2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #51341

Multi-Agent Orchestration Patterns for Context-Aware Intelligence Systems - Application to Course Recommendation

Sanidhya Karnik, Northeastern University

Data scientist specializing in AI/ML systems and agentic workflows.

Venkata Nithya Ala, Northeastern University
Vignan S Kamarthi, Northeastern University
Trishul Chowdhury, Northeastern University

AI Researcher | Chief AI Scientist | GenAI Leader |

Srinivasan Radhakrishnan, Northeastern University
Prof. Xiaoning Jin, Northeastern University

Prof. Xiaoning (Sarah) Jinâ€™s research focus is in the area of modeling and analysis for intelligent and advanced manufacturing processes and systems, with a specialization in diagnostics and prognostics (D&P), control and predictive decision making.

©American Society for Engineering Education, 2026

Multi-Agent Orchestration Patterns for Context-Aware Intelligence Systems: Application to Course Recommendation

Sanidhya Karnik
karnik.san@northeastern.edu

Venkata Nithya Ala
ala.v@northeastern.edu

Vignan Kamarthi
kamarthi.v@northeastern.edu

Trishul Chowdhury
chowdhury.tri@northeastern.edu

Srinivasan Radhakrishnan
s.radhakrishnan@northeastern.edu

Xiaoning Jin
xi.jin@northeastern.edu

Northeastern University

Abstract

The rapid growth of online and university-level educational offerings has increased decision fatigue for students and professionals seeking to transition into data-intensive fields such as Data Science and Artificial Intelligence. Existing course recommendation systems often rely on static learner profiles or historical interactions, limiting their ability to incorporate contextual information and evolving workforce demands. To address these challenges, this paper introduces a personalized course recommendation system that applies multi-agent orchestration patterns to recommend academic pathways aligned with learners' backgrounds, goals, and industry needs.

The proposed system delivers context-aware academic decision support through a modular, multi-agent AI architecture. Specialized agents integrate course catalog retrieval, peer-based collaborative filtering, resume-driven content analysis, and real-time labor market intelligence. Large language models interpret user intent and coordinate agent workflows, enabling real-time recommendations from heterogeneous data sources. These orchestration patterns are applied end-to-end to the course recommendation task, producing personalized learning pathways aligned with students' academic preparation, career objectives, and current industry demands.

Beyond course recommendation, this work contributes to engineering education by providing a scalable decision-support platform for undergraduate and graduate engineering students. The system improves student outcomes by reducing decision fatigue, supporting informed academic planning, and aligning coursework with learner goals and industry expectations. By incorporating students' academic backgrounds, experiences, and career objectives alongside real-time labor market signals, the system helps engineering students identify pedagogically appropriate, industry-relevant pathways. For students transitioning into data-intensive fields, this approach promotes efficient skill acquisition, improves workforce readiness, and supports sustained engagement as learner goals and market conditions evolve.

Code Availability: The complete implementation is publicly available at https://github.com/vignankamarthi/agent-coordination-framework

1. Introduction

The rapid expansion of online learning platforms and university level course offerings has created unprecedented opportunities for students to acquire new technical skills. However, this abundance has also introduced significant challenges, particularly for undergraduate engineering students and early career professionals seeking to navigate increasingly complex educational pathways. With thousands of courses available across institutions and platforms, learners often struggle to identify which options best align with their academic background, career aspirations, and the evolving demands of industry. This challenge is especially pronounced in rapidly evolving domains such as Data Science and Artificial Intelligence, where skill requirements can quickly erode [1].

Traditional course recommendation systems attempt to address this problem using collaborative filtering or content based approaches driven by historical enrollment data or static user profiles. While effective at scale, these systems often lack contextual awareness of individual learner's prior experience, professional goals, and real-time labor market trends [2]. As a result, recommendations may be generic, outdated, or misaligned with workforce needs, limiting their usefulness for students planning coherent and goal-directed learning trajectories. For engineering education, this disconnect contributes to decision fatigue, inefficient course selection, and a persistent gap between academic preparation and industry expectations.

To address these limitations, this work explores the use of a modular, multi-agent AI architecture for personalized course recommendation. Rather than relying on a monolithic recommendation pipeline, the proposed system decomposes the recommendation task into specialized agents that each focus on a distinct source of evidence, including structured course catalogs, peer-based learner similarity, resume-derived skill profiles, and external labor market intelligence. These agents are coordinated through a large language model driven intent interpretation and workflow orchestration, enabling the system to dynamically combine heterogeneous data sources and generate context-aware recommendations tailored to individual learners. This agentic design supports flexibility, extensibility, and real-time adaptation, which are increasingly important characteristics of modern AI-driven educational systems.

Beyond its functional role as a course recommendation platform, the system is intentionally designed as an educational artifact for undergraduate engineering students. Through the development and use of this system, students gain hands-on experience with key AI engineering concepts, including multi-agent system coordination, heterogeneous data integration, LLM-based decision-making, and scalable system deployment. By embedding these concepts within a real-world educational application, the project bridges theoretical instruction in artificial intelligence with practical system design and implementation.

2. Related Work/Literature Review

The field of course recommendation systems has gained increasing attention in recent years as educational institutions seek to align academic offerings with rapidly evolving labor market demands. Prior studies have highlighted the value of labor market analytics in bridging this gap. For instance, Li et al. [3] conducted a gap analysis of data science and domain knowledge requirements in the manufacturing industry using EMSI job posting data, demonstrating how workforce analytics can inform curriculum development. Such approaches illustrate the potential of linking academic content to industry skill needs, but often remain limited by static datasets, restricted scalability, and the absence of real-time market integration.

Beyond such early explorations, prior research in e-learning recommendation systems has emphasized hybrid approaches that combine collaborative and content-based filtering. Salehi and Kmalabadi (2012) [4] proposed an attribute-based hybrid recommender that integrates learner profiles with course metadata to improve personalization. Similarly, Buncle et al. (2013) [5] explored recommendation chaining for e-learning, where different recommender models are applied sequentially to refine results. Mangina and Kilbride (2008) [6] evaluated key-phrase extraction and tiling for resource recommendation in online learning environments, emphasizing the importance of accurate content analysis in guiding learners to relevant material.

Recent advances in educational recommendation have begun to incorporate multi-agent architectures and large language models to enhance personalization and contextual awareness. Moustamidi and Bouhdidi (2025) [7] proposed a modular multi-agent system integrating RAG (Retrieval-Augmented Generation) and LLMs for secondary education, where specialized agents handle profiling, retrieval, generation, and coordination tasks. Their system combines collaborative and content-based filtering with k-Nearest Neighbors clustering and achieved 95% precision and recall on real educational data. This work demonstrates the potential of agent-based architectures to deliver semantically rich, context-aware recommendations through distributed reasoning and real-time adaptation.

Commercial learning platforms (e.g., large-scale MOOC systems) have also integrated recommendation features, primarily leveraging collaborative filtering, implicit feedback, and engagement histories. While these systems are effective at driving learner engagement, their recommendation logic is typically opaque, and their reliance on user interaction histories can make it difficult to adapt rapidly to emerging labor market trends.

Our work advances these prior efforts by combining early labor market–driven approaches with modern AI engineering pipelines. Specifically, we integrate large language models (LLMs) for semantic query evaluation, vector-based embeddings for semantic similarity search, and knowledge graph driven collaborative filtering for peer-based recommendations. Additionally, by incorporating external market intelligence through a web search agent, our system addresses a critical limitation of earlier solutions: the inability to adapt dynamically to emerging skill demands in today’s job market. This positions our approach as a scalable and intelligent extension of the existing body of literature on AI-powered educational recommendation systems.

3. Methodology

The architecture of the proposed system was chosen to balance personalization, scalability, and adaptability, three qualities often lacking in traditional recommendation systems. A multi-agent design was adopted to ensure modularity, allowing each component (content-based, collaborative, or database-driven) to specialize in its respective task while being orchestrated by a central manager. This structure enables seamless integration of heterogeneous data sources like course catalogs, resumes, academic papers, and labor market signals, while maintaining flexibility for future extensions. By combining LLM-powered intent classification, vector-based semantic retrieval, and graph-based collaborative filtering, the architecture ensures that recommendations remain context-aware, evidence-backed, and responsive to evolving industry needs. Additionally, containerized deployment and structured logging were implemented to create a production-ready system that could seamlessly scale beyond the prototype stage.

3.1 System Architecture and Orchestration

The system is orchestrated using LangGraph, a framework for building stateful, multi-agent workflows with large language models. LangGraph manages workflow state and dynamically routes queries to specialized agents based on user intent. As illustrated in Figure 1, the process begins with intent classification, performed primarily by Cohere's LLM. Queries are mapped to high-level subagents including learn_courses, job_info, and trending_skills. If the LLM output is ambiguous, a robust rule-based fallback ensures intents are categorized based on keyword matching. This design allows the system to direct queries toward the most appropriate recommendation pathway.

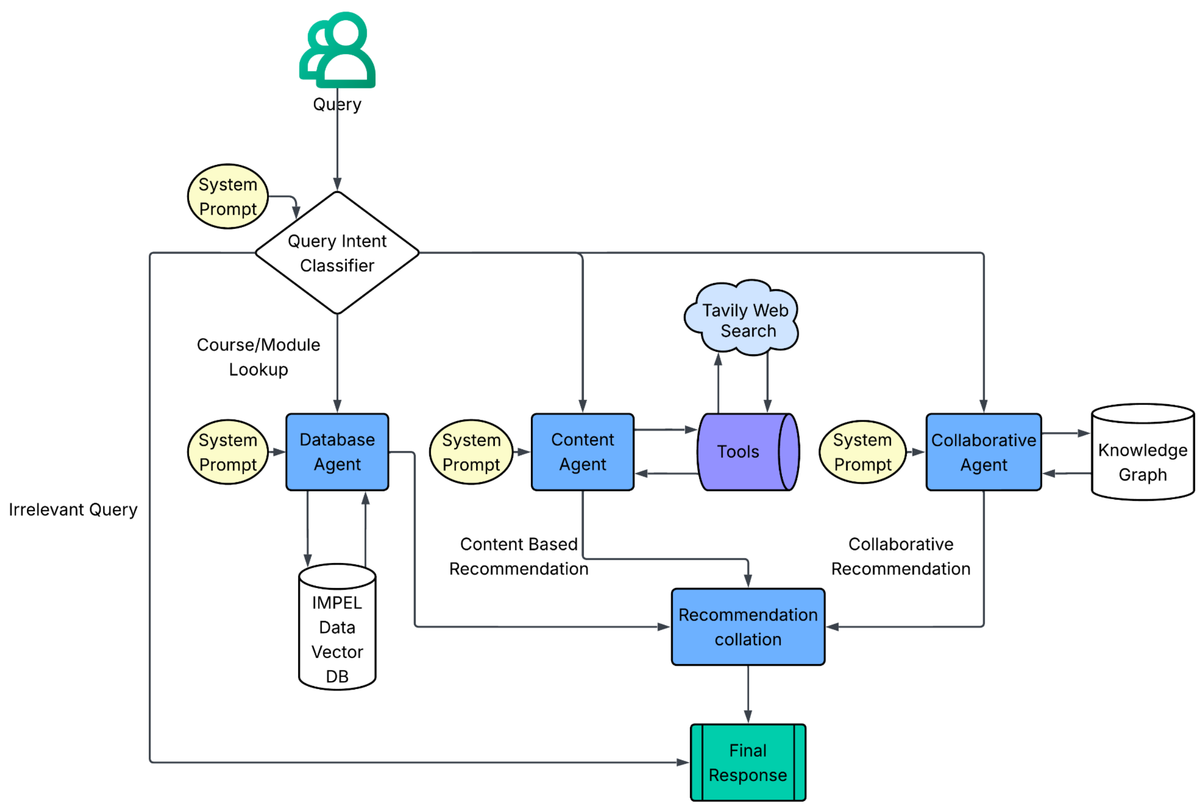

Fig. 1. AI Engineering Pipeline

3.2 Data Sources and Storage

The recommendation system integrates multiple heterogeneous data sources to support hybrid recommendation strategies. The IMPEL (Integrative Manufacturing and Production Engineering Curricula that Leverage Data Science) course catalog (online data science curricula targeting professionals and students interested in advancing their skills and knowledge for advanced manufacturing) is stored with MySQL, containing structured information on courses and modules, including titles, descriptions, and learning outcomes. This relational database provides a reliable foundation for transactional queries and course catalog exploration.

To support semantic retrieval, vector embeddings are generated for both courses and a curated set of 15 peer-reviewed research articles, and indexed using FAISS for efficient similarity search (see References [3], [8]–[21]). These embeddings enable the retrieval of conceptually related course modules or literature when a user specifies a target skill, role, or interest.

In addition to static course and research content, the system captures dynamic user interaction data using Neo4j, a graph database platform. This graph-based representation, visualized in Figure 4, models relationships among users, queries, and recommended courses as an interconnected network. Nodes in the graph correspond to entities such as users, their interactions, and courses enrolled enabling the system to identify communities of learners with similar goals and histories, which is the foundation for collaborative filtering.

Neo4j was chosen over a relational alternative because graph databases naturally represent multi-dimensional relationships and support efficient similarity search across interconnected entities. Unlike relational joins, graph traversals scale effectively with dense interaction data, allowing queries such as “find students with similar course trajectories and retrieve the modules they enrolled in” to execute with low latency. Furthermore, Neo4j’s integration with graph traversal algorithms (e.g., node similarity, community detection, pathfinding) allows the system to identify learner clusters and recommend content based on peer behaviors. This makes the graph database not just a storage layer, but an active information engine that is the backbone of the collaborative filtering agent.

By combining relational storage for static course data, vector stores for semantic similarity, and graph databases for dynamic interaction data, the system ensures both breadth and depth in recommendation generation, enabling highly personalized and context-aware outputs.

3.3 Content-Based Recommendation Engine

The content-based recommendation agent, illustrated in Figure 2, makes use of both user-provided context and external information. User resumes in PDF or DOCX format are parsed using PyMuPDF and docx2txt, producing structured text representations. These are combined with semantic embeddings of courses and research papers to identify content aligned with the user's career goals.

As shown in Figure 2, the agent integrates Tavily to retrieve real-time labor market insights, such as trending skills and job descriptions. The workflow begins with intent classification using an LLM-based classifier that routes queries to appropriate sub-workflows (job_info, trending_skills, or learn_courses). For job-related queries, the system generates section-wise recommendations covering role responsibilities, required skills, and relevant IMPEL courses. Results from Tavily are summarized through role-specific LLM prompts, ensuring that recommendations remain relevant to current industry demands.

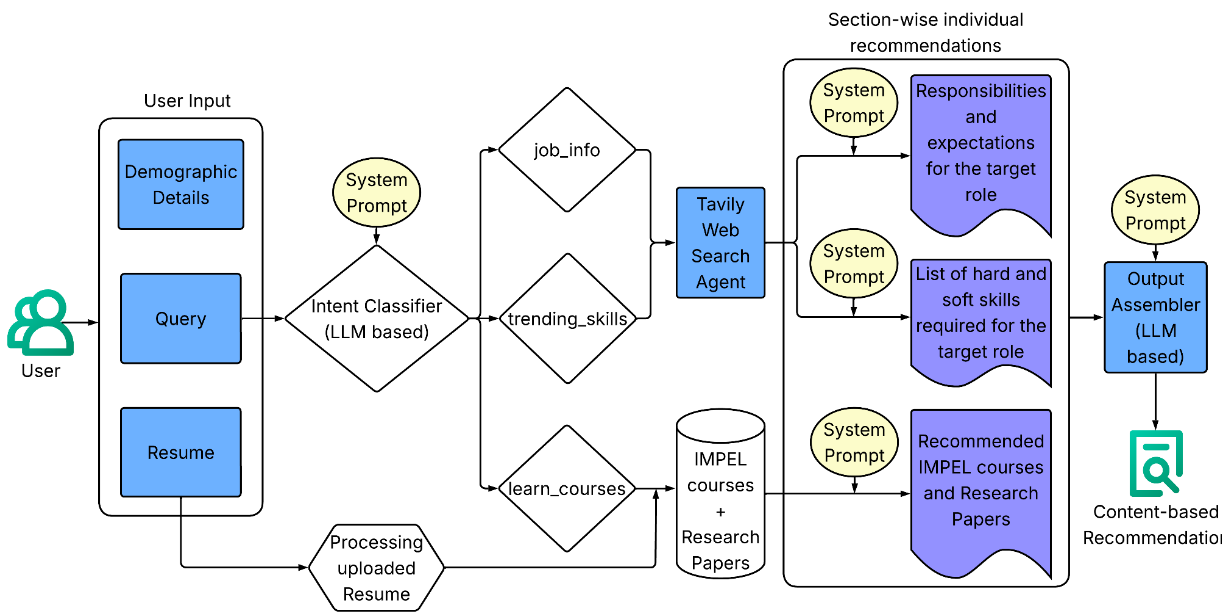

Fig. 2. Content Agent Architecture

3.4 Collaborative Filtering Engine

The collaborative filtering agent, depicted in Figure 3, operates on the Neo4j graph database, where user vectors are constructed from profile and interaction data. As illustrated in the workflow, the system begins with intent classification to determine if collaborative recommendations are appropriate. When triggered, the agent performs similarity searches to identify learners with comparable backgrounds and career objectives through the User Interactions Graph stored in Neo4j.

The LangGraph workflow, shown in Figure 3, orchestrates three main stages: collecting user data to build user vectors, generating recommendations based on similar user pathways, and storing the interaction back into the graph database for future collaborative filtering. Recommendations are generated by drawing on successful course pathways of these similar users, specifically analyzing which IMPEL courses similar users have enrolled in and completed. Cohere's LLM refines the recommendations generated by collaborative insights in the context of the current user's query to improve contextual alignment.

The system outputs two key components: personalized recommendations derived from similar user behaviors and a list of courses that similar users have successfully enrolled in, ensuring that suggestions are grounded in proven learning trajectories.

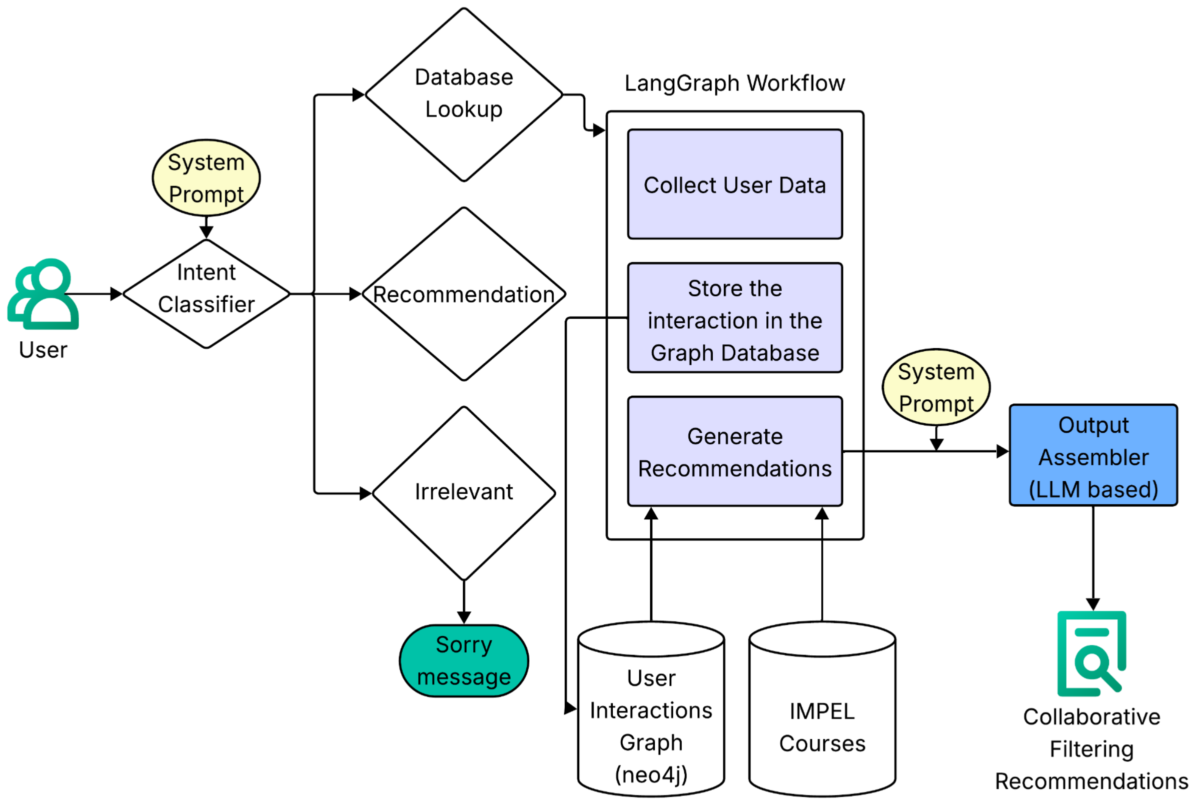

Fig. 3. Collaborative Agent Architecture

3.5 Workflow Execution and Response Generation

Each agent operates independently, but results are aggregated within a unified workflow. Depending on the query type, the system may produce course recommendations, structured learning paths, or market-aligned skill summaries. These outputs are presented through a Gradio-based web interface, which also supports resume uploads.

3.6 Reliability and Deployment

The system is engineered for robustness and scalability. A custom logging framework, SystemLogger, captures detailed runtime events across INFO, DEBUG, and ERROR levels with contextual messages. Structured error handling ensures resilience against common failures such as API timeouts or database connection issues. Deployment is containerized using Docker, providing persistent storage (volumes) for MySQL and Neo4j, and simplified dependency management. API keys and other runtime parameters are configured via a global environment variable file for secure and reproducible deployment.

4. Experimental Setup

4.1 Backend Infrastructure

The system integrates multiple backend components to support personalized course recommendations. A MySQL relational database stores structured information on IMPEL courses and modules, enabling efficient retrieval of course metadata such as summaries. A vector database (FAISS) constructed using embeddings of IMPEL courses and research papers, allows semantic similarity search and evidence-based recommendations grounded in both curriculum content and academic literature.

To model user interactions, a Neo4j graph database is employed, capturing relationships between users, their enrolled courses, and peer similarity networks. Figure 4 illustrates the structure of this graph network, where nodes represent three distinct entity types (users, interactions, and courses) and edges represent the relationships between them. As users interact with the system through the user interface, each interaction is logged in the graph database together with the user's educational and professional background, demographic attributes (e.g., age), query text, and the corresponding system-generated response. This continuous accumulation of interaction data enriches the graph representation (network effect) and enhances the collaborative filtering process by enabling recommendations that draw on evolving patterns of user similarity.

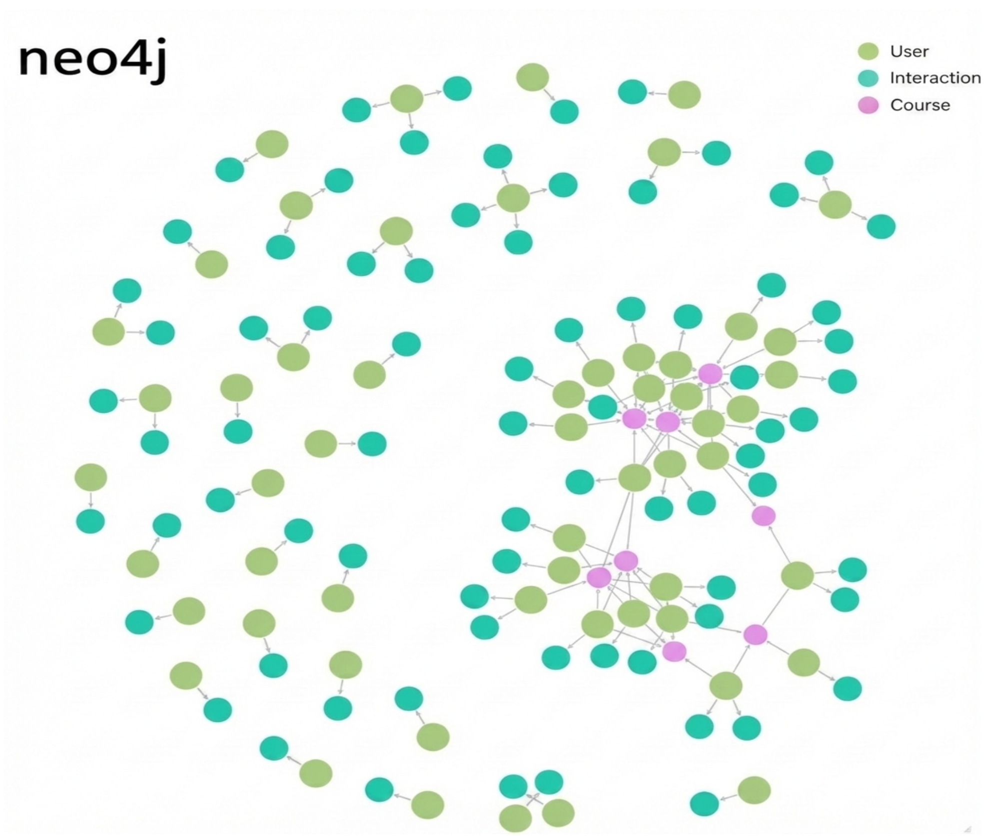

Fig. 4. Neo4j Graph Representation of Users, Interactions, and Courses

4.2 External Tools and Services

The system employs external APIs to enhance recommendation quality with real-time intelligence. In particular, Tavily is integrated to capture up-to-date labor market trends, emerging skill demands, and job-related insights. These signals are incorporated into content-based recommendations, ensuring alignment between course suggestions and current industry requirements.

4.3 Multimodal Input Processing

The platform supports multimodal inputs, allowing users to interact through multiple channels. In addition to natural language queries, users may upload resumes (PDF or DOCX file types)). Uploaded resumes are parsed, vectorized, and integrated into the recommendation workflow to capture individualized skill profiles. Users can also provide demographic details (Age, Education, Professional Status), which serve as additional features for refining recommendation outputs.

4.4 Front-End Interface

The recommendation system exposes its functionality through a Gradio-based web interface, enabling seamless user interaction with the backend services. The interface supports demographic form inputs, free-text queries, and file uploads, while presenting personalized recommendations in real time. By integrating all backend pipelines with a lightweight front-end, the system provides an accessible environment for both development and user evaluation.

4.5 Experimental Protocol

To evaluate system performance, we designed four experimental scenarios reflecting the system's core use cases: (1) IMPEL course recommendation, (2) professional skill related queries based on user profile, (3) job market insights (4) collaborative filtering based on peer similarity. All experiments were conducted in the context of individuals studying or working in the mechanical engineering field who were seeking to learn data science and AI or transition into data-focused professional roles. For each scenario, the system generated course recommendations that were assessed on multiple criteria: relevance (measured through human evaluation and semantic similarity scores), diversity of recommended courses, and system reliability (percentage of queries successfully executed without errors).

5. Results and Discussion

To evaluate the system's capability to correctly classify intents and route queries through the appropriate agent workflow, we conducted tests across four different scenarios: content analysis with trending skills, direct database lookup (IMPEL courses), collaborative recommendation generation, and irrelevant query filtering. Each test demonstrates the orchestrator's decision-making process and the specialized agents' execution patterns as captured through LangSmith tracing.

5.1 Content Analysis for Trending Skills

The content analysis workflow was triggered by a query about trending AI skills, demonstrating the system's ability to integrate real-time market intelligence with course recommendations.

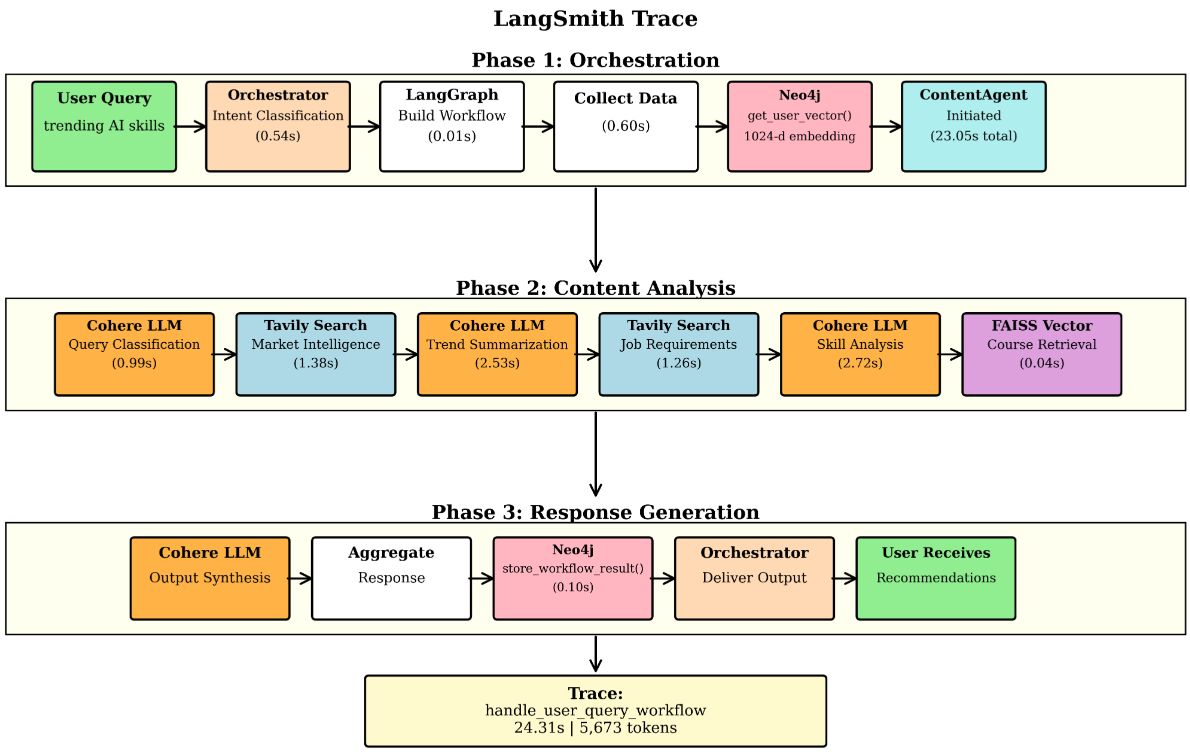

Fig. 5. Langsmith Trace for Process Flow

The LangSmith trace reveals the complete execution pipeline through a three-phase workflow progression as illustrated in Figure 5. Upon receiving a user query, the Orchestrator performs intent classification and routes to LangGraph, which constructs the appropriate workflow. The data collection phase vectorizes user demographics via Neo4j's *get_user_vector()* function, producing a high-dimensional embedding that initializes the ContentAgent. Within the ContentAgent, a multi-stage analysis unfolds: Cohere LLM classifies the query, Tavily web search calls gather market intelligence and job requirement data, and interleaves Cohere LLM invocations that perform trend summarization and skill demand analysis. The FAISS vector store retrieves semantically relevant IMPEL courses and research papers. The agent then synthesizes these heterogeneous data streams into a structured response containing trending skills, job role insights, and personalized course recommendations. Finally, the *store_workflow_result()* operation persists both the response and the user interaction vector to Neo4j, enriching the graph for future collaborative filtering.

5.2 Database Lookup Operations

Direct course catalog queries demonstrate the DatabaseAgent's specialized retrieval capabilities for users seeking specific IMPEL course information. Intent classification accurately routes requests such as "list all IMPEL courses" to the database lookup operation based on explicit course reference patterns. The workflow begins with a data collection phase, followed by the generation of user embeddings within the Neo4j graph database. The DatabaseAgent then performs contextual MySQL queries against the course modules view, filtering and ranking results according to the similarity between the user profile and course prerequisites and learning objectives. Retrieved courses are organized hierarchically and prioritized based on user background, with Data Analytics modules emphasized for analytically oriented users and Computation and Visualization modules weighted more heavily for programming-focused queries. The resulting personalized course ordering reflects user context prior to interaction storage.

5.3 Collaborative Filtering Recommendations

Collaborative filtering demonstrates the system's ability to integrate user similarity analysis with personalized recommendation generation. In this scenario, intent classification identifies a career transition query, such as "become a data scientist," as requiring personalized recommendations. The workflow initializes with user data collection and vectorization, producing a 1024-dimensional embedding derived from educational background, age group, and professional status. The CollaborativeAgent then performs similarity matching against the Neo4j graph to identify users with comparable profiles. Based on the interaction histories and successful learning paths of similar users, the agent queries enrolled courses and completion outcomes. A Cohere language model synthesizes this collaborative intelligence with the IMPEL course catalog to generate targeted recommendations aligned with proven peer trajectories. The system specifically recommends the Data Management for Analytics course sequence, structured NoSQL modules, and complementary visualization coursework, all validated through successful outcomes observed in similar users. The complete interaction is subsequently stored to enhance the collaborative filtering graph.

5.4 Irrelevant Query Filtering

The system effectively identifies and handles out-of-scope queries, preventing inappropriate routing and preserving recommendation focus. In this case, the orchestrator applies early termination by analyzing a weather-related query during intent classification. The query is correctly identified as falling outside the educational domain, and the system responds with a polite redirection message. No database connections, vector operations, or external API calls are invoked. This efficient rejection mechanism conserves system resources and reinforces clear architectural boundaries, highlighting robust intent classification as a gateway control mechanism.

5.5 System Performance Observations

Across all evaluated scenarios, the system exhibits consistent architectural behavior characterized by accurate intent classification, appropriate agent selection, and graceful handling of diverse query types. LangSmith traces confirm that each specialized agent operates within its designated scope, with the DatabaseAgent supporting structured retrieval, the ContentAgent integrating market intelligence, and the CollaborativeAgent enabling peer-based recommendations. Orchestrator state management through LangGraph ensures reliable data flow across workflow nodes, while comprehensive error handling mechanisms prevent cascading failures. Collectively, these results validate the effectiveness of the proposed multi-agent architecture in delivering contextually appropriate educational recommendations while maintaining operational efficiency through intelligent query routing.

6. Conclusion

This research demonstrates the feasibility and effectiveness of an AI-powered, multi-agent course recommendation system that bridges academic offerings with evolving industry skill requirements. Addressing the persistent misalignment between learners' course selections and rapidly changing workforce demands, the proposed system integrates LLM-based natural language understanding, content-based retrieval, collaborative filtering, and external labor market intelligence to overcome key limitations of traditional recommendation systems, particularly limited personalization and weak industry alignment.

Results show that the system generates goal-aligned, context-aware recommendations that reduce decision fatigue while providing structured educational pathways. The use of resumes and demographic inputs enables deeper personalization beyond static user profiles, while real-time market signals ensure alignment with current skill trends. The modular multi-agent architecture, implemented using LangGraph and containerized deployment, demonstrates extensibility, robustness, and practical applicability for real-world educational advising in the Data Science and AI domains.

The key contributions of this work include: (1) Context-Aware Personalization, achieved through resume and demographic inputs to tailor academic pathways; (2) Industry Relevance, enabled by leveraging external labor market signals to align recommendations with job demand; and (3) Scalable Architecture, realized through a modular, agent-based design supporting extensibility, robustness, and real-world deployment. Together, these contributions validate that multi-agent orchestration combined with industry-aware intelligence can significantly enhance course recommendation quality and establish a strong foundation for intelligent, adaptable educational guidance systems.

6.1 Future Research

Building on the validated architecture and empirical findings of this study, focused extensions can further enhance system performance and applicability. Systematic prompt optimization across sub-agents is enabled by LangSmith's tracing and evaluation capabilities. Structured A/B testing of prompt variations for intent classification, content generation, and recommendation synthesis will enable data-driven improvements grounded in observed system behavior.

Interaction traces and successful workflows captured through LangSmith enable domain adaptation of language models for educational recommendation tasks. Curated traces, user interaction patterns, and agent decision paths support parameter-efficient fine-tuning using methods such as LoRA and QLoRA, improving domain awareness and alignment with educational pathways while maintaining computational efficiency. Trace analysis guides responsiveness optimizations in inter-agent communication without altering the core system design.

The system may also be extended through integration with the Model Context Protocol (MCP), enabling external AI assistants to access recommendation workflows or individual agent capabilities. This integration provides a practical pathway for broader adoption while preserving the modular and extensible design principles demonstrated in this work.

Acknowledgements

The authors gratefully acknowledge the support of the IMPEL (Integrative Manufacturing and Production Engineering Curricula that Leverage Data Science) program for providing access to the course catalog and educational materials that served as the foundation for this recommendation system.

Special thanks to the open-source community for maintaining the essential libraries and frameworks that enabled this work, including LangGraph, LangSmith, Neo4j, FAISS, and Gradio. The availability of these tools was critical to the successful implementation of our modular, scalable architecture.

Finally, we acknowledge the contributions of peer reviewers and colleagues whose constructive feedback strengthened both the technical implementation and the presentation of this research.

References

[1] S. A. Undare, K. Karami, and M. S. Trimboli, "AI-Driven Course Recommendation System for Enhanced Career Alignment in Engineering Education," in *Proc. ASEE Annual Conference & Exposition, Montreal, QC, Canada*, Jun. 2025, Paper no. 57586, doi: 10.18260/1-2-57586.

[2] J. Frej, A. Dai, S. Montariol, A. Bosselut, and T. Käser, "Course recommender systems need to consider the job market," arXiv:2404.10876v1 [cs.IR], 2024. [Online]. Available: https://arxiv.org/pdf/2404.10876.

[3] G. Li, C. Yuan, S. Kamarthi, M. Moghaddam, and X. Jin, "Data science skills and domain knowledge requirements in the manufacturing industry: A gap analysis," *Journal of Manufacturing Systems*, vol. 60, pp. 692-706, Jul. 2021, doi: 10.1016/j.jmsy.2021.07.007.

[4] M. Salehi and I. N. Kmalabadi, “A hybrid attribute–based recommender system for e–learning material recommendation,” IERI Procedia, vol. 2, pp. 565–570, 2012, doi: 10.1016/j.ieri.2012.06.135.

[5] J. Buncle, R. Anane, and M. Nakayama, "A Recommendation Cascade for e-learning," in Proc. 27th IEEE Int. Conf. Advanced Information Networking and Applications (AINA), Barcelona, Spain, Mar. 2013, pp. 740-747, doi: 10.1109/AINA.2013.142.

[6] E. Mangina and J. Kilbride, "Evaluation of key phrase extraction algorithm and tiling process for a document/resource recommender within e-learning environments", Computers & Education, vol. 50, no. 3, pp. 807-820, 2008. Available: 10.1016/j.compedu.2006.08.012.

[7] A. E. Moustamidi and J. E. Bouhdidi, “A modular multi-agent architecture for hybrid educational recommendation systems integrating RAG and LLMs,” *Journal of Information Systems Engineering and Management*, vol. 10, no. 58s, pp. 485-495, 2025.

[8] A. Verma, K. M. Yurov, P. L. Lane, and Y. V. Yurova, “An investigation of skill requirements for business and data analytics positions: A content analysis of job advertisements,” *Journal of Education for Business*, vol. 94, no. 4, pp. 243-250, 2019, doi: 10.1080/08832323.2018.1520685.

[9] E. Ras, F. Wild, C. Stahl, and A. Baudet, “Bridging the skills gap of workers in Industry 4.0 by human performance augmentation tools: Challenges and roadmap,” in *Proc. 10th Int. Conf. Pervasive Technol. Relat. Assist. Environ. (PETRA ’17)*, Rhodes, Greece, Jun. 2017, pp. 428-432, doi: 10.1145/3056540.3076192.

[10] C. Javdekar, E. Watson, V. Kapilow, M. Bograd, P. Boyer, I. Zeid, and C. Duggan, “Closing the advanced manufacturing talent gap,” *Procedia Manufacturing*, vol. 5, pp. 1197-1207, 2016, doi: 10.1016/j.promfg.2016.08.094.

[11] M. Lenard, “Contemporary labor market information in postsecondary contexts,” Society for Research on Educational Effectiveness (SREE), Washington, DC, USA, Mar. 2021.

[12] A. Skevi, H. Szigeti, S. Perini, M. Oliveira, M. Taisch, and D. Kiritsis, “Current skills gap in manufacturing: Towards a New skills Framework for Factories of the future,” in *Lecture Notes in Computer Science*, 2014, pp. 175-183. doi: 10.1007/978-3-662-44739-0_22.

[13] F. Tao, Q. Qi, A. Liu, and A. Kusiak, “Data-driven smart manufacturing,” *Journal of Manufacturing Systems*, vol. 48, Part C, pp. 157–169, Jul. 2018, doi: 0.1016/j.jmsy.2018.01.006.

[14] F. Yang and S. Gu, "Industry 4.0, a revolution that requires technology and national strategies," Complex Intell. Syst., vol. 7, pp. 1311-1325, 2021, doi: 0.1007/s40747-020-00267-9.

[15] S. Vaidya, P. Ambad, and S. Bhosle, "Industry 4.0–a glimpse," Procedia Manufacturing, vol. 20, pp. 233-238, 2018.

[16] P. Mikalef and J. Krogstie, "Investigating the Data Science Skill Gap: An Empirical Analysis," in Proc. IEEE Global Engineering Education Conference (EDUCON), Dubai, UAE, 2019, pp. 1273-1279.

[17] D. Chenoy, S. M. Ghosh, and S. K. Shukla, "Skill development for accelerating the manufacturing sector: the role of 'new-age' skills for 'Make in India,'" *International Journal of Training Research*, vol. 17, no. sup1, pp. 112-130, Jul. 2019, doi: 10.1080/14480220.2019.1639294.

[18] A. Weaver and P. Osterman, "Skill demands and mismatch in U.S. manufacturing," ILR Review, vol. 70, no. 2, pp. 275-307, Mar. 2017, doi: 10.1177/0019793916660067.

[19] J. Kimmell and S. A. Martin, "Sorting out the Skills Gap: Analyzing the Evidence for a Shortage of Middle-Skill Workers in the Manufacturing and Healthcare Industries in the Portland Region," 2015. [Online]. Available: https://pdxscholar.library.pdx.edu/metropolitianstudies/123.

[20] L. Moldovan, "State-of-the-art analysis on the knowledge and skills gaps on the topic of industry 4.0 and the requirements for work-based learning," *Procedia Manufacturing*, vol. 32, pp. 294-301, Jan. 2019, doi: 10.1016/j.promfg.2019.02.217.

[21] A. Vista, "Data-driven identification of skills for the future: 21st-century skills for the 21st-century workforce," *SAGE Open*, vol. 10, no. 2, Apr. 2020, doi: 10.1177/2158244020915904.