**MODELS:**

User.ts

export class User{

  id:number;

  username:string;

  password:string;

  admin:boolean;

  email:string;

  phone:number;

  confirmed:boolean;

  code:number;

  dp:string;

}

Company.ts

export class Company{

    id:number;

    name: string;

    turnover:number;

    ceo:string;

    board\_of\_directors:string[];

    listed\_in\_stock\_exchanges:string[];

    sector:string;

    brief:string;

    stock\_code:string;

    activated:boolean;

}

Stock-exchange.ts

export class StockExchange{

  id:number;

  name:string;

  brief:string;

  address:string;

  remarks:string;

}

Ipo.ts

export class IPO{

    id:number;

    company\_name:string;

    stock\_exchange:string;

    price\_per\_share:number;

    total\_shares:number;

    open\_date:string;

    open\_time:string;

    remarks:string;

}

Sector.ts

export class StockPrice{

  id:number;

  name:string;

  brief:string;

}

Stock-price.ts

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

export class StockPrice{

  company\_id:number;

  stock\_exchange:string;

  current\_price:number;

  date:Date;

  time:Time;

}

**SERVICES:**

User.service.ts:

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

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

import { Observable } from 'rxjs';

import { User } from 'src/models/user';

@Injectable({

  providedIn: 'root'

})

export class UserService {

  httpUrl="http://localhost:3001/users/";

  constructor(private httpClient:HttpClient) { }

  saveUser(user:User):Observable<User>{

    return this.httpClient.post<User>(this.httpUrl, user);

  }

  removeUser(id:number):Observable<User>{

    return this.httpClient.delete<User>(this.httpUrl + id);

  }

  getUserById(id:number):Observable<User>{

    return this.httpClient.get<User>(this.httpUrl + id);

  }

  getAllUsers():Observable<User[]>{

    return this.httpClient.get<User[]>(this.httpUrl);

  }

  updateUser(user:User):Observable<User>{

    return this.httpClient.put<User>(this.httpUrl + user.id , user);

  }

  isLoggedIn(){

    let userId=sessionStorage.getItem("userId");

    if(userId==null){

      return false;

    }else{

      return true;

    }

  }

  isAdmin():boolean{

    let admin:boolean;

    let userId=sessionStorage.getItem("userId");

    if(userId==null){

      this.getUserById(+userId).subscribe(data=>{

        admin=data.admin;

      });

      if(admin)

        return true;

      else

        return false;

    }else{

      return false;

    }

  }

  isActivated():boolean{

    let activated:boolean;

    let userId=sessionStorage.getItem("userId");

    if(userId==null){

      this.getUserById(+userId).subscribe(data=>{

        activated=data.admin;

      });

      if(activated)

        return true;

      else

        return false;

    }else{

      return false;

    }

  }

}

Company.service.ts

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

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

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

import { Observable } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class CompanyService {

  httpUrl="http://localhost:3000/companies/";

  constructor(private httpClient:HttpClient) { }

  saveCompany(company:Company):Observable<Company>{

    return this.httpClient.post<Company>(this.httpUrl, company);

  }

  removeCompany(id:number):Observable<Company>{

    return this.httpClient.delete<Company>(this.httpUrl + id);

  }

  getAllCompanies():Observable<Company[]>{

    return this.httpClient.get<Company[]>(this.httpUrl);

  }

  getCompanyById(id:number):Observable<Company>{

    return this.httpClient.get<Company>(this.httpUrl + id);

  }

  updateCompany(company:Company):Observable<Company>{

    return this.httpClient.put<Company>(this.httpUrl + company.id,company);

  }

}

Stock-exchange.service.ts:

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

import { StockExchange } from 'src/models/stock-exchange';

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

import { Observable } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class StockExchangeService {

  httpUrl="http://localhost:3002/stock\_exchanges/";

  constructor(private httpClient:HttpClient) { }

  saveStockExchange(stockExchange:StockExchange):Observable<StockExchange>{

    return this.httpClient.post<StockExchange>(this.httpUrl,stockExchange);

  }

  getAllStockExchanges():Observable<StockExchange[]>{

    return this.httpClient.get<StockExchange[]>(this.httpUrl);

  }

  getStockExchangeById(id:number):Observable<StockExchange>{

    return this.httpClient.get<StockExchange>(this.httpUrl + id);

  }

  updateStockExchange(stockExchange:StockExchange):Observable<StockExchange>{

    return this.httpClient.put<StockExchange>(this.httpUrl + stockExchange.id, stockExchange);

  }

}

Ipo.service.ts:

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

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

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

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

import { IPO } from 'src/models/ipo';

import { Observable } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class IPOService {

  httpUrl="http://localhost:3003/ipos/";

  constructor(private httpClient:HttpClient) { }

  saveIpo(ipo:IPO):Observable<IPO>{

    return this.httpClient.post<IPO>(this.httpUrl, ipo);

  }

  removeIPO(id:number):Observable<IPO>{

    return this.httpClient.delete<IPO>(this.httpUrl + id);

  }

  updateIPO(ipo:IPO):Observable<IPO>{

    return this.httpClient.put<IPO>(this.httpUrl + ipo.id, ipo);

  }

  getAllIPOs():Observable<IPO[]>{

    return this.httpClient.get<IPO[]>(this.httpUrl);

  }

  getIPOById(id:number):Observable<IPO>{

    return this.httpClient.get<IPO>(this.httpUrl + id);

  }

}

Stock-price.service.ts:

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

@Injectable({

  providedIn: 'root'

})

export class StockPriceService {

  constructor() { }

}

Sector.service.ts:

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

@Injectable({

  providedIn: 'root'

})

export class SectorService {

  constructor() { }

}

**COMPONENTS:**

Index.html:

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>Project</title>

  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="icon" type="image/x-icon" href="favicon.ico">

  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" integrity="sha384-WskhaSGFgHYWDcbwN70/dfYBj47jz9qbsMId/iRN3ewGhXQFZCSftd1LZCfmhktB"

    crossorigin="anonymous">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

</head>

<body class="pt-5">

    <app-root></app-root>

  <script src="https://kit.fontawesome.com/b1d2774758.js" crossorigin="anonymous"></script>

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"

    crossorigin="anonymous"></script>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"

    crossorigin="anonymous"></script>

  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js" integrity="sha384-smHYKdLADwkXOn1EmN1qk/HfnUcbVRZyYmZ4qpPea6sjB/pTJ0euyQp0Mk8ck+5T"

    crossorigin="anonymous"></script>

</body>

</html>

Styles.css:

.avatar {

  vertical-align: middle;

  width: 50px;

  height: 50px;

}

main{

  min-height: 85vh;

}

.border-curved{

  border-radius:20px;

  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

}

.border-curved:hover{

  box-shadow: 0 4px 8px 0 grey, 0 6px 20px 0 grey;

}

.my-card:hover{

  cursor: pointer;

}

.my-card>img{

  height: 300px;

  width: 300px;

  border-radius: 50px;

}

.stroke{

  -webkit-text-stroke: 1px black;

    -webkit-text-fill-color: white;

}

.carousel-item:hover{

  cursor: pointer;

}

.responsive-footer{

  font-size: 2vw;

}

.responsive-logo{

  font-size:2vw;

}

body {

  background-image: url("./assets/images/background3.jpg");

  background-repeat: no-repeat;

  background-attachment: fixed;

  background-size:cover;

}

.h2{

  -webkit-text-stroke: 1px black;

    -webkit-text-fill-color: white;

}

 .table>tbody>tr>td{

  vertical-align: middle !important;

}

.hover:hover{

  cursor: pointer;

}

App.component.ts:

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

import { ErrorService } from './error.service';

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

}

App.component.html:

<header>

    <nav class="navbar navbar-expand-md navbar-light bg-light fixed-top">

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

            <img class="avatar" src="../assets/images/syed-logo.svg" alt="Stock Logo">

        </a>

        <a routerLink="" class="h3 nav-link text-dark">StockMarket</a>

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

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

        </button>

        <div class="collapse navbar-collapse justify-content-between" id="collNav">

            <ul class="navbar-nav">

                <li class="nav-item active">

                    <a routerLink="" class="nav-link text-center">Home</a>

                </li>

                <li class="nav-item">

                    <a routerLink="login" class="nav-link text-center">Login</a>

                </li>

                <li class="nav-item">

                    <a routerLink="register" class="nav-link text-center">Register</a>

                </li>

                <li class="nav-item">

                    <a routerLink="admin" class="nav-link text-center">Admin</a>

                </li>

                <li class="nav-item">

                    <a routerLink="user" class="nav-link text-danger text-center">User</a>

                </li>

                <li class="nav-item">

                    <a routerLink="" class="nav-link text-danger text-center">Logout</a>

                </li>

            </ul>

        </div>

    </nav>

</header>

<main>

    <div id="body" class="container my-lg-5 my-md-3 my-2">

        <!-- <div class="row">

            <ul \*ngFor="let error of errors">

                <li class="text-center text-danger bg-light">{{error}}</li>

            </ul>

          </div> -->

        <router-outlet></router-outlet>

    </div>

</main>

<footer class="page-footer bg-secondary p-3 container-fluid mt-5">

  <div class="row">

      <div class="col-3 text-center responsive-logo">

          <i class="fas fa-copyright fa-2x text-light mx-2"></i>

      </div>

      <div class="col-6 text-center ">

          <p class="font-italic text-light h5 responsive-footer">Created by: Syed Ehsan Ali Razvi</p>

      </div>

      <div class="col">

          <a href="">

          </a>

          <a href="">

          </a>

          <a href="">

          </a>

          <a href="">

          </a>

      </div>

  </div>

</footer>

App-routing.module.ts:

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

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

import { RegisterFormComponent } from './register-form/register-form.component';

import { LoginFormComponent } from './login-form/login-form.component';

import { LandingPageComponent } from './landing-page/landing-page.component';

import { AdminPageComponent } from './admin-page/admin-page.component';

import { UserPageComponent } from './user-page/user-page.component';

import { DisplayUsersComponent } from './display-users/display-users.component';

import { AddCompanyComponent } from './add-company/add-company.component';

import { AddStockExchangeComponent } from './add-stock-exchange/add-stock-exchange.component';

import { DeactivateCompanyComponent } from './deactivate-company/deactivate-company.component';

import { ResetComponent } from './reset/reset.component';

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

import { ImportDataComponent } from './import-data/import-data.component';

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

import { ManageStockExchangeComponent } from './manage-stock-exchange/manage-stock-exchange.component';

import { AddIpoComponent } from './add-ipo/add-ipo.component';

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

import { UpdateUserComponent } from './update-user/update-user.component';

import { ChangePasswordComponent } from './change-password/change-password.component';

import { ResetCodeComponent } from './reset-code/reset-code.component';

import { ResetPasswordComponent } from './reset-password/reset-password.component';

import { UserProfileComponent } from './user-profile/user-profile.component';

import { UploadDpComponent } from './upload-dp/upload-dp.component';

import { UpdateStockExchangeComponent } from './update-stock-exchange/update-stock-exchange.component';

import { DisplayIposComponent } from './display-ipos/display-ipos.component';

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

import { ViewIposComponent } from './view-ipos/view-ipos.component';

const routes: Routes = [

  {path:'login',component:LoginFormComponent},

  {path:'register',component:RegisterFormComponent},

  {path:'admin',component:AdminPageComponent,canActivate:[AuthGuard]},

  {path:'user',component:UserPageComponent,canActivate:[AuthGuard]},

  {path:'display-users',component:DisplayUsersComponent,canActivate:[AuthGuard]},

  {path:'add-company' ,component: AddCompanyComponent,canActivate:[AuthGuard]},

  {path:'add-stock-exchange' ,component: AddStockExchangeComponent,canActivate:[AuthGuard]},

  {path:'add-ipo' ,component: AddIpoComponent,canActivate:[AuthGuard]},

  {path:'deactivate-company' ,component: DeactivateCompanyComponent,canActivate:[AuthGuard]},

  {path:'reset' ,component: ResetComponent },

  {path:'update-ipo' ,component: UpdateIpoComponent,canActivate:[AuthGuard]},

  {path:'update-company' ,component: UpdateCompanyComponent,canActivate:[AuthGuard]},

  {path:'import-data' ,component: ImportDataComponent,canActivate:[AuthGuard]},

  {path:'manage-company' ,component: ManageCompanyComponent,canActivate:[AuthGuard]},

  {path:'manage-stock-exchange' ,component: ManageStockExchangeComponent,canActivate:[AuthGuard]},

  {path:'update-user',component: UpdateUserComponent,canActivate:[AuthGuard]},

  {path:'change-password',component: ChangePasswordComponent,canActivate:[AuthGuard]},

  {path:'reset-code',component:ResetCodeComponent},

  {path:'reset-password',component:ResetPasswordComponent},

  {path:'user-profile',component:UserProfileComponent},

  {path:'upload-dp',component:UploadDpComponent,canActivate:[AuthGuard]},

  {path:'update-stock-exchange',component:UpdateStockExchangeComponent,canActivate:[AuthGuard]},

  {path:'display-ipos',component:DisplayIposComponent,canActivate:[AuthGuard]},

  {path:'view-ipos',component:ViewIposComponent,canActivate:[AuthGuard]},

  {path:'\*\*',component:LandingPageComponent},

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

add-company.component.ts:

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

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

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

import { CompanyService } from 'src/services/company.service';

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

@Component({

  selector: 'app-add-company',

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

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

})

export class AddCompanyComponent implements OnInit {

  registerCompany:FormGroup;

  constructor(private formBuilder:FormBuilder,private companyService:CompanyService,private router:Router) { }

  addCompany(){

    this.companyService.saveCompany(this.registerCompany.value).subscribe(data=>{

      alert("company added successfully");

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

    });

  }

  ngOnInit() {

    this.registerCompany=this.formBuilder.group({

      id:[],

      name:['',Validators.required],

      turnover:['',Validators.required],

      ceo:['',Validators.required],

      board\_of\_directors:['',Validators.required],

      listed\_in\_stock\_exchanges:['',Validators.required],

      sector:['',Validators.required],

      brief:['',Validators.required],

      stock\_code:['',Validators.required],

      activated:['true',Validators.required]

    });

  }

}

Add-company.component.html:

<div class="row">

  <div class="col-3"></div>

  <div class="col-6">

    <form [formGroup]="registerCompany" class="form-group border p-5 border-curved mt-5">

      <p class="h2 text-muted text-uppercase text-center mb-4">Company</p>

      <label for="name">Company Name:</label>

      <input type="text" name="name" formControlName="name" class="form-control" placeholder="Name of Company">

      <label for="turnover">Turnover:</label>

      <input type="text" name="turnover" formControlName="turnover" class="form-control" placeholder="Company Turnover">

      <label for="ceo">CEO:</label>

      <input type="text" name="ceo" formControlName="ceo" class="form-control" placeholder="CEO of Company">

      <label for="board-of-directors">Board of Director:</label>

      <input type="text" name="board-of-directors" formControlName="board\_of\_directors" class="form-control mb-4" placeholder="Board of Director of Company">

      <label for="listed-in-stock-exchanges">Listed in Stock Exchanges:</label><br>

      <small class="text-info">Hold [Ctrl] to Select Multiple Stock Exchanges</small>

      <select class="form-control" name="listed-in-stock-exchanges" formControlName="listed\_in\_stock\_exchanges" multiple size="2">

          <option value="bse">BSE</option>

          <option value="nse">NSE</option>

      </select>

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

      <input type="text" name="sector" formControlName="sector" class="form-control" placeholder="Name of Sector">

      <label for="brief">Brief:</label>

      <textarea class="form-control" name="brief" formControlName="brief" id="" cols="30" rows="3" placeholder="Brief Information about Company"></textarea>

      <label for="stock\_code">Stock Code:</label>

      <input type="text" name="stock\_code" formControlName="stock\_code" class="form-control mb-4" placeholder="Unique Stock Code">

      <button type="submit" class="btn btn-outline-success text-dark text-center form-control" (click)="addCompany()" [disabled]="!registerCompany.valid">Add Company</button>

    </form>

  </div>

</div>

Add-ipo.component.ts:

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

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

import { IPOService } from 'src/services/ipo.service';

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

@Component({

  selector: 'app-add-ipo',

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

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

})

export class AddIpoComponent implements OnInit {

  constructor(private formBuilder:FormBuilder,private ipoService:IPOService, private router:Router) { }

  addIPO:FormGroup;

  ngOnInit() {

    this.addIPO=this.formBuilder.group({

      id:[],

      company\_name:['apple',Validators.required],

      stock\_exchange:['bse',Validators.required],

      price\_per\_share:['',Validators.required],

      total\_shares:['',Validators.required],

      open\_date:['',Validators.required],

      open\_time:['',Validators.required],

      remarks:['']

    });

  }

  addIpo(){

    this.ipoService.saveIpo(this.addIPO.value).subscribe(data=>{

      alert('IPO Inserted Successfully.');

      this.router.navigate(['/display-ipos']);

    })

  }

}

Add-ipo.component.html:

<div>

<div class="row">

  <div class="col-lg-3 col-md-2 col-sm-1"></div>

      <div class="col-lg-6 col-md-8 col-sm-10">

    <form [formGroup]="addIPO" class="form-group border p-3 p-lg-5 border-curved mt-5" (ngSubmit)="addIpo()">

      <p class="h2 text-muted text-uppercase text-center mb-3 ">IPO</p>

      <label for="username">Company Name:</label>

      <select name="company" id="" class="form-control" formControlName="company\_name">

          <option value="apple">Apple</option>

          <option value="microsoft">Microsoft</option>

          <option value="google">Google</option>

      </select>

      <label for="username">Stock Exchange:</label>

      <select class="form-control" name="stock-exchange" formControlName="stock\_exchange">

          <option value="bse" selected>BSE</option>

          <option value="nse">NSE</option>

      </select>

      <label for="username">Price per Share:</label>

      <div class="input-group">

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

              <i class="fas fa-rupee-sign input-group-text"></i>

          </div>

          <input type="number" class="form-control" placeholder="Price in Rupees" formControlName="price\_per\_share">

      </div>

      <label for="username">No. of Shares:</label>

      <input type="number" class="form-control" placeholder="Total Number of Shares" formControlName="total\_shares">

      <label for="username">Open Date:</label><br>

      <input class="form-control" type="date" name="date" id="" formControlName="open\_date">

      <label for="username">Open Time:</label><br>

      <input class="form-control" type="time" name="time" id="" formControlName="open\_time">

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

      <textarea class="form-control mb-4" name="brief" id="" cols="30" rows="3" placeholder="Remarks about this IPO" formControlName="remarks"></textarea>

      <button class="btn btn-outline-success text-dark text-center form-control" [disabled]="!addIPO.valid">Add IPO</button>

    </form>

  </div>

</div>

</div>

Add-stock-exchange.component.ts:

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

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

import { StockExchangeService } from 'src/services/stock-exchange.service';

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

@Component({

  selector: 'app-add-stock-exchange',

  templateUrl: './add-stock-exchange.component.html',

  styleUrls: ['./add-stock-exchange.component.css']

})

export class AddStockExchangeComponent implements OnInit {

  constructor(private formBuilder:FormBuilder,private seService:StockExchangeService,private router:Router) { }

  addSE:FormGroup;

  ngOnInit() {

    this.addSE=this.formBuilder.group({

      id:[],

      name:['',Validators.required],

      brief:[''],

      address:['',Validators.required],

      remarks:['']

    });

  }

  addSe(){

    this.seService.saveStockExchange(this.addSE.value).subscribe(data=>{

      alert('Stock Exchange Inserted Successfully.');

      this.router.navigate(['/manage-stock-exchange']);

    });

  }

}

Add-stock-exchange.component.html:

<div class="row">

  <div class="col-3"></div>

  <div class="col-6">

    <form [formGroup]="addSE" (ngSubmit)="addSe()" class="form-group border p-5 border-curved mt-5">

      <p class="h2 text-muted text-uppercase text-center mb-4">Stock Exchange</p>

      <label for="username">Stock Exchange Name:</label>

      <input type="text" class="form-control" placeholder="Name of Stock Exchange" formControlName="name">

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

      <textarea class="form-control" placeholder="Brief About Stock Exchange" rows="3" formControlName="brief"></textarea>

      <label for="username">Contact Address:</label>

      <input type="text" class="form-control" placeholder="Contact Address of Stock Exchange" formControlName="address">

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

      <textarea class="form-control mb-4" placeholder="Remarks About Stock Exchange" rows="3" formControlName="remarks"></textarea>

      <button class="btn btn-outline-success text-dark text-center form-control" [disabled]="!addSE.valid">Add Stock Exchange</button>

    </form>

  </div>

</div>

Admin-page.component.ts:

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

@Component({

  selector: 'app-admin-page',

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

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

})

export class AdminPageComponent implements OnInit {

  constructor() { }

  ngOnInit() {

  }

}

Admin-page.component.html:

<div>

  <div class="row mb-5 pt-5">

      <div class="col-md-6 my-3">

          <div class="card my-card border border-curved text-center h-100" routerLink="/import-data">

                <div class="card-body">

                    <h4 class="card-title display-4">Import Data</h4>

                </div>

          </div>

      </div>

      <div class="col-md-6 my-3">

          <div class="card my-card border border-curved text-center h-100" routerLink="/manage-company">

              <div class="card-body">

                  <h4 class="card-title display-4">Manage Companies</h4>

              </div>

          </div>

      </div>

      <div class="col-md-6 my-3">

          <div class="card my-card border border-curved text-center h-100" routerLink="/manage-stock-exchange">

              <div class="card-body">

                  <h4 class="card-title display-4">Manage Exchanges</h4>

              </div>

          </div>

      </div>

      <div class="col-md-6 my-3">

          <div class="card my-card border border-curved text-center h-100" routerLink="/display-ipos">

              <div class="card-body">

                  <h4 class="card-title display-4">Manage IPOs</h4>

              </div>

          </div>

      </div>

      <div class="col-md-6 my-3">

        <div class="card my-card border border-curved text-center h-100" routerLink="/display-users">

            <div class="card-body">

                <h4 class="card-title display-4">Manage Users</h4>

            </div>

        </div>

    </div>

  </div>

</div>

Change-password.component.ts:

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

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

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

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

import { User } from 'src/models/user';

@Component({

  selector: 'app-change-password',

  templateUrl: './change-password.component.html',

  styleUrls: ['./change-password.component.css']

})

export class ChangePasswordComponent implements OnInit {

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

  updatePassword:FormGroup;

  user:User;

  ngOnInit() {

    this.updatePassword=this.formBuilder.group({

      old\_password:['',Validators.required],

      password:['',Validators.required],

      confirm\_password:['',Validators.required],

    });

    const id=localStorage.getItem('userId');

    this.userService.getUserById(+id).subscribe(data=>{

      this.user=data;

    })

  }

  updateThisPassword(){

      if(this.user.password === this.updatePassword.controls['old\_password'].value){

        if((this.updatePassword.controls['password'].value === this.updatePassword.controls['confirm\_password'].value)){

          this.user.password=this.updatePassword.controls.password.value;

          this.userService.updateUser(this.user).subscribe(u=>{

          this.router.navigate(['display-users']);

          alert('Your Password is Changed.');

        });

      }else{

        alert('Your Passwords doesn\'t match');

      }

    }else{

      alert('Your Old Password is Incorrect!');

    }

  }

}

Change-password.component.html:

<div>

  <div class="row">

      <div class="col-lg-3 col-md-2 col-sm-1"></div>

      <div class="col-lg-6 col-md-8 col-sm-10">

        <form [formGroup]="updatePassword" class="form-group border p-3 p-lg-5 border-curved mt-5">

          <p class="h2 text-muted text-uppercase text-center">Change Password</p>

          <!-- <input type="number" name="id" formControlName="id" placeholder="Id" class="form-control" hidden> -->

          <label for="old\_password">Old Password:</label>

          <input name="old\_password" formControlName="old\_password" type="password" class="form-control" placeholder="Your Old Password">

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

          <input name="password" formControlName="password" type="password" class="form-control" placeholder="Your New Password">

          <label for="confirm\_password">Confirm Password:</label>

          <input name="confirm\_password" formControlName="confirm\_password" type="password" class="form-control" placeholder="Confirm Your Password">

          <button class="btn btn-outline-warning text-dark text-center form-control my-3" (click)="updateThisPassword()" [disabled]="!updatePassword.valid">Change Password</button>

        </form>

      </div>

  </div>

</div>

Deactivate-company.component.ts:

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

@Component({

  selector: 'app-deactivate-company',

  templateUrl: './deactivate-company.component.html',

  styleUrls: ['./deactivate-company.component.css']

})

export class DeactivateCompanyComponent implements OnInit {

  constructor() { }

  ngOnInit() {

  }

}

Deactivate-company.component.html:

<div class="row">

  <div class="col-3"></div>

  <div class="col-6">

    <div class="form-group border p-5 border-curved mt-5">

      <p class="h2 text-muted text-uppercase text-center mb-4">Deactivate</p>

      <label for="company">Select Company:</label>

      <select class="form-control mb-4" name="company" id="">

          <option value="apple">Apple</option>

          <option value="microsoft">Microsoft</option>

          <option value="google">Google</option>

      </select>

      <button class="btn btn-danger text-white text-center form-control" data-toggle="modal" data-target="#deactivate">Deactivate</button>

  </div>

  </div>

</div>

<div id="deactivate" class="modal fade" tabindex="-1">

<div class="modal-dialog" role="document">

<div class="modal-content">

  <div class="modal-header">

    <h5 class="modal-title">Are You Sure You Want To Deactivate This Company?</h5>

    <button class="close" data-dismiss="modal">&times;</button>

  </div>

  <div class="modal-body">

      <button class="btn btn-block btn-danger text-white" onclick="location.href='manage-company.html'">Confirm</button>

  </div>

  <div class="modal-footer">

    <button class="btn btn-block btn-outline-warning" data-dismiss="modal">No</button>

  </div>

</div>

</div>

</div>

Display-ipos.component.ts:

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

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

import { IPOService } from 'src/services/ipo.service';

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

import { IPO } from 'src/models/ipo';

@Component({

  selector: 'app-display-ipos',

  templateUrl: './display-ipos.component.html',

  styleUrls: ['./display-ipos.component.css']

})

export class DisplayIposComponent implements OnInit {

  constructor(private ipoService:IPOService, private router:Router) { }

  ipos:IPO[];

  ngOnInit() {

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

      this.ipos=data;

    });

  }

  updateIpo(ipo:IPO){

    localStorage.removeItem('ipoId');

    localStorage.setItem('ipoId',ipo.id.toString());

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

  }

}

Display-ipos.component.html:

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

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

import { IPOService } from 'src/services/ipo.service';

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

import { IPO } from 'src/models/ipo';

@Component({

  selector: 'app-display-ipos',

  templateUrl: './display-ipos.component.html',

  styleUrls: ['./display-ipos.component.css']

})

export class DisplayIposComponent implements OnInit {

  constructor(private ipoService:IPOService, private router:Router) { }

  ipos:IPO[];

  ngOnInit() {

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

      this.ipos=data;

    });

  }

  updateIpo(ipo:IPO){

    localStorage.removeItem('ipoId');

    localStorage.setItem('ipoId',ipo.id.toString());

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

  }

}

Display-users.component.ts:

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

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

import { User } from 'src/models/user';

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

@Component({

  selector: 'app-display-users',

  templateUrl: './display-users.component.html',

  styleUrls: ['./display-users.component.css']

})

export class DisplayUsersComponent implements OnInit {

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

  users:User[];

  ngOnInit() {

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

      this.users=data;

    })

  }

  deleteUser(user:User){

    this.userService.removeUser(user.id).subscribe(data=>{

      alert("User "+user.username+" Deleted Successfully");

    });

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

  }

  updateUser(user:User){

    localStorage.removeItem('userId');

    localStorage.setItem('userId',user.id.toString());

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

  }

  viewProfile(user:User){

    localStorage.removeItem('userId');

    localStorage.setItem('userId',user.id.toString());

    this.router.navigate(['/user-profile']);

  }

}

Display-users.component.html:

<div class="row py-5 py-md-4 py-3">

  <div class="col mx-auto table-responsive">

      <table class="table border-curved p-1 p-md-3 p-lg-5">

          <caption style="caption-side: top;"><h1 class=" h2 text-center text-uppercase" >All Users</h1></caption>

  <thead>

    <th>Id</th>

    <th>Username</th>

    <th>Password</th>

    <th>Admin</th>

    <th>Email</th>

    <th>Phone</th>

    <th>Confirmed</th>

    <th>Actions</th>

  </thead>

  <tbody \*ngFor="let user of users">

    <td>{{user.id}}</td>

    <td>{{user.username}}</td>

    <td>{{user.password}}</td>

    <td>{{user.admin}}</td>

    <td>{{user.email}}</td>

    <td>{{user.phone}}</td>

    <td>{{user.confirmed}}</td>

    <td><button class="btn btn-danger mx-auto m-2" (click)="deleteUser(user)">Delete</button>

      <button class="btn btn-warning mx-auto m-2" (click)="updateUser(user)">Update</button>

      <button class="btn btn-info mx-auto m-2" (click)="viewProfile(user)">View Profile</button>

    </td>

  </tbody>

</table>

<button class="btn btn-block btn-outline-secondary" routerLink="/register">Add New User</button>

    </div>

</div>

Import-data.component.ts:

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

@Component({

  selector: 'app-import-data',

  templateUrl: './import-data.component.html',

  styleUrls: ['./import-data.component.css']

})

export class ImportDataComponent implements OnInit {

  constructor() { }

  ngOnInit() {

  }

}

Import-data.component.html:

<div class="row">

  <div class="col-lg-3 col-md-2 col-sm-1"></div>

  <div class="col-lg-6 col-md-8 col-sm-10">

    <form action="" class="form-group border p-3 p-lg-5 border-curved mt-5">

      <p class="h2 text-muted text-uppercase text-center mb-3">Import (Excel)</p>

      <input type="file" name="upload" id="" class="form-control-file mt-4">

      <small class="form-text text-muted mb-2" id="fileHelp">Max 3mb size</small>

      <p class="text-center w-100 mb-4"><a href="../../assets/files/sample\_stock\_data.xlsx">Download Sample File (.xlsx)</a></p>

      <button class="btn btn-outline-success text-dark text-center form-control">Import</button>

    </form>

  </div>

</div>

Landing-page.component.ts:

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

@Component({

  selector: 'app-landing-page',

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

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

})

export class LandingPageComponent implements OnInit {

  constructor() { }

  ngOnInit() {

  }

}

Landing-page.component.html:

  <div>

    <div class="row justify-content-center pt-3">

      <div class="col-lg-2 col-3 p-3">

        <img src="../../assets/images/syed-logo.svg" class="img-fluid welcome-image" alt="Stock Market Logo">

      </div>

    </div>

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

        <p class="welcome-text text-center">Welcome to <span class="font-weight-bold">Stock Market Charting</span></p>

      </div>

      <div class="col-sm-12 col-lg-9 m-auto">

        <div id="example3" class="carousel slide my-5" data-ride="carousel">

          <ul class="carousel-indicators">

            <li data-target="#example3" data-slide-to="0" class="active"></li>

            <li data-target="#example3" data-slide-to="1"></li>

          </ul>

          <div class="carousel-inner">

            <div class="carousel-item active">

              <img src="../../assets/images/stock1.jpg" alt="Stock 1" class="img-fluid" routerLink="/admin">

              <div class="carousel-caption d-none d-md-block">

                <h2 class="stroke h1">Admin Page</h2>

              </div>

            </div>

            <div class="carousel-item">

              <img src="../../assets/images/stock4.jpg" alt="Stock 2" class="img-fluid" routerLink="/user">

              <div class="carousel-caption d-none d-md-block">

                <h2 class="stroke h1">User Page</h2>

              </div>

            </div>

          </div>

          <a class="carousel-control-prev" href="#example3" data-slide="prev">

            <span class="carousel-control-prev-icon"></span>

            <span class="sr-only">Previous</span>

          </a>

          <a class="carousel-control-next" href="#example3" data-slide="next">

            <span class="carousel-control-next-icon"></span>

            <span class="sr-only">Next</span>

          </a>

        </div>

      </div>

  </div>

Landing-page.component.css:

.welcome-text{

    font-size: 5vw;

}

.welcome-image{

    width: 10vw;

}

Login-form.component.ts:

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

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

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

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

import { User } from 'src/models/user';

import { ErrorService } from '../error.service';

@Component({

  selector: 'app-login-form',

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

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

})

export class LoginFormComponent implements OnInit {

  loginUser:FormGroup;

  users:User[];

  constructor(private router:Router,private formbuilder:FormBuilder,private userService:UserService,private errorService:ErrorService) { }

  ngOnInit() {

    this.loginUser=this.formbuilder.group({

      username:['',Validators.required],

      password:['',Validators.required]

    });

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

      this.users=data;

    });

  }

  checkLoginUser(){

    let uname=this.loginUser.controls.username.value;

    let pwd=this.loginUser.controls.password.value;

    let flag:boolean;

    let current\_user:User;

    for(let user of this.users){

      if(user.username === uname){

        flag=true;

        current\_user=user;

      }

    }

    if(flag){

      if(current\_user.password == pwd){

        sessionStorage.removeItem("userId");

        sessionStorage.setItem("userId",current\_user.id.toString());

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

      }else{

        alert('Incorrect Password.');

      }

    }else{

      alert('User Not Found.');

    }

  }

}

Login-form.component.html:

<div>

  <!-- <div class="row" \*ngIf="isError()">

    <ul \*ngFor="let error of errors">

        <li class="text-center text-danger bg-light">hello world</li>

    </ul>

  </div> -->

  <div class="row">

      <div class="col-lg-3 col-md-2 col-sm-1"></div>

      <div class="col-lg-6 col-md-8 col-sm-10">

        <form [formGroup]="loginUser" class="form-group border p-3 p-lg-5 border-curved mt-5">

          <p class="h2 text-muted text-uppercase text-center mb-3 ">Login</p>

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

          <input type="text" formControlName="username" class="form-control" placeholder="Your Username">

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

          <input type="password" formControlName="password" class="form-control" placeholder="Your Password">

          <p class="text-center my-2 w-100"><a routerLink="/reset">Forget Password? Click Here to Reset.</a></p>

          <p class="text-center w-100 my-2"><a routerLink="/register">Don't Have an Account? Click Here to Create.</a></p>

          <button class="btn btn-outline-success text-dark text-center form-control" (click)="checkLoginUser()" [disabled]="!loginUser.valid">Login</button>

        </form>

      </div>

  </div>

</div>

Manage-company.component.ts:

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

import { CompanyService } from 'src/services/company.service';

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

import { User } from 'src/models/user';

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

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

@Component({

  selector: 'app-manage-company',

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

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

})

export class ManageCompanyComponent implements OnInit {

  constructor(private companyService:CompanyService,private router:Router) { }

  companies:Company[];

  updateFormGroup:FormGroup;

  selectedCompany:Company;

  ngOnInit() {

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

      this.companies=data;

    })

  }

  updateCompany(company:Company){

    localStorage.removeItem('companyId');

    localStorage.setItem('companyId',company.id.toString());

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

  }

  deactivateCompany(company:Company){

    localStorage.removeItem('companyId');

    localStorage.setItem('companyId',company.id.toString());

  }

  deactivateThisCompany(){

    const id=+localStorage.getItem('companyId');

    this.companyService.getCompanyById(id).subscribe(company=>{

      company.activated=false;

      this.companyService.updateCompany(company).subscribe(c=>{

        alert('Company Deactivated Successfully');

      })

    })

  }

}

Manage-company.component.html:

<div class="row py-5 py-md-4 py-3">

    <div class="col mx-auto table-responsive">

        <table class="table border-curved p-1 p-md-3 p-lg-5">

            <caption style="caption-side: top;"><h1 class="h2 text-center text-uppercase" >Companies</h1></caption>

            <thead>

                <tr>

                    <th>Id</th>

                    <th>Company Name</th>

                    <th>Turnover</th>

                    <th>CEO</th>

                    <th>Board Of Directors</th>

                    <th>Listed In Stock Exchanges</th>

                    <th>Sector</th>

                    <th>Brief</th>

                    <th>Stock Code</th>

                    <th>Actions</th>

                </tr>

            </thead>

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

                <tr>

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

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

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

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

                    <td>{{company.board\_of\_directors}}</td>

                    <td>{{company.listed\_in\_stock\_exchanges}}</td>

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

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

                    <td>{{company.stock\_code}}</td>

                    <td><button class="btn btn-warning mx-auto m-2" (click)="updateCompany(company)">Update</button><button class="btn btn-danger mx-auto m-2" data-toggle="modal" data-target="#deactivate" (click)="deactivateCompany(company)">Deactivate</button></td>

                </tr>

            </tbody>

        </table>

        <button class="btn btn-block btn-outline-secondary" routerLink="/add-company">Add New Company</button>

    </div>

</div>

<div id="deactivate" class="modal fade" tabindex="-1">

    <div class="modal-dialog" role="document">

    <div class="modal-content">

      <div class="modal-header">

        <h5 class="modal-title">Are You Sure You Want To Deactivate This Company?</h5>

        <button class="close" data-dismiss="modal">&times;</button>

      </div>

      <div class="modal-body">

          <button class="btn btn-block btn-danger text-white" (click)="deactivateThisCompany()" data-dismiss="modal">Confirm</button>

      </div>

      <div class="modal-footer">

        <button class="btn btn-block btn-outline-warning" data-dismiss="modal">No</button>

      </div>

    </div>

    </div>

    </div>

Manage-company.component.css:

.responsive-table{

    font-size: 1vw;

}

Manage-stock-exchange.component.ts:

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

import { CompanyService } from 'src/services/company.service';

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

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

import { StockExchangeService } from 'src/services/stock-exchange.service';

import { StockExchange } from 'src/models/stock-exchange';

@Component({

  selector: 'app-manage-stock-exchange',

  templateUrl: './manage-stock-exchange.component.html',

  styleUrls: ['./manage-stock-exchange.component.css']

})

export class ManageStockExchangeComponent implements OnInit {

  ses:StockExchange[];

  constructor(private seService:StockExchangeService,private router:Router) { }

  ngOnInit() {

    this.seService.getAllStockExchanges().subscribe(data=>{

      this.ses=data;

    });

  }

  updateStockExchange(se:StockExchange){

    localStorage.removeItem('seId');

    localStorage.setItem('seId',se.id.toString());

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

  }

}

Manage-stock-exchange.component.html:

    <div class="row py-5 py-md-4 py-3">

            <div class="col-lg-9 col-md-10 col-12 mx-auto table-responsive">

                <table class="table border-curved p-1 p-md-3 p-lg-5">

                    <caption style="caption-side: top;"><h1 class="h2 text-center text-uppercase" >Stock Exchanges</h1></caption>

                    <thead>

                        <tr>

                            <th>Id</th>

                            <th>Stock Exchange</th>

                            <th>Brief</th>

                            <th>Contact Address</th>

                            <th>Remarks</th>

                            <th>Actions</th>

                        </tr>

                    </thead>

                    <tbody>

                        <tr \*ngFor="let se of ses">

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

                            <td>{{se.stock\_exchange}}</td>

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

                            <td>{{se.address}}</td>

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

                            <td>

                                <button class="btn btn-warning m-3 mx-auto" (click)="updateStockExchange(se)">Update</button>

                            </td>

                        </tr>

                    </tbody>

                </table>

                <button class="btn btn-block btn-outline-secondary" routerLink="/add-stock-exchange">Add Stock Exchange</button>

            </div>

        </div>

Register-form.component.ts:

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

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

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

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

import { User } from 'src/models/user';

@Component({

  selector: 'app-register-form',

  templateUrl: './register-form.component.html',

  styleUrls: ['./register-form.component.css']

})

export class RegisterFormComponent implements OnInit {

  registerUser:FormGroup;

  users:User[];

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

  addUser(){

    let u=this.registerUser.controls.username.value;

    let e=this.registerUser.controls.email.value;

    let p=this.registerUser.controls.phone.value;

    if(this.registerUser.controls['password'].value === this.registerUser.controls['confirm\_password'].value){

      if(this.usernameValid(u)){

        if(this.emailValid(e)){

          if(this.numberValid(p)){

            this.userService.saveUser(this.registerUser.value).subscribe(data => {

              alert('User Inserted Successfully');

              this.router.navigate(['/display-users']);

            });

          }else{

            alert('Mobile Number Already Exist.');

          }

        }else{

          alert('Email Already Exist.');

        }

      }else{

        alert('Username Already Exist.');

      }

    }else{

      alert('Your Passwords doesn\'t Match');

    }

  }

  usernameValid(u:string){

    for(let user of this.users){

      if(user.username===u){

        return false;

      }

    }

    return true;

  }

  emailValid(e:string){

    for(let user of this.users){

      if(user.email===e){

        return false;

      }

    }

    return true;

  }

  numberValid(p:number){

    for(let user of this.users){

      if(user.phone==p){

        return false;

      }

    }

    return true;

  }

  ngOnInit() {

    this.registerUser=this.formBuilder.group({

      id:[],

      username: ['',Validators.required],

      password: ['',Validators.required],

      confirm\_password: ['',Validators.required],

      admin: ['false'],

      email: ['',[Validators.required,Validators.email]],

      phone: ['',Validators.required],

      confirmed: ['false'],

      dp: ['../../assets/images/default.png']

    });

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

      this.users=data;

    });

  }

}

Register-form.component.html:

<div>

  <div class="row">

      <div class="col-lg-3 col-md-2 col-sm-1"></div>

      <div class="col-lg-6 col-md-8 col-sm-10">

        <form [formGroup]="registerUser" class="form-group border p-3 p-lg-5 border-curved mt-5">

          <p class="h2 text-muted text-uppercase text-center">Registration</p>

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

          <input name="username" formControlName="username" type="text" class="form-control" placeholder="Your Username">

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

          <input name="password" formControlName="password" type="password" class="form-control" placeholder="Your Password">

          <label for="confirm-password">Confirm Password:</label>

          <input name="confirm-password" formControlName="confirm\_password" type="password" class="form-control" placeholder="Confirm Password">

          <label for="email">Email:</label>

          <input name="email" formControlName="email" type="email" class="form-control" placeholder="Your Email">

          <label for="phone">Mobile Number:</label>

          <input name="phone" formControlName="phone" type="number" class="form-control" placeholder="Your Mobile Number">

          <p class="text-center w-100 my-2"><a routerLink="/login">Already Have an Account? Click Here to Login.</a></p>

          <button class="btn btn-outline-success text-dark text-center form-control" (click)="addUser()" [disabled]="!registerUser.valid">Register</button>

        </form>

      </div>

  </div>

</div>

Reset.component.ts:

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

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

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

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

import { User } from 'src/models/user';

@Component({

  selector: 'app-reset',

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

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

})

export class ResetComponent implements OnInit {

  validateUser:FormGroup;

  users:User[];

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

  ngOnInit() {

    this.validateUser=this.formbuilder.group({

      username:['',Validators.required]

    });

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

      this.users=data;

    });

  }

  setResetCode(){

    let flag:boolean;

    let id:number;

    let uname=this.validateUser.controls.username.value;

    for(let user of this.users){

      if(user.username === uname){

        flag=true;

        id=user.id;

      }

    }

    if(flag){

      let code=Math.ceil(Math.random()\*10000000000);

      localStorage.setItem('code',code.toString());

      localStorage.setItem('userId',id.toString());

      console.log('Activation Code: '+code);

      this.router.navigate(['/reset-code']);

    }else{

      alert('Username not found.');

    }

  }

}

Reset.component.html:

<div class="row">

  <div class="col-lg-3 col-md-2 col-sm-1"></div>

  <div class="col-lg-6 col-md-8 col-sm-10">

    <form [formGroup]="validateUser" class="form-group border p-3 p-lg-5 border-curved mt-5">

      <p class="h2 text-muted text-uppercase text-center">Reset Password</p>

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

      <input class="form-control mb-4" type="text" name="username" id="" placeholder="Your Username" formControlName="username">

      <button class="btn btn-outline-success text-dark text-center form-control" (click)="setResetCode()" [disabled]="!validateUser.valid">Reset</button>

    </form>

  </div>

</div>

Reset-code.component.ts:

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

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

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

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

@Component({

  selector: 'app-reset-code',

  templateUrl: './reset-code.component.html',

  styleUrls: ['./reset-code.component.css']

})

export class ResetCodeComponent implements OnInit {

  validateCode:FormGroup;

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

  ngOnInit() {

    this.validateCode=this.formbuilder.group({

      code:['',Validators.required]

    });

  }

  checkCode(){

    let code=+localStorage.getItem('code');

    let user\_code=+this.validateCode.controls.code.value;

    if(code === user\_code){

      this.router.navigate(['/reset-password']);

    }else{

      alert('Incorrect Activation Code.');

    }

  }

}

Reset-code.component.html:

<div class="row">

    <div class="col-lg-3 col-md-2 col-sm-1"></div>

    <div class="col-lg-6 col-md-8 col-sm-10">

      <form [formGroup]="validateCode" class="form-group border p-3 p-lg-5 border-curved mt-5">

        <p class="h2 text-muted text-uppercase text-center">Reset Password</p>

        <label for="code">Activation Code:</label>

        <input class="form-control mb-4" type="text" name="username" id="" placeholder="Paste Your Activation Code Here" formControlName="code">

        <button class="btn btn-outline-success text-dark text-center form-control" (click)="checkCode()" [disabled]="!validateCode.valid">Reset</button>

      </form>

    </div>

  </div>

Reset-password.component.ts:

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

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

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

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

import { User } from 'src/models/user';

@Component({

  selector: 'app-reset-password',

  templateUrl: './reset-password.component.html',

  styleUrls: ['./reset-password.component.css']

})

export class ResetPasswordComponent implements OnInit {

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

  updatePassword:FormGroup;

  user:User;

  ngOnInit() {

    this.updatePassword=this.formBuilder.group({

      password: ['',Validators.required],

      confirm\_password: ['',Validators.required]

    });

    const id=localStorage.getItem('userId');

    this.userService.getUserById(+id).subscribe(data=>{

      this.user=data;

    })

  }

  resetPassword(){

    if((this.updatePassword.controls['password'].value === this.updatePassword.controls['confirm\_password'].value)){

        this.user.password=this.updatePassword.controls['password'].value;

        this.userService.updateUser(this.user).subscribe(u=>{

          this.router.navigate(['display-users']);

          alert('Your Password is Changed.');

        });

    }else{

      alert('Your Passwords doesn\'t match');

    }

  }

}

Reset-password.component.html:

<div>

    <div class="row">

        <div class="col-lg-3 col-md-2 col-sm-1"></div>

        <div class="col-lg-6 col-md-8 col-sm-10">

          <form [formGroup]="updatePassword" class="form-group border p-3 p-lg-5 border-curved mt-5">

            <p class="h2 text-muted text-uppercase text-center">Reset Password</p>

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

            <input name="password" formControlName="password" type="password" class="form-control" placeholder="Your New Password">

            <label for="confirm\_password">Confirm Password:</label>

            <input name="confirm\_password" formControlName="confirm\_password" type="password" class="form-control" placeholder="Confirm Your Password">

            <button class="btn btn-outline-warning text-dark text-center form-control my-3" (click)="resetPassword()" [disabled]="!updatePassword.valid">Change Password</button>

          </form>

        </div>

    </div>

  </div>

Update-company.component.ts:

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

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

import { CompanyService } from 'src/services/company.service';

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 {

  updateFormGroup:FormGroup;

  constructor(private formBuilder:FormBuilder, private companyService:CompanyService, private router:Router) { }

  ngOnInit() {

    this.updateFormGroup=this.formBuilder.group({

      id:['',Validators.required],

      name:['',Validators.required],

      turnover:['',Validators.required],

      ceo:['',Validators.required],

      board\_of\_directors:['',Validators.required],

      listed\_in\_stock\_exchanges:['',Validators.required],

      sector:['',Validators.required],

      brief:['',Validators.required],

      stock\_code:['',Validators.required],

      activated:[true],

    });

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

    if(+id > 0){

      this.companyService.getCompanyById(+id).subscribe(company=>{

        this.updateFormGroup.patchValue(company);

      });

    }

  }

  updateThisCompany(){

    this.companyService.updateCompany(this.updateFormGroup.value).subscribe(c=>{

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

    })

  }

}

Update-company.component.html:

<div class="row">

  <div class="col-3"></div>

  <div class="col-6">

    <form [formGroup]="updateFormGroup" class="form-group border p-5 border-curved mt-5">

      <p class="h2 text-muted text-uppercase text-center mb-4">Update Company</p>

      <label for="id">ID:</label>

      <input type="number" name="id" formControlName="id" class="form-control" placeholder="Company ID" disabled>

      <label for="username">Company Name:</label>

      <input type="text" class="form-control" placeholder="Name of Company" formControlName="name">

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

      <input type="text" class="form-control" placeholder="Company Turnover" formControlName="turnover">

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

      <input type="text" class="form-control" placeholder="CEO of Company" formControlName="ceo">

      <label for="username">Board of Director:</label>

      <input type="text" class="form-control mb-4" placeholder="Board of Director of Company" formControlName="board\_of\_directors">

      <label for="username">Listed in Stock Exchanges:</label><br>

      <small class="text-info">Hold [Ctrl] to Select Multiple Stock Exchanges</small>

      <select class="form-control" name="listed" id="" multiple size="2" formControlName="listed\_in\_stock\_exchanges">

          <option value="bse">BSE</option>

          <option value="nse">NSE</option>

      </select>

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

      <input type="text" class="form-control" placeholder="Name of Sector" formControlName="sector">

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

      <textarea class="form-control" name="brief" id="" cols="30" rows="3" placeholder="Brief Information about Company" formControlName="brief"></textarea>

      <label for="username">Stock Code:</label>

      <input type="text" class="form-control mb-4" placeholder="Unique Stock Code" formControlName="stock\_code">

      <button class="btn btn-warning text-dark text-center form-control" (click)="updateThisCompany()">Update Company</button>

    </form>

  </div>

</div>

Update-ipo.component.ts:

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

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

import { IPOService } from 'src/services/ipo.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 {

  constructor(private formBuilder:FormBuilder,private ipoService:IPOService,private router:Router) { }

  updateIPO:FormGroup;

  ngOnInit() {

    this.updateIPO=this.formBuilder.group({

      id:[],

      company\_name:['',Validators.required],

      stock\_exchange:['',Validators.required],

      price\_per\_share:['',Validators.required],

      total\_shares:['',Validators.required],

      open\_date:['',Validators.required],

      open\_time:['',Validators.required],

      remarks:['']

    });

    const id=localStorage.getItem('ipoId');

    if(+id>0){

      this.ipoService.getIPOById(+id).subscribe(ipo=>{

        this.updateIPO.patchValue(ipo);

      })

    }

  }

  updateIpo(){

    this.ipoService.updateIPO(this.updateIPO.value).subscribe(data=>{

      alert('IPO Updated Successfully');

      this.router.navigate(['/display-ipos']);

    })

  }

}

Update-ipo.component.html:

<div>

  <div class="row">

    <div class="col-lg-3 col-md-2 col-sm-1"></div>

        <div class="col-lg-6 col-md-8 col-sm-10">

      <form [formGroup]="updateIPO" class="form-group border p-3 p-lg-5 border-curved mt-5" (ngSubmit)="updateIpo()">

        <p class="h2 text-muted text-uppercase text-center mb-3 ">IPO</p>

        <label for="id">Id</label>

        <input class="form-control" type="number" name="number" id="" formControlName="id" placeholder="Your ID" disabled>

        <label for="username">Company Name:</label>

        <select name="company" id="" class="form-control" formControlName="company\_name">

            <option value="apple">Apple</option>

            <option value="microsoft">Microsoft</option>

            <option value="google">Google</option>

        </select>

        <label for="username">Stock Exchange:</label>

        <select class="form-control" name="stock-exchange" formControlName="stock\_exchange">

            <option value="bse" selected>BSE</option>

            <option value="nse">NSE</option>

        </select>

        <label for="username">Price per Share:</label>

        <div class="input-group">

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

                <i class="fas fa-rupee-sign input-group-text"></i>

            </div>

            <input type="number" class="form-control" placeholder="Price in Rupees" formControlName="price\_per\_share">

        </div>

        <label for="username">No. of Shares:</label>

        <input type="number" class="form-control" placeholder="Total Number of Shares" formControlName="total\_shares">

        <label for="username">Open Date:</label><br>

        <input class="form-control" type="date" name="date" id="" formControlName="open\_date">

        <label for="username">Open Time:</label><br>

        <input class="form-control" type="time" name="time" id="" formControlName="open\_time">

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

        <textarea class="form-control mb-4" name="brief" id="" cols="30" rows="3" placeholder="Remarks about this IPO" formControlName="remarks"></textarea>

        <button class="btn btn-warning text-dark text-center form-control" [disabled]="!updateIPO.valid">Update IPO</button>

      </form>

    </div>

  </div>

  </div>

Update-stock-exchange.component.ts:

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

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

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

import { StockExchangeService } from 'src/services/stock-exchange.service';

import { StockExchange } from 'src/models/stock-exchange';

@Component({

  selector: 'app-update-stock-exchange',

  templateUrl: './update-stock-exchange.component.html',

  styleUrls: ['./update-stock-exchange.component.css']

})

export class UpdateStockExchangeComponent implements OnInit {

  constructor(private formBuilder:FormBuilder, private router:Router,private seService:StockExchangeService) { }

  updateSE:FormGroup;

  se:StockExchange;

  ngOnInit() {

    this.updateSE=this.formBuilder.group({

      id:['',Validators.required],

      stock\_exchange:['',Validators.required],

      brief:['',Validators.required],

      address:['',Validators.required],

      remarks:['',Validators.required]

    });

    const id=localStorage.getItem('seId');

    if(+id > 0){

      this.seService.getStockExchangeById(+id).subscribe(se=>{

        this.updateSE.patchValue(se);

        this.se=se;

      });

    }

  }

  updateThisSE(){

    this.seService.updateStockExchange(this.updateSE.value).subscribe(c=>{

      this.router.navigate(['/manage-stock-exchange']);

      alert('Stock Exchange Updated Successfully.');

    })

  }

}

Update-stock-exchange.component.html:

<div class="row">

  <div class="col-3"></div>

  <div class="col-6">

    <form [formGroup]="updateSE" class="form-group border p-5 border-curved mt-5">

      <p class="h2 text-muted text-uppercase text-center mb-4">Stock Exchange</p>

      <label for="id">Stock Exchange Id:</label>

      <input type="number" class="form-control" placeholder="Name of Stock Exchange" formControlName="id" disabled>

      <label for="username">Stock Exchange Name:</label>

      <input type="text" class="form-control" placeholder="Name of Stock Exchange" formControlName="stock\_exchange">

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

      <textarea class="form-control" placeholder="Brief About Stock Exchange" rows="3" formControlName="brief"></textarea>

      <label for="username">Contact Address:</label>

      <input type="text" class="form-control" placeholder="Contact Address of Stock Exchange" formControlName="address">

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

      <textarea class="form-control mb-4" placeholder="Remarks About Stock Exchange" rows="3" formControlName="remarks"></textarea>

      <button class="btn btn-warning text-dark text-center form-control" (click)="updateThisSE()" [disabled]="!updateSE.valid">Update Stock Exchange</button>

    </form>

  </div>

</div>

Update-user.component.ts:

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

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

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

import { User } from 'src/models/user';

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

@Component({

  selector: 'app-update-user',

  templateUrl: './update-user.component.html',

  styleUrls: ['./update-user.component.css']

})

export class UpdateUserComponent implements OnInit {

  updateUser:FormGroup;

  user:User;

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

  ngOnInit() {

    this.updateUser=this.formBuilder.group({

      id: [],

      username: ['',Validators.required],

      email: ['',[Validators.required,Validators.email]],

      phone: ['',Validators.required]

    });

    const id=localStorage.getItem('userId');

    if(+id > 0){

      this.userService.getUserById(+id).subscribe(user => {

        this.updateUser.patchValue(user);

        this.user=user;

      });

    }

  }

  updateThisUser(){

    this.user.username=this.updateUser.controls.username.value;

    this.user.email=this.updateUser.controls.email.value;

    this.user.phone=this.updateUser.controls.phone.value;

    this.userService.updateUser(this.user).subscribe( u => {

      localStorage.setItem('userId',this.user.id.toString());

      this.router.navigate(['user-profile']);

      alert('User Updated Successfully.');

    });

  }

}

Update-user.component.html:

<div>

  <div class="row">

      <div class="col-lg-3 col-md-2 col-sm-1"></div>

      <div class="col-lg-6 col-md-8 col-sm-10">

        <form [formGroup]="updateUser" class="form-group border p-3 p-lg-5 border-curved mt-5">

          <p class="h2 text-muted text-uppercase text-center">Update User</p>

          <img src="{{user.dp}}" alt="Profile Picture" class="hover avatar m-3 mx-auto" routerLink="/upload-dp">

          <br>

          <label for="id">Id:</label>

          <input name="id" formControlName="id" type="number" class="form-control" placeholder="Your Id" disabled>

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

          <input name="username" formControlName="username" type="text" class="form-control" placeholder="Your Username">

          <label for="email">Email:</label>

          <input name="email" formControlName="email" type="email" class="form-control" placeholder="Your Email">

          <label for="phone">Mobile Number:</label>

          <input name="phone" formControlName="phone" type="number" class="form-control" placeholder="Your Mobile Number">

          <button class="btn btn-block btn-secondary m-3 mx-auto" routerLink="/change-password">Change Password</button>

          <button class="btn btn-warning text-dark text-center form-control my-3" (click)="updateThisUser()" [disabled]="!updateUser.valid">Update User</button>

        </form>

      </div>

  </div>

</div>

Update-user.component.css:

.avatar {

  vertical-align: middle;

  width: 100px;

  height: 100px;

  border-radius: 50%;

}

Upload-dp.component.ts:

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

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

@Component({

  selector: 'app-upload-dp',

  templateUrl: './upload-dp.component.html',

  styleUrls: ['./upload-dp.component.css']

})

export class UploadDpComponent implements OnInit {

  constructor(private userService:UserService) { }

  ngOnInit() {

  }

}

Upload-dp.component.html:

<div class="row">

  <div class="col-lg-3 col-md-2 col-sm-1"></div>

  <div class="col-lg-6 col-md-8 col-sm-10">

    <form action="" class="form-group border p-3 p-lg-5 border-curved mt-5">

      <p class="h2 text-muted text-uppercase text-center mb-3">Upload DP</p>

      <input type="file" name="upload" id="" class="form-control-file mt-4">

      <small class="form-text text-muted mb-2" id="fileHelp">Max 3mb size</small>

      <button class="btn btn-outline-success text-dark text-center form-control">Upload</button>

    </form>

  </div>

</div>

User-page.component.ts:

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

@Component({

  selector: 'app-user-page',

  templateUrl: './user-page.component.html',

  styleUrls: ['./user-page.component.css']

})

export class UserPageComponent implements OnInit {

  constructor() { }

  ngOnInit() {

  }

}

User-page.component.html:

<div>

    <div class="row mb-5 pt-5">

        <div class="col-md-6 my-3">

            <div class="card my-card border border-curved text-center h-100" routerLink="">

                <div class="card-body">

                    <h4 class="card-title display-4">Compare Company</h4>

                </div>

            </div>

        </div>

        <div class="col-md-6 my-3">

            <div class="card my-card border border-curved text-center h-100" routerLink="">

                <div class="card-body">

                    <h4 class="card-title display-4">Compare Sectors</h4>

                </div>

            </div>

        </div>

        <div class="col-md-6 my-3">

            <div class="card my-card border border-curved text-center h-100" routerLink="/view-ipos">

                  <div class="card-body">

                      <h4 class="card-title display-4">Display IPO</h4>

                  </div>

            </div>

        </div>

        <!-- <div class="col-md-6 my-3">

            <div class="card my-card border border-curved text-center h-100" routerLink="/add-ipo">

                <div class="card-body">

                    <h4 class="card-title display-4">Add IPO Details</h4>

                </div>

            </div>

        </div> -->

    </div>

  </div>

User-profile.component.ts:

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

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

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

import { User } from 'src/models/user';

@Component({

  selector: 'app-user-profile',

  templateUrl: './user-profile.component.html',

  styleUrls: ['./user-profile.component.css']

})

export class UserProfileComponent implements OnInit {

  user:User;

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

  ngOnInit() {

    const id=localStorage.getItem('userId');

    this.userService.getUserById(+id).subscribe(data=>{

      this.user=data;

    })

  }

  updateUser(){

    localStorage.setItem('userId',this.user.id.toString());

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

  }

  deactivateUser(){

    this.user.confirmed=false;

    this.userService.updateUser(this.user).subscribe(data=>{

      alert('Account Deactivated Successfully.');

      this.router.navigate(['/display-users']);

    })

  }

}

User-profile.component.html:

<div>

    <div class="row">

        <div class="col-lg-3 col-md-2 col-sm-1"></div>

        <div class="col-lg-6 col-md-8 col-sm-10">

          <div class="form-group border p-3 p-lg-5 border-curved mt-5">

            <p class="h2 text-muted text-uppercase text-center">User Profile</p>

            <div class="">

                <img src="{{user.dp}}" alt="Profile Picture" class="avatar m-3">

                <label class="p-3 pl-5 h5" for="id">ID:</label>

                <span class="p-3 lead">{{user.id}}</span>

                <br>

            <label class="p-3 h5" for="username">Username:</label>

            <span class="p-3 lead">{{user.username}}</span>

            <br>

            <label class="p-3 h5" for="email">Email:</label>

            <span class="p-3 lead">{{user.email}}</span>

            <br>

            <label class="p-3 h5" for="phone">Mobile Number:</label>

            <span class="p-3 lead">{{user.phone}}</span>

            <br>

            <button class="btn btn-outline-warning text-dark text-center form-control my-3" (click)="updateUser()">Edit Profile</button>

            <button class="btn btn-outline-danger text-dark text-center form-control my-3" (click)="deactivateUser()">Deactivate Account</button>

            </div>

          </div>

        </div>

    </div>

  </div>

User-profile.component.css:

.avatar {

    vertical-align: middle;

    width: 100px;

    height: 100px;

    border-radius: 50%;

  }

View-ipos.component.ts:

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

import { IPOService } from 'src/services/ipo.service';

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

import { IPO } from 'src/models/ipo';

@Component({

  selector: 'app-view-ipos',

  templateUrl: './view-ipos.component.html',

  styleUrls: ['./view-ipos.component.css']

})

export class ViewIposComponent implements OnInit {

  constructor(private ipoService:IPOService, private router:Router) { }

  ipos:IPO[];

  ngOnInit() {

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

      this.ipos=data;

    });

  }

}

View-ipos.component.html:

<div class="row py-5 py-md-4 py-3">

  <div class="col mx-auto table-responsive">

      <table class="table border-curved p-1 p-md-3 p-lg-5">

          <caption style="caption-side: top;"><h1 class=" h2 text-center text-uppercase" >All IPOs</h1></caption>

  <thead>

    <th>Id</th>

    <th>Company Name</th>

    <th>Stock Exchange</th>

    <th>Price Per Share</th>

    <th>Total Shares</th>

    <th>Open Date</th>

    <th>Open Time</th>

    <th>Remarks</th>

  </thead>

  <tbody \*ngFor="let ipo of ipos">

    <td>{{ipo.id}}</td>

    <td>{{ipo.company\_name}}</td>

    <td>{{ipo.stock\_exchange}}</td>

    <td>{{ipo.price\_per\_share}}</td>

    <td>{{ipo.total\_shares}}</td>

    <td>{{ipo.open\_date}}</td>

    <td>{{ipo.open\_time}}</td>

    <td>{{ipo.remarks}}</td>

  </tbody>

</table>

    </div>

</div>

Auth.guard.ts:

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

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

import { Observable } from 'rxjs';

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

@Injectable({

  providedIn: 'root'

})

export class AuthGuard 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.isLoggedIn()){

      return true;

    }else{

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

      // sessionStorage.setItem("error","Please login first to Access this webpage.");

      alert("Please login first to Access this webpage.");

      return false;

    }

  }

}