|  |  |
| --- | --- |
| **Full Name** | MEGHA MANOHAR DODKE |
| **Batch** | FS JAVA DEVELOPER-DEC-2021 |
| **Project Title** | Kitchen Story |
| **Project Submission Date** | 25-06-2022 |
| **Duration of Sprints** | 6 days |
| **GIT HUB DETAILS** | git@github.com:MeghaDodke/phase4-kitchenstory.git |

**Source Code**

**Components:**

**Addcake:**

**addcakecomponent.css**

.container{

    width:70%;

    margin:1em auto;

}

.form-group,button{

    margin-top:1em;

}

**Addcakecomponent.html**

<div class="container">

    <form [formGroup]="cakeForm" (submit)="cakeForm.valid && addcake()">

        <div class="form-group">

          <label for="exampleInputEmail1">Name of Cake</label>

          <input formControlName="name" class="form-control" aria-describedby="nameHelp" placeholder="Choclate cake" [(ngModel)]="cakedata.name">

          <div \*ngIf="cakeForm.controls['name'].invalid && (cakeForm.controls['name'].dirty || cakeForm.controls['name'].touched)" class="alert">

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

            Please enter cake  name

          </div>

        </div>

        </div>

        <div class="form-group">

          <label for="exampleInputEmail1">Price</label>

          <input formControlName="price" class="form-control" aria-describedby="weightHelp" placeholder="100" [(ngModel)]="cakedata.price">

          <div \*ngIf="cakeForm.controls['price'].invalid && (cakeForm.controls['price'].dirty || cakeForm.controls['price'].touched)" class="alert">

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

            Please enter cake price

          </div>

        </div>

        </div>

        <div class="form-group">

          <label for="exampleInputEmail1">Type of cake</label>

          <select class="form-select" formControlName="type">

              <option value="Birthday Cake">Birthday Cake</option>

              <option value="Fruit Cake">Fruit Cake</option>

              <option value="Muffin">Muffin</option>

              <option value="Sponge Cake">Sponge Cake</option>

          </select>

          <div \*ngIf="cakeForm.controls['type'].invalid && (cakeForm.controls['type'].dirty || cakeForm.controls['type'].touched)" class="alert">

            <div style=color:red \*ngIf="cakeForm.controls['type'].errors.required">

            Please select a type

          </div>

        </div>

        </div>

        <div class="form-group">

          <label for="exampleInputEmail1">Weight</label>

          <input formControlName="weight" class="form-control" aria-describedby="weightHelp" placeholder="Enter weight within the range of 0-5 to 5" [(ngModel)]="cakedata.weight">

          <div \*ngIf="cakeForm.controls['weight'].invalid && (cakeForm.controls['weight'].dirty || cakeForm.controls['weight'].touched)" class="alert">

            <div style=color:red \*ngIf="cakeForm.controls['weight'].errors.required">

            Please select a type

          </div>

        </div>

        </div>

        <div class="form-group">

          <label for="exampleInputEmail1">Ingredient</label>

          <input formControlName="ingredient" class="form-control" aria-describedby="emailHelp" placeholder="Provide comma after each item ex:item1,item2.." [(ngModel)]="cakedata.ingredient">

          <div \*ngIf="cakeForm.controls['ingredient'].invalid && (cakeForm.controls['ingredient'].dirty || cakeForm.controls['ingredient'].touched)" class="alert">

            <div style=color:red \*ngIf="cakeForm.controls['ingredient'].errors.required">

            Please enter ingredients

          </div>

        </div>

        </div>

        <div class="form-group">

          <label for="imageUrl">Upload a image</label>

          <div class="d-flex">

            <input  type="file" (change)="getFile($event)">

            <button class="btn btn-outline-success" (click)="upload()">Upload</button>

          </div>

        </div>

        <div class="form-group">

            <div>

              <input id="male" type="radio" value="eggless" name="eggless" formControlName="eggless">

                 <label for="male">Eggless</label>

            </div>

            <div>

              <input id="female" type="radio" value="egg" name="eggless" formControlName="eggless">

                <label for="female">Egg</label>

              </div>

              <div \*ngIf="cakeForm.controls['eggless'].invalid && (cakeForm.controls['eggless'].dirty || cakeForm.controls['eggless'].touched)" class="alert">

                <div style=color:red \*ngIf="cakeForm.controls['eggless'].errors.required">

                Please enter the description

              </div>

            </div>

          </div>

        <div class="form-group">

            <label for="exampleFormControlTextarea1" class="form-label">Description</label>

            <textarea formControlName="description" class="form-control" id="exampleFormControlTextarea1" rows="3" [(ngModel)]="cakedata.description"></textarea>

            <div \*ngIf="cakeForm.controls['description'].invalid && (cakeForm.controls['description'].dirty || cakeForm.controls['description'].touched)" class="alert">

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

              Please enter the description

            </div>

          </div>

          </div>

          <button style="margin-top:1em" type="submit" class="btn btn-primary">Submit</button>

      </form>

</div>

**Addcakecomponent.specs.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { AddcakeComponent } from './addcake.component';

describe('AddcakeComponent', () => {

  let component: AddcakeComponent;

  let fixture: ComponentFixture<AddcakeComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ AddcakeComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(AddcakeComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Addcakecomponent.ts**

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

import { CakeService } from 'src/app/service.service';

import {FormBuilder,Validators} from "@angular/forms"

import { ToastrService } from 'ngx-toastr';

@Component({

  selector: 'app-addcake',

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

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

})

export class AddcakeComponent implements OnInit {

  cakedata:any={}

  file:any

  cakeIngredient:any

  cakeForm:any

  imageSource:any

  constructor(private service: CakeService,private formbuilder:FormBuilder,private toaster:ToastrService) {

    this.cakeForm=this.formbuilder.group({

      name:["",[Validators.required ]],

      price:[0,[Validators.required]],

      type:["",[Validators.required]],

      weight:[0,[Validators.required,Validators.max(5),Validators.min(0.5)]],

      ingredient:["",[Validators.required]],

      eggless:["",[Validators.required]],

      description:["",[Validators.required]]

    })

   }

  ngOnInit(): void {

  }

  addcake(){

    if(this.cakeForm.valid){

    let body={

      name:this.cakeForm["value"].name,

      price:parseInt(this.cakeForm["value"].price),

      description:this.cakeForm["value"].description,

      ingredients:this.cakeForm["value"].ingredient.split(','),

      image:this.imageSource,

      type:this.cakeForm["value"].type,

      weight:this.cakeForm["value"].weight,

      eggless:this.cakeForm["value"].eggless=="eggless"?true:false

    }

    let url="https://apifromashu.herokuapp.com/api/addcake"

    this.service.cake\_post(url,body).subscribe({

      next:(response:any)=>{

        console.log(response.data)

        this.toaster.success("New cake has been succesfully added to cake database")

      },

      error:(error)=>{

        console.log("Error from api",error)

      }

    })

    }

    else{

      console.log("The form is not valid")

    }

  }

  selectTypeofcake(event: any){

    console.log(event.target.value)

  }

  getFile(event:any){

    this.file=event.target.files[0]

    console.log("Get fILE")

    console.log(this.file)

  }

  upload(){

    console.log("File",this.file)

    var url="https://apifromashu.herokuapp.com/api/upload"

    var formdata=new FormData()

    formdata.append("file",this.file)

    this.service.cake\_post(url,formdata).subscribe({

      next:(response:any)=>{

        if(response){

          console.log(response)

          this.imageSource=response.imageUrl

          this.toaster.success("Image has been succesfully loaded the image source is",response.imageUrl)

        }

      },

      error:(error)=>{

        console.log("Error",error)

      }

    })

  }

}

**Address:**

**addresscomponent.html**

<div class="container">

    <form [formGroup]="addressform" (submit)="addressform.valid && addAddress()">

    <div class="form-group mt-3">

        <label for="">Name</label>

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

        <div \*ngIf="addressform.controls['name'].invalid && (addressform.controls['name'].dirty || addressform.controls['name'].touched)" class="alert">

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

            Please enter your name

          </div>

        </div>

    </div>

    <div class="form-group mt-3">

        <label for="">Address</label>

        <input formControlName="address" required [(ngModel)]="userDetail.address"  type="text" class="form-control">

        <div \*ngIf="addressform.controls['address'].invalid && (addressform.controls['address'].dirty || addressform.controls['address'].touched)" class="alert">

            <div  style=color:red \*ngIf="addressform.controls['address'].errors.required">

            Please enter your address

          </div>

        </div>

    </div>

     <div class="form-group mt-3">

        <label for="">City</label>

        <input formControlName="city" required [(ngModel)]="userDetail.city"  type="text" class="form-control">

        <div \*ngIf="addressform.controls['city'].invalid && (addressform.controls['city'].dirty || addressform.controls['city'].touched)" class="alert">

            <div  style=color:red \*ngIf="addressform.controls['city'].errors.required">

            Please enter your city

          </div>

        </div>

    </div>

    <div class="form-group mt-3">

        <label for="">Pincode</label>

        <input formControlName="pincode" required [(ngModel)]="userDetail.pincode"  type="text" class="form-control">

        <div \*ngIf="addressform.controls['pincode'].invalid && (addressform.controls['pincode'].dirty || addressform.controls['pincode'].touched)" class="alert">

            <div  style=color:red \*ngIf="addressform.controls['pincode'].errors.required">

            Please enter your pincode

          </div>

        </div>

    </div>

    <div class="form-group mt-3">

        <label for="">Phone no</label>

        <input formControlName="phone\_no" required [(ngModel)]="userDetail.phone\_no"  type="text" class="form-control">

        <div \*ngIf="addressform.controls['phone\_no'].invalid && (addressform.controls['phone\_no'].dirty || addressform.controls['phone\_no'].touched)" class="alert">

            <div  style=color:red \*ngIf="addressform.controls['phone\_no'].errors.required">

            Please enter your phone no

          </div>

        </div>

    </div>

    <button style="margin-top:1em" type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

**Addresscomponent.specs.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { AddressComponent } from './address.component';

describe('AddressComponent', () => {

  let component: AddressComponent;

  let fixture: ComponentFixture<AddressComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ AddressComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(AddressComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Addresscomponent.ts**

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

import { ToastrService } from 'ngx-toastr';

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

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

@Component({

  selector: 'app-address',

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

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

})

export class AddressComponent implements OnInit {

  userDetail:any={

  }

  addressform:any

  constructor(private toaster:ToastrService,private formbuilder:FormBuilder,private router:Router) {

      this.addressform=this.formbuilder.group({

        name:["",[Validators.required]],

        address:["",[Validators.required]],

        city:["",[Validators.required]],

        pincode:["",[Validators.required,Validators.maxLength(6),Validators.minLength(6)]],

        phone\_no:["",[Validators.required,Validators.maxLength(10),Validators.minLength(10)]]

      })

  }

  addAddress(){

    if(this.addressform.valid){

      localStorage.setItem('user\_address',JSON.stringify(this.userDetail))

      this.toaster.success("Address is succesfully added!")

      this.router.navigate(["payment"])

    }

    else{

      this.toaster.error("Fill in the detail appropirately")

    }

  }

  ngOnInit(): void {

  }

}

**Cake:**

**Cakecomponents.css**

.product-card{

    width:100%;

    background-color:rgba(221, 221, 221, 0.285);

    height:20em;

    margin:1em auto;

}

.image{

    height:50%;

    background-color:yellow;

}

.info{

    text-align:center;

    padding:0.5em;

}

.bottom-row{

    display:flex;

    justify-content:space-around;

    flex-direction:row;

}

p{

    font-weight: bold;

}

**Cakecomponent.html**

<div (click)="detail()" class="product-card">

        <!--Image Container-->

        <div class="image">

            <img src={{cakedata.image}} alt={{cakedata.name}} style="width:100%;height:100%;object-fit: cover;">

        </div>

        <!--Product Info-->

        <div class="info">

            <h4>{{cakedata.name}}</h4>

            <p> {{cakedata.price | currency:"INR"}}</p>

        </div>

 </div>

**Cakecomponent.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { CakeComponent } from './cake.component';

describe('CakeComponent', () => {

  let component: CakeComponent;

  let fixture: ComponentFixture<CakeComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ CakeComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(CakeComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Cake.component.specs.ts**

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

import {Cake} from "src/app/Cake"

import {CakeService} from 'src/app/service.service';

import { ToastrService } from 'ngx-toastr';

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

@Component({

  selector: 'app-cake',

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

  styleUrls: ['./cake.component.css']

})

export class CakeComponent implements OnInit {

  signupForm:any

  @Input() cakedata!:Cake

  constructor(private service:CakeService,private toaster: ToastrService,private router:Router) { }

  detail(){

    console.log("Testing")

    this.router.navigate(

      ["detail/",this.cakedata.cakeid] )

  }

  ngOnInit(): void {

  }

}

**Cakelist:**

**Cake-list.component.css**

.main{

    margin:2em;

  }

p{

    cursor: pointer;

}

p:hover{

    font-weight: bold;

}

**Cake-list.component.html**

<style>

    .main{

      margin:2em;

    }

    .filter-container{

        display:flex;

        flex-direction: row;

        width:30%;

        margin-top:2em;

        justify-content:space-evenly;

    }

</style>

<div class="main">

      <h2>Discover our cakes!!</h2>

      <div class="filter-container">

          <h5>Sort by |</h5>

          <p (click)="ascSort()" id="t">Low to high</p> |

          <p (click)="dscSort()">High to low</p>

      </div>

      <div class="row">

        <div class="col-md-4"  \*ngFor="let cake of cakes">

          <app-cake [cakedata]="cake"></app-cake>

          </div>

      </div>

</div>

**Cake-list.component.specs.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { CakeListComponent } from './cake-list.component';

describe('CakeListComponent', () => {

  let component: CakeListComponent;

  let fixture: ComponentFixture<CakeListComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ CakeListComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(CakeListComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Cake-list.component.ts**

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

import {Cake} from "src/app/Cake"

import { CakeService } from 'src/app/service.service';

import { ToastrService } from 'ngx-toastr';

import {HttpClient} from "@angular/common/http"

import {NgxUiLoaderService} from 'ngx-ui-loader'

@Component({

  selector: 'app-cake-list',

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

  styleUrls: ['./cake-list.component.css']

})

export class CakeListComponent implements OnInit {

  //Defines array of object called cakes with data type Cake

  cakes!: Cake[];

  //Initializa the service

  constructor(

    private service:CakeService,

    private https: HttpClient,

)

    {

      var url='https://apifromashu.herokuapp.com/api/allcakes'

      this.https.get(url).subscribe({

        next:(response:any) => {

          console.log(response)

          this.cakes=response.data

        },

        error:(error)=>{

          console.log(error.message)

        }

      })

  }

  //Services

  ascSort(){

    this.cakes=this.service.ascending(this.cakes,"price")

  }

  dscSort(){

    this.cakes=this.service.descending(this.cakes,"price")

  }

  ngOnInit(): void {

  }

}

**Cart:**

**Cartcomponent.cs**

.inc\_dec:hover {

    background-color: RoyalBlue;

  }

**Cartcomponent.html**

<div \*ngIf="cartItems.length>0"class="container">

    <table class="table">

        <thead>

          <tr>

            <th scope="col">#</th>

            <th scope="col">Name</th>

            <th scope="col">Image</th>

            <th scope="col">Price</th>

            <th scope="col">Quantity</th>

            <th scope="col">Action</th>

          </tr>

        </thead>

        <tbody>

          <tr \*ngFor="let item of cartItems ; let i=index">

            <th scope="row">{{i+1}}</th>

            <td>{{item.name}}</td>

            <td><img style="height: 60px; width:60px" [src]="item.image" alt=""></td>

            <td>{{item.price | currency:"INR"}}</td>

            <td><button (click)="inc\_dec('+',i)" class="btn btn-light">+</button>{{item.quantity}}<button class="btn btn-light"  (click)="inc\_dec('-',i)">-</button></td>

            <td><button (click)="removefromcart(i)" class="btn btn-danger"><i class="bi bi-trash"></i></button></td>

          </tr>

        </tbody>

      </table>

      <div class="d-flex flex-wrap justify-content-between " style="height: 50px;">

          <h3>Total Billing Amount: {{totalAmount | currency:"INR"}}</h3>

          <button type="button" class="btn btn-warning" routerLink="/checkout">Checkout</button>

    </div>

</div>

<div \*ngIf="!cartItems.length" style="text-align:center;">

    <h2>Your Cart is empty</h2>

</div>

**Cartcomponent.specs.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { CartComponent } from './cart.component';

describe('CartComponent', () => {

  let component: CartComponent;

  let fixture: ComponentFixture<CartComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ CartComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(CartComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Cartcomponent.ts**

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

import {CakeService} from "src/app/service.service"

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

import {NgxUiLoaderService} from 'ngx-ui-loader'

@Component({

  selector: 'app-cart',

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

  styleUrls: ['./cart.component.css']

})

export class CartComponent implements OnInit {

  cartItems:any=[]

  totalAmount:number=0

  constructor(private service:CakeService,private router:Router,private loader:NgxUiLoaderService) {

    console.log("called")

        var url="https://apifromashu.herokuapp.com/api/cakecart"

        this.service.cake\_post(url,{}).subscribe({

          next:(response:any)=>{

            this.cartItems=response.data

            this.cartItems.map((item:any)=>{

              this.totalAmount+=item.quantity\*item.price

            })

          },

          error:(error)=>{

            console.log(error)

          }

        })

   }

   inc\_dec(character:string,index:any){

     let body={

       cakeid:this.cartItems[index].cakeid,

     }

     if(character=="-"){

       console.log("decrement has been called")

      this.service.cake\_post("https://apifromashu.herokuapp.com/api/removeonecakefromcart",body).subscribe({

        next:(response:any) => {

          if(response && this.cartItems[index].quantity>1){

            console.log(this.cartItems)

            this.totalAmount-=this.cartItems[index].price

            this.cartItems[index].quantity--

            this.loader.stop()

          }

          else if(this.cartItems[index].quantity==1 && this.cartItems.length>1){

            this.removefromcart(index)

          }

          else{

            this.cartItems.pop()

          }

        },

        error:(error)=>{

          console.log(error)

        }

      })

     }

     else{

       this.loader.start()

       this.service.cake\_post("https://apifromashu.herokuapp.com/api/addcaketocart", this.cartItems[index]).subscribe({

          next:(response:any)=>{

            if(response){

              console.log(this.cartItems)

              this.totalAmount+=this.cartItems[index].price

              this.cartItems[index].quantity+=1;

            }

          }

       })

     }

   }

  removefromcart(index:any){

    // hit the api

    var body = {

      cakeid:this.cartItems[index]["cakeid"]

    }

    var url = "https://apifromashu.herokuapp.com/api/removecakefromcart"

    this.loader.start()

    this.service.cake\_post(url,body).subscribe({

      next:(response:any)=>{

        this.loader.stop()

        console.log("Response from remove from cart api", response)

        if(response.message=="Removed whole cake  item from cart"){

          this.totalAmount = this.totalAmount-this.cartItems[index].quantity\*this.cartItems[index].price

          this.cartItems.splice(index,1)

        }

      },

      error:(error:any)=>{

        this.loader.stop()

        console.log("Error from remove from cart api")

      }

    })

 }

  ngOnInit(): void {

    this.loader.start();

    setTimeout(()=>{

      this.loader.stop()

    },2000)

  }

  ngOnChanges(): void {

  }

}

**Checkout:**

**Checkout.component.html**

<div class="container">

    <div class="row">

        <div class="col-md-4">

            <nav class="nav flex-column">

                <a class="nav-link" routerLink="address">Addresses</a>

                <a class="nav-link" routerLink="payment">Payment</a>

            </nav>

        </div>

        <div class="col-md-8">

            <router-outlet></router-outlet>

        </div>

        </div>

    </div>

**Checkout.component.specs.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { CheckoutComponent } from './checkout.component';

describe('CheckoutComponent', () => {

  let component: CheckoutComponent;

  let fixture: ComponentFixture<CheckoutComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ CheckoutComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(CheckoutComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Checkoutcomponents.ts**

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

@Component({

  selector: 'app-checkout',

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

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

})

export class CheckoutComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

}

**Detail-component:**

**Detail-component.component.html**

<div \*ngIf="cakeData" class="container mt-4">

    <h1>{{cakeData.name}}</h1>

    <section class="cake-details mb-5 pt-4 pb-4">

    <div  class="col-md-12 row">

      <div  class="col-md-4 mb-4 mb-md-0">

        <div  class="mdb-lightbox">

          <div appHighlight  class="row product-gallery mx-1">

            <div  class="col-12 mb-0">

              <div  class="view overlay rounded z-depth-1 main-img">

                <a href={{cakeData.image}} data-size="710x823">

                  <img src={{cakeData.image}}   class="img-fluid z-depth-1" style="width:100%"/>

                </a>

              </div>

            </div>

          </div>

        </div>

      </div>

      <div class="col-md-4">

        <h5>{{cakeData.name}}</h5>

        <p class="mb-2 text-muted text-uppercase small">{{cakeData.reviews}} reviews</p>

        <p><span class="mr-1"><strong>{{cakeData.price | currency:"INR"}}</strong></span></p>

        <p class="pt-1">{{cakeData.description}}</p>

        <div class="table-responsive">

          <table class="table table-sm table-borderless mb-0">

            <tbody>

              <tr>

                <th class="pl-0 w-25" scope="row"><strong>Flavour</strong></th>

                <td>{{cakeData.flavour}}</td>

              </tr>

              <tr>

                <th class="pl-0 w-25" scope="row"><strong>Ingredients</strong></th>

                <td>{{cakeData.ingredients.join()}}</td>

              </tr>

              <tr>

                <th class="pl-0 w-25" scope="row"><strong>Occasion</strong></th>

                <td>{{cakeData.type}}</td>

              </tr>

              <tr>

                <th class="pl-0 w-25" scope="row"><strong>Weight</strong></th>

                <td>{{cakeData.weight}} kg</td>

              </tr>

              <tr>

                <th class="pl-0 w-25" scope="row"><strong>Weight</strong></th>

              </tr>

            </tbody>

          </table>

        </div>

        <hr />

        <button  \*ngIf="!isAdding" (click)="addToCart()"  type="button" class="btn btn-primary btn-md mr-1 mb-2">Buy now</button>

        <!-- <button \*ngIf="!isAdding" (click)="addToCart()"  type="button" class="btn btn-light btn-md mr-1 mb-2"><i

            class="fas fa-shopping-cart pr-2"></i>Add to cart</button> -->

        <button \*ngIf="isAdding" disabled  type="button" class="btn btn-light btn-md mr-1 mb-2"><i

              class="fas fa-shopping-cart pr-2"></i>Adding to cart... Please Wait</button>

        <!-- <button type="button" class="btn btn-light btn-md mr-1 mb-2" disabled><i

            class="fas fa-shopping-cart pr-2"></i> Please wait... Adding to cart</button> -->

      </div>

    </div>

  </section></div>

**Detail-component.component.specs.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { DetailComponentComponent } from './detail-component.component';

describe('DetailComponentComponent', () => {

  let component: DetailComponentComponent;

  let fixture: ComponentFixture<DetailComponentComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ DetailComponentComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(DetailComponentComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Detail-component.component.ts**

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

import {ActivatedRoute,Router} from "@angular/router"

import {CakeService} from "src/app/service.service"

import {HttpHeaders} from "@angular/common/http"

import { ToastrService } from 'ngx-toastr';

@Component({

  selector: 'app-detail-component',

  templateUrl: './detail-component.component.html',

  styleUrls: ['./detail-component.component.css']

})

export class DetailComponent implements OnInit {

  cakeId:any;

  cakeData:any={}

  isadding:boolean=false

  isAdding:boolean=false

  constructor(private route : ActivatedRoute,private router:Router, private service :CakeService,private toaster:ToastrService) {

    this.cakeId = this.route.snapshot.params["cakeid"]

    console.log(this.cakeId)

    var url="https://apifromashu.herokuapp.com/api/cake/"+this.cakeId

    this.service.getCakeDetail(url).subscribe({

      next:(response:any)=>{

        console.log(response)

        this.cakeData=response.data

        console.log("Cake data",this.cakeData)

      },

      error:(error)=>{

        console.log("Error from cake Details api",error)

      }

    })

  }

  addToCart(){

    if(localStorage["token"]){

    this.isAdding=true

    var body :any={

        cakeid:this.cakeData.cakeid,

        name:this.cakeData.name,

        weight:this.cakeData.weight,

        price:this.cakeData.price,

        image:this.cakeData.image,

      }

    var url="https://apifromashu.herokuapp.com/api/addcaketocart"

      this.service.cake\_post(url,body).subscribe({

      next:(response:any)=>{

        console.log(response)

        this.isAdding=false

        this.router.navigate(["/cart"])

      },

      error:(error)=>{

        console.log(error)

      }

    })

    }

    else{

      this.router.navigate(["/login"])

    }

  }

  ngOnInit(): void {

  }

}

**Forget-password:**

**Forget-password.component.html**

<h1>

    Forget Password works

</h1>

**Forget-password.component.specs.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { ForgetPasswordComponent } from './forget-password.component';

describe('ForgetPasswordComponent', () => {

  let component: ForgetPasswordComponent;

  let fixture: ComponentFixture<ForgetPasswordComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ ForgetPasswordComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(ForgetPasswordComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Forget-password.component.ts**

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

@Component({

  selector: 'app-forget-password',

  templateUrl: './forget-password.component.html',

  styleUrls: ['./forget-password.component.css']

})

export class ForgetPasswordComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

}

**Home:**

[**Home.component.html**](http://Home.component.html)

**<**ngx-ui-loader></ngx-ui-loader>

<app-carousel></app-carousel>

<app-cake-list ></app-cake-list>

**Home.component.specs.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

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

describe('HomeComponent', () => {

  let component: HomeComponent;

  let fixture: ComponentFixture<HomeComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ HomeComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(HomeComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Home.component.ts**

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

import { NgxUiLoaderService } from 'ngx-ui-loader';

@Component({

  selector: 'app-home',

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

  styleUrls: ['./home.component.css']

})

export class HomeComponent implements OnInit {

  constructor(private loader : NgxUiLoaderService) { }

  ngOnInit(): void {

  }

}

**Login:**

**Login.component.css**

.login-container{

    width:40%;

    margin:1em auto;

    padding:1em;

    background-color:rgba(204, 204, 204, 0.347);

}

.form-group,button{

    margin-top:1em;

}

**Login.component.html**

.login-container{

    width:40%;

    margin:1em auto;

    padding:1em;

    background-color:rgba(204, 204, 204, 0.347);

}

.form-group,button{

    margin-top:1em;

}

**Login.component.specs.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

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

describe('LoginComponent', () => {

  let component: LoginComponent;

  let fixture: ComponentFixture<LoginComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ LoginComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(LoginComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Login.component.ts**

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

import { CakeService } from 'src/app/service.service';

import {Router} from "@angular/router"

@Component({

  selector: 'app-login',

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

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

})

export class LoginComponent implements OnInit {

  constructor(private service : CakeService,private router:Router) { }

  responseError:any

  userDetail:any={}

  login(){

    var url="https://apifromashu.herokuapp.com/api/login"

    console.log(this.userDetail)

    this.service.login(url,this.userDetail).subscribe({

      next:(response : any)=>{

          if(response.token){

              localStorage["token"]=response.token

              localStorage["loggeinUser"]=response.email

              var url="https://apifromashu.herokuapp.com/api/cakecart"

              this.service.cake\_post(url,{}).subscribe({

                next:(response : any)=>{

                  this.service.cartItems=response.data

                },

                error:(error)=>{

                  console.log("Error from getcart API",error)

                }

              })

              console.log(localStorage["token"])

              this.router.navigate([""])

          }

          else{

            this.responseError="Invalid Login"

          }

      },

      error:(err)=>{

          console.log(err)

      }

    })

  }

  ngOnInit(): void {

  }

}

**Payment:**

**Payment.component.html**

<div class="container">

    <table class="table">

        <thead>

          <tr>

            <th scope="col">#</th>

            <th scope="col">Name</th>

            <th scope="col">Image</th>

            <th scope="col">Price</th>

            <th scope="col">Quantity</th>

            <th scope="col">Total price</th>

          </tr>

        </thead>

        <tbody>

          <tr \*ngFor="let item of cartItems ; let i=index">

            <th scope="row">{{i+1}}</th>

            <td>{{item.name}}</td>

            <td><img style="height: 60px; width:60px" [src]="item.image" alt=""></td>

            <td>{{item.price}}</td>

            <td>{{item.quantity}}</td>

            <td>{{item.quantity \* item.price}}</td>

          </tr>

        </tbody>

      </table>

      <h3>Total Amount: $ {{totalAmount}}.00</h3>

      <div \*ngIf="userDetails" style="margin-top:2em;">

          <h4>Your Billing Address</h4>

          <p>Name:{{userDetails.name}}</p>

          <p>Address:{{userDetails.address}}</p>

          <p>City:{{userDetails.city}}</p>

          <p>Pincode:{{userDetails.pincode}}</p>

      </div>

      <div \*ngIf="!userDetails">

        <h4>Please add billing address </h4>

      </div>

      <div class="d-flex flex-wrap justify-content-between " style="height: 50px;">

          <button type="button" class="btn btn-primary" (click)="order()">Place Order</button>

    </div>

</div>

**Payment.component.specs.html**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { PaymentComponent } from './payment.component';

describe('PaymentComponent', () => {

  let component: PaymentComponent;

  let fixture: ComponentFixture<PaymentComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ PaymentComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(PaymentComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Payment.component.ts**

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

import { CakeService } from 'src/app/service.service';

import { ToastrService } from 'ngx-toastr';

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

@Component({

  selector: 'app-payment',

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

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

})

export class PaymentComponent implements OnInit {

  cartItems:any=[]

  userDetails:any={

  }

  totalAmount:number = 0

  body:any={}

  constructor(private service:CakeService,private router:Router,private toaster:ToastrService) {

    let url ="https://apifromashu.herokuapp.com/api/cakecart"

    this.service.cake\_post(url,{}).subscribe({

      next:(response:any)=>{

        this.cartItems=response.data

        console.log(this.cartItems)

        this.cartItems.map((\_item: any)=>{

         this.totalAmount+=\_item.quantity \* \_item.price

        })

      },

      error:(error)=>{

        console.log(error)

      }

    })

    this.userDetails=localStorage.getItem('user\_address')

    this.userDetails=JSON.parse(this.userDetails)

   }

  order(){

    let url="https://apifromashu.herokuapp.com/api/addcakeorder"

    console.log(this.cartItems)

    let body={

      cakes:this.cartItems,

      price:this.totalAmount,

      name:this.userDetails.name,

      address:this.userDetails.address,

      city:this.userDetails.city,

      pincode:this.userDetails.pincode,

      phone:this.userDetails.phone\_no

    }

    console.log(body)

    this.service.cake\_post(url,body).subscribe({

      next:(response:any) => {

        console.log("On success",response)

        this.toaster.success(`You have succesfully placed your order and your order id is ${response["order"].orderid}`)

        this.router.navigate(["/"])

      },

      error:(error)=>{

        console.log(error)

      }

    })

  }

  ngOnInit(): void {

  }

}

**Search:**

**Search.component.css**

.row{

    padding:1em;

}

**Search.component.html**

<div class="row">

    <h4>{{searchItems.length}} results found</h4>

    <div class="col-md-4"  \*ngFor="let cake of searchItems">

        <app-cake [cakedata]="cake"></app-cake>

        </div>

</div>

**Search.component.spec.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

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

describe('SearchComponent', () => {

  let component: SearchComponent;

  let fixture: ComponentFixture<SearchComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ SearchComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(SearchComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Search.component.ts**

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

import {ActivatedRoute} from "@angular/router"

import {CakeService} from "src/app/service.service"

@Component({

  selector: 'app-search',

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

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

})

export class SearchComponent implements OnInit {

  searchItems:any=[]

  constructor(private service: CakeService,private router:ActivatedRoute) {

    //  var searchText=this.router.snapshot.queryParams["q"]

    //  console.log("Search text",searchText)

    //  var url="https://apifromashu.herokuapp.com/api/searchcakes?q="+searchText

    //  this.service.search(url).subscribe({

    //    next:(response :any)=>{

    //      console.log(response)

    //      this.searchItems=response.data

    //    },

    //    error:(error)=>{

    //      console.log(error)

    //    }

    //  })

   this.router.queryParams.subscribe((query:any)=>{

     var searchText=query["q"]

     var url="https://apifromashu.herokuapp.com/api/searchcakes?q="+searchText

     this.service.search(url).subscribe({

       next:(response :any)=>{

         console.log("Response from Search API",response.data)

          this.searchItems=response.data

       },

       error:(error)=>{

         console.log("Error from Search API",error)

       }

     })

   })

   }

  ngOnInit(): void {

  }

}

**Signup:**

**Signup.component.css**

.container{

    width:50%;

    background-color:#ddd;

    padding:1em;

    margin-top:1em;

}

label{

    padding:0.5em;

}

.form-group{

    margin-top:1em;

}

.table{

    width:90%;

    margin:2em auto;

    background-color:lightgreen;

}

**Signup.component.html**

<div class="container">

    <h3>Signup</h3>

    <form [formGroup]="signupForm">

                <div \*ngIf="responseError" class="form-group mt-3">

                   <label class="alert alert-danger">{{responseError}}</label>

                </div>

                <div class="form-group mt-3">

                    <label for="exampleInputEmail1">Name</label>

                    <input formControlName="name" [(ngModel)]="userDetail.name"  class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter email">

                </div>

                <div class="form-group mt-3">

                    <label for="exampleInputEmail1">Email address</label>

                    <input formControlName="email" type="email" [(ngModel)]="userDetail.email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter email">

                    <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>

                </div>

                <div class="form-group mt-3">

                    <label for="exampleInputPassword1">Password</label>

                    <input formControlName="password" [(ngModel)]="userDetail.password" type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">

                </div>

                <button (click)="signup()" type="submit" class="btn btn-primary">Submit</button>

        </form>

</div>

**Signup.component.spec.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

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

describe('SignupComponent', () => {

  let component: SignupComponent;

  let fixture: ComponentFixture<SignupComponent>;

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      declarations: [ SignupComponent ]

    })

    .compileComponents();

  });

  beforeEach(() => {

    fixture = TestBed.createComponent(SignupComponent);

    component = fixture.componentInstance;

    fixture.detectChanges();

  });

  it('should create', () => {

    expect(component).toBeTruthy();

  });

});

**Signup.component.ts**

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

import { CakeService } from 'src/app/service.service';

import { ToastrService } from 'ngx-toastr';

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

@Component({

  selector: 'app-signup',

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

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

})

export class SignupComponent implements OnInit {

  userDetail:any={}

  responseError:any

  signupForm:any

  users: any=[];

  constructor(private service : CakeService,private toaster:ToastrService,private formbuilder:FormBuilder) {

      this.signupForm=this.formbuilder.group({

        email:["",[Validators.required,Validators.email ]],

        name:["",[Validators.required]],

        password:["",[Validators.required]],

        pincode:["",[Validators.required,Validators.maxLength(6),Validators.minLength(6)]],

      })

  }

  signup(){

    if(this.signupForm.valid){

      this.toaster.success("Your form is valid")

    }

  }

  addUser(){

    console.log(this.userDetail)

    var url='https://apifromashu.herokuapp.com/api/register'

    this.service.post(url,this.userDetail).subscribe({

      next:(response:any)=>{

        console.log(response.message)

        this.responseError=response.message

      },

      error:(error)=>{

        console.log("Error",error)

      }

    })

    var user={...this.userDetail}

    this.users.push(user)

    // console.log(this.users)

  }

  deleteUser(email:String){

    this.users=this.users.filter( (user: { email: String; }) =>user.email!==email)

  }

**Approuting.module.ts**

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

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

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

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

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

import {DetailComponent} from "./components/detail-component/detail-component.component"

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

import { CartComponent } from './components/cart/cart.component';

import {CheckoutComponent} from "./components/checkout/checkout.component";

import { AddressComponent } from './components/address/address.component';

import { PaymentComponent } from './components/payment/payment.component';

import { AddcakeComponent } from './components/addcake/addcake.component';

import {PreviousOrdersComponent} from './components/previous-orders/previous-orders.component'

import { PreviousCakesDetailsComponent } from './components/previous-cakes-details/previous-cakes-details.component';

import { AuthGuard } from './auth.guard';

import {UserGuard} from './user.guard';

const routes: Routes = [

 {path:"login",component:LoginComponent,canActivate:[AuthGuard] },

 {path:"",component:HomeComponent},

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

 {path:"search",component:SearchComponent},

 {path:"detail/:cakeid",component:DetailComponent},

 {path:"addcake",component:AddcakeComponent},

 {path:"previous-orders",component:PreviousOrdersComponent,canActivate:[UserGuard]},

 {path:"previous-cakes/:id",component:PreviousCakesDetailsComponent,canActivate:[UserGuard]},

 //Nested Components

 {path:"checkout",component:CheckoutComponent,

 children:[

  {path:"", component:AddressComponent},

 {path:"address" ,component:AddressComponent},

 {path:"payment",component:PaymentComponent}

 ]},

 {path:"cart",component:CartComponent,canActivate:[UserGuard]}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

**app.component.html**

<app-navabar></app-navabar>

<router-outlet></router-outlet>

<ngx-ui-loader></ngx-ui-loader>

**App.component.specs.ts**

import { TestBed } from '@angular/core/testing';

import { RouterTestingModule } from '@angular/router/testing';

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

describe('AppComponent', () => {

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      imports: [

        RouterTestingModule

      ],

      declarations: [

        AppComponent

      ],

    }).compileComponents();

  });

  it('should create the app', () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app).toBeTruthy();

  });

  it(`should have as title 'Angular-app-1'`, () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app.title).toEqual('Angular-app-1');

  });

  it('should render title', () => {

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.nativeElement as HTMLElement;

    expect(compiled.querySelector('.content span')?.textContent).toContain('Angular-app-1 app is running!');

  });

});

**App.component.ts**

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

import { CakeService } from './service.service';

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  constructor(private service:CakeService){

  }

}

**App.module.ts**

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

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

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

import {ToastrModule} from 'ngx-toastr'

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

import { NgxUiLoaderModule} from "ngx-ui-loader"

import {HttpClientModule} from "@angular/common/http"

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

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

import { CarouselComponent } from './components/carousel/carousel.component';

import { CakeComponent } from './components/cake/cake.component';

import { CakeListComponent } from './components/cake-list/cake-list.component';

import { NavabarComponent } from './components/navabar/navabar.component';

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

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

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

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

import { DetailComponent } from './components/detail-component/detail-component.component';

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

import { CartComponent } from './components/cart/cart.component';

import { CheckoutComponent } from './components/checkout/checkout.component';

import { AddressComponent } from './components/address/address.component';

import { PaymentComponent } from './components/payment/payment.component';

import { DiscountPipe } from './discount.pipe';

import { AddcakeComponent } from './components/addcake/addcake.component';

import {FontAwesomeModule} from "@fortawesome/angular-fontawesome";

import { ForgetPasswordComponent } from './components/forget-password/forget-password.component';

import { PreviousOrdersComponent } from './components/previous-orders/previous-orders.component';

import { PreviousCakesDetailsComponent } from './components/previous-cakes-details/previous-cakes-details.component'

//Guard

import { AuthGuard } from './auth.guard';

import {UserGuard} from './user.guard';

@NgModule({

  declarations: [

    AppComponent,

    CarouselComponent,

    CakeComponent,

    CakeListComponent,

    NavabarComponent,

    SignupComponent,

    LoginComponent,

    HomeComponent,

    DetailComponent,

    SearchComponent,

    CartComponent,

    CheckoutComponent,

    AddressComponent,

    PaymentComponent,

    DiscountPipe,

    AddcakeComponent,

    ForgetPasswordComponent,

    PreviousOrdersComponent,

    PreviousCakesDetailsComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    FormsModule,

    ReactiveFormsModule,

    BrowserAnimationsModule,

    HttpClientModule,

    NgxUiLoaderModule,

    ToastrModule.forRoot(),

    FontAwesomeModule

  ],

  providers: [AuthGuard,UserGuard],

  bootstrap: [AppComponent]

})

export class AppModule { }

**auth.guard.spec.ts**

import { TestBed } from '@angular/core/testing';

import { AuthGuard } from './auth.guard';

describe('AuthGuard', () => {

  let guard: AuthGuard;

  beforeEach(() => {

    TestBed.configureTestingModule({});

    guard = TestBed.inject(AuthGuard);

  });

  it('should be created', () => {

    expect(guard).toBeTruthy();

  });

});

**Auth.guaqrd.ts**

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

import { ActivatedRouteSnapshot, CanActivate, RouterStateSnapshot, UrlTree } from '@angular/router';

import { Observable } from 'rxjs';

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

@Injectable({

  providedIn: 'root'

})

export class AuthGuard implements CanActivate {

  constructor(private router: Router){

  }

  canActivate(){

    if(localStorage['token']){

        this.router.navigate([""])

        return false

    }

    return true

  }

}

**Cake.ts**

export class Cake{

    name!: string

    price!: number

    image!: string

    cakeid!:string

    special!:boolean

}

export class UserInfo{

    username!: String

    email!: String

    password!:String

}

**Discount.pipe.specs.ts**

import { DiscountPipe } from './discount.pipe';

describe('DiscountPipe', () => {

  it('create an instance', () => {

    const pipe = new DiscountPipe();

    expect(pipe).toBeTruthy();

  });

});

**Discount.pipe.ts**

import { Pipe, PipeTransform } from '@angular/core';

@Pipe({

  name: 'discount'

})

export class DiscountPipe implements PipeTransform {

  transform(value: unknown, ...args: unknown[]): unknown {

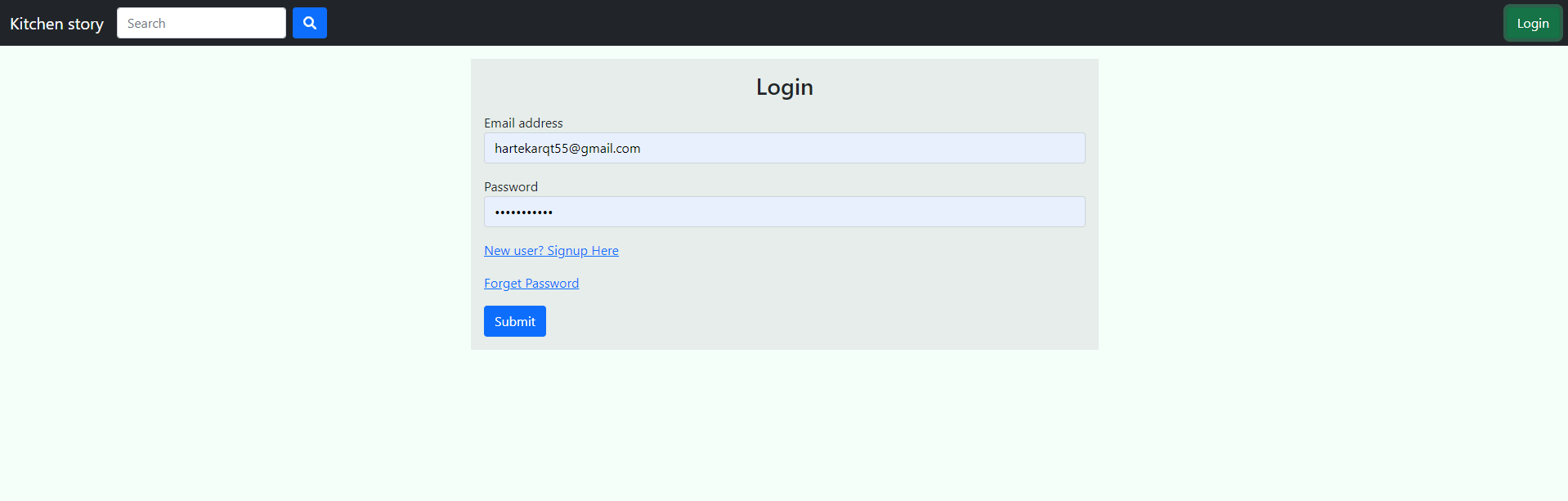
    return null;

  }

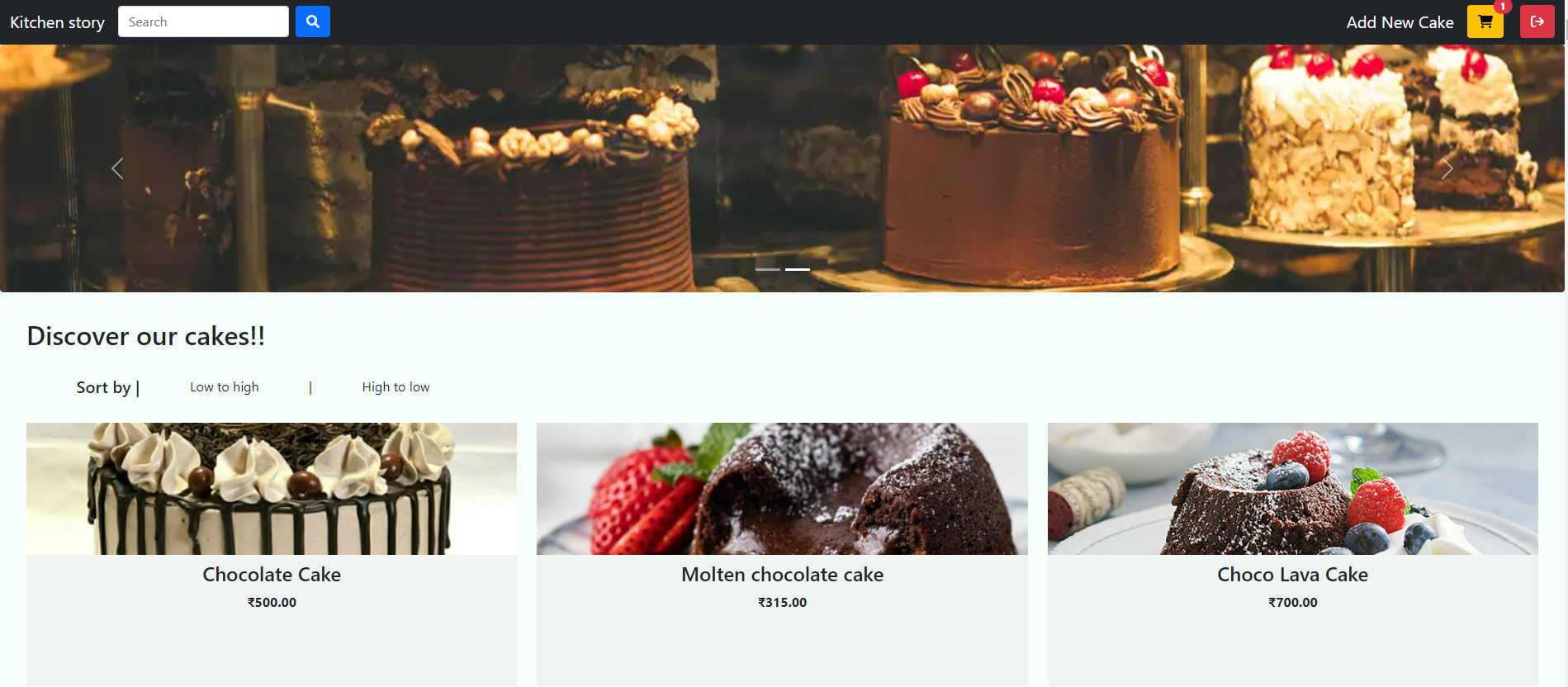
}

**SCREENSHOTS.**

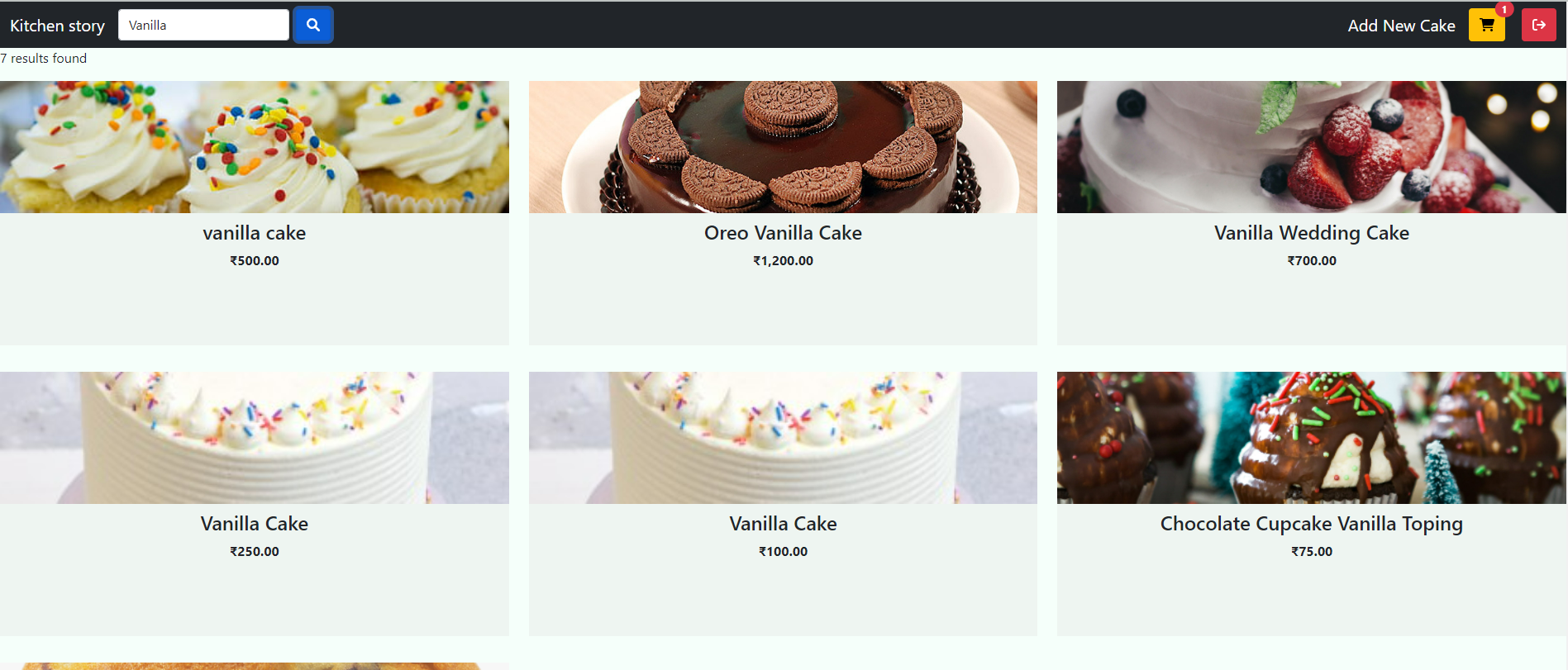
**1.Login**



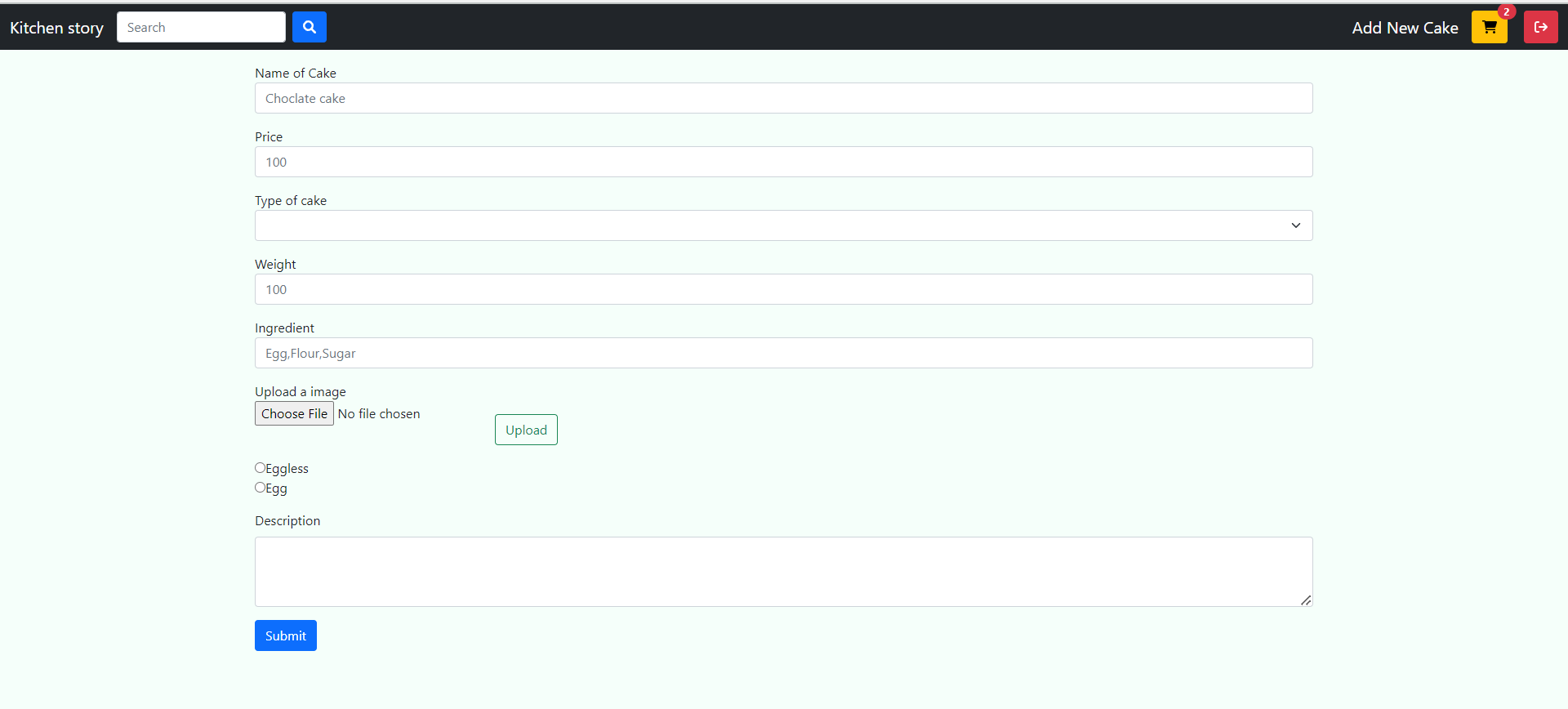
**2.home**



**3.search**



**4.addcake**



**5.checkout**

