SMART FASHION RECOMMENDER APPLICATION

PROJECT REPORT

Submitted by

KALI APPAN A(Team Member)	**950619106005**
MANI GURU VELLAIAN K(Team Member)	**950619106008**
SANTHANAKUMAR V(Team Member)	**950619106023**
TAMIL VASIKARAN S(Team Leader)	**950619106027**

TEAM ID-PNT2022TMID49933

HX8001 PROFESSIONAL READYINESS FOR INNOVATION EMPLOYABLITY AND ENTREPRENEURSHIP

in the department

of

ELECTRONICS AND COMMUNICATION ENGINEERING

EINSTEIN COLLEGE OF ENGINEERING, TIRUNELVELI-627 012
ANNA UNIVERSITY: CHENNAI 600 025
NOVEMBER: 2022

BONAFIDE CERTIFICATE

Certified this Report "**SMART FASHION RECOMMENDER APPLICATION**" for the project, is the bonafied work of **TAMIL VASIKARAN S(950619106027), KALIAPPAN A(950619106005), MANI GURU VELLAIAN K(950619106008) and SANTHANAKUMAR V (950619106023)**who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or award was co-offered on the earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr. P. REVATHY	**Dr. P.REVATHY**
HOD/ECE	**MENTOR**

EVALUATOR	**SPOC**
Mrs. T. Viji	**Mr.S.ARUNSINGH**
Asst.prof/ECE	**Asst.prof/ECE**

ACKNOWLEDGEMENT

We have successfully completed the project with blessings showered onus by god, the almighty, A project of this nature needs co-operation and support from many for successful completion.

We express our heartfelt thanks to **Mr.A.MATHIVANAN., M.sc.,(Agri),** Managing Trustee of Einstein college of Engineering, Tirunelveli, for his mortal support and device.

Our thanks to **Prof.A.AMUTHAVANAN, BE., M.S (USA).B.L,**Chairman of our college for making necessary arrangements to do this project.

Our heartly thanks to **Prof.EZHILVANAN, MBA.,** Secretary of our college for making necessary arrangements to do this project.

We wish to express our gratitude to **Dr.VELAYUTHAM,M.E,Ph.D.,FIE.** Principal for the support he provided us to carry out this project successfully.

We are very much thankful to **Dr.P.REVATHY** , Head of the Department , Electronics and communication Engineering who is always a constant of inspiring us.

We are extended our sincere thanks to our project evaluator **Mrs.T.VIJI M.E** and project SPOC **Mr.S.ARUNSINGH B.Tech** and friends for their help in completing this project.

.

ABSTRACT

Fashion is perceived as a meaningful way of self-expressing that people use for different purposes. It seems to be an integral part of every person in modern societies, from everyday life to exceptional events and occasions. Fashionable products are highly demanded, and consequently, fashion is perceived as a desirable and profitable industry. Although this massive demand for fashion products provides an excellent opportunity for companies to invest in fashion-related sectors, it also faces different challenges in answering their customer needs. Fashion recommender systems have been introduced to address these needs. This thesis aims to provide deeper insight into the fashion recommender system domain by conducting a comprehensive literature review on more than 100 papers in this field focusing on image-based fashion recommender systems considering computer vision advancements. Justifying fashion domain-specific characteristics, the subtle notions of this domain and their relevancy have been conceptualized. Four main tasks in image-based fashion recommender systems have been recognized, including cloth-item retrievals, Complementary item recommendation, Outfit recommendation, and Capsule wardrobes. An evolvement trajectory of image-based fashion recommender systems concerning computer vision advancements has been illustrated consists of three main eras and the most recent developments. Finally, a comparison between traditional computer vision techniques and deep learning-based has been made. Although the main objective of this literature review was to perform a comprehensive, integrated overview of researches in this field, there is still a need for conducting further studies considering image-based fashion recommender systems from a more practical perspective

SMART FASHION RECOMMENDER APPLICATION

List of Figure

List of Table

CHAPTER 1
1. INTRODUCTION

1.1 Project Overview

We have come up with a new innovative solution through which you can directly do your online shopping based on your choice without any search. It can be done by using the chatbot. The role of the admin is to check out the database about the stock and have a track of all the things that the users are purchasing. The user will login into the website and go through the products available on the website. Instead of navigating to several screens for booking products online, the user can directly talk to Chatbot regarding the products. Get the recommendations based on information provided by the user.

1.2 Purpose

Online shopping is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet using a web browser or a mobile app.

Customers can purchase items from the comfort of their own homes or workplace. Shopping is made easier and convenient for the customer through the internet. It is also easy to cancel the transactions. Saves time and efforts.

- Cart and checkout.
- Product information management.
- Order management.
- Pricing and promotions engines.
- Personalization engines.
- Content management.
- Analytics.
- SEO tools.

CHAPTER 2

2. LITERATURE SURVEY

2.1 References

1. Chen, L., Yang, F., Yang, H.: Image-based product recommendation system with convolutional neural networks (2017)

We are gather from this paper, It present a smart search engine for online shopping. Basically it uses images as its input, and tries to understand the information about products from these images. We first use a neural network to classify the input image as one of the product categories. Then use another neural network to model the similarity score between pair images, which will be used for selecting the closest product in our e-item database. We use Jaccard similarity to calculate the similarity score for training data. We collect product information data (including image, class label etc.) from Amazon to learn these models. Specifically, our dataset contains information about 3.5 million products with image, and there are 20 categories in total. Our method achieves a classification accuracy of 0.5. Finally we are able to recommend products with similarity higher than 0.5, and offer fast and accurate on-line shopping support.

2. Häubl, G., Murray, K.B.: Double agents: assessing the role of electronic product recommendation systems. MIT Sloan Manag. Rev. 47(3), 8–12 (2006)

We are gather from this paper, Electronic information can easily overwhelm people with large volumes of data. An abundance of information often strains human limits: attention, memory, motivation, or other factors. In response to this challenge, software tools that assist humans in filtering and organizing information into more digestible amounts and formats have appeared. This article focuses on tools that provide online shoppers with personalized product recommendations and the benefits and potential difficulties consumers may experience when using such tools

3. He, R., McAuley, J.: VBPR: visual Bayesian personalized ranking from implicit feedback. In: AAAI, pp. 144–150 (2016)

We are gather from this paper, Modern recommender systems model people and items by discovering or `teasing apart' the underlying dimensions that encode the properties of items and users' preferences toward them. Critically, such dimensions are uncovered based on user feedback, often in implicit form (such as purchase histories, browsing logs, etc.); in addition, some recommender systems make use of side information, such as product attributes, temporal information, or review text.However one important feature that is typically ignored by existing personalized recommendation and ranking methods is the visual appearance of the items being considered. In this paper we propose a scalable factorization model to incorporate visual signals into predictors of people's opinions, which we apply to a selection of large, realworld datasets. We make use of visual features extracted from product images using (pre-trained) deep networks, on top of which we learn an additional layer that uncovers the visual dimensions that best explain the variation in people's feedback. This not only leads to significantly more accurate personalized ranking methods, but also helps to alleviate cold start issues, and qualitatively to analyze the visual dimensions that influence people's opinions.

4. Liu, Z., Luo, P., Qiu, S., et al.: DeepFashion: powering robust clothes recognition and retrieval with rich annotations. In: CVPR, pp. 1096–1104 (2016)

We are gather from this paper ,clothes recognition have been driven by the construction of clothes datasets. Existing datasets are limited in the amount of annotations and are difficult to cope with the various challenges in real-world applications. In this work, we introduce DeepFashion, a large-scale clothes dataset with comprehensive annotations. It contains over 800,000 images, which are richly annotated with massive attributes, clothing landmarks, and correspondence of images taken under different scenarios including store, street snapshot, and consumer. Such rich annotations enable the development of powerful algorithms in clothes recognition and facilitating future researches. To demonstrate the advantages of DeepFashion, we propose a new deep model, namely FashionNet, which learns clothing features by jointly predicting clothing attributes and landmarks. The estimated landmarks are then employed to pool or gate the learned features. It is optimized in an iterative manner. Extensive experiments demonstrate the effectiveness of FashionNet and the usefulness of DeepFashion.

5. Schafer, J.B., Konstan, J.A., Riedl, J.: E-commerce recommendation applications. Data Min. Knowl. Discov. 5(1–2), 115–153 (2001)

We are gather from this paper, Recommender systems are being used by an ever-increasing number of E-commerce sites to help consumers find products to purchase. What started as a novelty has turned into a serious business tool. Recommender systems use product knowledge—either handcoded knowledge provided by experts or “mined” knowledge learned from the behavior of consumers—to guide consumers through the often-overwhelming task of locating products they will like. In this article we present an explanation of how recommender systems are related to some traditional database analysis techniques. We examine how recommender systems help E-commerce sites increase sales and analyze the recommender systems at six market-leading sites. Based on these examples, we create a taxonomy of recommender systems, including the inputs required from the consumers, the additional knowledge required from the database, the ways the recommendations are presented to consumers, the technologies used to create the recommendations, and the level of personalization of the recommendations. We identify five commonly used Ecommerce recommender application models, describe several open research problems in the field of recommender systems, and examine privacy implications of recommender systems technology.

6. Ghimire, Devndra Comparative study on Python web frameworks: Flask and Django (2020)

We are gather from this paper, it was found that the most significant advantages of Flask were that it provides simplicity, flexibility, fine-grained control and quick and easy to learn. On the other hand, Django was easy to work with because of its extensive features and support for libraries. Another main advantage of Django is its scalability. It is best fit for a large-scale application. Each framework has its limitations and radiates a fair share of disadvantages. For example, Django is a bit cumbersome for smaller sized applications. However, Flask is too simple to not have the necessary features within the framework.

7. Parag Sunil Shukla, Dr. Priti V. Nigam E- Shopping using Mobile Apps and the Emerging Consumer in the Digital Age of Retail Hyper personalization: An Insight (2018)

We are gather from this paper ,the development of mobile applications (App) has been gradually become the focal point which the enterprises pay attention. Since the “mobile applications” owns the characteristics of entertainment, functionality, information, socialization as well as intellectual stimulation and so on, therefore it gradually becomes the emerging innovative marketing tools for marketing. In this research paper, an attempt has been made to understand the key linkages between experiential value and usage attitude of the shoppers' who prefer to use shopping applications. In this study an attempt has been made conceptually understand the complicated courtship between the connected consumers in today's digital age and the emergence online shopping using mobile apps. This research paper aims to provide a valuable reference for enterprises which are initiating or conducting the implementation of mobile shopping applications, and for researchers interested in the technological innovations in the future

2.2 Problem Statement Definition

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

CHAPTER 3

3. IDEATION & PROPOSED SOLUTION

In this Phase the Planning and Project designing of the application were performed. The Ideation and Proposed solution performs, How the Customer got problems and how they overcome the problems were analyzed.

3.1 Empathy Map

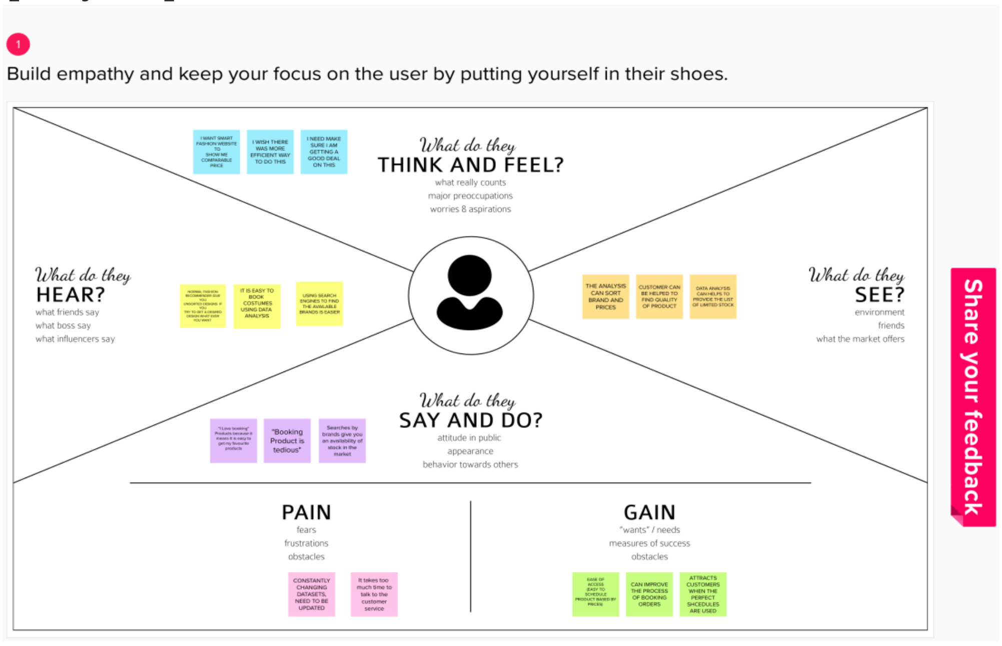

1
Build empathy and keep your focus on the user by putting yourself in their shoes.
What do they THINK AND FEEL?
what really counts
major preoccupations
worries 8 aspirations
I WANT SMART FASHION WEBSITE TO SHOW ME COMPARABLE PRICE
I WISH THERE WAS MORE EFFICIENT WAY TO DO THIS
I NEED MAKE SURE I AM GETTING A GOOD DEAL ON THIS
What do they HEAR?
what friends say
what boss say
what influencers say
IT IS EASY TO BOOK COSTUMES USING DATA ANALYSIS
USING SEARCH ENGINES TO FIND THE AVAILABLE BRANDS IS EASIER
What do they SEE?
environment
friends
what the market offers
THE ANALYSIS CAN SORT BRAND AND PRICES
CUSTOMER CAN BE HELPED TO FIND QUALITY OF PRODUCT
DATA ANALYSIS CAN HELPS TO PROVIDE THE LIST OF LIMITED STOCK
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
"Booking Product is tedious"
Searches by brands give you an availability of stock in the market
PAIN
fears
frustrations
obstacles
CONSTANTLY CHANGING DATASETS, NEED TO BE UPDATED
It takes too much time to talk to the customer service
GAIN
"wants" / needs
measures of success
obstacles
CAN IMPROVE THE PROCESS OF BOOKING ORDERS
ATTRACTS CUSTOMERS WHEN THE PERFECT SHCEDULES ARE USED
Share your feedback

Figure 3.1 Empathy Map

3.2 Ideation & Brainstorming

Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge.

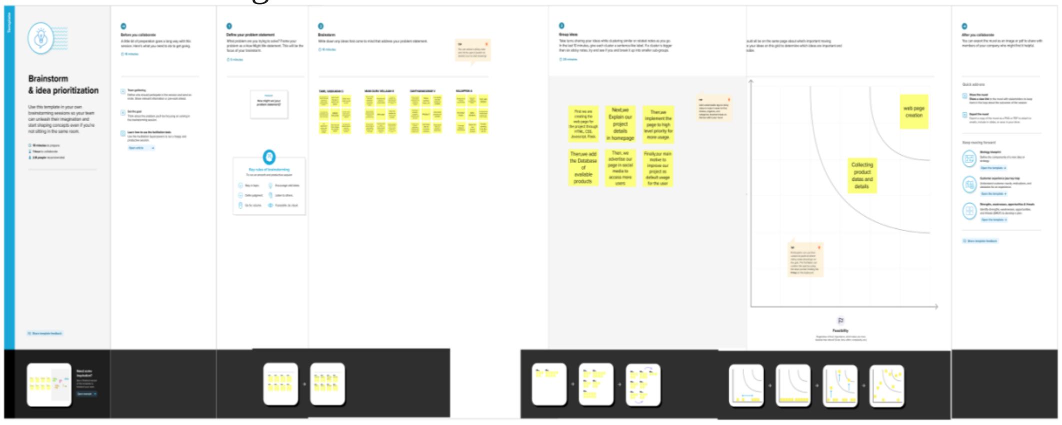


Figure 3.2 Brainstroming

3.3 Proposed Solution

Your proposed solution should relate the current situation to a desired result and describe the benefits that will accrue when the desired result is achieved. So, begin your proposed solution by briefly describing this desired result.

S.No	Parameter	Description
1	Problem Statement (Problem to be solved)	Our application cannot be store the all product in the world level
2	Idea / Solution description	So we are connect the several physical shopping stores. So our application sale product easily and people was purchase product easily.
3	Novelty / Uniqueness	It can promote the best deals and offers on that day.
4	Social Impact / Customer Satisfaction	It purchase the product in any were and any place. So our application give the convenient to purchasing product to customer.
5	Business Model (Revenue Model)	B2C businesses sell directly to their end-users. Anything you buy in an online store as a consumer from wardrobe and household supplies to entertainment is done as part of a B2C transaction
6	Scalability of the Solution	The world is going to fast by the technology. The people was purchase the product by online mode. Our application level in increase in future

Table 3.1 proposed solution

3.4 Problem Solution Fit

CUSTOMER STATE FIT: To explore our product to customer, what can we make for them, and how it suitable for them.	**CUSTOMER LIMITATION:** The application provides the detail about our quality products to the user, it can order from any places , the user can receive the order in correct destination ,if the	**AVAILABLE SOLUTION:** To solve the wrong destination Problem, only one solution is there , which is if your are in another place your product will provided to your neighbor with

	user in wrong destination, the order will not provide to them.	confirmation of OTP
PROBLEMS & PAINS: The main problem for the application is to maintain the user level. Because the every one can purchase the product in every places so our application facing server down in festival time there will be a problem occurs.	**SOLUTION GUESS:** The website also notes that shoppers need not create a profile in the retailer's eCommerce site if they had previously made purchases from the retailer's site. Once the profile is created, the application's AI will access the database of connected retailers and recommend items appropriate to the shopper's specifications.	**HOW WE DIFFER FROM:** We planned to give an ads on every user friendly apps, News papers. Giving awareness to all patients.
WHO IS YOUR CUSTOMER: Everyone are our customers	**LIMITATIONS TO BUY :** This is not a certain need in all peoples life, so there is no limitations.	**COMMUNICATION:** BARRIER: There will creating the server will be difficult and maintenance will be lagging.

Table 3.2 Problem Solution Fit

CHAPTER 4

4. REQUIREMENT ANALYSIS

A solution requirement is aimed at the concerns of the people who will build and deliver the solution. It tells those people what the functional and non-functional requirements for the solution will be and how the solution will deliver on the business and stakeholder requirements.

4.1 Functional Requirements:

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Profile	Confirmation via Mobile phone
FR-4	User required product	Confirmation via Pay on delivery

Table 4.1 Functional Requirement

4.2 Non-functional Requirements:

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

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	Our product used for buying products through online
NFR-2	**Security**	Application Is more secure because of high security database
NFR-3	**Reliability**	User can feel free to upload their profile.
NFR-4	**Performance**	Application can move the product purchase to next level
NFR-5	**Availability**	User can easily Download it on Google Play Store
NFR-6	**Scalability**	This applicationprovide large no of server to show multiple product in a fraction of time

Table 4.2 Non-Functional Requirement

CHAPTER 5

5. PROJECT DESIGN

5.1 Data Flow Diagram

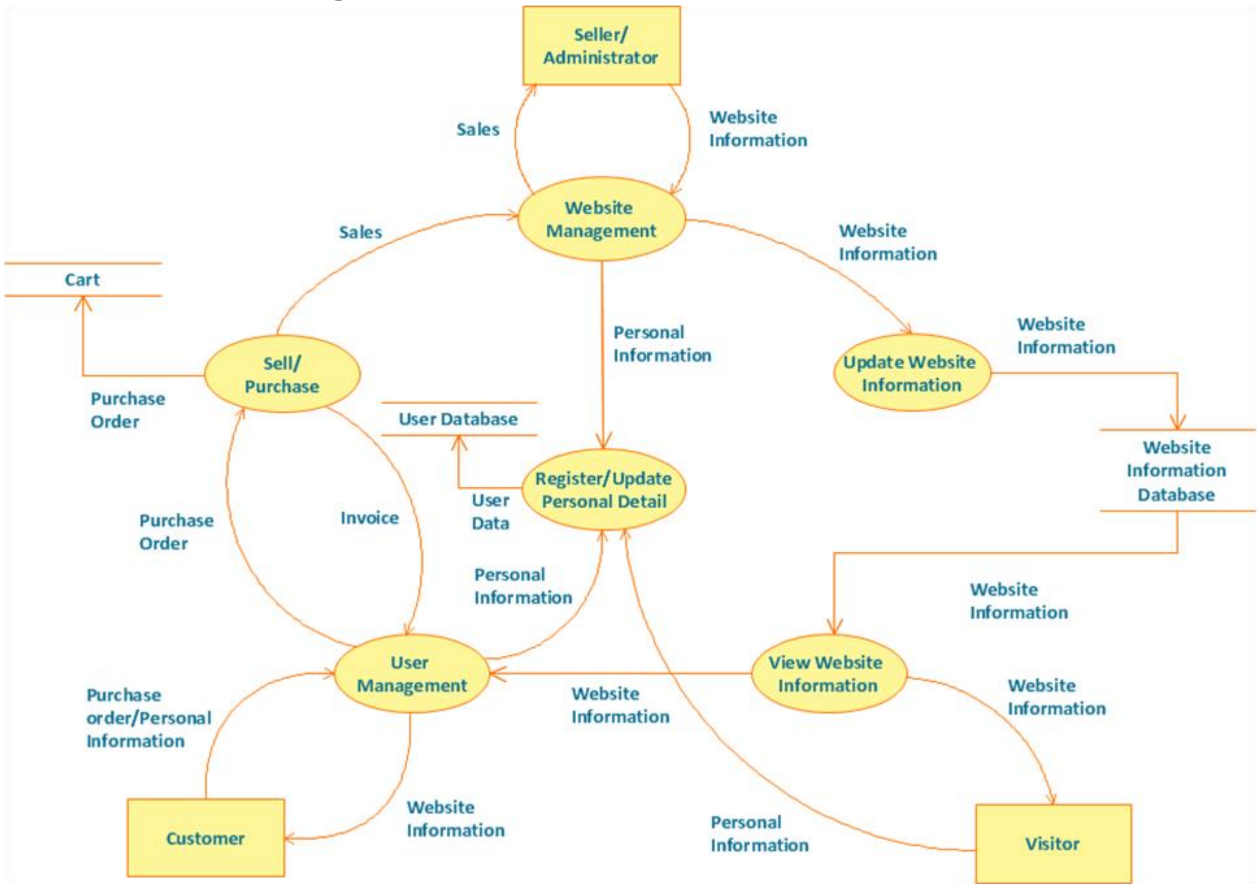


Figure 5.1 Data Flow Diagram

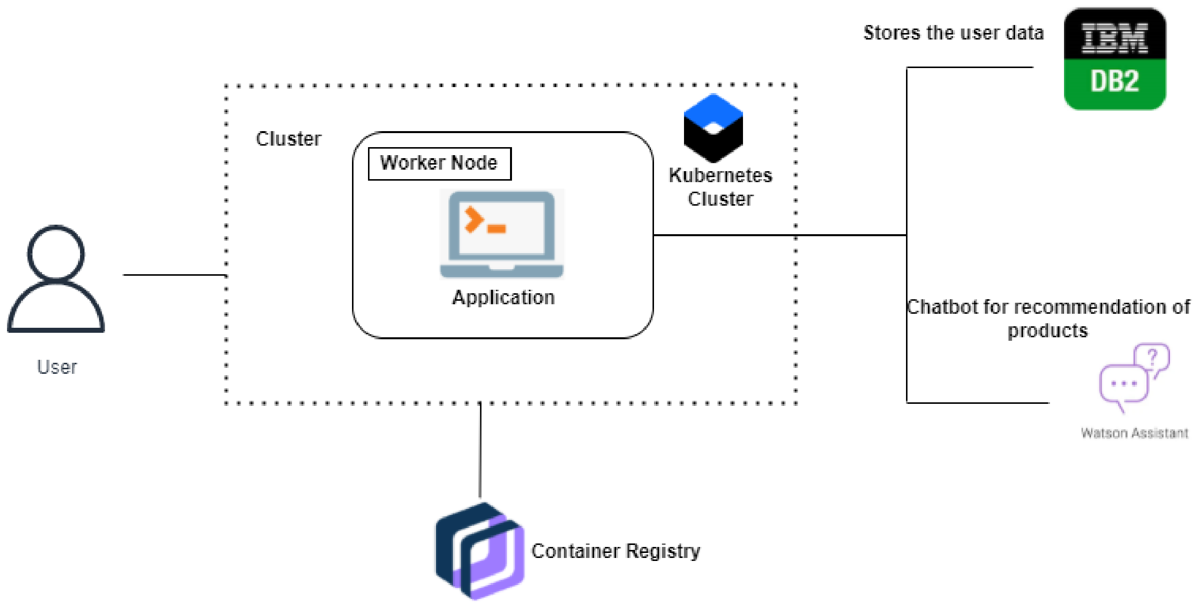


Figure 5.2 Solution and Technical Architecture

5.1 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,Flask etc.
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, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table 5.1 Components & Technologies

5.2 Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	IBM cloud
3.	Scalable	Justify the scalability of	HTML,CSS,JAVASC

	Architecture	architecture (3 – tier, Micro-services)	RIPT,PYTHON(FLASK)
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	HTML,CSS,JAVASCRIPT,PYTHON(FLASK)
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	HTML,CSS,JAVASCRIPT,PYTHON(FLASK)

Table 5.2 Application Characteristics

5.3 User Stories

A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

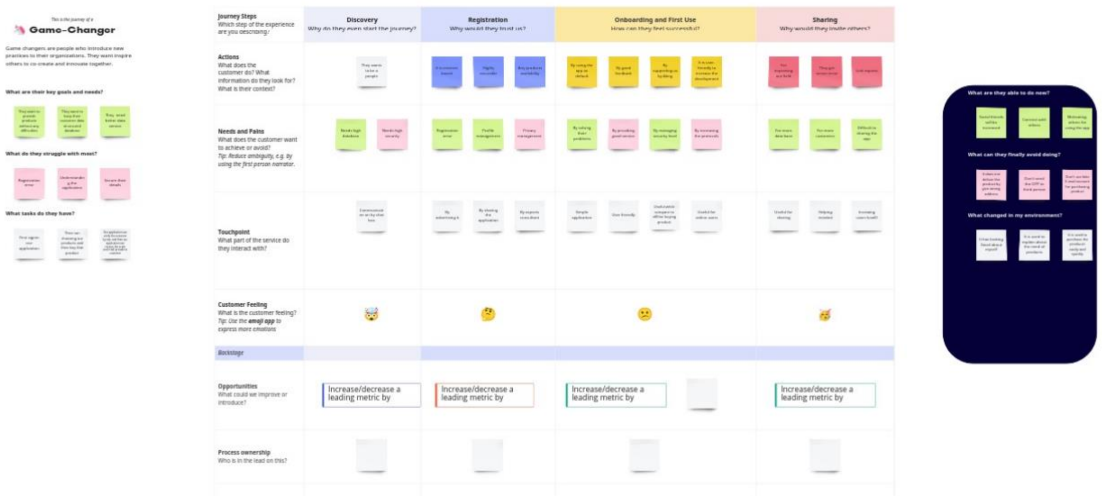

Game-Changer
What are their key goals and needs?
What do they struggle with most?
What tasks do they have?
Journey Steps
Which step of the experience are you describing?
Discovery
Why do they even start the journey?
Registration
Why would they trust us?
Onboarding and First Use
How can they feel successful?
Sharing
Why would they invite others?
Actions
What does the customer do? What information do they look for? What is their context?
Needs and Pains
What does the customer want to achieve or avoid?
Touchpoint
What part of the service do they interact with?
Customer Feeling
What is the customer feeling?
Backstage
Opportunities
What could we improve or introduce?
Increase/decrease a leading metric by
Process ownership
Who is in the lead on this?
What are they able to do now?
What can they finally avoid doing?
What changed in my environment?

Figure 5.3 User Stories

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Delivery Schedule

S.NO	ACTIVITY TITLE	ACTIVITY DESCRIPTION	DURATION
1	Project preparation	Assign team members, Create repository in the GitHub, download rocket-chat essentials and join respective project channel.	1 WEEK
2	Attend class	Attend sessions on IBM, team leader assign task to each member of the project, attendquiz , submit assignment.	1 WEEK
3	Working on different phases of project	Ideation phase-literature survey, Project design phase I-proposed solution, solution architecture, project design phaseII-customer journey ,data flow ,technical architecture, planning phase-milestones, tasks, sprint schedule.	4 WEEK
4	Developing project	Develop the code, test and push it to GitHub, clarify queries.	2 WEEK
5	Budget and scope of project	Analyze and making the project budget and discuss with team	1 WEEK

Table 6.1 Milestone & Activity List

6.2 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation :

Sprint	Functional Requirement(Epic)	User Story Number	User Story/Task	Story point	Priority	Team Members
Sprint-1	Registration	USN-2	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Tamil vasikaran S
Sprint-1		USN-2	As a user, I will receive confirmation email once I have register for the application	1	High	Mani guru vellaian
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Santhanakumar V
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Kaliappan A
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Tamil vasikaran S

Table 6.2 Sprint Delivery Plan

6.3 Project Tracker, Velocity & Burndown Chart

Sprint	Total Story point	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Point Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	26 Oct 2022	31 Oct 2022	20	31 Oct 2022

Sprint-2	20	6 Days	1 Nov 2022	6 Nov 2022	20	6 Nov 2022
Sprint-3	20	6 Days	7 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	13 Nov 2022	18 Nov 2022	20	18 Nov 2022

Table 6.3 Project Tracker, Velocity & Burndown Chart

Velocity: AV = Sprint duration/velocity

= 20/6 = 3.33

6.3 Reports from JIRA

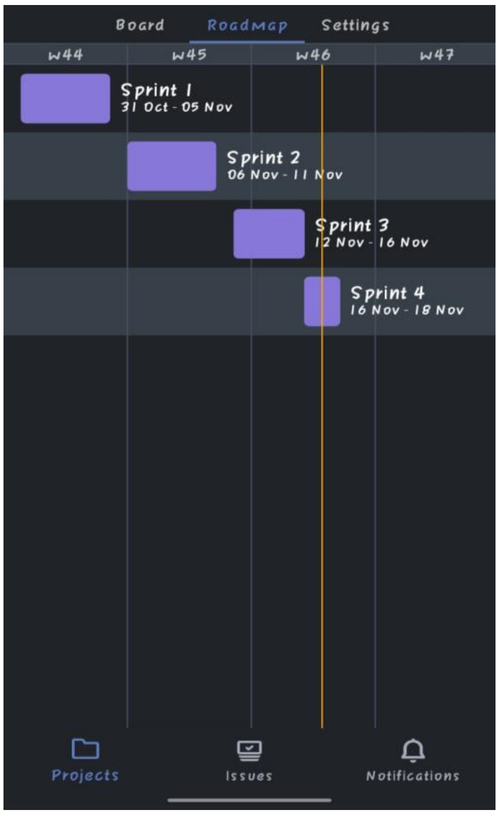


Figure 6.1 Reports from JIRA

CHAPTER 7
7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1 Software Requirement Specification (SRS)

IMPLEMENTING THE WEB APPLICATION:

If you've done a good job planning and designing your site, then building the HTML and CSS will be easier. For many of us, this is the best part.

You will use lots of different technologies to build your site:

- HTML: this is the basis of your website, and if you learn nothing else, you should learn HTML.
- CSS: Once you know HTML, CSS helps you create the design you planned. And CSS is easy to learn.
- Python (flask)
- JavaScript
- My SQL

REGISTRATION PAGE:

Registration page is used for user can register the personal details in our application, once the user can register in our website any time can purchase the product easily can delivered to our destination place.

```
<section class="container-fluid h-100 login-container">
<div class="row h-100 justify-content-center align-items-center">
<div class="col-12 col-md-6 d-none d-md-block"></div>
<div class="col-12 d-flex justify-content-center col-md-6">
<div class="login-form-container p-3">
<h2 class="mb-4 login-text"><img class="login-logo-img"
src="/assets/logo.png" alt="logo">Sign up</h2>
<div>
<div>
<input type="email" class="login-input" placeholder="Email">
</div>
<div>
<input type="text" class="login-input" placeholder="User name">
</div>
<div>
<input type="tel" class="login-input" placeholder="Mobile number">
</div>
<div>
<input type="password" class="login-input" placeholder="password">
</div>
<p class="text-center" style="font-size: 10px;">
            By signup, you agree to our Terms of Use and Privacy Policy.
```

```
</p>
<div class="text-center">
<button class="primary-btn" routerLink="/">Sign up</button>
</div>
<!-- new account -->
<p class="text-center mt-3">Existing User <a routerLink="/login"
                    class="text-decoration-none">Login</a></p>
</div>
</div>
</div>
</div>
</section>
```

Login Page:

The login page allows a user to gain access to an application by entering their username and password or by authenticating using a social media login.

```
<section class="container-fluid h-100 login-container">
<div class="row h-100 justify-content-center align-items-center">
<div class="col-12 col-md-6 d-none d-md-block"></div>
<div class="col-12 d-flex justify-content-center col-md-6">
<div class="login-form-container p-3">
<h2 class="mb-4 login-text"><img class="login-logo-img"
src="/assets/logo.png" alt="logo">Login</h2>
<div>
<div>
<input type="tel" class="login-input" placeholder="Mobile number">
</div>
<div>
<input type="password" class="login-input" placeholder="password">
</div>
<p class="text-center" style="font-size: 10px;">
                 By login, you agree to our Terms of Use and Privacy Policy.
</p>
<div class="text-center">
<button class="primary-btn" routerLink="/">Login</button>
</div>
<!-- new account -->
<p class="text-center mt-3">New to Fashion <a routerLink="/signup"
                    class="text-decoration-none">Create an account</a></p>
</div>
</div>
</div>
</div>
</section>
```

View Product Page:

A product page isa page on a company website that showcases the product inventory a customer is able to buy. It's a page that helps customers decide what they want to buy according to different specifications like price, features, reviews, and product comparison.

```
<section class="container-fluid pt-3 pt-md-4">
<!-- search filters -->
<div class="row">
<div class="col-12 col-md-6">
<div class="d-flex border">
<select name="category" class="border border-end-0 border-dark py-3">
<option value="all">All</option>
<option value="t-shirt">T-shirt</option>
<option value="t-shirt">T-shirt</option>
<option value="t-shirt">T-shirt</option>
</select>
<input type="text" placeholder="search" style="flex: 1;" class="border-0 py-
2">
</div>
</div>
<div class="col-12 col-md-6">
<div class="d-flex justify-content-around justify-content-md-end mt-3 mt-md-
0">
<!-- chat bot -->
<button class="chat-btn me-md-3">Let's chat</button>
<!-- sort by -->
<select name="sort" class="border-0 py-3">
<option value="">Sort by</option>
<option value="low-high">Price: Low to High</option>
<option value="high-low">Price: High to Low</option>
</select>
</div>
</div>
</div>
<!-- product list items -->
<div class="row mt-2">
<div class="col-12">
<div class="d-flex justify-content-between">
<p class="fw-bold">Total products : 545</p>
<p class="d-none d-md-block">Let's make changes in buying</p>
</div>
<app-product-card [product]="productDatails"></app-product-card>
</div>
```

```
<!-- pagination -->
<div class="col-12 mt-5 d-flex justify-content-center">
<nav aria-label="Page navigation example">
<ul class="pagination">
<li class="page-item"><a class="page-link" href="#">1</a></li>
<li class="page-item"><a class="page-link" href="#">2</a></li>
<li class="page-item"><a class="page-link" href="#">3</a></li>
<li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>
</nav>
</div>
</div>
</section>
```

Add Product Page:

A product page is what defines the features, manufacturer, uses and a lot more, about a certain product, in e-commerce. It is a page on an e-commerce website that defines a product in its entirety. This allows the users to look deeply into what a product offers and how it will benefit them once they buy it.

```
<head>
   // other head tags
<link rel="stylesheet" href="css/signup.css">
<link rel="stylesheet" href="css/addProduct.css">
</head>
<body>
<imgsrc="img/loader.gif" class="loader" alt="">

<div class="alert-box">
<imgsrc="img/error.png" class="alert-img" alt="">
<p class="alert-msg"></p>
</div>

<script src="js/token.js"></script>
<script src="js/addProduct.js"></script>
</body>
```

FLASK PROJECT

```
from turtle import st
from flask import Flask, render_template, request, redirect, url_for, session
from markupsafe import escape

import ibm_db
conn = ibm_db.connect("DATABASE=<databasename>;HOSTNAME=<your-
hostname>;PORT=<portnumber>;SECURITY=SSL;SSLServerCertificate=Digi
CertGlobalRootCA.crt;UID=<username>;PWD=<password>",",")
```

```
app = Flask(__name__)



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

@app.route('/addstudent')
def new_student():
  return render_template('add_student.html')

@app.route('/addrec',methods = ['POST', 'GET'])
def addrec():
  if request.method == 'POST':

    name = request.form['name']
    address = request.form['address']
    city = request.form['city']
    pin = request.form['pin']

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

    if account:
      return render_template('list.html', msg="You are already a member, please
login using your details")
    else:
insert_sql = "INSERT INTO students VALUES (?,?,?,?)"
prep_stmt = ibm_db.prepare(conn, insert_sql)
ibm_db.bind_param(prep_stmt, 1, name)
ibm_db.bind_param(prep_stmt, 2, address)
ibm_db.bind_param(prep_stmt, 3, city)
ibm_db.bind_param(prep_stmt, 4, pin)
ibm_db.execute(prep_stmt)

    return render_template('home.html', msg="Student Data saved successfuly..")

@app.route('/list')
```

```
def list():
  students = []
sql = "SELECT * FROM Students"
stmt = ibm_db.exec_immediate(conn, sql)
  dictionary = ibm_db.fetch_both(stmt)
  while dictionary != False:
    # print ("The Name is : ",  dictionary)
students.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)

  if students:
    return render_template("list.html", students = students)

@app.route('/delete/<name>')
def delete(name):
sql = f"SELECT * FROM Students WHERE name='{escape(name)}'"
  print(sql)
stmt = ibm_db.exec_immediate(conn, sql)
  student = ibm_db.fetch_row(stmt)
  print ("The Name is : ",  student)
  if student:
sql = f"DELETE FROM Students WHERE name='{escape(name)}'"
    print(sql)
stmt = ibm_db.exec_immediate(conn, sql)

    students = []
sql = "SELECT * FROM Students"
stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
students.append(dictionary)
      dictionary = ibm_db.fetch_both(stmt)
    if students:
      return render_template("list.html", students = students, msg="Delete
successfully")



  # # while student != False:
  # #   print ("The Name is : ",  student)

  # print(student)
  return "success..."
```

```
# @app.route('/posts/edit/<int:id>', methods=['GET', 'POST'])
# def edit(id):

#     post = BlogPost.query.get_or_404(id)

#     if request.method == 'POST':
#         post.title = request.form['title']
#         post.author = request.form['author']
#         post.content = request.form['content']
#         db.session.commit()
#         return redirect('/posts')
#     else:
#         return render_template('edit.html', post=post)
```

7.2 Feature 2

Integrate Chat Bot Withweb Page:

What is a chatbot?

A chatbot is a support system for your customer service. Using artificial intelligence and natural language processing, your chatbot can simulate conversation with a user through messaging applications, websites, mobile apps and more, giving them accurate and relevant information. By powering your AI chatbot with Watson Assistant, you can avoid the difficulties associated with traditional chatbot building platforms and build a tool that will improve your customer support.

How to add Chatbot to your website

Go to the Integrations section and select Chat Widget.
Click on the Publish your bot section.
Click Copy to clipboard to copy the code.
Paste the code to your website's source code before the /body closing tag.

Integrating SendGrid Service

Create an API key:

Sign in to SendGrid and go to Settings > API Keys.

1. Create an API key.
2. Select the permissions for the key. At a minimum, the key must have Mail send permissions to send email.
3. Click Save to create the key.
4. SendGrid generates a new key.

SendGrid Integration With Python Code

To send emails from the application we need to integrate the SendGrid Service. Please reference the link.
Email API Quickstart for Python
In this quickstart, you'll learn how to send your first email using the Twilio SendGrid Mail Send API and Python.
Prerequisites
Be sure to perform the following prerequisites to complete this tutorial. You can skip ahead if you've already completed these tasks.

5. Sign up for a SendGrid account.
6. Enable Two-factor authentication.
7. Create and store a SendGrid API Key with **Mail Send**>**Full Access** permissions
8. Complete Domain Authentication.
9. Install Python.

Skip the prerequisites
Sign up for a SendGrid account
When you sign up for a free SendGrid account, you'll be able to send 100 emails per day forever. For more account options, see ourpricing page.
Enable Two-factor authentication
Twilio SendGrid requires customers to enable Two-factor authentication (2FA). You can enable 2FA with SMS or by using the Authy app. See the 2FA section of our authentication documentation for instructions.
Create and store a SendGrid API key
Unlike a username and password — credentials that allow access to your full account — an API key is authorized to perform a limited scope of actions. If your API key is compromised, you can also cycle it (delete and create another) without changing your other account credentials.
Visit ourAPI Key documentation for instructions on creating an API key and storing an API key in an environment variable. To complete this tutorial, you can create a Restricted Access API key with **Mail Send**>**Full Access** permissions only, which will allow you to send email and schedule emails to be sent later. You can edit the permissions assigned to an API key later to work with additional services.
Once your API key is assigned to an environment variable — this quickstart uses SENDGRID_API_KEY — you can proceed to the next step.
export SENDGRID_API_KEY=<Your API Key>
Verify your Sender Identity
To ensure our customers maintain the best possible sender reputations and to uphold legitimate sending behavior, we require customers to verify theirSender Identities by completing Domain Authentication. A Sender Identity represents your 'From' email address—the address your recipients see as the sender of your emails.

7.3 Database Schema

- **Simple, functional database structure:** The database table structure is simple but covers all the required functionality without compromising the user experience.
- **High performance:** Database queries execute quickly to facilitate live customer interactions and support a frictionless shopping experience. Therefore, the selected database should have good indexing and performance optimization options.
- **High availability and scalability:** A good database design is highly available with automatic snapshots and enables automatic scaling to support future platform growth as well as sudden traffic spikes.

Based on these characteristics, a good e-commerce database design involves three key parts:

- **Database scope:** The scope refers to the planned functionality of the database. The underlying table structure of the database, its relationships, and indexes all depend on the functionality of the e-commerce platform.
- **Database type:** The type can vary from a relational database to a NoSQL database or a hybrid approach depending on the requirements and the underlying data structure.
- **Database infrastructure:** Your database can be either unmanaged or managed. The former means spinning up your own database service; the latter means using something like Amazon RDS or Amazon DynamoDB

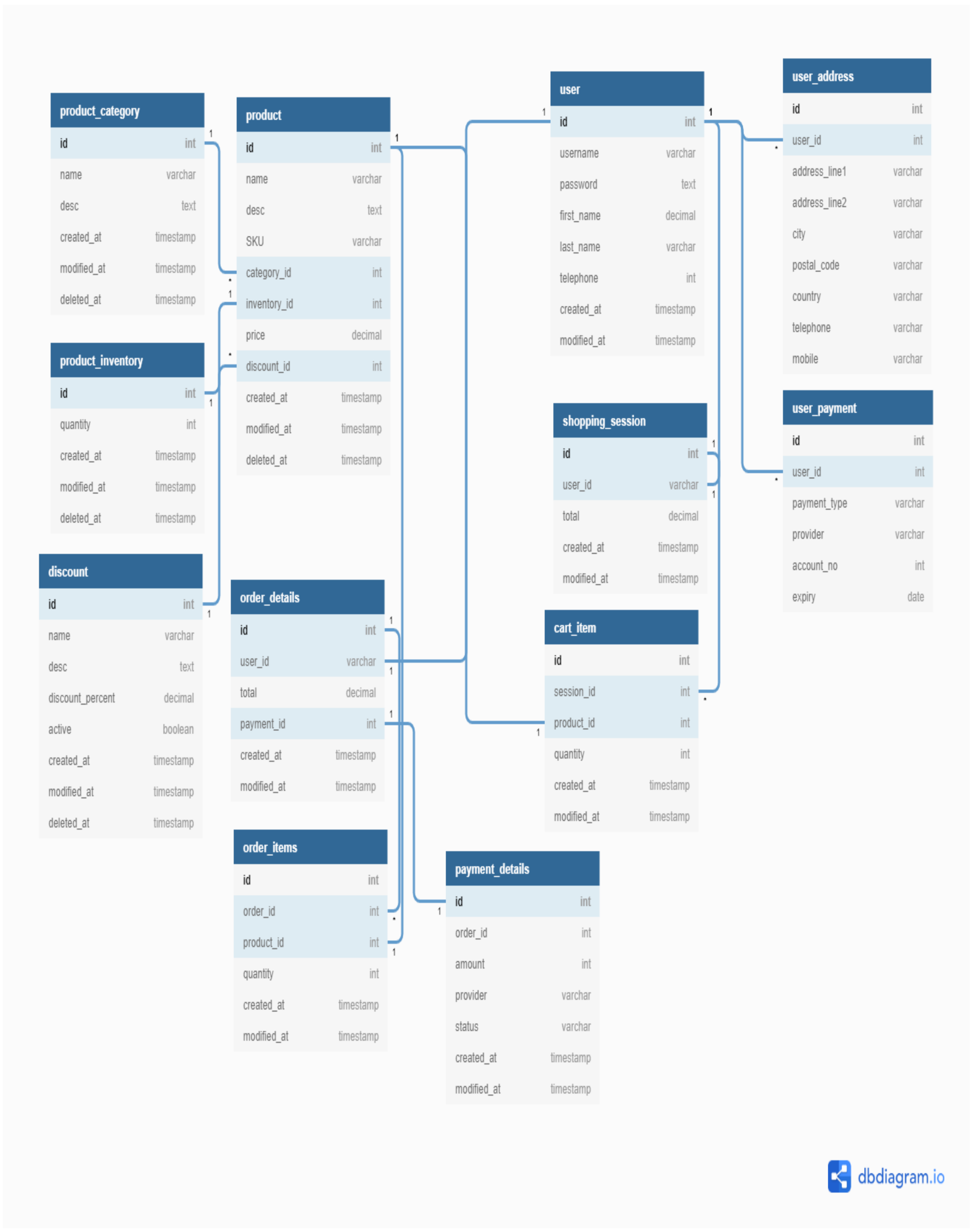


Figure 7.1 Database Schema

CHAPTER 8
TESTING

8.1 Test Cases

- **System is properly linked or not** - Whether they are redirected to desired page or not.
- **Information passed** – If a page passes some parameter to another page then it should be checked that the page get the correct information, whatever is passed by the previous page.
- **Output should be correct** – Every functionality of the system should be checked properly whether it gives the right result or not generally test is performed with known results. If the output of the system is matched with that result the system is working fine.

LOGIN FOR USER

Serial No	**Description**	**Expected Result**	**Actual Result**	**Result**
1.	This page contains 2 fields user name and password and a login button to submit the information. User is entering correct information.	User home page should open after successful login.	Respective user home page is opening after successful login by user.	Passed
2.	If either user name or password is filled incorrect or left	An error message should be	When wrong information is entered by	Passed

	blank.	displayed and user should be asked fill the information again.	user then an error message is displayed.	

Table 8.1 Login For User

STRUCTURAL TESTING

Structural Testing takes into account the internal mechanism of a system or component. Fatigue Testing is carried out with the objective of determining the relationship between the stress range and the number of times it can be applied before causing failure. So when your product’s structural durability needs to be predicted, verified and validated, turn to DTB's Structural Testing and Fatigue Testing experts. We provide you with the necessary structural testing and fatigue testing equipment and personnel to test the design and manufacturing integrity of your product. Call upon our vast experience in commercial and military applications. Software Structural Testingis a 2-day course designed to provide an excellent knowledge base and practical skills for anyone interested in improving Software Structural Testing techniques and practices in their organization. This course starts with an overview of software testing basics, including discussions of the importance of software testing, the different levels of testing and basic testing principles. Basic testing terminology is defined. Techniques for ensure test coverage of requirements, different types of testing documentation and various test activities are discussed.

Course attendees will learn how to utilize various techniques for performing systematic structural testing, including decision/condition coverage, loop testing and basis path testing. Strategies for performing software and system integration testing are also covered.

FUNCTIONAL TESTING

It is very useful and convenient in support of functional testing. Although JMeter is known more as a performance testing tool, functional testing elements can be integrated within the Test Plan, which was originally designed to support load testing. Many other load-testing tools provide little or none of this feature, restricting themselves to performance-testing purposes. Besides integrating functional-testing elements along with load-testing elements in the Test Plan, you can also create a Test Plan that runs these exclusively. In other words, aside from creating a Load Test Plan, it also allows you to create a **Functional Test Plan**. This flexibility is certainly resource-efficient for the testing project.

This will give a walkthrough on how to create a Test Plan as we incorporate and/or configure its elements to support functional testing. This created a Test Plan for a specific target web server. We will begin the chapter with a quick overview to prepare you with a few expectations; we will create a new Test Plan, only smaller. The Test Plan we will create and run at the end of this chapter will incorporate elements that support functional testing, exclusively.

8.2 User Acceptance Testing

The objective of this step is to produce a set of test data that may be used to test the system. Whenever a new system is developed it need to be tested to confirm its validity and to determine whether it meets the user requirements. The system was also tested with some sample records. The records were entered into the system and various reports were generated to check the system.

System testing is a critical phase of implementation. Testing of the system involves hardware devices and debugging of computer programs and testing information processing procedures. Testing can be done with test data, which attempt to simulate all possible condition that may rise during processing. The testing methods adopted during the testing of system are unit testing and integration testing.

UNIT TESTING

Unit testing focuses on the modules independently locate the errors. This enables the tester to detect errors in coding. It is the process of taking a module and running it in isolation from rest of the software product by using prepared test cases and comparing the actual result with the result redirected with the specifications and design of the module. One purpose of testing is to find and remove as many errors in the software as practical. There are number of reason in support of unit testing-:

- The size of module single module is small that we can locate an error fairly easily.
- The module is small enough that we can attempt to test it in some demonstrably exhaustive fashion.
- Confusing interactions of multiple errors in widely different parts of software are eliminated.

There are problem associated with testing a module in isolation. How do we run a module without anything to call it, to be called by it, possibly to output intermediate values obtained during execution? One approach is to construct an appropriate driver routine to call it, and simply stubs to be called by it, and to insert output statements in it. Stubs serve to replace modules that are subordinate to the module to be tested. A stub or dummy subprogram uses the subordinate module’s interface, may do minimal data manipulation, prints verification of entry and returns.

INTEGRATION TESTING

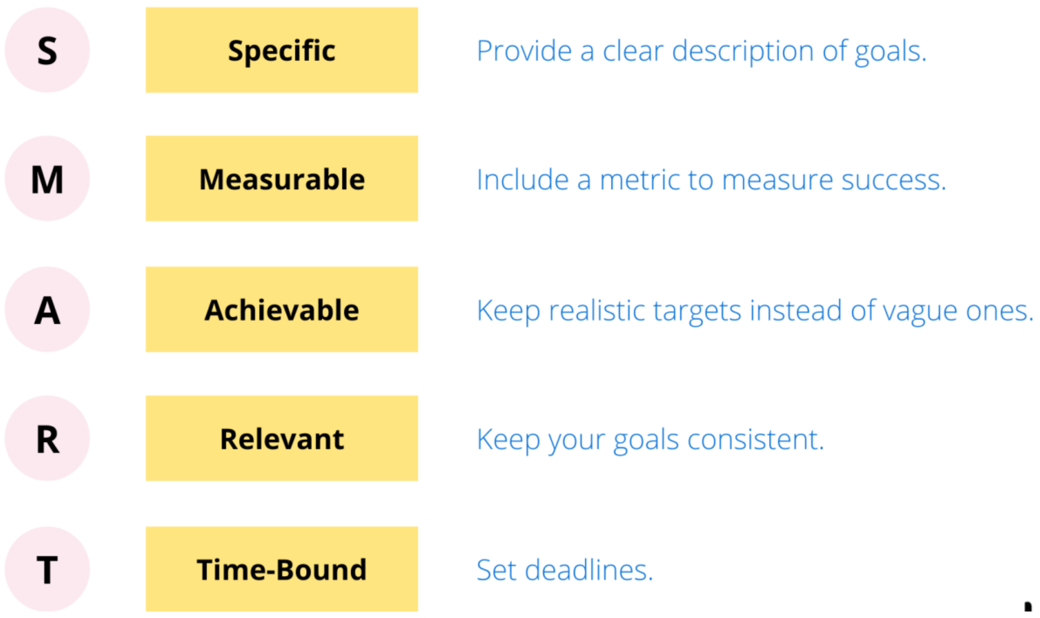
This is a systematic technique for constructing the program structure while at the same time to uncover the errors associated with the interface. The objective is to take unit tested module and build a program structure that has been detected by designing. The main purpose of integration testing is to determine that the interfaces between modules are correct or not. One specific target of integration testing is the interface: whether parameter matches on both sides as to type, permissible ranges, meaning & utilization. There are 3 types of integration testing-

- **Top Down Approach-** Top Down integration proceeds down the invocation hierarchy, adding one module at a time until an entire tree level is generated.
- **Bottom Up Approach** – The Bottom up strategy works similarly from the bottom to up.
- **Sandwich Strategy** – A sandwich strategy runs from top and bottom simultaneously.

CHAPTER 9

9.RESULTS

9.1 Performance Metrics



S	Specific	Provide a clear description of goals.
M	Measurable	Include a metric to measure success.
A	Achievable	Keep realistic targets instead of vague ones.
R	Relevant	Keep your goals consistent.
T	Time-Bound	Set deadlines.

Figure 9.1 Performance Metrics

CHAPTER 10

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Faster buying process.
- Store and product listing creation.
- Cost reduction.
- Affordable advertising and marketing.
- Flexibility for customers.
- No reach limitations.
- Product and price comparison.
- Faster response to buyer/market demands.

DISADVANTAGES

1. Delay in delivery. ...
2. Lack of significant discounts in online shops. ...
3. Lack of touch and feel of merchandise in online shopping. ...
4. Lack of interactivity in online shopping. ...
5. Lack of shopping experience. ...
6. Lack of close examination in online shopping. ...
7. Frauds in online shopping.

CHAPTER 11

11.CONCLUSION

Technology has made significant progress over the years to provide consumers a better online shopping experience and will continue to do so for years to come. With the rapid growth of products and brands, people have speculated that online shopping will overtake in-store shopping. While this has been the case in some areas, there is still demand for brick and mortar stores in market areas where the consumer feels more comfortable seeing and touching the product being bought. However, the availability of online shopping has produced a more educated consumer that can shop around with relative ease without having to spend a large amount of time. In exchange, online shopping has opened up doors to many small retailers that would never be in business if they had to incur the high cost of owning a brick and mortar store. At the end, it has been a win-win situation for both consumer and sellers.

CHAPTER 12

12.FUTURE SCOPE

1. Omnichannel presence and support

Figure 12.1 Omnichannel presence and support

Today, people carry out research, consideration, and purchase across multiple channels. So, they expect seamless across these channels and devices. E-commerce businesses should therefore make themselves omnichannel-ready. You need to provide customers with the content they want at the time they want, and the place they want it.

So, you need to adopt some of the latest technologies that perform well in an interconnected buyer or user journey to tap into the *scope of e-commerce*. Some of these include-

- Video Chat that lets your brand converse face-to-face with customers.
- Co-browsing- a visual engagement system that simultaneously brings together your customers and agents on the same page so that the agents can smoothly guide them through complicated procedures.
- Screen Sharing- an interactive technique where your customers can share their screen with your agents so that their issues with completing transactions, filling forms, etc. can be resolved.
- Document Interaction- a platform where your agents can safely and securely interact with your customers' documents.

2. Customer experience

Figure 12.2 Customer experience

Customer experience, both in-store and online, matters a lot for the success of your e-commerce business. No wonder it's increasingly becoming one of the best digital marketing trend.

You can ensure a good customer experience online in the following ways.

- Ensure that your site loads fast. Today people expect sites to load within 3 seconds. If not, they lose patience and move on to a fast-loading site.

- Take care that your site navigation is simple and easy. User testing your site can help you identify issues that people face in navigating your site. This will help you create a customer-friendly experience.

- A/B test your messaging, product page flow, info asked for during the checkout, checkout process length, and call to action. You can also improve your SEO with A/B testing.

- Post only high-quality product images. People seek high-resolution images to determine whether or not to make a purchase.

- Provide apt and honest product descriptions. This will help build your customers' trust. One of the most common e-commerce marketing mistakes is relying on manufacturers' descriptions. So, avoid it and post your own clear, professional description for each product in your catalog.

- Integrate educational content into your site. People make purchases based on blog content.

- Provide ways for customers to share their product reviews and ratings. This not only improves your credibility but also allows other customers to make informed buying decisions.

3. High levels of personalization

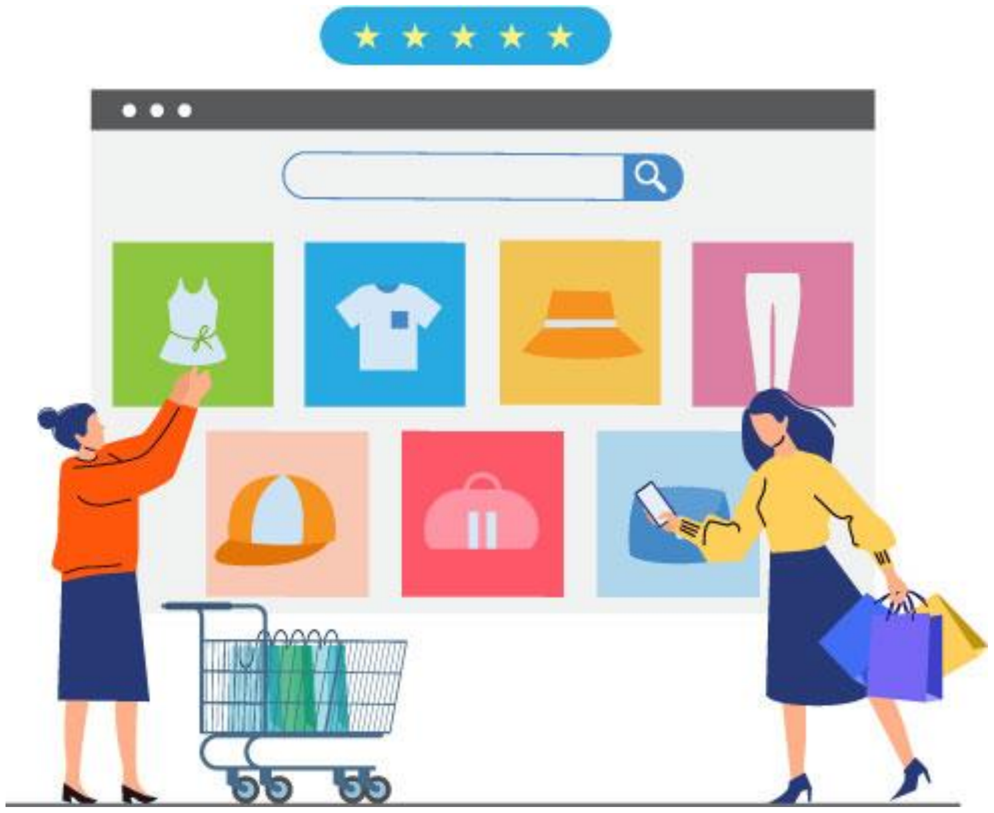


Figure 12.3 High levels of personalization

The biggest trend in e-commerce now is personalization. People expect a shopping experience that is highly relevant to them based on their personal preferences. Studies say that over 78% of customers ignore impersonalized offers. So, personalize your **customer interactions** based on their browsing behavior, past purchases, and preferences specified.

Employing Artificial Intelligence (AI) and machine learning (ML) can capture almost all user actions online, store those, and derive valuable insights from those. This lets businesses know **customer behavior** patterns, expectations, desires, and more. This in turn creates endless possibilities (like upselling, cross-selling, etc.) for e-commerce businesses. Such smart personalization can bring in a 20 fold return, says a study by Liveclicker, a digital marketing solutions provider.

4. Mobile-friendliness

Figure 12.4 Mobile-friendliness

Today, many people use their smartphones to shop online. Online purchases made using smartphones in 2021 are $345 billion (Source: *Statista*).

People increasingly resort to using their mobile phones for making purchases as these are more convenient than desktops. So, the scope of e-commerce businesses is largely a mobile-first approach. Those that haven't made their e-stores mobile-friendly would be losing a lot of business opportunities.

A big part of making your e-store website mobile-friendly is ensuring its responsiveness. That is, users should be able to view and interact with elements thereof without having to manipulate the view. They need not resize, scroll, zoom, or pan their screen. Your site's elements will automatically adjust and rearrange as per the screen of the device on which it is viewed. Hence, your customers will enjoy a good UX at your e-commerce site.

Another aspect of mobile-friendliness is synchronization between your mobile and desktop site. That is, the actions that your customer has taken on your desktop site should reflect on the mobile site, and vice versa. For instance, if she has added some items to her cart on the desktop, the info should update on the mobile. This way, there should be a seamless shopping experience across devices.

5.Image recognition turned product recognition

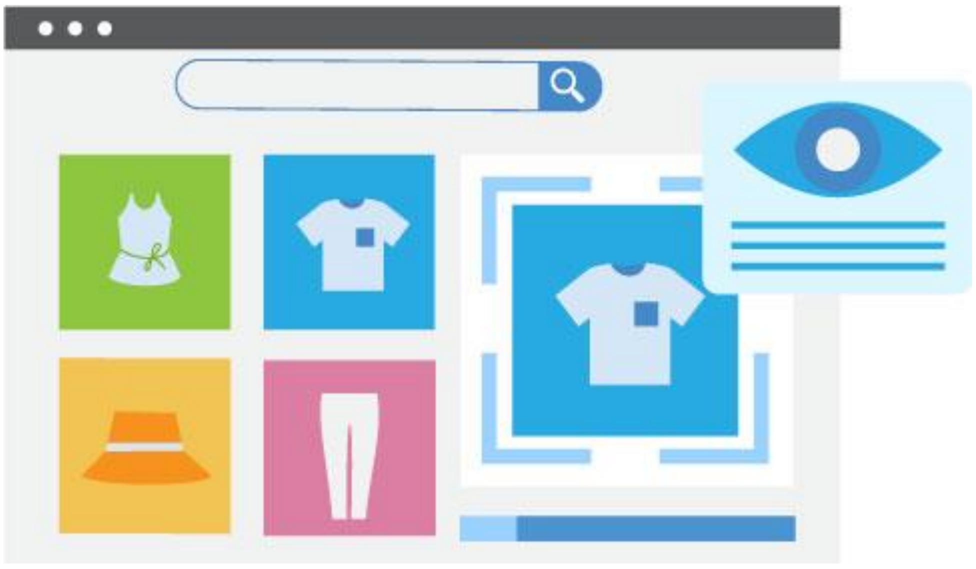


Figure 12.5 Image recognition turned product recognition

The recently introduced image recognition functionality in smartphones has been extended to e-commerce as well. Some of the well-performing e-commerce businesses are already tapping into its immense potential. It’s used to recognize products and items in an image and pick them out from its vast catalogs.

People can now present the scanned image or photograph of the product they’re looking for to an e-store. The image recognition functionality incorporated will immediately scan through all of its products to find matches. This is especially highly useful in the clothing sector. So, this trend will soon become the mainstream as people are finding this useful technology an increasingly indispensable feature in their shopping.

CHAPTER 13

13 APPENDIX

Source Code:

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")
```

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"))
```

Test.py

```
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)
```

GitHub & Project Demo link

GitHub : https://github.com/IBM-EPBL/SI-GuidedProject-43623-1665945038

Demo Link:https://drive.google.com/file/d/1CD9uRqFSveWHhz9zG0PR7ojcs-P50aWh/view?usp=share_link