

A PROJECT REPORT

On

“LEXA”

A Dissertation Report

Submitted in partial fulfillment of the requirements for the award of the degree of

“BACHELOR OF COMPUTER APPLICATION”



BENGALURU NORTH UNIVERSITY
2022

Submitted by:

Mr. Allen Albino
Mr. Prince M Varghese
Miss. Meghna M.K

Register Number:

R1918412
R1918486
R1918459

Under the guidance of

Prof. Shohinee Mandel





CERTIFICATE

This is to certify that **Mr. Allen Albino (R1918412), Mr. Prince M Varghese (R1918486) Miss. Meghna MK(R1918459)** of Koshy's Institute of Management Studies is confide students of thiscollege. They are conducted a project entitled “Flight Booking System” is an original study conducted by them.

The project undertaken by the candidate has been executed by him under the guidance of **Prof. Shohinee Mandel**, Koshy's Institute of Management Studies. This project report has not formed a basis for the award of the Degree of Bangalore University or any University.

Signature of HOD

Signature of Principal



GUIDE CERTIFICATE

This is to certify that the project entitled “**LEXA**” prepared by **Mr. Allen Albino (R1918412), Mr. Prince M Varghese (R1918486) Miss. Meghna MK(R1918459)** has been conducted under my direct supervision and guidance of **Prof. Shohinee Mandel.**

This study is in partial fulfillment for the award of “**Bachelor of Computer Application**” from Bangalore University. Further it is certified that this project has not been previously submitted to any other university for the requirement of Bachelor of Computer Application.

Place:Bangalore

Date:

Signature of Guide



UNDERTAKING

We, **Mr. Allen Albino (R1918412)**, **Mr. Prince M Varghese (R1918486)** **Miss. Meghna MK(R1918459)** studying in 5th Semester BCA ensure that the project entitled “LEXA” submitted to the Department of Computer Application has been carried out by us, as per the requirement of Bangalore University. This Report has not formed the basis for the award of the Degree of the Bangalore University or any other University.

Signature of the Student

Mr. Allen Albino (R1918417)

Mr. Prince M Varghese (R1918486)

Miss. Meghna M.K (R1918459)

Place: Bangalore

Date:



ACKNOWLEDGEMENT

We take this opportunity to express my deep sense of gratitude to the Director **Dr. Prakash B Nayak**, Academic Director **Dr. Sara Kunnath** and Vice Principal **Prof. Sandhya S** for the valuable guidance, keen interest and helpful criticism, during the course of studies.

We express sincere thanks to our respected HOD **Prof. Uma Mageshwari** for providing all necessary helps during our project and enabling us to draft this project and the successful implementation of the project.

Also, we express sincere thanks to our respected Guide **Prof. Shohinee Mandel**, and all our faculties for providing all necessary helps during our project.

Last but not the least We would like to thank our Friends, the Teaching and Nonteaching Staffs and all other who have directly and indirectly helped me in successful completion of the project.

Place: Bangalore

Date:

CONTENTS

1. ABSTRACT	1
2. INTRODUCTION	2
2.1 Admin Module	
2.2 User Module	
2.3 Purpose	
2.4 Scope	
3. SOFTWARE REQUIREMENT & SPECIFICATION....	4
3.1 Hardware Configuration	
3.2 Software Configuration	
4. SYSTEM DESIGN & ANALYSIS.....	7
4.1 Analysis	
4.2 Design Introduction	
4.3 Use case Diagram	
4.3.1 Admin	
4.3.2 User	
4.4 Block Diagram	
4.5 E R Diagram	
5. CODING	18

6.	TESTING & IMPLEMENTATION.....	50
6.1	Need For Testin	
6.2	Black Box Testing	
6.3	White Box Testing	
6.4	Unit Testing	
6.5	Integration Testing	
6.6	System Testing	
6.6	Validation Testing	
6.7	Integration Testing	
6.8	Output Testing	
7.	SCREENSHOTS.....	55
8.	CONCLUSION.....	64
9.	BIBLIOGRAPHY.....	65

ABSTRACT

Online Shopping system is a website primarily for use in the E-commerce industry.

This system will allow shops to increase the scope of business by reducing labor cost involved.

The system also allows to quickly and easily manage an online items which customers can browse and use to place order with just few clicks. Textile employees then use these order through and easy to navigate graphical interface for efficient processing

CHAPTER 1

INTRODUCTION

1.1 Overview of project

Online Shopping system is a website designed primarily for use in the E-commerce industry.

This system will allow Shops to increase scope of business by reducing labor cost involved.

The system also allows to quickly and easily manage an online items which customers can browse and use to place order with just few clicks. Textile employees then use these order through and easy to navigate graphical interface for efficient processing.

Online Shopping system that we are proposing here, greatly simplifies the ordering process for both the customers and the textiles.

System presence an interactive and up -to-date menu with all available options in an easy to use manner.

Customer can choose one or more items to place an order which will land in the cart.

Customer can be view all the order details in the cart before checking out. At the end, customer gets order confirmation details.

Once the order is placed it is entered in the database and retrieved in pretty much real time.

This allow textile employees to quickly go through the orders as they are received and process all the orders efficiently and effectively with minimum delays and confusion .

1.2 Modules Description

The major objective of this project is to develop an application that is designed for the Online Food Ordering Management System and provide user friendly environment for the user. All the options needed for the user is available in this application.

It has been modularized into following modules

- Administrator Module
- User Module
- Manager Module

➤ **Manager Module:**

- View dress items: allows to check their menu
- Add dress items: allows dress items to add to their menus
- Edit dress items: allow to edit and their dress items (price, description, image, name etc).
- Delete dress items: allows to delete dress items from their menu

➤ **User Module:**

User can have a preview about the website.

User can select their desired dress from the menu and add to their cart. Allows users to see details of the items placed in cart.

Details include item name, item image, item description, quality, unit price, total per items and final total of the order.

It allows update and delete an item using a single button click. User can then use a 'proceed to checkout' button to proceed further.

Once, checkout button is selected, user will be presented with a “Review Order” page, which will display payment information along with order details review User can then use a 'check out' button to place order.

Once order is placed user will be presented with an appropriate order confirmation success or failure message.

CHAPTER 2

SOFTWARE REQUIRMENT SPECIFICATION

2.1 Hardware Requirements

Processor: Intel Core Duo 2.0 GHz or higher.

RAM:1 GB and above.

Hard Disk: 60 GB and above.

Monitor: Plug and play Monitor.

Mouse: Microsoft PS/2 mouse

Keyboard: Standard 101/102 key.

2.1.1 About the Program Language

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics developed by Guido van Rossum. It was originally released in 1991. Designed to be easy as well as fun, the name "Python" is a nod to the British comedy group Monty Python. Python has a reputation as a beginner-friendly language, replacing Java as the most widely used introductory language because it handles much of the complexity for the user, allowing beginners to focus on fully grasping programming concepts rather than minute details.

Python is used for server-side web development, software development, mathematics, and system scripting, and is popular for Rapid Application Development and as a scripting or glue language to tie existing components because of its high-level, built-in data structures, dynamic typing, and dynamic binding. Program maintenance costs are reduced with Python due to the easily learned syntax and emphasis on readability. Additionally, Python's support of modules and packages facilitates modular programs and reuse of code. Python is an open source community language, so numerous independent programmers are continually building libraries and functionality for it.

2.1.2 About Django

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

2.2 Software Requirements

System requirements are expressed in a software requirement document. The software requirement document (also known as software requirement specification or SRS) is the official statement of what is required of the system developers.

- Software : Django
- Operating System : Windows 7 or higher
- Browser : Google Chrome, Edge,etc.
- Front End : Python
- Back End : SQLite
- Server : Django

SQLite- SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

- **Django** - Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support. Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product", it all works seamlessly together, follows consistent design principles, and has extensive and up-to-date documentation. Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is built with Django! Internally, while it provides choices for almost any functionality you might want (e.g. several popular databases, templating engines, etc.), it can also be extended to use other components if needed.
- **HTML**- HTML, in full hypertext markup language, a formatting system for displaying material retrieved over the Internet. Each retrieval unit is known as a Web page (from World Wide Web), and such pages frequently contain hypertext links that allow related pages to be retrieved. HTML is the markup language for encoding Web pages. It was designed by the British scientist Sir Tim Berners-Lee at the CERN nuclear physics laboratory in Switzerland during the 1980s. HTML markup tags specify document elements such as headings, paragraphs, and tables. They mark up a document for display by a computer program known as a Web browser. The browser interprets the tags, displaying the headings, paragraphs, and tables in a layout that is adapted to the screen size and fonts available to it.
- **Python**- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.
- **Reporting Tool** i.e through Data Report

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing_system

In the present scenario people have to physically visit cloth shop for buying dress, and have to make payment through cash mode most of the times due to unawareness of advanced technologies at certain places.

In this method time as well as physical work is required, among which time is something that no one has in ample amount. The traditional online shopping ordering procedure is not efficient enough for boutique and textiles, as they have to deal with crowd, in their shop. The old methods can be classified into categories which are paper grounded and verbal grounded. For paper based work, staff have to show every dress type and colours. Also from the owner's point of view maintaining data record and the accounts in physical file is cumbersome and tedious work to do. And also it is full of risk as anyone can access it and modify the data.

3.2 Proposed_system

This system is a bunch of benefits from various point of views. As this online application enables the end users to register to the system online, select the dress items of their choice from the menu list, and shop online.

Also the payment can be made through online mode or at the time of home delivery depending upon the customer's choice and convenience. The selection made by the customers will be available to the retail reception or to the person handling work assignment. Now this same person will assign the orders to the workers to be completed within a fixed duration of time.

As soon as the dress had packed, the later person forwards the parcels to the delivery persons assigned with the location and customer identity of the customer along with the bill status.

With this application the work load of the staff in the textiles are reduced or in some situations the work is abolished.

One of the various benefits of this is system is that if there is rush or a huge crowd present in the shop then in that case sometimes unavailability of space to roam in the shop.

Also there will be chances that the staff are unavailable as they are busy in handling others, so the customer can directly purchase dress online by using this website, by checking the product in the product available in the shop. This system allow the staff to serve customer within less time as compared to the manual system.

3.3 Applications

This Online Shopping Management System Is In Python,HTML and CSS. Talking about the features of the system, It contains the Manager and user sections. From here, the manager can view, add and edit the dress Items. The design of this project is pretty simple so that the user won't find any difficulties while working on it

CHAPTER 4

SYSTEM DESIGN& FEASIBILITY STUDY

4.1Database Design

The main motive of database is to help manage the complete data records in a proper manner. A good database must create tables such that their individual relevance can be understood.

The SQLite is responsible for enforcing the database structure, including:

Manager Table:-

Column Name	Data Type	Description
Manager_id	Int	Primary Key
Username	Varchar	username for login
Password	Varchar	Password for login
Date	Datetime	Datetime of account created

User Table:-

Column Name	Data Type	Description
Full Name	Varchar	Name
Username	Varchar	Login Name
Address	Varchar	Address of User
Email	Varchar	Email of User
Contact	Int	Phone no of User
Password	Varchar	Password of User
Date	Datetime	Date created

4.1.1 Maintaining relationship between data in the database

Ensuring that data stored correctly and that the rules defining data relationships are not violated. Recover all data a point of known consistency in case of system failures.

4.1.2 Relational database

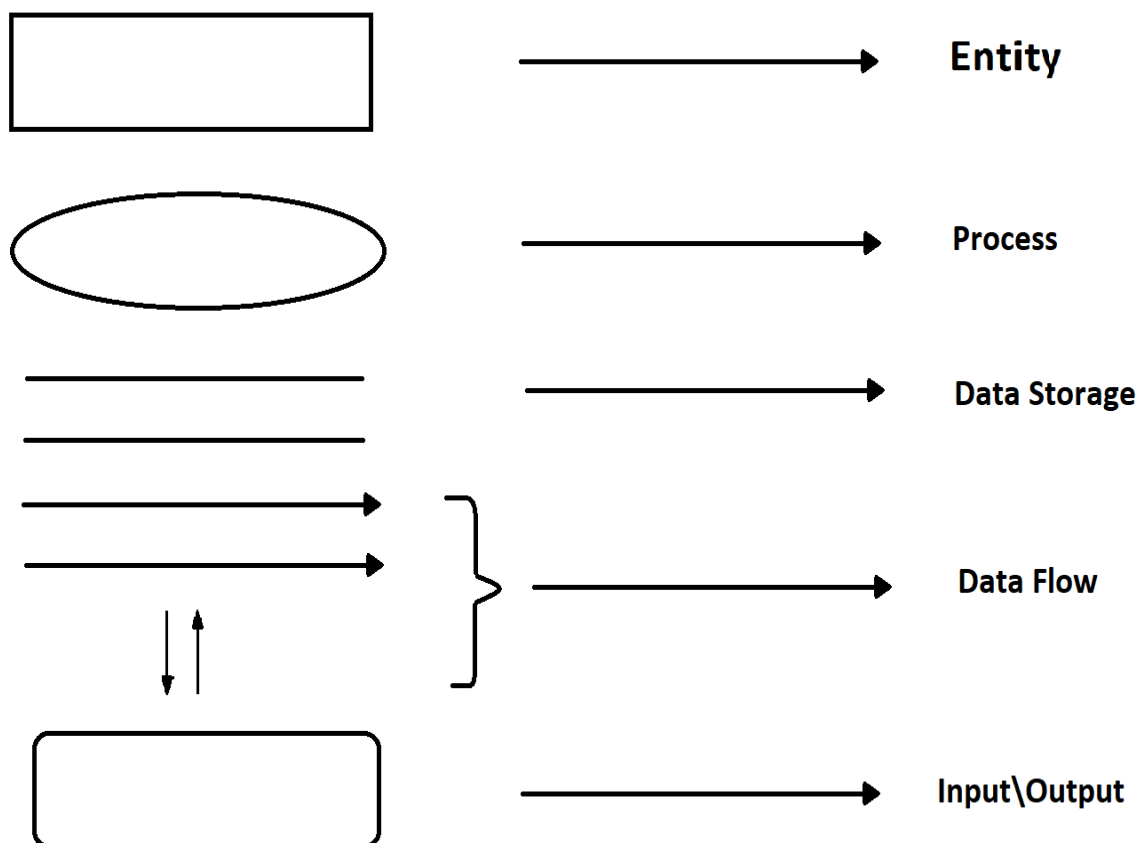
Although there are different ways to organize data in a database, relational database are one of the most effective. Relational database are an application of mathematical set theory to the problem of effectively organizing data, in a relational database; data is collected into tables (called relations in relation theory). At table represents some class of objects that are important to an organization.

4.2 Data Flow Diagram

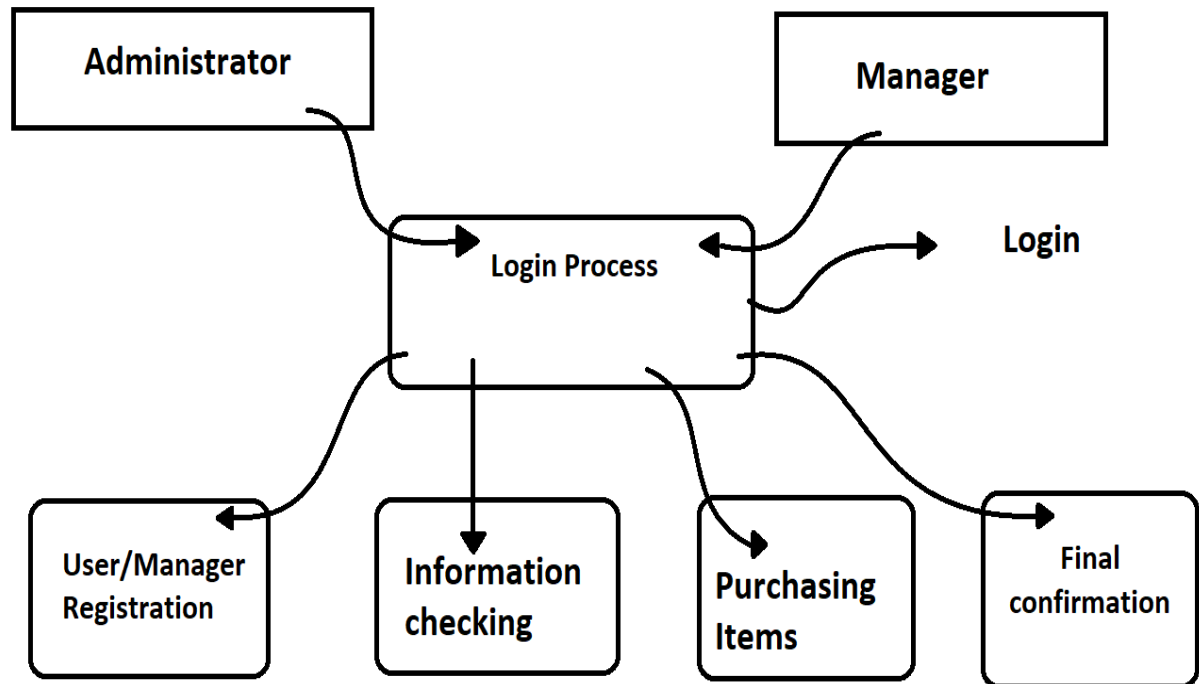
4.2.1 Introduction

Goal of SDLC is to deliver system in line with the users requirement analysis is the Data flow diagram illustrates how data flows through a system and how output is derived from the input through a sequence a functional transformations,

Data Flow Diagrams are useful and innovative way of describing n system. They are normally understandable without special training, especially if control information is executed. They show end to end processing that is, the flow of processing from when data enters the system to when it leaves the system can be traced. Data Flow design is an integral part of a number of design methods and more CASE tools supports data flow design creation, it could maintain implanted as a number of communcing task.



DATA FLOW DIAGRAM



4.3 ER - Diagram

An entity - relationship (ER) diagram is a specialized graphic that illustrates the interrelationship between in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationship diagrams don't show single entities or single instance of relations. Rather, they show entity sets and relationship sets.

Lines are drawn between entity sets and the relationship sets they are involved in. If all entities in an entity set must participate in the relationship set, a thick or double line is drawn. This is called participation constraints. If each entity set can participate in at most one relationship set, an arrow is drawn from the entity set to the relationship set.

This is called key constraints. To indicate that each entity in the entity set is involved in one relationship, a thick arrow is drawn.

Associative entity is used to solve the problem of the two entities with a many-to-many relationship.

Definition of Symbols:

Attribute



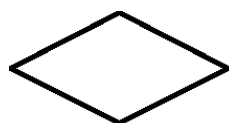
Key Attribute



Entity

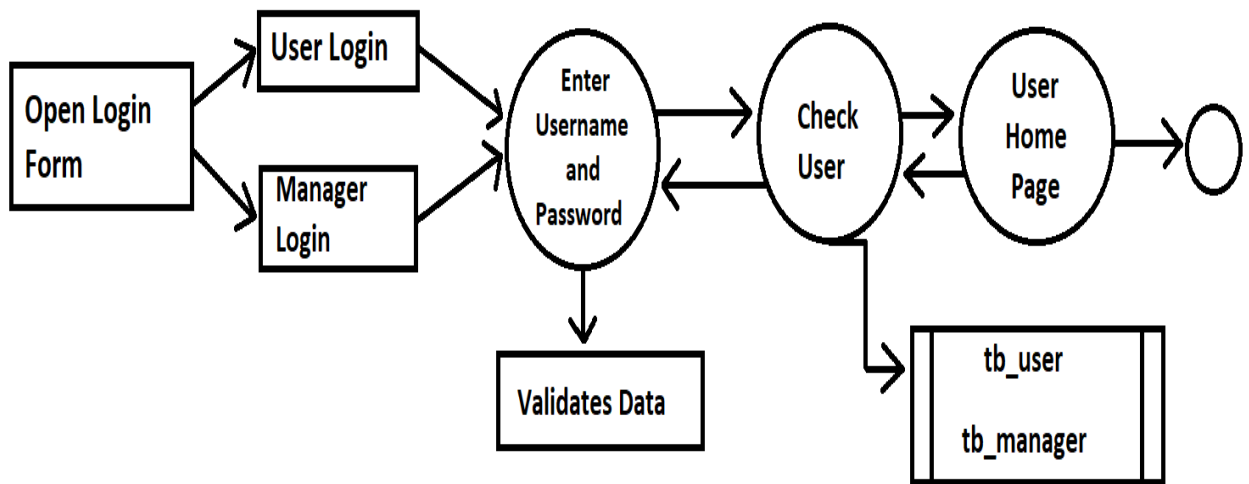


Entity relations

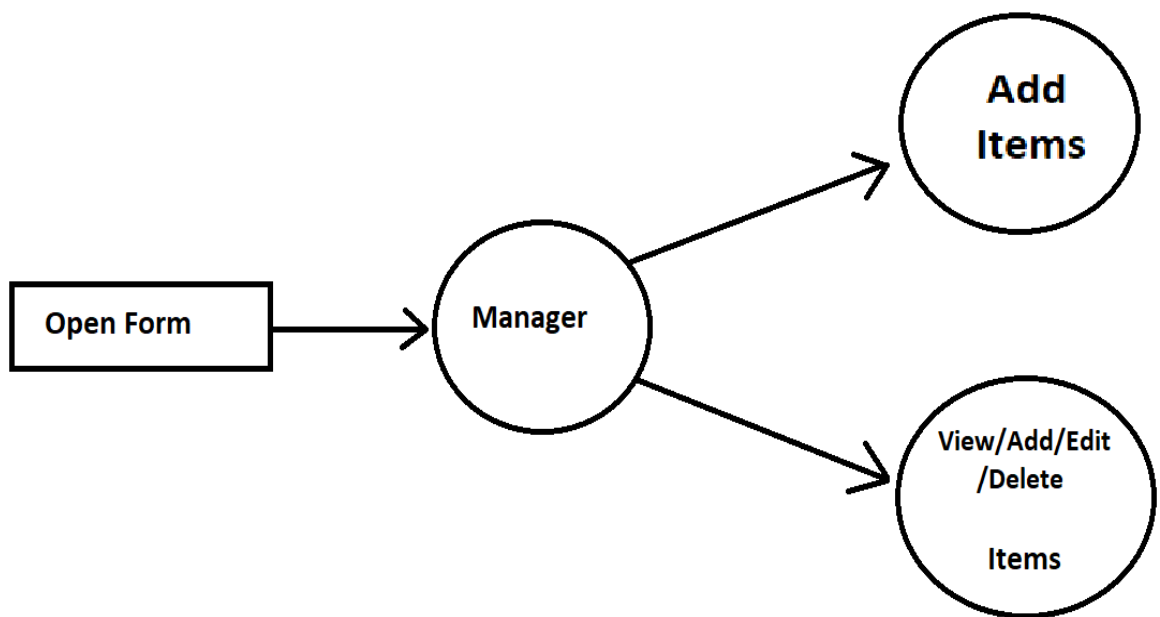


ER-DIAGRAM

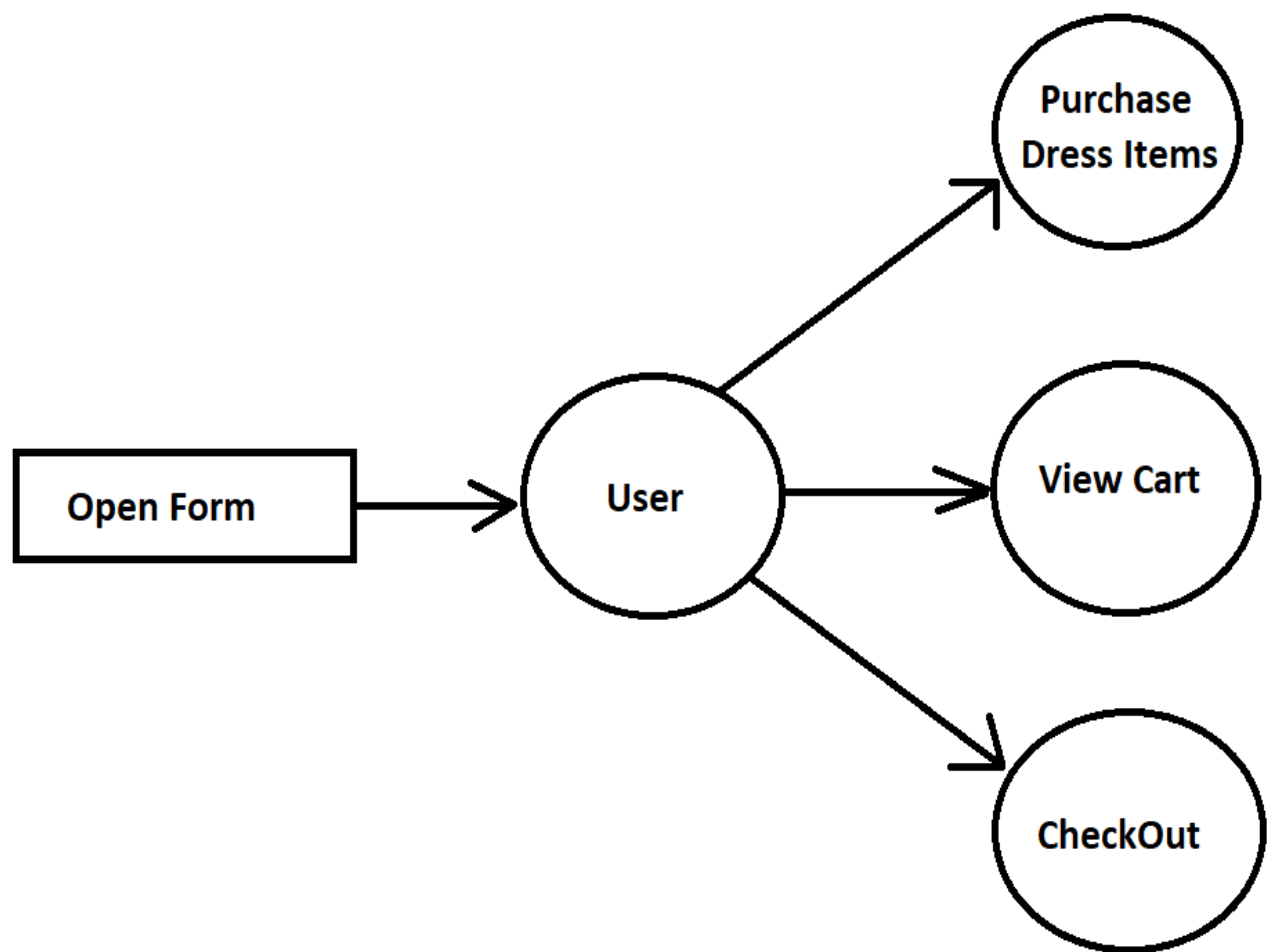
LEVEL 0



LEVEL 1:



LEVEL2:



4.4 Input Design

Input design is the method by which valid data are accepted from the user. The valid data turn is stored as operational data in the database. Incorrect input data are the most common cause of errors in the data processing. The input design is carried out in such a way that the input screens are user friendly. The goals of designing input design are to make input data entry as easy and error free. Input screen takes care to filter the valid data from being becoming an operational data at data entry phase.

Input design is the part of the overall system design that careful attention and is the most expensive phase. It is the point of most contact for the users with the system and so it is prone to errors.

4.5 Output Design

The output design defines the output required and the format in which it is to be produced. Care must be given to present the right information.

The output is the most important and direct source of information to the user. Efficient, output design should improve the systems relationship with the user and helps in decision making. A major form of output is a hard copy from the printer. Printouts should be designed around the output requirement to the user. The standard that is maintained for output design is clear. Output provides a permanent copy of the results for the later consultation.

4.6 Feasibility Study

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution. The analyst conducting the study gathers information using a variety of methods, the most popular of which are:

- Interviewing users, managers.
- Developing and administering questionnaires to interested stakeholders, such as potential users of the information system.
- Observing or monitoring users of the current system to determine their needs as well as their satisfaction and dissatisfaction with the current system.
- Collecting, examining, and analyzing documents, reports, layouts, procedures, manuals, and any other documentation relating to the operations of the current system.
- Modeling, observing, and simulating the work activities of the current system

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components. These components are:

CHAPTER 5

CODING

5.1 Webpage design

```
from django.shortcuts import render,redirect,reverse
from . import forms,models
from django.http import HttpResponseRedirect,HttpResponse
from django.core.mail import send_mail
from django.contrib.auth.models import Group
from django.contrib.auth.decorators import login_required,user_passes_test
from django.contrib import messages
from django.conf import settings

def home_view(request):
    products=models.Product.objects.all()
    if 'product_ids' in request.COOKIES:
        product_ids = request.COOKIES['product_ids']
        counter=product_ids.split('|')
        product_count_in_cart=len(set(counter))
    else:
        product_count_in_cart=0
    if request.user.is_authenticated:
        return HttpResponseRedirect('afterlogin')
    return
render(request,'ecom/index.html',{ 'products':products,'product_count_in_cart':product_count_in_cart})

#for showing login button for admin(by sumit)
def adminclick_view(request):
    if request.user.is_authenticated:
        return HttpResponseRedirect('afterlogin')
    return HttpResponseRedirect('adminlogin')

def customer_signup_view(request):
    userForm=forms.CustomerUserForm()
    customerForm=forms.CustomerForm()
    mydict={'userForm':userForm,'customerForm':customerForm}
    if request.method=='POST':
        userForm=forms.CustomerUserForm(request.POST)
        customerForm=forms.CustomerForm(request.POST,request.FILES)
        if userForm.is_valid() and customerForm.is_valid():
            user=userForm.save()
            user.set_password(user.password)
            user.save()
            customer=customerForm.save(commit=False)
            customer.user=user
            customer.save()
            my_customer_group = Group.objects.get_or_create(name='CUSTOMER')
```

```
        my_customer_group[0].user_set.add(user)
    return HttpResponseRedirect('customerlogin')
return render(request,'ecom/customersignup.html',context=mydict)
```

```
#-----for checking user iscustomer
def is_customer(user):
    return user.groups.filter(name='CUSTOMER').exists()
```

```
#-----AFTER ENTERING CREDENTIALS WE CHECK WHETHER USERNAME
AND PASSWORD IS OF ADMIN,CUSTOMER
```

```
def afterlogin_view(request):
    if is_customer(request.user):
        return redirect('customer-home')
    else:
        return redirect('admin-dashboard')
```

```
#-----
#----- ADMIN RELATED VIEWS START -----
#-----
```

```
@login_required(login_url='adminlogin')
def admin_dashboard_view(request):
    # for cards on dashboard
    customercount=models.Customer.objects.all().count()
    productcount=models.Product.objects.all().count()
    ordercount=models.Orders.objects.all().count()
```

```
    # for recent order tables
    orders=models.Orders.objects.all()
    ordered_products=[]
    ordered_bys=[]
    for order in orders:
        ordered_product=models.Product.objects.all().filter(id=order.product.id)
        ordered_by=models.Customer.objects.all().filter(id = order.customer.id)
        ordered_products.append(ordered_product)
        ordered_bys.append(ordered_by)
```

```
    mydict={
        'customercount':customercount,
        'productcount':productcount,
        'ordercount':ordercount,
        'data':zip(ordered_products,ordered_bys,orders),
    }
    return render(request,'ecom/admin_dashboard.html',context=mydict)
```

```
# admin view customer table
@login_required(login_url='adminlogin')
def view_customer_view(request):
```

```
customers=models.Customer.objects.all()
return render(request,'ecom/view_customer.html',{'customers':customers})
```

```
# admin delete customer
@login_required(login_url='adminlogin')
def delete_customer_view(request,pk):
    customer=models.Customer.objects.get(id=pk)
    user=models.User.objects.get(id=customer.user_id)
    user.delete()
    customer.delete()
    return redirect('view-customer')

@login_required(login_url='adminlogin')
def update_customer_view(request,pk):
    customer=models.Customer.objects.get(id=pk)
    user=models.User.objects.get(id=customer.user_id)
    userForm=forms.CustomerUserForm(instance=user)
    customerForm=forms.CustomerForm(request.FILES,instance=customer)
    mydict={'userForm':userForm,'customerForm':customerForm}
    if request.method=='POST':
        userForm=forms.CustomerUserForm(request.POST,instance=user)
        customerForm=forms.CustomerForm(request.POST,instance=customer)
        if userForm.is_valid() and customerForm.is_valid():
            user=userForm.save()
            user.set_password(user.password)
            user.save()
            customerForm.save()
            return redirect('view-customer')
    return render(request,'ecom/admin_update_customer.html',context=mydict)
```

```
# admin view the product
@login_required(login_url='adminlogin')
def admin_products_view(request):
    products=models.Product.objects.all()
    return render(request,'ecom/admin_products.html',{'products':products})
```

```
# admin add product by clicking on floating button
@login_required(login_url='adminlogin')
def admin_add_product_view(request):
    productForm=forms.ProductForm()
    if request.method=='POST':
        productForm=forms.ProductForm(request.POST, request.FILES)
        if productForm.is_valid():
            productForm.save()
            return HttpResponseRedirect('admin-products')
    return render(request,'ecom/admin_add_products.html',{'productForm':productForm})
```

```
@login_required(login_url='adminlogin')
def delete_product_view(request,pk):
```

```
product=models.Product.objects.get(id=pk)
product.delete()
return redirect('admin-products')
```

```
@login_required(login_url='adminlogin')
def update_product_view(request,pk):
    product=models.Product.objects.get(id=pk)
    productForm=forms.ProductForm(instance=product)
    if request.method=='POST':
        productForm=forms.ProductForm(request.POST,request.FILES,instance=product)
        if productForm.is_valid():
            productForm.save()
            return redirect('admin-products')
    return render(request,'ecom/admin_update_product.html',{'productForm':productForm})
```

```
@login_required(login_url='adminlogin')
def admin_view_booking_view(request):
    orders=models.Orders.objects.all()
    ordered_products=[]
    ordered_bys=[]
    for order in orders:
        ordered_product=models.Product.objects.all().filter(id=order.product.id)
        ordered_by=models.Customer.objects.all().filter(id = order.customer.id)
        ordered_products.append(ordered_product)
        ordered_bys.append(ordered_by)
    return
    render(request,'ecom/admin_view_booking.html',{'data':zip(ordered_products,ordered_bys,orders)})
```

```
@login_required(login_url='adminlogin')
def delete_order_view(request,pk):
    order=models.Orders.objects.get(id=pk)
    order.delete()
    return redirect('admin-view-booking')
```

```
# for changing status of order (pending,delivered...)
@login_required(login_url='adminlogin')
def update_order_view(request,pk):
    order=models.Orders.objects.get(id=pk)
    orderForm=forms.OrderForm(instance=order)
    if request.method=='POST':
        orderForm=forms.OrderForm(request.POST,instance=order)
        if orderForm.is_valid():
            orderForm.save()
            return redirect('admin-view-booking')
    return render(request,'ecom/update_order.html',{'orderForm':orderForm})
```

```
# admin view the feedback
```

```

@login_required(login_url='adminlogin')
def view_feedback_view(request):
    feedbacks=models.Feedback.objects.all().order_by('-id')
    return render(request,'ecom/view_feedback.html',{'feedbacks':feedbacks})

#-----
#----- PUBLIC CUSTOMER RELATED VIEWS START -----
#-----
def search_view(request):
    # whatever user write in search box we get in query
    query = request.GET['query']
    products=models.Product.objects.all().filter(name__icontains=query)
    if 'product_ids' in request.COOKIES:
        product_ids = request.COOKIES['product_ids']
        counter=product_ids.split('|')
        product_count_in_cart=len(set(counter))
    else:
        product_count_in_cart=0

    # word variable will be shown in html when user click on search button
    word="Searched Result :"

    if request.user.is_authenticated:
        return
    render(request,'ecom/customer_home.html',{'products':products,'word':word,'product_count_in_cart':product_count_in_cart})
    return
    render(request,'ecom/index.html',{'products':products,'word':word,'product_count_in_cart':product_count_in_cart})

# any one can add product to cart, no need of signin
def add_to_cart_view(request,pk):
    products=models.Product.objects.all()

    #for cart counter, fetching products ids added by customer from cookies
    if 'product_ids' in request.COOKIES:
        product_ids = request.COOKIES['product_ids']
        counter=product_ids.split('|')
        product_count_in_cart=len(set(counter))
    else:
        product_count_in_cart=1

    response = render(request,
'ecom/index.html',{'products':products,'product_count_in_cart':product_count_in_cart})

    #adding product id to cookies
    if 'product_ids' in request.COOKIES:

```

```

    product_ids = request.COOKIES['product_ids']
    if product_ids=="":
        product_ids=str(pk)
    else:
        product_ids=product_ids+"|"+str(pk)
    response.set_cookie('product_ids', product_ids)
else:
    response.set_cookie('product_ids', pk)

product=models.Product.objects.get(id=pk)
messages.info(request, product.name + ' added to cart successfully!')

return response

# for checkout of cart
def cart_view(request):
    #for cart counter
    if 'product_ids' in request.COOKIES:
        product_ids = request.COOKIES['product_ids']
        counter=product_ids.split('|')
        product_count_in_cart=len(set(counter))
    else:
        product_count_in_cart=0

    # fetching product details from db whose id is present in cookie
    products=None
    total=0
    if 'product_ids' in request.COOKIES:
        product_ids = request.COOKIES['product_ids']
        if product_ids != "":
            product_id_in_cart=product_ids.split('|')
            products=models.Product.objects.all().filter(id__in = product_id_in_cart)

            #for total price shown in cart
            for p in products:
                total=total+p.price
    return
render(request,'ecom/cart.html',{ 'products':products,'total':total,'product_count_in_cart':product_count_in_cart})

def remove_from_cart_view(request,pk):
    #for counter in cart
    if 'product_ids' in request.COOKIES:
        product_ids = request.COOKIES['product_ids']
        counter=product_ids.split('|')
        product_count_in_cart=len(set(counter))
    else:
        product_count_in_cart=0

```

```

# removing product id from cookie
total=0
if 'product_ids' in request.COOKIES:
    product_ids = request.COOKIES['product_ids']
    product_id_in_cart=product_ids.split('|')
    product_id_in_cart=list(set(product_id_in_cart))
    product_id_in_cart.remove(str(pk))
    products=models.Product.objects.all().filter(id__in = product_id_in_cart)
    #for total price shown in cart after removing product
    for p in products:
        total=total+p.price

# for update coookie value after removing product id in cart
value=""
for i in range(len(product_id_in_cart)):
    if i==0:
        value=value+product_id_in_cart[0]
    else:
        value=value+"|"+product_id_in_cart[i]
response = render(request,
'ecom/cart.html',{'products':products,'total':total,'product_count_in_cart':product_count_in_cart})
if value=="":
    response.delete_cookie('product_ids')
    response.set_cookie('product_ids',value)
    return response

def send_feedback_view(request):
    feedbackForm=forms.FeedbackForm()
    if request.method == 'POST':
        feedbackForm = forms.FeedbackForm(request.POST)
        if feedbackForm.is_valid():
            feedbackForm.save()
            return render(request, 'ecom/feedback_sent.html')
    return render(request, 'ecom/send_feedback.html', {'feedbackForm':feedbackForm})

#-----
#----- CUSTOMER RELATED VIEWS START -----
#-----
@login_required(login_url='customerlogin')
@user_passes_test(is_customer)
def customer_home_view(request):
    products=models.Product.objects.all()
    if 'product_ids' in request.COOKIES:
        product_ids = request.COOKIES['product_ids']
        counter=product_ids.split('|')
        product_count_in_cart=len(set(counter))
    else:
        product_count_in_cart=0

```

```

    return
render(request,'ecom/customer_home.html',{ 'products':products,'product_count_in_cart':product_count_in_cart})

# shipment address before placing order
@login_required(login_url='customerlogin')
def customer_address_view(request):
    # this is for checking whether product is present in cart or not
    # if there is no product in cart we will not show address form
    product_in_cart=False
    if 'product_ids' in request.COOKIES:
        product_ids = request.COOKIES['product_ids']
        if product_ids != "":
            product_in_cart=True
    #for counter in cart
    if 'product_ids' in request.COOKIES:
        product_ids = request.COOKIES['product_ids']
        counter=product_ids.split('|')
        product_count_in_cart=len(set(counter))
    else:
        product_count_in_cart=0

    addressForm = forms.AddressForm()
    if request.method == 'POST':
        addressForm = forms.AddressForm(request.POST)
        if addressForm.is_valid():
            # here we are taking address, email, mobile at time of order placement
            # we are not taking it from customer account table because
            # these thing can be changes
            email = addressForm.cleaned_data['Email']
            mobile=addressForm.cleaned_data['Mobile']
            address = addressForm.cleaned_data['Address']
            #for showing total price on payment page.....accessing id from cookies then fetching
            price of product from db
            total=0
            if 'product_ids' in request.COOKIES:
                product_ids = request.COOKIES['product_ids']
                if product_ids != "":
                    product_id_in_cart=product_ids.split('|')
                    products=models.Product.objects.all().filter(id__in = product_id_in_cart)
                    for p in products:
                        total=total+p.price

            response = render(request, 'ecom/payment.html',{ 'total':total})
            response.set_cookie('email',email)
            response.set_cookie('mobile',mobile)
            response.set_cookie('address',address)
            return response
    return
render(request,'ecom/customer_address.html',{ 'addressForm':addressForm,'product_in_cart':product_in_cart,'product_count_in_cart':product_count_in_cart})

```



```

# here we are just directing to this view...actually we have to check whether payment is
successful or not
#then only this view should be accessed
@login_required(login_url='customerlogin')
def payment_success_view(request):
    # Here we will place order | after successful payment
    # we will fetch customer mobile, address, Email
    # we will fetch product id from cookies then respective details from db
    # then we will create order objects and store in db
    # after that we will delete cookies because after order placed...cart should be empty
    customer=models.Customer.objects.get(user_id=request.user.id)
    products=None
    email=None
    mobile=None
    address=None
    if 'product_ids' in request.COOKIES:
        product_ids = request.COOKIES['product_ids']
        if product_ids != "":
            product_id_in_cart=product_ids.split('|')
            products=models.Product.objects.all().filter(id__in = product_id_in_cart)
            # Here we get products list that will be ordered by one customer at a time

    # these things can be change so accessing at the time of order...
    if 'email' in request.COOKIES:
        email=request.COOKIES['email']
    if 'mobile' in request.COOKIES:
        mobile=request.COOKIES['mobile']
    if 'address' in request.COOKIES:
        address=request.COOKIES['address']

    # here we are placing number of orders as much there is a products
    # suppose if we have 5 items in cart and we place order....so 5 rows will be created in
orders table
    # there will be lot of redundant data in orders table...but its become more complicated if we
normalize it
    for product in products:

models.Orders.objects.get_or_create(customer=customer,product=product,status='Pending',e
mail=email,mobile=mobile,address=address)

    # after order placed cookies should be deleted
    response = render(request,'ecom/payment_success.html')
    response.delete_cookie('product_ids')
    response.delete_cookie('email')
    response.delete_cookie('mobile')
    response.delete_cookie('address')
    return response

```

```

@login_required(login_url='customerlogin')
@user_passes_test(is_customer)
def my_order_view(request):
    customer=models.Customer.objects.get(user_id=request.user.id)
    orders=models.Orders.objects.all().filter(customer_id = customer)
    ordered_products=[]
    for order in orders:
        ordered_product=models.Product.objects.all().filter(id=order.product.id)
        ordered_products.append(ordered_product)

    return render(request,'ecom/my_order.html',{'data':zip(ordered_products,orders)})

```

```

#-----for discharge patient bill (pdf) download and printing
import io
# from xhtml2pdf import pisa
from django.template.loader import get_template
from django.template import Context
from django.http import HttpResponse

```

```

def render_to_pdf(template_src, context_dict):
    template = get_template(template_src)
    html = template.render(context_dict)
    result = io.BytesIO()
    pdf = pisa.pisaDocument(io.BytesIO(html.encode("ISO-8859-1")), result)
    if not pdf.err:
        return HttpResponse(result.getvalue(), content_type='application/pdf')
    return

```

```

@login_required(login_url='customerlogin')
@user_passes_test(is_customer)
def download_invoice_view(request,orderID,productID):
    order=models.Orders.objects.get(id=orderID)
    product=models.Product.objects.get(id=productID)
    mydict={
        'orderDate':order.order_date,
        'customerName':request.user,
        'customerEmail':order.email,
        'customerMobile':order.mobile,
        'shipmentAddress':order.address,
        'orderStatus':order.status,

        'productName':product.name,
        'productImage':product.product_image,
        'productPrice':product.price,
        'productDescription':product.description,

    }

```

```
# return render_to_pdf('ecom/download_invoice.html',mydict)
return mydict
```

```
@login_required(login_url='customerlogin')
@user_passes_test(is_customer)
def my_profile_view(request):
    customer=models.Customer.objects.get(user_id=request.user.id)
    return render(request,'ecom/my_profile.html',{'customer':customer})
```

```
@login_required(login_url='customerlogin')
@user_passes_test(is_customer)
def edit_profile_view(request):
    customer=models.Customer.objects.get(user_id=request.user.id)
    user=models.User.objects.get(id=customer.user_id)
    userForm=forms.CustomerUserForm(instance=user)
    customerForm=forms.CustomerForm(request.FILES,instance=customer)
    mydict={'userForm':userForm,'customerForm':customerForm}
    if request.method=='POST':
        userForm=forms.CustomerUserForm(request.POST,instance=user)
        customerForm=forms.CustomerForm(request.POST,instance=customer)
        if userForm.is_valid() and customerForm.is_valid():
            user=userForm.save()
            user.set_password(user.password)
            user.save()
            customerForm.save()
            return HttpResponseRedirect('my-profile')
    return render(request,'ecom/edit_profile.html',context=mydict)
```

```
#-----
#----- ABOUT US AND CONTACT US VIEWS START -----
#-----
```

```
def aboutus_view(request):
    return render(request,'ecom/aboutus.html')
```

```
def contactus_view(request):
    sub = forms.ContactusForm()
    if request.method == 'POST':
        sub = forms.ContactusForm(request.POST)
        if sub.is_valid():
            email = sub.cleaned_data['Email']
            name=sub.cleaned_data['Name']
            message = sub.cleaned_data['Message']
            send_mail(str(name)+' || '+str(email),message, settings.EMAIL_HOST_USER,
settings.EMAIL_RECEIVING_USER, fail_silently = False)
            return render(request, 'ecom/contactussuccess.html')
        return render(request, 'ecom/contactus.html', {'form':sub})
```

Setting.py

Django settings for ecommerce project.

Generated by 'django-admin startproject' using Django 3.0.5.

For more information on this file, see
<https://docs.djangoproject.com/en/3.0/topics/settings/>

For the full list of settings and their values, see
<https://docs.djangoproject.com/en/3.0/ref/settings/>
"""

```
import os
```

```
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
TEMPLATE_DIR = os.path.join(BASE_DIR, 'templates')
STATIC_DIR = os.path.join(BASE_DIR, 'static')
```

```
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/
```

```
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '#vw(03o=(9kbvg!&2d5i!2$_58x@_-3l4wujpow6(ym37jxnza'
```

```
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True
```

```
ALLOWED_HOSTS = []
```

```
# Application definition
```

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'ecom',
    'widget_tweaks',
```

```
]
```

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
  
]
```

```
ROOT_URLCONF = 'ecommerce.urls'
```

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [TEMPLATE_DIR,],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    ],  
]
```

```
WSGI_APPLICATION = 'ecommerce.wsgi.application'
```

```
# Database  
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
    }  
}
```

```
# Password validation  
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators
```

```
AUTH_PASSWORD_VALIDATORS = [
```

```
{
    'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
},
]
```

```
# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/
```

```
LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'
```

```
USE_I18N = True
```

```
USE_L10N = True
```

```
USE_TZ = True
```

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.0/howto/static-files/
```

```
STATIC_URL = '/static/'
```

```
STATICFILES_DIRS=[STATIC_DIR,]
```

```
MEDIA_ROOT=os.path.join(BASE_DIR,'static')
```

```
LOGIN_REDIRECT_URL='/afterlogin'
```

```
#for contact us give your gmail id and password
EMAIL_BACKEND='django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST='smtp.gmail.com'
```

```
EMAIL_USE_TLS = True
EMAIL_PORT = 587
EMAIL_HOST_USER = 'from@gmail.com' # this email will be used to send emails
EMAIL_HOST_PASSWORD = 'xyz' # host email password required
# now sign in with your host gmail account in your browser
# open following link and turn it ON
# https://myaccount.google.com/lesssecureapps
# otherwise you will get SMTPAuthenticationError at /contactus
# this process is required because google blocks apps authentication by default
EMAIL_RECEIVING_USER = ['to@gmail.com'] # email on which you will receive messages sent
from website
```

url.py

```
"""
```

```
from django.contrib import admin
from django.urls import path
from ecom import views
from django.contrib.auth.views import LoginView, LogoutView
from django.views.generic.base import RedirectView
from django.contrib.staticfiles.storage import staticfiles_storage
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", views.home_view, name=""),
    path('afterlogin', views.afterlogin_view, name='afterlogin'),
    path('logout', LogoutView.as_view(template_name='ecom/logout.html'), name='logout'),
    path('aboutus', views.aboutus_view),
    path('contactus', views.contactus_view, name='contactus'),
    path('search', views.search_view, name='search'),
    path('send-feedback', views.send_feedback_view, name='send-feedback'),
    path('view-feedback', views.view_feedback_view, name='view-feedback'),
```

```
    path('adminclick', views.adminclick_view),
    path('adminlogin',
LoginView.as_view(template_name='ecom/adminlogin.html'), name='adminlogin'),
    path('admin-dashboard', views.admin_dashboard_view, name='admin-dashboard'),
```

```
    path('view-customer', views.view_customer_view, name='view-customer'),
    path('delete-customer/<int:pk>', views.delete_customer_view, name='delete-customer'),
    path('update-customer/<int:pk>', views.update_customer_view, name='update-customer'),
```

```
    path('admin-products', views.admin_products_view, name='admin-products'),
    path('admin-add-product', views.admin_add_product_view, name='admin-add-product'),
    path('delete-product/<int:pk>', views.delete_product_view, name='delete-product'),
    path('update-product/<int:pk>', views.update_product_view, name='update-product'),
```

```
    path('admin-view-booking', views.admin_view_booking_view, name='admin-view-booking'),
    path('delete-order/<int:pk>', views.delete_order_view, name='delete-order'),
    path('update-order/<int:pk>', views.update_order_view, name='update-order'),
```

```

        path('customersignup', views.customer_signup_view),
        path('customerlogin',
LoginView.as_view(template_name='ecom/customerlogin.html'),name='customerlogin'),
        path('customer-home', views.customer_home_view,name='customer-home'),
        path('my-order', views.my_order_view,name='my-order'),
        path('my-profile', views.my_profile_view,name='my-profile'),
        path('edit-profile', views.edit_profile_view,name='edit-profile'),
        path('download-invoice/<int:orderID>/<int:productID>',
views.download_invoice_view,name='download-invoice'),

        path('add-to-cart/<int:pk>', views.add_to_cart_view,name='add-to-cart'),
        path('cart', views.cart_view,name='cart'),
        path('remove-from-cart/<int:pk>', views.remove_from_cart_view,name='remove-from-cart'),
        path('customer-address', views.customer_address_view,name='customer-address'),
        path('payment-success', views.payment_success_view,name='payment-success'),
        path(
            "lexa.png",
            RedirectView.as_view(url=staticfiles_storage.url("lexa.png")),
        ),

]

```

Front end index.html

```

{% extends 'ecom/homebase.html' %}
{% load static %}
{% block content %}

```

```

<style media="screen">
    body,
html {
    height: 100%;
}

```

```

.d-flex {
display: -webkit-box;
display: -ms-flexbox;
display: flex;
}

```

```

.align-center {
-webkit-box-align: center;
-ms-flex-align: center;
align-items: center;
}

```



```
.flex-centerY-centerX {
  justify-content: center;
  -webkit-box-pack: center;
  -ms-flex-pack: center;
  justify-content: center;
  -webkit-box-align: center;
  -ms-flex-align: center;
  align-items: center;
}
```

```
body {
  /* background-color:rgb(146, 146, 146); */
  background-image: linear-gradient(to bottom right, rgb(0, 0, 0) 50%, rgb(131, 131, 131));
  background-repeat: no-repeat;
}
```

```
#bgclr {
  background-image: linear-gradient(to bottom right, rgb(0, 0, 0) 50%, rgb(131, 131, 131));
}
```

```
.page-wrapper {
  height: 100%;
  display: table;
}
```

```
.page-wrapper .page-inner {
  display: table-cell;
  vertical-align: middle;
}
```

```
.el-wrapper {
  width: 360px;
  padding: 15px;
  margin: 15px auto;
  /* background-color: rgba(149, 245, 149,0.3); */
  background-color: linear-gradient(to bottom right, rgb(0, 0, 0) 50%, rgb(131, 131, 131));
}
```

```
@media (max-width: 991px) {
  .el-wrapper {
    width: 345px;
  }
}
```

```
@media (max-width: 767px) {
  .el-wrapper {
    width: 290px;
    margin: 30px auto;
  }
}
```

```
}
```

```
.el-wrapper:hover .h-bg {  
  left: 0px;  
}
```

```
.el-wrapper:hover .price {  
  left: 20px;  
  -webkit-transform: translateY(-50%);  
  -ms-transform: translateY(-50%);  
  -o-transform: translateY(-50%);  
  transform: translateY(-50%);  
  color: #818181;  
}
```

```
.el-wrapper:hover .add-to-cart {  
  left: 50%;  
}
```

```
.el-wrapper:hover .img {  
  webkit-filter: blur(7px);  
  -o-filter: blur(7px);  
  -ms-filter: blur(7px);  
  filter: blur(7px);  
  filter: progid:DXImageTransform.Microsoft.Blur(pixelradius='7', shadowopacity='0.0');  
  opacity: 0.4;  
}
```

```
.el-wrapper:hover .info-inner {  
  bottom: 155px;  
}
```

```
.el-wrapper:hover .a-size {  
  -webkit-transition-delay: 300ms;  
  -o-transition-delay: 300ms;  
  transition-delay: 300ms;  
  bottom: 50px;  
  opacity: 1;  
}
```

```
.el-wrapper .box-down {  
  width: 100%;  
  height: 60px;  
  position: relative;  
  overflow: hidden;  
}
```

```
.el-wrapper .box-up {  
  width: 100%;
```

```
height: 300px;
position: relative;
overflow: hidden;
text-align: center;
}
```

```
.el-wrapper .img {
padding: 20px 0;
-webkit-transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
-moz-transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
-o-transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
-webkit-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-moz-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-o-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
}
```

```
.h-bg {
-webkit-transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
-moz-transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
-o-transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
-webkit-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-moz-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-o-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
width: 660px;
height: 100%;
background-color: #3f96cd;
position: absolute;
left: -859px;
}
```

```
.h-bg .h-bg-inner {
width: 50%;
height: 100%;
background-color: #464646;
}
```

```
.info-inner {
-webkit-transition: all 400ms cubic-bezier(0, 0, 0.18, 1);
-moz-transition: all 400ms cubic-bezier(0, 0, 0.18, 1);
-o-transition: all 400ms cubic-bezier(0, 0, 0.18, 1);
transition: all 400ms cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
-webkit-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-moz-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-o-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
}
```

```
transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
position: absolute;
width: 100%;
bottom: 25px;
}
```

```
.info-inner .p-name,
.info-inner .p-company {
  display: block;
}
```

```
.info-inner .p-name {
  font-family: 'PT Sans', sans-serif;
  font-size: 18px;
  color: #252525;
}
```

```
.info-inner .p-company {
  font-family: 'Lato', sans-serif;
  font-size: 15px;
  text-transform: uppercase;
  color: #FFE11B;
}
```

```
.a-size {
  -webkit-transition: all 300ms cubic-bezier(0, 0, 0.18, 1);
  -moz-transition: all 300ms cubic-bezier(0, 0, 0.18, 1);
  -o-transition: all 300ms cubic-bezier(0, 0, 0.18, 1);
  transition: all 300ms cubic-bezier(0, 0, 0.18, 1);
  /* ease-out */
  -webkit-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
  -moz-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
  -o-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
  transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
  /* ease-out */
  position: absolute;
  width: 100%;
  bottom: -20px;
  font-family: 'PT Sans', sans-serif;
  color: #828282;
  opacity: 0;
}
```

```
.a-size .size {
  color: #252525;
}
```

```
.cart {
  display: block;
  position: absolute;
```

```
width: 100%;
height: 100%;
top: 0;
left: 0;
font-family: 'Lato', sans-serif;
font-weight: 700;
}
```

```
.cart .price {
  -webkit-transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
  -moz-transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
  -o-transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
  transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
  /* ease-out */
  -webkit-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
  -moz-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
  -o-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
  transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
  /* ease-out */
  -webkit-transition-delay: 100ms;
  -o-transition-delay: 100ms;
  transition-delay: 100ms;
  display: block;
  position: absolute;
  top: 50%;
  left: 50%;
  -webkit-transform: translate(-50%, -50%);
  -ms-transform: translate(-50%, -50%);
  -o-transform: translate(-50%, -50%);
  transform: translate(-50%, -50%);
  font-size: 16px;
  color: #252525;
}
```

```
.cart .add-to-cart {
  -webkit-transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
  -moz-transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
  -o-transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
  transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
  /* ease-out */
  -webkit-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
  -moz-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
  -o-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
  transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
  /* ease-out */
  -webkit-transition-delay: 100ms;
  -o-transition-delay: 100ms;
  transition-delay: 100ms;
  display: block;
  position: absolute;
  top: 50%;
  left: 110%;
  -webkit-transform: translate(-50%, -50%);
  -ms-transform: translate(-50%, -50%);
  -o-transform: translate(-50%, -50%);
}
```

```
transform: translate(-50%, -50%);
}
```

```
.cart .add-to-cart .txt {
  font-size: 8px;
  color: #fff;
  letter-spacing: 0.045em;
  text-transform: uppercase;
  white-space: nowrap;
}
* {
  box-sizing: border-box;
}
```

```
/* Create three equal columns that floats next to each other */
.column {
  float: left;
  width: 33.33%;
  padding: 10px;
  height: 300px; /* Should be removed. Only for demonstration */
}
```

```
/* Clear floats after the columns */
.row:after {
  content: "";
  display: table;
  clear: both;
}
```

```
/* Responsive layout - makes the three columns stack on top of each other instead of next to each other */
@media screen and (max-width: 600px) {
  .column {
    width: 100%;
  }
}
</style>
<div id="bgclr">
{%if products%}
<h3 style="text-align:center; color:yellow;">{{word}}</h3>
<br>
<!-- products data start----->
<div class="row">
```

```
  {% for p in products %}
  <!-- card1 -->
  <div class="column">
  <div class="container page-wrapper" style="width: 450px;">
    <div class="page-inner">
      <div class="row">
        <div class="el-wrapper">
          <div class="box-up">
```

```

        
        <div class="img-info">
            <div class="info-inner">
                <span style="background-color:rgba(95,95,95,0.3); color:rgb(15, 15, 15);"class="p-
company">{{ p.name }}</span>

                </div>
                <div class="a-size">{{ p.description }}</div>
            </div>
        </div>

        <div class="box-down">
            <div class="h-bg">
                <div class="h-bg-inner"></div>
            </div>

            <a class="cart" href="{% url 'add-to-cart' p.id %}">
                <span class="price">₹ {{ p.price }}</span>
                <span class="add-to-cart">
                    <span class="txt">Add to cart</span>
                </span>
            </a>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- card1 end -->
{% if forloop.counter|divisibleby:"3" %}
</div>

<div class="row">

    <br><br><br><br><br>
{% endif %}
{% endfor %}
</div>

<!-- products data enddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd -->
{%else%}
<h3 style="text-align:center; color:yellow;">No Search Found</h3>
{%endif%}

<script>
    {%if messages %}
    {%for message in messages%}
    alert('{{ message }}');
    {%endfor%}

```

```
    {%endif% }
</script>
<br><br><br>
<br><br><br><br><br><br><br><br><br><br>
{% endblock content %}
</div>
```

Admin panal

```
{% extends 'ecom/admin_base.html' %}
{% load widget_tweaks %}
{% block content %}
```

```
<head>
<style media="screen">
a:link {
    text-decoration: none;
}
```

```
.note {
    text-align: center;
    height: 80px;
    background: -webkit-linear-gradient(left, #0072ff, #130f14);
    color: #fff;
    font-weight: bold;
    line-height: 80px;
}
```

```
.form-content {
    padding: 5%;
    border: 1px solid #ced4da;
    margin-bottom: 2%;
}
```

```
.form-control {
    border-radius: 1.5rem;
}
```

```
.btnSubmit {
    border: none;
    border-radius: 1.5rem;
    padding: 1%;
    width: 20%;
    cursor: pointer;
    background: #0062cc;
    color: #fff;
}
```

```
.menu {
    top: 50px;
}
</style>
```



```

<link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet"
id="bootstrap-css">
<script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
</head>
<br><br><br><br><br><br>
<form method="post" enctype="multipart/form-data">
  {% csrf_token %}
  <div class="container register-form">
    <div class="form">
      <div class="note">
        <p>ADD PRODUCTS</p>
      </div>
      <div class="form-content">
        <div class="row">
          <div class="col-md-6">
            <div class="form-group">
              {% render_field productForm.name class="form-control" placeholder="Product Name" %}
            </div>
            <div class="form-group">
              {% render_field productForm.description class="form-control" placeholder="Description"
%}
            </div>

          </div>
          <div class="col-md-6">
            <div class="form-group">
              {% render_field productForm.price class="form-control" placeholder="Price" %}
            </div>
            <div class="form-group">
              {% render_field productForm.product_image class="form-control"
placeholder="product_image" %}
            </div>

          </div>
        </div>
        <button type="submit" class="btnSubmit">ADD</button>
      </div>
    </div>
  </form>
{% endblock content %}

```

Admin Dash

```

<!DOCTYPE html>
<html lang="en" dir="ltr">

```

```

<head>
  <meta charset="utf-8">
  <title></title>

```

```
<style media="screen">
.market-update-block {
  padding: 2em 2em;
  background: #999;
}

.market-update-block h3 {
  color: #fff;
  font-size: 2.5em;
  font-family: 'Carrois Gothic', sans-serif;
}

.market-update-block h4 {
  font-size: 1.2em;
  color: #fff;
  margin: 0.3em 0em;
  font-family: 'Carrois Gothic', sans-serif;
}

.market-update-block p {
  color: #fff;
  font-size: 0.8em;
  line-height: 1.8em;
}

.market-update-block.clr-block-1 {
  background: #ff0000;
  margin-right: 0.8em;
  box-shadow: 0 2px 5px 0 rgba(0, 0, 0, 0.16), 0 2px 10px 0 rgba(0, 0, 0, 0.12);
  transition: 0.5s all;
  -webkit-transition: 0.5s all;
  -moz-transition: 0.5s all;
  -o-transition: 0.5s all;
}

.market-update-block.clr-block-2 {
  background: #FC8213;
  margin-right: 0.8em;
  box-shadow: 0 2px 5px 0 rgba(0, 0, 0, 0.16), 0 2px 10px 0 rgba(0, 0, 0, 0.12);
  transition: 0.5s all;
  -webkit-transition: 0.5s all;
  -moz-transition: 0.5s all;
  -o-transition: 0.5s all;
}

.market-update-block.clr-block-3 {
  background: #1355f9;
  box-shadow: 0 2px 5px 0 rgba(0, 0, 0, 0.16), 0 2px 10px 0 rgba(0, 0, 0, 0.12);
  transition: 0.5s all;
```

```
-webkit-transition: 0.5s all;
-moz-transition: 0.5s all;
-o-transition: 0.5s all;
}
```

```
.market-update-block.clr-block-1:hover {
  background: #3C3C3C;
  transition: 0.5s all;
  -webkit-transition: 0.5s all;
  -moz-transition: 0.5s all;
  -o-transition: 0.5s all;
}
```

```
.market-update-block.clr-block-2:hover {
  background: #3C3C3C;
  transition: 0.5s all;
  -webkit-transition: 0.5s all;
  -moz-transition: 0.5s all;
  -o-transition: 0.5s all;
}
```

```
.market-update-block.clr-block-3:hover {
  background: #3C3C3C;
  transition: 0.5s all;
  -webkit-transition: 0.5s all;
  -moz-transition: 0.5s all;
  -o-transition: 0.5s all;
}
```

```
.market-update-right i.fa.fa-user-o {
  font-size: 3em;
  color: #68AE00;
  width: 80px;
  height: 80px;
  background: #fff;
  text-align: center;
  border-radius: 49px;
  -webkit-border-radius: 49px;
  -moz-border-radius: 49px;
  -o-border-radius: 49px;
  line-height: 1.7em;
}
```

```
.market-update-right i.fa.fa-shopping-basket {
  font-size: 3em;
  color: #FC8213;
  width: 80px;
  height: 80px;
  background: #fff;
  text-align: center;
  border-radius: 49px;
  -webkit-border-radius: 49px;
}
```

```
-moz-border-radius: 49px;
-o-border-radius: 49px;
line-height: 1.7em;
}
```

```
.market-update-right i.fa.fa-bar-chart {
font-size: 3em;
color: #337AB7;
width: 80px;
height: 80px;
background: #fff;
text-align: center;
border-radius: 49px;
-webkit-border-radius: 49px;
-moz-border-radius: 49px;
-o-border-radius: 49px;
line-height: 1.7em;
}
```

```
.market-update-left {
padding: 0px;
}
```

</style>

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

</head>

<body>

<div class="market-updates">

<div class="col-md-4 market-update-gd">

<div class="market-update-block clr-block-1">

<div class="col-md-8 market-update-left">

<h3>{{ customercount }}</h3>

<h4>Total Customer</h4>

</div>

<div class="col-md-4 market-update-right">

<i class="fa fa-user-o"></i>

</div>

<div class="clearfix"> </div>

</div>

</div>

<div class="col-md-4 market-update-gd">

<div class="market-update-block clr-block-2">

<div class="col-md-8 market-update-left">

<h3>{{ productcount }}</h3>

<h4>Total Product</h4>

```

        </div>
        <div class="col-md-4 market-update-right">
            <i class="fa fa-shopping-basket"></i>
        </div>
        <div class="clearfix"> </div>
    </div>
</div>
<div class="col-md-4 market-update-gd">
    <div class="market-update-block clr-block-3">
        <div class="col-md-8 market-update-left">
            <h3>{{ ordercount }}</h3>
            <h4>Total Orders</h4>

            </div>
            <div class="col-md-4 market-update-right">
                <i class="fa fa-bar-chart"> </i>
            </div>
            <div class="clearfix"> </div>
        </div>
    </div>

</div>
<br><br>
</body>
</html>
{% extends 'ecom/admin_base.html' %}
{% load widget_tweaks %}
{% block content %}

<head>
<style media="screen">
a:link {
    text-decoration: none;
}

.note {
    text-align: center;
    height: 80px;
    background: -webkit-linear-gradient(left, #0072ff, #8811c5);
    color: #fff;
    font-weight: bold;
    line-height: 80px;
}

.form-content {
    padding: 5%;
    border: 1px solid #ced4da;
    margin-bottom: 2%;
}

.form-control {

```

```
border-radius: 1.5rem;
}
```

```
.btnSubmit {
border: none;
border-radius: 1.5rem;
padding: 1%;
width: 20%;
cursor: pointer;
background: #0062cc;
color: #fff;
}
```

```
.menu {
top: 50px;
}
</style>
```

```
<link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet"
id="bootstrap-css">
<script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
</head>
<br><br><br><br>
<form method="post" enctype="multipart/form-data">
{ % csrf_token % }
<div class="container register-form">
<div class="form">
<div class="note">
<p>Update Customer Details</p>
</div>
<div class="form-content">
<div class="row">
<div class="col-md-6">
<div class="form-group">
{ % render_field userForm.first_name class="form-control" placeholder="First Name" % }
</div>
<div class="form-group">
{ % render_field userForm.username class="form-control" placeholder="Username" % }
</div>
<div class="form-group">
{ % render_field customerForm.mobile class="form-control" placeholder="Mobile" % }
</div>
<div class="form-group">
{ % render_field customerForm.profile_pic class="form-control" placeholder="Profile
Picture" % }
</div>
</div>
<div class="col-md-6">
<div class="form-group">
{ % render_field userForm.last_name class="form-control" placeholder="Last Name" % }
</div>
<div class="form-group">
{ % render_field userForm.password class="form-control" placeholder="Password" % }
```

```

        </div>
        <div class="form-group">
            { % render_field customerForm.address class="form-control" placeholder="Address" % }
        </div>

        </div>
    </div>
    <button type="submit" class="btnSubmit">Update</button>
</div>
</div>
</div>
</form>
{% endblock content %}

{% extends 'ecom/homebase.html' %}
{% load static %}
{% block content %}

<head>
    <link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet"
id="bootstrap-css">
    <script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>
    <script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
    <style media="screen">
        .button {
            display: inline-block;
            border-radius: 4px;
            background-color: #f4511e;
            border: none;
            color: #FFFFFFF;
            text-align: center;
            font-size: 28px;
            padding: 20px;
            width: 200px;
            transition: all 0.5s;
            cursor: pointer;
            margin: 5px;
        }

        .button span {
            cursor: pointer;
            display: inline-block;
            position: relative;
            transition: 0.5s;
        }

        .button span:after {
            content: '\00bb';
            position: absolute;
            opacity: 0;
            top: 0;
            right: -20px;

```

```
    transition: 0.5s;
}
```

```
.button:hover span {
    padding-right: 25px;
}
```

```
.button:hover span:after {
    opacity: 1;
    right: 0;
}
```

```
.button2 {
    background-color: white;
    color: black;
    border: 2px solid #008CBA;
}
```

```
.button2:hover {
    background-color: #008CBA;
    color: white;
}
```

```
</style>
```

```
</head>
```

```
<br><br><br><br>
```

```
<div class="container">
  <div class="panel panel-primary">
    <div class="panel-heading">
      <h6 style="text-align:center;" class="panel-title">My Cart</h6>
    </div>
    <table class="table table-hover" id="dev-table">
      <thead>
        <tr>
          <th>Name</th>
          <th>Image</th>
          <th>Price</th>
          <th>Description</th>
          <th>Remove From Cart</th>

          </tr>
        </thead>
        {% for p in products %}
        {% if p %}

          <tr>
```



```

        <td> {{ p.name }} </td>
        <td> </td>
        <td> {{ p.price }} </td>
        <td> {{ p.description }} </td>

        <td><a class="btn btn-danger btn-xs" href="{% url 'remove-from-cart' p.id %}"><span
class="glyphicon glyphicon-trash"></span></a></td>

    </tr>
    {% else % }
    <br><br><br>
    <h1>no products</h1>
    {% endif % }
    {% endfor % }
</table>
</div>
</div>
<br><br><br>
<div style="text-align: center;">
    <button class="button button2">Total {{ total }} </button>
    <a href="/customer-address"><button class="button" ><span>Purchase </span></button></a>
</div>

<br><br><br><br><br>
{% endblock content % }

{% extends 'ecom/customer_base.html' % }
{% load widget_tweaks % }
{% block content % }

<head>
<style media="screen">
    body {
        background-color: #ffffff;
    }
    a:link {
        text-decoration: none;
    }

    .note {
        text-align: center;
        height: 80px;
        background: -webkit-linear-gradient(left, #0072ff, #8811c5);
        color: #fff;
        font-weight: bold;
        line-height: 80px;
    }

```

```
.form-content {  
  padding: 5%;  
  border: 1px solid #ced4da;  
  margin-bottom: 2%;  
}
```

```
.form-control {  
  border-radius: 1.5rem;  
}
```

```
.btnSubmit {  
  border: none;  
  border-radius: 1.5rem;  
  padding: 1%;  
  width: 20%;  
  cursor: pointer;  
  background: #0062cc;  
  color: #fff;  
}
```

```
.menu{  
  top: 50px;  
}  
</style>
```

```
</head>  
<br><br><br><br><br>  
<form method="post" enctype="multipart/form-data">  
  {% csrf_token %}  
  <div class="container register-form">  
    <div class="form">  
      <div class="note">  
        <p>EDIT PROFILE</p>  
      </div>  
      <div class="form-content">  
        <div class="row">  
          <div class="col-md-6">  
            <div class="form-group">  
              {% render_field userForm.first_name class="form-control" placeholder="First Name" %}  
            </div>  
            <div class="form-group">  
              {% render_field userForm.username class="form-control" placeholder="Username" %}  
            </div>  
            <div class="form-group">  
              {% render_field customerForm.mobile class="form-control" placeholder="Mobile" %}  
            </div>  
            <div class="form-group">  
              {% render_field customerForm.profile_pic class="form-control" placeholder="Profile  
Picture" %}  
            </div>  
          </div>  
        </div>  
      </div>  
    </div>  
  </div>
```

```

    <div class="form-group">
      { % render_field userForm.last_name class="form-control" placeholder="Last Name" % }
    </div>
    <div class="form-group">
      { % render_field userForm.password class="form-control" placeholder="Password" % }
    </div>
    <div class="form-group">
      { % render_field customerForm.address class="form-control" placeholder="Address" % }
    </div>

    </div>
  </div>
  <button type="submit" class="btnSubmit">Edit</button>
</div>
</div>
</div>
</form>
{ % endblock content % }

```

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
{ % load static % }

```

```

<head>
<meta charset="utf-8">
<style>
  @page {
    size: A4;
    margin: 1cm;

```

```

    @frame footer {
      -pdf-frame-content: footerContent;
      bottom: 0cm;
      margin-left: 9cm;
      margin-right: 9cm;
      height: 1cm;
    }
  }
}

```

```

.invoice-box {
  max-width: 800px;
  margin: auto;
  padding: 30px;
  border: 1px solid #eee;
  box-shadow: 0 0 10px rgba(0, 0, 0, .15);
  font-size: 16px;
  line-height: 24px;
  font-family: 'Helvetica Neue', 'Helvetica', Helvetica, Arial, sans-serif;
  color: #555;
}

```

```
.invoice-box table {  
  width: 100%;  
  line-height: inherit;  
  text-align: left;  
}
```

```
.invoice-box table td {  
  padding: 5px;  
  vertical-align: top;  
}
```

```
.invoice-box table tr td:nth-child(2) {  
  text-align: right;  
}
```

```
.invoice-box table tr.top table td {  
  padding-bottom: 20px;  
}
```

```
.invoice-box table tr.top table td.title {  
  font-size: 45px;  
  line-height: 45px;  
  color: #333;  
}
```

```
.invoice-box table tr.information table td {  
  padding-bottom: 40px;  
}
```

```
.invoice-box table tr.heading td {  
  background: #eee;  
  border-bottom: 1px solid #ddd;  
  font-weight: bold;  
}
```

```
.invoice-box table tr.details td {  
  padding-bottom: 20px;  
}
```

```
.invoice-box table tr.item td {  
  border-bottom: 1px solid #eee;  
}
```

```
.invoice-box table tr.item.last td {  
  border-bottom: none;  
}
```

```
.invoice-box table tr.total td:nth-child(2) {  
  border-top: 2px solid #eee;  
  font-weight: bold;  
}
```

```
@media only screen and (max-width: 600px) {  
  .invoice-box table tr.top table td {  
    width: 100%;  
    display: block;  
    text-align: center;  
  }  
}
```

```
.invoice-box table tr.information table td {  
  width: 100%;  
  display: block;  
  text-align: center;  
}  
}
```

```
/** RTL **/  
.rtl {  
  direction: rtl;  
  font-family: Tahoma, 'Helvetica Neue', 'Helvetica', Helvetica, Arial, sans-serif;  
}
```

```
.rtl table {  
  text-align: right;  
}
```

```
.rtl table tr td:nth-child(2) {  
  text-align: left;  
}
```

```
</style>  
</head>
```

```
<body>
```

```
<br><br><br>  
<div class="invoice-box">  
  <table cellpadding="0" cellspacing="0">  
    <tr class="top">  
      <td colspan="2">  
        <table>  
          <tr>  
            <td class="title">  
              <h5>LEXA STORE</h5>  
            </td>  
  
            <td>
```

Order Date: {{orderDate}}

</td>
</tr>
</table>
</td>
</tr>

<tr class="information">
<td colspan="2">
<table>
<tr>
<td>
Customer Name :
{{customerName}}

Customer Email :
{{customerEmail}}

</td>

<td>
Customer Mobile :

{{customerMobile}}

Shipment Address :
{{shipmentAddress}}

</td>
</tr>
</table>
</td>
</tr>

<tr class="information">
<td colspan="2">
<table>
<tr>
<td>
Product Image :

<td>
Product Price :

{{productPrice}}
</td>
<td>
Product Name:
{{productName}}

Product Description:
{{productDescription}}
</td>
</tr>

```
</table>
</td>
</tr>
```

```
<tr class="information">
  <td colspan="2">
    <table>
      <tr>
        <td>
          Order Status :<br><br>
          {{orderStatus}}
        </td>
      </tr>
    </table>
  </td>
</tr>
```

```
</table>
</div>
</body>
</html>
```

```
<!DOCTYPE html>
{% load widget_tweaks %}
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>
```

```
<style media="screen">
  body {
    margin: 0;
    padding: 0;
    font-family: sans-serif;
    background: linear-gradient(to right, #b92b27, #1565c0)
  }
```

```
.box {
  width: 500px;
  padding: 40px;
  position: absolute;
  top: 50%;
```

```
    left: 50%;  
    background: #191919;  
    ;  
    text-align: center;  
    transition: 0.25s;  
    margin-top: 100px  
}
```

```
.box input[type="text"],  
.box input[type="password"] {  
    border: 0;  
    background: none;  
    display: block;  
    margin: 20px auto;  
    text-align: center;  
    border: 2px solid #3498db;  
    padding: 10px 10px;  
    width: 250px;  
    outline: none;  
    color: white;  
    border-radius: 24px;  
    transition: 0.25s  
}
```

```
.box h1 {  
    color: white;  
    text-transform: uppercase;  
    font-weight: 500  
}
```

```
.box input[type="text"]:focus,  
.box input[type="password"]:focus {  
    width: 300px;  
    border-color: #2ecc71  
}
```

```
.box input[type="submit"] {  
    border: 0;  
    background: none;  
    display: block;  
    margin: 20px auto;  
    text-align: center;  
    border: 2px solid #2ecc71;  
    padding: 14px 40px;  
    outline: none;  
    color: white;  
    border-radius: 24px;  
    transition: 0.25s;  
    cursor: pointer  
}
```

```
.box input[type="submit"]:hover {
```



```
    background: #2ecc71
  }

.forgot {
  text-decoration: underline
}

ul.social-network {
  list-style: none;
  display: inline;
  margin-left: 0 !important;
  padding: 0
}

ul.social-network li {
  display: inline;
  margin: 0 5px
}

.social-network a.icoFacebook:hover {
  background-color: #3B5998
}

.social-network a.icoTwitter:hover {
  background-color: #33ccff
}

.social-network a.icoGoogle:hover {
  background-color: #BD3518
}

.social-network a.icoFacebook:hover i,
.social-network a.icoTwitter:hover i,
.social-network a.icoGoogle:hover i {
  color: #fff
}

a.socialIcon:hover,
.socialHoverClass {
  color: #44BCDD
}

.social-circle li a {
  display: inline-block;
  position: relative;
  margin: 0 auto 0 auto;
  border-radius: 50%;
  text-align: center;
```

```

width: 50px;
height: 50px;
font-size: 20px
}

```

```

.social-circle li i {
margin: 0;
line-height: 50px;
text-align: center
}

```

```

.social-circle li a:hover i,
.triggeredHover {
transform: rotate(360deg);
transition: all 0.2s
}

```

```

.social-circle i {
color: #fff;
transition: all 0.8s;
transition: all 0.8s
}

```

```

</style>
</head>
<body>
{% include "ecom/navbar.html" %}
<div class="container">
<div class="row">
<div class="col-md-6">
<div class="card">
<form class="box" method="post" enctype="multipart/form-data">
{% csrf_token %}
<h1>CUSTOMER SIGNUP</h1>
<p class="text-muted"> Please enter your details to create account !</p>
{% render_field userForm.first_name class="form-control" placeholder="First Name"
%}
{% render_field userForm.last_name class="form-control" placeholder="Last Name"
%}
{% render_field userForm.username class="form-control" placeholder="Username"
%}
{% render_field userForm.password class="form-control" placeholder="Password"
%}
{% render_field customerForm.address class="form-control"
placeholder="Address" %}
{% render_field customerForm.mobile class="form-control"
placeholder="Mobile" %}
{% render_field customerForm.profile_pic class="form-control"
placeholder="Profile pic" %}

<input type="submit" name="" value="Create">

</form>
</div>

```

```

    </div>
  </div>
</div>
<br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br>
<br><br><br><br><br>
{% include "ecom/footer.html" % }
</body>
</html>

```

```

<!DOCTYPE html>
{% load widget_tweaks %}
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>E-Commerce</title>
    <style media="screen">
      body {
        margin: 0;
        padding: 0;
        font-family: sans-serif;
        background: linear-gradient(to right, #b92b27, #1565c0)
      }
    </style>
  </head>
  <body>
    <div class="box">
      <div class="input">
        <input type="text"/>
      </div>
    </div>
  </body>
</html>

```

```

.box {
  width: 500px;
  padding: 40px;
  position: absolute;
  top: 50%;
  left: 50%;
  background: #191919;
  ;
  text-align: center;
  transition: 0.25s;
  margin-top: 100px
}

```

```

.box input[type="text"],
.box input[type="password"] {
  border: 0;
  background: none;
  display: block;
  margin: 20px auto;
  text-align: center;
  border: 2px solid #3498db;
  padding: 10px 10px;
  width: 250px;
  outline: none;
  color: white;
  border-radius: 24px;
  transition: 0.25s
}

```

```
.box h1 {  
  color: white;  
  text-transform: uppercase;  
  font-weight: 500  
}
```

```
.box input[type="text"]:focus,  
.box input[type="password"]:focus {  
  width: 300px;  
  border-color: #2ecc71  
}
```

```
.box input[type="submit"] {  
  border: 0;  
  background: none;  
  display: block;  
  margin: 20px auto;  
  text-align: center;  
  border: 2px solid #2ecc71;  
  padding: 14px 40px;  
  outline: none;  
  color: white;  
  border-radius: 24px;  
  transition: 0.25s;  
  cursor: pointer  
}
```

```
.box input[type="submit"]:hover {  
  background: #2ecc71  
}
```

```
.forgot {  
  text-decoration: underline  
}
```

```
ul.social-network {  
  list-style: none;  
  display: inline;  
  margin-left: 0 !important;  
  padding: 0  
}
```

```
ul.social-network li {  
  display: inline;  
  margin: 0 5px  
}
```

```
.social-network a.icoFacebook:hover {
```

```
background-color: #3B5998
}
```

```
.social-network a.icoTwitter:hover {
background-color: #33ccff
}
```

```
.social-network a.icoGoogle:hover {
background-color: #BD3518
}
```

```
.social-network a.icoFacebook:hover i,
.social-network a.icoTwitter:hover i,
.social-network a.icoGoogle:hover i {
color: #fff
}
```

```
a.socialIcon:hover,
.socialHoverClass {
color: #44BCDD
}
```

```
.social-circle li a {
display: inline-block;
position: relative;
margin: 0 auto 0 auto;
border-radius: 50%;
text-align: center;
width: 50px;
height: 50px;
font-size: 20px
}
```

```
.social-circle li i {
margin: 0;
line-height: 50px;
text-align: center
}
```

```
.social-circle li a:hover i,
.triggeredHover {
transform: rotate(360deg);
transition: all 0.2s
}
```

```
.social-circle i {
color: #fff;
transition: all 0.8s;
transition: all 0.8s
}
```

```

}
</style>
</head>
<body>
  {% include "ecom/navbar.html" %}
  <div class="container">
    <div class="row">
      <div class="col-md-6">
        <div class="card">
          <form class="box" method="post">
            {% csrf_token %}
            <h1>Customer Login</h1>
            <p class="text-muted"> Please enter your login and password!</p>
            {% render_field form.username class="form-control" placeholder="Username" %}
            {% render_field form.password class="form-control" placeholder="Password" %}
            <input type="submit" name="" value="Login" >

          </form>
        </div>
      </div>
    </div>
  </div>
  <br><br><br><br><br><br><br><br>
  <br><br><br><br><br><br><br><br>
  <br><br><br><br><br><br><br><br>
  {% include "ecom/footer.html" %}
</body>
</html>

```

```

{% load static %}
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>

```

```

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.min.css">
    <!-- jQuery library -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

```

```

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
    <!-- Latest compiled JavaScript -->
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.0/js/bootstrap.min.js"></script>

```

```

<style>
#logo {
height: 10px;
width: 40px;

```

```
position: absolute;
z-index: 300;
}
body {
  background-color: #ebe0e0
}
```

```
#navbar {
  background-color: #000;
  color: black;
}
```

```
.row1 {
  padding-top: 10px;
  padding-right: 50px
}
```

```
.row2 {
  padding-bottom: 20px
}
```

```
.navbar-input {
  padding: 11px 16px;
  border-radius: 2px 0 0 2px;
  border: 0 none;
  outline: 0 none;
  font-size: 15px
}
```

```
.navbar-button {
  /* search button */
  background-color: rgb(149, 245, 149);
  border: 1px solid rgb(163, 243, 163);
  border-radius: 0 2px 2px 0;
  color: #ffffff;
  padding: 10px 0;
  height: 43px;
  cursor: pointer
}
```

```
.cart-button {
  background-color: rgb(149, 245, 149);
  box-shadow: 0 4px 4px 0 rgba(0, 0, 0, .23), inset 1px 1px 0 0 hsla(0, 0%, 100%, .2);
  padding: 10px 0;
  text-align: center;
  height: 41px;
  border-radius: 2px;
  font-weight: 500;
  width: 120px;
  display: inline-block;
  color: #FFFFFF;
}
```

```
text-decoration: none;
color: inherit;
border: none;
outline: none
}
```

```
.cart-button:hover {
text-decoration: none;
color: #fff;
cursor: pointer
}
```

```
.cart-svg {
display: inline-block;
width: 16px;
height: 16px;
vertical-align: middle;
margin-right: 8px
}
```

```
.item-number {
border-radius: 3px;
background-color: rgba(0, 0, 0, .1);
height: 20px;
padding: 3px 6px;
font-weight: 500;
display: inline-block;
color: #fff;
line-height: 12px;
margin-left: 10px
}
```

```
.upper-links {
display: inline-block;
padding: 0 11px;
line-height: 23px;
font-family: 'Roboto', sans-serif;
letter-spacing: 0;
color: inherit;
border: none;
outline: none;
font-size: 12px
}
```

```
.dropdown {
position: relative;
display: inline-block;
margin-bottom: 0px
}
```

```
.dropdown:hover {
```



```
    background-color: #fff  
}
```

```
.dropdown:hover .links {  
    color: #000  
}
```

```
.dropdown:hover .dropdown-menu {  
    display: block  
}
```

```
.dropdown .dropdown-menu {  
    position: absolute;  
    top: 100%;  
    display: none;  
    background-color: #fff;  
    color: #333;  
    left: 0px;  
    border: 0;  
    border-radius: 0;  
    box-shadow: 0 4px 8px -3px #555454;  
    margin: 0;  
    padding: 0px  
}
```

```
.links {  
    color: #fff;  
    text-decoration: none  
}
```

```
.links:hover {  
    color: #fff;  
    text-decoration: none  
}
```

```
.profile-links {  
    font-size: 12px;  
    font-family: 'Roboto', sans-serif;  
    border-bottom: 1px solid #e9e9e9;  
    box-sizing: border-box;  
    display: block;  
    padding: 0 11px;  
    line-height: 23px  
}
```

```
.profile-li {  
    padding-top: 2px  
}
```

```
.largenav {
  display: none
}
```

```
.smallnav {
  display: block
}
```

```
.smallsearch {
  margin-left: 15px;
  margin-top: 15px
}
```

```
.menu {
  cursor: pointer
}
```

```
@media screen and (min-width: 768px) {
  .largenav {
    display: block
  }
```

```
  .smallnav {
    display: none
  }
```

```
  .smallsearch {
    margin: 0px
  }
}</style>
```

```
</head>
```

```
<body>
  <div id="navbar">
    <div class="container">
      <div class="row row1">
        <ul class="largenav pull-right">

          <li class="upper-links"><a class="links" href="/my-order">Orders</a></li>
          <li class="upper-links"><a class="links" href="/my-profile">Profile</a></li>
          <li class="upper-links"><a class="links" href="/logout">Logout</a></li>
          <li class="upper-links dropdown"><a class="links" href="">More</a>
            <ul class="dropdown-menu">
              <li class="profile-li"><a class="profile-links" href="/send-feedback">Send
Feedback</a></li>
```

```

        <li class="profile-li"><a class="profile-links" href="/contactus">Contact
Us</a></li>
        <li class="profile-li"><a class="profile-links" href="/aboutus">About Us</a></li>
    </ul>
</li>
</ul>
</div>
<div class="row row2">
    <div class="col-sm-2">
        <!-- <h2 style="margin:0px;"><a style="text-decoration:none; color:white;"
href="/"><span class="largenav">LEXA STORE </span></a></h2> -->
        <div id="logo">
            <a href="/">
                
            </a>
        </div>
    </div>
</div>

```

```

<div class="navbar-search smallsearch col-sm-8 col-xs-11">
    <form action="/search" method="get">
        <div class="row"> <input class="navbar-input col-xs-11" type="search"
placeholder="Search for Products, Brands and more" name="query" id="query">
        <button class="navbar-button col-xs-1" type="submit">
            <svg width="15px" height="15px">
                <path d="M11.618 9.89714.224 4.212c.092.09.1.23.02.3121-1.464 1.46c-.08.08-.
.222.072-.314-.02L9.868 11.66M6.486 10.9c-2.42 0-4.38-1.955-4.38-4.367 0-2.413 1.96-4.37 4.38-
4.37s4.38 1.957 4.38 4.37c0 2.412-1.96 4.368-4.38 4.368m0-10.834C2.904.066 0 2.96 0 6.533 0
10.105 2.904 13 6.486 13s6.487-2.895 6.487-6.467c0-3.572-2.905-6.467-6.487-6.467 "></path>
            </svg>
        </button>
    </div>
</form>
</div>

```

```

<div class="cart1 largenav col-sm-2"> <a class="cart-button" href="/cart"> <svg
class="cart-svg " width="16 " height="16 " viewBox="0 0 16 16 ">
    <path d="M15.32 2.405H4.887C3 2.405 2.46.805 2.46.805L2.257.21C2.208.085
2.083 0 1.946 0H.336C.1 0-.064.24.024.461.644 1.945L3.11 9.767c.047.137.175.23.32.23h8.418l-
.493 1.958H3.768l.002.003c-.017 0-.033-.003-.05-.003-1.06 0-1.92.86-1.92 1.92s.86 1.92 1.92
1.92c.99 0 1.805-.75 1.91-1.712l5.55.076c.12.922.91 1.636 1.867 1.636 1.04 0 1.885-.844 1.885-
1.885 0-.866-.584-1.593-1.38-1.814l2.423-8.832c.12-.433-.206-.86-.655-.86 " fill="#fff "></path>
    </svg> Cart <span class="item-number ">{{ product_count_in_cart }}</span> </a>
</div>
</div>
</div>
</div>

```

```

<script>
$(document).ready(function(){

```

```
function openNav() {  
document.getElementById("mySidenav").style.width = "70%";
```

```
  
document.body.style.backgroundColor = "rgba(0,0,0,0.4)";  
}
```

```
  
function closeNav() {  
document.getElementById("mySidenav").style.width = "0";  
document.body.style.backgroundColor = "rgba(0,0,0,0)";  
}
```

```
  
});  
</script>
```

```
  
</body>  
</html>
```

```
  
{% extends 'ecom/customer_base.html' %}  
{% load static %}  
{% block content %}
```

```
  
<style media="screen">  
body,  
html {  
height: 100%;  
}
```

```
  
.d-flex {  
display: -webkit-box;  
display: -ms-flexbox;  
display: flex;  
}
```

```
  
.align-center {  
-webkit-box-align: center;  
-ms-flex-align: center;  
align-items: center;  
}
```

```
  
.flex-centerY-centerX {  
justify-content: center;  
-webkit-box-pack: center;  
-ms-flex-pack: center;
```

```
justify-content: center;
-webkit-box-align: center;
-ms-flex-align: center;
align-items: center;
}
```

```
body {
  background-image: linear-gradient(to bottom right, rgb(0, 0, 0) 50%, rgb(131, 131, 131));
}
```

```
.page-wrapper {
  height: 100%;
  display: table;
}
```

```
.page-wrapper .page-inner {
  display: table-cell;
  vertical-align: middle;
}
```

```
.el-wrapper {
  width: 360px;
  padding: 15px;
  margin: 15px auto;
  background-color: #fff;
}
```

```
@media (max-width: 991px) {
  .el-wrapper {
    width: 345px;
  }
}
```

```
@media (max-width: 767px) {
  .el-wrapper {
    width: 290px;
    margin: 30px auto;
  }
}
```

```
.el-wrapper:hover .h-bg {
  left: 0px;
}
```

```
.el-wrapper:hover .price {
  left: 20px;
  -webkit-transform: translateY(-50%);
}
```

```
-ms-transform: translateY(-50%);
-o-transform: translateY(-50%);
transform: translateY(-50%);
color: #818181;
}
```

```
.el-wrapper:hover .add-to-cart {
  left: 50%;
}
```

```
.el-wrapper:hover .img {
  webkit-filter: blur(7px);
  -o-filter: blur(7px);
  -ms-filter: blur(7px);
  filter: blur(7px);
  filter: progid:DXImageTransform.Microsoft.Blur(pixelradius='7', shadowopacity='0.0');
  opacity: 0.4;
}
```

```
.el-wrapper:hover .info-inner {
  bottom: 155px;
}
```

```
.el-wrapper:hover .a-size {
  -webkit-transition-delay: 300ms;
  -o-transition-delay: 300ms;
  transition-delay: 300ms;
  bottom: 50px;
  opacity: 1;
}
```

```
.el-wrapper .box-down {
  width: 100%;
  height: 60px;
  position: relative;
  overflow: hidden;
}
```

```
.el-wrapper .box-up {
  width: 100%;
  height: 300px;
  position: relative;
  overflow: hidden;
  text-align: center;
}
```

```
.el-wrapper .img {
  padding: 20px 0;
  -webkit-transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
  -moz-transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
}
```

```
-o-transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
-webkit-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-moz-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-o-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
}
```

```
.h-bg {
-webkit-transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
-moz-transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
-o-transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
transition: all 800ms cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
-webkit-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-moz-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-o-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
width: 660px;
height: 100%;
background-color: #3f96cd;
position: absolute;
left: -659px;
}
```

```
.h-bg .h-bg-inner {
width: 50%;
height: 100%;
background-color: #464646;
}
```

```
.info-inner {
-webkit-transition: all 400ms cubic-bezier(0, 0, 0.18, 1);
-moz-transition: all 400ms cubic-bezier(0, 0, 0.18, 1);
-o-transition: all 400ms cubic-bezier(0, 0, 0.18, 1);
transition: all 400ms cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
-webkit-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-moz-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-o-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
position: absolute;
width: 100%;
bottom: 25px;
}
```

```
.info-inner .p-name,
.info-inner .p-company {
display: block;
```

```
}
```

```
.info-inner .p-name {  
  font-family: 'PT Sans', sans-serif;  
  font-size: 18px;  
  color: #252525;  
}
```

```
.info-inner .p-company {  
  font-family: 'Lato', sans-serif;  
  font-size: 12px;  
  text-transform: uppercase;  
  color: #FFE11B;  
}
```

```
.a-size {  
  -webkit-transition: all 300ms cubic-bezier(0, 0, 0.18, 1);  
  -moz-transition: all 300ms cubic-bezier(0, 0, 0.18, 1);  
  -o-transition: all 300ms cubic-bezier(0, 0, 0.18, 1);  
  transition: all 300ms cubic-bezier(0, 0, 0.18, 1);  
  /* ease-out */  
  -webkit-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);  
  -moz-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);  
  -o-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);  
  transition-timing-function: cubic-bezier(0, 0, 0.18, 1);  
  /* ease-out */  
  position: absolute;  
  width: 100%;  
  bottom: -20px;  
  font-family: 'PT Sans', sans-serif;  
  color: #828282;  
  opacity: 0;  
}
```

```
.a-size .size {  
  color: #252525;  
}
```

```
.cart {  
  display: block;  
  position: absolute;  
  width: 100%;  
  height: 100%;  
  top: 0;  
  left: 0;  
  font-family: 'Lato', sans-serif;  
  font-weight: 700;  
}
```

```
.cart .price {  
  -webkit-transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
```



```

-moz-transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
-o-transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
-webkit-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-moz-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-o-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
-webkit-transition-delay: 100ms;
-o-transition-delay: 100ms;
transition-delay: 100ms;
display: block;
position: absolute;
top: 50%;
left: 50%;
-webkit-transform: translate(-50%, -50%);
-ms-transform: translate(-50%, -50%);
-o-transform: translate(-50%, -50%);
transform: translate(-50%, -50%);
font-size: 16px;
color: #252525;
}

```

```

.cart .add-to-cart {
-webkit-transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
-moz-transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
-o-transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
transition: all 600ms cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
-webkit-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-moz-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
-o-transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
transition-timing-function: cubic-bezier(0, 0, 0.18, 1);
/* ease-out */
-webkit-transition-delay: 100ms;
-o-transition-delay: 100ms;
transition-delay: 100ms;
display: block;
position: absolute;
top: 50%;
left: 110%;
-webkit-transform: translate(-50%, -50%);
-ms-transform: translate(-50%, -50%);
-o-transform: translate(-50%, -50%);
transform: translate(-50%, -50%);
}

```

```

.cart .add-to-cart .txt {
font-size: 12px;
color: #fff;
letter-spacing: 0.045em;
text-transform: uppercase;
white-space: nowrap;
}

```

```
* {
  box-sizing: border-box;
}
```

```
/* Create three equal columns that floats next to each other */
.column {
  float: left;
  width: 33.33%;
  padding: 10px;
  height: 300px; /* Should be removed. Only for demonstration */
}
```

```
/* Clear floats after the columns */
.row:after {
  content: "";
  display: table;
  clear: both;
}
```

```
/* Responsive layout - makes the three columns stack on top of each other instead of next to each other */
@media screen and (max-width: 600px) {
  .column {
    width: 100%;
  }
}
</style>
```

```
{% if products %}
<h3 style="text-align:center; color:yellow;">{{ word }}</h3>
<br>
<!-- products data start----->
<div class="row">
```

```
  {% for p in products %}
  <!-- card1 -->
  <div class="column">
    <div class="container page-wrapper" style="width: 450px;">
      <div class="page-inner">
        <div class="row">
          <div class="el-wrapper">
            <div class="box-up">
              
              <div class="img-info">
                <div class="info-inner">
                  <span style="background-color:#08050a; color:yellow;" class="p-
company">{{ p.name }}</span>
```

```

        </div>
        <div class="a-size">{{ p.description }}</div>
    </div>
</div>

<div class="box-down">
    <div class="h-bg">
        <div class="h-bg-inner"></div>
    </div>

    <a class="cart" href="{% url 'add-to-cart' p.id %}">
        <span class="price">₹ {{ p.price }}</span>
        <span class="add-to-cart">
            <span class="txt">Add in cart</span>
        </span>
    </a>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- card1 end -->
{% if forloop.counter|divisibleby:"3" %}
    </div>
    <div class="row">
        <br><br><br><br><br>
    </div>
{% endif %}

{% endfor %}
</div>
<!-- products data enddddddddddddddddddddddddddddddddddddddddddddddddddddddd -->

{% else %}
<h3 style="text-align:center; color:yellow;">No Search Found</h3>
{% endif %}

<script>
    {% if messages %}
    {% for message in messages %}
    alert('{{ message }}');
    {% endfor %}
    {% endif %}
</script>
<br><br><br>
<br><br><br><br><br><br><br><br><br>
{% endblock content %}

<html>

```

```
{% load static %}
<head>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.min.css">
<!-- jQuery library -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.0/js/bootstrap.min.js"></script>
```

```
<style>
body {
    background-color: #ebe0e0
}
```

```
#navbar {
    background-color: #000;
    color: black;
}
```

```
.row1 {
    padding-top: 10px;
    padding-right: 50px
}
```

```
.row2 {
    padding-bottom: 20px
}
```

```
.navbar-input {
    padding: 11px 16px;
    border-radius: 2px 0 0 2px;
    border: 0 none;
    outline: 0 none;
    font-size: 15px
}
```

```
.navbar-button {
    background-color: rgb(149, 245, 149);
    border: 1px solid rgb(163, 243, 163);
    border-radius: 0 2px 2px 0;
    color: #565656;
    padding: 10px 0;
    height: 43px;
    cursor: pointer
}
```

```
.cart-button {
```

```
background-color: rgb(149, 245, 149);;
box-shadow: 0 4px 4px 0 rgba(0, 0, 0, .23), inset 1px 1px 0 0 hsla(0, 0%, 100%, .2);
padding: 10px 0;
text-align: center;
height: 41px;
border-radius: 2px;
font-weight: 500;
width: 120px;
display: inline-block;
color: #FFFFFF;
text-decoration: none;
color: inherit;
border: none;
outline: none
}

.cart-button:hover {
  text-decoration: none;
  color: #fff;
  cursor: pointer
}

.cart-svg {
  display: inline-block;
  width: 16px;
  height: 16px;
  vertical-align: middle;
  margin-right: 8px
}

.item-number {
  border-radius: 3px;
  background-color: rgba(0, 0, 0, .1);
  height: 20px;
  padding: 3px 6px;
  font-weight: 500;
  display: inline-block;
  color: #fff;
  line-height: 12px;
  margin-left: 10px
}

.upper-links {
  display: inline-block;
  padding: 0 11px;
  line-height: 23px;
  font-family: 'Roboto', sans-serif;
  letter-spacing: 0;
  color: inherit;
  border: none;
  outline: none;
  font-size: 12px
}
```

```
.dropdown {  
  position: relative;  
  display: inline-block;  
  margin-bottom: 0px  
}
```

```
.dropdown:hover {  
  background-color: #fff  
}
```

```
.dropdown:hover .links {  
  color: #000  
}
```

```
.dropdown:hover .dropdown-menu {  
  display: block  
}
```

```
.dropdown .dropdown-menu {  
  position: absolute;  
  top: 100%;  
  display: none;  
  background-color: #fff;  
  color: #333;  
  left: 0px;  
  border: 0;  
  border-radius: 0;  
  box-shadow: 0 4px 8px -3px #555454;  
  margin: 0;  
  padding: 0px  
}
```

```
.links {  
  color: #fff;  
  text-decoration: none  
}
```

```
.links:hover {  
  color: #fff;  
  text-decoration: none  
}
```

```
.profile-links {  
  font-size: 12px;  
  font-family: 'Roboto', sans-serif;  
  border-bottom: 1px solid #e9e9e9;  
  box-sizing: border-box;  
  display: block;
```

```
padding: 0 11px;
line-height: 23px
}
```

```
.profile-li {
padding-top: 2px
}
```

```
.largenav {
display: none
}
```

```
.smallnav {
display: block
}
```

```
.smallsearch {
margin-left: 15px;
margin-top: 15px
}
```

```
.menu {
cursor: pointer
}
```

```
/*-----
Social section
-----*/
```

```
footer {
padding: 0px 0px 0px 0px;
background-color: black;
margin: 0px;
}
```

```
#ftr {

padding: 20px;
}
```

```
.fa {

font-size: 23px;
width: 60px;
text-align: center;
text-decoration: none;
margin: 5px 2px;
```

```
border-radius: 50%;  
}
```

```
.fa:hover {  
  opacity: 0.5;  
  text-decoration: none;  
}
```

```
.fa-facebook {  
  background: #3B5998;  
  color: white;  
  margin-top: 30px;  
}
```

```
.fa-whatsapp {  
  background: #25d366;  
  color: white;  
}
```

```
.fa-twitter {  
  background: #55ACEE;  
  color: white;  
}
```

```
.fa-instagram {  
  background: #125688;  
  color: white;  
}
```

```
p {  
  text-align: center;  
  
}
```

```
@media screen and (min-width: 768px) {  
  .largenav {  
    display: block  
  }  
}
```

```
.smallnav {  
  display: none  
}
```

```
.smallsearch {  
  margin: 0px  
}
```



```

}
</style>
</head>
<body>
<div id="navbar">
  <div class="container">
    <div class="row row1">
      <ul class="largenav pull-right">

        <li class="upper-links"><a class="links" href="/my-order">Orders</a></li>
        <li class="upper-links"><a class="links" href="/my-profile">Profile</a></li>
        <li class="upper-links"><a class="links" href="/logout">Logout</a></li>
        <li class="upper-links dropdown"><a class="links" href="">More</a>
          <ul class="dropdown-menu">
            <li class="profile-li"><a class="profile-links" href="/send-feedback">Send
Feedback</a></li>
            <li class="profile-li"><a class="profile-links" href="/contactus">Contact Us</a></li>
            <li class="profile-li"><a class="profile-links" href="/aboutus">About Us</a></li>
          </ul>
        </li>
      </ul>
    </div>
    <div class="row row2">
      <div class="col-sm-2">
        <!-- <h2 style="margin:0px;"><a style="text-decoration:none; color:white;" href="/"><span
class="largenav">LEXA STORE </span></a></h2> -->
        <div id="logo">
          <a href="/">
            
          </a>
        </div>
      </div>

      <div class="navbar-search smallsearch col-sm-8 col-xs-11">
        <form action="/search" method="get">
          <div class="row"> <input class="navbar-input col-xs-11" type="search"
placeholder="Search for Products, Brands and more" name="query" id="query">
            <button class="navbar-button col-xs-1" type="submit">
              <svg width="15px" height="15px">
                <path d="M11.618 9.89714.224 4.212c.092.09.1.23.02.312l-1.464 1.46c-.08.08-
.222.072-.314-.02L9.868 11.66M6.486 10.9c-2.42 0-4.38-1.955-4.38-4.367 0-2.413 1.96-4.37 4.38-
4.37s4.38 1.957 4.38 4.37c0 2.412-1.96 4.368-4.38 4.368m0-10.834C2.904.066 0 2.96 0 6.533 0
10.105 2.904 13 6.486 13s6.487-2.895 6.487-6.467c0-3.572-2.905-6.467-6.487-6.467 "></path>
              </svg>
            </button>
          </div>
        </form>
      </div>

      <div class="cart1 largenav col-sm-2"> <a class="cart-button" href="/cart"> <svg class="cart-
svg " width="16 " height="16 " viewBox="0 0 16 16 ">
        <path d="M15.32 2.405H4.887C3 2.405 2.46.805 2.46.805L2.257.21C2.208.085
2.083 0 1.946 0H.336C.1 0-.064.24.024.461.644 1.945L3.11 9.767c.047.137.175.23.32.23h8.418l-

```

```

.493 1.958H3.768l.002.003c-.017 0-.033-.003-.05-.003-1.06 0-1.92.86-1.92 1.92s.86 1.92 1.92
1.92c.99 0 1.805-.75 1.91-1.712l5.55.076c.12.922.91 1.636 1.867 1.636 1.04 0 1.885-.844 1.885-
1.885 0-.866-.584-1.593-1.38-1.814l2.423-8.832c.12-.433-.206-.86-.655-.86 " fill="#fff "></path>
</svg> Cart <span class="item-number ">{{ product_count_in_cart }}</span> </a>
</div>
</div>
</div>
</div>
<!-- content start-->
{% block content %}

{% endblock content %}
<!-- content end-->
<br><br><br>
<footer>
<p>
<a id="ftr" href="#" class="fa fa-facebook"></a>
<a id="ftr" href="#" class="fa fa-whatsapp"></a>
<a id="ftr" href="#" class="fa fa-instagram"></a>
<a id="ftr" href="#" class="fa fa-twitter"></a>
</p>

<br>
</footer>
<script>
$(document).ready(function(){

function openNav() {
document.getElementById("mySidenav").style.width = "70%";

document.body.style.backgroundColor = "rgba(0,0,0,0.4)";
}

function closeNav() {
document.getElementById("mySidenav").style.width = "0";
document.body.style.backgroundColor = "rgba(0,0,0,0)";
}

});
</script>

</body>
</html>

```



CHAPTER 6

TESTING & IMPLIMENTATION

6.1 Need for Testing

Testing is the major quality control measure and during software development. Its basic is to detect errors in the software. The system is tested to its accuracy, speed data entry and response .system requires a test plan, which specifies conditions, different units to be tested and the manner in which modules to be integrated together. After a test plan has been development, system testing begins with testing program modules separately followed by testing bundles modules as an limit.

6.2 Black Box Testing

The black box testing takes an external perspective of the test objective to derive lest case this test can be functional or non-functional, though usually functional. The test designer selects valid and invalid input and determines the correct output. There is no knowledge of the test objects internal structure. This method of test design applicable to all level of software testing; unit, integration, functional testing, system and acceptance .The higher the level, and hence bigger and more complex the box, the more one is forced to use black box testing to simplify. While this method can uncover unimplemented part of the specification, one cannot be sure that all existent paths are tested. Black box testing is not type of testing; instead is a testing strategy, which does not need any knowledge of internal design or code etc. As the same "black box" suggest, no knowledge of internal logic or code structure is required.

6.3 White Box Testing

White box testing (A.K.A .Clear box testing, glass box testing or structural testing)uses an internal perspective of the system to design test cases based on internal structural requires programming skill to identify all through the software. The tester chooses test case input to exercise path through the code and determines the appropriate output.In this type of testing inside the programmer himself identifies the control and flow condition testing inside the

program and decides the test data so that it will pass through all the statement. By passing the data to the system will identify pick out as logical errors and conditional errors.

6.4 Unit Testing

Unit testing is the best method to verify the logic and coding, commutation/ functionality and error handling of each module .It involves testing the entire software ,unit by unit as the online voting system is developed as different modules. Here white box testing has been adopted, which assures the full coverage of the code. The white box testing is rigorous activity. It is a case design method that uses the control structure of the procedural design to derive test cases.

6.5 Integration Testing

For the online cline management system integration test was conducted. In this phase of software testing in which individual software modules are combined and tested as a group .If follow unit testing and precedes system testing. Integration testing takes as its inputs modules that have checked out by unit testing groups them larger aggregates, and delivers as its output the integrated system ready for system testing. The modules that has passed the unit testing like admin and membership detail view are combined together to check in information module are combined together to check if the prepared result is correct.

6.6 System testing

This provides the final assurance that the software meets all the functional, behavioral and performance requirements .The software is completely assembled as a package validation succeeds when the software function is a manner in which user wishes validation refer to the process of using software in live environment in order to find errors .During the course of validation the system failure may occur and sometimes the coding as to be changed according to the requirement. Thus the feedback from the validation phase generally produces changes in the software. Once the application was made all logical and interface errors, inputting dummy date ensured that the software developed satisfied all the requirements of the user .This dummy date known as test case.

In this the system as a whole is tested against the design specifications;

- system testing will be responsibility of testing engineer
- system and Integration testing shall be performed to ensure that the system work as a whole,
- system testing will be performed in the test environment located on the server,
- A system test plan will be prepared by the test manager
- system testing will be performed in the test environment located on the server,
- system testing will include volume testing(with a high number of transaction processed) and stress testing (with transactions processed at high frequency)
- Result of tests will be recorded and where system components do not perform as expected, a test problem report will be raised.

6.7 Validation Testing

It involves checking process which makes sure that software conforms to its specification and meets the needs of the customer. It starts requirement review, continues through design and code review to product testing. They are concerned with the analysis and checking of system representation such as the requirement document, design diagram and the program code. They even involve executing an implementation of the software with test data and examining the output of the software to check that it is performing as required.

6.8 INTEGRATION TESTING

Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. In the project, all the modules are combined and then the entire program is tested as a whole. In the integration-testing step, all the error uncovered is corrected for the next testing steps.

6.9 Output testing

After performing the validation testing, the next step is output testing of the proposed system. Since, no system would be termed as useful unit it does produce the requested output in the specified format. Here the output format is considered in two ways.

One is on the screen and other is on the printer. The output forms on the screen are found to be correct since the format was designed according to user needs. For hard copy also the output comes out as the specified requirements of the work. Here the output testing does not result in any correction in the system.

- User acceptance testing

The system has met the user's requirement in the following case.

- Data entry
- Error handling
- Reporting and correcting
- System output
- Implementation

Implementation involves:

- Software installation

In order to run the package "SALE OR RENT" the software's like visual python and Django server must be installed.

- User testing

After installing the package the user must be explained about the module functionalities and screen working.

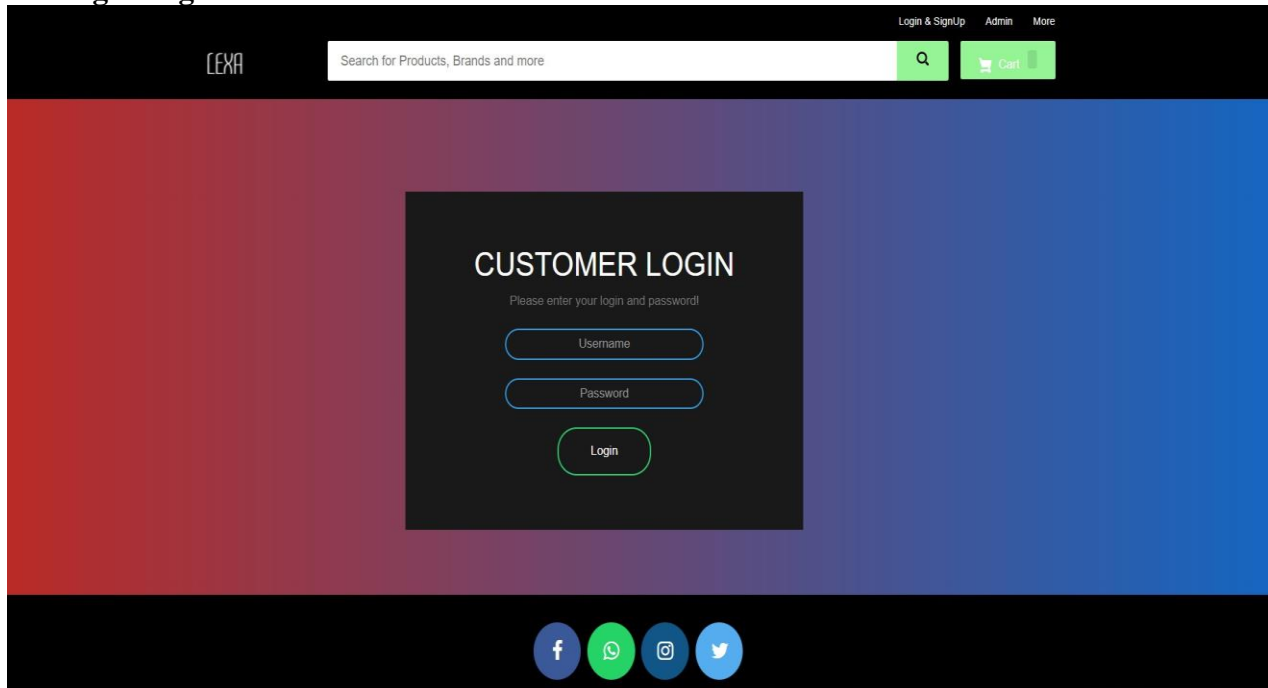
- ☐ The tools used for implementations are Windows 10 operating system.
- ☐ Django server as back-end interface.
- ☐ HTML as the front-end interface.
- ☐ Triggers for various operations.
- ☐ Reporting generation using web browser data reporting designer.

CHAPTER 7

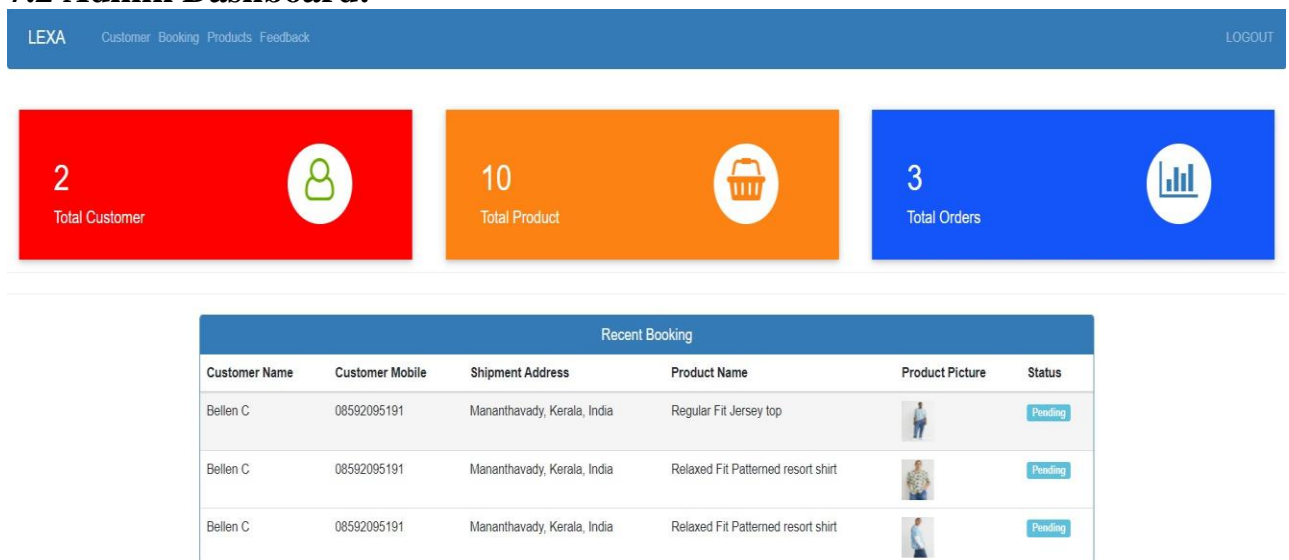
OUTPUT SCREENS

In this chapter, we are going to display some of the output screens of online food management system to give the idof how the software looks like in the execution part.

7.1 Login Page:



7.2 Admin Dashboard:



7.4 SignUp Page

CEXA

Search for Products, Brands and more

Q

Cart 0

CUSTOMER SIGNUP

Please enter your details to create account !

First Name

Last Name

Username

Password

Address

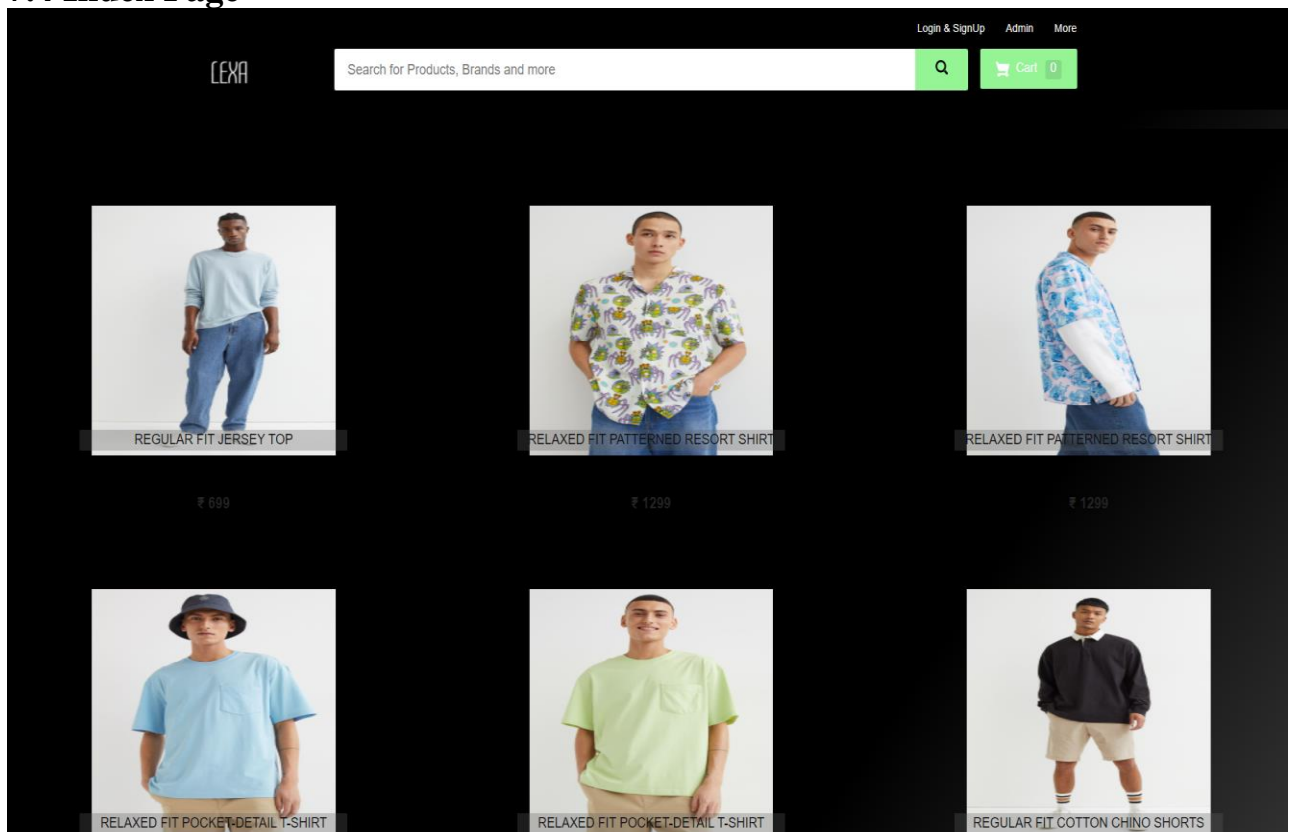
Mobile

Choose File

No file chosen

Create

7.4 Index Page

































7.5 Manager Control Panel:


LEXA

CustomerBookingProductsFeedback

LOGOUT

Total Product List

Name	Image	Price	Description	Update	Delete
Regular Fit Jersey top		699	Light blue		
Relaxed Fit Patterned resort shirt		1299	White/Rick and Morty		
Relaxed Fit Patterned resort shirt		1299	Light pink/SpongeBob		
Relaxed Fit pocket-detail T-shirt		799	Light blue		
Relaxed Fit pocket-detail T-shirt		799	Neon green		
Regular Fit Cotton chino shorts		799	Light beige		
Regular Fit Cotton T-shirt		799	Light grey/Pixels		
Relaxed Fit Printed sweatshorts		799	White		
Slim Fit Polo shirt		799	Turquoise		
Shacket		3999	Light beige/Black		




7.6 Cart Page

LEXA









Search for Products, Brands and more

Q

 Cart 4

OrdersProfileLogoutMore

My Cart

Name	Image	Price	Description	Remove From Cart
Regular Fit Jersey top		699	Light blue	
Relaxed Fit Patterned resort shirt		1299	White/Rick and Morty	
Relaxed Fit Patterned resort shirt		1299	Light pink/SpongeBob	
Relaxed Fit pocket-detail T-shirt		799	Neon green	

Total 4096


Purchase

7.7 Contact Page:

CEXA

Search for Products, Brands and more

Q

 Cart

Orders Profile Logout More


Send Us Your Valuable Feedback !


Name:


Email:


Message:

Send Message









CHAPTER 8

CONCLUSION

This online Shopping management system helps to enhance and develop skills, abilities and knowledge. New features, modules, and other components can be incorporated into the system as per user requirements in future. It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of programming in python web based application and Django Server, and also about all handling procedure related with "Online Shopping Management System". It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

BIBLOGRAPHY

Github forum

<https://github.community/>

Stackoverflow

<https://stackoverflow.blog/community/>

Django projects

<https://www.djangoproject.com/>