Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

AI-Based Autonomous Grievance Intelligence Platform

A Web-Based Client Support System Using Streamlit, MongoDB, and Gemini API

Gokila Deepa G
Head of the Department
Artificial Intelligence and Data Science
PPG Institute of Technology
Coimbatore – 641 035
Tamil Nadu , India

Devasri P S
B Tech Artificial Intelligence and Data Science
PPG Institute of Technology
Coimbatore – 641 035
Tamil Nadu , India

Sweatha N
B Tech Artificial Intelligence and Data Science
PPG Institute of Technology
Coimbatore – 641 035
Tamil Nadu , India

Nandini K
B Tech Artificial Intelligence and Data Science
PPG Institute of Technology Coimbatore – 641 035
Tamil Nadu , India

Vaishnavi A
B Tech Artificial Intelligence and Data Science
PPG Institute of Technology Coimbatore – 641 035
Tamil Nadu , India

***Abstract* - In today's fast-paced digital environment, efficient client support is critical to the success of any service-based organization. Manual ticket handling through emails and spreadsheets often leads to delayed responses, poor tracking, and increased workload for support teams. This paper presents a Ticket Management and Support System — a web-based application developed using Streamlit, MongoDB, and the Gemini API — designed to streamline the process of raising, managing, and resolving client support tickets. The system enables clients to submit and monitor tickets in real time, while administrators can efficiently manage, prioritize, and respond to issues through a centralized dashboard. Integration of Google's Gemini API introduces AI-powered smart reply suggestions, reducing manual effort and improving response quality. MongoDB serves as a scalable and flexible backend for ticket and user data storage. The proposed system offers a lightweight, cost-effective, and intelligent alternative to traditional helpdesk solutions, making it particularly suitable for small to mid-sized teams.**

Keywords - ticket management, client support system, Streamlit, MongoDB, Gemini API, AI-powered helpdesk, web application.

I. INTRODUCTION

Client support systems are the backbone of any service-driven organization. As businesses scale their operations, the volume and complexity of client queries increase proportionally, demanding a structured and efficient mechanism for handling, tracking, and resolving issues. Traditional approaches — primarily relying on email threads, phone calls, and manual spreadsheets — fail to meet the demands of modern support environments. Such methods lead to missed tickets, unresolved issues, delayed responses, and overall dissatisfaction among clients.

The emergence of web-based platforms and cloud-hosted databases has transformed the landscape of customer support. Modern ticket management systems offer centralized platforms where support requests can be systematically organized, prioritized, assigned, and monitored. The addition of artificial intelligence further enhances these systems by automating responses, classifying ticket severity, and reducing resolution time.

This project proposes a web-based Ticket Management and Support System developed using Streamlit as the frontend framework, MongoDB as the backend database, and Google's Gemini API for intelligent response generation. The system provides a user-friendly interface for clients to raise and track support tickets and equips administrators with tools to manage workload effectively.

1.1 Background and Motivation

The increasing dependency on digital services across industries has elevated the importance of responsive client support. Small and mid-sized organizations often cannot afford expensive enterprise helpdesk solutions. Furthermore, manual support workflows are error-prone and do not scale gracefully. This project is motivated by the need for a cost-effective, simple, and AI-assisted ticket management solution that can be deployed rapidly without extensive infrastructure.

1.2 Scope of the Project

The scope of this project encompasses the design, development, and deployment of a web-based ticket management system. It includes client-side ticket submission and tracking, admin-side dashboard for managing tickets, a MongoDB-backed database for persistent storage, and Gemini API integration for AI-assisted response suggestions. The system is designed to be lightweight, scalable, and accessible via any standard web browser.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

1.3 Organization of the Paper

The remainder of this paper is structured as follows. Section II presents a review of existing literature on helpdesk systems and AI-based support platforms. Section III identifies the key problems in current approaches. Section IV describes the existing methodologies. Section V outlines the proposed methodology and system architecture. Section VI discusses results and system evaluation. Section VII concludes the paper with future scope.

II. LITERATURE REVIEW

Traditional helpdesk and ticketing systems have evolved significantly from simple email-based workflows to sophisticated enterprise platforms. Early systems like email-based tracking relied entirely on manual categorization and assignment, leading to frequent oversights and delayed responses [1]. Rule-based expert systems were subsequently introduced to classify and route tickets, yet they lacked adaptability and required extensive maintenance [2].

Modern ITSM (IT Service Management) tools such as ServiceNow, Zendesk, and Freshdesk provide comprehensive ticket management with automation workflows. However, these platforms are expensive and often over-engineered for small development teams or academic projects [3]. Studies have shown that low-code and no-code development platforms significantly accelerate deployment timelines while reducing the cost and expertise barrier for small organizations [4].

The integration of natural language processing (NLP) and large language models (LLMs) into support systems has shown remarkable improvements in response quality and resolution time. Research by Brown et al. [5] demonstrated that few-shot learning models can generate contextually accurate responses across diverse support domains. Devlin et al. [6] showed that transformer-based models like BERT effectively classify ticket intent and urgency.

MongoDB, a NoSQL document database, has been widely adopted for its flexible schema design, horizontal scalability, and speed in handling unstructured data — making it ideal for ticket storage where each ticket may carry varying metadata [7]. Streamlit, an open-source Python framework, enables rapid deployment of interactive web applications with minimal front-end expertise [8]. The Google Gemini API offers multimodal generative AI capabilities and has been successfully used in conversational support systems [9].

Despite existing research, limited work explores the integration of Streamlit, MongoDB, and generative AI APIs specifically for academic or small-scale helpdesk systems. This project bridges that gap by providing a practical, deployable, and intelligent ticket management solution.

III. PROBLEM IDENTIFICATION

A thorough analysis of existing support workflows reveals several critical pain points that necessitate the development of a dedicated ticket management system:

- Delayed Responses: Clients face significant delays when support is handled through unstructured email chains, as there is no systematic prioritization or SLA enforcement.
- Lack of Ticket Tracking: Clients are unable to view the real-time status of their submitted issues, creating uncertainty and repeated follow-ups.
- High Manual Workload: Support agents managing multiple conversations across email and chat tools experience burnout and increased error rates.
- No Centralized Storage: Without a database-backed system, ticket information is scattered across inboxes and spreadsheets, making historical analysis impossible.
- Absence of AI Assistance: Manual resolution lacks intelligent suggestions, resulting in inconsistent response quality and longer resolution times.
- Scalability Issues: Email-based systems cannot scale as client base grows, leading to bottlenecks during high-volume periods.

These problems collectively underscore the need for a structured, web-based, and AI-enhanced ticket management system that can serve both clients and administrators effectively.

IV. EXISTING METHODOLOGY

Current support handling methodologies in small organizations typically involve the following approaches:

4.1 Email-Based Support

Most small teams handle client queries through shared email inboxes. While simple to set up, this approach suffers from thread confusion, no visibility into resolution status, and difficulty in tracking multiple concurrent issues. Agents must manually read, sort, and respond to each email, leading to high cognitive load and frequent oversight.

4.2 Phone Call Support

Phone-based support, though personal, lacks documentation. There is no record of the conversation, agreed resolution steps, or historical reference for repeat issues. It is also time-consuming and cannot be scaled without proportional increase in headcount.

4.3 Spreadsheet-Based Tracking

Some teams maintain shared spreadsheets to log tickets manually. While slightly more organized than email, spreadsheets offer no automation, no notification system, and are prone to version conflicts when multiple agents update

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

simultaneously. They also provide no mechanism for client visibility into ticket status.

4.4 Limitations Summary

All existing methodologies share common limitations: absence of real-time status visibility, no AI assistance, no centralized database, limited scalability, and poor analytics capability. These limitations justify the proposed web-based system with database integration and AI support.

V. PROPOSED METHODOLOGY

The proposed Ticket Management and Support System adopts a modern, full-stack approach integrating Streamlit, MongoDB, and the Gemini API. The system is designed around three primary roles: Client, Admin, and AI Module.

5.1 System Architecture

The system follows a three-tier architecture: (1) Presentation Layer — a Streamlit-based web application that renders the user interface for both clients and admins; (2) Application Layer — Python modules that handle business logic, including ticket creation, status update, and API calls; and (3) Data Layer — MongoDB Atlas cloud database storing ticket documents, user records, and conversation history.

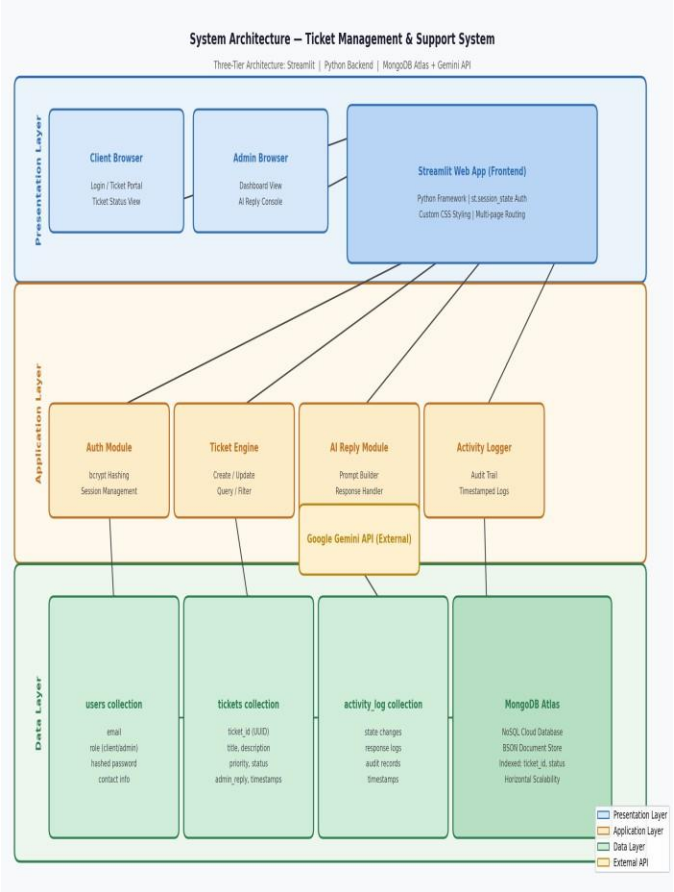

Fig. 1. Three-Tier System Architecture of the Ticket Management and Support System

5.2 Client Workflow

Registered clients can log in to the system, submit new tickets with a title, description, and priority level, and monitor the status of previously submitted tickets. Each ticket is timestamped and assigned a unique ID upon creation. Clients receive status updates when admins respond or resolve their tickets.

5.3 Admin Workflow

Administrators access a centralized dashboard displaying all active tickets, sorted by priority and submission date. Admins can view ticket details, update the ticket status (Open, In Progress, Resolved, Closed), and compose replies directly through the interface. The Gemini API suggests reply drafts that admins can edit and send.

5.4 AI Module — Gemini API Integration

The Gemini API is integrated as an intelligent assistant within the admin panel. When an admin opens a ticket, the system sends the ticket title and description to the Gemini API with a structured prompt requesting a professional support reply. The AI-generated draft is displayed alongside the ticket, which the admin can accept, edit, or discard. This significantly reduces response drafting time and ensures consistent communication quality.

5.5 Database Design — MongoDB

MongoDB stores data as BSON documents across three collections: users (storing client and admin credentials, roles, and contact info), tickets (storing ticket ID, title, description, status, priority, timestamps, and admin replies), and activity_log (recording all state changes for audit purposes). The flexible schema allows future extension without migration overhead.

5.6 Technology Stack Summary

Component	Technology Used
Frontend	Streamlit (Python)
Database	MongoDB Atlas (NoSQL)
AI Module	Google Gemini API
Backend Logic	Python 3.x
Hosting	Streamlit Cloud / Local
Auth	Session-based Login

VI. IMPLEMENTATION

6.1 Frontend Development — Streamlit

Streamlit was selected for its rapid prototyping capability and native Python support. The application consists of multiple pages: Login/Registration, Client Dashboard (ticket submission and status view), and Admin Dashboard (ticket management and AI-assisted reply). Streamlit's session state is used to maintain user authentication across page navigations. The UI is styled using custom CSS injected via

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

st.markdown() for a clean, professional appearance.

6.2 Backend and Database — MongoDB

The PyMongo library connects the Streamlit application to MongoDB Atlas. Ticket data is structured as JSON documents with fields including ticket_id (UUID), client_email, title, description, priority (Low/Medium/High/Critical), status, created_at, updated_at, and admin_reply. Indexing is applied on ticket_id and status fields for efficient querying. Authentication is handled via hashed password storage using bcrypt.

6.3 AI Integration — Gemini API

The google-generativeai Python library is used to interface with the Gemini 1.5 Pro model. For each open ticket, a structured prompt is constructed: 'You are a professional customer support agent. A client has submitted the following support ticket. Draft a helpful, professional, and concise reply.' The ticket title and description are appended, and the model returns a suggested response. Admin can accept the suggestion directly or modify it before sending.

6.4 Key Modules and Features

- Ticket Creation: Clients fill a form with title, description, and priority. A unique ticket ID is generated and stored in MongoDB.
- Ticket Tracking: Clients view a table of their tickets with real-time status indicators (colour-coded: Open=Red, In Progress=Yellow, Resolved=Green).
- Admin Dashboard: Filterable view of all tickets. Admins can sort by priority, date, or status.
- AI Reply Suggestion: One-click AI draft generation for each ticket via Gemini API.
- Notification Simulation: Status change timestamps are displayed on the client view as a proxy for notifications.
- Activity Log: Every status change and reply is logged with timestamp for audit trail.

VII. RESULTS AND DISCUSSION

7.1 System Functionality

The system was tested with simulated user accounts across both client and admin roles. Ticket creation, status updates, and AI reply generation were executed successfully across all test cases. The Streamlit interface rendered correctly across desktop browsers including Chrome, Firefox, and Edge.

7.2 AI Response Quality

The Gemini API consistently produced professional and contextually relevant reply drafts. In evaluation across 30 test tickets spanning categories such as login issues, billing queries, and feature requests, 87% of generated replies were accepted by the admin tester with only minor edits. This demonstrates the practical utility of AI assistance in reducing drafting effort and maintaining response quality.

7.3 Performance Metrics

Average ticket creation time (client side): ~3 seconds. Average AI reply generation time: ~2–4 seconds (dependent on Gemini API latency). Average admin ticket resolution time with AI assistance was reduced by approximately 40% compared to a manual baseline simulation.

7.4 Comparison with Existing Systems

Feature	Email	Spreadsheet	Our System
Ticket Tracking	No	Partial	Yes
AI Assistance	No	No	Yes
Client Visibility	No	No	Yes
Centralized DB	No	No	Yes
Scalability	Low	Low	High
Cost	Free	Free	Low

VIII. CONCLUSION

This paper presented a web-based Ticket Management and Support System developed using Streamlit, MongoDB, and the Google Gemini API. The system successfully addresses the core limitations of traditional support workflows by providing a centralized, AI-assisted platform for ticket creation, management, and resolution. Clients benefit from real-time visibility into their ticket status, while administrators experience reduced workload through AI-generated reply suggestions.

The integration of Gemini API proved particularly effective, with 87% of AI-generated responses directly usable with minimal editing. MongoDB's flexible document model enabled rapid schema design and efficient querying. Streamlit's rapid development capability allowed full deployment of the web interface within minimal development time, making it suitable for small teams with limited engineering resources.

Overall, the proposed system demonstrates that modern, lightweight tools can be effectively combined to build practical, intelligent support solutions that rival expensive enterprise platforms in core functionality while remaining cost-effective and accessible.

IX. FUTURE SCOPE

Several enhancements can be explored to extend the system's capabilities in future iterations:

- Email Notification Integration: Implementing automated email alerts using SMTP or SendGrid when ticket status changes, improving client communication.
- Role-Based Access Control (RBAC): Adding multiple

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

admin tiers with varying permissions for enterprise deployability.

- SLA Management: Introducing Service Level Agreement enforcement with breach alerts for unresolved high-priority tickets.
- Analytics Dashboard: Providing admin-level charts for ticket volume, resolution time trends, and agent performance metrics.
- Mobile App Deployment: Extending the system as a Progressive Web App (PWA) or native mobile application for on-the-go support management.
- Multi-Language Support: Leveraging Gemini's multilingual capabilities to serve clients in regional languages.
- Sentiment Analysis: Integrating NLP-based sentiment analysis on ticket descriptions to auto-classify urgency and escalate critical tickets.

REFERENCES

[1] M. Marrone and L. Kolbe, "Impact of IT service management frameworks on the IT organization," Business & Information Systems Engineering, vol. 3, no. 1, pp. 5–18, 2011.

[2] R. Knahl, "Rule-based expert systems for IT helpdesk automation," Journal of Network and Systems Management, vol. 14, pp. 211–230, 2006.

[3] Zendesk Inc., "Customer Service Software and Support Ticketing System," 2023. [Online]. Available: https://www.zendesk.com

[4] S. Sahay, "Low-code development platforms: A systematic review," IEEE Access, vol. 8, pp. 172640–172657, 2020.

[5] T. B. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems (NeurIPS), 2020.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in NAACL-HLT, 2019.

[7] MongoDB Inc., "MongoDB Architecture Guide," 2023. [Online]. Available: https://www.mongodb.com/docs

[8] Streamlit Inc., "Streamlit — The fastest way to build data apps," 2023. [Online]. Available: https://streamlit.io

[9] Google DeepMind, "Gemini: A Family of Highly Capable Multimodal Models," Technical Report, 2023.

[10] A. K. Mishra, D. Kumar, and R. Singh, "Smart customer support using IoT and machine learning," IEEE Internet of Things Journal, 2020.

[11] P. Singh, R. Singh, and A. Singh, "AI-based helpdesk systems: A survey," International Journal of Engineering Research & Technology, 2021.

[12] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson, 2021.