PROJECT REPORT DOCUMENT

Date	15 November 2023
Team ID	592046
Project Name	Restaurant recommendation system

INTRODUCTION:

Project overview:

Our recommender system adopts a user preference model by using features of visited/likedrestaurants by the user and takes relevant context factors, e.g., time of the day, location, currentweather, and restaurant ratings, into account. The final case study for Context Attributes andUsers show that the proposed restaurant recommendation algorithm can effectively utilizethe user's culinary preference and related context factors to generate robust recommendationand recommend personalized restaurants for different users.

Exploiting sentiment analysis has become popular in designing recommender systems in various fields,including the restaurant and food area. However, most of the sentiment analysis-based restaurant recommendersystems only use static information such as food quality, price, and service quality. The analysis of users'opinions and the extraction of their food preferences lead to the provision of personalized recommendations,which is a research gap in literature; In this paper, a context-aware recommender system is proposed thatextracts the food preferences of individuals from their comments and suggests restaurants in accordance withthese preferences. For this

purpose, the semantic approach is used to cluster the name of foods extracted fromusers' comments and analyze their sentiments about them. Finally, nearby open restaurants are recommendedbased on their similarity to user preferences.

Purpose:

In this age of information, people use a variety of approaches and techniques to help them decide what to buy, where to eat, how to spend their leisure time. Recommender Systems help to automate some of these decision-making strategies by providing high quality and personalized recommendations, which is unique to every personusing the system.Recommender Systems are commonly known as software tools and techniques utilized to suggest certain items to users in a predictive manner. The suggestions may relate to variousdecision-making processes that the users might be interested in.Recommender Systems are primarily directed towards individuals who lack sufficient personal experience or competence to evaluate thepotentially overwhelming number of alternative items that a website and any other app mightoffer .Also, with the rapid development of Mobile Phones and Applications, it is faster than everfor end-users to find the information they want whenever they want but also for engineers tocollect user data, analyze them to get better insights, and perfect the model. Mobile devices have become one of the leading ways to introduce new technologies to the masses, be it 3DApplications, gaming, or something related to AI and Machine Learning. One of the popular uses cases in this scenario has been the extensive use of Recommender Systems. Netflix, Yelp,App Stores are some examples of how to implement these kinds of systems. With the abundance of dining options, users can save time and effort by relying on the recommendation system to quickly narrow down choices, reducing decision fatigue. Some systems incorporate social elements, allowing users to see recommendations from friends or a broader community,

fostering social interaction around dining choices. A restaurant recommendation system serves to simplify the process of choosing where to dine by leveraging data and algorithms to provide personalized, relevant suggestions, ultimately enhancing user satisfaction and engagement.

LITERATURE SURVEY :

Existing problem:

Restaurant recommendation systems aim to provide users with personalized suggestions based on their preferences, behavior, and other relevant factors. While these systems can be beneficial, they may also face several challenges and problems. When a new user signs up, the system may lack sufficient data about their preferences and behavior, making it challenging to provide accurate recommendations. Similarly, if a new restaurant is added to the platform, the system may struggle to recommend it because there is limited or no user interaction data available.

Some recommendation systems tend to recommend popular items, leading to a lack of diversity in suggestions. Users may end up in a "filter bubble" where they are repeatedly recommended similar items, limiting their exposure to new and diverse options.Users often have multiple criteria when choosing a restaurant, such as cuisine type, price range, ambiance, and dietary restrictions. Recommendation systems that only consider a subset of these criteria may not fully meet user preferences.If recommendation algorithms are trained on biased data, they may perpetuate and even amplify existing biases. This can result in recommendations that are unfair or discriminatory, favoring certain demographics over others.

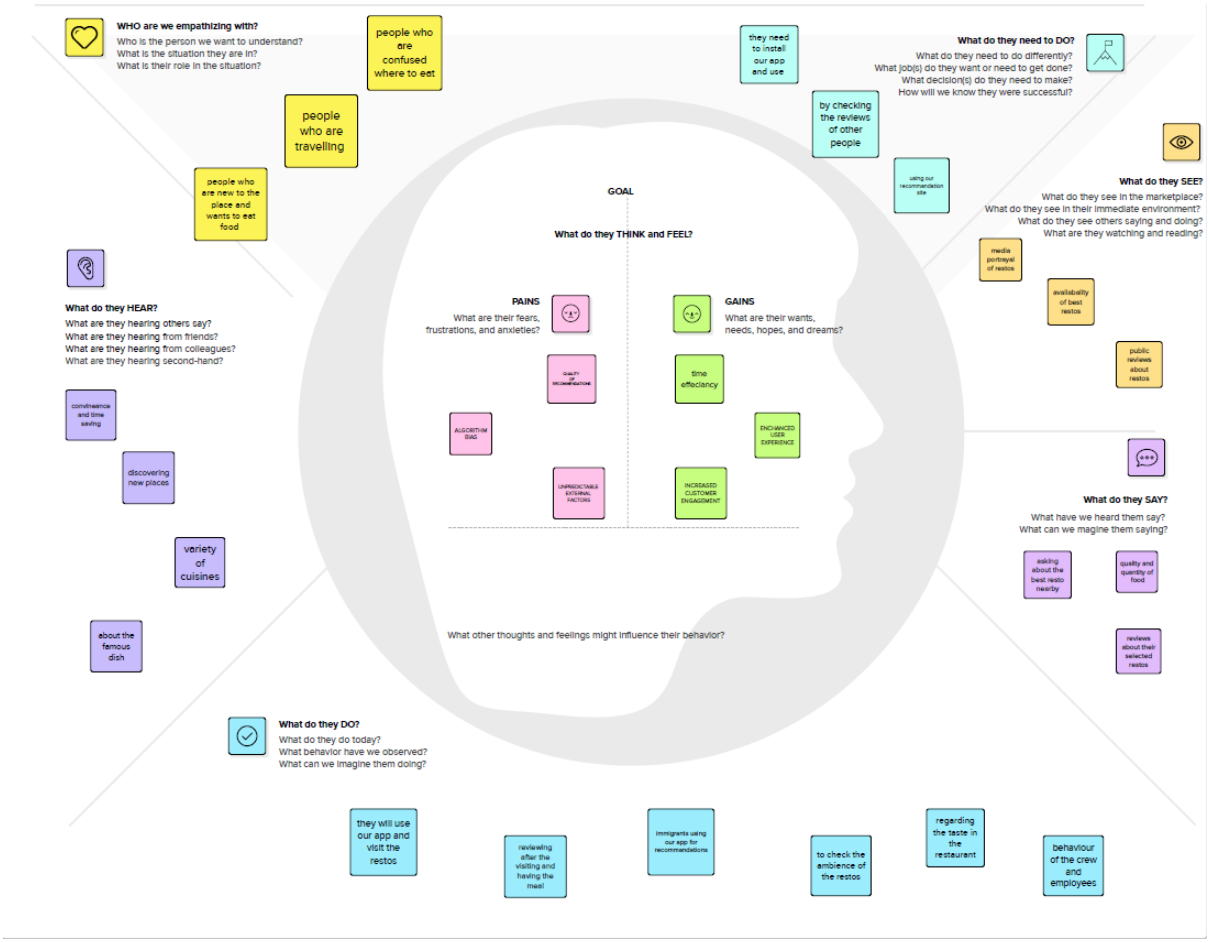
REFERENCES:

PROBLEM STATEMENT DEFINITION:

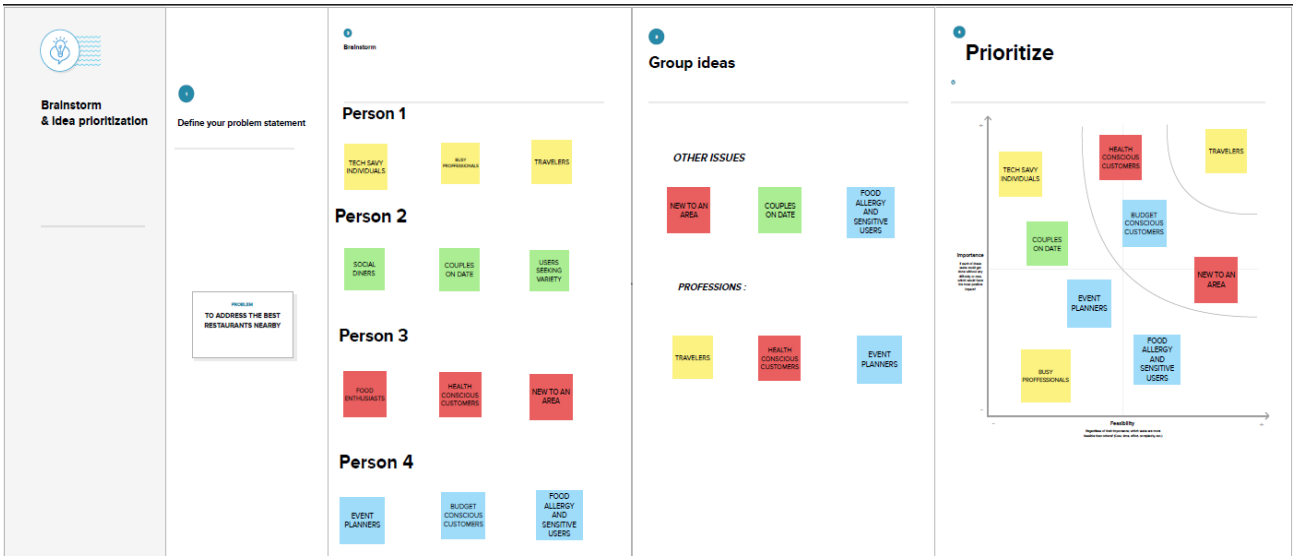
Many diners struggle to find suitable restaurants that match their preferences and dietary restrictions. This can lead to frustration and wasted time searching for a restaurant that meets their needs. Therefore, there is a need for a restaurant recommendation system that can provide personalized recommendations based on user preferences, location, and dietary restrictions. The system should be easy to use and provide accurate and relevant recommendations to help diners make informed decisions about where to eat.

Ideation and proposed solution :

Empathy map canvas :



IDEATION AND BRAINSTORMING :



Requirement analysis

Functional requirement :

The software requirements are the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirementsprovide a basis for creating the software requirements specification. It is useful inestimating cost, planning team activities, performing tasks and tracking the teams andtracking the team's progress throughout the development activity.

Operating system: Windows 11.

Used: Python, Jupyter Notebook and Anaconda Prompt , HTML , collabrative filtering.

Non functional requirements :

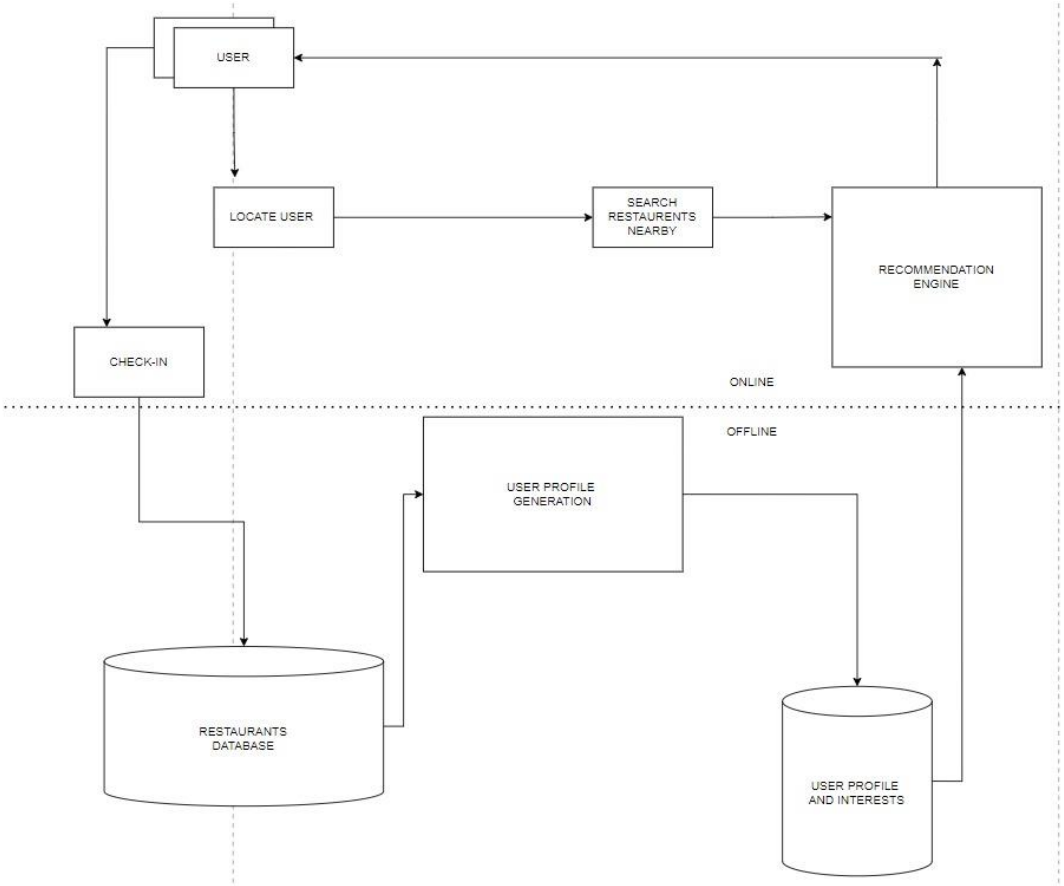
The hardware requirements may serve as the basis for a contract forthe implementation of the system and should therefore be a complete and consistentspecification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

Hardware: Intel i3 Core

RAM: 4GB

Project design

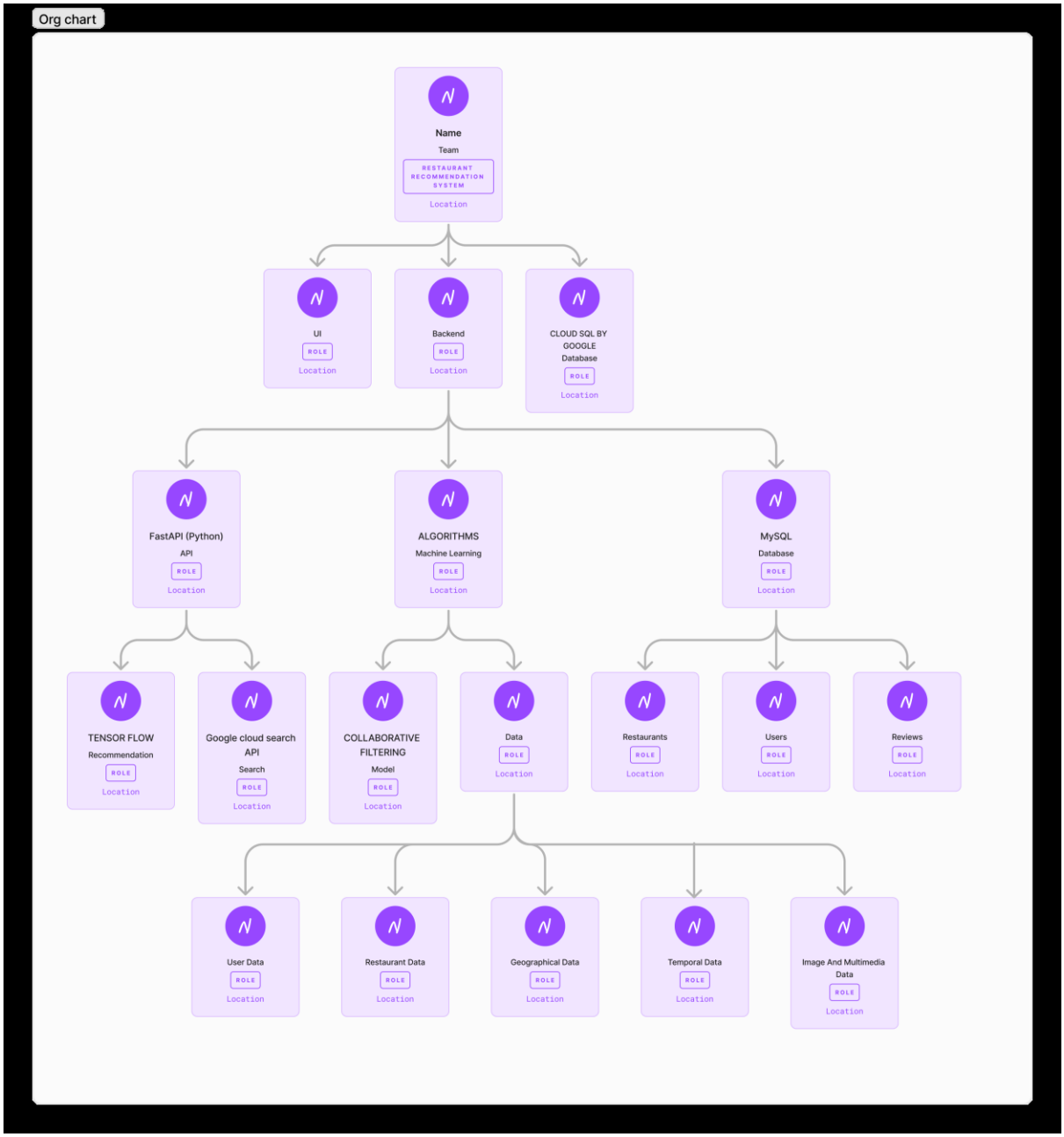
Data flow diagrams:



User stories:

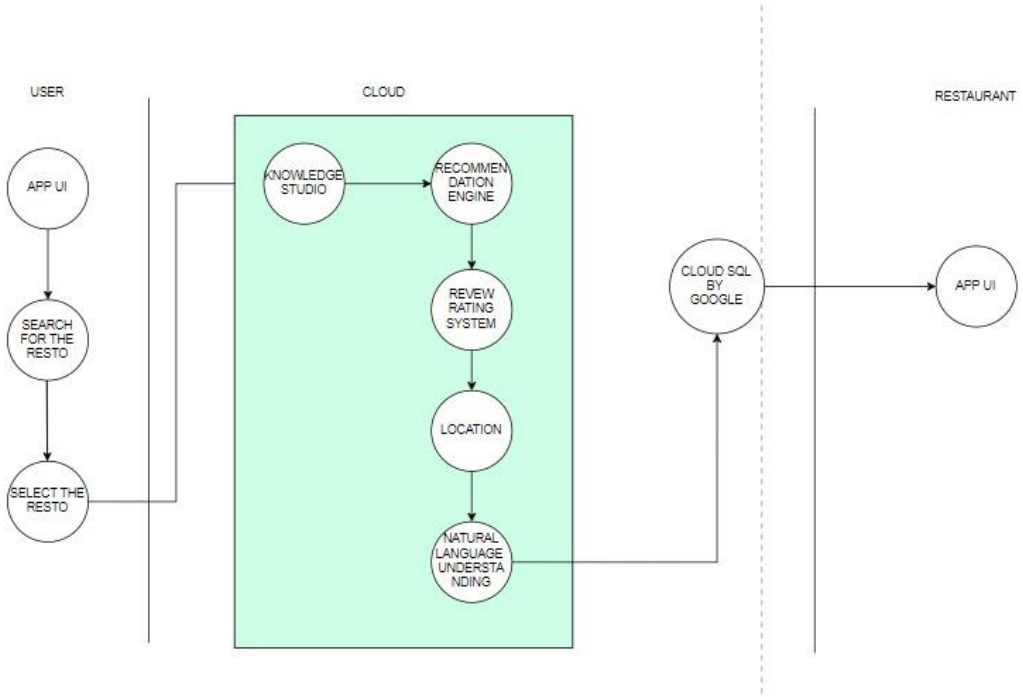
User Type	Functional Requirement	User Story Number	User story/Talk	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user i can register in the application by creating unique account name and entering password and confirming password	I can accept my account and personal data	High	Sprint-1
		USN-2	The application to provides me with a personalized restaurant recommandations and suggest resources or actions I can take, based on my results.	Offer a user-friendly interface for completing the questionnaire.	High	Sprint-1
		USN-3	I can access analytics and insights within the application to access nearby restaurants info	It offers recommendations of nearby restos and kitchens	Low	Sprint-2
		USN-4	As a user, I can sign up for the application by choosing a unique account name and setting a password	I can check my logins and my history of risks and suggestions	Medium	Sprint-1
Customer (Web user)						
Customer Care Executive						
Administrator						

Solution architecture :



PROJECT PLANNING AND SCHEDULING

Technical architecture :



Sprint planning and estimation :

Sprint delivery schedule :

Coding and solutioning :

App.py :

```
import pandas as pd
from flask import Flask, render_template, request


app = Flask(_name_)


# Load the dataset
restaurant_data = pd.read_csv("restaurant1.csv")
```

```
@app.route('/')
def home():
    return render_template('home.html')

@app.route('/recommendation', methods=['POST'])
def recommendation():
    # Get user input from the form
    user_preference = request.form.get('preference')

    # Recommendation logic (Replace this with your actual
recommendation logic)
    recommended_restaurants =
recommend_restaurants(user_preference)

    return render_template('recommendation.html',
user_preference=user_preference,
recommendations=recommended_restaurants)

def recommend_restaurants(user_preference):
    # Replace this with your actual recommendation logic
    # For now, it just returns a random restaurant
    recommended_restaurants = restaurant_data.sample(5)
    return recommended_restaurants

if _name_ == '_main_':
    app.run(debug=True)
```

home.html:

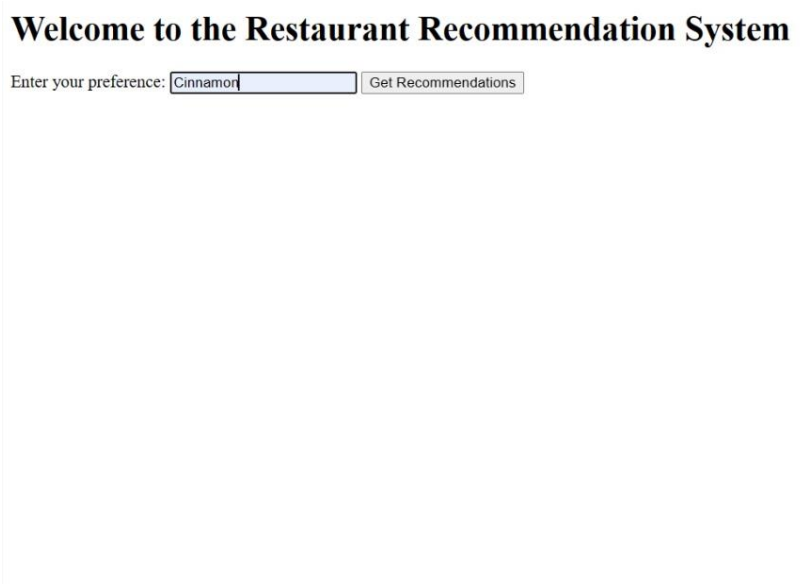
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Restaurant Recommendation System</title>
</head>
<body>
    <h1>Welcome to the Restaurant Recommendation System</h1>
    <form action="/recommendation" method="post">
        <label for="preference">Enter your preference:</label>
        <input type="text" name="preference" required>
        <button type="submit">Get Recommendations</button>
    </form>
</body>
</html>
```

Recommendation.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Restaurant Recommendations</title>
</head>
<body>
    <h1>Restaurant Recommendations for {{ user_preference }}</h1>
    <ul>
        {% for index, restaurant in recommendations.iterrows() %}
            <li>{{ restaurant['name'] }} - Cuisine: {{ restaurant['cuisine'] }} | Rating: {{ restaurant['rating'] }} | Cost: {{ restaurant['cost'] }}</li>
        {% endfor %}
    </ul>
</body>
</html>
```

Results :



Restaurant Recommendations for Cinnamon

- World of Waffles - Cuisine: | Rating: | Cost: 400.0
- Mimansa @ Foxtrot - Cuisine: | Rating: | Cost: 1.0
- The Flying Squirrel - Cuisine: | Rating: | Cost: 450.0
- Manasa Fishland - Cuisine: | Rating: | Cost: 650.0
- Bangalore Pub Exchange - Cuisine: | Rating: | Cost: 1.0

Restaurant Recommendations for Grand Village

- Mumbai Bistro - Cuisine: | Rating: | Cost: 500.0
- Patisserie Nitash - Cuisine: | Rating: | Cost: 500.0
- Halli Mane Food Court - Cuisine: | Rating: | Cost: 100.0
- Yo! Thali - Cuisine: | Rating: | Cost: 400.0
- Java City - Cuisine: | Rating: | Cost: 450.0

Advantages and disadvantages :

Advantages :

Recommendation systems analyze user preferences, behavior, and past choices to provide personalized suggestions. This ensures that users receive restaurant recommendations that align with their tastes and

preferences. Users can save time by quickly finding suitable restaurants without manually searching through numerous options. The system streamlines the decision-making process by presenting relevant choices based on individual preferences. A well-designed recommendation system can keep users engaged with the platform, leading to increased user retention. Users are more likely to return to a service that consistently provides them with relevant and satisfying recommendations. Recommendation systems can introduce users to new and unfamiliar restaurants that match their preferences. This encourages users to explore a variety of dining options, contributing to a more diverse culinary experience. By offering accurate and personalized recommendations, users are more likely to be satisfied with their dining choices. This positive experience contributes to customer loyalty and positive reviews, enhancing the reputation of both the recommendation system and the restaurants it suggests. Systems can collect user feedback on recommendations, allowing for continuous improvement. Analyzing user interactions and feedback helps the system adapt and refine its algorithms over time, ensuring better accuracy and relevance. In addition to recommending restaurants, a recommendation system can suggest related services or products, such as special promotions, menu items, or even partner services. This can contribute to increased revenue for both the platform and the recommended restaurants.

Disadvantages:

Recommendation systems heavily depend on user data, and there is a risk of the system making recommendations based on past choices, potentially limiting the discovery of new or diverse options. Users may find themselves stuck in a "filter bubble" where they are only exposed to similar recommendations. Collecting and analyzing user data for personalized recommendations can raise privacy concerns. Users may be uncomfortable with platforms tracking their preferences and behavior, especially if the data is not handled securely or transparently.

Recommendation algorithms may inadvertently reinforce existing biases present in the data they are trained on. If the training data contains biases related to demographics or user preferences, the system may perpetuate those biases in its recommendations, leading to unfair or stereotypical suggestions. Recommendation systems may struggle to understand the full context of a user's preferences and requirements. Factors like special occasions, dietary restrictions, or the desire for a specific atmosphere might not be adequately captured by algorithms, leading to recommendations that do not fully meet user expectations. Recommender systems may face challenges when dealing with new users or restaurants with limited data. Without sufficient user interaction history, the system may struggle to make accurate and relevant recommendations, resulting in a "cold start" problem. User preferences can change over time due to various factors such as mood, season, or trends. Recommendation systems might not always adapt quickly enough to these changes, leading to less accurate suggestions. Some recommendation algorithms are complex and may lack transparency. Users might not understand why a particular restaurant is being recommended to them, which can lead to a lack of trusin the system. Ensuring fairness in recommendation systems is a challenge. There may be concerns about whether the algorithms treat all users and restaurants fairly, without favoring specific groups or types of establishments.Popular restaurants might receive more visibility in recommendation systems, potentially overshadowing smaller, lesser-known establishments. This can create challenges for small businesses trying to gain visibility and attract customers.

CONCLUSION:

In this paper, a method to extract user preferences of food from their online comments about restaurants has been presented. This method is based on the use of natural language processing techniques for processing the text of user comments and extracting the desired food

names. The semantic similarity approach has been exploited to conceptually cluster food names. Finally, a recommender system has been proposed that suggests the nearby restaurants that match the food preferences of the user. The proposed recommender system is context-aware, so that, by using user preferences, location, time and feedback of all users, it recommends nearby restaurants that are open at that time and are well-matched with user's preferences.

In this project, we developed a app which recommend the restaurant based on the choice of your interest. This is used for the users to predict the suitable and bestrestaurant as per their tastes. The content based filtering and collaborative basedfiltering makes the recommendation more efficient so that each user can use this

application for their easy prediction of restaurant. In this project, we developed a model that could recommend the restaurant based on thechoice of interest.

FUTURE SCOPE :

In the future, this work could be improved upon by improving the existing context-aware algorithm where more contexts, such as weather, traffic conditions, and other relevant factors,could be taken into account to design a smarter and more robust recommender system with better prediction accuracy. As users usually visit restaurants in groups, group-based restaurant recommendation is a main future direction for this research.

AI and ML models will aim to enhance contextual understanding, taking into account specific user contexts, such as dietary restrictions, special occasions, or the purpose of the meal (casual, business, celebration). This will result in more precise and relevant recommendations.