From Confusion to Clarity in Introductory Computer Science through Intelligent Discussion Board Systems*

Tirth Shah[1], Saloni Tilekar[1], Kalpit Sharma[1]
Abbas Attarwala, Ph.D.[1], Ed Lindoo, Ph.D.[2]
[1]Computer Science Department
California State University
Chico, CA, 95973
tshah@csuchico.edu, sbtilekar@csuchico.edu, ksharma12@csuchico.edu
aattarwala@csuchico.edu
[2]Computer Information Systems
Anderson School of Business
Regis University
Denver, CO, 80221
elindoo@regis.edu

Abstract

First and second year computer science students often struggle to connect theoretical concepts with practical applications, a challenge that becomes particularly evident during online discussions. Although these discussions are valuable for peer interaction, confusion can arise when students cannot find a solution on their own and instructor responses are delayed, especially around foundational topics like data structures, algorithms, and core programming concepts. This paper presents an intelligent discussion board system designed specifically for introductory CS courses that addresses this challenge through AI-powered just-in-time

learning support. Our system allows students to engage in peer discussions while providing access to a specialized AI assistant that can be triggered using "Explain" prompts when confusion arises. The AI assistant uses retrieval-augmented generation (RAG) to pull relevant content from instructor-uploaded course materials, including textbook chapters, lecture slides, and supplementary resources, ensuring that responses are grounded in official course content. The system does not provide direct solutions to programming problems. Instead, it acts as a tutor by offering simplified explanations, generating pseudocode scaffolds, and providing step-by-step conceptual breakdowns that guide students toward understanding while maintaining academic integrity. We present the system architecture and highlight key features, such as the content retrieval mechanisms and the design of tutoring-focused AI responses. Our work contributes to the growing field of AI-enhanced educational technology by demonstrating how course-specific content retrieval can support student learning without compromising educational objectives.

1 Introduction

Discussion boards have emerged as fundamental tools for collaborative learning in CS education, enabling students to engage with course material through peer interaction and instructor guidance. Previous research has demonstrated their pedagogical value, showing that students often use online forums such as Piazza not only for logistical questions but also to support problem-solving and reasoning [13]. Platforms like Piazza are now widely used to facilitate structured question-and-answer exchanges between students and instructors. These platforms support active learning by encouraging students to articulate their understanding, seek clarification on complex topics, and learn from the perspectives of their peers'. However, existing discussion board systems face significant limitations in providing immediate contextual support when students encounter confusion, particularly outside of regular class hours or instructor office hours, when instructors may not be available to respond promptly. Research has shown that in the absence of timely instructor feedback, students can experience confusion or propagate misconceptions, underscoring the need for more responsive support mechanisms in asynchronous discussions [16]. Although students frequently use general-purpose AI tools such as ChatGPT when human guidance is unavailable, these tools function independently of course and context-specific information. This disconnect can lead to misleading or inconsistent explanations, especially in domains that require precision and structured conceptual development. AI-generated responses risk introducing misinformation through hallucinations and confusing outputs, potentially leading to serious negative consequences when users internalize and spread

inaccurate information [17].

This paper introduces ChatBook, an intelligent discussion board system designed to enhance student engagement through AI-assisted dialogue grounded in course-specific materials. Originally conceived to support textbook-based conversations, ChatBook also accommodates a broad range of instructional content typically shared in PDF format, including lecture slides, supplementary handouts, and publicly available textbooks. Aimed at first and second-year CS courses, the system bridges the gap between asynchronous discussion and timely, contextualized support by integrating RAG to ensure all AI responses are based on instructor-curated resources. When students encounter confusion and invoke the AI using 'Explain' prompts, ChatBook retrieves relevant passages and delivers simplified explanations, or conceptual scaffolding—without directly solving problems—thereby upholding academic integrity while supporting understanding. Another feature in ChatBook is the "I am still confused" feature that analyzes the previous five messages in a discussion thread to provide contextually aware follow-up responses, supporting iterative learning that builds over time and also may generate pseudocode if necessary.

Although students can access the AI assistant at any time, instructors are encouraged to adopt a discussion-first model that invites peer engagement before AI support. This approach reflects established pedagogical principles, particularly social constructivism and instructional scaffolding [14], in which learners co-construct knowledge before receiving individualized assistance. By constraining AI output to vetted course materials, ChatBook promotes alignment with instructor pedagogy and mitigates the cognitive dissonance that can occur with general-purpose AI tools.

For instructors managing large-enrollment CS courses, ChatBook offers practical benefits: it offloads routine explanation tasks, provides consistent and pedagogically aligned support, and generates discussion logs that surface common misconceptions and learning patterns—enabling data-informed teaching adjustments. Although developed with CS courses in mind, the architecture of the system is broadly applicable to any discipline where students benefit from context-based AI support, from literary analysis to historical interpretation. Students benefit from immediate and reliable assistance that complements peer collaboration and instructor guidance. In traditional classroom settings, ChatBook can enable students to revisit concepts, prepare thoughtful questions, and extend the discussion beyond the time constraints of live sessions. In classes taught online, whether synchronously or asynchronously, the system can promote a sense of connection, helping to counteract isolation that can undermine learning and engagement. By integrating scalable, course-aligned AI assistance into collaborative learning environments, ChatBook addresses a key gap in CS education. It ensures that technology supports—not replaces—human inter-

action, preserving the social and instructional dynamics essential to effective learning.

2 Literature Review

Previous work in digital learning environments has examined a wide range of tools and frameworks designed to support student engagement, collaborative learning, and real-time assistance. These include live chat tools for synchronous support, educational chatbots for project-based learning, curriculum-aligned AI systems, and RAG systems. The work in this paper builds on by offering a pedagogically grounded, course-specific, and discussion-integrated AI system—ChatBook—that supports just-in-time learning assistance without replacing peer collaboration or instructor authority.

Studies have consistently found that immediacy and synchronicity in communication tools can improve students' sense of support and connection—particularly in online learning contexts. For example, [2] found that the live chat features provided online and blended learners with a valuable channel for just-in-time academic support, and students particularly appreciated instant responses during assessment periods and when immediate help was needed. Similarly, [4] demonstrated that the Google Hangouts improved student satisfaction with online teamwork and helped students develop a sense of community in virtual settings, highlighting the value of real-time communication in online learning environments. Although ChatBook shares the goal of timely support, it differs in its asynchronous architecture, offering immediate contextualized AI assistance even when live human interaction is unavailable. Unlike live chat or video conferencing, ChatBook integrates directly into asynchronous discussion threads, offering responsiveness without requiring coordination across time zones or schedules.

Several studies have explored the integration of chatbots and AI-driven conversational agents in educational settings. [6] and [5] developed chatbot systems aimed at supporting collaborative learning, offering scaffolding and task-related feedback. Although these systems demonstrated positive impacts on teamwork and cognitive engagement, they did not consistently improve learning outcomes, emotional engagement, or motivation—partly due to limited contextual awareness and mistimed interventions. Similarly, [9] proposed the CHAT-ACTS framework to integrate chatbots with self-regulated learning and active learning models. Their work provides pedagogical guidance for chatbot integration, but focuses on theoretical principles rather than implementation with specific course materials. ChatBook builds on these efforts by incorporating RAG and aligning all AI output with instructor-provided course materials, such as textbooks and lecture slides. This ensures epistemic con-

sistency and mitigates the risk of conflicting or hallucinated explanations—a limitation in many generative AI tools [17].

[7] introduced EduBot, a curriculum-aligned language learning chatbot fine-tuned. EduBot outperformed ChatGPT in maintaining topical relevance and vocabulary alignment, demonstrating the benefits of tailoring AI systems to specific curricular content. ChatBook extends this principle by embedding RAG-based retrieval across a wide variety of instructional formats (not just language learning), including PDFs of textbooks, lecture slides, and handouts. Unlike EduBot, which is domain-specific, ChatBook is designed as a flexible cross-disciplinary platform for contextualized discussion support.

Several systems have addressed collaborative learning through technical affordances and social interaction modeling. [11] contrasted forums and instant messaging tools in Computer-supported collaborative learning environments, showing that discussion forums promote more knowledge-building interactions while instant messaging facilitates social interaction. [8] incorporated social networking awareness into a mobile assessment platform, finding that peer visibility and social influence enhanced sustained engagement. [12] developed PeerTalk to enable synchronous collaboration and AI support in MOOCs. Although these systems underscore the importance of peer-to-peer interaction, none explicitly embed AI assistance within student-led discussion workflows. ChatBook is designed not only to preserve collaborative learning but to amplify it by offering just-in-time scaffolding within the peer discourse itself, optionally triggered via 'Explain' or 'I am still confused' prompts. This model complements discussion rather than displacing it, aligning with social constructivist pedagogy [14].

The work most technically aligned with ChatBook is ChatEd [15],which integrates large language models with retrieval-based chatbots to support contextual accuracy in student queries. ChatEd focuses on LMS integration and response accuracy, outperforming ChatGPT on course-specific tasks. However, ChatEd functions more as an on-demand Q&A bot, whereas ChatBook is embedded within the ongoing discourse of a discussion board. The ChatBook iterative response mechanism (e.g., the feature "I am still confused") represents a novel contribution to the RAG+AI design space, allowing AI to consider previous context in shaping its reply.

3 System Design

Our system is built as a full-stack web application with a modular architecture that separates content processing, real-time communication, AI assistance, and user interface components. This design ensures scalability and maintainability while supporting the complex workflows required for intelligent educational

discussions. The backend foundation relies on Flask with Flask-SocketIO to enable real-time bidirectional communication, supporting both synchronous discussions and immediate AI assistance delivery. Authentication and security are managed through Auth0 for identity management and JWT tokens for session control, to ensure that students can only access their course-specific content.

Content processing begins when instructors upload PDF course materials to the system. PyPDF2 extracts raw text and metadata including page numbers and chapter headings, which are then stored in MongoDB Atlas. We chose MongoDB Atlas for its flexibility in handling both structured course metadata and unstructured textual content as vector embeddings that vary significantly across different educational materials. A critical design decision involves maintaining dual storage approaches to address competing technical requirements. Raw text content is processed and indexed for AI operations, as language models work most effectively with plain text for semantic understanding. However, students require properly formatted, page-specific content for effective learning, so original PDFs are preserved and rendered using PDF.js. This dual approach allows the system to display exact page references with proper formatting, figures, and layout when presenting AI-generated responses to students. The core intelligence emerges through our RAG pipeline, which processes student "Explain" queries by using OpenAI API embeddings for semantic similarity matching between student questions and stored course content. Once relevant content sections are identified, the OpenAI API generates contextually appropriate responses, with specific prompts to reference page numbers from course materials. For the "I am still confused" feature, we use Google Gemini API for analyzing the last five discussion messages to provide increasingly targeted explanations such as generating pseudocode and maintaining conversation context throughout the learning process. The frontend implementation uses HTML5/CSS3 with Bootstrap for responsive design across different devices. Socket.IO manages real-time message delivery to ensure that students see new discussion posts and AI responses immediately, while Marked.js handles markdown rendering and Highlight.js provides syntax highlighting for code snippets. The seamless PDF.js integration allows students to view specific referenced pages without leaving the discussion context, maintaining optimal learning flow from AI explanation to source material review. Security and performance considerations include HTTPS communication, CORS policies, and CSRF protection for secure operations. Performance optimization utilizes Gzip compression and browser caching, while lazy loading efficiently handles large PDF documents and extensive discussion threads that could otherwise impact system responsiveness. The deployment strategy supports both development and production environments through virtual environments and LocalTunnel

for testing, while production deployment utilizes Gunicorn with Nginx as a reverse proxy for load balancing and static file serving. This architecture successfully enables multiple concurrent discussions with immediate, contextually relevant AI assistance grounded in instructor-curated content, scaling effectively to support large courses while maintaining educational effectiveness.

4 Features and Capabilities

Although formal user testing and deployment in actual courses remain part of our future work—including planned randomized controlled trials comparing our system against Piazza following IRB approval—this section demonstrates the current capabilities and pedagogical value of our application through practical examples of student-AI interactions.

The core functionality of ChatBook centers on its ability to respond to student queries by retrieving relevant course content and generating contextualized, pedagogically aligned explanations. This feature addresses a critical challenge in asynchronous learning environments: providing timely, content-specific help when human assistance may be delayed or unavailable [2, 4]. Immediate feedback on code comprehension can support learning in introductory programming courses, where students frequently struggle with algorithmic abstraction [10].

In Figure 1, we show a typical use case in which two students engage in a discussion about graph theory. When a student invokes the AI with the phrase "Explain graphs," the system retrieves specific page references (2, 6, 7, and 8) from the uploaded textbook and displays them alongside a tailored explanation. This design aligns conversational practice with curriculum content, ensuring students engage with topics and vocabulary from their textbooks [7].

ChatBook supports seamless transitions in discussion dynamics as students join or leave an ongoing conversation. New participants can review the full discussion history to quickly gain context. In the example shown in Figure 2, while the roster of participants changes, an existing student, Saloni, requests help with "weighted graphs." Our system responds by retrieving relevant material and providing a focused explanation. This design provides adaptive scaffolding that responds to students' evolving needs during collaborative learning [5, 1].

A distinctive pedagogical benefit lies in the system's support for page-specific content retrieval. Figure 3 shows how our system can focus entirely on the content of a single page (e.g., page 6) to provide fine-grained and localized explanations. This mirrors the instructional value of worked examples in programming education and facilitates incremental comprehension—a critical skill in early CS learning [10]. By allowing students to query and reflect on a specific page's content, the system turns passive reading into an interactive process.

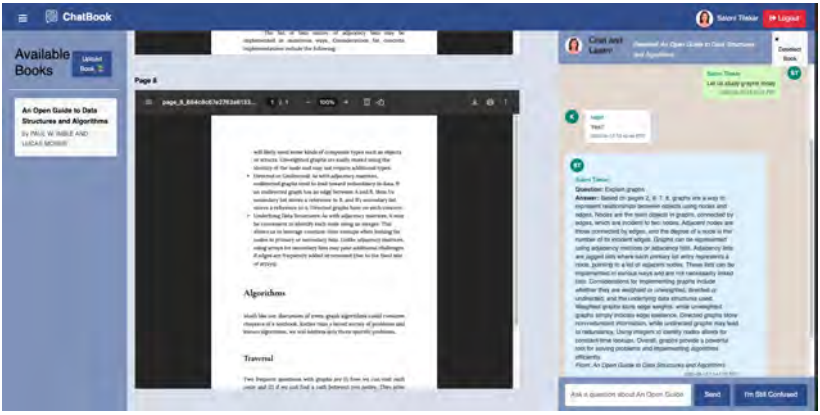

Figure 1: Students engaging in discussion with AI-triggered explanation of graphs, showing relevant textbook pages and AI-generated summary.

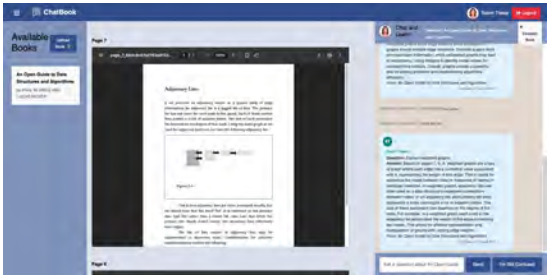

Figure 2: Dynamic discussion showing student participation changes and AI response to weighted graph query with specific page references.

Such targeted engagement exemplifies active learning, which has been shown to enhance motivation and learning outcomes in introductory programming courses [3].

Figure 4 shows the system's content alignment enforcement feature. When a student asks about a topic not covered in the uploaded instructional materials (e.g., parallel programming), our system explicitly states that the topic lies outside its current scope. This maintains alignment with instructor-provided content, promoting epistemic fidelity and reducing the risk of misinformation—a concern noted in previous evaluations of generative AI in education [15].

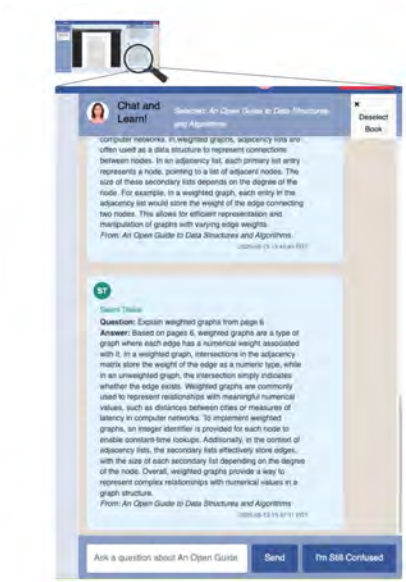

Figure 3: Student requesting detailed explanation of weighted graphs from specific page 6, showing AI's ability to focus on particular content sections.

Figure 4: AI appropriately responding that parallel programming information is not available in the course content, maintaining course-specific boundaries.

In Figure 5, our system generates pseudocode in response to a query about directed graphs. Because the relevant pages only describe adjacency lists, the AI adapts its response accordingly—demonstrating a key instructional design goal: help students generalize from partial knowledge without overreaching the curriculum boundaries.

Figure 5: AI generating pseudocode for directed graphs based on adjacency list representation from page 7 of the textbook.

Figure 6: Contextual response when student clicks "I am still confused," incorporating both course content and discussion thread context.

The "I am still confused" feature, shown in Figure 6, allows students to solicit more nuanced help. By analyzing the five previous messages in the discussion, the AI offers context-sensitive elaborations. This strategy embodies the principles of effective question-asking during problem-solving [1] and provides synchronous support that students find valuable for just-in-time help [2]. Together, these capabilities position ChatBook as a scalable learning companion that supports active, self-regulated, and peer-mediated learning. Its design reflects effective pedagogical principles: students benefit the most when instruction is interactive, provides real-time support, and encourages active problem-solving [3, 1]. Rather than replacing peer or instructor feedback, ChatBook complements it by ensuring that students have continuous access to support that reinforces instructor-defined goals and curriculum coverage.

5 Conclusion

This paper presents an intelligent discussion board system designed to address the growing need for AI-enhanced educational tools that maintain pedagogi-

cal integrity while providing immediate learning support. By integrating RAG with instructor-curated content, our system bridges the gap between collaborative peer learning and contextual AI assistance in introductory CS education. The key innovation of our system lies in its constraint-based approach to AI integration. Rather than relying on general-purpose AI tools that may provide inconsistent or inappropriate guidance, our system ensures that all AI responses are grounded in official course materials uploaded by instructors.

Acknowledgment

We gratefully acknowledge the use of OpenAI's ChatGPT for proofreading, grammatical checks, and other text editing tasks.

References

[1] Kristy Elizabeth Boyer et al. "Principles of asking effective questions during student problem solving". In: *Proceedings of the 41st ACM technical symposium on Computer science education*. 2010, pp. 460–464.

[2] Jaclyn Broadbent and Jason Lodge. "Use of live chat in higher education to support self-regulated help seeking behaviours: a comparison of online and blended learner perspectives". In: *International journal of educational technology in higher education* 18.1 (2021), p. 17.

[3] Ricardo Caceffo, Guilherme Gama, and Rodolfo Azevedo. "Exploring active learning approaches to computer science classes". In: *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 2018, pp. 922–927.

[4] Jinxia He and Xiaoxia Huang. "Collaborative online teamwork: Exploring students' satisfaction and attitudes with google hangouts as a supplementary communication tool". In: *Journal of Research on Technology in Education* 49.3-4 (2017), pp. 149–160.

[5] Wanqing Hu, Jirong Tian, and Yanyan Li. "Enhancing student engagement in online collaborative writing through a generative AI-based conversational agent". In: *The Internet and Higher Education* 65 (2025), p. 100979.

[6] Jeya Amantha Kumar. "Educational chatbots for project-based learning: investigating learning outcomes for a team-based design course". In: *International journal of educational technology in higher education* 18.1 (2021), p. 65.

[7] Yu Li et al. “Curriculum-driven edubot: A framework for developing language learning chatbots through synthesizing conversational data”. In: *arXiv preprint arXiv:2309.16804* (2023).

[8] Jian-Wei Lin, Hsieh-Hong Huang, and Chia-Wen Tsai. “Mobile collaborative learning with social network awareness”. In: *Journal of Computer Assisted Learning* 39.5 (2023), pp. 1690–1700.

[9] Michael Pin-Chuan Lin and Daniel Chang. “CHAT-ACTS: A pedagogical framework for personalized chatbot to enhance active learning and self-regulated learning”. In: *Computers and Education: Artificial Intelligence* 5 (2023), p. 100167.

[10] Leigh Ann Sudol-DeLyser, Mark Stehlik, and Sharon Carver. “Code comprehension problems as learning events”. In: *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*. 2012, pp. 81–86.

[11] Zhong Sun et al. “A tale of two communication tools: Discussion-forum and mobile instant-messaging apps in collaborative learning”. In: *British Journal of Educational Technology* 49.2 (2018), pp. 248–261.

[12] Stergios Tegos, Apostolos Mavridis, and Stavros Demetriadis. “Agent-supported peer collaboration in MOOCs”. In: *Frontiers in Artificial Intelligence* 4 (2021), p. 710856.

[13] Mickey Vellukunnel et al. “Deconstructing the Discussion Forum: Student Questions and Computer Science Learning”. In: *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. SIGCSE ’17. Seattle, Washington, USA: Association for Computing Machinery, 2017, pp. 603–608. ISBN: 9781450346986. DOI: 10.1145/3017680.3017745. URL: https://doi.org/10.1145/3017680.3017745.

[14] Lev S Vygotsky. *Mind in society (M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Eds.)* 1978.

[15] Kevin Wang, Jason Ramos, and Ramon Lawrence. “ChatEd: a chatbot leveraging ChatGPT for an enhanced learning experience in higher education”. In: *arXiv preprint arXiv:2401.00052* (2023).

[16] TII Washington et al. “Today’s discussion boards: the good, the bad, and the ugly”. In: *The Online Journal of New Horizons in Education* 9.3 (2019), p. 219.

[17] Huiwen Xiong. “Research on confusing responses based on ChatGPT”. In: *Applied and Computational Engineering* 57 (2024), pp. 90–97.