

PROJECT REPORT

SMART FASHION RECOMMENDER APPLICATION

CLOUD APPLICATION

TEAM ID: PNT2022MID29786

Submitted by

HARISH.S	(610519104035)
JAVEED HUSSAIN.N	(610519104040)
MADHAN KUMAR.R	(610519104057)
HARI HARAN. C. S	(610519104036)

CHAPTER NO

TITLE

1. **INTRODUCTION**
 - 1.1 Project Overview
 - 1.2 Purpose
2. **LITERATURE SURVEY**
 - 2.1 Existing problem
 - 2.2 References
 - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
 - 3.1 Empathy Map Canvas
 - 3.2 Ideation & Brainstorming
 - 3.3 Proposed Solution
 - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
 - 4.1 Functional requirement
 - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
 - 5.1 Data Flow Diagrams
 - 5.2 Solution & Technical Architecture
 - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
 - 6.1 Sprint Planning & Estimation
 - 6.2 Sprint Delivery Schedule
 - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING**

(Explain the features added in the project along with code)

 - 7.1 Feature 1
 - 7.2 Feature 2
 - 7.3 Database Schema (if Applicable)

8.

TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9.

RESULTS

9.1 Performance Metrics

10.

ADVANTAGES & DISADVANTAGES

11.

CONCLUSION

12.

FUTURE SCOPE

1.INTRODUCTION

1.1Project Overview:

In recent years, the textile and fashion industries have witnessed an enormous amount of growth in fast fashion. On e-commerce platforms, where numerous choices are available, an efficient recommendation system is required to sort, order, and efficiently convey relevant product content or information to users. Image-based fashion recommendation systems (FRSs) have attracted a huge amount of attention from fast fashion retailers as they provide a personalized shopping experience to consumers. With the technological advancements, this branch of artificial intelligence exhibits a tremendous amount of potential in image processing, parsing, classification, and segmentation.

Present apparel e-commerce system that encourages online shopping has major issues to deal with catalog based online shopping. As there is a lack of customized services, the users may face difficulties to find discrimination over different types of retailers available on electronic product catalogs, they may also be confused with complex navigations that redirect to other pages based on their selection. This drawback can be overwhelmed by following suggestions on categories that they have chosen or from the products that they have already viewed. Multiple number of online marketing companies around world-wide has been practicing the naive method for apparel marketing website. This paper aims to simulate this recommendation system on real world data set taken from the marketing giant, Amazon's Product Advertising API, in a policy compliant manner by following the procedure in three steps Analyzing the data to select the pivot for the recommendation system, Data preprocessing to remove invalid sections and to implement and find appropriate choices among the techniques like Bag of Words (BOW) and TF-IDF for better recommendation

1.2 Purpose:

There is currently no existing system that is capable of recommending clothes based on the occasion. Different occasions call for different clothing. Moreover, a lot of fashion is based on the color combinations of outfits. A person with no or little fashion sense will have a hard time to decide on clothes that leave a lasting impression. The proposed Fashion Recommendation System is intended to be used by individual users in order to store images of the clothes that they own in what is called a digital wardrobe and also to get recommendations by the system on what clothes to wear for a given occasion. The main aim of the project is to recommend the most appropriate clothes for a given occasion based on the clothes existing in the user's wardrobe to relieve the user of the burden of making decisions about what clothing to wear. Such a system should be capable of helping someone who has no fashion sense to wear clothes that leave a good impression on others. The system should be such that it is easily accessible and easy to take advantage of the various features that it provides. This feature can also be used by the recommendation algorithm to recommend the clothes. Another feature is the classification of the type and color of the clothing that is uploaded by the user. The system should be capable of handling the 4 basic clothing types: Shirt, T-Shirt, Pants and Shoes.

2.LITERATURE SURVEY

The work of this project is based on combining two deep learning models to detect the type and color of the clothing in the given image. The recommendation algorithm however is written by us. Hence, it is safe to say that no existing system has been proposed but work has been done in detecting objects, types and colors of clothes by using public datasets and applying machine learning techniques. With the quick rise in living standards, people's shopping passion grew, and their desire for clothing grew as well. A growing number of people are interested in fashion these days. However, when confronted with a large number of garments, consumers are forced to try them on multiple times, which takes time and energy. As a result of the suggested Fashion Recommendation System, a variety of online fashion businesses and web applications allow buyers to view collages of stylish items that look nice together. Clients and sellers benefit from such recommendations. On the one hand, customers can make smarter shopping decisions and discover new articles of clothes that complement one other. Complex outfit recommendations, on the other hand, assist vendors in selling more products, which has an impact on their business. Fashioned is made up of two parts: a feature network for extracting features and a matching network for calculating compatibility. A deep convolutional network is used to achieve the former. For the latter, a multi-layer completely connected network.

2.1 Existing problem:

There is an existing problem in the fashion industry that needs to be addressed. A lot of designers are using unethical practices to make money. This includes using cheap materials and cutting corners to increase profits. This is not fair to the consumers, and it needs to stop. We need better, more ethical fashion brands out there, and we need them fast.

There is an existing problem with fashion recommender systems that dictates that they are often ineffective and unreliable. Fashion recommendation systems rely on data from users to create personalized recommendations, but this data can be unreliable. This means that the system may recommend items that are not in line with the user's interests or style. Additionally, the system may not be able to recommend items that the user is actively looking for. This can lead to frustrating experiences when trying to find clothing that fits well or finding new styles to try.

The current systems rely on user feedback and ratings to generate recommendations, but this method is flawed because it does not take into account the user's own style. This means that the recommendations are ineffective for people who prefer different styles of clothing than the average person.

2.2 References:

Bayshore Pravin Bluer, Pratik Sha Tulshiram Bandos, Mohali Shivar Amaka and Savita Praha Lodi ale Smart Chakraborty,

Saiful Saiful Hoque Namur Rahman Jermaine Chandra Biswas deepayan Barchan[1] Mohamed Elechi,

Anis Mezghan, Mariam Cheatham, Manji Keralan “Clothing Classification using Deep CNN Architecture based on Transfer Learning” ,2021 DOI:10.1007/978-3-030-49336-3_24 [2] Saurabh Gupta,

Siddhartha Agarwal, Apoorva Dave. “Apparel Classifier and Recommender using Deep Learning.” (2015). [3] Bussard, Lukas,

Matthias Danton, Christian Leister, Christian Wenger, Till Quack and Luc Van Gool. “Apparel Classification with Style.” ACCV (2012). [4] Kievsky, Alex, Ilya Subsieve and Geoffrey E.

“ImageNet classification with deep convolutional neural networks.” Communications of the ACM 60 (2012): 84 - 90. [5] Congeeing Guan, Sheng Feng Qin, Yang Long, (2019) \"Apparel-based deep learning system design for apparel style recommendation\", International Journal of Clothing Science and Technology. [6] Stephen Marshland

Machine Learning – An Algorithmic Perspective, Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 20 Illa pavan Kumar, Swathi Sambangi

Present apparel e-commerce system that encourage online shopping ,has major issues to deal with catalog based online shopping. As there is a lack of customized services, the users may face difficulties to find discrimination over different types of retailers available on electronic product catalogs, they may also be confused with complex navigations that redirect to other pages based on their selection. This drawback can be overwhelmed by following suggestions on categories that they have chosen or from the products that they have already viewed. Multiple number of online marketing companies around world-wide ,has been practicing the naive method for apparel marketing website

2.3 Problem Statement Definition:

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

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

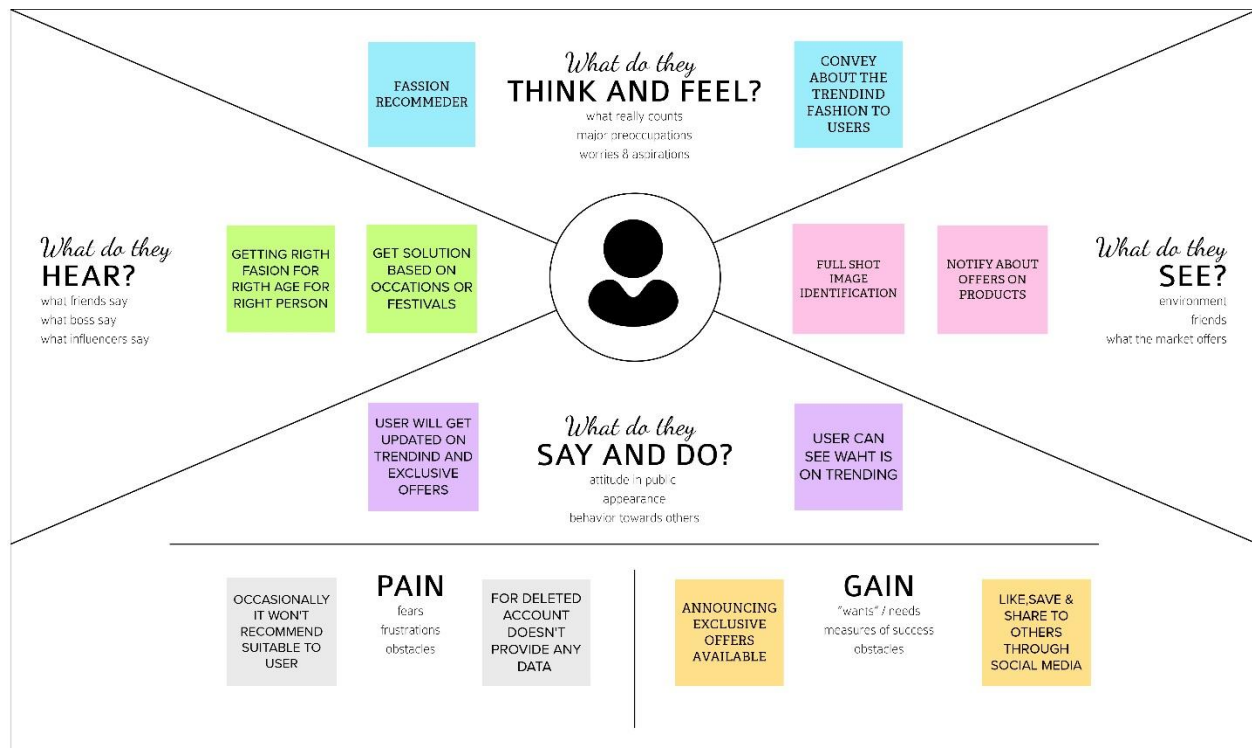
Using chatbot we can manage user's choices and orders. The chatbot can give recommendations to the users based on their interests. It can promote the best deals and offers on that day.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

An **empathy map** is a collaborative visualization used to articulate what we know about a particular type of user. It externalizes knowledge about users to

- 1) create a shared understanding of user needs, and
- 2) aid in decision making.



Brainstrom & Idea Listing and Grouping:

[illegible]

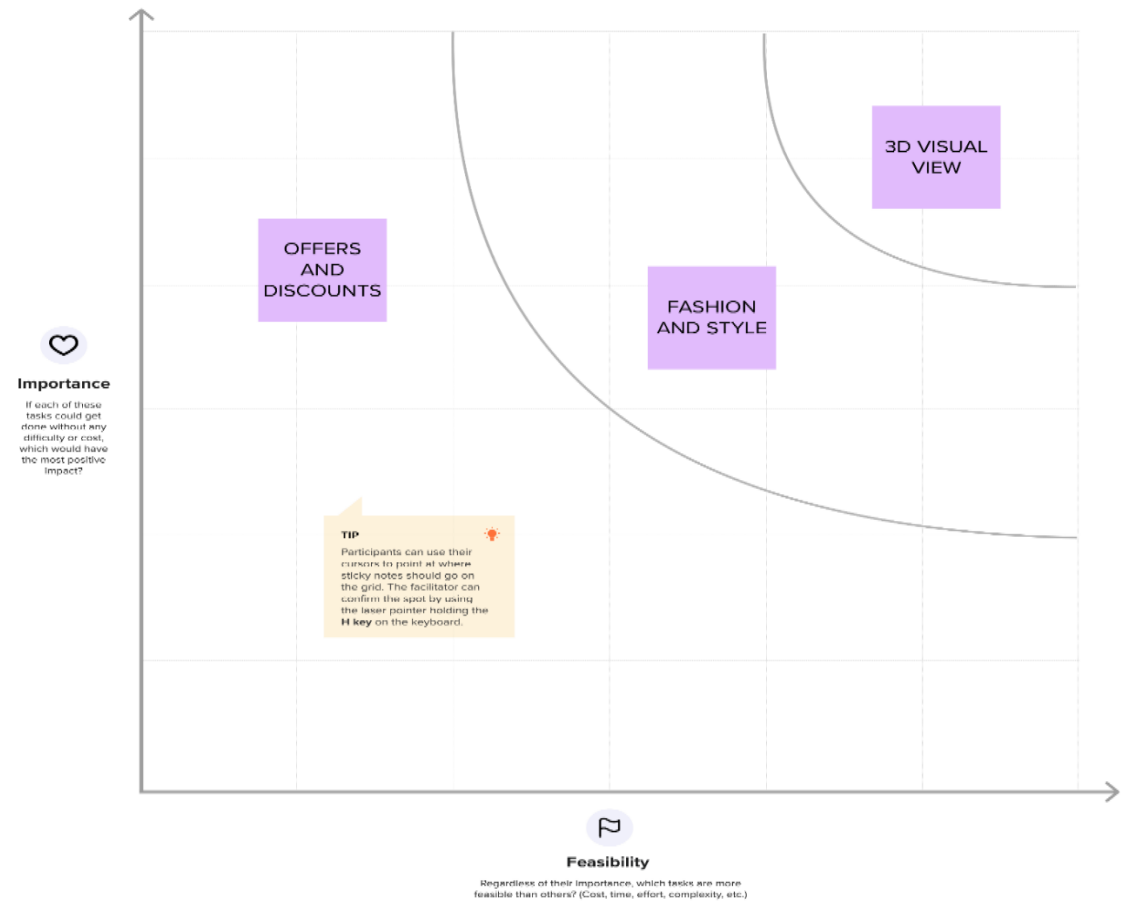
Idea Prioritization:

4

Prioritize

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

🕒 20 minutes



3.3 Proposed Solution:

1. Problem Statement (Problem to be solved):

Recommending trending fashion. Suggesting users matched fashions.

2. Idea / Solution description:

In this platform, there are numerous choices are available, an efficient recommendation system is required to sort, order, and efficiently convey relevant product content or information to users. The chatbot can give recommendations to the users based on their interests. It can promote the best deals and offers on that day.

3. Novelty / Uniqueness:

Instead of navigating to several screens for booking products online, the user can directly talk to Chatbot regarding the products. Get the recommendations based on information provided by the user. Using chatbot we can manage user's choices and orders.

4. Social Impact / Customer Satisfaction:

User can see what is on trending. User will get updated on trending and exclusive offers.

5. Business Model (Revenue Model):

Revenue can be generated by selling the products and they can get offers on any occasions or upcoming festivals. The revenue can also be obtained by maintaining customers services when they needed, for that services it can be charged for it.

6. Scalability of the Solution:

Through this, the user can get the right fashion to the right person.

3.4 Problem Solution fit:

Define CS, fit into CL	1. CUSTOMER SEGMENT(S) CS <ul style="list-style-type: none"> • Chatbot Shoppers who prefer the ease of contacting a chatbot to buy a product instead of search. • Website shoppers who browse online to buy products. • Discount seeking customers who often seek for discount in the product. 	6. CUSTOMER LIMITATIONS CL <small>EG. BUDGET, DEVICES</small> <ul style="list-style-type: none"> • Website speed and search function. • A quick finding of customer-related products. • Reviews and ratings can distract customers. • Customers cannot bargain. 	5. AVAILABLE SOLUTIONS AS <small>PROS & CONS</small> <ul style="list-style-type: none"> • FAQs to sort out queries of customers. • Availability of refund and return option. • Search for a specific product through the search bar. • Showing similar products of the selected product. • Category-wise product arrangement. 	Explore AS, differentiate
Focus on PR, tap into BE, understand RC	2. PROBLEMS / PAINS PR <small>• ITS FREQUENCY</small> <ul style="list-style-type: none"> • The presence of a chatbot can help in asking and resolving customer queries. • Customer review of a product. • Availability of sort and filter options to show products relevant to customers. • Showing a comparison between products. • Showing products that are most relevant to them. • Availability of refund and return policies. • Track Order option. 	9. PROBLEM ROOT / CAUSE RC <ul style="list-style-type: none"> • Network issue so that product could not load fast. • Long delivery. • Poor Tracking. • Product research and cross shopping. • During the festival, times may face network traffics and not-on-time delivery issue. • May be slight variations in dresses on delivery than they ordered. • Sometimes customer service is not available. 	7. BEHAVIOR BE <small>• ITS INTENSITY</small> <ul style="list-style-type: none"> • Cross-check and compare with other sites. • Purchase the product and write a review. • Dispose goods and services over the internet. • Monitoring and evaluation. • Identify the issues. • Searching for the best fashion and good product. • Giving the best deals from sellers to customers. 	Focus on PR, tap into BE, understand RC
Identify strong TR & EM	3. TRIGGERS TO ACT TR <ul style="list-style-type: none"> • Easy return and refund policy. • Time-consuming. • Social proof and novelty. • Through advertisements, the users are triggered. 	10. YOUR SOLUTION SL <ul style="list-style-type: none"> • A chatbot will recommend products related to the shoppers' search. • Get detailed information about the product and the product care. • Availability of review and rating options to give their feedback about the product bought. • Can compare products with various brands. • To provide a fashion recommender filter that clears their fashion queries. • And save the customer-related queries for future recommendation. 	8. CHANNELS of BEHAVIOR CH <div> ONLINE <ul style="list-style-type: none"> • Chat with chatbot. • Buy products. • Track and pay for the purchased products. </div> <div> OFFLINE <ul style="list-style-type: none"> • In place search for a relevant shop. • Search for products by walk. • Purchasing and manual billing. • Buy the products from the salesperson directly. </div>	Extract online & offline CH of BE
	4. EMOTIONS EM <small>BEFORE / AFTER</small> <ul style="list-style-type: none"> • Before: Want to buy products on huge rush and frequently ask the vendor to show more products. • After: Anywhere anytime shopping and can easily see any number of products even if they don't buy. 			

4.REQUIREMENT ANALYSIS

4.1 Functional requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Orders	As a user, can order the product through online shopping
FR-4	Chatbot	To finding the feelgood product by using automated chatbot
FR-5	Billing	Numerous payment option like google pay, amazon pay

4.2 Non-Functional requirements:

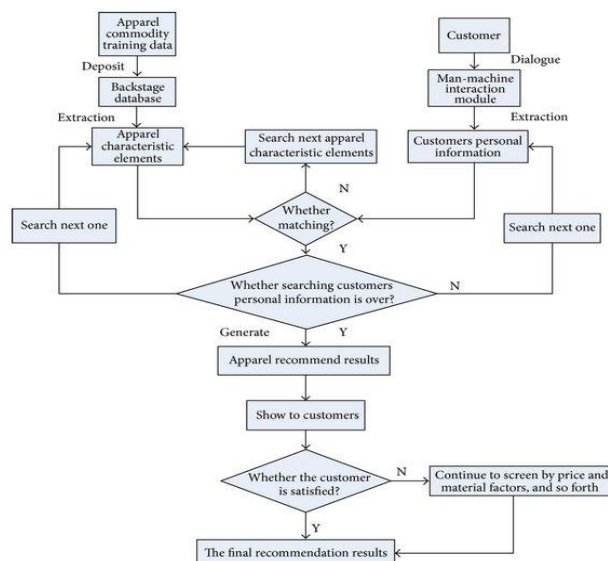
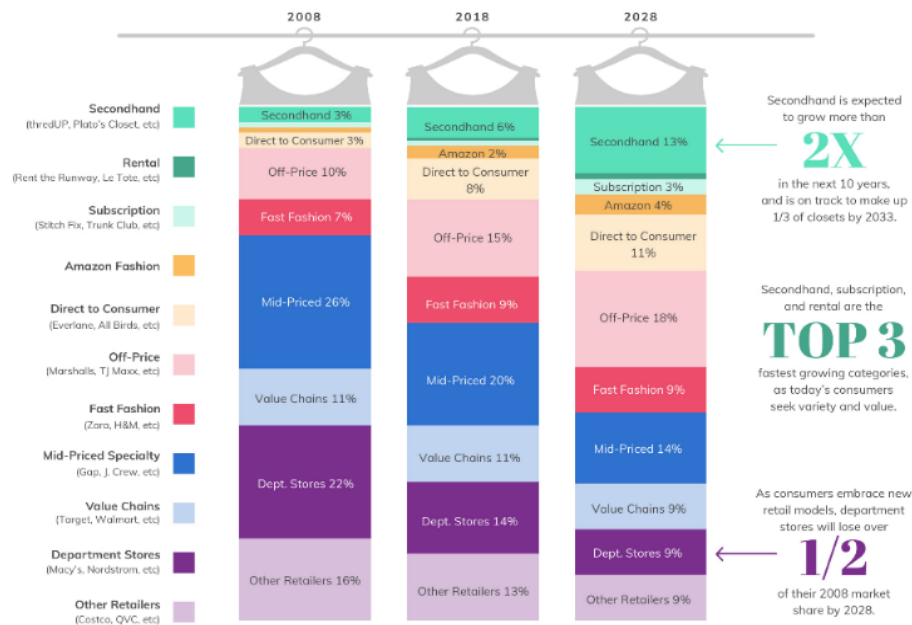
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Regardless of the size of your business, the website of your business should be easy to use for even a non-technical user
NFR-2	Security	This NFR deals with the security of the user data captured by the applications against any breach
NFR-3	Reliability	It is also considered for the different admin roles by which you can control who can create, view, copy, change or delete information
NFR-4	Performance	For increasing the traffic on your website, you have to give special attention to the performance
NFR-5	Availability	Auto scaling, Availability zones, Elastic load balancing
NFR-6	Scalability	Scalability is a measure of a system being able to appropriately handle the increase/decrease in workload

5.PROJECT DESIGN

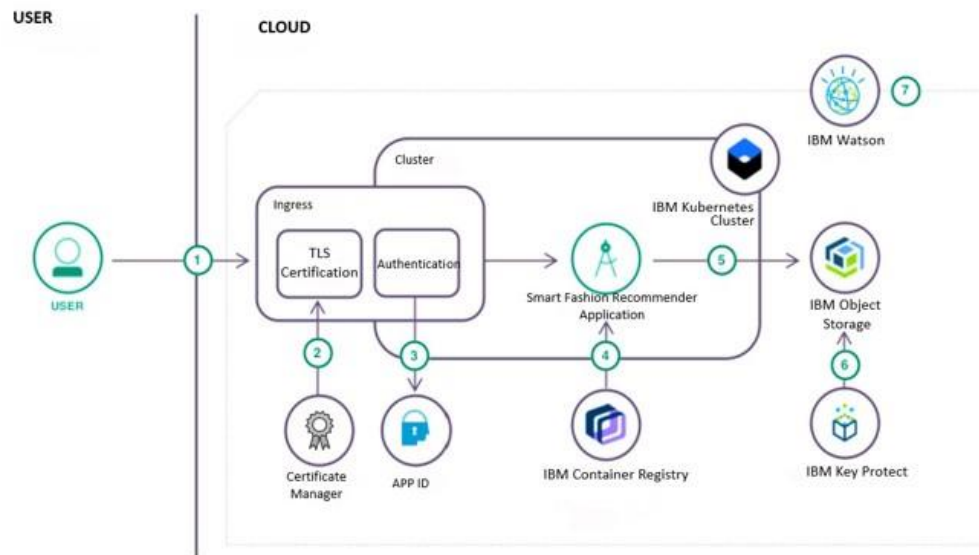
5.1 Data Flow Diagrams:

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.

Example: (Simplified)



5.2 Solution & Technical Architecture:



5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-1
		USN-4	As a user, I can register for the application through Gmail	I can access my account / dashboard	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can receive confirmation email & click confirm	High	Sprint-1
	Dashboard	US	As a user can view the new and latest product on the dashboard	I can receive the latest product details	High	Sprint-2
Customer (Web user)	Website search	US	As a user, can search the product on the various website	Happy as the customer finding the numerous	High	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Quality	US	Opens a page to check product details checks other products for proper comparison	Higher quality products in the top categories I can gets a good product	High	Sprint-2
Product Evaluation	Request	US	To finding the feelgood product by using automated chat bot	if the product may be available, it will be display	High	Sprint-3
Chat bot	Orders	US	As a user, can order a product	I can receive the confirmation message when ordering the product	High	Sprint-3
Customer (Activity)	Exchange and cancelling	US	As a user, can exchange the product and cancelling the product I	I can receive the message when i cancel the product and exchange the product	High	Sprint-4
	Billing	US	Makes the Payment for the preferred product as a user, can cancel the product and the payment can be returned to their account	numerous payment option like google pay, amazon pay an easy check out process	High	Sprint-3
Payment	Dashboard	US	As a user, can share their feedback for the product on the feedback page	Friendly and help full customer support	High	Sprint-4

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

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-1
		USN-4	As a user, I can register for the application through Gmail	I can access my account / dashboard	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can receive confirmation email & click confirm	High	Sprint-1
	Dashboard	US	As a user can view the new and latest product on the dashboard	I can receive the latest product details	High	Sprint-2
Customer (Web user)	Website search	US	As a user, can search the product on the various websites	Happy as the customer finding the numerous options	High	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Quality	US	Opens a page to check product details checks other products for proper comparison	Higher quality products in the top categories I can gets a good product	High	Sprint-2
Product Evaluation	Request	US	To finding the feelgood product by using automated chat bot	if the product may be available, it will be display	High	Sprint-3
Chat bot	Orders	US	As a user, can order a product	I can receive the confirmation message when ordering the product	High	Sprint-3
Customer (Activity)	Exchange and cancelling	US	As a user, can exchange the product and cancelling the product I	I can receive the message when i cancel the product and exchange the product	High	Sprint-4
	Billing	US	Makes the Payment for the preferred product as a user, can cancel the product and the payment can be returned to their account	numerous payment option like google pay, amazon pay an easy check out process	High	Sprint-3
Payment	Dashboard	US	As a user, can share their feedback for the product on the feedback	Friendly and help full customer support	High	Sprint-4

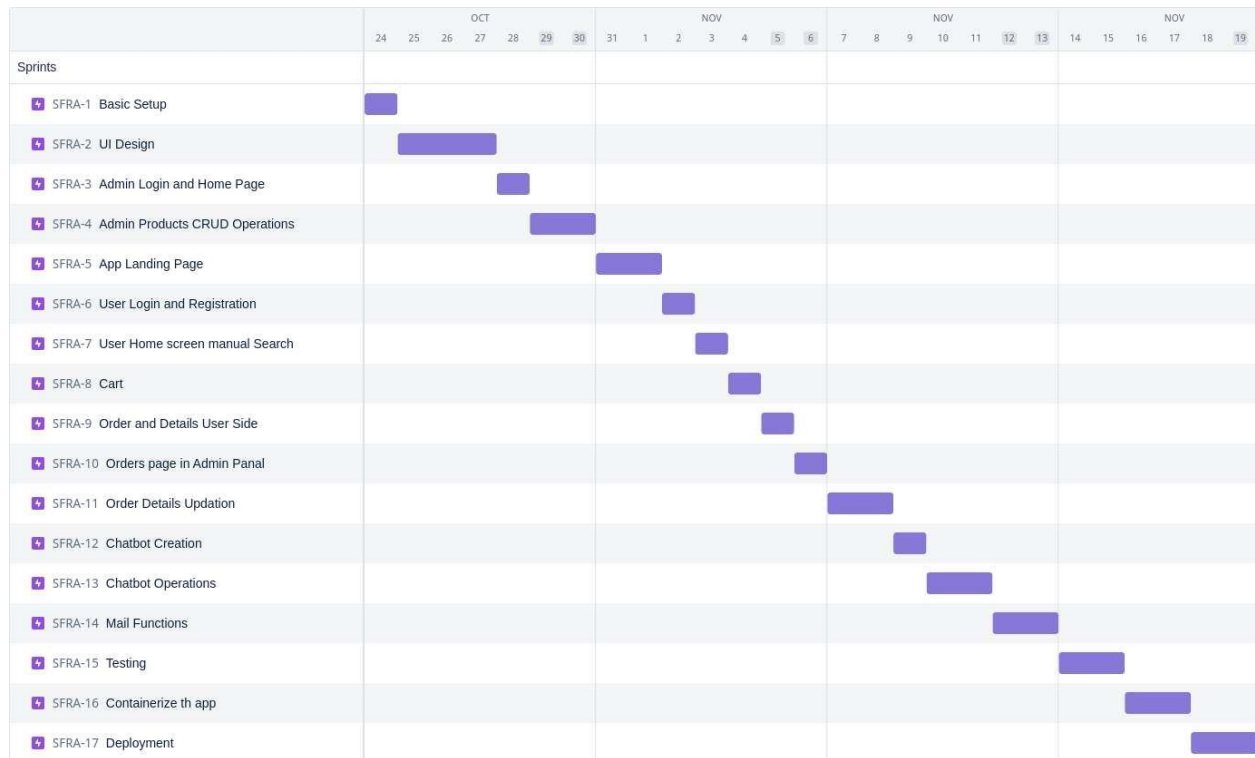
6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	TeamMembers
Sprint-1	AdminPanel	USN-1	As a Admin, <ul style="list-style-type: none"> Can login Products- CRUD User List OrdersList 	20	High	S HARISH N JAVEED HUSSAIN R MADHAN KUMAR C S HARIHARAN
Sprint-2	User Panel	USN-2	As a user, <ul style="list-style-type: none"> Register , Login , EmailVerification ManualSearch Order placement, OrderDetails 	20	High	S HARISH N JAVEED HUSSAIN R MADHAN KUMAR C S HARIHARAN
Sprint-3	ChatBot	USN-3	<ul style="list-style-type: none"> Automatic product search based on user information Can make order without any manual process 	20	High	S HARISH N JAVEED HUSSAIN R MADHAN KUMAR C S HARIHARAN
Sprint-4	Testing & Deploy	USN-4	<ul style="list-style-type: none"> UnitTesting Containerizethe app Deploymentin IBM CLOUD 	20	High	S HARISH N JAVEED HUSSAIN R MADHAN KUMAR C S HARIHARAN

6.2 Sprint Delivery Schedule:

Sprint	Total Story Points	Duration	SprintStart Date	SprintEndDate (Planned)	Story Points Completed (as on PlannedEndDate)	SprintRelease Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022		29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		19 Nov 2022

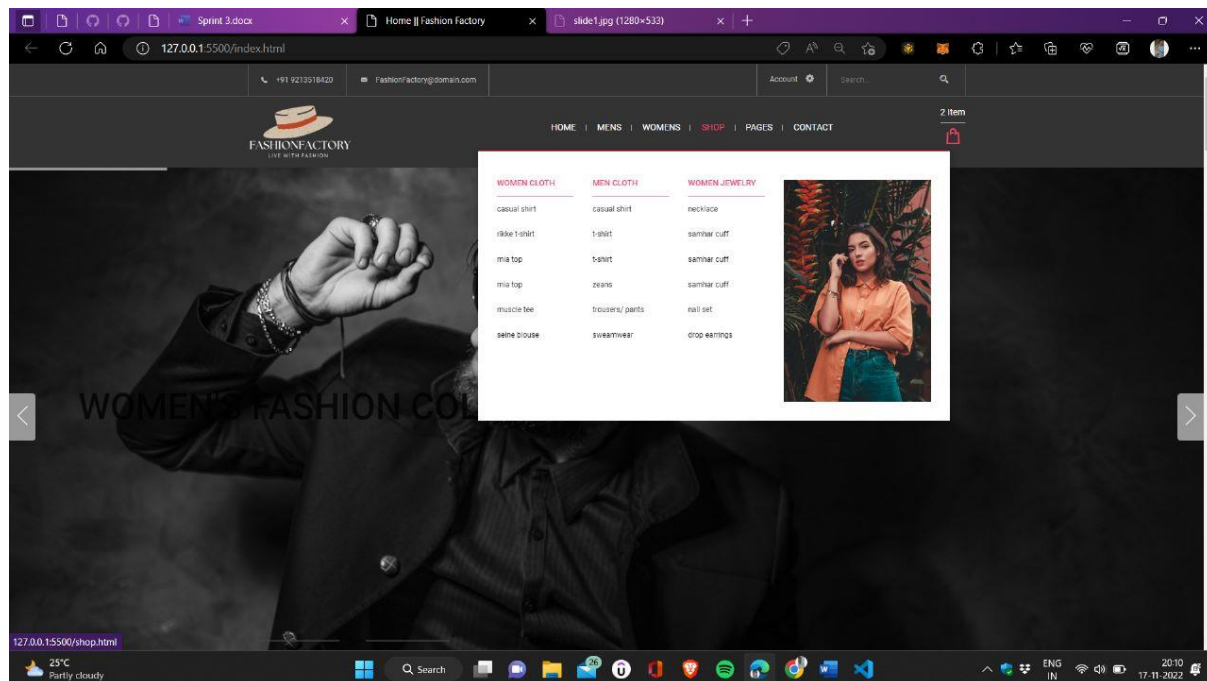
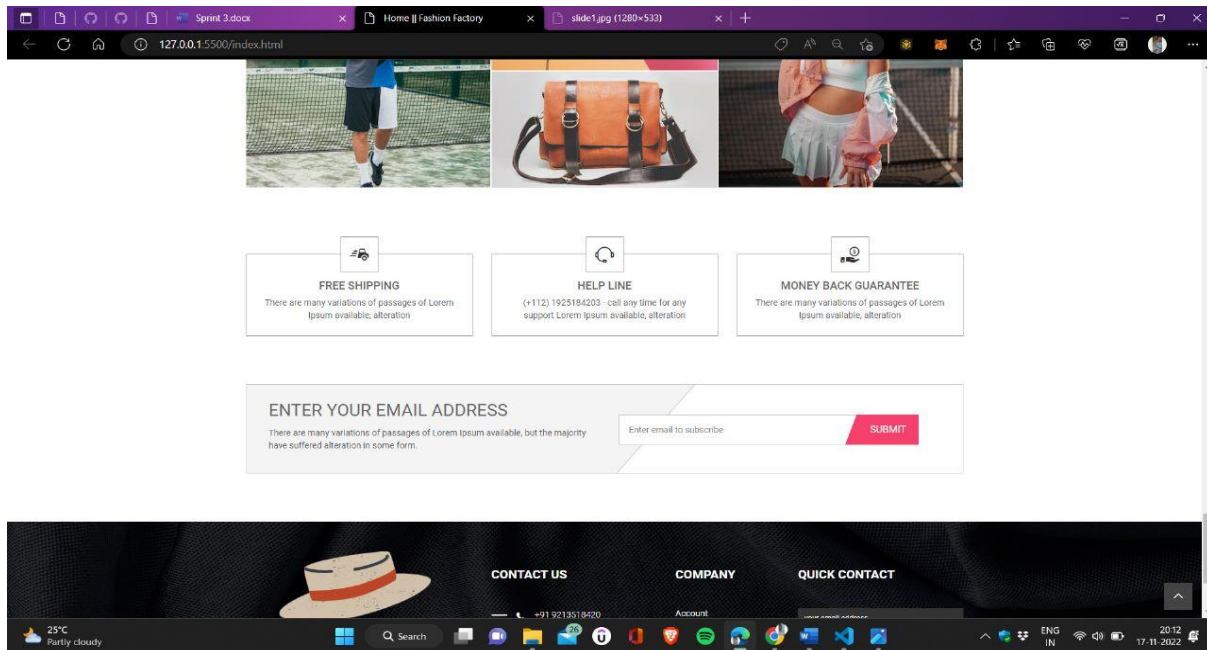
6.3 Reports from JIRA:

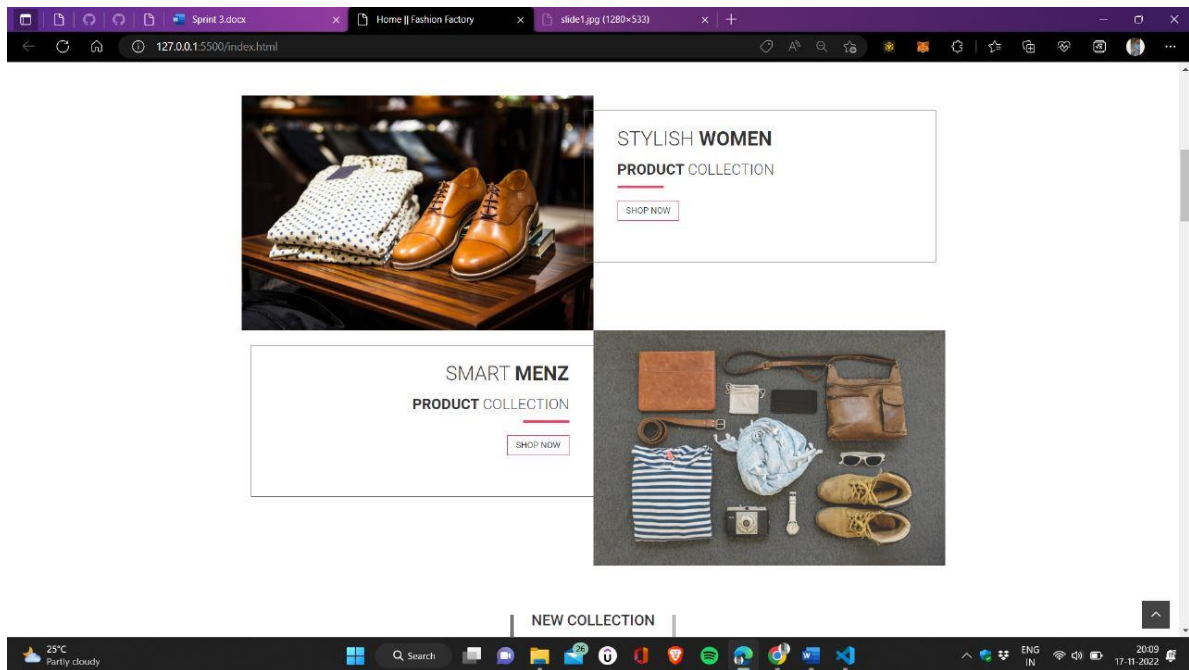


7. CODING & SOLUTIONING:

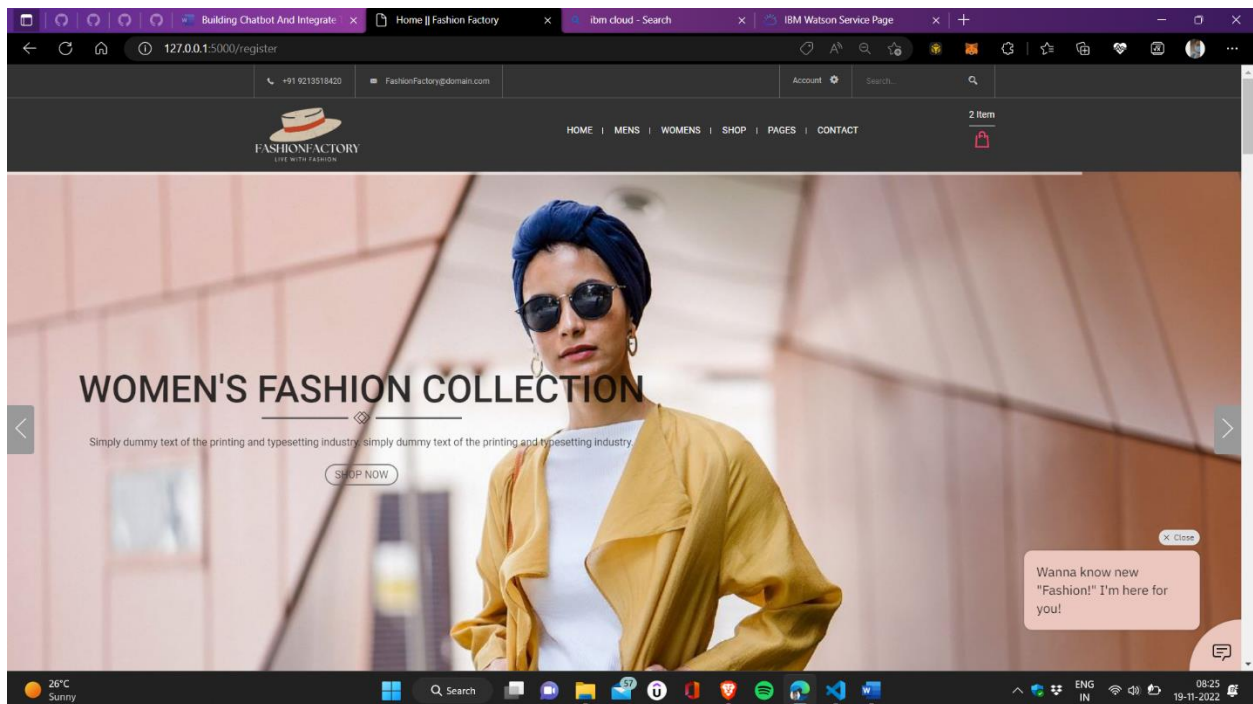
(Explain the features added in the project along with code)

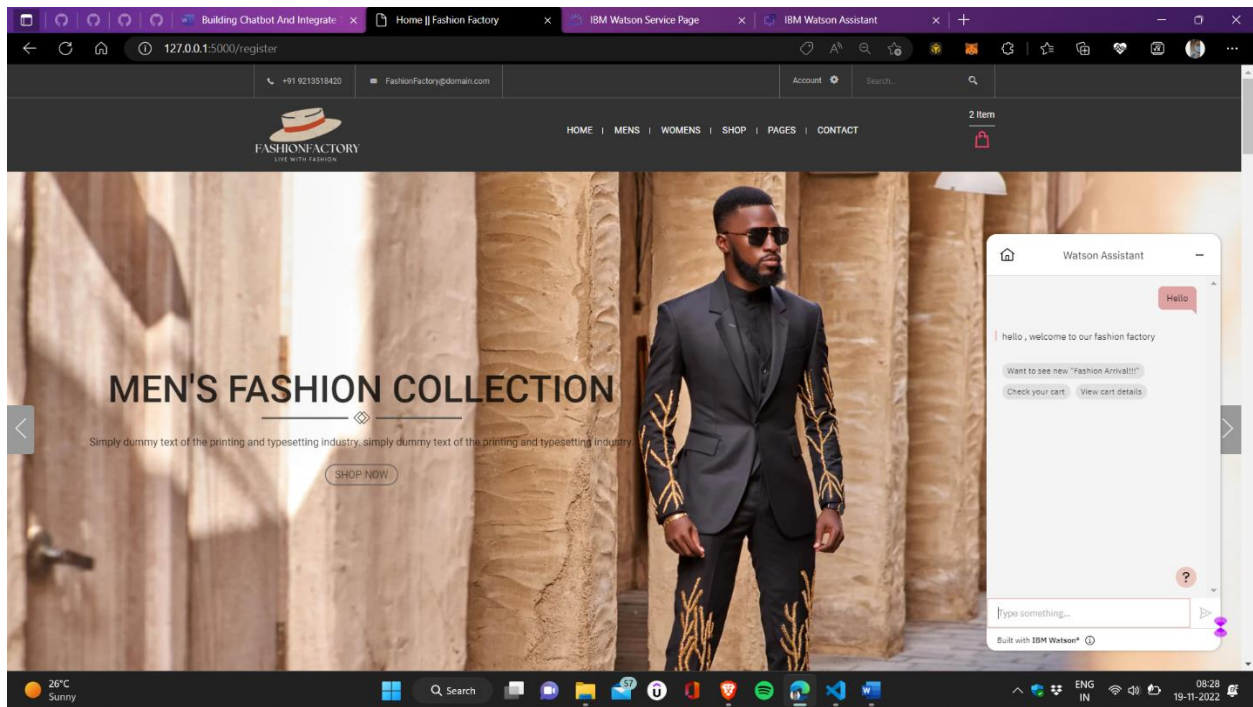
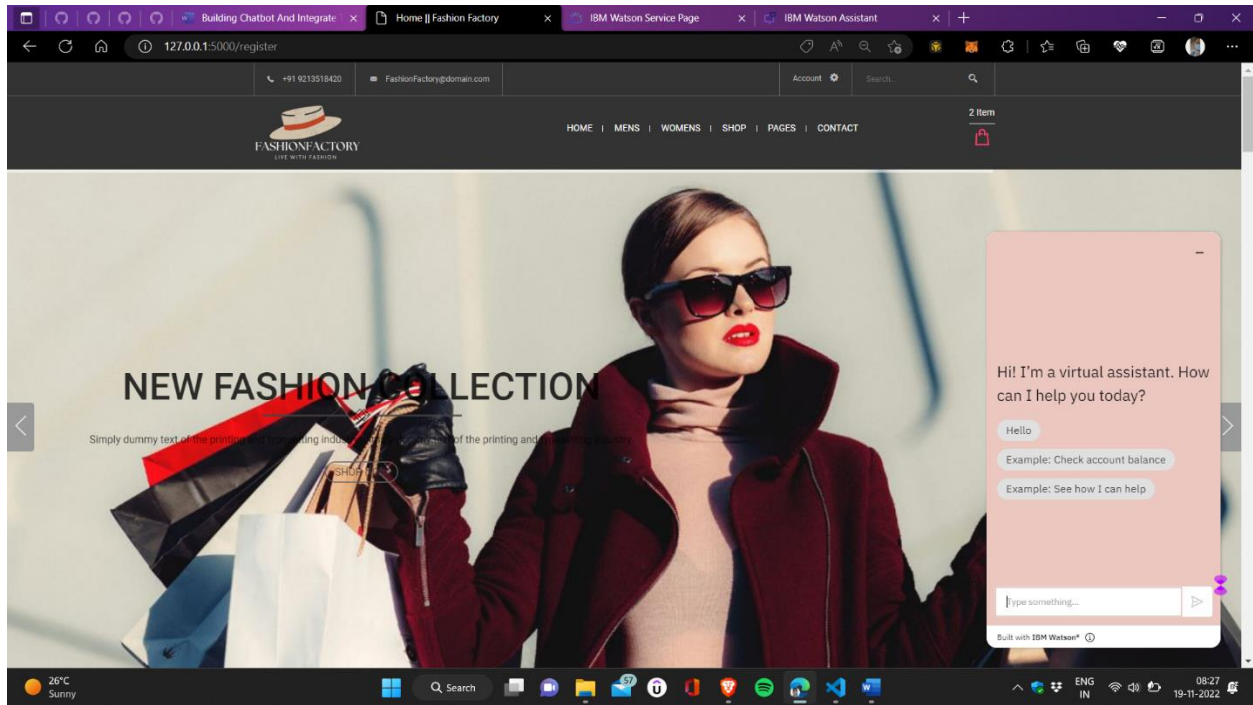
7.1 Features 1: Various collection and combination of outfit.

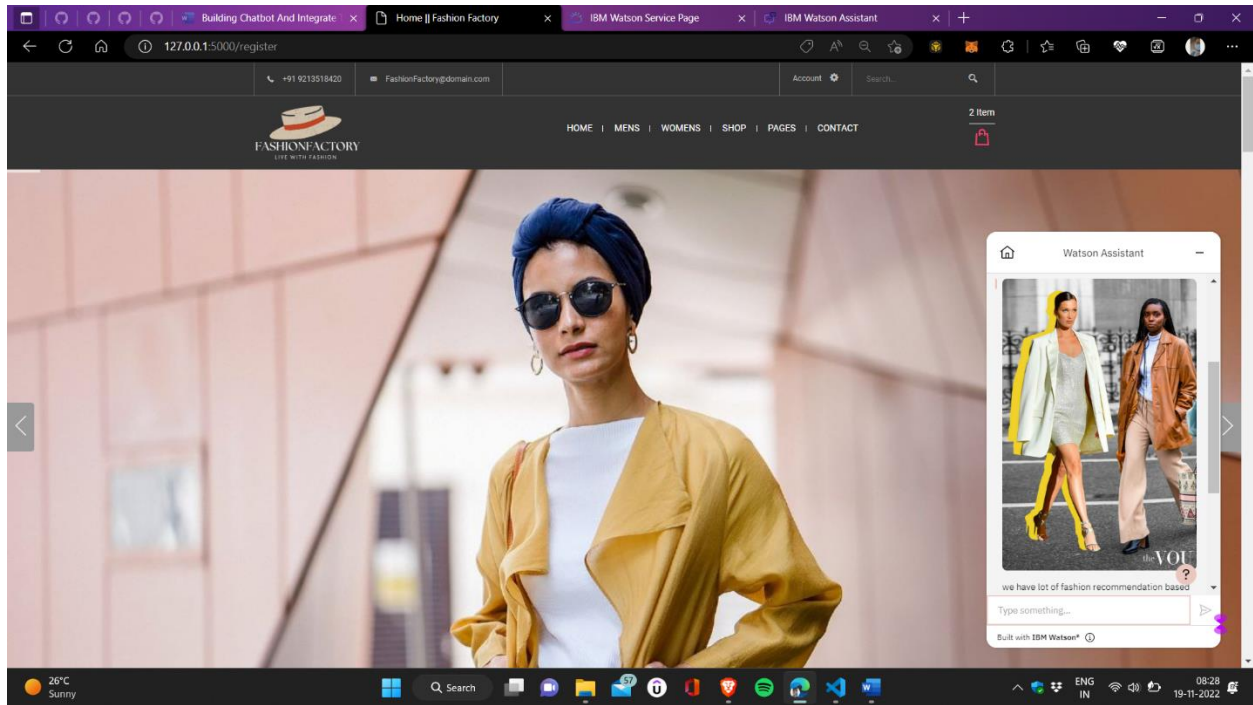




7.2 Feature 2: Chatbot feature.







Coding:

App.py

```
from flask import Flask, render_template, redirect, request, session, url_for, flash
from markupsafe import escape
# import requests
# import json
# from turtle import st
import ibm_db

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31198;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=pgw62994;PWD=rxrMsd0egpHDdyvI", '', '')

print(conn)
print("connection successful...")

app = Flask(__name__)
app.secret_key = 'gtvhfryj123#@%'
```

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

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

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

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

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

@app.route('/myaccount')
def myaccount():
    return render_template('my-account.html')

@app.route('/shop-list')
def shoplist():
    return render_template('shop-list.html')

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

@app.route('/single-product')
def singleproduct():
    return render_template('single-product.html')

@app.route('/thank-you')
def thankyou():
    return render_template('thank-you.html')

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

```

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        name = request.form['name']
        email = request.form['email']
        password = request.form['password']
        phonenumber = request.form['phonenumber']

        sql = "SELECT * FROM SFRA WHERE email = ?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            return render_template('index.html', msg="already members")
        else:
            insert_sql = "INSERT INTO SFRA VALUES (?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, name)
            ibm_db.bind_param(prepare_stmt, 2, email)
            ibm_db.bind_param(prepare_stmt, 3, password)
            ibm_db.bind_param(prepare_stmt, 4, phonenumber)
            ibm_db.execute(prepare_stmt)

            return render_template('index.html', msg="saved successfully")

@app.route('/login', methods=['GET', 'POST'])
def loginagent():
    app.secret_key = 'praveenkumhesbf/.[.;;ar'
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        print(email,password)

        sql = f"SELECT * FROM sfra WHERE email='{escape(email)}' and password='{escape(password)}'"
        stmt = ibm_db.exec_immediate(conn, sql)
        data = ibm_db.fetch_both(stmt)

        if data:
            session["mail"] = escape(email)
            session["password"] = escape(password)
            return redirect(url_for('home'))

```

```
        else:
            return render_template('login.html', msg="Account does not exist or
invalid Credentials")

        return "NOT WORKING!!??"

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

Output:

<ibm_db.IBM_DBConnection object at 0x000002E2109FE2B0>

connection successful...

* Serving Flask app 'app'

* Debug mode: on

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

* Running on http://127.0.0.1:5000

Press CTRL+C to quit

* Restarting with stat

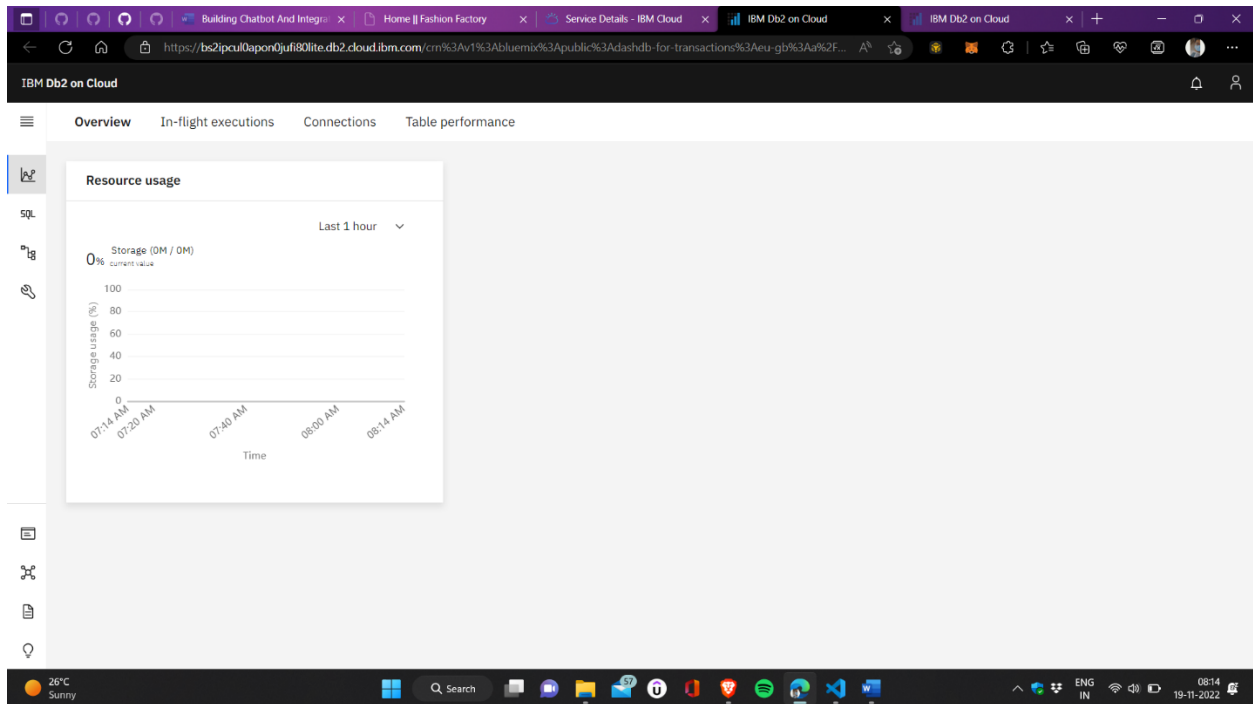
<ibm_db.IBM_DBConnection object at 0x0000026E897ECEB0>

connection successful...

* Debugger is active!

* Debugger PIN: 572-315-639

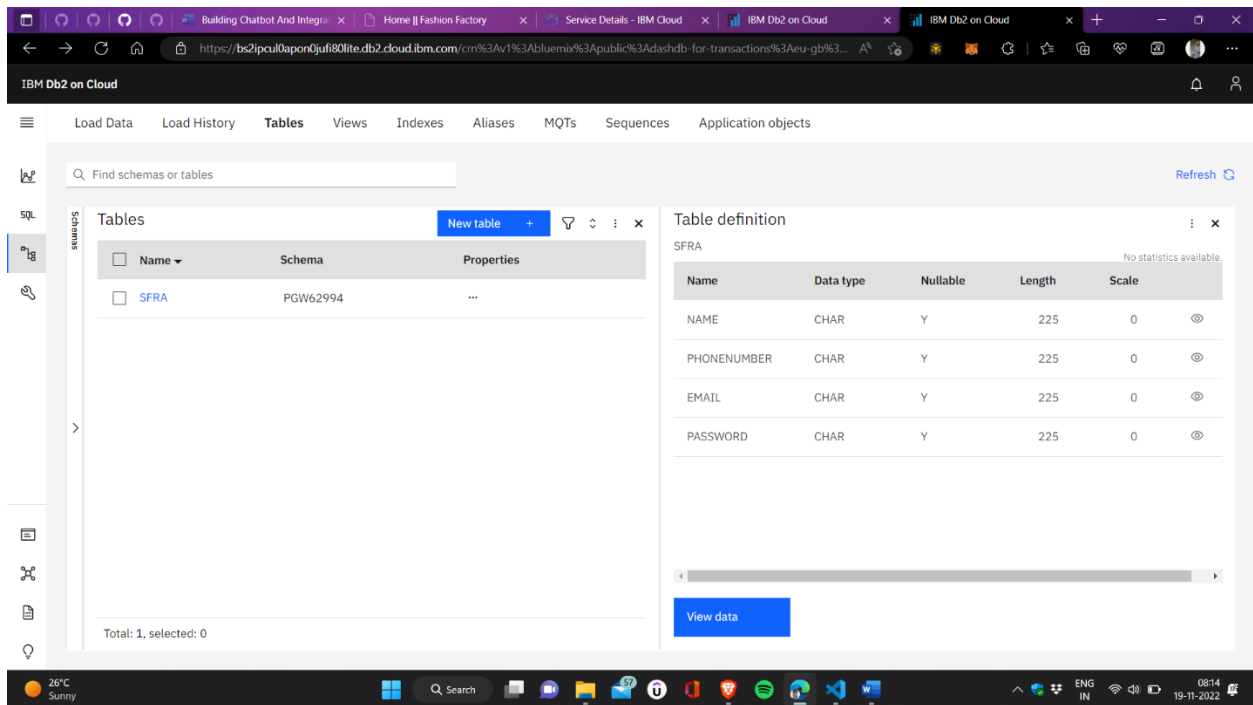
7.3 Database Schema (if Applicable):



The screenshot shows the 'Tables' section of the IBM Db2 on Cloud console. The 'Schemas' table lists the schema 'PGW62994' of type 'User' with 1 table. The 'Tables' table lists the table 'SFRA' in the 'PGW62994' schema. The 'Schemas' table has a total of 1 selected, and the 'Tables' table has a total of 1 selected.

Name	Type	Tables
PGW62994	User	1

Name	Schema	Properties
SFRA	PGW62994	...



8.TESTING

Software testing is used to assess the quality of the product. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

8.1Test Cases:

Testcase1: Does the flask application is perfectly created and in works in very good condition?

Testcase2: Does the Send-Grid integration is working correctly?

Testcase3: Does the db2 is perfectly connected to the application?

Testcase4: Can the chat-bot which is created using Watson assistant is correctly recommending the job for the end users?

Testcase 5: Whether the application is working correctly without any interruptions?

8.2 User Acceptance Testing:

User acceptance testing is a type of testing that is used to determine whether a software system is suitable for use by end users. It is the process of verifying that a system meets the requirements of the user, and that the user can use the system for its intended purpose. User acceptance testing (UAT) is a process of verifying that a system meets the needs of the end users and that they can use it. This can be done through a variety of methods, such as interviews, surveys, or observation. UAT is important in water quality analysis and prediction because it helps ensure that the system being developed will be useful to those who will be using it. By testing with act users, developers can get feedback on the system and make sure it is meeting the needs of the users.

9. RESULTS

9.1 Performance Metrics:

a) Implementation of web application:

To create the web application to interact with the users. The users here is commonly job seeker and job provider. Login, Signup, Job searching have separate pages where we can access into different work functions.

b) SendGrid Integration:

The flask application that we created is to get integrated with sendgrid which provide the e-mail interface for communication purpose.

c) Developing chatbot:

To develop a chat-bot so that, that can be very interactive to the users who are using the application and to recommend the jobs based on the job seekers interests.

d) Deployment of Application:

Finally, the developed application is to be deployed in the cloud.

1. Accuracy

The accuracy metric is one of the simplest Classification metrics to implement, and it can be determined as the number of correct predictions to the total number of predictions.

To implement an accuracy metric, we can compare ground truth and predicted value in a loop.

10. ADVANTAGES AND DISADVANTAGES

Advantages:

- The main advantage of our application is that there is a direct one-way communication between the fashion seeker and the designers.
- There is a chat-bot which gives the directions to the users what the user get benefit and it recommends the live fashion based on the user interest.
- The application is an open source one which does not ask for the money.

Disadvantage:

- One disadvantage of the application is that it is not a full paced one.
- Another disadvantage of smart fashion recommender application is that it is unaware of machine language stack. No AI is implemented here.

11. CONCLUSION

In this work, we have presented our proposal for the automatic recommendation of new fashion arrival. Our goal here is being able to build methods being able to deliver appropriate new fashion to those customers that could be potentially interested on that design and trending one. To do that, we have based our research efforts on two well known

classification methods: random forests (RF) and support vector machines (SVM). Our empirical evaluation shows us interesting facts. For example, RF are more likely to be interpreted although they do not present a particularly good performance in relation to SVM. On the other hand, SVM are more accurate, although they work with a model being much harder to interpret by human. What it is clear is, that in both cases, we have shown that these two methods are quite appropriate for accurately

working in the context of automatic smart fashion recommender application. The users don't know the trend only because of lack of trending fashion set in their domain or missing out new arrivals. When a customer is missing out new fashion because not of getting the notification, he might lose the trending fashion which he really deserved for it. The main purpose of the smart fashion recommender application is to provide new fashion for every single person. The only thing the customer wants to do is just to approach the application and apply for the smart fashion recommender application. He will be provided with the login credentials with the confirmation email. There he can find numerous of new fashion. Customers will be guided with the in-built chat bot, which guides the new fashion to apply for the smart fashion and the availability of design based on their interest. The chat bot is built with IBM Watson Assistant that is very much helpful in collecting the fashion interests and guides them to apply for it.

12. FUTURE SCOPE

As future work, we propose to design novel computational methods being able to process the textual description from the fashion arrivals. At that point, we were using just the quantitative information that is advertised. However, we think that the way an offer is written can help attracting potential candidates as well, maybe new methods for natural language processing using neural networks could help in this task. We also would like to explore the possibilities to work with expert knowledge via kernel mapping in the case of SVM as we mentioned earlier. Finally, it is also necessary to study how to integrate this technology with existing web information systems so that these two methods can be put into operation by the industry.