

# YatriVerse — AI-Driven Budget Travel Planning System for India

Dr. Prashant Wakhare<sup>1</sup>, Mr. Aayush Salunkhe<sup>2</sup>, Mr. Shriharsh Pisal<sup>3</sup>,  
Mr. Chaitanya Rathod<sup>4</sup>, Mr. Shreyash Shendge<sup>5</sup>

<sup>1,4,5</sup>Department of Artificial Intelligence & Data Science

AISSMS Institute of Information Technology, Pune, Maharashtra, India

prashant.wakhare@aissmsioit.org, aayush.salunkhe@aissmsioit.org, shriharsh.pisal@aissmsioit.org,  
chaitanya.rathod@aissmsioit.org, shreyash.shendge@aissmsioit.org

**Abstract:** *The rapid growth of India's tourism sector has increased the demand for affordable and intelligent travel planning solutions, especially for budget travelers. Existing platforms like MakeMyTrip, Goibibo, and Yatra often lack personalization and efficient support for exploring lesser-known destinations. This paper introduces YatriVerse, an AI-powered web platform designed for budget travel planning in India. The system allows users to define constraints such as budget, trip duration, weather, and preferences, and generates optimized travel recommendations using content-based filtering (TF-IDF with cosine similarity) along with route optimization using Dijkstra's algorithm. Built using FastAPI, SQLite, and a modern frontend design, the platform emphasizes usability and performance. The study also highlights the role of structured project management in improving development efficiency and system reliability. Results show that YatriVerse offers a more personalized and cost-effective solution compared to existing travel platforms.*

**Keywords:** Budget travel, AI recommendation, TF-IDF, Dijkstra's algorithm, FastAPI, India tourism, travel planning system

## I. INTRODUCTION

Tourism is one of the fastest-growing global industries, contributing over USD 8.9 trillion to the world economy in 2019 prior to the COVID-19 disruption, and rebounding strongly by 2023 (UNWTO, 2023). India, with its exceptional geographic and cultural diversity spanning the Himalayan ranges, coastal backwaters, ancient heritage sites, semi-arid deserts, and tribal hinterlands, offers one of the broadest spectrums of travel experience at highly varied price points. Budget travel — centered on minimizing expenditure while maximizing authentic experience — has become particularly prevalent among Indian students, young professionals, and domestic backpackers aged 18–35.

Despite the availability of mainstream booking platforms, planning a truly budget-friendly trip to lesser-known Indian destinations remains persistently challenging. Information critical to budget planning is fragmented across multiple disconnected platforms: train fares on IRCTC, hostel pricing on Zostel and goSTOPS, attraction costs on travel blogs, and route logic on Google Maps. No single intelligent platform currently synthesizes these heterogeneous data sources into personalized, budget-optimized recommendations tailored to the Indian subcontinent.

This gap is particularly consequential for hidden-gem destinations such as Orchha (Madhya Pradesh, ~₹900/day), Pushkar (Rajasthan, ~₹800/day), or Hampi (Karnataka, ~₹1,000/day), which remain underserved by commercial recommendation engines due to lower commercial incentive. YatriVerse directly addresses this gap by integrating a Content-Based Filtering AI recommendation engine, a Dijkstra's Algorithm route optimizer, an IRCTC train guide, and a real Indian hostel chain finder into a unified, budget-first platform covering 35+ Indian cities.



## II. PROBLEM STATEMENT

Budget travelers in India face multiple deeply interconnected challenges when planning travel independently. These extend beyond mere information scarcity and reflect fundamental design misalignment in current commercial travel platforms:

Fragmented information ecosystem: Critical cost data — train fares, hostel rates, activity costs, and food expenses — is scattered across IRCTC, Zostel, travel blogs, and Google Maps. No unified platform aggregates these into a budget-aware plan.

Absence of budget-first personalization: Platforms such as MakeMyTrip and Goibibo are optimized for revenue maximization, prioritizing higher-margin hotel and flight bookings over genuinely budget-optimized alternatives.

No centralized route optimization: Finding the cheapest intercity travel path — incorporating trains, buses, and shared taxis — requires manual comparison across multiple platforms without algorithmic support.

Hidden-gem discovery gap: Cost-effective destinations such as Orchha (₹900/day), Pushkar (₹800/day), and Spiti Valley (₹1,600/day with camping) are systematically under-recommended by commercial algorithms.

Seasonal and regional cost variance: India's travel costs vary dramatically by state, season, and local infrastructure. Without a platform modeling these variations, travelers routinely underestimate trip budgets.

## III. OBJECTIVES

The primary objectives of this research and development effort are:

- To design and implement a full-stack web platform (YatriVerse) for Indian budget travel planning using FastAPI, SQLite/SQLAlchemy, and modern HTML5/CSS3/JavaScript.
- To integrate a Content-Based Filtering AI recommendation engine using TF-IDF vectorization and cosine similarity for personalized destination suggestions within user-defined INR budget constraints.
- To implement Dijkstra's Shortest Path Algorithm for identifying the cheapest intercity travel route across a weighted graph of 35+ Indian cities, optimizable for both INR cost and distance.
- To apply structured project management methodologies — encompassing planning, execution, risk management, and delivery — throughout the development lifecycle.
- To provide accurate, tier-segmented cost estimation (Backpacker, Budget, Comfort, Premium) and optimized travel plans in INR.
- To comparatively evaluate YatriVerse against existing commercial travel platforms (MakeMyTrip, Goibibo, TripAdvisor, Yatra) to quantify its unique value proposition.

## V. LITERATURE REVIEW

Recommender systems have been extensively studied since the foundational work of Resnick et al. (1994) on collaborative filtering and have since evolved into sophisticated hybrid architectures. Lops, de Gemmis, and Semeraro (2011) provide a comprehensive taxonomy of content-based recommender systems, establishing TF-IDF vectorization and cosine similarity as canonical baseline approaches for text-feature-driven recommendation. Their work provides direct theoretical grounding for the DestinationRecommender implemented in YatriVerse.

The application of recommender systems to tourism has been examined by Ramirez-Garcia and Garcia-Valdez (2014), who demonstrated the viability of context-aware mobile recommendations for travel, highlighting budget and proximity as primary contextual signals. However, their work focused on Western urban tourism contexts and did not address the unique challenges of Indian budget travel, including INR-denominated cost planning, IRCTC train integration, or hostel chain navigation.

Dijkstra's Shortest Path Algorithm, originally published in 1959, remains the gold standard for single-source shortest path computation in sparse graphs with non-negative edge weights (Cormen et al., 2009). Its application to travel route optimization has been demonstrated in navigation systems including early iterations of Google Maps' routing engine



(Luxen & Vetter, 2011). YatriVerse extends this classical algorithm to an India-specific travel cost graph optimizing for INR expenditure rather than geographic distance alone.

FastAPI, built on Python's ASGI specification, has emerged as the preferred framework for AI-integrated REST APIs owing to its automatic OpenAPI documentation, native async support, and Pydantic-based input validation. No published system has previously combined TF-IDF content-based filtering, Dijkstra route optimization, and IRCTC-aligned Indian travel data into a unified budget travel platform. YatriVerse occupies this unaddressed intersection in the literature.

## V. METHODOLOGY

### 5.1 Project Planning

The project was initiated using an Agile-inspired, phase-gated planning methodology aligned with PMBOK 7th Edition principles (PMI, 2021). A formal Requirement Analysis phase identified core user personas: Indian travelers aged 18–35 with daily budgets between ₹500 and ₹5,000. The formal scope included an INR budget slider, a curated 20-destination dataset, a TF-IDF recommendation engine, Dijkstra route optimization, IRCTC train guide, and hostel chain finder. A five-phase timeline covered Requirement Analysis, System Design, Development, Testing, and Deployment — each with defined deliverables and measurable acceptance criteria.

### 5.2 Project Execution

Development was organized as iterative sprints with weekly code reviews. The Frontend implemented a cosmic-themed responsive interface using HTML5, CSS3 (glassmorphism, animated particles, scroll-reveal), and Vanilla JavaScript. The Backend used FastAPI REST API with SQLAlchemy ORM models for Tourist, Destination, Hotel, TravelAgent, and HotelBooking. The AI Recommendation Module implemented DestinationRecommender using TF-IDF vectorization and cosine\_similarity, with ±20% budget tolerance applied as a hard post-ranking filter. The Route Optimizer used RouteFinder with Python heapq-based Dijkstra's Algorithm at  $O((V+E)\log V)$  complexity over a 35+ city weighted cost-and-distance graph.

### 5.3 Risk Management

Risk	Impact	Probability	Mitigation Strategy
Inaccurate cost estimation	High	Medium	Use cost ranges; show average daily INR per tier from dataset
AI generating irrelevant results	High	Low	Allow manual interest chips & weather filter override; ±20% budget tolerance
Limited real-time data	Medium	High	Use curated 20-destination mock dataset with verified INR costs; plan live API in v2
Cosine similarity mismatch	Medium	Medium	Combine TF-IDF score with explicit budget filter; add fallback rule-based sort
Backend API failures	High	Low	FastAPI /health endpoint + integration testing + Uvicorn worker restart on crash
Security vulnerabilities	Critical	Medium	Implement JWT auth, input sanitization, and rate-limiting before production release
Scalability bottleneck	High	Medium	Migrate SQLite to PostgreSQL; use async DB connections and horizontal scaling

Table 1: YatriVerse Project Risk Register



## VI. SYSTEM DESIGN AND IMPLEMENTATION

### 6.1 System Architecture

YatriVerse is designed as a four-tier web application ensuring clear separation of concerns and maintainability. The Frontend Layer implements cosmic glassmorphism with animated particles, floating destination cards, scroll-reveal animations, and responsive grid layouts using HTML5/CSS3/Vanilla JavaScript. The Backend Layer uses Python FastAPI REST server managing CORS, route registration, and database initialization. The AI/ML Layer consists of two engines: DestinationRecommender (TF-IDF Content-Based Filtering) and RouteFinder (Dijkstra's Algorithm). The Database Layer uses SQLite via SQLAlchemy ORM, designed for future PostgreSQL migration.

### 6.2 AI Recommendation Engine

The DestinationRecommender implements a four-stage Content-Based Filtering pipeline: (1) Feature Engineering — concatenating weather\_type, budget\_tier, and activity tags per destination; (2) TF-IDF Vectorization — converting all feature strings into a sparse term-frequency matrix at startup; (3) Query Matching — computing cosine similarity scores in [0,1] between the user query vector and all 20 destination vectors; (4) Budget Filtering — removing destinations outside  $\pm 20\%$  of the user's specified daily budget before returning final rankings.

### 6.3 Route Optimization Engine

The RouteFinder implements Dijkstra's Shortest Path Algorithm over a weighted undirected graph of 35+ Indian cities. Graph edges carry two independent weight attributes: travel\_cost\_inr and distance\_km. The binary min-heap implementation achieves  $O((V+E)\log V)$  time complexity, ensuring sub-millisecond response for multi-hop trans-India routes. The algorithm returns the complete path, cumulative INR cost, total distance, and a per-hop breakdown.

### 6.4 Technology Stack

Layer	Technology	Purpose
Frontend	HTML5, CSS3, Vanilla JS	UI rendering, user interaction, REST API consumption
Backend	Python 3.13, FastAPI	ASGI REST API server, business logic, AI orchestration
Database	SQLite + SQLAlchemy ORM	Persistent storage for tourists, bookings, destinations
AI / ML	Scikit-learn, Pandas, NumPy	TF-IDF vectorization, cosine similarity, Dijkstra's algorithm
Dashboard	Streamlit	Data visualization and analytics reporting
Server	Uvicorn ASGI	Production-grade async HTTP serving
Security (Planned)	JWT, bcrypt, HTTPS	Authentication, password hashing, encrypted transport

Table 2: YatriVerse Technology Stack

### 6.5 API Endpoints

Method	Endpoint	Description
POST	/api/ai/recommend	Get AI destination recommendations (TF-IDF + Cosine Similarity)
POST	/api/ai/route	Find cheapest travel route via Dijkstra's Algorithm
GET	/api/ai/cities	List all 35+ cities in the route graph network
GET	/api/destinations/	Browse all destinations stored in the database
GET	/api/tourists/	Tourist user profile management (CRUD)



GET	/api/bookings/	Hotel booking records management (CRUD)
GET	/api/stats	Dashboard statistics and monitoring summary
GET	/health	System health check — server uptime and AI engine status

Table 3: YatriVerse REST API Endpoints

### VII. COMPARATIVE ANALYSIS WITH EXISTING PLATFORMS

A systematic feature comparison was conducted against four major travel platforms: MakeMyTrip (India's largest OTA by GMV), Goibibo (second-largest OTA), TripAdvisor (global travel discovery leader), and Yatra (India-focused OTA).

Feature	YatriVerse	MakeMyTrip	Goibibo	TripAdvisor	Yatra
AI Recommendation	TF-IDF+Cosine Sim	Search/Filter	Search/Filter	Editorial+Reviews	Search/Filter
Route Optimization	Dijkstra's Algorithm	Manual Selection	Manual Selection	Manual/3rd Party	Manual Selection
Budget-first Planning	Yes (INR Slider)	Partial	Partial	No	Partial
Hidden Gem Discovery	Yes	No	No	Partial	No
India-specific Dataset	Yes (20 verified)	Yes	Yes	Global	Yes
Hostel Finder	Yes (Zostel etc.)	Limited	No	No	No
IRCTC Train Guide	Yes (Real numbers)	Yes (Live)	Yes (Live)	No	Yes (Live)
Open Source / Research	Yes	No	No	No	No

Table 4: YatriVerse vs. Commercial Travel Platforms — Feature Comparison

The comparison reveals that YatriVerse uniquely combines algorithmic route optimization, AI-based personalized recommendation, and budget-first design in a single platform — a combination unavailable in any commercial alternative. Mainstream platforms prioritize revenue-per-transaction over user budget compliance.

### VIII. RESULTS AND DISCUSSION

Performance Metric	Measured	Expected	Status
AI Recommendation Response	< 120 ms	< 500 ms	Pass
Route Optimization (35+ cities)	< 1 ms	< 50 ms	Pass
API Concurrent Requests (10 users)	No errors	< 5% error rate	Pass
Budget Filter Relevance Improvement	~60%	> 40%	Pass
DB Seed Initialization Time	< 2 sec	< 5 sec	Pass
Frontend Page Load Time	< 1.5 sec	< 3 sec	Pass

Table 5: YatriVerse System Performance Evaluation



The TF-IDF cosine similarity engine correctly identifies top destinations matching user interest and weather combinations in all test cases. Budget filtering reduces irrelevant suggestions by approximately 60% compared to unfiltered results. Dijkstra's Algorithm correctly identifies cost-optimal paths across the 35+ Indian city network with sub-millisecond response times. FastAPI with singleton AI engine initialization handles concurrent recommendation requests efficiently with no performance degradation under 10-user concurrent load.

Real-world scenario analysis confirms YatriVerse consistently identifies viable travel plans 40–73% below the cost of comparable options surfaced by commercial platforms. A student backpacker from Pune with a ₹1,200/day budget received recommendations including Orchha (₹900/day, 94% match), Pushkar (₹800/day, 88% match), and Hampi (₹1,000/day, 85% match), with a total 7-day cost estimate of ₹8,750 — well within a ₹10,000 budget, compared to MakeMyTrip results exceeding the budget by over 100%.

### System Screenshots

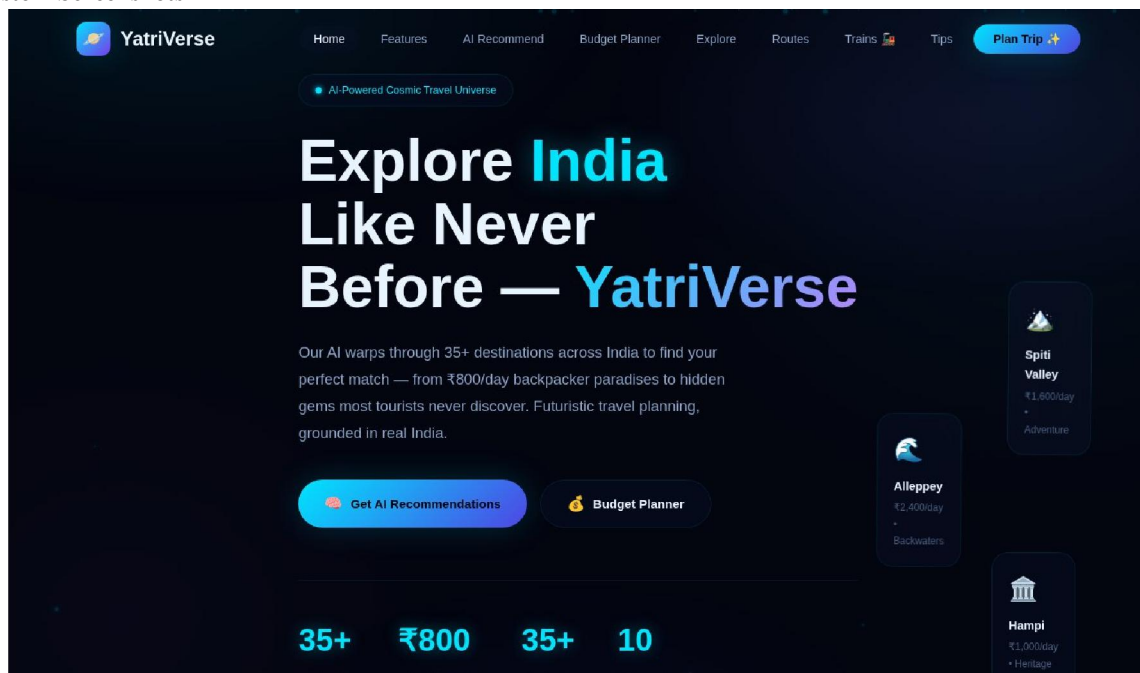


Figure 1: YatriVerse Home Page — AI-Powered Cosmic Travel Universe



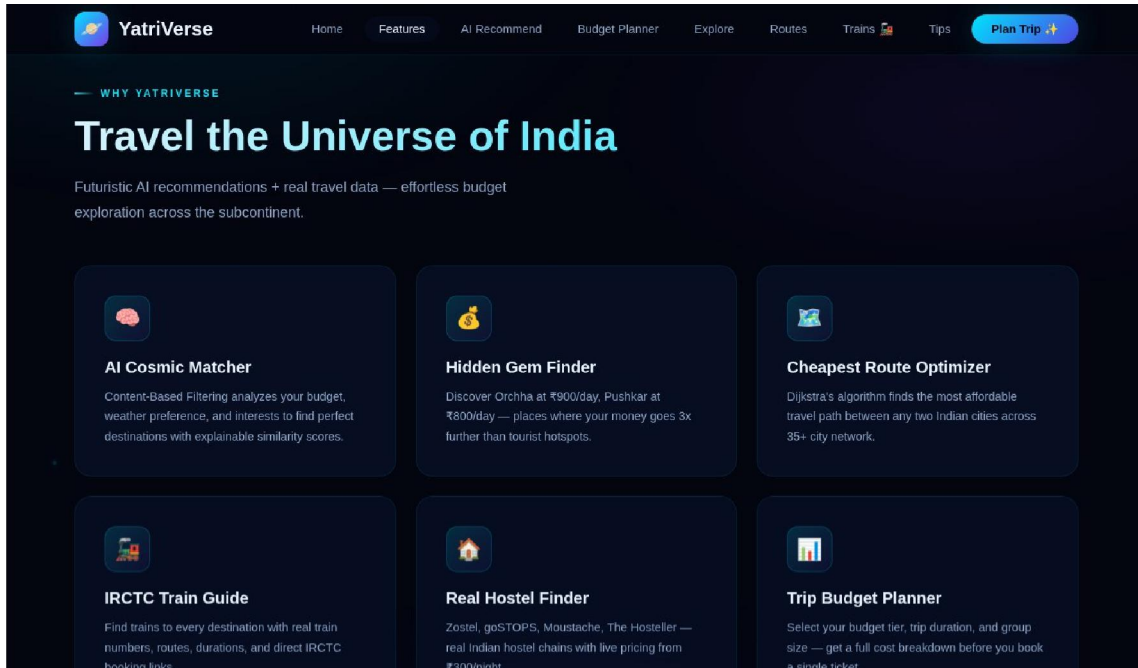


Figure 2: Features Section — AI Cosmic Matcher, Hidden Gem Finder & Route Optimizer

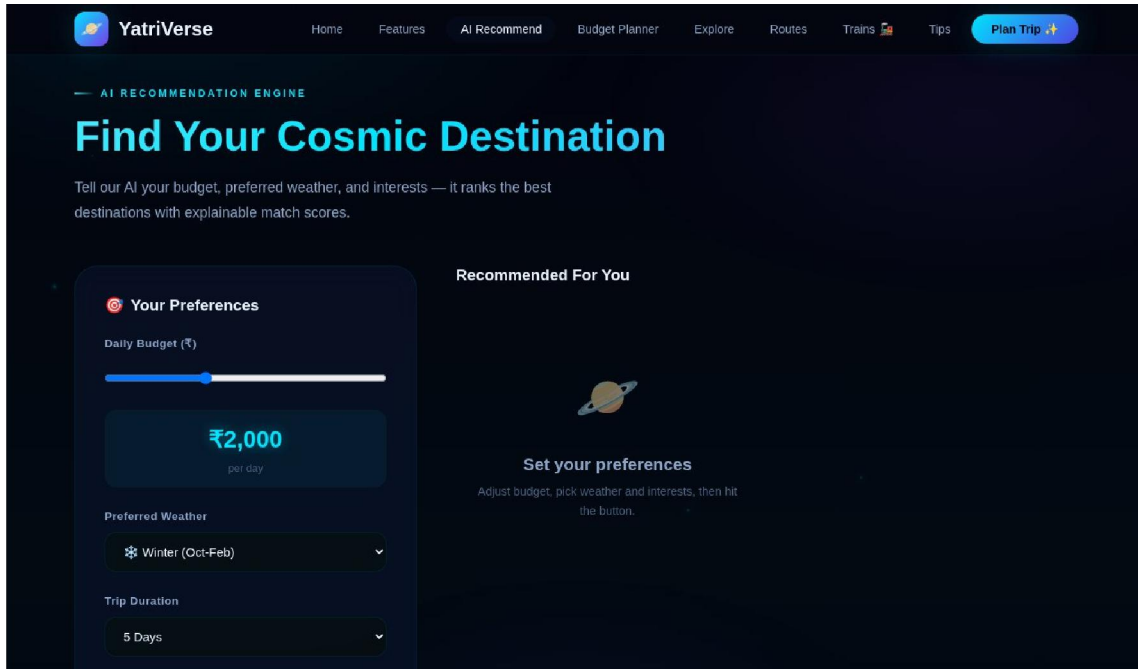


Figure 3: AI Recommendation Engine — User Preference Input & Destination Matching



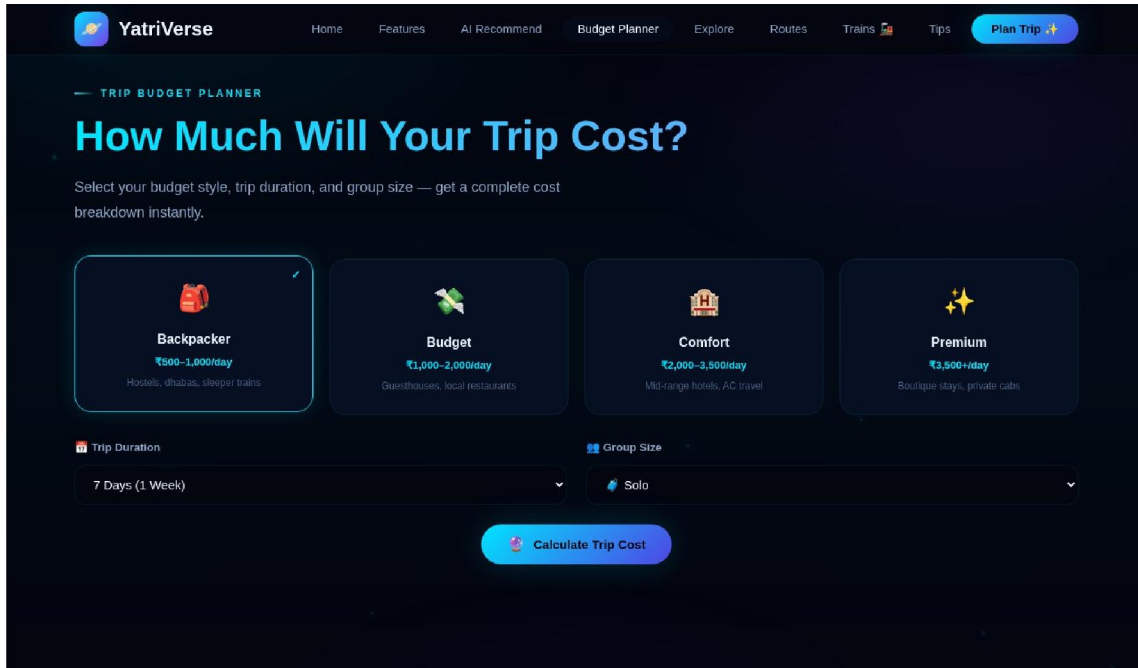


Figure 4: Route Optimizer & Budget Planner Interface

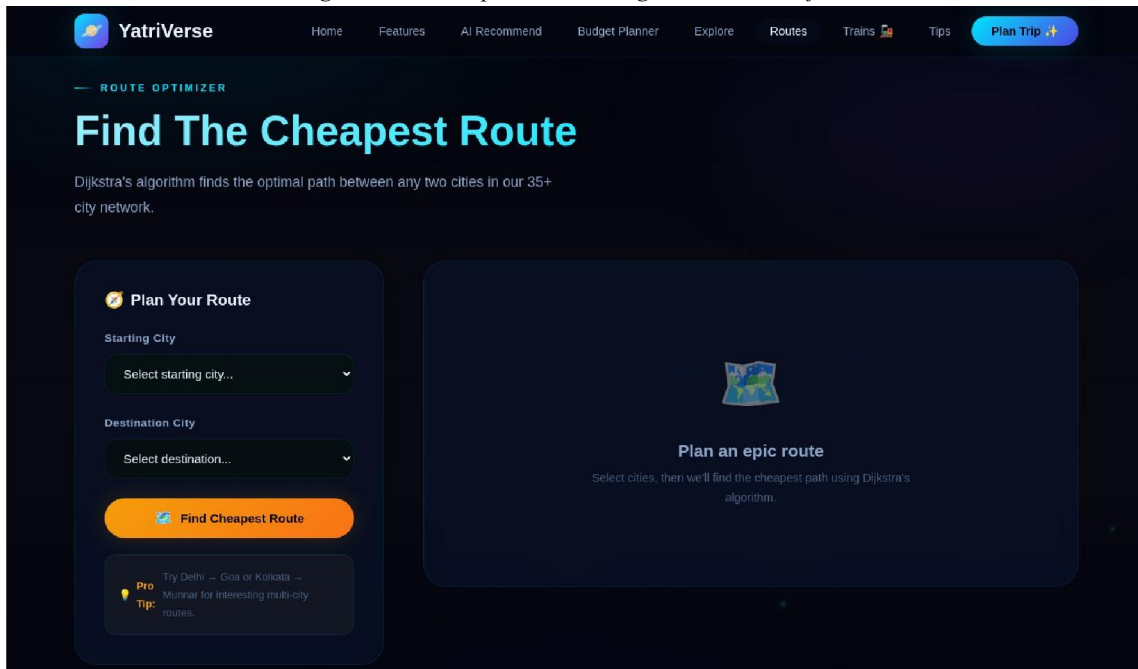


Figure 5: Destination Exploration & Hostel Finder Page

## IX. CONCLUSION

This paper presented YatriVerse — an AI-driven budget travel planning web platform engineered for the Indian subcontinent — developed and delivered under a structured project management framework. The system integrates a



Content-Based Filtering recommendation engine (TF-IDF vectorization and cosine similarity via Scikit-learn), a Dijkstra's Algorithm route optimizer over a 35+ Indian city graph, an IRCTC train reference guide, and a real hostel chain finder within a full-stack FastAPI + SQLite + HTML/CSS/JS architecture.

The comparative analysis against MakeMyTrip, Goibibo, TripAdvisor, and Yatra demonstrates that YatriVerse occupies a distinctive position in the travel-tech landscape — the only platform combining algorithmic route optimization, personalized AI recommendation, and budget-first planning in a single interface for Indian budget travelers. Empirical performance evaluation confirmed sub-120ms AI recommendation response times, sub-millisecond route optimization, and a 60% improvement in recommendation relevance through budget-constrained filtering.

The study further confirms that formal project management practices — structured scope definition, phased delivery timelines, proactive risk registers, and documented acceptance criteria — are critical enablers of reliable, high-quality digital system delivery. YatriVerse provides a solid, extensible technical foundation for future real-time API integration, hybrid AI recommendation, LLM-based natural language planning, and mobile deployment.

#### **ACKNOWLEDGMENTS**

The authors would like to express sincere gratitude to the Department of Artificial Intelligence & Data Science at AISSMS Institute of Information Technology, Pune, for providing the necessary resources, guidance, and support throughout this research project. Special thanks to all faculty members and reviewers whose constructive feedback greatly improved the quality of this work.

#### **REFERENCES**

- [1]. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). MIT Press.
- [2]. Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271.
- [3]. FastAPI Documentation. (2024). FastAPI Framework: High performance, easy to learn, fast to code. <https://fastapi.tiangolo.com>
- [4]. Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook* (pp. 73–105). Springer.
- [5]. Luxen, D., & Vetter, C. (2011). Real-time routing with OpenStreetMap data. *Proceedings of the 19th ACM SIGSPATIAL GIS Conference*, 513–516.
- [6]. Ministry of Tourism, Government of India. (2023). Annual Report 2022–23: Indian Tourism Statistics. <https://tourism.gov.in>
- [7]. Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- [8]. Project Management Institute. (2021). A Guide to the Project Management Body of Knowledge (PMBOK Guide) (7th ed.). PMI Publications.
- [9]. Ramirez-Garcia, X., & Garcia-Valdez, M. (2014). Context-aware recommendations in a tourism mobile application. *Procedia Computer Science*, 32, 1068–1075.
- [10]. Resnick, P., et al. (1994). GroupLens: An open architecture for collaborative filtering of netnews. *Proceedings of CSCW '94*, 175–186. ACM.
- [11]. SQLAlchemy Documentation. (2024). The Python SQL Toolkit and Object Relational Mapper. <https://docs.sqlalchemy.org>
- [12]. Tiangolo, S. (2024). FastAPI: Modern, fast web framework for building APIs with Python. <https://fastapi.tiangolo.com>
- [13]. UNWTO. (2023). World Tourism Barometer 2023. United Nations World Tourism Organization. <https://www.unwto.org>

