SMART FASHION RECOMMENDER APPLICATION

Project Report

College Name: PSR Engineering College

Team ID: PNT2022TMID13258

Team Members:

Gayathri S [951919CS024]

Kaviya R [951919CS041]

Harivanisri G [951919CS030]

Leelavinothini B [951919CS045]

CONTENTS

S.NO	Title	Page No
1	Introduction	3
	1.1 Project Overview	3
	1.2 Purpose	3
2	Literature Survey	4
	2.1 Existing Problem	4
	2.2 Reference	5
	2.3 Problem Statement Definition	5
3	Ideation & Proposed Solution	6
	3.1 Empathy Map Canvas	6
	3.2 Ideation & Brainstorming	7
	3.3 Proposed Solution	7
	3.4 Problem Solution fit	8
4	Requirement Analysis	8
	4.1 Functional Requirement	8
	4.2 Non-Functional Requirement	9
5	Project Design	9
	5.1 Data Flow Diagrams	9
	5.2 Solution & Technical Architecture	10
	5.3 User Stories	12
6	Project Planning & Scheduling	13
	6.1 Sprint Delivery Schedule	13
7	Coding and Solution	15
	7.1 Feature	15
	7.2Database Schema	17
8	Results	46
	8.1 Performance Metrics	46
9	Advantages and Disadvantages	47
10	Conclusion	48
11	Future Scope	49
12	Appendix	49

1.INTRODUCTION

1.1 Project Overview

Recommender systems help users navigate large collections of products to find items relevant to their interests leveraging large amounts of product information and user signals like product views, followed or ignored items, purchases or web-page visits to determine how, when and what to recommend to customers by UI. Recommender systems have grown to be an essential part of all large Internet retailers, driving up to 35% of Amazon sales or over 80% of the content watched on Netflix. In this work we are interested in recommender systems that operate in one particular vertical market: garments and fashion products. This setting introduces a particular set of challenges and sub-problems, that are relevant for developing effective recommender systems. Due to market dynamics and customer preferences, there is a large vocabulary of distinct fashion products, as well as high turnover. Furthermore, precise and detailed product information is often not available, making it difficult to establish similarity between products. To deal with the aforementioned problems, and given the visual and aesthetic nature of fashion products, there is a growing body of computer vision research addressing tasks like localizing fashion items determining their category and attributes or establishing the degree of similarity to other products to name only a few. Although works in the computer vision literature often don't consider personalization (or recommendation), their predictions and embeddings can be leveraged by recommender systems and chatbot, thus mitigating sparsity and cold start problems. Another relevant fashion problem that has attracted the attention of computer vision research is that of combining garments into complete outfits. Several works have studied how to learn the compatibility between fashion items using both professional photos of models wearing designer-created outfits, and social media pictures from 'influencers' and normal people. In addition to allowing recommendations in chatbot by UI.

1.2 Purpose

Recommender system has the ability to predict whether a particular user would prefer an item or not based on the user's profile. Recommender systems are beneficial to both service providers and users. They reduce transaction costs of finding and selecting items in an online shopping environment.

2.LITERATURE SURVEY

2.1 Existing Problem

In existing system only simple web application and their rating has been implemented in existing system, An ecommerce product recommendation engine is a piece of technology that displays recommended products to shoppers throughout your store. It uses machine learning to get smarter and show increasingly relevant products to shoppers based on their interests and previous browsing behavior. In existing model is content based filtering scheme has been employed in existing model **The content-based filtering method** analyzes customer data on the likes and dislikes of each user (cookies allow tracking over multiple visits), then makes recommendations based on the browsing history of that user. The idea behind content-based filtering is that if you enjoy a certain item, you'll likely also enjoy a similar item. An example of a content-based filtering system would be if you were listening to Pandora and consistently 'liked' downtempo jazz music.

The collaborative-filtering method incorporates data from users who have purchased similar products, then combines that information to make decisions about recommendations. The advantage to this filtering method is that it is capable of making complex recommendations on items such as music or movies without having to 'understand' what the item is. This method of filtering operates under the assumption that users will prefer recommendations that are based on purchases they made in the past. Here's an example: If customer A likes a specific line of products that customer B also likes (assuming they have similar interests), then collaborate-filtering would assume that customer A would like other products that customer B purchased and vice versa.

A hybrid method combines the content-based and collaborative-based methods to incorporate group decisions but focuses the output based on the attributes of a specific visitor. An example of a hybrid filtering system would be how Spotify curates its personalized 'Discover Weekly' playlists. If you've ever listened to a personalized Spotify playlist, it's shocking how accurately they're able to recommend songs based on what you like. The secret behind how they pull this off is through a complex hybrid filtering system that aggregates data on your listening habits as well as similar users' listening habits, to create a playlist of unique songs that align with your personal taste.

2.2 Reference

[1] Mohamed Elleuch, Anis Mezghani, Mariem Khemakhem, Monji Kherallah “Clothing Classification using Deep CNN Architecture based on Transfer Learning”, 2021

[2] Saurabh Gupta, Siddartha Agarwal , Apporve Dave. “Apparel Classifier and Recommender using Deep Learning” (2015).

[3] Bossard, Lukas, Matthias Dantone, Christian Leistner, Christian Wengert, Till Quack and Luc Van Gool. “Apparel Claasification with Style”. ACCV (2012).

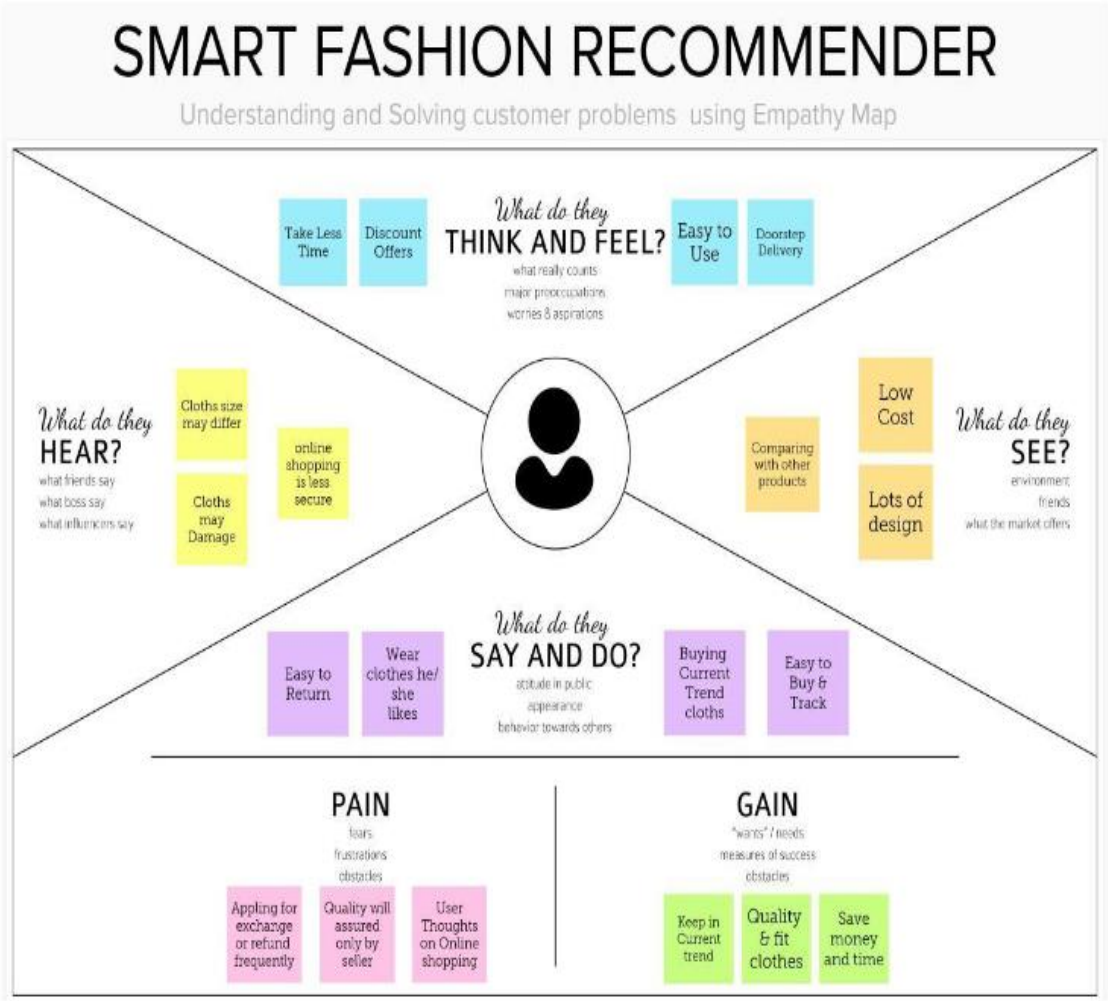
[4] Krizhevsky, Alex, Ilya Sutskever and Geoffery E. Hinton. “ImageNet claasifiaction with deep convolutional neural networks”. Communications of the ACM 60 (2012).

2.3 Problem Statement Definition

Over the years, much research has been conducted on fashion recommendation systems. Different techniques such as image processing, machine learning, or deep learning have been incorporated in the recommendation systems. Online e-stores like Amazon, eBay, etc. customize fashion recommendation systems to satisfy the daily requirements of their customers. A number of different approaches are proposed to study the purchase pattern of the customers. Our project is also works as a fashion recommenders using cloud application

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

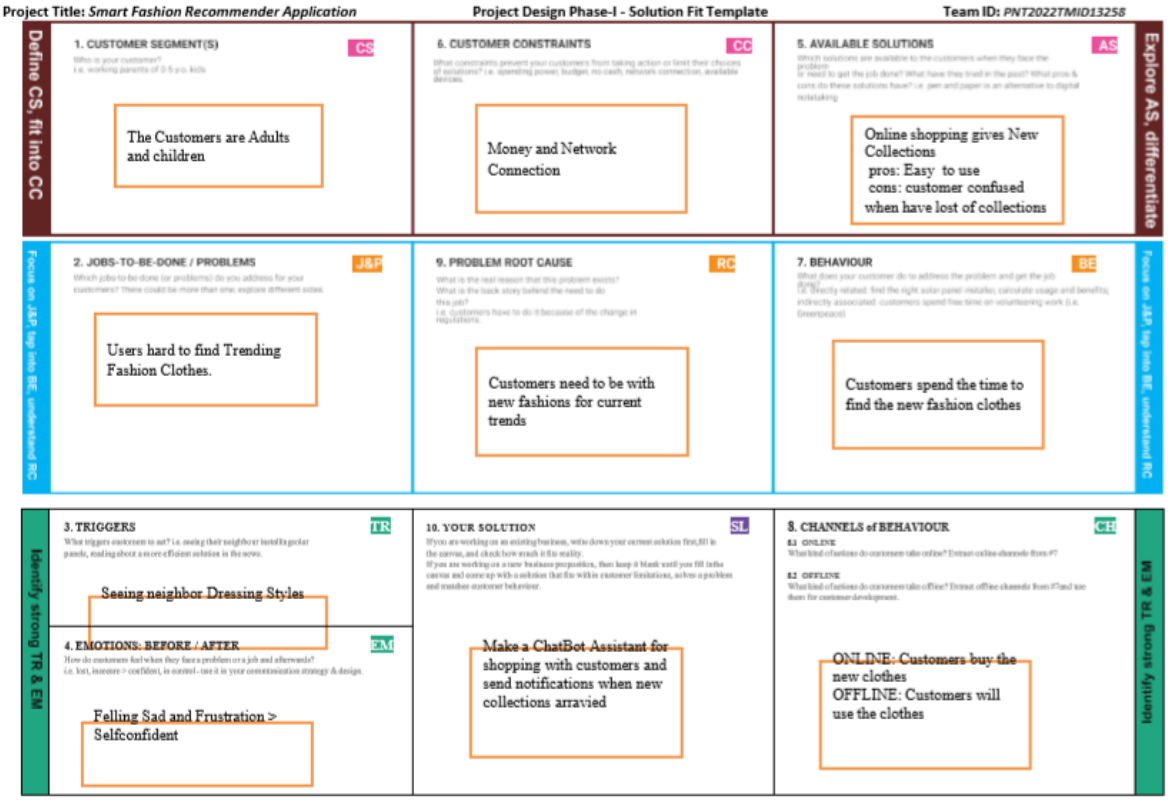


3.2 Ideation & Brainstorming

3.3 Proposed Solution

The proposed solutions web-based chatbot based applications is implemented with dash board applications is implemented for web applications The aim of this research is to build a perfume recommendation system. This system will help the user to get required perfumes. For that user has to provide description as a search query about the perfume according to his interest. This description can contain feelings, emotions, description, likes, dislikes and brand of the perfume. A chat bot will help the user to get the input in the form of search query and then provide the output as a recommended perfume what user is looking for. Initial work for research is collecting a data. Data required for this research contained the details in the form of name, brand, text descriptions, reviews, a list of notes. As we are using natural language processing, the text data must be pre-processed. It covers some tasks like making text data to lower case, removing stop words, tokenization, stemming, etc. shows tasks of pre-processing of data. Lowercasing – Lowercasing is the first step in data.

3.4 Problem Solution fit:



4. REQUIREMENT ANALYSIS:

4.1 Functional Requirement

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:

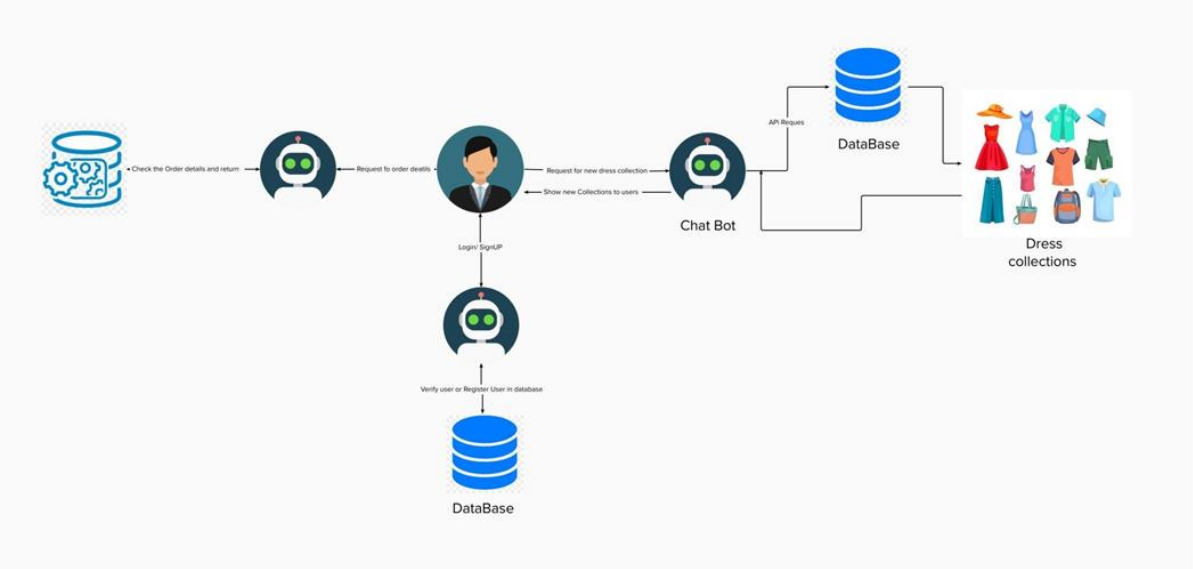
Following are the non-functional requirements of the proposed solution.

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**	Its 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:

Technical Architecture:

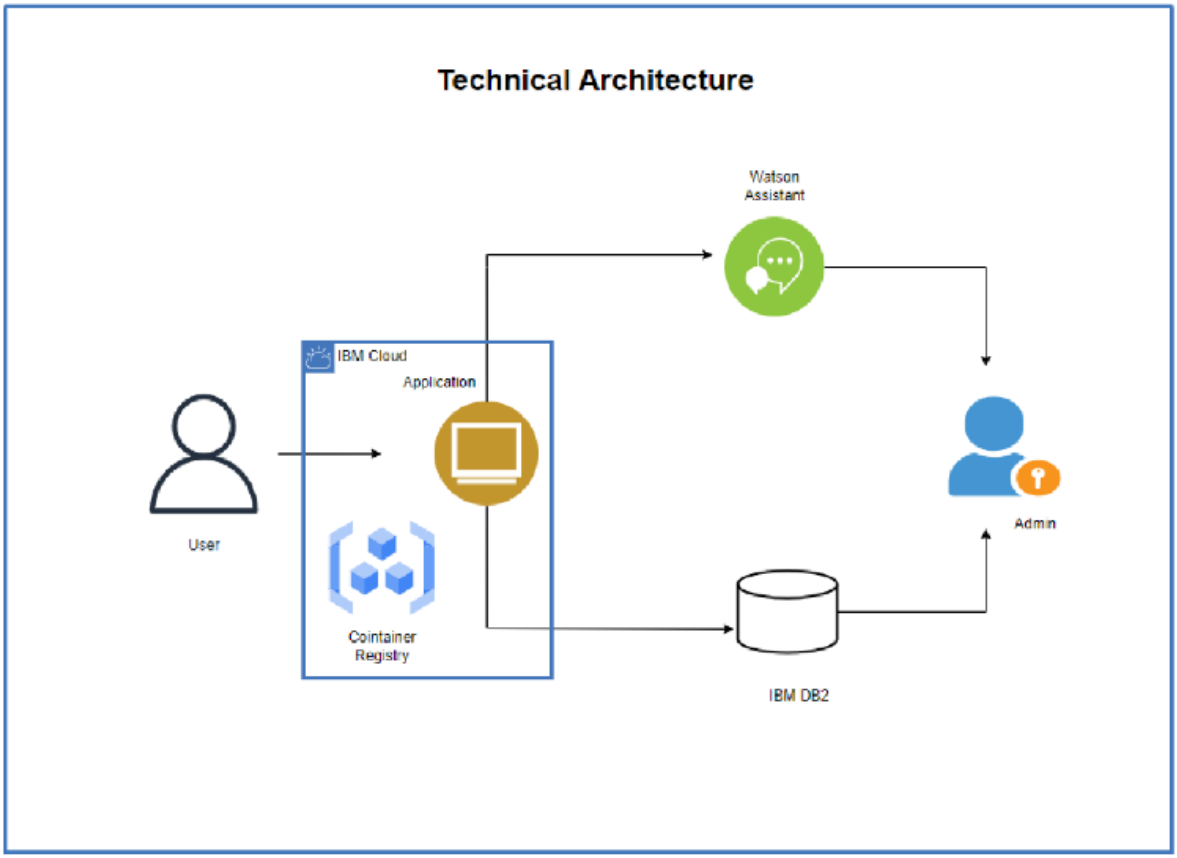


Table-1: Components & Technologies:

S.NO	Component	Description	Technology
1.	User Interface	How user interacts with application e.g., Web UI, Mobile App, ChatBot etc.	HTML,CSS, Javascript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configuration etc.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2
7.	File Storage	File storage requirements	IBM Block Storage
8.	Infrastructure (Server/Cloud)	Application Deployment on cloud Server Configuration: Db2/python	Kubernetes

Table-2: Application Characteristics:

S.NO	Characteristics	Description	Technology
1	Open-Source Frameworks	Flask	Python
2	Encryption Hashing and Salting	Encryption Hashing and Salting	Encryptions
3	Scalable Architecture	Getting resources to different parts of the system that need it	Microservices Architecture
4	Availability	The Application available 24//7	IBM Cloud
5	Performance	1000 request per day	IBM Watson

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 emailonce I have registered for the application	I can receive confirmationemail & 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 FacebookLogin	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 bylogin	High	Sprint-1
	Dashboard	USN-6	As a user , I can view the dashboard and byproducts		High	Sprit -2
Customer (Webuser)	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 Delivery Schedule:

Product Backlog, Sprint Schedule, and Estimation

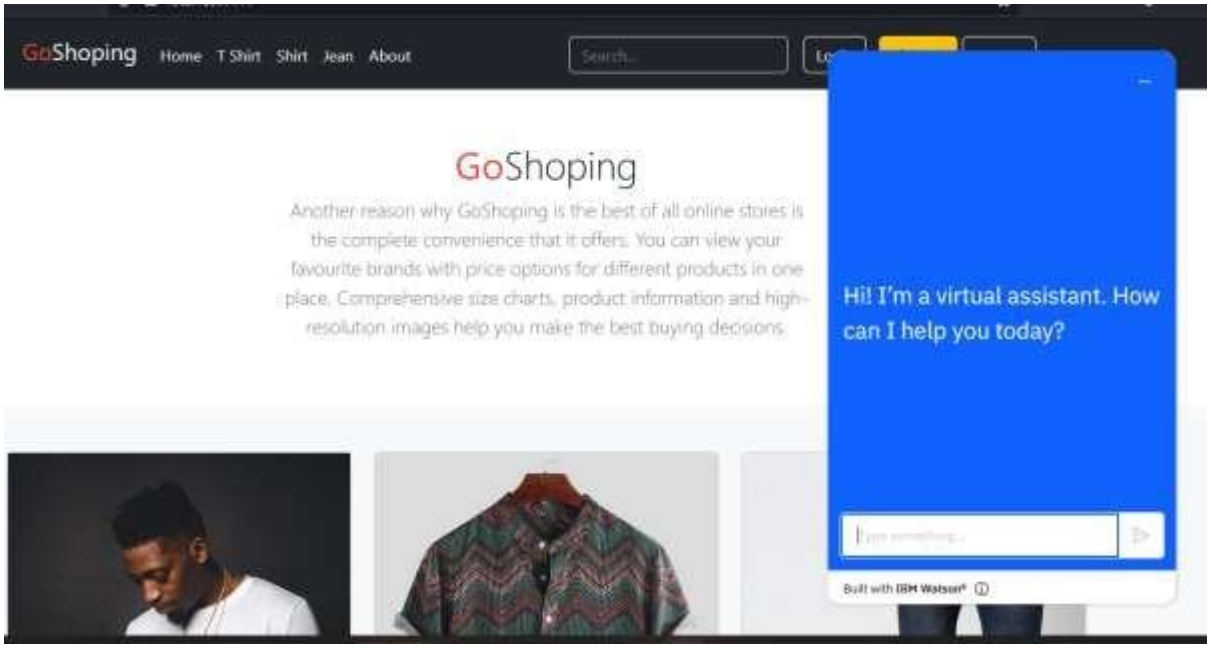
Use the below template to create product backlog and sprint schedule

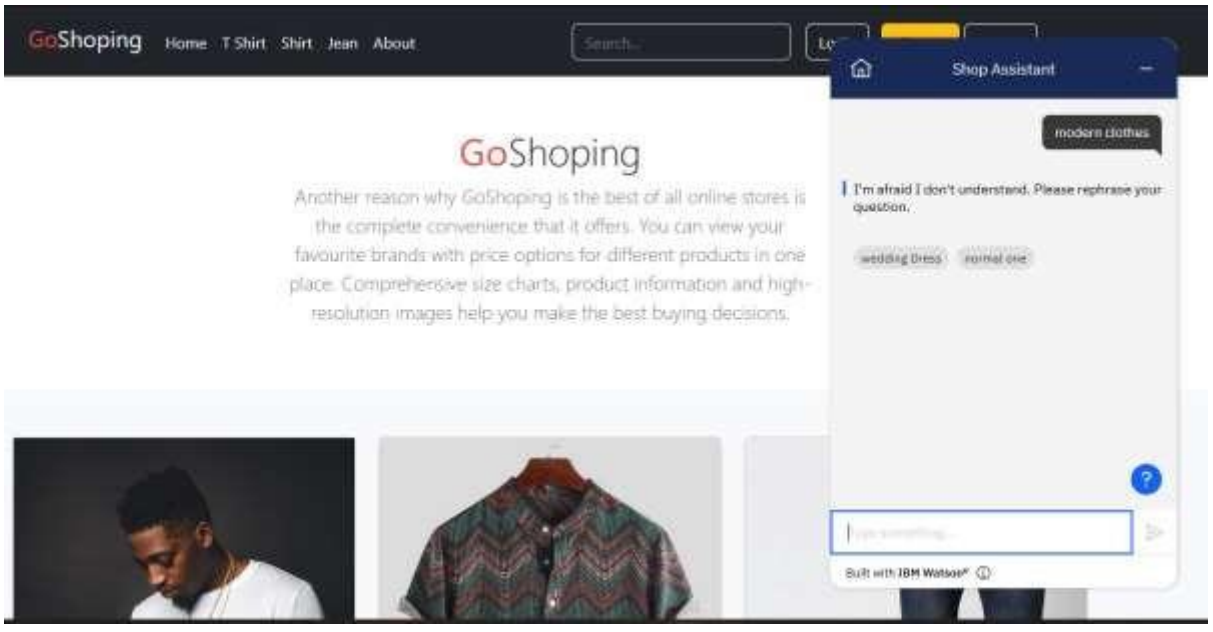
Sprint	Functional Requirement (Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint 1	User Panel	USN-1	The user will login into the website and go through the products available on the website	20	High	Gayathri S Kaviya R Harivansri G Leelavinothini B

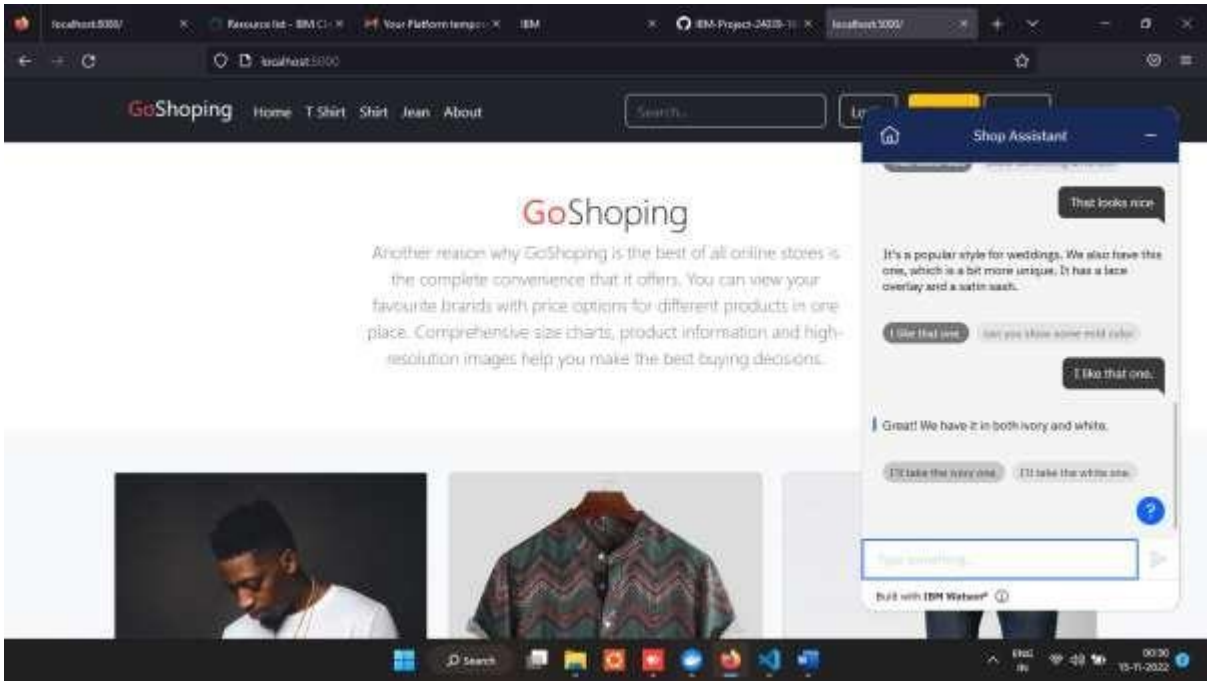
Sprint 2	Admin Panel	USN-2	The role of the admin is to check out the database about the stock and have a track of all things that the users are purchasing.	20	High	Gayathri S Kaviya R Harivansri G Leelavinothini B
Sprint 3	Chat Bot	USN-3	The user can directly talk to chatbot regarding the products. Get the recommendations based on information provided by the user.	20	High	Gayathri S Kaviya R Harivansri G Leelavinothini B
Sprint 4	Final Delivery	USN-4	Container of applications using docker kubernetes and deployment the application. Create the documentation and final submit the application	20	High	Gayathri S Kaviya R Harivansri G Leelavinothini B

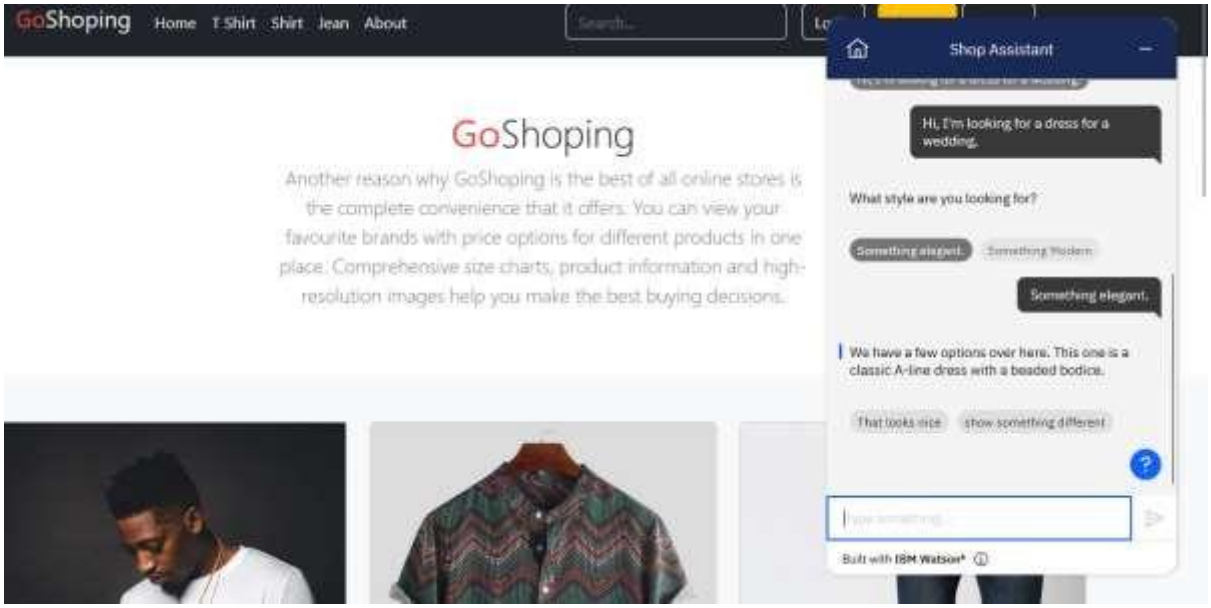
7. CODING AND SOLUTION:

7.1 Feature 1:Chat Bot



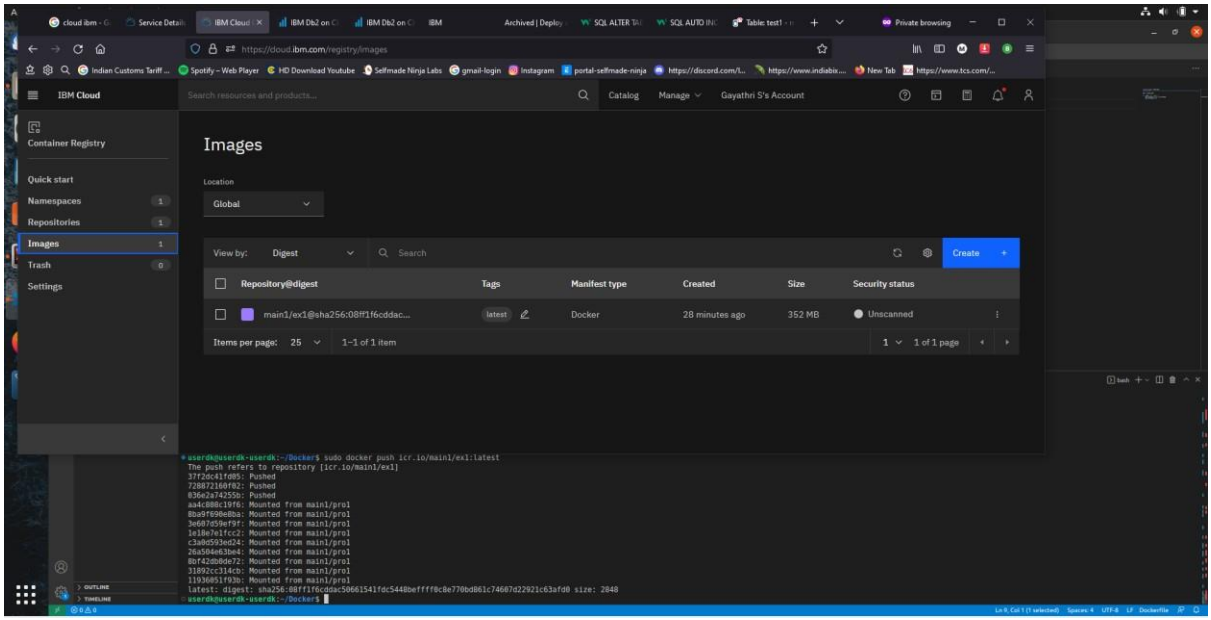


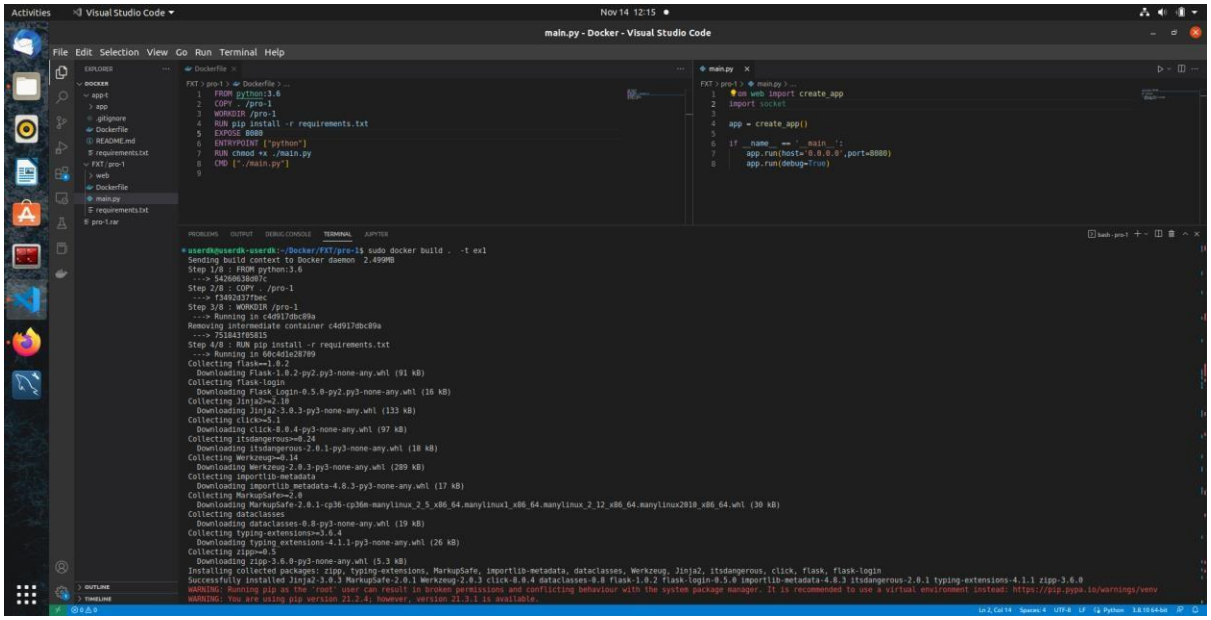

GoShoping
Home
T Shirt
Shirt
Jean
About
GoShoping
Another reason why GoShoping is the best of all online stores is the complete convenience that it offers. You can view your favourite brands with price options for different products in one place. Comprehensive size charts, product information and high-resolution images help you make the best buying decisions.
Shop Assistant
That looks nice
It's a popular style for weddings. We also have this one, which is a bit more unique. It has a lace overlay and a satin sash.
I like that one.
Great! We have it in both ivory and white.
Built with IBM Watson®

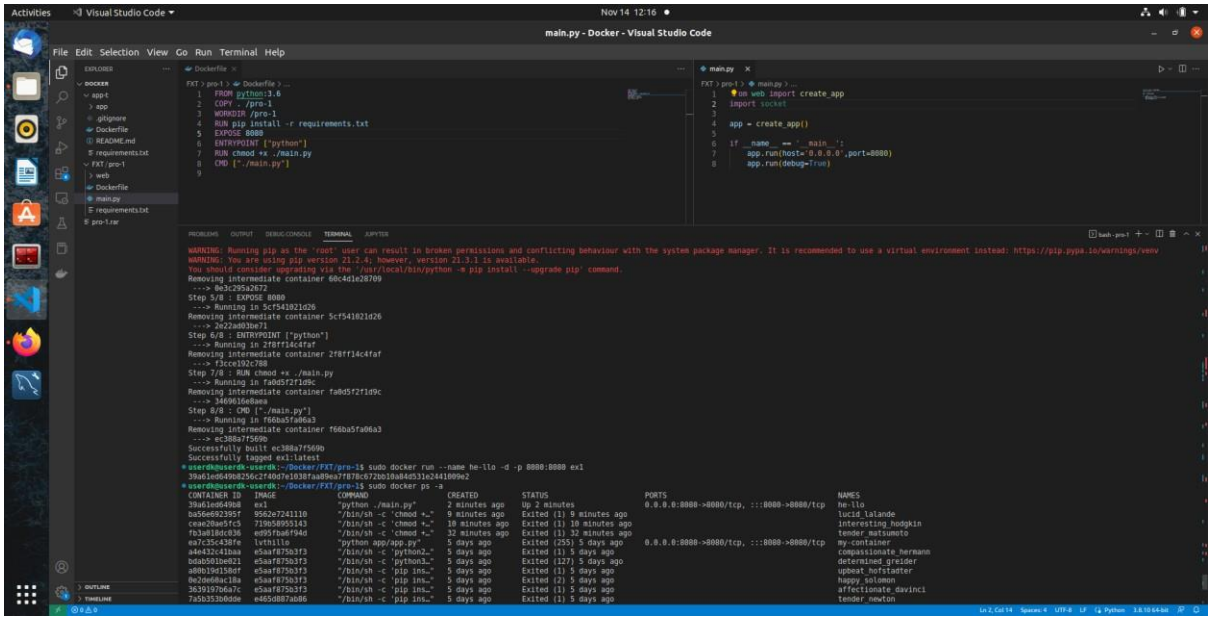

GoShoping
Home
T Shirt
Shirt
Jean
About
GoShoping
Another reason why GoShoping is the best of all online stores is the complete convenience that it offers. You can view your favourite brands with price options for different products in one place. Comprehensive size charts, product information and high-resolution images help you make the best buying decisions.
Shop Assistant
Hi, I'm looking for a dress for a wedding.
What style are you looking for?
Something elegant.
Something elegant.
We have a few options over here. This one is a classic A-line dress with a beaded bodice.
That looks nice
show something different
Built with IBM Watson®

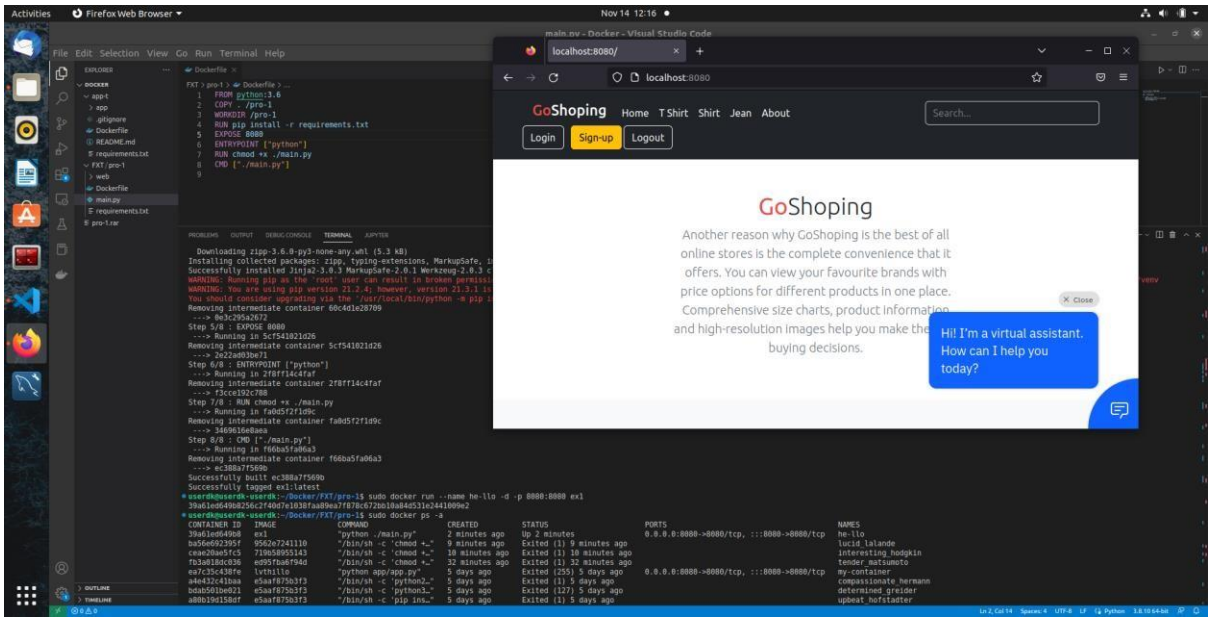
7.2 Database Schema

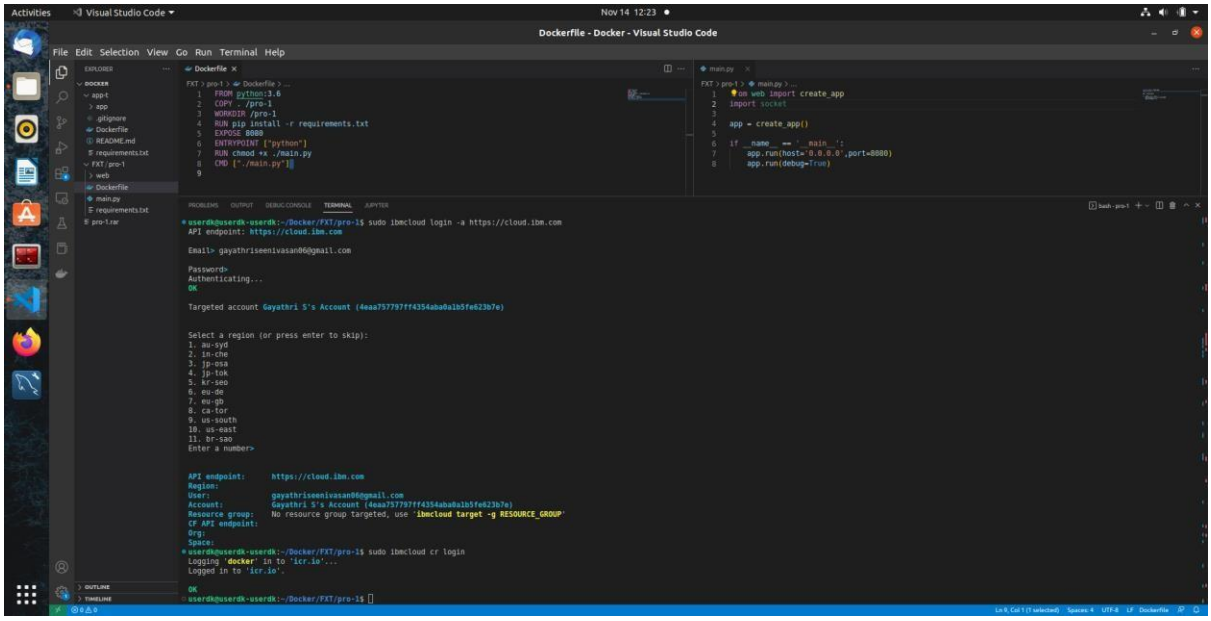
We use docker to store our images

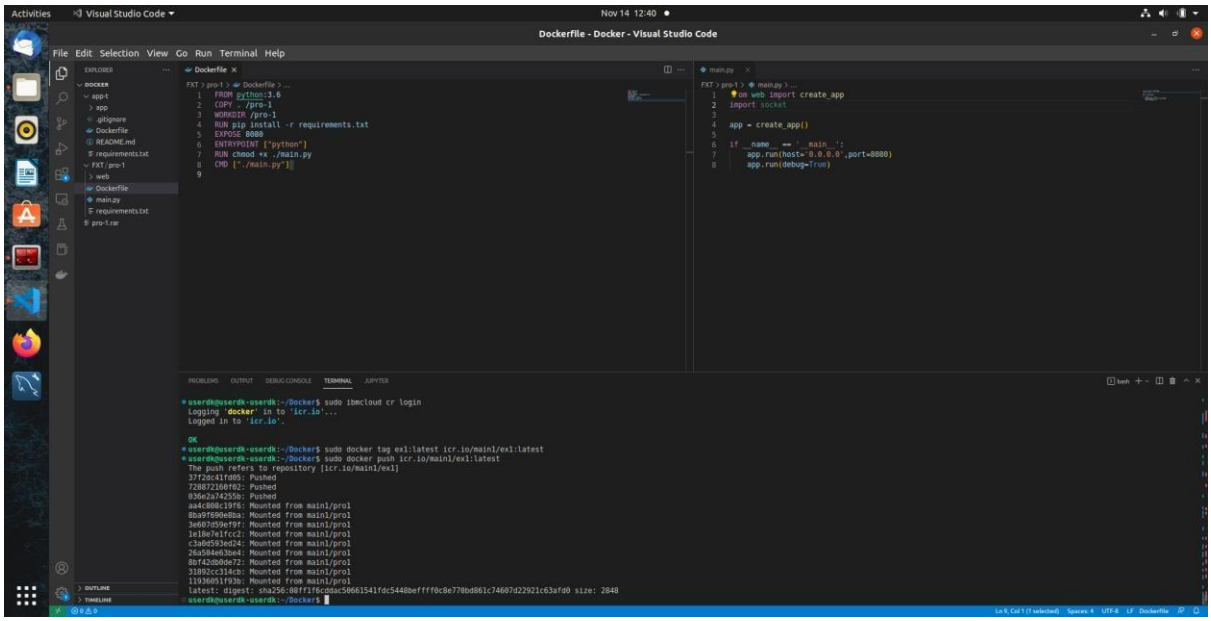





Activities
Visual Studio Code
Nov 14 12:16
main.py - Docker - Visual Studio Code
File Edit Selection View Go Run Terminal Help
FROM python:3.6
COPY . /pro-1
WORKDIR /pro-1
RUN pip install -r requirements.txt
EXPOSE 8080
ENTRYPOINT ["python"]
RUN chmod +x ./main.py
CMD ["./main.py"]
from web import create_app
import socket
app = create_app()
if __name__ == '__main__':
app.run(host='0.0.0.0',port=8080)
app.run(debug=True)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 21.2.4; however, version 21.3.1 is available.
You should consider upgrading via the '/usr/local/bin/python -m pip install --upgrade pip' command.
Removing intermediate container 60c4d1e28709
---> 0e3c295a2672
Step 5/8 : EXPOSE 8080
---> Running in 5cf541021d26
Removing intermediate container 5cf541021d26
---> 2e22ad03be71
Step 6/8 : ENTRYPOINT ["python"]
---> Running in 2f8ff14c4faf
Removing intermediate container 2f8ff14c4faf
---> f3cce192c788
Step 7/8 : RUN chmod +x ./main.py
---> Running in fa0d5f2f1d9c
Removing intermediate container fa0d5f2f1d9c
---> 3469616e8aea
Step 8/8 : CMD ["./main.py"]
---> Running in f66ba5fa06a3
Removing intermediate container f66ba5fa06a3
---> ec388a7f569b
Successfully built ec388a7f569b
Successfully tagged ex1:latest
userdk@userdk-userdk:~/Docker/FXT/pro-1$ sudo docker run --name he-llo -d -p 8080:8080 ex1
39a61ed649b8256c2f40d7e1038faa89ea7f878c672bb10a84531e2441009e2
userdk@userdk-userdk:~/Docker/FXT/pro-1$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
39a61ed649b8 ex1 "python ./main.py" 2 minutes ago Up 2 minutes 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp he-llo
ba56e692395f 9562e7241110 "/bin/sh -c 'chmod +…" 9 minutes ago Exited (1) 9 minutes ago lucid_lalande
ceae20ae5fc5 710b58955143 "/bin/sh -c 'chmod +…" 18 minutes ago Exited (1) 18 minutes ago interesting_hodgkin
fb3a018dc036 ed95fba6f94d "/bin/sh -c 'chmod +…" 32 minutes ago Exited (1) 32 minutes ago tender_matsumoto
ea7c35c438fe lvthillo "python app/app.py" 5 days ago Exited (255) 5 days ago 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp my-container
a4e432c41baa e5aaf875b3f3 "/bin/sh -c 'python2…" 5 days ago Exited (1) 5 days ago compassionate_hermann
bdab501be021 e5aaf875b3f3 "/bin/sh -c 'python3…" 5 days ago Exited (127) 5 days ago determined_greider
a80b19d158df e5aaf875b3f3 "/bin/sh -c 'pip ins…" 5 days ago Exited (1) 5 days ago upbeat_hofstadter
0e2de6ac18a e5aaf875b3f3 "/bin/sh -c 'pip ins…" 5 days ago Exited (2) 5 days ago happy_solomon
3639179b6a7c e5aaf875b3f3 "/bin/sh -c 'pip ins…" 5 days ago Exited (1) 5 days ago affectionate_davinci
7a5b353b0dde e465d887ab86 "/bin/sh -c 'pip ins…" 5 days ago Exited (1) 5 days ago tender_newton


Activities
Firefox Web Browser
Nov 14 12:16
localhost:8080/
localhost:8080
GoShoping
Home T Shirt Shirt Jean About
Search...
Login
Sign-up
Logout
GoShoping
Another reason why GoShoping is the best of all online stores is the complete convenience that it offers. You can view your favourite brands with price options for different products in one place. Comprehensive size charts, product information and high-resolution images help you make the buying decisions.
Close
Hi! I'm a virtual assistant. How can I help you today?





Source Code:

Index.html

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="description" content="">
<meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap
```

```
contributors">
<meta name="generator" content="Hugo 0.104.2">
<title>Home</title>
<link rel="stylesheet" href="{{
url_for('static', filename='css/style.css') }}">
<!-- CSS only -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrapicons@
1.9.1/font/bootstrap-icons.css">
<link rel="canonical"
href="https://getbootstrap.com/docs/5.2/examples/album/">
<!-- <title>{% block title %}Home{% endblock %}</title> -->
<!-- <div class="container">{% block content %} {% endblock %}</div> -->
<link href="../assets/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
.bd-placeholder-img {
font-size: 1.125rem;
text-anchor: middle;
-webkit-user-select: none;
-moz-user-select: none;

user-select: none;
}
@media (min-width: 768px) {
.bd-placeholder-img-lg {
font-size: 3.5rem;
}
}
.b-example-divider {
height: 3rem;
background-color: rgba(0, 0, 0, .1);
border: solid rgba(0, 0, 0, .15);
border-width: 1px 0;
box-shadow: inset 0 .5em 1.5em rgba(0, 0, 0, .1), inset 0 .125em .5em
rgba(0, 0, 0, .15);
}
.b-example-vr {
flex-shrink: 0;
width: 1.5rem;
height: 100vh;
}
.bi {
vertical-align: -.125em;
fill: currentColor;
}
.nav-scroller {
```

```
position: relative;
z-index: 2;
height: 2.75rem;
overflow-y: hidden;
}
.nav-scroller .nav {
display: flex;
flex-wrap: nowrap;
padding-bottom: 1rem;
margin-top: -1px;
overflow-x: auto;
text-align: center;
white-space: nowrap;
-webkit-overflow-scrolling: touch;
}
*{
margin: 0;
padding: 0;
box-sizing: border-box;
}
img{
width: 100%;
height: 450px;
object-fit: cover;
object-position: 50% 0;
}
</style>
</head>
<body>
<header class="p-3 text-bg-dark">
<div class="container">
<div class="d-flex flex-wrap align-items-center justify-content-center
justify-content-lg-start">
<a href="/" class="d-flex align-items-center mb-2 mb-lg-0 text-white
text-decoration-none">
<h4 class="px-3"><span
style="color:#eb4d4b;">Go</span>Shoping</h4>
<!-- <svg class="bi me-2" width="40" height="32" role="img" arialabel="
Bootstrap"><use xlink:href="#bootstrap"></use></svg> -->
</a>
<ul class="nav col-12 col-lg-auto me-lg-auto mb-2 justify-contentcenter
mb-md-0">
<li><a href="/" class="nav-link px-2 text-white"> Home</a></li>
<li><a href="#tshirt" class="nav-link px-2 text-white">T
Shirt</a></li>
<li><a href="#formal shit" class="nav-link px-2 textwhite">
Shirt</a></li>
<li><a href="#" class="nav-link px-2 text-white">Jean</a></li>
<li><a href="#" class="nav-link px-2 text-white">About</a></li>
</ul>
```

```
<form class="col-12 col-lg-auto mb-3 mb-lg-0 me-lg-3" role="search">
<input type="search" class="form-control form-control-dark text-bgdark"
placeholder="Search..." aria-label="Search">
</form>
<div class="text-end">
<a href="/login"><button type="button" class="btn btn-outline-light
me-2">Login</button></a>
<a href="/signup"><button type="button" class="btn btnwarning">
Sign-up</button></a>
<a href="/login"><button type="button" class="btn btn-outline-light
me-2">Logout</button></a>
</div>
<div class="dropdown text-end mx-3">
<a href="#" class="d-block link-dark text-decoration-none dropdowntoggle"
data-bs-toggle="dropdown" aria-expanded="false">
<!-- <img src="https://github.com/mdo.png" alt="mdo" width="32"
height="32" class="rounded-circle"> -->
</a>
<ul class="dropdown-menu text-small" >
<li><a class="dropdown-item" href="#">Profile</a></li>
<li><a class="dropdown-item" href="#">Dashboard</a></li>
<li><hr class="dropdown-divider"></li>
<li><a class="dropdown-item" href="#">Sign out</a></li>
</ul>
</div>
</div>
</div>
</header>
<main>
<section class="py-1 text-center container">
<div class="row py-lg-5">
<div class="col-lg-6 col-md-8 mx-auto">
<h1 class="fw-light"><span style="color:#eb4d4b;font-weight:
400;">Go</span>Shoping</h1>
<p class="lead text-muted">Another reason why GoShoping is the
best of all online stores is the complete convenience that it offers. You can
view your favourite brands with price options for different products in one
place. Comprehensive size charts, product information and high-resolution
images help you make the best buying decisions.</p>
<!-- <a href="#" class="btn btn-primary my-2">Main call to
action</a>
<a href="#" class="btn btn-secondary my-2">Secondary
action</a> -->
</p>
</div>
</div>
</section>
<!-- <div class="card" style="width: 18rem;">
<img src="..." class="card-img-top" alt="...">
<div class="card-body">
```

```
<h5 class="card-title">Card title</h5>
<p class="card-text">Some quick example text to build on the
card title and make up the bulk of the card's content.</p>
<a href="#" class="btn btn-primary">Go somewhere</a>
</div>
</div> -->
<div class="album py-5 bg-light">
<div class="container">
<div class="row row-cols-1 row-cols-sm-1 row-cols-sm-3 g-4">
<div class="col">
<div class="card shadow-sm" id="tshirt">
<img src="{{ url_for('static', filename='clothimg/1.jpg')
}}" alt="">
<div class="card-body">
<h6>T Shirt</h6>
<p class="card-text">
Symbol Men's Regular Polo Shirt</p>
<div class="d-flex justify-content-between align-itemscenter">
<div class="btn-group">
<button type="button" class="btn btn-sm btn-outlinesecondary">$
9</button>
<button type="button" class="btn btn-sm btn-outlinesecondary"><
s>($15)</s> 27% off</button>
</div>
<small class="text-muted">Prime</small>
</div>
</div>
</div>
</div>
<div class="col">
<div class="card shadow-sm"><img src="https://imagesdo.
nyc3.cdn.digitaloceanspaces.com/lAVtCJXFVr/product_images/1618468852.BKT005
.jpeg" alt="Khankudi's Zigzag Printed Premium Tencel Kurta Shirt" class="imgf-
con">
<div class="card-body">
<h6>Shirt</h6>
<p class="card-text">Men Slim Fit Striped Spread Collar
Casual Shirt</p>
<div class="d-flex justify-content-between align-itemscenter">
<div class="btn-group">
<button type="button" class="btn btn-sm btn-outlinesecondary">$
12</button>
<button type="button" class="btn btn-sm btn-outlinesecondary"><
s>($19)</s> 15% off</button>
</div>
<small class="text-muted">Prime</small>
</div>
</div>
</div>
</div>
```

```
<div class="col">
<div class="card shadow-sm">
<img class="img-alignment"
src="https://d1pdzcnm6xgxlz.cloudfront.net/bottoms/8905074715749-9.jpg">
<div class="card-body">
<h6>Jean</h6>
<p class="card-text">Fit Men Jeans</p>
<div class="d-flex justify-content-between align-itemscenter">
<div class="btn-group">
<button type="button" class="btn btn-sm btn-outlinesecondary">$
17</button>
<button type="button" class="btn btn-sm btn-outlinesecondary"><
s>($26)</s> 15% off</button>
</div>
<!-- <small class="text-muted">9 mins</small> -->
</div>
</div>
</div>
</div>
<div class="col">
<div class="card shadow-sm">
<img class="_2r_T1I _396QI4" alt=""
src="https://rukminim1.flixcart.com/image/832/832/xif0q/cargo/4/q/y/xxlbblcargo-
p17-blive-original-imaggsnaur8ptuee.jpeg?q=70">
<div class="card-body">
<h6>Jogger</h6>
<p class="card-text">Jogger Fit Men Grey Jeans</p>
<div class="d-flex justify-content-between align-itemscenter">
<div class="btn-group">
<button type="button" class="btn btn-sm btn-outlinesecondary">$
15</button>
<button type="button" class="btn btn-sm btn-outlinesecondary"><
s>($21)</s> 15% off</button>
</div>
<small class="text-muted">Prime</small>
</div>
</div>
</div>
</div>
<div class="col">
<div class="card shadow-sm">
<img class="_2r_T1I _396QI4" alt=""
src="https://rukminim1.flixcart.com/image/832/832/knrsjgw0/short/l/g/y/30-
hlv8000515-highlander-original-imag2dhdcw88vx5k.jpeg?q=70">
<div class="card-body">
<h6>Men Shorts</h6>
<p class="card-text">Solid Men Shorts, Regular Shorts</p>
<div class="d-flex justify-content-between align-itemscenter">
<div class="btn-group">
<button type="button" class="btn btn-sm btn-outlinesecondary">$
```

```
8</button>
<button type="button" class="btn btn-sm btn-outlinesecondary"><
s>($11)</s> 15% off</button>
</div>
<small class="text-muted"></small>
</div>
</div>
</div>
</div>
<div class="col">
<div class="card shadow-sm">
<img class="_2r_T1I _396QI4" alt=""
src="https://rukminim1.flixcart.com/image/832/832/ksuowi80/shirt/w/p/o/xxlfrml-
st2-vebnor-original-imag6bkegkwmqv75.jpeg?q=70">
<div class="card-body">
<h6>Formal Shirt</h6>
<p class="card-text">Men Slim Fit Collar Casual Shirt</p>
<div class="d-flex justify-content-between align-itemscenter">
<div class="btn-group">
<button type="button" class="btn btn-sm btn-outlinesecondary">$
11</button>
<button type="button" class="btn btn-sm btn-outlinesecondary"><
s>($19)</s> 15% off</button>
</div>
<small class="text-muted">Prime</small>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</main>
<footer class="text-muted py-5">
<div class="container">
<p class="float-end mb-1">
<a href="#">Back to top</a>
</p>
<!-- <p class="mb-1">Album example is &copy; Bootstrap, but please
download and customize it for yourself!</p>
<p class="mb-0">New to Bootstrap? <a href="/">Visit the homepage</a> or
read our <a href="../getting-started/introduction/">getting started
guide</a>.</p> -->
</div>
</footer>
<script>
window.watsonAssistantChatOptions = {
integrationID: "7106a37a-cf0c-4e54-a22c-0b49fbe2e8e2", // The ID of this
```

```
integration.
region: "au-syd", // The region your integration is hosted in.
serviceInstanceID: "3ea83de1-6380-4510-8ebc-0bab99f8c852", // The ID of
your service instance.
onLoad: function(instance) { instance.render(); }
};
setTimeout(function(){
const t=document.createElement('script');
t.src="https://webchat.
global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
document.head.appendChild(t);
});
</script>
<script src="https://code.jquery.com/jquery-3.6.1.slim.min.js"
integrity="sha256-w8CvhFs7iHNVUtnSP0YKEg00p9Ih13rlL9zGqvLdePA="
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
</body>
</html>
```

Signup.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>Signup</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<link
rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.m
in.css"
integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous"
/>
<link
rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/fontawesome.
```

```
min.css"
crossorigin="anonymous"
/>
<link rel="stylesheet" href="{{
url_for('static', filename='css/signup.css') }}">
<style>
.bd-placeholder-img {
font-size: 1.125rem;
text-anchor: middle;
-webkit-user-select: none;
-moz-user-select: none;
user-select: none;
}
@media (min-width: 768px) {
.bd-placeholder-img-lg {
font-size: 3.5rem;
}
}
.b-example-divider {
height: 3rem;
background-color: rgba(0, 0, 0, .1);
border: solid rgba(0, 0, 0, .15);
border-width: 1px 0;
box-shadow: inset 0 .5em 1.5em rgba(0, 0, 0, .1), inset 0 .125em .5em
rgba(0, 0, 0, .15);
}
.b-example-vr {
flex-shrink: 0;
width: 1.5rem;
height: 100vh;
}
.bi {
vertical-align: -.125em;
fill: currentColor;
}
.nav-scroller {
position: relative;
z-index: 2;
height: 2.75rem;
overflow-y: hidden;
}
.nav-scroller .nav {
display: flex;
flex-wrap: nowrap;
padding-bottom: 1rem;
margin-top: -1px;
overflow-x: auto;
text-align: center;
white-space: nowrap;
-webkit-overflow-scrolling: touch;
```

```
}
</style>
</head>
<body>
<header class="p-3 text-bg-dark">
<div class="container">
<div class="d-flex flex-wrap align-items-center justify-content-center
justify-content-lg-start">
<a href="/" class="d-flex align-items-center mb-2 mb-lg-0 text-white
text-decoration-none">
<h4 class="px-3"><span style="color:#eb4d4b;">Go</span>Shoping</h4>
</a>
<ul class="nav col-12 col-lg-auto me-lg-auto mb-2 justify-contentcenter
mb-md-0">
<!-- <li><a href="#" class="nav-link px-2 text-white">Home</a></li>
<li><a href="#" class="nav-link px-2 text-white">Features</a></li>
<li><a href="#" class="nav-link px-2 text-white">Pricing</a></li>
<li><a href="#" class="nav-link px-2 text-white">FAQs</a></li>
<li><a href="#" class="nav-link px-2 text-white">About</a></li> -->
</ul>
<!-- <form class="col-12 col-lg-auto mb-3 mb-lg-0 me-lg-3"
role="search">
<input type="search" class="form-control form-control-dark text-bgdark"
placeholder="Search..." aria-label="Search">
</form> -->
<div class="text-end">
<!-- <button type="button" class="btn btn-outline-light me-
2">Login</button> -->
<a href="/login"><button type="button" class="btn btnwarning">
Login</button></a>
</div>
</div>
</div>
</header>
{% with messages = get_flashed_messages(with_categories=true) %} {% if
messages %} {% for category, message in messages %} {% if category ==
'error' %}
<div class="alert alert-danger alter-dismissable fade show" role="alert">
{{ message }}
<button type="button" class="close" data-dismiss="alert">
<span aria-hidden="true">&times;</span>
</button>
</div>
{% else %}
<div class="alert alert-success alter-dismissable fade show" role="alert">
{{ message }}
<button type="button" class="close" data-dismiss="alert">
<span aria-hidden="true">&times;</span>
</button>
</div>
```

```
{% endif %} {% endfor %} {% endif %} {% endwith %}
<main class="form-signin w-100 text-center">
<form method="post" >
<!-- <img class="mb-4" src="/docs/5.2/assets/brand/bootstrap-logo.svg"
alt="" width="72" height="57"> -->
<h1 class="h3 mt-4 mb-5 fw-normal">Please SIGN UP</h1>
<div class="form-floating">
<input type="email" name="email" class="form-control"
id="floatingInput" placeholder="name@example.com">
<label for="floatingPassword">Email</label>
</div>
<div class="form-floating">
<input type="Username" name="username" class="form-control"
id="floatingPassword" placeholder="Username">
<label for="floatingPassword">Username</label>
</div>
<div class="form-floating">
<input type="password" name="password" class="form-control"
id="floatingPassword" placeholder="Password">
<label for="floatingPassword">Password</label>
</div>
<div class="form-floating">
<input type="phone" name="phone" class="form-control"
id="floatingPassword" placeholder="Phone">
<label for="floatingPassword">Phone</label>
</div>
<div class="checkbox mt-4 mb-4">
<label>
<input type="checkbox" value="remember-me"> Remember me
</label>
</div>
<button class="btn1" type="sumit"><span>Sign up</span></button>
<p class="mt-5 mb-3 text-white">© 2022–2023</p>
</form>
</main>
<script
src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"
></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.mi
n.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"
></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
```

```
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"
></script>
</body>
</html>
```

Login.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>Login</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<link rel="stylesheet" href="{{
url_for('static', filename='css/login.css') }}">
<style>
.bd-placeholder-img {
font-size: 1.125rem;
text-anchor: middle;
-webkit-user-select: none;
-moz-user-select: none;
user-select: none;
}
@media (min-width: 768px) {
.bd-placeholder-img-lg {
font-size: 3.5rem;
}
}
.b-example-divider {
height: 3rem;
background-color: rgba(0, 0, 0, .1);
border: solid rgba(0, 0, 0, .15);
border-width: 1px 0;
box-shadow: inset 0 .5em 1.5em rgba(0, 0, 0, .1), inset 0 .125em .5em
rgba(0, 0, 0, .15);
}
.b-example-vr {
flex-shrink: 0;
width: 1.5rem;
height: 100vh;
}
.bi {
vertical-align: -.125em;
```

```
fill: currentColor;
}
.nav-scroller {
position: relative;
z-index: 2;
height: 2.75rem;
overflow-y: hidden;
}
.nav-scroller .nav {
display: flex;
flex-wrap: nowrap;
padding-bottom: 1rem;
margin-top: -1px;
overflow-x: auto;
text-align: center;
white-space: nowrap;
-webkit-overflow-scrolling: touch;
}
</style>
</head>
<body>
<header class="p-3 text-bg-dark">
<div class="container">
<div class="d-flex flex-wrap align-items-center justify-content-center
justify-content-lg-start">
<a href="/" class="d-flex align-items-center mb-2 mb-lg-0 text-white
text-decoration-none">
<h4 class="px-3"><span style="color:#eb4d4b;">Go</span>Shoping</h4>
</a>
<ul class="nav col-12 col-lg-auto me-lg-auto mb-2 justify-contentcenter
mb-md-0">
<!-- <li><a href="#" class="nav-link px-2 text-white">Home</a></li>
<li><a href="#" class="nav-link px-2 text-white">Features</a></li>
<li><a href="#" class="nav-link px-2 text-white">Pricing</a></li>
<li><a href="#" class="nav-link px-2 text-white">FAQs</a></li>
<li><a href="#" class="nav-link px-2 text-white">About</a></li> -->
</ul>
<!-- <form class="col-12 col-lg-auto mb-3 mb-lg-0 me-lg-3"
role="search">
<input type="search" class="form-control form-control-dark text-bgdark"
placeholder="Search..." aria-label="Search">
</form> -->
<div class="text-end">
<!-- <button type="button" class="btn btn-outline-light me-
2">L</button> -->
<a href="/signup"><button type="button" class="btn btnwarning">
Sign-up</button></a>
</div>
</div>
</div>
```

```
</header>
<main class="form-signin w-100 text-center">
<form method="post">
<!-- <img class="mb-4" src="/docs/5.2/assets/brand/bootstrap-logo.svg"
alt="" width="72" height="57"> -->
<h1 class="h3 mt-4 mb-5 fw-normal">LOGIN </h1>
<div class="form-floating">
<input type="Username" name="username" class="form-control"
id="floatingInput" placeholder="name@example.com">
<label for="floatingPassword">Email or Username</label>
</div>
<div class="form-floating">
<input type="password" name="password" class="form-control"
id="floatingPassword" placeholder="Password">
<label for="floatingPassword">Password</label>
</div>
<div class="checkbox mt-4 mb-4">
<label>
<input type="checkbox" value="remember-me"> Remember me
</label>
</div>
<a href="/"><button class="btn1"
type="button"><span>Login</span></button></a></form>
<p class="mt-5 mb-3 text-white">© 2022–2023</p>
</form>
</main>
</body>
</html>
```

Home.html:

```
<!-- {% extends "index.html" %}{% block title %}Home{% endblock %} -->
Static:
Login.css:
/* html,
body {
height: 100%;
} */
body {
/* display: flex;
align-items: center;
padding-top: 40px;
padding-bottom: 40px; */
background-image: url(../images/3.jpg);
background-color: #cccccc;
background-position: center;
background-repeat: no-repeat;
background-size: cover;
}
.form-signin {
```

```
width: 100%;
margin: auto;
max-width: 390px;
padding: 15px;
margin-top: 100px;
backdrop-filter: blur(2px) saturate(180%);
-webkit-backdrop-filter: blur(13px) saturate(180%);
background-color: rgba(17, 25, 40, 0.75);
border-radius: 12px;
border: 1px solid rgba(255, 255, 255, 0.125);
}
.form-floating .form-control{
font-weight:bold;
color: black;
}
.form-signin .checkbox {
font-weight: 400;
}
.form-signin .form-floating{
margin-left: 25px;
margin-right: 25px;
}
.form-signin .form-floating:focus-within {
z-index: 2;
}
.form-signin input[type="name"] {
margin-bottom: -1px;
border-bottom-right-radius: 0;
border-bottom-left-radius: 0;
opacity: 0.5;
}
.form-signin input[type="username"] {
margin-bottom: -1px;
/* border-top-left-radius: 0;
border-top-right-radius: 0; */
border-bottom-right-radius: 0;
border-bottom-left-radius: 0;
opacity: 0.5;
}
/* .form-signin input[type="password"] {
margin-bottom: -1px;
border-bottom-right-radius: 0;
border-bottom-left-radius: 0;
border-top-left-radius: 0;
border-top-right-radius: 0;
opacity: 0.5;
} */
.form-signin input[type="password"] {
margin-bottom: 10px;
border-top-right-radius: 0;
```

```
border-top-left-radius: 0;
opacity: 0.5;
}
h1{
color:white;
}
label[for=floatingPassword]{
color:black;
font-weight: 500;
}
.checkbox label{
color:white;
font-weight:100;
}
input[type="name"]:focus,
input[type="password"]:focus,
input[type="phone"]:focus,
input[type="username"]:focus{
border: 1px solid white ;
box-shadow: 0 9px 9px rgb(255, 255, 255) inset, 0 0 11px rgb(255, 255,
255);
outline: 0 none;
}
.btn1 {
display: inline-block;
border-radius: 4px;
background-color: #3d405b;
border: none;
color: #FFFFFF;
text-align: center;
font-size: 17px;
padding: 11px;
width: 220px;
transition: all 0.5s;
cursor: pointer;
margin: 3px;
}
.btn1 span {
cursor: pointer;
display: inline-block;
position: relative;
transition: 0.5s;
}
.btn1 span:after {
content: '»';
position: absolute;
opacity: 0;
top: 0;
right: -15px;
transition: 0.5s;
```

```
}
.btn1:hover span {
padding-right: 15px;
}
.btn1:hover span:after {
opacity: 1;
right: 0;
}
```

Signup.css:

```
/* html,

body {

height: 100%;

} */

body {

/* display: flex;

align-items: center;

padding-top: 40px;

padding-bottom: 40px; */

background-image: url(../images/4.jpg);

background-color: #cccccc;

background-position: center;

background-repeat: no-repeat;

background-size: cover;

}

.form-signin {

width: 100%;

margin: auto;

max-width: 390px;
```

```
padding: 15px;
margin-top: 70px;
backdrop-filter: blur(2px) saturate(180%);
-webkit-backdrop-filter: blur(13px) saturate(180%);
background-color: rgba(17, 25, 40, 0.75);
border-radius: 12px;
border: 1px solid rgba(255, 255, 255, 0.125);
}
.form-floating .form-control{
font-weight:bold;
color: black;
}
.form-signin .checkbox {
font-weight: 400;
}
.form-signin .form-floating{
margin-left: 25px;
margin-right: 25px;
}
.form-signin .form-floating:focus-within {
z-index: 2;
}
.form-signin input[type="email"] {
margin-bottom: -1px;
```

```
border-bottom-right-radius: 0;
border-bottom-left-radius: 0;
opacity: 0.5;
}
.form-signin input[type="username"] {
margin-bottom: -1px;
border-top-left-radius: 0;
border-top-right-radius: 0;
border-bottom-right-radius: 0;
border-bottom-left-radius: 0;
opacity: 0.5;
}
.form-signin input[type="password"] {
margin-bottom: -1px;
border-bottom-right-radius: 0;
border-bottom-left-radius: 0;
border-top-left-radius: 0;
border-top-right-radius: 0;
opacity: 0.5;
}
.form-signin input[type="phone"] {
margin-bottom: 10px;
border-top-right-radius: 0;
border-top-left-radius: 0;
```

```
opacity: 0.5;
}
h1{
color:white;
}
label[for=floatingPassword]{
color:black;
font-weight: 500;
}
.checkbox label{
color:white;
font-weight:100;
}
input[type="email"]:focus,
input[type="password"]:focus,
input[type="phone"]:focus,
input[type="username"]:focus{
border: 1px solid white ;
box-shadow: 0 9px 9px rgb(255, 255, 255) inset, 0 0 11px rgb(255, 255,
255);
outline: 0 none;
}
.btn1 {
display: inline-block;
```

```
border-radius: 6px;
background-color: #3d405b;
border: none;
color: #FFFFFF;
text-align: center;
font-size: 20px;
padding: 8px;
width: 220px;
transition: all 0.5s;
cursor: pointer;
margin: 3px;
}
.btn1 span {
cursor: pointer;
display: inline-block;
position: relative;
transition: 0.5s;
}
.btn1 span:after {
content: '»';
position: absolute;
opacity: 0;
top: 0;
right: -15px;
```

```
transition: 0.5s;
}
.btn1:hover span {
padding-right: 15px;
}
.btn1:hover span:after {
opacity: 1;
right: 0;
}
```

Style.css:

```
*{
margin: 0;
padding: 0;
box-sizing: border-box;
}
img{
width: 100%;
height: 450px;
object-fit: cover;
object-position: 50% 0;
}
```

Database.py:

```
import mysql.connector
import json
```

```
class Database:
@staticmethod
def database_Connection(self):
with open('cred.json') as file:
credentials = json.load(file)
self.mydb = mysql.connector.connect(
host = credentials['host'],
username = credentials['username'],
password = credentials['password'],
database = credentials['db']
)
return self.mydb
def del(self):
self.mycursor.close()
self.mydb.close()
```

user.py:

```
from database import Database
class User:
def __init__(self):
self.mydb = Database.database_Connection(self)
self.mycursor = self.mydb.cursor()
# self.username = username
# self.password = password
# self.email = email
```

```
# self.phone = phone
def Signup(self,username , password , email , phone):
sql = "INSERT INTO AUTH (EMAIL, USERNAME , PASSWORD , PHONE) VALUES
(%s,%s,%s, %s)"
val = (email, username, password,phone)
self.mycursor.execute(sql, val)
self.mydb.commit()
__init__.py:
from flask import Flask
def create_app():
app= Flask(__name__)
app.config['SECRET_KEY'] = 'fersdvdfvv'
from .views import views
from .auth import auth
app.register_blueprint(views,url_prefix='/')
app.register_blueprint(auth,url_prefix='/')
return app
auth.py:
from flask import Blueprint,render_template,request,flash
# from lib.user import User
# import mysql.conne
# import json
# def database_Connection(self):
# with open('cred.json') as file:
```

```
# credentials = json.load(file)
# mydb = mysql.connector.connect(
# host = credentials['host'],
# username = credentials['username'],
# password = credentials['password'],
# database = credentials['db']
# )
# return mydb
# # def del(self):
# # self.mycursor.close()
# # self.mydb.close()
# def Signup(username , password , email , phone):
# sql = "INSERT INTO AUTH (EMAIL, USERNAME , PASSWORD , PHONE) VALUES
(%s,%s,%s, %s)"
# val = (email,username,password, phone)
# mycursor.execute(sql, val)
# mydb.commit()
# #Database Conn
auth = Blueprint('auth',__name__)
@auth.route('/login',methods=['GET','POST'])
def login():
return render_template("login.html")
@auth.route('/logout')
def logout():
```

```
return "<p>logout</p>"

@auth.route('/signup',methods=['GET','POST'])

def signup():

if request.method == 'POST':

email = request.form.get('email')

username = request.form.get('username')

password = request.form.get('password')

phone = request.form.get('phone')

# user = User.query.filter_by(email=email).first()

# if user:

# flash('Email already exists.', category='error')

if len(email) < 4:

flash('Email must be greater than 3 characters.',

category='error')

elif len(username) < 2:

flash('User name must be greater than 1 character.',

category='error')

elif len(password) < 7:

flash('Password must be at least 7 characters.', category='error')

else:

# u = User()

# u.Signup(username,password,email,phone)

flash('Account created!', category='success')

return render_template("signup.html")
```

{} cred.json:

```
{
"host" : "fbd88901-ebdb-4a4f-a32e-
9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731",
"username" : "dww40023",
"password" : "kl9VrHdh29NdsjtI",
"db" : "DWW40023"
}
```

Views.py:

```
from flask import Blueprint,render_template
views = Blueprint('views',__name__)
@views.route('/')
def home():
return render_template("home.html")
```

main.py:

```
from web import create_app
app = create_app()
if __name__ == '__main__':
app.run(debug=True)
```

chatbot:

```
<script>
window.watsonAssistantChatOptions = {
integrationID: "7106a37a-cf0c-4e54-a22c-0b49fbe2e8e2", // The ID of this
integration.
```

```
region: "au-syd", // The region your integration is hosted in.
serviceInstanceID: "3ea83de1-6380-4510-8ebc-0bab99f8c852", // The ID of
your service instance.
onLoad: function(instance) { instance.render(); }
};
setTimeout(function(){
const t=document.createElement('script');
t.src="https://webchat.
global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
document.head.appendChild(t);
});
</script
```

8. RESULTS:

8.1 Performance Metrics:

1. 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).

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Severity 5
By Design	8	3	2	2	15
Duplicate	1	0	2	0	3
External	1	2	0	2	5
Fixed	10	1	3	18	32
Not Reproduced	0	0	2	0	2
Skipped	0	0	2	2	4
Won't Fix	0	5	1	1	7
Totals	20	11	12	25	68

3. Test Case Analysis

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

Section	Total cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	46	0	0	46
Security	2	0	0	2
Outsource Shipping	2	0	0	2
Exception Reporting	7	0	0	7
Final Report Output	3	0	0	3
Version Control	4	0	0	4

9. ADVANTAGE AND DISADVANTAGE:

Advantage:

Convert Shoppers to Customers:

Converting shoppers into customers takes a special touch. Personalized interactions from a ChatBot show your customer that he or she is valued as an individual, in turn, engendering long-term loyalty.

Increase Average Order Value:

Average order values typically go up when an engine is leveraged to display personalized options as shoppers are more willing to spend generously on items they thoroughly covet.

Increase Number of Items per Order:

In addition to the average order value rising, the number of items per order also typically rises when an engine is employed. When the customer is shown options that meet his or her interest, they are far more likely to add items to to their active purchase cart.

Disadvantage:

- Recommendations can also be the result of personal choice and taste. So, something you love may not go well with their former employer. Therefore, you may get a negative recommendation from your former employer.
- The former employer may also not be thrilled with the employee's departure. Therefore, they can try to sabotage her employment.
- The former employer may not be completely honest about the employee. In some cases, the employer may be difficult to satisfy, which may reflect in the letter of recommendation. However, that does not mean that helper is not good.

10. 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. 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.

11. FUTURE SCOPE:

Forecasting:

The majority of fashion recommendations are focused on predicting the "present" based on previous interactions and trends, i.e., what the user will do right now based on their history. How might such models be utilized to make predictions regarding the fashion trends of the future? This aspect can be linked to popularity forecasting in the fashion domain since trendy items will likely be popular.

Ethics:

What are issues of fairness and bias in fashion recommendation, e.g. to what extent are underrepresented users (plus size, race, body type, etc.) poorly served by recommendations? The ethics of "fast fashion" is also worth considering. Fast fashion is a term that refers to the manufacturing process, which means that the consumer may have to choose between purchasing something cheaper but manufactured under dubious conditions or paying more (or even having a very limited selection) for sustainable and ethically superior clothing.

Deployability:

What are the practical considerations in terms of deploying models from academia? Perhaps complexity and being naturally black-box models (relating to the preceding discussion about interpretation) are important aspects for consideration given that the majority of the techniques described are based on neural networks. Beyond these, fashion recommender systems are beginning to intersect with related domains including conversational models , models involving text, and augmented reality (virtual try on, etc.)

12. APPENDIX:

GITHUB:

https://github.com/IBM-EPBL/IBM-Project-24039-1659936030

Project Demo link:

https://www.youtube.com/watch?v=NVBNSFnzb_c