
CAPSTONE PROJECT

SmartTravelPlanner

Presented By:

1. Sahithi Gogula-Vasavi College of Engineering-Information Technology

OUTLINE

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

Travel Planner Agent-PackMate

The Challenge - A Travel Planner Agent is an AI-powered assistant that helps users plan trips efficiently and intelligently.

It uses real-time data to suggest destinations, build itineraries, and recommend transport and accommodation options.

By understanding user preferences, budgets, and constraints, it tailors personalized travel plans.

Integrated with maps, weather updates, and local guides, it ensures a smooth travel experience.

The agent can also manage bookings, alert users to changes, and optimize schedules on the go.

This smart assistant transforms complex travel planning into a seamless, enjoyable process.

PROPOSED SOLUTION

- The proposed system aims to simplify and personalize the travel planning process by leveraging AI and automation. It helps users generate travel itineraries, packing lists, and suggestions based on their preferences, while also ensuring real-time responsiveness. The solution includes the following components:
- **Data Collection:**
 - Collect user inputs such as destination, travel dates, budget, preferences (e.g., adventure, leisure), and group size.
 - Integrate external data sources like weather, local events, and popular attractions.
- **Data Preprocessing:**
 - Clean and preprocess data to handle missing values, outliers, and inconsistencies.
 - Derive meaningful features such as trip duration, travel type, and destination characteristics.
- **Machine Learning Algorithm:**
 - Use ML models to recommend destinations, activities, and personalized packing lists based on user preferences, weather, and trip duration.
 - Implement a content-based or hybrid recommendation engine to suggest places, accommodations, and experiences tailored to the user's profile.
- **Deployment:**
 - Develop a user-friendly web or mobile interface for real-time visualization of predicted bike demand.
 - Deploy the solution on a scalable cloud platform ensuring fast response and high availability.
- **Evaluation:**
 - Evaluate the user experience and satisfaction through feedback forms.
 - Continuously improve AI suggestions and packing recommendations based on user interactions and preferences.
- **Result:** An intelligent, user-friendly Travel Planner Agent that saves time, enhances trip planning accuracy, and offers a personalized travel experience from start to finish.

SYSTEM APPROACH

■ System Requirements

- **Hardware:**
 - 64-bit processor
 - Minimum 8 GB RAM
 - Internet connectivity for real-time data access
- **Software:**
 - Python (3.x)
 - Jupyter Notebook or any IDE (e.g., VS Code, PyCharm)
 - Web browser (for interface viewing if deployed)

■ Libraries Required

- pandas – Data manipulation and analysis
- NumPy – Numerical operations
- matplotlib / seaborn – Data visualization
- scikit-learn – Data preprocessing and evaluation
- statsmodels – Time-series modeling (ARIMA, SARIMA)
- TensorFlow or PyTorch – Deep learning models (LSTM)
- joblib or pickle – Model serialization

ALGORITHM & DEPLOYMENT

■ Algorithm Selection

A **rule-based and ML-enhanced recommendation system** is used to personalize travel itineraries and packing lists.

- Rule-based logic filters recommendations based on fixed conditions (e.g., climate, trip type).
- ML models (optional) such as clustering or decision trees enhance personalization based on past user data.

■ Data Input

The system takes the following as inputs:

- User preferences (budget, trip type, preferred activities)
- Destination details (location, climate, time zone)
- Trip duration and travel dates
- External data (weather, local events, travel advisories)

■ Training Process

- If ML is used, the model is trained on historical travel records or user behavior patterns.
- Techniques like **k-means clustering** (for traveler profiles) or **collaborative filtering** (for activity suggestions) can be used.
- Minimal or no training is required for rule-based filtering.

■ Prediction Process

- Based on user input and available data, the system generates:
- A **custom itinerary** with daily plans
- A **smart packing list** tailored to destination, activities, and weather

RESULT

Build

AI Model: mistral-large

LangGraph ReAct

Instructions [Advanced configuration](#)

You are a smart, real-time AI-powered Travel Planner Agent named WanderFree AI.
You help users plan personalized trips based on their:

- Budget
- Dates

Knowledge [Clear chat](#)

Vector index

MotherLand.pdf

If you want to add a vector index to the agent, clear the agent preview chat.

Tools

Agent preview

SG You: 07:51 PM

I'm going to Manali for 5 days in December. What should I pack and where can I stay on a ₹2000 budget?

watsonx Agent 07:51 PM

Here's a packing list for your trip to Manali in December:

Clothing

- Warm sweaters or jackets
- Thermal undergarments
- Long-sleeved shirts or turtlenecks
- Heavy woolen or fleece pants
- Warm socks
- Warm woolen sweaters
- Thermal underwear

Medications

- Electronics
- Extra pair of gloves for daytime use

Best Places to Stay in Manali on a ₹2000 Budget

1. **Hotel Yak Mall Road** - ₹2,000–₹2,500
2. **Budget Hotels near Solang Valley** - Starting at ₹102
3. **Budget Hotels in Soil** - Starting at ₹794
4. **Best Manali under 2000 Hotels** - Various options available
5. **Budget Hotels in Manali** - Starting at ₹134

These options should help you find affordable accommodation within your budget.
Enjoy your trip to Manali!

► **How did I get this answer?**

Type something...

CONCLUSION

■ The Travel Planner Agent project effectively streamlines the travel planning process by generating personalized itineraries and smart packing lists based on user preferences, destination data, and real-time factors like weather. The solution has proven to be efficient and user-friendly, enhancing the overall travel experience through automation and intelligent recommendations.

During implementation, challenges such as handling incomplete user inputs and integrating external data were encountered but addressed through iterative development. The system performs well in delivering relevant suggestions, though future improvements such as group planning, live cost estimates, and dynamic booking integration can further enhance its usefulness.

FUTURE SCOPE

- ❑ **Group Travel Planning:**
Enable collaborative planning for families or friends with shared itineraries and split preferences.
- ❑ **Live Booking Integration:**
Integrate APIs for flights, hotels, and activities to allow direct bookings from the app.
- ❑ **Budget Estimator:**
Add dynamic cost estimation based on selected destinations, transport, and accommodation.
- ❑ **Voice and Chatbot Interface:**
Implement voice assistance or conversational AI to make planning more interactive and accessible.
- ❑ **Multilingual Support:**
Expand accessibility for global users by supporting multiple languages.

REFERENCES

- [IBM Cloud Documentation](#)
- [IBM Watson Assistant](#)
- [Python Official Documentation](#)
- [Pandas Documentation](#)
- [NumPy Documentation](#)
- [OpenWeatherMap API](#)
- [Scikit-learn](#)
- [TensorFlow](#)
- [Google Maps Platform](#)

IBM CERTIFICATIONS



IBM CERTIFICATIONS

In recognition of the commitment to achieve
professional excellence



SAHITHI GOGULA

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution



Issued on: Jul 23, 2025
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/a3131aab-7409-4d3a-b9ef-e9de9e08bb1f>



IBM CERTIFICATIONS

IBM **SkillsBuild**

Completion Certificate



This certificate is presented to

SAHITHI GOGULA

for the completion of

**Lab: Retrieval Augmented Generation with
LangChain**

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 24 Jul 2025 (GMT)

Learning hours: 20 mins



THANK YOU