1. **Project Background:**

In today's digital age, a Restaurant Recommender is an intelligent system designed to suggest restaurants based on user preferences, location, cuisine preferences, reviews, and other factors. Using machine learning and data analysis, it tailored recommendations to individual tastes, ensuring users discover new and highly rated dining experiences. Whether you're searching for a hidden gem, a trendy spot, or a budget-friendly option, a Restaurant Recommender makes dining decisions effortless and personalized.

1. **Project Goal:**

To address this, we will develop **RestaurantRecommender AI,** a chatbot that combines the power of large language models and rule-based functions to ensure accurate and reliable information delivery.

1. **Problem Statement:**

Given Zomato restaurants dataset from Kaggle containing information about restaurants (restaurant names, restaurant types, rating, average cost, cuisine type, area etc.), I will build a chatbot that parses the dataset and provides accurate restaurants recommendations based on the user requirements.

Link to dataset: <https://www.kaggle.com/datasets/abhijitdahatonde/zomato-restaurants-dataset>

1. **Approach:**
2. **Conversation and Information Gathering**: The chatbot will utilize language models to understand and generate natural responses. Through a conversational flow, it will ask relevant questions to gather information about the user's requirements.
3. **Information Extraction**: Once the essential information is collected, rule-based functions come into play, extracting top 3 restaurant that best matches the user's needs. We will be using function calling to extract information from the csv file & for data comparisons
4. **Personalized Recommendation**: Leveraging this extracted information, the chatbot engages in further dialogue with the user, efficiently addressing their queries and aiding them in finding the perfect restaurant
5. **Design choice:**

Please refer to the document attached, ‘**SystemDesign\_Sahil\_Avasthi.pptx’**

1. **Detail understanding of rule-based function / compare\_restaurants\_with\_user\_described\_features function:**

There will be 8 parameters to be used for scoring,

1. area
2. restaurant type
3. online order
4. table booking
5. minimum number of reviews
6. minimum rating
7. average cost for 2 persons
8. cuisines type

The following parameters are mandatory for scoring: **area, restaurant type, minimum rating, average cost for 2 persons, and cuisine type**. If any of these do not match user requirements, that record will be excluded from the comparison. However, **Online order, table booking, and minimum reviews are optional**. Even if they do not match then other parameters will still be used to generate a score.

For each match, the score will be incremented by 1. However, if the cuisine type has multiple matches, such as when a user requests only North Indian cuisine and the record includes North Indian, Seafood, and Chinese cuisines, the cuisine type will receive a score of 2 (but no more than that)

Other than the score we will use another parameter **budget\_remaining**. The value of budget\_remaining will be calculated as,

* user budget minus average cost of 2 persons in the restaurant.
  + For example, if user budget is 5000 INR and average cost of 2 persons is 400 INR then budget remaining will be 4600 INR

Finally, the dataset will be sorted based on 3 parameters, score, budget\_remaining and rating all in the **descending order**,

* Records with higher scores will appear first in the recommendation list
* If the scores are equal, the record with a higher remaining budget will be recommended
* If the score, remaining budget are equal the record with the higher rating will be recommended
* If all options are identical, select either one

1. **Challenges faced:** Writing the scoring logic with consistent response