SMART FASHION RECOMMENDER APPLICATION

PROJECT REPORT

Submitted by

ASHWIN K (19EUCS018)

AVINASH S (19EUCS019)

GOWTHAM S (19EUCS035)

KABIL S(20EUCS506)

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

COIMBATORE

(An Autonomous Institution)

(Approved by AICTE and Affiliated to Anna University,Chennai)

ACCREDITED BY NAAC WITH “A” GRADE

NOVEMBER 2022

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 andto reduce navigating pages to find appropriate products.

2. LITERATURE SURVEY

2.1 Existing problem

Fashion Recommender system with an increase in the standard of living, peoples' attention gradually moved towards fashion that is concerned to be a popular aesthetic expression. Humans are inevitably drawn towards something that is visually more attractive. This tendency of humans has led to the development of the fashion industry over the course of time. However, given too many options of garments on the e-commerce websites, has presented new challenges to the customers in identifying their correct outfit. Thus, in this project, we proposed a personalized Fashion Recommender system that generates recommendations for the user based on an input given. Unlike the conventional systems that rely on the user's previous purchases and history, this project aims at using an image of a product given as input by the user to generate recommendations since many-a-time people see something that they are interested in and tend to look for products that are similar to that. We use neural networks to process the images from Fashion Product Images Dataset and the Nearest neighbour backed recommender to generate the final recommendations.

2.2 References

1.Paper Title: A Comprehensive Review On Online Fashion Recommendation

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 tosupport 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 singleitems 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 Recommeder System

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 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 sharedyour interests in step one.

3. Paper Title: Fashion Recommendation Systems

Methodology: Fast fashion has grown significantly over the past few years, which has had a significant impact on the textile and fashion industries. Aneffective 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

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.

2.3 Problem Statement Definition

Problem Statement 1:

The User Needs a way to Find Trending Fashion Clothes so that Here find the All Collections.

Problem Statement 2:

The User Needs a way to Find Offers and Discounts so that Here User easy to find Daily Offers.

Problem Statement 3:

The User Needs a way to Assistant for finding Clothes so that Here User got the Chat Bot assistant.

Problem Statement 4:

The Sellers Needs a way to struggling to sells products offline so that Here Sellers will Sell Productsvia our application.

3.IDEATION & PROPOSED SYSTEM

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. Theexercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.

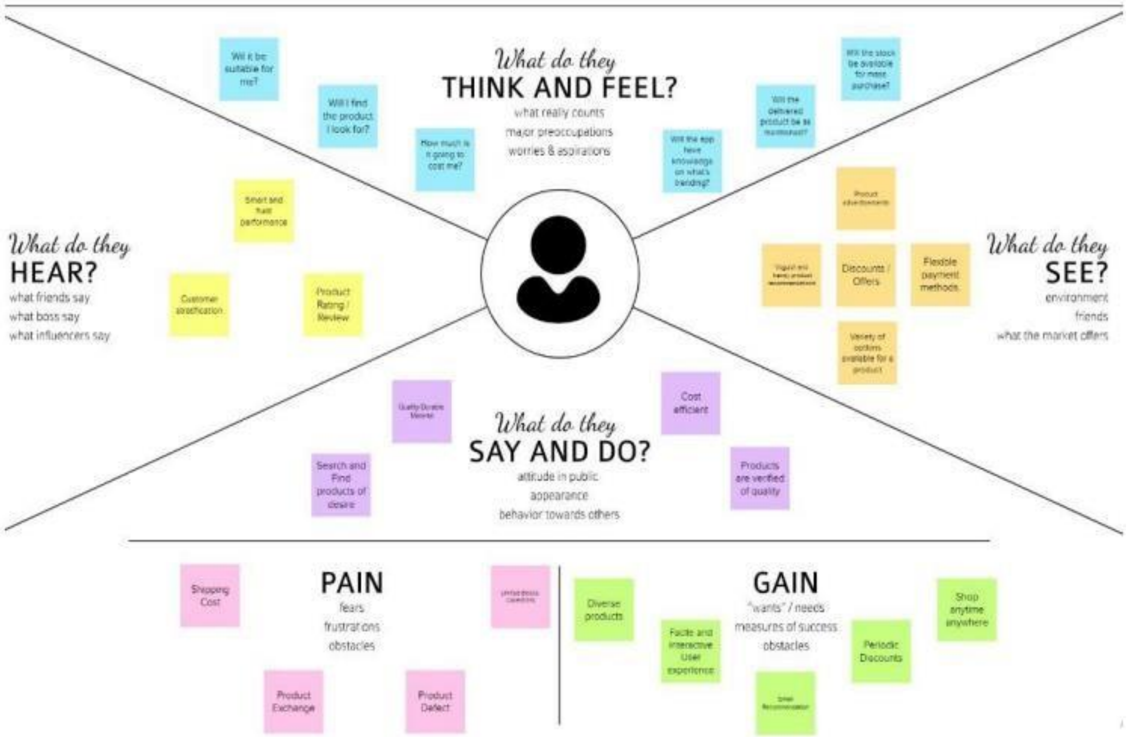
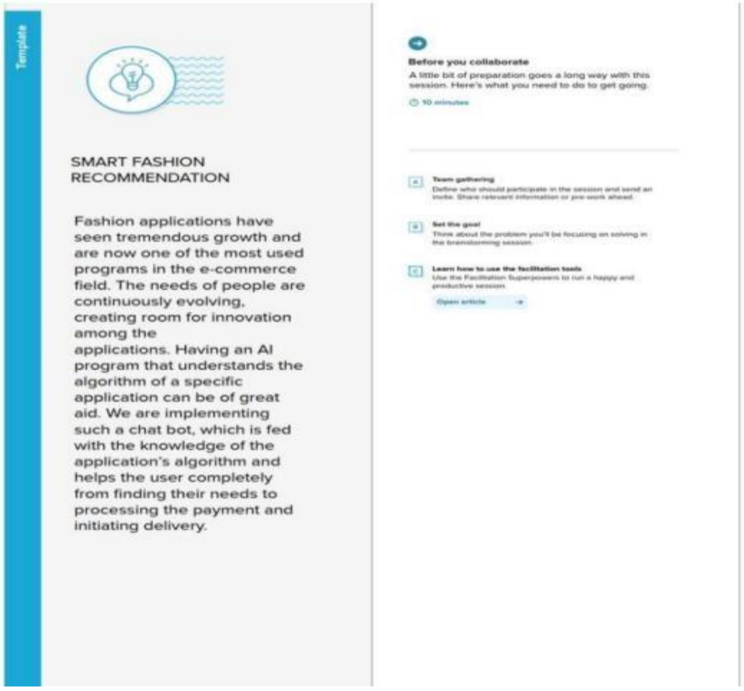


FIG 3.1 EMPATHY MAP CANVAS

3.2 Ideation & Brainstroming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process thatleads 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



SMART FASHION RECOMMENDATION

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. 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.

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

1 **Define your problem statement**

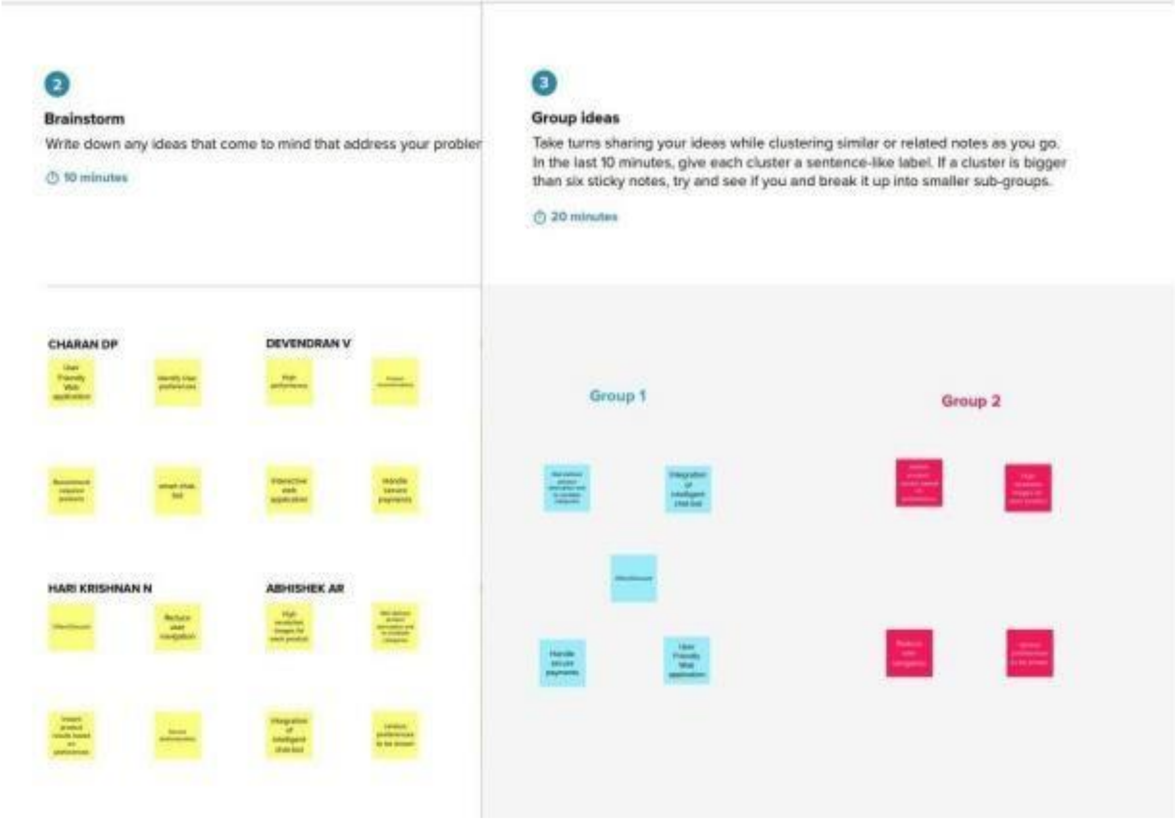
Unavailability of chatbots that are interactive enough to navigate the user to do whatever they want.The amount of toil a user has to go through to look for a product they desire for. Need for a more User-friendly Interface.
The main aim of the project is to develop a smart chat-bot that is able to understand the needs of the user and recommend products of desire.

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2 **Brainstorm**

Write down any ideas that come to mind that address your probler

10 minutes

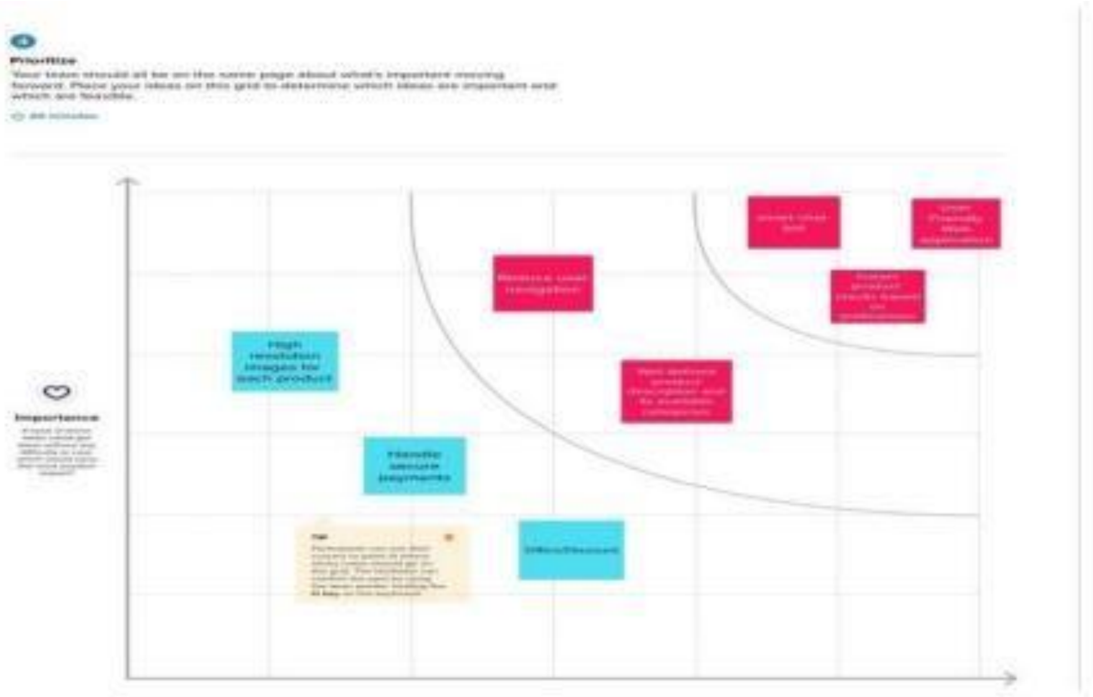
3 **Group ideas**

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

CHARAN DP
User Friendly Web application
Identify User preferences
smart chat bot
DEVENDRAN V
High performance
Interactive web application
Handle secure payments
HARI KRISHNAN N
Reduce user navigation
ABHISHEK AR
High resolution images for each product
Integration of intelligent chat bot
Group 1
Integration of intelligent chat bot
Handle secure payments
User Friendly Web application
Group 2

Step-3: Idea Prioritization

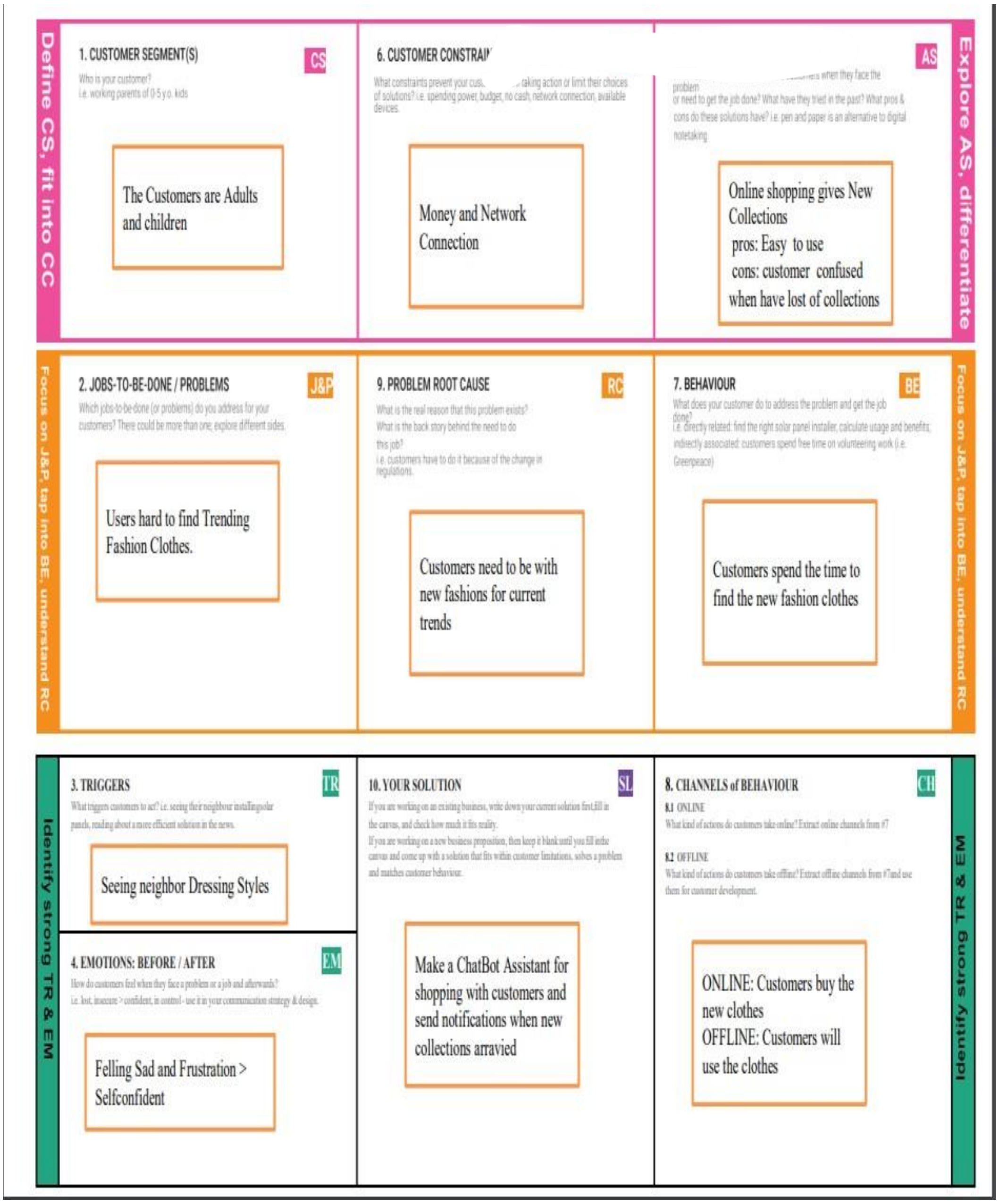

Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
Importance
Reduce user navigation
High resolution images for each product
Handle secure payments

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 tothe user without making them to search through various filters. Reduces time inchoosing 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



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

The Customers are Adults and children

6. CUSTOMER CONSTRAI

What constraints prevent your cus. aking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Money and Network Connection

AS

ers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Online shopping gives New Collections
pros: Easy to use
cons: customer confused when have lost of collections

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Users hard to find Trending Fashion Clothes.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

Customers need to be with new fashions for current trends

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Customers spend the time to find the new fashion clothes

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installingsolar panels, reading about a more efficient solution in the news.

Seeing neighbor Dressing Styles

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Felling Sad and Frustration > Selfconfident

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first,fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill inthe canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Make a ChatBot Assistant for shopping with customers and send notifications when new collections arravied

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7and use them for customer development.

ONLINE: Customers buy the new clothes
OFFLINE: Customers will use the clothes

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

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

4.2 Non-Functional requirements

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

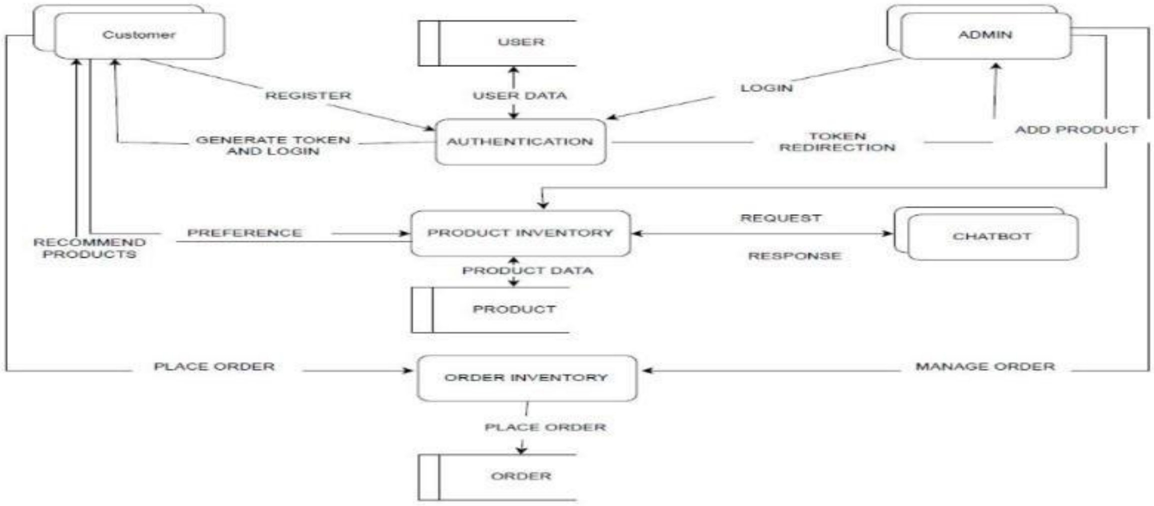
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	Using Android or IOS or windows applications.
NFR-2	**Security**	The user data is stored securely in IBM cloud.
NFR-3	**Reliability**	The Quality of the services are trusted.
NFR-4	**Performance**	Its Provide smooth user experience.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

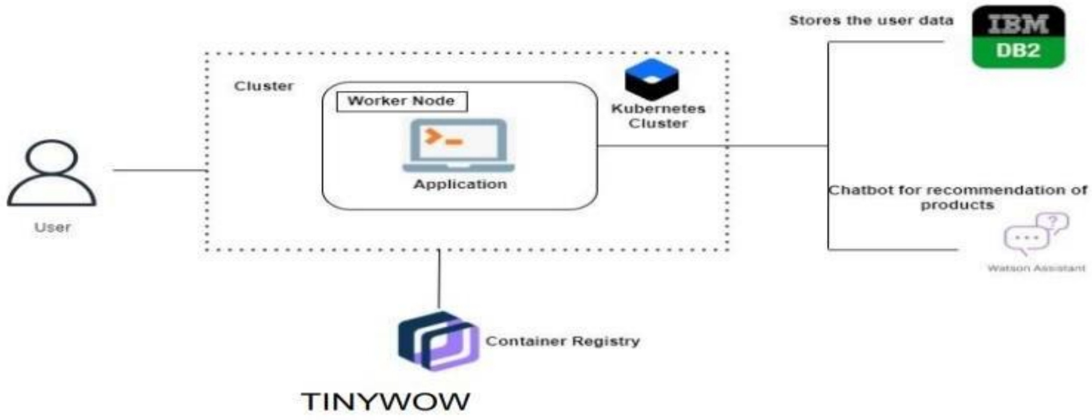
5.1 Data Flow Diagrams

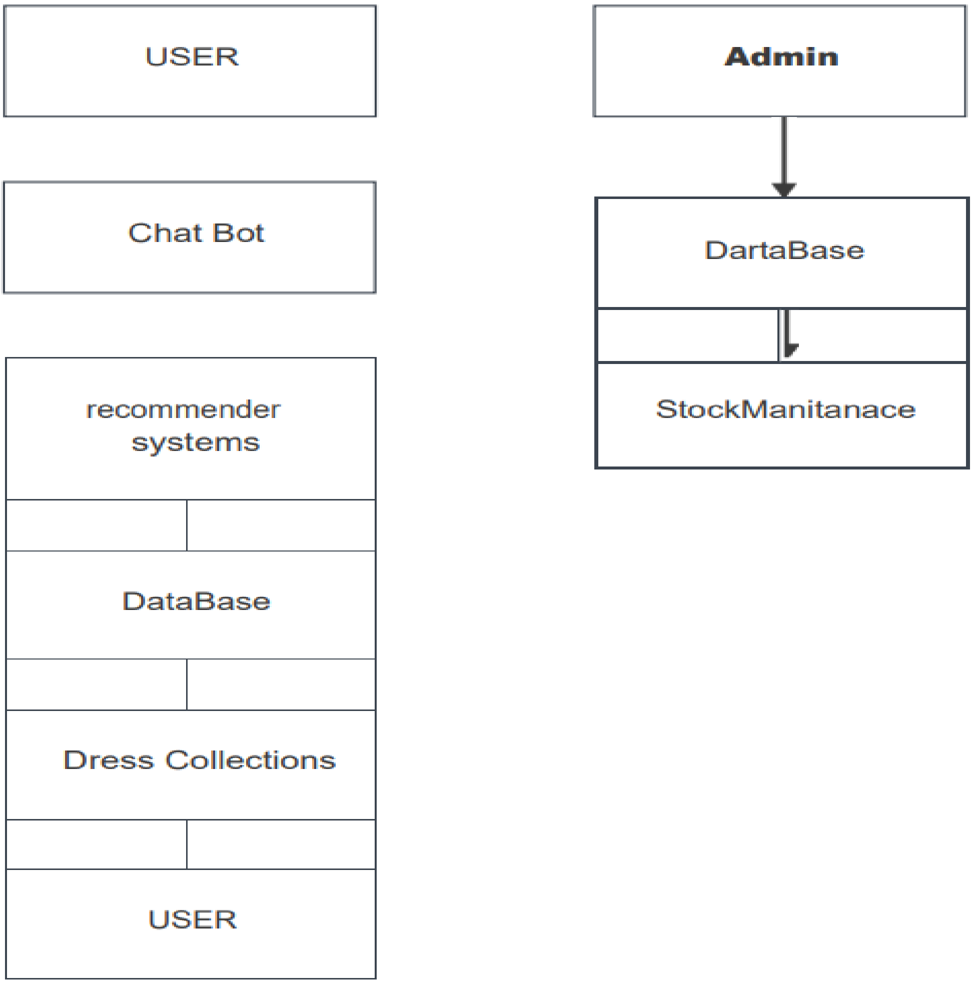
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



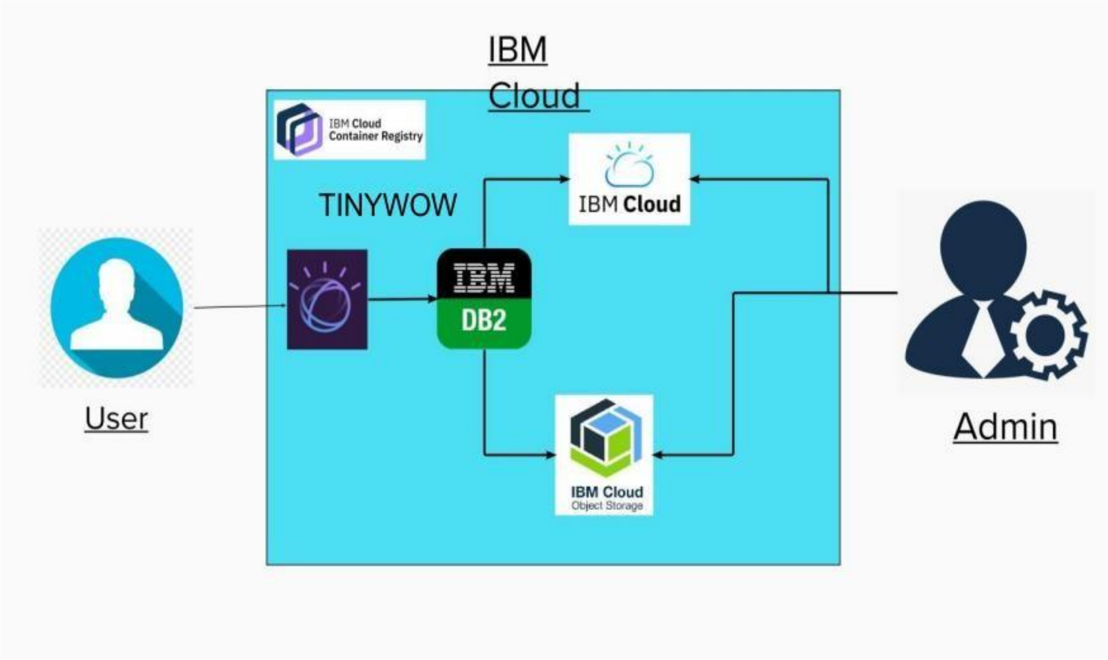
5.2 Solution & Technical Architecture

SOLUTION ARCHITECTURE



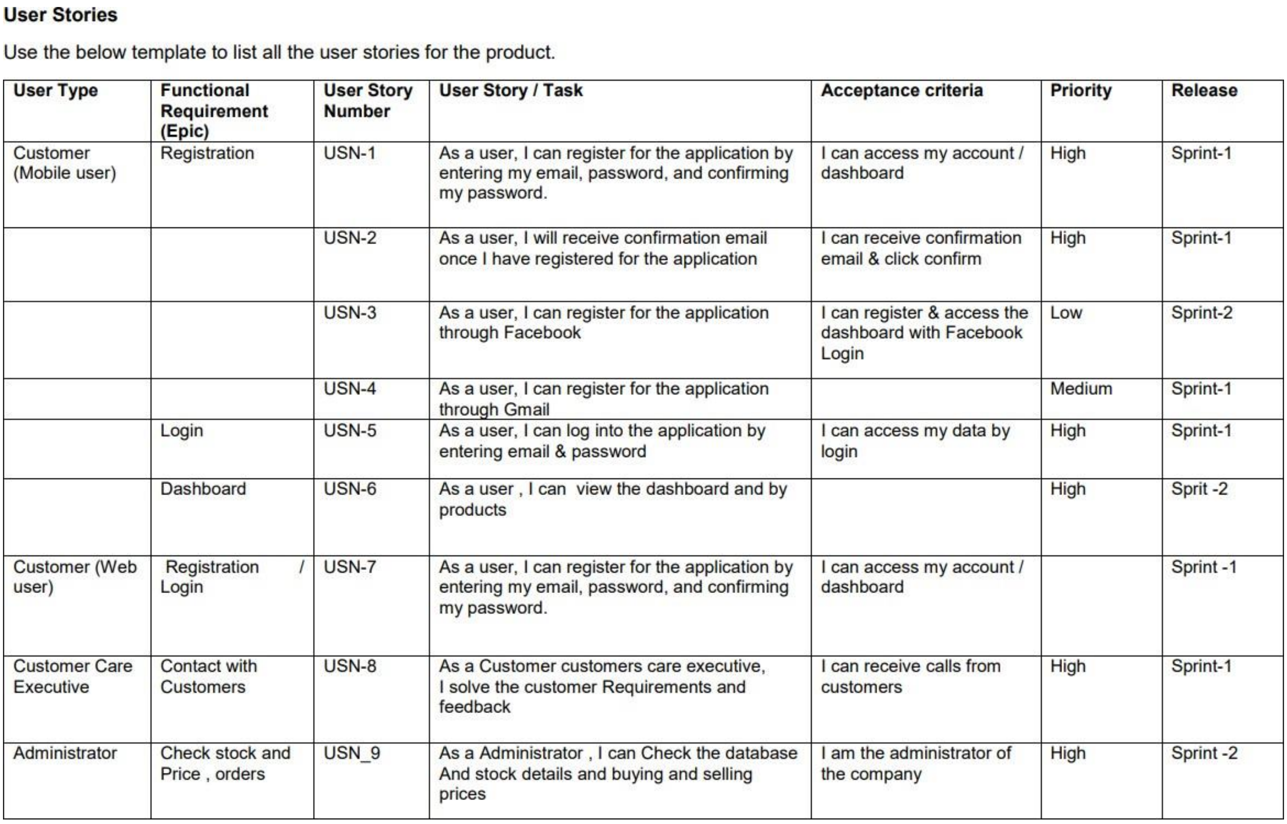


TECHNICAL ARCHITECTURE:



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 ofa user story is to articulate how a



User Stories

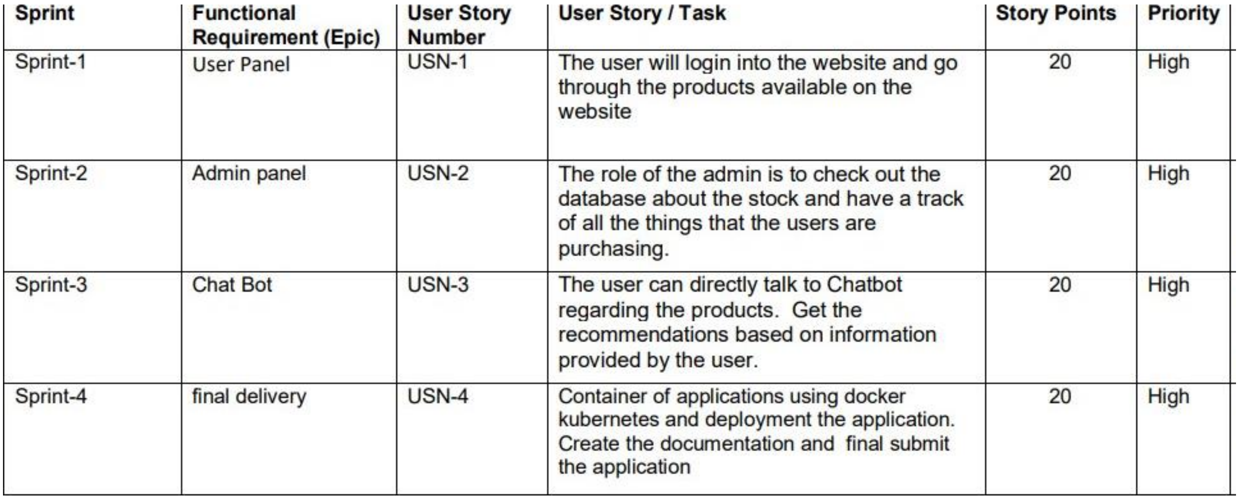
Use the below template to list all the user stories for the product.

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 my data by login	High	Sprint-1
	Dashboard	USN-6	As a user , I can view the dashboard and by products		High	Sprit -2
Customer (Web user)	Registration / Login	USN-7	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard		Sprint -1
Customer Care Executive	Contact with Customers	USN-8	As a Customer customers care executive, I solve the customer Requirements and feedback	I can receive calls from customers	High	Sprint-1
Administrator	Check stock and Price , orders	USN_9	As a Administrator , I can Check the database And stock details and buying and selling prices	I am the administrator of the company	High	Sprint -2

piece of work will deliver a particular value back to the customer.

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

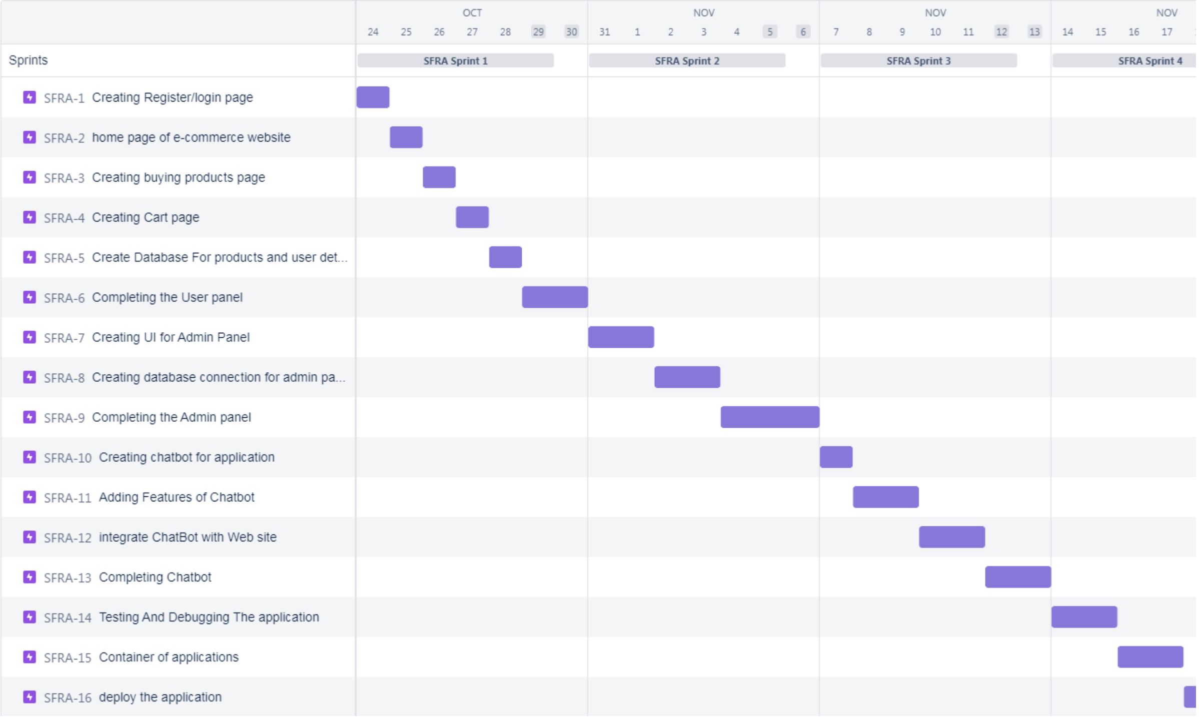


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	User Panel	USN-1	The user will login into the website and go through the products available on the website	20	High
Sprint-2	Admin panel	USN-2	The role of the admin is to check out the database about the stock and have a track of all the things that the users are purchasing.	20	High
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
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

6.2 SPRINT DELIEVERY SCHEDULE

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

6.3 Burndown Chart:



Sprints

SFRA-1 Creating Register/login page

SFRA-2 home page of e-commerce website

SFRA-3 Creating buying products page

SFRA-4 Creating Cart page

SFRA-5 Create Database For products and user det...

SFRA-6 Completing the User panel

SFRA-7 Creating UI for Admin Panel

SFRA-8 Creating database connection for admin pa...

SFRA-9 Completing the Admin panel

SFRA-10 Creating chatbot for application

SFRA-11 Adding Features of Chatbot

SFRA-12 integrate ChatBot with Web site

SFRA-13 Completing Chatbot

SFRA-14 Testing And Debugging The application

SFRA-15 Container of applications

SFRA-16 deploy the application

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

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>View Product</title>
    <script
src="https://cdn.jsdelivr.net/npm/@splidejs/splide@2.4.21/dist/js/splide.min.js"></script>
        <link
rel="stylesheet"   href="https://cdn.jsdelivr.net/npm/@splidejs/splide@2.4.21/dist/css/splide.
min.css"/>
        <meta content="text/html; charset=iso-8859-2" http-equiv="Content-Type">
        <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
        <link rel="stylesheet" href="{{url_for('static',filename='styles/home.css')}}">
<style>
     @import
url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,
500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&displa
y=swap');
h1,h2,h3{
    color: #fff;
    font-family: 'Poppins', sans-serif;
}

</style>

</head>
<body>
  <div class="navbar">
    <a href="/logout">LOG OUT</a>
    <div class="title">
```

```
              <h3  style="right:0;font-family:'Lobster', cursive; color: #fff;margin:
1%;">InfiniteArt</h3>
            </div>
        </div>
        <div class="co">
          <div class="w3-content w3-section" style="max-width:100%;;">
              <img class="mySlides" src="{{url_for('static',filename='figma/slider_1.jpg')}}"
style="width:100%">
              <img class="mySlides" src="{{url_for('static',filename='figma/silder_2.jpg')}}"
style="width:100%">
              <img class="mySlides" src="{{url_for('static',filename='figma/silder_3.jpg')}}"
style="width:100%">
              </div>
        </div>

      <h2>TRENDING SHIRTS</h2>
      <div class="splide" style="padding: 0;margin:0">
          <div class="splide__track">
              <ul class="splide__list">
                {% for item in dictionary %}
                <li class="splide__slide">
                  <div class="container">
                      <div class="card">

                          <div class="image">
                              <img src="{{item.IMAGE}}">

                          </div>

                          <div class="descrpton">

                              <h2>{{item.NAME}}</h2>
                              <div class="size">
                                <span>7</span>
                                <span>8</span>
                                <span>9</span>
                                <span>10</span>
                              </div>
                              <h3>&#8360;{{item.RATE}}</h3>
                              <div class="color">
                                <h3>Color: </h3>
                                <span></span>
                                <span></span>
                                <span></span>
                              </div>
                              <a href="#">Buy now</a>
                          </div>


                          </div>
                  </div>
                  </li>
                  {% endfor %}
                  <!-- pant -->
              </ul>
          </div>
      </div>
      <H2>TRENDING PANTS</H2>
      <div id="pant"  class="splide"  style="margin:0;background: #131313;" >
          <div class="splide__track">
              <ul class="splide__list">
                {% for item in pants %}
                <li class="splide__slide">
                  <div class="container">
                      <div class="card">
```

```
          <div class="image">
            <img src="{{item.IMAGE}}">

          </div>

          <div class="descrpton">

            <h2>{{item.NAME}}</h2>

            <div class="size">
              <span>7</span>
              <span>8</span>
              <span>9</span>
              <span>10</span>
            </div>
            <h3>&#8360;{{item.RATE}}</h3>
            <div class="color">
              <h3>Color: </h3>
              <span></span>
              <span></span>
              <span></span>
            </div>
            <a href="#">Buy now</a>
          </div>
          </div>
      </div>
      </li>
      {% endfor %}
      <!-- pant -->
      </ul>
      </div>
</div>
```

```
<h2>TRENDING WATCHES</h2>
<div id="watch"  class="splide" style="padding: 0;margin:0;background: #131313;" >
    <div class="splide__track">
      <ul class="splide__list">
       {% for item in watchs %}
       <li class="splide__slide">
        <div class="container">
          <div class="card">

            <div class="image">
              <img src="{{item.IMAGE}}">

            </div>

            <div class="descrpton">

              <h2>{{item.NAME}}</h2>

              <div class="size">
                <span>7</span>
                <span>8</span>
                <span>9</span>
                <span>10</span>
              </div>
              <h3>&#8360;{{item.RATE}}</h3>
              <div class="color">
                <h3>Color: </h3>
                <span></span>
                <span></span>
                <span></span>
              </div>
              <a href="#">Buy now</a>
            </div>
            </div>
        </div>
        </li>
        {% endfor %}
        <!-- pant -->
        </ul>
        </div>
</div>
<h2>TRENDING RINGS</h2>
 <div id="ring"  class="splide" style="padding: 0;margin:0;background: #131313;"  >
         <div class="splide__track">
           <ul class="splide__list">
            {% for item in rings %}
            <li class="splide__slide">
             <div class="container">
               <div class="card">

                 <div class="image">
                   <img src="{{item.IMAGE}}">

                 </div>

                 <div class="descrpton">

                   <h2>{{item.NAME}}</h2>

                   <div class="size">
                     <span>7</span>
                     <span>8</span>
                     <span>9</span>
                     <span>10</span>
                   </div>
                   <h3>{{item.RATE}}</h3>
```

```
;                       <div class="color">
                            <h3>Color: </h3>
                            <span></span>
                            <span></span>
                            <span></span>
                        </div>
                        <a href="#">Buy now</a>
                    </div>
                    </div>
                </div>
                </li>
                {% endfor %}
                <!-- pant -->
                </ul>
                </div>
</div>
<script>
var splide = new Splide( '.splide', {
type    : 'loop',
perPage : 4,
autoplay: true,
} );

splide.mount();

document.addEventListener('DOMContentLoaded', function () {
new Splide('#pant', {
perPage: 4,
perMove: 1,
gap: "30px",
pagination: false,
}).mount();
});
```

```
document.addEventListener('DOMContentLoaded', function () {
new Splide('#watch', {
perPage: 4,
perMove: 1,
gap: "30px",
pagination: false,
}).mount();
});

document.addEventListener('DOMContentLoaded', function () {
new Splide('#ring', {
perPage: 4,
perMove: 1,
gap: "30px",
pagination: false,
}).mount();
});

</script>

<script>
var splid = new Splide( '.splid', {
type    : 'loop',
perPage : 4,
autoplay: true,
} );
splid.mount();
</script>
<script>
var myIndex = 0;
carousel();
```

```
    function carousel() {
    var i;
    var x = document.getElementsByClassName("mySlides");
    for (i = 0; i < x.length; i++) {
    x[i].style.display = "none";
    }
    myIndex++;
    if (myIndex > x.length) {myIndex = 1}
    x[myIndex-1].style.display = "block";
    setTimeout(carousel, 2000); // Change image every 2 seconds
    }
    </script>

    </body>
    </html>
```

App.py

```
import secrets
from turtle import title
from unicodedata import category
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import bcrypt
import base64
from PIL import Image
import io

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-
a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SECURI
TY=SSL;
SSLServerCertificateDigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=nhl80748;PWD
=3yD0G9e6VuQHsOBX;", "", "")
#url_for('static', filename='style.css')
```

```
app = Flask(__name__)
app.secret_key = b'_5#y2L"F4Q8z\n\xec]/'

@app.route("/",methods=['GET'])
def home():
    if 'email' not in session:
      return redirect(url_for('index'))
    return render_template('home.html',name='Home')
@app.route("/index")
def index():
  return render_template('index.html')

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

    if not username or not email or not phoneno or not password:
      return render_template('register.html',error='Please fill all fields')
    hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())
    query = "SELECT * FROM user_detail WHERE email=? OR phoneno=?"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.bind_param(stmt,2,phoneno)
    ibm_db.execute(stmt)
    isUser = ibm_db.fetch_assoc(stmt)
    if not isUser:
      insert_sql = "INSERT INTO user_detail(username, email, phoneno, password) VALUES (?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, username)
      ibm_db.bind_param(prep_stmt, 2, email)
      ibm_db.bind_param(prep_stmt, 3, phoneno)
      ibm_db.bind_param(prep_stmt, 4, hash)
      ibm_db.execute(prep_stmt)
      return render_template('register.html',success="You can login")
    else:
      return render_template('register.html',error='Invalid Credentials')

  return render_template('register.html',name='Home')

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

      if not email or not password:
        return render_template('login.html',error='Please fill all fields')
      query = "SELECT * FROM user_detail WHERE email=?"
      stmt = ibm_db.prepare(conn, query)
      ibm_db.bind_param(stmt,1,email)
      ibm_db.execute(stmt)
      isUser = ibm_db.fetch_assoc(stmt)
      print(isUser,password)

      if not isUser:
        return render_template('login.html',error='Invalid Credentials')

      isPasswordMatch = bcrypt.checkpw(password.encode('utf-
8'),isUser['PASSWORD'].encode('utf-8'))
```

```
      if not isPasswordMatch:
        return render_template('login.html',error='Invalid Credentials')

      session['email'] = isUser['EMAIL']
      return redirect(url_for('home'))

    return render_template('login.html',name='Home')

@app.route("/admin",methods=['GET','POST'])
def adregister():
  if request.method == 'POST':
    username = request.form['username']
    email = request.form['email']
    phoneno = request.form['phoneno']
    password = request.form['password']

    if not username or not email or not phoneno or not password:
      return render_template('adminregister.html',error='Please fill all fields')
    hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())
    query = "SELECT * FROM admin_detail WHERE email=? OR phoneno=?"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.bind_param(stmt,2,phoneno)
    ibm_db.execute(stmt)
    isUser = ibm_db.fetch_assoc(stmt)
    if not isUser:
      insert_sql = "INSERT INTO admin_detail(username, email, phoneno, password)
VALUES (?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, username)
      ibm_db.bind_param(prep_stmt, 2, email)
      ibm_db.bind_param(prep_stmt, 3, phoneno)
      ibm_db.bind_param(prep_stmt, 4, hash)
      ibm_db.execute(prep_stmt)
      return render_template('adminregister.html',success="You can login")
    else:
      return render_template('adminregister.html',error='Invalid Credentials')

  return render_template('adminregister.html',name='Home')

@app.route("/adminlogin",methods=['GET','POST'])
def adlogin():
    if request.method == 'POST':
      email = request.form['email']
      password = request.form['password']

      if not email or not password:
        return render_template('adminlogin.html',error='Please fill all fields')
      query = "SELECT * FROM admin_detail WHERE email=?"
      stmt = ibm_db.prepare(conn, query)
      ibm_db.bind_param(stmt,1,email)
      ibm_db.execute(stmt)
      isUser = ibm_db.fetch_assoc(stmt)
      print(isUser,password)

      if not isUser:
        return render_template('adminlogin.html',error='Invalid Credentials')

      isPasswordMatch = bcrypt.checkpw(password.encode('utf-
8'),isUser['PASSWORD'].encode('utf-8'))

      if not isPasswordMatch:
        return render_template('adminlogin.html',error='Invalid Credentials')

      session['email'] = isUser['EMAIL']
      return redirect(url_for('home'))
    return render_template('adminlogin.html',name='Home')

@app.route("/addproduct",methods=['GET','POST'])
def addproduct():
  if request.method == 'POST':
    types=request.form['cc']
    name = request.form['name']
    image = request.form['image']
    rate = request.form['rate']
    categorie = request.form['categorie']
    if types =='shirt':
      insert_sql = "INSERT INTO SHIRT(name, image, categorie,rate) VALUES (?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, name)
      ibm_db.bind_param(prep_stmt, 2, image)
      ibm_db.bind_param(prep_stmt, 3, categorie)
      ibm_db.bind_param(prep_stmt, 4, rate)
      ibm_db.execute(prep_stmt)
    if types =='pant':
      insert_sql = "INSERT INTO PANT(name, image, rate) VALUES (?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, name)
      ibm_db.bind_param(prep_stmt, 2, image)
      ibm_db.bind_param(prep_stmt, 3, rate)
      ibm_db.execute(prep_stmt)
    if types =='watch':
      insert_sql = "INSERT INTO WATCH(name, image, rate) VALUES (?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, name)
      ibm_db.bind_param(prep_stmt, 2, image)
      ibm_db.bind_param(prep_stmt, 3, rate)
      ibm_db.execute(prep_stmt)
    if types =='ring':
      insert_sql = "INSERT INTO RINGS(name, image, categorie,rate) VALUES (?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, name)
      ibm_db.bind_param(prep_stmt, 2, image)
      ibm_db.bind_param(prep_stmt, 3, categorie)
      ibm_db.bind_param(prep_stmt, 4, rate)
      ibm_db.execute(prep_stmt)

  return render_template('addproduct.html',success="You can login")

@app.route("/data")
def display():
  shirt_list=[]
  pant_list=[]
  watch_list=[]
  ring_list=[]

  #selecting_shirt
  sql = "SELECT * FROM SHIRT"
  stmt = ibm_db.exec_immediate(conn, sql)
  shirt = ibm_db.fetch_both(stmt)
  while shirt != False :
      shirt_list.append(shirt)
      shirt = ibm_db.fetch_both(stmt)
  print(shirt_list)

 #selecting_pant

  sql1="SELECT * FROM PANT"
  stmt1 = ibm_db.exec_immediate(conn, sql1)
  pant=ibm_db.fetch_both(stmt1)
  while pant != False :
      pant_list.append(pant)
      pant = ibm_db.fetch_both(stmt1)
  print(pant_list)

#selecting_watch
  sql2="SELECT * FROM WATCH"
  stmt2 = ibm_db.exec_immediate(conn, sql2)
  watch=ibm_db.fetch_both(stmt2)
  while watch != False :
      watch_list.append(watch)
      watch = ibm_db.fetch_both(stmt2)
  print(watch_list)

  #selecting_rings
  sql3="SELECT * FROM RINGS"
  stmt3 = ibm_db.exec_immediate(conn, sql3)
  ring=ibm_db.fetch_both(stmt3)
  while ring != False :
      ring_list.append(ring)
      ring = ibm_db.fetch_both(stmt3)
  print(ring_list)
  #returning to HTML
  return render_template('home.html',dictionary=
shirt_list,pants=pant_list,watchs=watch_list,rings=ring_list)

@app.route("/home")
def dis():
  watch_list=[]
  sql2="SELECT * FROM WATCH"
  stmt2 = ibm_db.exec_immediate(conn, sql2)
  watch=ibm_db.fetch_both(stmt2)
  while watch != False :
      watch_list.append(watch)
      watch = ibm_db.fetch_both(stmt2)
  print(watch_list)
  return render_template('home.html',watchs=watch_list)



@app.route('/logout')
def logout():
    session.pop('email', None)
    return redirect(url_for('login'))
if __name__ == "__main__":
    app.run(debug=True)
```

8.TESTING

8.1 Test Cases

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

Section	Total Cases	Not Tested	Fail	Pass
Login	5	0	0	5
Register	5	0	0	5
Home Page	2	0	0	2
Order page	2	0	0	2
Order products	5	0	0	5
Final Report Output	2	0	0	2

8.2 USER ACCEPTANCE TESTING

Purpose of Document

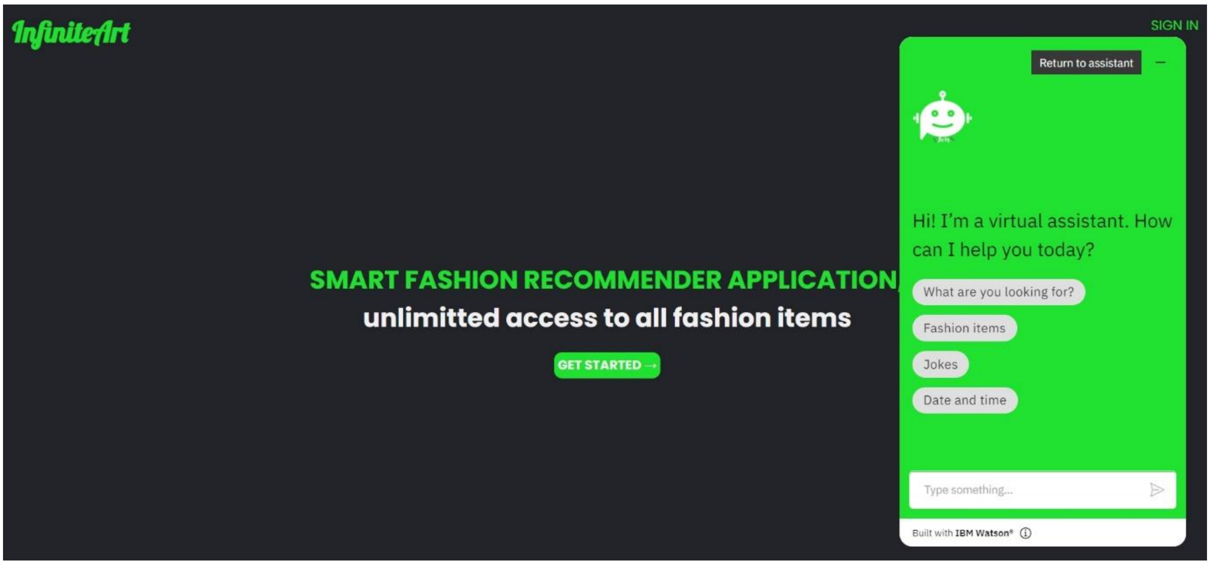
The purpose of this document is to briefly explain the test coverage and open issues of the SmartFashion Recommender Application project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	5	2	3	15
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	10	2	4	15	31
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	18	15	13	21	67

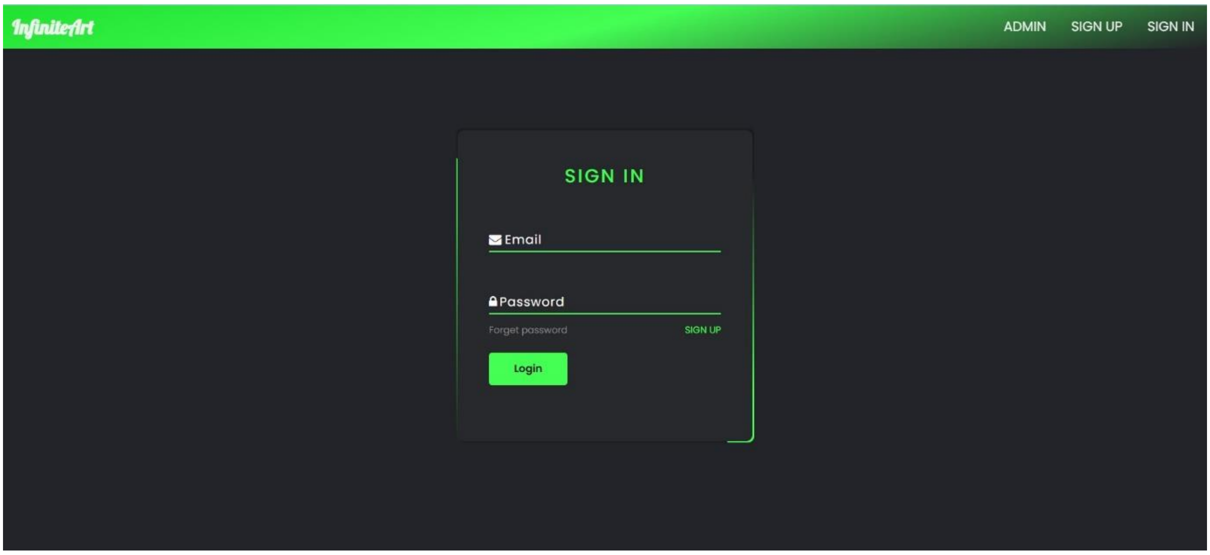
9. RESULTS

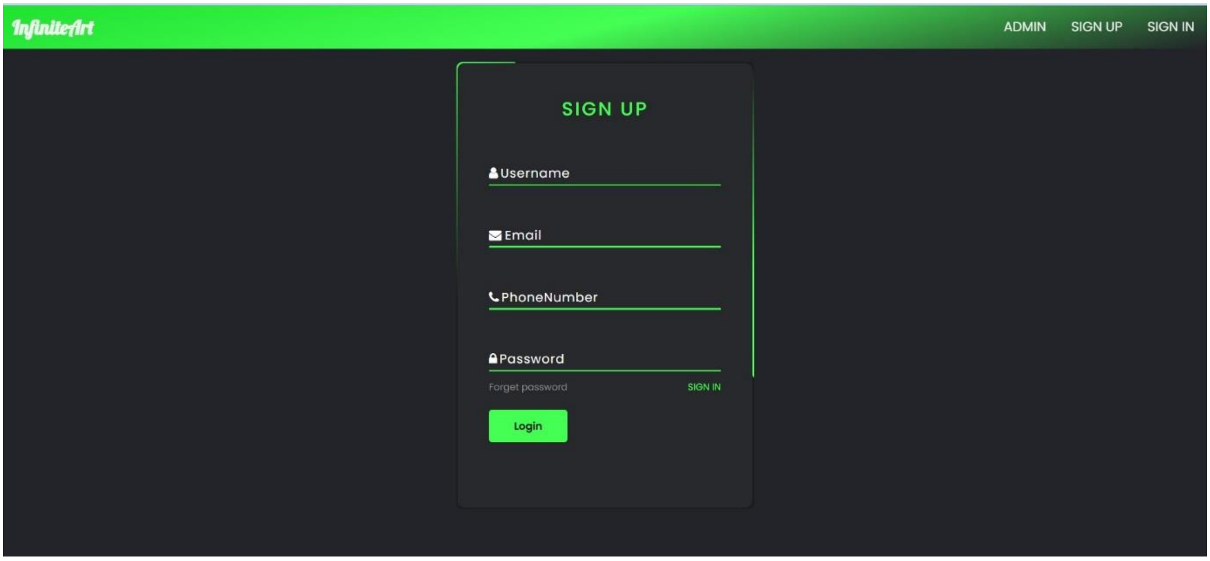


SMART FASHION RECOMMENDER APPLICATION

unlimitted access to all fashion items

Hi! I'm a virtual assistant. How can I help you today?





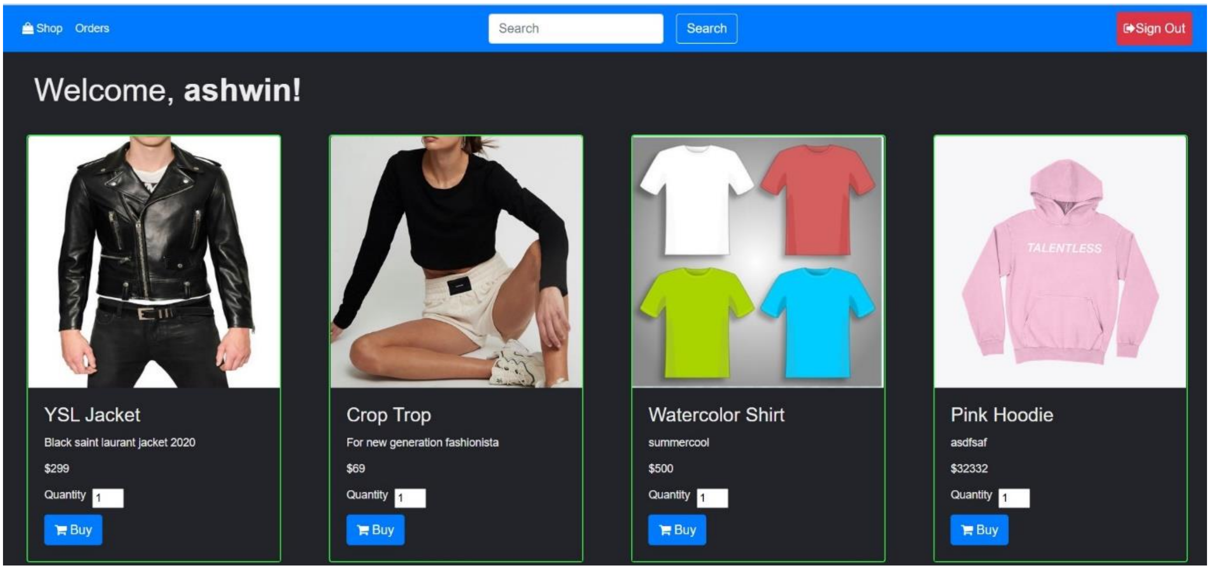
SIGN UP

Username

Email

PhoneNumber

Password



Welcome, **ashwin!**

YSL Jacket

Black saint laurant jacket 2020

$299

Quantity 1

Crop Trop

For new generation fashionista

$69

Quantity 1

Watercolor Shirt

summercool

$500

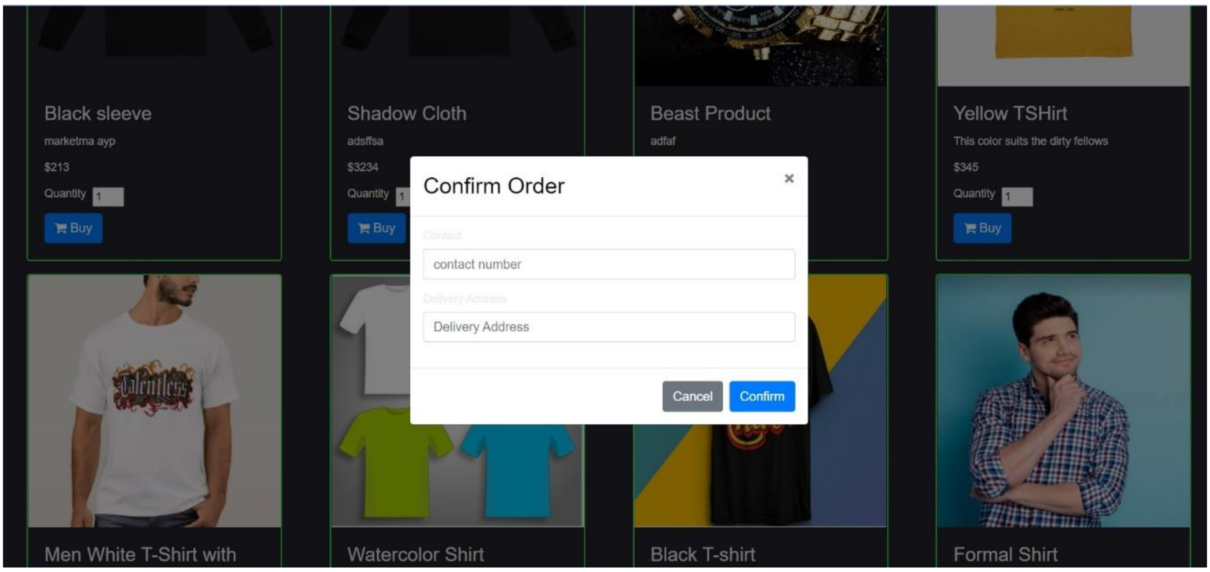
Quantity 1

Pink Hoodie

asdfsaf

$32332

Quantity 1


Black sleeve

marketma ayp

$213

Quantity 1

Shadow Cloth

adsffsa

$3234

Quantity 1

Beast Product

adfaf

Yellow TSHirt

This color suits the dirty fellows

$345

Quantity 1

Confirm Order

Contact

contact number

Delivery Address

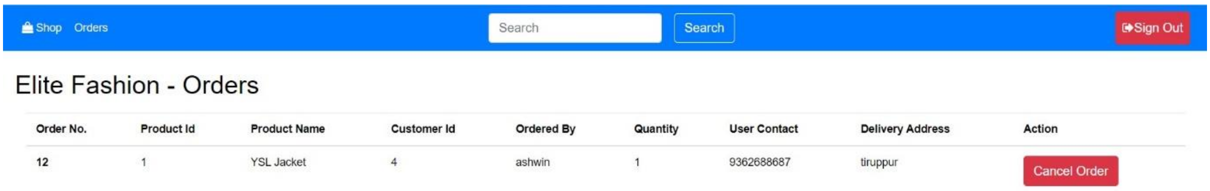
Delivery Address

Men White T-Shirt with

Watercolor Shirt

Black T-shirt

Formal Shirt



Elite Fashion - Orders

Order No.	Product Id	Product Name	Customer Id	Ordered By	Quantity	User Contact	Delivery Address	Action
12	1	YSL Jacket	4	ashwin	1	9362688687	tiruppur	Cancel Order

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 typeof 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.

1. Hours worked : 50 hours

2. Sticking to Timelines : 100%

3. Consistency of the product : 75%

4. Efficiency of the product : 80%

5. Quality of the product : 85%

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 recoveryis at a high development and can be utilized in making fashion choices.

12. FUTURE SCOPE

There has been significant progress recently in fashion recommendation system research, which will benefit both consumers and retailers soon. The use of product and user images, textual content, demographic history, and cultural information is crucial in developing recommendation frameworks. Product attributes and clothing style matching are common features of collaborativeand content-based filtering techniques. Researchers can develop more sophisticated hyper personalized filtering techniques considering the correlation between consumers' clothing styles and personalities. The methods based on employing a scoring system for quantifying each productattribute will be helpful in increasing the precision of the model. The use of virtual sales advisersin an online shopping portal would provide consumers with a real time offline shopping experience. Retailers can collect the data on users' purchase history and product reviews from therecommendation system and subsequently use them in style prediction for the upcoming seasons.The integration of different domain information strengthens the deep learning paradigm by enabling the detection of design component variation, which improves the performance of the recommendation system in the long run. Deep learning approaches should be more frequently usedto quickly explore fashion items from different online databases to provide prompt recommendations to users or consumers.

13. APPENDIX

Home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>View Product</title>
    <script src="https://cdn.jsdelivr.net/npm/@splidejs/splide@2.4.21/dist/js/splide.min.js"></script>
        <link rel="stylesheet"   href="https://cdn.jsdelivr.net/npm/@splidejs/splide@2.4.21/dist/css/splide.min.css"/>
        <meta content="text/html; charset=iso-8859-2" http-equiv="Content-Type">
        <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
        <link rel="stylesheet" href="{{url_for('static',filename='styles/home.css')}}">
<style>
     @import url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');
h1,h2,h3{
    color: #fff;
    font-family: 'Poppins', sans-serif;
}


</style>


</head>
<body>
  <div class="navbar">
    <a href="/logout">LOG OUT</a>
  <div class="title">
    <h3  style="right:0;font-family:'Lobster', cursive; color: #fff;margin: 1%;">InfiniteArt</h3>
   </div>
 </div>
 <div class="co">
  <div class="w3-content w3-section" style="max-width:100%;;">
    <img class="mySlides" src="{{url_for('static',filename='figma/slider_1.jpg')}}" style="width:100%">
    <img class="mySlides" src="{{url_for('static',filename='figma/silder_2.jpg')}}" style="width:100%">
    <img class="mySlides" src="{{url_for('static',filename='figma/silder_3.jpg')}}" style="width:100%">
    </div>
 </div>

<h2>TRENDING SHIRTS</h2>
<div class="splide" style="padding: 0;margin:0">
  <div class="splide__track">
    <ul class="splide__list">
     {% for item in dictionary %}
     <li class="splide__slide">
      <div class="container">
        <div class="card">

          <div class="image">
            <img src="{{item.IMAGE}}">

          </div>

          <div class="descrpton">

            <h2>{{item.NAME}}</h2>
```

```
                    <div class="size">
                        <span>7</span>
                        <span>8</span>
                        <span>9</span>
                        <span>10</span>
                    </div>
                    <h3>&#8360;{{item.RATE}}</h3>
                    <div class="color">
                        <h3>Color: </h3>
                        <span></span>
                        <span></span>
                        <span></span>
                    </div>
                    <a href="#">Buy now</a>
                </div>



                </div>
        </div>
        </li>
        {% endfor %}
        <!-- pant -->
    </ul>
  </div>
</div>
<H2>TRENDING PANTS</H2>
<div id="pant"  class="splide"  style="margin:0;background: #131313;" >
  <div class="splide__track">
    <ul class="splide__list">
      {% for item in pants %}
      <li class="splide__slide">
```

```
<div class="container">
    <div class="card">

        <div class="image">
            <img src="{{item.IMAGE}}">

        </div>

        <div class="descrpton">

            <h2>{{item.NAME}}</h2>

            <div class="size">
                <span>7</span>
                <span>8</span>
                <span>9</span>
                <span>10</span>
            </div>
            <h3>&#8360;{{item.RATE}}</h3>
            <div class="color">
                <h3>Color: </h3>
                <span></span>
                <span></span>
                <span></span>
            </div>
            <a href="#">Buy now</a>
        </div>
        </div>
</div>
</li>
{% endfor %}
<!-- pant -->
```

```
      </ul>
      </div>
</div>
<h2>TRENDING WATCHES</h2>
<div id="watch"  class="splide" style="padding: 0;margin:0;background: #131313;" >
    <div class="splide__track">
      <ul class="splide__list">
       {% for item in watchs %}
       <li class="splide__slide">
        <div class="container">
          <div class="card">

            <div class="image">
              <img src="{{item.IMAGE}}">

            </div>

            <div class="descrpton">

              <h2>{{item.NAME}}</h2>

              <div class="size">
                <span>7</span>
                <span>8</span>
                <span>9</span>
                <span>10</span>
              </div>
              <h3>&#8360;{{item.RATE}}</h3>
              <div class="color">
                <h3>Color: </h3>
                <span></span>
                <span></span>
                <span></span>
              </div>
              <a href="#">Buy now</a>
            </div>
            </div>
        </div>
        </li>
        {% endfor %}
        <!-- pant -->
        </ul>
        </div>
</div>
<h2>TRENDING RINGS</h2>
 <div id="ring"  class="splide" style="padding: 0;margin:0;background: #131313;"  >
         <div class="splide__track">
           <ul class="splide__list">
            {% for item in rings %}
            <li class="splide__slide">
             <div class="container">
               <div class="card">

                 <div class="image">
                   <img src="{{item.IMAGE}}">

                 </div>

                 <div class="descrpton">

                   <h2>{{item.NAME}}</h2>

                   <div class="size">
                     <span>7</span>
```

```
                                        <span>8</span>
                                        <span>9</span>
                                        <span>10</span>
                                    </div>
                                    <h3>{{item.RATE}}</h3>
                                    <div class="color">
                                        <h3>Color: </h3>
                                        <span></span>
                                        <span></span>
                                        <span></span>
                                    </div>
                                    <a href="#">Buy now</a>
                                </div>
                                </div>
                        </div>
                        </li>
                        {% endfor %}
                        <!-- pant -->
                        </ul>
                        </div>
</div>
<script>
var splide = new Splide( '.splide', {
type    : 'loop',
perPage : 4,
autoplay: true,
} );

splide.mount();

document.addEventListener('DOMContentLoaded', function () {
new Splide('#pant', {
```

```
perPage: 4,
perMove: 1,
gap: "30px",
pagination: false,
}).mount();
});

document.addEventListener('DOMContentLoaded', function () {
new Splide('#watch', {
perPage: 4,
perMove: 1,
gap: "30px",
pagination: false,
}).mount();
});

document.addEventListener('DOMContentLoaded', function () {
new Splide('#ring', {
perPage: 4,
perMove: 1,
gap: "30px",
pagination: false,
}).mount();
});

</script>

<script>
var splid = new Splide( '.splid', {
type    : 'loop',
perPage : 4,
autoplay: true,
} );
splid.mount();
</script>
<script>
var myIndex = 0;
carousel();

function carousel() {
var i;
var x = document.getElementsByClassName("mySlides");
for (i = 0; i < x.length; i++) {
x[i].style.display = "none";
}
myIndex++;
if (myIndex > x.length) {myIndex = 1}
x[myIndex-1].style.display = "block";
setTimeout(carousel, 2000); // Change image every 2 seconds
}
</script>

</body>
</html>
```

Login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Log in</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap');
        @import url('https://fonts.googleapis.com/css2?family=Lobster&display=swap');
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}
body{
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background-color: #23242a;

}
.box{
    position: relative;
    width: 380px;
    height: 400px;
    background-color: #1c1c1c;
    border-radius: 8px;
    overflow: hidden;
}
.box::before{
    content: '';
    position: absolute;
    top: -50%;
    left: -50%;
    width: 380px;
    height: 420px;
    background: linear-gradient(0deg, transparent,#45ff55,#45ff55);
    transform-origin: bottom right;
    animation: animate 6s linear infinite;
}
.box::after{
    content: ";
    position: absolute;
    top: -50%;
    left: -50%;
    width: 380px;
    height: 420px;
    background: linear-gradient(0deg, transparent,#45ff55,#45ff55);
    transform-origin: bottom right;
    animation: animate 6s linear infinite;
    animation-delay: -3s;
}
@keyframes animate {
    0%{
        transform: rotate(0deg);
    }
    100%{
        transform: rotate(350deg);
    }
}

/* Form */
.form{
    position: absolute;
    inset: 2px;
    border-radius: 8px;
    background: #28292d;
    z-index: 10;
    padding: 40px 40px;
    display: flex;
    flex-direction: column;
}
.form h2{
    color: #45ff55;
    font-weight: 500;
    text-align: center;
    letter-spacing: 0.1em;
}
.inputBox{
    position: relative;
    width: 100%;
    margin-top: 35px;
}

.inputBox input{
    position: relative;
    width: 100%;
    padding: 10px 10px 10px;
    background: transparent;
    border: none;
    outline: none;
    color: #23242a;
    font-size: 1em;
    letter-spacing: 0.05em;
    z-index: 10;
}
.inputBox span{
    position: absolute;
    left: 0;
    padding: 20px 0px 10px;
    font-size: 1em;
    color: white;
    pointer-events: none;
    letter-spacing: 0.05em;
    transition: 0.5s;
}
.inputBox input:valid ~ span,
.inputBox input:focus ~ span{
    color: #45ff55;
    transform: translateX(0px) translateY(-34px);
    font-size: 0.75em;
}
.inputBox i{
    position: absolute;
    left: 0;
    bottom: 0;
    width: 100%;
    height: 2px;
    background: #45ff55;
    border-radius: 4px;
    transition: 0.5s;
    pointer-events: none;
    z-index: 0;
}
.inputBox input:valid ~ i,
.inputBox input:focus ~ i{
     height: 44px;
}

.links{
    display: flex;
    justify-content: space-between;

}

.links a {
    margin: 10px 0;
    font-size: 0.75em;
    color: #8f8f8f;
    text-decoration: none;
}
.links a:hover,
.links a:nth-child(2){
    color: #45ff55;
}

input[type="submit"]{
    border: none;
    outline: none;
    background: #45ff55;
    padding: 11px 25px;
    width: 100px;
    margin-top: 10px;
    border-radius: 4px;
    font-weight: 600;
    cursor: pointer;
}
input[type="submit"]:active{
    opacity: 0.8;
}

  .font{
    font-family: 'Poppins', sans-serif;
    top: 0;
    margin-left: 120%;
    text-align: left;
  }
  .fa{
    color:  aliceblue;


  }

  .navbar {
  overflow: hidden;
  background:linear-gradient(-13deg, transparent,#45ff55,#21e231);
  position: fixed;
  top: 0;
  width: 100%;
  font-size: 3vh;
}

.navbar a {
  float: right;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}

.navbar a:hover {
  background: #ddd;
  color: black;
```

```
}
.title{
    padding: 12px 12px;
    color: #f2f2f2;
}

    </style>
</head>
<body>
    <div class="navbar">
        <a href="/login">SIGN IN </a>
        <a href="/register">SIGN UP</a>
        <a href="/adminlogin">ADMIN</a>
        <div class="title">
            <h3  style="right:0;font-family:'Lobster', cursive;">InfiniteArt</h3>
          </div>
      </div>



    <form method="post">
        <div class="box">
            <div class="form">
                <h2>SIGN IN</h2>

                <div class="inputBox"  >

                    <input type="text" name="email"   required>
                    <span  class="fa fa-envelope"> <span class="font">Email</span></span>
                    <i></i>
                </div>

                <div class="inputBox">
```

```
                    <input type="password" name="password" required>
                    <span  class="fa fa-lock"> <span class="font">Password</span></span>
                    <i></i>
                </div>
                <p>{{success}}</p>
                <p style="color: red">{{error}}</p>
                <div class="links">
                    <a href="#">Forget password</a>
                    <a href="/register">SIGN UP</a>
                </div>
                <input type="submit" value="Login">
            </div>
        </div>
    </form>
</body>
</html>
```

Register.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Log in</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

    <link rel="stylesheet" href="{{url_for('static', filename='styles/style.css')}}">
</head>
<body>
    <div class="navbar">
        <a href="/login">SIGN IN </a>
```

```
            <a href="/register">SIGN UP</a>
            <a href="/adminlogin">ADMIN</a>
            <div class="title">
                <h3  style="right:0;font-family:'Lobster', cursive;">InfiniteArt</h3>
              </div>
          </div>
        <form method="post">
            <div class="box">
                <div class="form">
                    <h2>SIGN UP</h2>

                    <div class="inputBox"  >


                        <input type="text" name="username"   required>
                        <span  class="fa fa-user"> <span class="font">Username</span></span>
                        <i></i>
                    </div>
                    <div class="inputBox"  >


                        <input type="email" name="email"   required>
                        <span  class="fa fa-envelope"> <span class="font">Email</span></span>
                        <i></i>
                    </div>
                    <div class="inputBox"  >


                        <input type="text" name="phoneno"   required>
                        <span  class="fa fa-phone"> <span class="font">PhoneNumber</span></span>
                        <i></i>
                    </div>
                    <div class="inputBox">
                        <input type="password" name="password" required>
                        <span  class="fa fa-lock"> <span class="font">Password</span></span>
                    <i></i>
                </div>
                <p>{{success}}</p>
                <p style="color: red">{{error}}</p>
                <div class="links">
                    <a href="#">Forget password</a>
                    <a href="/login">SIGN IN</a>
                </div>
                <input type="submit" value="Login">
            </div>
        </div>
    </form>
</body>
</html>
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
    <link href="//db.onlinewebfonts.com/c/d8a3c95906aec0c2483082a82e72cb40?family=WanderlustShine-Regular" rel="stylesheet" type="text/css"/>

    <title>SMART FASHION RECOMMENDED APPLICATION</title>
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap');
```

```
         @import url('https://fonts.googleapis.com/css2?family=Lobster&display=swap');
         @font-face{
            font-family: Wanderlust;
            src: url('/Wanderlust Letters-Font/OTF/WanderlustLetters-Regular.otf');
            font-weight: bold;}
*{
    margin: 0;
    padding: 0;

    font-family: 'Poppins', sans-serif;
}
body{
    text-align: center;

    background-color: #23242a;
    color: #f2f2f2;

}

          .navbar {
            overflow: hidden;



            position: fixed;
            top: 0;
            width: 100%;
            }
        .navbar a {
        float: right;
        display: block;
        color: #21e231;
        text-align: center;
```

```
padding: 14px 16px;
text-decoration: none;
font-size: 17px;
}

.navbar a:hover {
background: #f2f2f2;
color: black;
}
.navbar h3{
   padding: 12px 12px;
   color: #21e231;
   font-size: 5vh;
   font-weight: 700;
   top:0;
   text-align: left;
}

.content {
width:100%;
height:100%;
display: table;
border:2px solid white;
}

.image{
   width:5%;
   height: 90%;
   float: left;
   padding: 2px;
}
.image img{
    height: 46px;
    width:80px;
}
.banner{
    width: 100%;
    height: 730px;
    background-image: url("{{url_for('static',filename='background.png')}}");
    background-attachment: fixed;
    background-size: cover;
}
.con{
    text-align: center;
    padding: 2%;
}
.con a{
    color: #f2f2f2;
    font-family: 'Poppins', sans-serif;
    font-weight: bolder;
    text-decoration: none;
    background: #21e231;
    padding: 5px;
    border-radius: 10px;

}
.con a:hover{
    background:#f2f2f2;
    color: #21e231;
    cursor: pointer;
    font-family: 'Poppins', sans-serif;
    font-weight: bolder;
}
.get{
    margin: 20% 20%;
}
.about,.cart,.chatbot{
    margin: 2%;
    width: 90%;
    height: 300px;
    box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
    background-color: #111111;
    transition: 0.3s;
    padding: 1%;
}
.about:hover,.cart:hover,.chatbot:hover {
    box-shadow: 0 8px 16px 0 rgb(52, 230, 28);
    }

.left{


    width: 10%;
    height: 100%;
    float: left;
    margin: 1%;




}
.about img,.chatbot img{
    width: 20%;
    height: 85%;
    float: left;
    top: 0;
    margin: 0% 6%;
}
```

```
        .cart img{
            width: 20%;
            height: 85%;
            float: right;
            top: 0;
            margin: 0% 6%;
        }
        .txt{

            width: 55%;
            float: right;
            text-align: justify;
            margin: 4% 5%;



        }

    </style>

</head>
<body>
        <div class="navbar">
            <a href="/login">SIGN IN </a>
            <div class="title">
           <!-- <div class="image"> <img src="{{url_for('static',
      filename='logo.png')}}"></div> -->
            <div><h3 style="right:0;font-family:'Lobster', cursive;">InfiniteArt </h3></div>
            </div>


        </div>
      <div class="banner">
        <div class="con">
```

```
  <div class="get"><h1><span style="color:#21e231; ">SMART FASHION
RECOMMENDER APPLICATION,</span> unlimitted access to all fashion
items</h1><br>

     <a href="/register"  name="get started">GET STARTED &#8594;</a></div>

 </div>

</div>

<br>

<div class="about">

  <center><h2>ABOUT</h2></center>

      <img src="{{url_for('static',filename='about.png')}}">


         <div class="txt">

             <p><h3 style="color:#21e231;">A warm welcome to your online fashion
lane.</h3>


             We are aware of how much you value fashion and how much you enjoy online
shopping's convenience. When you're on the go, the InfiniteArt online shopping app
makes sure you don't miss out on a chic shopping experience. If you want to shop for
clothing for men and women, shoes, accessories, or even the newest electronics and tech
gadgets, InfiniteArt is always there to meet all of your style needs

         </p></div>


</div>

<div class="cart">

  <center><h2>RECOMMENDATION&numsp;SYSTEM</h2></center>

      <img src="{{url_for('static',filename='cart.png')}}">


         <div class="txt">

             <p><h3 style="color:#21e231;">Your Favourite Items are here.</h3>


                 Here, we used a recommendation system that is to categorise the user's
clothing and suggest the best outfit for a particular occasion based on a recommendation
algorithm. The suggested system demonstrates that it can analyse the user's attire from the
images, determine the type and colour of the outfit, and then suggest the most appropriate
outfit for the situation based on the user's current attire.
```

```
          </p></div>


</div>
<div class="chatbot">
  <center><h2>CHATBOT</h2></center>
       <img src="{{url_for('static',filename='chatbot.png')}}">


          <div class="txt">
              <p><h3 style="color:#21e231;">Your Heleper Is Here.</h3>


                  Chatbots can also be used to gather visitor data, which can then be used to improve product recommendations and suggestions. You can personalise product pages and increase customer loyalty and affinity by having a thorough understanding of customer inquiries, needs, and preferences. Chatbots can also inform customers when an item is out of stock, suggest suitable alternatives based on their preferences, and let them know when their order is expected to arrive.
          </p></div>


</div>
<div class="devlopers"></div>
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "3e3ba4e3-0db1-48fc-854c-8dedf13768bf", // The ID of this integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "f8e994d6-a0d7-422f-a9f3-852a308d5f26", // The ID of your service instance.
    onLoad: function(instance) { instance.render(); }
  };
  setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
  });
```

```
    </script>
</body>
</html>
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-15173-1659594585

DEMO LINK:

https://drive.google.com/file/d/19ZRuA3cR40CBgtxStcq8qE0TUneVWI0i/view?usp=sharing