|  |
| --- |
| **SAVITRIBAI PHULE PUNE UNIVERSITY**    **In Partial Fulfilment of**  **MASTERS IN COMPUTER APPLICATION**  **SINHGAD INSTITUTE OF BUSINESS ADMINISTRATION AND RESEARCH**  **KONDHWA, PUNE-411048**    **2022-2024**  **A**  **PROJECT REPORT**  **ON**  **E-PLANT NUSERY**  **BY**  **Sayli Milind Jawle (94)**  **Kazi Zainab Nazimuddin (138)** |

**CERTIFICATE OF ORIGINALITY**

This is to certify that the Mini Project (SEM I/II/III) report entitled **“E-Plant Nursery Application”** Submitted to the Department of Computer Applications, Sinhgad Institute of Business Administration and is an original work carried out by

Mr/Ms………………………………………………………….Exam No…………….…..

under my guidance.

The matter embodied in this Mini Project is a genuine work done by the student and has not been submitted whether to this Organization or to any other University/ Organization for the fulfillment of the requirement of any course of study.

Signature of the Student :

Name of the student :

Signature of the Guide :

Name and Designation of the Guide :

Signature of Director-MCA

**Dr. Netra Patil**  :

Date :

**CERTIFICATE OF APPROVAL**

This is to certify that the project report entitled…………………………………………………….

………………………………………………………………………………………………………

submitted to the Department of Computer Application, Sinhgad Institute of Business Administration and Research in partial fulfillment of the requirement for the award of the Degree of MASTER OF COMPUTER APPLICATIONS (MCA Affiliated to Savitribai Phule Pune University) is an original work carried out by Mr./Ms……………………………………………..

………………………………………………………………………Exam No……………

The matter embodied in this project is a genuine work done by the student and has been certified by the following internal and external examiners deputed by Savitribai Phule Pune University.

Internal Examiner External Examiner

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Chapter** | **Page No.** |
| 1 | **CHAPTER 1: INTRODUCTION** |  |
| 1.1 | Abstract |  |
| 1.2 | Existing System and Need for System |  |
| 1.3 | Scope of System |  |
| 1.4 | Operating Environment Hardware and Software |  |
| 1.5 | Technology used |  |
| 2 | **CHAPTER 2: PROPOSED SYSTEM** |  |
| 2.1 | Feasibility Study |  |
| 2.2 | Objectives of the proposed system |  |
| 2.3 | Users of the system |  |
| 3 | **CHAPTER 3: ANALYSIS AND DESIGN** |  |
| 3.1 | Entity Relationship Diagram (ERD) |  |
| 3.2 | Class Diagram |  |
| 3.3 | Use Case Diagrams |  |
| 3.4 | Activity Diagram |  |
| 3.5 | Sequence Diagram |  |
| 3.6 | Collaboration Diagram |  |
| 3.8 | Web site map diagram |  |
| 3.9 | Module |  |
| 3.10 | Sample Input and Output Screens |  |
| 3.11 | Functional Requirements |  |
| 3.12 | Non-Functional Requirement |  |
| 4 | **CHAPTER 4: CODING Sample code** |  |
| 5 | **CHAPTER 5: LIMITATIONS OF SYSTEM** |  |
| 6 | **CHAPTER 7: CONCLUSION** |  |
| 7 | **CHAPTER 8: BIBLIOGRAPHY** |  |

**1.INTRODUCTION**

**1.1Abstract**

E–Nursery web application consist purchasing and selling products which is intended to benefit both gardeners and users. This is a unique place for nature enthusiasts to browse & order the best quality of plants online and get them deliveredThe E-Nursery is an online nursery that allows you to shop for plants from the comfort and convenience of your homes.

In this system there are two users Admin and Customer. The admin can view, add, delete, and update flowering plants detail. Customer can view the plants and their details and add it to their cart or keep it in their Wishlist. Customer can view their cart and make payments. A nursery is a place where plants are propagated and grown to a desired age.

**1.2 Existing System and Need for System**

The manual processing is quite time consuming, less accurate in comparison processing. Offline shopping of plants is very tiring. The choice of plants available in the physical/local plant nursery is limited and outdated, owing to several factors. The rate of the product is also decided by the plant seller by taking their maximum profit. The buyer can be fooled is he doesn’t know prices of plants. It is very slow in processing data. It is very difficult to create manual working. It is time Consuming & having Lot of Paper work. Much time and energy are wasted

**1.3 Scope of System**

Main purpose of this project is to buy plants online easily without visiting the nurseries. And this system helps customer to save time and customers can compare Product’s Price with other shopkeeper without spending much time. To evaluate the extent of use of social‐media marketing by nurseries and garden centres. To identify key factors influencing the daily use of social‐media marketing by nurseries and garden centres. To examine the impact of social‐media marketing use on sales. Number of shopkeeper’s can register to the web portal for increase their sale. This application can be used by any user to purchase the online plants and get appropriate information by viewing short summery about the plant’s items through videos. If any changes to make customer can purchase the plants through different payment schemes like debit card, credit card, pat, phone pay, cash on delivery etc.

**1.4 Operating environment Hardware & Software**

**Hardware:**

* Laptop/Desktop
* Minimum 1GB of RAM
* Minimum 1GB of HDD

**Software:**

* Window’s Operating System
* Python 3.7 or its equivalent software
* MySQL server 5.1 must be installed along with MySQL in python program

**1.4 PROBLEM STATEMENT**

Many people want to buy plants and they directly concerned to the nursery and buy the plants but sometimes people don’t know specific information about particular plant items as well as seller which are not technically skilled. Customer does not compare plant price with other shopkeepers at the same time. In nursery there is no facility for online payment only cash may be consumed. We cannot purchase plants through online mode. Limited customers reached to the nursery because sometime customer need to travel for long distance as nursery is far from their home.

The system has the provision of orders entered by the clients along with their contact details, grading specifications, special services, job codes, and amount of request. After an order is entered, an order confirmation report will be sent to the client for review. When all orders have been entered, a surplus for sale report will be created.

**1.5 Technology used**

**Front-End languages:**

1.HTML

2.CSS and Bootstrap

3.Javascript

**Back-End languages:**

1.Python

2.SQLite

**2.Proposed system:**

**2.1 Feasibility Study:**

Introduction:

The E-Plant Nursery a vital sector of the economy, and there is a need for a platform that can provide an efficient and seamless way for E-plant nursery stakeholders to interact. The web-based application aims to provide a one-stop- shop for buyers and sellers of plants to carry out transactions in a secure and transparent manner.

**Market Analysis:**

The market for the E- Plant nursery is vast, with numerous potential customers and competitors. The study has identified that there is a high demand for a web-based application in the E-Plant nursery sector. The platform will target nursery owners, gardeners, wholesalers, and retailers who trade in plants, seeds and fertilizer products. The competition in this sector is relatively low, and there is a potential to capture a significant market share.

**Technical Feasibility:**

The project is technically feasible, and the required hardware and software resources are readily available. provides a rich set of components for building desktop applications, and it can be easily integrated with web technologies such as HTML, CSS, and JavaScript. The development team has the necessary technical expertise to develop and deploy the application.

**Financial Feasibility:**

The project is financially feasible, and the estimated costs of development, deployment, and maintenance are within reasonable limits. The revenue streams include transaction fees, subscription fees, and advertising. The projected return on investment is expected to be positive within three years.

**Operational Feasibility:**

The project is operationally feasible, and the organizational structure, staffing, and training requirements are manageable. The project team will be responsible for managing the day-to-day operations of the application, including customer support, technical support, and marketing.

**2.2 Objective of Proposed System**

Collecting the information from various research papers and other sources. We analysis that many people wants to buy plants and they have to directly concern with the nursery. Sometimes people do not know specific information about particular plant items as well as seller is not technically skilled. Customer does not compare plants prices with different shopkeeper and there is no facility for online payment only cash may consume. So, in this case e-nursery is platform where customer can compare plants pricing and make online payment easily. Customer service is extremely important. We want each customer to have a pleasant shopping experience, and it is the intention of our staff to answer questions with expertise and to offer advice when we feel it is needed. Retain customers to generate repeat purchases and make referrals. Continue to expand daily sales by adding to the variety of plants we sell. Communication with our customers through creative advertising. Customers can buy plants from their home. Customers can view a large number of plants available in a nursery. When an admin decides to check out the order, then information including the buyer’s name, address and billing instruction is record in system for future references

**2.3Users of system:**

**Nursery Owners:**

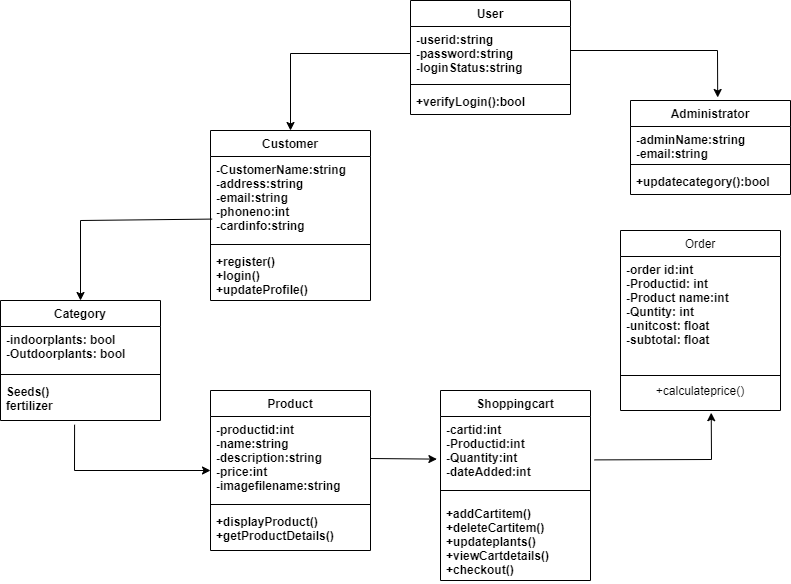
Number of shopkeeper’s can register to the web portal for increase their sale. They can provide the product which is in demand globally. Shopkeeper’s can makegood profits.

**Buyers:**

This application can be used by any user to purchase the online plants and get appropriate information by viewing short summery. If any changes to make customer can purchase the plants through different payment schemes like debit card, credit card, pat, phone

pay, cash on delivery

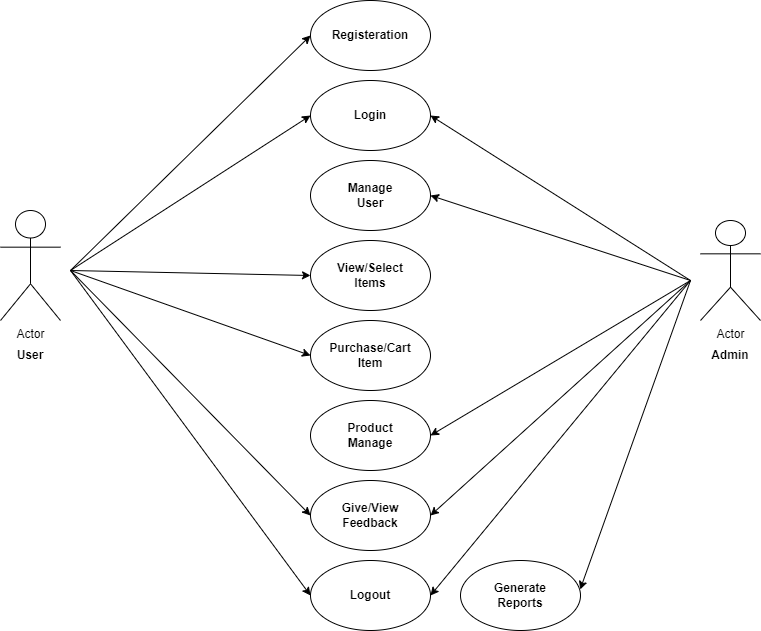
# **3. SYSTEM ANALYSIS & DESIGN:**



**• Analysis Phase**

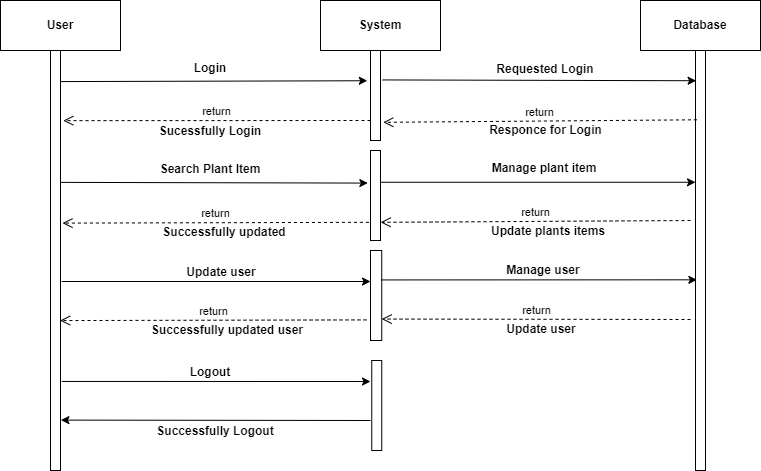
**1.1 Class Diagram:**

**1.2] Use Case Diagram:**

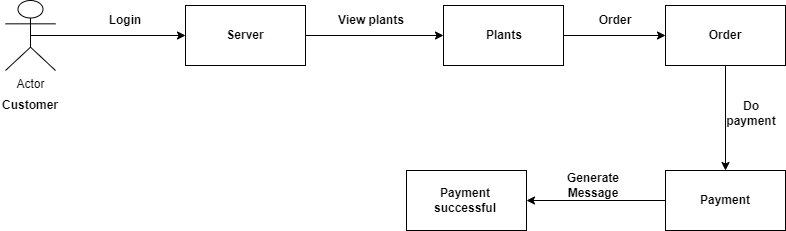


**• Design phase**

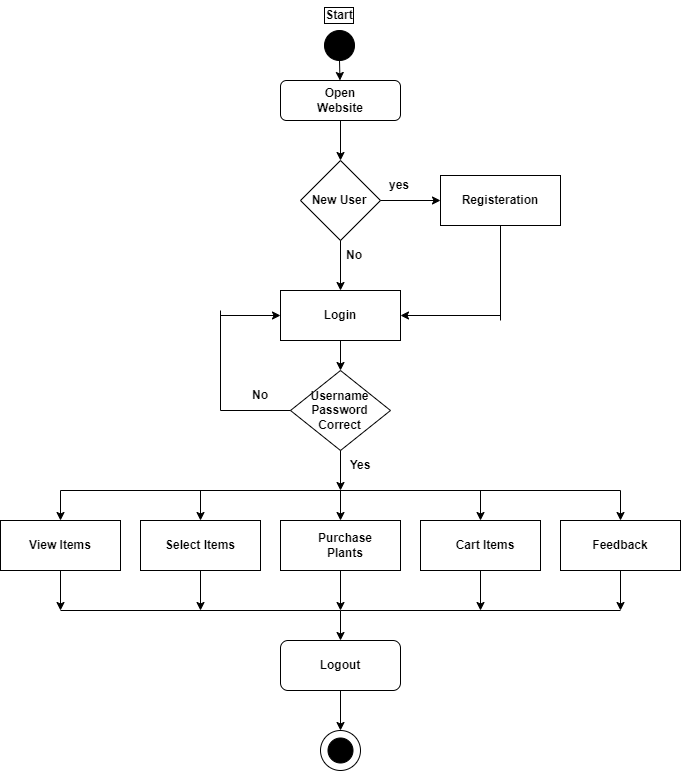
* 1. **Sequence diagram:**



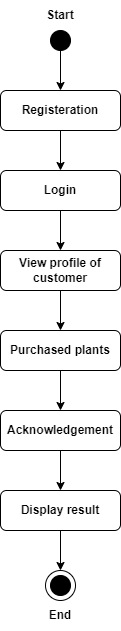
**1.4 Collaboration diagram:**



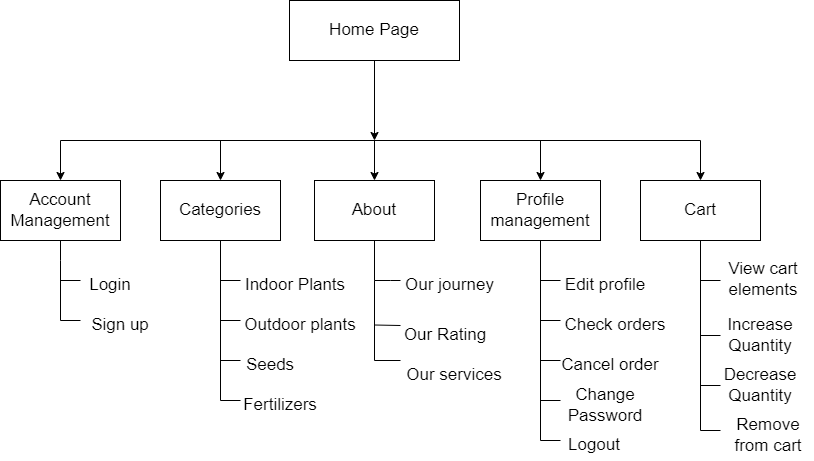
**1.5 Activity Diagram:**



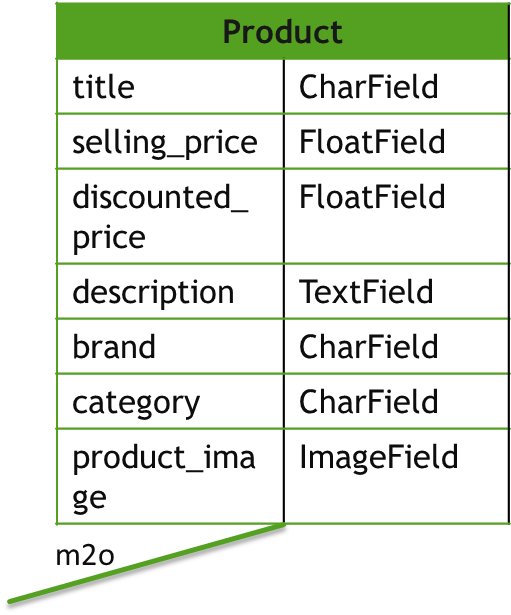
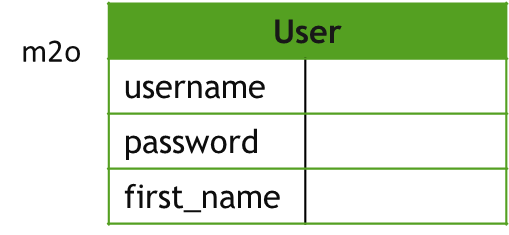
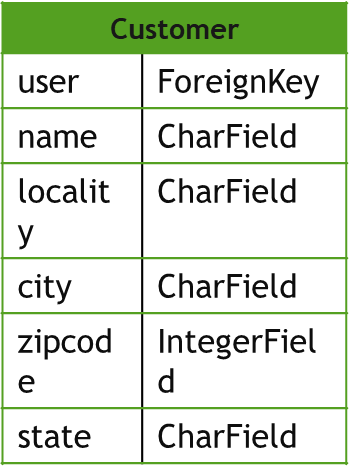
**1.6 State Chart Diagram:**



**1.7 Web site map diagram**



1.8 ERD (Entity relationship Diagram):

****

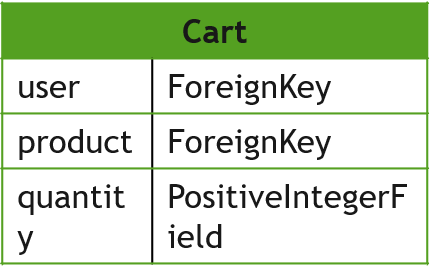
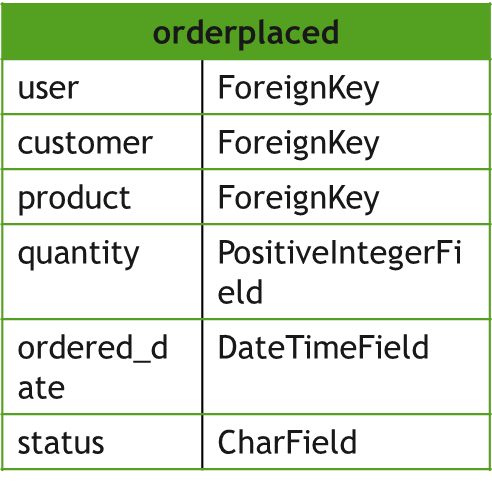
m2o

m2o

m2o

m2o

m2o

****

**3.9 Modules Specification**:

1]Registration

2]Admin

3]Products

4]Cart

5]Orders

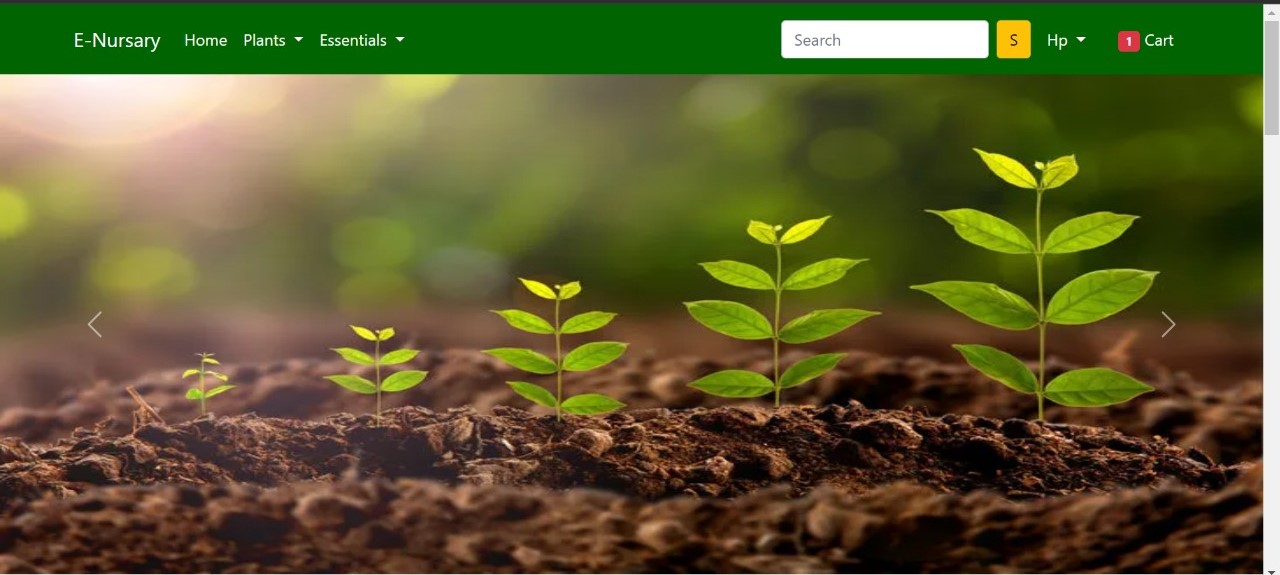
**Admin module:**

Admin first register and login himself in the website. Admin can view the order which can be added to card from the customer side. Admin can send order confirmation message to the customer and also if any plant wants to add. Admin can update the information.

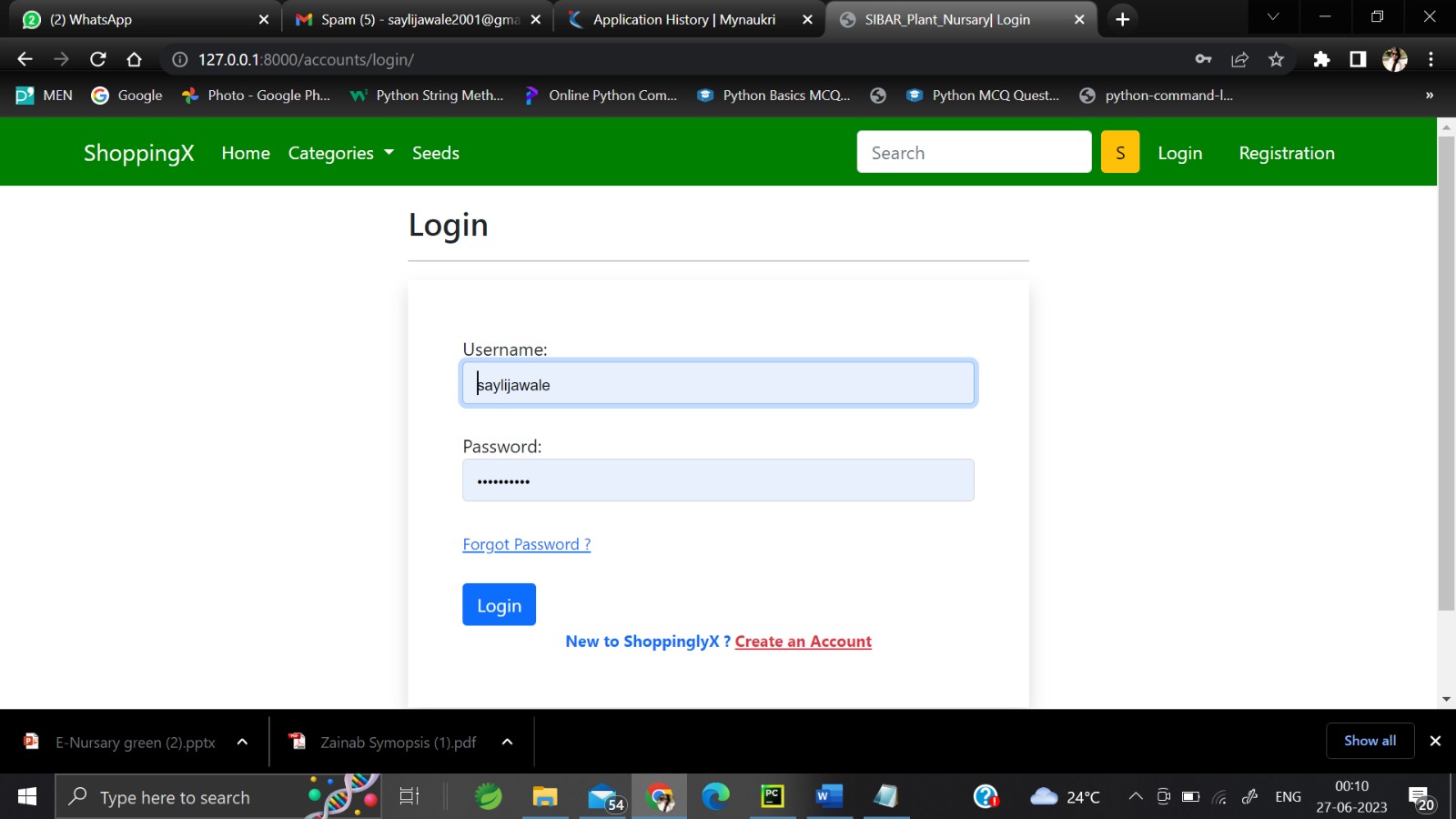
**Customer module**

Customer register and login himself in the website. Customer can select the plant item and purchase the plant by comparing prices with different shop keeper. Purchased plant details will be added to the card. Customer can pay amount through credit card, debit card, phone pay. If any complaint about the product then customer can give the feedback.

**3.10 Sample Input and Output Screens**

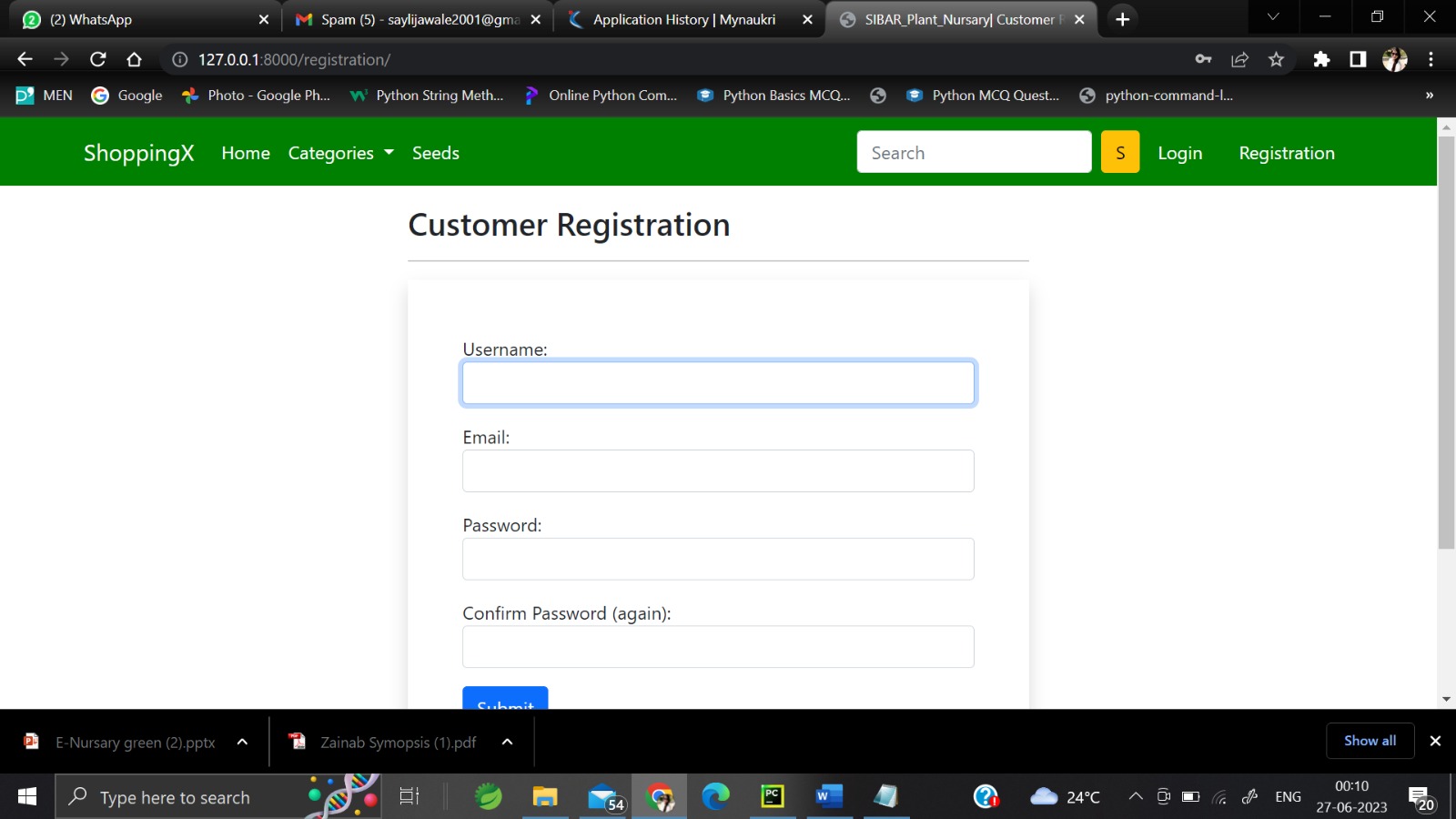


Home page:



Login page:

Registration Page



**3.10 Functional Requirement**

1. Inventory tracking
2. Check Out Flow
3. User Friendly UI
4. Mobile Responsiveness
5. Quick by feature
6. Technical details

**3.11 NON functional requirements**

1)Security

2)Storage

3)Performance

4)Attractive Design

5)Reliability

**4: CODING Sample code**

from django.shortcuts import render, redirect, HttpResponse

from .models import Customer, Product, Cart, OrderPlaced

from .forms import CustomerRegistrationForm, CustomerProfileForm

from django.views import View

from django.contrib import messages

from django.http import JsonResponse

from django.db.models import Q

from django.contrib.auth.decorators import login\_required

from django.utils.decorators import method\_decorator

class ProductView(View):

def get(self, request):

totalitem = 0

topwears = Product.objects.filter(category='TW')

bottomwears = Product.objects.filter(category='BW')

mobiles = Product.objects.filter(category='M')

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

return render(request, 'app/home.html', {'topwears':topwears, 'bottomwears':bottomwears, 'mobiles':mobiles, 'totalitem':totalitem})

class ProductDetailView(View):

def get(self, request, pk):

totalitem = 0

product = Product.objects.get(pk=pk)

print(product.id)

item\_already\_in\_cart=False

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

item\_already\_in\_cart = Cart.objects.filter(Q(product=product.id) & Q(user=request.user)).exists()

return render(request, 'app/productdetail.html', {'product':product, 'item\_already\_in\_cart':item\_already\_in\_cart, 'totalitem':totalitem})

@login\_required()

def add\_to\_cart(request):

user = request.user

item\_already\_in\_cart1 = False

product = request.GET.get('prod\_id')

item\_already\_in\_cart1 = Cart.objects.filter(Q(product=product) & Q(user=request.user)).exists()

if item\_already\_in\_cart1 == False:

product\_title = Product.objects.get(id=product)

Cart(user=user, product=product\_title).save()

messages.success(request, 'Product Added to Cart Successfully !!' )

return redirect('/cart')

else:

return redirect('/cart')

# Below Code is used to return to same page

# return redirect(request.META['HTTP\_REFERER'])

@login\_required

def show\_cart(request):

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

user = request.user

cart = Cart.objects.filter(user=user)

amount = 0.0

shipping\_amount = 70.0

totalamount=0.0

cart\_product = [p for p in Cart.objects.all() if p.user == request.user]

print(cart\_product)

if cart\_product:

for p in cart\_product:

tempamount = (p.quantity \* p.product.discounted\_price)

amount += tempamount

totalamount = amount+shipping\_amount

return render(request, 'app/addtocart.html', {'carts':cart, 'amount':amount, 'totalamount':totalamount, 'totalitem':totalitem})

else:

return render(request, 'app/emptycart.html', {'totalitem':totalitem})

else:

return render(request, 'app/emptycart.html', {'totalitem':totalitem})

def plus\_cart(request):

if request.method == 'GET':

prod\_id = request.GET['prod\_id']

c = Cart.objects.get(Q(product=prod\_id) & Q(user=request.user))

c.quantity+=1

c.save()

amount = 0.0

shipping\_amount= 70.0

cart\_product = [p for p in Cart.objects.all() if p.user == request.user]

for p in cart\_product:

tempamount = (p.quantity \* p.product.discounted\_price)

# print("Quantity", p.quantity)

# print("Selling Price", p.product.discounted\_price)

# print("Before", amount)

amount += tempamount

# print("After", amount)

# print("Total", amount)

data = {

'quantity':c.quantity,

'amount':amount,

'totalamount':amount+shipping\_amount

}

return JsonResponse(data)

else:

return HttpResponse("")

def minus\_cart(request):

if request.method == 'GET':

prod\_id = request.GET['prod\_id']

c = Cart.objects.get(Q(product=prod\_id) & Q(user=request.user))

c.quantity-=1

c.save()

amount = 0.0

shipping\_amount= 70.0

cart\_product = [p for p in Cart.objects.all() if p.user == request.user]

for p in cart\_product:

tempamount = (p.quantity \* p.product.discounted\_price)

# print("Quantity", p.quantity)

# print("Selling Price", p.product.discounted\_price)

# print("Before", amount)

amount += tempamount

# print("After", amount)

# print("Total", amount)

data = {

'quantity':c.quantity,

'amount':amount,

'totalamount':amount+shipping\_amount

}

return JsonResponse(data)

else:

return HttpResponse("")

@login\_required

def checkout(request):

user = request.user

add = Customer.objects.filter(user=user)

cart\_items = Cart.objects.filter(user=request.user)

return render(request, 'app/checkout.html', {'add':add, 'cart\_items':cart\_items})

@login\_required

def payment\_done(request):

custid = request.GET.get('custid')

print("Customer ID", custid)

user = request.user

cartid = Cart.objects.filter(user = user)

customer = Customer.objects.get(id=custid)

print(customer)

for cid in cartid:

OrderPlaced(user=user, customer=customer, product=cid.product, quantity=cid.quantity).save()

print("Order Saved")

cid.delete()

print("Cart Item Deleted")

return redirect("orders")

def remove\_cart(request):

if request.method == 'GET':

prod\_id = request.GET['prod\_id']

c = Cart.objects.get(Q(product=prod\_id) & Q(user=request.user))

c.delete()

amount = 0.0

shipping\_amount= 70.0

cart\_product = [p for p in Cart.objects.all() if p.user == request.user]

for p in cart\_product:

tempamount = (p.quantity \* p.product.discounted\_price)

# print("Quantity", p.quantity)

# print("Selling Price", p.product.discounted\_price)

# print("Before", amount)

amount += tempamount

# print("After", amount)

# print("Total", amount)

data = {

'amount':amount,

'totalamount':amount+shipping\_amount

}

return JsonResponse(data)

else:

return HttpResponse("")

@login\_required

def address(request):

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

add = Customer.objects.filter(user=request.user)

return render(request, 'app/address.html', {'add':add, 'active':'btn-primary', 'totalitem':totalitem})

@login\_required

def orders(request):

op = OrderPlaced.objects.filter(user=request.user)

return render(request, 'app/orders.html', {'order\_placed':op})

#changed mobile to indoor

def indoor(request, data=None):

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

if data==None :

indoor\_plants = Product.objects.filter(category='M')

elif data == 'greeny' :

indoor\_plants = Product.objects.filter(category='M').filter(brand=data)

elif data == 'monsoon':

indoor\_plants = Product.objects.filter(category='M').filter(brand=data)

elif data == 'below':

indoor\_plants = Product.objects.filter(category='M').filter(discounted\_price\_\_lt=500)

elif data == 'above':

indoor\_plants = Product.objects.filter(category='M').filter(discounted\_price\_\_gt=1000)

return render(request, 'app/indoor.html', {'indoor\_plants':indoor\_plants, 'totalitem':totalitem})

#function to filter outdoor data

def outdoor(request, data=None):

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

if data==None :

outdoor\_plants = Product.objects.filter(category='BW')

elif data == 'greeny' :

outdoor\_plants = Product.objects.filter(category='BW').filter(brand=data)

elif data == 'monsoon':

outdoor\_plants = Product.objects.filter(category='BW').filter(brand=data)

elif data == 'below':

outdoor\_plants = Product.objects.filter(category='BW').filter(discounted\_price\_\_lt=500)

elif data == 'above':

outdoor\_plants = Product.objects.filter(category='BW').filter(discounted\_price\_\_gt=1000)

return render(request, 'app/outdoor.html', {'outdoor\_plants':outdoor\_plants, 'totalitem':totalitem})

#function for medicinal plants

def medicinal(request, data=None):

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

if data==None :

medicinal\_plants = Product.objects.filter(category='TW')

elif data == 'greeny' :

medicinal\_plants = Product.objects.filter(category='TW').filter(brand=data)

elif data == 'monsoon':

medicinal\_plants = Product.objects.filter(category='TW').filter(brand=data)

elif data == 'below':

medicinal\_plants = Product.objects.filter(category='TW').filter(discounted\_price\_\_lt=500)

elif data == 'above':

medicinal\_plants = Product.objects.filter(category='TW').filter(discounted\_price\_\_gt=1000)

return render(request, 'app/medicinal.html', {'medicinal\_plants ':medicinal\_plants , 'totalitem':totalitem})

# function for seeds

def seeds(request, data=None):

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

if data==None :

seeds\_c = Product.objects.filter(category='S')

elif data == 'greeny' :

seeds\_c = Product.objects.filter(category='S').filter(brand=data)

elif data == 'monsoon':

seeds\_c = Product.objects.filter(category='S').filter(brand=data)

elif data == 'below':

seeds\_c = Product.objects.filter(category='S').filter(discounted\_price\_\_lt=500)

elif data == 'above':

seeds\_c = Product.objects.filter(category='S').filter(discounted\_price\_\_gt=1000)

return render(request, 'app/seeds.html', {'seeds\_c ':seeds\_c , 'totalitem':totalitem})

class CustomerRegistrationView(View):

def get(self, request):

form = CustomerRegistrationForm()

return render(request, 'app/customerregistration.html', {'form':form})

def post(self, request):

form = CustomerRegistrationForm(request.POST)

if form.is\_valid():

messages.success(request, 'Congratulations!! Registered Successfully.')

form.save()

return render(request, 'app/customerregistration.html', {'form':form})

@method\_decorator(login\_required, name='dispatch')

class ProfileView(View):

def get(self, request):

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

form = CustomerProfileForm()

return render(request, 'app/profile.html', {'form':form, 'active':'btn-primary', 'totalitem':totalitem})

def post(self, request):

totalitem = 0

if request.user.is\_authenticated:

totalitem = len(Cart.objects.filter(user=request.user))

form = CustomerProfileForm(request.POST)

if form.is\_valid():

usr = request.user

name = form.cleaned\_data['name']

locality = form.cleaned\_data['locality']

city = form.cleaned\_data['city']

state = form.cleaned\_data['state']

zipcode = form.cleaned\_data['zipcode']

reg = Customer(user=usr, name=name, locality=locality, city=city, state=state, zipcode=zipcode)

reg.save()

messages.success(request, 'Congratulations!! Profile Updated Successfully.')

return render(request, 'app/profile.html', {'form':form, 'active':'btn-primary', 'totalitem':totalitem})

**5. Limitations of project**

No Instruction or guidance of how to manage the plant.

-No community available for open discussion

-Use of google API such as Google Lens , Google Map and Google News ineffectively

as function of application.

-Plant reference could not be visualized method.

-No introduction for new user of overall application description.

**6. CONCLUSION:**

The proposed system can guarantee to keep the records are safe and privacy which is stored in the database. It converts unstructured data

into structured data and sorted format. It is very helpful, reliable and performs well functional to get an alert message and emails on the cell

phone.

1. In this dissertation, we have developed an approach to allow customers to buy plants without even visiting shop.

2. Being able to buy anytime, anywhere, any place.

3. Site enables them to browse before they shop, and to research the product so they have more confidence in what they are buying.

4. Online shopping becomes more enjoyable and easier than real- world shopping.

5. It provides online payment system.

6. Customer can track their order detail and give the feedback if any problem occur during shipment

**7 Reference**

1. Krishnan, P.R., Kaila, R.K., Mewari, J.C. and Roy, M.M. (2014) Plant Nursery Management and Plant Nursery Management:

Principles and Practices, Central Arid

2. Kumar. N., (1997) Introduction to Horticulture. Raja Lakshmi Publications, 28/5 – 693,Vepamoodu Junction, Nagercoil. Pp.: 15.47-

15.50.

3. Landis, T.D., Tinos, R.W., McDonald, S.E., and Barnett, J.P. (1994) Nursery Planning, Development and Management. Vol. 1, the

container tree nursery manual. Agriculture Handbook 674. Washington, DC, USA: US Department

4. Www. The free dictionary -com/business. copyright (c)2011 Retrieved 2011-09-15

5. Nestor, O.G., John, H. and Steve, H. (undated) The Operational Effectiveness of The Forest Nursery Sector in Leyte, The

Philippines. Improving the Triple Bottom Line Return from Small Scale Forestry. Pp. 155-165

6. Xiasong Hu ,Wen Zhu,“HCRS, A Hybrid

Clothes Recommender System Based On

User,” in proceedings of the ratings and plant

features, International Conference on

Management of e-commerce and e-

government,2020.

7. Xiang Liu, “Fashion Evaluation Method for

Clothing Recommendation Based on Weak

Appearance Feature”,2017.