SMART FASHION RECOMMENDER APPLICATION

A PROJECT REPORT
Submitted by

AMBIKA J - 953419104007
ANUSIYA J - 953419104012
BAMA P - 953419104017
GIFTY ANNIE EVANGELINE P - 953419104022

in partial fulfillment for the award of degree of
Bachelor of Engineering (B.E.) in
COMPUTER SCIENCE AND ENGINEERING

VV COLLEGE OF ENGINEERING
ANNA UNIVERSITY NOVEMBER 2022

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our **Faculty Mentor** and **Industry Mentor** for their support and guidance in completing our project on the Smart Fashion Recommender Application.

We would like to extend our gratitude to the **IBM** for **Nalaiya Thiran** project for providing us with all the facility that was required.

It was a great learning experience.We would like to take this opportunity to express our gratitude.

DATE
18.11.2022

TEAM MEMBERS
AMBIKA J
ANUSIYA J
BAMA P
GIFTY ANNIE EVANGELINE P

CHAPTER -1

1. INTRODUCTION

Recent technological advancements have enabled consumers to track current fashion trends around the globe, which influence their choices. The fashion choices of consumers depend on many factors, such as demographics, geographic location, individual preferences, interpersonal influences, age, gender, season, and culture. Moreover, previous fashion recommendation research shows that fashion preferences vary not only from country to country but also from city to city. The combination of fashion preferences and the abovementioned factors associated with clothing choices could transmit the image features for a better understanding of consumers' preferences. Therefore, analysing consumers' choices and recommendations is valuable to fashion designers and retailers. Additionally, consumers' clothing choices and product preference data have become available on the Internet in the form of text or opinions and images or pictures. Since these images contain information about people from all around the world, both online and offline fashion retailers are using these platforms to reach billions of users who are active on the Internet. Therefore, e-commerce has become the predominant channel for shopping in recent years. The ability of recommendation systems to provide personalized recommendations and respond quickly to the consumer's choices has contributed significantly to the expansion of e-commerce sales

1.1 PROJECT OVERVIEW

The main aim of this system to develop an e-commerce platforms, where numerous choices are available, an efficient recommendation system is required to sort, order, and efficiently convey relevant product content or information to users. Image-based fashion recommendation systems (FRSs) have attracted a huge amount of attention from fast fashion retailers as they provide a personalized shopping experience to consumers. In addition, this review also

explores various potential models that could be implemented to develop fashion recommendation systems in the future.

1.2 PURPOSE

- The fashion choices of consumers depend on many factors, such as demographics, geographic location, individual preferences, interpersonal influences, age, gender, season, and culture.
- The Fashion Recommendation System is mainly used to recommend the best possible outfit combinations to a user who has no fashion sense based on their wardrobe.
- It mat not always provide the best possible outfit to wear for an occasion as the system clothes present in the users wardrobe.
- As with other products such as electronics and books, fashion products were also recommended based on the user's previouspurchase history.

CHAPTER -2

2. LITERATURE SURVEY

Girshick et.al |11(2018) propose to recommend images by explicitly learning and exploiting part based similarity. Here the author proposed a novel approach of learning discriminative features from weakly supervised data by using visual attention Over the parts and a texture encoding network. It shows that the learned features surpass the state-of-the-art in retrieval task on the Deep Fashion dataset. Finally, the author used the proposed model to recommend fashion images having an explicit variation with respect to similarity of any of the parts.

Goel D et.al |21 (2015) online shopping systems are looking for a method that can recommend items according to the user preference. The author proposes a content-based clothing recommender system using deep neural networks. In content-based systems, product features are required for prediction of unobserved items ratings. ln here the author proposed a system by using a deep neural network, the cloth category is obtained and the need to manually extract the product features is eliminated by producing the required features With a large and useful volume. The main advantage of the author system is to specify gender as a feature in making Suggestions then shows the results to the user.

sinkaye F et.al |3| (2015) the author present an item-to-set metric learning framework that learns to compute the similarity between a set of historical fashion items of a user to a new fashion item. To Extract features from multi-modal street - view fashion items, the author propose an embedding module that performs multi-modality feature extraction and cross-modality gated fusion. To validate the effectiveness of the author approach, the author collects a real-world social media dataset.

Lieberman H and Lam F 4| (2021) purpose of the author is to develop a system

which outputs outfit images with partially modified outfit of the input image according to the user s preferred style. The author creates the user's original dataset to learn the user s preference in advance by asking the user to classify a group of images into 4 styles. So he conducted an evaluation experiment of our system to

confirm that our system reflects the user s individual preference. As a result of the evaluation-Experiment. it was confirmed that the same image was recommended as different styles for different users, and that the users also thought that the recommended style matched the current style of the user s classification of the style.

Ren S et.al |5| (2017) propose a collaborative fashion recommendation system called CFRS. The author proposed a new metric called trend score. Trend score show trendy a product 1s and calculated the rating provided by CFRS users. Finally the author used the algorithm to analyse the trend score and sorting the product of cache category from trendiest.

2.1 EXISTING PROBLEM

- Significant investment required.
- Too many choices
- The complex onboarding process
- Lack of data analytics capability
- The ‘cold start’ problem
- Inability to capture changes in user behaviour
- Privacy concerns

2.2 REFERENCE

[1] Girshick R et al, "Rich feature hierarchies for accurate object detection and semantic segmentation," in 2018 IEEE Conference on Computer Vision and Pattern Recognition, 2018. DOI: 10.1109/CVPR.2018.81.

[2] Gocl D, Chaudhury S and Ghosh H. "Recommendation of complementary garments using ontology", 2015 Fifth Nat. Conf. on Compute. Vision, Pattern Recognition. Image Process. and Graph. (NCVPRIPG). 2015.

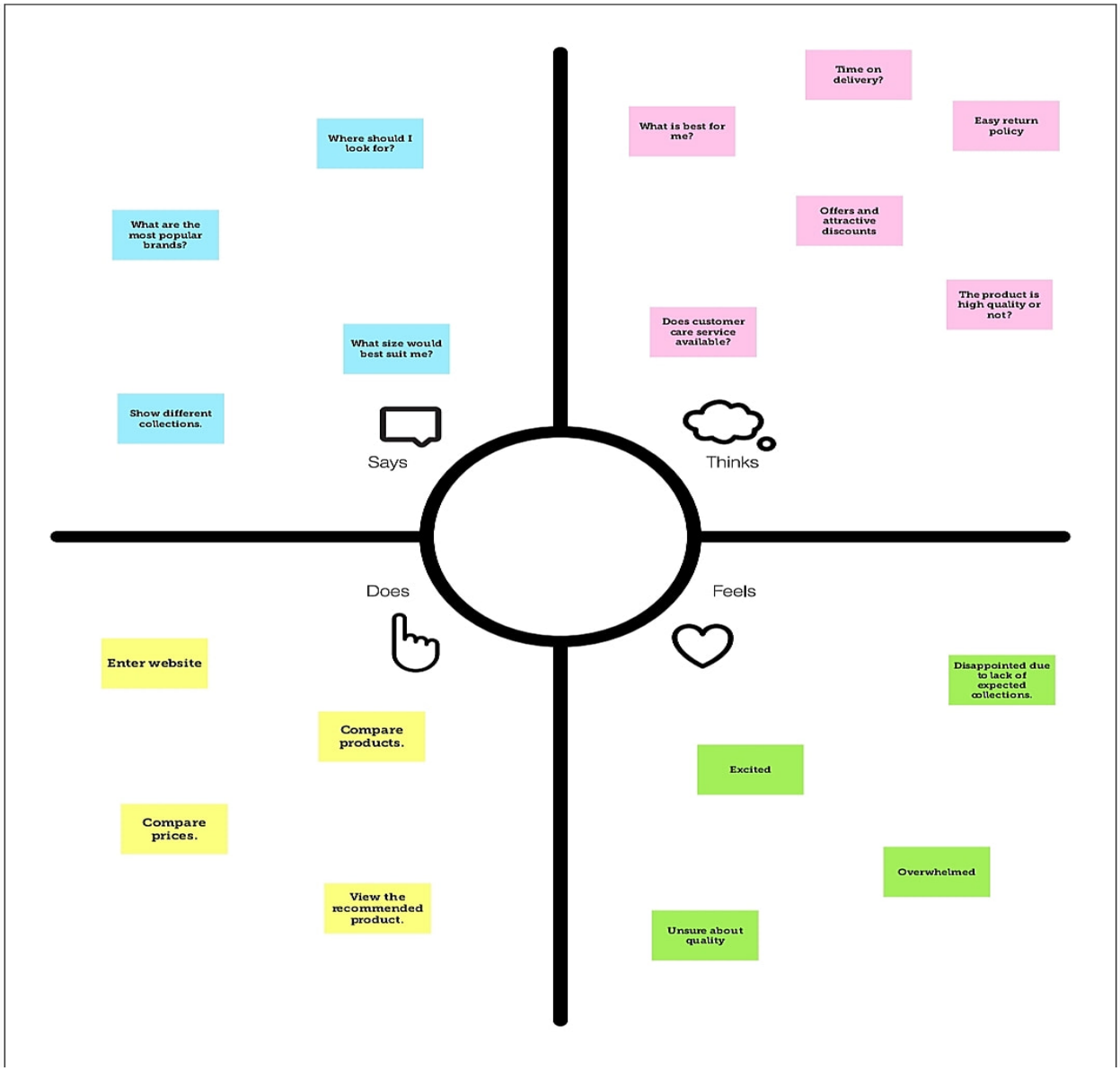
2.3 PROBLEM STATEMENT DEFINITION

Create a Smart Fashion Recommender System that provides personalized recommendation and respond quickly to the consumer through chatbot which improves consumers overall purchasing experience.

CHAPTER -3

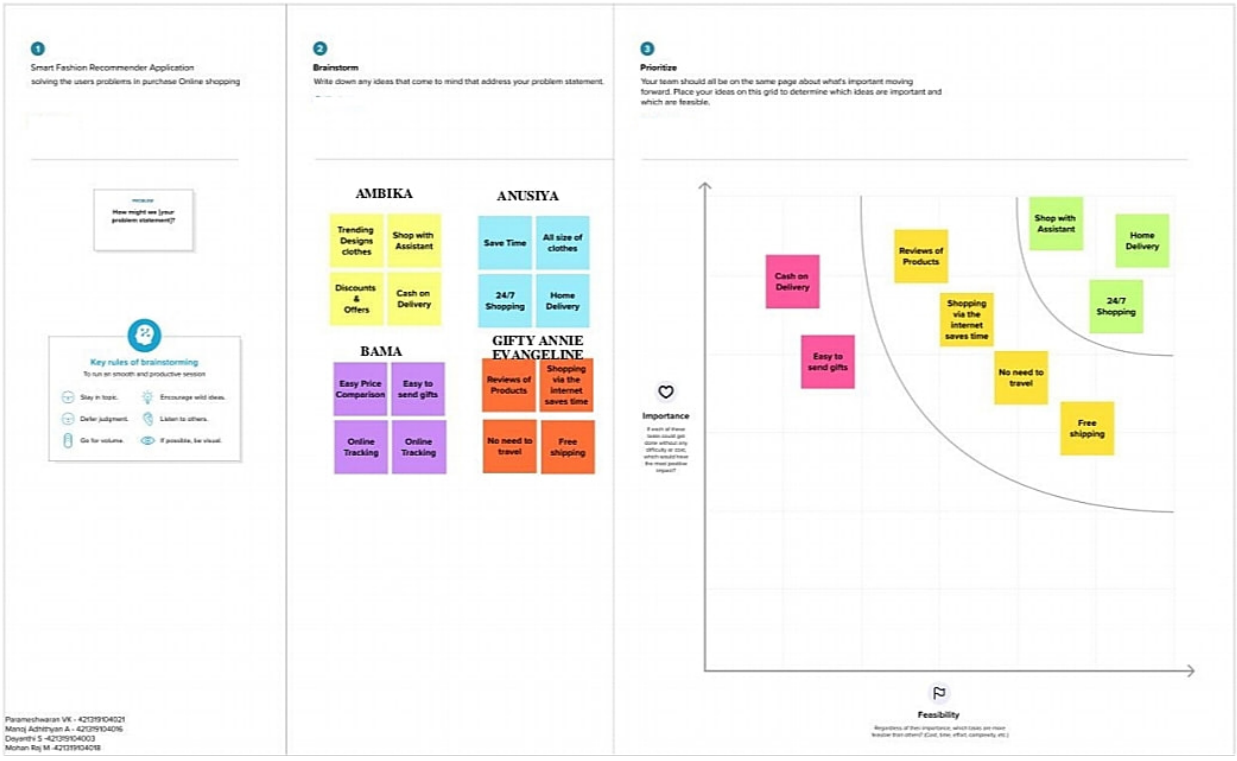
3. IDEATION & PROPOSED SYSTEM

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTROMING

Brainstorm & Idea Prioritization



3.3 PROPOSED SOLUTION

Project team shall fill the following information in proposed solution.

S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement(Problem to be solved)	• Navigating between various screens to make an online purchase. This one is the grumpy one. • Typically, e-commerce features include searching for a users product may take more time. • Search bar leads to the customer finding a product that is unrelated to what they were looking for.
2.	Idea/Solution Description	• You can directly do your online shopping based on your choice withoutany search. It can be done by using a chatbot. • User recommendations can be madeby the chatbot depending on their interests. • It may advertise the day's top specials and promotions. • It will keep a database of the Customer s information and orders. • If the order is accepted, the chatbot will notify the customers.

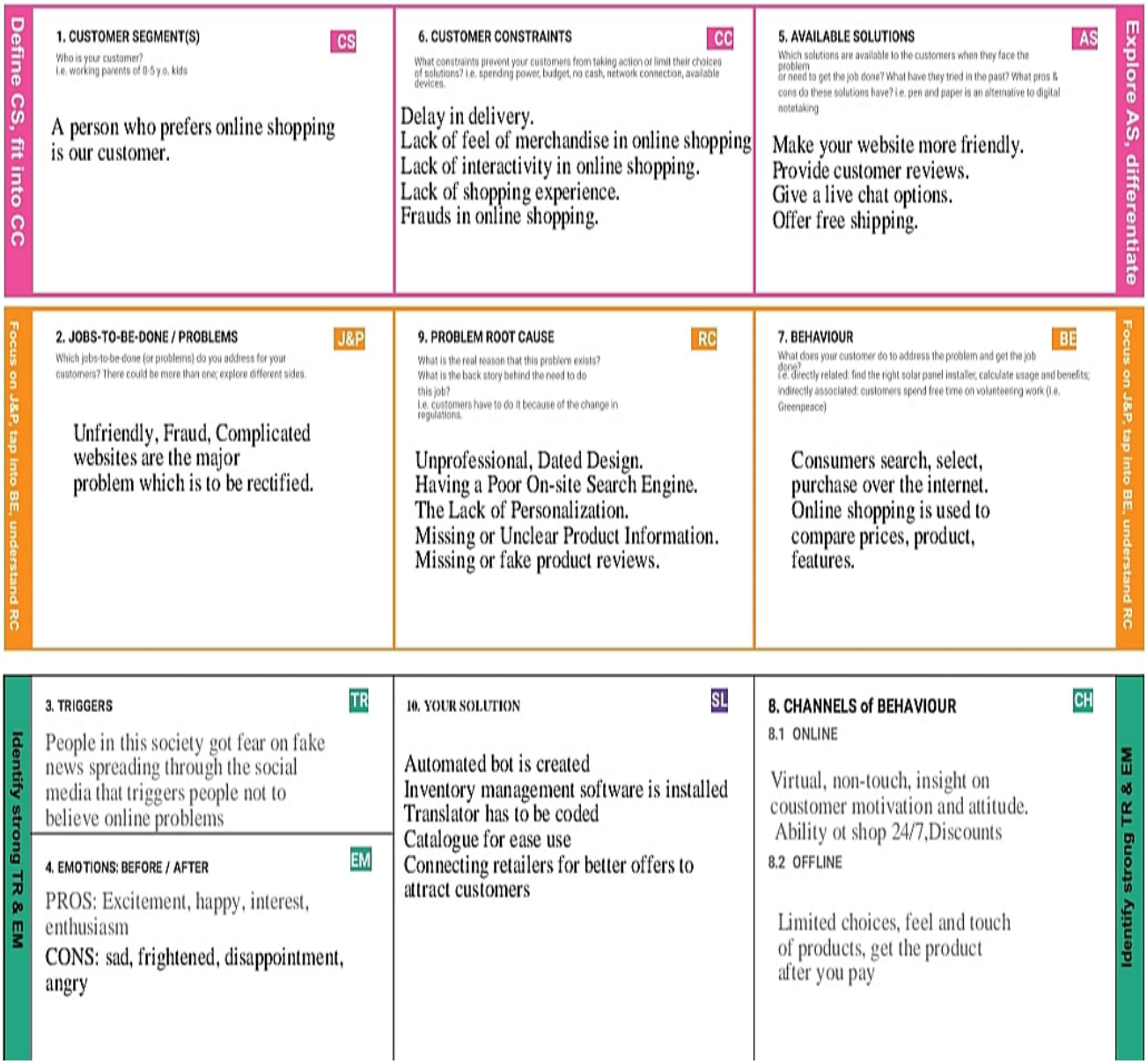
		• Chatbots can be useful for gathering Consumer reviews.
3.	Novelty/Uniqueness	• Chatbot is like talking to a person • Customerized product search • 24*7 support • Personalization • Reduced costs • Prioritize responsiveness
4.	Social Impact(Revenue Model)	• Increase sales and conversation • Personalize the customer experience. • Build brand awareness • Deal with customer queries. • Accurate and quick product search
5.	Business Model(Revenue Model)	• Growth opportunities • Fits into the pocket • Economical Development • Uncomplicated interface
6.	Scalability of the Solution	• Improved customer engagement • Drive sales

3.4 PROBLEM SOLUTION FIT

Project Title: Smart Fashion Recommender Application

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID50854



CHAPTER -4

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

- User Registration

Registration through Form

Registration through Gmail

Registration through mobile number

- User Confirmation

Confirmation via Email

Confirmation via Mobile number

- Advanced Search

Capabilities sorting and filtering options

- Checking item availability

item availability in specific locations

- Super-fast checkout

Online transfer,

credit card payment,

paying with mobile wallets

- Checking the shipping status

Option to easily check the shipping status of items order in the store

4.2 NON-FUNCTIONAL REQUIREMENTS

Following are the non-functional requirements of the proposed solution.

- Usability

Specific user in a specific context can use a product/design to achieve a defined goal effectively, efficiently and satisfactorily.

- Security

This Application will collect a lot of users' private information to complete a purchase(banking, shipping/home address, email, etc) Data protection is the priority.

- Reliability

Ability of the software to perform critical tasks like collecting and securing customer data, providing payment gateway to function correctly in a given environment,for a particular amount of time.

- Performance:

Online shopping behaviour is no different from offline people love places and platform that help them to find the best deals and product in a single place within a minimal effort.

- Availability

Online consumers do not adhere to closing times.
Information should be available wherever and whenever required within a time limit specified.

- Scalability

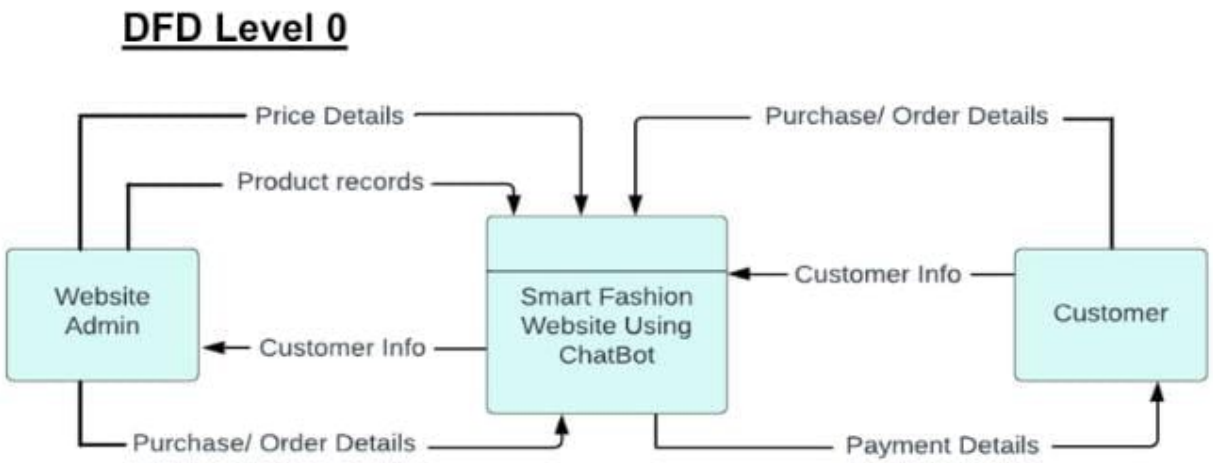
Having a plan to handle demand peaks Avoid downtime, preserve the customer experience, and ensure deliveries go out on time at all costs.

CHAPTER-5

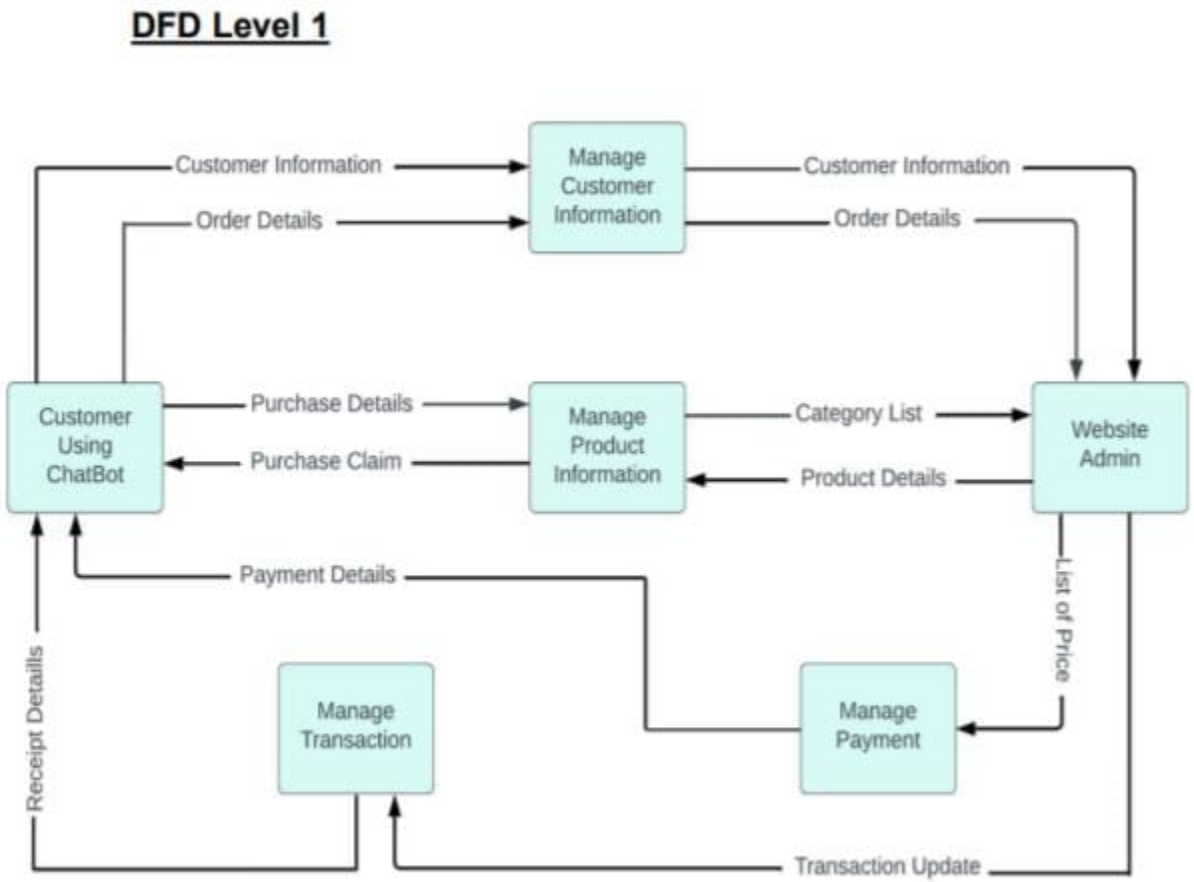
5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

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



- Admin is maintaining all the things that the users are purchasing
- To keep track of the stock information
- The chatbot can give payment details to the users.
- We can manage user selections and orders using chatbots.



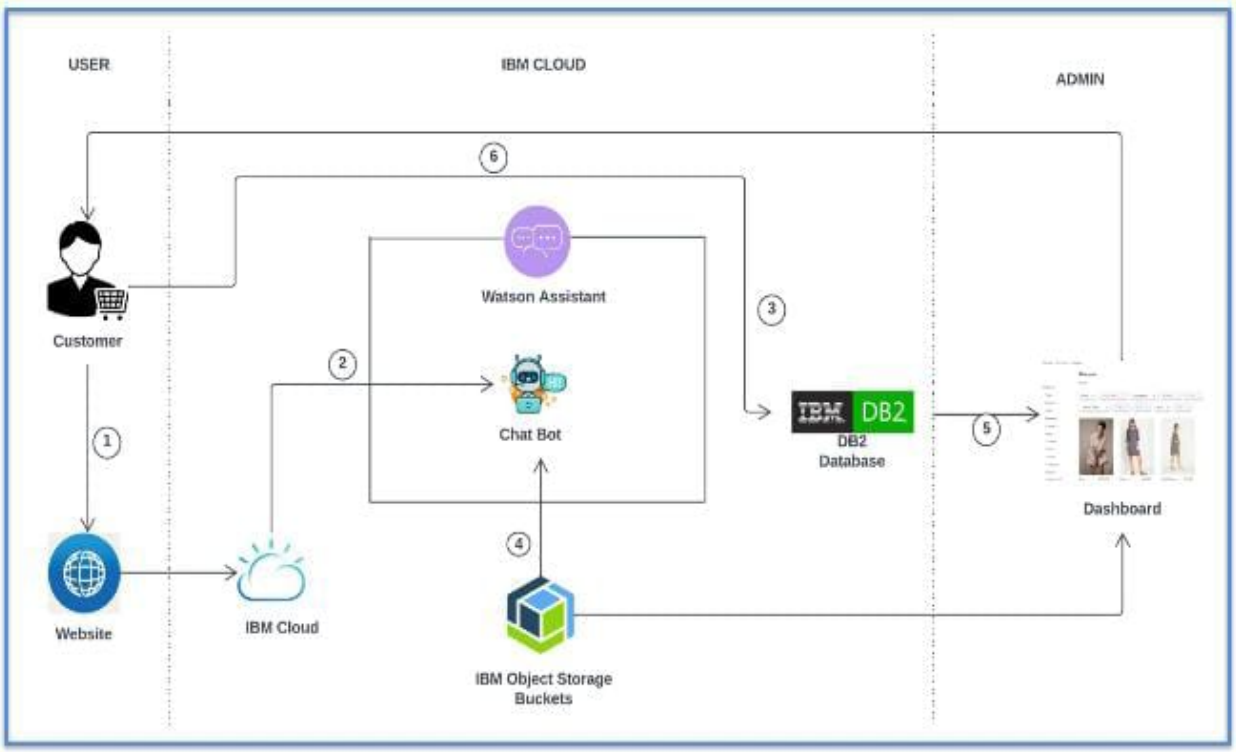
5.2 SOLUTION & TECHNICAL ARCHITECHTURE

COMPONENT	DESCRIPTION	TECHNOLOGY
Website	Website Customer can proceed the website and interact with the chatbot to get the desire product	HTML, CSS, JavaScript, Watson chatbot

Docker	Service for storing the private container images	Container
IBM Object Storage	Bucket are used to upload the images and files	Bucket
Kubernetes	Manage the complete process in the stable state If any software crash it automatically restart the work	Kubernetes
DB2	Data types are String, Numeric, Date, time, and timestamp distinct types. Act_ sortmem_ limit, auto_ del_ rec _ obj, auto_ maint Configuration .	MySQL
Cloud DB2	A fully managed cloud database with AI capabilities that keep our website running 24*7.	IBM DB2
Watson chatbot	Customers can search the product easily by human- like interaction with bot.	IBM Watson Assistant
Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Anaconda	Local, Cloud Foundry, Kubernetes, etc.

	Cloud Sever Configuration: IBM cloud	

TECHNICAL ARCHITECTURE DIAGRAM



5.3 USER STORIES

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
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
	Login	USN-3	As a user, I can login for the application by entering email & password	I can login for the application by entering email & password	High	Sprint-1
	Dashboard	USN-4	Browse the products that are offered on the website. The consumer can speak with chatbot directly about the products rather of having to navigate through numerous menus to make an online purchase	I can purchase product using chatbot instead of searching		Sprint-1
	Chatbot	USN-5	Using chatbot we can manage user's choices and orders		High	Sprint-2
		USN-6	The chatbot can give recommendations to the users based on their interests.	I can view related products of my interests	High	Sprint-3
		USN-7	It can promote the best deals and offers on that day.	I can view offers and prizes	High	Sprint-2
		USN-8	It will store the customer's details and orders in the database.		High	Sprint-3
		USN-9	The chatbot will send a notification to customers if the order is confirmed.	I get confirmation message	High	Sprint-2
	The flow of orders and checkout	USN-10	Order statuses are displayed on the website: confirmed, processing, shipped, returned.	I can view my status in each step		
Admin	Maintaining	USN-1	They maintain a record of everything that users are purchasing.		High	Sprint-2
		Usn-2	The administrator's job is to search the stock database.			Sprint-2

CHAPTER - 6

6. PROJECT PLANNING & SHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Remaining tasks (Milestones & Activities) to be completed

Milestones	Activities	Description
Project Development Phase	Delivery of Sprint – 1,2,3,4	To develop the code and submit the developed code by testing it
Setting up App environment	Create IBM Cloud account	Signup for an IBM Cloud account
	Create flask project	Getting started with Flask to create project
	Install IBM Cloud CLI	Install IBM Command Line Interface
	Docker CLI Installation	Installing Docker CLI on laptop
	Create an account in sendgrid	Create an account in sendgrid. Use the service as email integration to our application for sending emails
Implementing web Application	Create UI to interact with Application	Create UI • Registration page • Login page • View products page • Add products page
	Create IBM DB2 & connect with python	Create IBM DB2 service in IBM Cloud and connect with python code with DB
Integrating sendgrid service	Sendgrid integration with python	To send emails form the application we need to integrate the Sendgrid service
Developing a chatbot	Building a chatbot and Integrate to application	Build the chatbot and Integrate it to the flask application
Deployment of App in IBM Cloud	Containerize the App	Create a docker image of your application and push it to the IBM container registry
	Upload image to IBM container registry	Upload the image to IBM container registry
	Deploy in kubernetes cluster	Once the image is uploaded to IBM Container registry deploy the image to IBM Kebernetes cluster

Finished tasks (Milestones & Activities)

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

6.2 SPRINT DELIVERY SHEDULE

Product Backlog, Sprint Schedule, Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story points	Priority	Team Members
Sprint-1	Setting up App environment	USN-1	As a user, I can register in ICTA Academy and create IBM cloud account.	2	High	4
Sprint-1		USN-2	As a user, I will create a flask project	1	Low	4
Sprint-1		USN-3	As a user, I will install IBM Cloud CLI	2	Medium	4
Sprint-2	Setting up App environment	USN-4	As a user, I can install Docker CLI	1	Low	4
Sprint-2		USN-5	As a user, I will Create an account in sendgrid	2	Medium	4
Sprint-3	Implementing web application	USN-6	As a user, I Create UI to interact with the application	1	High	4
Sprint-3		USN-7	As a user, I Create IBM DB2 and connect with Python	3	High	4
Sprint-3	Integrating sendgrid service	USN-8	As a user, I will integrating sendgrid with python code	2	High	4
Sprint-3	Developing a chatbot	USN-9	As a user, I have to build a chatbot and Integrate to application	1	Medium	4
Sprint-4	Development of App in IBM Cloud	USN-10	As a user, I will Containerize the App	1	Low	4
Sprint-4		USN-11	As a user, I will upload image to IBM Container registry	2	Medium	4
Sprint-4		USN-12	As a user, I will deploy App in Kebernetes cluster	3	High	4
Sprint-4	User panel		As a user • Register, Login, Email, Verification • Manual Search • Order placement, Order Details	3	High	4

Project Tracker, Velocity & Burndown Chart

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	18	6 Days	24 Oct 2022	29 Oct 2022	24	29 Oct 2022
Sprint-2	18	6 Days	31 Oct 2022	05 Nov 2022	24	05 Nov 2022
Sprint-3	18	6 Days	07 Nov 2022	12 Nov 2022	24	12 Nov 2022
Sprint-4	18	6 Days	14 Nov 2022	19 Nov 2022	24	19 Nov 2022

Velocity

Imagine we have a 6-day sprint duration, and the velocity of the team is 18(points per sprint). Let's calculate the team's average

velocity (AV) per iteration unit (story points per day)

AV = Sprint Duration / Velocity

AV = 24/6 = 4

Burndown Chart

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


Goal : 80 hours in 4 weeks
Burndown Chart
Effort(hours)
90
80
70
60
50
40
30
20
10
0
Setting up App Environment
Integrating sendgrid service
Developing a chatbot
Implementing web App
Deployment of app in IBM Cloud
Actual Effort
Estimated Effort
Oct 24 - 29 Week 1
Oct 31 – Nov 05 Week 2
Nov 07 – 12 Week 3
Nov 14 – 19 Week 4
Remaining (Days)

CHAPTER - 7

7.CODING & SHEDULING

7.1 FEATURE 1

login.html

```
<html>
<head>
<title> Fashion Vibe </title>
</head>
<style>
*{
margin: 0;
padding: 0;
font-family:"Century Gothic", CenturyGothic, AppleGothic, sans-serif;
}
.main{
width: 100%;
background:#DFF6FF;
background-position: center;
background-size: cover;
height: 100%;
```

```
font-family:"Century Gothic", CenturyGothic, AppleGothic, sans-serif;
}
.navbar{
width: 100%;
height: 75px;
margin: auto;
}
.icon{
width: 200px;
float: left;
height: 70px;
}
.logo{
    color:#25316D;
font-size: 35px;
padding-left: 20px;
float: left;
padding-top: 10px;
}
.menu{
```

```
width: 400px;
float: left;
height: 70px;
}
ul{
float: left;
display: flex;
justify-content: center;
align-items: center;
}
ul li{
list-style: none;
margin-left: 62px;
margin-top: 27px;
font-size: 14px;
}
ul li a{
text-decoration: none;
color:#0F3460;
font-weight: bold;
```

```
transition: 0.4s ease-in-out;
}
ul li a:hover{
color: white;
}
.search{
width: 330px;
float: left;
margin-left: 270px;
}
.srch{
width: 200px;
height: 40px;
background: transparent;
border: 1px solid black;
margin-top: 13px;
color: yellow;
border-right: none;
font-size: 16px;
float: left;
```

```
padding: 10px;
border-bottom-left-radius: 5px;
border-top-left-radius: 5px;
}
.btn{
width: 100px;
height: 40px;
background:#850E35;
border: 2px solid black;
margin-top: 13px;
color: #ffff;
font-size: 15px;
border-bottom-right-radius: 5px;
border-bottom-right-radius: 5px;
}
.btn:focus{
outline: none;
}
.srch:focus{
outline: none;
```

```
}
.content{
width: 1200px;
height: auto;
margin: auto;
color:#0F3460;
position: relative;
}
.content.par{
padding-left: 20px;
padding-bottom: 25px;
letter-spacing: 1.2px;
line-height: 30px;
}
.content h1{
font-size: 50px;
padding-left: 20px;

margin-top: 9%;
letter-spacing: 2px;
```

```
}
.content .cn{
width: 160px;
height: 40px;
background: greenyellow;
border: none;
margin-bottom: 10px;
margin-left: 20px;
font-size: 18px;
border-radius: 10px;
cursor: pointer;
transition: .4s ease;
}
.content .cn a{
text-decoration: none;
color: #000;
transition: .3s ease;
}
.cn:hover{
background-color: #fff;
```

```
}
.content span{
    color:#0F3460;
font-size: 60px;
}
.form{
width: 250px;
height: 380px;
background: linear-gradient(to top,rgba(0,0,0,0.8)50%,rgba(0,0,0,0.8)50%);
position: absolute;
top: -20px;
left: 870px;
border-radius: 10px;
padding: 25px;
}
.form h2{
width: 220px;
text-align: center;
color:yellowgreen;
font-size: 22px;
```

```
border-radius: 10px;
margin: 2px;
padding: 8px;
}
.form input{
width: 240px;
height: 35px;
background: transparent;
}
.form input{
width: 240px;
height: 35px;
background: transparent;
border-bottom: 1px solid yellowgreen;
border-top: none;
border-right: none;
border-left: none;
color: #fff;
font-size: 15px;
letter-spacing: 1px;
```

```
margin-top: 30px;
}
.form input:focus{
outline: none;
}
::placeholder{
color: #fff;
}
.btnn{
width: 240px;
height: 40px;
background:#850E35;
border: none;
margin-top: 30px;
font-size: 18px;
border-radius: 10px;
cursor: pointer;
color: #fff;
/* color:white; */
transition: 0.4s ease;
```

```
}
.btnn:hover{
background: #fff;
color: yellowgreen;
}
.btnn a{
text-decoration: none;
color: #000;
font-weight: bold;
}
.form .link{
font-size: 17px;
padding-top: 20px;
color: white;
text-align: center;

}
.form .link a{
text-decoration: none;
color: yellowgreen;
```

```
}
.liw{
padding-top: 15px;
padding-bottom: 10px;
text-align: center;
}
</style>
<body>
<div class="main">
<div class="navbar">
<div class="icon">
<h2 class="logo">Fashion Vibe</h2>
</div>
<div class="menu">
<ul>
<li><a href="#">HOME</a></li>
<li><a href="#">ABOUT</a></li>
</ul>
</div>
<div class="search">
```

```
<input class="srch" type="search" name="" placeholder="">
<a href="#"><button class="btn">Search</button></a>
</div>
</div>
<div class="content">
<h1>Smart Fashion <br><span>Application</span></h1>
<div class="form">
<h2>FASHION LOGIN</h2>
<form action="D:\Codes\Python\Homepage.html" method="post">
<input type="text" name="username" placeholder="Enter Username Here">
<input type="password" name="password" placeholder="Enter Password here">
<button type="submit" class="btnn"><a
href="FashionVibe.html">Login</a></button>
<p class="link">Don't have an account<br>
<a href="D:\Codes\Python\index.html">Sign up </a> here</a></p>
</form>
</div>
</div>
</div>
</body>
</html>
```

homepage.html

```
<html>
<head>
<title> Fashion Vibe </title>
</head>
<style>
*{
margin: 0;
padding: 0;
font-family:"Century Gothic", CenturyGothic, AppleGothic, sans-serif;
}
.main{
width: 100%;
background: #DFF6FF;
background-position: center;
background-size: cover;
height: 100%;
font-family:"Century Gothic", CenturyGothic, AppleGothic, sans-serif;
}
.navbar{
width: 100%;
height: 75px;
margin: auto;
}
.icon{
width: 200px;
float: left;
height: 70px;
```

```
}
.logo{
color:#25316D;
font-size: 35px;
padding-left: 20px;
float: left;
padding-top: 10px;
}
.menu{
width: 400px;
float: left;
height: 70px;
}
ul{
float: left;
display: flex;
justify-content: center;
align-items: center;
}
ul li{
list-style: none;
margin-left: 62px;
margin-top: 27px;
font-size: 14px;
}
ul li a{
text-decoration: none;
color:#0F3460;
font-weight: bold;
transition: 0.4s ease-in-out;
```

```
}
ul li a:hover{
color: rgb(98, 246, 152);
}
.search{
width: 330px;
float: left;
margin-left: 270px;
}
.srch{
width: 200px;
height: 40px;
background: transparent;
border: 1px solid black;
margin-top: 13px;
color: #ffff;
border-right: none;
font-size: 16px;
float: left;
padding: 10px;
border-bottom-left-radius: 5px;
border-top-left-radius: 5px;
}
.btn{
width: 100px;
height: 40px;
background:#850E35;
border: 2px solid black;
margin-top: 13px;
color: #ffff;
```

```
font-size: 15px;
border-bottom-right-radius: 5px;
border-bottom-right-radius: 5px;
}
.btn:focus{
outline: none;
}
.srch:focus{
outline: none;
}
.content{
width: 1200px;
height: auto;
margin: auto;
color: #ffff;
position: relative;
}
.content.par{
padding-left: 20px;
padding-bottom: 25px;
letter-spacing: 1.2px;
line-height: 30px;
}
.content h1{
font-size: 50px;
padding-left: 20px;
margin-top: 9%;
letter-spacing: 2px;
}
.content .cn{
```

```
width: 160px;
height: 40px;
background: rgb(98, 246, 152);
border: none;
margin-bottom: 10px;
margin-left: 20px;
font-size: 18px;
border-radius: 10px;
cursor: pointer;
transition: .4s ease;
}
.content .cn a{
text-decoration: none;
color: #000;
transition: .3s ease;
}
.cn:hover{
background-color: #fff;
}
.content span{
color:rgb(98, 246, 152);
font-size: 60px;
}
.form{
width: 250px;
height: 380px;
background: linear-gradient(to
top,rgba(0,0,0,0.8)50%,rgba(0,0,0,0.8)50%);
position: absolute;
top: -20px;
```

```
left: 870px;
border-radius: 10px;
padding: 25px;
}
.form h2{
width: 220px;
text-align: center;
color:rgb(98, 246, 152);
font-size: 22px;
border-radius: 10px;
margin: 2px;
padding: 8px;
}
.form input{
width: 240px;
height: 35px;
background: transparent;
}
.form input{
width: 240px;
height: 35px;
background: transparent;
border-bottom: 1px solid rgb(98, 246, 152);
border-top: none;
border-right: none;
border-left: none;
color: #fff;
font-size: 15px;
letter-spacing: 1px;
margin-top: 30px;
```

```
}
.form input:focus{
outline: none;
}
::placeholder{
color: #fff;
}
.btnn{
width: 240px;
height: 40px;
background: rgb(98, 246, 152);
border: none;
margin-top: 30px;
font-size: 18px;
border-radius: 10px;
cursor: pointer;
color: #fff;
transition: 0.4s ease;
}
.btnn:hover{
background: #fff;
color: rgb(98, 246, 152);
}
.btnn a{
text-decoration: none;
color: #000;
font-weight: bold;
}
.form .link{
font-size: 17px;
```

```
padding-top: 20px;
text-align: center;
}
.form .link a{
text-decoration: none;
color: rgb(98, 246, 152);
}
.liw{
padding-top: 15px;
padding-bottom: 10px;
text-align: center;
}
</style>
<body>
   <script>
       window.watsonAssistantChatOptions = {
         integrationID: "1a8c11c0-839e-4442-8b03-59f7c12ce5f5", // The
ID of this integration.
         region: "au-syd", // The region your integration is hosted in.
         serviceInstanceID: "bada3725-51e6-42fe-bccc-3e2603433478", //
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>
<div class="main">
<div class="navbar">
<div class="icon">
<h2 class="logo">Fashion Vibe</h2>
</div>
<div class="menu">
<ul>
<li><a href="#">HOME</a></li>
<li><a href="#">ABOUT</a></li>
</ul>
</div>
<div class="search">
<input class="srch" type="search" name="" placeholder="">
<a href="#"><button class="btn">Search</button></a>
</div>
</div>
</div>
</body>
</html>
```

7.2 FEATURE 2

chat.js

```
<script>
        let subMenu = document.getElementById("subMenu");
        function toggleMenu(){
            subMenu.classList.toggle("open-menu");
        }
```

```
        window.watsonAssistantChatOptions = {
            integrationID: "1a8c11c0-839e-4442-8b03-59f7c12ce5f5", // The
ID of this integration.
            region: "au-syd", // The region your integration is hosted in.
            serviceInstanceID: "bada3725-51e6-42fe-bccc-3e2603433478",
// 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>
```

server.py

```
from flask import Flask, render_template, request
import os


appFlask = Flask(_name_)
picFolder = os.path.join('static','images')
appFlask.config['UPLOAD_FOLDER'] = picFolder


@appFlask.route('/')
```

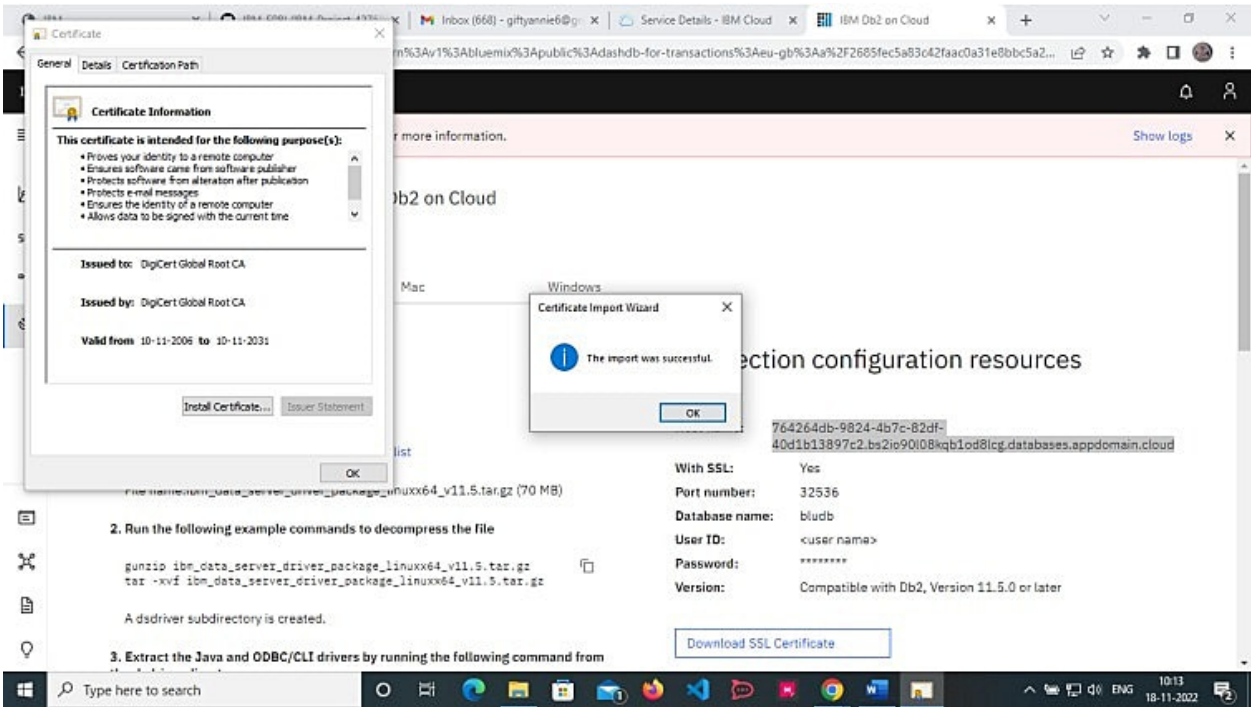
```
@appFlask.route('/out')
def index():
    return render_template("login.html")

@appFlask.route('/login',methods = ['POST', 'GET'])
def my_forum_post():
    return render_template('FashionVibe.html')

@appFlask.route('/index',methods = ['POST', 'GET'])
def my_forum_posts():
    return render_template('index.html')

@appFlask.route('/Feed',methods = ['POST', 'GET'])
def my_forum_posts1():
    return render_template('Feedback.html')



if _name_ == "_main_":
    appFlask.run(debug=True)
```

7.3 DATABASE SCHEMA

IBMDB2 WITH PYTHON:

```
import ibm_db
hostname=""
uid=""
pwd=""
driver="{IBM DB2 ODBC DRIVER}"
db="bludb"
port=""
```

```
protocol="TCPIP"
cert="Certificate.crt"
dsn=( "DATABASE={0};"
"HOSTNAME={1};"
"PORT={2};"
"UID={3};"
"SECURITY=SSL;"
"SSLServerCertificate={4};"
"PWD={5};"
).format(db,hostname,port,uid,cert,pwd)print(dsn)
try:
db2=ibm_db.connect(dsn,"","")
print("connected to data base")
except:
print("Unable to connect",ibm_db)
```



CHAPTER - 8

8.1 TEST CASES

SYSTEM TESTING

Testing is a set activity that can be planned and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it is vital success of the system.

Testing Objectives:

There are several rules that can serve as testing objectives, they are

a. Testing is a process of executing a program with the intent of finding an error
b. A good test case is one that has high probability of finding an undiscovered error.
c. A successful test is one that uncovers an undiscovered error.

If testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrates that software functions appear to the working according to the specification, that performance requirements appear to have been met.

There are three ways to test a program

- For Correctness
- For Implementation efficiency
- For Computational Complexity.

Testing Correctness

The following ideas should be a part of any testing plan:

- Preventive Measures
- Spot checks
- Testing all parts of the program
- Test Data

- Looking for trouble
- Time for testing
- Re Testing

UNIT TESTING

As this system was partially GUI based WINDOWS application, the following were tested in this phase

- Tab Order
- Reverse Tab Order
- Fie id length
- Front end validations

INTEGRATION TESTING

Test data should be prepared carefully since the data only determines the efficiency and accuracy of the system. Artificial data are prepared solely for testing. Every program validates the input data.

VALIDATION TESTING

In this, all the Code Modules were tested individually one after the other. In our case all the modules were combined and given the test data. The combined module works successfully without any side effect on other programs. Everything was found fine working.

OUTPUT TESTING

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or system testing. Black Box testing methods focus on the functional requirement of the software.

8.2 USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Smart Fashion Recommender Application project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	5	2	3	21
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

3. Test Case Analysis

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

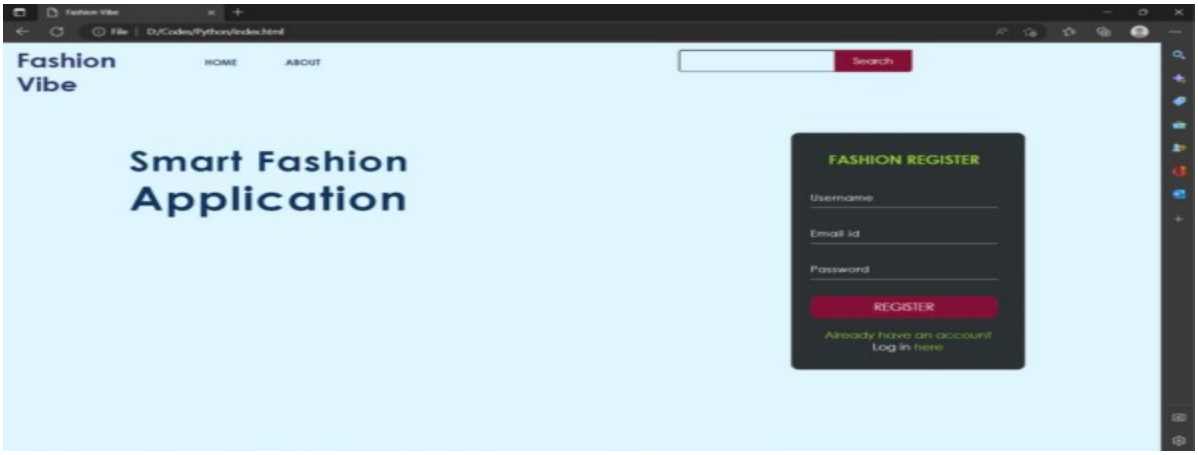
Section	Total Cases	Not Tested	Fail	Pass
Login	5	0	0	5
Register	7	0	0	7
Home Page	2	0	0	2
Order page	3	0	0	3
Order products	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

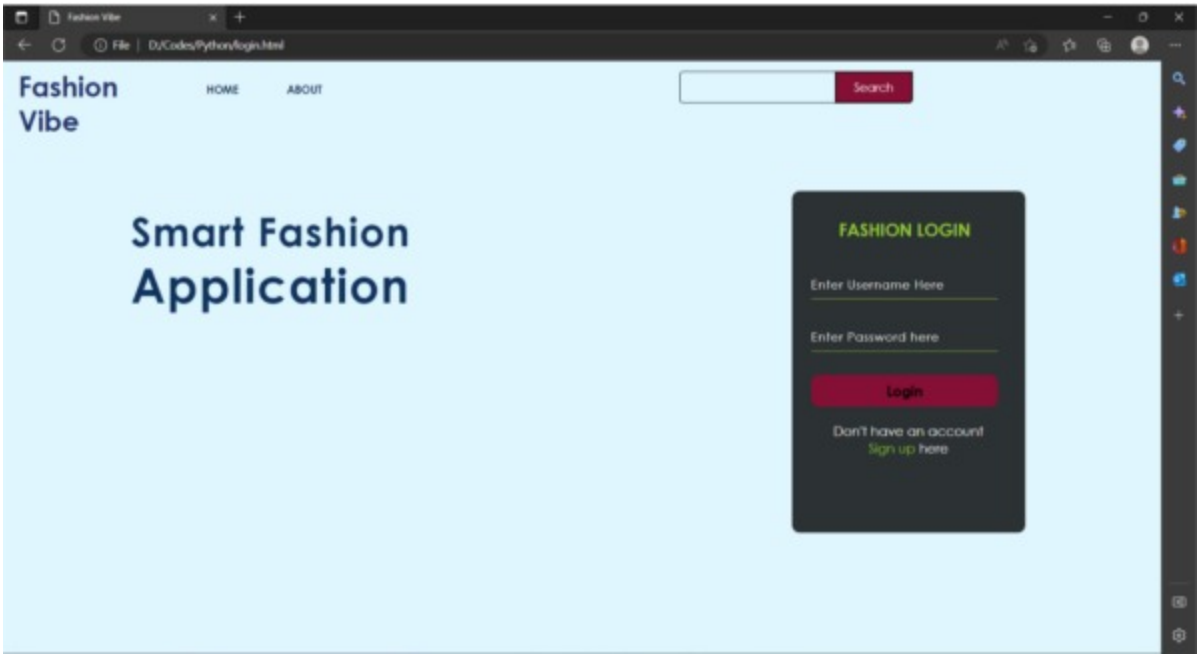
CHAPTER - 9

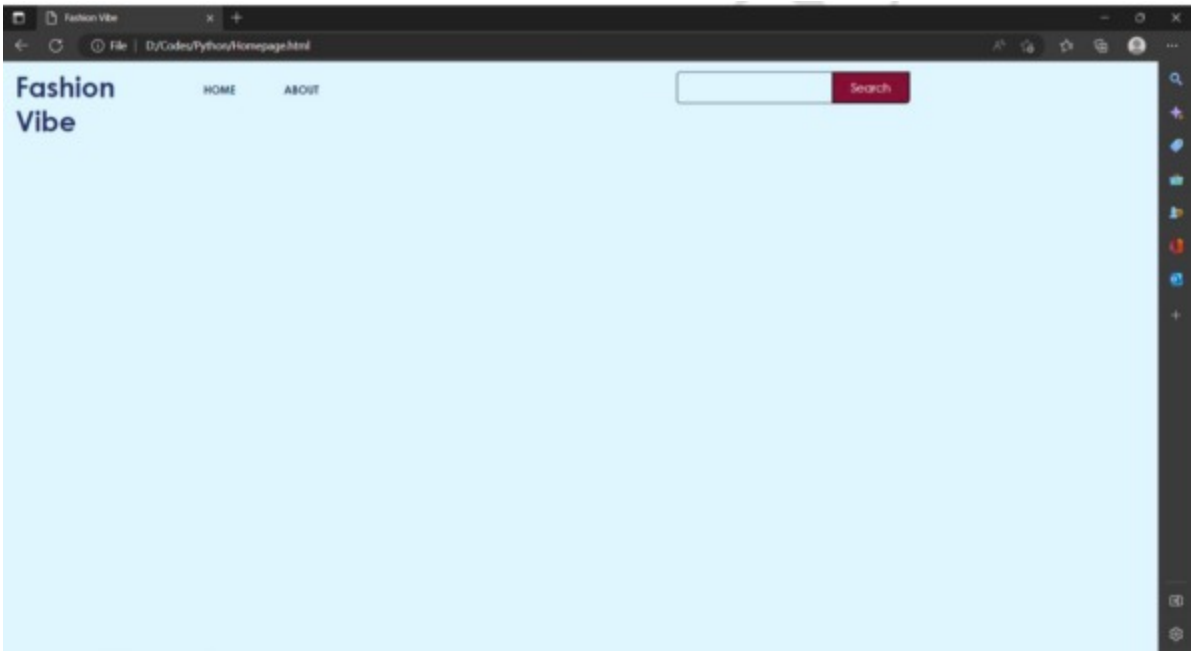
9. RESULTS

9.1 PERFORMANCE METRICS

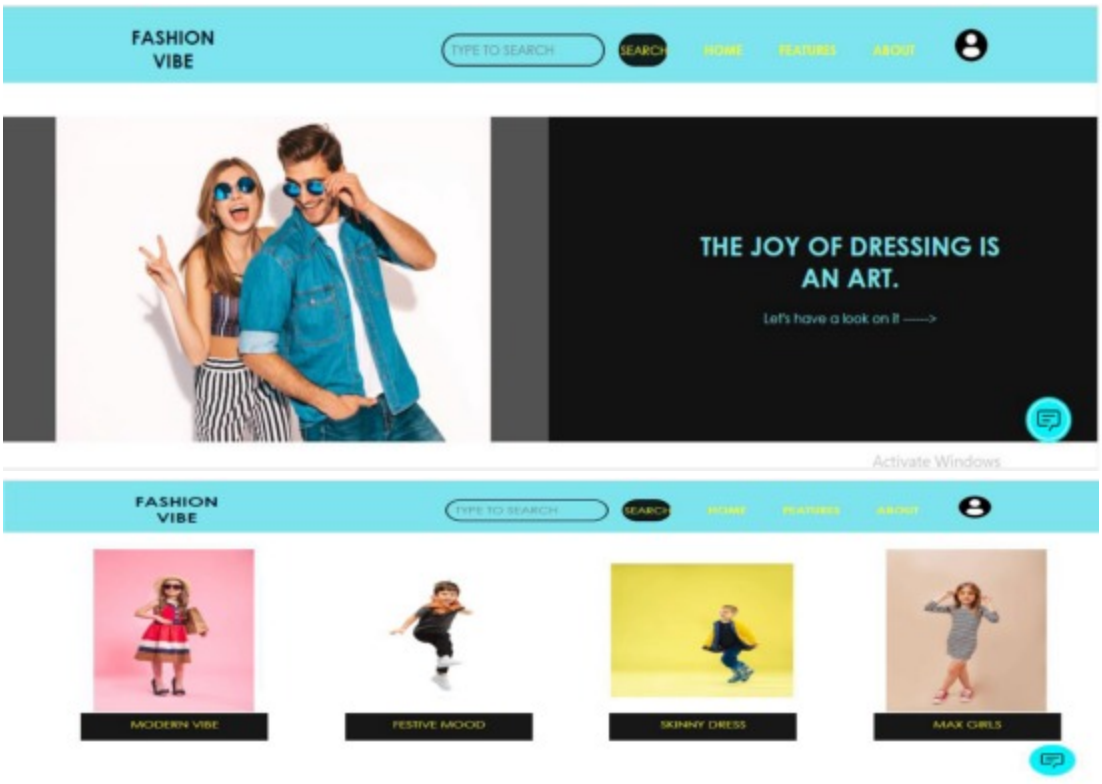
LOGIN PAGE






Fashion Vibe
HOME
ABOUT
Search

HOME PAGE


FASHION VIBE
TYPE TO SEARCH
SEARCH
HOME
FEATURES
ABOUT
THE JOY OF DRESSING IS AN ART.
Let's have a look on it ——>
Activate Windows
FASHION VIBE
TYPE TO SEARCH
SEARCH
HOME
FEATURES
ABOUT
MODERN VIBE
FESTIVE MOOD
SKINNY DRESS
MAX GIRLS

FASHION VIBE

TYPE TO SEARCH | SEARCH | HOME | FEATURES | ABOUT

FASHION VIBE

TYPE TO SEARCH | SEARCH | HOME | FEATURES | ABOUT


WEDDING SAREES


SALWAR KAMEEZ


CASUAL KURTIS


BRIDAL LEHENGA

FASHION VIBE

TYPE TO SEARCH | SEARCH | HOME | FEATURES | ABOUT

LOVABLE KIDS ATTIRE
———Smiles are always in FASHION———

FASHION VIBE

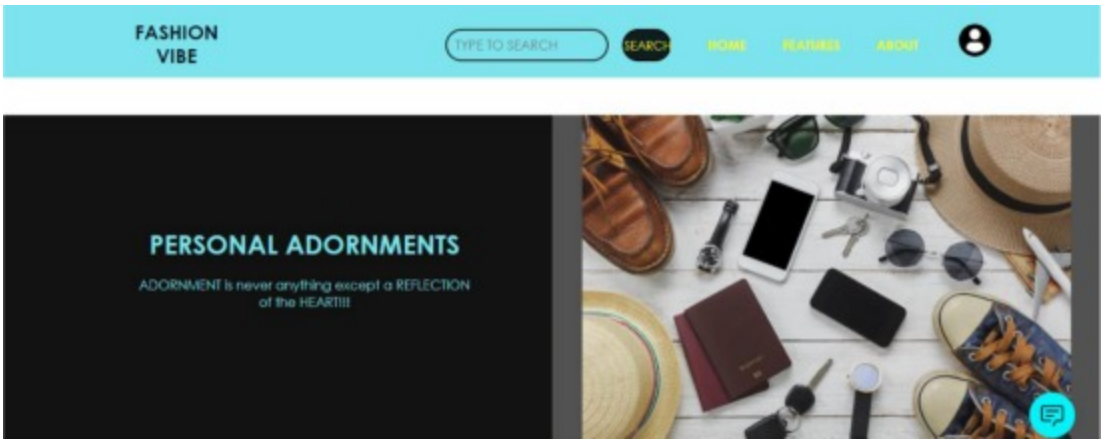
TYPE TO SEARCH | SEARCH | HOME | FEATURES | ABOUT


POLO T-SHIRTS

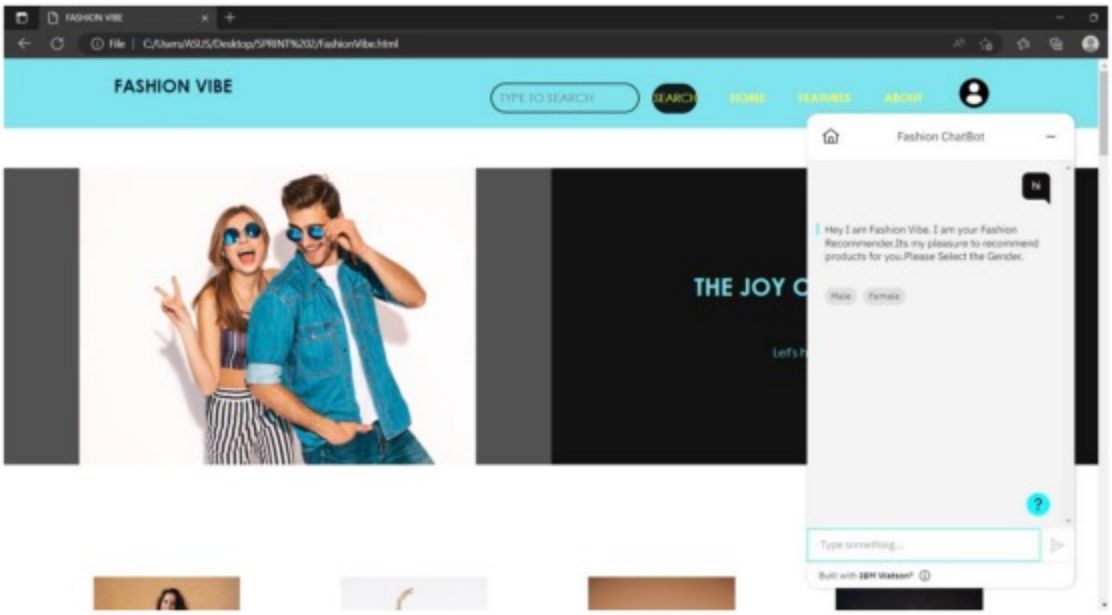

HOODIES


MEN CASUALS

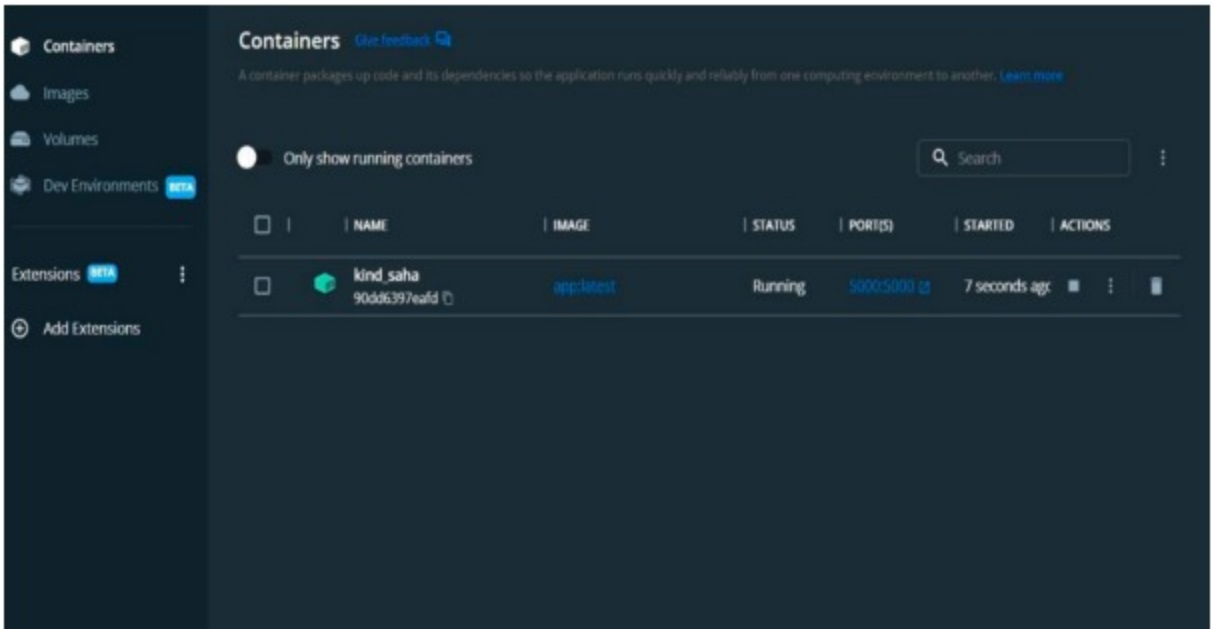

FORMAL SHIRTS



CHAT BOT




```
##############################################################
#                          WARNING!!!!                       #
# This is a sandbox environment. Using personal credentials  #
# is HIGHLY! discouraged. Any consequences of doing so are   #
# completely the user's responsibilites.                     #
#                                                            #
# The PWD team.                                              #
##############################################################
[node1] (local) root@192.168.0.28 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
841194d080c8: Pull complete
Digest: sha256:fe371ff5a69549269b24073a5ab1244dd4c0b834cbadf244870572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
[node1] (local) root@192.168.0.28 ~
$ docker run uifd/ui-for-docker
2022/11/08 09:13:47 Unix socket /var/run/docker.sock does not exist
[node1] (local) root@192.168.0.28 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
b8bddb08bb99652d924c9bcca447ad1bcdf4de7c625d20ca5dbc5e4c70cd2a00
[node1] (local) root@192.168.0.28 ~
$
```


Containers
Images
Volumes
Dev Environments
BETA
Extensions
BETA
Add Extensions
Containers
Give feedback
A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more
Only show running containers
Search
NAME
IMAGE
STATUS
PORT(S)
STARTED
ACTIONS
kind_saha
90dd6397eafd
applatest
Running
5000:5000
7 seconds ago

FEEDBACK


FEEDBACK FORM
FIRST NAME
LAST NAME
MAIL ID
COUNTRY
FEEDBACK

CHAPTER - 10

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Convenience
- Better prices
- Easy to send gifts
- Easy to send gifts
- More control
- Easy price comparisons
- No crowds
- Access to used or damaged inventory
- Privacy for discreet purchases
- The model can help users discover new interests.

DISADVANTAGES

- Lack of Data
- Shipping problems and delays
- Risk of fraud
- Less contact with your community
- Spending too much time online

CHAPTER -11

CONCLUSION

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

Moreover, different stat-of-the-art algorithms have been developed to recommend products based on users' interactions with their social groups.

Therefore, research on embedding social media images within fashion recommendation systems has gained huge popularity in recent times. This paper presented a review of the fashion recommendation systems, algorithmic models and filtering techniques based on the academic articles related to this topic.

The technical aspects, strengths and weaknesses of the filtering techniques have been discussed elaborately, which will help future researchers gain an in- depth understanding of fashion recommender systems. However, the proposed prototypes should be tested in commercial applications to understand their feasibility and accuracy in the retail market, because inaccurate recommendations can produce a negative impact on a customer.

Moreover, future research should concentrate on including time series analysis and accurate categorization of product images based on thevariation in color, trend and clothing style in order to develop an effectiverecommendation system.

CHAPTER - 12

FUTURE SCOPE

Online selling and purchasing offer innumerable benefits to both sellers and buyers, and these advantages are also the reasons for the rising scope of eCommerceWell, to put it bluntly, the scope of e-business in the near future looks to be ever-increasing and growing, because the trend has really caught on here. E-commerce giant Amazon is keen to conquer the Indian market and has already invested a great deal, especially with its 49% stake in the Future Group. Indian online retail giant Flipkart has already opened a few offline stores and plans more stores in smaller cities. They plan to combine online and offline stores to maximize their selling potential.Google and Tata Trust have launched a joint program ‘Saathi’ to increase internet and mobile penetration among rural women.The Government of India is also making a huge push for Ecommerce by providing numerous sops to startups, cyberparks, and so on through its Digital India program.As of now, there are close to 20,000 E-commerce companies in India, with many more expected to join the bandwagon every month.