**Project objective:**

Kitchen Story is an e-commerce portal that lets people shop basic food items on their website. The website needs to have the following features:

* A search form in the home page to allow entry of the food items to be purchased by the customer.
* Based on item details entered, it will show available food items with price.
* Once a person selects an item to purchase, they will be redirected to the list of available items. In the next page, they are shown the complete breakout of the order and details of the payment to be made in the payment gateway. When payment is done, they are shown a confirmation page with details of the order.

**Source Code**

Admin.ts

export class Admin{

    username:string;

    password:string;

}

admin.service.ts

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

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

@Injectable({

  providedIn: 'root'

})

export class AdminService {

  constructor(private router:Router) { }

  admin= {"username":"admin","password":"admin123"}

  login(uname:string,password:string)

  {

      if(uname===this.admin.username && password===this.admin.password)

      {

        this.router.navigateByUrl('adash');

        return true;

      }

      else

      return false;

  }

  changepwd(uname:string,password:string)

  {

    if(uname===this.admin.username)

    {

      this.admin.password=password;

      return true;

    }

    else

    return false;

  }

}

Product.ts

export class Product{

    id:number;

    name:string;

    price:number;

    description:string;

}

product.service.ts

import { getQueryPredicate } from '@angular/compiler/src/render3/view/util';

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

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

import { BehaviorSubject, of } from 'rxjs';

import { Product } from './Product';

@Injectable({

  providedIn: 'root'

})

export class ProductService {

  products:Product[]=[];

  pro:Product[]=[];

  pro1:Product[]=[];

  public subject=new BehaviorSubject<Product[]>([]);

  public subject1=new BehaviorSubject<Number>(0);

  constructor(private router:Router) { }

  addProduct(product:any)

  {

    this.products.push(product);

  }

  getAllProducts():Product[]

  {

    return this.products;

  }

  deleteProduct(id: number)

  {

    this.products.splice(id,1);

  }

  search(name: String)

  {

    this.pro = this.getAllProducts().filter(x => x.name == name)

    return this.pro;

  }

  purchase(id: number)

  {

    this.pro1=this.getAllProducts().filter(y => y.id == id)

    this.subject.next(this.pro1);

  }

  orderdetails(price: number,n: number)

  {

    var t=price\*n;

    this.subject1.next(t);

  }

}

User.ts

export class User{

    username:string;

    password:string;

}

user.service.ts

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

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

@Injectable({

  providedIn: 'root'

})

export class UserService {

  constructor(private router:Router) { }

  user= {"username":"user","password":"user123"}

  login(uname:string,password:string)

  {

      if(uname===this.user.username && password===this.user.password)

      {

        this.router.navigateByUrl('udash');

        return true;

      }

      else

      return false;

  }

  changeupwd(uname:string,password:string)

  {

    if(uname===this.user.username)

    {

      this.user.password=password;

      return true;

    }

    else

    return false;

  }

}

app-routing.module.ts

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

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

import { AddproductComponent } from './addproduct/addproduct.component';

import { AdminComponent } from './admin/admin.component';

import { AdmindashboardComponent } from './admindashboard/admindashboard.component';

import { AdminlogoutComponent } from './adminlogout/adminlogout.component';

import { ChangepasswordComponent } from './changepassword/changepassword.component';

import { ChangepassworduserComponent } from './changepassworduser/changepassworduser.component';

import { ListComponent } from './list/list.component';

import { OrderdetailsComponent } from './orderdetails/orderdetails.component';

import { PurchaseComponent } from './purchase/purchase.component';

import { RemoveproductComponent } from './removeproduct/removeproduct.component';

import { SearchproductComponent } from './searchproduct/searchproduct.component';

import { UserComponent } from './user/user.component';

import { UserdashboardComponent } from './userdashboard/userdashboard.component';

import { UserlogoutComponent } from './userlogout/userlogout.component';

const routes: Routes = [

  {"path":"admin",component:AdminComponent},

  {"path":"adash",component:AdmindashboardComponent},

  {"path":"adash/add",component:AddproductComponent},

  {"path":"adash/list",component:ListComponent},

  {"path":"adash/list/remove",component:RemoveproductComponent},

  {"path":"adash/pwd",component:ChangepasswordComponent},

  {"path":"adash/alogout",component:AdminlogoutComponent},

  {"path":"user",component:UserComponent},

  {"path":"udash",component:UserdashboardComponent},

  {"path":"udash/upwd",component:ChangepassworduserComponent},

  {"path":"udash/ulogout",component:UserlogoutComponent},

  {"path":"udash/search",component:SearchproductComponent},

  {"path":"udash/search/purc",component:PurchaseComponent},

  {"path":"udash/search/purc/order",component:OrderdetailsComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

app.component.css

#c1{

    text-align: center;

    background-color: beige;

    position:absolute;

    top:0px;

    right:0px;

    bottom:0px;

    left:0px;

    font-size: large;

    font-family: Georgia, 'Times New Roman', Times, serif;

}

app.component.html

<div id="c1" class="container-fluid">

    <h1>Welcome to KitchenStory</h1>

    <a routerLink="admin">Admin Login</a>|

    <a routerLink="user">User Login</a>

    <router-outlet></router-outlet>

    <img src="./assets/images/bg.jpg"  width="100%">

</div>

addproduct.component.ts

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

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

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

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

import { ProductService } from '../product.service';

@Component({

  selector: 'app-addproduct',

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

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

})

export class AddproductComponent implements OnInit {

  constructor(private formBuilder: FormBuilder,

    private service: ProductService, private router:Router) { }

  addForm: FormGroup;

  product:Product=new Product();

  submitted: boolean = false;

  ngOnInit(): void {

    this.addForm = this.formBuilder.group({

      id: ['', Validators.required],

      name: ['', Validators.required],

      price: ['', Validators.required],

      description: ['', Validators.required]

    })

  }

  get f() {

    return this.addForm.controls;

  }

  onSubmit() {

    this.submitted = true;

    if (this.addForm.invalid)

      return;

    else {

      console.log(this.product);

      this.service.addProduct(this.product);

      this.router.navigateByUrl('adash/list');

    }

  }

  getf()

  {

    return this.addForm.controls;

  }

}

addproduct.component.html

<div class="jumbotron">

    <div class="container">

        <div class="row">

            <div class="col-md-6 offset-md-3">

                <h2>Add Product Form</h2>

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

                    <div class="form-group">

                        <label>Id:</label>

                        <input type="text" [(ngModel)]="product.id" formControlName="id" class="form-control"

                        [ngClass]="{'is-invalid':submitted && f['id'].errors}">

                        <div \*ngIf="submitted && f['id'].errors" class="invalid-feedback">

                            <div \*ngIf="f['id'].errors['required']" style="color: red;">

                                Required

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <label>Name:</label>

                        <input type="text" [(ngModel)]="product.name" formControlName="name" class="form-control"

                        [ngClass]="{'is-invalid':submitted && f['name'].errors}">

                        <div \*ngIf="submitted && f['name'].errors" class="invalid-feedback">

                            <div \*ngIf="f['name'].errors['required']" style="color: red;">

                                Required

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <label>Price:</label>

                        <input type="text" [(ngModel)]="product.price" formControlName="price" class="form-control"

                        [ngClass]="{'is-invalid':submitted && f['price'].errors}">

                        <div \*ngIf="submitted && f['price'].errors" class="invalid-feedback">

                            <div \*ngIf="f['price'].errors['required']" style="color: red;">

                                Required

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <label>Description:</label>

                        <input type="text" [(ngModel)]="product.description" formControlName="description" class="form-control"

                        [ngClass]="{'is-invalid':submitted && f['description'].errors}">

                        <div \*ngIf="submitted && f['description'].errors" class="invalid-feedback">

                            <div \*ngIf="f['description'].errors['required']" style="color: red;">

                                Required

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <button class="btn btn-primary">Add Product</button>

                    </div>

                </form>

            </div>

        </div>

    </div>

</div>

admin.component.ts

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

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

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

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

import { AdminService } from '../admin.service';

@Component({

  selector: 'app-admin',

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

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

})

export class AdminComponent implements OnInit {

  constructor(private formBuilder: FormBuilder,

    private service: AdminService, private router:Router) { }

  loginForm: FormGroup;

  admin:Admin=new Admin();

  a:boolean;

  submitted: boolean = false;

  ngOnInit(): void {

    this.loginForm = this.formBuilder.group({

      uname: ['', Validators.required],

      password: ['', Validators.required]

    })

  }

  get f() {

    return this.loginForm.controls;

  }

  onSubmit() {

    this.submitted = true;

    if (this.loginForm.invalid)

      return;

    else {

      this.a=this.service.login(this.admin.username,this.admin.password);

    }

  }

  getf()

  {

    return this.loginForm.controls;

  }

}

admin.component.html

<div class="jumbotron">

    <div class="container">

        <div class="row">

            <div class="col-md-6 offset-md-3">

                <h2>Admin Login</h2>

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

                    <div class="form-group">

                        <label>Enter Username:</label>

                        <input type="text" placeholder="Username" [(ngModel)]="admin.username" formControlName="uname" class="form-control"

                            [ngClass]="{'is-invalid':submitted && f['uname'].errors}">

                        <div \*ngIf="submitted && f['uname'].errors" class="invalid-feedback">

                            <div \*ngIf="f['uname'].errors['required']" style="color: red;">

                                Username Required!

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <label>Enter Password:</label>

                        <input type="password" placeholder="Password" [(ngModel)]="admin.password" formControlName="password" class="form-control"

                            [ngClass]="{'is-invalid':submitted && f['password'].errors}">

                        <div \*ngIf="submitted && f['password'].errors" class="invalid-feedback">

                            <div \*ngIf="f['password'].errors['required']" style="color: red;">

                                Password Required!

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <button class="btn btn-primary">Login</button>

                    </div>

                </form>

            </div>

        </div>

    </div>

</div>

<div \*ngIf="a==false">

    <h3>Invalid Credentials! Please try again</h3>

</div>

admindashboard.component.html

<br><br>

<h1> Admin Dashboard </h1>

<a routerLink="add">Add Product</a>|

<a routerLink="list">List Products</a>|

<a routerLink="pwd">Change Password</a>|

<a routerLink="alogout">Logout</a>

<router-outlet></router-outlet>

adminlogout.component.html

<h1>Logged out successfully</h1>

changepassword.component.ts

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

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

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

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

import { AdminService } from '../admin.service';

@Component({

  selector: 'app-changepassword',

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

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

})

export class ChangepasswordComponent implements OnInit {

  constructor(private formBuilder: FormBuilder,

    private service: AdminService, private router:Router) { }

  chgpwdForm: FormGroup;

  admin:Admin=new Admin();

  a:boolean;

  submitted: boolean = false;

  ngOnInit(): void {

    this.chgpwdForm = this.formBuilder.group({

      uname: ['', Validators.required],

      password: ['', Validators.required]

    })

  }

  get f() {

    return this.chgpwdForm.controls;

  }

  onSubmit() {

    this.submitted = true;

    if (this.chgpwdForm.invalid)

      return;

    else {

      this.a=this.service.changepwd(this.admin.username,this.admin.password);

    }

  }

  getf()

  {

    return this.chgpwdForm.controls;

  }

}

changepassword.component.html

<div class="jumbotron">

    <div class="container">

        <div class="row">

            <div class="col-md-6 offset-md-3">

                <h2>Change Password</h2>

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

                    <div class="form-group">

                        <label>Enter Username:</label>

                        <input type="text" placeholder="Existing Username" [(ngModel)]="admin.username" formControlName="uname" class="form-control"

                        [ngClass]="{'is-invalid':submitted && f['uname'].errors}">

                        <div \*ngIf="submitted && f['uname'].errors" class="invalid-feedback">

                            <div \*ngIf="f['uname'].errors['required']" style="color: red;">

                                Username Required!

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <label>Enter New Password:</label>

                        <input type="password" placeholder="New Password" [(ngModel)]="admin.password" formControlName="password" class="form-control"

                        [ngClass]="{'is-invalid':submitted && f['password'].errors}">

                        <div \*ngIf="submitted && f['password'].errors" class="invalid-feedback">

                            <div \*ngIf="f['password'].errors['required']" style="color: red;">

                                Password Required!

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <button class="btn btn-primary">Change Password</button>

                    </div>

                </form>

            </div>

        </div>

    </div>

</div>

<div \*ngIf="a==false;">

    <h3>Invalid Username! Please try again</h3>

</div>

<div \*ngIf="a==true;">

    <h3>Password Changed, Please Login again </h3>

</div>

changepassworduser.component.ts

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

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

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

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

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

@Component({

  selector: 'app-changepassworduser',

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

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

})

export class ChangepassworduserComponent implements OnInit {

  constructor(private formBuilder: FormBuilder,

    private service: UserService, private router:Router) { }

  chgupwdForm: FormGroup;

  user:User=new User();

  a:boolean;

  submitted: boolean = false;

  ngOnInit(): void {

    this.chgupwdForm = this.formBuilder.group({

      uname: ['', Validators.required],

      password: ['', Validators.required]

    })

  }

  get f() {

    return this.chgupwdForm.controls;

  }

  onSubmit() {

    this.submitted = true;

    if (this.chgupwdForm.invalid)

      return;

    else {

      this.a=this.service.changeupwd(this.user.username,this.user.password);

    }

  }

  getf()

  {

    return this.chgupwdForm.controls;

  }

}

changepassworduser.component.html

<div class="jumbotron">

    <div class="container">

        <div class="row">

            <div class="col-md-6 offset-md-3">

                <h2>Change User Password</h2>

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

                    <div class="form-group">

                        <label>Enter Username:</label>

                        <input type="text" placeholder="Existing Username" [(ngModel)]="user.username" formControlName="uname" class="form-control"

                        [ngClass]="{'is-invalid':submitted && f['uname'].errors}">

                        <div \*ngIf="submitted && f['uname'].errors" class="invalid-feedback">

                            <div \*ngIf="f['uname'].errors['required']" style="color: red;">

                                Username Required

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <label>Enter New Password:</label>

                        <input type="password" placeholder="Password" [(ngModel)]="user.password" formControlName="password" class="form-control"

                        [ngClass]="{'is-invalid':submitted && f['password'].errors}">

                        <div \*ngIf="submitted && f['password'].errors" class="invalid-feedback">

                            <div \*ngIf="f['password'].errors['required']" style="color: red;">

                                Password Required

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <button class="btn btn-primary">Change Password</button>

                    </div>

                </form>

            </div>

        </div>

    </div>

</div>

<div \*ngIf="a==false;">

    <h3>Invalid Username! Please try again</h3>

</div>

<div \*ngIf="a==true;">

    <h3>Password Changed, Please Login again </h3>

</div>

list.component.ts

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

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

import { ProductService } from '../product.service';

@Component({

  selector: 'app-list',

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

  styleUrls: ['./list.component.css']

})

export class ListComponent implements OnInit {

  constructor(private service:ProductService) { }

  products:Product[];

  ngOnInit(): void {

      this.products=this.service.getAllProducts();

  }

  removeproduct(id: number)

  {

    this.service.deleteProduct(id-1);

  }

}

list.component.html

<h1>List of Products</h1>

<table class="table">

    <thead>

        <tr>

            <th>Product ID</th>

            <th>Product Name</th>

            <th>Price</th>

            <th>Description</th>

        </tr>

    </thead>

    <tbody \*ngFor="let p of products" class="container-fluid">

        <tr>

            <td>{{p.id}}</td>

            <td>{{p.name}}</td>

            <td>{{p.price}}</td>

            <td>{{p.description}}</td>

            <td> <button routerLink="remove" (click)=removeproduct(p.id)>Remove</button> </td>

        </tr>

    </tbody>

</table>

<router-outlet></router-outlet>

orderdetails.component.ts

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

import { Observable, share } from 'rxjs';

import { ProductService } from '../product.service';

@Component({

  selector: 'app-orderdetails',

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

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

})

export class OrderdetailsComponent implements OnInit {

  public total: Observable<Number>;

  constructor(private service:ProductService) {

    this.total=this.service.subject1.pipe(share());

  }

  ngOnInit(): void {

  }

}

orderdetails.component.html

<h3> Total Price </h3>

<p \*ngIf="total">

    {{ total | async }}

</p>

<h3> Purchase Done Successfully! </h3>

purchase.component.ts

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

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

import { Observable, share } from 'rxjs';

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

import { ProductService } from '../product.service';

@Component({

  selector: 'app-purchase',

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

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

})

export class PurchaseComponent implements OnInit {

  public products: Observable<Product[]>;

  constructor(private service: ProductService, private router:Router) {

    this.products=this.service.subject.pipe(share());

  }

  ngOnInit(): void {

  }

  tprice(price: number,n: number)

  {

    this.service.orderdetails(price,n);

    this.router.navigateByUrl('udash/search/purc/order');

  }

}

purchase.component.html

<h1> Order Details </h1>

<table class="table">

    <thead>

        <tr>

            <th>Product Name</th>

            <th>Price</th>

            <th>Description</th>

        </tr>

    </thead>

    <tbody \*ngFor="let p of products | async" class="container-fluid">

        <tr>

            <td>{{p.name}}</td>

            <td>{{p.price}}</td>

            <td>{{p.description}}</td>

            Enter No. of Products: <input #n type="number" name="n"><br>

            <button (click)=tprice(p.price,n.valueAsNumber)>Click here to finalise Booking</button>

        </tr>

    </tbody>

</table>

removeproduct.component.ts

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

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

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

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

import { ProductService } from '../product.service';

@Component({

  selector: 'app-removeproduct',

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

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

})

export class RemoveproductComponent implements OnInit {

  constructor(private router:Router) { }

  ngOnInit(): void {

    this.router.navigateByUrl('adash/list');

  }

}

searchproduct.component.ts

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

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

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

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

import { ProductService } from '../product.service';

@Component({

  selector: 'app-searchproduct',

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

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

})

export class SearchproductComponent implements OnInit {

  constructor(private formBuilder: FormBuilder,

    private service: ProductService, private router:Router) { }

  searchForm: FormGroup;

  product:Product=new Product();

  pro:Product[];

  submitted: boolean = false;

  ngOnInit(): void {

    this.searchForm = this.formBuilder.group({

      name: ['', Validators.required]

    })

  }

  get f() {

    return this.searchForm.controls;

  }

  onSubmit() {

    this.submitted = true;

    if (this.searchForm.invalid)

      return;

    else {

      this.pro=this.service.search(this.product.name);

      console.log(this.pro);

    }

  }

  getf()

  {

    return this.searchForm.controls;

  }

  showpurchase(id: number)

  {

      this.service.purchase(id);

  }

}

searchproduct.component.html

<div class="jumbotron">

    <div class="container">

        <div class="row">

            <div class="col-md-6 offset-md-3">

                <h2>Search Product</h2>

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

                    <div class="form-group">

                        <label>Enter Product Name:</label>

                        <input type="text" placeholder="Product Name" [(ngModel)]="product.name" formControlName="name"

                            class="form-control" [ngClass]="{'is-invalid':submitted && f['name'].errors}">

                        <div \*ngIf="submitted && f['name'].errors" class="invalid-feedback">

                            <div \*ngIf="f['name'].errors['required']" style="color: red;">

                                Required

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <button class="btn btn-primary">Search</button>

                    </div>

                </form>

            </div>

        </div>

    </div>

</div>

<br><br><br>

<h1>List of Products</h1>

<table class="table">

    <thead>

        <tr>

            <th>Product Name</th>

            <th>Price</th>

            <th>Description</th>

        </tr>

    </thead>

    <tbody \*ngFor="let p of pro" class="container-fluid">

        <tr>

            <td>{{p.name}}</td>

            <td>{{p.price}}</td>

            <td>{{p.description}}</td>

            <td> <button routerLink="purc" (click)=showpurchase(p.id)>Purchase</button> </td>

        </tr>

    </tbody>

</table>

<router-outlet></router-outlet>

user.component.ts

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

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

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

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

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

@Component({

  selector: 'app-user',

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

  styleUrls: ['./user.component.css']

})

export class UserComponent implements OnInit {

  constructor(private formBuilder: FormBuilder,

    private service: UserService, private router:Router) { }

  uloginForm: FormGroup;

  user:User=new User();

  a:boolean;

  submitted: boolean = false;

  ngOnInit(): void {

    this.uloginForm = this.formBuilder.group({

      uname: ['', Validators.required],

      password: ['', Validators.required]

    })

  }

  get f() {

    return this.uloginForm.controls;

  }

  onSubmit() {

    this.submitted = true;

    if (this.uloginForm.invalid)

      return;

    else {

      this.a=this.service.login(this.user.username,this.user.password);

    }

  }

  getf()

  {

    return this.uloginForm.controls;

  }

}

user.component.html

<div class="jumbotron">

    <div class="container">

        <div class="row">

            <div class="col-md-6 offset-md-3">

                <h2>User Login</h2>

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

                    <div class="form-group">

                        <label>Enter Username:</label>

                        <input type="text" placeholder="Username" [(ngModel)]="user.username" formControlName="uname" class="form-control"

                            [ngClass]="{'is-invalid':submitted && f['uname'].errors}">

                        <div \*ngIf="submitted && f['uname'].errors" class="invalid-feedback">

                            <div \*ngIf="f['uname'].errors['required']" style="color: red;">

                                Username Required

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <label>Enter Password:</label>

                        <input type="password" placeholder="Password" [(ngModel)]="user.password" formControlName="password" class="form-control"

                            [ngClass]="{'is-invalid':submitted && f['password'].errors}">

                        <div \*ngIf="submitted && f['password'].errors" class="invalid-feedback">

                            <div \*ngIf="f['password'].errors['required']" style="color: red;">

                                Password Required

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <button class="btn btn-primary">Login</button>

                    </div>

                </form>

            </div>

        </div>

    </div>

</div>

<div \*ngIf="a==false">

    <h3>Invalid Credentials! Please try again</h3>

</div>

userdashboard.component.html

<br><br>

<h1> User Dashboard </h1>

<a routerLink="search">Search Product</a>|

<a routerLink="upwd">Change Password</a>|

<a routerLink="ulogout">Logout</a>

<router-outlet></router-outlet>

userlogout.component.html

<h1>Logged out successfully</h1>