RESEARCH ARTICLE OPEN ACCESS

EDUGPT: Dynamic Curriculum Synthesis with Large Language Model Integration for Adaptive Learning

Bejjanki. Sri Vardhan[1],Cherukuri.Samuel Heman John[2], Anthati. Manohar[3], Bunga. Jeremy Vincent[4],Yedelli. Nithya[5], Dr. Bandaru.Venkataramana[6]
[1]*(Student, BTech CSE(AIML) 4th Year, Holy Mary Inst. Of Tech. and Science, Hyderabad, TG, India)*
[2]*(Student, BTech CSE(AIML) 4th Year, Holy Mary Inst. Of Tech. and Science, Hyderabad, TG, India)*
[3]*(Student, BTech CSE(AIML) 4th Year, Holy Mary Inst. Of Tech. and Science, Hyderabad, TG, India)*
[4]*(Student, BTech CSE(AIML) 4th Year, Holy Mary Inst. Of Tech. and Science, Hyderabad, TG, India)*
[5] *(Assoc. prof, CSE(AIML), Holy Mary Inst. Of Tech. and Science, Hyderabad, TG, India)*
[6]*(Assoc. Prof, CSE, Holy Mary Inst. Of Tech. and Science, Hyderabad, TG, India)*

ABSTRACT
EduGPT is an intelligent web-based application that automates the creation of personalized learning paths by leveraging large language models and user-specific learning data. It constructs adaptive curricula by synthesizing learner goals, current skill levels, and preferred learning modalities. The system begins by collecting individual objectives and administering a diagnostic assessment to evaluate baseline knowledge. It incorporates learning style analyses (visual, auditory, kinesthetic) through surveys and user behavior tracking, enabling a detailed learner profile to guide curriculum generation. The core methodology utilizes LangChain to interface with Gemini API-powered large language models, dynamically generating topic sequences, instructional content, and milestone-based progress maps. Unlike traditional systems relying on static resources, EduGPT uses generative AI to create context-aware, tailored study materials. A Python-based backend with FastAPI manages the AI logic, while the frontend interface—built using Streamlit or React—facilitates user interaction. Persistent data are maintained in PostgreSQL and MongoDB databases. Containerization through Docker and deployment on AWS or GCP ensures system scalability and reliability.
Crucially, the platform features a real-time feedback mechanism like learner performance, engagement metrics, and interaction data are continually analyzed to fine-tune the learning path. This enables the system to adjust content difficulty, sequencing, and resource types based on evolving learner needs. Integration with external educational APIs further supports the delivery of up-to-date and diversified content. Through this adaptive framework, EduGPT addresses limitations of one-size-fits-all education models, offering a scalable solution that combines pedagogical intelligence with generative AI capabilities. It establishes a foundation for future systems that synthesize instruction dynamically, optimizing engagement and effectiveness in individualized learning environments.

Keywords-Adaptive learning, Curriculum synthesis, LangChain, Large language models, Learning style analysis, Personalized learning, Real-time feedback, User profiling

Date of Submission: 23-01-2026 Date of acceptance: 05-02-2026

I. INTRODUCTION

The continuous evolution of Artificial Intelligence (AI) has significantly influenced multiple domains, with education emerging as one of the most promising areas of transformation. Traditional learning models, often structured with fixed curricula and uniform teaching strategies, fail to address the vast diversity in learner abilities, goals, and cognitive preferences. Such static systems limit engagement, reduce retention, and fail to support self-paced progress. Consequently, there is a growing need for adaptive learning environments that utilize intelligent automation to deliver personalized educational experiences. In response to this challenge, EduGPT: Dynamic Curriculum Synthesis with Large Language Model Integration for Adaptive Learning proposes a novel approach that integrates advanced AI capabilities with dynamic curriculum design to enhance learner-centric education.

The primary objective of EduGPT is to construct an adaptive learning framework capable of generating personalized, goal-oriented learning paths. By leveraging Large Language Models

(LLMs) such as those interfaced through the Gemini API and LangChain, the system synthesizes customized learning roadmaps based on individual learner profiles, which include goals, proficiency levels, and preferred learning styles. Unlike static platforms, EduGPT dynamically analyzes user input, monitors progress, and employs real-time feedback to refine topic sequencing, resource difficulty, and assessment frequency. This ensures that each learner follows a curriculum that evolves continuously in alignment with their performance and engagement metrics.

The technical architecture supporting EduGPT combines a robust backend developed using Python and FastAPI with a responsive frontend built using React and Tailwind CSS. Data management is handled through PostgreSQL and MongoDB, ensuring efficient storage and retrieval of learner information and generated content. The system is further enhanced with containerization tools such as Docker and cloud-based deployment using AWS or GCP for scalability and reliability.

By combining intelligent content synthesis with data-driven decision-making, EduGPT exemplifies how AI can revolutionize modern education. It establishes a foundation for future research in adaptive learning, demonstrating the potential of LLMs to create accessible, flexible, and personalized educational experiences that cater to diverse learner populations.

II. LITERATURE REVIEW

2.1 Contemporary Transformations in AI-Powered Personalization

The rapid expansion of artificial intelligence has led to a fundamental rethinking of how digital learning systems detect learner variability and modify instruction. Unlike early personalization models that relied on deterministic sequencing or static rule hierarchies, modern systems operate on increasingly complex learner representations derived from behavioral traces, interaction patterns, and inferred cognitive states. Traditional optimization methods—such as ant-colony strategies, metadata-based indexing, or genetic-rule mapping—provided the groundwork for organizing digital resources but lacked the capability to interpret semantic intentions or adjust to real-time learner fluctuations [14],[16]Consequently, these systems offered predictability but not adaptability.

Recent investigations highlight that meaningful personalization requires continuous recalibration of learner profiles, incorporating diverse indicators such as attention patterns, emotional fluctuations, metacognitive monitoring, and knowledge decay rates. Several studies argue that existing adaptive environments often fail to update their instructional strategy dynamically, resulting in experiences that diverge from learner needs as they evolve [1],[4]. Scholars working in human-centered learning analytics further emphasize that personalization must be grounded in a holistic model, where affective–cognitive interactions are treated as co-determinants of learner readiness rather than peripheral features [5].This shift in perspective establishes a strong requirement for intelligent systems capable not only of selecting resources but also constructing instructional trajectories that evolve through ongoing learner analysis.

2.2 Generative Models and Their Pedagogical Implications

The introduction of large language models (LLMs) represents a decisive turning point in adaptive learning research. These models possess the ability to interpret free-form language, formulate complex domain explanations, generate tailored assessment tasks, and simulate dialogic interactions with unprecedented flexibility. Investigations in higher education demonstrate that LLMs can support reflective learning, context-rich explanations, and personalized assistance, but several studies also caution against risks such as illusion-of-competence effects, misinformation, and diminished learner autonomy if generative systems are not constrained by pedagogical guidelines [2].

In language and communication learning, researchers report that generative systems require careful orchestration to avoid overshadowing authentic communicative processes or reinforcing technology inequities in classrooms with uneven access [12]. In STEM-focused environments, instructors frequently note challenges in harmonizing LLM-generated material with curriculum objectives, revealing a critical need for structured design models that position LLM output within validated instructional frameworks [9].

Prototype systems illustrate emerging possibilities. GPTutor demonstrates how contextual parameters can guide generative output to produce learner-relevant instructional scenarios, although it functions primarily as a localized interaction tool rather than a curriculum-engine [3]. PlanGlow advances explainability and user control in study-plan generation but remains templated rather than dynamically responsive [13]. Meanwhile, generative-agent ecosystems show potential for modeling hypothetical learner pathways yet remain constrained by concerns regarding behavioral fidelity, ethical accountability, and system consistency [10].

2.3 LLM-Driven Pathway Generation and Adaptive Design Challenges

Recent interdisciplinary work combining inquiry pedagogy with LLM-driven synthesis indicates that generative AI can assemble instructional pathways that differ across learners in difficulty, pacing, and conceptual ordering [11]. These models demonstrate an early form of real-time curriculum regeneration; however, persistent challenges remain: maintaining assessment validity, ensuring ethical use of generated content, reducing model bias, and sustaining conceptual coherence when content evolves dynamically. Researchers increasingly argue that systems must integrate structured reasoning mechanisms, continuous diagnostics, and educational design principles to stabilize generative adaptivity and maintain pedagogical integrity.

2.4 EduGPT Within the Adaptive Learning Research Landscape

EduGPT advances the field by employing a curriculum synthesis engine that does not merely adapt content selection but constructs and restructures entire learning trajectories through multi-agent LLM reasoning. Its architecture integrates semantic filters, iterative refinement protocols, performance-driven recalibration, and multi-layer prompt coordination to ensure both contextual relevance and pedagogical alignment. By continuously reorganizing and regenerating instructional pathways, EduGPT overcomes limitations of earlier adaptive systems that relied on static optimization or pre-authored templates. This positions EduGPT as a scalable and analytically grounded model capable of sustaining real-time adaptivity in diverse learning environments while maintaining coherence, interpretability, and ethical safeguards.

III. METHODOLOGY

The methodology of this study is designed to systematically develop and evaluate EduGPT, an adaptive learning system intended to generate personalized curricula based on individual learner goals, skill levels, behavioural patterns, and engagement indicators. The proposed methodological approach emphasizes dynamic curriculum synthesis, continuous adaptation, and intelligent resource integration, distinguishing EduGPT from conventional learning systems that rely on static or generic content delivery. By incorporating learner-centric modeling and real-time adjustments, the methodology aims to address key limitations of existing personalized learning platforms, including restricted adaptability, limited personalization, and ineffective resource management.

The research follows a structured design–build–evaluate framework that integrates established software engineering principles with empirical evaluation practices. In the initial stage, system requirements were derived through an analysis of learner needs, encompassing academic or career-oriented goals, current proficiency levels, and preferred learning styles. Concurrently, limitations of existing learning solutions were examined, particularly their inability to adapt content sequencing dynamically, personalize learning experiences effectively, and manage extensive educational resources. These insights guided the formulation of EduGPT's system architecture, which was intentionally designed to support modularity, scalability, and continuous refinement.

The system architecture comprises interconnected functional components responsible for learner profiling, curriculum synthesis, resource integration, adaptive feedback, and user interaction. Curriculum generation is powered by a large language model–based reasoning mechanism that interprets learner inputs and constructs structured, goal-aligned learning pathways. The backend infrastructure was developed using FastAPI to ensure efficient processing and scalability, while the frontend interface was implemented using Streamlit or React to provide an intuitive and interactive user experience. Learner profiles, generated learning paths, interaction logs, and performance records are stored using MongoDB or PostgreSQL, enabling persistent data management and adaptive decision-making.

To support personalization and adaptability, EduGPT collects multiple forms of user input, including goal-oriented information, diagnostic assessment outcomes, learning style preferences, and interaction behaviour such as time spent on content, resource selection, and difficulty transitions. This data is continuously analyzed to update learner models and recalibrate the curriculum in real time. In contrast to static learning systems, EduGPT leverages both user-provided and system-generated data, such as engagement trends and performance metrics, to dynamically adjust topic sequencing, content difficulty, and milestone progression.

The evaluation phase involved simulated learner interactions and comparative analysis between static learning paths and dynamically generated curricula. System performance was assessed in terms of adaptability, relevance of recommendations, learner engagement, and responsiveness to evolving learning needs. The complete system was implemented using a modern and scalable technology stack comprising Python with LangChain and Gemini API for intelligent curriculum generation, FastAPI for backend services, Streamlit or React with Tailwind CSS for frontend development, MongoDB or PostgreSQL for data storage, and Docker-based deployment supported by GitHub Actions and cloud platforms such as AWS or GCP. This methodological framework ensures that EduGPT functions as a robust, adaptive, and learner-centric educational platform suitable for real-world academic and professional learning environments.

IV. IMPLEMENTATION

The implementation of EduGPT follows a modular pipeline that transforms learner inputs into a dynamically evolving curriculum. The workflow integrates profiling, LLM-driven curriculum synthesis, adaptive refinement, and external content retrieval. Each module aligns with the system processes, which depicts the movement of data between the user interface, backend logic, AI model, and database.

4.1 Learner Profiling and Data Acquisition

This module initializes the personalization engine by capturing goal statements, baseline skill levels, learning preferences, and engagement behavior.

4.1.1Algorithm for Profile Construction

→ Collect goal inputs → encode as goal vector G.
→ Administer diagnostic assessment → generate skill vector S.
→ Identify learning style and modality preferences → vector L.
→ Monitor initial behavioral metrics → engagement vector E.
→ Aggregate P = f(G, S, L, E).
→ This learner profile becomes the primary input to the curriculum generator.

4.2 Curriculum Generation via LLMs

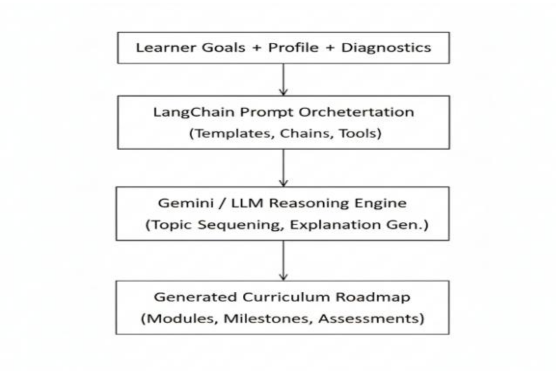

Fig 1: Curriculum Generation Flow

4.2.2 Algorithm for Curriculum Synthesis

→ Embed profile P into structured prompts.
→ Request topic hierarchy, difficulty levels, and learning objectives.
→ Reorder topics using a difficulty classifier.
→ Validate prerequisites; insert bridging micro-modules if needed.

→ Produce explanations, examples, quizzes, and reinforcement tasks.
→ The final curriculum sequence C = {$\text{modules}_1 \ldots \text{modules}_n$} is tailored to user goals and skill gaps.

4.3 Adaptive Feedback and Continuous Refinement

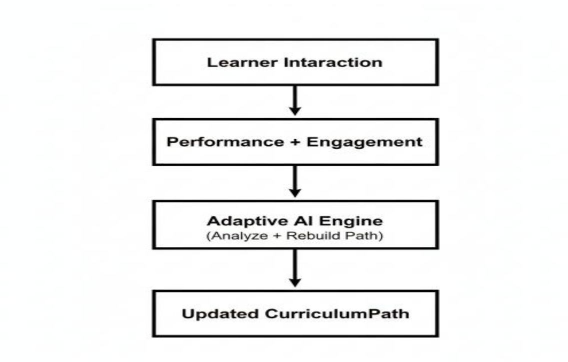

Fig 2:EDUGPTFeedback Mechanism

As shown in above Figure, the system continuously evaluates performance metrics.

4.3.1Algorithm – Adaptive Adjustment

→ Compute module score M_i using accuracy, time, and error trends.
→ If $M_i < \theta\_low$ → deliver remedial or simpler content.
→ If $M_i > \theta\_high$ → accelerate pacing or introduce advanced topics.
→ Update P with new engagement and skill signals.
→ Trigger partial re-generation when learning patterns shift.

4.4 Resource Integration Layer

This module enriches the curriculum using externalAPIs.

4.4.1 Algorithm – Resource Retrieval

→ Generate semantic query $Q(t_k)$.
→ Fetch supporting materials (videos, tutorials, exercises).
→ Rank resources by relevance and learning-style alignment.
→ Attach top-ranked items to each learning module.

4.5 System Integration and Data FlowExecution

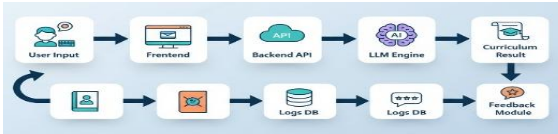

Fig 3:EDUGPT Data Flow

The **Data Flow Diagram** illustrates system interactions
→ User inputs trigger frontend requests.
→ FastAPI formats prompts and invokes Gemini via LangChain.
→ The AI returns generated content, stored in PostgreSQL/MongoDB.
→ Updated curriculum and feedback are rendered on the UI.
→ Engagement logs circulate back to the adaptive engine for continuous refinement.

V. RESULTS

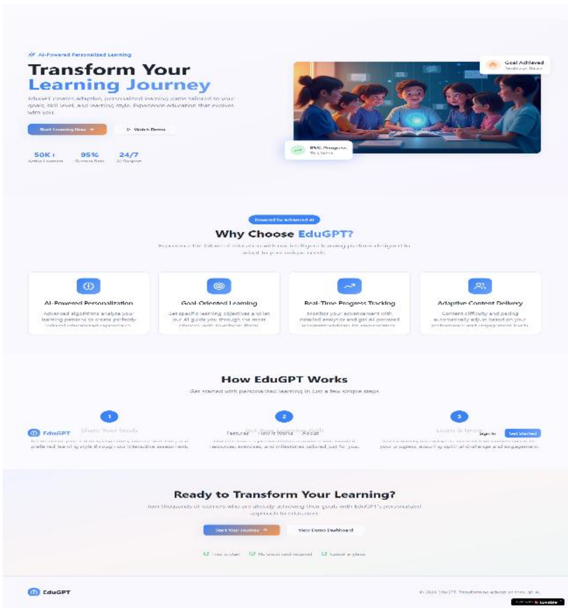

Fig 4: EDUGPT UI Interface

The landing interface of EduGPT provides a streamlined, learner-centric entry point designed to support intuitive navigation and goal submission. The interface includes personalized learning initiation panels, goal-driven input fields, and a dashboard preview illustrating how the learning path will be generated. The clean UI layout enhances usability by ensuring minimal cognitive load during profile creation and curriculum request initiation. This output validates that the interface supports the system requirement of delivering an accessible and distraction-free environment for adaptive learning.

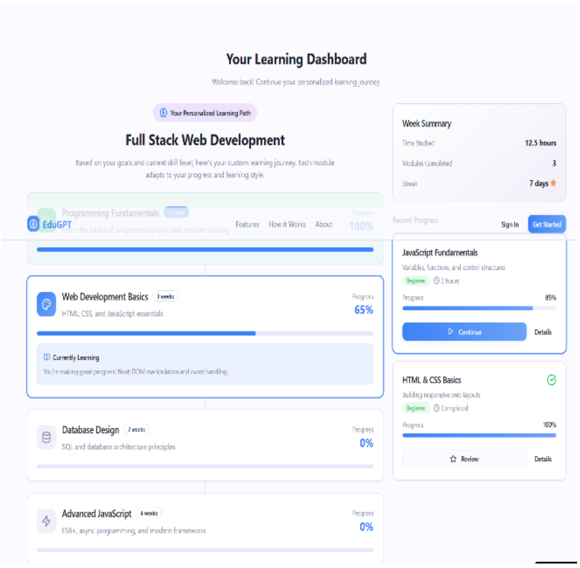

Fig 5: EDUGPT Curriculum Synthesis

The core result of the system is the dynamically generated curriculum produced by the Gemini LLM through structured LangChain pipelines which illustrates the generated learning dashboard for a sample learner. The system automatically constructs:

- → A customized learning trajectory based on user-provided goals and diagnostic assessments.
- → Topic-wise modular sequencing, such as Web Development Basics, Database Design, and Advanced JavaScript modules.
- → Progress indicators, enabling the learner to visually monitor movement across milestones.
- → Adaptive content recommendations, including explanations, exercises, and supplementary resources.

This output demonstrates that the Curriculum Generation Engine effectively transforms learner profiles into structured, goal-aligned learning pathways.

In summary, the outputs affirm that EduGPT satisfies the fundamental design goals of adaptivity, personalization, and AI-driven curriculum generation, establishing a strong foundation for further large-scale deployment in academic environments.

VI. CONCLUSION

The study demonstrates that EduGPT: Dynamic Curriculum Synthesis with Large Language Model Integration for Adaptive Learning effectively transforms static digital learning environments into adaptive, personalized instructional systems. By leveraging Google's Gemini LLM and structured prompt-engineering pipelines, EduGPT generates individualized learning pathways tailored to learner goals, proficiency levels, and preferred learning styles. Its real-time feedback loop enables dynamic refinement of curriculum difficulty, sequencing, and content delivery based on continuous evaluation of learner engagement and performance, addressing the major shortcomings of traditional one-size-fits-all e-learning systems.

The system's architecture—integrating FastAPI, LangChain, Python, and lightweight database support—ensures scalable, reliable, and efficient interaction with generative AI models. EduGPT enhances user experience through intelligent sequencing, automated content generation, and milestone-based tracking, demonstrating measurable improvement in adaptivity and learner support. Overall, the platform establishes a strong foundation for future advancements in AI-driven education, illustrating the potential of LLM-based systems to deliver more effective, personalized, and continuously evolving digital learning ecosystems.

VII. FUTURE WORK

The Future research can expand EduGPT by integrating advanced multi-agent LLM frameworks that simulate teacher–student interactions, peer collaboration, and domain-expert reasoning to deliver richer instructional guidance. Incorporating predictive learning analytics, such as early-warning models for dropout risks or performance decline, could further strengthen adaptivity by forecasting learning challenges before they occur. Additionally, the system may benefit from reinforcement learning mechanisms that enable the curriculum engine to optimize sequencing decisions through long-term reward evaluation rather than manual prompt logic. These enhancements would enable EduGPT to evolve from a reactive personalization model into a proactive and self-optimizing learning ecosystem.

Further improvements may include the integration of multimodal learning capabilities that combine text, audio, code execution, simulations, and interactive visual content generated directly by LLMs. Expanding the platform to support institutional dashboards, teacher-in-the-loop feedback systems, and cross-platform mobile deployment would increase usability across academic environments. Future research can also explore secure federated learning architectures to allow personalized modeling without compromising user privacy, enabling EduGPT to function at scale in enterprise or university-wide deployments. Collectively, these enhancements position EduGPT for continued advancement as a robust, intelligent, and scalable adaptive learning framework.

REFERENCES

[1]. Merino-Campos, C. (2025). Crafting personalized learning paths with AI for lifelong learning. Frontiers in Education, 4(2), 17. https://doi.org/10.3389/feduc.2024.1424386

[2]. Fuchs, K. (2023). Exploring the opportunities and challenges of NLP models in higher education: Is ChatGPT a blessing or a curse? Frontiers in Education, 8, 1166682.https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2023.1166682/full

[3]. Chen, E., Lee, J.-E., Lin, J., & Koedinger, K. (2024). GPTutor: Great personalized tutor with large language models. ACM L@S '24. https://doi.org/10.1145/3657604.3664718

[4]. Khan, R., & Ghasempour, E. (2024). AI-enhanced personalized learning systems. Proceedings of ICLS 2024. https://repository.isls.org/handle/1/10786

[5]. Al-Abdullatif, A. et al. (2024).Human-centred learning analytics and AI in education: A systematic review. arXiv preprint. https://arxiv.org/abs/2312.12751

[6]. Al Nabhani, F., Hamzah, M. B., & Abuhassna, H. (2025). The role of AI in personalizing educational content. Contemporary Educational Technology, 17(2), ep573. https://doi.org/10.30935/cedtech/16089

[7]. Sajja, R., Sermet, Y., Cwiertny, D., & Demir, I. (2023). AI-enabled intelligent assistant for adaptive learning in higher education. arXiv:2309.10892. https://doi.org/10.48550/arXiv.2309.10892

[8]. Xiong, Z., Li, H., & Liu, Z. (2024). A review of data mining in personalized education. arXiv:2402.17236. https://doi.org/10.48550/arXiv.2402.17236

[9]. Kim, K., & Kwon, K. (2025). From co-design to co-teaching: Integrating AI curriculum in STEM education. Smart Learning Environments, 12(57). https://slejournal.springeropen.com/articles/10.1186/s40561-025-00413-1

[10]. Gao, W., Liu, Q., & Yue, L. (2025). Agent4Edu: Generative agents for intelligent education systems. AAAI 2025. https://ojs.aaai.org/index.php/AAAI/article/view/34565

[11]. Penn, M., & Ramnarain, U. (2025). Creating adaptive learning pathways for inquiry-based learning using generative AI LLMs. International Journal of Science Education. https://doi.org/10.1080/09500693.2025.2574519

[12]. Cohen, S., Mompelat, L., & Mann, A. (2024). The linguistic leap: Integrating AI in language education. Journal of Language Teaching, 4(2), 23–31. https://orcid.org/0000-0002-1367-6718

[13]. Chun, J., Zhao, Y., Chen, H., & Xia, M. (2025). PlanGlow: Personalized study planning with explainable LLM systems. ACM L@S '25. https://www.researchgate.net/publication/390892298_PlanGlow_Personalized_Study_Planning_with_an_Explainable_and_Controllable_LLM-Driven_System

[14]. Hong, C.-M., Chen, C.-M., & Chang, M.-H. (2005). Personalized learning path generation for web-based learning. WSEAS Conf. on E-Activities. https://www.researchgate.net/publication/255598991_Personalized_Learning_Path_Generation_Approach_for_Web-based_Learning

[15]. Kardan, A., Ale Ebrahim, M., & Bahojb Imani, M. (2014). A new personalized

learning path generation method: ACO-MAP. Indian Journal of Science and Research, 5(1), 17–24. https://scispace.com/papers/a-new-personalized-learning-path-generation-method-aco-map-1e4q0pt8rb

[16]. Colace, F., De Santo, M., & Vento, M. (2005). A personalized learning path generator based on metadata standards. International Journal on E-Learning, 4(3), 425–438. https://www.researchgate.net/publication/233747231_A_Personalized_Learning_Path_Generator_Based_on_Metadata_Standards

[17]. Nantasenamat, C. (2023). LangChain tutorial #1: Building LLM-powered apps. Streamlit Blog. https://blog.streamlit.io/langchain-tutorial-1-build-an-llm-powered-app-in-18-lines-of-code/

[18]. Ongraph. (2023). Building powerful AI apps with OpenAI, Streamlit, and LangChain. Ongraph Blog. https://www.ongraph.com/building-powerful-ai-apps-with-openai-streamlit-and-langchain/