Team Members:

Roshini Varada-class Id:26

V H N V R S Sai Pavan-class Id:27

There are 4 COMPONENTS.

1.Home component

2.Sign up component

3.Food details Component

And there is a service file named as api.service.ts

Home component:

Homecomponent.html:

<div class="sidenav">

    <div class="login-main-text">

       <h2>Application<br> Login Page</h2>

       <p>Login from here to access.</p>

    </div>

 </div>

 <div class="main">

  <div class="container">

      <div class="col-md-12">

        <br><h2 class="text-center">Welcome to Smart Fitness</h2>

        <br><p class="text-center">Please Login and enter the Name of the food item to Know the calories in the food item</p>

        <div class="underline" (click)="signup()" >Signup</div>

        <div \*ngIf="loginErrorUsename && !loginErrorPswd">

          <div class="errorName" \*ngIf="loginErrorUsename">Please signup to continue</div>

      </div>

      <div \*ngIf="loginErrorPswd">

        <div class="errorName" \*ngIf="loginErrorPswd">WelcomeBack!Please enter the proper password</div>

    </div>

        <form  [formGroup]="LoginForm" (ngSubmit)="Submit()">

          <div class="form-group">

            <label for="UserName">Enter the User Name</label>

            <input formControlName="UserName" placeholder="Enter the user Name" class="form-control" [ngClass]="{ 'is-invalid': submitted && f.UserName.errors }">

            <div \*ngIf="submitted && f.UserName.errors" class="invalid-feedback">

              <div \*ngIf="f.UserName.errors.required">User Name is required</div>

          </div>

        </div>

          <div class="form-group">

            <label for="Password">Enter the Password</label>

            <input formControlName="Password" placeholder="Enter the Password" class="form-control" [ngClass]="{ 'is-invalid': submitted && f.Password.errors || submitted && loginError }">

            <div \*ngIf="submitted && f.Password.errors" class="invalid-feedback">

              <div \*ngIf="f.Password.errors.required">Enter the password</div>

          </div>

          <div \*ngIf="loginErrorPswd" class="invalid-feedback">

            <div \*ngIf="loginErrorPswd">Welcome back! Please enter proper password to continue</div>

        </div>

        </div>

          <div class="row">

              <div class="col">

                  <button id="btnSubmit" class="btn btn-black addmargin" type="submit" (click)="signin()">Sign in</button>

              </div>

            </div>

        </form>

      </div>

      </div>

</div>

Homecomponent.css  
body {

  font-family: "Lato", sans-serif;

}

.main-head{

  height: 150px;

  background: #FFF;

}

.sidenav {

  height: 100%;

  background-color: #000;

  overflow-x: hidden;

  padding-top: 20px;

}

.main {

  padding: 0px 10px;

}

@media screen and (max-height: 450px) {

  .sidenav {padding-top: 15px;}

}

@media screen and (max-width: 450px) {

  .login-form{

      margin-top: 10%;

  }

  .register-form{

      margin-top: 10%;

  }

}

@media screen and (min-width: 768px){

  .main{

      margin-left: 40%;

  }

  .sidenav{

      width: 40%;

      position: fixed;

      z-index: 1;

      top: 0;

      left: 0;

  }

  .login-form{

      margin-top: 80%;

  }

  .register-form{

      margin-top: 20%;

  }

}

.errorName{

 color:red;

}

.addmargin{

  margin-left:114px;

}

.addmarginSubmit{

  margin-left:199px;

}

.login-main-text{

  margin-top: 20%;

  padding: 60px;

  color: #fff;

}

.login-main-text h2{

  font-weight: 300;

}

.btn-black{

  background-color: #000 !important;

  color: #fff;

}

Homecomponent.ts

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

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

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

import { analyzeAndValidateNgModules } from '@angular/compiler';

@Component({

  selector: 'app-home',

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

  styleUrls: ['./home.component.css']

})

export class HomeComponent implements OnInit {

  loginError: boolean;

  enterDetailsError=false;

  jsontypedata: {

    UserName: string;

    Password: string;

    RetypePswd:string

}[]

  no\_loop=false;

  userAlreadyexist=false;

  loop\_ended=false;

  loginErrorUsename=false;

  loginErrorPswd: boolean;

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

  LoginForm: FormGroup;

  details:any;

  data:any;

  submitted = false;

  show=false;

  jsondata:any;

  users:any;

  ngOnInit() {

    this.LoginForm = this.formBuilder.group({

     UserName:['',Validators.required],

      Password: ['', Validators.required]

    }, {

        //validator: MustMatch('Password', 'RetypePswd')

    });

    }

    get f() { return this.LoginForm.controls; }

    login(){

      for(let i=0;i<this.jsontypedata.length;i++){

        if(this.LoginForm.value.UserName==this.jsontypedata[i].UserName){

         if(this.LoginForm.value.Password==this.jsontypedata[i].Password){

           this.loginError=false;

           this.router.navigate(['details']);

         }else{

            this.loginErrorPswd=true;

         }

        }else{

         this.loginErrorUsename=true;

        }

      }

    }

    signup(){

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

    }

    Submit(){

      this.submitted = true;

      if (this.LoginForm.invalid) {

        return;

    }else{

      this.data= JSON.stringify(this.LoginForm.value);

      if(this.LoginForm.value.UserName=="" || this.LoginForm.value.Password==""){

        this.enterDetailsError=true;

      }else{

        this.jsondata= localStorage.getItem("dataLogin");

        this.jsontypedata=JSON.parse(this.jsondata);

              this.login();

      }

    }

    }

    signin() {

    this.show=false;

    }

}

Sign up component.html

<div class="sidenav">

    <div class="login-main-text">

       <h2>Application<br> Signup Page</h2>

       <p>Please register from here to access.</p>

    </div>

 </div>

 <div class="main">

    <div class="container">

        <div class="col-md-12">

          <br><h2 class="text-center">Welcome to Smart Fitness</h2>

          <br><p class="text-center">Please Sign up and enter the Name of the food item to Know the calories in the food item</p>

         <div  class="underline" (click)="login()" >Login</div>

          <div \*ngIf="userAlreadyexist">

            Welcome back!  Please Login to continue

           </div>

          <form  [formGroup]="LoginForm" (ngSubmit)="Submit()">

            <div class="form-group">

              <label for="UserName">Enter the User Name</label>

              <input formControlName="UserName" placeholder="Enter the user Name" class="form-control" [ngClass]="{ 'is-invalid': submitted && f.UserName.errors }">

              <div \*ngIf="submitted && f.UserName.errors" class="invalid-feedback">

                <div \*ngIf="f.UserName.errors.required">User Name is required</div>

            </div>

          </div>

            <div class="form-group">

              <label for="Password">Enter the Password</label>

              <input formControlName="Password" placeholder="Enter the Password" class="form-control" [ngClass]="{ 'is-invalid': submitted && f.Password.errors || submitted && loginError }">

              <div \*ngIf="submitted && f.Password.errors" class="invalid-feedback">

                <div \*ngIf="f.Password.errors.required">Enter the password</div>

            </div>

          </div>

            <div class="form-group">

                <label for="RetypePswd">Enter the Password Again</label>

                <input formControlName="RetypePswd" placeholder="Enter the Password Again" class="form-control"  [ngClass]="{ 'is-invalid': submitted && f.RetypePswd.errors }">

                <div \*ngIf="submitted && f.RetypePswd.errors" class="invalid-feedback">

                  <div \*ngIf="f.RetypePswd.errors.required">Type the password Again</div>

              </div>

            </div>

            <div class="row">

                <div class="col">

                  <button class="btn btn-secondary addmargin" type="button" type="submit" (click)="Register()">Sign up</button>

                  </div>

              </div>

          </form>

        </div>

        </div>

 </div>

Signupcomponent.css

body {

    font-family: "Lato", sans-serif;

  }

  .main-head{

    height: 150px;

    background: #FFF;

  }

  .sidenav {

    height: 100%;

    background-color: #000;

    overflow-x: hidden;

    padding-top: 20px;

  }

  .main {

    padding: 0px 10px;

  }

  @media screen and (max-height: 450px) {

    .sidenav {padding-top: 15px;}

  }

  @media screen and (max-width: 450px) {

    .login-form{

        margin-top: 10%;

    }

    .register-form{

        margin-top: 10%;

    }

  }

  @media screen and (min-width: 768px){

    .main{

        margin-left: 40%;

    }

    .sidenav{

        width: 40%;

        position: fixed;

        z-index: 1;

        top: 0;

        left: 0;

    }

    .login-form{

        margin-top: 80%;

    }

    .register-form{

        margin-top: 20%;

    }

  }

  .addmargin{

    margin-left:114px;

  }

  .addmarginSubmit{

    margin-left:199px;

  }

  .underline{

  text-decoration:underline;

  }

  .login-main-text{

    margin-top: 20%;

    padding: 60px;

    color: #fff;

  }

  .login-main-text h2{

    font-weight: 300;

  }

  .btn-black{

    background-color: #000 !important;

    color: #fff;

  }

Signupcomponent.ts

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

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

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

@Component({

  selector: 'app-signup',

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

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

})

export class SignupComponent implements OnInit {

  loginError: boolean;

  enterDetailsError=false;

  jsontypedata: {

    UserName: string;

    Password: string;

    RetypePswd:string

}[]

  no\_loop=false;

  userAlreadyexist=false;

  loop\_ended=false;

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

  LoginForm: FormGroup;

  details:any;

  data:any;

  submitted = false;

  show=false;

  jsondata:any;

  users:any;

  ngOnInit() {

    this.LoginForm = this.formBuilder.group({

     UserName:['',Validators.required],

      Password: ['', Validators.required],

      RetypePswd:['',Validators.required]

    }, {

        //validator: MustMatch('Password', 'RetypePswd')

    });

    }

    get f() { return this.LoginForm.controls; }

    login(){

      this.router.navigate(['homeDetails']);

    }

    signup(){

      if(this.jsontypedata){

        for(let i=0;i<this.jsontypedata.length;i++){

            if(this.LoginForm.value.UserName==this.jsontypedata[i].UserName){

              this.loop\_ended=true;

            }

        }

        if( this.loop\_ended!=true){

          this.jsontypedata.push(JSON.parse(this.data));

          localStorage.setItem("dataLogin",JSON.stringify(this.jsontypedata));

          this.router.navigate(['details']);

        }else{

          this.userAlreadyexist=true;

          this.LoginForm.reset();

        }

       }else{

        this.jsontypedata=[ {

          UserName: "Roshini",

          Password: "1234",

          RetypePswd:"1234"

      }, {

        UserName: "Roshini2",

        Password: "12345",

        RetypePswd:"12345"

    },]

        for(let i=0;i<this.jsontypedata.length;i++){

          if(this.no\_loop!=true){

            if(this.LoginForm.value.UserName==this.jsontypedata[i].UserName){

              this.userAlreadyexist=true;

              this.no\_loop=true;

            }else{

              this.jsontypedata.push(JSON.parse(this.data));

              localStorage.setItem("dataLogin",JSON.stringify(this.jsontypedata));

              this.userAlreadyexist=false;

              this.no\_loop=true;

              this.router.navigate(['details']);

            }

          }

        }

       }

    }

    Submit(){

      this.submitted = true;

      if (this.LoginForm.invalid) {

        return;

    }else{

      this.data= JSON.stringify(this.LoginForm.value);

      if(this.LoginForm.value.UserName=="" || this.LoginForm.value.Password==""){

        this.enterDetailsError=true;

      }else{

        this.jsondata= localStorage.getItem("dataLogin");

        this.jsontypedata=JSON.parse(this.jsondata);

        this.signup();

      }

    }

    }

    Register() {

        this.show=true;

    }

}

DetailsComponent.html

<div class="sidenav">

    <div class="login-main-text">

       <h2>Food Calories<br> Details Page</h2>

       <p>Search to check inorder to be fit</p>

    </div>

 </div>

    <div class="main">

        <div>

            <div class="col-md-12">

                    <br><h2 class="text-center">Welcome</h2>

                    <br><p class="text-center">Please enter the Name of the food item</p>

                    <div class="text-center">

                    <label for="ItemName">Enter the Item Name</label>

                    <input class="marginadd"  [(ngModel)] ="ItemName" name="ItemName"  id="ItemName" type="text">

                    </div>

                    <div class="row text-center">

                        <div class="col">

                        <button class="btn btn-secondary" type="button" type="submit" (click)="Search()">Search</button>

                        </div>

                    </div>

<div class="card container" \*ngIf="showtable">

    <div class="card-header" style="text-align: center;">

      <b> Recipe Details</b>

    </div>

    <div class="card-body padding">

        <div \*ngFor="let x of detailsresult.hits; let i= index">

          <div class="row">

            <div class="card col-md-12" \*ngIf="i<10">

              <div class="card-header"><b>{{detailsresult.hits[i].fields.item\_name}}</b></div>

              <div class="card-body">

                <table>

                  <tr >

                    <td style="text-align: center;">Calories </td>

                    <td >:--></td>

                    <td>{{detailsresult.hits[i].fields.nf\_calories}}</td>

                  </tr>

                  <tr>

                    <td style="text-align: center;">Total Fat </td>

                    <td>:--></td>

                    <td>{{detailsresult.hits[i].fields.nf\_total\_fat}}</td>

                  </tr>

                  <tr>

                    <td style="text-align: center;">Serving weight(gms)</td>

                    <td>:--></td>

                    <td>{{detailsresult.hits[i].fields.nf\_serving\_weight\_grams}}</td>

                  </tr>

                </table>

              </div>

            </div>

          </div>

        </div>

    </div>

  </div>

            </div>

       </div>

    </div>

DetailsComponent.css

.container {

    box-shadow: rgba(0,0,0,.24) 0px 2px 2px, rgba(0,0,0,.12) 0px 0px 2px;

    border: 1px solid rgba(0,0,0,.03);

    background-color: whitesmoke;

  }

  form {

    display: flex;

  }

  .card-body {

    padding:0.25rem

  }

  table {

    width: 100%;

  }

 .marginadd{

    margin-bottom: 12px;

 }

  body {

    font-family: "Lato", sans-serif;

  }

  .main-head{

    height: 150px;

    background: #FFF;

  }

  .sidenav {

    height: 100%;

    background-color: #000;

    overflow-x: hidden;

    padding-top: 20px;

  }

  .main {

    padding: 0px 10px;

  }

  @media screen and (max-height: 450px) {

    .sidenav {padding-top: 15px;}

  }

  @media screen and (max-width: 450px) {

    .login-form{

        margin-top: 10%;

    }

    .register-form{

        margin-top: 10%;

    }

  }

  @media screen and (min-width: 768px){

    .main{

        margin-left: 40%;

    }

    .sidenav{

        width: 40%;

        position: fixed;

        z-index: 1;

        top: 0;

        left: 0;

    }

    .login-form{

        margin-top: 80%;

    }

    .register-form{

        margin-top: 20%;

    }

  }

  .addmargin{

    margin-left:114px;

  }

  .addmarginSubmit{

    margin-left:199px;

  }

  .login-main-text{

    margin-top: 20%;

    padding: 60px;

    color: #fff;

  }

  .login-main-text h2{

    font-weight: 300;

  }

  .btn-black{

    background-color: #000 !important;

    color: #fff;

  }

DetailsComponent.ts

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

import { ApiService } from '../api.service';

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

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

declare const responsiveVoice;

@Component({

  selector: 'app-details',

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

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

})

export class DetailsComponent implements OnInit {

  ItemName="";

  detailsresult: any;

  showtable=false;

  constructor(private apiService:ApiService ,private formBuilder: FormBuilder,private router: Router) { }

  RecipeForm: any;

  ngOnInit() {

  }

  Search(){

    responsiveVoice.speak(this.ItemName, "UK English Male", {volume: 1});

  this.apiService.displayDetails(this.ItemName)

      .subscribe( data => {

        this.detailsresult=data;

        this.showtable=true;

      });

}

}

Apiservice.ts

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

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

import { throwError } from 'rxjs';

import { catchError } from 'rxjs/operators';

@Injectable({

  providedIn: 'root'

})

export class ApiService {

  constructor(private httpClient: HttpClient) { }

  handleError(error: HttpErrorResponse) {

    let errorMessage = 'Unknown error!';

    if (error.error instanceof ErrorEvent) {

      errorMessage = `Error: ${error.error.message}`;

    } else {

      errorMessage = `Error Code: ${error.status}\nMessage: ${error.message}`;

    }

    return throwError(errorMessage);

  }

  public displayDetails(name){

    return this.httpClient.get<any>("https://api.nutritionix.com/v1\_1/search/' "+name+" '?results=0:1&fields=\*&appId=ea1a433d&appKey=9009e40b76a78c86426713c2d145bca2").pipe(catchError(this.handleError));

  }

}

Appmodule.ts

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

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

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

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

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

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

import { HomeComponent } from './home/home.component';

import { DetailsComponent } from './details/details.component';

import { SignupComponent } from './signup/signup.component';

@NgModule({

  declarations: [

    AppComponent,

    HomeComponent,

    DetailsComponent,

    SignupComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    HttpClientModule,

    FormsModule,

    ReactiveFormsModule,

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

Approutingmodule.ts

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

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

import { HomeComponent } from './home/home.component';

import { DetailsComponent } from './details/details.component';

import { SignupComponent } from './signup/signup.component';

const routes: Routes = [

  { path : '', component : HomeComponent},

  { path : 'signup', component : SignupComponent},

  {path:'details',component:DetailsComponent},

  { path: 'homeDetails', component: HomeComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

index.html:

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>Calrories</title>

  <base href="/">

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

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

  <meta http-equiv="X-UA-Compatible" content="ie=edge">

  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

  <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">

  <script src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

  <script src="//code.jquery.com/jquery-1.11.1.min.js"></script>

  <script src="https://code.responsivevoice.org/responsivevoice.js?key=6zku89hI"></script>

</head>

<body>

  <app-root></app-root>

</body>

</html>