

**Muniba Manaal****FA22-BAI-032****Final Project**




# Travel Recommendation System

It is a travel recommender system that interacts with users to gather their travel preferences including climate, budget, activity, and cultural scene. It uses these preferences to suggest suitable destinations from a diverse knowledgebase. The system dynamically collects user responses, validates them, and if the response is 'exit', it aborts the program. For recommendations, it matches user preferences with destination attributes and provides a sorted list of unique destinations that meet the criteria. The program includes a robust database of fictional and real-world destinations with various characteristics, ensuring a wide range of recommendations for different user preferences.

## Prolog code:

```
1 :- dynamic user_preference/2.
2 :- discontinuous matches_preference/2.
3 :- use_module(library(random)).
4
5 start :-
6     retractall(user_preference(_, _)),
7     greet_user,
8     ask_preferences,
9     recommend,
10    goodbye.
11
12 greet_user :-
13     write('Welcome to Travel Recommender!'), nl,
14     write('Please enter your preferences for a personalized travel experie
15
16 goodbye :-
17     write('Thank you for using Travel Recommender! Have a great trip!'), n
18
19
20 ask_preferences :-
21     ask_climate,
22     ask_budget,
23     ask_activity,
24     ask_culture.
25
26
27 ask_climate :-
28     ask_preference('climate', [moderate, tropical, warm, cold, any]).
```

## Queries:


?- start.



Welcome to Travel Recommender!

Please enter your preferences for a personalized travel experience.

What is your preferred climate? ([moderate,tropical,warm,cold,any]/exit)

What is your preferred budget? ([high,moderate,low,any]/exit)

What is your preferred activity? ([beaches,history,adventure,culture,none,any]/exit)

What is your preferred culture? ([vibrant,quiet,indifferent,any]/exit)

Recommended Destinations:

- celestialcity
- frostfire
- harmonyHaven
- mystica
- serenityshore
- solaris
- tranquilTropics
- valoria
- whisperingwoods
- zenithZephyr

Thank you for using Travel Recommender! Have a great trip!

true

1

## Travel Recommender System - Detailed Analysis

- Overview:** The program is a Prolog-based interactive travel recommender system. It engages the user in a dialogue to collect travel preferences and generates suitable destination recommendations.
- User Interaction:**
  - Initiation:** Begins with a welcoming message and an invitation to input preferences.
  - Preferences Gathering:** Collects user preferences for climate, budget, activities, and cultural scene through a series of prompts.
  - Input Validation:** Validates user responses, offering a chance to re-enter in case of invalid inputs or exit the program.
- Recommendation Engine:**
  - Matching Logic:** Utilizes user preferences to filter a pre-defined list of destinations. It accommodates broad preferences by including 'any' as an option.
  - Unique Recommendations:** Ensures uniqueness in the recommendations to prevent repetition.
  - Dynamic Knowledgebase:** Features a versatile database of both fictional and real-world destinations, each characterized by attributes corresponding to the preference categories.
- Program Structure:**
  - Dynamic Preference Management:** Uses dynamic predicates to store and manage user preferences.
  - Modular Design:** The program is structured into distinct sections for preference gathering, recommendation processing, and user interaction, enhancing readability and maintainability.
- User Experience:**
  - Interactive and User-Friendly:** Offers a responsive and engaging user experience.

- **Customized Recommendations:** Tailors destination suggestions based on individual preferences, enhancing the relevance of recommendations.

## 6. Conclusion:

- The program effectively marries the logical capabilities of Prolog with an interactive user interface, providing a practical and enjoyable tool for travel planning based on personal preferences.

This program showcases the application of Prolog in creating interactive and personalized user experiences, combining logical reasoning with user-centered design.

# Functionality:

## 1. Initialization and Greeting ( `start` , `greet_user` , `goodbye` ):

- The `start` predicate initializes the program, clearing any existing preferences.
- `greet_user` displays a welcome message.
- `goodbye` shows a farewell message at the end.

## 2. Collecting User Preferences ( `ask_preferences` , `ask_preference` , `handle_input` ):

- `ask_preferences` sequentially calls functions to collect user preferences for climate, budget, activity, and culture.
- `ask_preference` is a generic function used to ask for each type of preference, displaying the options and reading user input.
- `handle_input` processes the user's input, handling 'exit' commands and validating choices.

## 3. Generating Recommendations ( `recommend` , `find_destinations` , `matches_preference` , `matches_culture_preference` ):

- `recommend` uses collected preferences to generate destination recommendations.
- `find_destinations` filters the destinations from the knowledge base according to the user's preferences.
- `matches_preference` and `matches_culture_preference` are helper predicates used to check if a destination matches the given preferences.

## 4. Displaying Recommendations ( `display_recommendations` , `print_list` ):

- `display_recommendations` handles the output of recommended destinations.
- `print_list` is a utility for printing the list of destinations.

## 5. Knowledge Base (Destinations):

- The program includes a knowledge base of destinations (like `destination(elysium, moderate, high, history, yes)` ), each with attributes for climate, budget, activity type, and cultural vibrancy.

This structure allows your program to interactively collect user preferences and suggest suitable travel destinations based on those preferences.

# Conclusion:

The Prolog program is an interactive travel recommender system. It starts by clearing previous preferences ( `retractall` ) and greeting the user. It then sequentially asks the user for their preferences in climate, budget, activities, and culture. Each preference category is handled by `ask_preference` , which validates user input or allows exit. The `recommend` function collects these preferences and uses `find_destinations` to filter destinations from a predefined list, ensuring unique recommendations. The recommendation display includes a check for empty results and formats the output list. The program ends with a goodbye message. The knowledge base contains fictional and real-world destinations, each characterized by attributes like climate, budget, activity type, and cultural vibrancy. The program's structure supports dynamic preference management and a user-friendly interface, catering to diverse user needs and enhancing the travel planning experience.

In conclusion, the Prolog program is an expert system designed for personalized travel recommendations. It effectively interacts with users to gather their preferences regarding climate, budget, activities, and culture. The program then intelligently filters and suggests destinations from its knowledge base, matching these

preferences. The use of dynamic predicates and utility functions for handling user inputs and displaying recommendations ensures a user-friendly and responsive experience. This system demonstrates the power of Prolog in developing rule-based reasoning applications, particularly in the domain of personalized decision-making and recommendations.