# Project Report

## Introduction

### Project Overview:

The Restaurant Recommendation System is an AI-powered tool designed to enhance the dining experience of users by providing personalized restaurant suggestions. This project aims to overcome the limitations of traditional recommendation methods, which often rely on suggestions from friends and are thus limited to their experiences.

Our system leverages advanced Convolutional Neural Networks (CNN) and web application development techniques to analyze user data, which includes past restaurant ratings, cuisine preferences, location, and dining history. By processing this data, the system can generate accurate and personalized restaurant recommendations.

Unlike traditional methods, our system is not limited to suggesting restaurants based on user's past behavior. It also considers other factors such as restaurant reviews, popularity, and proximity to the user's location. This ensures that the recommendations are not only tailored to the user's preferences but are also practical and convenient.

This project aims to bridge the gap between diners and restaurants, providing a win-win situation for both parties. Users get to discover and experience new dining options, and restaurants receive increased visibility and patronage.

The Restaurant Recommendation System is a step towards a more personalized and enjoyable dining experience, making restaurant selection easy, convenient, and tailored to each user's unique tastes.

### Purpose:

The purpose of the Restaurant Recommendation System project is to revolutionize the way users discover restaurants, making the process more personalized, efficient, and enjoyable. The system aims to address the limitations of traditional recommendation methods by providing suggestions based on a

comprehensive analysis of user preferences, past dining experiences, restaurant reviews, and location data.

The project seeks to:

1. Improve User Experience: By providing personalized restaurant recommendations, the system aims to enhance the dining experience of users, helping them discover new restaurants that align with their tastes and preferences.
2. Overcome Limitations of Traditional Methods: Traditional recommendation methods are often limited to the experiences of friends and acquaintances. This system aims to overcome these limitations by leveraging AI technologies to provide recommendations based on a wider range of data.
3. Benefit Restaurants: By increasing their visibility among potential customers, the system also aims to benefit restaurants. It can help restaurants reach out to a larger audience, potentially increasing their patronage.
4. Leverage AI Technologies: The project provides an opportunity to apply and showcase the capabilities of AI technologies like Convolutional Neural Networks (CNN) in a real-world application, demonstrating their potential in improving user experience in various domains.

In essence, the purpose of this project is to leverage AI technologies to bridge the gap between diners and restaurants, making restaurant selection a more personalized and enjoyable experience.

# Literature Survey

## Existing Problem:

1. Lack of Data: Recommender systems need a lot of data to effectively make recommendations. This could be a significant challenge if your system doesn't have enough user data to start with.
2. Changing User Preferences: User preferences can change over time, making it difficult for the system to provide accurate recommendations.
3. Cold Start Problem: This is a common issue in recommendation systems where the system doesn't have enough data about new users or new items to make accurate recommendations.
4. Scalability: As the number of users and restaurants increases, the system should be able to scale and provide recommendations efficiently.
5. Long Tail Problem: Recommending less popular or less visited restaurants can be challenging.

6. Accuracy of Prediction: The system should be able to accurately predict the user's preferences
7. Novelty and Diversity: The system should be able to recommend new and diverse restaurants to the users.
8. Sparse, Missing, Erroneous and Malicious Data: The system should be robust to handle sparse data, missing values, erroneous data, and potential malicious data.
9. Unpredictable Items: Some items can be unpredictable in their popularity, making it difficult for the system to recommend them.
10. Complexity: Building a recommender system can be complex, involving various factors like user preferences, item characteristics, and contextual information.

## References:


1. https://www.cs.cornell.edu/~rahmtin/Files/YelpClassProject.pdf
2. https://github.com/topics/restaurant-recommendation
3. adarsh18raj/Yelp-Restaurant-Recommendation-System - GitHub
4. https://thecleverprogrammer.com/2022/07/26/restaurant-recommendation-system-using-python/
5. https://www.kaggle.com/code/midouazerty/restaurant-recommendation-system-using-ml


## Problem Statement Definition

In the current digital age, users rely heavily on online platforms to find restaurant recommendations. However, existing systems often fail to provide personalized and diverse suggestions, limiting users to popular or well-known places. This lack of personalization can lead to unsatisfactory dining experiences and missed opportunities to discover new and unique restaurants.

Furthermore, these systems often struggle with the "cold start" problem, where they fail to provide accurate recommendations for new users due to a lack of historical data. They also face challenges in adapting to changing user preferences over time.

The goal of this project is to develop an Artificial Intelligence-based Restaurant Recommendation System that addresses these issues. The system should be capable of providing personalized, diverse, and accurate restaurant recommendations to users, taking into account their unique preferences and dining history. It should also be robust

enough to handle the “cold start” problem and adapt to changing user preferences over time.

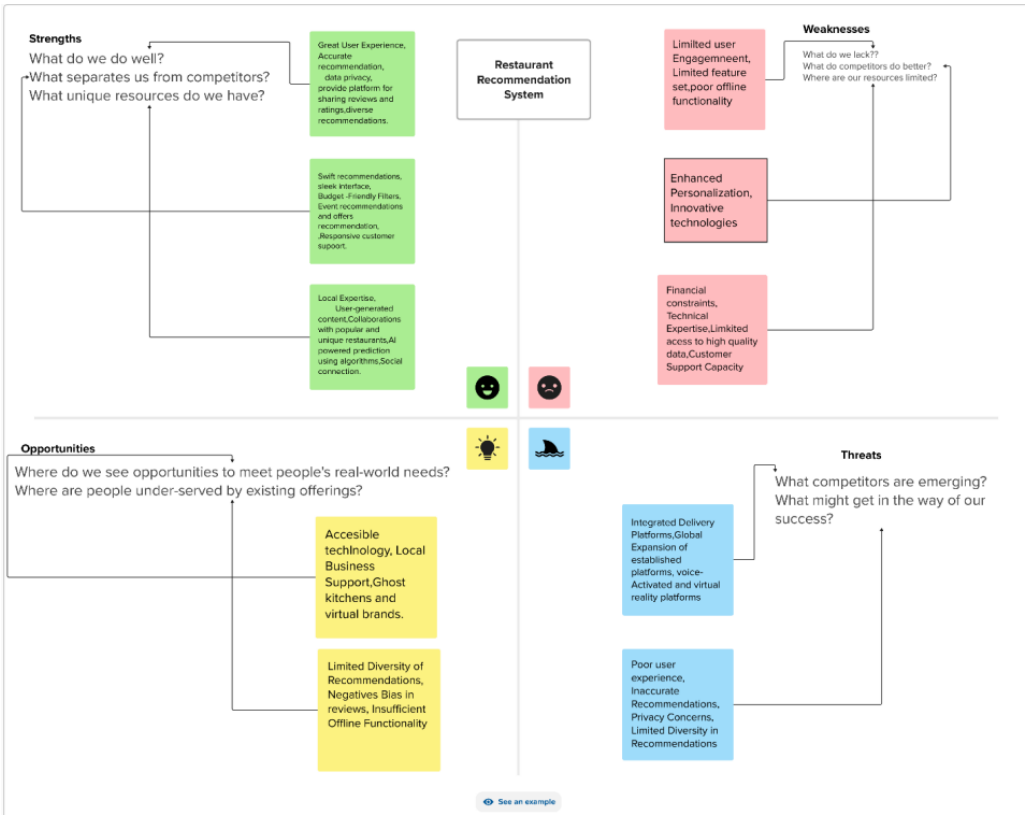
The success of this project will be measured by the accuracy of the recommendations provided, the diversity of the restaurants suggested, and the overall user satisfaction with the recommended restaurants.

This project requires skills in Convolutional Neural Networks (CNN) and Web App development. The system will be developed and tested using real-world data, ensuring its practical applicability and effectiveness.

By addressing these challenges, this project aims to enhance the dining experience of users and help them discover new and unique restaurants that align with their preferences.

## IDEATION & PROPOSED SOLUTION

### Empathy Map Canvas:

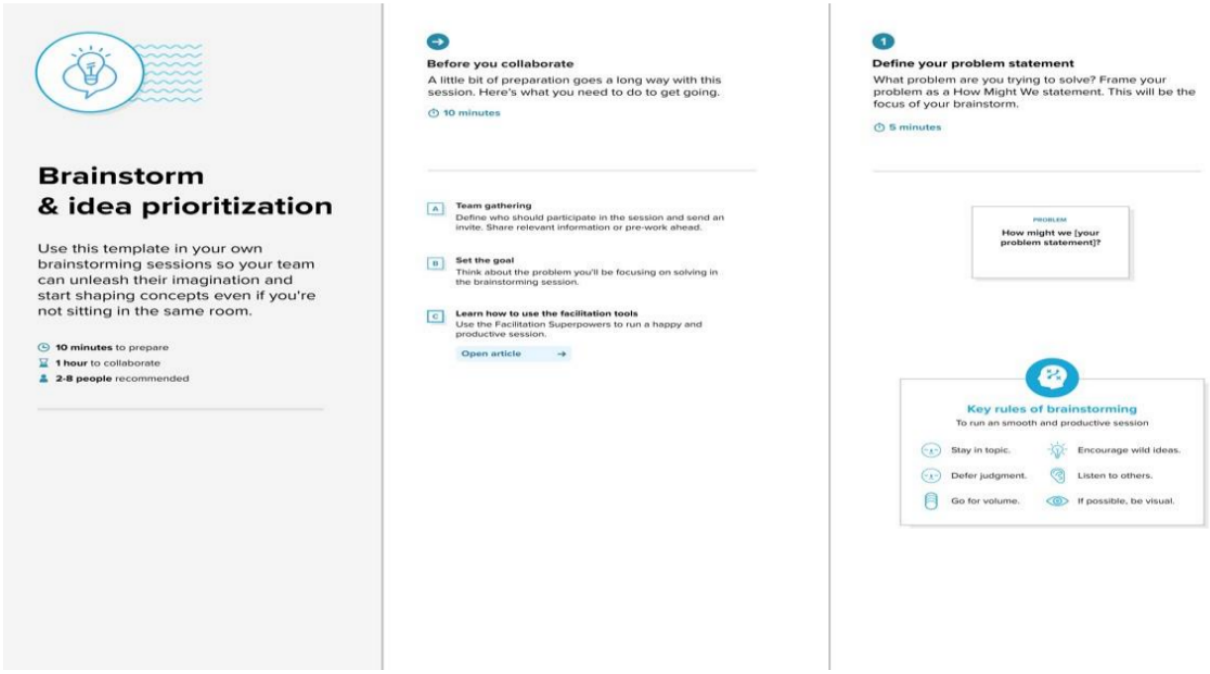


## Ideation & Brainstorming:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich number of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference: https://www.mural.co/templates/ideate

Step-1 : Team Gathering, Collaboration on and Select the Problem Statement




Step 2:- Brainstorm,Idea listing and Grouping

**Aditya Kumar Jha**

Advanced Personalization:

Implement machine learning algorithms to understand individual preferences better and provide highly personalized restaurant recommendations based on past behavior, ratings, and context.

Real-Time Contextual Recommendations:

Offer real-time recommendations based on factors like weather, time of day, and local events to provide users with contextually relevant suggestions.

Collaborative Filtering:

Introduce collaborative filtering techniques to enhance recommendations by analyzing user behavior and preferences, fostering a sense of community within the user base.

**Anish Singh**

Blockchain for Transparent Reviews:

Explore the use of blockchain technology to create a transparent and tamper-proof system for user reviews, ensuring the authenticity of feedback.

Augmented Reality (AR) for Menu Exploration:

Implement AR features that allow users to explore restaurant menus in a more interactive and engaging way, visualizing dishes and getting additional information about ingredients.

Dynamic Pricing Information:

Provide real-time pricing information for menu items, allowing users to make informed decisions based on current costs.

Prajwal Jha

Offline Functionality:

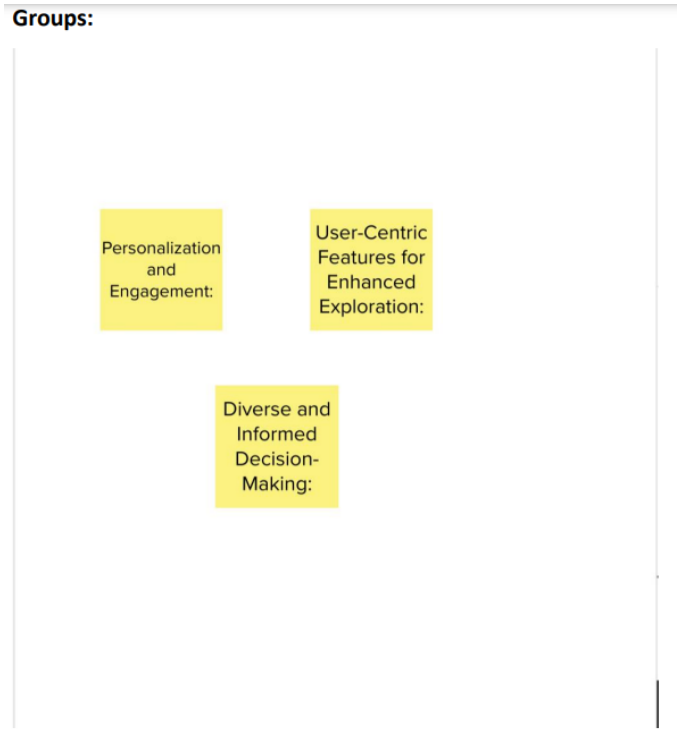
Enhance offline functionality to enable users to access basic features and recommendations even in areas with poor or no internet connectivity.

Health and Wellness Focus:

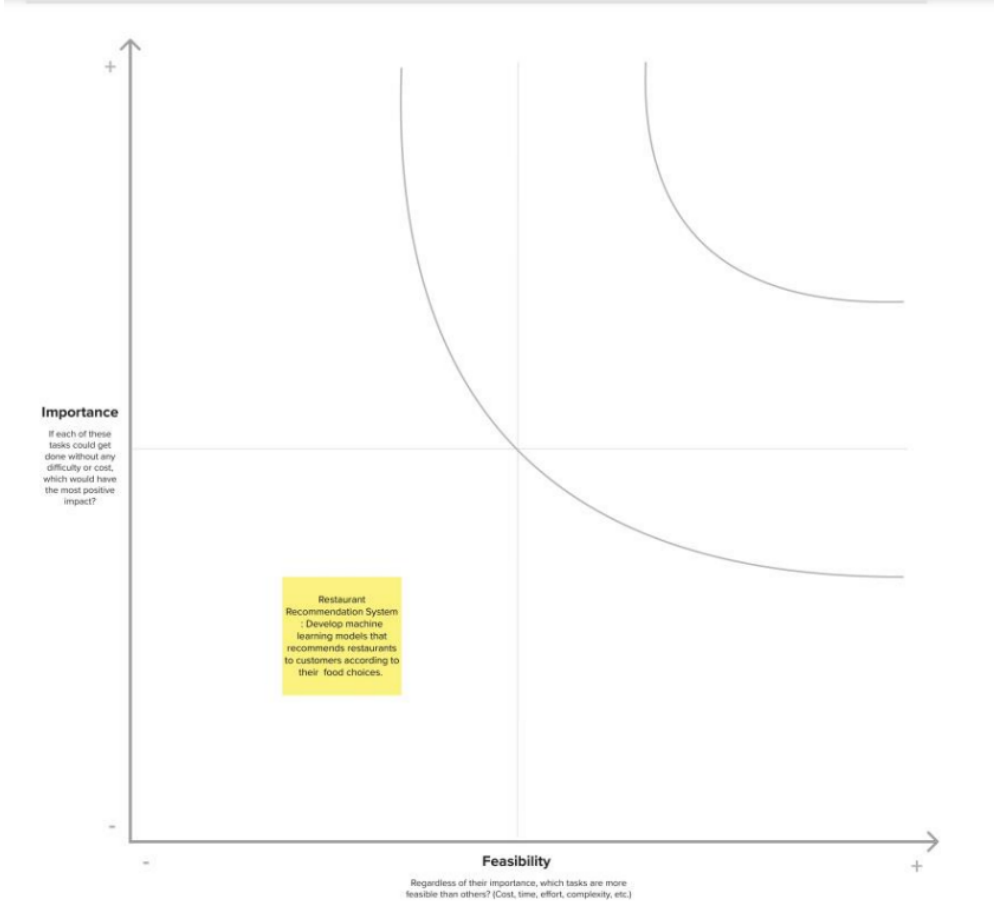
Incorporate features that highlight healthier dining options, provide nutritional information, and support users in making choices aligned with their health goals.

Diverse Dietary Preferences:

Expand the system to cater to various dietary needs, including vegetarian, vegan, gluten-free, and other specific preferences to accommodate a broader range of users.




Step 3: Idea Prioritization



# REQUIREMENT ANALYSIS

## Functional Requirement:

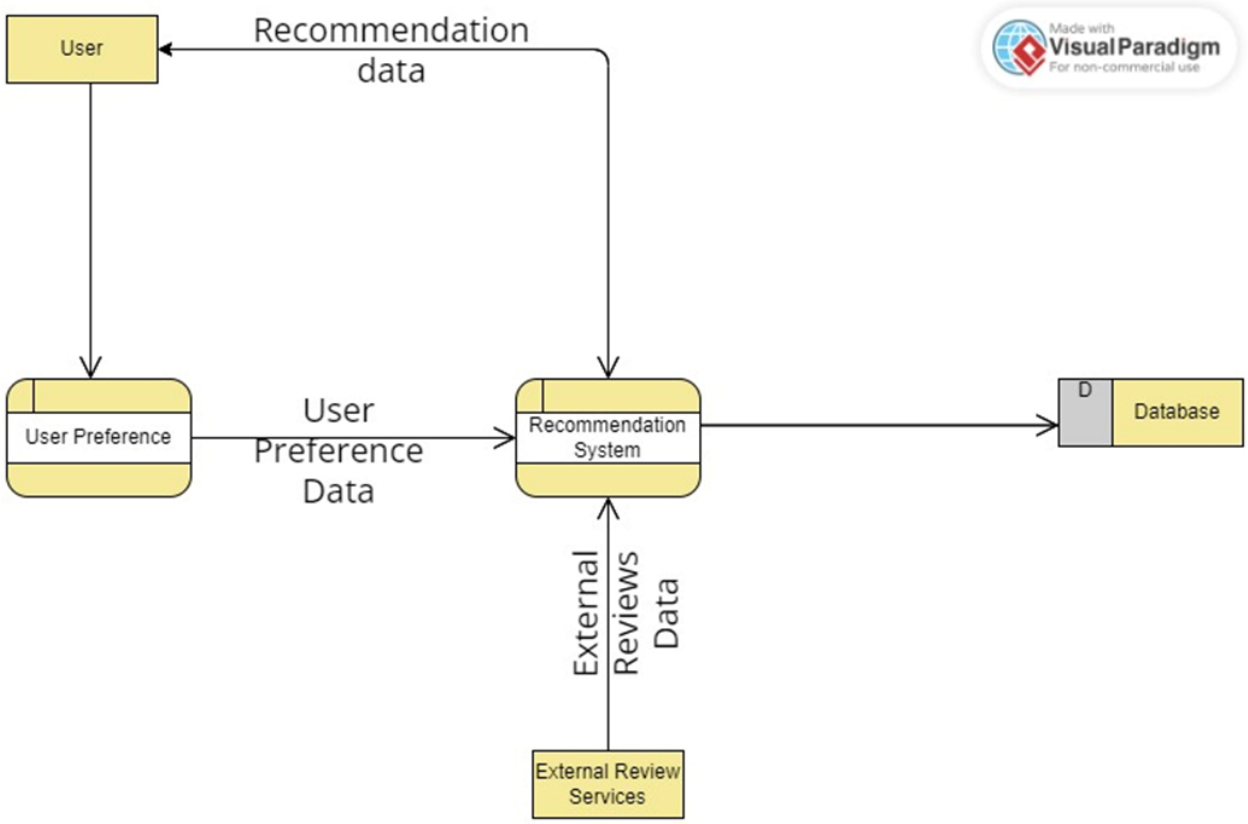
1. **User Registration:** The system should allow new users to register and existing users to log in.
2. **Search Functionality:** Users should be able to search for restaurants based on various parameters such as location, cuisine, price range, etc.
3. **Recommendation Engine:** The system should provide restaurant recommendations based on the user's past ratings, preferences, and search history.
4. **Ratings and Reviews:** Users should be able to rate and review restaurants. These ratings and reviews should be visible to other users.
5. **Reservation System:** The system should allow users to make reservations at restaurants.
6. **Location Services:** The system should be able to suggest restaurants in the user’s vicinity

## Non-Functional Requirement:

1. **Usability:** The system should provide an interactive user-friendly interface that is easily understandable for all users.
2. **Performance:** The system should be able to handle multiple requests at a time and provide fast response times.
3. **Security:** User data should be securely stored and transmitted. The system should also have measures in place to prevent unauthorized access.
4. **Availability:** The system should be available at least during the restaurant operating hours and must be recovered within an hour or less if it fails.
5. **Scalability:** The system should be able to handle an increasing amount of work by adding resources.
6. **Maintainability:** The system should be designed in a way that allows for easy updates and improvements.

# PROJECT DESIGN

Data Flow Diagrams & User Stories:



**User Stories**

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Regular User | Personalized Recommendations | USN-1 | As a regular user, I want personalized recommendations when I use the app. | 1.User receives restaurant recommendations based on preferences.<br>2.Recommendations include user's preferred cuisine types.<br>3.User can easily view details and reviews for recommended places. | High | v1.0 |
| New User | Onboarding Experience | USN-2 | As a new user, I want a seamless onboarding experience to get started with the app. | 1. User can easily sign up using email or social media accounts.<br>2.User is prompted to set initial preferences during onboarding.<br>3. Clear instructions and tooltips guide the user through the process. | Medium | v1.0 |
| Premium User | Exclusive Features | USN-3 | As a premium user, I want exclusive features and benefits to enhance my experience. | 1. Premium users have access to a wider range of restaurant filters.<br>2.Premium users receive priority access to newly added restaurants.<br>3.Premium users can bookmark and share personalized lists of favorite restaurants. | Low | v1.1 |

# Solution Architecture

### 1. Business Problem and Tech Solution:

**Problem:**

- Users want efficient and personalized restaurant recommendations beyond their immediate social circles.

**Solution:**

- A Restaurant Recommendation System utilizing machine learning algorithms for personalized suggestions.

### 2. Structure, Characteristics, and Behavior:

**Structure:**

1. **User Interface (UI):**
    - Responsive and intuitive design for user interaction.
2. **Application Layer:**
    - Implements business logic and core functionalities.
    - Manages user requests, communication with the database, and external services.
3. **Database:**
    - Stores user profiles, restaurant data, and preferences.
    - Provides efficient data retrieval for recommendation generation.
4. **Recommendation Engine:**
    - Utilizes machine learning algorithms for personalized restaurant suggestions.
5. **APIs:**
    - Facilitates communication between components.
6. **Authentication and Authorization:**
    - Ensures secure user access.
7. **External Services Integration:**
    - Integrates with external services for additional data (reviews, ratings).
8. **Caching Mechanism:**
    - Improves performance by caching frequently accessed data.
9. **Logging and Monitoring:**
    - Monitors system health, logs events for debugging.

**Characteristics:**

- Scalable, secure, and efficient in processing and retrieving data.
- Adaptable to changing user preferences and business requirements.

**Behavior:**

- Responsive UI for user interaction.
- Real-time recommendation generation based on user input.
- Continuous learning from user feedback to enhance future recommendations.

### 3. Features, Development Phases, and Solution Requirements:

**Features:**

1. **User Registration and Authentication:**
    - Secure user registration and login functionality.
2. **User Profile Management:**
    - Allows users to manage their preferences and provide feedback.
3. **Restaurant Recommendation Generation:**
    - Utilizes machine learning algorithms for personalized suggestions.
4. **External Services Integration:**
    - Retrieves and integrates data from external review platforms.
5. **User Feedback Loop:**
    - Allows users to provide feedback on recommended restaurants.

**Development Phases:**

1. **Phase 1 - MVP (Minimum Viable Product):**
    - User registration, login, and basic recommendation functionality.
2. **Phase 2 - Enhanced Features:**
    - User profile management, external services integration, and improved recommendation algorithms.
3. **Phase 3 - Optimization and Scaling:**
    - Database optimization, caching mechanisms, and scalability improvements.

**Solution Requirements:**

- Mobile responsiveness for widespread accessibility.
- Integration with popular external review platforms.
- Continuous improvement of recommendation algorithms based on user feedback.

### 4. Specifications for Solution Management and Delivery:

1. **Agile Development:**
    - Adopt an agile development methodology for iterative releases and quick adaptation to changing requirements.
2. **Continuous Integration/Continuous Deployment (CI/CD):**
    - Implement CI/CD pipelines for automated testing, deployment, and rapid delivery.
3. **User Training and Support:**

- Provide user training resources and support for efficient onboarding.

4. **Security Measures:**
   - Regularly update security protocols to address emerging threats.
5. **Scalability Plan:**
   - Outline a scalability plan for future growth and increased user demand.
6. **Documentation:**
   - Maintain comprehensive documentation for stakeholders and future development teams.

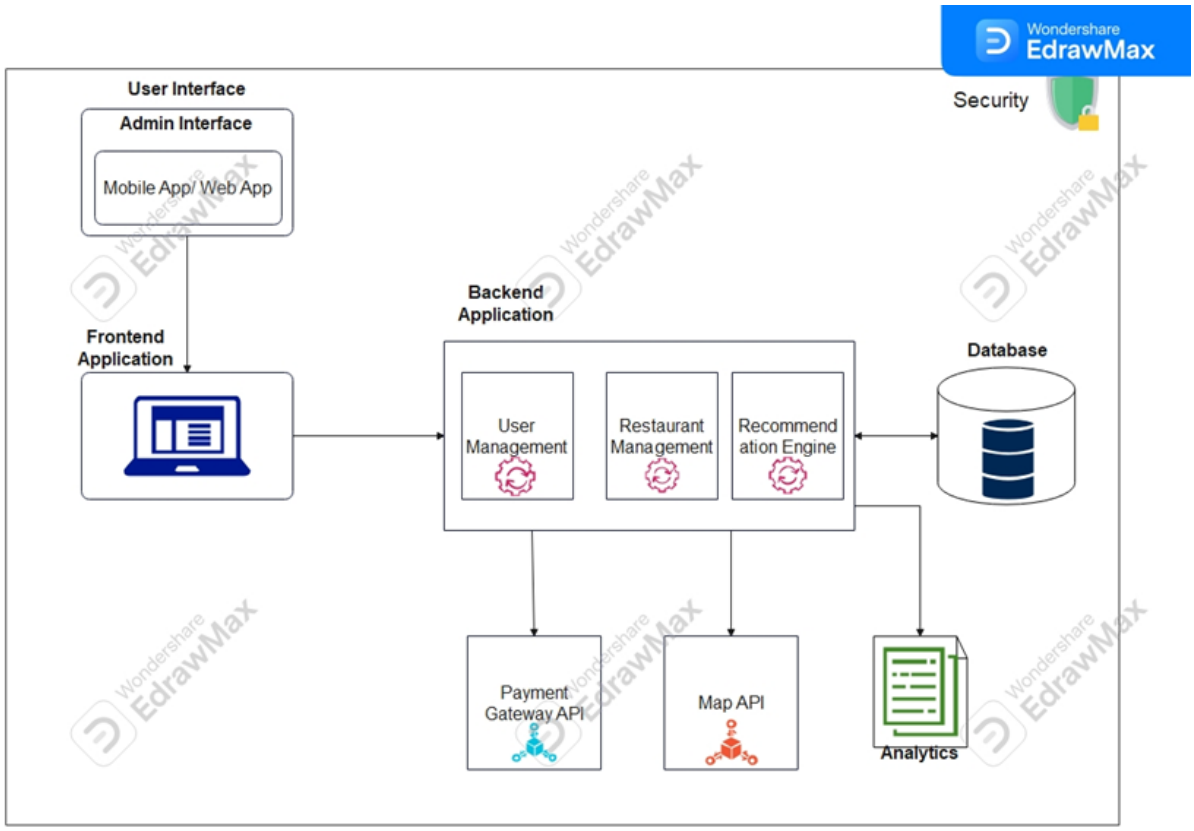
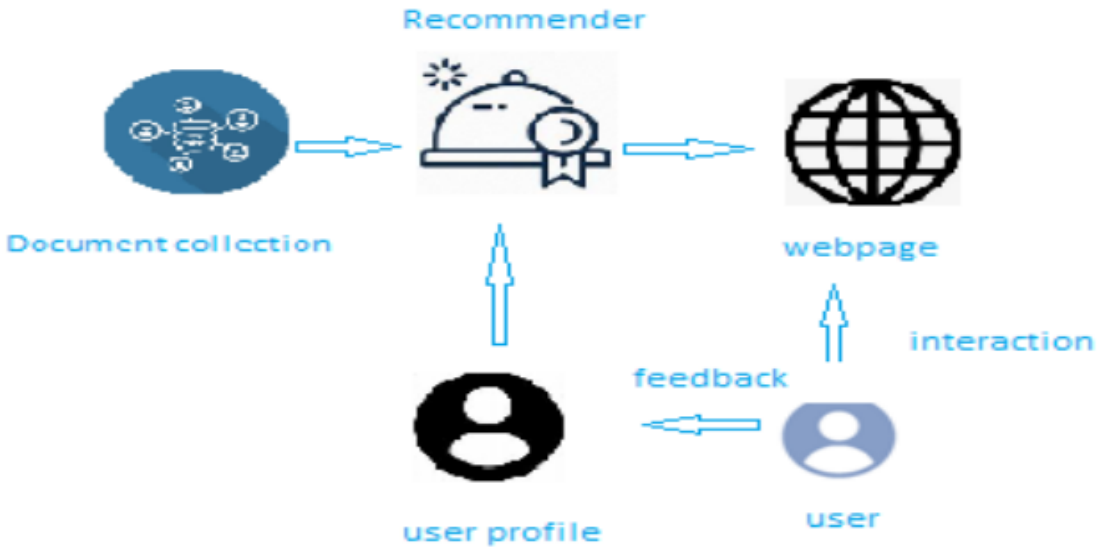



Fig: Solution architecture diagram for restaurant recommendation system

# PROJECT PLANNING & SCHEDULING

Technical Architecture:

## ARCHITECTURE




Sprint Planning & Estimation

| Sprint | Functional | User | User Story / Task | Story | Priority | Team |
|---|---|---|---|---|---|---|

| | Requirement (Epic) | Story Number | | Points | | Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Ideation Phase | USN-1 | Complete all the template in the Ideation Phase and submit it to the Github Repo | 2 | Medium | Prajwal |
| Sprint-2 | Project Design Phase | USN-2 | Complete all the template in the Project Design Phase and submit it to the Github Repo | 2 | Medium | Anish |
| Sprint-3 | Project Planning Phase | USN-3 | Complete all the template in the Project Planning Phase and submit it to the Github Repo | 2 | Medium | Aditya |
| Sprint-4 | Model Building | USN-4 | Build the AI model for Recommender System | 3 | High | Aditya |
| Sprint-5 | Back end web application | USN-5 | Build the Web application | 3 | High | Anish |
| Sprint-6 | Front End Development | USN-6 | Web development to integrate the model with UI | 3 | High | Prajwal |
| Sprint-7 | Performance and Testing | USN-7 | Check for the performance of the code and test the result.Upload the necessary document in git repo | 2 | Medium | Aditya |
| Sprint-8 | Final Submission | USN-8 | Final report making and Submitting | 3 | High | Anish Prajwal |

# Sprint Delivery Schedule

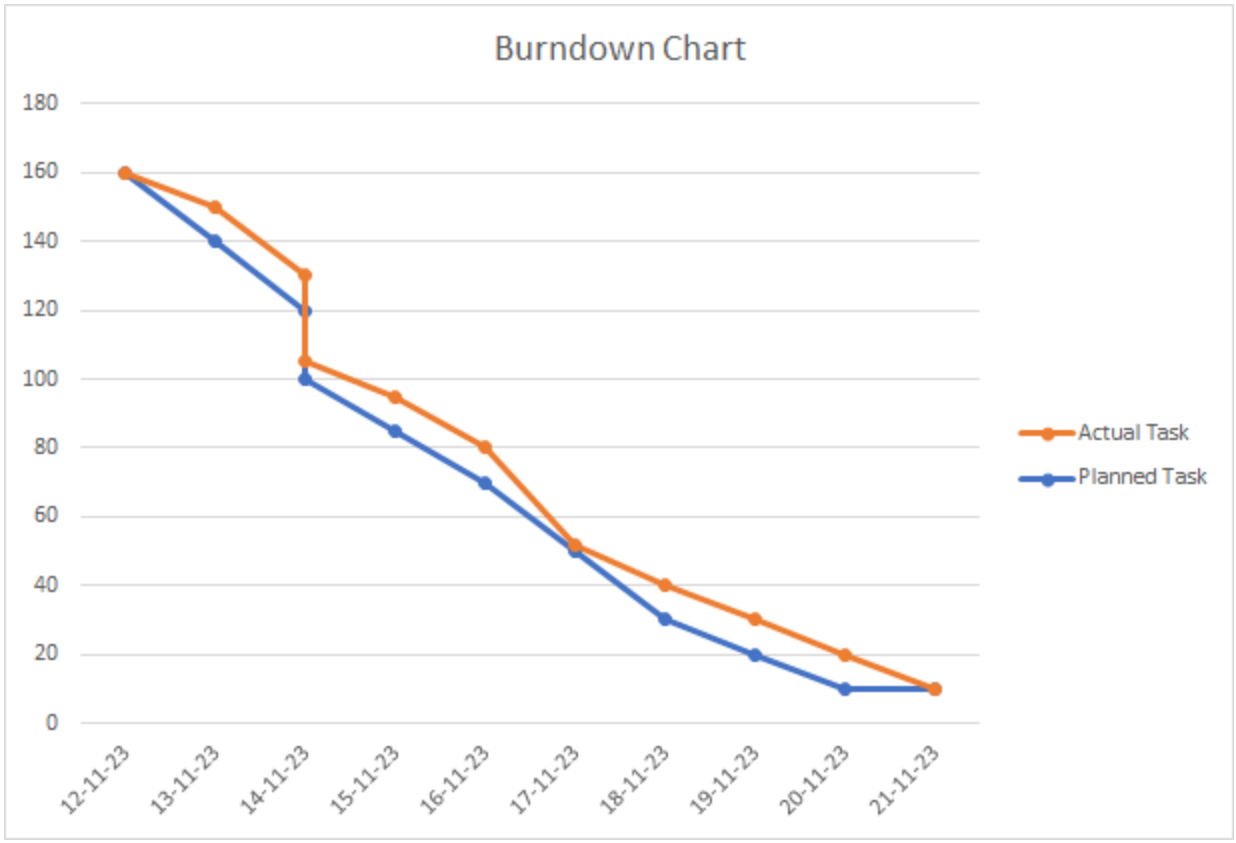
**Project Tracker, Velocity & Burndown Chart: (4 Marks)**

| 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 | 1 Day | 12 Nov 2023 | 12 Nov 2023 | 20 | 12 Nov 2023 |
| Sprint-2 | 20 | 2 Days | 19 Nov 2023 | 20 Nov 2023 | 20 | 20 Nov 2023 |
| Sprint-3 | 20 | 2 Days | 13 Nov 2023 | 14 Nov 2023 | 20 | 14 Nov 2023 |
| Sprint-4 | 20 | 2 Days | 15 Nov 2023 | 16 Nov 2023 | 20 | 16 Nov 2023 |
| Sprint-5 | 20 | 2 Days | 16 Nov 2023 | 18 Nov 2023 | 20 | 18 Nov 2023 |
| Sprint-6 | 20 | 2 Days | 18 Nov 2023 | 19 Nov 2023 | 20 | 19 Nov 2023 |
| Sprint-7 | 20 | 2 Days | 19 Nov 2023 | 20 Nov 2023 | 20 | 20 Nov 2023 |
| Sprint-8 | 20 | 2 days | 20 Nov 2023 | 21 Nov 2023 | 20 | 21 Nov 2023 |

**Velocity:**

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

AV= 160/15 =10.667


Burndown Chart
180
160
140
120
100
80
60
40
20
0
12-11-23
13-11-23
14-11-23
15-11-23
16-11-23
17-11-23
18-11-23
19-11-23
20-11-23
21-11-23
Actual Task
Planned Task

# CODING & SOLUTIONING

Feature 1:

### Recommendation Function:

**Cosine Similarities:** The code computes cosine similarities between restaurant reviews using the linear kernel.

**Recommendation Logic:** The `recommend` function takes a restaurant name as input, calculates cosine similarities, and recommends similar restaurants based on the 'Mean Rating.'

**Data Filtering:** The recommendations are filtered, sorted, and limited to the top 10 based on the 'Mean Rating.'

Feature 2:

### Web Interface:

- **HTML Templates:** The code uses Flask's `render_template` to render HTML templates for the main page ('index.html') and the recommendations page ('recommendations.html').
- **Form Submission:** The user enters a restaurant name in a form on the main page. When submitted, the form data is sent to the `/recommendations` route for processing.
- **Output Format:** The recommendations are returned in the form of a dictionary suitable for rendering in the HTML template.

# PERFORMANCE TESTING

The model is related to creating a restaurant recommendation system using content-based filtering. However, it doesn't include any explicit training or validation steps. Content-based filtering models like the one shown in the code don't typically involve a traditional training and validation process with accuracy metrics as you might find in supervised machine learning tasks.

Content-based filtering relies on the characteristics of items (in this case, restaurants) and user preferences to make recommendations. The TF-IDF vectorization and cosine similarity calculations are methods for understanding the content of the items and finding similar items based on that content.

# RESULTS

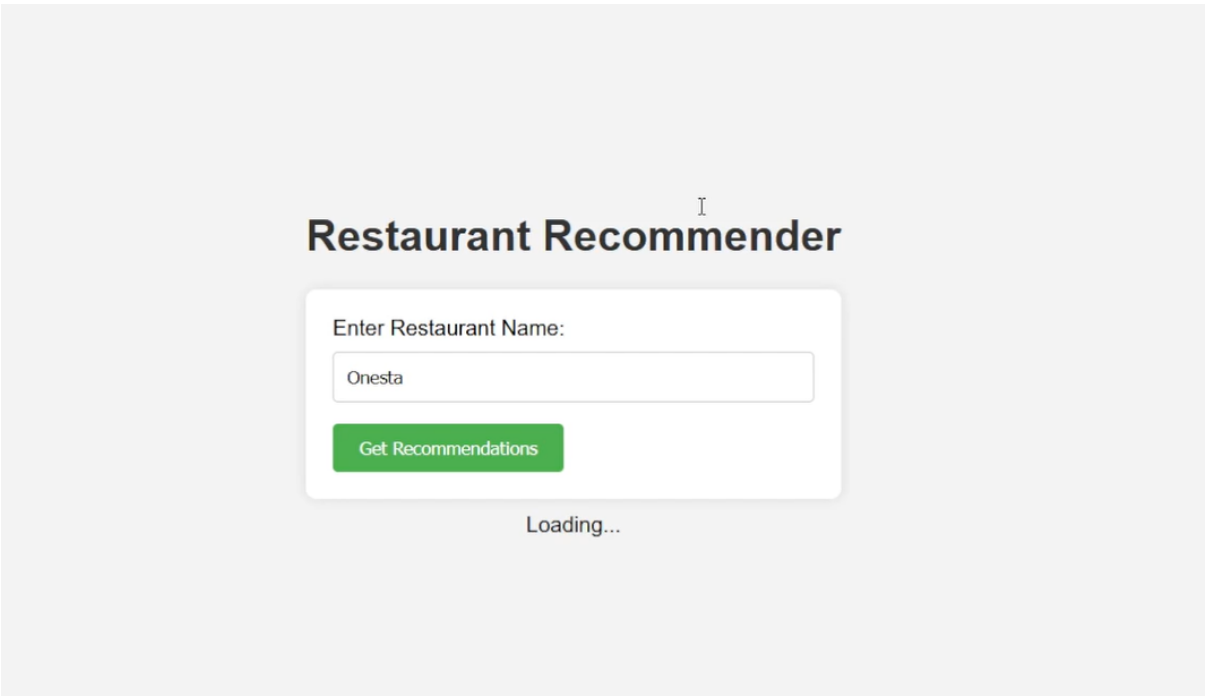
## Output Screenshots:

Jupyter Notebook:

Out[63]:

| | cuisines | Mean Rating | cost |
|---|---|---|---|
| **Onesta** | Pizza, Cafe, Italian | 4.37 | 600.0 |
| **Ayodhya Upachar** | South Indian, North Indian, Chinese, Street Food | 4.23 | 200.0 |
| **The Coffee Shack** | Cafe, Chinese, Continental, Italian | 4.10 | 500.0 |
| **Upahara Darshini** | South Indian, North Indian, Chinese | 3.81 | 400.0 |
| **Faasos** | North Indian, Biryani, Fast Food | 3.75 | 500.0 |
| **The Lassi Park** | Juices, Desserts | 3.73 | 200.0 |
| **CakeZone** | Bakery, Desserts | 3.72 | 200.0 |
| **Cake Town Cafe** | Bakery, Desserts | 3.71 | 500.0 |
| **Matru Sagar** | South Indian, North Indian | 3.59 | 300.0 |
| **Cafe Monarch Luxur** | North Indian, South Indian, Chinese, Fast Food | 3.58 | 800.0 |

Website:




## Top 10 Recommendations

| Restaurant Name | Cuisines | Mean Rating | Cost |
|---|---|---|---|
| Onesta | Pizza, Cafe, Italian | 4.37 | 600.0 |
| Salut | Continental, Finger Food, Seafood, Pizza | 3.97 | 1.2 |
| SLV Corner Restaurant | South Indian, Chinese, North Indian, Street Food, Rolls, Juices | 3.97 | 550.0 |
| Italino | Italian, Mexican | 3.95 | 900.0 |
| Dadi's Dum Biryani | Biryani, North Indian, Mughlai | 3.94 | 800.0 |
| Little Cafe | Ice Cream, Desserts, Beverages, Sandwich | 3.88 | 400.0 |
| Matru Sagar | North Indian, South Indian | 3.85 | 1.0 |
| Cafe Aira | Cafe, Desserts | 3.84 | 700.0 |
| South Kitchen | Kerala, North Indian | 3.71 | 350.0 |

# ADVANTAGES & DISADVANTAGES

Some advantages and disadvantages of a restaurant recommendation system are :

**Advantages:**

1. **Personalized Suggestions:** Recommendation systems provide personalized suggestions by analyzing user data and understanding individual preferences.
2. **Efficient Search:** They reduce search and evaluation time, drive sales, and introduce new items to consumers.
3. **Increased Sales:** Businesses can build loyalty and drive sales through differentiated experiences.
4. **Discovery of New Options:** They can show users products that they have not seen in the past and might like.
5. **Adaptability:** Recommendation systems can evolve and adapt to changing user preferences.

**Disadvantages:**

1. **Shaping Preferences:** Recommendations do more than just reflect consumer preferences — they actually shape them, which can fuel biases and affect sales in unexpected ways.
2. **Limited Recommendations:** The recommendations from friends or other common people are limited to those places they have visited before.
3. **Lesser Interaction:** There might be lesser interaction between the waitstaff and the customer, impacting the overall customer experience.
4. **Complex Algorithms:** Recommendation systems operate on complex algorithms that process large volumes of data, which might be challenging to manage.
5. **Difficulty in Finding Appropriate Features:** Finding the appropriate features for recommendation can be hard.

# CONCLUSION

In conclusion, a restaurant recommendation system has the potential to be an effective tool for patrons and companies alike. Businesses can benefit from personalized recommendations, effective search, and more sales, and users can find new possibilities and be flexible to suit their evolving preferences. But it also has its own set of drawbacks, including determining the right features, influencing user preferences, providing few recommendations, reducing engagement, and using complicated algorithms.

The way such a system is designed and implemented will determine how successful it is. Before beginning the development process, it is imperative to carry out a comprehensive requirements analysis and take both functional and non-functional needs into account. Notwithstanding the difficulties, a well-executed restaurant recommendation system can greatly improve patrons' dining experiences while offering businesses insightful data.

# FUTURE SCOPE

The future scope of a restaurant recommendation system is vast and promising. Here are some potential areas of growth and improvement:

1. **Advanced Technologies:** With the advent of cloud-based hosting, next-generation restaurant management, and point-of-sale (POS) systems, the future of restaurant success lies in technology.These systems have improved by leaps and bounds in terms of functionality.
2. **Improved Customer Experience:** 95% of restaurant operators view the ability to improve guest satisfaction and the quality of the guest experience as one of the biggest benefits one can expect to gain with the right restaurant management and POS system.
3. **Operational Efficiency:** 89% of restaurant operators cite the need to drive increased operational efficiency, including increasing staff productivity and reducing inventory waste, as a key success factor.
4. **Performance Improvement:** 78% of full-service restaurants and 62% of quick-service and fast-casual restaurants achieved “significant” or “dramatic” improvement in both operations and revenue performance after deploying a next-generation restaurant management and POS system.
5. **Evolution of Recommender Systems:** The current and future state of recommender systems is evolving. They were simple in their early days, and have since evolved into more complex models.The common idea in different recommender systems is that there needs to be some measure of similarity.
6. **Hybrid Approach:** Recommender systems could also combine item-based and customer-based methods into a hybrid approach which can benefit from the strengths of both methods.
7. **Healthy Food Domain:** There is also a scope for recommendation techniques for individuals and groups in the healthy food domain.
8. **Leveraging Data and Automation:** Future restaurants will need to focus on leveraging data and automation to give their guests the hospitality they expect.
9. **Content-Based Recommender System:** The aim is to create a content-based recommender system in which when we will write a restaurant name, the Recommender system will look at the reviews of other restaurants, and the System will recommend us other restaurants with similar reviews and sort them from the highest-rated.

In conclusion, the future of restaurant recommendation systems is bright and holds immense potential.

# APPENDIX

GitHub & Project Demo Link:

https://github.com/smartinternz02/SI-GuidedProject-615819-1700656792.git

https://drive.google.com/file/d/1mTGQKqKTnoTWND6IzlJ11S2ZxqG02FEF/view?usp=sharing