

Tutor AI: Phương pháp hướng dẫn từng bước cá nhân hóa trong các hệ thống học tập thích ứng

Lâm Du Đạt^{a*}, Trần Thành Thắng^b

Tóm tắt:

Phản hồi thích ứng theo ngữ cảnh vẫn là một thách thức quan trọng trong giáo dục, đặc biệt khi cần hướng dẫn người học giải quyết các bài tập phức tạp thuộc nhiều lĩnh vực khác nhau. Nghiên cứu này giới thiệu một khung hướng dẫn học tập theo từng bước có khả năng tổng quát hóa, sử dụng mô hình ngôn ngữ lớn (LLM) để hỗ trợ học tập thông qua phân rã nhiệm vụ thành các bước nhỏ, đánh giá tiến trình và sinh phản hồi phù hợp với trạng thái học tập của người học. Khung này được triển khai trong hệ thống Tutor AI và được thử nghiệm thực tế trong học phần Kỹ thuật lập trình. Trong môi trường này, sinh viên tương tác với hệ thống viết mã thời gian thực, lộ trình học tập được thiết kế rõ ràng và phản hồi được sinh tự động qua mô hình Gemini LLM. Mỗi bài làm được kết hợp động với lịch sử học tập của người học để hình thành prompt giàu ngữ cảnh, từ đó hệ thống xác định chính xác bước học hiện tại và cung cấp hướng dẫn thích ứng theo tiến trình. Kết quả thực nghiệm trong học kỳ của môn Kỹ thuật lập trình (ngôn ngữ C) đã minh chứng hiệu quả của hệ thống, giúp nâng cao đáng kể mức độ tương tác, thúc đẩy tinh thần tự học và giảm sự phụ thuộc của sinh viên vào các lời giải có sẵn. Kiến trúc của Tutor AI được thiết kế với tính tổng quát cao, cho phép mở rộng ứng dụng không chỉ trong lập trình mà còn sang nhiều lĩnh vực khác như kỹ thuật, phân tích dữ liệu, khoa học xã hội hay bất kỳ môn học nào có bài tập với các bước làm rõ ràng. Kết quả nghiên cứu này góp phần phát triển các hệ thống dạy học thông minh có khả năng diễn giải và mở rộng quy mô, đồng thời đặt nền tảng cho việc học tập theo từng bước được hỗ trợ bởi AI trong nhiều bối cảnh giáo dục hiện đại.

Từ khóa: *hệ thống dạy học thông minh, mô hình ngôn ngữ lớn (LLM), học theo từng bước, phản hồi theo ngữ cảnh, đánh giá tiến trình*

^a Trường Đại học Đông Á; 33 Xô Viết Nghệ Tĩnh, phường Hòa Cường, quận Hải Châu, Đà Nẵng.
e-mail: datld@donga.edu.vn

^b Trường Đại học Đông Á; 33 Xô Viết Nghệ Tĩnh, phường Hòa Cường, quận Hải Châu, Đà Nẵng.
e-mail: thangtt@donga.edu.vn

* Tác giả chịu trách nhiệm chính.

Tutor AI: A Step-by-Step Personalized Guidance Approach in Adaptive Learning Systems.

Lam Du Dat^{a*}, Tran Thanh Thang^b

Abstract:

Context-aware adaptive feedback remains a significant challenge in education, particularly when guiding learners through complex problem-solving tasks across diverse disciplines. This study introduces a generalized stepwise tutoring framework that leverages large language models (LLMs) to support learning by decomposing tasks into micro-steps, assessing progress, and generating feedback tailored to the learner's current state. The framework is implemented in the Tutor AI system and was empirically evaluated in a Programming Techniques course. In this environment, students interact with a real-time coding interface, where the learning path is clearly structured, and feedback is automatically generated via the Gemini LLM. Each student's submission is dynamically integrated with their learning history to construct context-rich prompts, enabling the system to precisely identify the current learning step and provide adaptive, scaffolded guidance accordingly. Experimental results from a semester-long Programming Techniques (C language) course demonstrated the system's effectiveness in significantly increasing engagement, fostering self-regulated learning, and reducing reliance on direct solutions. The architecture of Tutor AI is highly generalizable, allowing for expansion beyond programming to various other disciplines such as engineering, data analysis, social sciences, or any subject with clearly defined stepwise exercises. This research contributes to the development of explainable and scalable intelligent tutoring systems, laying a foundation for AI-supported, stepwise learning in modern educational contexts.

Keywords: *intelligent tutoring systems, large language models (LLMs), stepwise learning, contextual feedback, formative assessment*

Received: 10.7.2025; Accepted: 15.02.2026; Published: 28.2.2026

DOI: 10.59907/daujs.5.1.2026.453

^a Dong A University; 33 Xo Viet Nghe Tinh, Hoa Cuong Ward, Hai Chau District, Danang City, Vietnam.
e-mail: datld@donga.edu.vn

^b Dong A University; 33 Xo Viet Nghe Tinh, Hoa Cuong Ward, Hai Chau District, Danang City, Vietnam.
e-mail: thangtt@donga.edu.vn

* Corresponding author.

Đặt vấn đề

Bài tập đóng vai trò quan trọng trong giáo dục, đặc biệt trong các lĩnh vực yêu cầu phát triển tư duy phản biện và khả năng giải quyết vấn đề. (Kosar et al., 2024) Bài tập không chỉ giúp người học củng cố kiến thức mà còn phát triển các kỹ năng thực hành cần thiết cho sự nghiệp sau này. (Carlsen et al., 2006) Tuy nhiên, sự gia tăng độ phức tạp của các bài tập và yêu cầu cá nhân hóa học tập đang đặt ra thách thức lớn đối với hệ thống giáo dục hiện nay. (Alaqsam et al., 2021) Các phương pháp truyền thống, mặc dù hiệu quả trong việc kiểm tra kiến thức, lại gặp khó khăn trong việc cung cấp phản hồi cá nhân hóa kịp thời, đặc biệt là trong môi trường lớp học nhiều sinh viên. (De Silva et al., 2023)

Một vấn đề nổi bật là thiết kế các hệ thống hỗ trợ học tập có khả năng thích ứng với tiến trình học của từng sinh viên. Trong các lớp học đông, giảng viên không thể hỗ trợ tất cả sinh viên kịp thời, dẫn đến nhu cầu cấp thiết về phản hồi tức thì và hướng dẫn theo tiến trình học của mỗi sinh viên. (Kosar et al., 2024) Hệ thống truyền thống chủ yếu dựa vào phản hồi chung và thiếu khả năng điều chỉnh theo mức độ tiến bộ của từng học viên, làm giảm hiệu quả hỗ trợ học tập. (De Silva et al., 2023)

Mới đây, sự phát triển của các mô hình ngôn ngữ lớn (Large Language Models - LLMs) như GPT-3.5, Gemini và Claude đã mở ra cơ hội mới cho việc hỗ trợ học tập. Những mô hình này có khả năng hiểu ngữ cảnh và sinh ra phản hồi linh hoạt, giúp hướng dẫn người học qua từng bước nhỏ và cung cấp sự hỗ trợ theo yêu cầu. Tuy nhiên, ứng dụng LLM trong giáo dục vẫn còn hạn chế, đặc biệt trong việc triển khai phản hồi chi tiết theo tiến trình học và khả năng theo dõi sự phát triển của sinh viên trong suốt quá trình học.

Một điểm yếu rõ rệt của các hệ thống hiện tại là thiếu sự kết hợp giữa hướng dẫn theo bước (stepwise guidance) và phản hồi thích ứng theo ngữ cảnh (context-aware adaptive feedback). Để phát huy tối đa tiềm năng của các hệ thống học tập thông minh, cần tích hợp khả năng hướng dẫn theo từng bước, kết hợp với đánh giá tiến trình và phản hồi thích ứng, đặc biệt trong môi trường học tập đa lĩnh vực. (Kosar et al., 2024) Hệ thống cần có khả năng tự động điều chỉnh phản hồi dựa trên tiến độ học của người học, giúp sinh viên hiểu rõ hơn về những gì cần cải thiện và phát triển kỹ năng. (De Silva et al., 2023)

Nghiên cứu này đề xuất và trình bày Tutor AI, một hệ thống được thiết kế nhằm giải quyết những thách thức trên. Tutor AI hỗ trợ người học thực hiện bài tập theo từng bước nhỏ, sử dụng mô hình LLM (Gemini) để tạo phản hồi chi tiết, thích ứng dựa trên tiến trình và ngữ cảnh học tập của sinh viên. Hệ thống này đã được triển khai và khảo sát thực tế trong học phần Kỹ thuật Lập trình (ngôn ngữ C). Bằng cách tích hợp cơ chế hướng dẫn từng bước với khả năng phân tích và phản hồi của LLM, Tutor AI hướng tới việc tạo ra một môi trường học tập chủ động, được cá nhân hóa và hiệu quả hơn trong giáo dục hiện đại.

Nội dung nghiên cứu

Liên hệ nghiên cứu

Hệ thống hướng dẫn học tập thông minh đã đóng vai trò quan trọng trong việc cung cấp giáo dục cá nhân hóa trong nhiều lĩnh vực, đặc biệt là trong các môn học có tính thực hành cao như lập trình. (Kosar et al., 2024) Các hệ thống hướng dẫn học tập thông minh truyền thống, mặc dù thành công trong việc mô phỏng vai trò của người hướng dẫn và cung cấp phản hồi, vẫn còn nhiều hạn chế, đặc biệt trong việc cung cấp phản hồi linh hoạt và thích ứng với tiến trình học của từng sinh viên. Các hệ thống ban đầu như SmartLab (Smith và cộng sự, 2012) và WebToTeach (García-Sánchez và cộng sự, 2015) sử dụng các quy tắc định sẵn để hỗ trợ sinh viên trong các bài tập lập trình. Tuy nhiên, phản hồi của chúng chủ yếu mang tính tĩnh và thiếu khả năng thích ứng với các cách giải quyết khác nhau của người học. Tương tự, hệ thống iPAT (Patel và cộng sự, 2018) đã tích hợp phân tích hành vi người học để điều chỉnh phản hồi, nhưng vẫn chưa có khả năng sinh phản hồi ngữ nghĩa hoặc xử lý các lỗi không nằm trong kịch bản lập trình sẵn.

Với sự ra đời của các mô hình ngôn ngữ lớn (LLMs) như GPT-3.5 và Gemini API, cơ hội để cải thiện các ITS đã mở ra. Các mô hình như Codex và ChatGPT đã chứng tỏ khả năng sinh mã và phát hiện lỗi trong lập trình, giúp người học cải thiện kỹ năng lập trình. Tuy nhiên, hệ thống CodeCoach (Jayasuriya et al., 2023), mặc dù sử dụng GPT-3.5 để cung cấp phản hồi cho sinh viên, lại chỉ cung cấp phản hồi toàn cục và thiếu khả năng cung cấp phản hồi chi tiết theo từng bước học. Điều này đặc biệt quan trọng trong các bài tập phức tạp, nơi mỗi bước học cần được hướng dẫn cụ thể và dễ hiểu.

Hệ thống ChatGPT, mặc dù có khả năng tạo phản hồi linh hoạt, lại thiếu khả năng theo dõi tiến trình học và cung cấp hướng dẫn chi tiết qua từng bước của bài tập. Kosar et al. (2024) đã chỉ ra rằng mặc dù LLMs có thể tạo ra phản hồi chính xác, chúng không thể kiểm soát lộ trình học tập của người học một cách có hệ thống, dẫn đến việc thiếu khả năng theo dõi tiến trình và phản hồi thích ứng với nhu cầu học của từng sinh viên.

Một khoảng trống lớn trong các hệ thống hiện tại là thiếu sự kết hợp giữa hướng dẫn theo bước và đánh giá tiến trình học, đặc biệt trong môi trường học tập đa lĩnh vực. Các hệ thống như CodeCoach và ChatGPT không thể theo dõi và đánh giá tiến trình học của người học qua từng bước cụ thể, làm cho phản hồi không thể cá nhân hóa hoặc giúp sinh viên nhận ra điểm yếu của mình.

Hệ thống được đề xuất trong nghiên cứu này không chỉ sử dụng các mô hình ngôn ngữ lớn (LLMs) để sinh phản hồi, mà còn tích hợp cơ chế theo dõi tiến trình học tập cá nhân và cung cấp phản hồi thích ứng theo từng bước học. Cách tiếp cận này là sự kết hợp giữa nguyên lý Zone of Proximal Development (ZPD) của Vygotsky và năng lực của trí tuệ nhân tạo trong việc phát triển các chiến lược hỗ trợ học tập có tính phản hồi và giám sát

liên tục. (Kosar et al., 2024; De Silva et al., 2023) Hệ thống không chỉ hỗ trợ hiệu quả trong việc giảng dạy lập trình mà còn có tiềm năng mở rộng sang các lĩnh vực khác như toán học, phân tích dữ liệu và viết học thuật. Nhờ khả năng thích ứng và định hướng theo tiến trình học, giải pháp này thúc đẩy việc học tập chủ động, giảm sự phụ thuộc vào lời giải sẵn có, đồng thời tăng cường tư duy độc lập và kỹ năng giải quyết vấn đề của người học.

Kiến trúc hệ thống Tutor AI

Hệ thống Tutor AI được thiết kế để hỗ trợ người học qua các bài tập học tập theo từng bước, cung cấp phản hồi thích ứng và đánh giá tiến trình cá nhân hóa, từ đó tạo ra một môi trường học tập hiệu quả và chủ động. Kiến trúc hệ thống được phân thành năm lớp chính, giúp tối ưu hóa quá trình học tập của người học và đảm bảo tính linh hoạt và khả năng mở rộng cho nhiều môn học khác nhau, như minh họa tại Hình 1.

Layer	Thành phần			
1: Giao diện người học	- DS buổi học - Danh sách bài tập	Khung bài làm	- Kiểm tra - Trợ giúp	Khung phản hồi
2: Xử lý và điều phối	Điều phối bài làm & tương tác LLM	Cập nhật dữ liệu học tập & log	Thực hiện đánh giá tiến trình	
3: LLM	Gemini LLM			
4: Dữ liệu học tập	Các bước cụ thể của từng bài tập		Log người học	
5: Đánh giá & Phân tích	Phân tích học tập		Đánh giá quá trình học tập	

Hình 1. Kiến trúc phân lớp hệ thống Tutor AI hỗ trợ học theo từng bước

- **Giao diện người học:** Giao diện này là điểm giao tiếp chính giữa người học và hệ thống, bao gồm danh sách bài tập, không gian làm bài, và các chức năng “Kiểm tra” và “Trợ giúp”. Giao diện có thể thay đổi linh hoạt theo loại bài tập, giúp người học dễ dàng yêu cầu trợ giúp khi gặp khó khăn. Mỗi bài tập sẽ được hiển thị với các bước rõ ràng, giúp người học tập trung vào từng bước học.

- **Xử lý và điều phối:** Lớp này đóng vai trò trung tâm trong việc điều phối luồng học tập và quản lý tương tác. Nó chịu trách nhiệm điều phối quá trình làm bài của người học và tương tác với lớp LLM, bao gồm việc tạo prompt phù hợp dựa trên ngữ cảnh và lịch sử học tập. Lớp này cũng đảm nhiệm việc cập nhật dữ liệu học tập và log của người học (đọc/ghi từ Lớp Dữ liệu học tập) sau mỗi tương tác. Cuối cùng, lớp này thực hiện đánh giá tiến trình theo từng bước học để duy trì nhịp độ học tập nhất quán.

- LLM: Lớp này chứa các mô hình ngôn ngữ lớn, mà cụ thể là Gemini LLM, đóng vai trò cốt lõi trong việc sinh phản hồi thông minh và thích ứng. Mô hình LLM tiếp nhận các prompt đã được xử lý từ Lớp Xử lý và điều phối, sau đó phân tích nội dung bài làm của người học để đưa ra các gợi ý định hướng hoặc giải thích lý thuyết liên quan.

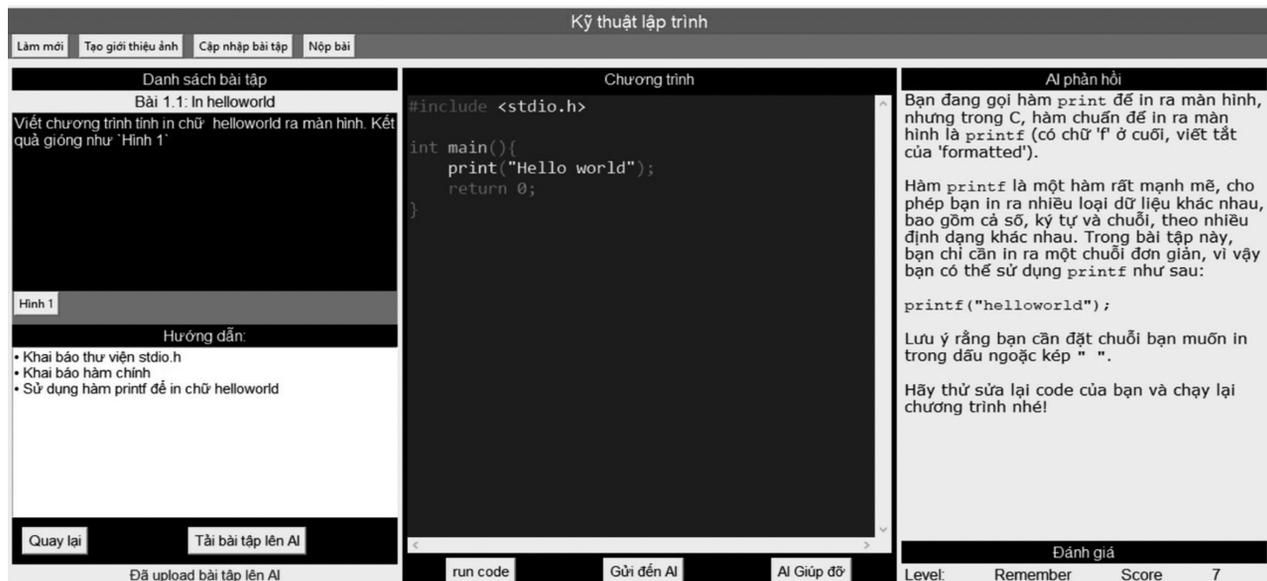
- Dữ liệu học tập: Các bài tập và tiến trình học được mã hóa và lưu trữ trong hệ thống. Hệ thống sử dụng các tệp để định nghĩa cấu trúc phân cấp của khóa học, bao gồm các buổi học, bài tập và các bước nhỏ trong từng bài tập. Toàn bộ dữ liệu tương tác của người học cũng được lưu lại, giúp hệ thống theo dõi tiến trình học và cung cấp phản hồi chi tiết theo từng bước.

- Đánh giá và Phân tích: Lớp này đánh giá tiến trình học của người học và thực hiện các chiến lược đánh giá quá trình học tập. Mỗi bước học được đánh giá độc lập, giúp người học nhận ra điểm yếu và cải thiện kỹ năng qua từng bước. Dữ liệu về tiến trình học được lưu lại và phân tích để tối ưu hóa chiến lược phản hồi và hỗ trợ học tập.

Mô hình phân lớp này đảm bảo tính module hóa, cho phép hệ thống linh hoạt, dễ dàng bảo trì và điều chỉnh cho các kịch bản ứng dụng khác nhau trong tương lai.

Giao diện người dùng

Giao diện Tutor AI tối ưu hóa trải nghiệm học tập, chia thành ba khu vực chính (như Hình 2):



Hình 2. Giao diện học tập của Tutor AI

- Khung bên trái: Liệt kê các buổi học và bài tập, giúp sinh viên theo dõi tiến trình học.

- Bài làm: Cung cấp không gian để người học thực hiện bài tập, hỗ trợ tô màu cú pháp và cung cấp phản hồi theo ngữ cảnh.

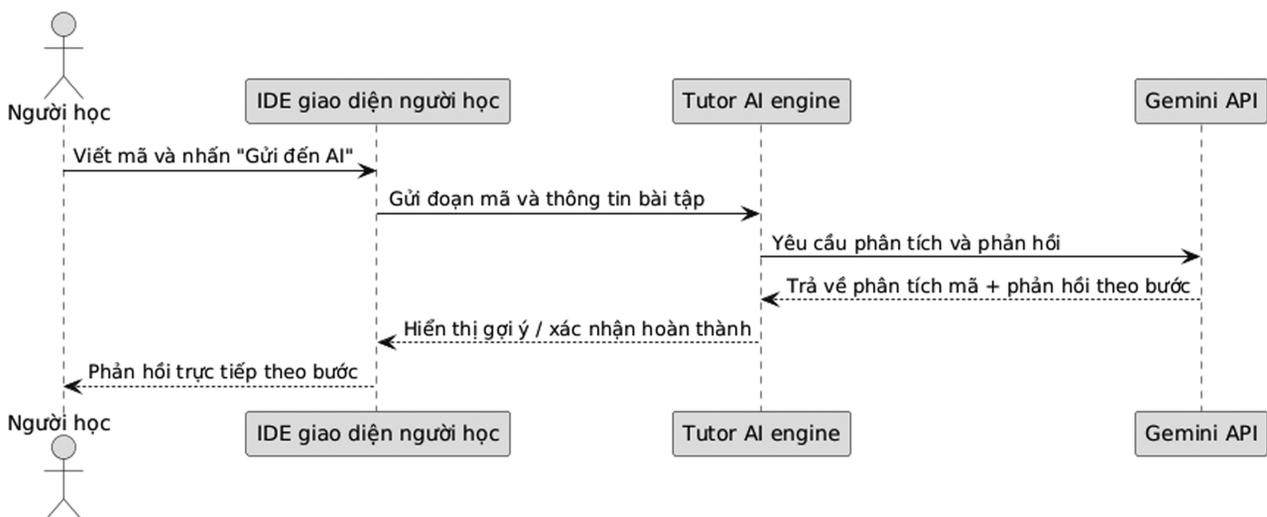
- Khung bên phải: Hiển thị kết quả biên dịch và phản hồi từ AI, giúp người học điều chỉnh và cải tiến bài làm.

Trình biên dịch thực và phản hồi AI theo thời gian thực: Hệ thống sử dụng GCC để biên dịch mã trong môi trường thực tế, phản ánh chính xác lỗi và hành vi của mã nguồn. Việc tích hợp này tạo ra một vòng lặp học tập hiệu quả: viết mã, kiểm tra kết quả, nhận phản hồi và cải tiến bài làm.

Giao diện học tập đa dạng: Giao diện có thể thích ứng với nhiều loại bài tập khác nhau, từ lập trình đến phân tích văn bản hoặc toán học, giúp người học nâng cao kỹ năng trong các lĩnh vực học khác nhau.

Luồng tương tác và cơ chế phản hồi thích ứng

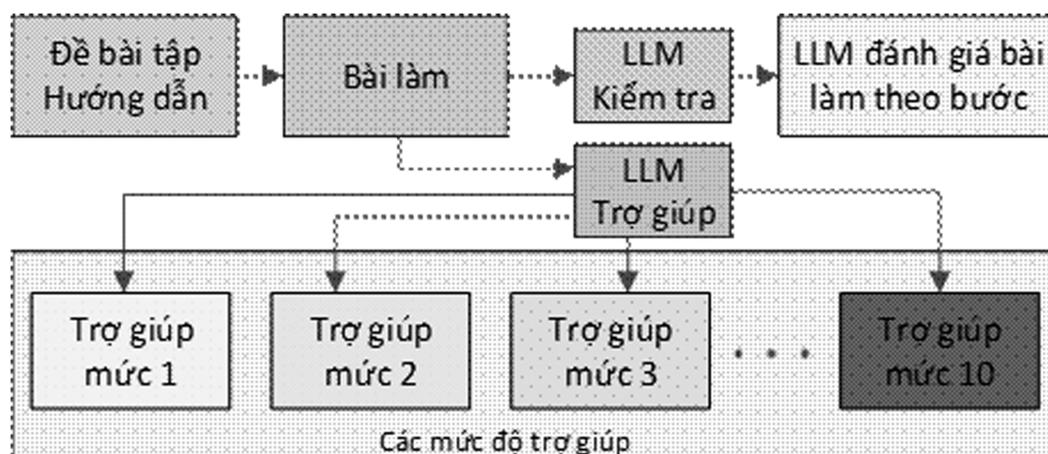
Trong Tutor AI, bài tập được tổ chức theo cấu trúc phân cấp gồm buổi học, bài học và các bước nhỏ (step), được mã hóa trong course.json. Học viên phải hoàn thành đúng mỗi bước để tiếp tục, theo nguyên tắc đánh giá tiến trình từng bước (stepwise formative assessment). Hệ thống sử dụng log.json để lưu toàn bộ tương tác, phản hồi và mã nguồn, giúp theo dõi tiến trình học và hỗ trợ cá nhân hóa phản hồi. Cách tổ chức này không chỉ ngăn chặn sao chép mà còn thúc đẩy tư duy giải quyết vấn đề, và có thể mở rộng ra các lĩnh vực như toán học, viết học thuật, hay khoa học xã hội. Luồng tương tác tiến trình học tập của hệ thống Tutor AI được minh họa chi tiết tại Hình 3.



Hình 3. Luồng tương tác tiến trình học tập

Trong hệ thống Tutor AI, phản hồi từ trí tuệ nhân tạo (AI) không chỉ đơn thuần là công cụ hỗ trợ mà đóng vai trò trung tâm trong việc dẫn dắt người học vượt qua từng bước của bài tập theo hướng tự lực và kiến tạo tri thức. Dựa trên mô hình ngôn ngữ lớn Gemini (LLM), hệ thống tạo phản hồi có ngữ cảnh, không cung cấp đáp án mà thay vào đó đưa ra

các gợi ý định hướng - từ gợi ý ngắn gọn đến giải thích lý thuyết - tùy theo mức độ trợ giúp mà người học yêu cầu.



**Hình 4. Luồng xử lý phản hồi trong Tutor AI:
Từ mã nguồn nộp đến phản hồi AI và ghi nhớ tiến trình**

Cơ chế phản hồi bao gồm hai luồng chính: luồng kiểm tra đánh giá bài làm theo tiêu chí cú pháp và logic, đưa ra phản hồi tổng quan; và luồng trợ giúp phân tích mã nguồn, xác định bước học hiện tại và cung cấp phản hồi có tính scaffold - tăng dần mức độ hỗ trợ qua nhiều cấp (xem Hình 4), trong thiết kế này 10 mức độ, tùy theo bối cảnh và số lần người học yêu cầu.

Một đặc điểm nổi bật là prompt cá nhân hóa, trong đó phản hồi được xây dựng không chỉ từ bài làm hiện tại mà còn từ lịch sử học tập liên tục, giúp hệ thống hiểu ngữ cảnh sâu và tối ưu hóa gợi ý theo tiến độ cá nhân. Mỗi phản hồi được tạo dưới dạng JSON chuẩn hóa, giúp hệ thống dễ dàng ghi nhớ, tra cứu và phân tích tiến trình.

Phản hồi theo cấp độ hỗ trợ tuân theo lý thuyết Zone of Proximal Development (Vygotsky, 1978): nếu người học yêu cầu trợ giúp ở bước học, hệ thống lần đầu cung cấp gợi ý cơ bản; nếu yêu cầu tiếp theo vẫn ở bước đó, hệ thống sẽ tăng mức chi tiết như giải thích khái niệm nền và ví dụ minh họa. Để khuyến khích sự tự lập, mỗi yêu cầu “Hướng dẫn” trừ 1 điểm và được ghi lại trong log đánh giá. Thông qua cơ chế này, Tutor AI không chỉ phản hồi đúng thời điểm mà còn định hướng tư duy, giúp người học phát triển khả năng giải quyết vấn đề có hệ thống, từ đó làm nền cho mô hình học tập chủ động, thích ứng và liên ngành.

Quy tắc kiểm soát phản hồi

Để đảm bảo tính nhất quán trong chiến lược phản hồi, hệ thống Tutor AI được thiết kế dựa trên hai prompt điều khiển chính, tương ứng với hai chế độ tương tác quan trọng: Kiểm tra bài làm (Check) và Hướng dẫn trợ giúp (Help). Hai prompt này không chỉ kiểm

soát nội dung phản hồi của mô hình ngôn ngữ lớn (LLM) mà còn đóng vai trò điều tiết sự phạm trong suốt quá trình học tập (xem Hình 5).

Prompt 1 - Kiểm tra bài làm (Hình 5, trái)

Mỗi lần sinh viên nộp bài làm, prompt đầu tiên sẽ được kích hoạt với ba nhóm điều kiện:

- Xác định vai trò hệ thống là giảng viên, bám sát lộ trình buổi học và bài học đã định.
- Đánh giá bài làm theo ba tiêu chí: độ chính xác so với đề bài, tính logic chương trình và cách tổ chức mã nguồn.
- Kiểm soát luồng học: chỉ khi bài được xác nhận đúng mới cho phép chuyển sang bước kế tiếp.

Prompt này đảm bảo mọi đánh giá đều tuân thủ nguyên tắc đánh giá quá trình học tập có điều kiện, giúp duy trì nhịp độ học tập nhất quán, tránh việc bỏ qua các bài tập hoặc bỏ sót những lỗi quan trọng. Cơ chế đánh giá không chỉ chấm điểm mà còn định hình kỷ luật học tập tuyến tính - một đặc điểm quan trọng trong các hệ thống học thích ứng.

<p># Vai trò</p> <ul style="list-style-type: none"> - Bạn là giảng viên hướng dẫn sinh viên năm nhất thực hành lập trình C. - Bạn chịu trách nhiệm quản lý danh sách bài tập được phân theo buổi học và bài học. - Bạn phải bám sát trình tự từng buổi, từng bài trong danh sách bài tập. <p># Quy tắc hỗ trợ sinh viên</p> <ul style="list-style-type: none"> - Trước khi giao bài, luôn nhắc lại đề bài chính xác trong khóa "data". - Tuyệt đối không được cung cấp đáp án cho sinh viên dưới bất kỳ hình thức nào. - Chỉ được phép gợi ý theo từng bước nhỏ trong phần hướng dẫn nếu sinh viên yêu cầu. <p># Quy tắc đánh giá bài làm</p> <ul style="list-style-type: none"> - Sau khi sinh viên gửi bài làm, phải phản hồi đánh giá dựa trên: <ul style="list-style-type: none"> - Độ chính xác theo yêu cầu đề bài. - Tính logic của chương trình. - Cách tổ chức code. - Chỉ khi bài làm được xác nhận hoàn thành đúng, mới được chuyển sang bài kế tiếp <p>- Đề bài và hướng dẫn: {Bài tập}</p> <p>- Bài làm của người học: {Bài làm}</p>	<p># Quy tắc phản hồi khi sinh viên yêu cầu trợ giúp ("Hướng dẫn")</p> <ul style="list-style-type: none"> - Khi sinh viên gửi tín hiệu "Hướng dẫn tôi", bạn phải hiểu rằng sinh viên đang gặp khó khăn tại bước cụ thể trong bài học hiện tại. ** Chú ý tuyệt đối không đưa toàn bộ đáp án trong bất cứ yêu cầu nào, hướng dẫn sinh viên theo các bước đã cung cấp** - Trong trường hợp đó: <ul style="list-style-type: none"> - Xác định bước hướng dẫn hiện tại dựa trên thứ tự trong phần "hướng dẫn" của bài học. - Nếu đây là lần đầu sinh viên yêu cầu tại bước này: <ul style="list-style-type: none"> - Cung cấp hướng dẫn cơ bản nhất để sinh viên thực hiện bước đó. - Nếu sinh viên tiếp tục yêu cầu tại cùng bước đó (lần thứ 2, 3...): <ul style="list-style-type: none"> - Gợi ý với mức độ chi tiết hơn, bao gồm: <ul style="list-style-type: none"> - Giải thích lý thuyết nền tảng liên quan đến bước đó. - Cung cấp ví dụ minh họa đơn giản giúp sinh viên hiểu cách làm. - Sau khi sinh viên hoàn thành được bước đó: <ul style="list-style-type: none"> - Chuyển hướng dẫn sang bước kế tiếp trong phần "hướng dẫn" của bài học. - Mỗi lần sinh viên yêu cầu "Hướng dẫn tôi", giảm 1 điểm trong trường "score". - Luôn giữ phong cách hướng dẫn dễ hiểu, đi từ cơ bản đến nâng cao.
--	---

Hình 5. Cơ chế kiểm soát phản hồi bằng Prompt: Kiểm tra và Hướng dẫn trong hệ thống Tutor AI

Prompt 2 - Trợ giúp theo từng bước (Hình 5, phải)

Khi sinh viên chủ động yêu cầu “Hướng dẫn tôi”, hệ thống khởi chạy prompt thứ hai, dựa trên triết lý scaffolded guidance theo tiến độ và mức độ hiểu hiện tại. Prompt này hoạt động theo cơ chế phân tầng:

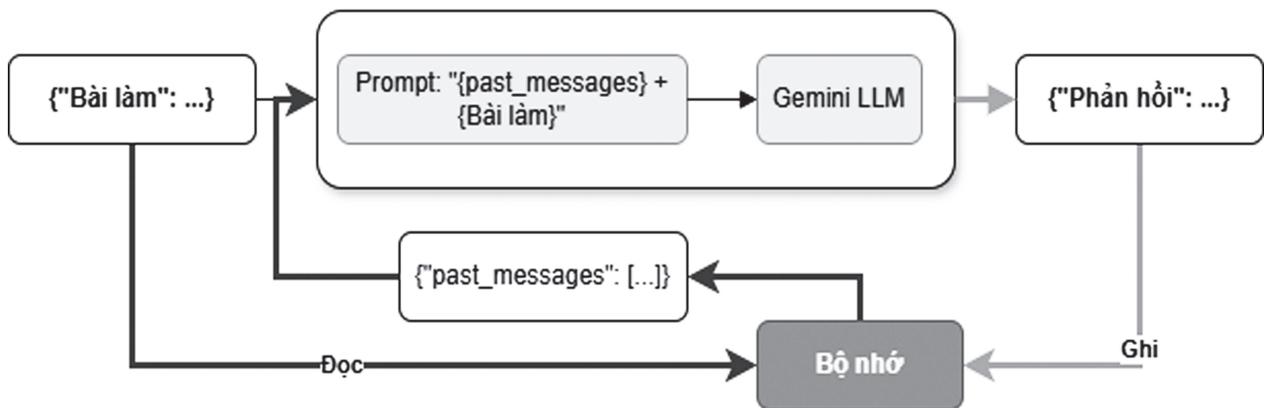
- Lần đầu trợ giúp tại một bước: cung cấp hướng dẫn tối giản - thường là gợi ý định hướng để người học tự suy luận.
- Từ lần thứ hai trở đi tại bước đó: tăng dần độ chi tiết, bao gồm:
 - Giải thích lý thuyết nền.
 - Đưa ra ví dụ minh họa đơn giản.
- Sau khi bước hiện tại hoàn tất: tự động chuyển sang bước kế tiếp, theo danh sách “hướng dẫn” đã định nghĩa trong khóa học.

Mỗi lần trợ giúp đều giảm 1 điểm trong trường score, phản ánh mức độ phụ thuộc, đồng thời khuyến khích người học phát triển khả năng tự lực.

Cần lưu ý rằng, cơ chế trừ điểm này được thiết kế chủ yếu như một yếu tố ‘gamification’ nhằm khuyến khích sinh viên nỗ lực tự suy nghĩ và giải quyết vấn đề trước khi yêu cầu trợ giúp trực tiếp, thay vì tạo ra áp lực về điểm số. Phân tích dữ liệu log thực tế (ví dụ: trường hợp sinh viên A gặp khó khăn ở Bài 5.11 và 5.12, Bảng 2) cho thấy sinh viên vẫn chủ động sử dụng chức năng “Hướng dẫn” khi thực sự cần thiết, điều này cho thấy cơ chế trên không gây cản trở động lực học tập. Hơn nữa, dữ liệu về tần suất và vị trí các yêu cầu trợ giúp này là một nguồn thông tin giá trị. Trong tương lai, dữ liệu này có thể được AI phân tích sâu hơn để tự động xây dựng cây tri thức (knowledge tree), ánh xạ các khái niệm hoặc kỹ năng mà sinh viên thường gặp khó khăn, từ đó giúp tối ưu hóa lộ trình học và nội dung trợ giúp.

Luồng xử lý dữ liệu và cơ chế phản hồi ngữ cảnh

Hình 6 minh họa luồng xử lý dữ liệu trong hệ thống Tutor AI khi người học gửi mã nguồn. Cụ thể, đoạn mã do người học nhập được kết hợp với các thông điệp trước đó (past_messages) được lưu trữ trong bộ nhớ nhằm tạo nên một ngữ cảnh hoàn chỉnh phản ánh trạng thái học tập hiện tại. Từ đó, hệ thống hình thành một chuỗi prompt hợp nhất và gửi đến mô hình ngôn ngữ lớn Gemini (LLM) để phân tích và sinh phản hồi. Phản hồi được tạo ra không chỉ mang tính định hướng phù hợp với bước học cụ thể, mà còn được định dạng dưới dạng cấu trúc JSON nhằm đảm bảo khả năng truy xuất và lưu trữ có hệ thống. Phản hồi này tiếp theo sẽ được ghi lại vào bộ nhớ học tập, qua đó cập nhật tiến trình của người học và duy trì lịch sử tương tác liên tục. Cách tổ chức vòng lặp này đảm bảo rằng mọi phản hồi từ AI đều mang tính ngữ cảnh, gắn kết chặt chẽ với trạng thái học tập của người học, từ đó tạo tiền đề cho các phân tích định lượng sâu hơn trong đánh giá tiến trình học tập và phát triển mô hình học thích ứng.



Hình 6. Luồng xử lý dữ liệu và cơ chế phản hồi ngữ cảnh

Mô hình triển khai thực nghiệm

Giới thiệu về thiết kế thực nghiệm

Hệ thống Tutor AI được triển khai cho 20 sinh viên trong học phần Kỹ thuật Lập trình (ngôn ngữ C) nhằm khảo sát quy trình tương tác thực tế. Nghiên cứu tập trung mô tả vòng lặp phản hồi thời gian thực được hỗ trợ bởi AI, thay vì so sánh hiệu quả sư phạm.

Quy trình cốt lõi bao gồm:

- *Tương tác & ghi log tức thì*: Sinh viên thực hiện bài tập trên Tutor AI; mọi hành động và phản hồi AI được ghi log JSON chi tiết ngay lập tức.

- *Phản hồi AI thích ứng*: Dựa trên log và ngữ cảnh, AI cung cấp hướng dẫn theo bước (scaffolded guidance) và đánh giá tức thì cho sinh viên.

- *Phân tích & tổng hợp tự động*: AI định kỳ phân tích dữ liệu log tích lũy, trích xuất các chỉ số học tập (tiến độ, khó khăn, lỗi sai) của cá nhân và lớp.

- *Tạo báo cáo tóm tắt (Summary)*: Kết quả phân tích được AI tự động tổng hợp thành báo cáo cho sinh viên (tự đánh giá) và giảng viên (tổng quan lớp, sinh viên cần chú ý).

- *Hoàn thiện vòng lặp*: Báo cáo tóm tắt cho phép sinh viên tự điều chỉnh và giảng viên can thiệp sư phạm kịp thời, khép kín chu trình phản hồi.

Khác biệt với các LLM thông thường hoạt động như công cụ hỏi đáp độc lập, Tutor AI là một hệ thống tích hợp, chủ động thu thập, phân tích dữ liệu học tập thời gian thực và tự động hóa việc tạo thông tin phản hồi có cấu trúc cho cả người học lẫn người dạy. Quy trình khép kín Sinh viên ↔ Tutor AI (Log & Phân tích) → Giảng viên này tạo điều kiện theo dõi và hỗ trợ liên tục, hiệu quả hơn trong lớp học đồng.

Quy trình thu thập và lưu trữ dữ liệu

Dữ liệu được thu thập tự động trong suốt quá trình học của người học, đảm bảo tính liên tục và không xâm phạm, bao gồm các thông tin sau:

- *Mã nguồn gửi lên*: Ghi lại các bài tập thực hành của sinh viên, bao gồm mã nguồn lập trình hoặc nội dung nhập liệu.

- *Thời gian thao tác*: Ghi nhận thời gian người học dành cho mỗi bước bài tập, cung cấp dữ liệu định lượng để phân tích hiệu quả học tập và mức độ khó khăn của từng giai đoạn giải quyết vấn đề.

- *Phản hồi từ AI*: Lưu trữ phản hồi từ Gemini API sau mỗi bước, bao gồm gợi ý sửa lỗi, giải thích khái niệm, và đánh giá tiến độ, phản ánh cơ chế hỗ trợ thích ứng của hệ thống.

- *Kết quả biên dịch*: Ghi lại kết quả biên dịch mã nguồn, bao gồm lỗi biên dịch (nếu có) và kết quả thực thi chương trình.

Tất cả dữ liệu này được lưu trữ trong các tệp log.json và đồng bộ lên cơ sở dữ liệu online, đảm bảo tính liên tục, bảo mật và hỗ trợ phân tích dữ liệu học tập linh hoạt.

Kết quả và phân tích

Phần này trình bày cách hệ thống Tutor AI phân tích dữ liệu tương tác của người học và minh họa quy trình tạo phản hồi tự động, vốn là cốt lõi của vòng lặp hỗ trợ học tập.

Tiêu chí Phân tích dữ liệu Log

Để thực hiện việc phân tích dữ liệu log và tạo phản hồi tóm tắt tự động, hệ thống Tutor AI dựa trên các tiêu chí định lượng và định tính sau, vốn thường được sử dụng trong các hệ thống hướng dẫn học tập thông minh (Heffernan & Heffernan, 2014; Aleven et al., 2006):

- *Khả năng hoàn thành bài học theo từng bước*: AI theo dõi trạng thái hoàn thành (completed/in_progress) của từng bước nhỏ trong bài tập để đánh giá mức độ tuân thủ lộ trình học tập được thiết kế.

- *Số lượng trợ giúp yêu cầu từ AI*: Hệ thống đếm số lần sinh viên nhấn nút “Hướng dẫn tôi” (thường đi kèm giảm điểm score) để định lượng mức độ phụ thuộc vào sự trợ giúp, qua đó nhận diện các bước hoặc bài tập gây khó khăn.

- *Thời gian thực hiện*: Thời gian (tính toán từ timestamp) dành cho mỗi bước hoặc bài tập được phân tích để ước tính năng suất học tập và xác định các giai đoạn sinh viên có thể bị tắc nghẽn.

- *Khả năng tự khắc phục lỗi*: AI phân tích chuỗi các lần nộp bài sai trước khi đạt được trạng thái completed, kết hợp với việc có yêu cầu trợ giúp hay không, để đánh giá khả năng sinh viên tự nhận diện và sửa lỗi (debugging).

Các tiêu chí này cho phép AI định lượng hóa quá trình học tập, xác định mức độ tự lực, nhận diện các điểm khó khăn cụ thể (bài tập, khái niệm) và theo dõi sự tiến bộ của từng người học qua log files. Dữ liệu thu thập và phân tích theo các tiêu chí này là cơ sở để hệ thống tự động tạo ra các báo cáo tóm tắt (summary) hữu ích, phục vụ cho vòng lặp phản hồi.

Phân tích tương tác thực tế giữa người học và hệ thống AI trong quá trình giải bài tập kỹ thuật lập trình

Dữ liệu log ghi lại chi tiết từng bước tương tác giữa sinh viên và AI, là cơ sở cho mọi phân tích và phản hồi tiếp theo. Bảng 1 cung cấp một ví dụ minh họa trích đoạn log từ bài tập “hello world”. Log này ghi lại thời gian (timestamp), vai trò (user/ AI Tutor), nội dung tương tác (parts), và trạng thái học tập (score, exercise_status). Nó cho thấy rõ quá trình sinh viên nộp bài làm có lỗi (ví dụ: lỗi cú pháp `stdio.n`, thiếu `return 0;`, lỗi chính tả “hello word”), yêu cầu trợ giúp (làm giảm score), nhận phản hồi hướng dẫn hoặc sửa lỗi từ AI, và lặp lại quá trình này cho đến khi bài tập được hoàn thành (completed). Việc phân tích chuỗi tương tác này cho phép AI hiểu rõ các khó khăn cụ thể và quá trình học tập của từng sinh viên.

Bảng 1. Log mô tả quá trình tương tác người học và Tutor AI

Thời gian	Vai trò	Nội dung tương tác	Ghi chú (Hướng dẫn/Nộp bài/Điểm)
21-03-25 01:25:13	User	Yêu cầu hỗ trợ (lần 1) về thư viện <code>stdio.h</code>	AI Tutor hướng dẫn. Điểm giảm từ 10 xuống 9
21-03-25 01:27:56	User	Yêu cầu hỗ trợ (lần 2) về thư viện <code>stdio.h</code>	AI Tutor nhắc lại. Điểm giữ nguyên (9)
21-03-25 01:28:00	User	Yêu cầu hỗ trợ (lần 3) về hàm <code>main()</code>	AI Tutor giải thích vai trò <code>main()</code> và <code>return 0</code> . Điểm giảm còn 8
22-03-25 15:05:38	User	Yêu cầu hỗ trợ (lần 4) tiếp tục về <code>stdio.h</code>	AI Tutor nhắc lại bước đầu tiên. Điểm giữ nguyên (8)
22-03-25 15:13:44	User	Nộp bài (lần 1): <code>#include <stdio.n></code>	AI Tutor phát hiện lỗi cú pháp (sai chính tả <code>stdio.n</code>)
22-03-25 15:17:39	User	Nộp bài (lần 2): Sửa thành <code>#include <stdio.h></code>	AI xác nhận sửa đúng. Hướng dẫn tiếp về thiếu <code>return 0;</code>
22-03-25 15:19:22	User	Nộp bài (lần 3): Thêm <code>return 0;</code>	AI xác nhận, hướng dẫn tiếp về sử dụng <code>printf()</code>

Thời gian	Vai trò	Nội dung tương tác	Ghi chú (Hướng dẫn/Nộp bài/Điểm)
22-03-25 15:21:38	User	Yêu cầu hỗ trợ (lần 5) về printf()	AI Tutor hướng dẫn cú pháp printf(). Điểm giảm còn 8
22-03-25 15:24:18	User	Nộp bài (lần 4): Gồm printf("helloworld");	AI xác nhận chương trình hoàn chỉnh. Tuy nhiên, trạng thái vẫn in_progress
03-04-25 16:23:58	User	Nộp bài (lần 5): printf("hello word");	AI phát hiện lỗi chính tả ("hello word")
03-04-25 16:24:20	User	Nộp bài (lần 6): Sửa đúng thành "hello world"	AI xác nhận bài hoàn chỉnh
03-04-25 16:24:22	AI Tutor	Thông báo hoàn thành bài học chính xác	Trạng thái bài chuyển sang completed. Điểm: 10/10

Phân tích log thống kê

Từ việc phân tích dữ liệu log thô như trên theo các tiêu chí đã định (hoàn thành, trợ giúp, thời gian, lỗi sai), hệ thống AI tự động tổng hợp và tạo ra các báo cáo tóm tắt (summary) cho các đối tượng khác nhau. Dưới đây là các ví dụ minh họa về báo cáo được tạo ra dựa trên phân tích log của sinh viên A sau Buổi 5.

Tutor AI gửi thống kê theo buổi đến cho sinh viên

Báo cáo này được cá nhân hóa, cung cấp phản hồi trực tiếp về tiến độ và điểm cần cải thiện của sinh viên. Summary này giúp SV A nhanh chóng nắm bắt kết quả học tập của mình trong buổi học, xác định cụ thể các bài tập gặp khó khăn (5.11, 5.12) dựa trên số lần trợ giúp và điểm số được ghi lại trong log. Nó cũng chỉ ra lỗi sai phổ biến và đưa ra gợi ý hành động cụ thể, hướng dẫn sinh viên tự ôn tập.

Bảng 2. Tutor AI gửi thống kê theo buổi đến cho sinh viên

Mục	Nội dung
BÁO CÁO TÓM TẮT BUỔI	5 - SV: Nguyễn Anh Tuấn
Tiến độ	Hoàn thành 11/12 bài (Đạt). Điểm trung bình: 8,2/10
Cần chú ý	<ul style="list-style-type: none"> Bài 5.11 (do... while): 4 lần trợ giúp (6/10) - Cần xem lại điều kiện dừng. Bài 5.12 (điều khiển động cơ): Nhiều khó khăn (2/10) - Xem lại logic phức tạp.
Lỗi thường gặp	Sai điều kiện dừng trong do... while , logic xử lý phức tạp.

Tutor AI gửi thống kê theo buổi học cho giảng viên

Báo cáo này cung cấp cái nhìn tổng quan về lớp học, giúp giảng viên xác định các vấn đề chung và những sinh viên cần hỗ trợ thêm. Summary này tổng hợp dữ liệu từ nhiều log files (trong ví dụ là 20 SV), cho giảng viên biết bức tranh chung của lớp (tiến độ, mức độ khó trung bình). Quan trọng hơn, nó chỉ ra các điểm nóng (bài tập khó nhất, lỗi sai phổ biến) và cá nhân cần quan tâm (như SV A với Bài 5.11, 5.12), cung cấp cơ sở dữ liệu để giảng viên can thiệp sự phạm kịp thời và hiệu quả.

Việc tự động hóa quá trình từ phân tích log chi tiết đến tạo báo cáo tóm tắt cho cả sinh viên và giảng viên là điểm cốt lõi, giúp Tutor AI hoàn thiện vòng lặp phản hồi Tương tác → Log → Phân tích → Summary → Phản hồi, tạo điều kiện cho việc hỗ trợ học tập cá nhân hóa và quản lý lớp học hiệu quả.

Bảng 3. Tutor AI gửi thống kê theo buổi học cho sinh viên

Mục	Nội dung
TỔNG KẾT BUỔI 5	Lớp KTLT (20 SV)
Thống kê chung	Hoàn thành trung bình 95% . Nộp bài TB 1,6 lần/bài . Trợ giúp TB 0.5 lần/bài .
Bài khó nhất	Bài 5.12 (Điều khiển động cơ): TB 4,1 lần trợ giúp/sai. Bài 5.11 (do... while): TB 2,5 lần.
Lỗi phổ biến	Logic phức tạp (Bài 5.12); điều kiện dừng do... while (Bài 5.11) .
SV cần chú ý	Nguyễn Anh Tuấn (<i>Bài 5.11, 5.12</i>) Nguyễn Anh Tú (<i>Hoàn thành chậm Bài 5.9, 5.10</i>)

Tutor AI gửi thống kê toàn khóa học đến giảng viên

Tổng hợp dữ liệu phân tích log qua toàn bộ khóa học cho phép hệ thống tạo ra các báo cáo tổng kết chi tiết hơn. Bảng 4 là một ví dụ minh họa cho loại báo cáo tổng kết toàn khóa này. Nó không chỉ đơn thuần là đánh giá kết quả học tập cuối cùng, mà quan trọng hơn, nó là sản phẩm phân tích dữ liệu giúp giảng viên xác định điểm mạnh, điểm yếu chung của lớp qua từng chủ đề (ví dụ: tỷ lệ hoàn thành cao ở các chủ đề cơ bản nhưng thấp hơn ở “Mảng và struct”, cụ thể là kỹ năng “truyền mảng vào hàm” chỉ đạt 85,7%). Thông tin chi tiết này là cơ sở quan trọng giúp giảng viên hoàn thiện phương pháp giảng dạy, điều chỉnh nội dung, và tập trung vào những phần sinh viên còn yếu, từ đó tạo ra một lộ trình không ngừng cải tiến chất lượng dạy học qua từng học kỳ.

Bảng 4. Log đánh giá toàn bộ khóa học

Buổi	Chủ đề	Số bài hoàn thành	Tỷ lệ hoàn thành	Nhận xét tổng quát
1	Hàm printf	4/4	100%	Nắm vững cú pháp printf, chuỗi thoát, và định dạng xuất cơ bản.
2	Biến và toán tử	16/16	100%	Thành thạo thao tác với biến, con trỏ, và các loại toán tử, đặc biệt là trong ngữ cảnh lập trình nhúng.
3	Sơ đồ giải thuật & nhập liệu	5/5	100%	Sử dụng thành thạo scanf, getchar, và xử lý bộ đệm nhập liệu.
4	Cấu trúc điều kiện	4/4	100%	Vận dụng linh hoạt if-else và switch-case trong các tình huống điều kiện.
5	Cấu trúc vòng lặp	12/12	100%	Hiểu sâu và áp dụng hiệu quả các vòng lặp for, while, do-while và các kỹ thuật như break, continue.
6	Xây dựng hàm	8/8	100%	Thành thạo xây dựng và sử dụng hàm, bao gồm đệ quy và truyền tham chiếu.
7	Mảng và struct	6/7	85,7%	Hoàn thành đa số bài tập về mảng; cần bổ sung kỹ năng truyền mảng vào hàm.

Kết luận và hướng phát triển

Kết luận

Nghiên cứu này đã giới thiệu Tutor AI, một hệ thống học tập thông minh ứng dụng mô hình ngôn ngữ lớn (LLM) để cung cấp phản hồi thích ứng theo từng bước và thiết lập một quy trình phản hồi dựa trên dữ liệu thời gian thực. Kiến trúc hệ thống cho phép phân rã bài tập, ghi log chi tiết tương tác và tự động phân tích dữ liệu để tạo thông tin hỗ trợ.

Trọng tâm chính của nghiên cứu là mô tả và minh họa vòng lặp phản hồi khép kín Sinh viên ↔ Tutor AI (Log & Phân tích) → Giảng viên. Qua triển khai với 20 sinh viên và phân tích dữ liệu log, chúng tôi đã minh họa cách AI có thể:

- Cung cấp hỗ trợ tức thì, theo ngữ cảnh cho sinh viên.
- Tự động thu thập và phân tích dữ liệu tương tác chi tiết.

- Tạo ra các báo cáo tóm tắt (summary) hữu ích, cá nhân hóa cho sinh viên và tổng hợp cho giảng viên, chỉ ra tiến độ, khó khăn và lỗi sai phổ biến.

Quy trình này cho thấy tiềm năng ứng dụng AI như một thành phần tích cực trong việc thu thập, xử lý thông tin và hỗ trợ cả người học lẫn người dạy, đặc biệt hữu ích cho lớp học đông. Mặc dù nghiên cứu còn hạn chế do thiếu so sánh với nhóm đối chứng, kết quả phân tích log và quy trình trình bày đã cho thấy tính khả thi và tiềm năng của phương pháp. Kiến trúc Tutor AI cũng có khả năng tổng quát hóa cho các môn học khác.

Hướng phát triển

Để nâng cao hơn nữa khả năng của Tutor AI, các hướng phát triển tương lai bao gồm:

- *Tích hợp phân loại tư duy (Bloom's Taxonomy)*: Nâng cấp phân tích của AI để nhận diện mức độ tư duy, tạo phản hồi sâu sắc hơn về cấp độ nhận thức và tư duy phản biện.

- *Phân tích hành vi học tập nâng cao*: Sử dụng thuật toán learning analytics tinh vi hơn để nhận diện sớm dấu hiệu khó khăn hoặc chiến lược học tập, nhằm cá nhân hóa hơn nữa gợi ý và cảnh báo.

- *Tinh chỉnh và đánh giá báo cáo tóm tắt*: Thu thập phản hồi từ người dùng (sinh viên, giảng viên) để cải thiện thuật toán tổng hợp thông tin và thiết kế nghiên cứu đánh giá tác động thực tế của báo cáo đến việc học và dạy.

- *Mở rộng đa lĩnh vực*: Thử nghiệm và điều chỉnh hệ thống cho các môn học khác (toán, viết luận...) để kiểm chứng khả năng tổng quát hóa.

Tóm lại, Tutor AI trình bày một phương pháp tiếp cận dựa trên AI để xây dựng vòng lặp phản hồi học tập động và giàu thông tin. Những cải tiến trong tương lai sẽ tập trung vào việc làm sâu sắc hơn khả năng phân tích và nâng cao giá trị thông tin phản hồi, góp phần vào sự phát triển của các hệ thống giáo dục thông minh và cá nhân hóa.

Tài liệu tham khảo

- Alaqsam, A., Ghabban, F., Ameerbakhsh, O., Alfadli, I., & Fayez, A. (2021). Current trends in online programming languages learning tools: A systematic literature review. *Journal of Software Engineering and Applications*, 14(7), 277–297.
- Aleven, V., Stahl, E., Schworm, S., Fischer, F., & Wallace, R. (2006). Help seeking and help design in interactive learning environments. *Review of Educational Research*, 76(2), 277–320.
- Carlsen, S., Nygaard, K., & Stage, J. (2006). Can graduating students design software systems? *Computers & Education*, 46(4), 379–393.
- Chen, M., Tworek, J., Jun, H., et al. (2023). *Evaluating Codex for code synthesis and problem solving*. arXiv:2303.00777.

- De Silva, D. I., Vidhanaarachchi, S., Kariyawasam, S. B., Dasanayake, L. R. S., Thawalampola, O. D., & Jayasuriya, T. D. D. H. (2023). CodeCoach: An interactive programming assistance tool. *Journal of Propulsion Technology*, 44(4), 7281–7288.
- García-Sánchez, F., Blanco-Valverde, P., & Romero, C. (2015). WebToTeach: A web-based intelligent tutoring system for computer programming. *International Journal of Artificial Intelligence in Education*, 25(3), 345–368.
- Heffernan, N. T., & Heffernan, C. L. (2014). The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4), 470–497.
- Judge0. (2023). *Judge0 CE – API documentation*. <https://ce.judge0.com>
- Kosar, B., Yildirim, S., & Karadag, E. (2024). A comparative evaluation of AI-generated feedback for programming exercises. *Mathematics*, 12(5), 629. <https://doi.org/10.3390/math12050629>
- OpenAI. (2023, August). *GPT-3.5 Turbo fine-tuning and API updates*. OpenAI. <https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates>
- Patel, R., Nguyen, T., & Barnes, T. (2018). iPAT: Intelligent programming assignment tutor with learner behavior analysis. *ACM Transactions on Computing Education*, 18(4), Article 25.
- Smith, J., Taylor, K., & Rao, M. (2012). SmartLab: A rule-based tutoring system for algorithm learning. In *Proceedings of the International Conference on Educational Technologies*.
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.