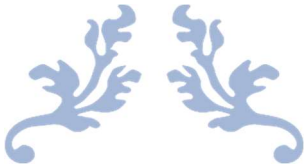


RESTAURANT RECOMMENDATION SYSTEM

PROJECT REPORT

NOVEMBER 21, 2023

TEAM - 592864

Table of Contents

CHAPTER 1
INTRODUCTION

Lately, there has been a surge in restaurant apps that enable customers to order and pay for their meals via a mobile app while dining in. This has streamlined the order process and allowed restaurants to serve food more efficiently. However, these apps could benefit from a recommendation system that suggests new dishes to customers. Such a system could significantly boost sales and make it simpler for customers to decide what to order next.

The concept of a food recommendation system has arisen as a solution to the challenges customers face when ordering and staying up to date with the latest meals at restaurants that match their profile. This system significantly enhances the user experience by simplifying the ordering process and suggesting meals based on their profile, previous orders, and other factors. Previously, ordering food and learning about new dishes could be a hassle for both customers and servers, resulting in delays for all involved.

The most important component of the project is to provide an accurate prediction of the user’s next meal, which may be done by looking at the user’s previous orders.

This project involves developing a recommendation system that works alongside existing ordering apps and serves as a basis for assessing the idea and potential improvements. The process involves two key components: predicting the user's next meals and gathering feedback from users to gauge the system's accuracy.

1.1 Project Overview

The aim of this project is to develop an intelligent restaurant recommendation system that assists users in discovering and selecting restaurants based on their preferences, location, and past behavior. The system will leverage machine learning algorithms to analyze user data and provide personalized recommendations, enhancing the dining experience for users.

Key Features:

- **User Profiling:** Implement a system to create user profiles based on their preferences, such as cuisine type, price range, location, and past restaurant ratings.
- **Data Collection and Processing:** Gather restaurant data from various sources (e.g., APIs, web scraping) and preprocess it for analysis and recommendation generation.
- **Machine Learning Algorithms:** Utilize recommendation algorithms such as collaborative filtering, content-based filtering, or hybrid models to generate personalized restaurant recommendations for users.
- **User Interface:** Develop an intuitive and user-friendly interface (web or mobile application) for users to interact with the recommendation system, allowing them to input preferences and view recommended restaurants.
- **Feedback Loop:** Implement a mechanism for users to provide feedback on recommended restaurants, improving the accuracy of future recommendations.

1.2 Purpose

The major purpose of this project is to build a food recommendation system, and the following sub-objectives must be completed to complete this implementation and meet the project's objectives:

- Build a database based on real data from restaurants and users.
- Analyze the available recommendation systems and compare their strengths and weaknesses.
- The research about all the methods that are used in recommendation system applications.
- Perform tests between all the different methods.
- Implement the best method and improve it.

- Carry out tests based on the data we have.
- Receive feedback from users and compare the prediction results with the user's expectations.

These objectives oversee steering the entire process of completing the project, as well as achieving the desired results and establishing the project's starting points.

CHAPTER 2
LITERATURE SURVEY

A personalized recommender system victimization machine learning based mostly on sentiment analysis over social information (2016). This paper proposes such a social framework, and it provides the user with faster and additional relevant information, avoiding moot information and providing abundant required personalization.

- Analysis of classification models supported cookery prediction exploitation machine learning (2017). The correlation between numerous recipes and their ingredient sets was investigated with the assistance of common classification techniques. The tests were conducted on the dataset compiled from numerous sources additionally the} accuracy of classifiers who wanted to predict the cuisines were also compared.
- A hybrid recommendation system considering visual info for predicting favorite restaurants (2017). during this paper, particularly investigate the influence of visual info, i.e., photos taken by customers and placed on blogs, on predicting favorite restaurants for any given user. It offers visual info that effectively aids favorite eating place prediction.
- Machine learning is primarily based on food direction recommendation system (2018). In this, they use similarity techniques of user primarily based approach and introduce fastened size neighborhood and threshold-based neighborhood to identical. The performance for the Allrecipes knowledge set is found to be higher than the simulated dataset since there is an additional range of interactions between users and things.
- Recommendation system supported item and user similarity on restaurants directory online (2018). It merges the item similarity and user similarity options to come up with recommendations. analysis shows that the advice system-supported item similarity yields a higher F1-measure price once examination of user similarity.

2.1 Existing problem

While many existing recommender systems mainly target individuals, there is a remarkable increase of recommender systems which generate suggestions for groups. Some early systems were developed in a variety of domains, such as,

1. group web page recommendation (Lieberman et al. 1999),
2. tour packages for groups of tourists (Ardissono et al. 2003),
3. music tracks and playlists for large groups of listeners (Crossen et al. 2002),
4. movies and TV programs for friends and family (O'Connor et al. 2001, Yu et al. 2006).

Group scenarios are especially popular in the food domain in which a group of family members, friends or colleagues wants to have a party or simply have a meal together.

However, the complexity significantly increases when food recommender systems need to consider the preferences of all group members and strategies for achieving the consensus within group members. From the survey, we have inferred that they have developed a recommended system just to search for food. Many Restaurants stores and maintain their day-to-day transactions manually. But some of them have automation systems which help them to store the data. But such restaurants are storing the information about the orders and the customer information. They don't have the facility to store the information of feedback and favorite orders of customers over some period. Restaurants have standalone applications so at one time, they have the facility of many screens or many operations which are happening at one time. So, they are storing them and then at last, the restaurant managers will be able to see the data of the last day.

2.2 References

Research on restaurant recommendation systems is a dynamic field, and several studies contribute to addressing various challenges in this domain.

a) Notable references include:
    1. "**Collaborative Filtering for Implicit Feedback Datasets**" by Yifan Hu et al., which explores collaborative filtering techniques tailored for implicit user feedback, offering insights into enhancing recommendation accuracy.
    2. The paper "**Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model**" by Yehuda Koren

provides a comprehensive understanding of matrix factorization methods for collaborative filtering, offering valuable perspectives on improving recommendation quality.

b) For context-aware recommendations:
    1. "**Context-aware Collaborative Filtering for Yelp Restaurant Recommendations**" by Sujoy Chatterjee et al. investigates the impact of contextual information on restaurant suggestions, providing guidance on incorporating temporal and spatial factors. These references, among others, contribute to the ongoing development and refinement of restaurant recommendation systems, offering valuable insights for researchers and practitioners in the field.

If we analyze these data together with the graph of number of companies that deliver food, we observe that the increase in the number of delivery food companies and the increase of the wages lead to the increase in the number of people that use online ordering food applications. Also, As demonstrated in the primary motivation for consumers ordering food is to save time. Another aspect that are many channels on different platforms, such as YouTube, contains specific categories only for food that would make people curious and in passion to try out different cultures food that they can see throughout the “influencers”, by ordering them from their original restaurants to try out the original taste of the food.

2.3 Problem Statement Definition

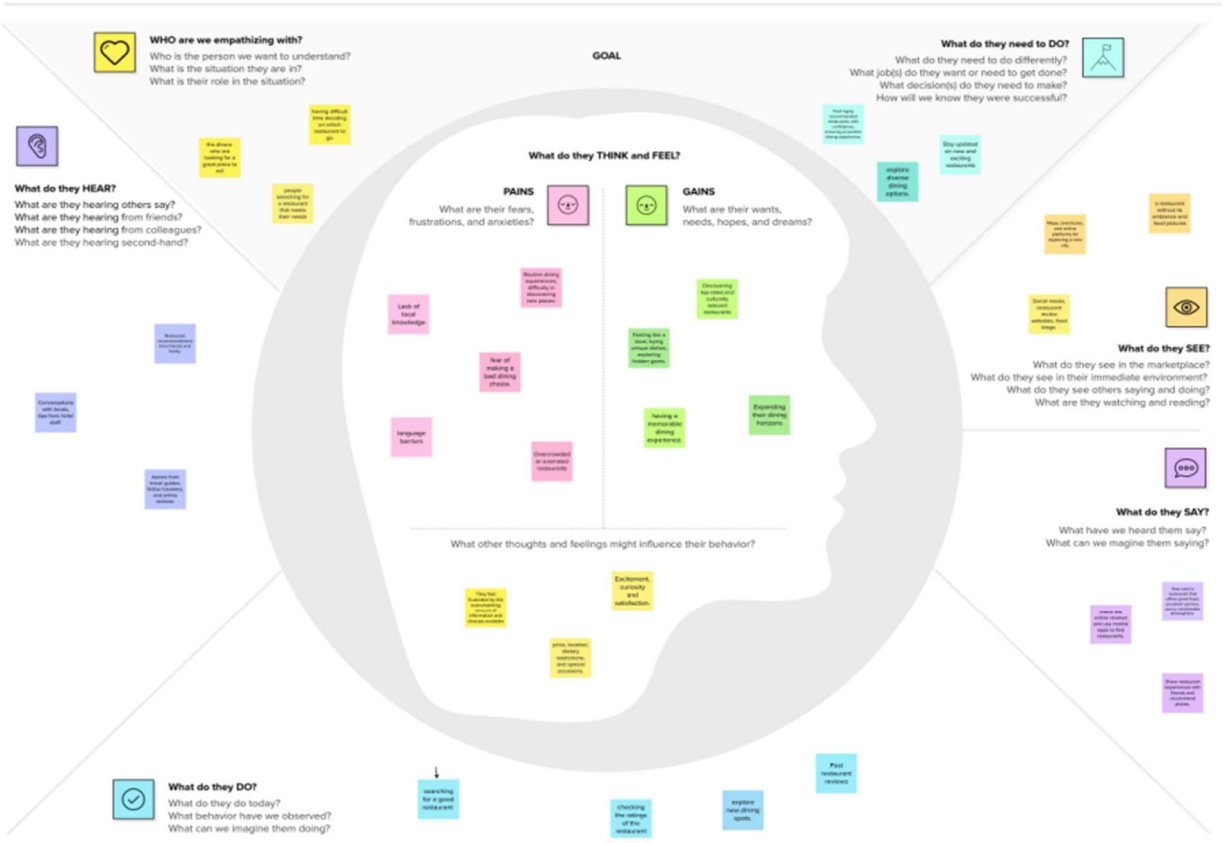
We are proposing a system which uses stream lit to develop a small web app. Our main aim is to build a restaurant recommendation system that provides personalized restaurant recommendations to users. Since different people have different food preferences and dietary restrictions, we perform careful feature selection to take advantage of the information reflected in a user’s reviews. We develop a restaurant recommendation system using the Latent Factor Collaborative Filtering Optimization.

This system recommends restaurants for users or groups of users based on their preferences such as beautiful ambience, good food, tasty desserts and so. Our system provides personalized restaurant recommendations to users. The recommendation for new users is done by considering the user’s location by tracking his location through GPS, and by the ratings we got from Zomato’s API. The restaurant that is near to the user, whose ratings are high will be recommended. For the old user, we will consider his behavior while using the application along with his location and restaurant popularity.

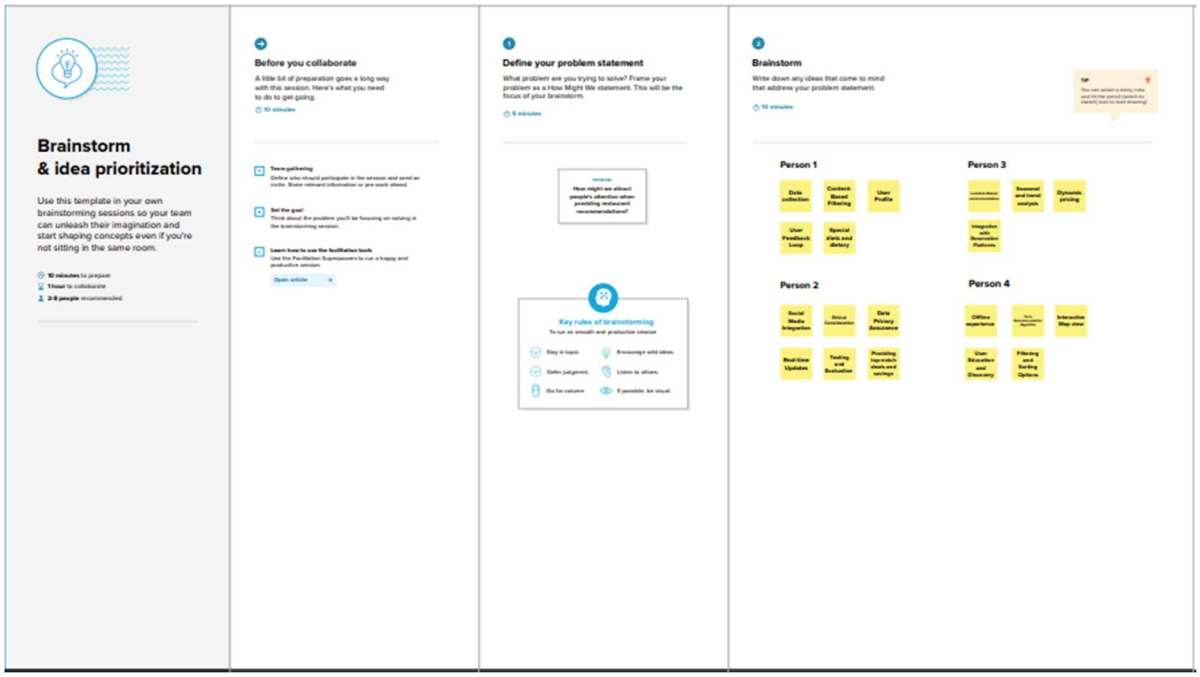
CHAPTER 3

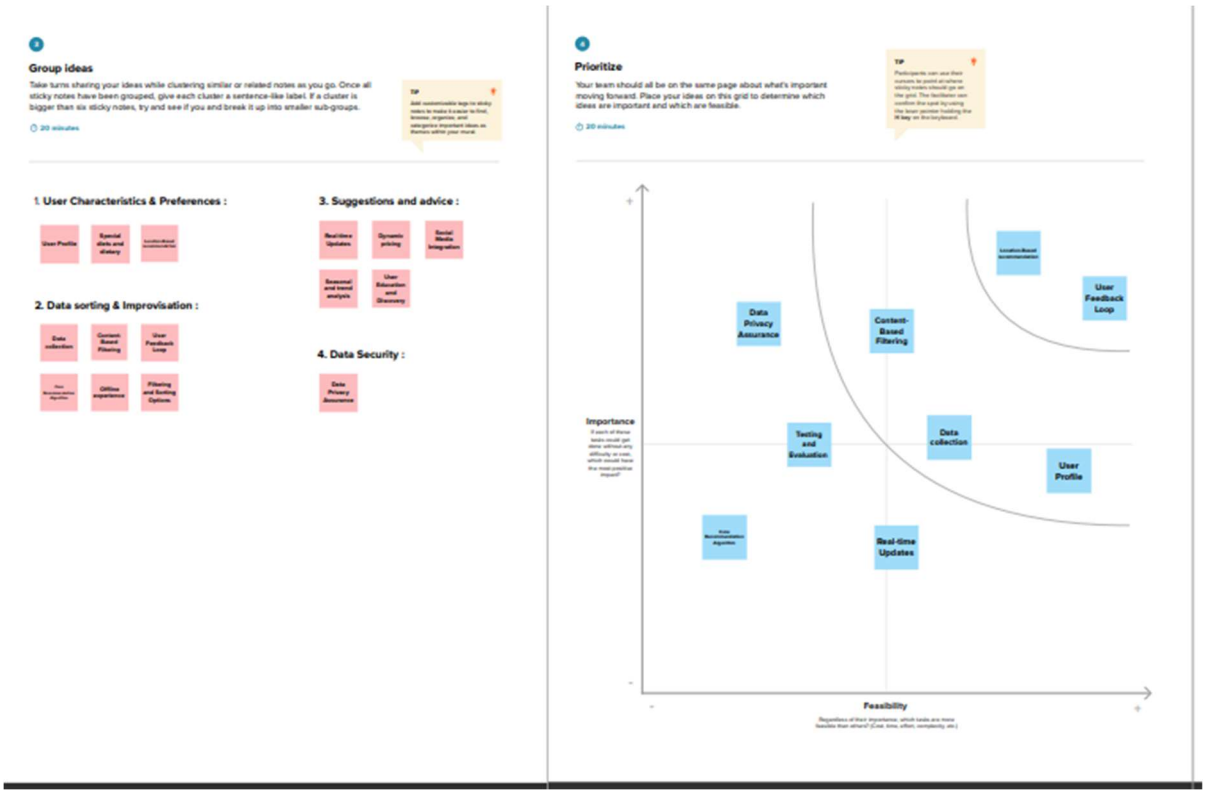
IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming





CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional requirement

A software function is defined by a functional need. A set of inputs, behaviors, and output is referred to as a function. These are the kinds of requirements that a recommendation system would examine and evaluate.

Below is a list of the functional criteria. Functional requirements for a restaurant recommendation system project define the specific features and capabilities that the system must have to fulfill its purpose.

Here are some key functional requirements for such a project:

1. **User Registration and Profile Creation:**
   Users should be able to register and create profiles with details such as preferences, dietary restrictions, and past restaurant ratings.
2. **Restaurant Data Management:**
   The system must manage a comprehensive database of restaurant information, including details on cuisine, location, price range, operating hours, and user reviews.
3. **Search and Filtering Options:**
   Provide users with the ability to search for restaurants based on criteria like cuisine type, location, price range, and specific features (e.g., outdoor seating, Wi-Fi).
4. **User Feedback Mechanism:**
   Include a feedback system for users to rate and review recommended restaurants, contributing to the improvement of future recommendations.
5. **User Interface (UI):**
   Design an intuitive and user-friendly interface, accessible via web or mobile applications, allowing users to easily interact with the recommendation system and view suggested restaurants.
6. **User Authentication and Security:**
   Incorporate robust user authentication mechanisms to protect user accounts and personal information. Ensure secure data transmission and storage.

4.2 Non-Functional requirements

These requirements are characteristics that is used to assess a system's functioning rather than its specific behaviors, they can be related to issues like accuracy.

Below is a list of non-functional requirements Recommendation system methods:

- NFR-01: The data filled in should be correct and accurate.
- NFO-02: User must be created, and profile must be filled.
- NFO-03: Rating should be positive.
- NFO-04: To consider product for rating it should have minimum amount of voting by users.
- NFR-05: Snowflake must be used to implement the database.

Business rules These are not functional requirements that the system must follow to meet quality standards, such as behaviors and rules.

The business rules for this system are listed below:

- BR-01: Many products are related to many restaurants.
- BR-02: Many products are related to many Orders.
- BR-03: Many orders are related to one user.
- BR-04: Many allergies are related to many users.

Data requirement:

These are a type of non-functional requirement that specifies the information that will be used in the system.

The system's information requirements are stated below:

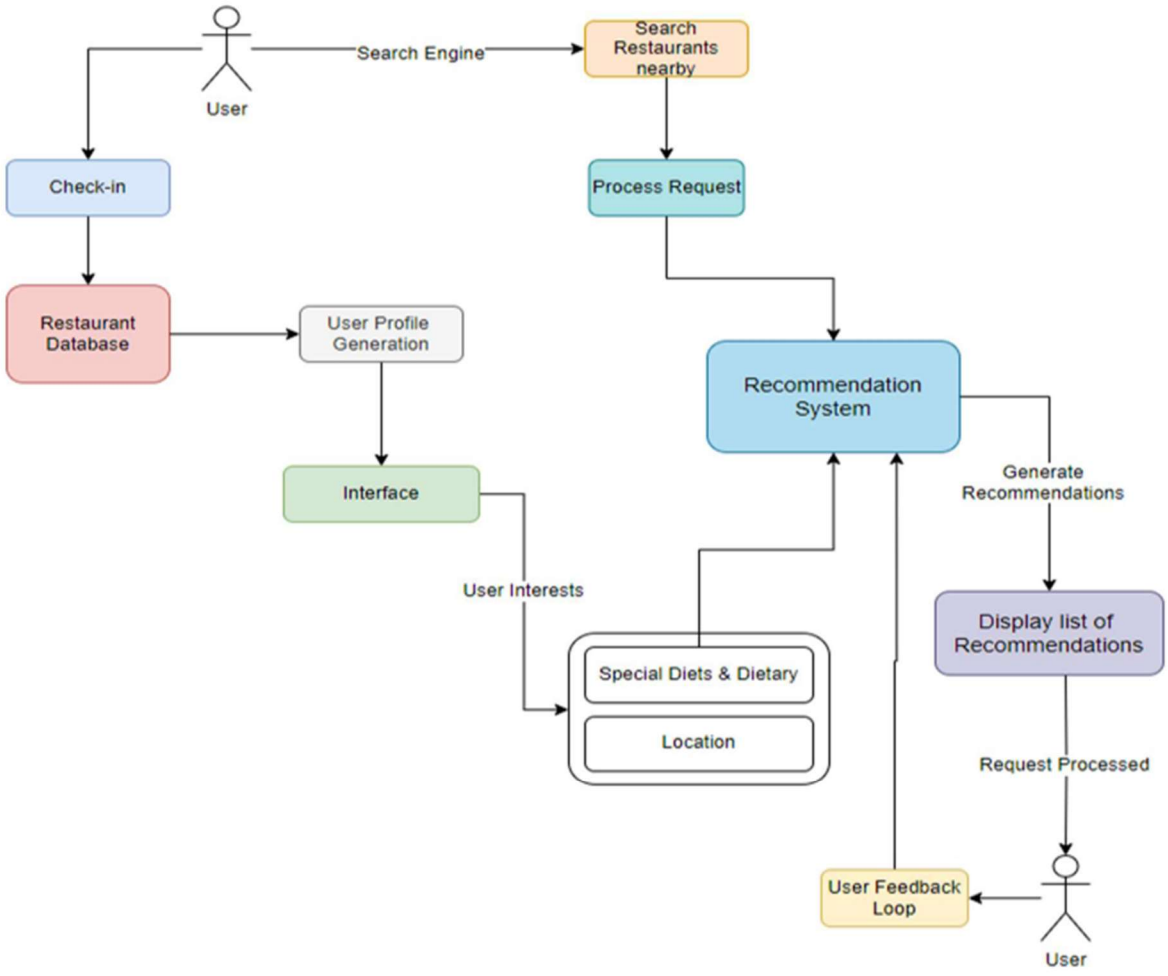
- DR-01: For each user will be stored: User ID - Name - Gender – Age - Nationality - Allergies – Preferences.
- DR-02: For each product will be stored: Product ID - Name - Allergy type - Origin country - Preparation time - Discount - Ingredients - Description - Vote average - Vote count – Type.
- DR-03: For each order will be stored: Order ID - User ID - Product ID - Day - Weather – Time.

CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

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.

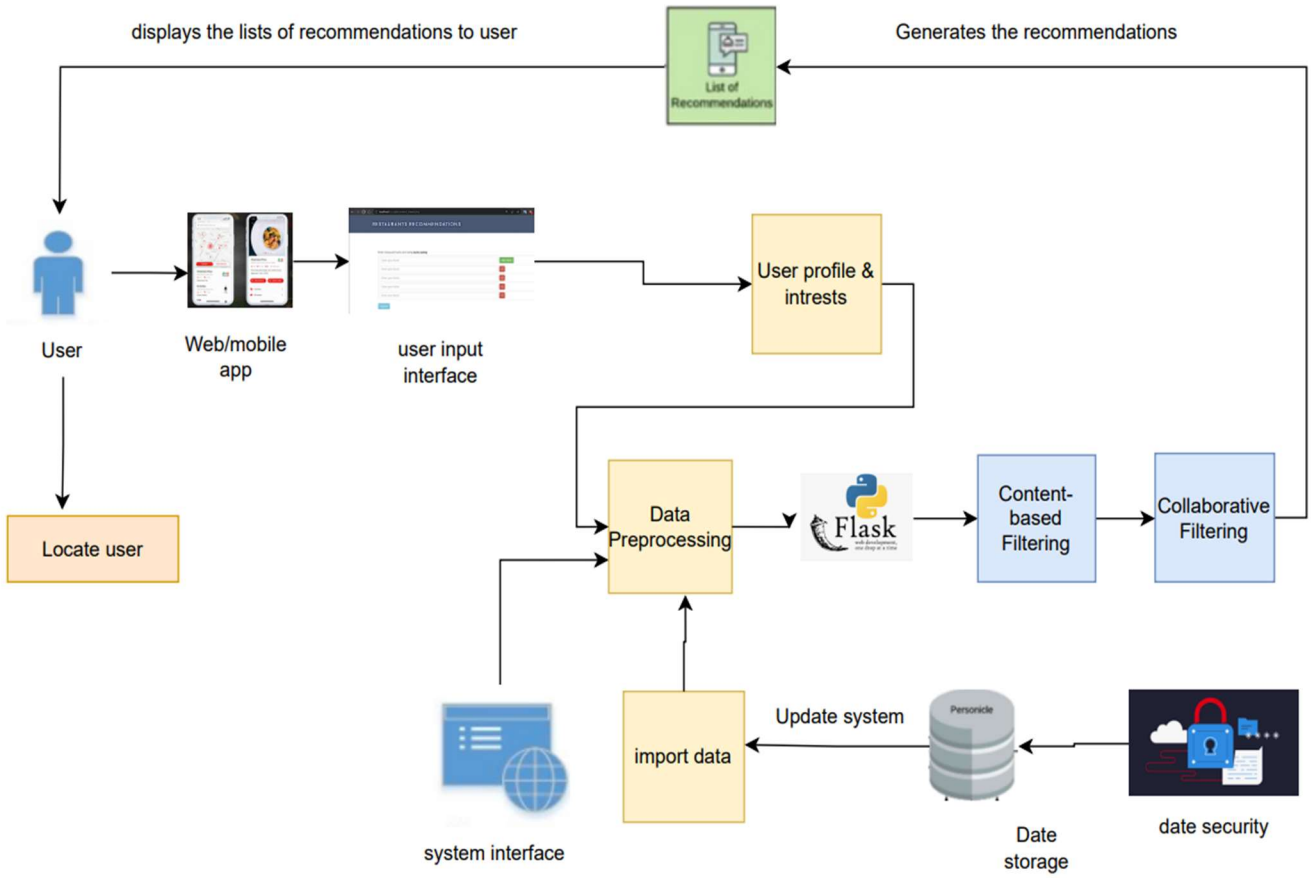


User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User/ Consumer	User profile creation	US001	As a user, I want to create a profile to receive personalized restaurant recommendations.	Users can create a profile with basic information. The system should store and update profiles accurately.	High	1.0
User/ Consumer	Real-time Recommendations	US002	As a user, I want to receive real-time restaurant recommendations based on my location and preferences.	The system recommends nearby restaurants based on user location and preferences. Recommendations are dynamically updated based on contextual factors.	High	1.0
Business Owner/ Restaurant	Login	US003	As a restaurant owner, I want to register my restaurant and manage its profile on the recommendation system.	Restaurants can register with essential details. Owners can manage and update restaurant profiles, including menu and promotions.	High	1.0
Business Owner/ Restaurant	Dashboard	US004	As a restaurant owner, I want to promote my restaurant by purchasing a prominent listing in user recommendations.	Owners can pay for promoted listings, gaining increased visibility in the recommendation feed. Promoted listings are clearly labeled to users.	Medium	1.1
Consumer/User	Culinary Challenges	US005	As a user, I want to participate in culinary challenges to explore new cuisines and earn rewards.	Users can join culinary challenges, receive rewards for trying new cuisines, and see progress updates. Challenges are engaging and diverse.	Medium	1.1

5.2 Solution Architecture

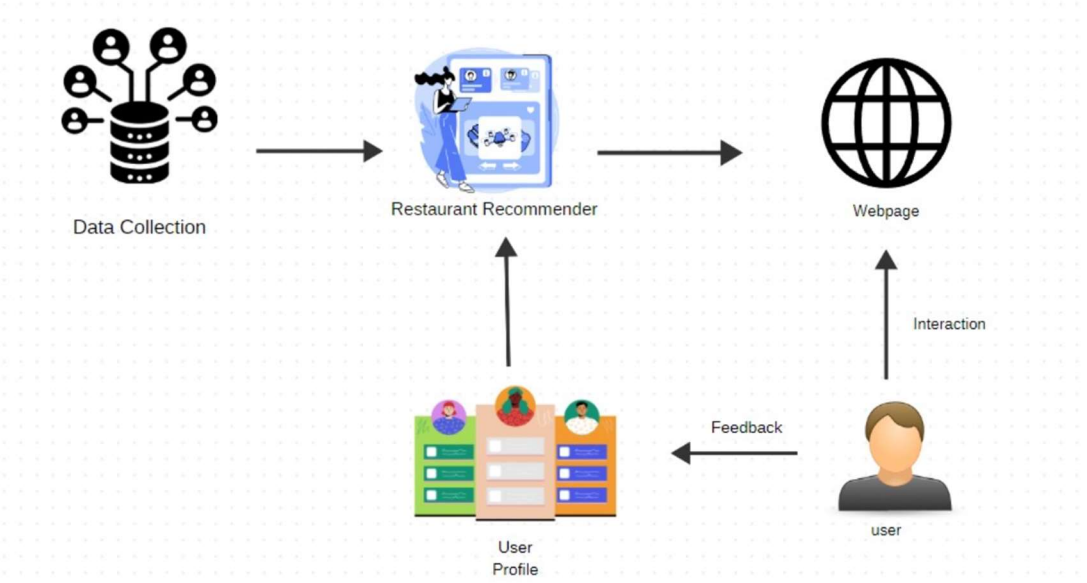
Designing a solution architecture for a restaurant recommendation system involves outlining the high-level structure, components, and interactions within the system.



CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture



6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Profile Creation	US001	As a user, I want to create a profile to receive personalized restaurant recommendations	5	High	Sree Nidhitha, Shloka
Sprint-1	Real-time Recommendations	US002	As a user, I want to receive real-time restaurant recommendations based on my location and preferences.	8	High	Suneel, Vaibhav
Sprint-1	Login	US003	As a restaurant owner, I want to register my restaurant and manage its profile on the recommendation system.	8	High	Shloka
Sprint-2	Dashboard	US004	As a restaurant owner, I want to promote my restaurant by purchasing a prominent listing in user recommendations.	5	Medium	Sree Nidhitha
Sprint-3	Culinary Challenges	US005	As a user, I want to participate in culinary challenges to explore new cuisines and earn rewards.	8	Medium	Vaibhav
Sprint-4	Payment Processing	US006	As a user, I want a seamless payment process for promoted listings or event tickets.	5	High	Suneel

6.3 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	18	6 Days	2 Nov 2023	8 Nov 2023	18	8 Nov 2023
Sprint-2	16	3 Days	9 Nov 2023	12 Nov 2023	15	12 Nov 2023
Sprint-3	20	3 Days	13 Nov 2022	15 Nov 2023	20	15 Nov 2023
Sprint-4	22	4 Days	16 Nov 2022	19 Nov 2023	22	19 Nov 2023

CHAPTER 7

CODING & SOLUTIONING

7.1 Feature 1

Building a restaurant recommendation system involves several steps, including data preprocessing, similarity computation, and recommendation generation. Below is a simplified example of a restaurant recommendation system. Note that this is a basic illustration, and in a real-world scenario, you might want to use more advanced techniques and libraries.

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import CountVectorizer
import string
```

[2]:
```
# Function to clean and preprocess text data
def preprocess_text(text):
    text = text.lower().translate(str.maketrans('', '', string.punctuation))
    return text
```

[3]:
```
# Function to get top words from text series
def get_top_words(text_series, top_n, ngram_range):
    vectorizer = CountVectorizer(ngram_range=ngram_range, stop_words='english')
    matrix = vectorizer.fit_transform(text_series)
    word_count = np.asarray(matrix.sum(axis=0)).ravel()
    words = np.array(vectorizer.get_feature_names_out())
    words_df = pd.DataFrame({'word': words, 'count': word_count})
    return words_df.sort_values(by='count', ascending=False).head(top_n)
```

[4]:
```
# Load the dataset
zomato_df = pd.read_csv('zomato.csv')
```

[5]:
```
zomato_df.head()
```

[5]:

	url	address	name	online_order	book_table	rate	votes	phone	location	rest_type	dish_liked
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	080 42297555\r\n+91 9743772233	Banashankari	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	080 41714161	Banashankari	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8/5	918	+91 9663487993	Banashankari	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...
3	https://www.zomato.com/bangalore/addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91 9620009302	Banashankari	Quick Bites	Masala Dosa
4	https://www.zomato.com/bangalore/grand-village...	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8/5	166	+91 8026612447\r\n+91 9901210005	Basavanagudi	Casual Dining	Panipuri, Gol Gappe

[6]:
```
# Data Cleaning and Preprocessing
zomato_df = zomato_df.drop(columns=['url', 'phone', 'dish_liked'])
zomato_df.dropna(how='any', inplace=True)
zomato_df.drop_duplicates(inplace=True)
```

[7]:
```
zomato_df = zomato_df.rename(columns={'approx_cost(for two people)': 'cost',
                                      'listed_in(type)': 'type',
                                      'listed_in(city)': 'city'})
```

[*]:
```
zomato_df['rate'] = pd.to_numeric(zomato_df['rate'].str.replace('/5', '').str.strip(), errors='coerce')
zomato_df['cost'] = zomato_df['cost'].str.replace(',', '.').astype(float)

zomato_df['reviews_list'] = zomato_df['reviews_list'].apply(preprocess_text)
zomato_df['cuisines'] = zomato_df['cuisines'].apply(preprocess_text)
```

[15]:
```
# Group by 'name' column and calculate the mean of 'rate' column
grouped_restaurants = zomato_df.groupby('name', as_index=False)['rate'].mean().round(2)

# Combine all reviews and cuisines for each unique name
combined_restaurants = zomato_df.groupby('name', as_index=False).agg({'reviews_list': 'sum', 'cuisines': 'sum'})

# Merge the two dataframes on 'name' column
merged_restaurants = pd.merge(grouped_restaurants, combined_restaurants, on='name')

# Create a 'tags' column
merged_restaurants['tags'] = merged_restaurants['reviews_list'] + merged_restaurants['cuisines']

# Drop unnecessary columns
final_df = merged_restaurants.drop(columns=['reviews_list', 'cuisines'])

# Add 'cost' column to the new dataframe
final_df['cost'] = zomato_df['cost']
```

[*]:
```
# Use CountVectorizer for text vectorization
vectorizer = CountVectorizer(max_features=5000, stop_words='english')
tags_vector = vectorizer.fit_transform(final_df['tags']).toarray()

# Calculate cosine similarity
similarity_matrix = cosine_similarity(tags_vector)
```

[17]:
```
# Function to recommend restaurants
def recommend_similar_restaurants(target_restaurant):
    target_restaurant_lower = preprocess_text(target_restaurant)

    if target_restaurant_lower not in final_df['name'].str.lower().values:
        print(f"Restaurant '{target_restaurant}' data not found.")
        return

    target_index = final_df[final_df['name'].str.lower() == target_restaurant_lower].index[0]
    distances = sorted(enumerate(similarity_matrix[target_index]), reverse=True, key=lambda x: x[1])

    print(f"Top 10 recommended restaurants for '{target_restaurant}':")
    for i in distances[1:11]:
        recommended_restaurant = final_df.iloc[i[0]]['name']
        print(recommended_restaurant)
```

[18]:
```
recommend_similar_restaurants("Red Chilliez")
```

```
Top 10 recommended restaurants for 'Red Chilliez':
Melange - Hotel Ekaa
Inchara Restaurant
Wazir's
Punjabi Tasty Khana
Garma Garam
Swad
B.M.W - Bhookh Mitaane Wala
Punjabi Dawat
Marwa Restaurant
Foodiction
```

[20]:
```
recommend_similar_restaurants("1947")
```

```
Top 10 recommended restaurants for '1947':
Atithi
The Rasoiya Street
Flavours - Octave Hotel & Spa
Paprica
The Onyx - The HHI Select Bengaluru
Cinnamon
Nouvelle Garden
Palki's
Thamboola
TBC Sky Lounge
```

[21]:
```
recommend_similar_restaurants("Random Value")
```

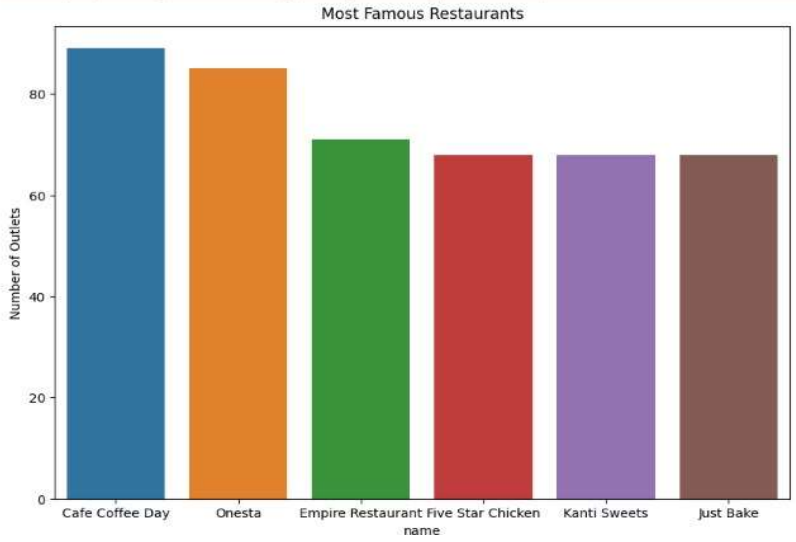
```
Restaurant 'Random Value' data not found.
```

7.2 Feature 2

Data visualization

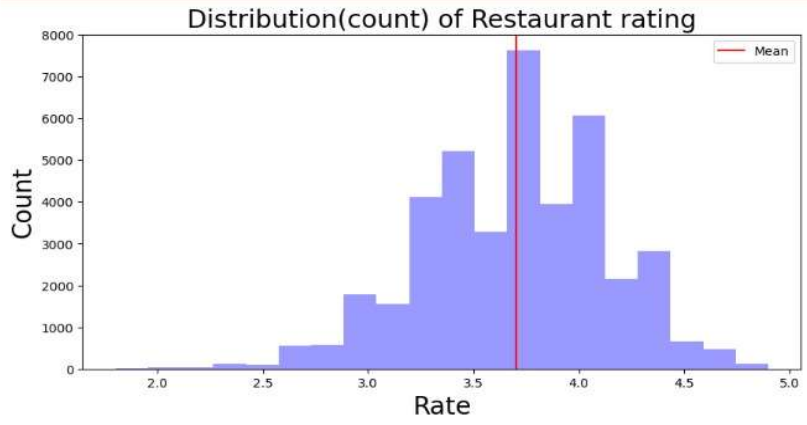
```
[9]: # Visualizations
plt.figure(figsize=(10, 7))
famous_restaurants = zomato_df['name'].value_counts()[:6]
sns.barplot(x=famous_restaurants.index, y=famous_restaurants, palette='tab10')
plt.title('Most Famous Restaurants')
plt.ylabel('Number of Outlets')
plt.show()
```

```
C:\Users\INFINITY\AppData\Local\Temp\ipykernel_4296\575231512.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for th
e same effect.

  sns.barplot(x=famous_restaurants.index, y=famous_restaurants, palette='tab10')
```




```
[10]: fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(10, 5))
sns.distplot(zomato_df['rate'], kde=False, color='b', ax=ax, bins=20)
ax.axvline(zomato_df['rate'].mean(), 0, 1, color='r', label='Mean')
ax.legend()
ax.set_ylabel('Count', size=20)
ax.set_xlabel('Rate', size=20)
ax.set_title('Distribution(count) of Restaurant rating', size=20)
plt.show()
```

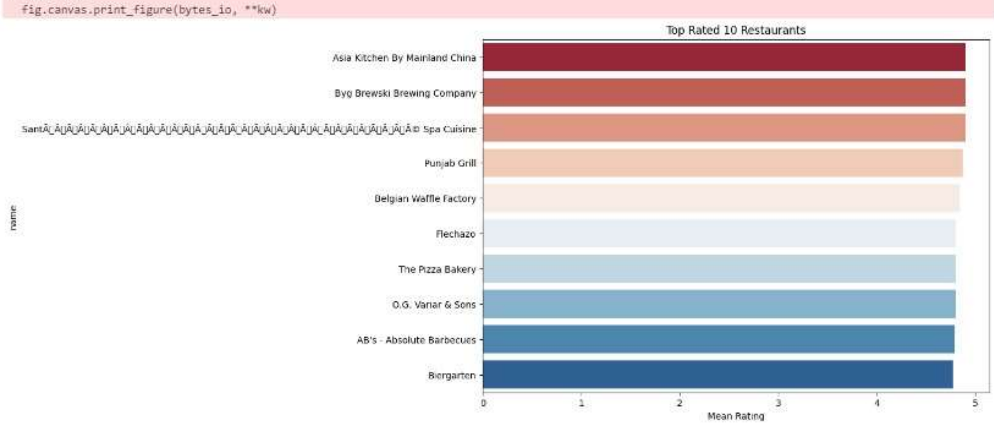
```
C:\Users\INFINITY\AppData\Local\Temp\ipykernel_4296\4072945885.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(zomato_df['rate'], kde=False, color='b', ax=ax, bins=20)
```



[11]:
```python
# Top Rated 10 Restaurants Visualization
df_rating = zomato_df.groupby('name', as_index=False)['rate'].mean().round(2)
df_rating = pd.DataFrame({'name': df_rating['name'], 'Mean Rating': df_rating['rate']})
df_rating = df_rating.sort_values(by='Mean Rating', ascending=False).head(10)
```

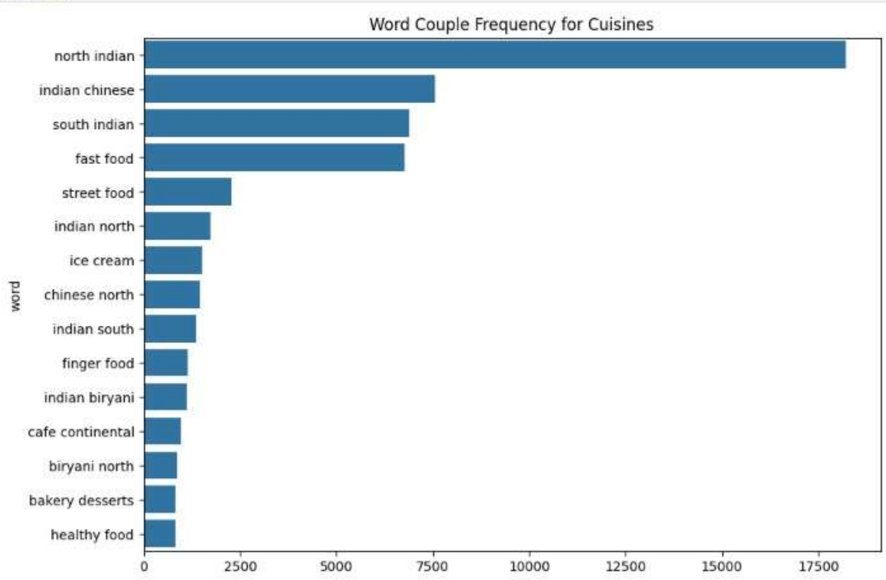
[12]:
```python
plt.figure(figsize=(10, 7))
sns.barplot(data=df_rating, x='Mean Rating', y='name', palette='RdBu')
plt.title('Top Rated 10 Restaurants')
plt.show()
```

```
C:\Users\INFINITY\AppData\Local\Temp\ipykernel_4296\967173411.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for th
e same effect.

  sns.barplot(data=df_rating, x='Mean Rating', y='name', palette='RdBu')
C:\Users\INFINITY\AppData\Local\Programs\Python\Python312\Lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Glyph 131 (\x83) missing from
current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\INFINITY\AppData\Local\Programs\Python\Python312\Lib\site-packages\IPython\core\pylabtools.py:152: UserWarning: Glyph 130 (\x82) missing from
current font.
  fig.canvas.print_figure(bytes_io, **kw)
```



[13]:
```python
# Word Couple Frequency for Cuisines Visualization
lst = get_top_words(zomato_df['cuisines'], 15, (2, 2))
df_words = pd.DataFrame(lst, columns=['word', 'count'])
```

[14]:
```python
plt.figure(figsize=(10, 7))
sns.barplot(data=df_words, x='count', y='word')
plt.title('Word Couple Frequency for Cuisines')
plt.show()
```



7.3 Database Schema

Restaurant Data Schema:

1. **restaurant_id (int):** Unique identifier for each restaurant.
2. **name (str):** Name of the restaurant.
3. **cuisine (str):** Type of cuisine offered by the restaurant.
4. **location (str):** Location or address of the restaurant.
5. **rating (float):** Average user rating for the restaurant.

User Data Schema:

1. **user_id (int):** Unique identifier for each user.
2. **name (str):** User's name or username.
3. **preferences (str or list):** User's preferences, such as favorite cuisine or location.

Interaction Data Schema:

1. **interaction_id (int):** Unique identifier for each interaction.
2. **user_id (int):** Foreign key referencing the user who interacted.
3. **restaurant_id (int):** Foreign key referencing the restaurant involved in the interaction.
4. **interaction_type (str):** Type of interaction (e.g., "viewed," "rated," "visited").
5. **timestamp (datetime):** Timestamp indicating when the interaction occurred.

CHAPTER 8

PERFORMANCE TESTING

8.1 Performance Metrics

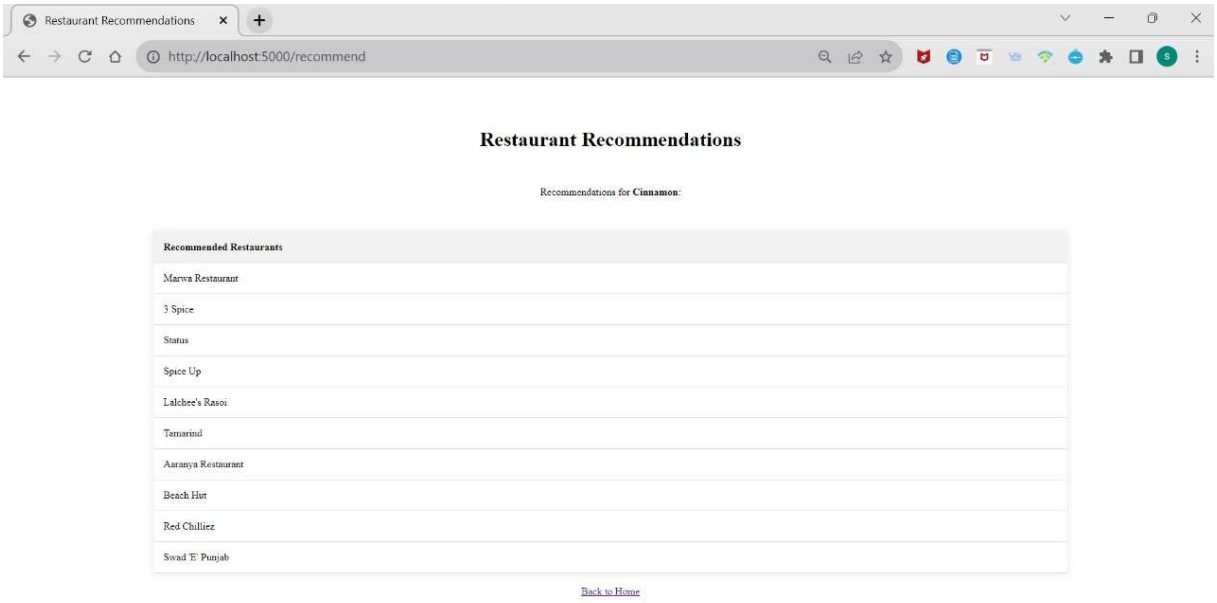
Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Parameter example 1	I have used Natural Language Processing as my model, so there's no as such the parameters. I can show some of the examples of the recommendations. This is for Red chillez.	
2.	Parameter example 2	This example is for the restaurants from1947-year.	

CHAPTER 9

RESULTS

9.1 Output Screenshots



CHAPTER 10

ADVANTAGES & DISADVANTAGES

A restaurant recommendation system can have various advantages and disadvantages, depending on its implementation and the perspective of different stakeholders.

Here are some advantages and disadvantages to consider.

ADVANTAGES:

1. **Enhanced User Experience:**
- Personalization: Recommending restaurants based on user preferences can enhance the overall user experience, making it more tailored to individual tastes and preferences.
- Convenience: Users can quickly discover new and suitable dining options, saving time and effort in the decision-making process.
2. **Increased Business for Restaurants:**
- Increased Visibility: Restaurants recommended by the system may experience increased visibility and customer traffic, helping them attract more patrons.
- Marketing Opportunities: The system can serve as a marketing tool, promoting certain restaurants based on user preferences and trends.
3. **Data-Driven Decision Making:**
- Analytics: The system generates valuable data on user preferences, helping restaurants and platforms make data-driven decisions to improve their services and offerings.
4. **Adaptability:**
- Dynamic Updates: The system can adapt to changing trends, ensuring that recommendations stay relevant and up to date with evolving consumer preferences.

DISADVANTAGES:

1. **Privacy Concerns:**
- Data Collection: The recommendation system relies on collecting and analyzing user data, raising concerns about privacy and data security. Users may be uncomfortable with the amount of personal information being used to tailor recommendations.

2. **Bias and Limited Diversity:**

- Algorithmic Bias: If the recommendation algorithm is not carefully designed, it may inadvertently reinforce biases, recommending similar types of restaurants and limiting diversity in suggestions.
- Popularity Bias: The system may favor popular or mainstream restaurants, potentially overshadowing hidden gems, or local establishments.

3. **Overreliance on Technology:**

- Loss of Human Element: Overreliance on algorithms may result in a loss of the human touch, with users missing out on the nuanced recommendations that a human expert might provide.

4. **Quality of Recommendations:**

- Inaccuracy: If the recommendation algorithm is not accurate, users may receive suggestions that do not align with their preferences, leading to dissatisfaction.
- Limited Context: The system may not always consider contextual factors such as special occasions, dietary restrictions, or specific user moods.

5. **Technical Challenges:**

- Maintenance and Updates: The system requires continuous maintenance and updates to remain effective. Technical issues or outdated algorithms may diminish the quality of recommendations.

CHAPTER 11
CONCLUSION

In this project, we developed a web app that recommends the restaurant based on the choice of your interest. This is used for the users to predict the suitable and best restaurant as per their tastes. Content-based filtering and collaborative-based filtering make the recommendation more efficient so that each user can use this application for their easy prediction of the restaurant.

The project Restaurant Recommendation System was successfully completed by using latent factor collaborative filtering or matrix factorization. In this project, we developed a model that could recommend the restaurant based on the chosen interest.

I learned how to design a recommendation system utilizing various recommendation methods while also grappling with issues such as which approach would be most appropriate for this system and how much redundancy to maintain, both of which were key components of the system.

A small-scale inquiry was carried out, and the findings were merged. This has helped me to gain a better understanding of the documentation, testing, and validation of hypotheses. Through testing to validate this hypothesis, feedback from users has been asked for, to obtain the most accurate possible results to predict a product. I learned about machine learning and artificial intelligence while working on this project, a field that is leading the road to a future world with more job options.

Switching from one technology to another has been a challenge in the beginning. This has allowed me to exceed myself and improve my capacity to adapt to new technology far more quickly than previously.

Finally, because of the scope of this project, I've had to learn to manage my work on my own, which was at first challenging, but now I know how to manage my time and set boundaries to organize my time.

CHAPTER 12

FUTURE SCOPE

The future of restaurant recommendation systems holds great promise, driven by advancements in technology and a growing emphasis on enhancing user experiences. One significant avenue for development lies in the realm of personalization. As artificial intelligence (AI) and deep learning techniques continue to evolve, recommendation systems can become even more adept at understanding nuanced user preferences. This enhanced personalization may encompass not only individual tastes but also consider contextual factors like the time of day, weather conditions, or special occasions, offering users recommendations that align seamlessly with their specific situations.

Another area of future development involves the integration of emerging technologies. Augmented Reality (AR) and Virtual Reality (VR) could play a pivotal role in transforming the way users interact with restaurant recommendations. Imagine a scenario where users can virtually explore restaurants or preview menu items in an immersive environment.

Collaborations with smart devices offer another exciting avenue for development. By working in tandem with smart home devices, recommendation systems can gain valuable insights into user behavior, preferences, and real-time data, offering a more refined and personalized service.

Additionally, there's potential for increased social integration, with recommendation systems leveraging social media data to provide insights into user preferences and trends. This could involve showcasing where friends have dined or highlighting trending restaurants, adding a social dimension to the dining experience.

Moreover, the future may witness a shift towards promoting local and sustainable options. Recommender systems could actively support local businesses, contributing to community well-being and environmental sustainability. Real-time updates and notifications, gamification elements such as reward systems, and an ongoing commitment to user-centric design are additional facets that could shape the future of restaurant recommendation systems.

CHAPTER 13

APPENDIX

GitHub & Project Demo Link

GitHub link:

https://github.com/smartinternz02/SI-GuidedProject-611904-1700064149

Project Demo link:

https://drive.google.com/file/d/1qZLHCKihHARo6fsP34HMKhjguyu4aQzg/view?usp=sharing