2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #50133

# Work in Progress: GUIDE—An AI-Powered Co-Pilot for Student Academic Pathways in Engineering

**Zhuo Zhang, University of Toronto**
**Dr. Hamid S Timorabadi P.Eng., University of Toronto**

Hamid Timorabadi received his B.Sc, M.A.Sc, and Ph.D. degrees in Electrical Engineering from the University of Toronto. He has worked as a project, design, and test engineer as well as a consultant to industry. His research interests include the applications in Engineering Education.

# GUIDE—An AI-Powered Co-Pilot for Student Academic Pathways in Engineering

**Abstract**
Flexible engineering curricula offer students significant autonomy, yet many struggle to translate this flexibility into coherent, career-aligned academic pathways. Prior work suggests that this challenge introduces substantial extraneous cognitive load, often leading students to select courses based on convenience rather than strategic skill development. This Work-In-Progress paper reports on the design and deployment of GUIDE (Guided University Interface for Degree Exploration), an AI-assisted academic planning platform intended to support exploratory reasoning about degree pathways rather than automate advising decisions.

GUIDE employs a hybrid Retrieval-Augmented Generation (RAG) architecture backed by a Neo4j knowledge graph. Unlike standard Large Language Models (LLMs), GUIDE utilizes a deterministic slot-filling methodology to map natural language career goals to verified curricular opportunities without hallucination. This ensures pathways are not only degree-compliant but professionally relevant, bridging the gap between academic choices and skill development. Validation with engineering students demonstrates the platform's efficacy, yielding 100% positive usability ratings and indicating a significant reduction in administrative cognitive load.

## Introduction

Undergraduate engineering curricula are increasingly designed with flexibility in mind, allowing students to shape their learning to align with emerging technologies and specific career goals. In the Electrical and Computer Engineering (ECE) program at our institution, this flexibility manifests as a large and diverse set of technical electives and specialization areas. Undergraduate students are presented with approximately 90–120 distinct ECE courses alone, not including additional technical electives available from other departments such as Computer Science. Navigating this breadth of options under prerequisite, scheduling, and degree constraints presents a significant cognitive and decision-making challenge for students.

While this flexibility is intended to help students grow, it often creates a disconnect between academic choices and professional outcomes. Without structured guidance, students struggle to identify which combinations of courses translate into coherent skill sets. This often leads to selecting courses based on scheduling convenience or vague descriptions rather than their relevance to a desired career path or technical skill [1].

The cognitive load required to bridge this gap is substantial. To plan a meaningful pathway, a student must currently synthesize information from fragmented sources: the university catalog (for course content), industry trends (for skill relevance), and their own developing interests. Existing digital planning tools, such as Elcano, function primarily as passive repositories [2]. While foundational tools have successfully aggregated fragmented information sources and provided a flexible interface for course planning, they remain passive systems that leave the burden of strategic synthesis entirely on the student. Consequently, students often graduate with a grab bag of credits rather than a “T-shaped” professional profile, missing opportunities to develop deep expertise in areas like machine learning or embedded systems [3].

Recent advancements in Large Language Models (LLMs) offer a potential solution for personalized, career-aligned guidance. However, off-the-shelf models (e.g., Gemini 3) are ill-suited for specific academic advising due to their lack of institutional context [4]. A standard chatbot might recommend generic engineering concepts but cannot map a student's interest in "autonomous vehicles" to the specific, code-designated electives offered by the university that teach those skills [5].

To bridge this gap, we introduce GUIDE (Guided University Interface for Degree Exploration), an AI-powered co-pilot designed to transform academic planning from a compliance exercise into a career-focused exploration. GUIDE advances beyond passive tools by employing a hybrid Retrieval-Augmented Generation (RAG) architecture backed by a Neo4j knowledge graph. Unlike standard LLMs, GUIDE utilizes a semantic search engine to map natural language career goals (e.g., "I want to work in robotics") to specific curricular opportunities. By grounding AI reasoning in verified course data, GUIDE empowers students to build pathways that are not only degree-compliant but professionally coherent, ensuring their academic journey aligns with their long-term aspirations. In this paper, we present the system architecture, design, and results from a validation study that examines the platform's ability to process curricular information and reduce planning-related anxiety. The study focuses on the graph-based planning interface, which supports students in navigating the tension between rigid academic requirements and flexible, interest-driven course selection.

**Background and Motivation**

Cognitive Load Theory (CLT) provides a useful framework for understanding the burden placed on engineering students. CLT distinguishes between intrinsic load which is the inherent difficulty of the subject matter (e.g., learning electromagnetics), and extraneous load which is the mental effort imposed by the way information is presented or organized [6]. Engineering students already face high intrinsic load due to the rigorous nature of their coursework. The administrative task of navigating a flexible curriculum adds a massive layer of extraneous load. To make a valid decision, a student must simultaneously hold multiple constraints in working memory: scheduling logistics, degree requirements, and personal interests. When this total load exceeds cognitive capacity, students often abandon strategic planning (germane load) and resort to heuristic shortcuts, such as selecting "easy" electives or copying peers, rather than optimizing for their own professional development [7]. GUIDE is designed specifically to offload this extraneous burden, automating the logistics so students can focus on the strategic alignment of their education.

While Generative AI offers a powerful interface for reducing this cognitive load, the underlying architecture of Large Language Models (LLMs) presents significant risks in an academic context. As described by Bender et al., LLMs function as "stochastic parrots", probabilistic engines that predict the next likely token based on statistical patterns in their training data, rather than reasoning engines grounded in factual truth [4][8].

In the domain of academic advising, this probabilistic nature manifests in two dangerous ways:

1. Entity Hallucination: LLMs often generate plausible-sounding but non-existent entities. For example, when asked to recommend a course on "robotics", a standard model might

invent a course code like "ECE 499: Robotics and Embedded System". While the title sounds appropriate, the course does not exist in the university catalog.

2. Constraint Blindness: LLMs struggle with strict Boolean logic and negative constraints. A model might recommend a valid course while failing to recognize that it is only offered in the Fall semester, or that it conflicts with a mandatory core requirement.

This creates a trust gap. In creative writing, a 95% accuracy rate is acceptable; in degree planning, a single error regarding a prerequisite or offering term can result in a delayed graduation. Consequently, a naked chatbot cannot be safely deployed for advising. The system requires a Retrieval-Augmented Generation (RAG) architecture, where the generative capabilities of the LLM are strictly constrained by a deterministic retrieval layer that grounds every response in verified institutional data [9].

While curriculum data can be stored in traditional relational databases (SQL), the rigid tabular structure of SQL is ill-suited for the inherent complexity of engineering pathways. A curriculum is not a list of isolated records. Rather, it is a dense network of dependencies, prerequisites, and thematic clusters. Modeling this in SQL requires complex, recursive joins that are computationally expensive and difficult to query for deep insights.

A Knowledge Graph, implemented here using Neo4j, offers a superior representation by modeling courses as nodes and requirements as edges. Crucially, this architecture enhances the RAG pipeline by decoupling structure from semantics. In a standard vector-only RAG system, one would attempt to embed these complex relationships into the vector space, relying on the model to infer connections based on proximity. This is error-prone for strict academic constraints. By using a Graph, we do not need to embed the relationships since they are stored explicitly as traversable edges. This allows the system to perform hybrid retrieval that is using vector embeddings to find courses that match a student's interests (semantic search), while simultaneously traversing the graph to ensure those courses fit the prerequisite structure (structural retrieval) [10].

**System Architecture**

The GUIDE platform is built upon a hybrid Retrieval-Augmented Generation (RAG) architecture. This architecture is designed to bridge the gap between the rigid logic of university regulations and the flexible, natural-language reasoning of modern AI, as illustrated in Figure 1. The system is organized into four distinct layers: a Client Layer that provides the interactive React-based user interface; an Application Layer built on Python and LangChain that handles orchestration, API routing, and state management; a Data Layer that manages structural and semantic storage via a Neo4j graph database and vector index; and a Model Service Layer that integrates external Large Language Model (LLM) providers to supply reasoning and generation capabilities. Unlike standard RAG implementations that rely solely on vector similarity, GUIDE employs a dual-retrieval strategy. It combines Structural Retrieval (traversing the graph to enforce hard constraints like prerequisites), with Semantic Retrieval (using vector embeddings to match course information with student interests) [11][12].

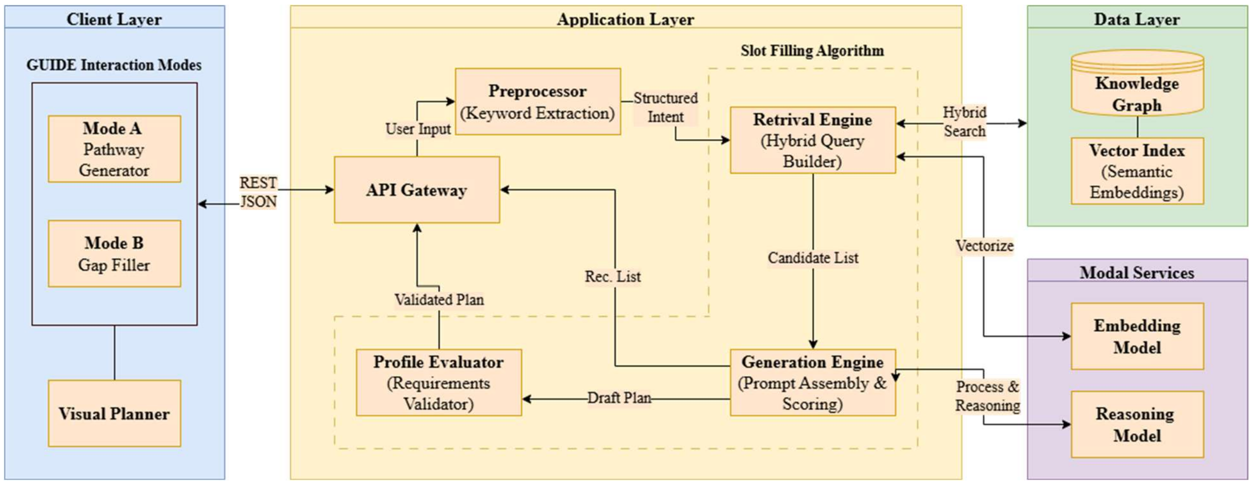


**Figure 1: High-Level System Architecture**

To support this hybrid retrieval, the curriculum is modeled not as a static table, but as a Knowledge Graph. We utilize a schema where Course nodes represent atomic curricular units (containing properties like course code, description and credits) and Application Area nodes represent semantic clusters (e.g., "Machine Learning and Artificial Intelligence", "Robotics / Control Systems"). An illustrative schema demonstrating these relationships is shown in Figure 2. The logic of the curriculum is encoded in the edges: HAS_PREREQUISITE relationships form a Directed Acyclic Graph (DAG) used for validity checking, while HAS_APPLICATION_AREA relationships carry a weighted relevance score (1-3) to drive interest-based recommendations. This graph-based representation allows the system to perform complex traversals, such as identifying all downstream courses unlocked by a specific prerequisite, that would be computationally prohibitive in a traditional relational database [13][14]. (Note: A comprehensive visualization of the full subgraph centered on ECE311: Introduction to Control Systems is provided in Appendix A).


**Figure 2: Conceptual Knowledge Graph Schema**

The core of the recommendation engine lies in its ability to synthesize these structural and semantic signals. When a student queries the system, GUIDE employs a Hybrid Scoring Algorithm to rank potential courses. First, Structural Queries (Cypher) filter the graph to identify the set of valid courses where all prerequisites are met. Simultaneously, Semantic Search computes the cosine similarity between the student's natural language intent (e.g., "I want to work in robotics") and pre-computed course embeddings. These signals are combined using a weighted formula:

$$Score = w_1(Prerequisite) + w_2(Interest) + w_3(Requirement)$$

This weighted formula allows the recommendation engine to adapt dynamically to the student's current planning state [15]. In Mode A (Pathway Generator), the system adopts a proactive stance by maximizing $w_2$ (Interest). This heavily prioritizes courses aligned with the student's semantic career goals (e.g., "autonomous vehicles") to construct a cohesive, fully degree-compliant long-term roadmap. Because the Slot-Filling engine guarantees the structural validity of this initial plan, the student can then transition to Mode B (Gap Filler) with confidence to personalize their trajectory. In this mode, students can explore alternative electives, swap courses for similar topics, or adjust for personal scheduling preferences. To safely support these manual adjustments, the system shifts its weighting to prioritize $w_1$ (Prerequisite) and $w_3$ (Requirement). This ensures that any newly discovered courses the student wishes to swap into their plan are immediately accessible and satisfy remaining graduation constraints, effectively answering the localized "What else am I eligible to take right now?" question while preserving overall degree compliance.

To address the critical issue of LLM hallucination, GUIDE utilizes a Deterministic Slot-Filling methodology for pathway generation, illustrated in Figure 3. Rather than asking the LLM to generate a plan from scratch, which risks the invention of non-existent courses, the system algorithmically fills pre-defined semester slots. For a given term, the system defines specific slots (e.g., Kernel Course, Depth Elective, Complementary Studies). For each slot, the backend executes a specific database query to retrieve the best existing course that fits the criteria. The LLM is then invoked as a Semantic Selector. It evaluates this list of valid candidates and selects the optimal course for the slot based on the user's nuanced context, subsequently generating a natural language explanation for the choice. This approach guarantees 100% existence of recommended courses while preserving the personalized "co-pilot" experience [16].

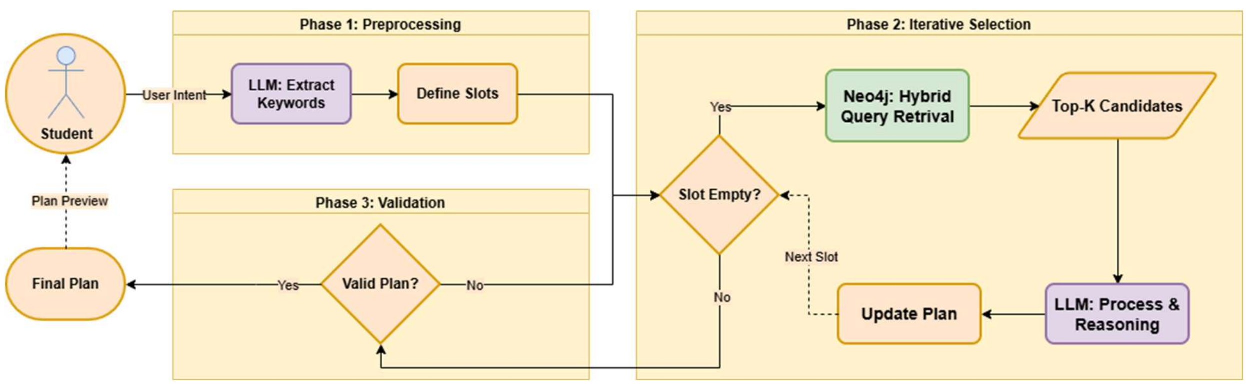


**Figure 3: The Slot-Filling Process**

Finally, all generated pathways are validated by the Profile Evaluator, a rule-based engine that models degree requirements as a Constraint Satisfaction Problem (CSP). The evaluator applies constraints in a strict priority order to minimize resource contention. Deterministic Constraints (mandatory courses like Engineering Economics) are applied first, followed by Combinatorial Constraints (selecting k courses from n areas), and finally Cardinality Constraints (total credit counts). This priority-ordered evaluation ensures that scarce courses, those that satisfy unique requirements, are allocated efficiently, preventing the system from suggesting a plan that is mathematically impossible to complete [17].

**Implementation and User Experience**

The GUIDE platform is implemented as a modern single-page application (SPA) built using React.js, featuring a responsive design that ensures full functionality across both desktop and mobile devices. To provide an intuitive interaction of multi-year planning, the interface incorporates a drag-and-drop system, allowing students to manipulate their course grids seamlessly, shown in Figure 4. This frontend communicates with a high-performance Python/FastAPI backend for handling concurrent RAG requests without blocking the user interface. The backend acts as the central orchestrator, routing requests between the Neo4j graph database, the vector search index, and the AI model layer [18].

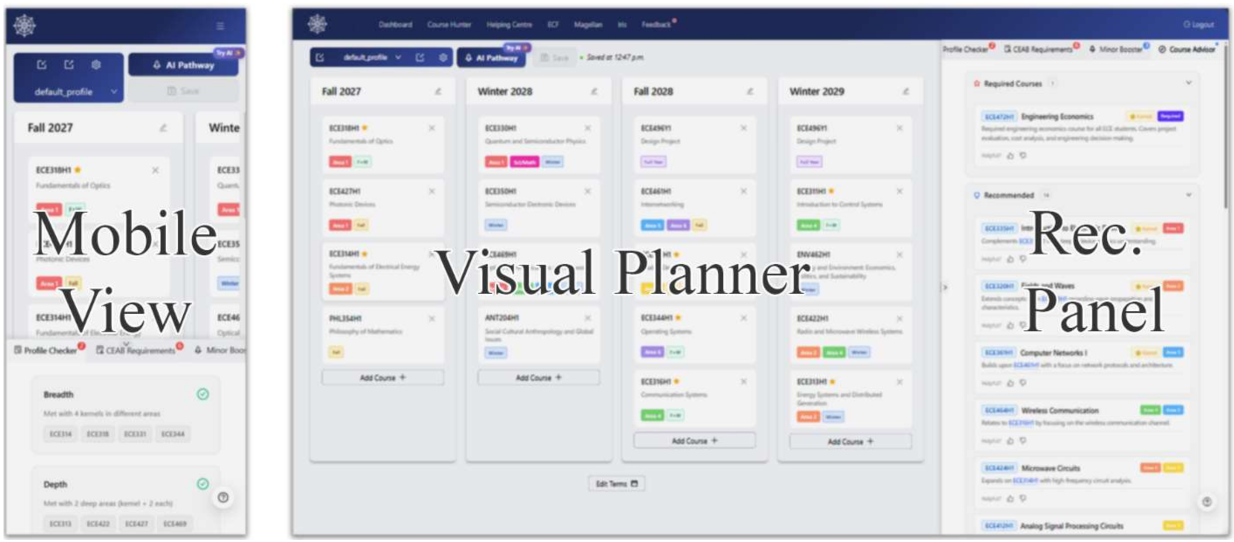


**Figure 4: User Interface Markups**

To manage the complexity of the AI interactions, we employ LangChain not only for orchestration but as a comprehensive observability pipeline. LangChain manages the Chain of Thought (CoT) flow, dynamically routing queries to OpenRouter, which provides model agnosticism that allowing the system to switch between high-reasoning models (e.g., Gemini 3 Pro) for complex pathway generation and lighter, faster models (e.g., Llama3 8B) for simple queries. Crucially, this layer instruments real-time performance metrics, capturing Time to First Token (TTFT) and Tokens Per Second (TPS). These metrics are utilized for system benchmarking and performance monitoring, ensuring that the latency of the retrieval and generation pipelines remains within acceptable limits for an interactive user experience [19]. Furthermore, the implementation includes an integrated feedback loop: every AI-generated plan or course explanation includes in-line binary feedback controls (thumbs-up/thumbs-down). This user feedback is logged directly alongside the generation metadata, creating a labeled dataset for future fine-tuning of the prompt engineering strategies.

The user experience is divided into two distinct modes of interaction, designed to support different stages of the planning lifecycle. For students starting from scratch or seeking a complete overhaul, Mode A (Pathway Generator) offers a guided pathway generation experience implemented as a modal overlay. This wizard abstracts the complexity of the curriculum into a high-level conversation. The flow begins with the student selecting broad interest areas or typing specific keywords (e.g., “robotics”, “control systems”). The system then executes the Deterministic Slot-Filling algorithm in the background, querying the graph to construct a valid multi-year skeleton. The user is then presented with a Plan Preview, allowing them to review the proposed trajectory before instantiating it as a new academic profile. As illustrated in Figure 5, this preview displays a single-year snippet of the generated schedule alongside a dedicated section below the course list explaining the AI's holistic reasoning. Once created, this plan is fully editable; student can freely rearrange, add, or remove courses using the visual planner.

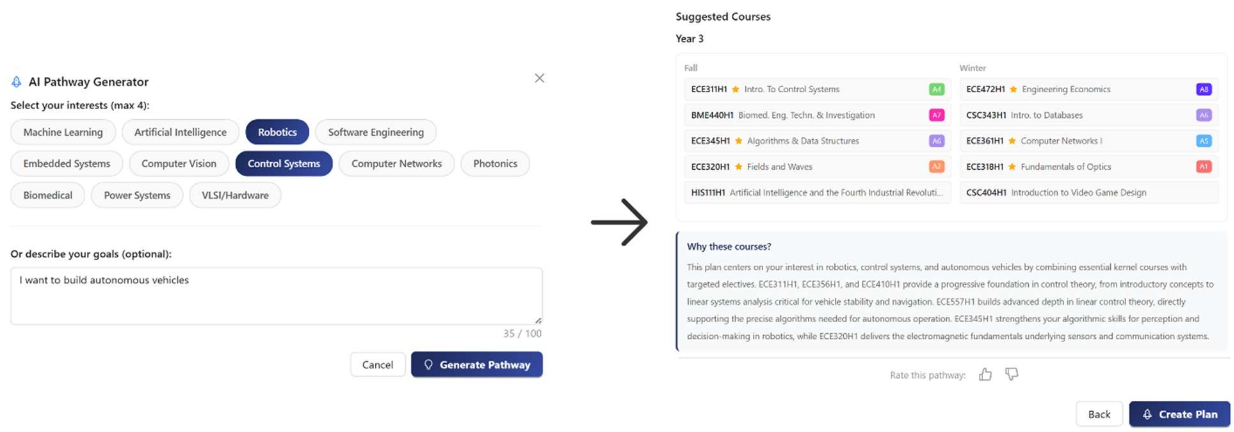


**Figure 5: Pathway Generator Modal**

To support the refinement of generated plans, Mode B (Gap Filler) provides a reactive dashboard interface. Within the responsive layout shown in Figure 4, the visual planner and dynamic Recommendation Panel transition from side-by-side desktop interface to stacked mobile view. This panel categorizes suggestions into three distinct tiers based on the Hybrid Scoring algorithm: Required (mandatory courses needed for graduation), Recommended (courses where high interest matches satisfied prerequisites), and Explorer (high-interest courses that may require future prerequisites). To highlight the AI's text-based reasoning while maintaining document legibility, Figure 6 presents a condensed, rearranged view of these three categories. This categorization helps students distinguish between immediate obligations and long-term aspirations [20]. As a student saves their plan, the recommendation engine instantly recalculates and the interface updates in real-time to show the next logical steps.

The power of GUIDE lies in the seamless integration of these two modes. A typical user journey might begin in the Pathway Generator (Mode A), where a student generates a base plan focused on “Control Systems”. Once this skeleton is instantiated, the student switches to the Gap Filler (Mode B) to refine the trajectory and broaden their skill set. Here, the Recommendation Panel proactively surfaces “ECE316: Communication Systems” under the Recommended category. The system calculates a high semantic score by identifying a strong latent overlap between this course and the student’s declared focus areas. Seeing this suggestion, the student adds ECE316 into their schedule to replace a generic technical elective. This interaction demonstrates the system’s capacity to bridge the semantic gap: it translates a broad interest in automation technologies into specific, code-designated coursework without requiring the student to manually search the catalog, effectively turning the planning process from a search task into a discovery experience [21].

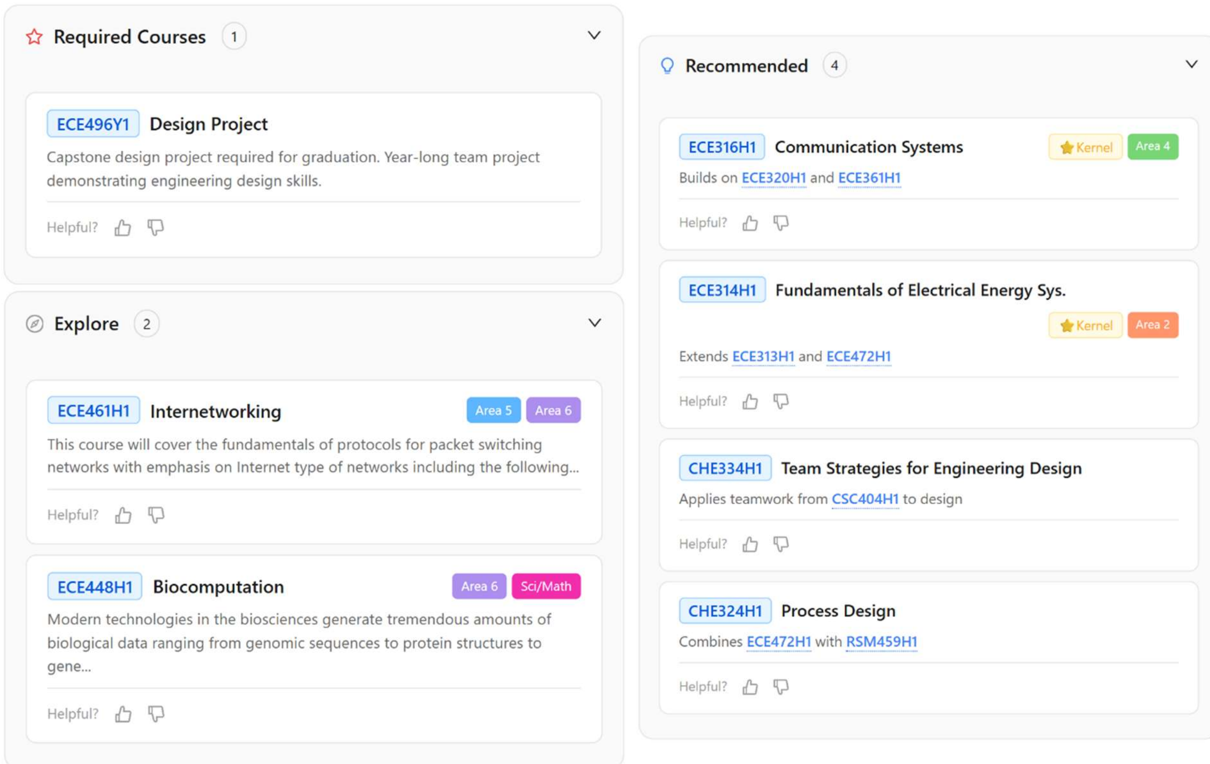


**Figure 6: The Recommendation Panel**

## Results and Evaluation

Evaluation of the GUIDE system was conducted to assess both the human-centric usability of the platform and the overall technical and logical efficacy of the underlying AI architecture. A comprehensive pilot evaluation was conducted with a diverse cohort of undergraduate ECE students across different years of study (N=9). Participants were tasked with using both the Pathway Generator (Mode A) and the Gap Filler (Mode B) to construct career-aligned degree plans, followed by a comprehensive system evaluation survey.

As summarized in Figure 7, survey results indicated strong user acceptance and a notable reduction in administrative cognitive load. Overall, eight of the nine participants rated their overall experience a 4 or 5 (out of 5), while all nine participants reported that the AI co-pilot was significantly faster than manual planning. Demonstrating high practical utility, students retained an average of 50% of the AI-generated pathway. In this workflow, the AI automates the foundational skeleton and core constraints, while students subsequently explore, swap, and personalize the reminder of their schedule. Qualitative feedback praised the system's semantic alignment in successfully translating abstract keywords like "robotics" into relevant suggestions, as well as the clear LLM-generated reasoning and seamless UI integration.

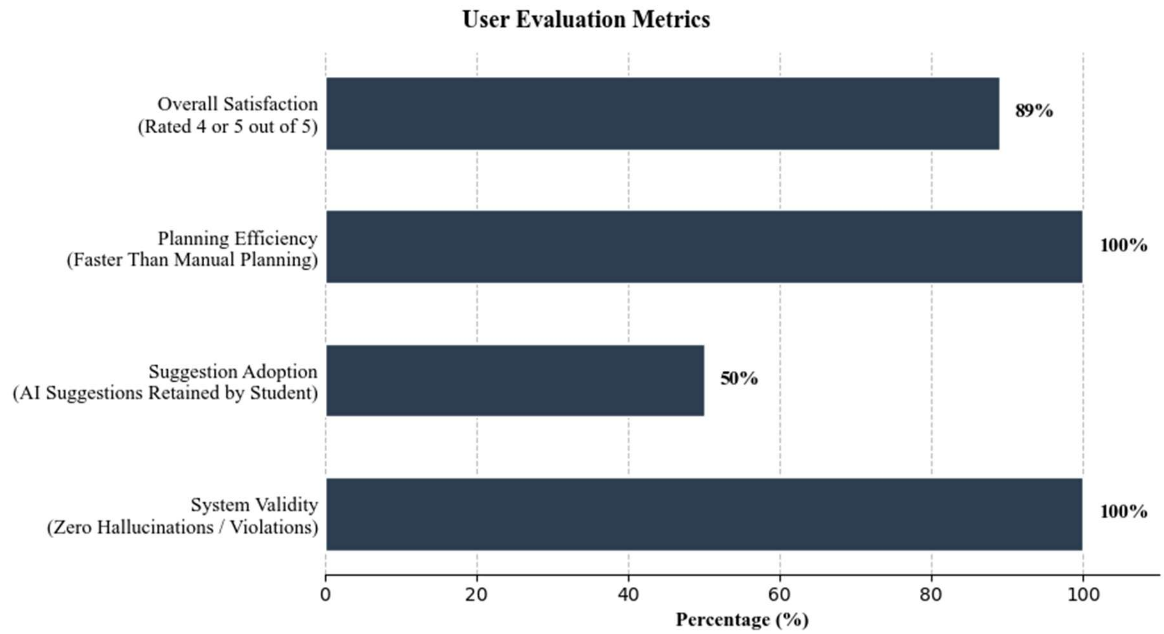


**Figure 7: User Evaluation Metrics**

From a technical perspective, the hybrid RAG architecture was engineered to balance complex constraint validation with user responsiveness. System latency was optimized through minimized token payloads, batched inference, and task-specific dual-model routing. For complex reasoning, such as multi-year pathway generation, high-capability models (e.g., Gemini 3 Pro) achieving a median end-to-end Time to First Token (TTFT) of 5.39 seconds at 60 Tokens Per Second (TPS). Conversely, auxiliary tasks like keyword extraction leveraged lightweight models (e.g., Llama3 8B), achieving a 0.5-second TTFT at 95 TPS.

The deterministic slot-filling algorithm achieved a 100% validity rate, with zero reported hallucinations or prerequisite violations. By restricting the LLM to reasoning over a pre-filtered context of valid database entities, the architecture successfully balanced generative flexibility with the strict logical constraints required for high-stakes academic advising

**Discussion**

The primary barrier to adopting Generative AI in institutional settings is the trust gap. In high-stakes academic advising, where a single erroneous recommendation can delay graduation, the probabilistic nature of standard LLMs is a liability. GUIDE addresses this by fundamentally decoupling reasoning from knowledge. In our architecture, the LLM is never treated as a knowledge store; it is never asked to remember course codes or prerequisite rules. Instead, the Neo4j Knowledge Graph serves as the immutable source of truth (the Knowledge), while the LLM serves as the semantic processor (the Reasoning).

This hybrid approach that combining the deterministic rigidity of a graph database with the semantic flexibility of an LLM, represents a blueprint for trustworthy educational AI. By restricting the LLM's role to explaining and selecting from a pre-validated list of options, we effectively eliminate the risk of hallucination. This architecture allows institutions to leverage the empathetic and explanatory power of Generative AI without exposing themselves to the liability of hallucinated advising, ensuring that the system remains a helpful co-pilot rather than an unreliable narrator [22].

Beyond technical reliability, GUIDE facilitates a pedagogical shift from compliance to identity formation. Traditional planning tools force students to think in the language of the university catalog (e.g., “I need an elective from this list”), which often leads to a tick-box mentality. However, students typically think in the language of their aspirations (e.g., “I want to build autonomous vehicles”). There’s a noticeable gap between these two worlds.

GUIDE bridges this gap through its semantic search capabilities. By mapping natural language career goals to specific curricular opportunities, the system helps students visualize how abstract requirements translate into concrete professional skills. A student interested in “robotics” is no longer left guessing which electives are relevant; the system explicitly links that interest to “ECE470: Robot Modeling and Control” and its prerequisites “ECE311: Introduction to Control Systems”. This transforms the planning process from an administrative burden into an exercise in professional identity formation, empowering students to build a “T-shaped” profile with deep expertise in their chosen domain rather than a disjointed collection of credits [23].

While GUIDE was developed within the context of an Electrical and Computer Engineering (ECE) program, the underlying architecture is domain-agnostic. The graph schema is a universal representation of academic curricula. This framework can be readily adapted to other engineering disciplines, such as Mechanical or Civil Engineering, or even interdisciplinary programs by simply ingesting a different dataset. The constraint satisfaction logic is similarly modular. This suggests that the Graph-RAG approach is a scalable solution for university-wide academic advising, capable of handling the unique logic of diverse departments without requiring a redesign of the core reasoning engine [24].

## Conclusion

The increasing flexibility of modern engineering curricula, while enabling specialization, places a significant cognitive burden on students tasked with constructing coherent academic pathways. GUIDE addresses this challenge by reframing degree planning from a passive administrative activity into an active, AI-assisted process that supports student reasoning and decision-making. By combining a hybrid Retrieval-Augmented Generation (RAG) approach with a Neo4j-based Knowledge Graph, the platform demonstrates a practical and trustworthy method for supporting academic planning in high-stakes educational contexts.

The validation results presented in this paper indicate that centralizing curricular information and enforcing degree constraints through a graph-based interface can meaningfully reduce planning-related anxiety and administrative cognitive load. Rather than replacing human advising, GUIDE functions as a structured co-pilot that helps students navigate complex requirements while maintaining agency over their academic choices. Together, these findings contribute early evidence that carefully constrained AI systems can play a constructive role in supporting student sense-making and pathway coherence in engineering education.

## References


[1] Jensen, D., and Jetten, J. “The Importance of Developing Students’ Academic and Professional Identities in Higher Education.” Journal of College Student Development. 2016; 57. https://doi.org/10.1353/csd.2016.0097.

[2] Mahidashti, D., Yang, B., Li, L., Srivart, K., and Timorabadi, H. “ELCANO: ENHANCING ACADEMIC PATHWAYS FOR COURSE PLANNING & CAREER ALIGNMENT IN HIGHER EDUCATION.” Proceeding of the 2024 8th International Conference on Education and E-Learning. 2024 https://doi.org/10.1145/3719487.3719507.

[3] Moore, E., Weitnauer, M., and Nagel, K. “Providing flexible and career-correlated paths through the undergraduate electrical and computer engineering curriculum.” 2019 IEEE Frontiers in Education Conference (FIE). 2019 https://doi.org/10.1109/fie43999.2019.9028425.

[4] Xu, Z., Jain, S., and Kankanhalli, M. “Hallucination is Inevitable: An Innate Limitation of Large Language Models.” ArXiv. 2024; abs/2401.11817. https://doi.org/10.48550/arxiv.2401.11817.

[5] Lin, Z., Guan, S., Zhang, W., Zhang, H., Li, Y., and Zhang, H. “Towards trustworthy LLMs: a review on debiasing and dehallucinating in large language models.” Artificial Intelligence Review. 2024; 57. https://doi.org/10.1007/s10462-024-10896-y.

[6] Sweller, J. “Cognitive Load During Problem Solving: Effects on Learning.” Cogn. Sci.. 1988; 12. https://doi.org/10.1207/s15516709cog1202_4.

[7] Skulmowski, A., and Xu, K. “Understanding Cognitive Load in Digital and Online Learning: a New Perspective on Extraneous Cognitive Load.” Educational Psychology Review. 2021; 34. https://doi.org/10.1007/s10648-021-09624-7.

[8] Bender, E., Gebru, T., McMillan-Major, A., and Shmitchell, S. “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜.” Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. 2021 https://doi.org/10.1145/3442188.3445922.

[9] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kuttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.” ArXiv. 2020; abs/2005.11401.

[10] Li, Y., Liang, Y., Yang, R., Qiu, J., Zhang, C., and Zhang, X. “CourseKG: An Educational Knowledge Graph Based on Course Information for Precision Teaching.” Applied Sciences. 2024 https://doi.org/10.3390/app14072710.

[11] Peng, B., Zhu, Y., Liu, Y., Bo, X., Shi, H., Hong, C., Zhang, Y., and Tang, S. “Graph Retrieval-Augmented Generation: A Survey.” ArXiv. 2024; abs/2408.08921. https://doi.org/10.1145/3777378.

[12] Zhu, X., Xie, Y., Liu, Y., Li, Y., and Hu, W. “Knowledge Graph-Guided Retrieval Augmented Generation.” . 2025 https://doi.org/10.48550/arxiv.2502.06864.

[13] Aliyu, I., D., K., and Aliyu, S. “Development of Knowledge Graph for University Courses Management.” International Journal of Education and Management Engineering. 2020 https://doi.org/10.5815/ijeme.2020.02.01.

[14] Li, Z., Cheng, L., Zhang, C., Zhu, X., and Zhao, H. “Multi-Source Education Knowledge Graph Construction and Fusion for College Curricula.” 2023 IEEE International Conference on Advanced Learning Technologies (ICALT). 2023 https://doi.org/10.1109/icalt58122.2023.00111.

[15] Yang, Q., Yuan, P., and Zhu, X. “Research of Personalized Course Recommended Algorithm based on the Hybrid Recommendation.” . 2018; 173. https://doi.org/10.1051/matecconf/201817303067.

[16] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Chen, D., Dai, W., Madotto, A., and Fung, P. “Survey of Hallucination in Natural Language Generation.” ACM Computing Surveys. 2022; 55. https://doi.org/10.1145/3571730.

[17] Barták, R., and Salido, M. “Constraint satisfaction for planning and scheduling problems.” Constraints. 2011; 16. https://doi.org/10.1007/s10601-011-9109-4.

[18] Naveed, Q., Choudhary, H., Ahmad, N., Alqahtani, J., and Qahmash, A. “Mobile Learning in Higher Education: A Systematic Literature Review.” Sustainability. 2023 https://doi.org/10.3390/su151813566.

[19] Chang, Y., Wang, X., Wang, J., Wu, Y., Zhu, K., Chen, H., Yang, L., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P., Yang, Q., and Xie, X. “A Survey on Evaluation of Large Language Models.” ACM Transactions on Intelligent Systems and Technology. 2023; 15. https://doi.org/10.1145/3641289.

[20] Knijnenburg, B., Willemsen, M., Gantner, Z., Soncu, H., and Newell, C. “Explaining the user experience of recommender systems.” User Modeling and User-Adapted Interaction. 2012; 22. https://doi.org/10.1007/s11257-011-9118-4.

[21] Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P., Quinn, K., Teevan, J., Kikin-Gil, R., and Horvitz, E. “Guidelines for Human-AI Interaction.” Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. 2019 https://doi.org/10.1145/3290605.3300233.

[22] Holmes, W., and Tuomi, I. “State of the art and practice in AI in education.” European Journal of Education. 2022 https://doi.org/10.1111/ejed.12533.

[23] Ulriksen, L., Holmegaard, H., and Madsen, L. “Making sense of curriculum—the transition into science and engineering university programmes.” Higher Education. 2017; 73. https://doi.org/10.1007/s10734-016-0099-4.

[24] Ramazanova, V., Sambetbayeva, M., Serikbayeva, S., Sadirmekova, Z., and Yerimbetova, A. “Development of a Knowledge Graph-Based Model for Recommending MOOCs to Supplement University Educational Programs in Line With Employer Requirements.” IEEE Access. 2024; 12. https://doi.org/10.1109/access.2024.3519263.

**Appendix A: Detailed Curriculum Subgraph**

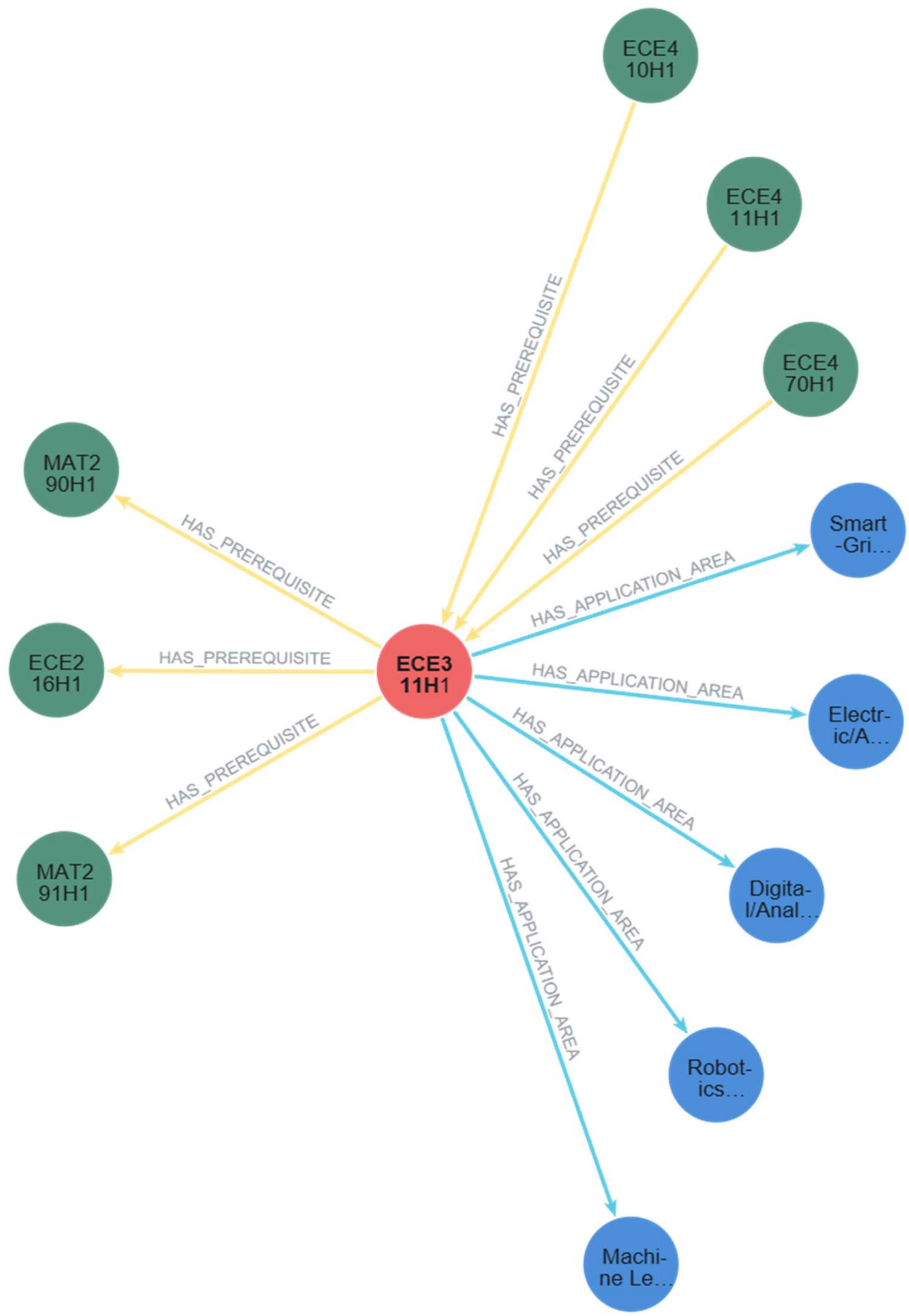


**Figure A1:** A complete Neo4j subgraph centered on ECE311: Introduction to Control Systems. This visualization illustrates the bidirectional network of prerequisite dependencies (HAS_PREREQUISITE), showing both the foundational courses required to take ECE311 and the advanced downstream courses it unlocks, alongside the thematic linkages (HAS_APPLICATION_AREA).