IBM – NALAIYA THIRAN PROJECT

SMART FASHION RECOMMENDER APPLICATION

INDUSTRY MENTOR : KRISHNA CHAITANYA

FACULTY MENTOR : MERBIN JOSE P J

TEAM ID	**PNT2022TMID34679**
TEAM LEADER	SHIJU JERCY J
TEAM MEMBER	LIFERIN JENI B
TEAM MEMBER	RAJA GOWSHIKA V
TEAM MEMBER	NIDHIN R S

INDEX

1. **INTRODUCTION**

1. PROJECT OVERVIEW

2. PURPOSE

2. **LITERATURE SURVEY**

1. EXISTING PROBLEM

2. REFERENCES

2. PROBLEM STATEMENT DEFINITION

3. **IDEATION & PROPOSED SOLUTION**

1. EMPATHY MAP CANVAS

2. IDEATION & BRAINSTROMING

3. PROPOSED SOLUTION

4. PROBLEM SOLUTION FIT

4. **REQUIREMENT ANALYSIS**

1. FUNCTIONAL REQUIREMENT

2. NON-FUNCTIONAL REQUIREMENTS

5. **PROJECT DESIGN**

1. DATA FLOW DIAGRAMS

2. SOLUTION & TECHNICAL ARCHITECTURE

3. USER STORIES

6. **PROJECT PLANNING & SCHEDULING**

1. SPRINT PLANNING & ESTIMATION

2. SPRINT DELIVERY SCHEDULE

3. REPORTS FROM JIRA

7. **CODING & SOLUTIONING**

1. FEATURE 1

2. FEATURE 2

3. DATABASE SCHEMA

8. **TESTING**

1. TEST CASES

2. USER ACCEPTANCE TESTING

9. **RESULTS**

1. PERFORMANCE METRICS

10. **ADVANTAGES & DISADVANTAGES**

11. **CONCLUSION**

12. **FUTURE SCOPE**

13. **APPENDIX**

1. SOURCE CODE

2. GITHUB & PROJECT DEMO LINK

1. INTRODUCTION

1. PROJECT OVERVIEW :

The Fashion industry is one of the larger industries around the world. One of the things that has remained constant throughout human civilization is humans covering their bodies with a piece of cloth. Initially, this cloth was worn as protection from the harsh climates of those ages. Later on, as we humans learned to fend for ourselves from the unforgiving climates, the cloth started to serve a different purpose. Fashion these days showcases the individuality of the person. There are many things that can be said about a person based on their fashion sense.

2. PURPOSE:

There is currently no existing system that is capable of recommending clothes based on the occasion. Different occasions call for different clothing. Moreover, a lot of fashion is based on the color combinations of outfits. A person with no or little fashion sense will have a hard time to decide on clothes that leave a lasting impression. The proposed Fashion Recommendation System is intended to be used by individual users in order to store images of the clothes that they own in what is called a digital wardrobe and also to get recommendations by the system on what clothes to wear for a given occasion. The main aim of the project is to recommend the most appropriate clothes for a given occasion based on the clothes existing in the user's wardrobe to relieve the user of the burden of making decisions about what clothing to wear. Such a system should be capable of helping someone who has no fashion sense to wear clothes that leave a good impression on others. The system should be such that it is easily accessible and easy to take advantage of the various features that it provides. One of the features should be the ability to store images that the user uploads into a wardrobe. A wardrobe is a very useful entity that the user can use to view and manage the images of clothes that they have uploaded. This feature can also be used by the recommendation algorithm to recommend the clothes. Another feature is the classification of the type and color of the clothing that is uploaded by the user. The system should be capable of handling the 4 basic clothing types: Shirt, T-Shirt, Pants and Shoes.

2. LITERATURE SURVEY

1. EXISTING PROBLEM:

In the existing system only a simple web application and their rating has been implemented in existing system, An e-commerce product recommendation engine is a piece of technology that displays recommended products to shoppers throughout your store. It uses machine learning to get smarter and show increasingly relevant products to shoppers based on their interests and previous browsing behaviour.

In the existing model is content-based filtering scheme has been employed in existing model **The content-based filtering method** analyzes customer data on the likes and dislikes of each user (cookies allow tracking over multiple visits), then makes recommendations based on the browsing history of that user. The idea behind contentbased filtering is that if you enjoy a certain item, you'll likely also enjoy a similar item. An example of a content-based filtering system would be if you were listening to Pandora and consistently 'liked' downtempo jazz music.

The collaborative-filtering method incorporates data from users who have purchased similar products, then combines that information to make decisions about recommendations. The advantage to this filtering method is that it is capable of making complex recommendations on items such as music or movies without having to 'understand' what the item is. This method of filtering operates under the assumption that users will prefer recommendations that are based on purchases they made in the past. Here's an example: If customer A likes a specific line of products that customer B also likes (assuming they have similar interests), then collaborate-filtering would assume that customer A would like other products that customer B purchased and vice versa.

A hybrid method combines the content-based and collaborative-based methods to incorporate group decisions but focuses the output based on the attributes of a specific visitor. An example of a hybrid filtering system would be how Spotify curates its personalized 'Discover Weekly' playlists. If you've ever listened to a personalized Spotify playlist, it's shocking how accurately they're able to recommend songs based on

what you like. The secret behind how they pull this off is through a complex hybrid filtering system that aggregates data on your listening habits as well as similar users' listening habits, to create a playlist of unique songs that align with your personal taste.

2. REFERENCES:

1. Liu, C., & Wu, X. (2016). Large-scale recommender system with compact latent factor model, 64, 467 475.doi:10.1016/j.eswa.2016.08.009.

2. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734–749. doi:10.1109/TKDE.2005.99.

3. Zhang, Y.; Caverlee, J. Instagrammers, Fashionistas, and Me: Recurrent Fashion Recommendation with Implicit Visual Influence. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1583–1592. [Google Scholar] [CrossRef][Green Version].

4. JH (Janghyun), Baek; John, Tsai; Justin, Shamoun; Muriel, Marable; Ying Cui,Ying;(2020) Amazon Recommender System. 5. Qingqing Tu,Le Dong -An Intelligent Personalized Fashion Recommendation System -2010 6. batuhan aşiroğlu- smart clothing recommendation system with deep learning 2019 3rd International Symposium on MultidisciplinaryYew Cheong Hou and K. S. M. Sahari, "Identifying corners of clothes using image processing method," (2010) In International Conference on Intelligent and Advanced
    a. Systems, Manila, 2010, pp. 1-5.
5. M. Yang and K. Yu, "Real-time clothing recognition in surveillance videos," (2011) In 18th IEEE International Conference on Image Processing, Brussels, 2011, pp. 2937-2940.
6. Y. Wen, X. Liu and B. Xu, "Personalized Clothing Recommendation Based on Knowledge Graph,"(2018) 2018 International Conference on Audio, Language and Image Processing (ICALIP), Shanghai, pp. 1-5.

7. O'Connell, L. (n.d.). Topic: Apparel Market Worldwide. Retrieved August 30, 2020, from https://www.statista.com/topics/5091/apparel- market-worldwide/

8. Zhang, Yan & Liu, Xiang & Shi, Yunyu & Guo, Yunqi & Xu, Chaoqun & Zhang, Erwen & Tang, Jiaxun & Fang, Zhijun. (2017). Fashion Evaluation Method for
    a. Clothing Recommendation Based on Weak Appearance Feature. Scientific

       Programming. 2017. 1-12. 10.1155/2017/8093057.

3. PROBLEM STATEMENT DEFINITION

The personal information collected by recommenders raises the risk of unwanted exposure of that information. Also, malicious users can bias or sabotage the recommendations that are provided to other users.In recent years, the textile and fashion industries have witnessed an enormous amount of growth in fast fashion. On e-commerce platforms, where numerous choices are available, an efficient recommendation system is required to sort, order, and efficiently convey relevant product content or information to users.

1. The problem of the work is to design static web applications deployments with customer deployment
2. Lack of interaction between application and user
3. User need to navigate across multiple pages to choose right product
4. Confusion in choosing product
5. Lack of sales
6. Complex User Interface.
7. Lack of proper guidance.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
1	Having a lack of data	Overcome those problems	The good recommendation is not taking in progress	More item and user data recommender system has to work with	To get good recommendations

2	Always changing user preferences	Sort out the issue here is that while today I have a particular intension when browsing	It is very challenging to handle those things	Of its user preference	challengable

3. IDEATION & PROPOSED SOLUTION

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

In this project you will be working on two modules :

1. Admin and
2. User

ADMIN:

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

USER :

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

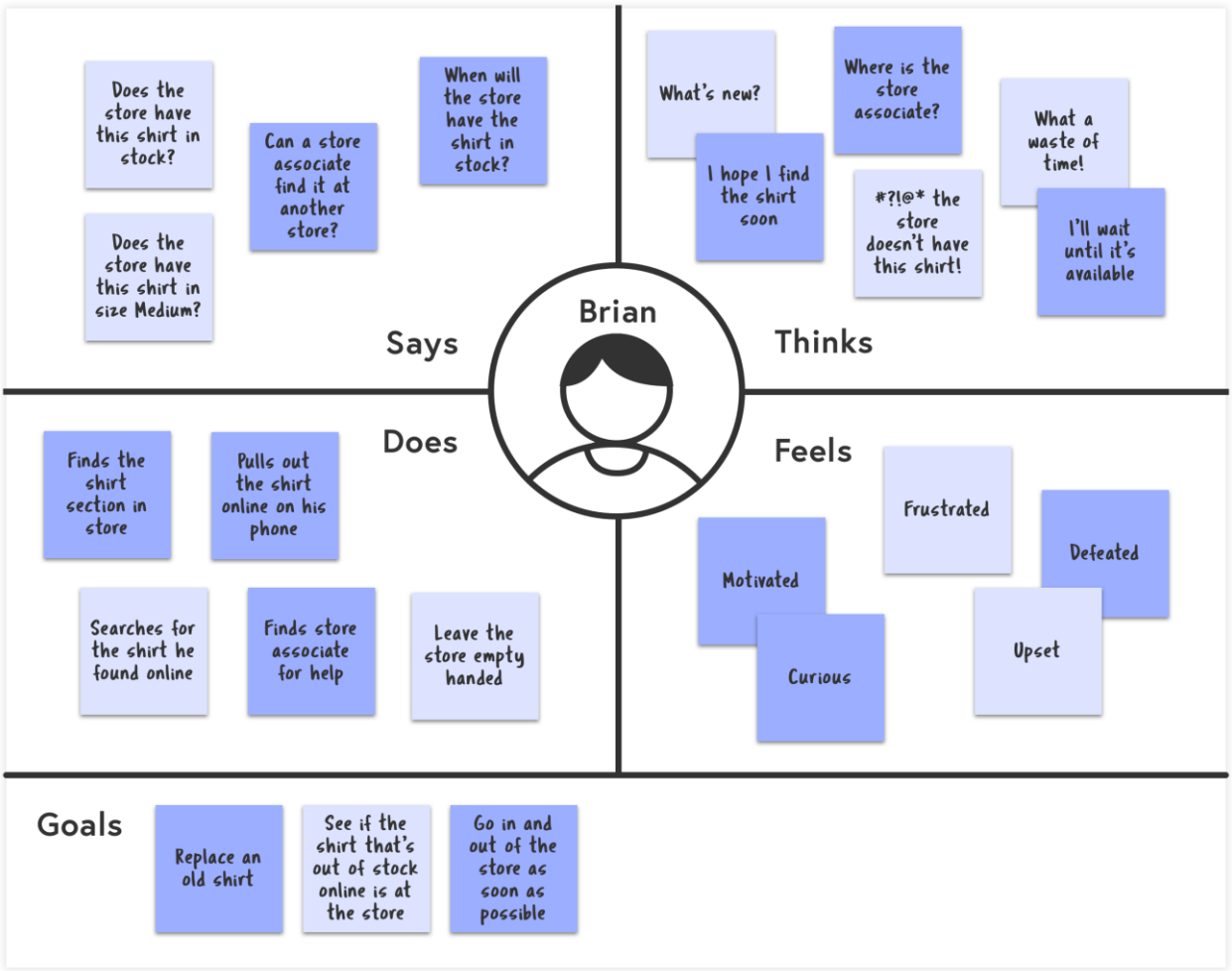
FEATURES OF CHATBOT :

i. Using chatbot we can manage user's choices and orders. ii. The chatbot can give recommendations to the users based on their interests.
iii. It can promote the best deals and offers on that day. iv. It will store the customer's details and orders in the database.

v. The chatbot will send a notification to customers if the order is confirmed.
vi. Chatbots can also help in collecting customer feedback.

1. EMPATHY MAP CANVAS:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges. An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers.



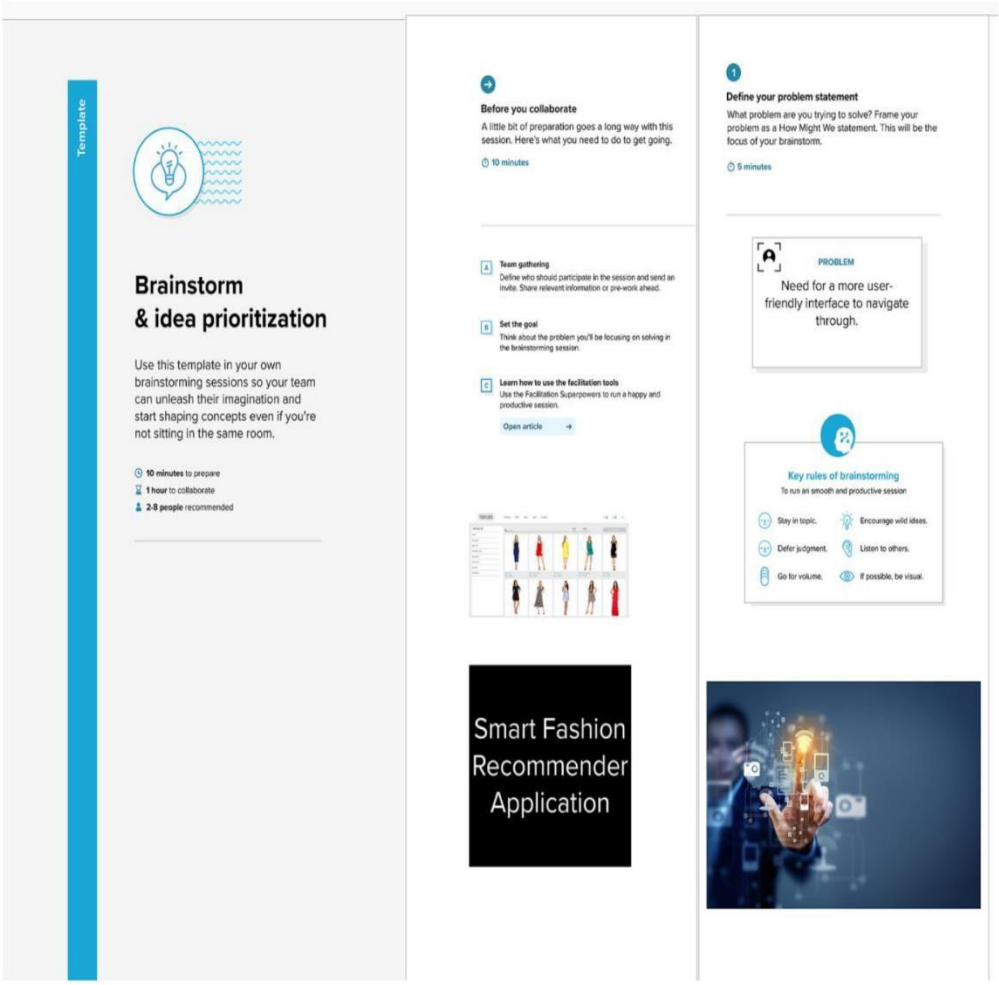
2. IDEATION & BRAINSTROMING:

i. A group problem-solving technique that involves the spontaneous contribution of ideas from all members of the group.

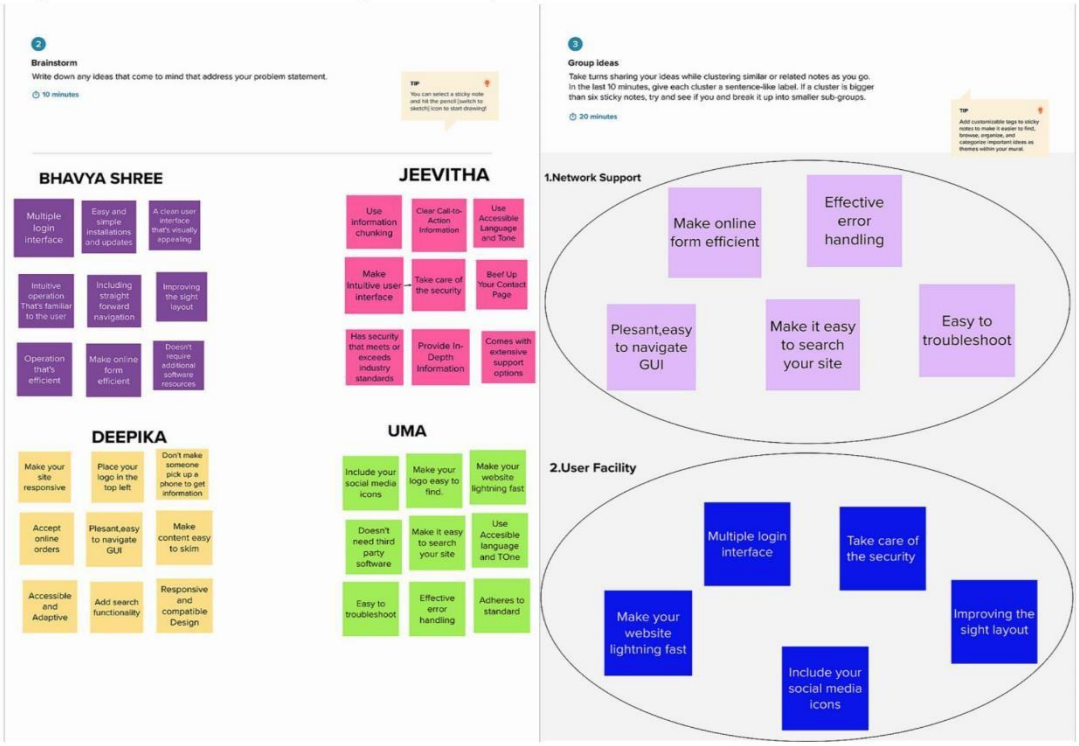
ii. The mulling over of ideas by one or more individuals in an attempt to devise or find a solution to a problem.

Brainstorm & Idea Prioritization Template:

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



Step-2: Brainstorm, Idea Listing and Grouping:

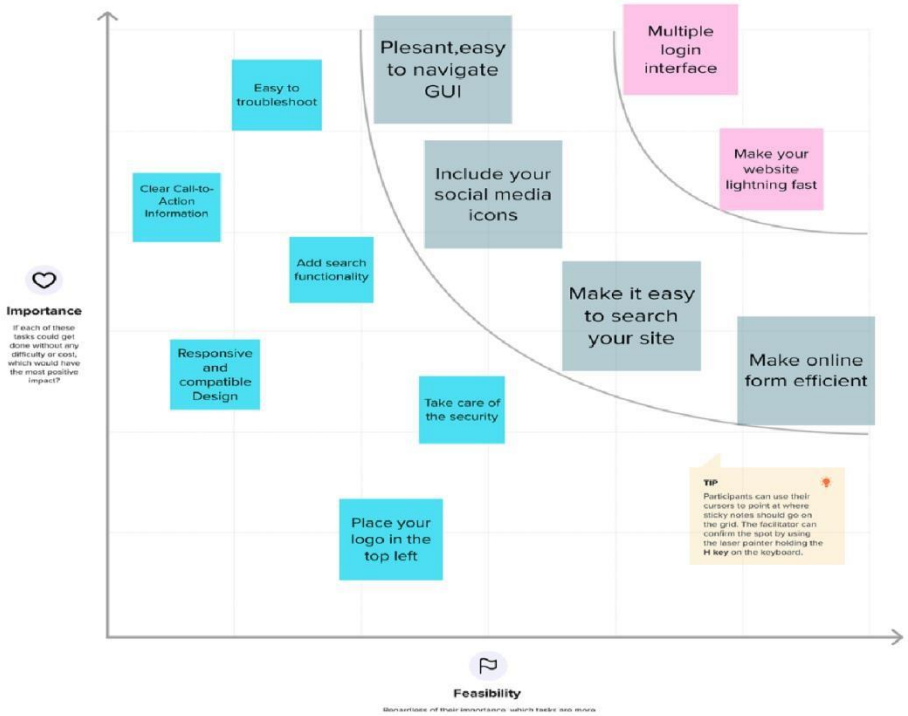


4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

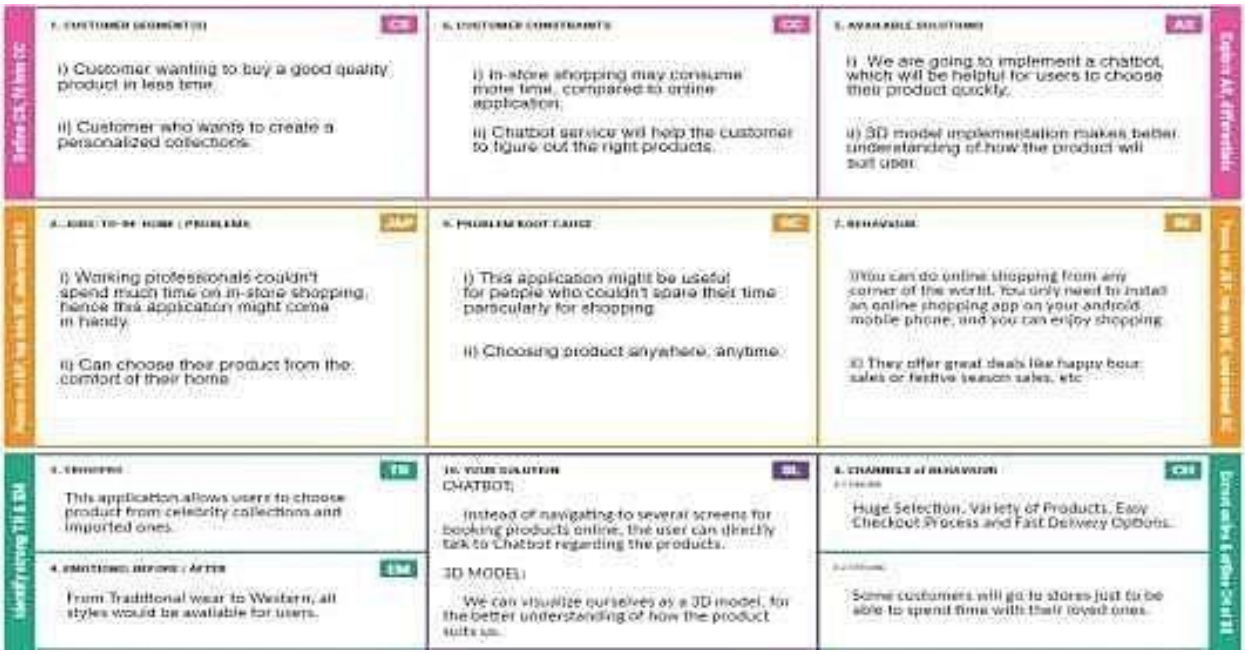


3. PROPOSED SOLUTION:

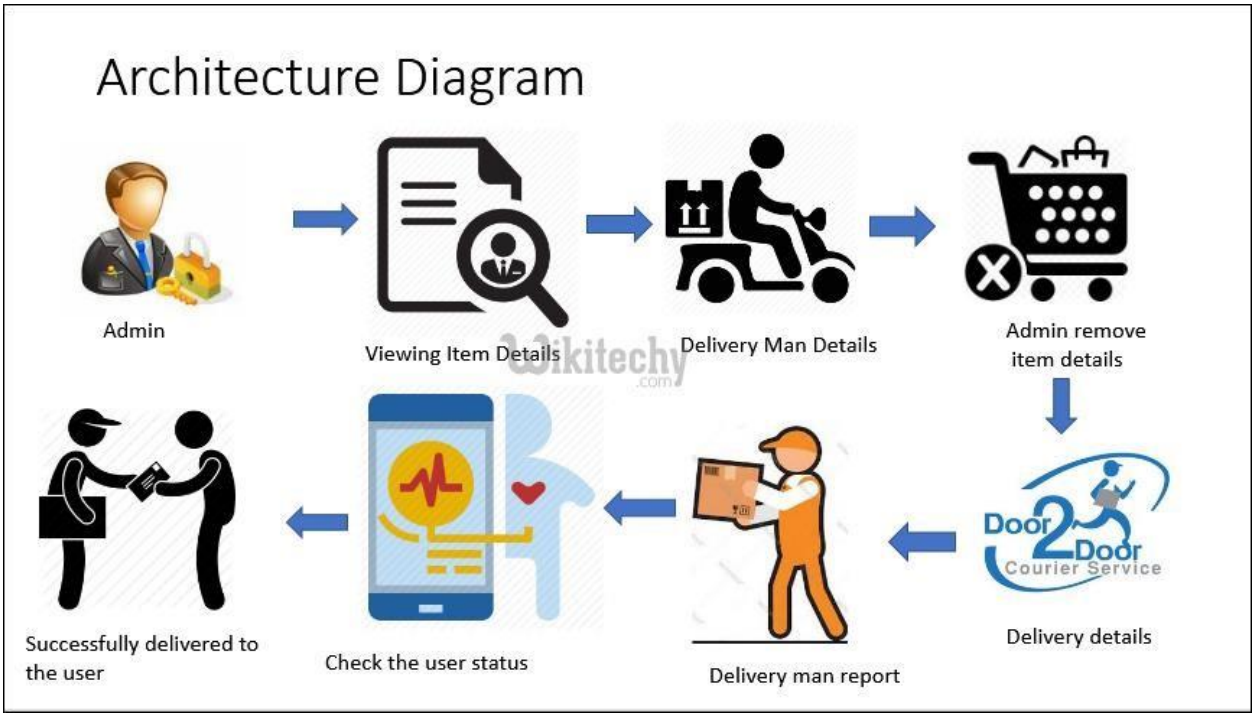
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Globally, fashion creates 40 million tons of textile waste annually, according to the Ellen MacArthur Foundation, and much of it is unnecessary — despite cotton's recyclable nature, less than 1% of cotton materials were recycled in 2020.
2.	Idea / Solution description	Clothing that monitors the wearer's physical condition. Smart shirts and body suits provide biometric data, such as pulse rate, temperature, muscle stretch, heart rhythm and physical movement, and the data are transmitted via Bluetooth to an app inreal-time
3.	Novelty / Uniqueness	Wearable devices can monitor body temperature, blood pressure and location. Wearable smart clothing can also be fabricated with sensors to monitor health and fitness. There are numerous medical uses for smart clothing. Wearable sensors can monitor vital signs, such as blood glucose levels, heart rate and oxygen levels.
4.	Social Impact / Customer Satisfaction	1. Justify the Sale With Social Proof. 2. Give Shoppers a Gift & Surprise Them With Another One. 3. Take Notes.

		4. Train Staff to Handle Disgruntled Customers.
5.	Business Model (Revenue Model)	The business models were related to the concepts identified in the co-creation workshop, which preceded the organization of the business model one. The projects which presented the best potential were further explored, and translated into viable products or services.
6.	Scalability of the Solution	Recommendation systems have the potential to explore new opportunities for retailers by enabling them to provide customized recommendations to consumers based on information retrieved from the Internet

4. PROBLEM SOLUTION FIT



Architecture Diagram



4.REQUIREMENT ANALYSIS

1. FUNCTIONAL REQUIREMENT:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	• Registration through registration Form , Gmail ,mobile number.
FR-2	User Confirmation	• User confirmation via Email and email – OTP.
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	Checking Availability	• Item availability in specific locations.
FR-5	Shopping cart	• My cart button , Add-to-cart button , Remove-fromcart button.
FR-6	Super-fast checkout	• Online transfer, • Credit card payment, • Paying with mobile wallets
FR-7	Checking the shipping status	• Option to easily check the shipping status of items ordered in the store.

2. NON-FUNCTIONAL REQUIREMENTS:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	If people search on google for a product you offer it should be on the first page of result and good quality images that will attract buyers.
NFR-2	**Security**	This Application will collect a lot of users' private information to complete a purchase (banking, shipping/home address, email, etc.) Data protection is the priority.
NFR-3	**Reliability**	Ability of the software to perform critical tasks like collecting and securing customer data, providing payment gateway to function correctly in a given environment, for a particular amount of time.

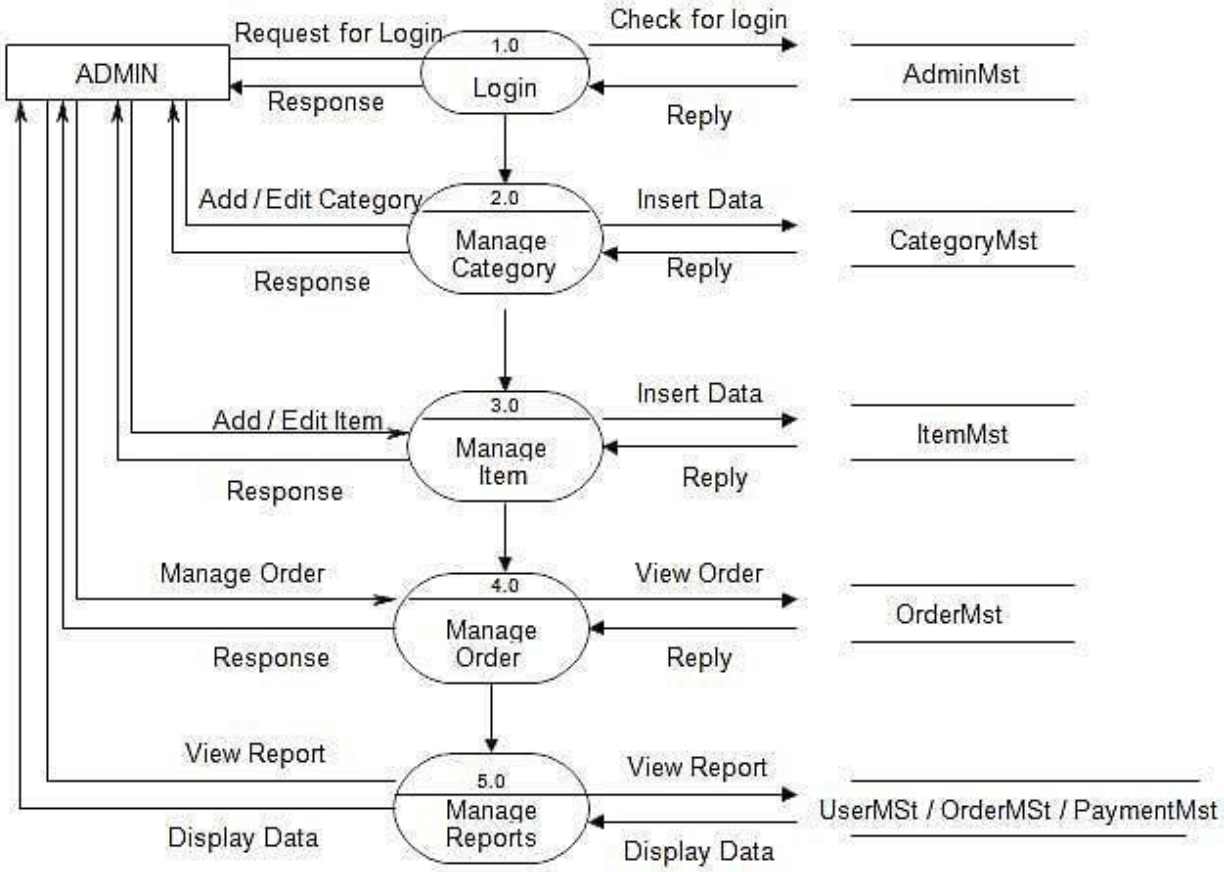
NFR-4	**Performance**	*Speed up the webpage and Site optimization based on the data analysis. *Good use of the product description.
NFR-5	**Availability**	The administrator needs to look up the stock availability in the database
NFR-6	**Scalability**	Having a plan to handle demand peaks. Avoid downtime, preserve the customer experience, and ensure deliveries go out on time at all costs. Chatbots to provide scalable customer Support.

5.PROJECT DESIGN

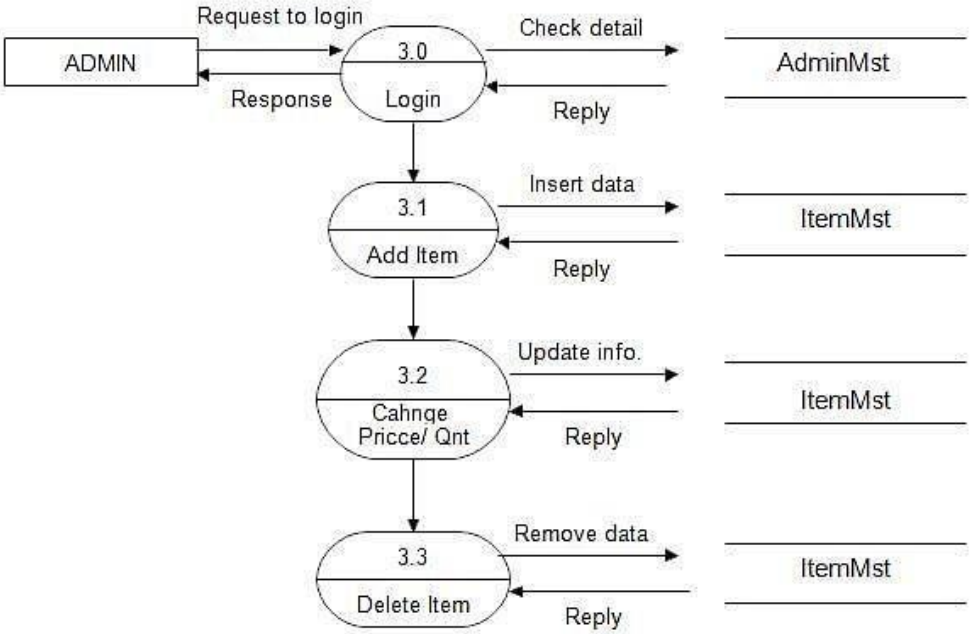
1. DATA FLOW DIAGRAMS:

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

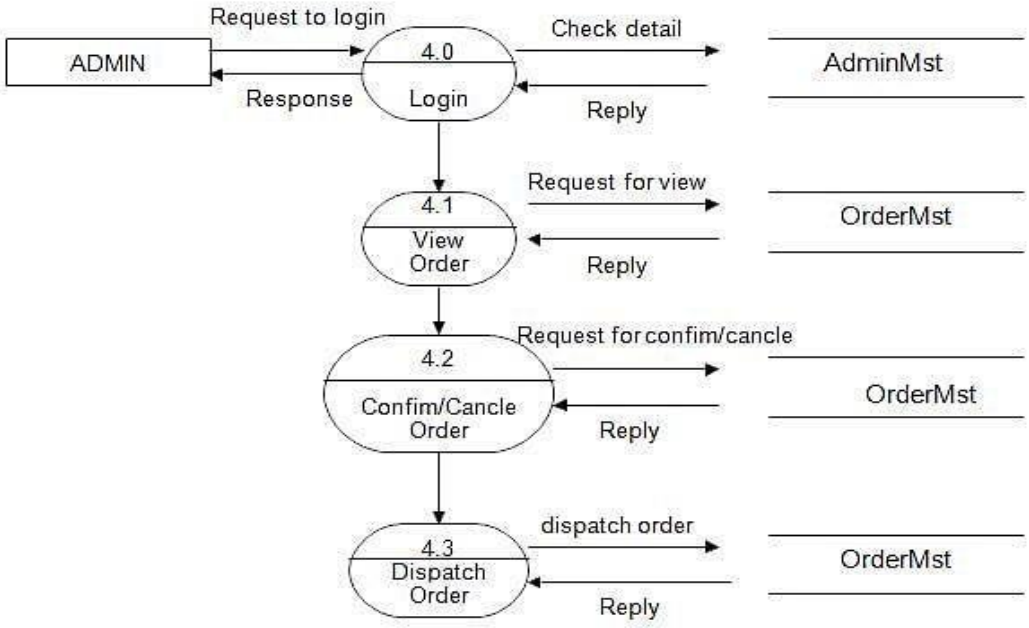
Admin Side DFD - 1st Level



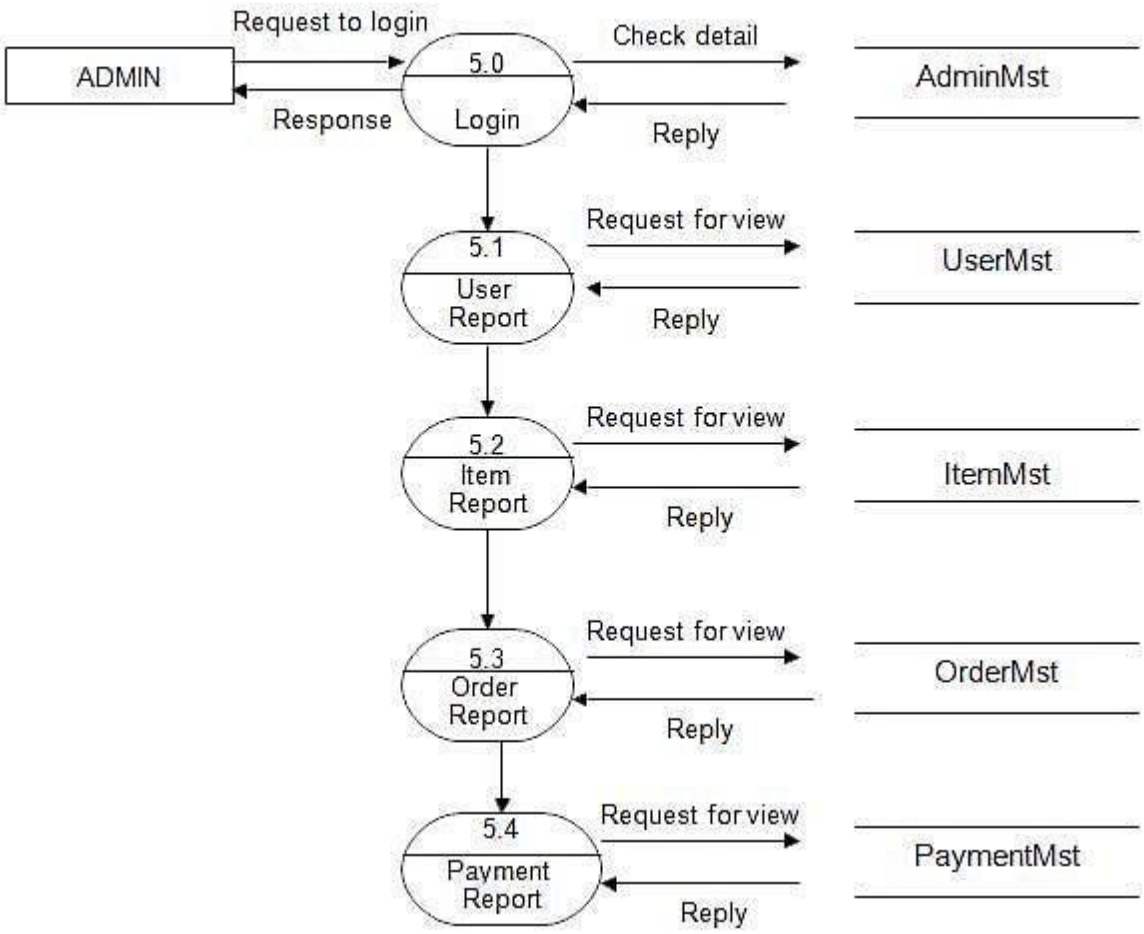
2nd Level Admin DFD - (3.0)



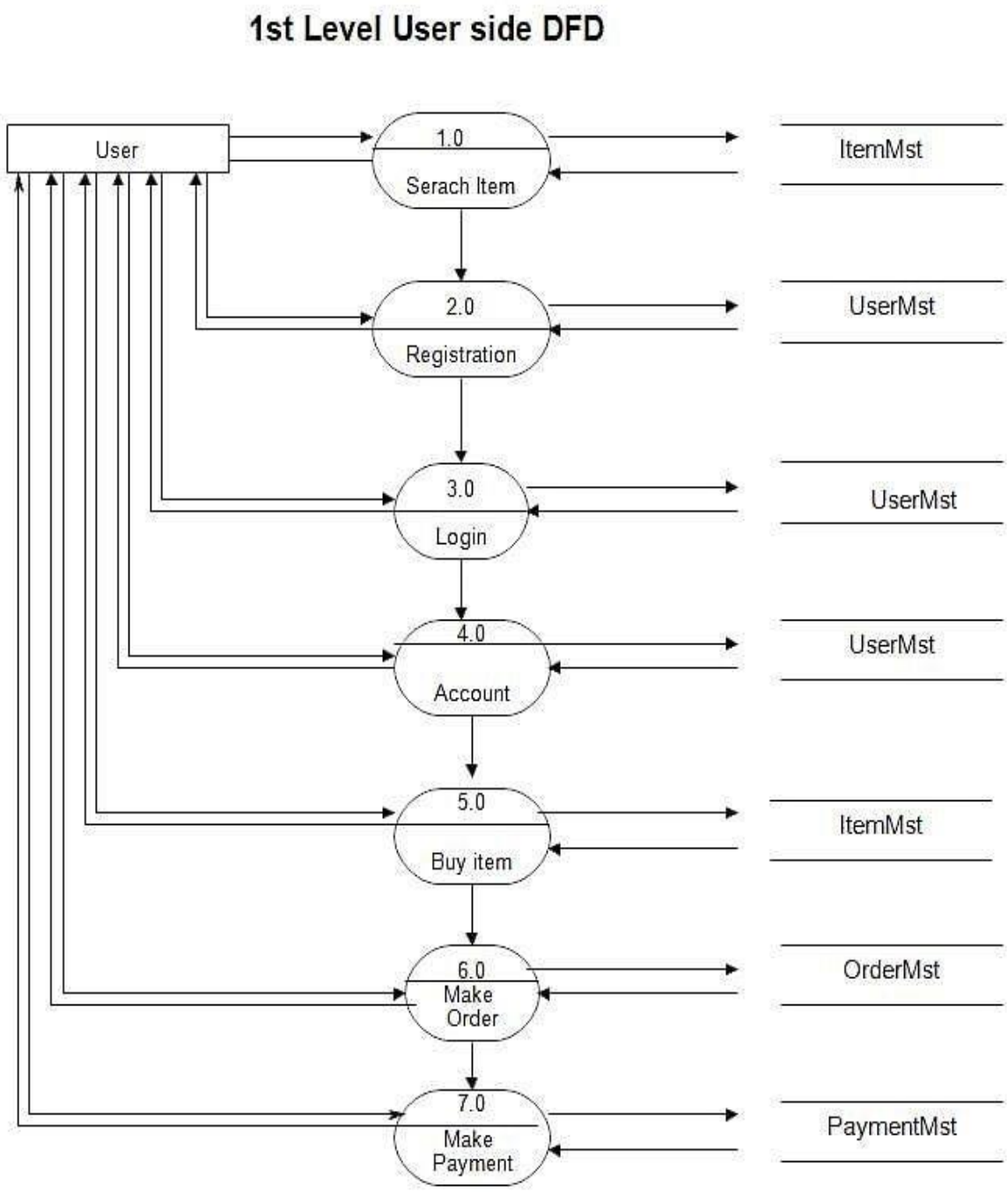
2nd Level Admin DFD - (4.0)



2nd Level Admin DFD - (5.0)



User side Data flow Diagram



2. SOLUTION & TECHNICAL ARCHITECTURE:

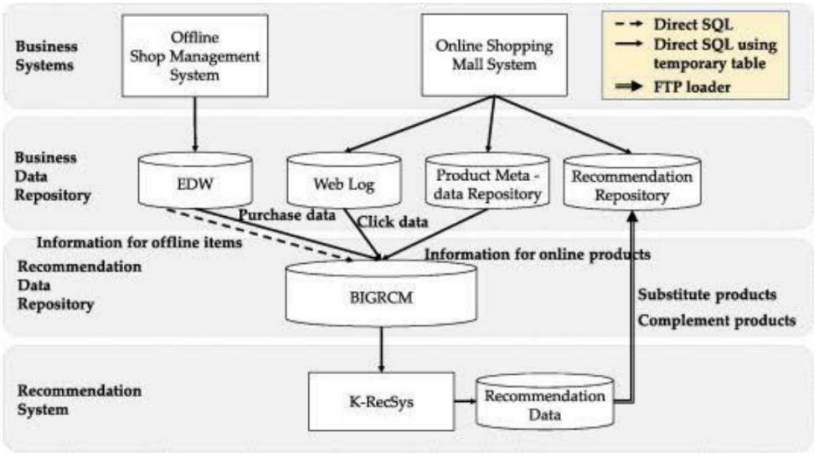
We have developed a new innovative solution through which you can directly do your online shopping based on your choice without any search. It can be done by using the chatbot. In this project you will be working on two modules:

1. Admin

2. User

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.

Solution Architecture Diagram:



Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	The Customer/ user interacts with the application through chatbot.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	The Customer will register and login to the website.	Javascript
3.	Application Logic-2	The Customer can sign In via Google, LinkedIn, Facebook etc.	Javascript
4.	Cloud Database	The User data and products data will be stored in IBM cloud storage.	IBM DB2, IBM Cloudant etc.
5.	File Storage	The product details like price, quantity, and other details etc will be available in cloud bucket.	IBM Block Storage or Other Storage Service or Local Filesystem
6.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud	Local, Cloud Foundry, Kubernetes, etc.

3. USER STORIES

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 login 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	I can receive confirmation	High	Sprint-1

			have registered for the particular menu	email & click confirm		
		USN-3	As a user, I can view the menus through Facebook and other social media platforms	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	USN-1	I can see the dashboard with reviews, ratings, menus etc…			
Customer (Web user)	Registration	USN-1	As a user ,I can sign up for the dishes which was ordered 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	
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 users menus and their order queries	I can authorise transactions and products.	Medium	

6. PROJECT PLANNING & SCHEDULE

1. SPRINT PLANNING & ESTIMATION:

MILESTONES	ACTIVITY	DESCRIPTION

Project development phase	Delivery of Sprint-1,2, 3, 4	To develop the code and submit the develop the code after completion of testing
Setting-up app environment	Create IBM cloud account	Sign up IBM cloud account
	Create flask project	Getting started with the flask to create project
	Install IBM cloud cli	Install IBM command line interface (CLI)
	Docker CLI installation	Installing docker CLI
	Create an account in sendgrid	Create an account in sendgrid Use service as e-mail integration to the application for sending emails
Implementing web application	Create UI to interact with the application	Create UI 1. registration page 2. login page 3. view the products page 4. add products page
	Create IBM DB2 and connect with the Python	Create IBM DB2 in IBM cloud and link with the Python
Integrating sendGrid service	SendGrid integration with the Python	To send emails from the application we need to integrate the SendGrid services
Developing a chatbot	Building a chatbot and integrate it with the application	Build the chatbot and integrate it to the flask application
Deployment of the app in the IBM cloud	Containerize the app	Create a docker image of the application, in addition, to push it to the IBM container registry
	Upload image to IBM container registry	Upload the image to IBM container registry
	Deploy in in Kubernetes cluster	Once the image is uploaded to IBM container registry deploy the image toward IBM Kubernetes cluster

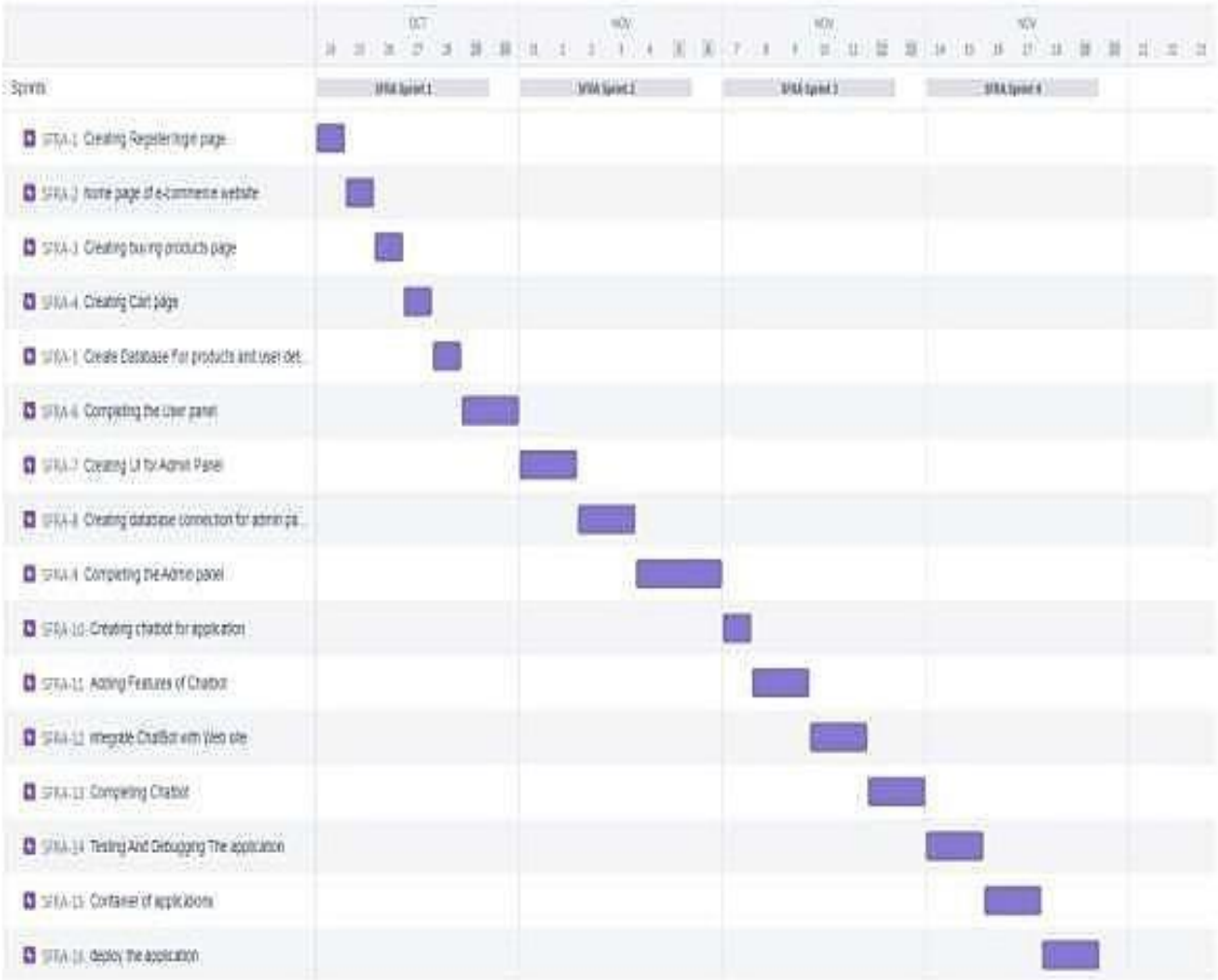
Ideation phase	Literature survey	Literature survey on the selected project and information gathering
	Empathy map	Prepare an empathy map to capture the user pains and gains, prepare a list of problem statement
	Ideation	Organizing the brainstorming session and prioritize the top three ideas based on feasibility hand importance
Project design phase 1	Proposed solution	Prepare proposed solution document which includes novelty, feasibility of ideas, business model, social impact, scalability of solution
	Problem solution fit	Prepare problem solution fit documents
	Solution architecture	Prepare solution architecture document
Project design phase 2	Customer journey map	Prepare a customer journey map to understand the user interactions and experience with the application
	Functional requirements	Prepare functional and non-functional necessity document
	Data flow diagram	Prepare data flow diagram and user stories
	Technology architecture	Draw technology architecture diagram
Project planning phase	Milestones and activity list	Prepare milestones and an activity list of the project
	Sprint delivery plan	Prepare spring delivery plan

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	19 Nov 2022	20	19 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	19 Nov 2022	20	19 Nov 2022
Sprint-3	20	6 Days	05 Nov 2022	19 Nov 2022	20	19 Nov 2022
Sprint-4	20	6 Days	07 Nov 2022	19 Nov 2022	20	19 Nov 2022

3. REPORTS FROM JIRA:

Burndown Chart:



7. CODING & SOLUTIONING

1. FEATURE-1: Login.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
   <!-- Design by foolishdeveloper.com -->
   <title>Login</title>

   <link rel="preconnect" href="https://fonts.gstatic.com">
   <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/5.15.4/css/all.min.css">
   <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;600&dis
play=swap" rel="stylesheet">
   <!--Stylesheet-->
   <style media="screen">
      *,
      *:before,
*:after {
padding: 0;
margin: 0;
         box-sizing: border-box;
      }

      body {
         background-color: #50469cfc;
      }

      .background {
width: 430px;
```

```
height: 520px;
position: absolute;
        transform: translate(-50%, -50%);
left: 50%;          top: 50%;
    }

    .background .shape {
height: 200px;
        width: 200px;
        position: absolute; border-radius:
        50%;
    }

    .shape:first-child {
        background: linear-gradient(#1845ad,
            #23a2f6);
left: -80px;          top:
-80px;
    }

    .shape:last-child {
        background: linear-gradient(to right,
            #ff512f,
#f09819);          right: -
30px;
        bottom: -80px;
    }

    form {
height: 520px;
width: 400px;
        background-color: rgba(255, 255, 255, 0.13);
        position: absolute;
transform: translate(-50%, -50%);
        top: 50%;          left: 50%;          border-
radius: 10px;          backdrop-filter: blur(10px);
border: 2px solid rgba(255, 255, 255, 0.1);
```

```
box-shadow: 0 0 40px rgba(8, 7, 16, 0.6);
padding: 50px 35px;
      }

      form * {
         font-family: 'Poppins', sans-serif; color:
         #ffffff;
         letter-spacing: 0.5px;
         outline: none;
         border: none;
      }

      form h3 {
font-size: 32px;
font-weight: 500;
line-height: 42px;
         text-align: center;
      }
       label {
display: block;
margin-top: 30px;
font-size: 16px;
         font-weight: 500;
      }

      input {
display: block;
height: 50px;
width: 100%;
         background-color: rgba(255, 255, 255, 0.07);
         border-radius: 3px;
padding: 0 10px;          margin-
top: 8px;          font-size: 14px;
         font-weight: 300;
      }

      ::placeholder {
color: #e5e5e5;
```

```
    }

    button {
        margin-top: 30px;
        width: 100%; background-color:
        #ffffff;
        color: #080710;
padding: 15px 0;          font-
size: 18px;          font-weight:
600;          border-radius: 5px;
        cursor: pointer;
    }

    .social {
        /*margin-top: 30px;*/
        display: flex;
    }

    .social div {          background: red;
width: 150px;          border-radius: 3px;
padding: 5px 10px 10px 5px;          background-
color: rgba(255, 255, 255, 0.27);
        color: #eaf0fb;
text-align: center;
    }

    .social div:hover {
        background-color: rgba(255, 255, 255, 0.47);
    }

    /* .social .fb {
        margin-left: 25px;
    }

    .social i {
margin-right: 4px;
    }*/
  </style>
```

```
</head>

<body>
   <div class="background">
      <div class="shape"></div>
      <div class="shape"></div>
   </div>
   <form method="POST">
      <h3>{{my}}</h3>

      <label for="username">Username</label>
      <input type="text" name="question1_field" placeholder="Email or Phone"
id="username">

      <label for="password">Password</label>
      <input type="password" name="lpas" placeholder="Password" id="password">

      <button type="submit">Log In</button>

      <div class="social">

         <a href="/register"
            style="position:absolute;top:85%;right:
159px;fontsize:larger;color:#f9f100;">Register</a>

   </form>
</body>

</html>
```

Register 2:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
   <link rel="stylesheet" href="
```

```
        https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstra
p.min.css">
   <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.mi
n.js">


</head>
<body>
   <form action="" method="POST">
      <div class="mb-3">
         <label class="form-label">Name</label>
         <input type="text" name='username' class="form-control"  >

      <div class="mb-3">
        <label class="form-label">Email address</label>
        <input type="text" name='email' class="form-control"
ariadescribedby="emailHelp">

      <div class="mb-3">
        <label   class="form-label">Password</label>
        <input type="text" name='inves' class="form-control" >
      </div>
      <div class="mb-3">
         <label  class="form-label">Password</label>
         <input type="text" name='question1_field' class="form-control" >
        </div>
      <input type="submit" value="register" class="btn btn-primary">
</form>

</body> </html>
```

Register:

```
!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Signup</title>
  <link rel="stylesheet" href="
       https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstra
p.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.mi
n.js">


</head>

<body>
  <section class="h-100" style="background-color: #028ecf;">

     <div class="container h-100">
       <div class="row d-flex justify-content-center align-items-center h-100 p-
xl5">
          <div class="col-lg-12 col-xl-11">
            <div class="card text-black" style="border-radius: 25px;">
              <div class="card-body">
                <div class="row justify-content-center">
                  <div class="col-md-10 col-lg-6 col-xl-5 order-2 order-lg-1">

                    <p class="text-center h1 fw-bold mb-5 mx-1 mx-md-4
mt4">Sign up</p>

                    <form method="POST" action="" class="mx-1 mx-md-4">

                      <div class="d-flex flex-row align-items-center mb-4">
                        <i class="fas fa-user fa-lg me-3 fa-fw"></i>
                        <div class="form-outline flex-fill mb-0">
                          <input type="text" id="form3Example1c"
class="formcontrol" name="username" />
                          <label class="form-label" for="form3Example1c">Your
Name</label>
                        </div>
                      </div>
```

```
                        <div class="d-flex flex-row align-items-center mb-4">
                           <i class="fas fa-envelope fa-lg me-3 fa-fw"></i>
                           <div class="form-outline flex-fill mb-0">
                              <input type="text" name="email"  class="form-control"
/>
                              <label class="form-label" for="form3Example3c">Your
Email</label>
                           </div>
                        </div>

                        <div class="d-flex flex-row align-items-center mb-4">
                           <i class="fas fa-lock fa-lg me-3 fa-fw"></i>
                           <div class="form-outline flex-fill mb-0">
                              <input type="text" name="question1_field"
id="form3Example4c" class="form-control" />
                              <label class="form-label"
for="form3Example4c">Password</label>
                           </div>
                        </div>

                        <div class="d-flex flex-row align-items-center mb-4">
                           <i class="fas fa-key fa-lg me-3 fa-fw"></i>
                           <div class="form-outline flex-fill mb-0">
<input type="text" id="form3Example4cd" name="cinves" class="form-
control" />
                              <label class="form-label"
for="form3Example4cd">Repeat your
                                 password</label>
                           </div>
                        </div>

                        <div class="form-check d-flex justify-content-center mb-5">
<input class="form-check-input me-2" type="checkbox" value=""
                              id="form2Example3c" />
<label class="form-check-label"
for="form2Exaemailmple3">
```

```
                                I agree all statements in <a href="#!">Terms of
service</a>
                              </label>
                            </div>

                            <div class="d-flex justify-content-center mx-4 mb-3 mb-
lg4">
                              <button type="submit" class="btn btn-primary
btnlg">Register</button>
                            </div>
                          </form>

                        </div>
                        <div class="col-md-10 col-lg-6 col-xl-7 d-flex align-items-center
order-1 order-lg-2">

                          <img src="https://mdbcdn.b-
cdn.net/img/Photos/newtemplates/bootstrap-registration/draw1.webp"
                            class="img-fluid" alt="Sample image">

                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>

  </section>
  <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min
.js"
    integrity="sha384-
oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3"
crossorigin="anonymous"></script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
integrity="sha384-
```

```
IDwe1+LCz02ROU9k972gdyvl+AESN10+x7tBKgc9I5HFtuNz0wWnPclzo6p9vxnk"
crossorigin="anonymous"></script>
</body>

</html> Index:
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.mi
n.js">

  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N"
crossorigin="anonymous">
  <title>javascript</title>
  <link rel="stylesheet" href="{{url_for('static',filename='css/main.css')}}">
</head>

<body>
  <section class="bg-white position-sticky border-bottom border-light" style="top:
0; z-index: 20;">
    <nav class="navbar navbar-expand-lg navbar-light">
      <a class="navbar-brand" href="#"><img src="{{user_image}}" alt=""></a>
      <ul class="navbar-nav ml-auto">
        <li class="nav-item ">
          <a class="nav-link active" style="color:#ffffff;" href="/index">HOME <span
class="sr-only">(current)</span></a>
        </li>
        <li class="nav-item">
          <a class="nav-link" style="color:#ffffff;" href="/register">REGISTER</a>
        </li>
```

```
      <li class="nav-item">
        <a class="nav-link" style="color:#ffffff;" href="/login">LOGIN</a>
      </li>

    </ul>
    </div>
  </nav>
 </section>
 <section class="section-main">
  <div class="container">
    <div class="row mx-auto  ">


      <div class="col d-flex justify-content-center align-items-center">
        <div class="box">
          <div class="title">
            <span class="block"></span>
            <h1 style="font-weight: bold;color: #f87a04;font-size:
90px;">Smart<span></span></h1>
          </div>
        </div>
      </div>

    </div>

  </div>
  <div class="row mx-auto my-4 ">
    <div class="col d-flex justify-content-center align-items-center">
      <h2 style="font-weight: bold;color: #8d1600;font-size: 90px;">Fashion</h2>
</div>

  </div>
  <div class="row mx-auto my-4 ">
    <div class="col d-flex justify-content-center align-items-center">
<h3 style="font-weight: bold;color: #007ea7;font-size:
90px;">Recommender</h3>
    </div>
  </div>
```

```
  </div>
  <div class="row mx-auto ">
    <div class="col d-flex justify-content-center align-items-center">
<h4 style="font-weight: bold;color: #00916e;font-size:
90px;">Application</h4>
    </div>

  </div>
  </div>
 </section>
 <section class="section-footer">
  <div class="container-fluid">
   <div class="row">
    <div class="col">
     <p>Design and Developed By:<br>Sherin Dhanya & Team </p>
    </div>
   </div>
  </div>
 </section>

</body>

</html>
```

Chatbot code:

```
<!DOCTYPE html>
<html lang="en">
<head>

  <title>Chatbot prototype</title>
</head>
<body>

  <h1 style="margin-left:25% ; margin-top:5% ">Fashion recommender Chatbot
prototype</h1>    <script>
      window.watsonAssistantChatOptions = {          integrationID: "9f1ae07d-
be7d-4243-a88b-3101eb53e043", // The ID of this integration.          region: "au-
syd", // The region your integration is hosted in.
```

```
        serviceInstanceID: "7c0b5e17-48c0-4aec-9ed6-f1e8cdd5222d", // The ID of
your service instance.          onLoad: function(instance) { instance.render(); }
      };
      setTimeout(function(){
        const t=document.createElement('script');
        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
      });
    </script>
</body> </html>
```

Page code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<!--  Load CSS in Flask with html-->
   <link rel="stylesheet" href='/static/style.css' />

<!--  Load JavaScript in Flask with html-->
 <!-- <script>
  alert("Page has been loaded successfully")
 </script>-->
<script src="/static/app.js" defer></script>

  <script src="/static/response.js" defer></script>
 </head>
 <body>
  <div class="cnt">
      <center><h2>Hi {{vrb}}</h2></center>

      </div>
      <marquee>Welcome to Cedric Fashion Store</marquee>

      <div class="slideshow-container">

      <div class="mySlides fade">
        <div class="numbertext">1 / 3</div>
       <a href="/shirts" target="_blank"> <img src="{{url_for('static',
filename='shirt.jpg')}}" height="500px" width="350px"></a>
```

```
      <div class="text">Caption Text</div>
    </div>

    <div class="mySlides fade">
      <div class="numbertext">2 / 3</div>
      <a href="/chudr" target="_blank"><img src="{{url_for('static',
filename='chu.jpg')}}" height="500px" width="300px"></a>
      <div class="text">Caption Two</div>
    </div>

    <div class="mySlides fade">
      <div class="numbertext">3 / 3</div>
      <a href="/jk" target="_blank"><img src="{{url_for('static',
filename='jkin.jpg')}}" height="500px" width="300px"></a>
      <div class="text">Caption Three</div>
    </div>

    </div>
    <br>

    <div style="text-align:center">
      <span class="dot"></span>
      <span class="dot"></span>
      <span class="dot"></span>
    </div>


    <div class="container">
      <div class="chat-header">
        <div class="logo">
          <img src="{{url_for('static', filename='cbts.jpg')}}" alt="cwt" />
</div>
        <div class="title">Let's Chat</div>
      </div>
      <div class="chat-body"></div>
      <div class="chat-input">
        <div class="input-sec">
          <input type="text" id="txtInput" placeholder="Type here" autofocus />
</div>
        <div class="send">
```

```
            <img src="{{url_for('static', filename='send.png')}}" alt="send"
height="10px"/>
          </div>
        </div>
      </div>



      <script>        let
slideIndex = 0;
showSlides();

      function showSlides() {
        let i;
        let slides = document.getElementsByClassName("mySlides");
let dots = document.getElementsByClassName("dot");
        for (i = 0; i < slides.length; i++) {
slides[i].style.display = "none";
        }
        slideIndex++;
        if (slideIndex > slides.length) {slideIndex = 1}
for (i = 0; i < dots.length; i++) {            dots[i].className
= dots[i].className.replace(" active", "");
        }
        slides[slideIndex-1].style.display = "block";
dots[slideIndex-1].className += " active";
        setTimeout(showSlides, 3000); // Change image every 2 seconds
      }
      </script>

  </body>
</html>
```

Connect IBM DB2:

```
from flask import Flask,render_template,request,flash,url_for,redirect,session
import ibm_db.connector #from flask_mysqldb import SQL
import os
import json
```

```
image=os.path.join('static') app=Flask(__name__)
app.config['UPLOAD_FOLDER']=image


#app.config["SQL_HOST"]="localhost"
#app.config["SQL_USER"]="root"
#app.config["SQL_PASSWORD"]=""
#app.config["SQL_DB"]="flask_registration"
#sql=SQL(app)

mydb=sql.connector.connect(
host='localhost',
user='root',     password='',
   database='oneline'
)



@app.route('/connect') def
connect():
#display=mydb
   return render_template("connect.html")
@app.route('/index') def
index():
   myimg=os.path.join(app.config['UPLOAD_FOLDER'],'college.png')
return render_template("index.html",user_image=myimg)
@app.route('/login',methods=['GET','POST'])
def login():     if request.method == 'POST':
#details=request.form
lgname=request.form['lname']
pas=request.form['lpas']
mycursor=mydb.cursor()
       sql="""SELECT * FROM `registration` WHERE `name`=%s AND `pas`=%s
"""
       mycursor.execute(sql,(lgname,pas))
myres = mycursor.fetchall()         for x
in myres:              name = x[1]
pss = x[3]             print(name,pss)
```

```
        #mycursor.execute("SELECT *  from registration WHERE name=abish AND
pass=1234")



            if name == lgname or pss == pas:
                #lgname=myresult['name']
                #pas=myresult['pas']
                #name=lgname
#pss=pas
                return redirect('/page')



            #session["lpas"]=myresult('pass')

        #else:
            #return redirect('/index')
return render_template("login.html")

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


def register():     if
request.method =='POST':
details=request.form
        uname=details['username']

        email=details['email']
        inves=details['question1_field']
        # usd="wfwef"

        cpassw=details['cinves']
mycursor=mydb.cursor()
        sql="INSERT INTO registration(name,email,pas,cpass) VALUES
(%s,%s,%s,%s)"
        val=(uname,email,inves,cpassw)
mycursor.execute(sql,val)
mydb.commit()        mycursor.close()
#iden=mycursor.rowcount
```

```
        #con=sql.connection.cursor()
        #sql="insert into registration(name,email,password,confirm password) value
(%s,%s,%s,%s)"
        #con.execute(sql,[uname,email,passw,cpassw])
        #sql.connection.commit()
        #con.close
        return redirect('/login')
    return render_template("register.html")
@app.route('/rer', methods=['GET','POST']) def
rer():
    if request.method == 'POST':
details=request.form         uname=details['username']
email=details['email']         inves=details['inves']
cinves=details['cinves']
mycursor=mydb.cursor()
        sql="""INSERT INTO registration(name,email,pass,cpass) VALUES
(%s,%s,%s,%s)"""
        val=(uname,email,inves,cinves)
mycursor.execute(sql,val)
mydb.commit()         mycursor.close()
    return render_template('register2.html')

@app.route('/page') def bat():     return
render_template('page.html')

@app.route('/shirts') def shirts():
return render_template('shirts.html')
@app.route('/chudr') def chudr():
return render_template('chudr.html')
@app.route('/jk') def jk():      return
render_template('jk.html')



if __name__ == '__main__':

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

Index.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
>

  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4
HI+N" crossorigin="anonymous">
  <title>javascript</title>
  <link rel="stylesheet" href="{{url_for('static',filename='css/main.css')}}">
</head>

<body>
  <section class="bg-white position-sticky border-bottom border-light" style="top: 0;
z-index: 20;">
    <nav class="navbar navbar-expand-lg navbar-light">
      <a class="navbar-brand" href="#"><img src="{{user_image}}" alt=""></a>
      <ul class="navbar-nav ml-auto">
        <li class="nav-item ">
          <a class="nav-link active" style="color:#ffffff;" href="/index">HOME
<span class="sr-only">(current)</span></a>
        </li>
        <li class="nav-item">
          <a class="nav-link" style="color:#ffffff;" href="/register">REGISTER</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" style="color:#ffffff;" href="/login">LOGIN</a>
        </li>

      </ul>
      </div>
```

```
    </nav>
  </section>
  <section class="section-main">
    <div class="container">
      <div class="row mx-auto  ">


        <div class="col d-flex justify-content-center align-items-center">
          <div class="box">
            <div class="title">
              <span class="block"></span>
              <h1 style="font-weight: bold;color: #f87a04;font-size:
90px;">Smart<span></span></h1>
            </div>
          </div>
        </div>

      </div>

    </div>
    <div class="row mx-auto my-4 ">
      <div class="col d-flex justify-content-center align-items-center">
        <h2 style="font-weight: bold;color: #8d1600;font-size: 90px;">Fashion</h2>
</div>

    </div>
    <div class="row mx-auto my-4 ">
      <div class="col d-flex justify-content-center align-items-center">
<h3 style="font-weight: bold;color: #007ea7;font-size:
90px;">Recommender</h3>
      </div>
    </div>

    </div>
    <div class="row mx-auto ">
      <div class="col d-flex justify-content-center align-items-center">
<h4 style="font-weight: bold;color: #00916e;font-size:
90px;">Application</h4>
      </div>
```

```
  </div>
  </div>
 </section>
 <section class="section-footer">
  <div class="container-fluid">
   <div class="row">
    <div class="col">
     <p>Design and Developed By:<br>Sherin Dhanya & Team </p>
    </div>
   </div>
  </div>
 </section>

</body>

</html> Category
code:
<!DOCTYPE html>
<html>
<head> <style>
img{    height:
300px;;    width:
100%;
 } div.gallery {
margin: 5px;   border:
1px solid #ccc;   float:
left;   width: 180px;
}

div.gallery:hover {
border: 1px solid #777;
}

/* div.gallery img {
width:          100%;
height: auto;
} */
```

```
div.desc {
padding: 15px;
text-align: center;
}

.button {
  background-color: #f8ff98;
border: none;   color: black;
padding: 15px 32px;   text-
align: center;   text-
decoration: none;   display:
inline-block;   font-size:
16px;   margin: 4px 2px;
cursor: pointer;
}


</style>
</head>
<body>

<div class="gallery">
  <a target="_blank" href="img_5terre.jpg">
    <img src="/static/images/shirtimg/sh1.jpg" alt="Cinque Terre"  >
</a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>

<div class="gallery">
  <a target="_blank" href="img_forest.jpg">
    <img src="/static/images/shirtimg/sh2.jpg" alt="Forest" >
  </a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>


<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/shirtimg/sh3.jpeg" alt="Mountains" >
  </a>
```

```
  <div class="desc"><button class="button">Buy Now</button></div>
</div>
<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/shirtimg/sh4.jpg" alt="Mountains" >
  </a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>


<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/shirtimg/sh5.jpg" alt="Mountains" >
  </a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>
<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/shirtimg/sh6.jpg" alt="Mountains" >
  </a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>
<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/shirtimg/sh7.jpg" alt="Mountains" >
  </a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>

<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/shirtimg/sh8.jpg" alt="Mountains" >
  </a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>


<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
```

```
   <img src="/static/images/shirtimg/sh9.jpg" alt="Mountains" >
  </a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>
<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/shirtimg/sh10.jpg" alt="Mountains" >
  </a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>

<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/shirtimg/sh11.jpg" alt="Mountains" >
  </a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>

<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/shirtimg/sh12.jpg" alt="Mountains" >
  </a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>


</body>
</html> Product
code:
<!DOCTYPE html>
<html>
<head> <style>
img{     height:
300px;;
    width: 100%;
  } div.gallery {
margin: 5px;   border:
1px solid #ccc;   float:
left;   width: 180px;
}
```

```
div.gallery:hover {
border: 1px solid #777;
}

/* div.gallery img {
width:          100%;
height: auto;
} */

div.desc {
padding: 15px;
text-align: center;
}

.button {
  background-color: #f8ff98;
border: none;   color: black;
padding: 15px 32px;   text-
align: center;   text-
decoration: none;   display:
inline-block;   font-size:
16px;   margin: 4px 2px;
cursor: pointer;
}



</style>
</head>
<body>

<div class="gallery">
  <a target="_blank" href="img_5terre.jpg">
    <img src="/static/images/chudrimg/chudr1.jpg" alt="Cinque Terre"  >
</a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>
<div class="gallery">
  <a target="_blank" href="img_forest.jpg">
    <img src="/static/images/chudrimg/chudr2.jpg" alt="Forest" >
  </a>
```

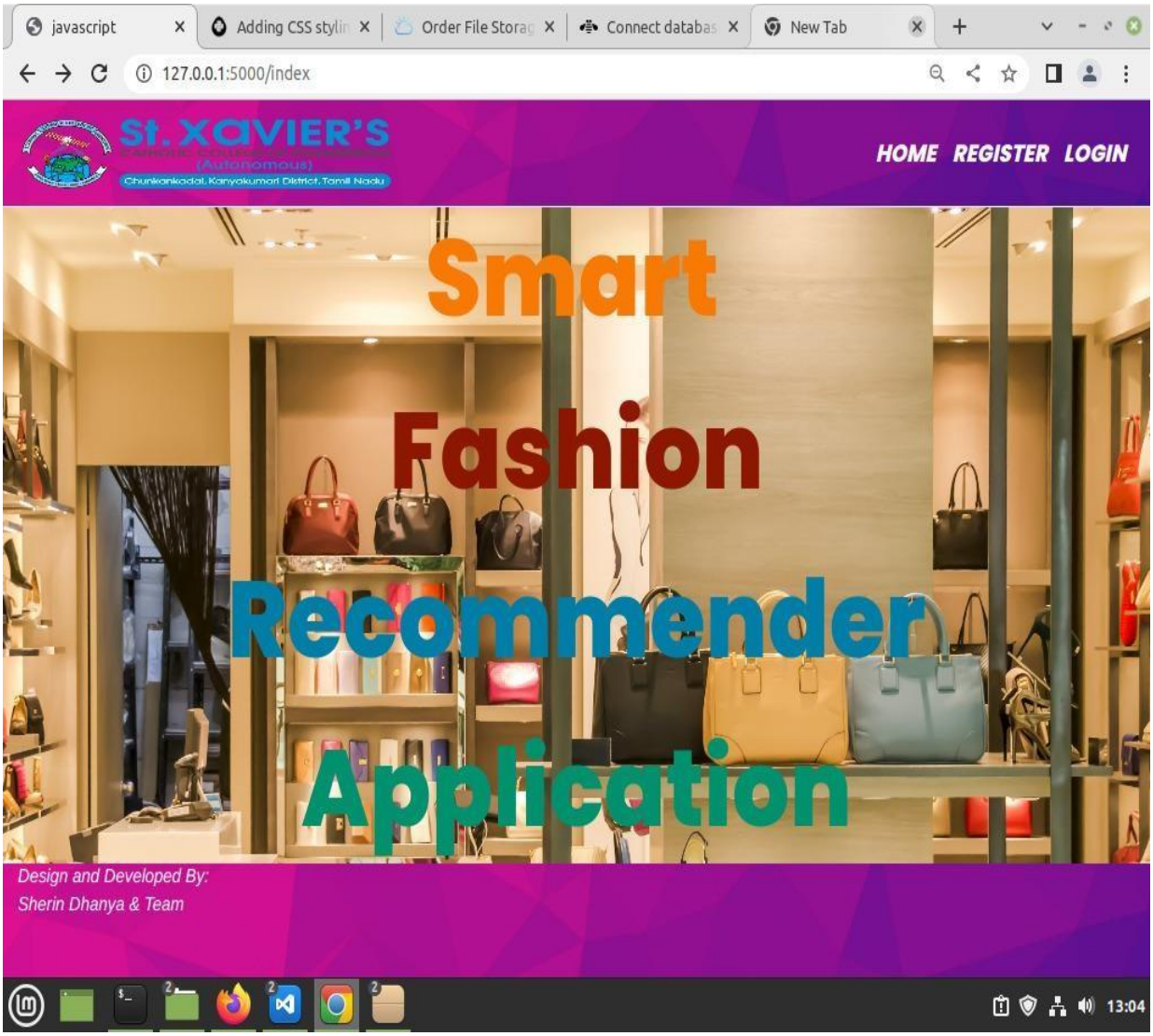
```
  <div class="desc"><button class="button">Buy Now</button></div>
</div>


<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/chudrimg/chudr3.jpg" alt="Mountains" >
</a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>
<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/chudrimg/chudr4.jpg" alt="Mountains" >
</a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>


<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/chudrimg/chudr5.jpg" alt="Mountains" >
</a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>
<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/chudrimg/chudr6.jpg" alt="Mountains" >
</a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>
<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/chudrimg/chudr7.jpg" alt="Mountains" >
</a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>

<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
```

```
   <img src="/static/images/chudrimg/chudr8.jpg" alt="Mountains" >
</a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>


<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/chudrimg/chudr9.jpg" alt="Mountains" >
</a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>

<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/chudrimg/chudr10.jpg" alt="Mountains" >
</a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>

<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/chudrimg/chudr11.jpg" alt="Mountains" >
</a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>

<div class="gallery">
  <a target="_blank" href="img_mountains.jpg">
    <img src="/static/images/chudrimg/chudr12.jpg" alt="Mountains" >
</a>
  <div class="desc"><button class="button">Buy Now</button></div>
</div>


</body>
</html>
```

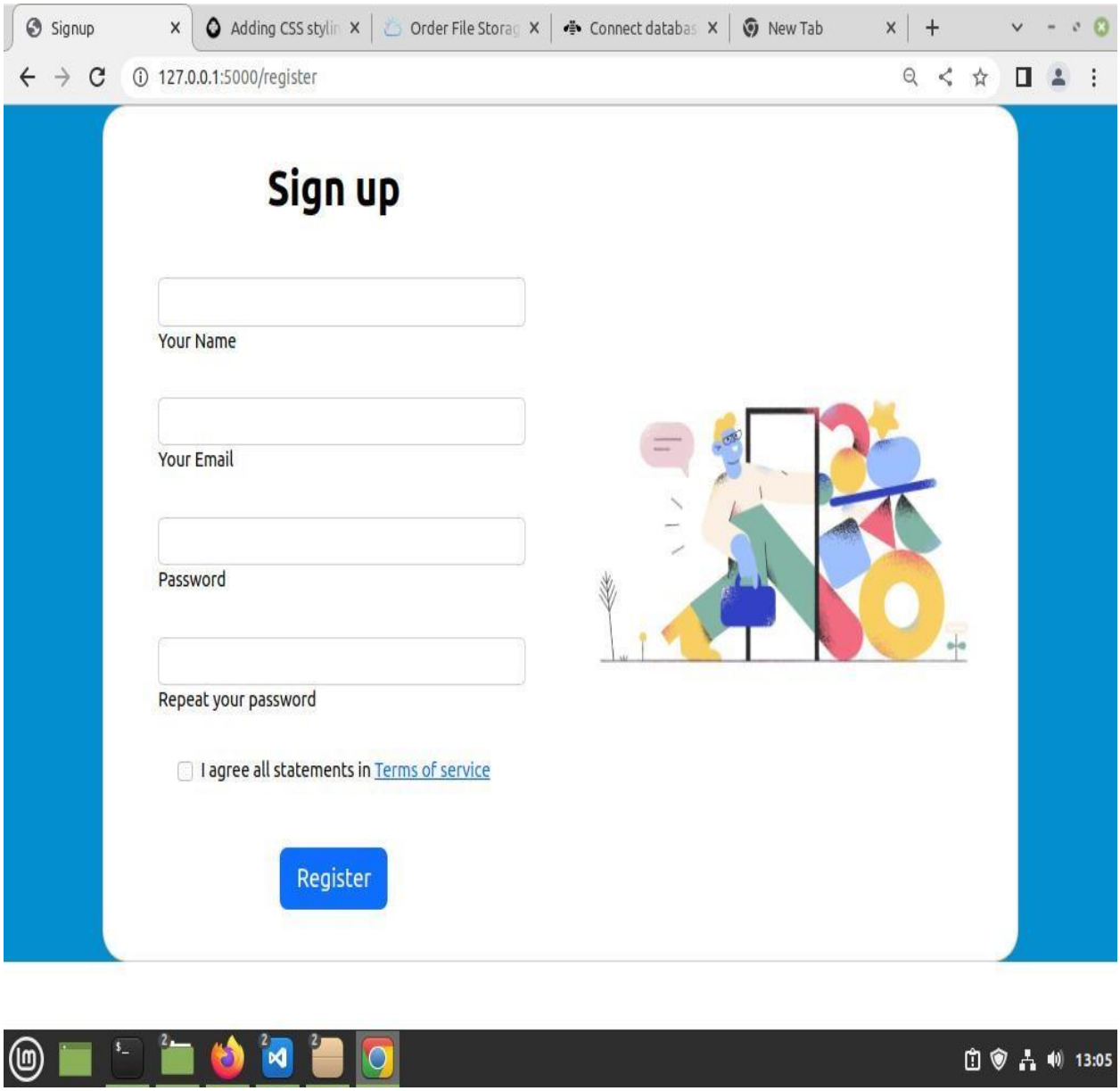
8. TESTING

TEST CASES:

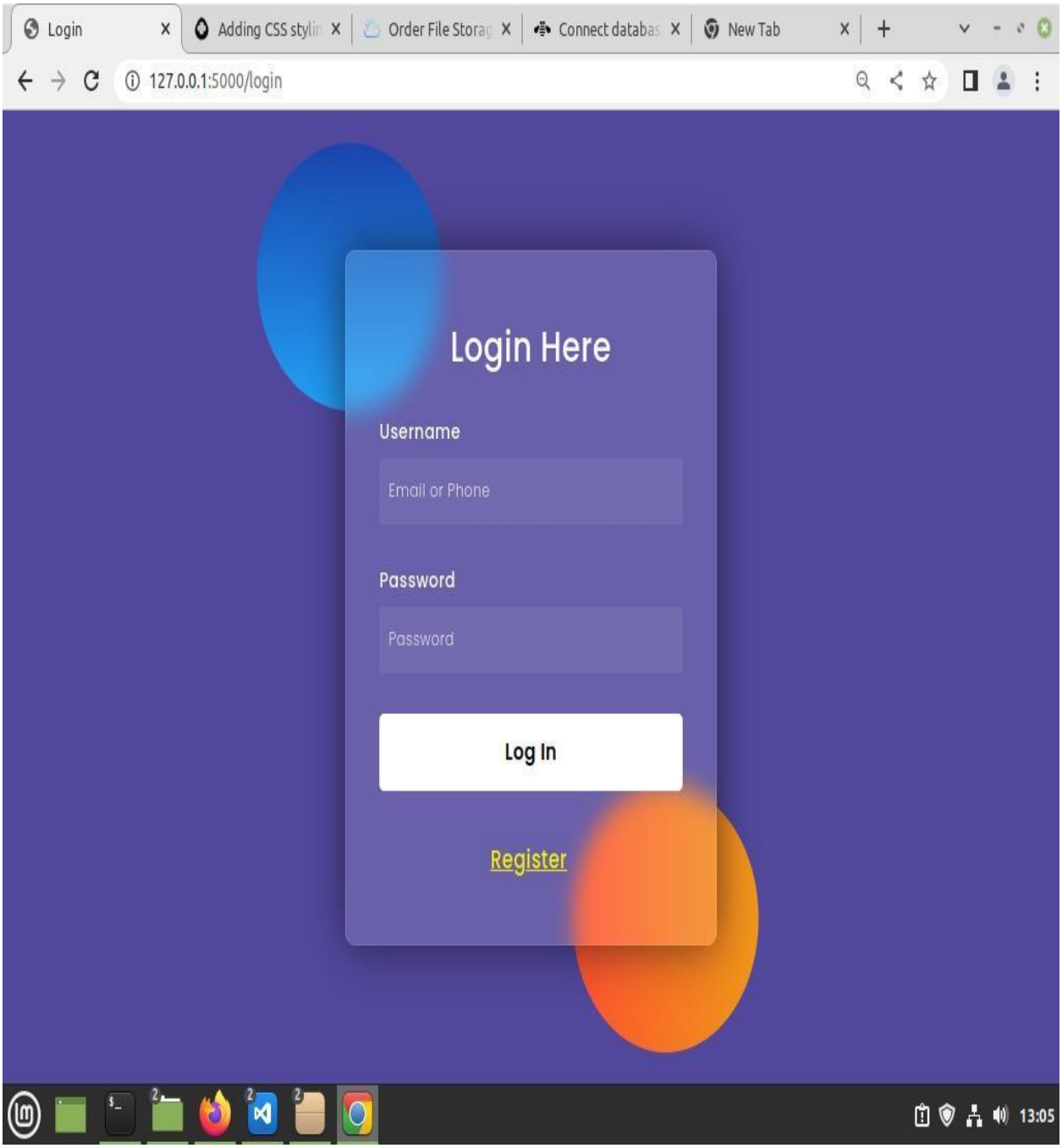
Home page:



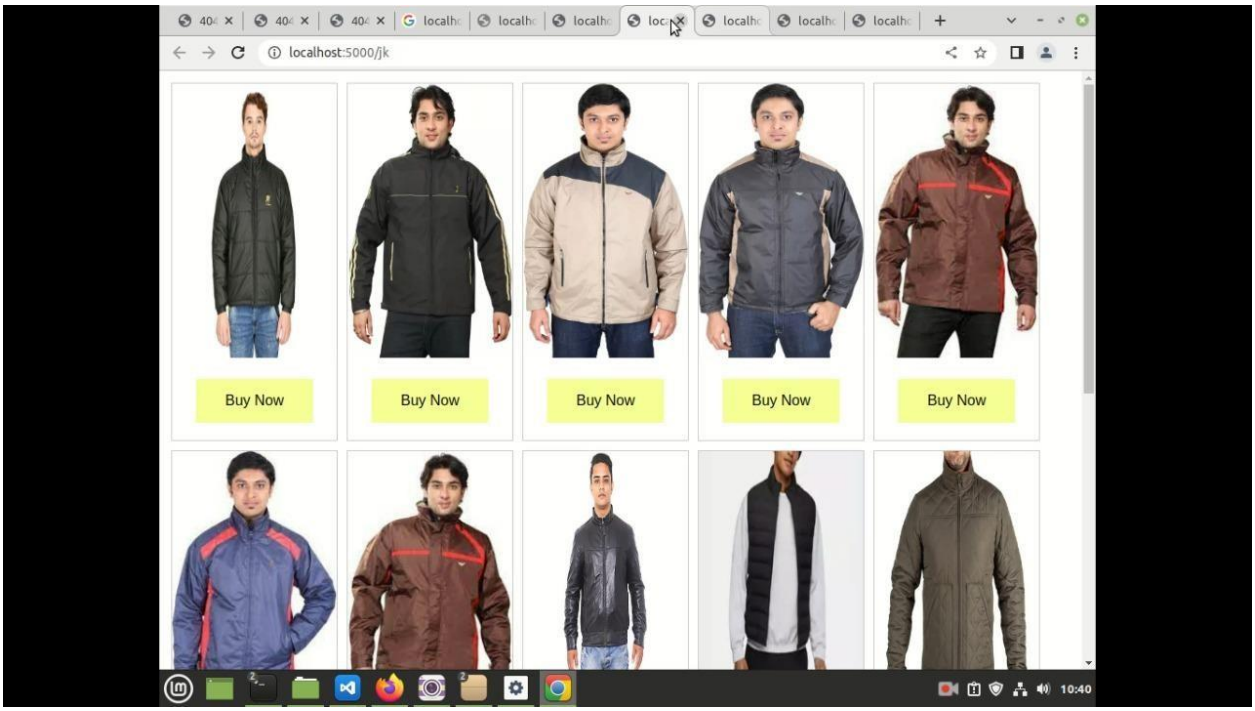
Sign up Page:

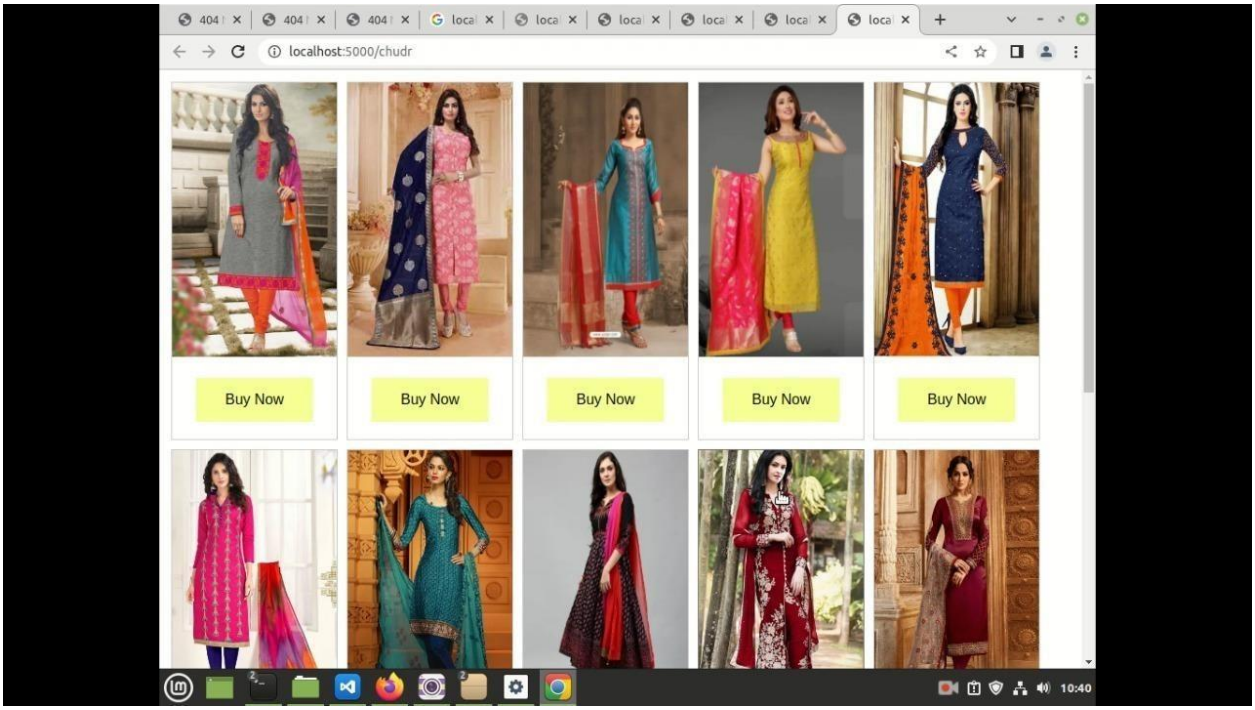

Signup
Adding CSS stylin
Order File Storag
Connect databas
New Tab
127.0.0.1:5000/register
Sign up
Your Name
Your Email
Password
Repeat your password
I agree all statements in Terms of service
Register
13:05

Login Page:



Product Page:





9.RESULTS

1. PERFORMANCE METRICS:

The performance of a recommendation algorithm is evaluated by using some specific metrics that indicate the accuracy of the system. The type of metric used depends on the 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 recommendation model. RMSE, as mentioned by [61], can be as represented as follows:

where, N_p is the total number of predictions, p_{ui} is the predicted rating that a user u will select an item i and r_{ui} 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 can be as represented as follows:

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

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

It is also defined as the ratio of the number of relevant recommended items to the total number 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 represents precision and recall as a single metric and can be as represented as follows:

Coverage. Coverage is used to measure the percentage of items which are recommended by the algorithm 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, which can be as represented as follows:

Intersection over union (IoU). It represents the accuracy of an object detector used on a specific dataset [62].

ROC. ROC curve is used to conduct a comprehensive assessment of the algorithm's performance [57].

AUC. AUC measures the performance of recommendation and its baselines as well as the quality of the ranking based on pairwise comparisons [5].

Rank aware top-N metrics. The rank aware top-N recommendation metric finds some of the interesting and unknown items that are presumed to be most attractive to a user [63]. Mean reciprocal rank (MRR), mean average precision (MAP) and normalized discounted cumulative gain (NDCG) are three most popular rank aware metrics.

MRR. MRR is calculated as a mean of the reciprocal of the position or rank of first relevant recommendation [64][65]. MRR as mentioned by [64][65] can be expressed as follows:

where u, N_u and R_u indicate specific user, total number of users and the set of items rated by the user, respectively. L indicates list of ranking length (n) for user (u) and k represents the position of the item found in the he lists L.

MAP: MAP is calculated by determining the mean of average precision at the points where relevant products or items are found. MAP as mentioned by [65] can be expressed as follows.

where P_u represents precision in selecting relevant item for the user. NDCG: NDCG is calculated by determining the graded relevance and positional information of the recommended items, which can be expressed as follows [65].

where $D(k)$ is a discounting function, $G(u, n, k)$ is the gain obtained recommending an item found at k-th position from the list L and $G^*(u, n, k)$ is the gain related to k-th item in the ideal ranking of n size for u user.

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1. Smart fashion recommender application is the user friendly.
2. With the help of chatbot user cand find the products very easily.
3. This application used to discover the product based on the user's choice , very easily and quickly.
4. It have ability to reduce transaction costs for consumers and increase revenue for retailers.

DISADVANTAGES:

1. It need active internet connection.
2. Privacy concerns.
3. Too many choices.

11. CONCLUSION

The Fashion Recommendation System is mainly used to recommend the best possible outfit combinations to a user who has no fashion sense based on their wardrobe . It may not always provide the best possible outfit to wear for an occasion as the system is dependent completely on the clothes present in the user's wardrobe. Also another reason is that fashion is highly dependent on the time period. However the system does a great job in inculcating a fashion sense

among the users and can provide the best recommendations based on the user's wardrobe. Since the system is implemented as a website, it is very easy for the end users to access as well as use. The scope of this system can be expanded by including the ability to detect the various design and patterns on clothing, and to increase the number of occasions.

Recommendation systems have the potential to explore new opportunities for retailers by enabling them to provide customized recommendations to consumers based on information retrieved from the Internet. They help consumers to instantly find the products and services that closely match with their choices.

12. FUTURE SCOPE

In the future, to implement this recommendation system to be extended to include male and nonbinary fashion items including apparel, footwear, accessories etc. This work can further be enhanced to predict fashion items based on the skin colour and weather conditions.

Future research should concentrate on including time series analysis and accurate categorization of product images based on the variation in colour, trend and clothing style in order to develop an effective recommendation system. The proposed model will follow brand specific personalization campaigns and hence it will ensure highly curated and tailored. offerings for users. Hence, this research will be highly beneficial for researchers interested in using augmented and virtual reality features to develop recommendation systems.

For different markets, it could split in short-term and long-term recommendations in the future research. Current discussions and reviews are all based on short-term recommendations toward apparel retailing markets. It delivers real-time recommendations straight to the online shoppers as shopping advice and suggestions. Apart from online shopping, recommendations could also be utilized in design and manufacture by providing long term recommendations, such as predicting new trends through years and seasons.

13. APPENDIX

SOURCE CODE:

Github Link :https://github.com/IBM-EPBL/IBM-Project-40953-1660637727