SMART FASHION RECOMMENDER APPLICATION

IBM – DOCUMENTATION

UNDER THE GUIDANCE OF

INDUSTRY MENTOR(S) NAME : KRISHNA CHAITANYA

FACULTY MENTOR(S) NAME : SEENUVASAN P

TEAM ID : PNT2022TMID29381

SUBMITTED BY:

KUBERAN V	422519205021
SHARULATHA I	422519205038
MARUTHAN G	422519205024
JAFRI MELBA W	422519205017

DEPARTMENT OF INFORMATION TECHNOLOGY

UNIVERSITY COLLEGE OF ENGINEERING,VILLUPURAM

ANNA UNIVERSITY :: 2019 – 2023

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:

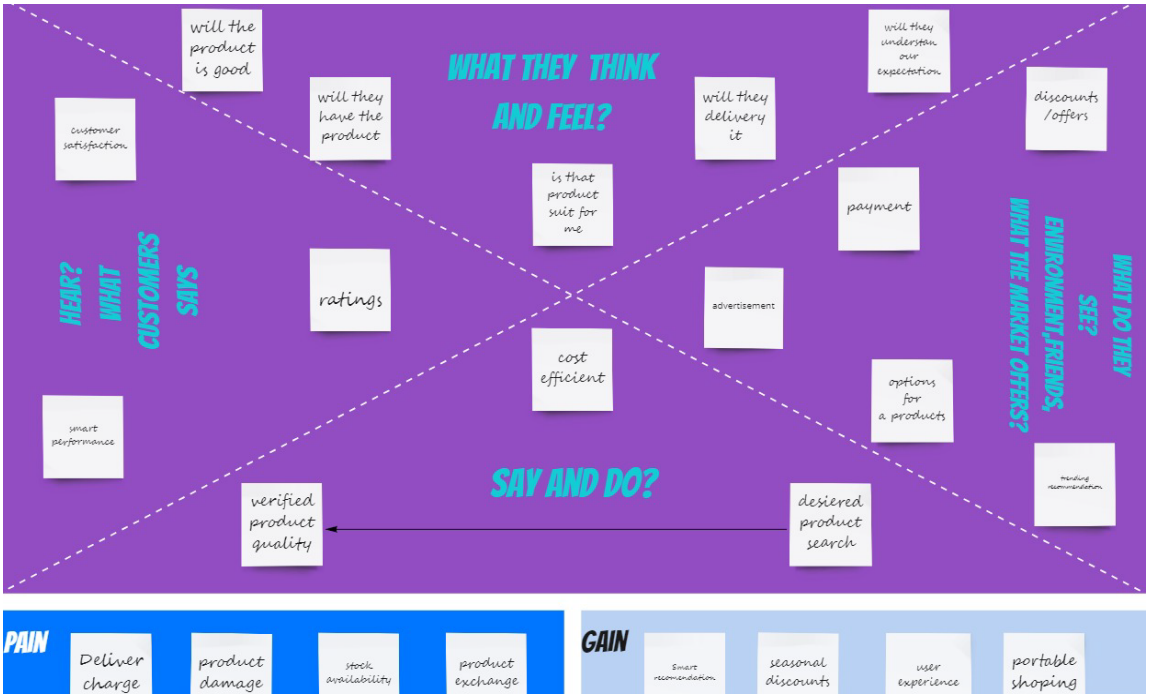
The User Needs a way to Assistant for finding Clothes so that Here User got the Chat Bot assistant

Problem Statement 4:

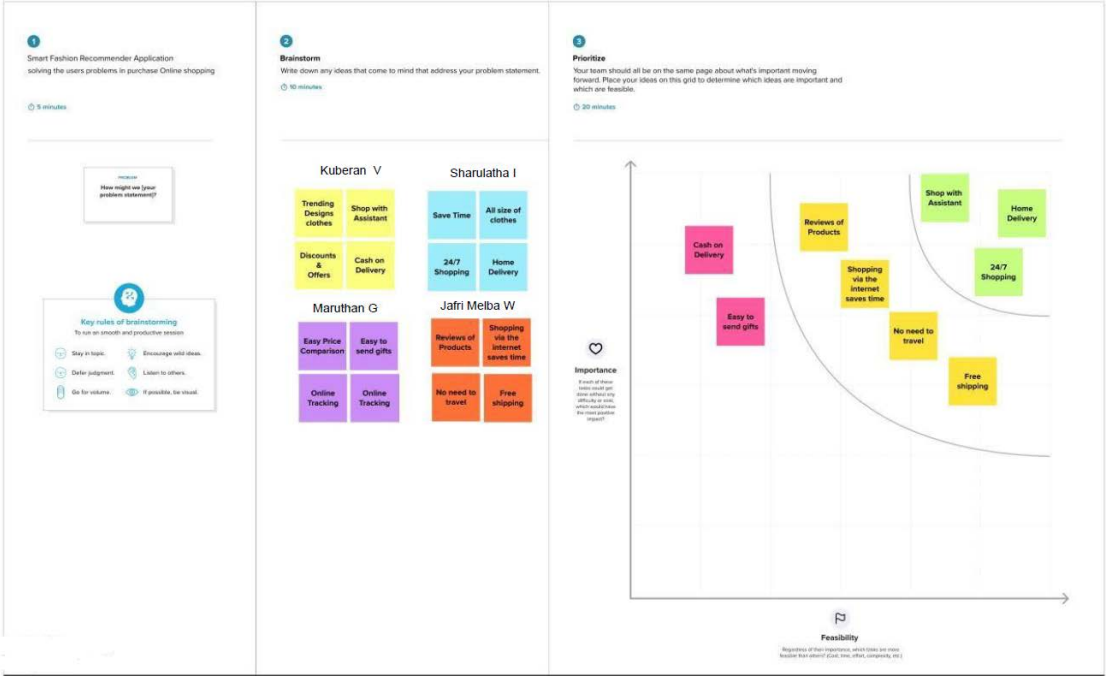
The Sellers Needs a way to struggling to sells products offline so that Here Sellers will Sell Products via our application.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 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

Problem-Solution fit canvas 2.0

Smart Fashion Recommender Application

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

- Customers are those who want to purchase fashion items in a short time

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.

- Most of the solution available in the internet hosts a lot of adds limiting its usability.
- Needs a proper network connection

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 dothese solutions have? i.e. pen and paper is an alternative to digital notetaking

- Smart Fashion Recommender which are supported in many browsers
- Smart Fashion Recommender Chatbot is developed in thisproject.

Explore AS,

Focus on J&P, tap into BE,

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.

- To collect data about our visitors and leverage it to make better product suggestions and recommendations
- Understanding customer inquiries, their needs, and preferences can allow you to personalize product pages and build customer loyalty and affinity.

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.

- For No-Pressure Shopping Experiences
- Customer service will be available for 24/7
- Chatbot can help with recovering abandoned carts

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)

- Seamless Real-Life Interaction
- Customer Data Security
- Reduce Customer Frustration

Focus on J&P, tap into BE,

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installingsolar panels, reading about a more efficient solution in the news.

- Improve Lead Generation.
- Reduce Customer Service Costs.
- Monitor Consumer Data to Gain Insights.

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.

- Took longer time to process and respond to the query

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 inthe canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

- Instead of navigating to several screens for booking products online, the user can directly talk to Chatbot regarding the products.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

- Able to serve customers with a consistent level of quality in a short period of time across different channels,

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

- Make sure they are aware of the usage of the chatbots

Extract online & offline CH of BE

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

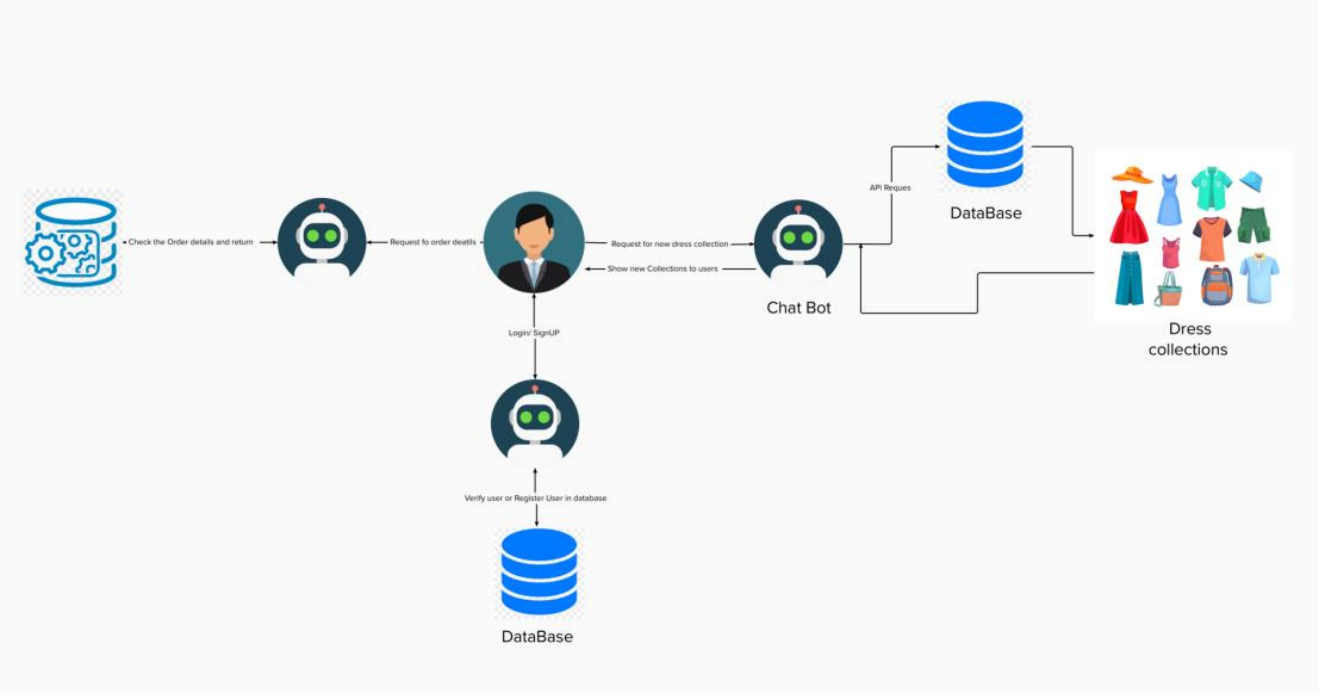
Following are the non-functional requirements of the proposed solution.

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.

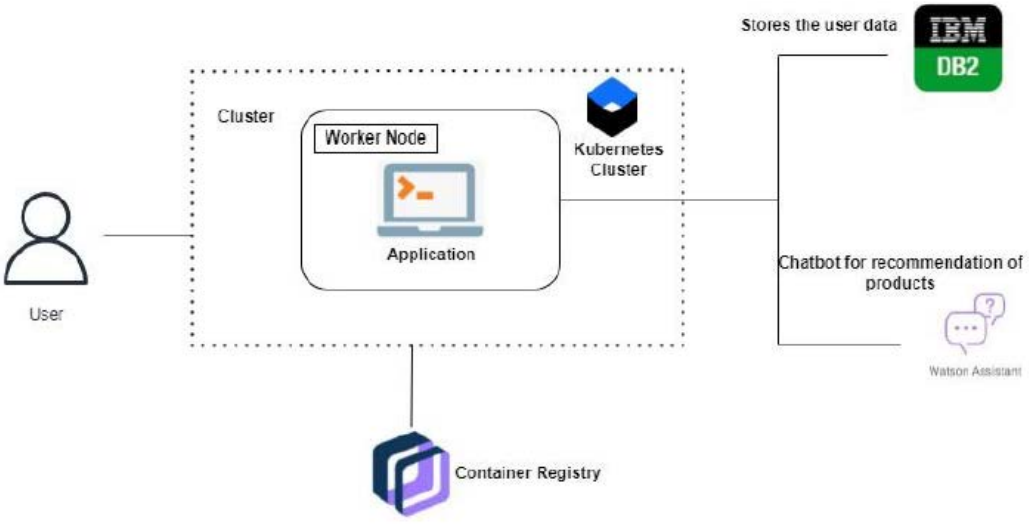
5. PROJECT DESIGN

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	Sprit -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

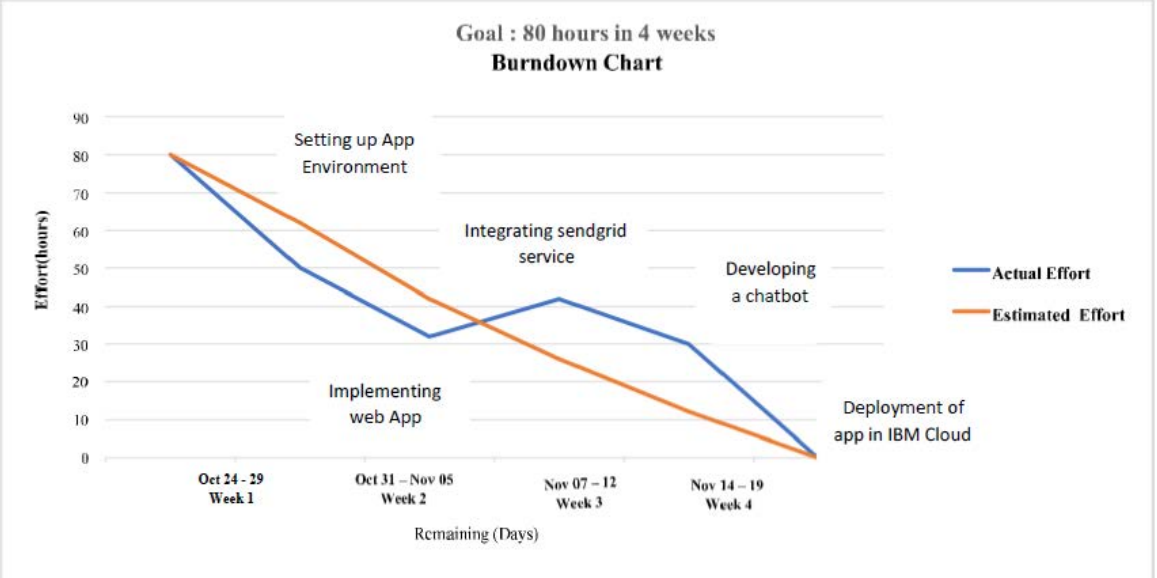
6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story points	Priority	Team Members
Sprint-1	Setting up App environment	USN-1	As a user, I can register in ICTA Academy and create IBM cloud account.	2	High	Kuberan V Sharulatha I
Sprint-1		USN-2	As a user, I will create a flask project	1	Low	Maruthan G Jafri melba W
Sprint-1		USN-3	As a user, I will install IBM Cloud CLI	2	Medium	Kuberan V Maruithan G
Sprint-2	Setting up App environment	USN-4	As a user, I can install Docker CLI	1	Low	Kuberan V Sharulatha I
Sprint-2		USN-5	As a user, I will Create an account in sendgrid	2	Medium	Maruthan G Jafri melba W

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	18	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	18	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	18	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	18	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

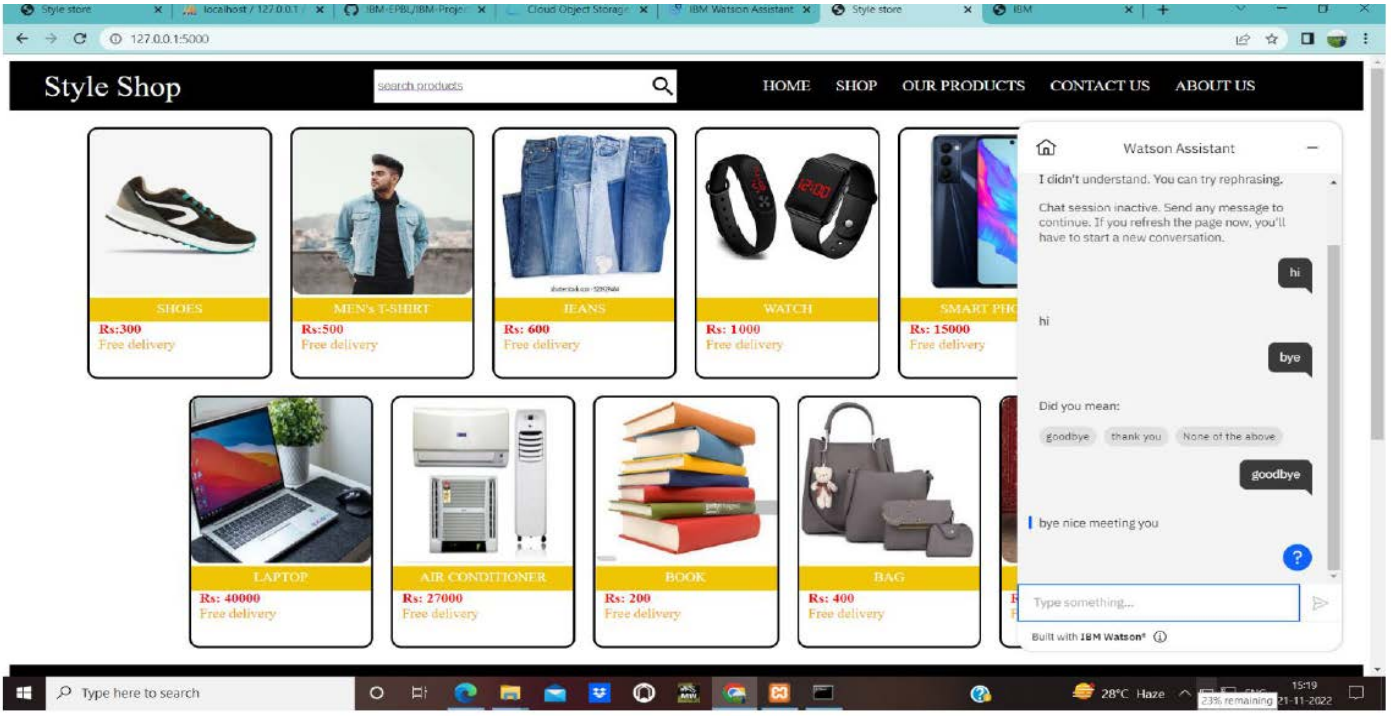
6.3 REPORTS FROM JIRA



7.CODING & SOLUTIONING

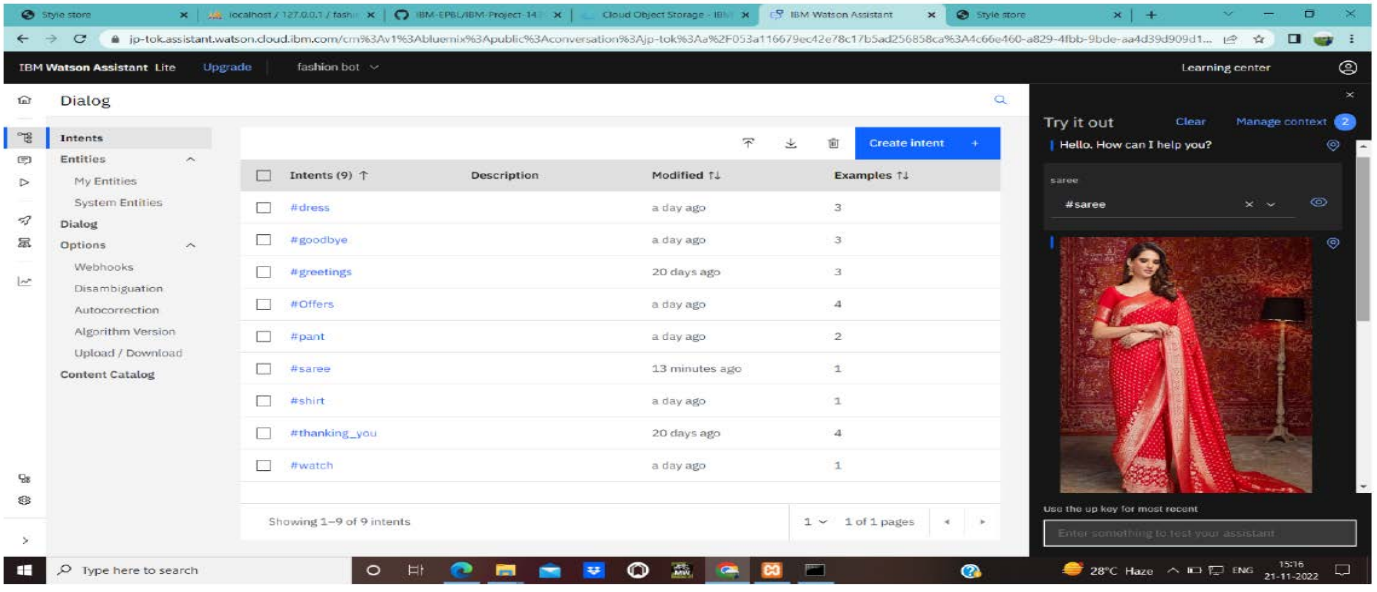
7.1 FEATURE 1

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

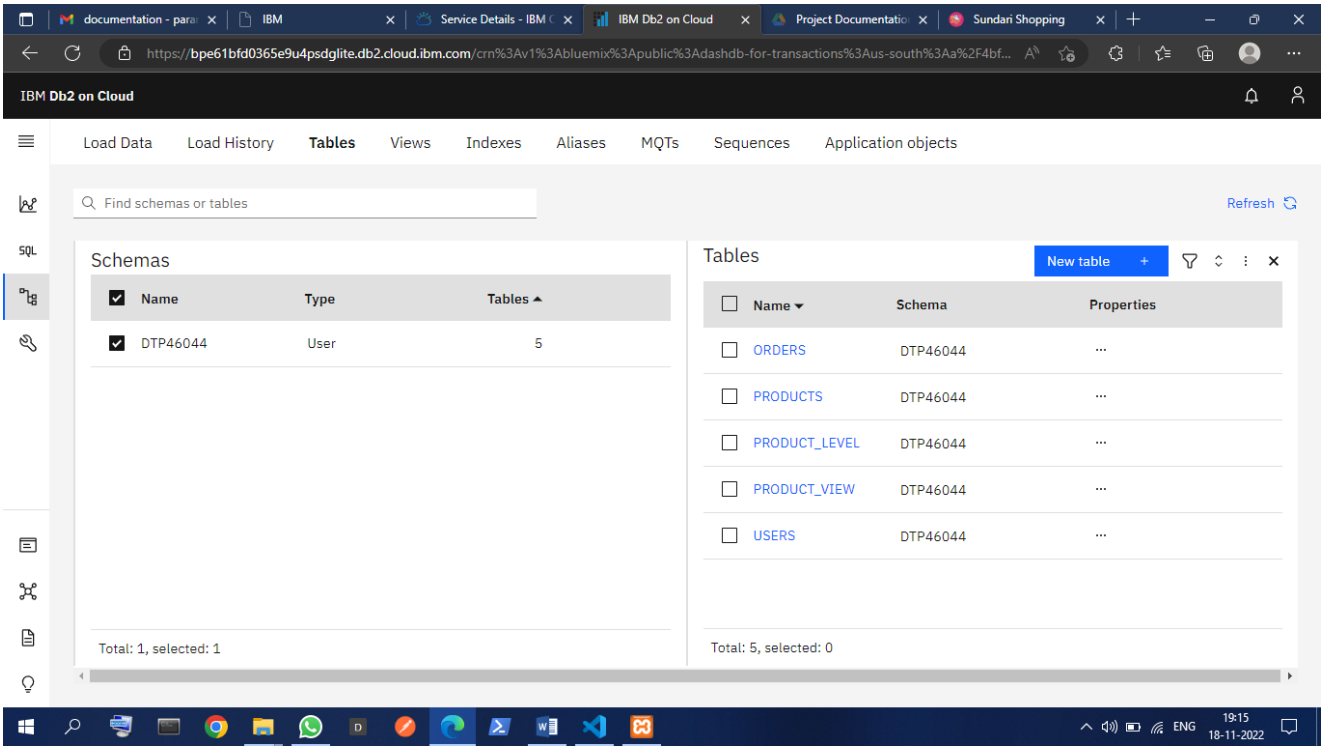


7.2 FEATURE 2

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



7.3 DATABASE SCHEMA



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/Voluem Changes	Risk Score	Justification
1	Smart Fashion Recommender Application	New	Low	No Changes	Moderate		>5 to 10%	ORANGE	As we have seen the chnages

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	vkparameshwaran

			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)	Approvals/SignOff
1	**Smart Fashion Recommender Application**	Manule		Worked as we expected		Use Laptop / desktop Mode	No Defects	Kuberan V

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 received 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 stat-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.

13. APPENDIX

13.1 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)
```

```
            userid = account['USERNAME']
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'

            msg = 'Logged in successfully !'
            return render_template('main.html', msg=msg)
        else:
            msg = 'Incorrect username / password !'
    else:
        return render_template('login.html', msg=msg)




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

@app.route('/signup', methods = ('POST','GET'))
def signup():
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM users WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
```

```
        print(account)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'name must contain only characters and numbers !'
        else:
            insert_sql = "INSERT INTO  users VALUES (?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, password)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully registered !'
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg=msg)

@app.route('/shoes', methods = ('POST','GET'))
def shoes():
        msg = ''
        if request.method == 'POST':
                product = 'shoes'
                name = request.form['name']
                mail = request.form['mail']
                address = request.form['address']
                mobile = request.form['mobile']
                quantity = request.form['quantity']
```

```
            sql = "SELECT * FROM orders;"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
            print(account)
            insert_sql = "INSERT INTO  orders VALUES (?, ?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, product)
            ibm_db.bind_param(prep_stmt, 3, mail)
            ibm_db.bind_param(prep_stmt, 4, address)
            ibm_db.bind_param(prep_stmt, 5, quantity)
            ibm_db.bind_param(prep_stmt, 6, mobile)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully ordered !'
            return render_template('shoes.html',msg=msg)
        return render_template('shoes.html',msg=msg)

@app.route('/saree', methods = ('POST','GET'))
def saree():
        msg = ''
        if request.method == 'POST':
            product = 'saree'
            name = request.form['name']
            mail = request.form['mail']
            address = request.form['address']
            mobile = request.form['mobile']
            quantity = request.form['quantity']
            sql = "SELECT * FROM orders;"
```

```
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
            print(account)
            insert_sql = "INSERT INTO  orders VALUES (?, ?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, product)
            ibm_db.bind_param(prep_stmt, 3, mail)
            ibm_db.bind_param(prep_stmt, 4, address)
            ibm_db.bind_param(prep_stmt, 5, quantity)
            ibm_db.bind_param(prep_stmt, 6, mobile)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully ordered !'
            return render_template('saree.html',msg=msg)
        return render_template('saree.html',msg=msg)

@app.route('/bag', methods = ('POST','GET'))
def bag():
        msg = ''
        if request.method == 'POST':
            product = 'bag'
            name = request.form['name']
            mail = request.form['mail']
            address = request.form['address']
            mobile = request.form['mobile']
            quantity = request.form['quantity']
            sql = "SELECT * FROM orders;"
            stmt = ibm_db.prepare(conn, sql)
```

```
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
            print(account)
            insert_sql = "INSERT INTO  orders VALUES (?, ?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, product)
            ibm_db.bind_param(prep_stmt, 3, mail)
            ibm_db.bind_param(prep_stmt, 4, address)
            ibm_db.bind_param(prep_stmt, 5, quantity)
            ibm_db.bind_param(prep_stmt, 6, mobile)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully ordered !'
            return render_template('bag.html',msg=msg)
        return render_template('bag.html',msg=msg)

@app.route('/book')
def book():
        msg = ''
        if request.method == 'POST':
            product = 'book'
            name = request.form['name']
            mail = request.form['mail']
            address = request.form['address']
            mobile = request.form['mobile']
            quantity = request.form['quantity']
            sql = "SELECT * FROM orders;"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt)
```

```
            account = ibm_db.fetch_assoc(stmt)
            print(account)
            insert_sql = "INSERT INTO  orders VALUES (?, ?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, product)
            ibm_db.bind_param(prep_stmt, 3, mail)
            ibm_db.bind_param(prep_stmt, 4, address)
            ibm_db.bind_param(prep_stmt, 5, quantity)
            ibm_db.bind_param(prep_stmt, 6, mobile)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully ordered !'
            return render_template('book.html',msg=msg)
        return render_template('book.html',msg=msg)

@app.route('/laptop', methods = ('POST','GET'))
def laptop():
        msg = ''
        if request.method == 'POST':
            product = 'laptop'
            name = request.form['name']
            mail = request.form['mail']
            address = request.form['address']
            mobile = request.form['mobile']
            quantity = request.form['quantity']
            sql = "SELECT * FROM orders;"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
```

```
            print(account)
            insert_sql = "INSERT INTO  orders VALUES (?, ?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, product)
            ibm_db.bind_param(prep_stmt, 3, mail)
            ibm_db.bind_param(prep_stmt, 4, address)
            ibm_db.bind_param(prep_stmt, 5, quantity)
            ibm_db.bind_param(prep_stmt, 6, mobile)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully ordered !'
            return render_template('laptop.html',msg=msg)
        return render_template('laptop.html',msg=msg)

@app.route('/tv', methods = ('POST','GET'))
def tv():
        msg = ''
        if request.method == 'POST':
            product = 'Television'
            name = request.form['name']
            mail = request.form['mail']
            address = request.form['address']
            mobile = request.form['mobile']
            quantity = request.form['quantity']
            sql = "SELECT * FROM orders;"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
            print(account)
```

```
            insert_sql = "INSERT INTO  orders VALUES (?, ?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, product)
            ibm_db.bind_param(prep_stmt, 3, mail)
            ibm_db.bind_param(prep_stmt, 4, address)
            ibm_db.bind_param(prep_stmt, 5, quantity)
            ibm_db.bind_param(prep_stmt, 6, mobile)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully ordered !'
            return render_template('tv.html',msg=msg)
        return render_template('tv.html',msg=msg)

@app.route('/phone', methods = ('POST','GET'))
def phone():
        msg = ''
        if request.method == 'POST':
            product = 'phone'
            name = request.form['name']
            mail = request.form['mail']
            address = request.form['address']
            mobile = request.form['mobile']
            quantity = request.form['quantity']
            sql = "SELECT * FROM orders;"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
            print(account)
            insert_sql = "INSERT INTO  orders VALUES (?, ?, ?, ?, ?, ?)"
```

```
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, product)
            ibm_db.bind_param(prep_stmt, 3, mail)
            ibm_db.bind_param(prep_stmt, 4, address)
            ibm_db.bind_param(prep_stmt, 5, quantity)
            ibm_db.bind_param(prep_stmt, 6, mobile)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully ordered !'
            return render_template('phone.html',msg=msg)
        return render_template('phone.html',msg=msg)

@app.route('/watch', methods = ('POST','GET'))
def watch():
        msg = ''
        if request.method == 'POST':
            product = 'watch'
            name = request.form['name']
            mail = request.form['mail']
            address = request.form['address']
            mobile = request.form['mobile']
            quantity = request.form['quantity']
            sql = "SELECT * FROM orders;"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
            print(account)
            insert_sql = "INSERT INTO  orders VALUES (?, ?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
```

```
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, product)
            ibm_db.bind_param(prep_stmt, 3, mail)
            ibm_db.bind_param(prep_stmt, 4, address)
            ibm_db.bind_param(prep_stmt, 5, quantity)
            ibm_db.bind_param(prep_stmt, 6, mobile)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully ordered !'
            return render_template('watch.html',msg=msg)
        return render_template('watch.html',msg=msg)

@app.route('/shirt', methods = ('POST','GET'))
def shirt():
        msg = ''
        if request.method == 'POST':
            product = 'shirt'
            name = request.form['name']
            mail = request.form['mail']
            address = request.form['address']
            mobile = request.form['mobile']
            quantity = request.form['quantity']
            sql = "SELECT * FROM orders;"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
            print(account)
            insert_sql = "INSERT INTO  orders VALUES (?, ?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
```

```
            ibm_db.bind_param(prep_stmt, 2, product)
            ibm_db.bind_param(prep_stmt, 3, mail)
            ibm_db.bind_param(prep_stmt, 4, address)
            ibm_db.bind_param(prep_stmt, 5, quantity)
            ibm_db.bind_param(prep_stmt, 6, mobile)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully ordered !'
            return render_template('shirt.html',msg=msg)
        return render_template('shirt.html',msg=msg)

@app.route('/jeans', methods = ('POST','GET'))
def jeans():
        msg = ''
        if request.method == 'POST':
            product = 'jeanspant'
            name = request.form['name']
            mail = request.form['mail']
            address = request.form['address']
            mobile = request.form['mobile']
            quantity = request.form['quantity']
            sql = "SELECT * FROM orders;"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
            print(account)
            insert_sql = "INSERT INTO  orders VALUES (?, ?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, product)
```

```
                ibm_db.bind_param(prep_stmt, 3, mail)
                ibm_db.bind_param(prep_stmt, 4, address)
                ibm_db.bind_param(prep_stmt, 5, quantity)
                ibm_db.bind_param(prep_stmt, 6, mobile)
                ibm_db.execute(prep_stmt)
                msg = 'You have successfully ordered !'
                return render_template('jeans.html',msg=msg)
        return render_template('jeans.html',msg=msg)

@app.route('/admin', methods=['GET', 'POST'])
def admin():
    global userid
    msg = ''

    if request.method == 'POST':
        name = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM admin WHERE name =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, name)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['loggedin'] = True
            session['id'] = account['NAME']
            userid = account['NAME']
            session['name'] = account['NAME']
            msg = 'Logged in successfully !'
```

```
        msg = 'Logged in successfully !'
        return render_template('adminportal.html', msg=msg)
    else:
        msg = 'Incorrect username / password !'
  else:
    return render_template('admin.html', msg=msg)

@app.route('/users')
def users():
        global userid
        accounts = []
        sql = "SELECT * FROM users"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        while account:
                accounts.append(account)
                account = ibm_db.fetch_assoc(stmt)
        return render_template('users.html',data=accounts)

@app.route('/orders')
def orders():
        global userid
        accounts = []
        sql = "SELECT * FROM orders"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.execute(stmt)
```

```
        order = ibm_db.fetch_assoc(stmt)
        while order:
                accounts.append(order)
                order = ibm_db.fetch_assoc(stmt)
        return render_template('orders.html',data=accounts)
```

Home.html

```
<!DOCTYPE html>
 <html lang="en">
 <head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Style store</title>
  <link rel="stylesheet" href="/static/main.css">
  <link href="https://unpkg.com/ionicons@4.5.10-0/dist/css/ionicons.min.css" width="10%" >
 </head>
 <body>
  <header>
   <div class="logo"><a href="#">Style Shop</a></div>
   <div class="menu">
    <a href=""><ion-icon name="close" class="close"></ion-icon></a>
    <ul>
     <li><a href="/" class="under">HOME</a></li>
     <li><a href="/login" class="under">Login</a></li>



     <li><a href="/about" class="under">About us</a></li>
        <li><a href="/signup" class="under">Signup</a></li>
```

```
    </ul>
  </div>
  <div class="search">
    <a href=""><input type="text" placeholder="search products" id="input">
      <ion-icon class="s" name="search"></ion-icon>
    </a>
  </div>
  <div class="heading">
    <ul>
      <li><a href="/" class="under">HOME</a></li>
      <li><a href="/login" class="under">LOGIN</a></li>

      <li><a href="/about" class="under">ABOUT US</a></li>
                                <li><a href="/signup" class="under">SIGNUP</a></li>
    </ul>
  </div>
  <div class="heading1">
    <ion-icon name="menu" class="ham"></ion-icon>
  </div>
</header>
<section>
  <div class="section">
    <div class="section2">
      <div class="container">
        <div class="items">
          <div class="img img1"><img
              src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/shoos.jpg"
              alt=""></div>
          <div class="name">SHOES</div>
```

```
  <div class="price">Rs:300</div>
  <div class="info"><a href="/shoes">Buy</a></div>
</div>
<div class="items">
  <div class="img img2"><img
      src="https://images.pexels.com/photos/3649765/pexels-photo-
3649765.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500"
      alt=""></div>
  <div class="name">MEN's T-SHIRT</div>
  <div class="price">Rs:500</div>
  <div class="info"><a href="/shirt">Buy</a></div>
</div>
<div class="items">
  <div class="img img3"><img
      src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/jeans.jpeg"
      alt=""></div>
  <div class="name">JEANS</div>
  <div class="price">Rs: 600</div>
  <div class="info"><a href="/jeans">Buy</a></div>
</div>
<div class="items">
  <div class="img img1"><img
      src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/watch.jpeg"
      alt=""></div>
  <div class="name">WATCH</div>
  <div class="price">Rs: 1000</div>
  <div class="info"><a href="/watch">Buy</a></div>
</div>
<div class="items">
```

```
  <div class="img img1"><img
      src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/mobile.jpg"
      alt=""></div>
  <div class="name">SMART PHONE</div>
  <div class="price">Rs: 15000</div>
  <div class="info"><a href="/phone">Buy</a></div>
</div>
<div class="items">
  <div class="img img1"><img
      src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/tv.jpeg"
      alt=""></div>
  <div class="name">TELEVISION</div>
  <div class="price">Rs: 20000</div>
  <div class="info"><a href="/tv">Buy</a></div>
</div>


<div class="items">
  <div class="img img1"><img
      src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/lap.jpeg"
      alt=""></div>
  <div class="name">LAPTOP</div>
  <div class="price">Rs: 40000</div>
  <div class="info"><a href="/laptop">Buy</a></div>
</div>


<div class="items">
  <div class="img img1"><img
```

```
            src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/book.jpeg"
            alt=""></div>
          <div class="name">BOOK</div>
          <div class="price">Rs: 200</div>
          <div class="info"><a href="/book">Buy</a></div>
        </div>
        <div class="items">
          <div class="img img1"><img
            src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/bag.jpeg"
            alt=""></div>
          <div class="name">BAG</div>
          <div class="price">Rs: 400</div>
          <div class="info"><a href="/bag">Buy</a></div>
        </div>
        <div class="items">
          <div class="img img1"><img
            src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/saree.jpeg"
            alt=""></div>
          <div class="name">SAREES</div>
          <div class="price">Rs: 700</div>
          <div class="info"><a href="/saree">Buy</a></div>
        </div>
      </div>
    </div>
  </div>
</section>
<footer>
  <div class="footer0">
    <h1>StyleShop</h1>
```

```
</div>
<div class="footer1 ">
  Connect with us at<div class="social-media">
    <a href="#">
      <ion-icon name="logo-facebook"></ion-icon>
    </a>
    <a href="#">
      <ion-icon name="logo-linkedin"></ion-icon>
    </a>
    <a href="#">
      <ion-icon name="logo-youtube"></ion-icon>
    </a>
    <a href="#">
      <ion-icon name="logo-instagram"></ion-icon>
    </a>
    <a href="#">
      <ion-icon name="logo-twitter"></ion-icon>
    </a>
  </div>
</div>
<div class="footer2">
  <div class="product">
    <div class="heading">Products</div>
    <div class="div">Sell your Products</div>
    <div class="div">Advertise</div>
    <div class="div">Pricing</div>
    <div class="div">Product Buisness</div>
  </div>
  <div class="services">
```

```
      <div class="heading">Services</div>
      <div class="div">Return</div>
      <div class="div">Cash Back</div>
      <div class="div">Affiliate Marketing</div>
      <div class="div">Others</div>
    </div>
    <div class="Company">
      <div class="heading">Company</div>
      <div class="div">Complaint</div>
      <div class="div">Careers</div>
      <div class="div">Affiliate Marketing</div>
      <div class="div">Support</div>
    </div>
    <div class="Get Help">
      <div class="heading">Our Team</div>
      <div class="div">Kuberan v</div>
      <div class="div">Maruthan G</div>
      <div class="div">Jafri W</div>
      <div class="div">Sharulatha I</div>
    </div>
  </div>
  <div class="footer3">Copyright © <h4>StyleShop</h4> 2021-2028</div>
 </footer>
 <script src="https://unpkg.com/ionicons@4.5.10-0/dist/ionicons.js"></script>

</body>
</html>
```

Admin.html

```
<!DOCTYPE html>

<html lang="en">


<head>

   <meta charset="UTF-8">

   <meta http-equiv="X-UA-Compatible" content="IE=edge">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <title>Admin Login Page</title>

   <!-- favicon -->

   <!-- <link rel="shortcut icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->

   <!-- <link rel="icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->

   <link rel="icon" type="image/png" sizes="16x16" href="/assets/img/favicon-32x32.png">

   <!-- bootstrap css cdn -->

   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"

      integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">

   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.css">

   <!-- css stylesheet -->

   <link rel="stylesheet" href="/static/style2.css">

   <!-- font styles cdn -->

   <link rel="preconnect" href="https://fonts.gstatic.com">

   <link href="https://fonts.googleapis.com/css2?family=Alegreya&display=swap" rel="stylesheet">

   <link href="https://fonts.googleapis.com/css2?family=Alegreya:wght@600&display=swap"
rel="stylesheet">

</head>
```

```
<body>

  <div class="login text-center mt-5">

    <form action="/admin" method="post">
      <h2 class="form-text-h2"> Login Form </h2>
      <div class="msg">{{ msg }}</div>
      <div class="form-group mb-3">
        <label class="label" for="username">Username</label>
        <input type="text" class="form-control" placeholder="Username" name="username" required>
      </div>
      <div class="form-group mb-3">
        <label class="label" for="password">Password</label>
        <input type="password" class="form-control" placeholder="Password" name="password"
required>
      </div>

      <button type="submit" id="button" class="btn btn-primary"> Login </button>

    </form>
  </div>



</body>

</html>
```

Login.html

```
<!DOCTYPE html>

<html lang="en">


<head>

   <meta charset="UTF-8">

   <meta http-equiv="X-UA-Compatible" content="IE=edge">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <title>Smart Fashion Login</title>

   <!-- favicon -->

   <!-- <link rel="shortcut icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->

   <!-- <link rel="icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->

   <link rel="icon" type="image/png" sizes="16x16" href="/assets/img/favicon-32x32.png">

   <!-- bootstrap css cdn -->

   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"

      integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">

   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.css">

   <!-- css stylesheet -->

   <link rel="stylesheet" href="/static/style3.css">

   <!-- font styles cdn -->

   <link rel="preconnect" href="https://fonts.gstatic.com">

   <link href="https://fonts.googleapis.com/css2?family=Alegreya&display=swap" rel="stylesheet">

   <link href="https://fonts.googleapis.com/css2?family=Alegreya:wght@600&display=swap"
rel="stylesheet">

</head>
```

```
<body>

  <div class="login text-center mt-5">

    <form action="/" method="post">
      <h2 class="form-text-h2"> Login Form </h2>
      <div class="msg">{{ msg }}</div>
      <div class="form-group mb-3">
        <label class="label" for="username">Username</label>
        <input type="text" class="form-control" placeholder="Username" name="username" required>
      </div>
      <div class="form-group mb-3">
        <label class="label" for="password">Password</label>
        <input type="password" class="form-control" placeholder="Password" name="password"
required>
      </div>

      <button type="submit" id="button" class="btn btn-primary"> Login </button>
      <div class="note mt-3 text-center">
        <!--Register form -->
        <p> Don't have an account yet? Click here to <a href="/signup">register! </a> </p>
      </div>
    </form>
  </div>



</body>

</html>
```

Adminportal.html

```
<html>

<head>

<title>Admiin-Portal</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body {

  background-image: url("https://img.freepik.com/free-vector/technology-face-circuit-diagram-background_1017-18300.jpg?w=2000");

  margin: 0;

  font-family: Arial, Helvetica, sans-serif;

}


.topnav {

  overflow: hidden;

  background-color: whitesmoke;

  height: 10%;

}


.topnav a {

  float: left;

  color: black;

  text-align: center;

  padding: 20px 30px;

  text-decoration: none;

  font-size: 17px;

}


.topnav a:hover {
```

```
  background-color: blueviolet;
  color: black;
}

.topnav a.active {
  background-color: #B8B8B8;
  color: white;
}
</style>
</head>
<body>
<div class="topnav">
  <a href="/orders" target="mainframe">Orders</a>
  <a href="/users" target="mainframe">Users</a>
  <a href="/admin">Signout</a>
  </div>
  <div class="main">
<iframe name="mainframe" height="100%" width="100%" ></iframe>
<div>
</body>
</html>
```

Bag.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Order here</title>
<style>
```

```
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-image: url("https://www.logicinbound.com/wp-content/uploads/2018/01/shutterstock_779835055-1024x737.jpg");
            width900vw;
            height:90vh;

}
button {
      background-color: #4CAF50;
      width: 100%;
       color: orange;
       padding: 15px;
       margin: 10px 0px;
       border: none;
       cursor: pointer;
        }
 form {
       border: 3px solid #f1f1f1;
            width: 50%;
            margin: 0 auto;
            border-radius: 30px;

   }
 input[type=text], input[type=password]{
       width: 100%;
       margin: 8px 0;
       padding: 12px 20px;
       display: inline-block;
```

```
        border: 2px solid green;
        box-sizing: border-box;
    }
 button:hover {
        opacity: 0.7;
    }




 .container {
        padding: 25px;
        background-color: lightblue;
            border-radius: 30px;
}
.div1
{

            float: left;
            width: 40%;
            padding: 25px;
        background-color: lightblue;
            border-radius: 30px;

}
.div2
{
            width: 50%;
            float: right;
}
```

```
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

</style>
</head>
<body>
<div class="div1">
<div class="pic">
<center><h1 style="background-color:orange;">Bag<h1></center>
<br>
               <img
src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/bag.jpeg"    width="300"
height="300" alt="your product">
</div>
</div>
<div class="div2">

   <form action="/bag" method="post">
       <div class="container">
         <label>Username : </label>
         <input type="text" placeholder="Enter Username" name="name" required>
         <label>quantity : </label>
```

```
        <input type="text" placeholder="number of quantity " name="quantity" required>
                <label>mobile : </label>
        <input type="text" placeholder="mobile number " name="mobile" required>
                <label>your address : </label>
        <input type="text" placeholder="address" name="address" required>
                <label>email : </label>
        <input type="text" placeholder="email" name="mail" required>
        <button type="submit">Order</button>
          </div>
   </form>
          <div class="msg">{{ msg }}</div><br>
          <center><a href="/main" target="_blank">Continue Shooping</a></center>
          </div>
</body>
</html>
```

Book.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Order here</title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-image: url("https://www.logicinbound.com/wp-content/uploads/2018/01/shutterstock_779835055-1024x737.jpg");
            width900vw;
            height:90vh;
```

```
}
button {
      background-color: #4CAF50;
      width: 100%;
       color: orange;
       padding: 15px;
       margin: 10px 0px;
       border: none;
       cursor: pointer;
        }
 form {
       border: 3px solid #f1f1f1;
             width: 50%;
             margin: 0 auto;
             border-radius: 30px;

   }
 input[type=text], input[type=password]{
       width: 100%;
       margin: 8px 0;
       padding: 12px 20px;
       display: inline-block;
       border: 2px solid green;
       box-sizing: border-box;
   }
 button:hover {
       opacity: 0.7;
   }
```

```
 .container {
        padding: 25px;
        background-color: lightblue;
              border-radius: 30px;
}
.div1
{

              float: left;
              width: 40%;
              padding: 25px;
        background-color: lightblue;
              border-radius: 30px;

}
.div2
{
              width: 50%;
              float: right;
}

a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
```

```
  display: inline-block;
}

</style>
</head>
<body>
<div class="div1">
<div class="pic">
<center><h1 style="background-color:orange;">Book<h1></center>
<br>
                <img
src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/book.jpeg"    width="300"
height="300" alt="your product">
</div>
</div>
<div class="div2">

   <form action="/book" method="post">
      <div class="container">
        <label>Username : </label>
        <input type="text" placeholder="Enter Username" name="name" required>
        <label>quantity : </label>
        <input type="text" placeholder="number of quantity " name="quantity" required>
                <label>mobile : </label>
        <input type="text" placeholder="mobile number " name="mobile" required>
                <label>your address : </label>
        <input type="text" placeholder="address" name="address" required>
                <label>email : </label>
        <input type="text" placeholder="email" name="mail" required>
```

```
        <button type="submit">Order</button>
          </div>
   </form>
          <div class="msg">{{ msg }}</div><br>
          <center><a href="/main" target="_blank">Continue Shooping</a></center>
          </div>
</body>
</html>
```

Jeans.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Order here</title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-image: url("https://www.logicinbound.com/wp-content/uploads/2018/01/shutterstock_779835055-1024x737.jpg");
          width900vw;
          height:90vh;


}
button {
      background-color: #4CAF50;
      width: 100%;
       color: orange;
       padding: 15px;
       margin: 10px 0px;
```

```
        border: none;
        cursor: pointer;
         }
  form {
        border: 3px solid #f1f1f1;
              width: 50%;
              margin: 0 auto;
              border-radius: 30px;

    }
  input[type=text], input[type=password]{
        width: 100%;
        margin: 8px 0;
        padding: 12px 20px;
        display: inline-block;
        border: 2px solid green;
        box-sizing: border-box;
    }
  button:hover {
        opacity: 0.7;
    }




  .container {
        padding: 25px;
        background-color: lightblue;
              border-radius: 30px;
 }
```

```
.div1
{

        float: left;
        width: 40%;
        padding: 25px;
    background-color: lightblue;
        border-radius: 30px;

}
.div2
{
        width: 50%;
        float: right;
}

a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

</style>
</head>
<body>
<div class="div1">
```

```
<div class="pic">
<center><h1 style="background-color:orange;">Jeans<h1></center>
<br>
                <img
src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/jeans.jpeg" width="300"
height="300" alt="your product">
</div>
</div>
<div class="div2">


  <form action="/jeans" method="post">
    <div class="container">
      <label>Username : </label>
      <input type="text" placeholder="Enter Username" name="name" required>
      <label>quantity : </label>
      <input type="text" placeholder="number of quantity " name="quantity" required>
            <label>mobile : </label>
      <input type="text" placeholder="mobile number " name="mobile" required>
            <label>your address : </label>
      <input type="text" placeholder="address" name="address" required>
            <label>email : </label>
      <input type="text" placeholder="email" name="mail" required>
      <button type="submit">Order</button>
        </div>
  </form>
        <div class="msg">{{ msg }}</div><br>
        <center><a href="/main" target="_blank">Continue Shooping</a></center>
        </div>
</body>    </html>
```

Laptop.html

```
<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Order here</title>

<style>

Body {

  font-family: Calibri, Helvetica, sans-serif;

  background-image: url("https://www.logicinbound.com/wp-content/uploads/2018/01/shutterstock_779835055-1024x737.jpg");

            width900vw;

            height:90vh;


}

button {

      background-color: #4CAF50;

      width: 100%;

       color: orange;

       padding: 15px;

       margin: 10px 0px;

       border: none;

       cursor: pointer;

        }

 form {

       border: 3px solid #f1f1f1;

            width: 50%;

            margin: 0 auto;

            border-radius: 30px;
```

```
  }
input[type=text], input[type=password]{
      width: 100%;
      margin: 8px 0;
      padding: 12px 20px;
      display: inline-block;
      border: 2px solid green;
      box-sizing: border-box;
  }
button:hover {
      opacity: 0.7;
  }




.container {
      padding: 25px;
      background-color: lightblue;
            border-radius: 30px;
}
.div1
{

            float: left;
            width: 40%;
            padding: 25px;
      background-color: lightblue;
            border-radius: 30px;
```

```
}
.div2
{
        width: 50%;
        float: right;
}

a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

</style>
</head>
<body>
<div class="div1">
<div class="pic">
<center><h1 style="background-color:orange;">Laptop<h1></center>
<br>
                <img
src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/lap.jpeg"    width="300"
height="300" alt="your product">
</div>
</div>
```

```
<div class="div2">


   <form action="/laptop" method="post">
       <div class="container">
         <label>Username : </label>
         <input type="text" placeholder="Enter Username" name="name" required>
         <label>quantity : </label>
         <input type="text" placeholder="number of quantity " name="quantity" required>
                    <label>mobile : </label>
         <input type="text" placeholder="mobile number " name="mobile" required>
                    <label>your address : </label>
         <input type="text" placeholder="address" name="address" required>
                    <label>email : </label>
         <input type="text" placeholder="email" name="mail" required>
         <button type="submit">Order</button>
           </div>
   </form>
           <div class="msg">{{ msg }}</div><br>
           <center><a href="/main" target="_blank">Continue Shooping</a></center>
           </div>
</body>
</html>
```

Orders.html

```
<!DOCTYPE html>
<html>
<head>
           <title>Index Page</title>
           <style>
```

```
        table, th, td {
         border:1px solid black;
        padding: 8px;
        }
        tr:nth-child(even){background-color: #f2f2f2}
        </style>
</head>
<body>
        <h3>Orders</h3>

        <br>
        <table style="width:100%">
                <thead>
                        <th>Customer Name</th>
                        <th>Product</th>
                        <th>e-mail</th>
                        <th>Address</th>
                        <th>Quantity</th>
                        <th>Mobile number</th>
                </thead>
                {% for row in data %}
                <tbody>
                        <td>{{row['NAME']}}</td>
                        <td>{{row['PRODUCT']}}</td>
                        <td>{{row['MAIL']}}</td>
                        <td>{{row['ADDRESS']}}</td>
                        <td>{{row['QUANTITY']}}</td>
                        <td>{{row['mobile']}}</td>
                </tbody>
```

```
                    {% endfor%}
            </table>
</body>
</html>
```

Phone.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Order here</title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-image: url("https://www.logicinbound.com/wp-content/uploads/2018/01/shutterstock_779835055-1024x737.jpg");
            width900vw;
            height:90vh;


}
button {
      background-color: #4CAF50;
      width: 100%;
       color: orange;
       padding: 15px;
       margin: 10px 0px;
       border: none;
       cursor: pointer;
        }
 form {
```

```
        border: 3px solid #f1f1f1;
            width: 50%;
            margin: 0 auto;
            border-radius: 30px;

    }
  input[type=text], input[type=password]{
        width: 100%;
        margin: 8px 0;
        padding: 12px 20px;
        display: inline-block;
        border: 2px solid green;
        box-sizing: border-box;
    }
  button:hover {
        opacity: 0.7;
    }




  .container {
        padding: 25px;
        background-color: lightblue;
            border-radius: 30px;
 }
 .div1
 {

            float: left;
```

```
            width: 40%;
            padding: 25px;
        background-color: lightblue;
            border-radius: 30px;

}
.div2
{
            width: 50%;
            float: right;
}

a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

</style>
</head>
<body>
<div class="div1">
<div class="pic">
<center><h1 style="background-color:orange;">Phone<h1></center>
<br>
```

```
               <img
src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/mobile.jpg"
width="300" height="300" alt="your product">

</div>

</div>

<div class="div2">


   <form action="/phone" method="post">

      <div class="container">

        <label>Username : </label>

        <input type="text" placeholder="Enter Username" name="name" required>

        <label>quantity : </label>

        <input type="text" placeholder="number of quantity " name="quantity" required>

                <label>mobile : </label>

        <input type="text" placeholder="mobile number " name="mobile" required>

                <label>your address : </label>

        <input type="text" placeholder="address" name="address" required>

                <label>email : </label>

        <input type="text" placeholder="email" name="mail" required>

        <button type="submit">Order</button>

          </div>

   </form>

          <div class="msg">{{ msg }}</div><br>

          <center><a href="/main" target="_blank">Continue Shooping</a></center>

          </div>

</body>

</html>
```

Register.html

```
<!DOCTYPE html>
```

```
<html lang="en">


<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Smart Fashion register</title>




   <!-- bootstrap css cdn -->
   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
      integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.css">
   <!-- css stylesheet -->
   <link rel="stylesheet" href="/static/style.css">
   <!-- font styles cdn -->
   <link rel="preconnect" href="https://fonts.gstatic.com">
   <link href="https://fonts.googleapis.com/css2?family=Alegreya&display=swap" rel="stylesheet">
   <link href="https://fonts.googleapis.com/css2?family=Alegreya:wght@600&display=swap"
rel="stylesheet">
</head>


<body>
   <!-- bootstrap navbar -->
   <!-- navbar ends -->
   <!-- Login form -->
```

```
  <div class="login text-center mt-5">


    <form action="/signup" method="post">
      <h2 class="form-text-h2"> Register Form </h2>
      <div class="msg">{{ msg }}</div>
      <input type="text" class="form-control" name="username" placeholder="Enter Your Username" id="username"
        required>
      <input type="email" class="form-control" name="email" placeholder="Enter Your Email ID" id="email" required>
      <input type="password" class="form-control" name="password" placeholder="Enter Your Password" id="password"
        required>
      <button type="submit" id="button" class="btn btn-primary"> Register </button>
      <div class="note mt-3 text-center">
        <!--Register form -->
        <p> already have an account ? please login <a href="/">login! </a> </p>


      </div>
    </form>


  </div>




</body>


</html>
```

Saree.html

```
<!DOCTYPE html>
```

```
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Order here</title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-image: url("https://www.logicinbound.com/wp-content/uploads/2018/01/shutterstock_779835055-1024x737.jpg");
            width900vw;
            height:90vh;

}
button {
      background-color: #4CAF50;
      width: 100%;
       color: orange;
       padding: 15px;
       margin: 10px 0px;
       border: none;
       cursor: pointer;
        }
 form {
       border: 3px solid #f1f1f1;
            width: 50%;
            margin: 0 auto;
            border-radius: 30px;

   }
```

```
input[type=text], input[type=password]{
    width: 100%;
    margin: 8px 0;
    padding: 12px 20px;
    display: inline-block;
    border: 2px solid green;
    box-sizing: border-box;
  }
button:hover {
    opacity: 0.7;
  }




.container {
    padding: 25px;
    background-color: lightblue;
        border-radius: 30px;
}
.div1
{

        float: left;
        width: 40%;
        padding: 25px;
    background-color: lightblue;
        border-radius: 30px;

}
```

```
.div2
{
        width: 50%;
        float: right;
}

a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

</style>
</head>
<body>
<div class="div1">
<div class="pic">
<center><h1 style="background-color:orange;">Saree<h1></center>
<br>
                <img
src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/saree.jpeg"     width="300"
height="300" alt="your product">
</div>
</div>
<div class="div2">
```

```
  <form action="/saree" method="post">
     <div class="container">
       <label>Username : </label>
       <input type="text" placeholder="Enter Username" name="name" required>
       <label>quantity : </label>
       <input type="text" placeholder="number of quantity " name="quantity" required>
                 <label>mobile : </label>
       <input type="text" placeholder="mobile number " name="mobile" required>
                 <label>your address : </label>
       <input type="text" placeholder="address" name="address" required>
                 <label>email : </label>
       <input type="text" placeholder="email" name="mail" required>
       <button type="submit">Order</button>
         </div>
  </form>
         <div class="msg">{{ msg }}</div><br>
         <center><a href="/main" target="_blank">Continue Shooping</a></center>
         </div>
</body>
</html>
```

Shirt.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Order here</title>
<style>
Body {
```

```
  font-family: Calibri, Helvetica, sans-serif;

  background-image: url("https://www.logicinbound.com/wp-
content/uploads/2018/01/shutterstock_779835055-1024x737.jpg");

            width900vw;

            height:90vh;


}

button {

      background-color: #4CAF50;

      width: 100%;

       color: orange;

       padding: 15px;

       margin: 10px 0px;

       border: none;

       cursor: pointer;

        }

 form {

       border: 3px solid #f1f1f1;

            width: 50%;

            margin: 0 auto;

            border-radius: 30px;


   }

 input[type=text], input[type=password]{

       width: 100%;

       margin: 8px 0;

       padding: 12px 20px;

       display: inline-block;

       border: 2px solid green;
```

```
        box-sizing: border-box;
    }
 button:hover {
        opacity: 0.7;
    }




 .container {
        padding: 25px;
        background-color: lightblue;
            border-radius: 30px;
}
.div1
{

            float: left;
            width: 40%;
            padding: 25px;
        background-color: lightblue;
            border-radius: 30px;

}
.div2
{
            width: 50%;
            float: right;
}
```

```
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

</style>
</head>
<body>
<div class="div1">
<div class="pic">
<center><h1 style="background-color:orange;">Shirt<h1></center>
<br>
                <img
src="https://images.pexels.com/photos/3649765/pexels-photo-
3649765.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500"  width="300" height="300" alt="your
product">
</div>
</div>
<div class="div2">

   <form action="/shirt" method="post">
       <div class="container">
         <label>Username : </label>
         <input type="text" placeholder="Enter Username" name="name" required>
         <label>quantity : </label>
         <input type="text" placeholder="number of quantity " name="quantity" required>
```

```
                <label>mobile : </label>
        <input type="text" placeholder="mobile number " name="mobile" required>
                <label>your address : </label>
        <input type="text" placeholder="address" name="address" required>
                <label>email : </label>
        <input type="text" placeholder="email" name="mail" required>
        <button type="submit">Order</button>
          </div>
   </form>
          <div class="msg">{{ msg }}</div><br>
          <center><a href="/main" target="_blank">Continue Shooping</a></center>
          </div>
</body>
</html>
```

Shirt.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Order here</title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-image: url("https://www.logicinbound.com/wp-content/uploads/2018/01/shutterstock_779835055-1024x737.jpg");
          width900vw;
          height:90vh;

}
```

```
button {
      background-color: #4CAF50;
      width: 100%;
       color: orange;
       padding: 15px;
       margin: 10px 0px;
       border: none;
       cursor: pointer;
        }
 form {
       border: 3px solid #f1f1f1;
             width: 50%;
             margin: 0 auto;
             border-radius: 30px;


   }
 input[type=text], input[type=password]{
       width: 100%;
       margin: 8px 0;
       padding: 12px 20px;
       display: inline-block;
       border: 2px solid green;
       box-sizing: border-box;
   }
 button:hover {
       opacity: 0.7;
   }
```

```
 .container {
        padding: 25px;
        background-color: lightblue;
              border-radius: 30px;
}
.div1
{

              float: left;
              width: 40%;
              padding: 25px;
        background-color: lightblue;
              border-radius: 30px;

}
.div2
{
              width: 50%;
              float: right;
}

a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
```

```
}


</style>

</head>

<body>

<div class="div1">

<div class="pic">

<center><h1 style="background-color:orange;">Shirt<h1></center>

<br>

                <img
src="https://images.pexels.com/photos/3649765/pexels-photo-
3649765.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500"  width="300" height="300" alt="your
product">

</div>

</div>

<div class="div2">


   <form action="/shirt" method="post">

      <div class="container">

         <label>Username : </label>

         <input type="text" placeholder="Enter Username" name="name" required>

         <label>quantity : </label>

         <input type="text" placeholder="number of quantity " name="quantity" required>

                  <label>mobile : </label>

         <input type="text" placeholder="mobile number " name="mobile" required>

                  <label>your address : </label>

         <input type="text" placeholder="address" name="address" required>

                  <label>email : </label>

         <input type="text" placeholder="email" name="mail" required>

         <button type="submit">Order</button>
```

```
            </div>
    </form>
            <div class="msg">{{ msg }}</div><br>
            <center><a href="/main" target="_blank">Continue Shooping</a></center>
            </div>
</body>
</html>
```

Shoes.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Order here</title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-image: url("https://www.logicinbound.com/wp-content/uploads/2018/01/shutterstock_779835055-1024x737.jpg");
            width900vw;
            height:90vh;


}
button {
      background-color: #4CAF50;
      width: 100%;
       color: orange;
       padding: 15px;
       margin: 10px 0px;
       border: none;
```

```
        cursor: pointer;
         }
  form {
        border: 3px solid #f1f1f1;
              width: 50%;
              margin: 0 auto;
              border-radius: 30px;

    }
  input[type=text], input[type=password]{
        width: 100%;
        margin: 8px 0;
        padding: 12px 20px;
        display: inline-block;
        border: 2px solid green;
        box-sizing: border-box;
    }
  button:hover {
        opacity: 0.7;
    }




  .container {
        padding: 25px;
        background-color: lightblue;
              border-radius: 30px;
 }
 .div1
```

```
{

        float: left;
        width: 40%;
        padding: 25px;
    background-color: lightblue;
        border-radius: 30px;

}
.div2
{
        width: 50%;
        float: right;
}

a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

</style>
</head>
<body>
<div class="div1">
<div class="pic">
```

```
<center><h1 style="background-color:orange;">Shoes<h1></center>

<br>

               <img
src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/shoos.jpg" width="300"
height="300" alt="your product">

</div>

</div>

<div class="div2">


  <form action="/shoes" method="post">

    <div class="container">

      <label>Username : </label>

      <input type="text" placeholder="Enter Username" name="name" required>

      <label>quantity : </label>

      <input type="text" placeholder="number of quantity " name="quantity" required>

                <label>mobile : </label>

      <input type="text" placeholder="mobile number " name="mobile" required>

                <label>your address : </label>

      <input type="text" placeholder="address" name="address" required>

                <label>email : </label>

      <input type="text" placeholder="email" name="mail" required>

      <button type="submit">Order</button>

        </div>

  </form>

        <div class="msg">{{ msg }}</div><br>

        <center><a href="/main" target="_blank">Continue Shooping</a></center>

        </div>

</body>

</html>
```

Tv.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Order here</title>
<style>
Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-image: url("https://neilpatel.com/wp-content/uploads/2015/04/ecommerce.jpg");
            width900vw;
            height:90vh;


}
button {
      background-color: #4CAF50;
      width: 100%;
       color: orange;
       padding: 15px;
       margin: 10px 0px;
       border: none;
       cursor: pointer;
        }
 form {
       border: 3px solid #f1f1f1;
            width: 50%;
            margin: 0 auto;
            border-radius: 30px;
```

```
  }
input[type=text], input[type=password]{
      width: 100%;
      margin: 8px 0;
      padding: 12px 20px;
      display: inline-block;
      border: 2px solid green;
      box-sizing: border-box;
  }
button:hover {
      opacity: 0.7;
  }




.container {
      padding: 25px;
      background-color: lightblue;
            border-radius: 30px;
}
.div1
{

            float: left;
            width: 40%;
            padding: 25px;
      background-color: lightblue;
            border-radius: 30px;
```

```
}
.div2
{
        width: 50%;
        float: right;
}

a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

</style>
</head>
<body>
<div class="div1">
<div class="pic">
<center><h1 style="background-color:orange;">Television<h1></center>
<br>
                <img
src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/tv.jpeg"    width="300"
height="300" alt="your product">
</div>
</div>
```

```
<div class="div2">


   <form action="/tv" method="post">
       <div class="container">
         <label>Username : </label>
         <input type="text" placeholder="Enter Username" name="name" required>
         <label>quantity : </label>
         <input type="text" placeholder="number of quantity " name="quantity" required>
                    <label>mobile : </label>
         <input type="text" placeholder="mobile number " name="mobile" required>
                    <label>your address : </label>
         <input type="text" placeholder="address" name="address" required>
                    <label>email : </label>
         <input type="text" placeholder="email" name="mail" required>
         <button type="submit">Order</button>
           </div>
   </form>
           <div class="msg">{{ msg }}</div><br>
           <center><a href="/main" target="_blank">Continue Shooping</a></center>
           </div>
</body>
</html>
```

Users.html

```
<!DOCTYPE html>
<html>
<head>
           <title>Index Page</title>
           <style>
```

```
        table, th, td {
         border:1px solid black;
        padding: 8px;
        }
        tr:nth-child(even){background-color: #f2f2f2}
        </style>
</head>
<body>
        <h3>Profiles</h3>

        <br>
        <table style="width:100%">
                <thead>
                        <th>name</th>
                        <th>email</th>
                        <th>pass</th>
                </thead>
                {% for row in data %}
                <tbody>
                        <td>{{row['USERNAME']}}</td>
                        <td>{{row['EMAIL']}}</td>
                        <td>{{row['PASSWORD']}}</td>

                </tbody>
                {% endfor%}
        </table>
</body>
</html>
```

Watch.html

```
<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Order here</title>

<style>

Body {

  font-family: Calibri, Helvetica, sans-serif;

  background-image: url("https://www.logicinbound.com/wp-content/uploads/2018/01/shutterstock_779835055-1024x737.jpg");

            width900vw;

            height:90vh;


}

button {

      background-color: #4CAF50;

      width: 100%;

       color: orange;

       padding: 15px;

       margin: 10px 0px;

       border: none;

       cursor: pointer;

        }

 form {

       border: 3px solid #f1f1f1;

            width: 50%;

            margin: 0 auto;

            border-radius: 30px;
```

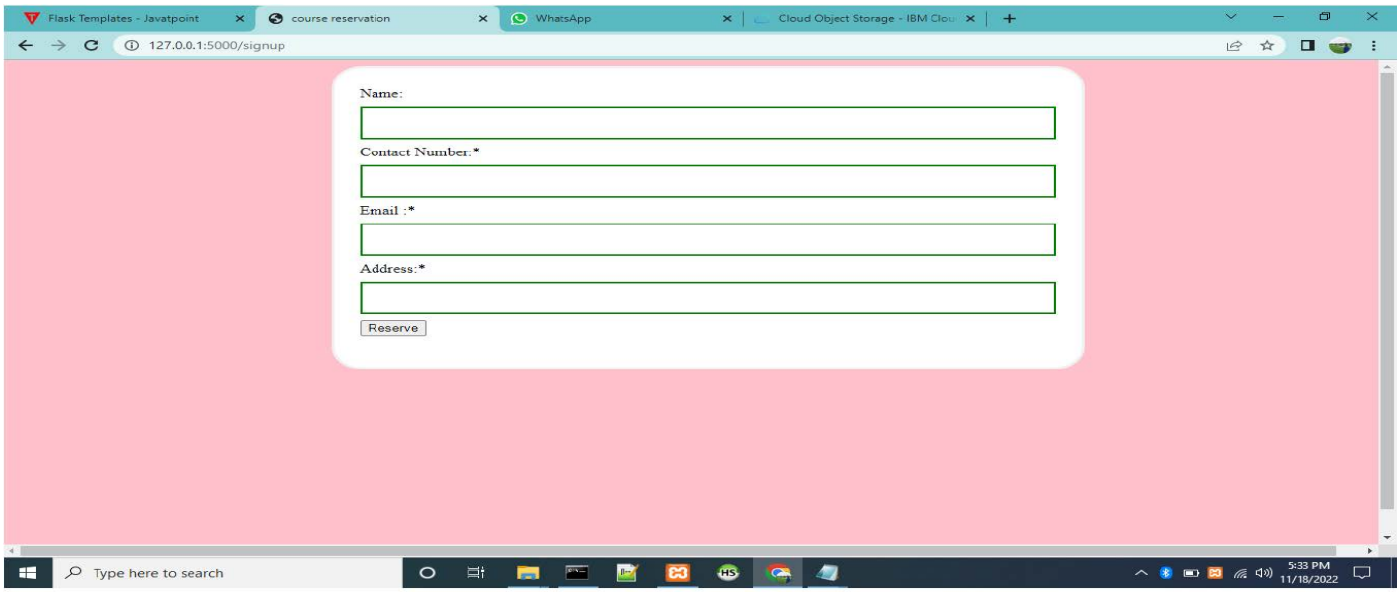
```
  }
input[type=text], input[type=password]{
      width: 100%;
      margin: 8px 0;
      padding: 12px 20px;
      display: inline-block;
      border: 2px solid green;
      box-sizing: border-box;
  }
button:hover {
      opacity: 0.7;
  }




.container {
      padding: 25px;
      background-color: lightblue;
            border-radius: 30px;
}
.div1
{

            float: left;
            width: 40%;
            padding: 25px;
      background-color: lightblue;
            border-radius: 30px;
```

```
}
.div2
{
        width: 50%;
        float: right;
}

a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

</style>
</head>
<body>
<div class="div1">
<div class="pic">
<center><h1 style="background-color:orange;">Watch<h1></center>
<br>
                <img
src="https://fashion-bucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/watch.jpeg"    width="300"
height="300" alt="your product">
</div>
</div>
```

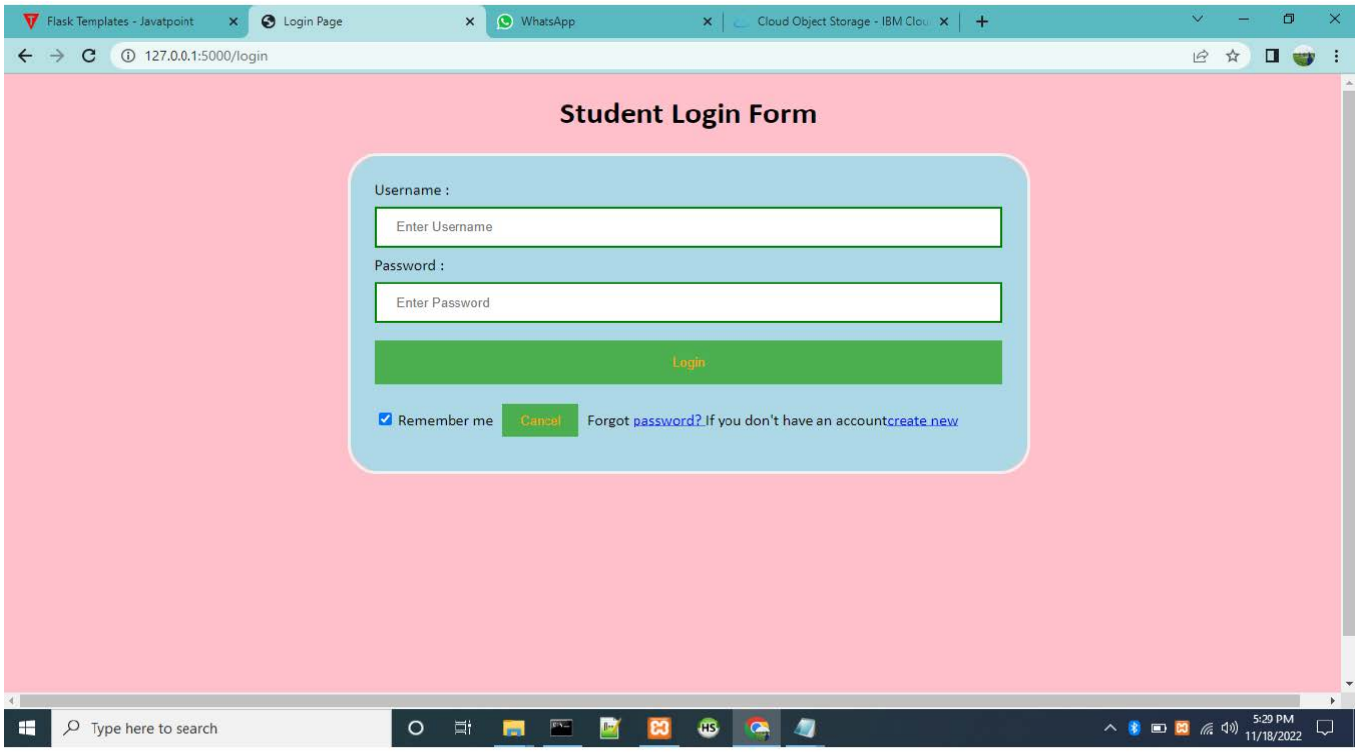
```
<div class="div2">


   <form action="/watch" method="post">
       <div class="container">
         <label>Username : </label>
         <input type="text" placeholder="Enter Username" name="name" required>
         <label>quantity : </label>
         <input type="text" placeholder="number of quantity " name="quantity" required>
                     <label>mobile : </label>
         <input type="text" placeholder="mobile number " name="mobile" required>
                     <label>your address : </label>
         <input type="text" placeholder="address" name="address" required>
                     <label>email : </label>
         <input type="text" placeholder="email" name="mail" required>
         <button type="submit">Order</button>
           </div>
   </form>
           <div class="msg">{{ msg }}</div><br>
           <center><a href="/main" target="_blank">Continue Shooping</a></center>
           </div>
</body>
</html>
```

SCREENSHOTS

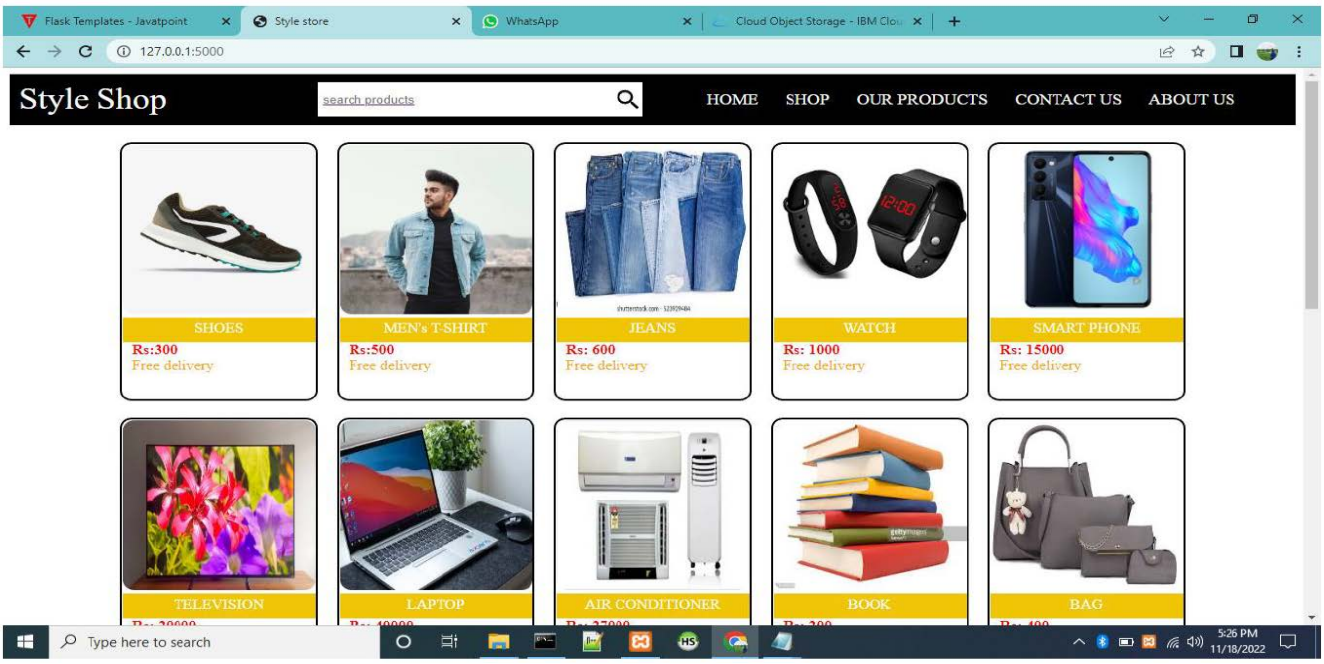
Register page



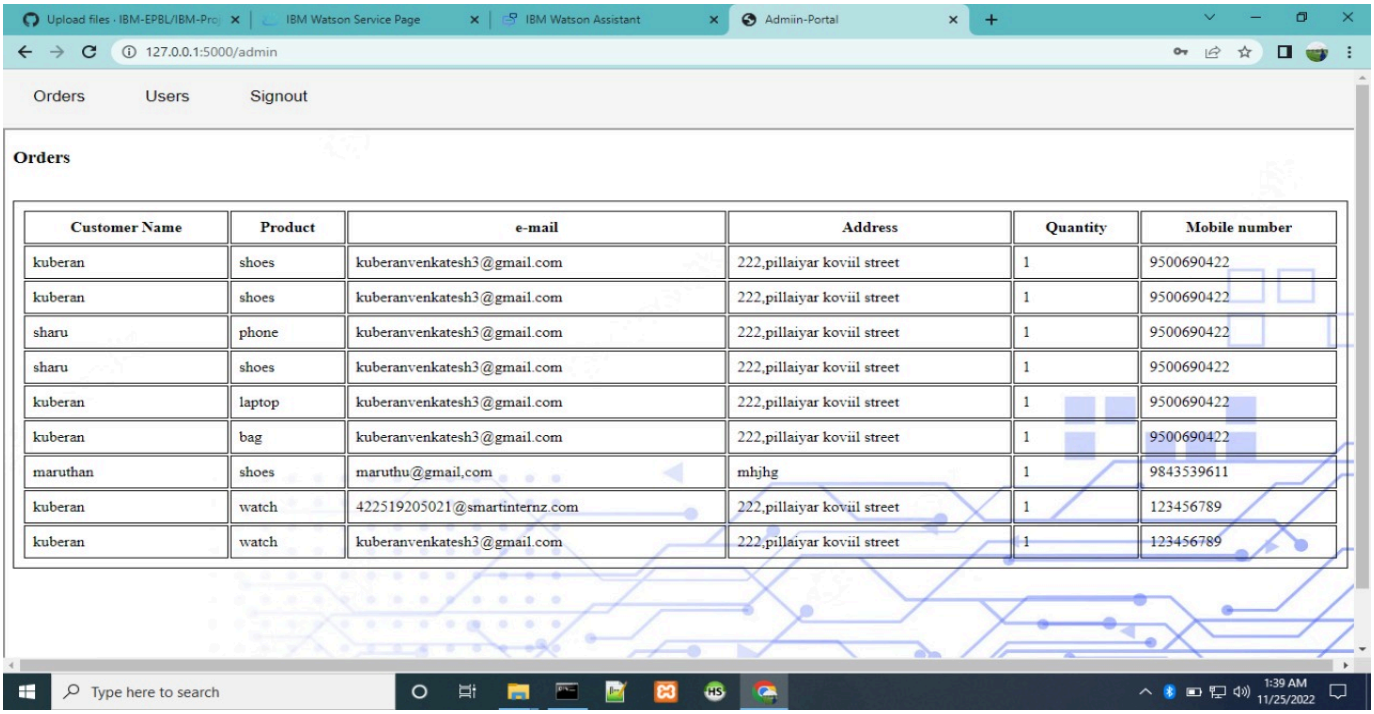
Login Page



Home page

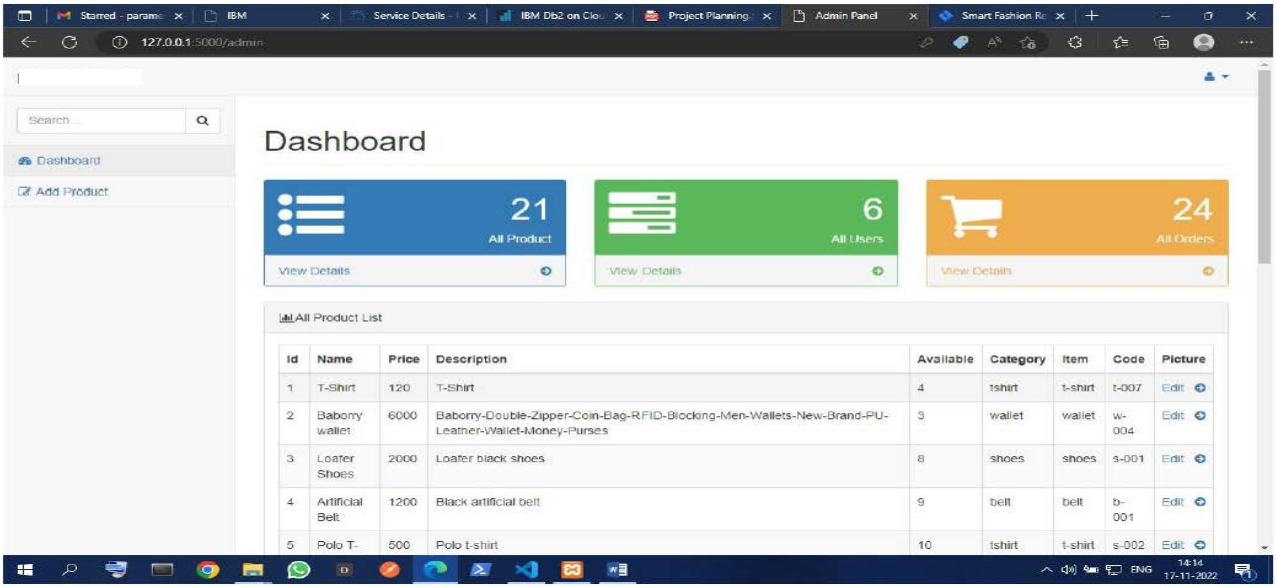


Order Page

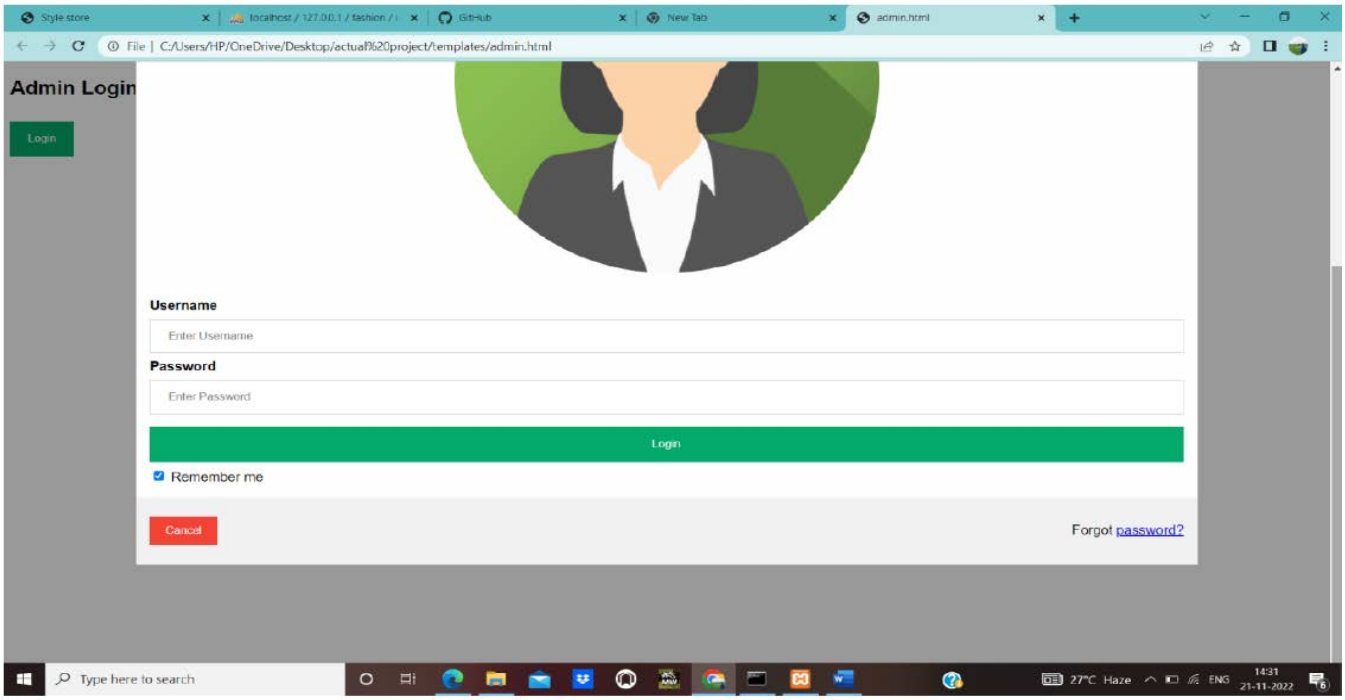


Customer Name	Product	e-mail	Address	Quantity	Mobile number
kuberan	shoes	kuberanvenkatesh3@gmail.com	222,pillaiyar koviil street	1	9500690422
kuberan	shoes	kuberanvenkatesh3@gmail.com	222,pillaiyar koviil street	1	9500690422
sharu	phone	kuberanvenkatesh3@gmail.com	222,pillaiyar koviil street	1	9500690422
sharu	shoes	kuberanvenkatesh3@gmail.com	222,pillaiyar koviil street	1	9500690422
kuberan	laptop	kuberanvenkatesh3@gmail.com	222,pillaiyar koviil street	1	9500690422
kuberan	bag	kuberanvenkatesh3@gmail.com	222,pillaiyar koviil street	1	9500690422
maruthan	shoes	maruthu@gmail,com	mhjhg	1	9843539611
kuberan	watch	422519205021@smartinternz.com	222,pillaiyar koviil street	1	123456789
kuberan	watch	kuberanvenkatesh3@gmail.com	222,pillaiyar koviil street	1	123456789

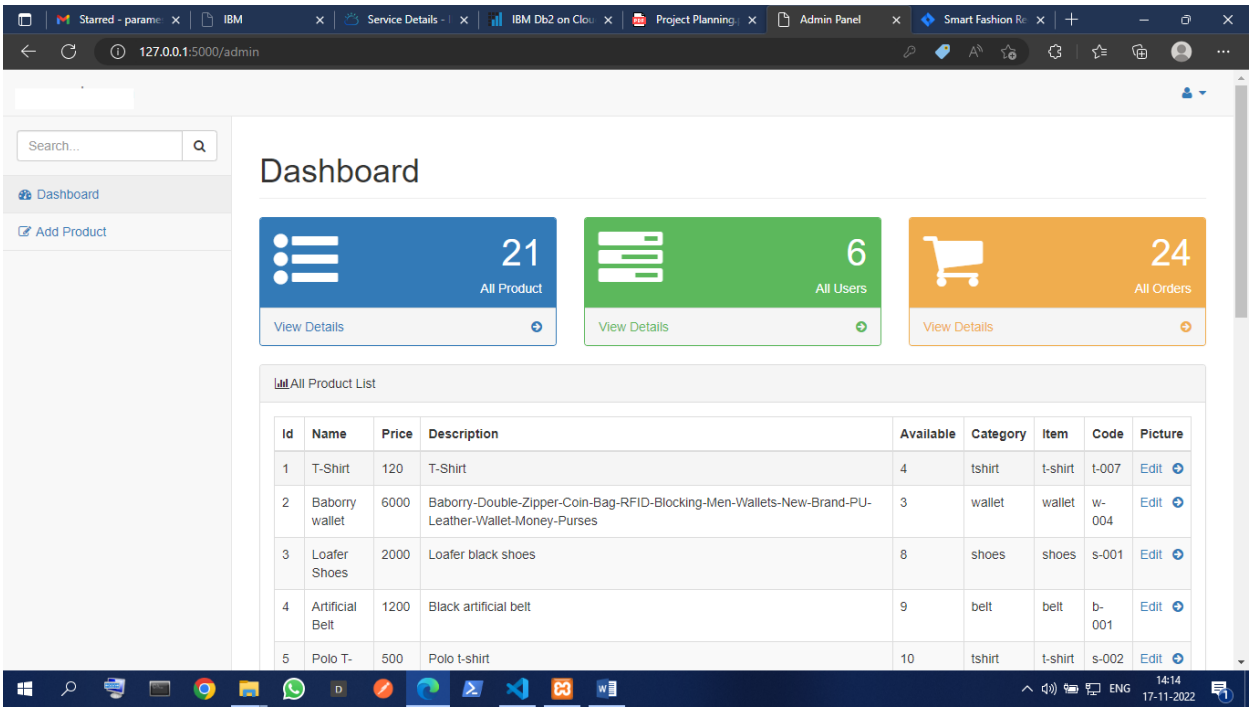
Profile Page



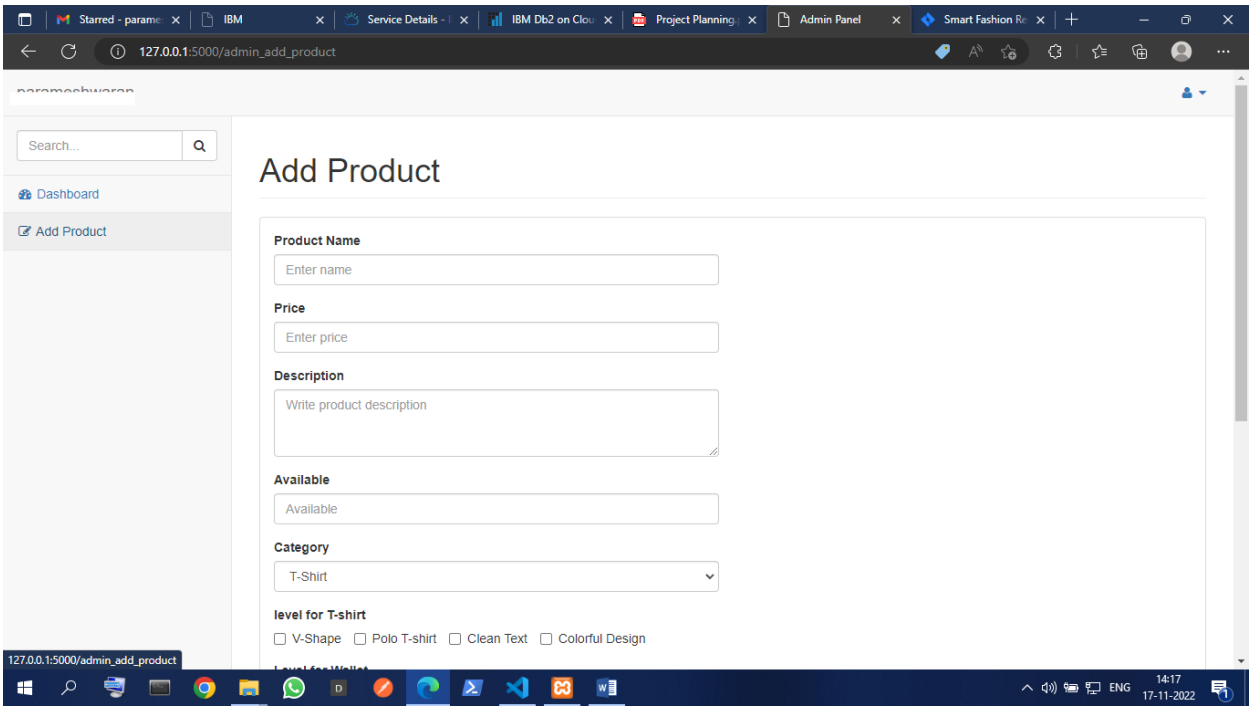
Admin Login Page



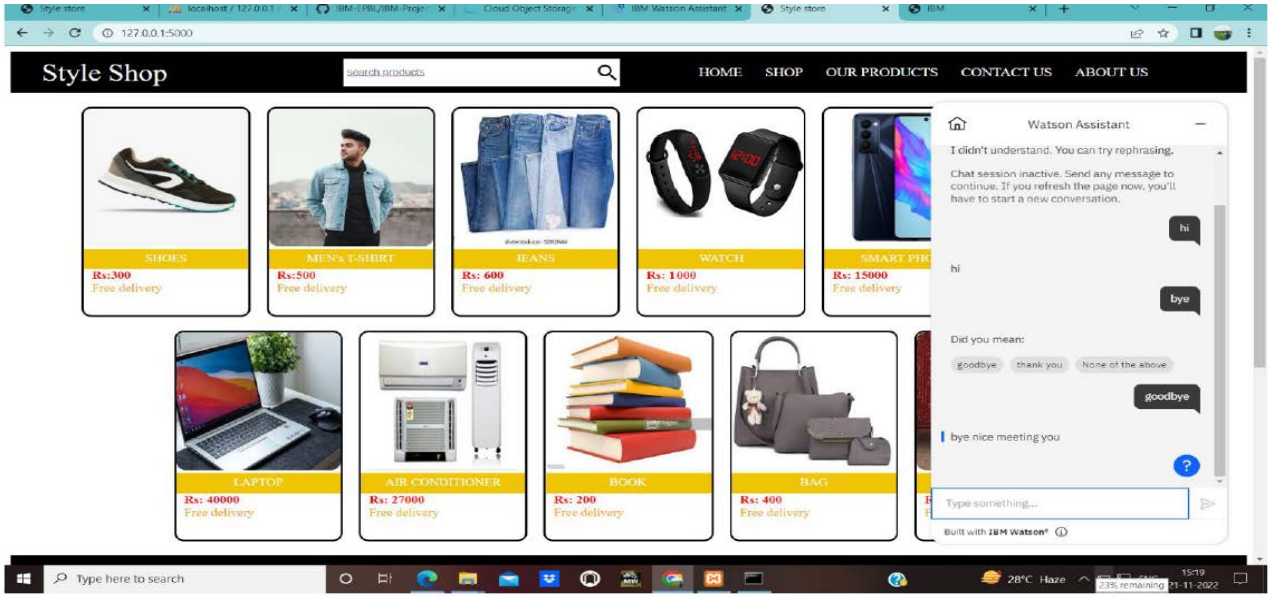
Admin Dashboard Page

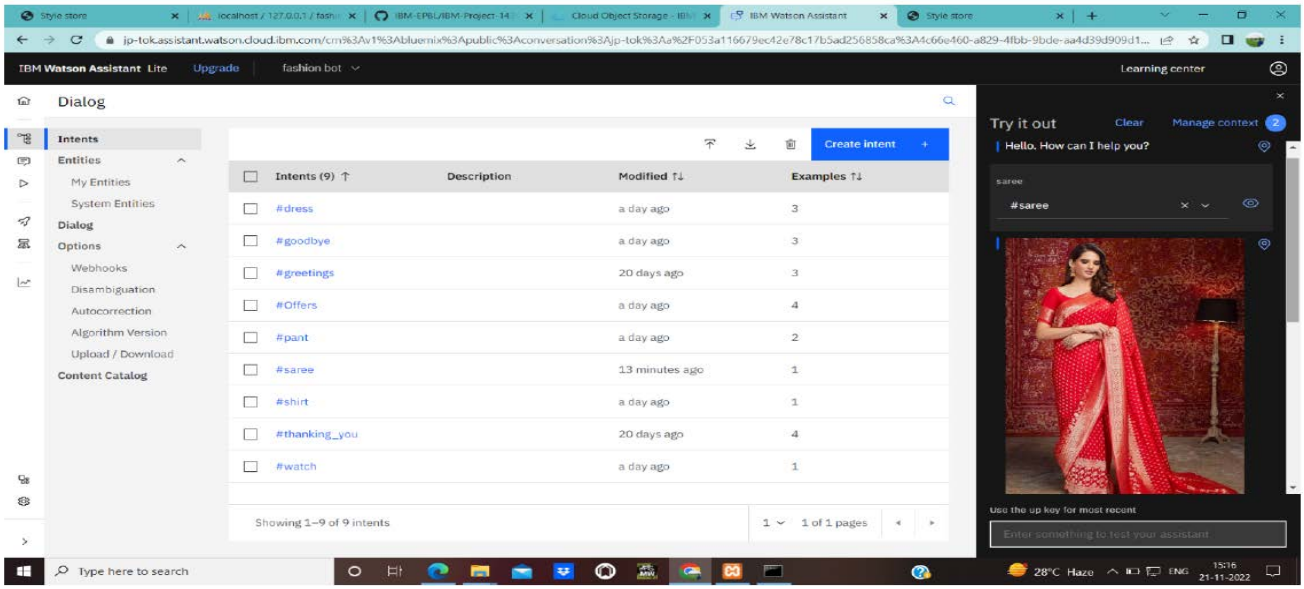


Add Product Page



Integrate Chat Bot





13.2 GITHUB & PROJECT DEMO LINK

- Our GitHub Repository Direct Link
  https://github.com/IBM-EPBL/IBM-Project-14341-1664357686.git
- Project Demonstration Video Direct Link
- https://youtu.be/pL0QvgHXUb8