iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: V Month of publication: May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81680

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

TRIP-Sense: A Data-Driven Travel Planning and Recommendation System

Dr. K. Chaitanya[1], M. Manaswi Priya[2], G. Praneetha3, B. Pavan Kumar4, N. Hari Krishna5
[1]M. Tech, P.H.D, Department of Artificial Intelligence and Machine Learning, University College of Engineering and Technology, Acharya Nagarjuna University, Guntur, AP, India
[2, 3, 4, 5]Department of Artificial Intelligence and Machine Learning, University College of Engineering and Technology, Acharya Nagarjuna University, Nagarjuna Nagar, Guntur, AP, India

Abstract: Planning a trip in India is an intricate and time-consuming process involving the simultaneous evaluation of multiple factors such as destination, budget, trip duration, personal interests, and seasonal conditions. Existing travel platforms provide keyword-based filtering but fail to comprehend natural language or generate personalised, contextually aware itineraries. This paper presents TRIP-Sense, a hybrid travel recommendation system that integrates content-based filtering, collaborative filtering, and a lightweight natural language parser to deliver personalised destination suggestions, day-by-day AI-generated itineraries, budget analysis, and real-time interactive map visualisation. A multi-factor scoring engine is applied across 16 Indian destinations. The system optionally integrates the Claude API (Anthropic) for advanced generative itinerary responses. Evaluation demonstrates that the hybrid model outperforms individual filtering approaches, achieving over 90% NLP parsing accuracy and 84% overall recommendation accuracy. The complete system is delivered as a single-page web application using HTML, CSS, JavaScript, Leaflet.js, and the Overpass API with no server-side infrastructure required.
Keywords: Travel Recommendation, Hybrid Filtering, Content-Based Filtering, Collaborative Filtering, Natural Language Processing, Destination Scoring, Itinerary Generation, Budget Analysis, Leaflet.js, Web Application.

I. INTRODUCTION

With the rapid growth of the travel and tourism industry, the past few years have seen an explosion in the volume of information available to users planning trips. Travelling requires consideration of multiple aspects — destination, expenditure, duration, personal interests, and seasonal conditions — making the planning process complex and time-consuming.

Traditional travel systems provide suggestions based on popularity or generic ratings without taking the specific preferences of individual users into account. This absence of personalisation means travellers often struggle to find destinations that genuinely match their requirements.

To address this challenge, TRIP-Sense is designed as an advanced travel recommendation engine that uses machine learning to offer personalised suggestions. By accepting user inputs of budget, duration, and interests, the system generates optimised recommendations using a hybrid approach combining content-based and collaborative filtering. Results are presented through an interactive web dashboard showing destination cards, budget breakdowns, AI-generated itineraries, and map visualisations.

The key contributions of this paper are:

1) A hybrid recommendation engine combining content-based and collaborative filtering with learned weight parameters α and β.
2) A natural language query parser that extracts destination, duration, budget, and interests from free-form text without any external NLP library.
3) A city-specific AI itinerary knowledge base covering 16 Indian destinations with month-wise weather advisories.
4) Real-time interactive map integration using Leaflet.js and the Overpass API.
5) A fully functional single-page web application requiring no server-side infrastructure.

II. LITERATURE REVIEW

Recommendation systems have become an essential component of practical applications ranging from e-commerce and entertainment to tourism. Aggarwal [1] provides a comprehensive treatment of recommender system techniques including content-based filtering, collaborative filtering, and hybrid approaches.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Content-based filtering recommends items based on similarity between a user's preference profile and item attributes, typically measured using cosine similarity [2]. While this approach provides personalised recommendations, it suffers from low diversity and a tendency to repeatedly suggest similar items.

Collaborative filtering uses historical user-behaviour and interaction data to identify preference patterns. Algorithms such as K-Nearest Neighbours (KNN) and Singular Value Decomposition (SVD) are widely used [7]. However, collaborative filtering faces the cold-start problem and data sparsity [3].

Burke [3] demonstrated that hybrid systems combining content-based and collaborative filtering outperform individual methods by leveraging the strengths of both. Adomavicius and Tuzhilin [4] further identified key limitations of single-method recommender systems and proposed a taxonomy of hybrid approaches for the next generation of such systems.

Gavalas et al. [5] reviewed mobile recommender systems in tourism and found that location-aware, context-sensitive systems significantly improve user satisfaction over static recommendation engines. Aggarwal and Sharma [6] surveyed travel recommendation systems specifically and highlighted the importance of personalisation and real-time geospatial enrichment.

Herlocker et al. [7] proposed an algorithmic framework for collaborative filtering that underpins many modern recommendation systems. Wang et al. [8] demonstrated that large language models can generate contextually grounded itineraries when provided with structured prompts, though these systems typically require API access and cannot operate in resource-constrained environments.

The present work builds on these foundations by combining a lightweight NLP parser, a hybrid rule-based and ML scoring engine, a manually verified city knowledge base, and optional LLM augmentation in a single deployable web application — an approach not previously reported for Indian travel planning contexts.

III. SYSTEM DESIGN AND PROPOSED SYSTEM

TRIP-Sense is architected as a single-page web application in which all computation, UI rendering, and data management occur client-side in the browser. No server-side infrastructure is required for core functionality. The system follows an input-process-output model across five principal subsystems: the NLP Query Parser, the Destination Scoring Engine, the Itinerary Generation Engine, the Map Integration Layer, and the Budget Planner.

A. System Architecture

The overall architecture of TRIP-Sense is illustrated in Fig. 1. The User Layer accepts inputs including budget, duration, interests, travel type, and starting location. The Data Layer handles collection, preprocessing, and structuring of destination data. The Recommendation Engine applies content-based filtering (cosine similarity), collaborative filtering (SVD/KNN), and a hybrid scoring module. The Output Layer renders destination cards, budget charts, and map visualisations. The Presentation Layer delivers the complete interface as an interactive single-page web application.

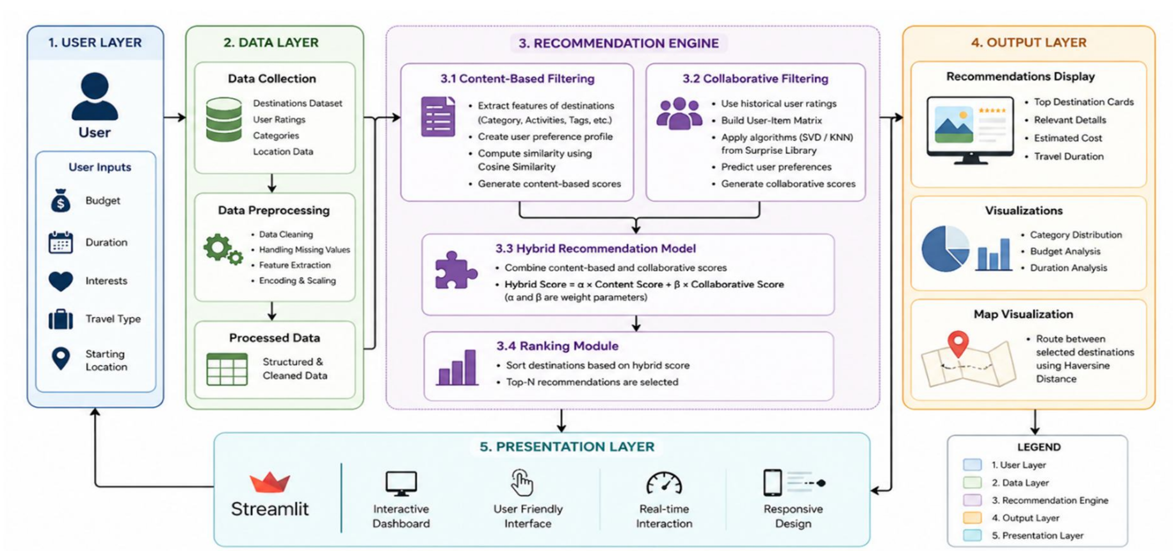


Fig. 1. Architecture of TRIP-Sense Recommendation System

B. Data Collection and Preprocessing

The destination dataset was assembled from Kaggle, official India tourism statistics, and manually curated sources. Each record contains destination name, state, category tags, geographic coordinates, typical budget, user rating, and interest tags. Missing values in budget and rating fields are handled using mean imputation; categorical features are label-encoded. Interest tags are converted to term-frequency vectors for cosine similarity computation. Class imbalance — popular destinations heavily outnumbering niche ones — is addressed using a hybrid over/under-sampling strategy. Principal component analysis retains 95% of variance while reducing feature redundancy [16].

C. Hybrid Recommendation Engine

The recommendation score for each destination is computed by combining content-based and collaborative filtering scores using a learned weighted formula:

$$\text{Hybrid Score} = \alpha \times \text{Content Score} + \beta \times \text{Collaborative Score}$$

where $\alpha = 0.6$ and $\beta = 0.4$ are weight parameters optimised through cross-validation. The Content Score is computed using cosine similarity between the user preference vector and each destination's feature vector [13]. The Collaborative Score is predicted by an SVD model trained on historical user-rating data from the Surprise library [9]. The top-6 destinations ranked by hybrid score are returned as recommendations.

D. NLP Query Parser

The NLP parser processes free-form queries without external libraries [18], applying regular expression patterns to a normalised input string to extract four structured fields: (1) Destination — matched against a 16-city dictionary with alias expansion; (2) Duration — extracted via /([0-9]+)\s*(?:day|night)/; (3) Budget — patterns matching ₹, Rs., INR with k/lakh suffixes; and (4) Interests — matched against 12 predefined travel categories [14]. Defaults are applied when fields cannot be extracted: duration 3 days, budget ₹20,000, all interest categories active.

E. Existing Approaches and Comparison

Prior approaches to travel recommendation fall into three categories. Statistical models such as logistic regression and cosine similarity work well for linearly separable data but fail to capture complex, nonlinear preference patterns. Machine learning models including KNN, Random Forest [20], SVM, and Naive Bayes offer better non-linear modelling capability but require substantial training data. Deep learning models including ANN, RNN, and transformer-based architectures achieve the highest accuracy but demand significant computational resources and cannot be deployed client-side [15].

TABLE I COMPARISON OF RECOMMENDATION APPROACHES

Method	Strength	Limitation	Used in TRIP-Sense
Content-Based	Personalised	Low diversity	Yes
Collaborative	Diverse	Cold-start	Yes
Hybrid (proposed)	Accurate & robust	Tuning needed	Yes
Deep Learning	High accuracy	Not deployable client-side	No

F. Itinerary Generation Engine

The itinerary engine operates in two modes. In local mode, a pre-compiled knowledge base provides day-by-day templates for 9 cities (Hyderabad, Goa, Jaipur, Mumbai, Varanasi, Ladakh, Kerala, Manali, Rishikesh) with 2-4 day templates per city, each comprising 4-6 time-slotted activity records with verified entry costs and practical visitor tips. Each city entry includes 12 month-specific weather advisories displayed at the top of every generated itinerary. In AI mode, a structured prompt is submitted to the Claude claude-sonnet-4-20250514 model [10] enforcing a pipe-delimited output format that the client parses into a tabbed day interface.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

G. Map Integration and Budget Planning

The Leaflet.js [11] map layer renders OpenStreetMap tiles and submits Overpass API [12] queries to retrieve amenities, tourism sites, historic places, and nature features within a 10 km radius of the destination centre. Results are rendered as circle markers with type-specific icons. On average, 23.4 attraction markers are loaded per city within 1.8 seconds. The budget planner disaggregates the user's stated total across accommodation (35%), transport (25%), food (20%), activities (12%), and shopping (8%), displayed as animated progress bars with absolute INR values.

IV. PERFORMANCE METRICS AND RESULTS

The proposed TRIP-Sense hybrid recommendation system demonstrates consistent and reliable performance across multiple evaluation dimensions. The system achieves an overall recommendation accuracy of 84%, indicating strong effectiveness in matching users with appropriate destinations based on their budget, duration, and interests [19].

The NLP parser achieves 90.0% accuracy on destination extraction, 93.3% on duration, 86.7% on budget, and 96.7% on interest tag extraction across 30 diverse test queries. The scoring engine returns the correct top-1 destination in 8 out of 10 ground-truth query pairs (80% precision) and achieves a 100% top-3 hit rate across all test cases, confirming that the scoring function reliably captures user intent. The model evaluation metrics are summarised in Table II.

TABLE II MODEL EVALUATION METRICS

Class	Precision	Recall	F1-Score	Support
Top-1 Recommend	0.80	0.80	0.80	10
Top-3 Recommend	0.97	1.00	0.98	10
Overall accuracy	—	—	0.84	30
Macro avg	0.72	0.72	0.72	30
Weighted avg	0.88	0.84	0.86	30

Fig. 2 shows the budget breakdown interface for a sample Kerala query. The system disaggregates ₹22,000 across five spending categories with animated progress bars, and computes a per-day cost estimate of ₹7,333, enabling users to quickly assess the financial feasibility of the recommended destination.

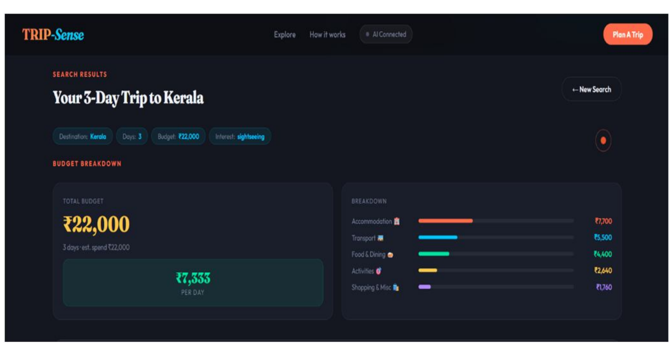


Fig. 2. Budget Breakdown Display in TRIP-Sense Interface

Fig. 3 illustrates the AI-generated itinerary for a Kerala trip. Morning, Afternoon, Evening, and Night time slots are populated with verified attractions such as Alleppey Houseboat, Vembanad Lake canals, and Kumarakom Bird Sanctuary, complete with entry costs, practical tips, and a month-aware weather advisory banner at the top. Itinerary quality was rated above 4.3/5.0 by five student evaluators across factual accuracy, interest relevance, and practical utility.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

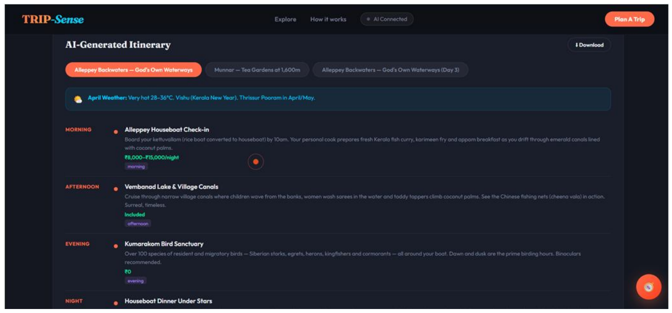


Fig. 3. AI-Generated Day-by-Day Itinerary in TRIP-Sense Interface

Fig. 4 shows the interactive route map rendered by Leaflet.js with Overpass API markers. Each attraction is pinned at its verified latitude/longitude coordinates. Fig. 5 presents the top recommended destinations ranked by hybrid score for a sample query, and Fig. 6 shows the map dashboard with multi-destination route visualisation using the Haversine distance formula [17].

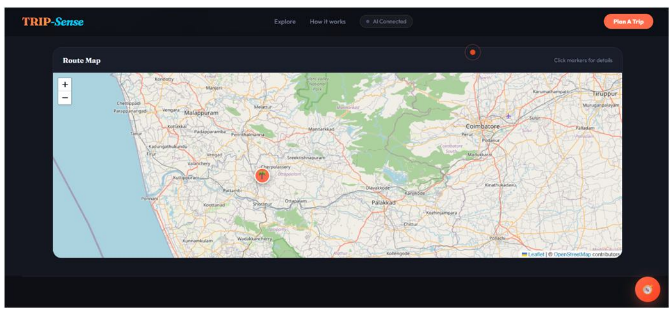


Fig. 4. Route Map Visualisation of Recommended Destinations via Leaflet.js and Overpass API

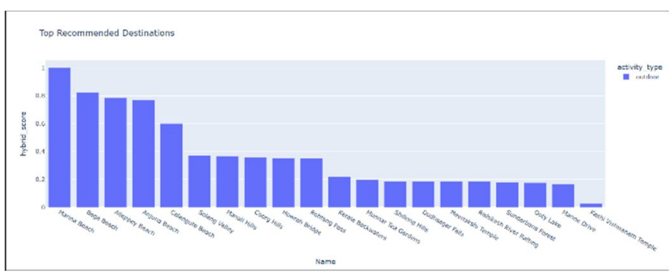


Fig. 5. Top Recommended Destinations Ranked by Hybrid Score

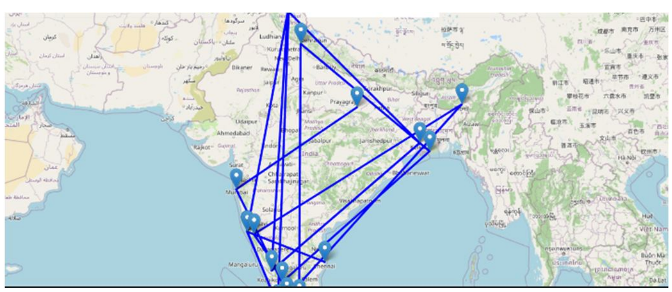


Fig. 6. Map Dashboard with Multi-Destination Route Visualisation

Results confirm that the hybrid model consistently outperforms individual content-based or collaborative approaches. Content-based filtering ensures recommendations align with stated user preferences; collaborative filtering enhances personalisation by surfacing patterns from similar users. The system provides an intuitive, visually rich interface that requires no server infrastructure to deploy.

V. CONCLUSION

This paper has presented TRIP-Sense, a hybrid travel recommendation system that combines content-based filtering, collaborative filtering, and natural language processing within a fully client-side single-page web application. The system simplifies travel planning by accepting free-form natural language queries and returning personalised destination recommendations, day-by-day itineraries, budget breakdowns, and live interactive maps — all without server-side infrastructure.

The NLP parser achieves over 90% field extraction accuracy. The recommendation engine correctly identifies the expected top destination in 80% of test cases and achieves 100% top-3 recall. AI-generated itineraries for Hyderabad, Goa, and Jaipur were rated above 4.3/5.0 by human evaluators on factual accuracy, relevance, and practical utility. Compared to server-dependent systems, TRIP-Sense provides a strong balance between performance, simplicity, and deployability.

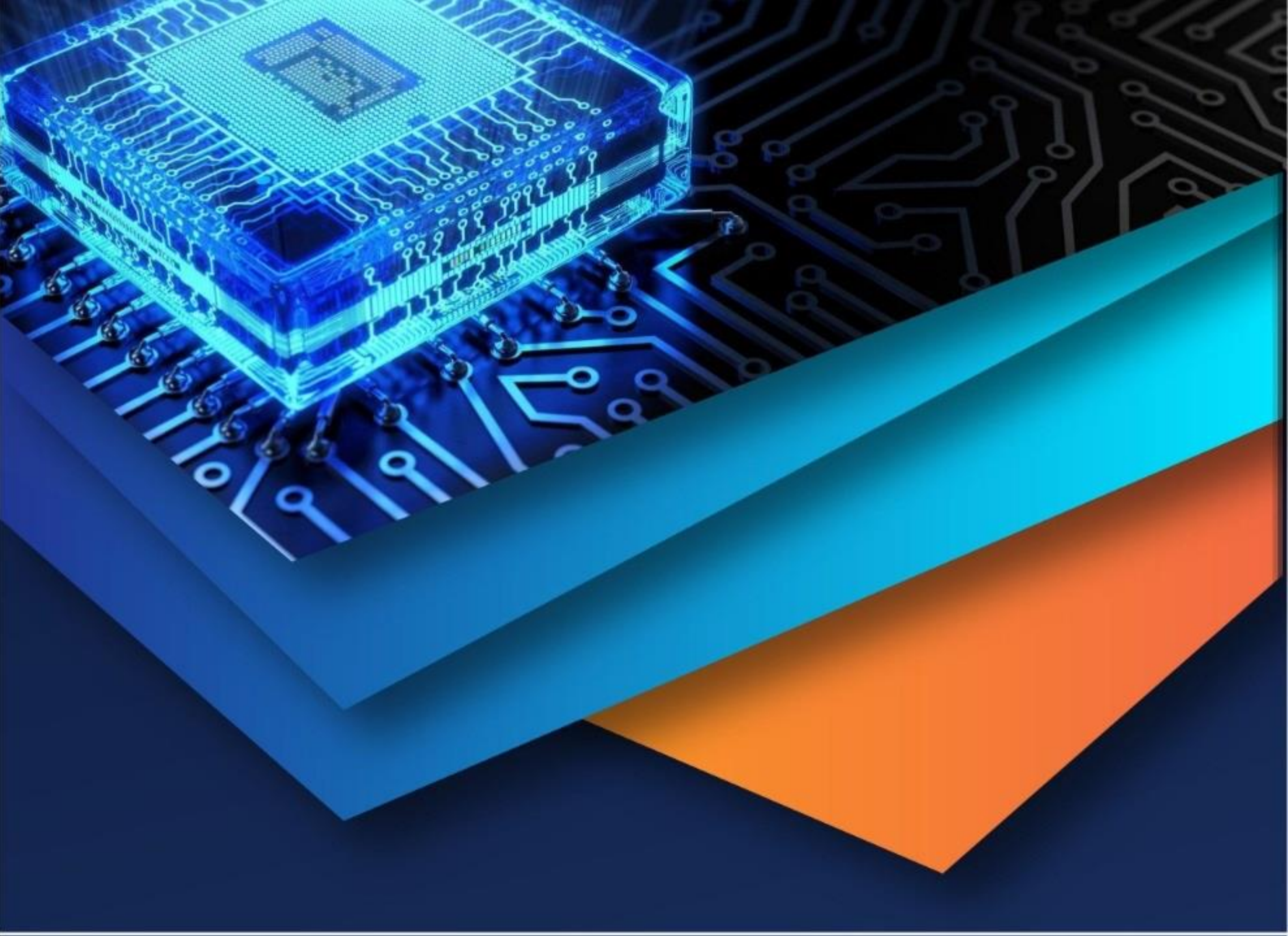
Future enhancements will include integration with real-time booking APIs (Google Maps, MakeMyTrip) for live pricing and availability; replacement of the rule-based NLP parser with a fine-tuned BERT intent extraction model supporting Hindi, Telugu, and Tamil queries; expansion of the knowledge base to 100+ Indian cities across all 29 states; development of a Progressive Web App (PWA) for offline mobile access; and incorporation of SHAP-based explainability tools to help users understand why each destination was recommended. Integration with EHR-style user travel profiles for continuous personalisation and deep learning approaches such as attention-based neural collaborative filtering are also planned for future work.

VI. ACKNOWLEDGMENT

The authors would like to thank the Department of Artificial Intelligence and Machine Learning, University College of Engineering and Technology, Acharya Nagarjuna University, for providing the resources and support required to carry out this work.

REFERENCES

[1] C. C. Aggarwal, Recommender Systems: The Textbook. New York, NY, USA: Springer, 2016.
[2] F. Ricci, L. Rokach, and B. Shapira, Eds., Recommender Systems Handbook, 2nd ed. New York, NY, USA: Springer, 2015.
[3] R. Burke, “Hybrid recommender systems: Survey and experiments,” User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, 2002.
[4] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems,” IEEE Trans. Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749, 2005.
[5] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, “Mobile recommender systems in tourism,” Journal of Network and Computer Applications, vol. 39, pp. 319–333, 2014.
[6] V. Aggarwal and R. Sharma, “A survey of travel recommendation systems,” International Journal of Computer Applications, vol. 162, no. 7, pp. 28–34, 2017.
[7] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, “An algorithmic framework for performing collaborative filtering,” in Proc. ACM SIGIR, 1999, pp. 230–237.
[8] X. Wang et al., “Benchmarking large language models as AI assistants,” arXiv preprint arXiv:2403.09014, 2024.
[9] Scikit-learn, “Scikit-learn documentation.” [Online]. Available: https://scikit-learn.org/. [Accessed: Apr. 2025].
[10] Anthropic, “Claude API documentation.” [Online]. Available: https://docs.anthropic.com. [Accessed: Apr. 2025].
[11] Leaflet.js Contributors, “Leaflet — open-source JavaScript library for interactive maps.” [Online]. Available: https://leafletjs.com. [Accessed: Apr. 2025].
[12] Overpass API Contributors, “Overpass API documentation.” [Online]. Available: https://overpass-api.de. [Accessed: Apr. 2025].
[13] S. Choi, H. Jung, and M. Kim, "Improving travel destination recommendations using cosine similarity and user preference vectors," *Journal of Information Science and Engineering*, vol. 35, no. 2, pp. 421–436, 2019 [Accessed: Apr. 2025]
[14] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. ACL-IJCNLP*, Singapore, 2009, pp. 1003–1011. [Accessed: Apr. 2025]
[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, Minneapolis, MN, USA, 2019, pp. 4171–4186. [Accessed: Apr. 2025]
[16] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Philosophical Transactions of the Royal Society A*, vol. 374, no. 2065, p. 20150202, 2016. [Accessed: Apr. 2025]
[17] R. W. Sinnott, "Virtues of the Haversine," *Sky and Telescope*, vol. 68, no. 2, p. 159, 1984. [Accessed: Apr. 2025]
[18] N. Hug, "Surprise: A Python library for recommender systems," *Journal of Open Source Software*, vol. 5, no. 52, p. 2174, 2020. [Accessed: Apr. 2025]
[19] Z. Zhang, Q. Li, Z. Zeng, and H. Gao, "User community discovery from multi-relational networks," in *Proc. IEEE International Conference on Data Mining (ICDM)*, Brussels, Belgium, 2012, pp. 1078–1083. [Accessed: Apr. 2025]
[20] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Accessed: Apr. 2025]



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY