Mini Project Report

on

DECH

Submitted By:

NAMRATA TIWARI(161500342)

NIKHIL PANDEY(161500356)

YASH JALAN(161500638)

PRATEEK PANDEY(161500404)

VISHISHTA AGRAWAL(161500626)

Institute of Engineering & Technology

ABSTRACT

This project is a web based shopping system for an existing shop. The project objective is to deliver the online shopping products.This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet through website. Thus the customer will get the service of online shopping and home delivery from his favorite shop. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the customers to the trending online shops such as flipkart or ebay.

ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. I extend my sincere and heartfelt thanks to our esteemed guide, **Mr. MANDEEP SINGH SIR**, for providing me with the right guidance and advice at the crucial junctures and for showing me the right way. I extend my sincere thanks to our respected **Head of the division**, for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during the course of our work.

Signature:

TABLE OF CONTENTS

INTRODUCTION

This project is a web based shopping system for an existing shop. The project objective is to deliver the online shopping application into android platform. Online shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using website. Thus the customer will get the service of online shopping and home delivery from his favorite shop.

3.1 PROJECT OBJECTIVE

The objective of the project is to make a website to purchase items in an existing shop. In order to build such an application complete web support need to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project.

3.2 PROJECT SCOPE

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

3.3 PROJECT OVERVIEW

The central concept of the application is to allow the customer to shop virtually using the Internet and allow customers to buy the items and articles of their desire from the store. The information pertaining to the products are stores on an RDBMS at the server side (store).

The Server process the customers and the items are shipped to the address submitted by them. The application was designed into two modules first is for the customers who wish to buy the articles. Second is for the storekeepers who maintains and updates the information pertaining to the articles and those of the customers. The end user of this product is a departmental store where the application is hosted on the web and the administrator maintains the database. The application which is deployed at the customer database, the details of the items are brought forward from the database for the customer view based on the selection through the menu and the database of all the products are updated at the end of each transaction. Data entry into the application can be done through various screens designed for various levels of users. Once the authorized personnel feed the relevant data into the system, several reports could be generated as per the security

4. ADMINISTRATOR

The administrator is the super user of this application. Only admin have access into this admin page. Admin may be the owner of the shop. The administrator has all the information about all the users and about all products.

This module is divided into different sub-modules.

1. Manage Products
2. Manage Users
3. Manage Orders

1. MANAGE PRODUCTS

➢ Add Products

The shopping cart project contains different kind of products. The products can be classified into different categories by name. Admin can add new products into the existing system with all its details including an image.

➢ Delete Products

Administrator can delete the products based on the stock of that particular product.

2. MANAGE USER

➢ View Users

The admin will have a list view of all the users registered in the system. Admin can view all the details of each user in the list except password.

➢ **Add Users**

Admin has privileges to add a user directly by providing the details.

➢ **Delete and Block Users**

Administrator has a right to delete or block a user. The default status of a new user registered is set as blocked. The admin must accept the new user by unblocking him.

3. MANAGE ORDERS

➢ **View Order**

Administrator can view the Orders which is generated by the users. He can verify the details of the purchase.

➢ **Delete order**

Admin can delete order from the orders list when the product is taken for delivery.

4.2 MODERATORS

A moderator is considered as a staff who can manage orders for the time being. As a future update moderator may give facility to add and manage his own products . Moderators can reduce the work load of admin. Now moderator has all the privilege an admin having except managing other moderators. He can add products and users. He can also check the orders and edit his profile.

➢ Manage products

- Manage user
- Manage orders

4.3 USERS

A new user will have to register in the system by providing essential details in order to view the products in the system. The admin must accept a new user by unblocking him.

Login

A user must login with his user name and password to the system after registration.

View Products

User can view the list of products based on their names after successful login. A detailed description of a particular product with product name, products details, product image, price can be viewed by users.

Search Product

Users can search for a particular product in the list by name.

Add to cart:

- The user can add the desired product into his cart by clicking add to cart option on the product.
- He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove.

Submit Cart:

After confirming the items in the cart the user can submit the cart by providing a delivery address.On successful submitting the cart will become empty.

➢ **History**

In the history the user will have a view of pending orders.

➢ **Edit Profile**

The user can view and edit the profile

5. SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem solving activity that requires intensive communication between the system users and system developers.

System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

5.1 EXISTING SYSTEM

The current system for shopping is to visit the shop manually and from the available product choose the item customer want and buying the item by

payment of the price of the item.

1. It is less user-friendly.
2. User must go to shop and select products
3. It is difficult to identify the required product.
4. Description of the product limited.
5. It is a time consuming process
6. Not in reach of distant users.

5.2 PROPOSED SYSTEM

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

6. SYSTEM REQUIREMENT SPECIFICATION

6.1 GENERAL DESCRIPTION

1.Product Description:

The system consists of a web page .A web application which can provide the online shopping service for the customer to access the web service from his Web page. Web application should be able to help the customer for selecting his item and to help the owner in managing the orders from the customers.

2.Problem Statement:

As online shopping became a trend nowadays the regular shops are losing their customers to online brands. Customers have effortless shopping experience and saving time through shopping online. For competing with those online brands , If shops are providing an online portal where their customers can shop through internet and get the products at their doors it will increase the number of customers.

6.2 SYSTEM OBJECTIVES

➢ To provide a web application for online shopping of products in an existing shop.

7.NON FUNCTIONAL REQUIREMENTS

i. EFFICIENCY REQUIREMENT

When an online shopping cart web application implemented customer can purchase product in an efficient manner.

ii. RELIABILITY REQUIREMENT

The system should provide a reliable environment to both customers and owner. All orders should be reaching at the admin without any errors.

iii. USABILITY REQUIREMENT

The android application is designed for user friendly environment and ease of use.

iv. IMPLEMENTATION REQUIREMENT

Implementation of the system using css and html in front end with jsp as

back end and it will be used for database connectivity. And the database part is developed by mysql. Responsive web designing is used for making the website compatible for any type of screen.

v. DELIVERY REQUIREMENT

The whole system is expected to be delivered in four months of time with a weekly evaluation by the project guide

USER

➢ **USER LOGIN**

This feature used by the user to login into system. A user must login with his username and password to the system after registration. If they are invalid, the user not allowed to enter the system.

Functional requirement

- Username and password will be provided after user registration is confirmed.
- Password should be hidden from others while typing it in the field

➢ **REGISTER NEW USER**

A new user will have to register in the system by providing essential details in order to view the products in the system. The admin must accept a new user by unblocking him.

- System must be able to verify and validate information.
- The system must encrypt the password of the customer to provide security.

PURCHASING AN ITEMS

The user can add the desired product into his cart by clicking add to cart option on the product. He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove. After confirming the items in the cart the user can submit the cart by providing a delivery address. On successful submitting the cart will become empty.

- System must ensure that, only a registered customer can purchase items.

ADMIN

➢ MANAGE USER

The administrator can add user, delete user, view user and block user.

➢ MANAGE MODERATOR

The administrator can add moderator, delete moderator, block moderator and search for a moderator.

➢ MANAGE PRODUCTS

The administrator can add product, delete product and view product.

➢ MANAGE ORDERS

The administrator can view orders and delete orders.

-The system must identify the login of the admin.

-Admin account should be secured so that only owner of the shop can access that account.

8.FUNCTIONAL REQUIREMENT

8.1 Provide shopping cart facility -

- The system shall provide shopping cart during online purchase.
- The system shall allow user to add/remove products in the shopping cart. Later customer can confirm orders for purchase.

8.2 Online tracking of shipments -

- The system shall allow user to enter the order information for tracking.
- The system shall display the current tracking information about the order The system notifies seller about delivery of product to the consumer.

8.3 Allow multiple payment methods -

- The system shall display available payment methods for payment.
- The system shall allow user to select the payment method for order

8.4 Provide Customer Support. -

- The system shall provide online help, FAQ's customer support, and sitemap options for customer support.
- The system shall allow user to select the support type he wants.
- The system shall allow user to enter the customer and product information for the support.
- The system shall display user contact of seller and a support desk
- The system shall display the online help upon request.
- The system shall display the FAQ's upon request.

9. Modules

9.1. Customer module

The main purpose of this module is to provide all the functionality related to customer . It tracks all the information and details of the customer. We have developed all type of CRUD (Create, Read, Update and Delete) operations of the customers. This is a role based module where admin can perform each and every operations on data but the customer will be able to view only his/her data, so access level restrictions has also been implemented on the project.

Features :

- Admin can add new customers records
- Admin can see the list of customers details
- Only admin can edit and update the record of the customers
- Admin will be able to delete the records of the customers
- All customers forms are validated on client side using JavaScript
- Customer will be able to see his detail.
- Customer will be able to update his details

9.2 Product module

The main purpose for developing this module is to manage the product data wise. So all product will be managed by admin and customer will be able able to see product. Admin can see the list of all the product and filter it according to the customers.

Features:

- Admin can manage the product
- Admin can edit/delete the product
- Admin can see the list of all product
- Customer can see product

9.3. Product Type Module

The main purpose for developing this module is to manage the product type. So all product company will be managed by admin and customer will be able to see the product type.

Features:

- Admin can manage the product type
- Admin can edit/delete the product type
- Admin can see the list of all product type
- Customer can see product type

9.4. Product Company Module

The main purpose for developing this module is to manage the product company. So all product company will be managed by admin and customer will be able to see the product company.

Features:

- Admin can manage the product company
- Admin can edit/delete the product company
- Admin can see the list of all product company
- Customer can see product company

9.5 Order Module

The main purpose for developing this module is to manage the customer orders. Order is the main module in this project. DECH which has been developed on Python, Django and MySQL So all orders will be managed by admin and customer will be able to see his order and their payment receipt.

Features:

- Admin can manage the order
- Admin can edit/delete the order
- Admin can see the list of all order

10. User Characteristics

10.1. Customer :

He or she is a verified user of website who is intended to buy a product from the seller via the Dech platform. The customer must have a username and password to make a purchase. The person is regularly updated and fed with latest offers and discounts according their interest.

10.2. Seller :

He or she is a verified person who is allowed to sell items over the platform. Seller’s details are stored on database and all the products are listed under

him that he is ready to sell or are available. He is responsible to set products details, price, and quantity.

10.3. Administrator :

He or she is responsible for monitoring functions and procedures on platform. Administrator is responsible to provide valid information of a purchase to the concerned authority in case of any dispute between the customer and seller or in case of exchange.

11. Software Interfaces

HTML :- Page layout has been designed in HTML.

CSS :- CSS has been used for all the desigining part.

Javascript :-All the validation task and animations has been developed by Javascript.

Python :- All the business logic has been implemented in Python.

MYSQL :- MYSQL database has been used as database for the project.

Django :- Project has been developed over the Django framework.

Pycharm :- PyCharm is an integrated development environment used in computer programming, specifically for the Python language.

CODES

Code for Order-form

```
from django import forms

from .models import Order

class OrderCreateForm(forms.ModelForm):

  class Meta:

    model = Order

    fields = ['first_name', 'last_name', 'email', 'address', 'postal_code', 'city']
```

Code for Orders-View

```
from django.shortcuts import render

from .models import OrderItem

from .forms import OrderCreateForm

from cart.cart import Cart

def order_create(request):

  cart = Cart(request)

  if request.method == 'POST':

    form = OrderCreateForm(request.POST)

    if form.is_valid():

      order = form.save()

      for item in cart:

        OrderItem.objects.create(

          order=order,

          product=item['product'],

          price=item['price'],

          quantity=item['quantity']

        )

      cart.clear()

    return render(request, 'orders/order/created.html', {'order': order})

  else:

    form = OrderCreateForm()
```

```
    return render(request, 'orders/order/create.html', {'form': form})
```

Code for Orders-Model

```
from django.db import models
from shop.models import Product
class Order(models.Model):
    first_name = models.CharField(max_length=60)
    last_name = models.CharField(max_length=60)
    email = models.EmailField()
    address = models.CharField(max_length=150)
    postal_code = models.CharField(max_length=30)
    city = models.CharField(max_length=100)
    created = models.DateTimeField(auto_now_add=True)
    updated = models.DateTimeField(auto_now=True)
    paid = models.BooleanField(default=False)
    class Meta:
        ordering = ('-created', )
    def __str__(self):
        return 'Order {}'.format(self.id)

    def get_total_cost(self):
```

```
    return sum(item.get_cost() for item in self.items.all())

class OrderItem(models.Model):

  order = models.ForeignKey(Order, related_name='items', on_delete=models.CASCADE)

  product = models.ForeignKey(Product, related_name='order_items',
on_delete=models.CASCADE)

  price = models.DecimalField(max_digits=10, decimal_places=2)

  quantity = models.PositiveIntegerField(default=1)

  def __str__(self):

    return '{}'.format(self.id)

  def get_cost(self):

    return self.price * self.quantity
```

Code for Order-Admin

```
from django.contrib import admin

from .models import Order, OrderItem

class OrderItemInline(admin.TabularInline):

  model = OrderItem

  raw_id_fields = ['product']

class OrderAdmin(admin.ModelAdmin):

  list_display = ['id', 'first_name', 'last_name', 'email', 'address', 'postal_code', 'city', 'paid',
'created',  'updated']
```

```
  list_filter = ['paid', 'created', 'updated']

  inlines = [OrderItemInline]

admin.site.register(Order, OrderAdmin)
```

Code For Cart

```
from decimal import Decimal

from django.conf import settings

from shop.models import Product

class Cart(object):

  def __init__(self, request):

    self.session = request.session

    cart = self.session.get(settings.CART_SESSION_ID)

    if not cart:

      cart = self.session[settings.CART_SESSION_ID] = {}

    self.cart = cart

 def add(self, product, quantity=1, update_quantity=False):

    product_id = str(product.id)

    if product_id not in self.cart:

      self.cart[product_id] = {'quantity': 0, 'price': str(product.price)}

    if update_quantity:

      self.cart[product_id]['quantity'] = quantity

    else:
```

```
            self.cart[product_id]['quantity'] += quantity
        self.save()

    def save(self):
        self.session[settings.CART_SESSION_ID] = self.cart
        self.session.modified = True

    def remove(self, product):
        product_id = str(product.id)
        if product_id in self.cart:
            del self.cart[product_id]
            self.save()

    def __iter__(self):
        product_ids = self.cart.keys()
        products = Product.objects.filter(id__in=product_ids)
        for product in products:
            self.cart[str(product.id)]['product'] = product

        for item in self.cart.values():
            item['price'] = Decimal(item['price'])
            item['total_price'] = item['price'] * item['quantity']
```

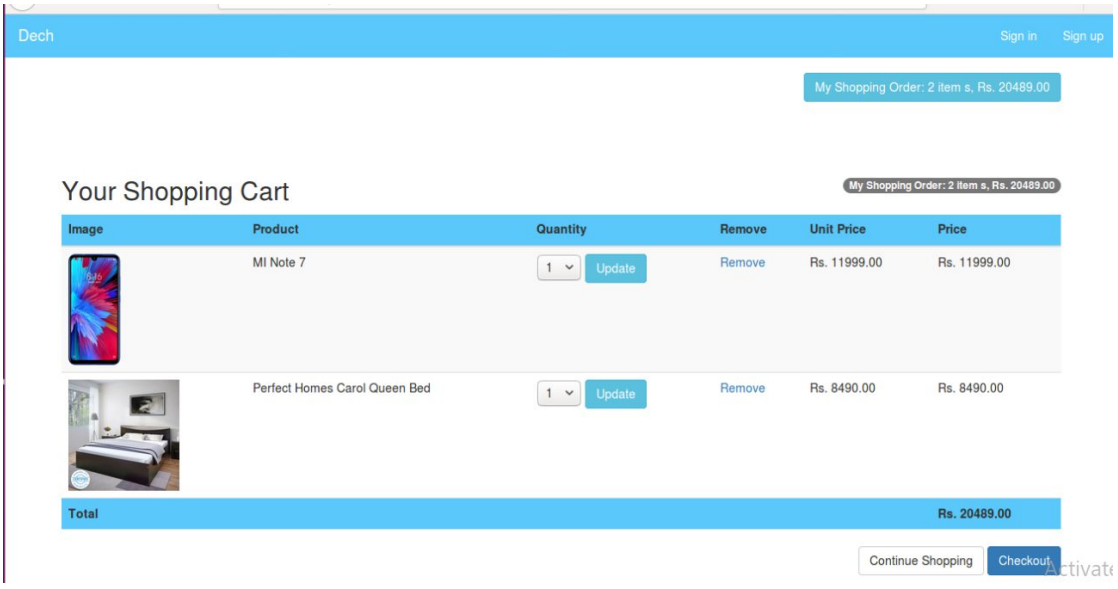
```
        yield item


    def __len__(self):
        return sum(item['quantity'] for item in self.cart.values())


    def get_total_price(self):
        return sum(Decimal(item['price']) * item['quantity'] for item in self.cart.values())


    def clear(self):
        del self.session[settings.CART_SESSION_ID]
        self.session.modified = True
```

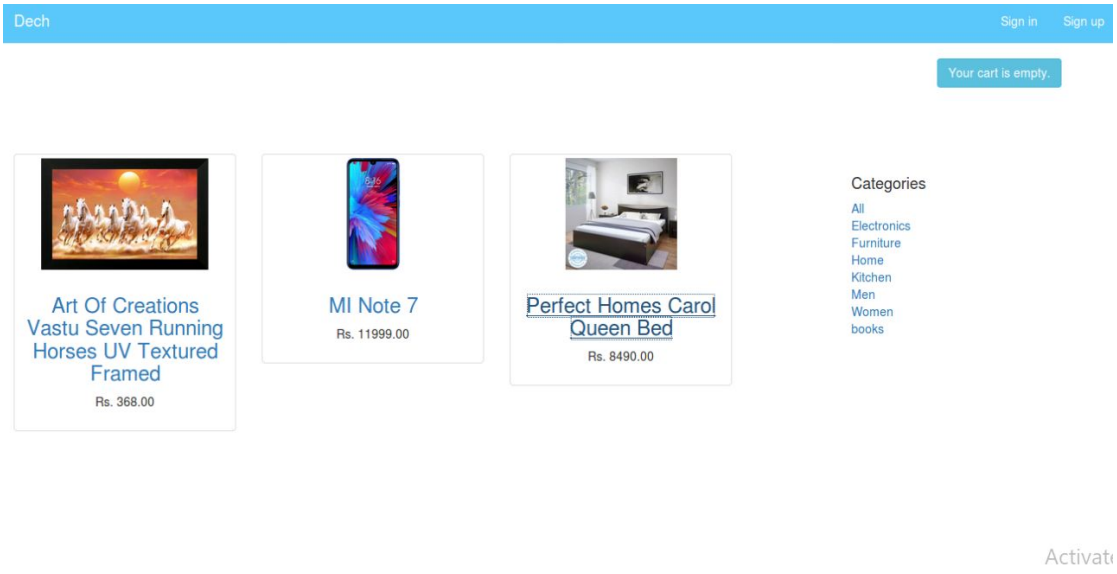
Code for Cart-view

```
from django.shortcuts import render, redirect, get_object_or_404
from django.views.decorators.http import require_POST
from shop.models import Product
from .cart import Cart
from .forms import CartAddProductForm

@require_POST
def cart_add(request, product_id):
    cart = Cart(request)
    product = get_object_or_404(Product, id=product_id)
```

```
    form = CartAddProductForm(request.POST)
    if form.is_valid():
        cd = form.cleaned_data
        cart.add(product=product, quantity=cd['quantity'], update_quantity=cd['update'])
    return redirect('cart:cart_detail')


def cart_remove(request, product_id):
    cart = Cart(request)
    product = get_object_or_404(Product, id=product_id)
    cart.remove(product)
    return redirect('cart:cart_detail')


def cart_detail(request):
    cart = Cart(request)
    for item in cart:
        item['update_quantity_form'] = CartAddProductForm(initial={'quantity': item['quantity'],
'update': True})
    return render(request, 'cart/detail.html', {'cart': cart})
```

SCREENSHOT




Dech
Sign in
Sign up
Your cart is empty.
Art Of Creations Vastu Seven Running Horses UV Textured Framed
Rs. 368.00
MI Note 7
Rs. 11999.00
Perfect Homes Carol Queen Bed
Rs. 8490.00
Categories
All
Electronics
Furniture
Home
Kitchen
Men
Women
books