

IT - ITes SSC  
NASSCOM



# SMART FASHION RECOMMENDER APPLICATION

IBM – DOCUMENTATION

UNDER THE GUIDANCE OF

INDUSTRY MENTOR(S) NAME : KRISHNA CHAITANYA

FACULTY MENTOR(S) NAME : Dr KALAMANI C

TEAM ID: PNT2022TMID08771

SUBMITTED BY:

SABARI M	19BEC006
GURUSUDHAN M	19BEC012
NIYAS Y	19BEC036
AGALUDHAYA G	19BEC040



DEPARTMENT OF ELECTRONICS AND COMMUNICATION

ENGINEERING

DR. MAHALINGAM COLLEGE OF ENGINEERING AND

TECHNOLOGY

ANNA UNIVERSITY: 2019 – 2023

<b>S.NO</b>	<b>TABLE OF CONTENT</b>	<b>PG.NO</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	PROJECT OVERVIEW	<b>1</b>
1.2	PURPOSE	<b>1</b>
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>2</b>
2.1	EXISTING PROBLEM	<b>2</b>
2.2	REFERENCES	<b>3</b>
2.3	PROBLEM STATEMENT DEFINITION	<b>3</b>
<b>3</b>	<b>IDEATION &amp; PROPOSED SOLUTION</b>	<b>4</b>
3.1	EMPATHY MAP CANVAS	<b>4</b>
3.2	IDEATION & BRAINSTORMING	<b>4</b>
3.3	PROPOSED SOLUTION	<b>5</b>
3.4	PROBLEM SOLUTION FIT	<b>6</b>
<b>4</b>	<b>REQUIREMENT ANALYSIS</b>	<b>7</b>
4.1	FUNCTIONAL REQUIREMENT	<b>7</b>
4.2	NON-FUNCTIONAL REQUIREMENT	<b>7</b>
<b>5</b>	<b>PROJECT DESIGN</b>	<b>8</b>
5.1	DATA FLOW DIAGRAM	<b>8</b>
5.2	SOLUTION & TECHNICAL ARCHITECTURE	<b>8</b>
5.3	USER STORIES	<b>9</b>
<b>6</b>	<b>PROJECT PLANNING &amp; SCHEDULING</b>	<b>10</b>
6.1	SPRINT PLANNING & ESTIMATION	<b>10</b>
6.2	SPRINT DELIVERY SCHEDULE	<b>10</b>
6.3	REPORTS FROM JIRA	<b>11</b>
<b>7</b>	<b>CODING &amp; SOLUTIONING</b>	<b>12</b>
7.1	FEATURE 1	<b>12</b>

7.3	DATABASE SCHEMA	13
8	TESTING	14
8.1	TEST CASES	14
8.2	USER ACCEPTANCE TESTING	15
9	RESULTS	16
9.1	PERFORMANCE METRICS	16
10	ADVANTAGES & DISADVANTAGES	17
11	CONCLUSION	18
12	FUTURE SCOPE	19
13	APPENDIX	20
13.1	SOURCE CODE	20
13.2	GITHUB & PROJECT DEMO LINK	121

# 1. INTRODUCTION

## 1.1 PROJECT OVERVIEW

Nowadays, fashion applications and e-commerce are growing more and more, and it also has some problems when finding the customer's wanted product in the web applications. Having a chatbot that understands the algorithm of a specific application can be of great aid. We are implementing such a chat bot in a web application, 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 by getting simple user information and activities. The application also has two main UI interactions: one is the user panel and the other one is the admin panel. Users can interact with the chat bot to search for products, order them from the manufacturer or distributor through chatbot AI, and it can also make payment transactions, track the delivery, and so on. The admin interface enables the user to upload products' details, user details, orders and find how many products have been bought; supervise the stock availability; and interact with the buyer regarding the product reviews.

We have come up with a new innovative solution through which you can directly do your online shopping based on your choice without any search. It can be done by using the chat bot.

In this project you will be working on two modules:

1. Admin and
2. User

Admin:

The role of the admin is to check out the database about the stock and have a track of all the things that the users are purchasing.

User:

The user will login into the website and go through the products available on the website. Instead of navigating to several screens for booking products online, the user can directly talk to Chat bot regarding the products. Get the recommendations based on information provided by the user.

## 1.2 PURPOSE

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

## **2. LITERATURE SURVEY**

### **2.1 EXISTING PROBLEM**

#### **1. Fashion item representation**

Traditional recommender systems such as Collaborative Filtering or Content-Based Filtering have difficulties in the fashion domain due to the sparsity of purchase data, or the insufficient detail about the visual appearance of the product in category names. Instead, more recent literature has leveraged models that capture a rich representation of fashion items through product images, text descriptions or customer reviews, or videos which are often learned through surrogate tasks like classification or product retrieval. However, learning product representations from such input data requires large datasets to generalize well across different image (or text) styles, attribute variations, etc. Furthermore, constructing a representation that learns which product features customers take most into account when evaluating fashion products is still an open research problem.

#### **2. Fashion item compatibility**

Training a model that is able to predict if two fashion items ‘go together,’ or directly combine several products into an outfit, is a challenging task. Different item compatibility signals studied in recent literature include co-purchase data, outfits composed by professional fashion designers, or combinations found by analyzing what people wear in social media pictures.

#### **3. Personalization and fit**

The best fashion product to recommend depends on factors such as the location where the outfit will be used, the season or occasion, or the cultural and social background of the customer. A challenging task in fashion recommendation systems is how to discover and integrate these disparate factors. Current research often tackles these tasks by utilizing large-scale social media data.

#### **4. Interpretability and explanation**

Most of the existing fashion recommender systems in the literature focus on improving predictive performance, treating the model as a black box. However, deploying accountable and interpretable systems able to explain their recommendations can foster user loyalty in the long term and improve the shopping experience

#### **5. Discovering trends**

Being able to forecast consumer preferences is valuable for fashion designers and retailers in order to optimize product-to-market fit, logistics and advertising.

## **2.2 REFERENCES**

### **“A Systematic Study on the Recommender Systems in the E-Commerce”**

Electronic commerce or e-commerce includes the service and good exchange through electronic support like the Internet. It plays a crucial role in today's business and users' experience. Also, e-commerce platforms produce a vast amount of information. So, Recommender Systems (RSs) are a solution to overcome the information overload problem. They provide personalized recommendations to improve user satisfaction. The present article illustrates a comprehensive and Systematic Literature Review (SLR) regarding the papers published in the field of e-commerce recommender systems. We reviewed the selected papers to identify the gaps and significant issues of the RSs' traditional methods, which guide the researchers to do future work. So, we provided the traditional techniques, challenges, and open issues concerning traditional methods of the field of review based on the selected papers. This review includes five categories of the RSs' algorithms, including Content-Based Filtering (CBF), Collaborative Filtering (CF), Demographic-Based Filtering (DBF), hybrid filtering, and Knowledge-Based Filtering (KBF).

## **2.3 PROBLEM STATEMENT DEFINITION**

Problem Statement 1:

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

Problem Statement 2:

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

Problem Statement 3:

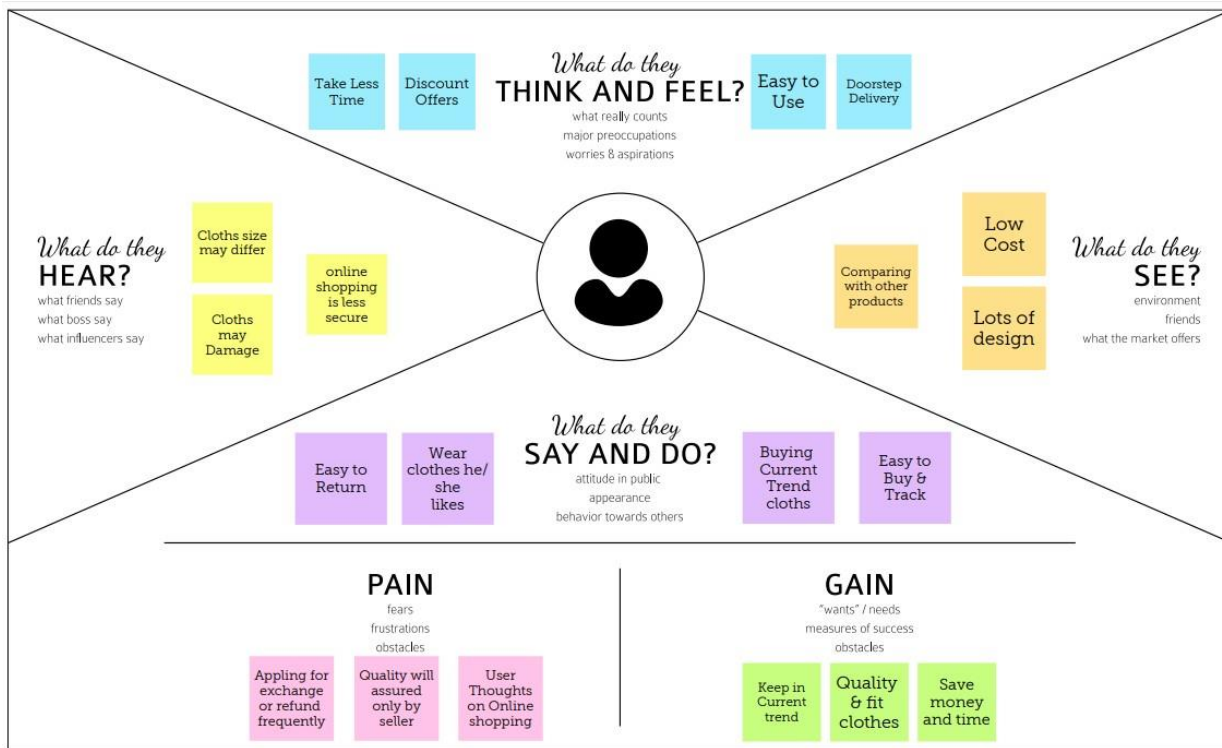
The User Needs a way to Assistant for finding Clothes so that Here User got the Chat Bot assistant

Problem Statement 4:

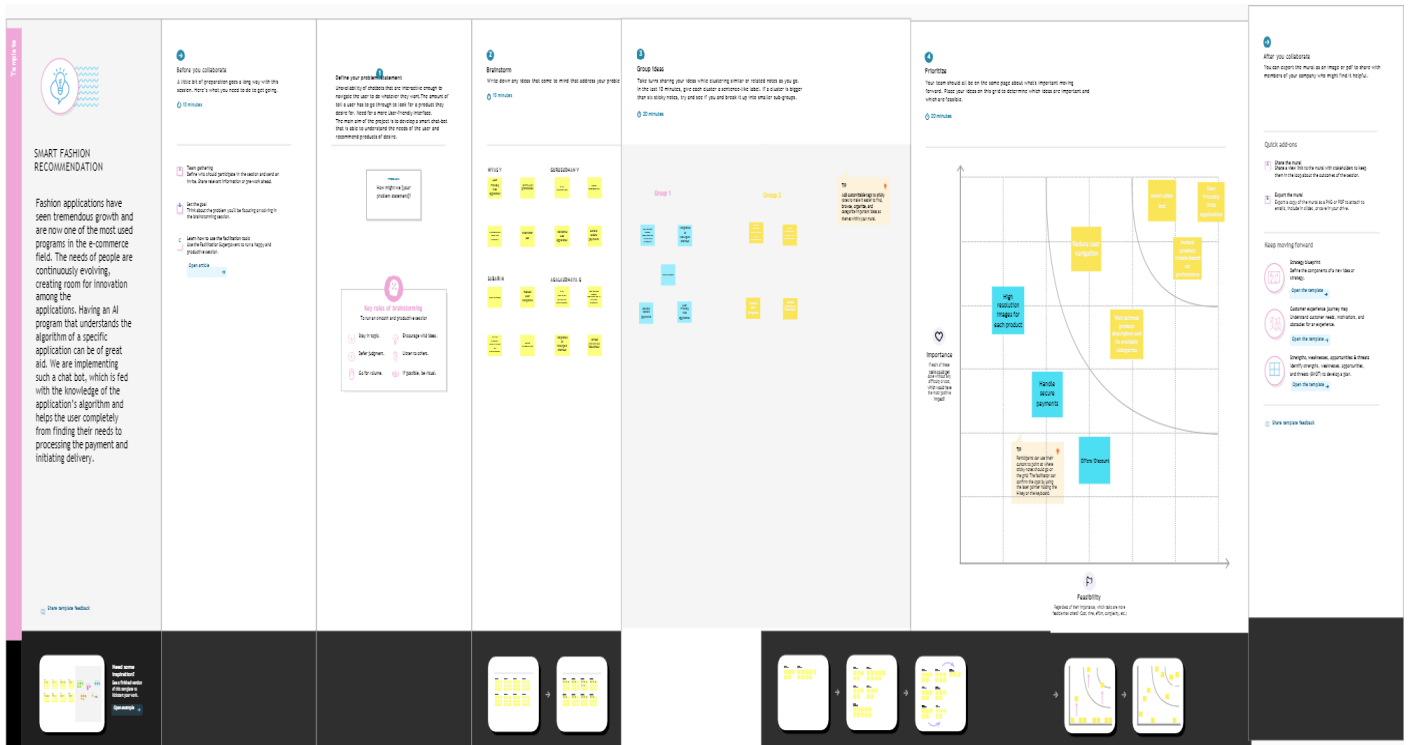
The Sellers Needs a way to struggling to sells products offline so that Here Sellers will Sell Products via our application.

### 3. IDEATION & PROPOSED SOLUTION

#### 3.1 EMPATHY MAP CANVAS



#### 3.2 IDEATION & BRAINSTORMING



### 3.3 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 customer 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



### 3.4 PROBLEM SOLUTION FIT

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <small>Who is your customer? i.e. working parents of 0-5 y.o. kids</small> <div style="border: 1px solid black; padding: 10px; text-align: center;">The Customers are Adults and children</div>	<b>6. CUSTOMER CONSTRAINTS</b> <small>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.</small> <div style="border: 1px solid black; padding: 10px; text-align: center;">Money and Network Connection</div>	<b>5. AVAILABLE SOLUTIONS</b> <small>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 &amp; cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking</small> <div style="border: 1px solid black; padding: 10px; text-align: center;">Online shopping gives New Collections pros: Easy to use cons: customer confused when have lost of collections</div>	Explore AS, differentiate
	Focus on J&P, tap into BE, understand RC	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <small>Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one, explore different sides.</small> <div style="border: 1px solid black; padding: 10px; text-align: center;">Users hard to find Trending Fashion Clothes.</div>	<b>9. PROBLEM ROOT CAUSE</b> <small>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.</small> <div style="border: 1px solid black; padding: 10px; text-align: center;">Customers need to be with new fashions for current trends</div>	
Identify strong TR & EM		<b>3. TRIGGERS</b> <small>What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.</small> <div style="border: 1px solid black; padding: 10px; text-align: center;">Seeing neighbor Dressing Styles</div>	<b>10. YOUR SOLUTION</b> <small>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 in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.</small> <div style="border: 1px solid black; padding: 10px; text-align: center;">Make a ChatBot Assistant for shopping with customers and send notifications when new collections arrived</div>	<b>8. CHANNELS of BEHAVIOUR</b> <b>8.1 ONLINE</b> <small>What kind of actions do customers take online? Extract online channels from #7</small> <b>8.2 OFFLINE</b> <small>What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.</small> <div style="border: 1px solid black; padding: 10px; text-align: center;">ONLINE: Customers buy the new clothes OFFLINE: Customers will use the clothes</div>
	<b>4. EMOTIONS: BEFORE / AFTER</b> <small>How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure &gt; confident, in control - use it in your communication strategy &amp; design.</small> <div style="border: 1px solid black; padding: 10px; text-align: center;">Felling Sad and Frustration &gt; Selfconfident</div>			

## 4. REQUIREMENT ANALYSIS

### 4.1 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

### 4.2 NON-FUNCTIONAL REQUIREMENTS

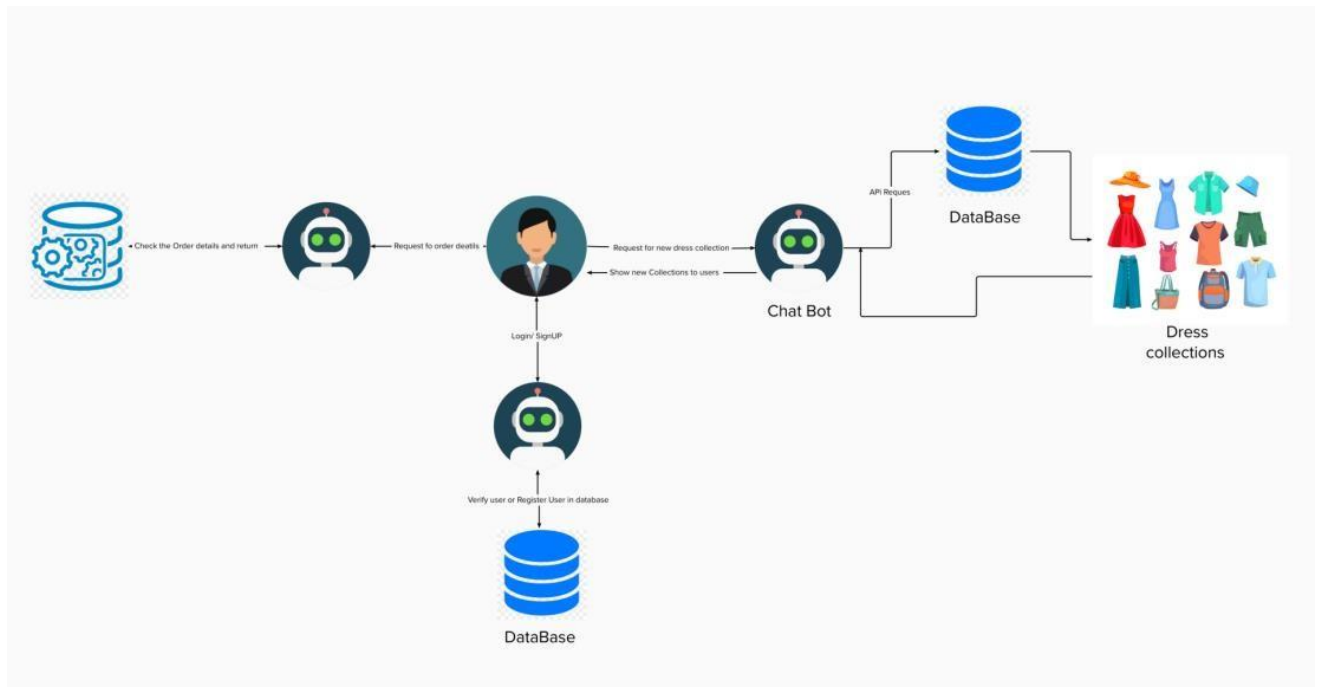
Following are the non-functional requirements of the proposed solution.

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

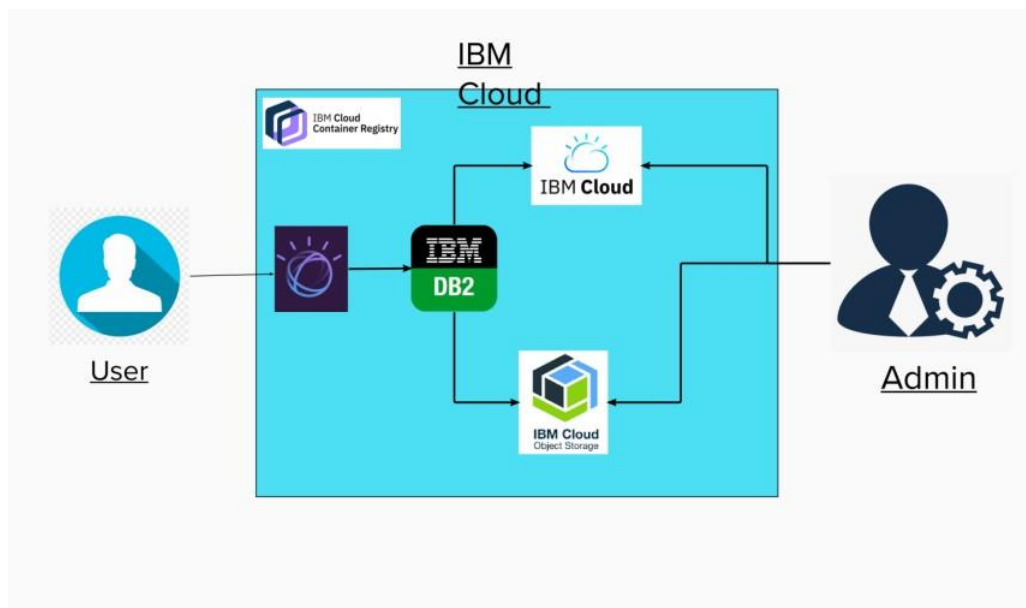
## 5. PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAMS

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



### 5.2 SOLUTION & TECHNICAL ARCHITECTURE



## 5.3 USER STORIES

### User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my data by login	High	Sprint-1
	Dashboard	USN-6	As a user , I can view the dashboard and by products		High	Sprint -2
Customer (Web user)	Registration / Login	USN-7	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard		Sprint -1
Customer Care Executive	Contact with Customers	USN-8	As a Customer customers care executive, I solve the customer Requirements and feedback	I can receive calls from customers	High	Sprint-1
Administrator	Check stock and Price , orders	USN_9	As a Administrator , I can Check the database And stock details and buying and selling prices	I am the administrator of the company	High	Sprint -2

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 SPRINT PLANNING & ESTIMATION

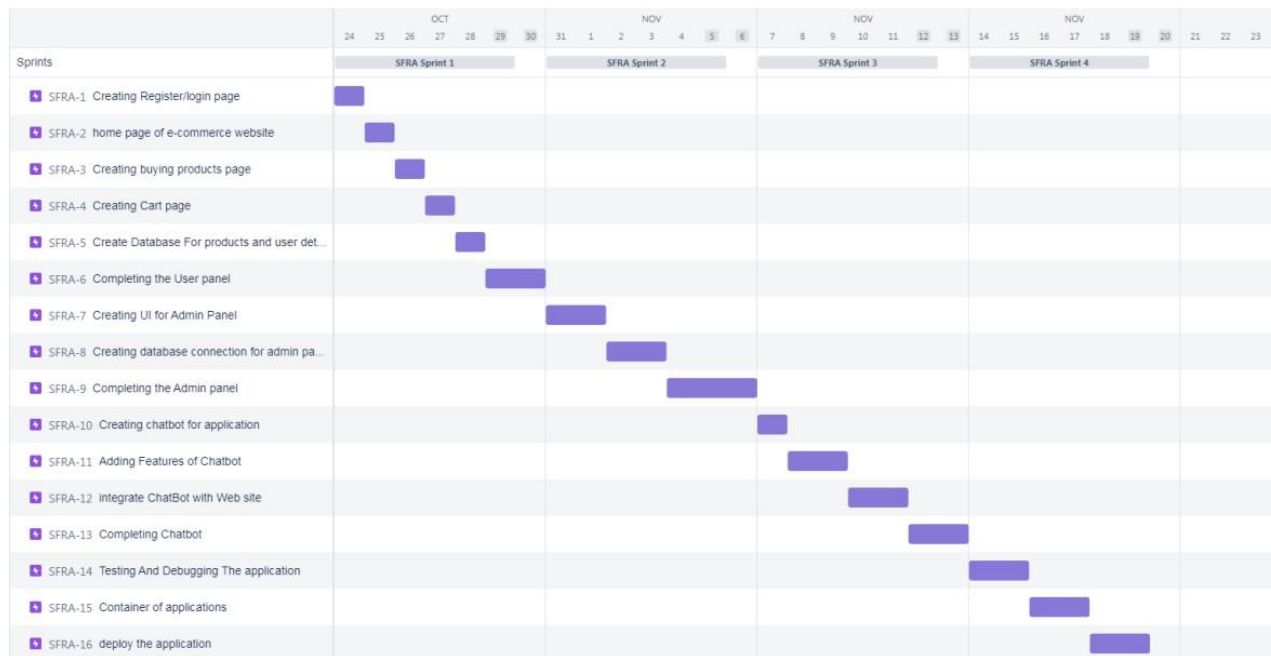
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	User Panel	USN-1	The user will login into the website and go through the products available on the website	20	High
Sprint-2	Admin panel	USN-2	The role of the admin is to check out the database about the stock and have a track of all the things that the users are purchasing.	20	High
Sprint-3	Chat Bot	USN-3	The user can directly talk to Chatbot regarding the products. Get the recommendations based on information provided by the user.	20	High
Sprint-4	final delivery	USN-4	Container of applications using docker kubernetes and deployment the application. Create the documentation and final submit the application	20	High

### 6.2 SPRINT DELIEVERY SCHEDULE

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

## 6.3 REPORTS FROM JIRA

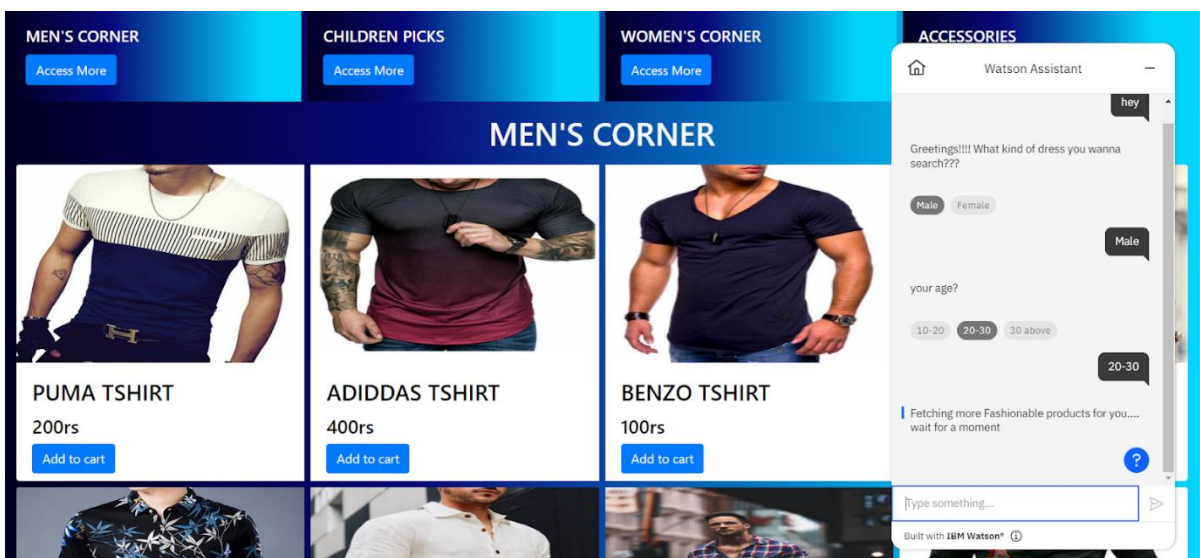
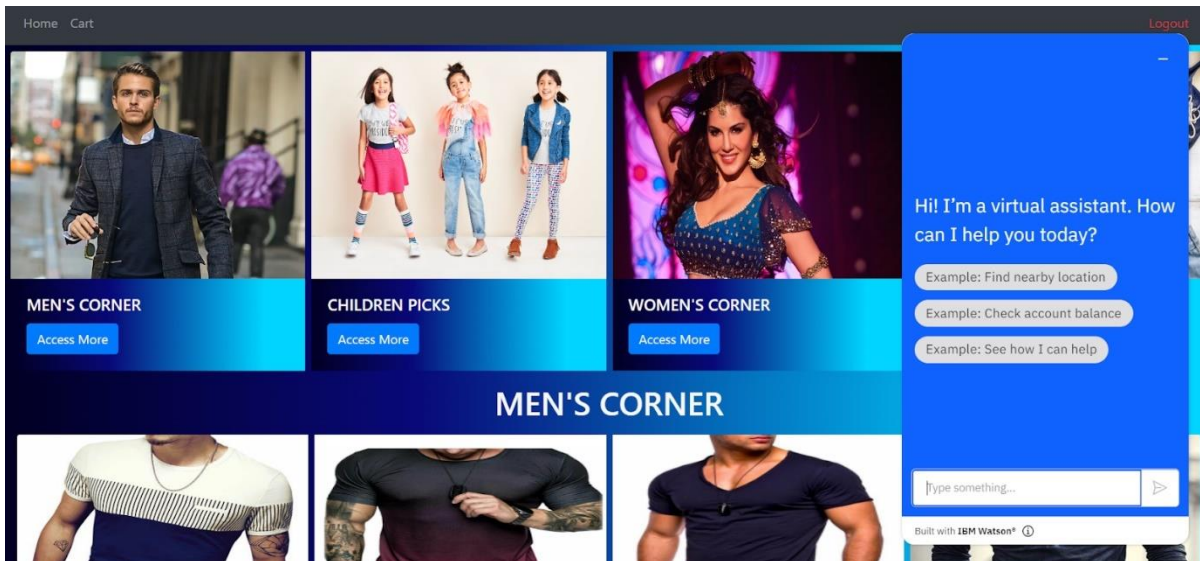
**Burndown Chart:**



## 7. CODING & SOLUTIONING

### 7.1 FEATURE 1

Using chat bot we can fetch products requiring.





## 7.2 DATABASE SCHEMA

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

LPR87378

Tables

USERS

Views

MQTs

Aliases

Nicknames

\*Untitled - 1 x +

1 SELECT \* FROM USERS

Syntax assistant Run all

History Results

Result set 1 Details

Filter table Total:10

EMAIL	FIRSTNAME	PWD
19bec012@mcet.in	gurusudhan	\$2b\$12\$S4vRoJqSIIGkBM30NusfFuUBovWWvcJXTZ89g/9dyqhbwiqPU9BgC
gurusudhan25@gmail.com	Gurusudhan	\$2b\$12\$Q76doOJ.MSWmV2Uc5N0j1en2U18T03UNihj9vijy7xk4Om1rHpIIdy
niyas@gmail.com	Niyas	\$2b\$12\$yVcenAgy.7WzF7jIj6aNbuM3m780.s12Do/o9KD7s.1rRkSH64oSO
dhevadarshiri@gmail.com	Dheva	\$2b\$12\$xDa2si8GhJWw00r753NjZORkUE1INyBD10ovGL3B9txN6mduCPK
19bec006@mcet.in	Sabari	\$2b\$12\$Fg/ZL92A7.JSY1cwrQp4j.YXcwxDav6qM5NNzp9ywnPBID1a6TpQu
gurusudhan000@gmail.com	Sudhan V	\$2b\$12\$NFX8m2Tuxy8FuJ343nHXkIOO1VvhGHnUGEhsaNd8mBZwqtOGMDeJI6
727619bec006@smartinternz.com	Abinaya	\$2b\$12\$4R4fK1fW8b8MkU931aTgeuCpNrgg.VzRxEK233wuBIop/.DSrS7e.
check@email.com	Guru	\$2b\$12\$eQHfg0IC2gXnxOfYf6PWRO7Z1ki.ulTE6MNEoOu9ZOXdlhg0ohrYm
sindhu@gmail.com	sindhu	\$2b\$12\$st7S/yqPOHKAQH6p2sdb310mUBxf17.D4VcnzmF9N19lqaKNAssh5m



## 8. TESTING

### 8.1 TEST CASES

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

Section	Total Cases	Not Tested	Fail	Pass
Login	5	0	0	5
Register	7	0	0	7
Home Page	2	0	0	2
Order page	3	0	0	3
Order products	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

## 8.2 USER ACCEPTANCE TESTING

### Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the 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	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	5	2	3	21
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

## 9. RESULT

### 9.1 PERFORMANCE METRICS

Project team shall fill the following information in model performance testing.

NFT - Risk Assessment									
S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volumem Changes	Risk Score	Justification
1	Smart Fashion Recommender Application	New	Low	No Changes	Moderate		>\$ to 10%	ORANGE	As we have seen the changes

NFT - Detailed Test Plan				
S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	Smart Fashion Recommender Application	Manual testing	laptop or mobile with internet connection	vikparameshwaran

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)
1	Smart Fashion Recommender Application	Manule		Worked as we expected		Use Laptop / desktop Mode	No Defects
							Vikparameshwaran

## **10. ADVANTAGES & DISADVANTAGES**

### **ADVANTAGES:**

- Its helps to user Shopping with Assistant
- Its helps to user manage their order list
- Its helps to user shopping at home

### **DISADVANTAGES:**

- User have fear about online shopping
- User have sometimes received wrong items
- User have fear about online payment

## 11. CONCLUSION

Recommendation systems have the potential to explore new opportunities for retailers by enabling them to provide customized recommendations to consumers based on information retrieved from the Internet. They help consumers to instantly find the products and services that closely match with their choices. Moreover, different state-of-the-art algorithms have been developed to recommend products based on users' interactions with their social groups. Therefore, research on embedding social media images within fashion recommendation systems has gained huge popularity in recent times. This paper presented a review of the fashion recommendation systems, algorithmic models and filtering techniques based on the academic articles related to this topic. The technical aspects, strengths and weaknesses of the filtering techniques have been discussed elaborately, which will help future researchers gain an in-depth understanding of fashion recommender systems. However, the proposed prototypes should be tested in commercial applications to understand their feasibility and accuracy in the retail market, because inaccurate recommendations can produce a negative impact on a customer. Moreover, future research should concentrate on including time series analysis and accurate categorization of product images based on the variation in color, trend and clothing style in order to develop an effective recommendation system. The proposed model will follow brand specific personalization campaigns and hence it will ensure highly curated and tailored offerings for users. Hence, this research will be highly beneficial for researchers interested in using augmented and virtual reality features to develop recommendation systems.

## **12. FUTURE SCOPE**

There has been significant progress recently in fashion recommendation system research, which will benefit both consumers and retailers soon. The use of product and user images, textual content, demographic history, and cultural information is crucial in developing recommendation frameworks. Product attributes and clothing style matching are common features of 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 history and product reviews from the recommendation system and subsequently use them in style prediction for the upcoming seasons. The integration of different domain information strengthens the deep learning paradigm by enabling the detection of design component variation, which improves the performance of the recommendation system in the long run. Deep learning approaches should be more frequently used to quickly explore fashion items from different online databases to provide prompt recommendations to users or consumers.

## 13. APPENDIX

### 13.1 SOURCE CODE

#### App.py

```
from flask import Flask, render_template, request,
redirect, url_for, session
import ibm_db
import bcrypt
import sendgrid
import os
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ea286ace-
86c7-4d5b-8580-
3fbfa46b1c66.bs2io90l08kqb1od8lclg.databases.appdomain.cl
oud;PORT=31505;SECURITY=SSL;SSLServiceCertificate=DigiCe
rtGlobalRootCA.crt;UID=lpr87378;PWD=z1LR9MwJzLBYvmD4",
'', '')
app = Flask(__name__)
app.secret_key = b'_5#y2L"F4Q8z\n\xec]/'

@app.route("/", methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        print(email, password)
        if not email or not password:
            return render_template('login.html',
error='Please fill all the fields')
        query = "SELECT * FROM USERS 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)

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

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

        if not isPasswordMatch:
            return render_template('login.html',
error='Incorrect password')
            return render_template('home.html')

        return render_template("Login.html")

@app.route("/signup", methods=['GET', 'POST'])
def signup():
    if request.method == 'POST':
        email = request.form['email']
        firstName = request.form['firstName']
        password1 = request.form['password1']
        password2 = request.form['password2']
        print(password1, password2)
        if not email or not firstName or not password1
or not password2:
            return render_template('signup.html',
error='Please fill all fields')
            hash = bcrypt.hashpw(password1.encode('utf-8'),
bcrypt.gensalt())

```



```

    print(hash)
    query = "SELECT * FROM Users 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)
    if isUser:
        return render_template('login.html',
msg="You are already a member, please login using your
details")
    if not isUser:
        insert_sql = "INSERT INTO Users VALUES
(?,?,?)"

        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt, 1, email)
        ibm_db.bind_param(prepare_stmt, 2, firstName)
        ibm_db.bind_param(prepare_stmt, 3, hash)
        ibm_db.execute(prepare_stmt)
        return redirect(url_for('login'))
    else:
        return render_template('signup.html',
error='Invalid Credentials')

    return render_template("signup.html")

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

@app.route("/cart")
def cart():

```

```

    return render_template("cart.html")

def sendemail():
    sg =
sendgrid.SendGridAPIClient(api_key=os.environ.get('SENDGRID_API_KEY'))
    data = {
        "personalizations": [
            {
                "to": [
                    {
                        "email":
"gurusudhan25@gmail.com"
                    }
                ],
                "subject": "Thankyou for purchasing"
            }
        ],
        "from": {
            "email": "19bec012@mcet.in"
        },
        "content": [
            {
                "type": "text/plain",
                "value": "Oder successfull , your order
arrive within 5 working days"
            }
        ]
    }
    response =
sg.client.mail.send.post(request_body=data)
    print(response)

```

```
if __name__ == "__main__":  
    app.run(debug=True)
```

Login Page

```
<!DOCTYPE html>  
<html lang="en">  
    <head>  
        <meta charset="UTF-8" />  
        <meta http-equiv="X-UA-Compatible" content="IE=edge"  
/>  
        <meta name="viewport" content="width=device-width,  
initial-scale=1.0" />  
        <link rel="icon" href="{{ url_for('static', filename  
= './icon/fashion-icon-app.png') }}" />  
        <link  
            href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1  
/dist/css/bootstrap.min.css"  
            rel="stylesheet"  
        />  
        <link  
            rel="stylesheet"  
            href="{{ url_for('static', filename =  
'./css/style.css') }}"  
        />  
        <title>{%block title%}Home{%endblock%}</title>  
</head>  
<body>
```

```

<div class="more">
  {% block content2 %}
  <div class="header">
    <!--Content before waves-->
    <div class="inner-header flex">
      <div
        class="container-fluid w-custom h-custom mt-custom
border pl-custom shadow-lg"
      >
        <form method="POST">
          <div class="name">
            <h3 align="center">USER LOGIN</h3>
          </div>
          <div class="form-group ">
            <input
              type="email"
              class="form-control active border-custom
shadow mt-3"
              id="email"
              name="email"
              placeholder="Enter email" required
            />
          </div>
          <div class="form-group">
            <input
              type="password"
              class="form-control border-custom shadow mt-3"
              id="password"
              name="password"
              placeholder="Enter password" required
            />
          </div>
        </form>
      </div>
    </div>
  </div>

```

```

    <br />
    <button type="submit" class="btn btn-custom mb-
custom mb-2">Login</button>
    <div class="bottom">
        <input type="checkbox" , id="rem">
        <label for="rem">Remember Me</label>
        <a href="#" class="fl">Forgot Password?</a>
    </div>
    <button class="btn btn-cus mt-2"><a
href="/signup">Create an Account</a></button>
</form>
</div>

</div>

<!--Waves Container-->
<div>
    <svg class="waves mt-5"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 24
150 28" preserveAspectRatio="none" shape-
rendering="auto">
        <defs>
            <path id="gentle-wave" d="M-160 44c30 0
58-18 88-18s 58 18 88 18 58-18 88-18 58 18 88 18 v44h-
352z" />
        </defs>
        <g class="parallax">
            <use xlink:href="#gentle-wave" x="48"
y="0" fill="rgba(255,255,255,0.7)" />
            <use xlink:href="#gentle-wave" x="48"
y="3" fill="rgba(255,255,255,0.5)" />

```

```

        <use xlink:href="#gentle-wave" x="48"
y="5" fill="rgba(255,255,255,0.3)" />
        <use xlink:href="#gentle-wave" x="48"
y="7" fill="#fff" />
    </g>
</svg>
</div>
<!--Waves end-->

</div>
<!--Header ends-->
{%endblock%}
<script>
    window.watsonAssistantChatOptions = {
        integrationID: "ed691e2a-3755-46f6-ba80-
2937b1ab411d", // The ID of this integration.
        region: "au-syd", // The region your
integration is hosted in.
        serviceInstanceID: "e0026043-09be-4231-8e79-
09ea7ca8992f", // 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.clientVer
sion || "latest") +

```

```

        "/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
    });
</script>
</body>
</html>

```

Signup page

```

{% extends "Login.html" %} {% block title %}Signup{%
endblock %} {%block
content2%}
<div class="header">
    <!--Content before waves-->
    <div class="inner-header flex">
        <div
            class="container-fluid w-custom mt-custom pt-
custom border border-dark pl-custom"
        >
            <form method="POST">
                <div class="name mt-0">
                    <h3 align="center">SIGN UP</h3>
                </div>
                <div class="form-group">
                    <input
                        type="email"
                        class="form-control border-custom shadow mt-
3"
                        id="email"
                        name="email"
                        placeholder="Enter email"
                        required

```

```

        />
</div>
<div class="form-group">
  <input
    type="text"
    class="form-control border-custom shadow mt-
3"
    id="firstName"
    name="firstName"
    placeholder="Enter first name"
    required
  />
</div>
<div class="form-group">
  <input
    type="password"
    class="form-control border-custom shadow mt-
3"
    id="password1"
    name="password1"
    placeholder="Enter password"
    required
  />
</div>
<div class="form-group">
  <input
    type="password"
    class="form-control border-custom shadow mt-
3"
    id="password2"
    name="password2"
    placeholder="Confirm password"

```



```

        required
    />
</div>
<br />
<button type="submit" class="btn btn-custom mb-
2">Submit</button>
<button class="btn btn-cus mb-3">
    <a href="/">Already Have An Account</a>
</button>
</form>
</div>
</div>
<!--Waves Container-->
<div>
    <svg
        class="waves mt-5"
        xmlns="http://www.w3.org/2000/svg"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        viewBox="0 24 150 28"
        preserveAspectRatio="none"
        shape-rendering="auto"
    >
        <defs>
            <path
                id="gentle-wave"
                d="M-160 44c30 0 58-18 88-18s 58 18 88 18 58-
18 88-18 58 18 88 18 v44h-352z"
            />
        </defs>
        <g class="parallax">
            <use
                xlink:href="#gentle-wave"

```

```

        x="48"
        y="0"
        fill="rgba(255,255,255,0.7"
    />
    <use
        xlink:href="#gentle-wave"
        x="48"
        y="3"
        fill="rgba(255,255,255,0.5)"
    />
    <use
        xlink:href="#gentle-wave"
        x="48"
        y="5"
        fill="rgba(255,255,255,0.3)"
    />
    <use xlink:href="#gentle-wave" x="48" y="7"
fill="#fff" />
    </g>
</svg>
</div>
<!--Waves end-->
</div>
{% endblock %}

```

Home page

```

<!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>Home</title>
    <link
        rel="icon"
        href="{{ url_for('static', filename =
'./icon/fashion-icon-app.png') }}"
    />
    <link
        rel="stylesheet"
        href="https://stackpath.bootstrapcdn.com/bootstrap
/4.4.1/css/bootstrap.min.css"
        integrity="sha384-
Vkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf
23Q9Ifjh"
        crossorigin="anonymous"
    />
    <link
        rel="stylesheet"
        href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css"
        crossorigin="anonymous"
    />
    <link
        rel="stylesheet"
        href="{{ url_for('static', filename =
'/css/home.css') }}"
    />
</head>
<body>

```

```

<nav class="navbar navbar-expand-lg navbar-dark bg-
dark">
  <button
    class="navbar-toggler"
    type="button"
    data-toggle="collapse"
    data-target="#navbar"
  >
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbar">
    <ul class="navbar-nav">
      <li>
        <a class="nav-item nav-link" id="home"
href="/home">Home</a>
      </li>
      <li>
        <a class="nav-item nav-link" id="home"
href="/cart">Cart</a>
      </li>
    </ul>
  </div>
  <div id="navbar-nav float-right">
    <a class="nav-item nav-link text-danger"
id="logout" href="/">Logout</a>
  </div>
</nav>
<div class="content">
  {%block content%}
  <div class="card-group">
    <div class="card ml-2 mt-2 h-custom"
style="width: 18rem">

```

```

        
        <div class="card-body">
            <h5 class="card-title">MEN'S CORNER</h5>
            <a href="#MENS" class="btn btn-primary"
id="MEN'S CORNER"
            >Access More</a>
        >
    </div>
</div>
<div class="card ml-2 mt-2 h-custom"
style="width: 18rem">
    
    <div class="card-body">
        <h5 class="card-title">CHILDREN PICKS</h5>
        <a href="#CHILD" class="btn btn-
primary">Access More</a>
    </div>
</div>
<div class="card ml-2 mt-2 h-custom"
style="width: 18rem">
    
    <div class="card-body">
        <h5 class="card-title">WOMEN'S CORNER</h5>
        <a href="#WOMEN" class="btn btn-
primary">Access More</a>
    </div>
</div>
<div class="card ml-2 mt-2 mr-2 h-custom"
style="width: 18rem">
    
    <div class="card-body">
        <h5 class="card-title">ACCESSORIES</h5>
        <a href="#ACCESS" class="btn btn-
primary">Access More</a>
    </div>
</div>
</div>
<div class="card mt-3 ml-2 mr-2 p-0 bg-dark text-
light" id="MENS">
    <h1 align="center">MEN'S CORNER</h1>
    <div class="card-group">
        <div class="card c2 ml-2 mt-2" style="width:
18rem">
            
    <div class="card-body na">
        <h3 class="card-title">PUMA TSHIRT</h3>
        <h4>200rs</h4>
        <button align="center" class="btn btn-
primary">
            Add to cart
        </button>
    </div>
</div>
<div class="card c2 ml-2 mt-2" style="width:
18rem">
    
    <div class="card-body na">
        <h3 class="card-title">ADIDDAS TSHIRT</h3>
        <h4>400rs</h4>
        <button class="btn btn-primary float-
left">Add to cart</button>
    </div>
</div>
<div class="card c2 ml-2 mt-2" style="width:
18rem">
    
    <div class="card-body na">
        <h3 class="card-title">BENZO TSHIRT</h3>
        <h4>100rs</h4>
        <button align="center" class="btn btn-
primary">
            Add to cart
        </button>
    </div>
</div>
<div class="card c2 ml-2 mt-2 mr-2"
style="width: 18rem">
    
    <div class="card-body na">
        <h3 class="card-title">HIFI TSHIRT</h3>
        <h4>300rs</h4>
        <button align="center" class="btn btn-
primary">
            Add to cart
        </button>
    </div>
</div>
</div>

```



```

<div class="card-group">
  <div class="card c2 ml-2 mt-2" style="width:
18rem">
    
    <div class="card-body na">
      <h3 class="card-title">PETER ENGLAND</h3>
      <h4>980rs</h4>
      <button align="center" class="btn btn-
primary">
        Add to cart
      </button>
    </div>
  </div>
  <div class="card c2 ml-2 mt-2" style="width:
18rem">
    
    <div class="card-body na">
      <h3 class="card-title">LEVI'S</h3>
      <h4>780rs</h4>
      <button class="btn btn-primary float-
left">Add to cart</button>
    </div>
  </div>
</div>

```

```

</div>
<div class="card c2 ml-2 mt-2" style="width:
18rem">
    
    <div class="card-body na">
        <h3 class="card-title">RAZE SHIRT</h3>
        <h4>1000rs</h4>
        <button align="center" class="btn btn-
primary">
            Add to cart
        </button>
    </div>
</div>
<div class="card c2 ml-2 mt-2 mr-2"
style="width: 18rem">
    
    <div class="card-body na">
        <h3 class="card-title">PRINTED SHIRT</h3>
        <h4>1300rs</h4>
        <button align="center" class="btn btn-
primary">
            Add to cart

```

```

        </button>
      </div>
    </div>
  </div>
</div>
<div class="card mt-3 ml-2 mr-2 p-0 bg-dark text-
light" id="WOMEN">
  <h1 align="center">WOMEN'S CORNER</h1>
  <div class="card-group">
    <div class="card c2 ml-2 mt-2" style="width:
18rem">
      
      <div class="card-body na">
        <h3 class="card-title">JEAN AND
TSHIRT</h3>
        <h4>200rs</h4>
        <button align="center" class="btn btn-
primary">
          Add to cart
        </button>
      </div>
    </div>
    <div class="card c2 ml-2 mt-2" style="width:
18rem">
      
    <div class="card-body na">
        <h3 class="card-title">CROP TOP</h3>
        <h4>400rs</h4>
        <button class="btn btn-primary float-
left">Add to cart</button>
    </div>
</div>
<div class="card c2 ml-2 mt-2" style="width:
18rem">
    
    <div class="card-body na">
        <h3 class="card-title">BALLON TSHIRT
LEGGIN</h3>
        <h4>100rs</h4>
        <button align="center" class="btn btn-
primary">
            Add to cart
        </button>
    </div>
</div>
<div class="card c2 ml-2 mt-2 mr-2"
style="width: 18rem">
    
    <div class="card-body na">
        <h3 class="card-title">LEHENG</h3>
        <h4>300rs</h4>
        <button align="center" class="btn btn-
primary">
            Add to cart
        </button>
    </div>
</div>
<div class="card-group">
    <div class="card c2 ml-2 mt-2" style="width:
18rem">
        
        <div class="card-body na">
            <h3 class="card-title">RED HOT TSHIRT</h3>
            <h4>980rs</h4>
            <button align="center" class="btn btn-
primary">
                Add to cart
            </button>
        </div>
    </div>
</div>

```

```

    </div>
    <div class="card c2 ml-2 mt-2" style="width:
18rem">
        
        <div class="card-body na">
            <h3 class="card-title">LEVI'S WOMEN'S
SHIRT</h3>
            <h4>780rs</h4>
            <button class="btn btn-primary float-
left">Add to cart</button>
        </div>
    </div>
    <div class="card c2 ml-2 mt-2" style="width:
18rem">
        
        <div class="card-body na">
            <h3 class="card-title">RAZE WOMEN'S
SHIRT</h3>
            <h4>1000rs</h4>
            <button align="center" class="btn btn-
primary">
                Add to cart

```

```

        </button>
      </div>
    </div>
    <div class="card c2 ml-2 mt-2 mr-2"
style="width: 18rem">
      
      <div class="card-body na">
        <h3 class="card-title">CROP TSHIRT</h3>
        <h4>1300rs</h4>
        <button align="center" class="btn btn-
primary">
          Add to cart
        </button>
      </div>
    </div>
  </div>
  <div class="card mt-3 ml-2 mr-2 p-0 bg-dark text-
light" id="CHILD">
    <h1 align="center">KIDS'S CORNER</h1>
    <div class="card-group">
      <div class="card c2 ml-2 mt-2" style="width:
18rem">
        
    <div class="card-body na">
        <h3 class="card-title">KID'S HOODIE</h3>
        <h4>200rs</h4>
        <button align="center" class="btn btn-
primary">
            Add to cart
        </button>
    </div>
</div>
<div class="card c2 ml-2 mt-2" style="width:
18rem">
    
    <div class="card-body na">
        <h3 class="card-title">KIDS FAV</h3>
        <h4>400rs</h4>
        <button class="btn btn-primary float-
left">Add to cart</button>
    </div>
</div>
<div class="card c2 ml-2 mt-2" style="width:
18rem">
    <img
        class="card-img-top h1-custom"

```



```

        src="{{ url_for('static', filename =
'/children/pic3.jpeg') }}"
        alt="Card image cap"
    />
    <div class="card-body na">
        <h3 class="card-title">BENZO BLACK
TSHIRT</h3>
        <h4>100rs</h4>
        <button align="center" class="btn btn-
primary">
            Add to cart
        </button>
    </div>
</div>
<div class="card c2 ml-2 mt-2 mr-2"
style="width: 18rem">
    
    <div class="card-body na">
        <h3 class="card-title">COAT SUIT</h3>
        <h4>300rs</h4>
        <button align="center" class="btn btn-
primary">
            Add to cart
        </button>
    </div>
</div>
</div>

```

```

<div class="card-group">
  <div class="card c2 ml-2 mt-2" style="width:
18rem">
    
    <div class="card-body na">
      <h3 class="card-title">KID'S PANT
SHIRT</h3>
      <h4>980rs</h4>
      <button align="center" class="btn btn-
primary">
        Add to cart
      </button>
    </div>
  </div>
  <div class="card c2 ml-2 mt-2" style="width:
18rem">
    
    <div class="card-body na">
      <h3 class="card-title">PAIJAMA</h3>
      <h4>780rs</h4>
      <button class="btn btn-primary float-
left">Add to cart</button>

```

```

        </div>
    </div>
    <div class="card c2 ml-2 mt-2" style="width:
18rem">
        
        <div class="card-body na">
            <h3 class="card-title">PINK DRESS</h3>
            <h4>1000rs</h4>
            <button align="center" class="btn btn-
primary">
                Add to cart
            </button>
        </div>
    </div>
    <div class="card c2 ml-2 mt-2 mr-2"
style="width: 18rem">
        
        <div class="card-body na">
            <h3 class="card-title">SHOE WITH
LIGHT</h3>
            <h4>1300rs</h4>

```

```

        <button align="center" class="btn btn-
primary">
            Add to cart
        </button>
    </div>
</div>
</div>
</div>
<div class="card mt-3 ml-2 mr-2 p-0 bg-dark text-
light" id="ACCESS">
    <h1 align="center">ACCESSORIES</h1>
    <div class="card-group">
        <div class="card c2 ml-2 mt-2" style="width:
18rem">
            
            <div class="card-body na">
                <h3 class="card-title">CAP</h3>
                <h4>200rs</h4>
                <button align="center" class="btn btn-
primary">
                    Add to cart
                </button>
            </div>
        </div>
    </div>
    <div class="card c2 ml-2 mt-2" style="width:
18rem">
        
    <div class="card-body na">
        <h3 class="card-title">HAND BAG</h3>
        <h4>400rs</h4>
        <button class="btn btn-primary float-
left">Add to cart</button>
    </div>
</div>
<div class="card c2 ml-2 mt-2" style="width:
18rem">
    
    <div class="card-body na">
        <h3 class="card-title">WOMEN'S PURSE</h3>
        <h4>100rs</h4>
        <button align="center" class="btn btn-
primary">
            Add to cart
        </button>
    </div>
</div>
<div class="card c2 ml-2 mt-2 mr-2"
style="width: 18rem">
    
    <div class="card-body na">
        <h3 class="card-title">WATCH</h3>
        <h4>300rs</h4>
        <button align="center" class="btn btn-
primary">
            Add to cart
        </button>
    </div>
</div>
<div class="card-group">
    <div class="card c2 ml-2 mt-2" style="width:
18rem">
        
        <div class="card-body na">
            <h3 class="card-title">SUNNY LEONE'S
COMB</h3>
            <h4>980rs</h4>
            <button align="center" class="btn btn-
primary">
                Add to cart
            </button>

```

```

        </div>
    </div>
    <div class="card c2 ml-2 mt-2" style="width:
18rem">
        
        <div class="card-body na">
            <h3 class="card-title">HAND BAND</h3>
            <h4>780rs</h4>
            <button class="btn btn-primary float-
left">Add to cart</button>
        </div>
    </div>
    <div class="card c2 ml-2 mt-2" style="width:
18rem">
        
        <div class="card-body na">
            <h3 class="card-title">BELT</h3>
            <h4>1000rs</h4>
            <button align="center" class="btn btn-
primary">
                Add to cart
            </button>

```

```

        </div>
    </div>
    <div class="card c2 ml-2 mt-2 mr-2"
style="width: 18rem">
        
        <div class="card-body na">
            <h3 class="card-title">FACE GEL</h3>
            <h4>1300rs</h4>
            <button align="center" class="btn btn-
primary">
                Add to cart
            </button>
        </div>
    </div>
</div>
{%endblock%}
</div>
<script>
    window.watsonAssistantChatOptions = {
        integrationID: "bc30e1f2-df6f-4ca4-a430-
4925b339dd36", // The ID of this integration.
        region: "au-syd", // The region your integration
is hosted in.
        serviceInstanceID: "e0026043-09be-4231-8e79-
09ea7ca8992f", // The ID of your service instance.
    }
</script>

```



```

        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>
</body>
</html>

```

CSS

```

@import
url(//fonts.googleapis.com/css?family=Lato:300:400);

body {
    margin: 0;
}

h1 {
    font-family: "Lato", sans-serif;
    font-weight: 300;
    letter-spacing: 2px;
    font-size: 48px;
}

p {

```

```
font-family: "Lato", sans-serif;
letter-spacing: 1px;
font-size: 14px;
color: #333333;
}
```

```
.header {
  position: relative;
  background: linear-gradient(
    60deg,
    rgba(84, 58, 183, 1) 0%,
    rgba(0, 172, 193, 1) 100%
  );
  color: white;
}
```

```
.inner-header {
  height: 65vh;
  width: 100%;
  margin: 0;
  padding: 0;
}
```

```
.flex {
  /*Flexbox for containers*/
  display: flex;
  justify-content: center;
  align-items: center;
  text-align: left;
}
```

```
.waves {
```

```

    position: relative;
    width: 100%;
    height: 15vh;
    margin-bottom: -12px; /*Fix for safari gap*/
    min-height: 100px;
    max-height: 150px;
}

/* Animation */

.parallax > use {
    animation: move-forever 25s cubic-bezier(0.55, 0.5,
0.45, 0.5) infinite;
}
.parallax > use:nth-child(1) {
    animation-delay: -2s;
    animation-duration: 7s;
}
.parallax > use:nth-child(2) {
    animation-delay: -3s;
    animation-duration: 10s;
}
.parallax > use:nth-child(3) {
    animation-delay: -4s;
    animation-duration: 13s;
}
.parallax > use:nth-child(4) {
    animation-delay: -5s;
    animation-duration: 20s;
}
@keyframes move-forever {
    0% {

```

```

        transform: translate3d(-90px, 0, 0);
    }
    100% {
        transform: translate3d(85px, 0, 0);
    }
}
/*Shrinking for mobile*/
@media (max-width: 768px) {
    .waves {
        height: 40px;
        min-height: 40px;
    }
    .content {
        height: 30vh;
    }
    h1 {
        font-size: 24px;
    }
}

.bg-custom {
    background-color: white;
}
.text-custom {
    color: black;
}
.h1 {
    font-size: 350%;
}
.h2 {
    font-size: 350%;
}

```

```
.w-custom {  
  width: 350px;  
}  
.h-custom {  
  height: 350px;  
}  
.mt-custom {  
  margin-top: 40px;  
}  
.mb-custom {  
  margin-bottom: 40px;  
}  
  
.border {  
  border-style: solid;  
  border-width: thick;  
  border-radius: 20px;  
}  
.pl-custom {  
  padding-left: 15px;  
}  
  
.container-fluid {  
  background: white;  
  color: black;  
}  
.pt-custom {  
  padding-top: 20px;  
}  
  
.text {  
  font-family: Georgia, "Times New Roman", Times, serif;
```

```

}
.name {
  color: black;
  font-weight: bolder;
  font-style: oblique;
  margin-top: 10px;
  font-weight: bolder;
}
.font-custom {
  font-family: serif;
}
.border-custom:hover {
  border-color: blue;
  border-width: 3px;
  transform: scale3d(1, 1, 1);
  transition: transform 0.5s;
  transition: color 0.25s;
}
.form-control:focus {
  border-color: lightblue;
  box-shadow: 0px 1px 1px rgba(0, 0, 0, 0.075) inset,
    0px 0px 8px rgba(255, 100, 255, 0.5);
}
.btn-cus {
  background: black;
  color: white;
  font-size: larger;
  width: 320px;
  height: 50px;
}
.fl {
  float: right;
}

```

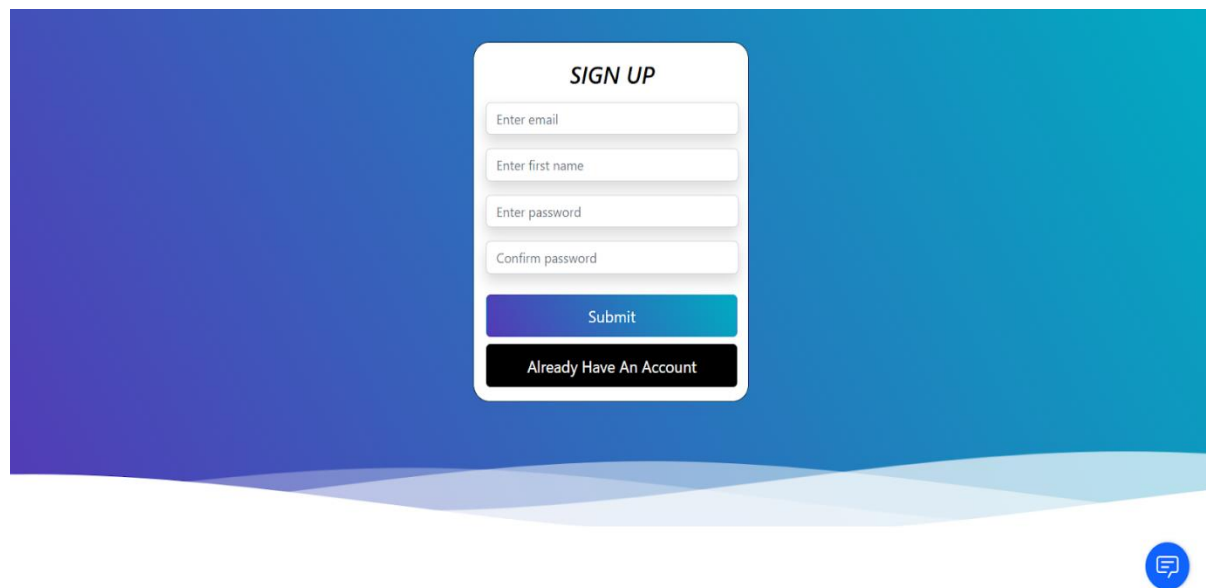
```

    text-align: end;
    color: black;
    text-decoration: none;
    text-align: right;
}
.btn-custom {
    background: linear-gradient(
        60deg,
        rgba(84, 58, 183, 1) 0%,
        rgba(0, 172, 193, 1) 100%
    );
    color: white;
    font-size: larger;
    width: 320px;
    height: 50px;
}
button:hover {
    color: white;
}
a {
    text-decoration: none;
    color: white;
}
a:hover {
    color: black;
}

```

## SCREENSHOTS

**Register page**



**SIGN UP**

Enter email

Enter first name

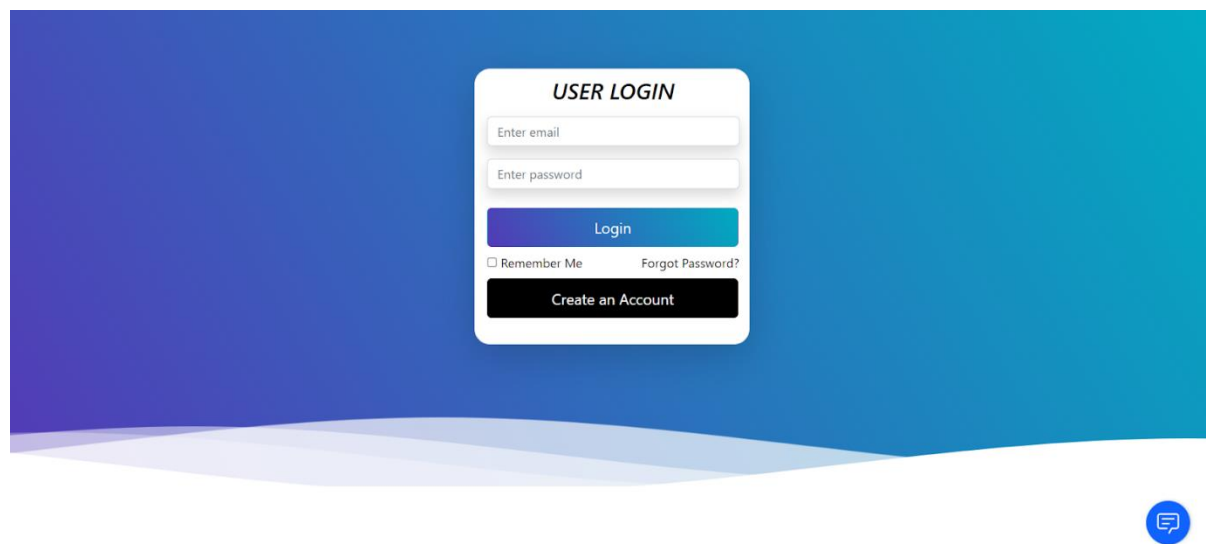
Enter password

Confirm password

Submit

Already Have An Account

## Login Page



**USER LOGIN**

Enter email

Enter password

Login


☐ Remember Me      [Forgot Password?](#)

Create an Account




## Home page


[Home](#) [Cart](#) [Logout](#)




**MEN'S CORNER**  
[Access More](#)



**CHILDREN PICKS**  
[Access More](#)




**WOMEN'S CORNER**  
[Access More](#)




**ACCESSORIES**  
[Access More](#)


### MEN'S CORNER




**PUMA TSHIRT**  
200rs  
[Add to cart](#)




**ADIDAS TSHIRT**  
400rs  
[Add to cart](#)




**BENZO TSHIRT**  
100rs  
[Add to cart](#)




**HIFI TSHIRT**  
300rs  
[Add to cart](#)




**PETER ENGLAND**  
980rs  
[Add to cart](#)



**LEVI'S**  
780rs  
[Add to cart](#)



**RAZE SHIRT**  
1000rs  
[Add to cart](#)



**PRINTED SHIRT**  
1300rs  
[Add to cart](#)


## Order Page

T-Shirt

Wallet

Belt

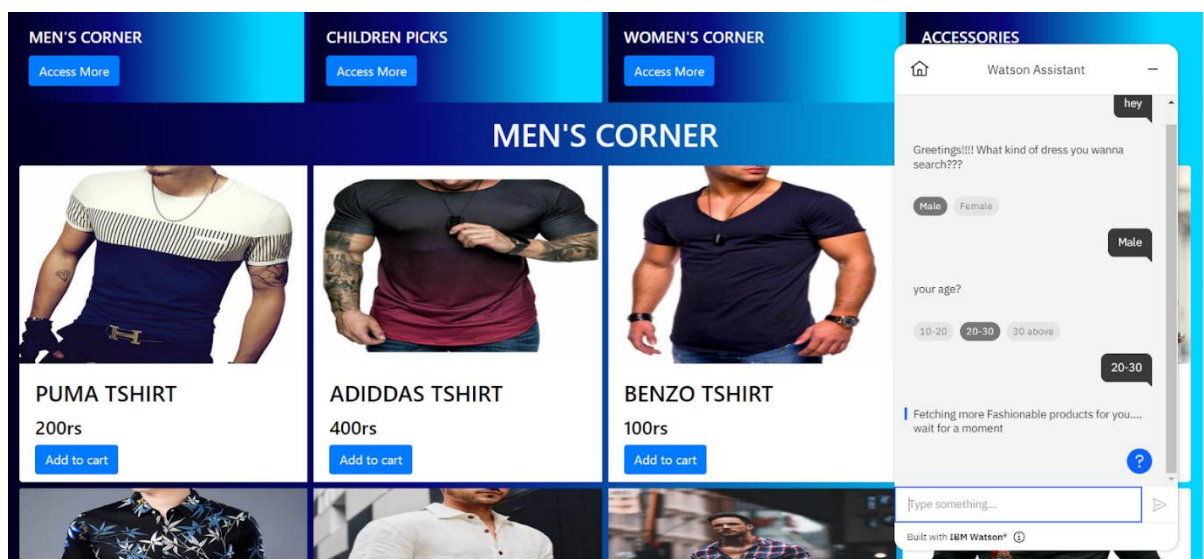
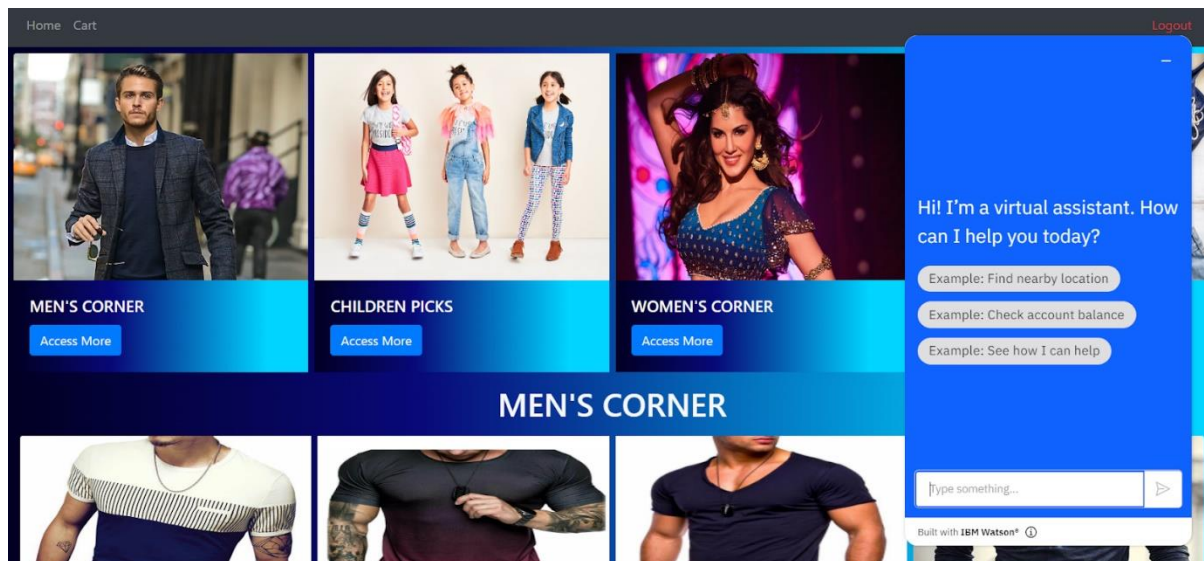
Shoes



Order Details

[Confirm Order](#)

## Integrate Chat Bot



## 13.2 GITHUB & PROJECT DEMO LINK

- Our GitHub Repository Direct Link  
<https://github.com/IBM-EPBL/IBM-Project-8027-1658906531>