";

import { Company } from "src/app/models/company";

import { faTrash } from "@fortawesome/free-solid-svg-icons";

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

@Component({

selector: "app-update-company",

templateUrl: "./update-company.component.html",

styleUrls: ["./update-company.component.css"]

})

export class UpdateCompanyComponent implements OnInit {

updateCompanyForm: FormGroup;

noOfDirectors: number;

noOfExchanges: number;

company: Company;

faTrash = faTrash;

constructor(

private formBuilder: FormBuilder,

private companiesService: CompaniesService,

private router: Router

) {}

ngOnInit() {

bsCustomFileInput.init();

const id = localStorage.getItem("companyId");

console.log(id);

if (+id > 0) {

this.companiesService.getCompayById(+id).subscribe(data => {

this.noOfDirectors = data.directors.length;

this.noOfExchanges = data.stockExchanges.length;

console.log(this.noOfDirectors + " " + this.noOfExchanges);

while (this.noOfDirectors > 0) {

this.addDirector();

this.noOfDirectors--;

}

while (this.noOfExchanges > 0) {

this.addStockExchange();

this.noOfExchanges--;

}

this.updateCompanyForm.patchValue(data);

});

}

this.updateCompanyForm = this.formBuilder.group({

id: [""],

name: ["", Validators.required],

sector: ["", Validators.required],

ceo: ["", Validators.required],

directors: this.formBuilder.array([]),

stockExchanges: this.formBuilder.array([]),

turnover: ["", Validators.required],

brief: ["", Validators.required]

});

}

onSubmit() {

this.companiesService

.updateCompany(this.updateCompanyForm.value)

.subscribe(data => {

this.updateCompanyForm.reset();

});

}

addDirector() {

const control = this.formBuilder.control("");

(<FormArray>this.updateCompanyForm.get("directors")).push(control);

}

removeDirector(i: number) {

(<FormArray>this.updateCompanyForm.get("directors")).removeAt(i);

}

addStockExchange() {

const stockExGroup = this.formBuilder.group({

stockExchange: ["", Validators.required],

stockCode: ["", Validators.required]

});

(<FormArray>this.updateCompanyForm.get("stockExchanges")).push(

stockExGroup

);

}

removeStockExchange(i: number) {

(<FormArray>this.updateCompanyForm.get("stockExchanges")).removeAt(i);

}

goBack() {

this.router.navigate(["manage-company"]);

}

}

### UpdateExchange Component

#### update-exchnage.component.html

<main class="container">

<div class="display-4 text-center my-4">Add New Stock Exchange</div>

<div class="row justify-content-center">

<div class="card col-md-8 col-sm-10 col-11 mx-auto shadow">

<div class="card-body">

<form [formGroup]="updateExchangeForm" (ngSubmit)="onSubmit()">

<div class="form-group">

<label for="exchangeName" class="form-control-label">Name:</label>

<div class="input-group">

<input type="text" name="exchangeName" id="exchangeName" formControlName="name"

class="form-control" placeholder="Enter exchange name..." required>

</div>

</div>

<div class="form-group">

<label for="brief" class="form-control-label">Brief Description:</label>

<div class="input-group">

<textarea type="text" name="brief" id="brief" formControlName="brief"

class="form-control"></textarea>

</div>

</div>

<div class="form-group">

<label for="contactAddress" class="form-control-label">Contct address:</label>

<div class="input-group">

<input type="text" name="contactAddress" id="contactAddress"

formControlName="contactAddress" class="form-control"

placeholder="Enter contact address..." required>

</div>

</div>

<div class="form-group">

<label for="remarks" class="form-control-label">Remarks:</label>

<div class="input-group">

<input type="text" name="remarks" id="remarks" formControlName="remarks"

class="form-control" placeholder="Enter remarks...">

</div>

</div>

<div class="row justify-content-around">

<div class="col-6">

<button type="button" class="btn btn-secondary btn-block" (click)="goBack()">Cancel</button>

</div>

<div class="col-6">

<button type="submit" class="btn btn-danger btn-block">Update</button>

</div>

</div>

</form>

</div>

</div>

</div>

</main>

#### update-exchange.component.ts

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

import { FormGroup, FormBuilder, Validators } from "@angular/forms";

import { StockExchangesService } from "src/app/services/stock-exchanges.service";

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

@Component({

selector: "app-update-exchange",

templateUrl: "./update-exchange.component.html",

styleUrls: ["./update-exchange.component.css"]

})

export class UpdateExchangeComponent implements OnInit {

updateExchangeForm: FormGroup;

constructor(

private formBuilder: FormBuilder,

private stockExService: StockExchangesService,

private router: Router

) {}

ngOnInit() {

const stockExId = localStorage.getItem("stockExId");

if (+stockExId > 0) {

this.stockExService.getExchangeById(+stockExId).subscribe(data => {

this.updateExchangeForm.patchValue(data);

});

}

this.updateExchangeForm = this.formBuilder.group({

id: [""],

name: ["", Validators.required],

brief: [""],

contactAddress: ["", Validators.required],

remarks: ["", Validators.required]

});

}

onSubmit() {

this.stockExService

.updateStockExchange(this.updateExchangeForm.value)

.subscribe(data => {

this.updateExchangeForm.reset();

this.goBack();

});

}

goBack() {

this.router.navigate(["manage-exchange"]);

}

}

### UpdateIPO Component

#### update-ipo.component.html

<main class="container-fluid">

<div class="display-4 text-center my-4">Update IPO</div>

<div class="row justify-content-center">

<div class="card col-md-8 col-sm-10 col-11 mx-auto shadow">

<div class="card-body">

<form [formGroup]="updateIPOForm" (ngSubmit)="onSubmit()">

<div class="form-group">

<label for="companyName" class="form-control-label">Company name:</label>

<div class="input-group">

<input type="text" name="companyName" id="companyName" formControlName="companyName"

class="form-control" placeholder="Enter company name..." required>

</div>

</div>

<div class="form-group">

<label for="stockExchange" class="form-control-label">Stock Exchange:</label>

<div class="input-group">

<input type="text" name="stockExchange" id="stockExchange" formControlName="stockExchange"

class="form-control" placeholder="Enter stock exchange..." required>

</div>

</div>

<div class="form-group">

<label for="pricePerShare" class="form-control-label">Price per Share:</label>

<div class="input-group">

<input type="number" name="pricePerShare" id="pricePerShare" formControlName="pricePerShare"

class="form-control" placeholder="Enter price per share..." required>

</div>

</div>

<div class="form-group">

<label for="totalShares" class="form-control-label">Total shares:</label>

<div class="input-group">

<input type="number" name="totalShares" id="totalShares" formControlName="totalShares"

class="form-control" placeholder="Enter total shares..." required>

</div>

</div>

<div class="form-group">

<label for="openDateTime" class="form-control-label">Opening Date/Time: </label>

<div class="input-group">

<input type="datetime-local" name="openDateTime" id="openDateTime"

formControlName="openDateTime" class="form-control"

placeholder="Enter the opening date and time..." required>

</div>

</div>

<div class="form-group">

<label for="remarks" class="form-control-label">Remarks:</label>

<div class="input-group">

<textarea type="text" name="remarks" id="remarks" formControlName="remarks"

class="form-control"></textarea>

</div>

</div>

<div class="row justify-content-around">

<div class="col-6">

<button type="button" class="btn btn-secondary btn-block" (click)="this.router.navigate(['manage-ipos'])">Cancel</button>

</div>

<div class="col-6">

<button type="submit" class="btn btn-primary btn-block" [disabled]="!updateIPOForm.valid">Save</button>

</div>

</div>

</form>

</div>

</div>

</div>

</main>

#### update-ipo.component.ts

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

import { FormBuilder, FormGroup, Validators } from "@angular/forms";

import { IposService } from "src/app/services/ipos.service";

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

@Component({

selector: "app-update-ipo",

templateUrl: "./update-ipo.component.html",

styleUrls: ["./update-ipo.component.css"]

})

export class UpdateIpoComponent implements OnInit, OnDestroy {

updateIPOForm: FormGroup;

constructor(

private formBuilder: FormBuilder,

private iposService: IposService,

private router: Router

) {}

ngOnInit() {

const id = sessionStorage.getItem("ipoId");

if (+id > 0) {

this.iposService.getIPOById(+id).subscribe(data => {

this.updateIPOForm.patchValue(data);

});

} else this.router.navigate(["manage-ipos"]);

this.updateIPOForm = this.formBuilder.group({

id: [""],

companyName: ["", Validators.required],

stockExchange: ["", Validators.required],

pricePerShare: ["", Validators.required],

totalShares: ["", Validators.required],

openDateTime: ["", Validators.required],

remarks: ["", Validators.required]

});

}

onSubmit() {

this.iposService.updateIPO(this.updateIPOForm.value).subscribe(data => {

this.router.navigate(["manage-ipos"]);

});

}

ngOnDestroy() {

sessionStorage.removeItem("ipoId");

}

}

### UpdateSector Component

#### update-sector.component.html

<main class="container-fluid">

<div class="display-4 text-center my-4">Update IPO</div>

<div class="row justify-content-center">

<div class="card col-md-8 col-sm-10 col-11 mx-auto shadow">

<div class="card-body">

<form [formGroup]="updateSectorForm" (ngSubmit)="onSubmit()">

<div class="form-group">

<label for="name" class="form-control-label">Sector name:</label>

<div class="input-group">

<input type="text" name="name" id="name" formControlName="name" class="form-control"

placeholder="Enter sector name..." required>

</div>

</div>

<div class="form-group">

<label for="brief" class="form-control-label">Brief:</label>

<div class="input-group">

<textarea type="text" name="brief" id="brief" formControlName="brief"

class="form-control"></textarea>

</div>

</div>

<div class="row justify-content-around">

<div class="col-6">

<button type="cancel" class="btn btn-secondary btn-block"

(click)="this.router.navigate(['manage-sectors'])">

Cancel

</button>

</div>

<div class="col-6">

<button type="submit" class="btn btn-primary btn-block"

[disabled]="!updateSectorForm.valid">

Save

</button>

</div>

</div>

</form>

</div>

</div>

</div>

</main>

#### update-sector.component.ts

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

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

import { FormBuilder, FormGroup, Validators } from "@angular/forms";

import { SectorsService } from "src/app/services/sectors.service";

@Component({

selector: "app-update-sector",

templateUrl: "./update-sector.component.html",

styleUrls: ["./update-sector.component.css"]

})

export class UpdateSectorComponent implements OnInit {

updateSectorForm: FormGroup;

constructor(

private sectorsService: SectorsService,

private formBuilder: FormBuilder,

private router: Router

) {}

ngOnInit() {

const id = sessionStorage.getItem("sectorId");

if (+id > 0) {

this.sectorsService.getSectorById(+id).subscribe(data => {

this.updateSectorForm.patchValue(data);

});

} else this.router.navigate(["manage-sectors"]);

this.updateSectorForm = this.formBuilder.group({

id: [""],

name: ["", Validators.required],

brief: ["", Validators.required]

});

}

onSubmit(){

this.sectorsService.updateSector(this.updateSectorForm.value).subscribe(data=>{

this.router.navigate(['manage-sectorsF'])

})

}

}

### ViewUsers Component

#### view-users.component.html

<main class="container">

<div class="display-4 text-center my-4">List of Users</div>

<div class="row justify-content-center">

<div class="table-responsive">

<table class="table table-hover table-sm table-stripped mx-auto w-auto bg-light">

<thead class="thead-light">

<th>ID</th>

<th>Username</th>

<th>Email</th>

<th>Phone</th>

<th>Confirmed</th>

<th class="text-center">Actions</th>

</thead>

<tbody>

<tr \*ngFor="let user of users">

<td>{{ user.id }}</td>

<td>{{ user.username }}</td>

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

<td>{{ user.phoneNo }}</td>

<td>{{ user.confirmed }}</td>

<td>

<button type="button" class="btn btn-danger" (click)="deleteUser(user.id)">

<fa-icon [icon]="faTrash"></fa-icon>

</button>

</td>

</tr>

</tbody>

</table>

</div>

</div>

</main>

#### view-users.component.ts

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

import { UserService } from "src/app/services/user.service";

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

import { User } from "src/app/models/users";

import { faUserMinus } from "@fortawesome/free-solid-svg-icons";

@Component({

selector: "app-view-users",

templateUrl: "./view-users.component.html",

styleUrls: ["./view-users.component.css"]

})

export class ViewUsersComponent implements OnInit {

faTrash = faUserMinus;

users: User[];

constructor(private userService: UserService, private router: Router) {}

ngOnInit() {

this.userService.getAllUsers().subscribe(data => {

this.users = data;

});

}

deleteUser(id: number) {

this.userService.deleteUser(id).subscribe(data => {

this.users = this.users.filter(user => user.id !== id);

});

}

}

## User Components

### Account Component

#### account.component.html

<main class="container ">

<div class="card w-md-75 w-100 mx-auto mt-5">

<div class="card-body">

<div class="row mx-5">

<div class="row">

<img src="../../../assets/ph.svg" alt="Profile image" class="rounded-circle profile mt-2 p-0">

<h5 class="display-4 ml-3 m-0 align-self-end d-md-block d-none"> {{user.username}} </h5>

<h4 class="ml-3 m-0 align-self-end d-block d-md-none"> {{user.username}} </h4>

</div>

</div>

<div class="row mt-5 mx-5">

<span class="col-auto">Email:</span>

<span class="col-auto"> {{user.email}} </span>

</div>

<div class="row mt-4 mx-5">

<span class="col-auto">Phone Number:</span>

<span class="col-auto"> {{user.phoneNo}} </span>

</div>

</div>

<div class="card-footer text-right">

<!-- <button type="button" class="btn btn-primary">Change Password</button> -->

<button type="button" class="btn btn-secondary ml-2" routerLink="/account-update">Update Details</button>

</div>

</div>

</main>

#### account.component.ts

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

import { UserService } from "src/app/services/user.service";

import { User } from "src/app/models/users";

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

@Component({

selector: "app-account",

templateUrl: "./account.component.html",

styleUrls: ["./account.component.css"]

})

export class AccountComponent implements OnInit {

user: User;

constructor(private userService: UserService, private router: Router) {}

ngOnInit() {

let id: string;

if(sessionStorage.getItem("userId") !== null)

id = sessionStorage.getItem("userId");

else

id = localStorage.getItem("userId");

if (+id > 0) {

this.userService.getUserById(+id).subscribe(data => {

this.user = data;

});

} else this.router.navigate(['/landing']);

}

}

### AccountUpdate Component

#### account-update.component.html

<main class="container">

<div class="display-4 text-center my-4">Account Update</div>

<div class="row justify-content-center">

<div class="card col-md-8 col-sm-10 col-11 mx-auto shadow">

<div class="card-body">

<form [formGroup]="accountUpdateForm" (ngSubmit)="onSubmit()">

<div class="form-group">

<label for="username" class="form-control-label">Username:</label>

<div class="input-group">

<input type="text" name="username" id="username" class="form-control"

placeholder="Enter username..." formControlName="username">

</div>

</div>

<div class="form-group">

<label for="email" class="form-control-label">E-mail:</label>

<div class="input-group">

<input type="email" name="email" id="email" class="form-control"

placeholder="Enter your email..." formControlName="email">

</div>

</div>

<div class="form-group">

<label for="phoneNo" class="form-control-label">Phone number: </label>

<div class="input-group">

<input type="number" name="phoneNo" id="phoneNo" class="form-control"

placeholder="Enter phone number..." formControlName="phoneNo">

</div>

</div>

<div class="row justify-content-around">

<div class="col-6">

<button type="button" class="btn btn-secondary btn-block" (click)="goBack()">

Cancel

</button>

</div>

<div class="col-6">

<button type="submit" class="btn btn-primary btn-block"

[disabled]="!accountUpdateForm.valid">Update</button>

</div>

</div>

</form>

</div>

</div>

</div>

</main>

#### account-update.component.ts

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

import { Validators, FormBuilder, FormGroup } from "@angular/forms";

import { UserService } from "src/app/services/user.service";

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

@Component({

selector: "app-account-update",

templateUrl: "./account-update.component.html",

styleUrls: ["./account-update.component.css"]

})

export class AccountUpdateComponent implements OnInit {

accountUpdateForm: FormGroup;

constructor(

private formBuilder: FormBuilder,

private userService: UserService,

private router: Router

) {}

ngOnInit() {

let id: string;

if(sessionStorage.getItem("userId") !== null)

id = sessionStorage.getItem("userId");

else

id = localStorage.getItem("userId");

if (+id > 0) {

this.userService.getUserById(+id).subscribe(data => {

this.accountUpdateForm.patchValue(data);

});

} else this.router.navigate(["/landing"]);

this.accountUpdateForm = this.formBuilder.group({

id: [""],

username: ["", Validators.required],

email: ["", [Validators.required, Validators.email]],

password: [""],

rePassword: [""],

phoneNo: [""],

isAdmin: [""],

confirmed: [""]

});

}

onSubmit() {

console.log(this.accountUpdateForm.value);

this.userService

.updateUser(this.accountUpdateForm.value)

.subscribe(data => {

this.accountUpdateForm.reset();

});

}

goBack() {

this.router.navigate(["/account"]);

}

}

### Compare Component

#### compare.component.html

<main class="container">

<div class="display-4 text-center my-4">Compare Companies</div>

<div class="row justify-content-center">

<div class="card col-md-6 col-sm-10 mx-auto shadow">

<div class="card-body">

<form [formGroup]="compareForm">

<div class="form-group">

<label for="company-sector" >Select Company/Sector:</label>

<select name="company-sector" id="company-sector" formControlName="companySector" class="custom-select" required>

<option value="" selected disabled>Select comapny/sector...</option>

<option value="">Company</option>

<option value="">Sector</option>

</select>

</div>

<div class="form-group">

<label for="stock-ex" >Select stock exchange:</label>

<select name="stock-ex" id="stock-ex" formControlName="stockEx" class="custom-select" required>

<option value="" selected disabled>Select stock exchange...</option>

<option value="">One</option>

<option value="">Two</option>

<option value="">Three</option>

</select>

</div>

<div class="form-group">

<label for="name" >Company Name: </label>

<input type="text" name="name" id="name" formControlName="name" class="form-control"

placeholder="Enter company/sector name..." required>

</div>

<div class="row">

<div class="col-6">

<label for="from-date">From: </label>

<input type="date" name="from-date" id="from-date" formControlName="fromDate" class="form-control"

placeholder="Confirm your password..." required>

</div>

<div class="col-6">

<label for="to-date" >To: </label>

<input type="date" name="to-date" id="to-date" formControlName="toDate" class="form-control"

placeholder="Confirm your password..." required>

</div>

</div>

</form>

</div>

</div>

</div>

<button type="submit" class="btn btn-primary btn-block mt-3 col-md-6 mx-auto">Compare</button>

<button type="button" class="btn btn-primary rounded-circle mx-auto add-btn shadow">

<fa-icon [icon]="faPlus" class="h2"></fa-icon>

</button>

</main>

#### compare.component.ts

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

import { FormGroup, FormBuilder, Validators } from "@angular/forms";

import { faPlus } from "@fortawesome/free-solid-svg-icons"

@Component({

selector: "app-compare",

templateUrl: "./compare.component.html",

styleUrls: ["./compare.component.css"]

})

export class CompareComponent implements OnInit {

compareForm: FormGroup;

faPlus = faPlus;

constructor(private formBuilder: FormBuilder) {}

ngOnInit() {

this.compareForm = this.formBuilder.group({

compantSector: ["", Validators.required],

stockEx: ["", Validators.required],

name: ["", Validators.required],

formatDate: ["", Validators.required],

toDate: ["", Validators.required]

});

}

}

### OTP Component

#### otp.component.html

<main class="container">

<div class="display-4 text-center my-4">Verify Account</div>

<div class="row justify-content-center">

<div class="card col-md-8 col-sm-10 col-11 mx-auto shadow">

<div class="card-body">

<form [formGroup]="otpForm" (ngSubmit)="verifyOTP()">

<div class="form-group">

<label for="otp" class="form-control-label"></label>

<input type="text" name="otp" id="otp" formControlName="otp" class="form-control"

placeholder="Enter your OTP..." required>

</div>

<button type="submit" class="btn btn-primary btn-block mt-3" [disabled]="!this.otpForm.valid">Verify</button>

</form>

</div>

</div>

</div>

</main>

#### otp.component.ts

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

import { FormGroup, FormBuilder, Validators } from "@angular/forms";

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

declare var $: any;

@Component({

selector: "app-otp",

templateUrl: "./otp.component.html",

styleUrls: ["./otp.component.css"]

})

export class OtpComponent implements OnInit {

otpForm: FormGroup;

constructor(private formBuilder: FormBuilder, private router: Router) {}

ngOnInit() {

this.otpForm = this.formBuilder.group({

otp: ["", Validators.required]

});

}

verifyOTP() {

console.log(this.otpForm);

$("#login-modal").modal("show");

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

}

}

### Signup Component

#### signup.component.html

<main class="container">

<div class="display-4 text-center my-4">Signup</div>

<div class="row justify-content-center">

<div class="card col-md-8 col-sm-10 col-11 mx-auto shadow">

<div class="card-body">

<form [formGroup]="signupForm" (ngSubmit)="onSubmit()">

<div class="form-group">

<label for="username" class="form-control-label">Name:</label>

<input type="text" name="username" id="username" class="form-control"

placeholder="Enter username..." formControlName="username" (keyup)="checkUsername()" (change)="checkUsername()">

<span class="text-danger" \*ngIf="usernameTaken"><small>This username is already taken.</small></span>

</div>

<div class="form-group">

<label for="email" class="form-control-label">E-mail:</label>

<input type="email" name="email" id="email" class="form-control"

placeholder="Enter your email..." formControlName="email" (keyup)="checkEmail()" (change)="checkEmail()">

<span class="text-danger" \*ngIf="emailTaken"><small>This email is already in use.</small></span>

<span class="text-danger" \*ngIf="!signupForm.get('email').valid && signupForm.get('email').touched"><small>Enter a valid email id</small></span>

</div>

<div class="form-group">

<label for="password" class="form-control-label">Password: </label>

<input type="password" name="password" id="password" class="form-control"

placeholder="Enter your password..." formControlName="password">

<span class="text-danger" \*ngIf="!signupForm.get('password').valid && signupForm.get('password').touched"><small>Enter a valid password</small></span>

</div>

<div class="form-group">

<label for="rePassword" class="form-control-label">Confirm Password: </label>

<input type="password" name="rePassword" id="rePassword" class="form-control"

placeholder="Confirm your password..." formControlName="rePassword" (keyup)="checkPasswordMatch()">

<span class="text-danger" \*ngIf="!passwordsMatch"><small>The passwords do not match</small></span>

<span class="text-danger" \*ngIf="!signupForm.get('rePassword').valid && signupForm.get('rePassword').touched"><small>Please reenter password</small></span>

</div>

<div class="form-group">

<label for="phoneNo" class="form-control-label">Phone number: </label>

<input type="number" name="phoneNo" id="phoneNo" class="form-control"

placeholder="Enter phone number..." formControlName="phoneNo">

<span class="text-danger" \*ngIf="!signupForm.get('phoneNo').valid && signupForm.get('phoneNo').touched"><small>Enter a valid phone number</small></span>

</div>

<button type="submit" class="btn btn-primary btn-block mt-3" [disabled]="!signupForm.valid || !passwordsMatch || usernameTaken || emailTaken">Signup</button>

<span class="row text-muted justify-content-center my-2"><small>Already a user? <a href="#"

data-toggle="modal" data-target="#login-modal">Login</a></small></span>

</form>

</div>

</div>

</div>

</main>

<div class="modal fade" id="successModal" data-backdrop="static">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content">

<div class="modal-body">

You have been successfully registered.<br>

<p class="pt-2">

An OTP has been sent to your email. Please enter the otp on the next page.

</p>

</div>

<div class="modal-footer">

<button class="btn btn-primary" (click)="gotoOTP()">Ok</button>

</div>

</div>

</div>

</div>

#### signup.component.ts

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

import { FormGroup, Validators, FormBuilder } from "@angular/forms";

import { UserService } from "src/app/services/user.service";

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

import { User } from 'src/app/models/users';

declare var $: any;

@Component({

selector: "app-signup",

templateUrl: "./signup.component.html",

styleUrls: ["./signup.component.css"]

})

export class SignupComponent implements OnInit {

signupForm: FormGroup;

users:User[];

usernameTaken:boolean = false;

emailTaken: boolean = false;

passwordsMatch: boolean = true;

constructor(

private formBuilder: FormBuilder,

private userService: UserService,

private router: Router

) {}

ngOnInit() {

this.userService.getAllUsers().subscribe(data =>{

this.users = data;

})

this.signupForm = this.formBuilder.group({

id: [""],

username: ["", Validators.required],

email: ["", [Validators.required, Validators.email]],

password: ["", Validators.required],

rePassword: ["", Validators.required],

phoneNo: ["", Validators.required],

isAdmin: [false],

confirmed: [false]

});

}

checkUsername(){

this.usernameTaken = false;

for(let user of this.users){

if(user.username == this.signupForm.get("username").value){

this.usernameTaken = true;

break;

}

}

}

checkEmail(){

this.emailTaken = false;

for(let user of this.users){

console.log('ggg');

if(user.email == this.signupForm.get("email").value){

this.signupForm.setErrors({

valid: false

});

this.emailTaken = true;

console.log('here');

break;

}

}

}

checkPasswordMatch(){

if(this.signupForm.get("password").value == this.signupForm.get("rePassword").value){

this.passwordsMatch = true;

} else {

this.passwordsMatch = false;

}

}

onSubmit() {

console.log(this.signupForm.value);

this.userService.registerUser(this.signupForm.value).subscribe(data => {

this.signupForm.reset();

$("#successModal").modal("show");

});

}

gotoOTP() {

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

$("#successModal").modal("hide");

}

}

### UserHeader Component

#### user-header.component.html

<header>

<nav class="navbar navbar-expand-md navbar-dark bg-dark sticky-top">

<a routerLink="landing" class="navbar-brand">

<img src="../assets/logo.svg" alt="StockView" id="logo" class="undraggable">

<span class="text-light mx-2 my-auto undraggable">StockView</span>

</a>

<div class="nav-item dropdown text-light ml-auto ml-md-5 pl-0 pl-md-5 col-auto not-md">

<a class="btn dropdown-toggle text-light nav-link" data-toggle="dropdown">

<fa-icon [icon]="faUser"></fa-icon>

</a>

<div class="dropdown-menu dropdown-menu-right">

<a routerLink="/account" class="dropdown-item">My Account</a>

<a class="dropdown-item btn" (click)="logout()">Logout</a>

</div>

</div>

<button type="button" class="navbar-toggler" data-toggle="collapse" data-target="#navbar1">

<span class="navbar-toggler-icon"></span></button>

<div class="navbar-collapse collapse" id="navbar1">

<ul class="navbar-nav ml-auto">

<li class="nav-item"><a routerLink="/landing" class="nav-link active">Home</a></li>

<li class="nav-item"><a routerLink="/compare" class="nav-link">Compare</a>

</li>

<li class="nav-item"><a routerLink="/view-sectors" class="nav-link">Sectors</a></li>

<li class="nav-item"><a routerLink="/view-ipos" class="nav-link">IPO</a></li>

<li class="nav-item"><a routerLink="/view-companies" class="nav-link">Companies</a></li>

<li class="nav-item"><a routerLink="/view-exchanges" class="nav-link">Exchanges</a></li>

</ul>

</div>

<div class="nav-item dropdown text-light ml-auto ml-md-5 pl-0 pl-md-5 col-auto md-only">

<a class="btn dropdown-toggle text-light nav-link" data-toggle="dropdown">

<fa-icon [icon]="faUser"></fa-icon>

</a>

<div class="dropdown-menu dropdown-menu-right">

<a routerLink="/account" class="dropdown-item">My Account</a>

<a class="dropdown-item btn" (click)="logout()">Logout</a>

</div>

</div>

</nav>

</header>

#### user-header.component.ts

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

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

import { faUser } from "@fortawesome/free-solid-svg-icons"

@Component({

selector: 'app-user-header',

templateUrl: './user-header.component.html',

styleUrls: ['./user-header.component.css']

})

export class UserHeaderComponent implements OnInit {

faUser = faUser;

constructor(private router: Router) { }

ngOnInit() {

}

logout(){

console.log('logging out user');

localStorage.removeItem("userType");

sessionStorage.removeItem("userType");

localStorage.removeItem("userId");

sessionStorage.removeItem("userId");

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

}

}

### UserHome Component

#### user-home.component.html

<section class="jumbotron jumbotron-fluid py-4">

<div class="container">

<h1 class="display-4 text-center">Hi! <span \*ngIf="user">{{ user.username }}</span>,<br>

<p class="lead text-center mt-5">What would you like to do?</p>

</h1>

</div>

</section>

<main class="container">

<div class="row card-deck">

<a class="card w-100 shadow hover-enlarge mt-4 mx-4 mx-sm-3" routerLink="/compare">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../../../assets/ph.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">Compare companies</h4>

<p class="card-text text-center mb-3">Get charts comparing the stocks of one company/sector or different

companies/sectors ovr a period of time.</p>

</div>

</a>

<a class="card w-100 shadow hover-enlarge mt-4 mx-4 mx-sm-3" routerLink="/view-sectors">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../../../assets/ph.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">View Sectors</h4>

<p class="card-text text-center mb-3">See all the available sectors.</p>

</div>

</a>

<!-- </div>

<div class="row card-deck"> -->

<a class="card w-100 shadow hover-enlarge mt-4 mx-4 mx-sm-3" routerLink="/view-exchanges">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../../../assets/ph.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">View Exchanges</h4>

<p class="card-text text-center mb-3">View the available exchanges and their details.</p>

</div>

</a>

</div>

<div class="row card-deck">

<a class="card w-100 shadow hover-enlarge mt-4 mx-4 mx-sm-3" routerLink="/view-companies">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../../../assets/ph.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">View Companies</h4>

<p class="card-text text-center mb-3">View the available companies and their details.</p>

</div>

</a>

<!-- </div>

<div class="row card-deck"> -->

<a class="card w-100 shadow hover-enlarge mt-4 mx-4 mx-sm-3" routerLink="/view-ipos">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../../../assets/ph.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">View IPOs</h4>

<p class="card-text text-center mb-3">Check on upcoming IPOs</p>

</div>

</a>

<a class="card w-100 shadow hover-enlarge mt-4 mx-4 mx-sm-3" routerLink="/account">

<img class="card-img-top pt-4 mx-auto adv-imgs" src="../../../assets/ph.svg" alt="">

<div class="card-body">

<h4 class="card-title text-center">Manage Account</h4>

<p class="card-text text-center mb-3">Manage your account.</p>

</div>

</a>

</div>

</main>

#### user-home.component.ts

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

import { User } from 'src/app/models/users';

import { UserService } from 'src/app/services/user.service';

@Component({

selector: 'app-user-home',

templateUrl: './user-home.component.html',

styleUrls: ['./user-home.component.css']

})

export class UserHomeComponent implements OnInit {

user: User;

constructor(private userService: UserService) { }

ngOnInit() {

let id: string;

if(sessionStorage.getItem("userId") !== null){

id = sessionStorage.getItem("userId");

}

else{

id = localStorage.getItem("userId");

}

if (+id > 0) {

this.userService.getUserById(+id).subscribe(data => {

this.user = data;

});

}

}

}

### ViewCompanies Component

#### view-companies.component.html

<main class="container">

<div class="display-4 text-center my-4">Companies</div>

<div class="row justify-content-center">

<div class="table-responsive">

<table class="table table-stripped table-hover text-center bg-light">

<thead class="thead-light">

<th>ID</th>

<th>Name</th>

<th>Sector</th>

<th>CEO</th>

<th>Board of Directors</th>

<th>Stock Exchanges</th>

<th>Turnover</th>

<th>Brief</th>

</thead>

<tbody>

<tr \*ngFor="let company of companies">

<td> {{company.id}} </td>

<td> {{company.name}} </td>

<td> {{company.sector}} </td>

<td> {{company.ceo}} </td>

<td>

<div \*ngFor="let director of company.directors" class="row justify-content-center">

{{director}}

</div>

</td>

<td>

<div \*ngFor="let exchange of company.stockExchanges" class="row justify-content-center">

{{exchange.stockExchange}} : {{exchange.stockCode}}

</div>

</td>

<td> {{company.turnover}} </td>

<td> {{company.brief}} </td>

</tr>

</tbody>

</table>

</div>

</div>

</main>

#### view-companies.component.ts

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

import { CompaniesService } from 'src/app/services/companies.service';

import { Company } from 'src/app/models/company';

@Component({

selector: 'app-view-companies',

templateUrl: './view-companies.component.html',

styleUrls: ['./view-companies.component.css']

})

export class ViewCompaniesComponent implements OnInit {

companies: Company[];

constructor(private companiesService: CompaniesService) { }

ngOnInit() {

this.companiesService.getAllCompanies().subscribe(data => {

this.companies = data;

});

}

}

### ViewExchanges Components

#### view-exchanges.component.html

<main class="container">

<div class="display-4 text-center my-4">Exchanges</div>

<div class="table-responsive">

<table class="table table-stripped table-hover text-center bg-light rounded">

<thead class="thead-light">

<tr>

<th>ID</th>

<th>Name</th>

<th>Brief</th>

<th>Contact Address</th>

<th>Remarks</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let stockEx of stockExchanges">

<td> {{stockEx.id}} </td>

<td> {{stockEx.name}} </td>

<td> {{stockEx.brief}} </td>

<td> {{stockEx.contactAddress}} </td>

<td> {{stockEx.remarks}} </td>

</tr>

</tbody>

</table>

</div>

</main>

#### view-exchanges.component.ts

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

import { StockExchangesService } from "src/app/services/stock-exchanges.service";

import { StockExchange } from "src/app/models/stockExchange";

@Component({

selector: "app-view-exchanges",

templateUrl: "./view-exchanges.component.html",

styleUrls: ["./view-exchanges.component.css"]

})

export class ViewExchangesComponent implements OnInit {

stockExchanges: StockExchange[];

constructor(private stockExService: StockExchangesService) {}

ngOnInit() {

this.stockExService.getAllExchanges().subscribe(data => {

this.stockExchanges = data;

console.log(this.stockExchanges);

});

}

}

### ViewIPO Component

#### view-ipo.component.html

<main class="container">

<div class="display-4 text-center my-4">IPOs</div>

<div class="table-responsive">

<table class="table table-stripped table-hover text-center bg-light">

<thead class="thead-light">

<tr>

<th>ID</th>

<th>Company name</th>

<th>Stock Exchange</th>

<th>Price per share</th>

<th>Total shares</th>

<th>Open Date/Time</th>

<th>Remarks</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let ipo of ipos">

<td> {{ipo.id}} </td>

<td> {{ipo.companyName}} </td>

<td> {{ipo.stockExchange}} </td>

<td> {{ipo.pricePerShare | currency: "INR"}} </td>

<td> {{ipo.totalShares}} </td>

<td> {{ipo.openDateTime | date: "dd/MM/yyyy hh:mm a"}} </td>

<td> {{ipo.remarks}} </td>

</tr>

</tbody>

</table>

</div>

<button type="button" class="btn btn-primary rounded-circle mx-auto add-btn shadow" (click)="gotoNew()">

<fa-icon [icon]="faPlus" class="h2"></fa-icon>

</button>

</main>

#### view-ipo.component.ts

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

import { IPO } from 'src/app/models/ipo';

import { IposService } from 'src/app/services/ipos.service';

@Component({

selector: 'app-view-ipo',

templateUrl: './view-ipo.component.html',

styleUrls: ['./view-ipo.component.css']

})

export class ViewIpoComponent implements OnInit {

ipos: IPO[];

constructor(private iposService: IposService) {}

ngOnInit() {

this.iposService.getAllIPOs().subscribe(data => {

this.ipos = data;

})

}

}

### ViewSectors Component

#### view-sectors.component.html

<main class="container">

<div class="display-4 text-center my-4">Sectors</div>

<div class="table-responsive">

<table class="table table-stripped table-hover bg-light text-center">

<thead class="thead-light">

<tr>

<th>ID</th>

<th>Sector name</th>

<th>Brief</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let sector of sectors">

<td> {{sector.id}} </td>

<td> {{sector.name}} </td>

<td> {{sector.brief}} </td>

</tr>

</tbody>

</table>

</div>

<button type="button" class="btn btn-primary rounded-circle mx-auto add-btn" (click)="gotoNew()">

<fa-icon [icon]="faPlus" class="h2"></fa-icon>

</button>

</main>

#### view-sectors.component.ts

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

import { Sector } from "src/app/models/sector";

import { SectorsService } from "src/app/services/sectors.service";

@Component({

selector: "app-view-sectors",

templateUrl: "./view-sectors.component.html",

styleUrls: ["./view-sectors.component.css"]

})

export class ViewSectorsComponent implements OnInit {

sectors: Sector[];

constructor(private sectorsService: SectorsService) {}

ngOnInit() {

this.sectorsService.getAllSectors().subscribe(data => {

this.sectors = data;

});

}

}

# Services

## User Service

### user.service.ts

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

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

import { Observable } from "rxjs";

import { User } from "../models/users";

@Injectable({

providedIn: "root"

})

export class UserService {

url = "http://localhost:3000/users/";

constructor(private httpClient: HttpClient) {}

getAllUsers(): Observable<User[]> {

return this.httpClient.get<User[]>(this.url);

}

getUserById(id: number): Observable<User> {

return this.httpClient.get<User>(this.url + id);

}

registerUser(user: User): Observable<User> {

return this.httpClient.post<User>(this.url, user);

}

deleteUser(id: number): Observable<User> {

return this.httpClient.delete<User>(this.url + id);

}

updateUser(user: User): Observable<User> {

return this.httpClient.put<User>(this.url + user.id, user);

}

isAdmin(): boolean {

if (

localStorage.getItem("userType") == "admin" ||

sessionStorage.getItem("userType") == "admin"

) {

return true;

} else {

return false;

}

}

isUser(): boolean {

if (

localStorage.getItem("userType") == "user" ||

sessionStorage.getItem("userType") == "user"

) {

return true;

} else {

return false;

}

}

}

## Companies Service

### companies.service.ts

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

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

import { Company } from "../models/company";

import { Observable } from "rxjs";

@Injectable({

providedIn: "root"

})

export class CompaniesService {

url = "http://localhost:5100/companies/";

constructor(private httpClient: HttpClient) {}

getAllCompanies(): Observable<Company[]> {

return this.httpClient.get<Company[]>(this.url);

}

getCompayById(id: number): Observable<Company> {

return this.httpClient.get<Company>(this.url + id);

}

addCompany(company: Company): Observable<Company> {

return this.httpClient.post<Company>(this.url, company);

}

deleteCompany(id: number): Observable<Company> {

return this.httpClient.delete<Company>(this.url + id);

}

updateCompany(company: Company): Observable<Company> {

return this.httpClient.put<Company>(this.url + company.id, company);

}

}

## IPOs Service

### ipos.service.ts

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

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

import { Observable } from "rxjs";

import { IPO } from "../models/ipo";

@Injectable({

providedIn: "root"

})

export class IposService {

url = "http://localhost:5200/ipos/";

constructor(private httpClient: HttpClient) {}

getAllIPOs(): Observable<IPO[]> {

return this.httpClient.get<IPO[]>(this.url);

}

getIPOById(id: number): Observable<IPO> {

return this.httpClient.get<IPO>(this.url + id);

}

addNewIPO(ipo: IPO): Observable<IPO> {

return this.httpClient.post<IPO>(this.url, ipo);

}

updateIPO(ipo: IPO): Observable<IPO> {

return this.httpClient.put<IPO>(this.url + ipo.id, ipo);

}

deleteIPO(id: number): Observable<IPO> {

return this.httpClient.delete<IPO>(this.url + id);

}

}

## Sectors Service

### sectors.service.ts

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

import { Observable } from "rxjs";

import { Sector } from "../models/sector";

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

@Injectable({

providedIn: "root"

})

export class SectorsService {

url = "http://localhost:5300/sectors/";

constructor(private httpClient: HttpClient) {}

getAllSectors(): Observable<Sector[]> {

return this.httpClient.get<Sector[]>(this.url);

}

getSectorById(id: number): Observable<Sector> {

return this.httpClient.get<Sector>(this.url + id);

}

addNewSector(Sector: Sector): Observable<Sector> {

return this.httpClient.post<Sector>(this.url, Sector);

}

updateSector(Sector: Sector): Observable<Sector> {

return this.httpClient.put<Sector>(this.url + Sector.id, Sector);

}

deleteSector(id: number): Observable<Sector> {

return this.httpClient.delete<Sector>(this.url + id);

}

}

## StockExchanges Service

### stock-exchanges.service.ts

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

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

import { Observable } from "rxjs";

import { StockExchange } from "../models/stockExchange";

@Injectable({

providedIn: "root"

})

export class StockExchangesService {

url = "http://localhost:5400/stockexchanges/";

constructor(private httpClient: HttpClient) {}

getAllExchanges(): Observable<StockExchange[]> {

return this.httpClient.get<StockExchange[]>(this.url);

}

getExchangeById(id: number): Observable<StockExchange> {

return this.httpClient.get<StockExchange>(this.url + id);

}

addStockExchange(stockEx: StockExchange): Observable<StockExchange> {

return this.httpClient.post<StockExchange>(this.url, stockEx);

}

updateStockExchange(stockEx: StockExchange): Observable<StockExchange> {

return this.httpClient.put<StockExchange>(

this.url + stockEx.id,

stockEx

);

}

deleteStockExchange(stockEx: StockExchange): Observable<StockExchange> {

return this.httpClient.delete<StockExchange>(this.url + stockEx.id);

}

}

# Guards

## Login Guard

### login.guard.ts

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

import {

CanActivate,

ActivatedRouteSnapshot,

RouterStateSnapshot,

UrlTree,

Router

} from "@angular/router";

import { Observable } from "rxjs";

import { UserService } from '../services/user.service';

declare var $:any;

@Injectable({

providedIn: "root"

})

export class LoginGuard implements CanActivate {

constructor(private userService: UserService, private router: Router) {}

canActivate(

next: ActivatedRouteSnapshot,

state: RouterStateSnapshot

):

| Observable<boolean | UrlTree>

| Promise<boolean | UrlTree>

| boolean

| UrlTree {

if (this.userService.isAdmin() || this.userService.isUser()) {

return true;

} else {

alert("You need to login first!");

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

$("#login-modal").modal("show");

return false;

}

}

}

## AdminAuth Guard

### admin-auth.guard.ts

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

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

import { Observable } from 'rxjs';

import { UserService } from '../services/user.service';

declare var $:any;

@Injectable({

providedIn: 'root'

})

export class AdminAuthGuard implements CanActivate {

constructor(private userService: UserService, private router: Router) {}

canActivate(

next: ActivatedRouteSnapshot,

state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {

if (this.userService.isAdmin()) {

return true;

}

else if(this.userService.isUser()){

alert("You do not have access to this page!\nOnly admin can access");

this.router.navigate(['landing']);

return false;

}

else {

alert("You need to login first!");

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

$("#login-modal").modal("show");

return false;

}

}

}

## UserAuth Guard

### user-auth.guard.ts

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

import {

CanActivate,

ActivatedRouteSnapshot,

RouterStateSnapshot,

UrlTree,

Router

} from "@angular/router";

import { Observable } from "rxjs";

import { UserService } from "../services/user.service";

declare var $: any;

@Injectable({

providedIn: "root"

})

export class UserAuthGuard implements CanActivate {

constructor(private userService: UserService, private router: Router) {}

canActivate(

next: ActivatedRouteSnapshot,

state: RouterStateSnapshot

):

| Observable<boolean | UrlTree>

| Promise<boolean | UrlTree>

| boolean

| UrlTree {

if (this.userService.isUser()) {

return true;

}

else if (this.userService.isAdmin()) {

alert("You do not have access to this page!");

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

return false;

}

else {

alert("You need to login first!");

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

$("#login-modal").modal("show");

return false;

}

}

}

## NoLogin Guard (Does not allow loggin users to acces some pages)

### no-login.guard.ts

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

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

import { Observable } from 'rxjs';

import { UserService } from '../services/user.service';

@Injectable({

providedIn: 'root'

})

export class NoLoginGuard implements CanActivate {

constructor(private userService: UserService, private router: Router) {}

canActivate(

next: ActivatedRouteSnapshot,

state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {

if (!this.userService.isAdmin() && !this.userService.isUser()) {

return true;

}

else {

this.router.navigate(['landing']);

return false;

}

}

}

# styles.css

body {

background-image: url("./assets/background.jpg");

background-repeat: no-repeat;

background-size: cover;

background-position-y: top;

background-position-x: left;

width: 100%;

background-attachment: fixed;

}

footer {

min-height: 2rem;

}

main {

min-height: 75vh;

/\* margin-top: 5rem; \*/

}

#logo {

max-height: 3rem;

-moz-user-select: none;

-webkit-user-drag: none;

}

#logo-foot {

max-height: 4rem;

}

.undraggable,

.jumbotron,

a.card {

-moz-user-select: none;

-webkit-user-drag: none;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

.adv-imgs{

width: 15vw;

}

.social-icon {

font-size: 1.5rem;

}

.hover-enlarge {

transition: transform ease-in 0.2s;

}

.hover-enlarge:hover {

transform: scale(1.05, 1.05);

}

a.card,

a.crad:hover {

color: inherit;

text-decoration: none;

}

.table > tbody > tr > td {

vertical-align: middle !important;

}

.noresize {

resize: none;

}

.add-btn {

height: 10vh;

width: 10vh;

right: 2rem;

bottom: 5rem;

position: fixed;

text-align: center;

line-height: 28px;

text-decoration: none;

text-align: center;

z-index: 1;

}

@media (max-width: 991px) {

.company-logo,

.exchange-logo {

width: 20rem;

margin-left: auto;

margin-bottom: 2rem;

/\* display: none; \*/

}

}

@media (max-width: 768px) {

.md-only {

display: none;

}

}

@media (min-width: 768px) {

.not-md {

display: none;

}

}

.limit-text {

display: block;

width: 15rem;

overflow: hidden;

white-space: nowrap;

text-overflow: ellipsis;

}

.text-shadow {

text-shadow: black 4px 4px 4px;

}

.btn:disabled {

cursor: context-menu;

}

input.ng-invalid.ng-touched {

border: 1px solid red;

}

# json files

## companies.json

{

"companies": [

]

}

## ipos.json

{

"ipos": [

]

}

## sectors.json

{

"sectors": [

]

}

## stockexchanges.json

{

"stockexchanges": [

]

}

## users.json

{

"users": [

{

"id": 1,

"username": "admin",

"email": "admin@stockview.com",

"password": "admin",

"rePassword": "admin",

"phoneNo": 1234,

"isAdmin": true,

"confirmed": true

}

]

}