An AI-Driven Intelligent Travel Planning Platform for Personalized, Safe and Sustainable Tourism

Akshatha K S[1], Jeevan K GZ, Mithun G N3, Nishanth R N[4], Hanumantharaju B N[5]

[1]Assistant Professor, Department of ISE, East West Institute of Technology, Bengaluru, Karnataka, India
ZUG Scholar (1EW23IS039), Department of ISE, East West Institute of Technology, Bengaluru, Karnataka, India
3UG Scholar (1EW23IS056), Department of ISE, East West Institute of Technology, Bengaluru, Karnataka, India
[4]UG Scholar (1EW23IS061), Department of ISE, East West Institute of Technology, Bengaluru, Karnataka, India
[5]UG Scholar (1EW23IS031), Department of ISE, East West Institute of Technology, Bengaluru, Karnataka, India

***Abstract*— *Trip planning today typically requires several disconnected applications for navigation, accommodation, transport, flights, itinerary preparation, and emergency help. Users must collect and reconcile this information manually, which costs time and weakens coordination and safety during travel. This paper presents SMART-ROUTE, a mobile travel-planning platform that brings these services into one application. The user enters a source, destination, trip duration, budget, number of travelers, and travel type (Family, Friends, Couple, or Solo). The backend generates a day-wise itinerary and returns budget information, hotels, transport options, a transport recommendation, flight information, attractions, an agentic recommendation, and sustainability information. A map view shows the route between source and destination, and navigation is handed off to the device's Google Maps application. Emergency shortcuts for hospital, police, fuel, and mechanic assistance and a trip-aware AI travel assistant are included in the same interface. Registered users can save multiple trips and retrieve them later from MongoDB. The system uses React Native with Expo and Expo Router on the client, and Node.js and Express.js REST APIs with JWT authentication on the server. Data is stored in MongoDB Atlas through Mongoose. The work supports SDG 11 (Sustainable Cities and Communities) through consolidated planning and sustainability-oriented recommendations. This paper separates implemented functionality from proposed functionality. It does not report accuracy, efficiency, response-time, or user-adoption figures. The evaluation is functional and prototype-level, and larger user studies are identified as future work.*

***Keywords*—** Intelligent Travel Planning, Itinerary Generation, Generative AI, Agentic Recommendation, Smart Tourism, Location-Based Services, Emergency Assistance, React Native, Node.js, MongoDB, Sustainable Cities, SDG 11.

I. INTRODUCTION

Planning even a short trip involves many decisions. A traveler chooses a destination and duration, estimates a budget, compares transport and accommodation, decides which attractions fit the available days, checks the route, and considers what to do if something goes wrong on the way. In practice each of these decisions is served by a different application or website. One application handles maps and navigation, another handles bookings, a third gives weather, and itineraries are assembled by hand from search results. A user completing a single journey plan therefore switches between platforms repeatedly, which makes planning slow, confusing, and prone to poor decisions.

Three concerns make this fragmentation worse. First, budget and time constraints interact. A hotel choice affects how much remains for transport and activities, and the number of days limits how many attractions are feasible. These trade-offs are difficult to manage when information is scattered. Second, travel-type differences matter. A family, a group of friends, a couple, and a solo traveler generally need different pacing and different recommendations, so a generic listing serves none of them well. Third, unexpected situations such as a medical problem, a vehicle breakdown, running out of fuel, or a need for police assistance rarely have a quick entry point in planning tools. Emergency information is usually searched for only after the problem occurs.

Recent AI-based tourism work has addressed parts of this problem. Recommendation models improve prediction of what a traveler may like [1], AI guide services offer intelligent guidance [2], AI has been discussed as a transformative force in tourism [3], and generative AI has been combined with personality models to personalize recommendations [4]. An AI-driven smart tourism platform has also been proposed that integrates group recommendation, itinerary planning, and budgeting [5]. However, according to the limitations noted for these works, none of them combines itinerary generation with integrated navigation, travel services, and emergency support in a single system.

Travel planning also has a sustainability dimension. Sustainable Development Goal 11 (Sustainable Cities and Communities) concerns inclusive, safe, and sustainable urban and community development, and travel is closely tied to it. Better-informed trip planning can reduce unnecessary movement and duplicated effort, and recommendations that surface lower-impact transport and responsible tourism practices can support more sustainable travel behavior. This paper does not claim any measured environmental effect.

Problem addressed. Given a traveler's source, destination, duration, budget, group size, and travel type, how can a single mobile application generate a personalized itinerary, present the relevant transport, accommodation, flight, and attraction information alongside a route map, offer emergency access and an AI assistant within the same interface, and persist the plan for later retrieval per user?

The objectives of SMART-ROUTE are to:

1) develop a unified platform for travel planning;
2) generate a smart, personalized itinerary from user requirements;

3) provide route visualization with navigation hand-off;
4) provide transportation, accommodation, flight, and attraction information;
5) integrate emergency support for hospital, police, fuel, and mechanic assistance;
6) provide an AI travel assistant;
7) support secure authentication and per-user saved trips; and
8) incorporate sustainability information aligned with SDG 11.

The main contributions of this work are:

- a working mobile prototype (React Native/Expo) backed by a Node.js/Express REST API and MongoDB Atlas that combines itinerary generation, travel services, maps, emergency shortcuts, an AI assistant, and saved trips in a single workflow;
- a JWT-authenticated, user-specific saved-trip mechanism that stores the complete generated plan and allows multiple trips per user; and
- an explicit separation between implemented and proposed functionality, together with an evaluation that reports only what the project evidence supports.

These contributions are not presented as unprecedented. Each individual service exists in other applications. The contribution lies in integrating them, at prototype level, within one implemented system. The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the proposed system. Section IV presents the architecture. Section V describes the methodology. Section VI details the implementation. Section VII presents the functional evaluation. Section VIII discusses results. Section IX outlines limitations, Section X future scope, and Section XI concludes.

II. RELATED WORK / LITERATURE SURVEY

This section reviews work on (1) tourism recommendation models, (2) AI tour-guide services, (3) AI in tourism, (4) generative AI for personalized travel, and (5) integrated AI tourism platforms. The survey is limited to the sources compiled during the project.

Karras *et al.* [1] study tourism recommendation models. Their merit is improved prediction accuracy of recommendations. They do not support complete trip planning, since budget, schedule, and route handling are outside their scope. SMART-ROUTE addresses this by combining recommendations with budget-aware, day-wise itinerary generation.

Wang *et al.* [2] discuss the outlook and direction of AI tour-guide services from a lifelong machine learning view. The work offers intelligent recommendation and guidance and points to future directions. Its limitation is that budgeting and full trip automation are only weakly integrated. SMART-ROUTE includes budget information and generates a complete plan from user inputs.

Manoharan *et al.* [3] present "Voyage Virtuoso" in a book chapter on how artificial intelligence transforms tourism. The chapter's merit is showing how AI can enhance the tourist experience. It does not provide integrated real-time navigation or emergency support. SMART-ROUTE places a map view with navigation hand-off and emergency shortcuts in the same dashboard as the itinerary.

Aribas and Daglarli [4] integrate generative AI with personality models to personalize travel recommendations. Their strength is personalization. The work centers on recommendations and does not cover complete travel services such as navigation and emergency support. SMART-ROUTE uses generative AI for itinerary and assistant functions while also providing the surrounding travel services. It does not implement personality modeling.

Srikanthan *et al.* [5] propose an AI-driven smart tourism platform for personalized, safe, and sustainable travel planning, integrating group recommendation, itinerary planning, and budgeting. This is the closest related work. As noted in the project review, it does not integrate real-time navigation, weather updates, and emergency services into the planning system. SMART-ROUTE targets this gap, with the caveat that its weather and real-time features are only partially implemented (Section III).

TABLE I COMPARISON OF RELATED TRAVEL PLANNING SYSTEMS

Ref.	Year	Approach	Task / Scope	Key Finding / Merit	Limitation Relative to SMART-ROUTE
[1]	2024	Tourism recommendation models	Tourist recommendation	Improved prediction accuracy	No complete trip planning (budget, schedule, routes)
[2]	2021	AI tour-guide services (lifelong ML view)	Intelligent guidance and recommendation	Intelligent recommendations and guidance	Limited budgeting and full-trip automation
[3]	2024	AI in tourism (book chapter)	Enhancing the tourism experience with AI	AI improves the tourism experience	No integrated real-time navigation or emergency support
[4]	2024	Generative AI + personality models	Personalized travel recommendation	Enhanced personalization	Recommendation-focused; no navigation or emergency services
[5]	2025	AI-driven smart tourism platform	Group recommendation, itinerary, budgeting	Integrates recommendation, itinerary, and budgeting	No integrated real-time navigation, weather, or emergency services
Proposed	2026	React Native + Node/Express + MongoDB; generative AI itinerary; agentic recommendation	Itinerary, hotels, transport, flights, attractions, sustainability, map, emergency shortcuts, AI assistant, saved trips	Unified mobile workflow with per-user persistence	Prototype scale; no quantitative or field evaluation; some services are link-outs or proposed

Table I summarizes these works alongside the proposed system. The literature shows work on recommendation quality, personalization, and platform integration, but the reviewed works do not combine itinerary generation with navigation and emergency support. This is the integration gap addressed here. SMART-ROUTE's scope is more modest than research systems in one respect: it has not been quantitatively benchmarked against them.

III. PROPOSED SYSTEM

A. System Overview

SMART-ROUTE is a mobile application that replaces separate planning, mapping, transport, and emergency tools with one workflow. The user authenticates, enters trip details, and receives a dashboard containing the itinerary and related services. Table II summarizes the modules and their status.

TABLE II SMART-ROUTE FUNCTIONAL MODULES AND STATUS

Module	Description	Status
Registration / Login	Email-based account creation and login; JWT issued	Implemented
Persistent session	Token stored in AsyncStorage	Implemented
Travel input	Source, destination, days, budget, travelers, travel type	Implemented
AI itinerary generation	Day-wise plan via /plan	Implemented
Budget information	Returned with the plan	Implemented
Hotels	/hotels	Implemented
Transport and recommendation	/transport, /transport-recommendation	Implemented
Flights	/flights; "Search Live Flights" opens Google Flights	Implemented (link-out)
Attractions	/attractions	Implemented
Agentic recommendation	/agent/recommend	Implemented (internal logic [INFORMATION NOT PROVIDED IN PROJECT FILES])
Sustainability information	/sustainability	Implemented
Map and route view	Source/destination markers and route line; navigation opens Google Maps	Implemented
AI travel assistant	/ai-assistant with quick prompts	Implemented
Emergency services	Hospital, police, fuel, mechanic endpoints	Implemented
Saved trips	Multiple, user-specific, full itinerary retrieval	Implemented
Destination search	/destinations, dashboard search	Implemented
Weather forecasting	Real-time weather updates	Proposed / to be verified
Optimized routing algorithm	Algorithmic route optimization	Proposed
Live tracking, digital transactions	Described in the project review	Proposed

B. Implemented Features

1) **Authentication.** Users register and log in. The backend issues a JWT, which the app stores in AsyncStorage so the session persists. Protected endpoints require an Authorization: Bearer header.
2) **Travel input and itinerary generation.** The form collects source, destination, days, budget, travelers, and travel type (Family, Friends, Couple, Solo). The /plan API returns the itinerary, budget, hotels, transport, flights, attractions, agentic recommendations, and sustainability information.
3) **Unified dashboard.** The dashboard presents the route map, day-wise itinerary, and service sections in one screen, with a search box for flights, hotels, and places.
4) **Emergency assistance.** Four shortcuts (hospital, police, fuel, mechanic) are available from the dashboard.
5) **AI travel assistant.** Users ask destination, itinerary, hotel, flight, or transport questions, with quick prompts such as "Day 2?" and "Transport?".
6) **Saved trips.** The plan is stored per user in MongoDB and can be retrieved in full. Multiple trips are supported, and "Plan Another Trip" restarts the flow.

C. Proposed / Future Features

Real-time weather integration, algorithmic route optimization, live tracking, secure in-app transactions, and large-scale deployment are described in the project review as goals but are not evidenced as completed. They are treated as proposed work (Section X).

IV. SYSTEM ARCHITECTURE

Fig. 1 shows the architecture adopted from the project design. The flow is: User → Mobile App → Backend Server → AI & Travel Services → Database.

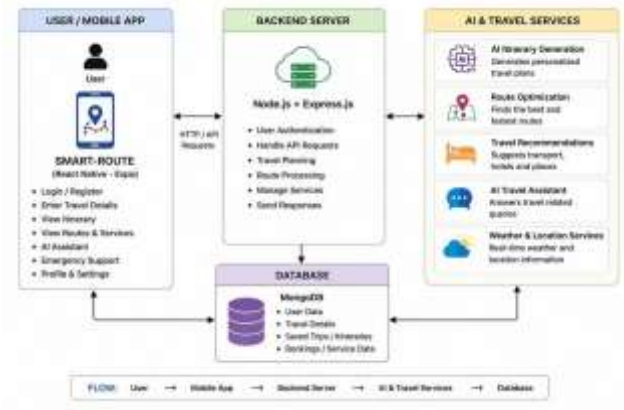


Fig. 1. Overall architecture of SMART-ROUTE.

Mobile application layer. The React Native (Expo) client provides login/register, travel-detail entry, itinerary viewing, routes and services, the AI assistant, emergency support, and profile/settings. The client communicates with the backend over HTTP.

Backend layer. A Node.js/Express.js server handles user authentication, API requests, travel planning, route processing, service management, and response delivery. The server runs on port 5000, and a /health endpoint reports application name, database status, and version. At the time of writing the health response reported the server running, MongoDB connected, and version 5.0.0.

AI and travel services layer. This layer comprises AI itinerary generation, route optimization, travel recommendations (transport, hotels, places), the AI travel assistant, and weather/location services, as depicted in the architecture diagram. Table II marks which of these are implemented.

Database layer. MongoDB Atlas, accessed through Mongoose, stores user data, travel details, and saved trips. A SavedTrip document contains approximately: userId, source, destination, days, budget, travelers, travelType, plan, createdAt, and updatedAt.

Authentication flow. On login, the server issues a JWT that the client stores in AsyncStorage and attaches to protected requests such as saved-trip operations. Saved trips are therefore scoped to the authenticated user.

Client-side data handoff. The complete /plan response is large. Passing it as a navigation parameter was replaced by writing it to AsyncStorage under the key currentGeneratedTrip and navigating with only small parameters (source, destination, days, budget, travelers, travelType). The dashboard reads the full plan from AsyncStorage.

V. METHODOLOGY

The methodology follows the numbered stages of the implemented workflow.

1) **Registration and login.** A user creates an account and logs in through /auth/register and /auth/login.
2) **Authentication.** A signed JWT is returned and stored on the device. Protected routes require it in the Authorization header. Password-storage details are [INFORMATION NOT PROVIDED IN PROJECT FILES].
3) **Requirement collection.** The user enters source, destination, days, budget, travelers, and travel type.
4) **Input validation.** Inputs are checked before the request is issued. The specific validation rules are [INFORMATION NOT PROVIDED IN PROJECT FILES].
5) **AI itinerary generation.** The inputs are sent to /plan, which returns a day-wise itinerary. Generative AI is used for itinerary generation and travel assistance. The specific model or API is [INFORMATION NOT PROVIDED IN PROJECT FILES]; the paper therefore makes no claim about model architecture, prompt design, or output accuracy.
6) **Service retrieval.** The plan response includes hotels, transport, flights, and attractions. The dashboard also queries the corresponding endpoints to display each section. Hotel responses are handled defensively so that either an array or an object containing a hotels array is accepted.
7) **Route and location processing.** Coordinates for the source and destination are loaded and shown on a map as markers joined by a route line. Navigation is delegated to Google Maps on the device.
8) **Personalized and agentic recommendations.** The travel type and budget accompany the request, and /agent/recommend and /transport-recommendation provide recommendation output. The agentic component is described only as an intelligent recommendation function whose internal logic is not documented in the project files.
9) **Sustainability recommendations.** /sustainability supplies sustainability information for the trip, tied to the SDG 11 goal.
10) **Emergency integration.** Hospital, police, fuel, and mechanic shortcuts are exposed on the dashboard and backed by dedicated endpoints.
11) **Trip presentation.** The dashboard renders the route, itinerary, budget, hotels, attractions, transport, sustainability information, emergency section, and AI assistant.
12) **Save trip.** With a valid token, the plan and trip parameters are stored as a SavedTrip linked to userId.
13) **Retrieval.** /saved-trips returns the user's stored trips, and a selected trip restores its complete itinerary.

VI. SYSTEM IMPLEMENTATION

Frontend. React Native with Expo SDK 54 (expo ~54.0.36, react 19.1.0, react-native 0.81.5), Expo Router ~6.0.24, TypeScript/TSX, react-native-maps 1.20.1, and AsyncStorage 2.2.0. Supporting libraries include react-native-gesture-handler, react-native-reanimated, react-native-safe-area-context, and react-native-screens. The dashboard is a single large TSX screen that loads the generated trip, coordinates, and service data.

Backend. Node.js with Express.js exposing REST endpoints: /auth/register, /auth/login, /plan, /hotels, /transport, /transport-recommendation, /agent/recommend, /attractions, /flights, /sustainability, /ai-assistant, /hospital, /police, /fuel, /mechanic, /saved-trips, /destinations, and /health.

Database. MongoDB Atlas with Mongoose, as described in Section IV.

Tools. VS Code, Postman, PowerShell/CMD, Expo, and EAS Build for producing an Android APK.

Interface. Figs. 2–7 are screenshots of the implemented interface.

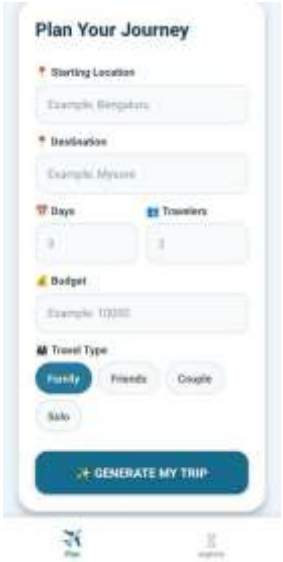


Fig. 2. User login interface.

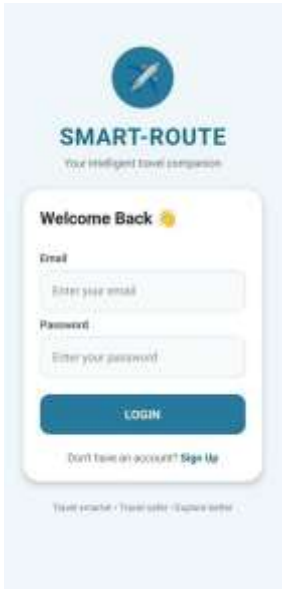


Fig. 3. Travel planning input interface.



Fig. 6. AI-generated three-day itinerary for Mysuru.



Fig. 7. Emergency assistance and SMART-ROUTE AI assistant.



Fig. 4. Trip dashboard with summary, search, and flight-details card.

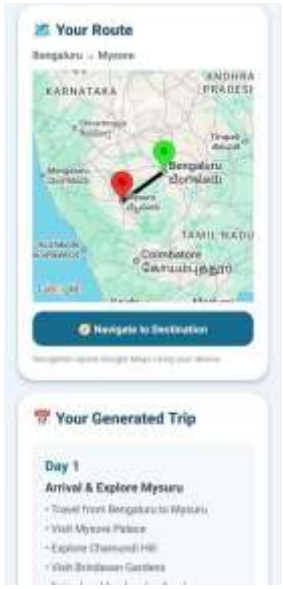


Fig. 5. Route map (Bengaluru → Mysuru) with navigation button.

The screenshots show a Bengaluru → Mysuru, 3-day, 2-traveler plan. Day 1 covers arrival, Mysore Palace, Chamundi Hill, Brindavan Gardens, and local food. Days 2 and 3 include Chamundi Hill, Brindavan Gardens, Mysore Zoo, and Devaraja Market. The flights card states that flight search opens Google Flights for the selected route, so flight data is a link-out rather than an embedded live-fare integration.

Interface observations. The implemented interface provides a single workflow from travel-detail entry to the generated trip dashboard. The login and planning screens support account access and collection of the source, destination, trip duration, traveler count, budget, and travel type. The dashboard presents the generated trip together with route information, flight search, emergency assistance, and the SMART-ROUTE AI assistant. The route screen displays the Bengaluru–Mysuru journey with source and destination markers and provides a navigation hand-off. The generated itinerary is presented day-wise, while the emergency screen provides access points for hospital, police, fuel, and mechanic assistance. These screenshots document the interface-level implementation described in this section; they are not used as quantitative evidence of recommendation accuracy, response time, or user adoption.

VII. FUNCTIONAL / EXPERIMENTAL EVALUATION

This is a student prototype without a multi-user deployment, so no accuracy, efficiency, response-time, or adoption figures are reported. Table III lists functional checks and the evidence available in the project record. "Evidence pending" means the feature is reported as implemented but a test record or screenshot has not been supplied.

TABLE III FUNCTIONAL EVALUATION OF SMART-ROUTE

Test Case	Input / Action	Expected Result	Observed Result	Status
Backend health	GET /health	Server, database, version reported	Reported running, MongoDB connected, v5.0.0; also reachable from phone browser on the same Wi-Fi	Pass
MongoDB connection	Server start	Atlas connection established	Connected	Pass
Itinerary generation (API)	POST /plan (Bengaluru→Mysore, 3 days, ₹10,000, 2 travelers, Friends) via curl	Complete plan returned	Returned itinerary, budget, hotels, transport, flights, attractions, agentic recommendations, sustainability info	Pass
Login screen and planning form	Open app	Screens render	Screenshots in Figs. 2–3	Pass
Dashboard display	Generate trip	Route map, itinerary, flights card, emergency and AI sections	Screenshots in Figs. 4–7	Pass (screenshots)
Login / server communication in APK	Log in from installed APK	Successful authentication	Reported working	Pass (reported)
Dashboard in standalone Android APK	Press Generate Trip in APK	Dashboard opens	App closes after /plan succeeds; cause under investigation	Fail / unresolved
Registration, JWT persistence, logout	Register, restart, log out	Account created, session persists	[Evidence pending]	Pending
Hotels, transport, attractions, sustainability	Dashboard requests	Data returned	[Evidence pending per endpoint]	Pending
AI assistant	Ask a question	Response returned	Interface shown in Fig. 7; response transcript [Evidence pending]	Pending
Emergency services	Tap Hospital / Police / Fuel / Mechanic	Relevant result	Interface shown in Fig. 7; behavior [Evidence pending]	Pending
Save Trip / Retrieve Saved Trip	Save, reopen	Stored per user in MongoDB and restored	[Evidence pending]	Pending

The single failing row matters. During APK testing the app closed when opening the generated dashboard although the backend completed the request without error. The suspected area is the native Android map configuration, but this has not been confirmed by device logs. Results for the dashboard therefore rest on the screenshots in Figs. 4–7.

VIII. RESULTS AND DISCUSSION

Unified planning. Figs. 3–7 show that one application can take a trip request and present the itinerary, route map, flight entry point, emergency services, and an AI assistant in one flow, without moving to other apps during planning.

Personalized itinerary generation. The /plan API returned a complete plan for the example request, and Fig. 6 shows a coherent day-wise itinerary for Mysuru. The paper does not quantify itinerary quality, and the generated days repeat some attractions (e.g., Chamundi Hill and Brindavan Gardens on more than one day), which suggests that plan quality would benefit from evaluation.

Location-based functionality. The map view (Fig. 5) shows the source and destination with a route line and opens Google Maps for navigation. This provides route visualization and navigation hand-off, not an in-app optimization algorithm.

Emergency support and AI assistance. Fig. 7 shows the emergency shortcuts and the assistant integrated into the trip screen, which addresses the emergency-access gap noted in Section II. Their usefulness has not been evaluated.

Persistence. The architecture stores saved trips per user in MongoDB with JWT protection. Persistence tests should be attached before this is claimed as verified (Table III).

Sustainability. The system surfaces sustainability information and transport recommendations in support of SDG 11. No environmental impact has been measured.

Scope of claims. These results show functional integration at the prototype level. They do not establish gains in planning time, recommendation accuracy, or safety, which would require a user study.

IX. LIMITATIONS

- **Android build stability:** the standalone APK closes when opening the generated dashboard, and the root cause is not yet confirmed.
- **Prototype-level evaluation:** testing was limited to development use, with no independent users, no quantitative benchmarks, and no automated test suite in the provided material.
- **Dependence on external services:** maps, flights, and navigation rely on Google services, and flight search opens Google Flights rather than returning embedded live fares.
- **Local-network backend:** development testing used a local Wi-Fi address, so the APK depends on that address unless the backend is hosted publicly.

- **Undocumented AI internals:** the AI model, prompting, and agentic logic are not specified, so their behavior cannot yet be characterized or validated.
- **Itinerary quality:** repeated attractions across days (Fig. 6) indicate that de-duplication and pacing are not guaranteed.
- **Feature scope:** weather, algorithmic route optimization, live tracking, and digital transactions are not evidenced as implemented.
- **Destination coverage:** implemented data is focused on Indian destinations, whereas the project aim includes international travel.

X. FUTURE SCOPE

1) Diagnose and resolve the Android runtime issue using device logs, then validate the full feature set on the APK.
2) Host the backend publicly so the app does not depend on a local network address.
3) Integrate real-time weather, traffic, and public-transport data.
4) Add algorithmic route optimization for multi-stop itineraries.
5) Integrate live flight and hotel data through booking or fare APIs.
6) Improve itinerary generation with de-duplication, pacing, and budget-constrained scheduling.
7) Develop a more rigorous sustainability score for transport and activities.
8) Add voice interaction, multilingual support, and offline access.
9) Support real-time emergency coordination and location sharing.
10) Conduct a structured usability study and report metrics only after real data is collected.

Each item is proposed work and is not part of the implemented system described in Sections III–VIII.

XI. CONCLUSION

This paper presented SMART-ROUTE, a mobile travel-planning platform that addresses the fragmentation of travel services. It combines AI-assisted itinerary generation, travel-service information, a route map with navigation hand-off, emergency shortcuts, an AI travel assistant, and per-user saved trips within one application. The system is built on React Native/Expo, Node.js/Express REST APIs with JWT authentication, and MongoDB Atlas. The evidence shows a working backend and a functioning trip dashboard in the interface, and it aligns the system with SDG 11 through consolidated planning and sustainability information. The paper has been explicit that the evaluation is functional rather than quantitative, that AI details are undocumented, and that the standalone Android build had an unresolved dashboard crash at the time of writing. Future work includes real-time data, route optimization, and a user study, which are needed before any claims about efficiency, safety, or sustainability impact can be made.

REFERENCES

[1] G. Karras *et al.*, "Tourism Recommendation Models," 2024. **[Incomplete: venue, volume, pages, DOI needed. Verify before submission.]**

[2] S. Wang, X. Hong, N. Karia, and S.-U. Guan, "Outlook and Direction of AI Tour Guide Services—From the Lifelong Machine Learning View," in *Proc. 2nd Int. Conf. Electronics, Communications and Information Technology (CECIT)*, 2021, pp. 802–806, doi: 10.1109/CECIT53797.2021.00145.

[3] G. Manoharan, C. V. G. Rao, S. P. Ashtikar, S. Kumar, and M. Nivedha, "Voyage Virtuoso: Artificial Intelligence in Transforming Tourism," in *Utilizing Smart Technology and AI in Hybrid Tourism and Hospitality*, 2024, pp. 79–97, doi: 10.4018/979-8-3693-1978-9.ch004.

[4] E. Aribas and E. Daglarli, "Transforming Personalized Travel Recommendations: Integrating Generative AI with Personality Models," *Electronics*, vol. 13, no. 23, Art. no. 4751, 2024, doi: 10.3390/electronics13234751.

[5] S. Srikanthan, T. Jayalath, C. Senevirathne, K. Rajendran, and T. Rasarathnam, "AI Driven Smart Tourism Platform for Personalized Safe and Sustainable Travel Planning," in *Proc. IEEE ICITR*, 2025. **[Pages/DOI needed.]**