

Online Grocery Shop (Farm 2 Fork)

A PROJECT REPORT

Submitted for the course: Internet and Web Programming (CSE3002)

By

Jerelyn P. Premjit

18BCE2359

Slot: B2

Name of faculty: Prof. Ragunthar. T

(SCHOOL OF COMPUTER SCIENCE AND ENGINEERING)



25th OCTOBER 2020

CERTIFICATE

This is to certify that the project work entitled “*Online Grocery Shop (Farm 2 Fork)*” that is being submitted by “*Jerelyn P. Premjit (18BCE2359)*” for Internet and Web Programming (CSE3002) is a record of bonafide work done under our supervision. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted for any other course.

Place: Vellore

Date: 25th October, 2020

Signature of Student:

Jerelyn P. Premjit

Signature of Faculty:

Prof. Ragunthar.T.

ACKNOWLEDGEMENTS

We express gratitude to our guide, **Dr. T. Ragunthar**, for guidance and suggestions that helped us to complete the project on time. Words are inadequate to express our gratitude to the faculty and staff members who encouraged and supported us during the project. Finally, we would like to thank our ever-loving parents for their blessings and our friends for their timely help and support.

Signatories

Jerelyn P. Premjit

Table of Content

1. Abstract	6
2. Introduction	7
Problem Statement	7
Technical Specification	7
i. Front-End Development Languages	8
ii. Server-Side Programming	8
Advance Technologies to be used:	8
Possible tools to be used	9
iii. Eclipse IDE	9
iv. Java and server	9
v. MySQL	9
vi. Apache TomCat	9
3. Existing System Problems	10
4. Proposed System Design	10
MVC and Servlet functions	10-11
5. Modules and their Description	12
Login Module	12
Registration Module	12
OTP Verification Module	12
Shop By Category Module	12
Display Shop Category Module	12
Shopping Cart Module	12
Order Confirmation Module	12
Database Module	13
Subroutines	13
6. MySQL tables and screenshots	14-21
7. Module Implementation Screenshots	22-26
8. Conclusion	27
9. Future Work	27
10. References	28
11. Appendix 1 (Views)	29-56
index.jsp	
Register.jsp	
otp.jsp	

ShopByCategory.jsp	
Shop.jsp	
Viewcartindex.jsp	
orderconfirmation.jsp	
12. Appendix 2 (Database Integration)	57-67
DBUtil.java	
ShopUtil.java	
ValidateLogin.java	
13. Appendix 3(Servlet)	68-81
AjaxHandler.java.	
ListCartItem.java	
Login.java	
PlaceOrder.java	
Shopping.java	

1. Abstract

An online shopping system that permits a customer to submit online orders for items and/or services from a store that serves both walk-in customers and online customers. The online shopping system presents an online display of an order cut off time and an associated delivery window for items selected by the customer. The system accepts the customer's submission of a purchase order for the item in response to a time of submission being before the order cut off time. The online shopping system does not settle with a credit supplier of the customer until the item selected by the customer is picked from inventory but before it is delivered. Therefore, the customer can go online and make changes to the order.

In addition, available service windows are presented to the customer as a function of customer selected order and service types and further, the order picking is assigned in accordance with a picker's preference.

2. Introduction

When ordering goods, many shopping systems provide a virtual shopping cart for holding items selected for purchase. Successive items selected for purchase are placed into the virtual shopping cart until a customer completes their shopping trip. Virtual shopping carts may be examined at any time, and their contents can be edited or deleted at the option of the customer. Once the customer decides to submit a purchase order, the customer may print the contents of the virtual shopping basket in order to obtain a hard copy record of the transaction.

We will also be adding a special module to our project to add to its novelty and user experience termed Focused Shopper.

a. Problem Statement

Due to the Covid-19 pandemic, most of our shopping and activities are conducted in the online mode. There may be many online grocery stores but the user interface may be very complex and hard for new users and the older generation. Hence, this project aims to build the ideal online grocery shopping website using J2EE Technology.

b. Technical Specification

i. Front-End Development Languages:

Language to be used: HTML, CSS, Javascript, JSP

HTML: HTML stands for Hyper Text Markup Language. It is used to design the front end portion of web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. The markup language is used to define the text documentation within the tag which defines the structure of web pages.

CSS: Cascading Style Sheets fondly referred to as CSS is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

Javascript: JavaScript is a famous scripting language used to create the magic on the sites to make the site interactive for the user. It is used to enhance the functionality of a website to run cool games and web-based software.

Java Server Pages (JSP): Different pages in the applications are designed using jsp. A Java ServerPages component is a type of Java servlet that is designed to fulfil the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands. Using JSP, one can collect input from users through web page

ii. Backend

Language to be used: Java

Server : Apache TomCat Web server + Servlet programming

Application : J2EE Technology

Database : Connect to MySQL Database using JDBC Connectivity

c. Advance Technologies to be used

J2EE Technology

J2EE is a platform-independent, Java-centric environment from Sun for developing, Building and deploying Web-based enterprise applications online. The **J2EE** platform consists of a set of services, APIs, and protocols that provide the functionality for developing multi tiered, Web-based applications

d. Possible tools to be used:

- a) Eclipse IDE
- b) MySQL

For database management on the server side.

- c) Java installer
 - JDK 14.0
 - MySQL Community Installer
 - MYSQL Workbench
- d) **Apache Tom Cat(version 8) -**

Apache Tomcat is an open-source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and WebSocket technologies. Tomcat provides a "pure Java" HTTP web server environment in which Java code can run.

3. Existing System Problems

Due to the Covid-19 pandemic, Most of our shopping and activities are conducted in the online mode.

There may be many online grocery stores but the user interface may be very complex and hard for new users and the older generation which is resolved in our project.

Also if it is offline then there will be some problems such as

Lack of security of Data.

Time Consuming.

Needs Manual Calculation.

4. Proposed System Design

The aim of the proposed system is to develop a system of improved facilities. The proposed system can overcome the limitations of the existing system. The system reduces the manual work. The customer can go online and make changes to the order.

Our project is based on an online shopping application for a grocery shop. The project follows **MVC Architecture**, which stands for **Model View Controller**. Model is a database, View is the user interface and controller is the backend servlet.

Model : MySQL Database

View : Java Server Pages

Controller : Java Servlet

We have developed this on Java JDK14.2 and the Web Server, Apache TomCat (version 8). We used JSP, HTML, CSS, Java Beans and Servlet Coding to bring this application to life.

The Application comprises four **main controller servlets**..Login Servlet, Shopping Servlet AjaxHandler Servlet and PlaceOrder Servlet.

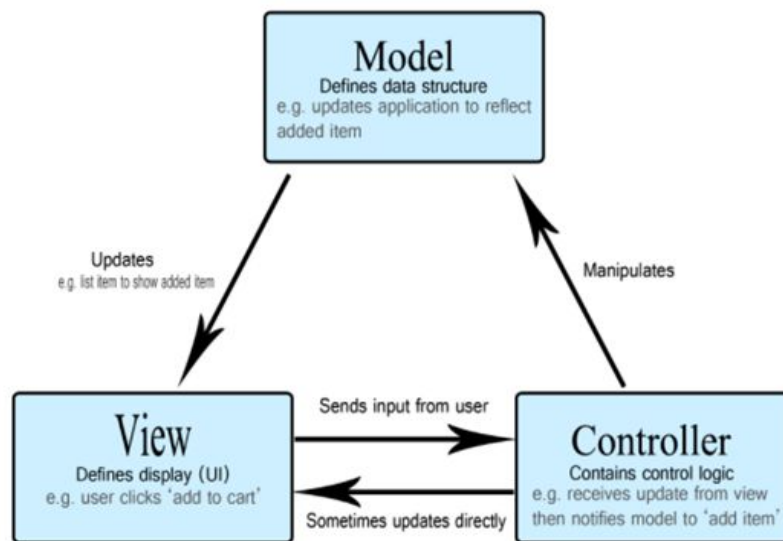
Login Servlet Authenticates the existing user and provides the facility to register new users.

The **Shopping servlet** creates a dynamic user Interface based on the shopping category

selection and allows the user to generate a dynamic shop.jsp page

The **Ajaxhandler servlet** allows the customer to add items to their shopping cart without altering the the display or behaviour of the page

The **PlaceOrder servlet** that allows the customer check out with the items in their shopping cart.



5. Modules and their Description

1) LOGIN MODULE:

This module is the login form where customers can login to the system if they have an account, else they may proceed to the next module (OTP Verification module). The page also has a 1 min session timeout.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%session.setMaxInactiveInterval(1); %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Farm2Fork</title>
</head>
```

2) REGISTRATION MODULE:

In this module a new user will have to sign up to our website. With his email address, password, address and mobile number.

This module will then redirect the user back to the login module for otp verification

3) OTP VERIFICATION MODULE:

This module begins by generating a random 4 digit number and passing it onto the ValidateLogin servlet functions to verify if the right number is entered.

Once verified, with the database and if the OTP hasn't expired the user may proceed to the ShopByCategory page.

```
while (result.next()) {
    db_u_name = result.getString(1);
    db_otp = result.getInt(2);
    db_timediff = result.getInt(3);
    System.out.print("Database Name : " + db_u_name + "\nDatabase OTP : " + db_otp);

    if (db_timediff <= 40) {
        if (otpcheck == db_otp) {
            returnvalue = "OTP verified";
            System.out.println("YAY! YOU DID IT!");
        } else {
            returnvalue = "Wrong OTP";
            System.out.println("OH NO!");
        }
    }
} else {
    returnvalue = "OTP Expired";
    System.out.println("OH NO!");
}
```

4) SHOP BY CATEGORY MODULE

In this module, we query the database table for the different categories and display them onto a single screen so that the user may easily find the items they are looking for.

5) DISPLAY SHOP CATEGORIES MODULE

We create a dynamic webpage here called shop.jsp that uses a java bean to read the product master in the database and display it onto the screen accordingly. This is done by utilizing the requestbuffer in the session.

6) SHOPPING CART MODULE

When the shopping cart is clicked on, we add all the items to the cart using a Ajax handler that utilizes a hashmap to implement the same.

We also print the total number of items ordered and the total amount to be paid in the process.

7) ORDER CONFIRMATION MODULE

Under this module we send all the values displayed on the shopping cart to the database and generate a unique order number that we then display on the final jsp page (orderConfirmed.jsp)

8) DATABASE MODULE

we execute all our sql queries on a mysql database (jdbc connectivity)

The codes regarding the same is under topic [MySQL]

9) SUBROUTINES (INTEGRATES DATABASE MODULES WITH VIEW MODULES)

The servlets with their various subroutines integrated with the database are as follows :

DBUtil.java → getMysqlConnection();

ValidateLogin.java → updateOTP, validateOTP, newRegistration(),
getCustomerDetails();//for passwordverification

Login.java →

ShopUtil.java → getOrderNumber(), addtoOrderMaster(); addOrderdetail();
getProdListbyCat()

6. MySQL tables :

```
CREATE TABLE cust_mast(  
  cust_id varchar(50) NOT NULL UNIQUE,  
  cust_name varchar(50),  
  cust_password varchar(15),  
  addressline_1 varchar(60),  
  addressline_2 varchar(60),  
  city varchar(30),  
  state varchar(20),  
  pin INT(6),  
  mobile_number varchar(20),  
  OTP_number INT(4),  
  OTP_verified char(1),  
  reg_date DATE,  
  last_login DATE  
);
```

```
insert into cust_mast values('dada@gmail.com', 'TheMeanIndianDad', 'daddy', 'Somewhere in Riyadh',  
'Near NCB Bank', 'Riyadh', 'Tamil Nadu', 600091 , '0508061480' , null , null, '2020-08-23', null );
```

```
insert into cust_mast cust_id,cust_name,cust_password,addressline_1 ,addressline_2 ,city ,state ,pin  
,mobile_number, reg_date values(?,?,?,?,?,?,?,?,?,?);
```

```
String u_id, String u_name, String u_pass, cu_pass, u_add1, String u_add2,String u_city,String  
u_state,String u_pin, String u_mobile
```

```
CREATE TABLE cat_mast(  
  cat_id char(3) NOT NULL UNIQUE,  
  cat_desc varchar(40) NOT NULL,  
  cat_status char(1) NOT NULL);
```

```
insert into cat_mast values('001','Fruit And vegetables','Y');  
insert into cat_mast values('002','Meat','Y');  
insert into cat_mast values('003','Bakery','Y');  
insert into cat_mast values('004','Dairy','Y');  
insert into cat_mast values('005','Deli','Y');  
insert into cat_mast values('006','Seafood','Y');  
insert into cat_mast values('007','Oil And Ghee','Y');  
insert into cat_mast values('008','Rice and Pasta','Y');  
insert into cat_mast values('009','Beverages','Y');  
insert into cat_mast values('010','Health and Beauty','Y');  
insert into cat_mast values('011','Snacks','Y');  
insert into cat_mast values('012','Frozen Foods','Y');
```

```
CREATE TABLE prod_mast(  
  prod_id char(5) NOT NULL UNIQUE,  
  prod_cat_id char(3) NOT NULL ,  
  prod_name varchar(20)NOT NULL,  
  prod_stock INT(4) NOT NULL,  
  prod_price_perunit FLOAT(3) NOT NULL,  
  prod_unit varchar(5) NOT NULL  
);
```

1)FRUITS AND VEGETABLES

```
insert into prod_mast values('00001','001','Apple',500 ,15.00, 'Per Kg');
insert into prod_mast values('00002','001','Blueberry',500 ,24.00, 'Per Kg');
insert into prod_mast values('00003','001','Onions',500 ,17.00, 'Per Kg');
insert into prod_mast values('00004','001','Oranges',500 ,16.00, 'Per Kg');
insert into prod_mast values('00005','001','Plum',500 ,30.00, 'Per Kg');
insert into prod_mast values('00006','001','Guava',500 ,10.00, 'Per Kg');
insert into prod_mast values('00007','001','Banana',500 ,13.00, 'Per Kg');
insert into prod_mast values('00008','001','Tomatos',500 ,10.00, 'Per Kg');
insert into prod_mast values('00009','001','Sweet Potato',500 ,15.00, 'Per Kg');
insert into prod_mast values('00010','001','Beetroot',500 ,18.00, 'Per Kg');
insert into prod_mast values('00011','001','Red Bell Pepper',500 ,19.00, 'Per Kg');
insert into prod_mast values('00012','001','Garlic',500 ,7.00, 'Per Kg');
```

2)MEAT

```
insert into prod_mast values('00013','002','Ground Beef',200 ,20.00, 'Per Kg');
insert into prod_mast values('00014','002','Beef Patties',200 ,100.00, 'Per Kg');
insert into prod_mast values('00015','002','Butchers Sausage',200 ,345.00, 'Per Kg');
insert into prod_mast values('00016','002','Beef ShortRibs',200 ,200.00, 'Per Kg');
insert into prod_mast values('00017','002','Fresh Chicken',200 ,150.00, 'Per Kg');
insert into prod_mast values('00018','002','Chicken Patties',200 ,150.00, 'Per Kg');
insert into prod_mast values('00019','002','Breaded Chicken',200 ,200.00, 'Per Kg');
insert into prod_mast values('00020','002','BBQ Seasoned Chicken',200 ,150.00, 'Per Kg');
insert into prod_mast values('00021','002','Shawarma Chicken',200 ,250.00, 'Per Kg');
insert into prod_mast values('00022','002','Chicken Kebab',200 ,110.00, 'Per Kg');
insert into prod_mast values('00023','002','Marinated Chicken',200 ,150.00, 'Per Kg');
insert into prod_mast values('00024','002','Sliced Lamb',200 ,250.00, 'Per Kg');
```

3)BAKERY

```
insert into prod_mast values('00025','003','Chocolate Cake',50 ,45.00, '1 Pc');
insert into prod_mast values('00026','003','Fruit Cake',50 ,45.00, '1 Pc ');
insert into prod_mast values('00027','003','German Bread',50 ,50.00, '1 Pc');
insert into prod_mast values('00028','003','Dinner Roll',50 ,30.00, '6 Pcs');
insert into prod_mast values('00029','003','Butter Roll',50 ,30.00, '6 Pcs');
insert into prod_mast values('00030','003','Brownies',50 ,35.00, '6 Pcs');
insert into prod_mast values('00031','003','Banana Cake',50 ,40.00, '1 Pc');
insert into prod_mast values('00032','003','Cinnamon Roll',50 ,40.00, '1 Pc');
insert into prod_mast values('00033','003','Oreo Cake',10 ,150.00, '1 Pc');
insert into prod_mast values('00034','003','CupCake',200 ,60.00, '1 Pc');
insert into prod_mast values('00035','003','Donuts',300 ,10.00, '1 Pc');
insert into prod_mast values('00036','003','Apple Pie',200 ,120.00, '1 Pc');
```

4)DAIRY

```
insert into prod_mast values('00037','004','Eggs',200 ,120.00, '6 PcS');
insert into prod_mast values('00038','004','Yogurt',200 ,20.00, '1 Pc');
insert into prod_mast values('00039','004','Greek Yogurt',200 ,70.00, '1 Pc');
insert into prod_mast values('00040','004','Milk',500 ,50.00, '1 Pc');
insert into prod_mast values('00041','004','Labanah',500 ,40.00, '1 Pc');
insert into prod_mast values('00042','004','Butter',500 ,30.00, '1 Pc');
insert into prod_mast values('00043','004','Cheese Slice',500 ,50.00, '6 Pcs');
insert into prod_mast values('00044','004','Mozarella',100 ,250.00, '1 Pc');
insert into prod_mast values('00045','004','Creme Cheese',100 ,90.00, '8 Pc');
```

5)DELI

```
insert into prod_mast values('00046','005','Chicken Mortedella',200 ,20.00, '1 Pc');
```

```

insert into prod_mast values('00047','005','Pepperoni',200 ,20.00,'30 Pcs');
insert into prod_mast values('00048','005','Black Olives',200 ,20.00,'1 Pc');
insert into prod_mast values('00049','005','Green Olives',200 ,20.00,'1 Pc');
insert into prod_mast values('00050','005','Stuffed Olives',200 ,20.00,'1 Pc');
insert into prod_mast values('00051','005','Baramili Cheese',200 ,20.00,'1 Pc');
insert into prod_mast values('00052','005','Feta Cheese',200 ,20.00,'1 Pc');

```

6)SEAFOOD

```

insert into prod_mast values('00053','006','CatFish',200 ,420.00,'Per Kg');
insert into prod_mast values('00054','006','Clams',200 ,220.00,'Per Kg');
insert into prod_mast values('00055','006','Crab',200 ,320.00,'Per Kg');
insert into prod_mast values('00056','006','Tuna',200 ,120.00,'Per Kg');
insert into prod_mast values('00057','006','Peeled Shrimp',200 ,320.00,'Per Kg');
insert into prod_mast values('00058','006','Salmon Steak',200 ,220.00,'Per Kg');
insert into prod_mast values('00059','006','Fresh Shrimp',200 ,320.00,'Per Kg');
insert into prod_mast values('00060','006','Salmon Fillet',200 ,220.00,'Per Kg');

```

7)OIL AND GHEE

```

insert into prod_mast values('00061','007','Olive Oil',200 ,150.00,'Per L');
insert into prod_mast values('00062','007','Corn Oil',200 ,170.00,'Per L');
insert into prod_mast values('00063','007','Vegetable Oil',200 ,200.00,'Per L');
insert into prod_mast values('00064','007','Sunflower Oil',200 ,250.00,'Per L');
insert into prod_mast values('00065','007','Canola Oil',200 ,150.00,'Per L');
insert into prod_mast values('00066','007','Ghee',200 ,100.00,'Per L');

```

8)RICE AND PASTA

```

insert into prod_mast values('00067','008','Basmati Rice',200 ,220.00,'Per Kg');
insert into prod_mast values('00068','008','Punjabi Rice',200 ,320.00,'Per Kg');
insert into prod_mast values('00069','008','Brown Rice',200 ,400.00,'Per Kg');
insert into prod_mast values('00070','008','Calrose Rice',200 ,320.00,'Per Kg');
insert into prod_mast values('00071','008','Penne Pasta',200 ,440.00,'Per Kg');
insert into prod_mast values('00072','008','Spaghetti',200 ,420.00,'Per Kg');
insert into prod_mast values('00073','008','Marinara Sauce',200 ,300.00,'Per Kg');
insert into prod_mast values('00074','008','Garlic Parmesan',200 ,220.00,'Per Kg');
insert into prod_mast values('00075','008','Macaroni',200 ,120.00,'Per Kg');
insert into prod_mast values('00076','008','Oats Pasta',200 ,500.00,'Per Kg');
insert into prod_mast values('00077','008','Jumbo Shell Pasta',200 ,430.00,'Per Kg');

```

9)BEVERAGES

```

insert into prod_mast values('00078','009','Ceaser AppleJuice',200 ,40.00,'1 Pc');
insert into prod_mast values('00079','009','Ceaser OrangeJuice',200 ,40.00,'1 Pc');
insert into prod_mast values('00080','009','Alrabie AppleJuice',200 ,80.00,'1 box');
insert into prod_mast values('00081','009','KDD GuavaJuice',200 ,100.00,'1 box');
insert into prod_mast values('00082','009','Red Grape Juice',200 ,70.00,'1 Pc');
insert into prod_mast values('00083','009','Mineral Water',200 ,60.00,'1 Pc');
insert into prod_mast values('00084','009','Coffee',200 ,60.00,'1 Pc');
insert into prod_mast values('00085','009','Tea',200 ,50.00,'1 Pc');

```

10)HEALTH AND BEAUTY

```

insert into prod_mast values('00086','010','Dove Soap',200 ,40.00,'1 Pc');
insert into prod_mast values('00087','010','Pears Soap',200 ,40.00,'1 Pc');
insert into prod_mast values('00088','010','Colgate Toothpaste',200 ,30.00,'1 Pc');
insert into prod_mast values('00089','010','Nivea Deodorant',200 ,150.00,'1 Pc');

```



```
insert into prod_mast values('00090','010','Always Pads',200 ,200.00,'1 Pc');
insert into prod_mast values('00091','010','MouthWash',200 ,40.00,'1 Pc');
insert into prod_mast values('00092','011','Veet Wax Strips',200 ,50.00,'1 Pc');
```

11)SNACKS

```
insert into prod_mast values('00093','011','Kit Kat',200 ,50.00,'1 Pc');
insert into prod_mast values('00094','011','Malteasers',200 ,80.00,'1 Pc');
insert into prod_mast values('00095','011','M&Ms',200 ,40.00,'1 Pc');
insert into prod_mast values('00096','011','Hersheys',200 ,40.00,'1 Pc');
insert into prod_mast values('00097','011','Dairy Milk',200 ,30.00,'1 Pc');
insert into prod_mast values('00098','011','Vanilla Wafers',200 ,90.00,'1 Pc');
insert into prod_mast values('00099','011','Oreo',200 ,100.00,'1 Box');
insert into prod_mast values('00100','011','Apple Strudel',200 ,20.00,'1 Pc');
```

12)FROZEN FOODS

```
insert into prod_mast values('00101','012','BR Vanilla',200 ,90.00,'1 Pc');
insert into prod_mast values('00102','012','BR Chocolate',200 ,90.00,'1 Pc');
insert into prod_mast values('00103','012','BR Berry',200 ,90.00,'1 Pc');
insert into prod_mast values('00104','012','Mars Icecream',200 ,110.00,'1 Pc');
insert into prod_mast values('00105','012','Frozen Raspberries',200 ,70.00,'1 Pc');
insert into prod_mast values('00106','012','French Fries',200 ,90.00,'1 Pc');
insert into prod_mast values('00107','012','Frozen Peas',200 ,70.00,'1 Pc');
insert into prod_mast values('00108','012','Frozen Corn',200 ,70.00,'1 Pc');
insert into prod_mast values('00109','012','Frozen Berries Mix',200 ,70.00,'1 Pc');
insert into prod_mast values('00110','012','Frozen Pizza',200 ,130.00,'1 Pc');
insert into prod_mast values('00111','012','Coffee Icecream',200 ,70.00,'1 Pc');
insert into prod_mast values('00112','012','Snickers Icecream',200 ,110.00,'1 Pc');
```

```
alter table prod_mast modify prod_unit varchar(8) NOT NULL;
SELECT prod_id
FROM prod_mast
WHERE prod_cat_id = (?);
```

OTP VERIFICATION.....TIMEOUT

```
select cust_id , OTP_number,TIMEDIFF(OTP_time,CURRENT_TIMESTAMP) from cust_mast
where cust_id = "jerelyn@gmail.com";
```

```
select cust_id , OTP_number from cust_mast where cust_id =
```

```
select cust_id , OTP_number,TIME_TO_SEC(TIMEDIFF(CURRENT_TIMESTAMP,OTP_time))
from cust_mast where cust_id = "jerelyn@gmail.com";
```

```
select cust_id , OTP_number,DATE_SUB(OTP_time,CURRENT_TIMESTAMP) from cust_mast
where cust_id = "jerelyn@gmail.com";
```

.....CART

```
CREATE TABLE cat_mast(
cat_id char(3) NOT NULL UNIQUE,
cat_desc varchar(40) NOT NULL,
```

```
cat_status char(1) NOT NULL);
```

```
pkup  
cod
```

```
.....order number  
CREATE TABLE order_mast(  
  order_number varchar(12) NOT NULL UNIQUE,  
  order_date DATE NOT NULL,  
  payment_type varchar(4) NOT NULL,  
  addline_1 varchar(60) NULL,  
  addline_2 varchar(60) NULL,  
  order_status varchar(20) NULL,  
  order_amount FLOAT(6,2) NOT NULL  
);
```

```
CREATE TABLE order_desc(  
  od_order_number varchar(12) NOT NULL,  
  od_prod_id char(5) NOT NULL ,  
  od_prod_cat_id char(3) NOT NULL ,  
  od_prod_name varchar(20) NOT NULL,  
  od_prod_qty INT NULL,  
  od_prod_price FLOAT(6,2) NOT NULL  
);
```

```
od_order_number,od_prod_id,od_prod_cat_id,od_prod_name,od_prod_qty,od_prod_price
```

```
order_number, order_date, payment_type, addline_1, addline_2, order_status, order_amount
```

```
SELECT cur_date, count(*)+1 from order_mast;
```

```
SELECT DATE_FORMAT(CURRENT_DATE(),"%Y%m%d"),count(*)+1 from order_mast;
```

```
SELECT UTC_DATE(), count(*)+1 from order_mast;
```

SCREENSHOTS

The first screenshot shows the MySQL 8.0 Command Line Client interface. It displays the login process, including the password prompt and the welcome message. The user enters the password and connects to the MySQL server. The user then uses the 'use shoponline;' command to switch to the 'shoponline' database. The 'show tables;' command is executed, showing a list of tables: 'Tables_in_shoponline', 'cat_mast', 'cust_mast', and 'prod_mast'. The user then executes 'desc cat_mast;', which displays the structure of the 'cat_mast' table.

```
mysql> use shoponline;
Database changed
mysql> show tables;
+-----+
| Tables_in_shoponline |
+-----+
| cat_mast              |
| cust_mast             |
| prod_mast             |
+-----+
3 rows in set (4.90 sec)

mysql> desc cat_mast;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| cat_id | char(3) | NO | PRI | NULL | |
| cat_desc | varchar(40) | NO | | NULL | |
| cat_status | char(1) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.25 sec)
```

The second screenshot shows the MySQL 8.0 Command Line Client interface. It displays the user attempting to describe the 'cust_mast' table, which results in an error: 'ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'cust_mast' at line 1'. The user then attempts to describe the 'cat_desc' table, which results in an error: 'ERROR 1146 (42S02): Table 'shoponline.cat_desc' doesn't exist'. The user then executes 'desc cust_mast;', which displays the structure of the 'cust_mast' table.

```
mysql> desc cust_mast;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'cust_mast' at line 1
mysql> desc cat_desc;
ERROR 1146 (42S02): Table 'shoponline.cat_desc' doesn't exist
mysql> desc cust_mast;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| cust_id | varchar(50) | NO | PRI | NULL | |
| cust_name | varchar(50) | YES | | NULL | |
| cust_password | varchar(15) | YES | | NULL | |
| addressline_1 | varchar(60) | YES | | NULL | |
| addressline_2 | varchar(60) | YES | | NULL | |
| city | varchar(30) | YES | | NULL | |
| state | varchar(20) | YES | | NULL | |
| pin | int | YES | | NULL | |
| mobile_number | varchar(20) | YES | | NULL | |
| OTP_number | int | YES | | NULL | |
| OTP_verified | char(1) | YES | | NULL | |
| reg_date | date | YES | | NULL | |
| last_login | date | YES | | NULL | |
| OTP_time | datetime | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
14 rows in set (0.28 sec)

mysql> desc prod_mast;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| prod_id | char(5) | NO | PRI | NULL | |
| prod_cat_id | char(3) | NO | | NULL | |
| prod_name | varchar(20) | NO | | NULL | |
| prod_stock | int | NO | | NULL | |
| prod_price_perunit | float | NO | | NULL | |
| prod_unit | varchar(8) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.19 sec)

mysql>
```

Fig 1 - Database for the login details and OTP and product details.

MySQL 8.0 Command Line Client

```
mysql> select * from prod_mast;
```

prod_id	prod_cat_id	prod_name	prod_stock	prod_price_perunit	prod_unit
00001	001	Apple	500	15	Per Kg
00002	001	Blueberry	500	24	Per Kg
00003	001	Onions	500	17	Per Kg
00004	001	Oranges	500	16	Per Kg
00005	001	Plum	500	30	Per Kg
00006	001	Guava	500	10	Per Kg
00007	001	Banana	500	13	Per Kg
00008	001	Tomatos	500	18	Per Kg
00009	001	Sweet Potato	500	15	Per Kg
00010	001	Beetroot	500	18	Per Kg
00011	001	Red Bell Pepper	500	19	Per Kg
00012	001	Garlic	500	7	Per Kg
00013	002	Ground Beef	200	20	Per Kg
00014	002	Beef Burger Patties	200	100	Per Kg
00015	002	Butchers Sausage	200	345	Per Kg
00016	002	Beef ShortRibs	200	200	Per Kg
00017	002	Fresh Chicken	200	150	Per Kg
00018	002	Chicken Patties	200	150	Per Kg
00019	002	Breaded Chicken	200	200	Per Kg
00020	002	BBQ Seasoned Chicken	200	150	Per Kg
00021	002	Shawarma Chicken	200	250	Per Kg
00022	002	Chicken Kebab	200	110	Per Kg
00023	002	Marinated Chicken	200	150	Per Kg
00024	002	Sliced Lamb	200	250	Per Kg
00025	003	Chocolate Cake	50	45	1 Pc
00026	003	Fruit Cake	50	45	1 Pc
00027	003	German Bread	50	50	1 Pc
00028	003	Dinner Roll	50	30	6 Pcs
00029	003	Butter Roll	50	30	6 Pcs
00030	003	Brownies	50	35	6 Pcs
00031	003	Banana Cake	50	40	1 Pc
00032	003	Cinnamon Roll	50	40	1 Pc
00033	003	Oreo Cake	10	150	1 Pc
00034	003	CupCake	200	60	1 Pc
00035	003	Donuts	300	10	1 Pc
00036	003	Apple Pie	200	120	1 Pc
00037	004	Eggs	200	120	6 Pcs
00038	004	Yogurt	200	20	1 Pc

Fig 2 - Database for the product details.

MySQL 8.0 Command Line Client

```
mysql> select * from cust_mast;
```

cust_id	cust_name	cust_password	addressline_1	addressline_2	city	state	pin	mobile_number	OTP_number
dada@gmail.com	TheMeanIndianDad	daddy	Somewhere in Riyadh	Near NCB Bank	Riyadh	Tamil Nadu	600091	0508061480	
jerelyn.preeja2018@vitstudent.ac.in	Jerelyn Premjit	12345	E-Block	VIT University	Vellore	Tamil Nadu	632014	8825899099	
7105	Jerelyn P. Premjit	1234	earth	somewhere	riyadh	riyadh	1234	8825899099	
7510	Neha Romie	3456	ROOM E-234	Vit University	vellore	Tamil Nadu	632014	6836488263	
8845	Shadden Premjit	2345	Room 101	baker street	Chennai	Tamil Nadu	632014	8823455678	
7327	Shadden Kurangu Preeja	1234	earth	ajdowjb	garbage	landfill	1234	8774784883	

6 rows in set (0.02 sec)

```
mysql> select * from cat_mast;
```

cat_id	cat_desc	cat_status
001	Fruit And vegetables	Y
002	Meat	Y
003	Bakery	Y
004	Dairy	Y
005	Deli	Y
006	Seafood	Y
007	Oil And Ghee	Y
008	Rice and Pasta	Y
009	Beverages	Y
010	Health and Beauty	Y
011	Snacks	Y
012	Frozen Foods	Y

12 rows in set (1.02 sec)

mysql>

Fig 3 - Database for the login details and category details.

MySQL 8.0 Command Line Client

```
mysql> use shonline;
Database changed
mysql> desc order_mast;
```

Field	Type	Null	Key	Default	Extra
order_number	varchar(12)	NO	PRI	NULL	
order_date	date	NO		NULL	
payment_type	varchar(4)	NO		NULL	
addline_1	varchar(60)	YES		NULL	
addline_2	varchar(60)	YES		NULL	
order_status	varchar(20)	YES		NULL	
order_amount	float(6,2)	NO		NULL	

7 rows in set (1.91 sec)

```
mysql> desc order_desc;
```

Field	Type	Null	Key	Default	Extra
od_order_number	varchar(12)	NO		NULL	
od_prod_id	char(5)	NO		NULL	
od_prod_cat_id	char(3)	NO		NULL	
od_prod_name	varchar(20)	NO		NULL	
od_prod_qty	int	YES		NULL	
od_prod_price	float(6,2)	NO		NULL	

6 rows in set (0.00 sec)

```
mysql> select * from order_mast;
```

order_number	order_date	payment_type	addline_1	addline_2	order_status	order_amount
20201027001	2020-10-27	cod	VIT University	Vellore	Order Placed	1860.00
202010270010	2020-10-27	cod	VIT University	Vellore	Order Placed	723.00
202010270011	2020-10-27	cod	VIT University	Vellore	Order Placed	723.00
202010270012	2020-10-27	cod	VIT University	Vellore	Order Placed	733.00
202010270013	2020-10-27	cod	VIT University	Vellore	Order Placed	419.00
202010270014	2020-10-27	cod	VIT University	Vellore	Order Placed	1379.00
20201027002	2020-10-27	cod	VIT University	Vellore	Order Placed	1860.00
20201027003	2020-10-27	cod	VIT University	Vellore	Order Placed	1860.00
20201027004	2020-10-27	cod	VIT University	Vellore	Order Placed	1860.00
20201027005	2020-10-27	cod	VIT University	Vellore	Order Placed	678.00

mysql> select * from order_desc;

od_order_number	od_prod_id	od_prod_cat_id	od_prod_name	od_prod_qty	od_prod_price
20201027001	00073	008	Marinara Sauce	1	300.00
20201027001	00027	003	German Bread	1	50.00
20201027001	00104	012	Mars Icecream	1	110.00
20201027001	00023	002	Marinated Chicken	1	150.00
20201027001	00068	008	Punjabi Rice	1	320.00
20201027001	00025	003	Chocolate Cake	1	45.00
20201027001	00026	003	Fruit Cake	1	45.00
20201027001	00069	008	Brown Rice	1	400.00
20201027001	00071	008	Penne Pasta	1	440.00
20201027002	00073	008	Marinara Sauce	1	300.00
20201027002	00027	003	German Bread	1	50.00
20201027002	00104	012	Mars Icecream	1	110.00
20201027002	00023	002	Marinated Chicken	1	150.00
20201027002	00068	008	Punjabi Rice	1	320.00
20201027002	00025	003	Chocolate Cake	1	45.00
20201027002	00026	003	Fruit Cake	1	45.00
20201027002	00069	008	Brown Rice	1	400.00

Fig 4&5 : Ordermaster and Order Description Pages

7. Module Implementation Screenshots:

Fig -1 - Login page

The image displays two screenshots of the Farm2Fork web application interface. The top screenshot shows the 'Sign Up' page, which includes a registration form with fields for Email, Full Name, Password, Confirm Password, Address Line 1, Address Line 2, Mobile Number, City, State, and PIN. A red 'SUBMIT' button and a 'Sign In' link are located at the bottom. The bottom screenshot shows the 'Sign In' page, which includes a login form with fields for Email and Password, a red 'LOG IN' button, and a 'Sign up' link for users without an account. Both pages feature a background image of corn cobs and the Farm2Fork logo.

Fig -2 - Sign Up Page

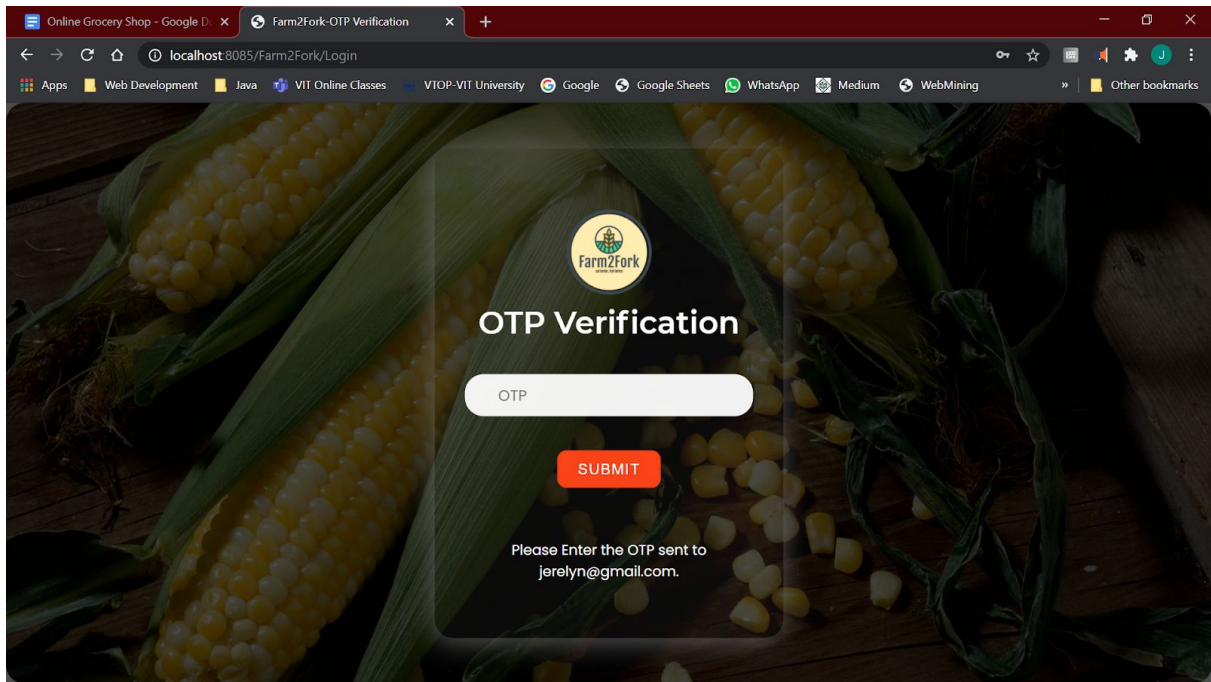
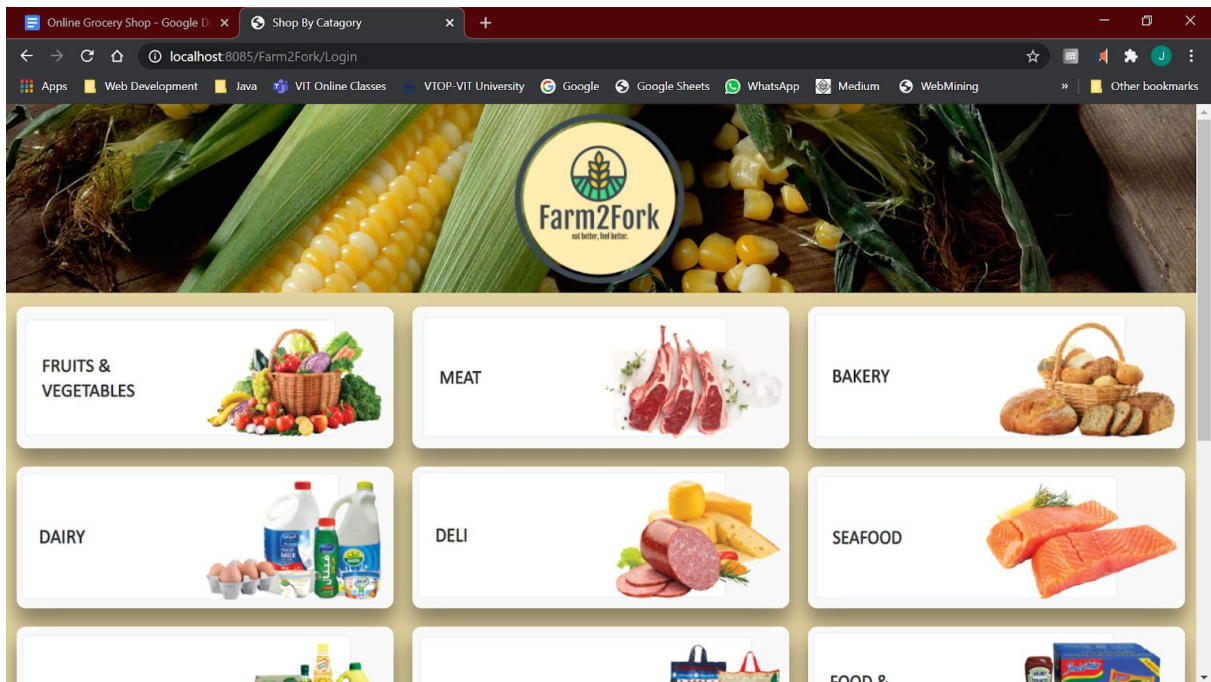


Fig -3 - OTP Verification



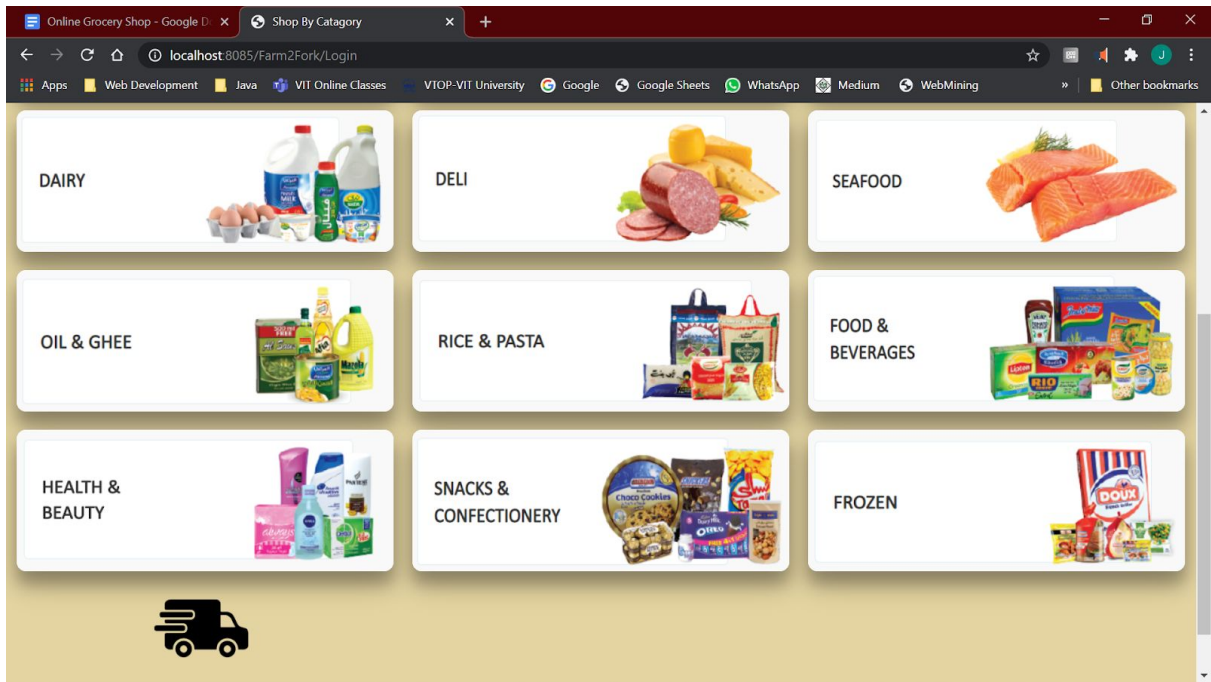
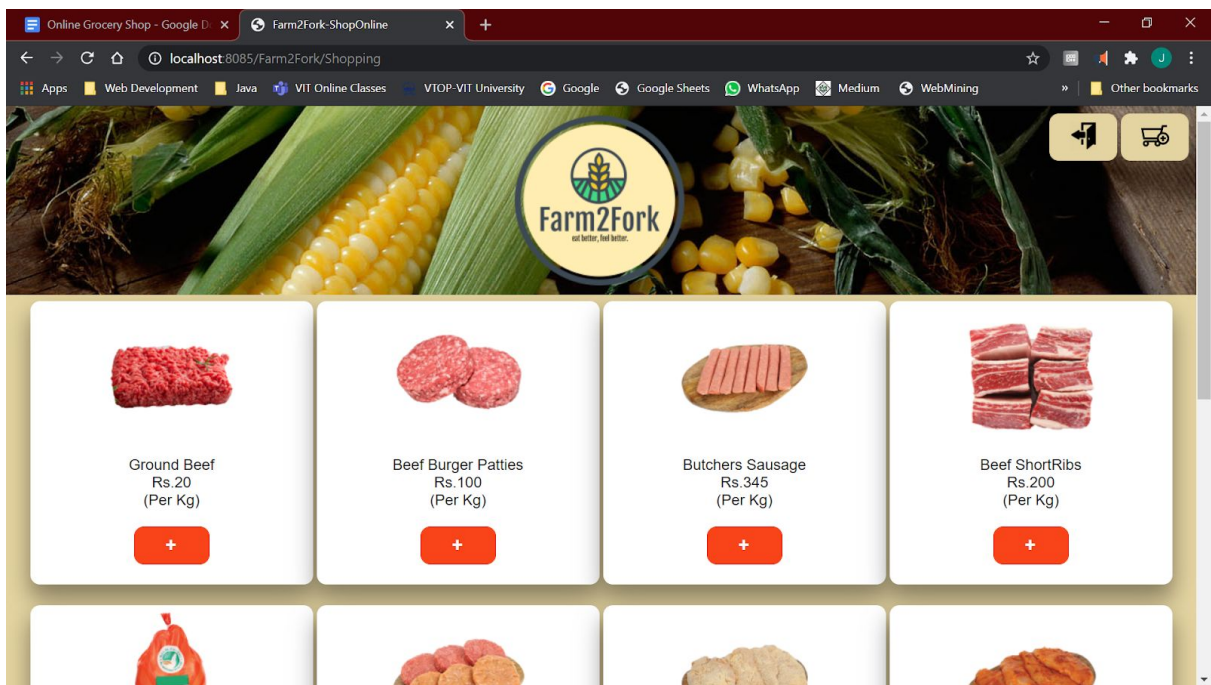


Fig 4 - All the categories of products available are listed here so that customers can add in the cart.



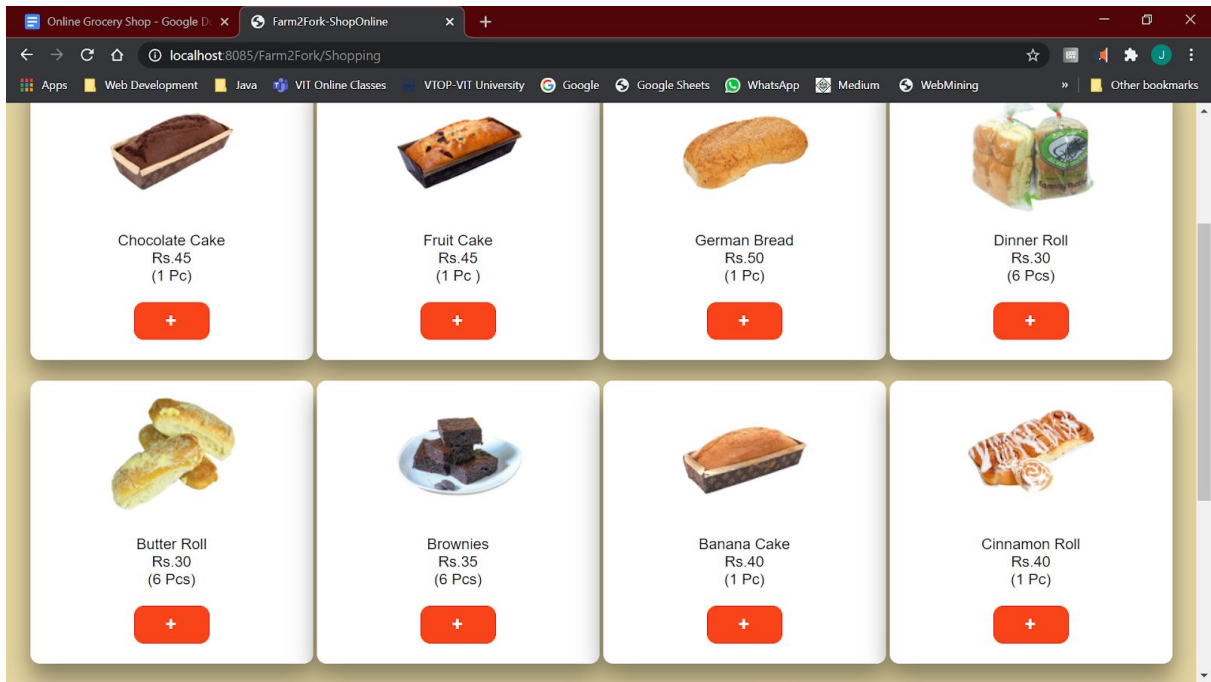


Fig 5 - After selecting the particular category we are directed to the products and now we can add to our cart.

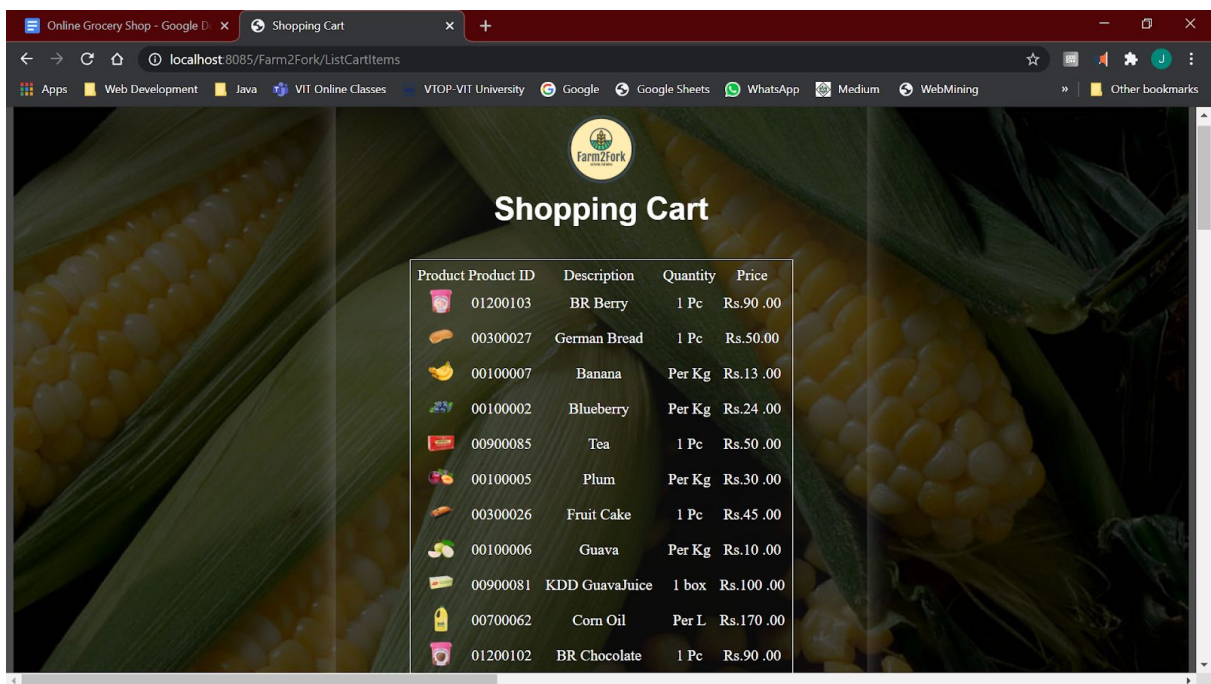


Fig 6 - All the products bought by the customer is shown with all the details.

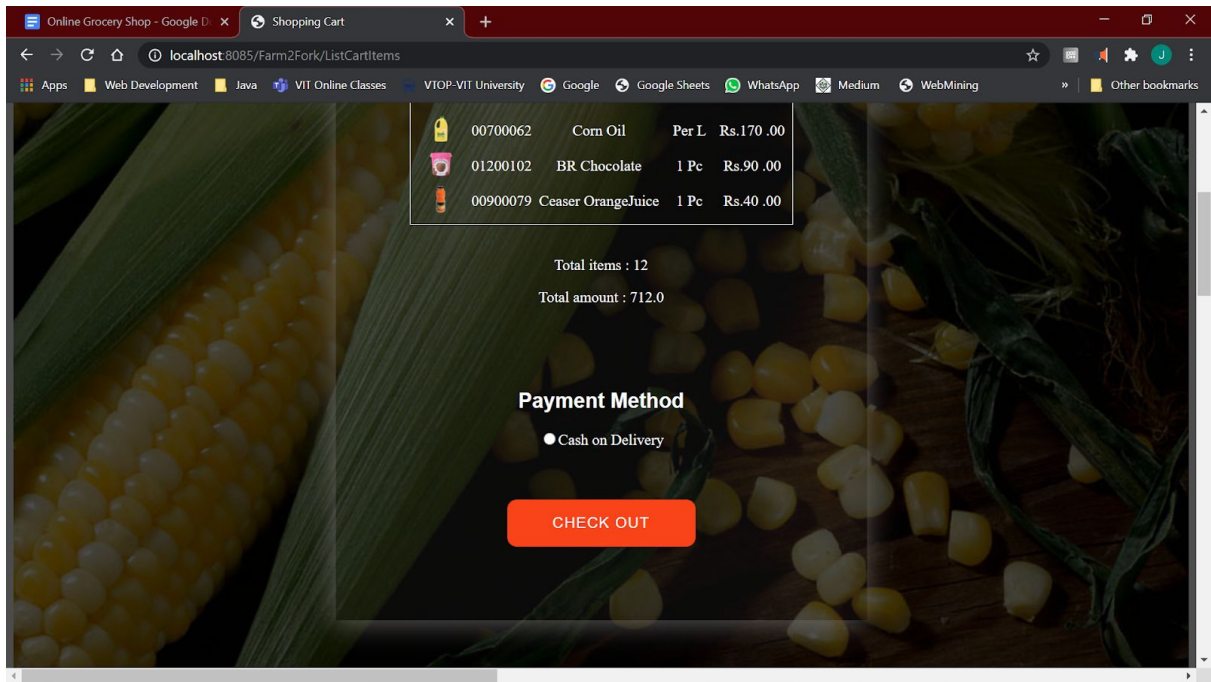


Fig 7 - Payment method will be provided to the customer either Pick up from store or cash on delivery.

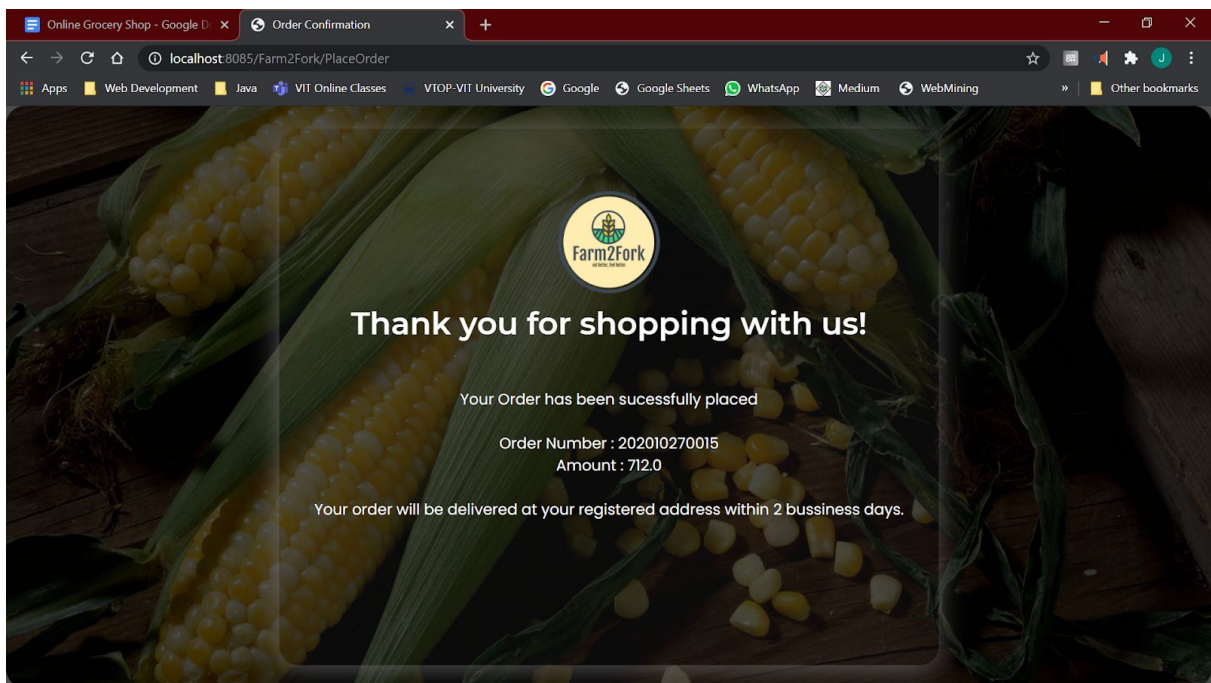


Fig 8 - The total amount and the order number is displayed .

8. Conclusion

The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application for purchasing items from a shop.

This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using HTML5 & CSS, usage of responsive templates, , and management of databases using MySQL, AJAX, servlet programming and various other technologies . The entire system is secured.

This project has given us great satisfaction in having designed a website which can be implemented to any nearby shops or branded shops selling various kinds of products by simple modifications.

9. Future Work

There is a scope for further development in our project to a great extend. A number of features can be added to this system in future like providing moderators more control over products so that each moderator can maintain their own products. Another feature we wished to implement was providing classes for customers so that different offers can be given to each class. System may keep track of the history of purchases of each customer and provide suggestions based on their history. These features could have been implemented unless the time did not limit us.

10. References

- <https://dev.mysql.com/doc/refman/8.0/en/programs-overview.html#:~:text=The%20MySQL%20server%2C%20mysqld%2C%20is%20the%20main%20program,assist%20you%20in%20starting%20and%20stopping%20the%20server%3A>
- <https://developer.mozilla.org/en-US/docs/Learn/Server-side>
- <https://www.javatpoint.com/java-bean>
- <https://www.geeksforgeeks.org/server-side-client-side-programming/#:~:text=Server%20side%20and%20Client%20side%20Programming%201%20Interact,data%20to%20server%205%20Send%20request%20to%20server>
- Syntax help (for all languages): <https://www.w3schools.co>

12. Appendix 1

index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <%session.setMaxInactiveInterval(1); %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Farm2Fork</title>
<script type="text/javascript">

                                function submitform() {

                                alert("registrationa is selected");
                                document.getElementById("command").value =

'register';

                                alert("Form will be submitted");
                                document.forms["frm"].submit();

                                }

</script>
</head>
<body>
<script>
<%!String returnvalue = "";%>
<%

                                returnvalue = "";

if (request.getAttribute("LoginStatus") != null) {
    if
(request.getAttribute("LoginStatus").equals("Invalid")) {
        returnvalue = "Password Incorrect";
    } else {
        returnvalue = "User Not Found";
    }
}
%>
</script>
<style>
    @import
url('https://fonts.googleapis.com/css?family=Montserrat:400,500,600,700|Poppins:400,500&display=
swap');
    *{
        margin: 0;
        padding: 0;
        box-sizing: border-box;
        user-select: none;
        border-radius: 20px;
```

```

}

.bg-img{
background-image: url("images/corn.jpg ");
height: 100vh;
background-size: cover;
background-position: center;
}
.bg-img:after{
position: absolute;
content: "";
top: 0;
left: 0;
height: 100%;
width: 100%;
background: rgba(0,0,0,0.7);
}
.content{
position: absolute;
top: 50%;
left: 50%;
z-index: 999;
text-align: center;
padding: 60px 32px;
width: 370px;
transform: translate(-50%,-50%);
background: rgba(247, 244, 244, 0.04);
box-shadow: -1px 4px 28px 0px rgba(121, 118, 118, 0.75);
}
.content header{
color: white;
font-size: 33px;
font-weight: 600;
margin: 0 0 35px 0;
font-family: 'Montserrat',sans-serif;
}
.field{
position: relative;
height: 45px;
width: 100%;
display: flex;
background: rgba(255,255,255,0.94);
}
.field span{
color: #222;
width: 40px;
line-height: 45px;
}
.field input{
height: 100%;
width: 100%;
background: transparent;
border: none;
outline: none;
color: #222;

```

```

    font-size: 16px;
    font-family: 'Poppins',sans-serif;
}
.space{
    margin-top: 16px;
}
i span{
    margin-left: 8px;
    font-weight: 500;
    letter-spacing: 1px;
    font-size: 16px;
    font-family: 'Poppins',sans-serif;
}
button {
width : 110px;
height : 40px;

justify-content: center;
border-radius: 10px;
border: 1px solid #f53b1a;
background-color:#f84418;
color: #FFFFFFF;
font-size: 16px;
letter-spacing: 1px;
text-transform: uppercase;
transition: transform 80ms ease-in;

}
.mess{
font-family: 'Poppins',sans-serif;

color: #FFFFFFF;
font-size: 15px;

}
.logo{
height : 120px;
width : 120px;
}

a{

color: #E4D4A1;

}
</style>

```

```

<div class="bg-img">

<div class="content">

<header>Sign In</header>
<br>
<form action="Login" id="frm" name="frm" method="post">
<div class="field">
<span class="fa fa-user"></span>
<input type="text" name="uname" required placeholder="Email"></div>

<br>
<div class="field">
<span class="fa fa-user"></span>

```

```

        <input type="password" name="upass" required placeholder="Password"></div>
        <br>
        <br>
name="uReged">LOG IN</button>

        <button type="submit"
        <br>
        <br>
        <div class="signup" >
            <p class="mess">Don't have an
account?
            <a href="Register.jsp" onclick="submitform()" name="uReged">Sign up</a></p>
        </div>

        <input type="hidden" value=""
id="command" name="command"/><br>

        <%=returnvalue%>
        </form>
        </div>

    </body>
</html>

```

Register.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Farm2Fork</title>
<script>

        function submitform1() {

            alert("registration is selected");
            document.getElementById("command").value =

'signup';

            alert("Form will be submitted");
            document.forms["frm1"].submit();

        }

</script>
</head>
<body>

        <script>

        <%!String returnvalue = "";%>

        <%returnvalue = "";

if (request.getAttribute("LoginStatus") != null) {

```



```

        if
(request.getAttribute("LoginStatus").equals("Invalid")) {
            returnvalue = "Password Incorrect";
        } else {
            returnvalue = "User Not Found";
        }
    }%>

</script>
<style>

@import
url('https://fonts.googleapis.com/css?family=Montserrat:400,500,600,700|Poppins:400,500&display=
swap')

;

* {

    margin: 0;
    padding: 0;
    box-sizing: border-box;
    user-select: none;
    border-radius: 10px;

}

.bg-img {

    background-image: url("images/corn.jpg ");
    height: 100vh; background-size : cover;
    background-position: center;
    background-size: cover;

}

.bg-img:after {

    position: absolute;
    content: "";
    top: 0;
    left: 0;
    height: 100%;
    width: 100%;
    background: rgba(0, 0, 0, 0.7);

}

.content {

    position: absolute;
    top: 50%;
    left: 50%;
    z-index: 999;
    text-align: center;
    padding: 60px 32px;
    width: 800px;
    transform: translate(-50%, -50%);
    background: rgba(247, 244, 244, 0.04);
    box-shadow: -1px 4px 28px 0px rgba(121, 118, 118,
0.75);
}

```

```

.content header {
    color: white;
    font-size: 33px;
    font-weight: 600;
    margin: 0 0 35px 0;
    font-family: 'Montserrat', sans-serif;
}

.field {
    position: relative;
    height: 40px;
    width: 100%;
    display: flex;
    background: rgba(255, 255, 255, 0.94);
    margin: 5px;
}

.field span {
    color: #222;
    width: 40px;
    line-height: 45px;
}

.field input {
    height: 100%;
    width: 100%;
    background: transparent;
    border: none;
    outline: none;
    color: #222;
    font-size: 16px;
    font-family: 'Poppins', sans-serif;
}

.space {
    margin-top: 16px;
}

i span {
    margin-left: 8px;
    font-weight: 500;
    letter-spacing: 1px;
    font-size: 16px;
    font-family: 'Poppins', sans-serif;
}

button {
    width: 110px;
    height: 40px;
    justify-content: center;
    border-radius: 10px;
    border: 1px solid #f53b1a;
    background-color: #f84418;
    color: #FFFFFF;
    font-size: 16px;
}

```

```

        letter-spacing: 1px;
        text-transform: uppercase;
        transition: transform 80ms ease-in;
    }

    .mess {

        font-family: 'Poppins', sans-serif;
        color: #FFFFFFF;
        font-size: 15px;

    }

    .logo {

        height: 120px;
        width: 120px;

    }

    a {

        color: #E4D4A1;

    }
</style>

<div class="bg-img">

    <div class="content">
         <br>

        <header>Sign Up</header>

        <form action="Login" id="frm1"
name="frm1" method="post">

            <table style="width: 100%">
                <tr>
                    <td>
                        <div
class="field">

                            <span class="fa fa-user"></span> <input type="email" name="uemail"
                                required placeholder="Email">
                        </div>
                    </td>
                    <td>
                        <div
class="field">

                            <span class="fa fa-user"></span> <input type="text" name="fname"
                                required placeholder="Full Name">
                        </div>
                    </td>
                </tr>
            </table>

```

```

class="field">
<span class="fa fa-user"></span> <input type="password" name="u_pass"
required placeholder="Password">
</div>
</td>
<td>
<div
class="field">
<span class="fa fa-user"></span> <input type="password" name="cu_pass"" required
placeholder="Confirm Password">
</div>
</td>
</tr>
</table>
<table style="width: 100%">
<tr>
<div class="field">
<span
class="fa fa-user"></span> <input type="text" name="add1 "
required placeholder="Address Line 1">
</div>
</tr>
</table>
<table style="width: 100">
<tr>
<div class="field">
<span
class="fa fa-user"></span> <input type="text" name="add2"
required placeholder="Address Line 2">
</div>
</tr>
<tr>
<td>
<div
class="field">
<span class="fa fa-user"></span> <input type="text" name="uphone"
required placeholder="Mobile Number">
</div>
</td>
<td>

```


Otp.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Farm2Fork-OTP Verification</title>
</head>
<body>
<script>
    function submitotp() {

                                                                    alert("OTP is submitted");

        document.getElementById("command").value='otp';

                                                                    alert("Form will be submitted");
                                                                    document.forms["frmotp"].submit();

                                                                    }

</script>
<style>
    @import
url('https://fonts.googleapis.com/css?family=Montserrat:400,500,600,700|Poppins:400,500&display=
swap');
    *{
        margin: 0;
        padding: 0;
        box-sizing: border-box;
        user-select: none;
        border-radius: 20px;
    }

    .bg-img{
        background-image: url("images/corn.jpg ");
        height: 100vh;
        background-size: cover;
        background-position: center;
    }
    .bg-img:after{
        position: absolute;
        content: ";
        top: 0;
        left: 0;
        height: 100%;
```

```

width: 100%;
background: rgba(0,0,0,0.7);
}
.content{
position: absolute;
top: 50%;
left: 50%;
z-index: 999;
text-align: center;
padding: 60px 32px;
width: 370px;
transform: translate(-50%,-50%);
background: rgba(247, 244, 244, 0.04);
box-shadow: -1px 4px 28px 0px rgba(121, 118, 118, 0.75);
}
.content header{
color: white;
font-size: 33px;
font-weight: 600;
margin: 0 0 35px 0;
font-family: 'Montserrat',sans-serif;
}
.field{
position: relative;
height: 45px;
width: 100%;
display: flex;
background: rgba(255,255,255,0.94);
}
.field span{
color: #222;
width: 40px;
line-height: 45px;
}
.field input{
height: 100%;
width: 100%;
background: transparent;
border: none;
outline: none;
color: #222;
font-size: 16px;
font-family: 'Poppins',sans-serif;
}
.space{
margin-top: 16px;
}
i span{
margin-left: 8px;
font-weight: 500;
letter-spacing: 1px;
font-size: 16px;
font-family: 'Poppins',sans-serif;
}
button {

```

```

width : 110px;
height : 40px;

justify-content: center;
border-radius: 10px;
border: 1px solid #f53b1a;
background-color:#f84418;
color: #FFFFFFF;
font-size: 16px;
letter-spacing: 1px;
text-transform: uppercase;
transition: transform 80ms ease-in;

}
.mess{
font-family: 'Poppins',sans-serif;

color: #FFFFFFF;
font-size: 15px;

}
.logo{
height : 100px;
width : 100px;
}
</style>

<div class="bg-img">

<div class="content">

<header>OTP Verification</header>
<form action="Login" id="frmotp" name ="frmotp" method="post">
<div class="field">
<span class="fa fa-user"></span>
<input type="text" name="uotp" required placeholder="OTP"></div>
<br>
<br>

<button
onclick="submitotp()">SUBMIT</button>

<br>
<br>
<br>
<input type="hidden" value=""
id="command" name="command"/><br>
OTP sent to <%=session.getAttribute("uname")%>.</p>

</form>
</div>

</body>
</html>

```


ShopByCategory.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <%session.setMaxInactiveInterval(1); %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Farm2Fork</title>
<script type="text/javascript">

                                function submitform() {

                                    alert("registration is selected");

document.getElementById("command").value = 'register';

                                    alert("Form will be submitted");
                                    document.forms["frm"].submit();

                                }

</script>
</head>
<body>
<script>
    <%!String returnvalue = "";%>
    <%
                                returnvalue = "";
        if (request.getAttribute("LoginStatus") != null) {
            if
(request.getAttribute("LoginStatus").equals("Invalid")) {
                                returnvalue = "Password Incorrect";
            } else {
                                returnvalue = "User Not Found";
            }
        }
    %>
</script>
<style>
    @import
url('https://fonts.googleapis.com/css?family=Montserrat:400,500,600,700|Poppins:400,500&display=
swap');
    *{
        margin: 0;
        padding: 0;
        box-sizing: border-box;
        user-select: none;
        border-radius: 20px;
    }
```

```

.bg-img{
  background-image: url("images/corn.jpg ");
  height: 100vh;
  background-size: cover;
  background-position: center;
}
.bg-img:after{
  position: absolute;
  content: "";
  top: 0;
  left: 0;
  height: 100%;
  width: 100%;
  background: rgba(0,0,0,0.7);
}
.content{
  position: absolute;
  top: 50%;
  left: 50%;
  z-index: 999;
  text-align: center;
  padding: 60px 32px;
  width: 370px;
  transform: translate(-50%,-50%);
  background: rgba(247, 244, 244, 0.04);
  box-shadow: -1px 4px 28px 0px rgba(121, 118, 118, 0.75);
}
.content header{
  color: white;
  font-size: 33px;
  font-weight: 600;
  margin: 0 0 35px 0;
  font-family: 'Montserrat',sans-serif;
}
.field{
  position: relative;
  height: 45px;
  width: 100%;
  display: flex;
  background: rgba(255,255,255,0.94);
}
.field span{
  color: #222;
  width: 40px;
  line-height: 45px;
}
.field input{
  height: 100%;
  width: 100%;
  background: transparent;
  border: none;
  outline: none;
  color: #222;
  font-size: 16px;
}

```

```

    font-family: 'Poppins',sans-serif;
}
.space{
    margin-top: 16px;
}
i span{
    margin-left: 8px;
    font-weight: 500;
    letter-spacing: 1px;
    font-size: 16px;
    font-family: 'Poppins',sans-serif;
}
button {
width : 110px;
height : 40px;

justify-content: center;
border-radius: 10px;
border: 1px solid #f53b1a;
background-color:#f84418;
color: #FFFFFFF;
font-size: 16px;
letter-spacing: 1px;
text-transform: uppercase;
transition: transform 80ms ease-in;
}
.mess{
font-family: 'Poppins',sans-serif;

color: #FFFFFFF;
font-size: 15px;
}
.logo{
height : 120px;
width : 120px;
}
a{

color: #E4D4A1;
}
</style>

<div class="bg-img">

<div class="content">

<header>Sign In</header>
<br>
<form action="Login" id="frm" name ="frm" method="post">
<div class="field">
<span class="fa fa-user"></span>
<input type="text" name="uname" required placeholder="Email"></div>

<br>
<div class="field">
<span class="fa fa-user"></span>
<input type="password" name="upass" required placeholder="Password"></div>

```

```

        <br>
        <br>
name="uReged">LOG IN</button>

        <button type="submit"
        <br>
        <br>
        <div class="signup" >
            <p class="mess">Don't have
an account?
            <a href="Register.jsp" onclick="submitform()" name="uReged">Sign up</a></p>
        </div>

        <input type="hidden" value=""
id="command" name="command"/><br>

        <%=returnValue%>
        </form>
        </div>

</body>
</html>

```

Shop.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@page import="java.util.ArrayList" %>
<%@page import="com.bean.ProductListByCat" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Farm2Fork-ShopOnline</title>
<script type="text/javascript">

function viewCart(){

    alert("in view cart");
    alert("Form will be submitted");
    document.forms["frmadd"].submit();

}

function addToCart(imgname,pcat,pid,pname,punit, uprice)
{
    var querystring =
"&imgname="+imgname+"&pcat="+pcat+"&pid="+pid+"&pname="+pname+"&punit="+punit+"&u
price="+uprice;

    alert("Form will be Santhosh" + querystring);
    console.log(querystring);

    httpRequest = new XMLHttpRequest();

```

```

instance');

'AjaxHandler?option=add&'+querystring);

== XMLHttpRequest.DONE)

step4");

document.getElementById('loading').innerHTML = "Adding to Cart";

(httpRequest.status == 200)

in step5");

array = httpRequest.response;

document.getElementById('loading').innerHTML = "Added to Cart";

alert(array)

console.log(array);

alert("Something went wrong..!!");

console.log('Something went wrong..!!');

}

</script>
<style>
body {

background:#E4D4A1;
display: flex;
justify-content: center;

```

```

    align-items: center;
    flex-direction: column;
    font-family: 'Montserrat', sans-serif;
    margin: -0px 0 50px;
}

.container {
    background:#E4D4A1;
    border-radius: 10px;
    box-shadow: 0 14px 28px rgba(0,0,0,0.25),
                0 14px 20px rgba(0,0,0,0.22);
    position : center;
    overflow: hidden;
    width: 300px;
    min-height: 300px;
}

.main-header {
    align-text:center;
}

button {
    height : auto;
    width : auto;

    justify-content: center;
    border-radius: 10px;

    color: #FFFFFFF;
    font-size: 10px;
    font-weight: bold;
    padding: 8px 20px 8px 20px;
    letter-spacing: 1px;
    text-transform: uppercase;
    transition: transform 80ms ease-in;
    border: 1px solid #E4D4A1;
    background-color:#E4D4A1;
}

.banner{
    position : relative;
    top : 0;
    justify-content: center;
    object-fit:cover;
}

.logo{
    position : absolute;
    top : 0px;
    left : 530px;
    height : 200px;
    width : 200px;
}

```

```

.add{
height : 40px;
width : 80px;
font-size: 20px;
padding: 2px 2px 2px 2px;

border: 1px solid #f53b1a;
background-color:#f84418;

}

.menu{
position : absolute;
top : 0px;
right:0px;;
align-text:center;
height:auto;
width:auto;
margin : 5px;
}

</style>
</head>
<body>

<div class="main-header" role="banner">
  
  
</div>
<div type="submit" class="addtocart" role="button">

<table class="menu">

<td>
  <button name = "logOut" value = "logout">
    
  </button>
</td>

<td>
<form name ="frmadd" action="ListCartItem" method="post">
  <button onclick="viewCart()" name = "viewCart" value = "view cart">
    
  </button></form>
</td>

<td>
  <button name = "CusDetails" value = "details">
    
  </button>
</td>
</table>

```



```
<%} %>
```

```
<%} %>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

viewcartindex.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```
    pageEncoding="ISO-8859-1"%>
```

```
    <%@page import="java.util.*" %>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="ISO-8859-1">
```

```
<title>Shopping Cart</title>
```

```
</head>
```

```
<body>
```

```
<style>
```

```
body{
```

```
color : white;
```

```
}
```

```
    .bg-img {
```

```
        background-image: url("images/corn.jpg ");
```

```
        height: 200vh;
```

```
        background-size: cover;
```

```
        background-position: center;
```

```
    }
```

```
    .bg-img:after{
```

```
        position: absolute;
```

```
        content: ";
```

```
        top: 0;
```

```
        left: 0;
```

```
        height: 500vh;
```

```
        width: 500vh;
```

```
        background: rgba(0,0,0,0.7);
```

```
    }
```

```
    .content{
```

```
        position: absolute;
```

```

top: 50%;
left: 50%;
z-index: 999;
text-align: center;
padding: 60px 32px;
width: 500px;
transform: translate(-50%,-50%);
background: rgba(247, 244, 244, 0.04);
box-shadow: -1px 4px 28px 0px rgba(121, 118, 118, 0.75);
}

```

```

.banner{
position : relative;
top : 0;
justify-content: center;
object-fit:cover;
}

```

```

.logo{
position : absolute;
top : 0px;
left : 530px;
height : 200px;
width : 200px;
}

```

```

.content header{
color: white;
font-size: 33px;
font-weight: 600;
margin: 0 0 35px 0;
font-family: 'Montserrat',sans-serif;
}

```

```

.content header1{
color: white;
font-size: 22px;
font-weight: 600;
margin: 0 0 35px 0;
font-family: 'Montserrat',sans-serif;
}

```

```

.centertable{
margin-left:auto;
margin-right:auto;
border : 1px solid white;
border-collapse : seperate;
padding: 5px;
}

```

```

button {
width : 200px;
height : 50px;

justify-content: center;
border-radius: 10px;
border: 1px solid #f53b1a;
background-color:#f84418;
color: #FFFFFF;
font-size: 16px;
letter-spacing: 1px;
text-transform: uppercase;
transition: transform 80ms ease-in;

}

</style>

<%!String message=""; %>
<%

HashMap map = new HashMap();
session.getAttribute("cart");
if(session.getAttribute("cart")!=null)
{
map =(HashMap)session.getAttribute("cart");
}
else
{
message="Product is not aviable in the cart";
}

if(map.size()==0)
{
message="Product is not aviable in the cart";
}

System.out.println("Message is as : " + message);
if(message.trim().length()<0)
{
%>

<%} else{ %>
<div class="bg-img">
<div class="content">
<caption >
<div class="main-header" role="banner">
<br>

```



```

String key = enry.getKey();
String value= enry.getValue();
String value_array [] = value.split("~");
tot = tot+Float.parseFloat(value_array[4]);
System.out.println("the sum is" + tot);

%>
<tr>
<td></td>
<td><%=value_array[1] %><%=value_array[0] %></td>
<td><%=value_array[2] %></td>
<td><%=value_array[3] %></td>
<td>Rs.<%=value_array[4] %>.00</td>

</tr>

<%
}
}%>

</table>
<center>
<br>

<p>Total items : <%=map.size()%></p>
<p>Total amount : <%=tot %></p>
<br>
<br>
<br>
<br>
<form action="PlaceOrder" id="orderfrm" name="orderfrm" method="post">

<header1>Payment Method</header1><br><br>
<input type="radio" name="mode" value="1">Cash on Delivery</label><br><br><br>
<input type="hidden" value="" id="command" name="command"/><br>
<button type="submit">CHECK OUT</button>

</form>

<br>
</center>

</div>
</div>

</body>
</html>

```

orderconfirmation.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Order Confirmation</title>
</head>
<body>

    <style>

    @import
    url('https://fonts.googleapis.com/css?family=Montserrat:400,500,600,700|Poppins:400,500&
    display=swap')
    ;

    body {

        color: white;
        font-family: 'Poppins', sans-serif;

    }

    * {

        margin: 0;
        padding: 0;
        box-sizing: border-box;
        user-select: none;
        border-radius: 20px;

    }

    .bg-img {

        background-image: url("images/corn.jpg ");
        height: 100vh;
        background-size: cover;
        background-position: center;

    }

    .bg-img:after {

        position: absolute;
        content: "";
        top: 0;
        left: 0;
        height: 100%;
        width: 100%;
        background: rgba(0, 0, 0, 0.7);

    }

    .content {
```

```

    position: absolute;
    top: 50%;
    left: 50%;
    z-index: 999;
    text-align: center;
    padding: 60px 32px;
    width: 700px;
    transform: translate(-50%, -50%);
    background: rgba(247, 244, 244, 0.04);
    box-shadow: -1px 4px 28px 0px rgba(121, 118,
118, 0.75);
}

.content header {

    color: white;
    font-size: 33px;
    font-weight: 600;
    margin: 0 0 25px 0;
    font-family: 'Montserrat', sans-serif;
}

.field {

    position: relative;
    height: 45px;
    width: 100%;
    display: flex;
    background: rgba(255, 255, 255, 0.94);
}

.field span {

    color: #222;
    width: 40px;
    line-height: 45px;
}

.field input {

    height: 100%;
    width: 100%;
    background: transparent;
    border: none;
    outline: none;
    color: #222;
    font-size: 16px;
    font-family: 'Poppins', sans-serif;
}

.space {

    margin-top: 16px;
}

```

```

i span {
    margin-left: 8px;
    font-weight: 500;
    letter-spacing: 1px;
    font-size: 16px;
    font-family: 'Poppins', sans-serif;
}

button {
    width: 110px;
    height: 40px;
    justify-content: center;
    border-radius: 10px;
    border: 1px solid #f53b1a;
    background-color: #f84418;
    color: #FFFFFF;
    font-size: 16px;
    letter-spacing: 1px;
    text-transform: uppercase;
    transition: transform 80ms ease-in;
}

.mess {
    font-family: 'Poppins', sans-serif;
    color: #FFFFFF;
    font-size: 15px;
}

.logo {
    height: 120px;
    width: 120px;
}

a {
    color: #E4D4A1;
}
</style>

<div class="bg-img">
    <center>
        <div class="content">
            
            <header>Thank you for
shopping with us!</header>
            <br>
            <p>Your Order has been
            <br>
            <p>
sucessfully placed</p>

```


	Order Number :
<%=session.getAttribute("orderno")%></p>	<p>
	Amount :
<%=session.getAttribute("tot")%></p>	
delivered at your registered address	<p>Your order will be
days.</p> 	within 2 bussiness
	</div>
	</center>
	</div>
</body>	
</html>	

Appendix 2

JAVA BEAN

```

package com.bean;

public class ProductListByCat {
    private String prod_id;
    private String prod_cat_id;
    private String prod_name;
    private String prod_stock;
    private String prod_price_perunit;
    private String prod_unit;

    public String getProd_id() {
        return prod_id;
    }
    public void setProd_id(String prod_id) {
        this.prod_id = prod_id;
    }
    public String getProd_cat_id() {
        return prod_cat_id;
    }
}

```

```

        public void setProd_cat_id(String prod_cat_id) {
            this.prod_cat_id = prod_cat_id;
        }
        public String getProd_name() {
            return prod_name;
        }
        public void setProd_name(String prod_name) {
            this.prod_name = prod_name;
        }
        public String getProd_stock() {
            return prod_stock;
        }
        public void setProd_stock(String prod_stock) {
            this.prod_stock = prod_stock;
        }
        public String getProd_price_perunit() {
            return prod_price_perunit;
        }
        public void setProd_price_perunit(String prod_price_perunit) {
            this.prod_price_perunit = prod_price_perunit;
        }
        public String getProd_unit() {
            return prod_unit;
        }
        public void setProd_unit(String prod_unit) {
            this.prod_unit = prod_unit;
        }
    }
}

```

DBUtil.java for database connectivity

```

package com.util;

import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.SQLException;

public class DBUtil {

    public static Connection getMySqlConnection()
    {
        Connection con=null;
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("Driver is loaded");

            con =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/shoponline", "root",

```

```
"775MysqlRoot775");
```

```
        if(con!=null)
        {
            System.out.println("Connection is created..");
        }
        else
        {
            System.out.println("Connection createion
failed.....");
        }

    }
    catch(ClassNotFoundException e)
    {
        e.printStackTrace();
    }
    catch (SQLException e) {
        // TODO: handle exception
    }
    catch (Exception e) {
        // TODO: handle exception
    }

    return con;
}
}
```

Taking items from database ShopUtil.java

```
package com.util;
import java.sql.*;
import java.util.ArrayList;
import com.bean.*;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
public class ShopUtil {

    public String GetProductListByCat(String u_catagory,HttpServletRequest request){
        Connection con = null;
        PreparedStatement pstm =null;
        ResultSet result1 = null;
        String returnvalue = "";
        HttpSession session = request.getSession();

        ArrayList product = new ArrayList();
```

```

        try
        {
            con= DBUtil.getMySqlConnection();
            pstmt = con.prepareStatement("SELECT prod_id, prod_cat_id, prod_name,
prod_stock, prod_price_perunit, prod_unit \r\n" +
                "FROM prod_mast \r\n" +
                "WHERE prod_cat_id = (?); ");
            pstmt.setString(1,u_catagory);

            result1 = pstmt.executeQuery();
            returnvalue = "Fail";
            while(result1.next()){
                //Retrieve by column name
                String id= result1.getString("prod_id");
                String id1= result1.getString("prod_cat_id");
                String productName= result1.getString("prod_name");
                int productStock= result1.getInt("prod_stock");
                int productPricePerUnit= result1.getInt("prod_price_perunit");
                String productUnit= result1.getString("prod_unit");
                //Display values
                ProductListByCat productbean = new ProductListByCat();

                productbean.setProd_cat_id(id);
                productbean.setProd_id(id1);
                productbean.setProd_name(productName);

                productbean.setProd_stock(""+productStock);
                productbean.setProd_price_perunit(""+productPricePerUnit);
                productbean.setProd_unit(productUnit);

                product.add(productbean);

                System.out.println("Product ID: " + id);
                System.out.println("Product Catagory ID: " + id1);
                System.out.println("Product Name: " + productName);
                System.out.println("Product Stock: " + productStock);
                System.out.println("Product Price Per Unit: " + productPricePerUnit);
                System.out.println("Product Unit: " + productUnit);
                System.out.println("*****");
            }
            returnvalue="Success";
            result1.close();

            session.setAttribute("prodcategory", product);// setting the arraylist in the session
            and array list has list product list.
        }

        catch (SQLException exc)

```

```

        {
            returnvalue="Fail";

            exc.printStackTrace();
        }
    catch (Exception exc)
    {
        returnvalue="Fail";

        exc.printStackTrace();
    }

    finally
    {
        try {
            if (result1 != null) {
                result1.close();
            }

            if (pstm != null) {
                pstm.close();
            }

            if (con != null) {
                con.close();
            }

        }
        catch(SQLException e)
        {
            returnvalue="Fail";

        }
        catch(Exception e)
        {
            returnvalue="Fail";

        }

    }

    return returnvalue;
}
}

```

ValidateLogin.java for password and otp verification for database

```
package com.util;

import java.sql.*;
import java.text.SimpleDateFormat;
import java.util.*;
import java.util.Date;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

public class ValidateLogin {

    public String UpdateOTP(int otp, String u_id) throws SQLException {
        Connection con = null;
        PreparedStatement pstmt = null;
        ResultSet result = null;
        String returnvalue = "";
        try {
            con = DBUtil.getMySqlConnection();
            pstmt = con.prepareStatement("UPDATE cust_mast SET
OTP_number=?,OTP_time=? WHERE cust_id=?");
            pstmt.setLong(1, otp);
            pstmt.setTimestamp(2,
java.sql.Timestamp.valueOf(java.time.LocalDateTime.now()));
            pstmt.setString(3, u_id);
            int resultotp = pstmt.executeUpdate();

            System.out.println(resultotp);
            System.out.println(otp);
            System.out.println(u_id);

            if (resultotp > 0) {
                System.out.println("OTP Updated to database");
            } else {
                System.out.println("Something went wrong");
            }
        }
    }
}
```

```

        finally {
            try {
                if (con != null) {
                    con.close();
                }
                if (pstm != null) {
                    pstm.close();
                }
                if (result != null) {
                    result.close();
                }
            }

            catch (SQLException e) {
                returnvalue = "Fail";
            } catch (Exception e) {
                returnvalue = "Fail";
            }
        }

        return returnvalue;
    }

    public String NewRegistration(String u_id, String u_name, String u_pass, String
u_add1, String u_add2,
        String u_city, String u_state, String u_pin, String u_mobile) {
        Connection con = null;
        PreparedStatement pstm = null;
        ResultSet result = null;
        String returnvalue = "";
        try {
            con = DBUtil.getMySQLConnection();
            pstm = con.prepareStatement(
                "insert into cust_mast
(cust_id,cust_name,cust_password,addressline_1 ,addressline_2 ,city ,state ,pin
,mobile_number, reg_date) values(?,?,?,?,?,?,?,?,?)");
            pstm.setString(1, u_id);
            pstm.setString(2, u_name);
            pstm.setString(3, u_pass);
            pstm.setString(4, u_add1);
            pstm.setString(5, u_add2);
            pstm.setString(6, u_city);
            pstm.setString(7, u_state);
            int u_pin_c = Integer.parseInt(u_pin);
            pstm.setInt(8, u_pin_c);
            pstm.setString(9, u_mobile);
            pstm.setTimestamp(10,
java.sql.Timestamp.valueOf(java.time.LocalDateTime.now()));

            /*

```

```

        * var d = new Date();
document.getElementById("demo").innerHTML = d.getDate();
        */

        int result1 = pstmt.executeUpdate();

        if (result1 > 0) {
            returnvalue = "Success";
        } else {
            returnvalue = "Fail";
        }
    }

    catch (SQLException exc) {
        returnvalue = "Fail";

        exc.printStackTrace();
    } catch (Exception exc) {
        returnvalue = "Fail";

        exc.printStackTrace();
    }

    finally {

        try {
            if (result != null) {
                result.close();
            }

            if (pstmt != null) {
                pstmt.close();
            }

            if (con != null) {
                con.close();
            }
        } catch (SQLException e) {
            returnvalue = "Fail";
        } catch (Exception e) {
            returnvalue = "Fail";
        }
    }

    return returnvalue;

```



```

    }

    public String validateOTP(String u_id, int otpcheck, HttpServletRequest request) {
        System.out.println("OTP Validation starts here ");
        Connection con = null;
        PreparedStatement pstmt = null;
        ResultSet result = null;

        String db_u_name = null;
        int db_otp = 0;
        int db_timediff = 0;

        HttpSession session = request.getSession();
        u_id = (String) session.getAttribute("uname");

        String returnvalue = "";
        try {
            con = DBUtil.getMySQLConnection();
            pstmt = con.prepareStatement(
                "select cust_id ,
OTP_number,TIME_TO_SEC(TIMEDIFF(CURRENT_TIMESTAMP,OTP_time)) from
cust_mast where cust_id = ?");
            pstmt.setString(1, u_id);

            result = pstmt.executeQuery();

            if (result == null) {
                returnvalue = "NO_VALUE_resultnull";
            } else {
                while (result.next()) {
                    db_u_name = result.getString(1);
                    db_otp = result.getInt(2);
                    db_timediff = result.getInt(3);
                    System.out.print("Database Name : " + db_u_name +
"\nDatabase OTP : " + db_otp);

                    if (db_timediff <= 40) {
                        if (otpcheck == db_otp) {
                            returnvalue = "OTP verified";
                            System.out.println("YAY! YOU DID
IT!");
                        } else {
                            returnvalue = "Wrong OTP";
                            System.out.println("OH NO!");
                        }
                    } else {
                        returnvalue = "OTP Expired";
                        System.out.println("OH NO!");
                    }
                }
            }
        }
    }

```

```

        }

    }

    if (returnvalue.trim().equals("")) {
        returnvalue = "NO_VALUE_inni";
    }

} catch (Exception e) {
    returnvalue = "NO_VALUE_exe";
} finally {
    try {
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

System.out.println(u_id);
System.out.println(db_u_name);
System.out.println(otpcheck);
System.out.println(db_otp);
System.out.println(returnvalue);
return returnvalue;
}

public String getCustDetails(String u_id) {
    System.out.println("getCustDetails start here ");
    Connection con = null;
    PreparedStatement pstm = null;
    ResultSet result = null;

    String db_u_name = null;
    String db_password = null;

    String returnvalue = "";
    try {
        con = DBUtil.getMySQLConnection();
        pstm = con.prepareStatement("select cust_id , cust_password from
cust_mast where cust_id = ?");
        pstm.setString(1, u_id);

        result = pstm.executeQuery();

        if (result == null) {
            returnvalue = "NO_VALUE";
        } else {
            while (result.next()) {
                db_u_name = result.getString(1);
                db_password = result.getString(2);
            }
        }
    }
}

```

```

        returnvalue = db_password;
    }

    }
    if (returnvalue.trim().equals("")) {
        returnvalue = "NO_VALUE";
    }

} catch (Exception e) {
    returnvalue = "NO_VALUE";
} finally {
    try {
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

return returnvalue;
}

}

```

Appendix 3

Ajaxhandler.java

```
package com.servlet;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import java.util.*;

/**
 * Servlet implementation class AjaxHandler
 */
@WebServlet("/AjaxHandler")
public class AjaxHandler extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public AjaxHandler() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
    response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        // TODO Auto-generated method stub

        response.setContentType("text/html");

        String message="";

        HttpSession session=request.getSession();
```

```

String option = request.getParameter("option");

String imgname = request.getParameter("imgname");
String pcat = request.getParameter("pcat");
String pid = request.getParameter("pid");
String pname= request.getParameter("pname");
String punit = request.getParameter("punit");
String uprice = request.getParameter("uprice");

System.out.println("Option is " + option);
System.out.println("imagename is " + imgname);
System.out.println("ID is " + pcat);
System.out.println("Catagory is"+pid);
System.out.println("NAME is " + pname);
System.out.println("PRICE is " + uprice);
System.out.println("QUANTITY is " + punit);
HashMap<String,String> map = new HashMap<String,String>();

if(session.getAttribute("cart")!=null)
{

    System.out.println("Cart is already created");
    map= (HashMap)session.getAttribute("cart");

    if(option.trim().equals("remove"))
    {
        if(map.get(imgname)!=null)
        {
            map.remove(imgname);
            message ="Product is removed from the
Cart";
            System.out.println("Cart is already
created");

        }
        else
        {
            message ="Product is not Presnet hence
can not be removed";

        }

    }

    ;

}

if(option.trim().equals("add"))
{

```

```

        if(map.get(imgname)!=null)
        {
            message ="Same Product
is already added";
System.out.println(message);
        }
        else
        {
            map.put(imgname,pcat+""+pid+""+pname+""+punit+""+uprice+"\n");
            message ="Product added
Successfully";
System.out.println(message);
        }
    }
}

```

```

    }
    else
    {
        if(option.trim().equals("add"))
        {
            System.out.println("Cart is being exist");
            map.put(imgname,pcat+""+pid+""+pname+""+punit+""+uprice);
            session.setAttribute("cart", map);
            message ="Product added Successfully";
        }
    }
}

```

```

        if(option.trim().equals("remove"))
        {
            message ="No Product is in the Cart";
        }
    }
}

```

```

        response.getWriter().write(message);
    }

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response)

```

```

throws ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
}
}

```

ListCartItems.java

```

package com.servlet;

import java.io.IOException;
import java.util.HashMap;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * Servlet implementation class ListCartItems
 */
@WebServlet("/ListCartItems")
public class ListCartItems extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ListCartItems() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub

        response.getWriter().append("Served at: ").append(request.getContextPath());
        System.out.println("ViewCart is being called.....");

        request.getRequestDispatcher("viewcartindex.jsp").forward(request,

```

```

response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub

        doGet(request, response);
    }
}

```

login.java

```

package com.servlet;

import java.io.IOException;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.util.ValidateLogin;

/**
 * Servlet implementation class Login
 */
@WebServlet("/Login")
public class Login extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Login() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**

```



```

    * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
    *     response)
    */
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // TODO Auto-generated method stub
    response.getWriter().append("Served at: ").append(request.getContextPath());
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
 *     response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // TODO Auto-generated method stub
    String LoginStatus = "";
    doGet(request, response);
    String uname = request.getParameter("uname");
    String pass = request.getParameter("upass");
    String uemail = request.getParameter("uemail");
    String fname = request.getParameter("fname");
    String u_pass = request.getParameter("u_pass");
    String cu_pass = request.getParameter("cu_pass");
    String add1 = request.getParameter("add1");
    String add2 = request.getParameter("add2");
    String city = request.getParameter("city");
    String state = request.getParameter("state");
    String pin = request.getParameter("pin");
    String uphone = request.getParameter("uphone");

    System.out.println(uname);
    System.out.println(pass);

    ValidateLogin obje = new ValidateLogin(); // object created to call
getCustDetails
    if (request.getParameter("command").trim().equals("otp")) {
        String otp_entered = request.getParameter("uotp");
        System.out.println("the otp isssssssssssssssssssssssssssss: " +
otp_entered);

        int otp_tocheck = Integer.valueOf(otp_entered);
        System.out.println(uname + "this is usernameeeee");
        String verifiedotp = obje.validateOTP("uname", otp_tocheck, request);

        if (verifiedotp == "OTP verified") {

request.getRequestDispatcher("ShopByCategory.jsp").forward(request, response);

```

```

        }
        else if(verifiedotp == "Wrong OTP") {
            request.getRequestDispatcher("otp.jsp").forward(request,
response);
        }
        else {
            request.getRequestDispatcher("index.jsp").forward(request,
response);
        }
    }

    else if (request.getParameter("command").trim().equals("register")) {
        request.getRequestDispatcher("Register.jsp").forward(request,
response);
    } else if (request.getParameter("command").trim().equals("signup")) {
        String verify = "no";
        System.out.println(uemail);
        System.out.println(fname);
        System.out.println(u_pass);
        System.out.println(cu_pass);
        System.out.println(add1);
        System.out.println(add2);
        System.out.println(city);
        System.out.println(state);
        System.out.println(pin);
        System.out.println(uphone);
        verify = obje.NewRegistration(uemail, fname, cu_pass, add1, add2,
city, state, pin, uphone);
        System.out.println("Im in try");
        System.out.println("This is the answer : " + verify);

//
//
//      } catch (SQLException e) {
//          // TODO Auto-generated catch block
//          System.out.println("Im in catch");
//          e.printStackTrace();
//      }

        if (verify.equals("Success")) {
            request.getRequestDispatcher("index.jsp").forward(request,
response);
        } else {
            request.getRequestDispatcher("Register.jsp").forward(request,
response);
        }
    } else {
//
request.getRequestDispatcher("ShopByCategory.jsp").forward(request,
// response);
        System.out.println("else post t");
        System.out.println("User name is as : " + uname);
    }
}

```

```

String status = obje.getCustDetails(uname);
System.out.println(status);

request.setAttribute("status", status);
java.util.Random rand = new java.util.Random();
int otpno = rand.nextInt((9999 - 100) + 1) + 10;
System.out.println(otpno);

if (status.equals(pass)) {
    try {
        String otpUpdateStatus = obje.UpdateOTP(otpno,
uname);

        } catch (SQLException e) {
            e.printStackTrace();
        }
        HttpSession session = request.getSession();
        session.setAttribute("uname", uname);
        request.getRequestDispatcher("otp.jsp").forward(request,
response);

    } else if (status.equals("NO_VALUE")) {
        // user not found
        LoginStatus = "NoUser";
        // HttpSession session=request.getSession();
        request.setAttribute("LoginStatus", LoginStatus);
        request.getRequestDispatcher("index.jsp").forward(request,
response);

    } else {
        LoginStatus = "Invalid";
        request.setAttribute("LoginStatus", LoginStatus);
        request.getRequestDispatcher("index.jsp").forward(request,
response);
    }
}
}
}

```

PlaceOrder.java

```

package com.servlet;

import java.io.*;
import java.util.*;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;

```

```

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.util.ShopUtil;

/**
 * Servlet implementation class PlaceOrder
 */
@WebServlet("/PlaceOrder")
public class PlaceOrder extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public PlaceOrder() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
        ShopUtil order_obj = new ShopUtil();
        HttpSession session=request.getSession();
        HashMap<String,String> map = new HashMap<String,String>();
        String orderno = order_obj.getOrderNumber();

        session.setAttribute("cart");

        if(session.getAttribute("cart")!=null)
        {

```

```

        map =(HashMap)session.getAttribute("cart");
    }
    else
    {
        System.out.println("Product is not aviable in the cart");
    }

    if(map.size()==0)
    {
        System.out.println("Product is not aviable in the cart.....");
    }
    if(map.size(>0)
    {
        Set set = map.entrySet();

        Iterator iterator = set.iterator();

        float tot=0;

        while(iterator.hasNext())
        {
            Map.Entry<String,String> enry =
(Map.Entry<String,String>)iterator.next();

            String key = enry.getKey();
            String value= enry.getValue();
            String value_array [] = value.split("~");
            tot = tot+Float.parseFloat(value_array[4]);
            System.out.println("the sum is" + tot);

            System.out.println(value_array[0]);
            System.out.println(value_array[1]);
            System.out.println(value_array[2]);
            System.out.println(value_array[3]);
            System.out.println(value_array[4]);

            order_obj.AddOrderDetail(orderno, value_array[0], value_array[1],
value_array[2] ,1 ,Float.valueOf(value_array[4]));
        }

        order_obj.AddtoOrderMaster(orderno, "cod", "VIT University", "Vellore"
,"Order Placed",(float) tot);
        session.setAttribute("tot", tot);
    }

    session.setAttribute("orderno", orderno);
    request.getRequestDispatcher("orderConfirmation.jsp").forward(request,
response);
}

```

```
}
```

Shopping.java

```
package com.servlet;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.util.ShopUtil;
import com.util.ValidateLogin;

/**
 * Servlet implementation class Shopping
 */
@WebServlet("/Shopping")
public class Shopping extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Shopping() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
    response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
    response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```

throws ServletException, IOException {
    // TODO Auto-generated method stub

    doGet(request, response);
    String u_catagory = request.getParameter("catagory");
    System.out.println(u_catagory+" Recieved in Shopping.java!");

    ShopUtil obje = new ShopUtil();
    String verifyCatagory = "no";

    if((request.getParameter("catagory").trim().equals("001"))||(request.getParameter("catagory")
.trim().equals("002"))||

    (request.getParameter("catagory").trim().equals("003"))||(request.getParameter("catagory").tri
m().equals("004"))||

    (request.getParameter("catagory").trim().equals("005"))||(request.getParameter("catagory").tri
m().equals("006"))||

    (request.getParameter("catagory").trim().equals("007"))||(request.getParameter("catagory").tri
m().equals("008"))||

    (request.getParameter("catagory").trim().equals("009"))||(request.getParameter("catagory").tri
m().equals("010"))||

    (request.getParameter("catagory").trim().equals("011"))||(request.getParameter("catagory").tri
m().equals("012")))
    {

        verifyCatagory =
obje.GetProductListByCat(u_catagory,request);
        if(verifyCatagory.equals("Success")) {

            request.getRequestDispatcher("shop.jsp").forward(request, response);
        }

    }

    else {

        request.getRequestDispatcher("ShopByCategory.jsp").forward(request, response);
    }

}

```

Apache TomCat for server

Apache Tomcat is an open-source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and WebSocket technologies. Tomcat provides a "pure Java" HTTP web server environment in which Java code can run.