Arogya Mitra: An AI-Powered Mobile Health Assistant Application using Google Gemini API and React Native

Aniket Sargar, Harshvardhan Tapkir, Dhanush
Department of Computer Applications (AIML)
Ajeenkya DY Patil University, Pune, Maharashtra, India
Guide: Prof. **Vivek More**
Corresponding Author: aniketsargar@example.com

How to Cite this Article:

Sargar, A., Tapkir, H. & Dhanush, (2026). Arogya Mitra: An AI-Powered Mobile Health Assistant Application using Google Gemini API and React Native. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(04). https://doi.org/10.55041/ijcope.v2i4.420

https://doi.org/10.55041/ijcope.v2i4.420

Abstract

The healthcare sector in India faces a critical challenge of accessibility, with a doctor-to-patient ratio of approximately 1:1456, significantly below the World Health Organization's recommended standard. This paper presents Arogya Mitra, an AI-powered mobile health assistant application designed to bridge the healthcare information gap through intelligent conversational interfaces. Built using React Native with Expo framework for the frontend, Node.js/Express.js for the backend, MongoDB Atlas for data persistence, and Google Gemini API for natural language understanding and response generation, Arogya Mitra provides a comprehensive suite of health management features including AI-driven health consultations, symptom assessment, location-based doctor discovery via Google Maps, prescription digitization through camera-based scanning with ML Kit barcode recognition, and medication adherence support through scheduled push notifications. The system architecture follows a three-tier client-server model with RESTful API communication, JWT-based authentication, and Firebase Cloud Messaging for notification delivery. The application was evaluated through multi-level testing including unit, integration, system, performance, and security testing, followed by User Acceptance Testing (UAT) with 30 participants from diverse demographics. UAT results demonstrated an overall satisfaction score of 4.17 out of 5.00 (83.4%), with the AI chat quality and doctor finder features receiving the highest ratings. Performance benchmarks showed an average Gemini API response time of 2.1 seconds, cold start time of 2.8 seconds, and API success rate of 98.7%. The paper discusses the system's architecture, implementation methodology, testing outcomes, limitations, and future enhancements including multi-language support and telemedicine integration.

Keywords: *Health Chatbot, Artificial Intelligence, Google Gemini API, React Native, Mobile Health (mHealth), Natural Language Processing, Symptom Assessment, MongoDB, Firebase, Healthcare Accessibility*

I. INTRODUCTION

Healthcare accessibility remains one of the most pressing challenges in developing nations, particularly in India, where the disparity between urban and rural healthcare infrastructure is stark. According to the National Health Profile 2021, India has approximately 1.4 physicians per 1,000 population, with the distribution heavily skewed toward metropolitan centers. Rural and semi-urban areas, home to approximately 65% of India's population, face acute shortages of qualified medical professionals, diagnostic facilities, and pharmacies [1].

The rapid proliferation of smartphones in India—exceeding 750 million users by 2025—has created a transformative opportunity to leverage mobile technology for healthcare delivery. Mobile health (mHealth) applications have emerged as viable tools for providing health information, facilitating remote consultations, and supporting preventive healthcare. The global digital health market, projected to reach $660 billion by 2025, reflects the growing recognition of technology's role in healthcare transformation [2].

The advent of large language models (LLMs) and advanced natural language processing (NLP) systems has further accelerated the potential of health chatbots. Unlike rule-based chatbots that rely on predefined decision trees, LLM-powered chatbots can understand contextual nuances, maintain conversational continuity, and generate human-like responses that make health information more accessible and comprehensible to lay users [3]. Google's Gemini API, with its multimodal understanding capabilities and healthcare-specific safety filters, represents a significant advancement in the tools available for building intelligent health assistants.

This paper presents Arogya Mitra (आरोग्य मित्र, meaning "Health Friend" in Hindi/Marathi), an AI-powered mobile health assistant application that integrates multiple healthcare functionalities into a unified platform. Arogya Mitra combines an LLM-driven conversational health assistant with practical health management tools including symptom assessment, location-based doctor discovery, prescription digitization, and medication adherence tracking. The application targets Android devices and is designed for the Indian healthcare context, with an emphasis on accessibility, affordability, and user-friendliness.

The key contributions of this paper are: (a) the design and implementation of a comprehensive mobile health assistant integrating Google Gemini API with healthcare-specific prompt engineering; (b) the integration of five distinct health management modules (chat, symptoms, doctor finder, prescription scanner, medicine reminders) within a single mobile application; (c) empirical evaluation through multi-level testing and user acceptance testing with 30 participants; and (d) a detailed analysis of technical challenges, performance metrics, and future directions for AI-powered health assistants in the Indian context.

II. RELATED WORK

A. AI Chatbots in Healthcare

Laranjo et al. [4] conducted a systematic review of conversational agents in healthcare, analyzing 17 systems across symptom assessment, health education, and medication adherence. Their findings indicated that chatbots were effective for structured health interactions but noted limitations in handling complex, multi-symptom presentations. Vaidyam et al. [5] further demonstrated the efficacy of AI chatbots in mental health screening, achieving comparable sensitivity to standardized clinical assessments.

Commercial health chatbots such as Ada Health, Babylon Health, and WebMD Symptom Checker have demonstrated varying approaches to AI-driven health assistance. Ada Health employs a probabilistic reasoning engine for symptom assessment, while Babylon Health combines AI triage with telemedicine consultations. However, these solutions are predominantly designed for Western healthcare contexts and are either subscription-based or not optimized for the affordability and accessibility constraints prevalent in developing nations [6].

B. Large Language Models in Healthcare

The application of LLMs in healthcare has been explored extensively since the introduction of transformer-based architectures. Singhal et al. [7] introduced Med-PaLM, a medical domain-adapted LLM that achieved expert-level performance on medical question-answering benchmarks. Their work demonstrated that with appropriate prompt engineering and domain-specific fine-tuning, general-purpose LLMs can provide clinically relevant health information.

Google's Gemini models, successors to PaLM, introduced multimodal capabilities that enable processing of text, images, and structured data within a single model [8]. For health chatbot applications, Gemini's advanced reasoning capabilities, contextual understanding, and built-in safety filters make it particularly suitable for generating health information that is both informative and appropriately cautious.

C. mHealth Applications in India

The Indian mHealth landscape has seen significant growth with applications such as Practo (doctor discovery and appointment booking), 1mg (medicine delivery and health information), and Aarogya Setu (COVID-19 contact tracing). Cho et al. [9] studied user adoption of mHealth applications in developing countries and identified ease of use, perceived usefulness, and trust as primary adoption factors. Kumar et al. [10] demonstrated that mobile-based medicine reminder systems improved medication adherence by 23% compared to traditional methods, providing empirical support for the inclusion of reminder features in health applications.

D. Research Gap

Despite the proliferation of health applications and AI chatbots, a significant gap exists in the availability of a unified platform that combines LLM-powered health conversations, symptom assessment, doctor discovery with maps integration, prescription scanning, and medication reminders—all within a single, free, mobile application optimized for the Indian context. Arogya Mitra addresses this gap by integrating these functionalities using modern technologies and the advanced Gemini API.

III. SYSTEM ARCHITECTURE AND DESIGN

A. Overall Architecture

Arogya Mitra employs a three-tier client-server architecture comprising the Presentation Layer, Application Layer, and Data Layer. This separation ensures modularity, scalability, and maintainability.

The Presentation Layer is implemented as a React Native mobile application using the Expo managed workflow, targeting Android SDK 24+ (Android 7.0 Nougat). The Application Layer consists of a Node.js/Express.js RESTful API server that processes client requests, implements business logic, and manages communication with external services. The Data Layer comprises MongoDB Atlas for persistent storage and Firebase for authentication state management and push notification delivery.

[Fig. 1: System Architecture Diagram – To be inserted]

B. Technology Stack

Layer	Technology	Purpose
Frontend	React Native + Expo	Cross-platform mobile UI
Backend	Node.js + Express.js	RESTful API server
Database	MongoDB Atlas	NoSQL document storage
AI Engine	Google Gemini API	NLU and response generation
Authentication	Firebase Auth + JWT	User authentication
Notifications	Firebase Cloud Messaging	Push notifications
Maps	Google Maps SDK	Location-based services
ML	Google ML Kit	Barcode/QR scanning
Camera	Expo Camera	Prescription image capture

Table I: Technology Stack

C. Database Design

The MongoDB database employs four primary collections: Users (profile, authentication, health information), Chats (conversation sessions with embedded message arrays), Reminders (medication schedules with adherence logs), and Prescriptions (image references with metadata). The document-oriented model is particularly suited for storing chat conversations, which have variable structures and nested data. Indexes on userId, email, and compound indexes on (userId, createdAt) optimize query performance for the most frequent operations.

D. API Design

The backend exposes 18 RESTful API endpoints organized into four modules: Authentication (5 endpoints for registration, login, profile management, and password reset), Chat (5 endpoints for session management and message processing), Reminders (5 endpoints for CRUD operations and adherence logging), and Prescriptions (3 endpoints for storage and retrieval). All endpoints except registration and login require JWT authentication, enforced through Express middleware.

IV. IMPLEMENTATION

A. AI Chat Module

The core AI functionality is powered by the Google Gemini API, accessed through RESTful HTTPS requests from the backend server. The implementation involves three critical components:

1) Prompt Engineering: The system prompt was engineered through iterative testing with over 200 sample health queries across categories including general health, chronic conditions, emergency symptoms, mental health, nutrition, and fitness. The prompt instructs the model to act as a knowledgeable health assistant, provide evidence-based information in accessible language, use severity indicators for symptoms, include medical disclaimers, and redirect emergencies to appropriate services (112 in India).

2) Context Management: To maintain conversational continuity, the last 10 messages from the active session are included in each API request. This sliding-window approach balances context awareness with API token limits and cost efficiency.

3) Safety Mechanisms: Every health-related response is appended with a standardized medical disclaimer. The Gemini API's built-in safety filters prevent generation of harmful content, and additional keyword-based filtering on the backend detects and handles requests for emergency situations, self-harm information, or prescription of specific medications.

B. Symptom Assessment Module

The symptom checker implements a guided, multi-step assessment flow. Users first select the affected body area from a visual body map interface, then choose applicable symptoms from a categorized list. Follow-up questions gather contextual information including duration, severity (using a slider control), frequency, and associated symptoms. The collected symptom data is formatted into a structured prompt and sent to the Gemini API for analysis. The AI-generated assessment presents possible conditions ranked by relevance with severity indicators (color-coded: green for mild, orange for moderate, red for urgent) and recommended next steps, including a direct link to the Doctor Finder for relevant specialists.

C. Doctor Finder Module

The doctor finder integrates Google Maps SDK for React Native with the Google Places API. The module detects the user's current location using expo-location with high-accuracy GPS and displays nearby healthcare facilities as categorized markers on an interactive map. Filter chips enable filtering by facility type (hospitals, clinics, pharmacies) and medical specialty (12 categories). A bottom sheet component displays facility details including name, address, rating, operating hours, phone number, and distance (calculated using the Haversine formula). Navigation integration launches the device's native maps application for turn-by-turn directions.

D. Prescription Scanner Module

The prescription scanner uses expo-camera for image capture with a guided overlay frame, followed by interactive cropping via CropImageActivity. Captured images are compressed (JPEG at 70% quality, maximum 1MB) and stored locally using expo-file-system. ML Kit's Barcode Scanner API provides on-device barcode and QR code recognition for medicine identification. The module maintains a searchable gallery of stored prescriptions with metadata including date, doctor name, and user-defined tags.

E. Medication Reminder Module

The reminder system employs a dual notification strategy. Local notifications are scheduled using Expo Notifications API for immediate, on-device reminders. Firebase Cloud Messaging (FCM) provides server-side notification delivery for reliability when the application is in the background or terminated. A backend cron job checks for due reminders every minute and dispatches FCM messages to registered device tokens. Adherence is tracked through notification responses (taken, snoozed, dismissed) logged to the database.

F. Security Implementation

Security measures include HTTPS (TLS 1.2+) for all API communication, bcrypt password hashing (10 salt rounds), JWT-based session management (HS256, 24-hour expiry), input sanitization against NoSQL injection and XSS attacks, rate limiting (100 requests per 15 minutes per IP), and file upload validation (type, size, magic byte verification). MongoDB Atlas enforces IP whitelisting and TLS connections.

V. EVALUATION AND RESULTS

A. Testing Methodology

The application was evaluated through five levels of testing: unit testing (15 test cases covering all modules), integration testing (8 test cases verifying inter-module communication), system testing (6 end-to-end scenarios), performance testing (response time, resource consumption, network resilience), and security testing (6 vulnerability assessments). All test levels achieved pass rates exceeding 95%.

B. Performance Metrics

Metric	Value	Benchmark
APK Size	156 MB	< 100 MB (optimization planned)
Cold Start Time	2.8 sec	< 3 sec
Gemini API Response	2.1 sec (avg)	< 5 sec
API Success Rate	98.7%	> 95%
Memory Usage (idle)	120 MB	< 200 MB
Memory Usage (active)	180 MB	< 256 MB
Battery Drain (30 min)	3.2%	< 5%
Crash Rate	0.5%	< 1%

Table II: Application Performance Metrics

C. User Acceptance Testing

UAT was conducted with 30 participants selected to represent diverse demographics: 15 students (age 18-25), 10 working professionals (age 26-45), and 5 senior citizens (age 50+). Participants used the application for a minimum of 20 minutes, interacting with all major features, and subsequently completed a structured questionnaire rating each feature on a 5-point Likert scale (1=Very Poor to 5=Excellent).

Evaluation Criteria	Mean Score	Std. Dev.
Ease of Use	4.10	0.84
AI Chat Response Quality	4.03	0.89
Symptom Checker Utility	3.87	0.97
Doctor Finder Utility	4.20	0.76

Evaluation Criteria	Mean Score	Std. Dev.
Prescription Scanner	3.93	0.91
Medicine Reminders	4.07	0.83
Visual Design	4.07	0.78
Response Speed	3.93	0.87
Overall Satisfaction	4.17	0.79

Table III: User Acceptance Testing Results (n=30)

The overall satisfaction score of 4.17/5.00 (83.4%) indicates strong user acceptance. The Doctor Finder received the highest rating (4.20), reflecting the practical value of location-based healthcare facility discovery. The Symptom Checker received the lowest rating (3.87), with participants noting that the body area selection interface was less intuitive on smaller screen devices. Qualitative feedback highlighted requests for multi-language support (mentioned by 73% of participants), voice input capability (60%), and integration with wearable health devices (47%).

D. Comparative Analysis

Feature	Ada Health	Practo	Arogya Mitra
AI Chatbot	Symptom-only	No	Full health chat
Symptom Checker	Yes	No	Yes
Doctor Finder + Maps	No	Limited	Full integration
Prescription Scanner	No	No	Yes (Camera + ML Kit)
Medicine Reminders	No	Yes	Yes (FCM + Local)
Free Access	Partial	Partial	Fully free
Indian Context	No	Yes	Yes
Offline Support	No	No	Partial
LLM-Powered	No	No	Yes (Gemini)

Table IV: Feature Comparison with Existing Applications

E. Device Compatibility

The application was tested on 8 Android devices spanning Android versions 10–14, screen sizes 6.4”–6.67”, and RAM configurations 2–8 GB. Full functionality was confirmed on all devices with 4 GB+ RAM. Devices with 2 GB RAM exhibited slower cold start times and map rendering lag but maintained core chat functionality.

VI. DISCUSSION

A. Key Findings

The development and evaluation of Arogya Mitra yield several notable findings. First, the Google Gemini API, when combined with domain-specific prompt engineering, produces health responses that users found helpful and trustworthy (4.03/5.00 rating), validating the approach of using general-purpose LLMs with healthcare-specific prompting rather than building custom medical NLP models. Second, the integration of five distinct health modules within a single application was technically feasible using the React Native/Expo stack and was positively received by users who valued having a unified health management tool. Third, the doctor finder with Google Maps integration received the highest user satisfaction score, suggesting that location-based healthcare discovery is a high-value feature in the Indian context where finding nearby facilities remains a practical challenge.

B. Limitations

The study has several limitations. The APK size of 156 MB exceeds recommended limits, primarily due to the inclusion of multiple native modules (Maps, Camera, ML Kit, Firebase). This can be addressed through ProGuard/R8 code shrinking and migration to Android App Bundle (AAB) format. The Gemini API free tier imposes rate limits (60 requests/minute) that would be insufficient for production-scale deployment. The UAT sample size (n=30) is modest; a larger-scale evaluation would provide more statistically robust results. The application currently supports English only, which limits accessibility for non-English-speaking users in India. Additionally, the application does not provide medical diagnoses and relies on AI-generated health information that, while generally accurate, may occasionally produce responses that require clinical validation.

C. Ethical Considerations

Health AI applications raise important ethical considerations regarding accuracy, bias, privacy, and user reliance. Arogya Mitra addresses these through: (a) mandatory medical disclaimers on all AI responses explicitly stating that the information is educational and not a substitute for professional medical consultation; (b) safety filters that prevent the AI from diagnosing conditions, prescribing medications, or providing information about self-harm; (c) data encryption in transit and at rest for health information; and (d) compliance with the Indian IT Act 2000 for health data handling. The application is positioned as a health information tool that complements rather than replaces professional healthcare services.

VII. FUTURE WORK

Future development of Arogya Mitra will focus on the following enhancements:

• Multi-language Support: Integration of Hindi, Marathi, and other regional languages using Gemini's multilingual capabilities to expand accessibility to non-English-speaking populations.

• Voice Input: Addition of speech-to-text functionality to improve accessibility for elderly users and individuals with visual impairments.

• Telemedicine Integration: Enabling video consultations with registered doctors for cases requiring professional assessment beyond AI capabilities.

• Wearable Device Integration: Connecting with fitness bands and smartwatches to incorporate real-time health data (heart rate, step count, sleep patterns) into AI-generated health insights.

• Multimodal AI Analysis: Leveraging Gemini's multimodal capabilities for analyzing medical images (skin conditions, rashes) to provide preliminary visual health assessments.

• APK Optimization: Implementing ProGuard/R8 minification, AAB format, Hermes JavaScript engine, and dependency audit to reduce APK size to under 50 MB.

• Large-Scale Clinical Validation: Conducting a controlled study with healthcare professionals to evaluate the clinical accuracy and safety of AI-generated health responses.

• Government Health Scheme Integration: Connecting with Ayushman Bharat Digital Mission (ABDM) health IDs for seamless healthcare access.

VIII. CONCLUSION

This paper presented Arogya Mitra, an AI-powered mobile health assistant application that addresses the healthcare accessibility gap in India by integrating Google Gemini API-based conversational health assistance with practical health management tools. The application combines five core health modules—AI chat, symptom assessment, doctor discovery, prescription scanning, and medication reminders—within a single, freely accessible Android application.

The technical implementation demonstrates the viability of using modern web technologies (React Native, Node.js, MongoDB) combined with advanced AI services (Google Gemini) to build comprehensive health applications. The three-tier architecture provides a scalable and maintainable foundation for future enhancements.

User acceptance testing with 30 participants yielded an overall satisfaction score of 4.17/5.00 (83.4%), validating the application's design and functionality. Performance benchmarks confirmed acceptable response times, resource consumption, and reliability across diverse Android devices.

While the current version has limitations including APK size optimization needs, English-only support, and free-tier API constraints, Arogya Mitra establishes a solid foundation for a production-quality health assistant. The planned enhancements—multi-language support, telemedicine integration, and multimodal AI capabilities—will further expand the application's reach and impact in making healthcare information accessible to underserved populations in India.

. ACKNOWLEDGMENT

The authors gratefully acknowledge the guidance and support of Prof. Vivek More, Department of Computer Applications (AIML), Ajeenkya DY Patil University, Pune. The authors also thank Google for providing access to the Gemini API and the open-source community for the React Native, Expo, and MongoDB ecosystems that made this project possible.

. REFERENCES

[1] National Health Profile 2021, Central Bureau of Health Intelligence, Ministry of Health and Family Welfare, Government of India, 2021.

[2] Grand View Research, "Digital Health Market Size, Share & Trends Analysis Report," 2023. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/digital-health-market

[3] A. Vaswani et al., "Attention is All You Need," in Proc. 31st Int. Conf. Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.

[4] L. Laranjo et al., "Conversational agents in healthcare: a systematic review," Journal of the American Medical Informatics Association, vol. 25, no. 9, pp. 1248–1258, 2018.

[5] A. N. Vaidyam et al., "Chatbots and Conversational Agents in Mental Health: A Review of the Psychiatric Landscape," The Canadian Journal of Psychiatry, vol. 64, no. 7, pp. 456–464, 2019.

[6] S. Gerke, T. Minssen, and G. Cohen, "Ethical and legal challenges of artificial intelligence-driven healthcare," Artificial Intelligence in Healthcare, pp. 295–336, 2020.

[7] K. Singhal et al., "Large language models encode clinical knowledge," Nature, vol. 620, pp. 172–180, 2023.

[8] Google DeepMind, "Gemini: A Family of Highly Capable Multimodal Models," arXiv preprint arXiv:2312.11805, 2023.

[9] H. Cho et al., "Factors influencing mHealth adoption in developing countries: A systematic review," mHealth, vol. 6, p. 24, 2020.

[10] S. Kumar et al., "Effectiveness of mobile-based medication reminder systems: A systematic review and meta-analysis," PLoS ONE, vol. 16, no. 3, e0248451, 2021.

[11] React Native Documentation, Meta Platforms, Inc. [Online]. Available: https://reactnative.dev/docs/getting-started

[12] Expo Documentation. [Online]. Available: https://docs.expo.dev/

[13] Google Gemini API Documentation. [Online]. Available: https://ai.google.dev/docs

[14] MongoDB Documentation. [Online]. Available: https://www.mongodb.com/docs/

[15] E. J. Topol, Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again. New York, NY: Basic Books, 2019.

[16] A. Esteva et al., "A guide to deep learning in healthcare," Nature Medicine, vol. 25, no. 1, pp. 24–29, 2019.

[17] National Digital Health Mission, Ministry of Health and Family Welfare, Government of India. [Online]. Available: https://abdm.gov.in/

[18] Firebase Cloud Messaging Documentation, Google. [Online]. Available: https://firebase.google.com/docs/cloud-messaging