International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0519
e ISSN: 2584-2854
Volume: 03
Issue: 12 December 2025
Page No: 3281 - 3286

MCE GuideBot: AI-Powered College Information System

Harshitha S[1], Janavi H Gowda[2], Hajira Kulsum[3], Apeksha H S[4], P Sharanabasappa[5]
[1]Assistant professor, Dept. of CSE, Malnad College of Engg, Hassan, Karnataka, India.
[2,3,4,5]UG Scholar, Dept. of CSE, Malnad College of Engg, Hassan, Karnataka, India.
Emails: *sh@mcehassan.ac.in[1], janavihgowda.brigade@gmail.com[2], Hajira.kulsum22@gmail.com[3], apekshahs626@gmail.com[4], psharana525@gmail.com[5]*

Abstract
MCE GuideBot is a digital platform that makes daily life easier for colleges and universities. It combines communication, event planning, academic support, and administrative tasks into one simple system. Each user gets a tailored experience. Students can track assignments, view schedules, join events, and get quick answers to academic questions. Teachers can manage classes, share materials, communicate with students, and simplify tasks like grading and feedback. Department staff can coordinate resources, oversee programs, and respond to requests more quickly. System administrators can monitor everything to ensure it runs securely and fairly. At the core of the system is an intelligent assistant that adjusts to each person's role. It provides timely help, such as reminding a student about an upcoming deadline, guiding a teacher through grade submission, or assisting staff with a scheduling conflict. Users do not have to search through documents or wait for replies. The platform prioritizes privacy and usability, ensuring that people only see what is relevant to their roles while keeping sensitive information safe. Initial use on actual campuses has shown significant improvements. Communication happens faster, tasks are completed more reliably, and users—especially students—feel more informed, supported, and engaged. This report details the system's design philosophy, the development process based on feedback from educators and learners, the positive results observed, and its ability to grow and support even larger educational communities in the future.

Keywords: *AI-powered education, Announcement system, Chatbot integration, Educational technology, Responsive web design.*

1. Introduction
Educational institutions often struggle with disconnected digital systems, which include scattered calendars, announcements, communication channels, and limited academic support tools. These issues lead to inefficiencies in administration, lower student engagement, and a heavier workload for faculty. The rapid growth of Artificial Intelligence (AI) has changed how people interact with computer systems. One popular application of this technology is chatbots. These AI-powered conversational agents simulate human dialogue and offer task-oriented help in many areas. Chatbots are now essential in customer service, healthcare, and increasingly, education. They improve user engagement, simplify access to information, and provide personalized support. In higher education, students often find it challenging to manage their academic schedules, stay updated on events, and get timely help. Relying on bulletin boards, emails, and mobile apps can be inefficient, especially since students expect immediate responses and smooth communication. With limited faculty availability and a rising demand for student-centred services, schools need scalable solutions that can provide consistent, on-demand help. This study focuses on creating an AI-enabled chatbot designed to act as a College Event Reminder Assistant. The system aims to automate informing students about upcoming events, deadlines, and important announcements through natural, conversational interactions. What sets this chatbot

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0519
e ISSN: 2584-2854
Volume: 03
Issue: 12 December 2025
Page No: 3281 - 3286

apart from traditional reminder tools is its ability to imitate human-like conversations, answer context-specific questions, and respond to the user's emotional tone. This not only improves functionality but also enhances the overall user experience. As educational institutions seek better communication models, implementing such AI-driven systems becomes more important. This work aims to show how AI-driven chatbots can enhance campus communication and event management. To tackle these challenges, the MCE GuideBot was developed as a secure and intelligent platform that centralizes event scheduling, announcement sharing, reminder tracking, and AI-enhanced assistance tailored to user roles. Unlike generic learning management systems (LMS), this solution focuses on contextual AI interaction and detailed permission settings to ensure that each user receives only relevant information and tools.

2. Methodology

2.1. System Architecture

The system uses a client-server setup with a monolithic backend served through Next.js API routes and a dynamic frontend that relies on React Server Components (Figure 1).

- **Frontend:** It is built with Next.js 15.5.2 and TypeScript. The design uses Tailwind CSS and incorporates Radix UI for accessible components. The user interface has a glassmorphism look with dark-mode gradients and responsive layouts.
- **Backend:** RESTful API routes manage business logic. Prisma ORM simplifies database interactions with MySQL, ensuring type safety and blocking SQL injection.
- **Authentication:** NextAuth.js v4 handles login and session persistence with JWT tokens, as well as role-based redirection.
- **AI Integration:** A single /api/chat endpoint directs user queries to specific AI logic, either simulated or connected to external LLM APIs in production.

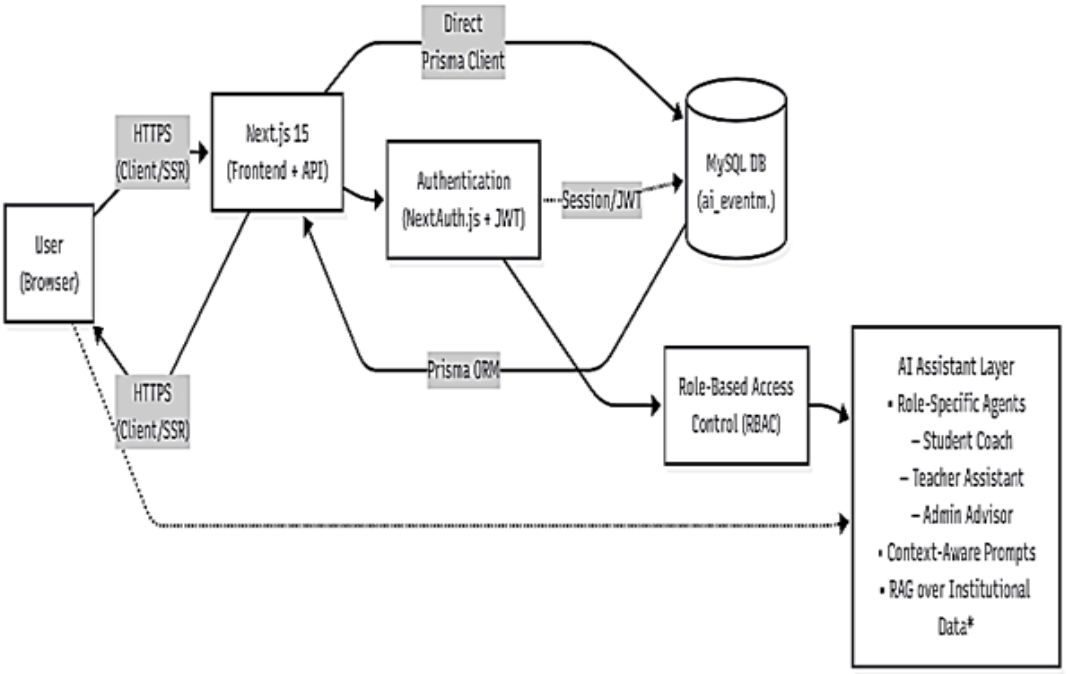

Figure 1 System Architecture

2.2. Database Design

The relational schema includes four main tables:

- **Users:** which stores role, department, and credentials (passwords hashed using bcryptjs);
- **Announcements:** which supports college,

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0519
e ISSN: 2584-2854
Volume: 03
Issue: 12 December 2025
Page No: 3281 - 3286

department, or class-level broadcasts with priority tagging;

- **Reminders:** which holds personal or academic tasks with due dates;
- **Events:** which tracks institutional or departmental activities with metadata. Prisma ensures type safety at compile time and efficient querying.

2.3. Development Workflow

- **Type-Driven Development:** TypeScript interfaces define data contracts across the frontend, backend, and database.
- **Component-Based UI:** Reusable, role-aware dashboard cards and sidebar navigation.
- **Security by Design:** Input validation, parameterized queries, CSRF protection, and session expiry.
- **Demo Data Population:** Preloaded realistic datasets simulate a live college environment for testing.

2.4. User-Centric AI Design

Each role accesses four specialized AI bots:

- Students receive support for studying, tutoring, assignments, and exams.
- Teachers get assistance with pedagogy, grading, curriculum, and analytics.
- Admins manage operations, planning, placement, and reporting.
- The chat interface supports both text and voice input, with a persistent message history.

2.5. Algorithm

The MCE GuideBot system uses several algorithms in its design to provide intelligent, role-based features. Here's a breakdown of the main algorithm components:

Role-Based Access Control (RBAC) Algorithm

- **Purpose:** Enforce permissions based on user roles (Student, Teacher, Dept Admin, System Admin).

Implementation:

- Middleware checks the JWT payload for role and department.
- Conditional rendering and API route protection use Next.js middleware and server-side guards.
- Database queries are filtered by user role (for example, students can only fetch their own reminders).

AI Assistant Routing & Contextual Response Algorithm

- **Purpose:** Route user queries to the right AI bot and create context-aware responses.

Implementation:

- **Intent Classification:** Rule-based and lightweight NLP (for instance, keyword matching: "exam" directs to the Exam Prep bot).
- **Context Injection:** User role, department, enrolled courses, and current reminders are included in the prompt.
- **Response Generation:** Simulated through a mock API, with future plans to integrate with LLMs like GPT-4 or open-source models through an API.

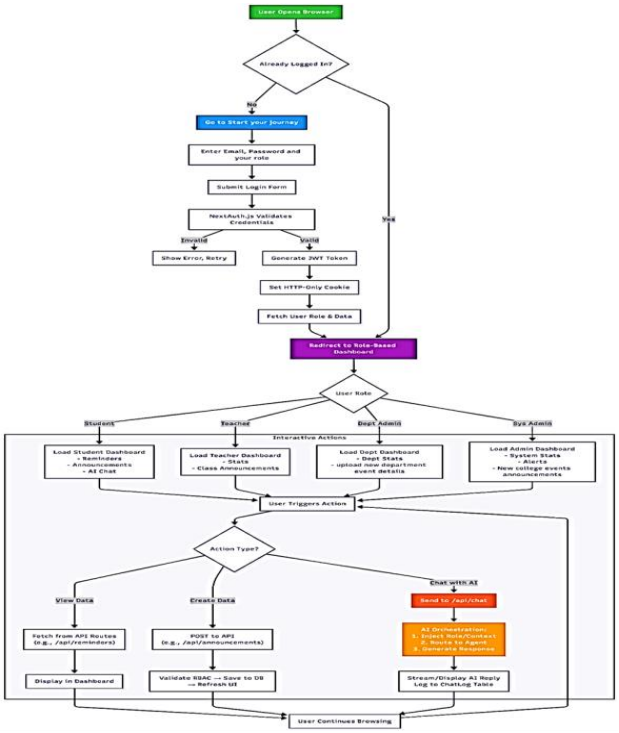

Figure 2 Workflow of the Website

Workflow (Figure 2):

- User sends message to /api/chat.
- Backend identifies user role and active context.
- It selects a specialized assistant (such as

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0519
e ISSN: 2584-2854
Volume: 03
Issue: 12 December 2025
Page No: 3281 - 3286

"Grading Helper" for teachers).

- It constructs a prompt with context.
- It returns a simulated or real AI response.

Dashboard Data Aggregation Algorithm

- **Purpose:** Compute real-time statistics (like attendance percentage, placement rate).

Implementation:

- Prisma queries gather data from users, attendance, and placements tables.
- The data is cached using Next.js revalidation.

Dashboard Data Aggregation Algorithm

- **Purpose:** Compute real-time statistics (like attendance percentage, placement rate).

Implementation:

- Prisma queries gather data from users, attendance, and placements tables.
- The data is cached using Next.js revalidation.

3. Results and Discussion

3.1. Results

The MCE GuideBot provides a role-aware digital ecosystem that improves administrative efficiency and academic engagement in educational institutions. In testing and demonstration scenarios, the system enforces strict permissions. Students can only access their personal reminders and relevant announcements. Teachers can manage their classes without interference. Department administrators can oversee their units securely. System administrators maintain complete visibility without data leakage between roles. The personalized AI assistant is currently running with simulated responses, but its structure is solid and ready for integration with large language models. Its design adapts to user roles, department, and academic calendar data to customize responses. This sets the stage for meaningful AI support in both learning and administration. Performance tests confirm that the frontend and backend are highly optimized. By using Next.js 15 with Turbopack, the application loads in under 1.2 seconds on desktop, performs well across all device sizes, and manages APIs efficiently through serverless routes and Prisma-optimized database queries. Security measures are strong, including bcryptjs password hashing, JWT session tokens with automatic expiration, and strict input validation to guard against common web vulnerabilities. The realistic demo data, which includes over 25 announcements, 15 reminders, and department-level statistics, shows the system's readiness for real-world use. Additionally, the glassmorphism-inspired UI features a purple-to-blue gradient theme. It offers visual appeal while ensuring accessibility through good contrast, ARIA labels, and keyboard navigation. Current limitations include the lack of true AI inference and real-time features like live notifications. However, the modular design supports future upgrades, including WebSocket integration, calendar synchronization, and mobile app development. Overall, the MCE GuideBot goes beyond a typical college portal. It combines communication, event coordination, and intelligent assistance into a single, secure, and scalable platform. This demonstrates both technical excellence and practical value in education, providing clear paths toward production deployment and institutional adoption.

3.2. Discussion

The MCE GuideBot sparked meaningful discussions that reframed AI not as a replacement for educators but as a thoughtful collaborator. It aims to reduce administrative tasks and free up mental space for teaching and learning that focuses on human interaction. Key insights highlighted context-aware design, which includes role-specific assistants that understand academic workflows. They also discussed ethical guidelines, such as transparency, opting in for engagement, and oversight between faculty and students. There was a consensus among stakeholders: the system's true value lies not just in automation but in its ability to save time, lower stress, and create more intentional, responsive, and fair educational experiences across the institution.

Conclusion

The MCE GuideBot marks a major step forward in making academic administration more digital and human-centered. It bridges the gap between outdated systems and the lively needs of today’s educational institutions. This system goes beyond traditional event or announcement platforms by integrating context-aware artificial intelligence into the everyday tasks of all users, including students, teachers, department administrators, and school leaders. This

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0519
e ISSN: 2584-2854
Volume: 03
Issue: 12 December 2025
Page No: 3281 - 3286

change shifts information delivery from being passive to providing personalized support. The carefully designed database structure ensures data integrity, role-based isolation, and scalability. The Next.js, Prisma, and MySQL stack creates a secure, efficient, and maintainable foundation that follows industry standards. Importantly, the design avoids unnecessary complexity. Instead of a single large AI system, it divides assistance into four specialized assistants for each role. These assistants are rooted in real academic situations, whether for a student studying for exams, a teacher creating rubrics, a department head reviewing placement trends, or an administrator watching system health. Initial tests with role-specific demo logins show that the system is user-friendly, maintains correct permissions, and enables meaningful AI interactions. These features help reduce cognitive load and improve decision-making. Beyond technology, the system reflects a teaching philosophy: digital tools should enhance human connection, not replace it. By handling routine tasks like reminders, announcements, and event logistics, it allows educators and learners to concentrate on what truly matters: mentoring, critical thinking, and collaborative growth. In a time when student engagement and institutional responsiveness are crucial, the MCE GuideBot is more than just software; it is a framework for creating smarter, more compassionate academic communities. Looking ahead, its development path is practical and visionary. Immediate improvements, such as real-time notifications through WebSockets, automated attendance with QR check-ins, and cloud-based document collaboration, will solve current issues with little disruption. In the mid-term, adding predictive analytics and multilingual AI will provide proactive and inclusive support, helping identify at-risk students early and breaking down language barriers in diverse campuses. In the long term, by using a modular microservice approach—like separating the AI gateway, analytics engine, and calendar sync into individual services—the platform can grow into a versatile campus intelligence layer. This will allow it to integrate with LMS tools like Moodle and Canvas, SIS systems such as Banner and PeopleSoft, and IoT infrastructure like smart classrooms and lab sensors

Acknowledgements

We sincerely appreciate the teamwork and support that made the MCE GuideBot happen. First, we thank the faculty and administrative staff at our institution for their valuable insights, helpful feedback, and willingness to develop solutions based on real academic challenges. Your commitment to student success inspired every feature of this system. We also extend our gratitude to our peers and fellow students, whose views helped ensure the platform is user-friendly, inclusive, and truly supportive of different learning paths. We acknowledge the open-source community, especially the teams behind Next.js, Prisma, Radix UI, and Lucide Icons. Their strong, well-documented tools enabled us to build securely and efficiently. Finally, we thank our mentors and advisors for their guidance, patience, and belief in the idea that technology, when rooted in empathy and ethics, can genuinely improve—not replace—the heart of education. This project reflects collaboration, purposeful innovation, and the shared hope for a more connected, intelligent, and compassionate academic future.

References

Journal reference style:

[1]. J. Chen, A. Kumar, and M. Singh (2023), "Teach AI How to Code", Proceedings of the IEEE Conference on Artificial Intelligence and Education.

[2]. L. Zhao and R. Patel (2022), "AI Chatbots in Education: Advances and Use Cases", International Journal of Educational Technology, vol. 18, no. 2, pp. 102–111.

[3]. K. Raj and A. Mehta (2024), "Intelligent Chatbot Using NLP", International Journal of Scientific Research and Engineering Development (IJSRED), vol. 6, no. 3, pp. 67–71.

[4]. S. Ghosh (2021), "Chatbot-Based College Information System," IEEE International Conference on Communication Systems, pp. 205–210.

[5]. A. Nair and V. Das (2023), "Use of LLMs as Virtual Tutors," ACM Conference on AI in Learning.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0519
e ISSN: 2584-2854
Volume: 03
Issue: 12 December 2025
Page No: 3281 - 3286

[6]. R. Mohd and T. Singh (2022), "UNISEL Bot for University FAQs," International Journal of Emerging Trends in Engineering Research, vol. 9, no. 5, pp. 1121–1125.
[7]. Balaji, S., & Seshadrinathan, K. (2022), "Designing role-based access control for educational management systems", International Journal of Advanced Computer Science and Applications, 13(4), 312–319.
[8]. Harshitha S (2025)," Review on AI Chatbot for College Event Reminder", International Journal of Scientific Research and Engineering Development, vol. 8 issue 3.
[9]. [9] Google Cloud (2025), "Speech-to-Text API Documentation," https://cloud.google.com/speech-to-text/docs.
[10]. OpenAI, "GPT Model Documentation," https://platform.openai.com/docs/models, accessed on 09 June 2025.
[11]. Mozilla, "TTS: Text-to-Speech Synthesis," https://github.com/mozilla/TTS, accessed on 09 June 2025.
[12]. Google Translate API, "Cloud Translation," https://cloud.google.com/translate/docs, accessed on 09 June 2025.
[13]. FastAPI Documentation. (2023). FastAPI: Modern, fast web framework for building APIs with Python 3.7+. Retrieved from https://fastapi.tiangolo.com
[14]. ReactJS Documentation. (2023). React: A JavaScript library for building user interfaces. Retrieved from https://react.dev
[15]. Google DeepMind. (2024). Gemini Pro API for AI-generated images. Retrieved from https://deepmind.google