# From Chatbot to Adaptive Syllabus-aware AI-mechanism: LLM Personalized Teaching Assistant in Higher Education

Or Peretz and Roei Zerahia *

School of Industrial Engineering and Management, The Engineering Faculty, Shenkar College-Engineering, Design, Art, Ramat-Gan, Israel
Email: or.perets@shenkar.ac.il (O.P.); roeizer@shenkar.ac.il (R.Z.)
*Corresponding author

Manuscript received December 12, 2025; revised January 21, 2026; accepted March 23, 2026; published July 22, 2026

***Abstract*—The integration of Artificial Intelligence (AI) into higher education offers scalable support for students but raises concerns regarding over-reliance, reduced effort, and diminished deep learning. This study introduces Michael, a syllabus-aware AI teaching assistant designed to scaffold reasoning through structured, hint-first dialogue aligned with course progression, rather than providing direct solutions. The system was deployed in an undergraduate Structured Query Language (SQL) course across three consecutive semesters and evaluated using a mixed-methods design combining interaction logs, pre–post questionnaires ($N$ = 170), and classroom observations. Results indicate high perceived ease of use ($M$ = 4.43) and a moderate but statistically significant increase in trust following exposure (from $M$ = 3.29 to $M$ = 3.58), while AI self-efficacy showed only minor changes. Usage patterns revealed a bifurcated structure, with students engaging in both short troubleshooting interactions and extended tutoring dialogues. Qualitative findings highlight adoption waves, tensions between efficiency and depth, and the sensitivity of trust to system reliability. These findings suggest that curriculum-aligned constraints and hint-first scaffolding can support instructional integration without displacing pedagogical goals. Rather than demonstrating causal learning gains, this study contributes design principles and in-situ evidence for deploying domain-specific AI assistants in technical higher-education contexts.**

***Keywords*—personalized teaching assistant, Large Language Models (LLMs) in higher education, Artificial Intelligence (AI)-powered assistants, AI-based structured academic dialogue, curricular sequencing, syllabus-aware guidance**

## I. Introduction

Large Language Models (LLMs) have entered university classrooms at a remarkable speed, promising scalable, always-available support while raising concerns about over-reliance on these tools, reduced effort, and erosion of critical thinking skills [1–3]. In technical subjects such as programming and database design and management, generic chatbots can easily provide overly complete responses, offering solutions that bypass productive struggle and disrupt instructional pacing [4, 5]. This raises a practical design question: how might domain-specific assistants scaffold reasoning, respect curricular sequencing, and augment rather than replace instructors?

Despite the rapidly increasing use of Artificial Intelligence (AI), there is limited real-classroom evidence on classroom-targeted tools, such as syllabus-aware, domain-bounded AI mentors that pace guidance to weekly course progression and deliberately avoid providing final answers. Prior work reports the benefits of intelligent tutoring systems and, more recently, Large Language Model (LLM)-based assistants; however, these rarely constrain assistance to a fixed syllabus with pedagogical guardrails or examine social and organizational consequences during live teaching [6, 7]. We addressed this gap with a self-developed AI tool called "Michael," a course-aligned AI teaching assistant for an undergraduate databases course for second-year students in the faculty of engineering. Michael is designed as an online mentor rather than an oracle: it is grounded exclusively in vetted course materials, gates content by calendar session (excluding vacations and non-study periods), and engages students through probing questions and partial hints to promote self-explanation and metacognitive monitoring [8, 9].

Michael couples a Generative Pre-Transformer (GPT) conversational core with retrieval over a vectorized repository of slides, assignments, and syllabus artifacts; behavior prompts enforce hint-first dialogue and restrict references to topics already covered. We evaluated Michael over a 14-session semester using a mixed-methods design that triangulated pre-adoption/post-use questionnaires (usefulness, ease of use, trust, and AI-related self-efficacy), in-class observations analyzed thematically, and fine-grained interaction logs capturing usage intensity and message dynamics. Thus, Michael is not a theoretical idea but a practical tool that was designed, built, tested, and run over a full academic year.

The present evidence is drawn from a single technical/procedural course context: second-year undergraduate engineering students enrolled in an academic 'Database' course based on Structured Query Language (SQL) with structured lab work, incremental prerequisites, and clear notions of correctness and stepwise problem solving. Accordingly, we examined students' perceptions, engagement patterns, and classroom integration mechanisms during authentic deployment. Our design recommendations and empirical patterns are best interpreted as applying primarily to similar higher-education settings (e.g., programming, databases, engineering problem sets) where curricular sequencing and scaffolded reasoning are central. We operationalized "pedagogy as constraint" using four design principles that can be generalized to similar technical/procedural higher-education courses:

- (Topic-gated support) Restricts retrieval and assistance to the currently enabled session and its learning objectives.
- (Hint-first) Prioritize stepwise reasoning prompts and partial hints over final answers.
- (Anti-shortcut enforcement) Use refusals and redirections when students request direct solutions without a process.
- (Instructor-centered integration) Preserve instructor authority and embed the tool within classroom routines

and norms rather than treating it as a standalone tutor.

The contributions of this study are twofold. First, we present a practical blueprint for a syllabus-aware assistant in database education that combines retrieval grounding with explicit prompting rules and content gating. Second, we provide an empirical account of classroom adoption and pedagogy under AI augmentation, including quantitative shifts in acceptance and self-efficacy and qualitative themes around adoption waves, efficiency-depth tensions, trust fragility under latency or outages, occasional syllabus "jump-ahead," and evolving student-teacher-AI roles. The remainder of this paper is organized as follows: Section II reviews related work on AI in higher education; Section III details the system architecture and student interaction flow; Section IV describes the methodology used in our study; Section V reports the quantitative and qualitative results; and Section VI discusses the implications, limitations, and future directions.

## II. Literature Review

Research on AI in Education (AIED) has progressed from early Intelligent Tutoring Systems (ITS) to contemporary Large Language Model (LLM)-based assistants capable of open-ended dialogue, code generation, and syllabus-aware guidance. This review traces this trajectory with an emphasis on technical education (programming and databases), pedagogies compatible with AI-supported learning, and evaluation frameworks appropriate for classroom deployments such as ours. We conclude by identifying the gaps that our system, Michael, is designed to address.

Early work on AIED established that computer tutors can diagnose student errors and personalize feedback, yielding meaningful learning gains relative to traditional instruction [4–6, 10]. Over the past decade, adoption in higher education has expanded beyond ITS toward predictive analytics, automated assessment, and conversational agents [1, 11, 12]. The emergence of LLMs built on transformer architectures has further enabled context-aware dialogue and on-demand generation of explanations, examples, and practice items [2, 13, 14]. Concurrently, a consistent body of work has highlighted the risks associated with generic chatbot use in education, including hallucinated or inaccurate content, biased outputs, academic integrity concerns, and misalignment between model behavior and instructional goals [3, 14–18]. These concerns have motivated calls for conservative prompting, retrieval grounding, transparency, and human-in-the-loop oversight in educational deployments [19, 20].

Classic ITS demonstrated sizable learning gains via domain models, stepwise feedback, and dialogic hints. AutoTutor advanced mixed-initiative natural-language tutoring with animated agents [21], whereas SQL-Tutor used constraint-based modeling to diagnose query errors and provide targeted feedback in databases [22]. Subsequent course-facing assistants, such as Jill Watson, automated large-class question-and-answer sessions, foreshadowing today's LLM-mediated support, but without syllabus-gated retrieval or explicit Socratic guardrails [23, 24]. Michael extends this arc by combining retrieval-augmented generation over vetted course artifacts with session-by-session gating and hint-first dialogue [25, 26].

In response to these documented risks, Retrieval-Augmented Generation (RAG) constrains LLM outputs to vetted course materials, improving accuracy, provenance, and curricular fit [25, 27]. Contemporary work has further explored Socratic/hint-first prompting and planning, showing that LLMs that elicit self-explanations and guide revisions can improve reasoning and learning processes in programming and math [26, 28–30]. These guardrails align with Cognitive Load Theory (CLT), which supports reducing extraneous load while preserving germane processing, and the zone of proximal development concept, with "desirable difficulties" supporting durable learning when feedback is appropriately staged [8, 31–33].

In technical domains such as SQL, learning hinges on iterative practice, immediate feedback, and principle-based mapping between syntax and semantics. Prior ITS demonstrated that constraint-based tutoring improves query formulation and debugging [22, 34], whereas dialogic systems such as AutoTutor have modeled mixed-initiative scaffolding [21]. Recent classroom deployments of LLM-mediated code assistants (e.g., CodeAid) have shown how no-solution and explanation-first policies can balance efficacy with depth, but also provided evidence of tensions between efficiency and conceptual understanding when assistants jump ahead in the syllabus [30]. These precedents motivated Michael's hint-only design for SQL tasks and its emphasis on stepwise decomposition and error-specific prompts.

Evidence links AI tutors with gains in performance and engagement, specifically when they complement, rather than replace, instruction [5, 12, 35]. Learners often value immediacy and low-stakes clarification; however, their acceptance of AI tools is sensitive to reliability, latency, and usability; expectation-experience gaps can depress uptake [36]. Accordingly, it is advisable to assess such systems using mixed-methods evaluations that combine interaction logs (usage intensity, question types, and feedback patterns) with perceptual measures and strategy instruments. Beyond the Technology Acceptance Model (TAM; [37]), the Acceptance and Use of Technology (AUT) explains the acceptance of new technologies via performance/effort expectancy, social influence, and facilitating conditions [38], whereas the Expectation-Confirmation Model (ECM) accounts for the post-adoption continuance of that acceptance [39]. Given our focus on reliability, trust-in-automation research is salient: meta-analyses and reviews consistently identify system performance and reliability as dominant drivers of calibrated trust, a pattern that has implications for guardrail design and transparency [40–42]. Consistent with these concerns, policy guidance emphasizes transparency, equity safeguards, and human-in-the-loop oversight-principles that align with session-gated access, hint-first dialogue, and syllabus-bounded retrieval [19, 20].

Despite promising results from the deployment of AIED systems, three gaps remain. First, many studies rely on short pilot studies or simulations, and it is unclear how well their results might translate to semester-length, in-class deployments that respect session-by-session pacing and institutional constraints [14, 17]. Second, systems are often generic; comparatively fewer assistants are domain-bounded,

syllabus-aware, and explicitly designed to scaffold rather than solve [1, 42]. Third, evaluations frequently under-theorize pedagogy, under-specify guardrails (e.g., hint-only policies), or fail to take into account how reliability shapes trust and continued use—areas in which integrating UTAUT/ECM with trust-in-automation offers greater explanatory power [38–40]. Our work with Michael addresses these gaps by combining RAG based on vetted materials, temporal scaffolding (session-gated access), and Socratic, hint-first dialogue, as well as by evaluating results in vivo across multiple semesters.

We treat Michael as a design intervention evaluated in vivo under classroom constraints. Accordingly, our goal is not to claim general learning gains from AI use but to contribute design principles and field evidence regarding how curriculum-aligned constraints, scaffolding policies, and classroom norms shape adoption, trust, and instructional integration.

## III. "MICHAEL": SCOPE AND ARCHITECTURE

This section presents the pedagogical stance and high-level system design of Michael, a course-specific AI teaching assistant for technical higher education courses. Fig. 1 shows the student-facing interface. Michael is designed to address a common failure mode of generic chatbots in education: over-answering. When students receive full solutions too early, the assistant can short-circuit productive struggle, disrupt instructional pacing, and reduce opportunities for self-explanation. Accordingly, Michael's goal is not to deliver final answers but to scaffold reasoning; in our case, it supports students in thinking through SQL problems and debugging their approach.

Michael is positioned as a mentor rather than an oracle. It is curriculum-aware, tracking the course's weekly progression and aligning the level of assistance with what has been taught so far. This "syllabus-scoped" behavior is intended to preserve active learning while expanding timely access to help. At inference time, Michael follows four enforced rules: hints precede answers, probing questions elicit intent and promote self-explanation, full solutions are refused when they would bypass the learning objective, and access is paced to the syllabus, constraining responses to the course scope and the student's current point in the semester.

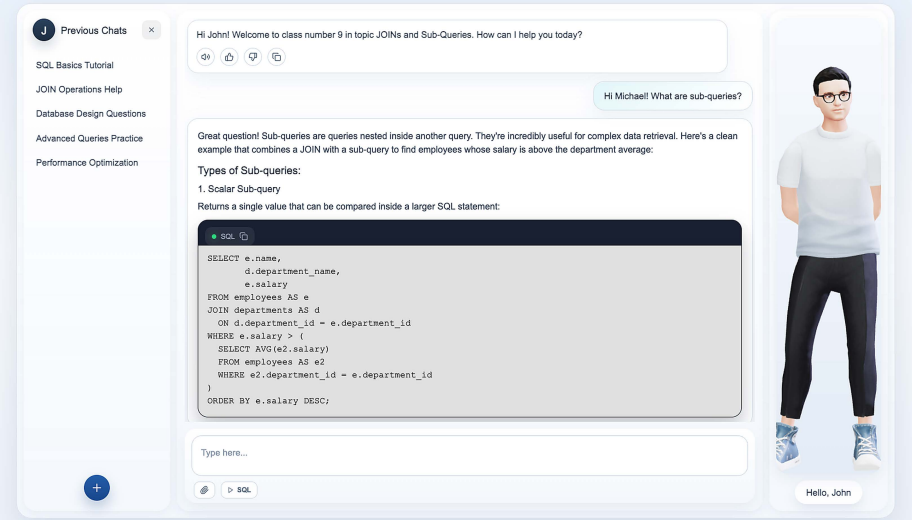


Fig. 1. Michael chat interface.

### A. System Architecture and Interaction Loop

Michael is a personalized three-tier web application that binds a large language model to a syllabus-scoped knowledge base and applies the above guardrails at inference time. Rather than relying on open-ended generation, responses are grounded in vetted course materials and constrained by the curriculum position. Additional implementation details (stack choices, metadata schema, and persistence/logging design) are reported in Appendix A1. Fig. 2 summarizes the seven steps of the student assistant interaction loop.

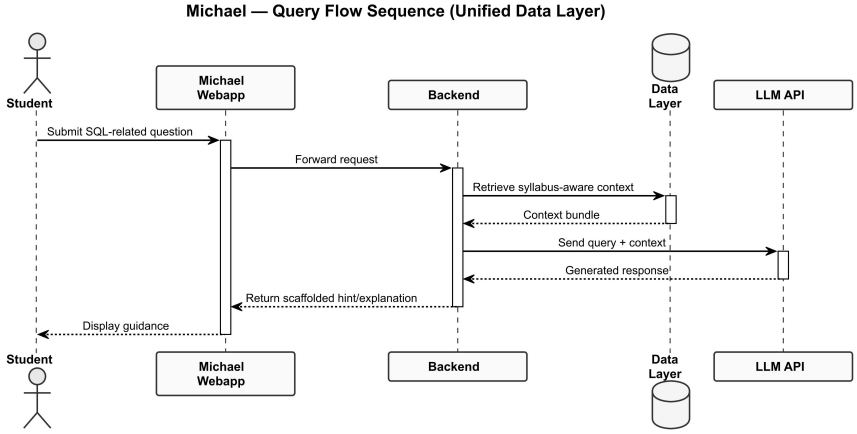


Fig. 2. Student-assistant interaction loop.

The steps (detailed in Appendix A2) implement a personalized, curriculum-aware, hint-first dialogue, in which each turn is grounded in vetted materials and constrained by the topic reached in the syllabus before response generation, and per-turn feedback and full telemetry support instruction-aligned analytics.

## IV. Method

This section presents the study context, design, instruments, and analytical strategy used to evaluate Michael, a syllabus-aware AI teaching assistant for a live, 14-session undergraduate class on SQL. Following a design-based arc (pilot, refinement, evaluation) across three semesters of the course, we report a mixed-methods evaluation. Specifically, the third semester served as the focal observational evaluation period to capture classroom dynamics during stabilized deployment, whereas the pre- and post-use questionnaire analyses were pooled across all three semesters to increase statistical power. This distinction ensured a robust quantitative evaluation of student perceptions while allowing for an in-depth qualitative focus on instructional integration.

We operationalize the assistant's effectiveness as a multi-dimensional, design-based outcome rather than a single learning-gain metric. Specifically, we evaluate reach and uptake (who used the tool and when) from interaction logs, engagement patterns (sessions, turns, intensity, and temporal dynamics) from logs, user acceptance and perceived experience (usefulness, ease of use, trust, and AI-related self-efficacy) from pre-and post-questionnaires, and instructional integration mechanisms (how students and instructors incorporated the assistant, including scaffolding behaviors, breakdowns, and role shifts) from structured classroom observations analyzed thematically.

The present study was not a controlled learning-outcomes experiment. We did not collect objective learning-gain measures suitable for causal inference (e.g., randomized access, standardized pre/post knowledge tests, or comparable exam-performance contrasts). Instead, we evaluated effectiveness in terms of observed instructional processes and classroom dynamics during use, adoption, and engagement patterns from logs, and student-reported perceptions (e.g., usefulness, ease of use, trust, self-efficacy, and perceived learning support). Accordingly, references to "learning improvement" in this manuscript are interpreted as perceived/experienced learning support rather than demonstrated gains in mastery. The research questions for this study are as follows:

**RQ1**: How do students adopt and engage with a syllabus-aware, hint-first assistant across a semester (for example, in terms of reach, session timing, and interaction intensity)?

**RQ2**: How do students' perceptions of usefulness, ease of use, trust, and AI-related self-efficacy change from pre- to post-use during classroom deployment?

**RQ3**: How is the assistant incorporated into classroom workflows (student strategies, instructor mediation, breakdowns), and what mechanisms shape trust and continued use?

### A. Participants and Setting

The study was conducted during a mandatory SQL course for second-year undergraduate students in the Department of Industrial Engineering and Management. Each weekly session comprised a two-hour lecture followed by a two-hour computer lab. The course was held thrice during one academic year, with 81, 72, and 39 students enrolled in the first (winter 2024), second (spring 2025), and third (summer 2025) semesters, respectively. For most, this was their first formal exposure to SQL. No student had previous experience with a syllabus-aware AI tutor; although, some reported informal use of public AI tools. Participation in research activities related to the AI tutor (surveys, voluntary use of the assistant during designated sessions, and non-participant observations) was optional and had no effect on grading. Participation was voluntary, unrelated to grading, and students could opt out without penalty. Logs were stored under access control and analyzed in aggregate, and quoted field notes were anonymized and, when necessary, lightly paraphrased to prevent re-identification.

### B. Study Design and Procedure

We employed a mixed-methods within-course study design. The first (winter) semester served as a pilot to identify technical and pedagogical issues, such as latency, prompt policy, and interface clarity. During the first trimester of this semester (course sessions 1–4 of the 14 total), Michael was unavailable to the students; lectures and lab sessions proceeded conventionally, supported only by the instructor and human teaching assistants. Students could access general web resources, including public AI tools, but not Michael.

In the spring semester, we deployed an improved version of Michael under stable conditions, accompanied by structured classroom observations. During the summer semester, we conducted the focal evaluation reported herein. All instruments, observation protocols, and procedures were pre-specified before the first lesson and remained constant throughout.

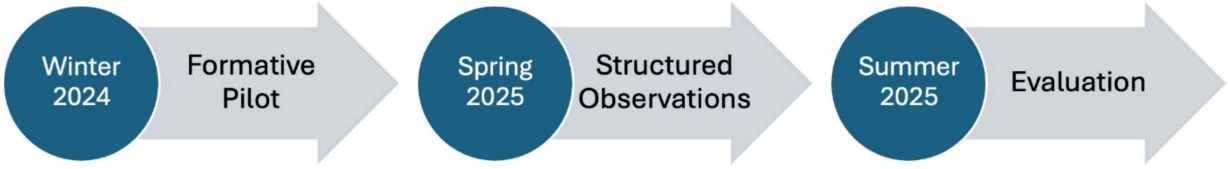


Fig. 3. Study timeline and access schedule.

In the summer semester, during which each session consisted of two classes in a row (the usual 14-week semester was compressed to 7 weeks with double lessons), the first two lessons were devoted to obtaining informed consent, administering the pre-questionnaire, and course logistics, with no access to Michael. During sessions 3–6, the assistant was enabled in six pre-specified sessions in which the instructor introduced SQL topics. Students were free to

consult Michael for hints and explanations, and participant observations were conducted in all intervention sessions. The last lesson (lesson 7) concluded the course summary and the study with the in-class post-questionnaire. The temporal structure of the study, including the alignment with course sessions, is summarized in Fig. 3.

### C. Instruments and Data Sources

Pre/post-questionnaires captured students' perceptions of Michael's usefulness and ease of use, trust and reliability, and students' AI-related self-efficacy. We used reliability to refer to system performance stability (e.g., latency and failures), trust to refer to students' judgment of whether the assistant is dependable and appropriate to rely on, and confidence only as a colloquial descriptor when not referring to a measured construct. The construct composition was as follows: usefulness (two items: AI learning potential and AI learning improvement), ease of use (one item), complexity (one reverse-coded item), self-efficacy-comfort/tech/AI (each one item), AI decision comfort (one item), trustworthiness (one item), and independence preference (one item). All items used a five-point Likert scale. Internal consistency was assessed using Cronbach's alpha (α) for multi-item scales. The two-item usefulness scale demonstrated acceptable internal consistency ($\alpha = 0.795$). All remaining constructs were assessed using single-item indicators; therefore, internal consistency reliability was not applicable to those measures. Identifiers collected for matching were immediately hashed and discarded after linkage to ensure participant anonymity. Analyses of change were restricted to students with matched pre/post responses for a given scale. Non-participant observations were conducted during the Michael sessions. Field notes focused on student strategies, instructor-student-AI coordination, visible breakdowns (e.g., latency and syllabus jump-ahead), and peer modeling of workflows. Table 1 summarizes the properties, wording, and composition of the perceptual constructs used in this study.

Application-level telemetry captured user, session, and turn metadata (timestamps, speaker role, retrieved sources, like/dislike flags). From these logs, we derived usage and engagement measures, including unique users, sessions, total turns, assistant/student turn ratios, messages per session, and per-student activity summaries. Temporal aggregates were computed by course session to align with intervention availability.

A central design goal was to keep assistance aligned with the course sequence. Michael enforces this primarily through session/topic-gated retrieval (restricting accessible materials to the currently enabled session and its associated topics) alongside refusal policies for direct "final answer" requests. Although the system logs include retrieval metadata, we did not compute a formal syllabus-alignment accuracy metric in this study. Instead, we provide the enforcement mechanism as part of the system description and qualitative proxy evidence from classroom observations documenting both alignment-preserving interactions and occasional "jump-ahead" incidents that were used as teachable verification moments.

Table 1. Summary of measurement constructs

| Construct | Items | Example Item Wording | Range | Direction |
|---|---|---|---|---|
| Usefulness | 2 | "The AI assistant improves my learning process in this course." | 1–5 | Positive |
| Ease of Use | 1 | "The assistant is easy to use and navigate." | 1–5 | Positive |
| Complexity | 1 | "Interacting with the AI system is overly complex." | 1–5 | Reverse |
| Self-Efficacy (Tech) | 1 | "I am comfortable using new educational technologies." | 1–5 | Positive |
| Self-Efficacy (AI) | 1 | "I feel confident in my ability to learn using AI tools." | 1–5 | Positive |
| AI Decision Comfort | 1 | "I am comfortable letting the AI guide my study steps." | 1–5 | Positive |
| Trustworthiness | 1 | "I trust the AI assistant to provide accurate and reliable guidance." | 1–5 | Positive |
| Independence Pref. | 1 | "I prefer to make academic decisions independently without AI input." | 1–5 | Positive |

### D. Data Analysis

#### 1) Quantitative analysis

Descriptive statistics (means, standard deviations, and 95% confidence intervals) were computed for each scale pre- and post-use of the tool. Pre- and post-change were assessed using paired t-tests, and Wilcoxon signed-rank tests were used when the Shapiro–Wilk test indicated non-normality of the difference scores. Effect sizes were reported with corrections for the family of scale tests. Exploratory associations related change scores on each perceptual scale to engagement metrics from logs (e.g., total messages, number of active sessions, and "like" ratio) using robust regressions that controlled baseline levels. Session duration is reported descriptively as the time between the first and last message in a chat. Because extreme values can reflect idle browser tabs or sessions left open, we interpret duration patterns cautiously and do not treat duration as a primary indicator of active learning.

#### 2) Qualitative analysis

The observation notes were coded inductively by the lead researcher in a process that progressed through familiarization, initial coding, theme generation, review, definition/naming, and finally reporting [43]. Trustworthiness was supported via an audit trail of code/theme revisions and the linkage of each theme to multiple time-stamped vignettes. To mitigate single-coder bias and enhance analytic rigor, the primary researcher maintained reflexive memos throughout the coding process, and two formal peer-debriefing milestones were utilized as analytic safeguards to challenge, refine, and validate the emerging thematic structure. Anticipated themes concerned adoption waves, efficiency-versus-depth tensions, trust fragility under technical friction, syllabus misalignment when the AI jumped ahead, and shifts in student-teacher-AI roles. Convergence coding aligned quantitative signals (e.g., shifts in perceived usefulness) with contemporaneous log trends (e.g., adoption surges after latency fixes) and thematic evidence from observations. Discrepancies (e.g., high usefulness alongside comparatively low trust) were examined explicitly to surface boundary conditions and guide design

recommendations.

## V. Results

We report findings from three complementary sources: (i) interaction logs quantifying the tool's reach, intensity, and temporal dynamics; (ii) student questionnaires assessing students' acceptance of the tool, trust in the tool, and AI-related self-efficacy; and (iii) in-class observations, which we analyzed thematically. We first discuss usage and engagement, then questionnaire outcomes, and finally observational themes, and conclude with a brief triangulated summary.

### A. Usage and Engagement

Engagement metrics were derived from interaction logs using four complementary indicators. A session was defined as a continuous chat interaction bounded by the first and last messages; the number of turns/messages reflected the number of exchanged messages per session or user and served as the primary indicator of engagement intensity. Session duration was computed as the elapsed time between the first and last messages and was treated as an approximate proxy, given that sessions may remain open during inactivity. Finally, engagement inequality across users was quantified using the Gini coefficient, which captures the degree to which activity was concentrated among a subset of students. Together, these measures characterize reach, intensity, temporal dynamics, and the heterogeneity of use.

Throughout the three consecutive semesters, 173 unique students out of 192 enrolled interacted with the Michael assistant at least once, representing 90.1% penetration. The logs contained 1275 chat sessions comprising 33,506 messages. Messages from the assistant numbered 21,693 (64.7%), and the remainder comprised student messages, consistent with Michael's hint-forward scaffolding style rather than free-form student monologues. The median number of messages per session was 13, and the mean was 26.3, with a substantial upper tail (75th percentile = 37.0; 90th percentile = 69.6; 99th percentile = 152.5), indicating that although many exchanges were short and targeted, a meaningful minority developed into extended tutoring dialogues.

Session duration (Fig. 4), computed as the span between the first and last message in a chat session, showed a similar heavy-tailed pattern. The median session duration was 8.7 min (IQR = 2.0–75.3 min), highlighting a heavy-tailed pattern in which the vast majority of sessions were brief, yet durations at the 90th percentile extended to 190.7 min.

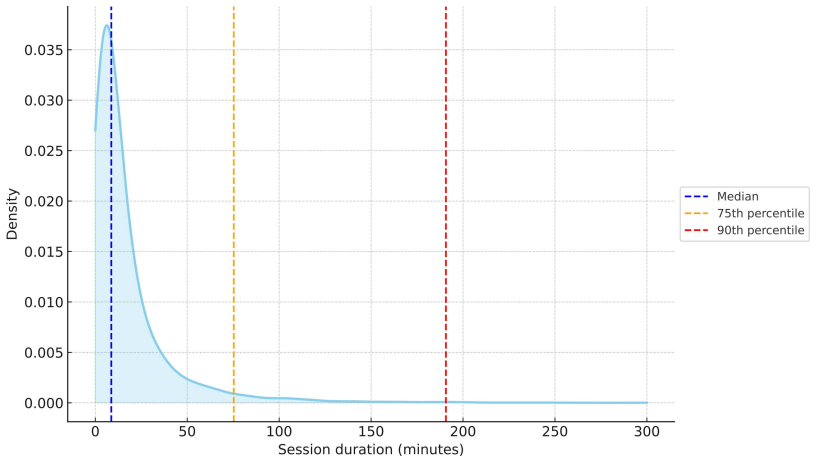


Fig. 4. Session duration distribution (minutes) in log scale, computed as the time between the first and last message in each chat.

At the user level, engagement followed a skewed distribution (Fig. 5). The Gini coefficient was 0.519, with the top 10% of users responsible for 35.6% of all messages. The median activity per user was 120 messages (mean = 193.5, SD = 208.4). The resulting long-tail distribution indicates the coexistence of brief, just-in-time troubleshooting and fewer but deeper mentoring exchanges.

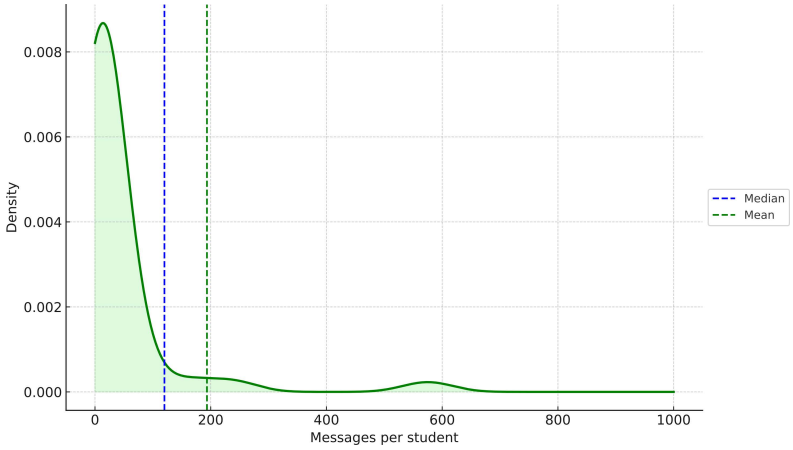


Fig. 5. User engagement distribution, showing total assistant-related messages per user across the semester.

The logs show that Michael was widely used whenever it was available, with students engaging in two distinct ways: quickly for focused troubleshooting or at greater length for more conversational tutoring. The fact that most turns came from the assistant fits its intended guidance-first design. A few unusually long sessions point to the need for better timeout rules, and the mix of short, simple interactions and longer, deeper interactions highlights the importance of supporting both "fast help" and "extended mentoring" modes in future versions.

### B. Questionnaire Outcomes

We examined student perceptions using pre- and post-use questionnaires. To evaluate the tool across its iterative deployment and maximize statistical power, analyses of pre/post changes were aggregated across all three semesters. Out of 192 enrolled students, 173 interacted with the system, and 170 successfully completed both the pre- and post-use questionnaires (an 88.5% response rate from total enrollment). For cross-sectional analyses across distinct deployment phases (after duplicate removal), valid post-use samples included 75 respondents in Semester 1 (pilot), 61 in Semester 2 (refinement), and 39 in Semester 3 (stabilized deployment).

#### 1) Analysis by phase

Across the three phases, a clear path emerged: early enthusiasm, mid-course friction, and eventual stabilization. Students rated the system's ease of use as high during the pilot (4.47), but this dipped noticeably in the refinement stage (4.00) before recovering once deployment settled (4.31). Perceived usefulness followed the same arc. Both learning potential and learning improvement started out strong (4.48 and 4.39), declined in semester 2 (4.07 and 3.90), and then rebounded in the final phase (4.33 and 4.21).

Complexity showed the opposite trend. Students found the system noticeably harder to handle during refinement (2.80, compared with 2.09 in the pilot), although this eased somewhat later (2.56). Trust hovered near the middle of the scale throughout (3.41, 3.34, 3.49), suggesting that students neither fully trusted nor fully relied on the system as an authority. Comfort with learning solely through AI was the measure with the lowest values: it remained consistently below the midpoint (2.67, 2.95, 2.69), underscoring students' hesitation about using AI as a stand-alone teacher. Finally, the desire to make independent decisions without AI input increased briefly during refinement (3.16) before returning to its earlier level once the system stabilized (2.90).

Taken together, these data show that the trajectory mirrors the standard rollout narrative: novelty drove early excitement, technical and pedagogical frictions surfaced mid-way, and stabilization restored confidence. The steadiness of trust indicates that students viewed the assistant less as an oracle and more as a guide, precisely the scaffolding role it was designed to play (Davis, 1989).

#### 2) Analysis of pre/post questionnaires

For 170 matched pairs, we tracked how individual students' views shifted after direct experience with the assistant. Several small but consistent changes appeared. The ease of use increased from 4.23 before exposure to 4.43 afterward, a statistically reliable gain. Perceived learning potential also increased slightly. At the same time, complexity ratings increased by approximately one-third of a scale point, suggesting that hands-on use increased students' awareness of the system's inner workings. Interestingly, ratings of learning improvement dipped slightly, hinting that once students engaged with real tasks, their initial optimism gave way to a more measured judgment.

Self-efficacy and attitude measures (such as comfort with independent learning, comfort with technology, comfort with AI, willingness to delegate decisions, trust, and preference for independence) showed only minor, mostly non-significant shifts. The exceptions were a small but clear rise in trust (from 3.29 to 3.58) and a parallel increase in preference for independence. Taken together, these shifts suggest that students became more comfortable with the system while reaffirming their own agency.

The correlational patterns tell a more layered story. Ease of use correlated only modestly with trust ($\rho$ = 0.22 at pre-test, 0.26 at post-test), indicating that usability mattered, but trust depended on other factors as well. These patterns are summarized in Fig. 6, which displays the full inter-scale correlation matrix at the post-snapshot. In contrast, ease of use was consistently and strongly tied to perceived usefulness. Comfort with AI decision-making tracked closely with trust, with high correlations ($\rho$ range 0.68–0.76) across phases. Preference for independent decision-making showed the opposite relationship, being consistently negatively correlated with trust, although this association weakened over time.

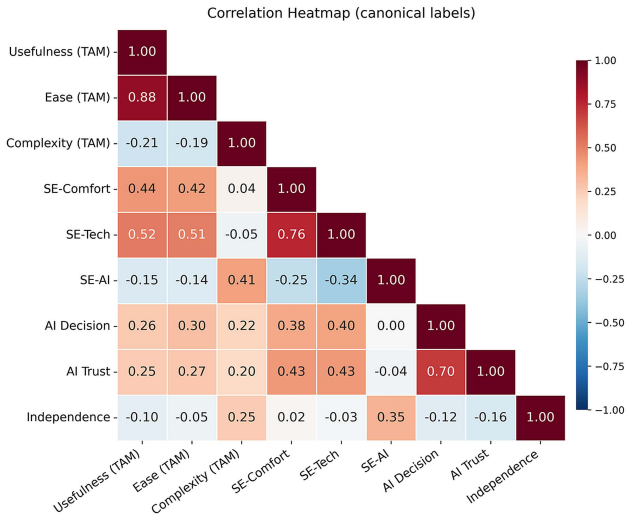


Fig. 6. Post-snapshot inter-scale correlation matrix (Spearman ρ).

Some relationships changed meaningfully across phases. The link between ease of use and perceived learning improvement weakened sharply: at first, a smooth interaction was equated with learning; however, by the third semester, students separated usability from learning outcomes. Similarly, the relationship between complexity and trust flipped in direction: early on, greater complexity was seen as a sign of seriousness, while later, it eroded confidence. Comfort with general technology and comfort with AI-only learning were consistently negatively related, underscoring that digital fluency in one area did not automatically carry over to the other.

Overall, the results point to trust as reflecting a multi-dimensional judgment. Although ease of use had an

effect, trust in the tool was shaped by a broader constellation of factors, including perceptions of usefulness, decision delegation, and independence, rather than being a simple extension of usability.

### C. Observational Themes

Following Braun and Clarke's six-phase procedure, our classroom observations yielded five interlocking themes that explain how students incorporated the Michael tool into their work and why the quantitative signals look the way they do. Across themes, we repeatedly observed the same causal micro-mechanism (event, reaction, consequence). For example, a failure or slowdown in the app would lead to a switch to a generic chatbot and thus an immediate dip in trust. The main themes in our conclusions are as follows:

**Adoption waves and social contagion**. Early adopters modeled effective "hint-first" querying; usage spread as peers copied strategies and stabilized after reliability fixes. This trajectory mirrored the mid-semester surge in unique users and messages.

**Efficiency vs. depth tension**. Students often invoked Michael for "grunt work" (such as creating a schema or table from scratch) while attempting a core query themselves; in parallel, a minority sought final answers on complex items, raising fears of "parroting" rather than understanding. A succinct classroom remark captured the concern: "They act like parrots." This tension mirrors the two usage modes in the logs (brief troubleshooting vs. extended dialogue) and underpins the call to design educational AI tools to preserve scaffolding without shortcutting reasoning.

**Trust fragility in relation to reliability shocks**. Trust oscillated with latency, crashes, and login frictions. In a typical sequence, Michael "not working" or being "slow" led students to immediately pivot to the generic and most-used AI tool, ChatGPT (a trust dip). After bug fixes/user interface improvements, enthusiasm and trust recovered ("Well done for improving the app"). The result is cautious, technology-contingent trust that helps explain the quantitative pattern of high ease, tempered confidence, and a weakened ease and trust linkage after routine exposure.

**Syllabus-technology gap and critical checking**. At times, the tool's guidance jumped ahead in advance of formal coverage of an area of the syllabus, prompting confusion and valuable moments of critical reasoning. Instructors leveraged these slips to re-anchor the scope and model verification strategies, turning misalignment into teachable checks.

**Reconfigured classroom roles**. As Michael absorbed routine clarifications, students approached the lecturer less frequently for quick fixes, and instructors increasingly initiated contact to press further about simple, routine questions, shifting the focus of authority toward metacognitive coaching. The notes describe an emergent "triangle of agency" (student, AI, and lecturer) and the need for the lecturer to reassert the framing role.

Themes co-varied systematically: adoption experiences directly shaped trust (1 and 5), the efficiency-depth dilemma sharpened when the AI proposed advanced content (2 and 4), and role reconfiguration (5) formed the backdrop to all other dynamics. Thematic evidence clarifies the quantitative profile:

- Adoption concentrates when availability and stability improve.
- Perceived usefulness and ease remain high, but trust tracks reliability shocks.
- Aligning with the observed post-exposure decoupling of ease from trust.
- The efficiency-depth trade-off explains students' mixed preferences for stepwise hints versus more directive help and supports an instructor stance centered on metacognitive prompts.

## VI. Discussion

We designed and deployed "Michael," a syllabus-aware, Socratic AI assistant for an undergraduate SQL course for students from the Faculty of Engineering, and evaluated it across a 14-session semester for three consecutive semesters with almost 200 students, in which the course AI assistant was intentionally enabled during the class session. A mixed-methods design triangulated fine-grained interaction logs, pre/post-questionnaires (usefulness/ease, trust, and AI-related self-efficacy), and thematic analysis of non-participant classroom observations. The system implements pedagogy as a constraint (hint-first dialogue, session-gated retrieval to prevent jumping ahead in the course syllabus, and refusal to provide final answers) and logs all interactions for analysis.

We position Michael through three complementary lenses that operate at different levels of explanation. TAM explains whether students will adopt the assistant and whether perceived usefulness and ease of use shape intention and continued use. CLT explains how interaction design can support learning processes: hint-first scaffolding and curriculum-aligned retrieval are intended to reduce extraneous load (e.g., irrelevant or overly advanced information) while preserving germane processing through stepwise reasoning. Social learning explains how use diffuses and stabilizes in classrooms: students learn "how to use" the tool by observing peers and negotiating roles with instructors, which reshapes norms (e.g., verification practices, help-seeking routines) and determines whether the assistant becomes a shared resource or a contested shortcut. Together, these lenses provide a coherent account of adoption, experience, and classroom integration.

### A. Main Findings

The following findings describe adoption, perceived usefulness/trust, engagement patterns, and classroom integration mechanisms, rather than causal effects on objective learning gains.

- **Adoption and engagement**. Michael achieved broad reach with a bifurcated usage pattern: many short, just-in-time exchanges coexisted with a smaller number of extended dialogues, and engagement was concentrated during the planned availability windows. Activity was moderately unequal (long-tailed), consistent with other experiences with classroom technologies in which a core of early adopters sustains deeper help-seeking while others dip in for quick hints; this is an expected social diffusion profile in visible, shared learning environments [44].
- **Perceptions and trust**. Usability remained high and stable, whereas trust was lower and sensitive to reliability

shocks (latency/outages). Notably, the pre-exposure association between ease and trust disappeared after routine use, indicating that post-exposure trust was driven less by interface comfort and more by perceived reliability, aligning with the technology acceptance model's emphasis on expectancy-experience alignment [36, 37].

- **Mechanisms from observations**. We identified five recurring themes that explained the patterns we observed. First, students picked up usage in waves, often after watching their peers model how to use the tool. Second, there was a tension between efficiency and depth: students used Michael for setup help but sometimes pushed for full answers. Third, trust proved fragile, and technical glitches reduced students' trust in the tool; however, trust recovered after the fixes. Fourth, jump-ahead moments occasionally surfaced in which the assistant previewed material later in the course syllabus, which helped clarify the scope and pacing. Finally, classroom roles shifted: as Michael handled routine questions, instructors focused more on coaching students' thinking. From the viewpoint of CLT and scaffolding, these behaviors make sense. Offloading setup tasks can reduce unnecessary load if guidance stays within students' learning zone. However, providing full solutions too early risks bypassing the need for students to process information more deeply. This highlights why the hint-first design and session-by-session guardrails are valuable [4, 8, 9, 45].
- **Pedagogy as system behavior**. Encoding the mentor stance at inference time via retrieval limited to vetted, session-appropriate materials and behavior prompts that refuse final answers appeared feasible in this classroom deployment and was perceived as educationally useful by students and instructors. When the system showed adequate reliability, this configuration supported timely, syllabus-aligned help without displacing instruction, consistent with evidence that tutor-like feedback complements (rather than replaces) human teaching [5, 45].

### B. Interpretation

From a TAM perspective, the combination of stably high ease of use and variable trust, though measured here via single-item indicators that capture broad student sentiment rather than multidimensional scale constructs, indicates that acceptance was gated by reliability rather than usability per se. The decoupling of ease from trust after routine exposure suggests that once interaction costs were low, students' judgments hinged on whether the system behaved predictably under load; outages and latency likely created expectancy violations that suppressed trust in the tool despite its perceived usefulness [36, 37]. In short, usability was a necessary but insufficient condition for acceptance. Across semesters, reliability appeared to function as a key precondition for trust and sustained use: periods of latency or failures were followed by rapid shifts to generic tools, while stability improvements coincided with renewed uptake.

CLT provides a complementary account of how students used Michael. Offloading routine setup (e.g., schema reminders or parameter scaffolds) plausibly reduced extraneous load, whereas hint-first dialogue preserved germane load for core SQL reasoning. Session-gated retrieval kept guidance within learners' proximal zone, and the explicit refusal to provide final answers helped prevent the short-circuiting of productive struggle [8, 9]. Occasional jumping-ahead incidents marked the boundary case: they momentarily raised intrinsic load but also enabled teachable moments of scope control and verification, reinforcing metacognitive habits rather than passive answer-taking.

The temporal pattern of adoption aligns with social learning theory. Early adopters modeled effective "hint-first" querying in public classroom contexts, and their visible success catalyzed the tool's diffusion once it became reliable and embedded in routine activities [44]. This social contagion helps explain the long-tailed engagement profile and the observed shift in instructional roles: as routine clarification moved to the assistant, instructors increasingly occupied a coaching space, directing attention to why a solution works rather than merely providing that solution.

Classroom AI raises equity and governance concerns that extend beyond usability. Access disparities (device quality, connectivity, language proficiency, and disability accommodations) may shape who benefits from AI support. Dependency risks are nontrivial, even with hint-first scaffolding; some students may over-rely on AI outputs or seek shortcuts, requiring explicit norms, verification prompts, and instructor oversight. Bias and error risks persist in LLM responses, and transparency about system limits, logging, and refusal policies is essential, as is preserving student agency (the assistant as guide, not authority). Finally, privacy protections and data minimization, controlled access to logs, anonymization, and clear consent procedures that separate participation from grading and ensure students can opt out without penalty are necessary.

### C. Limitations and Future Work

This study is limited to a single course and institution, which limits its generalizability. Reliability fluctuations early in the semester confound the attribution between pedagogy and stability. Students retained access to the public AI during comparison sessions, thus introducing contamination. Some analyses are sensitive to instrumentation: idle tabs inflated session durations, and not all scales yielded fully paired pre/post samples. Additionally, trust and several other perceptual constructs were measured using single-item indicators. While necessary to minimize survey fatigue during brief classroom intervals, single-item measures restrict construct depth and preclude formal internal consistency testing, which may limit the reliability of these specific findings. Like all self-report measures, these are also subject to familiar response biases. Collectively, these constraints temper causal claims while still offering actionable design signals. The tool is likely to be easily extensible to additional technological courses in the fields of engineering and computer science.

Future studies should address three main directions. First, increase the reliability of the tool and its instrumentation by defining service-level objectives, deploying circuit breakers, requesting de-duplication, and providing short-lived caching for frequent syllabus snippets. Upgrade observability with client/server heartbeats for accurate session processes and

flows, idle-timeout heuristics, and per-turn provenance that links each hint to its retrieved evidence and policy gates. These steps will stabilize trust and sharpen measurement fidelity. Second, stage the pedagogy and equip instructors: phase onboarding (orientation, guided use, autonomous practice) with guardrails surfaced within the user interface. Introduce a lightweight hint taxonomy and brief "AI-answer critique" activities and provide a live instructor dashboard alongside metacognitive prompt libraries to support the coaching role. Third, tighten evaluation designs: use counterbalanced or randomized access across sections; ensure fully paired identifiers for all scales; add objective outcomes (task correctness, error types, time to solution) plus delayed-transfer assessments; and pre-register contrasts that decouple reliability (uptime/latency strata) from pedagogy (hint depth, gating strictness), with policy-level A/B toggles to estimate causal effects. Michael is currently an assistant to help with lab classes, but the same mechanism is suitable for various additional pedagogical capabilities that can be integrated during an academic course, such as homework, exams based on study material, and providing enrichment for those interested.

## VII. Conclusion and Implications

This study presents "Michael," a syllabus-aware, hint-first AI teaching assistant designed for technical/procedural higher education contexts. Rather than positioning the system as a replacement instructor, we implemented pedagogy as a constraint, session/topic-gated retrieval aligned to the course sequence that privileges hints over final answers, and refusal behaviors intended to reduce shortcutting. Across three consecutive semesters of in-class deployment in an undergraduate "Database" course based on the SQL language for second-year students at the faculty of Engineering, our mixed-method evaluation triangulated interaction logs, student questionnaires, and classroom observations to characterize adoption, perceived usefulness/ease of use, trust dynamics, and emergent classroom roles.

The main contribution is a design blueprint evaluation pattern for curriculum-aligned educational AI. Practically, the results highlight several takeaways for instructors and ed-tech designers, as follows:

- Reliability is a precondition for trust in classroom AI, in which small technical frictions can quickly drive students back to generic tools.
- Gating and scaffolding policies reduce (but do not eliminate) syllabus drift, and "jump-ahead" moments can be leveraged as teachable verification prompts.
- Students naturally bifurcate into short "just-in-time" help and longer mentoring-style dialogues, suggesting that interfaces should support both fast troubleshooting and extended reasoning.
- The deployment of AI reshapes classroom roles toward a "triangle of agency" (student-AI-instructor), increasing the value of instructor oversight and metacognitive coaching.
- A shift in instructional and learning focus from repetitive technical setup to higher-order conceptual and design decisions, facilitated by offloading routine coding tasks to an AI assistant.

The results underscore that classroom AI adoption is shaped not only by model capability but also by pedagogical constraints and classroom norms. Topic-gated alignment and hint-first scaffolding appear to reduce syllabus drift and shortcutting pressures while preserving the instructor's role as the authority for validation and assessment. At the same time, trust remains fragile, and reliability issues can rapidly push students toward generic tools, suggesting that stable performance and transparent limitations are prerequisites for sustained, curriculum-aligned use.

## Appendix

### A. System Architecture and Interaction

#### 1) System architecture

The client tier is a Next.js chat interface that streams replies and records per-message likes/dislikes. The application tier centers on a retrieval orchestrator that mediates all model calls and application telemetry. A dedicated Model Context Protocol (MCP) service injects runtime metadata (such as course session, topic, language preference, and availability flags) to keep guidance synchronized with the syllabus. MongoDB persists users, sessions, messages, and feedback for evaluation and reliability monitoring. The model & data tier comprises the Language Model (LLM) API, a vector database indexing vetted course artifacts (slides, assignments, examples), and a versioned course data store used for ingestion and updates.

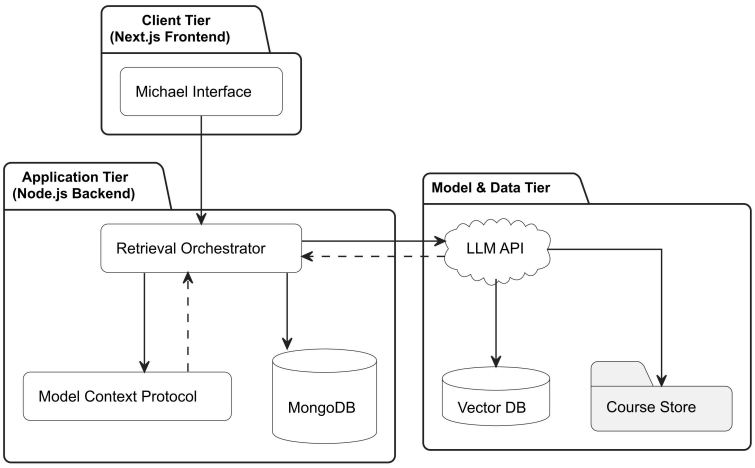


Fig. A1. System architecture of Michael.

Retrieval is limited to vetted artifacts and filtered by the MCP session and topic. For each turn, the orchestrator composes a prompt package with retrieved snippets, MCP metadata, and the behavior policy encoding Michael's mentor stance (Socratic dialogue, hint-first scaffolding, refusal to provide final solutions or to jump ahead). Lightweight post-processing rejects out-of-scope generations and reroutes students to prerequisites or brief, clearly labeled previews when needed (Fig. A1).

*2) Student interaction steps*

- **Query submission (client).** The student (after providing personal login authentication) asks a question in the Next.js chat. The client includes session identifiers and the prior turn's feedback state to maintain conversational continuity.
- **Reception and enrichment (application).** The backend receives the message and enriches it with MCP metadata (current calendar session, topic, feature flags) to align the request with the syllabus.
- **Syllabus-scoped retrieval.** The Retrieval Orchestrator queries the vector DB for vetted snippets filtered by MCP session/topic and compiles a prompt bundle (student query + retrieved snippets/citations + behavior policy).
- **Generation under guardrails (model and data).** The bundle is sent to the LLM API. A lightweight compliance pass rejects drafts that are out of scope or reveal solutions and, if necessary, re-prompts with stricter constraints.
- **Streaming scaffold (client).** The final reply is streamed in short turns that end with a probing question or next step to encourage self-explanation. If the query relates to material outside the scope of the covered session, the reply to redirects to prerequisites or offers a brief, labeled preview rather than jumping ahead.
- **Feedback and logging (application).** The student can like or dislike the message. The system logs messages, timestamps, retrieval context, and feedback to MongoDB, enabling later analyses of stumbling blocks and reliability monitoring.
- **Iterative loop and exit.** The dialogue iterates until the question is resolved or the system detects that the discussion has stagnated, at which point Michael summarizes progress and suggests next-step study actions or instructor follow-up.

## Conflict of Interest

The authors declare no conflict of interest.

## Author Contributions

O.P. conceived the study, designed and implemented the AI system, and led the data collection process. O.P. performed the quantitative and qualitative analyses and wrote the first draft of the manuscript. R.Z. contributed to the study design, interpretation of the results, and critical revision of the manuscript. Both authors reviewed and approved the final version.

## References


[1] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education—Where are the educators?" *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, 39, 2019. doi: 10.1186/s41239-019-0171-0

[2] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier, S. Krusche, G. Kutyniok, T. Michaeli, C. Nerdel, J. Pfeffer, O. Poquet, M. Sailer, A. Schmidt, T. Seidel, and G. Kasneci, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, 102274, 2023. doi: 10.1016/j.lindif.2023.102274

[3] N. Selwyn and J. Aagaard, "Impacts of artificial intelligence on education," *Learning, Media and Technology*, vol. 46, no. 2, pp. 175–189, 2021. doi: 10.1080/17439884.2021.1877654

[4] B. P. Woolf, *Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing e-Learning*, San Francisco, CA, USA: Morgan Kaufmann, 2009.

[5] J. A. Kulik and J. D. Fletcher, "Effectiveness of intelligent tutoring systems: A meta-analytic review," *Review of Educational Research*, vol. 86, no. 1, pp. 42–78, 2016. doi: 10.3102/0034654315581420

[6] I. Roll and R. Wylie, "Evolution and revolution in artificial intelligence in education," *International Journal of Artificial Intelligence in Education*, vol. 26, no. 2, pp. 582–599, 2016. doi: 10.1007/s40593-016-0088-x

[7] S. García-Méndez, F. Arriba-Pérez, and M. C. Somoza-López, "A review on the use of large language models as virtual tutors," *Science & Education*, 2024. doi: 10.1007/s11191-024-00530-2

[8] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cognitive Science*, vol. 12, no. 2, pp. 257–285, 1988. doi: 10.1207/s15516709cog1202_4

[9] M. T. H. Chi, M. Roy, and R. G. Hausmann, "Observing tutorial dialogues collaboratively: Insights about human tutoring effectiveness from vicarious learning," *Cognitive Science*, vol. 32, no. 2, pp. 301–341, 2008. doi: 10.1080/03640210701863396

[10] J. R. Carbonell, "AI in CAI: An artificial-intelligence approach to computer-assisted instruction," *IEEE Transactions on Man-Machine Systems*, vol. 11, no. 4, pp. 190–202, 1970. doi: 10.1109/TMMS.1970.299942

[11] F. Ouyang, L. Zheng, and P. Jiao, "Artificial intelligence in online higher education: A systematic review of empirical research from 2011 to 2020," *Education and Information Technologies*, vol. 27, no. 6, pp. 7911–7935, 2022. doi: 10.1007/s10639-022-10925-9

[12] H. Crompton and D. Burke, "Artificial intelligence in higher education: The state of the field (2016–2022)," *International Journal of Educational Technology in Higher Education*, vol. 20, no. 1, 22, 2023. doi: 10.1186/s41239-023-00392-8

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[14] C. McGrath, A. Farazouli, and T. Cerratto-Pargman, "Generative AI chatbots in higher education: A review of an emerging research area," *Higher Education*, 2024. doi: 10.1007/s10734-024-01288-w

[15] S. L. Blodgett, L. Green, and B. O'Connor, "Language (technology) is power: A critical survey of 'bias' in NLP," in *Proc. the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5454–5476. doi: 10.18653/v1/2020.acl-main.484

[16] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," arXiv preprint, arXiv:2311.05232, 2023.

[17] M. Bond, H. Khosravi, M. Laat, N. Bergdahl, and G. Siemens, "A meta-systematic review of artificial intelligence in higher education: A call for increased ethics, collaboration, and rigour," *International Journal of Educational Technology in Higher Education*, vol. 21, no. 1, 4, 2024. doi: 10.1186/s41239-023-00436-z

[18] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A systematic survey of prompt engineering in large language models: Techniques and applications," arXiv preprint, arXiv:2402.07927, 2024.

[19] F. Miao and W. Holmes, *Guidance for Generative AI in Education and Research*, Paris, France: UNESCO, 2023. doi: 10.54675/EWZM9535

[20] OECD, *Education Policy Outlook 2024: Reshaping Teaching into a Thriving Profession—From ABCs to AI*, Paris, France: OECD Publishing, 2024. doi: 10.1787/dd5140e4-en

[21] A. C. Graesser, P. Chipman, B. C. Haynes, and A. Olney, "AutoTutor: An intelligent tutoring system with mixed-initiative dialogue," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 612–618, 2005. doi: 10.1109/TE.2005.856149

[22] A. Mitrović, "SQLT-web: An intelligent SQL tutor on the web," *International Journal of Artificial Intelligence in Education*, vol. 13, no. 2–4, pp. 173–197, 2003.

[23] A. K. Goel and D. A. Joyner, "Using AI to teach AI: Lessons from an online AI class," *AI Magazine*, vol. 38, no. 2, pp. 48–59, 2017. doi: 10.1609/aimag.v38i2.2732
[24] B. Eicher, L. Polepeddi, and A. K. Goel, "Jill Watson doesn't care if you're pregnant: Grounding AI ethics in empirical studies," in *Proc. the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 88–94. doi: 10.1145/3278721.3278760
[25] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP," arXiv preprint arXiv:2005.11401, 2020.
[26] J. Liu, X. Tang, X. Wang, H. Jiang, S. Yang, and J. Zhou, "Retrieval-augmented generation for large language models: A survey," arXiv preprint arXiv:2312.10997, 2024.
[27] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *ACM Computing Surveys*, early access, 2024. doi: 10.1145/3686857
[28] H. He, H. Zhang, and D. Roth, "SocREval: Large language models with the Socratic method for reference-free reasoning evaluation," in *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024, pp. 2822–2840. doi: 10.18653/v1/2024.findings-naacl.177
[29] P. Kargupta, I. Agarwal, D. Hakkani-Tür, and J. Han, "Instruct, not assist: LLM-based multi-turn planning and hierarchical questioning for Socratic code debugging," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 9475–9495. doi: 10.18653/v1/2024.findings-emnlp.553
[30] M. Kazemitabaar, R. Ye, X. Wang, A. Z. Henley, P. Denny, M. Craig, and T. Grossman, "CodeAid: Evaluating a classroom deployment of an LLM-based programming assistant that balances student and educator needs," in *Proc. the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–20. doi: 10.1145/3613904.3642773
[31] F. G. W. C. Paas and J. J. G. Merriënboer, "Instructional control of cognitive load in the training of complex cognitive tasks," *Educational Psychology Review*, vol. 6, no. 4, pp. 351–371, 1994. doi: 10.1007/BF02213420
[32] L. S. Vygotsky, *Mind in Society: The Development of Higher Psychological Processes*, Cambridge, MA, USA: Harvard University Press, 1978.
[33] E. L. Bjork and R. A. Bjork, "Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning," *Psychology and the Real World: Essays Illustrating Fundamental Contributions to Society*, vol. 2, pp. 56–64, 2011.
[34] A. Mitrović, "Implementing CBM: SQL-Tutor after fifteen years," *International Journal of Artificial Intelligence in Education*, vol. 26, no. 1, pp. 131–161, 2016. doi: 10.1007/s40593-015-0079-3
[35] D. Ifenthaler and J. Y.-K. Yau, "Utilising digital technologies to support education in higher education," *Journal of Educational Technology & Society*, vol. 23, no. 1, pp. 245–257, 2020.
[36] E. Luger and A. Sellen, "'Like having a really bad PA': The gulf between user expectation and experience of conversational agents," in *Proc. the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 5286–5297. doi: 10.1145/2858036.2858288
[37] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, 1989. doi: 10.2307/249008
[38] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User acceptance of information technology: Toward a unified view," *MIS Quarterly*, vol. 27, no. 3, pp. 425–478, 2003. doi: 10.2307/30036540
[39] A. Bhattacherjee, "Understanding information systems continuance: An expectation-confirmation model," *MIS Quarterly*, vol. 25, no. 3, pp. 351–370, 2001. doi: 10.2307/3250921
[40] K. A. Hoff and M. Bashir, "Trust in automation: Integrating empirical evidence on factors that influence trust," *Human Factors*, vol. 57, no. 3, pp. 407–434, 2015. doi: 10.1177/0018720814547570
[41] P. A. Hancock, D. R. Billings, K. E. Schaefer, J. Y. C. Chen, E. J. Visser, and R. Parasuraman, "A meta-analysis of factors affecting trust in human-robot interaction," *Human Factors*, vol. 53, no. 5, pp. 517–527, 2011. doi: 10.1177/0018720811417254
[42] W. Holmes, M. Bialik, and C. Fadel, *Artificial Intelligence in Education: Promises and Implications for Teaching and Learning*, Boston, MA, USA: Center for Curriculum Redesign, 2019.
[43] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006.
[44] A. Bandura and R. H. Walters, *Social Learning Theory*, vol. 1. Englewood Cliffs, NJ, USA: Prentice-Hall, pp. 33–52, 1977.
[45] A. C. Graesser, "Conversations with AutoTutor help students learn," *International Journal of Artificial Intelligence in Education*, vol. 26, no. 1, pp. 124–132, 2016. doi: 10.1007/s40593-015-0086-4