

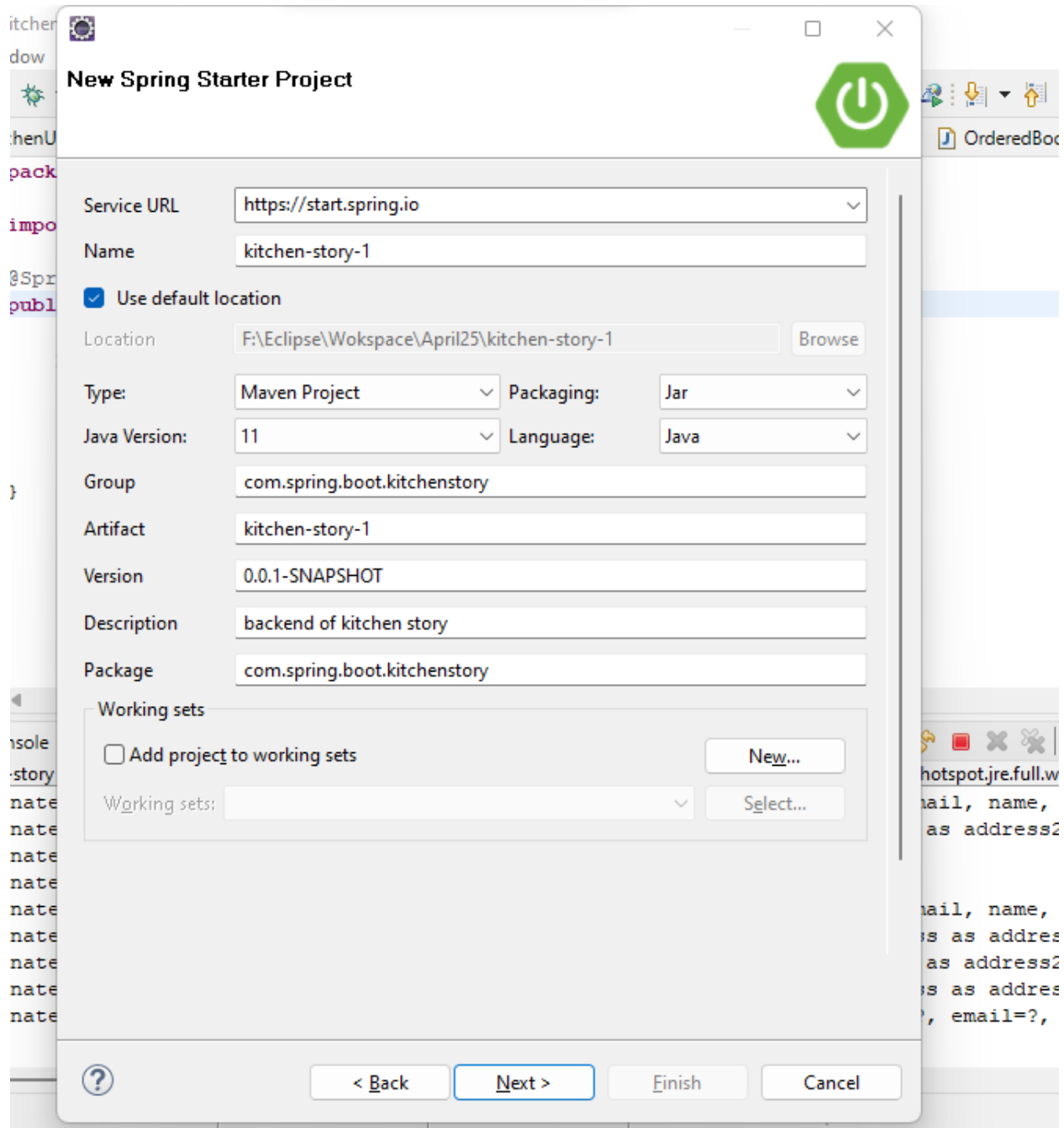
## Project –

### Develop a Web Application using Front-end (Phase- 5)

#### Source Code Kitchen Story-

→Back-end code-

Create SpringBoot project using spring Starter project-



The screenshot shows the 'New Spring Starter Project' dialog box in the Eclipse IDE. The dialog is titled 'New Spring Starter Project' and features a green power button icon in the top right corner. The fields are filled with the following information:

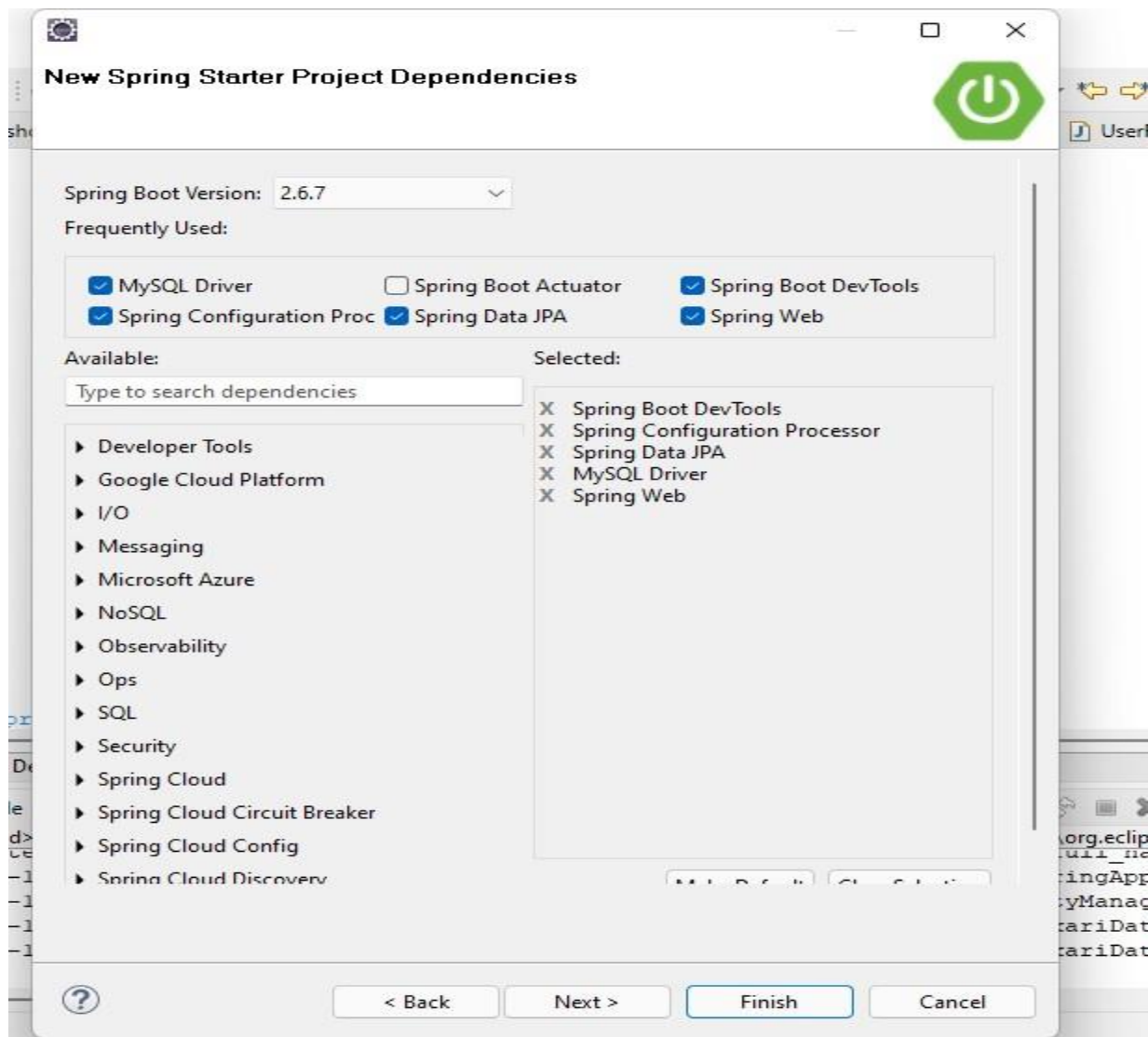
- Service URL: `https://start.spring.io`
- Name: `kitchen-story-1`
- ☒ Use default location
- Location: `F:\Eclipse\Wokspace\April25\kitchen-story-1` (with a 'Browse' button)
- Type: `Maven Project` (dropdown)
- Packaging: `Jar` (dropdown)
- Java Version: `11` (dropdown)
- Language: `Java` (dropdown)
- Group: `com.spring.boot.kitchenstory`
- Artifact: `kitchen-story-1`
- Version: `0.0.1-SNAPSHOT`
- Description: `backend of kitchen story`
- Package: `com.spring.boot.kitchenstory`

Below these fields is the 'Working sets' section, which includes:

- ☐ Add project to working sets (with a 'New...' button)
- Working sets: (dropdown menu) (with a 'Select...' button)

At the bottom of the dialog, there are four buttons: a help icon (?), '< Back', 'Next >', 'Finish', and 'Cancel'.

And after spring.io open enter the dependencies and spring project will we created.



After the finish button click, spring boot project is created.

Now setup the pom.xml in the spring boot project- **spring-boot-kitchenstory**.

**Pom.xml file-**

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.7.0</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.spring.boot.kitchenstory</groupId>
<artifactId>kitchen-story</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>kitchen-story</name>
<description>backend of kitchen story</description>
<properties>
  <java.version>17</java.version>
</properties>
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-configuration-processor</artifactId>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

```

    <build>
      <plugins>
        <plugin>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
      </plugins>
    </build>
  </project>

```

### Add dependency –

**spring-boot-starter-data-jpa** – for jpa repository use and for basic crud operation.

**spring-boot-starter-web**—for making web project **mysql-connector-java**—for use of mysql database **spring-boot-configuration-processor** –basic tools for configuration **spring-boot-devtools** –tool for devtools

→Now add **application.properties** in the **src/main/resources**

```

spring.mvc.pathmatch.matching-strategy=ANT_PATH_MATCHER

#database configuration: mysql
spring.datasource.url=jdbc:mysql://localhost:3306/fun
spring.datasource.username=root
spring.datasource.password=Chemistry22@
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

#Hibernate Configuration
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL57Dialect

```

**Database used-** fun

**Driver class for KitchenStory Code-**

**Path-**src/main/java

**Package-**com.spring.boot.sportyshoes

### Class-KitchenStoryApplication.java

```
package com.spring.boot.kitchenstory;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class KitchenStoryApplication {

    public static void main(String[] args) {
        SpringApplication.run(KitchenStoryApplication.class, args);
        System.out.println("Kitchen story");
    }

}
```

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.entities

**Class-** KitchenUserRegistered.java

**Table create in DB-**kitchen\_user\_registered

```
package com.spring.boot.kitchenstory.entities;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="KitchenUserRegistered")
public class KitchenUserRegistered {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    private String email;
    private String fullName;
    private String password;
    private String address;
    private String cardType;
    private int cardNumber;
}
```

```
private String type;
public int getId() {
    return id;
}
public void setId(int id) {
    this.id = id;
}
public String getEmail() {
    return email;
}
public void setEmail(String email) {
    this.email = email;
}
public String getFullName() {
    return fullName;
}
public void setFullName(String fullName) {
    this.fullName = fullName;
}
public String getPassword() {
    return password;
}
public void setPassword(String password) {
    this.password = password;
}
public String getAddress() {
    return address;
}
public void setAddress(String address) {
    this.address = address;
}
public String getCardType() {
    return cardType;
}
public void setCardType(String cardType) {
    this.cardType = cardType;
}
public int getCardNumber() {
    return cardNumber;
}
public void setCardNumber(int cardNumber) {
    this.cardNumber = cardNumber;
}
public String getType() {
    return type;
}
```

```

    }
    public void setType(String type) {
        this.type = type;
    }
    public KitchenUserRegistered() {
        super();
    }
    public KitchenUserRegistered(int id, String email, String fullName, String
password, String address,
        String cardType, int cardNumber, String type) {
        super();
        this.id = id;
        this.email = email;
        this.fullName = fullName;
        this.password = password;
        this.address = address;
        this.cardType = cardType;
        this.cardNumber = cardNumber;
        this.type = type;
    }
    public KitchenUserRegistered(String email, String fullName, String password,
String address) {
        super();
        this.email = email;
        this.fullName = fullName;
        this.password = password;
        this.address = address;
    }
}
}

```

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.entities

**Class-** KitchenProduct.java

**Table create in DB-**kitchen\_product

```
package com.spring.boot.kitchenstory.entities;
```

```
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="Kitchen_Product")
public class KitchenProduct {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int prodId;

    @Column
    private String prodName;

    @Column
    private String vendorName;

    @Column
    private double price;

    @Column
    private String imgUrl;

    @Column
    private String prodDetail;

    @Column
    private String type;

    public String getType() {
        return type;
    }

    public void setType(String type) {
        this.type = type;
    }

    public int getProdId() {
        return prodId;
    }
}
```



```
public void setProdId(int prodId) {
    this.prodId = prodId;
}

public String getProdName() {
    return prodName;
}

public void setProdName(String prodName) {
    this.prodName = prodName;
}

public String getVendorName() {
    return vendorName;
}

public void setVendorName(String vendorName) {
    this.vendorName = vendorName;
}

public double getPrice() {
    return price;
}

public void setPrice(double price) {
    this.price = price;
}

public String getImgUrl() {
    return imgUrl;
}

public void setImgUrl(String imgUrl) {
    this.imgUrl = imgUrl;
}

public String getProdDetail() {
    return prodDetail;
}

public void setProdDetail(String prodDetail) {
    this.prodDetail = prodDetail;
}
```

```

    public KitchenProduct(int prodId, String prodName, String vendorName, double
price, String imgUrl,
        String prodDetail, String type) {
        super();
        this.prodId = prodId;
        this.prodName = prodName;
        this.vendorName = vendorName;
        this.price = price;
        this.imgUrl = imgUrl;
        this.prodDetail = prodDetail;
        this.type=type;
    }

    public KitchenProduct() {
        super();
    }

}

```

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.entities

**Class-** KitchenOrderBooked.java

**Table create in DB-**kitchen\_order\_booked

```

package com.spring.boot.kitchenstory.entities;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="KitchenOrderBooked")
public class KitchenOrderBooked {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column
    private int orderId;
}

```

```
@Column
private String name;

@Column
private String country;

@Column
private String address;

@Column
private int postcode;

@Column
private String email;

@Column
private long phone;

@Column
private String productOrdered;

@Column
private double total;

@Column
private long cardNumber;

public int getOrderId() {
    return orderId;
}

public void setOrderId(int orderId) {
    this.orderId = orderId;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}
```

```
public String getCountry() {  
    return country;  
}  
  
public void setCountry(String country) {  
    this.country = country;  
}  
  
public String getAddress() {  
    return address;  
}  
  
public void setAddress(String address) {  
    this.address = address;  
}  
  
public int getPostcode() {  
    return postcode;  
}  
  
public void setPostcode(int postcode) {  
    this.postcode = postcode;  
}  
  
public String getEmail() {  
    return email;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
  
public long getPhone() {  
    return phone;  
}  
  
public void setPhone(long phone) {  
    this.phone = phone;  
}  
  
public String getProductOrdered() {  
    return productOrdered;  
}  
  
public void setProductOrdered(String productOrdered) {
```

```
        this.productOrdered = productOrdered;
    }

    public double getTotal() {
        return total;
    }

    public void setTotal(double total) {
        this.total = total;
    }

    public long getCardNumber() {
        return cardNumber;
    }

    public void setCardNumber(long cardNumber) {
        this.cardNumber = cardNumber;
    }

    public KitchenOrderBooked(String name, String country, String address, int
postcode, String email, long phone,
        String productOrdered, double total, long cardNumber) {
        super();
        this.name = name;
        this.country = country;
        this.address = address;
        this.postcode = postcode;
        this.email = email;
        this.phone = phone;
        this.productOrdered = productOrdered;
        this.total = total;
        this.cardNumber = cardNumber;
    }

    public KitchenOrderBooked() {
        super();
    }
}
```

## DAO interface for spring boot project-

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.repositories

**Interface-** KitchenUserDao.java

```
package com.spring.boot.kitchenstory.repositories;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.spring.boot.kitchenstory.entities.KitchenUserRegistered;

@Repository
public interface KitchenUserDao extends JpaRepository<KitchenUserRegistered, Integer> {
    //all basic CRUD operation
}
```

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.repositories

**Interface-** KitchenProductDao.java

```
package com.spring.boot.kitchenstory.repositories;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.spring.boot.kitchenstory.entities.KitchenProduct;

@Repository
public interface KitchenProductDao extends JpaRepository<KitchenProduct, Integer> {
    //CRUD operation for products
}
```

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.repositories

**Interface-** KitchenOrderBookedDao.java

```

package com.spring.boot.kitchenstory.repositories;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.spring.boot.kitchenstory.entities.KitchenOrderBooked;

@Repository
public interface KitchenOrderBookedDao extends JpaRepository<KitchenOrderBooked, Integer> {
    //all basic method of the Crud operation
}

```

**Interface services will be here to work after Dao.**

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.services

**Interface-**KitchenUserService.java

```

package com.spring.boot.kitchenstory.services;

import java.util.List;

import com.spring.boot.kitchenstory.entities.KitchenUserRegistered;

public interface KitchenUserService {

    public boolean addUserSignUp(KitchenUserRegistered user);

    public List<KitchenUserRegistered> getAllUser();

    public KitchenUserRegistered getUser(int user_id);

    public KitchenUserRegistered updateUserPassword(int user_id, KitchenUserRegistered user);

}

```

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.services

**Interface-**KitchenProductService.java

```

package com.spring.boot.kitchenstory.services;

import java.util.List;

import com.spring.boot.kitchenstory.entities.KitchenProduct;

public interface KitchenProductService {

    //add the product
    public boolean addProduct(KitchenProduct product);

    //show all LIST of PProduct
    public List<KitchenProduct> getAllProducts();

    //get product by id
    public KitchenProduct getProductById(int productId);

    //update product by using prodId
    public KitchenProduct updateProductById(KitchenProduct product, int prodId);

    //delete the product by prodId
    public boolean deleteProductById(int prodId);
}

```

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.services

**Interface-** KitchenOrderBookedService.java

```

package com.spring.boot.kitchenstory.services;
import java.util.List;

import com.spring.boot.kitchenstory.entities.KitchenOrderBooked;

public interface KitchenOrderBookedService {
    //to add the order booked

    public KitchenOrderBooked addOrder(KitchenOrderBooked order);

    //to get one order
    public KitchenOrderBooked getOneOrder(int orderId);

    //to update Order by id
    public KitchenOrderBooked updateOrder(int orderId,KitchenOrderBooked order);
}

```



```

    //getAll order list
    public List<KitchenOrderBooked> getAllOrder();
}

```

**After the service interfaces, the services are implemented using the another class servicemplementation.**

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.servicesImpl

**Class-** KitchenUserServiceImpl.java

```

package com.spring.boot.kitchenstory.servicesImpl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.spring.boot.kitchenstory.entities.KitchenUserRegistered;
import com.spring.boot.kitchenstory.repositories.KitchenUserDao;
import com.spring.boot.kitchenstory.services.KitchenUserService;

@Service
public class KitchenUserServiceImpl implements KitchenUserService {

    @Autowired
    private KitchenUserDao userDao;

    //addd the USer signUp
    @Override
    public boolean addUserSignUp(KitchenUserRegistered user) {
        boolean flag=false;
        userDao.save(user);
        flag=true;
        return flag;
    }

    @Override
    public List<KitchenUserRegistered> getAllUser() {
        return userDao.findAll();
    }
}

```

```

@Override
public KitchenUserRegistered getUser(int user_id) {
    return userDao.findById(user_id).get();
}

@Override
public KitchenUserRegistered updateUserPassword(int user_id,
KitchenUserRegistered user) {
    KitchenUserRegistered user1=userDao.findById(user_id).get();
    user1.setPassword(user.getPassword());
    user1.setFullName(user.getFullName());
    user1.setEmail(user.getEmail());
    user1.setAddress(user.getAddress());
    return userDao.save(user1);
}
}

```

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.servicesImpl

**Class-** KitchenProductServiceImpl.java

```

package com.spring.boot.kitchenstory.servicesImpl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.spring.boot.kitchenstory.entities.KitchenProduct;
import com.spring.boot.kitchenstory.repositories.KitchenProductDao;
import com.spring.boot.kitchenstory.services.KitchenProductService;

@Service
public class KitchenProductServiceImpl implements KitchenProductService {

    @Autowired
    private KitchenProductDao productDao;

    //add the product
    @Override
    public boolean addProduct(KitchenProduct product) {

```

```

        productDao.save(product);
        return true;
    }

    //LIST of allProducts
    @Override
    public List<KitchenProduct> getAllProducts() {
        return productDao.findAll();
    }

    //get Product by Id
    @Override
    public KitchenProduct getProductById(int productId) {
        return productDao.findById(productId).get();
    }

    //Update the product using prodId
    @Override
    public KitchenProduct updateProductById(KitchenProduct product, int prodId) {
        KitchenProduct pro;
        pro=new
KitchenProduct(prodId,product.getProdName(),product.getVendorName(),product.getPr
ice(),product.getImgUrl(),product.getProdDetail(),product.getType());
        productDao.save(pro);
        return pro;
    }

    @Override
    public boolean deleteProductById(int prodId) {
        boolean flag=false;
        KitchenProduct prod=productDao.findById(prodId).get();
        productDao.delete(prod);
        flag=true;
        return flag;
    }
}

```

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.servicesImpl

**Class-** KitchenOrderBookedServiceImpl.java

```
package com.spring.boot.kitchenstory.servicesImpl;
```

```

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.spring.boot.kitchenstory.entities.KitchenOrderBooked;
import com.spring.boot.kitchenstory.repositories.KitchenOrderBookedDao;
import com.spring.boot.kitchenstory.services.KitchenOrderBookedService;

@Service
public class KitchenOrderBookedServiceImpl implements KitchenOrderBookedService {

    @Autowired
    private KitchenOrderBookedDao orderDao;

    //adding the order in KitchenOrderServiceImpl
    @Override
    public KitchenOrderBooked addOrder(KitchenOrderBooked order) {
        //KitchenOrderBooked realOrder=new KitchenOrderBooked(order.getName(),
        order.getCompanyName(), order.getCountry(), order.getAddress(),
        order.getPostcode(), order.getEmail(), order.getPhone());
        orderDao.save(order);
        return order;
    }

    //to getone Order by id
    @Override
    public KitchenOrderBooked getOneOrder(int orderId) {
        return orderDao.findById(orderId).get();
    }

    //to update order by id
    @Override
    public KitchenOrderBooked updateOrder(int orderId, KitchenOrderBooked order)
{
        KitchenOrderBooked newOrder=orderDao.findById(orderId).get();
        newOrder.setCardNumber(order.getCardNumber());
        return orderDao.save(newOrder);
    }

    @Override
    public List<KitchenOrderBooked> getAllOrder() {
        // TODO Auto-generated method stub

```

```

        return orderDao.findAll();
    }
}

```

After service implementations are completed, all these services will be called by Controller of Admin using Angular project get,post,put url in service.ts file. So to establish the connection between these cross server app. Write `@CrossOrigin("*")` to make it work in Angular project.

**Path-** src/main/java

**Package-** com.spring.boot.kitchenstory.controllers

**Class-** AdminController.java

```

package com.spring.boot.kitchenstory.controllers;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

import com.spring.boot.kitchenstory.entities.KitchenOrderBooked;
import com.spring.boot.kitchenstory.entities.KitchenProduct;
import com.spring.boot.kitchenstory.entities.KitchenUserRegistered;
import com.spring.boot.kitchenstory.servicesImpl.KitchenOrderBookedServiceImpl;
import com.spring.boot.kitchenstory.servicesImpl.KitchenProductServiceImpl;
import com.spring.boot.kitchenstory.servicesImpl.KitchenUserServiceImpl;

@RestController
@CrossOrigin("*")
public class AdminController {

    @Autowired
    private KitchenUserServiceImpl userServiceImpl;
    @Autowired

```

```

private KitchenProductServiceImpl productServiceImpl;
@Autowired
private KitchenOrderBookedServiceImpl orderserviceImpl;

//add the sign up user
@PostMapping("/admin/addedSignUpUser")
public String addSignUser(@RequestBody KitchenUserRegistered user)
{
    if(userServiceImpl.addUserSignUp(user))
        return "User added successfully";
    else
        return "some error occurred";
}

//get all the Signed up user
@GetMapping("/admin/showAllUser")
public List<KitchenUserRegistered> getAllUser(){
    return userServiceImpl.getAllUser();
}

//get user by id
@GetMapping("/admin/changePassword/{user_id}")
public KitchenUserRegistered getUser(@PathVariable int user_id) {
    return userServiceImpl.getUser(user_id);
}

//update the password
@PutMapping("/admin/updatePassword/{user_id}")
public KitchenUserRegistered updateUserPassword(@PathVariable int
user_id,@RequestBody KitchenUserRegistered user) {
    return userServiceImpl.updateUserPassword(user_id, user) ;
}

//Add the product details to API
@PostMapping("/admin/addProductDetails")
public KitchenProduct addTheProduct(@RequestBody KitchenProduct product)
{
    if(productServiceImpl.addProduct(product))
        return product;
    else
        return null;
}

//get Productby ID
@GetMapping("/admin/product/{productID}")

```

```

public KitchenProduct getProductById(@PathVariable int productId) {
    return productServiceImpl.getProductById(productId);
}

//get all the Products
@GetMapping("/admin/showProducts")
public List<KitchenProduct> getAllProducts(){
    return productServiceImpl.getAllProducts();
}

//update the product by productId
@PutMapping("/admin/upadte/{productId}")
public KitchenProduct updateProduct(@RequestBody KitchenProduct
product,@PathVariable int productId) {
    return productServiceImpl.updateProductById(product, productId);
}

//Delete a product
@DeleteMapping("/admin/deleteProduct/{productId}")
public String deleteProduct(@PathVariable int productId) {
    if(productServiceImpl.deleteProductById(productId))
        return "product is deleted";
    else
        return "not deleted";
}

//Order booked
@PostMapping("/checkout/orderBooked")
public KitchenOrderBooked bookOrder(@RequestBody KitchenOrderBooked order) {
    return orderserviceImpl.addOrder(order);
}

//get OrderId
@GetMapping("/checkout/{orderId}")
public KitchenOrderBooked getOneOrder(@PathVariable int orderId) {
    return orderserviceImpl.getOneOrder(orderId);
}

@PutMapping("/checkout/{orderId}")
public KitchenOrderBooked updateOrderById(@PathVariable int
orderId,@RequestBody KitchenOrderBooked order)
{
    return orderserviceImpl.updateOrder(orderId, order);
}

```

```

    }

    @GetMapping("/allOrder")
    public List<KitchenOrderBooked> getAllOrder(){
        return orderserviceImpl.getAllOrder();
    }
}

```

## →Front-end code-

Creation of angular project : **KitchenStory** using node.js in command prompt.

```

Build at: 2022-06-15T22:37:10.138Z - Hash: 160eae95a3a55d63 - Time: 309ms
✓ Compiled successfully.

Session contents restored from 6/16/2022 at 4:26:50 AM

Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

F:\VsCode\19May>ng new KitchenStory[]

```

All the folder that are created after the project creation.

```

└─ KitchenStory-e-commerce \KitchenStoryFrontend
   └─ .angular\cache\13.3.7
      ├── angular-webpack
      ├── babel-webpack
      └─ .vscode
   ├── node_modules
   └─ src
      ├── app
      ├── assets
      ├── environments
      ├── ★ favicon.ico
      ├── <> index.html
      ├── TS main.ts
      ├── TS polyfills.ts
      ├── # styles.css
      ├── TS test.ts
      ├── .browserslistrc
      ├── .editorconfig
      ├── .gitignore
      ├── {} angular.json
      ├── K karma.conf.js
      ├── {} package-lock.json
      ├── {} package.json
      ├── i README.md
      └─ {} tsconfig.app.json

```



Add Bootstrap in the **index.html**

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>KitchenStoryFrontend</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <!-- bootstrap 5-->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD3yD65VohhpUuCOmLASjC"
crossorigin="anonymous">

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
<!--font awesome-->

</head>
<body>
  <app-root></app-root>
</body>
</html>
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app

**File name:** app.component.html

**File type:** Html

```
<div class="conatiner-fluid">
  <nav class="navbar relative-top navbar-expand-sm navbar-dark ">

    <a routerLink="/home" class="navbar-brand mb-0 h1">
      
      Kitchen Story
    </a>
```

```

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

<div class="collapse navbar-collapse" id="navbarNav">
  <ul class="navbar-nav">
    <li class="nav-item active">
      <a routerLink="/home" class="nav-link active">
        Home
      </a>
    </li>

    <li class="nav-item active">
      <a routerLink="/navbar" class="nav-link">
        About
      </a>
    </li>
    <li class="nav-item active">
      <a routerLink="/signIn" class="nav-link">
        SignIn
      </a>
    </li>
  </ul>
</div>
<!-- Search bar -->
<div class="form-group">
  <input type="text" class="form-control me-2" [(ngModel)]="searchTerm"
(keyup)="search($event)" placeholder="Search Vegetable">

</div>

<span class="fa fa-search search-icon"></span>
<button routerLink="cart" class="btn btn-black shop" >
  <i style="font-size:30px;" class="fa fa-shopping-cart"></i>
  <!-- for cart order number -->
  <div style="font-size:9px;" class="badge bg-
primary ">{{totalItem}}</div>

```

```

        </button>

    </nav>
<!-- nav br completed -->
<router-outlet></router-outlet>
<!-- <button type="button" class="btn btn-dark">Warning</button> -->
<!-- <a routerLink="/navbar">Home</a> -->
</div>

<!-- dashboard coing -->

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app

**File name:** app.component.css

**File type:** Css

```

.navbar{
    background-color:black;
}

.form-control{
    border-radius: 3px;
    width:100%;
    margin-top: 0px;
    position: relative;
    margin-right:1% ;
}

.search-icon{
    position: relative;

    margin-right: .1%;
    color:burlywood;
}

.shop{
    position: relative;

    margin-right: 0%;
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app

**File name:** app.component.ts

**File type:**typescript

```
import { Component } from '@angular/core';
import { OnInit } from '@angular/core';
import { CartService } from '../service/cart.service';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  title = 'KitchenStoryFrontend';
  public totalItem : number;
  public searchTerm:string='';

  constructor(private cartService:CartService) { }

  ngOnInit(): void {
    this.cartService.getProducts().subscribe(res=>{
      this.totalItem=res.length;
    });
  }

  search(event:any){

    this.searchTerm=(event.target as HTMLInputElement).value;
    console.log(this.searchTerm);
    this.cartService.search.next(this.searchTerm);
  }
}
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app

**File name:** app.module.ts

**File type:**typescript

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

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

```
import { AppComponent } from './app.component';
import { NavbarComponent } from './home/navbar/navbar.component';
import { HomeComponent } from './home/home/home.component';
import { SignInComponent } from './home/sign-in/sign-in.component';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { RegisterComponent } from './home/register/register.component';
import { NgxImageZoomModule } from 'ngx-image-zoom';
import { AddProductComponent } from './home/add-product/add-product.component';
import { HomeAdminComponent } from './home/home-admin/home-admin.component';
import { AdminViewUpdateProductComponent } from './home/admin-view-update-product/admin-view-update-product.component';
import { UpdateProductComponent } from './home/update-product/update-product.component';
import { ViewProductComponent } from './home/view-product/view-product.component';
import { ShopCartComponent } from './home/shop-cart/shop-cart.component';
import { PipePipe } from './search/pipe.pipe';
import { RouterModule } from '@angular/router';
import { CheckoutComponent } from './checkout/checkout.component';
import { ChangePasswordComponent } from './home/change-password/change-password.component';
import { AllUserComponent } from './home/all-user/all-user.component';
import { ViewuserComponent } from './home/viewuser/viewuser.component';
import { PayoutComponent } from './home/payout/payout.component';
import { OrderConfirmationComponent } from './home/order-confirmation/order-confirmation.component';
```

```
@NgModule({
  declarations: [
    AppComponent,
    NavbarComponent,
    HomeComponent,
    SignInComponent,
    RegisterComponent,
    AddProductComponent,
    HomeAdminComponent,
    AdminViewUpdateProductComponent,
    UpdateProductComponent,
    ViewProductComponent,
    ShopCartComponent,
    PipePipe,
    CheckoutComponent,
    ChangePasswordComponent,
    AllUserComponent,
```

```

        ViewuserComponent,
        PayoutComponent,
        OrderConfirmationComponent
    ],
    imports: [
        BrowserModule,
        AppRoutingModule,
        HttpClientModule,
        FormsModule,
        NgxImageZoomModule
    ],
    providers: [],
    bootstrap: [AppComponent]
  })
  export class AppModule { }

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app

**File name:**app-routing.module.ts

**File type:**typescript

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { CheckoutComponent } from '../checkout/checkout.component';

import { AddProductComponent } from '../home/add-product/add-product.component';
import { AdminViewUpdateProductComponent } from '../home/admin-view-update-product/admin-view-update-product.component';
import { AllUserComponent } from '../home/all-user/all-user.component';
import { ChangePasswordComponent } from '../home/change-password/change-password.component';
import { HomeAdminComponent } from '../home/home-admin/home-admin.component';
import { HomeComponent } from '../home/home/home.component';
import { NavbarComponent } from '../home/navbar/navbar.component';
import { OrderConfirmationComponent } from '../home/order-confirmation/order-confirmation.component';
import { PayoutComponent } from '../home/payout/payout.component';
import { RegisterComponent } from '../home/register/register.component';
import { ShopCartComponent } from '../home/shop-cart/shop-cart.component';
import { SignInComponent } from '../home/sign-in/sign-in.component';
import { UpdateProductComponent } from '../home/update-product/update-product.component';
import { ViewProductComponent } from '../home/view-product/view-product.component';


```

```
import { ViewuserComponent } from './home/viewuser/viewuser.component';

const routes: Routes = [
  {path: '', redirectTo: 'home', pathMatch: 'full'},
  {path: 'home' , component: HomeComponent},
  {path: "navbar", component:NavbarComponent},
  {path: "signIn", component:SignInComponent},
  {path: "register", component:RegisterComponent},
  {path: "addProduct", component:AddProductComponent},
  {path: "home-admin", component:HomeAdminComponent},
  {path: "adminViewUpdateProduct", component:AdminViewUpdateProductComponent},
  {path: "updateProduct/:productId", component:UpdateProductComponent},
  {path: "viewProduct/:productId", component:ViewProductComponent},
  {path: "cart", component:ShopCartComponent},
  {path: "checkout", component:CheckoutComponent},
  {path: "changePassword/:userId", component:ChangePasswordComponent},
  {path: "allUser", component:AllUserComponent},
  {path: "viewUser/:userId", component:ViewuserComponent},
  {path: "payout/:orderId", component:PayoutComponent},
  {path: "orderConfirmation", component:OrderConfirmationComponent}
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Creation of home component using node.js in command prompt.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  AZURE  JUPYTER

PS F:\VsCode\19May> cd .\KitchenStory-e-commerce\
PS F:\VsCode\19May\KitchenStory-e-commerce> cd .\KitchenStoryFrontend\
PS F:\VsCode\19May\KitchenStory-e-commerce\KitchenStoryFrontend> ng g c home/home
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/home

**File name:**home.component.html

**File type:** html

```

<div class="container-fluid">

  <!-- card top --dashboard -->
<div class="card-top container-fluid">
  <div class="container d-flex">

    <div class="item">
      <a class="aclass" (click)="filter('')" >
        
        <h6>All items</h6>
      </a>
    </div>
    <div class="item">
      <a class="aclass" (click)="filter('fruit')" >
        
        <h6>Fruits</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" (click)="filter('veg')">
        
        <h6>Vegetable</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" (click)="filter('juice')">
        
        <h6>Juice</h6>
      </a>
    </div>
  </div>
</div>

  <div class="container-fluid" >
    <div class="row my-3 ">
      <div class="card-down col-lg-3 col-md-4 col-sm-6 border
border-3 bg-dark text-white "
        *ngFor="let pro of filterCategory | pipe : searchKey :
'prodName' ">

        
        <!-- Text inside -->

```



```

                <h1><b> {{pro.prodName}}</b><br></h1>
                <p><i>{{pro.vendorName}}</i><br>
                ₹{{pro.price}}/-<br>
                {{pro.prodDetail}}</p><br>
                <!-- Nutton of the card -->
                <button class="btn btn-warning m-1
" (click)="addToCart(pro)"><b>Add to cart</b></button>
                <br>

                <div>

                </div>

                </div>

                <!--      <div *ngElse class="col-lg-3 col-md-4 col-sm-6  bg-dark text-white
">Column 2</div>
                <div class="col-lg-3 col-md-4 col-sm-6  bg-primary ">Column 3</div>
                <div class="col-lg-3 col-md-4 col-sm-6  bg-dark text-white">Column
4</div>
                <div class="col-lg-3 col-md-4 col-sm-6  bg-primary ">Column 5</div>
                <div class="col-lg-3 col-md-4 col-sm-6  bg-dark text-white ">Column
6</div> -->

</div>

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/home

**File name:**home.component.css

**File type:**css

```

.card-top{
  position: relative;
  display: flex;
  padding-top: 0%;
  flex-direction: column;
  min-width: 0;
  word-wrap: break-word;
  background: #fff;
  background-clip: border-box;
  border: 1px solid rgba(0,0,0,0.2);
  border-radius: .25rem;
}

```

```
.item img{
width: 50px;
height: 50px;
border-radius: 50%;
}

.card-down img{
width: 150px;
height: 150px;
}

.item{
margin: 0px 15px;
text-align: center;
}
.aclass:hover{
color: royalblue;
}

.aclass{
color: black;
}
.card-down{
text-align: center;
}
.card h1{
align-content: center;
}
.card-down img{
margin-top: 2%;
margin-left: 1%;
align-content: center;
margin-bottom: 3%;
transition: 0.3s ease-in-out;
}

.card-down img:hover{
transition: 0.3s ease-in;
transform: scale(1.05);
}
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/home

**File name:**home.component.ts

**File type:**typescript

```
import { Component, OnInit } from '@angular/core';
import { KitchenProduct } from 'src/app/model/kitchen-product';
import { KitchenUserRegistered } from 'src/app/model/kitchen-user-registered';
import { CartService } from 'src/app/service/cart.service';
import { ServiceService } from 'src/app/service/service.service';

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

  public filterCategory :any;
  searchKey :string="";
  allUser: KitchenUserRegistered[];
  allProduct:KitchenProduct[];
  constructor(private service:ServiceService,private cartService :CartService) {
  }

  ngOnInit(): void {
    this.getAllProduct();

    //here that searc func defined
    this.cartService.search.subscribe(val=>{this.searchKey=val;})
  }

  search(event:any){}
  private getAll(){
    this.service.getAllUser().subscribe(
      data=>{this.allUser=data},
      error=>console.log(error));
  }

  //indide this add to cart QTy and total is added
  private getAllProduct(){
    this.service.getAllProduct()
      .subscribe(data=>{
```

```

        this.allProduct=data;
        this.filterCategory=data;
        //set type of product here if not set
        this.allProduct.forEach((a:any)=>{
            Object.assign(a,{quantity:1,total:a.price});

        });
    })
}

//this add to cart items
addToCart(item:any){
    this.cartService.addToCart(item);
}

//to check if type is == "veg"
filter(type:string){
    this.filterCategory=this.allProduct.filter((a:any)=>{
        if(a.type===type || type=='')
            return a;

    });
}
}
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/sign-in

**File name:**sign-in.component.html

**File type:**html

```

<br><p>sign-in works!</p>
<div class="container p-5" style="width:400px" style="margin:0"
style="background-color:#66ffff">
    <h2 style="text-align:center">LogIN the KitchenHouse</h2>
    <form #loginForm="ngForm" (ngSubmit)="onSubmit(loginForm.value)" style="text-align:center">
        <table style="margin-left: auto;
margin-right: auto; "><tr><b><td>
            Email:</td></b><td><input type="email" name="email" [(ngModel)] =
"user.email"
            #email = "ngModel" required>

```

```

        <div class="text-danger" *ngIf = "loginForm.submitted && email.invalid">
            <p *ngIf = "loginForm.controls['email'].errors?.['required']">
                Email is required</p>
            </div></td>
        </tr><tr><td><b>
            Password:</b></td><td><input type="password" name="password" [(ngModel)] =
"user.password"
            #password = "ngModel" required>
            <div class="text-danger" *ngIf = "loginForm.submitted && password.invalid">
                <p *ngIf = "loginForm.controls['password'].errors?.['required']">
                    password is required</p>
                </div></td>
        </tr><tr>
            <td>
                <button class="btn btn-primary" type="submit">SignIn</button></td><td>
                <a routerLink="/register">New User</a></td>
        </tr>
    </table>

</form>

<table>
    <thead><tr><th>username</th></tr></thead>
    <tbody><tr *ngFor = "let us of allUser"><td>{{us.email}}</td> <td>{{us.type}}
    </td></tr>
    </tbody>
</table>

</div>

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/sign-in

**File name:**sign-in.component.ts

**File type:**typescript

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { KitchenUserRegistered } from 'src/app/model/kitchen-user-registered';
import { ServiceService } from 'src/app/service/service.service';

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

```

```

export class SignInComponent implements OnInit {
  // user={email:'',password:''};
  user= new KitchenUserRegistered();
  lastUser:KitchenUserRegistered;
  allUser:KitchenUserRegistered[];
  constructor(private service : ServiceService, private router: Router) { }

  ngOnInit(): void {
    this.getUser();
  }
  private getUser()
  {
    this.service.getAllUser().subscribe(data=>this.allUser=data);
  }
  private getStringValue(value: any): string {
    return String(value);
  }
  onSubmit(form:any){

    //console.log(form.email);
    for(let u of this.allUser)
    { this.lastUser=this.allUser[this.allUser.length-1];

      if(u.email===this.user.email && u.password===this.user.password &&
u.type==='Admin')
        this.router.navigate(['/home-admin']); //Admin home-2 page create
      else if(u.email===this.user.email && u.password===this.user.password)
        this.router.navigate(['/home']);
        // else if(u.email===this.lastUser.email){
        //   this.router.navigate(['/register']);
        // }
    }

  }
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/home-admin

**File name:** home-admin.component.html

**File type:**html

```

<p>home-admin works!</p>
<!-- card top --dashboard -->
<div class="card-top container-fluid">
  <div class="container d-flex">
    <div class="item">
      <a class="aclass" routerLink="/addProduct">
        <i class="fa fa-plus"></i>
        <h6>Add the product</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" routerLink="/adminViewUpdateProduct">
        <i class="fa fa-eye"></i>
        <h6>View/Update all the products</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" routerLink="/allUser">
        <i class="fa fa-wrench"></i>
        <h6>Change Password</h6>
      </a>
    </div>

  </div>
</div>

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/home-admin

**File name:**home-admin.component.css

**File type:**css

```

.card-top{
  position: relative;
  display: flex;
  padding-top: 0%;
  flex-direction: column;
  min-width: 0;
  word-wrap: break-word;
  background: #fff;
  background-clip: border-box;
  border: 1px solid rgba(0,0,0,0.2);
  border-radius: .25rem;

```

```

}

.item{
    margin: 0px 15px;
    text-align: center;
}

.aclass:hover{
    color: yellowgreen;
}

.aclass{
    color: black;
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/home-admin

**File name:**home-admin.component.ts

**File type:**typescript

```

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

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

  constructor() { }

  ngOnInit(): void {
  }

}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/admin-view-update-product

**File name:** admin-view-update-product.component.html

**File type:**html

```

<div class="container-fluid exapnd-md" >

```



```

<div class="container d-flex">
  <div class="item">
    <a class="aclass" routerLink="/addProduct">
      <i class="fa fa-plus"></i>
      <h6>Add the product</h6>
    </a>
  </div>

  <div class="item">
    <a class="aclass" routerLink="/adminViewUpdateProduct">
      <i class="fa fa-eye"></i>
      <h6>View/Update all the products</h6>
    </a>
  </div>

  <div class="item">
    <a class="aclass" routerLink="/allUser">
      <i class="fa fa-wrench"></i>
      <h6>Change Password</h6>
    </a>
  </div>

</div>
<h2>Product List of Kitchen story</h2>

<table class="table table-stripped">
  <thead class="header"><tr>
    <td>Image</td>
    <td>Product name:</td>
    <td>Vendor name:</td>
    <td>Price:</td>
    <td>Action:</td>
  </tr></thead>
  <tbody>
    <tr *ngFor="let pro of allProducts">
      <td><img class="imgadmin" src={{pro.imgUrl}}></td>
      <td>{{pro.prodName}}</td>
      <td>{{pro.vendorName}}</td>
      <td><b>{{pro.price}}</b></td>
      <td>
        <button (click)="viewProduct(pro.prodId)" class="btn btn-success"><i class="fa fa-info-circle "></i></button>
        <button (click)="updateProduct(pro.prodId)" class="btn btn-warning"><i class="fa fa-pencil-square-o"></i></button>
      </td>
    </tr>
  </tbody>
</table>

```

```

                <button (click)="deleteProduct(pro.prodId)" class="btn btn-
danger"><i class="fa fa-trash-o"></i></button>
            </td>
        </tr>
    </tbody>
</table>
</div>

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/admin-view-update-product

**File name:** admin-view-update-product.component.css

**File type:**css

```

.header{
    font-size: medium;
    font-weight: bold;
    font-family: Georgia, 'Times New Roman', Times, serif;
}

.imgadmin{
    width:50px;
    height:50px;
    border-radius: 50%;
}

.btn{
    margin-left:10px;
}

.card-top{
    position: relative;
    display:flex;
    padding-top: 0%;
    flex-direction: column;
    min-width: 0;
    word-wrap: break-word;
    background: #fff;
    background-clip: border-box;
    border:1px solid rgba(0,0,0,0.2);
    border-radius: .25rem;
}

```

```

}

.item{
    margin: 0px 15px;
    text-align: center;
}

.aclass:hover{
    color: yellowgreen;
}

.aclass{
    color: black;
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/admin-view-update-product

**File name:** admin-view-update-product.component.ts

**File type:**typescript

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { KitchenProduct } from 'src/app/model/kitchen-product';
import { ServiceService } from 'src/app/service/service.service';

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

  productId:number;
  allProducts:KitchenProduct[];
  constructor(private service:ServiceService,private route: Router) { }

  ngOnInit(): void {

    this.getAllProduct();
  }

  getAllProduct(){
    this.service.getAllProduct().subscribe(data=>this.allProducts=data);
  }

  updateProduct(productId:number){

```

```

    this.route.navigate([' /updateProduct',productId]);
  }

  viewProduct(productId:number){
    this.route.navigate([' /viewProduct',productId]);
  }

  deleteProduct(productId:number){
    this.service.deleteProduct(productId).subscribe(data=>{console.log(data);this
    .goToAdminViewPro();});
  }
  goToAdminViewPro(){
    this.route.navigate([' /home-admin']);
  }
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/update-product

**File name:** update-product.component.html

**File type:**html

```

<div class="conatiner-fluid">

  <div class="container d-flex">
    <div class="item">
      <a class="aclass" routerLink="/addProduct">
        <i class="fa fa-plus"></i>
        <h6>Add the product</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" routerLink="/adminViewUpdateProduct">
        <i class="fa fa-eye"></i>
        <h6>View/Update all the products</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" routerLink="/allUser">
        <i class="fa fa-wrench"></i>
        <h6>Change Password</h6>
      </a>
    </div>
  </div>

```

```

    </div>
<h2>update-product</h2>

<div class="col-md-6 offset-md-3">
  <form (ngSubmit)="onSubmit()">
    <div class="form-group">
      <label>Product Name:</label>
      <input type="text" class="form-control" id="prodName"
[(ngModel)]="product.prodName" name="prodName">
    </div>
    <div class="form-group">
      <label>Vendor Name:</label>
      <input type="text" class="form-control" id="vendorName"
[(ngModel)]="product.vendorName" name="vendorName">
    </div>

    <div class="form-group">
      <label>Price:</label>
      <input type="number" class="form-control" id="price"
[(ngModel)]="product.price" name="price">
    </div>

    <div class="form-group">
      <label>Details:</label>
      <input type="text" class="form-control" product.prodDetail
[(ngModel)]="product.prodDetail" name="prodDetail">
    </div>

    <div class="form-group">
      <label>Img URL:</label>
      <input type="file" class="form-control" id="imgUrl" name="imgUrl">
    </div><br>
    <button class="btn btn-outline-secondary" type="submit">
      Update Product
    </button>
    <button class="btn btn-outline-primary mx-3"
routerLink="/adminViewUpdateProduct">Back</button>
  </form>
</div>

</div>

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/update-product

**File name:** update-product.component.css

**File type:**css

```
.card-top{
  position: relative;
  display: flex;
  padding-top: 0%;
  flex-direction: column;
  min-width: 0;
  word-wrap: break-word;
  background: #fff;
  background-clip: border-box;
  border: 1px solid rgba(0,0,0,0.2);
  border-radius: .25rem;
}

.item{
  margin: 0px 15px;
  text-align: center;
}

.aclass:hover{
  color: yellowgreen;
}

.aclass{
  color: black;
}
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/update-product

**File name:** update-product.component.ts

**File type:**ts

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { KitchenProduct } from 'src/app/model/kitchen-product';
import { ServiceService } from 'src/app/service/service.service';

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

```

export class UpdateProductComponent implements OnInit {
  productId:number;
  product: KitchenProduct=new KitchenProduct();
  constructor(private service:ServiceService,private
  activateRoute:ActivatedRoute,private route:Router) { }

  ngOnInit(): void {
    this.productId=this.activateRoute.snapshot.params['productId'];
    this.service.getProductByProductId(this.productId).subscribe(data=>this.produ
ct=data);
  }
  onSubmit(){
    this.service.updateProduct(this.productId,this.product).subscribe(data=>this.
goToProductList());
  }

  goToProductList(){
    this.route.navigate(['/adminViewUpdateProduct']);
  }
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/view-product

**File name:** view-product.component.html

**File type:**html

```

<div class="container-fluid exapnd-md" >

  <div class="container d-flex">
    <div class="item">
      <a class="aclass" routerLink="/addProduct">
        <i class="fa fa-plus"></i>
        <h6>Add the product</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" routerLink="/adminViewUpdateProduct">
        <i class="fa fa-eye"></i>
        <h6>View/Update all the products</h6>
      </a>
    </div>
  </div>

```

```

        <div class="item">
            <a class="aclass" routerLink="/allUser">
                <i class="fa fa-wrench"></i>
                <h6>Change Password</h6>
            </a>
        </div>

    </div>

    <p>View the product with specific Id</p>
    <table class="table ">
        <tr>
            <td><b>Product id:</b></td>
            <td>{{product.prodId}}</td>
        </tr><tr>
            <td><b>Product name:</b></td>
            <td>{{product.prodName}}</td>
        </tr><tr>
            <td><b>Vendor Name:</b></td>
            <td>{{product.vendorName}}</td>
        </tr><tr>
            <td><b>Product Details:</b></td>
            <td>{{product.prodDetail}}</td>
        </tr><tr>
            <td><b>Price:</b></td>
            <td><b><i>{{product.price}}</i></b></td>
        </tr><tr>
            <td><b>Image:</b></td>
            <td></td>
        </tr><tr>
            <td><b>Image Url:</b></td>
            <td>{{product.imgUrl}}</td>
        </tr>
    </table>
    <form (ngSubmit)="onSubmit()">
        <button type="sumbit" class="btn btn-secondary"><i class="fa fa-hand-o-left"
style="font-size:70px; position: relative;"></i></button>
    </form>

</div>

```



**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/view-product

**File name:** view-product.component.css

**File type:**css

```
.card-top{
  position: relative;
  display: flex;
  padding-top: 0%;
  flex-direction: column;
  min-width: 0;
  word-wrap: break-word;
  background: #fff;
  background-clip: border-box;
  border: 1px solid rgba(0,0,0,0.2);
  border-radius: .25rem;
}

.item{
  margin: 0px 15px;
  text-align: center;
}

.aclass:hover{
  color: yellowgreen;
}

.aclass{
  color: black;
}

.imgView{
  width: 200px;
  height: 200px;
  border-radius: 300%;
}
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/view-product

**File name:** view-product.component.ts

**File type:**typescript

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { KitchenProduct } from 'src/app/model/kitchen-product';
import { ServiceService } from 'src/app/service/service.service';
```

```

@Component({
  selector: 'app-view-product',
  templateUrl: './view-product.component.html',
  styleUrls: ['./view-product.component.css']
})
export class ViewProductComponent implements OnInit {
  productId:number;
  product:KitchenProduct=new KitchenProduct();
  constructor(private service :ServiceService, private
  activatedRouter:ActivatedRoute, private route:Router) { }

  ngOnInit(): void {
    this.productId=this.activatedRouter.snapshot.params['productId'];

    this.service.getProductByProductId(this.productId).subscribe(data=>this.product
    =data);
  }

  onSubmit(){
    this.route.navigate(['/adminViewUpdateProduct']);
  }
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/add-product

**File name:** add-product.component.html

**File type:**html

```

<p>add-product works!</p>

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

  <form (ngSubmit)="onSubmit()">
    <div class="form-group">
      <label>Product Name:</label>
      <input type="text" class="form-control" id="prodName"
      [(ngModel)]="product.prodName" name="prodName">
    </div>
    <div class="form-group">
      <label>Vendor Name:</label>
      <input type="text" class="form-control" id="vendorName"
      [(ngModel)]="product.vendorName" name="vendorName">

```

```

    </div>

    <div class="form-group">
      <label>Price (₹):</label>
      <input type="number" class="form-control" id="price"
[(ngModel)]="product.price" name="price">
    </div>

    <div class="form-group">
      <label>Details:</label>
      <input type="text" class="form-control" id="prodDetail"
[(ngModel)]="product.prodDetail" name="prodDetail">
    </div>

    <div class="form-group">
      <label>Type:</label>
      <input type="text" class="form-control" id="type"
[(ngModel)]="product.type" name="type">
    </div>

    <div class="form-group">
      <!-- <label>Img URL:</label> -->
      <!-- <input type="file" class="form-control" [(ngModel)]="current"
(change)="onFileSelected($event)" > -->
    </div>

    <div class="form-group">
      <label>Image Url:</label>
      <input type="text" class="form-control" id="imgUrl"
[(ngModel)]="product.imgUrl" name="imgUrl">
    </div>
    <br>
    <button class="btn btn-outline-secondary " type="submit">
      Add Product
    </button>
    <button class="btn btn-outline-primary mx-3"
routerLink="/adminViewUpdateProduct">Back</button>
  </form>
</div>

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/add-product

**File name:** add-product.component.ts

**File type:** TypeScript

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { KitchenProduct } from 'src/app/model/kitchen-product';
import { ServiceService } from 'src/app/service/service.service';

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

  product:KitchenProduct =new KitchenProduct();
  current:any;
  constructor(private service:ServiceService, private route:Router) { }

  ngOnInit(): void {

  }

  //image

  onSubmit(){
    this.savePeoduct();
    console.log(this.product);
  }
  savePeoduct(){
    this.service.addProduct(this.product).subscribe(data=>{console.log(data);this
    .goToBack()},error=>console.log(error));
  }

  goToBack(){
    this.route.navigate([' /adminViewUpdateProduct']);
  }
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/change-password

**File name:** change-password.component.html

**File type:**html

```

<div class="conatiner-fluid">

  <div class="container d-flex">
    <div class="item">
      <a class="aclass" routerLink="/addProduct">
        <i class="fa fa-plus"></i>
        <h6>Add the product</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" routerLink="/adminViewUpdateProduct">
        <i class="fa fa-eye"></i>
        <h6>View/Update all the products</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" routerLink="/allUser">
        <i class="fa fa-wrench"></i>
        <h6>Change Password</h6>
      </a>
    </div>

  </div>
  <br>
<h2>update-user</h2>

<div class="col-md-6 offset-md-3">
  <form (ngSubmit)="onSubmit()">
    <div class="form-group">
      <label>User Name(Email):</label>
      <input type="text" class="form-control" id="email"
[(ngModel)]="user.email" name="email">
    </div>
    <div class="form-group">
      <label>Full Name:</label>
      <input type="text" class="form-control" id="fullName"
[(ngModel)]="user.fullName" name="fullName">
    </div>

    <div class="form-group">
      <label>Password:</label>
      <input type="text" class="form-control" id="password"
[(ngModel)]="user.password" name="password">

```

```

        </div>

        <div class="form-group">
            <label>Address:</label>
            <input type="text" class="form-control" id="address"
[(ngModel)]="user.address" name="address">
        </div>
        <br>
        <button class="btn btn-outline-warning" type="submit">
            Update User
        </button>
        <button class="btn btn-outline-danger mx-2"
routerLink="/allUser">Back</button>
    </form>
</div>

</div>

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/change-password

**File name:** change-password.component.css

**File type:**css

```

.card-top{
    position: relative;
    display: flex;
    padding-top: 0%;
    flex-direction: column;
    min-width: 0;
    word-wrap: break-word;
    background: #fff;
    background-clip: border-box;
    border: 1px solid rgba(0,0,0,0.2);
    border-radius: .25rem;
}

.item{
    margin: 0px 15px;
    text-align: center;
}

.a:hover{
    color: yellowgreen;
}

```

```
.aclass{  
  color: black;  
}
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/change-password

**File name:** change-password.component.ts

**File type:**typescript

```
import { Component, OnInit } from '@angular/core';  
import { ActivatedRoute, Router } from '@angular/router';  
import { KitchenUserRegistered } from 'src/app/model/kitchen-user-registered';  
import { ServiceService } from 'src/app/service/service.service';  
  
@Component({  
  selector: 'app-change-password',  
  templateUrl: './change-password.component.html',  
  styleUrls: ['./change-password.component.css']  
})  
export class ChangePasswordComponent implements OnInit {  
  userId:number;  
  user:KitchenUserRegistered=new KitchenUserRegistered();  
  
  constructor(private service:ServiceService,private  
activate:ActivatedRoute,private route:Router) { }  
  
  ngOnInit(): void {  
    this.userId=this.activate.snapshot.params['userId'];  
    this.service.getUserById(this.userId).subscribe(data=>this.user=data);  
  }  
  
  onSubmit(){  
    this.service.updateUser(this.userId,this.user).subscribe(data=>this.goToUserL  
ist());  
  }  
  
  goToUserList(){  
    this.route.navigate(['/allUser']);  
  }  
}
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/viewuser

**File name:** viewuser.component.html

**File type:**html

```
<div class="container-fluid exapnd-md" >

  <div class="container d-flex">
    <div class="item">
      <a class="aclass" routerLink="/addProduct">
        <i class="fa fa-plus"></i>
        <h6>Add the product</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" routerLink="/adminViewUpdateProduct">
        <i class="fa fa-eye"></i>
        <h6>View/Update all the products</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" routerLink="/allUser">
        <i class="fa fa-wrench"></i>
        <h6>Change Password</h6>
      </a>
    </div>

  </div>

  <h2>View the product with specific Id</h2>
  <table class="table ">
    <tr>
      <td><b>Admin id:</b></td>
      <td>{{user.id}}</td>
    </tr><tr>
      <td><b>Admin userID:</b></td>
      <td>{{user.email}}</td>
    </tr><tr>
      <td><b>Admin Name:</b></td>
      <td>{{user.fullName}}</td>
    </tr><tr>
```



```

        <td><b>Password:</b></td>
        <td>{{user.password}}</td>
    </tr><tr>
        <td><b>Address:</b></td>
        <td>{{user.address}}</td>
    </tr>
</table>
<form (ngSubmit)="onSubmit()">
    <button type="submit" class="btn btn-secondary"><i class="fa fa-hand-o-left"
style="font-size:70px; position: relative;"></i></button>
</form>

</div>

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/viewuser

**File name:** viewuser.component.css

**File type:**css

```

.card-top{
    position: relative;
    display:flex;
    padding-top: 0%;
    flex-direction: column;
    min-width: 0;
    word-wrap: break-word;
    background: #fff;
    background-clip: border-box;
    border:1px solid rgba(0,0,0,0.2);
    border-radius: .25rem;
}

.item{
    margin: 0px 15px;
    text-align: center;
}

.aclass:hover{
    color: yellowgreen;
}

.aclass{
    color: black;
}

```

```
.imgView{
  width: 200px;
  height:200px;
  border-radius: 300%;
}
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/viewuser

**File name:** viewuser.component.ts

**File type:**typescript

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { KitchenUserRegistered } from 'src/app/model/kitchen-user-registered';
import { ServiceService } from 'src/app/service/service.service';

@Component({
  selector: 'app-viewuser',
  templateUrl: './viewuser.component.html',
  styleUrls: ['./viewuser.component.css']
})
export class ViewuserComponent implements OnInit {
  userId:number;
  user:KitchenUserRegistered=new KitchenUserRegistered();
  constructor(private service:ServiceService,private
activate:ActivatedRoute,private route:Router) { }

  ngOnInit(): void {
    this.userId=this.activate.snapshot.params['userId'];
    this.service.getUserById(this.userId).subscribe(data=>this.user=data);
  }

  onSubmit(){
    this.route.navigate(['/allUser']);
  }
}
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/all-user

**File name:** all-user.component.html

**File type:**html

```

<div class="container-fluid exapnd-md" >

  <div class="container d-flex">
    <div class="item">
      <a class="aclass" routerLink="/addProduct">
        <i class="fa fa-plus"></i>
        <h6>Add the product</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" routerLink="/adminViewUpdateProduct">
        <i class="fa fa-eye"></i>
        <h6>View/Update all the products</h6>
      </a>
    </div>

    <div class="item">
      <a class="aclass" routerLink="/allUser">
        <i class="fa fa-wrench"></i>
        <h6>Change Password</h6>
      </a>
    </div>
  </div>
  <br>
  <h2>User List of Kitchen story</h2>

  <table class="table table-stripped">
    <thead class="header"><tr>
      <td>Username</td>
      <td>Password</td>
      <td>Full Name</td>
      <td>Address</td>
      <td>Action:</td>
    </tr></thead>
    <tbody *ngFor="let user of allUser ">
      <tr *ngIf="user.type=== 'Admin'">

        <td><b>{{user.email}}</b></td>
        <td>{{user.password}}</td>
        <td>{{user.fullName}}</td>
        <td>{{user.address}}</td>
        <td >
          <button (click)="viewUser(user.id)" class="btn btn-
success"><i class="fa fa-info-circle "></i></button>

```

```

                <button (click)="updatePassword(user.id)"class="btn btn-
outline-danger"><i class="fa fa-pencil-square-o"></i></button>
            </td>
        </tr>
    </tbody>
</table>

</div>

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/all-user

**File name:** all-user.component.css

**File type:**css

```

.header{
  font-size: medium;
  font-weight: bold;
  font-family: Georgia, 'Times New Roman', Times, serif;
}

.imgadmin{
  width:50px;
  height:50px;
  border-radius: 50%;
}

.btn{
  margin-left:10px;
}

.card-top{
  position: relative;
  display:flex;
  padding-top: 0%;
  flex-direction: column;
  min-width: 0;
  word-wrap: break-word;
  background: rgb(218, 15, 15);
  background-clip: border-box;
  border:1px solid rgba(16, 161, 187, 0);
  border-radius: .25rem;
}

```

```

}

.item{
    margin: 0px 15px;
    text-align: center;
}

.aclass:hover{
    color: yellowgreen;
}

.aclass{
    color: black;
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/all-user

**File name:** all-user.component.ts

**File type:**typescript

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { KitchenUserRegistered } from 'src/app/model/kitchen-user-registered';
import { ServiceService } from 'src/app/service/service.service';

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

  userId:number;
  allUser:KitchenUserRegistered[];
  constructor(private service:ServiceService,private route:Router) { }

  ngOnInit(): void {
    this.service.getAllUser().subscribe(data=>this.allUser=data);
  }

  viewUser(userId:number){
    this.route.navigate(['/viewUser',userId]);
  }

  updatePassword(userId:number){
    this.route.navigate(['/changePassword',userId]);
  }
}

```

```
}  
  
}
```

#### USER functions--

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/register

**File name:** register.component.html

**File type:**html

```
<h3>register Yourself!</h3>  
<div class="col-md-6 offset-md-3">  
  <form (ngSubmit)="onSubmit()">  
    <div class="form-group">  
      <label>Name:</label>  
      <input type="text" class="form-control" id="name"  
[(ngModel)]="user.fullName" name="name">  
    </div>  
    <div class="form-group">  
      <label>Password:</label>  
      <input type="password" class="form-control" id="password"  
[(ngModel)]="user.password" name="password">  
    </div>  
  
    <div class="form-group">  
      <label>Email:</label>  
      <input type="text" class="form-control" id="email"  
[(ngModel)]="user.email" name="email">  
    </div>  
  
    <div class="form-group">  
      <label>Address:</label>  
      <input type="text" class="form-control" id="address"  
[(ngModel)]="user.address" name="address">  
    </div><br>  
    <button class="btn btn-outline-success" type="submit">  
      Register Me  
    </button>  
  </form>  
</div>
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/register

**File name:** register.component.ts

**File type:**typescript

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { KitchenUserRegistered } from 'src/app/model/kitchen-user-registered';
import { ServiceService } from 'src/app/service/service.service';

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

  user: KitchenUserRegistered =new KitchenUserRegistered();
  constructor(private service:ServiceService, private route:Router) { }

  ngOnInit(): void {

  }

  saveUser(){
    this.service.addUser(this.user).subscribe(data=>{console.log(data);
    this.goToHome()},error=>console.error());
    ;
  }

  goToHome(){
    this.route.navigate(['/home']);
  }

  onSubmit(){
    console.log(this.user);
    this.saveUser();
  }
}
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/navbar

**File name:** navbar.component.html

**File type:**html

```
<div class="container-fluid">
  <div class="container">
    <h2>This site is created by RahulPersie66@gmail.com. Hope you enjoy the
delicious food of our website.</h2>
    <h3>We are eager to provide you best food for your kitchen, directly fresh
from the farm.</h3>

  </div>
</div>
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/navbar

**File name:** navbar.component.ts

**File type:**typescript

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

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

  constructor() { }

  ngOnInit(): void {
  }

}
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/shop-cart

**File name:** shop-cart.component.html

**File type:**html



```

<div class="container">

  <ng-container *ngIf="product.length!=0">

    <table class="table table-responsive">
      <thead>
        <tr>
          <th>Sr.No</th>
          <th>Product name</th>
          <th>Image</th>
          <th>Description</th>
          <th>Price</th>
          <th>Qty</th>
          <th>Total</th>
          <th>Action</th>
        </tr>
      </thead>
      <tbody>
        <tr *ngFor="let item of product;let i=index">
          <td>{{i+1}}</td>
          <td>{{item.prodName}}</td>
          <td></td>
          <td>{{item.prodDetail}}</td>
          <td>{{item.price}}</td>
          <td>{{item.quantity}}</td>
          <td>{{item.total}}</td>
          <td>
            <button class="btn btn-danger"
(click)="removeItem(item)"><i class="fa fa-trash"></i></button>
          </td>
        </tr>
        <tr>
          <td colspan="4"></td>
          <td><button class="btn btn-danger"
(click)="emptyCart()">Empty Cart</button></td>
          <td><button class="btn btn-primary"
routerLink="/home">Shop more</button></td>
          <td><button class="btn btn-warning"
routerLink="/checkout">CheckOut</button></td>
          <td><strong>Grand Total:
₹{{grandTotal}}</strong></td>
        </tr>
      </tbody>
    </table>
  </ng-container>
</div>

```

```

    </ng-container>
  </div>

  <!-- Empty cart -->
  <ng-container *ngIf="product.length==0">

    <div class="container">
      <div class="card">
        <h5 class="card-title">My Cart</h5>
      </div>
      <div class="center">
        
        <h4>Your Cart is empty.</h4>
        <h6>Add item to it now!!</h6>
        <button class="btn btn-primary"
routerLink="/home">Shopnow</button>
      </div>
    </div>

  </ng-container>

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/shop-cart

**File name:** shop-cart.component.css

**File type:**css

```

.card{
  height:40%;
  margin:0px;
  padding:2%;
}

.center{
  position: relative;
  top:40%;
  left:50%;
  transform: translate(-50%,50%);
  text-align: center;
}

.center img{
  text-decoration: none;
  color: inherit;
}

```

```

border:none;
outline: none;
height:20%;
width: 20%;
margin: 20px,20px;
}

h4,h6{
  font-weight: 400;
}

.center btn{
  font-size:14px !important;
  margin-top: 20px !important;
  font-weight: 400;
  padding: 12px 72px;
  border-radius: 3px !important;
}

.table-responsive img{
  width: 50%;
  height:40%;
  border-radius: 50%;
}

.card-table{
  position: relative;

  flex-direction: row;
  min-width: none;
  word-wrap: break-word;
  background-clip: border-box;
  border:1 px solid rgba(0,0,0,0.2);
  border-radius: 0.25rem;
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/shop-cart

**File name:** shop-cart.component.ts

**File type:**typescript

```

import { Component, OnInit } from '@angular/core';
import { CartService } from 'src/app/service/cart.service';

```

```

@Component({
  selector: 'app-shop-cart',
  templateUrl: './shop-cart.component.html',
  styleUrls: ['./shop-cart.component.css']
})
export class ShopCartComponent implements OnInit {
  public product:any=[];
  public grandTotal :number;
  constructor(private cartService:CartService) { }

  ngOnInit(): void {
    this.cartService.getProducts().subscribe(res=>{
      this.product=res;
      this.grandTotal=this.cartService.getTotalPrice();
    })
  }

  removeItem(item:any){
    this.cartService.removeCartItem(item);
  }

  emptyCart(){
    this.cartService.removeAllCart();
  }
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/checkout

**File name:** checkout.component.html

**File type:**html

```

<section class="page-header">
  <div class="overly"></div>
  <div class="container">
    <div class="row justify-content-center">
      <div class="col-lg-6">
        <div class="content text-center">
          <h1 class="mb-3">Checkout</h1>
          <p>Hath after appear tree great fruitful green dominion moveth sixth
abundantly image that midst of god day multiply you'll which</p>

          <nav aria-label="breadcrumb">
            <ol class="breadcrumb bg-transparent justify-content-center">

```

```
<li class="breadcrumb-item"><a routerLink="/">Home</a></li>  
<li class="breadcrumb-item active" aria-current="page">Checkout</li>  
</ol>  
</nav>  
</div>  
</div>  
</div>  
</div>  
</section>  
  
<div class="page-wrapper">  
  <div class="checkout shopping">  
    <div class="container">  
      <div class="row">  
        <div class="col-lg-8 pr-5">  
  
          <div class="billing-details mt-5">  
            <h4 class="mb-4">Billing Details</h4>  
            <form class="checkout-form " (ngSubmit)="onSubmit()">  
              <div class="row">  
                <div class="col-lg-6">  
                  <div class="form-group mb-4">  
                    <label for="first_name">First Name</label>  
                    <input type="text" class="form-control"  
[(ngModel)]="order.name" name="name" placeholder="">  
                  </div>  
                </div>  
                <div class="col-lg-6">  
                  <div class="form-group mb-4">  
                    <label for="last_name">Last Name</label>  
                    <input type="text" class="form-control"  
id="last_name" placeholder="">  
                  </div>  
                </div>  
  
                <div class="col-lg-12">  
                  <div class="form-group mb-4">  
                    <label for="company_name">Country</label>  
                    <input type="text" class="form-control"  
name="country" [(ngModel)]="order.country" placeholder="India">  
                  </div>  
                </div>  
  
                <div class="col-lg-12">
```

```

        <div class="form-group mb-4">
            <label for="first_name">Street Address</label>
            <input type="text" class="form-control"
id="street" placeholder="">
        </div>
    </div>

    <div class="col-lg-12">
        <div class="form-group mb-4">
            <label for="first_name">Town / City </label>
            <input type="text" class="form-control" id="city"
placeholder="Apartment">
        </div>
    </div>

    <div class="col-lg-12">
        <div class="form-group mb-4">
            <label for="company_name">District </label>
            <input type="text" class="form-control"
name="district">
        </div>
    </div>

    <div class="col-lg-12">
        <div class="form-group mb-4">
            <label for="first_name">Postcode / ZIP
(optional)</label>
            <input type="text" class="form-control"
name="postcode" [(ngModel)]="order.postcode" placeholder="">
        </div>
    </div>

    <div class="col-lg-12">
        <div class="form-group mb-4">
            <label for="first_name">Phone </label>
            <input type="text" class="form-control"
name="phone" [(ngModel)]="order.phone" placeholder="">
        </div>
    </div>

    <div class="col-lg-12">
        <div class="form-group mb-4">
            <label for="first_name">Email address </label>
            <input type="text" class="form-control"
name="email" [(ngModel)]="order.email" placeholder="">
        </div>
    </div>

```

```

        <div class="col-lg-12">
            <div class="form-group mb-4">
                <button type="submit" class="form-control btn btn-
secondary" >Place Order</button>
            </div>
        </div>

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

<!-- Order sidebar Summery -->
<div class="col-md-6 col-lg-4">
    <div class="product-checkout-details mt-5 mt-lg-0">
        <h4 class="mb-4 border-bottom pb-4">Order Summary</h4>

        <div class="media product-card" *ngFor="let pro of
product">

            <p>{{pro.prodName}}</p>
            <div class="media-body text-right">
                <p class="h5">1 x ₹{{pro.price}}</p>
            </div>
        </div>

        <ul class="summary-prices list-unstyled mb-4">
            <li class="d-flex justify-content-between">
                <span>Subtotal:</span>
                <span class="h5">₹{{grandTotal}}</span>
            </li>
            <li class="d-flex justify-content-between">
                <span>Shipping:</span>
                <span class="h5">Free</span>
            </li>
            <li class="d-flex justify-content-between">
                <span>Total(includes 7% CGTax + 7% SGTax)</span>
                <span class="h5">rs{{cgstGrand}}</span>
            </li>
        </ul>

    </div>
</div>

```

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

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/checkout

**File name:** checkout.component.ts

**File type:**typescript

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { KitchenOrderBooked } from '../model/kitchen-order-booked';
import { CartService } from '../service/cart.service';
import { ServiceService } from '../service/service.service';

@Component({
  selector: 'app-checkout',
  templateUrl: './checkout.component.html',
  styleUrls: ['./checkout.component.css']
})
export class CheckoutComponent implements OnInit {
  public orderId:number=133;
  public order:KitchenOrderBooked=new KitchenOrderBooked();
  public product:any=[];
  public grandTotal :number;
  public cgstGrand:number;
  public allItem:string;
  proComp:string;
  allOrder:KitchenOrderBooked[];
  constructor(private cartService:CartService,private
service:ServiceService,private route:Router) { }

  ngOnInit(): void {

    this.cartService.getProducts().subscribe(res=>{
      this.product=res;
      this.grandTotal=this.cartService.getTotalPrice();
      this.allItem=this.cartService.getAllItem();
      this.cgstGrand=this.grandTotal*0.14+this.grandTotal;
      this.order.total=this.cgstGrand;
      this.order.productOrdered=this.allItem;
    });
  }
}
```



```

    })

    this.service.getAllOrder().subscribe(data=>this.allOrder=data);

}

onSubmit(){
  console.log(this.order);
  this.saveOrderData();

  for(let pro of this.allOrder)
  {
    // console.log(this.proComp);
    if(pro.total==this.cgstGrand)
    {
      this.orderId=pro.orderId;
      break;
    }

  }
  this.route.navigate(['/payout',this.orderId]);
}

saveOrderData(){
  this.service.addOrderBooked(this.order).subscribe(data=>{
    console.log(this.order.orderId);
  });
}
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/payout

**File name:** payout.component.html

**File type:**html

```

<p>payout works!</p>
<div class="container" style="color:black">
  <form (ngSubmit)="onSubmit()">
    <div class="col-lg-6">
      <div class="form-group mb-4">

```

```

    <label>
      Enter Card Number
    </label>
    <input type="number" name="cardNumber" [(ngModel)]="order.cardNumber">
  </div>
</div>

<div class="col-lg-6">
  <div class="form-group mb-4">
    <label>
      Exp month/year
    </label>
    <input type="text">
  </div>
</div>

<div class="col-lg-6">
  <div class="form-group mb-4">
    <label>
      CVV
    </label>
    <input type="number">
  </div>
</div>

<div class="col-lg-6">
  <div class="form-group mb-4">
    <button type="submit" class="btn btn-secondary">Payment</button>
  </div>
</div>
</form>
</div>

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/payout

**File name:** payout.component.ts

**File type:** typeScript

```

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { KitchenOrderBooked } from 'src/app/model/kitchen-order-booked';
import { CartService } from 'src/app/service/cart.service';
import { ServiceService } from 'src/app/service/service.service';

@Component({

```

```

    selector: 'app-payout',
    templateUrl: './payout.component.html',
    styleUrls: ['./payout.component.css']
  })
  export class PayoutComponent implements OnInit {
    orderId:number=133;
    proComp:string;
    product:any;
    productOrdered:string;
    allOrder:KitchenOrderBooked[];
    order:KitchenOrderBooked=new KitchenOrderBooked();
    constructor(private route:Router,private cartService:CartService,private
service:ServiceService,private activate:ActivatedRoute) { }

    ngOnInit(): void {
      this.orderId=this.activate.snapshot.params['orderId'];
      this.service.getOrderById(this.orderId).subscribe(data=>this.order=data);
      this.service.getAllOrder().subscribe(data=>this.allOrder=data);

    }

    onSubmit(){

      this.service.updateOrderById(this.orderId,this.order).subscribe(data=>{this.o
rder=data; console.log(this.order)});
      this.route.navigate(['/orderConfirmation']);
    }
  }
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/order-confirmation

**File name:** order-confirmation.component.html

**File type:**html

```

<div class="conatiner">
  <h3>Thank you for purchasing at the KitchenStory.</h3>
  <h5>Your order has been succesfully placed.</h5>
  <h6>Details of Succesful Order placed-</h6>

  <div class="container">
    <table class="table table-stripped">
      <thead>

```

```

        <tr>
            <td>Product</td>
            <td>Quantity</td>
            <td>Price (₹)</td>
        </tr>
    </thead>
    <tbody>
        <tr *ngFor="let pro of cartProduct">
            <td>{{pro.prodName}}</td>
            <td>{{pro.quantity}}</td>
            <td>₹{{pro.price}}</td>
        </tr>
        <tr>
            <td colspan="2"></td>
            <td><b>{{grandTotal}}</b></td>
        </tr>
        <tr>
            <td colspan="2">includes CGST (7%) + SGST (7%)</td>
            <td><b>₹{{alltotal - grandTotal}}</b></td>
        </tr>
        <tr>
            <td colspan="2">Payment Made (₹)</td>
            <td><b>₹{{alltotal}}</b></td>
        </tr>
    </tbody>
</table>
<button class="btn btn-primary ml-1" routerLink="/home">More
Shop</button>
</div>

</div>

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/home/order-confirmation

**File name:** order-confirmation.component.ts

**File type:**typescript

```

import { Component, OnInit } from '@angular/core';
import { CartService } from 'src/app/service/cart.service';

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

```

```

})
export class OrderConfirmationComponent implements OnInit {

  cartProduct: any[];
  grandTotal:number;
  alltotal:number;
  constructor(private cart:CartService) { }

  ngOnInit(): void {

    this.cart.getProducts().subscribe(res=>
    {
      this.cartProduct=res;
      this.grandTotal=this.cart.getTotalPrice();
      this.alltotal=this.grandTotal*0.14+this.grandTotal;
    }
    );
  }
}

```

Now model classes creation to store the data in the database using mysql and springboot in backend.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  AZURE  JUPYTER

PS F:\VsCode\19May> cd .\KitchenStory-e-commerce\
PS F:\VsCode\19May\KitchenStory-e-commerce> cd .\KitchenStoryFrontend\
PS F:\VsCode\19May\KitchenStory-e-commerce\KitchenStoryFrontend> ng g class model/kitchen_user_registered

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/model/kitchen\_user\_registered

**File name:** kitchen-user-registered.ts

**File type:**typescript

```

export class KitchenUserRegistered {
  id: number;
  email :string;
  fullName:string;
  password:string;
  address:string;
  cardType:string;

```

```
    cardNumber: number;  
    type: string;  
}
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/model/kitchen\_product

**File name:** kitchen-product.ts

**File type:**typescript

```
export class KitchenProduct {  
    prodId: number;  
    prodName :string ;  
    vendorName :string ;  
    price : number;  
    imgUrl :string ;  
    prodDetail :string ;  
    type:string;  
}
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/model/kitchen\_order\_booked

**File name:** kitchen-order-booked.ts

**File type:**typescript

```
export class KitchenOrderBooked {  
    orderId:number;  
    name : string;  
    country : string;  
    address : string;  
    postcode : number;  
    email : string;  
    phone : number;  
    productOrdered:string;  
    total:number;  
    cardNumber:number;  
}
```

Now for, get, post, put and delete method of springboot to work we have to create service which will enable these methods.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  AZURE  JUPYTER

PS F:\VsCode\19May> cd .\KitchenStory-e-commerce\
PS F:\VsCode\19May\KitchenStory-e-commerce> cd .\KitchenStoryFrontend\
PS F:\VsCode\19May\KitchenStory-e-commerce\KitchenStoryFrontend> ng g s service/service
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/service

**File name:** service.service.ts

**File type:**typeScript

```
import { HttpClient, HttpClientModule } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { KitchenOrderBooked } from '../model/kitchen-order-booked';
import { KitchenProduct } from '../model/kitchen-product';
import { KitchenUserRegistered } from '../model/kitchen-user-registered';

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

  url= 'http://localhost:8080';

  constructor(private httpClient:HttpClient) { }

  getAllUser():Observable<KitchenUserRegistered[]>{
    return
this.httpClient.get<KitchenUserRegistered[]>(`${this.url}/admin/showAllUser`);
  }

  getAllProduct():Observable<KitchenProduct[]>{
    return
this.httpClient.get<KitchenProduct[]>(`${this.url}/admin/showProducts`);
  }

  addUser(user:KitchenUserRegistered) :Observable<Object>{
    return
this.httpClient.post(`${this.url}/admin/addedSignUpUser`,user,{responseType:'text
'});
  }
}
```

```

    }

    addProduct(product:KitchenProduct):Observable<Object>{
        return this.httpClient.post(`${this.url}/admin/addProductDetails`,product);
    }

    getProductByProductId(productId:number):Observable<KitchenProduct>{
        return
this.httpClient.get<KitchenProduct>(`${this.url}/admin/product/${productId}`);
    }

    getUserById(userId:number):Observable<KitchenUserRegistered>{
        return
this.httpClient.get<KitchenUserRegistered>(`${this.url}/admin/changePassword/${userId}`);
    }

    getOrderById(orderId:number):Observable<KitchenOrderBooked>{
        return
this.httpClient.get<KitchenOrderBooked>(`${this.url}/checkout/${orderId}`);
    }

    updateProduct(productId:number,product:KitchenProduct):Observable<KitchenProduct>{
        return
this.httpClient.put<KitchenProduct>(`${this.url}/admin/upadte/${productId}`,product);
    }

    updateUser(userId:number,user:KitchenUserRegistered):Observable<KitchenUserRegistered>{
        return
this.httpClient.put<KitchenUserRegistered>(`${this.url}/admin/updatePassword/${userId}`,user);
    }

    updateOrderById(orderId:number,order:KitchenOrderBooked):Observable<KitchenOrderBooked>{
        return
this.httpClient.put<KitchenOrderBooked>(`${this.url}/checkout/${orderId}`,order);
    }

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

```



```

        return
this.httpClient.delete(`${this.url}/admin/deleteProduct/${productId}`, {responseTy
pe: 'text'}));
    }

    addOrderBooked(order : KitchenOrderBooked):Observable<Object>{
        return this.httpClient.post(`${this.url}/checkout/orderBooked`, order);
    }

    getAllOrder():Observable<KitchenOrderBooked[]>{
        return this.httpClient.get<KitchenOrderBooked[]>(`${this.url}/allOrder`);
    }
}

```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/service

**File name:** cart.service.ts

**File type:**typescript

```

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

@Injectable({
  providedIn: 'root'
})
export class CartService {
  public cartItemList:any=[];
  public productList=new BehaviorSubject<any>([]);
  public search=new BehaviorSubject<string>("");
  constructor() { }

  //return Product list as Any observable type
  getProducts(){
    return this.productList.asObservable();
  }

  //set Product list
  setProduct(product:any){
    this.cartItemList.push(...product);
    this.productList.next(product);
  }

  addToCart(product:any){

```

```

    this.cartItemList.push(product);
    this.productList.next(this.cartItemList);
    this.getTotalPrice();
    this.getAllItem();
    console.log(this.cartItemList);
  }

  getTotalPrice():number{
    let grandTotal=0;
    this.cartItemList.map((a:any)=>{
      grandTotal+=a.price;
    })
    return grandTotal;
  }

  getAllItem():string{
    let allItem="";
    this.cartItemList.map((a:any)=>{
      allItem+=a.prodName;
      allItem+=", ";
    })
    return allItem;
  }

  removeCartItem(product:any){
    this.cartItemList.map((a:any,index:any)=>{
      if(product.prodId==a.prodId){
        this.cartItemList.splice(index,1);
      }
    })
    this.productList.next(this.cartItemList);
  }

  removeAllCart(){
    this.cartItemList=[];
    this.productList.next(this.cartItemList);
  }
}

```

Now to search bar to work create a filter using ng pipe command to search the related item name in the product database.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  AZURE  JUPYTER

PS F:\VsCode\19May> cd .\KitchenStory-e-commerce\
PS F:\VsCode\19May\KitchenStory-e-commerce> cd .\KitchenStoryFrontend\
PS F:\VsCode\19May\KitchenStory-e-commerce\KitchenStoryFrontend> ng g pipe filter[]
```

**Location:** KitchenStory-e-commerce/KitchenStoryFrontend/src/app/search

**File name:** pipe.pipe.ts

**File type:**typescript

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

@Pipe({
  name: 'pipe'
})
export class PipePipe implements PipeTransform {

  transform(value:any[],filterString:string,propName:string): any[] {
    const result:any=[];
    if(!value||filterString===' '|| propName==='')
    {
      return value;
    }

    value.forEach((a:any)=>{
      if(a[propName].trim().toLowerCase().includes(filterString.toLowerCase
    ())) {
        result.push(a);
      }
    });
    return result;
  }

}
```