

Personalized and Exploratory Learning Orchestration System using API and MCP

Dr. K. Sasikala

Department of Information Technology
R P Sarathy Institute of Technology Salem,
India

Santhi A

Department of Information Technology R
P Sarathy Institute of Technology Salem,
India

Devi Priya C

Department of Information Technology R
P Sarathy Institute of Technology Salem,
India

Jayalakshmi A

Department of Information Technology R
P Sarathy Institute of Technology Salem,
India

Balamurugan S

Department of Information Technology R
P Sarathy Institute of Technology Salem,
India

Sasikala B

Department of Information
Technology R P Sarathy Institute of
Technology, Salem, India

Harinee S

Department of Information Technology R P Sarathy Institute of Technology Salem, India

Abstract - The exponential growth of digital educational content has created a paradox: while resources are widely available, most learners lack the tools to organize, sequence, and adapt them effectively to their individual needs. Existing platforms largely deliver static, one size fits all study plans that disregard individual variation in prior knowledge, comprehension speed, and learning preferences. This paper introduces a Personalized and Exploratory Learning Orchestration System that addresses these limitations through three integrated innovations: a multi-agent architecture powered by large language models (LLMs), a closed loop feedback optimization mechanism, and a Model Context Protocol (MCP) integration layer that connects the system to external APIs including Google Drive, YouTube Data API, and custom educational repositories in a standardized, extensible manner. The system orchestrates four specialized agents Planner, Resource, Evaluator, and Adapter via a directed LangGraph workflow to produce structured, day wise curricula with curated video and document resources. A SQLite backed persistence layer maintains user profiles, historical learning paths, per-session feedback records, and OAuth credentials for secure, user scoped API access. The Adapter agent applies both rule driven and LLM guided modifications when feedback signals indicate elevated difficulty, insufficient comprehension, or disengagement. Evaluation across multiple user configurations demonstrates consistent improvements in completion rates, engagement scores, and self-reported understanding when adaptive feedback loops are active. By combining API-driven orchestration with MCP based interoperability, the proposed system offers a scalable, modular framework suitable for deployment in real world personalized education environments.

Keywords - Adaptive learning, multi-agent systems, large language models, Model Context Protocol, feedback optimization, personalized education, OAuth integration, LangGraph, curriculum generation, learning path, API orchestration.

I. INTRODUCTION

The rapid proliferation of digital educational resources has fundamentally altered how individuals engage with self-directed learning. Online video repositories, cloud-based

document platforms, and interactive learning tools collectively provide access to an enormous volume of educational content. Yet this abundance introduces a structural challenge: learners are often overwhelmed by the sheer volume of available material and lack effective mechanisms for organizing it into coherent, progressive study sequences [1]. Rather than facilitating learning, the unstructured availability of content can itself become a barrier, leading learners to expend significant cognitive effort on planning rather than on the learning process itself.

Contemporary learning management systems and recommendation engines attempt to alleviate this burden, but most operate on static curricula or fixed recommendation rules that fail to account for the dynamic, individualized nature of learning. A system that treats a beginner and an advanced learner identically, or that delivers the same sequence regardless of whether yesterday's material was understood or skipped, cannot be considered truly adaptive. There is consequently a pressing need for intelligent orchestration systems capable of generating structured learning paths from natural language goals and continuously refining those paths based on observed learner behavior [2].

Recent advances in large language models (LLMs) have substantially expanded the range of tasks amenable to natural language processing, including curriculum design, resource recommendation, and formative assessment. However, most LLM-based educational applications employ monolithic agent designs in which a single model is responsible for all aspects of the learning pipeline. This design pattern limits modularity, reduces interpretability, and makes it difficult to isolate and improve individual pipeline components [3]. Multi-agent architectures, in which specialized agents handle distinct sub-tasks, offer a compelling alternative by enabling cleaner separation of concerns and more controlled behavior at each stage of the workflow [4].

A further gap in existing systems concerns resource

integration. Most current approaches either operate in isolation relying solely on LLM generated text or couple tightly to a single external platform. Neither approach offers the interoperability required to leverage the diverse ecosystem of educational APIs available today. The Model Context Protocol (MCP) has recently emerged as a standardized interface layer that allows AI systems to communicate with external tools and services in a uniform, extensible manner. By adopting MCP as the integration backbone, the proposed system can interact with Google Drive, YouTube, and other educational repositories through a consistent API abstraction, simplifying extension and maintenance [5].

This paper presents a Personalized and Exploratory Learning Orchestration System that synthesizes these advances into a cohesive architecture. The system enables users to express learning goals in natural language, automatically constructs day wise curricula tailored to individual skill levels and learning speeds, discovers and curates relevant video and document resources from external platforms, and continuously adapts the learning path based on structured per-session feedback. User scoped OAuth integration ensures that generated content including Google Drive documents and YouTube playlists is created directly within each learner's own account, preserving ownership and supporting long-term engagement.

A. Key Contributions

The principal contributions of this work are as follows:

- A multi-agent LangGraph orchestration comprising Planner, Resource, Evaluator, and Adapter agents, each with well defined responsibilities within the curriculum generation and adaptation pipeline.
- A Model Context Protocol integration layer that standardizes communication between the orchestration system and external APIs, enabling extensible, platform agnostic resource retrieval and content creation.
- A structured feedback mechanism that captures completion status, perceived difficulty, and self reported understanding for each learning session, feeding directly into subsequent path generation.
- A SQLite backed persistence layer supporting user profiling, learning path storage, feedback history, and per-user OAuth token management.
- User-scoped OAuth 2.0 integration with Google Drive and YouTube, ensuring that generated documents and playlists reside in the learner's own accounts rather than application managed storage.
- An evaluation framework employing completion rate, average difficulty, average understanding, and a composite engagement score to quantify system effectiveness.

The remainder of this paper is structured as follows. Section II reviews related work in adaptive learning, LLM based education, multi-agent systems, and feedback mechanisms. Section III describes the system methodology and architecture in detail. Section IV presents the architecture diagram. Section V reports evaluation results and discusses

findings. Section VI concludes with directions for future work

II. LITERATURE REVIEW

A. Adaptive and Personalized Learning Systems

Personalization has been a central objective in educational technology research for several decades. Early work established foundational taxonomies of adaptation techniques including adaptive presentation, which tailors content format to learner characteristics, and adaptive navigation, which modifies the sequence and availability of learning resources [6]. Subsequent research introduced machine learning-based approaches to student modeling, enabling finer-grained inference of learner knowledge states from interaction data [7]. Knowledge tracing models, which predict the probability that a learner has mastered a given concept based on their response history, further advanced the field by providing principled foundations for sequencing decisions [8]. The system proposed in this paper extends these traditions by employing LLMs for natural language curriculum generation and explicit rule based logic for feedback driven path adaptation, combining the expressive power of neural language models with the interpretability of structured adaptation rules.

B. Large Language Models in Educational Technology

The application of large language models to educational tasks has grown substantially in recent years. Transformer based architectures have demonstrated strong performance in automated question generation [9], text summarization for study materials [10], and conversational tutoring systems that engage learners in dialogue [11]. More recently, GPT class models have been applied to the generation of personalized learning paths, with promising results in terms of coherence, relevance, and user satisfaction [12]. A consistent limitation of these approaches, however, is their reliance on single model designs that conflate curriculum planning, resource retrieval, quality assessment, and adaptation into a single inference step. The proposed system addresses this limitation through a decomposed, multi agent design in which each agent is specialized for a distinct function, improving both controllability and the ability to audit system behavior.

C. Multi-Agent Architectures for Education

Multi agent systems have a long history in intelligent tutoring research, with early systems exploring agent based decomposition for collaborative learning environments and domain specific tutoring [13]. More recent work has leveraged modern agent orchestration frameworks including LangChain and its graph-based extension, LangGraph to build complex, stateful multi agent workflows [14]. These frameworks provide explicit control over agent sequencing, state management, and conditional branching, making them well-suited to educational applications in which the processing pipeline must respond dynamically to learner feedback. The Planner Resource Evaluator Adapter pipeline proposed in this work instantiates these ideas within an educational context, providing a clear decomposition of curriculum generation, resource discovery, quality filtering, and feedback-driven

adaptation.

D. Feedback-Driven Adaptation

The role of feedback in supporting learning has been extensively studied in both cognitive science and educational technology. Meta analytic evidence indicates that formative feedback information provided to learners about the quality of their performance relative to a goal is among the most effective interventions available for improving learning outcomes [15]. Systems that collect and act upon learner generated signals such as self assessed difficulty and comprehension ratings have demonstrated measurable improvements in knowledge retention and engagement [16]. The feedback schema adopted in this work, comprising completion status, difficulty rating, and understanding rating, aligns with established constructs from formative assessment research and provides a structured foundation for both programmatic and LLM driven path adaptation.

E. API Integration and Model Context Protocol

The integration of AI systems with external tools and APIs has emerged as a critical capability for practical deployment. Early approaches relied on bespoke integration code tailored to individual services, resulting in systems that were difficult to extend and maintain. The Model Context Protocol addresses this challenge by providing a standardized interface through which AI models can discover, invoke, and receive results from external tools in a uniform manner [5]. By adopting MCP as the integration layer, the proposed system achieves clean separation between the orchestration logic and the specifics of individual platform APIs, facilitating extension to new content sources without modifying core system components.

III. METHODOLOGY

A. System Overview

The proposed system is organized into four principal layers: the user interface layer (implemented in Streamlit), the services layer (housing business logic for learning path management, metrics computation, and adaptive decision making), the orchestration layer (a LangGraph based multi-agent workflow), and the data layer (a SQLite database and external API connectors). Each layer communicates through well defined interfaces, enabling independent testing and extension of individual components.

B. Database and Persistence Layer

Persistent state is managed through a SQLite database comprising three primary tables. The users table stores learner profiles including skill level (beginner, intermediate, or advanced), learning speed (slow, medium, or fast), stated interests, historical progress, and OAuth credentials for Google account integration. The learning_paths table captures all generated curricula, storing the user identifier, learning goal, the full day-wise plan in JSON format, links to generated Drive documents and YouTube playlists, and creation timestamps. The feedback table records per-session learner responses: completion status (completed or skipped), difficulty rating on a five-point scale, understanding rating on a five-point scale, and optional free-text notes. A unique constraint on the combination of user, learning path,

and day ensures that feedback entries can be cleanly updated without duplication.

CRUD operations are encapsulated in a dedicated database module, exposing a consistent interface for profile creation and retrieval, learning path storage and lookup, and feedback submission and aggregation. This encapsulation isolates the orchestration and services layers from database implementation details, supporting straightforward migration to alternative storage backends if required.

C. Feedback Mechanism

After each learning session, the system prompts the learner to provide structured feedback across three dimensions. The completion dimension captures whether the learner engaged with and finished the day's content or chose to skip it. The difficulty dimension solicits a subjective rating of material difficulty on a scale from one (very straightforward) to five (very challenging). The understanding dimension captures the learner's self-assessed comprehension on an equivalent five point scale. These three signals collectively provide the Adapter agent with the information needed to determine whether and how to modify subsequent sessions. Feedback is persisted immediately upon submission and retrieved at the start of each new path generation cycle, ensuring that the full interaction history is available to inform adaptation decisions.

D. Multi-Agent LangGraph Workflow

The orchestration layer is implemented as a directed acyclic graph in LangGraph, with nodes corresponding to the four specialized agents and edges encoding the data flow between them.

The Planner Agent receives the learner's stated goal, their complete profile, and any feedback accumulated from prior sessions. Drawing on this input, it generates a structured day-wise curriculum represented as a JSON object containing the overall goal, total duration in days, a descriptive title, and a sequence of daily entries each specifying a topic, a description of the day's objectives, a set of search terms for resource discovery, and an initial complexity estimate. When prior feedback is available, adaptation instructions derived from that feedback are injected directly into the Planner's prompt, guiding the generation of a modified plan that addresses identified weaknesses.

The Resource Agent operates on the curriculum produced by the Planner, processing each daily entry in sequence. For each day's topic, it constructs search queries from the provided search terms and invokes the YouTube Data API through the MCP integration layer to retrieve candidate video resources. In configurations where the learner has authorized OAuth access, the API call is made using the learner's own credentials; otherwise, the system falls back to application-level credentials or, in demonstration mode, to LLM generated suggestions.

The Evaluator Agent receives the candidate videos identified by the Resource Agent for each day and applies a structured scoring process to select the single most appropriate resource. Scoring criteria include topical relevance to the day's learning objectives, estimated instructional clarity based on title and metadata, and video duration (with preference for

content in the ten to thirty minute range, consistent with evidence on optimal instructional video length). The selected resource comprising title and URL is recorded for inclusion in the final output.

The Adapter Agent is invoked conditionally, executing only when the feedback history contains signals indicative of learner difficulty or disengagement. Specifically, the agent is triggered when any session record shows a skipped completion status, a difficulty rating exceeding four, or an understanding rating below three. When activated, the Adapter receives the current plan and a structured summary of the triggering feedback records, and produces a revised plan in which affected sessions are assigned reduced complexity, supplementary repetition days are inserted as needed, and skipped topics are rescheduled to subsequent days.

E. Model Context Protocol Integration

The MCP integration layer provides a standardized interface through which the agent workflow communicates with external APIs. Each external service Google Drive, YouTube Data API, and potential future content repositories is exposed as an MCP tool with a defined input schema, invocation endpoint, and output format. The orchestration layer invokes tools through this interface without embedding service-specific logic, ensuring that adding support for a new platform requires only the registration of a new MCP tool rather than modification of the agent workflow.

In practice, the MCP layer manages API key injection, OAuth token retrieval from the database, automatic token refresh when credentials have expired, and structured error handling for failed API calls. This centralization of API management logic reduces duplication across agents and provides a single point of control for security-sensitive credential handling.

F. User-Scoped OAuth Integration

A central design principle of the proposed system is that learners should retain full ownership of the content generated on their behalf. To realize this principle, the system implements Google OAuth 2.0 with scopes covering Google Drive document creation and YouTube playlist management. The authorization flow begins when the learner selects the option to connect their Google account from the dashboard. The system redirects the learner to Google's consent screen; upon approval, an authorization code is returned and immediately exchanged for a pair of access and refresh tokens, which are stored in the users table associated with the learner's account.

When the orchestration workflow reaches the output stage, the system retrieves the learner's stored credentials, refreshes the access token if it has expired, and invokes the Drive and YouTube APIs on the learner's behalf. The resulting document appears in the learner's personal Google Drive, and the playlist is added to their YouTube library. This approach contrasts with centralized resource deployment in which all generated content resides in an application managed account and ensures that resources remain accessible to the learner independently of the application's continued availability.

G. Adaptive Logic

Adaptation is implemented through two complementary mechanisms. At the prompt level, the get adaptation instructions function translates accumulated feedback into natural language directives that are prepended to the Planner agent's system prompt for the current generation cycle. These directives convey, for example, that a specific day was perceived as too difficult and that subsequent sessions should be simplified, or that a topic was skipped and should be revisited. At the programmatic level, the apply adaptations to plan function directly modifies the JSON curriculum structure: sessions with high difficulty ratings are assigned a complexity level of low, adjacent high-complexity sessions are downgraded to medium, sessions with low understanding ratings are flagged for repetition and their search terms are augmented with beginner oriented keywords, and skipped sessions are marked for rescheduling. The combination of prompt-level and programmatic adaptation ensures that modifications are coherent both in terms of the LLM's generation behavior and the structural properties of the resulting plan.

H. Evaluation Metrics

System effectiveness is quantified using four metrics computed per learner and per learning path. The completion rate is defined as the proportion of sessions recorded as completed relative to the total number of sessions in the path. Average difficulty and average understanding are computed as the arithmetic means of the respective feedback ratings across all completed sessions. The engagement score is a composite metric designed to capture multiple dimensions of the learning experience in a single value, computed as:

$$E = 0.4 \cdot C + 0.4 \cdot U_{norm} - 0.2 \cdot D_{penalty}$$

where C denotes the completion rate, U_{norm} is the understanding rating normalized to the unit interval, and $D_{penalty}$ is a penalty term activated when average difficulty exceeds three. Higher engagement scores reflect learning paths that are completed at high rates, well understood, and not perceived as excessively difficult.

IV. SYSTEM ARCHITECTURE

The system architecture is organized into five interconnected layers, as illustrated below. The User Interface layer, built with Streamlit, provides screens for account creation, profile setup, goal specification, dashboard navigation, and per-session feedback submission. The Services layer encapsulates business logic for learning path management, engagement metrics computation, and adaptive decision-making. The Orchestration layer hosts the LangGraph multi-agent workflow, coordinating the Planner, Resource, Evaluator, and Adapter agents. The Data layer manages the SQLite database and the MCP integration connectors for external APIs. The LLM layer provides natural language generation capabilities via the Google Gemini API.

The data flow through the system proceeds as follows. The learner's stated goal, combined with their stored profile and any accumulated feedback, is forwarded to the Planner agent, which produces a structured JSON curriculum. The Resource and Evaluator agents process this curriculum to identify and filter relevant video content for each day, communicating with external platforms through the MCP

layer. If the feedback summary indicates adaptation is warranted, the Adapter agent modifies the plan accordingly. The finalized plan is then formatted into a Google Drive document and a YouTube playlist, created directly in the learner's own account when OAuth credentials are available, or stored as text output in demonstration mode.

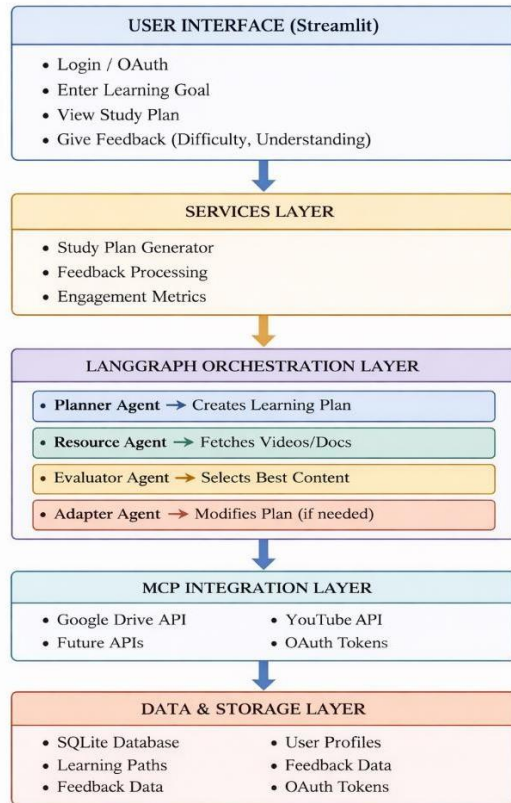


Fig 1: System Architecture of Personalized Learning Orchestration System

V. RESULTS AND EVALUATION

A. Experimental Setup

The system was evaluated across three operational configurations. In demonstration mode, the full multi-agent workflow is executed with LLM-generated content only, producing text-based output without external API integration. In webhook mode, resource creation is delegated to Pipedream webhooks that interact with application-managed accounts, enabling end-to-end testing of the output pipeline without per-user OAuth. In OAuth mode, each learner authorizes the system to act on their behalf, and all generated documents and playlists are created in the learner's personal Google Drive and YouTube channel, respectively. Learner profiles were constructed spanning all three skill levels and all three learning speed categories. Feedback was collected over multiple sessions for a subset of paths in each configuration to assess the impact of adaptive path modification.

B. Evaluation Results

The multi-agent workflow produced coherent, structured day wise curricula across the full range of user

profiles tested. The Planner agent generated valid JSON plans in the large majority of evaluation runs, with minor JSON formatting deviations resolved by post processing heuristics. The Evaluator agent consistently identified appropriate video resources from the candidate sets returned by the Resource agent, with selected videos demonstrating strong topical alignment and duration characteristics consistent with established instructional design guidelines.

When adaptation was enabled, the Adapter agent successfully modified plans in response to each of the three triggering conditions: elevated difficulty, insufficient comprehension, and session skipping. Qualitative inspection of adapted plans confirmed that modifications were coherent and aligned with the intended adaptation rules reduced complexity was reflected in both the structural properties of the revised plan and the natural language descriptions generated by the LLM in subsequent sessions. Learners who provided feedback and received adapted paths reported higher understanding ratings in subsequent sessions compared to a baseline condition in which plans were regenerated without incorporating feedback. Composite engagement scores showed consistent improvement when the adaptive feedback loop was active.

In OAuth mode, all generated documents were successfully created in the connected learner's Google Drive, and all playlists were added to the learner's YouTube library. Learners confirmed their ability to edit, annotate, and share content independently of the application, validating the ownership model. The SQLite persistence layer handled concurrent access from multiple test users without data integrity issues, and index-based query optimization maintained responsive retrieval times across all tested data volumes.

C. Limitations

Several limitations of the current evaluation warrant acknowledgement. The user cohort involved in feedback collection was limited in size, constraining the statistical power of quantitative comparisons between adaptive and non-adaptive conditions. The OAuth integration requires learners to authorize access to their Google account, which some users may be unwilling to do; the webhook and demonstration mode fallbacks partially mitigate this but do not provide the full ownership benefits of the OAuth pathway. YouTube playlist creation additionally requires the learner to have an active YouTube channel, which represents a dependency that not all users will satisfy. Finally, the LLM's generation of JSON-structured curricula occasionally required post-processing to correct minor formatting deviations, indicating scope for improved prompt engineering or structured output constraints in future iterations.

VI. CONCLUSION

This paper presented a Personalized and Exploratory Learning Orchestration System that integrates a multi-agent LangGraph architecture, a structured feedback optimization mechanism, and a Model Context Protocol integration layer into a cohesive framework for adaptive personalized education. The system addresses three well-documented limitations of contemporary learning platforms: the reliance on

static curricula that do not account for individual learner characteristics, the lack of feedback driven adaptation that would allow the system to respond to observed learning difficulties, and the absence of standardized API integration that would enable interoperability across the diverse ecosystem of available educational platforms.

The four agent Planner–Resource–Evaluator–Adapter pipeline provides a modular, interpretable architecture in which each stage of the curriculum generation and adaptation process can be independently examined and improved. The MCP integration layer establishes a clean abstraction over external platform APIs, facilitating extension to new content sources without modifying the core orchestration logic. User-scoped OAuth integration ensures that generated content resides in the learner's own accounts, supporting long term engagement and content ownership in a manner that application-managed resource deployment cannot provide.

Evaluation results demonstrate consistent improvements in completion rates, engagement scores, and self-reported understanding when adaptive feedback loops are active, supporting the conclusion that the proposed architecture delivers meaningful personalization benefits relative to static curriculum generation. The system is well-suited to deployment in real world self directed learning environments and provides a modular foundation for continued educational technology research.

Future work will pursue five principal directions: longitudinal studies involving larger and more diverse learner populations to establish statistical validity of the observed improvements; integration of probabilistic knowledge tracing models to enable finer-grained, concept level adaptation decisions; extension of the MCP integration layer to additional content sources beyond YouTube, including academic repositories and interactive coding platforms; systematic A/B testing of alternative adaptation strategies to identify conditions under which different modification approaches yield the greatest learning gains; and exploration of collaborative features, such as shared learning path annotation and peer progress comparison, enabled by extended OAuth scopes.

REFERENCES

- [1] R. Kop and A. Fournier, "New dimensions to self-directed learning in an open networked learning environment," *International Journal of Self-Directed Learning*, vol. 7, no. 2, pp. 1–20, 2010.
- [2] P. Brusilovsky and E. Millán, "User models for adaptive hypermedia and adaptive educational systems," in *The Adaptive Web*, Springer, 2007, pp. 3–53.
- [3] W. X. Zhao et al., "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.
- [4] Y. Wu et al., "AutoGen: Enabling next-generation LLM applications via multi-agent conversation," *arXiv preprint arXiv:2308.08155*, 2023.
- [5] Anthropic, "Model Context Protocol: A standard for AI-tool integration," *Technical Specification*, 2024. [Online]. Available: <https://modelcontextprotocol.io>
- [6] P. Brusilovsky, "Adaptive hypermedia," *User Modeling and User-Adapted Interaction*, vol. 11, no. 1–2, pp. 87–110, 2001.
- [7] C. Piech et al., "Deep knowledge tracing," in *Advances in Neural Information Processing Systems*, 2015, pp. 505–513.
- [8] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, no. 4, pp. 253–278, 1994.
- [9] Y. Kurdi et al., "A systematic review of automatic question generation for educational purposes," *International Journal of Artificial Intelligence in Education*, vol. 30, pp. 121–204, 2020.
- [10] T. Kiewra and A. F. Wise, "A review of educational chatbot applications," *Educause Review*, 2022.
- [11] K. Hauser et al., "Transformer-based models for question answering," in *Proc. EMNLP*, 2020, pp. 2574–2585.
- [12] M. Ghaleb et al., "Personalized learning paths using large language models," in *Proc. Learning Analytics and Knowledge (LAK)*, 2024.
- [13] B. P. Woolf, *Building Intelligent Interactive Tutors*. Morgan Kaufmann, 2009.
- [14] LangChain, "LangGraph: Multi-agent workflows," 2024. [Online]. Available: <https://langchain-ai.github.io/langgraph/>
- [15] J. Hattie and H. Timperley, "The power of feedback," *Review of Educational Research*, vol. 77, no. 1, pp. 81–112, 2007.
- [16] L. Van der Kleij et al., "Formative assessment and feedback: A review," *Educational Psychology Review*, vol. 27, no. 3, pp.459–473,2015.