IBM – NALAIYA THIRAN PROJECT

SMART FASHION RECOMMENDER APPLICATION

INDUSTRY MENTOR : KRISHNA CHAITANYA

FACULTY MENTOR : SENTHILNATHAN.N

ANNA UNIVERSITY REGIONAL CAMPUS TIRUCHIRAPPALLI

Electronics and communication Engineering Department

SUBMITTED BY:

TEAM ID : PNT2022TMID32443

TEAM LEADER :SAJEETH GURU G.P

TEAM MEMBER : MADHAVAN S

TEAM MEMBER : ROHITH R

TEAM MEMBER : SAMSUL GHUDHA A.S

TABLE OF CONTENTS

1. INTRODUCTION

:

INTRODUCTION:

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 processses 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.

OBJECTIVE:

Project Description We have developed a new innovative solution through which you can directly do your online shopping based on your choice without any search. It can be done by using the chat bot. In this project you will be working on two modules:

1. Admin and 2. User

2. LITERATURE SURVEY

1. **Paper Title : A Comprehensive Review on Online Fashion Reccommendation.**

Author name : Samit Chakraborty

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, well-informed, and simple to understand, statistical techniques including auto-regression, expoential smoothing, ARIMA, and SARIMA were frequently employed to assess the sales of clothing. A technique for forecasting retail products was proposed by Demerit (2018). weekly using linear regression models in multi-processing groups with both positive and negative commodities. The introduction of dynamic pricing models to support markdown choices in multi-item group predictions has since followed. In or

prevent overfitting, grouping items in predictive models can be seen as a way of variable selection. They then exhibited regression results from multiple-item groupings on the real- world dataset provided by a clothing company in addition to the findings from the single-item regression model. They also revealed the results of markdown optimization for single items andgroups of multiple items that serve as the foundation for multi-item forecasting models. The results suggested that regression models provide better estimates in many categories than the one-item model.

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

Author name : Shaghayegh Shirkhani

Collaborative filtering, the iterative filtering process, matrix factorization, and content-based systems. Systems for collaborative filtering make product recommendations based on user similarity metrics and/or by grouping things from similar users' purchases. Despite the variety of collaborative filtering methods, many widely used systems can be distilled down to just two steps:

1. Seek out users who have similar rating tendencies to the active user (the user whom the prediction is for).
2. To establish a prediction for the active user, utilise the ratings from the users who shared your interests in step one.

3. Paper Title: Fashion Recommendation Systems

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

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

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

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

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 hat are built for various aspects is undertaken in this research. Systems created to propose clothing using various methodologies have been researched, with both their benefits and drawbacks highlighted. It has also been investigated how to make clothing detecting systems user-friendly while accepting feedback from the user.

5. **Paper Title:** **Individualized fashion recommender system**

Author name: **MSridevi, N Manikya Arun, MSheshikala and E Sudarshan**

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

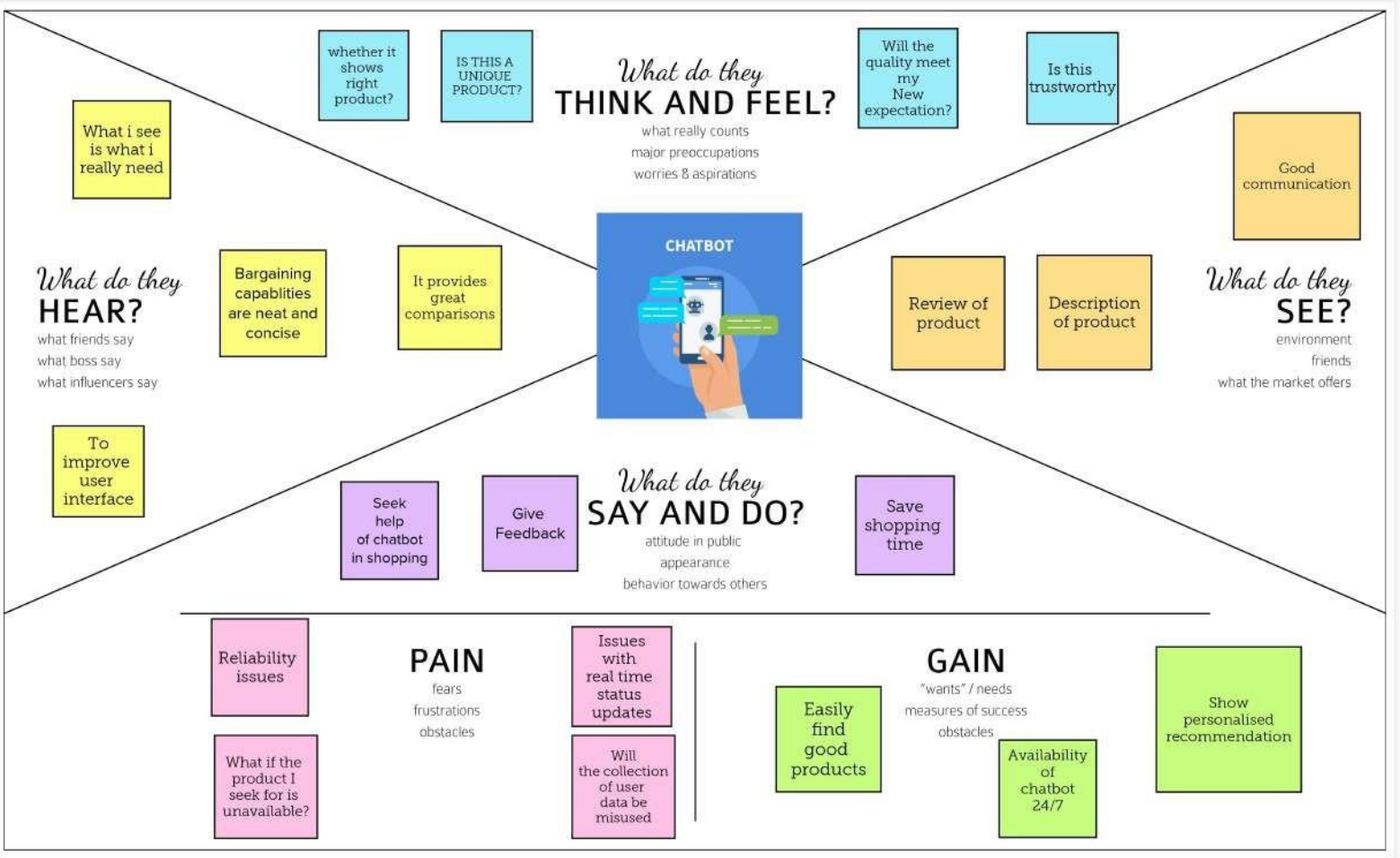
PROBLEM STATEMENT:

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, andrecommend 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 differentproducts and admin should be assigned the responsibility to create products with appropriate categories.The user should be able to mention their preferences using interacting with chat bots. The user must receive a notification on order confirmation/failure. The chat bot must gather feedback from the user at the end of order confirmation. The main objective of this application is to provide better interactivity
with the user and to reduce navigating pages to find appropriate products.

3. IDEATION & PROPOSED SOLUTION

EMPATHY MAP CANVAS:

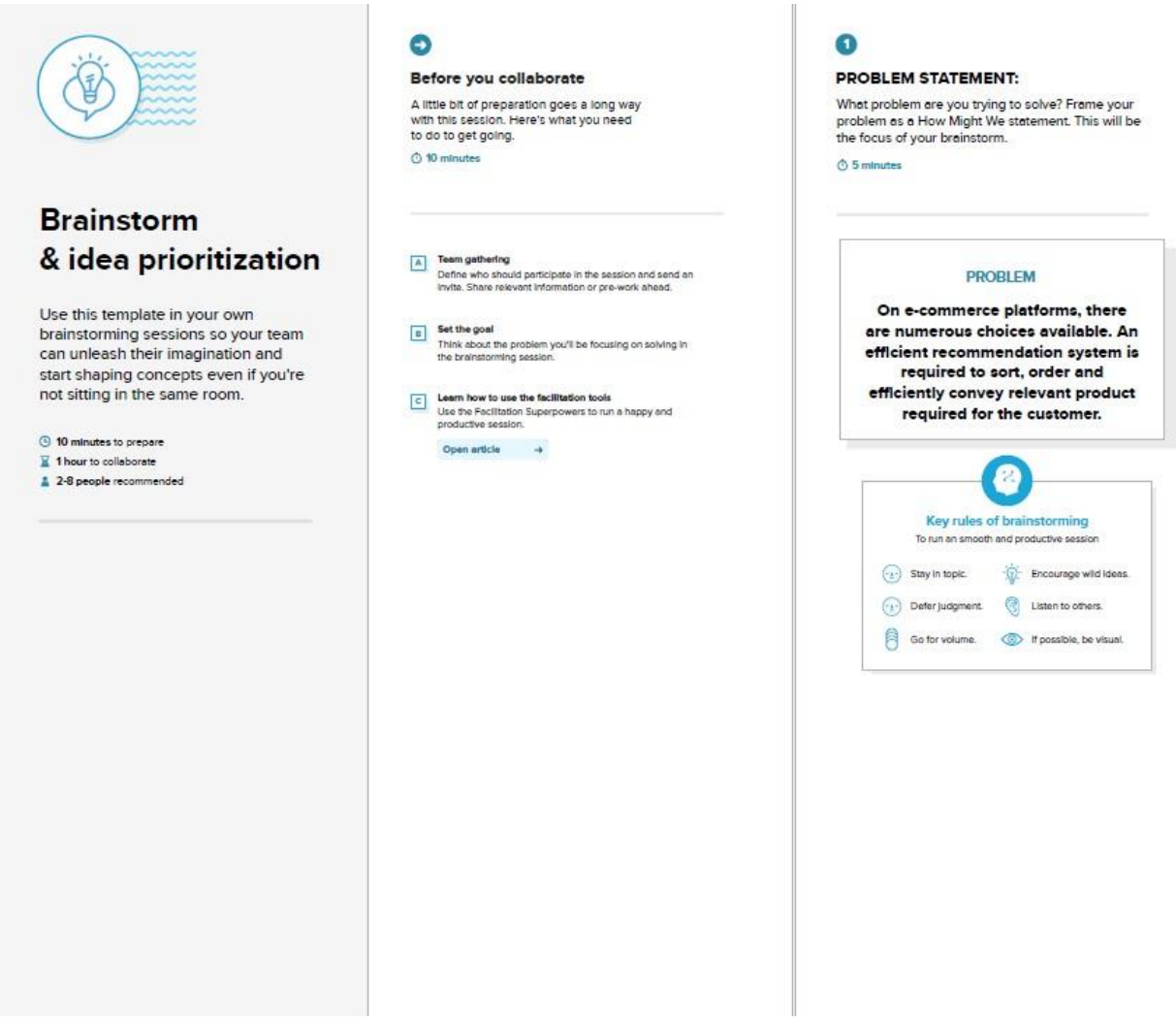
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers.



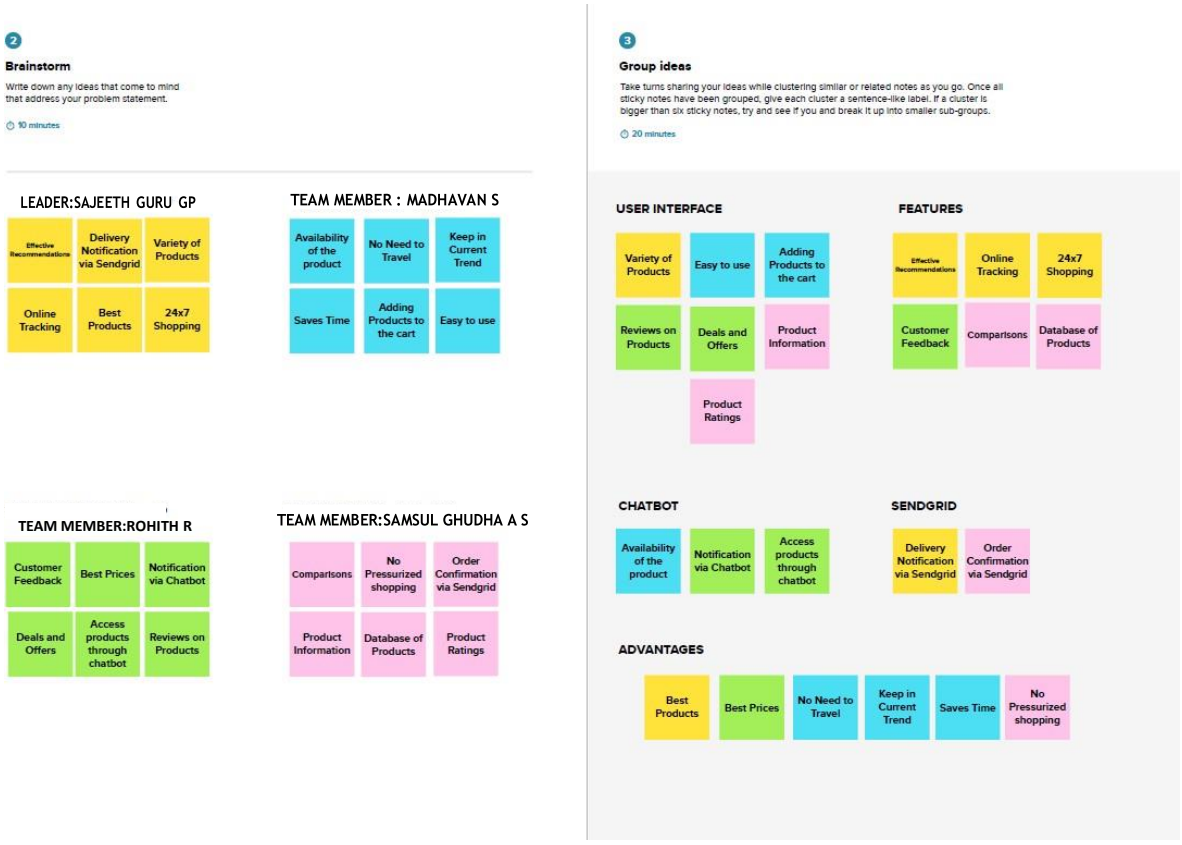
IDEATION & BRAINSTROMING:

A group problem-solving technique that involves the spontaneous contribution of ideas fromall members of the group.

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



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Customers feels difficult when Search many websites to find Fashion clothes and accessories.
2.	Idea / Solution description	Customers directly make online shopping based on their choice without any search.
3.	Novelty / Uniqueness	The customer will talk to Chat Bot regarding the Products. Get the recommendations based on information provided by the user
4.	Social Impact / Customer Satisfaction	The user friendly interface, Assistants form chat bot finding dress makes customer satisfied.
5.	Business Model (Revenue Model)	The chat bot sells our Products to customer. Customers buy our products and generate revenue
6.	Scalability of the Solution	We can easily scalable our Applications by increases the items and products

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 CONSTRAINTS — CC

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

Money and Network Connection

5. AVAILABLE SOLUTIONS — AS

Which solutions are available to the customers 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

Functional Requirements:

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

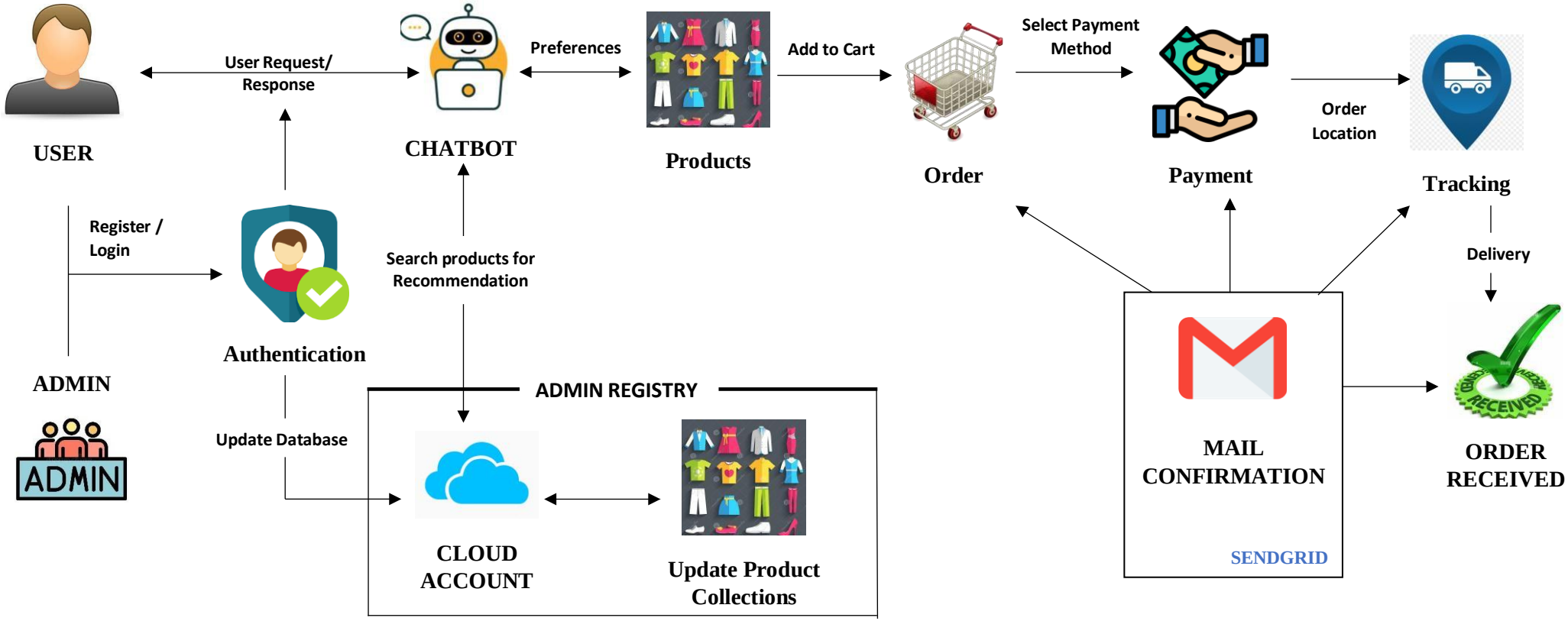
Non-functional Requirements:

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

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

5. PROJECT DESIGN

DATA FLOW DIAGRAMS (DFD LEVEL 0):



USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile/Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can see the Dashboard	Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password	I can see the Dashboard	High	Sprint-1
	Dashboard	USN-5	As a user, I can see the available products in the dashboard.	I can order the products	Medium	Sprint-2
		USN-6	As a user, I can access the Chatbot	I can interact with the Chatbot	High	Sprint-2
	Chatbot	USN-7	As a user, I provide my preferences to the Chatbot	I can order the products that I need	High	Sprint-3
	Notifications	USN-8	As a user, I can receive notifications about the status of the order.	I can access the status through email or app	High	Sprint-4
Administrator	Login	USN-1	As an admin, I can log into the administrator login portal.	I can access the admin cloud.	High	Sprint-1
	Cloud	USN-2	As an admin, I can update the products in the Cloud database.	I can manage the Cloud services	High	Sprint-2
	Order Status	USN-3	As an admin, I can confirm the order placement	I can manage the order confirmation	High	Sprint-3
		USN-4	As an admin, I can update the status of the order	I can manage the order status	High	Sprint-3
	Mail	USN-5	As an admin, I can update the status through email to the users	I can send mail to users	High	Sprint-4

TECHNICAL ARCHITECTURE:

Technical Architecture:

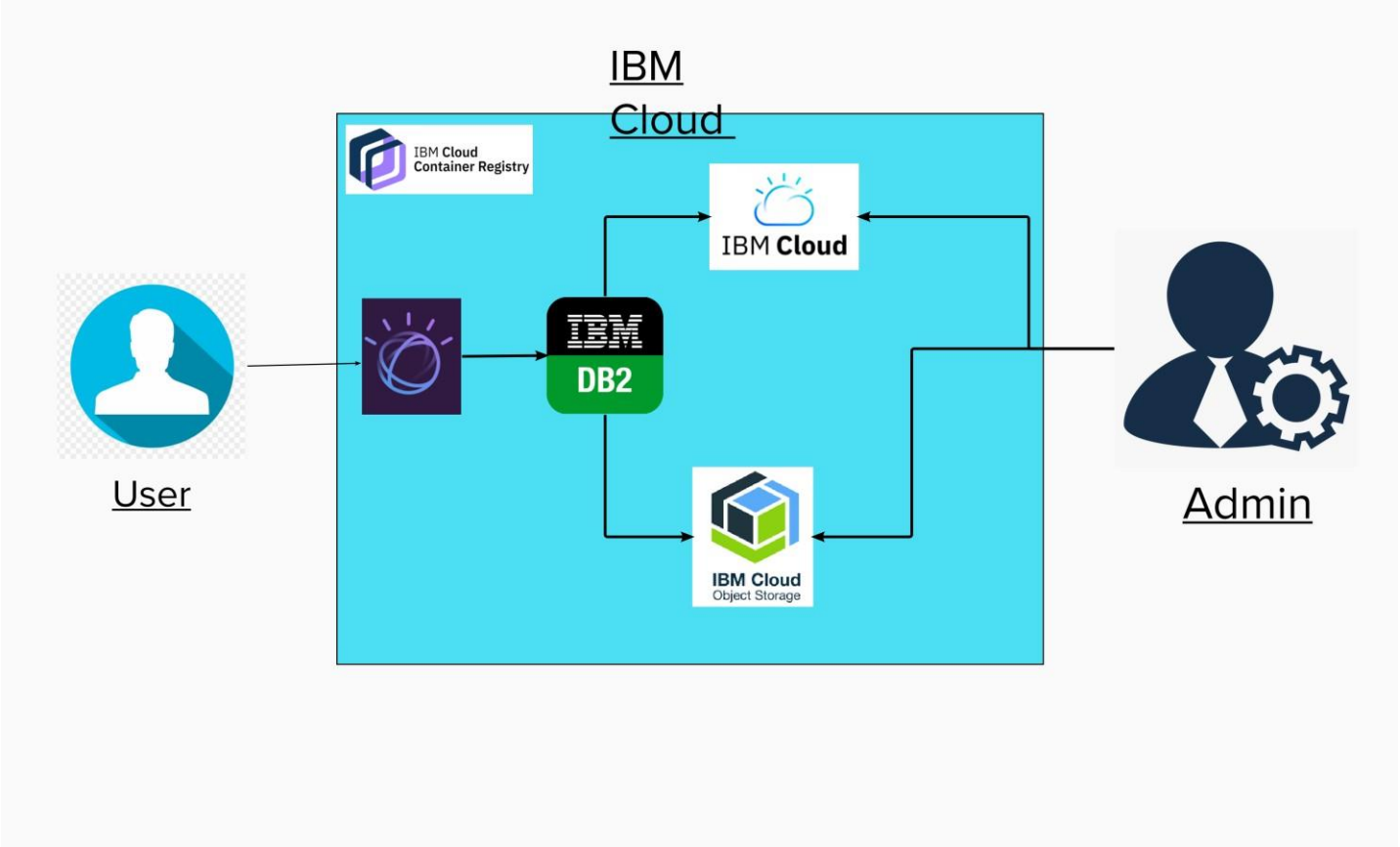


Table-1 : Components & Technologies:

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

Table-2: Application Characteristics:

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

6. PROJECT PLANNING & SCHEDULE

SPRINT PLANNING & ESTIMATION:

Remaining tasks (Milestones & Activities) to be completed

Milestones	Activities	Description
Project Development Phase	Delivery of Sprint – 1,2,3,4	To develop the code and submit the developed code by testing it
Setting up App environment	Create IBM Cloud account	Signup for an IBM Cloud account
	Create flask project	Getting started with Flask to create project
	Install IBM Cloud CLI	Install IBM Command Line Interface
	Docker CLI Installation	Installing Docker CLI on laptop
	Create an account in sendgrid	Create an account in sendgrid. Use the service as email integration to our application for sending emails
Implementing web Application	Create UI to interact with Application	Create UI • Registration page • Login page • View products page • Add products page
	Create IBM DB2 & connect with python	Create IBM DB2 service in IBM Cloud and connect with python code with DB
Integrating sendgrid service	Sendgrid integration with python	To send emails form the application we need to integrate the Sendgrid service
Developing a chatbot	Building a chatbot and Integrate to application	Build the chatbot and Integrate it to the flask application
Deploymentof App in IBM Cloud	Containerize the App	Create a docker image of your application and push it to the IBM container registry
	Upload image to IBM container registry	Upload the image to IBM container registry
	Deploy in kubernetes cluster	Once the image is uploaded to IBM Container registry deploy the image to IBM Kebernetes cluster

Finished tasks (Milestones & Activities)

Milestones	Activities	Description
Ideation Phase	Literature Survey	Literature survey on the selected project & information gathering
	Empathy Map	Prepare Empathy map to capture the user Panis & Gains, prepare list of problem statement
	Ideation	Organizing the brainstroming session and prioritise the top 3 ideas based on feasibility & Importance
Project Design Phase I	Proposed Solution	Prepare proposed solution document which includes novelty, feasibility of ideas, business model, social impact, Scalability of solution
	Problem Solution Fit	Prepare problem solution fit document
	Solution Architecture	Prepare solution architecture document
Project Design Phase	Customer Journey	Prepare customer journey map to understand the user interactions & experience with the application
	Functional requirement	Prepare functional & non functional requirement document
	Data Flow Diagram	Prepare Data Flow Diagram and user stories
	Technology architecture	Draw the technology architecture diagram
Project Planning Phase	Milestones & Activity list	Prepare milestones and activity list of the project
	Sprint Delivery Plan	Prepare sprint delivery plan

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

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.I will go through the products available on the website	20	High	Sajeeth Guru G P Madhavan S Rohith R Samsul Ghudha A S
Sprint-2	Admin	USN-2	As an Admin, I can check out the database about the stock and have a track of all the things that the users are purchasing.	20	High	Sajeeth Guru G P Madhavan S Rohith R Samsul Ghudha A S
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	Sajeeth Guru G P Madhavan S Rohith R Samsul Ghudha A S
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	Sajeeth Guru G P Madhavan S Rohith R Samsul Ghudha A S

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

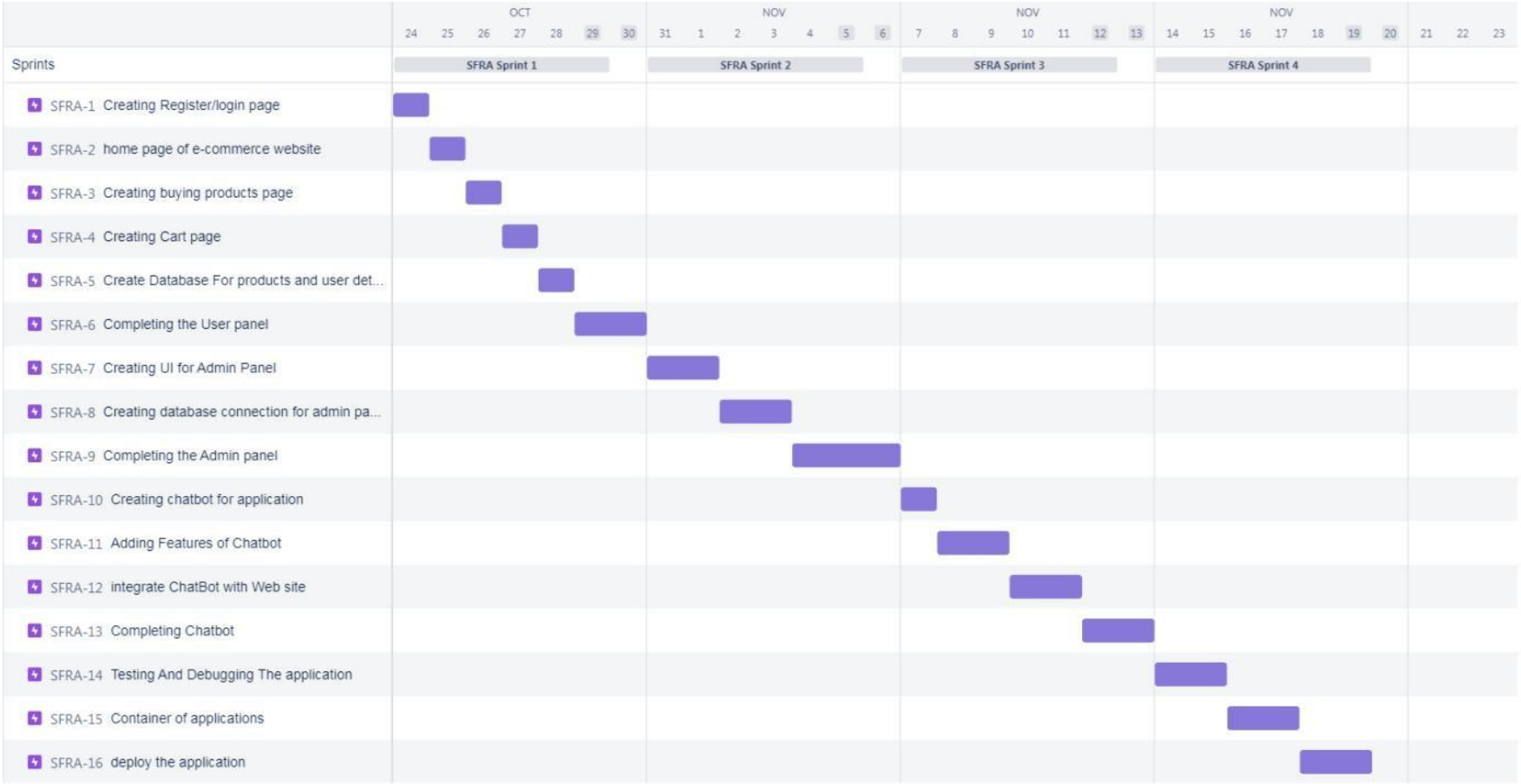
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022		29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12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

Velocity:

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

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

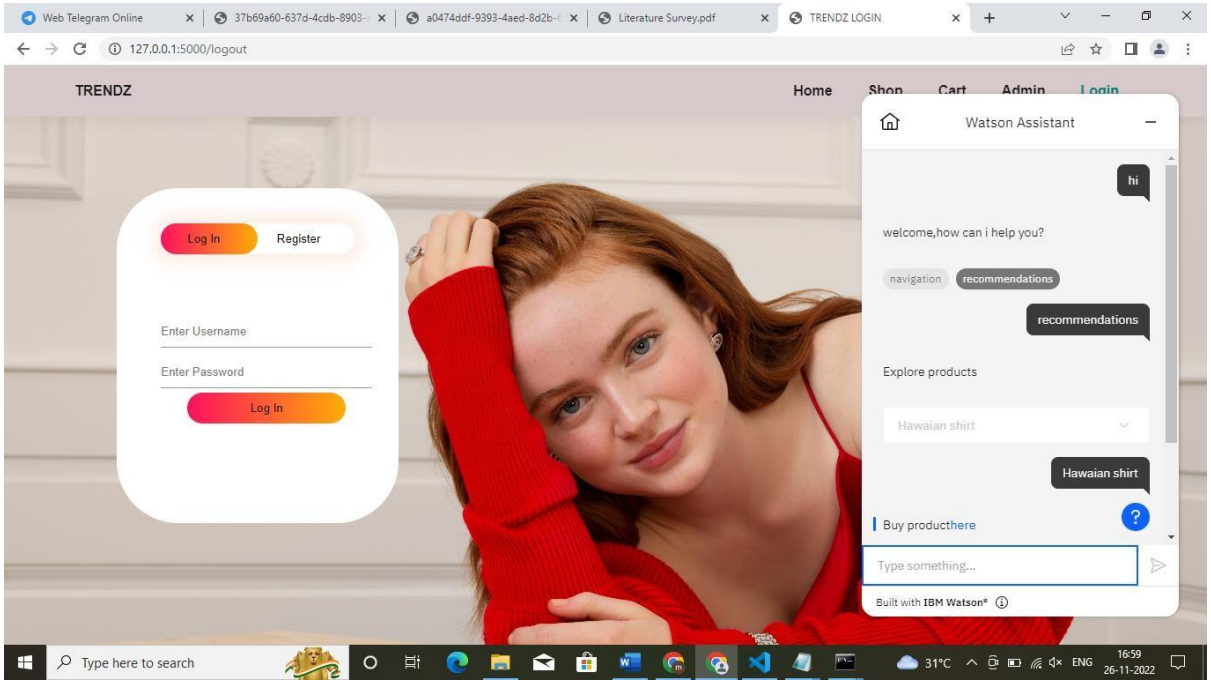
Burndown Chart:



7. CODING & SOLUTIONING

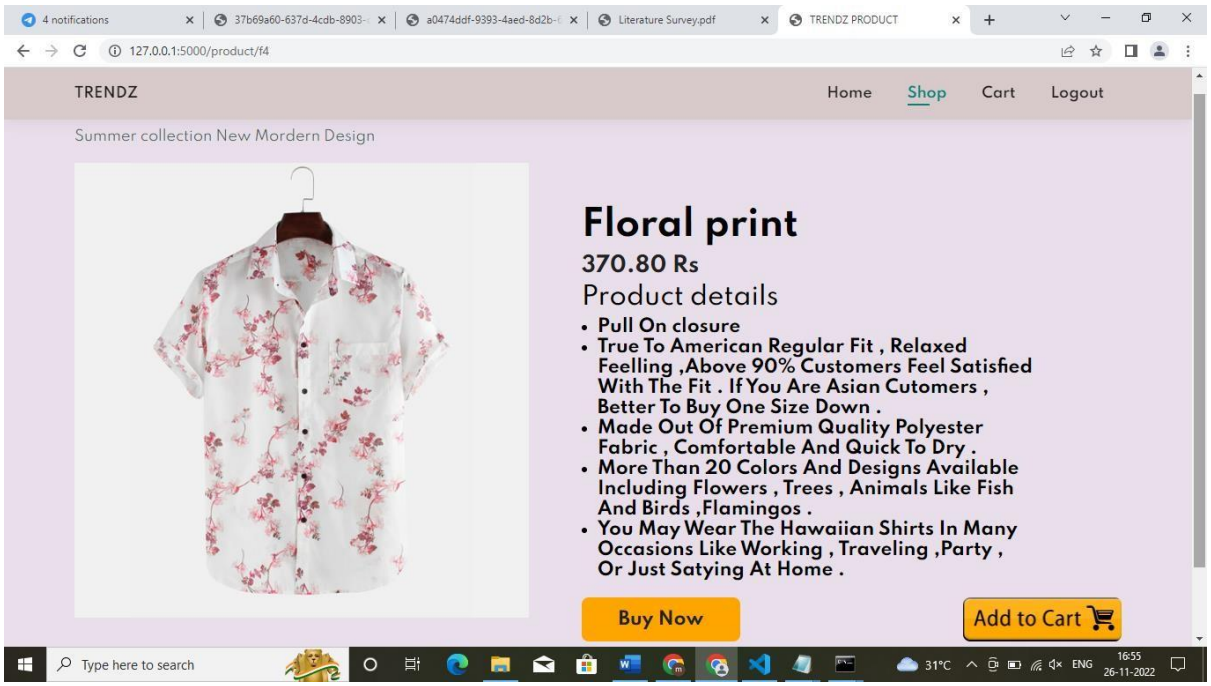
7.1 FEATURE 1:

Using chat bot we can manage user's choices and orders.

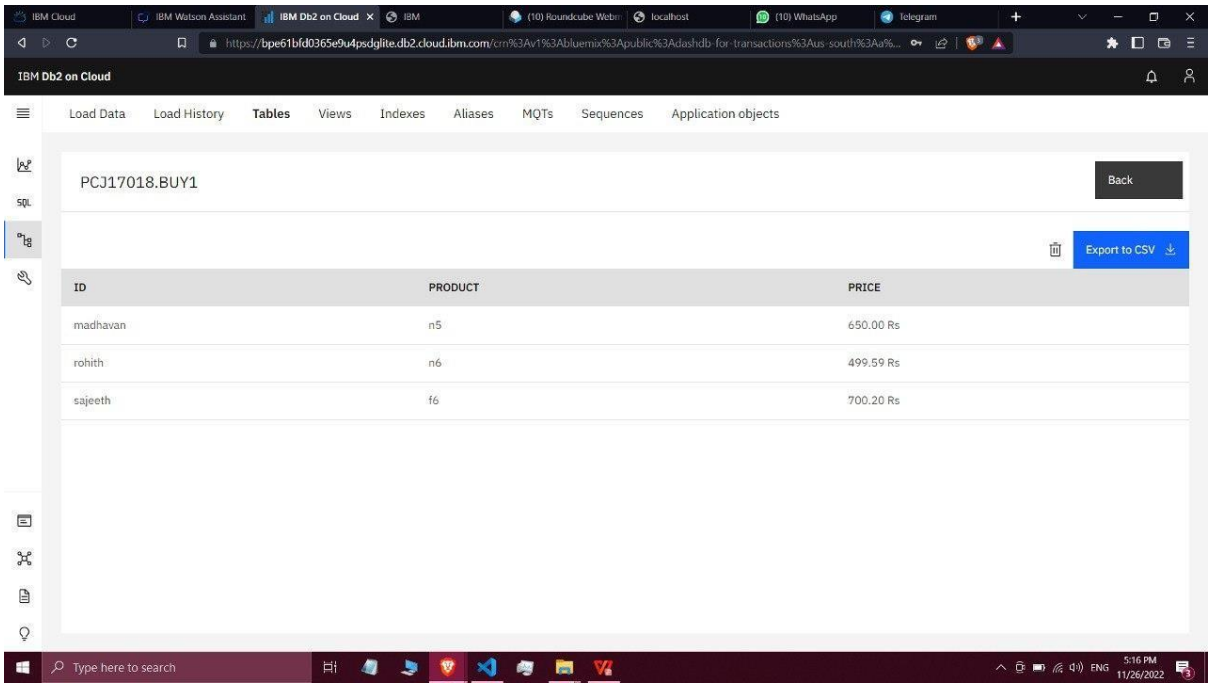


7.2 FEATURE 2 :

Chat Bot promote the best deals and offers on that day.



7.3 DATABASE:



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	7	0	0	7
Register	7	0	0	7
Home Page	7	0	0	7
Product Page	6	0	0	6
Cart Page	5	0	0	5
Admin Page	2	0	0	2
Final Report Output	4	0	0	4

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 Smart Fashion 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 they were resolved

Resolution	**Severity1**	**Severity2**	**Severity3**	**Severity4**
By Design	5	5	2	3
Duplicate	1	0	3	0
External	2	3	0	1
Fixed	11	2	4	20
Skipped	0	0	1	1
Won't Fix	0	5	2	1
Totals	24	14	12	26

9.RESULT:

9.1 PERFORMANCE TESTING:

NFT - Risk Assessment									
S.No	Project Name	Scope/ feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification
1	Smart Fashion Recommender Application	New	Low	No Changes	Moderate	Unable to Signup/register new user	> 5 to 10%	ORANGE	Customer may be provided incorrect details
2	Smart Fashion Recommender Application	Existing	No Changes	No Changes	Low	Couldn't change existing details	< 5%	GREEN	Given wrong details about the customer
3	Smart Fashion Recommender Application	New	Moderate	No Changes	Moderate	Unable to modify user	< 10%	ORANGE	Duplication of user arised
4	Smart Fashion Recommender Application	New	Low	No Changes	No Changes	Unable to update products stock	< 5%	GREEN	Wrong product details given

NFT - Detailed Test Plan				
S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	Smart Fashion Recommender Application	Incorrect username	Not registered	Login Unsuccessful
2	Smart Fashion Recommender Application	Repeating same details	User updation	Login successful
3	Smart Fashion Recommender Application	Incorrect user details	Login validation	Login Unsuccessful
4	Smart Fashion Recommender Application	No product updations	Database updation	Product details should be given

End Of Test Report								
S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	User	New User	Yes	Registration successful	GO	Validation login details	Closed	Success
2	Products	Checking Updation	Yes	Updation incomplete	GO	None	Closed	Success
3	Admin Page	Manage products	Yes	Product update incomplete	NO-GO	Update product details	Open	Success

9. ADVANTAGES & DISADVANTAGES:

ADVANTAGES:

- **Smart fashion recommender application is the user friendly.**
- **With the help of chatbot user can find the products very easily.**
- **This application used to discover the product based on the user's choice, very easily and quickly.**

DISADVANTAGES:

- **It needs active internet connection.**
- **Privacy concerns.**

10. CONCLUSION:

The Fashion Recommendation System is mainly used to recommend the best possible outfit combinations to a user who has no fashion sense based on their wardrobe . Also another reason is that fashion ishighly dependent on the time period. However the system does a great job in inculcating a fashion sense among the users and can provide the best recommendations based on the user's wardrobe. Since the system is implemented as a website, it is very easy for the end users to access as well as use. The scope of this system can be expanded by including the ability to detect the various design and patterns on clothing, and to increase the number of occasions.

11. 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 collaborative and content-based filtering techniques.

12. APPENDIX

SOURCE CODE

Python File (SFRA.PY)

```
from flask import Flask,render_template,request,redirect,url_for,session
import ibm_db
import re
from flask_mail import Mail,Message

app = Flask(_name_)
app.config['MAIL_SERVER']='mail.smartinternz.com'
app.config['MAIL_PORT']=465
app.config['MAIL_USERNAME'] = '810019106068@smartinternz.com'
app.config['MAIL_PASSWORD'] = 'PNTIBMCk77'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL']=True
mail = Mail(app)

app.secret_key='a'

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=125f9f61-9715-46f9-9399-
c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30426;SECURITY=
SSL;SSLServerCertificate=Cert.crt;UID=pcj17018;PWD=MPcY3wjqbL7MbLBm",'','')



@app.route('/')
def home():
    session['loggedin']=False
    return render_template('login.html')
```

```
@app.route('/login',methods=['GET','POST'])
def login():
    global userid
    msg=''

    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM users WHERE username=? AND password=?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if(account):
            session['loggedin']=True
            session['id']=account['USERNAME']
            userid = account['USERNAME']
            session['USERNAME']=account['USERNAME']
            msg='Welcome %s!'%userid
            email=account['EMAIL']
            session['EMAIL']=email

msg1=Message("Confirmation",sender='810019106068@smartinternz.com',recipients=[em
ail])
            msg1.body="you have logged in to TRENDZ! Enjoy Shopping."
            mail.send(msg1)
            return render_template('home.html',msg=msg)
        else:
```

```
            msg="Incorrect username/password"
    return render_template('login.html',msg=msg)



@app.route('/register',methods=['GET','POST'])
def register():
    msg=''
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM users WHERE username=?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg='Account already exists!'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+',email):
            msg='Invalid email address!'
        elif not re.match(r'[A-Za-z0-9]+',username):
            msg='Name must contain only characters and numbers!'
        else:
            insert_sql ="INSERT INTO users 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,password)
            ibm_db.execute(prep_stmt)
            msg='you have successfully registered!'

msg1=Message("Confirmation",sender='810019106068@smartinternz.com',recipients=[em
ail])
            msg1.body="Hurray! You have Successfully registered"
            mail.send(msg1)
    elif request.method == 'POST':
        msg = 'Please fill out the form'
    return render_template('register.html',msg=msg)


@app.route('/alogin',methods=['GET','POST'])
def alogin():
    global adminid
    msg=''

    if request.method == 'POST':
        aname = request.form['aname']
        apassword = request.form['apassword']
        sql = "SELECT * FROM ADMIN WHERE NAME=? AND PASSWORD=?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,aname)
        ibm_db.bind_param(stmt,2,apassword)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if(account):
            session['aloggedin']=True
            session['aid']=account['NAME']
```

```
            adminid = account['NAME']
            session['NAME']=account['NAME']
            msg='Welcome %s!'%adminid
            return render_template('admin.html',msg=msg)
        else:
            msg="Incorrect username/password"
    return render_template('adminlogin.html',msg=msg)



@app.route('/areg',methods=['GET','POST'])
def aregister():
    msg=''
    if request.method == 'POST':
        username = request.form['aname']
        email = request.form['aemail']
        password = request.form['apassword']
        sql = "SELECT * FROM ADMIN WHERE NAME=?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg='Account already exists!'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+',email):
            msg='Invalid email address!'
        elif not re.match(r'[A-Za-z0-9]+',username):
            msg='Name must contain only characters and numbers!'
```

```
        else:
            insert_sql ="INSERT INTO admin 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,password)
            ibm_db.execute(prep_stmt)
            msg='you have successfully registered!'
    elif request.method == 'POST':
        msg = 'Please fill out the form'
    return render_template('adminreg.html',msg=msg)

@app.route('/home')
def homepage():
    if session['loggedin']==True:
        msg='Welcome %s!'%userid
        return render_template('home.html',msg=msg)
    else:
        msg='Please Login!'
        return render_template('login.html',msg=msg)

@app.route('/shop')
def shop():
    if session['loggedin']==True:
        return render_template('shop.html')
    else:
        msg="Please Login!"
        return render_template('login.html',msg=msg)
```

```
@app.route('/product/<p1>')
def product(p1):
    img = 'https://sajee.s3.ap.cloud-object-storage.appdomain.cloud/products/%s.jpg' %p1
    print(img)
    product=p1
    sql="SELECT * FROM PRODUCT WHERE NAME =?"
    stmt=ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,product)
    ibm_db.execute(stmt)
    pro_sql="SELECT PRICE,PRONAME FROM PRODUCT WHERE NAME =?"
    pro_stmt=ibm_db.prepare(conn,pro_sql)
    ibm_db.bind_param(pro_stmt,1,product)
    ibm_db.execute(pro_stmt)
    tuple=ibm_db.fetch_tuple(pro_stmt)
    print(tuple)
    price=tuple[0]
    name=tuple[1]
    return
render_template('product.html',img=img,price=price,name=name,product=product)


@app.route('/admin')
def admin():
    if session['aloggedin']==True:
        msg='Welcome %s!'%adminid
        return render_template('admin.html',msg=msg)
    else:
        msg='Please Login!'
        return render_template('adminlogin.html',msg=msg)


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

```
def logout():
    session['loggedin']=False
    session.pop('id',None)
    session.pop('USERNAME',None)
    return render_template('login.html')

@app.route('/alogout')
def alogout():
    session['aloggedin']=False
    session.pop('aid',None)
    session.pop('NAME',None)
    return render_template('adminlogin.html')

@app.route('/cart')
def cart():
    if session['loggedin']==True:
        sql_c="SELECT COUNT(NAME) FROM CART1"
        stmt_c=ibm_db.prepare(conn,sql_c)
        ibm_db.execute(stmt_c)
        count1=ibm_db.fetch_tuple(stmt_c)
        print(count1[0])
        count=count1[0]
        sql_name="SELECT NAME FROM CART1"
        stmt_name=ibm_db.prepare(conn,sql_name)
        ibm_db.execute(stmt_name)
        name=ibm_db.fetch_assoc(stmt_name)
        print(name)

        sql_price="SELECT PRICE FROM CART1"
```

```
        stmt_price=ibm_db.prepare(conn,sql_price)
        ibm_db.execute(stmt_price)
        price=ibm_db.fetch_both(stmt_price)
        print(price)

        sql_pro="SELECT PRONAME FROM CART1"
        stmt_pro=ibm_db.prepare(conn,sql_pro)
        ibm_db.execute(stmt_pro)
        proname=ibm_db.fetch_tuple(stmt_pro)
        print(proname)

        sql_img="SELECT LINK FROM CART1"
        stmt_img=ibm_db.prepare(conn,sql_img)
        ibm_db.execute(stmt_img)
        img=ibm_db.fetch_tuple(stmt_img)

        return
render_template('cart.html',count=count,img=img,proname=proname,price=price,product=
name)
    else:
        msg='Please Login!'
        return render_template('login.html',msg=msg)

@app.route('/addcart/<product>')
def add_cart(product):
    p=product
    sql="SELECT NAME,PRICE,LINK,PRONAME FROM PRODUCT WHERE NAME=?"
    stmt=ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,p)
    ibm_db.execute(stmt)
```

```
    cart=ibm_db.fetch_tuple(stmt)
    name=cart[0]
    price=cart[1]
    link=cart[2]
    proname=cart[3]
    cart_sql="INSERT INTO CART1 values(?,?,?,?)"
    cart_stmt=ibm_db.prepare(conn,cart_sql)
    ibm_db.bind_param(cart_stmt,1,name)
    ibm_db.bind_param(cart_stmt,2,price)
    ibm_db.bind_param(cart_stmt,3,link)
    ibm_db.bind_param(cart_stmt,4,proname)
    ibm_db.execute(cart_stmt)
    msg='successfully added to cart!'
    return render_template('home.html',msg=msg)

@app.route('/buy/<product>')
def buy(product):
    p=product
    sql="SELECT NAME,PRICE,PRONAME FROM PRODUCT WHERE NAME=?"
    stmt=ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,p)
    ibm_db.execute(stmt)
    cart=ibm_db.fetch_tuple(stmt)
    name=cart[0]
    price=cart[1]
    id=session['id']
    pro=cart[2]
    email=session['EMAIL']
    buy_sql="INSERT INTO BUY1 values(?,?,?)"
```

```
    buy_stmt=ibm_db.prepare(conn,buy_sql)
    ibm_db.bind_param(buy_stmt,1,id)
    ibm_db.bind_param(buy_stmt,2,name)
    ibm_db.bind_param(buy_stmt,3,price)
    ibm_db.execute(buy_stmt)
    msg='Successfully Purchased!'
    msg1=Message("Purchase
Confirmation",sender='810019106068@smartinternz.com',recipients=[email])
    msg1.body="You Have Purchased the product:%s Successfully!"%pro
    mail.send(msg1)


    return render_template('home.html',msg=msg)


if__name__== '_main_':
    app.debug = True
    app.run(host='0.0.0.0')
```

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>TRENDZ</title>
    <link rel="stylesheet" type="text/css" href="../static/css/style.css">
</head>
<body>
    <section id="header">
        <a href="#" id="logolink"><div id="logo">TRENDZ</div></a>
```

```
      <div id="nav">
        <ul id="navbar">
          <li class="homenav"><a class="active" href="/home">HOME</a></li>
          <li class="shopnav"><a href="/shop">Shop</a></li>
          <li class="cartnav"><a href="/cart">Cart</a></li>
          <li class="logoutnav"><a href="/logout">Logout</a></li>
        </ul>
      </div>
    </section>
    <section id="hero">
      <p>{{msg}}</p>
      <h4>Trade-in-offer</h4>
      <h2>Super value deals</h2>
      <h1>On all Products</h1>
      <p>Save more with coupons & up to 70% off!</p>
      <button type="button" class="shop-btn"><a href="/shop">Shop Now</a></button>
    </section>
    <script>
      window.watsonAssistantChatOptions = {
        integrationID: "54de7925-7fe7-4be7-bd6d-5ba9c722bb1d", // The ID of this
integration.
        region: "au-syd", // The region your integration is hosted in.
        serviceInstanceID: "10b2ff7d-d1de-4719-ba8c-4564cdc072ce", // 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>
```

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>TRENDZ LOGIN</title>
  <link rel="stylesheet" type="text/css" href="../static/css/style.css">
  <style>
    *{
      margin: 0;
      padding: 0;
      font-family: sans-serif;
    }
    .hero{
      height: 100%;
      width: 100%;
      position: absolute;
    }
    .form-box{
      width: 320px;
      height: 380px;
      position: relative;
```

```
    margin: 6% 4%;
    background-color: #fff;
    padding: 5px;
    overflow: hidden;
    border-radius: 70px;
}
.button-box{
    width: 220px;
    margin: 35px auto;
    position: relative;
    box-shadow: 0 0 20px 9px #ff61241f;
    border-radius: 30px;
}
.toggle-btn{
    padding: 10px 30px;
    cursor: pointer;
    background: transparent;
    border: 0;
    outline: none;
    position: relative;
}
#btn{
    top: 0;
    left: 0;
    position: absolute;
    width: 110px;
    height: 100%;
    background: linear-gradient(to right, #ff105f,#ffad06);
    border-radius: 30px;
```

```
    transition: .5s;
}
.input-group{
    top: 140px;
    align-items: center;
    position: absolute;
    width: 280px;
    transition: .5s;
    padding-right: 40px;
}
.input-field{
    width: 100%;
    padding: 10px 0;
    margin: 5px 0;
    border-left: 0;
    border-top: 0;
    border-right: 0;
    border-bottom: 1px solid #999;
    background: transparent;
}
.submit-btn{
    width: 75%;
    padding: 10px 30px;
    cursor: pointer;
    display: block;
    margin: auto;
    background: linear-gradient(to right,#ff105f,#ffad06);
    border: 0;
    outline: none;
```

```
        border-radius: 30px;
    }
    #login{
        left:50px;
    }

  </style>
</head>
<body>
  <section id="header">
    <a href="#" id="logolink"><div id="logo">TRENDZ</div></a>
    <div id="nav">
      <ul id="navbar">
        <li class="homenav"><a href="/home">Home</a></li>
        <li class="shopnav"><a href="/shop">Shop</a></li>
        <li class="cartnav"><a href="/cart">Cart</a></li>
        <li class="adminnav"><a href="/alogin">Admin</a></li>
        <li class="loginnav"><a class="active" href="/login">Login</a></li>
      </ul>
    </div>
  </section>
  <section id="hero">
    <div class="form-box">
          <div class="button-box">
            <div id="btn"></div>
            <a href="/login" style="font-size: 13px;"><button type="button" class="toggle-btn" >Log In</button></a>
            <a href="/register" style="font-size: 13px;"><button type="button" class="toggle-btn" >Register</button></a>
          </div>
```

```
            <div id="msg">{{msg}}</div>
            <form id="login" action="/login" method="post" class="input-group">
              <input type="text" name="username" class="input-field" placeholder="Enter Username" required>
              <input type="password" name="password" class="input-field" placeholder="Enter Password" required>
              <button type="submit" class="submit-btn">Log In</button>
            </form>
      </div>
   </section>
   <script>
      window.watsonAssistantChatOptions = {
       integrationID: "54de7925-7fe7-4be7-bd6d-5ba9c722bb1d", // The ID of this integration.
       region: "au-syd", // The region your integration is hosted in.
       serviceInstanceID: "10b2ff7d-d1de-4719-ba8c-4564cdc072ce", // 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>
   <!--<script>
      var x=document.getElementById("login");
      var y=document.getElementById("register");
      var z=document.getElementById("btn");
```

```
        function register(){
            x.style.left = "-400px";
            y.style.left = "50px";
            z.style.left = "110px";
        }
        function login(){
            x.style.left = "50px";
            y.style.left = "450px";
            z.style.left = "0px";
        }
    </script>-->
</body>
</html>
```

Admin.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>Admin</title>
    <link rel="stylesheet" type="text/css" href="../static/css/style.css">
    <style>
        #body{
            padding: 50px;
        }
        p{
            font-size:45px;
```

```
    font-weight:400;
}
#buy table{
    width: 100%;
    border-collapse: collapse;
    table-layout: fixed;
    white-space: nowrap;
}
#buy table img{
    width: 70px;
}
#buy table td:nth-child(1){
    width: 100px;
    text-align: center;
}
#buy table td:nth-child(2){
    width: 150px;
    text-align: center;
}
#buy table td:nth-child(3){
    width: 200px;
    text-align: center;
}
#buy table td:nth-child(4){
    width: 250px;
    text-align: center;
}
#buy table thead{
    border: 1px solid #e2e2e1;
```

```
            border-left: none;
            border-right: none;
        }
        #buy table thead tr td{
            font-size: 13px;
            font-weight: 700;
            text-transform: uppercase;
            padding: 20px 0;
        }
        #buy table tbody tr td{
            padding-top: 15px;
        }
    </style>
</head>
<body>
    <section id="header">
        <a href="#" id="logolink"><div id="logo">TRENDZ</div></a>
        <div id="nav">
            <ul id="navbar">
                <li class="homenav"><a href="/home">HOME</a></li>
                <li class="shopnav"><a href="/shop">Shop</a></li>
                <li class="cartnav"><a href="/cart">Cart</a></li>
                <li class="Aregnav"><a href="/login">User</a></li>
                <li class="adminnav"><a class="active" href="#">Admin</a></li>
                <li class="alogout"><a href="/alogout">Logout</a></li>
            </ul>
        </div>
    </section>
    <section id="body">
```

```
    <p>{{msg}}</p>
  </section>
  <section id="buy" class="section-p1">
    <table width="100%">
      <thead>
        <tr>
          <td>user</td>
          <td>product</td>
          <td>product-id</td>
          <td>Price</td>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td><span>madhavan</span></td>
          <td><img src="https://sajee.s3.ap.cloud-object-
storage.appdomain.cloud/products/n5.jpg" alt=""></td>
          <td><span>n5</span></td>
          <td><span>650.00 Rs</span></td>
        </tr>
        <tr>
          <td><span>rohith</span></td>
          <td><img src="https://sajee.s3.ap.cloud-object-
storage.appdomain.cloud/products/n6.jpg" alt=""></td>
          <td><span>n6</span></td>
          <td><span>499.59 Rs</span></td>
        </tr>
        <tr>
          <td><span>sajeeth</span></td>
```

```
                <td><img src="https://sajee.s3.ap.cloud-object-
storage.appdomain.cloud/products/f6.jpg" alt=""></td>
                <td><span>f6</span></td>
                <td><span>370.80 Rs</span></td>
            </tr>



        </tbody>
    </table>
  </section>
</body>
</html>
```

Adminlogin.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>Admin Login</title>
  <link rel="stylesheet" type="text/css" href="../static/css/style.css">
  <style>
    *{
      margin: 0;
      padding: 0;
      font-family: sans-serif;
    }
    .hero{
      height: 100%;
```

```
    width: 100%;
    position: absolute;
}
.form-box{
    width: 320px;
    height: 380px;
    position: relative;
    margin: 6% 4%;
    background-color: #fff;
    padding: 5px;
    overflow: hidden;
    border-radius: 70px;
}
.button-box{
    width: 220px;
    margin: 35px auto;
    position: relative;
    box-shadow: 0 0 20px 9px #ff61241f;
    border-radius: 30px;
}
.toggle-btn{
    padding: 10px 30px;
    cursor: pointer;
    background: transparent;
    border: 0;
    outline: none;
    position: relative;
}
#btn{
```

```
    top: 0;
    left: 0;
    position: absolute;
    width: 110px;
    height: 100%;
    background: linear-gradient(to right, #ff105f,#ffad06);
    border-radius: 30px;
    transition: .5s;
}
.input-group{
    top: 140px;
    align-items: center;
    position: absolute;
    width: 280px;
    transition: .5s;
    padding-right: 40px;
}
.input-field{
    width: 100%;
    padding: 10px 0;
    margin: 5px 0;
    border-left: 0;
    border-top: 0;
    border-right: 0;
    border-bottom: 1px solid #999;
    background: transparent;
}
.submit-btn{
    width: 75%;
```

```
        padding: 10px 30px;
        cursor: pointer;
        display: block;
        margin: auto;
        background: linear-gradient(to right,#ff105f,#ffad06);
        border: 0;
        outline: none;
        border-radius: 30px;
      }
      #login{
        left:50px;
      }

  </style>
</head>
<body>
  <section id="header">
    <a href="#" id="logolink"><div id="logo">TRENDZ</div></a>
    <div id="nav">
      <ul id="navbar">
        <li class="homenav"><a href="/home">HOME</a></li>
        <li class="shopnav"><a href="/shop">Shop</a></li>
        <li class="cartnav"><a href="/cart">Cart</a></li>
        <li class="adminnav"><a class="active" href="/alogin">Admin</a></li>
        <li class="Aregnav"><a href="/login">User</a></li>
      </ul>
    </div>
  </section>
  <section id="hero">
```

```
    <div class="form-box">
        <div class="button-box">
          <div id="btn"></div>
          <a href="/alogin" style="font-size: 13px;"><button type="button" class="toggle-btn" >Log In</button></a>
          <a href="/areg" style="font-size: 13px;"><button type="button" class="toggle-btn" >Register</button></a>
        </div>
        <div id="msg">{{msg}}</div>
        <form id="login" action="/alogin" method="post" class="input-group">
          <input type="text" name="aname" class="input-field" placeholder="Enter Username" required>
          <input type="password" name="apassword" class="input-field" placeholder="Enter Password" required>
          <button type="submit" class="submit-btn">Log In</button>
        </form>
    </div>
  </section>
</body>
</html>
```

Adminreg.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>Admin Register</title>
  <link rel="stylesheet" type="text/css" href="../static/css/style.css">
```

```
<style>
    *{
        margin: 0;
        padding: 0;
        font-family: sans-serif;
    }
    .hero{
        height: 100%;
        width: 100%;
        position: absolute;
    }
    .form-box{
        width: 320px;
        height: 380px;
        position: relative;
        margin: 6% 4%;
        background-color: #fff;
        padding: 5px;
        overflow: hidden;
        border-radius: 70px;
    }
    .button-box{
        width: 220px;
        margin: 35px auto;
        position: relative;
        box-shadow: 0 0 20px 9px #ff61241f;
        border-radius: 30px;
    }
    .toggle-btn{
```

```
    padding: 10px 30px;
    cursor: pointer;
    background: transparent;
    border: 0;
    outline: none;
    position: relative;
}
#btn{
    top: 0;
    right: 0;
    position: absolute;
    width: 110px;
    height: 100%;
    background: linear-gradient(to right, #ff105f,#ffad06);
    border-radius: 30px;
    transition: .5s;
}
.input-group{
    top: 120px;
    align-items: center;
    position: absolute;
    width: 280px;
    transition: .5s;
    padding-right: 40px;
}
.input-field{
    width: 100%;
    padding: 10px 0;
    margin: 5px 0;
```

```
            border-left: 0;
            border-top: 0;
            border-right: 0;
            border-bottom: 1px solid #999;
            background: transparent;
        }
        .submit-btn{
            width: 75%;
            padding: 10px 30px;
            cursor: pointer;
            display: block;
            margin: auto;
            background: linear-gradient(to right,#ff105f,#ffad06);
            border: 0;
            outline: none;
            border-radius: 30px;
        }
        #register{
            left:50px;
        }
    </style>
</head>
<body>
    <section id="header">
        <a href="#" id="logolink"><div id="logo">TRENDZ</div></a>
        <div id="nav">
            <ul id="navbar">
                <li class="homenav"><a href="/home">Home</a></li>
                <li class="shopnav"><a href="/shop">Shop</a></li>
```

```
            <li class="cartnav"><a href="/cart">Cart</a></li>
            <li class="adminnav"><a class="active" href="/alogin">ADMIN</a></li>
            <li class="registernav"><a href="/login">User</a></li>
          </ul>
      </div>
    </section>
    <section id="hero">
      <div class="form-box">
            <div class="button-box">
              <div id="btn"></div>
              <a href="/alogin" style="font-size: 13px;"><button type="button" class="toggle-
btn" >Log In</button></a>
              <a href="/areg" style="font-size: 13px;"><button type="button" class="toggle-
btn" >Register</button></a>
            </div>
            <div id="msg">{{msg}}</div>
            <form id="register" action="/areg" method="post" class="input-group">
              <input type="text" name="aname" class="input-field" placeholder="Enter
Username" required>
              <input type="email" name="aemail" class="input-field" placeholder="Enter Email
Id" required>
              <input type="password" name="apassword" class="input-field"
placeholder="Enter Password" required>
              <button type="submit" class="submit-btn">Register</button>
            </form>
      </div>
    </section>
</body>
</html>
```

Reg.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>TRENDZ REGISTER</title>
    <link rel="stylesheet" type="text/css" href="../static/css/style.css">
    <style>
        *{
            margin: 0;
            padding: 0;
            font-family: sans-serif;
        }
        .hero{
            height: 100%;
            width: 100%;
            position: absolute;
        }
        .form-box{
            width: 320px;
            height: 380px;
            position: relative;
            margin: 6% 4%;
            background-color: #fff;
            padding: 5px;
            overflow: hidden;
            border-radius: 70px;
        }
        .button-box{
```

```
    width: 220px;
    margin: 35px auto;
    position: relative;
    box-shadow: 0 0 20px 9px #ff61241f;
    border-radius: 30px;
}
.toggle-btn{
    padding: 10px 30px;
    cursor: pointer;
    background: transparent;
    border: 0;
    outline: none;
    position: relative;
}
#btn{
    top: 0;
    right: 0;
    position: absolute;
    width: 110px;
    height: 100%;
    background: linear-gradient(to right, #ff105f,#ffad06);
    border-radius: 30px;
    transition: .5s;
}
.input-group{
    top: 120px;
    align-items: center;
    position: absolute;
    width: 280px;
```

```
            transition: .5s;
            padding-right: 40px;
        }
        .input-field{
            width: 100%;
            padding: 10px 0;
            margin: 5px 0;
            border-left: 0;
            border-top: 0;
            border-right: 0;
            border-bottom: 1px solid #999;
            background: transparent;
        }
        .submit-btn{
            width: 75%;
            padding: 10px 30px;
            cursor: pointer;
            display: block;
            margin: auto;
            background: linear-gradient(to right,#ff105f,#ffad06);
            border: 0;
            outline: none;
            border-radius: 30px;
        }
        #register{
            left:50px;
        }
    </style>
</head>
```

```
<body>
  <section id="header">
    <a href="#" id="logolink"><div id="logo">TRENDZ</div></a>
    <div id="nav">
      <ul id="navbar">
        <li class="homenav"><a href="/home">Home</a></li>
        <li class="shopnav"><a href="/shop">Shop</a></li>
        <li class="cartnav"><a href="/cart">Cart</a></li>
        <li class="adminnav"><a href="/alogin">Admin</a></li>
        <li class="registernav"><a class="active" href="/register">Register</a></li>
      </ul>
    </div>
  </section>
  <section id="hero">
    <div class="form-box">
          <div class="button-box">
            <div id="btn"></div>
            <a href="/login" style="font-size: 13px;"><button type="button" class="toggle-
btn" >Log In</button></a>
            <a href="/register" style="font-size: 13px;"><button type="button"
class="toggle-btn" >Register</button></a>
          </div>
          <div id="msg">{{msg}}</div>
          <form id="register" action="/register" method="post" class="input-group">
            <input type="text" name="username" class="input-field" placeholder="Enter
Username" required>
            <input type="email" name="email" class="input-field" placeholder="Enter
Email Id" required>
            <input type="password" name="password" class="input-field"
placeholder="Enter Password" required>
            <button type="submit" class="submit-btn">Register</button>
```

```
            </form>
        </div>
    </section>
    <script>
        window.watsonAssistantChatOptions = {
         integrationID: "54de7925-7fe7-4be7-bd6d-5ba9c722bb1d", // The ID of this integration.
         region: "au-syd", // The region your integration is hosted in.
         serviceInstanceID: "10b2ff7d-d1de-4719-ba8c-4564cdc072ce", // 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>
    <!--<script>
        var x=document.getElementById("login");
        var y=document.getElementById("register");
        var z=document.getElementById("btn");

        function register(){
            x.style.left = "-400px";
            y.style.left = "50px";
            z.style.left = "110px";
        }
        function login(){
```

```
        x.style.left = "50px";
        y.style.left = "450px";
        z.style.left = "0px";
      }
  </script>-->
</body>
</html>
```

Product.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>TRENDZ PRODUCT</title>
  <link rel="stylesheet" type="text/css" href="../static/css/style.css">

</head>
<body style="background-color: rgb(233, 223, 234);">
  <section id="header">
    <a href="#" id="logolink"><div id="logo">TRENDZ</div></a>
    <div id="nav">
      <ul id="navbar">
        <li class="homenav"><a href="/home">Home</a></li>
        <li class="shopnav"><a class="active" href="/shop">Shop</a></li>
        <li class="cartnav"><a href="/cart">Cart</a></li>
        <li class="logoutnav"><a href="/logout">Logout</a></li>
      </ul>
    </div>
```

```
</section>
<section id="product2" class="section-p1">
    <p style="padding-top: 30px;">Summer collection New Mordern Design</p>
    <div class="procontainer">
        <div class="image">
            <img src={{img}} alt="">
        </div>
        <div class="details">
            <h5 id="name2">{{name}}</h5>
            <h4 id="price2">{{price}}</h4>
            <h3 id="prod2"> Product details</h3>
            <h5 id="det2">
                <ul>
                    <li>Pull On closure</li>
                    <li>True To American Regular Fit , Relaxed Feelling ,Above 90% Customers Feel Satisfied With The Fit . If You Are Asian Cutomers , Better To Buy One Size Down .</li>
                    <li>Made Out Of Premium Quality Polyester Fabric , Comfortable And Quick To Dry .</li>
                    <li>More Than 20 Colors And Designs Available Including Flowers , Trees , Animals Like Fish And Birds ,Flamingos .</li>
                    <li>You May Wear The Hawaiian Shirts In Many Occasions Like Working , Traveling ,Party , Or Just Satying At Home .</li>
                </ul>
            </h5>
            <br>
            <div class="buttons">
            <a href="/buy/{{product}}" class="buy"><button type="button" id="buybutton"><h4>Buy Now</h4></button></a>
            <a href="/addcart/{{product}}" class="addcart"><button type="button" id="cartbutton"></button></a>
            </div>
```

```
            </div>
        </div>
    </section>
</body>
</html>
```

Shop.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>TRENDZ SHOP</title>
    <style>
        .pro,a{
            width: 15%;
        }
    </style>
    <link rel="stylesheet" type="text/css" href="../static/css/style.css">
</head>
<body style="background-color: rgb(233, 223, 234);">
    <section id="header">
        <a href="#" id="logolink"><div id="logo">TRENDZ</div></a>
        <div id="nav">
            <ul id="navbar">
                <li class="homenav"><a href="/home">Home</a></li>
                <li class="shopnav"><a class="active" href="/shop">Shop</a></li>
                <li class="cartnav"><a href="/cart">Cart</a></li>
                <li class="loginnav"><a href="/alogin">Admin</a></li>
```

```
            <li class="logoutnav"><a href="/logout">Logout</a></li>
          </ul>
        </div>
    </section>
    <section id="product1" class="section-p1">
      <p style="padding-top: 30px;">Summer collection New Mordern Design</p>
      <div class="pro-container" id="p">
        <h3 class="pro-container">Short Sleeve Mens Hawaiian Shirts</h3><br>
        <a href="/product/f1"><div class="pro">
          <img src="https://sajee.s3.ap.cloud-object-
storage.appdomain.cloud/products/f1.jpg" alt="">
          <div class="des">
            <span>adidas</span>
            <h5>Cartoon Shirts</h5>
            <h4>499.70 Rs</h4>
          </div>
        </div></a>
        <a href="/product/f2"><div class="pro">
          <img src="https://sajee.s3.ap.cloud-object-
storage.appdomain.cloud/products/f2.jpg" alt="">
          <div class="des">
            <span>adidas</span>
            <h5>King Kameha</h5>
            <h4>459.50 Rs</h4>
          </div>
        </div></a>
        <a href="/product/f3"><div class="pro">
          <img src="https://sajee.s3.ap.cloud-object-
storage.appdomain.cloud/products/f3.jpg" alt="">
          <div class="des">
```

```
            <span>adidas</span>
            <h5>Natural</h5>
            <h4>750.45 Rs</h4>
          </div>
        </div></a>
        <a href="/product/f4"><div class="pro">
          <img src="https://sajee.s3.ap.cloud-object-storage.appdomain.cloud/products/f4.jpg" alt="">
          <div class="des">
            <span>adidas</span>
            <h5>Floral print</h5>
            <h4>370.80 Rs</h4>
          </div>
        </div></a>
        <a href="/product/f5"><div class="pro">
          <img src="https://sajee.s3.ap.cloud-object-storage.appdomain.cloud/products/f5.jpg" alt="">
          <div class="des">
            <span>adidas</span>
            <h5>Slim fit</h5>
            <h4>560.00 Rs</h4>
          </div>
        </div></a>
        <h3 class="pro-container">Women Wear</h3><br>

        <a href="/product/f8"><div class="pro">
          <img src="https://sajee.s3.ap.cloud-object-storage.appdomain.cloud/products/f8.jpg" alt="">
          <div class="des">
            <span>adidas</span>
```

```
            <h5>Tops</h5>
            <h4>280.40 Rs</h4>
          </div>
        </div></a>
        <a href="/product/f7"><div class="pro">
          <img src="https://sajee.s3.ap.cloud-object-storage.appdomain.cloud/products/f7.jpg" alt="">
          <div class="des">
            <span>adidas</span>
            <h5>Casual pant</h5>
            <h4>430.40 Rs</h4>
          </div>
        </div></a>
        <h3 class="pro-container">Formal Shirts</h3><br>
        <a href="/product/n1"><div class="pro">
          <img src="https://sajee.s3.ap.cloud-object-storage.appdomain.cloud/products/n1.jpg" alt="">
          <div class="des">
            <span>adidas</span>
            <h5>Collarless shirt</h5>
            <h4>560.00 Rs</h4>
          </div>
        </div></a>
        <a href="/product/n2"><div class="pro">
          <img src="https://sajee.s3.ap.cloud-object-storage.appdomain.cloud/products/n2.jpg" alt="">
          <div class="des">
            <span>adidas</span>
            <h5>Semi formal shirt</h5>
            <h4>459.50 Rs</h4>
```

```
        </div>
      </div></a>
      <a href="/product/n3"><div class="pro">
        <img src="https://sajee.s3.ap.cloud-object-storage.appdomain.cloud/products/n3.jpg" alt="">
        <div class="des">
          <span>adidas</span>
          <h5>Formal shirt</h5>
          <h4>399.70 Rs</h4>
        </div>
      </div></a>
      <h3 class="pro-container">Casual Shirts</h3><br>
      <a href="/product/n4"><div class="pro">
        <img src="https://sajee.s3.ap.cloud-object-storage.appdomain.cloud/products/n4.jpg" alt="">
        <div class="des">
          <span>adidas</span>
          <h5>Printed Casual Shirt</h5>
          <h4>430.55 Rs</h4>
        </div>
      </div></a>
      <a href="/product/f6"><div class="pro">
        <img src="https://sajee.s3.ap.cloud-object-storage.appdomain.cloud/products/f6.jpg" alt="">
        <div class="des">
          <span>adidas</span>
          <h5>Casual Shirt</h5>
          <h4>700.20 Rs</h4>
        </div>
      </div></a>
```

```
        <h3 class="pro-container">Denim Shirts</h3><br>
        <a href="/product/n5"><div class="pro">
            <img src="https://sajee.s3.ap.cloud-object-
storage.appdomain.cloud/products/n5.jpg" alt="">
            <div class="des">
                <span>adidas</span>
                <h5>Denim shirt</h5>
                <h4>650.00 Rs</h4>
            </div>
        </div></a>
        <a href="/product/n7"><div class="pro">
            <img src="https://sajee.s3.ap.cloud-object-
storage.appdomain.cloud/products/n7.jpg" alt="">
            <div class="des">
                <span>adidas</span>
                <h5>Over Shirt</h5>
                <h4>650.99 Rs</h4>
            </div>
        </div></a>
        <a href="/product/n8"><div class="pro">
            <img src="https://sajee.s3.ap.cloud-object-
storage.appdomain.cloud/products/n8.jpg" alt="">
            <div class="des">
                <span>adidas</span>
                <h5>Linen Shirt</h5>
                <h4>560.89 Rs</h4>
            </div>
        </div></a>
        <h3 class="pro-container">Trousers</h3><br>
        <a href="/product/n6"><div class="pro">
```

```
            <img src="https://sajee.s3.ap.cloud-object-
storage.appdomain.cloud/products/n6.jpg" alt="">
            <div class="des">
               <span>adidas</span>
               <h5>Trouser</h5>
               <h4>499.59 Rs</h4>
            </div>
         </div></a>
      </div>
   </section>
</body>
</html>
```

Cart.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>CART</title>
   <link rel="stylesheet" type="text/css" href="../static/css/style.css">
   <style>
      #cart{
         padding-top: 6%;
      }
      #cart table{
         width: 100%;
         border-collapse: collapse;
         table-layout: fixed;
```

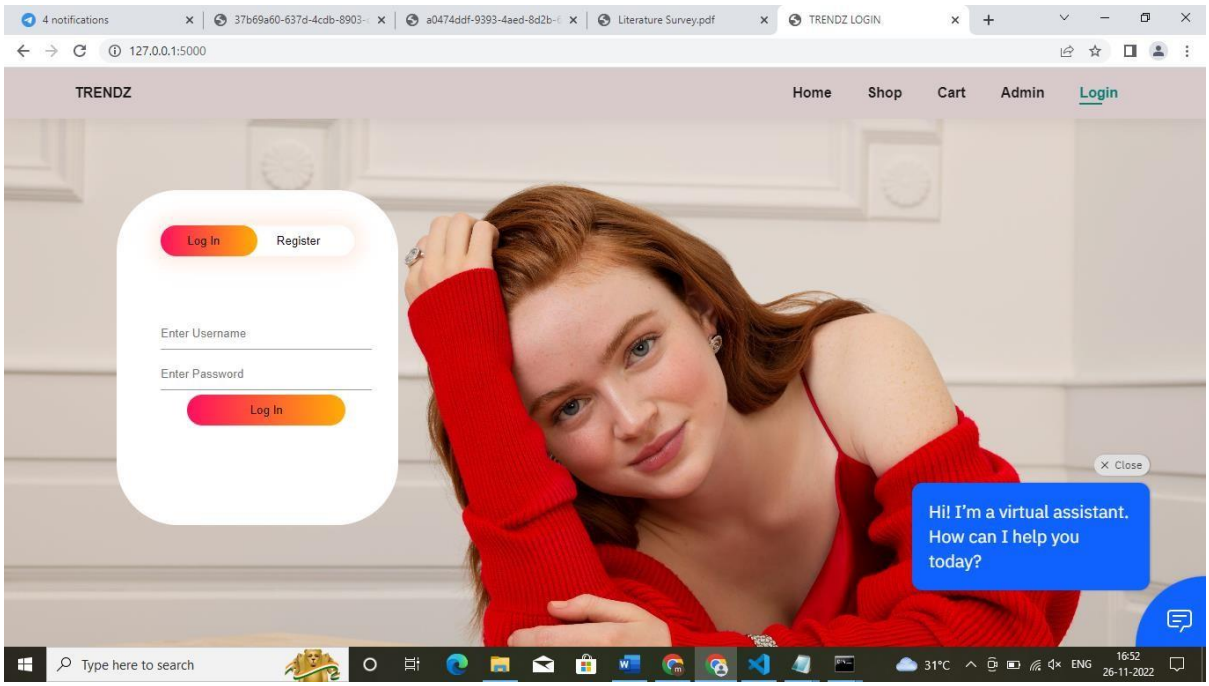
```
    white-space: nowrap;
}
#cart table img{
    width: 70px;
}
#cart table td:nth-child(1){
    width: 100px;
    text-align: center;
}
#cart table td:nth-child(2){
    width: 150px;
    text-align: center;
}
#cart table td:nth-child(3){
    width: 200px;
    text-align: center;
}
#cart table td:nth-child(4){
    width: 250px;
    text-align: center;
}
#cart table thead{
    border: 1px solid #e2e2e1;
    border-left: none;
    border-right: none;
}
#cart table thead tr td{
    font-size: 13px;
    font-weight: 700;
```

```
        text-transform: uppercase;
        padding: 20px 0;
      }
      #cart table tbody tr td{
        padding-top: 15px;
      }
    </style>
</head>
<body>
    <section id="header">
      <a href="#" id="logolink"><div id="logo">TRENDZ</div></a>
      <div id="nav">
        <ul id="navbar">
          <li class="homenav"><a href="/home">Home</a></li>
          <li class="shopnav"><a  href="/shop">Shop</a></li>
          <li class="cartnav"><a class="active" href="/cart">Cart</a></li>
          <li class="logoutnav"><a href="/logout">Logout</a></li>
        </ul>
      </div>
    </section>
    <section id="cart" class="section-p1">
      <table width="100%">
        <thead>
          <tr>
            <td>Remove</td>
            <td>Image</td>
            <td>Product</td>
            <td>Price</td>
            <td>Buy</td>
```

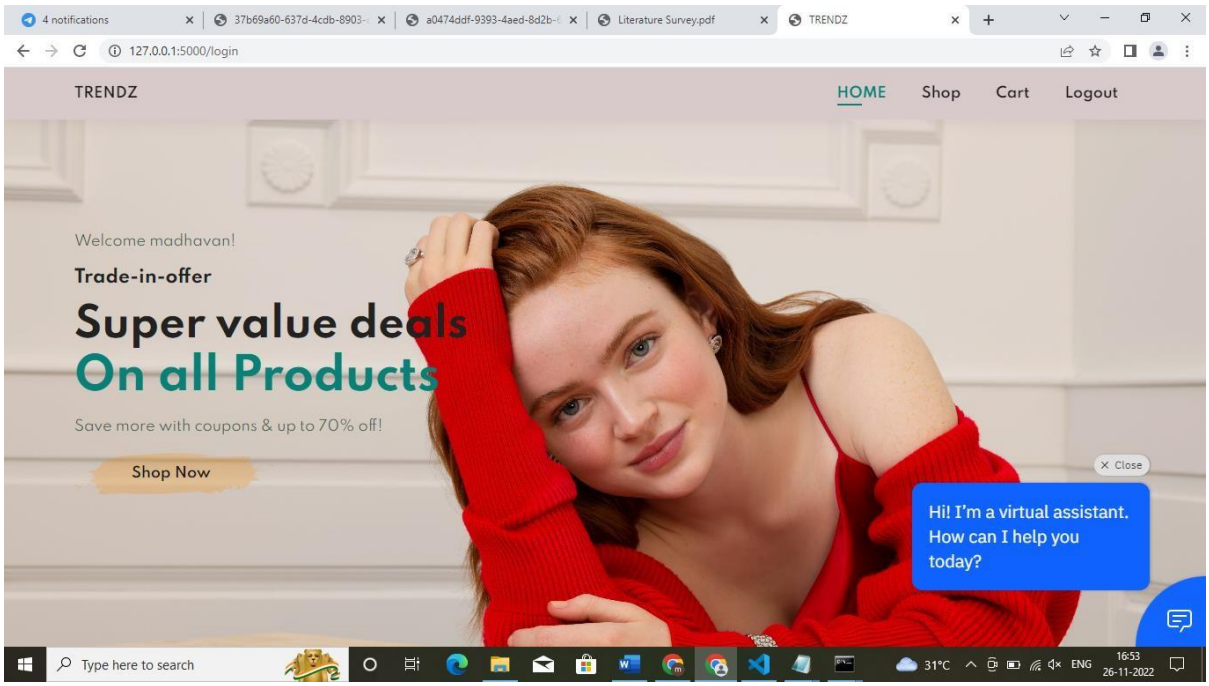
```
            </tr>
          </thead>
          <tbody>
            <tr>
                <td><a href="/remove/f5"><button style="background-color:transparent;
border:0"><img src="../static/img/delete.png" style="width: 25px; cursor:
pointer;"></button></a></td>
                <td><img src="https://sajee.s3.ap.cloud-object-
storage.appdomain.cloud/products/f5.jpg"></td>
                <td><span>Slim fit</span></td>
                <td><span>560.00 Rs</span></td>
                <td><a href="/buy/f5" class="buy"><button type="button"
id="buybutton"><h4>Buy Now</h4></button></a></td>
            </tr>
            </tr>
            <tr>
                <td><a href="/remove/f6"><button style="background-color:transparent;
border:0"><img src="../static/img/delete.png" style="width: 25px; cursor:
pointer;"></button></a></td>
                <td><img src="https://sajee.s3.ap.cloud-object-
storage.appdomain.cloud/products/f6.jpg"></td>
                <td><span>Casual Shirt</span></td>
                <td><span>700.20 Rs</span></td>
                <td><a href="/buy/f6" class="buy"><button type="button"
id="buybutton"><h4>Buy Now</h4></button></a></td>
            </tr>
          </tbody>
        </table>
    </section>
</body>
</html>
```

SCREENSHOTS:

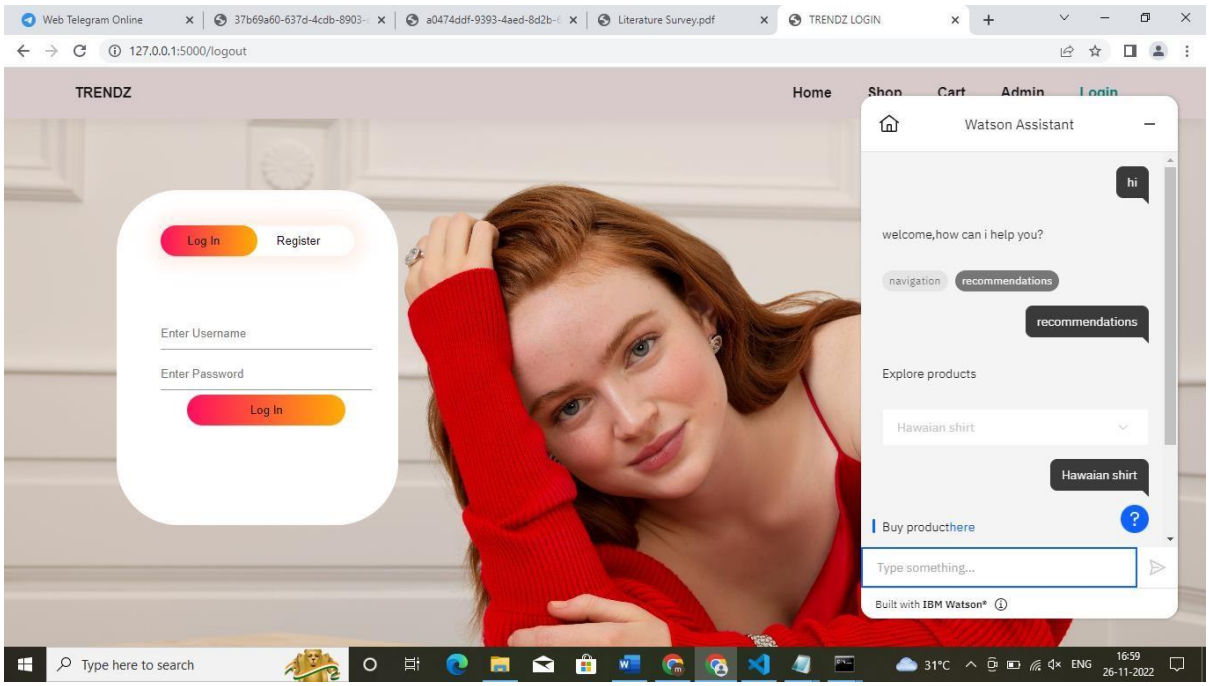
Login Page:



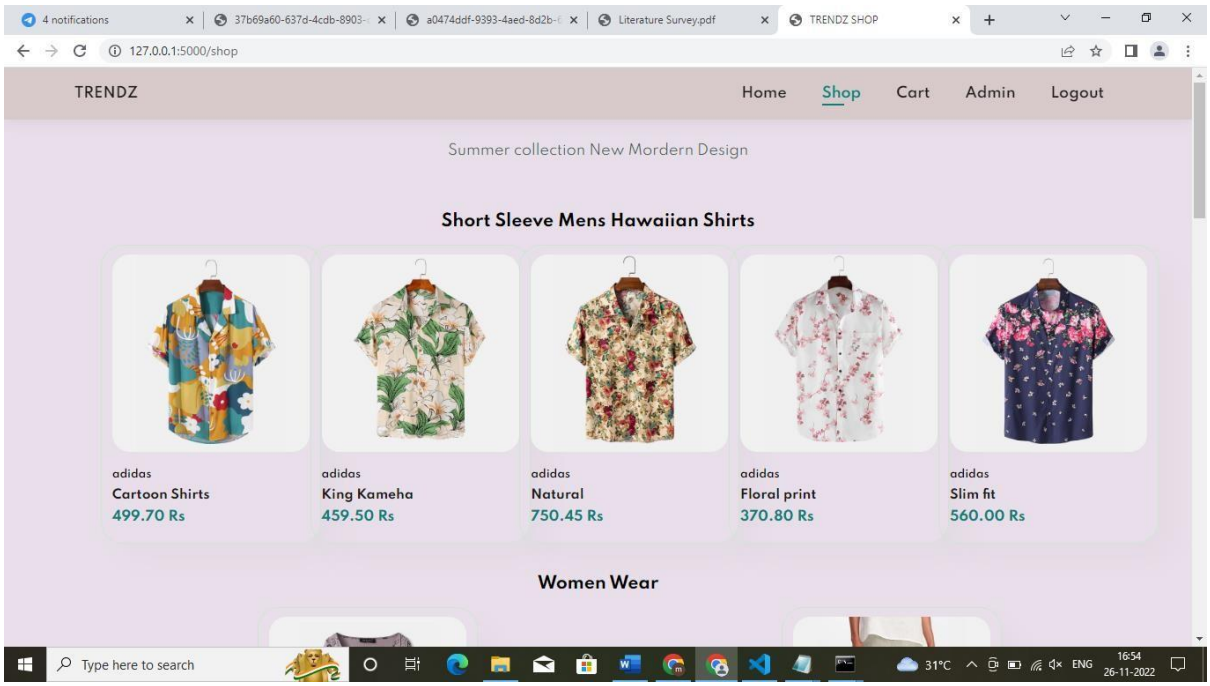
Home Page:

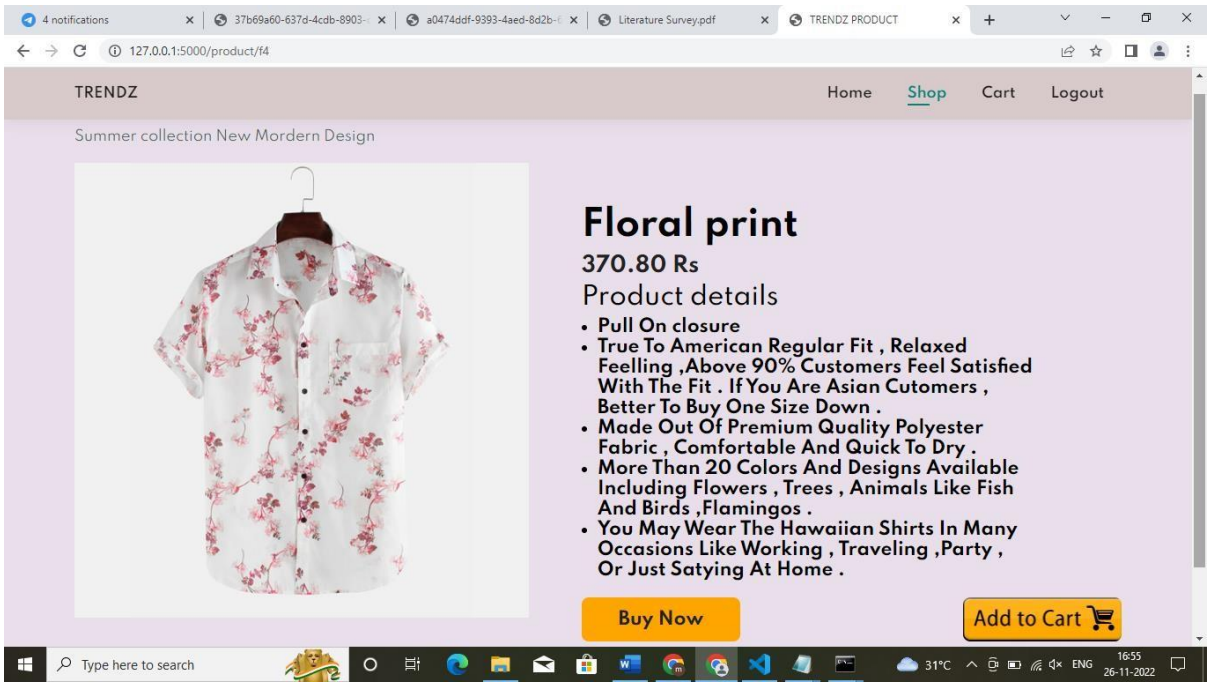


Integrate Chat Bot:

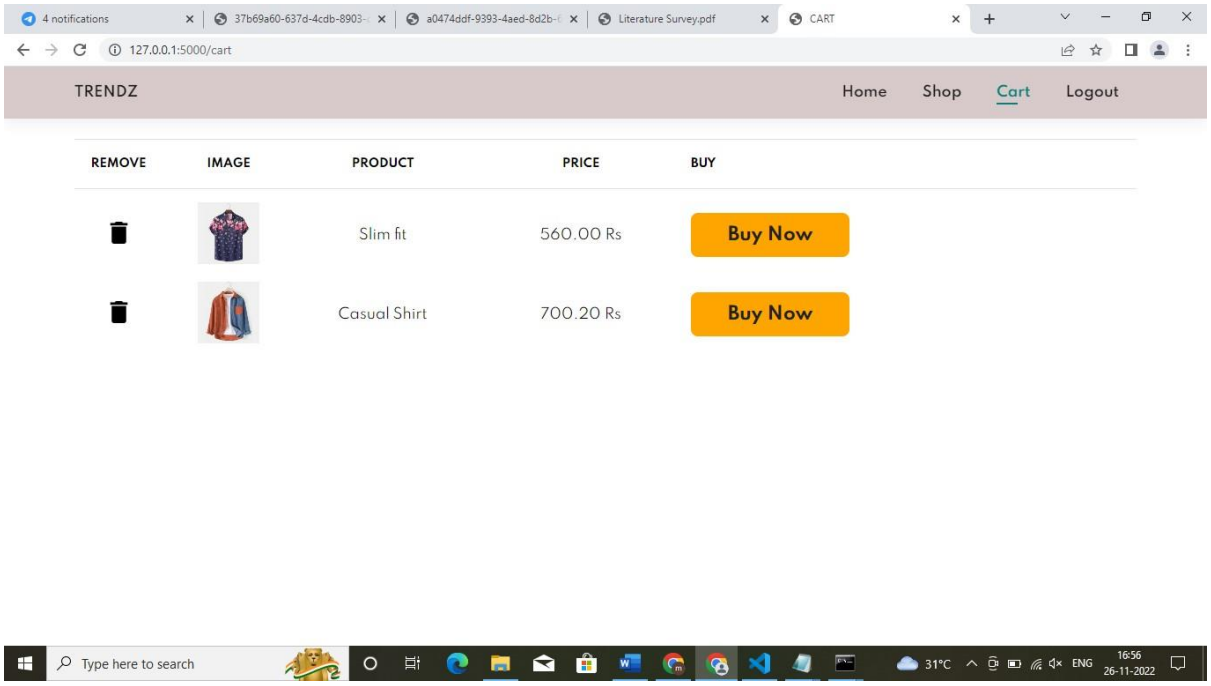


Product Page:

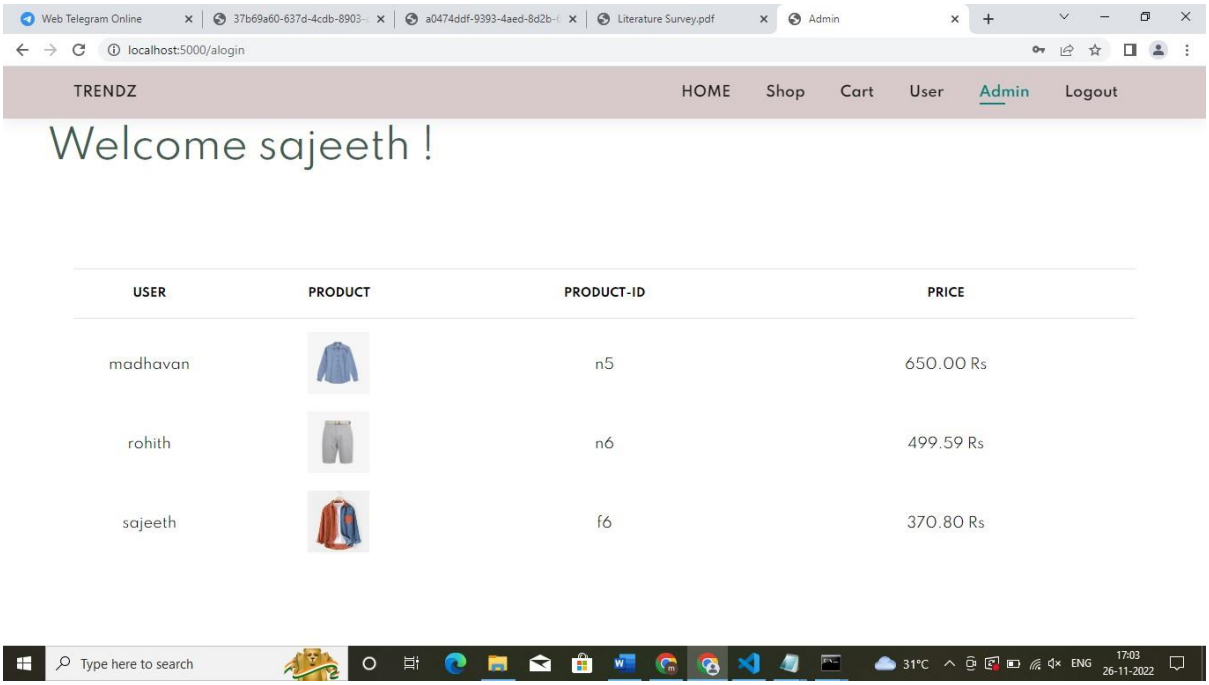




Cart Page:



Admin Page:



13.2 GITHUB & PROJECT DEMO LINK:

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-13598-1659523236

Project Demo LINK:

https://drive.google.com/file/d/1ijOiAAuK4qfmJQgu7U0ge8RvX8ALa6i6/view?usp=drivesdk