Dona Mary Shaju, 2026, 14:1
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Tourist Guide App

Dona Mary Shaju[1], Ancil Jacob[2], Archa .K. Udayan[3], Aswin Oommen Jacob[4], Dr. Rani Saritha R[5]

Department of Computer Applications Saintgits College of Engineering Kerala, India

Abstract- The Tourist Guide App is a mobile application developed to provide real-time, location-based travel assistance. The system is implemented using Flutter with Firebase for secure authentication and Cubit- based local caching to support efficient data management and limited offline access. Google Maps API enables live navigation, geolocation, and route visualization. An AI-based itinerary generation module produces optimized travel plans based on user preferences, destination, and travel duration. Additional features include destination search, favorites feature, and emergency service location. The proposed system demonstrates reliable performance, scalability, and improved usability for smart travel applications.

Keywords: Flutter, Firebase, Google Maps API, Location-Based Services, Artificial Intelligence, Itinerary Planning, Mobile Application, Cubit State Management, Smart Tourism.

I. INTRODUCTION

Tourism has rapidly evolved with the growth of mobile technology, as travelers increasingly rely on smartphones and internet access for instant, location-specific information. Traditional travel planning methods such as guidebooks and human guides are often time-consuming, expensive, and lack real-time updates, creating the need for smart digital solutions. To address these challenges, the proposed Tourist Guide App offers a lightweight, intelligent, and database-free mobile travel assistance system developed using Flutter.

The application provides real-time information on tourist spots, hotels, restaurants, and transport facilities, with Firebase used exclusively for secure user authentication and Cubit-based state management with caching enabling efficient local data handling and limited offline access. A key feature of the system is the AI-based itinerary generation module, which creates personalized and optimized travel plans based on user preferences, travel duration, and destination. Integration with Google Maps API allows real-time navigation and geolocation services, while additional features such as destination search, favorites management, reviews, and emergency service locators further enhance user convenience and safety. The paper is organized to present related work, system design, implementation results, and conclusions with future scope.

II. RELATED WORK

The rapid growth of mobile computing and location-based services has led to the development of many smart tourism applications aimed at improving travel experiences. Early digital tourist guide systems mainly provided static information through websites and mobile applications, including destination details, hotel listings, and basic navigation. However, these systems lacked real-time personalization, offline support, and intelligent travel planning capabilities.

Several existing applications integrate GPS technology and map services such as Google Maps API to support navigation and nearby location discovery. These systems provide accurate location tracking and route guidance but heavily depend on continuous internet connectivity and centralized cloud databases, which reduces performance in low-network environments.

Recent research has introduced artificial intelligence for personalized travel planning. AI-based recommendation systems analyze user preferences and geographic constraints to suggest suitable destinations. Although personalization improves user experience, most systems require complex backend infrastructure and continuous data synchronization, increasing maintenance cost and storage dependency.

Studies also highlight the importance of offline access and local caching in mobile applications. Local storage and state management techniques reduce latency and improve performance in poor network conditions. However, only a few systems effectively combine local caching, intelligent itinerary generation, and real-time navigation in a single platform.

Security remains an important concern in mobile applications. Cloud-based authentication ensures secure access but excessive reliance on cloud storage raises privacy and data management issues. From the survey of existing systems, it is evident that most applications focus on individual features rather than a unified solution. These limitations motivate the proposed Tourist Guide App, which integrates secure authentication, AI-based itinerary generation, local caching, real-time navigation, and emergency services within a lightweight and scalable architecture.

III. PROPOSED SYSTEM

The proposed Tourist Guide App is a mobile-based intelligent travel assistance system designed to provide real-time, location-based tourist information and personalized itinerary planning. The application is developed using the Flutter framework to ensure a responsive and cross-platform user interface. The main objective of the system is to help tourists efficiently explore destinations through smart navigation, automated travel planning, and integrated emergency support.

The system follows a lightweight and partially database-free architecture. Firebase is used exclusively for secure user authentication, while user preferences, favorites, and recent activity are managed locally using Cubit-based state management with caching. This approach reduces dependency on centralized databases, improves application performance, and supports limited offline access.

Google Maps API is integrated to enable real-time navigation, geolocation, and interactive map visualization. GPS-based tracking allows the system to identify nearby tourist attractions, hotels, restaurants, transportation facilities, and emergency services such as hospitals and police stations.

An artificial intelligence–based itinerary generation module generates optimized travel schedules based on user inputs such as destination, travel duration, and preferred categories. The system also includes destination search, favorites management, and review access features. The modular architecture ensures scalability, maintainability, and flexibility for future enhancements, making the application a reliable and intelligent travel assistance solution.

System Architecture

The proposed Tourist Guide App follows a lightweight, modular, and client-side architecture to ensure efficient performance and scalability. The user interface is developed using Flutter, which provides a responsive and cross-platform mobile environment. Core application functionalities such as destination search, itinerary generation, navigation, and emergency access are handled within the application logic layer.

Firebase is used exclusively for secure user authentication, including login and registration. Tourist information, user preferences, recently viewed locations, and favorites are managed locally using Cubit-based state management with caching. This approach eliminates the need for a centralized cloud database and supports limited offline access.
The system integrates the Google Maps API for real-time navigation, GPS-based location tracking, and interactive map visualization. Emergency services such as hospitals and police stations are also accessed through the Maps API. This architecture ensures secure access, fast data processing, and reliable real-time location-based services.

Modules

The proposed Tourist Guide App is organized into the following functional modules:

1. User Authentication Module – Handles secure user login and registration using Firebase authentication.

2. Location and Navigation Module – Provides real-time location tracking, route guidance, and map visualization using Google Maps API.
3. Itinerary Generation Module – Generates optimized and personalized travel schedules based on user preferences using artificial intelligence.
4. Destination Search Module – Enables users to search and explore nearby tourist attractions and relevant places.
5. Favorites and Review Module – Allows users to save preferred locations and view photos and reviews.
6. Emergency Services Module – Displays nearby emergency facilities such as hospitals and police stations for user safety.

Each module operates independently within a modular architecture and communicates efficiently through Cubit- based state management, ensuring scalability, maintainability, and reliable system performance.

Technologies Used

The proposed Tourist Guide App is developed using the Flutter framework with the Dart programming language to build a responsive and cross-platform mobile application. Firebase Authentication with Google Sign-In is implemented to provide secure, fast, and password-free user login and registration.

The Google Maps API is integrated to support real-time geolocation tracking, route navigation, and interactive map visualization. Cubit-based state management is used for efficient local data handling, caching, and user preference management. Gemini AI is employed for intelligent itinerary generation and personalized travel recommendations based on user inputs and preferences.

IV. METHODOLOGY

The methodology of the proposed Tourist Guide App describes the systematic process used for designing, developing, and implementing the application. The system follows a modular development approach to ensure scalability, maintainability, and ease of implementation. Each functional component is developed independently and integrated through efficient state management.

User authentication is implemented using Firebase Authentication with Google Sign-In to provide secure and seamless access. After successful login, the system retrieves the user's real-time location using GPS services. The Google Maps API is then used for map visualization, route navigation, and nearby location discovery.

The itinerary generation process utilizes Gemini AI to analyze user inputs such as destination, number of travel days, and preferred categories. Based on these inputs, the AI module generates an optimized and personalized travel schedule to improve travel efficiency.

Tourist-related data, user preferences, favorites, and recent activities are managed locally using Cubit-based state management with caching. This approach reduces network dependency, improves application performance, and supports limited offline access. The emergency services module continuously monitors the user's location and displays nearby hospitals and police stations to enhance traveler safety.

Workflow

The workflow of the proposed Tourist Guide App begins with user authentication using Firebase Authentication with Google Sign-In. After successful login, the system acquires the user's current location through GPS services and displays nearby tourist locations on the map using the Google Maps API. Users can search destinations, select preferences, and request itinerary generation. The entered details are processed by the Gemini AI module, which generates a personalized and optimized travel schedule. The generated itinerary is presented to the user for navigation and planning.

Users can save favorite locations, view reviews, and access emergency services at any time during the travel process. All user data and application states are managed locally using Cubit-based state management with caching. The workflow ensures secure access, real-time navigation, personalized planning, and improved traveler safety.

Algorithm

Algorithm 1: AI-Based Itinerary Generation Input: Source S, Destination D, Travel Days N, Preference Categories C

Output: Optimized Itinerary I

1. Read user inputs S, D, N, and C.
2. Retrieve nearby tourist locations L for destination D.
3. Filter locations in L based on user preferences C.
4. Collect metadata such as distance, visit duration, and ratings.
5. Construct the prompt and send it to the Gemini AI module.
6. Generate the itinerary for N days.
7. Validate and optimize the generated itinerary.
8. Display the final itinerary I using Google Maps.

UI/UX Design

The user interface of the Tourist Guide App is developed using the Flutter framework with emphasis on usability, accessibility, and smooth navigation. A clean and consistent layout is maintained across all screens to ensure an intuitive user experience. The authentication screen supports Google Sign-In for quick and secure access. The home screen displays nearby places along with search functionality, category-based filtering, and personalized recommendations.

The itinerary generation screen allows users to enter travel details such as destination, number of days, and preferences through a structured input form. The generated itinerary is displayed in a day-wise format with scheduled timings and location details. The map screen integrates the Google Maps API to provide live navigation, real-time location tracking, and route visualization. The place details screen presents images, ratings, and user reviews.

Additional features include favorites management for saving selected locations, itinerary sharing options, and an information screen displaying application and developer details. The overall UI/UX design ensures responsive interaction, clear icons, smooth transitions, and improved usability.

V. RESULTS AND DISCUSSION

The Tourist Guide App was successfully implemented using Flutter and evaluated on Android devices and emulators to assess functionality, performance, and user experience. Testing focused on major modules including secure authentication, real-time navigation, AI-based itinerary generation, Cubit-based local caching, emergency service access, and destination search. The application demonstrated stable performance across multiple test scenarios, confirming system reliability.

Firebase authentication was tested for new user registration and secure login. The system showed fast response time with consistent verification and minimal authentication failures. Since tourist data is not stored in a centralized database, Cubit-based local caching efficiently managed user preferences, favorite locations, and recently accessed destinations. This significantly reduced application loading time and improved performance, particularly in low- network environments.

Integration of the Google Maps API provided accurate navigation and real-time location tracking. The application successfully identified nearby tourist attractions, hotels, restaurants, hospitals, and police stations based on GPS data. Route optimization supported efficient travel paths, minimizing unnecessary travel. The emergency services module consistently displayed the nearest support facilities without noticeable delay, improving user safety.

The AI-based itinerary generation module was evaluated using various inputs such as destination, travel duration, and preferred categories. The generated itineraries were well structured and logically arranged, helping users reduce manual planning effort and improve travel efficiency.

From a usability perspective, the interface received positive feedback during testing. The Flutter-based design enabled smooth navigation, clear layouts, and fast response across different screen sizes. Features including favorites management, destination search, review viewing, and live updates enhanced overall user experience. The modular

architecture supports stable performance and future system expansion.

Overall, the results confirm that the proposed system meets its objectives by delivering secure authentication, intelligent travel planning, efficient local data handling, and reliable navigation, providing a practical and user-friendly solution for modern travelers.

VI. CONCLUSION

This project demonstrates the effective integration of mobile application development, location-based services, and artificial intelligence to enhance the travel experience. The Tourist Guide App provides real-time information on nearby tourist attractions, hotels, restaurants, transportation services, and emergency facilities through a user-friendly mobile interface developed using Flutter.

The system adopts a lightweight and database-free architecture, where Firebase is used exclusively for secure user authentication and Cubit-based local caching manages user preferences, favorites, and recently viewed locations. This design improves application performance, reduces dependency on continuous internet connectivity, and supports limited offline functionality. Integration of the Google Maps API enables accurate navigation and geolocation services, while the AI-based itinerary generation module produces personalized and optimized travel plans based on user inputs.

Experimental evaluation confirms stable performance across major functional modules including authentication, navigation, itinerary generation, emergency service access, and destination search. The application minimizes manual trip planning, improves travel efficiency, and enhances user safety and convenience.

In conclusion, the proposed Tourist Guide App serves as a comprehensive and intelligent travel assistance platform that supports smart tourism. The project highlights how modern technologies such as Flutter, Firebase, Google Maps API, Cubit state management, and artificial intelligence can be effectively combined to develop a scalable and practical mobile solution.

ACKNOWLEDGEMENTS

The authors sincerely express their gratitude to the Department of Computer Applications, Saintgits College of Engineering, for providing the necessary facilities and technical support to carry out this project successfully. The authors also thank the project guide and faculty members for their valuable guidance, continuous encouragement, and constructive feedback throughout the development of this work.

Special thanks are extended to all peers and participants who provided useful suggestions and support during the testing and evaluation phase of the application.

REFERENCES

1. Batty, M., Hudson-Smith, A., Milton, R., and Crooks, A. (2015). Map-based analysis in smart tourism. Computers, Environment and Urban Systems, 54, 92–102.
2. Zhang, J., Li, Y., and Li, M. (2020). Location-based services for smart tourism: A survey. IEEE Access, 8, 141– 158.
3. Google Developers. (2024). Google Maps Platform Documentation.
4. Flutter Team. (2024). Flutter SDK Documentation. Google.
5. Firebase Team. (2024). Firebase Authentication Documentation. Google.
6. Caddeo, F., Busacca, M. G., and Carta, S. (2019). Smart tourism and mobile applications: A review. Journal of Hospitality and Tourism Technology, 10(3), 290–305.
7. Gavalas, A., Konstantopoulos, C., and Pantziou, D. (2014). Mobile recommender systems in tourism. Journal of Network and Computer Applications, 39, 319–333.
8. Wixom, B. H., and Watson, H. J. (2010). The BI-based organization. International Journal of Business Intelligence Research, 1(1), 13–28.
9. Varma, K. R., and Reddy, N. (2020). Artificial intelligence based travel itinerary planning

system. International Journal of Engineering Research and Technology (IJERT), 9(6), 512–516.
10. OpenWeather Team. (2024). OpenWeather API Documentation.
11. Kenteris, S., Gavalas, D., and Economou, D. (2011). An innovative mobile electronic tourist guide application. Personal and Ubiquitous Computing, 15(5), 523–533.