SMART FASHION RECOMMENDER APPLICATION

PROJECT REPORT

Submitted by

SAJJA LOKENDRA	**LEADER**
VEMURI SUJIN	**MEMBER 1**
VELURU SAI TEJA	**MEMBER 2**
SAI BALAJI N	**MEMBER 3**

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

COMPUTER SCIENCE ENGINNERING

CHENNAI INSTITUTE OF TECHNOLOGY
SARATHY NAGAR, KANCHEEPURAM
CHENNAI- 600069

NOV- 2022

ABSTRACT

Fashion applications have seen tremendous growth and are now one of the most used programs in the e-commerce field. The needs of people are continuously evolving, creating room for innovation among the applications. One of the tedious processes and presumably the main activities is choosing what you want to wear. Having an AI program that understands the algorithm of a specific application can be of great aid.

We are implementing such a chat bot, which is fed with the knowledge of the application's algorithm and helps the user completely from finding their needs to processing the payment and initiating delivery. It works as an advanced filter search that can bring the user what they want with the help of pictorial and named representation.

The application also has two main user interfaces - the user and the admin. The users can interact with the chat bot, search for products, order them from the manufacturer or distributor, make payment transactions, track the delivery, and so on. The admin interface enables the user to upload products, find how many products have been bought, supervise the stock availability and interact with the buyer regarding the product as reviews.

The rapid progress of computer vision, cloud computing and artificial intelligence combined with the current growing urge for online shopping systems opened an excellent opportunity for the fashion industry. As a result, many studies worldwide are dedicated to modern fashion related applications such as virtual try-on and fashion synthesis.

Traditionally, keywords are used to retrieve images, but such methods require a lot of annotations on the image data, which will lead to serious problems such as inconsistent, inaccurate, and incomplete descriptions, and a huge amount of work.

However, the accelerated evolution speed of the field makes it hard to track these many research branches in a structured framework. Such hierarchical application-based multi-label classification of studies increases the visibility of current research, promotes the field, provides research directions, and facilitates access to related studies.

ACKNOWLEDGEMENT

A Nalayathiran project of this magnitude and nature requires kind cooperation and support from many, for successful completion. We wish to express our sincere thanks to all those who were involved in the completion of this Nalayathiran project.

Our sincere thanks to Our Beloved Secretary and Correspondent, Dr P. SRIRAM, for his sincere endeavor in educating us in his premier institution.

We also express our appreciation and gratefulness to Our Principal Dr K. MANI, M.E., PhD, who helped us in the completion of the project.

We wish to convey our thanks and gratitude to our head of the department, Dr M. HELDA MERCY, M.E., PhD, Department of Information Technology, for her support and for providing us ample time to complete our project.

We express our indebtedness and gratitude to our staff in charge, Mrs Gunavathie M A, M.Tech.,(PhD), Assistant Professor Department of Information Technology for his guidance throughout our project.

Last, we thank our parents and friends for providing their extensive moral support and encouragement during the project

Project Report Format

1. INTRODUCTION *1.1 Project Overview*

Instead of searching for products in the search bar and navigating to individual products to find required preferences, this project leverages the use of chatbots to gather all required preferences and recommend products to the user. The solution is implemented in such a way as to improve the interactivity between customers and applications. The chatbot sends messages periodically to notify offers and preferences. For security concerns, this application uses a token to authenticate and authorize users securely. The token has encoded user id and role. Based on the encoded information, access to the resources is restricted to specific users. User is divided base don their roles. The roles comprises of admin and user. Admin is provided access with adding new products, update a product track user details. Users of the application get periodic recommendation via chatbot base don their preference and search. The main purpose of this application is to make process of searching, filtering and ordering of products simple and quickly that enhances the overall user experience. The chatbot is trained by providing different categories of product information and its related details.

1.2 Purpose of the Project

Fashion applications have seen tremendous growth and are now one of the most used programs in the e-commerce field. The needs of people are continuously evolving, creating room for innovation among the applications. One of the tedious processes and presumably the main activities is choosing what you want to wear. Having an AI program that understands the algorithm of a specific application can be of great aid. We are implementing such a chatbot, 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 chatbot, 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. In E-commerce websites, users need to search for products and navigate across screens to view the product, add them to the cart, and order products. The smart fashion recommender application leverages the use of a chat bot to interact with the users, gather information about their preferences, and recommend suitable products to the users. This application has two predefined roles assigned to the users. The roles are customer and admin. The application demands redirection of the user to the

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

2. LITERATURE SURVEY *2.1 Existing Problem*

On 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 chatbot to interact with the users, gather information about their preferences, and recommend suitable products to the users. This application has two predefined roles assigned to the users. The roles are customer and admin. The application demands redirection of the user to the appropriate dashboard based on the assigned role. Admin should be able to track the number of different products and admin should be assigned the responsibility of creating products with appropriate categories. The user should be able to mention their preferences using interacting with chatbots. The user must receive a notification on order confirmation/failure. The chatbot must gather feedback from the user at the end of order confirmation. The main objective of this application is to provide better interactivity with the user and to reduce navigating pages to find appropriate products.

2.2 References

i) **Paper Title: A COMPREHENSIVE REVIEW ON ONLINE FASHION RECOMMENDATION**
Publication: December 2020

Author name: Samit Chakraborty

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

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

ii) Paper Title: Fashion Recommendation Systems

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

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

An effective recommendation system is needed in e-commerce platforms where there are many options available to sort, order, and effectively communicate to user’s pertinent product content or information.

Fast fashion retailers have paid a lot of attention to image-based fashion recommendation systems (FRSs), which offer customers a customised purchasing experience.

There aren't many academic studies on this subject, despite its enormous potential. The studies that are now accessible do not

conduct a thorough analysis of fashion recommendation systems and the accompanying filtering methods.

This review also looks at many potential models that might be

used to create future fashion suggestion systems. **iii) Paper Title:**

A Review on Clothes Matching and Recommendation System

Based on User Attributes.

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

Methodology: It's crucial to dress adequately while venturing out into the real world. The confidence of the individual is raised and a very positive impression is made when they are dressed appropriately in clothing that exhibits some degree of style and is worn in a way that complies with societal norms. The goal of the study is to make it easier for customers to locate the best-fitting outfits by taking into account fine elements like style, patterns, colors, and textures, as well as user characteristics like age, skin tone, and favorite colors. It seeks to assist the user in organizing 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 compliments their personal style.

In order to create a robust system that discovers the user's matching outfits and provides recommendations, an in-depth analysis of numerous systems that are built for various aspects is undertaken in this research. Systems created to propose clothing using various methodologies have been researched, with both their benefits and drawbacks highlighted.

It has also been investigated how to make clothing detecting systems user-friendly while accepting feedback from the user.

iv) Paper Title: Individualized fashion recommender system
Year: 10 October 2020

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

Methodology: This design seeks to use an image of a product provided by the stoner as input to prompt recommendations because people frequently see things that they're interested in and tend to look

for products that are similar to those. We reuse the Deep Fashion Dataset (DFD) photos using neural networks, and we generate the final suggestions using a closest neighbor backed recommender.

v) Paper Title: Image-based fashion recommender system. Publication: Year (2021).

Author name: Shaghayegh Shirkhani.

Methodology: Collaborative filtering, the iterative filtering process, matrix factorization, and content-based systems. Systems for collaborative filtering make product recommendations based on user similarity metrics and/or by grouping things from similar users' purchases.

Despite the variety of collaborative filtering methods, many widely used systems can be distilled down to just two steps: 1. Seek out users who have similar rating tendencies to the active user (the user whom the prediction is for). 2. To establish a prediction for the active user, utilize the ratings from the users who shared your interests in step one.

2.3 Problem statement

Problem Statement Definition

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love. A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face.

Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

How to write a problem statement?

A good problem statement can be created by identifying and answering several questions related to the problem,

1. Identify the Problem
2. Begin were statement with where ideal situation
3. Describe current gaps
4. State the consequence of the problem

5. Propose addressing the problem

KEY ELEMENTS OF PROBLEM STATEMENT:

- Reality - It will also describe when and where the problem was identified.
- Consequences - Problem statements should identify the consequences of the problem.
- Proposal - The proposal section of a problem statement may contain several possible solutions to the problem

I am	Describe customer with 3-4 key characteristics - *who are they?*	Describe the customer and their attributes here
I'm trying to	*List their outcome or "job" the care about - what are they trying to achieve?*	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way – *what bothers them most?*	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists – *what needs to be solved?*	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view – *how does it impact them emotionally?*	Describe the emotions the result from experiencing the problems or barriers

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	I am a student	I am trying to buy a formal dress for college event	I can't find perfect match	Either the brand or colour doesn't suit	I don't have any choice in choosing right brand.
PS-2	I am a bride	Looking for wedding and reception dress	Could not able to choose right dress	Availability of wide variety of products	Like I'm overthinking regarding how my dress should looks like.
PS-3	I am an Average Man	Buy clothes to attend family function.	could not find dress of my desire	Product price not within the budget	Give up and move for other ecommerce site

3. IDEATION & PROPOSED SOLUTION

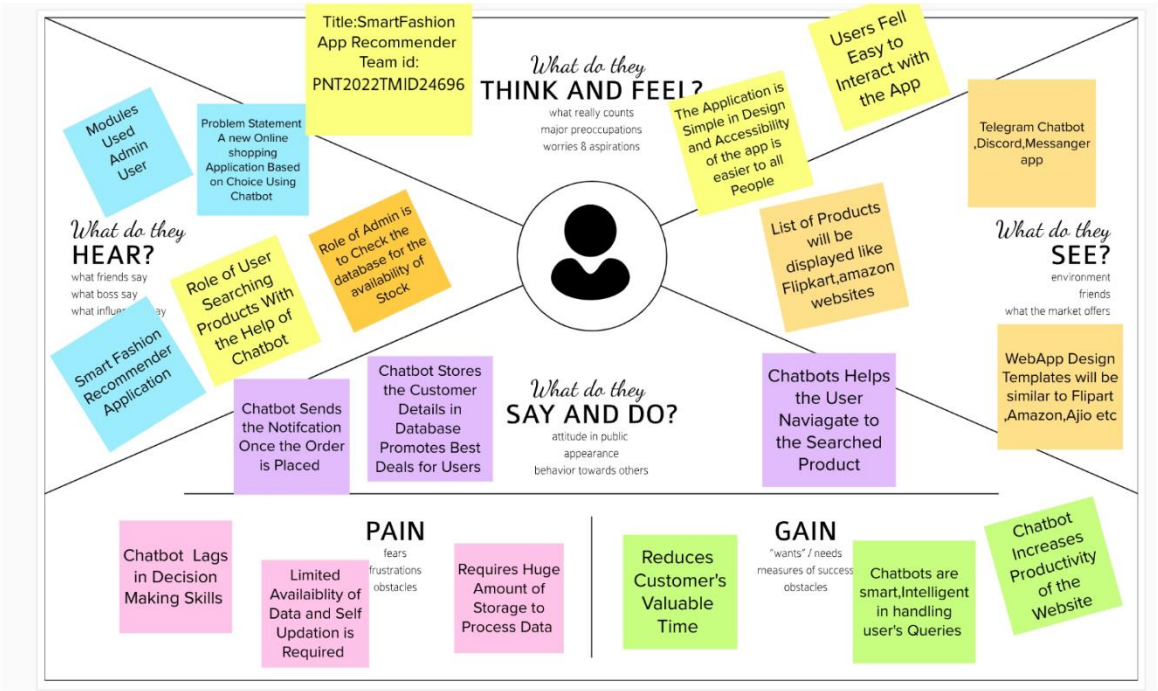
3.1 Empathy Map

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it.

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

Benefits of empathy map

- *It is simple and quick to complete an empathy map. There is no need to spend a lot of time or money on the undertaking. All you require is a team and basic tools. As an alternative, you can use expensive but cutting-edge software, even on a tight budget.*
- *The visualizing technique allows for widespread participation. You invite managers, designers, tech and non-tech experts, and product owners. The more opinions and ideas you have, the better.*
- *It aids creators in expanding their thinking and understanding that their ideal product vision may vary from the viewpoint of the customers.*
- *These resources are all-purpose and can be used in any field, sector, or type of business.*
- *Companies that engage in empathy mapping outperform those that do not.*



3.2 Brainstorming- Idea prioritization

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving.

Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

There are some brainstorming approaches:

Step 1: Prepare the Group

Step 2: Present the Problem

Step 3: Guide the Discussion

Step 1: Prepare the Group

How much background knowledge or planning is required for your team to generate problem-solving ideas? It's vital to keep in mind that preparation is necessary, but too much of it might hinder or even ruin a brainstorming session.

Once everyone is present, designate one person to take notes on the suggestions made during the session.

It's difficult to record and contribute at the same time, thus this individual shouldn't necessarily be the team manager.

Use a computer with a data projector, flip charts, or whiteboards to post notes where everyone can see them.

Step 2: Present the Problem

Lay up any requirements that must be met as well as the problem that you are trying to solve. Make it crystal clear that the goal of the meeting is to create as many ideas as you can.

Allow everyone plenty of quiet time to come up with as many original ideas as they can at the beginning of the session.

Ask them to share or present their thoughts after that, while ensuring that everyone has an equal chance to contribute.

Step 3: Guide the Discussion

Start a group conversation once everyone has given their views a chance to be heard in order to develop existing ideas and generate new ones.

One of the most beneficial features of group brainstorming is building on others' ideas.

Encourage everyone—even the most reserved individuals—to participate and develop ideas, and forbid anybody from critiquing others' ideas.

If you have any ideas, you should share them as the group facilitator, but otherwise focus your time and efforts on helping your team and facilitating the conversation.

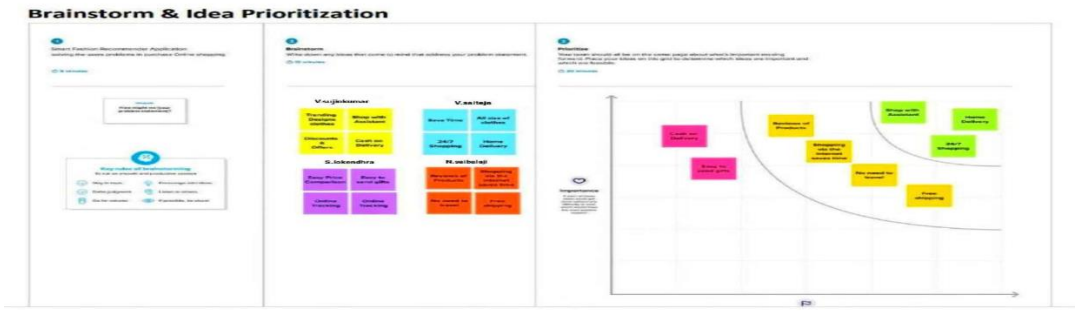
Keep each conversation to itself and bring the group back on track if it wanders off.

Don't spend too much time thinking along the same lines. Make sure to come up with lots of different ideas and thoroughly examine each one.

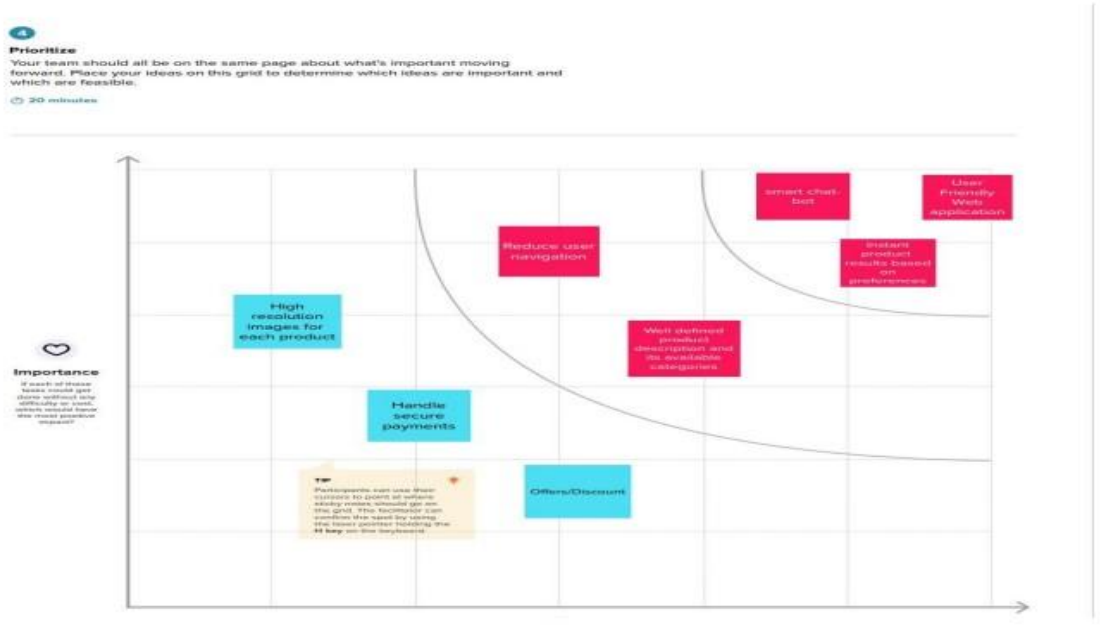
Give a team member the freedom to pursue an idea alone if they need to "tune out" for it.

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

Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



Benefits of using brainstorming:

- *There are several advantages to brainstorming.*

- *Building engagement, dedication, devotion, and passion through brainstorming.*
- *People's creative talents are stimulated and unlocked by participating in the workshops.*
- *Because participants are solicited for their input and ideas during a brainstorming session, this activity also boosts self-esteem.*
- *You can foster more teamwork and cooperation using brainstorming.*
- *People who participate in brainstorming exercises together are more likely to form stronger bonds and communicate more effectively.*
- *The main benefit is that you will generate many excellent ideas, some of which may even change the course of the company.*
- *I've found that brainstorming activities have produced a number of excellent ideas to..*
- *Encourages Critical Thinking*
- *When you practice brainstorming as a group, you take team ownership of a campaign, product or event.*
- *This means that one person isn't left feeling like he is carrying the workload for the entire company, and also cultivates a feeling of team ownership.*

3.3 Proposed Solution

Proposed solution Definition :

Proposed Solution refers to the technical response that the Implementation agency will offer in response to the Project's requirements and objectives.

How to write a problem statement

1. *Describe how things should work.*
2. *Explain the problem and state why it matters.*
3. *Explain your problem's financial costs.*
4. *Back up your claims.*
5. *Propose a solution.*
6. *Explain the benefits of your proposed solution(s).*
7. *Conclude by summarizing the problem and solution.*

The main goal of presenting a business proposal is to provide a solution to a problem faced by a potential buyer. This section should be as comprehensive as possible, and able to address all the needs that we have pointed out in the first section.

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

3.4 Problem Solution fit

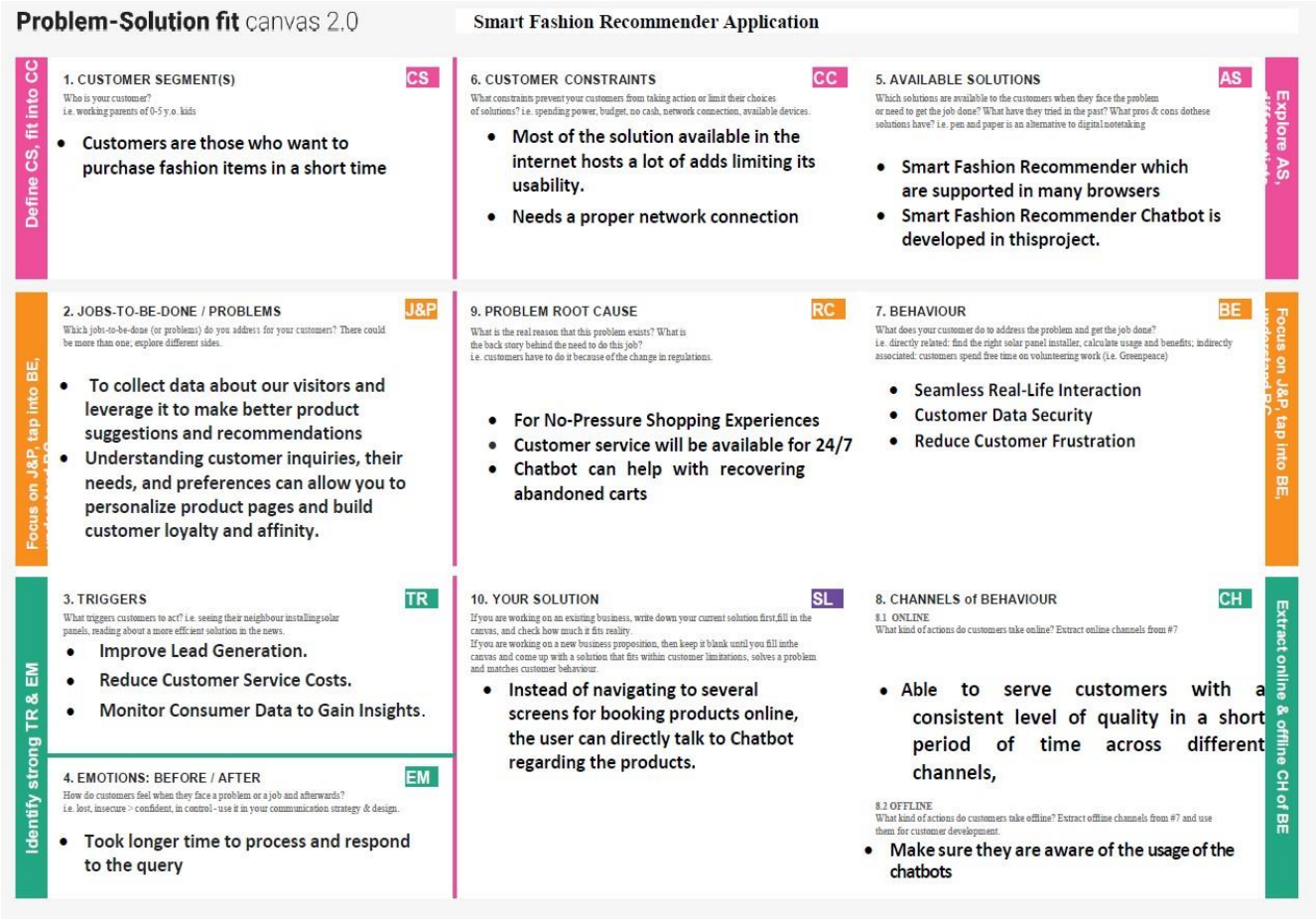
The Lean Startup, LUM (Lazy User Model), and User Experience design tenets serve as the foundation for the Problem-Solution Fit canvas. It aids in the identification of behavioural patterns by business innovators, marketers, and entrepreneurs.

- *It serves as a template for identifying solutions with the best prospects of being adopted, cutting downon testing time, and getting a clearer picture of the situation as it stands.*

My objective was to develop a tool that transforms a problem into a solution while considering client behavior and the surrounding circumstances.

- *With the help of this template, we will be able to thoroughly examine problem solving and take importantinformation into account earlier.*

- *It increases our chances of finding a product-market fit and a problem-solution fit.*



4. REQUIREMENT ANALYSIS:

4.1 Functional Requirements:

The functional requirements of the application are:

- Redirect users to their respective dashboards ● Allow admin to track sales of individual products
- Allow admin to manage orders made by a particular customer.
- Allow users to interact with the chatbot.
- Manage users' choices and charges using the chatbot.
- Promote the best deals and offers.
- Store customer details and orders.
- Send Notifications to customers if the order is confirmed. ● Collect user feedback.
- Recommend products based on user preference.
- Enable online payment features.
- Generate reports for order summary and order histories.

FR No.	Functional Requirements	Sub Registration
FR-1	Registration	Registration can be done using mobile number or gmail and needed some user information
FR-2	Login	User only log in by user id and password,Which is given during registration
FR-3	Delivery confirmation	Confirmation via email and phone number
FR-4	Assistance	Bot is integrated with the application to make the usability simple

4.2 Non-Functional Requirements

Following are the Non-Functional requirements of the proposed solution.

Performance Requirements

- The response should not take longer than 5 seconds to appear on the client side.
- The client application should lazy load images of the product to minimize network calls over ● the network.
- The responses from the server should be cached on the client side.

Security Requirements

- Credentials and secrets should be stored securely and should not be leaked.
- Secured connection HTTPS should be established for transmitting requests and responses
- between client and server.
- The system has different roles assigned to a user and every user has access constraints.
- User access token should be valid for a shorter period and needs to be refreshed periodically.
- Clients should implement mechanisms to prevent XSS attacks.
- The server should restrict access to the resources for the particular client domain.

Error Handling

The system should handle expected as well as unexpected errors and exceptions to avoid termination of the program.

Appropriate error messages should be generated and displayed to the client.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	A user-friendly interface with chat bot to make usability efficient
NFR-2	Security	Secured connection HTTPS should be established for transmitting requests and responses
NFR-3	Reliability	The system should handle excepted as well as unexpected errors and exceptions to avoid termination of the program
NFR-4	Performance	The system shall be able to handle multiple requests at any given point in time and generate an appropriate response.
NFR-5	Availability	It is a cloud based web application so user can access without any platform limitations ,just using a browsers with a internet connection is enough for use the application
NFR-6	Scalability	It has a quick request and response time, high throughput, enough network resources and so on.

5. PROJECT DESIGN

5.1 Data Flow Diagram

A data flow diagram (DFD) is a graphical or visual representation that describes how data is moved through an organization's operations using a standardized set of symbols and notations.

In formal methodologies like the Structured Systems Analysis and Design Method, they are frequently included. DFDs may initially resemble flowcharts or the Unified Modeling Language.

1. Advantages of data flow diagram:

○ It aids in describing the boundaries of the system.

○ It is beneficial for communicating existing system knowledge to the users.

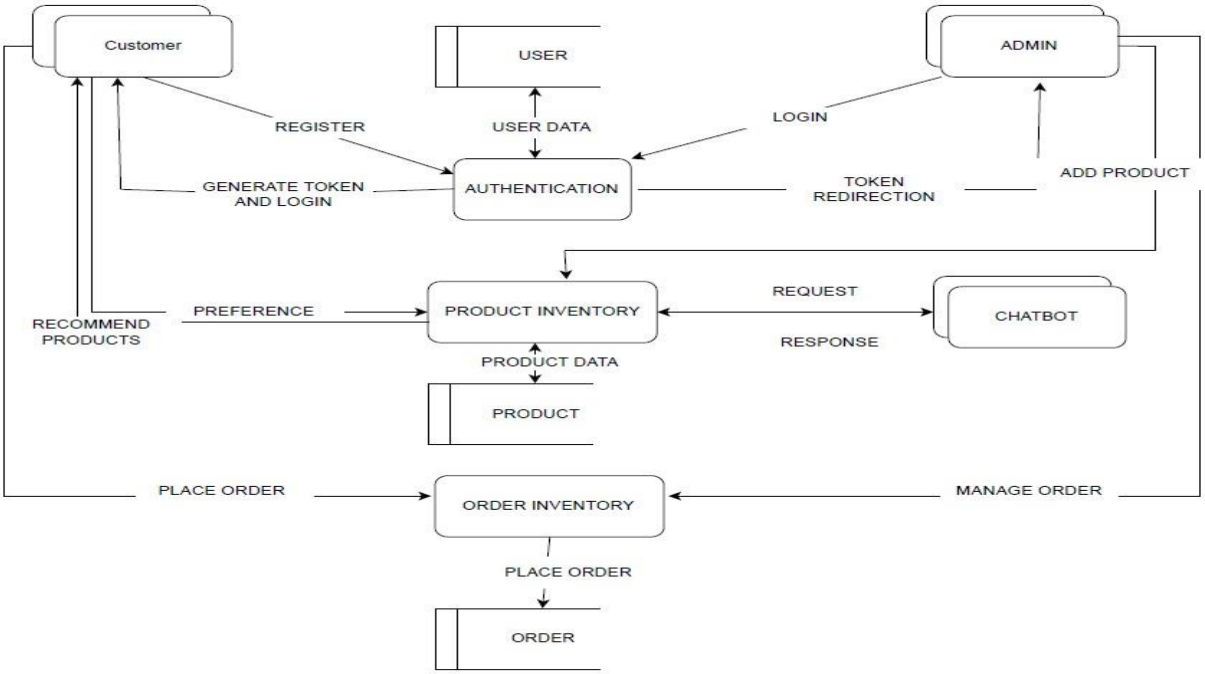
○ A straightforward graphical technique which is easy to recognise.

○ DFDs can provide a detailed representation of system components.

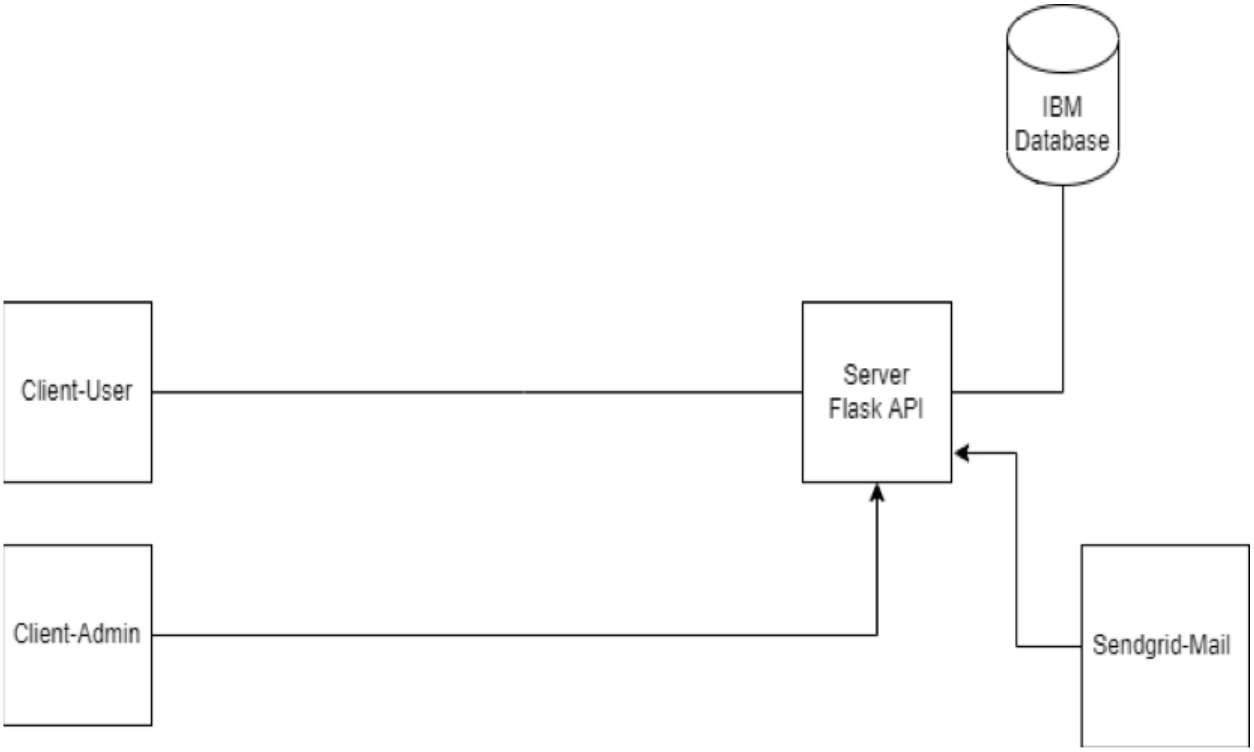
○ It is used as the part of system documentation file.

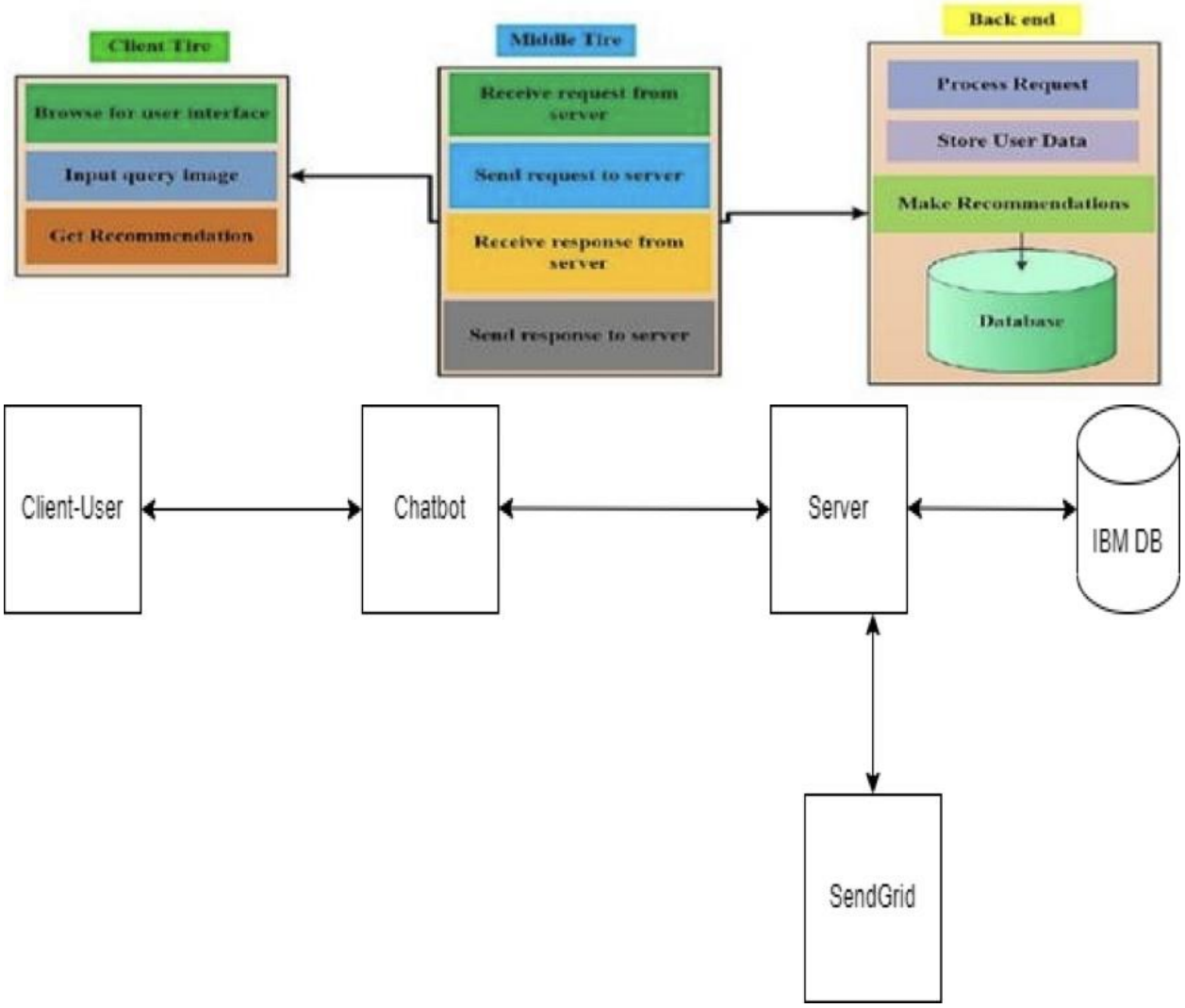
○ DFDs are easier to understand by technical and nontechnical audiences ○

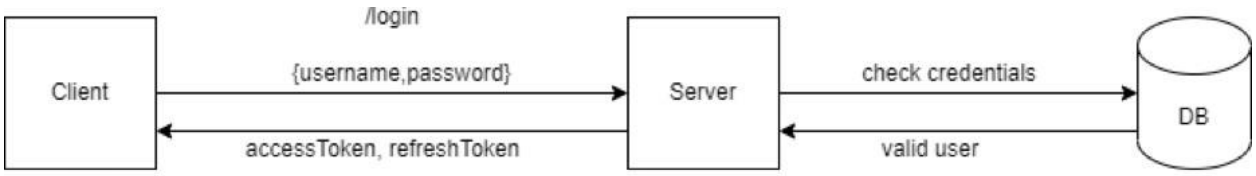
It supports the logic behind the data flow within the system.



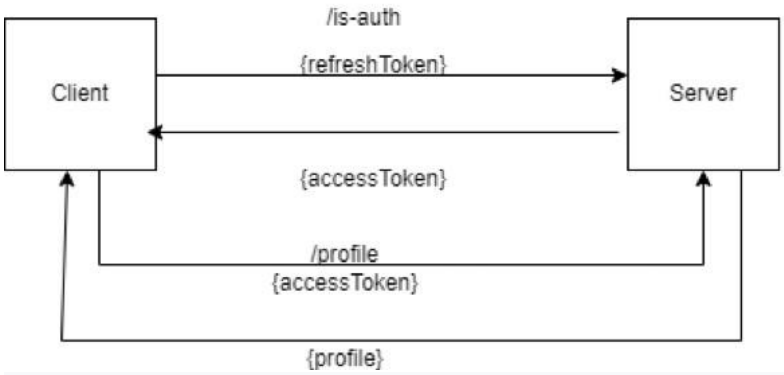
5.2 Solution & Technical Architecture

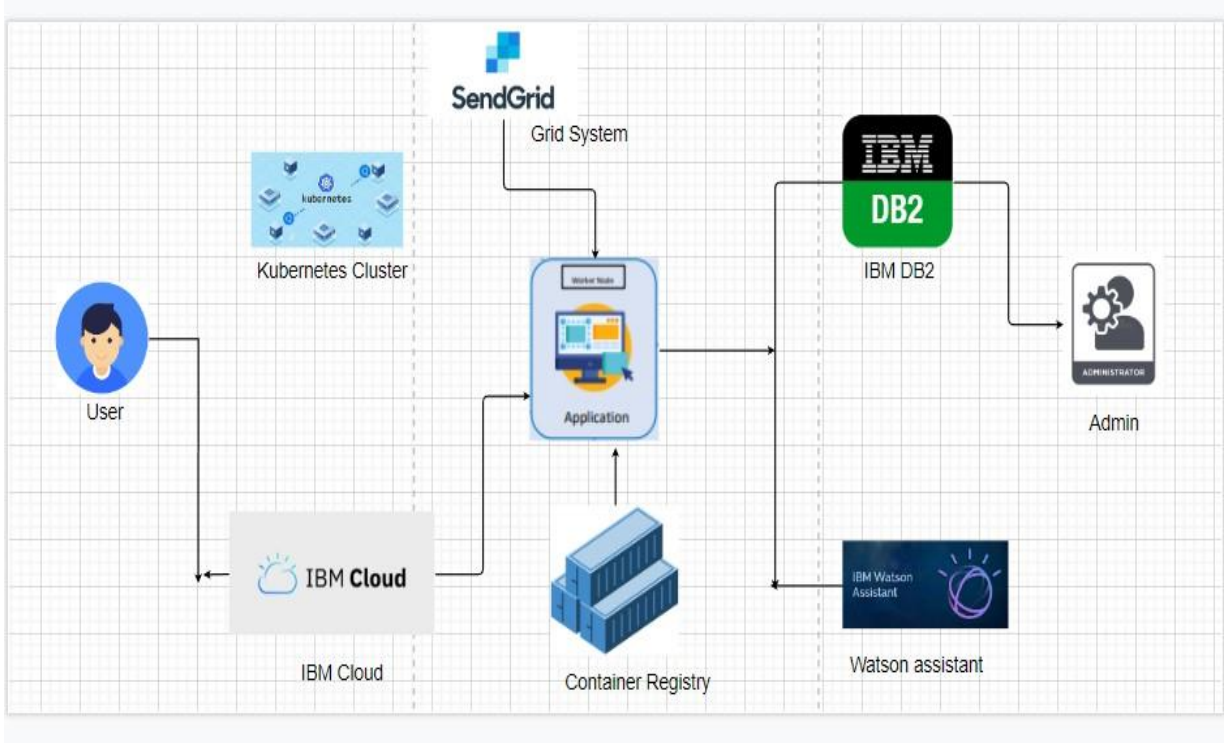



Client Tire
Browse for user interface
Input query image
Get Recommendation
Middle Tire
Receive request from server
Send request to server
Receive response from server
Send response to server
Back end
Process Request
Store User Data
Make Recommendations
Database
Client-User
Chatbot
Server
IBM DB
SendGrid

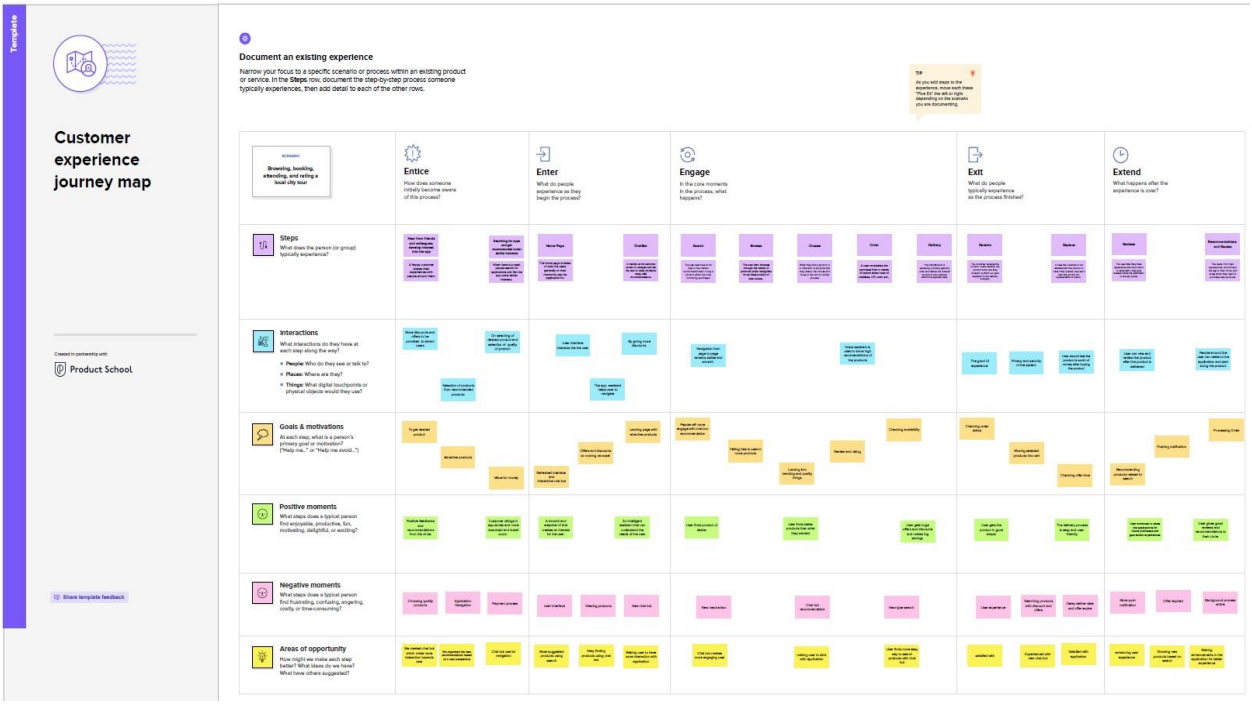


On New Session





5.3 User Stories



6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Use the below template to create product backlog and sprint schedule

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Panel	USN-1	The user will login into the website and go through the products available on the website	20	High	Sajja lokendra Vemuri sujin kumar Sai Balaji N Veluru sai teja
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	Sajja lokendra Vemuri sujin kumar Sai Balaji N Veluru sai teja
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	Sajja lokendra Vemuri sujin kumar Sai Balaji N Veluru sai teja
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	Sajja lokendra Vemuri sujin kumar Sai Balaji N Veluru sai teja

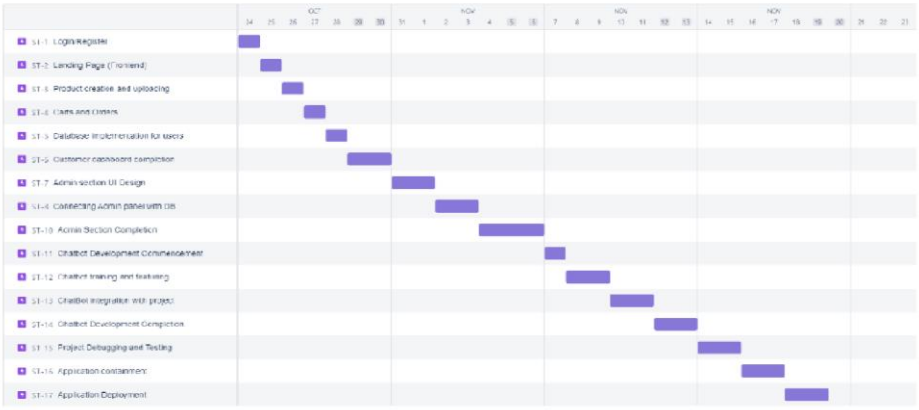
6.2 Sprint Delivery Schedule

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		29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		07 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

6.3 Reports from JIRA

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



Components and technology stack

S.NO	Component	Technology	Description
1.	User Interface	HTML, CSS, JavaScript/Angular JS / React JS.	User can interact with the application through chatbot for good Human-computer interface.
2.	Application Logic-1	Java Python	The application will have the login /signup page where the user can login into the main dashboard or they can register into the application.
3.	Application Logic-2	IBM Watson STT service	The application contains a Chatbot where the user needs to give their details like ✓ gender, ✓ age, ✓ type of product these were they wish to buy using Watson assistant through chatbot.
4.	Application Logic-3	IBM Watson Assistant	User's will get the recommendations based on their interests, can get the details about offers, discounts and chatbot will send a notification to customers if the order is confirmed.
5.	Database	MySQL, NoSQL,	Customer's details and order are stored in the database and whenever we can be fetch and retrieve data from database.
6.	Cloud Database	IBM DB2, IBM Cloud account	With use of Database Service on Cloud, user can access all the data stored in the cloud over a network from any device and user's data are stored in a well secure manner.
7.	File Storage	IBM Block Storage or Other Storage or Other Storage Service or Local Filesystem	Previously ordered product details and other customer details can be stored in the IBM Block Storage as the data kept inside are highly protected.
8.	Infrastructure (Server/Cloud)	Local, Cloud Foundry, Kubernetes, Docker	Chatbot with updated services can be deployed in an IBM cloud by using Watson assistant.

6.4 Components and technology stack

8. TESTING

Test Cases:

A test case is a series of operations carried out on a system to see if it complies with software requirements and operates properly. A test case's objective is to ascertain whether various system features operate as anticipated and to check that the system complies with all applicable standards, recommendations, and user requirements. The act of creating a test case can also aid in identifying flaws or mistakes in the sy stem.

A test case document includes test steps, test data, preconditions and the post conditions that verify requirements.

Why test case are so important:

Writing test cases is a significant task and is regarded as one of the most crucial components of software testing. The testing team, the development team, and management all use it. We can use a test case as a starting point document if an application doesn't have any documentation.

- *In other words, test cases make clear what must be done to test a system. It provides us with the actions we take in a system, the values we provide as input data, and the anticipated outcomes when we run a certain test case.*
- *Test cases give a precise picture of the expectations that must be met.*
- *Test cases demonstrate how you addressed and tested each requirement for the product.*
- *Test cases assist new team members in quickly becoming engaged in the project, learning about your product and test management processes, and running prepared test cases when necessary.*
- *A solid foundation for developing automated scripts and adding automated testing to the QA process is detailed manual test cases.*

1.User Acceptance Testing

User acceptance testing (UAT), also called application testing or end-user testing, is a phase of software development in which the software is tested in the real world by its intended audience.

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

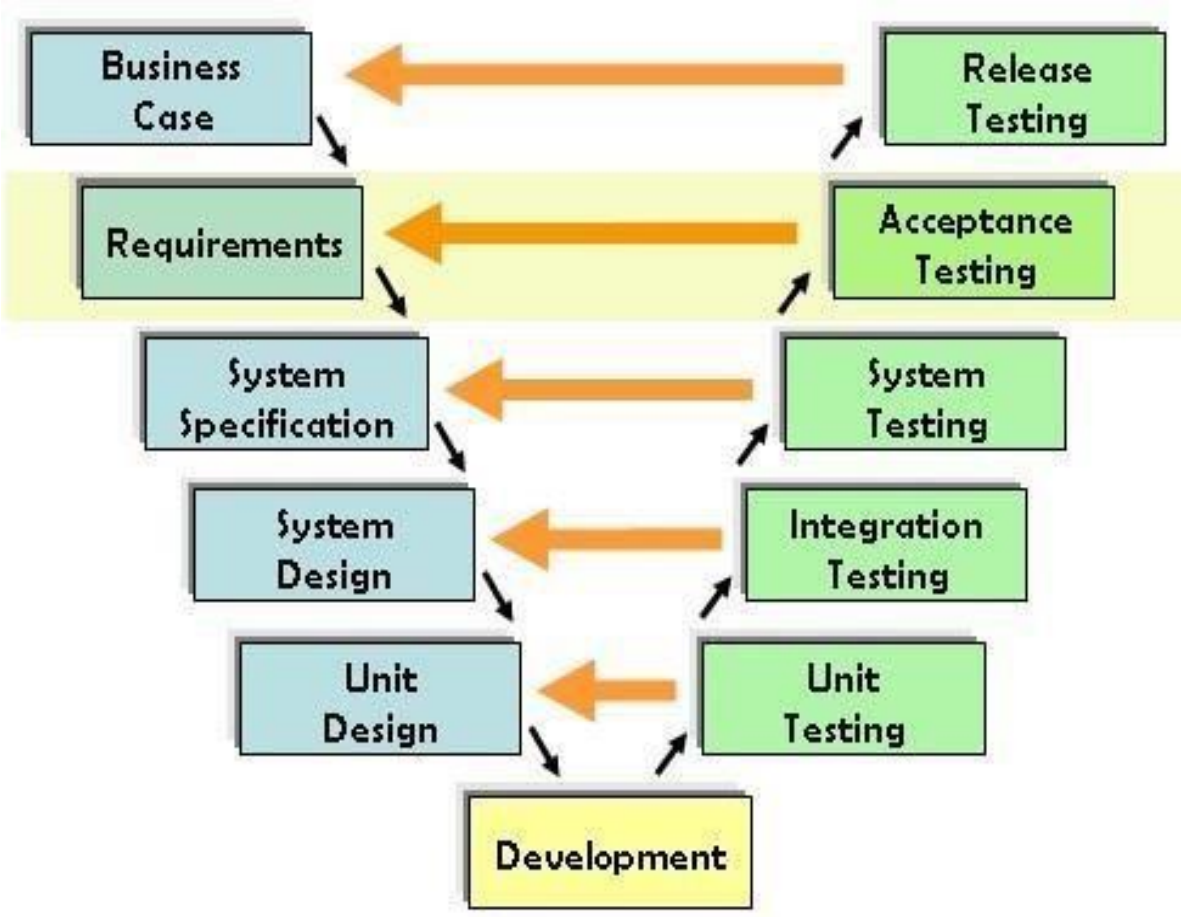


Fig 7.1) Testing architecture

UAT testing meaning can also be defined as the user methodology where the developed software is tested by the business user to validate if the software is working as per the specifications defined.

This type of testing is also known as beta testing, application testing, or more commonly end-user testing. The main Purpose of UAT is to validate end to end business flow.

It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup.

It is a kind of black box testing where two or more end- users will be involved.

1.Purpose of Document

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

project at the time of the release to User Acceptance Testing (UAT).

2. 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	4	2	3	14
Duplicate	1	0	3	0	4
External	0	1	0	1	2
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	1	1
Won't Fix	0	0	1	1	2
Totals	17	7	11	26	61

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

8. **CODING&SOLUTIONING**

```
import os
os.add_dll_directory('C:/Users/SUJINKUMAR/AppData/
Local/Programs/Python/Python39/Lib/site-
packages/clidriver/bin')
import json
from flask_session import Session
from flask import Flask, render_template, redirect,
request, session, jsonify, url_for
```

```
from datetime import datetime

import ibm_db
import ibm_db_dbi

from dotenv import load_dotenv
load_dotenv()


string_conn = "DATABASE=bludb;\
   HOSTNAME={0};\
   PORT={1};\
   SECURITY=SSL;\
   SSLServerCertificate=DigiCertGlobalRootCA.crt;\
   PROTOCOL=TCPIP;\
   UID={2};\
   PWD={3}"
string_conn =
string_conn.format(os.environ.get("HOSTNAME"),os.en
viron.get("PORT"),os.environ.get("UID"),os.environ.get(
"PWD"))
con = ibm_db.connect(string_conn,'','')

pconn = ibm_db_dbi.Connection(con)

app = Flask(__name__, template_folder="templates")

app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
Session(app)

app.secret_key = 'smart fashion recommender
application'

@app.route("/", methods = ['POST','GET'])
def Index():
   my_cursor = pconn.cursor()
   sql = 'select * from products'
   my_cursor.execute(sql)
   account = my_cursor.fetchall()
   print(account)
   return render_template("index.html", products =
account)

@app.route("/signup",methods = ['POST','GET'])
def signup():
   if 'user' in session:
      return redirect(url_for('Index'))
```

```
    else:
        if request.method == "POST":
            username = request.form["username"]
            email = request.form["email"]
            password = request.form["password"]

            sql = "select * from users where username = ? or
email = ?"
            ss = ibm_db.prepare(con, sql)
            ibm_db.bind_param(ss, 1, username)
            ibm_db.bind_param(ss, 2, email)
            ibm_db.execute(ss)
            dicts = ibm_db.fetch_row(ss)

            if dicts:
                return render_template("signup.html", msg =
"user with username or email alreay exists.")
            elif request.form['password'] !=
request.form['cpass']:
                return render_template("signup.html", msg =
"Password Mismatch.")
            else:
                sql = "insert into users(username, email,
password) values (?,?,?)"
                ss = ibm_db.prepare(con, sql)
                ibm_db.bind_param(ss, 1, username)
                ibm_db.bind_param(ss, 2, email)
                ibm_db.bind_param(ss, 3, password)
                try:
                    ibm_db.execute(ss)
                    return redirect(url_for('Index'))
                except:
                    return render_template("signup.html", msg =
"Error while trying to register user. Please try again after
some time.")
        else:
            return render_template("signup.html", msg = "")

@app.route("/login", methods = ['POST','GET'])
def login():
    if 'user' in session:
        return redirect(url_for('Index'))
    else:
        if request.method == "POST":
            username = request.form["username"]
            password = request.form["password"]
```

```
        sql = "select * from users where username = ? and
password = ?"
        ss = ibm_db.prepare(con, sql)
        ibm_db.bind_param(ss, 1, username)
        ibm_db.bind_param(ss, 2, password)
        ibm_db.execute(ss)
        dicts = ibm_db.fetch_assoc(ss)
        print(dicts)
        if dicts:
            session['user'] = username
            session['time'] = datetime.now()
            session['uid'] = dicts['ID']

        print(session)
        # Redirect to Home Page
        if 'user' in session:
            return redirect(url_for('Index'))
        else:
            return render_template("login.html", msg =
"Please Check The Credentials")

    else:
        return render_template("login.html", msg = "")

@app.route("/logout", methods = ['GET'])
def logout():
    session.clear()
    return redirect(url_for('Index'))

@app.route("/orders", methods = ['GET'])
def cart():
    account = ''
    FLAG = True
    my_cursor = pconn.cursor()
    sql = "select * from data where iden = ? and type = ?"
    try:
        params = (int(session.get("uid")),0)
        my_cursor.execute(sql,params)
        account = my_cursor.fetchall()
        print(account)
    except Exception as e:
        FLAG = e
    return render_template("shopping-cart.html", data =
account, Flag = FLAG)

@app.route("/add-cart", methods = ['POST'])
def addCart():
    print(session)
```

```
    req = json.loads(request.data)
    FLAG = True

    sql = "select * from data where iden = ? and name = ?"
    ss = ibm_db.prepare(con, sql)
    ibm_db.bind_param(ss, 1, int(req['jinja']))
    ibm_db.bind_param(ss, 2, (req['name']))
    try:
       ibm_db.execute(ss)
    except Exception as e:
       FLAG = False

    dicts = ibm_db.fetch_assoc(ss)
    if dicts:
       sql = "update data set quantity = ? where ID = ?"
       ss = ibm_db.prepare(con,sql)
       ibm_db.bind_param(ss, 1, int(req['quantity']))
       ibm_db.bind_param(ss, 2, int(dicts['ID']))
       try:
          ibm_db.execute(ss)
       except:
          FLAG = False
    else:
       try:
          sql = "insert into data(name, image, price,
quantity, category, type, iden) values (?,?,?,?,?,?,?)"
          ss = ibm_db.prepare(con, sql)
          print("name")
          ibm_db.bind_param(ss, 1, req['name'])
          print("image")
          ibm_db.bind_param(ss, 2, req['image'])
          print("price")
          ibm_db.bind_param(ss, 3, req['price'])
          print("quan")
          ibm_db.bind_param(ss, 4, int(req['quantity']))
          print("cat")
          ibm_db.bind_param(ss, 5, req['category'])
          print("typ")
          ibm_db.bind_param(ss, 6, int(req['type']))
          print("jin")
          ibm_db.bind_param(ss, 7, int(req['jinja']))

          ibm_db.execute(ss)
       except Exception as e:
          FLAG = e

    if not FLAG:
```

```
        return jsonify({"success" : FLAG})
    else:
        cnt = int(req['productCount']) - int(req['quantity'])
        sql = "update ZGS46818.PRODUCTS set quantity =
? where name = ?"
        ss = ibm_db.prepare(con,sql)
        ibm_db.bind_param(ss,1, int(req['productCount']) -
int(req['quantity']))
        ibm_db.bind_param(ss,2, req['name'])

        ibm_db.execute(ss)

        return jsonify({"success" : True})

@app.route("/shop", methods = ["GET"])
def shop():
    my_cursor = pconn.cursor()
    sql = 'select * from products'
    my_cursor.execute(sql)
    account = my_cursor.fetchall()
    return render_template("shop.html", products =
account)

@app.route("/contact", methods = ["GET"])
def contact():
    return render_template("contact.html")

@app.route("/admin", methods = ['GET'])
def admin():
    if not ('user' in session and session['user'] == 'admin'):
        return redirect(url_for('Index'))
    my_cursor = pconn.cursor()
    sql = 'select * from products'
    my_cursor.execute(sql)
    account = my_cursor.fetchall()
    proLen = len(account)

    sql2 = 'select * from users'
    my_cursor.execute(sql2)
    users = my_cursor.fetchall()
    userLen = len(users)

    sql3 = 'select * from data'
    my_cursor.execute(sql3)
    bought = my_cursor.fetchall()
    boughtLen = len(bought)
```

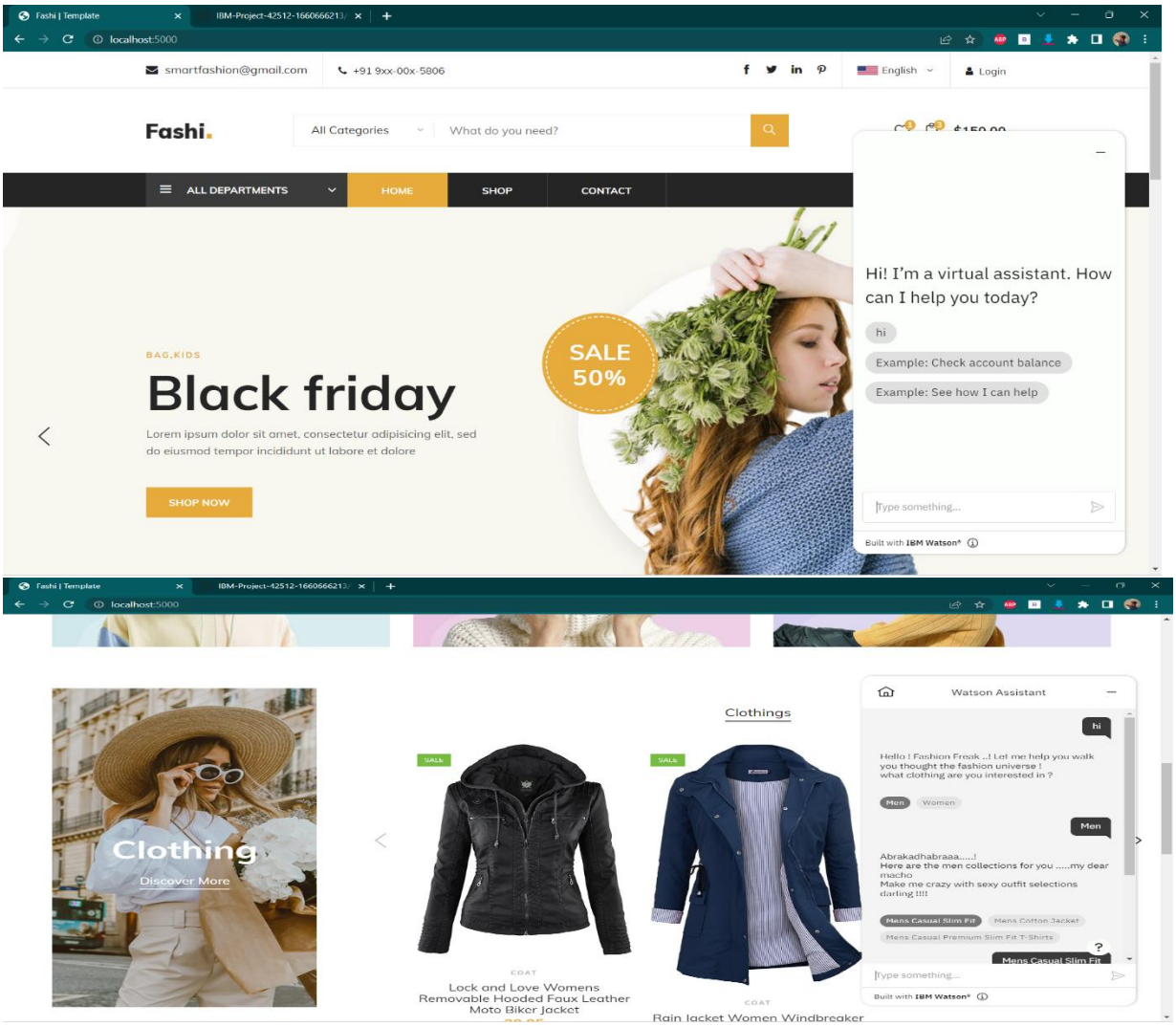
```
   return render_template('admin.html', account =
account, proLen = proLen, users = users, userLen =
userLen, bought = bought, boughtLen = boughtLen)

app.run(port=5000, debug=True)
```

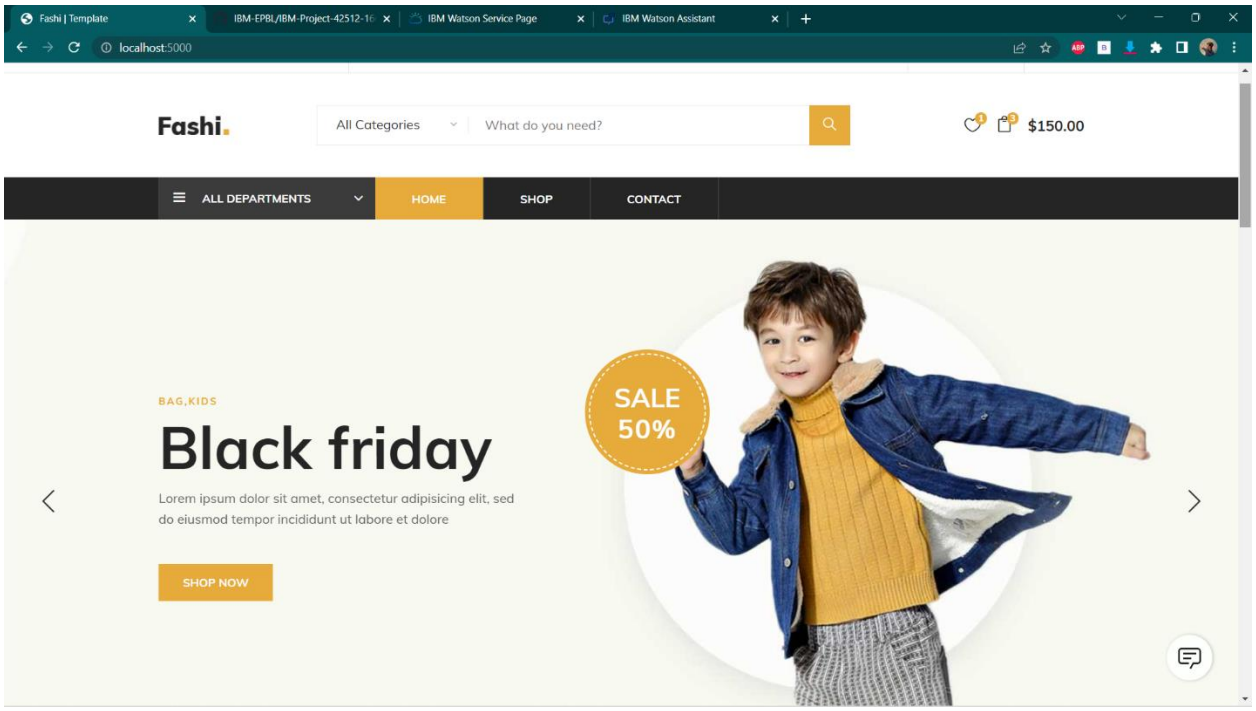
ADMIN.HTML

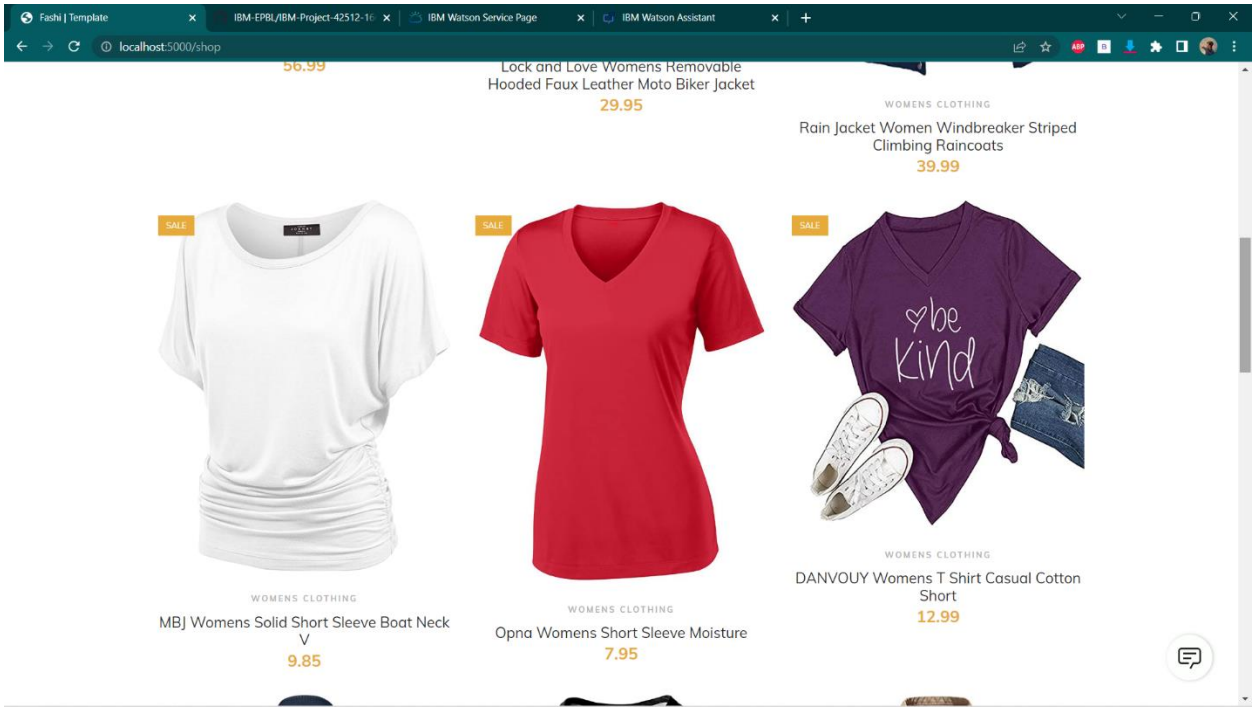
```
{% include 'home.html' %}
   <div class="conatiner d-flex justify-content-center
align-items-center flex-row gap-2 my-4">
      <div class="card">
         <h5 class="card-header">Products</h5>
         <div class="card-body">
         <h5 class="card-title">Total Number Of
Products</h5>
         <h2 class="card-title fw-bold">{{proLen}}</h2>
         </div>
      </div>
      <div class="card">
         <h5 class="card-header">Users</h5>
         <div class="card-body">
         <h5 class="card-title">Total Number Of
Users</h5>
         <h2 class="card-title fw-bold">{{userLen}}</h2>
         </div>
      </div>
      <div class="card">
         <h5 class="card-header">Bought</h5>
         <div class="card-body">
         <h5 class="card-title">Total Number Of Products
Bought</h5>
         <h2 class="card-title fw-
bold">{{boughtLen}}</h2>
         </div>
      </div>
   </div>
   <div class="cart-table">
      <table>
         <thead>
            <tr>
               <th>Image</th>
               <th class="p-name">Product Name</th>
               <th>Price</th>
```

```
                <th>Quantity</th>
                <th>Total</th>
              </tr>
            </thead>
            <tbody>
              {% for product in account %}
                <tr>
                  <td class="cart-pic first-row"><img src='{{
product[3] }}' alt=""></td>
                  <td class="cart-title first-row">
                    <h5>{{ product[1] }}</h5>
                  </td>
                  <td class="p-price first-row">{{ product[3]
}}</td>
                  <td class="qua-col first-row">
                    <div class="quantity">
                      <div class="pro-qty">
                        <input type="text" value='{{
product[5] }}'>
                      </div>
                    </div>
                  </td>
                  <td class="total-price first-row">${{
product[4] }}</td>
                </tr>
              {% endfor %}
            </tbody>
          </table>
        </div>
{% include 'footer.html' %}
</body>
</html>
```

Screenshots




Fashi | Template
IBM-EPBL/IBM-Project-42512-16
IBM Watson Service Page
IBM Watson Assistant
localhost:5000
Fashi.
All Categories
What do you need?
$150.00
ALL DEPARTMENTS
HOME
SHOP
CONTACT
BAG,KIDS
Black friday
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore
SHOP NOW
SALE 50%


Fashi | Template
IBM-EPBL/IBM-Project-42512-16
IBM Watson Service Page
IBM Watson Assistant
localhost:5000/shop
56.99
Lock and Love Womens Removable Hooded Faux Leather Moto Biker Jacket
29.95
WOMENS CLOTHING
Rain Jacket Women Windbreaker Striped Climbing Raincoats
39.99
SALE
WOMENS CLOTHING
MBJ Womens Solid Short Sleeve Boat Neck V
9.85
SALE
WOMENS CLOTHING
Opna Womens Short Sleeve Moisture
7.95
SALE
be kind
WOMENS CLOTHING
DANVOUY Womens T Shirt Casual Cotton Short
12.99