

SMART FASHION RECOMMENDER APPLICATION

Project Report

Team ID: PNT2022TMID38260

Team members:

B. Sairam (412319205023)

A.Lakshmanan(412319205015)

P.Govaradhan (412319205011)

C.Saravana Kumar(412319205025)

D.Praveen Kumar(412319205020)

In partial fulfilment for the award of degree of Bachelor of Engineering (B.Tech) Information & Technology and Engineering (IT).

Sri Ramanujar Engineering College

ANNA UNIVERSITY

CONTENTS

1. INTRODUCTION

- 1.1 Project Overview
- 1.2 Purpose

2. PROJECT DESIGN

- 2.1 Data Flow Diagrams
- 2.2 Solution & Technical Architecture
- 2.3 User Stories

3. IDEATION & PROPOSED SOLUTION

- 3.1 Empathy Map Canvas
- 3.2 Proposed Solution
- 3.3 Problem Solution fit

4. PROJECT PLANNING & SCHEDULING

- 4.1 Sprint Planning & Estimation
- 4.2 Sprint Delivery Schedule

5. Project Flow with Images

- 5.1 Customer Panel Flow
- 5.2 Admin Panel Flow

6.Source Code For Entire Project

- 6.1 Business Logic for project
- 6.2 Source Code

7. Github & Demo Video Link

1. INTRODUCTION

1.1 Project Overview

In our project the user can directly do the online shopping based on the choice without any search. It can be done by using the chatbot. Typically it will recommend new products to users according to the individual browsing records. Content-Based Filtering (CBF) can attain the specific interest of a user and make recommendations accordingly.

We have come up with a new innovative solution through which you can directly do your online shopping based on your choice without any search. It can be done by using the chatbot.

In this project you will be working on two modules :

1. Admin and

2. User

Admin:

The role of the admin is to check out the database about the stock and have a track of all the things that the users are purchasing.

User :

The user will login into the website and go through the products available on the website. Instead of navigating to several screens for booking products online, the user can directly talk to Chatbot regarding the products. Get the recommendations based on information provided by the user.

Features of Chatbot :

Using chatbot we can manage user's choices and orders.

The chatbot can give recommendations to the users based on their interests.

It can promote the best deals and offers on that day.

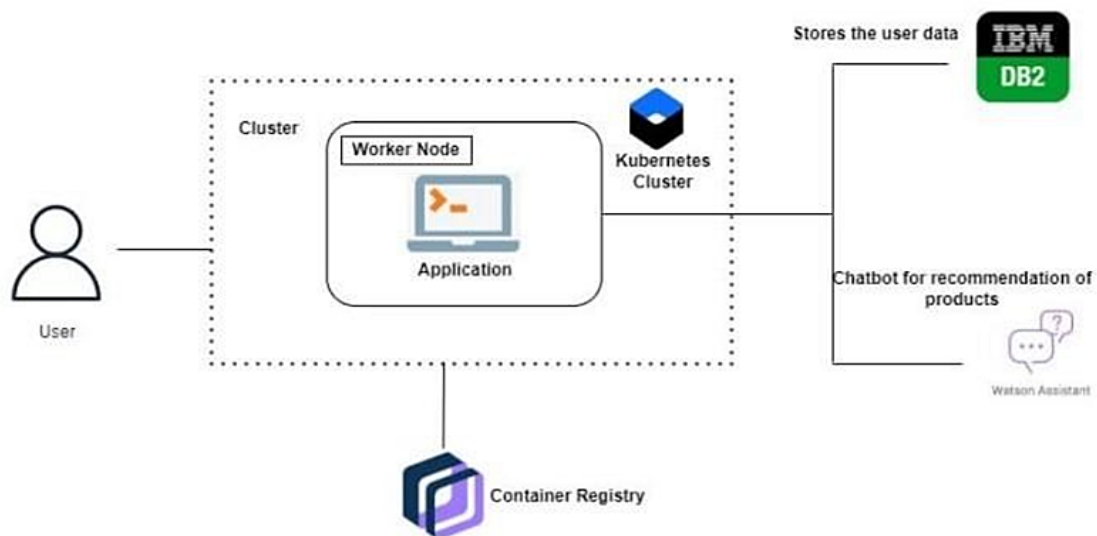
It will store the customer's details and orders in the database.

The chatbot will send a notification to customers if the order is confirmed.

Chatbots can also help in collecting customer feedback.

2. Project Design

2.1 Data Flow Diagrams



2.2 Solution & Technical Architecture :

Solution Requirements (Functional & Non-functional)

Date	03 October 2022
Team ID	PNT2022TMID38260
Project Name	Project – Smart Fashion Recommender Application
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
FR-2	User Interaction	Interact through Chat bot
FR-3	Product Recommendation	Recommendation through Chat bot
FR-4	Track Product	Track Product through Chat bot
FR-5	Order Product	Through Chat Bot

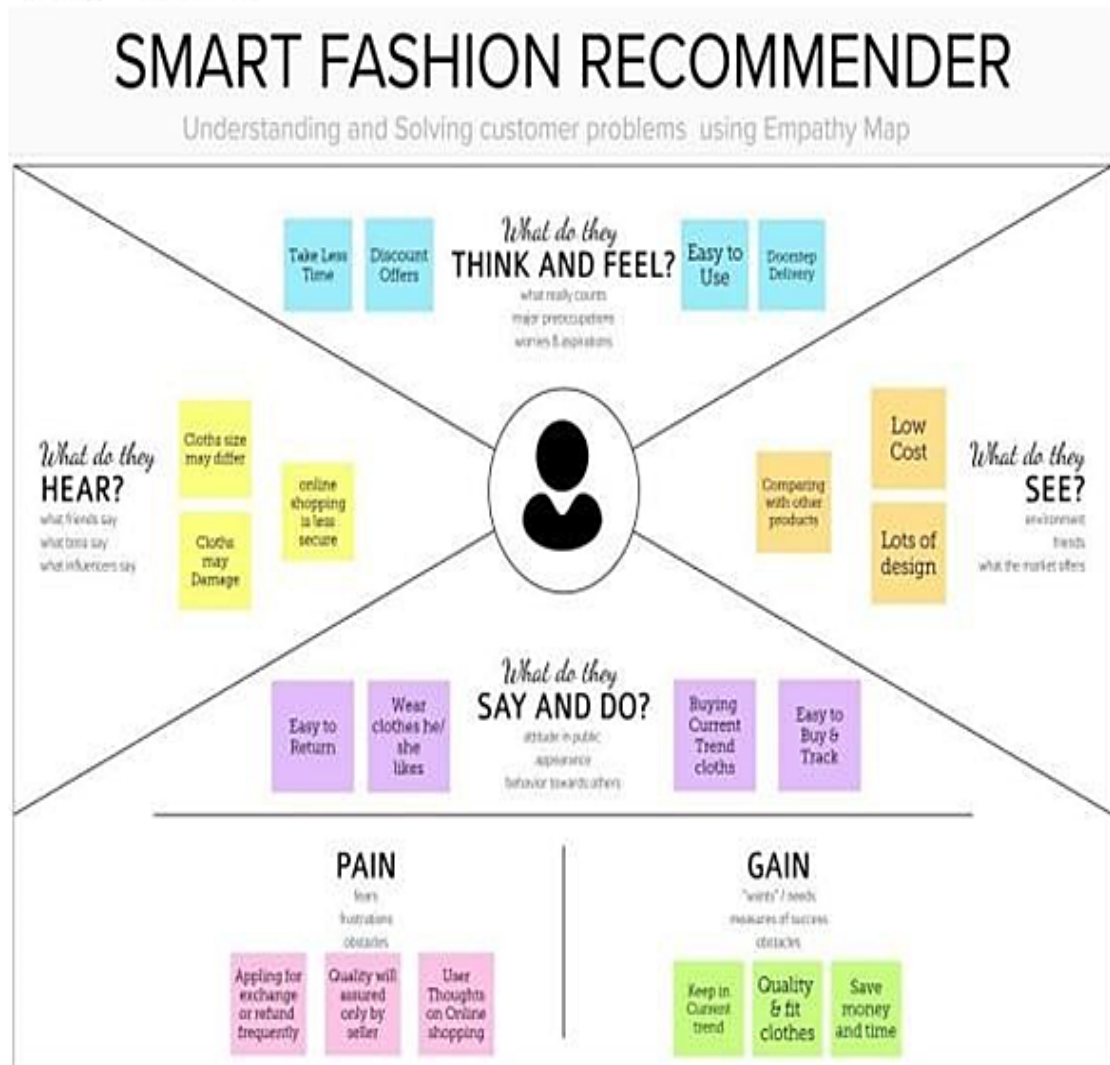
2.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)						
Customer Care Executive						
Administrator						

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Empathy Map Canvas:



3.2 Ideation & Brainstorming

1

Smart Fashion Recommender Application
solving the users problems in purchase Online shopping

5 minutes

How might we (your problem statement)?

Key rules of brainstorming
To run a smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, use visual.

2

pravee

10 minutes

saravanan

sairam & laxmanan

praveen

Govarathan

Trending Designs clothes

Shop with Assistant

Save Time

All size of clothes

Discounts & Offers

Cash on Delivery

24/7 Shopping

Home Delivery

Easy Price Comparison

Easy to send gifts

Reviews of Products

Shipping via the internet saves time

Online Tracking

Online Tracking

No need to travel

Free shipping

3

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance

If most of ideas have not get done, which ideas have the most positive impact?

4. PROJECT PLANNING & SCHEDULING

4.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Page	USN-1	The user will login into the website and go through the products available on the website	20	High	Sairam.B Govarthan.P Praveen Kumar.D Saravana Kumar.C Lakshmanan.A
Sprint-2	Admin page	USN-2	The role of the admin is to check out the database about the stock and have a track of all the things that the users are purchasing.	20	High	Sairam.B Govarthan.P Praveen Kumar.D Saravana Kumar.C Lakshmanan.A

4.2 Sprint Delivery Schedule

Sprint-3	Chat Bot Integration	USN-3	The user can directly talk to Chat bot regarding the products. Get the recommendations based on information provided by the user.	20	High	Sairam.B Govarthan.P Praveen Kumar.D Saravana Kumar.C Lakshmanan.A
Sprint-4	final delivery	USN-4	Container of applications using docker kubernetes and deployment the application. Create the documentation and final submit the application	20	High	Sairam.B Govarthan.P Praveen Kumar.A Saravana Kumar.C Lakshmanan.A

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	25 Oct 2022	29 Oct 2022		29 Oct 2022
Sprint-2	20	6 Days	30 Oct 2022	04 Nov 2022		04 Nov 2022

Sprint-3	20	6 Days	07 Nov 2022	13 Nov 2022		13 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		19 Nov 2022

Velocity:

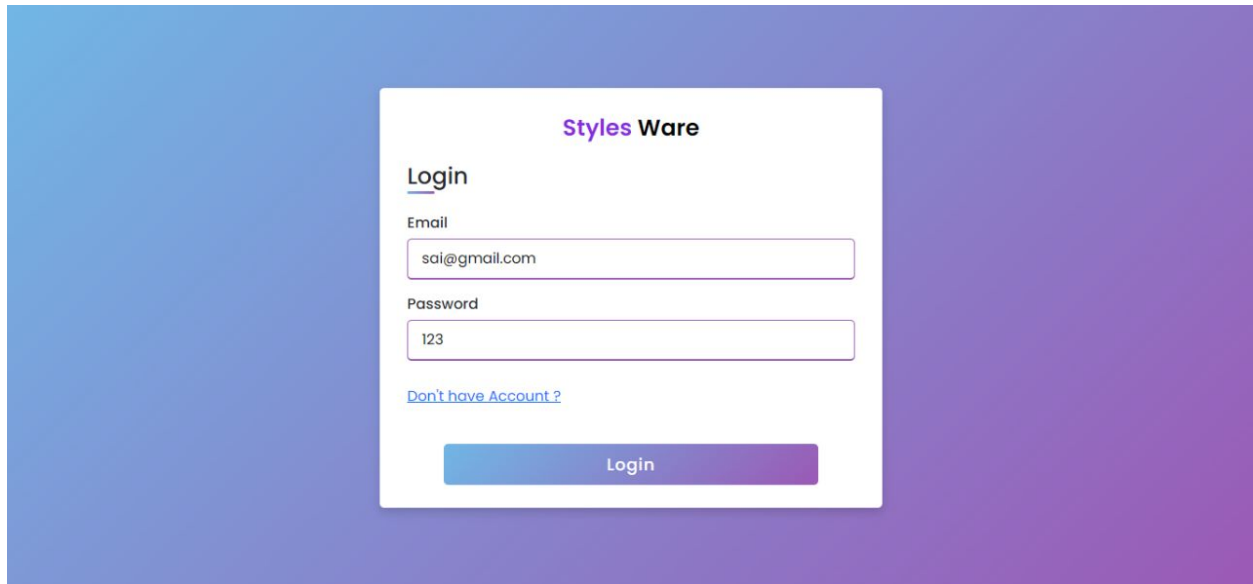
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

5. Project Flow with Images

5.1 Customer Panel Flow

Login page :



The login page features a white card centered on a blue-to-purple gradient background. The card has the 'Styles Ware' logo at the top, followed by the title 'Login'. Below the title are two input fields: 'Email' with the value 'sai@gmail.com' and 'Password' with the value '123'. A link for 'Don't have Account ?' is positioned below the password field. At the bottom of the card is a blue-to-purple gradient button labeled 'Login'.

Styles Ware

Login

Email

sai@gmail.com

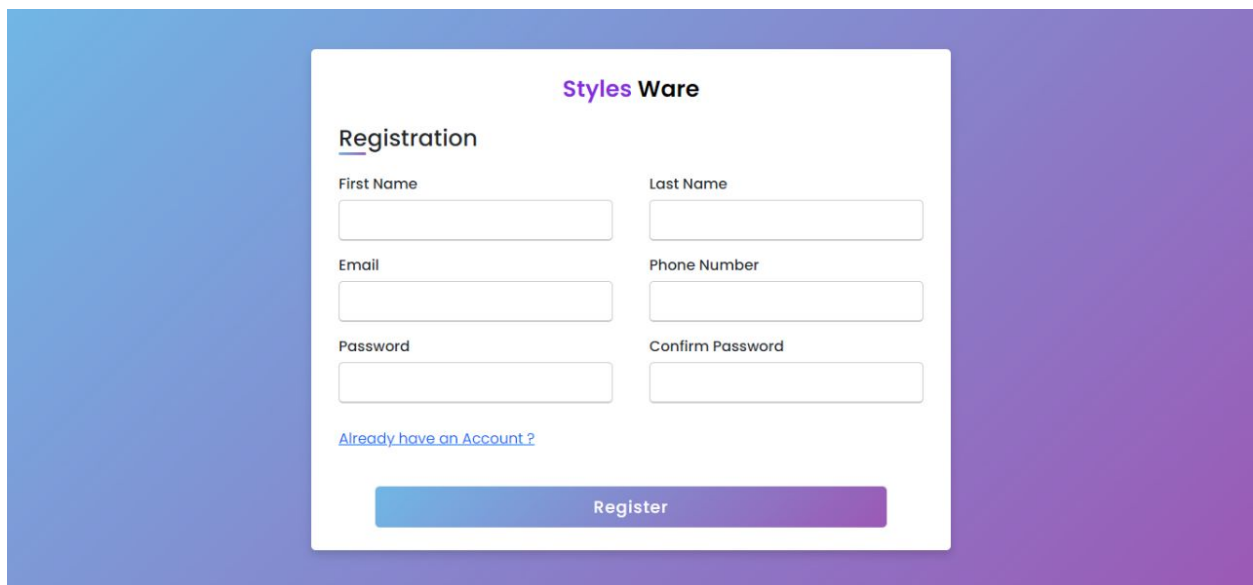
Password

123

[Don't have Account ?](#)

Login

Register page:



The registration page features a white card centered on a blue-to-purple gradient background. The card has the 'Styles Ware' logo at the top, followed by the title 'Registration'. Below the title are six input fields arranged in two columns: 'First Name', 'Last Name', 'Email', 'Phone Number', 'Password', and 'Confirm Password'. A link for 'Already have an Account ?' is positioned below the password fields. At the bottom of the card is a blue-to-purple gradient button labeled 'Register'.

Styles Ware

Registration

First Name

Last Name

Email

Phone Number

Password

Confirm Password

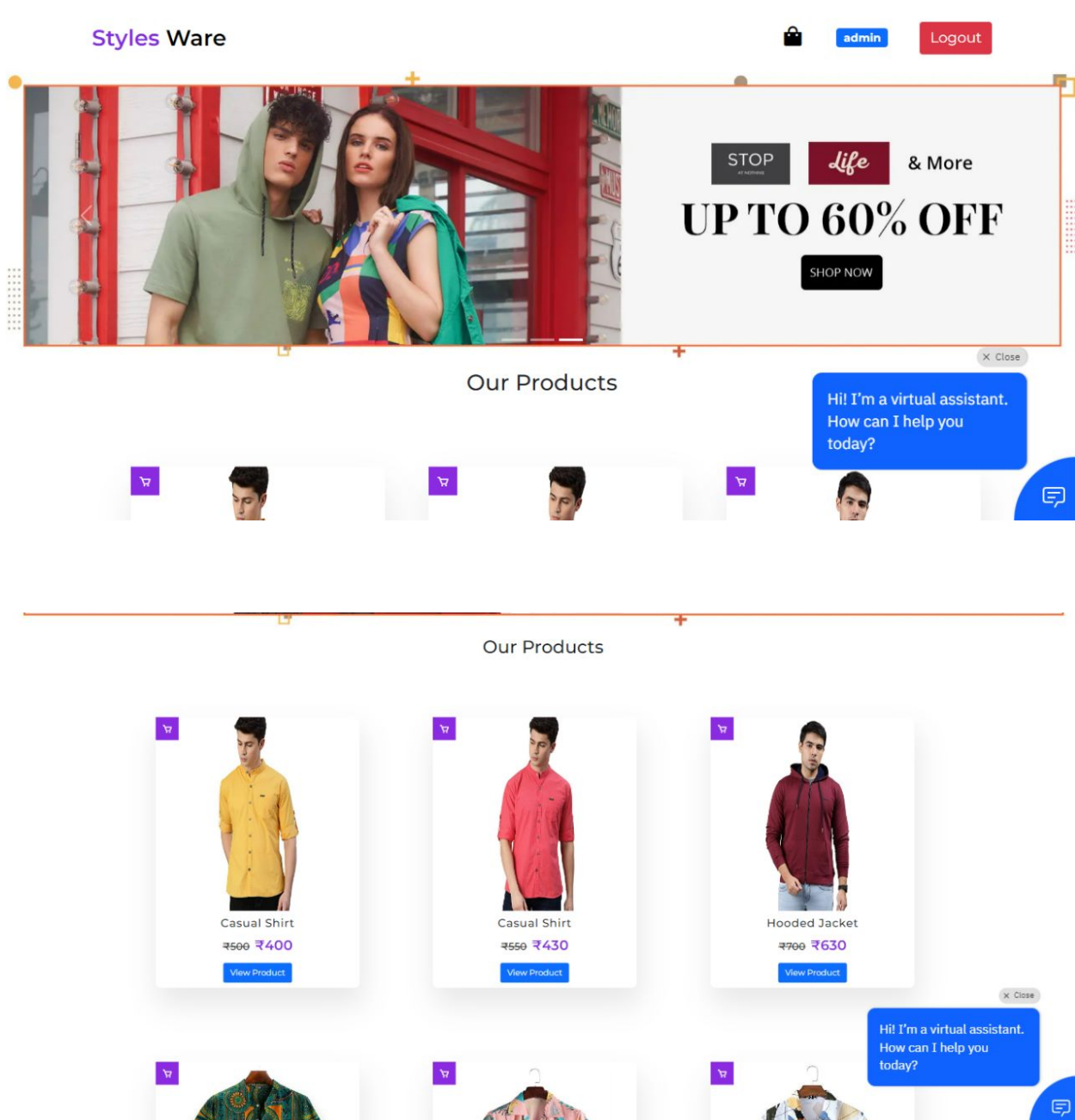
[Already have an Account ?](#)

Register

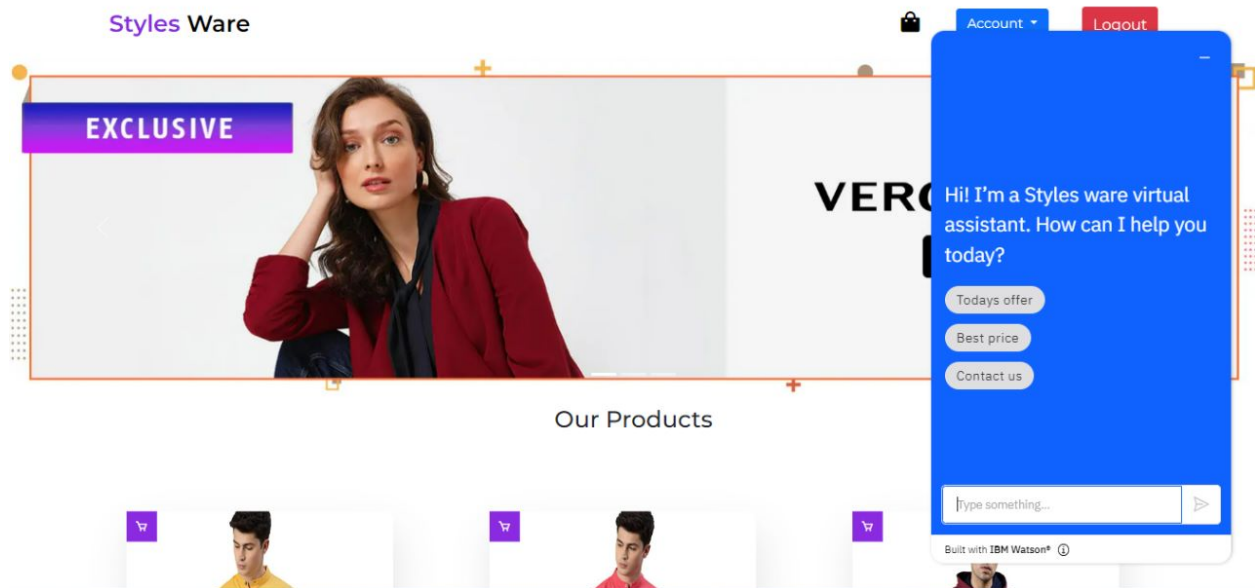


After Successful Login redirect to Home Page ↴

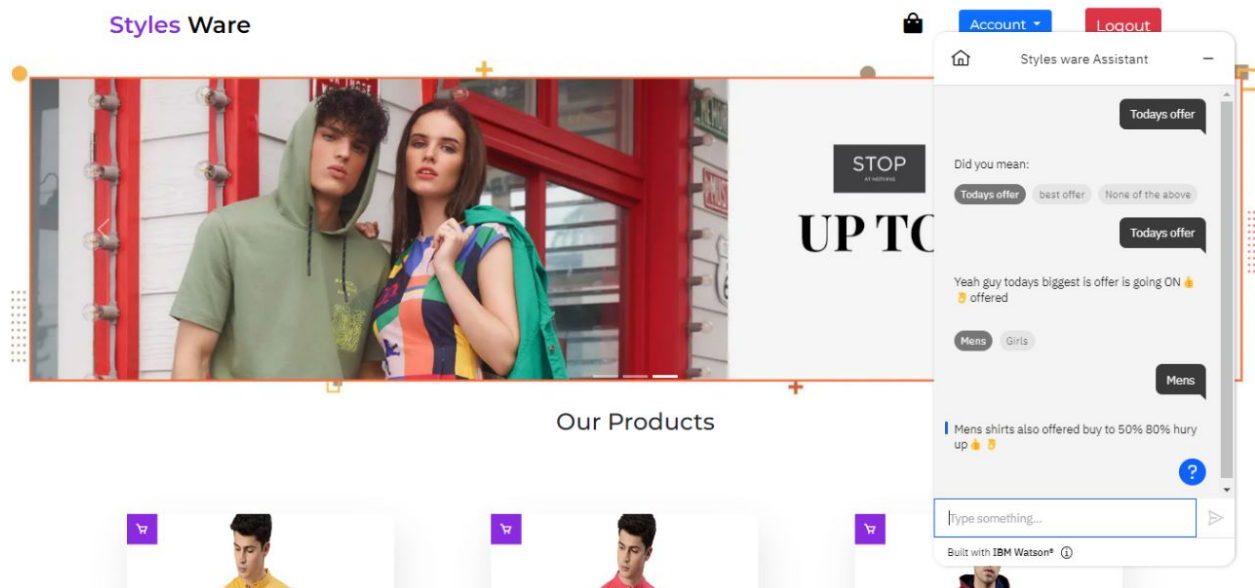
Home page:



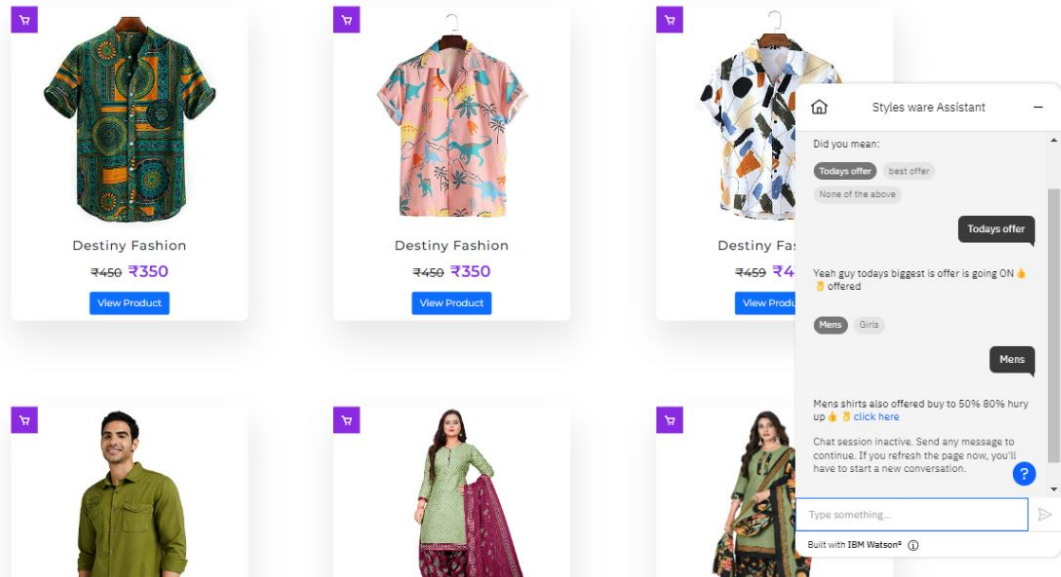
Intergeneration AI chatbot Watson Assistant ↴



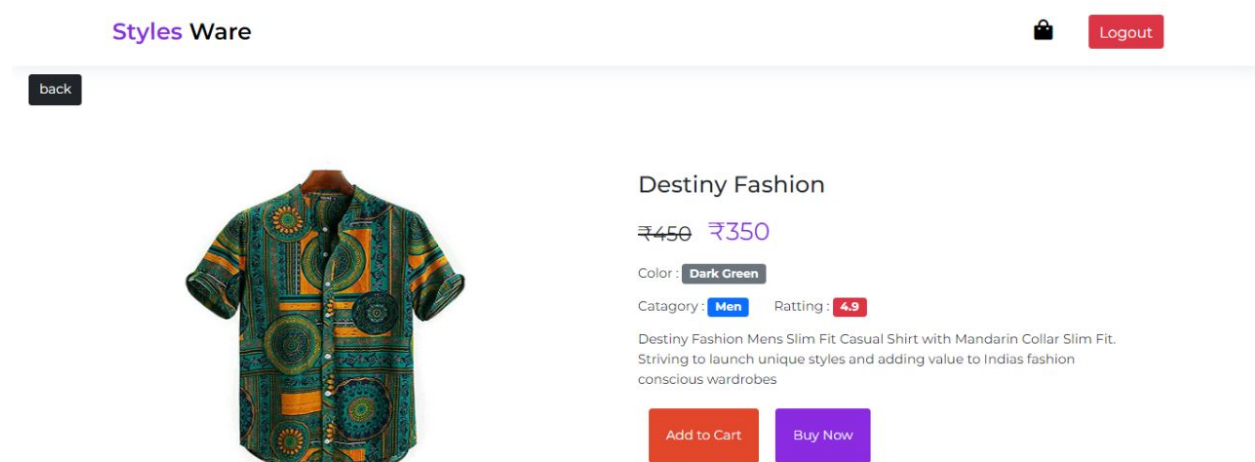
Customer can chat with chatbot to Know Today's offers:



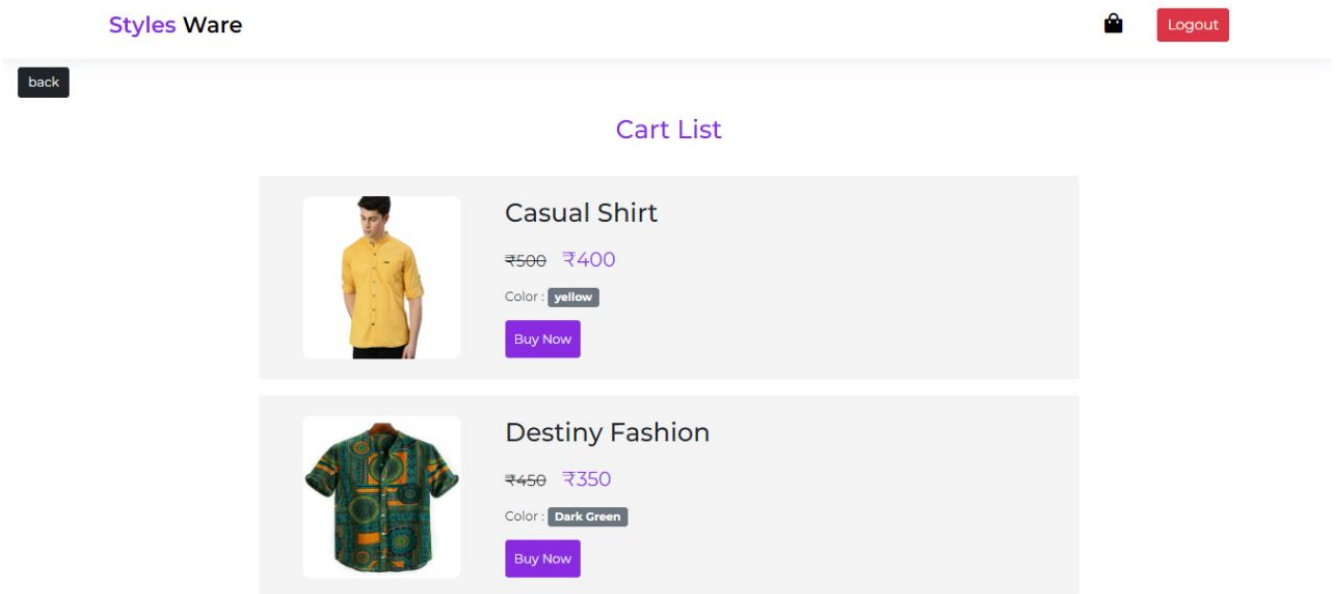
Customer can View the Product:



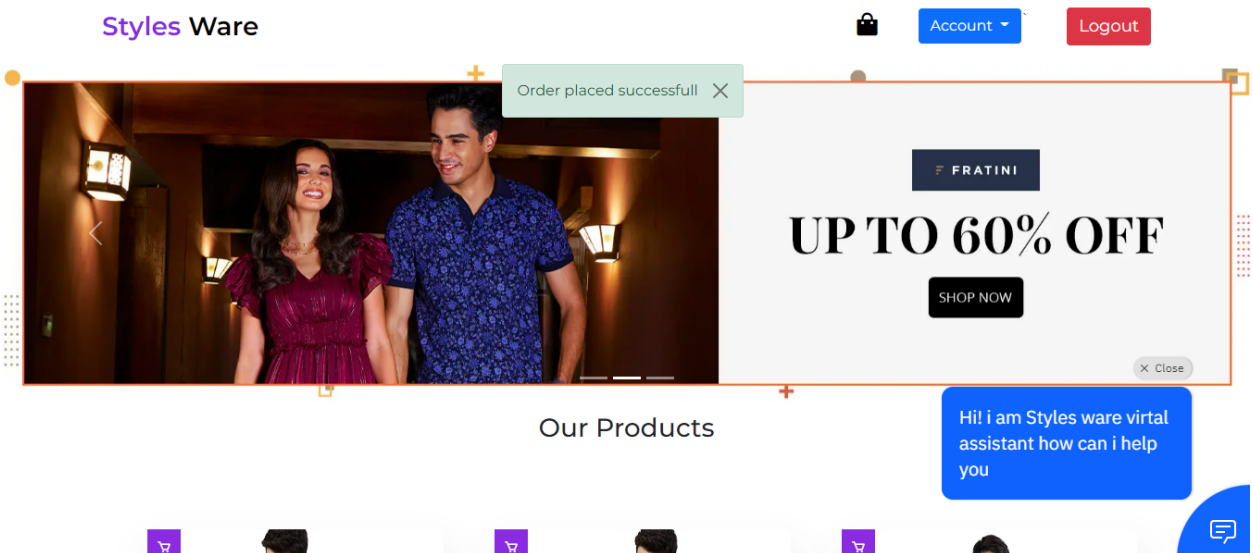
After Clicked the Product ↴



After Clicked Add to Cart Button ↴

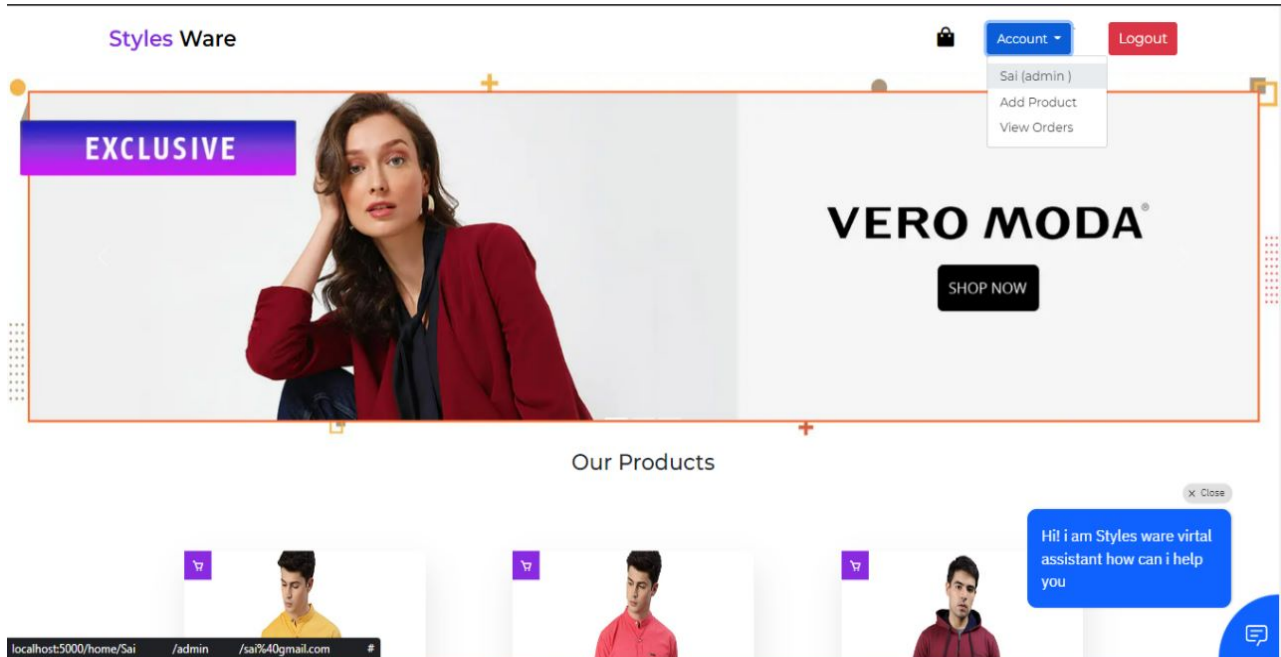


After Clicked Buy Now Button ↴



5.1 Admin Panel Flow ↴

Admin home page has **Add Product & view Order List**



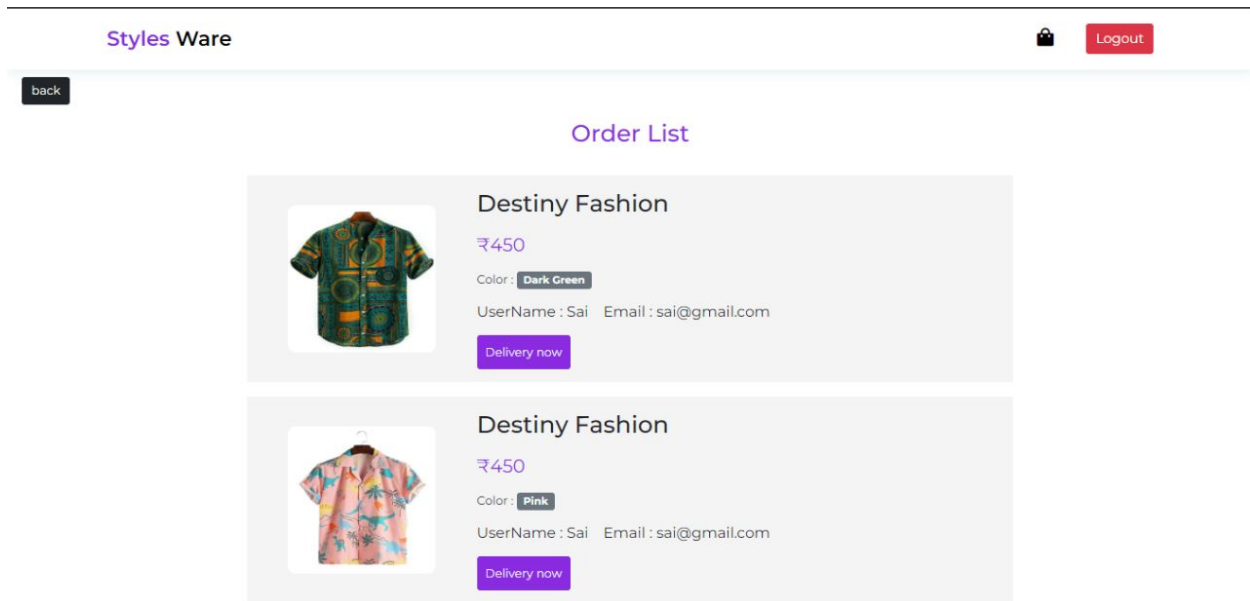
Add Product Page:

The screenshot shows the Styles Ware Add Product Page. At the top, there's a navigation bar with the 'Styles Ware' logo, a shopping cart icon, and a 'Logout' button. Below the navigation bar is a 'back' button. The main heading is 'Add Products'. The form contains the following fields:

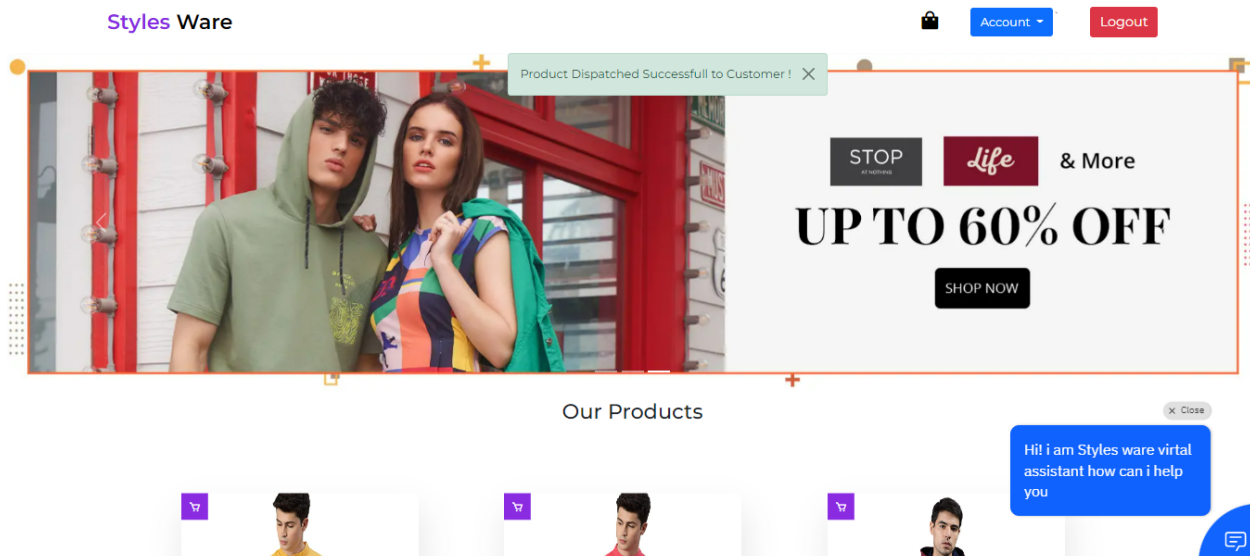
Product Name	Image Link
Men's Slim Fit	https://m.media-amazon.com/images/I/6TSUqgF0j-L
price	Color
500	Green
About	Offer Price
IndoPrimo Men's Slim Fit Casual Shirt Suitable for: S	400
Catagory	Set Rating
Men	4.0
Set ID	
1d	

At the bottom of the form is a large blue 'Add' button.

Admin Can list the Recived Orders:



After Clicked Buy now Button ↓
Will receive the message



6.Source Code For Entire Project

6.1 Business Logic for project (app.py)

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=8e359033-a1c9-4643-82ef-
8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30120;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=glp92926;PWD=EG6HdcoRqEbMWRsL ",
", ")
```

```
app = Flask(__name__)
app.secret_key="abc123"
```

```
def getList():
    products = []
    sql = "SELECT * FROM PRODUCTS"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        products.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
    return products
```

```
def getOrders():
    orders = []
    sql = "SELECT * FROM ORDERS"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        orders.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
    return orders
```

```
@app.route("/logout")
def logout():
    return redirect("/")
```



```
@app.route("/addProductpage")
def addProductpage():
    return render_template('addProduct.html')
```

```
@app.route("/orderpage")
def orderpage():
    orders = getOrders()
    return render_template('viewOrders.html', orders=orders)
```

```
@app.route('/delivery')
def delivery():
    name = session['user']
    role = session['role']
    email = session['email']
    products = getList()
    return render_template('home.html', name=name, role=role, products=products,
email=email,msg = "Product Dispatched Successfull to Customer !")
```

```
@app.route('/')
def login():
    return render_template('login.html')
```

```
@app.route('/register')
def register_Page():
    return render_template('register.html')
```

```
@app.route('/home/<name>/<role>/<email>')
def home_Page(name,role,email):
    products= getList()

    return render_template('home.html', name=name, role=role, products=products ,email=email)
```

```
@app.route('/back')
def back():
    name = session['user']
    role = session['role']
    email = session['email']
```

```
products = getList()
return render_template('home.html', name=name, role=role, products=products, email=email)
```

```
@app.route('/offerproduct')
def offProduct():
    products = getList()
    return render_template('offerproduct.html', products=products)
```

```
@app.route('/addproduct', methods=['GET', 'POST'])
def addproduct():
    pname = request.form['name']
    pimg = request.form['img']
    pprice = request.form['price']
    pcolor = request.form['color']
    pabout = request.form['about']
    poffprice = request.form['offprice']
    pcatagory = request.form['Catagory']
    pratting = request.form['ratting']
    pids = request.form['id']

    if request.method == 'POST':
        insert_sql = "INSERT INTO PRODUCTS VALUES (?,?,?,?,?,?,?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt, 1, pname)
        ibm_db.bind_param(prepare_stmt, 2, pimg)
        ibm_db.bind_param(prepare_stmt, 3, pprice)
        ibm_db.bind_param(prepare_stmt, 4, pcolor)
        ibm_db.bind_param(prepare_stmt, 5, pabout)
        ibm_db.bind_param(prepare_stmt, 6, poffprice)
        ibm_db.bind_param(prepare_stmt, 7, pcatagory)
        ibm_db.bind_param(prepare_stmt, 8, pratting)
        ibm_db.bind_param(prepare_stmt, 9, pids)
        ibm_db.bind_param(prepare_stmt, 10, 'nul')
        ibm_db.execute(prepare_stmt)
    name = session['user']
    role = session['role']
    email = session['email']
    products = getList()
```

```
    return render_template('home.html', name=name, role=role, products=products, email=email,
msg="Product Added Successfully!")
```

```
@app.route('/addUser', methods=['GET', 'POST'])
```

```
def addUser():
```

```
    Fname = request.form['fname']
```

```
    Lname = request.form['lname']
```

```
    email = request.form['email']
```

```
    password = request.form['password']
```

```
    re_Password = request.form['confirm_password']
```

```
    if password == re_Password:
```

```
        if request.method == 'POST':
```

```
            sql = "SELECT * FROM USERS WHERE EMAIL =?"
```

```
            stmt = ibm_db.prepare(conn, sql)
```

```
            ibm_db.bind_param(stmt, 1, email)
```

```
            ibm_db.execute(stmt)
```

```
            account = ibm_db.fetch_assoc(stmt)
```

```
            print(account)
```

```
            if account:
```

```
                return render_template('login.html', msg="You are already a member, please login using
your details")
```

```
            else:
```

```
                insert_sql = "INSERT INTO USERS VALUES (?, ?, ?, ?)"
```

```
                prep_stmt = ibm_db.prepare(conn, insert_sql)
```

```
                ibm_db.bind_param(prepare_stmt, 1, Fname)
```

```
                ibm_db.bind_param(prepare_stmt, 2, Lname)
```

```
                ibm_db.bind_param(prepare_stmt, 3, email)
```

```
                ibm_db.bind_param(prepare_stmt, 4, password)
```

```
                ibm_db.bind_param(prepare_stmt, 5, 'customer')
```

```
                ibm_db.execute(prepare_stmt)
```

```
                return render_template('login.html', msg="User Registered Successfully, Please Login")
```

```
            else:
```

```
                return render_template('register.html', msg = "password Not Match")
```

```
@app.route('/main', methods=['GET', 'POST'])
```

```
def loginUser():
```

```

email = request.form['email']
password = request.form['password']
if request.method == 'POST':
    sql = "SELECT * FROM USERS WHERE EMAIL=? AND PASSWORD=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, email)
    ibm_db.bind_param(stmt, 2, password)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)

    if account:
        # session['account'] = account
        name = account['FIRST_NAME']
        role = account['ROLE']
        email = account['EMAIL']
        session['email'] = account['EMAIL']
        session['user'] = account['FIRST_NAME']
        session['role'] = account['ROLE']
        session.modified = True
        return redirect(url_for('home_Page',name=name,role=role,email=email))
        # return render_template('home.html', user=account, msg="Login
successful!",products=products)
    else:
        return render_template('login.html', msg="Incorrect Email / Password")

```

```

@app.route('/setProduct/<id>')
def viewProduct(id):
    sql = "SELECT * FROM PRODUCTS WHERE SUB1 =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, id)
    ibm_db.execute(stmt)
    product = ibm_db.fetch_assoc(stmt)

    return render_template('viewProduct.html', product=product)

```

```

@app.route('/directBuy/<id>')
def directBuy(id):
    sql = "SELECT * FROM PRODUCTS WHERE SUB1 =?"

```

```

stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt, 1, id)
ibm_db.execute(stmt)
product = ibm_db.fetch_assoc(stmt)
name = product['NAME']
img = product['IMG']
color = product['COLOR']
price = product['PRICE']
user = session['user']
email = session['email']

insert_sql = "INSERT INTO ORDERS VALUES (?,?,?,?,?)"
prep_stmt = ibm_db.prepare(conn, insert_sql)
ibm_db.bind_param(prepare_stmt, 1, name)
ibm_db.bind_param(prepare_stmt, 2, img)
ibm_db.bind_param(prepare_stmt, 3, color)
ibm_db.bind_param(prepare_stmt, 4, price)
ibm_db.bind_param(prepare_stmt, 5, user)
ibm_db.bind_param(prepare_stmt, 6, email)
ibm_db.execute(prepare_stmt)

products = getList()
role = session['role']

return render_template('home.html', products=products, role=role,msg="Order placed
successfull")

@app.route('/addtocart/<id>')
def addtocart(id):
    sql = "SELECT * FROM PRODUCTS WHERE SUB1 =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, id)
    ibm_db.execute(stmt)
    product = ibm_db.fetch_assoc(stmt)
    name = product['NAME']
    img = product['IMG']
    color = product['COLOR']
    price = product['PRICE']

```

```

offPrice = product['OFFPRICE']
i = product['SUB1']

if 'cart' not in session:
    session['cart'] = []

session['cart'].append({'id':i,'name':name,'img':img,'color':color,'price':price,'offprice':offPrice})
session.modified = True

print(session['cart'])
print(len(session['cart']))
session['msg'] = "Product added successfully!"
session.modified = True

return redirect(url_for('viewProduct_', id=id))

@app.route('/setProduct_<id>')
def viewProduct_(id):
    sql = "SELECT * FROM PRODUCTS WHERE SUB1 =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, id)
    ibm_db.execute(stmt)
    product = ibm_db.fetch_assoc(stmt)

    return render_template('viewProduct.html', product=product, msg="Product Added Successfully")

@app.route('/viewCart')
def viewCart():
    products = session['cart']
    print(products[0]['name'])
    return render_template('cart.html',products=products)

```

Loign source code:

```
<body>

    {% if msg %}
    <div class="alert alert-primary alert-dismissible fade show" role="alert">
        {{msg}}
        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
    </div>
    {% endif %}

<div class="containes">
    <p style="color: rgb(0, 0, 0); text-align: center; font-size: 25px; font-weight: 600;"><span
style="color:blueviolet;">Styles</span> Ware</p>
    <div class="titles">Login</div>
    <div class="content">
        <form action="/main" method="post">
            <div class="user-details">
                <div class="input-box">
                    <span class="details">Email</span>
                    <input type="text" name="email" required>
                </div>
                <div class="input-box">
                    <span class="details">Password</span>
                    <input type="text" name="password" required>
                </div>
            </div>
            <a href="{{ url_for('register_Page') }}">Don't have Account ?</a>
            <div class="button">
                <input type="submit" value="Login">
            </div>
        </form>
    </div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA88Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

</body>
```

Register Source code:

```
<body>

    {% if msg %}
    <div class="alert alert-primary alert-dismissible fade show" role="alert">
        {{msg}}
        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
    </div>

    {% endif %}

<div class="container">
    <p style="color: rgb(0, 0, 0); text-align: center; font-size: 25px; font-weight: 600;"><span
style="color:blueviolet;">Styles</span> Ware</p>
    <div class="titles">Registration</div>
    <div class="content">
        <form action="/addUser" method="post">
            <div class="user-details">
                <div class="input-box">
                    <span class="details">First Name</span>
                    <input type="text" name="fname" required>
                </div>
                <div class="input-box">
                    <span class="details">Last Name</span>
                    <input type="text" name="lname" required>
                </div>
                <div class="input-box">
                    <span class="details">Email</span>
                    <input type="text" name="email" required>
                </div>
                <div class="input-box">
                    <span class="details">Phone Number</span>
                    <input type="text" name="mobile" required>
                </div>
                <div class="input-box">
                    <span class="details">Password</span>
                    <input type="text" name="password" required>
                </div>
                <div class="input-box">
                    <span class="details">Confirm Password</span>
                    <input type="text" name="confirm_password" required>
                </div>
            </div>
        </form>
    </div>
</div>
```



```

</div>
<a href="/">Already have an Account ?</a>
<div class="button">
  <input type="submit" value="Register">
</div>
</form>
</div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYIzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

</body>

```

Home page source code:

```

<body>

  <!--header start-->
  <div class="header">
    <p style="color: rgb(0, 0, 0); text-align: center; font-size: 28px; font-weight:
600;"><span style="color:blueviolet;">Styles</span> Ware</p>
    <div>
      <a href="/viewCart" style="color: black;"><i class='bx bxs-shopping-bag'></i></a>
      <!-- <span class="badge bg-primary " style="font-size: 14px;"></span> -->

      <div class="btn-group">
<button type="button" class="btn btn-primary dropdown-toggle" data-bs-toggle="dropdown" aria-
expanded="false">
          Account
        </button>
        <ul class="dropdown-menu">
          <li><a class="dropdown-item"
href="#">{{name}}{{role}}</a></li>
          <li><a class="dropdown-item" href="/addProductpage">Add
Product</a></li>
          <li><a class="dropdown-item" href="/orderpage">View
Orders</a></li>
        </ul>
      </div>
    </div>
  </div>

```

```

                <a href="/logout" style="font-size: 18px;" class="btn btn-danger">Logout</a>
            </div>
        </div>
        <div class="message">
            {% if msg %}
        <div class="alert alert-success alert-dismissible fade show" role="alert">
            {{msg}}
        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
        </div>
            {% endif %}
        </div>

        <div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">
            <div class="carousel-indicators">
                <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-
to="0" class="active" aria-current="true" aria-label="Slide 1"></button>
                <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-
to="1" aria-label="Slide 2"></button>
                <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-
to="2" aria-label="Slide 3"></button>
            </div>
            <div class="carousel-inner">
                <div class="carousel-item active">
                    
                </div>
                <div class="carousel-item">
                    
                </div>
                <div class="carousel-item">
                    
                </div>
            </div>
            <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleIndicators"
data-bs-slide="prev">
                <span class="carousel-control-prev-icon" aria-hidden="true"></span>
                <span class="visually-hidden">Previous</span>
            </button>

```

```

<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators"
data-bs-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Next</span>
</button>
</div>

```

```

<!--shop start-->
<section class="shop" id="shop">
  <h3 style="text-align:center; font-size: 28px;">Our Products</h3>
  <div class="container">
    {% for product in products %}
    <div class="box">
      
      <h4>{{product['NAME']}}</h4>
      <div class="price">
        <h5>₹{{product['PRICE']}}</h5>
        <p>₹{{product['OFFPRICE']}}</p>
      </div>
      <a href="/setProduct/{{product['SUB1']}}"
class="btn btn-primary"
>View Product</a>
      <div class="cart" title="Add to cart">
        <a href="#"><i class='bx bx-cart' ></i></a>
      </div>
    </div>
    {% endfor %}
  </div>
</section>

```

```

<!--about start-->
<section class="about" id="about">
  <div class="about-content">
    <h2>About</h2>
    <p>I'm a paragraph. Click here to add your own text and edit me. It's easy.
Just click "Edit Text" or double click me to add your own content and make changes to the font. I'm a
great place for you to tell a story and let your users know a little more about you.</p>
  </div>
</section>

```

```

<!--contact start-->

```

```
<section class="contact" id="contact">
  <div class="main-contact">
    <div class="contact-content">
      <li><a href="#shop">Shop</a></li>
      <li><a href="#about">About</a></li>
      <li><a href="#faq">Faq</a></li>
      <li><a href="#contact">Contact</a></li>
    </div>

    <div class="contact-content">
      <li><a href="#">Shipping & Returns</a></li>
      <li><a href="#">Store Policy</a></li>
      <li><a href="#">Payment Methods</a></li>
    </div>

    <div class="contact-content">
      <li><a href="#">Contact</a></li>
      <li><a href="#">Tel: 123-456-7890</a></li>
    </div>

    <div class="contact-content">
      <li><a href="#">Facebook</a></li>
      <li><a href="#">Instagram</a></li>
      <li><a href="#">Pinterest</a></li>
    </div>

  </div>

  <div class="action">
    <form>
      <input type="email" name="email" placeholder="Your Email" required>
      <input type="submit" name="submit" value="Submit" required>
    </form>
  </div>
</section>

<div class="last">
  <p>© 2023 by Tahmid. Proudly created with Tahmid Ahmed</p>
</div>

<!--custom js link-->
<script src="{ url_for('static', filename='home.js') }"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
```

```

integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
<script>
window.watsonAssistantChatOptions = {
  integrationID: "fc05e88a-9e7a-43a7-a707-3db931d45df6", // The ID of this integration.
  region: "au-syd", // The region your integration is hosted in.
  serviceInstanceID: "e5319e39-166f-463b-9da4-bd8699929c44", // The ID of your service instance.
  onLoad: function(instance) { instance.render(); }
};
setTimeout(function(){
  const t=document.createElement('script');
  t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
  document.head.appendChild(t);
});
</script>

</body>

```

Selected Product page:

```

<body>
  <div class="header">
    <p style="color: rgb(0, 0, 0); text-align: center; font-size: 28px; font-weight:
600;"><span style="color:blueviolet;">Styles</span> Ware</p>
    <div>
      <a href="/viewCart" style="color: black;"><i class='bx bxs-shopping-bag'
></i></a>
      <span class="badge bg-primary " style="font-size: 14px;">{{role}}</span>
      <a href="/logout" style="font-size: 18px;" class="btn btn-danger">Logout</a>
    </div>
  </div>
  <a href="/back" class="btn btn-dark" style="margin:10px 0px 0px 20px ;">back</a>
  <div class="message">
    {% if msg %}
    <div class="alert alert-success alert-dismissible fade show" role="alert">
      {{msg}}
      <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
    </div>
    {% endif %}
  </div>

```

```

<div class="viewBody">
  <div class="product_img">
    
  </div>
  <div class="product_details">
    <h3>{{product['NAME']}}</h3>
    <div class="price-part">
      <span>₹{{product['PRICE']}}</span>
      <p>₹{{product['OFFPRICE']}}</p>
    </div>
    <p>Color : <span class="badge bg-secondary " style="font-size:
14px;">{{product['COLOR']}}</span></P>
    <div class="catagory">
      <p>Catagory : <span class="badge bg-primary " style="font-size:
14px;">{{product['CATAGORY']}}</span></P>
      <p>Rating : <span class="badge bg-danger " style="font-size:
14px;">{{product['RATING']}}</span></P>
    </div>
    <p style="width:600px ;">{{product['ABOUT']}}</p>
    <div class="btn">
      <a class="btn" href="/addtocart/{{product['SUB1']}}">Add to Cart</a>
      <a class="btn" href="/directBuy/{{product['SUB1']}}">Buy Now</a>
    </div>
  </div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlrH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</body>

```

Cart page:

```

<body>
  <div class="header">
    <p style="color: rgb(0, 0, 0); text-align: center; font-size: 28px; font-weight:
600;"><span style="color:blueviolet;">Styles</span> Ware</p>
    <div>
      <a href="/viewCart" style="color: black;"><i class='bx bxs-shopping-bag'
></i></a>
      <span class="badge bg-primary " style="font-size: 14px;">{{role}}</span>
      <a href="/logout" style="font-size: 18px;" class="btn btn-danger">Logout</a>
    </div>
  </div>

```

```

        </div>
    </div>
    <a href="/back" class="btn btn-dark" style="margin:10px 0px 0px 20px ;">back</a>

<div class="message">
    {% if msg %}
    <div class="alert alert-success alert-dismissible fade show" role="alert">
        {{msg}}
        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
    </div>
    {% endif %}
</div>

<div class="container">
    <h2 style="text-align:center ; color: blueviolet; padding:20px ;">Cart List</h2>
        {% for product in products %}
        <div class="cart-item">
            <div class="img-cart">
                
            </div>
            <div class="cart-detail">
                <h2>{{product['name']}}</h2>
                <div class="price-part">
                    <span>₹{{product['price']}}</span>
                    <p>₹{{product['offprice']}}</p>
                </div>
                <P>Color : <span class="badge bg-secondary " style="font-size:
14px;">{{product['color']}}</span></P>
                <a class="btn" href="/directBuy/{{product['id']}}">Buy Now</a>
            </div>
        </div>
        {% endfor %}
    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</body>

```

Add Product page (ADMIN)

```
<body>
  <div class="header">
    <p style="color: rgb(0, 0, 0); text-align: center; font-size: 28px; font-weight:
600;"><span style="color:blueviolet;">Styles</span> Ware</p>
    <div>
      <a href="/viewCart" style="color: black;"><i class='bx bxs-shopping-bag'
></i></a>

      <span class="badge bg-primary " style="font-size: 14px;">{{role}}</span>
      <a href="/logout" style="font-size: 18px;" class="btn btn-danger">Logout</a>

    </div>
  </div>
  <a href="/back" class="btn btn-dark" style="margin:10px 0px 0px 20px ;">back</a>

<div class="message">
  {% if msg %}
  <div class="alert alert-success alert-dismissible fade show" role="alert">
    {{msg}}
    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
  </div>
  {% endif %}
</div>

<div class="container">
  <h2 style="text-align:center ; color: blueviolet; padding:20px ;">Add Products</h2>
  <div class="content">
    <form action="/addproduct" method="post">
      <div class="user-details">
        <div class="input-box">
          <span class="details">Product Name</span>
          <input type="text" name="name" maxlength="15" required>
        </div>

        <div class="input-box">
          <span class="details">Image Link</span>
          <input type="text" name="img" maxlength="100" required>
        </div>

        <div class="input-box">
          <span class="details">price</span>
          <input type="text" name="price" maxlength="10" required>
        </div>
      </div>
    </form>
  </div>
</div>
```



```
</div>

<div class="input-box">
  <span class="details">Color</span>
  <input type="text" name="color" maxlength="12" required>
</div>

<div class="input-box">
  <span class="details">About</span>
  <input type="text" name="about" maxlength="250" required>
</div>

<div class="input-box">
  <span class="details">Offer Price</span>
  <input type="text" name="offprice" maxlength="5" required>
</div>

<div class="input-box">
  <span class="details">Catagory</span>
  <input type="text" name="Catagory" maxlength="10" required>
</div>

<div class="input-box">
  <span class="details">Set Ratting</span>
  <input type="text" name="ratting" maxlength="10" required>
</div>

<div class="input-box">
  <span class="details">Set ID</span>
  <input type="text" name="id" maxlength="5" required>
</div>

</div>
<div class="button">
  <input type="submit" value="Add">
</div>
</form>
</div>
  </div>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</body>
```

View Order list (ADMIN)

```
<body>
  <div class="header">
    <p style="color: rgb(0, 0, 0); text-align: center; font-size: 28px; font-weight:
600;"><span style="color:blueviolet;">Styles</span> Ware</p>
    <div>
      <a href="/viewCart" style="color: black;"><i class='bx bxs-shopping-bag'
></i></a>
      <span class="badge bg-primary " style="font-size: 14px;">{{role}}</span>
      <a href="/logout" style="font-size: 18px;" class="btn btn-danger">Logout</a>
    </div>
  </div>
  <a href="/back" class="btn btn-dark" style="margin:10px 0px 0px 20px ;">back</a>

<div class="message">
  {% if msg %}
  <div class="alert alert-success alert-dismissible fade show" role="alert">
    {{msg}}
    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
  </div>
  {% endif %}
</div>

<div class="container">
  <h2 style="text-align:center ; color: blueviolet; padding:20px ;">Order List</h2>
  {% for order in orders %}
    <div class="cart-item">
      <div class="img-cart">
        
      </div>
      <div class="cart-detail">
        <h2>{{order['NAME']}}</h2>
        <div class="price-part">
          <p>₹{{order['PRICE']}}</p>
        </div>
        <P>Color : <span class="badge bg-secondary " style="font-size:
14px;">{{order['COLOR']}}</span></P>
        <div class="user">
          <p>UserName : <span>{{order['USER']}}</span> </p>
          <p>Email : <span> {{order['EMAIL']}}</span> </p>
        </div>
      </div>
    </div>
  </div>
  </div>
```

```
        <a class="btn" href="/delivery">Delivery now</a>
    </div>
        </div>
        {% endfor %}
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</body>
```

7. Github & Demo Video Link

GitHub link : https://github.com/IBM-EPBL/IBM-Project-52821-1661158210/tree/main/Final_Project%20Details

Demo Video Link : <https://www.youtube.com/watch?v=g3i92xNUTvU>