SMART FASHION RECOMMENDER SYSTEM

1. INTRODUCTION

2. LITERATURE SURVEY

3. IDEATION & PROPOSED SOLUTION

4. REQUIREMENT ANALYSIS

5.PROJECT DESIGN

6. PROJECT PLANNING & SCHEDULING

1. INTRODUCTION

1.1 Project Overview

Fashion applications have seen tremendous growth and are now one of the most used programs in the e-commerce field. The needs of people are continuously evolving, creating room for innovation among the applications. One of the tedious processes and presumably the main activities is choosing what you want to wear. Having an AI program that understands the algorithm of a specific application can be of great aid. We are implementing such a chat bot, which is fed with the knowledge of the application's algorithm and helps the user completely from finding their needs to processing the payment and initiating delivery. It works as an advanced filter search that can bring the user what they want with the help of pictorial and named representation. The application also has two main user interfaces - the user and the admin. The users can interact with the chat bot, search for products, order them from the manufacturer or distributor, make payment transactions, track the delivery, and so on. The admin interface enables the user to upload products, find how many products have been bought, supervise the stock availability and interact with the buyer regarding the product as reviews

1.2 Purpose

In E-commerce websites, users need to search for products and navigate across screens to view the product, add them to the cart, and order products. The smart fashion recommender application leverages the use of a chat bot to interact with the users, gather information about their preferences, and recommend suitable products to the users. This application has two predefined roles assigned to the users. The roles are customer and admin. The application demands redirection of the user to the appropriate dashboard based on the assigned role. Admin should be able to track the number of different products and admin should be assigned the responsibility to create products with appropriate categories. The user should be able to mention their preferences using interacting with chat bots. The user must receive a notification on order confirmation/failure. The chat bot must gather feedback from the user at the end of order confirmation. The main objective of this application is to provide better interactivity with the user and to reduce navigating pages to find appropriate products.

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

1.Paper Title: A COMPREHENSIVE REVIEW ON ONLINE FASHION RECOMMENDATION Publication: December 2020

Author name: Samit Chakraborty

Methodology: Auto Regression (AR) and Linear Regression Model. Auto Regression (AR) and Linear Regression Model Using photos pulled from social media, online fashion magazines, well-known e-commerce sites, fashion site blogs, and discussion forums, (Ngai et al., 2018) employed the autoregressive (AR) model (or ARMAX) to forecast style or trends. Due to the data patterns being obtained over a set amount of time, it makes precise trend prediction possible (Fung, Wong, Ho, & Mignolet, 2003). These forecasting models' detailed theoretical contents were demonstrated in two separate studies by Liu et al. (2013) and Nenni, Giustiniano, & Pirolo (2013), which also included several general approach forms. Because they were straightforward, quick, wellinformed, and simple to understand, statistical techniques including auto-regression, exponential smoothing, ARIMA, and SARIMA were frequently employed to assess the sales of clothing. A technique for forecasting retail products was proposed by Demerit (2018). weekly using linear regression models in multi-processing groups with both positive and negative commodities. The introduction of dynamic pricing models to support markdown choices in multi-item group predictions has since followed. In order to prevent overfitting, grouping items in predictive models can be seen as a way of variable selection. They then exhibited regression results from multiple-item groupings on the real-world dataset provided by a clothing company in addition to the findings from the single-item regression model. They also revealed the results of markdown optimization for single items and groups of multiple items that serve as the foundation for multi-item forecasting models. The results suggested that regression models provide better estimates in many categories than the one-item model.

2.Paper Title: Image-based fashion recommender system.

Publication: Year (2021).

Author name: Shaghayegh Shirkhani.

Methodology: Collaborative filtering, the iterative filtering process, matrix factorization, and content-based systems. Systems for collaborative filtering make product recommendations based on user similarity metrics and/or by grouping things from similar users' purchases. Despite the variety of collaborative filtering methods, many widely used systems can be distilled down to just two steps: 1. Seek out users who have similar rating tendencies to the active user (the user whom the prediction is for). 2. To establish a prediction for the active user, utilise the ratings from the users who shared your interests in step one.

3.Paper Title: Fashion Recommendation Systems

Author name: Samit Chakraborty , Md. Saiful Hoque, Naimur Rahman Jeem, Manik Chandra Biswas, Deepayan Bardhan and Edger Lobaton.

Methodology: Fast fashion has grown significantly over the past few years, which has had a significant impact on the textile and fashion industries. An effective recommendation system is needed in e-commerce platforms where there are many options available to sort, order, and effectively communicate to user’s pertinent product content or information. Fast fashion retailers have paid a lot of attention to image-based fashion recommendation systems (FRSs), which offer customers a customised purchasing experience. There aren't many academic studies on this subject, despite its enormous potential. The studies that are now accessible do not conduct a thorough analysis of fashion recommendation systems and the accompanying filtering methods. This review also looks at many potential models that might be used to create future fashion suggestion systems.

4.Paper Title: A Review on Clothes Matching and Recommendation System Based on User Attributes

Author name: Atharv Pandit , Kunal Goel , Manav Jain , Neha Katre

Methodology: It's crucial to dress adequately while venturing out into the real world. The confidence of the individual is raised and a very positive impression is made when they are dressed appropriately in clothing that exhibits some degree of style and is worn in a way that complies with societal norms. The goal of the study is to make it easier for customers to locate the best-fitting outfits by taking into account fine elements like style, patterns, colours, and textures, as well as user characteristics like age, skin tone, and favourite colours. It seeks to assist the user in organising their closet and making stylish clothing selections. It makes an effort to assist the user in dressing appropriately for the occasion and in finding clothing that complements their personal style. In order to create a robust system that discovers the user's matching outfits and provides recommendations, an in-depth analysis of numerous systems that are built for various aspects is undertaken in this research. Systems created to propose clothing using various methodologies have been researched, with both their benefits and drawbacks highlighted. It has also been investigated how to make clothing detecting systems user-friendly while accepting feedback from the user.

5.Paper Title: Individualized fashion recommender system

Year: 10 October 2020

Author name: M Sridevi, N ManikyaArun, MSheshikala and E Sudarshan

Methodology: This design seeks to use an image of a product provided by the stoner as input to prompt recommendations because people frequently see things that they're interested in and tend to look for products that are similar to those. We reuse the Deep Fashion Dataset (DFD) photos using neural networks, and we generate the final suggestions using a closest neighbour backed recommender

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

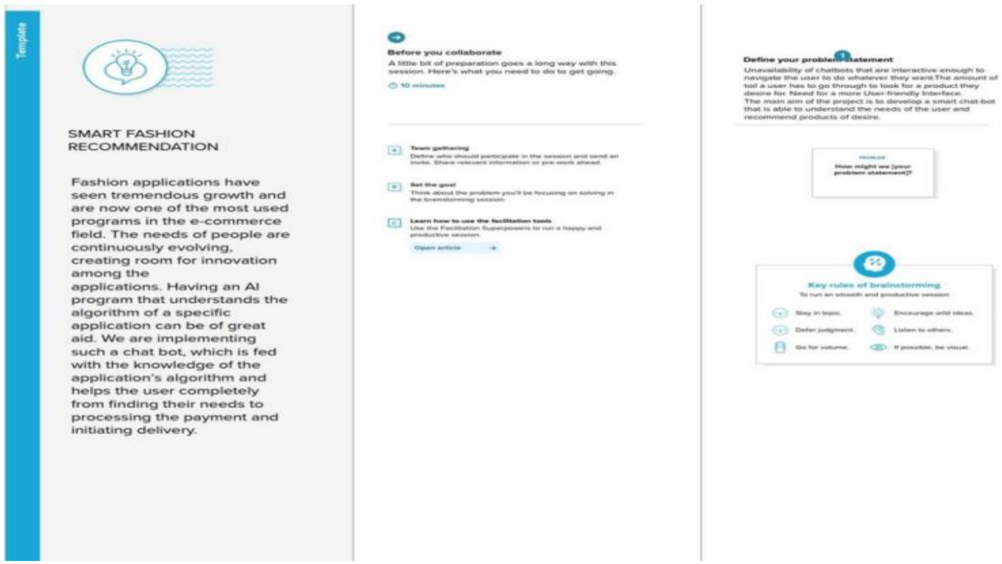
3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

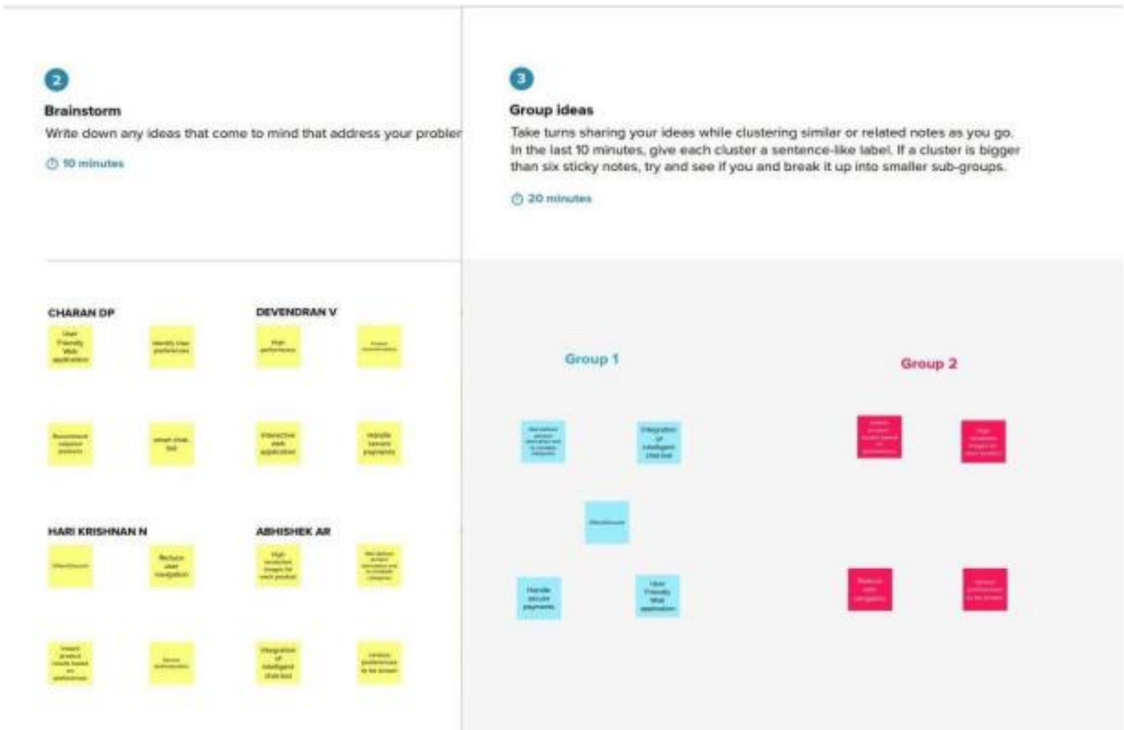
3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

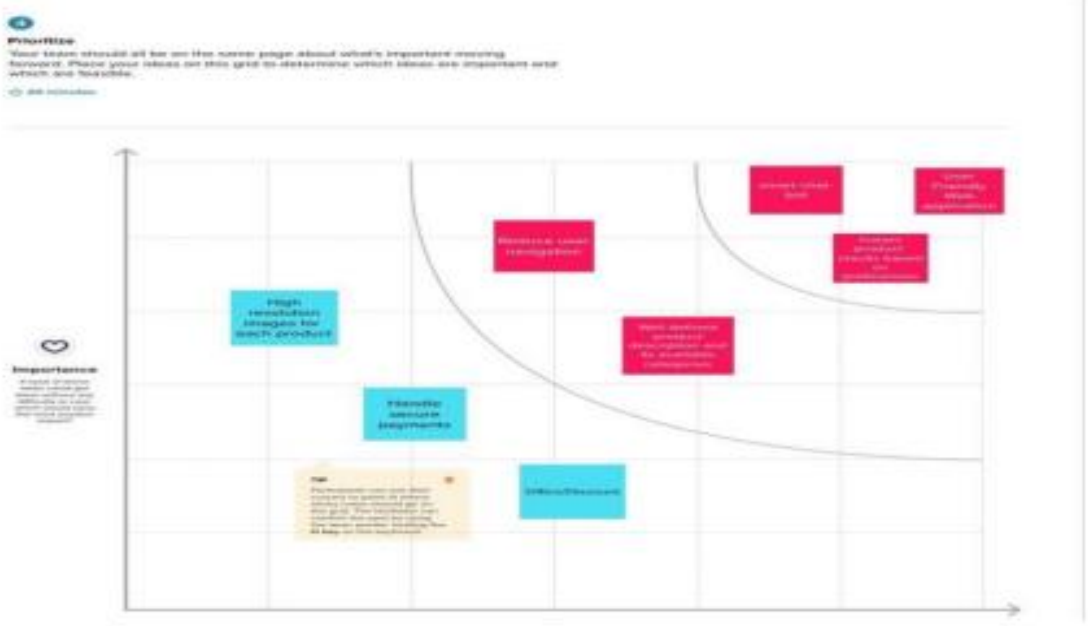
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Lack of interaction between application and user • User need to navigate across multiple pages to choose right product • Confusion in choosing product • Lack of sales • Complex User Interface. • Lack of proper guidance.
2.	Idea / Solution description	By using Smart fashion recommender application: • Improve customer relationship, interactivity and services. • Effective recommendation of products. • Recommendation within a single page via chat-bot • Collect feedback instantly. • Reduce human error • Proper guidance in accessing application.
3.	Novelty / Uniqueness	• Chat-bot asks and learns from user preference which recommends appropriate products to the user without making them to search through various filters. Reduces time in choosing right product thus increases sales.
4.	Social Impact / Customer Satisfaction	• Feedback from the user at the end of session or after placing order is one of the most important factor in deriving customer satisfaction and providing better services.
5.	Business Model (Revenue Model)	• The application can be developed at minimum cost with high performance and interactive user interface.
6.	Scalability of the Solution	• The solution can be made scalable by using micro service architecture provided that each server responsible for certain functionality of the application. Storing user preferences along with product in browser cookie will enable to provide response instantly and allows for fetching related products.

3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications.

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
FR-2	User Interaction	Interact through the Chat Bot
FR-3	Buying Products	Through the chat Bot Recommendation
FR-4	Track Products	Ask the Chat Bot to Track my Orders
FR-5	Return Products	Through the chat Bot
FR_6	New Collections	Recommended from chat Bot

4.2 Non-Functional requirements

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Using Android or IOS or windows applications.
NFR-2	**Security**	The user data is stored securely in IBM cloud.
NFR-3	**Reliability**	The Quality of the services are trusted.
NFR-4	**Performance**	Its Provide smooth user experience.
NFR-5	**Availability**	The services are available for 24/7.
NFR-6	**Scalability**	Its easy to scalable size of users and products.

5.PROJECT DESIGN

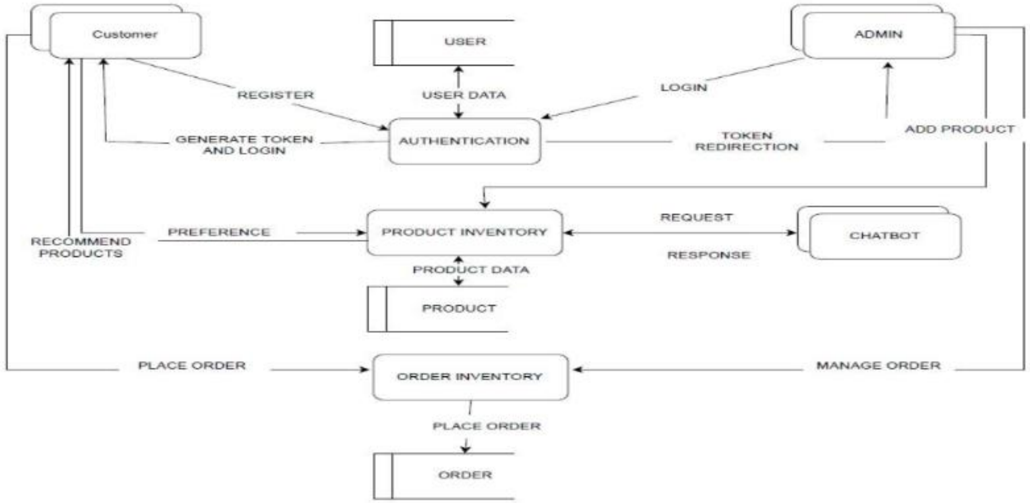
Project design is an early phase of the project lifecycle where ideas, processes, resources, and deliverables are planned out. A project design comes before a project plan as it’s a broad overview whereas a project plan includes more detailed information.

There are seven steps involved when creating a project design, including defining goals and using a visual aid to communicate objectives

These visual elements include a variety of met hods such as Gantt charts, Kanban boards, and flowcharts. Providing a visual representation of your project strategy can help create transparency between stakeholders and clarify different aspects of the project, including its overall feasibility.

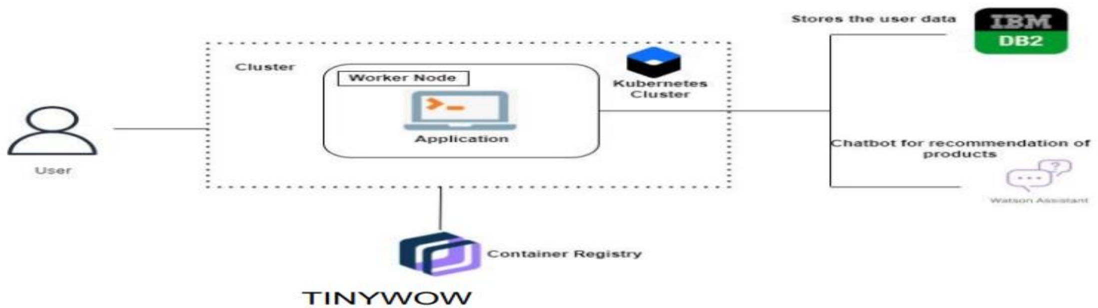
5.1 Data Flow Diagrams

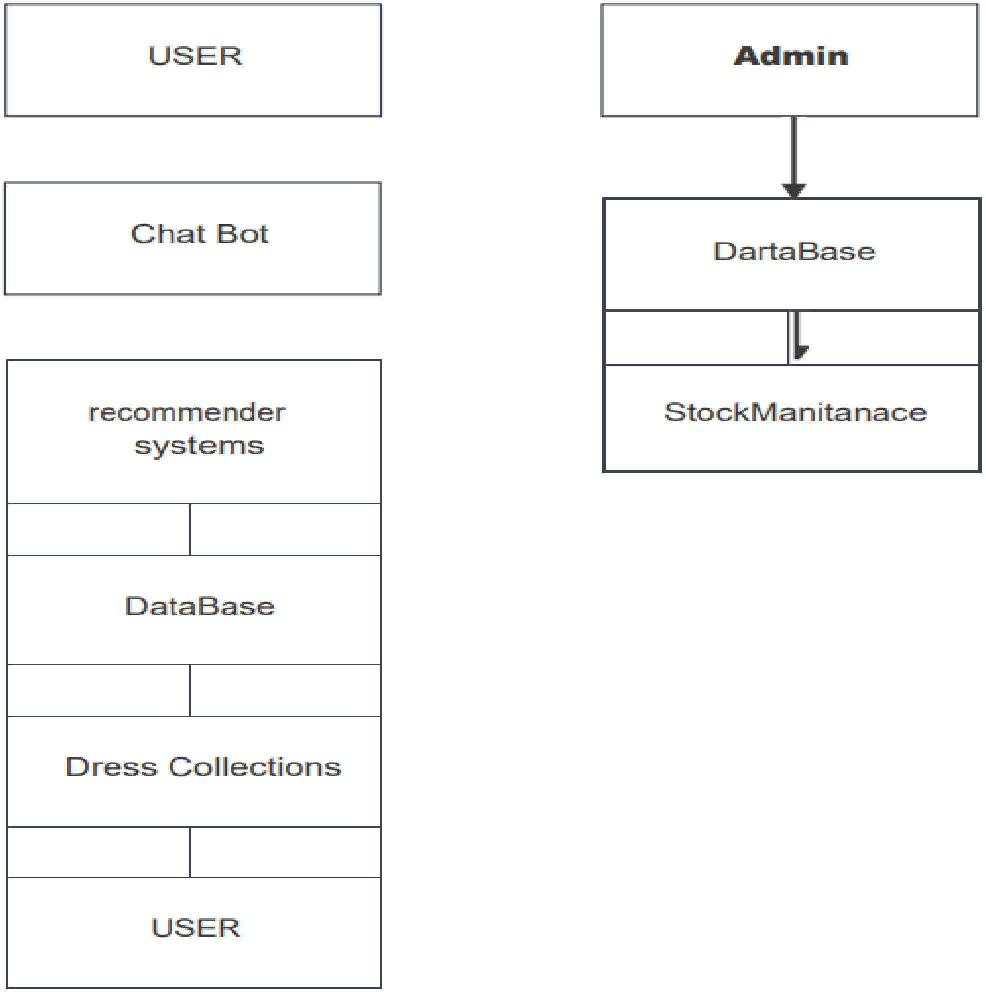
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture

SOLUTION ARCHTECTURE:




USER
Chat Bot
recommender systems
DataBase
Dress Collections
USER
Admin
DartaBase
StockManitanace

TECHNICAL ARCHITECTURE:

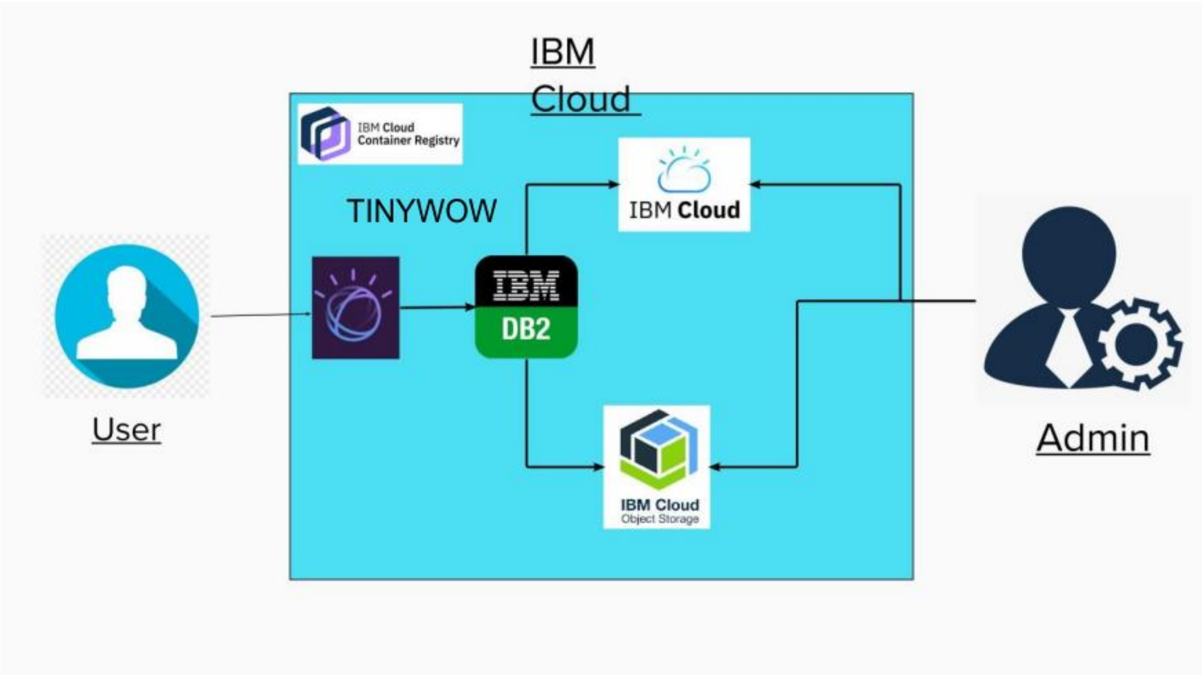


Table-1 : Components & Technologies:

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

Table-2: Application Characteristics:

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

5.3 User Stories

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.

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access and make purchases.	High	Sprint-1
	Dashboard					
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
		USN-5	As a user, I can log into the application by entering email & password	I can access and make purchases.	High	Sprint-1
Administrator	Login	USN-1	I enter my mail and password on organisation's approval	I can approve products and purchases	High	Sprint-1 Administrator

6. PROJECT PLANNING & SCHEDULING

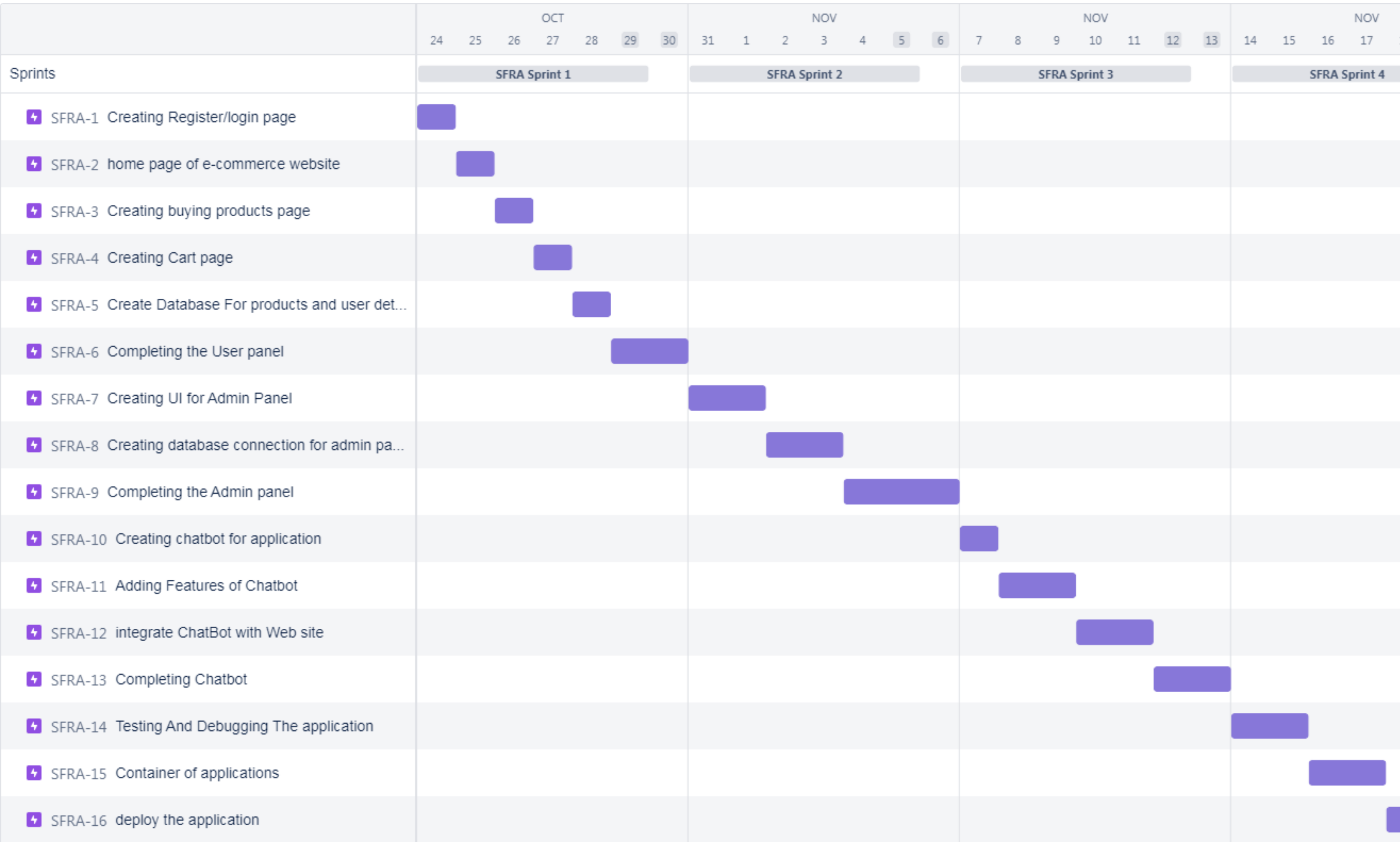
Product Backlog, Sprint Schedule, and Estimation (4 Marks)

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	NITHYA THENMOZHI SHUBHASHREE JAYASURYA

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	NITHYA THENMOZHI SHUBHASHREE JAYASURYA
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	NITHYA THENMOZHI SHUBHASHREE JAYASURYA
Sprint-4	final delivery	USN-4	Container of applications using docker kubernetes and deployment the application. Create the documentation and final submit the application	20	High	NITHYA THENMOZHI SHUBHASHREE JAYASURYA

Burndown Chart:



6.1 Sprint Planning & Estimation

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (a Planned End
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	
Sprint-4	20	6 Days	14 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)

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

7. CODING & SOLUTIONING

HOME PAGE CODE:

```
<html>
   <head>
      <meta name="viewpoint" content="width=device-width, initial-scale=1.0">
      <title>FASHION VIBE</title>
      <style>
       *{
   margin: 0;
   padding: 0;
   font-family:"Century Gothic", CenturyGothic, AppleGothic, sans-serif,'Courier New',
Courier, monospace;
   box-sizing: border-
   box; background:
   fixed;

}

.footer
{
   width:100
   %;   height:
   20%;
   display:flex
   ;
   background:#121212;
```

```
    margin-top: 5%;
    color: #7DE5ED;
    padding-left:
    22%; align-items:
    center; text-align:
    center;

}

.hero{
    width:
    100%;
    height:
    auto;
    background-color:#8f8f8f;
    color: #525252;

}
nav{
    background:
    #7DE5ED; width:
    100%;
    padding: 10px 10%;
    display: flex;
    align-items: center;
    justify-content: space-
    between; position: fixed;

}
.logo{

    width:200px
    ;
```

```
   height:50px
   ;
   text-
   decoration:none;
   text-align: center;
   color:#001128;
}
.user-pic{
   width:
   40px;
   border-radius:
   50%; cursor:
   pointer; margin-
   left: 30px;
}
nav ul{
   width: 100%;
   text-align:
   right;
   font-weight: bold;
}
nav ul li{
   display: inline-
   block; list-style:
   none; margin:
   10px 20px;
}
nav ul li a{
   color: rgb(252, 252,
   5); text-decoration:
   none;
}
.Banner
```

```
{
   float:left;
   width: 100%;
   height:
   400px;
   background-color:
   #121212; color:
   #7DE5ED;
   margin-top:
   10%; text-align:
   center;
}
.Bannerimg1
{
   float: left;
   width:50%;
   height:
   400px;
   background-color: #525252;
}
.Bannerimg2
{
   float: right;
   width:50%;
   height:
   400px;
   background-color: #525252;
}
.Adcontent
{
   width:45%;
   height: 400px;
   margin-left:
```

```
  55%; color:
  #7DE5ED; text-
  align: center;
  padding: 100px;
}
.Adcontent2
{
  width:45%;
  height: 400px;
  margin-left: 5%;
  color:
  #7DE5ED; text-
  align: center;
  padding: 100px;
}

.columnst {
  float:left;
  margin-
  left:7%;
  margin-top:
  10%;
  width:230px;
  height:400px;
  background-
  color:transparent; border:
  2px solid #74bde0; border-
  color:transparent;


}
.column {
  float:left;
  margin-
```

```
    left:7%;
    margin-top:
    10%;
    width:230px;
    height:400px;
    background-
    color:transparent; border:
    2px solid #74bde0; border-
    color:transparent;
}
.columnend
{
    float:left;
    margin-
    left:7%;
    margin-top: 10%;
    margin-bottom:
    10%; width:230px;
    height:400px;
    background-
    color:transparent; border:
    2px solid #74bde0; border-
    color:transparent;

}
.Bottom
{
    height:50px
    ;
    width:230px
    ;
    text-align: center;
    margin-top:300;
```

```
    background:
    #000000e8; color:
    rgb(252,252,5);
    padding: 5%;
}
.depimg
{
    float:left;
    width:228px;
    height:300p;
    background-
    color:transp
    arent;
    border: 2px
    solid
    #74bde0;
    border-
    color:transp
    arent;

}
.image
{
    width: 100%;
    height: 100%;
    object-fit:
    contain;
}
.search{
    width: 330px;
    margin-left:
    40%; color:
    #7DE5ED;
```

```
    position:fixed;
}

.srch{
    width:
    200px;
    height:
    40px;
    background: #7DE5ED;
    border: 2px solid
    #121212; margin-top:
    13px;
    margin-right:13px
    ; color: #FCE700;
    font-size: 16px;
    align-items:
    center; padding:
    10px;
    border-bottom-left-radius:
    25px; border-top-left-radius:
    25px; border-bottom-right-
    radius: 25px; border-top-right-
    radius: 25px;

}

.btn{
    width: 60px;
    height:
    40px;
    border: 2px solid
    #000000dd;
    background:#000000dd;
```

```
    margin-top: 13px;
    color: rgb(252, 252,
    5); align-items:
    center; font-size:
    15px;
    border-bottom-left-radius:
    25px; border-top-left-radius:
    25px; border-bottom-right-
    radius: 25px; border-top-right-
    radius: 25px;
}

.btn:focus{
    outline:
    none;
}

.srch:focus{
    outline: none;
}
.sub-menu-wrap{
    position:absolut
    e; top: 100%;
    right: 2%;
    width:
    320px;
    max-height:
    0px; overflow:
    hidden;
    transition: max-height 0.5s;


}
.sub-menu-wrap.open-
    menu{ max-height:
```

```
   400px;
}
.sub-menu{
   background:rgb(252, 252,
   5); padding: 20px;
   margin: 10px;
   border-radius:
   8%;

}
.user-info{
   display:
   flex;
   align-items: center;
}
.user-info h3{
   font-weight: 500;
}
.user-info
   img{ width:
   60px;
```

```
  border-radius:
  50%; margin-
  right: 15px;
}
.sub-menu hr{
  border: 0;
  height: 1px;
  width:
  100%;
  background:
  #525252; margin:
  15px 0 10px;
}

.sub-menu-
  link{ display:
  flex;
  align-items: center;
  text-decoration:
  none; color:
  #525252; margin:
  12px 0 ;
}
.sub-menu-link
  p{ width:
  100%;
}
.sub-menu-link img{
  width: 40px;
  background:
  #e5e5e5; border-
  radius: 50%;
  padding: 8px;
```

```
    margin-right: 15px;

}

.sub-menu-link spfont-size: 22px; transition: transform 0.5s;
}

.sub-menu-link:hover

    span{ transform:

    translateX(5px);

}

.sub-menu-link:hover

    p{ font-weight: 600;

}


      </style>

    </head>


    <body>


         <nav>

             <a class="logo" href="MadFinalhome.html"><h2>FASHION VIBE</h2></a>

             <ul>

                <li><input class="srch" type="search" name="" placeholder="TYPE TO SEARCH">

                <a href="#"><button class="btn">SEARCH</button></a></li>

                <li><a href="#">HOME</a></li>

                <li><a href="#">FEATURES</a></li>

                <li><a href="#">ABOUT</a></li>


             </ul>

             <img src="https://storagedemo-madzh.s3.jp-tok.cloud-object-
storage.appdomain.cloud/images/profile.jpeg" class="user-pic"
onclick="toggleMenu()">


             <div class="sub-menu-wrap" id="subMenu">
```

```
          <div class="sub-menu">
            <div class="user-info">
              <img src="https://storagedemo-madzh.s3.jp-tok.cloud-
object- storage.appdomain.cloud/images/profile.jpeg">
              <h2>NAME</h2>
            </div>
            <hr>

            <a href="#" class="sub-menu-link">
              <img src="https://storagedemo-madzh.s3.jp-tok.cloud-
object- storage.appdomain.cloud/images/profile.jpeg">
              <p>EDIT PROFILE</p>

            </a>

            <a href="#" class="sub-menu-link">
              <img src="https://storagedemo-madzh.s3.jp-tok.cloud-
object- storage.appdomain.cloud/images/settings.jpeg">
              <p>SETTING & PRIVACY</p>

            </a>

            <a href="#" class="sub-menu-link">
              <img src="https://storagedemo-madzh.s3.jp-tok.cloud-
object- storage.appdomain.cloud/images/help.jpeg">
              <p>HELP</p>

            </a>

            <a href="/Login" class="sub-menu-link">
```

```
                <img src="https://cdn-icons-png.flaticon.com/512/56/56805.png">
                <p>LOGOUT</p>

              </a>
            </div>
          </div>

        </nav>

      <div class="Banner">
        <div class="Bannerimg1"> <img img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/fashionimagebanner.webp">
</div>
          <div class="Adcontent">
            <h1><br>THE JOY OF DRESSING IS AN ART.</br></h1>
            <br>Let's have a look on it---></br>
          </div>
      </div>

      <div class="rowstart">
          <div class="columnst"> <div class="depimg"><img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/sarees.webp">
</div> <div class="Bottom">WEDDING SAREES</div> </div>

          <div class="columnst"> <div class="depimg"><img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/Salwar%20kameez.webp"> </div> <div
class="Bottom">SALWAR KAMEEZ</div> </div>

          <div class="columnst"> <div class="depimg"><img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/Kurtis.webp"> </div> <div
class="Bottom">CASUAL KURTIS</div> </div>
```

```
        <div class="columnst"> <div class="depimg"><img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/bridal%20lehenga.webp">
</div> <div class="Bottom">BRIDAL LEHENGA</div> </div>

      </div>


      <div class="Banner">

        <div class="Bannerimg2"> <img img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/lovablekidsattire.webp">
</div>

        <div class="Adcontent2">

          <h1 class="kids"><br>LOVABLE KIDS ATTIRE</br></h1>

          <br>-------Smiles are always in FASHION</br>

        </div>

      </div>


      <div class="row">


        <div class="column"> <div class="depimg"><img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/Modern%20vibe.webp">
</div> <div class="Bottom">MODERN VIBE</div> </div>


        <div class="column"> <div class="depimg"><img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/festivemood.webp">
</div> <div class="Bottom">FESTIVE MOOD</div> </div>


        <div class="column"> <div class="depimg"><img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/skinny%20dress.webp">
</div> <div class="Bottom">SKINNY DRESS</div> </div>


        <div class="column"> <div class="depimg"><img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/Max%20girls.webp">
</div> <div class="Bottom">MAX GIRLS</div> </div>

      </div>
```

```
<div class="Banner">

  <div class="Bannerimg1"> <img img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/mensfashion.webp">
</div>

  <div class="Adcontent">

    <h1><br>HANDSOME MEN ATTIRE</br></h1>

    <br>Always DRESS well, Keep it SIMPLE but SIGNIFICANT.     </br>

  </div>

</div>


<div class="row">

  <div class="column"> <div class="depimg"><img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/polo%20t%20shirts.webp"> </div> <div
class="Bottom">POLO T-SHIRTS</div> </div>


  <div class="column"> <div class="depimg"><img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/menhoodies.webp">
</div> <div class="Bottom">HOODIES</div> </div>


  <div class="column"> <div class="depimg"><img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/men%20casuals.webp">
</div> <div class="Bottom">MEN CASUALS</div> </div>


  <div class="column"> <div class="depimg"><img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/formal%20shirts.webp">
</div> <div class="Bottom">FORMAL SHIRTS</div> </div>

</div>


<div class="Banner">

  <div class="Bannerimg2"> <img class="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/adornments.webp">
</div>

  <div class="Adcontent2">

    <h1><br>PERSONAL ADORNMENTS</br></h1>
```

```
            <br>ADORNMENT is never anything except a REFLECTION of the HEART!!!</br>
          </div>
        </div>

        <div class="rowend">
        <div class="columnend"> <div class="depimg"><img
class="file:///C:/Users/ADMIN/Desktop/SPRINT%202/women%20ornmanets.webp">
</div> <div class="Bottom">JEWELLERY</div> </div>

        <div class="columnend"> <div class="depimg"><img class="image"
src="C:\Users\ADMIN\Desktop\SPRINT 2\watch.jpg"> </div> <div
class="Bottom">WATCHES</div>
</div>

        <div class="columnend"> <div class="depimg"><img class="image"
src="image"
src="file:///C:/Users/ADMIN/Desktop/SPRINT%202/hanbags.webp"> </div>
<div class="Bottom">HANDBAGS</div> </div>

        <div class="columnend"> <div class="depimg"><img class="image"
src="https://encrypted- tbn0.gstatic.com/images?q=tbn:ANd9GcQSWDKgpQeZ-
3VNR7- 9SfaVGVvqOawrkZiLdNfSpjNNQJNI6hl8cJg0Qs_DZfpJtizUst0&usqp=CAU">
</div> <div class="Bottom">HANDBAGS & CLUTCHES</div> </div>

        </div>

    <script>
        let subMenu =
        document.getElementById("subMenu"); function
        toggleMenu(){
          subMenu.classList.toggle("open-menu");
        }
    </script>
    </body>
```

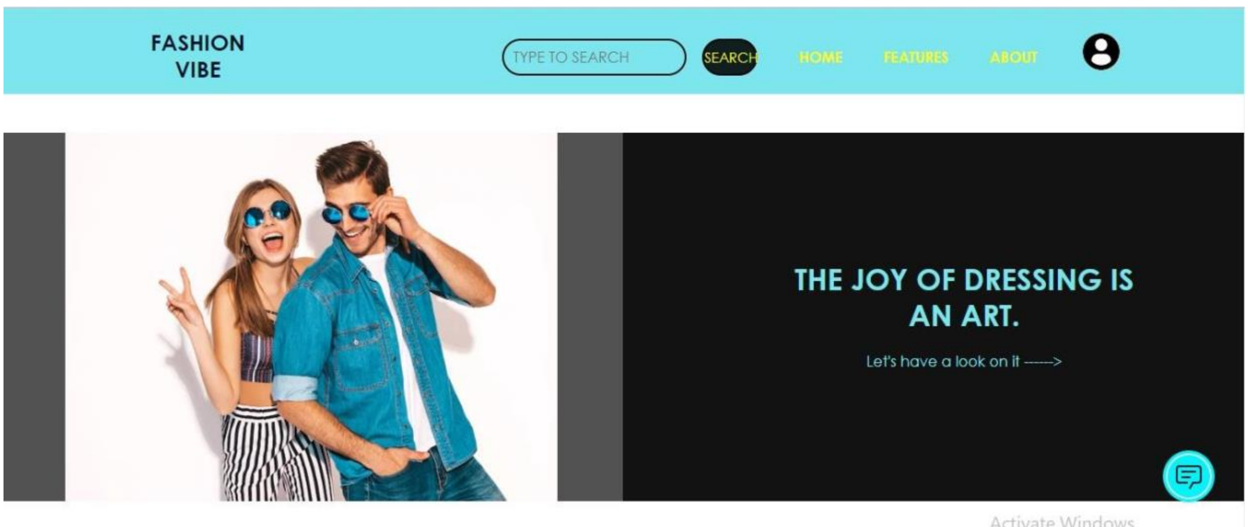
```
<footer>

  <div class="footer"> <H1>THANK YOU FOR PURCHASING.  WELCOME
  AGAIN!!!!</H1></div>


</footer>

</html>
```

OUTPUT:



,FASHION

.:?IBE

TYPE TO SEARCH SEARCH

WEDDING SAREES

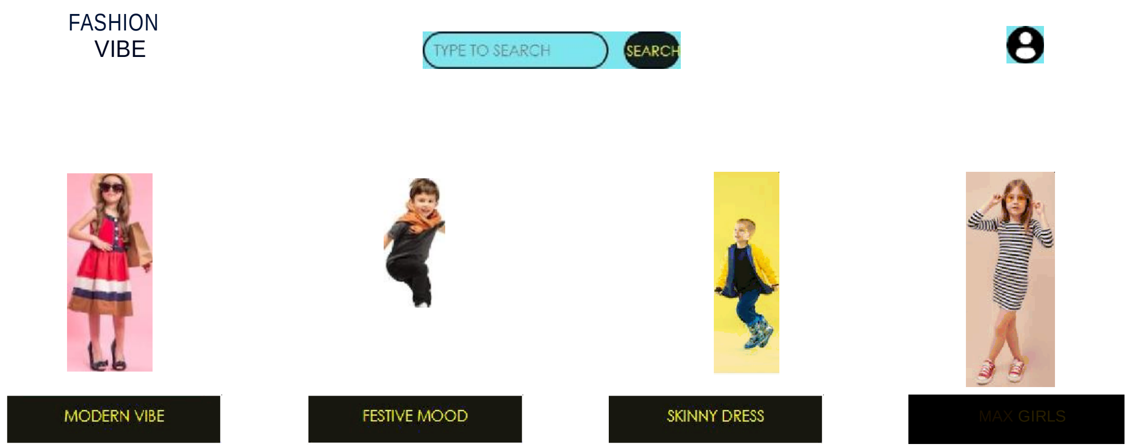
SALWAR KAMEEZ

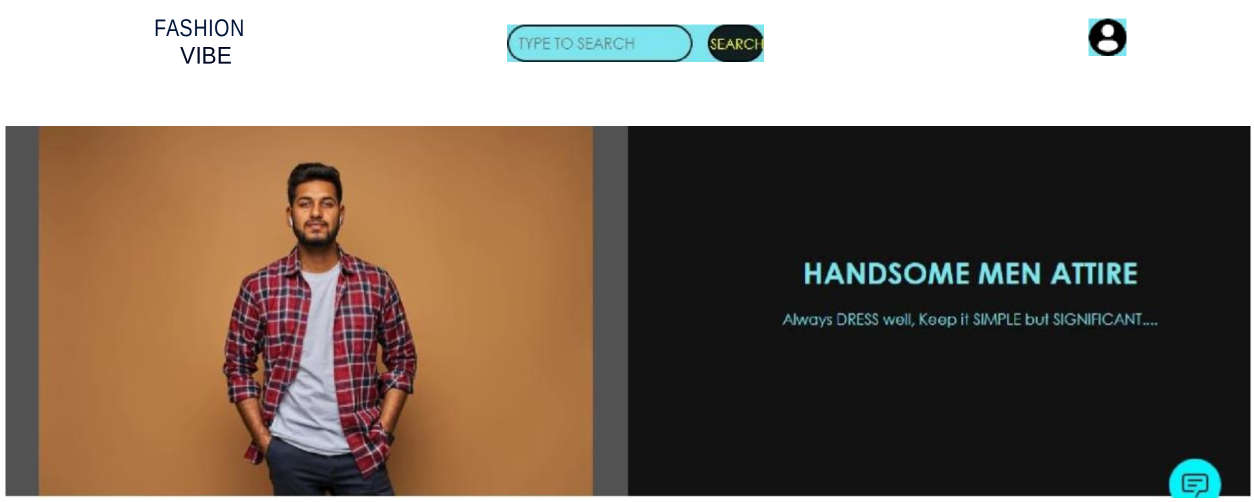
CASUAL KURTIS

FASHION.
VIBE

TYPE TO SEARCH SEARCH

LOVABLE KIDS ATTIRE


FASHION
VIBE
TYPE TO SEARCH
SEARCH
MODERN VIBE
FESTIVE MOOD
SKINNY DRESS
MAX GIRLS


FASHION
VIBE
TYPE TO SEARCH
SEARCH
HANDSOME MEN ATTIRE
Always DRESS well, Keep it SIMPLE but SIGNIFICANT....

FASHION
VIBE

TYPE TO SEARCH SEARCH

POLO T-SHIRTS

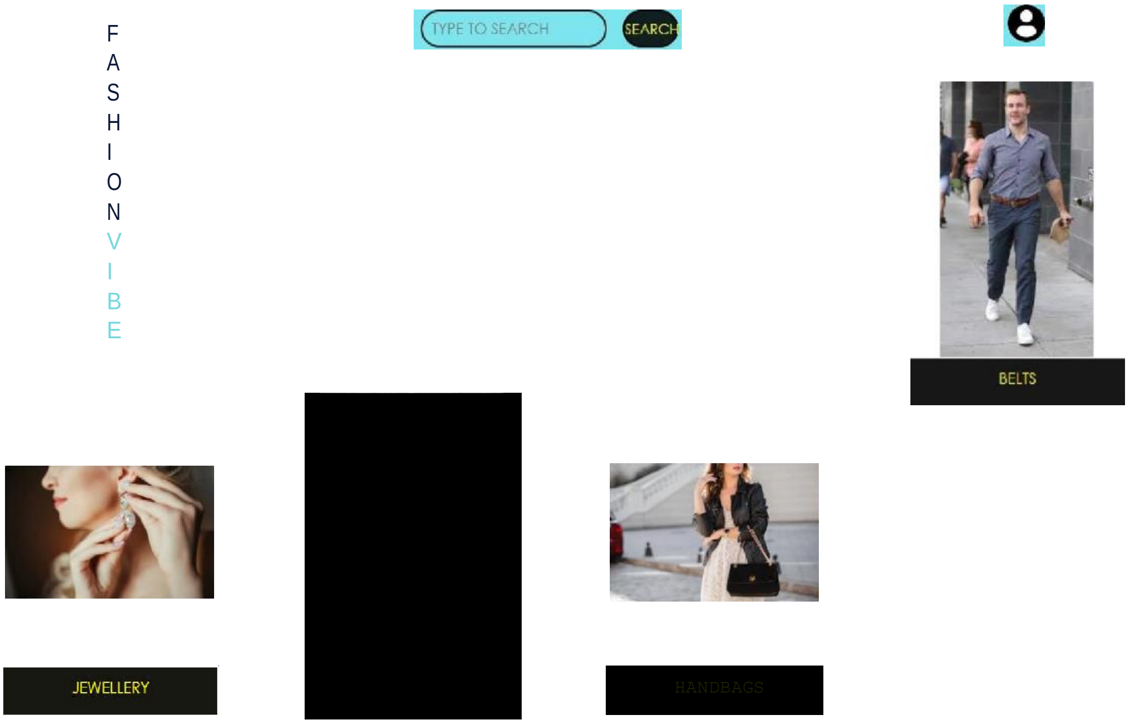
HOODIES

MEN CASUALS

FORMAL SHIRTS

FASHION
VIBE

TYPE TO SEARCH SEARCH



THANK YOU FOR PURCHASING. ... WELCOME AGAIN!!!!

7.1 Feature 1

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")
```

7.2 Feature 2

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
```

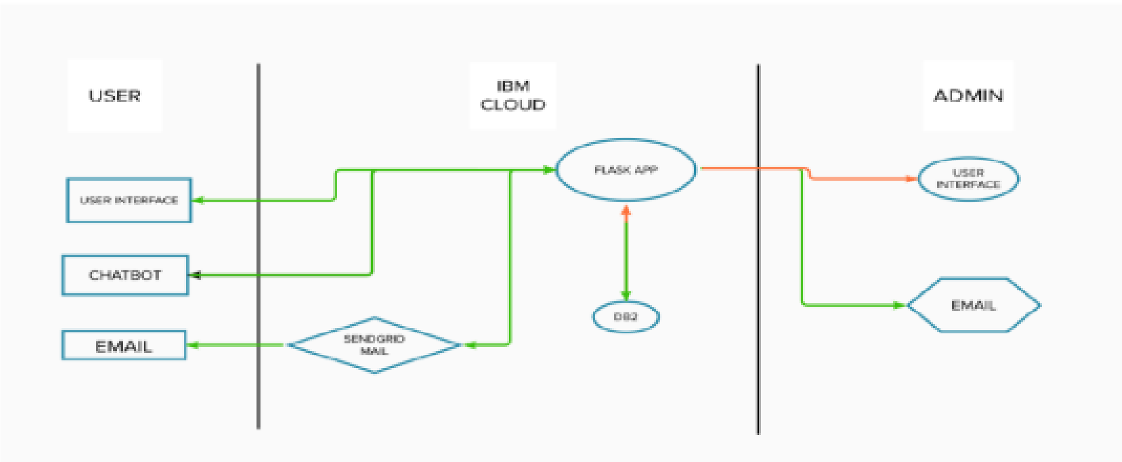
```
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.models import Sequential
from numpy.linalg import norm
from sklearn.neighbors import NearestNeighbors
import cv2
features_list = pickle.load(open("image_features_embedding.pkl", "rb"))
img_files_list = pickle.load(open("img_files.pkl", "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)
```

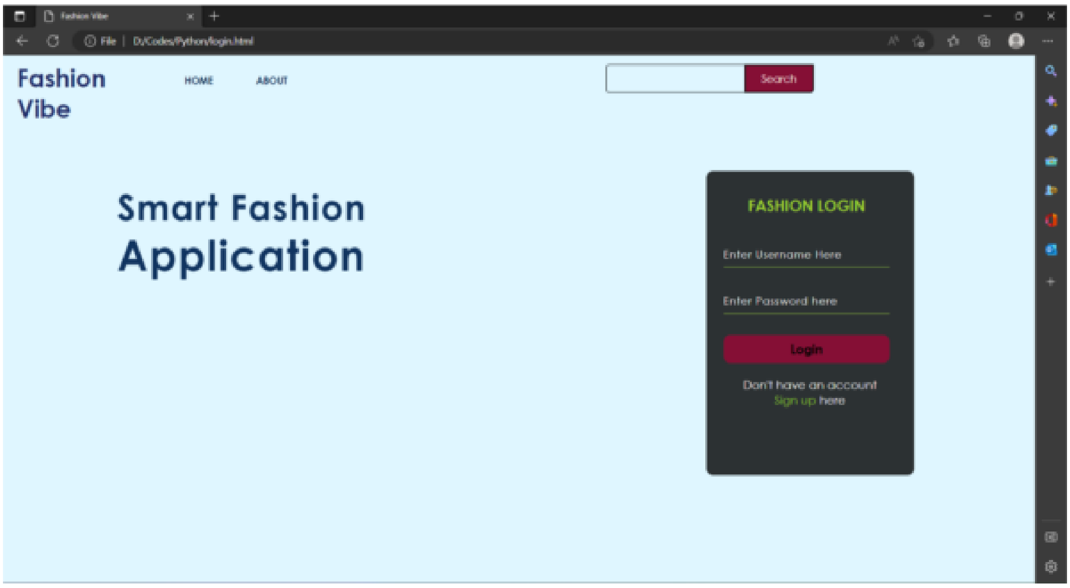
8. TESTING

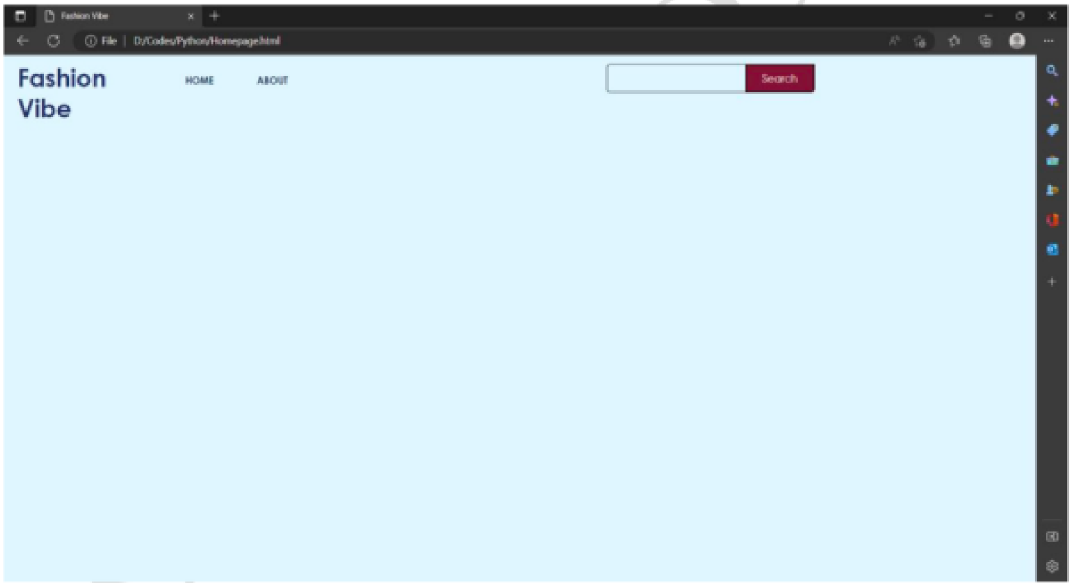
Black Box Testing is the method that does not consider the internal structure, design, and product implementation to be tested. In other words, the tester does not know its internal functioning. The Black Box only evaluates the external behavior of the system. The inputs received by the system and the outputs or responses it produces are tested.

The White Box Test method is the one that looks at the code and structure of the product to be tested and uses that knowledge to perform the tests. This method is used in the Unit Testing phase, although it can also occur in other stages such as Integration Tests. For the execution of this method, the tester or the person who will use this method must have extensive knowledge of the technology used to develop the program.

8.1 Test Cases

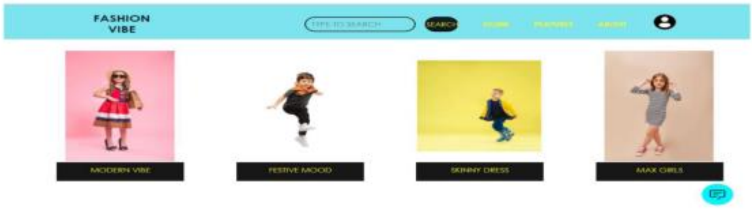


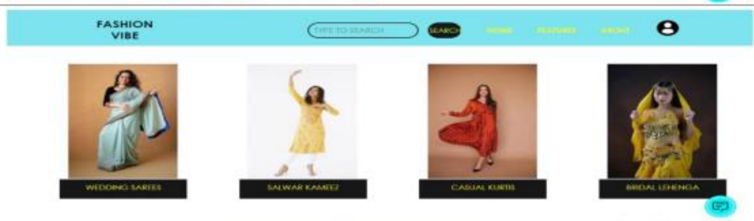

Fashion Vibe
D:/Codes/Python/login.html
Fashion
Vibe
HOME
ABOUT
Search
Smart Fashion
Application
FASHION LOGIN
Enter Username Here
Enter Password here
Login
Don't have an account
Sign up here

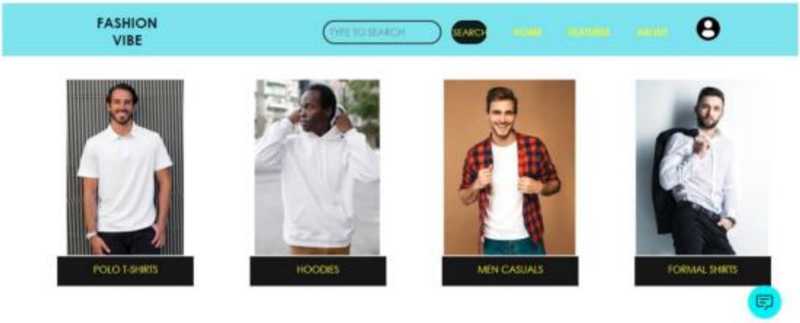

Fashion Vibe
D:/Codes/Python/Homepage.html
Fashion
Vibe
HOME
ABOUT
Search

9. RESULTS

HOME PAGE

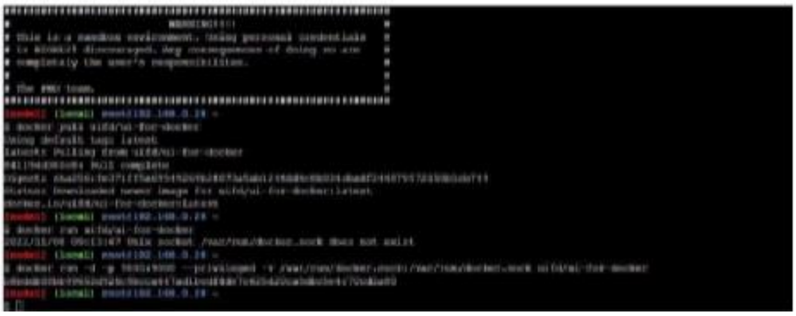





FASHION
VIBE
TYPE TO SEARCH
SEARCH
POLO T-SHIRTS
HOODIES
MEN CASUALS
FORMAL SHIRTS


FASHION
VIBE
TYPE TO SEARCH
SEARCH
PERSONAL ADORNMENTS
ADORNMENT is never anything except a REFLECTION
of the HEART!!!

CHAT BOT



FEEDBACK

9.1 Performance Metrics

The performance of a recommendation algorithm is evaluated by using some specific metrics that indicate the accuracy of the system. The type of metric used depends on the type of filtering technique. Root Mean Square Error (RMSE), Receiver Operating Characteristics (ROC), Area Under Cover (AUC), Precision, Recall and F1 score is generally used to evaluate the performance or accuracy of the recommendation algorithms.

Root-mean square error (RMSE). RMSE is widely used in evaluating and comparing the performance of a recommendation system model compared to other models. A lower RMSE value indicates higher performance by the recommendation model. RMSE, as mentioned by [69], can be as represented as follows:

$$RMSE=\sqrt{1/Np\sum u,i(pui-rui)^2} \quad (1)$$

where, N_p is the total number of predictions, p_{ui} is the predicted rating that a user u will select an item i and r_{ui} is the real rating.

Precision. Precision can be defined as the fraction of correct recommendations or predictions (known as True Positive) to the total number of recommendations provided, which can be as represented as follows:

$$\text{Precision} = \text{True Positive (TP)}/\text{True Positive(TP)} + \text{False Positive (FP)} \quad (2)$$

It is also defined as the ratio of the number of relevant recommended items to the number of recommended items expressed as percentages.

Recall. Recall can be defined as the fraction of correct recommendations or predictions (known as True Positive) to the total number of correct relevant recommendations provided, which can be as represented as follows:

$$\text{Recall} = \text{True Positive (TP)}/\text{True Positive(TP)} + \text{False Negative (FN)} \quad (3)$$

It is also defined as the ratio of the number of relevant recommended items to the total number of relevant items expressed as percentages.

F1 Score. F1 score is an indicator of the accuracy of the model and ranges from 0 to 1, where a value close to 1 represents higher recommendation or prediction accuracy. It represents precision and recall as a single metric and can be as represented as follows:

$$\text{F1 score} = 2 \times \text{Precision} * \text{Recall}/\text{Precision} + \text{Recall} \quad (4)$$

Coverage. Coverage is used to measure the percentage of items which are recommended by the algorithm among all of the items.

Accuracy. Accuracy can be defined as the ratio of the number of total correct recommendations to the total recommendations provided, which can be as represented as follows:

$$\text{Accuracy} = TP + FN/TP + FN + TN + FP \quad (5)$$

Intersection over union (IoU). It represents the accuracy of an object detector used on a specific dataset [70].

$$\text{IoU} = TP/TP + FN + FP \quad (6)$$

ROC. ROC curve is used to conduct a comprehensive assessment of the algorithm's performance [57].

AUC. AUC measures the performance of recommendation and its baselines as well as the quality of the ranking based on pairwise comparisons [5].

Rank aware top-N metrics. The rank aware top-N recommendation metric finds some of the interesting and unknown items that are presumed to be most attractive to a user [71]. Mean reciprocal rank (MRR), mean average precision (MAP) and normalized discounted cumulative gain (NDCG) are three most popular rank aware metrics.

MRR. MRR is calculated as a mean of the reciprocal of the position or rank of first relevant recommendation [72,73]. MRR as mentioned by [72,73] can be expressed as follows:

$$\text{MRR} = 1/Nu \sum u \in Nu \; 1/Lnu[k] \in Ru \quad (7)$$

where u, N_u and R_u indicate specific user, total number of users and the set of items rated by the user, respectively. L indicates list of ranking length (n) for user (u) and k represents the position of the item found in the he lists L.

MAP: MAP is calculated by determining the mean of average precision at the points where relevant products or items are found. MAP as mentioned by [73] can be expressed as follows.

MAP=1Nu|Ru|∑nk=1𝟙(Lnu[k]∈Ru)Pu@k (8)

where Pu represents precision in selecting relevant item for the user.

NDCG: NDCG is calculated by determining the graded relevance and positional information of the recommended items, which can be expressed as follows [73].

NDCGu=∑nk=1G(u,n,k)D(k)/∑nk=1G∗(u,n,k)D(k) (9)

where D (k) is a discounting function, G (u, n, k) is the gain obtained recommending an item found at k-th position from the list L and G^* (u, n, k) is the gain related to k-th item in the ideal ranking of n size for u user.

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

The Smart 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.

DISADVANTAGES:

Smart Fashion recommendation technology has been the most successful recommendation technology so far, but there are two major problems—recommendation quality and scalability. At present, research at home and abroad mainly focuses on recommendation quality, and there is less discussion on scalability. The scalability problem is that as the size of the system increases, the response time of the system increases to a point where users cannot afford it. Existing solutions often result in a significant drop in recommendation quality while reducing recommendation response time. In this paper, the clustering analysis subsystem based on the genetic algorithm is innovatively introduced into the traditional collaborative filtering recommendation system, and its design and implementation are given.

11. CONCLUSION

The present paper presents the development of a system that recognizes fashion similar images. We accomplish this by implementing an already existing CNN model with transfer learning for cloth image recognition using different libraries. For this purpose, we created a plan for collecting data and for developing the steps needed for preprocessing and cleaning up the data. We took into account features like patterns, machine, fabric, style etc. After extensive preprocessing and cleaning of data in a dataset, we constructed the model of stacked CNN to predict the features specific to these attributes and to train the models with the dataset to generate accurate predictions regarding almost all forms of images. A stacked CNN was used and implemented, with the help of this algorithm through which the system can recommend similar images This is the last test to assess if deep learning for style recovery is at a high development and can be utilized in making fashion choices.

12. FUTURE SCOPE

Considering the rapid growth of multimedia data, where visual information will be the critical component. More indepth research in applications of multi-model fusion and multi-task learning in fashion recommender systems are required to model recommender system to be capable of profiling users comprehensively. Besides, while the majority of researches in fashion recommender systems is mainly based on similaritybased retrieval techniques, there is a need for more studies in the development of new functions such as designing clothes, which are highly demanded in future fashion recommender systems. Furthermore, most of the current fashion datasets do not contain outfit compatibility annotations, or they are limited in terms of size and the type of annotations they provide. Consequently, most researchers built their dataset, which is a labor-costing process, and most of them are not accessible publicly for further research. So, the other future direction for subsequent studies may be focusing on developing automatic annotation methods, constructing large-scale rich annotated data sets for particular task definitions in fashion recommender systems. From an ethical perspective in fashion recommender systems also there is a need for performing the comprehensive study since it has not been studied in almost any of the researches, which have been reviewed through this thesis.

13. APPENDIX

FRS can be defined as a means of feature matching between fashion products and users or consumers under specific matching criteria. Different research addressed apparel attributes such as the formulation of colors, clothing shapes, outfit or styles, patterns or prints and fabric structures or textures. Guan et al. studied these features using image recognition, product attribute extraction and feature encoding. Researchers have also considered user features such as facial features, body shapes, personal choice or preference, locations and wearing occasions in predicting users' fashion interests. A well-defined user profile can differentiate a more personalized or customized recommendation system from a conventional system. Various research projects on apparel recommendation systems with personalized styling guideline and intelligent recommendation engines have been conducted based on similarity recommendation and expert advisor recommendation systems.

GitHub & Project Demo Link

https://drive.google.com/file/d/1gD7mejClaomyU2aZ8nDMPUdpwgvNhAdT/view?usp=drivesdk