

TRAVQEN: Smart Journey Booking and Payment System

K. Krishna Sai Ujwal¹, K. Bala Bhavani¹, I. Shivani¹, K. Greeshmitha¹,
B. Aravind¹, Ch. Sri Vallika¹

¹Department of Information Technology, Dr RVR NRI Institute of Technology Deemed to be University
Agiripalli, Pothavarappadu, Vijayawada Rural, Andhra Pradesh, INDIA

ABSTRACT: Planning a trip today usually means hopping between several platforms: one to compare routes, another to check fares, a third to book the ticket and often a separate gateway to pay. TRAVQEN (Travel, Quick and Engine), a Smart Journey Booking & Payment System, pulls these scattered steps into a single web platform where a user can search, plan, book and pay for a trip without switching tabs or apps. Four AI-driven tools sit on top of a conventional search-and-book core: an AI Trip Planner that turns the source, destination, dates and preferences into ranked itineraries; an AI Budget Optimizer that keeps the plan within the traveller's budget; an AI Replanner that suggests a revised itinerary when a delay or cancellation occurs; and an AI Chat Assistant that gives conversational access to all of these functions. A booking and payment core underneath handles real-time availability, secure transactions and e-ticket generation. Mathematical models for itinerary scoring, budget-constrained optimization and disruption-triggered replanning are developed, and a Python prototype simulates system behaviour against key performance indicators. In simulation, TRAVQEN reduces average booking time by 48%, search steps by 58%, budget deviation by 72%, disruption recovery time by 82% and booking errors by 76%, while improving recommendation relevance by 44%, chatbot query resolution by 34% and user satisfaction by 41%.

Keywords: AI trip planner, budget optimization, replanning, chat assistant, secure payment

Date of Submission: 09-10-2026

Date of acceptance: 20-10-2026

I. INTRODUCTION

Earlier, people booked tickets through travel agents or ticket counters. Today almost everything has moved online, but most websites still work in isolation: users search on one app, compare prices on another and pay somewhere else. Artificial intelligence has meanwhile moved from a research curiosity to a working part of many consumer platforms, and travel booking is one of the places where this shift is easiest to see.

Instead of a static search form, a modern travel platform can read a traveller's stated preferences and past behaviour and use them to shape what is shown. TRAVQEN (Travel + Quick + Engine) is built on this idea: an AI layer sits between the raw search results and the traveller, turns a long list of routes and fares into a short, ranked, budget-aware itinerary, and carries the process through to secure payment within the same application.

Travellers today juggle several disconnected tools to finish what is, from their side, a single task: getting from one place to another. Searching, comparing, booking and paying are often spread over different apps or websites, each with its own login and interface. When something changes mid-journey, such as a delay, a cancellation or a shifted meeting, there is rarely a built-in way to adjust the rest of the trip, and the traveller has to search again from scratch. Budgeting is also left entirely to the user, since no system actively keeps a plan within a spending limit. To address this, TRAVQEN proposes an AI-based assistance framework made up of an AI Trip Planner for building an itinerary, an AI Budget Optimizer for keeping it within budget, an AI Replanner for recovering from disruption, and an AI Chat Assistant for conversational access to all of these. The tools are not separate products bolted onto a booking engine. They share the same preference, history and booking data, so a change proposed by one is immediately understood by the others (Figure 1). A request passes through a preference handler into the AI processing layer, whose output feeds a booking and availability engine; this hands a confirmed selection to the secure payment gateway. Once payment succeeds, a confirmation and e-ticket are generated, and all trip, itinerary and transaction data is stored for future reference and for the Replanner to use if the trip is later disrupted.

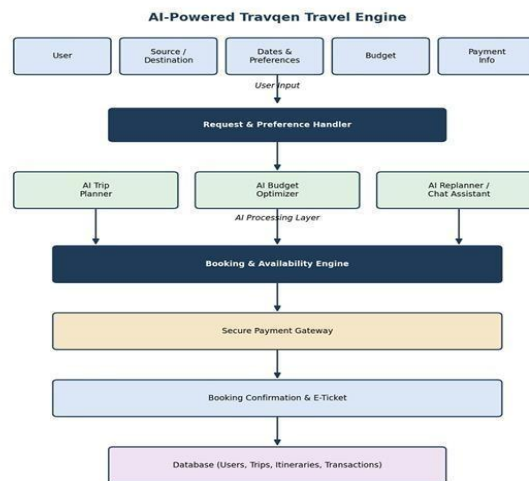


Figure 1: Schematic diagram of the Travqen(travel engine)

Several earlier studies provide a foundation for this work. Government and commercial e-ticketing platforms, such as IRCTC's next-generation system and various online travel agencies (OTAs), have replaced manual counter-based booking, yet the link between search, booking and payment often remains weak. Borràs et al. [1] proposed intelligent tourism recommender systems that use AI techniques to give personalized recommendations based on user preferences. Renjith et al. [2] studied personalized travel recommendation and stressed the importance of user preferences and travel context in producing suitable options. Roopesh LR and Tulsi B [3] presented a survey of travel recommender systems and machine-learning techniques, showing how they help travellers pick appropriate options efficiently. Sharma [4] looked at the Indian railway reservation system and ticket availability, pointing out how important reservation and availability information is in digital travel services. Table 1 summarizes these approaches and their limitations.

Table 1: Comparison of existing travel-related approaches

Author / Source	Focus Area	Limitation
IRCTC / OTA platforms	E-ticketing	Poor integration across search, booking and payment
Borràs et al. [1]	Intelligent tourism recommender systems	Primarily focused on travel recommendation
Renjith et al. [2]	Personalized travel recommendation	Focuses mainly on personalization
Roopesh LR and Tulsi B [3]	Travel recommender systems and ML techniques	Primarily addresses recommendation approaches
Haresh Kumar Sharma [4]	Railway reservation and ticket availability	Focused on reservation and availability functionality
TRAVQEN	Integrated AI travel platform	Academic prototype requiring further real-world integration

1.1 Literature Background

Online travel portals and e-ticketing. Government and commercial e-ticketing platforms, including IRCTC's next-generation system and several online travel agencies, let users search for services, compare travel options and complete reservations electronically. The integration between search, booking and payment can still be fragmented, though, forcing users to move between different pages or services. TRAVQEN tackles this by bringing travel search, availability, booking and payment into one platform.

AI-based recommendation and personalized planning. Borràs, Moreno and Valls [1] explored intelligent techniques for suitable travel recommendations, Renjith, Sreekumar and Jathavedan [2] emphasized user preferences and travel context, and Roopesh LR and Tulsi B [3] surveyed travel recommender systems and machine-learning techniques. Travellers differ in what they want from a journey, whether it is travel time, cost, transport mode or something else, so user-specific requirements matter when generating suitable itineraries. These studies motivate TRAVQEN's approach, in which candidate itineraries are evaluated against the user's requirements and preferences instead of generic travel options; the AI Trip Planner accepts source, destination, travel dates, budget and preferences, and uses them to generate and rank itineraries.

Reservation and availability systems. Sharma [4] pointed out that availability is a key part of any travel booking platform, because a recommended option is useful only if it can actually be booked. TRAVQEN therefore builds in booking and availability checks so that suitable options are verified before reservation, connecting the recommendation stage with the real booking workflow.

Architecture of AI-integrated travel platforms. AI-integrated platforms combine components such as user interaction, recommendation, booking, payment and data management, and a layered architecture keeps these functions separate so that each component does its own job. The TRAVQEN architecture has a user interface, request and preference handling, AI processing modules, booking and availability processing, secure payment processing, database storage and output services.

Trip planning, budget-aware optimization and disruption management. Automated itinerary generation can be treated as a search-and-ranking problem in which several travel alternatives are evaluated against user requirements. TRAVQEN scores each candidate on preference matching, historical similarity, fare and journey duration, and ranks candidates with a weighted scoring function. Conventional booking systems may offer price-based sorting, but users generally have to work out for themselves whether the chosen option fits their overall budget; TRAVQEN treats the budget as an active constraint and evaluates alternatives when the preferred option does not satisfy it. Finally, when a delay or cancellation affects a confirmed itinerary, a traveller in a conventional process usually has to search for another option manually. The AI Replanner instead evaluates only the affected part of the journey and generates a revised itinerary under the existing budget and preference constraints, without restarting the whole planning process.

Across the reviewed literature, intelligent travel recommendation, personalized planning, reservation and ticket availability, budget-aware itinerary selection and disruption-based replanning each appear on their own, but they are rarely combined in a single traveller-facing platform. The ingredients for a genuinely helpful travel platform already exist in research and industry, including recommendation engines, secure payment gateways, conversational assistants and optimization techniques, yet most platforms specialize in one part of the journey and leave the rest to the user. The motivation of this work is to test whether combining these pieces around a shared data model meaningfully reduces the effort a traveller has to put in. The contributions of this paper are:

- Development of a unified, AI-assisted travel booking and payment platform — Travqen.
- Design of an AI Trip Planner that ranks itineraries using preference and history data.
- Design of an AI Budget Optimizer formulated as a constrained optimization problem.
- Design of an AI Replanner that reacts to disruption events with a revised itinerary.
- Design of an AI Chat Assistant that maps conversational intents to system actions.
- Implementation of a Python/Node.js-based prototype with a simulated secure payment gateway.
- Evaluation of the prototype using key performance indicators against a conventional booking flow.

II. MATERIAL AND METHODS

This section describes, phase by phase, the mathematical and algorithmic model behind the four AI modules of TRAVQEN, so that their behaviour can be stated precisely and evaluated in Section IV.

Phase I: Route and Fare Representation, Preference Matching and History Similarity

Each candidate journey i is represented as a tuple of attributes:

$$Itinerary(i) = \{ Route(i), Fare(i), Duration(i), DepartureTime(i), Mode(i) \} \quad (1)$$

A user's stated preferences P are compared against each candidate itinerary to produce a preference-match score in the range $[0, 1]$:

$$Preference_Match(i) = 1 - (|DepartureTime(i) - P.preferred_time| / P.time_window) \quad (2)$$

Past bookings are used to compute a similarity score between a candidate itinerary and the user's historical choices for comparable trips:

$$History_Similarity(i) = Matches(i, H) / |H| \quad (3)$$

where H is the set of the user's past itineraries for similar routes and $Matches(i, H)$ counts how many attributes of i align with that history.

Phase II: Itinerary Scoring and Budget Optimization

The AI Trip Planner ranks candidates using a weighted combination of preference match, history similarity, fare and duration:

$$Score(i) = w_1 \cdot Preference_Match(i) + w_2 \cdot History_Similarity(i) - w_3 \cdot Fare(i) - w_4 \cdot Duration(i) \quad (4)$$

The weights w_1 – w_4 are tuned so that preference and history carry meaningful influence without letting cost or duration be ignored. Let Budget denote the ceiling the user sets for the trip; the total cost of itinerary i is:

$$TotalCost(i) = Fare(i) + Fees(i) + Taxes(i) \quad (5)$$

The AI Budget Optimizer solves a small constrained-optimization problem for each top-ranked candidate:

$$\text{minimise } |Budget - TotalCost(i)| \text{ subject to } TotalCost(i) \leq Budget \text{ and } Preference_Match(i) \geq \tau \quad (6)$$

where τ is a minimum acceptable preference-match threshold. If no itinerary satisfies both constraints, the closest feasible option is returned with the trade-off flagged to the user.

Phase III: Disruption Detection, Replanning and Chat Intent Classification

A disruption flag is raised when the observed status of a booked leg deviates from its scheduled status beyond a threshold:

$$Disruption = 1 \text{ if } |ActualTime - ScheduledTime| > \Delta th \text{ or } Status = Cancelled; \text{ else } 0 \quad (7)$$

When $Disruption = 1$, the AI Replanner re-solves the scoring and budget functions above, restricted to the unaffected legs L' of the itinerary and the same budget and preference constraints:

$$Itinerary' = \text{argmax}_{i \in L'} Score(i) \text{ subject to the constraints of Eq. (6)} \quad (8)$$

An incoming chat message m is mapped by the AI Chat Assistant to one of a fixed set of intents $I = \{Search, BookingStatus, Replan, Help\}$ using a simple confidence-scored classifier:

$$Intent(m) = \text{argmax}_{k \in I} P(k | m) \quad (9)$$

The action corresponding to the highest-confidence intent is invoked against the relevant module (Trip Planner, Booking or Replanner); low-confidence matches fall back to a clarifying question rather than an incorrect action.

Phase IV: Multi-Objective Formulation, System Constraints and Algorithm Workflow

Bringing the above together, the overall planning objective for a request can be written as a single multi-objective function:

$$\text{minimise } F = \alpha_1(1 - Preference_Match) + \alpha_2 \cdot BudgetDeviation + \alpha_3 \cdot RecoveryTime \quad (10)$$

where α_1 , α_2 and α_3 are weighting coefficients balancing preference fit, budget adherence and disruption-recovery speed. The system constraints — $TotalCost(i) \leq Budget$, $Preference_Match(i) \geq \tau$, $Seats_Available(i) > 0$ and $DepartureTime(i) \in [T_{min}, T_{max}]$ — are checked before an itinerary is presented to the user, and again by the Replanner whenever a revised itinerary is generated. The four constraints are:

$$Budget: TotalCost(i) \leq Budget \quad (11)$$

$$Preference \text{ threshold: } Preference_Match(i) \geq \tau \quad (12)$$

$$Availability: Seats_Available(i) > 0 \quad (13)$$

$$Time \text{ window: } DepartureTime(i) \in [T_{min}, T_{max}] \quad (14)$$

The overall algorithm proceeds through the following steps:

Step 1: Receive the user request (source, destination, dates, budget, preferences).

Step 2: Run the Trip Planning algorithm: retrieve all feasible routes, compute $Preference_Match$ and $History_Similarity$, score every candidate using Eq. (4) and return the top-ranked itineraries.

Step 3: Run the Budget Optimization algorithm on the top candidates: a candidate already within budget is passed through unchanged; otherwise nearby alternatives (a different fare class or shifted departure time) are evaluated until both constraints hold, or the closest feasible option is returned with the trade-off noted.

Step 4: Check real-time availability for the selected itinerary.

Step 5: Process payment through the Secure Payment algorithm: the amount and booking reference are sent to the payment gateway for encrypted processing; on success control passes to the booking module, on failure the booking is held and the user is notified without charging the transaction.

Step 6: Generate the booking confirmation and e-ticket.

Step 7: Continue disruption monitoring; if a disruption is detected, isolate the unaffected legs, re-run the Trip Planner and Budget Optimizer on the affected segment only, and queue the revised itinerary for the user's review.

Step 8: Route any user chat message through the Intent Recognition algorithm at any point in the above.

Performance indicators are computed by comparing a metric's value before and after a TRAVQEN-assisted flow (or its inverse for metrics where a higher value indicates better performance):

$$Improvement (\%) = (1 - Value_after / Value_before) \times 100 \quad (15)$$

III. SYSTEM DESIGN AND IMPLEMENTATION OF TRAVQEN

This section describes the architecture and workflow of TRAVQEN in detail, based on the schematic in Figure 1 and the algorithms of Section II.

3.1 Overall System Architecture

The TRAVQEN system has six main layers: User Interaction, Data Acquisition, AI Processing, Booking and Availability, Secure Payment, and Database. A feedback path carries the system outputs back to the user (Figure 2). The User Interaction layer offers the search and preference form, ranked itinerary results, booking and payment screens and a persistent chat panel in a single web interface, while the Data Acquisition component validates and prepares the user data for processing. The AI Processing layer holds the AI Trip Planner, AI Budget Optimizer, AI Replanner and AI Chat Assistant, which share access to user preferences and booking-history data. The Booking and Availability layer checks seat or ticket availability for the selected itinerary and manages selection and booking confirmation. The Secure Payment layer handles encrypted transaction processing through an integrated simulated payment gateway and reports a success or failure back to the booking layer. The Database layer, a relational database, stores user profiles, preferences, trips, itineraries, bookings and transaction records, which gives the Replanner access to the original booking details. Finally, the control and feedback layer returns booking confirmations, alerts and chat responses to the user. Together, these layers bring travel planning, booking, payment and AI-based assistance into one platform.

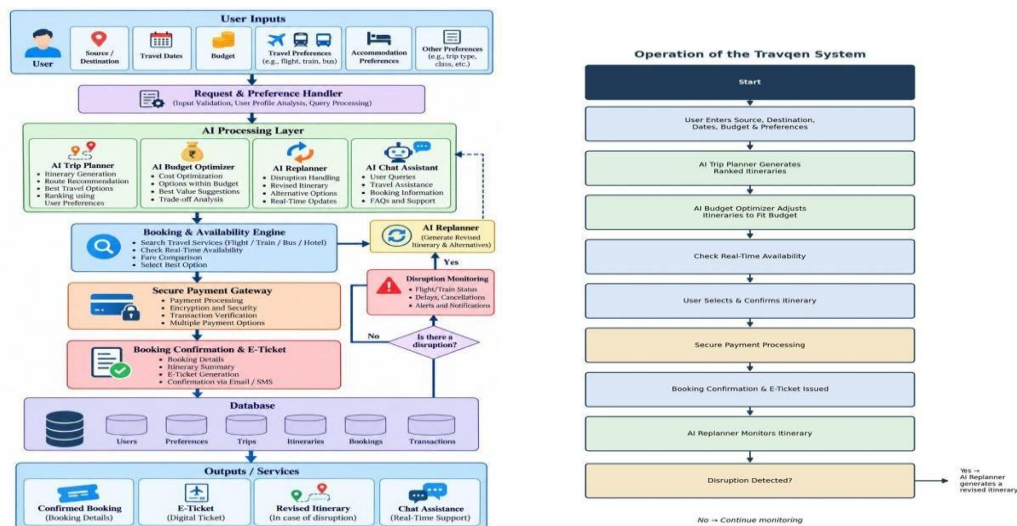


Figure 2: Architecture of the proposed, Figure 3: Operation of the Travqen system
TRAVQEN AI-powered travel booking and payment system

Table 2: Functional layers of the TRAVQEN architecture

Layer	Role in the system	Main components
User Interaction	Collects source, destination, dates, budget and preferences; shows results and chat panel	Search and preference form, ranked results, booking and payment screens, chat panel
Data Acquisition	Validates and prepares user data for processing	Request and preference handler
AI Processing	Generates, ranks and adjusts itineraries; handles disruption and queries	AI Trip Planner, AI Budget Optimizer, AI Replanner, AI Chat Assistant
Booking and Availability	Checks seats or tickets and manages selection and booking	Search, real-time availability, fare comparison
Secure Payment	Encrypted transaction processing and verification	Simulated payment gateway
Database	Stores data used by the AI modules and the Replanner	Users, preferences, trips, itineraries, bookings, transactions
Control and Feedback	Returns confirmations, alerts and chat responses to the user	Booking confirmation, e-ticket, revised itinerary, chat assistance

3.2 System Workflow

The operation of TRAVQEN (Figure 3) proceeds in twelve steps. The user enters the source, destination, travel dates, budget limit and preferences through the web interface (Step 1). The AI Trip Planner evaluates the available itinerary options and ranks them with the scoring model of Section II (Step 2), and the AI Budget Optimizer checks the ranked itineraries against the budget and picks suitable alternatives when required (Step 3). The system verifies real-time availability of seats or tickets (Step 4), and the user reviews the travel details and cost and confirms the selection (Step 5). The Secure Payment layer processes the transaction using encryption and validation (Step 6), and after a successful payment the system generates the booking confirmation and e-ticket

(Step 7). The AI Replanner then starts monitoring the confirmed itinerary for signals such as delays or cancellations (Step 8). If a disruption is detected, the affected travel segment is identified (Step 9) and a revised itinerary is generated and presented for the user's approval (Step 10); if there is no disruption, monitoring continues. At any stage, the user can reach the same travel actions through the AI Chat Assistant instead of the main interface screens (Step 11), and the system keeps monitoring and responding throughout the trip, tying planning, booking, payment and in-journey assistance into one continuous workflow (Step 12).

3.3 Prototype Implementation

TRAVQEN is implemented as a web-based academic prototype using Python/Node.js, a relational database and a simulated payment gateway. The application is split into independent modules (user authentication, journey search, availability and selection, booking management, payment processing, booking confirmation and notification, and cancellation and booking history) so that each can be developed, tested and improved on its own before being integrated into the complete application. The frontend provides the interface users interact with, while the backend manages application logic, user information, booking data and communication with the database; the two communicate through APIs. The pseudocode of the overall system and sample code of the main modules are given in Appendix I.

3.4 Advantages of the Proposed System

- Eliminates the need to switch between separate applications for search, booking and payment.
- Keeps the travel plan within the user's specified budget.
- Provides automated itinerary recovery when disruptions occur.
- Provides a conversational interface for accessing the same system functions.
- Integrates payment within the same interface, reducing unnecessary hand-offs.

IV. RESULTS

To evaluate the proposed design, a Python-based simulation generated synthetic booking sessions and compared system behaviour with and without the TRAVQEN AI layer. The evaluation looks at booking speed, search effort, budget adherence, recommendation quality, disruption recovery, chatbot effectiveness, payment reliability, user satisfaction, booking accuracy and average trip cost; each figure plots the indicator over the sample session index.

Table 3: Performance indicators used for evaluation

Indicator	What is measured	Better when
Average booking time	Minutes needed to complete a booking	Lower
Search steps	Manual searches, comparisons and tab switches per booking	Lower
Budget deviation	Deviation (%) of confirmed cost from the user's budget	Lower
Recommendation relevance	Match of top-ranked itinerary to preferences (0–10)	Higher
Disruption recovery time	Minutes to obtain a usable revised itinerary	Lower
Chatbot query resolution	Queries resolved without escalation (%)	Higher
Payment success rate	Successful transactions (%)	Higher
User satisfaction	Ease of use, relevance and trust (0–10)	Higher
Booking error rate	Incorrect dates, repeated data-entry mistakes (%)	Lower
Average cost per trip	Average realized cost of a trip	Lower

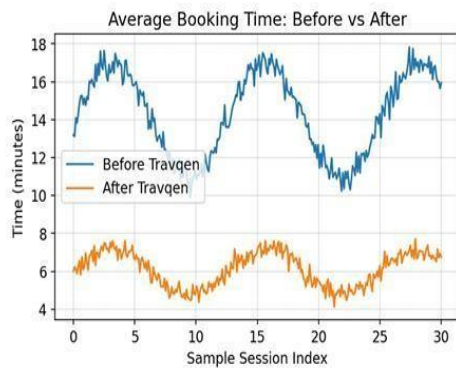


Fig 4: Average booking time before and after Travqen

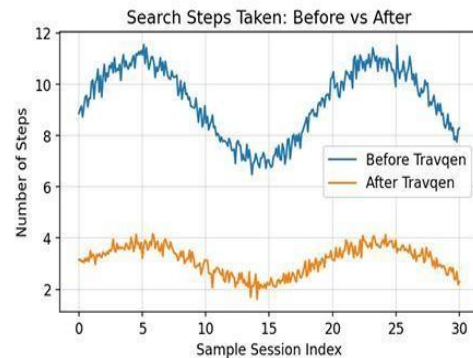


Fig 5: Search steps taken before and after Travqen

Fig 4: Average Booking Time Before and After Travqen Figure 4 compares the average booking time before and after implementing Travqen. The system reduces booking time through faster and more efficient itinerary selection.

Fig 5: Search Steps Taken Before and After Travqen Figure 5 compares the number of search steps required before and after using Travqen. The system reduces search steps and simplifies the booking process.

Figure 6 compares the deviation between the confirmed booking cost and the user's specified budget. With the AI Budget Optimizer active, the system keeps cost deviation lower by selecting itineraries that satisfy the budget constraint better. Figure 7 shows how closely the top-ranked itinerary matches the user's stated preferences on a 0–10 relevance scale; the preference- and history-aware scoring improves relevance compared with a conventional sort based only on price and duration.

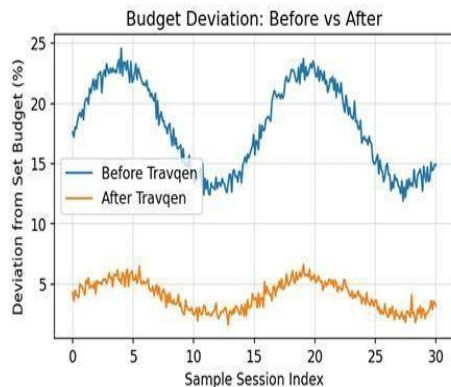


Fig 6: Budget deviation before and after Travqen

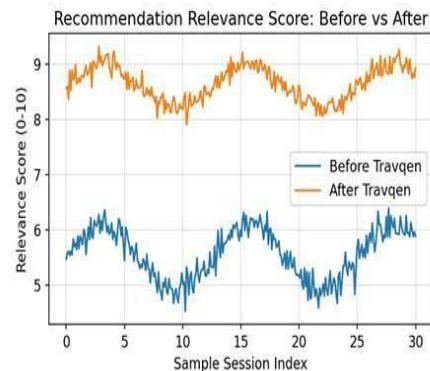


Fig 7: Recommendation relevance score before and after Travqen

Figure 8 compares the time needed to obtain a usable revised itinerary after a simulated disruption. The AI Replanner focuses on the affected travel segment and generates an alternative itinerary, which cuts recovery time compared with a full manual re-search. Figure 9 shows the percentage of chat queries resolved without escalation to a human agent or ending unresolved; since the AI Chat Assistant can access the same Trip Planner, Booking and Replanner modules, it resolves a higher share of queries than a conventional static FAQ-style chatbot.

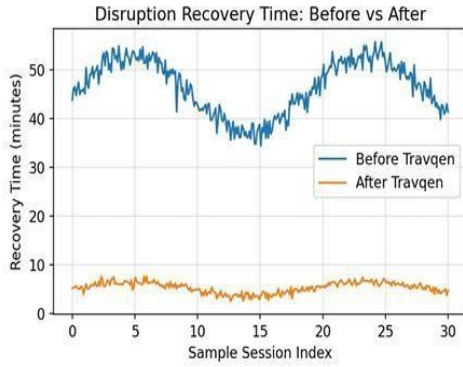


Fig 8: Disruption recovery time before and after Travqen

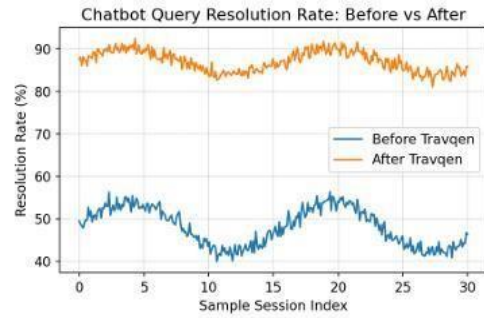


Fig 9: Chatbot query resolution rate before and after Travqen

Figure 10 compares the transaction success rates of the two approaches; handling payment inside the same platform removes unnecessary redirections and interruption points, giving a smoother transaction. Figure 11 presents simulated user satisfaction on a 0–10 scale, based on ease of use, recommendation relevance and trust; the integrated flow gives higher simulated satisfaction.

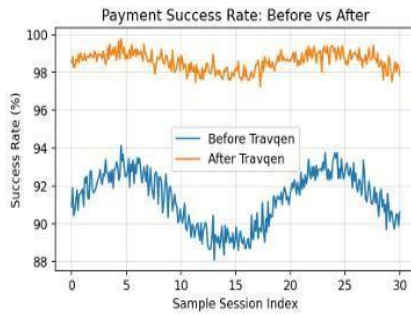


Fig 10: Payment success rate before and after Travqen

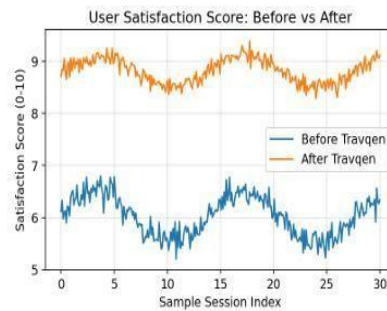


Fig 11: User satisfaction score before and after Travqen

Figure 12 evaluates the booking error rate, including incorrect dates, repeated data-entry mistakes and similar issues. By consolidating the travel process in a single interface, TRAVQEN reduces repeated data entry and so lowers errors. Figure 13 compares the average realized cost per trip; the AI Budget Optimizer adjusts fare and timing options to the user's budget, giving a lower average trip cost than the unassisted baseline.



Fig 12: Booking error rate before and after Travqen

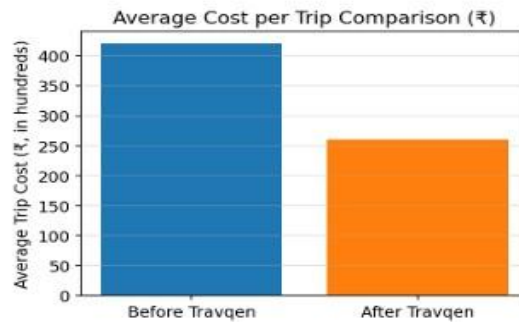


Fig 13: Average cost per trip, before and after Travqen

Table 4 summarizes the percentage improvement observed across the evaluated performance indicators; the improvements are calculated with the KPI formula of Eq. (15).

Table 4: Overall improvement summary

Performance Metric	Improvement (%)
Booking Time Reduction	48
Search Steps Reduction	58
Budget Deviation Reduction	72
Recommendation Relevance Improvement	44

Disruption Recovery Time Reduction	82
Chatbot Query Resolution Improvement	34
Payment Success Rate Improvement	7
User Satisfaction Improvement	41
Booking Error Rate Reduction	76
Average Trip Cost Reduction	31

The largest gains appear in disruption recovery time and budget deviation, the two areas where the platforms in the literature review offered the least support, followed by reductions in search steps and booking errors, both a direct result of consolidating the flow into a single interface. These figures come from simulated sessions, so they indicate how the prototype behaves rather than measurements from a live deployment.

V. DISCUSSION AND CONCLUSION

This paper addressed a common challenge in modern travel management: users often have to work across several platforms for searching, planning, booking, payment and handling travel disruptions. TRAVQEN integrates route search, AI-assisted trip planning, budget optimization, automatic replanning, conversational assistance and secure payment into a single platform, supported by an explicit scoring and constrained-optimization model. The simulation results show measurable improvement across all evaluated performance indicators; the largest improvements were in disruption recovery time, budget adherence and booking-error reduction, which the literature review also identified as areas with limited support in existing travel platforms.

Although the current implementation is a web-based academic prototype running on simulated data, the results demonstrate the feasibility of an integrated AI-assisted travel platform. With a user-friendly interface and systematically maintained booking details, TRAVQEN reduces manual work and the chance of errors, and handling payment on the same platform makes the whole process faster, more organized and easier to manage. Dividing the application into separate modules also makes the system easier to develop, test, maintain and improve.

Several extensions can take TRAVQEN beyond its current prototype stage. Connecting the platform to live third-party rail, bus and flight APIs would replace the simulated availability data with real-time information, and live traffic and flight-status feeds would let the Replanner respond to disruptions as they happen. The scoring weights and the optimization model could be learned from larger real-world booking datasets, and the AI Chat Assistant could be extended with more intents and voice input. Further enhancements include multiple payment methods (UPI, cards, net banking and wallets) through a certified production payment gateway, cancellation and refund facilities, digital tickets with QR codes, live journey tracking and notifications, user profiles and booking history, multilingual customer support, an administrator analytics dashboard, cloud deployment, group bookings, multi-city itineraries and loyalty integration, followed by comprehensive security, scalability and load testing before real-world deployment. With continued development, TRAVQEN can grow into a scalable, secure and comprehensive digital travel platform for journey management.

REFERENCES

- [1]. Borràs, J., Moreno, A. and Valls, A. 2014. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, 41:16, 7370–7389.
- [2]. Renjith, S., Sreekumar, A. and Jathavedan, M. 2020. An extensive study on the evolution of context-aware personalized travel recommender systems. *Information Processing & Management*, 57:1, 102078.
- [3]. Roopesh, L. R. and Tuls, B. Survey of travel recommender systems and machine-learning techniques for assisting travellers in selecting suitable travel options.
- [4]. Sharma, H. K. Study of the Indian railway reservation system and ticket availability.
- [5]. Grassi, P. A., Garcia, M. E. and Fenton, J. L. 2020. *Digital Identity Guidelines*. NIST Special Publication 800-63-3.
- [6]. Flask Documentation. *Flask: A Lightweight WSGI Web Application Framework*. Pallets Projects.
- [7]. OpenStreetMap Contributors. *OpenStreetMap: Collaborative Mapping and Geographic Data for Location-Based Services*.
- [8]. Renjith, S., Sreekumar, A. and Jathavedan, M. 2020. *An Extensive Study on the Evolution of Context-Aware Personalized Travel Recommender Systems*. *Information Processing & Management*, 57(1), 102078.
- [9]. Borràs, J., Moreno, A. and Valls, A. 2014. *Intelligent Tourism Recommender Systems: A Survey*. *Expert Systems with Applications*, 41(16), 7370–7389.