IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.141265

PERSONALISED RECOMMENDATION SYSTEM IN SMART CITIES

Ms. Punitha M R[1], B N Rushitha[2], Chaitra C[3], Harsha C V[4], and M Saija[5]

Assistant Prof., Department of CSE, K. S. School of Engineering and Management, Bangalore[1]

Student, Department of CSE, K. S. School of Engineering and Management, Bangalore[2-5]

Abstract: Smart cities provide a wide range of options for dining, leisure, and essential services, but users often struggle to identify the most suitable choices due to generalized search results and scattered information across multiple applications. This leads to increased decision-making time and reduced convenience. To address this challenge, we developed Explore Hub, a lightweight real-time recommendation platform designed to help users discover restaurants, weekend destinations, and essential locations efficiently. The system integrates Google Places and Geocoding APIs with a React frontend and a Node.js Express backend to deliver accurate place details, interactive map navigation, comparison features, and favorites management using Local Storage. The prototype demonstrates that real-time API-based recommendations, combined with map-based visualization, can enhance user experience and improve decision-making without requiring complex backend infrastructure or machine-learning models.

Keywords: Smart Cities, Recommendation System, Google Places API, React, Node.js, Map Visualization.

I. INTRODUCTION

Smart cities offer diverse services and recreational options, but users often struggle to identify suitable choices due to generalized search results and scattered information. Existing platforms lack personalized filtering, comparison support, and map-based clarity. To overcome these limitations, Explore Hub provides real-time recommendations using Google Places and Geocoding APIs, with a React–Express architecture and Leaflet map visualization. The system enables efficient discovery, comparison, and saving of favorite locations using Local Storage, improving decision-making without requiring complex backend infrastructure.

This research focuses on developing a lightweight and practical platform that streamlines the process of discovering nearby places without requiring complex backend systems. Instead of relying on historical user data or machine-learning models, ExploreHub uses real-time information from Google APIs to provide accurate recommendations and map-based visualization. The system emphasizes usability, speed, and efficiency by integrating comparison features and favorites management to support quick decision-making in everyday scenarios.

II. RELEVANT LITERATURE

Existing urban recommendation systems often rely on centralized databases, making them vulnerable to single points of failure. While some platforms have implemented basic recommendation algorithms, they frequently struggle with personalization and scalability. Recent work by Zhang and Liu (2021) and Zhang et al. (2015) has shown promise in addressing these challenges through partition-based modeling and social network integration.

However, these approaches still face limitations in handling dynamic urban environments. Our work builds upon these foundations while introducing a more robust and scalable architecture. Our system uses real-time data from Google APIs and LocalStorage, replacing the need for a central database in this prototype version.

III. SYSTEM DESIGN AND METHODOLOGY

System Components

The proposed system consists of the following components:

1. **User Interface (React**) - Users interact with the web application through a responsive UI built using React components.
2. **Navigation (React Router DOM) -** Provides smooth page transitions between search page, results, favorites, and map view without reloading.
3. **Frontend Request Handling (Axios) -** User actions such as search or filter selection trigger API requests from the frontend to the backend.

DOI: 10.17148/IJARCCE.2025.141265

4. **Backend Processing (Express server) -** The Express server receives the request, securely accesses Google APIs through server-side Axios, and manages environment variables via dotenv.
5. **External API Communication (Google APIs)-Google Places API:** Retrieves real-time place information (ratings, reviews, photos, open status).
   Google Geocoding API: Converts text queries into geographic coordinates.
6. **Response Parsing & Formatting -** The backend processes and formats the data before sending it back to the frontend.
7. **Visualization Layer -** Returned coordinates are plotted on an interactive map with markers for easier exploration and spatial context.
8. **Local Storage -** Saves user-selected favorite places locally for quick access and session persistence without needing a database.
9. **CORS & Security -** CORS enables secure communication between frontend and backend, while API keys remain hidden via environment variables.

Methodology

Frontend Layer (React):
Users enter search queries or apply filters through the React interface.
React Router DOM manages page navigation between Search, Results, Map, Comparison, and Favorites.
Axios sends the request from the frontend to the backend.

Backend (Express):
Express receives requests and extracts search parameters.
Backend uses Axios to call Google Places and Geocoding APIs with secure environment variables.
API responses are processed, formatted, and returned to the frontend.

Google APIs:
Places API: Provides real-time details including ratings, reviews, location coordinates, and photos.
Geocoding API: Converts location text into latitude and longitude.

Map Visualization (Leaflet):
Retrieved coordinates are plotted on an interactive Leaflet map.
Place markers allow visual comparison and access to detailed information.

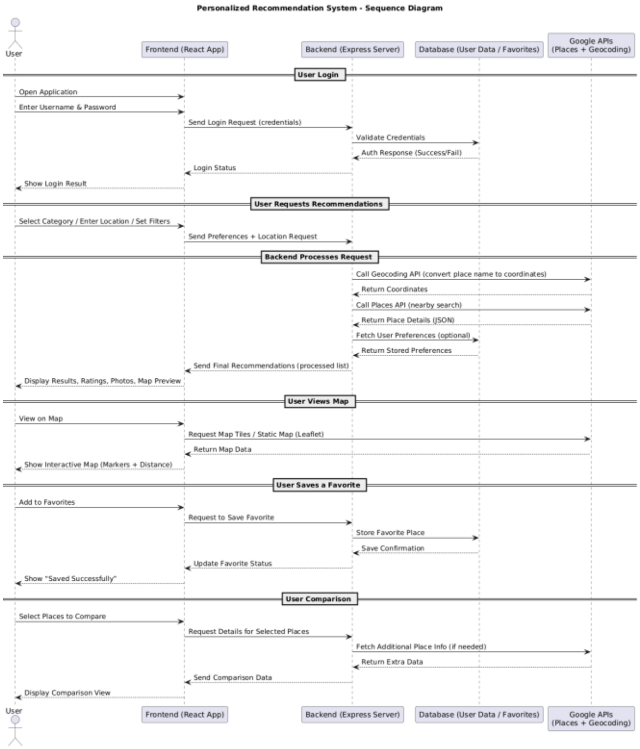
Personalized Recommendation System - Sequence Diagram
User
Frontend (React App)
Backend (Express Server)
Database (User Data / Favorites)
Google APIs (Places + Geocoding)
User Login
Open Application
Enter Username & Password
Send Login Request (credentials)
Validate Credentials
Auth Response (Success/Fail)
Login Status
Show Login Result
User Requests Recommendations
Select Category / Enter Location / Set Filters
Send Preferences + Location Request
Backend Processes Request
Call Geocoding API (convert place name to coordinates)
Return Coordinates
Call Places API (nearby search)
Return Place Details (JSON)
Fetch User Preferences (optional)
Return Stored Preferences
Send Final Recommendations (processed list)
Display Results, Ratings, Photos, Map Preview
User Views Map
View on Map
Request Map Tiles / Static Map (Leaflet)
Return Map Data
Show Interactive Map (Markers + Distance)
User Saves a Favorite
Add to Favorites
Request to Save Favorite
Store Favorite Place
Save Confirmation
Update Favorite Status
Show "Saved Successfully"
User Comparison
Select Places to Compare
Request Details for Selected Places
Fetch Additional Place Info (if needed)
Return Extra Data
Send Comparison Data
Display Comparison View
User
Frontend (React App)
Backend (Express Server)
Database (User Data / Favorites)
Google APIs (Places + Geocoding)

Fig. 1 High-Level Diagram

DOI: 10.17148/IJARCCE.2025.141265

IV. RESULTS AND DISCUSSIONS

The Explore Hub platform was tested with real user data and demonstrated the following outcomes:

1. The system successfully retrieves real-time recommendations for restaurants, weekend spots, and essential locations using Google Places and Geocoding APIs.
2. Interactive map visualization enabled users to explore nearby locations more easily and understand distance and spatial distribution.
3. Users were able to compare selected places side-by-side, helping them make quicker decisions**.**
4. Favorites stored using Local Storage remained available across browser sessions without requiring a database or login.
5. Initial user testing with a small group reported improved convenience and reduced time spent searching across multiple apps.
6. The prototype demonstrated fast response and smooth UI transitions due to the lightweight React–Express architecture.
7. Feedback indicated that the combination of list view and map-based view added clarity and enhanced usability.

V. CONCLUSION AND FUTURE WORK

The project demonstrates that an effective recommendation platform can be built using lightweight architecture without requiring machine learning or large historical datasets. By leveraging real-time Google API data and a simple client–server model, ExploreHub provides a practical solution for discovering dining and leisure options efficiently. Future enhancements include adding authentication, database storage for cross-device sync, and ML-based personalized predictions.

The introduction of mobile applications using React Native will improve accessibility and user engagement through push notifications and location-based services. Additionally, the platform will incorporate blockchain technology to enhance transaction security and user trust. Further enhancements include the integration of augmented reality for virtual property tours and service previews, as well as the development of a comprehensive loyalty program to reward active users and service providers. These advancements will solidify CityScape's position as a leader in urban lifestyle solutions while continuously adapting to the evolving needs of modern city dwellers.

REFERENCES

[1]. L. Zhang and Y. Liu, *"A Partition-Based Partial Personalized Model for Points-of-Interest Recommendations,"* **IEEE Transactions on Knowledge and Data Engineering**, vol. 33, no. 1, pp. 144–158, Jan. 2021.

[2]. F. Zhang, C. Yuan, D. Lian, X. Xie, and W. Ma, *"Personalized Point-of-Interest Recommendation Using Check-In History and Friends' Interests,"* in **Proc. 24th ACM Int. Conf. Information and Knowledge Management (CIKM)**, 2015, pp. 1211–1220.

[3]. H. Gao, J. Tang, and H. Liu, *"Exploring Temporal Effects for Location Recommendation on Location-Based Social Networks,"* in **Proc. 7th ACM Conf. Recommender Systems (RecSys)**, 2013, pp. 93–100.

[4]. J. Yuan, G. Cong, and A. Sun, *"Graph-Based Point-of-Interest Recommendation with Geographical and Temporal Influences,"* in **Proc. 23rd ACM Int. Conf. Information and Knowledge Management (CIKM)**, 2014, pp. 659–668.