IBM – NALAIYA THIRAN PROJECT

SMART FASHION RECOMMENDER APPLICATION

# INDUSTRY MENTOR : KRISHNA CHAITANYA FACULTY MENTOR : JALAJA.S

|  |  |
| --- | --- |
| TEAM ID | : PNT2022TMID22126 |
| TEAM LEAD | : Mahendra Babu Konda |
| TEAM MEMBER | : Sunkireddy Nagarjuna |
| TEAM MEMBER | : Guda Sainath |
| TEAM MEMBER | : Thaneeru Sasikanth |

## ABSTRACT

Fashion is perceived as a meaningful way of self-expressing that people use for different purposes. It seems to be an integral part of every person in modern societies, from everyday life to exceptional events and occasions. Fashionable products are highly demanded, and consequently, fashion is perceived as a desirable and profitable industry. Although this massive demand for fashion products provides an excellent opportunity for companies to invest in fashion- related sectors, it also faces different challenges in answering their customer needs.

In recent years, the textile and fashion industries have witnessed an enormous amount of growth in fast fashion. On e-commerce platforms, where numerous choices are available, an efficient recommendation system is required to sort, order, and efficiently convey relevant product content or information to users. Smart Fashion Recommender Application has attracted a huge amount of attention from fast fashion retailers as they provide a personalized shopping experience to consumers. Smart Fashion Recommender Application has been introduced to address these needs.

## TABLE OF CONTENTS

SL NO. CONTENTS PAGE NO.

INTRODUCTION

1.1 PROJECT OVERVIEW

04

1.2 PURPOSE

LITERATURE SURVEY

2.1 EXISTING PROBLEM

04

* 1. REFERENCES
  2. PROBLEM STATEMENT DEFINITION

IDEATION & PROPOSED SOLUTION

* 1. EMPATHY MAP CANVAS

3

* 1. IDEATION & BRAINSTROMING 06
  2. PROPOSED SOLUTION
  3. PROBLEM SOLUTION FIT

REQUIREMENT ANALYSIS

* 1. FUNCTIONAL REQUIREMENT

09

4

* 1. NON-FUNCTIONAL REQUIREMENTS

PROJECT DESIGN

* 1. DATA FLOW DIAGRAMS

5

SOLUTION & TECHNICALARCHITECTURE 11

* 1. USER STORIES

PROJECT PLANNING & SCHEDULING

* 1. SPRINT PLANNING & ESTIMATION

15

6

* 1. SPRINT DELIVERY SCHEDULE
  2. REPORTS FROM JIRA

CODING & SOLUTIONING

* 1. FEATURE 1

7

* 1. FEATURE 2 23
  2. DATABASE SCHEMA

TESTING

32

* 1. TEST CASES

8

* 1. USER ACCEPTANCE TESTING

RESULTS

1. 9.1 PERFORMANCE METRICS 35

1. ADVANTAGES & DISADVANTAGES 35

1. CONCLUSION 35

## 12 FUTURE SCOPE 37

13

APPENDIX 39

13.1 SOURCE CODE

13.2 GITHUB & PROJECT DEMO LINK

**SMART FASHION RECOMMENDER APPLICATION**

**1.INTRODUCTION:**

**1.1 PROJECT OVERVIEW:**

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

**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 [4–8]. Moreover, previous fashion recommendation research shows that fashion preferences vary not only from country to country but also from city to city [9]. The combination of fashion preferences and the abovementioned factors associated with clothing choices could transmit the im-age

features for a better understanding of consumers’ preferences.

**2.LITERATURE SURWAY:**

**2.1 EXISTING PROBLEM:**

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

1. **2.REFERENCES:**

Cheng, W. H., Song, S., Chen, C. Y., Hidayati, S. C., & Liu, J. (2020). Fashion meets computer vision: A survey. arXiv preprint arXiv:2003.13988.

Song, S., & Mei, T. (2018). When multimedia meets fashion. IEEE MultiMedia, 25(3), 102-108.

Guan, C., Qin, S., Ling, W., & Ding, G. (2016). Apparel recommendation system evolution: an empirical review.

International Journal of Clothing Science and Technology.

Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: a survey.

Decision Support Systems, 74, 12-32.

Liu, S., Liu, L., & Yan, S. (2014). Fashion analysis: Current techniques and future directions. IEEE MultiMedia, 21(2), 72-79.

**2.3 PROBLEM STATEMENT DEFINITION:**

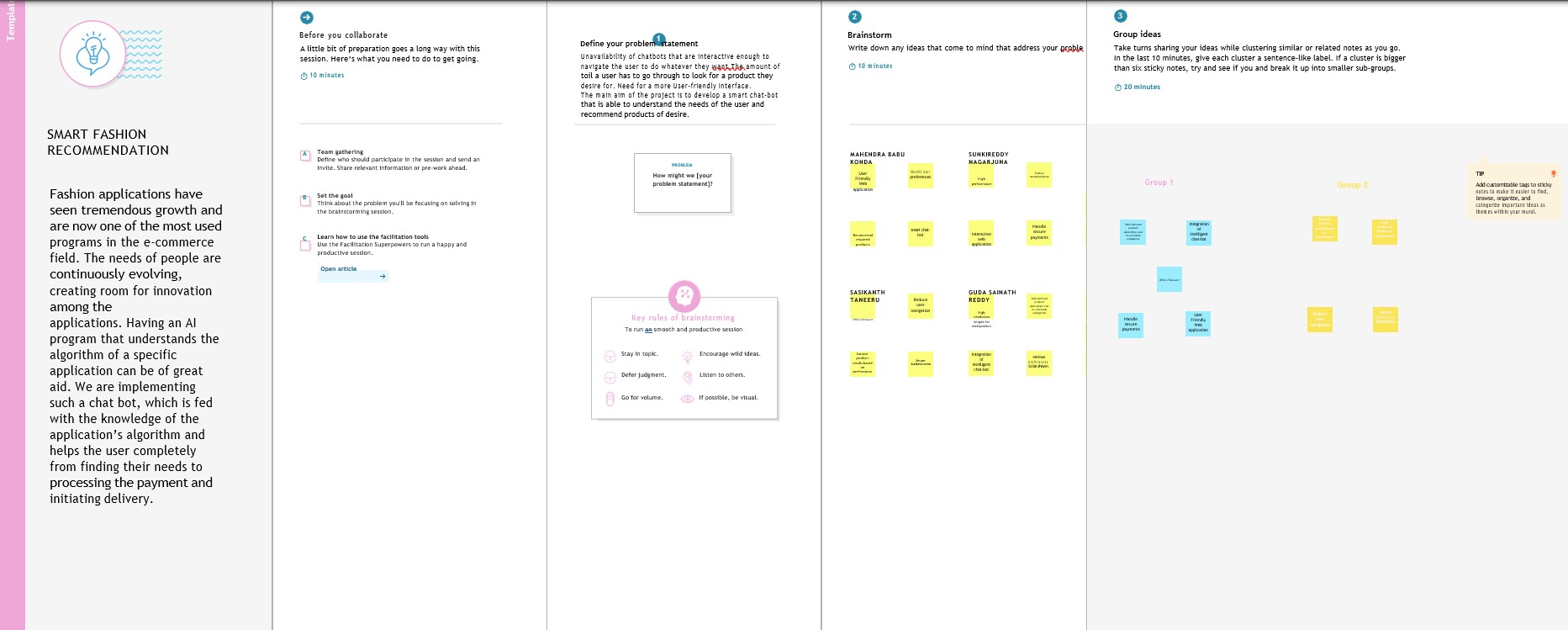
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem**  **Statement (PS)** | **I am**  **(Customer)** | **I’m trying to** | **But** | **Because** | **Which makes me feel** |
| PS-1 | I am a student | I am trying to buy a formal dress for college event | I can’t find perfect match | Either the brand or  colour  doesn’t suit | I don’t have any choice in choosing right brand. |
| PS-2 | I am a bride | Looking for wedding and reception dress | Could not able to choose right dress | Availability of wide variety of products | Like I’m overthinking regarding how my dress should looks like. |
| PS-3 | I am an  Average Man | Buy clothes to attend family function. | could not  find dress  of my  desire | Product price not within  the budget | Give up and move for other ecommerce site |

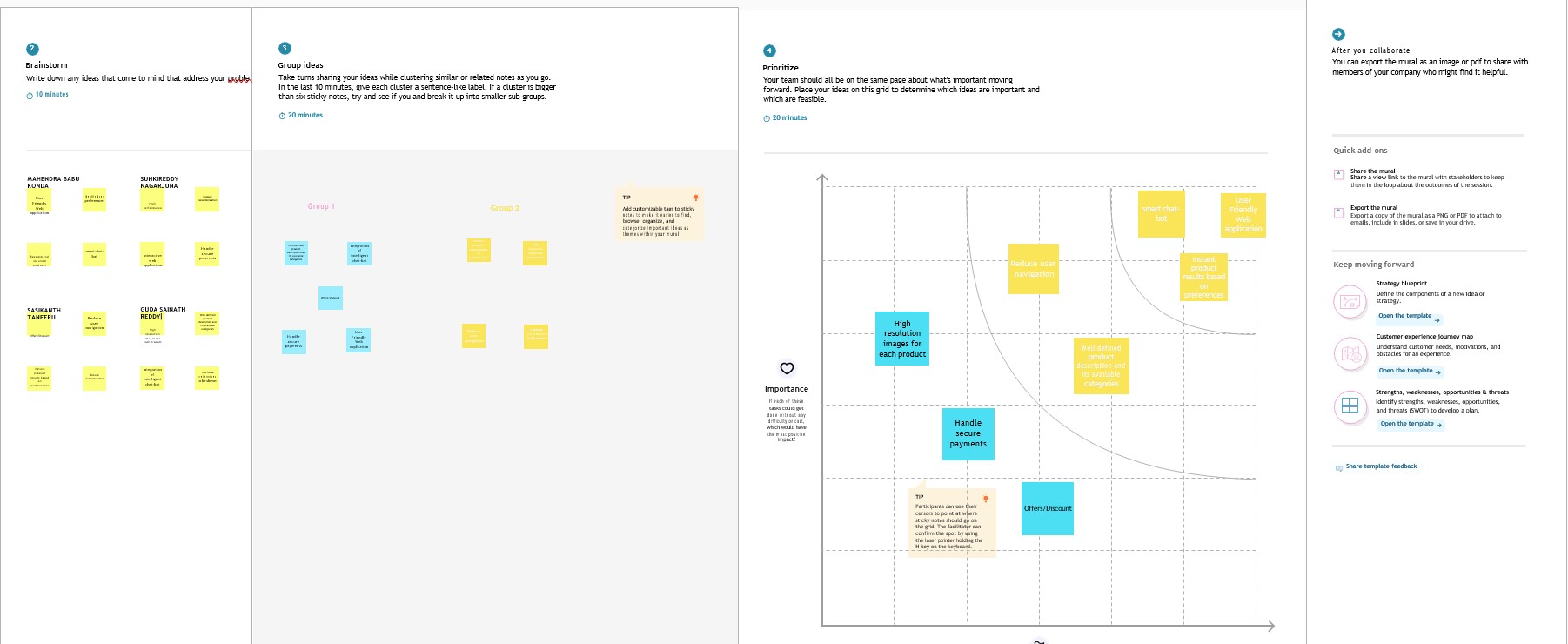
**3.IDEATION & PROPOSED SOLUTION:**

* 1. **EMPATHY MAP CANVAS:**



* 1. **IDEATION & BRAINSTORMING:**





* 1. **PROPOSED SOLUTION:**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |

|  |  |  |
| --- | --- | --- |
| 1. | Problem Statement  (Problem to be solved) | * Lack of interaction between application and user * User need to navigate across multiple pages to choose right product * Confusion in choosing product * Lack of sales * Complex User Interface. * Lack of proper guidance. |
| 2. | Idea / Solution description | By using Smart fashion recommender application:   * Improve customer relationship, interactivity and services. * Effective recommendation of products. * Recommendation within a single page via chat-bot * Collect feedback instantly. * Reduce human error * Proper guidance in accessing application. |
| 3. | Novelty / Uniqueness | • Chat-bot asks and learns from user preference which recommends appropriate products to the user without making them to search through various filters. Reduces time in choosing right product thus increases sales. |
| 4. | Social Impact /  Customer Satisfaction | • Feedback from the user at the end of session or after placing order is one of the most important factor in deriving customer satisfaction and providing better services. |
| 5. | Business Model (Revenue Model) | • The application can be developed at minimum cost with high performance and interactive user interface. |
| 6. | Scalability of the Solution | • The solution can be made scalable by using micro service architecture provided that each server responsible for certain functionality of the application. Storing user preferences along with product in browser cookie will enable to provide response instantly and allows for fetching related products. |

* 1. **PROBLEM SOLUTION FIT:**

What constraints prevent your customers from taking action or limit their choices**6. CUSTOMER CONSTRAINTS**  of **CC**  solutions? i.e. spending power, budget, no cash, network connection, available devices.

**. CUSTOMER SEGMENT(S)**

**1**

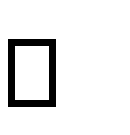
Who is your customer?

i.e. working parents of 0

-

y.o. kids

5



**Customers are those who**

**want to**

**purchase fashion items in a short time**

**CS**



* + **Most of the solution available in the internet hosts a lot of adds limiting its usability.**
  + **Needs a proper network connection**

**5**Which solutions are available to the **. AVAILABLE SOLUTIONS** customers when they face the problem

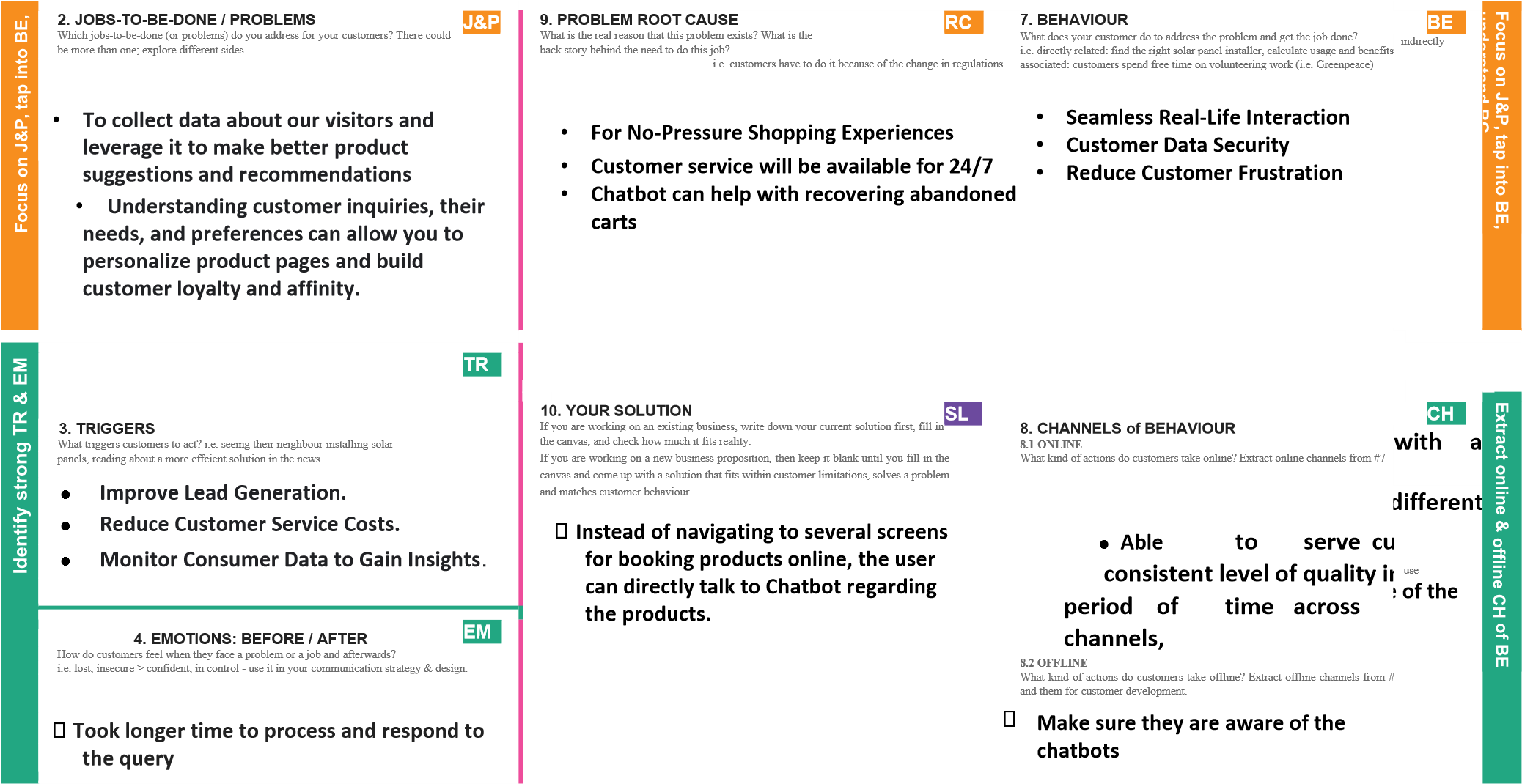
or need to get the job done? What have they tried in the past? What pros & cons these solutions have? i.e. pen and paper is an alternative to digital notetaking

* + **Smart Fashion Recommender which are supported in many browsers**
  + **Smart Fashion Recommender Chatbot is developed in this project.**

**AS**

do





**4.REQUIREMENT ANALYSIS:**

* 1. **FUNCTIONAL REQUIREMENTS:**

Following are the functional requirements of the proposed solution

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional**  **Requirements** | **Sub Registration** |
| FR-1 | Registration | Registration can be done using mobile number or gmail and needed some user  information |
| FR-2 | Login | User only log in by user id and password,Which is given during registration |
| FR-3 | Delivery confirmation | Confirmation via email and phone number |
| FR-4 | Assistance | Bot is integrated with the application to make the usability simple |

* 1. **NON-FUNCTIONAL REQUIREMENTS:**

Following are the Non-Functional requirements of the proposed solution.

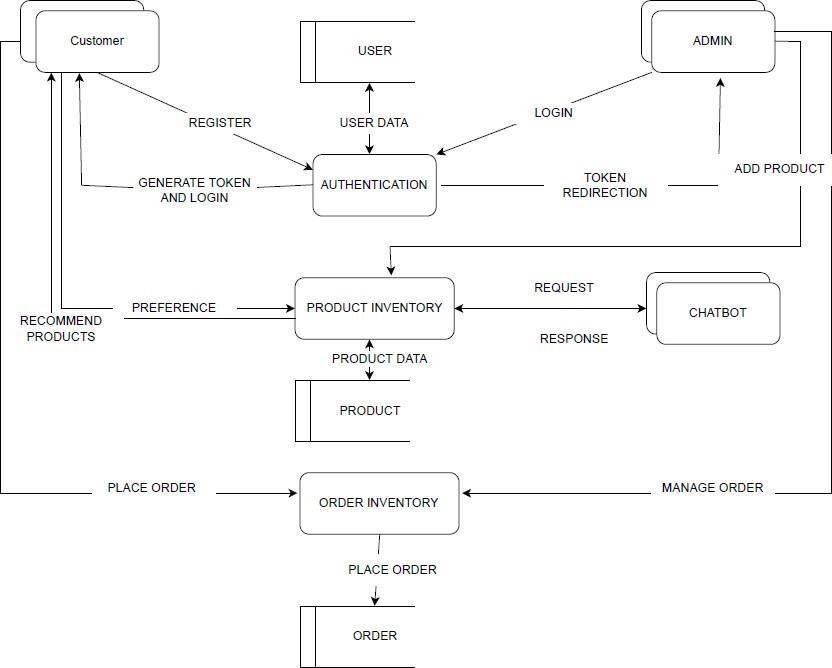
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional**  **Requirement** | **Description** |
| NFR-1 | **Usability** | A user-friendly interface with chat bot to make usability efficient |
| NFR-2 | **Security** | Secured connection HTTPS should be established for transmitting requests and  responses |
| NFR-3 | **Reliability** | The system should handle excepted as well as unexpected errors and exceptions to avoid  termination of the program |
| NFR-4 | **Performance** | The system shall be able to handle multiple requests at any given point in time and generate an appropriate response. |
| NFR-5 | **Availability** | It is a cloud based web application so user can access without any platform limitations ,just  using a browsers with a internet connection is  enough for use the application |
| NFR-6 | **Scalability** | It has a quick request and response time, high throughput, enough network resources and so on. |

**5.PROJECT DESIGN:**

**5.1 DATA FLOW DIAGRAMS:**

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

**Data Flow Diagram for the proposed solution:**



**5.2 SOLUTION AND TECHNICAL ARCHITECTURE:**

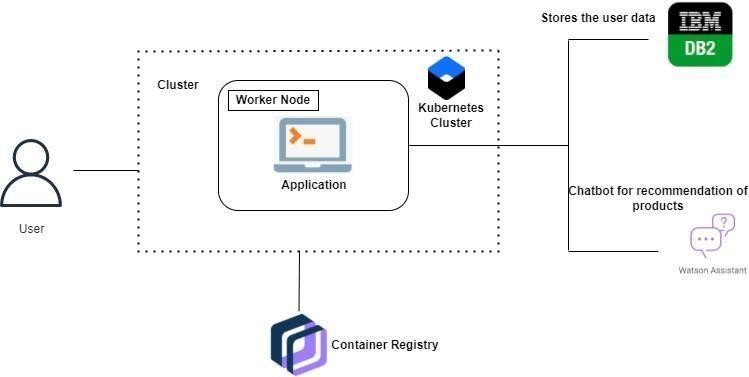
**TECHNICAL ARCHITECTURE:**

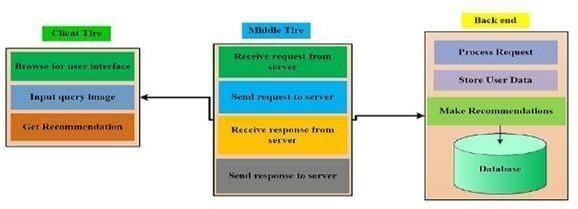
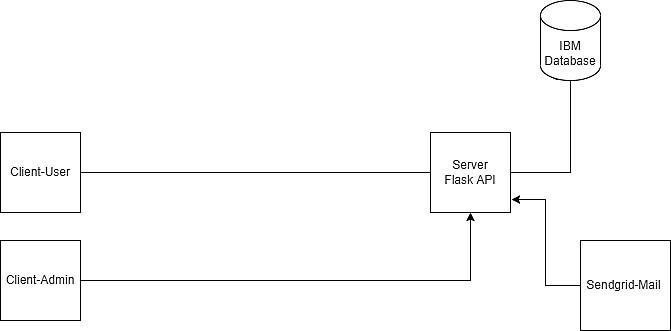
We have developed a new innovative solution through which you can directly do your online shopping based on your choice without any search. It can be done by using the chatbot. In this project you will be working on two modules:

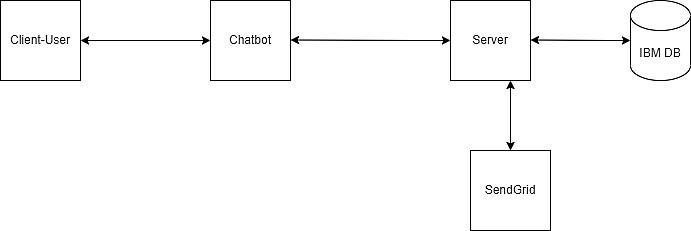
* + Admin
  + User

Instead of searching for products in the search bar and navigating to individual products to find required preferences, this project leverages the use of chatbots to gather all required preferences and recommend products to the user. The solution is implemented in such a way as to improve the interactivity between customers and applications. The chatbot sends messages periodically to notify offers and preferences. For security concerns, this application uses a token to authenticate and authorize users securely. The token has encoded user id and role.

Based on the encoded information, access to the resources is restricted to specific users. **Example - Solution Architecture Diagram:**

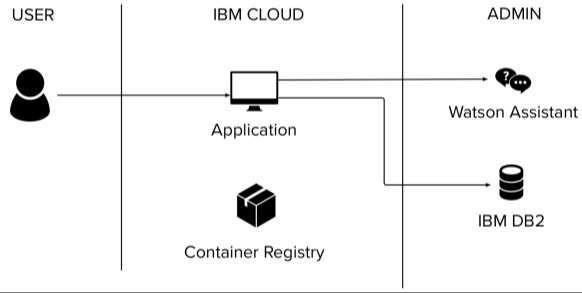






*Figure 1: Architecture and data flow of the smart fashion recommendation system*

**Technical Architecture:**



**TABLE-1:**

**COMPONENTS & TECHNOLOGIES:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1. | User Interface | How user interacts with application e.g. Web UI, Mobile App, Chatbot etc. | HTML, CSS, JavaScript /  Angular Js / React Js etc. |
| 2. | Application Logic-1 | Logic for a process in the application | Java / Python |
| 3. | Application Logic-2 | Logic for a process in the application | IBM Watson STT service |
| 4. | Application Logic-3 | Logic for a process in the application | IBM Watson Assistant |
| 5. | Database | Data Type, Configurations etc. | MySQL, NoSQL, etc. |
| 6. | Cloud Database | Database Service on Cloud | IBM DB2, IBM Cloudant etc. |
| 7. | File Storage | File storage requirements | IBM Block Storage or Other Storage Service or Local  Filesystem |
| 8. | External API-1 | Purpose of External API used in the application | IBM Weather API, etc. |
| 9. | External API-2 | Purpose of External API used in the application | Aadhar API, etc. |
| 10. | Machine Learning Model | Purpose of Machine Learning Model | Object Recognition Model, etc. |
| 11. | Infrastructure (Server /  Cloud) | Application Deployment on Local  System / Cloud  Local Server Configuration:  Cloud Server Configuration : | Local, Cloud Foundry,  Kubernetes, etc. |

**Table-2: Application Characteristics:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **Characteristics** | | **Description** | | **Technology** | |
| 1. | Open-Source Frameworks | | List the open-source frameworks used | | Technology of Opensource framework | |
| 2. | Security Implementations | | List all the security / access controls implemented, use of firewalls etc. | | e.g. SHA-256, Encryptions, IAM Controls, OWASP etc. | |
| 3. | Scalable Architecture | | Justify the scalability of architecture  (3 – tier, Micro-services) | | Technology used | |
| 4. |  | Availability |  | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) |  | Technology used |
| 5. |  | Performance |  | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. |  | Technology used |

**6.1 SPRINT PLANNING AND ESTIMATION:**

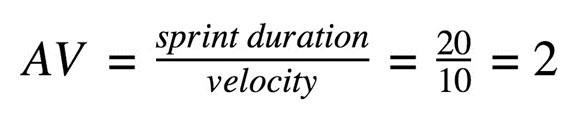
1. **PROJECT TRACKER:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sprint | Total  Story  Points | Duration | Sprint Start  Date | Sprint End  Date  (planned) | Story  Completed on Planned End Date) | Points  (as | Sprint  Release Date  (Actual) |
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 |  | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 |  | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 |  | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | | 19 Nov 2022 |

1. **VELOCITY:**

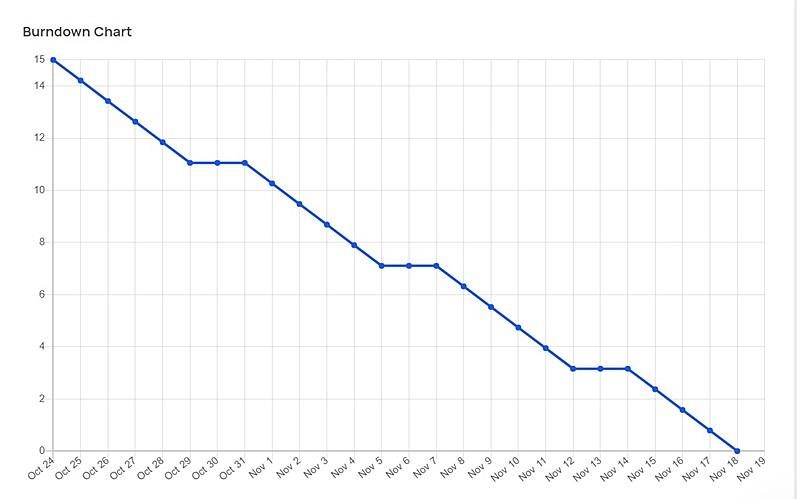
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint).

Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day).



1. **BURDOWN CHART:**

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile [software development me](https://www.visual-paradigm.com/scrum/what-is-agile-software-development/)thodologies such as [Scrum. H](https://www.visual-paradigm.com/scrum/scrum-in-3-minutes/)owever, burn down charts can be applied to any project containing measurable progress over time.



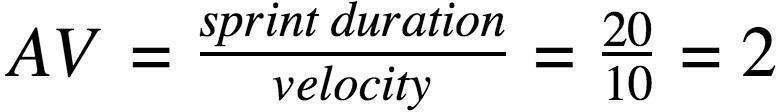
**6.1 SPRINT PLANNING AND ESTIMATION:**

1. **PROJECT TRACKER:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story**  **Points** | **Duration** | **Sprint Start Date** | **Sprint End**  **Date**  **(Planned)** | **Story**  **Points**  **Completed**  **(as on**  **Planned**  **End Date)** | **Sprint Release Date (Actual)** |
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

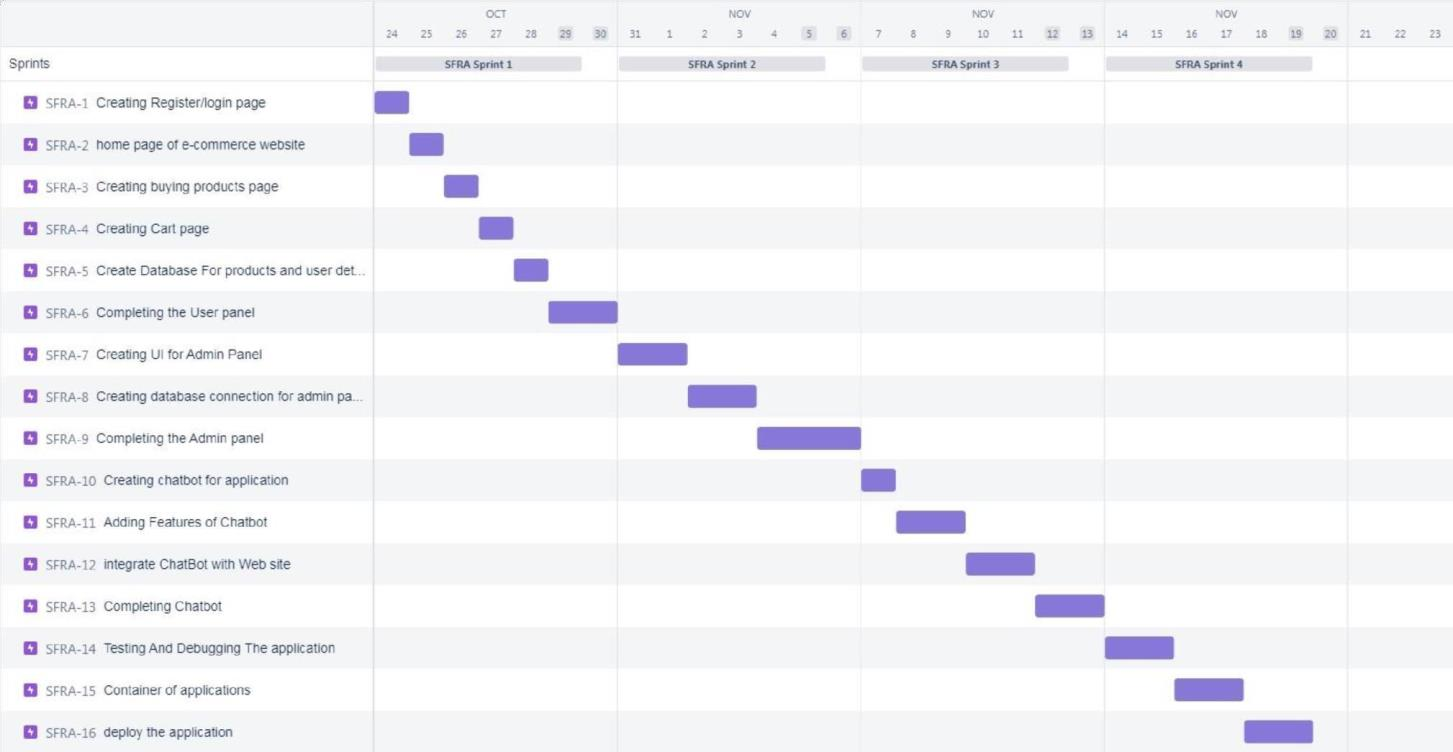
1. **VELOCITY:**

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)



1. **BURDOWN CHART:**

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile [software development me](https://www.visual-paradigm.com/scrum/what-is-agile-software-development/)thodologies such as [Scrum. H](https://www.visual-paradigm.com/scrum/scrum-in-3-minutes/)owever, burn down charts can be applied to any project containing measurable progress over time.

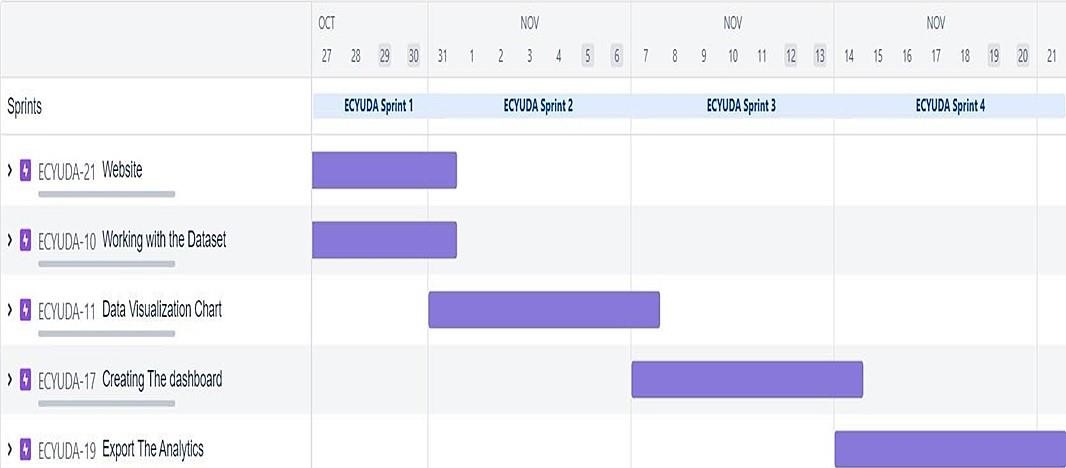


* 1. **SPRINT DELIVERY SCHEDULE:**

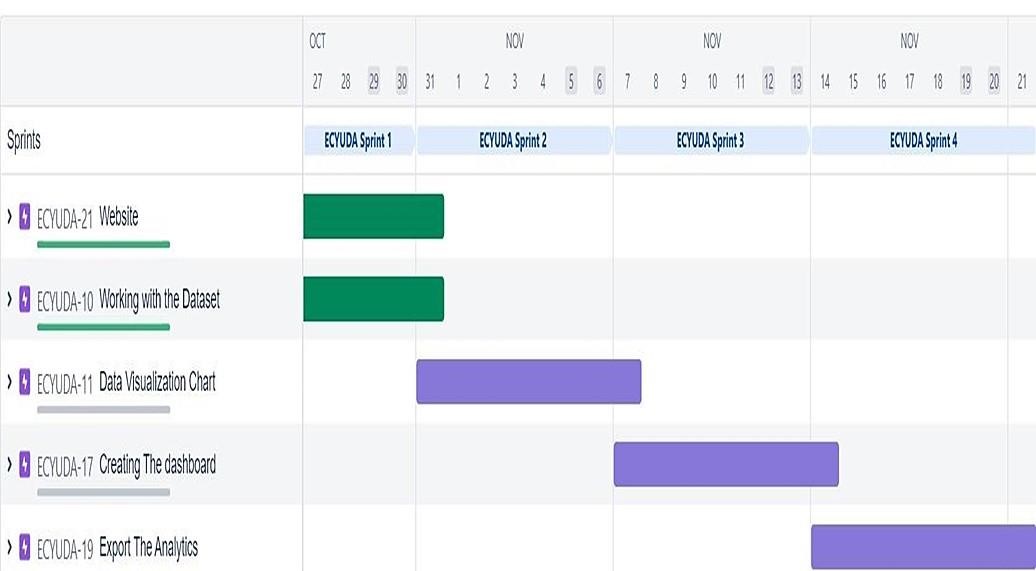
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement**  **(Epic)** | **User**  **Story**  **Number** | **User Story / Task** | **Story Points** | **Priority** | **Team**  **Members** |
| Sprint-1 | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password | 2 | High | Mahendra Babu  Konda,  Sunkireddy Nagarjuna, |
| Sprint-1 | Verification | USN-2 | As a user I will receive confirmation email once I | 2 | High | Taneeru  Sasikanth, |
|  |  |  | have registered for the application. |  |  | Guda  Sainath  Reddy |
| Sprint-1 | Login process | USN-3 | As a user I can login into the application by entering email &password. | 2 | Medium | Mahendra Babu  Konda,  Guda  Sainath  Reddy |
| Sprint -2 | Customer services | USN-4 | As a user I can contact to the customer care department on  1800 xxxx xxxx | 2 | High | Guda  Sainath  Reddy  ,Sunkireddy  Nagarjuna |
| Sprint-3 | Feedback, comment section. | USN-5 | As a user I can write a fashion review as both positive and negative. | 2 | High | Mahendra Babu, Guda  Sainath |
| Sprint-4 | Fashion sector | USN-6 | As a user I can behave differently according to the type of need. | 2 | High | Sunkireddy  Nagarjuna,  Taneeru  Sasikanth |

* 1. **REPORTS FROM JIRA:**

1. **BEFORE START OF THE SPRINT:**



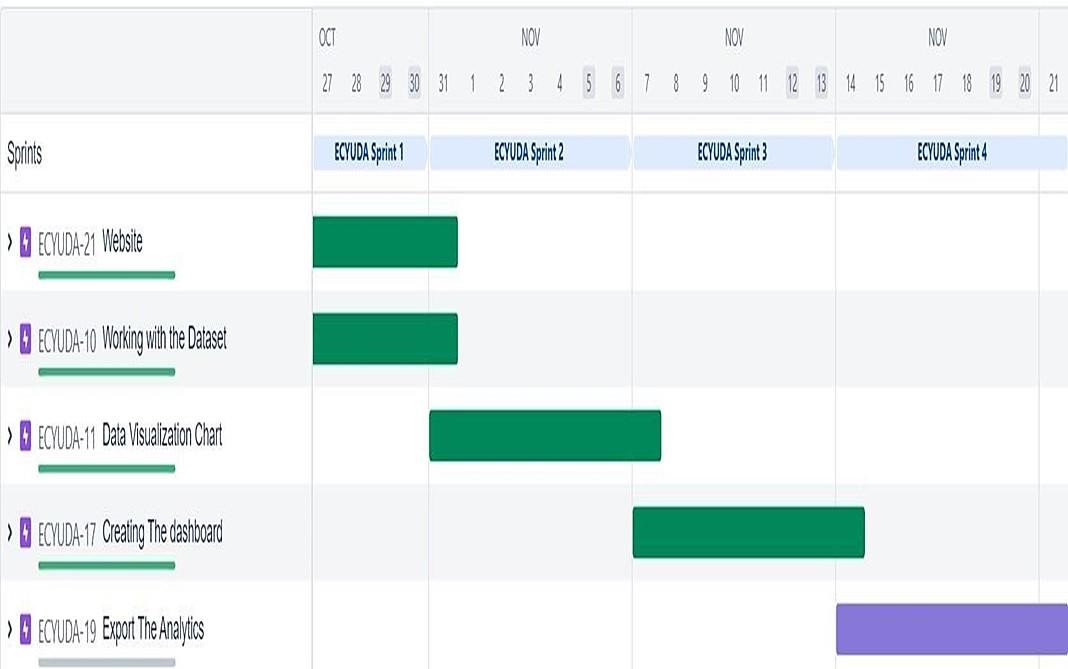
1. **SPRINT 1:**



1. **SPRINT 2:**



1. **SPRINT 3:**



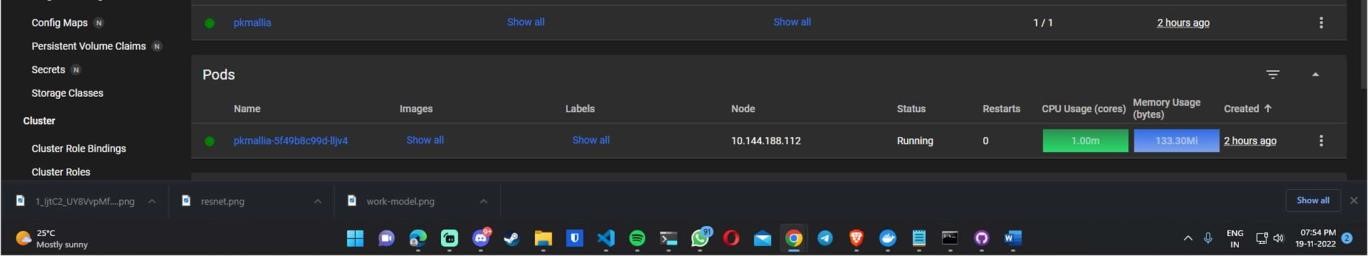
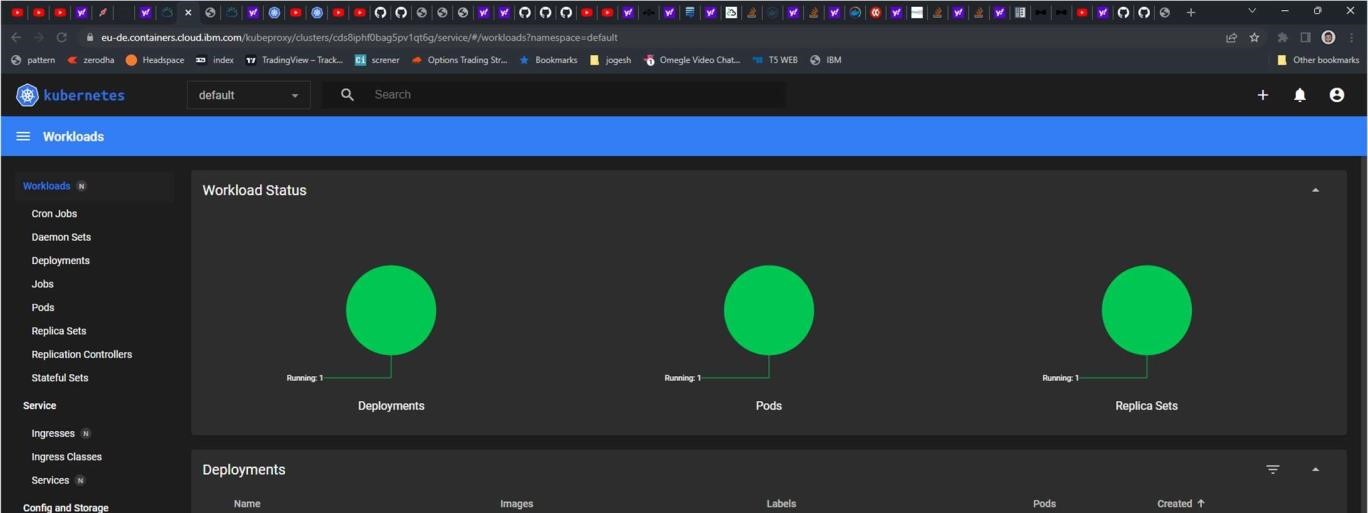
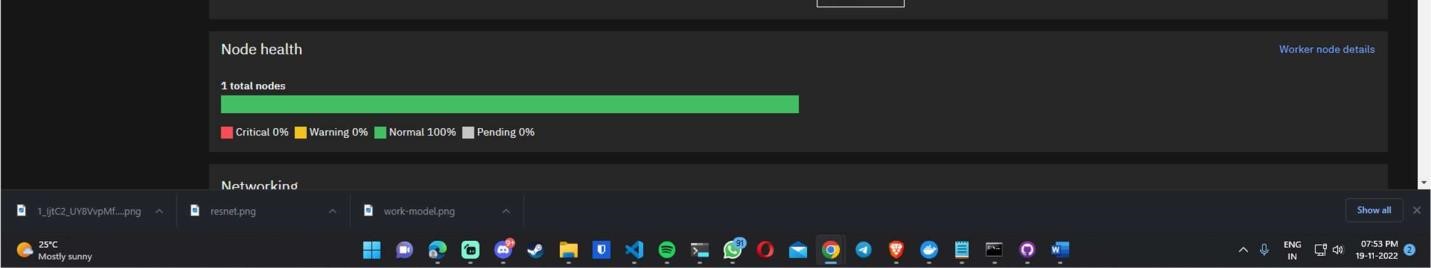
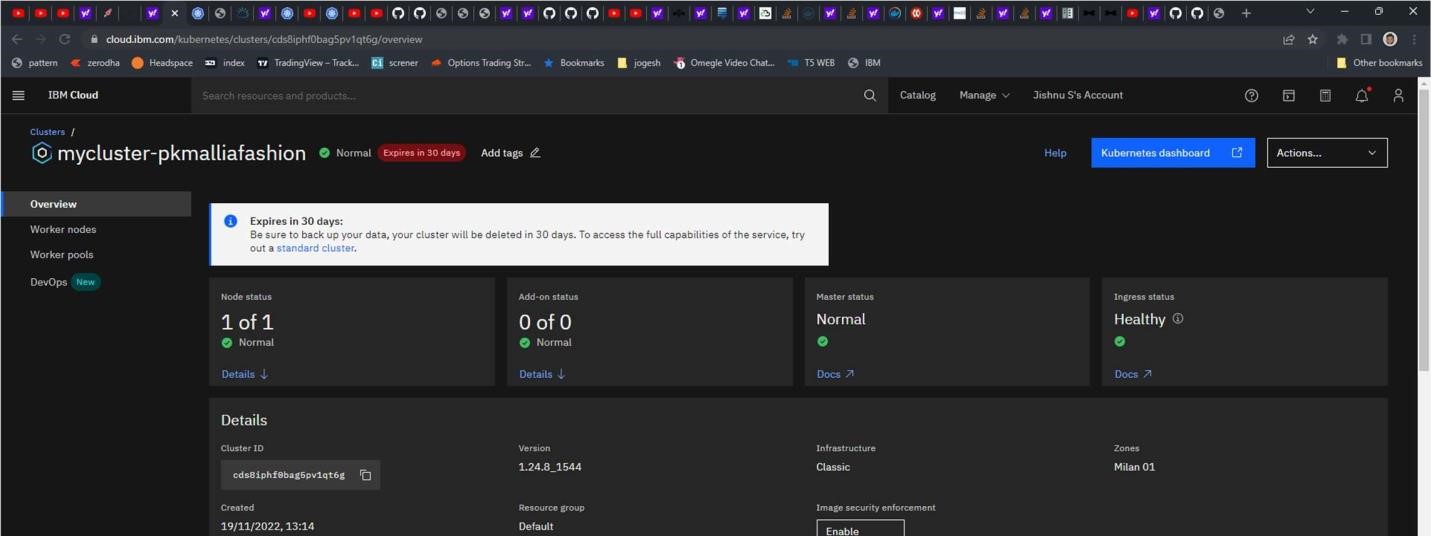
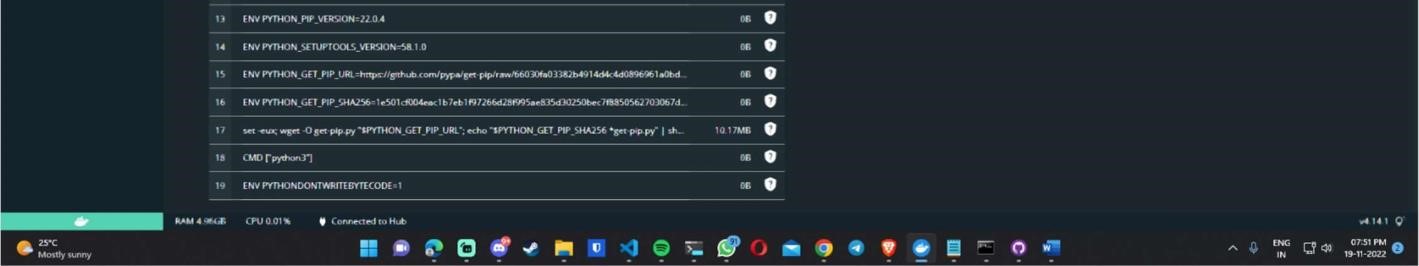
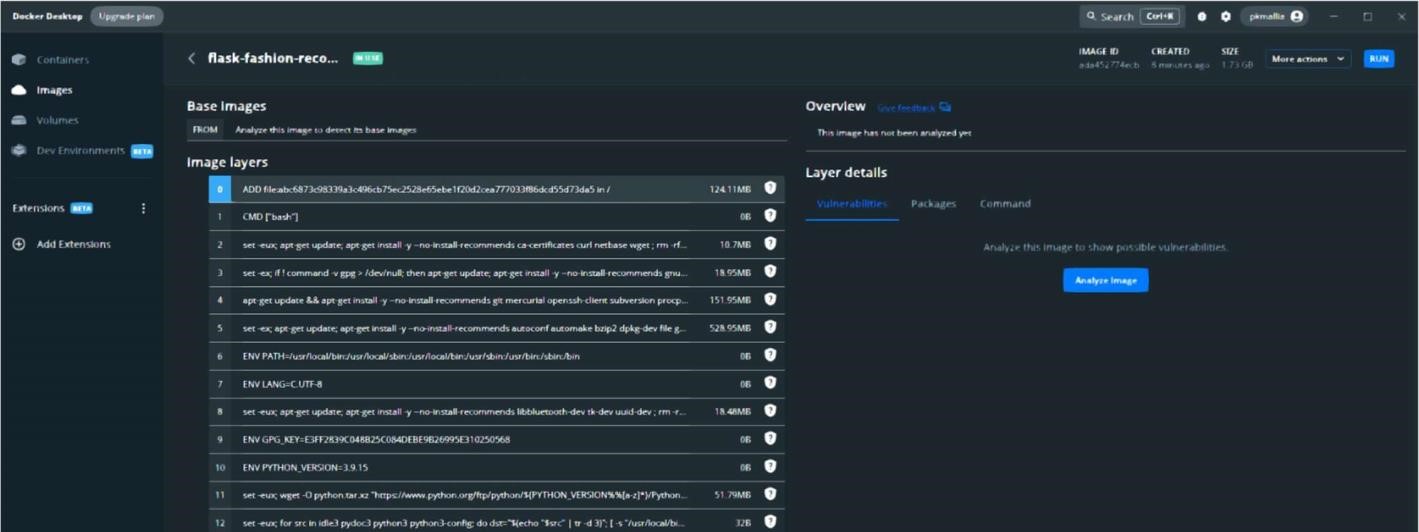
1. **SPRINT 4:**



**7.CODING & SOLUTIONING:**



7.



* 1. **Homepage HTML:**

|  |
| --- |
| {% extends 'layouts/base-fullscreen.html' %}  {% block title %} Sign IN {% endblock title %}    <!-- Specific CSS goes HERE -->  {% block stylesheets %}{% endblock stylesheets %}    {% block body\_class %} sign-in-illustration {% endblock body\_class %} {% block content %}    {% include "includes/navigation-auth.html" %}    <section>  <div class="page-header section-height-100">  <div class="container">  <div class="row">  <div class="col-xl-4 col-lg-5 col-md-7 d-flex flex-column mx-lg-0 mx-auto">  <div class="card card-plain">  <div class="card-header pb-0 text-left">  <h4 class="font-weight-bolder">Sign In</h4>  <p class="mb-0">  {% if msg %}  {{ msg | safe }}  {% else %}  Add your credentials  {% endif %}  </p>  </div> |

|  |
| --- |
| <div class="card-body">    <form method="post" action="" role="form">  {{ form.hidden\_tag() }}    <div class="mb-3">  {{ form.username(class="form-control form-control-lg", placeholder="Username") }}  </div>  <div class="mb-3">  {{ form.password(class="form-control form-control-lg", placeholder="Password", type="password") }}  </div>  <div class="form-check form-switch">  <input class="form-check-input" type="checkbox" id="rememberMe">  <label class="form-check-label" for="rememberMe">Remember me</label> </div>  <div class="text-center">  <button type="submit" name="login" class="btn btn-lg bg-gradient-primary btn-lg w100 mt-4 mb- 0">Sign in</button>  </div>  </form>  </div>  <div class="card-footer text-center pt-0 px-lg-2 px-1"> <p class="mb-4 textsm mx-auto">  Don't have an account?  <a href="{{ url\_for('authentication\_blueprint.register') }}" class="textprimary text-gradient font-weightbold">Sign UP</a>  </p>  </div>  </div>  </div>  <div class="col-6 d-lg-flex d-none h-100 my-auto pe-0 position-absolute top-0 end-0 text-center justify-content-center flex-column">  <div class="position-relative bg-gradient-primary h-100 m-3 px-7 border-radius-lg d-flex flex-column justify-contentcenter">  <img src="/static/assets/img/shapes/pattern-lines.svg" alt="pattern-lines" class="position-absolute opacity-4 start-0">  <div class="position-relative">  <img class="max-width-500 w-100 position-relative z-index-2" src="/static/assets/img/illustrations/chat.png">  </div>  <h4 class="mt-5 text-white font-weight-bolder">  SMART FASHION RECOMMENDER  </h4>  <p class="text-white">  &copy; <a target="\_blank" class="text-white" href="https://bit.ly/3fKQZaL" target="\_blank">Vel Tech High Tech </a> |
| STUDIOS</a>  - Coded by <a class="text-white" target="\_blank" href="https://appseed.us" target="\_blank">PKMALLIA</a>.  </p>  </div>  </div>  </div>  </div> </div>  </section>    {% endblock content %}    <!-- Specific JS goes HERE -->  {% block javascripts %}    <script src="/static/assets/js/soft-design-system.min.js?v=1.0.1" type="text/javascript"></script>    {% endblock javascripts %} |

**FEATURE 2:**

* 1. **finalhome.html:**

|  |
| --- |
| {% extends 'layouts/base-presentation.html' %}  {% block title %} Presentation {% endblock title %}  <!-- Specific CSS goes HERE -->  {% block stylesheets %}{% endblock stylesheets %}    {% block body\_class %} index-page {% endblock body\_class %}  {% block content %}    <header class="header-2">  <div class="page-header section-height-75 relative" style="background-image: url('/static/assets/img/curved-images/curved.jpg')">  <div class="container">  <div class="row">  <div class="col-lg-7 text-center mx-auto"> <h1 class="text-white pt-3 mt-n5"> <a class="text-white" target="\_blank" href="">Job Recommender</a>  </h1>  <p class="lead text-white mt-3"> recommends using closest near neighbours  <br />  <a class="text-white" target="\_blank" href="" target="\_blank"></a>.  </p>  </div>  </div>  </div>  <div class="position-absolute w-100 z-index-1 bottom-0"> <svg class="waves" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 24 150 40" preserveAspectRatio="none" shape-rendering="auto">  <defs>  <path id="gentle-wave" d="M-160 44c30 0 58-18 88-18s 58 18 88 18 58-18 88-18 58 18 88 18 v44h-352z" />  </defs>  <g class="moving-waves">  <use xlink:href="#gentle-wave" x="48" y="-1" fill="rgba(255,255,255,0.40" />  <use xlink:href="#gentle-wave" x="48" y="3" fill="rgba(255,255,255,0.35)" />  <use xlink:href="#gentle-wave" x="48" y="5" fill="rgba(255,255,255,0.25)" />  <use xlink:href="#gentle-wave" x="48" y="8" fill="rgba(255,255,255,0.20)" />  <use xlink:href="#gentle-wave" x="48" y="13" fill="rgba(255,255,255,0.15)" /> <use xlink:href="#gentle-wave" x="48" y="16" fill="rgba(255,255,255,0.95" />  </g>  </svg> |

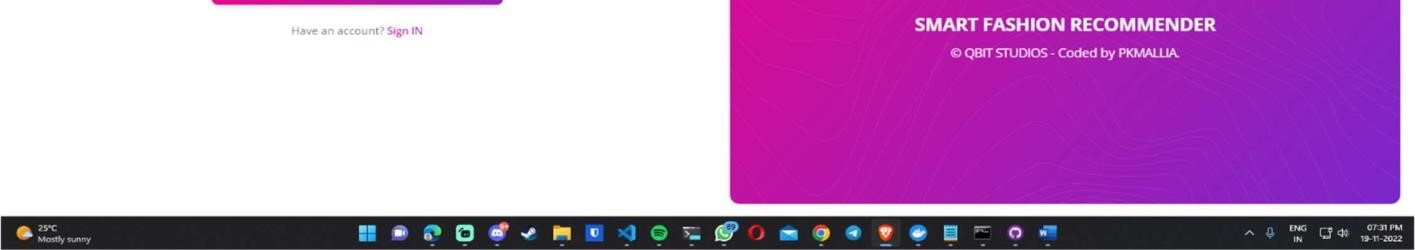
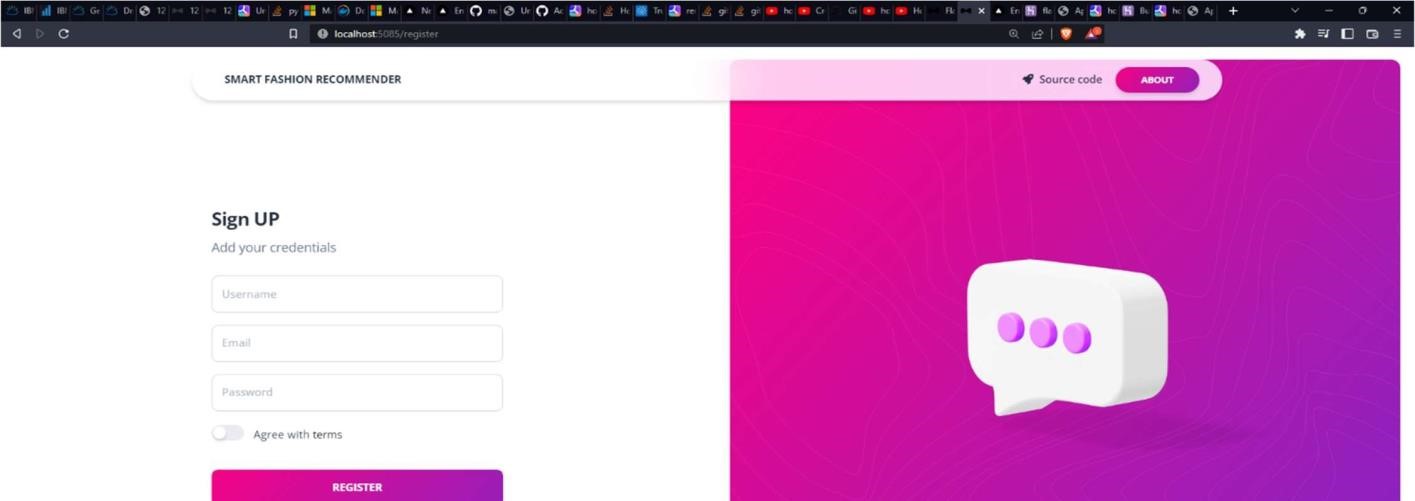
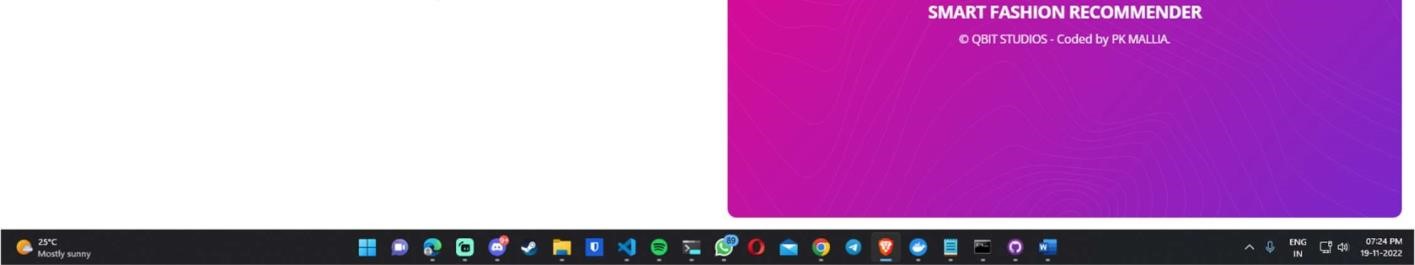
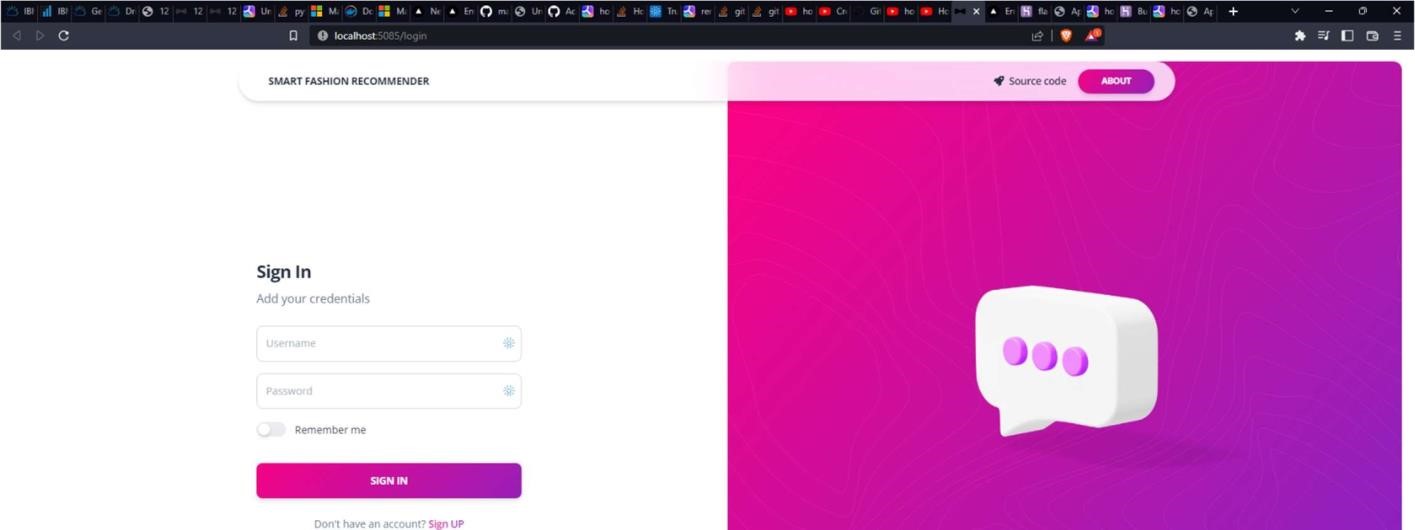
|  |
| --- |
| </div>  </div>  </header>    <script> window.watsonAssistantChatOptions = { integrationID: "0bb96b92-4e98-44c7-9dab-3a5fe2ff8562", // The ID of this integration. region: "au-syd", // The region your integration is hosted in.  serviceInstanceID: "e5babddc-2ad5-4eac-a0c5-6dad126622cb", // 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>    {% for items in product %}    <section class="my-5 py-5">  <div class="container">  <div class="row align-items-center">  <div class="col-lg-4 ms-auto me-auto p-lg-4 mt-lg-0 mt-4">  <div class="card card-background " data-tilt>  <div class="full-background" style="background-image:  url('https://images.unsplash.com/photo-1567095761054-7a02e69e5c43?ixlib=rb-  4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=687&q=80')"></ div>  <div class="card-body pt-7 text-center">  <h2 class="text-white up mb-0">{{product[items].Position}} <br />  {{product[items].Company}} <br> {{product[items].location}}</h2>  <a href="{{product[items].url}}" class="btn btn-outline-white mt-5 up btnround">Apply</a> </div> </div>  </div>  </div>  </div>  </section>  {% endfor%}    {% endblock content %}    <!-- Specific JS goes HERE -->  {% block javascripts %} |
| <script src="/static/assets/js/plugins/countup.min.js"></script>  <script src="/static/assets/js/plugins/choices.min.js"></script>  <script src="/static/assets/js/plugins/rellax.min.js"></script>  <script src="/static/assets/js/plugins/tilt.min.js"></script>  <script src="/static/assets/js/plugins/choices.min.js"></script>    <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDTTfWur0PDbZWPr7Pmq8K3jiDp0\_xUziI"></s cript>  <script src="/static/assets/js/soft-design-system.min.js?v=1.0.1" type="text/javascript"></script> <script type="text/javascript"> if  (document.getElementById('state1')) { const countUp = new CountUp('state1', document.getElementById("state1").getAttribute("countTo")); if (!countUp.error) { countUp.start();  } else { console.error(countUp.error);  } }  if (document.getElementById('state2')) { const countUp1 = new CountUp('state2',  document.getElementById("state2").getAttribute("countTo")); if  (!countUp1.error) { countUp1.start();  } else { console.error(countUp1.error);  } } if (document.getElementById('state3')) { const countUp2 = new CountUp('state3',  document.getElementById("state3").getAttribute("countTo")); if (!countUp2.error) { countUp2.start();  } else { console.error(countUp2.error);  };  }  </script>    {% endblock javascripts %} |

* 1. **DATABASE SCHEMA:**

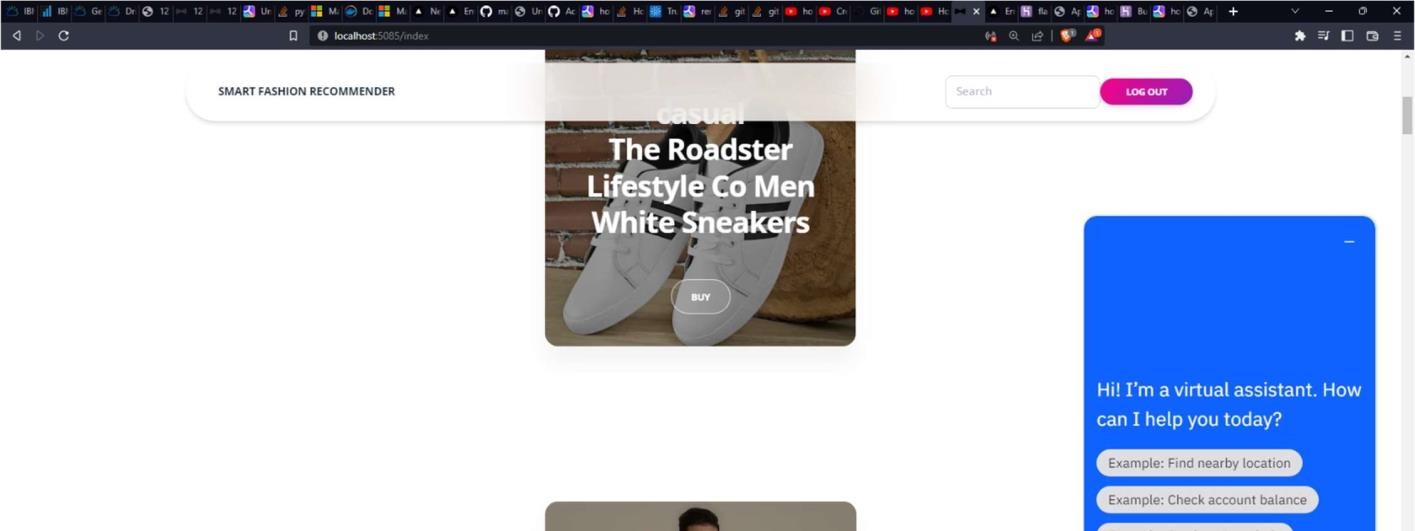
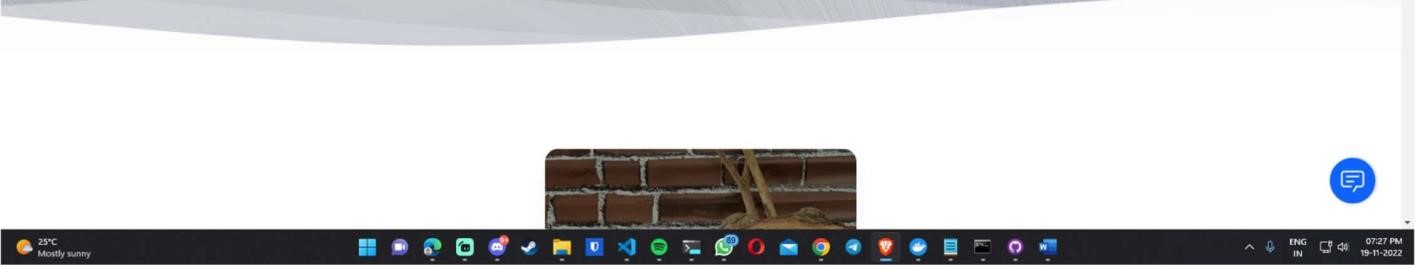
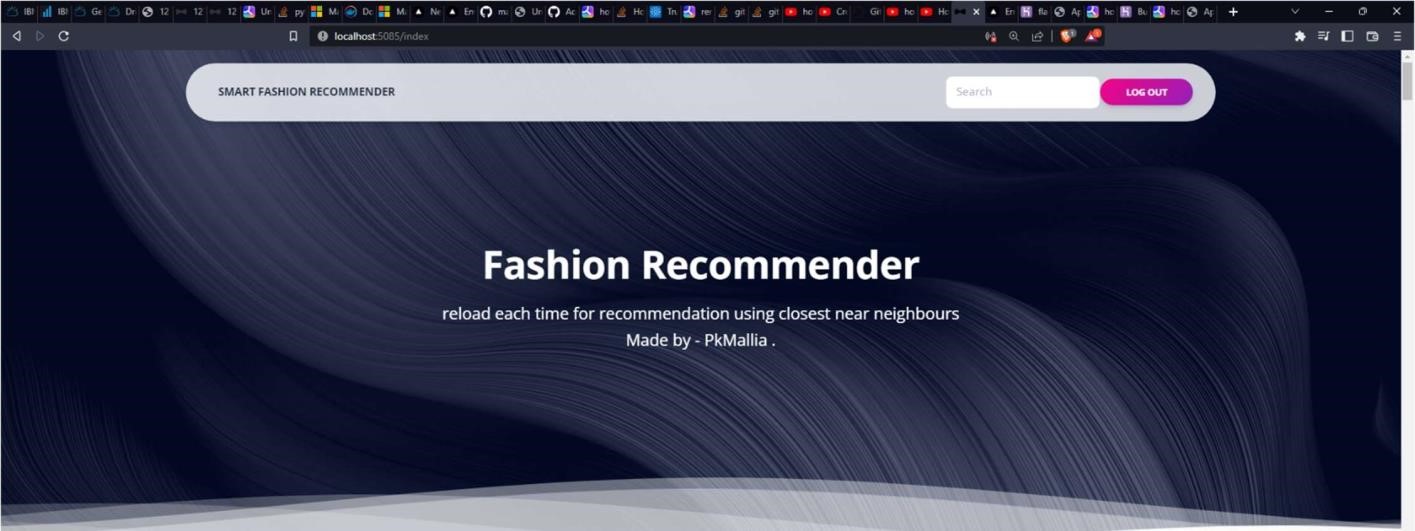
|  |
| --- |
| from flask import Flask from flask\_login import  LoginManager from flask\_sqlalchemy import  SQLAlchemy from importlib import import\_module    db = SQLAlchemy() login\_manager = LoginManager()    def register\_extensions(app): db.init\_app(app)  login\_manager.init\_app(app)    def register\_blueprints(app): for module\_name in ('authentication', 'home'): module  = import\_module('apps.{}.routes'.format(module\_name)) app.register\_blueprint(module.blueprint)    def configure\_database(app):    @app.before\_first\_request def initialize\_database(): db.create\_all()    @app.teardown\_request def shutdown\_session(exception=None):  db.session.remove()    def create\_app(config): app = Flask(\_\_name\_\_)  app.config.from\_object(config) register\_extensions(app) register\_blueprints(app) configure\_database(app) return app |

**8.TESTING:**

* 1. **TEST CASES**



* 1. **USER ACCEPTANCE TESTNG**



**9.RESULTS:**

**PERFORMANCE MATRICS:**

The performance of a recommendation algorithm is evaluated by using some specific metrics that indicate the accuracy of the system. The type of metric used depends on the type of filtering technique. Root Mean Square Error (RMSE), Receiver Operating Characteristics (ROC), Area Under Cover (AUC), Precision, Recall and F1 score is generally used to evaluate the performance or accuracy of the recommendation algorithms.

Root-mean square error (RMSE). RMSE is widely used in evaluating and comparing the performance of a recommendation system model compared to other models. A lower RMSE value indicates higher performance by the recommendation model. RMSE, as mentioned by, can be as represented as follows:

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

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



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

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



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

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



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

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

**10.ADVANTAGES & DISADVANTAGES:**

**ADVANTAGES:**

Benefits of a product recommendation engine. You do not need a market research to find out whether a customer is willing to purchase at a shop where they're getting maximum help in scouting the right product. They're also much more likely to return to such a shop in the future**.**

**DISADVANTAGES:**

The model can only make recommendations based on existing interests of the user. In other words, the model has limited ability to expand on the users' existing interests.

**11.CONCLUSION:** The

Recommendation systems have the potential to explore new opportunities for retail-ers by enabling them to provide customized recommendations to consumers based on information retrieved from the Internet. They help consumers to instantly find the products and services that closely match with their choices. 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 the variation in colour, trend and clothing style in order to develop an effective recommendation system

**12.FUTURE SCOPE:**

The dashboard creation, visualization have taken lots of procedures and steps. The aim of the future work is to analyze the target attribute by reducing the number of procedures and steps. To improve the accuracy of the analysis algorithm selection procedure need to be optimized. As a future work, the results of the analysis can be improved, using the large number of dresses and fashion cloth parameters. This can be also implemented in machine learning model to build in a strong textile prediction model and analysis of all the cloths with different climatic conditions and with different requirements in different areas.

**13.APPENDIX:**

**SOURCE CODE:**

Flask app.py:

|  |
| --- |
| from apps.home import blueprint from flask import  render\_template, request from flask\_login import login\_required from jinja2 import TemplateNotFound import pandas as pd import numpy as np from ftfy import fix\_text from nltk.corpus import stopwords  from sklearn.metrics.pairwise import cosine\_similarity from  sklearn.feature\_extraction.text import TfidfVectorizer from sklearn.neighbors import NearestNeighbors import re  @blueprint.route('/index')  @login\_required def index(): with open('/job\_final.csv',encoding="utf-8") as f:  df1 = pd.read\_csv(f) df1.to\_string()  df2 = df1.sample(25) df3 = df2.to\_dict('index') print(df3) print(type(df3))  return render\_template('home/index.html', product = df3, segment='index') # @blueprint.route('/search',methods=['GET','POST']) # def recommender():  # with open('G:\\Flask-Fashion-Recommender\\flask-pixel\\myntra-database.csv') as f1: # stopw = set(stopwords.words('english'))  # df11 = pd.read\_csv(f1)  # df11['test']=df11['gender','category','type','brand','description'].apply(lambda x: ' '.join([word for word in str(x).split() if len(word)>2 and word not in (stopw)])) # def ngrams(string, n=3):  # string = fix\_text(string) # fix text  # string = string.encode("ascii", errors="ignore").decode() #remove non ascii chars  # string = string.lower()  # chars\_to\_remove = [")","(",".","|","[","]","{","}","'"]  # rx = '[' + re.escape(''.join(chars\_to\_remove)) + ']' |

|  |
| --- |
| # string = re.sub(rx, '', string)  # string = string.replace('&', 'and')  # string = string.replace(',', ' ')  # string = string.replace('-', ' ')  # string = string.title() # normalise case - capital at start of each word # string = re.sub(' +',' ',string).strip() # get rid of multiple spaces and replace with a single  # string = ' '+ string +' ' # pad names for ngrams...  # string = re.sub(r'[,-./]|\sBD',r'', string)  # ngrams = zip(\*[string[i:] for i in range(n)])  # return [''.join(ngram) for ngram in ngrams]  # vectorizer = TfidfVectorizer(min\_df=1, analyzer=ngrams, lowercase=False)  # tfidf = vectorizer.fit\_transform(org\_name\_clean)      # def getNearestN(query):  # queryTFIDF\_ = vectorizer.transform(query)  # distances, indices = nbrs.kneighbors(queryTFIDF\_)  # return distances, indices    # nbrs = NearestNeighbors(n\_neighbors=1, n\_jobs=-1).fit(tfidf)  # unique\_org = (df11['test'].values)  # distances, indices = getNearestN(unique\_org)  # unique\_org = list(unique\_org)  # matches = []  # for i,j in enumerate(indices):  # dist=round(distances[i][0],2)    # temp = [dist]  # matches.append(temp)  # matches = pd.DataFrame(matches, columns=['Match confidence'])  # df11['match']=matches['Match confidence']  # df111=df11.sort\_values('match')  # df22=df111[['Product\_url',  'image\_url','type','category','description','brand']].head(30).reset\_index()  # return render\_template('includes/card-each.html', sproduct = df22, segment='index')    @blueprint.route('/<template>')  @login\_required def route\_template(template): try: if not template.endswith('.html'):  template += '.html'    # Detect the current page |
| segment = get\_segment(request)    # Serve the file (if exists) from app/templates/home/FILE.html return render\_template("home/" + template, segment=segment)  except TemplateNotFound: return render\_template('home/page-404.html'), 404 except: return render\_template('home/page-500.html'), 500  # Helper - Extract current page name from request def get\_segment(request): try:    segment = request.path.split('/')[-1] if segment == '': segment = 'index' return segment except:  return None |

# INTEGRATING APPLICATION WITH CHATBOT USING WATSON ASSISTANT

CODE :

<script> window.watsonAssistantChatOptions = { integrationID: "614a4315-ff80-4187-8fe42fd9b506b723", // The ID of this integration. region: "au-syd", // The region your integration is hosted in. serviceInstanceID: "9670dcf8-789f-4609-8d7a-6e25c412a9ec", // 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>

**13.2 GITHUB & PROJECT DEMO LINK:**

**GITHUB:** https://github.com/IBM-EPBL/IBM-Project-6657-1658834239

**PROJECT DEMO:**

https://drive.google.com/file/d/1Lk4Dx44fkwzdZ8PsJZz4C\_Julx6nKlav/view?usp=drivesdk