SMART FASHION RECOMMENDER APPLICATION

Team ID: PNT2022TMID49820

Team Members:

M.Karthick Raja

R.Hirthic Shyam

A.Edwin Kirubakaran

R.Muthu Karthick

INDEX

1. INTRODUCTION

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 detailsand orders in the database.Chatbots can also help in collecting customerfeedback and Application hosted in the python Flask.

Purpose

Weaim 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 accurateand quick productsearch. 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 desirableproducts and place order by doingpayments.Add their items in cart.

2.LITERATURE

SURVEY:

EXISTING PROBLEM

People find it difficultto navigate throughpages 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 wantsas they think. It may not resultin fulfilling the user searchand requirements. The era of recommendation systemsoriginally started in the 1990sbased on the

widespread research progress in Collective Intelligence. During this period, recommendations were generally provided to consumers based on their rating structure.

References

a. F. Ricci, L. Rokach and B. Shapira, "Introduction to recommender systems handbook, "in Recommender SystemsHandbook, Eds. F. Ricciet al.Springer US, pp. 1-35, 2011.
b. Dietmar Sannach,Ahtsham Manzoor, WanlingCai, Li Chen, "A Surveyon Conversational Recommendational Systems", May 2021.
c. T. Sekozawa,"One to one recommendation system for apparel onlineshopping", WSEAS Transaction on Systems, vol.9, no1, pp. 94-103, 2010.
d. W.K. Wong, X.H. Zeng, W.M.R. Au and P.Y.Mok, S.Y.S. Leung, "Afashion mixand-match expert system for fashion retailer using fuzzy screening approach", Expert Systems with Applications, vol.36, pp. 1750-1764, 2009"2009.
e. 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".

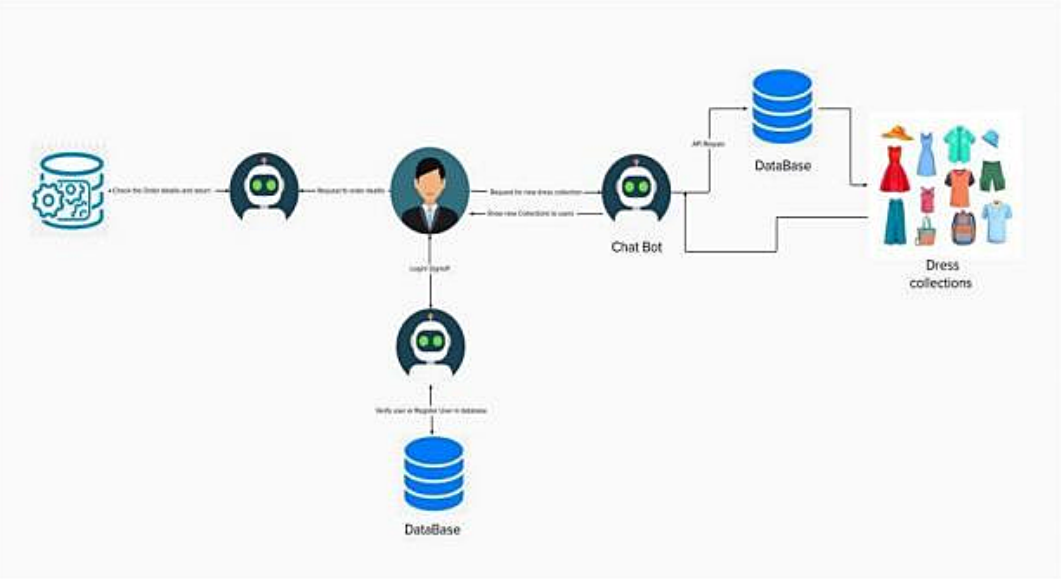
f. Neera Sanjay Agashe, "Product Recommender

Chatbot", ISSN: 2278-0181,June 2021.

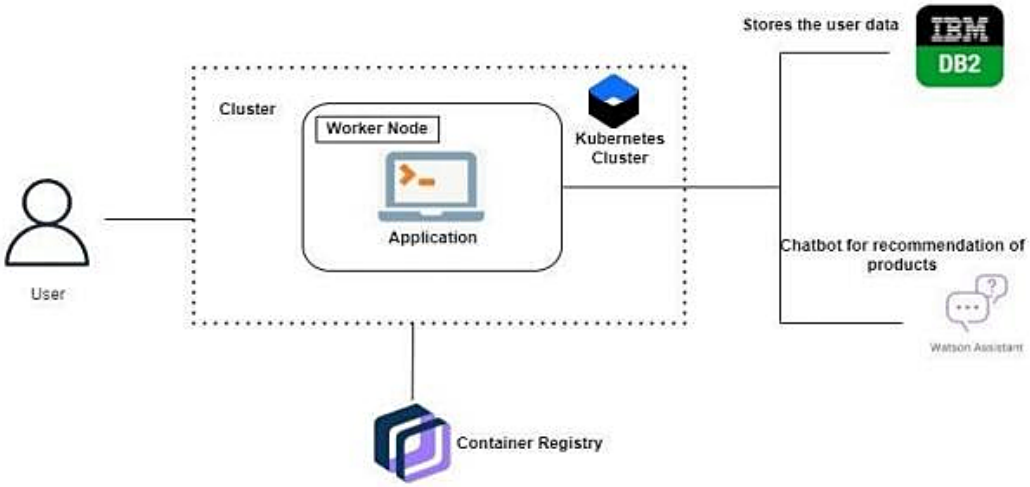
Problem Statement

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.



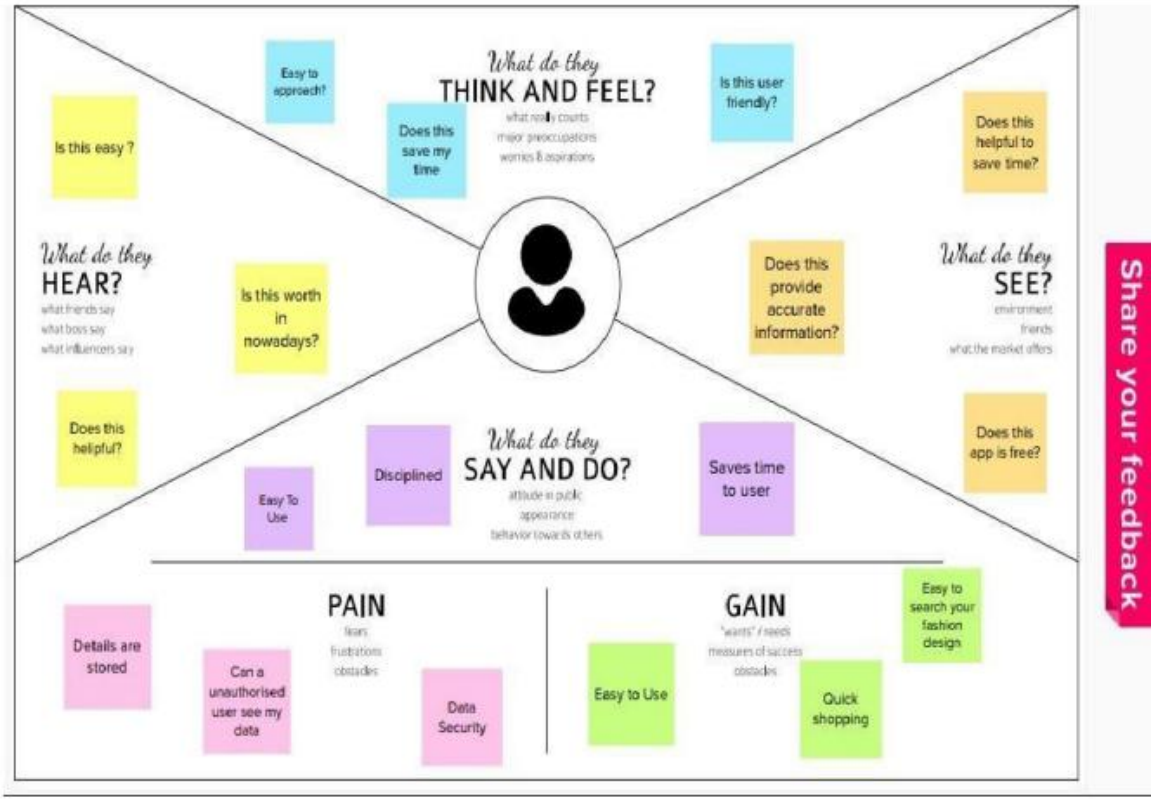
Solution & Technical Architecture:



2.IDEATION&PROPOSED

SOLUTION

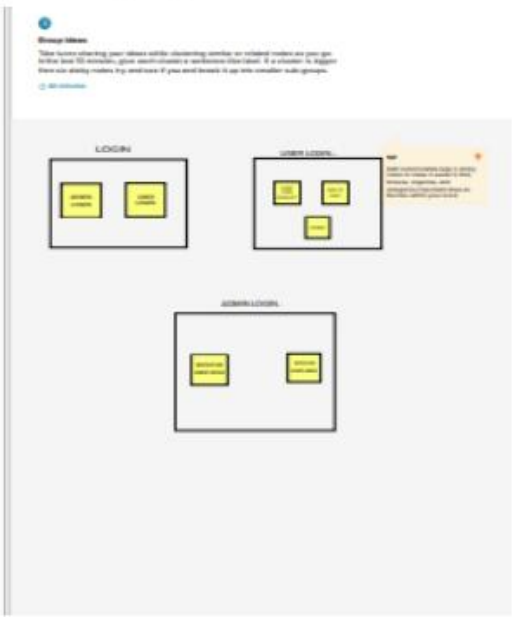
Empathy Map Canvas

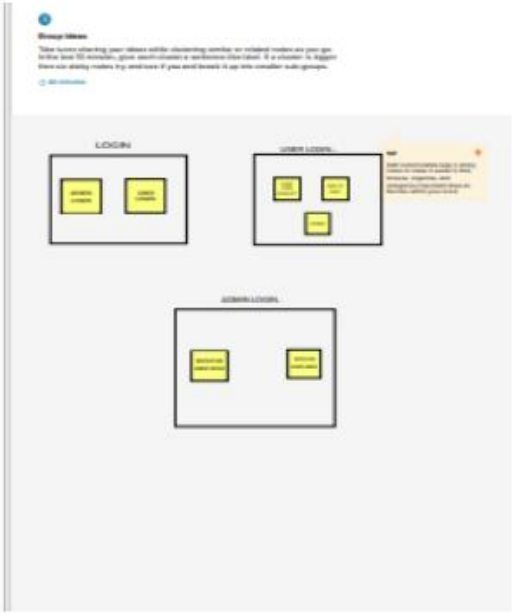


Ideation &Brainstorming

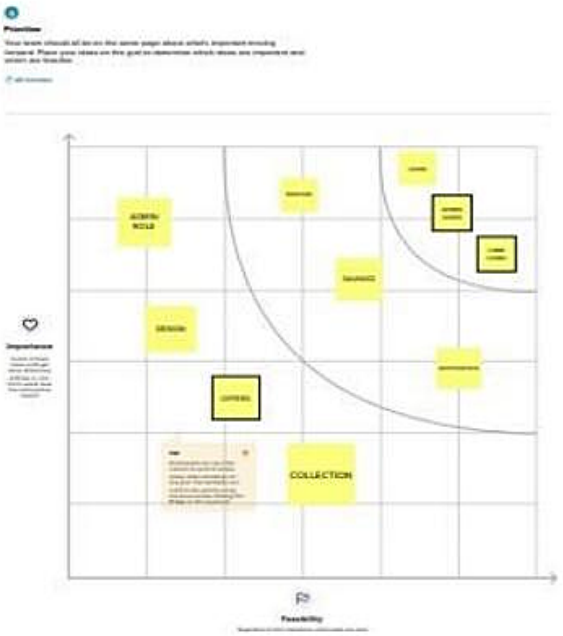
Step-1: Define Your ProblemStatement &Team Gathering.

Step-2: Brainstorm, Idea Listing and Grouping


LOGIN


LOGIN

Step-4: Prioritize



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

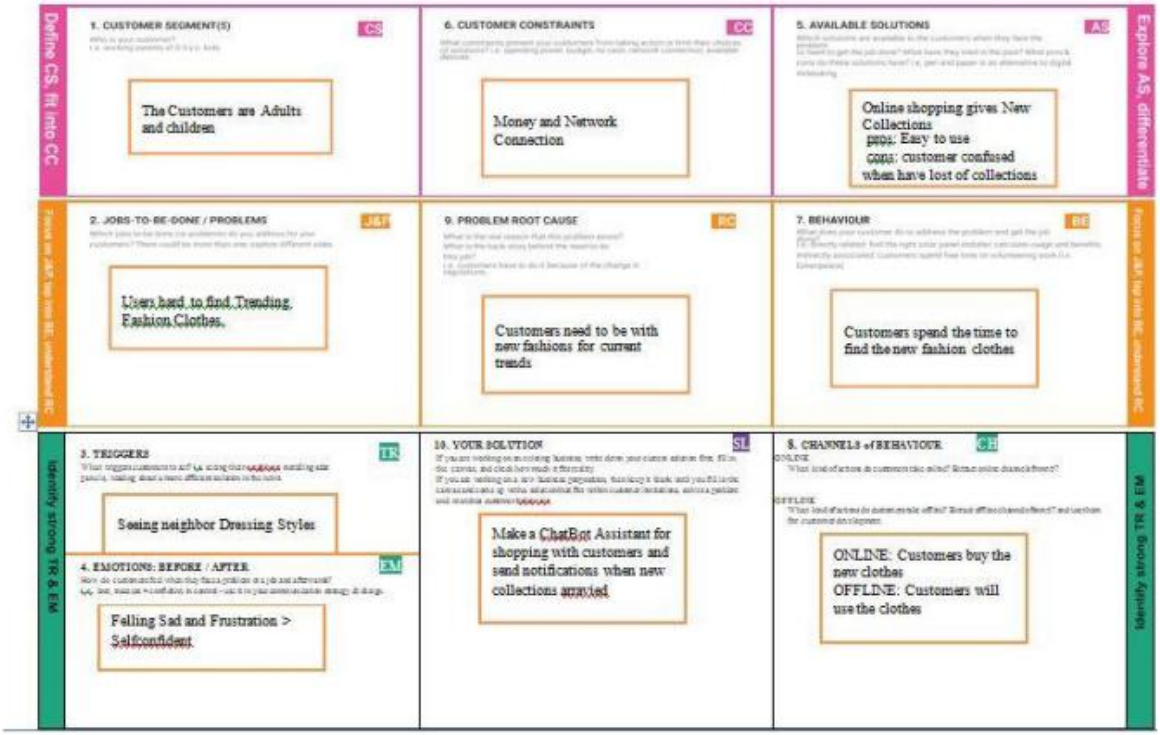
Problem Solutionfit

Providing fashion recommendation using chatbot.You can directly do your online shoppingbased on your choice withoutany 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'sinformation and orders.

Project Title: *Smart Fashion Recommender Application*

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID49820



REQUIREMENT ANALYSIS

Functional requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Registration through Form

FR-2	User Interaction	Interact throughthe 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 fromchat Bot

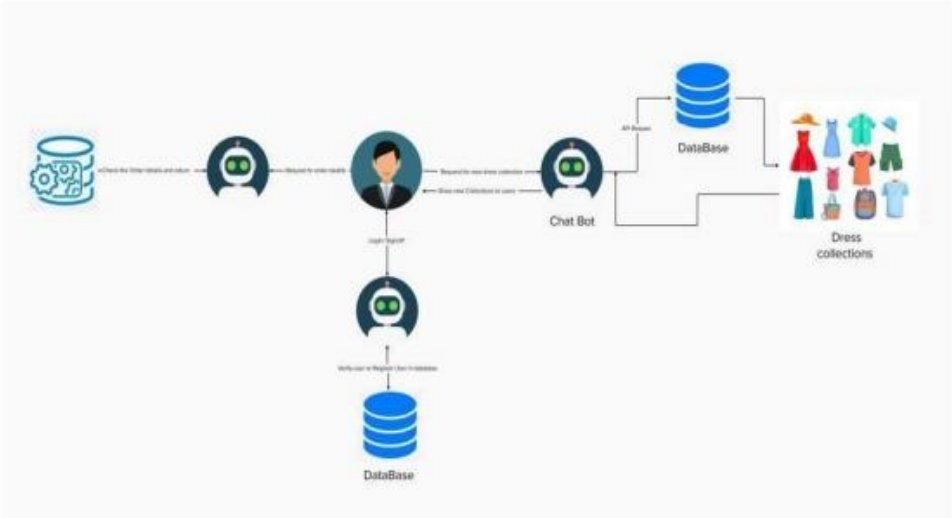
Non-functional Requirements:

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	Using Android or IOS or windowsapplications.
NFR-2	**Security**	The userdata is storedsecurely in IBMcloud.
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 usersand products.

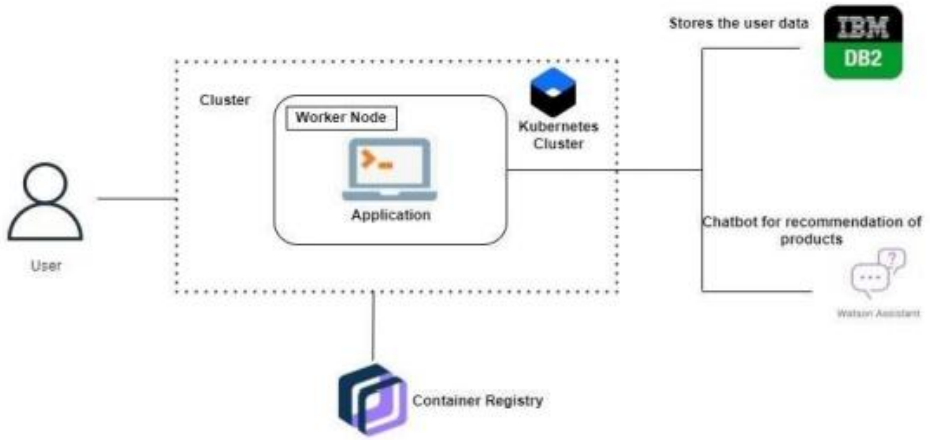
5.PROJECT DESIGN

Solution & Technical Architecture:

Data Flow Diagrams:



Solution & Technical Architecture:



User Stories:

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

2. PROJECT PLANNING &SCHEDULING

Sprint Planning & Estimation:

Milestones	Activities	Description
Ideation Phase	Literature Survey	Literature surveyon the selected project &information gathering

	Empathy Map	Prepare Empathy map to capture the user Panis & Gains,prepare list of problemstatement
	Ideation	Organizing the brainstroming session and prioritise the top 3 ideas basedon feasibility & Importance
Project DesignPhase I	Proposed Solution	Prepare proposed solution document which includes novelty, feasibility of ideas,business model, social impact, Scalability of solution
	Problem Solution Fit	Prepare problemsolution fitdocument
	Solution Architecture	Prepare solution architecturedocument
Project DesignPhase II	Customer Journey	Prepare customer journey map to understand the user interactions & experience with theapplication
	Functional requirement	Prepare functional & nonfunctional requirement document
	Data FlowDiagram	Prepare DataFlow Diagramanduser stories
	Technology architecture	Draw the technology architecture diagram

Project Planning Phase	Milestones &Activity list	Prepare milestones and activity listof the project
	Sprint Delivery Plan	Prepare sprint delivery plan

CODING & SOLUTIONING

```
<html>
<head>
<title>SmartFashion Recommender</title>
<link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css">
<script src='https://kit.fontawesome.com/a076d05399.js'
crossorigin='anonymous'></script>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<script>
</script>
<style>
form{
background-color:rgb(154, 123, 255);
border-style: solid;
border-color: white;
padding: 20px 30px;
border-radius:10px;
box-shadow:4px 4px 4px 4px black;
margin-left:250px;
}
.login-block {
float:left;
margin:0 49px 25px 0;
}
h1{
color:black;
font-family:algerian;
}
```

```
color:black;
font-size:17pt;
font-family:timesnewroman;
}
label{
font-family:timesnewroman;
}
.button {
border-radius:15px;
padding:2px 15px;
background-color:blue;
color:white;
cursor: pointer;
font-size:10pt;
font-family:timesnewroman;
}
body{
  width:100%;
  height:100%;
  background-image:
url(`https://cdn.pixabay.com/photo/2018/03/25/22/54/background-3261090_960_720.jpg`);
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-color: azure;
```

```
  background-size: cover;
}
.w3{
    background-color:rgb(241, 73, 73);
    padding: 10px;
    border: none;
    width: 500%;
    color:white;
    position: relative;
    top: 0px;
    font-size:15pt;
}
input{
font-family:timesnewroman;
}
</style>
</head>
<body>
<body style="background-image: url('static/img17.jpg')">
    <!-- <h1><center>SMARTFASHION RECOMMENDER
SYSTEM</center></h1> -->

  <header class="w3">
        <a href='/' class='topnav-icons fa fa-home w3-bar-item w3-
button' title='Home'>Home</a>
        <a href='/reg' class='topnav-icons fa fa-home w3-bar-item
w3-button' title='Home'>Home</a>
```

```
        <a href='/About' class='topnav-icons fa fa-comments-o w3-bar-item w3-button' title='About'>About</a>

        <a href='/reg'   class='topnav-icons fa fa-address-card-ow3-bar-item w3-button' title='Contact'>Register</a>

  </header>

<div class="login-block">
    <h3>Admin Login</h3>
<form style="width:300px" action="http://127.0.0.1:5000/login" method="POST" >
<label><b>Username : </label><br><input type=text id="uname"><br>
<br>
<label>Password :</label> <br><input type=password id="pwd"><br>
<br>
<input type="submit" class="button">LOGIN</input>
    </form>
</div>
<div class="login-block">
    <h3>User Login</h3>
<form style="width:300px" action="http://127.0.0.1:5000/login" method="POST">
<label>Username :</label> <br><input type=text id="field1" name="username"><br>
<br>
<label>Password :</label> <br><input type=password
```

```
name="password" id="field2"><br>

<br>
<input type="submit" class="button" >LOGIN</input>
</form>
</div>



</body>
</html>
```

Userpanel.html

```
<html>
<head>
<title>User Panel</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.mi
n.js"></script>

<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.2/css/all.min.css">
<script>
    window.watsonAssistantChatOptions = {
        integrationID: "9e1f0630-1095-4d2f-9215-c1750a168ddf", // The
ID of this integration.
        region: "jp-tok", // The region your integration is hosted
in.
        serviceInstanceID: "cd93e7bc-8deb-4aec-aa9e-5247a38c7653", //
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);
    });
$(document).ready(function(){
        $('#menu').click(function(){
                $(this).toggleClass('fa-times');
                $('.navbar').toggleClass('nav-toggle');
        });
        $('#login').click(function(){

                $('.login-form').addClass('popup');
        });
        $('.login-form form.fa-times').click(function(){
                $('.login-form').removeClass('popup');
        });
        $(window).on('load scroll',function(){
                $('#menu').removeClass('fa-times');
                $('.navbar').removeClass('nav-toggle');
                $('.login-form').removeClass('popup');
                $('section').each(function(){
                        let top = $(window).scrollTop();

                    let height= $(this).height();
```

```
                    let id = $(this).attr('id');
                    let offset = $(this).offset().top - 200;
                    if(top > offset&& top < offset + height){
                        $('.navbar ul li a').removeClass('active');

$('.navbar').find(`[href="#${id}"]`).addClass('active');
                    }
                });
        });
});
</script>
<style>
a{
color:white;
font-family:algerian;
}
:root{
        --gradient:linear-gradient(80deg, blue,red);
}
*{
        font-family: 'OpenSans', sans-serif;
        margin:0; padding:0;

        border:none; outline: none;
        text-decoration: none;
        text-transform: capitalize;
        transition: all.2s linear;
}
*::selection{
```

```
        background:red;
        color:red;
}
html{
        font-size: 65.5%;
        overflow-x: hidden;
}
header{
        width: 100%;
        display: flex;
        align-items: center;
        justify-content: space-between;
        top:0; left:0;
        z-index: 1000;
        padding:1.5rem 7%;
        background:var(--gradient);
        box-shadow: 0 .1rem.3rem rgba(0,0,0,.3);
}
header .logo{
        color:white;
        font-size: 2.5rem;
}
header .navbarul{
        display: flex;
        align-items: center;
        justify-content: center;
        list-style: none;
}

header .navbarul li{
```

```
        margin:0 1rem;
}
header .navbarul li a{
        font-size: 2rem;
        color:white;
}
S
header .navbarul li a.active,
header .navbarul li a:hover{
        color:white;
}
header #login{
        font-size: 3rem;
        color:black;
        cursor: pointer;

}
header #login:hover{
        color:black;

}
header #menu{
        font-size: 3rem;
        color:blue;
        cursor: pointer;
        display: none;
}
.login-form{
        height:100%;
```

```
        width:100%;
        position: fixed;
        top:-120%; left:0;
        z-index: 1000;

        background:rgba(0,0,0,.5);
        display: flex;
        align-items: center;
        justify-content: center;
}
.login-form.popup{
        top:0;
}
.login-form form{
        width:35rem;
        background:cyan;
        margin:0 2rem;
        padding:1rem 3rem;
        border-radius: .5rem;
              box-shadow: 0 .1rem.3rem #333;
        position: relative;
}
.login-form form h3{
        font-size: 3rem;
        color:black;
        padding:1rem 0;
}
.login-form form .box{
        width:100%;
        padding:1rem 0;
```

```
        margin:1rem 0;
        border-bottom: .2remsolid #666;
        font-size: 1.6rem;
        color:green;
        text-transform: none;
}
.login-form form.box::placeholder{
        text-transform: capitalize;
}

.login-form form .box:focus{
        border-color: green;
}
.login-form form.box:nth-child(3){
        margin-bottom: 5rem;
}
.login-form form p{
        font-size: 1.4rem;
        color:#666;
        padding:.5rem 0;
}
.login-form form p a{
        color:#f39c12;
}
.login-form form .btn{
        width: 100%;
        margin:2rem 0;
}
.login-form form .btn:hover{
        background:#444;
```

```
}
.login-form form .fa-times{
        position: absolute;
        top:1.5rem; right:1rem;
        font-size: 2.5rem;
        cursor: pointer;
        color:#999;
}
.login-form form .fa-times:hover{
        color:#444;
}
h2{
background-color:blue;
width:1357px;

height:46px;
color:white;
}
.btn {
    border:1px solidblack;
    background-color: inherit;
    padding: 7px 28px;
    font-size: 16px;
    cursor: pointer;
    display: inline-block;
    margin-left:-258px;
width:200px;
}
.btn1 {
    border: none;
```

```
    background-color: inherit;
    padding: 14px28px;
    font-size: 16px;
    cursor: pointer;
    display: inline-block;
}
.btn1:hover {background:gray;}
img{
padding: 25px 50px;
height:300px;
width:200px;
}
.btn:hover {
background: blue;
}
#myImg {
    border-radius: 5px;
    cursor: pointer;
    transition: 0.3s;

}
#myImg:hover {
    opacity: 0.7;
}
.modal {
    display: none;
    /* Hiddenby default */
    position: fixed;
    /* Stay in place*/
    z-index: 1;
```

```
    /* Sit on top */
    padding-top: 100px;
    /* Location of the box */
    left: 0;
    top: 0;
    width: 100%;
    /* Full width */
    height: 100%;
    /* Fullheight */
    overflow: auto;
    /* Enablescroll if needed */
    background-color: rgb(0,0, 0);
    /* Fallback color */
    background-color: rgba(0, 0, 0, 0.9);
    /* Blackw/ opacity */
}
.modal-content {
    margin: auto;
    display: block;
    width: 100%;
    height:100%;
    max-height:400px;
    max-width: 400px;
}

#caption {
    margin: auto;
    display: block;
    width: 80%;
    max-width: 700px;
```

```
    text-align: center;
    color: #ccc;
    padding: 10px 0;
    height: 150px;
}
.modal-content,
#caption {
    animation-name: zoom;
    animation-duration: 0.6s;
}
@keyframes zoom{
    from {
        transform: scale(0)
    }
    to {
        transform: scale(1)
    }
}
.close {
    position: absolute;
    top: 15px;
    right: 35px;
    color: #f1f1f1;
    font-size: 40px;
    font-weight: bold;
    transition: 0.3s;
}
.close:hover,
.close:focus {
```

```
    color: #bbb;
    text-decoration: none;
    cursor: pointer;
}
@media onlyscreen and (max-width: 700px) {
    .modal-content {
        width: 100%;
    }
}
body{
background-color:rgb(195, 255,225);
}
s{
margin-left:130px;
}
</style>
</head>
<body>
<header>
        <div id="menu" class="fas fa-bars"></div>
        <a href="/" class="logo"><i class="fas fa-user-tie"></i>SMART
FASHION RECOMMENDER APPLICATION</a>

        <nav class="navbar">
                <ul>
                        <li><a href="/cart">Cart</a></li>
                        <li><a href="/items">Items</a></li>
                       <li><a href="/contact">contact</a></li>
                  <li><a href="/">Logout</a></li>
```

```
<s>
                    <li><div id="login" class="fas fa-user-
circle"></div></l
        i></s>
                </ul>
        </nav>
        <div id="login" class="fas fa-user-circle">.</div>

</header>
<div class="login-form">
        <form action="">
                <h3>login</h3>
                <input type="email" placeholder="username" class="box">
                <input type="password" placeholder="password"
class="box">
                 <p>forget password? <ahref="#">click here</a></p>
                <p>don't havean account? <ahref="#">register
now</a></p>
                 <input type="submit" class="btn" value="login">
                <i class="fas fa-times"></i>
        </form>
</div>
<br>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-
object-storage.appdomain.cloud/11191030/img.jpg" "alt="Rs 500/-"
style="padding: 25px53px;">
<button   onclick="op()" class="btn">SHOP NOW</button>
```

<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img1.jpg" "alt="Rs 490/-"

style="padding: 25px53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud- object-storage.appdomain.cloud/11191030/img2.jpg" "alt="Rs 599/-
" style="padding: 25px 53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img3.jpg" " alt="Rs 600/-"

style="padding: 27px 52px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud- object-storage.appdomain.cloud/11191030/img4.jpg" "alt=Rs 700/-"
style="padding: 25px 53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img5.jpg" " alt="Rs 480/-"

style="padding: 25px 53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img6.jpg" " alt="Rs 650/-"

style="padding: 27px 52px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img7.jpg" " alt="Rs 800/-"

style="padding: 25px 53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img8.jpg" " alt="Rs 590/-"

style="padding: 25px 53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img9.jpg" " alt="Rs 680/-"

style="padding: 25px 53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img10.jpg" " alt="Rs

950/-" style="padding: 25px 53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img11.jpg" "alt="Rs
550/-" style="padding: 25px53px;">

<button onclick="op()" class="btn">SHOP NOW</button>

<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img12.jpg" " alt="Rs

880/-" style="padding: 25px 53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages"id="myImg" src="https://s3.che01.cloud-

object-storage.appdomain.cloud/11191030/img13.jpg" "alt="Rs

780/-" style="padding: 25px53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img14.jpg" " alt="Rs770/-"

style="padding: 25px 53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img15.jpg" " alt="Rs680/-"

style="padding: 25px 53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img18.jpg" "alt="Rs

780/-" style="padding: 25px53px;">

<button onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-storage.appdomain.cloud/11191030/img19.jpg" " alt="Rs980/-"

style="padding: 25px 53px;">

```
<button   onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-
object-storage.appdomain.cloud/11191030/img20.jpg" " alt="Rs580/-"

style="padding: 25px 53px;">

<button   onclick="op()" class="btn">SHOP NOW</button>
<img class="myImages" id="myImg" src="https://s3.che01.cloud-object-
storage.appdomain.cloud/11191030/img21.jpg" "alt="Rs
850/-" style="padding: 25px53px;">

<button   onclick="op()" class="btn">SHOP NOW</button>
<br>
<br>
<div id="myModal" class="modal">
    <span class="close">&times;</span>
    <img class="modal-content" id="img01">
    <div id="caption"></div>
</div>

<script>
// createreferences to themodal...
var modal= document.getElementById('myModal');
// to all images -- note I'm using a class!
var images= document.getElementsByClassName('myImages');
// the image in the modal
var modalImg = document.getElementById("img01");
// and thecaption in the modal
var captionText = document.getElementById("caption");
// Go through all of the images with our custom class
for (var i = 0; i < images.length; i++){
```

```
        var img = images[i];
        // and attach our click listener for this image.
        img.onclick = function(evt) {
            console.log(evt);
            modal.style.display = "block";
            modalImg.src = this.src;
            captionText.innerHTML = this.alt;
        }
    }
    var span= document.getElementsByClassName("close")[0];
    span.onclick = function() {
        modal.style.display = "none";
    }
    function op()
    {

            window.open("/detail");
    }
    </script>
    </body>
    </html>
```

Adminpanel.html

```
<html>
<head>
<title>CART</title>
```

```
<META charset="UTF-8">
<META name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<META NAME='rating' CONTENT='General' />
<META NAME='expires' CONTENT='never' />
<META NAME='language' CONTENT='English, EN' />
<META name="description" content="shopping cartproject with HTML5 andJavaScript">
<META name="keywords" content="HTML5,CSS,JavaScript, html5 session storage, html5 local storage">
<META name="author" content="dcwebmakers.com">
<script>
async function   SaveItem(e) {
                        e.preventDefault();
        varname = document.forms.ShoppingList.name.value;
        vardata = document.forms.ShoppingList.data.value;
        //localStorage.setItem(name, data);
        const product={name,data};
        try{
        const result=await fetch("http://127.0.0.1:5000/products/delete",
        {method :"POST",mode:"no-cors",headers:{"Access-Control-Allow-Origin:'*'"},body:JSON.stringify(product)}).
        then(r=>{
                console.log(r.json())
        });
        doShowAll(rs=>{
```

```
                console.log(rs);
        });
}catch(err){
        console.log(err);
}

}
function ModifyItem() {
            var name1= document.forms.ShoppingList.name.value;

              var data1= document.forms.ShoppingList.data.value;

        //check if name1 is already exists

//check if key exists
                        if(localStorage.getItem(name1) !=null)
                        {

                            //update
                            localStorage.setItem(name1,data1);
                            document.forms.ShoppingList.data.value =
localStorage.getItem(name1);
                        }


        doShowAll();
}
function RemoveItem() {
        varname = document.forms.ShoppingList.name.value;
        document.forms.ShoppingList.data.value =
localStorage.removeItem(name);
```

```
        doShowAll();
}
function ClearAll() {
        localStorage.clear();
        doShowAll();
}
function doShowAll() {
        if(CheckBrowser()) {
                varkey = "";
                varlist = "<tr><th>Item</th><th>Value</th></tr>\n";
                vari = 0;
                //for more advance feature, you can set cap on max
items in the cart
                    for (i = 0; i <=localStorage.length-1; i++) {
                        key= localStorage.key(i);
                        list += "<tr><td>" + key + "</td>\n<td>"
                                    +localStorage.getItem(key) +
"</td></tr>\n";
                }
                //if no item existsin the cart
                if(list == "<tr><th>Item</th><th>Value</th></tr>\n")
{
                        list +=
"<tr><td><i>empty</i></td>\n<td><i>empty</i></td></tr>\n";
                }
                //bind the data to html table
```

```
                //you can use jQuerytoo....
                document.getElementById('list').innerHTML = list;
        }else {
                alert('Cannot save shopping listas your browserdoes
not support HTML 5');
        }
}

function CheckBrowser() {
        if('localStorage' in window&& window['localStorage'] !==
null) {
                // we can uselocalStorage object to store data
                return true;
        }else {
                        return false;
        }
}
</script>
<style>
td,th {
        font-family: monospace;
        padding: 4px;
        background-color: #ccc;
}
label {
        vertical-align: top;
}
```

```
#main {
        border: 1px dotted blue;
        padding: 6px;
        background-color: #ccc;
        margin-right: 50%;
}
#items_table {
        border: 1px dotted blue;
        padding: 6px;
        margin-top: 12px;
        margin-right: 50%;
            background-color: #ccc;
}
#items_table h3 {
        font-size: 18px;
        margin-top: 0px;
        font-family: sans-serif;
}
body{
}
h1{
}
a{
}
.btn {
    background-color: #04AA6D;
    color: white;
```

```
    margin: 10px 0;
    border: none;
    width: 100%;
    border-radius: 3px;
    cursor: pointer;
    font-size: 17px;
        text-decoration: none;
    display: inline-block;
}
  c{
float:right;
margin-right:50px;
}
form{



}
.mainform{

    width:800px;
    margin:0 auto;
}
</style>
<script>
const crudform=(e)=>{
          e.preventDefault();
          alert(e.forms);
```

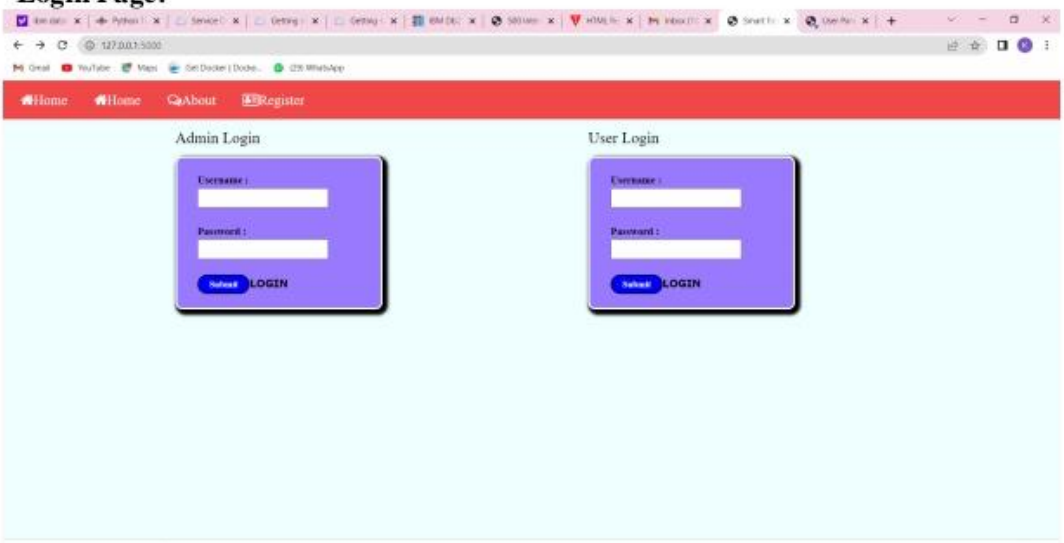
```
console.log(e);
}
</script>
</head>
<link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css">
<script src='https://kit.fontawesome.com/a076d05399.js'
crossorigin='anonymous'></script>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<body onload="doShowAll()">
<h1><center>Admin pannel</center></h1>
<a href='/' class='topnav-icons fa fa-sign-out   w3-bar-item w3-
button'style=float:right color:white>Back</a>
<div class="mainform">

<form name=ShoppingList
action="https://127.0.0.1:5000/Admin/CrudProduct" method="POST"
onsubmit="crudform()">
                <div id="main">
                        <table >
                                <tr>
                                        <td><b>Item:</b><input type=text
name=name></td>
                                        <td><b>Quantity:</b><input type=text
name=data></td>
                                </tr>
                                <tr>
```

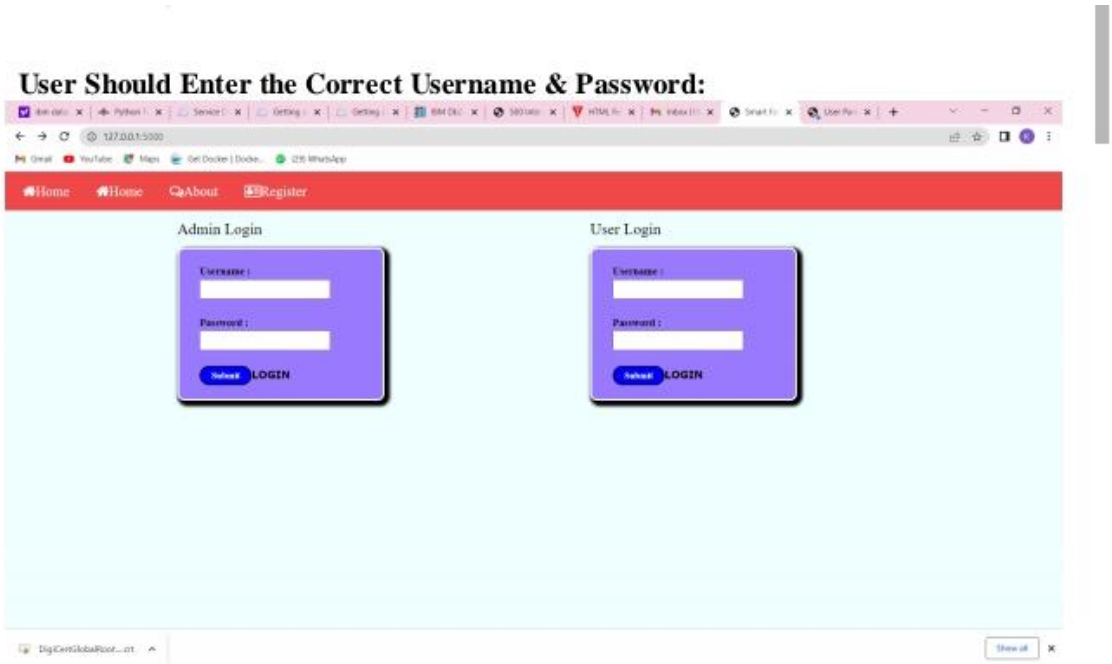
```
                                        <td>
                                            <input type="submit" value="Save"
name="save"    >
                                            <input type="submit" value="Update"
name="modify" onclick="ModifyItem()">
                                            <input type="submit" value="Delete"
name="remove" onclick="RemoveItem()">
                                        </td>
                                    </tr>
                                </table>
                            </div>
                            <div id="items_table">
                                <h3>Shopping List</h3>
                                <table id=list></table>
                                <p>
                                    <label><input type=button value="Clear"
onclick="ClearAll()">
                                </p>
                            </div>
                        </form>
                    </div>
                    </body>
                    </html>
```

Login Page:

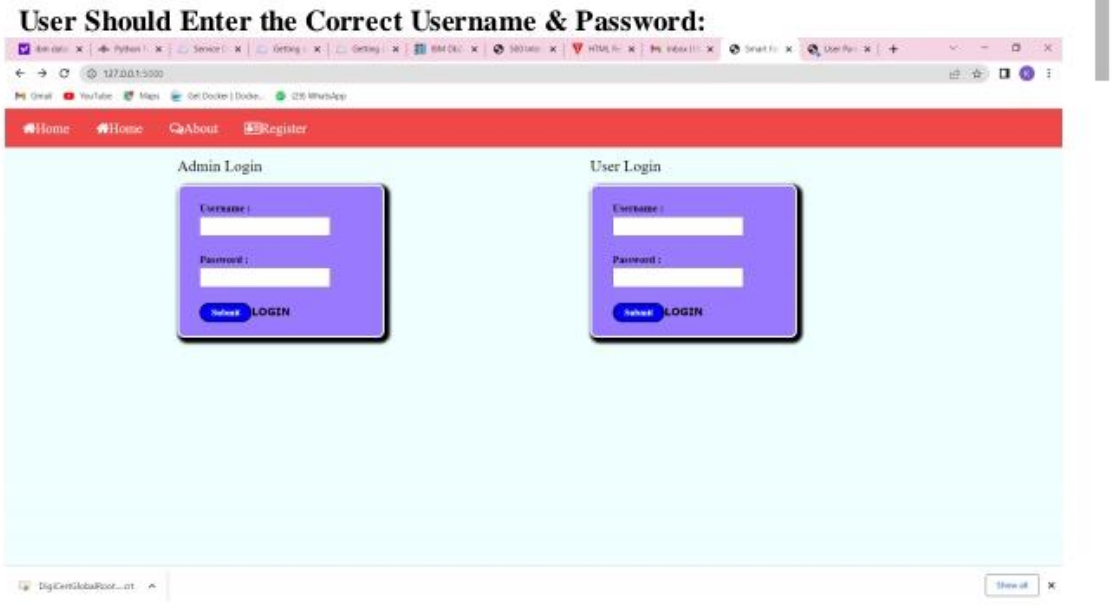
Login Page:



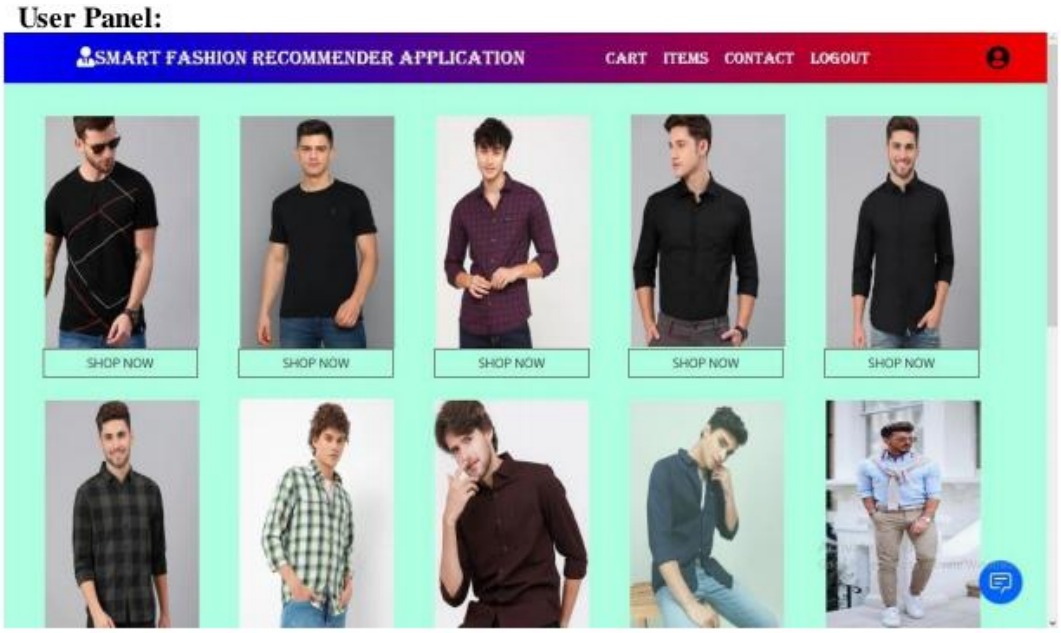
User Should Enter the CorrectUsername & Password:



Enterthe Username&Password:

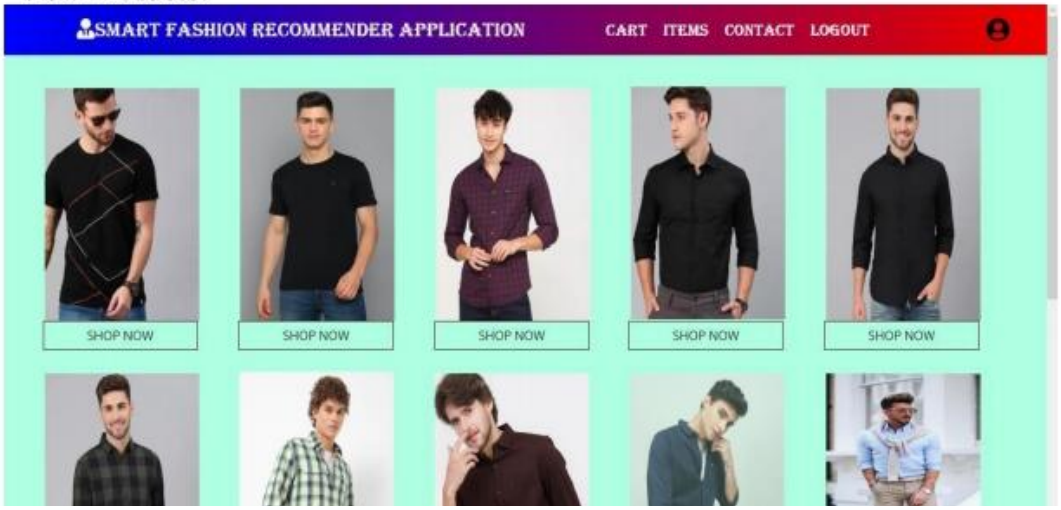


User Panel:



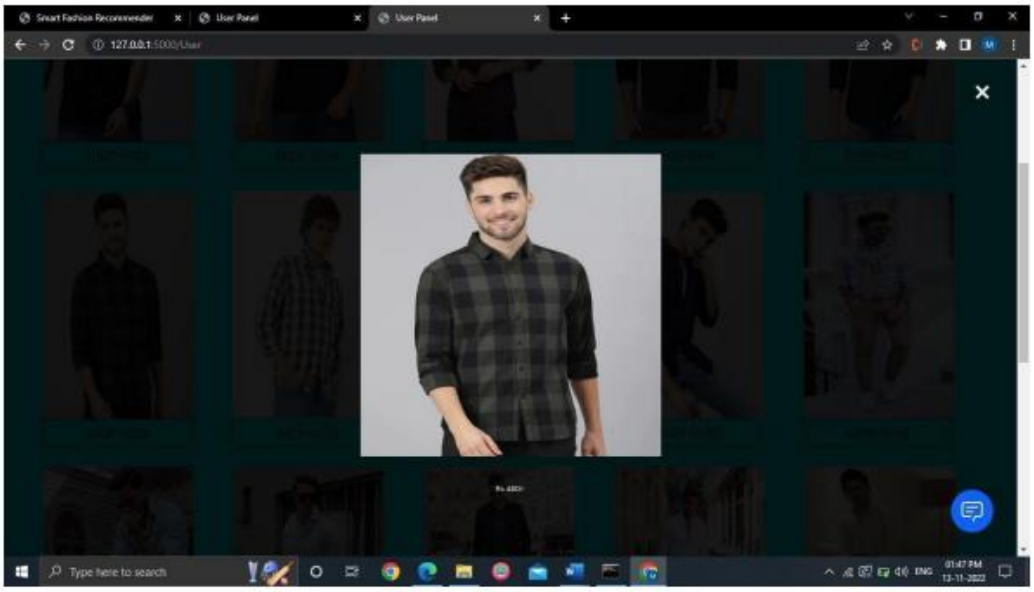
View Products:

View Products:



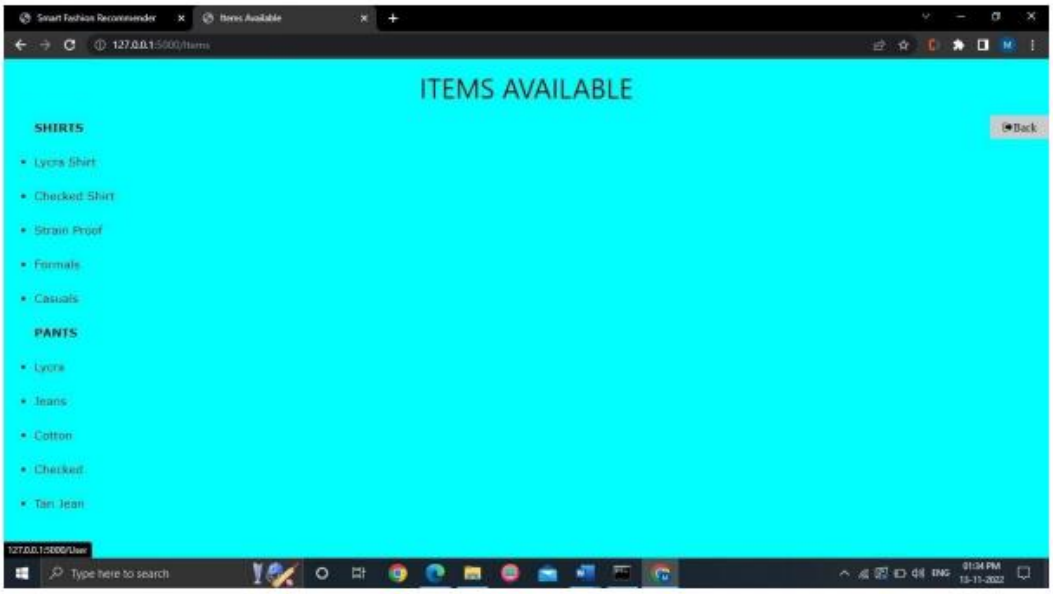
View Price:

View Price:



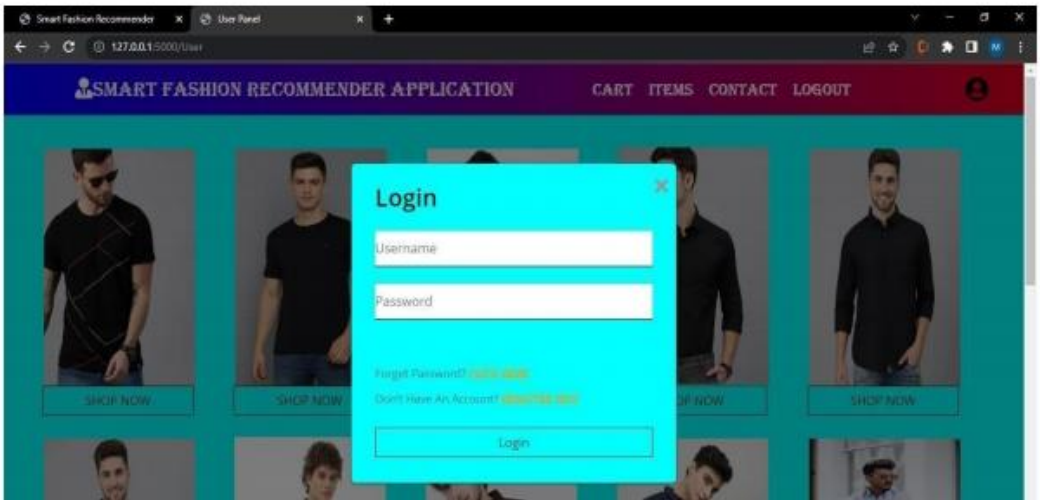
Items Available:

Items Available:



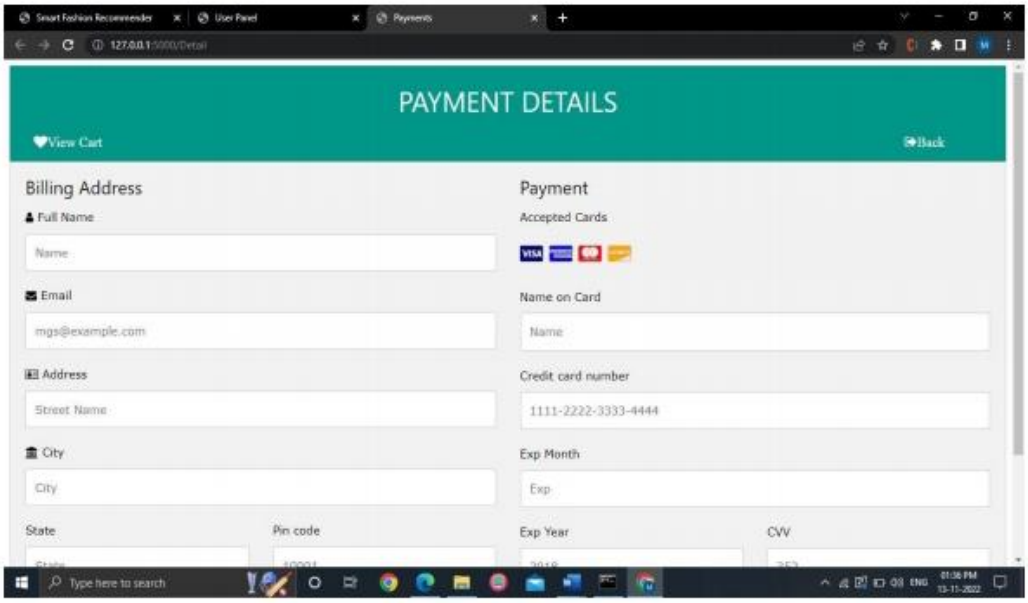
Login:

Login:



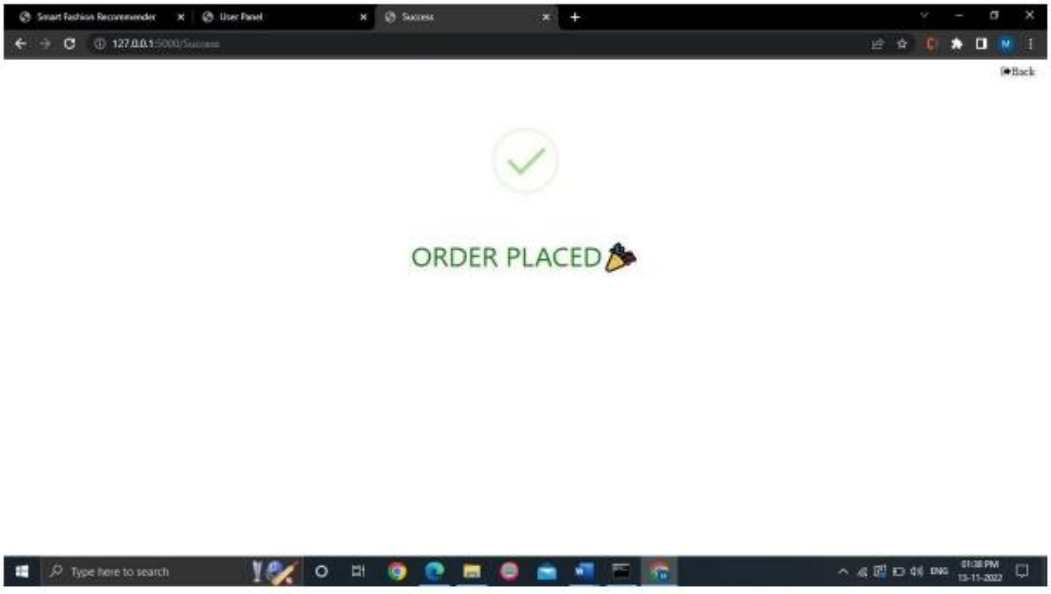
Payment Details:

Payment Details:



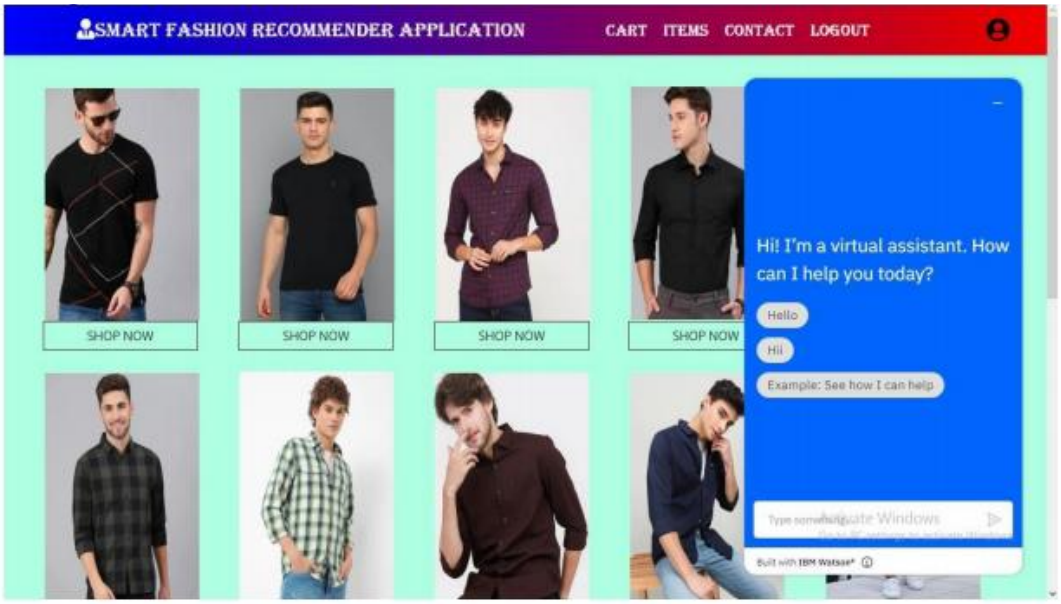
Order Placed:

Order Placed:



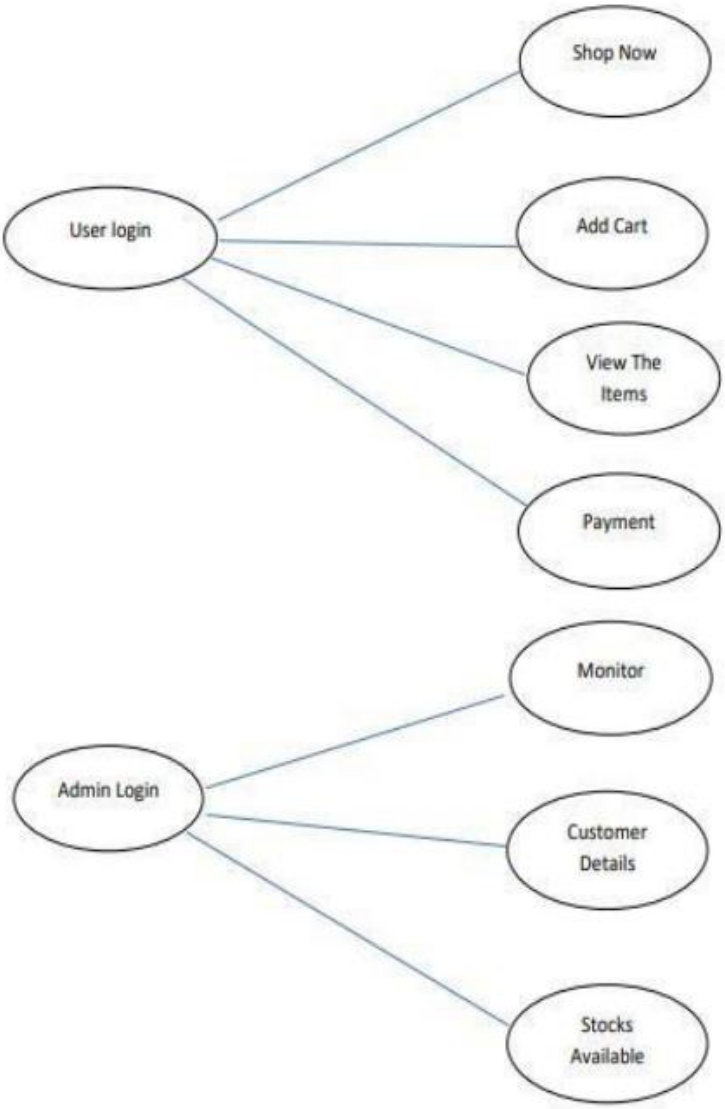
```
<script>

 window.watsonAssistantChatOptions = {
      integrationID: "9e1f0630-1095-4d2f-9215-c1750a168ddf", //
 TheID of thisintegration.
      region: "jp-tok", // The region your integration is hosted in.
      serviceInstanceID:"cd93e7bc-8deb-4aec-aa9e-5247a38c7653", //
 The ID of your service instance.
      onLoad: function(instance) { instance.render(); }
    };
    setTimeout(function(){
      const
      t=document.createElement('s
      cript'); t.src="https://web-
 chat.global.assistant.watson.appdomain.cloud/versions/" +
 (window.watsonAssistantChatOptions.clientVersion || 'latest') +
 "/WatsonAssistantChatEntry.js";
      document.head.appendChild(t);
    });
</script>
```


SMART FASHION RECOMMENDER APPLICATION
CART
ITEMS
CONTACT
LOGOUT
SHOP NOW
SHOP NOW
SHOP NOW
SHOP NOW
Hi! I'm a virtual assistant. How can I help you today?
Hello
Hii
Example: See how I can help
Built with IBM Watson®

Use Case

Use Case


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

3. **TESTING**

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 ReportOutput	5	0	0	5
Version Control	3	0	1	2

User Acceptance Testing

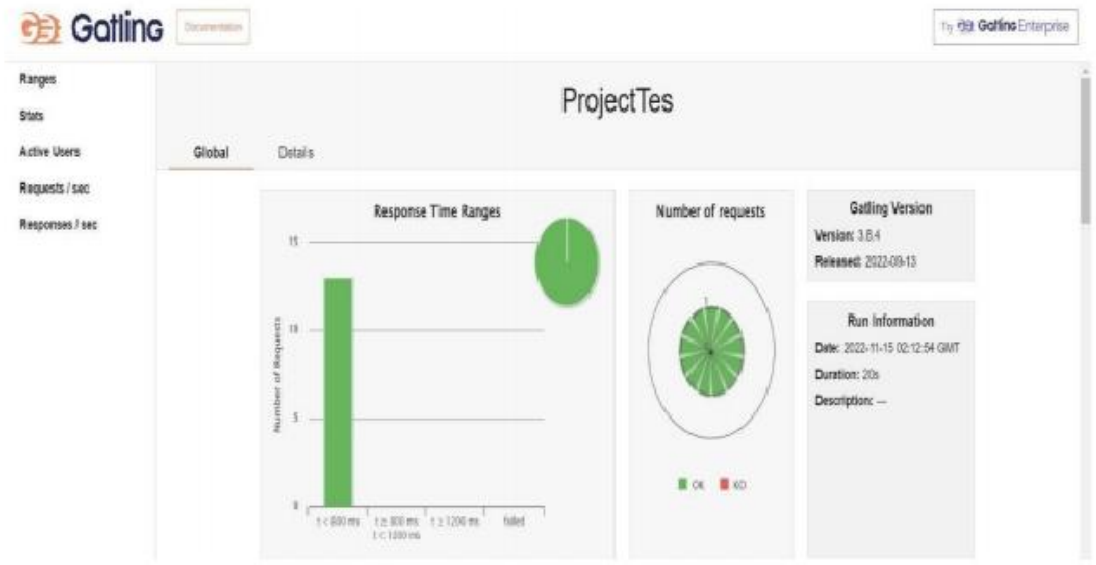
This report shows the number of resolved or closed bugs at each severitylevel,and how they were resolved.

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
NotReproduced	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

Performance Testing

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



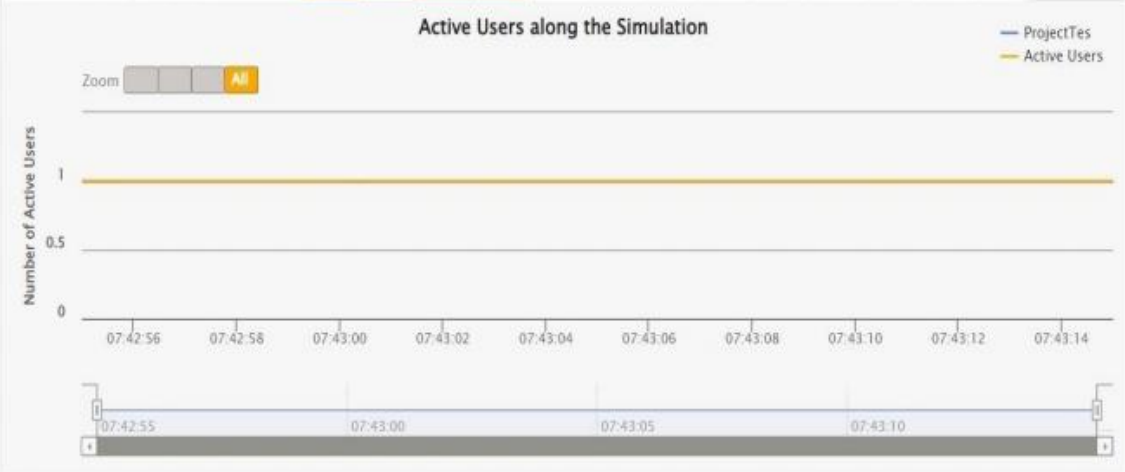
4. RESULTS

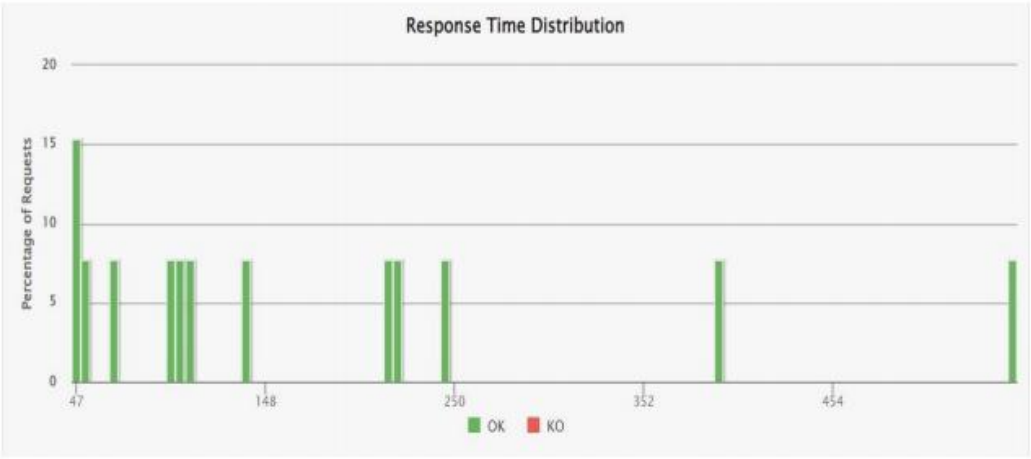
Performance Metrics

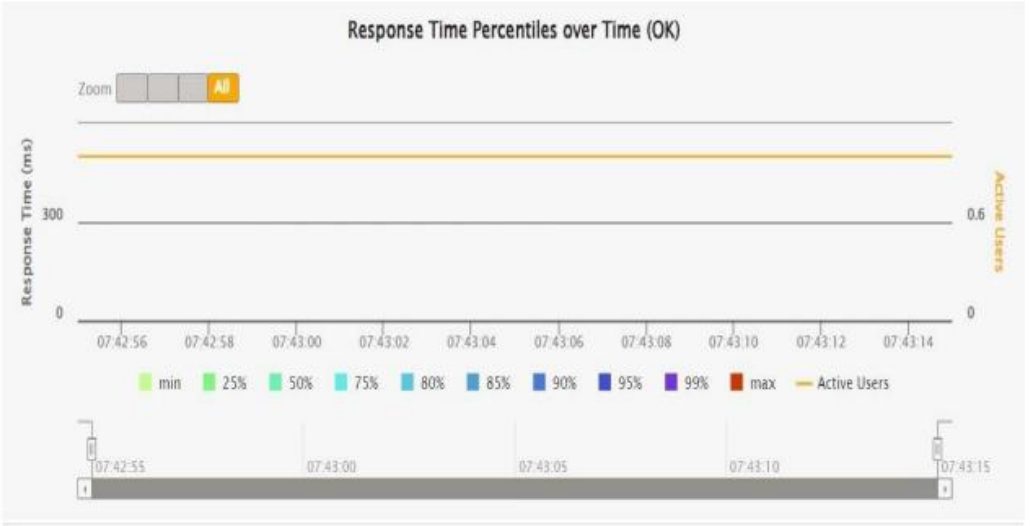
Performance Metrics

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
Active Users
0.6
0
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
Active Users
0.6
0
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
07:43:00
07:43:05
07:43:10
07:43:15

5. ADVANTAGES & DISADVANTAGES

Advantages

1. The key advantage is that it can bettercapture the dependencies amongitems.
2. It is useful for identifying recommendations that are as objective aspossible
3. It is used to model the bias of user.
4. Based on factors that include colour,colour pattern or clothing shapescan be personalised.

Disadvantages

5. Inability to capturechanges in user behavior.
6. Lack of data analytics capability.
7. It provides the user with too many choices.
8. Inaccurate estimations of customer's preferences.

6. CONCLUSION

We have identified a trend towards the use of chat-bots as co-creators ofvalue during purchase through different roles

specific to fashion. Thus it helpsto achieve next level satisfaction of the customers for whatever product they opt for and also fulfill the necessaries of the vendors, market analyst and othersuch actors.

7. 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 linkedto popularity forecasting in the fashiondomain since trendyitems will likelybe popular. Beyond these, fashion recommender systems are beginningto intersect with related domains including conversational models, models involving text, and augmented reality in future.

8. APPENDIX

```
from flask importFlask, render_template
app =
Flask(_
name_)
@app.rout
e("/")
```

```
def  homepage():
    return render_template("index.html",title="Home Page")
@app.route("/Admin")
def Admin():
    return render_template("Admin.html",title="LOGIN")
@app.route("/User")
def User():
    return render_template("User.html",title="LOGIN")
@app.route("/Detail")
def Detail():
    return render_template("Detail.html",title="Detail")
@app.route("/Contact1")
def Contact1():
    return render_template("Contact1.html",title="Contact1")
@app.route("/Contact")
def Contact():
    return render_template("Contact.html",title="Contact")
@app.route("/Cart")
def Cart():
    return render_template("Cart.html",title="Cart")
@app.route("/Success")

def Success():
    return render_template("Success.html",title="Success")
```

```
@app.route("/Items")
def Items():
    return render_template("Items.html",title="Items")
@app.route("/About")
def About():
    return
render_template("About.html",title="About
")if__name__=="___main__":
    app.run(debug=True)
```

dockerfile

```
. /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 flaskimport *
from flask_cors import CORS
import ibm_db
import re
app = Flask(__name__)
CORS(app)
cors =CORS(app, resources={
        r"/*":{
                "origins":"*"
        }
})
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=0c77d6f2-5da9-
48a9-81f8-
86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT
=31198;SECURITY=SSL;SSLServiceCertificate=DigiCertGlobalRootCA.c
rt;UID=kjf40222;PWD=mbSqNCJVKq3Ecbgi",",")
@app.route("/")
def hello():
```

```
        return render_template('index.html')
@app.route('/login',methods = ['POST'])
def login():
            uname=request.form['username']
            passwrd=request.form['password']
            sql= "SELECT * FROM usersWHERE USERNAME =? AND
PASSWORD=?"
            try:
                  stmt = ibm_db.prepare(conn, sql)
                  ibm_db.bind_param(stmt,1,uname)
                  ibm_db.bind_param(stmt,2,passwrd)
                  ibm_db.execute(stmt)
                  account = ibm_db.fetch_assoc(stmt)
                  print (account)
                  ifaccount["USERNAME"] == unameand
account["PASSWORD"] == passwrd:

                        msg = 'Logged in successfully !'
                        print(msg)
                        return redirect(url_for("user"))
                  else:
                              msg = 'Incorrect username / password !'
                     print(msg)
                     return render_template('index.html')
            except:

                  print("Error")
```

```
                return redirect(url_for("success"))
        else:
                print("complete")

@app.route('/register', methods =['GET', 'POST'])
def register():
    msg=" "
    ifrequest.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        try:
            sql= "SELECT * FROM usersWHERE 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)
            ifaccount:
                    msg = 'Account already exists!'
                    print(msg)
                    return redirect(url_for("reg"))
            else:
                    print("running")
                    print(username)
                    insert_sql = "INSERT INTOusers VALUES (?,
?)"
                    prep_stmt = ibm_db.prepare(conn,insert_sql)
```

```
                    ibm_db.bind_param(prep_stmt, 1, username)
                    ibm_db.bind_param(prep_stmt, 2, password)
                      ibm_db.execute(prep_stmt)
                    msg = 'You have successfully registered !'
                    print(msg)
                    return redirect(url_for("user"))
            except:
                    print("error")
                    return redirect(url_for("reg"))

    return render_template('register.html')
@app.route('/reg')
def reg():
    return render_template('register.html')
@app.route('/about')
def about():

        return render_template('About.html')
@app.route('/Admin')
def admin():
        return render_template('Admin.html')
@app.route('/cart')
def cart():
            return render_template('Cart.html')
@app.route('/contact')
def contact():
                return render_template('Contact.html')
@app.route('/detail')
def detail():
```

```
        return render_template('Detail.html')
@app.route('/items')
def items():
        return render_template('Items.html')
@app.route('/success')
def success():
                return render_template('Success.html')
@app.route('/user')
def user():
                return render_template('User.html')
if__name__== "__main__":
        app.run()
```