**CSEE 5590-0002 : Web/Mobile Programming**

**Project Source Code**

# **KC Donation Web Application**

# **Team 3**

## Roshna Toke, ClassID: 23

## Manaswini Vedula, ClassID: 24

## Yamini Saraswathi Bommineni, ClassID: 03

**Main page:**

**Html**

<!DOCTYPE html>  
<!--[if IE 9]> <html class="ie9 no-focus" lang="en"> <![endif]-->  
<!--[if gt IE 9]><!--> <html class="no-focus" lang="en"> <!--<![endif]-->  
<head>  
 <meta charset="utf-8">  
 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">  
 <title>Donation KC</title>  
</head>  
<body>  
<div class="header" id="Header">  
 <div class="navbar navbar-light bg-light">  
 <ul>  
 <li><a (click)="scroll('Header')">Home</a></li>  
 <li><a (click)="scroll('Gallary')">Gallary</a></li>  
 <li><a (click)="scroll('ContactUS')">Contact us</a></li>  
 <li><a (click)="scroll('About')">About</a></li>  
 <li></li>  
 <li><i class="fa fa-sign-in" style="font-size:24px"></i><a tabindex="-1" routerLink="/registration" routerLinkActive="active"> Sign-in</a></li>  
 <li><a tabindex="-1" routerLink="/loginpage" routerLinkActive="active">Login</a></li>  
 </ul>  
 </div>  
</div>  
<div class="home" id="Home" style="width: 100%; height:500px; background-color: #0a535e; margin-left: 5px;">  
 <div id="MainDiv" style="height: 100%; margin-left: 16%; margin-top: 10px;"></div>  
<!-- <div class="row">-->  
<!-- <div class="col-sm-4" id="image1">-->  
<!-- <img src="../../assets/img/donatefood.gif" alt="mainimage" style="width: 100%; margin: 15%">-->  
<!--&lt;!&ndash; <img src="https://www.ifhomeless.org/wp-content/uploads/2019/01/Thanksgiving-kit-550.jpg" alt="mainimage" style="width: 100%; margin: 15%">&ndash;&gt;-->  
<!-- </div>-->  
<!-- <div class="col-sm-1"></div>-->  
<!-- <div class="col-sm-5" style="padding-top: 2%; color: white; text-align: center">-->  
<!-- <h1 style="text-align: center">KC Donation</h1><br><br>-->  
<!-- <div>-->  
<!-- <strong><p>"Your Donation matters.<br>-->  
<!-- God has given us two hands one for helping ourselves and other is to help others.<br>-->  
<!-- Remember that the happiest people are not those getting more, but those giving more.<br>-->  
<!-- Giving is not just about donation, it's about making a difference. So, let's together strive for this change by donating." <br></p></strong><br></div>-->  
<!-- <br>-->  
<!-- <div>-->  
<!-- <strong>-->  
<!-- <p>The donation of commercially canned, boxed, and otherwise packaged foods is encouraged.<br>-->  
<!-- Foods prepared, cooked, cooled, or reheated at home (except for baked goods) are not accepted.<br></p></strong>-->  
<!-- </div>-->  
<!-- </div>-->  
<!-- </div>-->  
</div>  
<br><br><br><br><br><br>  
<div class="gallary" id="Gallary" style="width: 100%; height:500px; background-color: #0a535e; margin-left: 5px;">  
 <h1 style="color: white; margin-left: 25%; margin-top: 2%;"><strong>People will appreciate your donation...!!!</strong></h1>  
 <ul class="row">  
 <li><img src="../../assets/img/donate1.jpg" alt="donate1" height="250" width="350"></li>  
 <li><img src="../../assets/img/donate2.jpg" alt="donate2" height="250" width="350"></li>  
 <li><img src="../../assets/img/donate3.jpg" alt="donate3" height="250" width="350"></li>  
 <li><img src="../../assets/img/donate4.jpg" alt="donate4" height="250" width="350"></li>  
 <li><img src="../../assets/img/donate5.jpg" alt="donate5" height="250" width="350"></li>  
 <li><img src="../../assets/img/donate1.jpg" alt="donate1" height="250" width="350"></li>  
 <li><img src="../../assets/img/donate2.jpg" alt="donate2" height="250" width="350"></li>  
 <li><img src="../../assets/img/donate3.jpg" alt="donate3" height="250" width="350"></li>  
 </ul>  
</div>  
<br><br><br><br><br><br>  
<div class="contactUs" id="ContactUS" style="width: 100%; height:500px; background-color: #0a535e; margin-left: 5px;">  
 <div class="row">  
<!-- <div class="col-sm-4" id="image3">-->  
<!-- <img src="../../assets/img/donatefood.gif" alt="mainimage" style="width: 100%; margin: 15%">-->  
<!-- &lt;!&ndash; <img src="https://www.ifhomeless.org/wp-content/uploads/2019/01/Thanksgiving-kit-550.jpg" alt="mainimage" style="width: 100%; margin: 15%">&ndash;&gt;-->  
<!-- </div>-->  
<!-- <div class="col-sm-1"></div>-->  
<!-- <div class="col-sm-5" style="padding-top: 2%; color: white; text-align: center">-->  
<!-- <h1 style="text-align: center">Contact US</h1><br><br>-->  
<!-- <div>-->  
<!-- <strong><p>KC Donation<br><br>-->  
<!-- Address : 525 East Armour Blvd.,<br>-->  
<!-- Apt 112, The Newbern,<br>-->  
<!-- Kansas City, Missouri.<br>-->  
<!-- Pincode: 64109-->  
<!-- </p></strong><br><br>-->  
<!-- <strong><p>-->  
<!-- Email : rmtcxk@kcdonation.com<br>-->  
<!-- Contact No. +1 (832) 235-7930<br>-->  
<!-- </p></strong>-->  
<!-- </div>-->  
<!-- </div>-->  
 <div class="col-sm-4" id="image3">  
 <img src="../../assets/img/4303020/6.jpg" alt="mainimage" style="width: 110%; margin-left: 5%; height: 400px; margin-top: 8%">  
 </div>  
 <div class="col-sm-4" style="padding-top: 2%; color: white; text-align: center; margin-top: 5%">  
 <h1 style="text-align: center">Contact Us</h1><br><br>  
 <div>  
 <strong><p>KC Donation<br><br>  
 Address : 525 East Armour Blvd.,<br>  
 Apt 112, The Newbern,<br>  
 Kansas City, Missouri.<br>  
 Pincode: 64109  
 </p></strong><br><br>  
 <strong><p>  
 Email : rmtcxk@kcdonation.com<br>  
 Contact No. +1 (832) 235-7930<br>  
 </p></strong>  
 </div>  
 </div>  
 <div class="col-sm-4" style="padding-top: 2%; border-style: solid; text-align: center; margin-top: 3%">  
 <div>  
 <label style="color: ghostwhite;">Name:</label>&nbsp;<input type="text" name="Enter your name" style="margin-left: 10%"><br><br>  
 <label style="color: white;">Email-ID:</label><input type="text" name="Enter your Email-ID" style="margin-left: 7%"><br><br>  
 <label style="margin-right: 6%; color: white;">Address:</label><textarea type="text" name="Enter your Address" style="margin-left: 1%; width: 37%;"></textarea><br><br>  
 <label style="color: white;">Contact No.:</label><input type="text" name="Enter your Contact No." style="margin-left: 3%"><br><br>  
 <label style="margin-right: 32%; color: white;">Suggestion/Query:</label><br><br><textarea type="text" style="margin-right: 1%; width: 60%" name="Enter your Suggestion/Query"></textarea><br><br>  
 <button class="btn btn-info" name="Submit" style="color: black; width: 25%;">Submit</button><br><br>  
 </div>  
 </div>  
 </div>  
</div>  
<br><br><br><br><br><br>  
<div class="about" id="About" style="width: 100%; height:500px; background-color: #0a535e; margin-left: 5px;"><br>  
 <h1 style="margin-left: 47%; color: whitesmoke;"><u>About Us</u></h1><br><br><br><br>  
 <div class="row">  
 <div class="col-sm-4" id="yamini" style="text-align: center;">  
 <img src="../../assets/img/yamini.png" alt="roshna" style="width: 38%; margin-left: 10%"><br>  
 <label style="margin-left: 10%; color: whitesmoke;">Yamini</label><br>  
 <label style="margin-left: 10%; color: whitesmoke;">Software Developer</label>  
 </div>  
 <div class="col-sm-4" id="roshna" style="text-align: center;">  
 <img src="../../assets/img/roshna.png" alt="roshna" style="width: 42%; margin-left: 10%; height: 185px"><br>  
 <label style="margin-left: 10%; color: whitesmoke;">Roshna Toke</label><br>  
 <label style="margin-left: 10%; color: whitesmoke;">Software Developer</label>  
 </div>  
 <div class="col-sm-4" id="manaswini" style="text-align: center;">  
 <img src="../../assets/img/manaswini.png" alt="roshna" style="width: 38%; margin-left: 10%"><br>  
 <label style="margin-left: 10%; color: whitesmoke;">Manaswini Vedula</label><br>  
 <label style="margin-left: 10%; color: whitesmoke;">Software Developer</label>  
 </div>  
 </div>  
 <div style="margin-top: 50px; margin-left: 40%; font-size: 20px; font-family:verdana;">  
 <p style="color: whitesmoke;">“We make a living by what we get,</p>  
 <p style="color: whitesmoke">But we make a life by what we give.”</p>  
 </div>  
<!-- <div class="row">-->  
<!-- <div class="col-sm-4" id="image4">-->  
<!-- <img src="../../assets/img/about2.jpg" alt="mainimage" style="width: 100%; margin: 15%">-->  
<!-- </div>-->  
<!-- <div class="col-sm-1"></div>-->  
<!-- <div class="col-sm-5" style="padding-top: 2%; color: white; text-align: center">-->  
<!-- <h1 style="text-align: center">About</h1><br><br>-->  
<!-- <div>-->  
<!-- <strong><p>"Your Donation matters.<br>-->  
<!-- God has given us two hands one for helping ourselves and other is to help others.<br>-->  
<!-- Remember that the happiest people are not those getting more, but those giving more.<br>-->  
<!-- Giving is not just about donation, it's about making a difference. So, let's together strive for this change by donating." <br></p></strong><br></div>-->  
<!-- <br>-->  
<!-- <div>-->  
<!-- <strong>-->  
<!-- <p>The donation of commercially canned, boxed, and otherwise packaged foods is encouraged.<br>-->  
<!-- Foods prepared, cooked, cooled, or reheated at home (except for baked goods) are not accepted.<br></p></strong>-->  
<!-- </div>-->  
<!-- </div>-->  
<!-- </div>-->  
</div>  
<br><br><br><br><br>  
</body>  
<script src="mainpage.component.ts"></script>  
</html>

Type Script

import { ***Component***, OnInit } from '@angular/core';  
import { HttpService } from '../http.service';  
  
@Component({  
 selector: 'app-professor',  
 templateUrl: './mainpage.component.html',  
 styleUrls: ['./mainpage.component.css']  
})  
export class MainpageComponent implements OnInit {  
 constructor(public http: HttpService, private httpService: HttpService) { }  
  
 ngOnInit() {  
 }  
  
 scroll(id) {  
 const el = ***document***.getElementById(id);  
 el.scrollIntoView({behavior: 'smooth', block: 'center'});  
 }  
}

**Registration page:**

**Html**

<!DOCTYPE html>  
<!--[if IE 9]> <html class="ie9 no-focus" lang="en"> <![endif]-->  
<!--[if gt IE 9]><!--> <html class="no-focus" lang="en"> <!--<![endif]-->  
<head>  
 <meta charset="utf-8">  
 <title>Registration</title>  
 <meta name="viewport" content="width=device-width,initial-scale=1.0,user-scalable=0">  
  
 <!-- Bootstrap and OneUI CSS framework -->  
 <link rel="stylesheet" href="../../assets/css/bootstrap.min.css">  
 <link rel="stylesheet" id="css-main" href="../../assets/css/oneui.css">  
  
</head>  
<body>  
<!-- Register Content -->  
<div class="content overflow-hidden">  
 <div class="row">  
 <div class="col-sm-8 col-sm-offset-2 col-md-6 col-md-offset-3 col-lg-4 col-lg-offset-4">  
 <!-- Register Block -->  
 <div class="block block-themed animated fadeIn">  
 <div class="block-header" style="background-color: #0a535e">  
 <ul class="block-options">  
<!-- <li>-->  
<!-- <a href="#" data-toggle="modal" data-target="#modal-terms">View Terms</a>-->  
<!-- </li>-->  
 <li>  
 <a routerLink="/loginpage" routerLinkActive="active" data-toggle="tooltip" data-placement="left" title="Log In"><i class="si si-login"></i></a>  
 </li>  
 </ul>  
 <h3 class="block-title">Register</h3>  
 </div>  
 <div class="block-content block-content-full block-content-narrow">  
  
 <p>Please fill the following details to create a new account.</p>  
  
 <form class="js-validation-register form-horizontal push-50-t push-50" id="formData" #***userData***="ngForm" (ngSubmit)="registerUser(***userData***.value)">  
 <div class="form-group">  
 <div class="col-xs-12">  
 <div class="form-material form-material-success">  
  
 <select class="form-control" name="userType" id="userType" style="width: 50%" [(ngModel)]="userType">  
 <option value="">--Select--</option>  
 <option value="Donar">Donar</option>  
 <option value="Requester">Requester</option>  
 </select>  
 <label for="userType">User Type</label>  
 </div>  
 </div>  
 </div>  
 <div class="form-group">  
 <div class="col-xs-12">  
 <div class="form-material form-material-success">  
 <input class="form-control" type="text" id="registerUsername" name="registerUsername" placeholder="Please enter a username" [(ngModel)]="registerUsername"/>  
 <label for="registerUsername">Username</label>  
 </div>  
 </div>  
 </div>  
 <div class="form-group">  
 <div class="col-xs-12">  
 <div class="form-material form-material-success">  
 <input class="form-control" type="email" id="registerEmail" name="registerEmail" placeholder="Please provide your email" [(ngModel)]="registerEmail"/>  
 <label for="registerEmail">Email</label>  
 </div>  
 </div>  
 </div>  
 <div class="form-group">  
 <div class="col-xs-12">  
 <div class="form-material form-material-success">  
 <input class="form-control" type="password" id="registerPassword" name="registerPassword" placeholder="Choose a strong password.." [(ngModel)]="registerPassword"/>  
 <label for="registerPassword">Password</label>  
 </div>  
 </div>  
 </div>  
 <div class="form-group">  
 <div class="col-xs-12">  
 <div class="form-material form-material-success">  
 <input class="form-control" type="password" id="registerPassword2" name="registerPassword2" placeholder="..and confirm it" [(ngModel)]="registerPassword2"/>  
 <label for="registerPassword2">Confirm Password</label>  
 </div>  
 </div>  
 </div>  
<!-- <div class="form-group">-->  
<!-- <div class="col-xs-12">-->  
<!-- <label class="css-input switch switch-sm switch-success">-->  
<!-- <input type="checkbox" id="registerTerms" name="registerTerms"><span></span> I agree with terms &amp; conditions-->  
<!-- </label>-->  
<!-- </div>-->  
<!-- </div>-->  
 <div class="form-group">  
 <div class="col-xs-12 col-sm-6 col-md-5">  
 <button class="btn btn-block btn-success" type="submit" style="background-color: #0a535e"><i class="fa fa-plus pull-right"></i> Sign Up</button>  
 </div>  
 </div>  
 </form>  
 <!-- END Register Form -->  
 </div>  
 </div>  
 <!-- END Register Block -->  
 </div>  
 </div>  
</div>  
<script src="registration.component.ts"></script>  
</body>  
</html>

**Type Script:**

import { ***Component***, OnInit } from '@angular/core';  
import {Router} from '@angular/router';  
import {HttpService} from '../http.service';  
  
@Component({  
 selector: 'app-registration',  
 templateUrl: './registration.component.html',  
 styleUrls: ['./registration.component.css']  
})  
export class RegistrationComponent implements OnInit {  
 public userType;  
 public registerUsername;  
 public registerPassword;  
 public registerEmail;  
 public registerPassword2;  
  
 constructor(private router: Router, private httpService: HttpService) { }  
  
 ngOnInit() {  
 }  
  
 registerUser(user) {  
 // tslint:disable-next-line:max-line-length  
 if (this.userType === undefined || this.registerUsername === undefined || this.registerPassword === undefined || this.registerEmail === undefined || this.registerPassword2 === undefined) {  
 alert('Please Enter all details');  
 } else if (this.registerUsername.length < 3) {  
 alert('Username length should be greater than 3');  
 } else if (this.registerPassword.length < 5) {  
 alert('Password length should be greater than 5');  
 } else if (this.registerPassword !== this.registerPassword2) {  
 alert('Password does not match');  
 } else {  
 this.httpService.registerUser(user);  
 alert('Successful Registration');  
 this.router.navigate(['/mainpage']);  
 } // else block  
 }  
  
}

**Login Page:**

<!DOCTYPE html>  
<!--[if IE 9]> <html class="ie9 no-focus" lang="en"> <![endif]-->  
<!--[if gt IE 9]><!--> <html class="no-focus" lang="en"> <!--<![endif]-->  
<head>  
 <meta charset="utf-8">  
  
 <title>Login</title>  
 <link rel="stylesheet" href="../../assets/css/bootstrap.min.css">  
 <link rel="stylesheet" id="css-main" href="../../assets/css/oneui.css">  
  
</head>  
<body>  
<!-- Login Content -->  
<div class="content overflow-hidden">  
 <div class="row">  
 <div class="col-sm-8 col-sm-offset-2 col-md-6 col-md-offset-3 col-lg-4 col-lg-offset-4">  
 <!-- Login Block -->  
 <div class="block block-themed animated fadeIn">  
 <div class="block-header" style="background-color: #0a535e">  
 <ul class="block-options">  
 <li>  
 <a routerLink="/forgotpasswd" routerLinkActive="active">Forgot Password?</a>  
 </li>  
 <li>  
 <a routerLink="/registration" routerLinkActive="active">Sign Up?</a>  
 </li>  
 <li>  
 <a routerLink="/mainpage" routerLinkActive="active" data-toggle="tooltip" data-placement="left" title="Main Page"><i class="si si-login"></i></a>  
 </li>  
 </ul>  
 <h3 class="block-title">Login</h3>  
 </div>  
 <div class="block-content block-content-full block-content-narrow">  
 <!-- Login Title -->  
 <h1 class="h2 font-w600 push-30-t push-5">KC Donation</h1>  
 <p>Welcome, please login.</p>  
 <!-- END Login Title -->  
  
 <!-- Login Form -->  
 <form class="js-validation-login form-horizontal push-30-t push-50" id="formData" #***userData***="ngForm" (ngSubmit)="custLogin(***userData***.value)">  
 <div class="form-group">  
 <div class="col-xs-12">  
 <div class="form-material bmd-label-floating">  
 <input class="form-control" type="text" id="username" name="username" [(ngModel)]="username"/>  
 <label for="username">Username</label>  
 </div>  
 </div>  
 </div>  
 <div class="form-group">  
 <div class="col-xs-12">  
 <div class="form-material bmd-label-floating">  
 <input class="form-control" type="password" id="password" name="password" [(ngModel)]="password"/>  
 <label for="password">Password</label>  
 </div>  
 </div>  
 </div>  
  
<!-- <div class="form-group">-->  
<!-- <div class="col-xs-12">-->  
<!-- <label class="css-input switch switch-sm switch-primary">-->  
<!-- <input type="checkbox" id="login-remember-me" name="login-remember-me"><span></span> Remember Me?-->  
<!-- </label>-->  
<!-- </div>-->  
<!-- </div>-->  
  
 <div class="form-group">  
 <div class="col-xs-12 col-sm-6 col-md-4">  
 <button class="btn btn-block btn-primary" type="submit" id="login" style="background-color: #0a535e"><i class="si si-login pull-right"></i> Log in</button>  
 </div>  
 </div>  
 </form>  
 <!-- END Login Form -->  
 </div>  
 </div>  
 <!-- END Login Block -->  
 </div>  
 </div>  
</div>  
<!-- END Login Content -->  
  
<script src="loginpage.component.ts"></script>  
</body>  
</html>

**Type Script:**

import { ***Component***, OnInit } from '@angular/core';  
import {ActivatedRoute, Router} from '@angular/router';  
import {HttpService} from '../http.service';  
  
@Component({  
 selector: 'app-profile',  
 templateUrl: './profile.component.html',  
 styleUrls: ['./profile.component.css']  
})  
export class ProfileComponent implements OnInit {  
 public existingEmail;  
 public currentPass;  
 public newPass;  
 constructor(private route: ActivatedRoute, private router: Router, private httpService: HttpService) { }  
  
 ngOnInit() {  
 }  
 updateProfile(userData) {  
 if (this.existingEmail === undefined || this.existingEmail === '') {  
 alert('Please enter your email id');  
 } else if (this.currentPass === undefined || this.currentPass === '') {  
 alert('Please enter your current password');  
 } else if (this.newPass === undefined || this.newPass === '') {  
 alert('Please enter new password which you want change');  
 } else {  
 this.httpService.profUpdate(this.existingEmail, this.currentPass, this.newPass).subscribe(data => {  
 ***console***.log(data);  
 if (data.success) {  
 ***console***.log('data coming' + data.message);  
 ***window***.alert(data.message);  
 this.router.navigate(['/loginpage']);  
 } else {  
 ***console***.log('data coming error' + data.message);  
 ***window***.alert(data.message);  
 }  
 this.existingEmail = '';  
 this.currentPass = '';  
 this.newPass = '';  
 });  
 }  
 }

}

**Profile:**

**Html**

<!DOCTYPE html>  
<!--[if IE 9]> <html class="ie9 no-focus" lang="en"> <![endif]-->  
<!--[if gt IE 9]><!--> <html class="no-focus" lang="en"> <!--<![endif]-->  
<head>  
 <meta charset="utf-8">  
  
 <title>Update Profile</title>  
  
 <!-- Bootstrap and OneUI CSS framework -->  
 <link rel="stylesheet" href="../../assets/css/bootstrap.min.css">  
 <link rel="stylesheet" id="css-main" href="../../assets/css/oneui.css">  
</head>  
<body>  
<div class="content overflow-hidden">  
 <div class="row">  
 <div class="col-sm-8 col-sm-offset-2 col-md-6 col-md-offset-3 col-lg-4 col-lg-offset-4">  
 <div class="block block-themed animated fadeIn">  
 <div class="block-header bg-primary" style="background-color: #0a535e">  
 <ul class="block-options">  
 <li>  
 <a routerLink="/loginpage" routerLinkActive="active" data-toggle="tooltip" data-placement="left" title="Log In"><i class="si si-login"></i></a>  
 </li>  
 </ul>  
 <h3 class="block-title">Update Profile</h3>  
 </div>  
 <div class="block-content block-content-full block-content-narrow">  
 <h1 class="h2 font-w600 push-30-t push-5">KC Donation</h1>  
 <p>Please provide your account’s email and current password.</p>  
 <form class="form-horizontal push-30-t push-50" id="formData" #***userData***="ngForm" (ngSubmit)="updateProfile(***userData***.value)">  
 <div class="form-group">  
 <div class="col-xs-12">  
 <div class="form-material bmd-label-floating">  
 <input class="form-control" type="email" id="existingEmail" name="existingEmail" [(ngModel)]="existingEmail">  
 <label for="existingEmail">Email</label>  
 </div>  
 </div>  
 </div>  
 <div class="form-group">  
 <div class="col-xs-12">  
 <div class="form-material bmd-label-floating">  
 <input class="form-control" type="password" id="currentPass" name="currentPass" [(ngModel)]="currentPass"/>  
 <label for="currentPass">Current Password</label>  
 </div>  
 </div>  
 </div>  
 <div class="form-group">  
 <div class="col-xs-12">  
 <div class="form-material bmd-label-floating">  
 <input class="form-control" type="password" id="newPass" name="newPass" [(ngModel)]="newPass"/>  
 <label for="newPass">New Password</label>  
 </div>  
 </div>  
 </div>  
 <div class="form-group">  
 <div class="col-xs-12 col-sm-6 col-md-5">  
 <button class="btn btn-block btn-primary" type="submit" style="background-color: #0a535e"><i class="fa fa-edit pull-right"></i>Update</button>  
 </div>  
 </div>  
 </form>  
 </div>  
 </div>  
 </div>  
 </div>  
</div>  
</body>  
</html>

**Type Script:**

import { ***Component***, OnInit } from '@angular/core';  
import {ActivatedRoute, Router} from '@angular/router';  
import {HttpService} from '../http.service';  
  
@Component({  
 selector: 'app-profile',  
 templateUrl: './profile.component.html',  
 styleUrls: ['./profile.component.css']  
})  
export class ProfileComponent implements OnInit {  
 public existingEmail;  
 public currentPass;  
 public newPass;  
 constructor(private route: ActivatedRoute, private router: Router, private httpService: HttpService) { }  
  
 ngOnInit() {  
 }  
 updateProfile(userData) {  
 if (this.existingEmail === undefined || this.existingEmail === '') {  
 alert('Please enter your email id');  
 } else if (this.currentPass === undefined || this.currentPass === '') {  
 alert('Please enter your current password');  
 } else if (this.newPass === undefined || this.newPass === '') {  
 alert('Please enter new password which you want change');  
 } else {  
 this.httpService.profUpdate(this.existingEmail, this.currentPass, this.newPass).subscribe(data => {  
 ***console***.log(data);  
 if (data.success) {  
 ***console***.log('data coming' + data.message);  
 ***window***.alert(data.message);  
 this.router.navigate(['/loginpage']);  
 } else {  
 ***console***.log('data coming error' + data.message);  
 ***window***.alert(data.message);  
 }  
 this.existingEmail = '';  
 this.currentPass = '';  
 this.newPass = '';  
 });  
 }  
 }  
}

**Forgot password:**

**Html**

<!DOCTYPE html>  
<!--[if IE 9]> <html class="ie9 no-focus" lang="en"> <![endif]-->  
<!--[if gt IE 9]><!--> <html class="no-focus" lang="en"> <!--<![endif]-->  
<head>  
 <meta charset="utf-8">  
  
 <title>Password Reminder</title>  
  
 <!-- Bootstrap and OneUI CSS framework -->  
 <link rel="stylesheet" href="../../assets/css/bootstrap.min.css">  
 <link rel="stylesheet" id="css-main" href="../../assets/css/oneui.css">  
</head>  
<body>  
<div class="content overflow-hidden">  
 <div class="row">  
 <div class="col-sm-8 col-sm-offset-2 col-md-6 col-md-offset-3 col-lg-4 col-lg-offset-4">  
 <div class="block block-themed animated fadeIn">  
 <div class="block-header bg-primary" style="background-color: #0a535e">  
 <ul class="block-options">  
 <li>  
 <a routerLink="/loginpage" routerLinkActive="active" data-toggle="tooltip" data-placement="left" title="Log In"><i class="si si-login"></i></a>  
 </li>  
 </ul>  
 <h3 class="block-title">Password Reminder</h3>  
 </div>  
 <div class="block-content block-content-full block-content-narrow">  
 <!-- Reminder Title -->  
 <h1 class="h2 font-w600 push-30-t push-5">KC Donation</h1>  
 <p>Please provide your account’s email and we will send you your password.</p>  
 <form class="form-horizontal push-30-t push-50" id="formData" #***userData***="ngForm" (ngSubmit)="custLogin(***userData***.value)">  
 <div class="form-group">  
 <div class="col-xs-12">  
 <div class="form-material form-material-primary floating">  
 <input class="form-control" type="email" id="reminderEmail" name="reminderEmail" [(ngModel)]="reminderEmail"/>  
 <label for="reminderEmail">Email</label>  
 </div>  
 </div>  
 </div>  
 <div class="form-group">  
 <div class="col-xs-12 col-sm-6 col-md-5">  
 <button class="btn btn-block btn-primary" type="submit" style="background-color: #0a535e"><i class="si si-envelope-open pull-right"></i> Send Mail</button>  
 </div>  
 </div>  
 </form>  
 </div>  
 </div>  
 </div>  
 </div>  
</div>  
</body>  
</html>

**Type Script:**

import { ***Component***, OnInit } from '@angular/core';  
import {ActivatedRoute, Router} from '@angular/router';  
import {HttpService} from '../http.service';  
  
@Component({  
 selector: 'app-forgotpasswd',  
 templateUrl: './forgotpasswd.component.html',  
 styleUrls: ['./forgotpasswd.component.css']  
})  
export class ForgotpasswdComponent implements OnInit {  
 public reminderEmail;  
 public randomNum;  
 public randomPass = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';  
 loading = false;  
 buttionText = 'Send';  
 constructor(private route: ActivatedRoute, private router: Router, private httpService: HttpService) { }  
  
 ngOnInit() {  
 this.randomNum = this.getRandomPassword();  
 // alert(this.randomNum);  
 }  
 getRandomPassword() {  
 let password = '';  
 let n = 5;  
 while ( n > 0 ) {  
 password += this.randomPass.charAt(this.randomInt(0, 61));  
 n--;  
 }  
 return password;  
 }  
 randomInt(min, max) {  
 return ***Math***.floor(***Math***.random() \* (max - min + 1)) + min;  
 }  
 custLogin(emailID) {  
 if (this.reminderEmail === undefined) {  
 // base\_pages\_login.js - validation code  
 alert('Please enter your email id');  
 } else {  
 // alert(emailID.reminderEmail);  
 this.httpService.custExist(emailID).subscribe(data => {  
 ***console***.log(data);  
 if (data.success) {  
 ***console***.log(data.message);  
 // 04/23 - start  
 this.loading = true;  
 this.buttionText = 'Sending...';  
 const user = {  
 email: emailID.reminderEmail,  
 subject: 'forgotPass',  
 randomNum: this.randomNum  
 };  
 this.httpService.sendEmail('http://localhost:3000/sendmail', user).subscribe(  
 data2 => {  
 const res: any = data2;  
 ***console***.log(`For ${user.email}, mail has been sent and the message id is ${res.messageId}`);  
 this.httpService.upDatePassword(user.email, user.randomNum).subscribe(data1 => {  
 if (data1) {  
 alert('Password updated in user table successfully');  
 } else {  
 alert('Error in updating password in user table');  
 }  
 });  
 },  
 err => {  
 ***console***.log(err);  
 this.loading = false;  
 this.buttionText = 'Send';  
 }, () => {  
 this.loading = false;  
 this.buttionText = 'Send';  
 }  
 );  
 // 04/23 - end  
 } else {  
 ***window***.alert(data.message);  
 }  
 this.reminderEmail = '';  
 });  
 }  
 }  
}

**History:**

**Html**

<head>  
 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">  
</head>  
<body>  
<button class="btn success pull-left" (click)="goToMainPage()"><b>Back </b> <i class="fa fa-arrow-circle-left" style="font-size:20px"></i></button><br><br><br><br><br>  
  
<div id="row">  
 <table>  
 <tr>  
 <th>Item Category</th>  
 <th>Item Name</th>  
 <th>Item Weight/Number</th>  
 <th>Item Expiry Date</th>  
 <th>Item Price</th>  
 <th>Item Donor Name</th>  
 <th>Donor Added Date</th>  
 <th>Donor Address</th>  
 <th>Requester Name</th>  
 <th>Requester Completion date</th>  
<!-- <th>Delete History</th>-->  
 </tr>  
 <tr *\*ngFor*="let ***item*** of temp">  
 <td>{{***item***.itemCategory}}</td>  
 <td>{{***item***.itemName}}</td>  
 <td>{{***item***.itemWeight}}</td>  
 <td>{{***item***.itemExpiry}}</td>  
 <td>{{***item***.itemPrice}}</td>  
 <td>{{***item***.donarName}}</td>  
 <td>{{***item***.donarAddedDate}}</td>  
 <td>{{***item***.itemAddress}}</td>  
 <td>{{***item***.requesterName}}</td>  
 <td>{{***item***.requesterCompDate}}</td>  
<!-- <td><button style="margin-left: 25%"><i class="fa fa-trash"></i></button></td>-->  
 </tr>  
 </table>  
 <!-- </div>-->  
</div>  
</body>

**Type Script:**

import { ***Component***, OnInit } from '@angular/core';  
import {ActivatedRoute, NavigationEnd, Router} from '@angular/router';  
import {HttpService} from '../http.service';  
  
  
@Component({  
 selector: 'app-history',  
 templateUrl: './history.component.html',  
 styleUrls: ['./history.component.css']  
})  
export class HistoryComponent implements OnInit {  
 temp: any;  
 public itemName;  
 public itemAddress;  
 public itemExpiry;  
 public itemWeight;  
 public itemPrice;  
 public username;  
 public addeddate;  
 public previousState;  
  
 // tslint:disable-next-line:max-line-length  
 constructor(private route: ActivatedRoute, private router: Router, private httpService: HttpService) {  
 }  
  
 ngOnInit() {  
 this.username = this.route.snapshot.paramMap.get('username');  
 this.previousState = this.route.snapshot.paramMap.get('previousState');  
 this.getData();  
 // alert('previousState : ' + this.previousState);  
 }  
  
 goToMainPage() {  
 if (this.previousState === 'donar') {  
 this.router.navigate(['/donar', this.username]);  
 } else {  
 this.router.navigate(['/requester', this.username]);  
 }  
 }  
  
 getData() {  
 this.httpService.getUserHistory(this.username, this.previousState).subscribe(data => {  
 this.temp = data;  
 });  
 }  
}

**Donor:**

**Html**

<!DOCTYPE html>  
<!--[if IE 9]> <html class="ie9 no-focus" lang="en"> <![endif]-->  
<!--[if gt IE 9]><!--> <html class="no-focus" lang="en"> <!--<![endif]-->  
<head>  
 <meta charset="utf-8">  
  
 <title>Donor</title>  
 <meta name="viewport" content="width=device-width,initial-scale=1.0,user-scalable=0">  
 <!-- Bootstrap and OneUI CSS framework -->  
 <link rel="stylesheet" href="../../assets/css/bootstrap.min.css">  
 <link rel="stylesheet" id="css-main" href="../../assets/css/oneui.min.css">  
</head>  
<body>  
<div>  
 <div>  
 <div class="row items-push">  
<!-- <button class="btn success pull-left" (click)="openForm()">+Donate</button>-->  
 <button class="btn success pull-left" (click)="openDialog()" style="margin-top: 2%; margin-left: 2%;">+Donate</button>  
 <h1 class="h1 font-w700 text-black animated fadeInDown fa-pull-left" style="margin-left: 33%; margin-top: 2%; font-size: 50px;">Donor</h1>  
 <ul class="nav-header pull-right" style="margin-top: 2%; margin-right: 2%;">  
 <li ngbDropdown class="nav-item dropdown">  
 <div class="btn-group">  
 <button type="button" ngbDropdownToggle class="btn btn-default btn-image dropdown-toggle" id="navbarDropdownButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" style="border-color: black;">  
 <img src="../../assets/img/avatars/avatar10.jpg" alt="Avatar">  
 <span class="caret"></span>  
 </button>  
 <div ngbDropdownMenu class="dropdown-menu" aria-labelledby="navbarDropdownMenuButton">  
 <ul>  
 <li class="dropdown-header">Profile</li>  
 <li>  
 <a tabindex="-1" href="../profile">  
 Profile<i class="si si-user pull-right" style="margin-right: 22%"></i>  
 </a>  
 </li>  
 <li class="divider"></li>  
 <li class="dropdown-header">Actions</li>  
 <li>  
 <a tabindex="-1" routerLink="/mainpage" routerLinkActive="active">  
 Log out<i class="si si-logout pull-right" style="margin-right: 22%"></i>  
 </a>  
 </li>  
 </ul>  
 </div>  
  
 </div>  
 </li>  
 </ul>  
 <button class="btn success pull-right" style="margin-top: 2%;" (click)="goToHistory()">+History</button>  
 </div>  
 </div>  
</div>  
  
<div class="card">  
<!-- <img src="jeans3.jpg" alt="Denim Jeans" style="width:100%">-->  
 <div id="row">  
 <div *\*ngFor*="let ***item*** of temp" id="col" class="col-sm-3" style="background-color: whitesmoke;">  
 <h1 id="title" style="text-align: center;"> {{***item***.itemName}} </h1><br>  
 <p id="address"><label for="Address">Address:</label> {{***item***.itemAddress }} </p>  
 <p class="price" id="weight" ><label for="price">Weight:</label> {{***item***.itemWeight}}</p>  
 <p class="price" id="expdate" ><label for="price">Expiry Date:</label> {{***item***.itemExpiry}} </p>  
 <p class="price" id="price" ><label for="price">Price: $</label> {{***item***.itemPrice}} </p>  
 <p style="text-align: center;"><button type="button" style="background-color: slategrey; color: black; width: 20%" (click)="deleteItem(***item***.\_id)">Delete</button></p>  
 </div>  
 </div>  
</div>  
  
<div id="donarForm">  
 <form id="myForm" #***itemData***="ngForm" (ngSubmit)="submit(***itemData***.value)">  
 <div class="imgcontainer">  
 <img src="../../assets/img/logo.jpg" alt="Avatar" class="avatar">  
 </div>  
  
 <div class="container">  
 <label for="category">Category:</label>&nbsp;&nbsp;&nbsp;&nbsp;  
 <select id="category" name="itemCategory" [(ngModel)]="itemCategory">  
 <option value="" selected>--Select--</option>  
 <option value="Cloths">Cloths</option>  
 <option value="Food">Food</option>  
 <option value="Furniture">Furniture</option>  
 <option value="Vehicle">Vehicle</option>  
 <option value="Medicine">Medicine</option>  
 <option value="Other">Other</option>  
 </select>&nbsp;&nbsp;&nbsp;&nbsp;  
 <label><b>Item Name</b></label>&nbsp;&nbsp;  
 <input type="text" placeholder="Enter Item Name" name="itemName" [(ngModel)]="itemName" required><br>  
 <label><b>Address</b></label>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  
 <input type="text" placeholder="Enter Address" name="itemAddress" [(ngModel)]="itemAddress" required><br>  
 <label><b>Expiry Date</b>&nbsp;&nbsp;</label><input type="date" value="2017-06-01" name="itemExpdate" [(ngModel)]="itemExpiry">&nbsp;&nbsp;  
 <label>Weight/Number</label>&nbsp;<input type="text" style="width: 20%" name="itemWeight" [(ngModel)]="itemWeight">&nbsp;&nbsp;  
 <label>Price</label>&nbsp;<input type="text" style="width: 20%" name="itemPrice" [(ngModel)]="itemPrice">  
 <input type="text" style="width: 20%; visibility: hidden" name="username" [(ngModel)]="username">  
 <input type="date" style="width: 20%; visibility: hidden" name="addeddate" [(ngModel)]="addeddate">  
 <input type="text" style="width: 20%; visibility: hidden" name="referenceID" [(ngModel)]="referenceID">  
 </div>  
  
 <div class="container" style="background-color:lightgrey">  
 <button class="cancelbtn" id= submit style="margin-left:5%">Submit</button>  
 <button class="cancelbtn" style="margin-left:30%;" type="button" (click)="closeForm()">Cancle</button>  
 </div>  
 </form>  
</div>  
<!-- OneUI Core JS: jQuery, Bootstrap, slimScroll, scrollLock, Appear, CountTo, Placeholder, Cookie and App.js -->  
<script src="donar.component.ts"></script>  
<script src="../../assets/js/oneui.min.js"></script>  
  
  
</body>  
</html>

**Ts file:**

import { ***Component***, OnInit } from '@angular/core';  
import {ActivatedRoute, NavigationEnd, Router} from '@angular/router';  
import {HttpService} from '../http.service';  
import {MatDialog} from '@angular/material';  
import {DialogComponent} from '../dialog/dialog.component';  
  
@Component({  
 selector: 'app-globadmin',  
 templateUrl: './donar.component.html',  
 styleUrls: ['./donar.component.css']  
})  
export class DonarComponent implements OnInit {  
 public itemCategory;  
 public itemName;  
 public itemAddress;  
 public itemExpiry;  
 public itemWeight;  
 public itemPrice;  
 public username;  
 public addeddate;  
 public previousState;  
 public referenceID;  
 public requesterIds;  
 public searchCat = 'All';  
 temp: any;  
 arr: any;  
 loading = false;  
 buttionText = 'Send';  
 constructor(private route: ActivatedRoute, private router: Router, private httpService: HttpService, public dialog: MatDialog) {  
 this.router.events.subscribe((event) => { if ( event instanceof NavigationEnd) {  
 this.arr = event.url.split('/', 3);  
 this.previousState = this.arr[1];  
 }  
 });  
 }  
  
 ngOnInit() {  
 ***document***.getElementById('myForm').style.display = 'none';  
 this.getData(this.searchCat);  
 this.username = this.route.snapshot.paramMap.get('username');  
 this.addeddate = new ***Date***();  
 }  
 openDialog(): void {  
  
 const dialogRef = this.dialog.open(DialogComponent, {  
 width : '800px',  
 data : {username: this.username},  
 });  
  
 dialogRef.afterClosed().subscribe(result => {  
 ***console***.log('The dialog was closed');  
 });  
 }  
 closeForm() {  
 ***document***.getElementById('myForm').style.display = 'none';  
 this.searchCat = 'All';  
 this.getData(this.searchCat);  
 }  
  
 getData(searchCat) {  
 this.httpService.getItemData(searchCat).subscribe(data => {  
 this.temp = data;  
 });  
 this.httpService.getRequestersEmail().subscribe(data => {  
 this.requesterIds = data;  
 // alert('load:' + data);  
 });  
 }  
  
 deleteItem(itemId) {  
 this.httpService.deleteItem(itemId).subscribe(data => {  
 // console.log(data.success + ' ' + data.message);  
 if (data.success) {  
 this.getData(this.searchCat);  
 } else {  
 alert('Error in deleting Item');  
 }  
 });  
 }  
  
 openForm() {  
 ***document***.getElementById('myForm').style.display = 'block';  
 }  
  
 goToHistory() {  
 this.router.navigate(['/history', this.username, this.previousState]);  
 }  
  
 submit(itemData) {  
 // tslint:disable-next-line:max-line-length  
 ***console***.log('getitemData:' + ***JSON***.stringify(itemData));  
 ***console***.log('itemWeight:' + this.itemWeight);  
 // tslint:disable-next-line:max-line-length  
 if (this.itemName === undefined || this.itemName === '' || this.itemAddress === undefined || this.itemAddress === '' || this.itemCategory === '' || this.itemPrice === undefined || this.itemPrice === '') {  
 alert('Please Enter all details');  
 } else if (this.itemCategory === 'Food' && (this.itemWeight === undefined || this.itemWeight === '')) {  
 alert('Please Enter Food weight');  
 } else if (this.itemCategory === 'Food' && (this.itemExpiry === undefined || this.itemExpiry === '')) {  
 alert('Please Enter Food expiry date');  
 if (this.itemExpiry <= new ***Date***()) {  
 alert('Item Expiry should be greater than current date');  
 }  
 } else {  
 ***document***.getElementById('myForm').style.display = 'none';  
 this.httpService.submit(itemData).subscribe(data => {  
 if (data.success) {  
 alert('Item Data added successfully');  
 this.getData(this.searchCat);  
 this.httpService.addHistory(itemData, data.referenceID).subscribe(data1 => {  
 if (data1.success) {  
 alert('Item Data added in history table successfully');  
 // 04/23 - start  
 this.loading = true;  
 this.buttionText = 'Sending...';  
 const user = {  
 email: this.requesterIds,  
 subject: 'sendNotificationToReq'  
 };  
 // alert('user emails object are :' + user.email);  
 this.httpService.sendEmail('http://localhost:3000/sendmail', user).subscribe(  
 data2 => {  
 const res: any = data2;  
 ***console***.log(  
 `For ${user.email}, mail has been sent and the message id is ${res.messageId}`  
 );  
 },  
 err => {  
 ***console***.log(err);  
 this.loading = false;  
 this.buttionText = 'Send';  
 }, () => {  
 this.loading = false;  
 this.buttionText = 'Send';  
 }  
 );  
 // 04/23 - end  
 } else {  
 alert('Error in Adding data in history table');  
 }  
 });  
 } else {  
 alert('Error in Adding a Item');  
 }  
 });  
 }  
 }  
  
}

**Requestor**

**Html**

<!DOCTYPE html>  
<!--[if IE 9]> <html class="ie9 no-focus" lang="en"> <![endif]-->  
<!--[if gt IE 9]><!--> <html class="no-focus" lang="en"> <!--<![endif]-->  
<head>  
 <meta charset="utf-8">  
  
 <title>Requester Dashboard</title>  
  
 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400italic,600,700%7COpen+Sans:300,400,400italic,600,700">  
  
 <!-- Bootstrap and OneUI CSS framework -->  
 <link rel="stylesheet" href="../../assets/css/bootstrap.min.css">  
 <link rel="stylesheet" id="css-main" href="../../assets/css/oneui.css">  
  
</head>  
<body>  
<div>  
 <!-- Header -->  
 <div class="row items-push">  
 <!-- Header Navigation Right -->  
 <button class="btn success" (click)="goToHistory()" style="background-color: slategrey; margin-top: 2%; margin-left: 2%;"><b>+History</b></button>  
 <ul class="nav-header pull-right" style="margin-top: 2%; margin-right: 2%;">  
 <li ngbDropdown class="nav-item dropdown">  
 <div class="btn-group">  
 <button type="button" ngbDropdownToggle class="btn btn-default btn-image dropdown-toggle" id="navbarDropdownButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" style="border-color: black;">  
 <img src="../../assets/img/avatars/avatar10.jpg" alt="Avatar">  
 <span class="caret"></span>  
 </button>  
 <div ngbDropdownMenu class="dropdown-menu" aria-labelledby="navbarDropdownMenuButton">  
 <ul>  
 <li class="dropdown-header">Profile</li>  
 <li>  
 <a tabindex="-1" href="../profile">  
 Profile<i class="si si-user pull-right" style="margin-right: 22%"></i>  
 </a>  
 </li>  
 <li class="divider"></li>  
 <li class="dropdown-header">Actions</li>  
 <li>  
 <a tabindex="-1" routerLink="/mainpage" routerLinkActive="active">  
 Log out<i class="si si-logout pull-right" style="margin-right: 22%"></i>  
 </a>  
 </li>  
 </ul>  
 </div>  
 </div>  
 </li>  
 </ul>  
 <i class="fa fa-search pull-right" style="font-size:30px; margin-top: 2%;"></i>  
 <select id="itemCategory" class="pull-right" name="itemCategory" [(ngModel)]="itemCategory" (ngModelChange)="showCategoryCards()" style="height: 30px; margin-top: 2%; ">  
 <option value="All">All</option>  
 <option value="Cloths">Cloths</option>  
 <option value="Food">Food</option>  
 <option value="Furniture">Furniture</option>  
 <option value="Vehicle">Vehicle</option>  
 <option value="Medicine">Medicine</option>  
 <option value="Other">Other</option>  
 </select>  
 <p class="pull-right" style="color: black; margin-top: 2%; margin-right: 10px;">Search</p>  
 <h1 class="h1 font-w700 text-black animated fadeInDown pull-right" style="margin-right: 30%; margin-top: 2%; font-size: 50px;">Requester</h1>  
 <!-- END Header Navigation Right -->  
 </div>  
 <!-- END Header -->  
</div>  
  
<div class="card">  
 <!-- <img src="jeans3.jpg" alt="Denim Jeans" style="width:100%">-->  
 <div id="row">  
 <div *\*ngFor*="let item of temp" id="col" class="col-sm-3" style="background-color: whitesmoke;">  
 <h1 id="title" style="text-align: center;"> {{item.itemName}} </h1><br>  
 <p id="address"><label for="Address">Address:</label> {{item.itemAddress }} </p>  
 <p class="price" id="Weight" ><label for="price">Weight:</label> {{item.itemWeight}}<input type="text" style="width: 20%; visibility: hidden" name="username" [(ngModel)]="username"><input type="date" style="width: 20%; visibility: hidden" name="addeddate" [(ngModel)]="updateddate"></p>  
<!-- <input type="text" style="width: 20%;" name="reqweight" id="reqweight" [(ngModel)]="reqweight">-->  
 <p class="price" id="expdate" ><label for="price">Expiry Date:</label> {{item.itemExpiry}} </p>  
 <p class="price" id="price" ><label for="price">Price: $</label> {{item.itemPrice}} </p>  
 <p style="text-align: center;"><button type="button" style="background-color: slategrey; color: black; width: 35%;" (click)="deleteItem(item.\_id, item.itemWeight, username, updateddate)">Request Item</button></p>  
 </div>  
 </div>  
</div>  
  
<div class="container">  
 <div class="modal" id="loginModal" tabindex="-1">  
 <div class="modal-dialog">  
 <div class="modal-content">  
 <div class="modal-header">  
 <button class="close" data-dismiss="modal">&times;</button>  
 <h1 class="modal-title">Login</h1>  
 </div>  
 </div>  
 </div>  
 </div>  
</div>  
  
<script src="requester.component.ts"></script>  
  
</body>  
</html>

**Type Script:**

import { ***Component***, OnInit } from '@angular/core';  
import {ActivatedRoute, NavigationEnd, Router} from '@angular/router';  
import {HttpService} from '../http.service';  
// import {ModalService} from '../modal.service';  
  
@Component({  
 selector: 'app-orgadmin',  
 templateUrl: './requester.component.html',  
 styleUrls: ['./requester.component.css'],  
})  
export class RequesterComponent implements OnInit {  
 public itemCategory;  
 public itemName;  
 public itemAddress;  
 public itemExpiry;  
 public itemWeight;  
 public itemPrice;  
 public username;  
 public updateddate;  
 public searchCat;  
 temp: any;  
 public previousState;  
 arr: any;  
 public reqweight;  
 public bodyText;  
 // @ViewChild('Weight') Weight: ElementRef;  
 // tslint:disable-next-line:max-line-length  
 constructor(private route: ActivatedRoute, private router: Router, private httpService: HttpService) {  
 this.router.events.subscribe((event) => { if ( event instanceof NavigationEnd) {  
 this.arr = event.url.split('/', 3);  
 this.previousState = this.arr[1];  
 }  
 });  
 }  
 ngOnInit() {  
 this.searchCat = 'All';  
 this.getData(this.searchCat);  
 this.username = this.route.snapshot.paramMap.get('username');  
 this.updateddate = new ***Date***();  
 }  
 getData(searchCat) {  
 ***console***.log('Get Data calling');  
 this.httpService.getItemData(searchCat).subscribe(data => {  
 this.temp = data;  
 });  
 }  
 showCategoryCards() {  
 // alert('val is:' + this.itemCategory);  
 this.httpService.getItemData(this.itemCategory).subscribe(data => {  
 this.temp = data;  
 });  
 }  
 // openModal(id: string) {  
 // // alert('hii');  
 // this.modalService.open(id);  
 // }  
 // closeModal(id: string) {  
 // this.modalService.close(id);  
 // }  
 deleteItem(itemId, itemWeight, username, updateddate) {  
 // alert('Item wt: ' + itemWeight + ' Reqested wt: ' + this.reqweight);  
 if (confirm('Are you sure to delete ')) {  
 ***console***.log('Implement delete functionality here');  
 this.httpService.deleteItem(itemId).subscribe(data => {  
 // console.log(data.success + ' ' + data.message);  
 if (data.success) {  
 this.getData(this.searchCat);  
 alert('Item Data deleted from table successfully');  
 this.httpService.upDateHistory(itemId, username, updateddate).subscribe(data1 => {  
 if (data1) {  
 alert('Item Data updated in history table successfully');  
 } else {  
 alert('Error in updating data in history table');  
 }  
 });  
 } else {  
 alert('Error in deleting Item');  
 }  
 });  
 }  
 }  
  
 goToHistory() {  
 this.router.navigate(['/history', this.username, this.previousState]);  
 }  
}

**Routing**

import { ***NgModule*** } from '@angular/core';  
// import { CommonModule } from '@angular/common';  
import {Routes, RouterModule} from '@angular/router';  
import {LoginpageComponent} from './loginpage/loginpage.component';  
import {RegistrationComponent} from './registration/registration.component';  
import {ForgotpasswdComponent} from './forgotpasswd/forgotpasswd.component';  
import {DonarComponent} from './donar/donar.component';  
import {RequesterComponent} from './requester/requester.component';  
import {MainpageComponent} from './mainpage/mainpage.component';  
import {HistoryComponent} from './history/history.component';  
import {ProfileComponent} from './profile/profile.component';  
  
const routes: Routes = [  
 {path: '', redirectTo: '/mainpage', pathMatch: 'full'},  
 {path: 'loginpage', component: LoginpageComponent},  
 {path: 'registration', component: RegistrationComponent},  
 {path: 'forgotpasswd', component: ForgotpasswdComponent},  
 {path: 'profile', component: ProfileComponent},  
 {path: 'donar/:username', component: DonarComponent},  
 {path: 'requester/:username', component: RequesterComponent},  
 {path: 'history/:username/:previousState', component: HistoryComponent},  
 {path: 'mainpage', component: MainpageComponent}];  
  
@NgModule({  
 declarations: [],  
 imports: [  
 RouterModule.*forRoot*(routes)  
 ],  
 exports: [RouterModule]  
})  
export class AppRoutingModule {  
}

**Node module:**

const express = require("express");  
const cors = require("cors");  
const bodyParser = require("body-parser");  
const nodemailer = require("nodemailer");  
const mongoose = require("mongoose");  
const details = require("./details.json");  
const bodyparser = require('body-parser');  
const User = require("./User");  
const Quest = require("./History");  
const ItemData = require("./itemData");  
const History = require("./History");  
  
// const db = mongoose.connect('mongodb://localhost:27017/userData', function (err, response) {  
const db = mongoose.connect('mongodb+srv://roshna:kcdonation123@cluster0-gcp8v.mongodb.net/test?retryWrites=true&w=majority', function (err, response) {  
 if(err)  
 ***console***.log("Error in mongodb connection");  
 else  
 ***console***.log("Mongodb connection added");  
});  
  
const app = express();  
app.use(cors({ origin: "\*" }));  
app.use(bodyParser.json());  
  
app.listen(3000, () => {  
 ***console***.log("The server started on port 3000 !!!!!!");  
});  
  
// app.use(bodyParser.json());  
  
app.get("/", (req, res) => {  
 res.send(  
 "<h1 style='text-align: center'>Wellcome to AI Chatbot <br></h1>"  
 );  
});  
  
app.get("/getRequestersEmail", (req, res) => {  
 // console.log("Get request for all Items");  
 User.find({userType:"Requester"})  
 .exec(function (err, response) {  
 if(err) {  
 ***console***.log("Error in receiving Items");  
 } else {  
 const result = [];  
 response.forEach(function(u) { result.push(u.registerEmail)});  
 res.send(result);  
 }  
 })  
});  
  
app.get("/getUserHistory/:username/:previousState", (req, res) => {  
 // console.log("Get request for all Item History: "+JSON.stringify(req.params.username));  
  
 if (req.params.previousState ==='donar')  
 History.find({donarName: req.params.username})  
 .exec(function (err, itemdatas) {  
 if(err) {  
 ***console***.log("Error in receiving Item History");  
 } else {  
 res.json(itemdatas);  
 }  
 })  
 else  
 History.find({requesterName: req.params.username})  
 .exec(function (err, itemdatas) {  
 if(err) {  
 ***console***.log("Error in receiving Item History");  
 } else {  
 res.json(itemdatas);  
 }  
 })  
  
});  
  
app.delete("/deleteItem/:id", (req, res) => {  
 // ItemData.deleteOne({\_id: req.params.id}, function (err, deletedItem) {  
 ItemData.findByIdAndDelete(req.params.id, function (err, deletedItem) {  
 if(err) {  
 res.send({success: false, message: "Failed to delete an item"});  
 } else {  
 res.send({success: true, message: "Successfully deleted an item"});  
 }  
 })  
})  
  
app.get("/getItemData/:searchCat", (req, res) => {  
 ***console***.log("searchCat :" + req.params.searchCat);  
 if(req.params.searchCat === 'All')  
 ItemData.find({})  
 .exec(function (err, itemdatas) {  
 if(err) {  
 ***console***.log("Error in receiving Items");  
 } else {  
 res.json(itemdatas);  
 }  
 })  
 else  
 ItemData.find({itemCategory: req.params.searchCat})  
 .exec(function (err, itemdatas) {  
 if(err) {  
 ***console***.log("Error in receiving Items");  
 } else {  
 res.json(itemdatas);  
 }  
 })  
});  
app.patch("/upDateHistory", (req, res) => {  
 ***console***.log("request came in update history - body : " + ***JSON***.stringify(req.body.headers));  
 mongoose.set('useFindAndModify', false);  
 // ItemData.deleteOne({\_id: req.params.id}, function (err, updatehistory) {  
 History.findOneAndUpdate({referenceId: req.body.headers.*Id*}, {requesterName: req.body.headers.name, requesterCompDate: req.body.headers.date}, function (err, updatehistory) {  
 if(err) {  
 res.send({success: false, message: "Failed to update a item"});  
 } else {  
 res.send({success: true, message: "Successfully updated a item"});  
 }  
 })  
})  
app.patch("/upDatePassword", (req, res) => {  
 ***console***.log("request came in update password - body : " + ***JSON***.stringify(req.body.headers));  
 mongoose.set('useFindAndModify', false);  
 User.findOneAndUpdate({registerEmail: req.body.headers.emailID}, {registerPassword: req.body.headers.randomNumber}, function (err, upDatePassword) {  
 if(err) {  
 res.send({success: false, message: "Failed to update password"});  
 } else {  
 res.send({success: true, message: "Successfully updated passowrd"});  
 }  
 })  
})  
app.patch("/profUpdate", async (req, res) => {  
 ***console***.log("request came in update profile - body : " + ***JSON***.stringify(req.body.headers));  
 mongoose.set('useFindAndModify', false);  
 const resp = await User.findOne({registerEmail:req.body.headers.exEmail, registerPassword: req.body.headers.currPass});  
 // console.log(resp);  
 if(!resp) {  
 ***console***.log("incorrect details");  
 res.json({  
 success: false,  
 message: "Incorrect user details",  
 status: 500  
 })  
 } else {  
 User.findOneAndUpdate({registerEmail: req.body.headers.exEmail, registerPassword: req.body.headers.currPass}, {registerPassword: req.body.headers.nPass}, function (err, profUpdate) {  
 if(err) {  
 res.json({  
 success: false,  
 message: "Incorrect user details",  
 status: 500  
 })  
 } else {  
 // res.send({success: true, message: "Successfully updated profile"});  
 res.json({  
 success: true,  
 message: "Successfully updated profile",  
 status: 500  
 })  
 }  
 })  
 }  
})  
app.post('/registration', (req, res) => {  
 // console.log(req.body);  
 const userType = req.body.userType;  
 const registerUsername = req.body.registerUsername;  
 const registerEmail = req.body.registerEmail;  
 const registerPassword = req.body.registerPassword;  
  
 const user = new User();  
 user.userType = userType;  
 user.registerUsername = registerUsername;  
 user.registerEmail = registerEmail;  
 user.registerPassword = registerPassword;  
  
 user.save((err, result) => {  
 if(err) {  
 ***console***.log("There is a error while adding in db");  
 res.send({success: false, message: "Failed to add user"});  
 } else {  
 ***console***.log("successfuly saved");  
 res.send({success: true, message: "Successfuly added new user"});  
 }  
 })  
  
});  
  
app.post('/submitItem', (req, res) => {  
 ***console***.log(req.body);  
 const itemCategory = req.body.itemCategory;  
 const itemName = req.body.itemName;  
 const itemAddress = req.body.itemAddress;  
 const itemExpiry = req.body.itemExpdate;  
 const itemWeight = req.body.itemWeight;  
 const itemPrice = req.body.itemPrice;  
  
 const itemData = new ItemData();  
 itemData.itemCategory = itemCategory;  
 itemData.itemName = itemName;  
 itemData.itemAddress = itemAddress;  
 itemData.itemExpiry = itemExpiry;  
 itemData.itemWeight = itemWeight;  
 itemData.itemPrice = itemPrice;  
  
 itemData.save((err, result) => {  
 if(err) {  
 ***console***.log("There is a error while adding in db");  
 res.send({success: false, message: "Failed to add an item"});  
 } else {  
 ***console***.log("successfully saved");  
 ***console***.log("reference ID : " + result.\_id);  
 res.send({success: true, message: "Successfully added a new Item", referenceID:result.\_id});  
 }  
 })  
});  
  
app.post('/addHistory', (req, res) => {  
 ***console***.log('body: ' + ***JSON***.stringify(req.body) + ' headers :' + ***JSON***.stringify(req.headers));  
 mongoose.set('useFindAndModify', false);  
 const itemCategory = req.body.itemCategory;  
 const itemName = req.body.itemName;  
 const itemAddress = req.body.itemAddress;  
 const itemExpiry = req.body.itemExpdate;  
 const itemWeight = req.body.itemWeight;  
 const itemPrice = req.body.itemPrice;  
 const donarName = req.body.username;  
 const addedDate = req.body.addeddate;  
 const referenceID = req.headers.refid;  
  
 // Table to maintain history of Donars and Requesters  
 const history = new History();  
 history.donarName = donarName;  
 history.donarAddedDate = addedDate;  
 history.itemCategory = itemCategory;  
 history.itemName = itemName;  
 history.itemAddress = itemAddress;  
 history.itemExpiry = itemExpiry;  
 history.itemWeight = itemWeight;  
 history.itemPrice = itemPrice;  
 history.requesterName = "";  
 history.requesterCompDate = "";  
 history.referenceId = referenceID;  
 history.save((err, result) => {  
 if(err) {  
 ***console***.log("error in adding history");  
 res.send({success: false, message: "Failed to add a data in history"});  
 } else {  
 // console.log("reference ID : " + result.objectId);  
 res.send({success: true, message: "Successfully added a new data in history"});  
 }  
 })  
});  
  
app.post('/loginpage', async (req, res) => {  
 // console.log("Body: " + req.body);  
 const username = req.body.username;  
 const password = req.body.password;  
 ***console***.log("app.js " + username, password);  
 const resp = await User.findOne({registerUsername:username, registerPassword:password});  
 // console.log(resp);  
 if(!resp) {  
 ***console***.log("incoreect details");  
 res.json({  
 success: false,  
 message: "Incorrect user details",  
 status: 500  
 })  
 // res.send({success: "Incorrect user details", status: 500});  
 } else {  
 ***console***.log("loggin you in: " + resp.userType);  
 res.json({  
 success: true,  
 message: "Successfuly verified the user",  
 status: 200,  
 userType: resp.userType  
 })  
 // res.send({success: "Successfuly verified the user", status: 200, userType: resp.userType});  
 }  
});  
  
app.post('/custExist', async (req, res) => {  
 ***console***.log("Body: " + req.body.reminderEmail);  
 const userEmail = req.body.reminderEmail;  
 // console.log("app.js " + JSON.parse(req));  
 const resp = await User.findOne({registerEmail:userEmail});  
 // console.log(resp);  
 if(!resp) {  
 ***console***.log("incoreect details");  
 res.json({  
 success: false,  
 message: "Incorrect user details",  
 status: 500  
 })  
 // res.send({success: "Incorrect user details", status: 500});  
 } else {  
 ***console***.log("loggin you in: " + resp.userType);  
 res.json({  
 success: true,  
 message: "Successfuly verified the user",  
 status: 200,  
 userType: resp.userType  
 })  
 }  
});  
  
app.post("/sendmail", (req, res) => {  
 // console.log("request came");  
 let user = req.body;  
 sendMail(user, info => {  
 ***console***.log(`The mail has beed send, and the id is ${user}`);  
 res.send(info);  
 });  
});  
async function sendMail(user, callback) {  
 // create reusable transporter object using the default SMTP transport  
 ***console***.log("request came in sendMail");  
 let transporter = nodemailer.createTransport({  
 host: "smtp.gmail.com",  
 port: 587,  
 secure: false, // true for 465, false for other ports  
 auth: {  
 user: details.email,  
 pass: details.password  
 }  
 });  
 let mailOptions='';  
 if(user.subject==='sendNotificationToReq')  
 mailOptions = {  
 from: '"Roshna"<example.gimail.com>', // sender address  
 to: user.email, // list of receivers  
 subject: "KC Donation", // Subject line  
 html: `<h3>Hi,</h3><h3>New item has been added in the donation list.</h3><h3>Kindly, login to website to check for more information.</h3><br>  
 <h3>Thanks & Regards,</h3><h3>KC Donation</h3>`  
 };  
 else  
 mailOptions = {  
 from: '"Roshna"<example.gimail.com>', // sender address  
 to: user.email, // list of receivers  
 subject: "KC Donation", // Subject line  
 html: `<h4>Hi,</h4><h4>You new password is: <strong>${user.randomNum}</strong></h4>  
 <h4>Thanks & Regards,</h4><h4>KC Donation</h4>`  
 };  
 // send mail with defined transport object  
 let info = await transporter.sendMail(mailOptions);  
  
 callback(info);  
}