**MAHENDRA ENGINEERING COLLEGE FOR WOMEN**

**PROJECT REPORT**

**COMPUTER SCIENCE AND ENGINEERING**

**NAMAKKAL**

in

***Submitted by***

KAVITHA.M-419UCS032

MADHUBALA.E-419UCS036

JOTHIKA.V-419UCS027

JENIFERVINNARASI.P-419UCS25

***In partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

4.2 Non-Functional requirements

**5. PROJECT DESIGN**

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

**6. PROJECT PLANNING & SCHEDULING**

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

4.1 Functional requirement

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

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

**2. LITERATURE SURVEY**

**1. INTRODUCTION**

1.1 Project Overview

1.2 Purpose

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

**3. IDEATION & PROPOSED SOLUTION**

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

**4. REQUIREMENT ANALYSIS**

**11.CONCLUSION**

GitHub & Project Demo Link

Source Code

**13. APPENDIX**

**12. FUTURE SCOPE**

**10. ADVANTAGES & DISADVANTAGES**

9.1 Performance Metrics

**9. RESULTS**

8.2 User Acceptance Testing

8.1 Test Cases

**8. TESTING**

should be able to mention their preferences using interacting with chat bots. The user must

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

been bought, supervise the stock availability and interact with the buyer regarding the product

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.

used programs in the e-commerce field. The needs of people are continuously evolving,

**1. INTRODUCTION**

**Project Overview**

**1.1**

Fashion applications have seen tremendous growth and are now one of the most

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

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

and Nenni, Giustiniano, & Pirolo (2013), which also included several general approach

detailed theoretical contents were demonstrated in two separate studies by Liu et al. (2013)

trend prediction possible (Fung, Wong, Ho, & Mignolet, 2003). These forecasting models'

trends. Due to the data patterns being obtained over a set amount of time, it makes precise

et al., 2018) employed the autoregressive (AR) model (or ARMAX) to forecast style or

magazines, well-known e-commerce sites, fashion site blogs, and discussion forums, (Ngai

(AR) and Linear Regression Model Using photos pulled from social media, online fashion

Methodology: Auto Regression (AR) and Linear Regression Model. Auto Regression

forms. Because they were straightforward, quick, wellinformed, and simple to understand,

**2.2 References**

and the Nearest neighbour backed recommender to generate the final recommendations.

to that. We use neural networks to process the images from Fashion Product Images Dataset

people see something that they are interested in and tend to look for products that are similar

of a product given as input by the user to generate recommendations since many-a-time

SARIMA were frequently employed to assess the sales of clothing. A technique for

predictions has since followed. In order to prevent overfitting, grouping items in predictive

introduction of dynamic pricing models to support markdown choices in multi-item group

models in multi-processing groups with both positive and negative commodities. The

forecasting retail products was proposed by Demerit (2018). weekly using linear regression

and

ARIMA,

smoothing,

exponential

auto-regression,

including

techniques

statistical

tendency of humans has led to the development of the fashion industry over the course of

Humans are inevitably drawn towards something that is visually more attractive. This

gradually moved towards fashion that is concerned to be a popular aesthetic expression.

Fashion Recommender system with an increase in the standard of living, peoples' attention

time. However, given too many options of garments on the e-commerce websites, has

**2.1 Existing problem**

**2. LITERATURE SURVEY**

Fashion

that rely on the user's previous purchases and history, this project aims at using an image

recommendations for the user based on an input given. Unlike the conventional systems

generates

that

system

Recommender

personalized

a

proposed

we

project,

presented new challenges to the customers in identifying their correct outfit. Thus, in this

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

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. 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.

ratings from the users who shared your interests in step one.

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

2. Paper Title: Image Based Fashion Recommeder System

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.

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

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 Products via our application.

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

systems that are built for various aspects is undertaken in this research. Systems created to

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

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:



creative solutions. Use this template in your own brainstorming sessions so your team can

**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 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

unleash their imagination and start shaping concepts even if you're not sitting in the same

room.

**3.IDEATION & PROPOSED SYSTEM**

**3.1 Empathy Map Canvas**

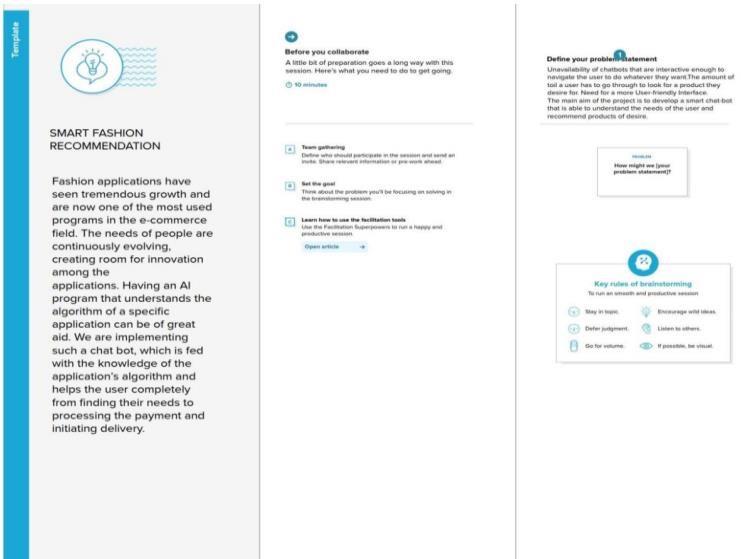
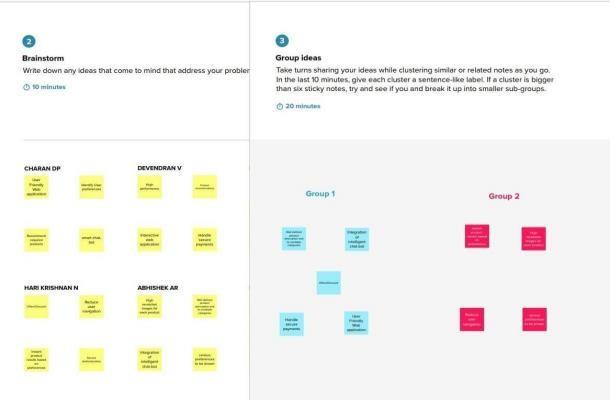
An empathy map is a simple, easy-to-digest visual that captures knowledge about a

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

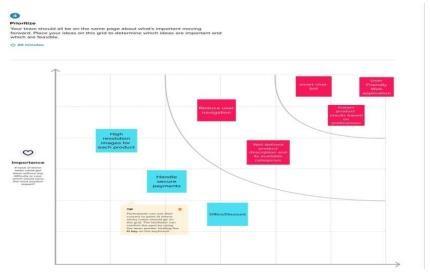
is experiencing it. The exercise of creating the map helps participants consider things from

the user’s perspective along with his or her goals and challenges.



Step-2: Brainstorm, Idea Listing and Grouping

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



Effective recommendation of products.

and services.



Improve customer relationship, interactivity

By using Smart fashion recommender application:

Idea / Solution description



2.



Lack of proper guidance.



Complex User Interface.



Lack of sales



Proper guidance in accessing application.



Reduce human error



Collect feedback instantly.

chat-bot



Recommendation within a single page via

**Parameter**

**S.No.**

**Description**

**3.3 Proposed Solution**

Step-3: Idea Prioritization



Confusion in choosing product

to choose right product



User need to navigate across multiple pages

user

be solved)



Lack of interaction between application and

Problem Statement (Problem to

1.

with product in browser cookie will enable to

the application. Storing user preferences along

server responsible for certain functionality of

micro service architecture provided that each



The solution can be made scalable by using

Scalability of the Solution

6.

provide response instantly and allows for fetching

user interface.

cost with high performance and interactive

Model)



The application can be developed at minimum

Business Model (Revenue

related products.

product thus increases sales.

various filters. Reduces time in choosing right

user without making them to search through

which recommends appropriate products to the



Chat-bot asks and learns from user preference

Novelty / Uniqueness

3.

or after placing order is one of the most

5.

satisfaction and providing better services.

customer

deriving

in

factor

important

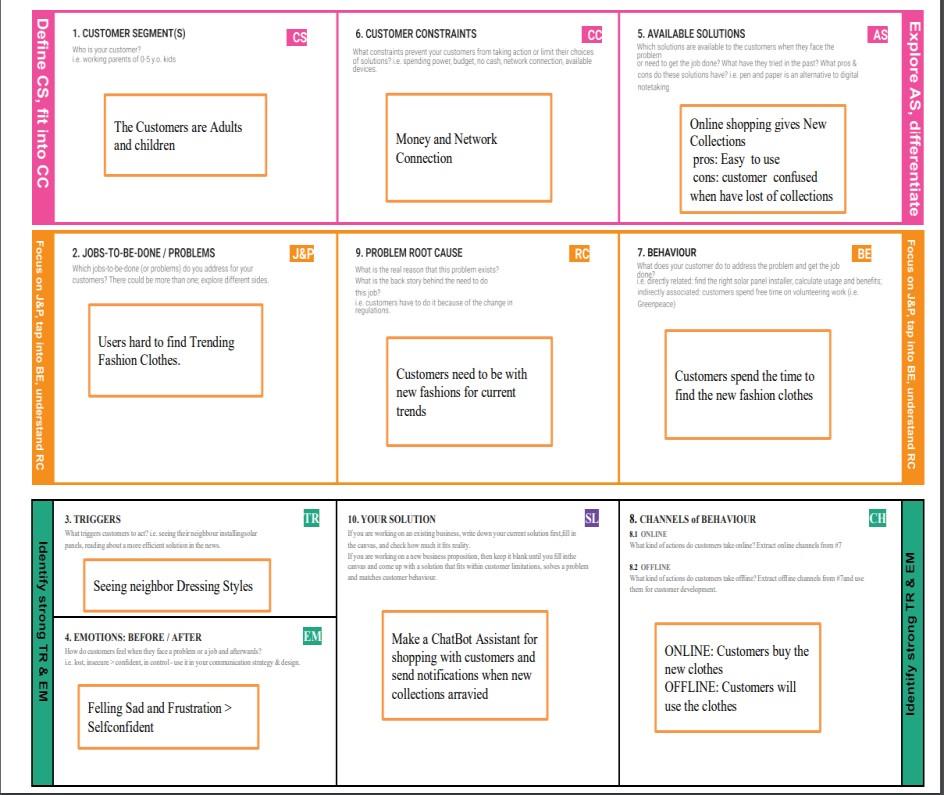
Satisfaction



Feedback from the user at the end of session

Social Impact / Customer

4.



**3.4 Problem Solution Fit**

NFR-1

Using Android or IOS or windows applications.

**Usability**

NFR-2

The user data is stored securely in IBM cloud.

**Security**

**4.2**

**Non-Functional requirements**

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

NFR-3

**FR**

**Non-Functional Requirement**

**Description**

**No.**

NFR-6

It’s easy to scalable size of users and products.

**Scalability**

**Performance**

The Quality of the services are trusted.

**Reliability**

NFR-4

Its Provide smooth user experience.

NFR-5

The services are available for 24/7.

**Availability**

User Registration

Registration through Form

FR-1

**(Epic)**

User Interaction

Interact through the Chat Bot

FR-2

**4. REQUIREMENT ANALYSIS**

**4.1**

**Functional requirement**

Following are the functional requirements of the proposed solution.

**Functional Requirement**

**FR No.**

**Sub Requirement (Story / Sub-Task)**

Return Products

Through the chat Bot

FR-5

New Collections

Recommended from chat Bot

FR\_6

Buying Products

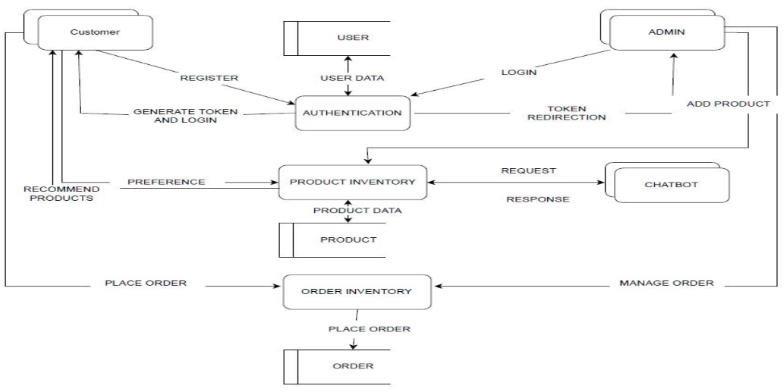
Through the chat Bot Recommendation

FR-3

Track Products

Ask the Chat Bot to Track my Orders

FR-4



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.

(DFD)

overview whereas a project plan includes more detailed information. There are seven steps

**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

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.

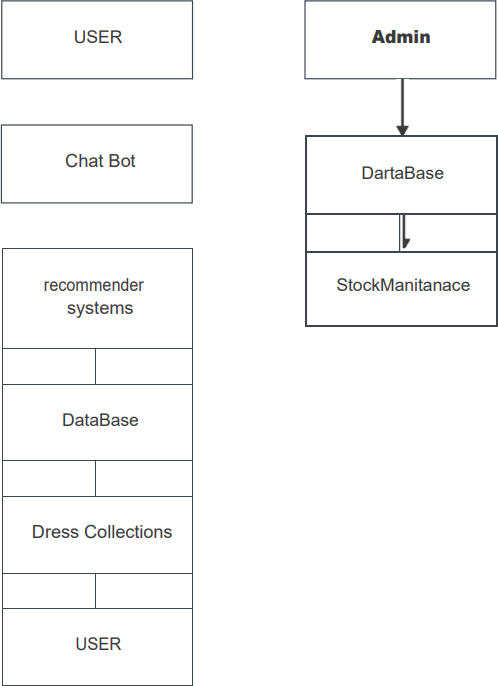
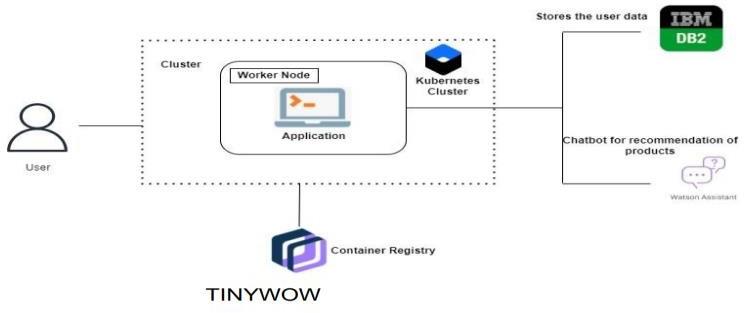
**5.1 Data Flow Diagrams**

A

Data

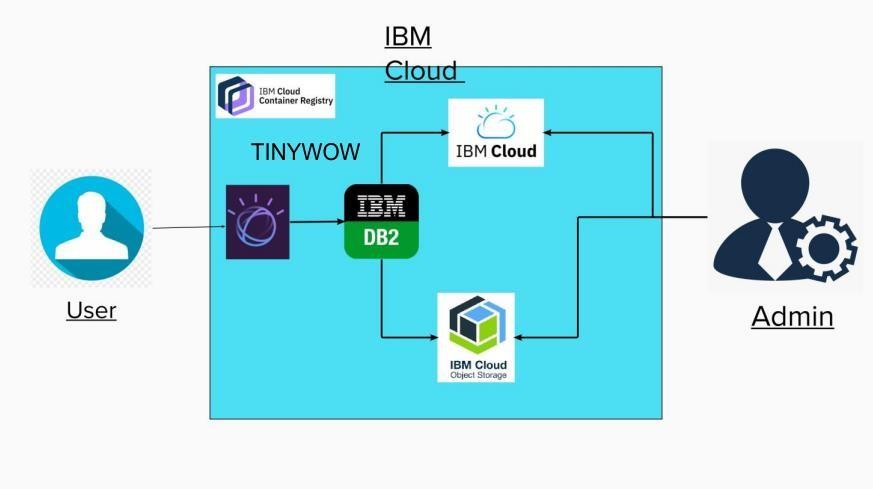
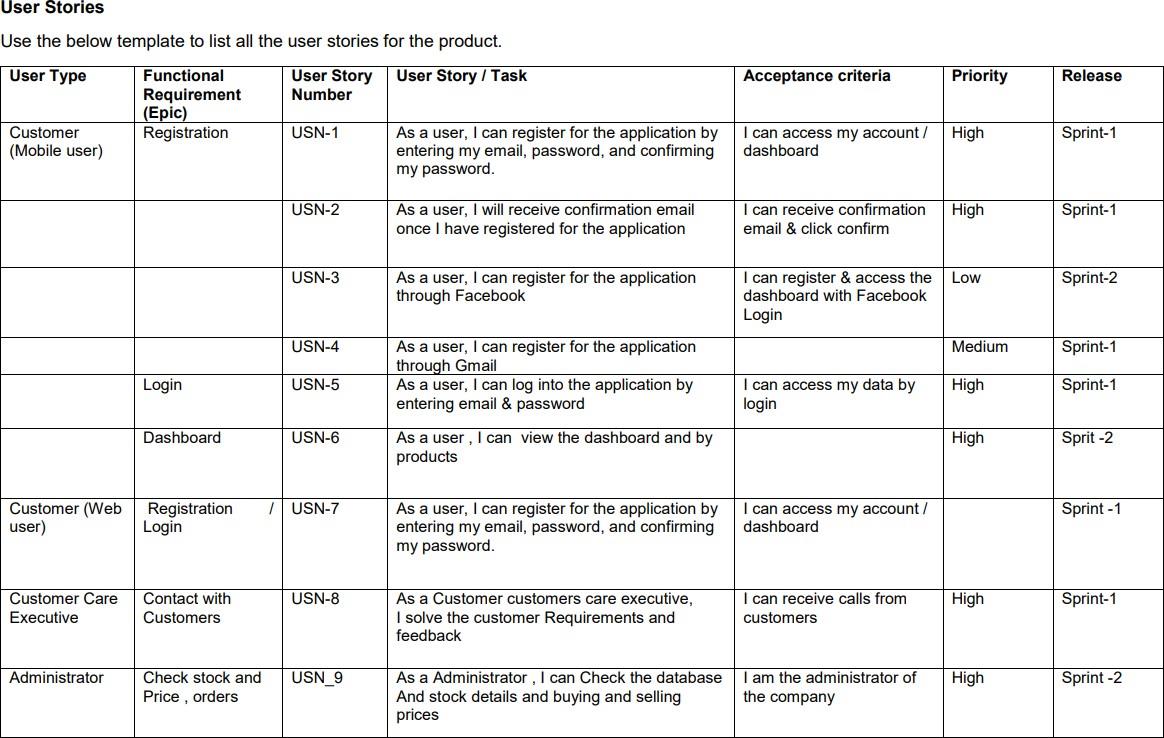
Flow

Diagram



**SOLUTION ARCHITECTURE**

**5.2 Solution & Technical Architecture**

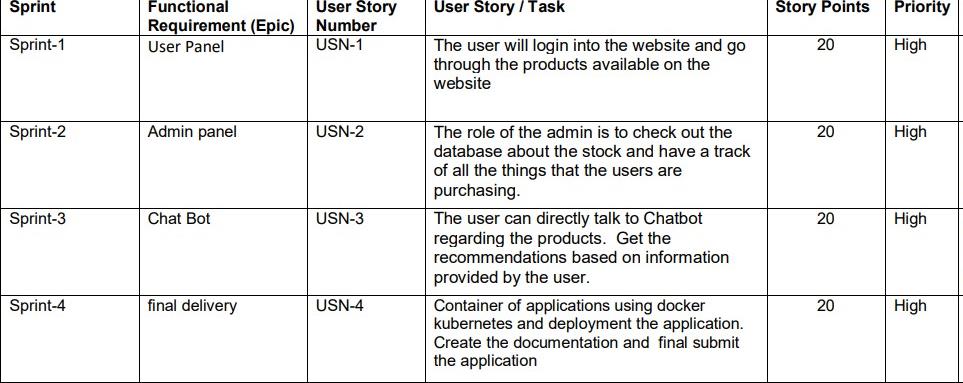


perspective of the end user or customer. The purpose of a user story is to articulate how a

A user story is an informal, general explanation of a software feature written from the

**5.3 User Stories**

**TECHNICAL ARCHITECTURE:**



20

6 Days

12 Nov 2022

20

12 Nov 2022

2022

14 Nov

19 Nov 2022

05 Nov 2022

20

Sprint-4

05 Nov 2022

2022

07 Nov

Sprint-3

20

6 Days

19 Nov 2022

6 Days

20

2022

**Sprint**

**Duration**

**Story**

**Start**

**End**

**Completed (as on**

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

**6. PROJECT PLANNING & SCHEDULING**

**6.1**

**SPRINT PLANNING & ESTIMATION**

**Release**

**6.2 SPRINT DELIEVERY SCHEDULE**

**Total**

**Sprint**

**Sprint**

**Story Points**

**Sprint**

2022

29 Oct 2022

20

29 Oct 2022

31 Oct

Sprint-2

20

**Points**

**Date**

**Date**

**Planned End**

**Date**

**(Planne**

**Date)**

**(Actual)**

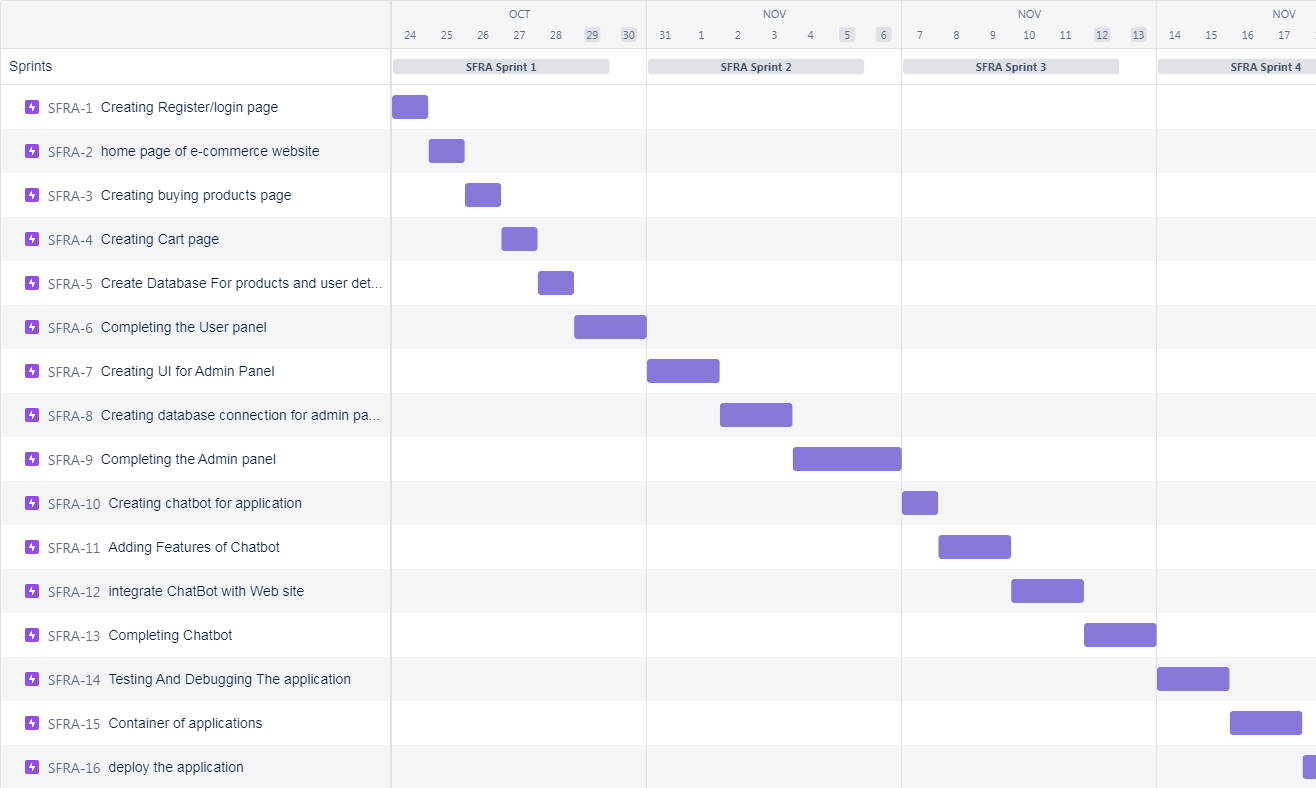
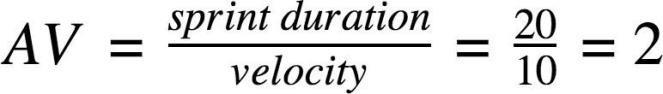
**d)**

24 Oct

Sprint-1

20

6 Days



Imagine we have a 10-day sprint duration, and the velocity of the team is 20

(story points per day)

(points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit

**VELOCITY:**

**6.3 Burndown Chart:**

<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>

<link rel="stylesheet" href="{{url\_for('static',filename='styles/home.css')}}">

</head>

<body>

<div class="navbar">

<a href="/logout">LOG OUT</a>

<div class="title">

<meta charset="UTF-8">

**7. CODING & SOLUTIONING**

<!DOCTYPE html>

<html lang="en">

<head>

<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">

</div>

<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}}">

<h2>TRENDING SHIRTS</h2>

<div class="descrpton">

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

<div class="co">

<h3 style="right:0;font-family:'Lobster', cursive; color: #fff;margin:

1%;">InfiniteArt</h3>

</div>

</div>

<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>

<div class="splide\_\_track">

</div>

</div>

</li>

{% endfor %}

<!-- pant -->

</ul>

</div>

</div>

<H2>TRENDING PANTS</H2>

<div id="pant" class="splide" style="margin:0;background: #131313;" >

<ul class="splide\_\_list">

{% for item in pants %}

<li class="splide\_\_slide">

<div class="container">

<div class="card">

<span>10</span>

<div class="size">

<span>7</span>

<span>8</span>

<span>9</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>

</li>

<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>

</div>

{% endfor %}

<!-- pant -->

</ul>

</div>

</div>

</div>

<div class="image">

<img src="{{item.IMAGE}}">

<div class="descrpton">

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

<div class="size">

<span>7</span>

<span>8</span>

<span>9</span>

<span>10</span>

<span></span>

<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>

</div>

<a href="#">Buy now</a>

</div>

{% for item in watchs %}

<h2>TRENDING WATCHES</h2>

<div id="watch" class="splide" style="padding: 0;margin:0;background: #131313;" >

<div class="splide\_\_track">

<ul class="splide\_\_list">

<li class="splide\_\_slide">

<div class="container">

<div class="card">

<div class="image">

<img src="{{item.IMAGE}}">

</div>

<div class="descrpton">

<span>7</span>

<div class="image">

<img src="{{item.IMAGE}}">

</div>

<div class="descrpton">

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

<div class="size">

<div class="card">

<span>8</span>

<span>9</span>

<span>10</span>

</div>

<h3>{{item.RATE}}</h3>

<!-- pant -->

</div>

</div>

</li>

{% endfor %}

</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">

perPage: 4,

<script>

var splide = new Splide( '.splide', {

type : 'loop',

perPage : 4,

autoplay: true,

} );

splide.mount();

document.addEventListener('DOMContentLoaded', function () {

new Splide('#pant', {

</div>

perMove: 1,

gap: "30px",

pagination: false,

}).mount();

});

<span></span>

<div class="color">

<h3>Color: </h3>

<span></span>

<span></span>

</div>

<a href="#">Buy now</a>

</div>

</div>

</div>

</li>

{% endfor %}

<!-- pant -->

</ul>

</div>

splid.mount();

}).mount();

});

</script>

<script>

var splid = new Splide( '.splid', {

type : 'loop',

perPage : 4,

autoplay: true,

} );

pagination: false,

</script>

<script>

var myIndex = 0;

carousel();

perMove: 1,

document.addEventListener('DOMContentLoaded', function () {

new Splide('#watch', {

perPage: 4,

gap: "30px",

pagination: false,

}).mount();

});

document.addEventListener('DOMContentLoaded', function () {

new Splide('#ring', {

perPage: 4,

perMove: 1,

gap: "30px",

a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SECURI

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-

App.py

TY=SSL;

SSLServerCertificateDigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=nhl80748;PWD

=3yD0G9e6VuQHsOBX;", "", "")

#url\_for('static', filename='style.css')

x[i].style.display = "none";

function carousel() {

var i;

var x = document.getElementsByClassName("mySlides");

for (i = 0; i < x.length; i++) {

}

myIndex++;

if (myIndex > x.length) {myIndex = 1}

x[myIndex-1].style.display = "block";

setTimeout(carousel, 2000); // Change image every 2 seconds

}

</script>

</body>

</html>

ibm\_db.bind\_param(stmt,2,phoneno)

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)

if request.method == 'POST':

ibm\_db.execute(stmt)

isUser = ibm\_db.fetch\_assoc(stmt)

if not isUser:

insert\_sql = "INSERT INTO user\_detail(username, email, phoneno, password) VALUES

(?,?,?,?)"

@app.route("/",methods=['GET'])

app = Flask(\_\_name\_\_)

app.secret\_key = b'\_5#y2L"F4Q8z\n\xec]/'

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 not isUser:

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)

email = request.form['email']

return render\_template('login.html',error='Invalid Credentials')

isPasswordMatch = bcrypt.checkpw(password.encode('utf-

8'),isUser['PASSWORD'].encode('utf-8'))

ibm\_db.bind\_param(prep\_stmt, 4, hash)

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.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':

VALUES (?,?,?,?)"

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)

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)

return redirect(url\_for('home'))

if not isPasswordMatch:

return render\_template('login.html',error='Invalid Credentials')

session['email'] = isUser['EMAIL']

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']

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'))

query = "SELECT \* FROM admin\_detail WHERE email=?"

if not isPasswordMatch:

return render\_template('adminlogin.html',error='Invalid Credentials')

session['email'] = isUser['EMAIL']

return redirect(url\_for('home'))

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')

insert\_sql = "INSERT INTO WATCH(name, image, rate) VALUES (?,?,?)"

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':

ibm\_db.bind\_param(prep\_stmt, 2, image)

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)

def addproduct():

return render\_template('adminlogin.html',name='Home')

@app.route("/addproduct",methods=['GET','POST'])

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)

#selecting\_pant

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)

watch\_list=[]

sql1="SELECT \* FROM PANT"

stmt1 = ibm\_db.exec\_immediate(conn, sql1)

pant=ibm\_db.fetch\_both(stmt1)

while pant != False :

ibm\_db.bind\_param(prep\_stmt, 2, image)

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, 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=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():

stmt3 = ibm\_db.exec\_immediate(conn, sql3)

sql2="SELECT \* FROM WATCH"

stmt2 = ibm\_db.exec\_immediate(conn, sql2)

watch=ibm\_db.fetch\_both(stmt2)

while watch != False :

watch\_list.append(watch)

#selecting\_watch

pant\_list.append(pant)

pant = ibm\_db.fetch\_both(stmt1)

print(pant\_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)

#selecting\_rings

sql3="SELECT \* FROM RINGS"

session.pop('email', None)

watch = ibm\_db.fetch\_both(stmt2)

print(watch\_list)

return render\_template('home.html',watchs=watch\_list)

@app.route('/logout')

def logout():

return redirect(url\_for('login'))

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

6

1

0

3

2

External

31

15

4

2

10

Fixed

5

5

By Design

2

**Subtotal**

**Severity 4**

4

0

3

0

1

Duplicate

15

3

8

1

2

5

0

Won't Fix

67

21

13

15

18

Totals

0

1

0

1

0

Not Reproduced

1

2

**Severity 3**

1

0

0

Skipped

5

0

0

5

Register

5

0

0

5

Login

2

0

0

2

Home Page

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

**8.1 Test Cases**

**8.TESTING**

**Pass**

**Fail**

**Not Tested**

**Total Cases**

**Section**

**8.2**

and how they were resolved

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

**Defect Analysis**

Acceptance Testing (UAT).

of the Smart Fashion Recommender Application project at the time of the release to User

The purpose of this document is to briefly explain the test coverage and open issues

**Purpose of Document**

**USER ACCEPTANCE TESTING**

**Severity 2**

**Severity 1**

**Resolution**

2

2

0

0

2

Order page

0

0

2

Final Report Output

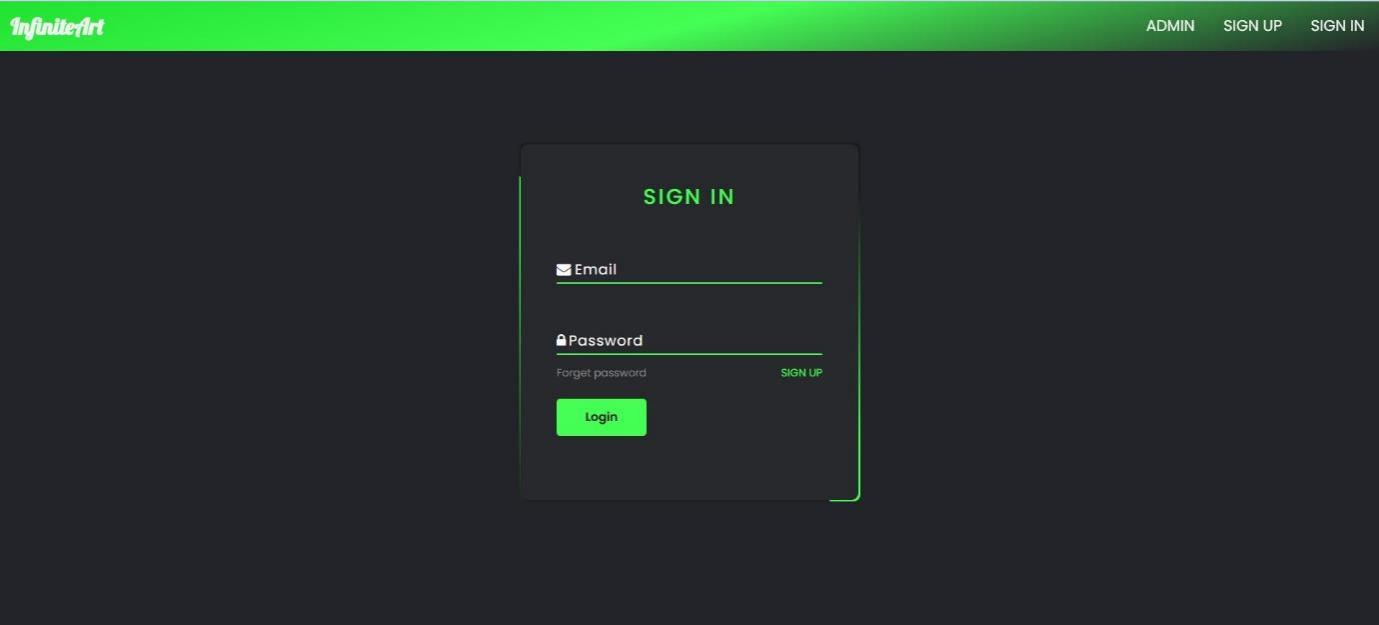
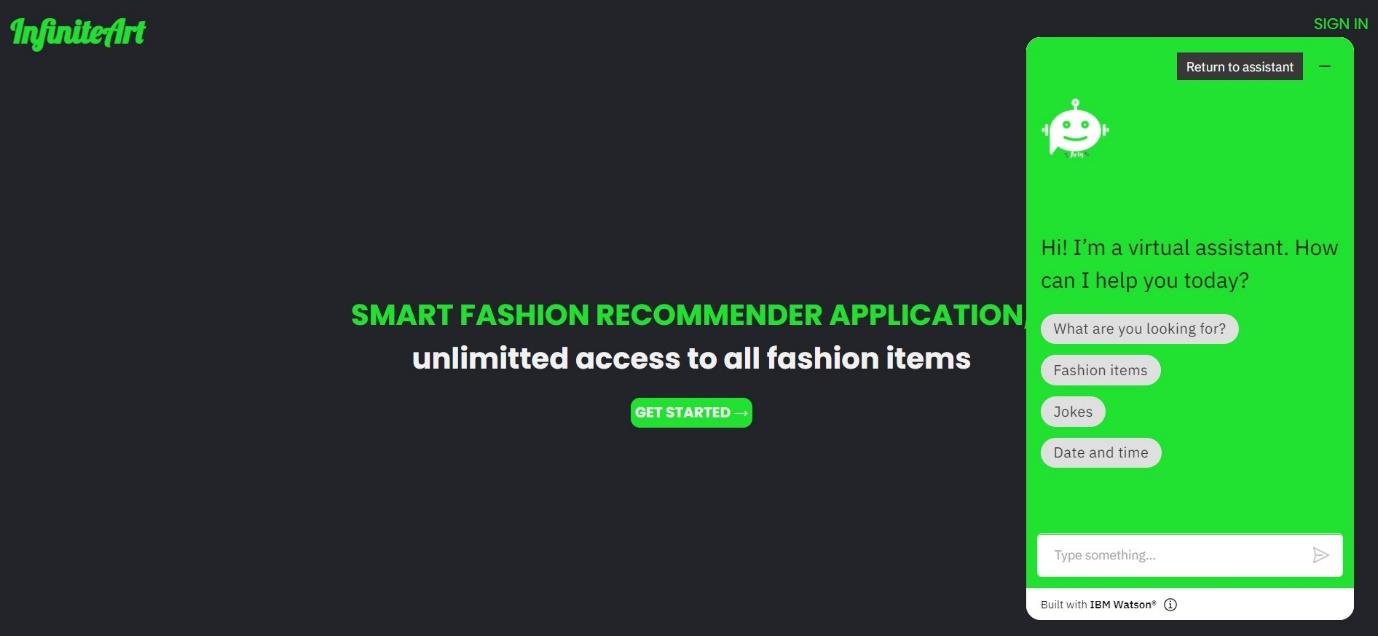
5

0

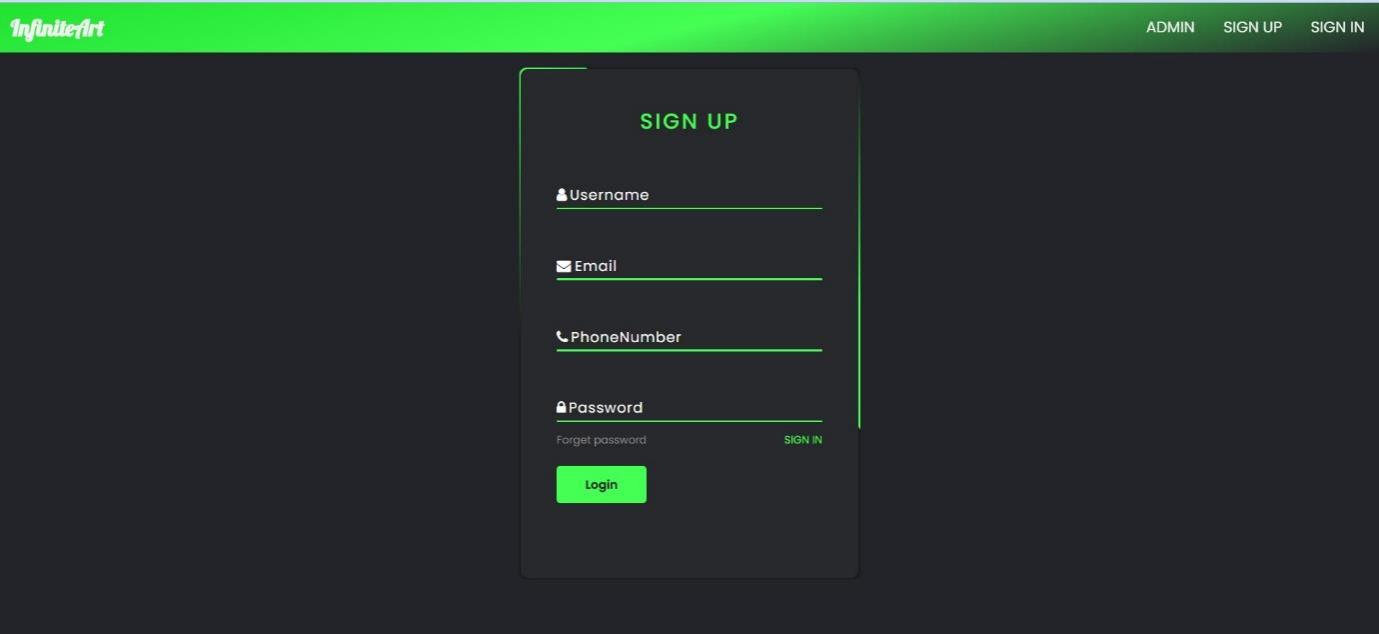
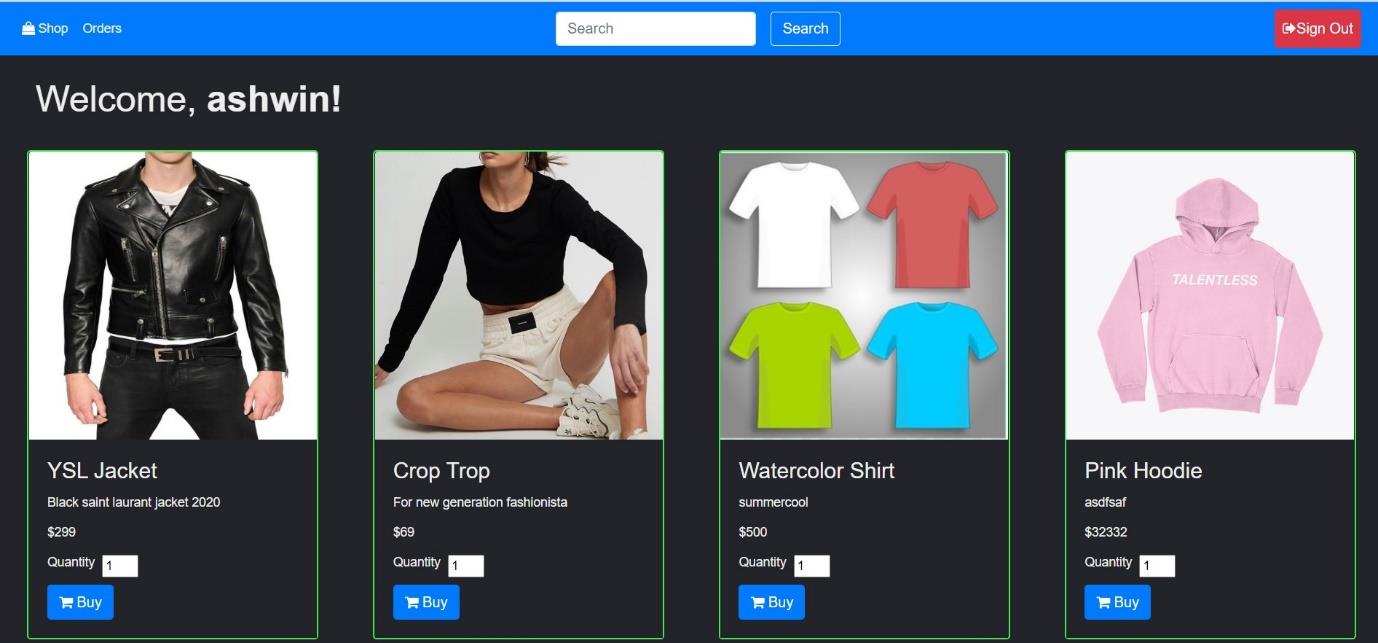
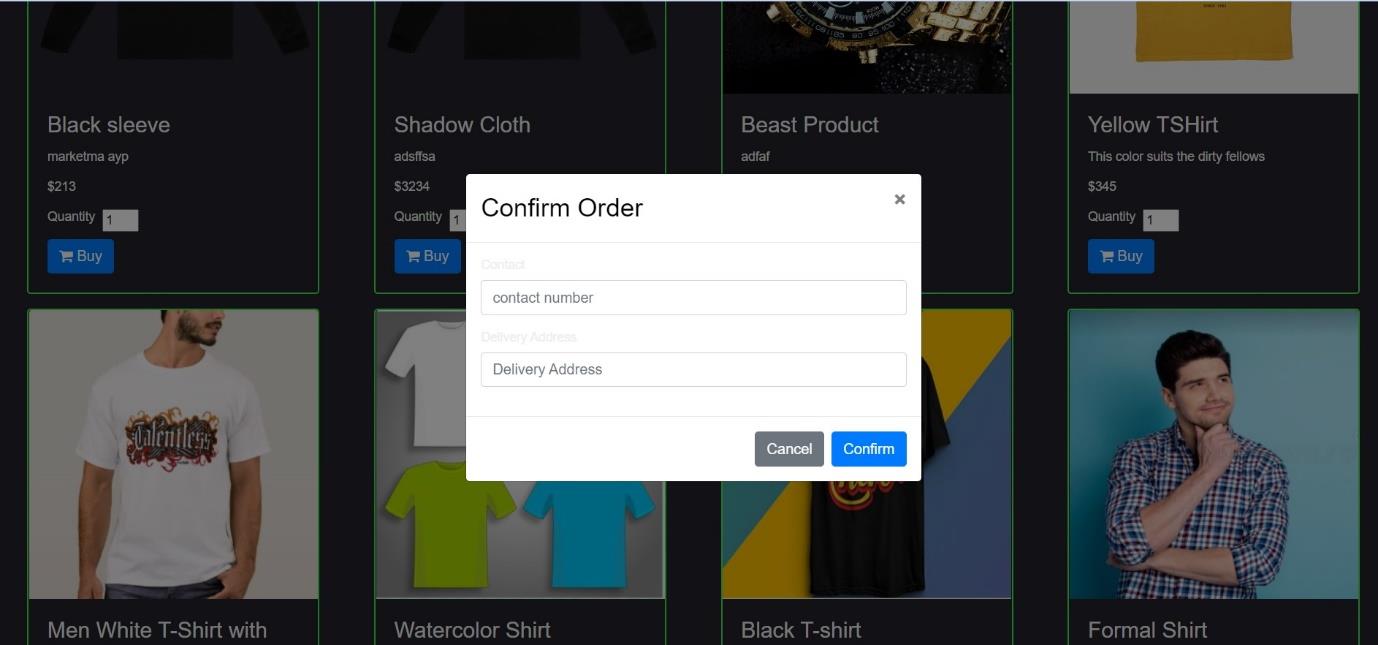
0

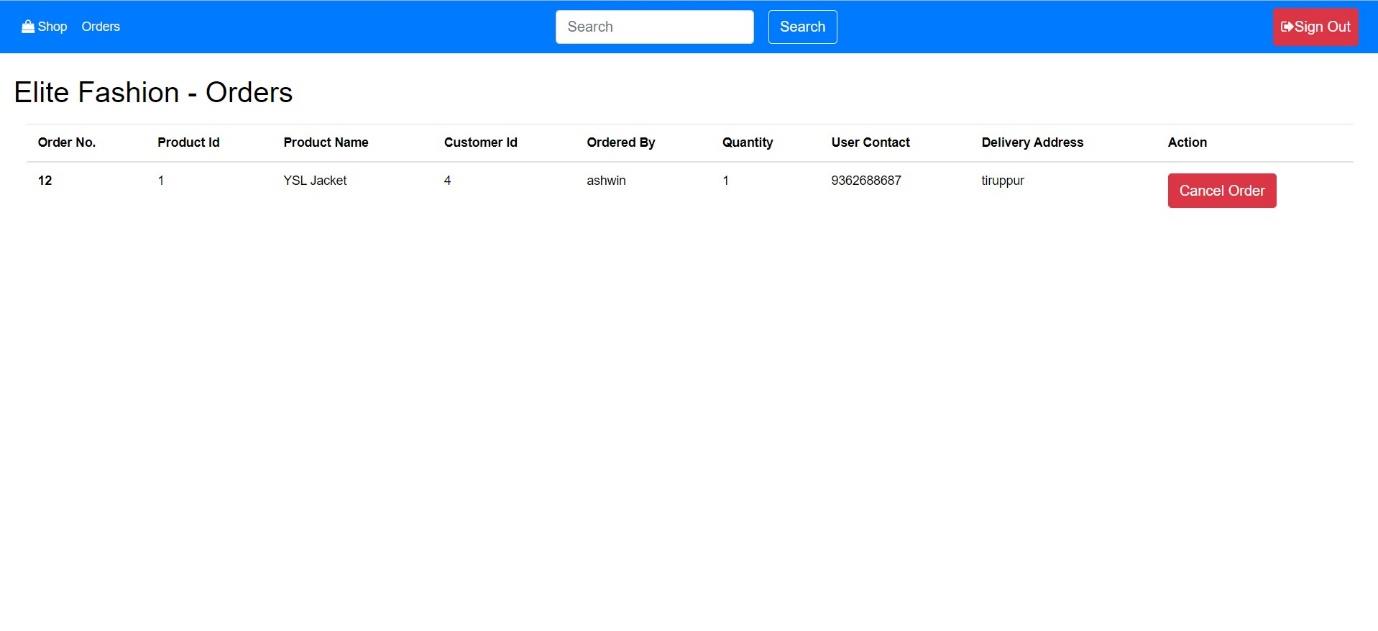
5

Order products



**9. RESULTS**





1.

Hours worked : 50 hours

2.

Sticking to Timelines : 100%

3.

Consistency of the product : 75%

4.

Efficiency of the product : 80%

algorithms.

5.

Quality of the product : 85%

on the type of filtering technique. Root Mean Square Error (RMSE), Receiver Operating

**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

Characteristics (ROC), Area Under Cover (AUC), Precision, Recall and F1 score is

generally

used

to

evaluate

the

performance

or

accuracy

of

the

recommendation

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.

significant drop in recommendation quality while reducing recommendation response

the system is dependent completely on the clothes present in the user’s wardrobe. Also

**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

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

purpose, we created a plan for collecting data and for developing the steps needed for

**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

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.

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 used to

quickly explore fashion items from different online databases to provide prompt

recommendations to users or consumers.

history and product reviews from the recommendation system and subsequently use

images,

**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

textual content,

demographic

history,

and

cultural

information is crucial in developing recommendation frameworks. Product attributes

and clothing style matching are common features of collaborative and 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 product attribute will be helpful in increasing the precision of the model. The use

of virtual sales advisers in an online shopping portal would provide consumers with a

real time offline shopping experience. Retailers can collect the data on users’ purchase

</style>

<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;0,40

0;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;

}

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

</head>

<body>

<div class="navbar">

<a href="/logout">LOG OUT</a>

<head>

**13. APPENDIX**

Home.html

<!DOCTYPE html>

<html lang="en">

<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/spli

de.min.css"/>

<meta content="text/html; charset=iso-8859-2" http-equiv="Content-Type">

<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>

<div class="title">

<h3 style="right:0;font-family:'Lobster', cursive; color: #fff;margin:

1%;">InfiniteArt</h3>

</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 PANTS</H2>

</div>

</div>

</li>

{% endfor %}

<!-- pant -->

</ul>

</div>

</div>

</div>

<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">

<span>9</span>

<div class="size">

<span>7</span>

<span>8</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>

<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>

<span>9</span>

</div>

</div>

</li>

{% endfor %}

<!-- pant -->

<img src="{{item.IMAGE}}">

<div class="container">

<div class="card">

<div class="image">

</div>

<div class="descrpton">

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

<div class="size">

<span>7</span>

<span>8</span>

</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>

<div id="watch" class="splide" style="padding: 0;margin:0;background: #131313;" >

</ul>

</div>

</div>

<h2>TRENDING WATCHES</h2>

<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 class="descrpton">

{% for item in rings %}

<li class="splide\_\_slide">

<div class="container">

<div class="card">

<div class="image">

<img src="{{item.IMAGE}}">

</div>

<ul class="splide\_\_list">

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

<div class="size">

<span>7</span>

</div>

<span></span>

</div>

<a href="#">Buy now</a>

</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">

} );

{% endfor %}

<!-- pant -->

</ul>

</div>

</div>

<script>

var splide = new Splide( '.splide', {

type : 'loop',

perPage : 4,

autoplay: true,

</li>

splide.mount();

document.addEventListener('DOMContentLoaded', function () {

new Splide('#pant', {

<h3>{{item.RATE}}</h3>

<span>8</span>

<span>9</span>

<span>10</span>

</div>

<div class="color">

<h3>Color: </h3>

<span></span>

<span></span>

<span></span>

</div>

<a href="#">Buy now</a>

</div>

</div>

</div>

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,

}).mount();

perPage: 4,

perMove: 1,

gap: "30px",

pagination: false,

});

document.addEventListener('DOMContentLoaded', function () {

new Splide('#watch', {

perPage: 4,

perMove: 1,

gap: "30px",

pagination: false,

}).mount();

});

<meta charset="UTF-8">

setTimeout(carousel, 2000); // Change image every 2 seconds

}

</script>

</body>

</html>

Login.html

<!DOCTYPE html>

<html lang="en">

<head>

x[myIndex-1].style.display = "block";

<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">

var myIndex = 0;

} );

splid.mount();

</script>

<script>

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}

.box::before{

background-color: #23242a;

}

.box{

position: relative;

width: 380px;

height: 400px;

background-color: #1c1c1c;

border-radius: 8px;

overflow: hidden;

}

min-height: 100vh;

content: '';

position: absolute;

top: -50%;

left: -50%;

width: 380px;

@import url('https://fonts.googleapis.com/css2?family=Lobster&display=swap');

<style>

@import

url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&disp

lay=swap');

\*{

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Poppins', sans-serif;

}

body{

display: flex;

justify-content: center;

align-items: center;

/\* Form \*/

}

@keyframes animate {

0%{

transform: rotate(0deg);

}

100%{

transform: rotate(350deg);

}

}

animation-delay: -3s;

.form{

position: absolute;

inset: 2px;

border-radius: 8px;

background: #28292d;

}

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;

letter-spacing: 0.05em;

.inputBox input{

position: relative;

width: 100%;

padding: 10px 10px 10px;

background: transparent;

border: none;

outline: none;

color: #23242a;

font-size: 1em;

}

z-index: 10;

}

.inputBox span{

position: absolute;

left: 0;

}

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:focus ~ i{

bottom: 0;

width: 100%;

height: 2px;

background: #45ff55;

border-radius: 4px;

transition: 0.5s;

pointer-events: none;

z-index: 0;

}

.inputBox input:valid ~ i,

left: 0;

height: 44px;

}

.links{

display: flex;

letter-spacing: 0.05em;

padding: 20px 0px 10px;

font-size: 1em;

color: white;

pointer-events: none;

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;

input[type="submit"]:active{

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"]{

opacity: 0.8;

}

.font{

font-family: 'Poppins', sans-serif;

.links a {

justify-content: space-between;

}

margin: 10px 0;

font-size: 0.75em;

color: #8f8f8f;

text-decoration: none;

}

.links a:hover,

.links a:nth-child(2){

color: #45ff55;

}

font-size: 17px;

font-size: 3vh;

}

.navbar a {

float: right;

display: block;

color: #f2f2f2;

text-align: center;

padding: 14px 16px;

text-decoration: none;

width: 100%;

}

.navbar a:hover {

background: #ddd;

color: black;

.fa{

top: 0;

margin-left: 120%;

text-align: left;

}

color: aliceblue;

}

.navbar {

overflow: hidden;

background:linear-gradient(-13deg, transparent,#45ff55,#21e231);

position: fixed;

top: 0;

<input type="text" name="email" required>

</div>

<form method="post">

<div class="box">

<div class="form">

<h2>SIGN IN</h2>

<div class="inputBox" >

</div>

<span class="fa fa-envelope"> <span class="font">Email</span></span>

<i></i>

</div>

<div class="inputBox">

}

}

.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>

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">

</html>

<link rel="stylesheet" href="{{url\_for('static', filename='styles/style.css')}}">

</head>

<body>

<div class="navbar">

<a href="/login">SIGN IN </a>

<p>{{success}}</p>

<input type="password" name="password" required>

<span class="fa fa-lock"> <span class="font">Password</span></span>

<i></i>

</div>

<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>

<span class="fa fa-phone"> <span class="font">PhoneNumber</span></span>

</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>

<i></i>

<i></i>

</div>

<div class="inputBox">

<input type="password" name="password" required>

<span class="fa fa-lock"> <span class="font">Password</span></span>

</div>

<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>

<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>

<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=Wanderl

ustShine-Regular" rel="stylesheet" type="text/css"/>

<html lang="en">

<title>SMART FASHION RECOMMENDED APPLICATION</title>

<style>

@import

url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&disp

lay=swap');

<a href="#">Forget password</a>

<i></i>

</div>

<p>{{success}}</p>

<p style="color: red">{{error}}</p>

<div class="links">

<a href="/login">SIGN IN</a>

</div>

<input type="submit" value="Login">

</div>

</div>

</form>

</body>

</html>

Index.html

<!DOCTYPE html>

}

}

.navbar {

overflow: hidden;

position: fixed;

top: 0;

width: 100%;

color: #f2f2f2;

.navbar a {

float: right;

display: block;

color: #21e231;

text-align: center;

font-weight: bold;}

@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');

\*{

margin: 0;

padding: 0;

font-family: 'Poppins', sans-serif;

}

body{

text-align: center;

background-color: #23242a;

width:5%;

}

.content {

width:100%;

height:100%;

display: table;

border:2px solid white;

}

.image{

text-align: left;

height: 90%;

float: left;

padding: 2px;

}

.image img{

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;

color: #21e231;

font-family: 'Poppins', sans-serif;

font-weight: bolder;

text-decoration: none;

background: #21e231;

padding: 5px;

border-radius: 10px;

}

.con a:hover{

background:#f2f2f2;

color: #f2f2f2;

cursor: pointer;

font-family: 'Poppins', sans-serif;

font-weight: bolder;

}

.get{

width: 100%;

height: 46px;

width:80px;

}

.banner{

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{

width: 20%;

width: 10%;

height: 100%;

float: left;

margin: 1%;

}

.about img,.chatbot img{

.left{

height: 85%;

float: left;

top: 0;

margin: 0% 6%;

}

width: 90%;

margin: 20% 20%;

}

.about,.cart,.chatbot{

margin: 2%;

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);

}

<div><h3 style="right:0;font-family:'Lobster', cursive;">InfiniteArt </h3></div>

</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>

</div>

<div class="banner">

<div class="con">

top: 0;

.cart img{

width: 20%;

height: 85%;

float: right;

margin: 0% 6%;

}

.txt{

width: 55%;

float: right;

text-align: justify;

margin: 4% 5%;

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

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">

shopping's convenience. When you're on the go, the InfiniteArt online shopping app

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.

</div>

<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>

<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

setTimeout(function(){

</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(); }

};

</p></div>

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);

});

<img src="{{url\_for('static',filename='chatbot.png')}}">

</p></div>

</div>

<div class="chatbot">

<center><h2>CHATBOT</h2></center>

<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.

<https://github.com/IBM-EPBL>

<https://youtu.be/lHjHkrKb7N4>

**DEMO LINK:**

**GITHUB LINK:**

</html>

</body>

</script>