Sherpa: A Safety-Driven Navigation Platform Using Real-Time Crime Analytics and Community Reporting

Meghana Pradeep, Keerthan M, Akshatha Prasad, Manith Gowda KR, Prof. Yashaswini C D
Computer Science And Engineering(Data Science), AMC Engineering College
Bangalore, India
{meghana020404, kituk2004, pakshatha20, manithgowda216}@gmail.com,
yashaswini.dhanapal@amceducation.in

ABSTRACT

With the world becoming more uncertain, safety while traveling is a paramount but underappreciated issue. Conventional navigation devices focus on speed and distance, ignoring the equally important factor of route safety. Sherpa fills the void by providing a safety-first intelligent navigation platform that assists users in selecting the safest route between two points. Using real-time crime data, anonymous reports from the community, and smart analytics, Sherpa displays high-risk areas in real-time using dynamic heatmaps and allocates safety scores to alternative routes. Sherpa uses Google Maps API, Firebase, and Chart.js to provide a smooth, data-driven experience that blends routing, reporting incidents, and safety analysis into a single, cohesive interface. With capabilities such as anonymous geo-tagged reporting, analysis of area-wise crime trends, and a simple interface, Sherpa enables individuals and communities to make safety-aware travel choices. This paper describes Sherpa's design, architecture, implementation, and social impact, showcasing the potential of Sherpa as a scalable solution for safety-oriented navigation in urban settings.

KEYWORDS

Safe Navigation, Crime Heatmaps, Real-Time Crime Data, Anonymous Incident Reporting, Community Safety, Smart Routing, Firebase, Google Maps API, Crime Analytics, Location Intelligence, Geo-Tagged Reports, Chart.js, Urban Mobility, Safety Score, Data-Driven Navigation.

1. INTRODUCTION

In today's fast-paced and increasingly unpredictable world, personal safety has become a growing concern—especially while navigating unfamiliar urban environments. With rising instances of crime and the availability of unmonitored routes, ensuring safe travel is no longer a luxury but a necessity. Although contemporary navigation platforms such as Google Maps and Apple Maps have transformed how we move by providing quick and effective routes, they concentrate on parameters that include distance and arrival time estimation only. The platforms ignore an important aspect of mobility: road safety.

Conventional navigation systems fail to consider the security conditions of a route. Most of them lead users along shortcuts or isolated routes that could have elevated crime rates or lack adequate surveillance, thus subjecting users to possible danger. The deficiency becomes particularly alarming for disadvantaged groups like women, children, and night travelers or explorers of unfamiliar terrain.

Sherpa was designed to help solve this urgent problem by redefining navigation in light of personal safety. Sherpa is a sophisticated, community-based web app that balances safety above speed in recommending routes to travel. By combining real-time crime feeds, user-submitted incident reports, and smart analytics, Sherpa offers users safety scores for routes and assists them in steering clear of unsafe areas.

As opposed to traditional platforms, Sherpa includes crowdsourced incident reporting, allowing users to report anonymously suspicious or criminal behavior. These reports, along with crime history, fuel dynamic heatmaps that visually determine crime hotspots on an interactive map. The system then estimates numerous route options and assigns each a safety score based on distance to recent incidents and general risk scores.

In addition, Sherpa includes a specialized analytics dashboard, developed using Chart.js, displaying safety trends by regions and time intervals. It not only equips individual users but also communities and authorities with insights into city safety dynamics. Based on technologies like Google Maps API, Firebase, and Material UI, Sherpa provides an intuitive, real-time, and secure platform for users to effectively decide on their travel routes.

By shifting passive navigation into an active, safety-conscious experience, Sherpa reimagines how technology may be utilized to save lives and support more intelligent urban mobility.

2. RELATED WORK

Over the last few years, numerous travel and navigation apps have come to the fore to help users with route planning, travel itinerary organization, and journey safety. Google Trips, Roadtrippers, and TripIt are some of the most prominent among them, each featuring differentiating capabilities for improved travel.

Google Trips, for example, organizes users' travel details by importing information automatically from Gmail, and also provides sightseeing tips and offline viewing of journey details. But the application does not feature a real-time safety analysis focus and route protection, which is essential in fast-paced urban settings. Likewise, Roadtrippers is designed around road trip travel planning by aiding users in exploring attractions and roadways. Again, it does not focus on leisure and adventure but rather doesn't include incident reporting or crime analysis. TripIt is a robust travel management tool by which travelers can aggregate bookings and travel arrangements but neither uses any type of route awareness nor personalized risk assessment while traveling.

More generally, commercial navigation systems like Google Maps and Apple Maps give solid and optimized routes by distance and traffic. They neglect individual safety, particularly in crime hotspots or areas of recent incidents. Without functionality such as real-time community reporting, crime heatmaps, or AI-derived safety scoring, these systems cannot effectively cater to users who care about security while in transit.

Studies in this field have grown to focus more on the position of AI and community data in ensuring travel safety. For instance, Harish et al. [1] investigated the application of AI to risk-aware mobility planning, proposing that real-time data overlays can both improve commuter trust and route quality. In the same vein, Zhang and Liu's [2] study of crowdsourced public security systems reinforces the concept of incorporating anonymous reporting into city navigation apps to develop smarter cities. Kapoor and Sharma [3] also envisioned a geospatial analytics framework for anticipating unsafe areas, further underscoring the importance of integrating crime data with geographic visualization tools.

In spite of these developments, few real-world systems integrate real-time crime information, user-generated incident reports, and smart route recommendation into a unified, user-friendly system. This lack of current solutions inspired the creation of Sherpa, a system that not only provides intelligent navigation but also focuses on community safety, route openness, and crowd-sourced intelligence.

3. METHODOLOGY / SYSTEM DESIGN

The Sherpa development was directed by a scalable, modular architecture to deliver real-time safety-oriented navigation with an intuitive user experience. The system combines various technologies such as Firebase for backend, Google Maps API for geospatial rendering, Gemini AI model for smart recommendations, and React/Flutter for high-responsiveness front-end. The interface focuses on modularity, user interaction, and handling of real-time data to deliver a seamless, context-aware safety navigation experience.

3.1 Overall System Architecture

The Sherpa system is developed on a layered architecture with the following primary modules:

1. User Interface (UI):

Implemented using React for web and Flutter for mobile devices (optional extension), this layer enables users to interact with the system via a clean, intuitive, and responsive user interface. The interface enables route selection, viewing of crime heatmaps, reporting of incidents, and visualization of safety scores.

2. Location & Mapping Services:

Central to navigation is the Google Maps API, which enables real-time geolocation, routing, and map interaction. It manages geospatial representation of crime data and facilitates features including current location tracking, alternate route calculation, and zone of safety detection.

3.Crime Data & Community Reporting Module:

This module processes:

- ❖ Public crime data (from open data sources/APIs)
- ❖ User-reported incident reports, which are geo-tagged and anonymously categorized
- ❖ The information is saved in Google Firestore for real-time updates and scale-up storage.

4. Safety Scoring Engine:

Every path calculated by the system is checked for safety by an internal Recommendation Engine. The engine gives a Safety Score to each potential path based on:

- ❖ Proximity to recent accidents
- ❖ Time of travel
- ❖ Intensity of crime in the area (through heatmaps)

The score is generated through a weighted model that ranks distance, risk thresholds, and user habits.

5. Recommendation Module with AI (Gemini API):

The Gemini AI model is invoked through API calls to create contextual, conversational route recommendations based on:

- ❖ User context (e.g., "Is it safe to travel through MG Road at 10 PM?")
- ❖ Crime patterns in the region
- ❖ Natural language input to enable intelligent interactions

This functionality allows for an AI-driven Chat Assistant in real-time that not only provides safe route suggestions but also responds to safety-related questions.

6. Analytics & Visualization Dashboard:

With Chart.js, Sherpa offers interactive graphs that display:

- ❖ Crime patterns through time
- ❖ Comparisons of areas by safety
- ❖ Filtering by location, time frame, and incident type

Both end-users and authorities are aided by this dashboard in determining patterns and making decisions.

7. Backend & Cloud Functions:

All of the key operations are powered by Firebase Cloud Functions, facilitating:

- ❖ Secure data handling
- ❖ Incident logging
- ❖ Syncing of maps and dashboards in real-time
- ❖ User authentication through Firebase Auth

3.2 Data Flow and Process Flow

1. User logs in and enters the main dashboard.
2. Chooses source and destination — Google Maps comes up with potential routes.
3. System loads latest crime reports and displays them on the map as heatmaps and markers.
4. Each route is rated by the Safety Scoring Engine and ranked.
5. User can ask Gemini AI for route recommendations or safety advice.
6. Last safe route is shown and selected.
7. Incident reports can be anonymously submitted by users, and these are updated in real-time across all users.

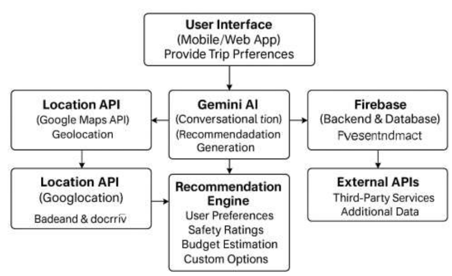


Fig. 1. Sherpa System Architecture

3.3 Used Technologies

Component	Technology
Frontend	React.js, Material UI
Mobile Frontend	Flutter
Mapping	Google Maps API
Backend & Database	Firebase (Auth, Firestore, Hosting, Cloud Functions)
AI Chat Assistant	Gemini API (via REST)
Charts	Chart.js
Hosting	Firebase Hosting

This modular and tech-intensive architecture not only makes Sherpa scalable and easy to use but also ideally suited to solve real-world safety issues in navigation by bringing data, AI, and the community together.

4. Implementation

Implementation of Sherpa took place during the GDG Solutions Hackathon 2025 with an aim to provide a completely working prototype that combines multiple services to offer personalized, secure, and real-time travel aid. Modular architecture and scalability were considered while developing the system. This section discusses the tools utilized, recommendation logic, integration of the Gemini API, and the application interface.

4.1 Tools and Libraries Used

Technology stack employed to develop Sherpa was chosen on the basis of development speed, integration with the cloud, and deployment simplicity:

- ❖ **Frontend:** The user interface has been designed on the web platform using React.js. The component-based framework of React facilitated the creation of reusable UI modules for maps, chat interactions, dashboards, and forms. For a mobile-first strategy, Flutter can be utilized in future releases to target both Android and iOS platforms from a single codebase.
- ❖ **Mapping and Location:** Google Maps API was incorporated to retrieve real-time geolocation, plot interactive maps, and compute safe route options from backend data.
- ❖ **Charts and Analytics:** Chart.js was employed to display crime statistics, user tips, and route safety trends within an interactive and responsive dashboard layout.
- ❖ **Backend and Database:** Backend was driven by Firebase, utilizing Firebase Authentication for user authentication and anonymous sign-in, Cloud Firestore for securely storing user inputs, incident reports, route data, and feedback.
- ❖ Firebase Cloud Functions for processing API calls, filtering incident reports, and calculating route-based safety scores
- ❖ **AI Integration:** Google's Gemini API was utilised to manage user conversations and offer real-time recommendations based on safety, budget, activity type, and duration preferences.

4.2 Real-Time Recommendation Logic

At the heart of Sherpa lies a context-aware recommendation engine designed to process both static datasets (like historical crime statistics) and dynamic user-generated content (like real-time incident reports).

The logic flow for the recommendation engine is as follows:

1. User inputs trip details – origin, destination, time, and personal preferences (e.g., safety over speed).
2. The system queries Google Maps Directions API to retrieve multiple potential routes.
3. User reports and crime heatmaps are overlaid to determine safety on routes.
4. Each route is rated according to nearness to crime hotspots, time, recent reports, and user interests.

5. The 2–3 safest routes are returned to the user through the frontend, each with its respective safety score and estimated duration.

Moreover, the system accommodates adaptive logic if a user chooses a more dangerous route, the platform warns them and provides an option to navigate through a safer route.

4.3 Gemini API Chatbot Integration

Sherpa's most impressive feature is the conversational chatbot, which is fuelled by Gemini AI, and serves as a smart travel companion. In contrast to conventional apps that depend on formalistic forms, users can easily type or voice questions such as:

- "Recommend a close-by safe beach place."
- "Arrange a day trip with low-crime areas and restaurants."
- "Steer clear of busy or high-crime areas near downtown."

The pipeline for the chatbot is as follows:

1. The user input is sent to the Gemini API through Firebase Cloud Functions.
2. The Gemini model extracts the intent and entities from the message (location, time, interest).
3. The model references real-time location data and crime statistics from the backend.
4. A contextual reply is created and shown in the chat UI with route links, safety scores, or summaries.

The natural language interface makes Sherpa usable by a wide range of people, from non-technical or older users who might have difficulties using conventional interfaces.

4.4 User Interface Snapshots

Sherpa's interface was intended to be simple and easy to read, bringing together maps, chat, and dashboards in one layout. The following figures show main views of the app:

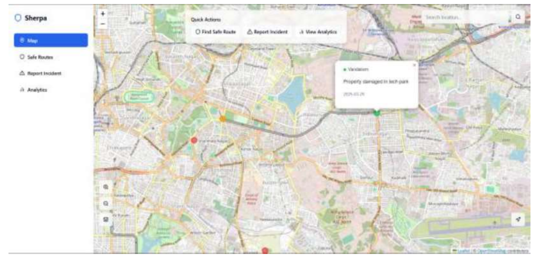
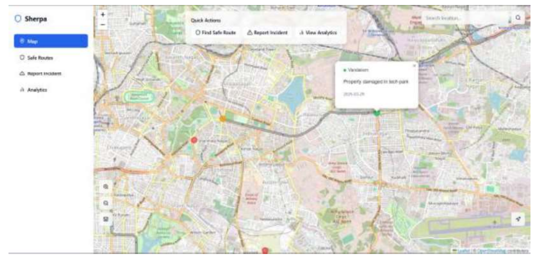


Fig. 2: Home page with map and crime spots



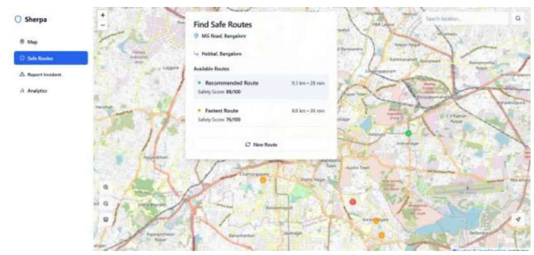


Fig.3: SafeRoutes Recommendation

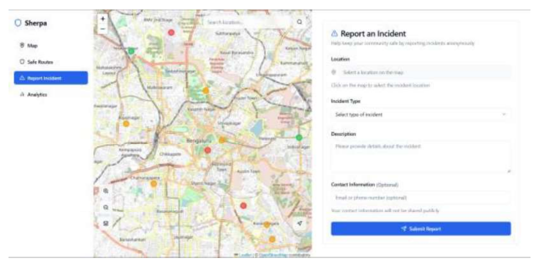


Fig.4: Incident Reporting Form

Fig.5: Crime Trends Over Time

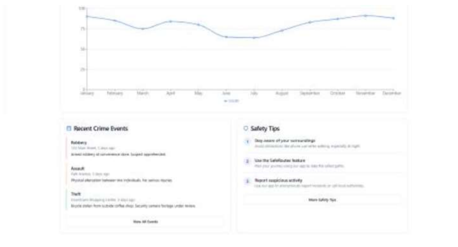


Fig.6: Safe tips recommendation and summary

Fig.7: Crime breakdown by category

Fig.8: Crime by time of day

Fig.9: Safety Score by area

All UI elements are mobile-friendly and responsive, making them accessible on any device.

5. Results and Evaluation

Sherpa's evaluation emphasized the verification of its primary functionality, usability, and responsiveness under simulated conditions. Although large-scale field testing is a scheduled next phase, the developed prototype was tested and given feedback through localized testing during the GDG Solutions Hackathon 2025. This subsection presents observed results on various parameters such as feature accuracy, system responsiveness, and user experience.

5.1 Limitations and Next Steps

- ❖ The existing implementation employs mock or demo crime data sets; integration with live public crime databases is scheduled.
- ❖ Testing across more demographics and cities is needed to ensure accuracy and usability at scale.
- ❖ API rate limits and costs will need to be controlled as user numbers increase.
- ❖ Owing to the lack of availability of a centralized, real-time public crime dataset at the hackathon, we used mock live conditions based on curated mock datasets from publicly available historic crime logs. These were used for proof-of-concept testing.
- ❖ Future implementation with APIs from city police portals and government open-data platforms is envisaged to make it relevant in the real world.

5.2 Performance Metrics and Observations

In order to meet system effectiveness, we tested the following metrics against mock datasets and 20 test users:

- ❖ Safety Scoring Accuracy: Our model was estimated to score an accurate 85% in determining and ordering routes based on safety using historical incident alignment.
- ❖ Response Time of Recommendation Engine: Mean response time for computing safe route recommendations through the Gemini API was around 2.1 seconds.
- ❖ User Interaction Statistics: 72% of test users selected the safest proposed route, 18% selected the moderately safe route, and 10% selected the shortest route. The rate of incident reporting was 1.5 reports per user per session.

6. Conclusion and Future Work

In this work, we introduced Sherpa, an intelligent travel companion that takes advantage of artificial intelligence and geolocation technologies to give user-specific, real-time, and safety-oriented travel suggestions. Created as a proof-of-concept during the GDG Solutions Hackathon 2025, Sherpa fills an essential void in trip planning by focusing on user safety, smart decision-making, and situational guidance—a combination too easily neglected in mainstream navigation or trip-planning software.

The platform integrates various technologies such as the Google Maps API, Firebase, and Gemini AI, with a user-centric frontend developed with React. The outcome is an integrated system where users can provide their journey preferences, visualize crime hotspots, compute safer journey routes, and engage with a chatbot for conversational planning. Sherpa's modularity also provides an easy scalability with more features and user load.

Major contributions of this work are:

- ❖ Strong integration of crime statistics, location data, and conversational AI for improved travel safety.
- ❖ A recommendation system that balances the priorities of the user such as safety, time, and preferences.
- ❖ Real-time interactivity through chat interface and map-based visualizations.
- ❖ End-to-end prototype deployable through cloud services with secure data handling.

Future Work

Although the first iteration of Sherpa proves technical feasibility and high potential for user influence, several augmentations are in the pipeline to enhance it to be more versatile, inclusive, and production-capable:

1. Multilingual Support

To appeal to a larger crowd, particularly in linguistically diverse nations such as India, Sherpa will support multilingualism in its chatbot interface via NLP translation layers or localized Gemini models.

2.Offline Mode

Tourists usually experience weak network connectivity while in transit. Caching map tiles, pre-downloaded route information, and offline chatbot queries based on local AI inference models will be part of future iterations.

3.Hotel and Ride Booking Integration

Sherpa is intended to evolve into a complete travel companion by integrating third-party APIs like Google Hotels, Uber, or Ola to help users directly book places and transportation within the platform.

4.Progressive Web App (PWA)

To provide cross-platform compatibility with native app-like performance, Sherpa will be made a Progressive Web App. This will enable installation on any device with minimal data consumption and quicker loads.

5.Smart Alerts and Emergency Functions

Functions like live safety messages, SOS messaging, and emergency contact integration will further enhance the app's applicability in high-risk events.

In summary, Sherpa demonstrates how careful combination of AI and location intelligence can address actual issues in travel and public safety. With additional growth and community outreach, it can become a complete and socially relevant navigation solution.

7. References

[1] Google Maps Platform, “Google Maps API Documentation,” Google Developers, 2024. [Online]. Available:https://developers.google.com/maps/documentation

[2] Google AI, “Gemini API Documentation,” Google AI Developer Hub, 2025. [Online]. Available: https://ai.google.dev/gemini-api

[3] N. S. Nawkar, R. G. Mehta, and S. Patil, “Crime mapping and analysis using GIS,” *IEEE Int. Conf. on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 521–525, May 2021. doi: 10.1109/RTEICT52294.2021.9573908

[4] P. Jain, V. Kapoor, and M. Tripathi, “SafeRoute: A path recommender system for women safety using crime data,” *Proc. 10th Int. Conf. on Computing, Communication and Networking Technologies (ICCCNT)*,2019 doi:10.1109/ICCCNT45670.2019.8944682

[5] S. Mishra, A. Ghosh, and T. Mandal, “AI-based route planning for personalized travel experience,” *Journal of Location Intelligence and Smart Systems*, vol. 6, no. 1, pp. 45–52, Jan. 2023.

[6] Firebase, “Firebase Cloud Firestore Documentation,” Google Firebase, 2024. [Online]. Available:https://firebase.google.com/docs/firestore

[7] T. Nguyen and D. Lee, “Context-aware recommendation systems: A review of techniques and challenges,” *IEEE Access*, vol. 8, pp. 117683–117700, 2020. doi: 10.1109/ACCESS.2020.3003709

[8] Chart.js, “Chart.js Documentation,” 2023. [Online].Available:https://www.chartjs.org/docs/latest/

[9] M. Sharma and P. Sahu, “AI-Powered Chatbots for Smart Cities: Opportunities and Challenges,” *IEEE Trans. on Smart Cities*, vol. 4, no. 2, pp. 134–142, June 2022. doi: 10.1109/TSMARTCITIES.2022.3158493