

CLOUD BASED SMART FASHION RECOMMENDER APPLICATION

A PROJECT REPORT

Submitted by

SOUNDAR M

SASIDHARAN S

RAMESHKRISHNAN M

NISHANTH R

Team ID : PNT2022TMID12171

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY



K.S.R COLLEGE OF ENGINEERING

(Autonomous)

TIRUCHENGODE - 637 215



ANNA UNIVERSITY: CHENNAI 600 025

NOVAMBER - 2022

K.S.R COLLEGE OF ENGINEERING

(Autonomous)

TIRUCHENGODE - 637 215

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**Smart Fashion Recommender Application**” is the bonafide work of “**SOUNDAR M(1921049), SASIDHARAN S (1921046), RAMESHKRISHNAN M (1921042), NISHANTH R (1921033)**” who carried out the project work under my Mentor.

SIGNATURE

Dr.G.SINGARAVEL.M.E.,Ph.D.

HEAD OF THE DEPARTMENT,

PROFESSOR & HEAD,

DEPARTMENT OF IT,

K.S.R. COLLEGE OF ENGINEERING,

TIRUCHENGODE - 637 215.

SIGNATURE MENTOR,

Mr.GOWTHAM.M.E CSE,

ASSISTANT PROFESSOR ,

DEPARTMENT OF IT,

K.S.R. COLLEGE OF ENGINEERING,

TIRUCHENGODE - 637 215.

Submitted for the Project viva-voce held on _____

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

We feel highly honoured to extend our sincere gratitude to our beloved Chairman, Lion Dr.K.S.RANGASAMY, MJF, K.S.R Educational Institutions and our Managing Trustee Mr. R.SRINIVASAN, B.B.M, MISTE, Aarthi Educational and Chairman Trust for providing all facilities to complete this project work.

We would like to acknowledge the constant and kind support provided by our highly Principal Dr. P.SENTHIL KUMAR, Ph.D.we feel highly elated to thank our respectable Head of the Department Dr. G.SINGARAVEL, M.E., Ph.D. and our project coordinator Dr. G.SINGARAVEL, M.E., Ph.D. who guided us and was a pillar of support for the successful completion of the project.

We are the most fortunate in having the opportunity to work under the Mentor Mr.Gowtham M.E CSE. express our sincere thanks to him.

Finally, I wish to thank all teaching and non-teaching staff members of the Department of Information Technology and my friends for their support towards the successful completion of this project.

I wish to acknowledge the help received from various departments and various individuals who are support to do my project in a successful manner.

SOUNDAR M

SASIDHARAN S

RAMESHKRISHNAN M

NISHANTH R

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Nowadays, fashion applications and e-commerce are growing more and more, and it also has some problems when finding the customer's wanted product in the web applications. Having a chatbot that understands the algorithm of a specific application can be of great aid. We are implementing such a chat bot in a web application, which is fed with the knowledge of the application's algorithm and helps the user completely from finding their needs to processing the payment and initiating delivery. It works as an advanced filter search that can bring the user what they want with the help of pictorial and named representation by getting simple user information and activities. The application also has two main UI interactions: one is the user panel and the other one is the admin panel. Users can interact with the chat bot to search for products, order them from the manufacturer or distributor through chatbot AI, and it can also make payment transactions, track the delivery, and so on. The admin interface enables the user to upload products' details, user details, orders and find how many products have been bought; supervise the stock availability; and interact with the buyer regarding the product reviews.

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 chat bot.

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 Chat bot regarding the products. Get the recommendations based on information provided by the user.

1.2 PURPOSE

- a) Using chatbot we can manage user's choices and orders.
- b) The chatbot can give recommendations to the users based on their interests.
- c) It can promote the best deals and offers on that day.
- d) It will store the customer's details and orders in the database.
- e) The chatbot will send a notification to customers if the order is confirmed.
- f) Chatbots can also help in collecting customer feedback.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

1. Fashion item representation

Traditional recommender systems such as Collaborative Filtering or Content-Based Filtering have difficulties in the fashion domain due to the sparsity of purchase data, or the insufficient detail about the visual appearance of the product in category names. Instead, more recent literature has leveraged models that capture a rich representation of fashion items through product images, text descriptions or customer reviews, or videos which are often learned through surrogate tasks like classification or product retrieval. However, learning product representations from such input data requires large datasets to generalize well across different image (or text) styles, attribute variations, etc. Furthermore, constructing a representation that learns which product features customers take most into account when evaluating fashion products is still an open research problem.

2. Fashion item compatibility

Training a model that is able to predict if two fashion items ‘go together,’ or directly combine several products into an outfit, is a challenging task. Different item compatibility signals studied in recent literature include co-purchase data, outfits composed by professional fashion designers, or combinations found by analyzing what people wear in social media pictures.

3. Personalization and fit

The best fashion product to recommend depends on factors such as the location where the outfit will be used, the season or occasion, or the cultural and social background of the customer. A challenging task in fashion recommendation systems is how to discover and integrate these disparate factors. Current research often tackles these tasks by utilizing large-scale social media data.

4. Interpretability and explanation

Most of the existing fashion recommender systems in the literature focus on improving predictive performance, treating the model as a black box. However, deploying accountable and interpretable systems able to explain their recommendations can foster user loyalty in the long term and improve the shopping experience

5. Discovering trends

Being able to forecast consumer preferences is valuable for fashion designers and retailers in order to optimize product-to-market fit, logistics and advertising.

2.2 REFERENCES

A Systematic Study on the Recommender Systems in the E-Commerce

Electronic commerce or e-commerce includes the service and good exchange through electronic support like the Internet. It plays a crucial role in today's business and users' experience. Also, e-commerce platforms produce a vast amount of information. So, Recommender Systems (RSs) are a solution to overcome the information overload problem. They provide personalized recommendations to improve user satisfaction. The present article illustrates a comprehensive and Systematic Literature Review (SLR) regarding the papers published in the field of e-commerce recommender systems. We reviewed the selected papers to identify the gaps and significant issues of the RSs' traditional methods, which guide the researchers to do future work. So, we provided the traditional techniques, challenges, and open issues concerning traditional methods of the field of review based on the selected papers. This review includes five categories of the RSs' algorithms, including Content-Based Filtering (CBF), Collaborative Filtering (CF), Demographic-Based Filtering (DBF), hybrid filtering, and Knowledge-Based Filtering (KBF).

2.3 Problem Statement Definition

Problem Statement 1:

The User Needs a way to Find Trending Fashion Clothes so that Here find the All Collections

Problem Statement 2:

The User Needs a way to Find Offers and Discounts so that Here User easy to find Daily Offers

Problem Statement 3:

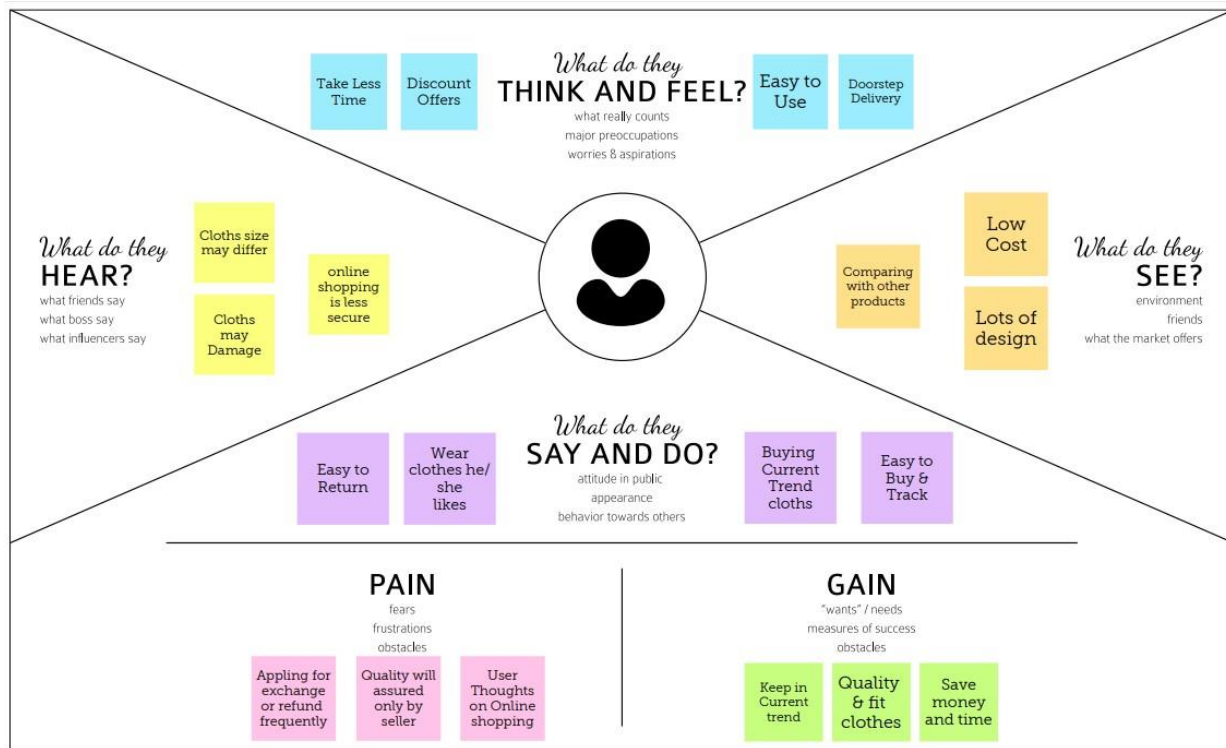
The User Needs a way to Assistant for finding Clothes so that Here User got the Chat Bot assistant

Problem Statement 4:

The Sellers Needs a way to struggling to sells products offline so that Here Sellers will Sell Products via our application.

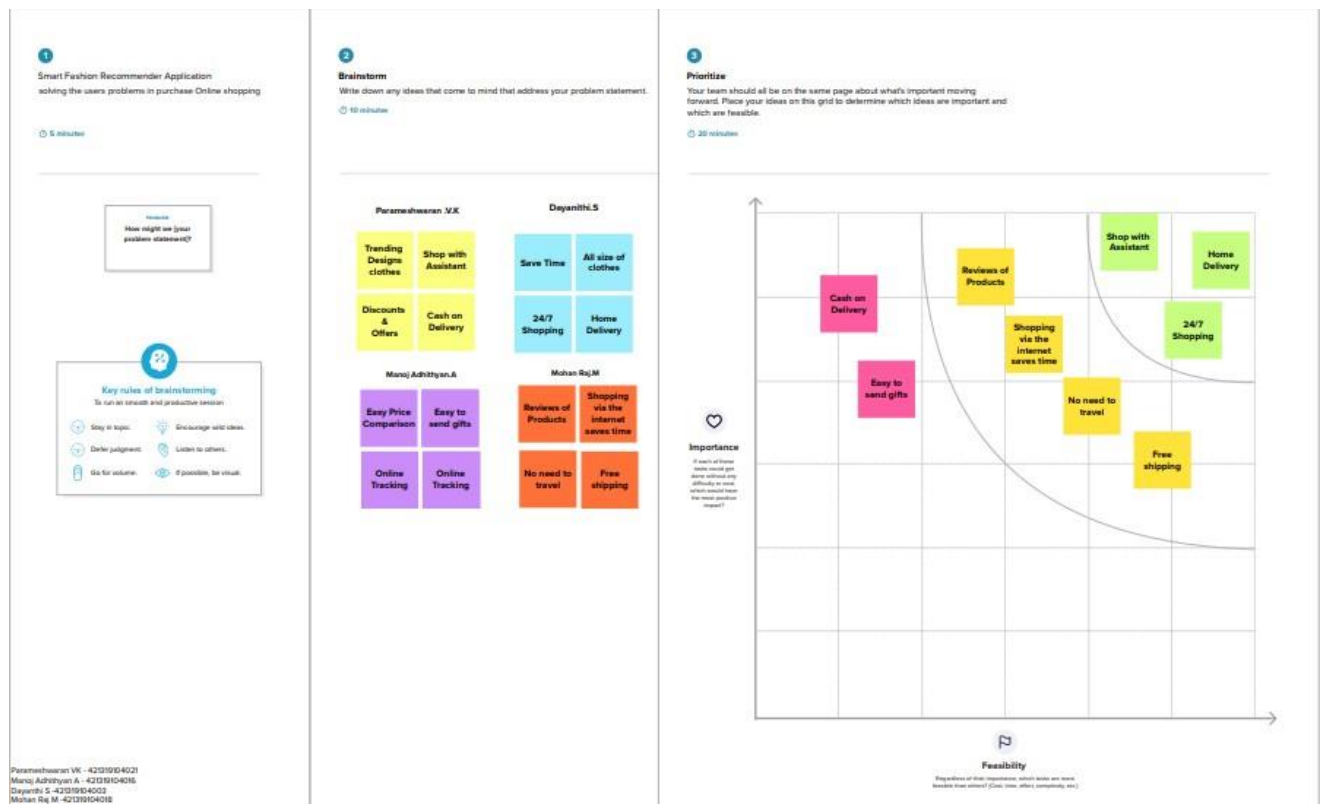
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2

3.2 Ideation & Brainstorming



3.3 Proposed Solution

S. No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Customers feels difficult when Search many websites to find Fashion clothes and accessories.
2.	Idea / Solution description	Customers directly make online shopping based on customer choice without any search.
3.	Novelty / Uniqueness	The customer will talk to Chat Bot regarding the Products. Get the recommendations based on information provided by the user
4.	Social Impact / Customer Satisfaction	The user friendly interface, Assistants form chat bot finding dress makes customer satisfied.
5.	Business Model (Revenue Model)	The chat bot sells our Products to customer. Customers buy our products and generate revenue
6.	Scalability of the Solution	We can easily scalable our Applications by increases the items and products

3.4 Problem Solution Fit

Project Title: *Smart Fashion Recommender Application*

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID38912

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Who is your customer? i.e. working parents of 0-5 y.o. kids <div>The Customers are Adults and children</div>	6. CUSTOMER CONSTRAINTS CC What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices <div>Money and Network Connection</div>	5. AVAILABLE SOLUTIONS AS Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking <div>Online shopping gives New Collections pros: Easy to use cons: customer confused when have lost of collections</div>	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. <div>Users hard to find Trending Fashion Clothes.</div>	9. PROBLEM ROOT CAUSE RC What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations. <div>Customers need to be with new fashions for current trends</div>	7. BEHAVIOUR BE What does your customer do to address the problem and get the job done? (i.e. Directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)) <div>Customers spend the time to find the new fashion clothes</div>	
Focus on J&P, map into BE, understand RC	3. TRIGGERS TR What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news. <div>Seeing neighbor Dressing Styles</div>	10. YOUR SOLUTION SL If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour. <div>Make a ChatBot Assistant for shopping with customers and send notifications when new collections arrived</div>	8. CHANNELS of BEHAVIOUR CH K1 ONLINE What kind of actions do customers take online? Extract online channels from #7 K2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. <div>ONLINE: Customers buy the new clothes OFFLINE: Customers will use the clothes</div>	Identify strong TR & EM
	4. EMOTIONS: BEFORE / AFTER EM How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure -> confident, in control - use it in your communication strategy & design. <div>Felling Sad and Frustration > Selfconfident</div>			

4. REQUIREMENT ANALYSIS

4.1 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 the Chat Bot
FR-3	Buying Products	Through the chat Bot Recommendation
FR-4	Track Products	Ask the Chat Bot to Track my Orders
FR-5	Return Products	Through the chat Bot
FR_6	New Collections	Recommended from chat Bot

4.2 Non-functional Requirements:

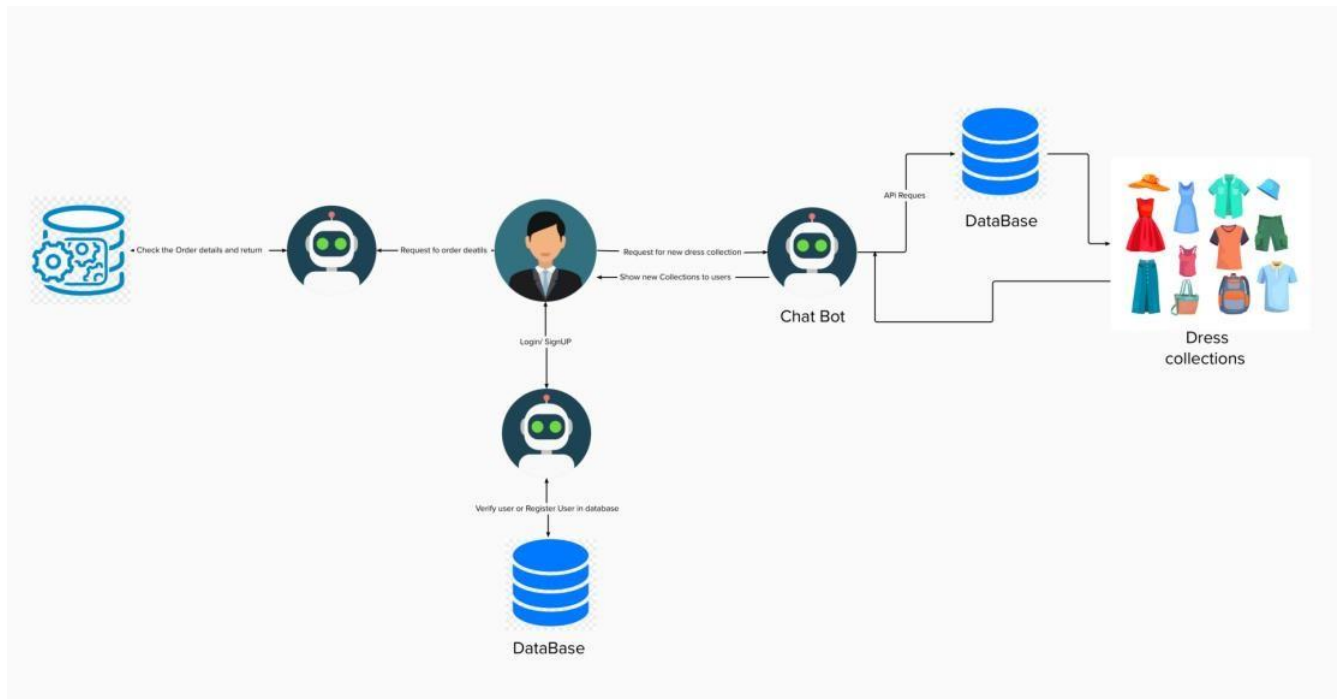
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Using Android or IOS or windows applications.
NFR-2	Security	The user data is stored securely in IBM cloud.
NFR-3	Reliability	The Quality of the services are trusted.
NFR-4	Performance	Its Provide smooth user experience.
NFR-5	Availability	The services are available for 24/7.
NFR-6	Scalability	It's easy to scalable size of users and products.

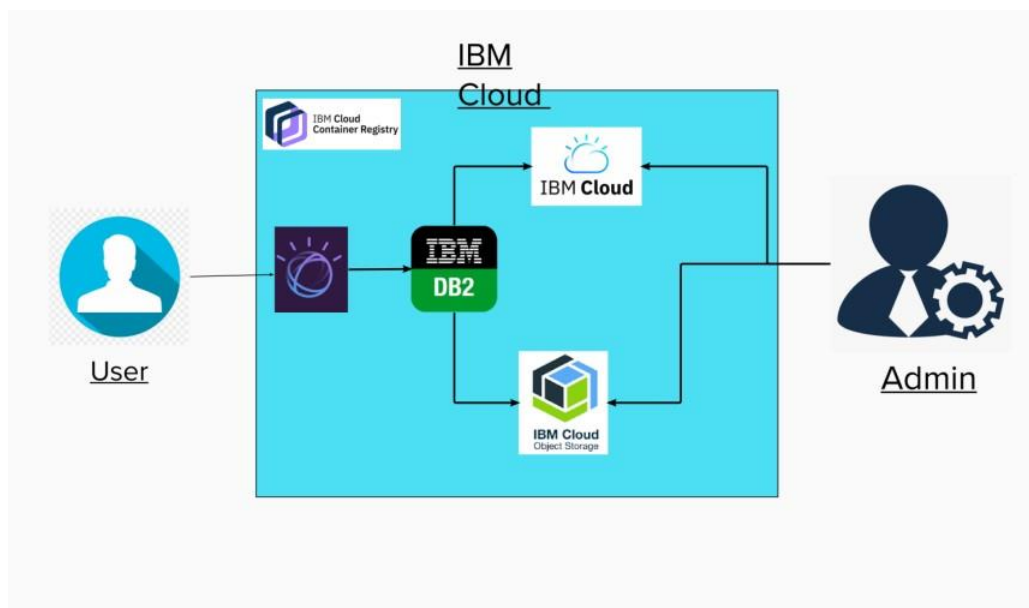
5. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Stories

Use the below template to list all the user stories for the product.

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	I can access my data by login	High	Sprint-1
	Dashboard	USN-6	As a user , I can view the dashboard and by products		High	Sprint -2
Customer (Web user)	Registration / Login	USN-7	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard		Sprint -1
Customer Care Executive	Contact with Customers	USN-8	As a Customer customers care executive, I solve the customer Requirements and feedback	I can receive calls from customers	High	Sprint-1
Administrator	Check stock and Price , orders	USN_9	As a Administrator , I can Check the database And stock details and buying and selling prices	I am the administrator of the company	High	Sprint -2

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

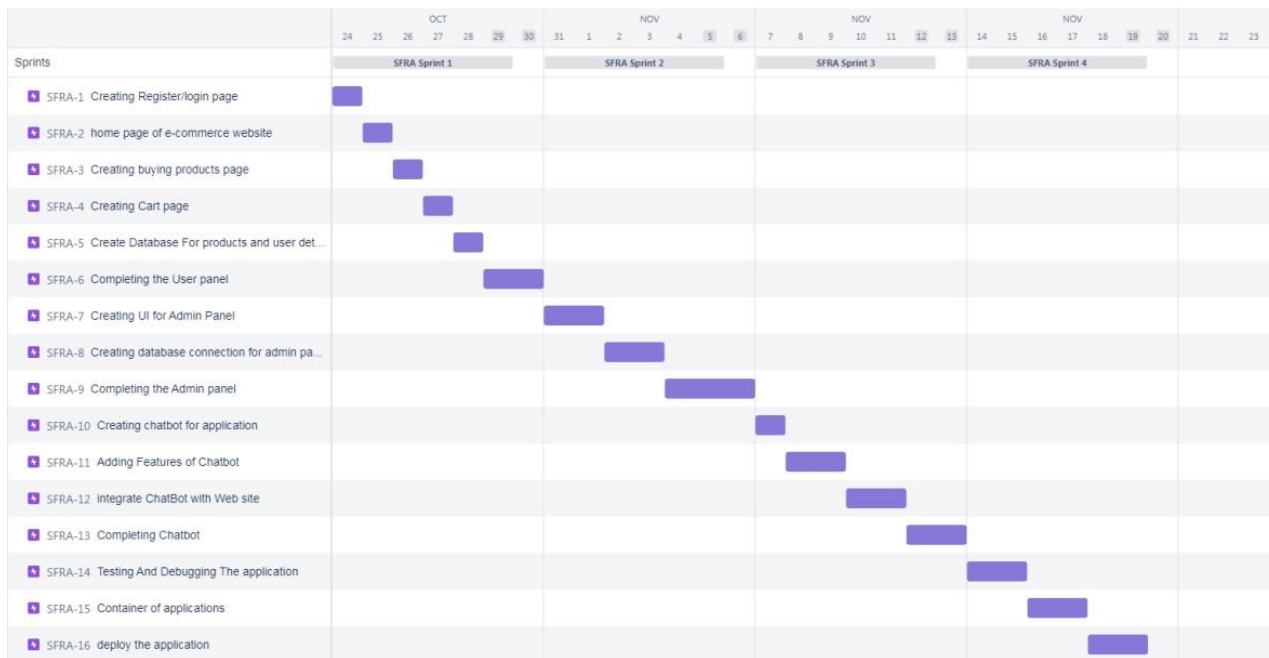
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Panel	USN-1	The user will login into the website and go through the products available on the website	20	High	PARAMESHWARAN V K MANOJ ADHITHYAN A MOHANRAJ M DAYANITHI S
Sprint-2	Admin panel	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	PARAMESHWARAN V K MANOJ ADHITHYAN A MOHANRAJ M DAYANITHI S
Sprint-3	Chat Bot	USN-3	The user can directly talk to Chatbot regarding the products. Get the recommendations based on information provided by the user.	20	High	PARAMESHWARAN V K MANOJ ADHITHYAN A MOHANRAJ M DAYANITHI S
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	PARAMESHWARAN V K MANOJ ADHITHYAN A MOHANRAJ M DAYANITHI S

6.2 SPRINT DELIEVERY SCHEDULE

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	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORTS FROM JIRA

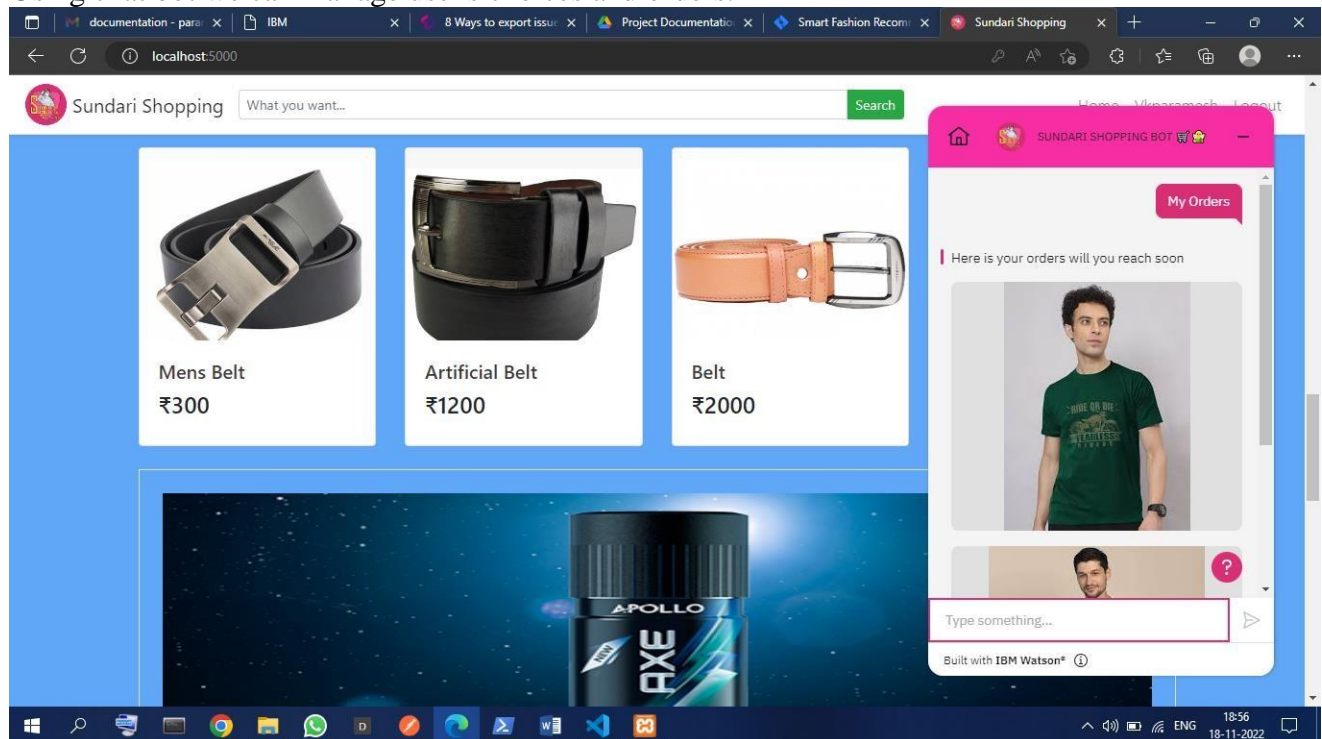
Burndown Chart:



7. CODING & SOLUTIONING

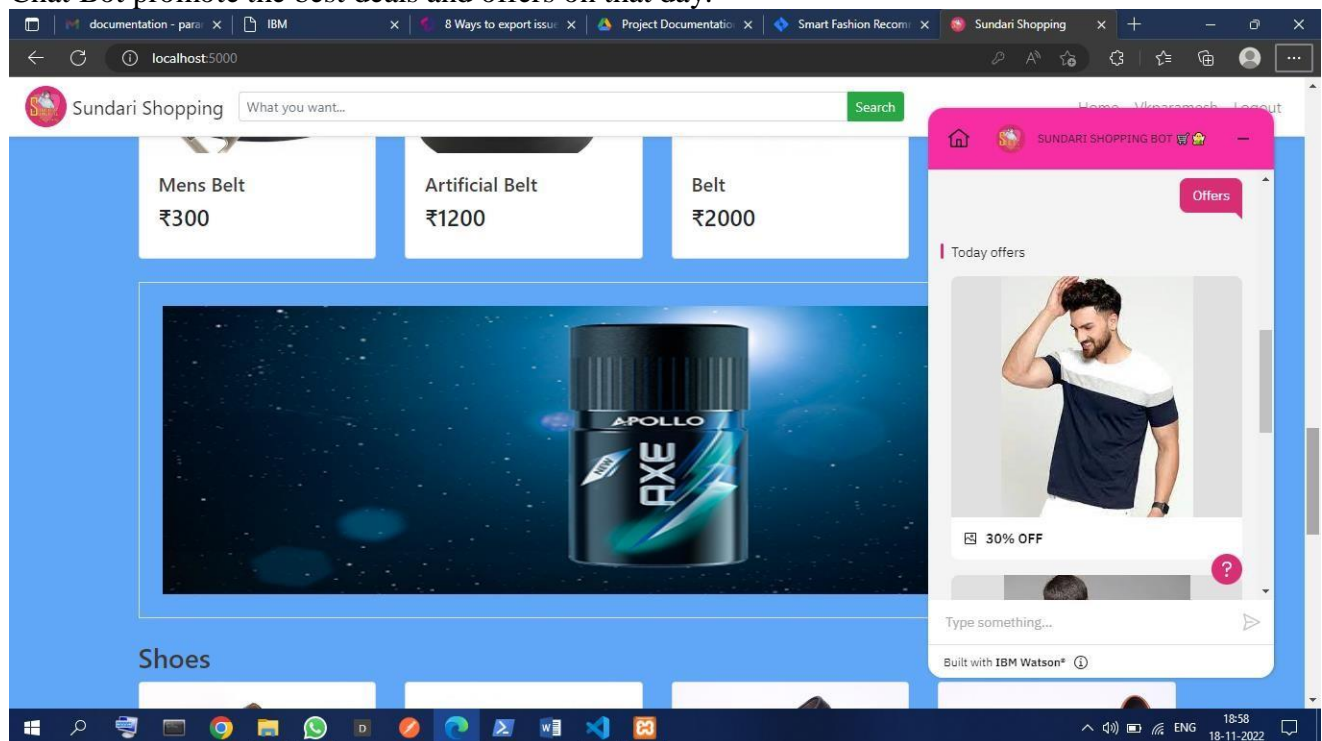
7.1 Feature 1

Using chat bot we can manage user's choices and orders.



7.2 Feature 2

Chat Bot promote the best deals and offers on that day.



7.3 Database Schema

The screenshot displays the IBM Db2 on Cloud console interface. The top navigation bar includes tabs for 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is currently selected.

Below the navigation bar, there is a search bar labeled 'Find schemas or tables' and a 'Refresh' button. The main content area is divided into two panels:

- Schemas:** This panel shows a table with columns 'Name', 'Type', and 'Tables'. It contains one entry: 'DTP46044' (User) with 5 tables. The status at the bottom indicates 'Total: 1, selected: 1'.
- Tables:** This panel shows a table with columns 'Name', 'Schema', and 'Properties'. It lists five tables: 'ORDERS', 'PRODUCTS', 'PRODUCT_LEVEL', 'PRODUCT_VIEW', and 'USERS', all belonging to the 'DTP46044' schema. The status at the bottom indicates 'Total: 5, selected: 0'.

The bottom of the image shows a Windows taskbar with various application icons and a system tray displaying the time as 19:15 on 18-11-2022.

8. TESTING

8.1 Test Cases

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Login	5	0	0	5
Register	7	0	0	7
Home Page	2	0	0	2
Order page	3	0	0	3
Order products	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

8.2 USER ACCEPTANCE TESTING

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Smart Fashion Recommender Application project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	5	2	3	21
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

9. RESULT

9.1 PERFORMANCE METRICS

Project team shall fill the following information in model performance testing.

NFT - Risk Assessment									
S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volumem Changes	Risk Score	Justification
1	Smart Fashion Recommender Application	New	Low	No Changes	Moderate		>\$ to 10%	ORANGE	As we have seen the changes

NFT - Detailed Test Plan				
S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	Smart Fashion Recommender Application	Manual testing	laptop or mobile with internet connection	vikparameshwaran

End Of Test Report							
S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)
1	Smart Fashion Recommender Application	Manule		Worked as we expected		Use Laptop / desktop Mode	No Defects
							Vikparameshwaran

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Its helps to user Shopping with Assistant
- Its helps to user manage there order list
- Its helps to user shopping at home

DISADVANTAGES:

- User have fear about online shopping
- User have sometimes receives wrong items
- User have fear about online payment

11. CONCLUSION

Recommendation systems have the potential to explore new opportunities for retailers by enabling them to provide customized recommendations to consumers based on information retrieved from the Internet. They help consumers to instantly find the products and services that closely match with their choices. Moreover, different state-of-the-art algorithms have been developed to recommend products based on users' interactions with their social groups. Therefore, research on embedding social media images within fashion recommendation systems has gained huge popularity in recent times. This paper presented a review of the fashion recommendation systems, algorithmic models and filtering techniques based on the academic articles related to this topic. The technical aspects, strengths and weaknesses of the filtering techniques have been discussed elaborately, which will help future researchers gain an in-depth understanding of fashion recommender systems. However, the proposed prototypes should be tested in commercial applications to understand their feasibility and accuracy in the retail market, because inaccurate recommendations can produce a negative impact on a customer. Moreover, future research should concentrate on including time series analysis and accurate categorization of product images based on the variation in color, trend and clothing style in order to develop an effective recommendation system. The proposed model will follow brand specific personalization campaigns and hence it will ensure highly curated and tailored offerings for users. Hence, this research will be highly beneficial for researchers interested in using augmented and virtual reality features to develop recommendation systems.

12. FUTURE SCOPE

There has been significant progress recently in fashion recommendation system research, which will benefit both consumers and retailers soon. The use of product and user images, textual content, demographic history, and cultural information is crucial in developing recommendation frameworks. Product attributes and clothing style matching are common features of collaborative and content-based filtering techniques. Researchers can develop more sophisticated hyper personalized filtering techniques considering the correlation between consumers' clothing styles and personalities. The methods based on employing a scoring system for quantifying each product attribute will be helpful in increasing the precision of the model. The use of virtual sales advisers in an online shopping portal would provide consumers with a real time offline shopping experience. Retailers can collect the data on users' purchase history and product reviews from the recommendation system and subsequently use them in style prediction for the upcoming seasons. The integration of different domain information strengthens the deep learning paradigm by enabling the detection of design component variation, which improves the performance of the recommendation system in the long run. Deep learning approaches should be more frequently used to quickly explore fashion items from different online databases to provide prompt recommendations to users or consumers.

12. APPENDIX

1. Source Code

App.py

```
from flask import Flask, render_template, flash, redirect, url_for, session, request, logging
from sqlalchemy import sqlalchemy
from wtforms import Form, StringField, TextAreaField, PasswordField, validators, SelectField
from passlib.hash import sha256_crypt
from functools import wraps
from flask_uploads import UploadSet, configure_uploads, IMAGES
import timeit
import datetime
from flask_mail import Mail, Message
import os
from wtforms import EmailField
import sendgrid
import os

db = sqlalchemy.create_engine('ibm_db_sa://dtp46044:95soX0sZGhb4ToUj@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:50000/bludb')
bcrypt = Bcrypt(app)
metadata = MetaData()

app = Flask(__name__)
app.secret_key = os.urandom(24)
app.config['UPLOADED_PHOTOS_DEST'] = 'static/image/product'
photos = UploadSet('photos', IMAGES)
configure_uploads(app, photos)

db.init_app(app)

def sendemail():
    sg = sendgrid.SendGridAPIClient(api_key=os.environ.get('SENDGRID_API_KEY'))
    data = {
        "personalizations": [
            {
                "to": [
                    {
                        "email": "vkdj3232@gmail.com"
                    }
                ]
            }
        ],
    },
```

```

        "subject": "Sundari Shopping.Thankyou for purchasing"
    }
],
"from": {
    "email": "parameshvk3232@gmail.com"
},
"content": [
    {
        "type": "text/plain",
        "value": "Oder successfull your order arive within 5 working days"
    }
]
}
response = sg.client.mail.send.post(request_body=data)
print(response.status_code)
print(response.body)
print(response.headers)

def is_logged_in(f):
    @wraps(f)
    def wrap(*args, **kwargs):
        if 'logged_in' in session:
            return f(*args, *kwargs)
        else:
            return redirect(url_for('login'))

    return wrap

def not_logged_in(f):
    @wraps(f)
    def wrap(*args, **kwargs):
        if 'logged_in' in session:
            return redirect(url_for('index'))
        else:
            return f(*args, *kwargs)

    return wrap

def is_admin_logged_in(f):
    @wraps(f)
    def wrap(*args, **kwargs):
        if 'admin_logged_in' in session:

```



```

        return f(*args, *kwargs)
    else:
        return redirect(url_for('admin_login'))

    return wrap

def not_admin_logged_in(f):
    @wraps(f)
    def wrap(*args, **kwargs):
        if 'admin_logged_in' in session:
            return redirect(url_for('admin'))
        else:
            return f(*args, *kwargs)

    return wrap

def wrappers(func, *args, **kwargs):
    def wrapped():
        return func(*args, **kwargs)

    return wrapped

def content_based_filtering(product_id):
    cur = db.connection.cursor()
    cur.execute("SELECT * FROM products WHERE id=%s", (product_id,)) # getting id row
    data = cur.fetchone() # get row info
    data_cat = data['category'] # get id category ex shirt
    print('Showing result for Product Id: ' + product_id)
    category_matched = cur.execute("SELECT * FROM products WHERE category=%s", (data_cat,)) # get all
shirt category
    print("Total product matched: " + str(category_matched))
    cat_product = cur.fetchall() # get all row
    cur.execute("SELECT * FROM product_level WHERE product_id=%s", (product_id,)) # id level info
    id_level = cur.fetchone()
    recommend_id = []
    cate_level = ['v_shape', 'polo', 'clean_text', 'design', 'leather', 'color', 'formal', 'converse', 'loafer', 'hook',
        'chain']
    for product_f in cat_product:
        cur.execute("SELECT * FROM product_level WHERE product_id=%s", (product_f['id'],))
        f_level = cur.fetchone()
        match_score = 0
        if f_level['product_id'] != int(product_id):

```

```

        for cat_level in cate_level:
            if f_level[cat_level] == id_level[cat_level]:
                match_score += 1
            if match_score == 11:
                recommend_id.append(f_level['product_id'])
    print("Total recommendation found: " + str(recommend_id))
    if recommend_id:
        cur = db.connection.cursor()
        placeholders = ','.join((str(n) for n in recommend_id))
        query = 'SELECT * FROM products WHERE id IN (%s)' % placeholders
        cur.execute(query)
        recommend_list = cur.fetchall()
        return recommend_list, recommend_id, category_matched, product_id
    else:
        return ""

@app.route('/')
@is_logged_in
def index():
    form = OrderForm(request.form)
    # Create cursor
    cur = db.connection.cursor()
    # Get message
    values = 'tshirt'
    cur.execute("SELECT * FROM products WHERE category=%s ORDER BY RAND() LIMIT 4", (values,))
    tshirt = cur.fetchall()
    values = 'wallet'
    cur.execute("SELECT * FROM products WHERE category=%s ORDER BY RAND() LIMIT 4", (values,))
    wallet = cur.fetchall()
    values = 'belt'
    cur.execute("SELECT * FROM products WHERE category=%s ORDER BY RAND() LIMIT 4", (values,))
    belt = cur.fetchall()
    values = 'shoes'
    cur.execute("SELECT * FROM products WHERE category=%s ORDER BY RAND() LIMIT 4", (values,))
    shoes = cur.fetchall()
    # Close Connection
    cur.close()
    return render_template('home.html', tshirt=tshirt, wallet=wallet, belt=belt, shoes=shoes, form=form)

class LoginForm(Form): # Create Login Form
    username = StringField("", [validators.length(min=1)],
        render_kw={'autofocus': True, 'placeholder': 'Username'})
    password = PasswordField("", [validators.length(min=3)],

```

```

        render_kw={'placeholder': 'Password'})

# User Login
@app.route('/login', methods=['GET', 'POST'])
@not_logged_in
def login():
    form = LoginForm(request.form)
    if request.method == 'POST' and form.validate():
        # GEt user form
        username = form.username.data
        # password_candidate = request.form['password']
        password_candidate = form.password.data

        # Create cursor
        cur = db.connection.cursor()

        # Get user by username
        result = cur.execute("SELECT * FROM users WHERE username=%s", [username])

        if result > 0:
            # Get stored value
            data = cur.fetchone()
            password = data['password']
            uid = data['id']
            name = data['name']

            # Compare password
            if sha256_crypt.verify(password_candidate, password):
                # passed
                session['logged_in'] = True
                session['uid'] = uid
                session['s_name'] = name
                x = '1'
                cur.execute("UPDATE users SET online=%s WHERE id=%s", (x, uid))

                return redirect(url_for('index'))

            else:
                flash('Incorrect password', 'danger')
                return render_template('login.html', form=form)

        else:
            flash('Username not found', 'danger')

        # Close connection

```

```

        cur.close()
        return render_template('login.html', form=form)
    return render_template('login.html', form=form)

@app.route('/out')
def logout():
    if 'uid' in session:
        # Create cursor
        cur = db.connection.cursor()
        uid = session['uid']
        x = '0'
        cur.execute("UPDATE users SET online=%s WHERE id=%s", (x, uid))
        session.clear()
        flash('You are logged out', 'success')
        return redirect(url_for('index'))
    return redirect(url_for('login'))

class RegisterForm(Form):
    name = StringField(" ", [validators.length(min=3, max=50)],
                        render_kw={'autofocus': True, 'placeholder': 'Full Name'})
    username = StringField(" ", [validators.length(min=3, max=25)], render_kw={'placeholder': 'Username'})
    email = EmailField(" ", [validators.DataRequired(), validators.Email(), validators.length(min=4, max=25)],
                      render_kw={'placeholder': 'Email'})
    password = PasswordField(" ", [validators.length(min=3)],
                             render_kw={'placeholder': 'Password'})
    mobile = StringField(" ", [validators.length(min=11, max=15)], render_kw={'placeholder': 'Mobile'})

@app.route('/register', methods=['GET', 'POST'])
@not_logged_in
def register():
    form = RegisterForm(request.form)
    if request.method == 'POST' and form.validate():
        name = form.name.data
        email = form.email.data
        username = form.username.data
        password = sha256_crypt.encrypt(str(form.password.data))
        mobile = form.mobile.data
        # insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
        # prep_stmt = ibm_db.prepare(conn, insert_sql)
        # ibm_db.bind_param(prepare_stmt, 1, username)
        # ibm_db.bind_param(prepare_stmt, 2, email)
        # ibm_db.bind_param(prepare_stmt, 3, password)

```

```

# ibm_db.execute(prepare_stmt)

# Create Cursor
cur = db.connection.cursor()
cur.execute("INSERT INTO users(name, email, username, password, mobile) VALUES(%s, %s, %s, %s, %s)",
            (name, email, username, password, mobile))

# Commit cursor
db.connection.commit()

# Close Connection
cur.close()

flash('You are now registered and can login', 'success')

return redirect(url_for('index'))
return render_template('register.html', form=form)

class MessageForm(Form): # Create Message Form
    body = StringField("", [validators.length(min=1)], render_kw={'autofocus': True})

@app.route('/chatting/<string:id>', methods=['GET', 'POST'])
def chatting(id):
    if 'uid' in session:
        form = MessageForm(request.form)
        # Create cursor
        cur = db.connection.cursor()

        # lid name
        get_result = cur.execute("SELECT * FROM users WHERE id=%s", [id])
        l_data = cur.fetchone()
        if get_result > 0:
            session['name'] = l_data['name']
            uid = session['uid']
            session['lid'] = id

        if request.method == 'POST' and form.validate():
            txt_body = form.body.data
            # Create cursor
            cur = db.connection.cursor()
            cur.execute("INSERT INTO messages(body, msg_by, msg_to) VALUES(%s, %s, %s)",
                        (txt_body, id, uid))

```

```

        # Commit cursor
        db.connection.commit()

    # Get users
    cur.execute("SELECT * FROM users")
    users = cur.fetchall()

    # Close Connection
    cur.close()
    return render_template('chat_room.html', users=users, form=form)
else:
    flash('No permission!', 'danger')
    return redirect(url_for('index'))
else:
    return redirect(url_for('login'))

@app.route('/chats', methods=['GET', 'POST'])
def chats():
    if 'lid' in session:
        id = session['lid']
        uid = session['uid']
        # Create cursor
        cur = db.connection.cursor()
        # Get message
        cur.execute("SELECT * FROM messages WHERE (msg_by=%s AND msg_to=%s) OR (msg_by=%s AND msg_to=%s) "
                    "ORDER BY id ASC", (id, uid, uid, id))
        chats = cur.fetchall()
        # Close Connection
        cur.close()
        return render_template('chats.html', chats=chats, )
    return redirect(url_for('login'))

class OrderForm(Form): # Create Order Form
    name = StringField(", [validators.length(min=1), validators.DataRequired()],
                        render_kw={'autofocus': True, 'placeholder': 'Full Name'})
    mobile_num = StringField(", [validators.length(min=1), validators.DataRequired()],
                             render_kw={'autofocus': True, 'placeholder': 'Mobile'})
    quantity = SelectField(", [validators.DataRequired()],
                           choices=[('1', '1'), ('2', '2'), ('3', '3'), ('4', '4'), ('5', '5')])
    order_place = StringField(", [validators.length(min=1), validators.DataRequired()],
                              render_kw={'placeholder': 'Order Place'})

```

```

@app.route('/tshirt', methods=['GET', 'POST'])
def tshirt():
    form = OrderForm(request.form)
    # Create cursor
    cur = db.connection.cursor()
    # Get message
    values = 'tshirt'
    cur.execute("SELECT * FROM products WHERE category=%s ORDER BY id ASC", (values,))
    products = cur.fetchall()
    # Close Connection
    cur.close()
    if request.method == 'POST' and form.validate():
        name = form.name.data
        mobile = form.mobile_num.data
        order_place = form.order_place.data
        quantity = form.quantity.data
        pid = request.args['order']
        now = datetime.datetime.now()
        week = datetime.timedelta(days=7)
        delivery_date = now + week
        now_time = delivery_date.strftime("%y-%m-%d %H:%M:%S")
        # Create Cursor
        curs = db.connection.cursor()
        if 'uid' in session:
            uid = session['uid']
            curs.execute("INSERT INTO orders(uid, pid, ofname, mobile, oplace, quantity, ddate) "
                        "VALUES(%s, %s, %s, %s, %s, %s, %s)",
                        (uid, pid, name, mobile, order_place, quantity, now_time))
        else:
            curs.execute("INSERT INTO orders(pid, ofname, mobile, oplace, quantity, ddate) "
                        "VALUES(%s, %s, %s, %s, %s, %s)",
                        (pid, name, mobile, order_place, quantity, now_time))
        # Commit cursor
        db.connection.commit()

        # Close Connection
        cur.close()
        sendemail()

        flash('Order successful', 'success')
        return render_template('tshirt.html', tshirt=products, form=form)
    if 'view' in request.args:
        product_id = request.args['view']
        curso = db.connection.cursor()
        curso.execute("SELECT * FROM products WHERE id=%s", (product_id,))

```

```

product = curso.fetchall()
x = content_based_filtering(product_id)
wrappered = wrappers(content_based_filtering, product_id)
execution_time = timeit.timeit(wrappered, number=0)
# print('Execution time: ' + str(execution_time) + ' usec')
if 'uid' in session:
    uid = session['uid']
    # Create cursor
    cur = db.connection.cursor()
    cur.execute("SELECT * FROM product_view WHERE user_id=%s AND product_id=%s", (uid,
product_id))
    result = cur.fetchall()
    if result:
        now = datetime.datetime.now()
        now_time = now.strftime("%y-%m-%d %H:%M:%S")
        cur.execute("UPDATE product_view SET date=%s WHERE user_id=%s AND product_id=%s",
            (now_time, uid, product_id))
    else:
        cur.execute("INSERT INTO product_view(user_id, product_id) VALUES(%s, %s)", (uid,
product_id))
        db.connection.commit()
    return render_template('view_product.html', x=x, tshirts=product)
elif 'order' in request.args:
    product_id = request.args['order']
    curso = db.connection.cursor()
    curso.execute("SELECT * FROM products WHERE id=%s", (product_id,))
    product = curso.fetchall()
    x = content_based_filtering(product_id)
    return render_template('order_product.html', x=x, tshirts=product, form=form)
return render_template('tshirt.html', tshirt=products, form=form)

@app.route('/wallet', methods=['GET', 'POST'])
def wallet():
    form = OrderForm(request.form)
    # Create cursor
    cur = db.connection.cursor()
    # Get message
    values = 'wallet'
    cur.execute("SELECT * FROM products WHERE category=%s ORDER BY id ASC", (values,))
    products = cur.fetchall()
    # Close Connection
    cur.close()

    if request.method == 'POST' and form.validate():

```



```

name = form.name.data
mobile = form.mobile_num.data
order_place = form.order_place.data
quantity = form.quantity.data
pid = request.args['order']

now = datetime.datetime.now()
week = datetime.timedelta(days=7)
delivery_date = now + week
now_time = delivery_date.strftime("%y-%m-%d %H:%M:%S")
# Create Cursor
curs = db.connection.cursor()

if 'uid' in session:
    uid = session['uid']
    curs.execute("INSERT INTO orders(uid, pid, ofname, mobile, oplace, quantity, ddate) "
                  "VALUES(%s, %s, %s, %s, %s, %s, %s)",
                  (uid, pid, name, mobile, order_place, quantity, now_time))
else:
    curs.execute("INSERT INTO orders(pid, ofname, mobile, oplace, quantity, ddate) "
                  "VALUES(%s, %s, %s, %s, %s, %s)",
                  (pid, name, mobile, order_place, quantity, now_time))

# Commit cursor
db.connection.commit()
# Close Connection

cur.close()
# sendemail()
sendemail()

flash('Order successful', 'success')
return render_template('wallet.html', wallet=products, form=form)
if 'view' in request.args:
    q = request.args['view']
    product_id = q
    x = content_based_filtering(product_id)
    curso = db.connection.cursor()
    curso.execute("SELECT * FROM products WHERE id=%s", (q,))
    products = curso.fetchall()
    return render_template('view_product.html', x=x, tshirts=products)
elif 'order' in request.args:
    product_id = request.args['order']
    curso = db.connection.cursor()
    curso.execute("SELECT * FROM products WHERE id=%s", (product_id,))
    product = curso.fetchall()
    x = content_based_filtering(product_id)

```

```

        return render_template('order_product.html', x=x, tshirts=product, form=form)
    return render_template('wallet.html', wallet=products, form=form)

@app.route('/belt', methods=['GET', 'POST'])
def belt():
    form = OrderForm(request.form)
    # Create cursor
    cur = db.connection.cursor()
    # Get message
    values = 'belt'
    cur.execute("SELECT * FROM products WHERE category=%s ORDER BY id ASC", (values,))
    products = cur.fetchall()
    # Close Connection
    cur.close()

    if request.method == 'POST' and form.validate():
        name = form.name.data
        mobile = form.mobile_num.data
        order_place = form.order_place.data
        quantity = form.quantity.data
        pid = request.args['order']
        now = datetime.datetime.now()
        week = datetime.timedelta(days=7)
        delivery_date = now + week
        now_time = delivery_date.strftime("%y-%m-%d %H:%M:%S")
        # Create Cursor
        curs = db.connection.cursor()
        if 'uid' in session:
            uid = session['uid']
            curs.execute("INSERT INTO orders(uid, pid, ofname, mobile, oplace, quantity, ddate) "
                        "VALUES(%s, %s, %s, %s, %s, %s, %s)",
                        (uid, pid, name, mobile, order_place, quantity, now_time))
        else:
            curs.execute("INSERT INTO orders(pid, ofname, mobile, oplace, quantity, ddate) "
                        "VALUES(%s, %s, %s, %s, %s, %s)",
                        (pid, name, mobile, order_place, quantity, now_time))

        # Commit cursor
        db.connection.commit()

        # Close Connection
        cur.close()
        sendemail()
        flash('Order successful', 'success')

```

```

        return render_template('belt.html', belt=products, form=form)
    if 'view' in request.args:
        q = request.args['view']
        product_id = q
        x = content_based_filtering(product_id)
        curso = db.connection.cursor()
        curso.execute("SELECT * FROM products WHERE id=%s", (q,))
        products = curso.fetchall()
        return render_template('view_product.html', x=x, tshirts=products)
    elif 'order' in request.args:
        product_id = request.args['order']
        curso = db.connection.cursor()

        curso.execute("SELECT * FROM products WHERE id=%s", (product_id,))
        product = curso.fetchall()
        x = content_based_filtering(product_id)
        return render_template('order_product.html', x=x, tshirts=product, form=form)
    return render_template('belt.html', belt=products, form=form)

@app.route('/shoes', methods=['GET', 'POST'])
def shoes():
    form = OrderForm(request.form)
    # Create cursor
    cur = db.connection.cursor()
    # Get message
    values = 'shoes'
    cur.execute("SELECT * FROM products WHERE category=%s ORDER BY id ASC", (values,))
    products = cur.fetchall()
    # Close Connection
    cur.close()

    if request.method == 'POST' and form.validate():
        name = form.name.data
        mobile = form.mobile_num.data
        order_place = form.order_place.data
        quantity = form.quantity.data
        pid = request.args['order']
        now = datetime.datetime.now()
        week = datetime.timedelta(days=7)
        delivery_date = now + week
        now_time = delivery_date.strftime("%y-%m-%d %H:%M:%S")
        # Create Cursor
        curs = db.connection.cursor()
        if 'uid' in session:
            uid = session['uid']

```

```

        curs.execute("INSERT INTO orders(uid, pid, ofname, mobile, oplace, quantity, ddate) "
                    "VALUES(%s, %s, %s, %s, %s, %s, %s)",
                    (uid, pid, name, mobile, order_place, quantity, now_time))
    else:
        curs.execute("INSERT INTO orders(pid, ofname, mobile, oplace, quantity, ddate) "
                    "VALUES(%s, %s, %s, %s, %s, %s)",
                    (pid, name, mobile, order_place, quantity, now_time))

    # Commit cursor
    db.connection.commit()
    # Close Connection
    cur.close()
    sendemail()

    flash('Order successful', 'success')
    return render_template('shoes.html', shoes=products, form=form)
if 'view' in request.args:
    q = request.args['view']
    product_id = q
    x = content_based_filtering(product_id)
    curso = db.connection.cursor()
    curso.execute("SELECT * FROM products WHERE id=%s", (q,))
    products = curso.fetchall()
    return render_template('view_product.html', x=x, tshirts=products)
elif 'order' in request.args:
    product_id = request.args['order']
    curso = db.connection.cursor()
    curso.execute("SELECT * FROM products WHERE id=%s", (product_id,))
    product = curso.fetchall()
    x = content_based_filtering(product_id)
    return render_template('order_product.html', x=x, tshirts=product, form=form)
return render_template('shoes.html', shoes=products, form=form)

@app.route('/admin_login', methods=['GET', 'POST'])
@not_admin_logged_in
def admin_login():
    if request.method == 'POST':
        # GEt user form
        username = request.form['email']
        password_candidate = request.form['password']

        # Create cursor
        cur = db.connection.cursor()

        # Get user by username

```

```

result = cur.execute("SELECT * FROM admin WHERE email=%s", [username])

if result > 0:
    # Get stored value
    data = cur.fetchone()
    password = data['password']
    uid = data['id']
    name = data['firstName']

    # Compare password
    if (password):
        # passed
        session['admin_logged_in'] = True
        session['admin_uid'] = uid
        session['admin_name'] = name

        return redirect(url_for('admin'))

    else:
        flash('Incorrect password', 'danger')
        return render_template('pages/login.html')

else:
    flash('Username not found', 'danger')
    # Close connection
    cur.close()
    return render_template('pages/login.html')
return render_template('pages/login.html')

@app.route('/admin_out')
def admin_logout():
    if 'admin_logged_in' in session:
        session.clear()
        return redirect(url_for('admin_login'))
    return redirect(url_for('admin'))

@app.route('/admin')
@is_admin_logged_in
def admin():
    curso = db.connection.cursor()
    num_rows = curso.execute("SELECT * FROM products")
    result = curso.fetchall()
    order_rows = curso.execute("SELECT * FROM orders")

```

```

users_rows = curso.execute("SELECT * FROM users")
return render_template('pages/index.html', result=result, row=num_rows, order_rows=order_rows,
                        users_rows=users_rows)

@app.route('/orders')
@is_admin_logged_in
def orders():
    curso = db.connection.cursor()
    num_rows = curso.execute("SELECT * FROM products")
    order_rows = curso.execute("SELECT * FROM orders")
    result = curso.fetchall()
    users_rows = curso.execute("SELECT * FROM users")
    return render_template('pages/all_orders.html', result=result, row=num_rows, order_rows=order_rows,
                           users_rows=users_rows)

@app.route('/users')
@is_admin_logged_in
def users():
    curso = db.connection.cursor()
    num_rows = curso.execute("SELECT * FROM products")
    order_rows = curso.execute("SELECT * FROM orders")
    users_rows = curso.execute("SELECT * FROM users")
    result = curso.fetchall()
    return render_template('pages/all_users.html', result=result, row=num_rows, order_rows=order_rows,
                           users_rows=users_rows)

@app.route('/admin_add_product', methods=['POST', 'GET'])
@is_admin_logged_in
def admin_add_product():
    if request.method == 'POST':
        name = request.form.get('name')
        price = request.form['price']
        description = request.form['description']
        available = request.form['available']
        category = request.form['category']
        item = request.form['item']
        code = request.form['code']
        file = request.files['picture']
        if name and price and description and available and category and item and code and file:
            pic = file.filename
            photo = pic.replace("'", "")
            picture = photo.replace(" ", "_")

```

```

if picture.lower().endswith(('png', 'jpg', 'jpeg')):
    save_photo = photos.save(file, folder=category)
    if save_photo:
        # Create Cursor
        curs = db.connection.cursor()
        curs.execute("INSERT INTO
products(pName,price,description,available,category,item,pCode,picture)"
                "VALUES(%s, %s, %s, %s, %s, %s, %s, %s)",
                (name, price, description, available, category, item, code, picture))
        db.connection.commit()
        product_id = curs.lastrowid
        curs.execute("INSERT INTO product_level(product_id)" "VALUES(%s)", [product_id])
    if category == 'tshirt':
        level = request.form.getlist('tshirt')
        for lev in level:
            yes = 'yes'
            query = 'UPDATE product_level SET {field}=%s WHERE
product_id=%s'.format(field=lev)
            curs.execute(query, (yes, product_id))
            # Commit cursor
            db.connection.commit()
        elif category == 'wallet':
            level = request.form.getlist('wallet')
            for lev in level:
                yes = 'yes'
                query = 'UPDATE product_level SET {field}=%s WHERE
product_id=%s'.format(field=lev)
                curs.execute(query, (yes, product_id))
                # Commit cursor
                db.connection.commit()
            elif category == 'belt':
                level = request.form.getlist('belt')
                for lev in level:
                    yes = 'yes'
                    query = 'UPDATE product_level SET {field}=%s WHERE
product_id=%s'.format(field=lev)
                    curs.execute(query, (yes, product_id))
                    # Commit cursor
                    db.connection.commit()
                elif category == 'shoes':
                    level = request.form.getlist('shoes')
                    for lev in level:
                        yes = 'yes'
                        query = 'UPDATE product_level SET {field}=%s WHERE
product_id=%s'.format(field=lev)

```

```

        curs.execute(query, (yes, product_id))
        # Commit cursor
        db.connection.commit()

    else:
        flash('Product level not fund', 'danger')
        return redirect(url_for('admin_add_product'))
    # Close Connection
    curs.close()

    flash('Product added successful', 'success')
    return redirect(url_for('admin_add_product'))
else:
    flash('Picture not save', 'danger')
    return redirect(url_for('admin_add_product'))
else:
    flash('File not supported', 'danger')
    return redirect(url_for('admin_add_product'))
else:
    flash('Please fill up all form', 'danger')
    return redirect(url_for('admin_add_product'))
else:
    return render_template('pages/add_product.html')

@app.route('/edit_product', methods=['POST', 'GET'])
@is_admin_logged_in
def edit_product():
    if 'id' in request.args:
        product_id = request.args['id']
        curso = db.connection.cursor()
        res = curso.execute("SELECT * FROM products WHERE id=%s", (product_id,))
        product = curso.fetchall()
        curso.execute("SELECT * FROM product_level WHERE product_id=%s", (product_id,))
        product_level = curso.fetchall()
        if res:
            if request.method == 'POST':
                name = request.form.get('name')
                price = request.form['price']
                description = request.form['description']
                available = request.form['available']
                category = request.form['category']
                item = request.form['item']
                code = request.form['code']
                file = request.files['picture']
                # Create Cursor

```



```

if name and price and description and available and category and item and code and file:
    pic = file.filename
    photo = pic.replace("", "")
    picture = photo.replace(" ", "")
    if picture.lower().endswith(('.png', '.jpg', '.jpeg')):
        file.filename = picture
        save_photo = photos.save(file, folder=category)
    if save_photo:
        # Create Cursor
        cur = db.connection.cursor()
        exe = cur.execute(
            "UPDATE products SET pName=%s, price=%s, description=%s, available=%s,
category=%s, item=%s, pCode=%s, picture=%s WHERE id=%s",
            (name, price, description, available, category, item, code, picture, product_id))
        if exe:
            if category == 'tshirt':
                level = request.form.getlist('tshirt')
                for lev in level:
                    yes = 'yes'
                    query = 'UPDATE product_level SET {field}=%s WHERE product_id=%s'.format(
                        field=lev)
                    cur.execute(query, (yes, product_id))
                    # Commit cursor
                    db.connection.commit()
            elif category == 'wallet':
                level = request.form.getlist('wallet')
                for lev in level:
                    yes = 'yes'
                    query = 'UPDATE product_level SET {field}=%s WHERE product_id=%s'.format(
                        field=lev)
                    cur.execute(query, (yes, product_id))
                    # Commit cursor
                    db.connection.commit()
            elif category == 'belt':
                level = request.form.getlist('belt')
                for lev in level:
                    yes = 'yes'
                    query = 'UPDATE product_level SET {field}=%s WHERE product_id=%s'.format(
                        field=lev)
                    cur.execute(query, (yes, product_id))
                    # Commit cursor
                    db.connection.commit()
            elif category == 'shoes':
                level = request.form.getlist('shoes')
                for lev in level:

```

```

        yes = 'yes'
        query = 'UPDATE product_level SET {field}=%s WHERE product_id=%s'.format(
            field=lev)
        cur.execute(query, (yes, product_id))
        # Commit cursor
        db.connection.commit()
    else:
        flash('Product level not fund', 'danger')
        return redirect(url_for('admin_add_product'))
    flash('Product updated', 'success')
    return redirect(url_for('edit_product'))
else:
    flash('Data updated', 'success')
    return redirect(url_for('edit_product'))
else:
    flash('Pic not upload', 'danger')
    return render_template('pages/edit_product.html', product=product,
                           product_level=product_level)
else:
    flash('File not support', 'danger')
    return render_template('pages/edit_product.html', product=product,
                           product_level=product_level)
else:
    flash('Fill all field', 'danger')
    return render_template('pages/edit_product.html', product=product,
                           product_level=product_level)
else:
    return render_template('pages/edit_product.html', product=product, product_level=product_level)
else:
    return redirect(url_for('admin_login'))
else:
    return redirect(url_for('admin_login'))

@app.route('/search', methods=['POST', 'GET'])
def search():
    form = OrderForm(request.form)
    if 'q' in request.args:
        q = request.args['q']
        # Create cursor
        cur = db.connection.cursor()
        # Get message
        query_string = "SELECT * FROM products WHERE pName LIKE %s ORDER BY id ASC"
        cur.execute(query_string, ('%' + q + '%',))
        products = cur.fetchall()

```

```

# Close Connection
cur.close()
flash('Showing result for: ' + q, 'success')
return render_template('search.html', products=products, form=form)
else:
    flash('Search again', 'danger')
    return render_template('search.html')

@app.route('/profile')
@is_logged_in
def profile():
    if 'user' in request.args:
        q = request.args['user']
        curso = db.connection.cursor()
        curso.execute("SELECT * FROM users WHERE id=%s", (q,))
        result = curso.fetchone()
        if result:
            if result['id'] == session['uid']:
                curso.execute("SELECT * FROM orders WHERE uid=%s ORDER BY id ASC", (session['uid'],))
                res = curso.fetchall()
                return render_template('profile.html', result=res)
            else:
                flash('Unauthorised', 'danger')
                return redirect(url_for('login'))
        else:
            flash('Unauthorised! Please login', 'danger')
            return redirect(url_for('login'))
    else:
        flash('Unauthorised', 'danger')
        return redirect(url_for('login'))

class UpdateRegisterForm(Form):
    name = StringField('Full Name', [validators.length(min=3, max=50)],
                        render_kw={'autofocus': True, 'placeholder': 'Full Name'})
    email = EmailField('Email', [validators.DataRequired(), validators.Email(), validators.length(min=4,
max=25)],
                        render_kw={'placeholder': 'Email'})
    password = PasswordField('Password', [validators.length(min=3)],
                             render_kw={'placeholder': 'Password'})
    mobile = StringField('Mobile', [validators.length(min=11, max=15)], render_kw={'placeholder': 'Mobile'})

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

```

```

@is_logged_in
def settings():
    form = UpdateRegisterForm(request.form)
    if 'user' in request.args:
        q = request.args['user']
        curso = db.connection.cursor()
        curso.execute("SELECT * FROM users WHERE id=%s", (q,))
        result = curso.fetchone()
        if result:
            if result['id'] == session['uid']:
                if request.method == 'POST' and form.validate():
                    name = form.name.data
                    email = form.email.data
                    password = sha256_crypt.encrypt(str(form.password.data))
                    mobile = form.mobile.data

                    # Create Cursor
                    cur = db.connection.cursor()
                    exe = cur.execute("UPDATE users SET name=%s, email=%s, password=%s, mobile=%s
WHERE id=%s",
                                   (name, email, password, mobile, q))
                    if exe:
                        flash('Profile updated', 'success')
                        return render_template('user_settings.html', result=result, form=form)
                    else:
                        flash('Profile not updated', 'danger')
                        return render_template('user_settings.html', result=result, form=form)
                else:
                    flash('Unauthorised', 'danger')
                    return redirect(url_for('login'))
            else:
                flash('Unauthorised! Please login', 'danger')
                return redirect(url_for('login'))
        else:
            flash('Unauthorised', 'danger')
            return redirect(url_for('login'))

class DeveloperForm(Form): #
    id = StringField("", [validators.length(min=1)],
                     render_kw={'placeholder': 'Input a product id...'})

@app.route('/developer', methods=['POST', 'GET'])
def developer():

```

```

form = DeveloperForm(request.form)
if request.method == 'POST' and form.validate():
    q = form.id.data
    curso = db.connection.cursor()
    result = curso.execute("SELECT * FROM products WHERE id=%s", (q,))
    if result > 0:
        x = content_based_filtering(q)
        wrapped = wrappers(content_based_filtering, q)
        execution_time = timeit.timeit(wrapped, number=0)
        seconds = ((execution_time / 1000) % 60)
        return render_template('developer.html', form=form, x=x, execution_time=seconds)
    else:
        nothing = 'Nothing found'
        return render_template('developer.html', form=form, nothing=nothing)
else:
    return render_template('developer.html', form=form)

if __name__ == '__main__':
    app.run(debug=True)

```

home.html

```

{% extends 'layout.html' %}
{% block body %}
<!-- Page Content -->
<div class="container">
    <div class="row">
        <!-- /.col-lg-3 -->
        <div class="col-lg-12">
            <div id="carouselExampleIndicators" class="carousel slide my-4" data-ride="carousel">
                <ol class="carousel-indicators">
                    <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
                    <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
                    <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
                </ol>
                <div class="carousel-inner" role="listbox" style="max-height: 300px">
                    <div class="carousel-item active">
                        

```

```

</div>
<div class="carousel-item">
  
  <div class="carousel-caption">
    <h3>Los Angeles</h3>
    <p>We had such a great time in LA!</p>
  </div>
</div>
<div class="carousel-item">
  
  <div class="carousel-caption">
    <h3>Los Angeles</h3>
    <p>We had such a great time in LA!</p>
  </div>
</div>
</div>
<a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-
slide="prev">
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  <span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-
slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="sr-only">Next</span>
</a>
</div>
<h3><a href="/tshirt">T-Shirt</a></h3>
<div class="row">
  {% for product in tshirt %}
  {% if product.category == 'tshirt' %}
  <div class="col-lg-3 col-md-4 col-sm-6">
    <div class="card mb-4">
      <a href="{% {product.category}}?view={% {product.id}}">
        <div class="card-img-top id_{{product.id}} "
          style="background: url('static/image/product/{% {product.category}}/{% {product.picture}}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">
        </div>
      </a>
      <div class="card-body">
        <h5><a class="card-title" href="javascript:void();">{{ product.pName }}</a></h5>
        <h4>₹ {{product.price}}</h4>

```

```

        <span class="text-warning"></span>
    </div>
</div>
<!-- /.card -->
</div>
{% endif %}
{% endfor %}
</div>
<!-- /.row -->
<div class="border p-4 mb-4" style="height: 350px">
    
</div>

<h3><a href="/wallet">Wallet</a></h3>
<div class="row">
    {% for product in wallet %}
    <div class="col-lg-3 col-md-4 col-sm-6">
        <div class="card mb-4">
            <a href="/{{ product.category }}?view={{ product.id }}">
                <div class="card-img-top"
                    style="background: url('static/image/product/{{ product.category }}/{{ product.picture }}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">
                </div>
            </a>
            <div class="card-body">
                <h5><a class="card-title" id="" href="/{{ product.category }}?view={{ product.id }}">{{
                    product.pName }}</a></h5>
                <h4>₹ {{ product.price }}</h4>
                <span class="text-warning"></span>
            </div>
        </div>
    </div>
    <!-- /.card -->
</div>
{% endfor %}
</div>
<!-- /.row -->
<div class="border p-4 mb-4" style="height: 350px">
    
</div>
<h3><a href="/belt">Belt</a></h3>
<div class="row">
    {% for product in belt %}
    <div class="col-lg-3 col-md-4 col-sm-6">
        <div class="card mb-4">
            <a href="/{{ product.category }}?view={{ product.id }}">
                <div class="card-img-top"

```

```

        style="background: url('static/image/product/{{ product.category }}/{{ product.picture }}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">
    </div>
</a>
<div class="card-body">
    <h5><a class="card-title" href="/{{ product.category }}?view={{ product.id }}">{{
        product.pName }}</a></h5>
    <h4>₹ {{ product.price }}</h4>
    <span class="text-warning"></span>
</div>
</div>
<!-- /.card -->
</div>
{% endfor %}
</div>
<div class="border p-4 mb-4" style="height: 350px">
    
</div>
<h3><a href="/shoes">Shoes</a></h3>
<div class="row">
    {% for product in shoes %}
    <div class="col-lg-3 col-md-4 col-sm-6">
        <div class="card mb-4">
            <a href="/{{ product.category }}?view={{ product.id }}">
                <div class="card-img-top"
                    style="background: url('static/image/product/{{ product.category }}/{{ product.picture }}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">
                </div>
            </a>
            <div class="card-body">
                <h5><a class="card-title" href="/{{ product.category }}?view={{ product.id }}">{{
                    product.pName }}</a></h5>
                <h4>₹ {{ product.price }}</h4>
                <span class="text-warning"></span>
            </div>
        </div>
    </div>
    <!-- /.card -->
    </div>
    {% endfor %}
</div>
<div class="border p-4 mb-4" style="height: 350px">
    
</div>
</div>

```



```

        <!-- /.col-lg-9 -->
    </div>
    <!-- /.row -->
</div>
<!-- /.container -->
{% endblock %}

```

Admin_login.html

```

{% extends 'layout.html' %}

{% block body %}
<div class="jumbotron">
    <h2 style="text-align: center; color: #206aaa;">Login Form</h2>
    {% from "includes/_formhelpers.html" import render_field %}
    <div>
        <form method="POST" action="">
            <div class="form-group">
                {{ render_field(form.username, class_="form-control") }}
            </div>
            <div class="form-group">
                <label><strong>Password</strong></label>
                <input type="password" name="password" class="form-control" value="{{ request.form.password }}">
            </div>
            <button type="submit" class="btn btn-primary">Login</button>
        </form>
    </div>
</div>
{% endblock %}

```

Belt.html

```

{% extends 'layout.html' %}
{% block body %}
<!-- Page Content -->
<div class="container">
    <div class="row">
        <div class="col-lg-3">
            <h1 class="my-4">Belt</h1>
            <div class="list-group">
                <a href="/tshirt" class="list-group-item">T-Shirt</a>
                <a href="/wallet" class="list-group-item">Wallet</a>
            </div>
        </div>
    </div>
</div>

```

```

    <a href="/belt" class="list-group-item active">Belt</a>
    <a href="/shoes" class="list-group-item">Shoes</a>
  </div>
</div>
<!-- /.col-lg-3 -->
<div class="col-lg-9">
  { % include 'includes/_flashmsg.html' % }
  <div id="carouselExampleIndicators" class="carousel slide my-4" data-ride="carousel">
    <ol class="carousel-indicators">
      <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
      <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
      <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
    </ol>
    <div class="carousel-inner" role="listbox" style="max-height: 300px">
      <div class="carousel-item active">
        
        <div class="carousel-caption">
          <h3>Los Angeles</h3>
          <p>We had such a great time in LA!</p>
        </div>
      </div>
      <div class="carousel-item">
        
        <div class="carousel-caption">
          <h3>Los Angeles</h3>
          <p>We had such a great time in LA!</p>
        </div>
      </div>
      <div class="carousel-item">
        
        <div class="carousel-caption">
          <h3>Los Angeles</h3>
          <p>We had such a great time in LA!</p>
        </div>
      </div>
    </div>
    <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-
slide="prev">
      <span class="carousel-control-prev-icon" aria-hidden="true"></span>
      <span class="sr-only">Previous</span>
    </a>

```

```

        <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-
slide="next">
            <span class="carousel-control-next-icon" aria-hidden="true"></span>
            <span class="sr-only">Next</span>
        </a>
    </div>

    <div class="row">
        {% for product in belt %}
        <div class="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-4">
            <div class="card h-100">
                <a href="/belt?view={{ product.id }}">
                    <div class="card-img-top"
                        style="background: url('static/image/product/{{ product.category }}/{{ product.picture }}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% "></div>
                </a>
                <div class="card-body">
                    <h5 class="card-title">
                        <a href="/belt?view={{ product.id }}">{{ product.pName }}</a>
                    </h5>
                    <h4>₹{{ product.price }}</h4>
                </div>
                <div class="card-footer">
                    <small class="text-muted"></small>
                </div>
            </div>
        </div>
        {% endfor %}
    </div>

    <!-- /.row -->

</div>
<!-- /.col-lg-9 -->

</div>
<!-- /.row -->

</div>
<!-- /.container -->
{% endblock %}

```

Layout.html

```
<html>
```

```

<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Sundari Shopping</title>
  <meta charset="utf-8">
  <link rel="icon" href="{ { url_for('static', filename='image/logo.png') } }" type="image/x-icon">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
  <style>
    body {
      background: #60a7f8;
    }

    a {
      color: rgb(65, 56, 56);
      text-decoration: none;
    }

    li {
      color: white;
    }
  </style>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
  <link rel="stylesheet" href="{ { url_for('static', filename='css/style.css') } }">
</head>

<body style=" margin-top: 57px ">
  {% include 'includes/_navbar.html' %}
  <div class="container">
    {% block body %} {% endblock %}
  </div>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/js/bootstrap.min.js"></script>
</body>
{% include 'includes/_footer.html' %}

</html>

```

Login.html

```

{% extends 'layout.html' %}

{% block body %}

```

```
{% include 'includes/_flashmsg.html' %}
<div class="jumbotron">
  <h2 style="text-align: center; color: #206aaa;">Login Form</h2>
  {% from "includes/_formhelpers.html" import render_field %}
  <div>
    <form method="POST" action="">
      <div class="form-group">
        {{ render_field(form.username, class_="form-control") }}
      </div>
      <div class="form-group">
        {{ render_field(form.password, class_="form-control") }}
      </div>
      <button type="submit" class="btn btn-primary">Login</button>
    </form>
  </div>
</div>
{% endblock %}
```

Orderproduct.html

```
{% extends 'layout.html' %}
{% block body %}
<!-- Page Content -->
<div class="container">
  <div class="row">
    <div class="col-lg-3 my-4">
      <div class="list-group">
        <a href="/tshirt" class="list-group-item">T-Shirt</a>
        <a href="/wallet" class="list-group-item">Wallet</a>
        <a href="/belt" class="list-group-item">Belt</a>
        <a href="/shoes" class="list-group-item">Shoes</a>
      </div>
    </div>
    <!-- /.col-lg-3 -->
    <div class="col-lg-9">
      <div class="row">
        {% for product in tshirts %}
        <div class="col-lg-6">
          <div class="card mt-4">
            
          </div>
        </div>
        <!-- /.card -->
      </div>
    </div>
  </div>
</div>
```

```

<div class="col-lg-6">
  <div class="card card-outline-secondary my-4">
    <div class="card-header">
      Order Details
    </div>
    <div class="card-body">
      { % from "includes/_formhelpers.html" import render_field % }
      <form method="POST" action="/{{ product.category }}?order={{ product.id }}">
        <div class="form-group">
          { {render_field(form.name, class_="form-control")} }
        </div>
        <div class="form-group">
          { {render_field(form.mobile_num, class_="form-control")} }
        </div>

        <div class="form-group">
          { {render_field(form.quantity, class_="form-control")} }
        </div>

        <div class="form-group">
          { {render_field(form.order_place, class_="form-control")} }
        </div>
        <button type="submit" class="btn btn-primary">Confirm Order</button>
      </form>
    </div>
  </div>
  { % endfor % }
<!-- /.card -->

</div>

{ % endif % }
-->
</div>
<!-- /.col-lg-9 -->

</div>
<!-- /.row -->

</div>
<!-- /.container -->
{ % endblock % }

```

Profile.html

```
{% extends 'layout.html' %}
{% block body %}
<!-- Page Content -->
<div class="container">
  <div class="row">
    <div class="col-lg-3">
      <h1 class="my-4">{{ session.s_name }}</h1>
      <div class="list-group">
        <a href="/profile?user={{ session.uid }}" class="list-group-item">Order List</a>
        <a href="/settings?user={{ session.uid }}" class="list-group-item">Settings</a>
      </div>
    </div>
    <!-- /.col-lg-3 -->
    <div class="col-lg-9">
      <div class="card card-default my-4">
        <div class="card-header">
          <i class="fa fa-bar-chart-o fa-fw"></i>My order list
        </div>
        <!-- /.panel-heading -->
        <div class="card-body">
          <div class="row">
            <div class="col-lg-12">
              {% if result %}
              <div class="table-responsive">
                <table class="table table-bordered table-hover table-striped">
                  <thead>
                    <tr>
                      <th>Order No</th>
                      <th>Product ID</th>
                      <th>Quantity</th>
                      <th>Order Place</th>
                      <th>Mobile</th>
                      <th>Order Date</th>
                      <th>Delivery Date</th>
                    </tr>
                  </thead>
                  <tbody>
                    {% for order in result %}
                    <tr>
                      <td>{{ order.id }}</td>
                      <td>{{ order.pid }}</td>
```

```

        <td>{{ order.quantity }}</td>
        <td>{{ order.oplace }}</td>
        <td>{{ order.mobile }}</td>
        <td>{{ order.odate }}</td>
        <td>{{ order.ddate }}</td>
    </tr>
    {% endfor %}
</tbody>
</table>
</div>
{% else %}
<h3>No orders found</h3>
{% endif %}
<!-- /.table-responsive -->
</div>
</div>
<!-- /.row -->
</div>
<!-- /.panel-body -->
</div>
</div>
<!-- /.col-lg-9 -->

</div>
<!-- /.row -->

</div>
<!-- /.container -->
{% endblock %}

```

Register.html

```

{% extends 'layout.html' %}

{% block body %}
<div class="jumbotron">
    <h2 style="text-align: center; color: #206aaa;">Sign Up Form</h2>
    {% from "includes/_formhelpers.html" import render_field %}
    <form method="POST" action="">
        <div class="form-group">
            {{ render_field(form.name, class_="form-control") }}
        </div>
        <div class="form-group">
            {{ render_field(form.email, class="form-control") }}

```



```

</div>
<div class="form-group">
  {{render_field(form.username, class="form-control")}}
</div>
<div class="form-group">
  {{render_field(form.password, class="form-control")}}
</div>
<div class="form-group">
  {{render_field(form.mobile, class="form-control")}}
</div>
<p>
  <input type="submit" class="btn btn-primary" value="Sign Up">
</p>
</form>
</div>
{% endblock %}

```

Search.html

```

{% extends 'layout.html' %}
{% block body %}
<!-- Page Content -->
<div class="container">
  <div class="row">
    <!-- /.col-lg-3 -->
    <div class="col-lg-12">
      {% include 'includes/_flashmsg.html' %}
      <div class="row">
        {% for product in products %}
        <div class="col-lg-3 col-md-4 col-sm-6">
          <div class="card mb-4">
            <a href="/{{product.category}}?view={{product.id}}">
              <div class="card-img-top"
                style="background: url('static/image/product/{{product.category}}/{{product.picture}}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">
              </div>
            </a>
            <div class="card-body">
              <h3><a class="card-title" id="" href="/{{product.category}}?view={{product.id}}">{{
                product.pName }}</a></h3>
              <h4>₹24.99</h4>
              <span class="text-warning"></span>
            </div>
          </div>
        </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <!-- /.card -->
    </div>
    {% endfor %}
</div>
</div>
<!-- /.col-lg-9 -->
</div>
<!-- /.row -->
</div>
<!-- /.container -->
{% endblock %}

```

Shoes.html

```

{% extends 'layout.html' %}
{% block body %}
<!-- Page Content -->
<div class="container">
    <div class="row">
        <div class="col-lg-3">
            <h1 class="my-4">Shoes</h1>
            <div class="list-group">
                <a href="/tshirt" class="list-group-item">T-Shirt</a>
                <a href="/wallet" class="list-group-item">Wallet</a>
                <a href="/belt" class="list-group-item">Belt</a>
                <a href="/shoes" class="list-group-item active">Shoes</a>
            </div>
        </div>
        <!-- /.col-lg-3 -->
        <div class="col-lg-9">
            {% include '_flashmsg.html' %}
            <div id="carouselExampleIndicators" class="carousel slide my-4" data-ride="carousel">
                <ol class="carousel-indicators">
                    <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
                    <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
                    <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
                </ol>
                <div class="carousel-inner" role="listbox" style="max-height: 300px">
                    <div class="carousel-item active">
                        
                        <div class="carousel-caption">
                            <h3>Los Angeles</h3>
                            <p>We had such a great time in LA!</p>

```

```

        </div>
    </div>
    <div class="carousel-item">
        
        <div class="carousel-caption">
            <h3>Los Angeles</h3>
            <p>We had such a great time in LA!</p>
        </div>
    </div>
    <div class="carousel-item">
        
        <div class="carousel-caption">
            <h3>Los Angeles</h3>
            <p>We had such a great time in LA!</p>
        </div>
    </div>
</div>
<div>
    <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-
slide="prev">
        <span class="carousel-control-prev-icon" aria-hidden="true"></span>
        <span class="sr-only">Previous</span>
    </a>
    <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-
slide="next">
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
        <span class="sr-only">Next</span>
    </a>
</div>

<div class="row">
    {% for product in shoes %}
    <div class="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-4">
        <div class="card h-100">
            <a href="/shoes?view={{ product.id }}">
                <div class="card-img-top"
                    style="background: url('static/image/product/{{ product.category }}/{{ product.picture }}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">
                </div>
            </a>
            <div class="card-body">
                <h5 class="card-title">
                    <a href="/shoes?view={{ product.id }}">{{ product.pName }}</a>
                </h5>
            </div>
        </div>
    </div>
    </div>

```

```

        <h4>₹ { {product.price} } </h4>
    </div>
    <div class="card-footer">
        <small class="text-muted"></small>
    </div>
</div>
</div>
    {% endfor %}
</div>
<!-- /.row -->

</div>
<!-- /.col-lg-9 -->

</div>
<!-- /.row -->

</div>
<!-- /.container -->
{% endblock %}

```

T-shirt.html

```

{% extends 'layout.html' %}
{% block body %}
<!-- Page Content -->
<div class="container">
    <div class="row">
        <div class="col-lg-3">
            <h1 class="my-4">T-Shirt</h1>
            <div class="list-group">
                <a href="/tshirt" class="list-group-item active">T-Shirt</a>
                <a href="/wallet" class="list-group-item">Wallet</a>
                <a href="/belt" class="list-group-item">Belt</a>
                <a href="/shoes" class="list-group-item">Shoes</a>
            </div>
        </div>
        <div class="col-lg-9">
            {% include 'includes/_flashmsg.html' %}
            <div id="carouselExampleIndicators" class="carousel slide my-4" data-ride="carousel">
                <ol class="carousel-indicators">
                    <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
                    <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
                </ol>
            </div>
        </div>
    </div>
</div>

```

```

    <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
</ol>
<div class="carousel-inner" role="listbox" style="max-height: 300px">
  <div class="carousel-item active">
    
    <div class="carousel-caption">
      <h3>Los Angeles</h3>
      <p>We had such a great time in LA!</p>
    </div>
  </div>
  <div class="carousel-item">
    
    <div class="carousel-caption">
      <h3>Los Angeles</h3>
      <p>We had such a great time in LA!</p>
    </div>
  </div>
  <div class="carousel-item">
    
    <div class="carousel-caption">
      <h3>Los Angeles</h3>
      <p>We had such a great time in LA!</p>
    </div>
  </div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-
slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-
slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>

<div class="row">
  { % for product in tshirt % }
  <div class="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-4">
    <div class="card h-100">
      <a href="/tshirt?view={{ product.id }}">

```

```

        <div class="card-img-top"
            style="background: url('static/image/product/{ {product.category} }/{ {product.picture} }');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">
        </div>
    </a>
    <div class="card-body">
        <h5 class="card-title">
            <a href="/tshirt?view={ {product.id} }">{ {product.pName} }</a>
        </h5>
        <h4>₹ { {product.price} }</h4>
    </div>
    <div class="card-footer">
        <small class="text-muted"></small>
    </div>
    </div>
    </div>
    { % endfor % }
</div>
<!-- /.row -->

</div>
<!-- /.col-lg-9 -->

</div>
<!-- /.row -->

</div>
<!-- /.container -->
{ % endblock % }

```

User_settings.html

```

{ % extends 'layout.html' % }
{ % block body % }
<!-- Page Content -->
<div class="container">
    <div class="row">
        <div class="col-lg-3">
            <h1 class="my-4">{ {session.s_name} }</h1>
            <div class="list-group">
                <a href="/profile?user={ {session.uid} }" class="list-group-item">Order List</a>
                <a href="/settings?user={ {session.uid} }" class="list-group-item">Settings</a>
            </div>
        </div>
    </div>
</div>

```

```

<!-- /.col-lg-3 -->
<div class="col-lg-9">
  {% include 'includes/_flashmsg.html' %}
  <div class="card card-default my-4">
    <div class="card-header">
      <i class="fa fa-bar-chart-o fa-fw"></i>Update profile info
    </div>
    <!-- /.panel-heading -->
    <div class="card-body">
      {% if result %}
      {% from "includes/_formhelpers.html" import render_field %}
      <form method="POST" action="/settings?user={ {result.id} }">
        <div class="form-group">
          {{ render_field(form.name, class_="form-control", value=result.name)}}
        </div>
        <div class="form-group">
          {{ render_field(form.email, class="form-control", value=result.email)}}
        </div>
        <div class="form-group">
          {{ render_field(form.password, class="form-control") }}
        </div>
        <div class="form-group">
          {{ render_field(form.mobile, class="form-control", value=result.mobile)}}
        </div>
        <p>
          <input type="submit" class="btn btn-primary" value="Update Settings">
        </p>
      </form>
      {% endif %}
    </div>
    <!-- /.panel-body -->
  </div>
</div>
<!-- /.col-lg-9 -->

</div>
<!-- /.row -->

</div>
<!-- /.container -->
{% endblock %}

```

View_product.html


```

<h2 class="mb-4">Similar product for you:</h2>
<div class="row">
  {% for product in x[0] %}
  <div class="col-lg-3 col-md-4 col-sm-6">
    <div class="card mb-4">
      <a href="/{{product.category}}?view={{product.id}}">
        <div class="card-img-top id_{{product.id}}
          style="background: url('static/image/product/{{product.category}}/{{product.picture}}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">
        </div>
      </a>
      <div class="card-body">
        <h5><a class="card-title" href="/{{product.category}}?view={{product.id}}">{{
          product.pName}}</a></h5>
        <h4>₹ {{product.price}}</h4>
      </div>
    </div>
    <!-- /.card -->
  </div>
  {% endfor %}
</div>
{% endif %}
</div>
<!-- /.col-lg-9 -->

</div>
<!-- /.row -->

</div>
<!-- /.container -->
{% endblock %}

```

Wallet.html

```

{% extends 'layout.html' %}
{% block body %}
<!-- Page Content -->
<div class="container">
  <div class="row">
    <div class="col-lg-3">
      <h1 class="my-4">Wallet</h1>
      <div class="list-group">
        <a href="/tshirt" class="list-group-item">T-Shirt</a>

```

```

    <a href="/wallet" class="list-group-item active">Wallet</a>
    <a href="/belt" class="list-group-item">Belt</a>
    <a href="/shoes" class="list-group-item">Shoes</a>
  </div>
</div>
<!-- /.col-lg-3 -->
<div class="col-lg-9">
  { % include 'includes/_flashmsg.html' % }
  <div id="carouselExampleIndicators" class="carousel slide my-4" data-ride="carousel">
    <ol class="carousel-indicators">
      <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
      <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
      <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
    </ol>
    <div class="carousel-inner" role="listbox" style="max-height: 300px">
      <div class="carousel-item active">
        
        <div class="carousel-caption">
          <h3>Los Angeles</h3>
          <p>We had such a great time in LA!</p>
        </div>
      </div>
      <div class="carousel-item">
        
        <div class="carousel-caption">
          <h3>Los Angeles</h3>
          <p>We had such a great time in LA!</p>
        </div>
      </div>
      <div class="carousel-item">
        
        <div class="carousel-caption">
          <h3>Los Angeles</h3>
          <p>We had such a great time in LA!</p>
        </div>
      </div>
    </div>
    <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-
slide="prev">
      <span class="carousel-control-prev-icon" aria-hidden="true"></span>
      <span class="sr-only">Previous</span>
    </a>

```

```

        <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-
slide="next">
            <span class="carousel-control-next-icon" aria-hidden="true"></span>
            <span class="sr-only">Next</span>
        </a>
    </div>

    <div class="row">
        {% for product in wallet %}
        <div class="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-4">
            <div class="card h-100">
                <a href="/wallet?view={{ product.id }}">
                    <div class="card-img-top"
                        style="background: url('static/image/product/{{ product.category }}/{{ product.picture }}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">
                    </div>
                </a>
                <div class="card-body">
                    <h5 class="card-title">
                        <a href="/wallet?view={{ product.id }}">{{ product.pName }}</a>
                    </h5>
                    <h4>₹ {{ product.price }}</h4>
                </div>
                <div class="card-footer">
                    <small class="text-muted"></small>
                </div>
            </div>
        </div>
        {% endfor %}
    </div>

    <!-- /.row -->

</div>
<!-- /.col-lg-9 -->

</div>
<!-- /.row -->

</div>
<!-- /.container -->
{% endblock %}

```

_flashmsg.html

```
{% with messages = get_flashed_messages(with_categories=true) %}
<!-- Categories: success (green), info (blue), warning (yellow), danger (red) -->
{% if messages %}
  {% for category, message in messages %}
    <div class="alert alert-{{ category }}" alert-dismissible my-4" role="alert">
      {{ message }}
    </div>
  {% endfor %}
{% endif %}
{% endwith %}
```

_footer.html

```
<!-- Footer -->
<section id="footer">
  <div class="container">
    <div class="row text-center text-xs-center text-sm-left text-md-left">
      <div class="col-xs-12 col-sm-4 col-md-4">
        <h5>Quick Links</h5>
        <ul class="list-unstyled quick-links">
          <li><a href="/"><i class="fa fa-angle-double-right"></i>Home</a></li>
          <li><a href="/tshirt"><i class="fa fa-angle-double-right"></i>T-Shirt</a></li>
          <li><a href="/belt"><i class="fa fa-angle-double-right"></i>Belt</a></li>
          <li><a href="/shoes"><i class="fa fa-angle-double-right"></i>Shoes</a></li>
          <li><a href="/wallet"><i class="fa fa-angle-double-right"></i>Wallet</a></li>
        </ul>
      </div>
      <div class="col-xs-12 col-sm-4 col-md-4">
        <h5>Social</h5>
        <ul class="list-unstyled quick-links">
          <li><a href="https://www.linkedin.com/in/vkparamesh3232">Linked In</a></li>
          <li><a href="https://github.com/Vkparamesh">Git Hub</a>
          </li>
        </ul>
      </div>
      <div class="col-xs-12 col-sm-4 col-md-4">
        <h5>About Team</h5>
        <ul class="list-unstyled quick-links">
          <li>Team ID : PNT2022TMID38912</li>
          <li>Team Leader : PARAMESHWARAN V K</li>
          <li>Team member : MANOJ ADHITHYAN A</li>
          <li>Team member : MOHANRAJ M</li>
          <li>Team member : DAYANITHI S</li>
        </ul>
      </div>
    </div>
  </div>
```

```

    </div>
  </div>
  <div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12 mt-2 mt-sm-2 text-center text-white">
      <p class="h6"> krishnasamy college of engineering and technology &copy All
        right Reversed.</p>
    </div>
  </div>
</div>
</section>
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "0319627f-d899-46cf-900d-e7d547432eff", // The ID of this integration.
    region: "eu-gb", // The region your integration is hosted in.
    serviceInstanceID: "2187bdc6-3463-4336-a0e5-105ac4d979db", // 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>
<!-- ./Footer -->

```

_formhelper.html

```

{% macro render_field(field) %}
<dt>{{ field.label }}</dt>
{{ field(**kwargs)|safe }}
{% if field.errors %}
  {% for error in field.errors %}
    <span class="help-inline">{{ error }}</span>
  {% endfor %}
{% endif %}
{% endmacro %}

```

_navbar.html

```

<nav class="navbar navbar-expand-lg navbar-light bg-white border-bottom shadow-sm fixed-top">
  

```

```

<a class="navbar-brand" href="/">Sundari Shopping</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive"
    aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarResponsive">
    <div class="input-group p-0 justify-content-center">
        <form action="/search" method="get" class="w-100 m-0" style="padding-right: 20%;">
            <div class="input-group-append rounded-left">
                <input type="text" class="form-control rounded-left form-control-sm" autocomplete="off"
                    style="float: left; border-radius: 0;" name="q" placeholder="What you want...">
                <button class="btn btn-success btn-sm rounded-right" style="border-radius: 0;"
                    type="submit">Search</button>
            </div>
        </form>
    </div>
    <ul class="navbar-nav ml-auto">
        { % if session.logged_in % }
        <li class="nav-item"><a href="/" class="nav-link"><span class=""></span> Home</a></li>
        <li class="nav-item"><a href="/profile?user={ { session.uid } }" class="nav-
link">{ { session.s_name } }</a>
        </li>
        <li class="nav-item"><a href="/out" class="nav-link"><span class=""></span> Logout</a></li>
        { % else % }
        <li class="nav-item"><a href="/" class="nav-link"><span class=""></span> Home</a></li>
        <li class="nav-item"><a href="/register" class="nav-link"><span class=""></span> Signup</a></li>
        <li class="nav-item"><a href="/login" class="nav-link"><span class=""></span> Login</a></li>
        { % endif % }
    </ul>
</div>
</nav>

```

Add_product.html

```

{ % extends 'pages/layout.html' % }
{ % block body % }
<div id="page-wrapper">
    <div class="row">
        <div class="col-lg-12">
            <h1 class="page-header">Add Product</h1>
            { % include 'includes/_flashmsg.html' % }
        </div>
    </div>

```

```

<!-- /.col-lg-12 -->
</div>
<!-- /.row -->
<div class="row">
  <div class="col-lg-12">
    <div class="panel panel-default">
      <div class="panel-body">
        <div class="row">
          <div class="col-lg-6">
            <form role="form" enctype=multipart/form-data class="" action="/admin_add_product"
              method="post">
              <div class="form-group">
                <label>Product Name</label>
                <input class="form-control" required="required" name="name" placeholder="Enter
name">
              </div>
              <div class="form-group">
                <label>Price</label>
                <input class="form-control" required="required" name="price" placeholder="Enter
price">
              </div>
              <div class="form-group">
                <label>Description</label>
                <textarea class="form-control" required="required" name="description" rows="3"
                  placeholder="Write product description"></textarea>
              </div>
              <div class="form-group">
                <label>Available</label>
                <input class="form-control" required="required" name="available"
placeholder="Available">
              </div>
              <div class="form-group">
                <label>Category</label>
                <select class="form-control" id="category_id" onchange="populateList();"
                  name="category">
                  <option value="tshirt">T-Shirt</option>
                  <option value="wallet">Wallet</option>
                  <option value="belt">Belt</option>
                  <option value="shoes">Shoes</option>
                </select>
              </div>
              <div class="form-group level-container">
                <label>level for T-shirt</label>
                <div class="form-group">
                  <label class="checkbox-inline">

```

```

        <input type="checkbox" name="tshirt" value="v_shape">V-Shape
    </label>
    <label class="checkbox-inline">
        <input type="checkbox" name="tshirt" value="polo">Polo T-shirt
    </label>
    <label class="checkbox-inline">
        <input type="checkbox" name="tshirt" value="clean_text">Clean Text
    </label>
    <label class="checkbox-inline">
        <input type="checkbox" name="tshirt" value="design">Colorful Design
    </label>
</div>

<label>Level for Wallet</label>
<div class="form-group">
    <label class="checkbox-inline">
        <input type="checkbox" name="wallet" value="chain">Chain
    </label>
    <label class="checkbox-inline">
        <input type="checkbox" name="wallet" value="leather">Leather
    </label>
    <label class="checkbox-inline">
        <input type="checkbox" name="wallet" value="design">Design
    </label>
</div>

<label>Level for Belt</label>
<div class="form-group">
    <label class="checkbox-inline">
        <input type="checkbox" name="belt" value="leather">Leather
    </label>
    <label class="checkbox-inline">
        <input type="checkbox" name="belt" value="hook">Hook
    </label>
    <label class="checkbox-inline">
        <input type="checkbox" name="belt" value="color">Color
    </label>
    <label class="checkbox-inline">
        <input type="checkbox" name="belt" value="design">Design
    </label>
</div>

<label>Level for Shoes</label>
<div class="form-group">
    <label class="checkbox-inline">
        <input type="checkbox" name="shoes" value="formal">Formal
    </label>
    <label class="checkbox-inline">

```



```

        <input type="checkbox" name="shoes" value="converse">Converse
      </label>
      <label class="checkbox-inline">
        <input type="checkbox" name="shoes" value="loafer">Loafer
      </label>
      <label class="checkbox-inline">
        <input type="checkbox" name="shoes" value="leather">Leather
      </label>
    </div>
  </div>
  <div class="form-group">
    <label>Item</label>
    <input class="form-control" required="required" name="item" placeholder="Ex:
watch/shoes">
  </div>
  <div class="form-group">
    <label>Product Code</label>
    <input class="form-control" required="required" name="code" placeholder="Enter
code">
  </div>
  <div class="form-group">
    <label>Add Image</label>
    <input type="file" required="required" name="picture">
  </div>
  <button type="submit" class="btn btn-default">Submit Button</button>
</form>
</div>
<!-- /.col-lg-6 (nested) -->
</div>
<!-- /.row (nested) -->
</div>
<!-- /.panel-body -->
</div>
<!-- /.panel -->
</div>
<!-- /.col-lg-12 -->
</div>
<!-- /.row -->
</div>
<!-- /#page-wrapper -->
{% endblock %}

```

All_orders.html

```

{% extends 'pages/layout.html' %}
{% block body %}
<div id="page-wrapper">
  <div class="row">
    <div class="col-lg-12">
      <h1 class="page-header">Dashboard</h1>
      {% include 'includes/_flashmsg.html' %}
    </div>
    <!-- /.col-lg-12 -->
  </div>
  <!-- /.row -->
  <div class="row">
    <div class="col-lg-4 col-md-6">
      <div class="panel panel-primary">
        <div class="panel-heading">
          <div class="row">
            <div class="col-xs-3">
              <i class="fa fa-list-ul fa-5x"></i>
            </div>
            <div class="col-xs-9 text-right">
              <div class="huge">{ { row } }</div>
              <div>All Product</div>
            </div>
          </div>
        </div>
        <div>
          <a href="/admin">
            <div class="panel-footer">
              <span class="pull-left">View Details</span>
              <span class="pull-right"><i class="fa fa-arrow-circle-right"></i></span>
              <div class="clearfix"></div>
            </div>
          </a>
        </div>
      </div>
    <div class="col-lg-4 col-md-6">
      <div class="panel panel-green">
        <div class="panel-heading">
          <div class="row">
            <div class="col-xs-3">
              <i class="fa fa-tasks fa-5x"></i>
            </div>
            <div class="col-xs-9 text-right">
              <div class="huge">{ { users_rows } }</div>
              <div>All Users</div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

    </div>
  </div>
  <a href="/users">
    <div class="panel-footer">
      <span class="pull-left">View Details</span>
      <span class="pull-right"><i class="fa fa-arrow-circle-right"></i></span>
      <div class="clearfix"></div>
    </div>
  </a>
</div>
</div>
<div class="col-lg-4 col-md-6">
  <div class="panel panel-yellow">
    <div class="panel-heading">
      <div class="row">
        <div class="col-xs-3">
          <i class="fa fa-shopping-cart fa-5x"></i>
        </div>
        <div class="col-xs-9 text-right">
          <div class="huge">{ { order_rows } }</div>
          <div>All Orders</div>
        </div>
      </div>
    </div>
  </div>
  <a href="/orders">
    <div class="panel-footer">
      <span class="pull-left">View Details</span>
      <span class="pull-right"><i class="fa fa-arrow-circle-right"></i></span>
      <div class="clearfix"></div>
    </div>
  </a>
</div>
</div>
</div>
<!-- /.row -->
<div class="row">
  <div class="col-lg-12">
    <div class="panel panel-default my-4">
      <div class="panel-heading">
        <i class="fa fa-bar-chart-o fa-fw"></i>All Orders
      </div>
      <!-- /.panel-heading -->
      <div class="panel-body">
        <div class="row">
          <div class="col-lg-12">

```

```

    { % if result % }
    <div class="table-responsive">
      <table class="table table-bordered table-hover table-striped">
        <thead>
          <tr>
            <th>Id</th>
            <th>User Id</th>
            <th>Name</th>
            <th>Product Id</th>
            <th>Quantity</th>
            <th>Order Place</th>
            <th>Mobile</th>
            <th>Order Date</th>
            <th>Delivery Date</th>
          </tr>
        </thead>
        <tbody>
          { % for product in result % }
          <tr>
            <td>{{ product.id }}</td>
            <td>{{ product.uid }}</td>
            <td>{{ product.ofname }}</td>
            <td>{{ product.pname }}</td>
            <td>{{ product.quantity }}</td>
            <td>{{ product.oplace }}</td>
            <td>{{ product.mobile }}</td>
            <td>{{ product.odate }}</td>
            <td>{{ product.ddate }}</td>
          </tr>
          { % endfor % }
        </tbody>
      </table>
    </div>
    { % else % }
    <h3>No orders found</h3>
    { % endif % }
    <!-- /.table-responsive -->
  </div>
</div>
<!-- /.row -->
</div>
<!-- /.panel-body -->
</div>
</div>
</div>

```

```

    <!-- /.row -->
</div>
{% endblock %}

```

All_users.html

```

{% extends 'pages/layout.html' %}
{% block body %}
<div id="page-wrapper">
  <div class="row">
    <div class="col-lg-12">
      <h1 class="page-header">Dashboard</h1>
      {% include 'includes/_flashmsg.html' %}
    </div>
    <!-- /.col-lg-12 -->
  </div>
  <!-- /.row -->
  <div class="row">
    <div class="col-lg-4 col-md-6">
      <div class="panel panel-primary">
        <div class="panel-heading">
          <div class="row">
            <div class="col-xs-3">
              <i class="fa fa-list-ul fa-5x"></i>
            </div>
            <div class="col-xs-9 text-right">
              <div class="huge">{{ row }}</div>
              <div>All Product</div>
            </div>
          </div>
        </div>
        <div>
          <a href="/admin">
            <div class="panel-footer">
              <span class="pull-left">View Details</span>
              <span class="pull-right"><i class="fa fa-arrow-circle-right"></i></span>
              <div class="clearfix"></div>
            </div>
          </a>
        </div>
      </div>
    </div>
    <div class="col-lg-4 col-md-6">
      <div class="panel panel-green">
        <div class="panel-heading">
          <div class="row">

```

```

        <div class="col-xs-3">
            <i class="fa fa-tasks fa-5x"></i>
        </div>
        <div class="col-xs-9 text-right">
            <div class="huge">{{ users_rows }}</div>
            <div>All Users</div>
        </div>
    </div>
</div>
<a href="/users">
    <div class="panel-footer">
        <span class="pull-left">View Details</span>
        <span class="pull-right"><i class="fa fa-arrow-circle-right"></i></span>
        <div class="clearfix"></div>
    </div>
</a>
</div>
</div>
<div class="col-lg-4 col-md-6">
    <div class="panel panel-yellow">
        <div class="panel-heading">
            <div class="row">
                <div class="col-xs-3">
                    <i class="fa fa-shopping-cart fa-5x"></i>
                </div>
                <div class="col-xs-9 text-right">
                    <div class="huge">{{ order_rows }}</div>
                    <div>All Orders</div>
                </div>
            </div>
        </div>
        <a href="/orders">
            <div class="panel-footer">
                <span class="pull-left">View Details</span>
                <span class="pull-right"><i class="fa fa-arrow-circle-right"></i></span>
                <div class="clearfix"></div>
            </div>
        </a>
    </div>
</div>
</div>
<!-- /.row -->
<div class="row">
    <div class="col-lg-12">
        <div class="panel panel-default my-4">

```

```

<div class="panel-heading">
  <i class="fa fa-bar-chart-o fa-fw"></i>All Orders
</div>
<!-- /.panel-heading -->
<div class="panel-body">
  <div class="row">
    <div class="col-lg-12">
      { % if result % }
      <div class="table-responsive">
        <table class="table table-bordered table-hover table-striped">
          <thead>
            <tr>
              <th>Id</th>
              <th>Name</th>
              <th>Email</th>
              <th>Username</th>
              <th>Mobile</th>
              <th>Registration Time</th>
              <th>Online</th>
            </tr>
          </thead>
          <tbody>
            { % for product in result % }
            <tr>
              <td>{ {product.id}} </td>
              <td>{ {product.name}} </td>
              <td>{ {product.email}} </td>
              <td>{ {product.username}} </td>
              <td>{ {product.mobile}} </td>
              <td>{ {product.reg_time}} </td>
              <td>{ {product.online}} </td>
            </tr>
            { % endfor % }
          </tbody>
        </table>
      </div>
      { % else % }
      <h3>No user found</h3>
      { % endif % }
    <!-- /.table-responsive -->
  </div>
</div>
<!-- /.row -->
</div>
<!-- /.panel-body -->

```

```

        </div>
    </div>
</div>
<!-- /.row -->
</div>
{% endblock %}

```

Edit_product.html

```

{% extends 'pages/layout.html' %}
{% block body %}
<div id="page-wrapper">
    <div class="row">
        <div class="col-lg-12">
            <h1 class="page-header">Edit Product</h1>
            {% include 'includes/_flashmsg.html' %}
        </div>
        <!-- /.col-lg-12 -->
    </div>
    <!-- /.row -->
    <div class="row">
        <div class="col-lg-12">
            <div class="panel panel-default">
                <div class="panel-body">
                    <div class="row">
                        <div class="col-lg-6">
                            {% for product in product%}
                            <form role="form" enctype=multipart/form-data class=""
action="/edit_product?id={{ product.id }}"
method="post">
                                <div class="form-group">
                                    <label>Product Name</label>
                                    <input class="form-control" required="required" name="name"
value="{{ product.pName }}" placeholder="Enter name">
                                </div>
                                <div class="form-group">
                                    <label>Price</label>
                                    <input class="form-control" required="required" name="price"
value="{{ product.price }}" placeholder="Enter price">
                                </div>
                                <div class="form-group">
                                    <label>Description</label>
                                    <textarea class="form-control" required="required" name="description" rows="3"
placeholder="Write product description">{{ product.description }}</textarea>

```



```

</div>
<div class="form-group">
  <label>Available</label>
  <input class="form-control" required="required" value="{{ product.available }}"
name="available" placeholder="Available">
</div>
<div class="form-group">
  <label>Category</label>
  <select class="form-control" id="category_id" onchange="populateList();"
name="category">
    {% if product.category == 'tshirt' % }
    <option value="tshirt" selected>T-Shirt</option>
    {% else % }
    <option value="tshirt">T-Shirt</option>
    {% endif % }
    {% if product.category == 'wallet' % }
    <option value="wallet" selected>Wallet</option>
    {% else % }
    <option value="wallet">Wallet</option>
    {% endif % }
    {% if product.category == 'belt' % }
    <option value="belt" selected>Belt</option>
    {% else % }
    <option value="belt">Belt</option>
    {% endif % }
    {% if product.category == 'shoes' % }
    <option value="shoes" selected>Shoes</option>
    {% else % }
    <option value="shoes">Shoes</option>
    {% endif % }
  </select>
</div>
{% for product_level in product_level % }
<div class="form-group level-container">
  {% if product.category == 'tshirt' % }
  <label>level for T-shirt</label>
  <div class="form-group">
    <label class="checkbox-inline active">
      {% if product_level.v_shape == 'yes' % }
      <input type="checkbox" checked name="tshirt" value="v_shape">V-Shape
      {% else % }
      <input type="checkbox" name="tshirt" value="v_shape">V-Shape
      {% endif % }
    </label>
    <label class="checkbox-inline">

```

```

        { % if product_level.polo == 'yes' % }
        <input type="checkbox" checked name="tshirt" value="polo">Polo T-shirt
        { % else% }
        <input type="checkbox" name="tshirt" value="polo">Polo T-shirt
        { % endif % }
    </label>
    <label class="checkbox-inline">
        { % if product_level.clean_text == 'yes' % }
        <input type="checkbox" checked name="tshirt" value="clean_text">Clean Text
        { % else% }
        <input type="checkbox" name="tshirt" value="clean_text">Clean Text
        { % endif % }
    </label>
    <label class="checkbox-inline">
        { % if product_level.design == 'yes' % }
        <input type="checkbox" checked name="tshirt" value="design">Colorful Design
        { % else% }
        <input type="checkbox" name="tshirt" value="design">Colorful Design
        { % endif % }
    </label>
</div>
{ % endif % }
{ % if product.category == 'wallet' % }
<label>Level for Wallet</label>
<div class="form-group">
    <label class="checkbox-inline">
        { % if product_level.chain == 'yes' % }
        <input type="checkbox" checked name="wallet" value="chain">Chain
        { % else% }
        <input type="checkbox" name="wallet" value="chain">Chain
        { % endif % }
    </label>
    <label class="checkbox-inline">
        { % if product_level.leather == 'yes' % }
        <input type="checkbox" checked name="wallet" value="leather">Leather
        { % else% }
        <input type="checkbox" name="wallet" value="leather">Leather
        { % endif % }
    </label>
    <label class="checkbox-inline">
        { % if product_level.design == 'yes' % }
        <input type="checkbox" checked name="wallet" value="design">Design
        { % else% }
        <input type="checkbox" name="wallet" value="design">Design
        { % endif % }
    </label>
</div>

```

```

    </label>
  </div>
  {% endif % }
  {% if product.category == 'belt' % }
  <label>Level for Belt</label>
  <div class="form-group">
    <label class="checkbox-inline">
      {% if product_level.leather == 'yes' % }
      <input type="checkbox" checked name="belt" value="leather">Leather
      {% else% }
      <input type="checkbox" name="belt" value="leather">Leather
      {% endif % }
    </label>
    <label class="checkbox-inline">
      {% if product_level.hook == 'yes' % }
      <input type="checkbox" checked name="belt" value="hook">Hook
      {% else% }
      <input type="checkbox" name="belt" value="hook">Hook
      {% endif % }
    </label>
    <label class="checkbox-inline">
      {% if product_level.color == 'yes' % }
      <input type="checkbox" checked name="belt" value="color">Color
      {% else% }
      <input type="checkbox" name="belt" value="color">Color
      {% endif % }
    </label>
    <label class="checkbox-inline">
      {% if product_level.design == 'yes' % }
      <input type="checkbox" checked name="belt" value="design">Design
      {% else% }
      <input type="checkbox" name="belt" value="design">Design
      {% endif % }
    </label>
  </div>
  {% endif % }
  {% if product.category == 'shoes' % }
  <label>Level for Shoes</label>
  <div class="form-group">
    <label class="checkbox-inline">
      {% if product_level.formal == 'yes' % }
      <input type="checkbox" checked name="shoes" value="formal">Formal
      {% else% }
      <input type="checkbox" name="shoes" value="formal">Formal
      {% endif % }
    </label>
  </div>
  {% endif % }

```

```

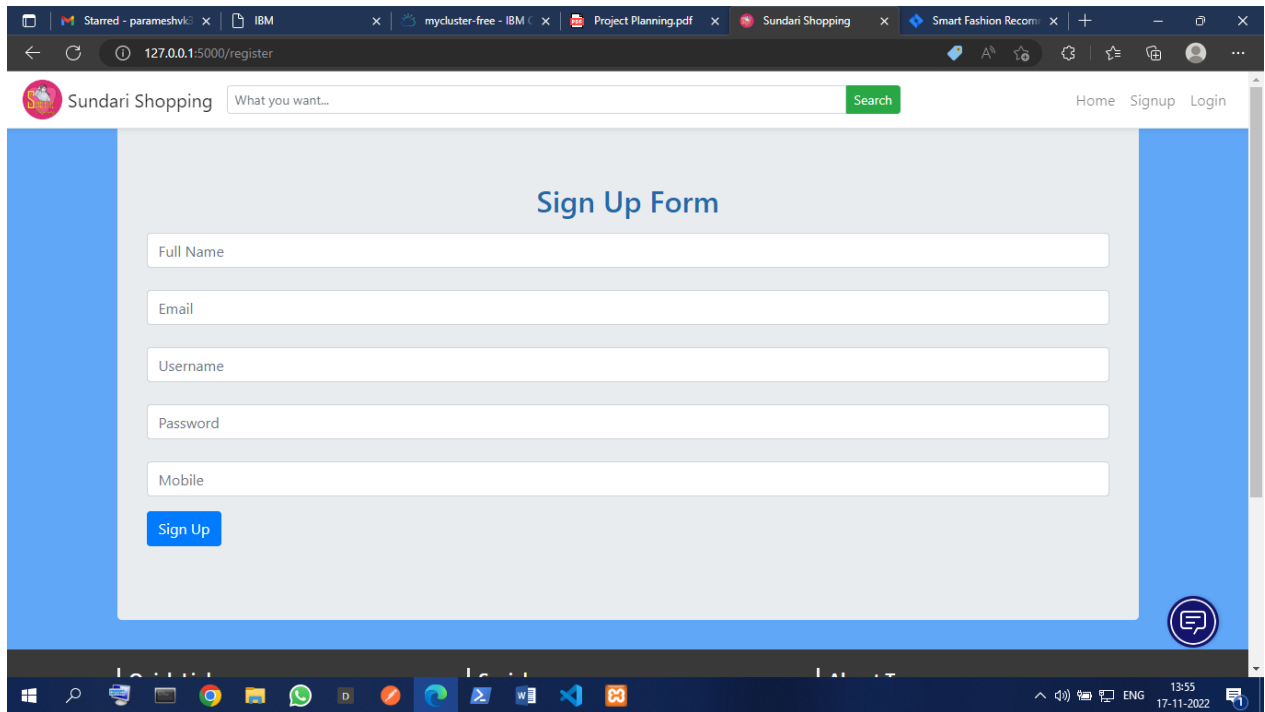
        </label>
        <label class="checkbox-inline">
            { % if product_level.converse == 'yes' % }
            <input type="checkbox" checked name="shoes" value="converse">Converse
            { % else% }
            <input type="checkbox" name="shoes" value="converse">Converse
            { % endif % }
        </label>
        <label class="checkbox-inline">
            { % if product_level.loafer == 'yes' % }
            <input type="checkbox" checked name="shoes" value="loafer">Loafer
            { % else% }
            <input type="checkbox" name="shoes" value="loafer">Loafer
            { % endif % }
        </label>
        <label class="checkbox-inline">
            { % if product_level.leather == 'yes' % }
            <input type="checkbox" name="shoes" checked value="leather">Leather
            { % else% }
            <input type="checkbox" name="shoes" value="leather">Leather
            { % endif % }
        </label>
    </div>
    { % endif % }
</div>
{ % endfor % }
<div class="form-group">
    <label>Item</label>
    <input class="form-control" required="required" name="item"
value="{{ product.item }}" placeholder="Ex: watch/shoes">
</div>
<div class="form-group">
    <label>Product Code</label>
    <input class="form-control" required="required" value="{{ product.pCode }}"
name="code" placeholder="Enter code">
</div>
<div class="form-group">
    <label>Add Image</label>
    <input type="file" name="picture">
</div>
<button type="submit" class="btn btn-default">Submit Button</button>
</form>
{ % endfor % }
</div>
<!-- /.col-lg-6 (nested) -->

```

```
        </div>
        <!-- /.row (nested) -->
    </div>
    <!-- /.panel-body -->
</div>
<!-- /.panel -->
</div>
<!-- /.col-lg-12 -->
</div>
<!-- /.row -->
</div>
<!-- /#page-wrapper -->
{% endblock %}
```

SCREENSHOTS

Register page



The screenshot displays the 'Sundari Shopping' website's registration page. The browser's address bar shows the URL '127.0.0.1:5000/register'. The page features a blue header with the site's logo, a search bar, and links for 'Home', 'Signup', and 'Login'. The main content area is a light gray box titled 'Sign Up Form' with a blue border. It contains five input fields: 'Full Name', 'Email', 'Username', 'Password', and 'Mobile'. A blue 'Sign Up' button is positioned below the 'Mobile' field. A chat icon is visible in the bottom right corner of the form area. The Windows taskbar at the bottom shows the system time as 13:55 on 17-11-2022.

Sundari Shopping

What you want... Search

Home Signup Login

Sign Up Form

Full Name

Email

Username

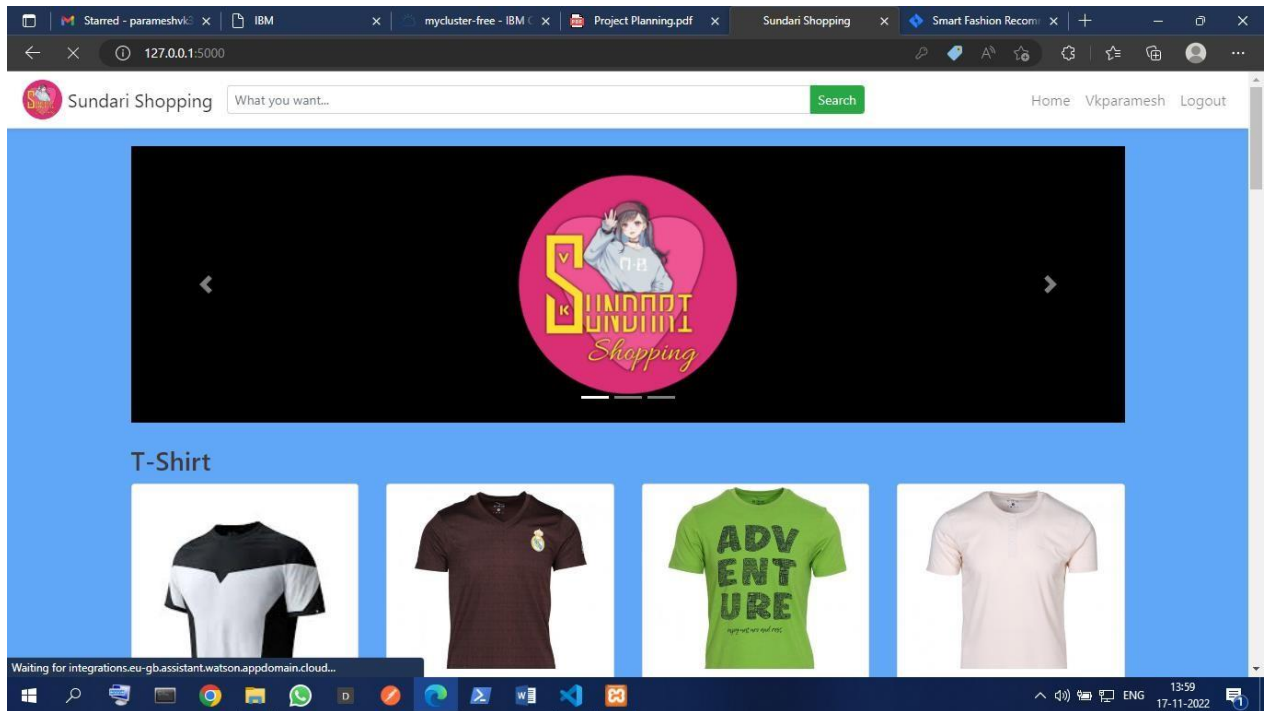
Password

Mobile

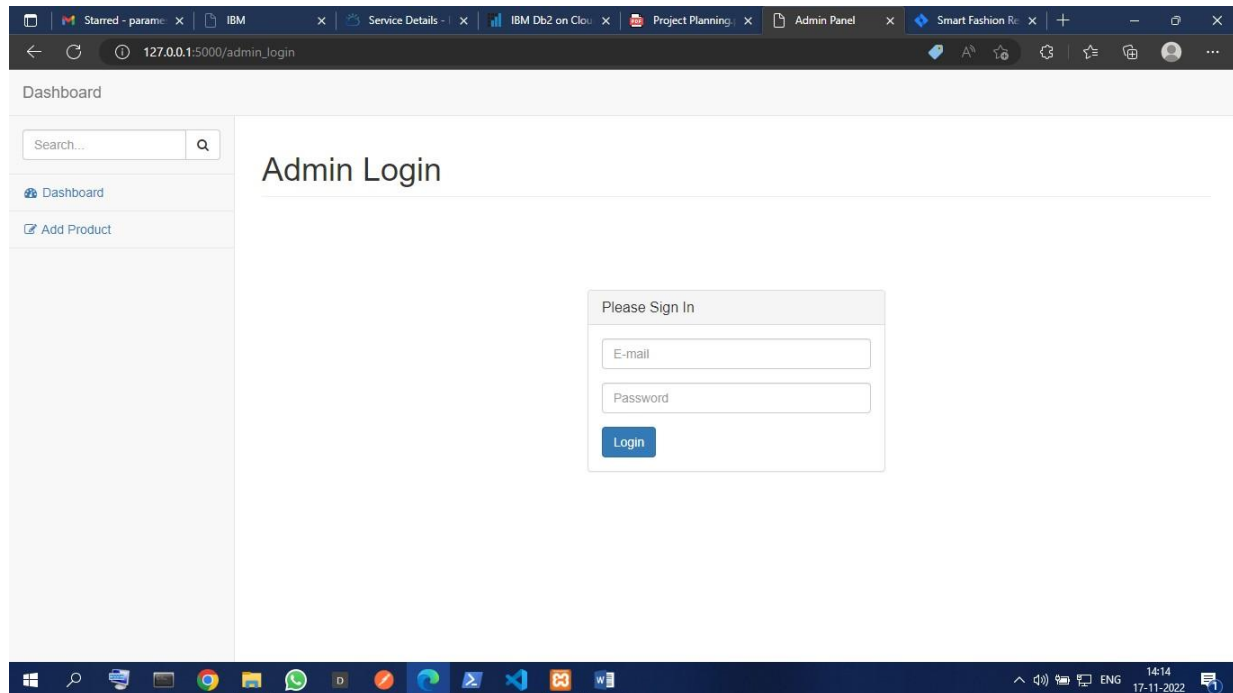
Sign Up

13:55
17-11-2022

Home page



Admin Login Page



Admin Dashboard Page

parameshwaran

Search...

Dashboard

Add Product

Dashboard

21
All Product
[View Details](#)

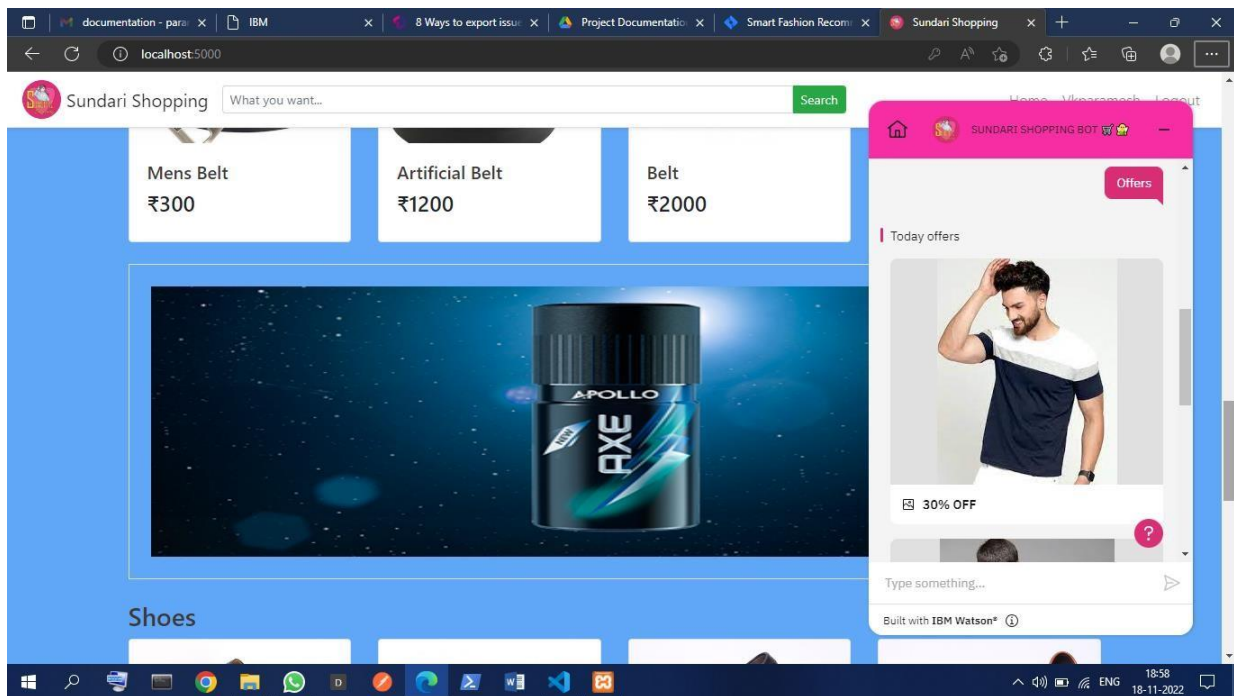
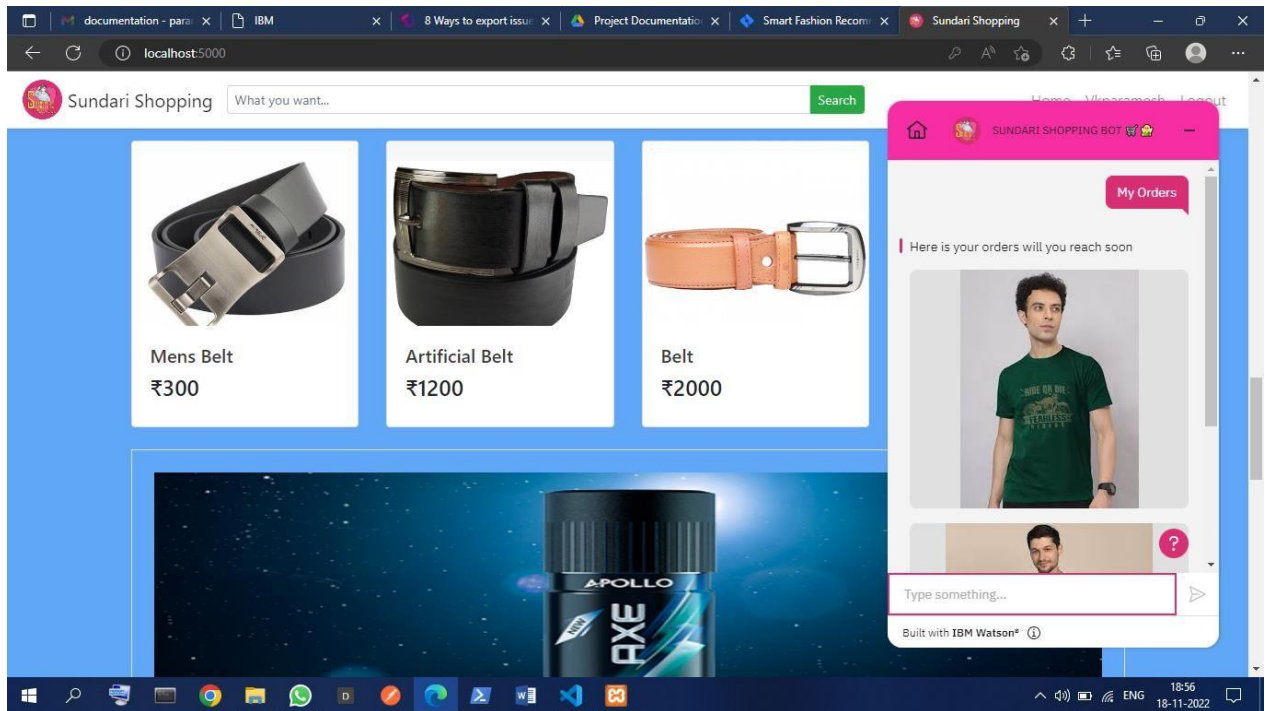
6
All Users
[View Details](#)

24
All Orders
[View Details](#)

All Product List

Id	Name	Price	Description	Available	Category	Item	Code	Picture
1	T-Shirt	120	T-Shirt	4	tshirt	t-shirt	t-007	Edit
2	Baborry wallet	6000	Baborry-Double-Zipper-Coin-Bag-RFID-Blocking-Men-Wallets-New-Brand-PU-Leather-Wallet-Money-Purses	3	wallet	wallet	w-004	Edit
3	Loafer Shoes	2000	Loafer black shoes	8	shoes	shoes	s-001	Edit
4	Artificial Belt	1200	Black artificial belt	9	belt	belt	b-001	Edit
5	Polo T-	500	Polo t-shirt	10	tshirt	t-shirt	s-002	Edit

Integrate Chat Bot



2. GitHub & Project Demo Link

- Our GitHub Repository Direct Link
- <https://github.com/IBM-EPBL/IBM-Project-37487-1660310518>
- Project Demonstration Video Direct Link
shorturl.at/eiGM4