A Project Report

On

Travel Vista: Tailored Travel Destinations Discovery

Submitted By

P. Sudhakar Reddy – sudhakarp2211@gmail.com
G. Shoyab Ali Khan –shoyabalikhan296@gmail.com
P. Nethravathi - nethravathi.palyam@gmail.com
R. Sandeep Gowda – snesasandeepr@gmail.com

MADANAPALLE INSTITUTE OF TECHNOLGY & SCIENCE

(UGC – AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu)

(Accredited by NBA, Approved by AICTE, New Delhi)

AN ISO 21001 Certified Institution

P. B. No: 14, Angallu, Madanapalle – 517325

2020-2024

ABSTRACT

The tourism recommendation system is an interactive web application designed to assist travelers in discovering and planning their trips more effectively. The system leverages data on cities and places of interest to provide personalized recommendations based on user preferences. Developed using Streamlit, a Python library for creating web applications, the system offers a user-friendly interface for exploring travel destinations.

The system allows users to choose between two main options: exploring cities or browsing places by category. When exploring cities, users can select a city of interest from a list and view recommended places to visit within that city. These recommendations include details such as the distance from the city center, ratings, and a brief description of each place.

Alternatively, users can browse places by category, selecting from a list of categories such as beaches, museums, parks, and more. After selecting a category, the system displays a list of recommended places from all cities that fall under the chosen category. This feature enables users to explore specific types of attractions across different destinations.

The recommendation system uses a content-based filtering approach, employing natural language processing techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) to analyze place descriptions and calculate similarity scores. These scores are then used to recommend places that are similar to those that the user has shown interest in.

Overall, the tourism recommendation system aims to enhance the travel planning experience by providing personalized and relevant recommendations, helping users discover new and exciting destinations to explore.

INDEX

CHAPTER – 1
INTRODUCTION

1.1 MOTIVATION:

The motivation behind this tourism recommendation system project is to address common challenges faced by travelers when planning trips. Traditional travel planning methods often involve sifting through large amounts of information from various sources, such as travel guides, websites, and recommendations from friends. This process can be time-consuming, overwhelming, and may not always result in finding the best-suited destinations or activities.

By developing a tourism recommendation system, this project aims to streamline the travel planning process and provide personalized recommendations to users based on their preferences. This includes factors such as preferred travel destinations, types of activities they enjoy, budget constraints, and travel dates. By leveraging machine learning algorithms and data analysis techniques, the system can analyze large datasets of tourist attractions, accommodation options, and user preferences to generate tailored recommendations.

The ultimate goal is to enhance the overall travel experience for users by helping them discover new and interesting destinations, find suitable accommodations and activities, and plan their trips more efficiently. Additionally, the project aims to demonstrate the practical applications of data science and machine learning in the travel and tourism industry, showcasing how these technologies can be used to create innovative solutions that benefit both travelers and travel service providers.

1.2 PROBLEM DEFINITION:

The problem addressed by this tourism recommendation system project is the inefficiency and complexity of the travel planning process. Traditional travel planning methods often involve manual research, which can be time-consuming and overwhelming, especially when considering multiple destinations, activities, and preferences.

The project aims to simplify this process by providing users with personalized recommendations for tourist destinations and activities. By analyzing user preferences, such as preferred travel destinations, types of activities, budget constraints, and travel dates, the system can suggest tailored recommendations that match the user's interests and requirements.

Additionally, the project aims to address the challenge of information overload by aggregating and analyzing data from various sources, including tourist attractions, accommodations, and user reviews. By leveraging machine learning algorithms and data analysis techniques, the system can provide users with relevant and useful recommendations, ultimately enhancing their travel planning experience.

1.3 OBJECTIVE OF THE PROJECT:

The objective of this tourism recommendation system project is to create a user-friendly platform that assists travelers in planning their trips more efficiently. The system aims to achieve the following objectives:

1. Personalized Recommendations: Provide users with personalized recommendations for tourist destinations, activities, and accommodations based on their preferences and constraints.

2. Simplified Travel Planning: Simplify the travel planning process by offering a streamlined interface for selecting destinations, activities, and travel dates.

3. Enhanced User Experience: Enhance the user experience by providing relevant information, such as ratings, reviews, and descriptions, for each recommended destination or activity.

4. Data-driven Decision Making: Utilize data analysis and machine learning techniques to analyze user preferences and historical data to generate accurate and useful recommendations.

5. Cost and Time Efficiency: Help users save time and effort by offering efficient recommendations that match their interests and budget constraints.

6. Increased Engagement: Increase user engagement by providing interactive features, such as maps, images, and social sharing options, to enhance the overall travel planning experience.

1.4 LIMITATIONS OF THE PROJECT:

The limitations of this tourism recommendation system project include:

1. Data Availability and Quality: The system's effectiveness heavily relies on the availability and quality of the data. Inaccurate or incomplete data may lead to less accurate recommendations.

2. Limited Scope: The system focuses primarily on recommending tourist destinations and activities based on user preferences. It does not account for other factors that may influence travel decisions, such as weather, local events, or travel restrictions.

3. Static Recommendations: The recommendations provided by the system are based on historical data and user preferences at a specific point in time. They may not reflect real-time changes or evolving user preferences.

4. User Dependency: The system requires user input to generate recommendations, such as selecting a city or category. It may not be suitable for users who prefer fully automated recommendation systems.

5. Algorithm Bias: The recommendation algorithms used in the system may exhibit bias based on the data they are trained on, potentially leading to skewed recommendations.

6. Lack of Customization: While the system aims to provide personalized recommendations, its level of customization may be limited compared to more sophisticated travel planning platforms.

7. Limited Geographic Coverage: The system's recommendations are based on the available dataset, which may have limited coverage of certain regions or destinations.

8. Resource Intensive: Implementing and maintaining the system, especially the machine learning components, may require significant computational resources and expertise.

These limitations highlight the need for ongoing refinement and improvement of the system to address these challenges and enhance its effectiveness and usability.

1.5 ORGANIZATION OF DOCUMENT:

1.5.1 Feasibility Study:

In this phase, the project's viability is assessed, and a very basic project design along with some cost estimates are presented in the business proposal. The proposed system's viability must be investigated during system analysis. This is to make sure the business won't be burdened by the suggested method. A basic understanding of the system's primary requirements is necessary for feasibility study.

The following four factors are crucial to the feasibility analysis:

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY
- OPERATIONAL FEASIBILITY

Economic Feasibility:

The purpose of this study is to evaluate the system's potential financial impact on the company. The corporation has a finite amount of money to dedicate to the system's research and development. The cost is justifiable. The constructed system is therefore well under the allocated budget, which was made possible by the fact that most of the technologies were publicly available. All the personalised items had been bought.

Technical Feasibility:

The purpose of this study is to evaluate the system's technical needs, or its technical feasibility. There is hardly a significant demand on the technical resources available to construct a system. The client will not encounter unjustified demands as a result of this. The designed system has a low requirement because its implementation just calls for minimal or no changes.

Social Feasibility:

The purpose of the study is to determine the level of system acceptability by the user. This involves teaching the user how to operate the technology effectively. The system must be accepted by the

user as a requirement rather than as a risk. The techniques used to educate and familiarise the user with the system will determine the extent of acceptance by the users. Since he is the system's last user, his confidence must be increased for him to offer some helpful assess, which is appreciated.

Operational Feasibility:

The software utilised in this project must be user-friendly, dependable, secure, portable, available, and maintainable to be considered operationally feasible.

CHAPTER – 2

ANALYSIS

2.1 INTRODUCTION:

The tourism recommendation system project aims to provide personalized recommendations for tourist destinations and activities. The system leverages machine learning algorithms to analyze user preferences and historical travel data to suggest places to visit based on user input.

The system's main features include:

- Recommendations based on selected cities or categories.
- Detailed information about recommended places, including distance, ratings, and descriptions.
- User-friendly interface with buttons for easy navigation.

The project's goal is to assist travellers in planning their trips more efficiently and discovering new and exciting destinations. By leveraging technology and data analytics, the system aims to enhance the overall travel experience and promote tourism in various regions.

2.2 REQUIREMENT SPECIFICATIONS:

Functional Requirements:

- User Selection: Users can select between exploring cities or specific categories of places.
- City Recommendations: When a city is selected, the system recommends places to visit within that city.
- Category Recommendations: When a category is selected, the system recommends places from all cities in that category.
- Place Details: The system displays details of recommended places, including distance, ratings, and descriptions.
- User Interaction: Users can interact with the system using buttons for navigation and selection.

Non-Functional Requirements:

- Performance: The system should respond quickly to user interactions and provide recommendations in a timely manner.
- User Interface: The user interface should be intuitive, visually appealing, and easy to use.
- Accuracy: The recommendations should be accurate and relevant to the user's selection.
- Security: The system should ensure the security and privacy of user data.
- Scalability: The system should be able to handle a large number of users and data sets.
- Reliability: The system should be reliable and available whenever users need it.
- Compatibility: The system should be compatible with different devices and web browsers.

- ✓ **Hardware and software requirements:** This includes the hardware and software components needed to run the software, such as the operating system, programming languages, and libraries required for the implementation of the Hybrid model.

    - **Hardware Requirements:**
        - System : 12th Gen Intel(R) Core(TM) i5-1235U 1.30 GHz
        - Hard Disk : 512 GB SSD
        - Monitor: 15'' LED
        - Input Devices: Keyboard, Touchpad
        - RAM: 16 GB
    - **Software Requirements:**
        - Operating System: Windows 11 Home
        - Coding Language: Python
        - Software used: Anaconda Navigator
- ✓ **Data requirements:**
    - **Dataset Link:** https://www.kaggle.com/datasets/naqibahmedkadri/famous-indian-tourist-places

2.3 CONTENT DIAGRAM OF PROJECT:

The tourism recommendation system is designed to provide personalized travel recommendations based on user preferences. The system utilizes two main datasets: city data and places data. The city data contains information about various cities, including their names, ideal durations, and best times to visit. The places data consists of details about different places within these cities, such as the distance from the city center, ratings, and descriptions.

The system employs natural language processing techniques, specifically TF-IDF (Term Frequency-Inverse Document Frequency), to extract features from the place descriptions. These features are used to calculate a similarity matrix using cosine similarity, which is then used to recommend similar places based on user inputs.

The user interface allows users to choose between exploring cities or specific types of places. If a city is selected, the system recommends places within that city based on similarity scores. If a specific category of places is chosen, the system recommends places from all cities that fall under that category.

The system aims to enhance the travel planning experience by providing users with tailored recommendations that match their interests and preferences. It is designed to be user-friendly and intuitive, allowing users to easily explore and discover new destinations.

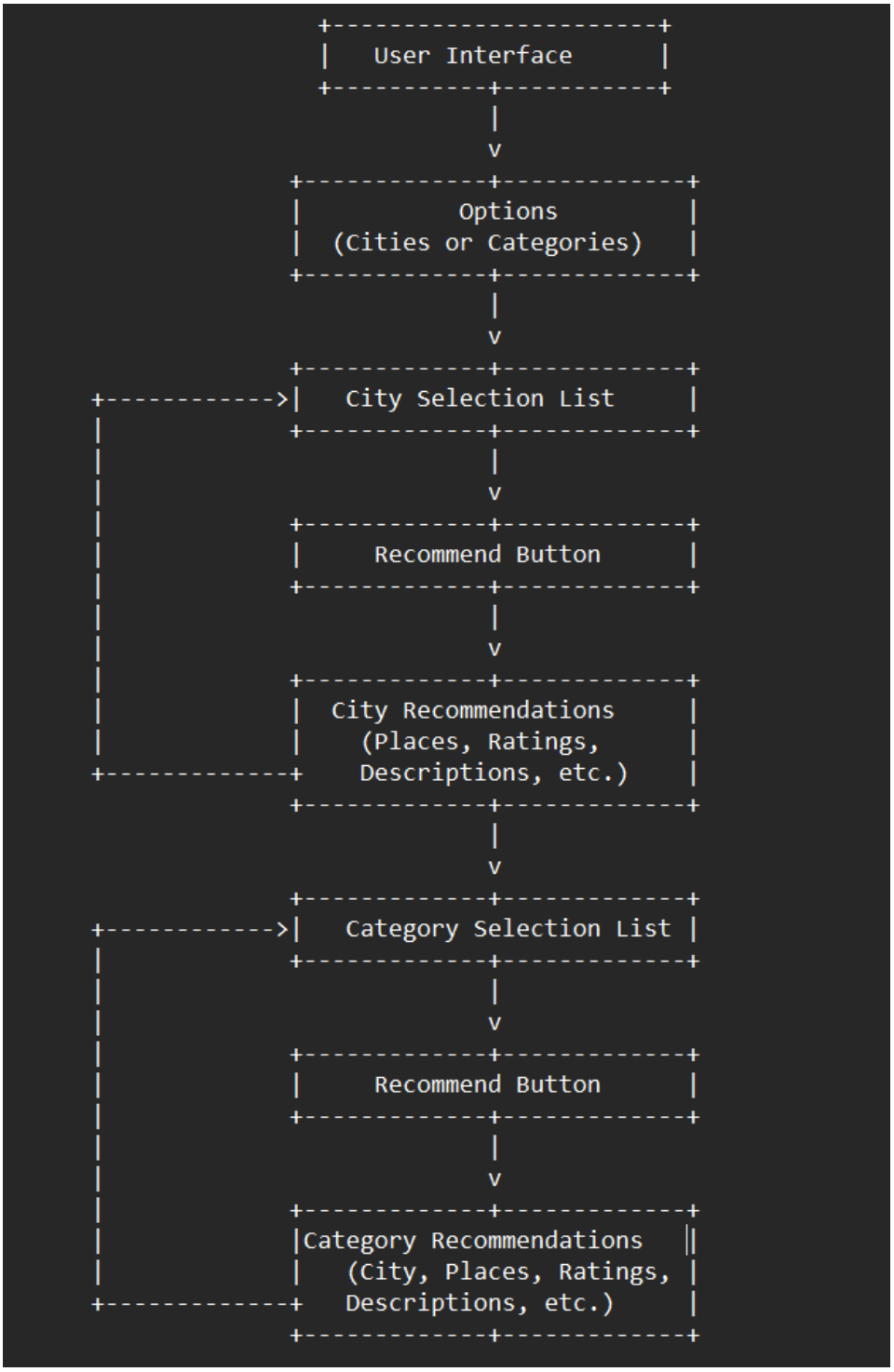

Fig 2.3.1 Content Diagram

CHAPTER – 3

DESIGN

3.1 INTRODUCTION:

The tourism recommendation system is designed to assist users in finding suitable travel destinations based on their preferences. The system's interface, built using Streamlit, offers users two primary options: selecting cities or choosing categories for recommendations. Upon selecting a city, users can view a list of places to visit within that city, along with details such as distance from the city center, ratings, and descriptions. Conversely, selecting a category displays places from all cities that fall under the chosen category. The system utilizes a recommendation engine to provide these suggestions, leveraging datasets containing city and place information. The interface is user-friendly, featuring dropdown menus for selections and a "Recommend" button to trigger the recommendation process. Additionally, the system supports the display of a background image for aesthetic purposes. Overall, the system aims to deliver personalized and relevant travel recommendations to enhance the user's travel planning experience.

3.2 UML DIAGRAM:

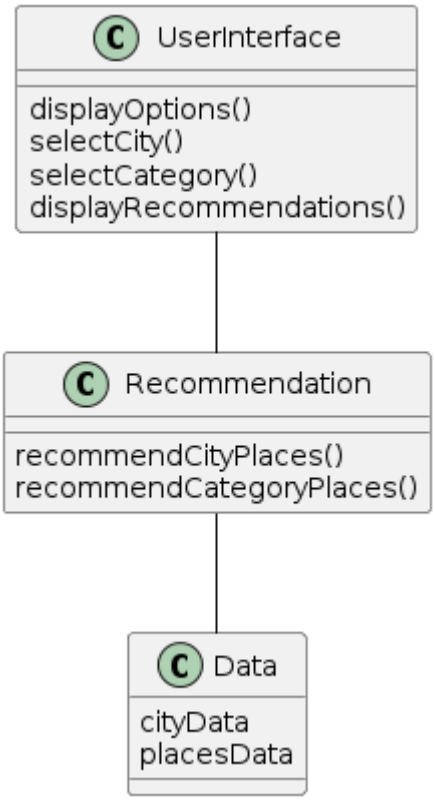

Fig 3.2.1 UML Diagram

3.3 SYSTEM ARCHITECTURE:

The tourism recommendation system's system architecture is designed to provide personalized recommendations to users based on their preferences. The system consists of several key components, including the user interface, recommendation engine, data storage, data preprocessing, and integration modules.

The user interface (UI) serves as the front-end of the system, allowing users to interact with the system. It presents options for users to choose between cities and categories of places, and it displays the recommendations based on the user's selections.

The recommendation engine is the core of the system, responsible for processing user inputs and generating recommendations. It utilizes algorithms such as TF-IDF for text analysis and cosine similarity for recommendation scoring. The engine accesses relevant data from the data storage component.

The data storage component stores the city and place data used by the recommendation engine. It provides efficient data access for the recommendation engine to generate recommendations.

Data preprocessing is performed on the data before it is used by the recommendation engine. This includes merging datasets, text vectorization, and computing similarity scores to prepare the data for recommendation generation.

Integration is crucial for the system to function seamlessly. It ensures that all components work together effectively. The UI sends user selections to the recommendation engine, which processes the data and sends back recommendations to be displayed on the UI.

External dependencies, such as map services for location information, may also be integrated into the system to enhance its functionality.

Overall, the system architecture is designed to provide users with a seamless and personalized experience, delivering relevant recommendations for places to visit based on their preferences.

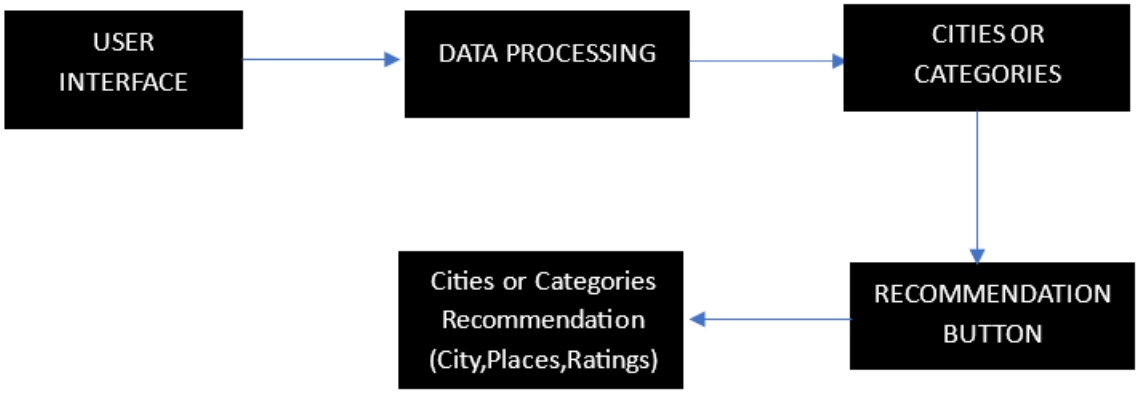

Fig 3.3.1 Architecture Diagram

3.4 MODULE DESIGNING AND ORGANIZATION:

The system architecture of the tourism recommendation system is designed to provide users with personalized recommendations for places to visit based on their preferences. The system consists of several modules that work together to achieve this goal.

The User Interface module is responsible for interacting with users, presenting them with options to choose between cities and categories of places, and displaying the recommendations based on their selections. It provides an intuitive and user-friendly interface for users to interact with the system.

The Recommendation Engine module is the core of the system, employing algorithms like TF-IDF and cosine similarity to process user inputs and generate recommendations. It accesses data from the Data Storage module, which stores information about cities, places, and their descriptions. The Data Preprocessing module prepares this data for the recommendation engine, merging datasets, and computing similarity scores.

The Integration module ensures seamless communication between the user interface, recommendation engine, and data storage modules. It also integrates external services, such as map services, to enhance the system's functionality. The External Dependencies module manages these integrations and processes data received from external services.

The Logging and Monitoring module tracks system events and performance, providing administrators with insights into the system's health. The Security module ensures that the system is secure, implementing features like authentication, authorization, and data encryption to protect user data and the system from security threats.

Overall, the system architecture is designed to provide users with a seamless and personalized experience, helping them discover new places to visit and enhancing their overall tourism experience.

CHAPTER – 4
IMPLEMENTATION AND RESULTS

4.1 INTRODUCTION:

The implementation of the tourism recommendation system involves several key components and technologies to create a functional and user-friendly application. The system is implemented using Python and the Streamlit framework for the web interface, along with pandas for data manipulation and scikit-learn for text processing and similarity calculations.

The system's backend consists of several modules, including a Data Storage module to manage and store information about cities, places, and their descriptions. The Data Preprocessing module processes this data to prepare it for the recommendation engine, which uses TF-IDF and cosine similarity to generate recommendations based on user inputs.

The User Interface module provides an interactive web interface for users to select cities or categories of places and view recommendations. The interface is designed to be intuitive and user-friendly, guiding users through the selection process and displaying recommendations in a clear and concise manner.

Overall, the implementation of the tourism recommendation system aims to provide users with a seamless and personalized experience, helping them discover new and exciting places to visit based on their preferences and interests.

4.2 METHOD OF IMPLEMENTATION:

The implementation of the tourism recommendation system is designed to provide a user-friendly experience for travelers seeking guidance on places to visit. The system is built using Python and various libraries such as Pandas for data manipulation, Scikit-learn for text processing and similarity calculations, and Streamlit for creating the web interface.

The data is loaded from CSV files containing information about cities, places, and their attributes. Text data is processed using TF-IDF vectorization to understand the importance of words in describing a place. Cosine similarity scores are then calculated to determine the similarity between different places based on their descriptions.

The web interface is designed to be intuitive and interactive. Users can choose between exploring cities or specific categories of places. When a city is selected, a list of places within that city is displayed along with their distance from the city center, ratings, and descriptions. If a category is chosen instead, the system displays places from all cities in that category, providing users with a variety of options to explore.

The system aims to assist travelers in discovering new and interesting places to visit, enhancing their overall travel experience. It can be deployed easily, making it accessible to a wide range of users looking for personalized recommendations for their travel destinations.

CODE:

```
#import libraries
import streamlit as st
import pandas as pd

# Load the datasets
city_data = pd.read_csv('C:/Users/sudha/OneDrive/Documents/Coapps/City.csv')
places_data = pd.read_csv('C:/Users/sudha/OneDrive/Documents/Coapps/Places.csv')
# Merge city and places data
data = pd.merge(city_data, places_data, on='City')

# Streamlit UI
def main():
    st.title('Tourism Recommendation System')
    option = st.radio('Choose an option:', ('Cities', 'Places'))

    if option == 'Cities':
        st.subheader('Select a city:')
        city_name = st.selectbox('', city_data['City'])
        if st.button('Recommend'):
            city_places = data[data['City'] == city_name]
            st.subheader(f'Places to visit in {city_name}:')
            for idx, place in city_places.iterrows():
                st.write('- Place:', place['Place'])
                st.write('  Distance from city center:', place['Distance'])
                st.write('  Ratings:', place['Rating'])
                st.write('  Place description:', place['Place_desc'])

    elif option == 'Places':
        st.subheader('Select a category:')
        categories = data['Category'].unique()
        selected_category = st.selectbox('', categories)
        category_places = data[data['Category'] == selected_category]
        if st.button('Recommend'):
```

```
        st.subheader(f'Places in category "{selected_category}":')
        for idx, place in category_places.iterrows():
            st.write('- City:', place['City'])
            st.write('  Place:', place['Place'])
            st.write('  Distance from city center:', place['Distance'])
            st.write('  Ratings:', place['Rating'])
            st.write('  Place description:', place['Place_desc'])

if __name__ == "__main__":
    main()
```

 # run through cmd prompt

streamlit run app.py

4.3 OUTPUT SCREEN AND RESULT ANALYSIS:

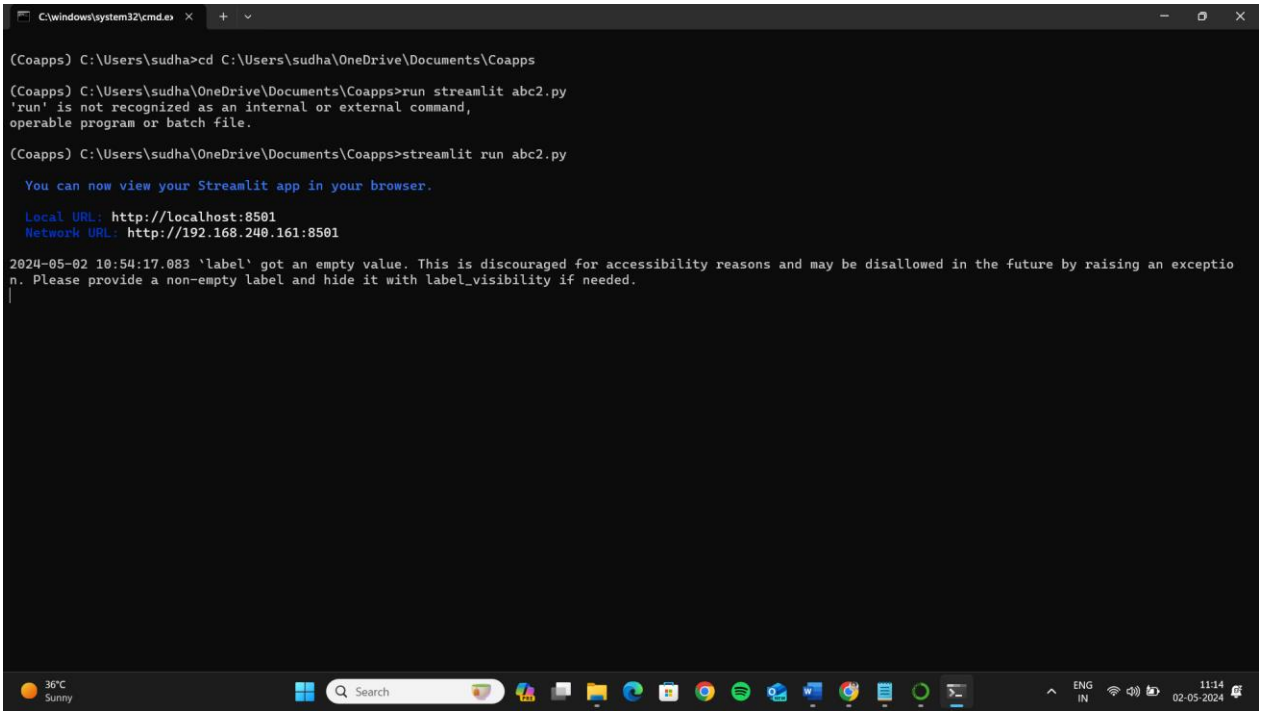

Fig 4.3.1 Run the file using cmd prompt

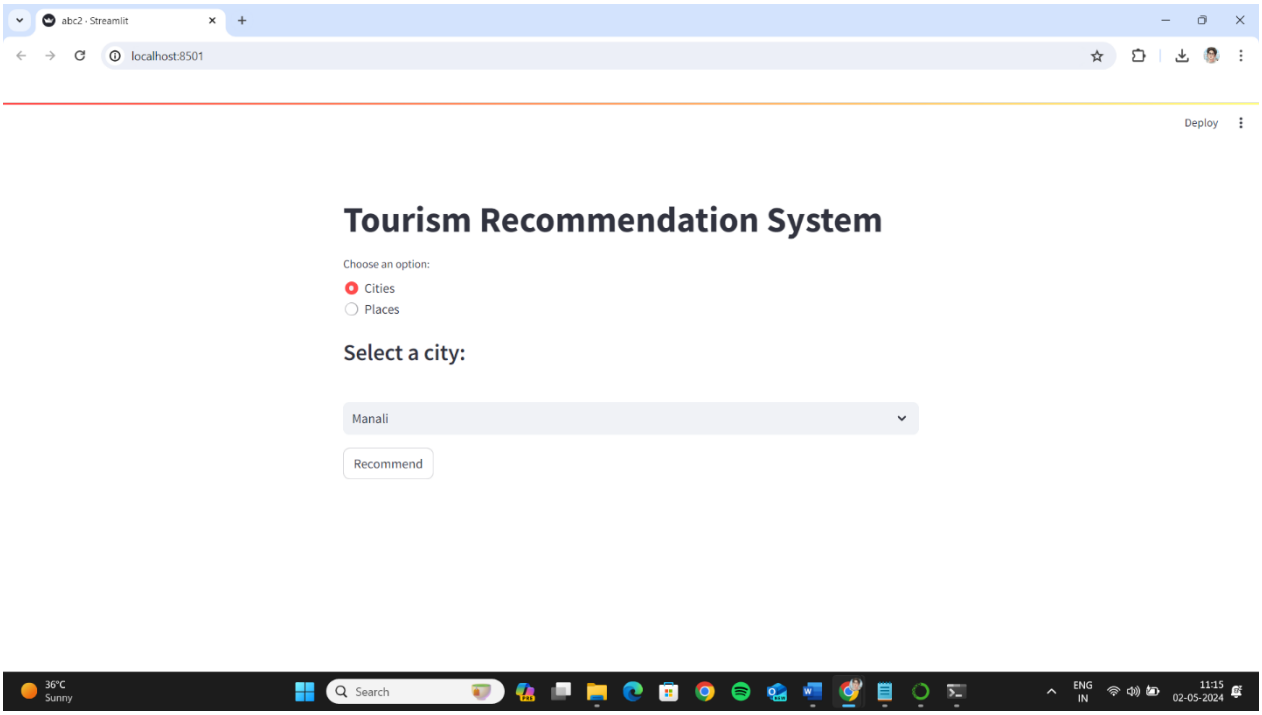

Fig 4.3.2 Home page of Web Interface

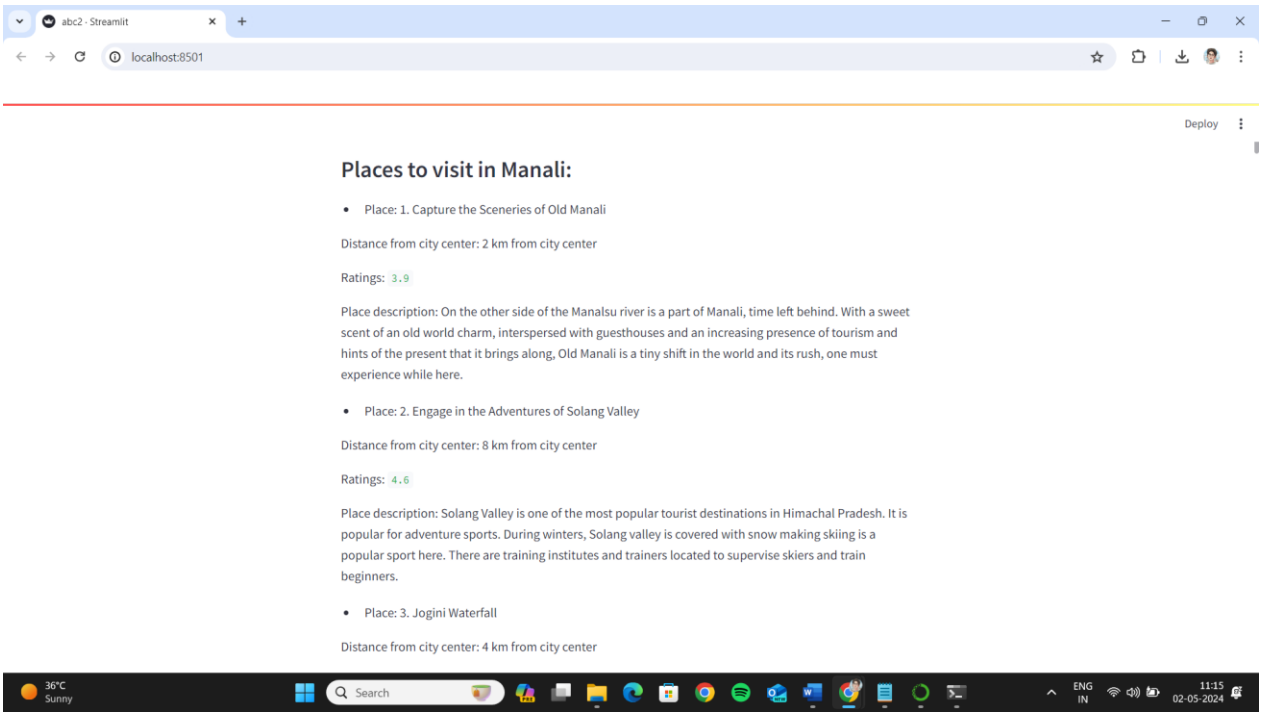

Fig 4.3.3 Recommendation of places according to cities.

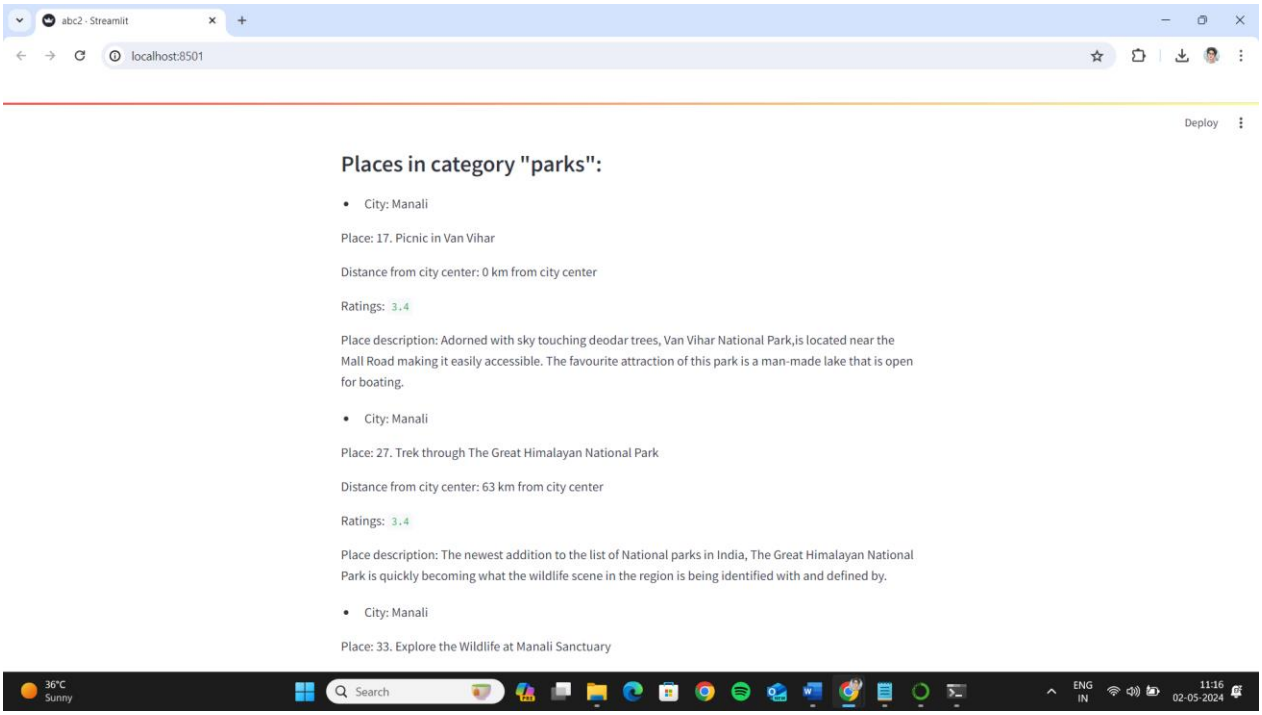

Fig 4.3.4 Recommendation of places according to categories.

CHAPTER – 5

CONCLUSION

5.1 CONCLUSION:

The tourism recommendation system developed in this project provides a valuable tool for travelers seeking personalized recommendations for places to visit. By leveraging natural language processing and machine learning techniques, the system is able to analyze and recommend places based on user preferences and descriptions.

The system's user-friendly interface makes it easy for travelers to explore cities and discover new places of interest. By providing detailed information such as distance from city center, ratings, and descriptions, the system helps users make informed decisions about their travel itinerary.

While the system has been designed to be robust and efficient, there are some limitations to be aware of. For instance, the recommendations are based on textual descriptions and ratings, which may not always accurately reflect a place's true appeal. Additionally, the system relies on the availability and quality of data, which can vary depending on the source.

Overall, the tourism recommendation system represents a valuable tool for travelers looking to enhance their travel experience. With further refinement and improvement, the system has the potential to become an indispensable companion for travelers around the world.

7.2 FUTURE ENHANCEMENTS:

Future enhancements for this project could focus on several key areas to further improve the user experience and the system's effectiveness. One potential enhancement could involve incorporating user feedback and interaction data to continuously refine and personalize the recommendations. By analyzing user behavior and preferences, the system could adapt its recommendations over time to better match individual user tastes and preferences.

Another enhancement could involve expanding the dataset to include more cities, places, and categories, thereby increasing the variety and specificity of recommendations. This could involve integrating additional data sources, such as user-generated content from social media platforms or travel review websites, to enhance the diversity and quality of recommendations.

Furthermore, incorporating more advanced machine learning techniques, such as deep learning or reinforcement learning, could enhance the system's ability to understand and predict user preferences. These techniques could help the system learn more complex patterns and relationships in the data, leading to more accurate and personalized recommendations.

Finally, enhancing the system's user interface and interaction design could further improve the user experience. This could involve creating a more visually appealing and intuitive interface, as well as incorporating features such as interactive maps, travel itineraries, and social sharing options to make trip planning more engaging and enjoyable.