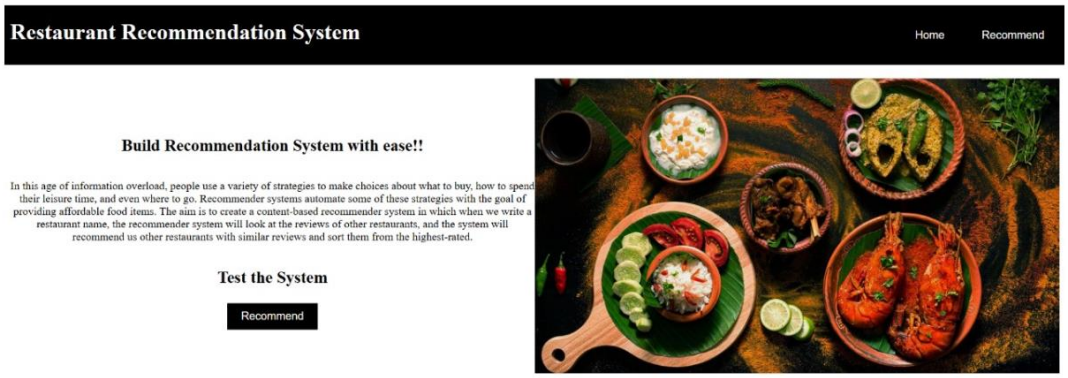
Restaurant Recommendation System

Project Description

- This project focuses on developing a smart restaurant recommendation system that delivers personalized suggestions based on user preferences.
- A hybrid recommendation model combining **collaborative filtering** and **content-based filtering** techniques is implemented to provide accurate and relevant results.
- The system adapts dynamically based on user inputs and geolocation data, enhancing the user's decision-making process when exploring dining options.

Screenshots

Home Page:



Input Page:

Recommendation Output:

Name	Cuisines	Mean Rating (out of 5)	Cost (in thousands)
The Black Pearl	north indian european mediterranean bbq	4.85	1.5
Barbeque Nation	north indian european mediterranean bbq kebab	4.7	1.6
Hunger Camp	north indian south indian chinese seafood	4.56	1.3
Hakuna Matata	north indian asian seafood chinese	4.41	1.2
Jalsa Gold	north indian mughlai italian	4.41	1.3
Deja Vu Resto Bar	north indian italian	4.26	900.0
Tipsy Bull - The Bar Exchange	north indian chinese continental mexican	4.26	1.4
Dhaba Estd 1986 Delhi	north indian	4.26	1.1
Float	north indian japanese	4.26	1.5
nu.tree	north indian healthy food beverages	4.26	400.0

Installation and Setup

Using Conda (Recommended)

```
# Create a new conda environment
conda create -n restaurant_recommender python=3.10

# Activate the environment
conda activate restaurant_recommender

# Clone the repository
git clone https://github.com/Rohitmh09/Restaurant-Recommendation-System.git
cd Restaurant-Recommendation-System/Flask

# Install dependencies
pip install -r requirements.txt
```

Using Python venv

```
# Create a virtual environment
python -m venv venv

# Activate the environment
# On Windows:
venv\Scripts\activate
# On macOS/Linux:
source venv/bin/activate

# Install dependencies
pip install -r requirements.txt
```

Running the Application

```
# Navigate to the Flask directory
cd Flask

# Run the Flask web application
python app1.py

# Visit http://localhost:5000 to access the app
```

Project Structure:

```
Restaurant-Recommendation-System/
├── Documentation...
├── Flask/
│   ├── __pycache__/               # Compiled Python cache files
│   ├── static/                    # Static assets (Images)
│   ├── templates/                 # HTML templates for the frontend
│   ├── app1.py                    # Flask application entry point
│   ├── Final_Development_Phase.ipynb  # Flask dev notebook
│   ├── requirements.txt           # Python dependencies
│   └── restaurant1.csv            # Restaurant dataset
├── Model/
    └── Final_Development_Phase.ipynb  # Model training and evaluation notebook
```

Technologies Used

- **Python 3.10** – Core programming language
- **Flask** – Lightweight web framework to build the backend API
- **Scikit-learn** – For building and evaluating machine learning models
- **Surprise** – Specialized library for collaborative filtering (e.g., SVD)
- **Pandas / NumPy** – For data manipulation and preprocessing
- **NLTK** – For natural language processing and cleaning review text
- **Matplotlib / Seaborn / Plotly** – For data visualization and EDA
- **HTML / CSS / JavaScript** – For designing the frontend interface

Model Architecture

This project uses a **Hybrid Recommendation Model** combining the strengths of:

🔹 Content-Based Filtering

- Analyzes restaurant attributes like cuisine type, average cost, rating, and delivery option.
- Matches these with user-stated preferences to recommend relevant restaurants.

🔹 Collaborative Filtering (SVD)

- Uses historical user rating data to find similar users and suggest restaurants based on collective behavior.
- Implemented using the `Surprise` library's Singular Value Decomposition (SVD) algorithm.

🔹 Hybrid Approach

- Merges both filtering strategies to solve the cold-start and sparsity problems.
- Balances personalization with discovery of new or less popular options.

Dataset

The dataset is sourced from **Kaggle** and titled:

> Zomato Bangalore Restaurants Dataset by Himanshu Poddar

Conclusion

This project successfully demonstrates how **machine learning and recommendation systems** can simplify dining decisions by tailoring restaurant suggestions to user preferences and behavior. It offers a powerful and adaptive solution for users in both familiar and unfamiliar areas. The hybrid model ensures balance between personalized results and discovery of new options. This project lays the foundation for more advanced, real-time, and location-aware food recommendation engines.