

A
PROJECT REPORT
ON
“E-FARMING: SHOPPING SYSTEM”
Towards partial fulfilment of the requirement in
5th Semester BCA (2022-23)

Submitted by:

Shree Koshti (200510101061)

Jay Gandhi (200510101034)

Purvesh Bavashiya (200510101011)

Submitted To:-



**Parul Institute of Computer Application,
Parul University**

Under the guidance of

Assistant Professor, Manoj Kamber

Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank Dr Priya Swaminarayan, Dean, FITCS for providing us an opportunity to do the project work in BCA and giving us all support and guidance, which made us complete the project duly. We are extremely thankful to Mam for providing her support and guidance, although she had busy schedule managing the academic affairs.

We would not forget to remember Prof. Hina Chokshi, HOD, BCA department for her encouragement and more over for her timely support and guidance till the completion of our project work.

We owe our deep gratitude to our project guide Prof. Manoj Kamber, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from our Parents, all Teaching staffs of BCA Department which helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in laboratory for their timely support.

[Shree Koshti] [200510101061]

[Jay Gandhi] [200510101034]

[Purvesh Bavashiya] [200510101011]



PARUL INSTITUTE OF COMPUTER APPLICATION

CERTIFICATE

This is to certify that SHREE KOSHTI, JAY GANDHI, PURVESH BAVASHIYA the student(s) of Parul Institute of Computer Application, has/have satisfactorily completed the project entitled "E-FARMING: SHOPPING SYSTEM" as a part of course curriculum in BCA / IMCA semester-V for the academic year 2022-2023 under guidance of Asst. Prof. MANOJ KAMBER.

Enrollment Number: 200510101061

Enrollment Number: 200510101034

Enrollment Number: 200510101011

Quality of work	Grade	Sign of Internal guide
Poor / Average / Good / Excellent	B /B+ / A / A+	

Date of submission:

HOD,

Prof. Hina Chokshi

Principal,

Dr Priya Swaminarayan

INDEX

1. Research.....	1
2. Feasibility Studies.....	2
3. System Requirement Specification	4
4. Technology Description	8
5. Data Flow Diagram.....	9
6. Use Case Diagram	17
7. Class Diagram.....	18
8. Activity Diagram.....	19
9. E-R Diagram.....	21
10. Data Dictionary.....	22
11. Form Design	34
12. What is testing?.....	46
13. Future Enhancement.....	48
14. References & Bibliography.....	49

INDEX OF FIGURES

Figure 1: Time Line Chart	7
Figure 2: Context Level DFD: 0 Level.....	9
Figure 3: User Level 1	10
Figure 4: Admin Level 1	11
Figure 5: User Level 2: All Products	12
Figure 6: Admin Level 2: Order List	12
Figure 7: Admin Level 2: Manage Payments	13
Figure 8: Admin Level 2: CMS Web Content	13
Figure 9: Admin Level 2: Manage User	14
Figure 10: User Level 2: Authentication	15
Figure 11: User Level 2: All Products	15
Figure 12: User Level 2: My Cart.....	16
Figure 13: User Level 2: Checkout.....	16
Figure 14: Use Case.....	17
Figure 15: Class Diagram	18
Figure 16: Activity Diagram: User	19
Figure 17: Activity Diagram: Admin.....	20
Figure 18: ER Diagram.....	21
Figure 19: Add to Cart Functionality.....	34
Figure 20: Product Page	34
Figure 21: Cart Page	35
Figure 22: Cart page Code.....	35
Figure 23 : Checkout Page.....	36
Figure 24 : Checkout Page Code	36
Figure 25 : Home Page - 1.....	37
Figure 26 : Home Page – 2	37
Figure 27 : Home Page Code.....	38
Figure 28 : My Account Design	38
Figure 29 : My Account Code	39
Figure 30 : Admin SideBar Design.....	40
Figure 31 : Admin Sidebar Code	40
Figure 32 : Products Page – Admin	41
Figure 33 : Product Page Code - Admin.....	41
Figure 34 : Paytm Payment Design	42
Figure 35 : Paytm Payment Code	42
Figure 36 : Order Confirm Email Template Design	43
Figure 37 : Order Confirm Email Template Code	43
Figure 38 : Change password Validations Page	44
Figure 39 : Edit Profile page Validations	44
Figure 40 : Login Page Validations	45

1. Research

1.1.What is research?

Research is defined as careful consideration of study regarding a particular concern or problem using scientific methods. It is a systematic inquiry to describe, explain, predict and control the observed phenomenon. It involves inductive and deductive methods.

1.2.Types of Research Methodology

Research methods are broadly classified as Qualitative and Quantitative. Both methods have distinctive properties and data collection methods.

1.2.1. Qualitative Methods

This is a type of research method in which we collect data using face-to-face conversation, usually with open-ended questions. Using this method, we came to know our lack of requirements in the eFarm: shopping system. These types methods include:

- One-to-one interview

At this stage, we collected information from selected candidates about our system UI and basic requirements.

- Focus Groups

At this stage, we took all the same selected candidates for group discussion. The primary focus of this discussion was gathering ideas from all of the candidates for our eFarm: shopping system.

- Ethnographic studies
- Text Analysis
- Case Study

1.2.2. Quantitative Methods

This is a type of research method in which we take very basic data from a group of samples in numbers and measurable forms. It uses a systematic way of investigating events or data. It answers questions to justify relationships with measurable variables to either explain, predict or control a phenomenon. Types of quantitative methods include:

- Survey research

At this stage, we conducted a survey to know the basic requirements for our system; those requirements which are countable or non-additional (here we used MCQ questions for the survey).

- Descriptive research

At this stage, we took a random sample of the population and got feedback from them about our system. In addition, we asked them about their ideas for our system.

2. Feasibility Studies

2.1. What is Feasibility?

As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable.

A well-designed study should offer a historical background of the business or project, such as a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such studies precede technical development and project implementation.

2.2. Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building—currently, this project is not technically feasible.

2.3. Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

2.4. Operational Feasibility

Determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

2.5. Importance of Feasibility Studies

A feasibility study is conducted to determine the success and minimize the risks related to the project. It is not merely project research, but a framework or a plan on how to establish and run a business successfully in the long run. It contains five essential components, including market research, financial research, management research, schedule determination, and technical research.

2.6. Feasibility Study of our Proposed System

2.6.1. Technical Feasibility:

- In this proposed system, technical feasibility depends on open-source tools and technologies.
- In this system technologies like HTML, CSS, JavaScript, Python (Django Framework), Postgres, Bootstrap used.

2.6.2. Economical Feasibility:

- Development Costs:

The system is economically feasible as its costs nothing because all depend on open source.

- Production Costs:

Hosting cost, operation, and maintenance cost including software and hardware upgrading.

2.6.3. Operational Feasibility:

- The main purpose of the proposed system is that it will solve the task of farmers by finding their framing products.
- Farmers will be able to find the needed product easily with our user-friendly UI.
- This system manages all the users and their orders.

3. System Requirement Specification

3.1. Introduction To SRS

3.1.1. What is SRS?

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements and may include a set of use cases that describe user interactions that the software must provide.

3.1.2. Need of SRS

To fully understand one's project, they must come up with an SRS listing out their requirements, how are they going to meet it and how will they complete the project. It helps the team to save upon their time as they are able to comprehend how are going to go about the project. Doing this also enables the team to find out about the limitations and risks early on.

3.2. Abstract

Our group has made a project on E-FARMING: SHOPPING SYSTEM. We have developed it by using python Django framework. The system is implemented to reduce the manual work and enhances the accuracy of work in a farming. This system manages and maintains the record of customers and their order online. This project has been made in a user-friendly interface. So that Customer can add and delete the products easily. Through the place ordering products, the customer can simply click and order the products. This system entirely reduces the unnecessary time of farmer.

Looking towards the problems faced by farmers, i.e., unavailability of good quality seeds, lack of modern equipment and machinery, poor irrigation facilities, and so on, our website will make them available with all of the products, which results in fewer conflicts

A farmer can purchase seeds, tools, fertilizer etc., so this can help the farmers by reducing their time of going out for getting products.

3.2. System Users

3.2.1. Admin

An admin or manager is the person who will manage the entire web site. He can manipulate web content, being able to see the number of users as well as payments made by any particular user. He can edit the product information, such as its price, items available, etc.

3.2.2. User

Users will be able to buy products, apply coupons, make payments, and view their profile.

3.3. Modules

- Authentication Module
- User Account Module
- Products Module
- Add to Cart Module
- Order Module
- Payment Module

3.4. Modules Description

3.4.1. Authentication Module

It is a vital feature of our site. Because without a login, placing an order and making a payment is not possible. And the other reason for logging in is to avoid (to some extent) placing the fake payments.

3.4.2. User Account Module

A user's user profile is also known as their mini-profile page. This will show all the details, like their ordered item list.

3.4.3. Products Module

This module will take care of all product details.

3.4.4. Add to cart Module

This module will take care of the user's selected products and show them the total cart price.

3.4.5. Order Module

The Order module is useful for the admin to maintain all the orders of the user.

3.4.6. Payment Module

For the payment, we are going to use the Paytm API. This will be managed under the payment module.

3.5. Hardware Requirements (Developer)

Name of Components	Specification
Processor	Intel Core i3/i5
RAM	8GB/12GB
Hard Disk	512GB/1TB

Table 1: Developer: Hardware Requirements

3.6. Software Requirements (Developer)

Name of Components	Specification
Operating System	Windows 8,9,10
Software development Kit	Python
Tools & languages	Python (Django Framework), HTML, CSS, JavaScript, Bootstrap, Postgres

Table 2: Developer: Software Requirements

3.7. Hardware Requirements (User)

Name of Components	Specification
Processor	Intel Core i3/i5, A15 Bionic Apple, Snapdragon 7
RAM	4GB
Hard Disk	1 GB

Table 3: User: Hardware Requirements

3.8. Software Requirements (User)

Name of Components	Specification
Operating System	Windows 8,9,10, Android, IOS, MAC
Basic Need/ Requirements	Internet Access
Tools	Chrome, MS Edge, Safari Etc.

Table 4: User: Software Requirements

3.9. Time Line Chart

Development phase	75 Days						Duration N (days)
	0to10 days	11to20 days	21to30 days	31to40 days	41to50 days	51to75 days	
Requirement Gathering	■						07
Analysis	■	■					09
Design		■	■				10
Development Phase 1			■				13
Development Phase 2				■			13
Development Phase 3					■		13
Documentation						■	10
Total time (Days)							75

Figure 1: Time Line Chart

4. Technology Description

In the development of this proposed system Python (Django) is used. Python (Django) is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. Python (Django) is a widely-used, free, and efficient alternative to competitors such as Flask. And other technology used are Python (Django framework), HTML, CSS, JavaScript and Postgres.

4.1. Features and Limitations of New System

Existing System	New System
Lack of payment options	New payment options added like UPI
Lack of login Options	New login options added like Google.
Admin page is not flexible	Admin page is flexible
User/User mail verification feature is not available.	User/User mail verification feature is available at the time of registration.

Table 5 : Comparison Between Existing System and New System

5. Data Flow Diagram

5.1. Context Level DFD's

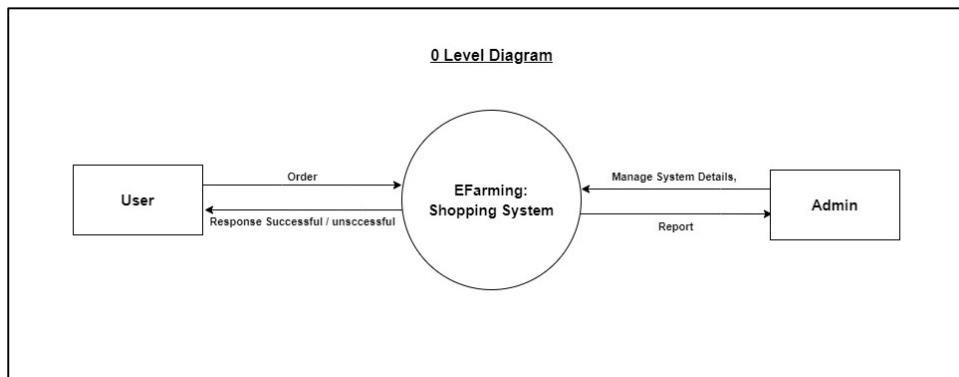


Figure 2: Context Level DFD: 0 Level

Description: DFD 0: this above diagram describe our system main activity or process.

5.2. Level 1 DFD's:

➤ User

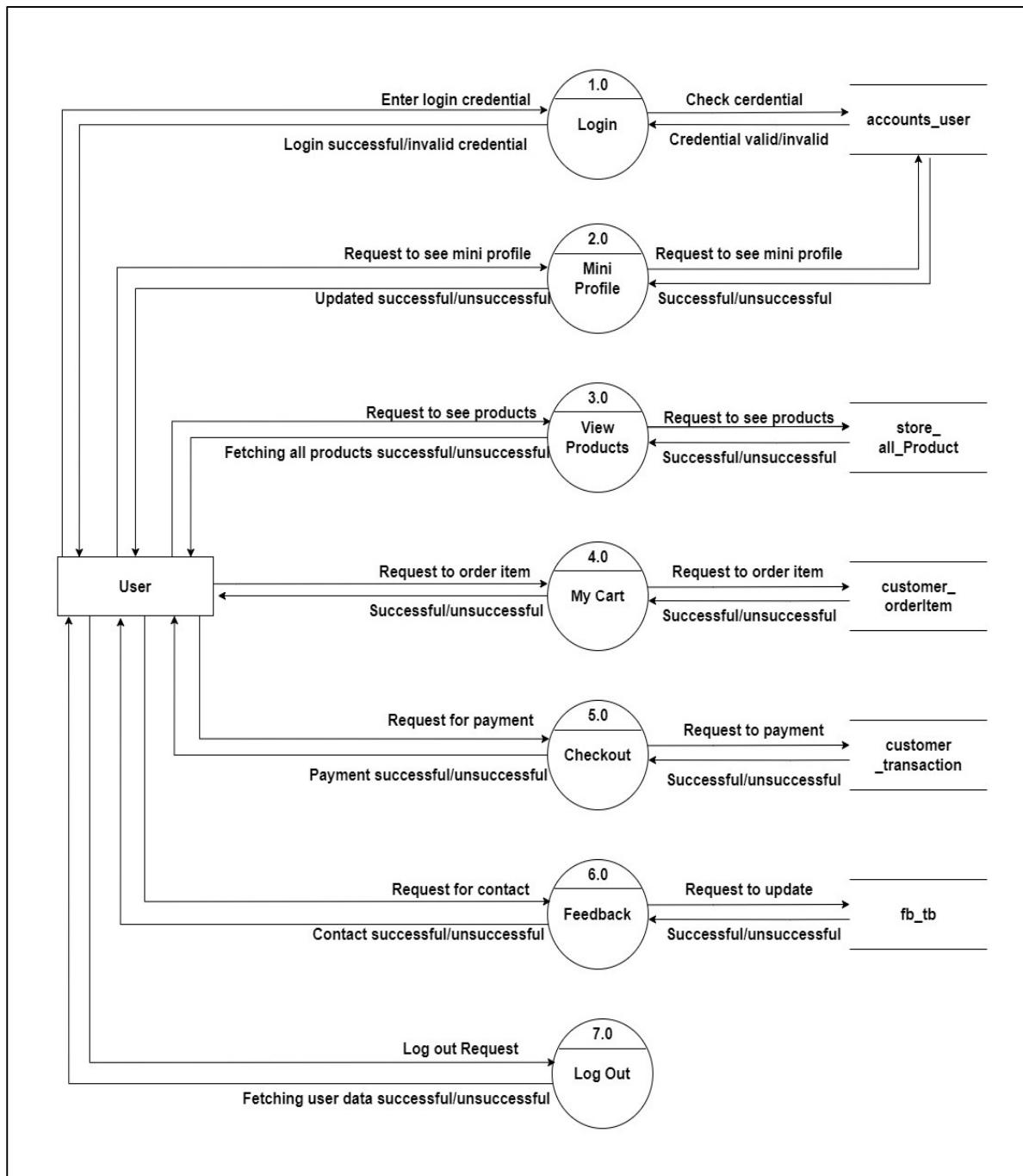


Figure 3: User Level 1

Description: DFD 1: this above diagram describes all the process that user can do.

➤ Admin :

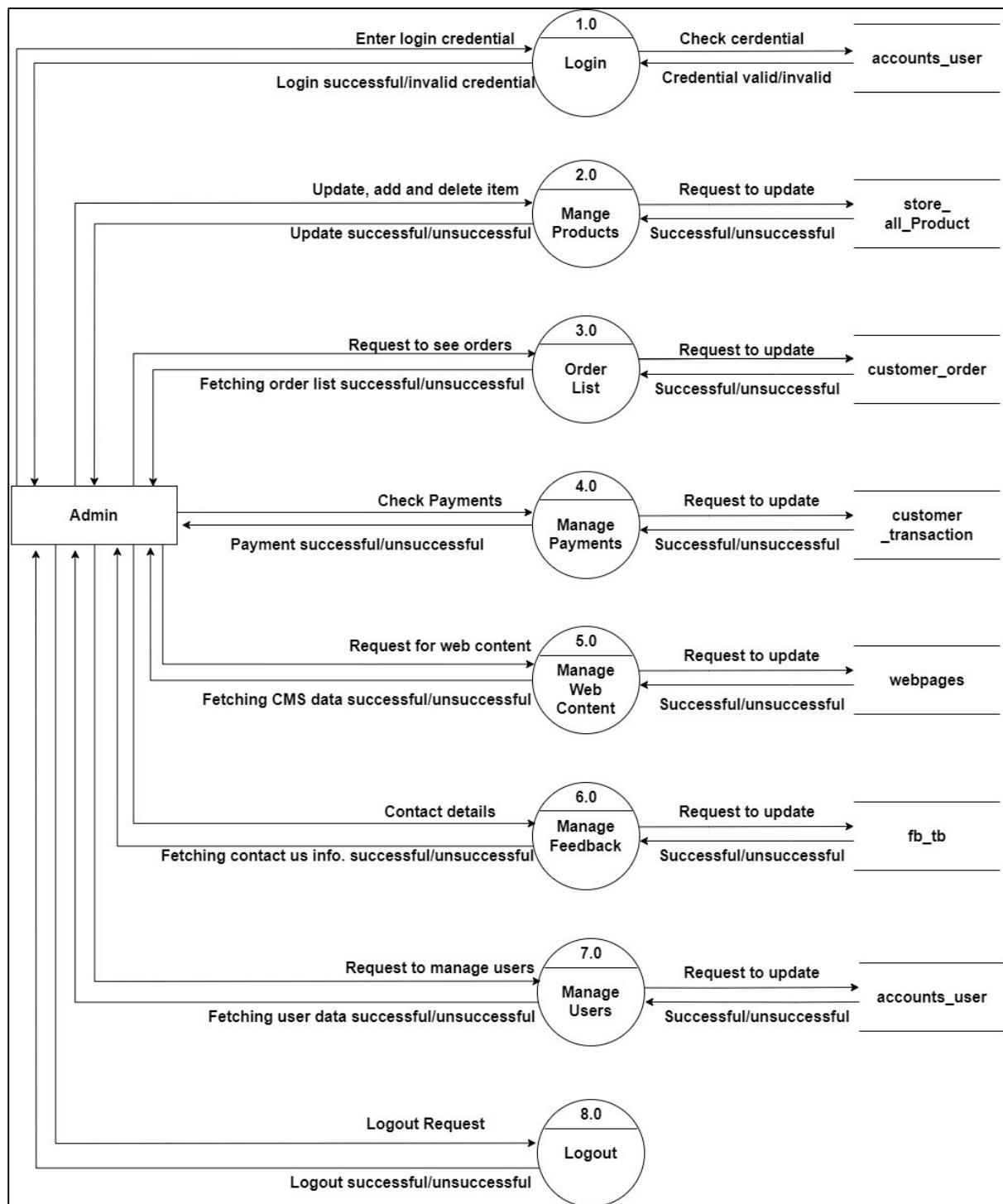


Figure 4: Admin Level 1

Description: DFD 1: this above diagram describes all the process that admin can do.

5.3. Level 2 DFD's

➤ Admin

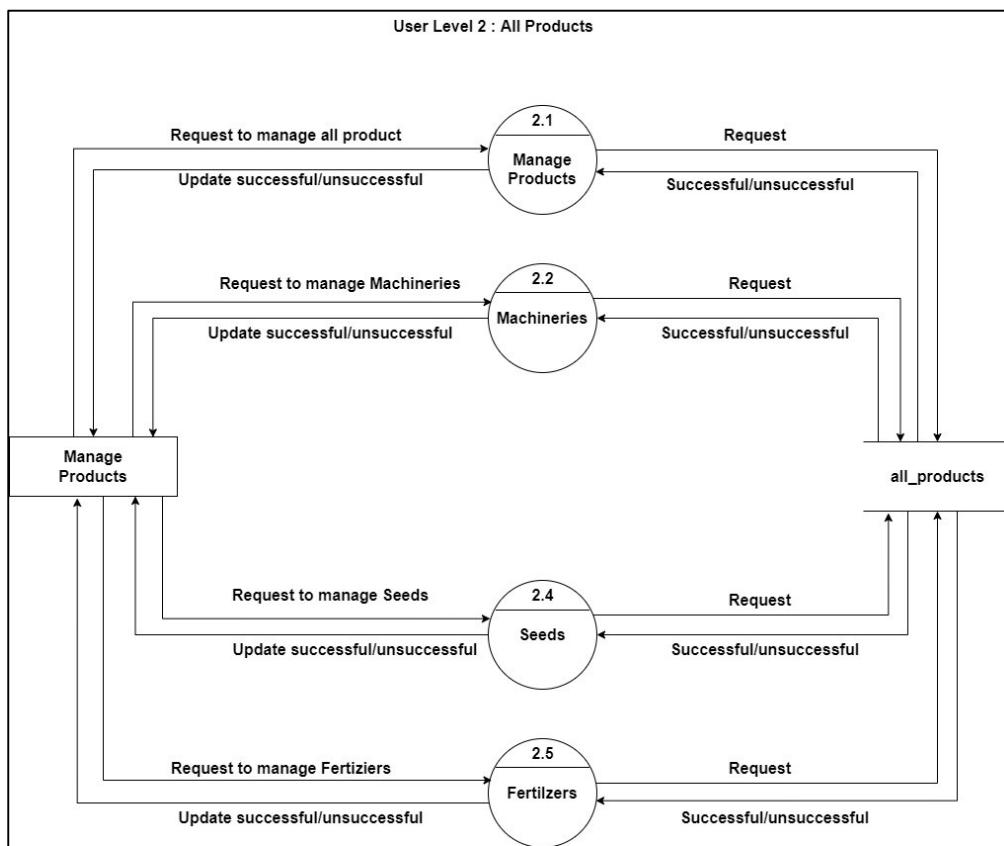


Figure 5: User Level 2: All Products

Description: DFD 2: this above diagram describes more in All products, shows us the subprocesses

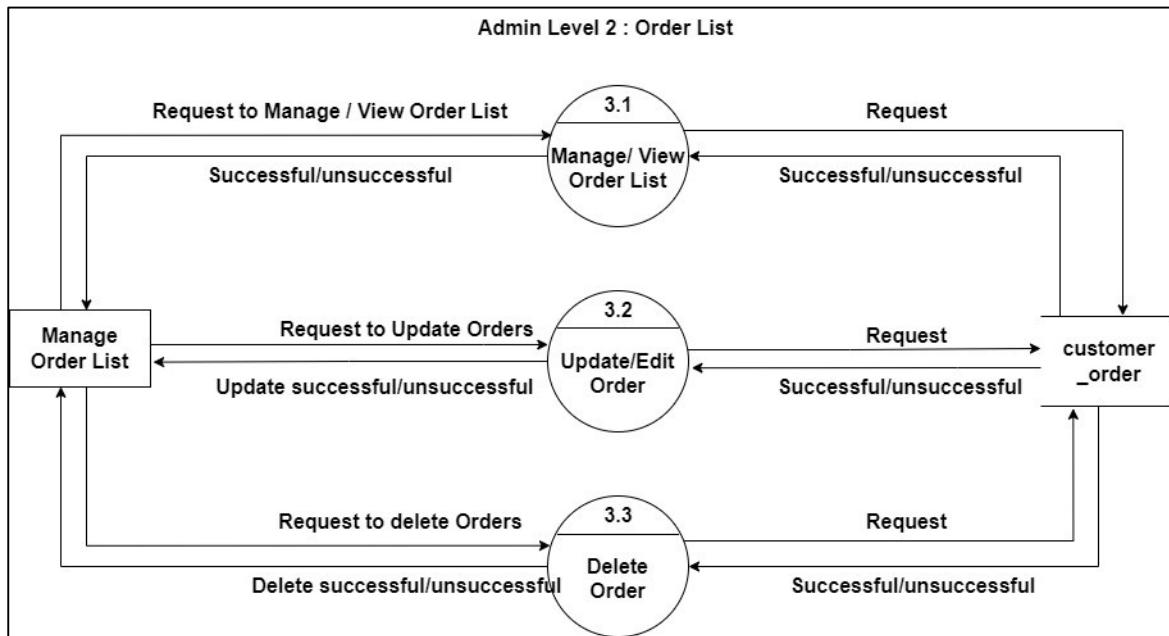


Figure 6: Admin Level 2: Order List

Description: DFD 2: this above diagram describes more in order list, shows us the subprocesses.

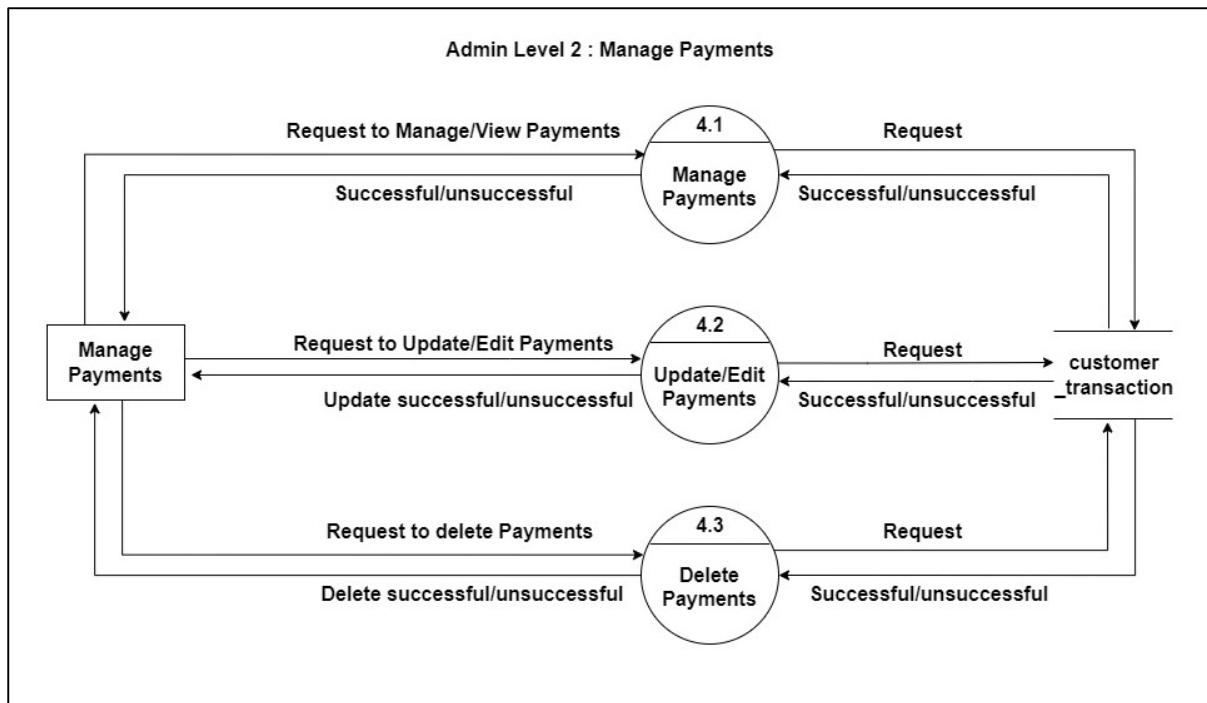


Figure 7: Admin Level 2: Manage Payments

Description: DFD 2: this above diagram describes more in manage payment, shows us the subprocesses.

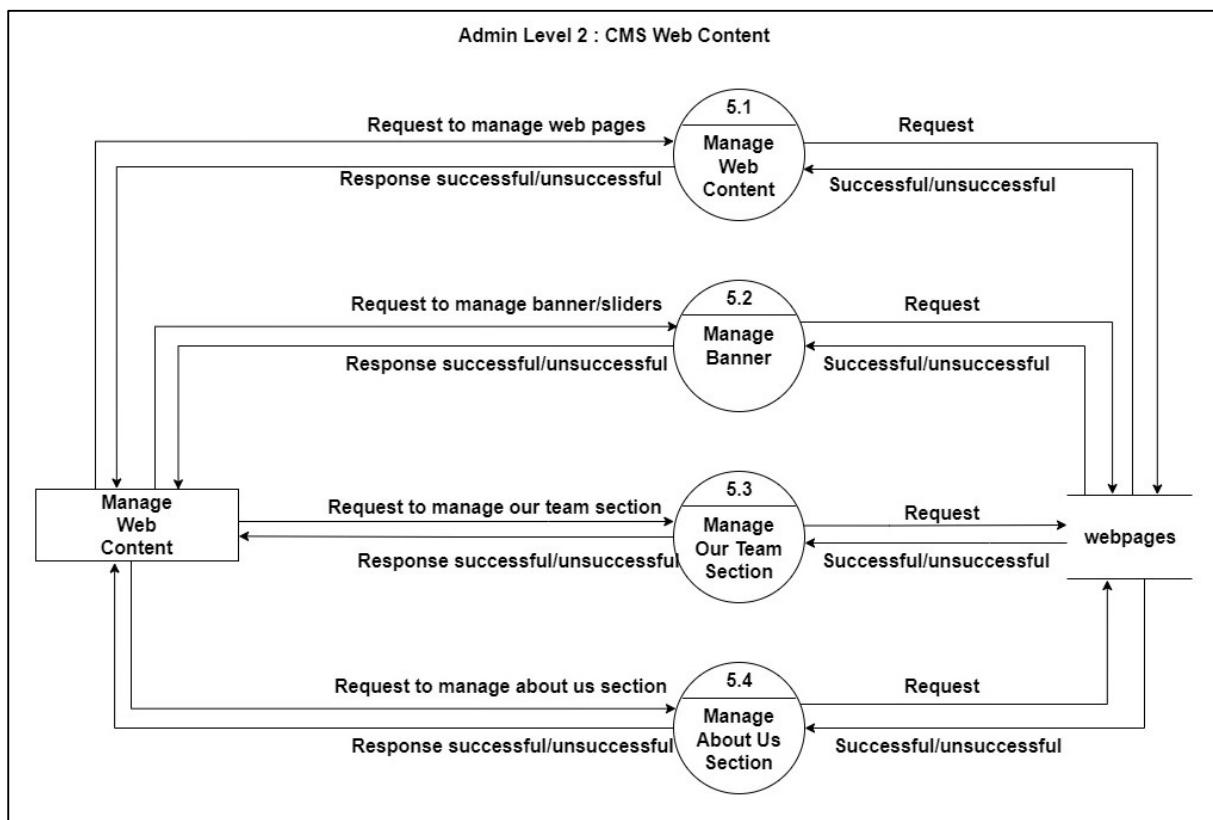


Figure 8: Admin Level 2: CMS Web Content

Description: DFD 2: this above diagram describes more in CMS Web Content, shows us the subprocesses

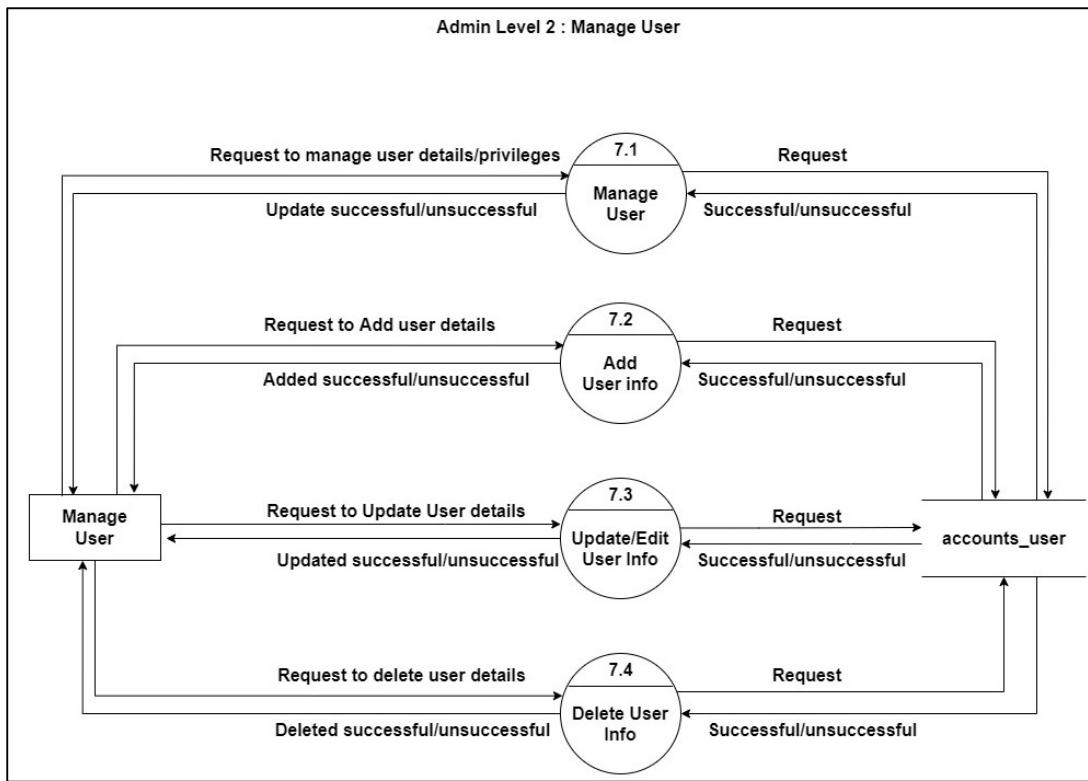


Figure 9: Admin Level 2: Manage User

Description: DFD 2: this above diagram describes more in Manage user, shows us the subprocesses.

➤ User

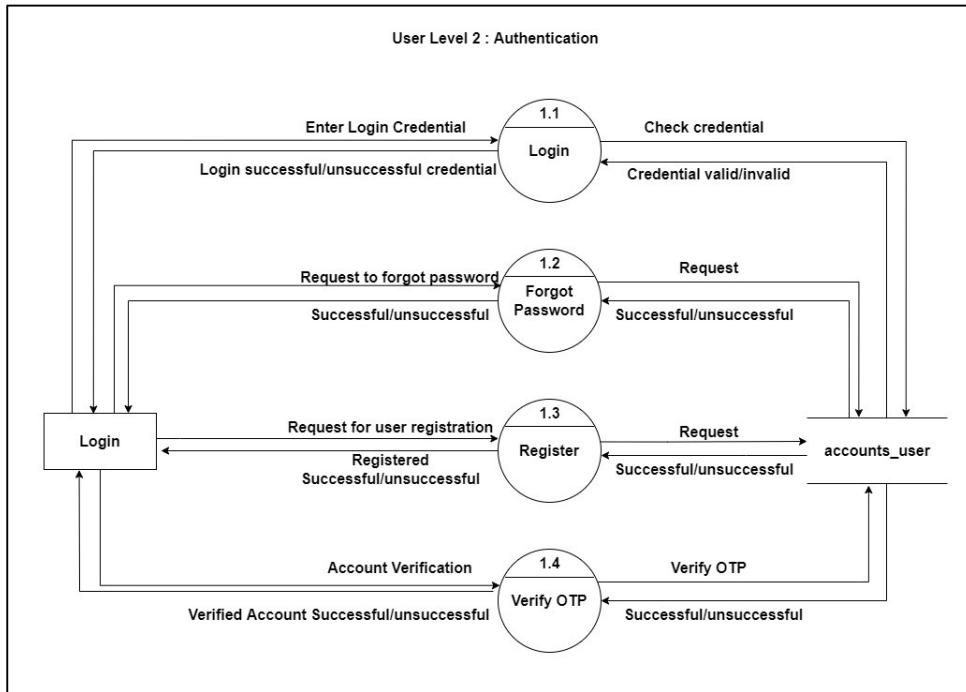


Figure 10: User Level 2: Authentication

Description: DFD 2: this above diagram describes more in Authentication, shows us the subprocesses.

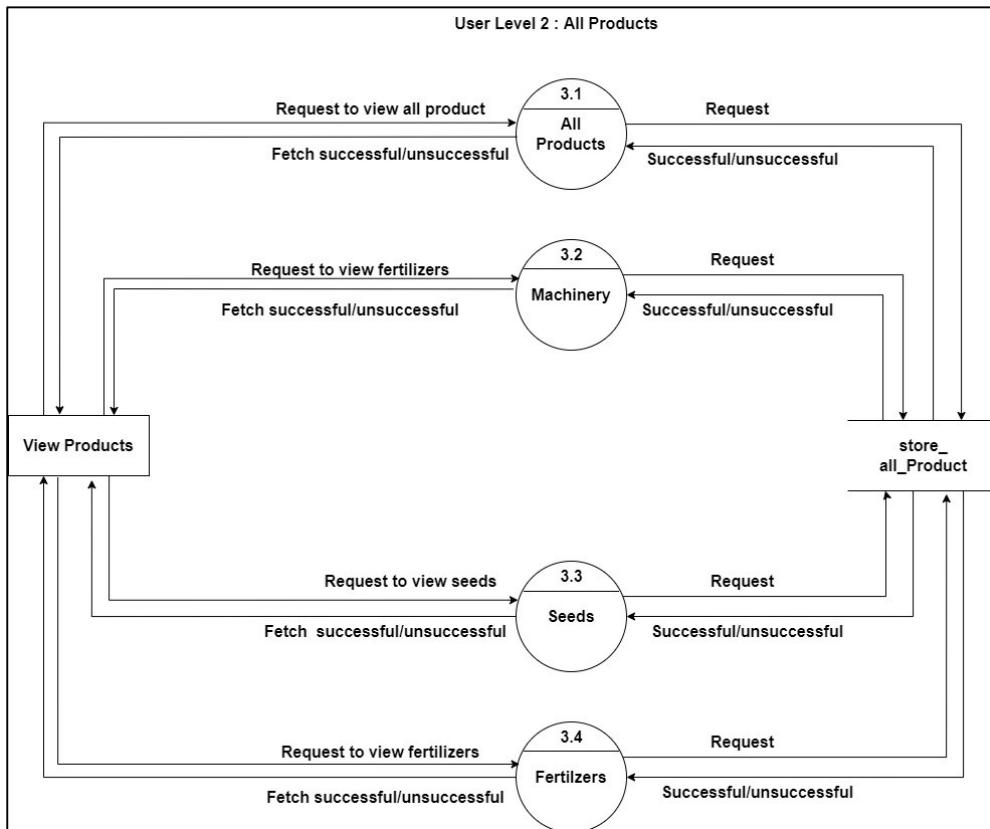


Figure 11: User Level 2: All Products

Description: DFD 2: this above diagram describes more in All products, shows us the subprocesses

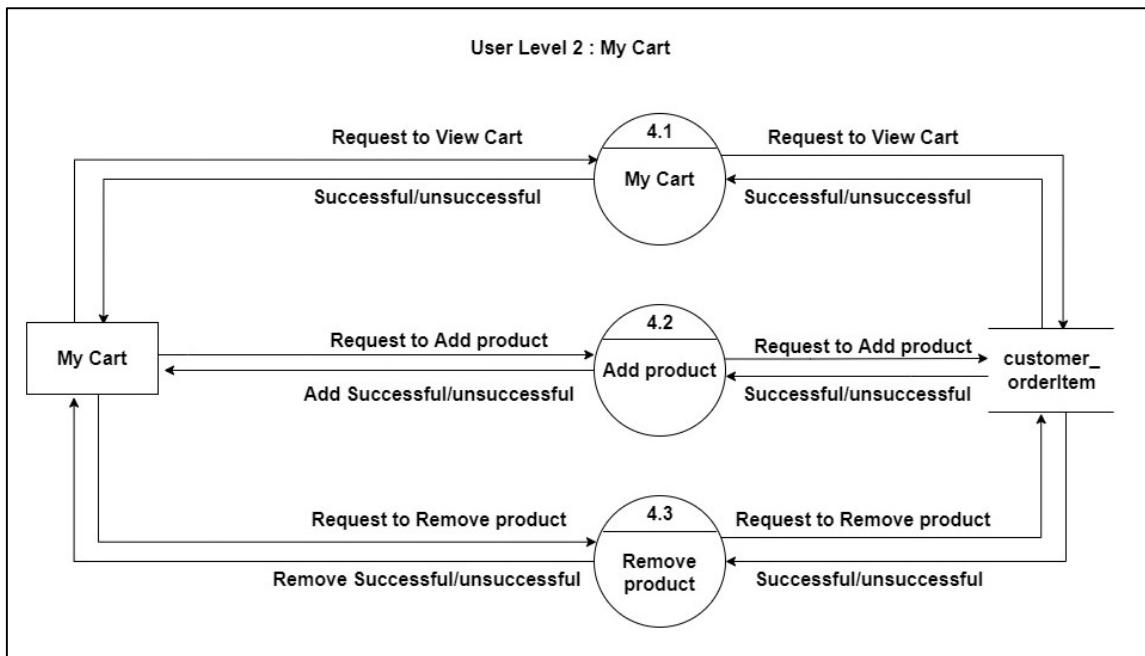


Figure 12: User Level 2: My Cart

Description: DFD 2: this above diagram describes more in My cart, shows us the subprocesses.

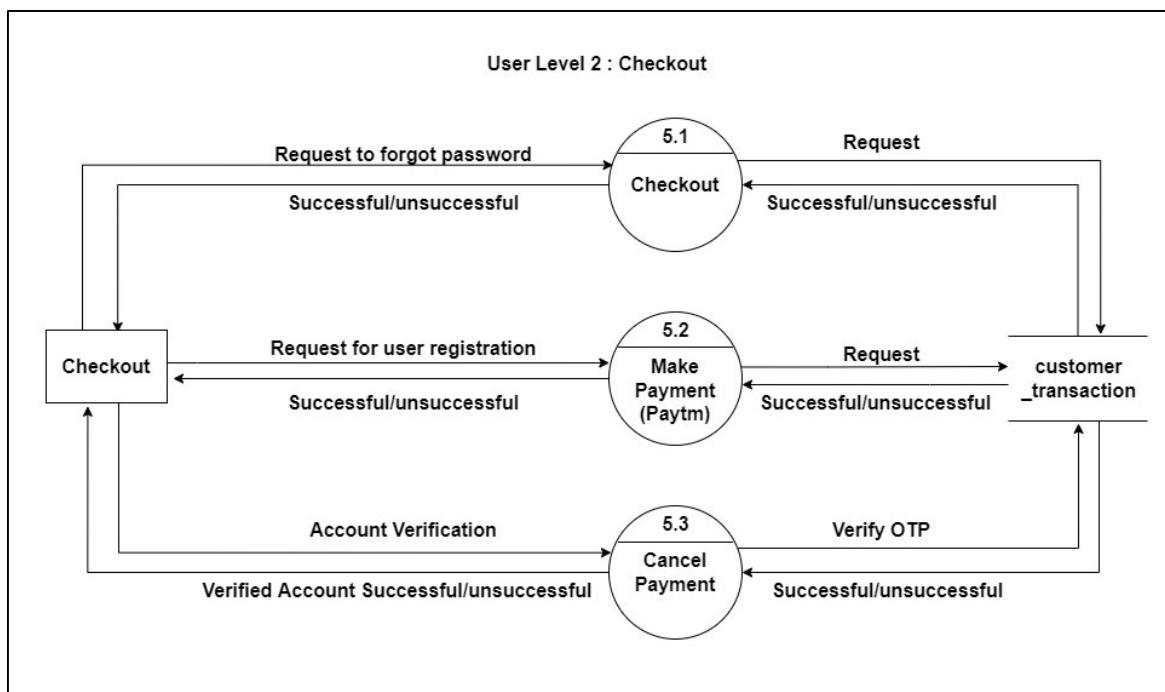


Figure 13: User Level 2: Checkout

Description: DFD 2: this above diagram describes more in Checkout, shows us the subprocesses.

6. Use Case Diagram

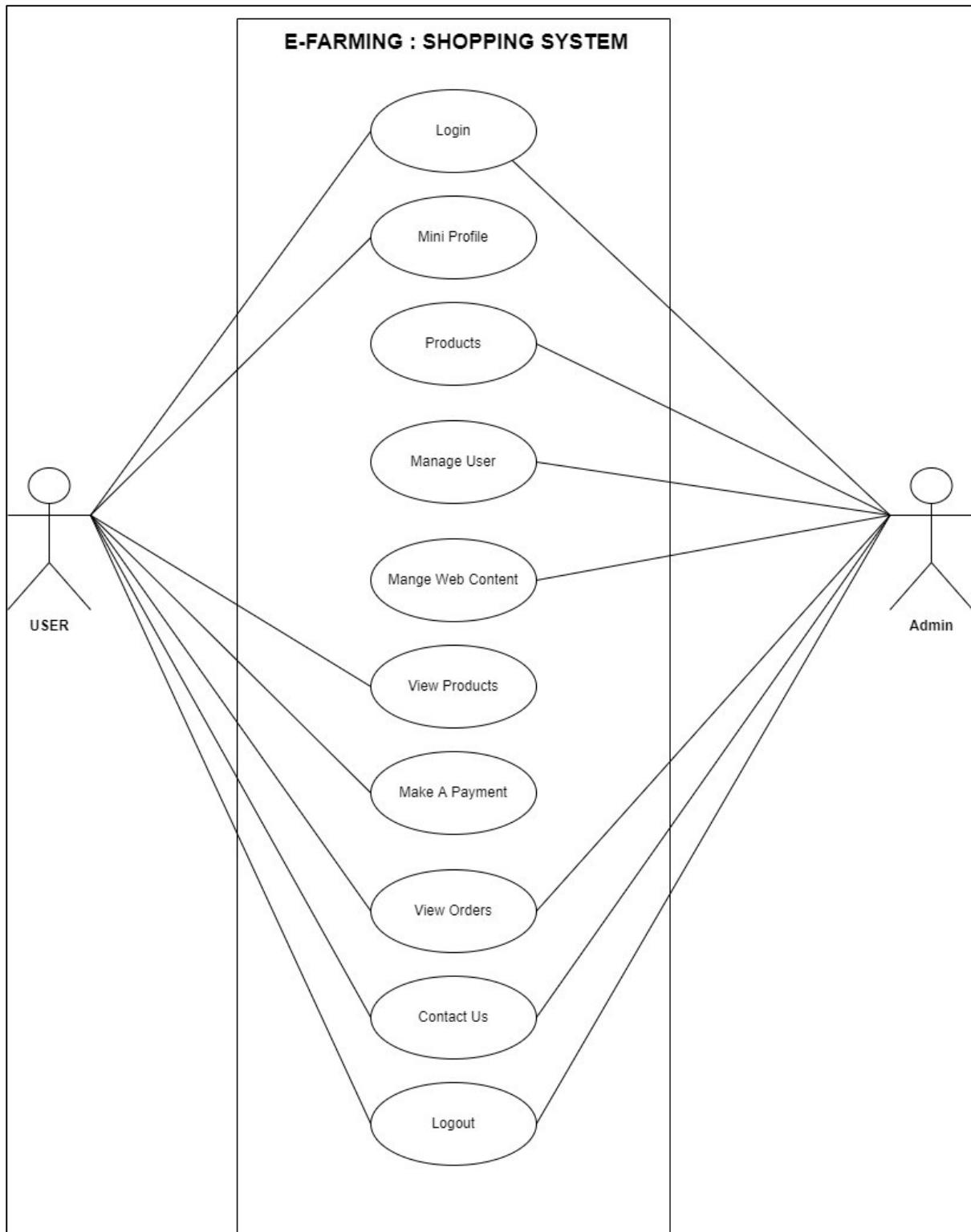


Figure 14: Use Case

Description: This is Use Case Diagram for E-farm: Shopping System, which shows the relationship between each actor and what functions they do.

7. Class Diagram

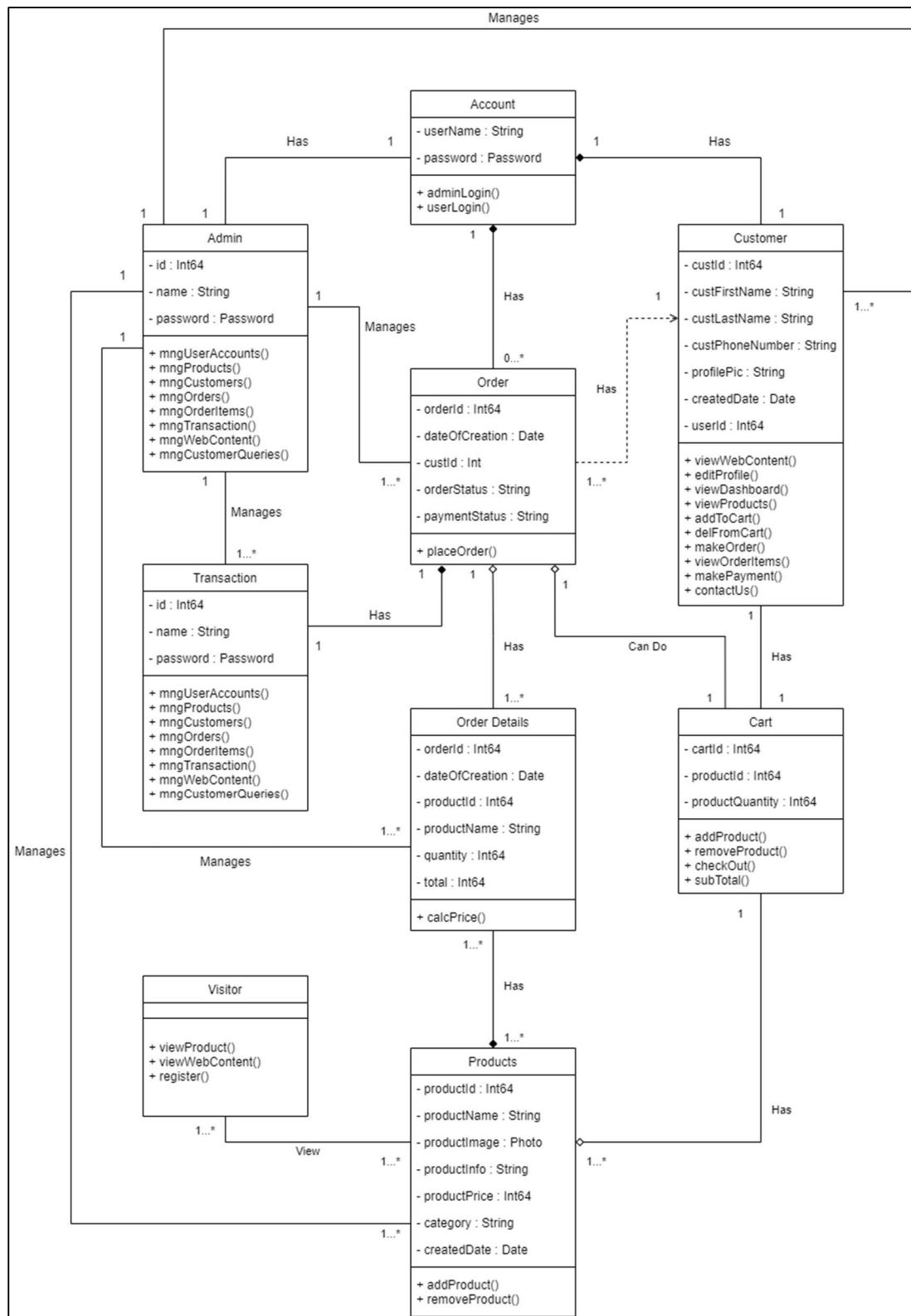


Figure 15: Class Diagram

Description: This is Class Diagram for E-farm: Shopping System, which shows the relationship between each entity and what functions they do.

8. Activity Diagram

➤ User

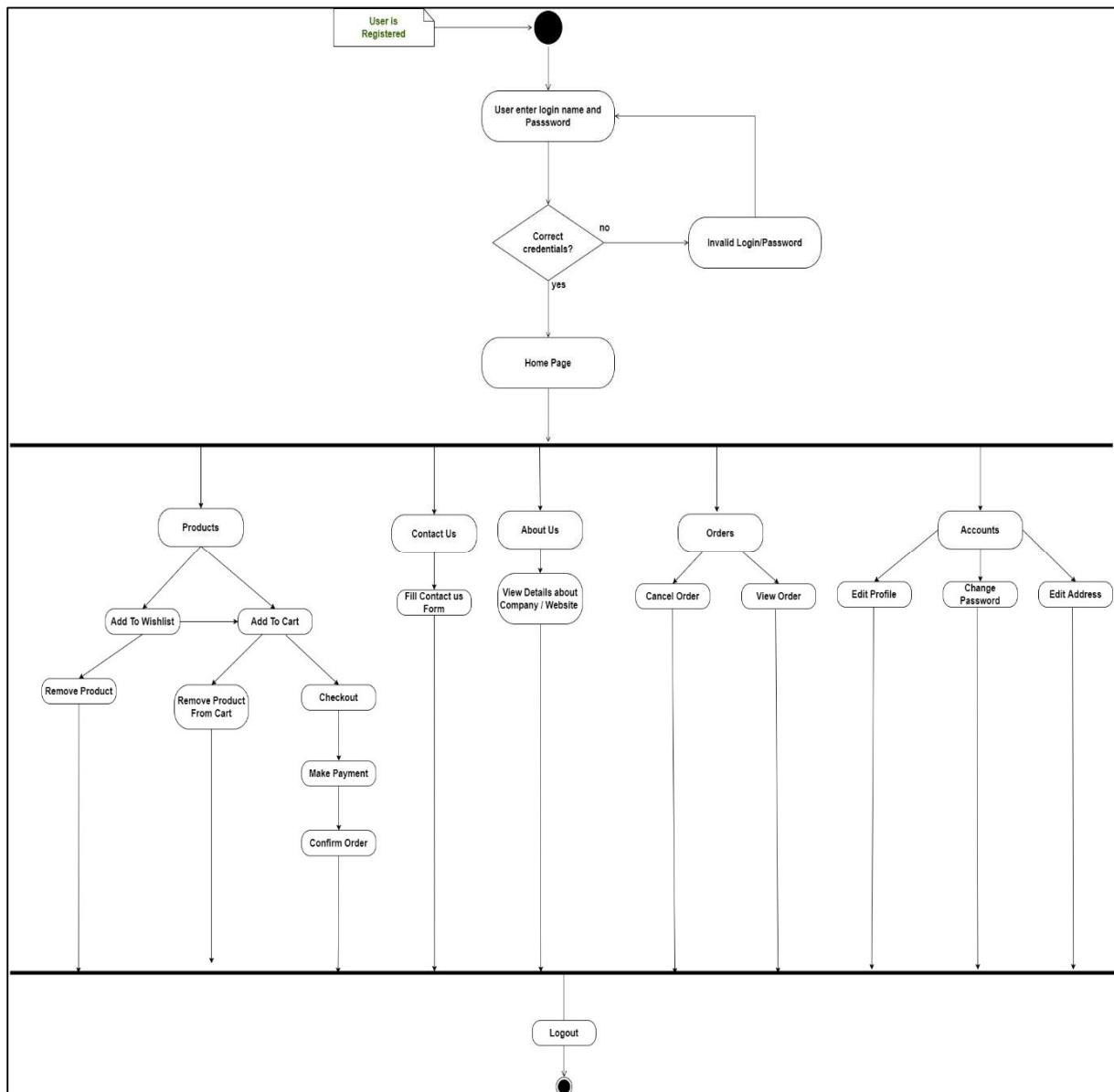


Figure 16: Activity Diagram: User

Description: This is an activity diagram of user for E-farm: Shopping System, which shows a series of action or flow of control in our system.

➤ Admin

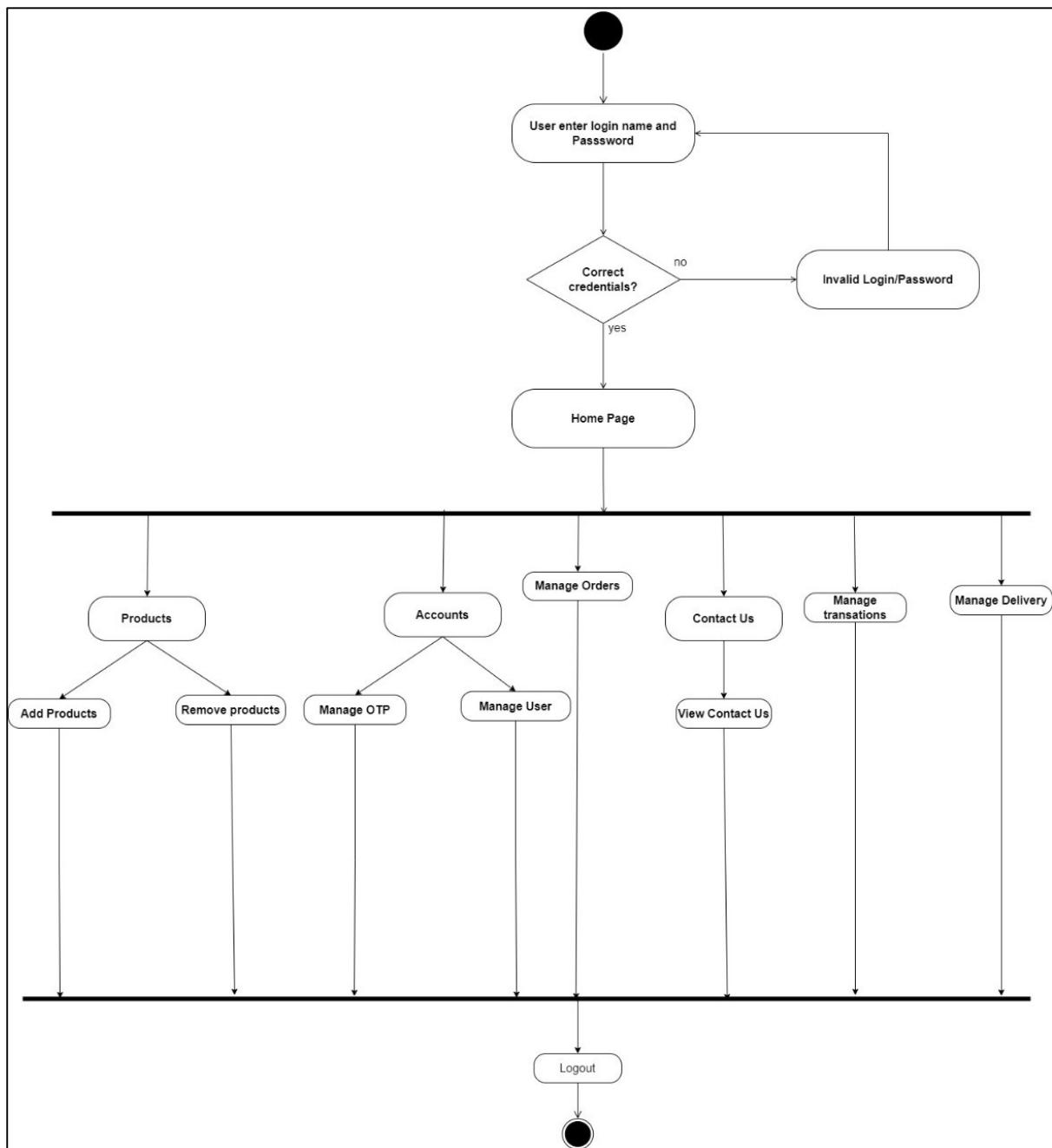


Figure 17: Activity Diagram: Admin

Description: This is an activity diagram of admin for E-farm: Shopping System which shows a series of action or flow of control in our system.

9. E-R Diagram

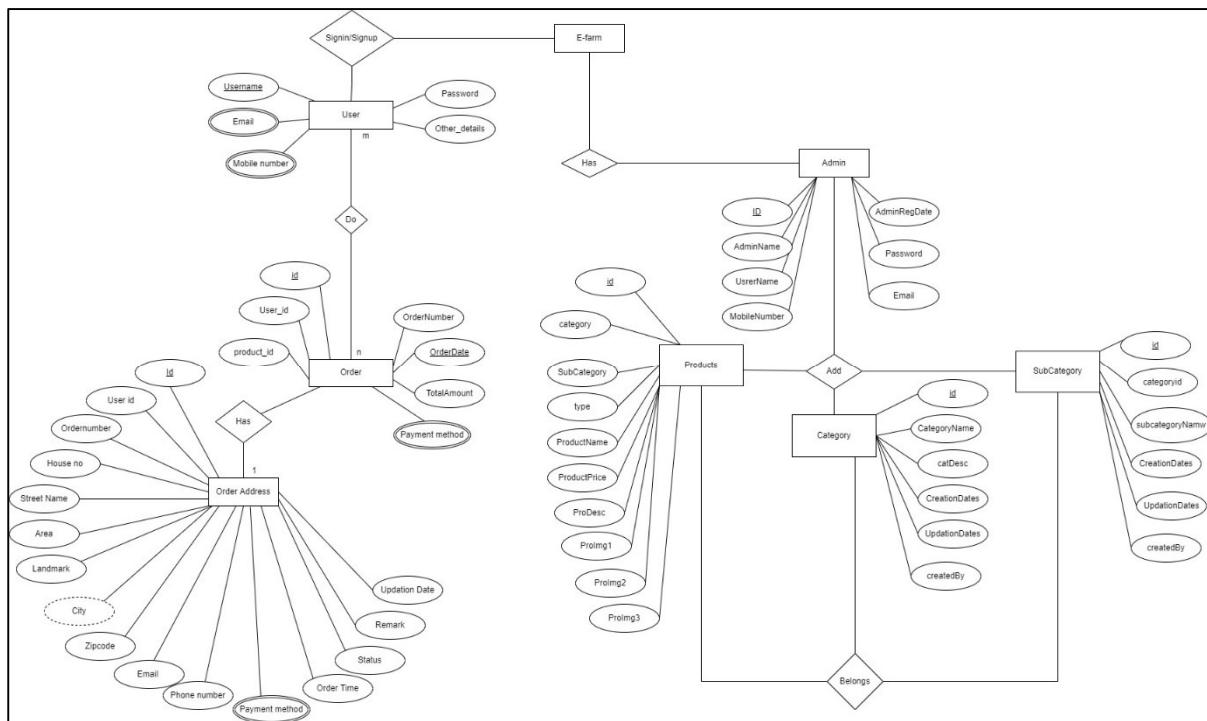


Figure 18: ER Diagram

Description: This is an ER Diagram for E-farm: Shopping System, which shows the graphical representation that depicts relationships among people, objects and events.

10. Data Dictionary

1. Table Name: accounts_customuser

Sr. No	Field Name	Datatype	Size	Description	Constraint	Example
1	last_login	timestamp	-	Primary Key, AI	Unique id	4
2	is_superuser	boolean	-	Not Null	Is user user is admin or not	True / False
3	username	Varchar	10	Not Null	user unique name	Virat67
4	first_name	varchar	15	Not Null	First Name of User	Virat
5	last_name	varchar	15	Not Null	Last Name Of user	Kohli
6	email	varchar	25	Not Null	User Email Id	abc@gmail.com
7	is_staff	boolean	-	Not Null	Is user is staff	True / False
8	is_active	boolean	-	Not Null	Is user Is active or not	True / False
9	date_joined	timestemp	-	Not Null	Date time of user	2022-09-10
10	phone	varchar	10	Null	User phone number	7423658910
11	is_verified	boolean	-	Not Null	Is user is verified or not	True / False
12	Password	Varchar	128	Not Null	Password of the account	*****

Table 6 : Table of Account_customer

Table Description: This table stores the information about registered user.

2. Table Name: accounts_otpcode

Sr. No	Field Name	Datatype	Size	Description	Constraint	Example
1	id	bigint	-	Primary Key, AI	Unique id	4
2	number	varchar	5	Not Null	Otp Number of user	14569
3	User_id	bigint	-	Foreign Key	Foreign Key from coustomer table	9

Table 7 : Table of Accounts_otpcode

Table Description: This table store all otp codes of user.

3. Table Name: customer_customer

Sr. No	Field Name	Datatype	Size	Constraint	Description	Example
1	Id	Bigint	-	Primary Key, AI	Unique id of this table	2
2	First_name	Varchar	255	NOT NULL	First name of Customer	Virat
3	Last_name	Varchar	255	NOT NULL	Last name of Customer	Kohli2
4	Phone_no	Bigint	-	NOT NULL	Phone number of the User	9875524456
5	User_id	Int	-	Foreign Key	(Foreign Key = id {From User Login Table (OneToOne Relationship)})	7
6	Email	Varchar	255	NOT NULL	User Email Id	luk@123@gmail.com
7	Shipping_addr	bigint	-	Foreign Key	Foreign Key from Shpping_Sddress table	2

Table 6: Table of customer_customer

Table Description: This table shows information about customers.

4. Table Name: Store_all_product

Sr. No	Field Name	Datatype	Size	Constraint	Description	Example
1	Id	Bigint	-	Primary Key, AI	Unique id	4
2	Photo	Varchar	100	NOT NULL	Photo of the item	store/product/2022/item1.jpg
3	Product name	Varchar	255	NOT NULL	Name of the item	Multi Harvester
4	Description	Varchar	255	NOT NULL	Product Info	Item Description
5	Original_Price	int	-	NOT NULL	Product Price	1000
6	Created date	Timestamp with time zone	-	NOT NULL	Date of updating that Product	2022-02-07 16:15:35.346967+05:30
7	Category	Varchar	255	NOT NULL	Machinery, Crops, Fertilizers, Seeds	Machinery
8	On_sale	boolen	-	Null	Product on sale	True
9	Is_featured	boolen	-	Null	Product on feature	True
10	After_discount_price	int	-	Null	Product price after discount	5000
11	Discount_amount	int	-	Null	Discount Amount in price	1500
12	Discount	varchar	5	Not Null	Discount Percentage	Default = '0%'

Table 9 : Table of Store_all_product

Table Description: This table has all the products divided on the category base.

5. Table Name: customer_order

Sr. No	Field Name	Datatype	Size	Constraint	Description	Example
1	id	bigint	-	Primary Key, AI	Unique id	4
2	Date_ordered	timestamp	-	Not Null	Order date	2022-09-12
3	Complete	boolen	-	Not Null	Order is completed or not	True / False
4	Order_number	varchar	15	Not Null	Order number	5467845
5	Order_status	varchar	30	Not Null	Order status	Pending
6	Payment_status	varchar	15	Not Null	Payment done or not	Pending
7	Order_made	int	3	Not Null	Order method	1
8	Shipping_method	varchar	15	Not Null	Shipping method	standard method
9	Deilivery_status	varchar	25	Not Null	Delivery status	In Progress
10	Refuned_amount	varchar	15	Not Null	Refunded amount	0
11	Customer_id	bigint	10	Foreign Key	Foreign Key from coustomer table	12

Table 10 : Table of customer_orders

Table Description: This table shows all customers orders.

6. Table Name: customer_orderitem

Sr. No	Field Name	Datatype	Size	Constraint	Description	Example
1	id	Bigint	-	Primary Key, AI	Unique id of this table	2
2	quantity	Int	-	NOT NULL	Quantity of Product	2
3	created_Date	Timestamp with time zone	-	NOT NULL	Creating date	2022-02-07 16:15:35.346967+05: 30
4	Order_id	Bigint	-	Foreign Key	Foreign Key from order table	5
5	Product_id	Bigint	-	Foreign key	Foreign key from product table	8
6	Customer_id	bigint	-	Foreign Key	Foreign Key from coustomer table	12
7	Delivery	varchar	25	Not Null	Shows the delivery status	In Progress

Table 11 : Table of customer_orderitem

Table Description: This table stores all the orderitem of each order.

7. Table Name: customer_shippingaddress

Sr. No	Field Name	Datatype	Size	Constraint	Description	Example
1	id	bigint	-	Primary Key, AI	Unique id	27
2	email	varchar	30	Not Null	Get user email	shreekoshti199@gmail.com
3	phone	bigint	10	Not Null	User phone nuber	0987654321
4	Address1	varchar	15	Not Null	Floor/ Buildling	G-1, Dewang Building, Bhagyoday Society,
5	Address2	varchar	25	Not Null	Street	Daman Rd, Chala;
6	landmark	varchar	15	Null	User landmark	Hanis Landmark
7	city	varchar	5	Not Null	User landmark	Pune
8	state	varchar	15	Not Null	User state	Maharashtra
9	zipcode	varchar	-	Not Null	User zipcode	396191
10	Shipping_method	varchar	-	Not Null	Get Shipping method	Standard Delivery
11	Date_added	timestemp	-	Not Null	Created date	2022-09-11 22:04:29.094151+05:30
12	Customer_id	bigint	-	Not Null	Foreign Key from customer table	9
13	Order_Id	bigint	-	Not Null	Foreign Key from Order table	17

Table 12 : Table of customer_shippingaddress

Table Description: This table shows all the Shipping Addresses

8. Table Name: accounts_deliverystatus

Sr. No	Field Name	Datatype	Size	Description	Constraint	Example
1	id	bigint	-	Primary Key ,AI	Unique id	4
2	Delivery_status	varchar	50	Not Null	Order Status	Order shipped
3	Customer_id	bigint	-	Foreign Key	Foreign Key from customer table	12
4	Order_id	bigint	-	Foreign Key	Foreign Key from orders table	23
5	Shipping_id	bigint	-	Foreign Key	Foreign Key from shippingaddress table	3
6	Transation_id_id	bigint	-	Foreign Key	Foreign Key from transations table	26

Table 13 : Table of Account_customer

Table Description: This table shows all customer's orders Delivery status.

9. Table Name: customer_transaction

Sr. No	Field Name	Datatype	Size	Constraint	Description	Example
1	id	bigint	-	Primary Key, AI	Unique id	1
2	Made_on	timestamp	-	Not Null	Date of transaction	2022-09-12 19:46:38.231652+05:30
3	amount	integer	-	Foreign Key	Foreign Key from Order table	5906
4	Transaction_id	varchar	-	Not Null	Unique Transaction is	PAY2ME20220912ODR 28
5	checksum	varchar	-	Not Null	For safe transaction	C4Cp3k1YrccsLwh5
6	result	varchar	-	Not Null	Payment status	Success
7	Payer_and_receiver	varchar	-	Not Null	Shows transaction credit / debit	Credit
8	Made_by_id	bigint	-	Foreign Key	Foreign Key from Order table	9
9	Order_id	bigint	-	Foreign Key	Foreign Key from Order table	22

Table 14 : Table of customer_transaction

Table Description: This table shows all the transactions of customers

10. Table Name: customer_returnorder_request

Sr. No	Field Name	Datatype	Size	Constraint	Description	Example
1	id	bigint	-	Primary Key, AI	Unique id	4
2	response	varchar	15	Not Null	Admin response to the return request	Accepted
3	Customer_Id	bigint	-	Foreign Key	Foreign Key from customer table	12
4	Delivery_status_Id	bigint	-	Foreign Key	Foreign Key from delivery status table	9
5	Order_Id	bigint	-	Foreign Key	Foreign Key from order table	4
6	Product_Id	bigint	-	Foreign Key	Foreign Key from product table	14
7	Shipping_address_Id	bigint	-	Foreign Key	Foreign Key from shipping table	2
8	Transaction_id_id	bigint	-	Foreign Key	Foreign Key from Transaction table	23

Table 15 : Table of customer_returnorder_request

Table Description: This table shows all return order requests

11. Table Name: Webpages_homeslider

Sr. No	Field Name	Datatype	Size	Constraint	Description	Example
1	id	Bigint	-	Primary Key, AI	Unique id of this table	2
2	title	Varchar	100	NOT NULL	Name of the Customer	Vladimir Punit
3	description	Varchar	255	NOT NULL	Email Id of the Customer	xyz@example.com
4	Btn_text	Varchar	255	NOT NULL	Subject of the Query	About stock of Forge Morage
5	image	Varchar	500	NOT NULL	Query Content	I wanted to buy the machinery but the product is not in stock can you please let me know when it is available.
6	Created_date	timestamp	-	Not Null	Entry created date	2022-09-12 19:46:38.231652+05: 30

Table 16 : Table of Webpages_homeslider

Table Description: This table stores the CMS (Home Page Sliders).

12. Table Name: Webpages_our_team

Sr. No	Field Name	Datatype	Size	Constraint	Description	Example
1	id	Bigint	-	Primary Key	Unique id of this table	2
2	First_name	Varchar	255	NOT NULL	First name of user	Rahul
3	Last_name	Varchar	255	NOT NULL	Last name of user	Jha
4	desc	Varchar	255	NOT NULL	Description	This is a CEO of of Company
5	Designation	Varchar	255	NOT NULL	Designation of the member	Web Devloper
6	Twitter link	Varchar	255	NOT NULL	Hyper link of twitter	https://twitter.co m/
7	Fb_link	Varchar	255	NOT NULL	Hyper link of Facebook	https://facebook.c om/
7	LinkedIn_link	Varchar	255	NOT NULL	Hyper link of LinkedIn	https://linkedin.co m/
8	photo	Varchar	100	NOT NULL	Member's photo	Image

Table 17 : Table of Webpages_our_team

Table Description: This table will manage all the front-end web content for Our Team Section.

13. Table Name: Contact_Us_contact_mail

Sr. No	Field Name	Datatype	Size	Constraint	Description	Example
1	id	Bigint	-	Primary Key	Unique id of this table	2
2	name	Varchar	100	NOT NULL	Name of the Customer	Vladimir Punit
3	email	Varchar	255	NOT NULL	Email Id of the Customer	xyz@example.co m
4	subject	Varchar	255	NOT NULL	Subject of the Query	About stock of Forge Morage
5	description	Varchar	500	NOT NULL	Query Content	I wanted to buy the machinery but the product is not in stock can you please let me know when it is available.

Table 7 : Table of Contact_Us_contact_mail

Table Description: This table stores the contact us mails from users.

11. Form Design

11.1. Development Phase – 1

1. Add to Cart Functionality:

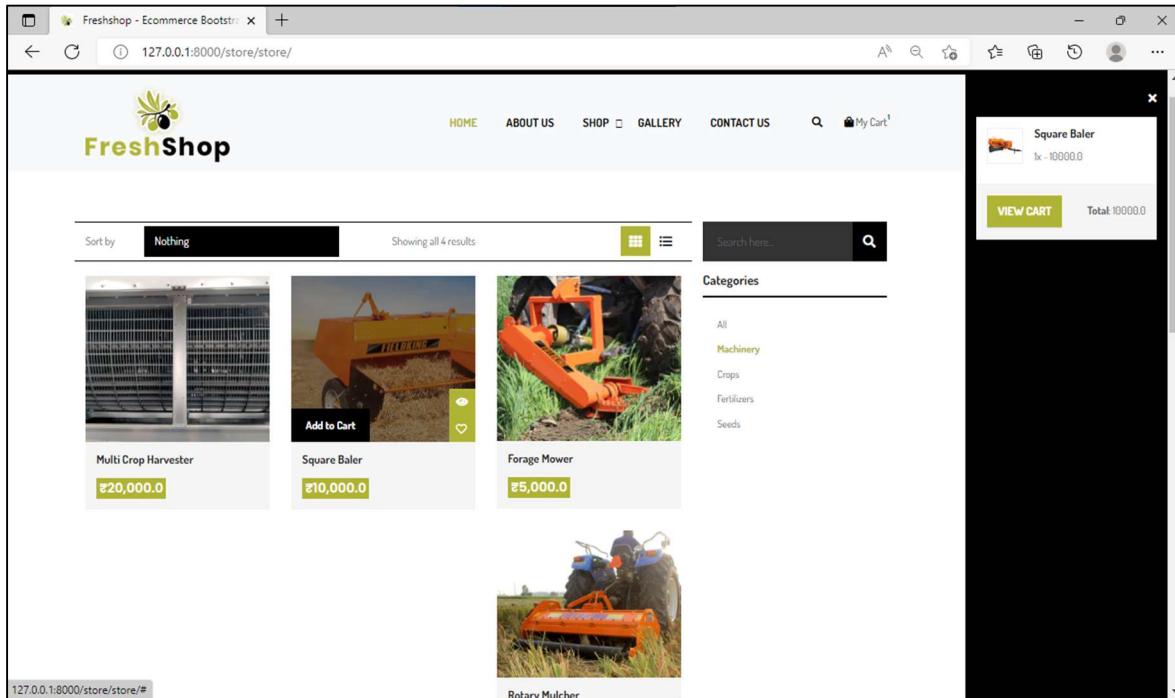


Figure 19: Add to Cart Functionality

Description: In the above figure, the middle cardboard section is our products. Once you hover, the add to cart button will appear. Clicking on that will update the user's cart, which is on the right side as shown in the fig.

- **Code highlights for the following:**

views.py ... \customer X 5 wishlist.html 5 nav.html 5 context_processors.py 5 store.html JS wishlist.js D ...

```
efarm > customer > views.py
1  from django.shortcuts import render
2  from .models import *
3  from django.http import JsonResponse
4  import json
5  from django.http import Http404, HttpResponseRedirect, HttpResponseForbidden
6  from django.contrib.auth.decorators import login_required
7
8  # Create your views here.
9
10 def cart(request):
11     # If user is Logged in then
12     try:
13         if request.user.is_authenticated:
14             customer = request.user.customer
15             order, created = Order.objects.get_or_create(customer=customer, complete=False)
16             items = order.orderitem_set.all()
17             cartItems = order.get_cart_items
18             print("Views.py Cart Logged in User")
19
20     # if not
21     else:
22
23     # WILL continue This once we done with Logged in user add to cart module
24     try:
25         cart = json.loads(request.COOKIES['cart'])
26     except:
27         cart = {}
28         print(cart)
29         items = []
30         order = {'get_cart_total': 0, 'get_cart_items': 0}
31         cartItems = order['get_cart_items']
32
33         for i in cart:
34             cartItems += cart[i]['quantity']


```

Figure 20: Product Page

2. User's Cart:

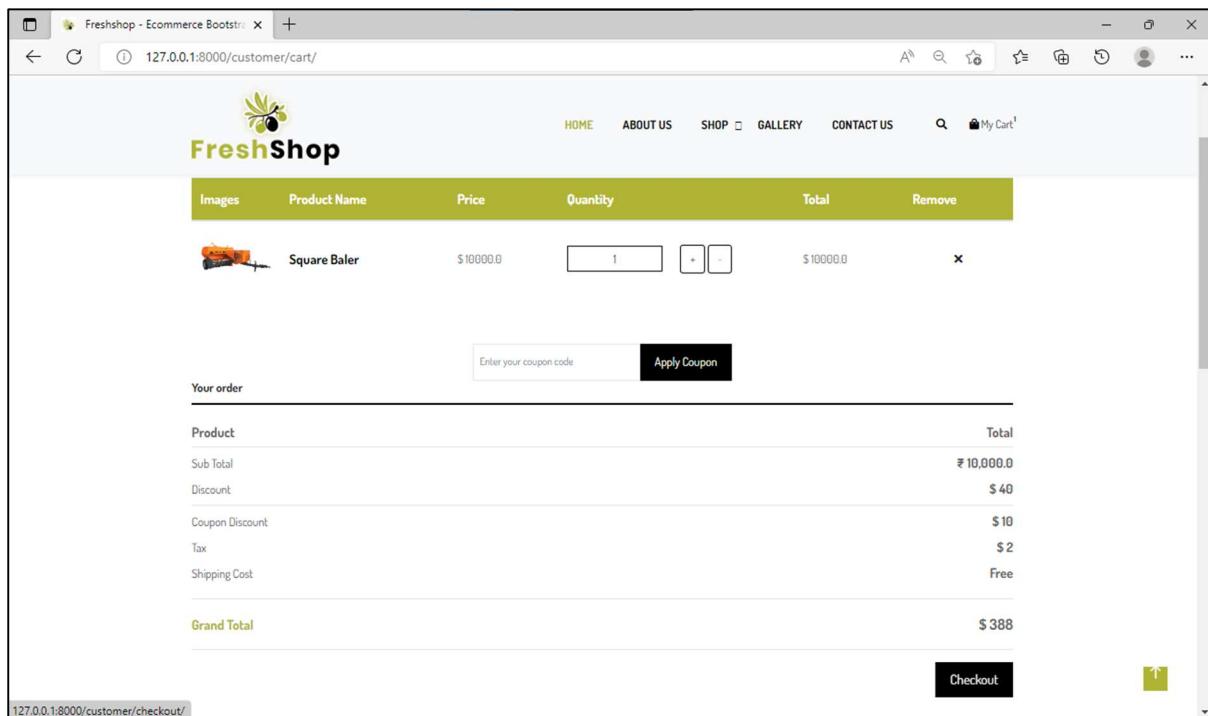


Figure 21: Cart Page

Description: Here, the user can see his/her cart and can easily update if needed. This page also shows the cart summary.

➤ Code highlights for the following:

```

customer      wishlist.html      nav.html      context_processors.py      store.html      JS wishlist.js      JS cart.js
efarm > customer > static > customer-static > JS cart.js > addCookieItem
  ...
  function updateUserOrder(productId, action) {
    console.log('User is logged in, sending data ...')
    var url = '/customer/update_item/'

    fetch(url, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
        'X-CSRFToken': csrftoken,
      },
      body: JSON.stringify({productId: productId, action: action})
    })

    .then((response) => {
      return response.json()
    })

    .then((data) => {
      console.log('data:', data)
      location.reload()
    })
  }

  function addCookieItem(productId, action) {
    console.log('Guest User Sending Data ....')

    if (action == 'add') {
      console.log('cart:', cart)
      if (cart[productId] === undefined) {
        cart[productId] = { 'quantity': 1 }
      } else {
        cart[productId]['quantity'] += 1
      }
    }
  }
}

```

Figure 22: Cart page Code

3. Checkout Page:

Figure 23 : Checkout Page

Description: This is the Checkout Page where users need to enter shipping details. To visit this page, a user needs to be logged in; otherwise, a user will be redirected to the login page. At the bottom right of the checkout page, there is a payment button to invoke the payment method.

➤ Code highlights for the following:

```

views.py ...\\customer X wishlist.html nav.html context_processors.py store.html JS wishlist.js > □ ...
efarm > customer > views.py
...
47     def updateItem(request):
48         data = json.loads(request.body)
49         productId = data['productId']
50         action = data['action']
51
52         print('Action: ', action)
53         print('ProductId: ', productId)
54
55         customer = request.user.customer
56         product = all_Product.objects.get(id=productId)
57         order, created = Order.objects.get_or_create(customer=customer, complete=False)
58
59         orderItem, created = OrderItem.objects.get_or_create(order_number=order, product=product)
60
61         if action == 'add':
62             | orderItem.quantity = (orderItem.quantity + 1)
63         elif action == 'remove':
64             | orderItem.quantity = (orderItem.quantity - 1)
65
66         orderItem.save()
67
68         if orderItem.quantity <= 0 or action == 'delete':
69             | orderItem.delete()
70
71
72     return JsonResponse({'Item was Added', safe=False})
73
74 @login_required
75 def checkout(request):
76     | return render(request, 'store/checkout.html')
77
78
79     def processOrder(request):
80         | return JsonResponse('Payment Complete', safe=False)
81

```

Figure 24 : Checkout Page Code

11.2. Development Phase – 2

1. Home Page:

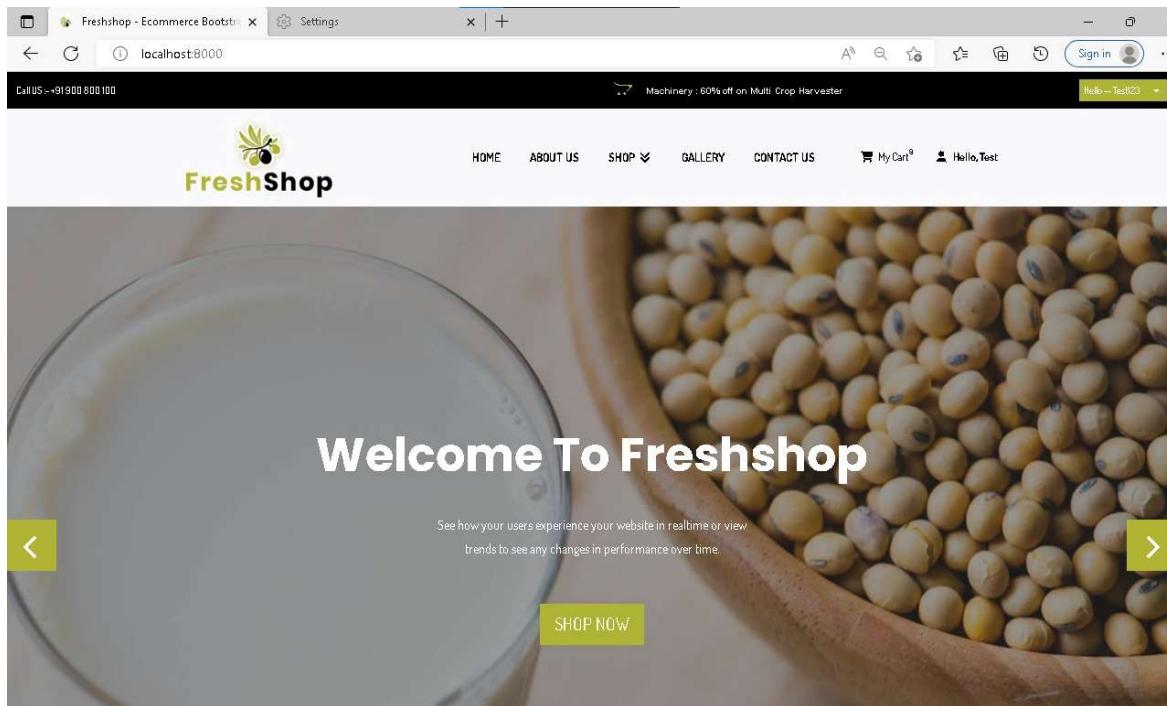


Figure 25 : Home Page - 1

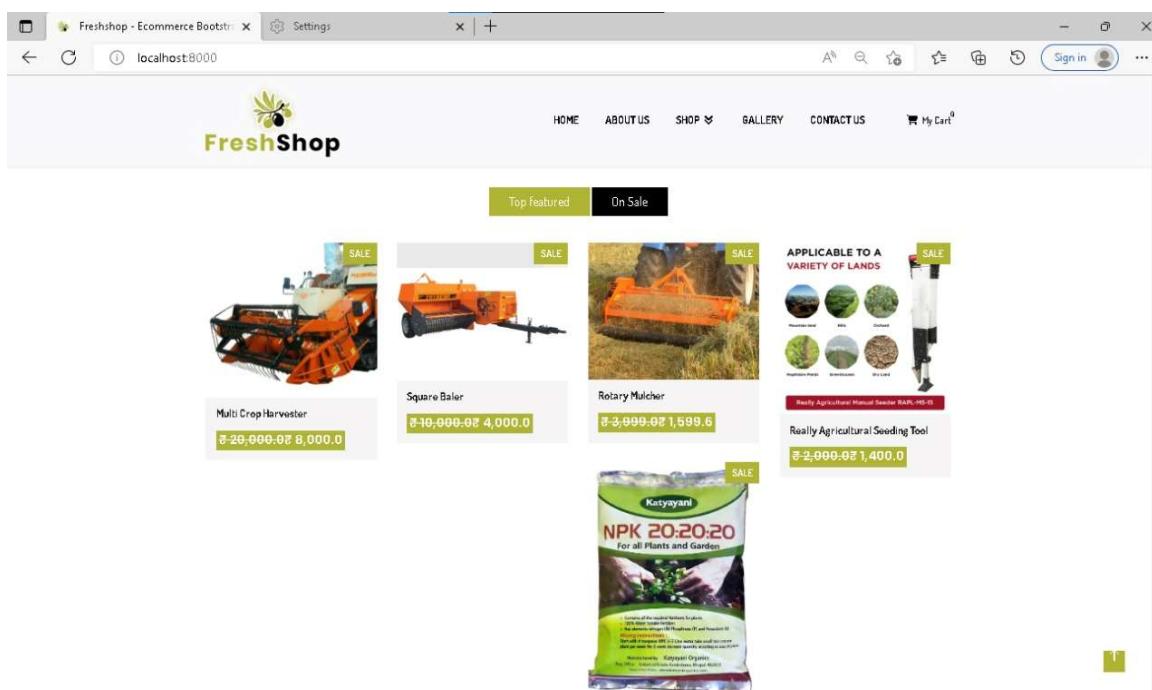
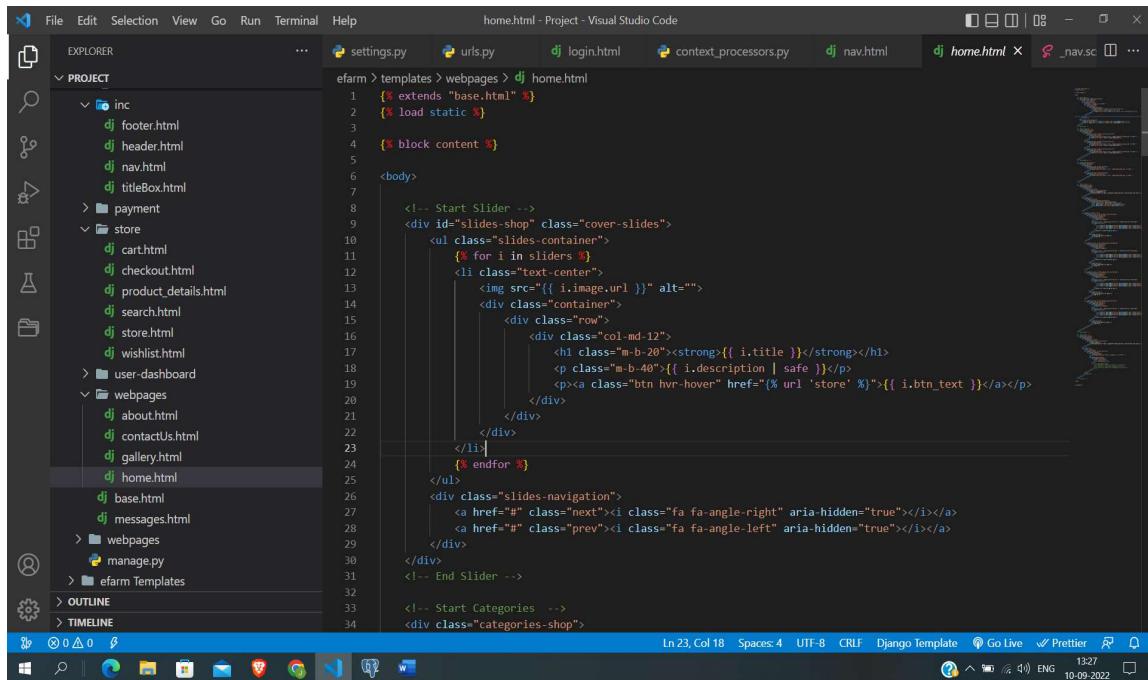


Figure 26 : Home Page – 2

Description: This is the Home Page where users comes when they open website.

➤ **Code highlights for the following:**



The screenshot shows the Visual Studio Code interface with the 'home.html' file open in the editor. The code is a Django template for the home page, featuring a slider and categories section. The code is color-coded for syntax highlighting.

```

1  {% extends "base.html" %} 
2  {% load static %} 
3 
4  {% block content %} 
5 
6  <body> 
7 
8      <!-- Start Slider --> 
9      <div id="slides-shop" class="cover-slides"> 
10         <ul class="slides-container"> 
11             {% for i in sliders %} 
12                 <li class="text-center"> 
13                      
14                     <div class="container"> 
15                         <div class="row"> 
16                             <div class="col-md-12"> 
17                                 <h1 class="m-b-20"><strong>{{ i.title }}</strong></h1> 
18                                 <p class="m-b-40">{{ i.description | safe }}</p> 
19                                 <p><a class="btn hvr-hover" href="{% url 'store' %}{{ i.btn_text }}</a></p> 
20                             </div> 
21                         </div> 
22                     </div> 
23                 {% endfor %} 
24             </ul> 
25             <div class="slides-navigation"> 
26                 <a href="#" class="next"><i class="fa fa-angle-right" aria-hidden="true"></i></a> 
27                 <a href="#" class="prev"><i class="fa fa-angle-left" aria-hidden="true"></i></a> 
28             </div> 
29         </div> 
30         <!-- End Slider --> 
31         <!-- Start Categories --> 
32         <div class="categories-shop"> 
33 
34 

```

Figure 27 : Home Page Code

2. My Accounts Page

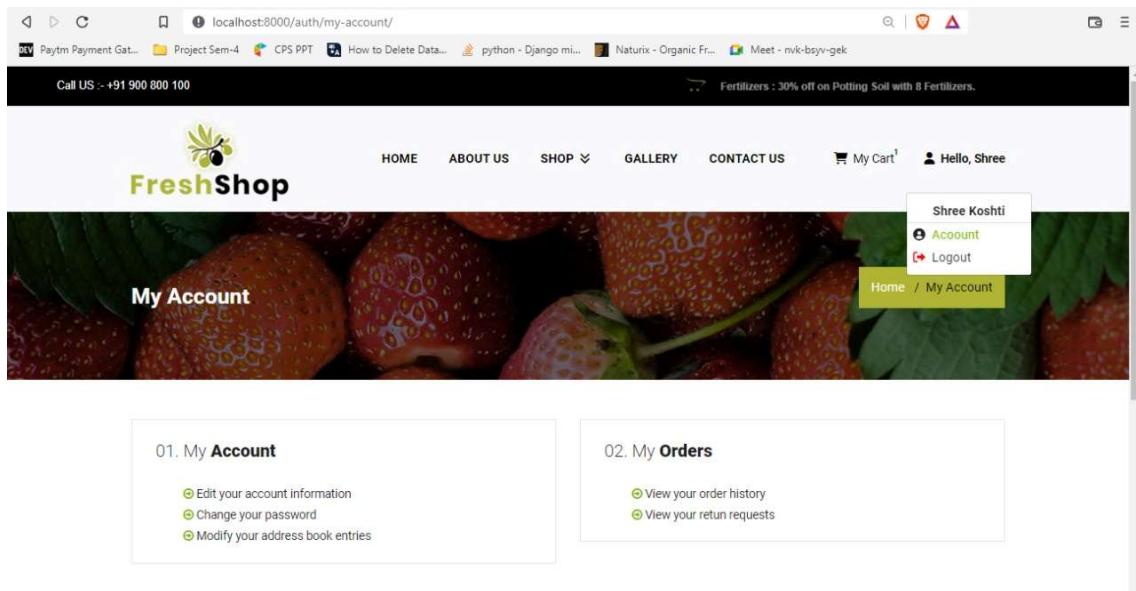


Figure 28 : My Account Design

Description: This is the Account Page where users can change their personal information , password as well as they can see orders option in that they can see all past orders list.

➤ **Code highlights for the following:**

```
File Edit Selection View Go Run Terminal Help
my-account.html - Project - Visual Studio Code
EXPLORER ... dj home.html dj login.html urls.py dj my-account.html x settings.py
PROJECT
  - settings.py
  - urls.py
  - wsgi.py
  media
  static
  store
  templates
    accounts
    admin
    accounts
    home
    includes
    layouts
      .gitkeep
    dashboard
    emails
    inc
    payment
    store
    templates
    user-dashboard
      account-information.html
      address-book.html
      change-password.html
      my-account.html
      order-history.html
      return-order.html
    webpages
    base.html
    messages.html
  OUTLINE
  TIMELINE
dj home.html dj login.html urls.py dj my-account.html x settings.py
  efarm > templates > user-dashboard > dj my-account.html
  1 extends "base.html"
  2 load humanize
  3 load static
  4
  5 % block content %
  6
  7
  8 <link rel="stylesheet" href="{% static 'user-dashboard-static/naturix/css/theme.css' %}" />
  9 {% comment %} <link rel="stylesheet" href="{% static 'user-dashboard-static/naturix/css/plugin/bootstrap.min.css' %}" />
  10 <link rel="stylesheet" href="{% static 'user-dashboard-static/naturix/css/plugin/bootstrap-select.min.css' %}" /> {% endcomment %}
  11 <link rel="stylesheet" href="{% static 'user-dashboard-static/naturix/css/plugin/owl.carousel.css' %}" />
  12 <link rel="stylesheet" href="{% static 'user-dashboard-static/naturix/css/plugin/fontawesome.min.css' %}" />
  13 <link rel="stylesheet" href="{% static 'user-dashboard-static/naturix/css/plugin/animate.css' %}" />
  14
  15 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-eJkQeHVZoWgkPjN8hWVd+qkZDfZLWxq0ZuZGvqK9XyvYQKc" crossorigin="anonymous">
  16
  17
  18
  19 <div class="all-title-box">
  20   <div class="container">
  21     <div class="row">
  22       <div class="col-lg-12">
  23         <h2>My Account</h2>
  24         <ul class="breadcrumb">
  25           <li class="breadcrumb-item"><a href="#">Home</a></li>
  26           <li class="breadcrumb-item active">My Account</li>
  27         </ul>
  28       </div>
  29     </div>
  30   </div>
  31 </div>
  32
  33 <!-- My Account Starts-->
  34 <section class="account-page ptb-50">
  35   <div class="container">
  36     <div class="row">
  37       <div class="col-6">
  38
  39
  40
  41
  42
  43
  44
  45
  46
  47
  48
  49
  50
  51
  52
  53
  54
  55
  56
  57
  58
  59
  60
  61
  62
  63
  64
  65
  66
  67
  68
  69
  70
  71
  72
  73
  74
  75
  76
  77
  78
  79
  80
  81
  82
  83
  84
  85
  86
  87
  88
  89
  90
  91
  92
  93
  94
  95
  96
  97
  98
  99
  100
  101
  102
  103
  104
  105
  106
  107
  108
  109
  110
  111
  112
  113
  114
  115
  116
  117
  118
  119
  120
  121
  122
  123
  124
  125
  126
  127
  128
  129
  130
  131
  132
  133
  134
  135
  136
  137
  138
  139
  140
  141
  142
  143
  144
  145
  146
  147
  148
  149
  150
  151
  152
  153
  154
  155
  156
  157
  158
  159
  160
  161
  162
  163
  164
  165
  166
  167
  168
  169
  170
  171
  172
  173
  174
  175
  176
  177
  178
  179
  180
  181
  182
  183
  184
  185
  186
  187
  188
  189
  190
  191
  192
  193
  194
  195
  196
  197
  198
  199
  200
  201
  202
  203
  204
  205
  206
  207
  208
  209
  210
  211
  212
  213
  214
  215
  216
  217
  218
  219
  220
  221
  222
  223
  224
  225
  226
  227
  228
  229
  230
  231
  232
  233
  234
  235
  236
  237
  238
  239
  240
  241
  242
  243
  244
  245
  246
  247
  248
  249
  250
  251
  252
  253
  254
  255
  256
  257
  258
  259
  260
  261
  262
  263
  264
  265
  266
  267
  268
  269
  270
  271
  272
  273
  274
  275
  276
  277
  278
  279
  280
  281
  282
  283
  284
  285
  286
  287
  288
  289
  290
  291
  292
  293
  294
  295
  296
  297
  298
  299
  300
  301
  302
  303
  304
  305
  306
  307
  308
  309
  310
  311
  312
  313
  314
  315
  316
  317
  318
  319
  320
  321
  322
  323
  324
  325
  326
  327
  328
  329
  330
  331
  332
  333
  334
  335
  336
  337
  338
  339
  340
  341
  342
  343
  344
  345
  346
  347
  348
  349
  350
  351
  352
  353
  354
  355
  356
  357
  358
  359
  360
  361
  362
  363
  364
  365
  366
  367
  368
  369
  370
  371
  372
  373
  374
  375
  376
  377
  378
  379
  380
  381
  382
  383
  384
  385
  386
  387
  388
  389
  390
  391
  392
  393
  394
  395
  396
  397
  398
  399
  400
  401
  402
  403
  404
  405
  406
  407
  408
  409
  410
  411
  412
  413
  414
  415
  416
  417
  418
  419
  420
  421
  422
  423
  424
  425
  426
  427
  428
  429
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  450
  451
  452
  453
  454
  455
  456
  457
  458
  459
  460
  461
  462
  463
  464
  465
  466
  467
  468
  469
  470
  471
  472
  473
  474
  475
  476
  477
  478
  479
  480
  481
  482
  483
  484
  485
  486
  487
  488
  489
  490
  491
  492
  493
  494
  495
  496
  497
  498
  499
  500
  501
  502
  503
  504
  505
  506
  507
  508
  509
  510
  511
  512
  513
  514
  515
  516
  517
  518
  519
  520
  521
  522
  523
  524
  525
  526
  527
  528
  529
  530
  531
  532
  533
  534
  535
  536
  537
  538
  539
  540
  541
  542
  543
  544
  545
  546
  547
  548
  549
  550
  551
  552
  553
  554
  555
  556
  557
  558
  559
  559
  560
  561
  562
  563
  564
  565
  566
  567
  568
  569
  570
  571
  572
  573
  574
  575
  576
  577
  578
  579
  580
  581
  582
  583
  584
  585
  586
  587
  588
  589
  589
  590
  591
  592
  593
  594
  595
  596
  597
  598
  599
  600
  601
  602
  603
  604
  605
  606
  607
  608
  609
  609
  610
  611
  612
  613
  614
  615
  616
  617
  618
  619
  619
  620
  621
  622
  623
  624
  625
  626
  627
  628
  629
  629
  630
  631
  632
  633
  634
  635
  636
  637
  638
  639
  639
  640
  641
  642
  643
  644
  645
  646
  647
  648
  649
  649
  650
  651
  652
  653
  654
  655
  656
  657
  658
  659
  659
  660
  661
  662
  663
  664
  665
  666
  667
  668
  669
  669
  670
  671
  672
  673
  674
  675
  676
  677
  678
  679
  679
  680
  681
  682
  683
  684
  685
  686
  687
  688
  689
  689
  690
  691
  692
  693
  694
  695
  696
  697
  698
  699
  699
  700
  701
  702
  703
  704
  705
  706
  707
  708
  709
  709
  710
  711
  712
  713
  714
  715
  716
  717
  718
  719
  719
  720
  721
  722
  723
  724
  725
  726
  727
  728
  729
  729
  730
  731
  732
  733
  734
  735
  736
  737
  738
  739
  739
  740
  741
  742
  743
  744
  745
  746
  747
  748
  749
  749
  750
  751
  752
  753
  754
  755
  756
  757
  758
  759
  759
  760
  761
  762
  763
  764
  765
  766
  767
  768
  769
  769
  770
  771
  772
  773
  774
  775
  776
  777
  778
  779
  779
  780
  781
  782
  783
  784
  785
  786
  787
  788
  789
  789
  790
  791
  792
  793
  794
  795
  796
  797
  798
  799
  799
  800
  801
  802
  803
  804
  805
  806
  807
  808
  809
  809
  810
  811
  812
  813
  814
  815
  816
  817
  818
  819
  819
  820
  821
  822
  823
  824
  825
  826
  827
  828
  829
  829
  830
  831
  832
  833
  834
  835
  836
  837
  838
  839
  839
  840
  841
  842
  843
  844
  845
  846
  847
  848
  849
  849
  850
  851
  852
  853
  854
  855
  856
  857
  858
  859
  859
  860
  861
  862
  863
  864
  865
  866
  867
  868
  869
  869
  870
  871
  872
  873
  874
  875
  876
  877
  878
  879
  879
  880
  881
  882
  883
  884
  885
  886
  887
  888
  889
  889
  890
  891
  892
  893
  894
  895
  896
  897
  898
  899
  899
  900
  901
  902
  903
  904
  905
  906
  907
  908
  909
  909
  910
  911
  912
  913
  914
  915
  916
  917
  918
  919
  919
  920
  921
  922
  923
  924
  925
  926
  927
  928
  929
  929
  930
  931
  932
  933
  934
  935
  936
  937
  938
  939
  939
  940
  941
  942
  943
  944
  945
  946
  947
  948
  949
  949
  950
  951
  952
  953
  954
  955
  956
  957
  958
  959
  959
  960
  961
  962
  963
  964
  965
  966
  967
  968
  969
  969
  970
  971
  972
  973
  974
  975
  976
  977
  978
  979
  979
  980
  981
  982
  983
  984
  985
  986
  987
  988
  989
  989
  990
  991
  992
  993
  994
  995
  996
  997
  998
  999
  999
  1000
  1001
  1002
  1003
  1004
  1005
  1006
  1007
  1008
  1009
  1009
  1010
  1011
  1012
  1013
  1014
  1015
  1016
  1017
  1018
  1019
  1019
  1020
  1021
  1022
  1023
  1024
  1025
  1026
  1027
  1028
  1029
  1029
  1030
  1031
  1032
  1033
  1034
  1035
  1036
  1037
  1038
  1039
  1039
  1040
  1041
  1042
  1043
  1044
  1045
  1046
  1047
  1048
  1048
  1049
  1050
  1051
  1052
  1053
  1054
  1055
  1056
  1057
  1058
  1059
  1059
  1060
  1061
  1062
  1063
  1064
  1065
  1066
  1067
  1068
  1069
  1069
  1070
  1071
  1072
  1073
  1074
  1075
  1076
  1077
  1078
  1078
  1079
  1080
  1081
  1082
  1083
  1084
  1085
  1086
  1087
  1088
  1088
  1089
  1090
  1091
  1092
  1093
  1094
  1095
  1096
  1097
  1097
  1098
  1099
  1099
  1100
  1101
  1102
  1103
  1104
  1105
  1106
  1107
  1108
  1109
  1109
  1110
  1111
  1112
  1113
  1114
  1115
  1116
  1117
  1118
  1119
  1119
  1120
  1121
  1122
  1123
  1124
  1125
  1126
  1127
  1128
  1129
  1129
  1130
  1131
  1132
  1133
  1134
  1135
  1136
  1137
  1138
  1138
  1139
  1140
  1141
  1142
  1143
  1144
  1145
  1146
  1147
  1148
  1148
  1149
  1150
  1151
  1152
  1153
  1154
  1155
  1156
  1157
  1158
  1158
  1159
  1160
  1161
  1162
  1163
  1164
  1165
  1166
  1167
  1168
  1168
  1169
  1170
  1171
  1172
  1173
  1174
  1175
  1176
  1177
  1178
  1178
  1179
  1180
  1181
  1182
  1183
  1184
  1185
  1186
  1187
  1187
  1188
  1189
  1190
  1191
  1192
  1193
  1194
  1195
  1196
  1197
  1198
  1198
  1199
  1199
  1200
  1201
  1202
  1203
  1204
  1205
  1206
  1207
  1208
  1208
  1209
  1210
  1211
  1212
  1213
  1214
  1215
  1216
  1217
  1218
  1218
  1219
  1220
  1221
  1222
  1223
  1224
  1225
  1226
  1227
  1228
  1228
  1229
  1230
  1231
  1232
  1233
  1234
  1235
  1236
  1237
  1238
  1238
  1239
  1240
  1241
  1242
  1243
  1244
  1245
  1246
  1247
  1248
  1248
  1249
  1250
  1251
  1252
  1253
  1254
  1255
  1256
  1257
  1258
  1258
  1259
  1260
  1261
  1262
  1263
  1264
  1265
  1266
  1267
  1268
  1268
  1269
  1270
  1271
  1272
  1273
  1274
  1275
  1276
  1277
  1277
  1278
  1279
  1280
  1281
  1282
  1283
  1284
  1285
  1286
  1287
  1287
  1288
  1289
  1290
  1291
  1292
  1293
  1294
  1295
  1296
  1297
  1297
  1298
  1299
  1299
  1300
  1301
  1302
  1303
  1304
  1305
  1306
  1307
  1308
  1308
  1309
  1310
  1311
  1312
  1313
  1314
  1315
  1316
  1317
  1318
  1318
  1319
  1320
  1321
  1322
  1323
  1324
  1325
  1326
  1327
  1328
  1328
  1329
  1330
  1331
  1332
  1333
  1334
  1335
  1336
  1337
  1338
  1338
  1339
  1340
  1341
  1342
  1343
  1344
  1345
  1346
  1347
  1348
  1348
  1349
  1350
  1351
  1352
  1353
  1354
  1355
  1356
  1357
  1358
  1358
  1359
  1360
  1361
  1362
  1363
  1364
  1365
  1366
  1367
  1368
  1368
  1369
  1370
  1371
  1372
  1373
  1374
  1375
  1376
  1377
  1378
  1378
  1379
  1380
  1381
  1382
  1383
  1384
  1385
  1386
  1387
  1387
  1388
  1389
  1390
  1391
  1392
  1393
  1394
  1395
  1396
  1397
  1398
  1398
  1399
  1400
  1401
  1402
  1403
  1404
  1405
  1406
  1407
  1408
  1408
  1409
  1410
  1411
  1412
  1413
  1414
  1415
  1416
  1417
  1418
  1418
  1419
  1420
  1421
  1422
  1423
  1424
  1425
  1426
  1427
  1428
  1428
  1429
  1430
  1431
  1432
  1433
  1434
  1435
  1436
  1437
  1438
  1438
  1439
  1440
  1441
  1442
  1443
  1444
  1445
  1446
  1447
  1448
  1448
  1449
  1450
  1451
  1452
  1453
  1454
  1455
  1456
  1457
  1458
  1458
  1459
  1460
  1461
  1462
  1463
  1464
  1465
  1466
  1467
  1468
  1468
  1469
  1470
  1471
  1472
  1473
  1474
  1475
  1476
  1477
  1478
  1478
  1479
  1480
  1481
  1482
  1483
  1484
  1485
  1486
  1487
  1488
  1488
  1489
  1490
  1491
  1492
  1493
  1494
  1495
  1496
  1497
  1498
  1498
  1499
  1500
  1501
  1502
  1503
  1504
  1505
  1506
  1507
  1508
  1508
  1509
  1510
  1511
  1512
  1513
  1514
  1515
  1516
  1517
  1518
  1518
  1519
  1520
  1521
  1522
  1523
  1524
  1525
  1526
  1527
  1528
  1528
  1529
  1530
  1531
  1532
  1533
  1534
  1535
  1536
  1537
  1538
  1538
  1539
  1540
  1541
  1542
  1543
  1544
  1545
  1546
  1547
  1548
  1548
  1549
  1550
  1551
  1552
  1553
  1554
  1555
  1556
  1557
  1558
  1558
  1559
  1560
  1561
  1562
  1563
  1564
  1565
  1566
  1567
  1568
  1568
  1569
  1570
  1571
  1572
  1573
  1574
  1575
  1576
  1577
  1578
  1578
  1579
  1580
  1581
  1582
  1583
  1584
  1585
  1586
  1587
  1587
  1588
  1589
  1590
  1591
  1592
  1593
  1594
  1595
  1596
  1597
  1598
  1598
  1599
  1600
  1601
  1602
  1603
  1604
  1605
  1606
  1607
  1608
  1608
  1609
  1610
  1611
  1612
  1613
  1614
  1615
  1616
  1617
  1618
  1618
  1619
  1620
  1621
  1622
  1623
  1624
  1625
  1626
  1627
  1628
  1629
  1629
  1630
  1631
  1632
  1633
  1634
  1635
  1636
  1637
  1638
  1638
  1639
  1640
  1641
  1642
  1643
  1644
  1645
  1646
  1647
  1648
  1648
  1649
  1650
  1651
  1652
  1653
  1654
  1655
  1656
  1657
  1658
  1658
  1659
  1660
  1661
  1662
  1663
  1664
  1665
  1666
  1667
  1668
  1668
  1669
  1670
  1671
  1672
  1673
  1674
  1675
  1676
  1677
  1678
  1678
  1679
  1680
  1681
  1682
  1683
  1684
  1685
  1686
  1687
  1687
  1688
  1689
  1690
  1691
  1692
  1693
  1694
  1695
  1696
  1697
  1698
  1698
  1699
  1700
  1701
  1702
  170
```

Figure 29 : My Account Code

11.3. Development Phase – 3

1. Admin SideBar

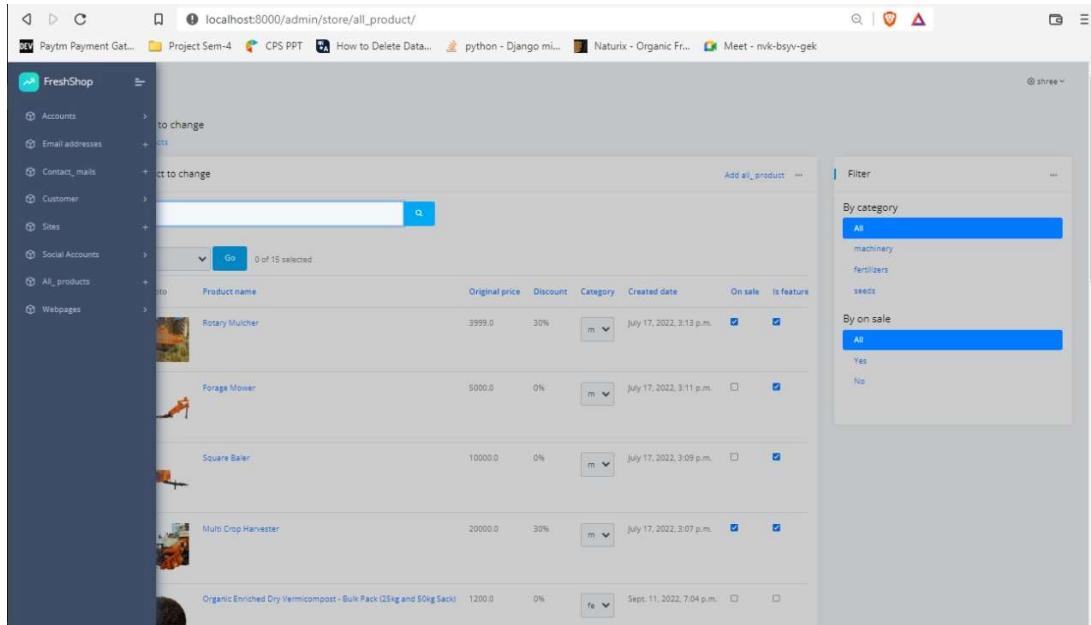


Figure 30 : Admin SideBar Design

Description: This is Admin Side navigation bar, this used redirect or go to another option.

- **Code highlights for the following:**

The screenshot shows a Windows desktop environment with Visual Studio Code running in the foreground. The code editor displays the file `index.html` from a project named "indexhtml". The project structure in the Explorer sidebar includes files like `accounts.html`, `home.html`, `about-us.html`, `author.html`, `contact-us.html`, `index.html`, `page-403.html`, `page-404.html`, `page-500.html`, `sign-in.html`, `sign-up.html`, `template.html`, `ui-catchers-alerts.html`, `ui-catchers-modals.html`, `ui-catchers-tooltips-popovers.html`, `ui-elements-avatars.html`, `ui-elements-badges.html`, `ui-elements-breadcrumbs.html`, `ui-elements-buttons.html`, `ui-elements-dropdowns.html`, `ui-elements-progress-bars.html`, `ui-elements-toggles.html`, `ui-elements-typography.html`, `ui-input-forms.html`, `ui-input-inputs.html`, `ui-navigation-nav-tabs.html`, `ui-navigation-navbars.html`, `ui-navigation-pagination.html`, `ui-sections-features.html`, and `ui-sections-hero-sections.html`. The `index.html` file itself is a Django template for a presentation page, featuring meta tags for mobile devices and various CSS and JS imports.

```
File Edit Selection View Go Run Terminal Help indexhtml - Project - Visual Studio Code

EXPLORER PROJECT
accounts
home
  about-us.html
  author.html
  contact-us.html
  index.html
  page-403.html
  page-404.html
  page-500.html
  sign-in.html
  sign-up.html
  template.html
  ui-catchers-alerts.html
  ui-catchers-modals.html
  ui-catchers-tooltips-popovers.html
  ui-elements-avatars.html
  ui-elements-badges.html
  ui-elements-breadcrumbs.html
  ui-elements-buttons.html
  ui-elements-dropdowns.html
  ui-elements-progress-bars.html
  ui-elements-toggles.html
  ui-elements-typography.html
  ui-input-forms.html
  ui-input-inputs.html
  ui-navigation-nav-tabs.html
  ui-navigation-navbars.html
  ui-navigation-pagination.html
  ui-sections-features.html
  ui-sections-hero-sections.html
includes
OUTLINE
TIMELINE

dj home.html dj login.html urls.py dj index.html settings.py

efarm > templates > admin > home > dj index.html
8 * Coded by www.creative-tim.com
9
10
11
12 * The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
13 <!DOCTYPE HTML>
14 <html lang="en" itemscope itemtype="http://schema.org/WebPage">
15
16 <head>
17   <meta charset="utf-8" />
18   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
19   <link rel="apple-touch-icon" sizes="76x76" href="{{ ASSETS_ROOT }}img/apple-icon.png">
20   <link rel="icon" type="image/png" href="{{ ASSETS_ROOT }}img/favicon.png">
21
22 <title>
23   | Django Material Kit 2 - Presentation | AppSeed
24 </title>
25
26   <!-- Fonts and icons -->
27   <link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700,900|Roboto+Slab:400,700" />
28   <!-- Nucleo Icons -->
29   <link href="{{ ASSETS_ROOT }}/css/nucleo-icons.css" rel="stylesheet" />
30   <link href="{{ ASSETS_ROOT }}/css/nucleo-svg.css" rel="stylesheet" />
31   <!-- Font Awesome Icons -->
32   <script src="https://kit.fontawesome.com/42d5adcbca.js" crossorigin="anonymous"></script>
33   <!-- Material Icons -->
34   <link href="https://fonts.googleapis.com/icon?family=Material+Icons+Round" rel="stylesheet" />
35   <!-- CSS files -->
36   <link id="pagestyle" href="{{ ASSETS_ROOT }}/css/material-kit.css?v=3.0.0" rel="stylesheet" />
37
38 </head>
39 <body class="presentation-page bg-gray-200">
40
41   {% include 'includes/navigation.html' %}
42
43   <header class="header-2">
44     <div class="page-header min-vh-75 relative" style="background-image: url('{{ ASSETS_ROOT }}img/bg2.jpg')">
```

Figure 31 : Admin Sidebar Code

2. Admin Products Page

The screenshot shows the Django Admin interface for managing products. On the left, there's a sidebar with various icons. The main area displays a table of products with columns for ID, Myphoto, Product name, Original price, Discount, Category, Created date, On sale, and Is feature. There are five products listed:

ID	Myphoto	Product name	Original price	Discount	Category	Created date	On sale	Is feature
104		Rotary Mulcher	3999.0	30%	m	July 17, 2022, 3:13 p.m.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
103		Forage Mower	5000.0	0%	m	July 17, 2022, 3:11 p.m.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
102		Square Baler	10000.0	0%	m	July 17, 2022, 3:09 p.m.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
101		Multi Crop Harvester	20000.0	30%	m	July 17, 2022, 3:07 p.m.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13		Organic Enriched Dry Vermicompost - Bulk Pack (25kg and 50kg Sack)	1200.0	0%	fe	Sept. 11, 2022, 7:04 p.m.	<input type="checkbox"/>	<input type="checkbox"/>

On the right, there are two filter panels: 'By category' and 'By on sale'. The 'By category' panel has 'All' selected, with 'machinery', 'fertilizers', and 'seeds' as options. The 'By on sale' panel also has 'All' selected, with 'Yes' and 'No' as options.

Figure 32 : Products Page – Admin

Description: This is Admin Manage product page where admin can manage all products. for instance: Add Products, Remove Products and Update Products.

➤ Code highlights for the following:

The screenshot shows the Visual Studio Code editor with the file 'ui-input-forms.html' open. The code is a template for a product form page. It includes meta tags, a title, and various CSS imports for fonts and icons. The code is well-structured with comments explaining the purpose of each section.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
    <link rel="apple-touch-icon" sizes="76x76" href="{{ ASSETS_ROOT }}img/apple-icon.png" />
    <link rel="icon" type="image/png" href="{{ ASSETS_ROOT }}img/favicon.png" />
    <title> Django Material Kit 2 - Forms | AppSeed </title>
    <!-- Fonts and icons -->
    <link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700,900|Roboto+Slab:400,700" />
    <!-- Nucleo Icons -->
    <link href="{{ ASSETS_ROOT }}/css/nucleo-icons.css" rel="stylesheet" />
    <link href="{{ ASSETS_ROOT }}/css/nucleo-svg.css" rel="stylesheet" />
    <!-- Font Awesome Icons -->
    <script src="https://kit.fontawesome.com/42d5adcbca.js" crossorigin="anonymous"></script>
    <!-- Material Icons -->
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons+Round" rel="stylesheet" />
    <!-- CSS Files -->
    <link id="pagestyle" href="{{ ASSETS_ROOT }}/css/material-kit.css?v=3.0.0" rel="stylesheet" />
  </head>
  <body>
    <div class="app">
      <div class="app-header">
        <div class="app-header-left">
          <div class="app-header-left-item">Home</div>
          <div class="app-header-left-item">About Us</div>
          <div class="app-header-left-item">Author</div>
          <div class="app-header-left-item">Contact</div>
          <div class="app-header-left-item">Index</div>
          <div class="app-header-left-item">Page 403</div>
          <div class="app-header-left-item">Page 404</div>
          <div class="app-header-left-item">Page 500</div>
          <div class="app-header-left-item">Sign In</div>
          <div class="app-header-left-item">Sign Up</div>
          <div class="app-header-left-item">Template</div>
          <div class="app-header-left-item">Ui Catchers Alerts</div>
          <div class="app-header-left-item">Ui Catchers Modals</div>
          <div class="app-header-left-item">Ui Catchers Tooltips Popovers</div>
          <div class="app-header-left-item">Ui Elements Avatars</div>
          <div class="app-header-left-item">Ui Elements Breadcrumbs</div>
          <div class="app-header-left-item">Ui Elements Buttons</div>
          <div class="app-header-left-item">Ui Elements Dropdowns</div>
          <div class="app-header-left-item">Ui Elements Progress Bars</div>
          <div class="app-header-left-item">Ui Elements Toggles</div>
          <div class="app-header-left-item">Ui Elements Typography</div>
        </div>
        <div class="app-header-right">
          <div class="app-header-right-item">Logout</div>
        </div>
      </div>
      <div class="app-content">
        <div class="app-content-left">
          <div class="app-content-left-item">About Us</div>
          <div class="app-content-left-item">Author</div>
          <div class="app-content-left-item">Contact</div>
          <div class="app-content-left-item">Index</div>
          <div class="app-content-left-item">Page 403</div>
          <div class="app-content-left-item">Page 404</div>
          <div class="app-content-left-item">Page 500</div>
          <div class="app-content-left-item">Sign In</div>
          <div class="app-content-left-item">Sign Up</div>
          <div class="app-content-left-item">Template</div>
          <div class="app-content-left-item">Ui Catchers Alerts</div>
          <div class="app-content-left-item">Ui Catchers Modals</div>
          <div class="app-content-left-item">Ui Catchers Tooltips Popovers</div>
          <div class="app-content-left-item">Ui Elements Avatars</div>
          <div class="app-content-left-item">Ui Elements Breadcrumbs</div>
          <div class="app-content-left-item">Ui Elements Buttons</div>
          <div class="app-content-left-item">Ui Elements Dropdowns</div>
          <div class="app-content-left-item">Ui Elements Progress Bars</div>
          <div class="app-content-left-item">Ui Elements Toggles</div>
          <div class="app-content-left-item">Ui Elements Typography</div>
        </div>
        <div class="app-content-right">
          <div class="app-content-right-item">About Us</div>
          <div class="app-content-right-item">Author</div>
          <div class="app-content-right-item">Contact</div>
          <div class="app-content-right-item">Index</div>
          <div class="app-content-right-item">Page 403</div>
          <div class="app-content-right-item">Page 404</div>
          <div class="app-content-right-item">Page 500</div>
          <div class="app-content-right-item">Sign In</div>
          <div class="app-content-right-item">Sign Up</div>
          <div class="app-content-right-item">Template</div>
          <div class="app-content-right-item">Ui Catchers Alerts</div>
          <div class="app-content-right-item">Ui Catchers Modals</div>
          <div class="app-content-right-item">Ui Catchers Tooltips Popovers</div>
          <div class="app-content-right-item">Ui Elements Avatars</div>
          <div class="app-content-right-item">Ui Elements Breadcrumbs</div>
          <div class="app-content-right-item">Ui Elements Buttons</div>
          <div class="app-content-right-item">Ui Elements Dropdowns</div>
          <div class="app-content-right-item">Ui Elements Progress Bars</div>
          <div class="app-content-right-item">Ui Elements Toggles</div>
          <div class="app-content-right-item">Ui Elements Typography</div>
        </div>
      </div>
    </div>
  </body>

```

Figure 33 : Product Page Code - Admin

11.4. Development Phase – 4

1. Paytm Payment Page

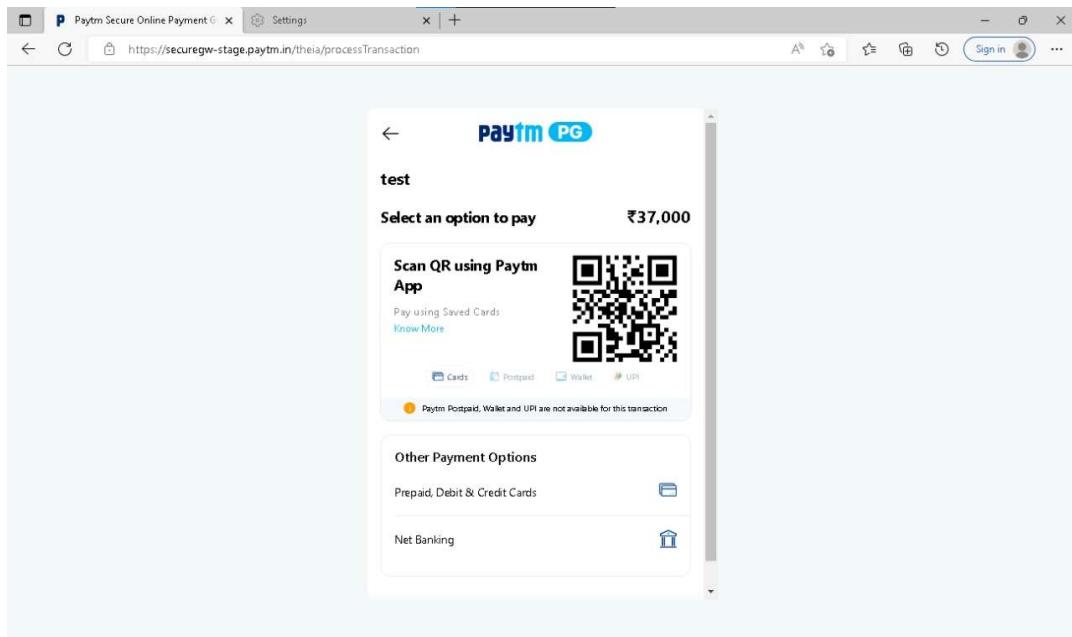


Figure 34 : Paytm Payment Design

Description: This is Paytm Paytm Page Where users can do online payment.

➤ Code highlights for the following:

```

File Edit Selection View Go Run Terminal Help
paytm_process.py - Project - Visual Studio Code
EXPLORER PROJECT
> django-admin-volt-main
> Docs
efarm
> accounts
> contact_us
customer
> migrations
> static
  __init__.py
  admin.py
  apps.py
  context_processors.py
  models.py
  paytm_process.py
  paytm.py
  tests.py
  urls.py
  views.py
efarm
  __init__.py
  asgi.py
  settings.py
  urls.py
  wsgi.py
OUTLINE
TIMELINE
Ln 16, Col 45  Spaces:4  UTF-8  CRLF  Python  3.10.4 64-bit  Go Live  Prettier
settings.py  paytm_process.py  urls.py  login.html  context_processors.py  nav.html
paytm_process.py
from django.conf import settings
from .models import *
from django.contrib import messages
import datetime
def paytm(request):
    customer = request.user.customer
    order = Order.objects.get(customer=customer, order_status='Pending', complete=False)
    print(order)
    print("Hello")
    # user = request.user
    transaction = Transaction.objects.create(made_by=customer, order_id=order, amount=order.get_cart_total, res
    transaction.save()
    print("First Time Save", transaction.transaction_id)
    merchant_key = settings.PAYTM_SECRET_KEY
    params = (
        ('MID', settings.PAYTM_MERCHANT_ID),
        ('ORDER ID', str(transaction.transaction_id)),
        ('CUST_ID', str(transaction.made_by.email)),
        ('TXN_AMOUNT', str(transaction.amount)),
        ('CHANNEL_ID', settings.PAYTM_CHANNEL_ID),
        ('WEBSITE', settings.PAYTM_WEBSITE),
        # ('EMAIL', request.user.email),
        # ('MOBILE NO', '9911223388'),
        ('INDUSTRY TYPE ID', settings.PAYTM_INDUSTRY_TYPE_ID),
        ('CALLBACK URL', 'http://127.0.0.1:8000/customer/callback/'),
        # ('CALLBACK_URL', 'https://fe1f-203-109-79-189.ngrok.io/customer/callback/'),
        # ('PAYMENT_MODE_ONLY', 'NO'),
    )
    paytm_params = dict(params)
    checksum = generate_checksum(paytm_params, merchant_key)
    print(checksum)

```

Figure 35 : Paytm Payment Code

2. Order Confirm Email Template

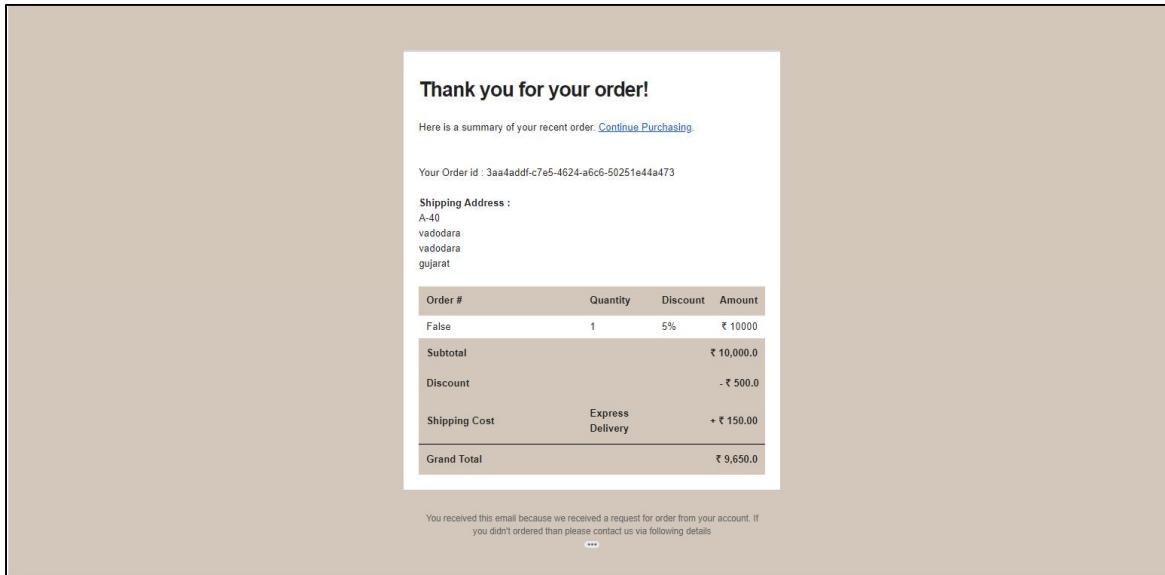


Figure 36 : Order Confirm Email Template Design

Description: This is Email Template used to send Confirmation order mail.

➤ Code highlights for the following:

```

File Edit Selection View Go Run Terminal Help
PROJECT
> django-admin-volt-main
> Docs
efarm
> accounts
> contact_us
> customer
efarm
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
media
static
store
templates
accounts
admin
dashboard
emails
discount.html
invoice.html
otp.html
return_order.html
try.html
inc
payment
store
templates
user-dashboard
OUTLINE
TIMELINE
invoice.html - Project - Visual Studio Code
efarm > templates > emails > invoice.html
1 1<!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta http-equiv="x-ua-compatible" content="ie=edge">
6     <title>Email Receipt</title>
7     <meta name="viewport" content="width=device-width, initial-scale=1">
8     <style type="text/css">
9       /*
10      * Google webfonts. Recommended to include the .woff version for cross-client compatibility.
11      */
12      @media screen {
13        @font-face {
14          font-family: 'Source Sans Pro';
15          font-style: normal;
16          font-weight: 400;
17          src: local('Source Sans Pro Regular'), local('SourceSansPro-Regular'), url(https://fonts.gstatic.com/s/sourcesanspro/v10/0f
18        }
19        @font-face {
20          font-family: 'Source Sans Pro';
21          font-style: normal;
22          font-weight: 700;
23          src: local('Source Sans Pro Bold'), local('SourceSansPro-Bold'), url(https://fonts.gstatic.com/s/sourcesanspro/v10/toadOcf
24      }
25      /*
26      * Avoid browser level font resizing.
27      * 1. Windows Mobile
28      * 2. iOS / OSX
29      */
30      body,
31      table,
32      tr,
33      td {
34        font-size: 1em;
35        line-height: 1.4;
36        color: #333;
37        font-family: 'Source Sans Pro', sans-serif;
38      }
39    
```

Figure 37 : Order Confirm Email Template Code

11.5. Validation

1. Change password validations

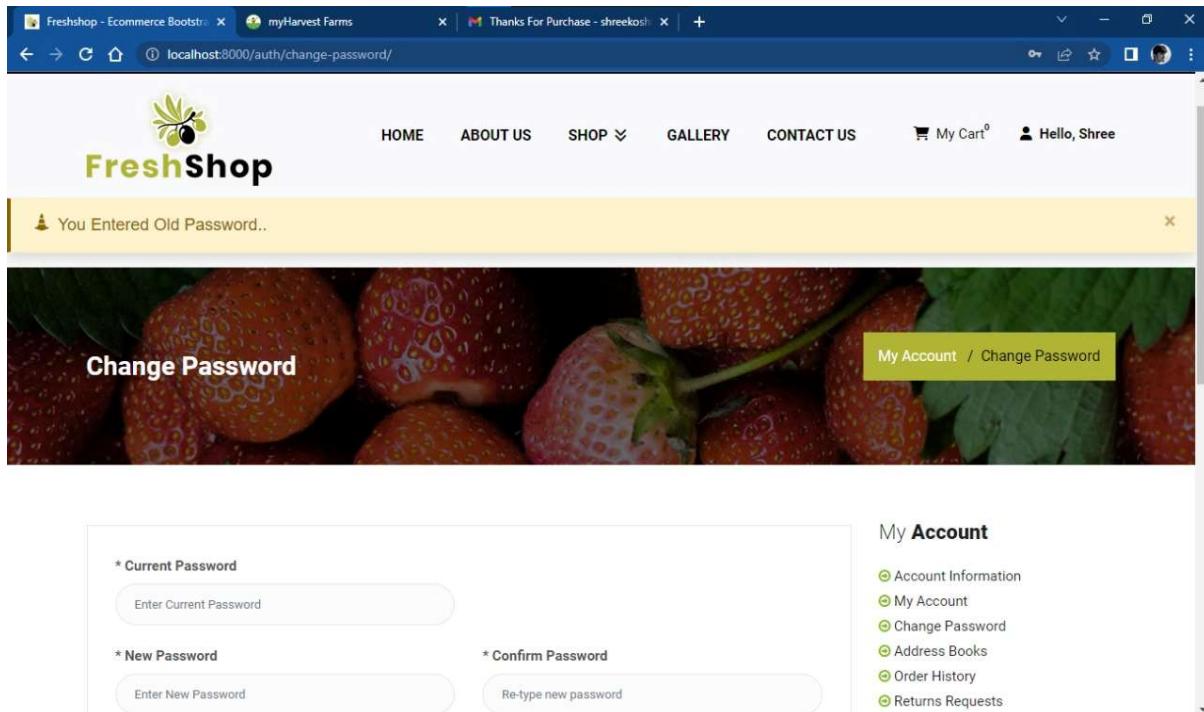


Figure 38 : Change password Validations Page

2. Edit profile validation

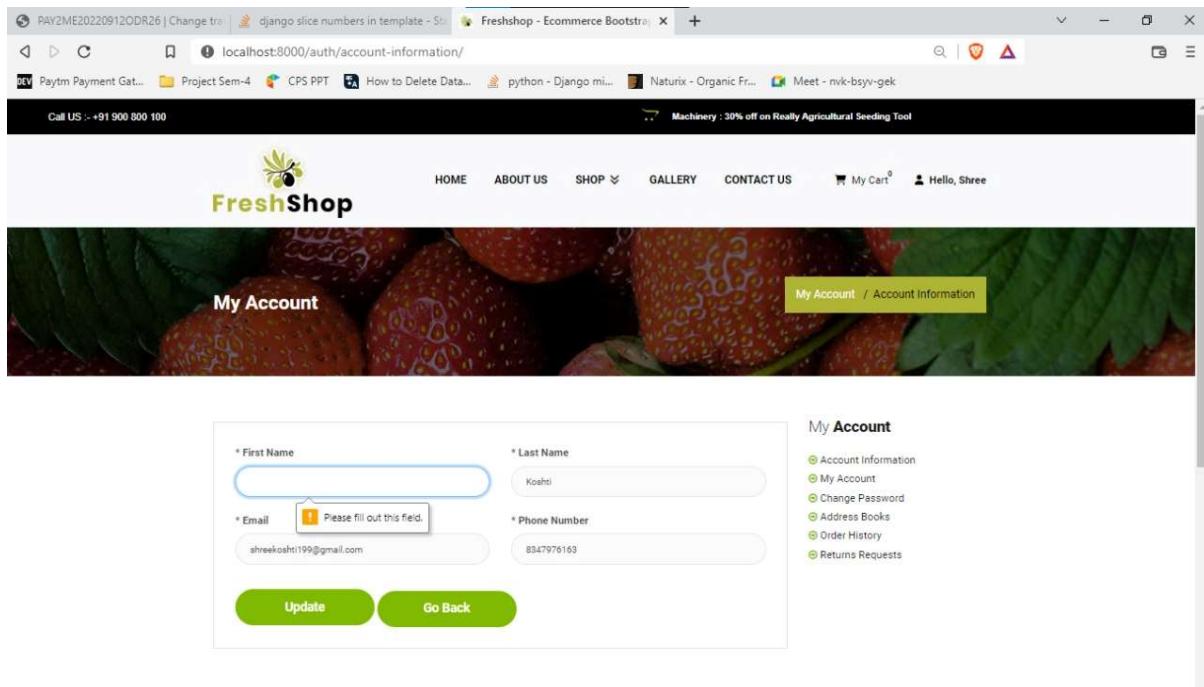


Figure 39 : Edit Profile page Validations

3. Login Validations

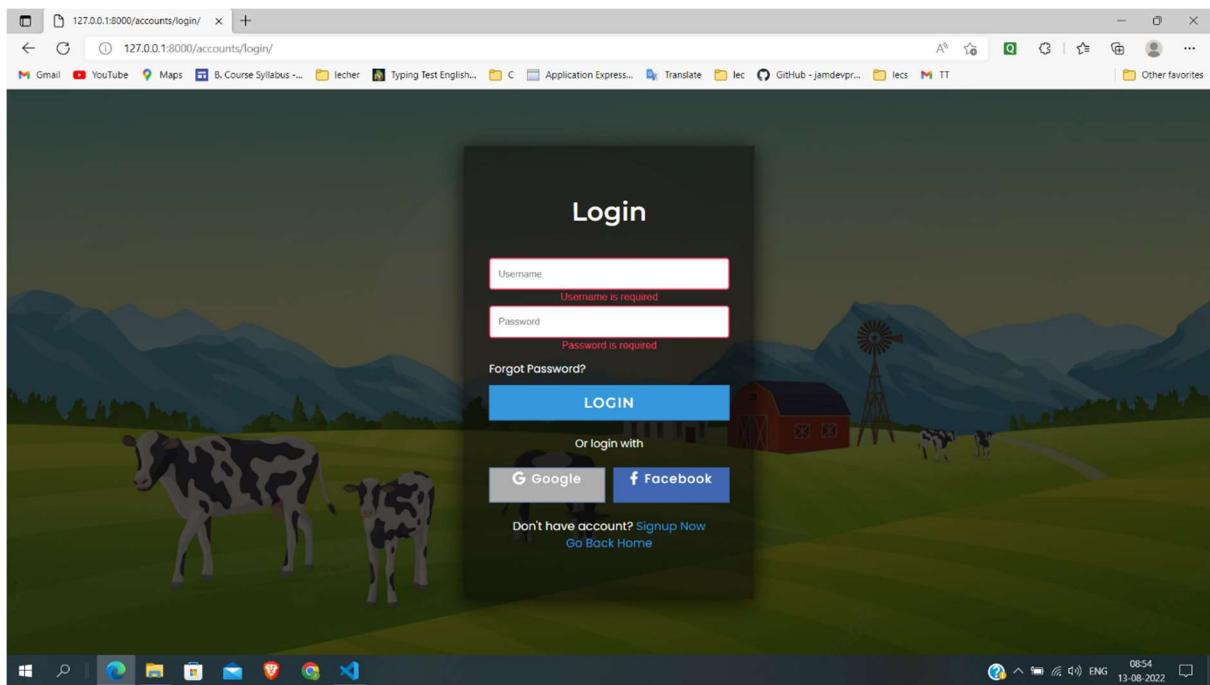


Figure 40 : Login Page Validations

12.What is testing?

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases. The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring the specification, functionality, and performance of a software program or application.

12.1. Importance and types of testing

To understand the importance of software testing, consider the example of Starbucks. In 2015, the company lost millions of dollars in sales when its point-of-sale (POS) platform shut down due to a faulty system refresh caused by a software glitch. This could have been avoided if the POS software had been tested thoroughly. Nissan also suffered a similar fate in 2016 when it recalled more than 3 million cars due to a software issue in airbag sensor detectors.

The following are important reasons why software testing techniques should be incorporated into application development:

- Identifies defects early
- Improves product quality
- Increases customer trust and satisfaction
- Detects security vulnerabilities
- Helps with scalability
- Saves money

12.2. Types of testing

➤ Unit Testing

It focuses on the smallest unit of software design. In this, we test an individual unit or group of interrelated units. It is often done by the programmer by using sample input and observing its corresponding outputs.

Example:

- In a program we are checking if the loop, method, or function is working fine
- Misunderstood or incorrect, arithmetic precedence.
- Incorrect initialization

➤ Integration Testing

The objective is to take unit-tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components is combined to produce output.

Integration testing is of four types:

- Top-down
- Bottom-up
- Sandwich
- Big-Bang

Example:

- (a) Black Box testing: - It is used for validation. In this, we ignore internal working mechanisms and focus on what is the output?
- (b) White box testing: - It is used for verification. In this, we focus on internal mechanisms i.e. how the output is achieved?

➤ **Regression Testing**

Every time a new module is added leads to changes in the program. This type of testing makes sure that the whole component works properly even after adding components to the complete program.

Example: In school, record suppose we have module staff, students and finance combining these modules and checking if on integration of these modules works fine in regression testing.

➤ **Smoke Testing**

This test is done to make sure that the software under testing is ready or stable for further testing. It is called a smoke test as the testing of an initial pass is done to check if it did not catch the fire or smoke in the initial switch on.

Example: If the project has 2 modules so before going to the module make sure that module 1 works properly.

13.Future Enhancement

- Delivery status will be more efficient at user side.
- More payment modes will be add Ex, UPI ,COD.
- Delivery status will be updated by delivery boy at mobile location.
- Make more user-friendly UI.
- More enhanced admin side.ssss
- In future we will create android as well as IOS mobile application.

14. References & Bibliography

➤ **Website:**

1. <https://docs.djangoproject.com/en/3.2/>
2. <https://www.python.org/doc/>
3. <https://stackoverflow.com/>

➤ **Book :**

1. Django for Beginners: Build websites with Python and Django
(William S Vincent)
2. Think Python : How to Think Like a Computer Scientist
(Allen Downey)
3. The Elements of Uml (tm) 2.0 Style
(Scott W. Ambler)