

Nghiên cứu phát triển hệ thống học lập trình trực tuyến tích hợp công cụ phản hồi và chấm điểm tự động

Tổng Văn Trường*

Trường Đại học Tân Trào, phường Minh Xuân, tỉnh Tuyên Quang, Việt Nam

Ngày nhận bài 1/7/2025; ngày chuyển phản biện 3/7/2025; ngày nhận phản biện 25/7/2025; ngày chấp nhận đăng 29/7/2025

Tóm tắt:

Trong bối cảnh chuyển đổi số trong giáo dục, việc nâng cao năng lực lập trình cho sinh viên, đặc biệt là sinh viên không chuyên công nghệ thông tin, ngày càng trở nên quan trọng. Tuy nhiên, nhiều sinh viên vẫn gặp khó khăn trong việc tiếp cận và thực hành lập trình một cách hiệu quả. Nghiên cứu này khảo sát thực trạng học lập trình và đánh giá nhu cầu hỗ trợ của 50 sinh viên không chuyên công nghệ thông tin thuộc các ngành Giáo dục Mầm non, Giáo dục Tiểu học và Nông nghiệp tại Trường Đại học Tân Trào. Dựa trên kết quả khảo sát, nhóm tác giả đã phát triển một hệ thống học lập trình trực tuyến tích hợp công cụ phản hồi và chấm điểm tự động, nhằm hỗ trợ sinh viên luyện tập thường xuyên và nhận phản hồi tức thì. Hệ thống được xây dựng theo kiến trúc mô-đun, gồm các thành phần chính như giao diện người dùng, máy chủ API (Application Programming Interface: Giao diện lập trình ứng dụng), cơ sở dữ liệu và công cụ chấm mã trong môi trường cách ly an toàn. Bài báo cũng đề xuất các chiến lược giảng dạy như dạy học phân hóa theo trình độ, sử dụng microlearning và thực hành lập trình thường xuyên trên nền tảng LMS (Learning Management System: Hệ thống quản lý học tập) của nhà trường. Kết quả đánh giá cho thấy, hệ thống đã góp phần nâng cao đáng kể hiệu quả học tập, sự tự tin và khả năng vận dụng kiến thức lập trình vào các bài toán thực tiễn.

Từ khóa: đánh giá mã nguồn, hệ thống chấm điểm tự động, học lập trình trực tuyến, nền tảng e-learning, phản hồi tự động.

Chỉ số phân loại: 1.2, 2.11

Developing and evaluating an online programming learning platform with integrated automated assessment for university students

Van Truong Tong*

Tan Trao University, Minh Xuan Ward, Tuyen Quang Province, Vietnam

Received 1 July 2025; revised 25 July 2025; accepted 29 July 2025

Abstract:

In the context of digital transformation in education, enhancing programming competence among university students, especially those who are not majoring in information technology, has become increasingly important. However, many non-IT students still face significant difficulties in effectively accessing and practicing programming skills. This study investigates the current status of programming learning and assesses the support needs of 50 non-IT students from various majors at Tan Trao University, including Early Childhood Education, Primary Education, and Agriculture. Based on the survey results, the authors developed an online programming learning system integrated with automated assessment to support regular practice and provide immediate feedback for students. The system is designed using a modular architecture, consisting of key components such as a user interface, an API (Application Programming Interface) server, a database, and a secure isolated code evaluation environment. The article also proposes teaching strategies including differentiated instruction based on student proficiency levels, the use of microlearning, and frequent programming practice through the university's LMS (Learning Management System). Evaluation results indicate that the system significantly improves learning effectiveness, student confidence, and the ability to apply programming knowledge to real-world problem-solving tasks.

Keywords: automated feedback, automated grading system, code evaluation, e-learning platform, online programming learning.

Classification numbers: 1.2, 2.11

*Email: tongtruongdht@gmail.com

1. Đặt vấn đề

Trong kỷ nguyên số, lập trình không còn là kỹ năng chuyên biệt dành riêng cho sinh viên ngành khoa học máy tính mà đã trở thành một năng lực nền tảng, có tính liên ngành cao, được ứng dụng rộng rãi trong các lĩnh vực như kinh tế, kỹ thuật, giáo dục và khoa học xã hội. Khi các ngành công nghiệp ngày càng phụ thuộc vào quá trình ra quyết định dựa trên dữ liệu, tự động hóa và chuyển đổi số, năng lực lập trình giúp sinh viên đại học hiểu sâu hơn về tư duy tính toán, phát triển kỹ năng giải quyết vấn đề và thích nghi hiệu quả với yêu cầu của thị trường lao động tương lai.

Tại Việt Nam, cũng như nhiều quốc gia đang phát triển khác, cải cách giáo dục ở tầm quốc gia đã nhấn mạnh vai trò của năng lực số và việc tích hợp công nghệ thông tin trong chương trình đào tạo. Tuy nhiên, khoảng cách giữa định hướng chính sách và năng lực thực tế của sinh viên vẫn là một thách thức lớn, đặc biệt tại các cơ sở giáo dục đại học vùng sâu, vùng xa như Trường Đại học Tân Trào, tỉnh Tuyên Quang.

Cách tiếp cận truyền thống trong giảng dạy lập trình tại các trường đại học Việt Nam chủ yếu vẫn dựa vào phương pháp thuyết trình, bài tập thực hành cố định và chấm điểm thủ công các bài nộp của sinh viên. Những phương pháp này bộc lộ nhiều hạn chế về hiệu quả, đặc biệt trong bối cảnh lớp học đông hoặc có nhiều sinh viên không đến từ nền tảng kỹ thuật. Một trong những điểm yếu cốt lõi là thiếu phản hồi kịp thời, đây là yếu tố then chốt trong giáo dục lập trình. Sinh viên không chỉ cần biết mã của mình có chạy được hay không, mà còn cần hiểu rõ nguyên nhân khi chương trình không hoạt động. Việc chấm bài thủ công thường tốn nhiều công sức và bị trì hoãn, gây ảnh hưởng tiêu cực đến hiệu quả học tập và làm giảm động lực học của sinh viên.

Bên cạnh đó, sự đa dạng về nền tảng kiến thức và trình độ lập trình giữa các sinh viên đòi hỏi một môi trường học tập mang tính cá nhân hóa, điều mà lớp học truyền thống khó đáp ứng. Một số sinh viên có thể tiếp thu nhanh các khái niệm cơ bản, trong khi những người khác lại cần sự hướng dẫn cụ thể, từng bước. Nếu thiếu các cơ chế học tập thích ứng, giảng viên sẽ gặp khó khăn trong việc điều chỉnh tốc độ giảng dạy sao cho phù hợp với tất cả người học. Ngoài ra, các yếu tố bên ngoài như sự chênh lệch trong khả năng tiếp cận máy tính, hạ tầng internet yếu kém hoặc thiếu công cụ lập trình hiện đại tại các trường đại học vùng nông thôn càng làm trầm trọng thêm rào cản giáo dục.

Để ứng phó với những thách thức nêu trên, bài báo này giới thiệu việc xây dựng và đánh giá một nền tảng học lập trình trực tuyến có tích hợp công cụ chấm điểm tự động. Nền tảng cho phép sinh viên viết mã trực tiếp trên trình duyệt mà không cần cài đặt phần mềm, hỗ trợ nhiều ngôn ngữ lập trình, cung cấp tính năng gợi ý mã (auto-completion), tô sáng cú pháp và tự động đánh giá bài nộp dựa trên các bộ kiểm thử được định nghĩa trước. Hệ thống chấm điểm cung cấp phản hồi tức thời về cả lỗi cú pháp và lỗi ngữ nghĩa, từ đó giúp sinh viên cải thiện mã nguồn một cách lặp đi lặp lại. Ngoài ra, nền tảng còn tích hợp công cụ hỗ trợ giảng viên theo dõi tiến độ học tập, giao bài và phân tích hiệu suất học tập ở cả cấp độ cá nhân và toàn lớp.

Nền tảng này đã được triển khai thử nghiệm trong một học phần lập trình cơ bản dành cho sinh viên không chuyên ngành công nghệ thông tin. Phương pháp đánh giá hỗn hợp được áp dụng, kết hợp dữ liệu định lượng (như tỷ lệ hoàn thành bài tập, điểm tiền kiểm tra và hậu kiểm tra, nhật ký sử dụng hệ thống) với phản hồi định tính thu thập từ khảo sát và phỏng vấn. Kết quả cho thấy mức độ tham gia của sinh viên được cải thiện rõ rệt, chu kỳ học tập ngắn hơn và khả năng ghi nhớ các khái niệm lập trình được nâng cao. Việc tự động hóa quá trình đánh giá không chỉ giúp giảm tải công việc cho giảng viên mà còn tạo nên một môi trường học tập chủ động, nơi sinh viên học thông qua thử - sai và phản hồi tức thời.

Thông qua việc phát triển và đánh giá nền tảng học lập trình trực tuyến này, nghiên cứu hướng đến việc cung cấp một giải pháp giảng dạy lập trình có khả năng mở rộng, thích ứng linh hoạt và tiết kiệm tài nguyên - đặc biệt phù hợp với bối cảnh giáo dục đại học tại Việt Nam. Bài báo cũng chứng minh rằng, việc tích hợp các công cụ học tập dựa trên công nghệ có thể góp phần giải quyết hiệu quả những thách thức về sư phạm và tổ chức, nhất là trong điều kiện cơ sở vật chất còn hạn chế. Các phát hiện từ nghiên cứu mang đến hàm ý thiết thực cho đội ngũ giảng viên, cán bộ quản lý và nhà hoạch định chính sách trong nỗ lực hiện đại hóa hoạt động giảng dạy lập trình và kỹ năng số trong các ngành học tại Việt Nam và những bối cảnh tương tự.

2. Cơ sở lý luận

Hoạt động giảng dạy và học tập lập trình từ lâu đã đặt ra nhiều thách thức đáng kể cho cả giảng viên và người học, do đặc thù mang tính trừu tượng cao, đòi hỏi tư duy logic và yêu cầu thực hành thường xuyên. Trong vài thập kỷ gần đây, đã có nhiều công trình nghiên cứu được công bố nhằm giải quyết các khó khăn này, đặc biệt tập trung vào việc phát triển các môi trường học lập trình trực tuyến, hệ thống chấm mã tự động và cơ chế học tập thích ứng nhằm hỗ trợ người học đa dạng. Phần này trình bày tổng quan các công trình tiêu biểu trong bốn hướng nghiên cứu chính: (1) Nền tảng học lập trình trực tuyến, (2) Công cụ chấm mã tự động, (3) Kỹ thuật phân tích và đánh giá mã nguồn, (4) Hệ thống học tập cá nhân hóa và thích ứng trong giáo dục lập trình.

2.1. Nền tảng học lập trình trực tuyến

Các môi trường học lập trình trực tuyến ngày càng trở nên phổ biến nhờ khả năng tiếp cận thuận tiện, tính mở rộng linh hoạt và tiềm năng hỗ trợ học tập tương tác theo nhịp độ cá nhân. Các nền tảng như Codecademy, Khan Academy, freeCodeCamp và Code.org đã góp phần dân chủ hóa việc học lập trình bằng cách cung cấp các khóa học tương tác miễn phí hoặc chi phí thấp, dẫn dắt người học vượt qua chuỗi thử thách lập trình có mức độ khó tăng dần. Những hệ thống này được đặc trưng bởi các tính năng như thực thi mã ngay trên trình duyệt, phản hồi theo thời gian thực và thiết kế bài học theo hướng trò chơi hóa - các yếu tố đã được chứng minh là có tác dụng thúc đẩy động lực và khả năng ghi nhớ của người học [1].

Trong bối cảnh giáo dục đại học chính quy, nhiều cơ sở đào tạo đã phát triển nền tảng riêng để đáp ứng yêu cầu sử dụng cụ thể. Chẳng hạn, hệ thống Kattis do Viện Công nghệ Hoàng gia Thụy Điển (KTH) phát triển phục vụ cho các khóa học và thi đấu lập trình; trong khi đó, CodeRunner - một plugin mã nguồn mở tích hợp với Moodle cho phép giảng viên tạo câu hỏi lập trình có thể chấm điểm tự động. Các nền tảng này hỗ trợ tạo bài tập tùy chỉnh, tích hợp với hệ thống quản lý học tập (LMS) và sử dụng nhiều bộ kiểm thử để đánh giá.

Tại Việt Nam, nhiều trường đại học hiện đang sử dụng các nền tảng phổ thông như HackerRank hoặc repl.it để hỗ trợ giảng dạy lập trình, tuy nhiên việc xây dựng và triển khai các hệ thống riêng có tích hợp công cụ đánh giá chuyên biệt vẫn còn hạn chế. Một trong những rào cản đáng kể là thiếu giao diện và nội dung bằng tiếng Việt, điều này gây khó khăn cho sinh viên không chuyên ngành hoặc sinh viên đến từ vùng sâu, vùng xa.

2.2. Công cụ chấm mã tự động

Một trong những đột phá quan trọng trong giáo dục lập trình là sự phát triển của các hệ thống chấm mã tự động, cho phép đánh giá bài làm của sinh viên theo thời gian thực dựa trên các bộ kiểm thử được định nghĩa sẵn. Những hệ thống này giúp giảm đáng kể khối lượng công việc chấm điểm cho giảng viên, đồng thời mang lại phản hồi tức thì - một yếu tố then chốt trong dạy học lập trình hiệu quả [2].

Các hệ thống đầu tiên như Mooshak, AutoLab và DOMjudge được phát triển phục vụ thi đấu lập trình và sau đó được điều chỉnh để sử dụng trong đào tạo. Các hệ thống này thường áp dụng phương pháp kiểm thử hộp đen ("black-box testing"), trong đó độ chính xác của lời giải được xác định bằng cách chạy mã trên các đầu vào ẩn và so sánh đầu ra với kết quả mong đợi. Những hệ thống hiện đại hơn như Gradescope và CodeGrade không chỉ tích hợp với LMS mà còn cung cấp phản hồi nâng cao về phong cách mã, cấu trúc và hiệu suất.

Dù đã có nhiều tiến bộ, nhưng nghiên cứu của H. Keuning và cs (2018) [3] cho thấy, các hệ thống hiện hành vẫn thiếu phản hồi có tính ngữ nghĩa sâu - yếu tố quan trọng với người học mới bắt đầu. Ngoài ra, phần lớn công cụ chỉ tập trung vào kiểm tra độ đúng của mã mà bỏ qua chất lượng mã nguồn, điều này hạn chế hiệu quả sư phạm. Xu hướng hiện nay là tích hợp phân tích mã tĩnh và thực thi ký hiệu (symbolic execution) vào quy trình đánh giá nhằm cung cấp thông tin phản hồi mang tính chẩn đoán chuyên sâu hơn.

2.3. Kỹ thuật phân tích và đánh giá mã nguồn

Để hỗ trợ đánh giá sâu hơn đối với bài lập trình, các nhà nghiên cứu đã đề xuất nhiều kỹ thuật phân tích mã vượt ra ngoài kiểm thử đơn thuần. Các phương pháp phân tích tĩnh như phân tích cây cú pháp trừu tượng (AST), phân tích luồng điều khiển và phân tích luồng dữ liệu được sử dụng để xác định đặc trưng cấu trúc của mã, đo lường độ phức tạp và phát hiện mẫu lập trình sai hoặc gian lận. Các công cụ như PMD, SonarQube và JPlag được ứng dụng rộng rãi trong môi trường học thuật để phân tích chất lượng mã và phát hiện đạo văn.

Trong các hệ thống chấm điểm tự động, phương pháp dựa trên AST cho phép giảng viên định nghĩa cấu trúc lời giải mẫu và so sánh với bài làm của sinh viên để đánh giá mức độ hiểu bài [4]. Ví dụ, so sánh AST có thể xác định sinh viên có sử dụng đúng cấu trúc đề quy khi cài đặt thuật toán sắp xếp hay không, thay vì chỉ gọi hàm thư viện. Những phương pháp này rất hữu ích cho đánh giá hình thành và nhận diện sai lệch nhận thức.

Một số hệ thống tiên tiến như PaireLearn hay FeedbackFruits còn triển khai mô hình đánh giá dựa trên lời giải chuẩn ("model-driven assessment"), trong đó hệ thống tự động phát hiện các sai lệch về logic, cấu trúc hoặc hiệu suất so với lời giải chuẩn. Kỹ thuật này hỗ trợ việc tạo phản hồi cá nhân hóa và được chứng minh là giúp nâng cao kết quả học tập trong các khóa học nhập môn lập trình.

2.4. Hệ thống học tập cá nhân hóa và thích ứng

Trước sự đa dạng về xuất phát điểm, kinh nghiệm lập trình và năng lực nhận thức của sinh viên, việc cá nhân hóa hoạt động học đã trở thành trọng tâm trong nghiên cứu giáo dục lập trình. Các hệ thống học tập thích ứng hướng tới việc điều chỉnh nội dung, tốc độ và phản hồi sao cho phù hợp với từng người học. Các hệ thống như IntelliJ Edu, SmartTutor và BlueJ đã tích hợp yếu tố thích ứng bằng cách theo dõi hiệu suất học tập và điều chỉnh mức độ khó phù hợp.

Gần đây, một số hướng tiếp cận mới đã ứng dụng thuật toán học máy để phân tích tương tác của sinh viên với các bài lập trình và dự đoán khả năng gặp khó khăn trong học tập. Theo đó, các nhà khoa học đã sử dụng mạng nơ-ron hồi tiếp (RNN) để mô hình hóa hành vi lập trình và dự báo kết quả học. Những hệ thống dựa trên dữ liệu này mở ra khả năng xây dựng các hệ thống gia sư thông minh có thể hỗ trợ chủ động như đưa gợi ý, đề xuất bài tập hoặc động viên học tập.

Tuy nhiên, việc phát triển và triển khai các hệ thống như vậy tại những môi trường hạn chế về nguồn lực - như các trường đại học vùng miền ở Việt Nam - còn đối mặt với thách thức kỹ thuật, sư phạm và hạ tầng [5-7]. Việc cân bằng giữa mục tiêu cá nhân hóa với thực tế hạn chế về năng lực giảng viên, băng thông internet và hỗ trợ thể chế là điều tất yếu.

2.5. Khoảng trống trong nghiên cứu

Mặc dù thế giới đã đạt được nhiều thành tựu trong việc thiết kế nền tảng học lập trình trực tuyến và hệ thống chấm điểm tự động, phần lớn các hệ thống hiện có được phát triển trong môi trường giàu tài nguyên và khó thích ứng với điều kiện thực tế của giáo dục đại học tại Việt Nam. Những thách thức điển hình tại các trường thuộc tỉnh như Tuyên Quang bao gồm thiếu hạ tầng máy tính, hạn chế về hỗ trợ kỹ thuật và thiếu đào tạo giảng viên trong việc ứng dụng công nghệ mới.

Hơn nữa, hiện vẫn còn thiếu các nghiên cứu thực nghiệm đánh giá hiệu quả của nền tảng lập trình tích hợp - đặc biệt là các hệ thống kết hợp giảng dạy, môi trường lập trình và phản hồi tự động - trong bối cảnh Việt Nam hoặc các quốc gia Đông Nam Á có

điều kiện tương tự. Nghiên cứu này nhằm lấp đầy khoảng trống đó thông qua việc thiết kế, triển khai và đánh giá một nền tảng học lập trình nhẹ, hoạt động trên trình duyệt, đảm bảo tính khả thi kỹ thuật và phù hợp sư phạm tại Trường Đại học Tân Trào.

3. Thiết kế và triển khai hệ thống

Nền tảng được đề xuất được thiết kế với kiến trúc dạng mô-đun và có khả năng mở rộng cao, nhằm hỗ trợ toàn bộ quy trình giảng dạy lập trình bao gồm: nộp bài, đánh giá tự động, cung cấp phản hồi và theo dõi tiến trình học tập. Hệ thống được xây dựng trên nền tảng các công nghệ mã nguồn mở hiện đại, đảm bảo tính linh hoạt, dễ triển khai và có thể mở rộng trong các môi trường giáo dục có nguồn lực hạn chế. Mô hình triển khai theo kiến trúc client-server điển hình, bao gồm bốn thành phần chính: giao diện web phía người dùng (frontend), máy chủ API (backend), cơ sở dữ liệu quan hệ và công cụ chấm điểm tự động trong môi trường cách ly an toàn (sandboxed auto-grading engine).

3.1. Giao diện người dùng (Frontend): Phát triển trên nền React

Phần giao diện người dùng được xây dựng bằng thư viện ReactJS - một công nghệ JavaScript phổ biến được sử dụng rộng rãi trong phát triển các ứng dụng web tương tác. Kiến trúc dựa trên thành phần (component-based) của React giúp phân tách rõ ràng các chức năng, tăng tính tái sử dụng và dễ dàng bảo trì. Giao diện được thiết kế cho hai nhóm người dùng chính là sinh viên và giảng viên, mỗi nhóm được cung cấp một bảng điều khiển (dashboard) tùy biến riêng.

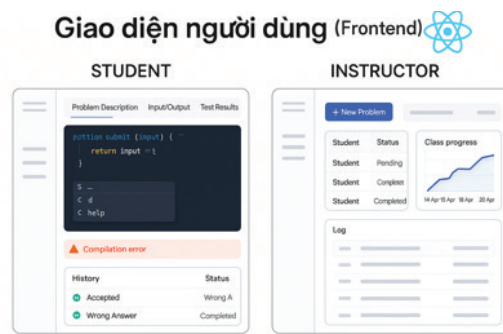
Đối với sinh viên, giao diện bao gồm:

- Trình soạn thảo mã (code editor) hỗ trợ tô sáng cú pháp và gợi ý mã tự động (sử dụng Monaco Editor - tương tự VSCode).
- Các tab để xem mô tả bài toán, ví dụ vào/ra và kết quả kiểm thử.
- Thông báo lỗi biên dịch hoặc lỗi thực thi theo thời gian thực.
- Bộ theo dõi lịch sử nộp bài cùng trạng thái đánh giá và thời gian nộp.

Đối với giảng viên, giao diện hỗ trợ:

- Tạo bài toán và quản lý bộ kiểm thử.
- Theo dõi tiến độ nộp bài của sinh viên theo thời gian thực.
- Trực quan hóa tiến độ lớp học thông qua biểu đồ và nhật ký hệ thống.
- Tùy chọn chấm điểm thủ công và điều chỉnh kết quả khi cần thiết.

Giao diện frontend tương tác với máy chủ backend thông qua API chuẩn REST trên giao thức HTTPS và hỗ trợ mở rộng giao diện đa ngôn ngữ (ví dụ: chuyển đổi Anh - Việt) phục vụ nội địa hóa (hình 1).



Hình 1. Giao diện người dùng dành cho sinh viên và giảng viên trên hệ thống, cho phép lập trình, theo dõi tiến độ và quản lý bài tập.

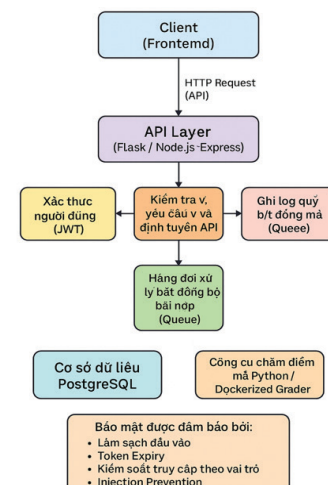
3.2. Máy chủ xử lý trung tâm (Backend): Lớp API (Flask hoặc Node.js)

Phần backend được phát triển bằng Python Flask hoặc Node.js kết hợp Express, tùy theo yêu cầu triển khai và sở thích của đội ngũ phát triển. Flask tích hợp tốt với các logic chấm điểm bằng Python và có độ nhẹ cao; trong khi đó Node.js hỗ trợ xử lý sự kiện bất đồng bộ (asynchronous), phù hợp với môi trường có số lượng truy cập đồng thời lớn.

Các chức năng chính của backend bao gồm:

- Xác thực và phân quyền người dùng sử dụng JWT (JSON Web Tokens).
- Điều hướng và kiểm tra hợp lệ các yêu cầu API (ví dụ: nộp bài, truy xuất bộ kiểm thử, lịch sử điểm).
- Quản lý hàng đợi xử lý bài nộp theo cơ chế bất đồng bộ.
- Giao tiếp với công cụ chấm điểm và cơ sở dữ liệu PostgreSQL.
- Ghi log sự kiện phục vụ phân tích và gỡ lỗi hệ thống.

Các yếu tố bảo mật được chú trọng như làm sạch đầu vào (input sanitization), thiết lập thời hạn cho token, kiểm soát truy cập theo vai trò nhằm ngăn chặn tấn công chèn mã độc và truy cập trái phép (hình 2).



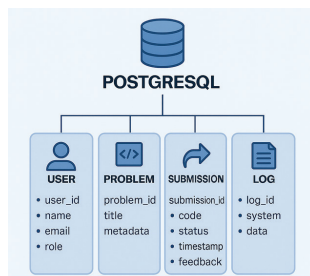
Hình 2. Kiến trúc hệ thống chấm điểm mã nguồn tự động sử dụng API, hàng đợi bất đồng bộ và công cụ chấm điểm trong Docker.

3.3. Cơ sở dữ liệu (Database): Hệ quản trị PostgreSQL

Hệ thống sử dụng cơ sở dữ liệu quan hệ PostgreSQL để lưu trữ toàn bộ dữ liệu lâu dài, bao gồm:

- Hồ sơ người dùng (sinh viên, giảng viên, quản trị viên).
- Danh sách bài toán lập trình và siêu dữ liệu liên quan.
- Bộ kiểm thử (các cặp dữ liệu vào/ra).
- Bản ghi nộp bài (mã nguồn, trạng thái, thời gian, phản hồi).
- Nhật ký hệ thống và dữ liệu phân tích.

PostgreSQL được lựa chọn nhờ độ ổn định cao, khả năng xử lý các truy vấn phức tạp, hỗ trợ chỉ mục, trường JSON và tiềm năng mở rộng theo mô hình lai (kết hợp dữ liệu quan hệ và tài liệu). Cơ sở dữ liệu được chuẩn hóa nhằm tối ưu hóa truy vấn và hỗ trợ kiểm tra truy vết (audit trail) đảm bảo tính toàn vẹn dữ liệu (hình 3).



Hình 3. Cấu trúc cơ sở dữ liệu PostgreSQL của hệ thống.

3.4. Công cụ chấm điểm tự động: Thực thi cách ly bằng Docker

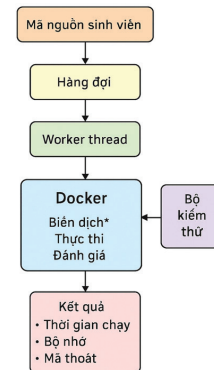
Trung tâm của hệ thống là công cụ chấm điểm tự động, được thiết kế để thực thi và đánh giá mã nguồn của sinh viên trong môi trường cách ly bảo mật. Nền tảng sử dụng công nghệ Docker để cô lập (sandbox) từng phiên thực thi mã nhằm đảm bảo an toàn, kiểm soát tài nguyên và khả năng tái lập.

Quy trình xử lý mỗi bài nộp diễn ra như sau:

- Mã nguồn của sinh viên được gửi đến backend và xếp vào hàng đợi.
- Một luồng xử lý (worker thread) khởi tạo container Docker mới theo môi trường ngôn ngữ lập trình tương ứng (ví dụ: Python, C++, Java).
- Bên trong container, mã được biên dịch (nếu cần), thực thi và đánh giá dựa trên bộ kiểm thử định sẵn.
- Thời gian chạy, bộ nhớ sử dụng và mã thoát (exit code) được ghi nhận. Nếu bài làm cho kết quả đúng với tất cả các kiểm thử trong giới hạn tài nguyên, bài được đánh dấu “Đạt”.
- Sau khi hoàn tất, container được hủy để tránh rò rỉ tài nguyên.

Docker đảm bảo giới hạn tài nguyên CPU và bộ nhớ cho mỗi container nhằm ngăn chặn lạm dụng (ví dụ: vòng lặp vô hạn). Hệ thống có cơ chế giới hạn thời gian chạy và kích thước đầu ra để

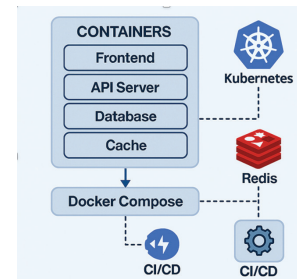
phòng ngừa treo hệ thống. Mọi bản ghi chấm điểm đều được lưu để phục vụ phân tích hậu kỳ và gỡ lỗi (hình 4).



Hình 4. Công cụ chấm điểm tự động: Thực thi cách ly bằng Docker.

3.5. Tích hợp và triển khai hệ thống

Toàn bộ nền tảng được đóng gói và quản lý bằng Docker Compose, cho phép triển khai nhanh chóng trên nhiều môi trường khác nhau (máy chủ nội bộ, máy chủ đám mây hoặc VPS). Đối với các cơ sở giáo dục có ngân sách hạn chế, hệ thống có thể chạy ổn định trên một máy chủ đơn. Trong tương lai, để mở rộng quy mô, các thành phần có thể được tách riêng và triển khai trên Kubernetes hoặc hệ thống điều phối container tương đương (hình 5). Ngoài ra, hệ thống hỗ trợ tích hợp tùy chọn với bộ nhớ đệm Redis để xử lý hàng đợi và tăng tốc truy xuất. Các tập lệnh CI/CD cơ bản (ví dụ: GitHub Actions) được sử dụng để tự động hóa quá trình cập nhật và triển khai hệ thống.



Hình 5. Mô hình tích hợp và triển khai hệ thống sử dụng Docker Compose, Redis, CI/CD và hỗ trợ mở rộng với Kubernetes.

4. Đánh giá hiệu quả

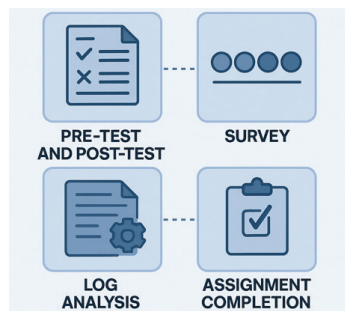
Để đánh giá hiệu quả của nền tảng học lập trình trực tuyến tích hợp hệ thống chấm điểm tự động được đề xuất, nghiên cứu đã tiến hành một quá trình đánh giá theo phương pháp hỗn hợp, kết hợp thu thập dữ liệu định lượng và định tính. Mục tiêu của đánh giá là làm rõ tác động của nền tảng đối với kết quả học tập, mức độ tham gia và sự hài lòng của sinh viên, đồng thời xem xét cách hệ thống hỗ trợ giảng viên trong việc quản lý bài tập và cung cấp phản hồi. Quá trình đánh giá được triển khai trong suốt sáu tuần của học phần “Lập trình cơ bản” tại Trường Đại học Tân Trào, với sự tham gia của 50 sinh viên đại học thuộc các ngành không chuyên công nghệ thông tin.

4.1. Thiết kế đánh giá

Khung đánh giá bao gồm bốn thành phần chính:

- Bài kiểm tra đầu khóa và cuối khóa nhằm đo lường sự tiến bộ về kiến thức lập trình và kỹ năng giải quyết vấn đề.
- Phiếu khảo sát theo thang đo Likert để thu thập nhận định của sinh viên về khả năng sử dụng hệ thống, hiệu quả phản hồi và mức độ hài lòng trong quá trình học.
- Phân tích nhật ký hoạt động của hệ thống nhằm lượng hóa hành vi học tập, tần suất nộp bài và hiệu suất qua các bộ kiểm thử.
- Đánh giá tỷ lệ hoàn thành và độ chính xác của bài tập để xác định mức độ tác động của hệ thống đối với hiệu quả học tập.

Nội dung học và thời lượng giảng dạy được giữ nguyên cho cả hai nhóm sinh viên: nhóm thực nghiệm ($n=25$) sử dụng nền tảng, và nhóm đối chứng ($n=25$) học theo phương pháp truyền thống với hình thức chấm điểm thủ công (hình 6).



Hình 6. Các phương pháp thu thập dữ liệu: bài kiểm tra trước và sau, khảo sát, phân tích log hệ thống và hoàn thành bài tập.

4.2. Phân tích kết quả tiền kiểm tra và hậu kiểm tra

Cả hai nhóm đều thực hiện bài kiểm tra chẩn đoán gồm 20 câu hỏi về các khái niệm lập trình cơ bản (biến, vòng lặp, điều kiện, hàm) vào đầu và cuối khóa học. Nhóm thực nghiệm cho thấy mức độ cải thiện có ý nghĩa thống kê, với điểm trung bình tăng từ 48,6 lên 76,2%, so với mức tăng từ 47,2 lên 63,8% của nhóm đối chứng. Kết quả kiểm định t (paired-samples t-test) cho thấy $p < 0,01$, khẳng định sự cải thiện trong nhóm thực nghiệm là có ý nghĩa và không xảy ra ngẫu nhiên.

Điều này cho thấy phản hồi tức thời và mang tính lặp lại do hệ thống chấm điểm tự động cung cấp đã đóng vai trò then chốt trong việc củng cố kiến thức và giúp sinh viên sửa lỗi sớm ngay trong quá trình học.

4.3. Phản hồi của sinh viên qua khảo sát Likert

Kết thúc học phần, sinh viên nhóm thực nghiệm đã hoàn thành bảng khảo sát chi tiết theo thang đo Likert 5 mức độ (1 = hoàn toàn không đồng ý, 5 = hoàn toàn đồng ý), tập trung vào bốn khía cạnh: khả năng sử dụng nền tảng, mức độ rõ ràng của phản hồi, cảm nhận về sự tiến bộ trong học tập và động lực học.

Một số kết quả nổi bật bao gồm:

- Khả năng sử dụng và tiếp cận: 88% sinh viên đồng ý hoặc hoàn toàn đồng ý rằng nền tảng dễ sử dụng, không yêu cầu cài đặt phức tạp.
- Chất lượng phản hồi: 92% đánh giá phản hồi tự động là kịp thời và hữu ích trong việc hiểu và sửa lỗi.
- Tiến bộ trong học tập: 86% tin rằng họ cải thiện kỹ năng lập trình nhanh hơn so với các trải nghiệm học tập trước đó.
- Động lực và mức độ gắn kết: 84% cho biết họ cảm thấy có động lực hơn khi làm bài nhờ cơ chế phản hồi tức thì và hệ thống chấm điểm minh bạch.

Trong phần phản hồi mở, nhiều sinh viên chia sẻ rằng việc được thử nghiệm mã và nhận phản hồi ngay lập tức khiến quá trình học “bớt áp lực” và “giống như một trò chơi khám phá”.

4.4. Phân tích nhật ký hệ thống và hành vi nộp bài

Để bổ sung cho dữ liệu khảo sát, nhóm nghiên cứu đã phân tích chi tiết nhật ký hoạt động của hệ thống. Các chỉ số thu thập bao gồm:

- Số lượt nộp bài trung bình mỗi bài tập: 6,2 lần, cho thấy hành vi học tập lặp lại - cải thiện qua thử nghiệm.
- Tỷ lệ hoàn thành bài tập: nhóm thực nghiệm đạt 96%, cao hơn đáng kể so với 68% của nhóm đối chứng.
- Thời gian trung bình để nộp bài thành công sau lần thử đầu tiên: giảm từ 42 phút (tuần 1) xuống còn 18 phút (tuần 6), cho thấy sự tiến bộ về hiệu quả giải quyết vấn đề.
- Tỷ lệ vượt qua từng bộ kiểm thử: tăng 27% trong suốt quá trình học, với số sinh viên vượt qua toàn bộ các trường hợp biên trong lần nộp đầu tiên nhiều hơn vào các tuần sau.

Dữ liệu từ hệ thống xác nhận rằng, sinh viên tích cực tham gia vào quá trình học thử - sai và sử dụng phản hồi để cải thiện dần bài làm. Đồng thời, dữ liệu này giúp giảng viên nhận diện các lỗi phổ biến để có biện pháp giảng dạy bổ trợ phù hợp, cả ở cấp độ lớp và cá nhân.

4.5. Quan điểm của giảng viên và giảm tải khối lượng công việc

Từ góc nhìn của giảng viên, hệ thống đã giúp giảm đáng kể thời gian dành cho các công việc chấm điểm lặp đi lặp lại. Trung bình, giảng viên báo cáo thời gian chấm bài giảm khoảng 70%, từ đó có thêm thời gian dành cho hỗ trợ cá nhân hóa và thảo luận chuyên sâu về nội dung môn học trong giờ học. Việc chấm điểm tự động cũng đảm bảo tính nhất quán và loại bỏ sự thiên lệch chủ quan - điều đặc biệt quan trọng trong các lớp học lập trình nhập môn với số lượng lớn sinh viên.

Bảng điều khiển của giảng viên cung cấp cái nhìn tức thời về tiến độ học tập của sinh viên, bao gồm các biểu đồ trực quan về tần suất nộp bài, xu hướng thành công và thời gian hoàn thành

mỗi bài. Các dữ liệu phân tích này được sử dụng để điều chỉnh tiến độ giảng dạy và xác định những sinh viên có nguy cơ tụt lại phía sau.

Tóm lại, quá trình đánh giá đã chứng minh rằng nền tảng học lập trình được đề xuất không chỉ cải thiện rõ rệt kết quả học tập và mức độ gắn kết của sinh viên mà còn mang lại giá trị hỗ trợ thực tiễn cho giảng viên trong việc giảng dạy hiệu quả hơn. Việc tích hợp cơ chế đánh giá tự động và phản hồi tương tác tạo ra một giải pháp học tập có khả năng mở rộng, tiết kiệm tài nguyên, đặc biệt phù hợp với các trường đại học tại vùng sâu, vùng xa đang trong quá trình hiện đại hóa chương trình đào tạo lập trình.

5. Kết luận và đề xuất

Nghiên cứu này đã trình bày quá trình thiết kế, triển khai và đánh giá một nền tảng học lập trình trực tuyến tích hợp hệ thống đánh giá tự động nhằm hỗ trợ hoạt động giảng dạy lập trình ở bậc đại học. Nền tảng được xây dựng với định hướng trọng tâm vào khả năng tiếp cận, khả năng mở rộng và hiệu quả sư phạm, đặc biệt phù hợp với các cơ sở giáo dục hoạt động trong điều kiện hạn chế về hạ tầng kỹ thuật. Thông qua việc kết hợp phản hồi tức thì, giao diện thân thiện với người dùng và cơ chế thực thi mã an toàn trong môi trường container cách ly, hệ thống cho phép sinh viên chủ động học tập theo tiến độ cá nhân và đồng thời giúp giảm đáng kể khối lượng công việc chấm điểm cho giảng viên.

Kết quả đánh giá thực nghiệm với 50 sinh viên đại học tại Trường Đại học Tân Trào đã khẳng định những lợi ích rõ rệt của hệ thống trong việc nâng cao kết quả học tập, tăng cường động lực học tập và cải thiện chất lượng giảng dạy lập trình nói chung. Sinh viên thể hiện mức độ hài lòng và sự tự tin cao hơn trong việc viết và gỡ lỗi chương trình, trong khi giảng viên được hỗ trợ đáng kể nhờ quy trình chấm điểm hiệu quả và minh bạch hơn. Đồng thời, phân tích nhật ký hệ thống cũng xác nhận rằng sinh viên đã chủ động tận dụng vòng phản hồi liên tục để cải thiện lời giải, qua đó làm nổi bật giá trị của việc đánh giá hình thành tức thời trong giáo dục lập trình.

Đóng góp của nghiên cứu này thể hiện trên hai phương diện chính. Thứ nhất, nghiên cứu cung cấp một giải pháp công nghệ khả thi, đáp ứng được các thách thức đặc thù mà các trường đại học tại khu vực đang phát triển thường gặp phải - đặc biệt là nhu cầu về một hệ thống học lập trình nhẹ, chạy trên trình duyệt và không phụ thuộc vào hạ tầng phức tạp. Thứ hai, nghiên cứu chứng minh rằng việc tích hợp có chủ đích giữa phản hồi tự động và phân tích học tập có thể thúc đẩy môi trường học tương tác, linh hoạt và phù hợp với người học mới bắt đầu.

Trong thời gian tới, nhiều hướng phát triển tiềm năng được đặt ra nhằm nâng cao hiệu quả và giá trị ứng dụng của nền tảng. Một trong những cải tiến quan trọng là tích hợp mô-đun gợi ý bài luyện tập cá nhân hóa, sử dụng các thuật toán học máy để phân tích hiệu suất của từng người học và đề xuất các bài tập phù hợp dựa trên khoảng trống kiến thức và lộ trình học tập. Điều

này sẽ giúp khắc phục sự chênh lệch về trình độ giữa các sinh viên và thúc đẩy các lộ trình học thích ứng. Ngoài ra, việc ứng dụng phân tích hành vi học tập - như theo dõi mẫu chỉnh sửa mã, mức độ sử dụng gợi ý và chiến lược sửa lỗi - có thể cung cấp những hiểu biết sâu hơn về hành vi giải quyết vấn đề và hỗ trợ can thiệp sớm đối với người học có nguy cơ gặp khó khăn.

Một số hướng mở rộng khác bao gồm phát triển tính năng lập trình hợp tác, tích hợp các yếu tố trò chơi hóa để duy trì động lực học tập, bổ sung hỗ trợ đa ngôn ngữ nhằm mở rộng khả năng tiếp cận và kết nối hệ thống với nền tảng quản lý học tập (LMS) của nhà trường để tối ưu hóa công tác quản lý khóa học. Đồng thời, một nghiên cứu theo chiều dọc với quy mô lớn hơn, bao gồm nhiều cơ sở đào tạo và đa dạng đối tượng người học cũng đang được lên kế hoạch, nhằm đánh giá tính khái quát và tác động dài hạn của nền tảng đối với giáo dục lập trình trong các bối cảnh học tập khác nhau.

Tổng kết lại, nghiên cứu này đóng góp một cách tiếp cận mang tính thực tiễn, dựa trên dữ liệu và có cơ sở sư phạm vững chắc nhằm cải thiện chất lượng giáo dục lập trình trong giáo dục đại học. Bằng cách khai thác hiệu quả công nghệ tự động hóa và phản hồi thông minh, nền tảng không chỉ nâng cao kết quả học tập của sinh viên mà còn hỗ trợ giảng viên trong việc triển khai giảng dạy lập trình hiệu quả, có thể mở rộng và bao trùm - điều đặc biệt cần thiết trong thời đại chuyển đổi số của giáo dục.

TÀI LIỆU THAM KHẢO

- [1] P.J. Guo, J. Kim, R. Rubin (2014), "How video production affects student engagement: An empirical study of MOOC videos", *Proceedings of The First ACM Conference on Learning at Scale*, pp.41-50.
- [2] P. Iantola, T. Ahoniemi, V. Karavirta, et al. (2010), "Review of recent systems for automatic assessment of programming assignments", *Proceedings of The 10th Koli Calling International Conference on Computing Education Research*, pp.86-93, DOI: 10.1145/1930464.1930480.
- [3] H. Keuning, J. Jeuring, B. Heeren (2018), "A systematic literature review of automated feedback generation for programming exercises", *ACM Transactions on Computing Education (TOCE)*, **19**(1), pp.1-43, DOI: 10.1145/3231711.
- [4] R. Zhi, A. Lishinski, R. Enbody (2019), "A novel AST-based programming assignment grader for Python", *Proceedings of The 50th ACM Technical Symposium on Computer Science Education*, pp.1084-1090, DOI: 10.1145/3287324.3287427.
- [5] T.A. Dung (2021), "Application of automatic grading system in teaching C++ programming at university", *University of Danang - Journal of Science and Technology*, **19**(3), pp.56-64 (in Vietnamese).
- [6] N.T. Chinh, N.T.T. Huyen, V.P. Duy (2022), "Designing an automatic grading system to improve the self-learning ability of students at Hanoi National University of Education", *HNUE Journal of Science*, **67**(2), pp.206-217 (in Vietnamese).
- [7] L. Hoan, N.Q. Anh, P.N. Linh (2023), "Applying automatic code grading system for teaching programming languages in universities", *TNU Journal of Science and Technology*, **228**(15), pp.224-234 (in Vietnamese).