Project Report of

Shahin Nasir Sayyed

(ITC41), (MCA)

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management

ASMA Institute Of Management College
Shivane Pune 411023

CERTIFICATE

This is to certify that **Miss. Shahin Nasir Sayyed** is a bonafide student of ASMA Institute of Management, Pune-23, has successfully completed the project work in the Partial Fulfillment of Master of Computer Application (MCA) Program as per guidelines provided by the Savitribai Phule Pune University for Academic Year 2023-24

Prof. Sarika Jadhav

Project Guide

Prof. Prakash Patil

HOD

Dr. Tannu Ganesh

Director

PSD Shastri Educational Foundation's
ASMA Institute of Management

External Examiner:

Internal Examiner:

Date: 28 / 05 /2024

Place: Pune

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management

Declaration

To,

The Principal,

ASMA Institute of Management

Shivane, Pune-23

Respected Sir,

    I, Miss **Shahin Nasir Sayyed** hereby declare that the Project entitled " **SHOP IT** " has been developed and submitted under the guidance of Prof. Sarika Jadhav is my original work.

    The Project work presented here is my own work and has not been duplicated from any other sources.

Date: /   /2024

Place : Pune

Yours Sincerely

Shahin Nasir Sayyed

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management

Acknowledgement

I would take this opportunity to thank Tannu Ganesh Sir, Director of ASMA for their extended support to me throughout the course.

I would like to thank Prof. Prakash Patil sir, HOD, MCA Department for scholarly disposition, timely guidance, support and cooperation.

I would like to thank Prof. Sarika Jadhav for his/her kind guidance, keen interest, continuous encouragement and inspiration throughout the project work.
Finally, I am grateful to all the staff members of ASMA for their cooperation and support.

I also thankful to get constant encouragement, support and guidance from all teaching and non-teaching staff for their support which helped us in successfully completion of our project work.

Miss. **Shahin Nasir Sayyed**
Exam Seat No :- 2320

Shastri Educational Foundation's
ASMA Institute of Management

Aimbreakthrough Pvt Ltd.
42/1 Mahindrakar Complex, 2nd Floor
NDA Road, Warje Malwadi
Pune 411058

Date: 02/12/2023

Internship Letter

Ms. Shahin Nasir Sayyed student of MCA, ASMA Institute of Management, Pune is working in this organization as an intern from 31st October 2023. She will be working in this organization till 30th April 2024.

She will be learning web application and related technologies during this period. She will follow the rules and regulations of the company during her work in this organization. All the work that she has worked will remain the property of Aimbreakthrough.

For Aimbreakthrough Pvt. Ltd.

Ashwin Bendre
(Director)

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management

INDEX

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management

INTRODUCTION

Shop It is the buying and selling of goods and services, or the transmitting of funds or data, over an electronic network, primarily the internet. These business transactions occur either as business to-business, business-to-consumer, consumer-to-consumer or consumer-to-business. E-commerce shops have become part of our daily lives. Technological advancement has made it possible for people to sit in the convenience of their homes and still shop online without going to a physical shop.

This is a project chosen with the objective to develop a basic web Application where a consumer is provided with a shopping cart application and also to know about the technologies used to develop such an application

1.1 Institute Profile

The P. S. D. Shastri Education Foundation (P.S.D.S.E.F.) presently has 7 institutes with individual infrastructure and additional hostels, harbouring constructive platform and conducive environment for learning. All the institutes have a highly modern infrastructure which is regularly upgraded. These modernised learning institutes also have the recognition of relevant statutory bodies. Every norm and standard has been thoroughly met with by every institute.

ASMA (approved by AICTE, New Delhi, affiliated to University of Pune and recognized by Government of Maharashtra), the sky

1

Depicts ETERNITY, INFINITY and VASTNESS. When captured in the realm of knowledge.... new horizons, new visions and new frontiers open up for aspiring and deserving students. At our Management Institute, the primary objective and focus is to provide a strong platform for students to groom into highly talented and trained professionals.

By bench-marking high standards of quality of both faculty and students; excellent state-of-the-art infrastructure; the physical ambience of the Institute, all accelerates to the complete development of each student. Our stringent interdisciplinary learning environment lends tremendous support to our mission of grooming young talent to match and meet the expectations of the ever increasing demands of the corporate world.

1.2 Abstract

At Internet-based enterprise that sells books, music, movies, housewares, electronics, toys, and many other goods, either directly or as the middleman between other retailers and Shop It millions of customers.

If user is a supplier, then it is a best solution for his/her to add their products on the portal and increase their sale. The supplier can sale all his products at a single location.

This application provides easy and safe payment option to user.
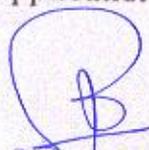
In order to facilitate online purchase a shopping cart is provided to the user.

After selection of the items, he is forwarded to Payment Gateway process.

Project includes Python, Django, HTML, CSS, Angular

1.3 Existing System

There is many software available in market which are having few same contents.

But these systems are not providing any option to sale the product on that same software.

Another disadvantage is if user want any security for Payment there is no any security providing system for payment.

Need For the Proposed System:

The online shopping is an easy to maintain, ready to run, scalable, affordable and reliable cost saving tool from Software Associates suited for small, medium, and large shopping complex and shopping malls.

Features and Benefits:
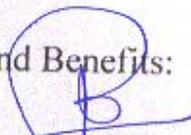
- Providing security

- Low cost

The proposed system can be used even by the naïve users and it does not require any educational level, experience, and technical expertise in computer field but it will be of good.

1.4 Scope of System

Any person can register as a customer by providing their details. A registered customer can order multiple products by using the application on the development of this product; a customer will be able to order electronics online within few minutes without the necessity of physically going to the store and wasting time during the process. After placing the order, the product will be delivered to the given address requested, by the user.

The system generates types of information that can be used for various purposes.

- It satisfies the user requirement
- Be easy to understand by the user and operator
- Have a good user interface
- Delivered on schedule within the budget
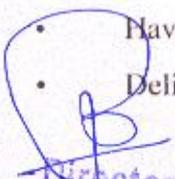
Consumer will be able to browse and search product under different categories. Selected product or item selected for purchase would be added into the virtual shopping cart.

1.5 Operating Environment

Processor: Pentium IV or Above

RAM: 2GB or above

Hard Disk: 50GB or above

Input Devices: Keyboard, Mouse

Output Devices: Monitor

Software Required:

Operating System: Linux, Ubuntu, Mac, Windows XP, 7, 8, 8.1, 10

Frontend: Python, HTML, CSS, Bootstrap, JavaScript

Backend: Django-SQlite3, JS

Local host: http://127.0.0.1:8000/

1.6 Brief Description of Technology Used

- Django – Used for creation REST API.

- Angular – Used for front end development.

- HTML. CSS – For creation Designing and View making

- Python – For Scripting

- Database – SQL (Heidi SQL)

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management

Chapter 2. Proposed System:

The proposed system consists of the registration module for the user. The registered user gets the accessibility and to purchase and buy the any type of electronics products. This system contains of following features:

1. Seller
2. Customer
3. Sign in
4. Sign up
5. View Profile
6. Edit Profile
7. Cart
8. Add Product
9. Logout

2.1 Study of Similar Systems

- There are many software systems available in the market that does not provide both B2B and B2C system.

- This system does not provide online quote submission and cart for B2B.

2.2   Feasibility Study

Technical feasibility:

Technical feasibility involves how difficult will be to build and whether the firm has enough experience using that technology.

Operational feasibility

Is a measure of how well a proposed system solves the problem user satisfies the requirements.

Scheduler feasibility

Typically, this name estimating how long system will take to develop.

2.3   Objectives of Proposed System

We are providing a software which is user friendly and useful in daily life. With the help of this user can easily buy or sale the product in a single location.

Advantages of Proposed System

- Manage supplier and customer details
- Category wise search available
- Supplier can add and manage product details

- Cart option is available
- Gov Links are available
- Online Payment is acceptable

Users of System

Business - Buying and selling as a way of earning money: commerce.

Vendors – The biggest difference between Amazon sellers and vendors is who's actually selling the product. Seller's list, price and market their products themselves.

Manufactures - With the help of E-Commerce distributors and manufacturers can measure and analyse the effectiveness of sales, campaigns for marketing, inventory turns, the effectiveness of customer sales, and engagement of customers.

Customers - Ecommerce customer service is how online businesses provide assistance to customers with everything from making online purchase decisions to resolving issues

Non – Functional Requirements:

Non-functional requirements define system properties and constraints. It arises through user needs, because of budget constraints or organizational policies, or due to the external factors such as safety regulations, privacy registration and so on. Nonfunctional requirements are:

- Security
- Reliability
- Maintainability
- Portability
- Application Compatibility

3. Entity Relationship Diagram (ERD)
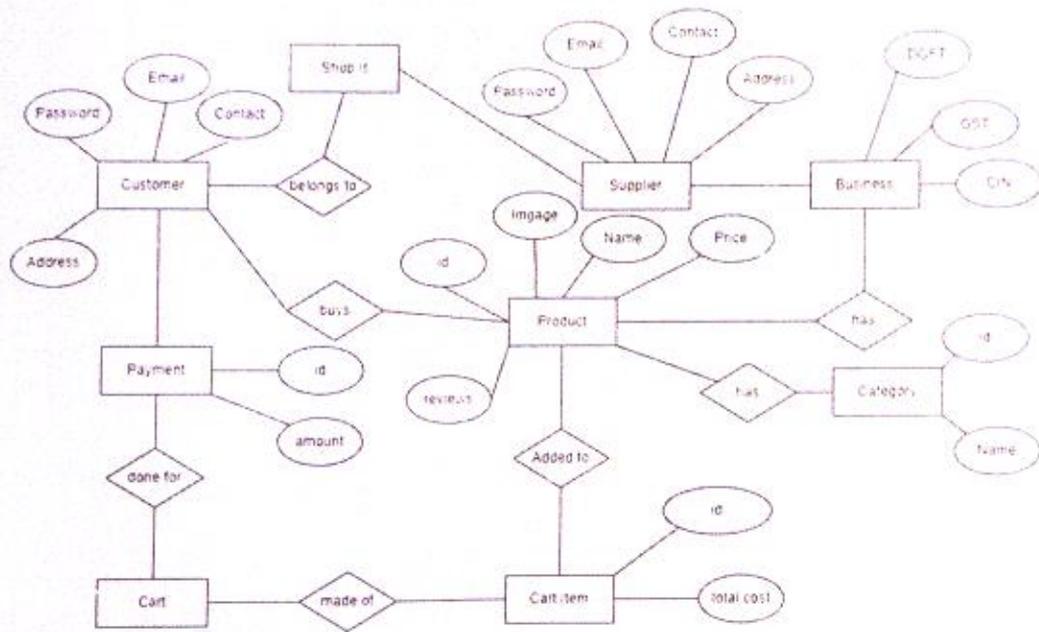
3.2 Table Structure

User table

id	INT (PK)

password	VARCHAR (128)
last_login	DATE
is_superuser	TINYINT (1)
username	VARCHAR (150)
is_staff	TINYNT (1)
date_joined	DATETIME (6)

Customer Table

id	INT (PK)
full_name	VARCHAR (50)
email	VARCHAR (255)
mobile	VARCHAR (12)
address	VARCHAR (150)
taluka	VARCHAR (50)
post	VARCHAR (6)
district	VARCHAR (50)

Supplier Table

id	INT (PK)
full_name	VARCHAR (50)

mobile	VARCHAR (12)
address	VARCHAR (150)
taluka	VARCHAR (50)
post	VARCHAR (6)
district	VARCHAR (50)
email	VARCHAR (255)
phone	VARCHAR (12)
city	VARCHAR (50)
designation	VARCHAR (50)
email_optional	VARCHAR (255)

Business table

id	INT (PK)
name	VARCHAR (50)
organisation_type	VARCHAR (20)
ownership_type	VARCHAR (15)
business_type	VARCHAR (15)
business_email	VARCHAR (255)
GST	VARCHAR (15)
PAN	VARCHAR (10)
supplier_id	INT (FK)

Product Table

id	INT (PK)
name	VARCHAR (50)
price	INT (10)
arrival	TINYINT (1)
unit	VARCHAR (5)
brand	VARCHAR (50)
code	INT (10)
offer	TINYINT (1)
additional_information	VARCHAR (50)
category_id	INT (FK)
business_id	INT (FK)

Order table

id	INT (PK)
Quantity	INT (5)
requirement	VARCHAR (50)
(FK) customer_id	INT
product_id	INT (FK)

Quote table

id	INT (PK)
quantity	INT (5)

supplier_id	
	INT (FK)
customer_id	
	INT (FK)

Auth Token table

id	
	INT (PK)
created	
	DATETIME
user_id	
	INT (FK)

Product Category table

id	INT (PK)
name	VARCHAR (50)

Auth Group Table

id	INT (PK)
name	VARCHAR (50)

User Groups

Column	Data Type
id	INT (PK)
user_id	INT (UNIQUE) (FK)
group_id	INT (UNIQUE) (FK)

3.4 Use Case Diagrams
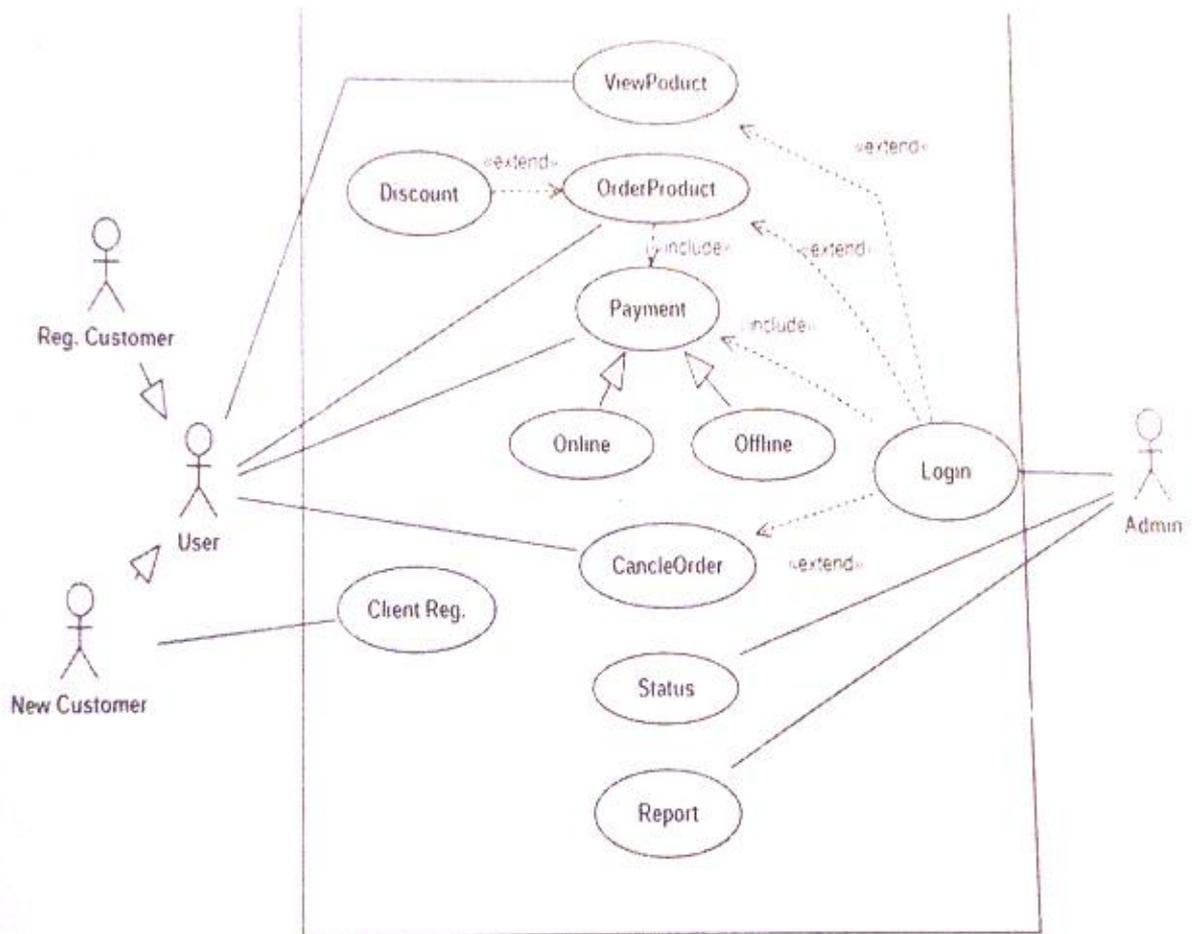
3.5 Class Diagrams

Web User

login_id: String {id}
password: String
state: UserState

Customer

id: String {id}
address: Address
phone: Phone
email: String

«enumeration» UserState

New
Active
Blocked
Banned

Payment

id: String {id}
paid: Date
total: Real
details: String

Account

id: String {id}
billing_address: Address
is_closed: Boolean
open: Date
closed: Date

Shopping Cart

created: Date

Order

number: String {id}
ordered: Date
shipped: Date
ship_to: Address
status: OrderStatus
total: Real

LineItem

quantity: Integer
price: Price

«enumeration» OrderStatus

New
Hold
Shipped
Delivered
Closed

Product

id: String {id}
name: String
supplier: Supplier

© uml-diagrams.or
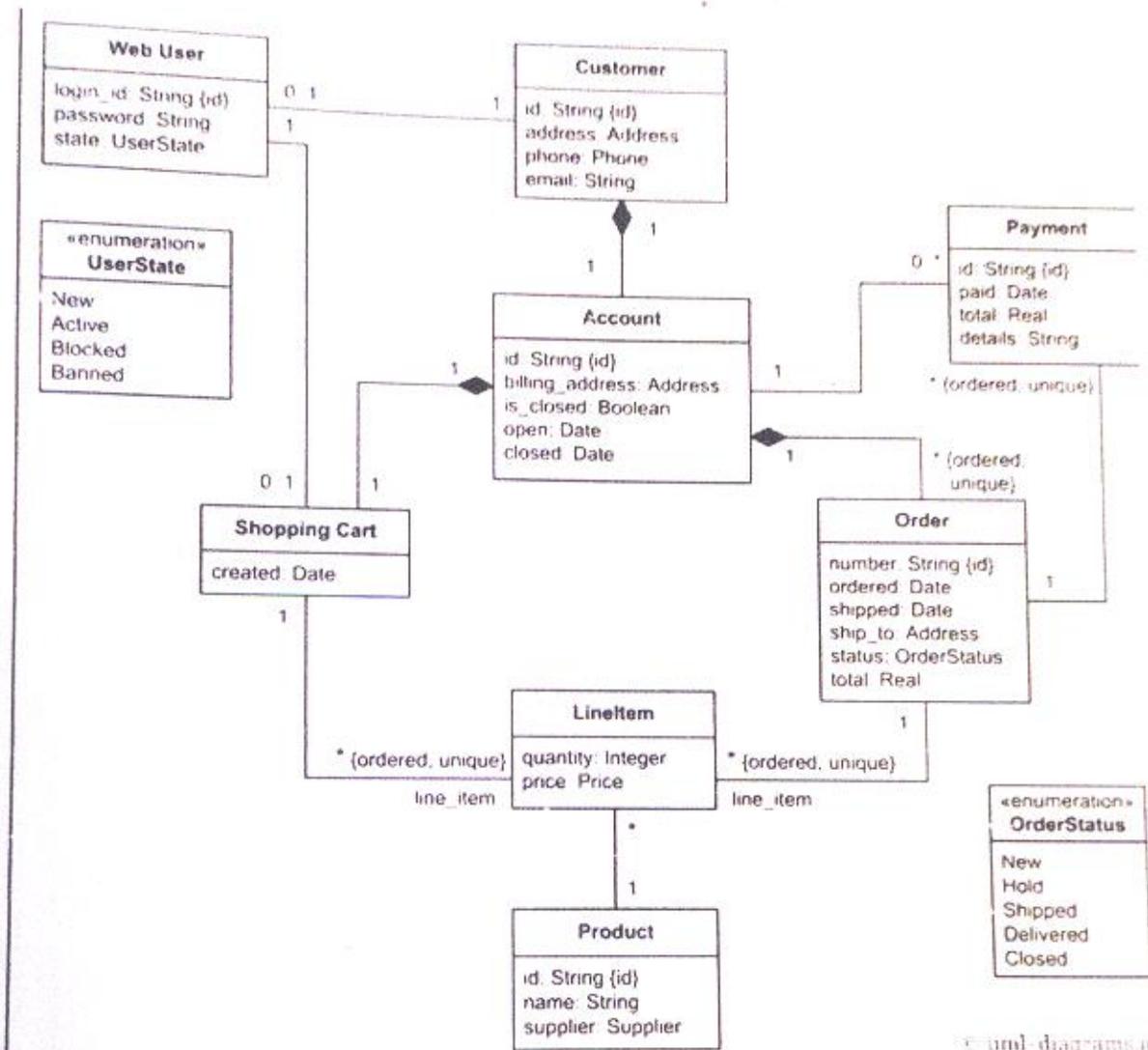
3.6 Activity Diagram
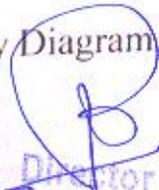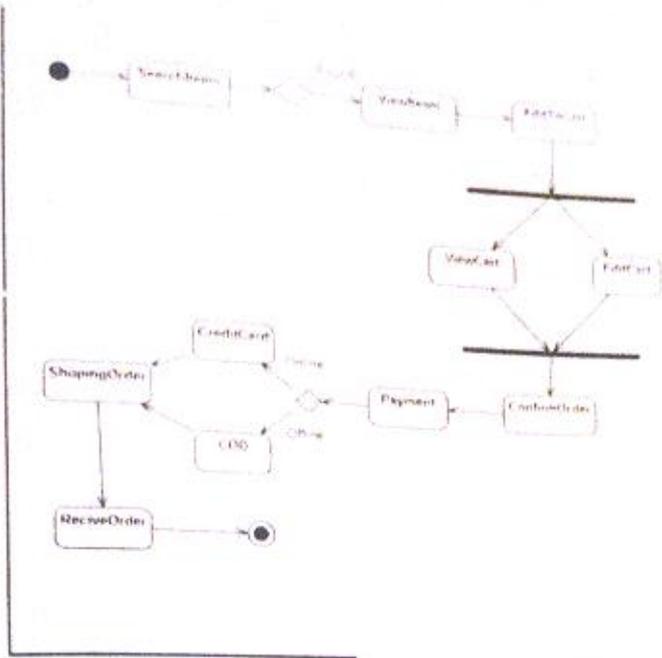
3.7 Deployment Diagram

Deployment Diagram For a Online Shopping System
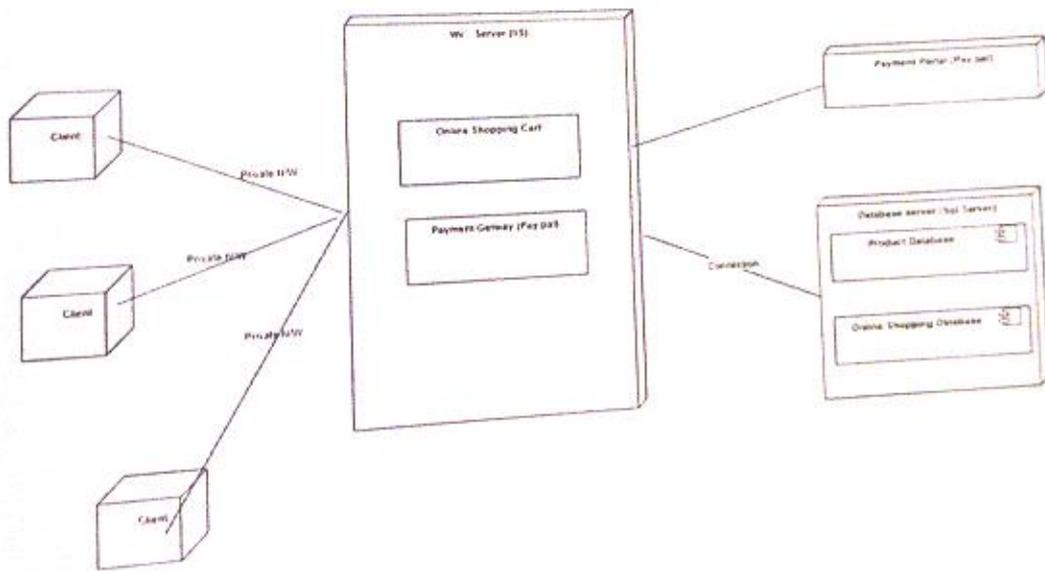
3.8 Module Hierarchy Diagram

```
                              E-Commerce
                                   │
                 ┌─────────────────┴─────────────────┐
                 │                                     │
        Customer Services                    Administrator Services
                 │                                     │
   ┌──────┬──────┴──────┬──────────┐                  │
   │      │             │          │             Add/Edit
Browse  Shopping     Customer   New User         Inventory
   │      Cart       Authentication Registration      │
   │       │             │          │          ┌──────┴──────┐
Search  ┌──┴──┐      User Profile  Create User  Add New Book  Update Inventory
   │  Create  Add/delete │         Profile
   │  Shopping  Items     │
Read   Cart          Purchase History
Reviews                   │
   │                  Purchase
Write                     │
Reviews         ┌─────────┴─────────┐
           Shipping_Details    Place Order
                                     │
                               Shipping_Details
                                     │
                                     │        Update Purchase
                                     │          History
                                CheckOut
                                             Update Number
                                             of Books in
                                             Inventory
```
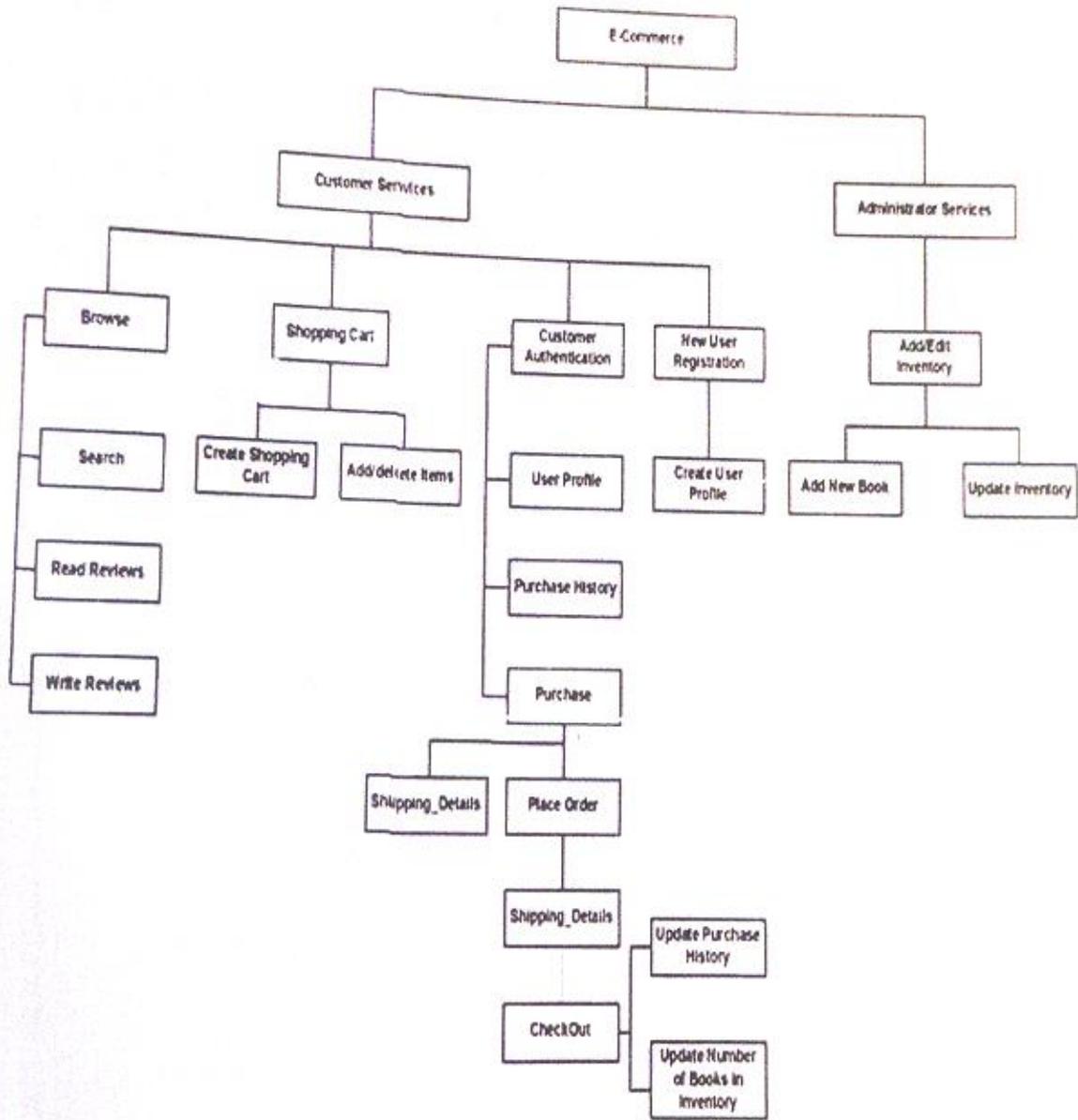
Sample Input and Output Screens

20

User Side Screens

Supplier Sign Up
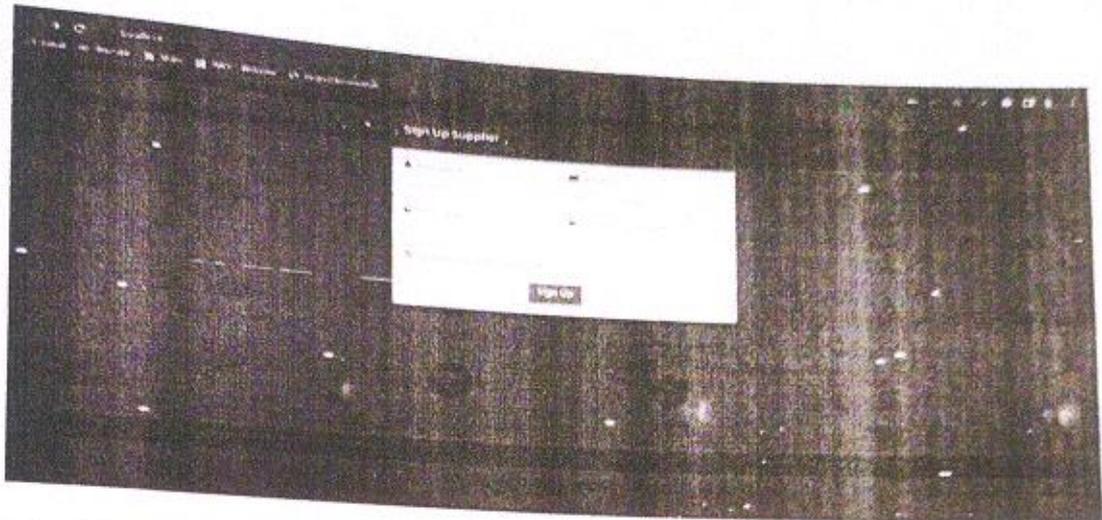
Supplier Sign In

Admin login

Add or Update:

Customer Sign in

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management

22
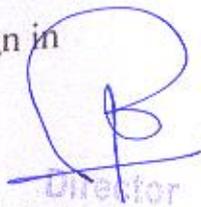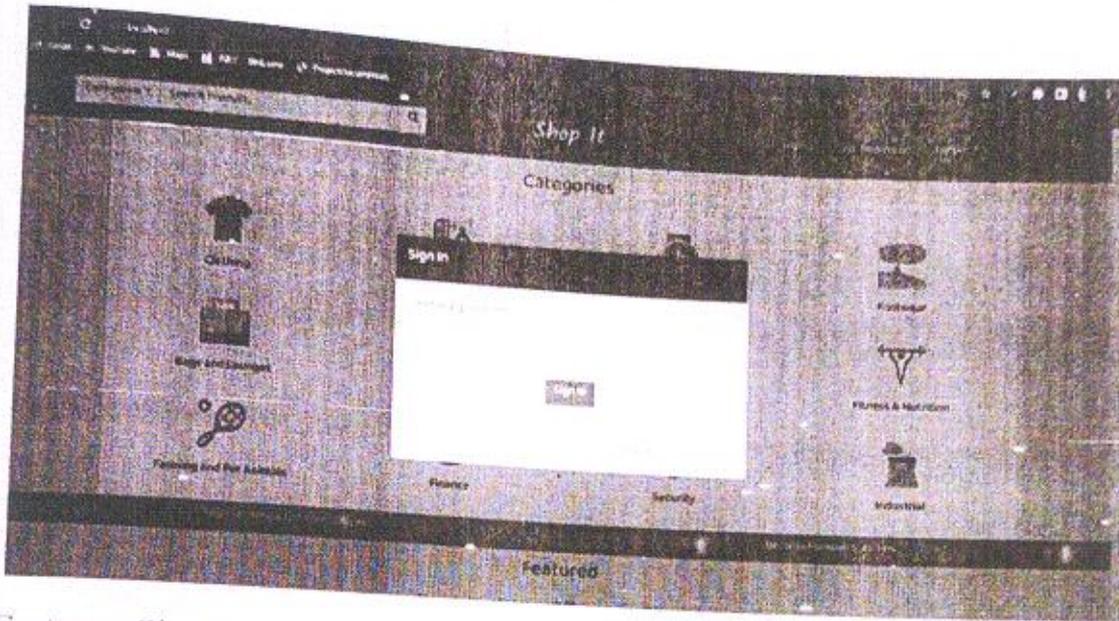
Customer Sign Up
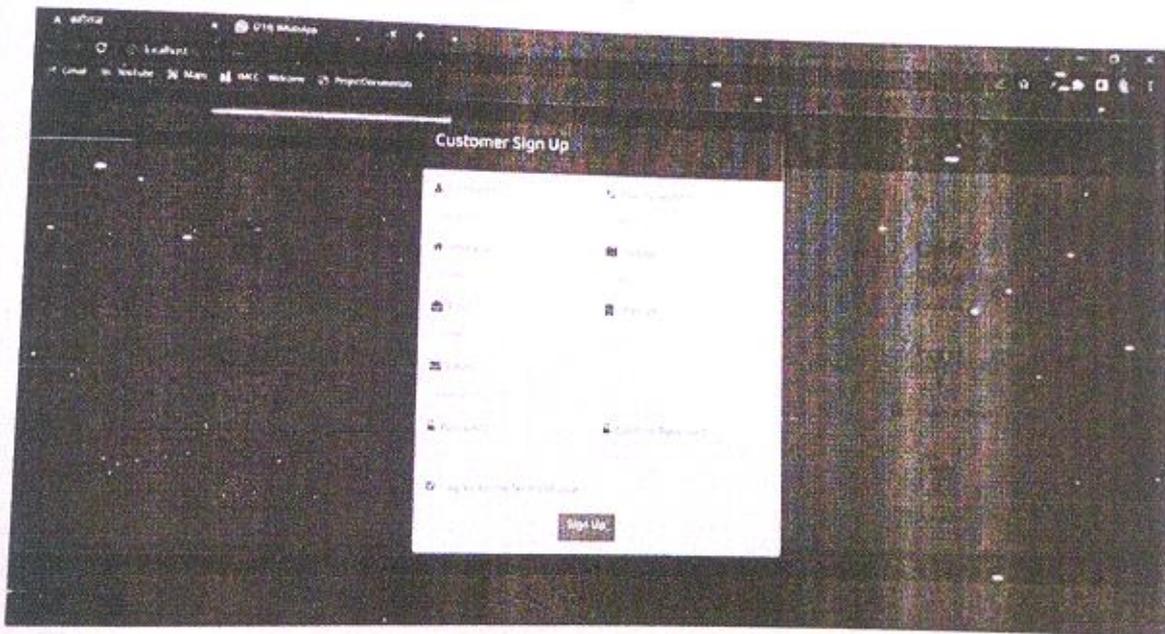
Contact Supplier

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management
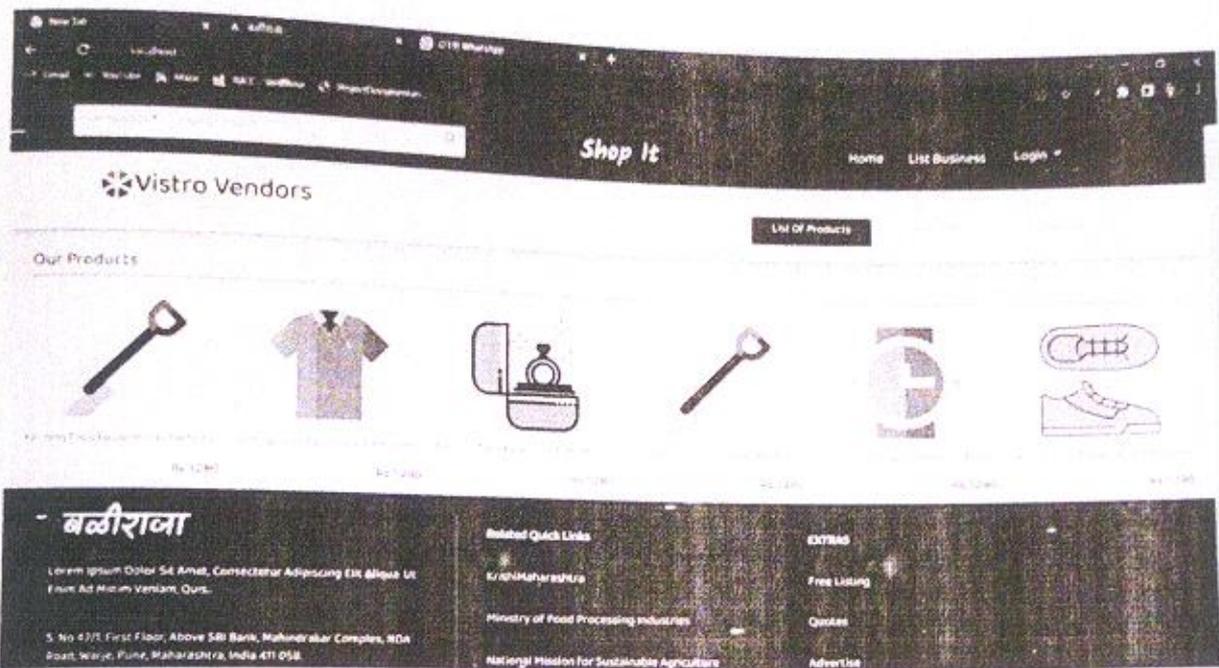
23

✳ Vistro Vendors

List Of Products

Our Products

बळीराजा

Lorem Ipsum Dolor Sit Amet, Consectetur Adipiscing Elit aliqua Ut
Enim Ad Minim Veniam, Quis.

S. No 47/1, First Floor, Above SBI Bank, Mahindrakar Complex, NDA
Road, Warje, Pune, Maharashtra, India 411 058.

Related Quick Links

KrishiMaharashtra

Ministry of Food Processing Industries

National Mission for Sustainable Agriculture

EXTRAS

Free Listing

Quotes

Advertise

Supplier Detail

Shop It

✳ Vistro Vendors

About Us

About Us

Basic Information

Business Details
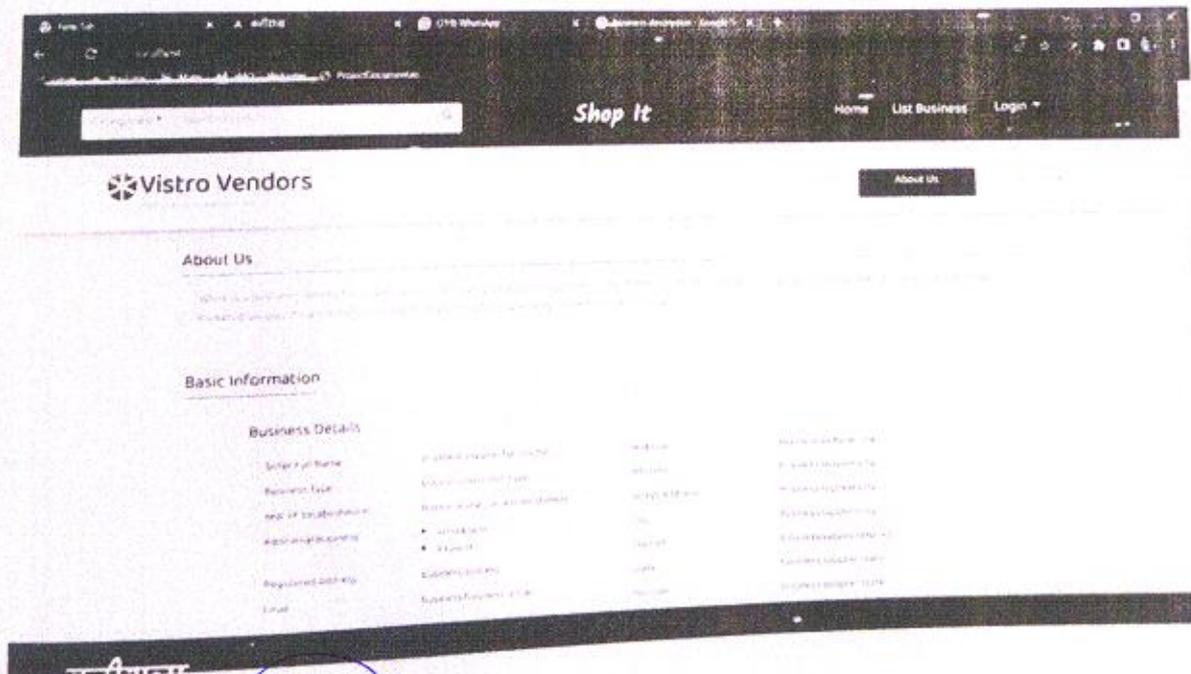
PSD Shastri Educational Foundation's
ASMA Institute of Management
Director
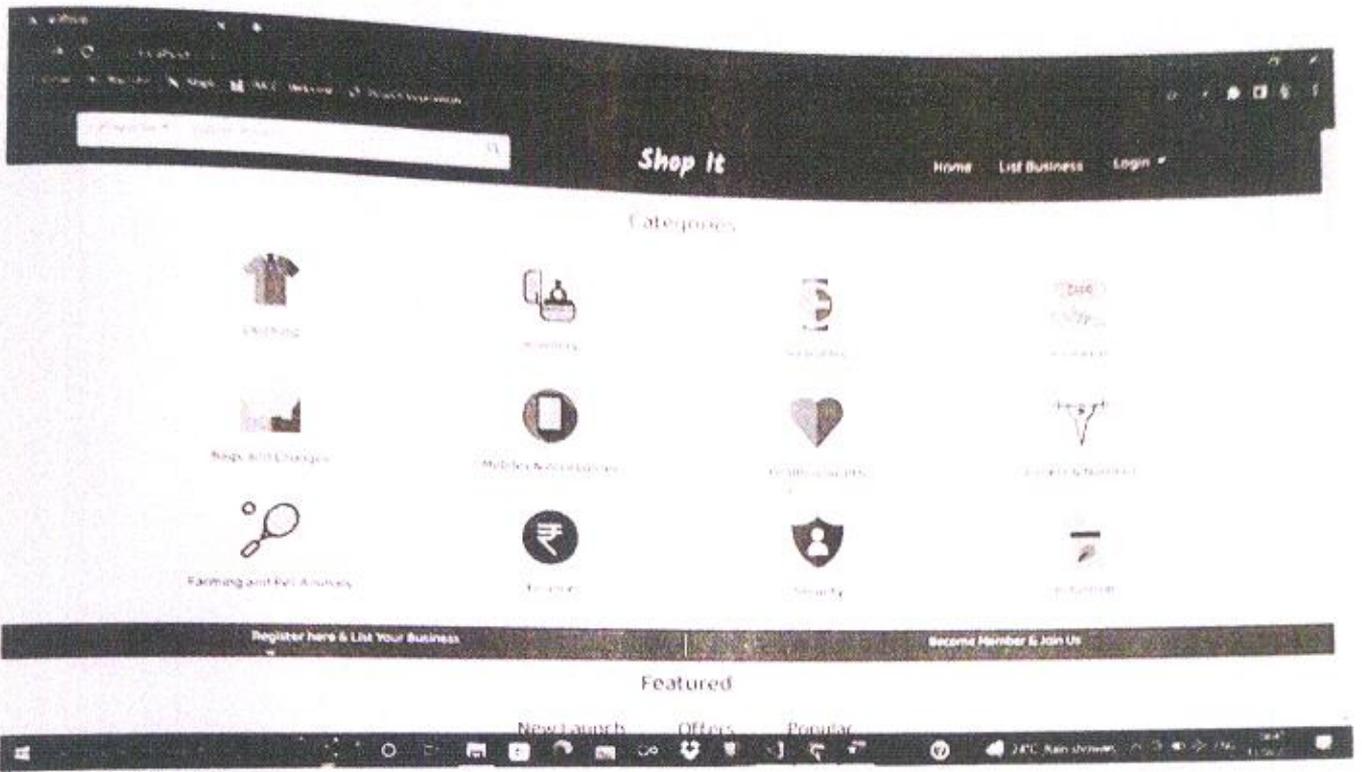
Home Page

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management

Seller's Dashboard

Product List Seller's Page

Add New Product

Add New Product

Supplier's Profile

Shop It Home List Business Logout

AGRICONIC Agriconic

Product Description Page

Your Bag

Name	Price per piece	Quantity	Total
Product name			

Customer's Orders

Cart   Orders

Orders

Product Name	Price Per Item	Quantity	Delivery Status
Butterfly Ring	10000	1	Placed
Mrignaini Gold Diamond Ring	12284	1	Out for delivery
Topaz	2145	1	Out for delivery
Tia Twin	166	10	Delivered
Mrignaini Gold Diamond Ring	12284	1	Out for delivery

Admin Reports

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management   51

Supplier's Report

Customer's Report

All Supplier

Name		Email	Address	State

Product's Report

Product

Code	Name	Category	Price	Supplier

53

4.1 Algorithms

Admin side:

Step 1 : Start

Step 2 : login

Step 3 : if (username & password do not match)then

Print " please enter correct username & password "

Else if (username is wrong) then

Print " please enter correct username "

Else if (password is wrong) then

Print "please enter correct password"

Else

Print "Login Successful"

User side:

Step 1 : Start

Step 2 : login

Step 3 : if (username & password do not match)then

Print " please enter correct username & password "

Else if (username is wrong) then

Print " please enter correct username "

Else if (password is wrong) then

Print " please enter correct password "

Else

Print 'Login Successful'

Step 4 : Add product

If (input name ,price, quantity Empty)then

Print " please enter required details "

Else

Print " Product added in cart successfully "

Step 5 : Cart

If (cart Empty)then

Print " Please select Product "

Else

Print " Please make payment"

Step 6 : Logout

4.2 Code Snippets: Customer

View

```
from rest_framework.authtoken.models import Token from
rest_framework.decorators import api_view from
rest_framework.response import Response from django.utils
import timezone from rest_framework import status from
django.contrib import auth from .models import Customer from
user.models import User

@api_view(['POST']) def
create_customer(request):
    try:
        full_name = request.data.get("fullName")                    mobile =
request.data.get("mobileNumber")              address = request.data.get("address")
taluka = request.data.get("district")              post = request.data.get("taluka")
district = request.data.get("pincode")          email = request.data.get("email")
password = request.data.get("password")              account_exists =
len(Customer.objects.filter(mobile
= mobile))      response = {}        print("account_exists - ", account_exists)
if(not account_exists):           user = User.objects.create(username = email)
print("user - ". user)          user.set_password(password)           user.save()
            Customer.objects.create(full_name = full_name, email = email, mobile =
mobile, address = address, taluka = taluka, post = post, district = district)
response = {"status": "Account Created"}          return Response(response, status
= status.HTTP_200_OK)    except Exception as e:
        print(e)        response = {"message" : "Internal Server Error"}
```

return Response(response, status =

```
status.HTTP_500_INTERNAL_SERVER_ERROR)

@api_view(['POST']) def
login_customer(request):
    try:
        email = request.data.get("email")         password =
request.data.get("password")
        user = auth.authenticate(username = email, password
= password)        print("user", user)
        response = {"response": "No Account Found Or Invalid
Number"}        token = None        if(user):
            token = Token.objects.filter(user = user)        user.last_login =
timezone.now()        user.save()        if not len(token):
            token = Token.objects.create(user = user)        print(token);        if
token:
            response = {"response": token.key}        return
Response(response, status = status.HTTP_200_OK)        except
Exception as ex:
        print(ex)        response = {"message" : "Internal Server Error"}
return Response(response, status =
status.HTTP_500_INTERNAL_SERVER_ERROR)
```

Product Views

```
import json from user.models import User from django.utils import
timezone from rest_framework import status from .models import
Product, Category from rest_framework.response import Response
from rest_framework.decorators import api_view from
supplier.models import Business, Supplier from
rest_framework.authtoken.models import Token from
django.views.decorators.csrf import csrf_exempt from .serializer
import CategorySerializer,
ProductSerializer
```

```python
@api_view(['GET']) def
get_business_product(request):    try:
    business_id = request.GET.get('business_id')    business =
Business.objects.get(id=business_id)    product =
Product.objects.filter(business=business)    product =
ProductSerializer(product, many=True)    product =
json.loads(json.dumps(product.data))    response = {"product":
Response(response, status=status.HTTP_200_OK)    except Exception as e:
    print(e)    response = {"message" : "Internal Server Error"}
return Response(response, status=
status.HTTP_500_INTERNAL_SERVER_ERROR)


@api_view(['GET']) def
get_all_categories(request):
    try:

    category = Category.objects.all()    category =
CategorySerializer(category, many=True)    category =
json.loads(json.dumps(category.data));    response = {"category":
status.HTTP_200_OK)    except
Exception as e:
    print(e)    response = {"message" : "Internal Server Error"}
return Response(response, status =
status.HTTP_500_INTERNAL_SERVER_ERROR)


@api_view(['POST']) def
create_product(request):    try:
    product = Product()
product.click = 0    product.name = request.data.get("name")
request.data.get("arrival")    product.price = request.data.get("price")    arrival =
False    product.unit = request.data.get("unit")    product.arrival = True if (arrival == "Yes") else
request.data.get("brand")    product.code = request.data.get("code")    product.brand =
= request.data.get("offer")    product.offer = True if (offer == "Yes") else    offer
False    product.additional_information =
request.data.get("additional_information")    product.specification_1 =
request.data.get("specification_1")    product.specification_2 =
request.data.get("specification_2")    product.specification_3 =
request.data.get("specification_3")    product.specification_4 =
request.data.get("specification_4")    product.specification_5 =
request.data.get("specification_5")    product.specification_6 =
request.data.get("specification_6")    product.specification_7 =
request.data.get("specification_7")    product.specification_8 =
request.data.get("specification_8")
```

return Response(response, status...

```
product.specification_9 = request.data.get("specification_9")
product.specification_10 = request.data.get("specification_10")
product.description_1 = request.data.get("description_1")
product.description_2 = request.data.get("description_2")
product.description_3 = request.data.get("description_3")
product.description_4 = request.data.get("description_4")
product.description_5 = request.data.get("description_5")
product.description_6 = request.data.get("description_6")
product.description_7 = request.data.get("description_7")
product.description_8 = request.data.get("description_8")
product.description_9 = request.data.get("description_9")
product.description_10 = request.data.get("description_10")
product.created_date = timezone.now()
category = int(request.data.get("categoryId"))
product.category = Category.objects.get(id = category)
token = request.headers.get("sessionId")
user = Token.objects.filter(key = token)
if(len(user)):
    user = user[0]
    user = User.objects.get(id = user.user_id)
    supplier = Supplier.objects.get(email = user.username)
    business = Business.objects.get(supplier = supplier)
    product.business = business
product.save()
response = {product.id}
status.HTTP_200_OK)
except Exception as e:
    print(e)
    response = {"message" : "Internal Server Error"}
return Response(response, status = status.HTTP_500_INTERNAL_SERVER_ERROR)


@csrf_exempt
@api_view(['POST'])
def update_product_image(request):
    try:
        image = request.data.get('image')
        product_id = request.data.get("productId")
        product = Product.objects.get(id = product_id)
        product.image = image
        product.save()
        response = True
        return Response(response, status = status.HTTP_200_OK)
    except Exception as e:
        print(e)
        response = {"message" : "Internal Server Error"}
return Response(response, status = status.HTTP_500_INTERNAL_SERVER_ERROR)


@api_view(['GET'])
def get_searched_product(request):
    try:
```
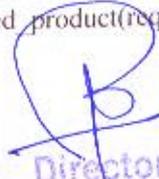
```
            product = request.GET.get('product')
request.GET.get('counter')                counter =
Product.objects.filter(name __icontains = product)
len(product_list)                product_list =
> 0): total_pages += 1      total_pages = total_product // 4     if(total_product % 4
= True)        product_list =    product_list = ProductSerializer(product_list, many
json.loads(json.dumps(product_list.data))
= {"total_pages": total_pages,                              response
"product_list" : product_list}
Response(response, status = status.HTTP_200_OK)      return
except Exception as e:
        print(e)        response = {"message" : "Internal Server Error"}
return Response(response, status =
status.HTTP_500_INTERNAL_SERVER_ERROR)

@api_view(['GET']) def
get_product_by_id(request):    try:
        product_id = request.GET.get("product_id")
Product.objects.get(id = product_id)                       product =
= 0        product.clicks = product.clicks + 1       if(not product.clicks): product.clicks
ProductSerializer(product, many = False)         product.save()        product =
json.loads(json.dumps(product.data))           product =
        "product": product              response = {

        }

    return Response(response, status =
status.HTTP_200_OK)    except Exception as e:
    print(e)

    response = {"message" : "Internal Server Error"}
Response(response, status =                           return
status.HTTP_500_INTERNAL_SERVER_ERROR)
```

Product View

```
<div class="container-fluid" style="cursor pointer">
<!-- ----------Navbar Position ...
<div class="row">
<div class="col-sm-2 col-md-3 col-lg-3 p-0 categoriesBelt">
<div class="sidenav">
style="text-decoration underline"; Filter</h5>
<!-- <ul class="navbar-nav nav flex-column">
<li class="nav-item dropdown active">
link dropdown-toggle" role="button" data-toggle="collapse"
datatarget="#coladrop1" aria-expanded="true"
ariacontrols="coladrop1">
        Related Categories
</a>

<div class="collapse show" id="coladrop1"
arialabelledby="dropdownMenuButton" data-parent="#accordion">
class="dropdown-item active" href="#maindiv">Agricultural Implements</a>
class="dropdown-item" href="#">Agricultural Machinery</a>
class="dropdown-item" href="#">Agricultural Machine Parts</a>
        <a class="dropdown-item" href="#">Agricultural Tools</a>
Equipment</a>

        <a class="dropdown-item"
href="#">Agricultural Tray</a>
class="dropdown-item" href="#">Agricultural
Ridger</a>

        <a class="dropdown-item" href="#">Another action</a>
        <a class="dropdown-item"
href="#">Agricultural Gyrovator</a>
class="dropdown-item" href="#">Agricultural
Wire</a>        <a class="dropdown-item"
href="#">Agricultural Tanks</a>
        <a class="dropdown-item" href="#">Sugarcane stubble shaver</a>
        <a class="dropdown-item" href="#">Agricultural Processing Plants</a>
    </div>
</li>


<li class="nav-item dropdown">        <a class="nav-
link dropdown-toggle" role="button" data-toggle="collapse"
datatarget="#coladrop2" aria-expanded="false"
ariacontrols="coladrop2">        Brands
</a>
```

```html
<div class="collapse" id="coladrop3"
arialabelledby="dropdownMenuButton" data-parent="#accordion">
class="dropdown-item" href="#">Honda</a>
    <a class="dropdown-item" href="#">AG
Champ</a>
    <a class="dropdown-item" href="#">RIS
Steel</a>
    <a class="dropdown-item" href="#">Kirloskar
Power</a>
    <a class="dropdown-item" href="#">Adani</a>         <a
class="dropdown-item" href="#">HDPE Coil
Pipes</a>
    <a class="dropdown-item" href="#">Lateral
Pipes</a>
    <a class="dropdown-item" href="#">MDPE Pipes</a>
    <a class="dropdown-item" href="#">HDPE Sprinkler Pipes</a>
    <a class="dropdown-item" href="#">UPVC Pressure Pipes</a>
    <a class="dropdown-item" href="#">Grey
Pipes</a>
    <a class="dropdown-item" href="#">LDPE Lay
Flat Pipes</a>
    <a class="dropdown-item" href="#">Hdpe
Tube</a>
    </div>
    </li>
    </ul> -->


    <p-accordion>
    <p-accordionTab header="Related Categories"
[selected]="true">
    <a class="dropdown-item active" href="#maindiv">Agricultural
Implements</a>
    <a class="dropdown-item" href="#">Agricultural Machinery</a>
    <a class="dropdown-item" href="#">Agricultural Machine Parts</a>
    <a class="dropdown-item" href="#">Agricultural
Tools</a>
    <a class="dropdown-item" href="#">Farm
Equipment</a>
    <a class="dropdown-item" href="#">Agricultural Tray</a>
    <a class="dropdown-item" href="#">Agricultural
Ridger</a>
    <a class="dropdown-item" href="#">Another action</a>
```

```html
<a class="dropdown-item" href="#">Agricultural Gyrovator</a>
<a class="dropdown-item" href="#">Agricultural Wire</a>
<a class="dropdown-item" href="#">Agricultural
Tanks</a>
<a class="dropdown-item" href="#">Agricultural
Processing Plants</a>
<a class="dropdown-item" href="#">Sugarcane stubble shaver</a>
<a class="dropdown-item" href="#">Agricultural
</p-accordionTab>
<p-accordionTab header="Brands">
<a class="dropdown-item" href="#">Honda</a>
<a class="dropdown-item" href="#">AG
Champ</a>
<a class="dropdown-item" href="#">RIS Steel</a>
<a class="dropdown-item" href="#">Kirloskar
Power</a>
<a class="dropdown-item" href="#">Adani</a>
<a class="dropdown-item" href="#">HDPE Coil
Pipes</a>
<a class="dropdown-item" href="#">Lateral
Pipes</a>
<a class="dropdown-item" href="#">MDPE Pipes</a>
<a class="dropdown-item" href="#">HDPE
Sprinkler Pipes</a>
<a class="dropdown-item" href="#">UPVC Pressure
Pipes</a>
<a class="dropdown-item" href="#">Grey
Pipes</a>
<a class="dropdown-item" href="#">LDPE Lay Flat Pipes</a>
<a class="dropdown-item" href="#">Hdpe Tube</a>
</p-accordionTab>
</p-accordion>
</div>
</div>


<div class="col-sm-10 col-md-9 col-lg-9 productFrame">          <div
class="row rowProduct" *ngFor = "let product of productList">
<div class="col-sm-3 col-md-3 col-lg-3 productImage mt-4">
<img class="img-fluid" src = "{{product.image}}" alt="Advertisement
Pic">
</div>
<div class="col-sm-9 col-md-9 col-lg-9 productDescription">
<a class = "productLink"
```

```html
(click)="productdesc(product.id)"
      <h5 class="card-title">{{product.name}}</h5>
      <p class="card-text m-
0">{{product.additionalDescription}}</p>
      <a>
      <div class="row">
        <div class="col-sm-12 col-md-8 col-lg-8 productDetails">
         <p class="productPara"><b>{{product.specification_1}}</b>
</p>
{{product.description_1}}</p>
class="productPara"><b>{{product.specification_2}}</b>
product.description_2}}</p>
class="productPara"><b>{{product.specification_3}}</b>
{{product.description_3}}</p>                    </p>
class="productPara"><b>{{product.specification_4}}</b>
{{product.description_4}}</p>

          <span class="fa fa-star checked"></span>
          <span class="fa fa-star checked"></span>
          <span class="fa fa-star checked"></span>
          <span class="fa fa-star"></span>
          <span class="fa fa-star"></span>

        </div>
        <div class="col-sm-12 col-md-4 col-lg-4 productCompany">
          <button type="button" class="btn btnprimary"
(click)="openGetQuote(product.id, product.name)">Get Quote</button><br>
          <button type="button" class="btn btnprimary"
(click)="openContactSupplier()">Contact
Supplier</button>
        </div>

      </div>
    </div>
    </div>

      <div class="text-center">
        <button class="text-center p-2 m-2" *ngFor="let counts of
counter(totalPage); let i = index"
(click)="nextPage(i)">{{i}}</button>
      </div>

    </div>
    </div>
```

35

```
</div>

<!-- Get Quote -->
<div class="modal fade modalGetQuote" bsModal #modalGetQuote="bs-modal"
id = "modalGetQuote" tabindex="-1" role="dialog" aria-
labelledby="myLargeModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-lg">
    <div class="modal-content">

    <div class="modal-header">
        <h5 class="modal-title">Get Quote for
{{productQuoteName}}</h5>
        <button type="button" class="close" datadismiss="modal" aria-
label="Close">
            <span aria-hidden="true">&times;</span>
        </button>
    </div>

    <div class="modal-body">
        <!-- <h5 id = "productName"
class="textcenter">{{productQuoteName}}</h5> -->
        <form id = "submitQuote" #submitQuote="ngForm"
(ngSubmit)="saveQuote()">
            <div class="form-group">
            <input type="number" name="productQty" id="productQty"
[(ngModel)]="quotesObject.quantity" class="form-control quoteControl"
placeholder="Quantity" min="1">
            <select name="selectUnit" id="selectUnit" class="form-control
quoteControl">
                <option value="" selected>Select
Unit</option>
                <option value="Kg">Kg</option>
                <option value="Pack">pack</option>
                <option value="Unit">Unit</option>
                <option value="Meter">Meter</option>
                <option value="Piece">Piece</option>
            </select>
        </div>

        <div class="form-group">
```

```html
        <input type="text" [(ngModel)] = "quotesObject.requirement"
name="productRequirements" id="productRequirements" class="form-control
quoteControl w-75" placeholder="Requirement">
        </div>

        <div class="form-group">
          <input type="submit" value="Submit" class="btn btn-primary float-
right quoteControl">
        </div>
      </form>
    </div>
   </div>
  </div>
 </div>
```

Chapter 5: Testing

Test Strategy:

Scope:

Scope of the project is to test full Shop It System

Not in scope: - Load Testing

In Scope: - Regression. Smoke, Performance and functionality

Test Approach:

It defines Process of testing Regression Testing Smoke Testing

Performance Testing

Testing levels: Iterations covered while testing are 3 Roles.

Tester	Module	Time	Responsibility
Shahin Sayyed	Full Application	2 Days	Complete functionality testing of Shop It

Types of Testing: - Load testing. Security testing, Performance testing

Testing approach & automation tool if applicable: - Selenium

Test Environment:
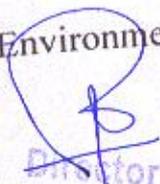
The software to be tested: - Shop It

The operating system: - Windows 10

Database: - SQL

The device on which the software is to be tested – desktop

Test automation frame-work: - Selenium

Testing Tools:

Automation and Test management tools needed for test

Execution: - Selenium

Release Control:

Release management plan with appropriate version history that will make sure test execution for all modification in that release

Sr. No	Application	Version
1	Shop It	1.0

Unit Test Plan:

Test Risks / Issues:
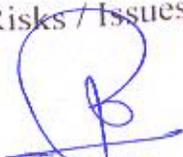
To test full Application with all it features and functionality.

Items to be Tested / Not Tested:

Full application to be tested with all its module and design not to be tested

Item To test	Test Description	Test Date	Responsibility
Login Page	To test login page with Proper details	05/08/2022	Tester
Admin Panel	To Test all functionality of admin panel	05/08/2022	Tester
User Dashb oard	To Test all Functionality of dashboard	06/08/2022	Tester

Acceptance Test Plan:

Test Risks / Issues:

To test full Application with all it features and functionality.

Items to be Tested / Not Tested:

Full application to be tested with all its module and design not to be tested

Item to test	Test Description	Test Date	Responsibility
Login Page	To test login page with proper details	05/08/2022	Tester
Admin Panel	To Test all functionality of admin panel	05/08/2022	Tester
User Dashboard	To Test al functionality of dashboard	06/08/2022	Tester

Test Cases/Test Scripts:

Test Scenario ID		Login-1		Test Case ID		Login 1A	
Test Case Description		Login test case		Test Priority		High	
S. No	Action	Inputs		Expected Output	Actual Output	Test Browser	Test Result

1	Enter Correct Email & Password and hit Login button	Emailid:test @xyz.com Password: ******	Login success	Login success	Chrome	Pass
2	Enter invalid Email & any Password and hit login button	Emailid: invalid@x yz.com Password: ******	Invalid Credenti als	Invalid Credenti als	Chrome	Pass
3	Enter valid email & incorrect Password and hit login button	Emailid: valid@xyz. com Password: *****	Invalid Credentia ls	Invalid Credentia ls	Chrome	Pass

4	Check if both password and confirm password are same	Email id – supplier@g mail.com Password – ******* Confirm Password ****	Password and confirm password does not match	Password and confirm password does not match	Chrome	Pass

42

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management

5	Check if valid email address is entered	Email id -- supplier@.com	Please enter valid email address	Please enter valid email address	Chrome	Pass
6	Password length check.	Password - ****	Password length should be greater than 8	Password length should be greater than 8	Chrome	Pass
7	Product Image Size	Product Image with larger size	Image Size Limit Exceeds	Image Size Limit Exceeds	Chrome	Pass
8	Product Price cannot be less than 1	Product Price cannot be zero	Product Price cannot be zero	Product Price cannot be zero	Chrome	Pass

9	Year of Establishme nt cannot be greater than today's date	Cannot Select date greater than today	Cannot Select date greater than today	Cannot Select date greater than today	Chrome	Pass

5.5   Defect Report/Test Log:

N/A

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management

Chapter 6: Limitations of Proposed System

1. Limitations of Proposed System

This system can be improved in terms of processing, requirement.

Chapter 7: Proposed Enhancements

- Proposed system allows easy product upload making easily available to the customer.
- Easy buy options to the customer.
- Instant stats of the customer behaviour on product or custom reviews.
- Product Review

2. Conclusion

- "Shop It" will allow grown exponentially from their BUSINESS.
- The future of E-commerce will drastically progress over the years as the amount of internet users among businesses and consumers grows drastically every year.
- Consumers can also represent themselves as a buyer a group.

3. Bibliography

- https://stackoverflow.com/
- https://dev.mysql.com/doc/
- https://youtube.com/
- https://forum.djangoproject.com/
- https://angular.io/api/
- https://google.com
- https://colorhunt.co
- https://www.geeksforgeeks.org/ User Manual

Director
PSD Shastri Educational Foundation's
ASMA Institute of Management