



ĐỀ CƯƠNG THỰC TẬP DỰ ÁN TỐT NGHIỆP
Xây dựng hệ thống chấm bài tích hợp AI
(AI-Assisted Grading System)

1 THÔNG TIN CHUNG

Người hướng dẫn:

– TS. Nguyễn Minh Huy (Khoa Công nghệ Thông tin)

[Nhóm] Sinh viên thực hiện:

1. Nguyễn Anh Hoàng (MSSV: 21127284)
2. Nguyễn Hoàng Khiêm (MSSV: 21127317)
3. Nguyễn Khắc Nhật Minh (MSSV: 21127357)
4. Phạm Văn Minh (MSSV: 21127360)
5. Đỗ Hoàng Khánh Duy (MSSV: 21127601)

Loại đề tài: Ứng dụng

Thời gian thực hiện: Từ 01/2025 đến 07/2025

2 NỘI DUNG THỰC HIỆN

2.1 Giới thiệu về đề tài

Trong bối cảnh giáo dục hiện nay, giáo viên và giảng viên thường phải đối mặt với khối lượng công việc chấm điểm nặng nề. Việc đánh giá thủ công các bài luận, dự án và bài tập lập trình không chỉ tốn nhiều thời gian mà còn dễ dẫn đến sự thiếu nhất quán, đặc biệt là khi phải quản lý những lớp học đông sinh viên hoặc nhiều lớp học cùng lúc. Đồng thời, các cơ sở giáo dục cũng đang tìm kiếm những giải pháp sáng tạo, ứng dụng công nghệ để nâng cao chất lượng giảng dạy và cải thiện hiệu quả của các công việc hành chính.

Hệ thống chấm điểm hỗ trợ AI được tạo ra nhằm giải quyết những thách thức này bằng cách ứng dụng Mô hình Ngôn ngữ Lớn (LLMs) để đẩy nhanh và chuẩn hóa quy trình chấm điểm. Hệ thống cho phép giáo viên tạo và tinh chỉnh các bảng tiêu chí chấm điểm (rubric), tải lên các định dạng bài tập đa dạng và nhận được điểm số do AI đề xuất. Bằng cách giảm đáng kể khối lượng công việc chấm điểm thủ công, nền tảng tạo điều kiện cho giáo viên tập trung nhiều hơn vào các hoạt động giá trị cao, chẳng hạn như phản hồi cá nhân và cố vấn một - một, từ đó nâng cao chất lượng đào tạo.

2.2 Mục tiêu đề tài

2.2.1 Động lực thực hiện

Việc chấm điểm thủ công tốn nhiều thời gian và đôi khi có thể dẫn đến sai sót. Thực tế này ảnh hưởng không nhỏ đến chất lượng giảng dạy và sự hài lòng của học sinh/sinh viên. Vì vậy, việc phát triển một hệ thống chấm điểm tự động với sự hỗ trợ của AI sẽ giúp giảm thiểu gánh nặng cho giáo viên và nâng cao hiệu quả giáo dục.

2.2.2 Lợi ích của đề tài

Một hệ thống chấm điểm tự động tích hợp AI giúp giảm thiểu thời gian chấm điểm và đảm bảo tính chính xác trong đánh giá. Hệ thống sẽ giúp giáo viên dễ dàng chấm điểm và cung cấp phản hồi, từ đó nâng cao chất lượng giảng dạy và kết quả học tập của sinh viên.

2.2.3 Ảnh hưởng và ý nghĩa có thể có của kết quả đối với vấn đề đã được đặt ra nói riêng và toàn bộ hướng nghiên cứu nói chung

Kết quả từ hệ thống chấm điểm AI sẽ có tác động lớn đến quy trình giáo dục, giúp giáo viên và sinh viên tiết kiệm thời gian, cải thiện sự chính xác trong đánh giá, và tạo nền tảng cho việc tích hợp các công nghệ AI khác vào trong giáo dục.

2.3 Phạm vi của đề tài

Hệ thống chấm bài tích hợp AI sử dụng Mô hình Ngôn ngữ Lớn (LLMs) để tự động hóa và chuẩn hóa quá trình chấm điểm trong giáo dục. Bằng cách giảm khối lượng công việc thủ công, hệ thống giúp giáo viên tập trung vào phản hồi cá nhân và cố vấn chuyên sâu, góp phần nâng cao chất lượng giảng dạy.

2.3.1 Các tính năng hệ thống cung cấp

- **Tính năng Tạo Rubric:** Cho phép giáo viên tạo hoặc phát sinh thang điểm (rubric) gồm các tiêu chí chấm điểm (criterion) thông qua việc sử dụng LLM. Giáo viên có thể chỉ định các tiêu chí, trọng số và mô tả mức độ hoàn thành cho từng tiêu chí.
- **Tính năng Chấm điểm bằng sự hỗ trợ của AI:** Cho phép hệ thống sử dụng LLM để tạo ra điểm số ban đầu hoặc điểm số chi tiết cho từng tiêu chí trong rubric. Giáo viên có thể tải lên các bài tập ở nhiều định dạng khác nhau (ví dụ: văn bản, PDF, tệp mã).
- **Tính năng Chỉnh sửa điểm thủ công:** Cung cấp một bảng điều khiển cho

phép giáo viên xem xét điểm số do AI tạo ra, điều chỉnh điểm số và cập nhật các tham số của rubric. Hệ thống cũng theo dõi các thay đổi để phục vụ mục đích kiểm tra và cải tiến lặp lại.

- **Tính năng Báo cáo và phản hồi:** Cung cấp một cái nhìn tổng quan về các bài tập, hiệu suất của sinh viên và sự phù hợp giữa điểm số do AI và giáo viên xác nhận. Hệ thống cũng cho phép xuất điểm số cuối cùng và phản hồi ngắn gọn do AI tạo ra sang các định dạng tiêu chuẩn (ví dụ: CSV, PDF).
- **Tính năng Tùy chỉnh công cụ chấm bài theo dạng Plugin:** Cung cấp các công cụ chuyên biệt để hỗ trợ chấm điểm chi tiết cho từng loại bài tập, đảm bảo đánh giá chính xác và toàn diện.

2.3.2 Thực thể liên quan

- **Giáo viên và trợ giảng:** Là những người dùng chính, trực tiếp tương tác với hệ thống.
- **Sinh viên:** Là đối tượng gián tiếp liên quan, những người sẽ được chấm điểm bài tập và nhận phản hồi từ hệ thống.
- **Rubric chấm điểm:** Là bộ tiêu chí chi tiết được sử dụng để đánh giá chất lượng bài làm của sinh viên. Hệ thống sẽ hỗ trợ tạo và tùy chỉnh rubric.
- **Bài tập của sinh viên:** Bao gồm các tệp văn bản, tài liệu PDF và tệp mã nguồn mà sinh viên nộp để được chấm điểm.
- **Kết quả chấm điểm của AI:** Điểm số và các nhận xét ban đầu được tạo ra bởi hệ thống dựa trên việc phân tích bài tập và so sánh với rubric.
- **Điểm số cuối cùng:** Điểm số sau khi đã được giáo viên xem xét, có thể điều chỉnh hoặc ghi đè lên điểm số của AI.
- **Phản hồi cho sinh viên:** Các nhận xét và góp ý được cung cấp cho sinh viên về bài làm của họ, có thể được tạo ra bởi AI hoặc được giáo viên tinh chỉnh.

- Các mô hình ngôn ngữ lớn (LLM): Các mô hình AI tiên tiến được sử dụng để phân tích ngôn ngữ tự nhiên và mã nguồn, phục vụ cho việc tạo rubric và chấm điểm.
- Các trường hợp kiểm thử: Các bộ kiểm thử được sử dụng để đánh giá tính đúng đắn và chức năng của mã nguồn trong các bài tập lập trình.

2.3.3 Tập dữ liệu

- Dữ liệu về các bài tập của sinh viên đã được chấm điểm thủ công bởi giáo viên. Tập dữ liệu này sẽ đóng vai trò quan trọng trong việc so sánh và đánh giá độ chính xác của hệ thống chấm điểm AI.
- Dữ liệu phản hồi từ giáo viên và sinh viên thu thập được trong quá trình thử nghiệm hệ thống. Những phản hồi này sẽ cung cấp thông tin về trải nghiệm người dùng, các vấn đề cần cải thiện và hiệu quả tổng thể của hệ thống.

2.3.4 Các giới hạn và ràng buộc của đề tài

- Phạm vi của đề tài trong giai đoạn phát triển ban đầu sẽ tập trung vào việc hỗ trợ các loại bài tập phổ biến như văn bản (tiểu luận, báo cáo), tài liệu PDF và các bài tập mã nguồn thuộc một số ngôn ngữ lập trình thông dụng. Các loại bài tập khác như bài tập trắc nghiệm, bài tập điền vào chỗ trống có thể được xem xét trong các giai đoạn phát triển tiếp theo.
- Đề tài sẽ sử dụng các mô hình LLM hiện có (ví dụ: các mô hình từ OpenAI, Google AI, hoặc các mô hình mã nguồn mở). Hiệu suất và độ chính xác của hệ thống sẽ bị ảnh hưởng bởi khả năng và giới hạn của các mô hình này.
- Thời gian và nguồn lực (bao gồm nhân lực và ngân sách) dành cho dự án có thể đặt ra những ràng buộc nhất định đối với phạm vi và mức độ phức tạp của các tính năng có thể được phát triển và thử nghiệm.
- Cần xem xét và tuân thủ các yêu cầu về tính bảo mật và quyền riêng tư của

dữ liệu liên quan đến bài tập của sinh viên và kết quả chấm điểm. Các biện pháp bảo vệ dữ liệu sẽ cần được tích hợp vào thiết kế và phát triển hệ thống.

2.4 Cách tiếp cận dự kiến

Đề tài này sẽ theo đuổi cách tiếp cận kết hợp giữa nghiên cứu lý thuyết về ứng dụng của LLM trong giáo dục và phát triển một hệ thống phần mềm thực tế để minh họa và đánh giá các phương pháp đã nghiên cứu. Chúng tôi dự kiến sẽ khám phá các mô hình LLM hiện đại và các kỹ thuật tương tác (prompt engineering) hiệu quả để giải quyết bài toán chấm điểm tự động cho nhiều loại hình bài tập khác nhau.

2.4.1 Nghiên cứu về việc áp dụng AI để chấm bài

- **Về phương pháp prompt tạo rubric**

Việc tạo ra một rubric đánh giá hiệu quả đòi hỏi sự cân nhắc về loại hình bài tập, trình độ người học, cấu trúc rubric, tiêu chí đánh giá, sự phù hợp với các chuẩn đánh giá, và giọng văn sử dụng. Dựa trên các phương pháp tốt nhất trong thiết kế rubric được gợi ý trong bài viết "Rubric Best Practices, Examples, and Templates" [1], một bộ câu hỏi đã được phân loại để hướng dẫn người hướng dẫn hoặc nhà thiết kế giảng dạy trong việc đưa ra yêu cầu (prompt) cho AI nhằm tạo ra các rubric chi tiết và chất lượng cao. Các câu hỏi này được nhóm thành các loại chính để định hướng AI:

- Loại hình bài tập/nhiệm vụ
- Trình độ học vấn
- Cấu trúc Rubric (Định dạng và tính điểm)
- Mục tiêu học tập và Tiêu chí đánh giá
- Sự phù hợp với Chuẩn đánh giá
- Giọng văn và Mức độ chi tiết

- **Thử nghiệm độ chính xác của AI khi dùng để chấm điểm**

Thông qua thử nghiệm thực tế thì có các nhận xét như sau:

- Nhìn chung khi chấm bằng rubric thì AI cho ra kết quả khá ổn định, trong khả năng chấp nhận được
- Nội dung chấm càng dài thì càng ổn định
- Đối với các tiêu chí không liên quan/không khả dụng với bài làm thì AI thường có xu hướng random chứ không cho điểm ở mức thấp nhất.

2.5 Phương pháp và cách tiếp cận

- **Phương pháp tạo và tùy chỉnh Rubric:**

- Sử dụng LLM để tự động đề xuất các thành phần chính của rubric dựa trên mô tả bài tập và mục tiêu học tập.
- Cung cấp giao diện trực quan cho phép giáo viên chỉnh sửa, bổ sung, điều chỉnh tiêu chí và trọng số.
- Xây dựng thư viện mẫu rubric sẵn có cho nhiều loại bài tập, giúp giáo viên tiết kiệm thời gian.

- **Phương pháp chấm điểm có sự hỗ trợ của AI:**

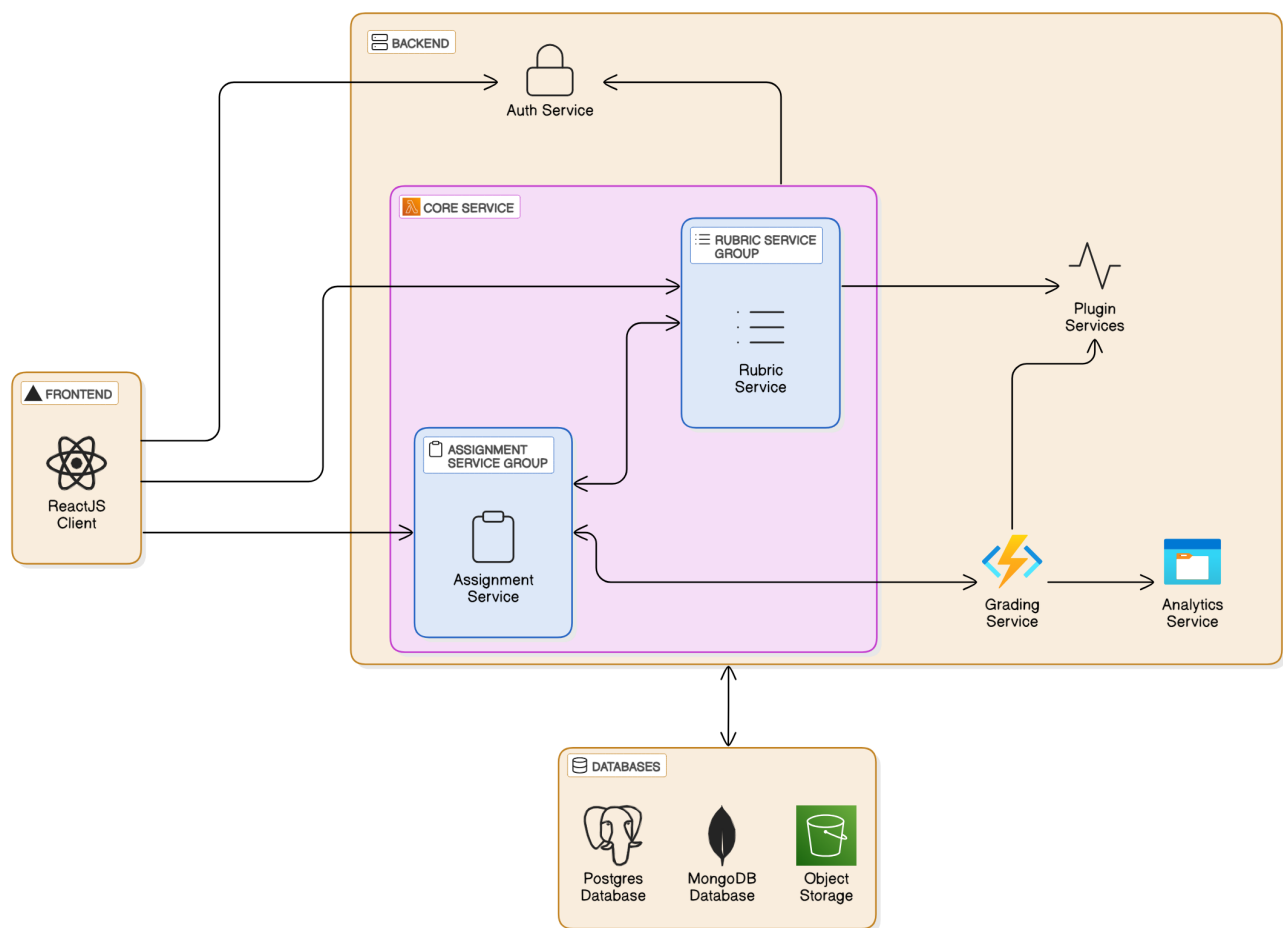
- Nghiên cứu và áp dụng kỹ thuật "prompt engineering" để hướng dẫn LLM phân tích và đánh giá bài tập dựa trên rubric.
- Sử dụng LLM để đánh giá các bài tập văn bản và PDF theo các tiêu chí như nội dung, cấu trúc, tính mạch lạc, ngữ pháp và chính tả.
- Đối với bài tập mã nguồn:
 - * Tích hợp công cụ phân tích tĩnh để kiểm tra cú pháp, phong cách mã hóa, lỗi tiềm ẩn và bảo mật.
 - * Hỗ trợ biên dịch và thực thi kiểm thử tự động để đánh giá tính đúng đắn và chức năng của mã.

- Triển khai cơ chế chấm điểm lặp lại để giảm thiểu sự không nhất quán do tính ngẫu nhiên của LLM.

- **Phương pháp cho phép giáo viên kiểm soát và tinh chỉnh:**

- Phát triển bảng điều khiển trực quan giúp giáo viên xem chi tiết kết quả chấm điểm AI đề xuất cho từng bài tập và tiêu chí.
- Cho phép giáo viên ghi đè điểm số AI đề xuất, cung cấp phản hồi cho sinh viên và tinh chỉnh rubric trong quá trình chấm điểm.
- Lưu trữ lịch sử thay đổi điểm số và rubric để phân tích hiệu quả của AI và cải tiến hệ thống theo thời gian.

2.5.1 Kiến trúc và công nghệ dự kiến:



- **Kiến trúc Microservices:** Ứng dụng được chia thành các dịch vụ nhỏ, độc lập, giao tiếp qua mạng, giúp tăng tính linh hoạt, khả năng mở rộng và bảo trì. Các microservice bao gồm quản lý rubric, chấm điểm AI (cho văn bản, mã nguồn), quản lý người dùng và xác thực, báo cáo và phân tích dữ liệu.
- **Sử dụng Message Broker:** Để đảm bảo giao tiếp hiệu quả giữa các microservice, hệ thống có thể sử dụng message broker (ví dụ: Apache Kafka, RabbitMQ), hỗ trợ truyền thông bất đồng bộ, tăng tính ổn định và chịu lỗi.
- **Mô hình học máy:** Tận dụng các mô hình LLM từ OpenAI, Google AI để xử lý ngôn ngữ tự nhiên và phân tích mã nguồn. Xem xét tinh chỉnh (fine-tuning) trên tập dữ liệu chuyên biệt nếu cần.

2.6 Kết quả dự kiến của đề tài

2.6.1 Số liệu định lượng

- **Độ chính xác của hệ thống:** Một trong những kết quả quan trọng nhất của đề tài là khả năng chứng minh độ chính xác của hệ thống chấm điểm AI. Chúng tôi kỳ vọng đạt được:
 - **Tỷ lệ tương quan cao giữa điểm số AI và giáo viên:** Mục tiêu đạt hệ số tương quan Pearson hoặc Spearman ít nhất 0.8 (hoặc giá trị phù hợp khác) giữa điểm số AI và giáo viên, đảm bảo đánh giá khách quan và phù hợp với tiêu chuẩn con người.
 - **Tỷ lệ chấp nhận điểm số AI cao:** Hướng đến tỷ lệ trên 70% (hoặc ngưỡng phù hợp) các điểm số AI được giáo viên chấp nhận mà không cần điều chỉnh, thể hiện mức độ tin cậy vào hệ thống.
 - **Sai số chấp nhận được:** Đặt khoảng sai số chấp nhận được (ví dụ: ± 1 điểm trên thang 10 hoặc một tỷ lệ phần trăm nhất định), với mục tiêu trên 90% các trường hợp nằm trong khoảng này, đảm bảo sự nhất quán giữa AI và con người.

- **Tốc độ thực thi:** Hiệu quả của hệ thống không chỉ nằm ở độ chính xác mà còn ở tốc độ xử lý. Chúng tôi kỳ vọng:
 - **Thời gian chấm điểm nhanh chóng:** Hệ thống có thể chấm điểm một bài tập (văn bản, PDF, mã nguồn) trong thời gian trung bình dưới một ngưỡng nhất định (ví dụ: dưới 5 giây cho bài văn bản, dưới 10 giây cho mã đơn giản), giúp giảm đáng kể thời gian chấm điểm thủ công.
 - **Khả năng xử lý lớn:** Hệ thống có thể xử lý đồng thời số lượng lớn bài tập trong thời gian ngắn, đáp ứng nhu cầu chấm điểm cho lớp học đông. Mục tiêu cụ thể sẽ được đặt ra về số bài tập hệ thống có thể xử lý trong một phút hoặc một giờ.
- **Hiệu quả sử dụng:** Sự thành công của hệ thống còn được đánh giá dựa trên mức độ mà nó mang lại lợi ích cho người dùng:
 - **Giảm đáng kể thời gian chấm điểm:** Hệ thống giúp giáo viên giảm ít nhất 50% thời gian chấm điểm, cho phép tập trung hơn vào giảng dạy và tương tác với sinh viên.
 - **Mức độ hài lòng cao của người dùng:** Khảo sát và phản hồi từ giáo viên, sinh viên nhằm đảm bảo trải nghiệm sử dụng tốt, tính hữu ích cao và chất lượng phản hồi đáng tin cậy.
 - **Tỷ lệ sử dụng tính năng:** Theo dõi tần suất sử dụng các chức năng (tạo rubric bằng AI, điều chỉnh điểm số, v.v.) để đánh giá hiệu quả và mức độ hữu ích của hệ thống.
- **Đánh giá về chất lượng phản hồi:** Phản hồi là một phần quan trọng của quá trình học tập. Chúng tôi mong muốn:
 - **Phản hồi hữu ích và chi tiết:** Giáo viên và sinh viên đánh giá cao tính hữu ích và mức độ chi tiết của các phản hồi được tạo ra bởi AI, giúp sinh

viên hiểu rõ hơn về những điểm cần cải thiện trong bài làm của mình. Chúng tôi sẽ thu thập phản hồi định tính để đánh giá khía cạnh này.

2.6.2 Sản phẩm đầu ra

- **Hệ thống phần mềm hỗ trợ chấm điểm AI:** Kết quả chính của đề tài sẽ là một hệ thống phần mềm (có thể triển khai dưới dạng ứng dụng web) có đầy đủ các chức năng cần thiết để hỗ trợ giáo viên trong quá trình chấm điểm, bao gồm:
 - Chức năng tạo và tùy chỉnh rubric: Cho phép giáo viên dễ dàng tạo mới hoặc chỉnh sửa các rubric chấm điểm thông qua giao diện trực quan, với sự hỗ trợ gợi ý từ LLM.
 - Chức năng tải lên và quản lý bài tập: Cho phép giáo viên tải lên các bài tập của sinh viên ở nhiều định dạng khác nhau (văn bản, PDF, mã nguồn) và quản lý chúng một cách hiệu quả.
 - Chức năng chấm điểm tự động: Sử dụng các mô hình LLM đã được tích hợp để tự động chấm điểm các bài tập dựa trên rubric đã được xác định.
 - Giao diện xem xét và điều chỉnh điểm số: Cung cấp một giao diện thân thiện để giáo viên xem xét kết quả chấm điểm của AI, so sánh với rubric, thực hiện các điều chỉnh cần thiết và thêm phản hồi cho sinh viên.
 - Chức năng báo cáo và thống kê: Cung cấp các báo cáo tổng quan về hiệu suất chấm điểm, mức độ tương quan giữa điểm AI và điểm của giáo viên, và các thống kê khác liên quan đến quá trình chấm điểm.
- **Tài liệu:** Để đảm bảo tính bền vững và khả năng tiếp tục phát triển của dự án, chúng tôi sẽ xây dựng các tài liệu sau:
 - Báo cáo tổng kết đề tài: Một tài liệu chi tiết mô tả toàn bộ quá trình nghiên cứu, các phương pháp tiếp cận đã sử dụng, các kết quả đạt được,

những phân tích và đánh giá về hiệu quả của hệ thống, cũng như các hướng phát triển tiềm năng trong tương lai.

- Tài liệu hướng dẫn sử dụng: Một tài liệu chi tiết và dễ hiểu, cung cấp hướng dẫn từng bước cho giáo viên về cách sử dụng hệ thống, từ việc tạo rubric, tải bài tập, xem kết quả chấm điểm đến việc điều chỉnh và cung cấp phản hồi.
- Tài liệu kỹ thuật: Một tài liệu mô tả chi tiết về kiến trúc hệ thống, các thành phần chính, các API được sử dụng, cách triển khai và cài đặt hệ thống, cũng như các thông tin kỹ thuật khác cần thiết cho việc bảo trì và phát triển hệ thống trong tương lai.

2.7 Kế hoạch thực hiện

Dưới đây là lịch trình thực hiện dự án PenK Assistant từ tháng 01/2025 đến tháng 07/2025, chia thành các sprint nửa tháng:

Sprint	Công việc	Thành viên	Thời gian
Sprint 1	Tìm hiểu công nghệ và thị trường	Tất cả	01/01 - 14/01
Sprint 2 - 3	Thử nghiệm công cụ chấm bài (Code)	Anh Hoàng	15/01 - 28/01
	Thử nghiệm công cụ chấm bài (Document)	Nhật Minh	15/01 - 28/01
	Thử nghiệm công cụ tạo câu hỏi	Khiêm, Văn Minh	15/01 - 28/01
	Thử nghiệm công cụ chạy test case	Duy	15/01 - 28/01
Sprint 4	Thử nghiệm tạo rubric bằng AI	Khiêm, Văn Minh	29/01 - 11/02
	Thử nghiệm chấm bài theo rubric	Nhật Minh	29/01 - 11/02
	Thử nghiệm tạo và chạy test case	Hoàng	29/01 - 11/02
	Mô tả chức năng và ưu tiên hệ thống	Tất cả	29/01 - 11/02
Sprint 5 - 6	Xây dựng use-case và wireframe	Tất cả	12/02 - 25/02
	Thử nghiệm cài phần mềm chạy code bên ngoài	Hoàng	12/02 - 25/02
	Thiết kế database và kiến trúc hệ thống	Nhật Minh, Khiêm	12/02 - 25/02
Sprint 7	Cải thiện các database và kiến trúc hệ thống	Tất cả	26/02 - 11/03
Sprint 8	Phát triển tính năng tạo rubric AI	Hoàng, Khiêm, Nhật Minh	12/03 - 25/03
	Phát triển giao diện cho tính năng tạo rubric	Duy, Văn Minh	12/03 - 25/03
Sprint 9	Phát triển tính năng nộp bài và	Hoàng, Khiêm, Nhật Minh	26/03 - 08/04
	Phát triển giao diện cho tính năng nộp bài	Duy, Văn Minh	26/03 - 08/04
Sprint 10	Phát triển tính năng chấm bài	Hoàng, Khiêm, Nhật Minh	09/04 - 22/04
	Phát triển giao diện cho tính năng chấm bài	Duy, Văn Minh	09/04 - 22/04
Sprint 11	Phát triển chỉnh sửa điểm thủ công	Hoàng, Khiêm, Nhật Minh	23/04 - 06/05
	Phát triển giao diện cho tính năng chấm bài	Duy, Văn Minh	23/04 - 06/05
Sprint 12	Phát triển tính năng sử dụng plugin chấm bài	Hoàng, Khiêm, Nhật Minh	07/05 - 20/05
	Phát triển giao diện cho tính năng sử dụng plugin chấm bài	Duy, Văn Minh	07/05 - 20/05
	Báo cáo và phản hồi cơ bản	Tất cả	07/05 - 20/05
Sprint 13	Phát triển tính năng báo cáo và thống kê	Hoàng, Khiêm, Nhật Minh	21/05 - 03/06
	Phát triển giao diện cho tính năng báo cáo và thống kê	Duy, Văn Minh	21/05 - 03/06
Sprint 14	Tiếp tục hoàn thiện sản phẩm	Tất cả	04/06 - 30/06
	Hoàn thành báo cáo	Tất cả	04/06 - 30/06

Bảng 1: Phân công công việc cho hệ thống chấm bài AI với thời gian dự kiến

Tài liệu

- [1] Online, “Rubric best practices, examples, and templates.”
[https://teaching-resources.delta.ncsu.edu/rubric_
best-practices-examples-templates/](https://teaching-resources.delta.ncsu.edu/rubric_best-practices-examples-templates/).

XÁC NHẬN
CỦA NGƯỜI HƯỚNG DẪN
(*Ký và ghi rõ họ tên*)

TP. Hồ Chí Minh, ngày... tháng... năm...
NHÓM SINH VIÊN THỰC HIỆN
(*Ký và ghi rõ họ tên*)