

SMART FASHION RECOMMENDER

Abstract:

Fashion applications have seen tremendous growth and are now one of the most used programs in the e-commerce field. The needs of people are continuously evolving, creating room for innovation among the applications. One of the tedious processes and presumably the main activities is choosing what you want to wear. Having an AI program that understands the algorithm of a specific application can be of great aid. We are implementing such a chat bot, which is fed with the knowledge of the application's algorithm and helps the user completely from finding their needs to processing the payment and initiating delivery. It works as an advanced filter search that can bring the user what they want with the help of pictorial and named representation. The application also has two main user interfaces - the user and the admin. The users can interact with the chat bot, search for products, order them from the manufacturer or distributor, make payment transactions, track the delivery, and so on. The admin interface enables the user to upload products, find how many products have been bought, supervise the stock availability and interact with the buyer regarding the product as reviews.

Introduction

Problem Statement

In E-commerce websites, users need to search for products and navigate across screens to view the product, add them to the cart, and order products. The smart fashion recommender application leverages the use of a chat bot to interact with the users, gather information about their preferences, and recommend suitable products to the users. This application has two predefined roles assigned to the users. The roles are customer and admin. The application demands redirection of the user to the appropriate dashboard based on the assigned role. Admin should be able to track the number of different products and admin should be assigned the responsibility to create products with appropriate categories. The user should be able to mention their preferences using interacting with chat bots. The user must receive a notification on order confirmation/failure. The chat bot must gather feedback from the user at the end of order confirmation. The main objective of this application is to provide better interactivity with the user and to reduce navigating pages to find appropriate products.

Project Description

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 chat bot. In this project you will be working on two modules:

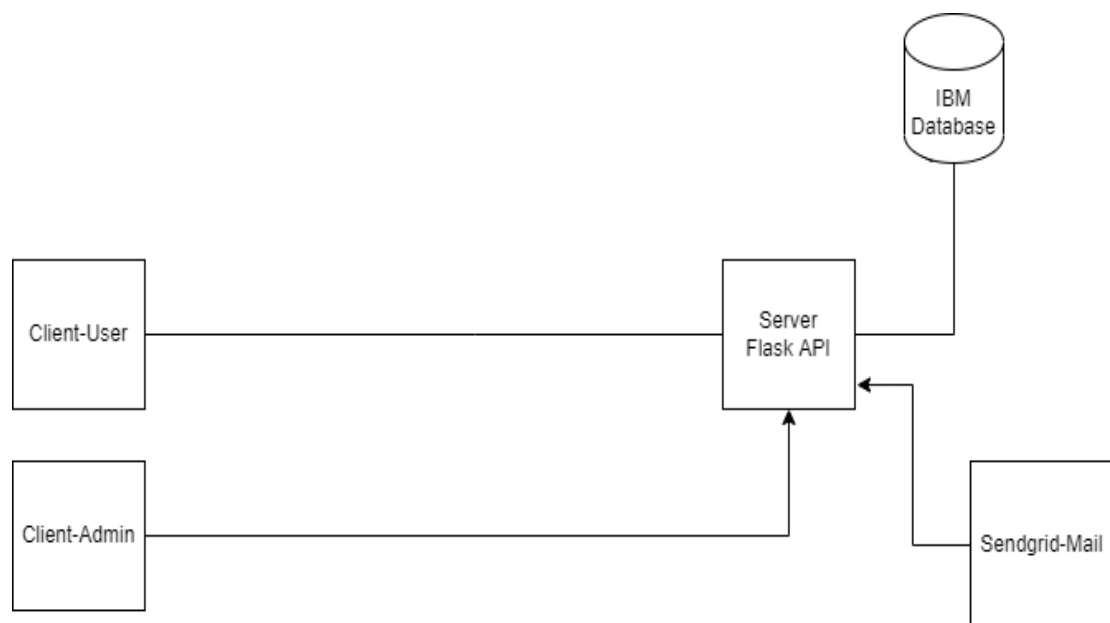
- Admin and
- User

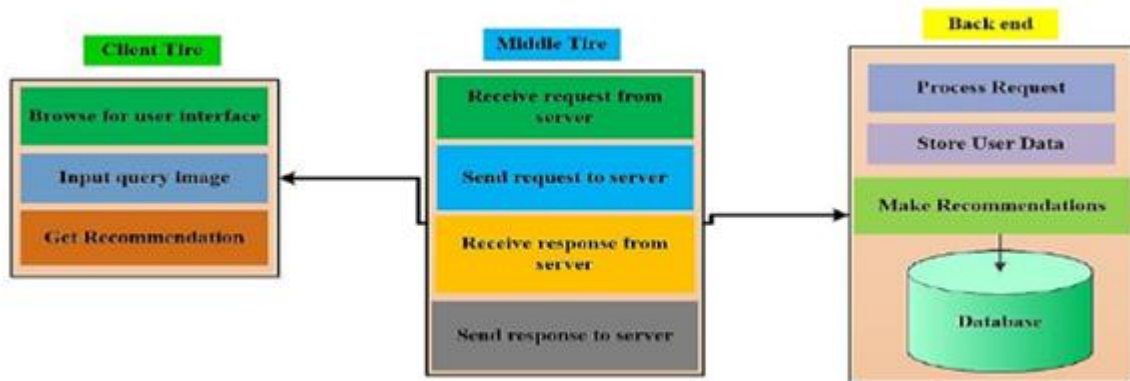
The methodology used in this solution

Instead of searching products in the search bar and navigating to individual products to find required preferences, this project leverages the use of chat bots 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 chat bot 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.

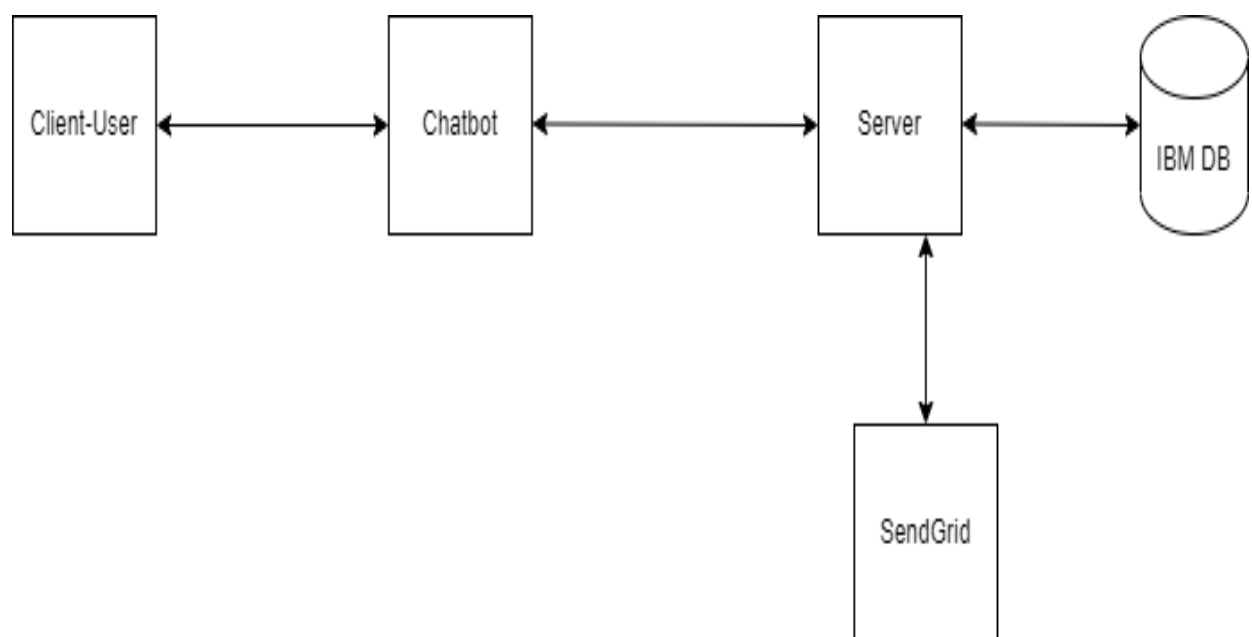
System Architecture:

Overall Architecture:

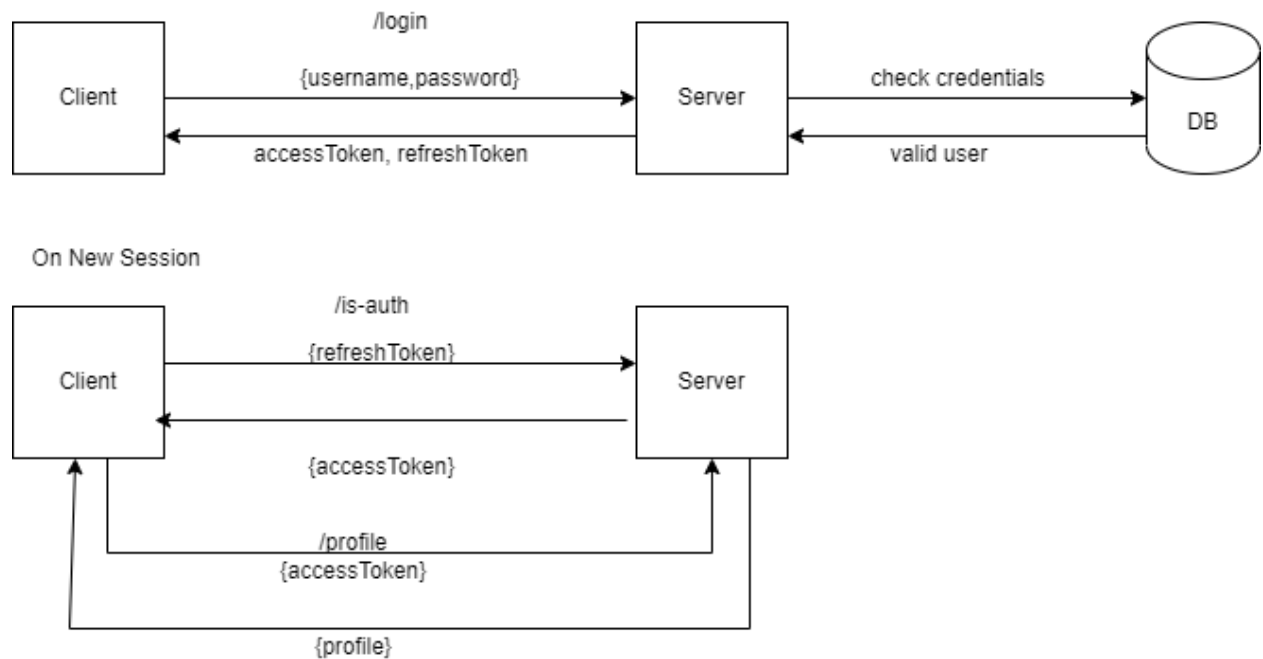




Customer



Authentication



Features:

- Using chat bot we can manage users' choices and orders.
- The chat bot can give recommendations to users based on their interests.
- It can promote the best deals and offers on that day.
- It will store the customer's details and orders in the database.
- The chat bot will send a notification to customers if the order is confirmed.
- Chat bots can also help in collecting customer feedback.

METHODOLOGY PROPOSED:

In traditional e-commerce websites, the users need to search for their required product using a search bar or go through the whole effects of their search. It will take a lot of users' time and it will create a lot of flawed user experiments. This approach will create bad marketing for the product. Later

when the user comes again to purchase the product it will create a bad impression on the user. Even though the product is good the user will not buy the product.

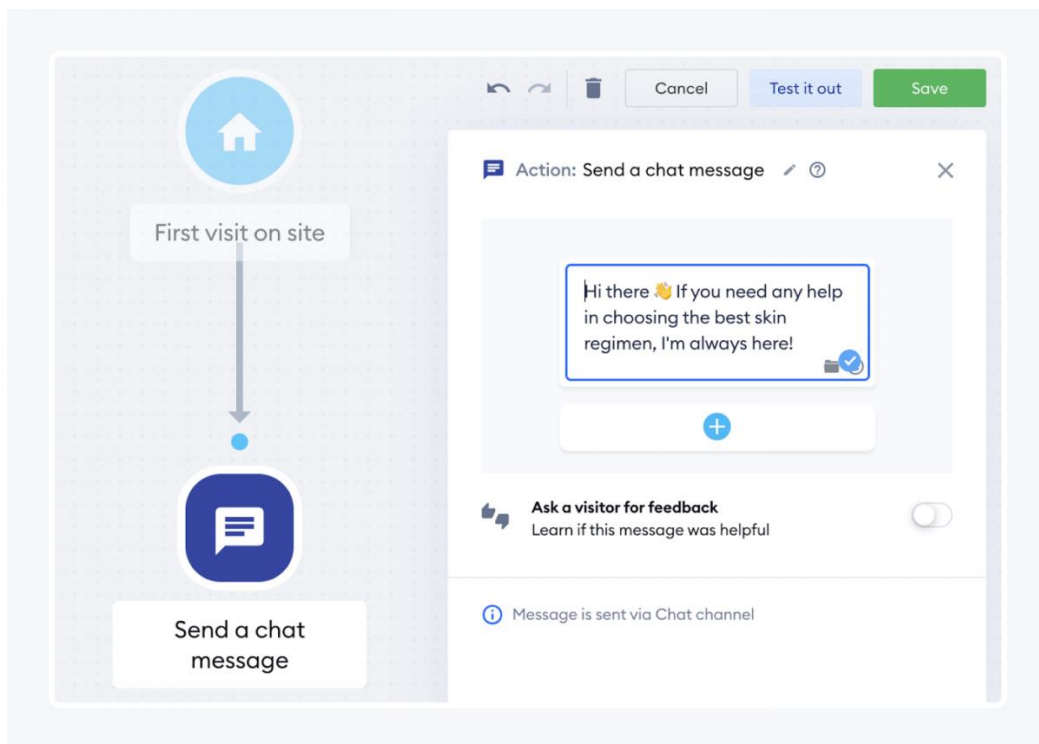
This type of search will create miss matched products when the product has a different name. Let's say we search for oranges on amazon. Sometimes it will show orange colour or sometimes it will show orange fruit. In recent times, fashion systems have been integrated with artificial intelligence and deep learning. These approaches provide a rich recommendation, but in most cases, it is prone to product mismatch. Even Though the recommender system recommends products based on the user's preference, this system lacks a chat bot that improves user experience by interacting with users. In most fashion systems, the user needs to navigate across multiple products to find the appropriate product. The users are made to filter products based on the long list of categories present in the system. If this system is integrated with an intelligent bot, it would be able to list out only required categories.

Results and discussions:

The smart fashion recommender system should recommend products based on user input. The user input can be any of the following

- Category
- Subcategory
- Product name
- Product brand
- Colour/Size
- Price Range
- Delivery Date/Time
- Discounts/Deals
- New Arrivals

Based on the user input, chat bot displays products sorted by required fields.



Functional Requirements:

The functional requirements of the application are:

- Redirect users to their respective dashboards based on their roles such as admin and customer
- Allow admin to track sales of individual products
- Allow admin to manage orders made by a particular customer.

- Allow users to interact with the chat bot.
- Manage users' choices and charges using the chat bot.
- Promote the best deals and offers.
- Store customer details and orders.
- Send Notifications to customers if the order is confirmed.
- Collect user feedback.
- Recommend products based on user preference.
- Enable online payment features.
- Generate reports for order summary and order histories.

Non-Functional Requirements

Performance Requirements

The system shall be able to handle multiple requests at any given point in time and generate an appropriate response.

- The response should not take longer than 5 seconds to appear on the client side.
- The client application should lazy load images of the product to minimize network calls over the network.
- The responses from the server should be cached on the client side.

Security Requirements

- Credentials and secrets should be stored securely and should not be leaked.
- Secured connection HTTPS should be established for transmitting requests and responses between client and server.
- The system has different roles assigned to a user and every user has access constraints.
- User access token should be valid for a shorter period and needs to be refreshed periodically.
- Clients should implement mechanisms to prevent XSS attacks.
- The server should restrict access to the resources for the particular client domain.

Error Handling

- The system should handle expected as well as unexpected errors and exceptions to avoid termination of the program.
- Appropriate error messages should be generated and displayed to the client.

Hardware Requirements

- 8GB RAM
- Intel Core i3
- Laptop/Desktop
- Windows/MAC/Linux OS.

Software Requirements

- Python
- Flask
- Docker
- Kubernetes
- IBM DB

Conclusion

The smart fashion recommender system uses a chat bot as a primary mechanism to interact with users, collect user interest and recommend products periodically. A chat bot is designed to improve user experience by interacting with users. Users need not navigate between multiple pages to find an appropriate product. The system is designed to minimize the efforts taken by customers to search for

the required product. The future enhancements of the chat bot include adding products to the cart, displaying cart items, order history, and payment through the chat bot.

References:

- Samit Chakraborty, Md. Saiful Hoque, Naimur Rahman Jeem-Fashion Recommendation Systems, Models and Methods: A Review
- Samit Chakraborty, Md. Saiful Hoque, S.M. Surid- A Comprehensive review on image-based style prediction and online fashion recommendation
- Qingqing Tu, Le Dong-An Intelligent Personalized Fashion Recommendation System
- M Sridevi, N ManikyaArun, M Shashikala, and E Sudarshan-Personalized fashion recommender system with image-based neural networks