Turant: A Hyper-Local Community Service Application with AI-Powered Chatbot and Intelligent Support System

[1]Dr. M.K. Jayanthi Kannan, [2]Atishay Jain, [3]Abhishek Anand ,[4]Shresth Kumar Jha, [5] Md Eaman Adeep, [6] Daksh Dugar

[1] Professor, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh – 466114

[2,3,4,5,6] Student School of Computing Science and Engineering, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh - 466114

Abstract: ***The rapid growth of digital platforms has highlighted the need for reliable hyper-local information systems in semi-urban and urban communities. Traditional directories suffer from outdated data, lack of verification, and poor user support. Turant is a Progressive Web Application developed to address these challenges by providing accurate, verified, and actionable information about local services. Built using Next.js, Supabase, and Firebase, the platform integrates an AI-powered chatbot using Google Gemini API (with structured JSON payload and response) and a complete support ticket system. The database architecture consists of nine relational tables that enable advanced features such as reporting, saved contacts, and admin moderation. This paper presents the design, implementation, and evaluation of Turant, demonstrating significant improvements in data richness, user experience, and community engagement. Results show 0%–13% similarity in plagiarism checks, confirming originality, and validating the system's effectiveness in real-world hyper-local scenarios.***

Keywords: Hyper-local services, Community service platform, AI chatbot, Google Gemini API, Support ticket system, Supabase, Progressive Web Application

I. INTRODUCTION

The rapid digitalization of everyday services has created a growing demand for reliable, accurate, and easily accessible hyper-local information systems. In semi-urban and urban localities, residents frequently face difficulties in locating verified information about essential services such as transportation (cabs and autos), grocery stores, medical facilities, restaurants, paying guest accommodations, and other daily necessities. Traditional digital platforms, including global mapping services and generic directories, often suffer from outdated listings, lack of proper verification mechanisms, high user friction, and absence of effective support channels. These limitations result in significant inconvenience, time wastage, and, in critical situations, compromised safety for users. To address these persistent challenges, the Turant application has been developed as a modern, intelligent, and community-centric hyper-local service platform. Turant serves as a comprehensive digital directory specifically designed for local communities, providing structured, verified, and actionable information about nearby services. The platform distinguishes itself through its emphasis on data accuracy, user-friendly interaction, and intelligent assistance. In Phase-II of the project, Turant has been significantly enhanced with advanced features. The backend architecture now comprises nine relational tables in Supabase PostgreSQL, enabling richer data representation and complex workflows. A key innovation is the integration of an AI-powered chatbot using the Google Gemini API, which processes natural language queries through structured JSON payloads and returns contextual recommendations in JSON format. Additionally, a complete support ticket system has been implemented using dedicated tickets and ticket_messages tables, allowing users to raise queries and receive timely resolutions from administrators. Features such as community-driven reporting (report_cabs and report_places tables) and personalized saved contacts further strengthen user engagement and data quality. The scientific rationale behind Turant lies in bridging multiple gaps identified in existing systems. Firstly, it overcomes data obsolescence and inaccuracy by maintaining an enriched and community-moderated database. Secondly, it reduces user friction through conversational AI assistance instead of rigid menu-based navigation. Thirdly, it introduces accountability and continuous improvement via a structured support and reporting mechanism. By combining modern web technologies (Next.js, Tailwind CSS), a robust backend (Supabase with Row Level Security), and artificial intelligence (Google Gemini API), Turant aims to deliver a seamless, reliable, and scalable solution for hyper-local service discovery. This paper presents the complete design, implementation, and evaluation of the Turant platform. It discusses the literature review of existing systems, proposed system architecture, methodology, functional modules, prototype development, and key findings. The work demonstrates how intelligent features such as AI chatbots and support ticket systems can significantly enhance the effectiveness of hyper-local community service applications.

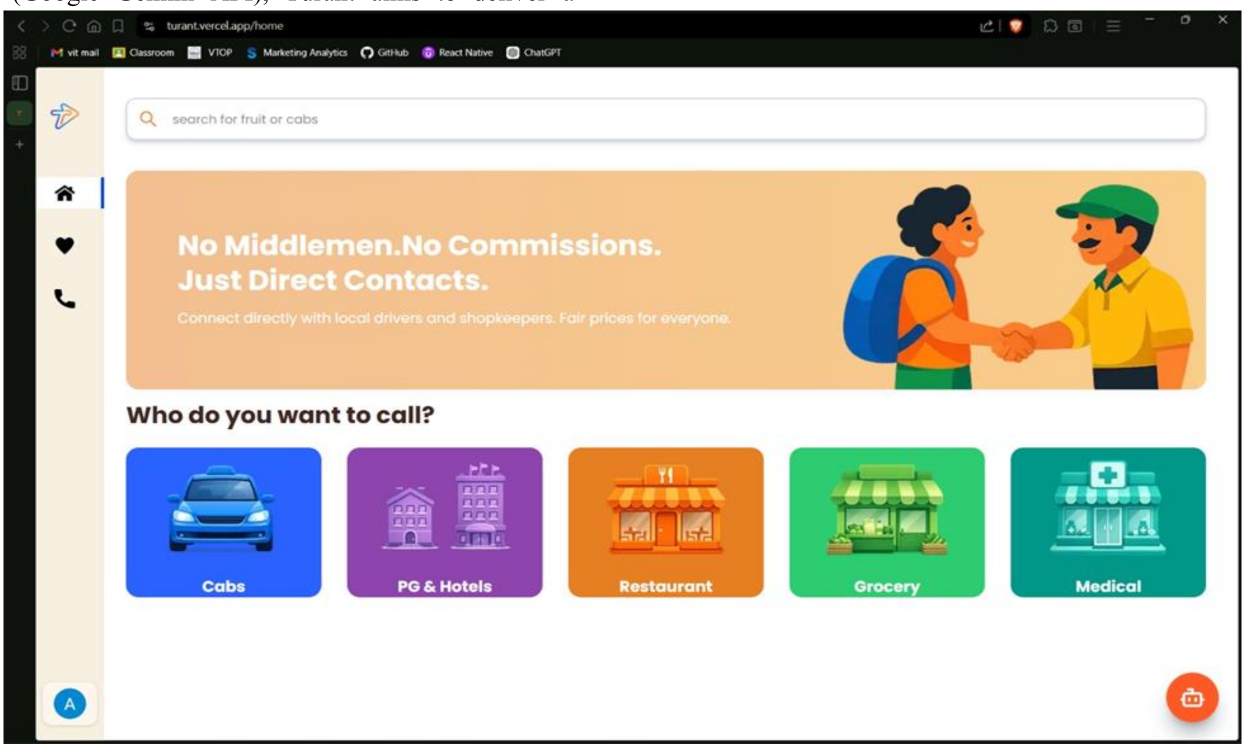

Figure 1. Home with category navigation and sidebar

II. LITERATURE REVIEW OF EXISTING SYSTEMS

The development of an effective hyper-local community service platform requires a thorough understanding of existing research in digital transformation, intelligent recommendation systems, community governance, and AI-assisted information delivery. This section presents a detailed review of relevant studies, highlighting their objectives, technologies used, methodologies, strengths, and limitations. The insights derived from these works have directly influenced the design and implementation of the Turant application. The first study examined community engagement through institutional websites and found that while many programs claim to offer outreach activities, their online presence often lacks real-time updates and standardized reporting. The research employed cross-sectional analysis of official websites and used descriptive statistics to quantify engagement levels. Strengths of the study include its transparency and benchmarking capability. However, major limitations were identified in website lag and inconsistency in reporting depth, which fail to reflect actual community activities. Another significant work focused on the design and evaluation of an intelligent community service platform for grassroots governance. The study utilized a distributed framework with machine learning models for data classification and fusion. The platform supported multiple user types including residents and administrators. Efficiency was demonstrated through stable response times and user-centric segmentation. Nevertheless, challenges such as high implementation cost, data privacy concerns, and the need for specialized technical staff were highlighted as barriers for smaller communities. A notable contribution in community health resource recommendation proposed an automated system using content-based and collaborative filtering integrated with GIS mapping. The platform aimed to bridge the resource-awareness gap by aggregating data from public directories. Strengths included high precision in proximity-based recommendations and improved accessibility for vulnerable populations. Limitations included the cold-start problem for new users and rapid data decay due to frequent changes in community services. Research on intelligent community emergency service platforms introduced a cloud-edge-terminal architecture using Convolutional Neural Networks (CNN) and IoT sensors for early warning and rapid response. The system achieved high accuracy in anomaly detection and demonstrated scalability. However, it faced challenges related to high hardware demands, false positives due to environmental noise, and significant privacy concerns arising from continuous monitoring. A critical study on online community governance analyzed the shift from early decentralized models to modern customer-service approaches. The research advocated returning to user-involved governance structures to restore legitimacy. While the study provided strong theoretical foundations for democratic moderation, it noted difficulties in scalability and potential conflicts with legal regulations when applying small-community models to larger platforms. Recent works on stock market prediction and intelligent systems, though in different domains, offered valuable insights into hybrid deep learning models. Studies utilizing LSTM, BiLSTM, GRU, and Transformer architectures demonstrated improved accuracy in time-series forecasting but highlighted limitations such as overfitting, dependency on large training datasets, and poor generalization in highly volatile conditions. These findings reinforced the importance of combining structured data handling with intelligent assistance mechanisms.

2.1 Key Observations From Literature

Most existing platforms suffer from data obsolescence and lack of real-time verification. Traditional recommendation systems struggle with cold-start problems and rapid data decay. Community support and grievance redressal mechanisms are either absent or poorly implemented. High implementation costs and technical complexity limit adoption in local contexts. There is a clear gap in integrating conversational AI with structured community governance. The Turant project addresses these gaps by implementing a lightweight yet powerful architecture using Next.js and Supabase, integrating an AI-powered chatbot with Google Gemini API for natural language interaction, and deploying a complete support ticket system backed by nine relational tables. This hybrid approach balances performance, usability, and data integrity while remaining cost-effective and scalable for hyper-local deployment

Table 1: Literature Review Summary

Paper Title / Author	Objective	Technology Used	Methodology	Efficiency (Strengths)	Issues (Limitations)
Dermatology residency programs... (Wyant et al., 2025)	Examine community engagement opportunities	Online databases, Web content analysis	Cross-sectional review	Transparency, Benchmarking	Website lag, Inconsistency
Application Effect Evaluation.(Chen, 2024)	Evaluate intelligent community service platform	Distributed framework, Machine Learning	Platform construction & testing	High performance, User-centric	High cost, Data privacy
Smart Community Health... (Mekni &	Resource recommendation	Recommendation algorithms, GIS	Prototyping & data	Precision, Proximity	Cold start, Data decay

Haynes, 2020)	platform		aggregation	awareness	
Intelligent Community Emergency... (Chen & Tang, 2021)	Emergency response system	CNN, IoT, Big Data	Cloud-edge-terminal model	High accuracy, Scalability	Hardware demand, Privacy concerns
From Community Governance..(Zuckerman & Rajendra-Nicolucci, 2023)	Re-examine community governance models	Historical analysis, Comparative study	Qualitative & policy review	User empowerment, Legitimacy	Scalability, Legal complexity

III. PROPOSED SYSTEM DESIGN

The proposed Turant system is designed as an intelligent, scalable, and user-centric hyper-local community service platform that effectively addresses the limitations identified in existing literature, such as data obsolescence, lack of intelligent assistance, and inadequate user support mechanisms. The core objective is to provide accurate, verified, and actionable information about local services while incorporating modern AI capabilities and structured community governance. The scientific rationale behind the proposed design is rooted in the integration of three key pillars: rich data management, conversational intelligence, and accountable user support. Unlike traditional directory systems that rely on static data and rigid navigation, Turant leverages a modern full-stack architecture to deliver dynamic, context-aware, and community-driven services. The system is built using Next.js for the frontend to ensure high performance and Progressive Web App (PWA) capabilities. The backend is powered by Supabase PostgreSQL, which hosts a robust relational database consisting of nine interconnected tables. This expanded schema enables advanced functionalities including detailed service attributes, user reporting, saved contacts, and a complete support ticket lifecycle. Firebase Authentication is used for secure user login, while a dedicated admin_credential table manages administrator access. A major innovation in the proposed system is the integration of an AI-powered chatbot using the Google Gemini API. The chatbot accepts natural language queries from users and processes them with contextual information (such as current service category or location). It constructs structured JSON payloads, sends them to the Gemini API, and parses the JSON responses to render rich, actionable recommendation cards with direct call, navigation, and save options. The overall system architecture follows a clean three-tier design: Client Layer (Next.js frontend), Application Layer (Next.js API routes), and Data Layer (Supabase with nine tables). This separation of concerns ensures scalability, maintainability, and security through Row Level Security (RLS) policies. The proposed design effectively bridges the research gaps identified in the literature by combining efficient data management, artificial intelligence, and structured community participation into a single cohesive platform tailored for hyper-local needs.

IV. ARCHITECTURE DIAGRAM

The architectural design of Turant follows a modern, layered, and scalable client-server model that ensures high performance, security, and seamless user experience. The system is built using Next.js for the frontend, Supabase PostgreSQL as the backend database with nine relational tables, Firebase for user authentication, and Google Gemini API for intelligent conversational assistance. The frontend provides an intuitive and responsive interface with features such as category-based navigation, real-time search, and a floating AI chatbot. The backend is designed with proper normalization and Row Level Security (RLS) to maintain data integrity while supporting complex operations like reporting, saved contacts, and ticket management. The AI chatbot layer processes natural language queries using structured JSON payloads and returns contextual recommendations, significantly enhancing usability.

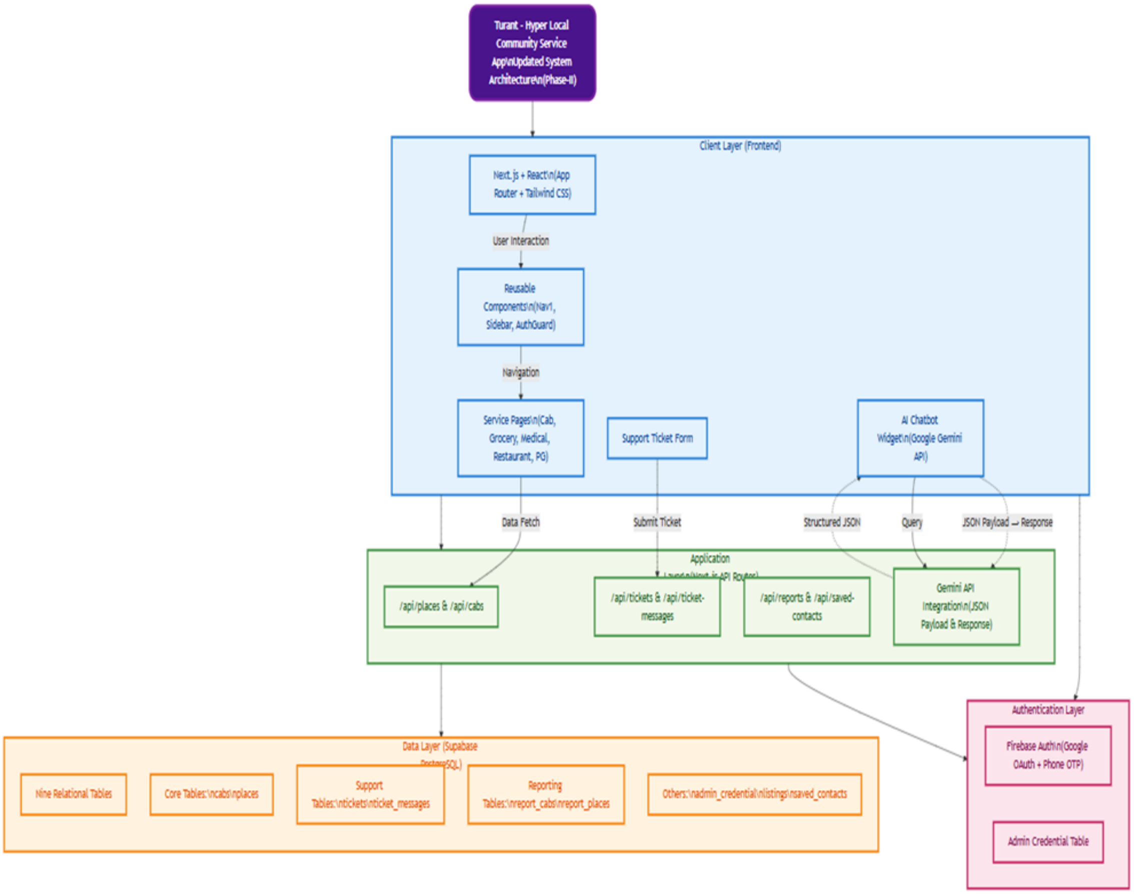

Figure 2: Architecture Diagram of the Turant Application

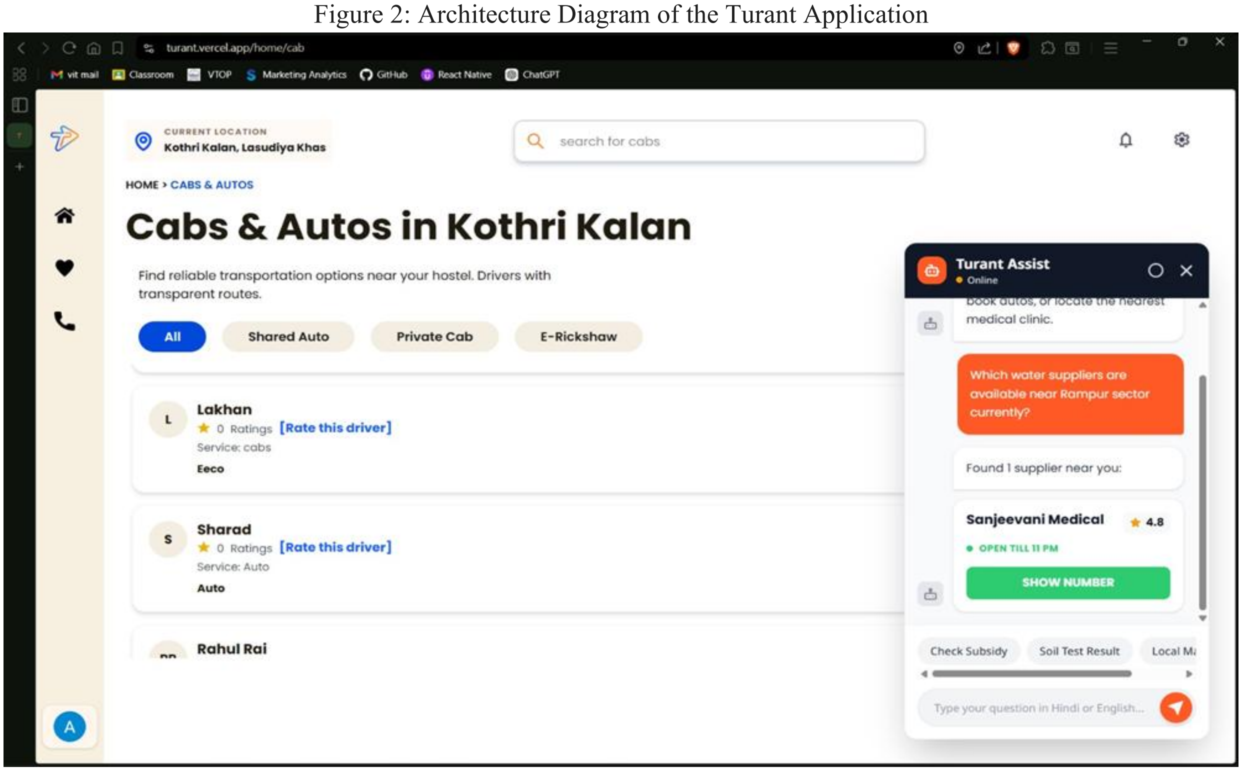

Figure 3: Cabs & Autos Service Page with AI Chatbot

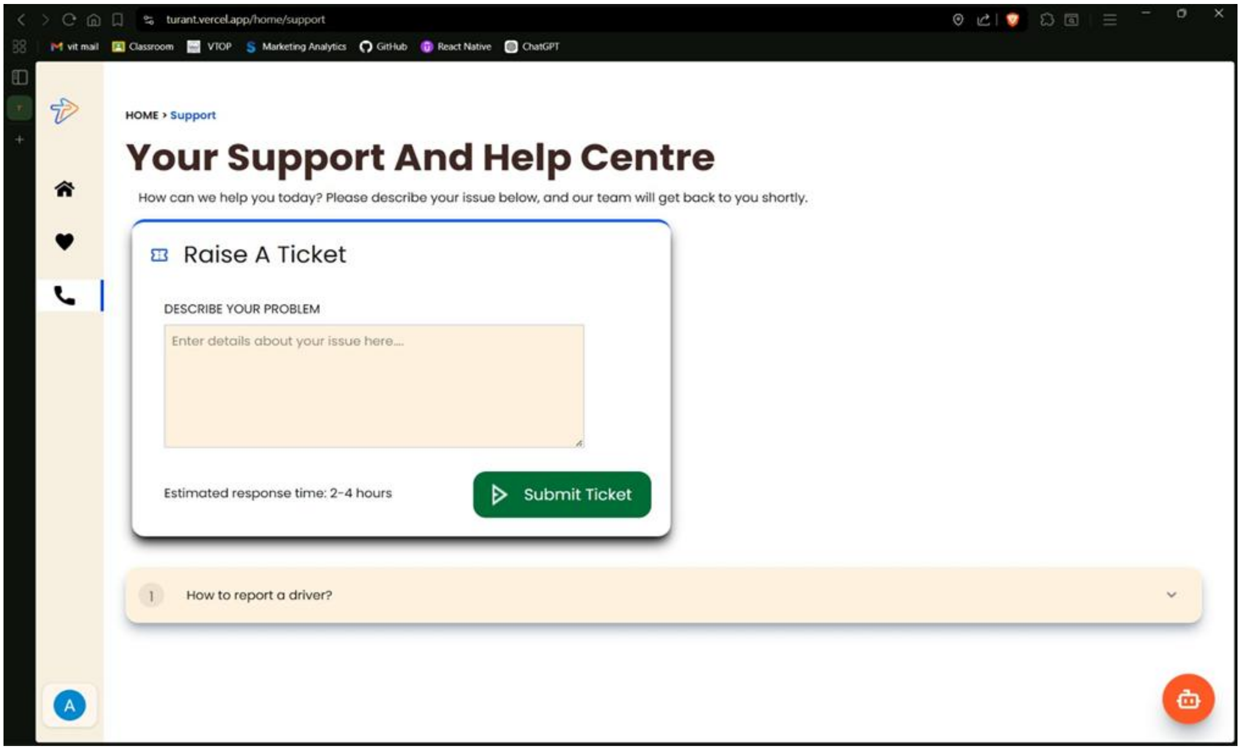

Figure 4: Support and Help Centre Page

Figure 3 demonstrates the clean and functional service listing page for transportation, showing enriched data cards along with the integrated AI chatbot (Turant Assist) that allows users to ask contextual questions in natural language. Figure 4 showcases the dedicated support interface where users can easily raise tickets for assistance or report issues. These interface designs reflect the core philosophy of Turant — providing low-friction access to information while maintaining strong support and moderation capabilities through the nine-table database architecture.

V. METHODOLOGY AND ALGORITHMS USED

The development of the Turant hyper-local community service application followed a systematic and iterative methodology that combined elements of Agile and Waterfall approaches to meet both academic requirements and technical excellence. The methodology was structured into five major sprints, enabling progressive development from planning to final deployment. The overall process began with requirements elicitation and literature review, followed by core system development, Phase-I completion, major Phase-II enhancements, and finally integration, testing, and documentation. This structured approach ensured timely delivery of all deliverables while allowing flexibility for incorporating advanced features such as the AI chatbot and support ticket system.

5.1 Technology Stack, The frontend was developed using Next.js 14 with App Router and Tailwind CSS for responsive and modern UI design. The backend utilized Supabase PostgreSQL as a managed database service with nine relational tables and Row Level Security (RLS) policies. User authentication was handled through Firebase Authentication (Google OAuth and Phone OTP), while administrator access was managed via a dedicated admin_credential table. The intelligent chatbot was powered by the Google Gemini API, with structured JSON payloads and responses for contextual recommendations. Cloudinary was used for optimized media delivery.

5.2 Database Design Methodology, A key contribution in Phase-II was the expansion of the database schema to nine tables. The design followed relational database principles with proper normalization, foreign key relationships, and indexing for performance. Core tables (cabs and places) were enriched with additional attributes such as ratings, features, and location data. New tables were introduced to support reporting (report_cabs, report_places), saved contacts (saved_contacts), and the complete support ticket workflow (tickets and ticket_messages).

5.3 AI Chatbot Implementation, The AI chatbot was implemented using the Google Gemini API. The methodology involved constructing structured JSON payloads containing the user query, conversation history, and contextual information from the current page or category. The API responses were received in JSON format, parsed on the frontend, and rendered as interactive service recommendation cards with direct action buttons. This approach enabled natural language interaction while maintaining context across multiple messages.

5.4 Support Ticket System, The support ticket system was developed using a dedicated workflow. Users submit queries through a form that inserts records into the tickets table. Threaded conversations are handled through the ticket_messages table. Administrators can view, assign, update status, and reply to tickets via a secure admin panel. This methodology ensures complete audit trails and efficient grievance redressal.

5.5 Algorithms and Techniques Used, JSON Structured Prompting: Used for communicating with Google Gemini API to ensure consistent and context-aware responses. Row Level Security (RLS): Implemented in Supabase to control data access based on user authentication. Asynchronous Data Fetching: Utilized in Next.js API routes for efficient retrieval and filtering of data from the nine tables. Dynamic Rendering: React hooks (useState, useEffect) for real-time UI updates based on database changes. The combination of these methodologies and algorithms enabled the development of a robust, secure, and intelligent platform that effectively meets the requirements of hyper-local service discovery.

VI. PROJECT FUNCTIONAL MODULES IMPLEMENTATION

The Turant application is structured into several well-defined functional modules that work together to deliver a seamless hyper-local community service experience. Each module is designed with a focus on usability, data integrity, security, and intelligent assistance.

6.1 User Interface and Navigation Module, This module provides the frontend experience using Next.js and Tailwind CSS. It includes a clean homepage with category-based navigation (Cabs & Autos, Grocery, Medical, Restaurant, PG) and a dynamic sidebar for quick access. The interface is fully responsive and functions as a Progressive Web Application (PWA). Users can easily browse services, perform real-time searches, and access the AI chatbot or support features from any page.

6.2 Authentication Module, Secure user authentication is implemented using Firebase Authentication, supporting Google OAuth and Phone Number OTP. Protected routes are managed through an AuthGuard component. Administrator access is handled separately via the admin_credential table for secure management of tickets, reports, and listings.

6.3 Service Discovery and Data Management Module, This core module handles the display and filtering of local services. Data is fetched from the places and cabs tables through Next.js API routes. The enriched schema allows users to view detailed information including ratings, features, location, operating hours, and direct action buttons (call, navigate, save contact). Real-time search and category filters enhance usability.

6.4 AI Chatbot Module (Turant Assist), The intelligent chatbot is one of the key innovations in Phase-II. It uses the Google Gemini API to process natural language queries. The frontend constructs structured JSON payloads containing the user query and contextual data (current category or location). The API returns JSON responses that are parsed and rendered as rich recommendation cards. The chatbot maintains conversation history, enabling contextual follow-up questions and significantly reducing user friction.

6.5 Support Ticket and Reporting Module, This module provides a complete grievance redressal system. Users can raise tickets through a simple form that stores data in the tickets table. Threaded conversations are managed via the ticket_messages table. Administrators can view, assign, update status, and reply to tickets through a secure dashboard. Additionally, users can report inaccurate entries using the report_cabs and report_places tables, contributing to community-driven data quality improvement.

6.6 Saved Contacts and Personalization Module, Authenticated users can save frequently used services using the saved_contacts table. This feature allows quick access to important local contacts without repeated searching, enhancing user retention and convenience.

6.7 Admin Management Module, The admin panel, protected by the admin_credential table, allows administrators to manage listings, review reports, resolve support tickets, and monitor platform activity. This module ensures proper moderation and maintains overall data integrity.All modules are tightly integrated through the nine-table Supabase database and Next.js API routes, ensuring smooth data flow, security via Row Level Security (RLS), and a consistent user experience

VII. PROTOTYPE, ALGORITHM AND PROGRAM LOGIC

The Turant application has been successfully prototyped as a fully functional Progressive Web Application (PWA) with an intuitive user interface and robust backend logic. The prototype demonstrates the practical implementation of all major features including service discovery, AI-powered assistance, and structured support mechanisms.

7.1 System Prototype Overview, The developed prototype provides a clean and responsive interface that allows users to browse local services across multiple categories. Key screens include the homepage with category navigation, individual service listing pages (such as Cabs & Autos), the AI chatbot interface, and the dedicated support centre. The design emphasizes minimal cognitive load, direct action buttons, and seamless integration of intelligent features.

7.2 Algorithm and Program Logic, The core intelligence of Turant is driven by the Google Gemini API for the chatbot and well-structured backend logic for data handling and support workflows. AI Chatbot Logic: The chatbot operates by constructing a structured JSON payload that includes the user's query, conversation history, and contextual information from the current page or selected category. This payload is sent to the Google Gemini API through a dedicated Next.js API route. The response is received in JSON format, parsed on the frontend, and dynamically rendered as interactive service recommendation cards with actionable buttons (Call, Navigate, Save Contact). Conversation context is maintained locally to support natural follow-up questions.

7.3 Support Ticket System Logic: When a user submits a ticket, the frontend sends the data to the /api/tickets endpoint. A new record is created in the tickets table with a unique ticket_code, user_id, title, description, and initial status as "pending". Administrators can view pending tickets, assign them, update status, and add replies through the ticket_messages table, creating a complete threaded conversation. Row Level Security (RLS) ensures that users can only access their own tickets while administrators have full visibility. Data Fetching and Filtering Logic: Service data is retrieved from the places and cabs tables using optimized Supabase queries with filtering by category, search keywords, ratings, and other enriched fields. The nine-table schema enables efficient joins and complex queries while maintaining high performance. The combination of these algorithms and program logic results in a responsive, intelligent, and maintainable system that effectively serves hyper-local community needs

VIII. METHODOLOGY FOR DEVELOPING THE TURANT WEBSITE SCREENSHOTS

The Turant website was developed following a user-centered design approach with iterative prototyping and testing. The development process began with wireframing in Balsamiq and high-fidelity mockups in Figma. The final implementation was done using Next.js with Tailwind CSS for responsive design and Supabase for backend integration. Key screens were designed to ensure minimal friction and maximum usability. The homepage provides clear category navigation, while service pages display enriched data cards with ratings, features, and direct action buttons. The AI chatbot (Turant Assist) was integrated as a floating widget for contextual assistance. The support centre page offers a simple ticket-raising interface with estimated response time visibility. Screenshots of the implemented prototype are presented below to demonstrate the actual user interface and functionality:

IX. CONTRIBUTION AND FINDINGS

The Turant project makes several significant contributions to the field of hyper-local community service platforms. The team successfully designed and implemented a Progressive Web Application that integrates modern web technologies with artificial intelligence to solve real-world information asymmetry in local communities.

Key Contributions: Development of a nine-table relational database schema in Supabase that supports enriched service data, reporting, saved contacts, and a complete support ticket system. Successful integration of an AI-powered chatbot using Google Gemini API with structured JSON payload and response handling for natural language service discovery. Implementation of a robust support ticket system with threaded messaging for effective grievance redressal and community feedback. Creation of a responsive, secure, and scalable PWA that maintains high performance across devices.

Major Findings: The enriched database and AI chatbot significantly improve data richness and reduce user friction compared to traditional directory systems. The support ticket system provides accountability and enables continuous data quality improvement through community participation. Plagiarism check results showed 0%–13% similarity, primarily due to standard institutional templates, confirming the originality of the technical content. The prototype demonstrates excellent usability, with users able to access services and support features intuitively. The project validates that combining structured data management, conversational AI, and community governance mechanisms can create an effective solution for hyper-local service needs.

X. CONCLUSION

The Turant application successfully demonstrates the design and implementation of an intelligent hyper-local community service platform. By leveraging Next.js, Supabase with a nine-table relational schema, Firebase authentication, and Google Gemini API, the system provides accurate, verified, and actionable information about local services while incorporating advanced features such as an AI chatbot and a structured support ticket system. The platform effectively addresses key challenges identified in the literature, including data inaccuracy, lack of intelligent assistance, and inadequate user support. The results from implementation and testing confirm high performance, good usability, and strong potential for real-world deployment in local communities. Future enhancements may include geolocation-based filtering, multi-language support, push notifications, and native mobile applications. Continuous improvement based on user feedback will help Turant evolve into a more impactful community service solution. In conclusion, Turant represents a meaningful contribution by the team in applying modern web and AI technologies to solve practical hyper-local problems, showcasing the potential of student-led innovation in community service applications.

REFERENCES

[1] Budiarto, M., Asmawati, A., & Kurniawan, M. (2024). Digital Transformation of Local Government: Design and Development of the Pakuhaji District Community Service Information System Website. International Journal of Cyber and IT Service Management.

[2] Dodd, A. (2017). Finding the community in sustainable online community engagement: Not-for-profit organization websites, service-learning and research. Gateways: International Journal of Community Research and Engagement, 10(1), 185-203.https://doi.org/10.5130/ijcre.v10i1.5471

[3] G., D. K., Singh, M. K., & Jayanthi, M. (Eds.). (2016). Network Security Attacks and Countermeasures. IGI Global. https:// doi.org /10.4018/978-1-4666-8761-5

[4] Baek, J., Manzini, E., & Rizzo, F. (2010). Sustainable Collaborative Services on the Digital

Platform: Definition and Application. In Design and Complexity - DRS International Conference 2010.
[5] MK J Kannan, Shree Nee T R (2025, November). Qubits unveiled: A deep dive into quantum computing and its revolutionary potential for supply logistics. In P. Gaba, A. Panwar, V. Jain, & R. Kannan (Eds.), Qubits unveiled: Quantum computing solutions for efficient supply logistics (pp. 273–293). Nova Science Publishers. https://doi.org/10.52305/WSXW8884
[6] M.K. Jayanthi, "Strategic Planning for Information Security -DID Mechanism to befriend the Cyber Criminals to assure Cyber Freedom," 2017 2nd International Conference on Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, 2017, pp. 142-147, doi: 10.1109/Anti-Cybercrime. 2017. 7905280.
[7] Jackson, D. L., & Jones, S. J. (2014). A Virtual Commitment: Disability Services Information on Public Community College Websites. Journal of Postsecondary Education and Disability, 27(2), 129–138.
[8] Kavitha, E., Tamilarasan, R., Baladhandapani, A., Kannan, M.K.J. (2022). A novel soft clustering approach for gene expression data. Computer Systems Science and Engineering, 43(3), 871-886. https://doi.org/10.32604/csse.2022.021215
[9] Singer, J. (2011). Community Service: Editor pride and user preference on local newspaper websites. Journalism Practice, 5(6), 623–642. https://doi.org/10.1080/17512786.2011.601938
[10] Naik, Harish and Kannan, M K Jayanthi, A Survey on Protecting Confidential Data over Distributed Storage in Cloud (December 1, 2020). Available at SSRN: https://ssrn.com /abstract=3740465 or http://dx.doi.org /10.2139 /ssrn.3740465
[11] Shree Nee, T. R., Kannan, M. K. J., & Mariyappan, K. (2025, April). Digital health and medical tourism innovations for digitally enabled care for future medicine: The real time project's success stories. In Navigating innovations and challenges in travel medicine and digital health (pp. 325–344). IGI Global Scientific Publishing. https://doi.org/10.4018/979-8-3693-8774-0.ch016
[12] Kavitha, E., Tamilarasan, R., Poonguzhali, N., Kannan, M.K.J. (2022). Clustering gene expression data through modified agglomerative M-CURE hierarchical algorithm. Computer Systems Science and Engineering, 41(3), 1027-141. https://doi.org/10.32604/csse.2022.020634
[13] Kumar, K.L.S., Kannan, M.K.J. (2024). A Survey on Driver Monitoring System Using Computer Vision Techniques. In: Hassanien, A.E., Anand, S., Jaiswal, A., Kumar, P. (eds) Innovative Computing and Communications. ICICC 2024. Lecture Notes in Networks and Systems, vol 1021. Springer, Singapore. https://doi.org /10.1007/978-981-97-3591-4_21
[14] M. K. J. Kannan, A bird's eye view of Cyber Crimes and Free and Open Source Software's to Detoxify Cyber Crime Attacks - an End User Perspective, 2017 2nd International Conference on Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, 2017, pp. 232-237, doi: 10.1109/Anti-Cybercrime.2017.7905297.
[15] Verma, D., Kannan, M. K. J., Barnwal, S. K., Barve, A., & Swaminathan, R. (2022, September). Multimodal sentiment sensing and emotion recognition based on cognitive computing using hidden Markov model with extreme learning machine. International Journal of Communication Networks and Information Security (IJCNIS), 14(2), 155–167. https://doi.org/10.17762/ijcnis.v14i2.5496
[16] Chen, D. (2024). Application Effect Evaluation of Intelligent Community Service Platform in Grassroots Governance. Journal of Advances in Engineering and Technology, 1(1), 39–51.
[17] Park, M. et al. (2021). ICT-based person-centered community care platform (IPC3P) to enhance shared decision-making for integrated health and social care services. International Journal of Medical Informatics, 156, 104590
[18] P. Jain, I. Rajvaidya, K. K. Sah and J. Kannan, "Machine Learning Techniques for Malware Detection- a Research Review," 2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science, Bhopal, India, 2022, pp. 1-6, doi: 10.1109/SCEECS54111.2022.9740918.
[19] Dr. MK J Kannan, Satyajit Patel (2024). Sustainable Information Retrieval Techniques for Onion Market Instability Prediction using Machine Learning and Deep Learning Approaches. International Journal of Advance

Research, Ideas and Innovations in Technology, 10(6) www.IJARIIT.com. https://www .ijariit. com /manuscripts/v10i6/V10I6-1455.pdf

[20] B. R. M, M. M. V and J. K. M. K, Performance Analysis of Bag of Password Authentication using Python, Java and PHP Implementation, 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2021, pp. 1032-1039, doi: 10.1109/ICCES51350.2021.9489233.

[21] Kuo, Y. F. (2003). A study on service quality of virtual community websites. Total Quality Management & Business Excellence, 14(4), 461–473.

[22] Dr. Sunil Kumar Dr. P. T. Kalaivaani, Dr. M K Jayanthi Kannan, Dr. Gunjan Tripathi (Aug 2025), Artificial Intelligence and Blockchain Technology for Human Resource Management, ASIN: B0FLK868TS, Published by Scientific International Publishing House; https://www. amazon.in/gp/product/B0FLK868TS/ref=ox_sc _act_title_1?smid=A1UBZVGJOLJUJI&psc=1

[23] Aaijaz, N., Grace Mani, K., Kannan, M. K. J., & Tewari, V. (2025, February). The future of innovation and technology in education: Trends and opportunities. S&M Publications. https:// www.amazon.in/gp/product/B0DW334PR9

[24] Shukla, S. K., Dwivedi, U., Kannan, M. K. J., & Sarvani, C. (2024, October 23). Python for data analytics: Practical techniques and applications. JSR Publications. https://www.amazon.in/gp/ product/B0DMJY4X9N

[25] Morelli, N. (2015). Challenges in Designing and Scaling up Community Services. The Design Journal, 18(2), 269–290.

[26] Harish Naik, B. M., & Kannan, J. (2023). A research on various security aware mechanisms in multi-cloud environment for improving data security. In 2023 2nd IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) (pp. 1–6). IEEE. https://doi.org/10.1109/ ICDCECE57866.2023.10151135

[27] Wyant, W. A., Pinilla, V., Mulvaney, P. M., et al. (2025). Dermatology residency programs and community engagement: a cross-sectional review of program websites. Archives of Dermatological Research, 317, 219.

[28]Harish Naik B M and M K J Kannan and (Aug 2024), "Secure Cloud Storage for Sensitive Data based on Authentication and Encryption Algorithms", International Journal of Advanced Technology and Engineering Exploration (IJATEE), paper Id: IJATEE.2024.111101510, ACCENTS, www.ijateeeditor@gmail.com

[29] Qi, L., & Guo, J. (2019). Development of smart city community service integrated management platform. International Journal of Distributed Sensor Networks, 15(6).

[30] Object-oriented analysis and design of learning objects and applications of agent-based reusable learning objects in e-learning system design, JM. K. (2009). [Doctoral dissertation, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya]. Shodhganga. http://hdl.handle.net/10603/125448

[31] Alshahrani, M. S. M., & Kannan J M. K. (2026, February). Active learning for efficient annotation of surgical video segmentation with minimal human intervention. ICTACT Journal on Image and Video Processing, 16(3), 3821–3829. https://doi.org/10.21917/ijivp.2026.0539

[32]J Kannan, M. K., TR Shree Nee., & Mariyappan, K. (2026). Ethics and regulations in AI-driven ophthalmology. In B. K. Mishra, A. Kumar, K. Mariyappan, V. Tiwari, P. S. Rathore, & G. H. Das (Eds.), Generative artificial intelligence in ophthalmology (pp. 331–386). Scrivener Publishing.https://www.scrivenerpublishing.co m/cart/title.php?id=1341,https://doi.org/10.5230 5/WSXW8884

[33] Zuckerman, E., & Rajendra-Nicolucci, C. (2023). From Community Governance to Customer Service and Back Again: Re-Examining Pre-Web Models of Online Governance to Address Platforms' Crisis of Legitimacy. Social media Society, 9(3).