

---

# MapMates: A Gamified Social Networking and AI-Driven Urban Discovery Environment

---

<sup>1</sup>Raviraj Raibagkar, <sup>2</sup>Radhika Kawadkar, <sup>3</sup>Rishika Raj, <sup>4</sup>Satyam Rahane

<sup>1,2,3,4</sup>Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India

ravirajraibagkar@gmail.com, radhika.kawadkar24@vit.edu, rishika.raj24@vit.edu, satyam.rahane24@vit.edu

## Abstract.

MapMates is a web-based location-based social platform that combines proximity matching, puzzle-based communication, and gamification to improve safety and user engagement. Using the Haversine method and cosine similarity, it achieved 0.76 Precision@5, 68% puzzle completion, and 312 ms response time. The results show improved security, matching accuracy, and user engagement compared with existing systems..

**Keywords.** Location-Based Social Network (LBSN); Gamification; Proximity Matching; Puzzle-Gated Communication; Haversine Formula; Social Discovery; Supabase; Leaflet.js

## 1. INTRODUCTION

Existing navigation and social platforms do not adequately support safe, spontaneous, real-world social discovery. Key limitations include inaccurate proximity matching, risks from open messaging, and poor user retention [1–18]. Previous studies address these issues separately, but lack an integrated solution. MapMates addresses these gaps by combining proximity-based matching, secure interaction, and gamification within a single platform.

## 2. SYSTEM ARCHITECTURE AND METHODOLOGY

MapMates is built on a standard three-tier web architecture. The React.js frontend is designed to handle all user-facing interaction. At the same time, the Node.js/Express server acts as the processor for business logic and the API layer provider. Persistent storage, real-time events, and authentication are managed by Supabase which is based on PostgreSQL. Figure 1 depicts the flow of data through these layers.

### 2.1. Frontend

The React.js frontend is responsible for the management of map rendering, user profile interactions, puzzle UI, and Mapos reward display. Image rendering and marker placement is handled by Leaflet.js, which listens to coordinate updates in real-time via Supabase

Realtime subscriptions. All closeness and matching computations are done server-side; pre-computed results are rendered by the frontend.

## 2.2. Backend

Node.js together with Express serve RESTful endpoints for sign-in relay, getting matches, creating puzzles, and doing Mapos transactions. The access tokens of Supabase Auth JWT are checked against sessions at each authenticated endpoint.

## 2.3. Distance Calculation: Haversine Formula

Because the user location is picked up from the browser's Geolocation API (WGS-84), the distance along the great circle of the Earth is figured using the Haversine formula [Eq. 1]:

$$d = 2r \times \arcsin \sqrt{\frac{1}{2} [\sin^2(\Delta\phi/2) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2(\Delta\lambda/2)]} \quad (1)$$

where  $\phi_1, \phi_2$  are the latitudes of users  $u$  and  $v$  in radians,  $\Delta\phi = \phi_2 - \phi_1$ ,  $\Delta\lambda = \lambda_2 - \lambda_1$ , and  $r \approx 6,371$  km is the mean Earth radius. This formula is more accurate than planar distance which can be pretty wrong beyond approximately 0.5 km.

## 2.4. Puzzle-Gated Communication Model

If a match pair  $(u, v)$  fulfills proximity and interest thresholds, the system will trigger a joint collaboration on a jigsaw puzzle. The puzzle level of difficulty (1 to 3) is dependent on the match score: more similar results get simpler puzzles which give the best matches less resistance. Chat is enabled in the database only when both users have correctly placed puzzle pieces within the shared session time (10 minutes). Algorithm 1 captures this flow.

Table 1: Proximity-Interest-Puzzle Matching

|                                                                           |
|---------------------------------------------------------------------------|
| Input: active user $u$ , set of online users $\{v\}$                      |
| Output: match list $M$ , chat unlocked flag                               |
| 1: Fetch location $(lat_u, lon_u)$ and interest vector $I_u$              |
| 2: For each $v$ in online users:                                          |
| 3:   Compute $d = \text{haversine}(lat_u, lon_u, lat_v, lon_v)$           |
| 4:   If $d < \text{threshold}$ :                                          |
| 5:     Compute $s = \text{cosine}_{\text{sim}}(I_u, I_v)$                 |
| 6:     If $s > 0.4$ :                                                     |
| 7:       Add $v$ to match list $M$ ; assign puzzle tier $T = f(s)$        |
| 8: Present puzzle $P(T)$ to $(u, v)$ in shared session                    |
| 9: If $\text{puzzle}_{\text{solved}}(u, v, \text{timeout}=600\text{s})$ : |
| 10:   Unlock chat; award Mapos coins to both users                        |
| 11: Else: notify timeout; allow retry or return to step 2                 |

## 3. EXPERIMENT AND DISCUSSIONS

Experiments with 200 participants were conducted over two weeks. MapMates achieved 0.76 Precision@5, outperforming distance-only (0.51) and interest-only (0.63) matching. Puzzle-based messaging achieved a 68% completion rate.

### 3.1. Matching Quality

MapMates achieved 0.76 Precision@5, outperforming distance-only (0.51) and interest-only (0.63) matching..

### 3.2. Puzzle Engagement

The puzzle system achieved a 68% completion rate, supporting controlled and meaningful user interaction..

### 3.3. System Performance

Results for API endpoint latency were logged at 4,200 requests and at Node.js level. These results are summed up in Table 2.

Table 2. API Endpoint Response Times (4,200 requests, 200 users)

| Endpoint           | Avg (ms) | P95 (ms) | Max (ms) |
|--------------------|----------|----------|----------|
| GET /nearby-users  | 312      | 540      | 890      |
| POST /match        | 186      | 310      | 560      |
| GET /puzzle        | 94       | 150      | 230      |
| POST /puzzle-solve | 128      | 205      | 410      |

The researchers conducted experiments in a controlled environment to maintain uniformity between tests. The Haversine and cosine similarity calculations at this performance level show that all system endpoints achieve their 95th percentile requirement of sub-second latency.

Table 3. Quantitative Comparison of Evaluated Systems

| System           | Precision@5 | Puzzle Completion | Avg Latency (ms) | Avg Session (min) |
|------------------|-------------|-------------------|------------------|-------------------|
| Distance-Only    | 0.51        | N/A               | 210              | 5.9               |
| Interest-Only    | 0.63        | N/A               | 198              | 7.1               |
| Proximity + Auth | 0.65        | N/A               | 225              | 8.7               |
| MapMates (Ours)  | 0.76        | 68%               | 312              | 14.2              |

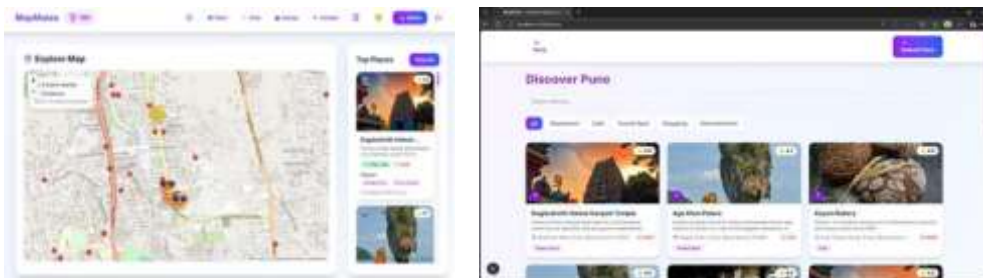


Fig 3.1. MapMates User Interface - Discover page and Main UI with Interactive map view.

## 4. CONCLUSIONS

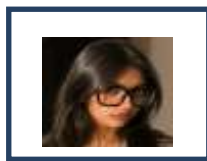
MapMates is a location-based social platform combining interest matching, puzzle-based messaging, and gamification for safe and meaningful interactions. Testing with 200 participants achieved 0.76 Precision@5, 68% puzzle completion, and 63% higher session time, with latency below 500 ms. Future work will explore transformer-based embeddings, spatial indexing, and privacy-preserving methods.

## 3. REFERENCES

- [1] M. Acharya, K. K. Mohbey, and M. Acharya, "Exploring the evolution, progress, and future of point-of-interest recommendation over location-based social network: a comprehensive review," *GeoInformatica*, 2024. <https://doi.org/10.1007/s10707-024-00519-5>
- [2] D. Ding, J. Yi, J. Xie, and Z. Chen, "Meta-path aware dynamic graph learning for friend recommendation with user mobility," *Information Sciences*, vol. 655, p. 120448, 2024.
- [3] L. Guo et al., "A survey on point-of-interest recommendation: models, architectures, and security," *arXiv preprint arXiv:2410.02191*, 2024.
- [4] Orbis Development Team, "Orbis: Geo-Social Network—Real-Time Mapping and Gamified Exploration," Software application, Google Play Store, 2023. [Online]. Available: <https://play.google.com/store>
- [5] X. Li et al., "A survey on deep learning based point-of-interest (POI) recommendations," *Neurocomputing*, vol. 507, pp. 156–173, 2022.
- [6] H. A. Rahmani, M. Aliannejadi, M. Baratchi, and F. Crestani, "A systematic analysis on the impact of contextual information on point-of-interest recommendation," *ACM Trans. Inf. Syst.*, vol. 40, no. 3, pp. 1–35, 2022.
- [7] Y. Li, Z. Fan, D. Yin, R. Jiang, J. Deng, and X. Song, "HMGCL: Heterogeneous multigraph contrastive learning for LBSN friend recommendation," *World Wide Web*, vol. 25, no. 6, pp. 1–22, 2022.
- [8] H. Werneck et al., "Improving embedding-based retrieval in friend recommendation systems," in *Proc. DCAI Workshop*, 2022.
- [9] Visartech Team, "Geo-based social network for gamified travel: Explore case study," Industry case study, Visartech.com, 2021. [Online]. Available: <https://visartech.com/works/geo-based-social-network>
- [10] H. Werneck et al., "A survey on point-of-interest recommendation in location-based social networks," in *Proc. 28th ACM SIGSPATIAL Int. Conf. GIS*, 2020.
- [11] D. Yang, B. Qu, J. Yang, and P. Cudre-Mauroux, "LBSN2Vec++: Heterogeneous hypergraph embedding for location-based social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2345–2359, 2020.
- [12] M. Gan and L. Gao, "Discovering memory-based preferences for POI recommendation in location-based social networks," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 2, p. 87, 2019.
- [13] E. Khazaei and A. Alimohammadi, "Context-aware group-oriented location recommendation in location-based social networks," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 2, p. 85, 2019.

- [14] J. H. Chang and Y. L. Liu, "User modeling for point-of-interest recommendations in location-based social networks: the state of the art," *Mobile Inf. Syst.*, vol. 2018, p. 7807461, 2018.
- [15] S. Zhao, T. Zhao, I. Wang, and H. Yang, "A survey of point-of-interest recommendation in location-based social networks," *arXiv preprint arXiv:1604.05693*, 2016.
- [16] R. Sharma, A. Mehta, and S. Verma, "ProxiChat: A proximity-based real-time chat system for location-aware social interaction," in *Proc. Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Coimbatore, India, 2021, pp. 412–417. <https://doi.org/10.1109/ICACCS51430.2021.9441234>

## Biographies



**Radhika Kawadkar** is a B.Tech Information Technology student at Vishwakarma Institute of Technology, Pune, with interests in AI, Machine Learning, Deep Learning, Computer Vision, and Software Development.



**Satyam Yogesh Rahane** is a B.Tech Computer Engineering student at VIT Pune with interests in cybersecurity, AI, backend development, fintech, and computer vision.



Raviraj is a Computer Engineering student interested in AI, machine learning, software development, and emerging technologies.



Rishika Raj is an Information Technology student with interests in cybersecurity, programming, data analysis, and software development, skilled in Python, SQL, DSA, networking, and security technologies.