PNT2022TMID01360

SMART FASHION RECOMMENDER APPLICATION

A PROJECT REPORT

SUBMITT

EDBY

RAGHURAMAN.R[211419205135]

SURIYA.N[211419205164]

THARUN VIGNESH.M

[211419205171]

VINOTH RAJIV.C.S[211419205181]

TEAM ID : PNT2022TMID01370
INDUSTRY MENTOR :KRISHNA
CHAITANYAFACULTY MENTOR :
RAMA DEVI K

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Fashion is perceived as a meaningful way of self-expressing that people use fordifferent purposes. It seems to be an integral part of every person in modern societies, from everyday life to exceptional events and occasions. With the increasing number of users who prefer to use online applications to shop, it is necessary to meet up with the customer and users needs adequately.With the development of AI-powered recommendation engines, recent research from Emerald Insight also shows that recommendation engines are becoming morecommon in the areas of fashion and apparel. Multi-chain clothing line outlets have their own websites or mobile applications which help customers purchasetheir products at the ease of their home.

In 2013, a McKinsey report noted that over 85 percent of Amazon sales revenue was generated from personalised recommendations. Since then, the use of personalised recommendations has grown.On average, today's personalised recommendations account for 27-percent of retail site revenue, according to a recent Salesforce report. This helps grow e-business across the globe. In spite of all the easy access of products,there are also some drawbacks.

Inorder to overcome the above mentioned defects, we have created a chatbot:Smart Fashion Recommender Application. The customer uses the chatbot topurchase products instead of the websites available. This multiples the probability of a personalised shopping experience for the customer. Our chatbots take customers beyond narrow browse and search functionality to
a place that can help them build the confidence that the purchase they'remaking is the right one for them.The chatbot is developed with the help of
IBM Watson Assistant. The database implemented is DB2. The web applicationis developed using flask. These include unease in navigating through the websites, inability to find a product in a specified material or colour.

1.2 PURPOSE

● For easy online purchasing of clothes.
● Inorder to receive customised recommendations.
● Multiple options of payment.
● It can provide personalised shopping experiences across physical andonline channels.
● Chatbots allow businesses to connect with customers in a personal waywithout the expense of human representatives.
● Instead of having to scroll through a dozen product pages, people onlyneed to answer a few questions and get the perfect recommendation withinminutes, if not seconds.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

E-Commerce cannot be held aside, particularly as it is the product of this growth. In the present Era most of the people have a smartphone with quick messaging and networking applications. A chat bot, is a piece of software that uses "quick messaging as the Program Interface" and allows customers to add the bot's name to their list in thesame way they add contacts and colleagues.

Currently, the conventional market is starting to be replaced with many online markets. The tight online market competition demands excellent service from sellers to buyers, so many online stores provide full 24-hour service. This service certainly requires a lotof money if done manually. This study proposes an intelligent chatbot system based onArtificial Intelligence Markup Language (AIML) which can be used as an e-commerce assistant.

The rise of e-commerce over the past 2 decades has had a major impact on society andthe way business is done on a global scale. Users have become more reliant on ecommerce than ever before in recent years. There's one visible downside about usingecommerce as a means to sell your products: some customers are wary of not having direct face to face contact with a sales representative. One of the major challenges of building an automated customer support system is categorizing natural language.
Several researches have been conducted on this topic.Androutsopoulos, G. D. Ritchieand P. Thanisch have shown different methods of natural language inferences to databases.

By offering an interactive and natural way for information seeking,multimodal chatbots are attracting increasing attention. The chabot can take data in the form of text and image as well.. Generally speaking, the system accepts multimodal utterances and thenclassifies user intentions to our predefined intention classes. The form of response (whether textual, visual or both) is also decided.

A chatbot or chat bot is a computer program designed to simulate an intelligent conversation with one or more human users via auditory or textual methods. Chatbotscan be programmed for small talk, or can also serve as a medium of interaction with users, providing them with answers based on regular questions. The chatbot understands context and delivers a response based on the message given to it. Chatbotis one of many examples of AI. Chatbots were initially designed as means of entertainment and some of them have been designed to pass the Turing Test.

Artificial intelligence (AI) based recommendation systems are commonplace for many years now, recent trend being conversational recommendation systems[2]. However, thefashion domain is unique given the tastes and preferences of individuals are different but at the same time each individual is influenced, in a greater or lesser degree, by fashion trends. In addition, fashion is often about combining various items together to present a look. A fashionable look is unique in that, although it consists of multiple items suggested together, they cannot be identified using traditional copurchase based analysis - instead a fashionable look needs to be learned based on an aesthetic sense as well as inputs from various sources like style magazines, social media etc.

The work is devoted to the development of the information system for creating a list of recommendations for fashionable style of clothing meeting the users` needs using NLPand chat bots. It provides studying and practical use of chat bots as virtual assistantsinvolving natural language processing. The purpose of this work is to develop software so that chat bot will be function on Telegram messenger base.

Conventional customer service chatbots are usually based on human dialogue, yet significant issues in terms of data scale and privacy. Distinct from existing counterparts,SuperAgent takes advantage of data from in-page product descriptions as well as usergenerated content from ecommerce websites, which is more practical and cost- effective when answering repetitive questions, freeing up human support staff to answer much higher value questions.

2.2 REFERENCES

E-COMMERCE ASSISTANT WITH A SMART CHATBOT USING AI By Manik Rakhra, Gurram Gopinadh, Shaik Aliraj, Sai Addepalli,Gurasis Singh, Siva Ganeshwar Reddy, Navaneshwar Reddy.

SMART CHATBOT SYSTEM FOR E-COMMERCE ASSITANCE BASED ON AIML By Arif Nursetyo, De Rosal Ignatius Moses Setiadi, Egia Rosi Subhiyakto.

DEVELOPMENT OF AN E-COMMERCE SALES CHATBOT By Mohammad Monirujjaman Khan.

KNOWLEDGE-AWARE MULTIMODAL FASHION CHATBOT By Lizi Liao , You Zhou , Yunshan Ma , Richang Hong , Tat-Seng Chua.

AN E-COMMERCE WEBSITE BASED CHATBOT By Deep Borkar, Chevelyn De Mello,Saurabh Patil.

RECOMMENDENCE AND FASHIONSENCE By Sapna Ria Chakraborty, Anagha M ,Kartikeya Vats,Khati Baradia Tanveer Khan ,Sandipan Sarkar,Sujoy Roychowdhury.

INFORMATION SYSTEM FOR RECOMMENDATION LIST FORMATION OF CLOTHES STYLE IMAGE SELECTION ACCORDING TO USE By Vitaliy Husak , Olga Lozynska, IhorKarpov, Ivan Peleshchak, Sofia Chyrun, Anatolii Vysotskyi.

SUPERAGENT: A CUSTOMER SERVICE CHATBOT FOR ECOMMERCE WEBSITES By Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, And Ming Zhou.

2.3 PROBLEM STATEMENT DEFINITION

Here the proposed methodology is about The major efficiency of using chatbots is automation of mundane tasks, like immediate answers when asked questions repetitively for different type and variety of customers. The application(chatbot) just byknowing their query in the chat section; it detects the keywords and gives appropriatereply to the users looking for a solution.

This study successfully built an intelligent chatbot system based on AIML in the Telegram application for E-commerce assistants. Input requests from users are carried out in three main processes, namely parsing, pattern matching, and data crawling. This paper has produced 100% response accuracy by testing using correct and formal wordsand sentences.

The reason for building such a modular system is to make the system available to moreplatforms. This present NLU engine trains its classifier from the classified training data provided by the admins.

In this work, the knowledge enriched multimodal fashion chatbot that is specifically designed to help users in searching for products and matching styles is presented. Thisalso gives customers more options to choose from as it also gives the ability to choosefrom an image the customer has.

A website based chatbot that attempts to improve User Interaction with the E- Commerce website. The chatbot has a stored set of responses, but also takes dynamicuser input into account and thus tends to provide relevant responses and product suggestions. Since the product database is independent of the stored responses, newerproducts under the respective category can be easily added and removed and require nomodification of the stored chatbot responses.

Our product is able to provide fashion recommendations to replicate the offline experience in the online world as much as possible. For deployment the model will needre-training and Fashionsence may need style database consisting of items which betterrepresent items housed in the inventory of the e-commerce client. Also, the outfit information to be used in the training process should reflect the desired fashion trends. Further enhancements include an end to end training architecture in the Fashionsencecomponent including the deep learning and LSH modules for better performance.

In this work the basic functions for the information system of forming the list of recommendations of the fashion style clothes according to the needs of the user usingthe NLP and chat bots are developed. The debuted software provides users with another way to search and select branded things in their own messenger, which in somecases can help increase sales to businesses as additional verticals.

We have developed SuperAgent, a customer service chatbot for e-commerce websites.Compared to conventional customer service chatbots, SuperAgent takes advantage oflarge-scale, publicly available, and crowd-sourced customer data. In addition, SuperAgent leverages state-of-the-art NLP and machine learning techniques, including fact QA, FAQ search, opinion-oriented text QA, as well as chit-chat conversation modeling.

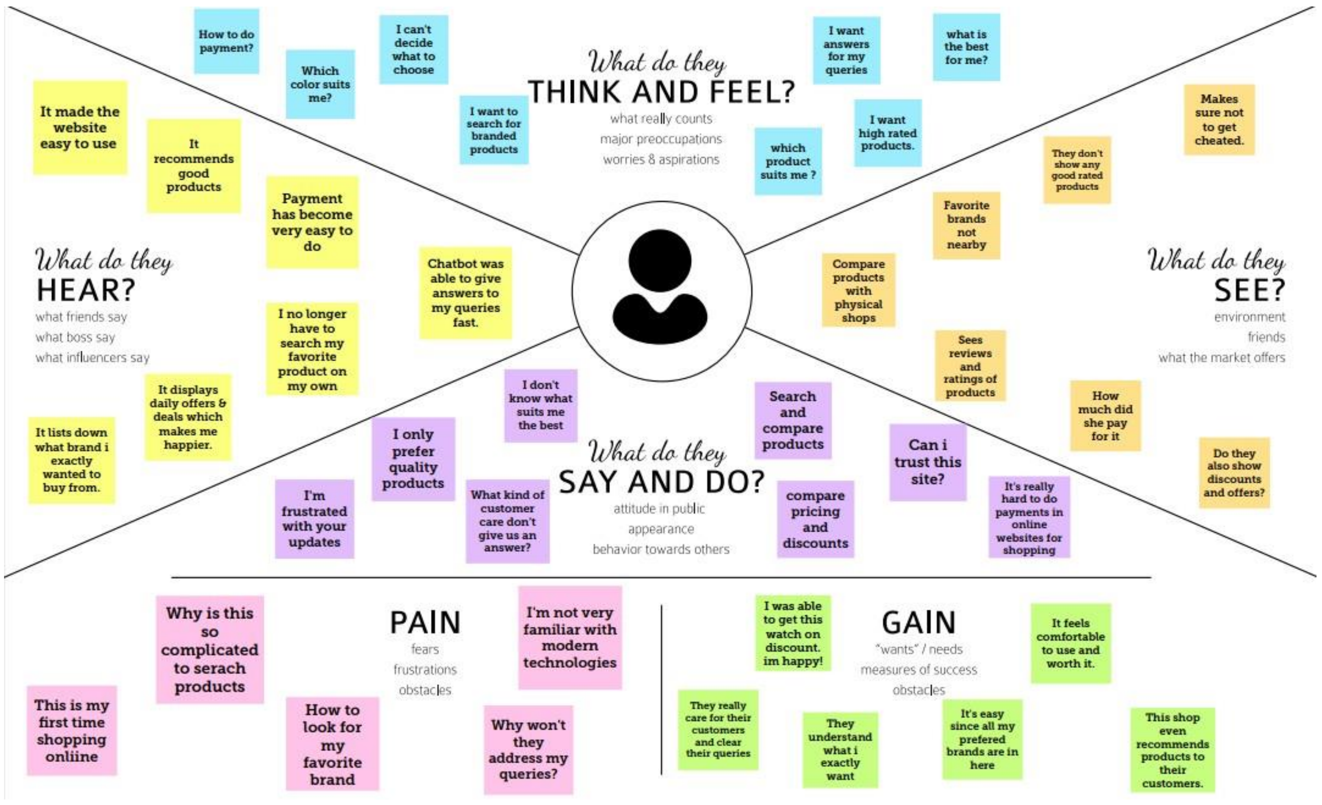
3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas:

An empathy map canvas serves as a foundation for outstanding user experiences, which focus on providing the experience customers want ratherthan forcing design teams to rely on guesswork.

Empathy map canvases help identify exactly what it is that users are lookingfor so brands can deliver. They can be particularly beneficial for getting teamson the same page about who users are and what they want from the brand.

An empathy map can be used by teams as a collaborative tool to better understand their clientele. Similar to user personas, an empathy map can represent a group of users, like a consumer sector. Teams can use the empathy map, which represents the principal user, to better understand thatperson's motivations, problems, and user experience. A simple yet effectiveworkshop called empathy mapping can be utilised with a variety of users, such as stakeholders, certain use cases, or entire teams.


What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
How to do payment?
Which color suits me?
I can't decide what to choose
I want to search for branded products
I want answers for my queries
what is the best for me?
I want high rated products.
which product suits me ?
What do they HEAR?
what friends say
what boss say
what influencers say
It made the website easy to use
It recommends good products
Payment has become very easy to do
Chatbot was able to give answers to my queries fast.
I no longer have to search my favorite product on my own
It displays daily offers & deals which makes me happier.
It lists down what brand i exactly wanted to buy from.
What do they SEE?
environment
friends
what the market offers
Makes sure not to get cheated.
They don't show any good rated products
Favorite brands not nearby
Compare products with physical shops
Sees reviews and ratings of products
How much did she pay for it
Do they also show discounts and offers?
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
I don't know what suits me the best
I only prefer quality products
I'm frustrated with your updates
What kind of customer care don't give us an answer?
Search and compare products
compare pricing and discounts
Can i trust this site?
It's really hard to do payments in online websites for shopping
PAIN
fears
frustrations
obstacles
Why is this so complicated to serach products
This is my first time shopping onliine
How to look for my favorite brand
I'm not very familiar with modern technologies
Why won't they address my queries?
GAIN
"wants" / needs
measures of success
obstacles
I was able to get this watch on discount. im happy!
They really care for their customers and clear their queries
They understand what i exactly want
It's easy since all my prefered brands are in here
It feels comfortable to use and worth it.
This shop even recommends products to their customers.

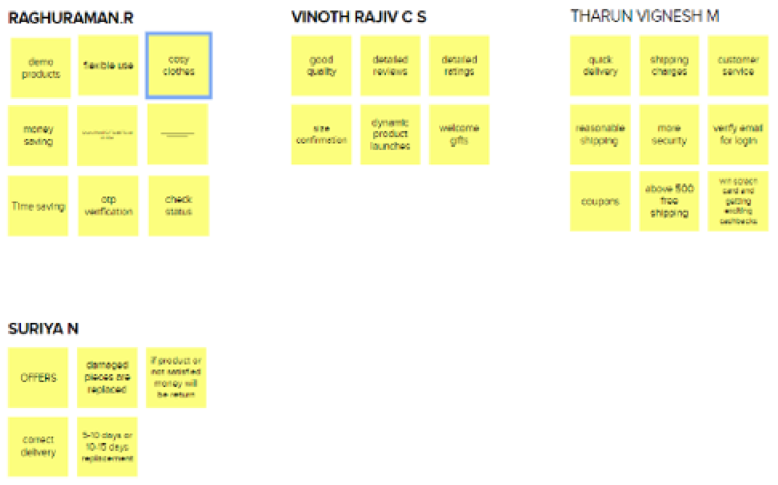
3.2 Ideation & Brainstroming

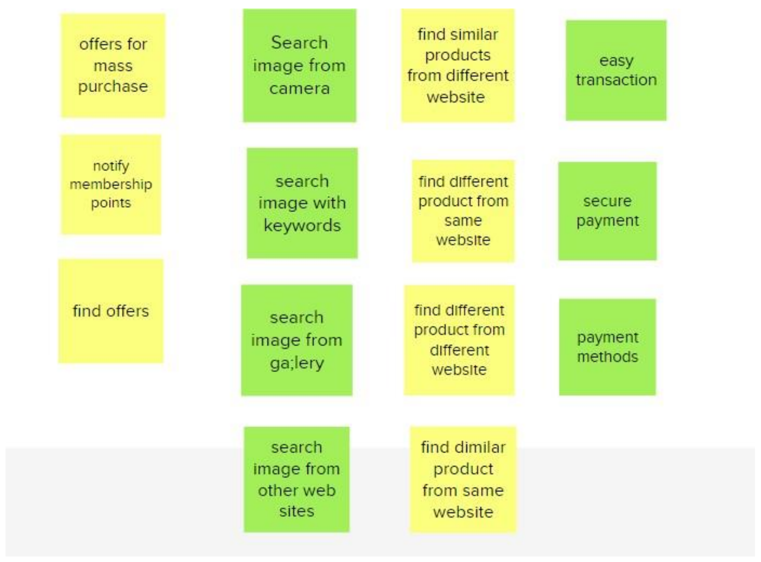
Brainstrom, Idea Listing and Grouping:

Brainstorming is a method of generating ideas and sharing knowledge to solve a particular commercial or technical problem, in which participants are encouraged to think without interruption. Brainstorming is a group activity where each participant shares their ideas as soon as they come to mind. At the conclusion of the session, ideas are categorised and ranked for follow-on action.

When planning a brainstorming session it is important to define clearly the topic to be addressed. A topic which is too specific can constrict thinking, while an ill-defined topic will not generate enough directly applicable ideas. The composition of the brainstorming group is important too. It should include people linked directly with the subject as well as those who can contribute novel and unexpected ideas. It can comprise staff from inside or outside the organisation.

PNT2022TMID01360

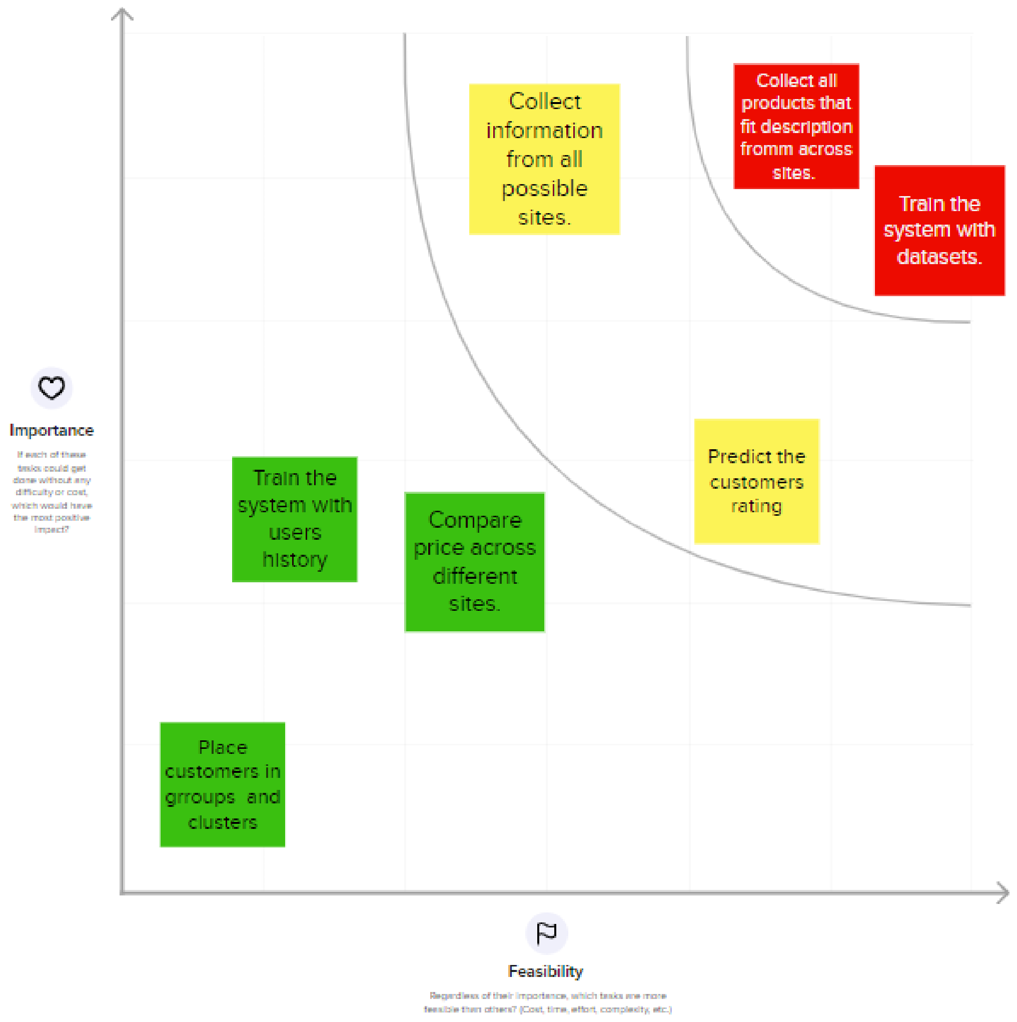



offers for mass purchase
notify membership points
find offers
Search image from camera
search image with keywords
search image from ga;lery
search image from other web sites
find similar products from different website
find different product from same website
find different product from different website
find dimilar product from same website
easy transaction
secure payment
payment methods

PNT2022TMID01360

<u>Idea prioritization:</u>

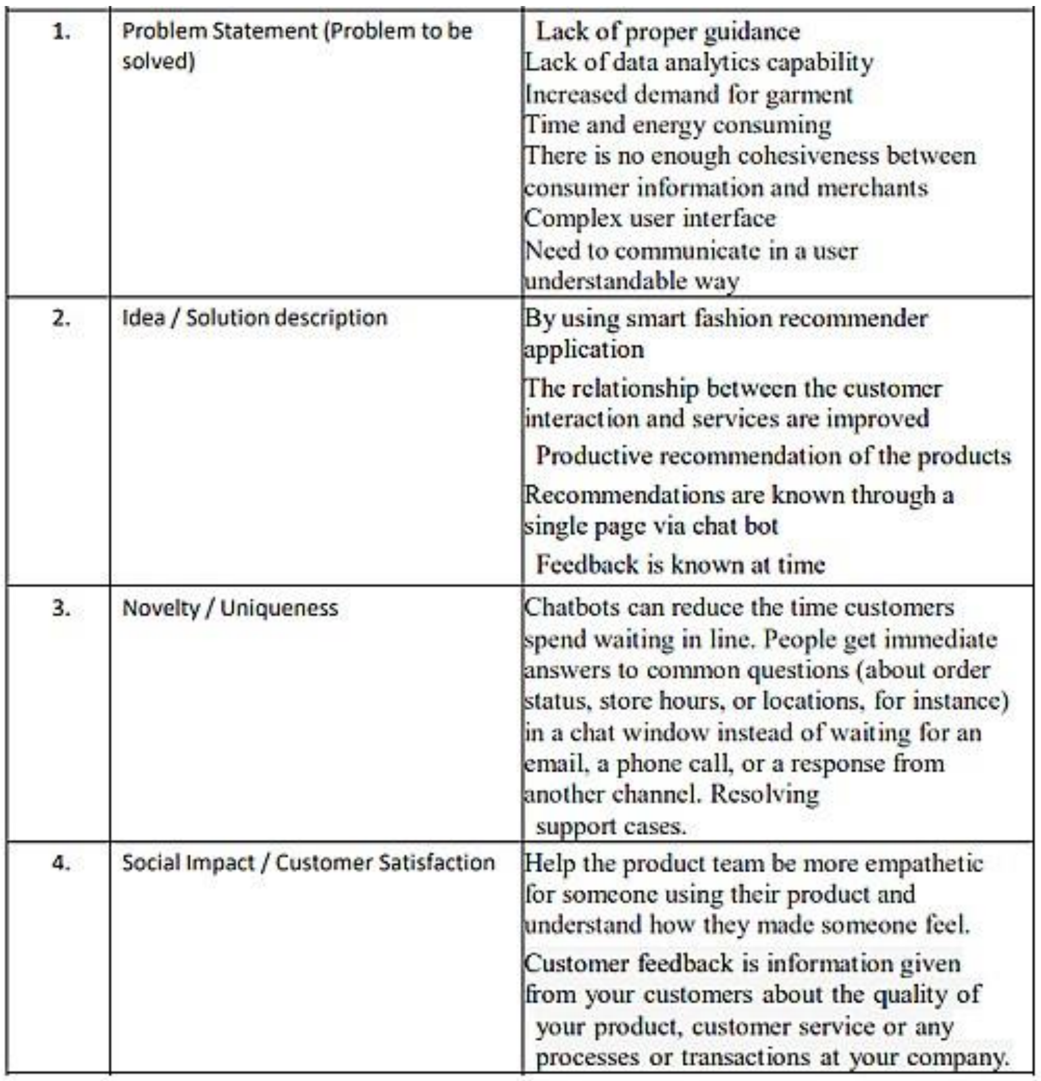
Only a small portion of the idea management process involves idea prioritisation. It takes time to develop an organised idea management strategy and a methodical approach to gathering, analysing, and prioritising new ideas.



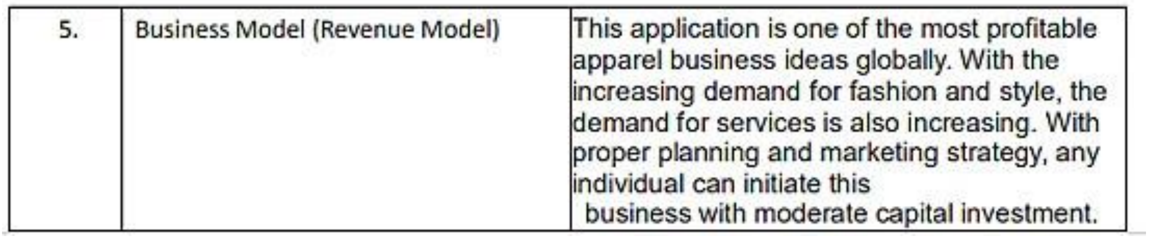
Collect information from all possible sites.
Collect all products that fit description fromm across sites.
Train the system with datasets.
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Train the system with users history
Compare price across different sites.
Predict the customers rating
Place customers in grroups and clusters
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

The technical solution that the implementation agency will present in response to the requirements
and goals of the project is referred to as the proposed solution.



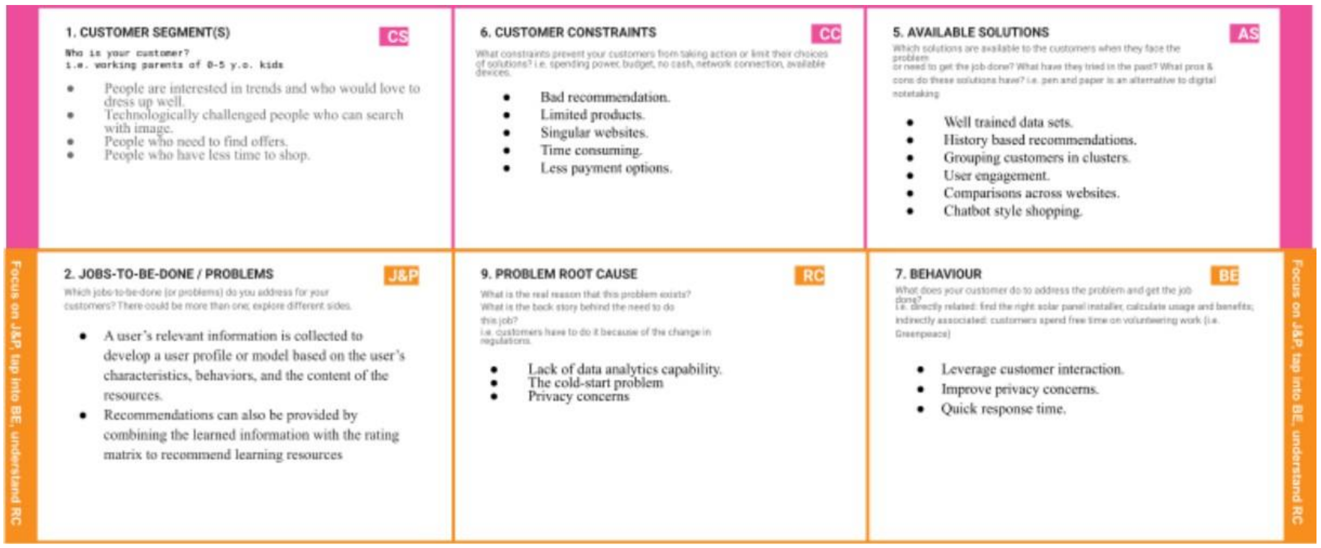
1.	Problem Statement (Problem to be solved)	Lack of proper guidance Lack of data analytics capability Increased demand for garment Time and energy consuming There is no enough cohesiveness between consumer information and merchants Complex user interface Need to communicate in a user understandable way
2.	Idea / Solution description	By using smart fashion recommender application The relationship between the customer interaction and services are improved Productive recommendation of the products Recommendations are known through a single page via chat bot Feedback is known at time
3.	Novelty / Uniqueness	Chatbots can reduce the time customers spend waiting in line. People get immediate answers to common questions (about order status, store hours, or locations, for instance) in a chat window instead of waiting for an email, a phone call, or a response from another channel. Resolving support cases.
4.	Social Impact / Customer Satisfaction	Help the product team be more empathetic for someone using their product and understand how they made someone feel. Customer feedback is information given from your customers about the quality of your product, customer service or any processes or transactions at your company.



5.	Business Model (Revenue Model)	This application is one of the most profitable apparel business ideas globally. With the increasing demand for fashion and style, the demand for services is also increasing. With proper planning and marketing strategy, any individual can initiate this business with moderate capital investment.

3.4 PROBLEM SOLUTION FIT

Finding an existing issue and finding a remedy that customers will find useful and satisfactory constitutes proposed solution fit.



1. CUSTOMER SEGMENT(S) CS

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

- People are interested in trends and who would love to dress up well.
- Technologically challenged people who can search with image.
- People who need to find offers.
- People who have less time to shop.

6. CUSTOMER CONSTRAINTS CC

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

- Bad recommendation.
- Limited products.
- Singular websites.
- Time consuming.
- Less payment options.

5. AVAILABLE SOLUTIONS AS

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

- Well trained data sets.
- History based recommendations.
- Grouping customers in clusters.
- User engagement.
- Comparisons across websites.
- Chatbot style shopping.

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

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

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

- A user's relevant information is collected to develop a user profile or model based on the user's characteristics, behaviors, and the content of the resources.
- Recommendations can also be provided by combining the learned information with the rating matrix to recommend learning resources

9. PROBLEM ROOT CAUSE RC

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

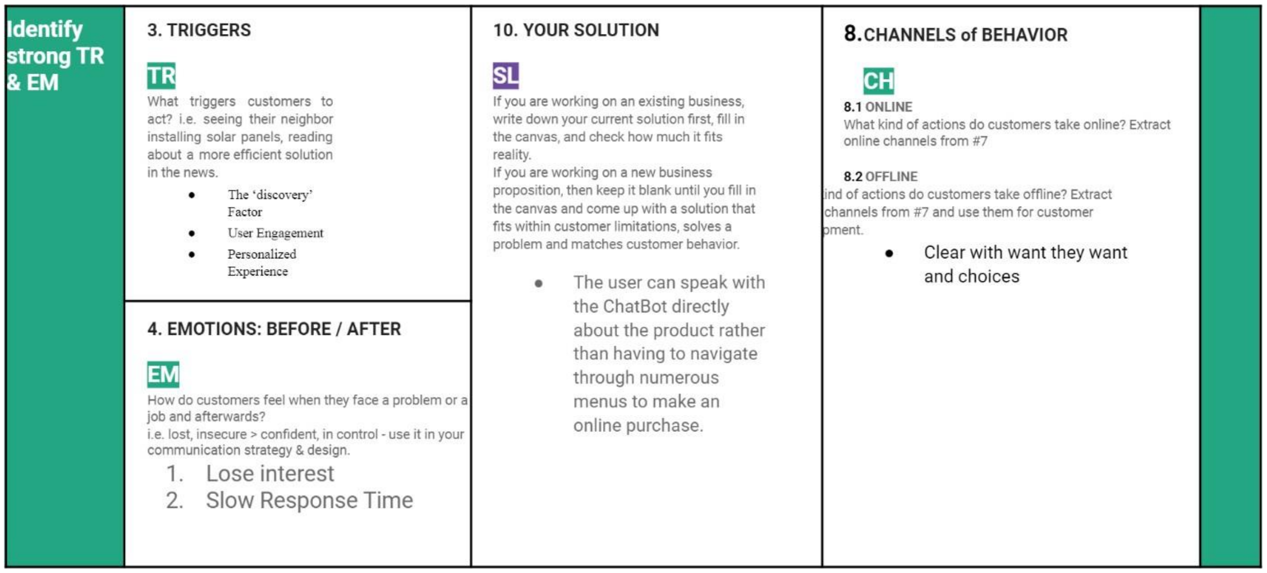
- Lack of data analytics capability.
- The cold-start problem
- Privacy concerns

7. BEHAVIOUR BE

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

- Leverage customer interaction.
- Improve privacy concerns.
- Quick response time.

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



Identify strong TR & EM

3. TRIGGERS TR

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

- The 'discovery' Factor
- User Engagement
- Personalized Experience

4. EMOTIONS: BEFORE / AFTER EM

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

1. Lose interest
2. Slow Response Time

10. YOUR SOLUTION SL

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

- The user can speak with the ChatBot directly about the product rather than having to navigate through numerous menus to make an online purchase.

8. CHANNELS of BEHAVIOR CH

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

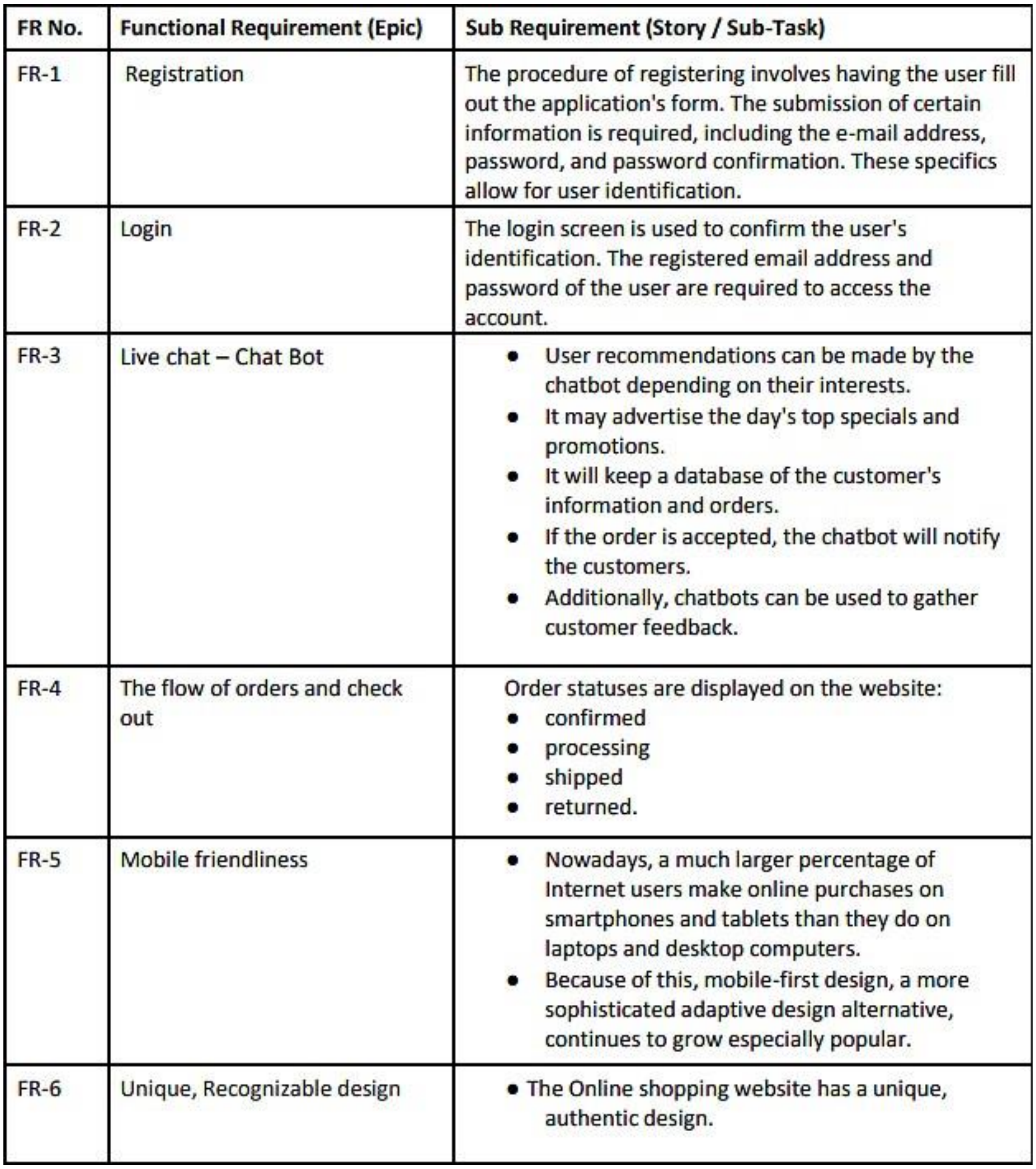
8.2 OFFLINE
ind of actions do customers take offline? Extract channels from #7 and use them for customer pment.

- Clear with want they want and choices

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

In software development and systems engineering, functional requirements are the desired operations of a programme or system. For consumers to perform their jobs, product features or functions must be developed by developers. It is essential to make them clear to the development team and the stakeholders. Functional requirements frequently describe how a system will act in specific situations.



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Registration	The procedure of registering involves having the user fill out the application's form. The submission of certain information is required, including the e-mail address, password, and password confirmation. These specifics allow for user identification.
FR-2	Login	The login screen is used to confirm the user's identification. The registered email address and password of the user are required to access the account.
FR-3	Live chat – Chat Bot	• User recommendations can be made by the chatbot depending on their interests. • It may advertise the day's top specials and promotions. • It will keep a database of the customer's information and orders. • If the order is accepted, the chatbot will notify the customers. • Additionally, chatbots can be used to gather customer feedback.
FR-4	The flow of orders and check out	Order statuses are displayed on the website: • confirmed • processing • shipped • returned.
FR-5	Mobile friendliness	• Nowadays, a much larger percentage of Internet users make online purchases on smartphones and tablets than they do on laptops and desktop computers. • Because of this, mobile-first design, a more sophisticated adaptive design alternative, continues to grow especially popular.
FR-6	Unique, Recognizable design	• The Online shopping website has a unique, authentic design.

4.2 NON FUNCTIONAL REQUIREMENTS

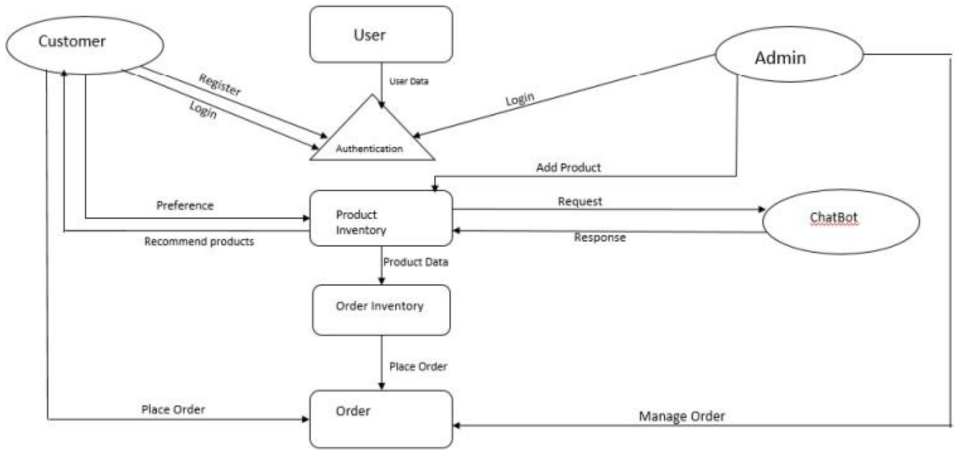
Non-functional requirements list the core attributes of a system. Sometimes, people refer to them as characteristics. The system's usability, scalability, maintainability, and performance are among the characteristics that are defined. They serve as restrictions or limitations on how the system is built for the different backlogs.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• Home page call to action- Make use of SEO, if people search on google for a product you offer it should be on the first page of result. • Easy navigation - The user can speak with the chatbot directly about the products. • product page optimization • good quality images that will attract buyers • Better Shopping cart • Enhance Payment site speed
NFR-2	**Security**	• Authentication and password management • Accountability - To authorize and monitor the use anonymous accounts and to remove • Confidentiality - Protect the user private information to prevent unauthorized access
NFR-3	**Reliability**	• Focusing on the Mediating Effect of Perceived Intelligence and Positive Cognition
NFR-4	**Performance**	• Speed up the webpage • Site optimization based on data analysis. • Strong SEO presence online. • Good use of the product description. • Comments and ratings
NFR-5	**Availability**	• The administrator needs to look up the stock availability in the database.
NFR-6	**Scalability**	• To expand your server capacity, memory, or disc space so that more people may transact on your website. • While expanding into new markets, the server side needs to add localization. • Chatbots to provide scalable customer support

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

It illustrates the various data kinds that will be imported into the system, exported from it, andstored there. A context diagram is commonly expanded into a DFD in order to show more of the system's finer details than the context diagram originally did.



5.2 SOLUTION AND TECHNICAL ARCHITECTURETECHNICAL ARCHITECTURE

The technical architecture includes the primary system parts, their connections, and the agreements that define how the parts interact. The goal of technical architects is to satisfy all business objectives with a performance- and security-optimized solution. establishing the foundation for technical systems. managing how programmes are carried out. Making ensuring the system runs properly by working with the software development team.

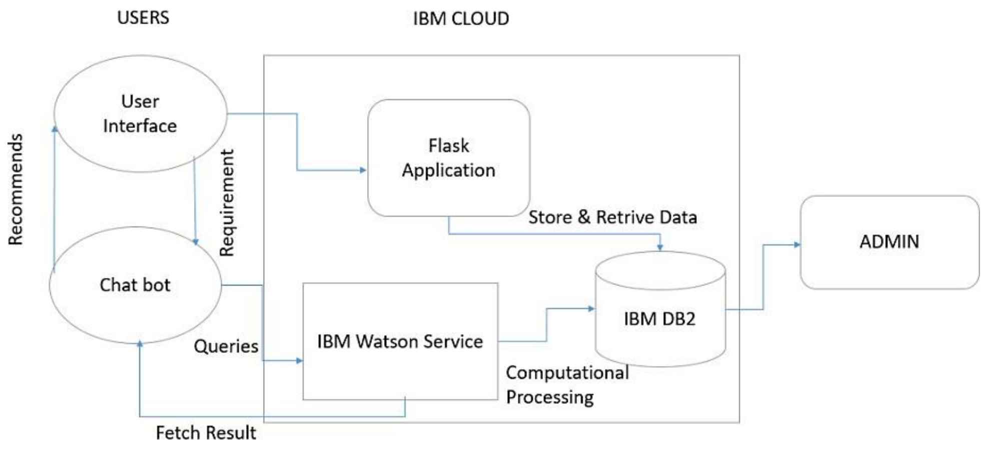
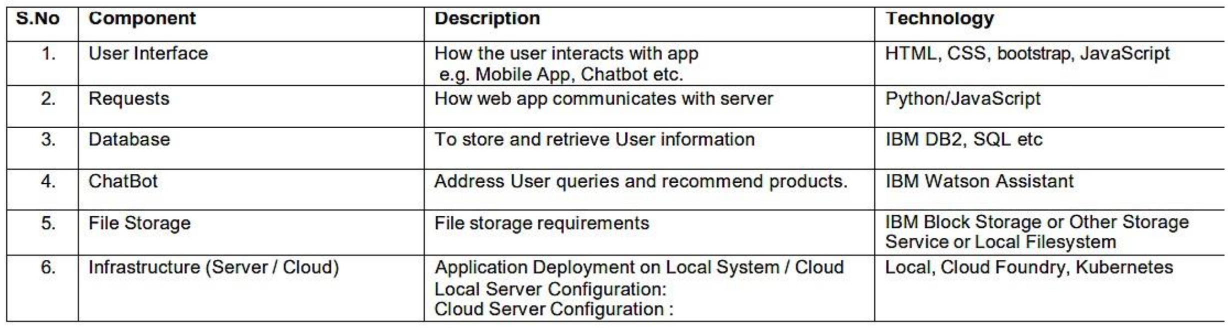
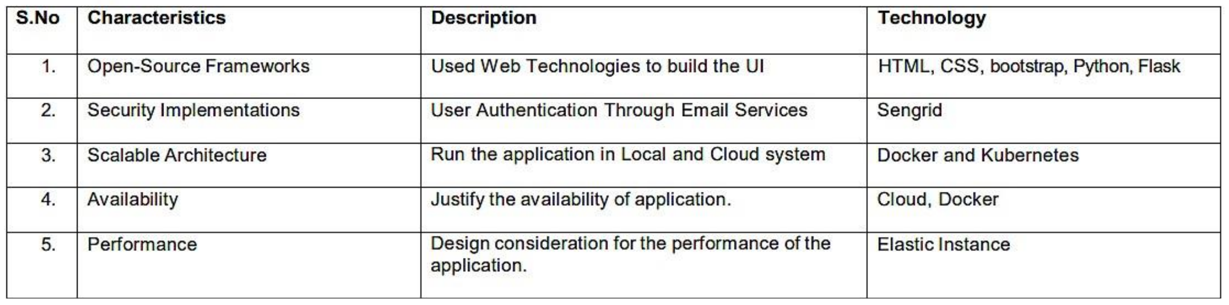


Table-1 : Components & Technologies:



S.No	Component	Description	Technology
1.	User Interface	How the user interacts with app e.g. Mobile App, Chatbot etc.	HTML, CSS, bootstrap, JavaScript
2.	Requests	How web app communicates with server	Python/JavaScript
3.	Database	To store and retrieve User information	IBM DB2, SQL etc
4.	ChatBot	Address User queries and recommend products.	IBM Watson Assistant
5.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
6.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes

Table-2: Application Characteristics:



S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Used Web Technologies to build the UI	HTML, CSS, bootstrap, Python, Flask
2.	Security Implementations	User Authentication Through Email Services	Sengrid
3.	Scalable Architecture	Run the application in Local and Cloud system	Docker and Kubernetes
4.	Availability	Justify the availability of application.	Cloud, Docker
5.	Performance	Design consideration for the performance of the application.	Elastic Instance

SOLUTION ARCHITECTURE

We have created a brand-new, ground-breaking method that enables you to conduct direct internet buying based solely on your preferences. The chatbot can be used to accomplish this. You will be working on two modules for this project:

Admin

User

This project makes use of chatbots to gather all necessary preferences and make product recommendations to the user, saving the user from having to search for products in the searchbar and navigate to specific products to find the relevant preferences.

The solution is put into practise in a way that enhances consumer and application interaction.

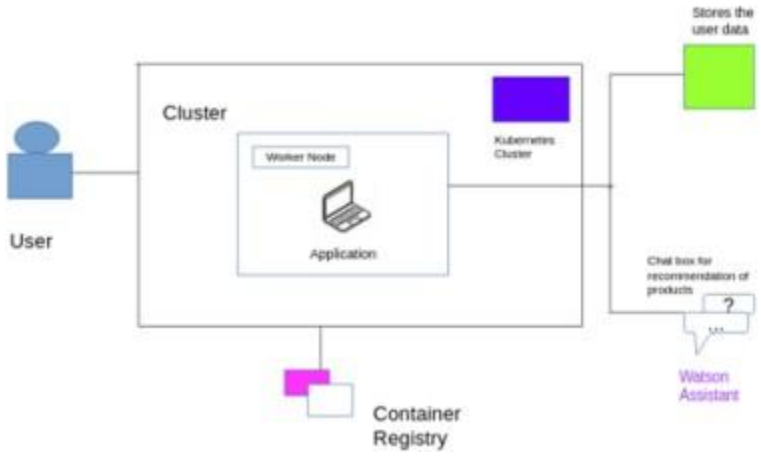
The chatbot periodically delivers messages to inform users about offers and preferences.

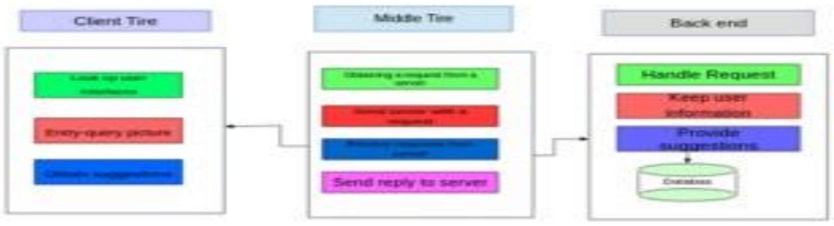
This application employs a token to securely authenticate and authorise users due to security concerns.

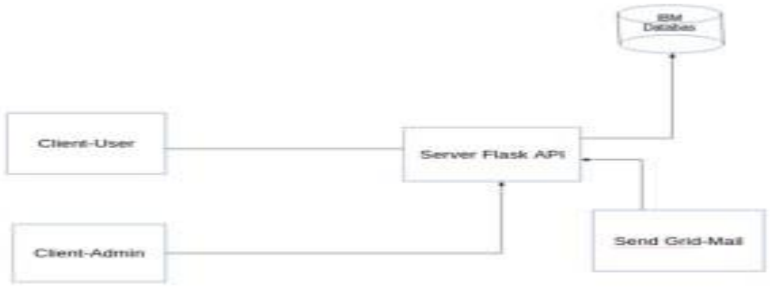
The user id and role are encoded in the token.

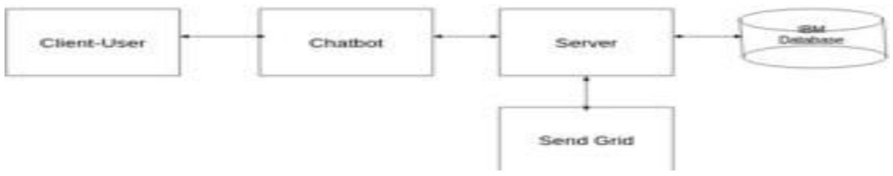
Access to the resources is limited to specified users based on the encoded information.

PNT2022TMID01360



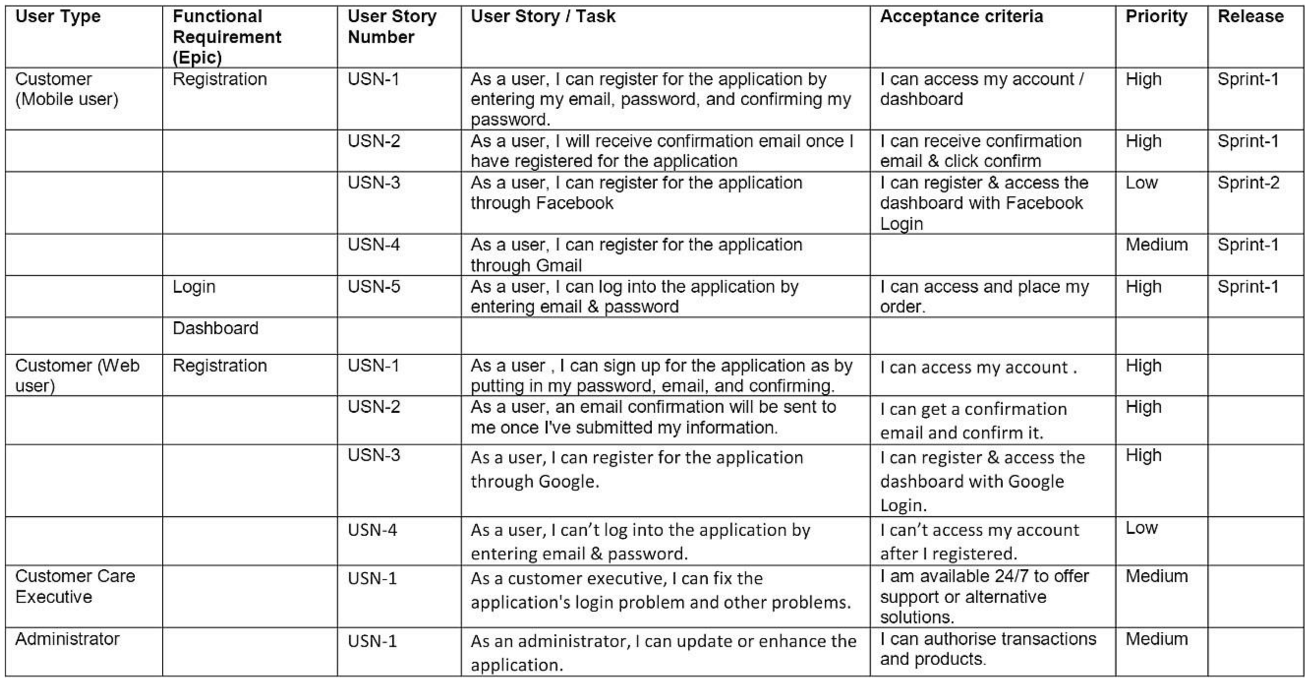






5.3 USER STORIES

An informal, comprehensive description of a software feature written from the client's or end user's perspective is known as a "user narrative." The purpose of a user story is to explain how a piece of work will give the client a particular value. The main benefit of employing user stories in agile product development may be that they are not designed to stand alone, unlike requirements or use cases. Instead, each user narrative serves as a pending topic for discussion with the development team.



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 and place my order.	High	Sprint-1
	Dashboard					
Customer (Web user)	Registration	USN-1	As a user , I can sign up for the application as by putting in my password, email, and confirming.	I can access my account .	High	
		USN-2	As a user, an email confirmation will be sent to me once I've submitted my information.	I can get a confirmation email and confirm it.	High	
		USN-3	As a user, I can register for the application through Google.	I can register & access the dashboard with Google Login.	High	
		USN-4	As a user, I can't log into the application by entering email & password.	I can't access my account after I registered.	Low	
Customer Care Executive		USN-1	As a customer executive, I can fix the application's login problem and other problems.	I am available 24/7 to offer support or alternative solutions.	Medium	
Administrator		USN-1	As an administrator, I can update or enhance the application.	I can authorise transactions and products.	Medium	

6. PROJECT PLANNING AND SCHEDULING

6.1 SPIRNT PLANNING & ESTIMATION

A sprint is an allotted time frame during which a specific amount of work on a project will be finished. A project using the agile methodology will be divided into a number of sprints, with each sprint bringing the project one step closer to completion.

Sprint planning kicks off a sprint in the scrum methodology. The purpose of sprint planning is to define what can be accomplished in a sprint and how it will be accomplished. Sprint planning is a collective effort across the entire scrum team.

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story /Task**	**Story Points**	**Priority**	**TeamMembers**
Sprint-1	User Panel	USN-1	The user will login into the website and go through the products available on the website	20	High	Raghuraman.R Vinoth Rajiv.C.S Tharun Vignesh M Suriya.N
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	Raghuraman.R Vinoth Rajiv.C.S Tharun Vignesh M Suriya.N
Sprint-3	Chat Bot	USN-3	The user can directly talk to Chatbot rega	20	High	Raghuraman.R Vinoth Rajiv.C.S Tharun Vignesh M Suriya.N
Sprint-4	Final delivery	USN-4	Container of applications using docker kubernetes and deployment the application. Create the documentation and final	20	High	Raghuraman.R Vinoth Rajiv.C.S Tharun Vignesh M Suriya.N

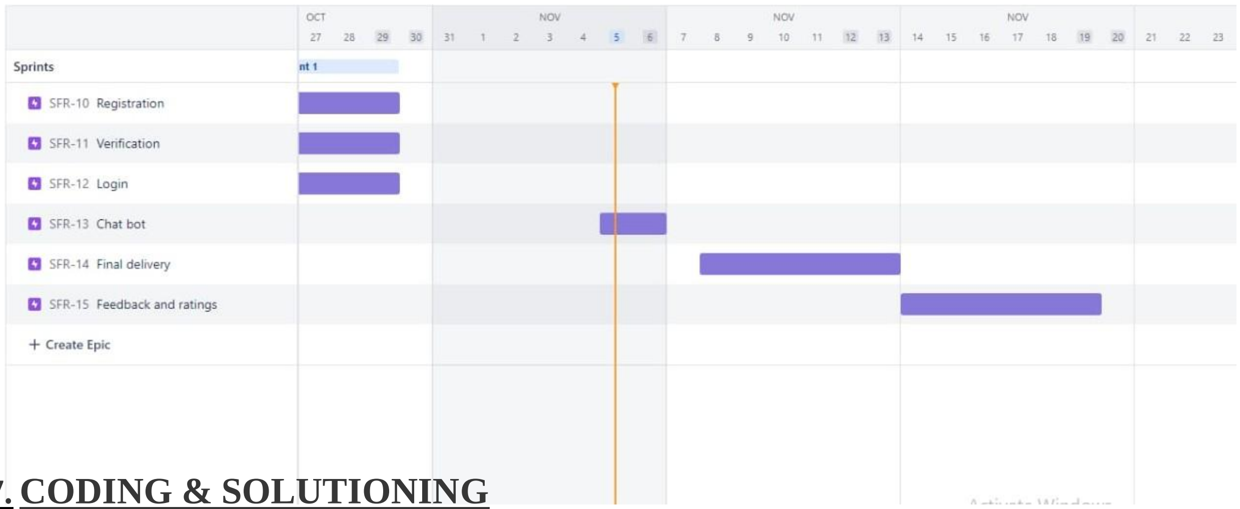
6.2 SPRINT DELIVERY SCHEDULE

A sprint schedule is a summary of the whole sprint planning procedure in writing. It requires adequate research, planning, and collaboration and is one of the first phases in the agile sprint planning process. A product backlog, which is a collection of open requests for development and iteration, serves as its focal point.

A project management chart is a burn down chart, which shows how quickly a team is workingthrough a customer's user stories. This agile tool documents how a feature is described fromthe perspective of the end user and contrasts the total effort with the amount of work for each agile sprint.

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	06 Nov 20222
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	13 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORTS FROM JIRA



7. CODING & SOLUTIONING

7.1 FEATURE 1

APP.PY

```
from flask import * import
sqlite3, hashlib, os
from werkzeug.utils import secure_filename

app = Flask(_name_) app.secret_key =
'random string' UPLOAD_FOLDER  =
'static/uploads'
ALLOWED_EXTENSIONS = set(['jpeg', 'jpg', 'png', 'gif', 'webp', 'avif'])
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
```

```
def getLoginDetails():
    with  sqlite3.connect('database.db')  as  conn:cur =
        conn.cursor()
        if 'email' not in session:
            loggedIn = False
            firstName = ''
            noOfItems = 0
        else:
            loggedIn = True
            cur.execute("SELECT userId, firstName FROM users WHERE email =
?", (session['email'], ))
            userId, firstName = cur.fetchone()
            cur.execute("SELECT count(productId) FROM kart WHERE userId =
?", (userId, ))
            noOfItems = cur.fetchone()[0]
    conn.close()
    return (loggedIn, firstName, noOfItems)

@app.route("/") def
root():
    loggedIn, firstName, noOfItems = getLoginDetails()with
    sqlite3.connect('database.db') as conn:
        cur = conn.cursor()
        cur.execute('SELECT productId, name, price, description, image, stockFROM
products')
        itemData = cur.fetchall()
        cur.execute('SELECT categoryId, name FROM categories')
        categoryData = cur.fetchall()
    itemData = parse(itemData)
    return render_template('home.html', itemData=itemData,
loggedIn=loggedIn, firstName=firstName, noOfItems=noOfItems,
categoryData=categoryData)

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


@app.route("/add") def
admin():
    with  sqlite3.connect('database.db')  as  conn:cur =
        conn.cursor()
        cur.execute("SELECT categoryId, name FROM categories")
        categories = cur.fetchall()
    conn.close()
    return  render_template('add.html',  categories=categories)

@app.route("/addItem", methods=["GET", "POST"])
def addItem():
    if request.method == "POST":
        name = request.form['name']
        price = float(request.form['price']) description
        = request.form['description'] stock =
        int(request.form['stock']) categoryId  =
        int(request.form['category'])

        #Uploading image procedure
        image = request.files['image']
        if image and allowed_file(image.filename): filename
            = secure_filename(image.filename)
          image.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
      imagename = filename
      with  sqlite3.connect('database.db')  as  conn:try:
            cur = conn.cursor()
            cur.execute('''INSERT INTO products (name, price, description, image, stock,
categoryId) VALUES (?, ?, ?, ?, ?, ?)''', (name, price, description, imagename, stock,
categoryId))
            conn.commit() msg="added
            successfully"
         except:
            msg="error occured"
            conn.rollback()
      conn.close()
      print(msg)
      return redirect(url_for('root'))

@app.route("/remove") def
remove():
   with  sqlite3.connect('database.db')  as  conn:cur =
      conn.cursor()
      cur.execute('SELECT productId, name, price, description, image, stock FROM
products')
      data = cur.fetchall()
   conn.close()
   return  render_template('remove.html',  data=data)

@app.route("/removeItem") def
removeItem():
   productId = request.args.get('productId') with
   sqlite3.connect('database.db')  as  conn:
      try:
            cur = conn.cursor()
            cur.execute('DELETE FROM products WHERE productID = ?', (productId, ))
            conn.commit()
            msg = "Deleted successsfully"except:
            conn.rollback()
            msg = "Error occured"
    conn.close()
    print(msg)
    return redirect(url_for('root'))

@app.route("/displayCategory") def
displayCategory():
        loggedIn, firstName, noOfItems = getLoginDetails()
        categoryId = request.args.get("categoryId")
        with sqlite3.connect('database.db') as conn:cur =
            conn.cursor()
            cur.execute("SELECT products.productId, products.name, products.price,
products.image, categories.name FROM products, categories WHERE
products.categoryId = categories.categoryId AND categories.categoryId = ?",
(categoryId, ))
            data = cur.fetchall()
        conn.close() categoryName =
        data[0][4]data = parse(data)
        return render_template('displayCategory.html', data=data, loggedIn=loggedIn,firstName=firstName,
noOfItems=noOfItems, categoryName=categoryName)

@app.route("/account/profile") def
profileHome():
    if 'email' not in session:
        return redirect(url_for('root'))
    loggedIn, firstName, noOfItems = getLoginDetails()
    return  render_template("profileHome.html",  loggedIn=loggedIn,  firstName=firstName,
noOfItems=noOfItems)

@app.route("/loginForm") def
loginForm():
    if 'email' in session:
        return redirect(url_for('root'))else:
        return  render_template('login.html',  error='')

@app.route("/login",  methods  =  ['POST',  'GET'])def
login():
    if request.method == 'POST': email
        = request.form['email']
        password = request.form['password']if
        is_valid(email, password):
            session['email'] = email return
            redirect(url_for('root'))
        else:
            error = 'Invalid UserId / Password'
            return  render_template('login.html',  error=error)

@app.route("/productDescription") def
productDescription():
    loggedIn, firstName, noOfItems = getLoginDetails()
    productId = request.args.get('productId')
    with  sqlite3.connect('database.db')  as  conn:cur =
        conn.cursor()
        cur.execute('SELECT productId, name, price, description, image, stock FROM
products WHERE productId = ?', (productId, ))
        productData = cur.fetchone()
    conn.close()
    return render_template("productDescription.html", data=productData, loggedIn =loggedIn,
firstName = firstName, noOfItems = noOfItems)
```

```
@app.route("/addToCart")
def addToCart():
    if 'email' not in session:
        return redirect(url_for('loginForm'))else:
        productId = int(request.args.get('productId'))with
        sqlite3.connect('database.db') as conn:
            cur = conn.cursor()
            cur.execute("SELECT userId FROM users WHERE email = ?", (session['email'], ))
            userId = cur.fetchone()[0]
            try:
                cur.execute("INSERT INTO kart (userId, productId) VALUES (?, ?)", (userId,
productId))
                conn.commit()
                msg = "Added successfully"
            except:
                conn.rollback()
                msg = "Error occured"
        conn.close()
        return redirect(url_for('root'))

@app.route("/cart") def
cart():
    if 'email' not in session:
        return  redirect(url_for('loginForm'))
    loggedIn, firstName, noOfItems = getLoginDetails()
    email = session['email']
    with  sqlite3.connect('database.db')  as  conn:cur =
        conn.cursor()
        cur.execute("SELECT userId FROM users WHERE email = ?", (email, ))
        userId = cur.fetchone()[0]
        cur.execute("SELECT products.productId, products.name, products.price, products.image
FROM products, kart WHERE products.productId = kart.productId ANDkart.userId = ?",
(userId, ))
```

```
        products = cur.fetchall()
    totalPrice = 0
    for row in products:
        totalPrice += row[2]
    return render_template("cart.html", products = products, totalPrice=totalPrice,loggedIn=loggedIn,
firstName=firstName, noOfItems=noOfItems)

@app.route("/removeFromCart") def
removeFromCart():
    if 'email' not in session:
        return redirect(url_for('loginForm'))email
    = session['email']
    productId = int(request.args.get('productId'))with
    sqlite3.connect('database.db') as conn:
        cur = conn.cursor()
        cur.execute("SELECT userId FROM users WHERE email = ?", (email, ))
        userId = cur.fetchone()[0]
        try:
            cur.execute("DELETE FROM kart WHERE userId = ? AND productId = ?", (userId,
productId))
            conn.commit()
            msg = "removed successfully"except:
            conn.rollback()
            msg = "error occured"
    conn.close()
    return redirect(url_for('root'))

@app.route("/logout") def
logout():
    session.pop('email', None) return
    redirect(url_for('root'))

def is_valid(email, password):
    con = sqlite3.connect('database.db')cur =
    con.cursor()
    cur.execute('SELECT email, password FROM users')data
    = cur.fetchall()
    for row in data:
        if row[0] == email and row[1] == hashlib.md5(password.encode()).hexdigest():return
            True
    return False

@app.route("/register", methods = ['GET', 'POST'])def
register():
    if request.method == 'POST':
        #Parse form data
        password = request.form['password']email
        = request.form['email'] firstName =
        request.form['firstName']lastName =
        request.form['lastName']address1 =
        request.form['address1'] address2 =
        request.form['address2'] zipcode =
        request.form['zipcode']
        city = request.form['city'] state
        = request.form['state']
        country = request.form['country']phone =
        request.form['phone']

       with sqlite3.connect('database.db') as con:try:
                cur = con.cursor()
                cur.execute('INSERT INTO users (password, email, firstName, lastName, address1,
address2, zipcode, city, state, country, phone) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?)', (hashlib.md5(password.encode()).hexdigest(), email, firstName, lastName, address1,address2,
zipcode, city, state, country, phone))

                con.commit()
                msg = "Registered Successfully"
            except:
                con.rollback()
                msg = "Error occured"
        con.close()
        return render_template("login.html", error=msg)

@app.route("/registerationForm") def
registrationForm():
    return render_template("register.html")

def allowed_file(filename):
    return '.' in filename and \
            filename.rsplit('.', 1)[1] in ALLOWED_EXTENSIONS

def parse(data):
    ans = []
    i = 0
    while i < len(data):
        curr = []
        for j in range(7):
            if i >= len(data):
                break
            curr.append(data[i]) i
            += 1
        ans.append(curr)
    return ans
if__name___== '_main_':
    app.run(debug=True)
```

7.2 FEATURE 2

HOME.HTML

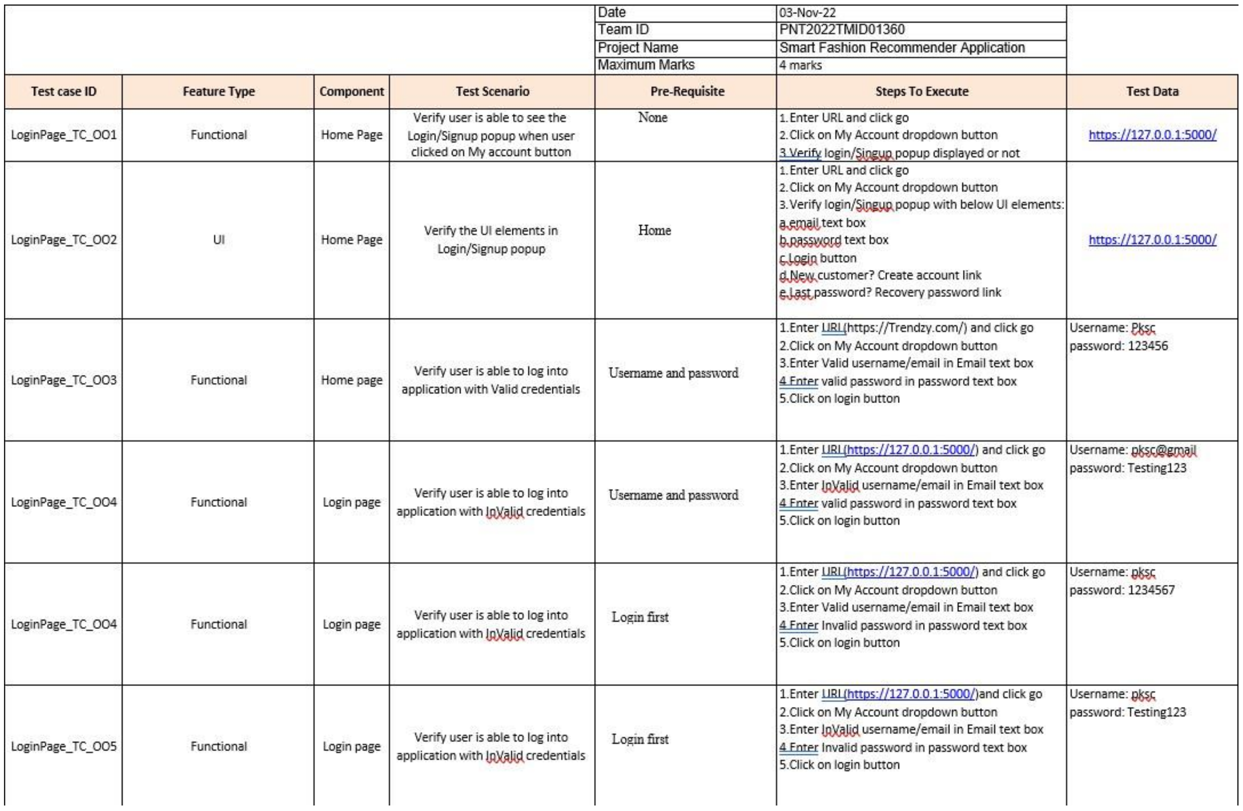
```
<!DOCTYPE HTML>
<html>
<head>
<title>Welcome</title>
<link rel="stylesheet" href="static/home.css">
<link rel="stylesheet" href="static/topStyle.css" />
</head>
<body>
    <div id="title">
        <a href="/"><img src="static/uploads/trendzylogo.png" width="60px" id="logo"></a>
        <form>
            <h1 id="cname">TRENDZY</h1>
        </form>

        {% if not loggedIn %}
      <div id="signInButton">
        <a class="link" href="/loginForm">Sign In</a>
    </div>
    {% else %}
    <div class="dropdown" id="signInButton">
        <button  class="dropbtn">Hello,{{firstName}}</button>
        <div class="dropdown-content">
            <a href="/logout">Sign Out</a>
        </div>
    </div>
    {% endif %}
    <div id="kart">
        <a class="link" href="/cart">
            <img src="static/uploads/cart.png" id="cartIcon" />CART
            {{noOfItems}}
         </a>
    </div>
    </div>
    <div class="homeimage">
        <div id="text">
          <h1 id="inside">World's Leading Fashion Community<br>TrendZy</h1>
        </div>
    </div>
    <div id="freeship">
        <h5>FREE SHIPPING AROUND THE WORLD</h5>
    </div>
<div class="display">
    <div class="displayCategory">
        <h2 id="shopbycategory">Shop By Category</h2>
            <table>
                <tr>
                    {% for row in categoryData %}
                    <td><a href="/displayCategory?categoryId={{row[0]}}"
id="categories">{{row[1]}}</a></td>
                    {% endfor %}
                </tr>
            </table>
    </div>
</div>
<h2 id="dealheader">Deal of the Day</h2>
<div id="deals">
    <div id="watches">
        <h2 id="imagetext">25% off on Axis Bank Debit Cards<br>Apple Watches</h2>
    </div>
    <div id="airpods">
        <h3 id="imagetext">AirPods<br>ONLY ₹9000</h3>
    </div>
    <div id="laptops">
        <h3 id="imagetext">Hp Pavillion New Launch!<br>Core i5 11th gen
Processors</h3>
    </div>
    <div id="iphones">
        <h3 id="imagetext">Now or never Offer!<br>Iphones at ₹60000</h3>
    </div>
</div>
<br>

<div id="bot">
    <h2>Need Assistance in choosing what's best for you?<br>Chat with our
QueenBee!</h2>
    <img  src="static/uploads/chatbot.webp"  width="200px">
</div>
<div id="about">
    <h1>About Us</h1>
    <p>TrendZy is an International website. Explore and browse all the latest collections:
handbags, leather goods, ready to wear, shoes, jewellery</p>
    <p>For Queries : trendzy@hotmail.com</p>
    <p>No: 9999999999</p>
    <img  src="static/uploads/aboutlogo.jpeg"  width="110px">
</div>
<script> window.watsonAssistantChatOptions =
    {
      integrationID: "ba2316ec-e598-43e0-889a-b0dbebf9a363", /  The ID of this
integration.
      region: "eu-gb", /  The region your integration is hosted in.
      serviceInstanceID: "1d7826e9-10ef-4a8d-9687-b17a98a71b25", / The ID of yourservice
instance.
      onLoad: function(instance) { instance.render(); }
    };
    setTimeout(function(){
      const  t=document.createElement('script');
      t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
      document.head.appendChild(t);
    });
  </script>
</body>
</html>
```

8. TESTING

8.1 TEST CASES



				Date	03-Nov-22	
				Team ID	PNT2022TMID01360	
				Project Name	Smart Fashion Recommender Application	
				Maximum Marks	4 marks	
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	None	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	https://127.0.0.1:5000/
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup	Home	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	https://127.0.0.1:5000/
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials	Username and password	1.Enter URL(https://Trendzy.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: Pksc password: 123456
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials	Username and password	1.Enter URL(https://127.0.0.1:5000/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: pksc@gmail password: Testing123
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials	Login first	1.Enter URL(https://127.0.0.1:5000/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: pksc password: 1234567
LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with InValid credentials	Login first	1.Enter URL(https://127.0.0.1:5000/)and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: pksc password: Testing123

Verify user is able to Buy products	Functional	order page	verify user is able to order products		user will buy the product in order page	user data
Verify user is get purchase conformation mail?	Functional	order page	verify user is able to get conformation mail	Report generation	user will get the mail Notification	email

8.2 USER ACCEPTANCE TESTING

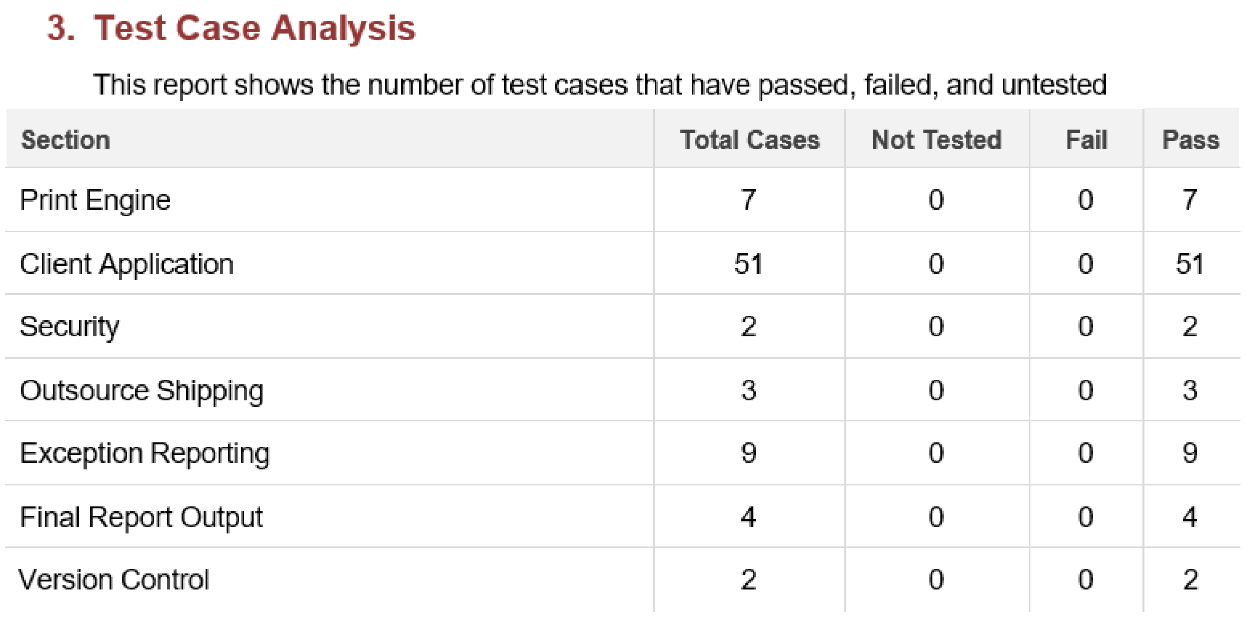
1. Purpose of Document

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	10	4	2	3	20
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	7



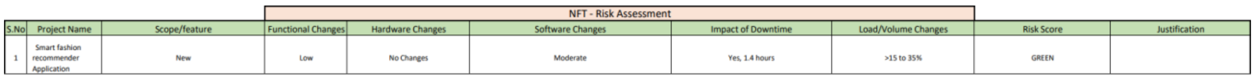
3. Test Case Analysis

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

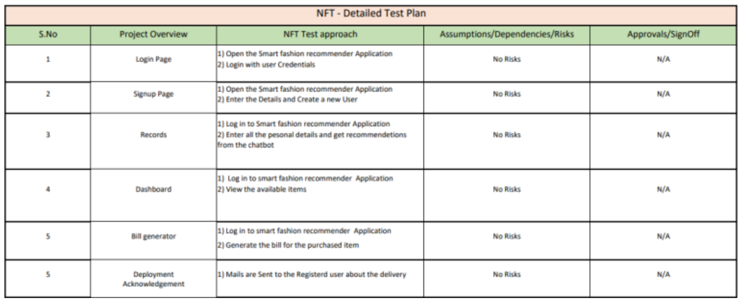
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 PERFORMANCE METRICS:



			NFT - Risk Assessment						
S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification
1	Smart fashion recommender Application	New	Low	No Changes	Moderate	Yes, 1.4 hours	>15 to 35%	GREEN	



NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	Login Page	1) Open the Smart fashion recommender Application 2) Login with user Credentials	No Risks	N/A
2	Signup Page	1) Open the Smart fashion recommender Application 2) Enter the Details and Create a new User	No Risks	N/A
3	Records	1) Log in to Smart fashion recommender Application 2) Enter all the pesonal details and get recommendetions from the chatbot	No Risks	N/A
4	Dashboard	1) Log in to smart fashion recommender Application 2) View the available items	No Risks	N/A
5	Bill generator	1) Log in to smart fashion recommender Application 2) Generate the bill for the purchased item	No Risks	N/A
5	Deployment Acknowledgement	1) Mails are Sent to the Registerd user about the delivery	No Risks	N/A

The performance of a recommendation algorithm is evaluated by using some specificmetrics that indicate the accuracy of the system. The type of metric used depends onthe type of filtering technique. Root Mean Square Error (RMSE), Receiver Operating Characteristics (ROC), Area Under Cover (AUC), Precision, Recall and F1 score is generally used to evaluate the performance or accuracy of the recommendation algorithms. Root-mean square error (RMSE). RMSE is widely used in evaluating and comparing the performance of a recommendation system model compared to other models. A lower RMSE value indicates higher performance by the recommendationmodel. RMSE, can be as represented as follows:

$$RMSE = \sqrt{\frac{1}{N_p} \sum_{u,i} (p_{ui} - r_{ui})^2} \quad (1)$$

where, Np is the total number of predictions, pui is the predicted rating that a user u willselect an item i and rui is the real rating.

Precision.

Precision can be defined as the fraction of correct recommendations or predictions (known as True Positive) to the total number of recommendations provided, which canbe as represented as follows:

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\,(TP) + False\ Positive\ (FP)} \quad (2)$$

It is also defined as the ratio of the number of relevant recommended items to thenumber of recommended items expressed as percentages.

Recall.

Recall can be defined as the fraction of correct recommendations or predictions (knownas True Positive) to the total number of correct relevant recommendations provided, which can be as represented as follows:

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\,(TP) + False\ Negative\ (FN)} \quad (3)$$

It is also defined as the ratio of the number of relevant recommended items to the totalnumber of relevant items expressed as percentages.

F1 Score.

F1 score is an indicator of the accuracy of the model and ranges from 0 to 1, where a value close to 1 represents higher recommendation or prediction accuracy. It representsprecision and recall as a single metric and can be as represented as follows:

$$F1\ score = 2 \times \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

Coverage.

Coverage is used to measure the percentage of items which are recommended by thealgorithm among all of the items.

Accuracy.

Accuracy can be defined as the ratio of the number of total correct recommendations to the total recommendations provided.

10. ADVANTAGES & DISADVANTAGES ADVANTAGES

- Easy to navigate through chatbot.
- Cost saving as it does not always require a person to help the customers.
- It is mostly generic and hence can be built commonly for all types of customers.
- It is constantly available at all times.
- It helps store customer data for later retrieval.
- It can recommend products based on customers' history.

DISADVANTAGES

- Only a limited amount of predefined routes.
- The chatbots are not able to make decisions.
- Customers have to go through multiple steps in order to reach the supportingteam.
- It is mostly not personalized or highly emotive.
- There is a high chance of misunderstanding.
- They need to be maintained.

11. CONCLUSION

The Smart Fashion Recommender Application mostly uses a user's closet to suggest the ideal dress combinations for a user who lacks sense of style. Due to the system's limitations, it might not always recommend the ideal attire for a given situation. depends solely on the clothing in the user's closet. Fashion's strong ties to historical periods are still another factor. However, the system does a remarkable job of helping users develop a sense of fashion, and it can provide the best suggestions based on theuser's clothing. The system is relatively simple for end users to access and utilise because it is implemented as a website. The system's range can be increased by addingthe capacity to recognise distinct apparel designs and patterns, as well as by raising thequantity of instances.

12. FUTURE SCOPE

Although it is a highly efficient chatbot, it could have more options to explore in the future. In the future instead of a text conversation A voice assistant could be used where the customer also uses their voice to converse. This could be an easier and innovative way of recommendation. This could also help older people who find it difficult to text. Chatbots could also be evolved to have seamless and realistic conversations with the customer in order to help business.

To create a useful recommendation system, further research should focus on along with analyses of time series and precise categorization of product photos based on variations in colour, trend, and dress style. Therefore, this study will be extremely benefited to academics who want to use augmented reality and virtual reality elementsto create fashion recommendation chatbot.

13. APPENDIX

SOURCE CODE

REGISTER.HT ML

```
<html>
<head>
<title>Registration</title>
<script type="text/javascript" src="{{ url_for('static', filename = 'js/validateForm.js') }}">
</script>
</head>
<style>
  body {font-family: Arial, Helvetica, sans-serif;}form
  {border: 3px solid #f1f1f1;}

  input[type=text], input[type=password], input[type=email] {width:
    100%;
    padding: 12px 20px;
    margin: 8px 0;
```

```
   display: inline-block;
   border: 1px solid #ccc;
   box-sizing: border-box;
 }

 #button {
   background-color: #04AA6D;
   color: white;
   padding: 14px 20px;
   margin: 8px 0;
   border: none; cursor:
   pointer; width: 100%;
 }

 #button:hover {
   opacity: 0.8;
 }

 .cancelbtn {
   width: auto;
   padding: 10px 18px;
   background-color: #f44336;
 }

 .imgcontainer {
   text-align: center;
   margin: 24px 0 12px 0;
 }

 img.avatar {
   width: 20%;
   border-radius: 50%;
 }
```

```
  .container {
    padding: 16px;
  }

  span.psw {
    float: right;
    padding-top: 16px;
  }

  /* Change styles for span and cancel button on extra small screens */@media
  screen and (max-width: 300px) {
    span.psw {
       display: block;
       float: none;
    }
    .cancelbtn { width:
       100%;
    }
  }
  </style>
<body>
<form action="/register" method="POST" onsubmit="return validate()">
  <div class="container">
     <label for="uname"><b>Email</b></label>
     <input type="email" name="email"></p>
     <label for="psw"><b>Password</b></label>
     <input type="password" name="password" id="password" required></p>
     <label><b>Confirm-Password</b></label>
     <input type="password" name="cpassword" id="cpassword"></p>

     <p><b>First Name: </b><input type="text" name="firstName"></p>
     <p><b>Last Name: </b><input type="text" name="lastName"></p>
     <p><b>Address line 1: </b><input type="text" name="address1"></p>
```

```
        <p><b>Address Line 2: </b><input type="text" name="address2"></p>
        <p><b>Zipcode: </b><input type="text" name="zipcode"></p>
        <p><b>City: </b><input type="text" name="city"></p>
        <p><b>State: </b><input type="text" name="state"></p>
        <p><b>Country: </b><input type="text" name="country"></p>
        <p><b>Phone Number: </b><input type="text" name="phone"></p>

        <p><input type="submit" value="Register" id="button"></p>

        <a href="/loginForm">Login here</a>
    </div>
</form>
</body>
</html>
```

ADD.HTML

```
<!DOCTYPE HTML>
<html>
<head>
<title>Admin</title>
</head>
<body>
<h2>Add items</h2>
<form action="/addItem" method="POST" enctype="multipart/form-data">Name:
    <input type="text" name="name"><br>
    Price: <input type="text" name="price"><br>
    Description: <textarea name="description" rows=3 cols="40"></textarea><br>Image:
    <input type="file" name="image"><br>
    Stock: <input type="text" name="stock"><br>
    Category: <select name="category">
        {% for row in categories %}
            <option  value="{{row[0]}}">{{row[1]}}</option>
        {% endfor %}
    </select><br>
```

```
   <input type="submit">
</form>
</body>
</html>
```

CART.HTML

```
<!DOCTYPE HTML>
<html>
<head>
<title>Your Cart</title>
<link rel="stylesheet" href="static/cart.css">
<link rel="stylesheet" href="static/topStyle.css">
</head>
<body>
<div id="title">
     {% Block Head %} {% End Block %}
</div>
<div id="cartItems">
   <h2 class="shoppingcartText">Shopping Cart</h2>
   <div id="tableItems">
      {% for row in products %}
      <div>
         <hr id="seperator">
         <div id="itemImage">
            <img src={{url_for('static', filename='uploads/'+row[3])}} id="image"/>
         </div>
         <div id="itemName">
            <span id="itemNameTag">{{row[1]}}</span><br> In
            stock<br>
            <a href="/removeFromCart?productId={{row[0]}}"
class="removebutton">Remove</a>
         </div>
         <div id="itemPrice">
```

```
            ${{row[2]}}
          </div>
        </div>
        {% endfor %}
        <hr id="seperator">
        <div id="total">
          <span  id="subtotal">Subtotal</span>  :  ${{totalPrice}}
        </div>
    </div>
</div>
<a  href="/orderplaced"  class="button">Place  Order</a>
</body>
</html>
```

<u>DISPLAY_CATEGORY.HTML</u>

```
<!DOCTYPE HTML>
<html>
<head>
<title>Category:   {{categoryName}}</title>
<link  rel="stylesheet"  href="static/home.css"  />
<link  rel="stylesheet"  href="static/topStyle.css"  />
</head>
<body>
    <div id="title">
        {% Block Head %} {% End Block %}
    </div>
<div>
    <h2  id="shopbycategory">Explore  Category  :  {{categoryName}}</h2>
    {% for itemData in data %}
    <table>
        <tr id="productName">
            {% for row in itemData %}
            <td>
                {{row[1]}}
            </td>
            {% endfor %}
        </tr>
        <tr  id="productImage">
            {% for row in itemData %}
            <td>
                <a   href="/productDescription?productId={{row[0]}}">
                    <img src={{ url_for('static', filename='uploads/' + row[3]) }} id="itemImage" />
                </a>
            </td>
            {% endfor %}
        </tr>
        <tr  id="productPrice">
            {% for row in itemData %}
            <td>
                ${{row[2]}}
            </td>
            {% endfor %}
        </tr>
    </table>
    {% endfor %}

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

PRODUCT_DESCRIPTION.HTML

```
<!DOCTYPE HTML>
<html>
<head>
<title>Product  Description</title>
<link  rel="stylesheet"  href="static/productDescription.css">
<link  rel="stylesheet"  href="static/topStyle.css">

</head>
<body>
   <div id="title">
      {% Block Head %} {% End Block %}
   </div>

<div id="display">
   <div id="productName">
      <h1 class="titles">Product name : {{data[1]}}</h1>
   </div>
   <div>
      <img  src={{url_for('static',  filename='uploads/'+data[4])  }}  id="productImage"/>
   </div>

   <div  id="productDescription">
      <h2  class="titles">Details</h2>
      <table id="descriptionTable">
         <tr>
            <td>Name :</td>
            <td>{{data[1]}}</td>
         </tr>
         <tr>
            <td>Price :</td>
            <td>${{data[2]}}</td>
            </tr>
            <tr>
                <td>Stock :</td>
                <td>{{data[5]}}</td>
            </tr>
        </table>
        <h2  class="titles">Description</h2>
        <p>{{data[3]}}</p>
    </div>
    <div  id="addToCart">
        <a href="/addToCart?productId={{request.args.get('productId')}}"
id="addtocart">Add to Cart</a>
    </div>
</div>
</body>
</html>
```

REMOVE_PRODUCT.HTML

```
<!DOCTYPE HTML>
<html>
<head>
<title>Remove</title>
<link  rel="stylesheet"  href="static/remove.css">
</link>
</head>
<body>
<table>
    {% for i in range(6) %}
    <tr>
        {% for row in data %}
        <td>
            <a   href="/removeItem?productId={{row[0]}}">
            {% if i == 4 %}
            <img src={{ url_for('static', filename='uploads/' + row[i]) }} id="itemImage"
/><br>
         {% else %}
            {{row[i]}}<br>
         {% endif %}
         </a>
      </td>
      {% endfor %}
   </tr>
   {% endfor %}
</table>
</body>
</html>
```

ORDER_PLACED.HTML

```
<html>
   <head>
      <title>Order</title>
   </head>
   <style>
      body,h1{
         text-align: center;
         font-family: system-ui;
         font-weight: lighter;
         font-size: xx-large;
         color: black;
         margin-top: 300px;
      }
      #text{
         margin: auto;
      }
   </style>
   <body>
      <h1>Hurray! Your order has been placed.<br>Continue shopping</h1>
```

```
        <a href="/" id="text">Back to home</a>
    </body>
</html>
```

GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-7777-1658899029.git