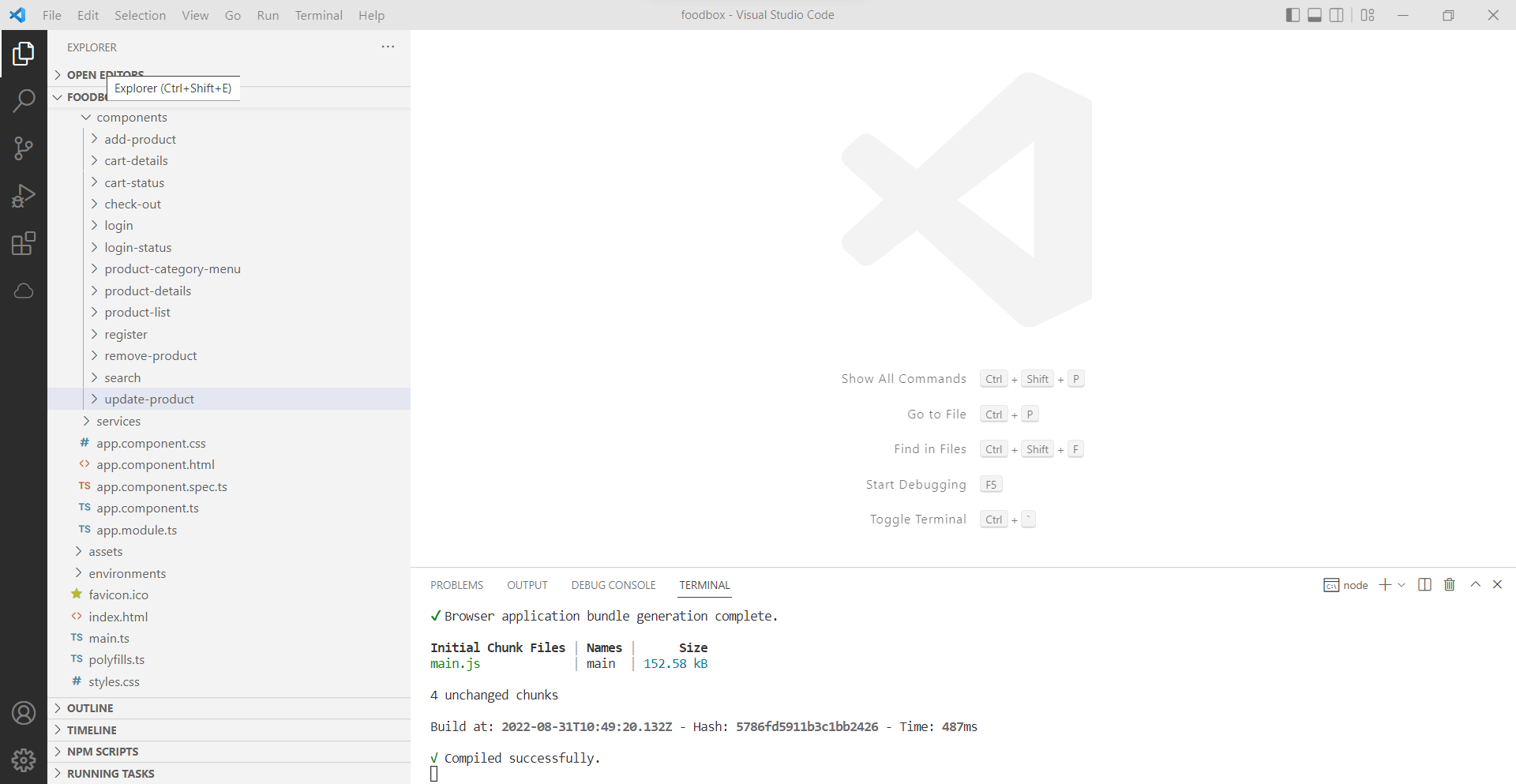
**FrontEnd(Angular – VS Code)**

**Project Structure:**

****

**App.component.html**

<div class="page-wrapper">

  <!-- MENU SIDEBAR-->

  <aside class="menu-sidebar d-none d-lg-block">

    <div class="logo">

      <a routerLink="/products">

        <img src="assets/foodboxlogo5.png" alt="food box" class="img-responsive">

      </a>

    </div>

    <app-product-category-menu></app-product-category-menu>

  </aside>

  <!-- END MENU SIDEBAR-->

  <div class="page-container">

    <!-- HEADER DESKTOP-->

    <header class="header-desktop">

      <div class="section-content section-content-p30">

        <div class="container-fluid">

          <div class="header-wrap">

            <app-search></app-search>

            <app-cart-status></app-cart-status>

            <app-login-status></app-login-status>

          </div>

        </div>

      </div>

    </header>

    <!-- END HEADER DESKTOP-->

    <!-- MAIN CONTENT-->

    <router-outlet></router-outlet>

  </div>

</div>

<!-- ENDS PAGE CONTAINER -->

<footer>

  <ul>

    <li><a href="#">About Us</a></li>

    <li><a href="#">Contact Us</a></li>

    <li><a href="#">Help</a></li>

  </ul>

</footer>

**App.module.ts**

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

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

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

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

import { ProductListComponent } from './components/product-list/product-list.component';

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

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

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

import { ProductCategoryMenuComponent } from './components/product-category-menu/product-category-menu.component';

import { SearchComponent } from './components/search/search.component';

import { ProductDetailsComponent } from './components/product-details/product-details.component';

import { NgbModule } from '@ng-bootstrap/ng-bootstrap';

import { CartStatusComponent } from './components/cart-status/cart-status.component';

import { CartDetailsComponent } from './components/cart-details/cart-details.component';

import { RegisterComponent } from './components/register/register.component';

import { LoginComponent } from './components/login/login.component';

import { LoginStatusComponent } from './components/login-status/login-status.component';

import { UpdateProductComponent } from './components/update-product/update-product.component';

import { RemoveProductComponent } from './components/remove-product/remove-product.component';

import { AddProductComponent } from './components/add-product/add-product.component';

import { CheckOutComponent } from './components/check-out/check-out.component';

const routes: Routes = [

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

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

  {path: 'checkout', component: CheckOutComponent},

  {path: 'add-product', component: AddProductComponent},

  {path: 'remove-product/:id', component: RemoveProductComponent},

  {path: 'update-product/:id', component: UpdateProductComponent},

  {path: 'cart-details', component: CartDetailsComponent},

  {path: 'products/:id', component: ProductDetailsComponent},

  {path: 'search/:keyword', component: ProductListComponent},

  {path: 'category/:id', component: ProductListComponent},

  {path: 'category', component: ProductListComponent},

  {path: 'products', component: ProductListComponent},

  {path: '', redirectTo: '/products', pathMatch: 'full'},

  {path: '\*\*', redirectTo: '/products', pathMatch: 'full'}

];

@NgModule({

  declarations: [

    AppComponent,

    ProductListComponent,

    ProductCategoryMenuComponent,

    SearchComponent,

    ProductDetailsComponent,

    CartStatusComponent,

    CartDetailsComponent,

    RegisterComponent,

    LoginComponent,

    LoginStatusComponent,

    UpdateProductComponent,

    RemoveProductComponent,

    AddProductComponent,

    CheckOutComponent

  ],

  imports: [

    RouterModule.forRoot(routes),

    BrowserModule,

    FormsModule,

    ReactiveFormsModule,

    HttpClientModule,

    NgbModule

  ],

  providers: [ProductService],

  bootstrap: [AppComponent]

})

export class AppModule { }

**Styles.css**

/\* You can add global styles to this file, and also import other style files \*/

/\*

 \* ----------------------------

 \* Common CSS

 \* ----------------------------

 \*/

 \*{

    margin:0;

    padding:0;

    -webkit-box-sizing:border-box;

    -moz-box-sizing:border-box;

    box-sizing:border-box

}

body{

    font-family:Poppins,sans-serif;

    font-weight:400;

    font-size:16px;

    line-height:1.625;

    color:#666;

    -webkit-font-smoothing:antialiased;

    -moz-osx-font-smoothing:grayscale

}

ul{

    margin:0

}

button,input[type=button]{

    cursor:pointer

}

button:focus,input:focus{

    outline:0

}

input{

    border:none

}

button{

    border:none;

    background:0 0

}

img{

    max-width:75%;

    height:auto

}

p{

    margin:0

}

h1,h2,h3,h4,h5,h6{

    color:rgb(0, 0, 0);

    font-weight:700;

    margin:0;

    line-height:1.2

}

h1{

    font-size:36px

}

h2{

    font-size:30px

}

h3{

    font-size:24px

}

h4{

    font-size:18px

}

h5{

    font-size:15px

}

h6{

    font-size:13px

}

a{

    display:inline-block

}

a:active,a:focus,a:hover{

    text-decoration:none;

    outline:0

}

a,a:hover{

    -webkit-transition:all .3s ease;

    -o-transition:all .3s ease;

    -moz-transition:all .3s ease;

    transition:all .3s ease

}

/\*Layout Css\*/

.section-content-p35{

    padding:0 35px

}

.page-wrapper{

    overflow:hidden;

    background:#ffffff;

    padding-bottom:8vh

}

.page-container{

    background:#fbff0020;

    padding-left:200px

}

.main-content{

    padding-top:116px;

    min-height:100vh

}

aside.menu-sidebar.d-none.d-lg-block.clr{

    background:0 0;

    z-index:99999999;

    height:75px;

}

/\*

 \* ----------------------------

 \* Header Desktop Section

 \* ----------------------------

 \*/

.header-desktop{

    background:#000000d5;

    -webkit-box-shadow:0 2px 5px 0 rgba(0,0,0,.1);

    -moz-box-shadow:0 2px 5px 0 rgba(0,0,0,.1);

    box-shadow:0 2px 5px 0 rgba(0,0,0,.1);

    position:fixed;

    top:0;

    right:0;

    left:200px;

    height:75px;

    z-index:3

}

.header-desktop .section-content{

    overflow:visible;

    position:absolute;

    top:50%;

    left:0;

    right:0;

    -webkit-transform:translateY(-50%);

    -moz-transform:translateY(-50%);

    -ms-transform:translateY(-50%);

    -o-transform:translateY(-50%);

    transform:translateY(-50%)

}

.header-wrap{

    display:-webkit-box;

    display:-webkit-flex;

    display:-moz-box;

    display:-ms-flexbox;

    display:flex;

    -webkit-box-align:center;

    -webkit-align-items:center;

    -moz-box-align:center;

    -ms-flex-align:center;

    align-items:center;

    -webkit-box-pack:justify;

    -webkit-justify-content:space-between;

    -moz-box-pack:justify;

    -ms-flex-pack:justify;

    justify-content:space-between

}

.header-button{

    display:-webkit-box;

    display:-webkit-flex;

    display:-moz-box;

    display:-ms-flexbox;

    display:flex;

    -webkit-box-align:center;

    -webkit-align-items:center;

    -moz-box-align:center;

    -ms-flex-align:center;

    align-items:center

}

.number{

    padding:6px 0

}

.arrows i{

    font-size:30px;

    padding:4px 4px

}

section{

    position:relative

}

.section-content{

    position:relative;

    margin:0 auto;

    z-index:1

}

.section-content-p30{

    padding:0 30px

}

.au-input-xl{

    min-width:935px

}

input.au-input.au-input-xl{

    border-radius:40px!important;

    margin-right:15px

}

.au-input{

    line-height:43px;

    border:1px solid #e5e5e5;

    font-size:14px;

    color:#666;

    padding:0 17px;

    -webkit-border-radius:3px;

    -moz-border-radius:3px;

    border-radius:3px;

    -webkit-transition:all .5s ease;

    -o-transition:all .5s ease;

    -moz-transition:all .5s ease;

    transition:all .5s ease

}

.cart-area.d-n{

    display:block

}

.cart-area i{

    font-size:22px;

    float:left;

    margin-top:16px;

    color:#fff;

    margin-left:15px

}

.total{

    background:#ffffff;

    line-height:43px;

    color:rgb(0, 0, 0);

    border-radius:30px;

    border:1px solid #0da8e4;

    float:left;

    margin-top:7px;

    padding-left:17px

}

.total span{

    display:inline-block;

    background:#fff;

    padding:0 13px;

    color:#000;

    border-top-right-radius:20px;

    border-bottom-right-radius:20px;

    margin-left:50px

}

.form-header{

    display:-webkit-box;

    display:-webkit-flex;

    display:-moz-box;

    display:-ms-flexbox;

    display:flex

}

.au-btn-submit{

    position:relative;

    right:0;

    min-width:95px;

    color:rgb(0, 0, 0);

    border:1px solid #205b8d;

    -webkit-border-radius:3px;

    -moz-border-radius:3px;

    border-radius:43px;

    background:#ffffff

}

.au-btn-submit:hover{

    background:#05143f

}

.au-btn-submit>i{

    font-size:20px;

    color:#fff;

    position:absolute;

    top:50%;

    left:50%;

    -webkit-transform:translate(-50%,-50%);

    -moz-transform:translate(-50%,-50%);

    -ms-transform:translate(-50%,-50%);

    -o-transform:translate(-50%,-50%);

    transform:translate(-50%,-50%)

}

/\*

 \* ----------------------------

 \* Left Menu Bar

 \* ----------------------------

 \*/

.menu-sidebar{

    width:200px;

    position:fixed;

    left:0;

    top:0;

    bottom:0;

    background:#fff;

    overflow-y:auto

}

.menu-sidebar .logo{

    background:#000000ab;

    height:75px;

    padding:0 35px;

    display:-webkit-box;

    display:-webkit-flex;

    display:-moz-box;

    display:-ms-flexbox;

    display:flex;

    -webkit-box-align:center;

    -webkit-align-items:center;

    -moz-box-align:center;

    -ms-flex-align:center;

    align-items:center;

    position:relative;

    z-index:3

}

.menu-sidebar .navbar-list .navbar-sub-list{

    display:none;

    padding-left:34px

}

.menu-sidebar .navbar-list .navbar-sub-list li a{

    padding:11.5px 0

}

.menu-sidebar-content{

    position:relative;

    height:-webkit-calc(100vh - 75px);

    height:-moz-calc(100vh - 75px);

    height:calc(100vh - 75px);

    background:#000000ab;

}

.navbar-sidebar{

    padding:35px;

    padding-bottom:0

}

.navbar-sidebar .navbar-list li a{

    color:rgb(255, 238, 0);

    font-size:22px;

    padding-left: 4%;

}

.navbar-sidebar .navbar-list li a i{

    margin-right:19px

}

.navbar-sidebar .navbar-list li a:hover{

    color:#42788d

}

/\*

 \* ----------------------------

 \* Header Mobile Section

 \* ----------------------------

 \*/

.header-mobile{

    position:absolute;

    top:0;

    left:0;

    right:0

}

.header-mobile .header-mobile-bar{

    padding:15px 0;    background: #fff;

}

.header-mobile .header-mobile-inner{

    display:-webkit-box;

    display:-webkit-flex;

    display:-moz-box;

    display:-ms-flexbox;

    display:flex;

    -webkit-box-align:center;

    -webkit-align-items:center;

    -moz-box-align:center;

    -ms-flex-align:center;

    align-items:center;

    -webkit-box-pack:justify;

    -webkit-justify-content:space-between;

    -moz-box-pack:justify;

    -ms-flex-pack:justify;

    justify-content:space-between

}

.header-mobile .hamburger{

    width:36px;

    height:36px;

    padding:0;

    line-height:1;

    vertical-align:top;

    background:#fff;

    display:-webkit-box;

    display:-webkit-flex;

    display:-moz-box;

    display:-ms-flexbox;

    display:flex;

    -webkit-box-pack:center;

    -webkit-justify-content:center;

    -moz-box-pack:center;

    -ms-flex-pack:center;

    justify-content:center;

    -webkit-box-align:center;

    -webkit-align-items:center;

    -moz-box-align:center;

    -ms-flex-align:center;

    align-items:center

}

.header-mobile .hamburger .hamburger-box{

    width:20px;

    height:15px

}

.header-mobile .hamburger .hamburger-box .hamburger-inner{

    width:20px;

    height:2px;

    -webkit-border-radius:0;

    -moz-border-radius:0;

    border-radius:0

}

.header-mobile .hamburger .hamburger-box .hamburger-inner:before{

    width:20px;

    height:2px;

    top:6px

}

.header-mobile .hamburger .hamburger-box .hamburger-inner:after{

    top:12px;

    width:20px;

    height:2px

}

.header-mobile .navbar-mobile{

    display:none;

    position:absolute;

    width:100%;

    top:88px;

    z-index:20

}

.header-mobile .navbar-mobile .navbar-mobile-list{

    background:#f8f8f8

}

.header-mobile .navbar-mobile .navbar-mobile-list>li>a{

    padding-left:15px!important

}

.header-mobile .navbar-mobile .navbar-mobile-list li a{

    color:rgb(234, 255, 0);

    display:block;

    padding:10px 15px;

    padding-right:25px;

    padding-left:0;

    border-bottom:1px solid #e6e6e6;

    text-transform:capitalize;

    line-height:inherit

}

.header-mobile .navbar-mobile .navbar-mobile-list li a:hover{

    color:#0da8e4

}

.header-mobile .navbar-mobile .navbar-mobile-list li a>i{

    margin-right:19px

}

.navbar-mobile-sub-list{

    display:none;

    padding-left:30px;

    background:#fff

}

.header-mobile .navbar-mobile .navbar-mobile-sub-list li a{

    padding-left:15px

}

.header-mobile-2{

    background:#393939;

    position:static

}

.header-mobile-2.header-mobile .navbar-mobile{

    top:82px

}

.header-mobile-2.header-mobile .hamburger{

    background:0 0

}

.header-mobile-2.header-mobile .hamburger .hamburger-box .hamburger-inner{

    background:#fff

}

.header-mobile-2.header-mobile .hamburger .hamburger-box .hamburger-inner::before{

    background:#fff

}

.header-mobile-2.header-mobile .hamburger .hamburger-box .hamburger-inner::after{

    background:#fff

}

.main-content.page-m{

    padding:10% 15%!important

}

/\*

 \* ----------------------------

 \* Products List

 \* ----------------------------

 \*/

.product-box img {

    width: 10px;

}

.items select {

    width: 10px;

    height: 40px;

    border-radius: 40px;

    padding-left: 15px;

    margin-bottom: 10px;

    outline: 0;

    margin-left: 5px;

}

.product-box{

    margin-bottom:60px

}

.price{

    margin:15px 0 0;

    font-weight:800;

    color:#1e1e1f

}

.product-box h1{

    font-size:18px;

    border-bottom:1px solid #ccc;

    padding-bottom:5px;

    font-weight:600;

    padding-top:12px;

    margin-bottom:7px

}

.product-box h2{

    font-size:15px;

    padding-bottom:0;

    font-weight:500;

    color:#585858

}

/\*

 \* ----------------------------

 \* Shopping Cart Checkout

 \* ----------------------------

 \*/

.form-area{

    background:#e5e5e5;

    position:relative;

    margin-bottom:45px;

    padding:50px 20px;

    border:3px solid #ccc

}

.form-area h3{

    position:absolute;

    top:-26px;

    left:13px;

    background:#e5e5e5;

    padding:10px 20px;

    font-weight:500;

    z-index:23;

    display:inline-block

}

.input-space{

    margin-bottom:20px

}

.input-space label{

    padding-left:31px;

    font-size:14px;

    margin-bottom:50px

}

.input-space input[type=text]{

    width:100%;

    height:40px;

    border-radius:30px;

    outline:0;

    padding:10px;

    border:1px solid #ccc

}

.input-space select{

    width:100%;

    height:40px;

    border-radius:30px;

    outline:0;

    padding:0 10px;

    border:1px solid #ccc

}

.au-checkbox{

    display:block;

    position:relative;

    cursor:pointer;

    font-size:22px;

    -webkit-user-select:none;

    -moz-user-select:none;

    -ms-user-select:none;

    user-select:none

}

.au-checkbox input{

    position:absolute;

    opacity:0;

    cursor:pointer

}

.au-checkmark{

    position:absolute;

    top:0;

    left:0;

    height:22px;

    width:22px;

    background-color:#fff!important;

    border:2px solid #e5e5e5;

    -webkit-border-radius:2px;

    -moz-border-radius:2px;

    border-radius:2px

}

.au-checkbox:hover input~.au-checkmark{

    background-color:transparent

}

.au-checkbox input:checked~.au-checkmark{

    background-color:#fff

}

.au-checkmark:after{

    content:"";

    position:absolute;

    display:none

}

.au-checkbox input:checked~.au-checkmark:after{

    display:block

}

label.date{

    padding-top:8px

}

.au-checkbox .au-checkmark:after{

    left:5px;

    top:-1px;

    width:9px;

    height:15px;

    border:solid #0da8e4;

    border-width:0 4px 4px 0;

    -webkit-border-radius:2px;

    -moz-border-radius:2px;

    border-radius:2px;

    -webkit-transform:rotate(45deg);

    -moz-transform:rotate(45deg);

    -ms-transform:rotate(45deg);

    -o-transform:rotate(45deg);

    transform:rotate(45deg)

}

option:first{

    color:#ccc

}

select{

    color:#828482

}

/\*

 \* ----------------------------

 \* Shopping Cart Details

 \* ----------------------------

 \*/

.table{

    margin:0

}

table.table.table-bordered{

    border:1px solid #ccc;

    background:#fff

}

.table-bordered td,.table-bordered th{

    border:1px solid #dee2e6

}

table.table.table-bordered th{

    BACKGROUND:#000000ab;

    color:#fff;

    font-size:14px;

    padding:7px

}

.table-responsive{

    padding-right:1px

}

.items select{

    width:100px;

    height:40px;

    border-radius:40px;

    padding-left:15px;

    margin-bottom:10px;

    outline:0;

    margin-left:5px

}

/\*

 \* ----------------------------

 \* Product Details

 \* ----------------------------

 \*/

 .price{

    margin:15px 0 0;

    font-weight:800;

    color:#000000

}

.detail-section{

    margin-top:60px

}

.detail-img{

    width: 30%;

}

/\*\*/

/\*

 \* ----------------------------

 \* Buttons

 \* ----------------------------

 \*/

 .btn-info{

    color:#fff;

    background-color:#070707!important;

    border-color:#000000!important

}

a.primary-btn{

    background:#eafa05;

    padding:7px 20px;

    text-align:center;

    display:inline-block;

    color:#fff;

    border-radius:2px;

    font-size:14px;

    margin-top:5px

}

a.primary-btn:hover{

    background:#060606;

    color:#fff

}

.active-link {

  font-weight: bold;

}

/\*

 \* ----------------------------

 \* Footer section

 \* ----------------------------

 \*/

footer ul li a{

    color:#fff

}

footer{

    background:#232324;

    z-index:1;

    position:bottom;

    text-align:center

}

footer li a{

    color:#313131;

    font-weight:500

}

footer ul li{

    display:inline-block;

    font-size:14px;

    padding:10px 20px

}

/\*

 \* ----------------------------

 \* Media Queries

 \* ----------------------------

 \*/

@media (max-width:1600px){

    .au-input-xl{

        min-width:290px

    }

}

@media (max-width:1315px) and (min-width:992px){

    .header-desktop3 .section-content{

        padding:0 15px

    }

}

@media (max-width:1570px) and (min-width:992px){

    .header-navbar ul li a{

        padding:25px 15px

    }

}

@media (max-width:1315px) and (min-width:992px){

    .header-navbar ul li a{

        font-size:13px;

        padding:27px 15px

    }

    .header-navbar ul li a i{

        margin-right:5px

    }

}

@media (min-width:1200px){

    .container{

        max-width:1320px

    }

}

@media (min-width:992px) and (max-width:1199px){

    .cart-area i{

        font-size:22px;

        float:left;

        margin-top:16px;

        color:#205b8d;

        margin-left:15px

    }

    .header-mobile .header-mobile-bar{

        padding:10px 0;

        background:#fff

    }

    .header-wrap .account-item>.content{

        display:none

    }

}

@media (max-width:991px){

     .section-content-p30{

        padding:0

    }

    .page-container{

        position:relative;

        top:88px;

        padding-left:0

    }

     .page-wrapper{

        background:#e5e5e5;

        padding-bottom:12vh

    }

     .main-content{

        padding-top:50px;

        padding-bottom:100px

    }

     .cart-area.d-n{

        display:none

    }

    .form-header{

        -webkit-box-pack:center;

        -webkit-justify-content:center;

        -moz-box-pack:center;

        -ms-flex-pack:center;

        justify-content:center

    }

     .au-input-xl{

        min-width:350px

    }

      .logo{

        text-align:center

    }

    .header-desktop{

        position:relative;

        top:0;

        left:0;

        height:80px

    }

    .header-wrap{

        -webkit-box-orient:vertical;

        -webkit-box-direction:normal;

        -webkit-flex-direction:column;

        -moz-box-orient:vertical;

        -moz-box-direction:normal;

        -ms-flex-direction:column;

        flex-direction:column

    }

    .header-button{

        margin-top:30px;

        width:100%;

        -webkit-box-pack:justify;

        -webkit-justify-content:space-between;

        -moz-box-pack:justify;

        -ms-flex-pack:justify;

        justify-content:space-between

    }

    .menu-sidebar2{

        top:0;

        right:-300px;

        left:auto;

        -webkit-box-shadow:0 5px 10px 0 rgba(0,0,0,.1);

        -moz-box-shadow:0 5px 10px 0 rgba(0,0,0,.1);

        box-shadow:0 5px 10px 0 rgba(0,0,0,.1)

    }

     .header-desktop2{

        left:0;

        position:relative

    }

     .header-desktop4{

        position:static

    }

     .section-content.section-content-p30.e-spc{

        padding-top:100px

    }

    header.header-desktop.extra{

        top:93px!important

    }

    .cart-area i{

        font-size:22px;

        float:left;

        margin-top:16px;

        color:#205b8d;

        margin-left:15px

    }

        .header-wrap2{

        -webkit-box-pack:justify;

        -webkit-justify-content:space-between;

        -moz-box-pack:justify;

        -ms-flex-pack:justify;

        justify-content:space-between

    }

}

@media (max-width:767px){

      .au-input-xl{

        min-width:150px

    }

    .header-button-item{

        font-size:22px;

        margin-right:15px

    }

        .main-content.page-m{

        padding:10% 0!important

    }

    .header-mobile .navbar-mobile{

        display:none;

        position:absolute;

        width:100%;

        top:66px;

        z-index:20

    }

    .section-content.section-content-p30.e-spc{

        padding-top:100px

    }

    header.header-desktop.extra{

        top:64px!important

    }

    .header-desktop{

        position:relative;

        top:0;

        left:0;

        height:80px

    }

    a.logo img{

        width:110px

    }

    .cart-area i{

        font-size:22px;

        float:left;

        margin-top:16px;

        color:#205b8d;

        margin-left:15px

    }

    .footer-pagination{

        text-align:center

    }

    .product-box{

        text-align:centerg

    }

    .page-container{

        position:relative;

        top:66px;

        padding-left:0

    }

    .au-btn-submit{

        position:relative;

        right:10px;

        min-width:85px;

        color:#fff;

        border:1px solid #205b8d;

        -webkit-border-radius:3px;

        -moz-border-radius:3px;

        border-radius:0;

        background:#205b8d

    }

    input.au-input.au-input-xl{

        border-radius:0!important;

        margin-right:0

    }

    .cart-area.d-b{

        display:block

    }

    .cart-area.d-n{

        display:none

    }

    .cart-area i{

        font-size:18px;

        float:left;

        margin-top:16px;

        color:#03a9f4;

        margin-left:15px

    }

    .total span{

        display:inline-block;

        background:#fff;

        padding:0 10px;

        color:#000;

        border-top-right-radius:20px;

        border-bottom-right-radius:20px;

        margin-left:6px

    }

    .total{

        background:#205b8d;

        line-height:30px;

        color:#fff;

        border-radius:30px;

        border:1px solid #205b8d;

        float:left;

        margin-top:9px;

        padding-left:12px

    }

    .header-mobile .header-mobile-bar{

        padding:6px 0;

        background:#fff;

        border:0;

        padding:8px 0

    }

}

.banner {

    display: flex;

    align-items: center;

  }

  .banner-start h3 {

    display: inline-block;

  }

  .banner-start i {

    padding: 0 6px;

  }

  .banner-middle {

    flex: 1;

  }

**Login.component.html**

<br>

<br>

<div class="login-form">

    <form  (ngSubmit)="onSubmit()" >

        <h2 class="text-center">Log in</h2>

        <div>

        <div class="form-group">

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

        </div>

        <br>

        <div class="form-group">

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

        </div>

        <br>

        <div class="form-group">

            <button type="submit" class="btn btn-dark btn-block">Log in</button>

        </div>

    </div>

    </form>

    <p class="text-center"><a routerLink="/register">Create an Account</a></p>

</div>

**productList.component.html**

<div class="main-content" >

    <div class="section-content section-content-p30">

        <div class="container-fluid">

            <div id="alert" style="margin-top: -23px;" \*ngIf="username != null" class="alert alert-warning col-xs-12 alert-dismissible" role="alert">

                Welecome {{username}}

            </div>

            <div class="row">

                <!-- loop over the collection of products -->

                <div \*ngFor="let tempProduct of products" class="col-md-4">

                    <div class="" style="width: 15rem; margin: 20px 28px;" >

                        <a routerLink="/products/{{ tempProduct.id }}">

                            <img style="width:340px;height:160px;" class="card-img-top"

                                src="{{ tempProduct.imageUrl }}">

                        </a>

                        <div >

                            <a style="margin-left: 0px; margin-top: 23px;"  routerLink="/products/{{ tempProduct.id }}">

                                <h6 style="font-size:20px;"> {{ tempProduct.name }}</h6>

                            </a>

                        </div>

                        <div class="card-body" style="margin-left: -15px;margin-top: -25px;"  >

                            <div class="price card-text">{{ tempProduct.price | currency:'INR' }}</div>

                            <button style="margin-top: 5px;"\*ngIf="username != 'admin'" (click)="addToCart(tempProduct)"

                                class="btn btn-dark btn-sm">Add to cart</button>

                            <button \*ngIf="username == 'admin'" routerLink="/update-product/{{ tempProduct.id }}"

                                class="btn btn-success btn-sm">Edit</button>

                                &nbsp;

                            <button \*ngIf="username == 'admin'" (click)="deleteProduct(tempProduct.id)"

                                class="btn btn-danger btn-sm">Remove</button>

                        </div>

                    </div>

                </div>

                <!-- if products empty then display a message-->

                <div \*ngIf="products?.length == 0" class="alert alert-danger col-md-12" role="alert">

                    No products found

                </div>

            </div>

            <!-- begin footer -->

            <div class="footer-pagination">

                <div class="row">

                    <div class="col-md-6"></div>

                    <div class="col-md-12">

                        <div class="row">

                            <div class="col-md-6" style="float: right;">

                                <ngb-pagination [(page)]="thePageNumber" [pageSize]="thePageSize"

                                    [collectionSize]="theTotalElements" [maxSize]="6" [boundaryLinks]="true"

                                    (pageChange)="listProducts()">

                                </ngb-pagination>

                            </div>

                            <div class="col-md-3 ;" style="float: right;">

                                <span class="mr-2">Sort by: </span>

                                <select (change)="updateSortingMethod($event.target.value)">

                                    <option>Name</option>

                                    <option>First Created</option>

                                    <option selected="true">Last Created</option>

                                </select>

                            </div>

                        <!--    <div class="col-md-3  " style="float: left;">

                                <span class="mr-2">Page Size</span>

                                <select (change)="updatePageSize($event.target.value)">

                                    <option>2</option>

                                    <option selected="true">5</option>

                                    <option>10</option>

                                    <option>20</option>

                                    <option>50</option>

                                </select>

                            </div>-->

                        </div>

                    </div>

                </div>

            </div>

            <!-- end footer -->

        </div>

    </div>

</div>

**Cartdetails.component.html**

<div class="main-content">

    <div class="section-content section-content-p30">

        <div class="container-fluid">

            <div \*ngIf="username == null" class="alert alert-danger col-md-12" role="alert">

                You need to be registered in the system.

            </div>

            <div \*ngIf="username != null && cartItems.length > 0">

                <table class="table table-bordered">

                    <tr>

                        <th width="20%">Product Image</th>

                        <th width="50%">Product Detail</th>

                        <th width="30%"></th>

                    </tr>

                    <tr \*ngFor="let tempCartItem of cartItems">

                        <td>

                            <img src="{{ tempCartItem.imageUrl }}" class="img-responsive" width="150px" />

                        </td>

                        <td>

                            <p>{{ tempCartItem.name }}</p>

                            <p>{{ tempCartItem.unitPrice | currency: 'INR' }}</p>

                        </td>

                        <td>

                            <div class="items">

                                <label>Quantity:</label>

                                <div class="row no-gutters">

                                    <div class="col">

                                        <button (click)="incrementQuantity(tempCartItem)" class="btn btn-success btn-sm">

                                            <i class="fas fa-plus"></i>

                                        </button>

                                    </div>

                                    <div class="col ml-2 mr-2">

                                        {{ tempCartItem.quantity }}

                                    </div>

                                    <div class="col">

                                        <button (click)="decrementQuantity(tempCartItem)" class="btn btn-danger btn-sm">

                                            <i class="fas fa-minus"></i>

                                        </button>

                                    </div>

                                    <div class="col-8"></div>

                                </div>

                            </div>

                            <button (click)="remove(tempCartItem)" class="au-btn-submit" style="width:22px; background: red; color: white" >Remove</button>

                            <p class="mt-2"></p>

                        </td>

                    </tr>

                    <tr>

                        <td colspan="2"></td>

                        <td style="font-weight: bold">

                            <p>Total Quantity: {{ totalQuantity }}</p>

                            <p>Total Price: {{ totalPrice | currency: 'USD' }}</p>

                            <a (click)="getTotalPrice()" routerLink="/checkout"><button class="au-btn-submit" style="width:81px; height: 35px; background: green; color: white"  >Confirm</button></a>

                            <br>

                        </td>

                    </tr>

                </table>

            </div>

            <!-- if cart is empty then display a message -->

            <div \*ngIf="cartItems.length == 0" class="alert alert-warning col-md-12" role="alert">

                Cart is empty - Please the add products.

            </div>

        </div>

    </div>

</div>

**Login.service.ts**

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

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

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

import { Observable } from 'rxjs';

import { map } from 'rxjs/operators';

@Injectable({

  providedIn: 'root'

})

export class LoginService {

  private baseUrl = 'http://localhost:8080/api/users';

  private baseUrlController = "http://localhost:8080/api/v1/users";

  constructor(private httpClient: HttpClient) { }

  getUser(username: string): Observable<User[]> {

    // search url

    const searchUrl = `${this.baseUrl}/search/findByUsername?username=${username}`;

    return this.httpClient.get<GetResponseUser>(searchUrl).pipe(

      map(response => response.\_embedded.users)

    );

  }

  createUser(user: User): Observable<Object>{

    return this.httpClient.post(`${this.baseUrlController}`, user);

  }

}

interface GetResponseUser {

  \_embedded: {

    users: User[];

  }}

**Product.service.ts**

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

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

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

import { Observable } from 'rxjs';

import { map } from 'rxjs/operators';

import { ProductCategory } from '../common/product-category';

@Injectable({

  providedIn: 'root'

})

export class ProductService {

  private baseUrl = 'http://localhost:8080/api/products';

  private categoryUrl = 'http://localhost:8080/api/cuisines';

  private newURL = "http://localhost:8080/api/v1/products";

  getListURL:string = null;

  constructor(private httpClient: HttpClient) { }

  updateProduct(id: number, product: Product): Observable<Object>{

    return this.httpClient.put(`${this.newURL}/${id}`, product);

  }

  addProduct(product: Product): Observable<Object>{

    return this.httpClient.post(`${this.newURL}`, product);

  }

  deleteProduct(id: number): Observable<Object>{

    return this.httpClient.delete(`${this.newURL}/${id}`);

  }

  getProduct(theProductId: number): Observable<Product> {

    // need to build URL based on product id

    const productUrl = `${this.baseUrl}/${theProductId}`;

    return this.httpClient.get<Product>(productUrl);

  }

  getProductCuisine(theProductId: number): Observable<ProductCategory> {

    // need to build URL based on product id

    const productUrl = `${this.baseUrl}/${theProductId}/cuisine`;

    return this.httpClient.get<ProductCategory>(productUrl);

  }

  getProductListPaginate(thePage: number,

                         thePageSize: number,

                         theCategoryId: number,

                         sortingMethod: string): Observable<GetResponseProducts> {

    if(sortingMethod == "First Created"){

        // need to build URL based on category id, page and size

      this.getListURL = `${this.baseUrl}/search/findByCuisine?cuisine=${theCategoryId}`

        + `&page=${thePage}&size=${thePageSize}`;

    }

    else if(sortingMethod == "Last Created"){

      this.getListURL = `${this.baseUrl}/search/findByCuisineOrderByDateCreatedDesc?cuisine=${theCategoryId}`

      + `&page=${thePage}&size=${thePageSize}`;

    }

    else{

      this.getListURL = `${this.baseUrl}/search/findByCuisineOrderByNameAsc?cuisine=${theCategoryId}`

      + `&page=${thePage}&size=${thePageSize}`;

    }

    return this.httpClient.get<GetResponseProducts>(this.getListURL);

  }

  getAllProductListPaginate(thePage: number,

    thePageSize: number,

    theCategoryId: number,

    sortingMethod: string): Observable<GetResponseProducts> {

      if(sortingMethod == "First Created"){

        // need to build URL based on category id, page and size

      this.getListURL = `${this.baseUrl}`;

    }

    else if(sortingMethod == "Last Created"){

      this.getListURL = `${this.baseUrl}/search/findAllByOrderByDateCreatedDesc?page=${thePage}&size=${thePageSize}`;

    }

    else{

      this.getListURL = `${this.baseUrl}/search/findAllByOrderByNameAsc?page=${thePage}&size=${thePageSize}`;

    }

    return this.httpClient.get<GetResponseProducts>(this.getListURL);

}

  getProductList(theCategoryId: number): Observable<Product[]> {

    // need to build URL based on category id

    const searchUrl = `${this.baseUrl}/search/findByCategoryId?id=${theCategoryId}`;

    return this.getProducts(searchUrl);

  }

  searchProducts(theKeyword: string): Observable<Product[]> {

    // need to build URL based on the keyword

    const searchUrl = `${this.baseUrl}/search/findByNameContaining?name=${theKeyword}`;

    return this.getProducts(searchUrl);

  }

  searchProductsPaginate(thePage: number,

                        thePageSize: number,

                        theKeyword: string): Observable<GetResponseProducts> {

    // need to build URL based on keyword, page and size

    const searchUrl = `${this.baseUrl}/search/findByNameContaining?name=${theKeyword}`

                    + `&page=${thePage}&size=${thePageSize}`;

    return this.httpClient.get<GetResponseProducts>(searchUrl);

  }

  private getProducts(searchUrl: string): Observable<Product[]> {

    return this.httpClient.get<GetResponseProducts>(searchUrl).pipe(map(response => response.\_embedded.products));

  }

  getProductCategories(): Observable<ProductCategory[]> {

    return this.httpClient.get<GetResponseProductCategory>(this.categoryUrl).pipe(

      map(response => response.\_embedded.cuisines)

    );

  }

}

interface GetResponseProducts {

  \_embedded: {

    products: Product[];

  },

  page: {

    size: number,

    totalElements: number,

    totalPages: number,

    number: number

  }

}

interface GetResponseProductCategory {

  \_embedded: {

    cuisines: ProductCategory[];

  }

}

**Cart.service.ts**

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

import { CartItem } from '../common/cart-item';

import { Subject } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class CartService {

  cartItems: CartItem[] = [];

  totalPrice: Subject<number> = new Subject<number>();

  totalQuantity: Subject<number> = new Subject<number>();

  constructor() { }

  addToCart(theCartItem: CartItem) {

    // check if we already have the item in our cart

    let alreadyExistsInCart: boolean = false;

    let existingCartItem: CartItem = undefined;

    if (this.cartItems.length > 0) {

      // find the item in the cart based on item id

      existingCartItem = this.cartItems.find( tempCartItem => tempCartItem.id === theCartItem.id );

      // check if we found it

      alreadyExistsInCart = (existingCartItem != undefined);

    }

    if (alreadyExistsInCart) {

      // increment the quantity

      existingCartItem.quantity++;

    }

    else {

      // just add the item to the array

      this.cartItems.push(theCartItem);

    }

    // compute cart total price and total quantity

    this.computeCartTotals();

  }

  computeCartTotals() {

    let totalPriceValue: number = 0;

    let totalQuantityValue: number = 0;

    for (let currentCartItem of this.cartItems) {

      totalPriceValue += currentCartItem.quantity \* currentCartItem.price;

      totalQuantityValue += currentCartItem.quantity;

    }

    // publish the new values ... all subscribers will receive the new data

    this.totalPrice.next(totalPriceValue);

    this.totalQuantity.next(totalQuantityValue);

    // log cart data just for debugging purposes

    this.logCartData(totalPriceValue, totalQuantityValue);

  }

  logCartData(totalPriceValue: number, totalQuantityValue: number) {

    console.log('Contents of the cart');

    for (let tempCartItem of this.cartItems) {

      const subTotalPrice = tempCartItem.quantity \* tempCartItem.price;

      console.log(`name: ${tempCartItem.name}, quantity=${tempCartItem.quantity}, price=${tempCartItem.price}, subTotalPrice=${subTotalPrice}`);

    }

    console.log(`totalPrice: ${totalPriceValue.toFixed(2)}, totalQuantity: ${totalQuantityValue}`);

    console.log('----');

  }

  decrementQuantity(theCartItem: CartItem) {

    theCartItem.quantity--;

    if (theCartItem.quantity === 0) {

      this.remove(theCartItem);

    }

    else {

      this.computeCartTotals();

    }

  }

  remove(theCartItem: CartItem) {

    // get index of item in the array

    const itemIndex = this.cartItems.findIndex( tempCartItem => tempCartItem.id === theCartItem.id );

    // if found, remove the item from the array at the given index

    if (itemIndex > -1) {

      this.cartItems.splice(itemIndex, 1);

      this.computeCartTotals();

    }

  }

}