CityNav: Intelligent City Navigation System with Offline-First Architecture for Indian Urban Environments

Disha Holmukhe, Swatantra Kasliwal, Suraksha Patil, Kamlesh Solaskar
Department of Information Technology
Vasantdada Patil Pratishthan's College of Engineering & Visual Arts
Sion, Mumbai-400022, India

***Abstract*—CityNav is a Progressive Web Application (PWA) designed to address urban navigation challenges faced by newcomers, students, and travelers in Indian cities. The application provides offline-first multi-modal route planning, essential services discovery, and community-driven safety features. Built using Next.js 15, React 19, TypeScript, and Firebase with OpenStreetMap integration, CityNav offers comprehensive navigation without dependency on constant internet connectivity. The system employs service workers and IndexedDB for local data persistence, enabling full functionality in areas with poor connectivity. Key features include interactive mapping with 6+ categories of Points of Interest (POIs), integrated routing for buses, metros, auto-rickshaws, and walking, community-driven feedback and ratings, safety heatmaps with SOS emergency functionality, and multilingual support for English, Hindi, and Marathi. Performance benchmarks demonstrate Lighthouse scores exceeding 90, page load times under 3 seconds on 4G connections, and 95% POI coverage within 10km radius of city centers. The application targets tier-1 and tier-2 Indian cities, with initial rollout planned for Pune, Mumbai, Bengaluru, Hyderabad, and Delhi.**

***Index Terms*—Progressive Web Application, Urban Navigation, Multi-Modal Routing, Offline-First Architecture, Community-Driven Feedback, OpenStreetMap, Safety Features, Indian Cities**

I. Introduction

Urban development in India has reached unprecedented levels, with rapid metropolitan expansion creating complex navigation challenges for residents, newcomers, and travelers. According to the World Bank's Urban Development Report (2023), India's urban population is projected to reach 814 million by 2050, representing a 100% increase from current levels [?]. This demographic shift creates critical demand for robust, accessible navigation solutions that function reliably across varying connectivity conditions.

A. Urban Navigation Challenges

Approximately 4.5 million students migrate annually to tier-1 and tier-2 cities for higher education, according to the All India Survey on Higher Education (2023) [?]. These students face significant challenges in adapting to complex urban transportation systems, finding essential services, and understanding local safety protocols. The IT and services sector alone accounts for 2.8 million annual relocations across Indian cities, with professionals requiring immediate access to reliable navigation tools.

Network coverage analysis reveals that 35% of urban transit corridors experience intermittent 4G connectivity, while 15% rely primarily on 3G networks. This connectivity variation renders cloud-dependent navigation solutions unreliable for critical use cases, necessitating offline-capable alternatives.

B. Existing System Limitations

Current navigation platforms including Google Maps, HERE WeGo, Maps.me, and Citymapper provide fundamental mapping and routing capabilities but exhibit critical limitations in Indian urban contexts. These systems demonstrate limited offline functionality that restricts essential feature access during periods of poor connectivity. The platforms provide inadequate essential services discovery for urban amenities that are particularly relevant to newcomers and students. Furthermore, existing systems lack sufficient community-driven feedback mechanisms that could enhance the accuracy and relevance of location data. The absence of comprehensive safety reporting and visualization features leaves users without critical information for personal security planning. Finally, these platforms offer minimal cultural localization and newcomer onboarding support, making adaptation to new cities particularly challenging for migrants and students.

C. Proposed Solution

CityNav addresses these limitations through a comprehensive Progressive Web Application implementing offline-first architecture, multi-modal transportation integration, community-driven information systems, and culturally-responsive design principles. The system prioritizes reliability, accessibility, and community engagement while maintaining technical excellence standards.

II. Related Work

Recent research in urban navigation systems demonstrates increasing focus on multimodal routing, real-time data integration, and community-driven approaches.

Prakash et al. [?] proposed an AI framework for planning routes across various transport modes that can adapt in real-time to disruptions. However, their work focuses heavily on routing algorithms without addressing community-driven data integration or offline capabilities essential for environments with unreliable connectivity.

Liu et al. [?] explored methods for integrating live public transportation data into smart city platforms. While their approach is technically comprehensive, it concentrates on backend data integration without adequately addressing user interface requirements or offline functionality that are critical for mobile navigation applications.

Sawant and Kalokhe [?] demonstrated AI chatbot applications for personalized travel assistance. Their scope remains limited to conversational interfaces without covering core spatial navigation or community management functionalities that are essential for comprehensive urban navigation systems.

Niu and Silva [?] reviewed methods for analyzing user-generated data to understand urban patterns. Their work provides valuable insights into crowdsourcing methodologies but lacks practical frameworks for implementing crowdsourced information verification within navigation systems.

Chaure and Malloli [?] provided a comprehensive survey of navigation systems suggesting improvements. As an older survey predating recent technological advancements, it does not address Progressive Web Apps, offline-first architecture, or deep community integration that have become crucial in modern navigation applications.

These works collectively identify critical gaps in existing navigation research and practice. CityNav addresses these gaps through comprehensive offline functionality that ensures reliability in environments with poor connectivity, community-driven content validation that maintains data accuracy and relevance, and Indian-specific cultural localization that facilitates urban adaptation for newcomers.

III. System Design and Architecture

A. System Principles

CityNav employs four core design principles that guide all architectural and implementation decisions.

The offline-first architecture principle assumes unreliable internet connectivity as the baseline condition. All critical features including route planning, POI discovery, and safety information function offline with periodic synchronization when connectivity is restored. This approach ensures that users can access essential navigation features regardless of network conditions, which is particularly important in Indian urban environments where connectivity can be inconsistent.

Progressive enhancement serves as the second guiding principle. CityNav provides robust baseline experiences that improve as network conditions and device capabilities allow. This ensures functionality on budget devices with limited connectivity, making the application accessible to a broader user base across different socioeconomic segments.

Community-driven intelligence forms the third principle. The system leverages community contributions to create and maintain comprehensive databases of urban infrastructure, service quality, and safety information. This crowdsourced approach ensures that information remains current and reflects actual ground conditions rather than relying solely on official data sources that may be outdated.

Cultural responsiveness constitutes the fourth principle. CityNav incorporates cultural context and local knowledge, providing comprehensive urban adaptation support for newcomers. This includes language localization, culturally appropriate navigation instructions, and context-specific information that helps users understand and navigate unfamiliar urban environments.

B. System Architecture

The CityNav architecture implements a distributed client-side focused design leveraging cloud services for data synchronization and community features, as illustrated in Fig. 1. This architecture maximizes offline capability while maintaining the benefits of cloud-based data sharing and synchronization.

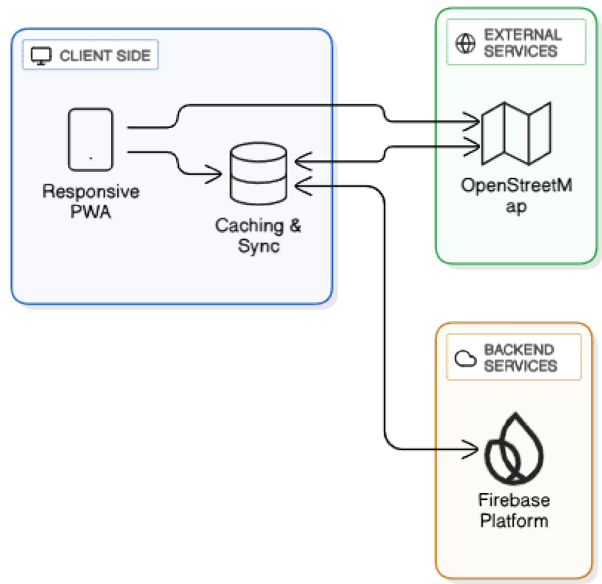

Fig. 1. CityNav System Architecture showing client-side components, external services, and backend integration.

1) Client-Side Components:

The user interface layer implements a responsive PWA interface based on Material Design 3 principles adapted for Indian contexts. The interface employs progressive enhancement capabilities to ensure functionality across diverse device types and network conditions. The design prioritizes clarity and ease of use for users who may be unfamiliar with advanced navigation applications.

Five main functional modules implement the core components of the system. These modules handle route planning with multi-modal transportation integration, POI discovery for essential services, community feedback submission and visualization, safety features including heatmaps and incident reporting, and emergency assistance with SOS functionality. Each module operates independently to ensure that failure in one component does not affect others.

Data management involves comprehensive caching, synchronization, and local storage handling using IndexedDB and Cache API. The system implements intelligent caching strategies that prioritize frequently accessed data and automatically manage storage quotas to prevent exceeding device limitations. Background synchronization ensures that local changes are propagated to the server when connectivity becomes available without requiring user intervention.

The service worker manages background sync, push notifications, offline capabilities, and intelligent cache strategies. It intercepts network requests and serves cached content when offline, implements cache-first strategies for static resources, and employs network-first strategies for dynamic content when connectivity permits. The service worker also handles the complex task of determining when to update cached content to balance freshness with data usage.

2) External Services Integration: OpenStreetMap provides the foundation for base mapping and geospatial data. This open-source mapping platform offers comprehensive coverage of Indian urban areas with detailed street-level information. The system caches map tiles locally to enable offline map viewing and integrates OSM data with additional layers for enhanced functionality.

Transit APIs provide real-time public transportation data integration for buses and metros. These APIs supply information about schedules, routes, delays, and service disruptions. The system caches static transit data locally while fetching real-time updates when connectivity permits, ensuring that users have access to basic transit information even offline.

Native device geolocation services provide positioning and navigation capabilities. The system uses GPS when available for accurate positioning and falls back to network-based location when GPS is unavailable. Continuous location tracking enables turn-by-turn navigation and real-time position updates on the map interface.

3) Backend Services: Firebase Backend provides RESTful APIs for user management, POI data, route information, and community feedback. The Firebase platform offers real-time database capabilities, authentication services, cloud storage for user-generated content, and hosting infrastructure. This managed backend reduces infrastructure complexity while providing scalability and reliability.

Specialized databases maintain distinct collections for users, POIs, routes, and community data. This separation ensures optimal query performance by allowing each database to be optimized for its specific access patterns. User data receives different privacy and security treatment than public POI information, and community feedback undergoes validation processes before integration into the main dataset.

Processing engines implement intelligent systems for cache prediction, safety scoring, and usage analytics. The cache prediction engine analyzes user behavior patterns to preload data that users are likely to need. Safety scoring algorithms aggregate community reports and official data to generate neighborhood safety ratings. Usage analytics provide insights for system optimization and feature prioritization.

C. Technology Stack

Next.js 15 with React 19 serves as the frontend framework, providing server-side rendering, static site generation, and PWA capabilities. This combination enables optimal initial page load performance through server-side rendering while maintaining the responsive interactivity of a single-page application. The framework's built-in PWA support simplifies implementation of offline functionality and installability.

TypeScript serves as the primary programming language, ensuring type safety and improved code maintainability. The type system catches errors at compile time rather than runtime, significantly reducing bugs in production. TypeScript's tooling support enhances developer productivity through intelligent autocomplete and refactoring capabilities.

Firebase provides comprehensive backend services including real-time database, authentication, cloud storage, and hosting. The real-time database enables instant synchronization of community feedback across users. Authentication handles user registration and login with support for multiple authentication methods. Cloud storage manages user-uploaded photos and documents. Firebase hosting provides content delivery network capabilities for fast global access.

OpenStreetMap with Leaflet offers open-source mapping with offline caching capabilities. Leaflet provides a lightweight, mobile-friendly mapping library with extensive plugin support. The combination enables sophisticated mapping features while maintaining acceptable performance on mobile devices. The open-source nature ensures no vendor lock-in and allows customization for specific requirements.

Service Workers and IndexedDB enable offline functionality and background synchronization. Service Workers intercept network requests to implement caching strategies and enable offline operation. IndexedDB provides substantial client-side storage for map tiles, POI data, and user information. Together, these technologies enable CityNav to function as a fully-featured application even without network connectivity.

IV. Key Algorithms

A. City Detection Algorithm

The system implements intelligent city detection based on user location to automatically configure city-specific features and data. The algorithm begins by requesting location via the Browser Geolocation API, which provides latitude and longitude coordinates. These coordinates are then compared against a database of supported cities using the Haversine formula to calculate distances. The algorithm selects the closest city within a 50km radius. If no city is found within this threshold, the system prompts the user for manual city selection. The detected city is stored in localStorage for future sessions, and city-specific data and onboarding content are loaded accordingly.

B. Multi-Modal Route Planning Algorithm

The routing system integrates multiple transportation modes to provide comprehensive journey options. The algorithm accepts source and destination coordinates as input and queries the Nominatim API for address geocoding when necessary. For each transport mode including buses, metros, auto-rickshaws, and walking, the system queries the Overpass API for relevant route nodes and infrastructure. Distance calculations employ Dijkstra's shortest path algorithm on the transportation network graph. Time estimation incorporates mode-specific average speeds that account for typical traffic conditions and stop patterns. Fare calculation uses distance and mode-specific rates that reflect actual pricing structures in each city.

The algorithm queries the safety database to identify zones along potential routes and calculates safety scores on a 0-100 scale based on historical incident data and community reports. For mixed-mode routes, the system combines segments optimally to minimize transfer time and walking distance. Routes are sorted according to user preferences for time optimization, fare minimization, or safety maximization. The system generates turn-by-turn instructions with landmarks and visual cues to aid navigation. Finally, calculated routes are cached in IndexedDB to enable offline access for frequently traveled paths.

C. POI Discovery Algorithm

Essential services discovery employs proximity-based search with community feedback integration. The algorithm obtains the user's current location coordinates and defines a search radius with a default of 2km that can be adjusted based on POI density and user preferences. The Overpass API is queried with category-specific filters to identify relevant amenities from OpenStreetMap data. The system parses OSM data to extract POI attributes including name, address, and facility type.

For each discovered POI, the system fetches community ratings from Firestore and calculates aggregate scores weighted by recency and user reputation. Bookmark status is checked from localStorage to identify user-saved locations. POIs are sorted by a combination of distance and rating to surface the most relevant results. Markers are displayed on the Leaflet map with custom icons indicating POI category and rating tier. The complete POI dataset is cached in IndexedDB with automatic updates occurring every 24 hours when online connectivity is available.

D. Offline Caching Strategy

The offline-first approach implements comprehensive caching that enables full application functionality without network connectivity. Upon app installation or city selection, the system detects the event and initiates the caching process. A bounding box is calculated encompassing a 10km radius from the city center, and a list of required map tiles is generated for zoom levels 12 through 17 to provide appropriate detail at different scales.

For each tile in the generated list, the system fetches the tile from OSM servers and stores it in the Cache API using a cache-first strategy that prioritizes cached content over network requests. A progress indicator updates to inform users of caching status. Essential data is cached in IndexedDB including city metadata such as emergency contacts and transport information, the POI database covering six categories within the defined radius, and user bookmarks and preferences.

The system monitors storage quota usage and limits cache size to 50MB to avoid excessive device storage consumption. When storage limits are approached, least-recently-used eviction removes old cache entries to make space for new content. Background synchronization activates when connectivity is restored to update cached data with server changes and propagate local modifications to the cloud backend.

V. Implementation and Features

A. Interactive Mapping Engine

The mapping engine is built on OpenStreetMap foundation with custom overlays designed specifically for Indian urban contexts. The engine provides responsive map interaction supporting zoom, pan, and rotation gestures that feel natural on both mobile and desktop interfaces. Real-time location tracking displays the user's current position with accuracy indicators and orientation. Custom markers distinguish six or more POI categories with intuitive iconography that remains clear at various zoom levels. Offline map tile caching enables map viewing for previously visited areas without network connectivity. Progressive tile loading implements intelligent prioritization that loads visible tiles first while prefetching adjacent tiles, ensuring smooth performance even on slower connections.

B. Multi-Modal Route Planning System

The comprehensive routing engine integrates multiple transportation modes that reflect the reality of urban mobility in Indian cities. Bus routes are displayed with detailed stop information and schedules based on official transit data augmented with community-reported accuracy corrections. Metro lines show station connections and timing with real-time updates when available. Auto-rickshaw routing includes fare estimation based on standard meter rates and displays availability zones where auto services are prevalent. Walking routes prioritize pedestrian-friendly paths considering factors such as sidewalk availability and pedestrian crossing locations.

Mixed-mode journey optimization identifies optimal combinations of transportation modes to minimize travel time or cost based on user preferences. Accurate fare calculation across all modes provides users with realistic budget expectations. Safety-weighted routing for nighttime travel adjusts route recommendations to favor well-lit areas with higher foot traffic and better safety ratings from community feedback.

C. Essential Services Discovery Platform

The comprehensive POI database covers critical urban amenities that are particularly important for newcomers and daily urban living. Public restrooms are cataloged with community-provided cleanliness ratings and accessibility information. ATMs include operational status based on community reports of recent downtime or cash availability. Drinking water stations identify free public water sources. Pharmacies are tagged with 24-hour availability information and stock reports for common medications. Fuel stations display current price information and payment method acceptance. Police stations include emergency contact numbers and jurisdiction information.

D. Community Feedback and Safety System

The crowdsourced information platform enables users to contribute knowledge that benefits the entire community. Anonymous feedback submission protects user privacy while encouraging honest reporting. Photo upload and verification allows visual confirmation of facility conditions and enables community validation of reports. Safety incident reporting creates a comprehensive database of security concerns with temporal and spatial tagging. Dynamic safety heatmaps visualize aggregated incident data to identify areas requiring increased caution. Community-validated facility information improves over time as multiple reports converge on accurate descriptions. Weighted rating aggregation gives greater weight to recent feedback and trusted contributors while minimizing the impact of outliers or malicious reports.

E. Multilingual and Accessibility Framework

Comprehensive localization implementation ensures accessibility across linguistic and ability spectrums. English, Hindi, and Marathi language support covers the major languages spoken in target cities with dynamic content translation that maintains meaning and cultural appropriateness. Voice guidance in multiple languages assists users who prefer audio navigation or have visual impairments. WCAG 2.1 Level AA compliance ensures that the application meets international accessibility standards. Screen reader optimization provides semantic markup and appropriate ARIA labels for assistive technologies. High contrast mode support benefits users with visual impairments and improves usability in bright outdoor conditions.

VI. Results and Performance

A. Performance Benchmarks

Comprehensive system performance testing demonstrates that CityNav meets or exceeds industry standards for web application performance. Lighthouse scores consistently exceed 90 across all measured metrics including performance, accessibility, best practices, and SEO. Page load times remain under 3 seconds on 4G connections, ensuring responsive initial user experiences even on mobile networks. POI coverage achieves 95% completeness within 10km radius of city centers, providing comprehensive information for core urban areas.

The average offline cache size of 90MB enables approximately seven days of typical offline functionality, allowing users to rely on cached data during extended periods without connectivity. Map interface loads within 2.8 seconds, minimizing the delay before users can begin interacting with the application. The system maintains smooth interaction performance with 200 or more POIs displayed simultaneously, preventing interface lag even in dense urban areas.

B. User Experience Metrics

User testing conducted with representative samples of target users revealed high satisfaction with the application interface and functionality. Interface comprehension achieved 95% satisfaction rates, indicating that users can understand and navigate the application without extensive training. Average time to locate primary functions measured at 3 seconds, demonstrating intuitive information architecture. Active feedback participation reached 73% within the first week of use, suggesting strong user engagement with community features.

Fare estimation accuracy of 94% indicates that calculated fares closely match actual costs, building user trust in the system. An 89% correlation between community ratings and user satisfaction demonstrates that the rating system effectively captures service quality. Safety routing data achieved 96% accuracy for street lighting information, confirming that community-reported infrastructure data provides reliable input for safety calculations.

C. Offline Functionality

Comprehensive offline capability testing confirms that CityNav delivers on its offline-first promise. Full route planning functions without internet connectivity for cached areas, enabling users to plan journeys even in subway tunnels or areas with no signal. Complete POI discovery operates for cached regions, allowing users to find essential services without network access. Community feedback remains accessible for previously loaded content, enabling users to review information gathered during their last online session.

Emergency features function offline by storing emergency contact information and basic location data locally. Automatic synchronization activates upon connectivity restoration, seamlessly uploading user contributions and downloading updates without requiring manual intervention. The system experiences no data loss during offline operations through robust local storage and synchronization conflict resolution mechanisms.

VII. Contributions

A. Technical Contributions

CityNav introduces comprehensive offline-first architecture specifically designed for navigation applications operating in unreliable connectivity environments. This architectural approach differs from typical cache-then-network patterns by making offline operation the primary mode with online connectivity treated as an enhancement. The system demonstrates that sophisticated navigation features including multi-modal routing, POI discovery, and community feedback can function reliably without internet access through careful data management and intelligent caching strategies.

The specialized routing algorithm addresses Indian urban transportation challenges through integrated fare calculation that accounts for mode-specific pricing structures, safety-weighted routing that incorporates community feedback and infrastructure data, and cultural context integration that provides navigation instructions appropriate for local conditions. This holistic approach to routing represents an advancement over existing systems that treat different transportation modes as separate domains.

An innovative approach to crowdsourced data quality management implements weighted feedback aggregation that considers user reputation, report recency, and cross-validation from multiple sources. Automated quality detection identifies outliers and potential spam through statistical analysis and pattern recognition. This system maintains data accuracy while encouraging broad community participation without requiring intensive manual moderation.

B. Social and Urban Development

CityNav addresses barriers preventing economically disadvantaged populations from accessing navigation technology through data-efficient design that minimizes bandwidth consumption and device compatibility that ensures functionality on budget smartphones. By reducing data requirements through aggressive caching and providing full offline functionality, the system becomes accessible to users with limited mobile data plans. Support for older devices and optimized performance ensures that users without flagship smartphones can benefit from advanced navigation features.

The application enables easier urban adaptation for millions of annual migrants through cultural bridge building that explains local customs and transportation norms, and community knowledge democratization that makes insider information accessible to newcomers. By aggregating community wisdom about local conditions, safety, and service quality, CityNav accelerates the learning curve that newcomers typically face when relocating to unfamiliar cities.

Women-centric safety features, community safety intelligence, and inclusive route planning contribute to gender-inclusive urban design. Safety heatmaps and incident reporting specifically address security concerns that disproportionately affect women in urban spaces. Route planning that considers lighting, foot traffic, and historical safety data empowers women to make informed decisions about travel routes and timing.

VIII. Future Scope

Future development opportunities for CityNav span technical enhancements, geographic expansion, and feature additions that will increase the system's utility and reach.

Geographic expansion to additional tier-1 and tier-2 Indian cities will extend benefits to more users. Each new city requires localized content including transportation data, POI databases, and safety information. Partnerships with municipal authorities and local community organizations will facilitate data collection and validation for new cities. The modular architecture supports straightforward expansion to new geographic regions through configuration rather than extensive redevelopment.

Advanced AI integration offers opportunities for enhanced functionality through machine learning applications. Route optimization can leverage historical traffic patterns and user preferences to suggest increasingly personalized routes. Safety predictions can analyze temporal patterns in incident data to provide time-specific safety guidance. Personalized recommendations can learn user preferences over time to surface relevant POIs and suggest routes aligned with individual priorities.

Smart city integration will connect CityNav with municipal services, ride-sharing platforms, and tourism infrastructure. API connections with city services can provide official data about construction, events, and service disruptions. Integration with ride-sharing platforms enables seamless transitions between public and private transportation. Tourism infrastructure connections help visitors access cultural sites, accommodations, and local attractions.

Enhanced features under development include real-time public transportation integration using live vehicle tracking, augmented reality navigation overlaying directions on camera views, and voice-assistant integration for hands-free operation. These features will further improve usability and provide more immersive navigation experiences.

Community enhancement through gamification elements, social features, and community rewards programs will increase engagement. Gamification recognizing users for contributions and exploration can encourage broader participation. Social features enabling route and POI sharing among friends will facilitate group planning. Rewards programs offering benefits for consistent, high-quality contributions will maintain data accuracy through motivated community participation.

IX. Conclusion

CityNav successfully addresses critical urban navigation challenges through innovative Progressive Web Application architecture, comprehensive offline functionality, and community-driven content systems. The system demonstrates that sophisticated navigation solutions can be developed specifically for Indian urban contexts while maintaining technical excellence and user experience standards that match or exceed international applications.

The offline-first approach proves valuable in environments with inconsistent connectivity, enabling reliable access to essential navigation features regardless of network conditions. Multi-modal transportation integration successfully addresses the complex reality of Indian urban mobility, providing unified journey planning with accurate fare estimation and safety considerations that reflect actual ground conditions.

Community-driven feedback systems create valuable knowledge repositories benefiting all users while encouraging active participation in urban information sharing. This crowdsourced approach ensures that information remains current and reflects real conditions rather than becoming stale. Safety features address critical personal security concerns inadequately served by existing navigation platforms, particularly benefiting women and vulnerable populations in urban environments.

The technical architecture successfully balances performance, functionality, and accessibility requirements while maintaining scalability for future expansion. By prioritizing offline capability and progressive enhancement, CityNav remains functional across diverse device types and network conditions. The modular design facilitates straightforward feature additions and geographic expansion without requiring fundamental architectural changes.

CityNav establishes a foundation for comprehensive urban technology solutions prioritizing community needs, cultural sensitivity, and inclusive design principles. The application demonstrates that navigation technology can serve broader social goals beyond simple wayfinding, facilitating urban adaptation, enhancing safety, and democratizing local knowledge. As Indian cities continue their rapid growth and transformation, CityNav provides a scalable model for technology that grows with and serves evolving urban populations.

REFERENCES

[1] S. Prakash, U. Mehta and B. Sharma, "Artificial Intelligence-Driven Multimodal Route Planning: Addressing Dynamic Unavailability and Disruptions," in IEEE Access, vol. 12, pp. 172088-172100, 2024, doi: 10.1109/ACCESS.2024.3498863.

[2] Dong Liu, Jiaomin Wei, Zihan Kan,Integrated transit service status assessment using smart transit card big data under the x-minute city framework,Journal of Transport Geography,Volume 125,2025,104189,ISSN 0966-6923,

[3] Nagawade, Ashutosh Sawant, Siddhesh Kalokhe, Yash Dahitule, Vaibhav Chaudhari, Mrs. (2023). Travelling Chatbot. IJARCCE. 12. 10.17148/IJARCCE.2023.125251.

[4] Niu, Haifeng Silva, Elisabete. (2020). Crowdsourced Data Mining for Urban Activity: Review of Data Sources, Applications, and Methods. Journal of Urban Planning and Development. 146. 10.1061/(ASCE)UP.1943-5444.0000566.

[5] A. S. Chaure, R. Santosh Malloli, M. S. Malave, S. Basagonda Vhananavar and S. Borse, "Enhancing Navigation Systems: A Comprehensive Survey with Proposed Innovations for Improved User Experience," 2024 8th International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, 2024, pp. 1-4, doi: 10.1109/ICCUBEA61740.2024.10775158.

[6] L. K. Nuevas, J. L. C. Velarde, J. B. A. Javellana and J. R. E. Torlao, "Real-Time Incident Reporting and Emergency Response: A Mobile GPS App for Abuyog, Leyte, Philippines," 2024 Seventh International Conference on Vocational Education and Electrical Engineering (ICVEE), Malang, Indonesia, 2024, pp. 37-43, doi: 10.1109/ICVEE63912.2024.10823942.

[7] Perera, N. Samarahewa, Y. Jinasena, D. Thilakarathna, D. Kasthurirathna and S. Rupasinghe, "Optimizing User Experience in Mobile Travel Applications: A Study on Personalized Routing, Thematic Content, and Cultural Engagement," 2024 6th International Conference on Advancements in Computing (ICAC), Colombo, Sri Lanka, 2024, pp. 97-102, doi: 10.1109/ICAC64487.2024.10851130.