

Angular Part

SRC-APP-about

about.component.html

```
<p>We at JOT provide you with most intuitive question answers to help sharpen  
your Java skills.</p>
```

addproduct

Add-product.component.css

```
body {  
    background: rgb(99, 39, 120)  
}  
  
.form-control:focus {  
    box-shadow: none;  
    border-color: #BA68C8  
}  
  
.profile-button {  
    background: rgb(99, 39, 120);  
    box-shadow: none;  
    border: none  
}  
  
.profile-button:hover {  
    background: #682773  
}  
  
.profile-button:focus {  
    background: #682773;  
    box-shadow: none  
}  
  
.profile-button:active {  
    background: #682773;  
    box-shadow: none  
}  
  
.back:hover {  
    color: #682773;  
    cursor: pointer  
}
```

```

.labels {
  font-size: 11px
}

.add-experience:hover {
  background: #BA68C8;
  color: #fff;
  cursor: pointer;
  border: solid 1px #BA68C8
}

```

add-product.component.html

```

<div class="container rounded bg-white mt-5 mb-5">
  <div class="row">
    <div class="col-md-3 border-right">
      <!-- <div class="d-flex flex-column align-items-center text-center
p-3 py-5"><img class="rounded-circle mt-5"
      src={{cuisine.image}}></div> -->
      <!--  -->

    </div>
    <div class="col-md-5 border-right">
      <div class="p-3 py-5">
        <div class="d-flex justify-content-between align-items-center
mb-3">

          <h4 class="text-right">Add New Product</h4>
        </div>
        <div class="row mt-2">
          <div class="col-md-6"><label class="labels">Product
Id</label>

          <input type="text" class="form-control" disabled
value="{{cuisine.product_id}}">
        </div>
          <div class="col-md-6"><label class="labels">Product
Name</label>

          <input type="text" class="form-control"
name="product_name" placeholder="Name"
[ src]="cuisine.product_name"
[(ngModel)]="cuisine.product_name">
        </div>
      </div>
      <div class="row mt-3">
        <div class="col-md-12"><label
class="labels">Description</label>

        <input type="text" class="form-control"
name="description" placeholder="Description"

```

```

                [src]="cuisine.description"
[(ngModel)]="cuisine.description">
            </div>
            <div class="col-md-12"><label class="labels">Price
(Rs.)</label>
                <input type="text" class="form-control" name="price"
placeholder="Price in Rs."
                [src]="cuisine.price" [(ngModel)]="cuisine.price">
            </div>
        </div>
        <div class="row mt-3">
            <div class="col-md-12"><label class="labels">Image</label>
                <input type="text" class="form-control" name="image"
placeholder="image path"
                [src]="cuisine.image" [(ngModel)]="cuisine.image"
value="">
            </div>
        </div>
        <div class="mt-5 text-center"><button class="btn btn-primary
profile-button" (click)="addDetails()" type="button">Save</button></div>
    </div>
</div>

```

add-product.component.ts

```

import { Component, OnInit } from '@angular/core';
import { Cuisine } from '../models/Cuisine';
import { CuisineService } from '../models/cuisine.service';

@Component({
  selector: 'app-add-product',
  templateUrl: './add-product.component.html',
  styleUrls: ['./add-product.component.css']
})
export class AddProductComponent implements OnInit {

  cuisine: Cuisine;

  constructor(private _cuisineService: CuisineService) { }

  ngOnInit(): void {
    this.cuisine = new Cuisine();
  }
}

```

```

public addDetails(){
    this._cuisineService.saveProductDetails(this.cuisine).subscribe();
}
}

```

Cart

Cart.component.html

```

<section class="jumbotron text-center">
  <div class="container">
    <h1 class="jumbotron-heading">Foodbox The Official Home Food
    </h1>
    <p class="lead text-muted">Your cart ..</p>
    <!-- <p>
      <a href="#" class="btn btn-primary my-2">Main call to
action</a>
      <a href="#" class="btn btn-secondary my-2">Secondary
action</a>
    </p> -->
  </div>
</section>

<div *ngIf="cuisineList.size>0;else other_content">
  <button class="btn btn-success btn-lg float-right"
(click)="checkout();">Checkout</button>
</div>
<ng-template #other_content>
  <p class="card-text"> <b> Your Cart is Empty</b></p>
</ng-template>
<div class="album py-5 bg-light">
  <div class="container">

    <div class="row">
      <div class="col-md-4" *ngFor="let cuisine of cuisineList">
        <div class="card mb-4 shadow-sm">
          
          <div class="card-body">
            <p class="card-text">{{ cuisine.description }}</p>
            <div class="d-flex justify-content-between align-
items-center">

              <div class="form-check-inline">
                <button>Add</button>

```

```

        <input class="quantity" min="0"
name="quantity" value="{{ cuisine.quantity }}"
        type="number">
        <!-- <div class="input-group">
            <span class="input-group-btn">
                <button type="button" class="btn btn-
default btn-number" disabled="disabled" data-type="minus" data-
field="quant[1]">
                    <span class="glyphicon glyphicon-
minus"></span>
                </button>
            </span>
            <input type="text" name="quant[1]"
class="form-control input-number" value="1" min="1" max="10">
            <span class="input-group-btn">
                <button type="button" class="btn btn-
default btn-number" data-type="plus" data-field="quant[1]">
                    <span class="glyphicon glyphicon-
plus"></span>
                </button>
            </span>
        </div> -->
        <!-- <div class="btn-group">
            <button type="button" class="btn btn-sm btn-
outline-secondary">View</button>
            <button type="button" class="btn btn-sm btn-
outline-secondary">Edit</button>
        </div> -->
        <small class="text-muted">9 mins</small>
    </div>
</div>
</div>
</div>

```

Cart.component.ts

```

import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import { CuisineService } from '../models/cuisine.service';
import { Router } from '@angular/router';

```

```

@Component({
  selector: 'app-cart',
  templateUrl: './cart.component.html',
  styleUrls: ['./cart.component.css']
})
export class CartComponent implements OnInit {

  cuisinelist: any;
  title: any;

  constructor(private router: Router, private _httpClient: HttpClient, private
    _cuisineService: CuisineService) {
    // this.title = this._cuisineService.title;
    // this.message = this._userService.getMessage();
  }

  ngOnInit(): void {
    //
    this._httpClient.get("http://localhost:3002/cuisines").subscribe((result) => {
      //   this.cuisineList = result;
      //   console.log(this.cuisineList);
      // }, (errors) => { console.log(errors); })

    this.cuisinelist = this._cuisineService.getSelectedItems();

  }

  public checkout() {
    this.router.navigateByUrl('/checkout');
  };
}

```

Checkout

Checkout.component.css

```

shopping-cart-img {
  width: 150px;
}

.number-input input[type="number"] {
  -webkit-appearance: textfield;
  -moz-appearance: textfield;
  appearance: textfield;
}

```

```
.number-input input[type=number]::-webkit-inner-spin-button,
.number-input input[type=number]::-webkit-outer-spin-button {
  -webkit-appearance: none;
}

.number-input {
  margin-bottom: 2rem;
}

.number-input button {
  -webkit-appearance: none;
  background-color: transparent;
  border: none;
  align-items: center;
  justify-content: center;
  cursor: pointer;
  margin: 0;
  position: relative;
}

.number-input button:before,
.number-input button:after {
  display: inline-block;
  position: absolute;
  content: '';
  height: 2px;
  transform: translate(-50%, -50%);
}

.number-input button.plus:after {
  transform: translate(-50%, -50%) rotate(90deg);
}

.number-input input[type=number] {
  text-align: center;
}

.number-input.number-input {
  border: 1px solid #ced4da;
  width: 10rem;
  border-radius: .25rem;
}

.number-input.number-input button {
  width: 2.6rem;
  height: .7rem;
}
```

```

.number-input.number-input button.minus {
  padding-left: 10px;
}

.number-input.number-input button:before,
.number-input.number-input button:after {
  width: .7rem;
  background-color: #495057;
}

.number-input.number-input input[type=number] {
  max-width: 4rem;
  padding: .5rem;
  border: 1px solid #ced4da;
  border-width: 0 1px;
  font-size: 1rem;
  height: 2rem;
  color: #495057;
}

@media not all and (min-resolution: .001dpcm) {
  @supports (-webkit-appearance: none) and (stroke-color:transparent) {

    .number-input.def-number-input.safari_only button:before,
    .number-input.def-number-input.safari_only button:after {
      margin-top: -.3rem;
    }
  }
}

.blue-border {
  border-top: 2px solid #4285f4;
}

.shopping-cart .def-number-input.number-input {
  border: none;
}

.shopping-cart .def-number-input.number-input input[type=number] {
  max-width: 2rem;
  border: none;
}

.shopping-cart .def-number-input.number-input input[type=number].black-text,
.shopping-cart .def-number-input.number-input input.btn.btn-link[type=number],
.shopping-cart .def-number-input.number-input input.md-toast-close-
button[type=number]:hover,

```



```

    .shopping-cart .def-number-input.number-input input.md-toast-close-
button[type=number]:focus {
    color: #212529 !important;
}

.shopping-cart .def-number-input.number-input button {
    width: 1rem;
}

.shopping-cart .def-number-input.number-input button:before, .shopping-cart
.def-number-input.number-input button:after {
    width: .5rem;
}

.shopping-cart .def-number-input.number-input button.minus:before,
.shopping-cart .def-number-input.number-input button.minus:after {
    background-color: #9e9e9e;
}

.shopping-cart .def-number-input.number-input button.plus:before, .shopping-
cart .def-number-input.number-input button.plus:after {
    background-color: #4285f4;
}

.shopping-cart .def-number-input.number-input.mt-n05 {
    margin-top: -.33rem;
}

.shopping-cart .back-link {
    position: absolute;
    bottom: 0;
}

```

Checkout.component.html

```

<!--Main layout-->
<div class="container wow fadeIn">

    <!-- Heading -->
    <h2 class="my-5 h2 text-center">Checkout form</h2>

    <!--Grid row-->
    <div class="row">

        <!--Grid column-->
        <div class="col-md-8 mb-4">

```

```

<!--Card-->
<div class="card">

  <!--Card content-->
  <form class="card-body">

    <!--Grid row-->
    <div class="row">

      <!--Grid column-->
      <div class="col-md-6 mb-2">

        <!--firstName-->
        <div class="md-form ">
          <input type="text" id="firstName" class="form-control">
          <label for="firstName" class="">First name</label>
        </div>

      </div>
      <!--Grid column-->

      <!--Grid column-->
      <div class="col-md-6 mb-2">

        <!--lastName-->
        <div class="md-form">
          <input type="text" id="lastName" class="form-control">
          <label for="lastName" class="">Last name</label>
        </div>

      </div>
      <!--Grid column-->

    </div>
    <!--Grid row-->

    <!--email-->
    <div class="md-form mb-5">
      <input type="text" id="email" class="form-control"
placeholder="youremail@example.com">
      <label for="email" class="">Email (optional)</label>
    </div>

    <!--address-->
    <div class="md-form mb-5">
      <input type="text" id="address" class="form-control"
placeholder="1234 Main St">

```

```

        <label for="address" class="">Address</label>
    </div>

    <!--address-2-->
    <div class="md-form mb-5">
        <input type="text" id="address-2" class="form-control"
placeholder="Apartment or suite">
        <label for="address-2" class="">Address 2 (optional)</label>
    </div>

    <!--Grid row-->
    <div class="row">

        <!--Grid column-->
        <div class="col-lg-4 col-md-12 mb-4">

            <label for="country">Country</label>
            <select class="custom-select d-block w-100" id="country"
required>

                <option value="">Choose...</option>
                <option>India</option>
            </select>
            <div class="invalid-feedback">
                Please select a valid country.
            </div>

        </div>
        <!--Grid column-->

        <!--Grid column-->
        <div class="col-lg-4 col-md-6 mb-4">

            <label for="state">State</label>
            <select class="custom-select d-block w-100" id="state"
required>

                <option value="">Choose...</option>
                <option>Maharashtra</option>
                <option>Pune</option>
                <option>Hyderabad</option>
                <option>Mumbai</option>
            </select>
            <div class="invalid-feedback">
                Please provide a valid state.
            </div>

        </div>
        <!--Grid column-->

```

```

        <!--Grid column-->
        <div class="col-lg-4 col-md-6 mb-4">

            <label for="zip">Zip</label>
            <input type="text" class="form-control" id="zip"
placeholder="" required>
            <div class="invalid-feedback">
                Zip code required.
            </div>

        </div>
        <!--Grid column-->

    </div>
    <!--Grid row-->

    <hr>

    <div class="custom-control custom-checkbox">
        <h4>Payment Details</h4>
    </div>

    <div class="d-block my-3">
        <div class="custom-control custom-radio">
            <input id="credit" name="paymentMethod" type="radio"
class="custom-control-input" checked required>
            <label class="custom-control-label" for="credit">Credit
card</label>
        </div>
        <div class="custom-control custom-radio">
            <input id="debit" name="paymentMethod" type="radio"
class="custom-control-input" required>
            <label class="custom-control-label" for="debit">Debit
card</label>
        </div>
        <div class="custom-control custom-radio">
            <input id="cod" name="paymentMethod" type="radio"
class="custom-control-input" required>
            <label class="custom-control-label" for="cod">Cash on
Delivery</label>
        </div>
    </div>
    <div class="row">
        <div class="col-md-6 mb-3">
            <label for="cc-name">Name on card</label>
            <input type="text" class="form-control" id="cc-name"
placeholder="" required>

```

```

        <small class="text-muted">Full name as displayed on
card</small>

        <div class="invalid-feedback">
            Name on card is required
        </div>
    </div>
    <div class="col-md-6 mb-3">
        <label for="cc-number">Card number</label>
        <input type="text" class="form-control" id="cc-number"
placeholder="" required>
        <div class="invalid-feedback">
            Card number is required
        </div>
    </div>
    <div class="row">
        <div class="col-md-3 mb-3">
            <label for="cc-expiration">Expiration</label>
            <input type="text" class="form-control" id="cc-expiration"
placeholder="" required>
            <div class="invalid-feedback">
                Expiration date required
            </div>
        </div>
        <div class="col-md-3 mb-3">
            <label for="cc-expiration">CVV</label>
            <input type="text" class="form-control" id="cc-cvv"
placeholder="" required>
            <div class="invalid-feedback">
                Security code required
            </div>
        </div>
    </div>
</form>
</div>
<!--/.Card-->

</div>
<!--Grid column-->

<!--Grid column-->
<div class="col-md-4 mb-4">

    <!-- Heading -->
    <h4 class="d-flex justify-content-between align-items-center mb-3">
        <span class="text-muted">Your cart</span>
        <span class="badge badge-secondary badge-pill">3</span>
    </h4>

```

```

        <!-- Cart -->
        <ul class="list-group mb-3 z-depth-1">
            <li *ngFor="let cuisine of cartItemList" class="list-group-item d-
flex justify-content-between lh-condensed">
                <div>
                    <h6 class="my-0">{{cuisine.product_name}}</h6>
                    <small class="text-muted">{{cuisine.quantity}}</small> -->
                </div>
                <span class="text-muted">INR
{{cuisine.price*cuisine.quantity}}</span>
            </li>

            <li class="list-group-item d-flex justify-content-between">
                <span>Total (INR)</span>
                <strong>{{totalPrice}}</strong>
            </li>
        </ul>
        <!-- Cart -->

        <!-- Promo code -->
        <form class="card p-2">
            <button class="btn btn-primary btn-lg btn-block"
(click)="confirmCheckout();">Continue to checkout</button>
        </form>
        <!-- Promo code -->

    </div>
    <!--Grid column-->

</div>
<!--Grid row-->

</div>

```

Checkout.component.ts

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { Cuisine } from '../models/Cuisine';
import { CuisineService } from '../models/cuisine.service';

@Component({
  selector: 'app-checkout',
  templateUrl: './checkout.component.html',

```

```

    styleUrls: ['./checkout.component.css']
  })
  export class CheckoutComponent implements OnInit {

    cartItemList : Cuisine[];
    totalPrice : number;

    constructor(private router: Router, private _cuisineService: CuisineService)
    { }

    ngOnInit(): void {
      this.cartItemList = this._cuisineService.getCart();
      this.totalPrice = 0;
      for(let cuisine of this.cartItemList){
        this.totalPrice = this.totalPrice + (cuisine.price * cuisine.quantity);
      }
    }

    public confirmCheckout(){
      this.router.navigateByUrl('/success');
    }
  }
}

```

Footer

Footer.component.html

```

<p>© Copyright 2021. All Rights Reserved.</p>

```

Home

Home.component.html

```

<section class="jumbotron text-center">
  <div class="container">
    <div class="card border-0 shadow my-5">
      <div class="card-body p-5">
        <h1 class="jumbotron-heading">FoodBox The Official Home Food
        </h1>
        <blockquote class="blockquote">
          <p class="lead text-muted">"There is one inexhaustible aspect of
          everyday life that never dulls, namely the simple pleasure of eating and
          drinking" </p>
          <p>- Andre Domine</p>

```

```

    </blockquote>
    <!-- <p>
      <a href="#" class="btn btn-primary my-2">Main call to action</a>
      <a href="#" class="btn btn-secondary my-2">Secondary action</a>
    </p> -->
  </div>
</div>
</div>
</section>
<script type="text/javascript" src="/theme/js/jquery.min.js"></script>
<script type="text/javascript" src="/theme/js/jquery.timer.min.js"></script>

```

Login

Login.component.css

```

:root {
  --jumbotron-padding-y: 3rem;
}

.jumbotron {
  padding-top: var(--jumbotron-padding-y);
  padding-bottom: var(--jumbotron-padding-y);
  margin-bottom: 0;
  background-color: #fff;
}

@media (min-width: 768px) {
  .jumbotron {
    padding-top: calc(var(--jumbotron-padding-y) * 2);
    padding-bottom: calc(var(--jumbotron-padding-y) * 2);
  }
}

.jumbotron p:last-child {
  margin-bottom: 0;
}

.jumbotron-heading {
  font-weight: 300;
}

.jumbotron .container {
  max-width: 40rem;
}

footer {
  padding-top: 3rem;
}

```



```
padding-bottom: 3rem;
}

footer p {
margin-bottom: .25rem;
}
```

Login.component.html

```
<div class="card border-0 shadow my-5">
  <div class="card-body p-5">
<form [formGroup]="form" #LoginInformation
(ngSubmit)="Login(LoginInformation)">

  <div class="row">
    <div class="col-md-offset-1 col-md-4">
      <h2>Login</h2>
    </div>
  </div>
</div>
<div class="row">
  <div class="col-md-offset-1 col-md-4">
    <h2></h2>
  </div>
</div>

  <div class="row">
    <div class="col-md-offset-1 col-md-4">
      <label for="username"> Username </label>
      <input formControlName="email" class="form-control" type="text">
    </div>
  </div>

  <div class="row">
    <div class="col-md-offset-1 col-md-4">
      <label for="password"> Password </label>
      <input formControlName="password" class="form-control"
type="password">
    </div>
  </div>

  <div class="row" style="margin-top: 40px;">
    <div class="col-md-offset-1 col-md-4">
      <button class="btn btn-md btn-primary btn-style" >Login</button>
    </div>
  </div>
```

```
</form>
</div>
</div>
```

Login.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormGroup, Validators, FormControl } from '@angular/forms';
import { AdminDetail } from '../models/admin-detail';
import { AdminService } from '../models/admin.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {

  private adminDetail = new AdminDetail();

  constructor(private _adminService : AdminService, private router : Router) {

  }

  ngOnInit() {
    if((this._adminService.isLoggedIn()) )
    {
      this.router.navigate(['/home']);
    }
    else
    {
      this.router.navigate(['/login']);
    }
  }

  // create the form object.
  form = new FormGroup({
    email : new FormControl('', Validators.required),
    password : new FormControl('', Validators.required)
  });

  Login(LoginInformation)
  {
    this.adminDetail.username = this.Email.value;
    this.adminDetail.password = this.Password.value;

    this._adminService.login(this.adminDetail).subscribe(
```

```

        response => {
            let result = response.json();

            if(result != null)
            {
                let token = result.token;
                //response.headers.get("Authorization");

                localStorage.setItem("token" , token);

                this._adminService.setAdminDetails(result.user);
                localStorage.setItem("role" ,
this._adminService.adminDetails.role);
                localStorage.setItem("username" ,
this._adminService.adminDetails.name);

                // this.router.navigate(['/home', result]);
                window.location.replace('/home');
            }
            else
            {
                alert("please register before login Or Invalid combination of
Email and password");
            }

        },
        error => {
            console.log("Error in authentication");
        }
    );
}

get Email(){
    return this.form.get('email');
}

get Password(){
    return this.form.get('password');
}
}

```

Logout

Logout.component.html

```
<p>logout works!</p>
```

Logout.component.ts

```
import { Component, OnInit } from '@angular/core';
import { AdminService } from '../models/admin.service';

@Component({
  selector: 'app-logout',
  templateUrl: './logout.component.html',
  styleUrls: ['./logout.component.css']
})
export class LogoutComponent implements OnInit {

  constructor(private _adminService :AdminService) { }

  ngOnInit(): void {
    this._adminService.logout();
    window.location.replace('/home');
  }

}
```

Main

Main.component.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="icon" href="/docs/4.1/assets/img/favicons/favicon.ico">

    <title>FoodBox Online Delivery</title>

    <link rel="canonical"
href="https://getbootstrap.com/docs/4.1/examples/album/">

    <!-- Bootstrap core CSS -->
    <!-- <link href="../../dist/css/bootstrap.min.css" rel="stylesheet"> -->
```

```

    <!-- Custom styles for this template -->
    <!-- <link href="album.css" rel="stylesheet"> -->
</head>

<body>

    <header>

        <div class="navbar navbar-dark bg-dark shadow-sm">
            <div class="container d-flex justify-content-between">
                <a href="#" class="navbar-brand d-flex align-items-center">
                    <!-- <svg xmlns="assets/img/foodbox_logo.png" width="20"
height="20" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-
width="2" stroke-linecap="round" stroke-linejoin="round" class="mr-2"><path
d="M23 19a2 2 0 0 1-2 2H3a2 2 0 0 1-2-2V8a2 2 0 0 1 2-2h4l2-3h6l2 3h4a2 2 0 0
1 2 2z"></path><circle cx="12" cy="13" r="4"></circle></svg> -->
                    <strong>FoodBox</strong>
                </a>
                <div *ngIf="isLoggedIn==true; else elseBlock">
                    <ul class="nav nav-pills">
                        <!-- <li><small class="nav-link"> Welcome {{ username
}}</small></li> -->
                        <!-- <li><a href="#" class="dropdown-toggle" data-
toggle="dropdown"> Welcome {{ username }} <b class="caret"></b></a></li> -->
                        <li class="nav-item"><a routerLink="home" class="nav-link
text-muted" aria-disabled=""> Welcome {{ username }}</a></li>
                        <li class="nav-item"><a routerLink="menu" class="nav-
link">Menu</a></li>
                        <li *ngIf="isAdmin==true" class="nav-item"><a
routerLink="product" class="nav-link">Manage Products</a></li>
                        <li class="nav-item"><a routerLink="cart" class="nav-
link">Goto Cart</a></li>
                        <li class="nav-item"><a routerLink="logout" class="nav-
link">Logout</a></li>
                    </ul>

                </div>
                <ng-template #elseBlock>
                    <ul class="nav nav-pills">
                        <li class="nav-item"><a routerLink="login" class="nav-
link">Login</a></li>
                        <li class="nav-item"><a routerLink="signup" class="nav-
link">Sign Up</a></li>
                    </ul>
                </ng-template>
            </div>
        </div>
    </header>

```

```

        <!-- <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarHeader" aria-controls="navbarHeader"
aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
        </button> -->
    </div>
</div>
</header>

<main class="container" style="text-align: center; align-self: center;">
    <div class="bg-light p-5 rounded">
        <router-outlet></router-outlet>
    </div>
</main>

<footer class="text-muted">
    <div class="container">

        <p>All rights reserved. Copyright © FoodBox Ltd. Disclaimer | T&C |
Privacy Policy</p>
    </div>
</footer>

<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script>window.jQuery || document.write('<script
src="../../assets/js/vendor/jquery-slim.min.js"></script>')</script>
    <script src="../../assets/js/vendor/popper.min.js"></script>
    <script src="../../dist/js/bootstrap.min.js"></script>
    <script src="../../assets/js/vendor/holder.min.js"></script>

</body>
</html>

```

Main.component.ts

```

import { Component, OnInit } from '@angular/core';
import { AdminService } from '../models/admin.service';
import { ActivatedRoute, Router } from '@angular/router';
import { FormGroup, Validators, FormControl } from '@angular/forms';

```

```

@Component({
  selector: 'app-main',
  templateUrl: './main.component.html',
  styleUrls: ['./main.component.css']
})
export class MainComponent implements OnInit {

  username : String = "";
  isAdmin : boolean = false;
  isLoggedIn: boolean = false;

  constructor(private _adminService : AdminService, private route :
ActivatedRoute, private router : Router) { }

  ngOnInit() {

    if((this._adminService.isLoggedIn()) ){
      this.isLoggedIn = true;
      this.username = localStorage.getItem("username");
      let role = localStorage.getItem("role");
      if(role == 'admin'){
        this.isAdmin = true;
      }else{
        this.isAdmin = false;
      }
    }else{
      this.isLoggedIn = false;
      this.isAdmin = false;
    }

    // if((this.adminService.isLoggedIn()) )
    // {
    //     this.router.navigate(['/home']);
    // }
    // else
    // {
    //     this.router.navigate(['/login']);
    // }

  }
}

```

Menu

Menu.component.css

```
:root {
  --jumbotron-padding-y: 3rem;
}

.jumbotron {
  padding-top: var(--jumbotron-padding-y);
  padding-bottom: var(--jumbotron-padding-y);
  margin-bottom: 0;
  background-color: #fff;
}

@media (min-width: 768px) {
  .jumbotron {
    padding-top: calc(var(--jumbotron-padding-y) * 2);
    padding-bottom: calc(var(--jumbotron-padding-y) * 2);
  }
}

.jumbotron p:last-child {
  margin-bottom: 0;
}

.jumbotron-heading {
  font-weight: 300;
}

.jumbotron .container {
  max-width: 40rem;
}

footer {
  padding-top: 3rem;
  padding-bottom: 3rem;
}

footer p {
  margin-bottom: .25rem;
}
```

Menu.component.html

```
<section class="jumbotron text-center">
  <div class="container">
    <h1 class="jumbotron-heading">Foodbox The Official Home Food
```



```

        </h1>
        <p class="lead text-muted">Choose from our exotic cuisines..</p>
        <!-- <p>
            <a href="#" class="btn btn-primary my-2">Main call to action</a>
            <a href="#" class="btn btn-secondary my-2">Secondary action</a>
        </p> -->
    </div>
</section>

<div class="album py-5 bg-light">
    <div class="container">

        <div class="row">
            <div class="col-md-4" *ngFor="let cuisine of cuisineList">
                <div class="card mb-4 shadow-sm" >
                    
                    <div class="card-body">
                        <p class="card-text"><b>{{ cuisine.product_name
}}</b></p>
                        <p class="card-text">{{ cuisine.description }}</p>
                        <div class="d-flex justify-content-between align-
items-center">
                            <div *ngIf="cuisine.enabled==true;else other_content">
                                <div class="form-check-inline">
                                    Add&nbsp;: &nbsp;
                                    <input class="quantity" style="text-
align:center;width:50px;" min="0" name="quantity"
                                        [value]="cuisine.quantity"
                                        [(ngModel)]="cuisine.quantity"
                                        type="number">
                                    <!-- <div class="input-group">
                                        <span class="input-group-btn">
                                            <button type="button" class="btn btn-default btn-
number" disabled="disabled" data-type="minus" data-field="quant[1]">
                                                <span class="glyphicon glyphicon-
minus"></span>
                                            </button>
                                        </span>
                                        <input type="text" name="quant[1]" class="form-
control input-number" value="1" min="1" max="10">
                                        <span class="input-group-btn">
                                            <button type="button" class="btn btn-default btn-
number" data-type="plus" data-field="quant[1]">
                                                <span class="glyphicon glyphicon-plus"></span>
                                            </button>
                                        </span>
                                    </div> -->
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

                <!-- <div class="btn-group">
                <button type="button" class="btn btn-sm btn-outline-
secondary">View</button>
                <button type="button" class="btn btn-sm btn-outline-
secondary">Edit</button>
                </div> -->
                <small class="text-muted" style="align-self: right;
"> INR {{ cuisine.price }}</small>
                </div></div>
                <ng-template #other_content>
                <p class="card-text">Not Available</p>
                </ng-template>

            </div>
        </div>
    </div>
</div>

```

Menu.component.ts

```

import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import { CuisineService } from '../models/cuisine.service';

@Component({
  selector: 'app-menu',
  templateUrl: './menu.component.html',
  styleUrls: ['./menu.component.css']
})

export class MenuComponent implements OnInit {

  cuisinelist: any;
  title: any;

  constructor(private _httpClient: HttpClient, private _cuisineService:
CuisineService) {
    // this.title = this._cuisineService.title;
    // this.message = this._userService.getMessage();
  }

  ngOnInit(): void {

```

```

    //
    this._httpClient.get("http://localhost:3002/cuisines").subscribe((result) => {
        // this.cuisineList = result;
        // console.log(this.cuisineList);
        // }, (errors) => { console.log(errors); })

    this._cuisineService.getAllCuisines().subscribe((result) => {
        this._cuisineService.cuisineList = result.json();
        this.cuisineList = this._cuisineService.cuisineList;
        console.log(this.cuisineList);
    }, (errors => console.log('There are some errors : ' + errors)))

    // this.cuisineList = this._cuisineService.getAllCuisines();
}

// if((this.adminService.isLoggedIn()) )
// {
//     this.route.paramMap.subscribe(params => {
//         this.adminId += params.get('adminId');
//     });
// }
// else
// {
//     this.router.navigate(['/login']);
// }

// }

// getAdminData()
// {
//     this.haveData = 0;

//     this.dataRequest = true;

//     this.adminService.getAdminDetail(this.adminId).subscribe(
//         response => {

//             let result = response.json();
//             this.data = result;

//             if(result == " ")
//             {
//                 this.haveData = 0;
//             }
//             else
//             {
//                 this.haveData = this.haveData + 1;

```

```

//          }
//      },
//      error => {
//          console.log("error while getting Admin Data");
//      }
//  );
// }
}

```

Models

Admin.details.ts

```

export class AdminDetail {
    adminId: string;
    username : string;
    name : string;
    password : string ;
    role : string;
}

```

Admin.service.ts

```

import { Injectable } from '@angular/core';
import { Http, RequestOptions , Headers } from '@angular/http';
import { Observable } from 'rxjs';
import { AdminDetail } from '../models/admin-detail';
import { Router } from '@angular/router';

import { JwtHelperService } from '@auth0/angular-jwt';

@Injectable({
    providedIn: 'root'
})
export class AdminService {

    // Base URL
    private baseUrl = "http://localhost:9090/";
    adminDetails: any;
    isAdmin: boolean = false;
}

```

```

constructor(private http: Http, private router : Router) { }

saveAdminDetails(adminDetail : AdminDetail) : Observable<any>
{
    let url = this.baseUrl + "register";
    return this.http.post(url,adminDetail);
}

login(adminDetail : AdminDetail) : Observable<any>
{
    let url = this.baseUrl + "authenticate";
    return this.http.post(url, adminDetail);
}

logout()
{
    // Remove the token from the localStorage.
    localStorage.removeItem('token');

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

/*
 * Check whether User is loggedIn or not.
 */

isLoggedIn() {

    // create an instance of JwtHelper class.
    let jwtHelper = new JwtHelperService();

    // get the token from the localStorage as we have to work on this token.
    let token = localStorage.getItem('token');

    // check whether if token have something or it is null.
    if(!token)
    {
        return false;
    }

    // get the Expiration date of the token by calling
    getTokenExpirationDate(String) method of JwtHelper class. this method accept a
    string value which is nothing but token.

```

```

    if(token)
    {
        let expirationDate = jwtHelper.getTokenExpirationDate(token);

        // check whether the token is expired or not by calling isTokenExpired()
        method of JwtHelper class.

        let isExpired = jwtHelper.isTokenExpired(token);

        return !isExpired;
    }
}

getAdminDetail(username) : Observable<any>
{
    let url = this.baseUrl + "getUserByName/" + username;

    // create an instance of Header object.
    let headers = new Headers();

    // get token from localStorage.
    let token = localStorage.getItem('token');

    // Append Authorization header.
    headers.append('Authorization' , 'Bearer ' + token);

    // create object of RequestOptions and include that in it.
    let options = new RequestOptions( { headers : headers } );

    return this.http.get(url , options);
}

setAdminDetails(adminDetails) {
    this.adminDetails = adminDetails;
    if(adminDetails.role == 'admin'){
        this.isAdmin = true;
    }else{
        this.isAdmin = false;
    }
}
}
}

```

Cuisine.service.ts

```

import { HttpClient, HttpHeaders } from '@angular/common/http';
import { Injectable } from "@angular/core";

```

```

import { Observable } from 'rxjs';
import { Cuisine } from '../cuisine';
import { Http, RequestOptions, Headers } from '@angular/http';
import { Router } from '@angular/router';

@Injectable()
export class CuisineService {

    baseUrl : String = '';
    cuisineList : Cuisine[];
    cart : Cuisine[] = [];

    // constructor(private _httpClient: HttpClient) {
    // //
    this._httpClient.get<Cuisine[]>("http://localhost:3002/cuisines").subscribe((result) => {
        // // this.cuisineList = result;
        // // console.log(this.cuisineList);
        // // }, (errors) => { console.log(errors); })
        // // this._httpClient.get("http://localhost:3002/cuisines");
        // }

    constructor(private _httpClient: HttpClient, private http: Http, private router : Router) { }

    title: string = "Cuisine Management System";

    getMessage(): string {
        return "Cuisine List";
    }

    getAllCuisines(): Observable<any> {

        // return
        this._httpClient.get<Cuisine[]>("http://localhost:3002/cuisines");
        let url = "http://localhost:9090/product";//this.baseUrl +
        "getAdminData/" + adminId;

        // create an instance of Header object.
        let headers = new Headers();

        // get token from localStorage.
        let token = localStorage.getItem('token');

        // let token =
        "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJqYXZhaW51c2UiLCJleHAiOjE2MjE5NzQ1NjEsIm1hdCI6MTYyMTk1NjU2MX0.FXRVeXEjJF7ReeGYcKlQvP_U_8hCwyvIL_VryIQdWGRHVJi1-o3_cPSILqMV-f2xQ2ZiFr2UsPJqUrSQ075e8g";//localStorage.getItem('token');

```

```

        // Append Authorization header.
        headers.append('Authorization' , 'Bearer ' + token);

        // create object of RequestOptions[ and include that in it.
        let options = new RequestOptions( { headers : headers } );

        return this.http.get(url , options);

    }

    // getAllCuisines(): Observable<any> {

        //      // return
        this._httpClient.get<Cuisine[]>("http://localhost:3002/cuisines");
        //      let url = "http://localhost:9090/product";//this.baseUrl +
        "getAdminData/" + adminId;

        //      // create an instance of Header object.
        // //      let headers = new Headers();

        //      // get token from localStorage.
        //      let token = localStorage.getItem('token');

        //      // Append Authorization header.
        // //      headers.append('Authorization' , 'Bearer ' + token);
        // let headers = new HttpHeaders();
        // //      headers.append('Authorization' , 'Bearer ' + "
        eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJqYXZhaW51c2UiLCJleHAiOjE2MjE3MjU5Im1hdCI6MTYyMTcwNTk1NX0.L6tE-
        7B7xzSmwWEX05knWpJDbyBQ9hdC7quAHhZy2_NT0qCviFJqc9LhSwfNyrdrfm7WtJy7zzitt9q0sJb
        -Og");
        // headers.set('Authorization' , 'Bearer ' + "
        eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJqYXZhaW51c2UiLCJleHAiOjE2MjE3MjU5Im1hdCI6MTYyMTcwNTk1NX0.L6tE-
        7B7xzSmwWEX05knWpJDbyBQ9hdC7quAHhZy2_NT0qCviFJqc9LhSwfNyrdrfm7WtJy7zzitt9q0sJb
        -Og");

        //      // create object of RequestOptions and include that in it.
        // //      let options = new RequestOptions( { headers : headers } );

        // //      return this.http.get(url , options);
        // //return
        this._httpClient.get<Cuisine[]>("http://localhost:9090/products");

        //      return
        this._httpClient.get<Cuisine[]>("http://localhost:9090/product",

```



```

        // {headers : headers });

        // // new HttpHeaders().set('Authorization', 'my-auth-token'),

        // }

        getCuisineById(id: number) : Observable<Cuisine> {
            return this._httpClient.get<Cuisine>("http://localhost:9090/product/"
+ id);
        }

        getCuisineList(){
            return this.cuisineList;
        }

        saveProductDetails(cuisine : Cuisine) : Observable<any>
        {

            // return
            this._httpClient.get<Cuisine[]>("http://localhost:3002/cuisines");
            let url = "http://localhost:9090/product";//this.baseUrl +
            "getAdminData/" + adminId;

            // create an instance of Header object.
            let headers = new Headers();

            // get token from localStorage.
            let token = localStorage.getItem('token');

            // let token =
            "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJqYXZhW51c2UiLCJleHAiOjE2MjE5NzQ1NjEsImhhdCI6
            MTYyMTk1NjU2MX0.FXRVeXEjJF7ReeGYcKlQvP_U_8hCwyviL_VryIQdWGRHVJi1-o3_cPSILqMV-
            f2xQ2ZiFr2UsPJqUrSQ075e8g";//localStorage.getItem('token');

            // Append Authorization header.
            headers.append('Authorization' , 'Bearer ' + token);

            // create object of RequestOptions and include that in it.
            let options = new RequestOptions( { headers : headers } );

            return this.http.post(url , cuisine, options);

            // //let url = this.baseUrl + "saveAdmin";
            // return this.http.post("http://localhost:9090/product", cuisine);
        }

```

```

deletProductById(id : number) : Observable<any>
{
    //let url = this.baseUrl + "saveAdmin";
    return this._httpClient.delete("http://localhost:9090/product/" + id);
    //sssreturn this.getAllCuisines();
}

getSelectedItems(){

    for(let cuisine of this.cuisineList){
        if(cuisine.quantity>0){
            if(!this.cart.includes(cuisine)){
                this.cart.push(cuisine);
            }
        }
    }

    return this.cart;
}

getCart(){
    return this.cart;
}
}

```

Cuisine.ts

```

export class Cuisine {
    product_id: number;
    product_name: String;
    image: String;
    description : String;
    price: number
    quantity: number = 0;
    enabled: boolean = true;
}

```

Pagenotfound

Pagenotfound.component.html

```

<p>pagenotfound works!</p>

```

Product

Product.component.html

```
<section class="jumbotron text-center">
  <div class="container">
    <h1 class="jumbotron-heading">Foodbox The Official Home Food
    </h1>
    <p class="lead text-muted">Manage our cuisines..</p>
    <!-- <p>
      <a href="#" class="btn btn-primary my-2">Main call to
action</a>

      <a href="#" class="btn btn-secondary my-2">Secondary
action</a>

    </p> -->
  </div>
</section>
<!-- <button class="btn btn-success btn-lg float-right"
type="submit">Add New</button> -->
<!-- <ul class="nav nav-pills">
  <li class="nav-item"><a [routerLink]="[/add-product]" class="nav-
link">Add New </a></li>
</ul> -->
<button type="button" class="btn btn-primary btn-lg">
  <a routerLink="../add-product" class="nav-link active"
style="color: aliceblue;"> Add New</a></button>

<div class="album py-5 bg-light">
  <div class="container">

    <div class="row">
      <div class="col-md-4" *ngFor="let cuisine of cuisineList">
        <div class="card mb-4 shadow-sm">
          
          <div class="card-body">
            <p class="card-text"><b>{{ cuisine.product_name
}}</b></p>

            <p class="card-text">{{ cuisine.description }}</p>

            <div class="d-flex justify-content-between align-
items-center">

              <div class="form-check-inline">
                <ul class="nav nav-pills">
                  <!-- <li class="nav-item"><a href="#"
(click)="addDetails()" class="nav-link">Delete</a></li> -->
```

```

                <li><button class="btn btn-primary profile-
button" (click)="deleteProduct(cuisine.product_id)">Delete</button></li>
                <!-- <li class="nav-item"><a
[routerLink]='["/product-detail", cuisine.product_id]' class="nav-
link">Edit</a></li>
                -->
                <li class="nav-item"><a
[routerLink]='["/product-detail", cuisine.product_id]' class="nav-
link">Edit</a></li>
                <li class="nav-item"> <small class="text-
muted">INR{{cuisine.price}}</small>
                </li>
            </ul>
        </div>
    </div>
</div>
</div>
</div>

```

Product.component.ts

```

import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import { CuisineService } from '../models/cuisine.service';

@Component({
  selector: 'app-product',
  templateUrl: './product.component.html',
  styleUrls: ['./product.component.css']
})
export class ProductComponent implements OnInit {

  cuisinelist: any;
  title: any;

  cache : any;
  cachedObservable : any;

  constructor(private _httpClient: HttpClient, private _cuisineService:
CuisineService) {
    // this.title = this._cuisineService.title;
    // this.message = this._userService.getMessage();
  }

```

```

}

ngOnInit(): void {
  //
  this._httpClient.get("http://localhost:3002/cuisines").subscribe((result) => {
    //   this.cuisineList = result;
    //   console.log(this.cuisineList);
    // }, (errors) => { console.log(errors); })

    this._cuisineService.getAllCuisines().subscribe((result) => {
      this._cuisineService.cuisineList = result.json();
      this.cuisineList = this._cuisineService.cuisineList;
      console.log(this.cuisineList);
    }, (errors => console.log('There are some errors : ' + errors)))

    // this.cuisineList = this._cuisineService.getAllCuisines();
  }

  public deleteProduct(id){
    console.log("In delete, id : "+ id);
    this._cuisineService.deleteProductById(id).subscribe((result) => {
      this._cuisineService.cuisineList = result;
      this.cuisineList = this._cuisineService.cuisineList;
      console.log(this.cuisineList);
    }, (errors => console.log('There are some errors : ' + errors)))
    this.cuisineList = this._cuisineService.cuisineList;
    console.log(this.cuisineList);
  }
}

```

Product-detail

Product-detail.component.css

```

body {
  background: rgb(99, 39, 120)
}

.form-control:focus {
  box-shadow: none;
  border-color: #BA68C8
}

.profile-button {
  background: rgb(99, 39, 120);
  box-shadow: none;
}

```

```

        border: none
    }

    .profile-button:hover {
        background: #682773
    }

    .profile-button:focus {
        background: #682773;
        box-shadow: none
    }

    .profile-button:active {
        background: #682773;
        box-shadow: none
    }

    .back:hover {
        color: #682773;
        cursor: pointer
    }

    .labels {
        font-size: 11px
    }

    .add-experience:hover {
        background: #BA68C8;
        color: #fff;
        cursor: pointer;
        border: solid 1px #BA68C8
    }
}

```

Product-detail.component.html

```

<div class="container rounded bg-white mt-5 mb-5">
    <!-- <form [formGroup]="form" #ProductInformation
    (ngSubmit)="save(ProductInformation)"> -->
    <!-- <form onsubmit="saveDetails()"> -->
    <div class="row">
        <div class="col-md-3 border-right">
            <!-- <div class="d-flex flex-column align-items-center text-center
            p-3 py-5"><img class="rounded-circle mt-5"
                src={{cuisine.image}}></div> -->
            

            </div>

```

```

        <div class="col-md-5 border-right">
            <div class="p-3 py-5">
                <div class="d-flex justify-content-between align-items-center mb-3">
                    <h4 class="text-right">Edit Product Details</h4>
                </div>
                <div class="row mt-2">
                    <div class="col-md-6"><label class="labels">Product
Id</label>
                        <input type="text" class="form-control" disabled
value="{{cuisine.product_id}}">
                    </div>
                    <div class="col-md-6"><label class="labels">Product
Name</label>
                        <input type="text" class="form-control" disabled
value="{{cuisine.product_name}}" >
                    </div>
                </div>
                <div class="row mt-3">
                    <div class="col-md-12"><label
class="labels">Description</label>
                        <input type="text" class="form-control"
name="description" placeholder="Description"
[src]="cuisine.description"
[(ngModel)]="cuisine.description">
                    </div>
                    <div class="col-md-12"><label class="labels">Price
(INR)</label>
                        <input type="text" class="form-control" name="price"
placeholder="Price in Rs."
[src]="cuisine.price" [(ngModel)]="cuisine.price">
                    </div>
                </div>
                <div class="row mt-3">
                    <div class="col-md-12"><label class="labels">Enabled
</label>
                        <input [(ngModel)]="cuisine.enabled" type="checkbox"/>
                        <!-- <input type="checkbox" class="form-control"
name="enabled"
[src]="cuisine.enabled" [(ngModel)]="cuisine.enabled"
> -->
                    </div>
                </div>
                <div class="row mt-3">
                    <div class="col-md-12"><label class="labels">Image</label>
                        <input type="text" class="form-control" name="image"
placeholder="image path"

```

```

                [src]="cuisine.image" [(ngModel)]="cuisine.image"
value="">
            </div>
        </div>
        <div class="mt-5 text-center"><button class="btn btn-primary
profile-button" (click)="saveDetails()" type="button">Save</button></div>
    </div>
</div>
<!-- </form> -->
</div>

```

Product-detail.component.ts

```

import { Cuisine } from '../models/Cuisine';
import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { CuisineService } from '../models/cuisine.service';

@Component({
  selector: 'app-product-detail',
  templateUrl: './product-detail.component.html',
  styleUrls: ['./product-detail.component.css']
})
export class ProductDetailComponent implements OnInit {

  id: any;
  cuisine: Cuisine = new Cuisine();

  constructor(private _cuisineService: CuisineService, private _route:
  ActivatedRoute, private _httpClient: HttpClient) { }

  ngOnInit(): void {
    this.id = this._route.snapshot.params.id;
    this._cuisineService.getCuisineById(this.id).subscribe((result) => {
      this.cuisine = result;
      console.log(this.cuisine);
    }, (errors => console.log('There are some errors : ' + errors)))
    console.log(this.id);
  }

  // // create the form object.
  // form = new FormGroup({
  //   fullName : new FormControl('' , Validators.required),

```



```
//      email : new FormControl('' , Validators.required),
//      password : new FormControl('' , Validators.required),
//      confirmPassword : new FormControl('' , Validators.required),
//      role : new FormControl('' , Validators.required),
//  });

// AdminForm(AdminInformation)
// {

  public saveDetails(){
    this._cuisineService.saveProductDetails(this.cuisine).subscribe();
  }

}
```

Quiz

Quiz.component.css

```
body {
  background-color: #eee
}

label.radio {
  cursor: pointer
}

label.radio input {
  position: absolute;
  top: 0;
  left: 0;
  visibility: hidden;
  pointer-events: none
}

label.radio span {
  padding: 4px 0px;
  border: 1px solid red;
  display: inline-block;
  color: red;
  width: 100px;
  text-align: center;
  border-radius: 3px;
  margin-top: 7px;
  text-transform: uppercase
}
```

```

label.radio input:checked+span {
  border-color: red;
  background-color: red;
  color: #fff
}

.ans {
  margin-left: 36px !important
}

.btn:focus {
  outline: 0 !important;
  box-shadow: none !important
}

.btn:active {
  outline: 0 !important;
  box-shadow: none !important
}

```

Quiz.component.html

```

<div class="container mt-5">
  <div class="d-flex justify-content-center row">
    <div class="col-md-10 col-lg-10">
      <div class="border">
        <div class="question bg-white p-3 border-bottom">
          <div class="d-flex flex-row justify-content-between align-items-center mcq">
            <h4>MCQ Quiz</h4>
          </div>
        </div>
        <div class="question bg-white p-3 border-bottom">
          <div class="d-flex flex-row align-items-center question-title">
            <h3 class="text-danger">Q.</h3>
            <h5 class="mt-1 ml-2">{{
questions[currentIndex].question }} ?</h5>
          </div>
          <div class="ans ml-auto">
            <label class="radio">
              <input type="radio"
                [value]="questions[currentIndex].options[0].id"
                name="selectedAnswer"
                [(ngModel)]="questions[currentIndex].selectedAnswer" />
            <span>{{questions[currentIndex].options[0].answer}}</span>

```

```

        </label>
    </div>
    <div class="ans ml-2">
        <label class="radio">
            <input type="radio"
                [value]="questions[currentIndex].options[1].id"
                name="selectedAnswer"
                [(ngModel)]="questions[currentIndex].selectedAnswer" />
        </label>
        <span>{{questions[currentIndex].options[1].answer}}</span>
    </div>
    <div class="ans ml-2">
        <label class="radio">
            <input type="radio"
                [value]="questions[currentIndex].options[2].id"
                name="selectedAnswer"
                [(ngModel)]="questions[currentIndex].selectedAnswer" />
        </label>
        <span>{{questions[currentIndex].options[2].answer}}</span>
    </div>
    <div class="ans ml-2">
        <label class="radio">
            <input type="radio"
                [value]="questions[currentIndex].options[3].id"
                name="selectedAnswer"
                [(ngModel)]="questions[currentIndex].selectedAnswer" />
        </label>
        <span>{{questions[currentIndex].options[3].answer}}</span>
    </div>
    <div class="d-flex flex-row justify-content-between align-items-center p-3 bg-white">
        <button class="btn btn-primary d-flex align-items-center btn-danger" type="button"
            (click)="getPreviousQuestion()">
            <i class="fa fa-angle-left mt-1 mr-1"></i>&nbsp;previous</button>

        <button class="btn btn-primary border-success align-items-center btn-success" type="button"
            (click)="submit()">
            Submit<i class="fa fa-angle-right ml-2"></i></button>

        <button class="btn btn-primary border-success align-items-center btn-success" type="button"

```

```

                (click)="getNextQuestion()">
                Next<i class="fa fa-angle-right ml-
2"></i></button></div>
            </div>
        </div>
    </div>
</div>

```

Quiz.component.ts

```

import { HttpClient } from '@angular/common/http';
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';

@Component({
  selector: 'app-quiz',
  templateUrl: './quiz.component.html',
  styleUrls: ['./quiz.component.css']
})
export class QuizComponent implements OnInit {

  questions : any;
  currentQuestion : any;
  currentIndex: number = 0;

  marksOutOf : number = 0;
  marksObtained : number = 0;

  constructor(private _httpClient: HttpClient, private _router: Router) { }

  ngOnInit(): void {
    this._httpClient.get("http://localhost:3002/questions")
      .subscribe(
        (result) => {
          this.questions = result;
          console.log(this.questions[0].question);
        },
        (errors) => {console.log(errors)}
      )
  }

  getNextQuestion(){
    console.log("getNextQuestion" + this.questions.length);
    console.log(this.questions[this.currentIndex])
    if(this.currentIndex<this.questions.length){

```

```

        this.currentIndex++;
    }

}

getPreviousQuestion(){
    console.log("getPreviousQuestion");
    if(this.currentIndex > 0){
        this.currentIndex--;
    }
}

submit(){
    console.log("SUBMITTED");

    this.questions.forEach(element => {
        this.marksOutOf = this.marksOutOf + element.weightage;
        console.log('element.correctAnswer.id = ' + element.correctAnswer.id );
        console.log('element.selectedAnswer = ' + element.selectedAnswer );

        if(element.correctAnswer.id == element.selectedAnswer){
            this.marksObtained = this.marksObtained + element.weightage;
        }
        this._router.navigate(['/result'],
        { queryParams:
            { marksOutOf: this.marksOutOf, marksObtained: this.marksObtained} }
        );

        console.log('Quiz.marksOutOf = ' + this.marksOutOf)
        console.log('Quiz.marksObtained = ' + this.marksObtained)

    });
}
}

```

Result

Result.component.html

```

<p>Your Score is...!!</p>
<p>Marks out of : {{ marksOutOf }}</p>
<p>Marks Obtained : {{ marksObtained }}</p>

<button type="button" class="btn btn-primary btn-lg">
    <a routerLink="../quiz" class="nav-link active" style="color: aliceblue;">
Restart Test</a></button>

```

Result.component.ts

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { Router } from '@angular/router';

@Component({
  selector: 'app-result',
  templateUrl: './result.component.html',
  styleUrls: ['./result.component.css']
})
export class ResultComponent implements OnInit {

  marksOutOf: string;
  marksObtained: string;

  constructor(private _route: ActivatedRoute, private _router : Router) {}

  ngOnInit() {
    this._route.queryParamMap.subscribe(
      (queryParamMap) => {
        console.log(queryParamMap);
        queryParamMap.get
        this.marksOutOf = queryParamMap.get('marksOutOf');
        this.marksObtained = queryParamMap.get('marksObtained');
      }
    );

    console.log('this.marksOutOf = ' + this.marksOutOf)
    console.log('this.marksObtained = ' + this.marksObtained)
  }
}
```

Signup

Signup.component.html

```
<form [formGroup]="form" #AdminInformation
(ngSubmit)="AdminForm(AdminInformation)">

  <div class="row">
    <div class=" col-md-offset-1 col-md-4">
```

```

        <h2>Sign Up</h2>
    </div>
</div>
<div class="row">
    <div class=" col-md-offset-1 col-md-4">
        <h2></h2>
    </div>
</div>

<div class="row">
    <div class=" col-md-offset-1 col-md-4">
        <label for="fullName"> Name </label>
        <input formControlName="fullName" class="form-control" type="text">
    </div>
</div>

<div class="row">
    <div class=" col-md-offset-1 col-md-4">
        <label for="email"> Email </label>
        <input formControlName="email" class="form-control" type="text">
    </div>
</div>

<div class="row">
    <div class=" col-md-offset-1 col-md-4">
        <label for="password"> Password </label>
        <input formControlName="password" class="form-control"
type="password">
    </div>
</div>

<div class="row">
    <div class=" col-md-offset-1 col-md-4">
        <label for="confirmPassword"> Confirm Password </label>
        <input formControlName="confirmPassword" class="form-control"
type="password">
    </div>
</div>

<div class="row">
    <div class=" col-md-offset-1 col-md-4">
        <label for="role"> Role </label>
        <input formControlName="role" class="form-control" type="text">
    </div>
</div>

<div class="row" style="margin-top: 40px;">
    <div class="col-md-offset-1 col-md-4">

```

```

        <button class="btn btn-md btn-primary btn-style" >Save</button>
    </div>
</div>

</form>

```

Signup.component.ts

```

import { Component, OnInit } from '@angular/core';
import { FormGroup, FormControl, Validators } from '@angular/forms';
import { AdminDetail } from '../models/admin-detail';
import { AdminService } from '../models/admin.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-signup',
  templateUrl: './signup.component.html',
  styleUrls: ['./signup.component.css']
})
export class SignupComponent implements OnInit {

  private adminDetail = new AdminDetail();

  constructor(private adminService : AdminService, private router : Router) {
  }

  ngOnInit() {
  }

  // create the form object.
  form = new FormGroup({
    fullName : new FormControl('', Validators.required),
    email : new FormControl('', Validators.required),
    password : new FormControl('', Validators.required),
    confirmPassword : new FormControl('', Validators.required),
    role : new FormControl('', Validators.required),
  });

  AdminForm(AdminInformation)
  {
    let pass = this.Password.value;
    let confirmPass = this.ConfirmPassword.value;

    if(pass == confirmPass)
    {
      this.adminDetail.name = this.FullName.value;
      this.adminDetail.username = this.Email.value;
    }
  }
}

```



```

    this.adminDetail.password = this.Password.value;
    this.adminDetail.role = this.Role.value;

    this.adminService.saveAdminDetails(this.adminDetail).subscribe(
      response => {
        let result = response.json();

        if(result != null)
        {
          this.router.navigate(['/login']);
        }
        else
        {
          alert("error occur while registering User. please try after
sometime.")
        }
      },
      error => {
        alert("error occur while registering User. please try after
sometime.")
      }
    );

  }
  else
  {
    alert("Password and confirm password not match.");
  }
}

get FullName(){
  return this.form.get('fullName');
}

get Email(){
  return this.form.get('email');
}

get Password(){
  return this.form.get('password');
}

get ConfirmPassword(){
  return this.form.get('confirmPassword');
}

get Role(){
  return this.form.get('role');
}

```

```
}  
  
}
```

Successpage

Sucesspage.component.html

```
<div class="container">  
  <div class="card border-0 shadow my-5">  
    <div class="card-body p-5">  
      <div class="col-md-6 mx-auto mt-5">  
        <div class="payment">  
          <div class="payment_header">  
            <div class="check"><i class="fa fa-check" aria-  
hidden="true"></i></div>  
          </div>  
          <div class="content">  
            <h1>Payment Successful !</h1>  
            <p>Sit back, your delicious FoodBox will be reaching you on  
time. You will receive call from our support team for further details. Thank  
you!! </p>  
          </div>  
        </div>  
      </div>  
    </div>  
  </div>  
</div>
```

App-routing.module.ts

```
import {Routes} from "@angular/router"  
import { AboutComponent } from "../about/about.component"  
import { AddProductComponent } from "../add-product/add-product.component"  
import { CartComponent } from "../cart/cart.component"  
import { CheckoutComponent } from "../checkout/checkout.component"  
import { HomeComponent } from "../home/home.component"  
import { LoginComponent } from "../login/login.component"  
import { LogoutComponent } from "../logout/logout.component"  
import { MenuComponent } from "../menu/menu.component"  
import { PagenotfoundComponent } from "../pagenotfound/pagenotfound.component"  
import { ProductDetailComponent } from "../product-detail/product-  
detail.component"  
import { ProductComponent } from "../product/product.component"
```

```

import { QuizComponent } from "../quiz/quiz.component"
import { ResultComponent } from "../result/result.component"
import { SignupComponent } from "../signup/signup.component"
import { SuccesspageComponent } from "../successpage/successpage.component"

export const applicationRoutes: Routes = [
  // {path: '', component: HomeComponent},
  {path: '', redirectTo:'home', pathMatch:'full'},
  {path: 'quiz', component: QuizComponent},
  {path: 'home', component: HomeComponent},
  {path: 'menu', component: MenuComponent},
  {path: 'cart', component: CartComponent},
  {path: 'product', component: ProductComponent},
  {path: 'login', component: LoginComponent},
  {path: 'logout', component: LogoutComponent},
  { path: 'product-detail/:id', component: ProductDetailComponent },
  {path: 'signup', component: SignupComponent},
  {path: 'add-product', component: AddProductComponent},
  {path: 'checkout', component: CheckoutComponent},
  {path: 'success', component: SuccesspageComponent},
  {path: 'about', component: AboutComponent},
  {path: 'result', component: ResultComponent},
  {path: '**', component: PagenotfoundComponent}
]

```

App.module.ts

```

import { NgModule } from '@angular/core';
import { RouterModule } from "@angular/router";
import { BrowserModule } from '@angular/platform-browser';
import { HttpClientModule } from '@angular/http';

import { MainComponent } from '../main/main.component';
import { HomeComponent } from '../home/home.component';
import { AboutComponent } from '../about/about.component';
import { PagenotfoundComponent } from '../pagenotfound/pagenotfound.component';
import { applicationRoutes } from '../app-routing.module';
import { FooterComponent } from '../footer/footer.component';
import { QuizComponent } from '../quiz/quiz.component';
import { HttpClientModule } from '@angular/common/http';
import { ResultComponent } from '../result/result.component';
import { MenuComponent } from '../menu/menu.component';
import { CuisineService } from '../models/cuisine.service';
import { CartComponent } from '../cart/cart.component';
import { ProductComponent } from '../product/product.component';
import { ProductDetailComponent } from '../product-detail/product-
detail.component';

```

```

import { LoginComponent } from './login/login.component';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { AddProductComponent } from './add-product/add-product.component';
import { CheckoutComponent } from './checkout/checkout.component';
import { SuccesspageComponent } from './successpage/successpage.component';
import { SignupComponent } from './signup/signup.component';
import { LogoutComponent } from './logout/logout.component';

@NgModule({
  declarations: [
    MainComponent,
    HomeComponent,
    AboutComponent,
    PagenotfoundComponent,
    FooterComponent,
    QuizComponent,
    ResultComponent,
    MenuComponent,
    CartComponent,
    ProductComponent,
    ProductDetailComponent,
    LoginComponent,
    AddProductComponent,
    CheckoutComponent,
    SuccesspageComponent,
    SignupComponent,
    LogoutComponent
  ],
  imports: [
    BrowserModule,
    HttpClientModule,
    HttpModule,
    FormsModule,
    ReactiveFormsModule,
    RouterModule.forRoot(applicationRoutes)
  ],
  providers: [CuisineService],
  bootstrap: [MainComponent]
})
export class AppModule { }

```

index.html

```

<!doctype html>
<html lang="en">
<head>

```

```

<meta charset="utf-8">
<title>OnlineTestApp</title>
<base href="/">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" type="image/x-icon" href="favicon.ico">
  <!-- CSS only -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-
e0JMYsd53ii+sc0/bJGfSiCZc+5NDVN2yr8+0RDqr0Ql0h+rpP48ckxlpbzkGwra6"
    crossorigin="anonymous" />
</head>
<body>
  <app-main></app-main>
  <!-- JavaScript Bundle with Popper -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
JEW9xMcG8R+ph31jmWwH6WWP0WintQrMb4s7Z0dauHnUtxwoG2vI5DkLtS3qm9Ekf"
    crossorigin="anonymous"></script>
  <app-footer></app-footer>
</body>
</html>

```

Menu.json

```

{
  "cuisines": [
    {
      "id": 1,
      "name": "Cuisine1",
      "image": "assets\\img\\italian-food.jpg",
      "description": "description1 description1 description1
description1",
      "price": 100
    },
    {
      "id": 2,
      "name": "Cuisine2",
      "image": "assets\\img\\italian-food.jpg",
      "description": "description2 description2 description2 description2",
      "price": 200
    },
    {
      "id": 3,
      "name": "Cuisine3",
      "image": "assets\\img\\italian-food.jpg",

```

```

        "description": "description3 description3 description3 description3",
        "price": 300
    },
    {
        "id": 4,
        "name": "Cuisine4",
        "image": "assets\\img\\italian-food.jpg",
        "description": "description4 description4 description4 description4",
        "price": 400
    },
    {
        "id": 5,
        "name": "Cuisine5",
        "image": "assets\\img\\italian-food.jpg",
        "description": "description5 description5 description5 description5",
        "price": 500
    }
]
}

```

Onlinetest.json

```

{
    "questions": [
        {
            "id": 1,
            "question": "What guarantees type-safety in a collection",
            "weightage": 2,
            "options": [
                {"id": 1, "answer": "Generics"},
                {"id": 2, "answer": "Abstract classes"},
                {"id": 3, "answer": "Interfaces"},
                {"id": 4, "answer": "Collection"}
            ],
            "correctAnswer": {"id": 1, "answer": "Generics"},
            "selectedAnswer": {"id": 4, "answer": "Collection"}
        },
        {
            "id": 1,
            "question": "HashSet internally uses",
            "weightage": 2,
            "options": [
                {"id": 1, "answer": "Set"},
                {"id": 2, "answer": "HashMap"},
                {"id": 3, "answer": "List"},
                {"id": 4, "answer": "Collection"}
            ],
            "correctAnswer": {"id": 2, "answer": "HashMap"},
            "selectedAnswer": {"id": 4, "answer": "Collection"}
        }
    ]
}

```

```

{
  "id": 2,
  "question": "The most used interfaces from the collection
framework are",
  "weightage": 2,
  "options": [{"id":1, "answer":"List"}
              ,{"id":2, "answer":"Map"}
              ,{"id":3, "answer":"Set"}
              ,{"id":4, "answer":"Class"}
              ],
  "correctAnswer": {"id":1, "answer":"List"},
  "selectedAnswer": {"id":4, "answer":"Class"}
},
{
  "id": 3,
  "question": "The root interface of Java collection framework
hierarchy is",
  "weightage": 2,
  "options": [{"id":1, "answer":"Collection"}
              ,{"id":2, "answer":"Root"}
              ,{"id":3, "answer":"Collections"}
              ,{"id":4, "answer":"Iterable"}
              ],
  "correctAnswer": {"id":4, "answer":"Iterable"},
  "selectedAnswer": {"id":2, "answer":"Root"}
},
{
  "id": 4,
  "question": "LinkedList implements",
  "weightage": 2,
  "options": [{"id":1, "answer":"List"}
              ,{"id":2, "answer":"Deque"}
              ,{"id":3, "answer":"None of the Above"}
              ,{"id":4, "answer":"Both of Above"}
              ],
  "correctAnswer": {"id":4, "answer":"Both of Above"},
  "selectedAnswer": {"id":3, "answer":"None of the Above"}
},
{
  "id": 5,
  "question": "Which of these methods deletes all the elements from
invoking collection",
  "weightage": 2,
  "options": [{"id":1, "answer":"clear()"}
              ,{"id":2, "answer":"reset()"}
              ,{"id":3, "answer":"delete()"}
              ,{"id":4, "answer":"refresh()"}
              ],

```

```

        "correctAnswer": { "id": 1, "answer": "clear()" },
        "selectedAnswer": { "id": 4, "answer": "refresh()" }
    }
}
]
}

```

EC2-SRC

App.ec2.spec.ts

```

import { browser, logging } from 'protractor';
import { AppPage } from './app.po';

describe('workspace-project App', () => {
    let page: AppPage;

    beforeEach(() => {
        page = new AppPage();
    });

    it('should display welcome message', async () => {
        await page.navigateTo();
        expect(await page.getTitleText()).toEqual('online-test-app app is running!');
    });

    afterEach(async () => {
        // Assert that there are no errors emitted from the browser
        const logs = await browser.manage().logs().get(logging.Type.BROWSER);
        expect(logs).not.toContain(jasmine.objectContaining({
            level: logging.Level.SEVERE,
        } as logging.Entry));
    });
});

```

App.po.ts

```

import { browser, by, element } from 'protractor';

export class AppPage {
    async navigateTo(): Promise<unknown> {
        return browser.get(browser.baseUrl);
    }

    async getTitleText(): Promise<string> {

```



```
    return element(by.css('app-root .content span')).getText();
  }
}
```

Protractor.conf.js

```
// @ts-check
// Protractor configuration file, see link for more information
// https://github.com/angular/protractor/blob/master/lib/config.ts

const { SpecReporter, StacktraceOption } = require('jasmine-spec-reporter');

/**
 * @type { import("protractor").Config }
 */
exports.config = {
  allScriptsTimeout: 11000,
  specs: [
    './src/**/*.e2e-spec.ts'
  ],
  capabilities: {
    browserName: 'chrome'
  },
  directConnect: true,
  SELENIUM_PROMISE_MANAGER: false,
  baseUrl: 'http://localhost:4200/',
  framework: 'jasmine',
  jasmineNodeOpts: {
    showColors: true,
    defaultTimeoutInterval: 30000,
    print: function() {}
  },
  onPrepare() {
    require('ts-node').register({
      project: require('path').join(__dirname, './tsconfig.json')
    });
    jasmine.getEnv().addReporter(new SpecReporter({
      spec: {
        displayStacktrace: StacktraceOption.PRETTY
      }
    }));
  }
};
```

Tsconfig.json

```
/* To learn more about this file see: https://angular.io/config/tsconfig. */
{
  "extends": "../tsconfig.json",
  "compilerOptions": {
    "outDir": "../out-tsc/e2e",
    "module": "commonjs",
    "target": "es2018",
    "types": [
      "jasmine",
      "node"
    ]
  }
}
```