



## **IBM PROJECT**

### **SMART FASHION RECOMMENDER APPLICATION**

**BATCH:**B6-6M2E

**TEAM ID:**PNT2022TMID47421

**TEAM LEADER:**JEYA ROSELIN J

#### **TEAM MEMBERS:**

ILAKKIYA SHENBAGAM M

SARULATHA M

PIRAMA RAJI R

**FACULTY MENTOR:**ANANDHI MEENA B

**INDUSTRY MENTOR:**KRISHNA CHAITANYA

## CONTENTS

<b>S.No</b>	<b>TITLE</b>	<b>Pg.no</b>
1	INTRODUCTION	4
2	LITERATURE SURVEY	
	A. Abstract	4
	B. References	5
	C. Problem statement	6
3	IDEATION AND PROPOSED SOLUTION	
	A. Empathy map	7
	B. Ideation	8
	C. Proposed solution	9
	D. Problem solution fit	10
4	SOFTWARE REQUIREMENTS	10
5	PROJECT DESIGN	
	A. Data flow diagram	11
	B. Technical architecture	11
	C. User stories	12
6	SOLUTION REQUIREMENTS	
	A. Functional requirements	12
	B. Non-functional requirements	13
7	PROJECT PLANNING	

	A. Sprint planning	13
	B. Sprint delivery schedule	14
	C. Burdown diagram	14
8	CONCLUSION	15
9	FUTURE SCOPE	15
10	SOURCE CODE	16
11	OUTPUT IMAGES	44
12	GITHUB LINK	49

## **1.INTRODUCTION**

Internet shopping is productively growing in today's world of e-commerce. So, product recommendation systems also has a chance to be developed. Since users need a relationship between them and system. 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.

## **2.LITERATURE SURVEY**

### **A. ABSTRACT**

There has been number of researches in the field of data analytics to recommend the products, and many of them are successfully recommend the desired content what user is asking for. There are several domains such as ecommerce, movies,musical instruments, websites, books, etc. where recommendation system has its own significance. The outcomes provided by these domains has the scope of improvements so that final result will reach to user's satisfaction. This research use the recommendation system and provide the output desired by user. Research is introducing the chat bot which recommends the product to the customers as per their requirement. The chat bot is basically order taking with minimal user input and suggested the appropriate product.Customer provide the details about the product

through the chat bot. And according to user description it will recommend the related products. Doc2Vec, Latent Semantic Analysis, and Sentiment Analysis come together to make relevant recommendations in a chat bot interface.

## **B. REFERENCES**

1. Chung, M., Ko, E., Joung, H., & Kim, S. J. (2020). Chatbot e-service And customer satisfaction regarding luxury brands. *Journal of Business Research*, 117, 587–595. Colombi, C., Kim, P., & Wyatt, N. (2018). Fashion retailing
2. Kasilingam, D. L. (2020). Understanding the attitude and intention to Use smartphone chatbots for shopping *Technology in Society*, 62, 101280
3. Liu, C., Jiang, J., Xiong, C., Yang, Y., & Ye, J. (2020). Towards Building An Intelligent chatbot for customer service: Learning to respond at the appropriate time. *ACM SIGKDD KDD*, 3377–3385.
4. Sanny, L., Susastra, A., Roberts, C., & Yusramdaleni, R. (2020). The Analysis of customer satisfaction factors which influence chatbot Acceptance in Indonesia. *Management Science Letters*, 10(6), 1225–1232.
5. Pricilla, C., Lestari, D. P., & Dharma, D. (2018). Designing interaction For chatbot-based conversational commerce with user-centered Design. *ICAICTA*, 2018, 244–249.
6. Mauldin, M. L. (1994). Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. *AAAI*, 94, 16–21.

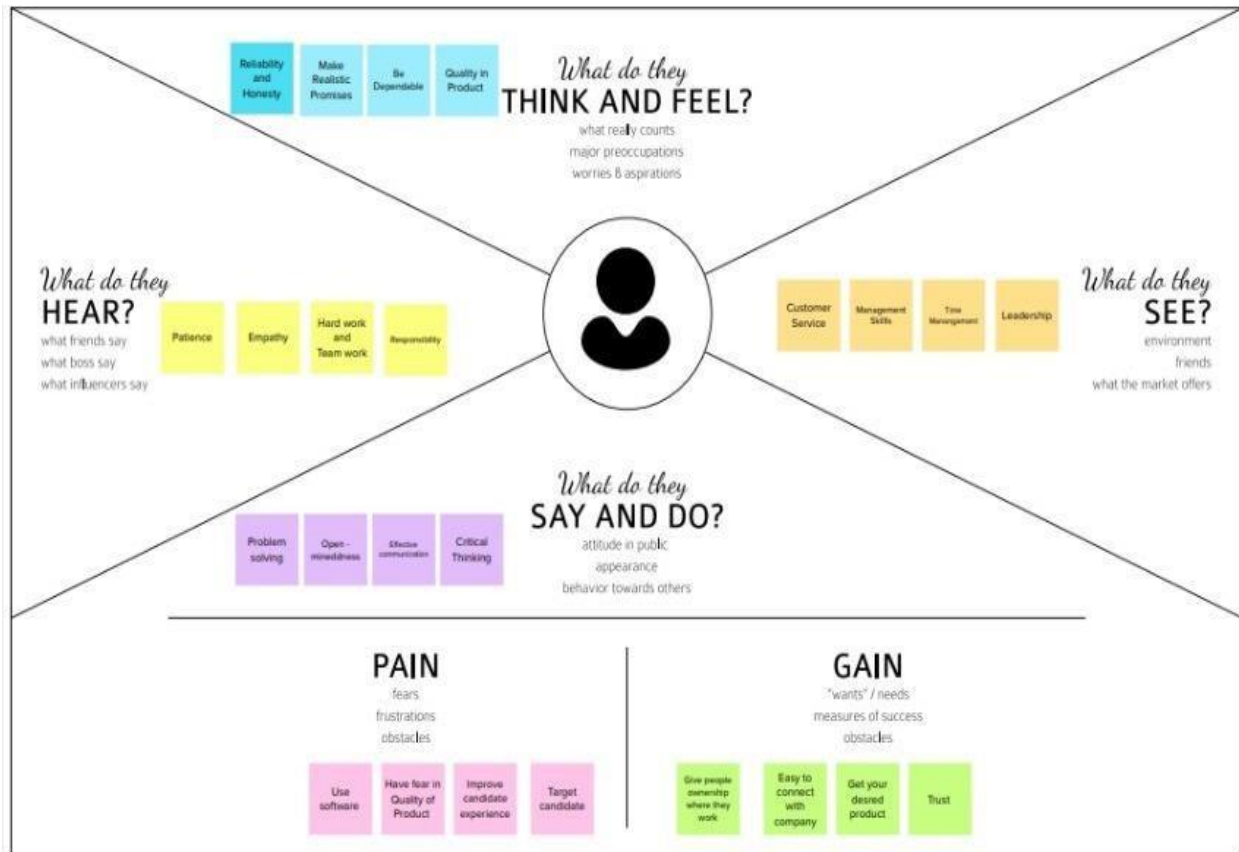
7. Kucherbaev, P., Psyllidis, A. and Bozzon, A. (2017). Chatbots as Conversational recommender systems in urban contexts Proceedings of the International Workshop on Recommender Systems for Citizens, CitRec '17, ACM, New York, NY, USA, pp. 6:1–6:2.
8. Nazir, A., Khan, M. Y., Ahmed, T., Jami, S. I., & Wasi, S. (2019). A novel Approach for ontology-driven information retrieving chatbot for Fashion brands. International Journal of Advanced Computer Science And Applications, 10(9), 546–552.
9. Qiu, M., Li, F. L., Wang, S., Gao, X., Chen, Y., Zhao, W., ... Chu, W. (2017). Alime chat: A sequence to sequence and rank based Chatbot engine. ACL 2017 – (long papers), 2, 498–503.

### **C. PROBLEM STATEMENT**

Internet shopping is productively growing in today's world of e-commerce. So, product recommendation systems also has a chance to be developed. Since users need a relationship between them and system. 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.

### 3.IDEATION AND PROPOSED SOLUTION

#### A. EMPATHY MAP



## B. IDEATION





## C. PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ul style="list-style-type: none"> <li>• Lack of interaction between application and user.</li> <li>• User need to navigate across multiple pages to choose right product.</li> <li>• Confusion in choosing product Lack of sales Complex User Interface.</li> <li>• Lack of proper guidance.</li> </ul>
2.	Idea / Solution description	<p>By using Smart fashion recommender application:</p> <ul style="list-style-type: none"> <li>• Improve customer relationship, interactivity and services.</li> <li>• Effective recommendation of products.</li> <li>• Recommendation within a single page via chat-bot.</li> <li>• Collect feedback instantly.</li> <li>• Reduce human error.</li> <li>• Proper guidance in accessing application.</li> </ul>
3.	Novelty / Uniqueness	<ul style="list-style-type: none"> <li>• 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.</li> </ul>
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> <li>• 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</li> </ul>
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"> <li>• The application can be developed at minimum cost with high performance and interactive user interface.</li> </ul>
6.	Scalability of the Solution	<ul style="list-style-type: none"> <li>• 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</li> </ul>

## D. PROBLEM SOLUTION FIT

<p><b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span></p> <p>Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one, explore different sides.</p> <p>•to collect data about visitors and leverage it to make better suggestions.</p> <p>*Understanding customer needs, inquiries, preferences to give the best suggestion.</p>	<p><b>9. PROBLEM ROOT CAUSE</b> <span>RC</span></p> <p>What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.</p> <p>*For no pressure shopping Experience</p> <p>*Service available in 24/7 hours.</p> <p>*Chatbot can help with Recovering abandoned carts.</p>	<p><b>7. BEHAVIOUR</b> <span>BE</span></p> <p>What does your customer do to address the problem and what does your customer feel the right user panel enables, calculate usage and benefits, indirectly associated: customers spend less time on volunteering work (i.e. Greenpeace)</p> <p>*Seamless real life interaction</p> <p>*Customer Data security</p> <p>*Reduce Customer frustration.</p>
<p><b>3. TRIGGERS</b> <span>TR</span></p> <p>What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.</p> <p>*Improve lead generation.</p> <p>*Reduce customer Service costs.</p> <p>*Monitor customer Data to gain insights.</p>	<p><b>10. YOUR SOLUTION</b> <span>SL</span></p> <p>If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour</p> <p>*Instead of navigating many screens for booking products, the user can directly talk to chatbot regarding booking.</p>	<p><b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span></p> <p><b>8.1 ONLINE</b> What kind of actions do customers take online? Extract online channels from #7</p> <p>*Able to serve customer with the consistent level of quality with the short period of time, across different channels.</p> <p><b>8.2 OFFLINE</b> What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.</p> <p>*Make sure they are aware of the use of chatbots.</p>
<p><b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span></p> <p>How do customers feel when they face a problem or a job and afterwards? i.e. feel reassured, confident, in control - use it in your communication strategy &amp; design.</p> <p>Took longer time to process and respond to the Query.</p>		

## 4.SOFTWARE REQUIREMENTS

- IBM Cloud
- HTML
- Javascript
- IBM Cloud Object Storage
- Python-Flask
- Kubernetes
- Docker
- IBM DB2
- IBM Container Registry

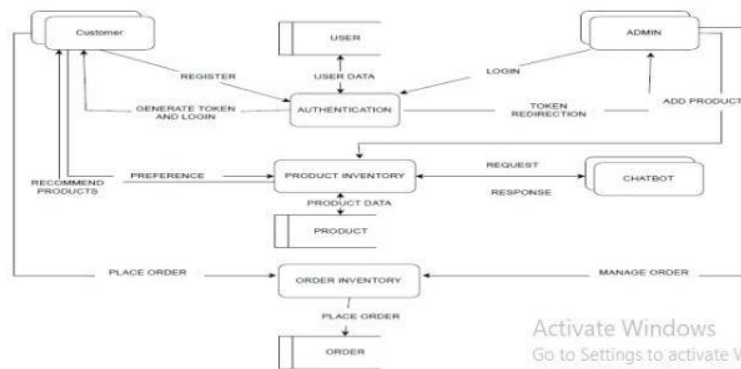
## 5.PROJECT DESIGN

### A. DATA FLOW DIAGRAM

#### Data Flow Diagrams:

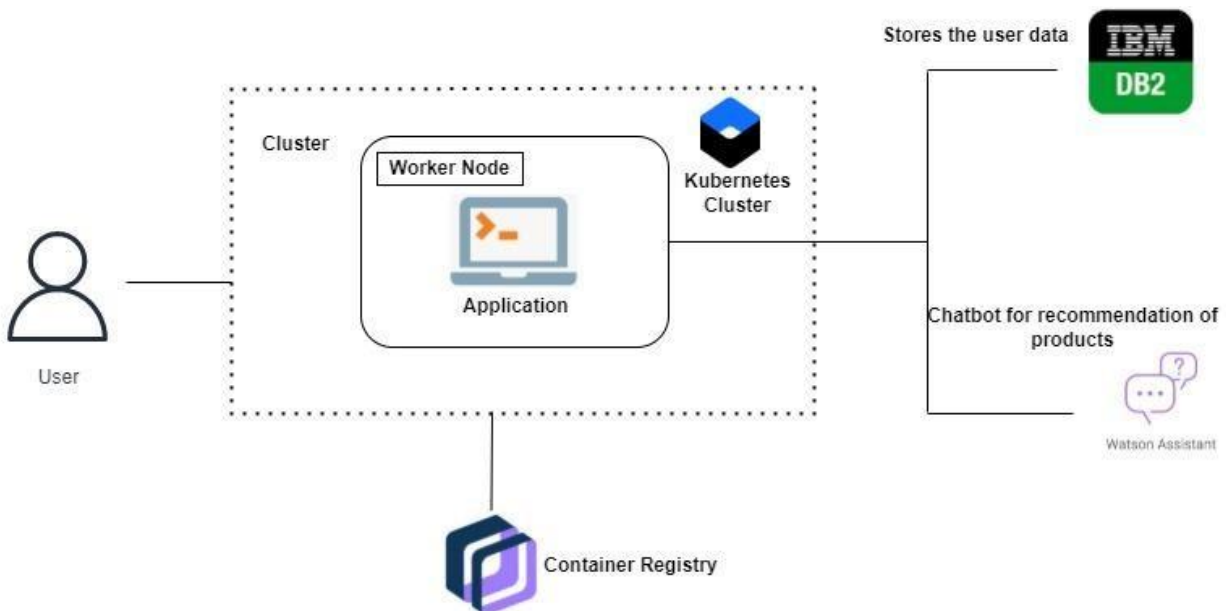
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and stored.

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



Activate Windows  
Go to Settings to activate Windows.

### B. TECHNICAL ARCHITECTURE



## C.USER STORIES

**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
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 and make purchases.	High	Sprint-1
	Dashboard					
Customer (Web 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
		USN-5	As a user, I can log into the application by entering email & password	I can access and make purchases.	High	Sprint-1
Administrator	Login	USN-1	I enter my mail and password on organisation's approval	I can approve products and purchases	High	Sprint-1 Administrator

## 6.SOLUTION REQUIREMENTS

### A. FUNCTIONAL REQUIREMENTS

#### Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Delivery confirmation	Confirmation via email and phone number
FR-4	Assistance	Bot is integrated with the application to make the usability simple

## B. NON-FUNCTIONAL REQUIREMENTS

### 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.

## 7.PROJECT PLANNING

### A. SPRINT PLANNING

#### Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User panel	USN-1	The user will login into the website and go through the products available on the website	2	High	M.Ilakkiya Shenbagam
Sprint-2	chatbot	USN-2	The user can directly talk to Chatbot regarding the products. Get the recommendations based on information provided by the user.	1	High	J.Jeya Roselin
Sprint-3	admin panel	USN-3	The role of the admin is to check out the database about the stock and have a track of all the things that the users are purchasing.	2	high	M.Saru Latha
Sprint-4	final Delivery	USN-4	Container of applications using docker kubernetes and deployment the application. Create the documentation and final submit the application	2	high	R.pirama Raji

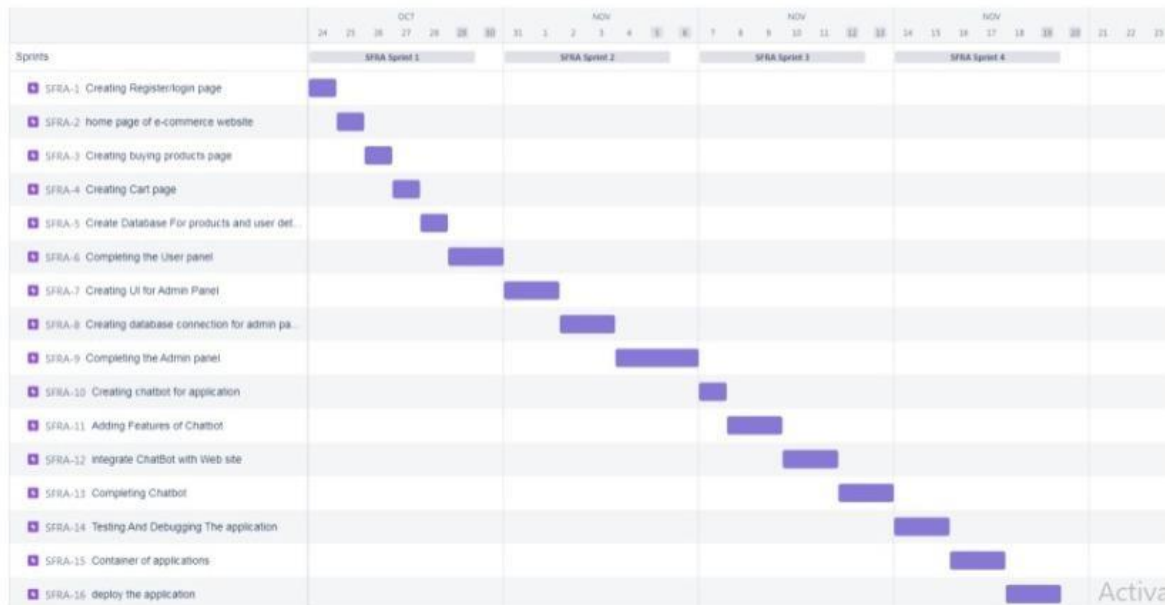
## B. SPRINT DELIVERY SCHEDULE

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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		05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		19 Nov 2022

## C. BURDOWN DIAGRAM

Burndown Chart:



## **8.CONCLUSION**

This paper presented an interdisciplinary, comprehensive review of what has been done in the field of conversational agents for fashion and retail e-commerce. Thus this study adds to specialized literature in the field of chatbot design for fashion e-commerce by providing a comprehensive map of chatbot approaches that can be deployed by retailers.

## **9.FUTURE SCOPE**

This project carries a broad prospective for future. In future, our project can be extended towards providing online payments for the users, feature to track the order and gps features to get the customer location.



## 10.SOURCE CODE

### Creating UI for application

#### Index.html

```
<!DOCTYPE html>

<html lang="en">


<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>RST/Home</title>

    <link rel="stylesheet"
href=https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css>

    <link rel="stylesheet" href=https://rakesh87.s3.us-
east.cloud-object-storage.appdomain.cloud/style.css>

</head>


<body>

    <section id="header">
```



<a href="#"><img src=[https://saravanan2001mohan.s3.us-east.cloud-object-storage.appdomain.cloud/rst%20\(1\).png](https://saravanan2001mohan.s3.us-east.cloud-object-storage.appdomain.cloud/rst%20(1).png)

Class="logo" alt="" width="150" height="60"></a>

<div>

<ul id="navbar">

<li><a class="active" href="index.html">Home</a></li>

<li><a href="login.html">Login</a></li>

<li><a href="product.html">Product</a></li>

<li><a href="about.html">About</a></li>

<li><a href="contact.html">Contact</a></li>

<li><a href="cart.html"><i class="fa fa-shopping-bag"></i></a></li>

<li><a href="register.html"><i class="fa fa-user-secret"></i></a></li>

</ul>

</section>

<section id="hero">

<h4>"Happiness Depends Upon Ourselves"</h4>

<h2>Smart Fashion Recommender</h2>

<h1>Application</h1>

```
<p>Dress Up In Confidence</p>
<button class="normal">Shop now</button>
</section>
<section id="feature" class="section-p1">
  <div class="fe-box">
    <img src=https://unik.s3.jp-tok.cloud-object-storage.appdomain.cloud/freeship.png alt="" width="100"
      Height="100">
    <h6>Free Shipping</h6>
  </div>
  <div class="fe-box">
    <img src=https://unik.s3.jp-tok.cloud-object-storage.appdomain.cloud/orderonline.jpg alt="" width="100"
      Height="100">
    <h6>Online Order</h6>
  </div>
  <div class="fe-box">
    <img src=https://unik.s3.jp-tok.cloud-object-storage.appdomain.cloud/savemoney.jpg alt="" width="100"
      Height="100">
    <h6>Save Money</h6>
```

</div>

<div class="fe-box">

<img src=<https://unik.s3.jp-tok.cloud-object-storage.appdomain.cloud/promotion.jpg> alt="" width="100"

Height="100">

<h6>Promotions</h6>

</div>

<div class="fe-box">

<img src=<https://unik.s3.jp-tok.cloud-object-storage.appdomain.cloud/happysell.jpg> alt="" width="100"

Height="100">

<h6>Happy Sell</h6>

</div>

<div class="fe-box">

<img src=<https://unik.s3.jp-tok.cloud-object-storage.appdomain.cloud/support.jpg> alt="" width="100"

Height="100">

<h6>F24/7 Support</h6>

</div>

<a href="#"><i class="fa fa-shopping-bag bag"></i></a>

```

</div>
</div>
</section>
<section id="sm-banner" class="section-p1">
<div class="banner-box">
<h4>Crazy Deals</h4>
<h2>Buy 1 get 1 free</h2>
<span>The best classic dress is on sale at U Nik</span>
<a href="products.html"><button class="white">Learn More
</button></a>
</div>
<div class="banner-box banner-box2">
<h4>spring/summer</h4>
<h2>upcommig seasons</h2>
<span>The best classic dress is on sale at U Nik</span>
<a href="blog.html"><button
class="white">Collection</button></a>
</div >
</section>
<section id="banner3">
<div class="banner-box">

```

```

<h2>SEASONAL SALE</h2>
<h3>Winter collection 50% off</h3>
</div >

<div class="banner-box banner-box2">
<h2>SEASONAL SALE</h2>
<h3>Winter collection 50% off</h3>
</div >

<div class="banner-box banner-box3">
<h2>SEASONAL SALE</h2>
<h3>Winter collection 50% off</h3>
</div >

</section> →

<section id="newsletter" class="section-p1">
  <div class="newstext">
    <h4>Sign up for NewsLetters</h4>
    <p>Get Email updates about our latest shop and
<span>special offer</span>
    </p>
  </div>
  <div class="form">

```

```

        <input type="text" placeholder="Your E-mail Address">
        <button class="normal">login</button>
    </div>
</section>
<footer class="section-p1">
    <div class="col">
        <img class="logo"
            Src=https://saravanan2001mohan.s3.us-east.cloud-object-storage.appdomain.cloud/rst%20\(1\).png alt=""
            Width="200" height="100">
        <h4>Contact</h4>
        <p><strong>E-Mail:
</strong><a href="mailto:rstsmartfashion@gmail.com">rstsmartfashion@gmail.com</a>
        <p><strong>Phone: </strong>1234567890</p>
        <div class="follow">
            <h4>Follow us</h4>
            <div class="icon">
                <i class="fa fa-facebook-f"></i>
                <i class="fa fa-twitter"></i>
                <i class="fa fa-instagram"></i>
                <i class="fa fa-pinterest-p"></i>

```

```

        <i class="fa fa-youtube"></i>
      </div>
    </div>
  </div>
<div class="col">

```

```
<script>
```

```

  Window.watsonAssistantChatOptions = {
    integrationID: "f575ccb9-7f1a-4ceb-b32b-23f9d2a71c7a", //
    The ID of this integration.

    Region: "au-syd", // The region your integration is hosted in.

    serviceInstanceID: "c8db2a25-708c-4bac-bb36-
    d9b5b790d6d3", // The ID of your service instance.

    onLoad: function(instance) { instance.render(); }
  };

  setTimeout(function(){
    const t=document.createElement('script');

    t.src=https://web-
    chat.global.assistant.watson.appdomain.cloud/versions/ +
    (window.watsonAssistantChatOptions.clientVersion || 'latest')
    + "/WatsonAssistantChatEntry.js";

    document.head.appendChild(t);

```

```
});  
</script>  
</body>  
  
</html>
```

## Login.html

```
<!DOCTYPE html>  
<!--Created By CodingNepal →  
<html lang="en" dir="ltr">  
  <head>  
    <meta charset="utf-8">  
    <title>Login and Registration Form in HTML |  
CodingNepal</title>  
    <link rel="stylesheet" href="style.css">  
    <meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
  </head>  
  <body>  
    <div class="wrapper">
```



```

<div class="title-text">
  <div class="title login">
    Login
  </div>
  <div class="title signup">
    Signup
  </div>
</div>
<div class="form-container">
  <div class="slide-controls">
    <input type="radio" name="slide" id="login" checked>
    <input type="radio" name="slide" id="signup">
    <label for="login" class="slide login">Login</label>
    <label for="signup" class="slide
signup">Signup</label>
  <div class="slider-tab"></div>
</div>
<div class="form-inner">
  <form action="#" class="login">
    <div class="field">

```

```

        <input type="text" placeholder="email-id"
required>
    </div>
    <div class="field">
        <input type="password" placeholder="password"
required>
    </div>
    <div class="pass-link">
        <a href="forgot.html">Forgot password?</a>
    </div>
    <div class="field btn">
        <div class="btn-layer"></div>
        <input type="submit" value="Login">
    </div>
    <div class="signup-link">
        Not a member? <a href="">Signup now</a>
    </div>
</form>
<form action="#" class="signup">
    <div class="field">

```

```

        <input type="text" placeholder="email-id"
required>
    </div>
    <div class="field">
        <input type="password" placeholder="password"
required>
    </div>
    <div class="field">
        <input type="password" placeholder="Confirm
password" required>
    </div>
    <div class="field btn">
        <div class="btn-layer"></div>
        <input type="submit" value="Signup">
    </div>
</form>
</div>
</div>
<script>

```

```
    Const loginText = document.querySelector(".title-text
.login");

    Const loginForm = document.querySelector("form.login");

    Const loginBtn = document.querySelector("label.login");

    Const signupBtn =
document.querySelector("label.signup");

    Const signupLink = document.querySelector("form
.signup-link a");

    signupBtn.onclick = (()=>{
        loginForm.style.marginLeft = "-50%";
        loginText.style.marginLeft = "-50%";
    });

    loginBtn.onclick = (()=>{
        loginForm.style.marginLeft = "0%";
        loginText.style.marginLeft = "0%";
    });

    signupLink.onclick = (()=>{
        signupBtn.click();
        return false;
    });
</script>
```

```
</body>
</html>
```

## Product.html

```
<div class="pro">
<img src=https://s3.jp-tok.cloud-object-storage.appdomain.cloud/unik/shoe1.jpg height="400" alt="">
<div class="des">
<span>Bata</span>
<h5>Brown leather shoe</h5>
<div class="star">
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
```

</div>

<h4>₹8000</h4>

</div>

<a href="#"><i class="fa fa-shopping-bag bag"></i></a>

</div>

<div class="pro">

<img src=<https://s3.jp-tok.cloud-object-storage.appdomain.cloud/unik/shoe2.jpg> height="400" alt="">

<div class="des">

<span>Nike</span>

<h5>Sandle High Heel</h5>

<div class="star">

<i class="fa fa-star"></i>

<i class="fa fa-star"></i>

<i class="fa fa-star"></i>

<i class="fa fa-star"></i>

<i class="fa fa-star"></i>

</div>

<h4>₹6600</h4>

</div>

```

<a href="#"><i class="fa fa-shopping-bag bag"></i></a>
</div>
<div class="pro">
<img src=https://s3.jp-tok.cloud-object-storage.appdomain.cloud/unik/shoe3.jpg height="400" alt="">
<div class="des">
<span>Walkaroo</span>
<h5>Brown High Heel</h5>
<div class="star">
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
</div>
<h4>₹7800</h4>
</div>
<a href="#"><i class="fa fa-shopping-bag bag"></i></a>
</div>
<div class="pro">

```

```

<img src=https://s3.jp-tok.cloud-object-storage.appdomain.cloud/unik/tops.jpg alt="">
<div class="des">
<span>Zara</span>
<h5>Black Smile T-Shirts</h5>
<div class="star">
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
</div>
<h4>₹900</h4>
</div>
<a href="#"><i class="fa fa-shopping-bag bag"></i></a>
</div>
<div class="pro">
<img src=https://s3.jp-tok.cloud-object-storage.appdomain.cloud/unik/tops1.jpg alt="">
<div class="des">
<span>Zara</span>

```



```

<h5>Pink dog T-Shirts</h5>
<div class="star">
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
</div>
<h4>₹900</h4>
</div>
<a href="#"><i class="fa fa-shopping-bag bag"></i></a>
</div>
<div class="pro">
<img src=https://s3.jp-tok.cloud-object-storage.appdomain.cloud/unik/tops2.jpg alt="">
<div class="des">
<span>Zara</span>
<h5>Black and White Trendy Shirt</h5>
<div class="star">
<i class="fa fa-star"></i>

```

```
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
</div>
<h4>₹1500</h4>
</div>
<a href="#"><i class="fa fa-shopping-bag bag"></i></a>
</div>
<div class="pro">
<img src=https://s3.jp-tok.cloud-object-storage.appdomain.cloud/unik/tops3.jpg alt="">
<div class="des">
<span>Zara</span>
<h5>Green Panda Shirt</h5>
<div class="star">
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
<i class="fa fa-star"></i>
```

```

<i class="fa fa-star"></i>
</div>
<h4>₹600</h4>
</div>
<!-- <a href="#"><i class="fa fa-shopping-bag bag"></i></a>
</div>
</div>
</section>
<section id="sm-banner" class="section-p1">
<div class="banner-box">
<h4>Crazy Deals</h4>
<h2>Buy 1 get 1 free</h2>
<span>The best classic dress is on sale at U Nik</span>
<a href="products.html"><button class="white">Learn More
</button></a>
</div>
<div class="banner-box banner-box2">
<h4>spring/summer</h4>
<h2>upcommig seasons</h2>
<span>The best classic dress is on sale at U Nik</span>

```

```

<a href="blog.html"><button
class="white">Collection</button></a>

</div >

</section>

<section id="banner3">
<div class="banner-box">
<h2>SEASONAL SALE</h2>
<h3>Winter collection 50% off</h3>
</div >

<div class="banner-box banner-box2">
<h2>SEASONAL SALE</h2>
<h3>Winter collection 50% off</h3>
</div >

<div class="banner-box banner-box3">
<h2>SEASONAL SALE</h2>
<h3>Winter collection 50% off</h3>
</div > →

<!-- </section> → →

    <!-- <section id="newsletter" class="section-p1">
        <div class="newstext">

```

```

    <h4>Sign up for NewsLetters</h4>
    <p>Get Email updates about our latest shop and
<span>special offer</span>
    </p>
</div>
<div class="form">
    <input type="text" placeholder="Your E-mail Address">
    <button class="normal">login</button>
</div>
</section> →
<!-- <footer class="section-p1">
    <div class="col">
        <img class="logo"
            Src=https://saravanan2001mohan.s3.us-east.cloud-object-storage.appdomain.cloud/rst%20\(1\).png alt=""
            Width="200" height="100">
        <h4>Contact</h4>
        <p><strong>E-Mail:
</strong><a href="mailto:rstsmartfashion@gmail.com">rstsmartfashion@gmail.com</a>
        <p><strong>Phone: </strong>1234567890</p>
        <div class="follow">

```

```

<h4>Follow us</h4>
<div class="icon">
    <i class="fa fa-facebook-f"></i>
    <i class="fa fa-twitter"></i>
    <i class="fa fa-instagram"></i>
    <i class="fa fa-pinterest-p"></i>
    <i class="fa fa-youtube"></i>
</div>
</div>
</div> →
<!-- <div class="col"> →
    <!-- <h4>About</h4>
    <a href="#">About us</a>
    <a href="#">Delivery Information</a>
    <a href="#">Privacy Policy</a>
    <a href="#">Terms & Conditions</a>
    <a href="#">Contact us</a>
</div>
<div class="col">
<h4>My Account</h4>

```

```

<a href="#">Sign In</a>
<a href="#">View Cart</a>
<a href="#">My Wishlist</a>
<a href="#">Track my order</a>
<a href="#">Help</a>
</div>
</footer>
<div class="copyright">
<center><p>© 2022, PNT2022TMI36870 – Smart Fashion
Recommender Application </p></center>
</div> →
<!-- <script>
    Window.watsonAssistantChatOptions = {
        integrationID: "f575ccb9-7f1a-4ceb-b32b-23f9d2a71c7a", //
The ID of this integration.
        Region: "au-syd", // The region your integration is hosted in.
        serviceInstanceID: "c8db2a25-708c-4bac-bb36-
d9b5b790d6d3", // The ID of your service instance.
        onLoad: function(instance) { instance.render(); }
    };
    setTimeout(function(){

```

```

const t=document.createElement('script');

t.src=https://web-
chat.global.assistant.watson.appdomain.cloud/versions/ +
(window.watsonAssistantChatOptions.clientVersion || 'latest')
+ "/WatsonAssistantChatEntry.js";

document.head.appendChild(t);

});
</script> →
</body></html>

```

## **Sendmail.py**

```

Import smtplib
Import sendgrid
Import os

From sendgrid.helpers.mail import Mail, Email, To, Content
SUBJECT = "Interview Call"

S = smtplib.SMTP('smtp.gmail.com', 587)

Def sendmail(TEXT,email):

    Print("sorry we cant process your candidature")

S = smtplib.SMTP('smtp.gmail.com', 587)

```



```

s.starttls()
s.login(il.pradeepthi@gmail.com, "oms@1Ram")
message = 'Subject: {} \n \n {}'.format(SUBJECT, TEXT)
s.sendmail(il.pradeepthi@gmail.com, email, message)
s.quit()

def sendgridmail(user,TEXT):

    sg =
    sendgrid.SendGridAPIClient('SG.nouVVZMwQTSYtih73r1TxQ.3H
    OkajWkEYpo0RV1iarxSVKbqvtjyZ_nhPbKi3zeZnc')

    from_email = Email(sandeep@thesmartbridge.com) # Change
    to your verified sender

    to_email = To(user) # Change to your recipient

    subject = "Sending with SendGrid is Fun"

    content = Content("text/plain",TEXT)

    mail = Mail(from_email, to_email, subject, content)

    # Get a JSON-ready representation of the Mail object

    Mail_json = mail.get()

    # Send an HTTP POST request to /mail/send

    Response = sg.client.mail.send.post(request_body=mail_json)

    Print(response.status_code)

    Print(response.headers)

```

## Deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: jobportal

spec:

replicas: 1

selector:

matchLabels:

app: flasknode

template:

metadata:

labels:

app: flasknode

spec:

containers:

Name: flasknode

Image: icr.io/helloworld1/newhelloworld

imagePullPolicy: Always

ports:

containerPort: 5000

## **Service.yaml**

apiVersion: v1

kind: Service

metadata:

name: flask-node-deployment

spec:

ports:

-port: 5000

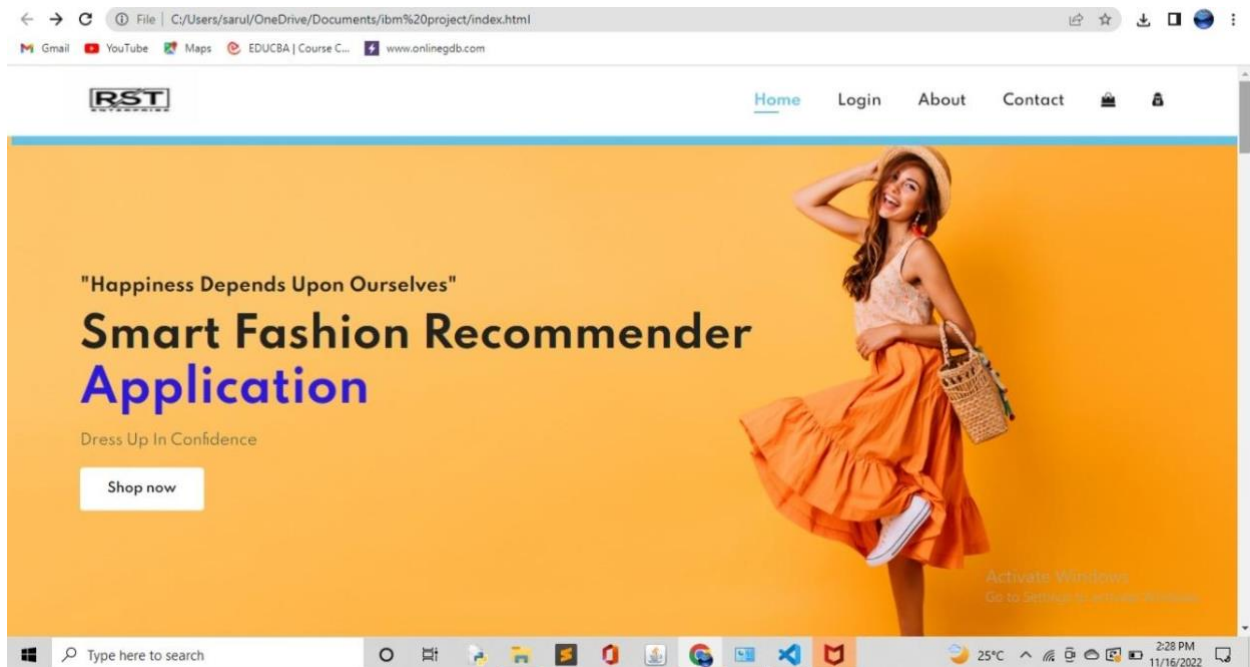
targetPort: 5000

selector:

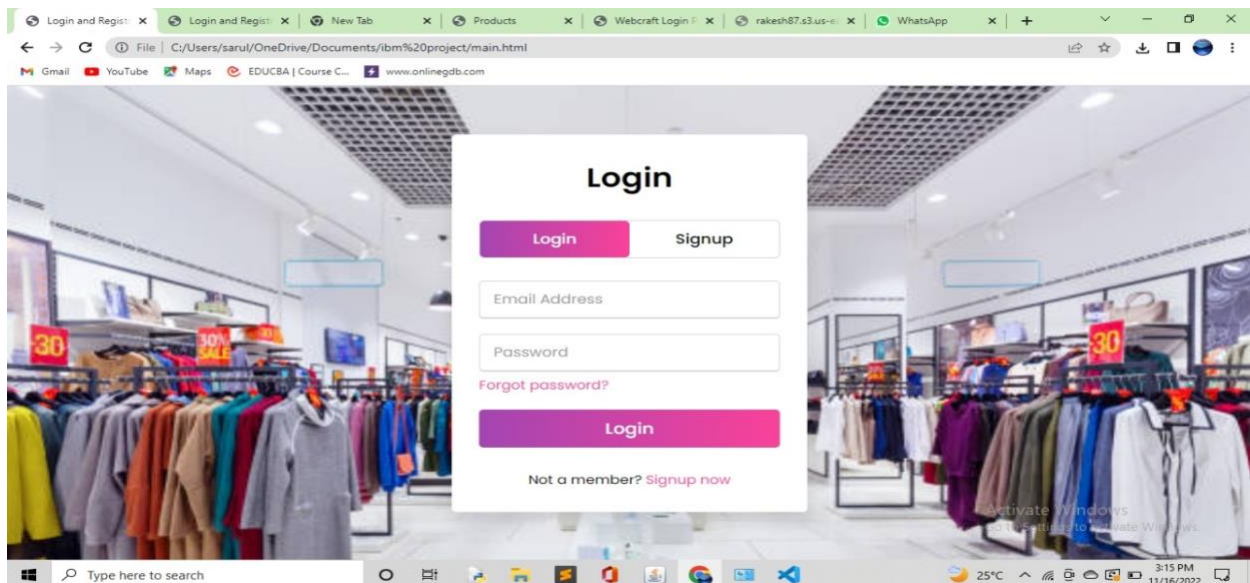
app: flasknode

## 11.OUTPUT IMAGES

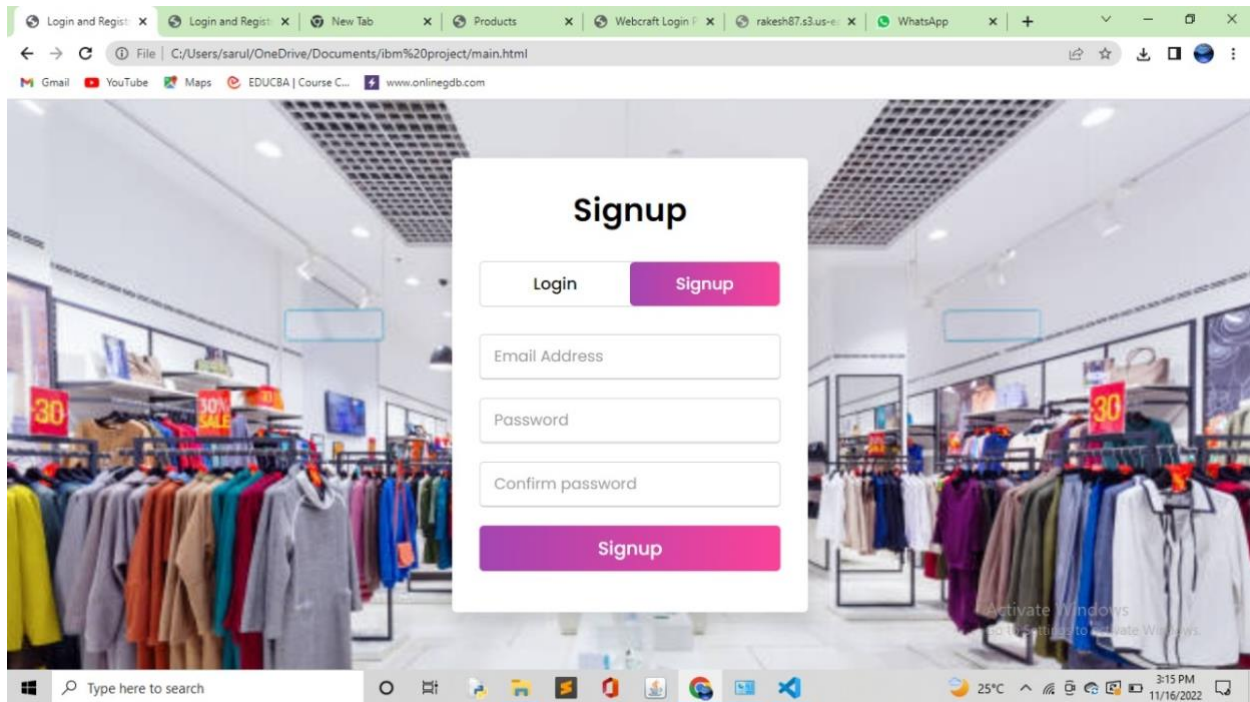
### Home page



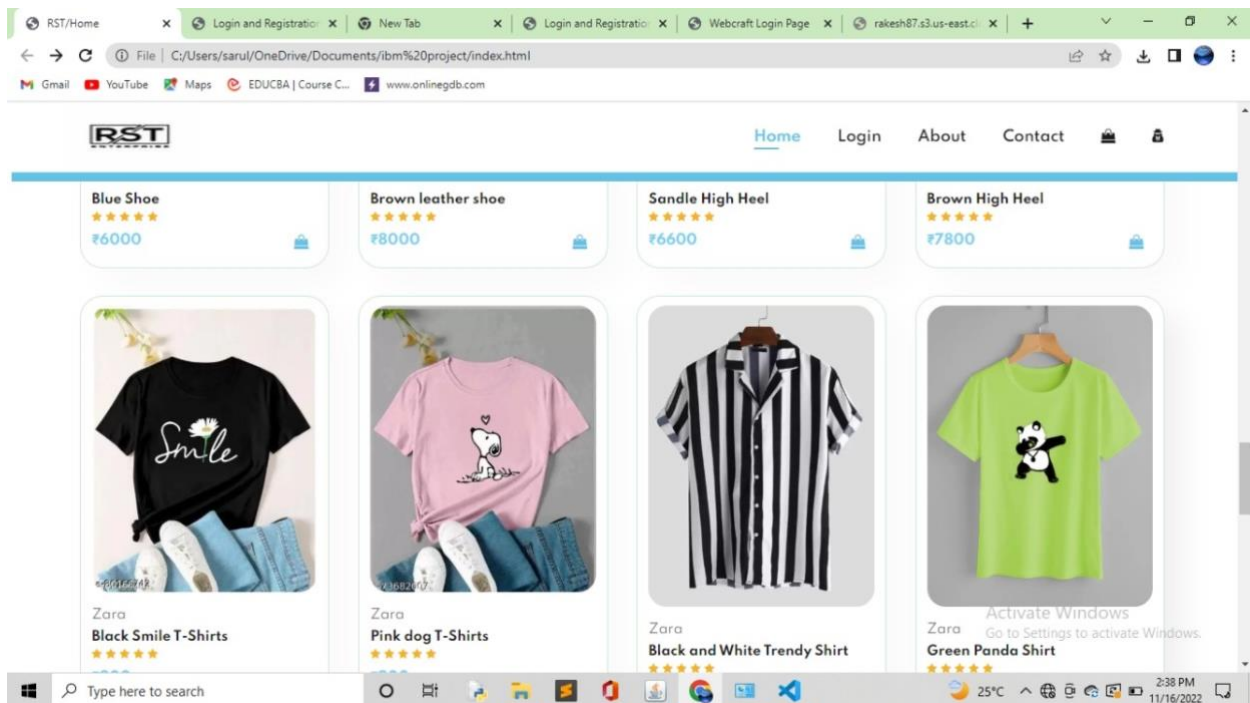
### Login page



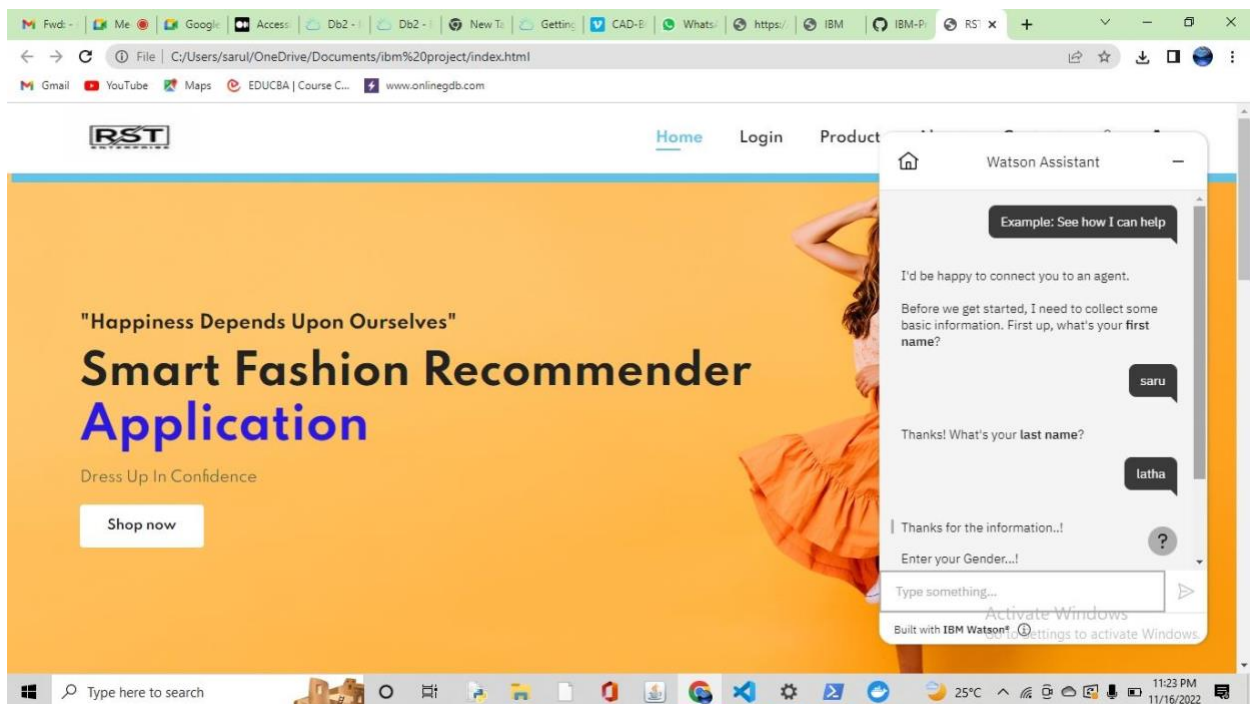
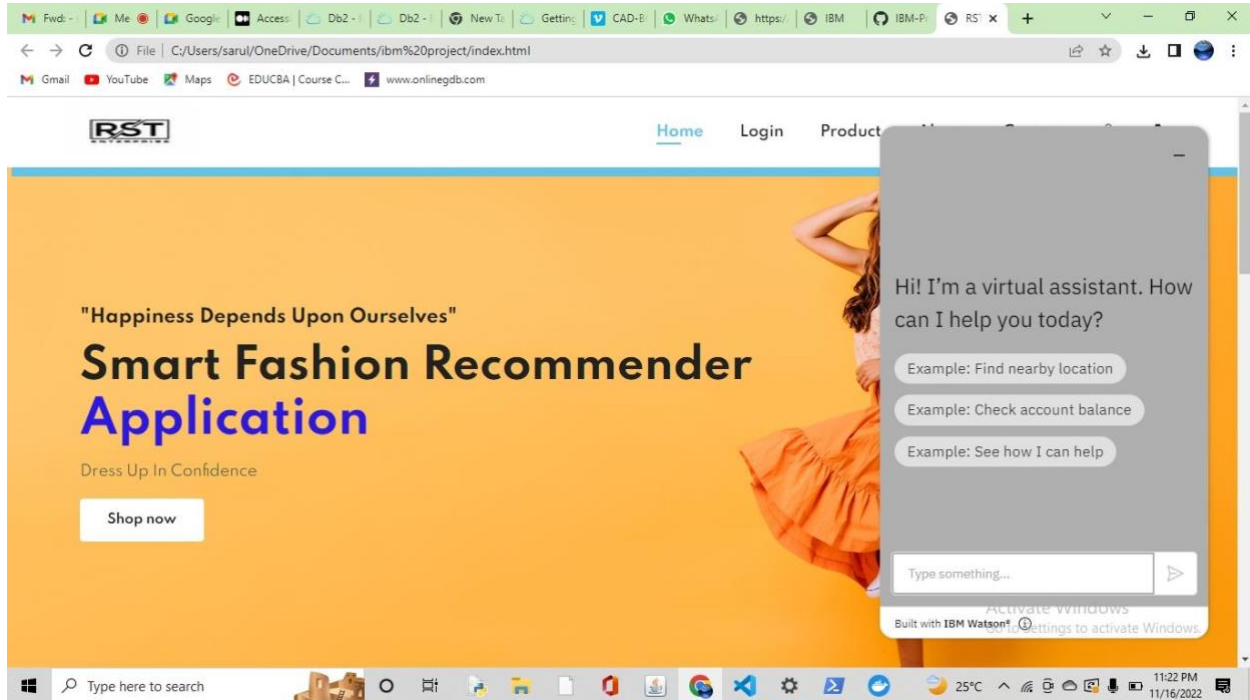
## Sign up page



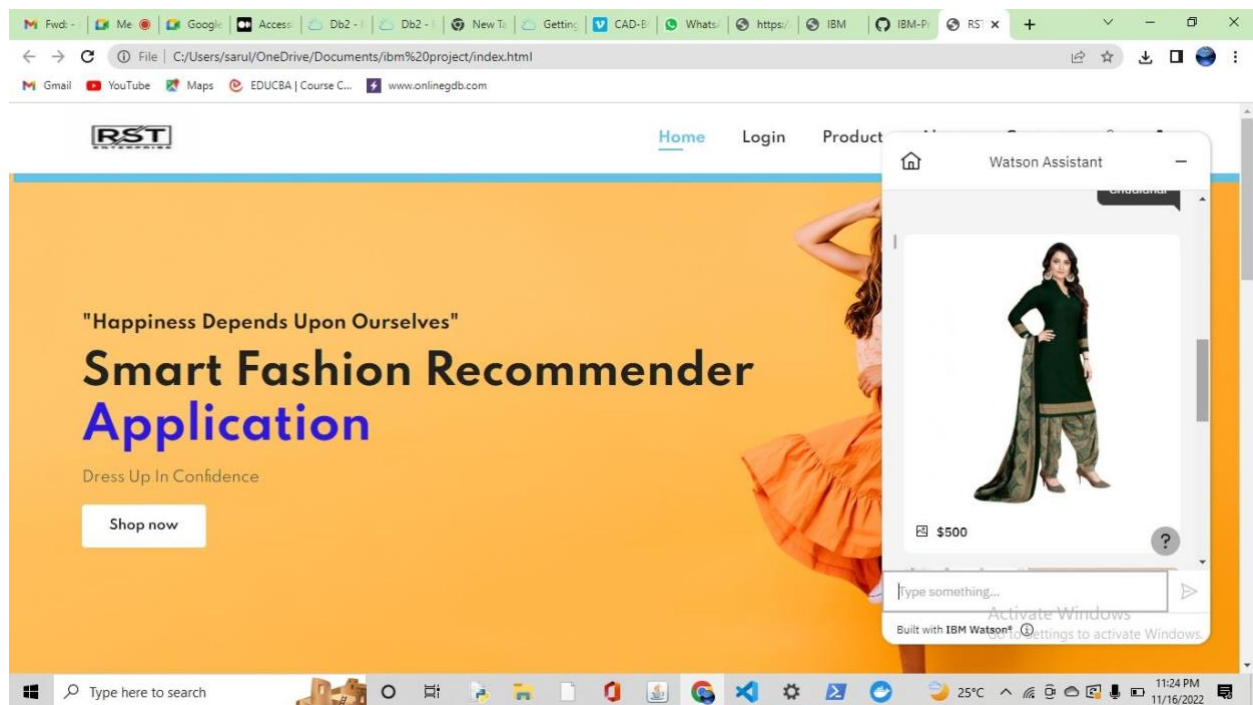
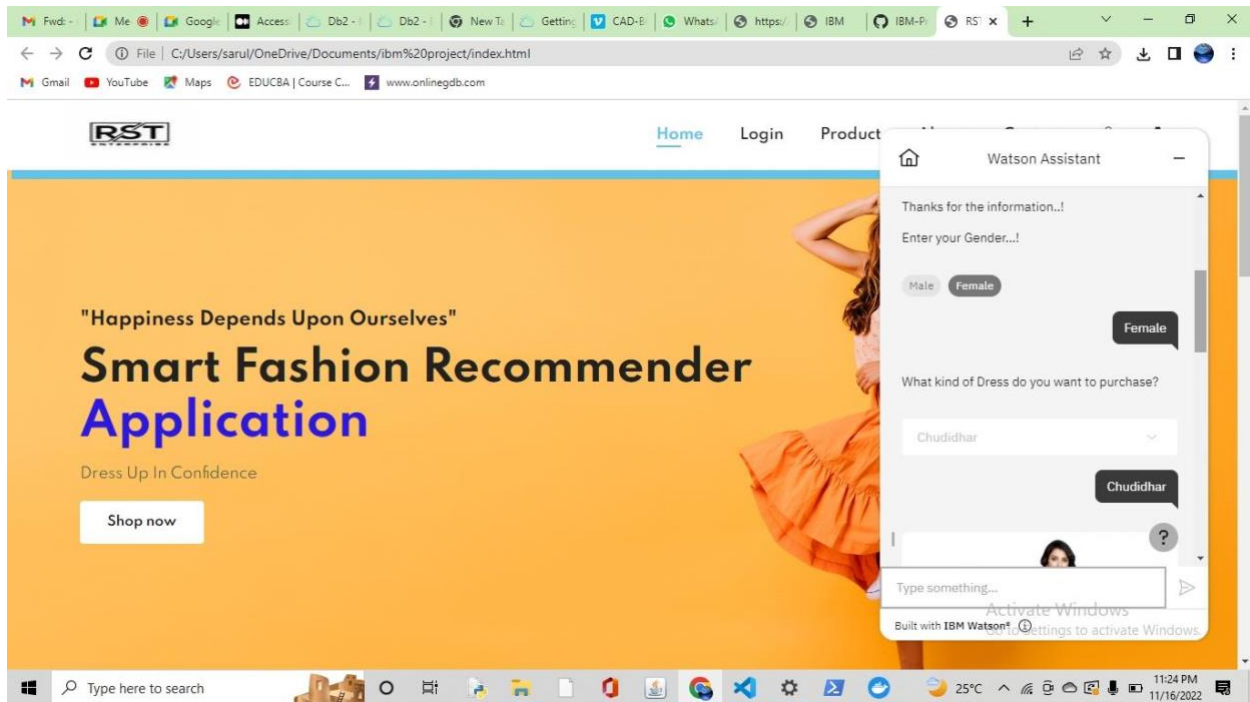
## Products

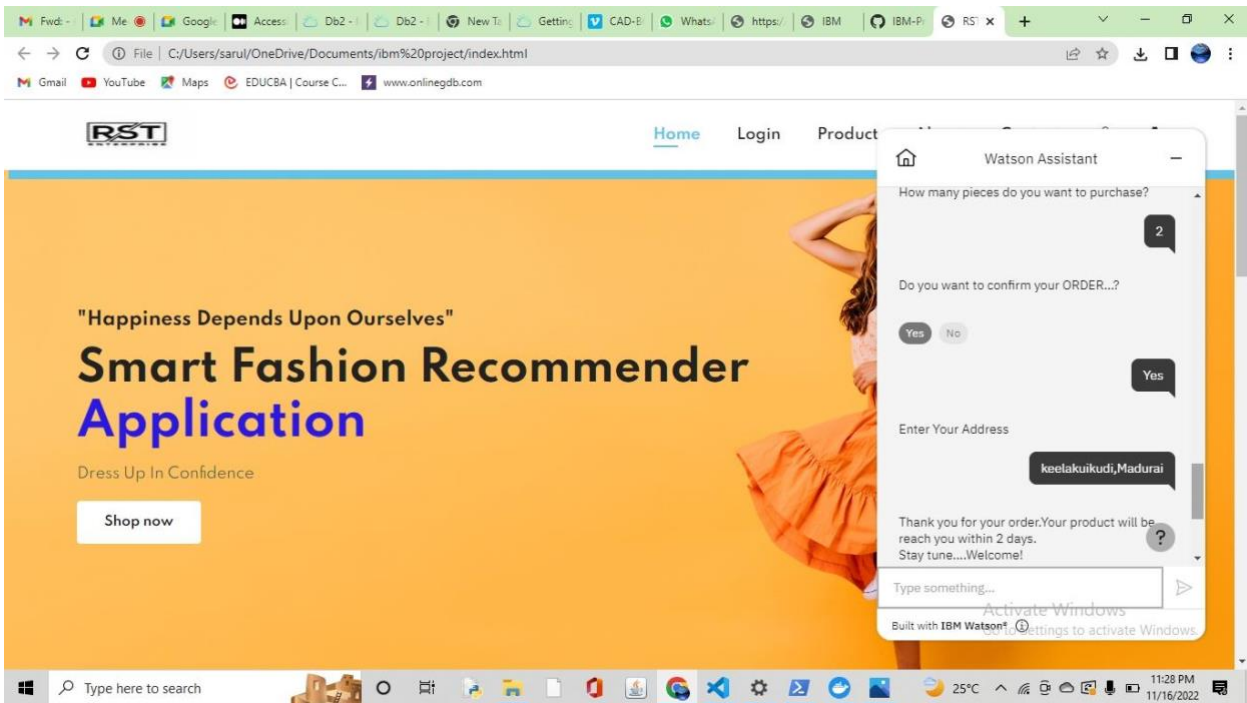
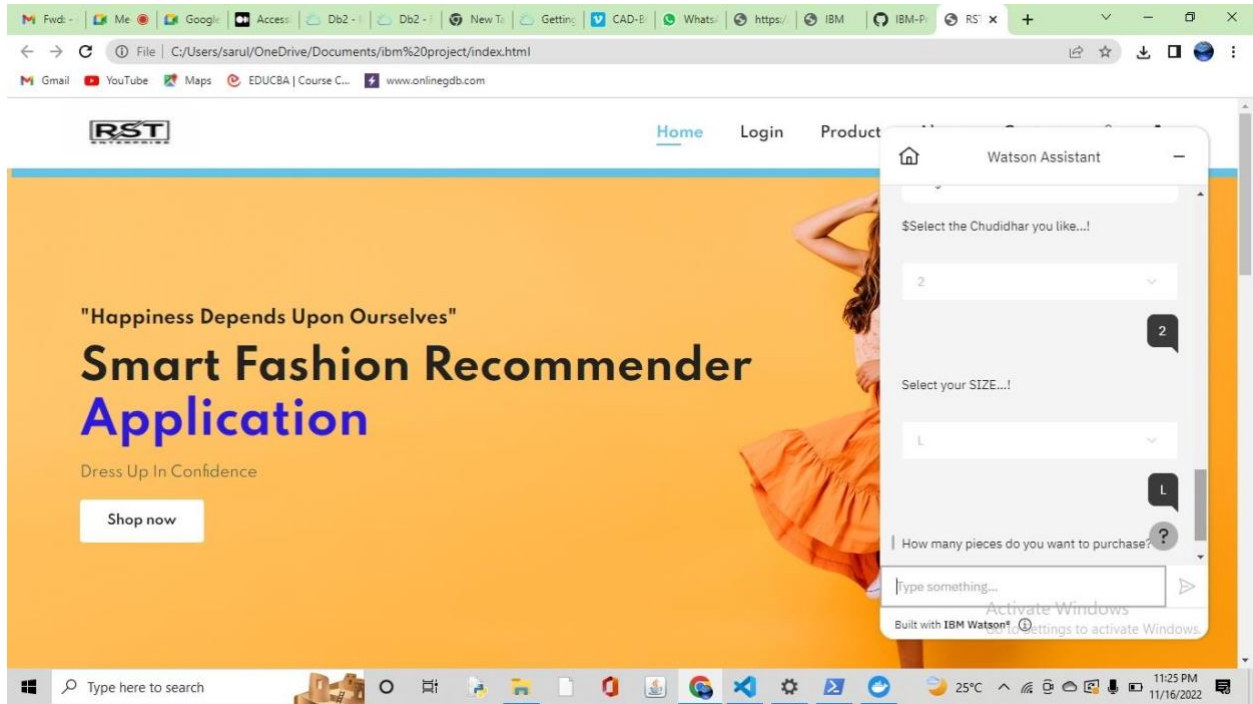


# Chatbot











## **12.GITHUB LINK**

<https://github.com/IBM-EPBL/IBM-Project-6737-1658835493>