SMART FASHION RECOMMENDER APPLICATION

Team ID:PNT2022TMID49753

Team Members:

M.Prema Kalyani

S.Saravana Priya

C.Sowmiga

G.Vincyavathi

INDEX

1. INTRODUCTION

a. Project Overview

A 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.Using chatbot we can manage users' choices and orders. The chatbot can give recommendations to the users based on their interests. It can promote the best deals and offers on that day. It will store the customer's details and orders in the database. Chatbots can also help in collecting customer feedback and Application hosted in the python Flask.

b. Purpose

We aim to Increase sales and conversations and to personalize the customer experience. This project can help to build brand awareness and deal with customer queries. This enables accurate and quick product search. Personalization can be offered. Immediate response for customer queries is the major aim. Customers will be able to shop leisurely without any difficulties by using a recommender which is an chatbot built using IBM Watson Assistant so that just in few actions, the customer will be able to view their desirable products and place order by doing payments.Add their items in cart.

2. LITERATURE SURVEY

a. Existing problem

People find it difficult to navigate through pages citing various products using normal search method in a shopping website related to fashion. The usual search method takes some time to display all the available products and doesn't satisfy the desires of a customer. The user is unable to input their needs and wants as they think. It may not result in fulfilling the user search and requirements. The era of recommendation systems originally started in the 1990s based on the widespread research progress in Collective Intelligence. During this period, recommendations were generally provided to consumers based on their rating structure .

b. References

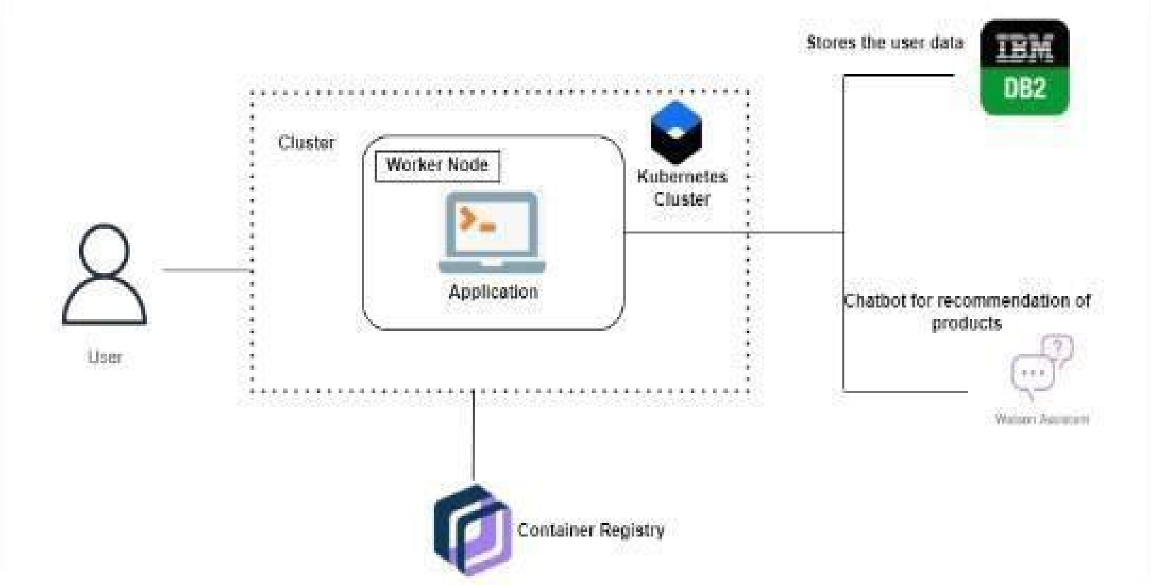
1. F. Ricci, L. Rokach and B. Shapira, "Introduction to recommender systems handbook, " in Recommender Systems Handbook, Eds. F. Ricci et al. Springer US, pp. 1-35, 2011.
2. Dietmar Sannach, Ahtsham Manzoor, Wanling Cai, Li Chen, "A Survey on Conversational Recommendational Systems", May 2021.
3. T. Sekozawa, "One to one recommendation system for apparel online shopping", WSEAS Transaction on Systems, vol.9, no1, pp. 94-103, 2010.
4. W.K. Wong, X.H. Zeng, W.M.R. Au and P.Y.Mok, S.Y.S. Leung, "A fashion mixand-match expert system for fashion retailer using fuzzy screening approach", Expert Systems with Applications, vol.36, pp. 17501764, 2009"2009.
5. A.R.D.B Landim, A.M.Pereira, T.Vieria, E.de B. Costa, J.A.B. Moura, V.Wanick,

   "Chatbot design approaches for fashion e-commerce".
6. Neera Sanjay Agashe, "Product Recommender Chatbot", ISSN: 22780181,June 2021.

c. Problem Statement

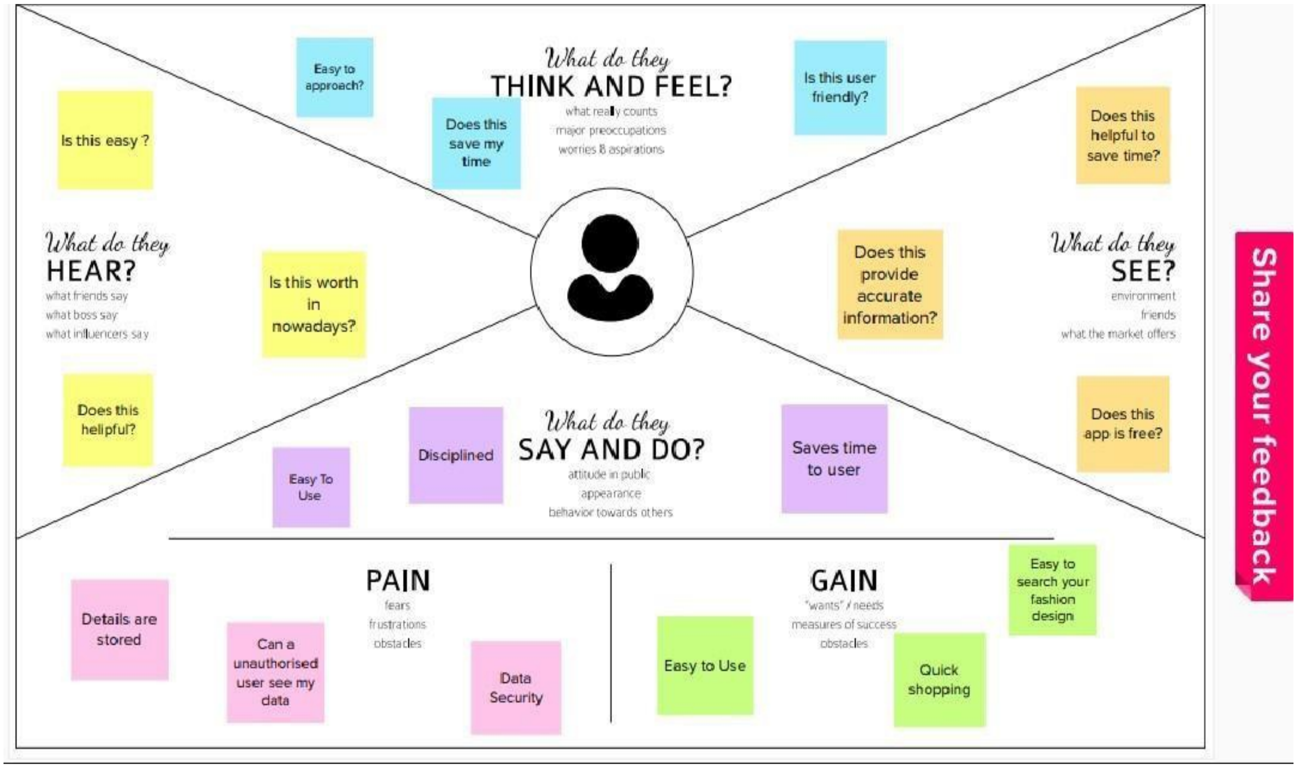
People find it difficult to navigate through pages citing various products using normal search method in a shopping website related to fashion. The usual search methods take some time to display all the available products and doesn't satisfy the desires of a customer .

The user is unable to input their needs and wants as they think. User faces several difficulties in user interface of the existing popular shopping websites. It is tough to match complicated user behaviour and to satisfy them.



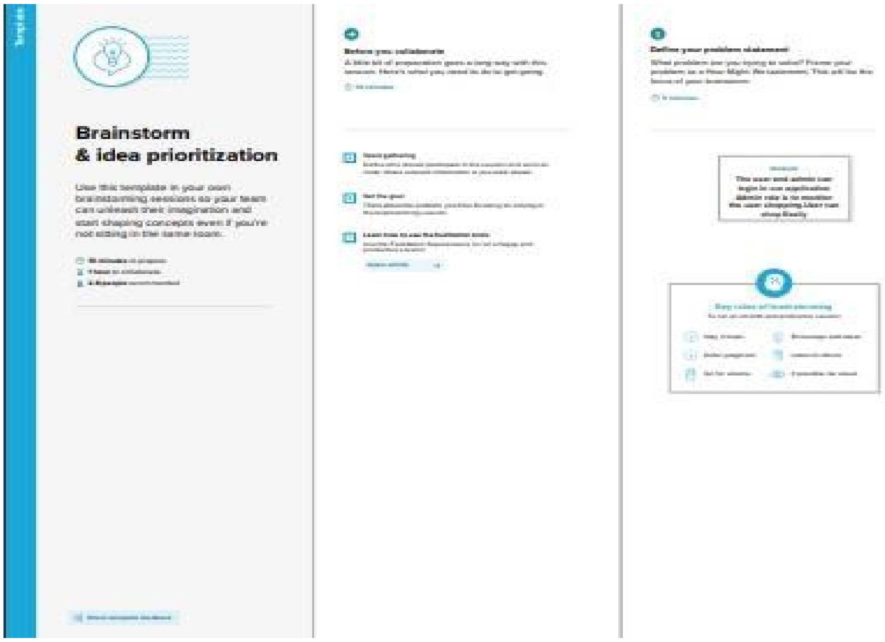
3. IDEATION & PROPOSED SOLUTION

a. Empathy Map Canvas



b. Ideation & Brainstorming

Step-1: Define Your Problem Statement &Team Gathering.



Brainstorm & idea prioritization

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

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

Team gathering

Set the goal

Learn how to use the facilitation tools

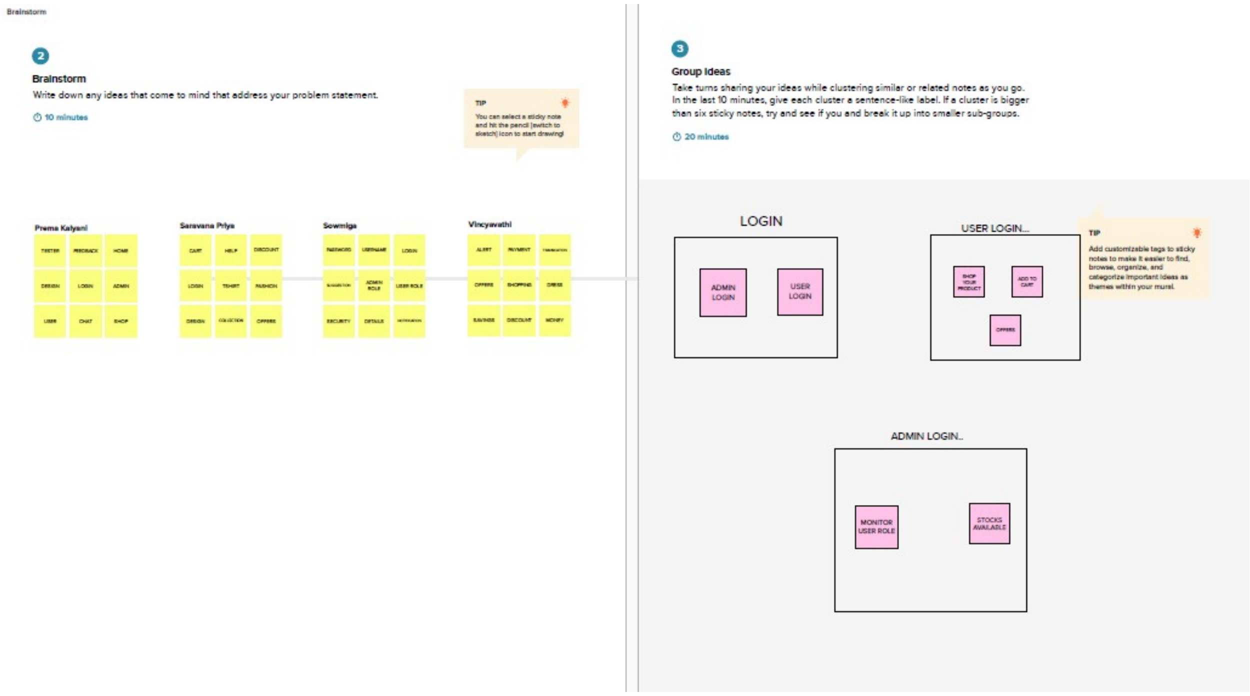
Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

The user and admin can login in our application admin role is to monitor the user shopping.User can shop Easily

Key rules of brainstorming

Step-2: Brainstorm, Idea Listing and Grouping



Brainstorm

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

3

Group Ideas

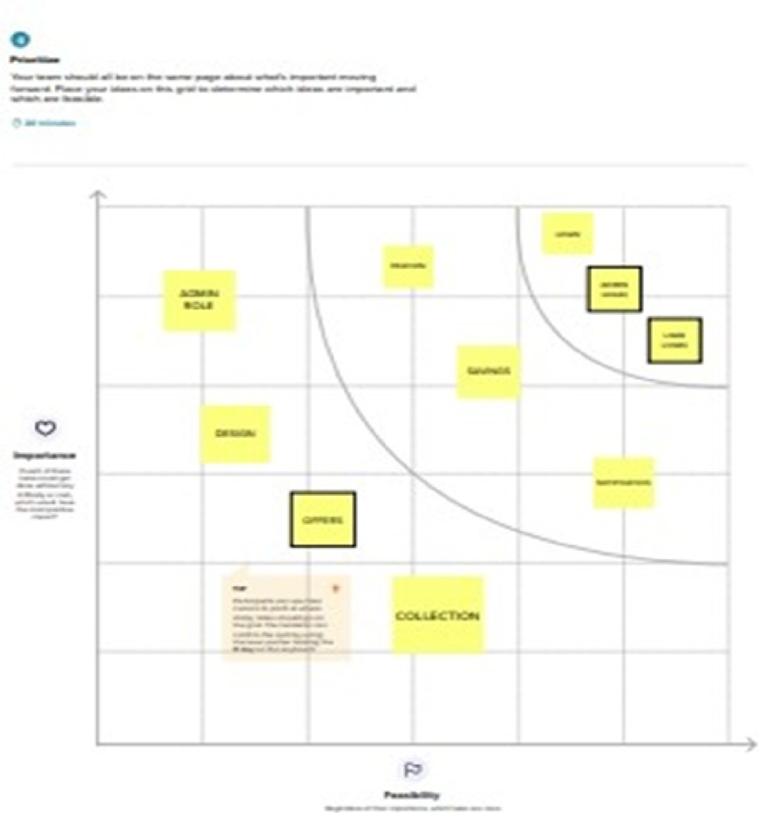
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Step-3: Prioritize

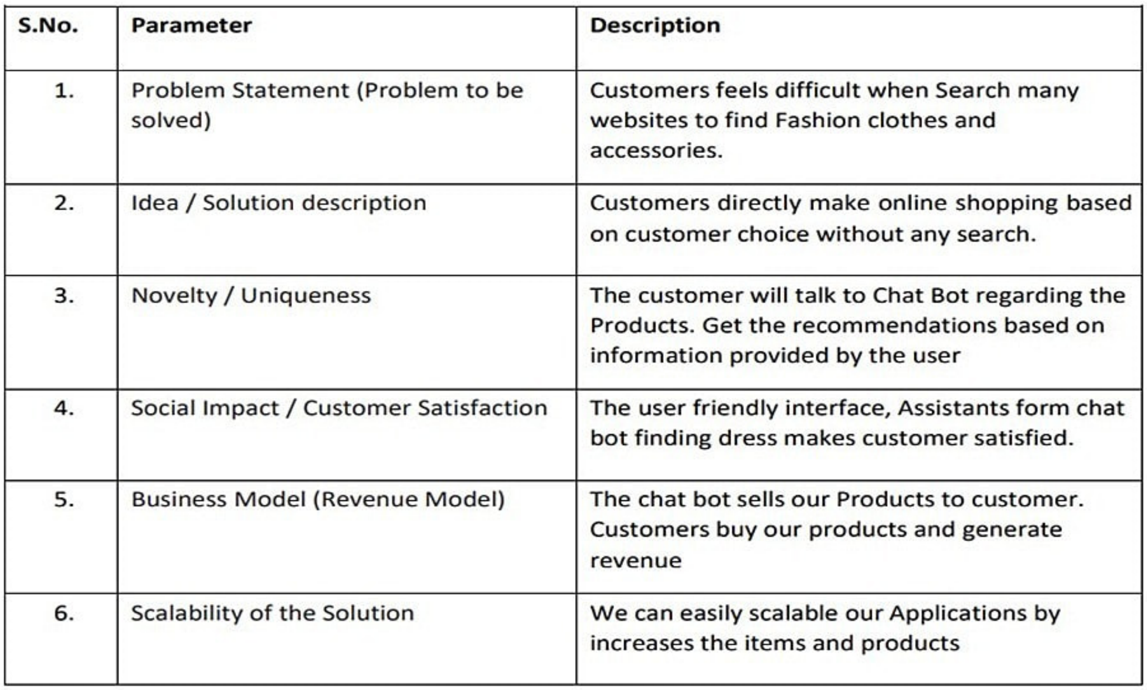


Prioritize

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

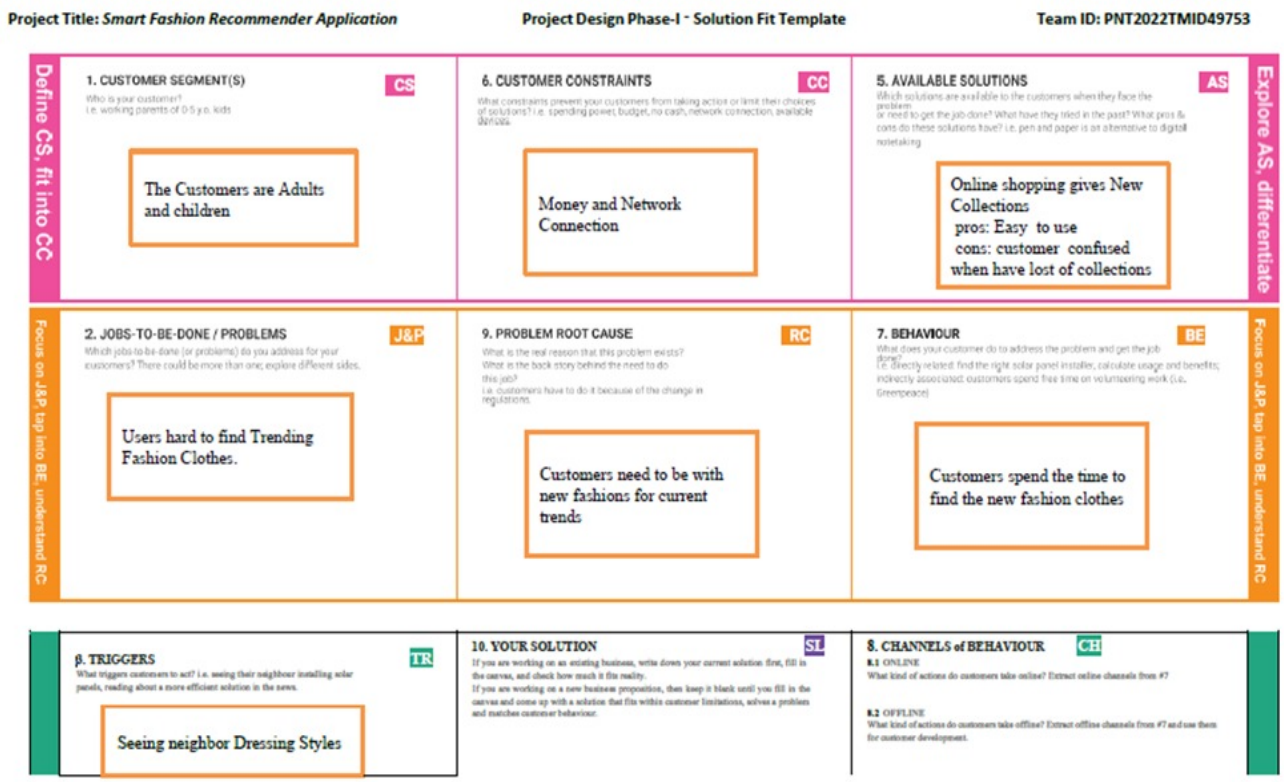
c. Proposed Solution



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

d. Problem Solution fit

Providing fashion recommendation using chatbot.You can directly do your online shopping based on your choice without any search. It can be done by using a chatbot. 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.



Project Title: *Smart Fashion Recommender Application*

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID49753

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

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

The Customers are Adults and children

6. CUSTOMER CONSTRAINTS CC

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

Money and Network Connection

5. AVAILABLE SOLUTIONS AS

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

Online shopping gives New Collections
pros: Easy to use
cons: customer confused when have lost of collections

Explore AS, differentiate

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

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

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

Users hard to find Trending Fashion Clothes.

9. PROBLEM ROOT CAUSE RC

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

Customers need to be with new fashions for current trends

7. BEHAVIOUR BE

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

Customers spend the time to find the new fashion clothes

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

β. TRIGGERS TR

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

Seeing neighbor Dressing Styles

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 behaviour.

8. CHANNELS of BEHAVIOUR CH

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

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

4. REQUIREMENT ANALYSIS

a. Functional requirements:

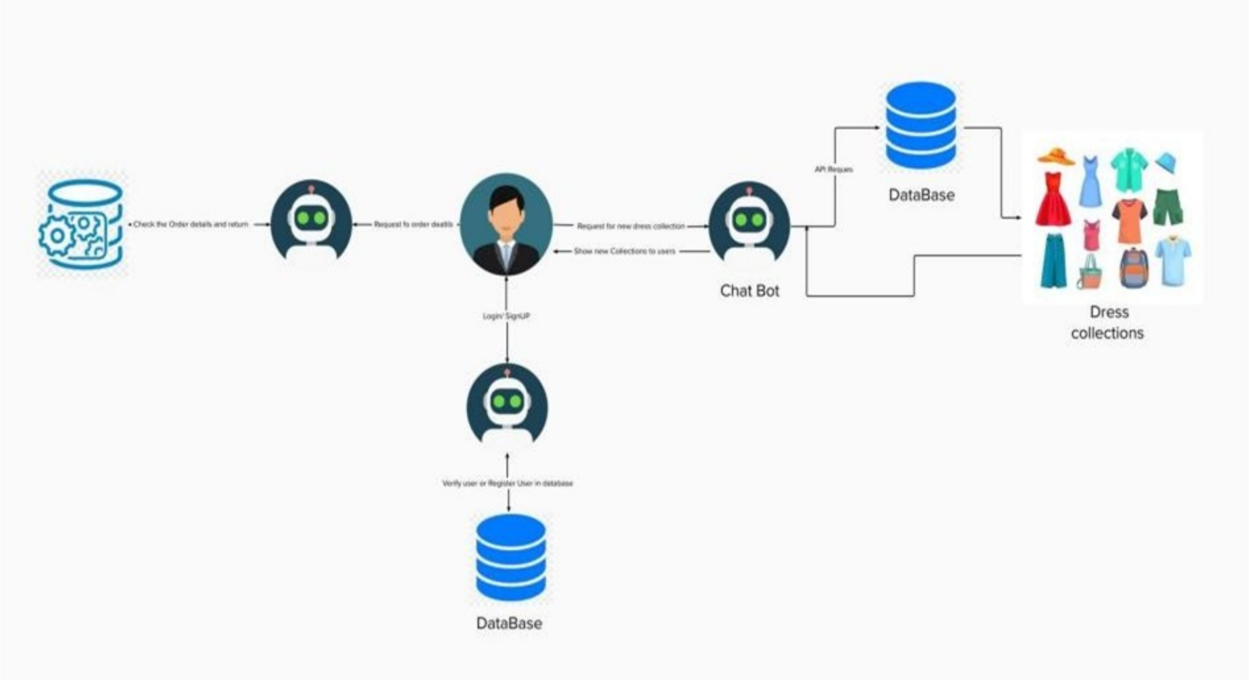
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
FR-2	User Interaction	Interact through the Chat Bot
FR-3	Buying Products	Through the chat Bot Recommendation
FR-4	Track Products	Ask the Chat Bot to Track my Orders
FR-5	Return Products	Through the chat Bot
FR_6	New Collections	Recommended from chat Bot

b. Non-functional Requirements:

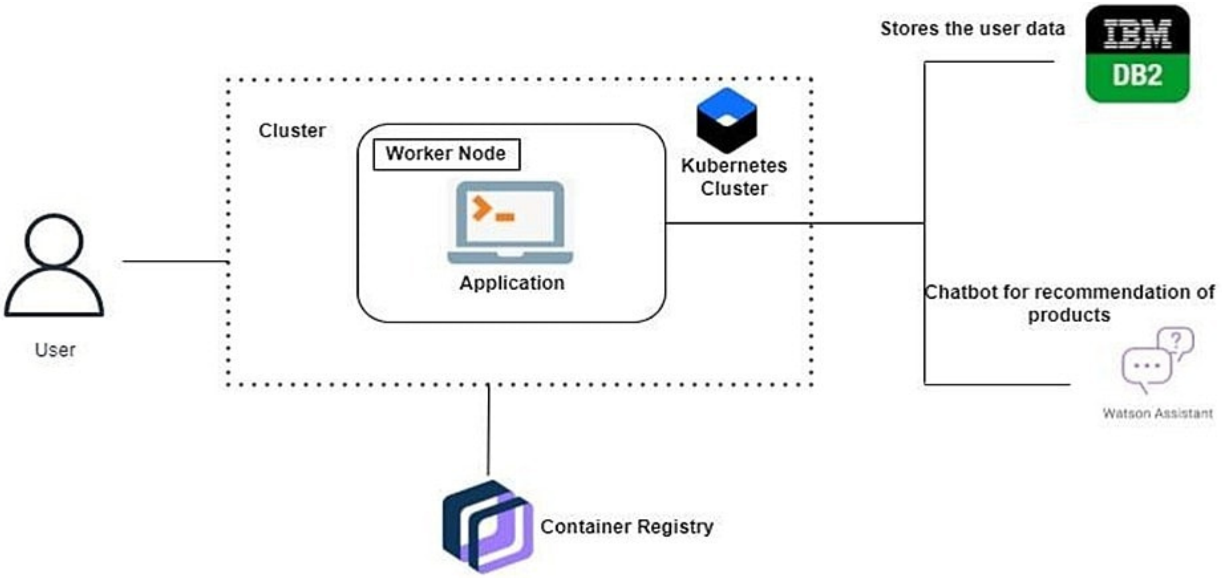
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Using Android or IOS or windows applications.
NFR-2	**Security**	The user data is stored securely in IBM cloud.
NFR-3	**Reliability**	The Quality of the services are trusted.
NFR-4	**Performance**	Its Provide smooth user experience.
NFR-5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

5. PROJECT DESIGN

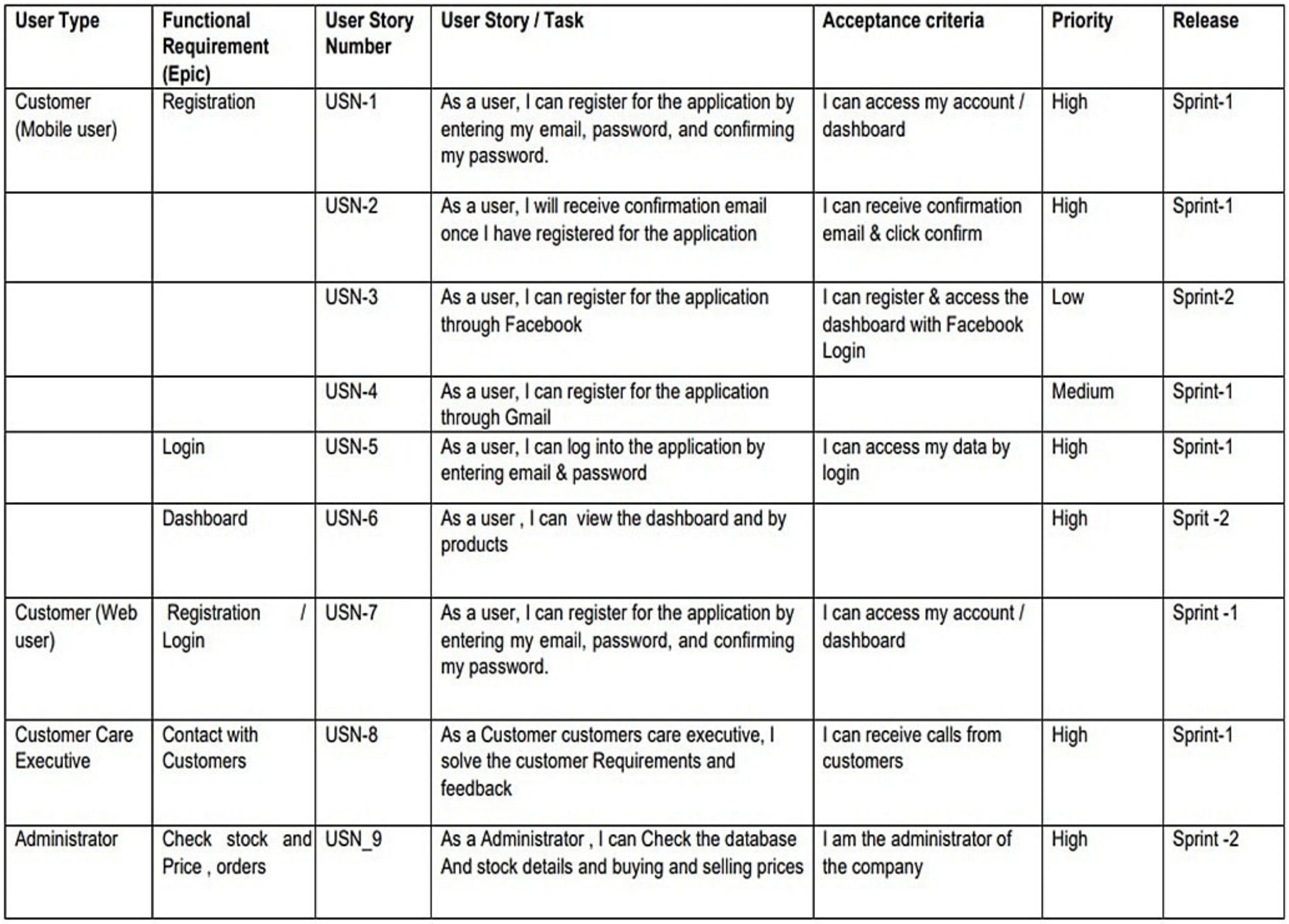
a. Data Flow Diagrams:


Check the Order details and return
Request to order deatils
Request for new dress collection
Show new Collections to users
Chat Bot
API Reques
DataBase
Dress collections
Login/ SignUP
Verify user or Register User in database
DataBase

b.Solution & Technical Architecture:



c. User Stories:



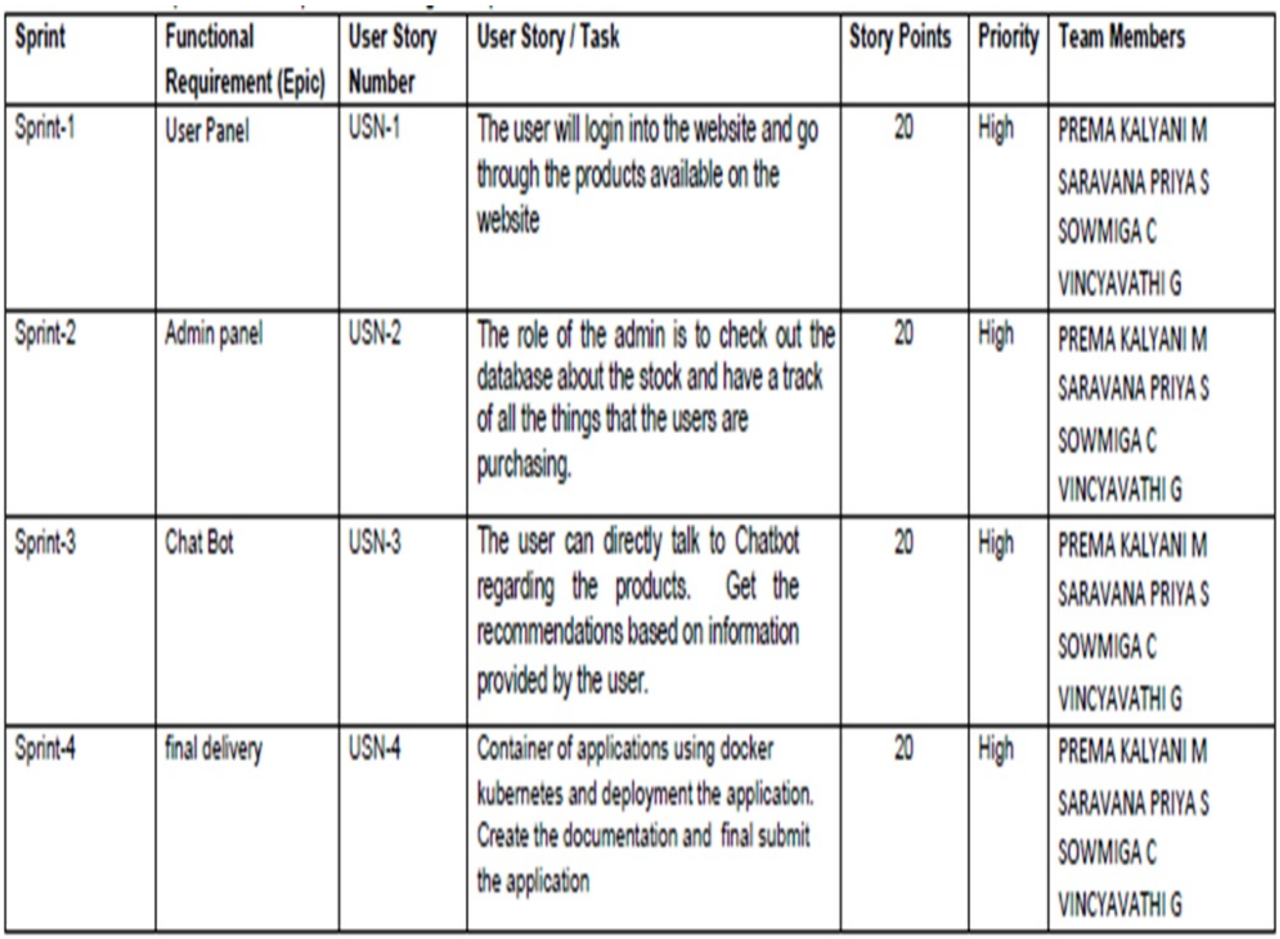
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High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

6. PROJECT PLANNING &SCHEDULING

a. Sprint Planning & Estimation:

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

b. **Sprint Delivery Schedule**



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Panel	USN-1	The user will login into the website and go through the products available on the website	20	High	PREMA KALYANI M SARAVANA PRIYA S SOWMIGA C VINCYAVATHI G
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	PREMA KALYANI M SARAVANA PRIYA S SOWMIGA C VINCYAVATHI G
Sprint-3	Chat Bot	USN-3	The user can directly talk to Chatbot regarding the products. Get the recommendations based on information provided by the user.	20	High	PREMA KALYANI M SARAVANA PRIYA S SOWMIGA C VINCYAVATHI G
Sprint-4	final delivery	USN-4	Container of applications using docker kubernetes and deployment the application. Create the documentation and final submit the application	20	High	PREMA KALYANI M SARAVANA PRIYA S SOWMIGA C VINCYAVATHI G

7. CODING & SOLUTIONING

a. Feature 1:

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">
    <title>Smart Fashion Recommanded Application</title>
```

```
  <link rel="stylesheet" href="style.css">
                                                                          <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <!----Navbar-->

  <nav class="navbar navbar-expand-lg navbar-dark" id="navbar">
    <div class="container-fluid">
            <a class="navbar-brand" href="#">Smart Fashion Recommender
Application</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-expanded="false">
      <span class="navbar-toggler-icon"></span>
     </button>
     <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav mx-auto">
       <li class="nav-item">
        <a class="nav-link" href="#home">Home</a>
       </li>
       <li class="nav-item">
        <a class="nav-link" href="#about">About</a>
       </li>
       <li class="nav-item">
        <a class="nav-link" href="#product">Products</a>
       </li>
       <li class="nav-item">
        <a class="nav-link" href="#contact">Contact</a>
       </li>
      </ul>
      <button class="btn p-2 my-lg-0 my-2">Sign In</button>
    </div>
   </nav>



   <!----Home-->

   <section id="home">
     <h1 class="text-center">Smart Fashion</h1>

     <div class="input-group m-4">
            <input  type="text"  class="form-control"  placeholder="UserName" required="">

      <a href="#product" class="btn signin">GET STARTED</a>

     </div>
   </section>



   <!-----About---->

   <section id="about">
     <div class="container-fluid">
       <div class="row">
         <div class="col-lg-6 col-md-6 col-12 my-2">
            <img src="about.png" class="img-fluid">
         </div>
         <div class="col-lg-6 col-md-6 col-12 p-lg-5 p-2 my-5">
           <h1>ABOUT US</h1>
            <p>Smart Fashion Recommender Application has grown throughout
```

the world nowadays. Generally, in physical stores, the sales representatives try to influence the buyers to buy the product. While in online shopping, you're free to do as you will.So there is no pressure in shopping.Customers do not have to stand in queues in cash counters to pay for the products that have been purchased by them. They can shop from their home or workplace and do not have to spend time travelling.This saves their time.Online consumers can track the order status and delivery status tracking of shipping is also available.To attract customers to shop online, e-retailers and marketers offer discounts to the customers as they have cut down on real estate and maintenance cost the sellers won't back out in giving huge discounts.This kind of shopping saves money also.

```
         </p>
         </div>
       </div>
     </div>
   </section>

   <section id="product">
     <div class="container m-5">
      <h1 class="text-center my-5">OUR PRODUCTS</h1>
      <div class="row">
        <div class="col-lg-4 col-md-4 col-12">
          <div class="card">
            <img src="21.jpg" class="card-img-top">
            <div class="card-body text-center">
             <h5 class="card-title">$500</h5>
             <a href="#PaymentProcess" class="btn signin">Buy Now</a>
            </div>
           </div>
        </div>
        <div class="col-lg-4 col-md-4 col-12">
```

```
    <div class="card">
      <img src="img14.jpg" class="card-img-top">
      <div class="card-body text-center">
       <h5 class="card-title">$750</h5>
       <a href="#PaymentProcess" class="btn signin">Buy Now</a>
      </div>
     </div>
  </div>
  <div class="col-lg-4 col-md-4 col-12">
    <div class="card">
      <img src="7.jpg" class="card-img-top">
      <div class="card-body text-center">
       <h5 class="card-title">$1000</h5>
       <a href="#PaymentProcess" class="btn signin">Buy Now</a>
      </div>
     </div>
  </div>
</div>
<div class="row">
  <div class="col-lg-4 col-md-4 col-12">
    <div class="card">
      <img src="9.jpg" class="card-img-top">
      <div class="card-body text-center">
       <h5 class="card-title">$700</h5>
       <a href="#PaymentProcess" class="btn signin">Buy Now</a>
      </div>
     </div>
  </div>
  <div class="col-lg-4 col-md-4 col-12">
    <div class="card">
      <img src="26.jpg" class="card-img-top">
```

```
            <div class="card-body text-center">
              <h5 class="card-title">$1500</h5>
              <a href="#PaymentProcess" class="btn signin">Buy Now</a>
            </div>
          </div>
    </div>
    <div class="col-lg-4 col-md-4 col-12">
        <div class="card">
            <img src="image13.jpg" class="card-img-top">
            <div class="card-body text-center">
              <h5 class="card-title">$400</h5>
              <a href="#PaymentProcess" class="btn signin">Buy Now</a>
            </div>
          </div>
    </div>
</div>
<div class="row">
    <div class="col-lg-4 col-md-4 col-12">
        <div class="card">
            <img src="5.jpg" class="card-img-top">
            <div class="card-body text-center">
              <h5 class="card-title">$800</h5>
              <a href="#PaymentProcess" class="btn signin">Buy Now</a>
            </div>
          </div>
    </div>
    <div class="col-lg-4 col-md-4 col-12">
        <div class="card">
            <img src="10.jpg" class="card-img-top">
            <div class="card-body text-center">
              <h5 class="card-title">$1200</h5>
          <a href="#PaymentProcess" class="btn signin">Buy Now</a>
         </div>
        </div>
     </div>
     <div class="col-lg-4 col-md-4 col-12">
       <div class="card">
         <img src="12.jpg" class="card-img-top">
         <div class="card-body text-center">
          <h5 class="card-title">550</h5>
          <a href="#PaymentProcess" class="btn signin">Buy Now</a>
         </div>


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

<section id="contact">
   <div class="container box my-5">
     <div class="row">
       <div class="col-lg-6 col-md-6 col-12">
         <img src="contact.jpg" class="img-fluid">

       </div>
       <div class="col-lg-6 col-md-6 col-12">
         <h1>CONTACT US</h1>
         <form class="mb-3">
            <input type="text" class="form-control" placeholder="Enter your name" required="">
```

```
                 <input type="email" class="form-control" placeholder="Enter
your mail" required="">
                     <textarea class="form-control" placeholder="Enter your
message"required=""></textarea>
              <button class="btn signin">Send Message</button>
            </form>
          </div>
        </div>
      </div>
   </section>

  <section id="PaymentProcess">
   <div class="row">
    <div class="col-75">
     <div class="container">
      <style>
       div {
        margin-bottom: 10px;
       }
       label {
        display: inline-block;
        width: 150px;
        text-align: right
       }
      </style>
      <form action="/action_page.php">

       <div class="row">
        <div class="col-50"></br><br/></br></br>
         <h3>Delivery Address</h3></br></br>
         <label for="fname"><i class="fa fa-user"></i> Name</label>
```

```
                                        <input type="text" id="fname" name="firstname"
placeholder=""required=""></br></br>
          <label for="email"><i class="fa fa-envelope"></i> Email</label>
                                          <input type="text" id="email" name="email"
placeholder=""required=""></br></br>
          <label for="adr"><i class="fa fa-address-card-o"></i> Address</label>
                                          <input type="text" id="adr" name="address"
placeholder=""required=""></div></br></br></br></br>
          <div class="row">
           <div class="col-50">
            <label for="state">MobileNo</label>
                                            <input type="text" id="state" name="state"
placeholder=""required=""></br></br>
            <label for="state">District</label>
                                            <input type="text" id="state" name="state"
placeholder=""required=""></br></br>
            <label for="state">State</label>
                                            <input type="text" id="state" name="state"
placeholder=""required=""></br></br></br></br>
            <a href="#confirm" class="btn signin">continue to process</a>
           </div>
          </div>
         </div>
        </form>
        <br/>
       </br>
      </br>
       </selection>
       <section id="confirm">
        <div class="row">
         <div class="col-75">
          <div class="container">
           <style>
            div {
             margin-bottom: 10px;
            }
            label {
             display: inline-block;
             width: 150px;
             text-align: right
            }
           </style>
         <div class="col-50">
          <h3>Payment Process</h3><br/></br>
          <div class="icon-container">
           <i class="fa fa-cc-visa" style="color:navy;"></i>
           <i class="fa fa-cc-amex" style="color:blue;"></i>
           <i class="fa fa-cc-mastercard" style="color:red;"></i>
           <i class="fa fa-cc-discover" style="color:orange;"></i>
          </div>
          <label for="cname">UPI</label>
              <input type="text" id="cname" name="cardname" placeholder=""
required=""></br></br>
          <label for="ccnum">WALLET /POSTPAID</label>
            <input type="text" id="ccnum" name="cardnumber" placeholder=""
required=""></br></br>
          <label for="expmonth">CREDIT/DEBIT/ATM CARD</label>
                     <input type="text" id="expmonth" name="expmonth"
placeholder=""required=""></br></br>

          <div class="row">
           <div class="col-50">
```

```
                     <label for="expyear">NET BANKING</label>
                                                <input  type="text"  id="expyear"  name="expyear"
placeholder=""required=""></br></br>
                   </div>
                   <div class="col-50">
                     <label for="cvv">CASH ON DELIVERY</label>
                                                            <input   type="text"   id="cvv"   name="cvv"
placeholder=""required=""></br></br>
                   </div>
                 </div>
               </div>

             </div>

             <a href="#success" class="btn signin">continue to process</a>

            </form>
          </div>
        </div>
      </div>
      <br/></br><br/></br><br/></br>
      <br/></br><br/></br><br/></br>
      <br/></br><br/></br><br/></br>
      <br/></br><br/></br><br/></br>
      <selection id="success">
<h1>Thankyou For Ordering!!!</h1>
      </selection>
  </section>
  <script>
    window.watsonAssistantChatOptions = {
         integrationID:  "67b0a2e3-5b12-4451-9f55-f7a9d920ac87",  //  The  ID  of  this
integration.
      region: "au-syd", // The region your integration is hosted in.
        serviceInstanceID: "68a44a85-8f97-48d4-b224-58ff667cbdc1", // 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>
```

style.css

```
@import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');

*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}
#navbar{
    position: sticky;
    top: 0;
    left: 0;
    z-index: 100;
    padding: .5rem 5rem;
    box-shadow: 5px 5px 20px rgba(0,0,0,.5);
    background: black;
}
.navbar .navbar-brand{
    font-size: 25px;
    font-weight: 800;
    color: #00f7eb !important;
}
#navbarSupportedContent a{
    color: #fff;
    border-bottom: 2px solid transparent;
}
#navbarSupportedContent a:hover{
    border-bottom: 2px solid #00e7f7;
}
#navbarSupportedContent button{
    background: #00eff7;
    width: 5rem;
    border-radius: 15px;
}
section{
    width: 100%;
    min-height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
}
```

```
/****Home*****/

#home{
   background: linear-gradient(rgba(0,0,0,.3),rgba(0,0,0,.4)),url(image.jpg);
   background-size: cover;
   background-position: center;
   flex-direction: column;
   margin-top: -50px;
}
#home h1{
   font-size: 100px;
   color: white;
   letter-spacing: 3px;
   font-family: Verdana, Geneva, Tahoma, sans-serif;
   text-shadow: 0px 1px 0px #CCC,
             0px 2px 0px #CCC,
             0px 3px 0px #CCC,
             0px 4px 0px #CCC,
             0px 5px 0px #CCC,
             0px 6px 0px #CCC,
             0px 7px 0px #CCC;
}
#home p{
   font-size: 18px;
   color: #fff;
}
#home .input-group{
   width: 40%;
   height: 45px;
}
```

```
.signin{
   background : rgb(0, 128, 107) !important;
   color: white !important;
   box-shadow: 2px 4px 5px rgba(0,0,0,.3);
}

/*****About******/

#about{
   background : linear-gradient(to right,#ea1d6f,#eb466b 100%)
}
#about h1{
   font-weight: 800;
   font-size: 50px;
   color:white;
}
#about p{
   color: white;
}

/*****Products****/

#product{
   background :#e5e5e5;
}
#product h1{
   font-size: 50px;
   letter-spacing: 2px;
   font-weight: 700;
}
#product img{
   width: 200px;
   height: 200px;
}
.card{
   width: 250px;
   height: 300px;
   background : #e5e5e5 !important;
   border :none !important;
   box-shadow: 15px 20px 20px rgba(0,0,0,.3),
               inset 4px 4px 10px white;
   border-radius: 20px !important;
   overflow: hidden;
   justify-content: center;
   align-items: center;
   margin: 20px 60px;
   transition: .2s;
}
.card:hover{
   box-shadow: inset 5px 5px 10px rgba(0,0,0,.3),
               inset -4px -4px 10px white;
   transition: .2s;
}

/****Contact****/
#contact img{
   height: 100%;
}
.box{
   width: 80% !important;
   margin-top: 10px;
}
```

```
form{
    display: flex;
    flex-direction: column;
}
#contact input{
    margin: 10px 0px;
}
#contact textarea{
    margin: 10px 0px;
}
footer{
    width: 100%;
    height: 30px;
    background: rgba(0,0,0,.8);
    margin-top: -30px;
    text-align: center;
    color: white;
    padding: 3px;
}
```

connection.php

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password);

// Check connection
```

```
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

cart.php

```
<?php

// php cart class
class Cart
{
    public $db = null;

    public function __construct(DBController $db)
    {
        if (!isset($db->con)) return null;
        $this->db = $db;
    }

    // insert into cart table
    public  function insertIntoCart($params = null, $table = "cart"){
        if ($this->db->con != null){
            if ($params != null){
                // "Insert into cart(user_id) values (0)"
                // get table columns
                $columns = implode(',', array_keys($params));

                $values = implode(',' , array_values($params));
                // create sql query
                  $query_string = sprintf("INSERT INTO %s(%s) VALUES(%s)", $table,
$columns, $values);

                // execute query
                $result = $this->db->con->query($query_string);
                return $result;
            }
        }
    }

    // to get user_id and item_id and insert into cart table
    public  function addToCart($userid, $itemid){
        if (isset($userid) && isset($itemid)){
            $params = array(
                "user_id" => $userid,
                "item_id" => $itemid
            );

            // insert data into cart
            $result = $this->insertIntoCart($params);
            if ($result){
                // Reload Page
                header("Location: " . $_SERVER['PHP_SELF']);
            }
        }
    }

    // delete cart item using cart item id
    public function deleteCart($item_id = null, $table = 'cart'){
        if($item_id != null){
                $result = $this->db->con->query("DELETE FROM {$table} WHERE
item_id={$item_id}");
            if($result){
                header("Location:" . $_SERVER['PHP_SELF']);
            }
            return $result;
        }
    }

    // calculate sub total
    public function getSum($arr){
        if(isset($arr)){
            $sum = 0;
            foreach ($arr as $item){
                $sum += floatval($item[0]);
            }
            return sprintf('%.2f' , $sum);
        }
    }

    // get item_it of shopping cart list
    public function getCartId($cartArray = null, $key = "item_id"){
        if ($cartArray != null){
            $cart_id = array_map(function ($value) use($key){
                return $value[$key];
            }, $cartArray);
            return $cart_id;
        }
    }

    // Save for later
```

```
        public function saveForLater($item_id = null, $saveTable = "wishlist",
$fromTable = "cart"){
        if ($item_id != null){
              $query = "INSERT INTO {$saveTable} SELECT * FROM {$fromTable}
WHERE item_id={$item_id};";
            $query .= "DELETE FROM {$fromTable} WHERE item_id={$item_id};";

            // execute multiple query
            $result = $this->db->con->multi_query($query);

            if($result){
                header("Location :" . $_SERVER['PHP_SELF']);
            }
            return $result;
        }
    }
}
```

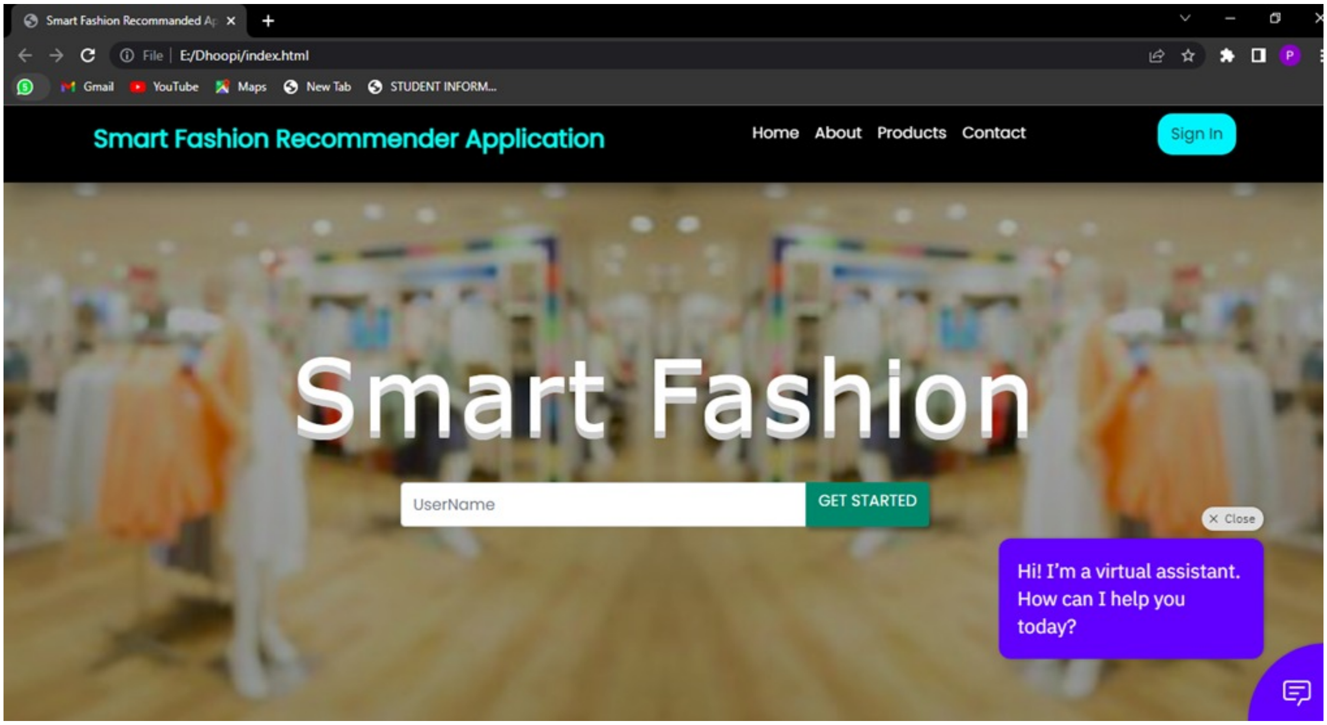
DB.php

```
import ibm_db

public  function addToCart($usernme){
        if (isset($username) && isset($itemid)){
            $params = array(
                "username" => $username


            );
CREATE TABLE `product` (
  `item_id` int(11) NOT NULL,
  `item_brand` varchar(200) NOT NULL,
  `item_name` varchar(255) NOT NULL,
  `item_price` double(10,2) NOT NULL,
  `item_image` varchar(255) NOT NULL,
  `item_register` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `username` (
  `username` int(11) NOT NULL,
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;


--
-- Dumping data for table `username`
--

INSERT INTO `user` (`username`) VALUES

ALTER TABLE `username`

  ADD PRIMARY KEY (`username`);
  ALTER TABLE `username`
        MODIFY `username` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=3;
COMMIT;
```

product.php

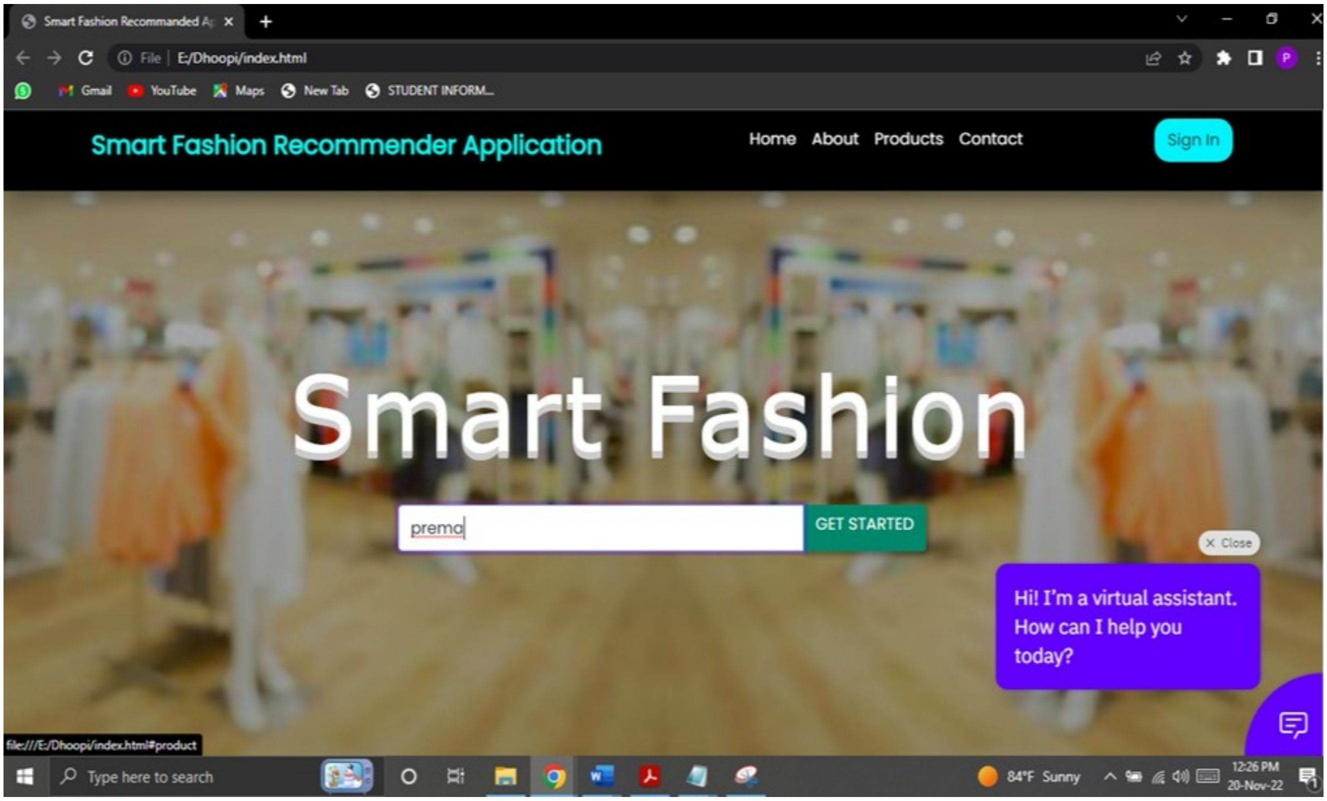
```
<?php

// Use to fetch product data
class Product
{
    public $db = null;

    public function __construct(DBController $db)
    {
        if (!isset($db->con)) return null;
        $this->db = $db;
    }

    // fetch product data using getData Method
    public function getData($table = 'product'){
        $result = $this->db->con->query("SELECT * FROM {$table}");

        $resultArray = array();

        // fetch product data one by one
        while ($item = mysqli_fetch_array($result, MYSQLI_ASSOC)){
            $resultArray[] = $item;
        }

        return $resultArray;
    }

    // get product using item id
    public function getProduct($item_id = null, $table= 'product'){
        if (isset($item_id)){
                $result = $this->db->con->query("SELECT * FROM {$table} WHERE
item_id={$item_id}");

            $resultArray = array();

            // fetch product data one by one
            while ($item = mysqli_fetch_array($result, MYSQLI_ASSOC)){
                $resultArray[] = $item;
            }
            return $resultArray;
        }
    }
}
```

Home Page:



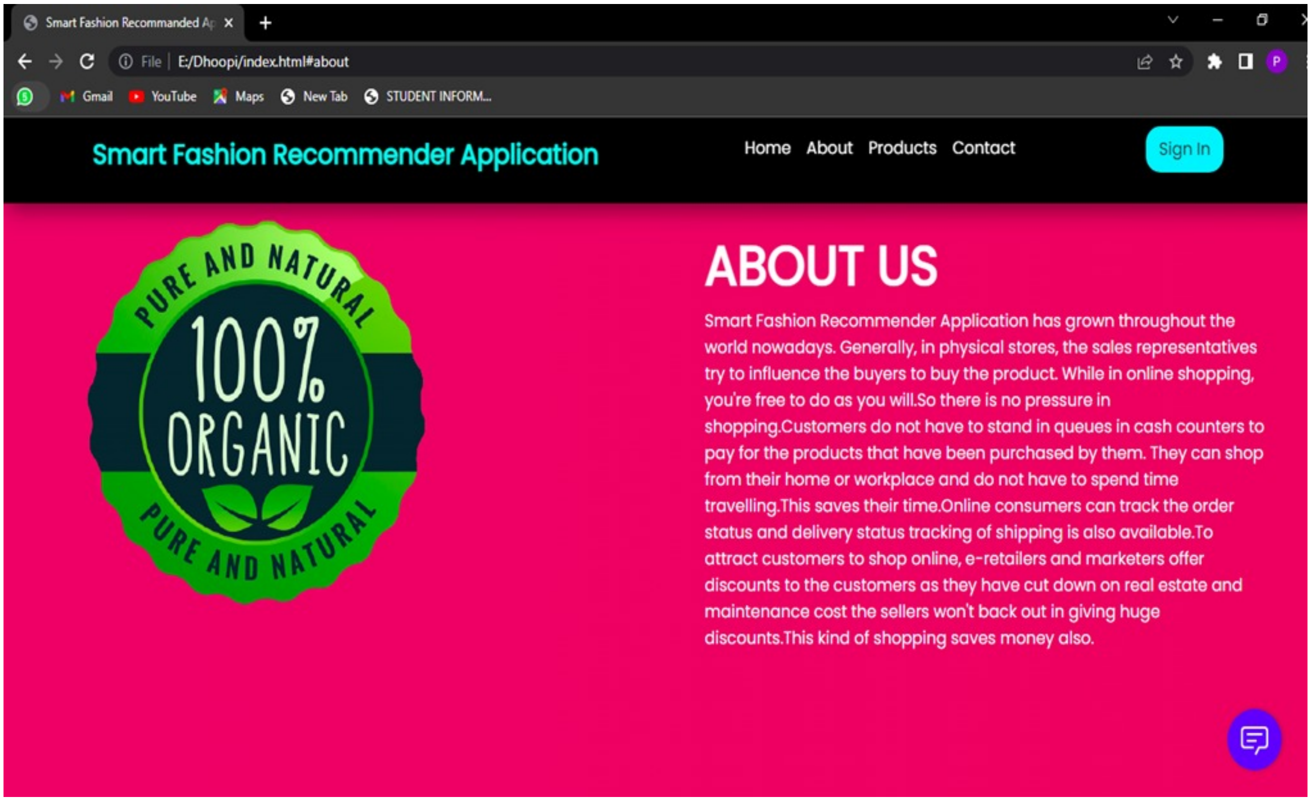
Hi! I'm a virtual assistant. How can I help you today?

Enter the Username and click Get Started



Hi! I'm a virtual assistant. How can I help you today?

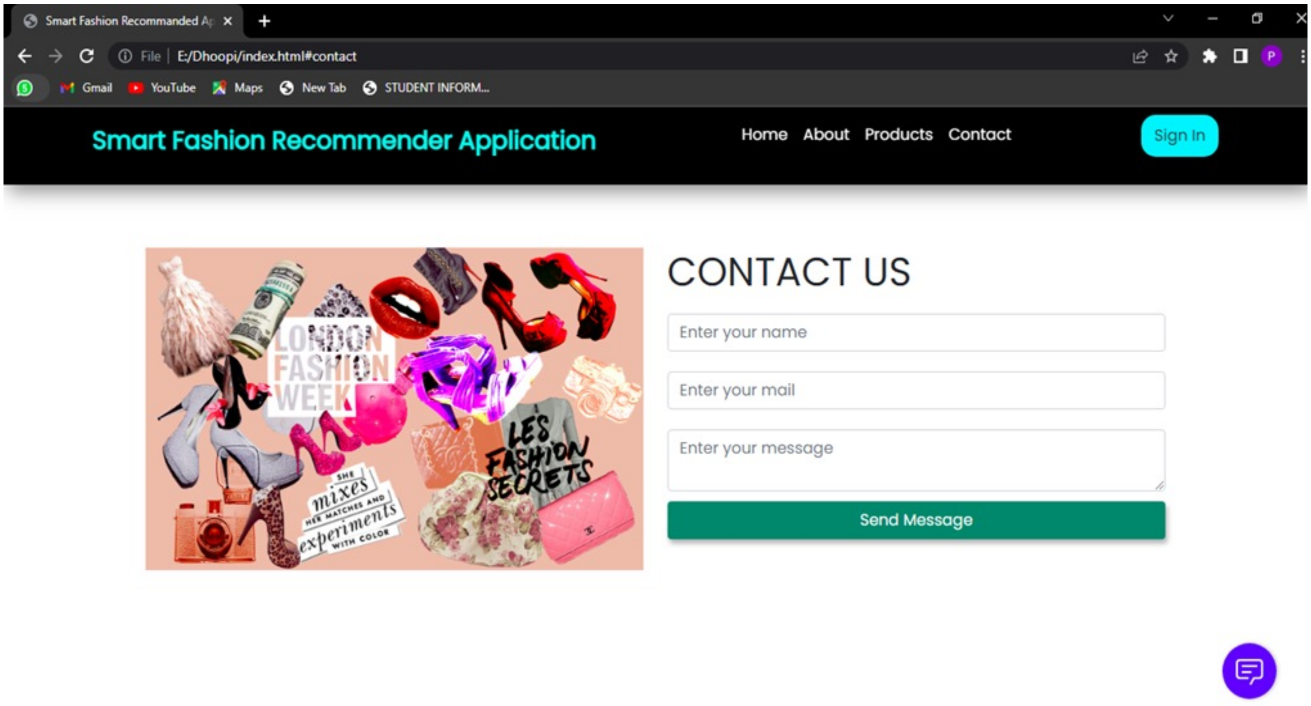
About us Page:



ABOUT US

Smart Fashion Recommender Application has grown throughout the world nowadays. Generally, in physical stores, the sales representatives try to influence the buyers to buy the product. While in online shopping, you're free to do as you will.So there is no pressure in shopping.Customers do not have to stand in queues in cash counters to pay for the products that have been purchased by them. They can shop from their home or workplace and do not have to spend time travelling.This saves their time.Online consumers can track the order status and delivery status tracking of shipping is also available.To attract customers to shop online, e-retailers and marketers offer discounts to the customers as they have cut down on real estate and maintenance cost the sellers won't back out in giving huge discounts.This kind of shopping saves money also.

Contact us Page:


Smart Fashion Recommanded Ap
File | E:/Dhoopi/index.html#contact
Gmail
YouTube
Maps
New Tab
STUDENT INFORM...
Smart Fashion Recommender Application
Home
About
Products
Contact
Sign In
LONDON FASHION WEEK
LES FASHION SECRETS
SHE mixes HER MATCHES AND experiments WITH COLOR

CONTACT US

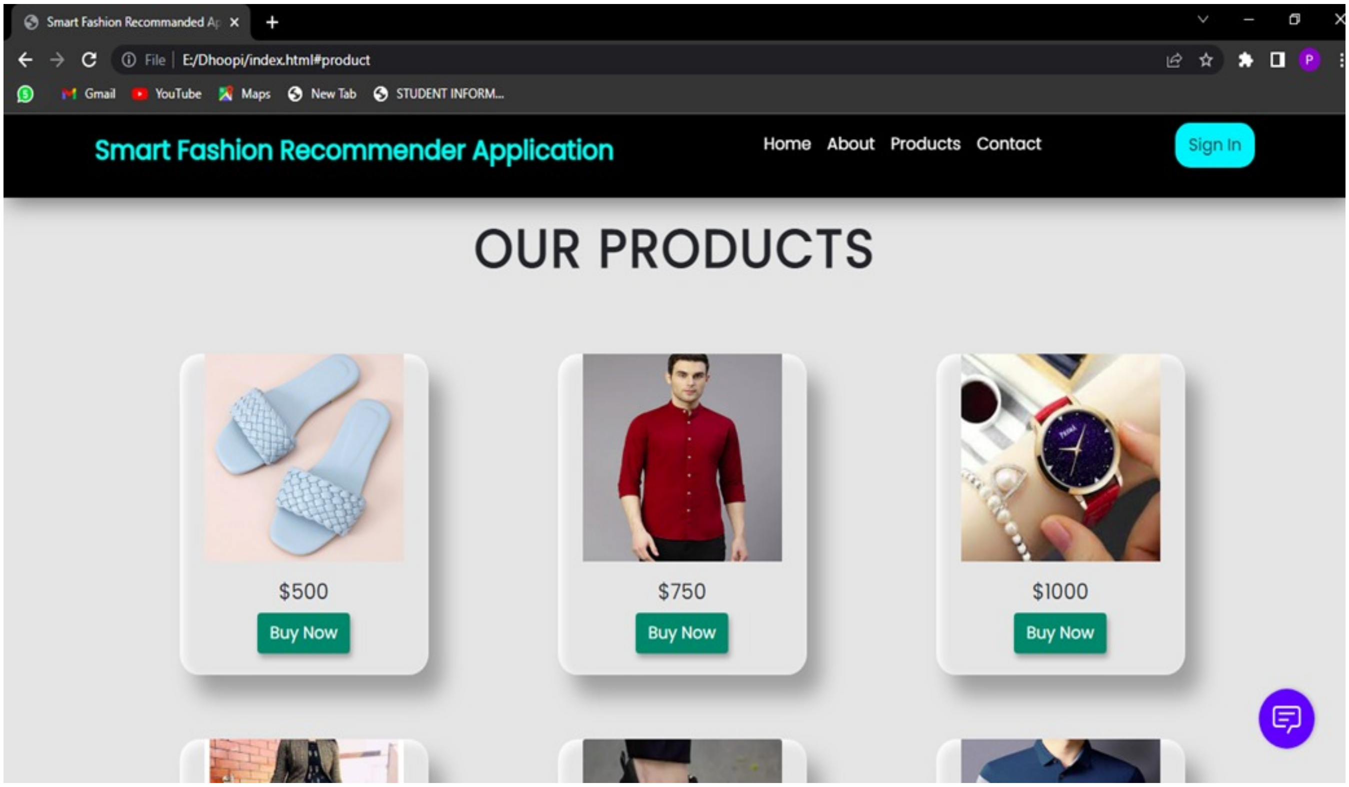
Enter your name

Enter your mail

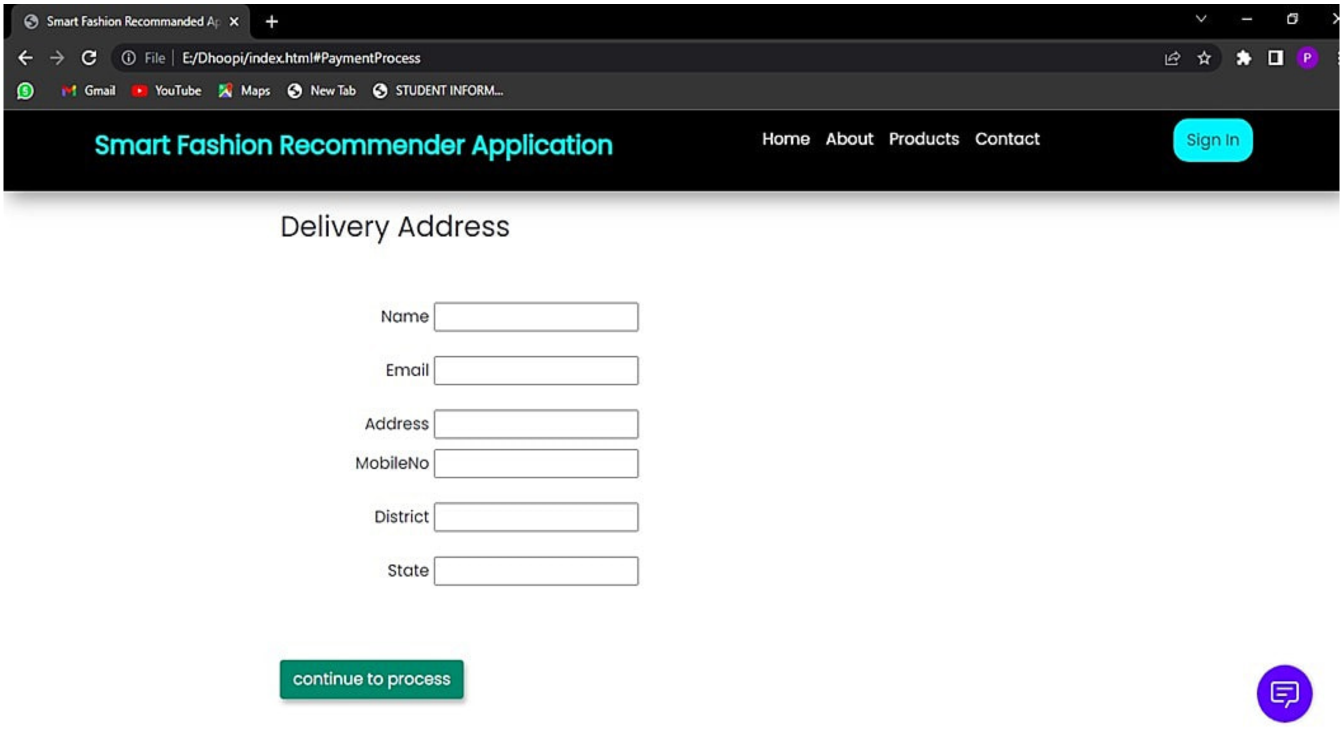
Enter your message

Send Message

Our Products Page:



After clicking “Buy Now” it goes to Delivery Address and Payment Page


Smart Fashion Recommanded Ap
File | E:/Dhoopi/index.html#PaymentProcess
Gmail
YouTube
Maps
New Tab
STUDENT INFORM...
Smart Fashion Recommender Application
Home
About
Products
Contact
Sign In
continue to process

Delivery Address

Name

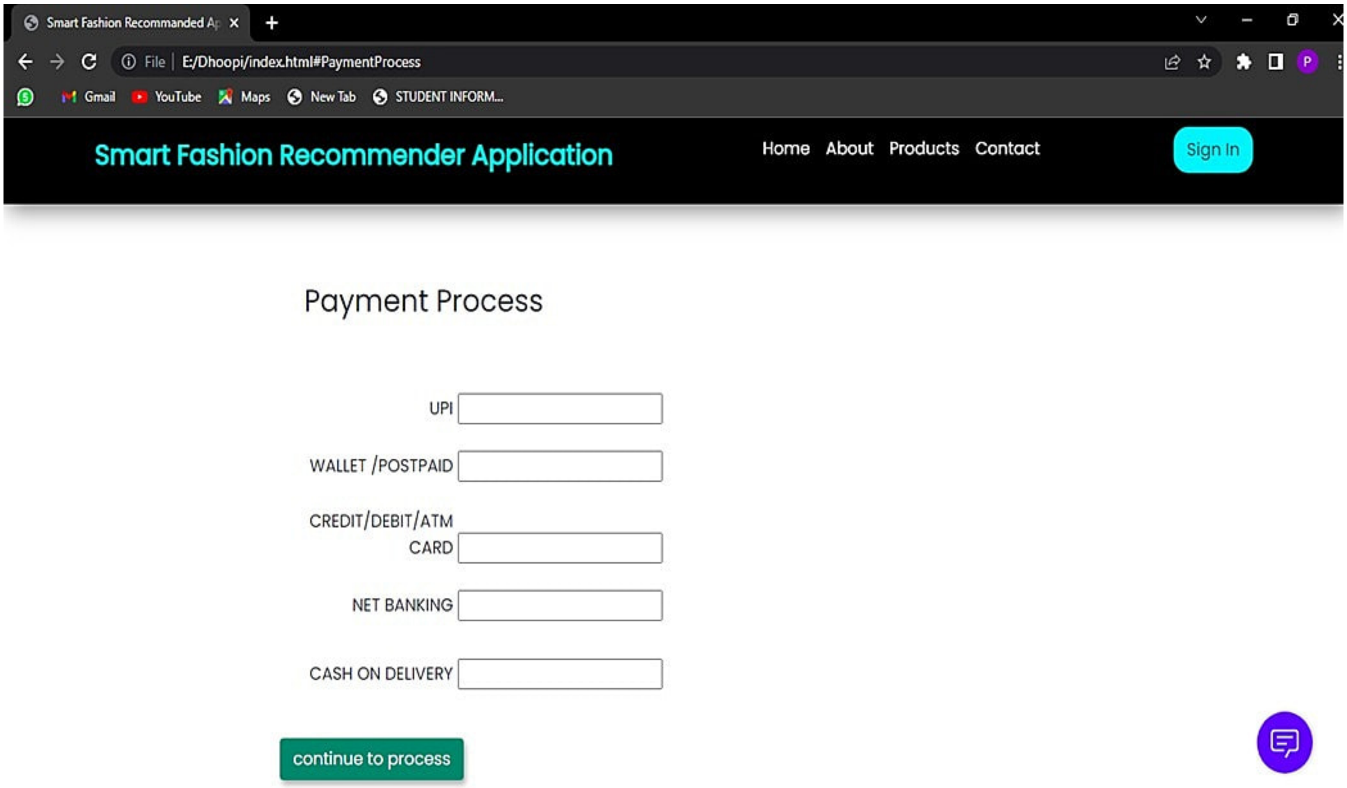
Email

Address

MobileNo

District

State


Smart Fashion Recommanded Ap
File | E:/Dhoopi/index.html#PaymentProcess
Gmail
YouTube
Maps
New Tab
STUDENT INFORM...
Smart Fashion Recommender Application
Home
About
Products
Contact
Sign In

Payment Process

UPI

WALLET /POSTPAID

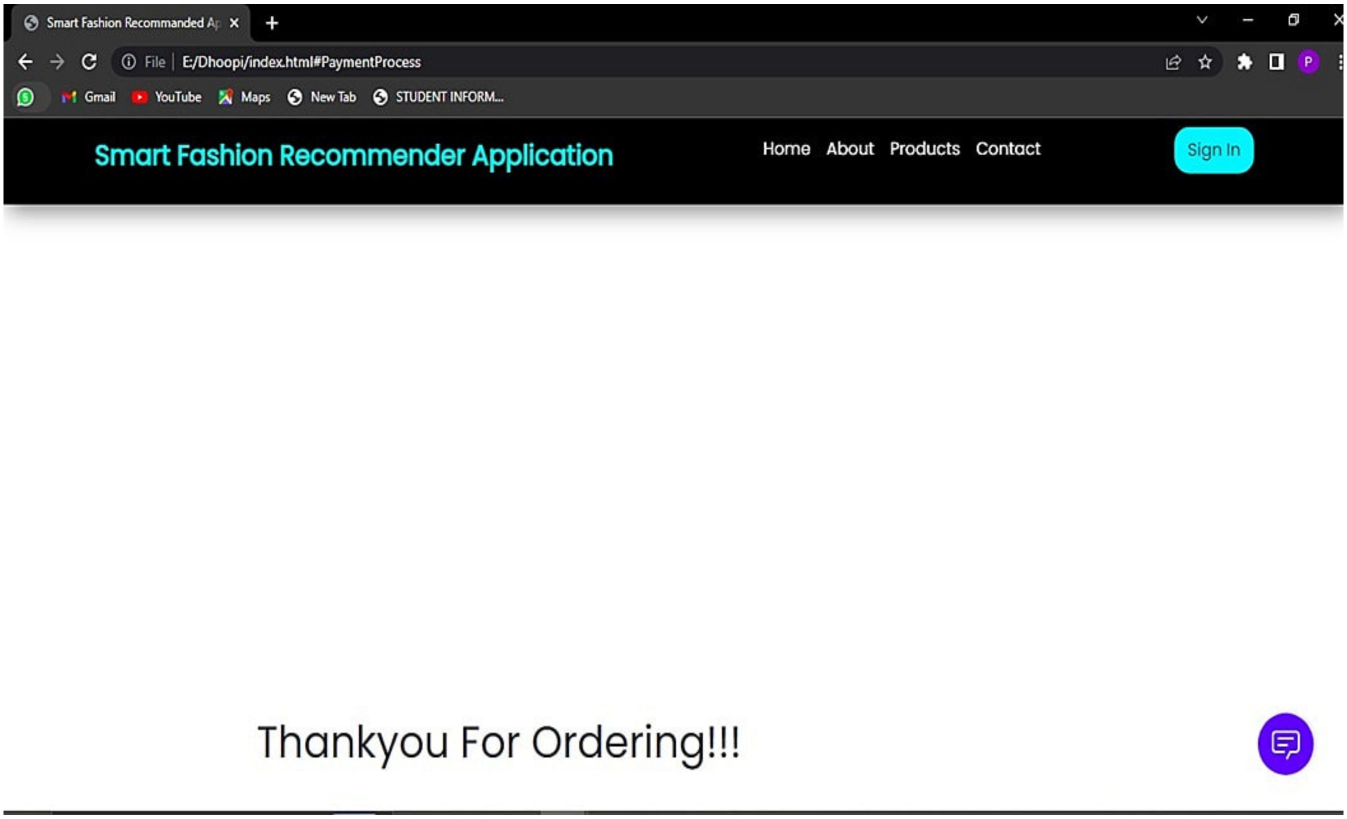
CREDIT/DEBIT/ATM CARD

NET BANKING

CASH ON DELIVERY

continue to process

Final Page:

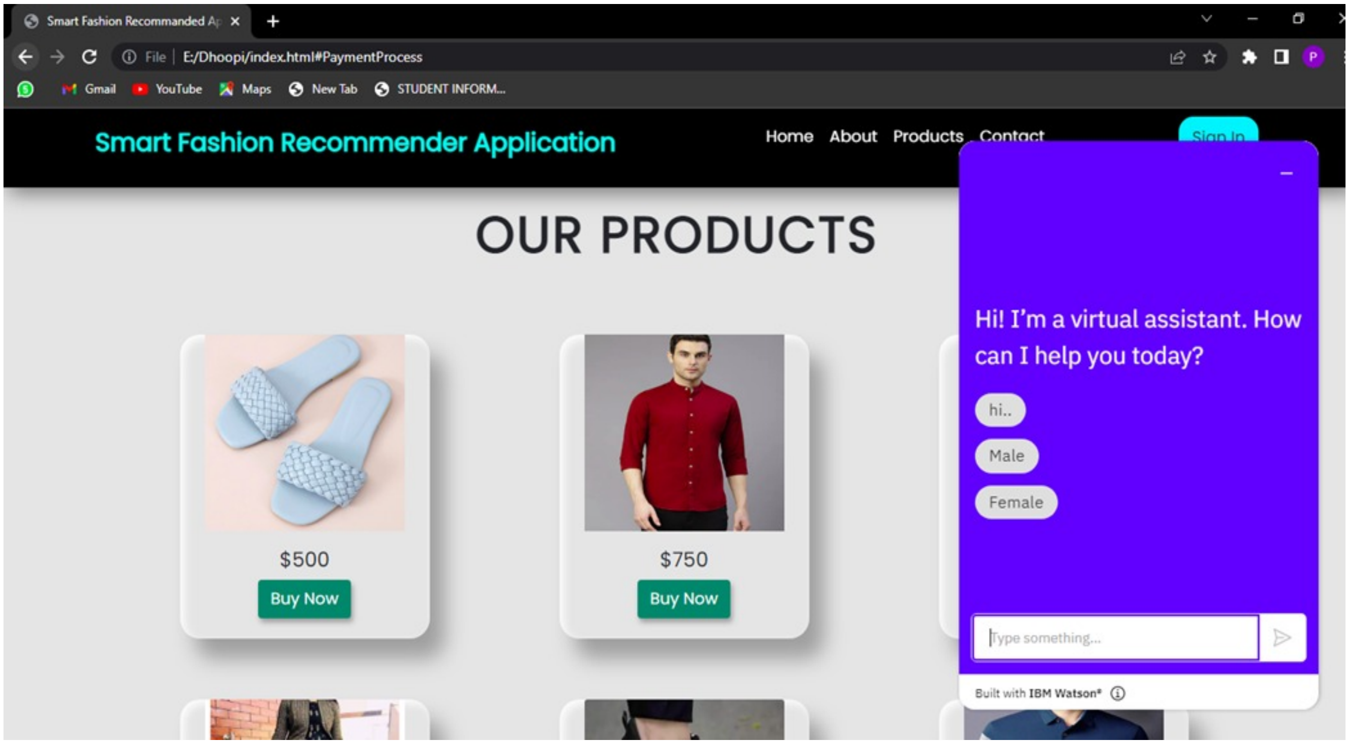

Smart Fashion Recommanded Ap
File | E:/Dhoopi/index.html#PaymentProcess
Gmail
YouTube
Maps
New Tab
STUDENT INFORM...
Smart Fashion Recommender Application
Home
About
Products
Contact
Sign In

Thankyou For Ordering!!!

b. Feature 2

Chatbot.js

```
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "67b0a2e3-5b12-4451-9f55-f7a9d920ac87", // The ID of this integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "68a44a85-8f97-48d4-b224-58ff667cbdc1", // 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>
```

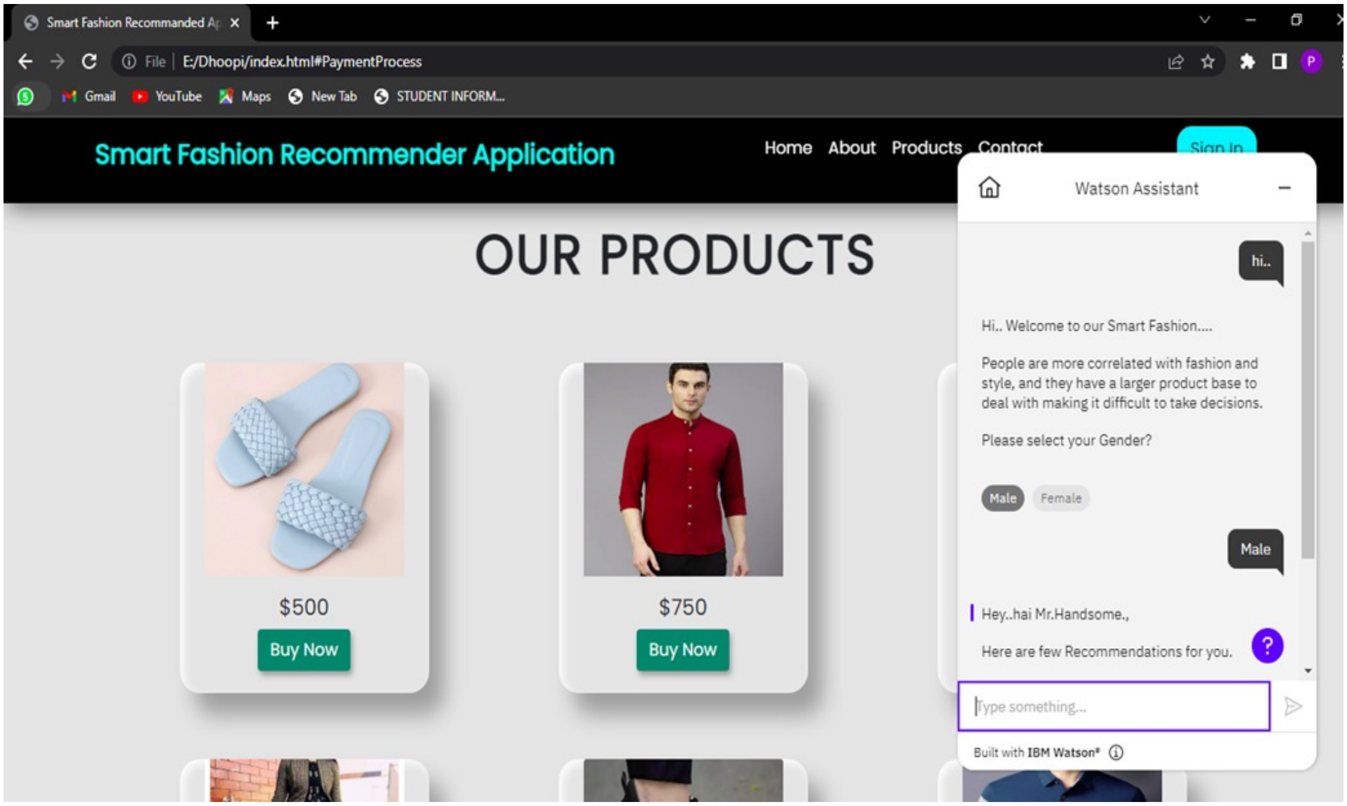


Hi! I'm a virtual assistant. How can I help you today?

hi..

Male

Female



hi..

Hi.. Welcome to our Smart Fashion....

People are more correlated with fashion and style, and they have a larger product base to deal with making it difficult to take decisions.

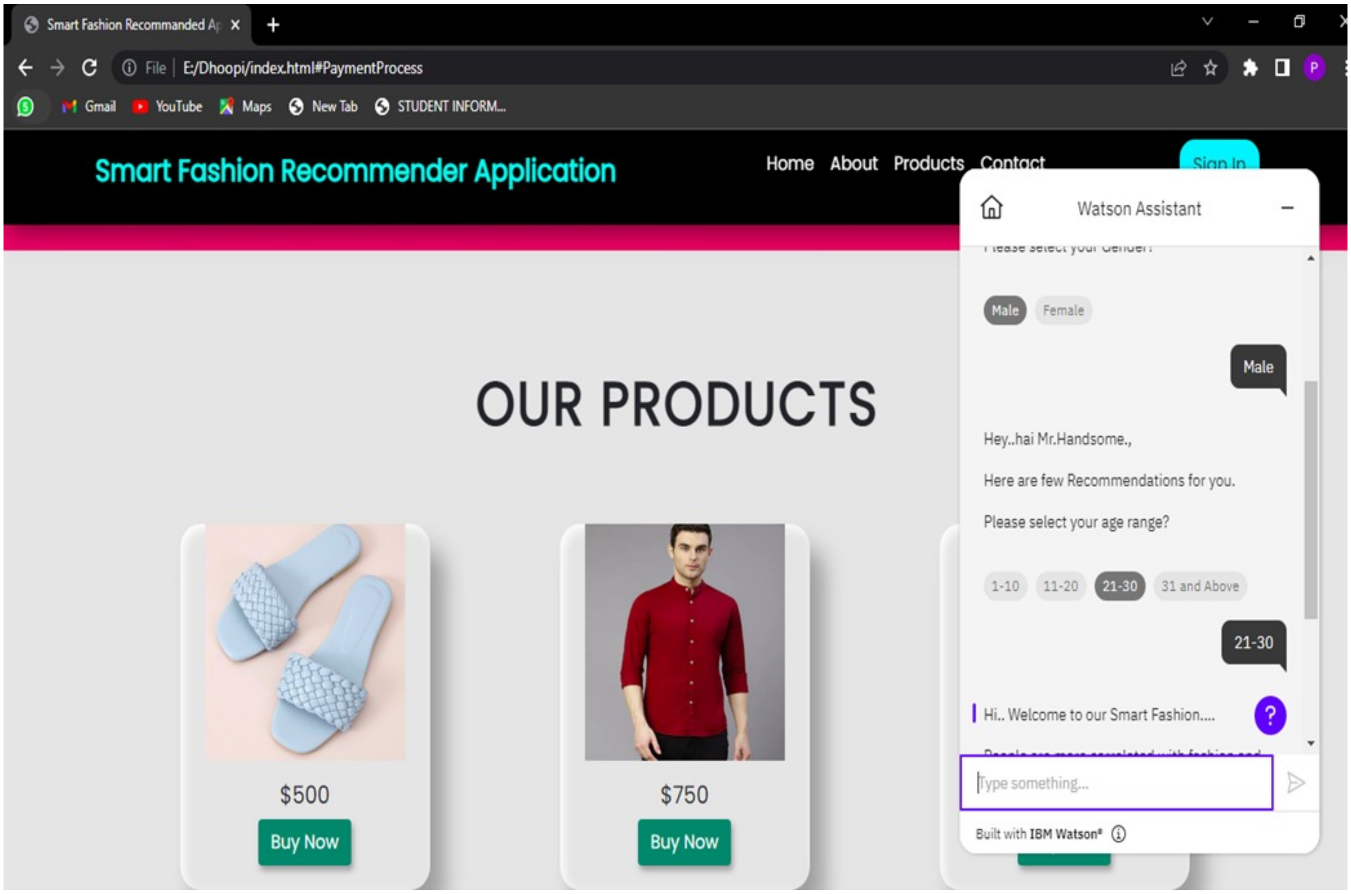
Please select your Gender?

Male Female

Male

Hey..hai Mr.Handsome.,

Here are few Recommendations for you.



Please select your Gender?

Male

Hey..hai Mr.Handsome.,

Here are few Recommendations for you.

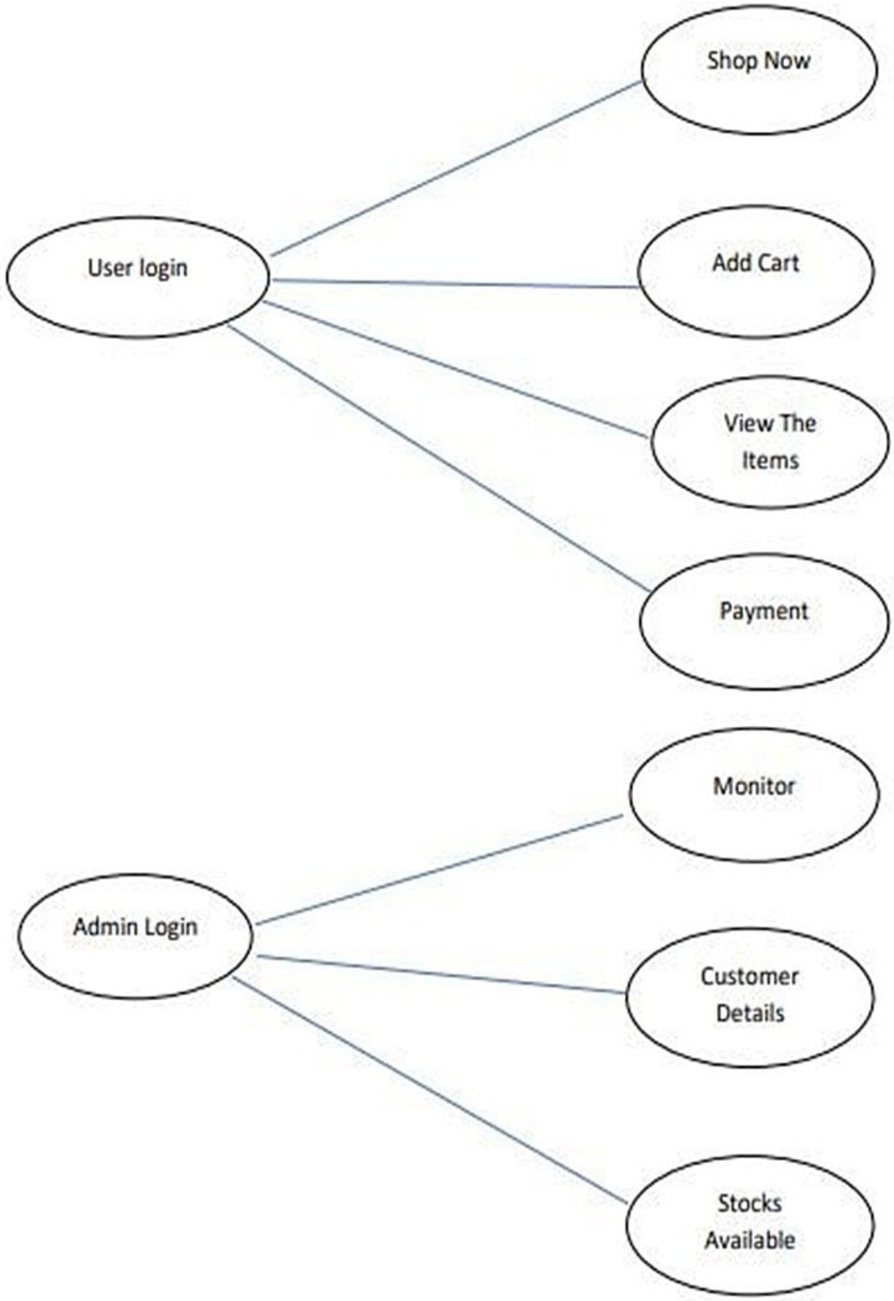
Please select your age range?

21-30

Hi.. Welcome to our Smart Fashion....

People are more associated with fashion and

c. Use Case


Shop Now
User login
Add Cart
View The Items
Payment
Monitor
Admin Login
Customer Details
Stocks Available

8. TESTING

a. Test Cases

Section	Total Case	Not Tested	Fail	Pass
Print Engine	5	0	1	4
Client Application	47	0	2	45
Security	3	0	0	3
Outsource Shipping	2	0	0	2
Exception Reporting	11	0	2	9
Final Report Output	5	0	0	5
Version Control	3	0	1	2

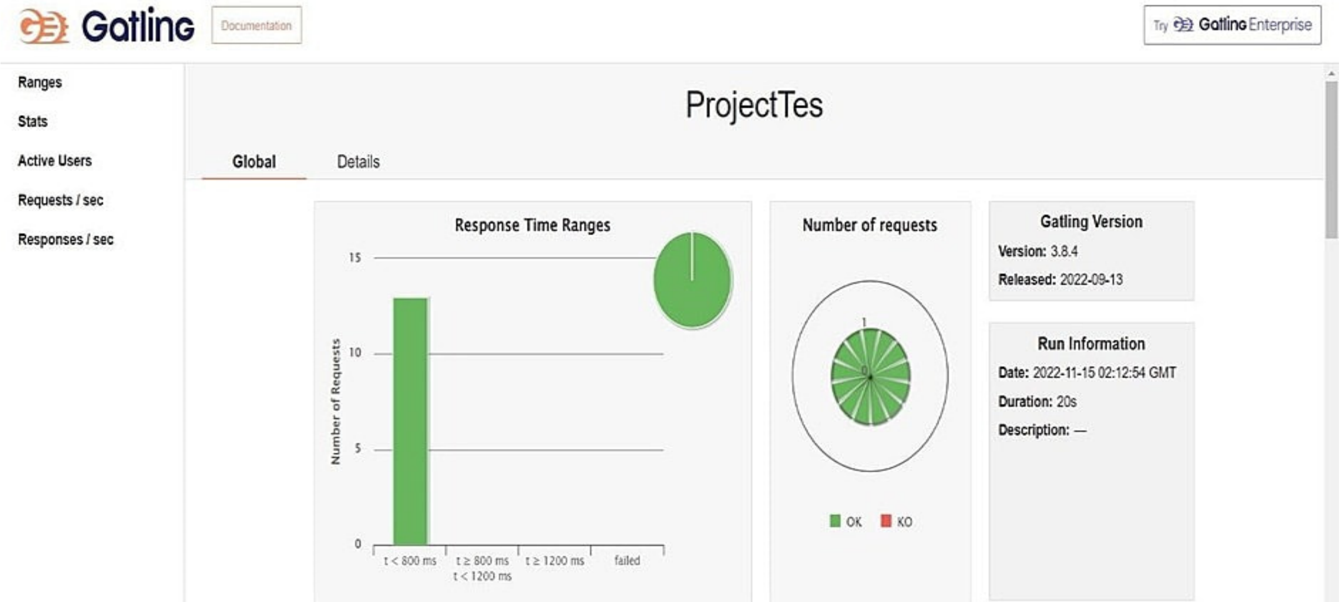
b.User Acceptance Testing

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	11	4	2	2	19
Duplicate	1	1	2	0	4
External	2	3	0	1	6
Fixed	10	2	3	20	35
Not Reproduced	0	0	2	0	2
Skipped	0	0	2	1	3
Won't Fix	0	5	2	1	8
Totals	24	15	13	25	77

c. Performance Testing

Performance testing is a non-functional software testing technique that determines how the stability, speed, scalability, and responsiveness of an application holds up under a given workload.



ProjectTes

Gatling Version

Version: 3.8.4

Released: 2022-09-13

Run Information

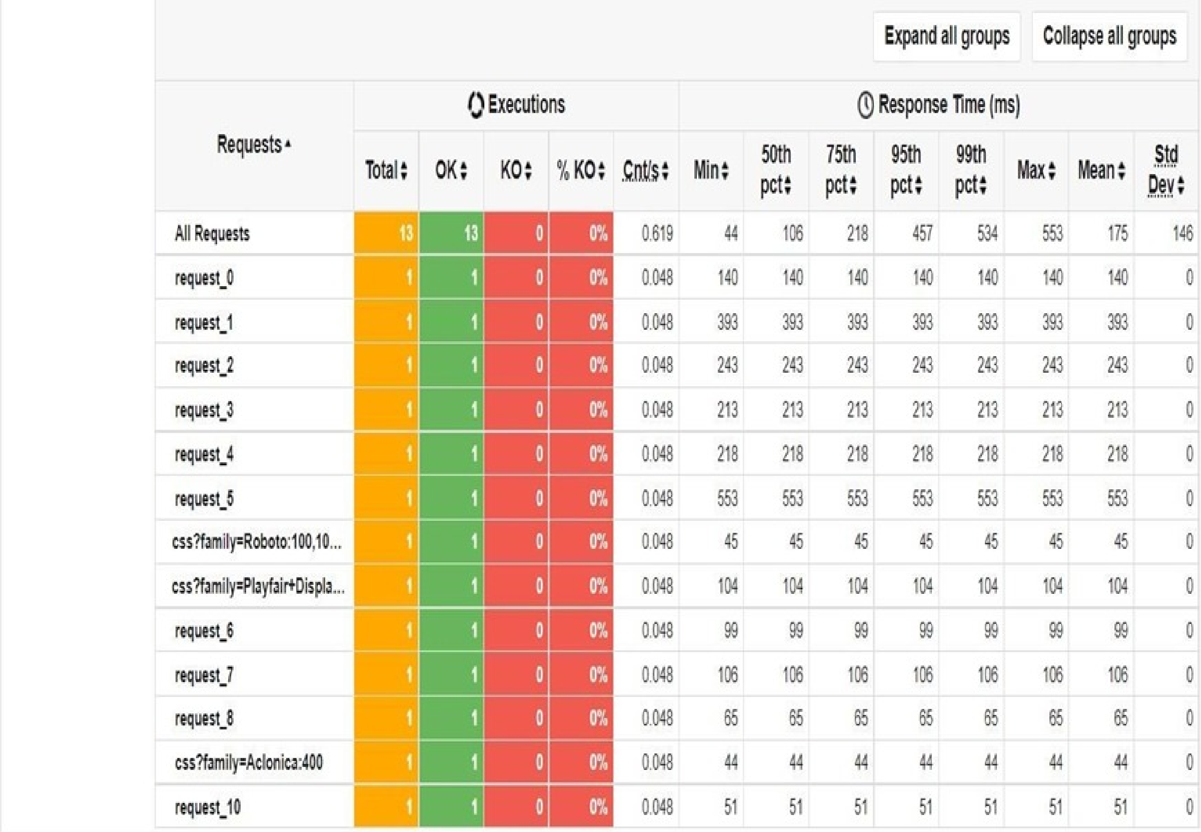
Date: 2022-11-15 02:12:54 GMT

Duration: 20s

Description: —

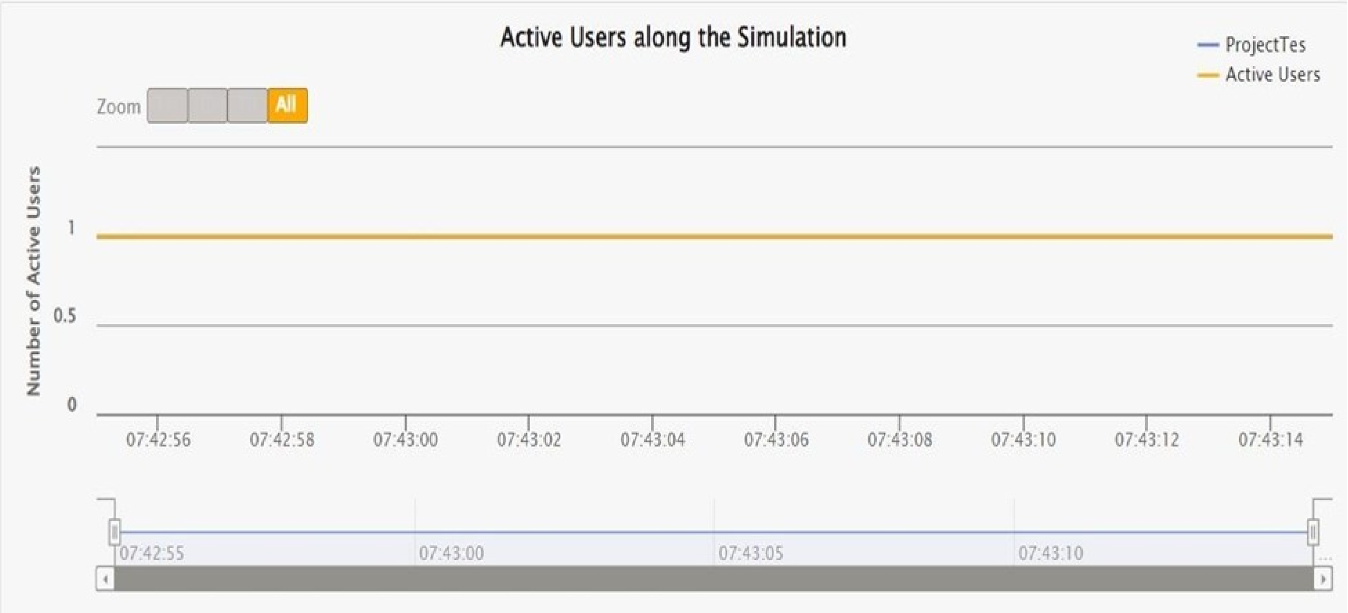
9. RESULTS

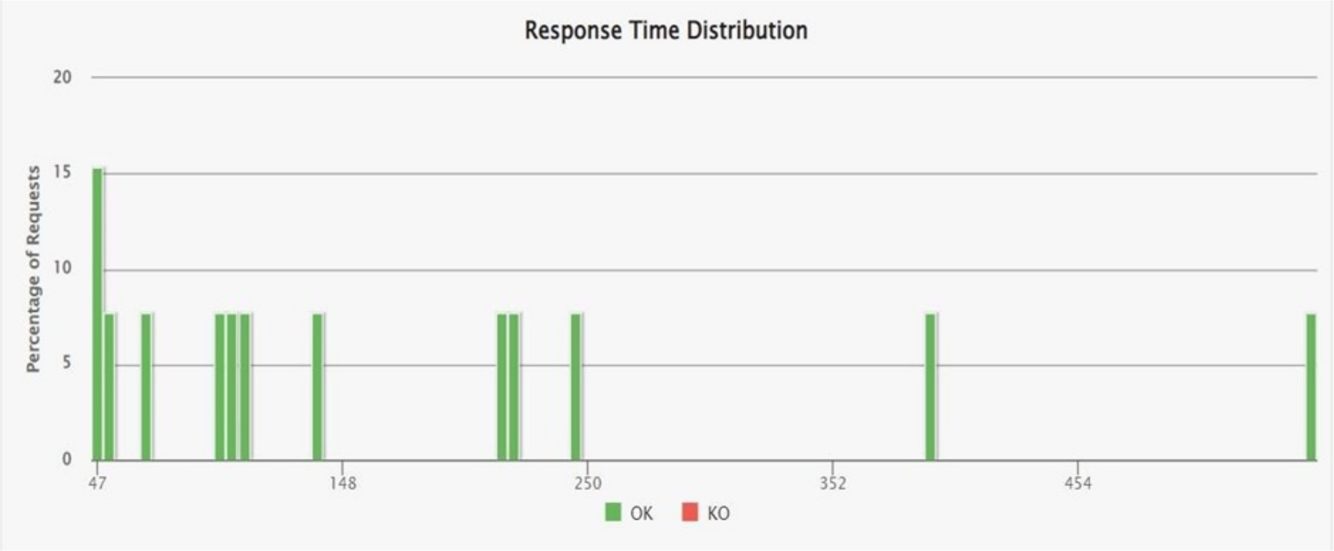
a. Performance Metrics

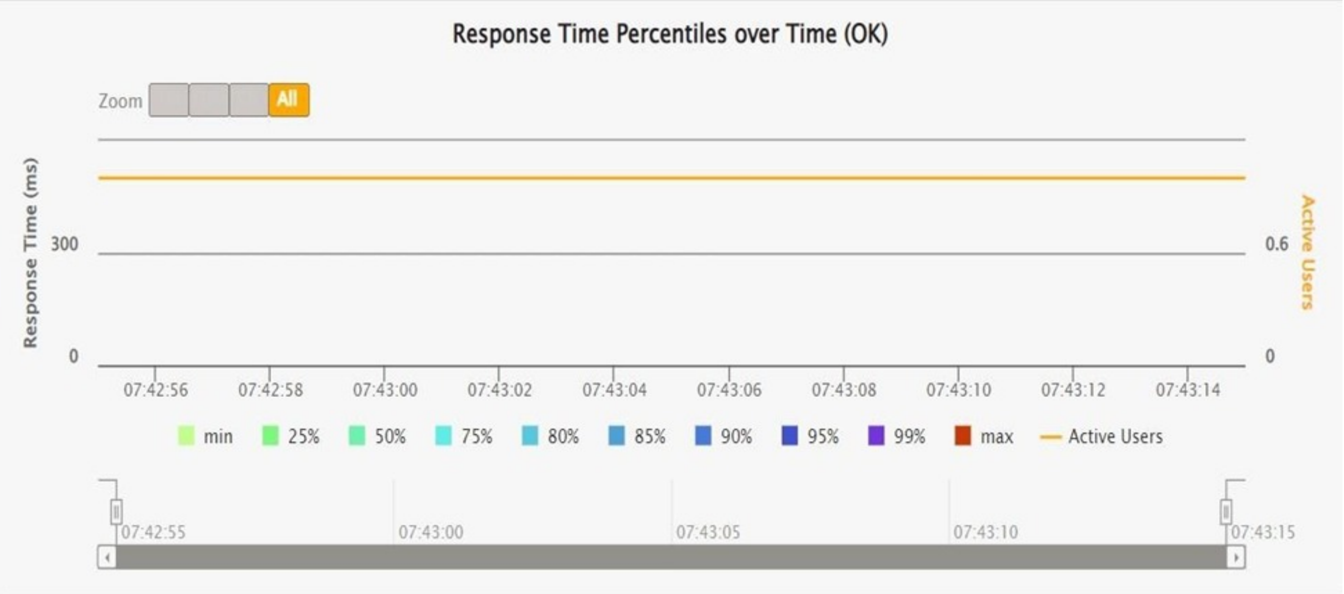


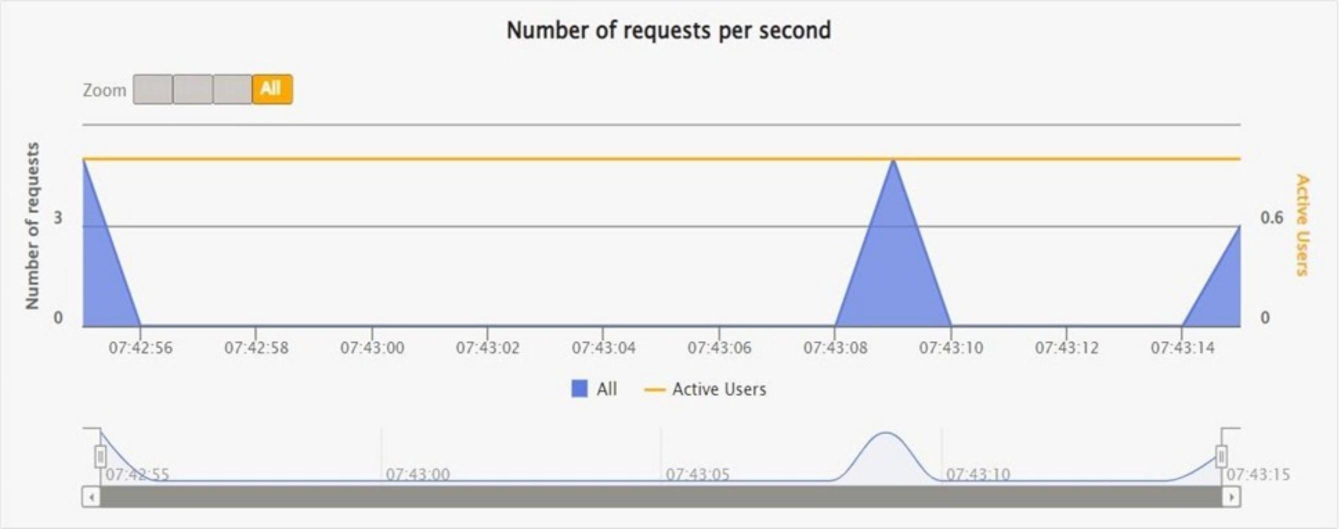
Expand all groups | Collapse all groups

Requests ▴	Executions					Response Time (ms)							
	Total ⬍	OK ⬍	KO ⬍	% KO ⬍	Cnt/s ⬍	Min ⬍	50th pct ⬍	75th pct ⬍	95th pct ⬍	99th pct ⬍	Max ⬍	Mean ⬍	Std Dev ⬍
All Requests	13	13	0	0%	0.619	44	106	218	457	534	553	175	146
request_0	1	1	0	0%	0.048	140	140	140	140	140	140	140	0
request_1	1	1	0	0%	0.048	393	393	393	393	393	393	393	0
request_2	1	1	0	0%	0.048	243	243	243	243	243	243	243	0
request_3	1	1	0	0%	0.048	213	213	213	213	213	213	213	0
request_4	1	1	0	0%	0.048	218	218	218	218	218	218	218	0
request_5	1	1	0	0%	0.048	553	553	553	553	553	553	553	0
css?family=Roboto:100,10...	1	1	0	0%	0.048	45	45	45	45	45	45	45	0
css?family=Playfair+Displa...	1	1	0	0%	0.048	104	104	104	104	104	104	104	0
request_6	1	1	0	0%	0.048	99	99	99	99	99	99	99	0
request_7	1	1	0	0%	0.048	106	106	106	106	106	106	106	0
request_8	1	1	0	0%	0.048	65	65	65	65	65	65	65	0
css?family=Aclonica:400	1	1	0	0%	0.048	44	44	44	44	44	44	44	0
request_10	1	1	0	0%	0.048	51	51	51	51	51	51	51	0




Response Time Distribution
Percentage of Requests
20
15
10
5
0
47
148
250
352
454
OK
KO


Response Time Percentiles over Time (OK)
Zoom
All
Response Time (ms)
300
0
0.6
0
Active Users
07:42:56
07:42:58
07:43:00
07:43:02
07:43:04
07:43:06
07:43:08
07:43:10
07:43:12
07:43:14
min
25%
50%
75%
80%
85%
90%
95%
99%
max
Active Users
07:42:55
07:43:00
07:43:05
07:43:10
07:43:15


Number of requests per second
Zoom
All
Number of requests
3
0
0.6
0
Active Users
07:42:56
07:42:58
07:43:00
07:43:02
07:43:04
07:43:06
07:43:08
07:43:10
07:43:12
07:43:14
All
Active Users
07:42:55
07:43:00
07:43:05
07:43:10
07:43:15

10. ADVANTAGES & DISADVANTAGES

a. Advantages

1. The key advantage is that it can better capture the dependencies among items.
2. It is useful for identifying recommendations that are as objective as possible
3. It is used to model the bias of user.
4. Based on factors that include colour,colour pattern or clothing shapes can be personalised.

b. Disadvantages

1. Inability to capture changes in user behavior.
2. Lack of data analytics capability.
3. It provides the user with too many choices.
4. Inaccurate estimations of customer's preferences.

11. CONCLUSION

We have identified a trend towards the use of chat-bots as co-creators of value during purchase through different roles specific to fashion. Thus it helps to achieve next level satisfaction of the customers for whatever product they opt for and also fulfill the necessaries of the vendors, market analyst and other such actors.

12. FUTURE SCOPE

The majority of fashion recommendations are focused on predicting the “present” based on previous interactions and trends, i.e., what the user will do right now based on their history. How might such models be utilized to make predictions regarding the fashion trends of the future? This aspect can be linked to popularity forecasting in the fashion domain since trendy items will likely be popular. Beyond these, fashion recommender systems are beginning to intersect with related domains including conversational models, models involving text, and augmented reality in future.

13. APPENDIX

a. Source Code Flask.py

```
from flask import Flask,
render_template app =
Flask(__name__)
@app.route("/") def homepage():
    return render_template("index.html",title="Home Page")
@app.route("/Ad
min") def
Admin():
    return render_template("Admin.html",title="LOGIN")
@app.route("/U
ser") def User():
    return render_template("User.html",title="LOGIN")
@app.route("/Det
ail") def Detail():
    return  render_template("Detail.html",title="Detail")
@app.route("/Conta
ct1") def Contact1():
    return  render_template("Contact1.html",title="Contact1")
```

```
@app.route("/Cont
act") def Contact():
      return  render_template("Contact.html",title="Contact")
@app.route("/C
art") def Cart():
      return  render_template("Cart.html",title="Cart")
@app.route("/Success")
def Success():
      return  render_template("Success.html",title="Success")
@app.route("/Ite
ms") def Items():
      return  render_template("Items.html",title="Items")
@app.route("/Ab
out") def About():
                                                  return
render_template("About.html",title="About")    if
__name__=="__main__":
app.run(debug=True)
```

dockerfile

```
FROM python:3.6
WORKDIR /app
ADD . /app
COPY requirements.txt /app
RUN python3 -m pip install -r requirements.txt
#RUN python3 -m pip
install ibm_db RUN
python3 -m pip install
requests
EXPOSE 8080
CMD ["python","app.py"]
```

requirements.txt

```
flask
ibm_
db
bcrypt
```

Database.py

```
from flask import Flask , render_template , request , redirect , url_for , session
import ibm_db import re
app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f
46df-8954-
7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733
;SECURITY=SSL;SSLServiceCertificate=certificate.crt;UID=jbh88994;PWD=
5spa7S6qozz1mYub",",")
@app.ro
ute('/')
def
homer():
return
render_template('home.html')
@app.route('/login',methods=['GET'
,'POST']) def login(): global
userid msg=" " if
request.method == 'POST' :
username = request.form['username'] password =
request.form['password'] sql = "SELECT * FROM users
WHERE username =? AND password=?" stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,username)
ibm_db.bind_param(stmt,2,password) ibm_db.execute(stmt)
```

```
account = ibm_db.fetch_assoc(stmt) print (account) if account:

session['loggedin'] = True session['id'] = account['USERNAME'] userid=
account['USERNAME']

session['username'] = account['USERNAME'] msg = 'Logged in successfully !'

msg = 'Logged in successfully !' return render_template('dashboard.html', msg =
msg) else:

msg = 'Incorrect username / password !'

  return render_template('login.html', msg =

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

def register(): msg =" " if request.method ==

'POST' : username =

request.form['username'] email =

request.form['email'] password =

 request.form['password'] sql = "SELECT * FROM users WHERE username =?"
stmt = ibm_db.prepare(conn, sql) ibm_db.bind_param(stmt,1,username)
ibm_db.execute(stmt) account = ibm_db.fetch_assoc(stmt) print(account) if
account:

  msg = 'Account already exists !' elif not re.match(r'[^@]+@[^@]+\.[^@]+',
email):

  msg = 'Invalid email address !' elif not re.match(r'[A-Za-z0-9]+', username): msg
= 'name must contain only characters and numbers !' else:

insert_sql = "INSERT INTO users VALUES

(?, ?, ?)" prep_stmt = ibm_db.prepare(conn,

insert_sql)ibm_db.bind_param(prep_stmt,

1, username)

ibm_db.bind_param(prep_stmt, 2, email)
ibm_db.bind_param(prep_stmt, 3, password)

ibm_db.execute(prep_stmt) msg = 'You have successfully registered !'

elif request.method == 'POST': msg = 'Please fill out the form !'

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

@app.route('/dashb oard') def dash():

    return render_template('dashboard.html')

@app.route('/apply',methods =['GET', 'POST']) def apply(): msg =" "

 if request.method == 'POST' : username = request.form['username'] email =
request.form['email'] qualification= request.form['qualification'] skills =
request.form['skills'] jobs = request.form['s']

    sql = "SELECT * FROM users WHERE username =?" stmt =
ibm_db.prepare(conn, sql) ibm_db.bind_param(stmt,1,username)
ibm_db.execute(stmt) account = ibm_db.fetch_assoc(stmt) print(account) if
account:

    msg = 'there is only 1 job position! for you'

 insert_sql = "INSERT INTO jobs VALUES (?, ?, ?, ?,

 ?)" prep_stmt = ibm_db.prepare(conn,

insert_sql) ibm_db.bind_param(prep_stmt, 1,

username) ibm_db.bind_param(prep_stmt, 2,

email) ibm_db.bind_param(prep_stmt, 3,

qualification) ibm_db.bind_param(prep_stmt, 4,

skills) ibm_db.bind_param(prep_stmt, 5, jobs)

ibm_db.execute(prep_stmt) msg = 'You have

successfully applied for job !' session['loggedin']

= True return render_template('apply.html', msg
= msg) elif request.method == 'POST': msg =
'Please fill out the form !' return
render_template('apply.html', msg = msg)


"""@app.route('/dis
play') def display():
 print(session["username"],session['id'])


cursor = mysql.connection.cursor()
cursor.execute('SELECT * FROM job WHERE userid = % s', (session['id'],))
account = cursor.fetchone() print("accountdislay",account) return
render_template('display.html',account = account)"""
@app.route('/lo gout') def logout():
session.pop('loggedin', None)
session.pop('id', None)
session.pop('username', None)
 return render_template('home.html')
 if __name__ == '__main__':
app.run(host='0.0.0.0')
```

b. GitHub Link

GitHub link: https://github.com/IBM-EPBL/IBM-Project-32110-1660208098