SMART FASHION RECOMMENDER

A PROJECT REPORT

Submitted by

M.ANITHA (814619104001)
M.KANMANI (814619104008)
R.MATHUMITHA (814619104013)
C.VIJAYALAKSHMI (814619104303)

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

TRICHY ENGINEERING COLLEGE, TRICHY

ANNA UNIVERSITY: CHENNAI 600 025

SEPTEMBER

SMART FASHION RECOMMENDER APPLICATION

ABSTRACT

The Online Shopping is a web based application intended for online retailers. The main objective of this application is to make it interactive and its ease of use. It would make searching, viewing and selection of a product easier. It contains a sophisticated search engine for user's to search for products specific to their needs. The search engine provides an easy and convenient way to search for products where a user can Search for a product interactively and the search engine would refine the products available based on the user’s input. The user can then view the complete specification of each product. They can also view the product reviews and also write their own reviews. The application also provides a drag and drop feature so that a user can add a product to the shopping cart by dragging the item in to the shopping cart. The main emphasis lies in providing a userfriendly search engine for effectively showing the desired results and its drag and drop behavior.

CHAPTER-1

1.INTRODUCTION

Humans are inevitably drawn towards something that is visually more attractive. This tendency of humans has led to development of fashion industry over the course of time. With introduction of recommender systems in multiple domains, retail industries are coming forward with investments in latest technology to improve their business. Fashion has been in existence since centuries and will be prevalent in the coming days as well. Women are more correlated with fashion and style, and they have a larger product base to deal with making it difficult to take decisions. It has become an important aspect of life for modern families since a person is more often than not judged based on his attire. Moreover, apparel providers need their customers to explore their entire product line so they can choose what they like the most which is not possible by simply going Humans into a cloth store.

1.1Project Overview

A recommendation system is a system that is programmed to predict future preferable items from a large set of collections. A recommendation system works either by using user preferences or by using the items most preferred by all users. The main challenge in building a fashion recommendation system is that it is a very dynamic industry. It changes very often when it comes to seasons, festivals, pandemic conditions. However, given too many options of garments on the e-commerce websites, has presented new challenges to the customers in identifying their correct outfit. Unlike the conventional systems that rely on the user's previous purchases and history, this project aims at using an image of a product given as input by the user to generate recommendations since many-a-time people see something that they are interested in and tend to look for products that are similar to that.

1.1Purpose

Clothing that monitors the wearer's physical condition. Smart shirts and body suits provide biometric data, such as pulse rate, temperature, muscle stretch, heart rhythm and physical movement, and the data are transmitted via Bluetooth to an app in real time. Clothing can insulate against cold or hot conditions, and it can provide a hygienic barrier, keeping infectious and toxic materials away from the body. It can protect feet from injury and discomfort or facilitate navigation in varied environments. Clothing also provides protection from ultraviolet radiation.

CHAPTER-2

1 LITERATURE SURVEY

TITLE	AUTHORS AND YEAR	MERITS
Fashion Recommendation System, Model And Methods	Chakraborty, S.Hoque, M.S.Jeem, N.R. Biswas, 2021	Content Based - Products recommended based on the evaluation of experienced uses. CBF does not need any information from other users, which makes this technique more feasible and less time consuming
A Review Of Modern Fashion Recommender System	Yashar Deldjoo , Polytechnic University of Bari, Italy - 2021	3D model of the human body that can be used for recommendations according to different body shapes. In fashion, it is essential to know your body shape and chooses clothes to make your body look good.

Imaged - Based Fashion Recommender System	Shaghayegh Shirkhani , Lulea University of Technology, Department of Computer Science -2021.	In this chapter, we introduce the significance of the research and the gap,our objectives,and contributions.
Content Based Apparel Recommendation System For Fashion Industry	Swathi Sambangi and Illa Pavan Kumar, Department of Information Technology-2019	The industry for apparels has a major role in the development and globalization of countries as it has been incorporated into the world economy. Fashion industry involves huge supply demand chain of designing garments and production of sales.
Recommendation System	Dr.Mohan Kubendiran and Nishal Pradhan ,School of Computer Science and Engineering -2019	Recommendation systems aim at identifying the best products and contents that suits the preference of a user.

2.1Existing Problem

A recommender system is a filtering technique which aim at predicting the preference of a user or rather how the user would rate an item and would prefer an item in the future. Let us look at an example from the online movie streaming company called Netflix. Netflix always recommends movies to users based on what he/she has watched or clicked on.

Suppose, a user likes action movies, Netflix would recommend that user action movies all the time. In the case of online shopping site called Amazon, when a user views a product, it also recommends other similar products viewed by other users to that user. Likewise, in the case of friend suggestions on Facebook, it always recommends friends based on location, user's details and history, mutual friends, etc.

Recommender system has become increasingly popular in this era which consists of technology and advancement in a number of areas, like recommending books, news articles, movies to a user, music, commercial products, restaurants, web pages, and many more. 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 their customers. Recommender systems have grown to be an essential part of all large Internet retailers, driving up to 35% of Amazon sales [103] or over 80% of the content watched on Netflix [31].

In this work 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.

Clothing is a kind of symbol that represents people's internal perceptions through their outer appearance. It conveys information about their choices, faith, personality, profession, social status, and attitude towards life. 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.

2.2Reference

1. W.-H. Cheng, S. Song, C.-Y. Chen, S. C. Hidayati, and J. Liu, “Fashion 1Meets Computer Vision: A Survey,” arXiv, Mar. 2020, [Online]. Available: http://arxiv.org/abs/2003.13988.

2. J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, “Image-Based Recommendations on Styles and Substitutes,” in Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Aug. 2015, pp. 43–52, doi: 10.1145/2766462.2767755.

3. H. Lee, J. Seol, and S. Lee, “Style2Vec: Representation Learning for Fashion Items from Style Sets,” arXiv, Aug. 2017, [Online]. Available: http://arxiv.org/abs/1708.04014.

4. X. Gu, F. Gao, M. Tan, and P. Peng, “Fashion analysis and understanding with artificial intelligence,” Inf. Process. Manag., vol. 57, no. 5, p. 102276, Sep. 2020, doi: 10.1016/j.ipm.2020.102276.

5.Prevayl Holdings Limited. Available online: https://www.prevayl.com (accessed on October 19, 2021).

2.3Problem Statement Definition

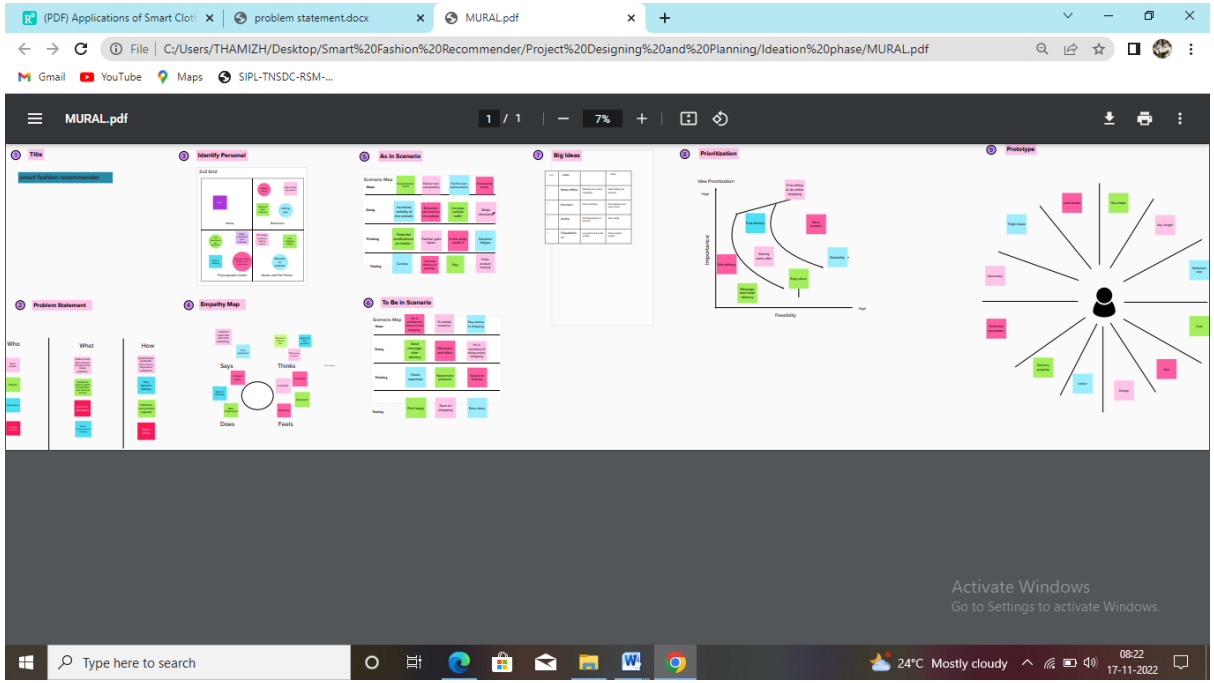
Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love. A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

Problem Statement (ps)	**I am (customer)**	**I am trying to**	**But**	**Because**	**Which makes me feel**
Ps-1	The customer	See variety of product in one place	Some fake product on different sites	Convenience	Happy
Ps-2	The customer	Ability to compare prices	Local brand	Best way to send gifts	Gloomy

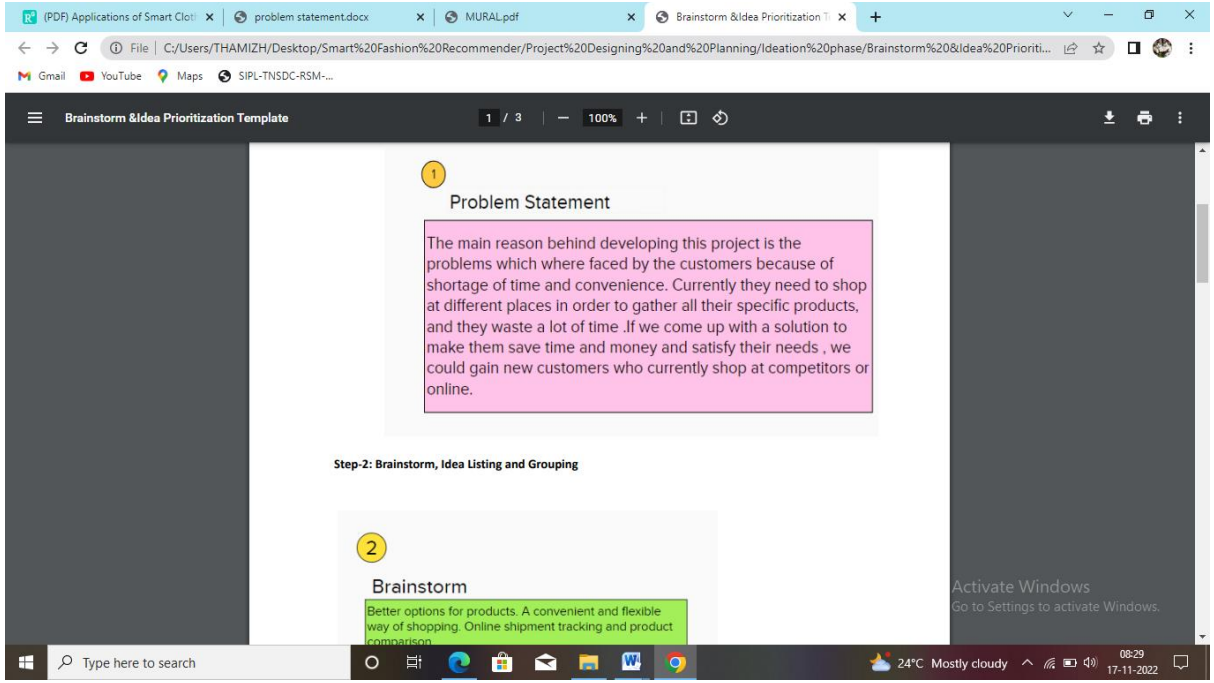
CHAPTER-3

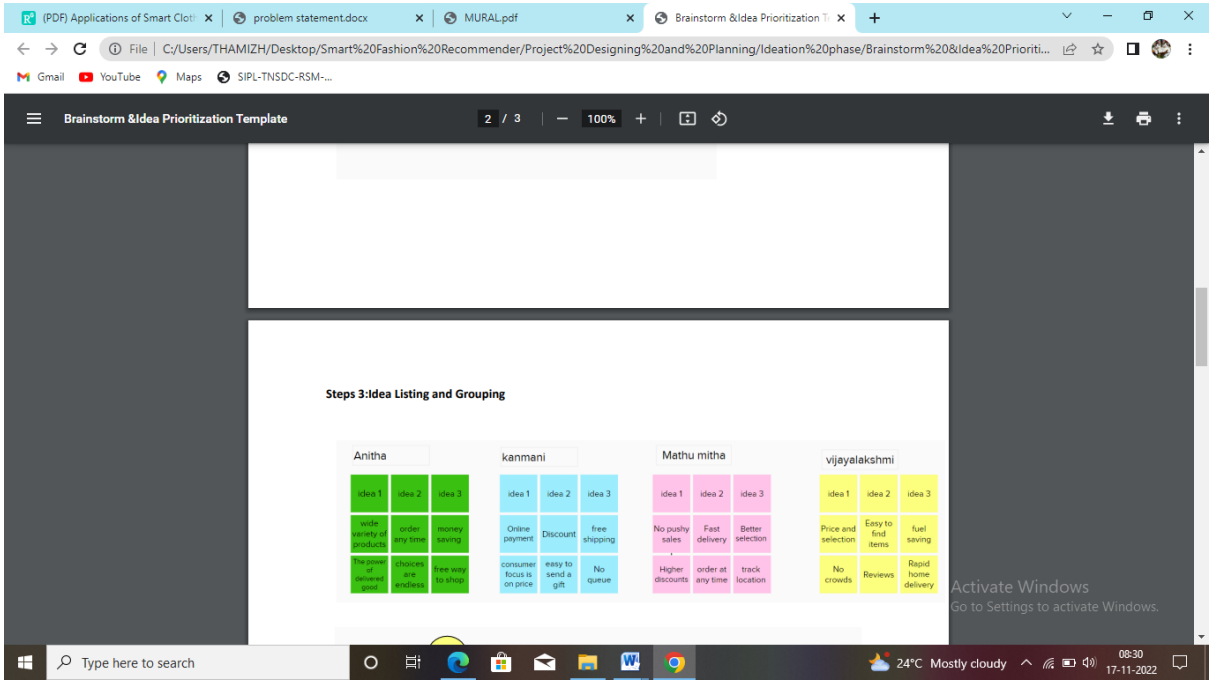
3.IDEATION & PROPOSED SOLUTION

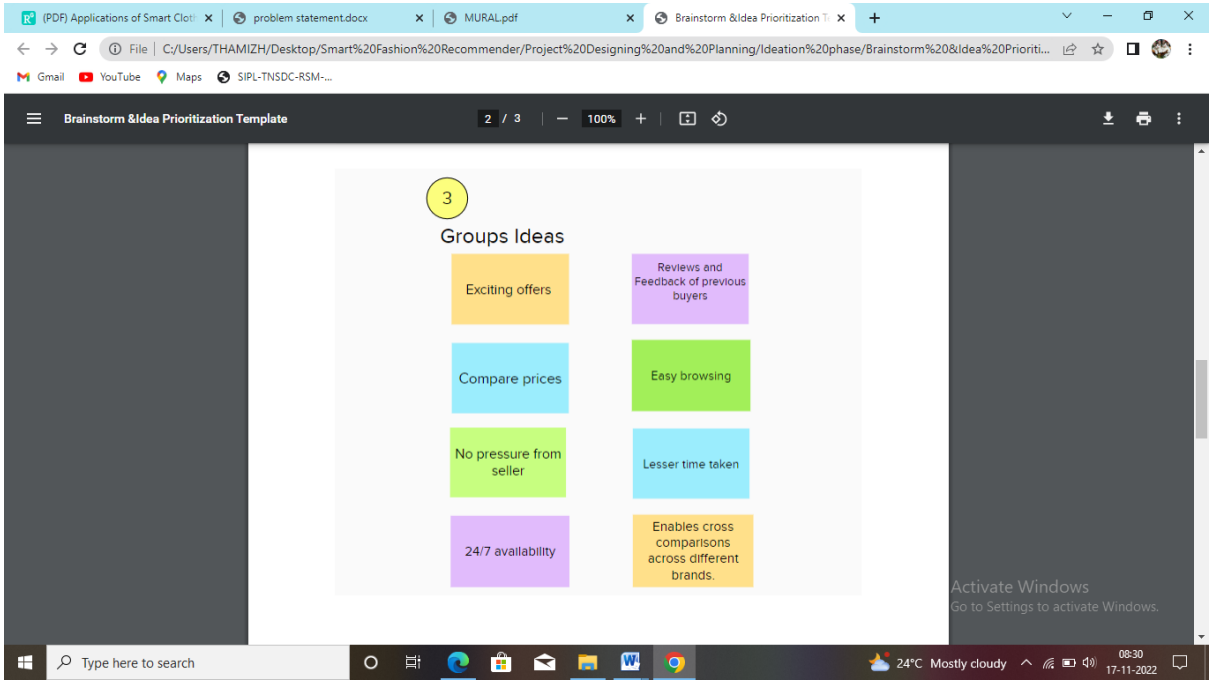
3.1 Empathy MAP Canvas



3.2 Ideation & Brainstorming




(PDF) Applications of Smart Clot
problem statement.docx
MURAL.pdf
Brainstorm &Idea Prioritization T
File | C:/Users/THAMIZH/Desktop/Smart%20Fashion%20Recommender/Project%20Designing%20and%20Planning/Ideation%20phase/Brainstorm%20&Idea%20Prioriti...
Gmail
YouTube
Maps
SIPL-TNSDC-RSM-...
Brainstorm &Idea Prioritization Template
2 / 3
100%
Steps 3:Idea Listing and Grouping
Anitha
idea 1
idea 2
idea 3
wide variety of products
order any time
money saving
The power of delivered good
choices are endless
free way to shop
kanmani
idea 1
idea 2
idea 3
Online payment
Discount
free shipping
consumer focus is on price
easy to send a gift
No queue
Mathu mitha
idea 1
idea 2
idea 3
No pushy sales
Fast delivery
Better selection
Higher discounts
order at any time
track location
vijayalakshmi
idea 1
idea 2
idea 3
Price and selection
Easy to find items
fuel saving
No crowds
Reviews
Rapid home delivery
Activate Windows
Go to Settings to activate Windows.
Type here to search
24°C Mostly cloudy
08:30
17-11-2022


(PDF) Applications of Smart Clot
problem statement.docx
MURAL.pdf
Brainstorm &Idea Prioritization T
File | C:/Users/THAMIZH/Desktop/Smart%20Fashion%20Recommender/Project%20Designing%20and%20Planning/Ideation%20phase/Brainstorm%20&Idea%20Prioriti...
Gmail
YouTube
Maps
SIPL-TNSDC-RSM-...
Brainstorm &Idea Prioritization Template
2 / 3
100%
3
Groups Ideas
Exciting offers
Reviews and Feedback of previous buyers
Compare prices
Easy browsing
No pressure from seller
Lesser time taken
24/7 availability
Enables cross comparisons across different brands.
Activate Windows
Go to Settings to activate Windows.
Type here to search
24°C Mostly cloudy
08:30
17-11-2022

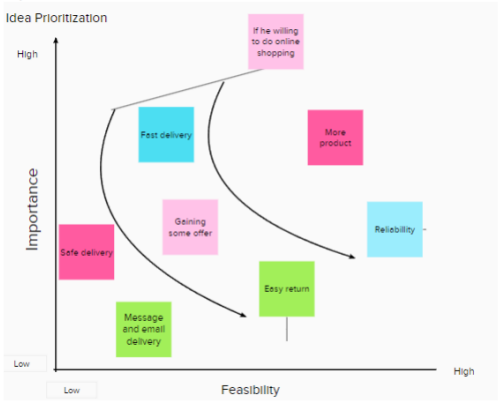
(PDF) Applications of Smart Clot | problem statement.docx | MURAL.pdf | Brainstorm &Idea Prioritization T

File | C:/Users/THAMIZH/Desktop/Smart%20Fashion%20Recommender/Project%20Designing%20and%20Planning/Ideation%20phase/Brainstorm%20&Idea%20Prioriti...

Gmail YouTube Maps SIPL-TNSDC-RSM-...

Brainstorm &Idea Prioritization Template 3 / 3 100%

Step-4: Idea Prioritization



Activate Windows
Go to Settings to activate Windows.

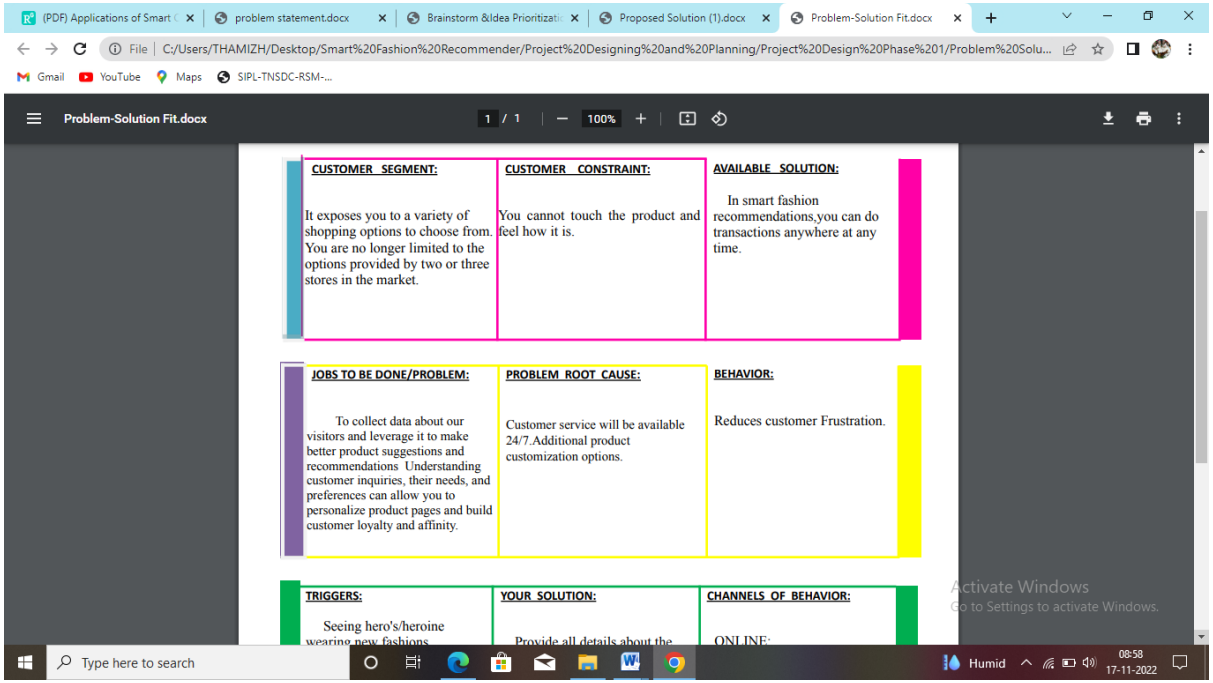
Type here to search

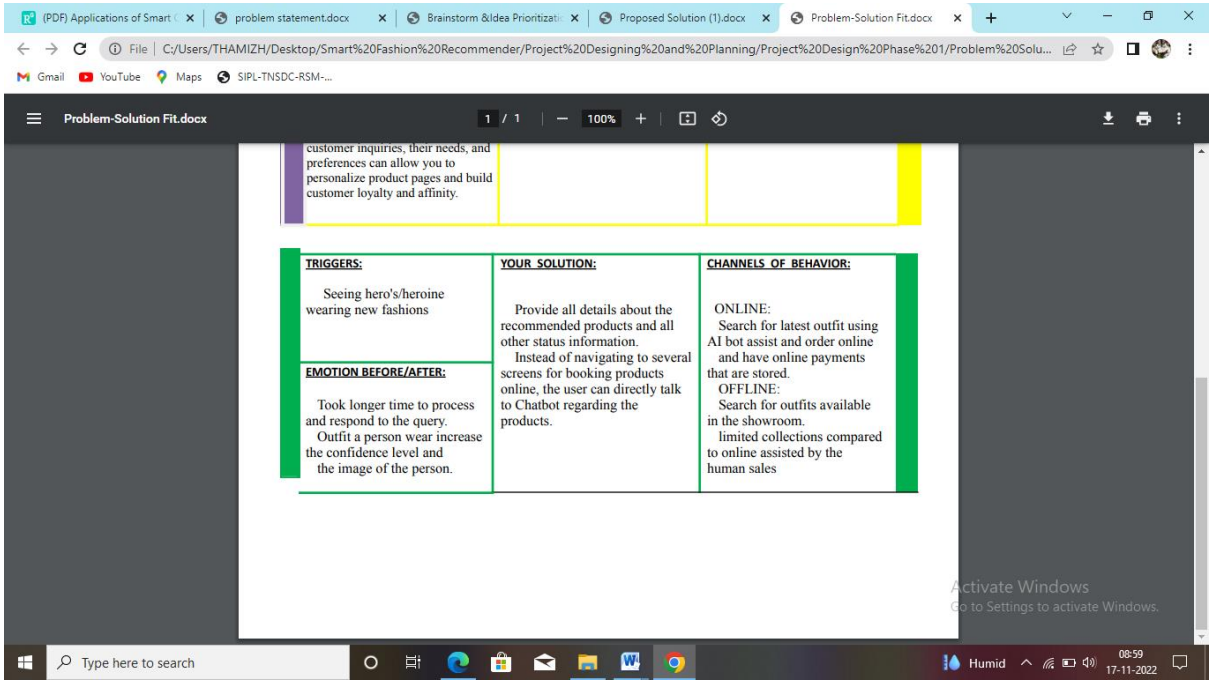
24°C Mostly cloudy 08:30 17-11-2022

3.3 Proposed Solution

S. No	Parameters	Description
1.	Problem Statement (Problem to be solved)	Poor tracking logistics and long delivery. lack of support and no live chat opportunity.
2.	Idea / Solution description	By using Smart Fashion Recommender application Reduce human effect. Customer assessment. The speed of navigation.
3.	Novelty/ Uniqueness	Reduces time in choosing right product thus increase sales. The customer will talk to Chat bot regarding the Products. Get the recommendations based on information provided by the user.
4.	Social Impact / Customer Satisfaction	Providing better services. The user friendly interface, Assistants form chat bot finding a dress makes the customer satisfied.
5.	Business Model (Revenue Model)	The chat bot sells our Products to customers. Customers buy our products and generate revenue
6.	Scalability of the Solution	We can easily scalable our Applications by increases the items and products

3.4 Problem Solution Fit




(PDF) Applications of Smart
problem statement.docx
Brainstorm &Idea Prioritizati
Proposed Solution (1).docx
Problem-Solution Fit.docx
File | C:/Users/THAMIZH/Desktop/Smart%20Fashion%20Recommender/Project%20Designing%20and%20Planning/Project%20Design%20Phase%201/Problem%20Solu...
Gmail
YouTube
Maps
SIPL-TNSDC-RSM-...
Problem-Solution Fit.docx
1 / 1
100%
customer inquiries, their needs, and preferences can allow you to personalize product pages and build customer loyalty and affinity.
TRIGGERS:
Seeing hero's/heroine wearing new fashions
EMOTION BEFORE/AFTER:
Took longer time to process and respond to the query.
Outfit a person wear increase the confidence level and the image of the person.
YOUR SOLUTION:
Provide all details about the recommended products and all other status information.
Instead of navigating to several screens for booking products online, the user can directly talk to Chatbot regarding the products.
CHANNELS OF BEHAVIOR:
ONLINE:
Search for latest outfit using AI bot assist and order online and have online payments that are stored.
OFFLINE:
Search for outfits available in the showroom.
limited collections compared to online assisted by the human sales
Activate Windows
Go to Settings to activate Windows.
Type here to search
Humid
08:59
17-11-2022

CHAPTER-4

4.REQUIREMENT ANALYSIS

4.1 Functional Requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirements	Sub Registration
FR-1	Mobile Responsive	The most of users for the website are coming through mobile devices especially smartphones.
FR-2	Order and Checkout Flow	You have to also mention the discount and coupon
FR-3	Delivery confirmation	Confirmation via email and phone number
FR-4	Product Attributes	Will, the customer chooses only product size and color

4.2 Non-Functional Requirements:

Following are the Non-Functional requirements of the proposed solution.

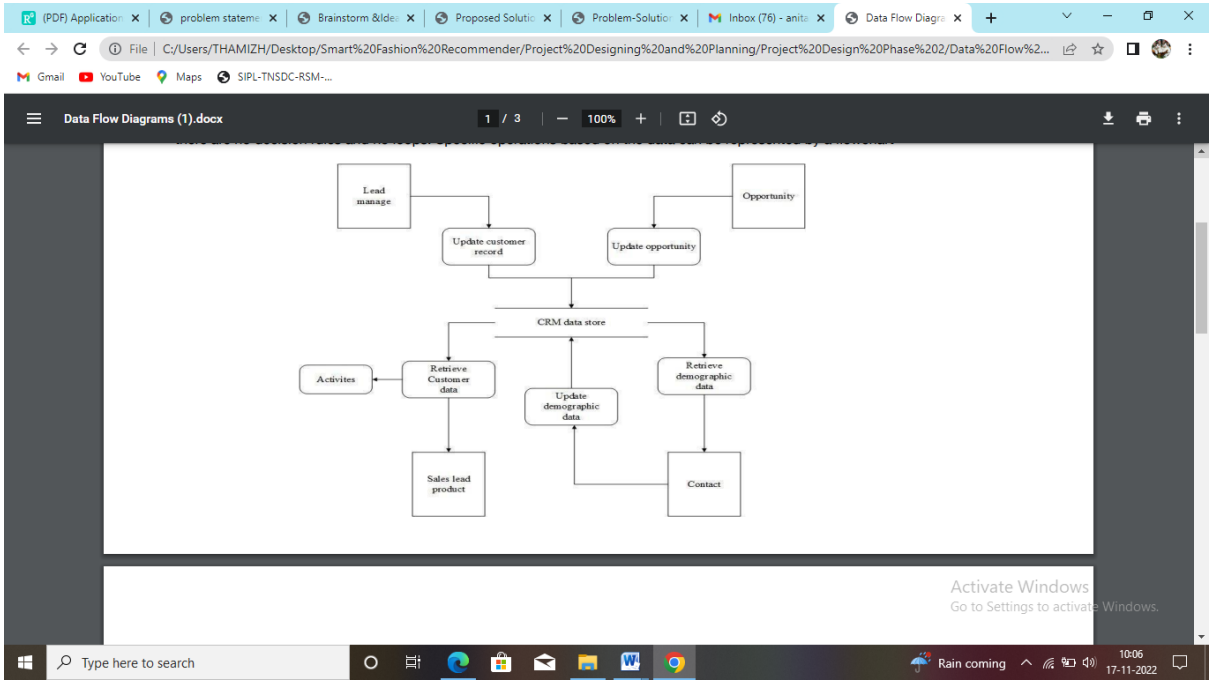
FR.No	Non-Functional Requirement	Description
NFR-1	Usability	A user-friendly interface with chat bot to make usability efficient.
NFR-2	Security	Secured connection HTTPS should be established for transmitting requests and responses.
NFR-3	Performance	Set up the speed benchmark, Store as fast as possible regardless of the integrations.
NFR-4	Maintainability	The operational costs for maintenance are the tricky part of planning a business budget.
NFR-5	Availability	It is a cloud based web application so user can access without any platform limitations ,just using a browsers with a internet connection is enough for use the application
NFR-6	Scalability	It has a quick request and response time, high throughput.

CHAPTER-5

5. PROJECT DESIGN

5.1 Data Flow Diagrams

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.



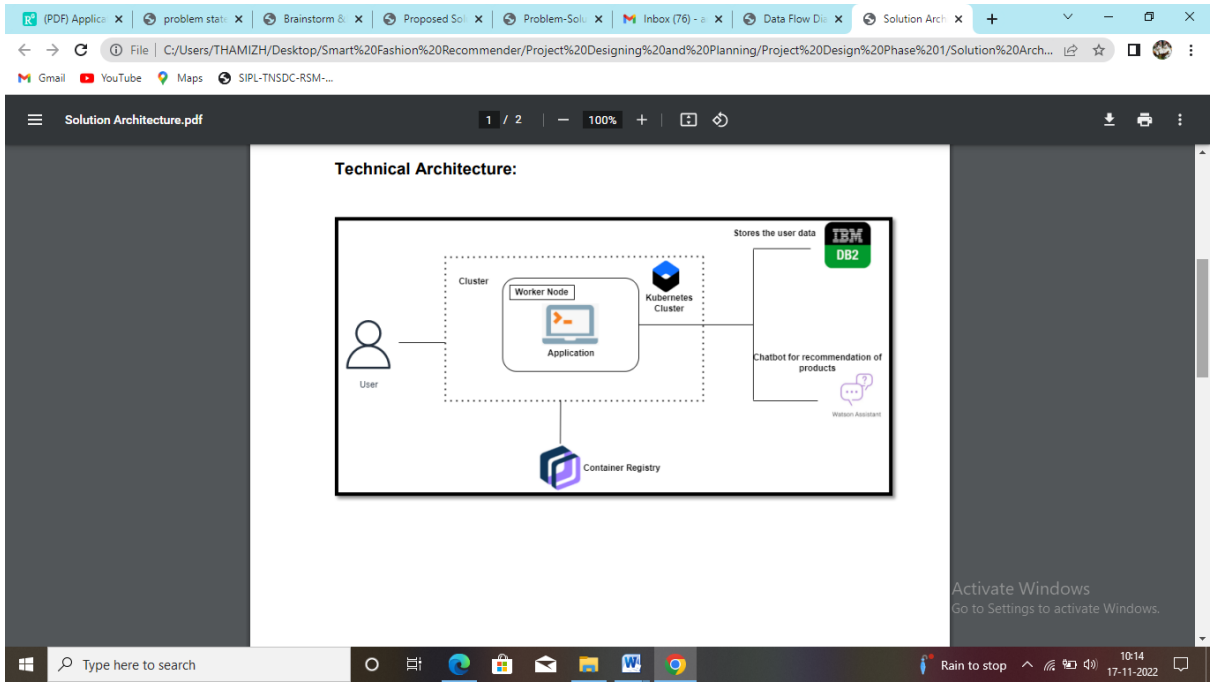
5.2 Solution And Technical Architecture

Solution Architecture

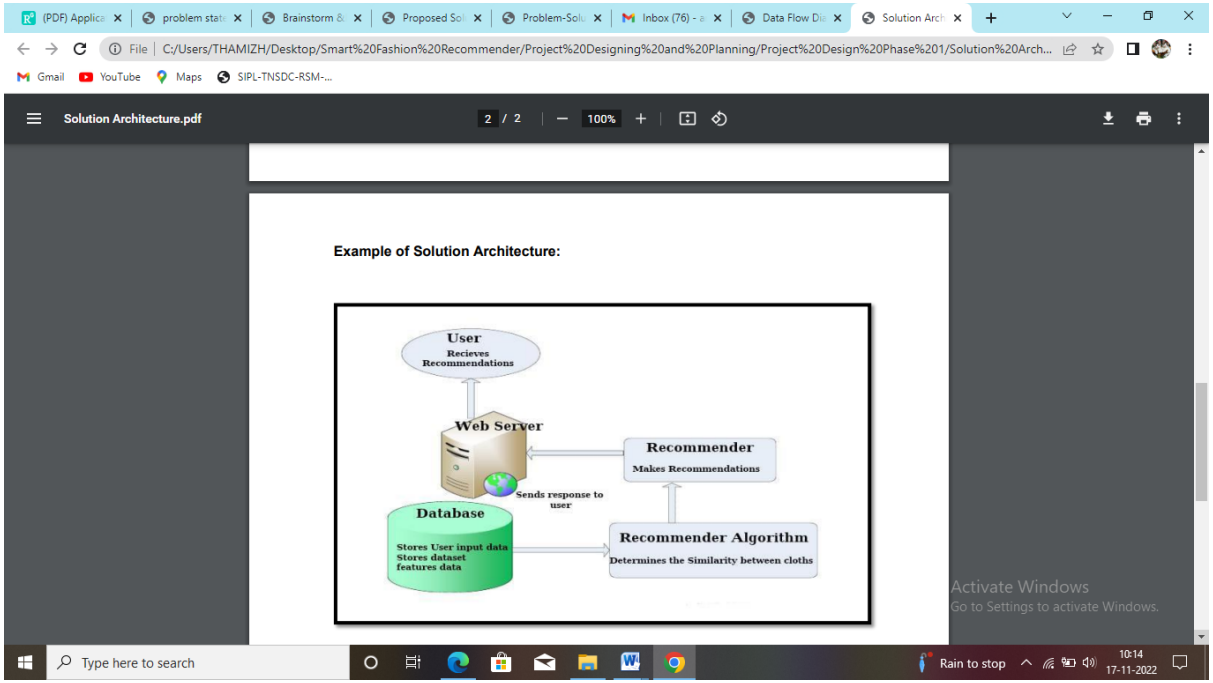
Solution architecture is a complex process – with many sub-processes that bridges the gap between business problems and technology solutions. Its goals are to:

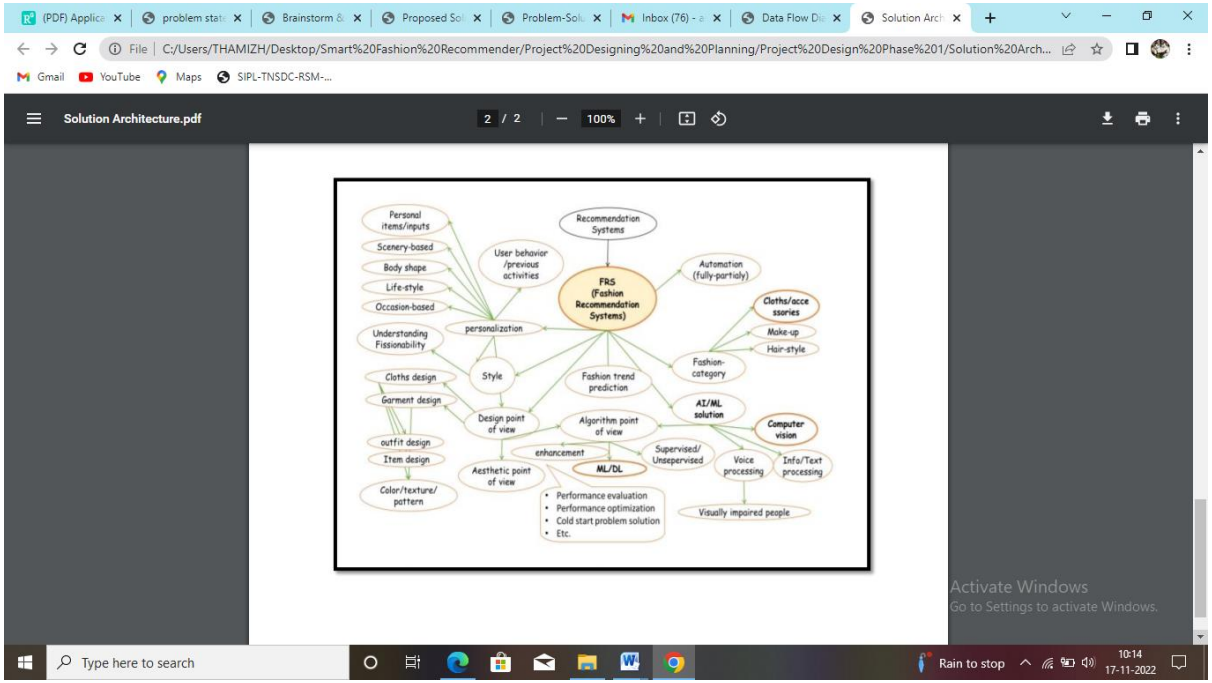
- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Technical Architecture



Example Of Solution Architecture:




Solution Architecture.pdf
2 / 2
100%
Personal items/inputs
Scenery-based
Body shape
Life-style
Occasion-based
Understanding Fissionability
Cloths design
Garment design
outfit design
Item design
Color/texture/ pattern
User behavior /previous activities
personalization
Style
Design point of view
Aesthetic point of view
Recommendation Systems
FRS (Fashion Recommendation Systems)
Fashion trend prediction
Algorithm point of view
enhancement
ML/DL
Performance evaluation
Performance optimization
Cold start problem solution
Etc.
Automation (fully-partialy)
Cloths/acce ssories
Make-up
Hair-style
Fashion-category
AI/ML solution
Computer vision
Supervised/ Unsepervised
Voice processing
Info/Text processing
Visually impaired people
Activate Windows
Go to Settings to activate Windows.
Type here to search
Rain to stop
10:14
17-11-2022

5.3 User Stories

User Type	Functional Requirement	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	1	I can register for the application by entering my email, password,and confirmingmy password.	I can access my account /dashboard	High	Sprint-1
		2	I will receive confirmation email once I have registered for the Application	I can receive confirmationemail & click confirm	High	Sprint-1
		3	I can register for the applicationthrough Facebook	I can register & access thedashboardwith Facebook Login	Low	Sprint-2
		4	I can register for the application through Gmail		Medium	Sprint-1
	Login	5	I can log into the application by entering email & password	I can access my data bylogin	High	Sprint-1
	Dashboard	6	I can view the dashboard and by products		High	Sprit -2
Customer (Webuser)	Registration /Login	7	I can register for the application by entering my email, password, and confirmingmy password.	I can access my account /dashboard		Sprint -1
Customer Care Executive	Contac twith Customes	8	I solve the customer Requirements and feedback	I can receive calls fromcustomers	High	Sprint-1

Administrator	Check stock andPrice , Orders	9	I can Check the databaseAnd stock details and buying and selling prices	I am the administrator ofthe company	High	Sprint -2

CHAPTER-6

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning And Estimation

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	ANITHA M KANMANI M MATHU MITHA R VIJAYALAKSHMI C
Sprint-2	Admin panel	USN-2	The role of the admin is to check out the database about the stock and have a track of all the things that the users are purchasing.	20	High	ANITHA M KANMANI M MATHU MITHA R VIJAYALAKSHMI C
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	ANITHA M KANMANI M MATHU MITHA R VIJAYALAKSHMI C
Sprint-4	final delivery	USN-4	Container of applications using docker kubernetes and deployment of the application. Create the documentation and final submit the application	20	High	ANITHA M KANMANI M MATHU MITHA R VIJAYALAKSHMI C

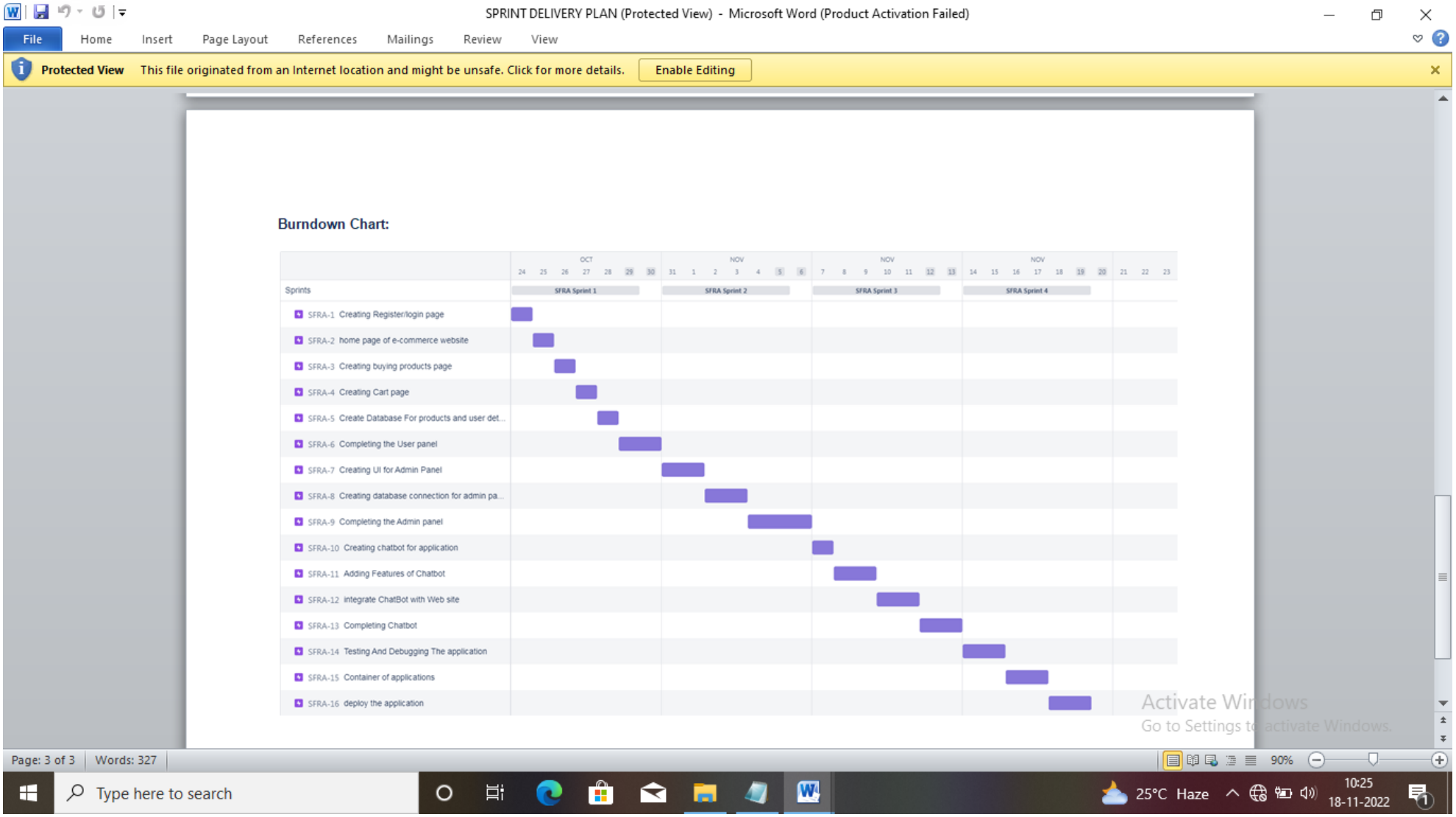
6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022		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

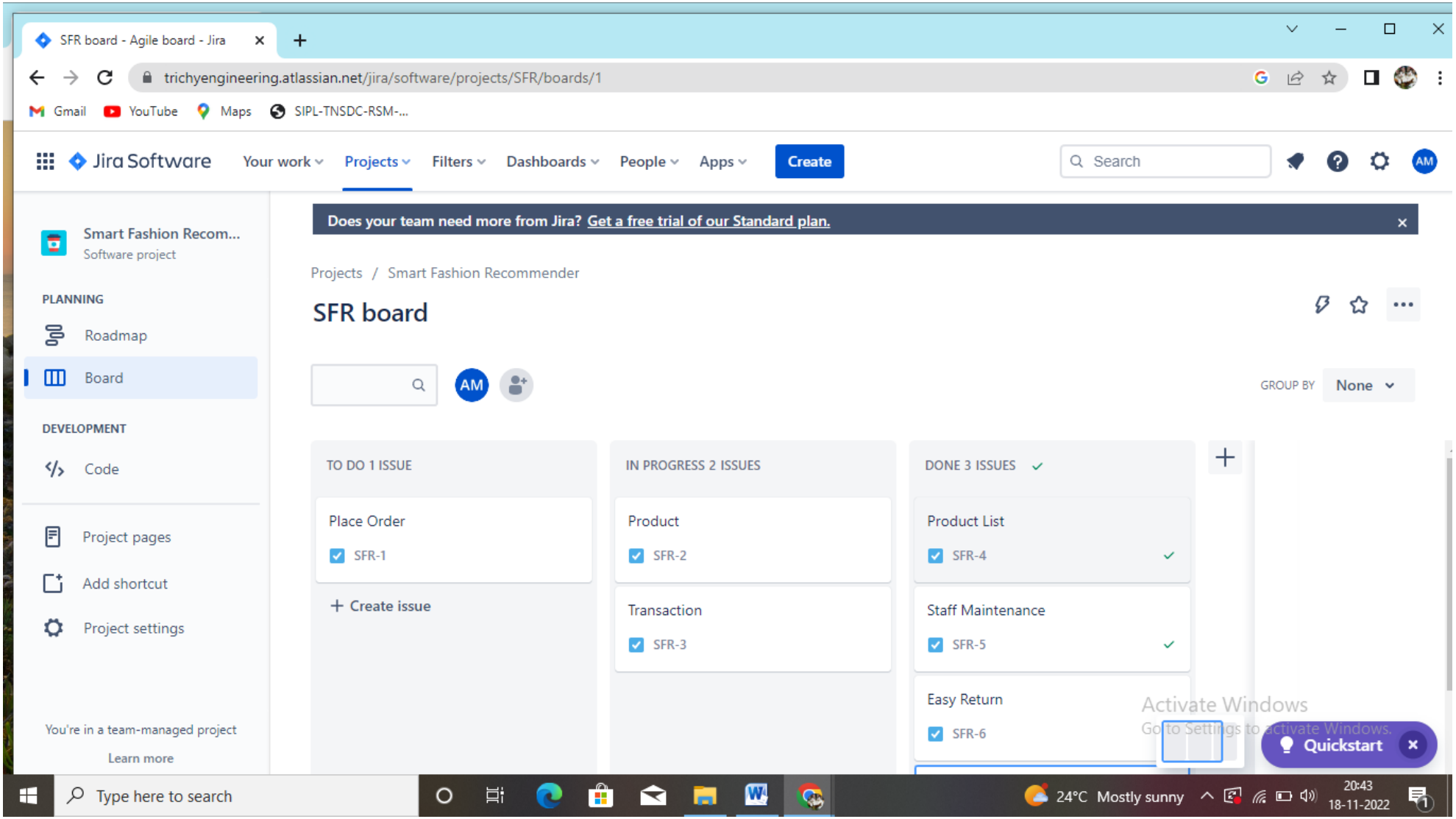
Velocity

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

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



6.3 Reports From JIRA



CHAPTER-7

7.CODING AND SOLUTIONING

7.1 Feature 1

App.py

```
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
from tensorflow.keras.models import Sequential
import numpy as np
from numpy.linalg import norm
import os
from tqdm import tqdm
import pickle


model = ResNet50(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
model.trainable = False


model = Sequential([model, GlobalMaxPooling2D()])
```

```
#model.summary()

def extract_features(img_path,model):
    img = image.load_img(img_path,target_size=(224,224))
    img_array = image.img_to_array(img)
    expand_img = np.expand_dims(img_array,axis=0)
    preprocessed_img = preprocess_input(expand_img)
    result_to_resnet = model.predict(preprocessed_img)
    flatten_result = result_to_resnet.flatten()
    # normalizing
    result_normlized = flatten_result / norm(flatten_result)

    return result_normlized
#print(os.listdir('fashion_small/images'))
img_files = []

for fashion_images in os.listdir('fashion_small/images'):
    images_path = os.path.join('fashion_small/images', fashion_images)
    img_files.append(images_path)
```

```
# extracting image features
image_features = []

for files in tqdm(img_files):
    features_list = extract_features(files, model)
    image_features.append(features_list)

pickle.dump(image_features, open("image_features_embedding.pkl", "wb"))
pickle.dump(img_files, open("img_files.pkl", "wb"))
```

7.2 Feature 2

Main.py

```
import streamlit as st
import tensorflow
import pandas as pd
from PIL import Image
import pickle
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.models import Sequential
from numpy.linalg import norm
from sklearn.neighbors import NearestNeighbors
import os

features_list = pickle.load(open("image_features_embedding.pkl", "rb"))
img_files_list = pickle.load(open("img_files.pkl", "rb"))
```

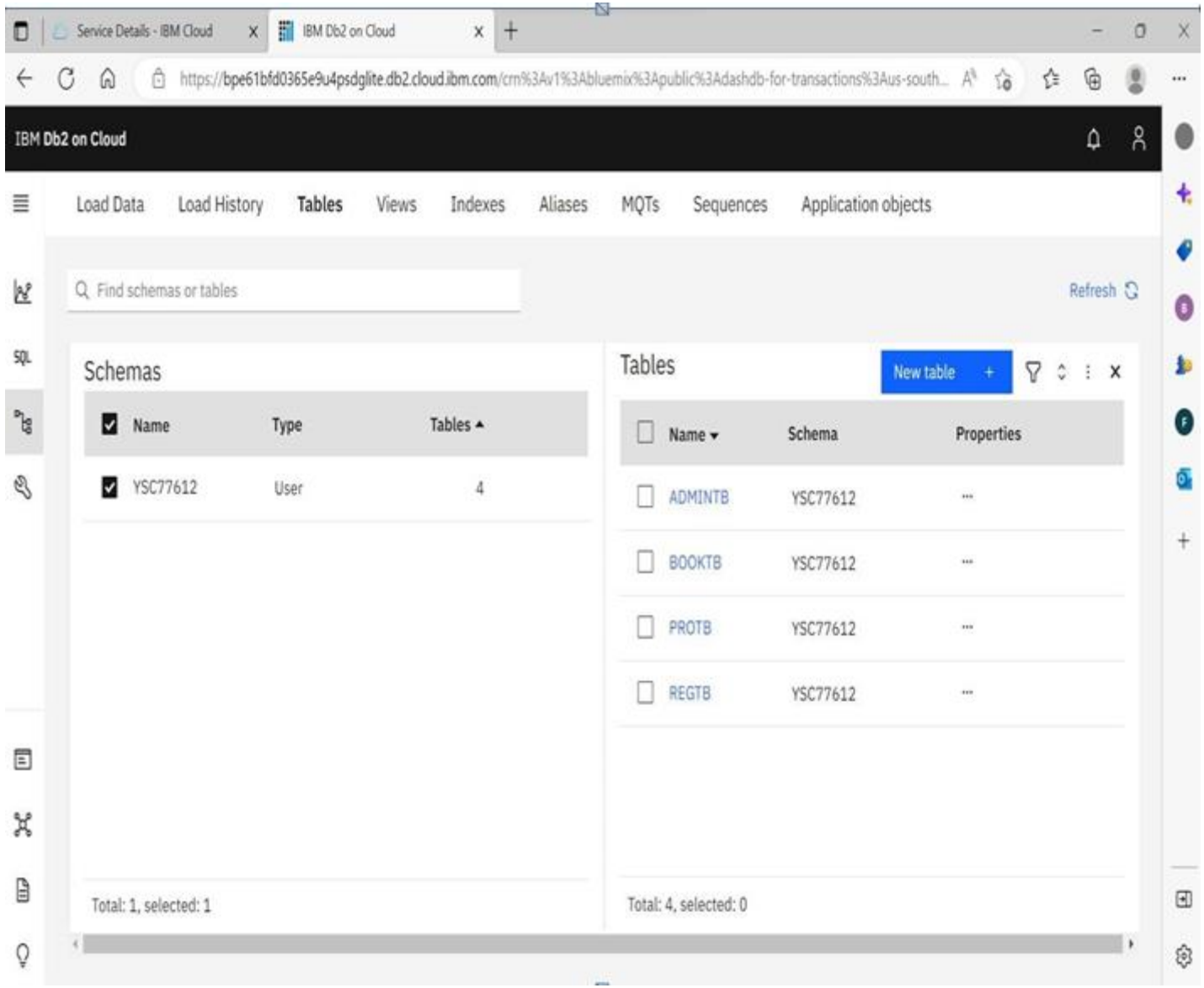
```
model = ResNet50(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
model.trainable = False
model = Sequential([model, GlobalMaxPooling2D()])

st.title('Clothing recommender system')



def save_file(uploaded_file):
    try:
        with open(os.path.join("uploader", uploaded_file.name), 'wb') as f:
            f.write(uploaded_file.getbuffer())
            return 1
    except:
        return 0



def extract_img_features(img_path, model):
    img = image.load_img(img_path, target_size=(224, 224))
```

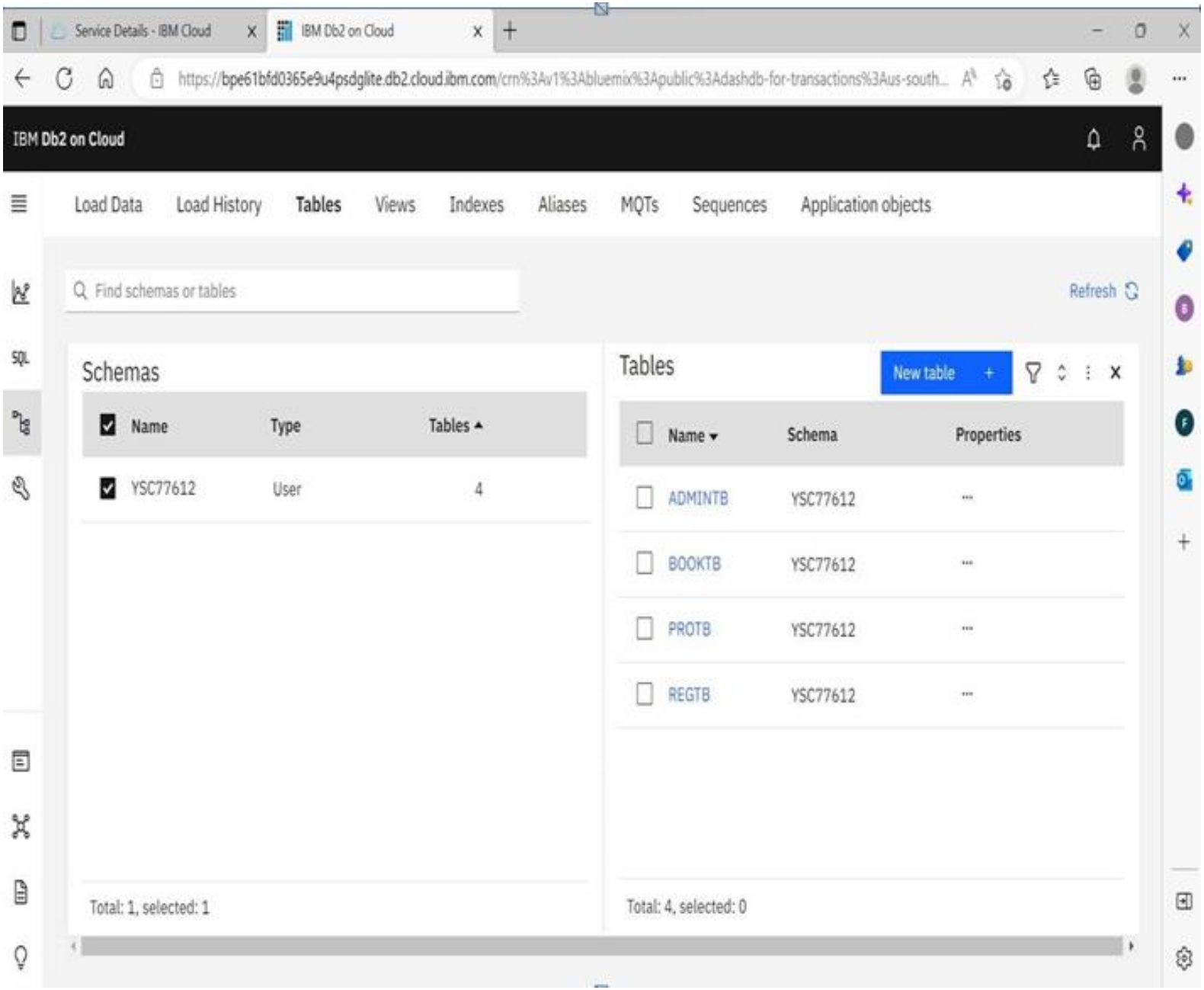
```
    img_array = image.img_to_array(img)
    expand_img = np.expand_dims(img_array, axis=0)
    preprocessed_img = preprocess_input(expand_img)
    result_to_resnet = model.predict(preprocessed_img)
    flatten_result = result_to_resnet.flatten()
    # normalizing
    result_normlized = flatten_result / norm(flatten_result)

    return result_normlized



def recommendd(features, features_list):
    neighbors = NearestNeighbors(n_neighbors=6, algorithm='brute', metric='euclidean')
    neighbors.fit(features_list)

    distence, indices = neighbors.kneighbors([features])

    return indices
```

```
uploaded_file = st.file_uploader("Choose your image")
if uploaded_file is not None:
    if save_file(uploaded_file):
        # display image
        show_images = Image.open(uploaded_file)
        size = (400, 400)
        resized_im = show_images.resize(size)
        st.image(resized_im)
        # extract features of uploaded image
        features = extract_img_features(os.path.join("uploader", uploaded_file.name), model)
        #st.text(features)
        img_indicess = recommendd(features, features_list)
        col1,col2,col3,col4,col5 = st.columns(5)

            with col1:
            st.header("I")
            st.image(img_files_list[img_indicess[0][0]])

        with col2:
```

```
        st.header("II")
        st.image(img_files_list[img_indicess[0][1]])

    with col3:
        st.header("III")
        st.image(img_files_list[img_indicess[0][2]])

    with col4:
        st.header("IV")
        st.image(img_files_list[img_indicess[0][3]])

    with col5:
        st.header("V")
        st.image(img_files_list[img_indicess[0][4]])
else:
    st.header("Some error occur")
```

7.3 Database Schema




Service Details - IBM Cloud
IBM Db2 on Cloud
https://bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south...
IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
Find schemas or tables
Refresh
Schemas
Name
Type
Tables
YSC77612
User
4
Total: 1, selected: 1
Tables
New table
Name
Schema
Properties
ADMINTB
YSC77612
BOOKTB
YSC77612
PROTB
YSC77612
REGTB
YSC77612
Total: 4, selected: 0

CHAPTER-8

8.TESTING

8.1 Test Cases

Verify that on the product page, the user can select the desired attribute of the product e.g. size, color.

User can add to the cart one or more products.

Total products is displayed on the category pages.

The products is correctly displayed on the search result page for a particular search term.

Users can add products to the wishlist.

The product return functionality works correctly.

Cash on Delivery option of payment is working fine.

S.NO	Scenario	Input	Excepted output	Actual output
1	User login	User name and password	Login	Login success.
2	Search Product	Show product list	Purchase successfully	User details are stored in a database.
3	Post Queries	Ask questions to chatbot	Get result from chatbot	Details are stored in a database.

8.2 User Acceptance Testing

Unable to choose any sizes for the products. displayed Some products are displayed without their listed price. Unable to enter payment information due to missing input fields on payment page .Pop in window fails to appear when users reach the homepage. Some items added to the basket fail to appear in the basket during checkout. When attempting to login into their account, users were erroneously redirected to the homepage without gaining access to their accounts.

Failure to redirect users to a product's information page after clicking on the product. Images are displayed on top of one another. Purchases are blocked by a server error. Unable to scroll up or down a page smoothly. Missing payment options (gift cards) payment page. Unable to remove items from the cart

CHAPTER-9

9.RESULT

9.1 PERFORMANCE METRICS

The application can be used for any Ecommerce application. It is easy to use, since it uses the GUI provided in the user dialog. User friendly screens are provided. The application is easy to use and interactive making online shopping a recreational activity for users. It has been thoroughly tested and implemented.

CHAPTER-10

10. ADVANTAGES AND DISADVANTAGES

Advantages

Track Physical Activity. Some smart clothes can track the wearer's movements.

Monitor Health Conditions.

Mood Monitoring is one of the Common Pros of Smart Clothes.

Monitor Sleep Patterns.

Improve Athletic Performance.

Disadvantages

The high cost of energy

Skin problems

Smart clothes are expensive

Not long lasting.

11.CONCLUSION

The Fashion Recommendation System is mainly used to recommend the best possible outfit combinations to a user who has no fashion sense based on their wardrobe . It may not always provide the best possible outfit to wear for an occasion as the system is dependent completely on the clothes present in the user's wardrobe. Also another reason is that fashion is highly dependent on the time period. However the system does a great job in inculcating a fashion sense among the users and can provide the best recommendations based on the user's wardrobe. Since the system is implemented as a website, it is very easy for the end users to access as well as use. The search engine provides an easy and convenient way to search for products where a user can Search for a product interactively and the search engine would refine the products available based on the user's input.

12.FUTURE SCOPE

Customers get a clear understanding of any online product with detailed information. Information may involve the products functionalities, specifications, and images with prices. It is easy to make additional enhancements to this system because of the way it was designed. customer ratings and feedback, price reductions, accessibility, a streamlined return policy, points that can be used for future purchase. The users could subscribe for price alerts which would enable them to receive messages when price for products fall below a particular level. The current system is confined only to the shopping cart process. It can be extended to have an easy to use check out process. Users can have multiple shipping and billing information saved. During checkout they can use the drag and drop feature to select shipping and billing information.

13 APPENDIX

Source code

```
import pickle
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.models import Sequential
from numpy.linalg import norm
from sklearn.neighbors import NearestNeighbors
import cv2

features_list = pickle.load(open("image_features_embedding.pkl", "rb"))
img_files_list = pickle.load(open("img_files.pkl", "rb"))

print(np.array(features_list).shape)
```

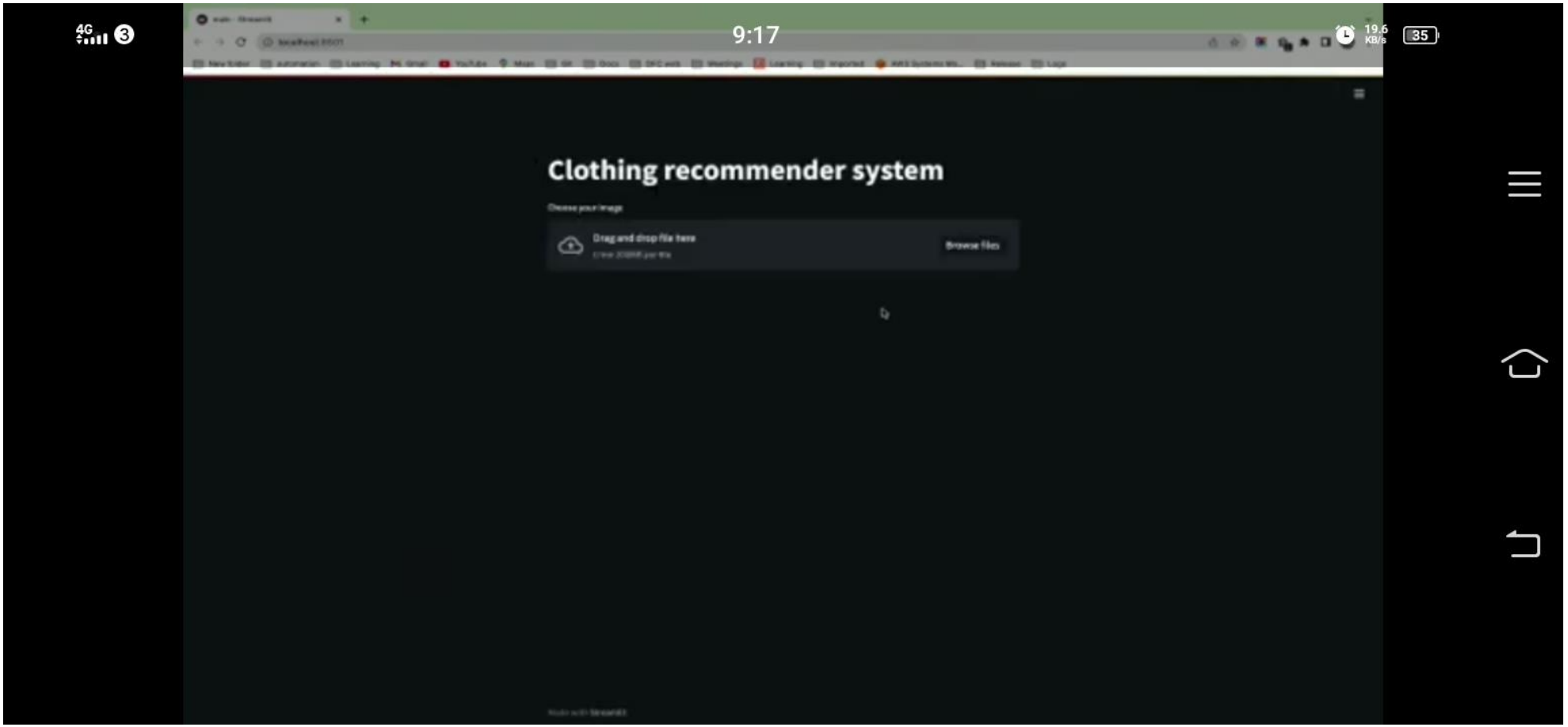
```
model = ResNet50(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
model.trainable = False

model = Sequential([model, GlobalMaxPooling2D()])

img = image.load_img('sample/shoes.jpg',target_size=(224,224))
img_array = image.img_to_array(img)
expand_img = np.expand_dims(img_array,axis=0)
preprocessed_img = preprocess_input(expand_img)
result_to_resnet = model.predict(preprocessed_img)
flatten_result = result_to_resnet.flatten()
# normalizing
result_normlized = flatten_result / norm(flatten_result)

neighbors = NearestNeighbors(n_neighbors = 6, algorithm='brute', metric='euclidean')
neighbors.fit(features_list)

distence, indices = neighbors.kneighbors([result_normlized])
```

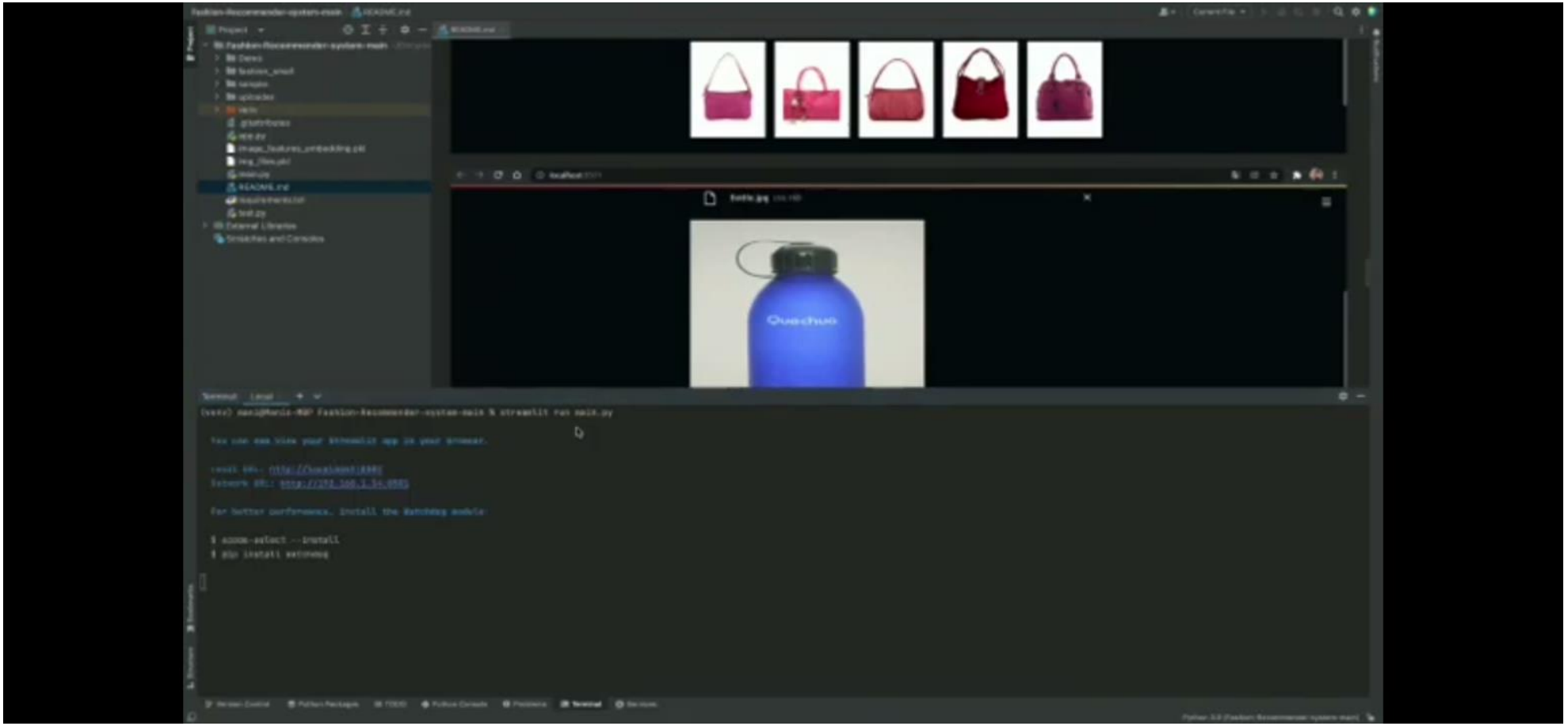
```
print(indices)


for file in indices[0][1:6]:
    print(img_files_list[file])
    tmp_img = cv2.imread(img_files_list[file])
    tmp_img = cv2.resize(tmp_img,(200,200))
    cv2.imshow("output", tmp_img)
    cv2.waitKey(0)
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
from tensorflow.keras.models import Sequential
import numpy as np
from numpy.linalg import norm
import os
from tqdm import tqdm
import pickle


model = ResNet50(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
```

```
model.trainable = False

model = Sequential([model, GlobalMaxPooling2D()])
#model.summary()

def extract_features(img_path,model):
    img = image.load_img(img_path,target_size=(224,224))
    img_array = image.img_to_array(img)
    expand_img = np.expand_dims(img_array,axis=0)
    preprocessed_img = preprocess_input(expand_img)
    result_to_resnet = model.predict(preprocessed_img)
    flatten_result = result_to_resnet.flatten()
    # normalizing
    result_normlized = flatten_result / norm(flatten_result)

    return result_normlized
#print(os.listdir('fashion_small/images'))
img_files = []
```

```
for fashion_images in os.listdir('fashion_small/images'):
    images_path = os.path.join('fashion_small/images', fashion_images)
    img_files.append(images_path)

# extracting image features
image_features = []

for files in tqdm(img_files):
    features_list = extract_features(files, model)
    image_features.append(features_list)

pickle.dump(image_features, open("image_features_embedding.pkl", "wb"))
pickle.dump(img_files, open("img_files.pkl", "wb"))
```

OUTPUT


9:17
Clothing recommender system
Drag and drop file here
Browse files


Quechua
(venv) nani@Mario-MBP Fashion-Recommender-system-main % streamlit run main.py
$ xcode-select --install
$ pip install watchdog

GITHUB AND PROJECT DEMO LINK

Githup Id

IBM-Project-39000-1660388626

Demo Link

https://youtu.be/KUAh--yHwx8