

PROJECT REPORT
“Online Shopping System
(Vinegar and Jaggery Store)”

REPORT SUBMITTED FOR THE PARTIAL FULFILMENT OF THE
REQUIREMENT FOR THE THREE YEAR DIPLOMA IN
“Computer Science/Information Technology”

SUBMITTED TO
“COLLEGE NAME”
CITY

SUBMITTED BY:
STUDENT NAME

(College Enrolment Number- **XXXXXXX**)
2020-2023



Under the Guidance of

Er. Brijesh Mishra

Senior Consultant

SOFTPRO INDIA COMPUTER TECHNOLOGIES (P) LTD



SOFTPRO INDIA

COMPUTER TECHNOLOGIES PVT.LTD.

(AN ISO 9001:2015 CERTIFIED COMPANY)

SPI/VT/2022/.....

CERTIFICATE OF COMPLETION

This is to certify that Mr. /Ms. **Student Name** of **Diploma (Computer Science)** from **College Name** was working on the project entitled “**Online Shopping System**” in Softpro India Computer Technologies (P) Ltd. She/he was engaged with us for training during **July 2022** to **August 2022** for 01 month.

She/he has done an excellent job during his/her engagement with the Software Development & Testing Division of the company. She/he has completed his/her project during the training tenure. His/her performance has been good and satisfactory.

I would like to take this opportunity to express my appreciation to Mr./Ms. **Student Name** for his/her work and wish him/her all the very best for his/her future endeavors.

Regards,

Ms. Yashi Asthana

C.E.O

Softpro India Computer Technologies (P) Ltd

Lucknow (U.P.)

DECLARATION

I, <student name>, hereby, declare that the Project Report submitted to <College Name> for the award of three-year Diploma in <Branch Name> was done under the guidance of Er. Brijesh Mishra, Senior Consultant, Softpro India Computer Technologies (P) Ltd.

I also declare that the information in this report is correct as per my knowledge and I bear any responsibilities for any error or omission, if any. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

Student Name

Date:

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my trainer and mentor Er. Brijesh Mishra, Senior Consultant, **(Softpro India Computer Technologies (P) Ltd.)**, who gave me his full support and encouraged me to work in an innovative and challenging project for Educational field. His knowledge and guidance were extremely helpful in completion of my training. I would also extend my thanks to every member of Softpro India Computer Technologies (P) Ltd. for their support and co-operation.

Student Name

INDEX

S. No.	Title
1	About the Organisation
2	About BTEUP
3	About Summer Training
4	Technologies Trained During Summer Training
5	Introduction of the Project
6	About the Project
7	Snapshot & Coding
8	Future Scope

ABOUT THE ORGANISATION

Softpro India Computer Technologies (P) Ltd. is a leading IT firm and the software development division of Softpro Group of Companies with its headquarter located in the capital city of Uttar Pradesh, Lucknow. Softpro India was established in 2004 by technocrats from IIT-Kanpur and IET Lucknow. Softpro Group of Companies is a cluster of companies working in multiple domains like Software Development, IT Trainings, Research and Designing. The Founder and Managing Director of Softpro Group of Companies is Er. Ajay Chaudhary with over 25 years of experience. Softpro India is the fastest growing IT company with the largest learning center of the region having experienced consultants of 15+ years and industry experts.

Softpro Group of Companies comprises of Softpro Learning Center (Training & Internship division – 2008), Softpro Innovations (R&D division – 2014) and Softpro Foods (Agro Production division – 2018). Softpro India has global presence with its Head Office and Training Center located in Lucknow, International Unit Office located in Malawi, Africa and Virtual Office located in Kuala Lumpur, Malaysia. Softpro India has successfully delivered Government Projects like the visionary project of Government of Uttar Pradesh – *URISE*. Softpro India also has signed MoU with Department of Technical Education, Government of Uttar Pradesh making it the authorised Training & Development partners to impart and technically upskill all the engineering students of polytechnics (government, private & aided) across Uttar Pradesh.

Softpro India's recent achievements include the MoU signing with Dr. A. P. J. Abdul Kalam Technical University, Uttar Pradesh. Technologies are transcending boundaries and their volatility is putting stringent demands on the time and mind-space of techno-professionals. At SPG, we update ourselves with technologies even before they become norms and master them long before they become redundant. That's why we are on the roster of clients from across the continents.

Softpro India offers training for all the branches of engineering (Computer Science, Information Technology, Electronics, Electrical, Civil, Mechanical) for updated and trending technologies. Softpro India also has several online and offline trainings like Summer Training, Industrial Training, Vocational Training, Apprenticeship Program, Employment Training Program and Online Courses. The learning material and other resources are available on Softpro India's Learning Management System (LMS) – “Polyprep – Knowledge @ Your Doorstep” and mobile application – “e-Study Zone”.

Come to think of it, we have engineered ourselves to be at the very forefront in Web based technology. Our core competencies span a spectrum of web-intensive services that range from website designing to robust backend management.

About BTEUP

State Board of Technical Education and Training was set up in the state in May, 1958. The Board conducted its first examination in 1960, for courses of diploma level and also for Draughtsman Certificate Course. The name of the state Board was changed to Board of Technical Education in 1962. In the same Year, the U.P. Pravidhic Shiksha Adhiniyam - 1962 was enacted awarding the Board statutory status. In the year 1962, year of its inception, the Board held the examination of about 2500 students, in three major disciplines of Civil, Electrical and Mechanical Engineering at 25 different Centers/Institutions. It has a chairman, vice-chairman and 40 members, named by the state government. Secretary, Board of Technical Education, U.P. is ex-officio member secretary of the

Board. The curriculum prepared by other institutions was adopted by the Board, till 1980 but thereafter the curriculum and syllabi were developed and revised, every five years at the Board level through its Curriculum development cell. The curriculum development work has been entrusted to I.R.D.T. Kanpur. The Board now examine and approve the syllabus developed by IRDT, Kanpur and prescribe it for institutions affiliated to Board of Technical Education, U.P. About 1,05,000 students in the 60 different disciplines of one year, two year, three year and four year durations are being examined at present, in the institutions, affiliated to the Board. The Board of has been constituted under U.P. Pravidhic Shiksha Adhiniyam - 1962. It has a Chairman and a Vice-Chairman and, 40 members nominated by the state Government. Secretary, Board of Technical Education is ex-officio member secretary of the Board.

SUMMER TRAINING

Summer training is an important part of the engineering curriculum. The summer training helps a student in getting acquainted with the manner in which his/her knowledge is being practically used outside his/her institute and this is normally different from what he/she has learnt from books. Hence, when the student switches from the process of learning to that of implementing his/her knowledge, he/she finds an abrupt change. This is exactly why summer training session during the B.T.E curriculum becomes all the more important. Summer training is prescribed for the student of Technical College as a part of the four-year degree course of engineering by the AICTE. We are required to undergo summer training for a period of 45 days after the completion of the 2nd year.

Summer Training allows industry exposure and understanding the working environment & it gives the industrial exposure. It is essential and helps to attain in-depth knowledge of the engineering stream. It enhances professional skills in a real-time environment. It helps us understand the area of interest and selection of an area of specialization. It also allows students to learn the basics of how to work as a team member to complete given tasks. It improves awareness of the industrial environment and work culture of the specific industry. It gives real-time work and the projects help to learn more analytically. It allows interaction with experts to solve queries with practical exposure. The Certificate obtained from reputed organization give weightage to resume or CV.

TECHNOLOGIES TRAINED ON DURING SUMMER TRAINING

1. **HTML:** - HTML stand for hypertext markup language, this markup language is used to design static web pages. HTML contain pre-defined tags, which are useful to design web pages. HTML describes the structure of a Web page. HTML consists of a series of elements.

HTML elements tell the browser how to display the content. HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

2. **CSS:** - CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files.
3. **C Language:** - C is a general-purpose programming language created by Dennis Ritchie at the Bell Laboratories in 1972. It is a very popular language, despite being old. C is strongly associated with UNIX, as it was developed to write the UNIX operating system. It is one of the most popular programming language in the world. If you know C, you will have no problem learning other popular programming languages such as Java, Python, C++, C#, etc, as the syntax is similar. C is very fast, compared to other programming languages, like Java and Python C is very versatile; it can be used in both applications and technologies.
4. **Javascript:** - JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS).
5. **Bootstrap:** - Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website. It is absolutely free to download and use. It is a front-end framework used for easier and faster web development. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others. It can also use JavaScript plug-ins. It facilitates you to create responsive designs.
6. **Database:** - A database is an organized collection of data, so that it can be easily accessed and managed. You can organize data into tables, rows, columns, and index it to make it easier to find relevant information. Database handlers create a database in such a way that only one set of software program provides access of data to all the users. The main purpose of the database is to operate a large amount of information by storing, retrieving, and managing data. There are many dynamic websites on the World Wide Web nowadays which are handled through databases. For example, a model that checks the availability of rooms in a hotel. It is an example of a dynamic website that uses a database. There are many databases available like MySQL, Sybase, Oracle, MongoDB, Informix, PostgreSQL, SQL Server, etc.

7. **Python:** - Python programming language is a high level, object-oriented programming language with dynamic semantics. It is the main technology of my internship. Python is a general purpose programming language which is useful to develop following types of applications:-

- Desktop application
- Web application
- Data Science and data visualization
- Machine Learning
- Embedded System
- Robotics
- Physical computing
- Internet of Things (IOT)
- Game Development.

8. **Django Framework:** - Django is a web framework written in powerful python programming language. It is used to develop scalable, secured and easily maintainable web applications. Basically frameworks are installable softwares, they provide us pre-designed architecture. By using framework you can develop software applications more easily. If you develop a software application by using software technology then you need to write common logics and application specific logics, whereas if you develop software application using framework then you need to write only application specific logic and framework take care of common logics. Django is a web framework which follows MVT (Model – View - Template) pattern. Explanation of MVT is given below:-

- Model – Model contain database part of django application.
- View – View decides response on given request.
- Template – Template is nothing but a simple html page.

INTRODUCTION OF THE CLIENT

The Gavai Swad Agro Products P Limited (Gavai Swad) is a company founded in 2020 during the lockdown period when the food services were not available in offline mode for window shopping.

The vision behind starting “Gavai Swad” is to provide the real taste of India and the real products made out of desi raw material. Nowadays, getting Organic fruits and vegetables is next to impossible since adulteration has become a common practice. Unlike conventional ones which are coated with

waxes and sprayed with chemicals, basically mummifying them. We, at Gavai Swad treat nature as it should be and provide a wide variety of the freshest possible products free from harmful pesticides and chemicals, that disrupt our physical health and mental wellbeing.

We offer a range of products like: Sugarcane sirka, pickles, murabba, jaggery products to our customers freshly made without using any chemicals & pesticides. Since now all of us are getting more internet & technology savvy and we all want the services at our doorstep and therefore gavaiswad.com portal came into existence. The products of Gavai Swad are available on Amazon, Flipkart, Swiggy & other e-commerce based food portals.

OBJECTIVES

An online shopping system is a process in which people (specifically customers) are being provided with the option of purchasing goods and services directly from the seller, all in a real-time environment. Online shopping is an application of the internet as electronic commerce. From the business perspective, customers usually find the products more attractive, on websites, as they get all the details available there.

People in large number are doing online shopping today, and it is not only because it is convenient as one can shop from home, but also because there is an ample number of varieties available, with a high competition of prices, and also it is easy to navigate for searching regarding any particular item.

For sellers, their product has access to the World-Wide market, which also increases the number of customers and enhances customer relationships. Also, web stores are a means for small-scale companies to launch their products at the global level. The main objective of this project is to develop a web-oriented application that can provide an online shopping feature to users.

In other words, the project aimed at creating a virtual shop environment for users, in some handy form, which will be available to them through the internet. Although the idea of developing online shopping websites is not new in the electronic market and has been evolved soon after the World Wide Web (www).

In the present scenario, the biggest market for this (online shopping) business is by highly educated people, mostly. This system has been designed keeping in mind all the aspects such as loading the data, complexity, and maintaining the security of user credentials. Here in this system, complexity refers to the total number of features being provided to users, and their smooth arrangement and functioning required.

Problems in Existing System

The present scenario for shopping is to visit the shops and market manually and then from the available product list one needs to choose the item he or she wants and then pay for the same item mainly in cash mode is done, as not every society is well educated and aware to use net banking or card modes or wallets etc.

This system is not much user-friendly as one needs to go to the market physically and then select items only from the available list. So mostly it is difficult to get the product as per our desire. Description About the products is less available and are mostly verbal only. For this type of shopping, one needs to have an ample amount of free time.

Also, not really good markets exist everywhere, so many times good markets become out of reach for certain people. In the proposed system customers need not go to the shops for purchasing the products. He/she can order the product he/she wishes to buy through the use of this system. The shop owner can be the admin of the system.

The shop owner can appoint officials particularly to handle this, who will help the owner in managing the customers and product orders. The system also endorses a home delivery system for delivering the purchased products.__

Proposed System

In the proposed system customer need not go to the shop for buying the products. He can order the product he wish to buy through the application in his Smartphone. The shop owner will be admin of the system. Shop owner can appoint moderators who will help owner in managing the customers and product orders. The system also recommends a home delivery system for the purchased products.

This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system recommends a facility to accept the orders 24*7 and a home delivery system which can make customers happy. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be

losing any more customers to the trending online shops such as **Flipkart or ebay**. Since the application is available in the Smartphone it is easily accessible and always available.

Features

1. Provides the searching facility based on various type of product to customers.
2. Online shopping system also manages the various type of product to customer.
3. It tracks all the information of various types of products to the end customers.
4. Manages the information of customers.
5. Shows the information and description of the various types of products to customer.
6. To increase efficiency of managing the e-commerce platform.
7. It deals with monitoring the information and transaction of payments.
8. Adding, Editing, and updating of records is improved which results in proper resource management of online shopping system data.

ABOUT THE PROJECT

This Project is an online shopping system. The analysis steps of project are given below:-

- **Feasibility Study**

Feasibility study is the measure of how beneficial or practical the development of an information system will be to an organization. The Feasibility analysis is a cross life cycle activity and should be continuously performed throughout the system life cycle.

- **Operational Feasibility: -**

By providing the web based application, all the users will get a very good facility of accessing the service to fulfil their requirements. All the user information, information sharing and selection process is done properly.

Users will feel comfortable by reduction of their work. The system will make handling of large databases easy. Losing of records will be avoided. Considering all these factors, we can conclude that all the users and end users will be satisfied by the system.

- **Technical Feasibility:-**

For the design and development of the system, several software products have been accommodated.

- Database design – SQLite
- Interface design – HTML, CSS, Java Script and Bootstrap.
- Coding – Python with django framework.

The technology (Python with django framework) has enough efficiency for the development of the system. Therefore the project is technically feasible.

- **Schedule Feasibility: -**

The duration of time required for the project has been planned appropriately and it is the same as the duration of time expected by the client. Therefore the website can be delivered to the client within the expected time duration, satisfying the client. Hence the project is feasible in scheduling.

- **Economic Feasibility: -**

According to the resources available and the project scheduling process it is estimated that the expenses allocated for the web application to be developed, by the client is sufficient enough. Hence the economical factor has been considered feasible.

- **Project Planning & Scheduling:-**

Planning is very important part of any software development. In the planning phase, we decide which features are to be included in the system to make a good system, how much time do we need to complete the project, what will the cost of the system etc...

A Software Life Cycle or software process is a series of identifiable stages that a software product undergoes during its development. A software product development effort usually starts with a project identification and selection stage and then requirements analysis; design, coding, testing, implementation and maintenance are undertaken.

A life cycle model identifies all the activities required to develop and maintain a software product and establishes a precedence ordering among the different activities.

The various phases of Software Development Life Cycle-

- Requirement Analysis

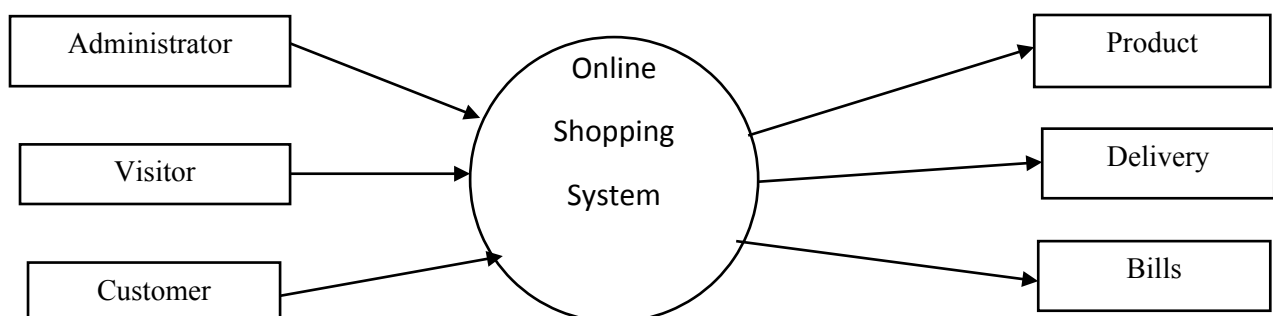
- System Design
- Coding
- Testing
- Implementation

SOFTWARE REQUIREMENT FOR DEVELOPMENT

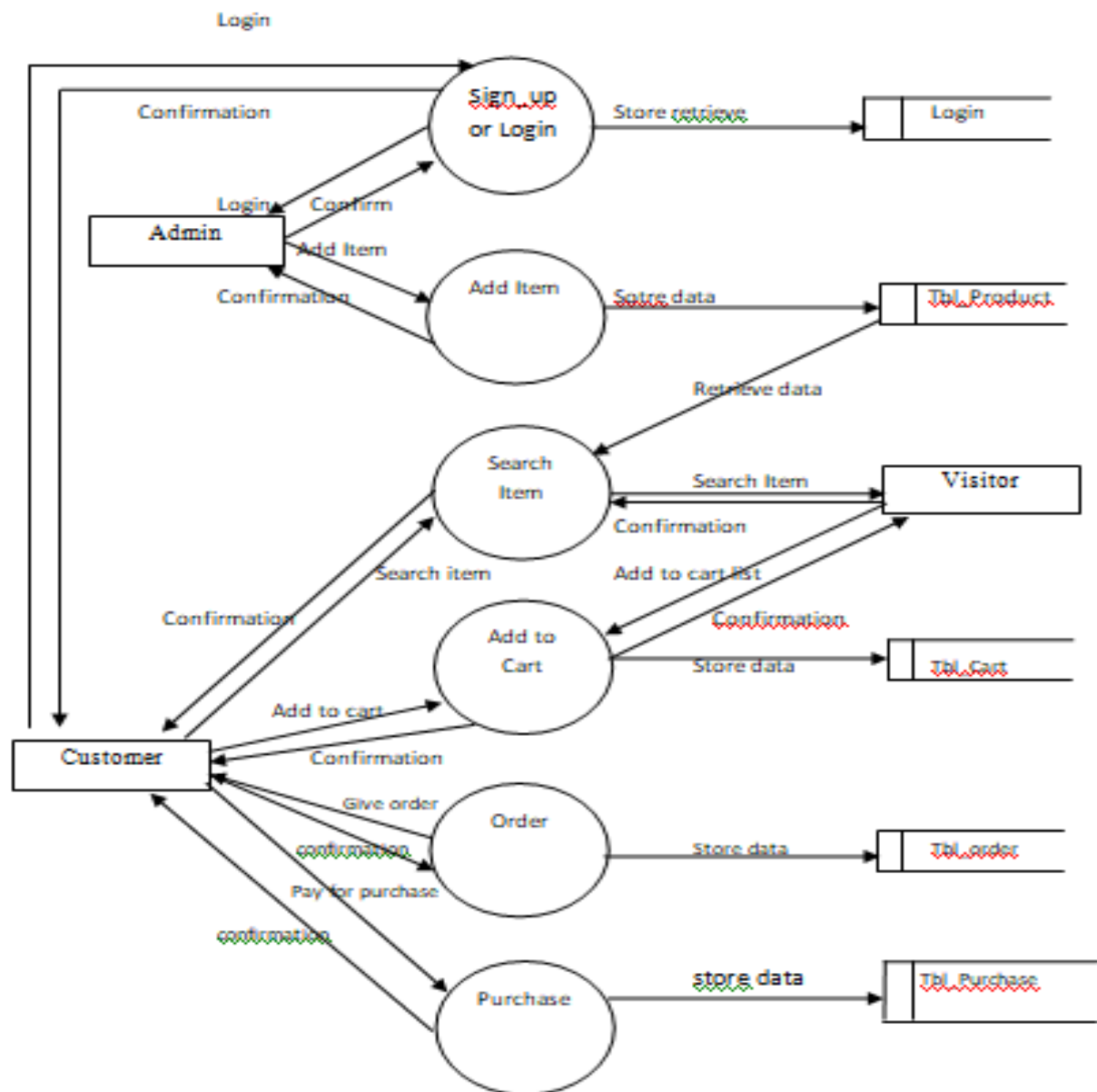
User Interface Designing	HTML5, CSS3, Java Script, Bootstrap
Programming Language	Python with Django Framework
Database	SQLite3
IDE	PyCharm

High Level Designing

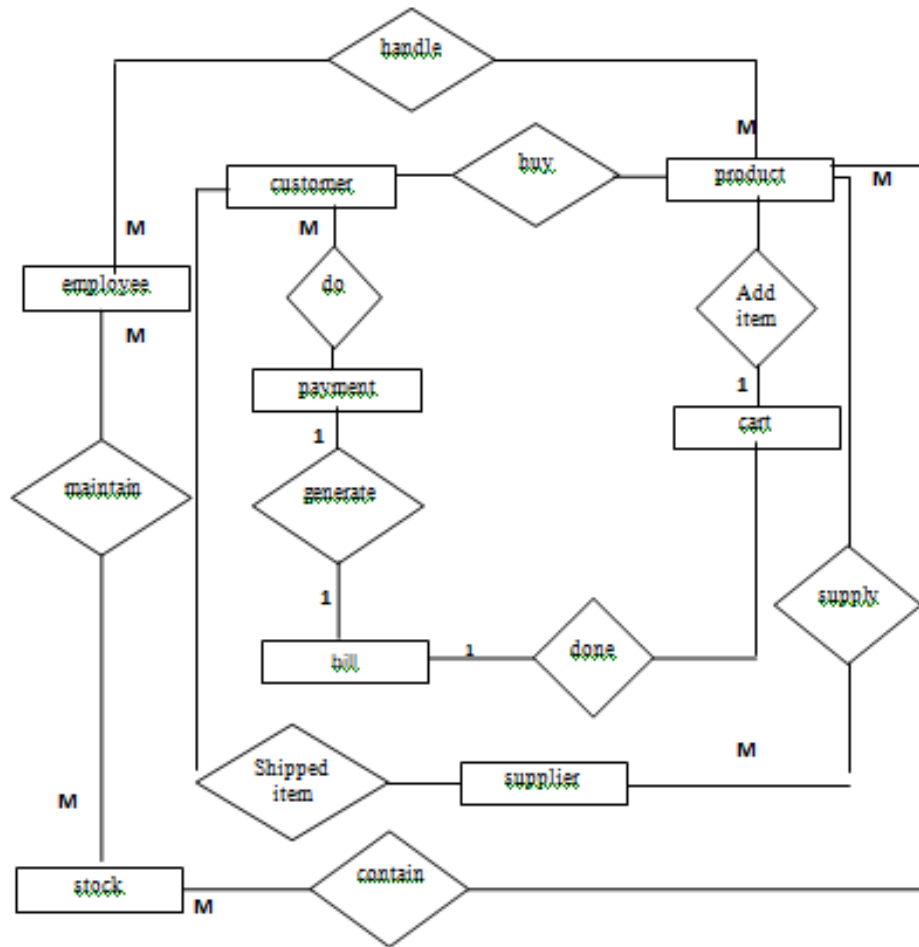
Context Level DFD



1 Level DFD



ER – DIAGRAM



Low Level Designing

Database Designing

Table Name : Customer

Column Name	Data Type
name	varchar(50)
gender	varchar(6)
address	varchar(255)
contactno	int
pincode	int
emailaddress	varchar(50) primary key
password	varchar(20)
regdate	datetime auto add current date

Table Name : Login

Column Name	Data Type
userid	varchar(50) primary key
password	varchar(20)
usertype	varchar(30)

Table Name : Category

Column Name	Data Type
id	int primary key auto_increment
name	varchar(50)

Table Name : Product

Column Name	Data Type
id	int primary key auto_increment
name	varchar(50)
price	int default=0
desc	varchar(200)
category	varchar(50) foreign key Category
image	ImageField

Table Name : Orders

Column Name	Data Type
id	int primary key auto_increment
customer	varchar(50) foreign key customer
product	varchar(50) foreign key product
address	varchar(200)
pincode	int
quantity	int
order_date	datetime
completed	boolean

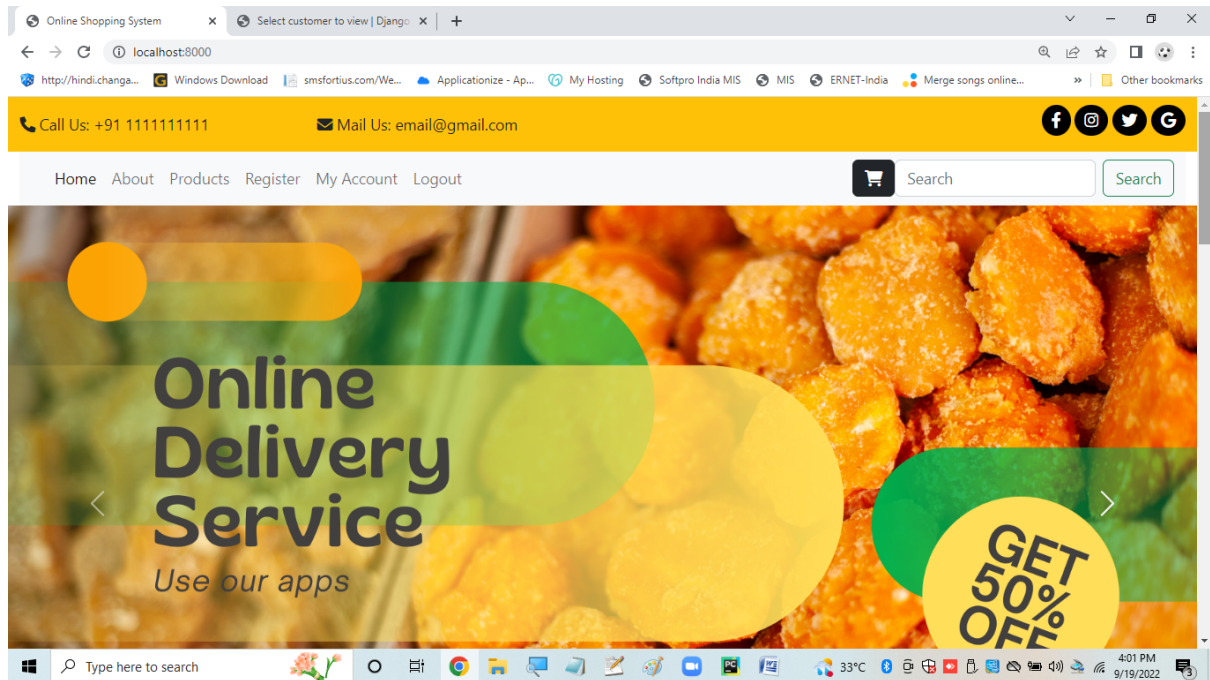
Table Name : ShoppingCart

Column Name	Data Type
-------------	-----------

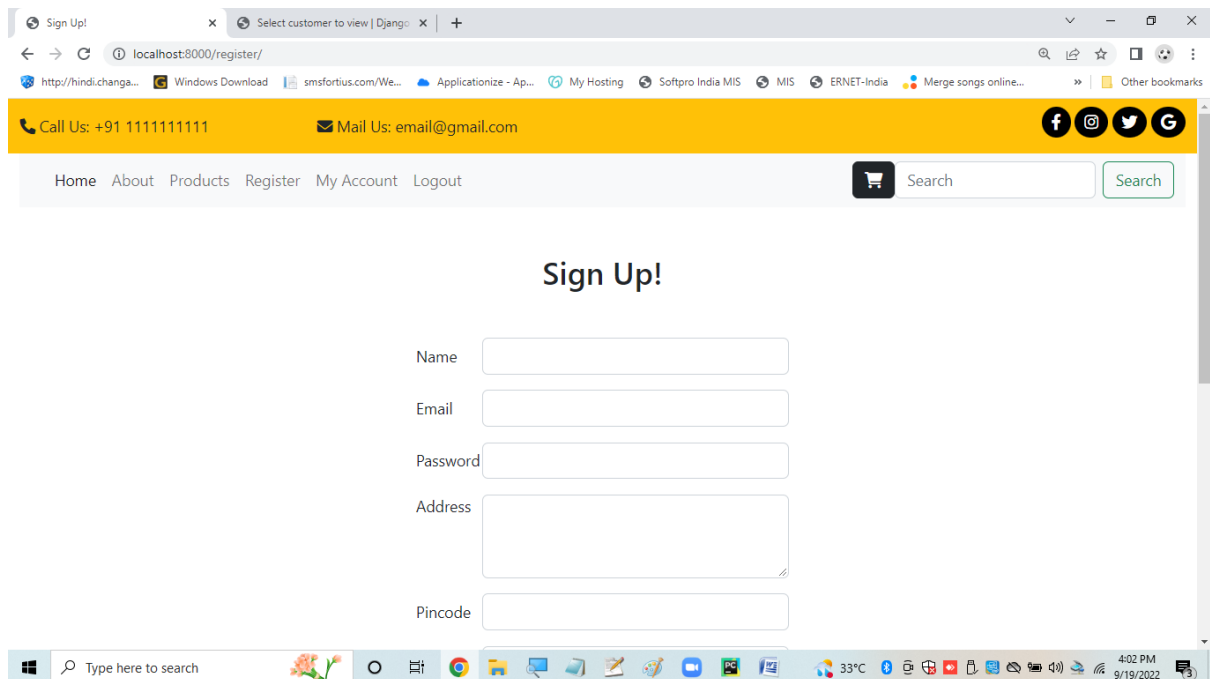
id	int primary key auto_increment
customer	varchar(50) foreign key customer
product	varchar(50) foreign key product
quantity	int

Snap Shots

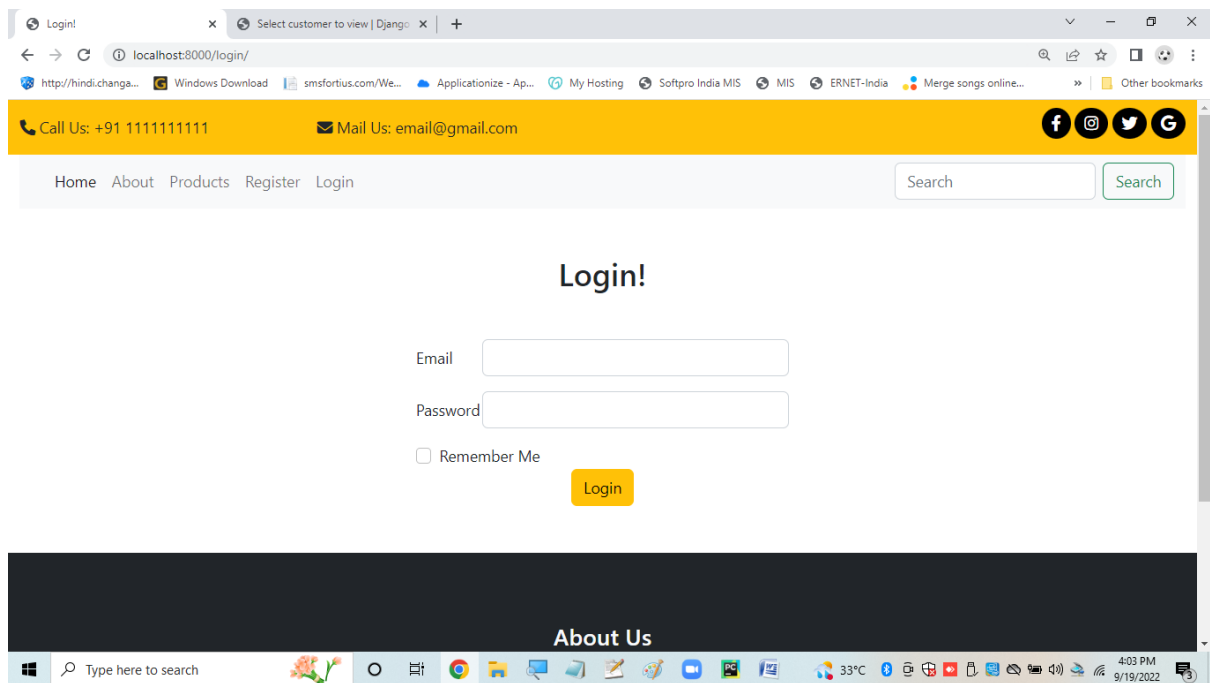
HOME PAGE



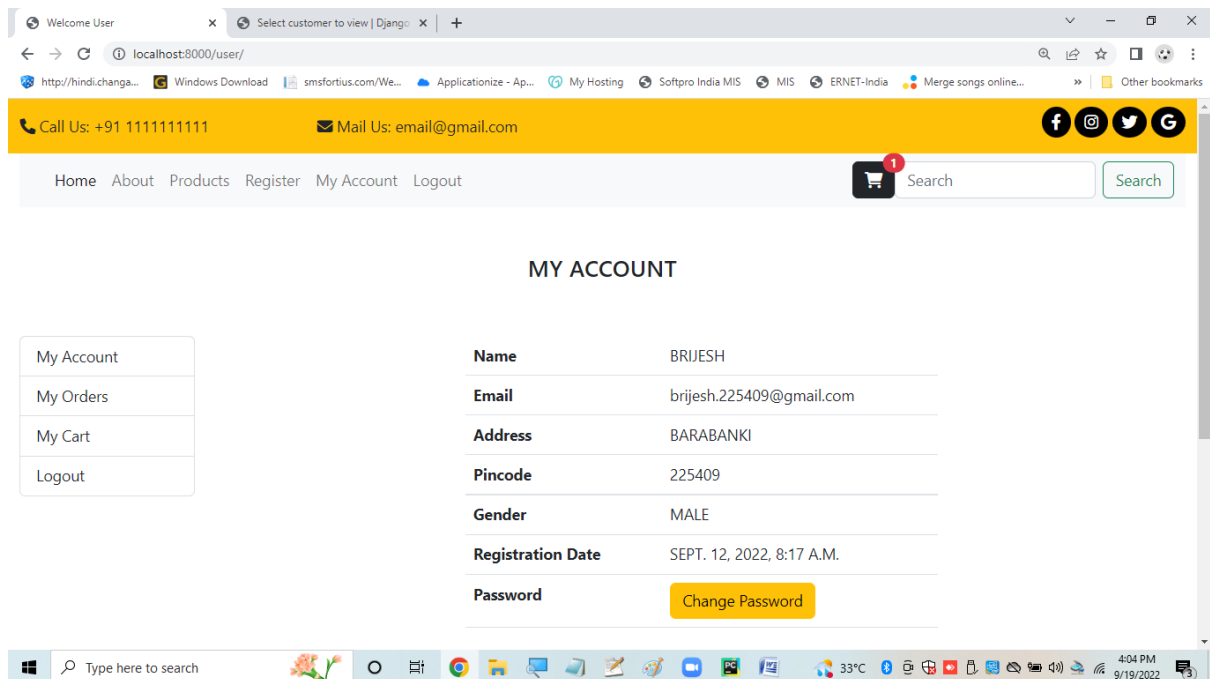
REGISTRATION PAGE



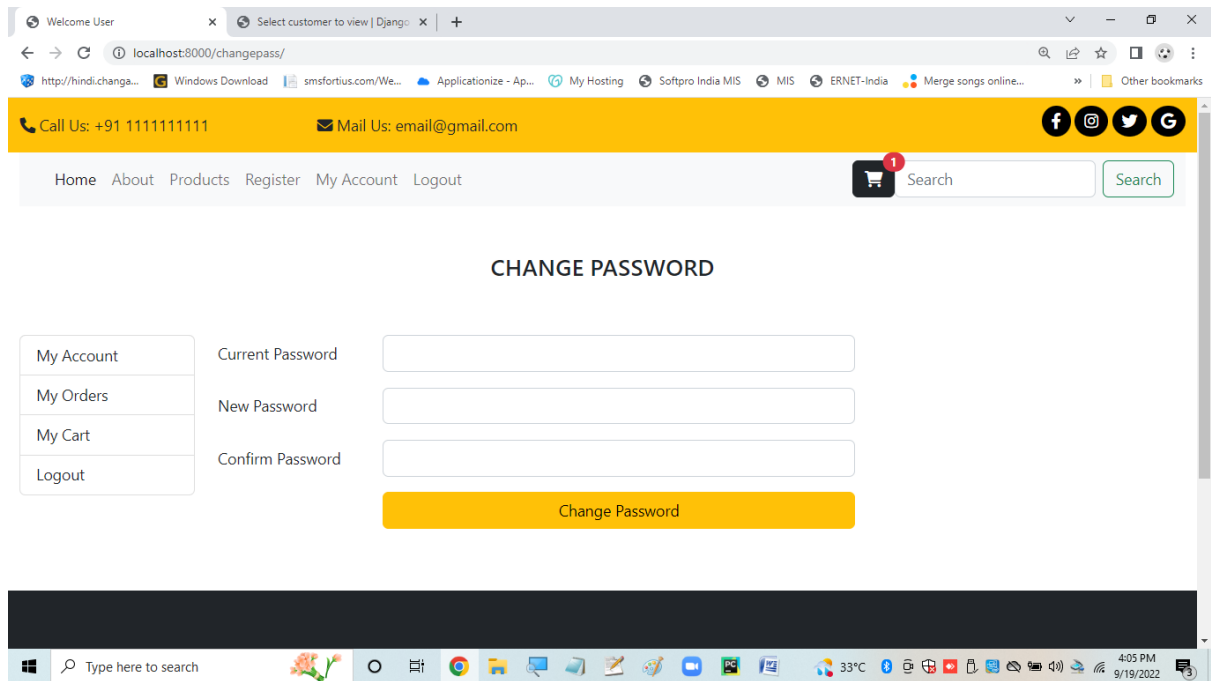
LOGIN PAGE



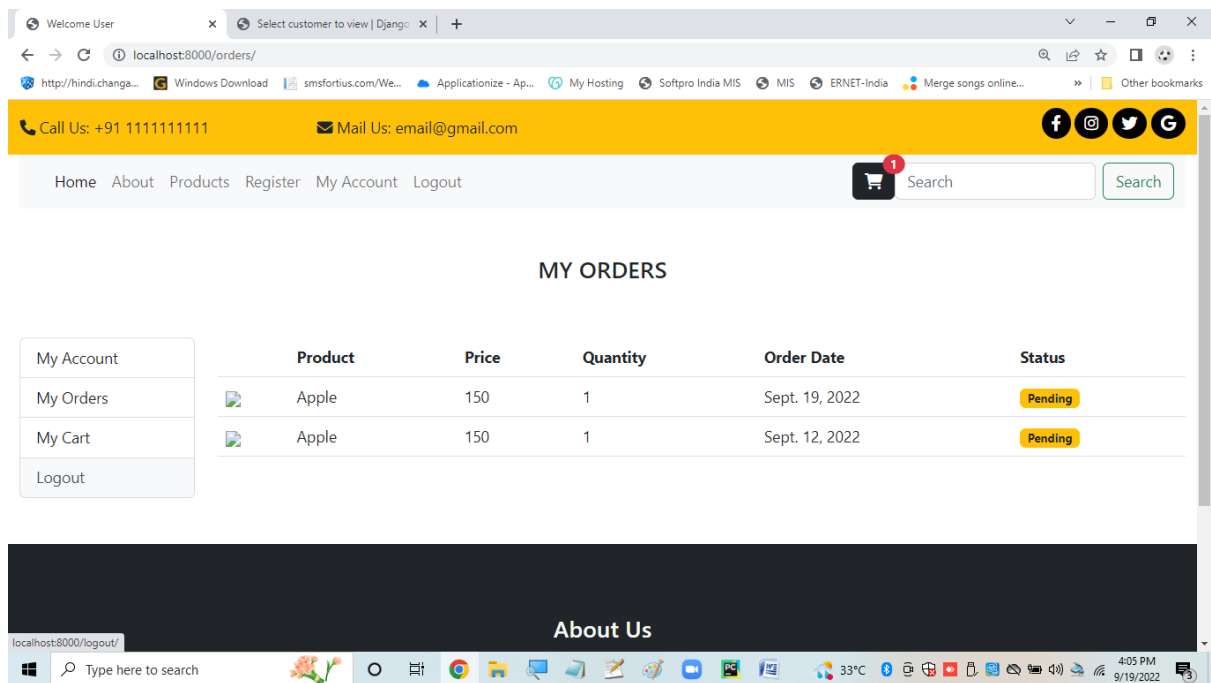
USER ACCOUNT PAGE



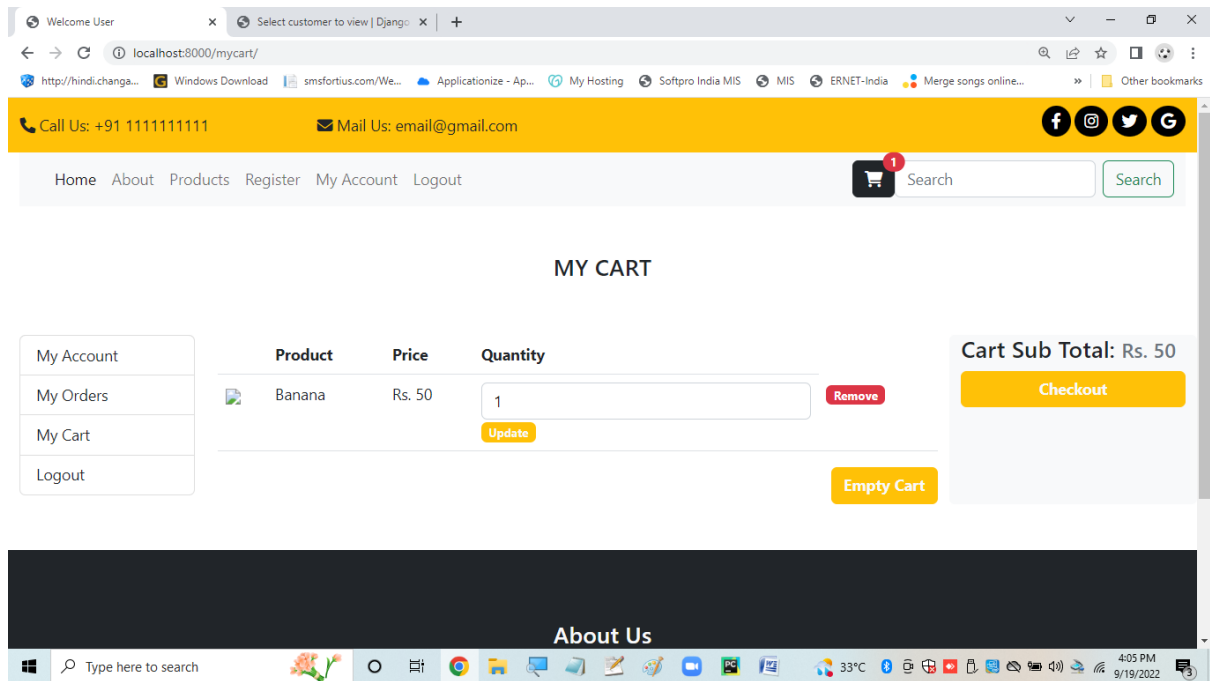
CHANGE PASSWORD PAGE



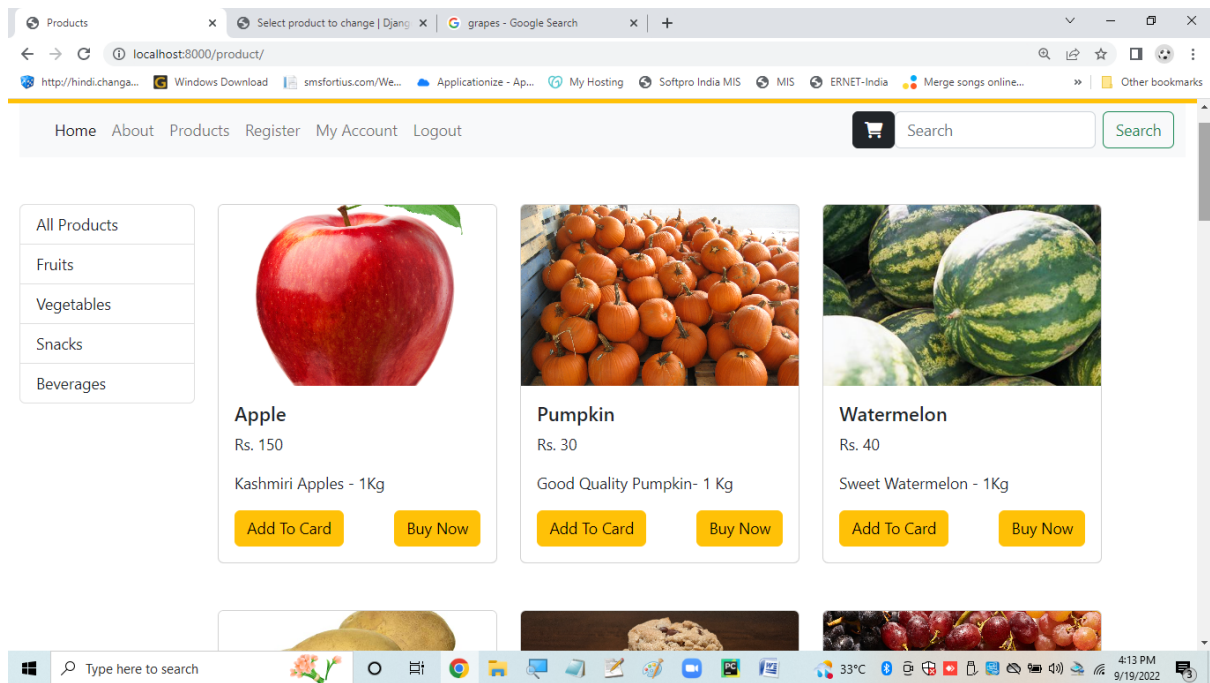
MY ORDERS PAGE



MY CART PAGE



PRODUCTS PAGE



Coding

models.py

```
import datetime

from django.db import models

class Customer(models.Model):

    name=models.CharField(max_length=50)

    gender=models.CharField(max_length=6)

    address=models.TextField()

    pincode=models.IntegerField()

    contactno=models.CharField(max_length=15)

    emailaddress=models.EmailField(max_length=50,primary_key=True)

    password = models.CharField(max_length=20, null=True)

    regdate=models.DateTimeField(auto_now_add=True)


    def is_exists(self):

        if Customer.objects.filter(emailaddress=self.emailaddress):

            return True

        else:

            False


    @staticmethod

    def get_customer_by_email(email):

        return Customer.objects.get(emailaddress=email)


class Login(models.Model):

    userid=models.CharField(max_length=50,primary_key=True)

    password=models.CharField(max_length=20)

    usertype=models.CharField(max_length=30)
```

```
class Category(models.Model):
```

```
    name= models.CharField(max_length=50)
```

```
    def __str__(self):
```

```
        return self.name
```

```
    @staticmethod
```

```
    def get_all_category():
```

```
        return Category.objects.all()
```

```
class Product(models.Model):
```

```
    name= models.CharField(max_length=50)
```

```
    price= models.IntegerField(default=0)
```

```
    desc =models.CharField(max_length=200)
```

```
    category= models.ForeignKey(Category, on_delete=models.CASCADE, default=1)
```

```
    image= models.ImageField(upload_to='newproducts/')
```

```
    @staticmethod
```

```
    def get_products_by_categoryid(category_id):
```

```
        return Product.objects.filter(category=category_id)
```

```
    @staticmethod
```

```
    def get_all_products():
```

```
        return Product.objects.all()
```

```
    @staticmethod
```

```
    def get_product_by_id(productid):
```

```
        return Product.objects.get(id=productid)
```

```
class Orders(models.Model):
```



```
customer=models.ForeignKey(Customer,on_delete=models.DO_NOTHING)

product=models.ForeignKey(Product,on_delete=models.DO_NOTHING)

price=models.IntegerField()

address=models.CharField(max_length=200)

pincode=models.CharField(max_length=10)

quantity= models.IntegerField()

order_date=models.DateField(default=datetime.datetime.today)

completed= models.BooleanField(default=False)
```

```
class ShoppingCart(models.Model):

    customer = models.ForeignKey(Customer, on_delete=models.DO_NOTHING)

    product = models.ForeignKey(Product, on_delete=models.DO_NOTHING)

    quantity = models.IntegerField()
```

admin.py

```
from django.contrib import admin

from . models import Login, Customer, Product, Category, ShoppingCart, Orders
```

```
class AdminProduct(admin.ModelAdmin):

    list_display=['name','price','desc','image']

class AdminCategory(admin.ModelAdmin):

    list_display=['name']
```

```
class AdminCustomer(admin.ModelAdmin):

    list_display = ['name', 'emailaddress', 'pincode', 'regdate']

    def has_add_permission(self, request, obj=None):

        return False

    def has_change_permission(self, request, obj=None):

        return False
```

```

class AdminOrders(admin.ModelAdmin):
    list_display=['customer','product','price','quantity','order_date']

class AdminShoppingCart(admin.ModelAdmin):
    list_display=['customer', 'product', 'quantity']

admin.site.register(Login)
admin.site.register(Customer,AdminCustomer)
admin.site.register(Product,AdminProduct)
admin.site.register(Category,AdminCategory)
admin.site.register(ShoppingCart, AdminShoppingCart)
admin.site.register(Orders, AdminOrders)

```

ossappurls.py

```

from django.urls import path
from . import views

urlpatterns=[
    path("",views.index,name='index'),
    path('about/',views.about,name='about'),
    path('product/',views.productPage,name='product'),
    path('register/', views.register, name='register'),
    path('login/', views.signin, name='login'),
    path('user/',views.user, name='user'),
    path('orders/',views.orders, name='orders'),
    path('logout/', views.signout, name='logout'),
    path('buynow/', views.buynow, name='buynow'),
    path('checkout/', views.checkout, name='checkout'),
    path('changepass/', views.changepass, name='changepass'),
    path('mycart/', views.mycart, name='mycart'),
    path('update_cart/',views.update_cart, name='update_cart'),
]

```

views.py

```
from django.shortcuts import render, redirect
from django.contrib.auth.hashers import make_password, check_password
from . models import Product, Category, Customer,Orders,ShoppingCart
from django.contrib.auth.models import User
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.decorators import login_required
```

```
def index(request):
```

```
    return render(request,'index.html')
```

```
def about(request):
```

```
    return render(request,'about.html')
```

```
def productPage(request):
```

```
    if request.method == "POST":
```

```
        return redirect('product')
```

```
    else:
```

```
        category_id=request.GET.get('category')
```

```
        data = {}
```

```
        if category_id:
```

```
            products= Product.get_products_by_categoryid(category_id)
```

```
        else:
```

```
            products = Product.get_all_products()
```

```
            category = Category.get_all_category()
```

```
            data['products'] = products
```

```
            data['category'] = category
```

```
            return render(request, 'product.html', data)
```

```
def register(request):
```

```
if request.method == 'POST':

    err=None

    name= request.POST.get('name')
    email = request.POST.get('email')
    passwd = request.POST.get('password')
    addr = request.POST.get('address')
    pin = request.POST.get('pincode')
    phone = request.POST.get('phone')
    gen = request.POST.get('gender')


    #Validation
    values={'name':name,
            'email':email,
            'addr':addr,
            'pin':pin,
            'phone':phone,
            'gen':gen,}

    customer = Customer(name=name, gender=gen, address=addr, pincode=pin, contactno=phone,
                        emailaddress=email,
                        password=passwd)


    if not name.isalpha():
        err="Invalid Name, please try again"

    if not phone.isnumeric() or len(phone)<10:
        err="Invalid Contact Number, please try again"

    if not pin.isnumeric():
        err="Invalid Pincode, please try again"

    if customer.is_exists():
        err="Email Already Exists"

    data={}
    data['err']=err
    data['values']=values
```

```

if err:

    return render(request, 'register.html',data)

#customer.password=make_password(passwd)

customer.save()

myuser = User.objects.create_user(username=email,email=email, password=passwd)

myuser.save()


err = "You are registered! Try Logging in"

return render(request, 'login.html',{'err':err})

else:

    return render(request, 'register.html')


def signin(request):

    if request.method == 'POST':

        email = request.POST.get('email')

        password = request.POST.get('password')

        myuser = authenticate(username=email, password=password)

        if myuser is not None:

            login(request, myuser)

            try:

cart=ShoppingCart.objects.filter(customer=Customer.objects.get(emailaddress=request.user.username
))

                except Exception as e:

                    print(e)

                    cart=[]

                    request.session['cart']=len(cart)

                    return redirect('user')

            else:

                msg="Incorrect Id or password"

                return render(request,'login.html',{'msg':msg})

    return render(request,'login.html')

```

```

@login_required(redirect_field_name='login')
def user(request):
    customer = Customer.get_customer_by_email(request.user.username)
    data={}
    if request.method=='POST':
        msg=None
        err=None
        current_password=request.POST.get('current_password')
        new_password = request.POST.get('new_password')
        confirm_password = request.POST.get('confirm_password')

        if customer.password==current_password:
            if new_password==confirm_password:
                customer.password=new_password
                customer.save()
                myuser=User.objects.get(username=request.user.username)
                myuser.password=make_password(new_password)
                myuser.save()
                data['msg']='Password Changed Successfully'

            else:
                return render(request, 'changepass.html', {'err': 'New password and confirm password does not match'})
        else:
            return render(request,'changepass.html',{'err':'Incorrect Password'})
    data['customer']=customer
    return render(request,'user.html',data)

@login_required(redirect_field_name='login')
def orders(request):
    orders=Orders.objects.filter(customer=request.user.username).order_by('-order_date')
    return render(request,'orders.html',{'orders':orders})

```

```
@login_required(redirect_field_name='login')
```

```
def changepass(request):
```

```
    return render(request,'changepass.html')
```

```
def signout(request):
```

```
    logout(request)
```

```
    request.session.flush()
```

```
    return redirect('index')
```

```
@login_required(redirect_field_name='login')
```

```
def buynow(request):
```

```
    data={}
```

```
    if request.method=='GET':
```

```
        total=request.GET.get('total')
```

```
        if int(total) < 1:
```

```
            err="Total cannot be zero"
```

```
            return redirect('/mycart/?err='+err)
```

```
        method = "add_to_cart"
```

```
        customer = Customer.get_customer_by_email(request.user.username)
```

```
        cart=ShoppingCart.objects.filter(customer=customer)
```

```
        data['cart']=cart
```

```
    else:
```

```
        method = "buy_now"
```

```
        productid=request.POST.get('product')
```

```
        #print(productid)
```

```
        product=Product.get_product_by_id(productid)
```

```
        email=request.user.username
```

```
        customer=Customer.get_customer_by_email(email)
```

```
        data['product']=product
```

```

        data['customer']=customer
    data['method']=method
    return render(request,'confirm.html',data)

@login_required(redirect_field_name='login')
def checkout(request):
    if request.method=='GET':
        customer = Customer.get_customer_by_email(request.user.username)
        cart = ShoppingCart.objects.filter(customer=customer)
        for items in cart:
            order = Orders(customer=items.customer, product=items.product, quantity=items.quantity,
                            price=items.product.price, address=items.customer.address,
                            pincode=items.customer.pincode)
            order.save()
        ShoppingCart.objects.filter(customer=Customer.get_customer_by_email(request.user.username)).delete()
        request.session['cart'] = 0

    else:
        product=Product.get_product_by_id(request.POST.get('productid'))
        customer =Customer.get_customer_by_email(request.POST.get('email'))
        quantity = request.POST.get('quantity')
        order=Orders(customer=customer,product=product,price=product.price,address=customer.address,pin
        code=customer.pincode,quantity=quantity)
        order.save()

    return render(request,'checkout.html')

@login_required(redirect_field_name='login')
def mycart(request):
    if request.method == "GET" and request.GET.get('flag') == "add_to_cart":
        category=request.GET.get('category')

```



```

product = Product.objects.get(id=request.GET.get('product_id'))
customer = Customer.get_customer_by_email(request.user.username)
cart = ShoppingCart(customer=customer, product=product, quantity=1)
cart.save()
print(request.session.get('cart'))
request.session['cart']=request.session.get('cart')+1
msg="Product Added to Cart"
#return render(request, 'product.html', {'msg':msg})
return redirect('/product/?msg='+msg+'&category='+category)
cart=ShoppingCart.objects.filter(customer=Customer.objects.get(emailaddress=request.user.username
))
total=sum([cartitem.product.price*cartitem.quantity for cartitem in cart])
return render(request,'mycart.html',{'cart':cart,'total':total})

@login_required(redirect_field_name='login')
def update_cart(request):
    err=None
    if request.method=='POST':
        cart_item_id=request.POST.get('cart_item_id')
        quantity=request.POST.get('quantity')
        if int(quantity) > 0:
            cart=ShoppingCart.objects.get(id=cart_item_id)
            cart.quantity=quantity
            cart.save()
            return redirect('mycart')
        else:
            err="Quantity cannot be less than 1"
    elif request.method=='GET' and request.GET.get('cart_item'):
        cart_item_id=request.GET.get('cart_item')
        try:
            ShoppingCart.objects.get(id=cart_item_id).delete()
            request.session['cart'] = request.session.get('cart') -1

```

```

        except Exception as e:
            print(e)
            pass
    else:
        try:
ShoppingCart.objects.filter(customer=Customer.get_customer_by_email(request.user.username)).delete()

            request.session['cart'] = 0
            err="Cart Empty"
        except Exception as e:
            pass
    if err:
        return redirect('/mycart/?err='+err)
    else:
        return redirect('/mycart/')

```

base.html

```

<!doctype html>
<html lang="en">
{% load static %}
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="{% static 'css/bootstrap.css' %}">
    <link rel="stylesheet" href="{% static 'css/style.css' %}">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.2/css/all.min.css" integrity="sha512-1sCRPdkRXhBV2PBLUdRb4tMg1w2YPf37qatUFeS7zlBy7jJI8Lf4VHwWfZZfpXtYSLy85pkm9GaYVYMfw5BC1A==" crossorigin="anonymous" referrerpolicy="no-referrer" />

    <script src="{% static 'js/bootstrap.bundle.js' %}"></script>

```

```
<title>{% block title %} {% endblock %}</title>
</head>
<body>
<div class="container-fluid">
  <!-- outer start -->
  {% include 'navbar.html' %}

  {% block content %}
  {% endblock %}

  {% include 'footer.html' %}
</div>
</body>
</html>
```

changepass.html

```
{% extends "userbase.html" %}
{% block page_title %}
Change Password
{% endblock %}
{% block user %}
{% if err %}
<div class="alert alert-danger" role="alert">
  {{err}}
</div>
{% endif %}
<div class="row">
  <div class="col-sm">
    <form action="{% url 'user' %}" method="post">
      {% csrf_token %}
      <div class="row mb-3">
        <label for="current_password" class="col-sm-2 col-form-label">Current Password</label>
```

```

        <div class="col-sm-6">
            <input type="password" name="current_password" class="form-control"
id="current_password">
        </div>
    </div>
    <div class="row mb-3">
        <label for="new_password" class="col-sm-2 col-form-label">New Password</label>
        <div class="col-sm-6">
            <input type="password" name="new_password" class="form-control"
id="new_password">
        </div>
    </div>
    <div class="row mb-3">
        <label for="confirm_password" class="col-sm-2 col-form-label">Confirm
Password</label>
        <div class="col-sm-6">
            <input type="password" name="confirm_password" class="form-control"
id="confirm_password">
        </div>
    </div>
    <div class="row mb-3" >
        <div class="col-sm-2"></div>
        <div class="col-sm-6 d-grid" >
            <button type="submit" class="btn btn-warning float-mid" >Change Password</button>
        </div>
    </div>

</form>
</div>
</div>
{% endblock %}

confirm.html
{{product.name}}

```

```

{{product.price}}

{{product.image.url}}

{{customer.address}}

{{customer.pincod}}

{% if method == "add_to_cart" %}

{% for item in cart %}

{{item.product.name}}

{% endfor %}

<a href="{% url 'checkout' %}">Confirm Order</a>

{% else %}

<form action="{% url 'checkout' %}" method="post">

    {% csrf_token %}

    <input type="text" name="productid" value="{{product.id}}" hidden>

    <br><input type="text" name="email" value="{{customer.emailaddress}}" hidden>

    <br><input type="text" name="price" value="{{product.price}}" >

    <br><input type="text" name="address" value="{{customer.address}}" >

    <br><input type="text" name="pincod" value="{{customer.pincod}}" >

    <br><input type="text" name="quantity" value="1" hidden>

    <br><button type="submit" class="btn btn-warning"> Checkout</button>

</form>

{% endif %}

```

index.html

```

<!doctype html>

<html lang="en">

{% load static %}

<head>

    <meta charset="utf-8">

```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<link rel="stylesheet" href="{% static 'css/bootstrap.css' %}">
```

```
<link rel="stylesheet" href="{% static 'css/style.css' %}">
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.2/css/all.min.css" integrity="sha512-1sCRPdkRXhBV2PBLUdRb4tMg1w2YPf37qatUFeS7zlBy7jJI8Lf4VHwWfZZfpXtYSLy85pkm9GaYVYMfw5BC1A==" crossorigin="anonymous" referrerpolicy="no-referrer" />
```

```
<script src="{% static 'js/bootstrap.bundle.js' %}"></script>
```

```
<title>Online Shopping System</title>
```

```
</head>
```

```
<body>
```

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

```
<!-- outer start -->
```

```
{% include "navbar.html" %}
```

```
<!-- slider start -->
```

```
<div class="row">
```

```
<div class="col-sm-12 p-0">
```

```
<div id="carouselExampleCaptions" class="carousel slide" data-bs-ride="false">
```

```
<div class="carousel-indicators">
```

```
<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>
```

```
<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="1" aria-label="Slide 2"></button>
```

```
<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="2" aria-label="Slide 3"></button>
```

```
</div>
```

```
<div class="carousel-inner">
```

```
<div class="carousel-item active">
```

```

    
    <div class="carousel-caption d-none d-md-block">
        <h5>First slide label</h5>
        <p>Some representative placeholder content for the first slide.</p>
    </div>
</div>
<div class="carousel-item">
    
    <div class="carousel-caption d-none d-md-block">
        <h5>Second slide label</h5>
        <p>Some representative placeholder content for the second slide.</p>
    </div>
</div>
<div class="carousel-item">
    
    <div class="carousel-caption d-none d-md-block">
        <h5>Third slide label</h5>
        <p>Some representative placeholder content for the third slide.</p>
    </div>
</div>
<button class="carousel-control-prev" type="button" data-bs-
target="#carouselExampleCaptions" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
</button>
<button class="carousel-control-next" type="button" data-bs-
target="#carouselExampleCaptions" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
</button>
</div>
</div>

```

</div>

<!-- slider end -->

<!-- section start -->

<div class="row mt-3">

<div class="col-sm-3 text-center">

<div class="section shadow-sm">

</div>

<button type="button" class="btn btn-warning mt-2">Warning</button>

</div>

<div class="col-sm-3 text-center">

<div class="section shadow-sm">

</div>

<button type="button" class="btn btn-warning mt-2">Warning</button>

</div>

<div class="col-sm-3 text-center">

<div class="section shadow-sm">

</div>

<button type="button" class="btn btn-warning mt-2">Warning</button>

</div>

<div class="col-sm-3 text-center">

<div class="section shadow-lg">

</div>

<button type="button" class="btn btn-warning mt-2">Warning</button>

</div>

</div>


```
<!-- section end -->
```

```
<!-- product start -->
```

```
<div class="row mt-5">
```

```
  <div class="col-sm-3">
```

```
    <div class="card">
```

```
      
```

```
      <div class="card-body">
```

```
        <h5 class="card-title">Rs.578 /-</h5>
```

```
        <p class="fw-bold">fruit vinegar</p>
```

```
        <small class="card-text d-block">Some quick example text to build on the card title  
and make up the bulk of the card's content.</small>
```

```
        <div class="mt-2">
```

```
          <a href="#" class="btn btn-warning">Add to cart</a>
```

```
          <a href="#" class="btn btn-warning float-end">Buy Now</a>
```

```
        </div>
```

```
      </div>
```

```
    </div>
```

```
  </div>
```

```
<div class="col-sm-3">
```

```
  <div class="card">
```

```
    
```

```
    <div class="card-body">
```

```
      <h5 class="card-title">Rs.500 /-</h5>
```

```
      <p class="fw-bold">fruit vinegar</p>
```

```
      <small class="card-text d-block">Some quick example text to build on the card title  
and make up the bulk of the card's content.</small>
```

```
      <div class="mt-2">
```

```
        <a href="#" class="btn btn-warning">Add to cart</a>
```

```
        <a href="#" class="btn btn-warning float-end">Buy Now</a>
```

```
      </div>
```

```

        </div>
    </div>
</div>
<div class="col-sm-3">
    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Rs.488 /-</h5>
            <p class="fw-bold">fruit vinegar</p>
            <small class="card-text d-block">Some quick example text to build on the card title
and make up the bulk of the card's content.</small>
            <div class="mt-2">
                <a href="#" class="btn btn-warning">Add to cart</a>
                <a href="#" class="btn btn-warning float-end">Buy Now</a>
            </div>
        </div>
    </div>
</div>
<div class="col-sm-3">
    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">Rs.987 /-</h5>
            <p class="fw-bold">fruit vinegar</p>
            <small class="card-text d-block">Some quick example text to build on the card title
and make up the bulk of the card's content.</small>
            <div class="mt-2">
                <a href="#" class="btn btn-warning">Add to cart</a>
                <a href="#" class="btn btn-warning float-end">Buy Now</a>
            </div>
        </div>
    </div>
</div>
</div>

```

</div>

<div class="row mt-5">

<div class="col-sm-3">

<div class="card">

<div class="card-body">

<h5 class="card-title">Rs.578 /-</h5>

<p class="fw-bold">fruit vinegar</p>

<small class="card-text d-block">Some quick example text to build on the card title and make up the bulk of the card's content.</small>

<div class="mt-2">

Add to cart

Buy Now

</div>

</div>

</div>

</div>

<div class="col-sm-3">

<div class="card">

<div class="card-body">

<h5 class="card-title">Rs.500 /-</h5>

<p class="fw-bold">fruit vinegar</p>

<small class="card-text d-block">Some quick example text to build on the card title and make up the bulk of the card's content.</small>

<div class="mt-2">

Add to cart

Buy Now

</div>

</div>

</div>

</div>

```

<div class="col-sm-3">

  <div class="card">

    <div class="card-body">

      <h5 class="card-title">Rs.488 /-</h5>

      <p class="fw-bold">fruit vinegar</p>

      <small class="card-text d-block">Some quick example text to build on the card title
and make up the bulk of the card's content.</small>

      <div class="mt-2">

        <a href="#" class="btn btn-warning">Add to cart</a>

        <a href="#" class="btn btn-warning float-end">Buy Now</a>

      </div>

    </div>

  </div>

</div>

<div class="col-sm-3">

  <div class="card">

    <div class="card-body">

      <h5 class="card-title">Rs.987 /-</h5>

      <p class="fw-bold">fruit vinegar</p>

      <small class="card-text d-block">Some quick example text to build on the card title
and make up the bulk of the card's content.</small>

      <div class="mt-2">

        <a href="#" class="btn btn-warning">Add to cart</a>

        <a href="#" class="btn btn-warning float-end">Buy Now</a>

      </div>

    </div>

  </div>

</div>

<!-- product end -->

{% include "footer.html" %}

```

```
<!-- outer end -->

</div>

</body>
</html>
```

login.html

```
{% extends "base.html" %}

{% block title %}
Login!
{% endblock %}

{% block content %}

<div class="row justify-content-md-center mt-5">
  <div class="col-sm-4">

    <h2 class="text-center">Login!</h2>
    {% if err %}
    <div class="alert alert-success" role="alert">
      {{err}}
    </div>
    {% endif %}

    {% if msg %}
    <div class="alert alert-danger" role="alert">
      {{msg}}
    </div>
    {% endif %}

    <form class="mt-5" action="{% url 'login' %}" method="post">
      {% csrf_token %}
      <div class="row mb-3">
```

```

        <label for="email" class="col-sm-2 col-form-label">Email</label>

        <div class="col-sm-10">
            <input name="email" type="email" class="form-control" id="email">
        </div>
    </div>

    <div class="row mb-3">
        <label for="password" class="col-sm-2 col-form-label">Password</label>
        <div class="col-sm-10">
            <input name="password" type="password" class="form-control" id="password">
        </div>
    </div>

    <div class="form-check">
        <input class="form-check-input" type="checkbox" value="" id="rememberme">
        <label class="form-check-label" for="rememberme">
            Remember Me
        </label>
    </div>

    <div class="text-center">
        <button type="submit" class="btn btn-warning ">Login</button>
    </div>

</form>

</div>
</div>
{% endblock %}

```

modal.html

```

<!doctype html>
<html lang="en">
{% load static %}
<head>

```

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="{% static 'css/bootstrap.css' %}">

<link rel="stylesheet" href="{% static 'css/style.css' %}">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.2/css/all.min.css" integrity="sha512-1sCRPdkRXhBV2PBLUdRb4tMg1w2YPf37qatUFeS7zlBy7jJI8Lf4VHwWfZZfpXtYSLy85pkm9GaYVYMfw5BC1A==" crossorigin="anonymous" referrerpolicy="no-referrer" />

<script src="{% static 'js/bootstrap.bundle.js' %}"></script>

<title>{% block title %}{% endblock %}</title>

</head>

<body>

<div class="container-fluid">

<!-- headline start -->

<div class="row bg-warning pt-2">

<div class="col-sm-3 pt-2"><i class="fa-solid fa-phone"></i> Call Us: +91 1111111111</div>

<div class="col-sm-3 pt-2"><i class="fa-solid fa-envelope"></i> Mail Us:
email@gmail.com</div>

<div class="col-sm-6 text-end">

<!-- =====social media icons===== -->

<ul class="smi">

<i class="fa-brands fa-facebook-f"></i>

<i class="fa-brands fa-instagram"></i>

<i class="fa-brands fa-twitter"></i>

<i class="fa-brands fa-google"></i>

</div>

</div>

```
<!-- headline end -->

<!-- navbar start -->

<div class="row">

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

    <nav class="navbar navbar-expand-lg bg-light">

      <div class="container-fluid">

        <a class="navbar-brand" href="#"></a>

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

          <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">

          <ul class="navbar-nav me-auto mb-2 mb-lg-0">

            <li class="nav-item">

              <a class="nav-link active" aria-current="page" href="{% url 'index'
%}">Home</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{% url 'about' %}">About</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{% url 'product' %}">Products</a>

            </li>

            <li class="nav-item" data-bs-toggle="modal" data-bs-target="#staticBackdrop">

              <a class="nav-link" href="#">Register</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{% url 'login' %}">Login</a>

            </li>
```



```

        </ul>

        <form class="d-flex" role="search">

            <input class="form-control me-2" type="search" placeholder="Search" aria-
label="Search">

            <button class="btn btn-outline-success" type="submit">Search</button>

        </form>

    </div>

</div>

</nav>

</div>

</div>

<!-- Modal -->

<div class="modal fade" id="staticBackdrop" data-bs-backdrop="static" data-bs-keyboard="false"
tabindex="-1" aria-labelledby="staticBackdropLabel" aria-hidden="true">

    <div class="modal-dialog">

        <div class="modal-content">

            <div class="modal-header">

                <h5 class="modal-title" id="staticBackdropLabel">Modal title</h5>

                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>

            </div>

            <div class="modal-body">

                ...

            </div>

            <div class="modal-footer">

                <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>

                <button type="button" class="btn btn-primary">Understood</button>

            </div>

        </div>

    </div>

</div>

<!-- navbar end -->

```

```
{% include 'footer.html' %}  
</div>
```

```
</body>
```

```
</html>
```

mycart.html

```
{% extends "userbase.html" %}  
  
{% block page_title %}  
My Cart  
{% endblock %}  
  
{% block user %}  
  
{% if request.GET.err %}  
<div class="alert alert-danger" role="alert">  
    {{request.GET.err}}  
</div>  
  
{% endif %}  
  
<div class="row">  
    <div class="col-sm-9">  
        <table class="table">  
            <thead>  
                <tr>  
                    <th scope="col"></th>  
                    <th scope="col">Product</th>  
                    <th scope="col">Price</th>  
                    <th scope="col">Quantity</th>  
  
                </tr>  
            </thead>  
            <tbody>  
  
                {% for cart_item in cart %}
```

```

<tr>

    <td></td>

    <td>{{cart_item.product.name}}</td>

    <td>Rs. {{cart_item.product.price}}</td>

    <td>

        <form action="{{url 'update_cart' }}" method="post">

            {% csrf_token %}

            <input type="number" class="form-control" name="quantity"
value="{{cart_item.quantity}}">

            <input type="text" name="cart_item_id" value="{{cart_item.id}}" hidden>

            <button type="submit" class="badge bg-warning text-light"
style="border:none">Update</button>

        </form>

    </td>

    <td>

        <a href="/update_cart/?cart_item={{cart_item.id}}" class="badge bg-danger"
style="text-decoration:none">Remove</a>

    </td>

</tr>

{% endfor %}

</tbody>

</table>

<div class="row">

    <div class="col-sm">

        <a href="{{url 'update_cart' }}" class="btn btn-warning text-light fw-bold float-end">
Empty Cart</a>

    </div>

</div>

</div>

<div class="col-sm border-light rounded bg-light text-dark">

<h4>

    Cart Sub Total:

```

```

        <small class="text-muted">Rs. {{total}}</small>

    </h4>

    <div class="row">

        <div class="col-sm d-grid" >

            <a href="/buynow/?total={{total}}" class="btn btn-warning text-light fw-bold float-right"
>Checkout</a>

        </div>

    </div>

</div>

</div>

{% endblock %}

```

navbar.html

```

{% load static %}

<!-- headline start -->

<div class="row bg-warning pt-2">

    <div class="col-sm-3 pt-2"><i class="fa-solid fa-phone"></i> Call Us: +91 1111111111</div>

    <div class="col-sm-3 pt-2"><i class="fa-solid fa-envelope"></i> Mail Us:
email@gmail.com</div>

    <div class="col-sm-6 text-end">

        <!-- =====social media icons===== -->

        <ul class="smi">

            <li><i class="fa-brands fa-facebook-f"></i></li>

            <li><i class="fa-brands fa-instagram"></i></li>

            <li><i class="fa-brands fa-twitter"></i></li>

            <li><i class="fa-brands fa-google"></i></li>

        </ul>

    </div>

</div>

<!-- headline end -->

<!-- navbar start -->

```

```
<div class="row">

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

    <nav class="navbar navbar-expand-lg bg-light">

      <div class="container-fluid">

        <a class="navbar-brand" href="#"></a>

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

          <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">

          <ul class="navbar-nav me-auto mb-2 mb-lg-0">

            <li class="nav-item">

              <a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{% url 'about' %}">About</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{% url 'product' %}">Products</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{% url 'register' %}">Register</a>

            </li>

            {% if user.is_authenticated %}

            <li class="nav-item">

              <a class="nav-link" href="{% url 'user' %}">My Account</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{% url 'logout' %}">Logout</a>

            </li>

          </ul>

        </div>

      </div>

    </div>

  </div>

</div>
```

```

        {% else %}

        <li class="nav-item">

            <a class="nav-link" href="{% url 'login' %}">Login</a>

        </li>

        {% endif %}

    </ul>

    <div class="">

        {% if user.is_authenticated %}

        <form action="{% url 'mycart' %}">

            <button type="submit" class="btn btn-dark position-relative">

                <i class="fa fa-shopping-cart large" aria-hidden="true"></i>

                {% if request.session.cart > 0 %}

                <span class="position-absolute top-0 start-100 translate-middle badge rounded-pill bg-
danger">

                    {{ request.session.cart }}

                <span class="visually-hidden">unread messages</span>

                {% endif %}

            </span>

            </button>

        </form>

        {% endif %}

    </div>

    <form class="d-flex" role="search">

        <input class="form-control me-2" type="search" placeholder="Search" aria-
label="Search">

        <button class="btn btn-outline-success" type="submit">Search</button>

    </form>

</div>

</div>

```

```
</nav>

</div>

</div>

<!-- navbar end -->
```

orders.html

```
{% extends "userbase.html" %}

{% block page_title %}

My Orders

{% endblock %}

{% block user %}

<div class="row">

  <div class="col-sm">

    <table class="table">

      <thead>

        <tr>

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

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

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

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

          <th scope="col">Order Date</th>

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

        </tr>

      </thead>

      <tbody>

        {% for order in orders %}

        <tr>

          <td></td>

          <td>{{ order.product.name }}</td>

          <td>{{ order.price }}</td>

          <td>{{ order.quantity }}</td>
```

```

        <td>{{order.order_date}}</td>

        <td>{% if not order.completed %}

            <span class="badge bg-warning text-dark">Pending</span>

            {% else %}

            <span class="badge bg-success ">Completed</span>

            {% endif %}

        </td>

    </tr>

{% endfor %}

</tbody>

</table>

</div>

</div>

{% endblock %}

```

products.html

```

{% extends "base.html" %}

{% block title %}

Products

{% endblock %}

{% block content %}

<!--

{% for x in products %}

<h4>{{x.name}} {{x.price}} {{x.image}}</h4>

{% endfor %}

-->

<!--

{% for ctg in category %}

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

{% endfor %}

-->

```



```

{% if request.GET.msg %}

<div class="row">

  <div class="col-sm">

    <svg xmlns="http://www.w3.org/2000/svg" style="display: none;">

      <symbol id="check-circle-fill" fill="currentColor" viewBox="0 0 16 16">

        <path d="M16 8A8 8 0 1 1 0 8a8 8 0 1 16 0zm-3.97-3.03a.75.75 0 0 0-1.08.022L7.477
9.417 5.384 7.323a.75.75 0 0 0-1.06 1.06L6.97 11.03a.75.75 0 0 0 1.079-.0213.992-4.99a.75.75 0 0 0-
.01-1.05z"/>

      </symbol>

    </svg>

    <div class="alert alert-success d-flex align-items-center" role="alert">

      <svg class="bi flex-shrink-0 me-2" width="24" height="24" role="img" aria-
label="Success:"><use xlink:href="#check-circle-fill"/></svg>

      <div>

        {{request.GET.msg}}

      </div>

    </div>

  </div>

</div>

{% endif %}

<div class="row mt-5">

  <div class="col-sm-2">

    <div class="list-group">

      <a href="/product?" class="list-group-item list-group-item-action">All Products</a>

      {% for ctg in category %}

        <a href="/product?category={{ctg.id}}" class="list-group-item list-group-item-
action">{{ctg.name}}</a>

      {% endfor %}

    </div>

  </div>

</div>

```

```

<div class="col-sm">

  <div class="row">

    {% for product in products %}

      <div class="col-sm-auto mb-5" >

        <div class="card" style="width: 18rem;">

          <div class="card-body">

            <h5 class="card-title">{{product.name}}</h5>

            <p class="card-text">Rs. {{product.price}}</p>

            <p class="card-text">{{product.desc}}</p>

            <div class="row">

              <div class="col-sm">

                <a
href="/mycart?product_id={{product.id}}&flag=add_to_cart&category={{request.GET.category}}"
class="btn btn-warning" value="">Add To Card</a>

              </div>

              <div class="col-sm">

                <form action="{% url 'buynow' %}" method="post">

                  {% csrf_token %}

                  <input type="text" name="product" value="{{product.id}}" hidden >

                  <button type="submit" class="btn btn-warning float-end">Buy Now</button>

                </form></div>

              </div>

            </div>

          </div>

        </div>

      {% endfor %}

    </div>

  </div>

  </div>

  {% endblock %}

```

register.html

{% extends "base.html" %}

{% block title %}

Sign Up!

{% endblock %}

{% block content %}

<div class="row justify-content-md-center mt-5">

<div class="col-sm-4">

<h2 class="text-center">Sign Up!</h2>

{% if err %}

<div class="alert alert-danger" role="alert">

{{err}}

</div>

{% endif %}

<form class="mt-5" action="{% url 'register' %}" method="post">

{% csrf_token %}

<div class="row mb-3">

<label for="name" class="col-sm-2 col-form-label">Name</label>

<div class="col-sm-10">

<input name="name" type="text" class="form-control" id="name"
value="{{ values.name }}" required>

</div>

</div>

<div class="row mb-3">

```
<label for="email" class="col-sm-2 col-form-label">Email</label>

<div class="col-sm-10">

    <input name="email" type="email" class="form-control" id="email"
value="{{values.email}}" required>

</div>

</div>

<div class="row mb-3">

    <label for="password" class="col-sm-2 col-form-label">Password</label>

    <div class="col-sm-10">

        <input name="password" type="password" class="form-control" id="password"
required>

    </div>

</div>

<div class="row mb-3">

    <label name='address' for="address" class="col-sm-2 form-label">Address</label>

    <div class="col-sm-10">

        <textarea name="address" class="form-control" id="address" rows="3"
required>{{values.addr}}</textarea>

    </div>

</div>

<div class="row mb-3">

    <label for="pincode" class="col-sm-2 col-form-label">Pincode</label>

    <div class="col-sm-10">

        <input name="pincode" type="text" value="{{values.pin}}" class="form-control"
id="pincode" required>

    </div>

</div>

<div class="row mb-3">

    <label for="phone" class="col-sm-2 col-form-label">Contact Number</label>

    <div class="col-sm-10">

        <input name="phone" type="text" class="form-control" value="{{values.phone}}"
id="phone" required>

    </div>

</div>
```

```

<fieldset class="row mb-3">
  <legend class="col-form-label col-sm-2 pt-0">Gender</legend>
  <div class="col-sm-10">
    <div class="form-check">
      <input class="form-check-input" type="radio" name="gender" id="gender1"
value="male" required >
      <label class="form-check-label" for="gender1">
        Male
      </label>
    </div>
    <div class="form-check">
      <input class="form-check-input" type="radio" name="gender" id="gender2"
value="female" required>
      <label class="form-check-label" for="gender2">
        Female
      </label>
    </div>
  </div>
</fieldset>

```

```

<div class="text-center">
  <button type="submit" class="btn btn-warning ">Sign Up</button>
</div>

```

```

</form>

```

```

</div>
</div>
{% endblock %}

```

user.html

```

{% extends "userbase.html" %}

```

{% block page_title %}

My Account

{% endblock %}

{% block user %}

{% if msg %}

<div class="alert alert-success" role="alert">

 {{msg}}

</div>

{% endif %}

<div class="row justify-content-center">

 <div class="col-sm-6">

 <table class="table" >

 <tbody>

 <tr>

 <td class="fw-bold">Name</td><td class="text-uppercase">{{customer.name}}</td>

 </tr>

 <tr>

 <td class="fw-bold">Email</td><td>{{customer.emailaddress}}</td>

 </tr>

 <tr>

 <td class="fw-bold">Address</td><td class="text-uppercase">{{customer.address}}</td>

 </tr>

 <tr>

 <td class="fw-bold">Pincode</td><td class="text-uppercase">{{customer.prcode}}</td>

 </tr>

 <tr>

 <td class="fw-bold">Gender</td><td class="text-uppercase">{{customer.gender}}</td>

 </tr>

 <tr>

 <td class="fw-bold">Registration Date</td><td class="text-uppercase">{{customer.regdate}}</td>

 </tr>

```

        <tr>

            <td class="fw-bold">Password</td><td><a href="{% url 'changepass' %}" class="btn btn-
warning">Change Password</a></td>

        </tr>

    </tbody>

</table>

</div>

</div>

```

```
{% endblock %}
```

userbase.html

```

{% extends "base.html" %}

{% block title %}

Welcome User

{% endblock %}

{% block content %}

<div class="row justify-content-center mt-5">

    <div class="col-sm-6 text-center">

        <h4 class="text-uppercase">

            {% block page_title %}

            {% endblock %}

        </h4>

    </div>

</div>

<div class="row mt-5">

    <div class="col-sm-2">

        <div class="list-group">

            <a href="{% url 'user' %}" class="list-group-item list-group-item-action"> My Account </a>

            <a href="{% url 'orders' %}" class="list-group-item list-group-item-action"> My Orders </a>

            <a href="{% url 'mycart' %}" class="list-group-item list-group-item-action"> My Cart </a>

```

```
<a href="{% url 'logout' %}" class="list-group-item list-group-item-action">Logout </a>
</div>
</div>

<div class="col-sm">

    {% block user %}
    {% endblock %}
</div>

</div>

{% endblock %}
```

Conclusion & Future Scope

The project entitled Online shopping system was completed successfully.

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 and an android 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 html & CSS, usage of responsive templates, designing of Web applications, and management of database using MySQL. The entire system is secured. The project also helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project.

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

There is a scope for further development in our project to a great extent. A number of features can be added to this system in future like providing moderator 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 history of purchases of each customer and provide suggestions based on their history. These features could have implemented unless the time did not limit us.