HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐỖ HƯƠNG HÀ

PHÁT TRIỂN HỆ THỐNG HỖ TRỢ CỐ VẤN HỌC TẬP TRONG CÔNG TÁC QUẢN LÝ SINH VIÊN

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

HÀ NỘI - 2026

Đề án tốt nghiệp được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: PGS.TS Ngô Quốc Dũng

(Ghi rõ học hàm, học vị)

Phản biện 1: PGS.TS Nguyễn Trọng Khánh

Phản biện 2: TS. Đinh Trường Duy

Đề án tốt nghiệp sẽ được bảo vệ trước Hội đồng chấm đề án tốt nghiệp thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ..... giờ ..... ngày ..... tháng .... năm .....

Có thể tìm hiểu đề án tốt nghiệp tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông.

MỞ ĐẦU

Với sự bùng nổ của công nghệ thông tin và dữ liệu trong những năm gần đây, các tổ chức, đặc biệt là các cơ sở giáo dục, đang đứng trước một cuộc chuyển đổi số sâu sắc. Cuộc Cách mạng công nghiệp 4.0 không chỉ đặt ra yêu cầu về việc số hóa các quy trình hiện tại, mà còn buộc giáo dục đại học phải thay đổi tư duy, từ việc quản lý thông tin theo phương pháp truyền thống sang hướng tiếp cận data-driven (hướng dữ liệu). Một trong những lĩnh vực quan trọng nhất trong giáo dục đại học là công tác cố vấn học tập (CVHT), nơi mà các giảng viên cần có cái nhìn sâu sắc, kịp thời về tình trạng học tập, rèn luyện, và tiến độ học vụ của sinh viên để có thể hỗ trợ và can thiệp sớm khi cần thiết.

Tuy nhiên, hiện nay, thông tin sinh viên tại các trường đại học vẫn tồn tại ở trạng thái phân tán, rời rạc, lưu trữ dưới nhiều hình thức khác nhau (Excel, hệ thống nội bộ, báo cáo thủ công) và không được đồng bộ với nhau. Tình trạng này không chỉ gây khó khăn trong việc tổng hợp, phân tích dữ liệu mà còn dẫn đến các sai sót do lỗi con người, khiến cho công tác cố vấn học tập bị chậm chễ và thiếu hiệu quả. Thêm vào đó, mặc dù dữ liệu thô có sẵn, nhưng nhiều trường đại học vẫn chưa được trang bị các hệ thống phân tích học tập (learning analytics) đủ chiều sâu để trích xuất thông tin hữu ích từ những dữ liệu này. Sự thiếu hụt công cụ này làm hạn chế khả năng nhận diện sớm sinh viên có nguy cơ, dự báo xu hướng hành vi, và thực hiện các can thiệp kịp thời.

Nhằm giải quyết những vấn đề này, đề tài này tập trung vào việc phát triển một hệ thống Data Pipeline hoàn chỉnh, tự động hóa quy trình từ thu thập, lưu trữ, xử lý dữ liệu đến trực quan hóa thông qua các bảng chỉ số (dashboard). Hệ thống sẽ áp dụng kiến trúc Medallion (Bronze-Silver-Gold) kết hợp với các công nghệ hiện đại như Apache Airflow cho orchestration, MinIO cho lưu trữ data lake, ClickHouse cho kho dữ liệu phân tích, và Apache Superset cho trực quan hóa dữ liệu. Cách tiếp cận này không chỉ giúp khép kín toàn bộ quy trình xử lý dữ liệu, mà còn tạo nền tảng vững chắc cho các ứng dụng phân tích nâng cao và học máy trong tương lai.

Đề tài này được tổ chức thành ba chương chính. Chương I trình bày bối cảnh, thực trạng, tính cấp thiết, cũng như phạm vi và đối tượng nghiên cứu của đề tài. Chương II đi sâu vào các kiến thức nền tảng về Data Pipeline, các kiến trúc pipeline hiện đại, cũng như các công nghệ được lựa chọn để xây dựng hệ thống. Chương III trình bày quá trình xây dựng hệ thống từ thiết lập môi trường triển khai, cấu hình các thành phần, cài đặt các module xử lý dữ liệu, cho đến đánh giá hiệu năng và tính hiệu quả của hệ thống. Thông qua việc triển khai thực tế này, đề tài không chỉ cung cấp một giải pháp công nghệ cụ thể mà còn đóng góp vào quá trình chuyển đổi số trong giáo dục đại học, giúp tối ưu hóa công tác quản lý và hỗ trợ sinh viên.

CHƯƠNG I: GIỚI THIỆU VÀ ĐẶT VẤN ĐỀ

1.1. Bối cảnh đề tài

Trong bối cảnh Cách mạng công nghiệp 4.0, giáo dục đại học đang chịu áp lực phải chuyển đổi số toàn diện, hướng tới mô hình quản trị và đào tạo dựa trên dữ liệu. Tuy nhiều trường đã triển khai các hệ thống LMS và công cụ trực tuyến, dữ liệu vẫn bị phân tán, thiếu kết nối và chưa được khai thác hiệu quả. Điều này ảnh hưởng trực tiếp đến công tác cố vấn học tập – hoạt động đòi hỏi dữ liệu phải thống nhất, chính xác và cập nhật liên tục. Vì vậy, việc ứng dụng các giải pháp phân tích dữ liệu và quy trình xử lý tự động (data pipeline) trở thành xu hướng tất yếu nhằm nâng cao hiệu quả hỗ trợ sinh viên.

1.2. Thực trạng

Thực trạng tại các cơ sở giáo dục đại học cho thấy dữ liệu sinh viên hiện chưa được tổ chức tập trung, còn rời rạc ở nhiều hệ thống và định dạng khác nhau. Công tác cố vấn học tập chủ yếu thực hiện thủ công, tốn thời gian và dễ phát sinh sai sót, khiến việc hỗ trợ sinh viên chưa kịp thời và khó phát hiện sớm các rủi ro học vụ. Bên cạnh đó, việc thiếu các hệ thống phân tích học tập chuyên sâu làm hạn chế khả năng khai thác dữ liệu cho dự báo và nhận diện xu hướng, trong khi mức độ triển khai các mô hình data pipeline tại Việt Nam vẫn còn thấp.

1.3. Tính cấp thiết nghiên cứu của đề tài

Việc xây dựng hệ thống Data Pipeline giúp tự động hóa toàn bộ quy trình từ thu thập, xử lý, phân tích đến trực quan hóa dữ liệu, góp phần hiện thực hóa mô hình đại học số. Kiến trúc tích hợp Data Lake – ETL – Data Warehouse giúp chuẩn hóa và tập trung dữ liệu, giảm sai sót và nâng cao hiệu quả xử lý. Các dashboard trực quan hỗ trợ giảng viên cố vấn theo dõi kịp thời tình trạng học tập và rèn luyện của sinh viên, tạo cơ sở cho các can thiệp sớm. Đồng thời, hệ thống dữ liệu này là nền tảng quan trọng cho việc ứng dụng các mô hình học máy trong tương lai.

1.4. Phạm vi và đối tượng trong nghiên cứu

Đề tài tập trung nghiên cứu các luồng dữ liệu liên quan đến kết quả học tập và điểm rèn luyện của sinh viên, cùng các quy trình kỹ thuật xây dựng data pipeline trong giáo dục đại học. Phạm vi kỹ thuật giới hạn ở việc thiết kế và thử nghiệm kiến trúc

gồm hồ dữ liệu, ETL, kho dữ liệu và dashboard. Dữ liệu sử dụng là dữ liệu giả lập hoặc đã phi định danh nhằm đảm bảo bảo mật, và việc kết nối trực tiếp với hệ thống đào tạo hiện hành được xác định là hướng phát triển trong tương lai.

CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1. Phân tích nghiệp vụ công tác cố vấn học tập (CVHT)

Cố vấn học tập (CVHT) giữ vai trò đặc biệt quan trọng trong công tác quản lý, theo dõi và hỗ trợ sinh viên trong suốt quá trình đào tạo tại các cơ sở giáo dục đại học. CVHT không chỉ là cầu nối giữa nhà trường và sinh viên mà còn trực tiếp nắm bắt tình hình học tập, rèn luyện, tư tưởng cũng như các khó khăn phát sinh của sinh viên để kịp thời tư vấn, định hướng và hỗ trợ. Trong bối cảnh đào tạo theo hệ thống tín chỉ, khi sinh viên có quyền chủ động cao trong việc đăng ký học phần và xây dựng kế hoạch học tập cá nhân, vai trò của CVHT ngày càng trở nên quan trọng và đòi hỏi chuyên môn quản lý ngày càng cao.

Nghiệp vụ công tác CVHT đặt ra nhiều yêu cầu như theo dõi tiến độ học tập của từng sinh viên, phát hiện sớm các trường hợp có nguy cơ học vụ, hỗ trợ sinh viên điều chỉnh kế hoạch học tập phù hợp, đánh giá kết quả rèn luyện theo quy định của nhà trường, đồng thời tổng hợp báo cáo phục vụ công tác quản lý của khoa và nhà trường. Tuy nhiên, với số lượng sinh viên lớn và dữ liệu phát sinh đa dạng, công tác CVHT trên thực tế gặp nhiều khó khăn trong việc tổng hợp, phân tích thông tin và đưa ra cảnh báo kịp thời.

Trong nghiệp vụ CVHT có sự tham gia của nhiều tác nhân khác nhau. Sinh viên là đối tượng trung tâm, vừa tham gia các hoạt động học tập và rèn luyện, vừa là nguồn phát sinh dữ liệu cho hệ thống quản lý. CVHT là tác nhân chính sử dụng thông tin để thực hiện các nghiệp vụ theo dõi, đánh giá và hỗ trợ sinh viên, đồng thời lập các báo cáo định kỳ. Giảng viên đóng vai trò cung cấp dữ liệu học tập và phản hồi chuyên môn thông qua việc cập nhật điểm số, tình trạng chuyên cần và nhận xét thái độ học tập của sinh viên.

Luồng thông tin trong công tác quản lý sinh viên bao gồm ba luồng chính: luồng thông tin học tập, luồng thông tin rèn luyện và luồng thông tin khảo sát – phản hồi. Các luồng thông tin này phản ánh toàn diện kết quả học tập, ý thức rèn luyện cũng như chất lượng giảng dạy và môi trường học tập. Việc khai thác hiệu quả các luồng dữ liệu này có ý nghĩa quan trọng trong việc đánh giá tình trạng sinh viên và hỗ trợ ra quyết định quản lý.

Mặc dù có vai trò quan trọng, công tác CVHT hiện nay vẫn còn tồn tại nhiều hạn chế. Dữ liệu phục vụ CVHT thường bị phân tán ở nhiều hệ thống khác nhau, thiếu sự tích hợp và đồng bộ, gây khó khăn trong việc tổng hợp và khai thác thông tin. Việc theo dõi học tập và rèn luyện chủ yếu dựa trên báo cáo thủ công hoặc bán tự động, dẫn đến thiếu tính kịp thời và không phản ánh đầy đủ diễn biến trong suốt quá trình học tập của sinh viên. Bên cạnh đó, cơ chế cảnh báo học vụ và rèn luyện còn mang tính thụ động, chủ yếu được kích hoạt khi sinh viên đã vi phạm các ngưỡng quy định. Ngoài ra, CVHT chưa được trang bị đầy đủ các công cụ phân tích và trực quan hóa dữ liệu, làm hạn chế khả năng đánh giá toàn diện và hiệu quả hỗ trợ sinh viên.

Từ những bất cập trên, nhu cầu xây dựng một hệ thống hỗ trợ CVHT dựa trên dữ liệu trở nên cấp thiết. Hệ thống cần được thiết kế theo hướng tích hợp dữ liệu từ nhiều nguồn khác nhau như dữ liệu học tập, dữ liệu rèn luyện và dữ liệu khảo sát; đồng thời áp dụng các giải pháp ETL kết hợp với kho dữ liệu nhằm đảm bảo dữ liệu được thu thập, xử lý và lưu trữ một cách thống nhất, chính xác và nhất quán. Trên nền tảng đó, hệ thống hỗ trợ CVHT thông qua các công cụ phân tích, báo cáo và dashboard trực quan, góp phần nâng cao khả năng theo dõi, cảnh báo sớm và hỗ trợ ra quyết định trong quản lý sinh viên. Việc xây dựng hệ thống này không chỉ đáp ứng yêu cầu thực tiễn mà còn phù hợp với xu hướng chuyển đổi số trong giáo dục đại học hiện nay.

2.2. Phân tích các use case chính của hệ thống hỗ trợ cố vấn học tập

2.2.1. Use case: Theo dõi tình trạng học tập và cảnh báo học vụ

Use case này cho phép hệ thống thu thập và phân tích dữ liệu học tập của sinh viên như điểm số, số tín chỉ và tình trạng chuyên cần nhằm phát hiện sớm các trường hợp có nguy cơ học vụ. Dựa trên các ngưỡng cảnh báo được cấu hình, hệ thống phân loại mức độ rủi ro và cung cấp danh sách sinh viên cần cảnh báo cho CVHT. Kết quả phân tích hỗ trợ CVHT chủ động tư vấn và can thiệp kịp thời.

2.2.2. Use case: Theo dõi điểm rèn luyện và cảnh báo rèn luyện

Use case theo dõi điểm rèn luyện tập trung vào việc tổng hợp và đánh giá điểm rèn luyện của sinh viên theo từng học kỳ. Hệ thống xác định các trường hợp chưa đạt yêu cầu hoặc còn thiếu tiêu chí rèn luyện, từ đó hỗ trợ CVHT theo dõi, nhắc nhở và hướng dẫn sinh viên hoàn thiện các yêu cầu theo quy định.

2.2.3. Use case: Gửi email tự động khi có cảnh báo

Use case này hỗ trợ việc truyền tải thông tin cảnh báo học vụ và rèn luyện một cách kịp thời và nhất quán. Khi phát sinh cảnh báo, hệ thống tự động tạo và gửi email đến sinh viên và CVHT, đồng thời lưu trữ lịch sử cảnh báo nhằm phục vụ công tác theo dõi và quản lý.

2.2.4. Use case: Thống kê và báo cáo tình trạng lớp

Use case thống kê và báo cáo cho phép tổng hợp dữ liệu học tập và rèn luyện theo lớp, học kỳ hoặc thời gian. Các báo cáo và dashboard trực quan hỗ trợ CVHT và lãnh đạo khoa, nhà trường đánh giá tổng thể tình trạng sinh viên và đưa ra các quyết định quản lý dựa trên dữ liệu.

CHƯƠNG III: XÂY DỰNG VÀ ĐÁNH GIÁ HỆ THỐNG

3.1. Kiến thức tổng quan

3.1.1. Data Pipeline

Data Pipeline là kiến trúc hệ thống cho phép tự động hóa quá trình thu thập, xử lý và di chuyển dữ liệu từ nhiều nguồn khác nhau đến các hệ thống đích nhằm phục vụ lưu trữ, phân tích và hỗ trợ ra quyết định. Trong bối cảnh dữ liệu ngày càng gia tăng về khối lượng và độ phức tạp, Data Pipeline đóng vai trò trung tâm trong việc khai thác hiệu quả giá trị của dữ liệu thông qua các quy trình xử lý được chuẩn hóa và tự động.

Về bản chất, Data Pipeline bao gồm chuỗi các bước xử lý liên tiếp như trích xuất, làm sạch, chuyển đổi và lưu trữ dữ liệu theo một quy trình khép kín. Một Data Pipeline hiệu quả cần đảm bảo tính nhất quán dữ liệu, khả năng mở rộng, cơ chế xử lý lỗi, phục hồi và giám sát. Kiến trúc tổng quát thường gồm các giai đoạn: thu thập dữ liệu, lưu trữ dữ liệu thô trong Data Lake, xử lý dữ liệu thông qua ETL/ELT, tổ chức dữ liệu trong Data Warehouse và trực quan hóa thông tin bằng các công cụ BI.

Data Pipeline có thể được triển khai theo hai phương thức chính là xử lý theo lô (batch processing) và xử lý theo luồng (stream processing). Trong lĩnh vực giáo dục, đặc biệt với các hệ thống hỗ trợ cố vấn học tập, xử lý theo lô thường được ưu tiên do dữ liệu học tập được cập nhật theo chu kỳ và không yêu cầu xử lý thời gian thực.

Việc triển khai Data Pipeline mang lại nhiều lợi ích như giảm thiểu thao tác thủ công, nâng cao chất lượng và tính nhất quán của dữ liệu, tăng hiệu quả xử lý dữ liệu lớn và hỗ trợ giám sát hệ thống. Tuy nhiên, quá trình triển khai cũng gặp phải một số thách thức như chuẩn hóa dữ liệu đa nguồn, truy vết dòng dữ liệu, xử lý lỗi và tối ưu hiệu năng khi quy mô dữ liệu tăng.

Trong giáo dục đại học, Data Pipeline được ứng dụng để tích hợp dữ liệu từ nhiều hệ thống nhằm phục vụ phân tích học tập, phát hiện sớm sinh viên có nguy cơ học vụ và nâng cao hiệu quả quản lý đào tạo. Đối với đề tài này, Data Pipeline đóng vai trò nền tảng trong việc tự động hóa quy trình xử lý dữ liệu học tập và rèn luyện của sinh viên, từ thu thập, xử lý ETL, lưu trữ tập trung đến trực quan hóa trên dashboard, qua đó hỗ trợ hiệu quả cho công tác cố vấn học tập.

3.1.2. Các kiến trúc pipeline hiện đại

Các kiến trúc Data Pipeline hiện đại phổ biến gồm Lambda Architecture, Kappa Architecture và Medallion Architecture, mỗi kiến trúc được thiết kế để đáp ứng các yêu cầu xử lý dữ liệu khác nhau về độ trễ, quy mô và mức độ phức tạp.

Lambda Architecture bao gồm ba tầng: Batch Layer, Speed Layer và Serving Layer. Kiến trúc này kết hợp xử lý theo lô và xử lý thời gian thực nhằm đảm bảo đồng thời độ chính xác cao và độ trễ thấp. Batch Layer xử lý toàn bộ dữ liệu lịch sử để tạo kết quả chính xác, trong khi Speed Layer xử lý dữ liệu mới phát sinh theo thời gian thực. Ưu điểm của Lambda là khả năng chịu lỗi tốt và cân bằng giữa tính chính xác và tốc độ. Tuy nhiên, việc duy trì hai pipeline xử lý song song làm tăng độ phức tạp trong phát triển, bảo trì và dễ gây không nhất quán dữ liệu tạm thời.

Kappa Architecture được đề xuất nhằm đơn giản hóa Lambda Architecture bằng cách loại bỏ Batch Layer và chỉ sử dụng xử lý theo luồng. Mọi dữ liệu được xem như các sự kiện trong một stream, xử lý thông qua các công nghệ stream processing. Kiến trúc này có ưu điểm là đơn giản, giảm chi phí bảo trì và đảm bảo tính nhất quán logic xử lý. Tuy nhiên, Kappa không phù hợp với tất cả bài toán, đặc biệt là các phép tính phức tạp hoặc yêu cầu truy cập toàn bộ dữ liệu lịch sử.

Medallion Architecture (Bronze–Silver–Gold) là kiến trúc được sử dụng rộng rãi trong các hệ thống Data Lakehouse hiện đại. Dữ liệu được tổ chức theo mức độ tinh chế tăng dần: Bronze lưu trữ dữ liệu thô, Silver thực hiện làm sạch và chuẩn hóa, Gold tối ưu dữ liệu cho phân tích và báo cáo. Kiến trúc này giúp tách biệt rõ ràng các giai đoạn xử lý, hỗ trợ cả batch và stream processing, dễ triển khai và linh hoạt trong mở rộng. Hạn chế của Medallion là có thể gia tăng chi phí lưu trữ và không phù hợp với các hệ thống yêu cầu độ trễ cực thấp.

So sánh tổng thể cho thấy Lambda phù hợp với hệ thống phức tạp yêu cầu xử lý real-time và batch song song; Kappa phù hợp với hệ thống ưu tiên stream processing; trong khi Medallion Architecture cân bằng tốt giữa tính đơn giản, linh hoạt và khả năng mở rộng, đặc biệt phù hợp với môi trường phân tích dữ liệu đa mục đích.

Trong phạm vi nghiên cứu này, Medallion Architecture được lựa chọn do dữ liệu học tập và rèn luyện của sinh viên chủ yếu được xử lý theo chu kỳ, không yêu cầu

thời gian thực. Việc tổ chức dữ liệu theo các tầng Bronze–Silver–Gold giúp quản lý hiệu quả quá trình chuyển đổi từ dữ liệu thô đến báo cáo phân tích, đồng thời tạo điều kiện thuận lợi cho việc mở rộng và điều chỉnh hệ thống trong tương lai.

3.1.3. Data Lake và Data Warehouse

3.1.3.1. Data Lake

Data Lake là mô hình lưu trữ dữ liệu tập trung cho phép lưu giữ dữ liệu thô ở định dạng nguyên bản mà không cần định nghĩa schema trước. Với nguyên tắc schema-on-read, Data Lake mang lại sự linh hoạt cao trong việc tiếp nhận và khai thác dữ liệu đa dạng, bao gồm dữ liệu có cấu trúc, bán cấu trúc và phi cấu trúc. Các nền tảng phổ biến để triển khai Data Lake gồm HDFS và các hệ thống object storage trên nền tảng đám mây hoặc on-premise. Ngoài chức năng lưu trữ, Data Lake còn tích hợp các thành phần quản lý metadata, data catalog và cơ chế bảo mật nhằm hỗ trợ truy xuất, kiểm soát và khai phá dữ liệu hiệu quả.

Trong hệ thống pipeline phục vụ cố vấn học tập, Data Lake đóng vai trò lưu trữ dữ liệu gốc như các file Excel từ các đơn vị chức năng. Việc này đảm bảo khả năng truy vết, kiểm toán dữ liệu và cho phép xử lý lại dữ liệu với các logic mới trong tương lai. Đề tài lựa chọn MinIO làm giải pháp Data Lake nhờ khả năng triển khai linh hoạt, chi phí thấp và tương thích với chuẩn S3.

3.1.3.2. Data Warehouse

Data Warehouse là kho dữ liệu được thiết kế chuyên biệt cho phân tích, với dữ liệu đã được làm sạch, tích hợp và tổ chức theo schema xác định trước (schema-on-write). Đặc trưng của Data Warehouse là hướng chủ đề, bất biến theo thời gian và tối ưu cho các truy vấn phân tích phức tạp. Dữ liệu thường được tổ chức theo các mô hình như star schema hoặc snowflake schema, bao gồm các bảng fact và dimension nhằm phục vụ hiệu quả cho báo cáo và phân tích nghiệp vụ.

Các hệ thống Data Warehouse hiện đại, đặc biệt trên nền tảng cloud hoặc kiến trúc columnar storage, cho phép xử lý nhanh các truy vấn trên khối lượng dữ liệu lớn. So với Data Lake, Data Warehouse có hiệu năng truy vấn cao hơn cho các phân tích định sẵn nhưng kém linh hoạt hơn trong việc tiếp nhận dữ liệu đa dạng.

Trong đề tài, Data Warehouse được sử dụng để lưu trữ dữ liệu sinh viên đã được chuẩn hóa, phục vụ các phân tích học vụ và rèn luyện như tính GPA, theo dõi tiến độ học tập và xác định sinh viên có nguy cơ cảnh báo. ClickHouse được lựa chọn nhờ khả năng truy vấn nhanh, lưu trữ dạng cột và khả năng mở rộng tốt.

3.1.4. ETL (Extract, Transform, Load)

ETL (Extract – Transform – Load) là quy trình trung tâm trong Data Pipeline, đóng vai trò kết nối giữa các nguồn dữ liệu và hệ thống lưu trữ phục vụ phân tích. Mục tiêu của ETL là tích hợp, chuẩn hóa và đảm bảo dữ liệu sẵn sàng cho các hoạt động phân tích và ra quyết định.

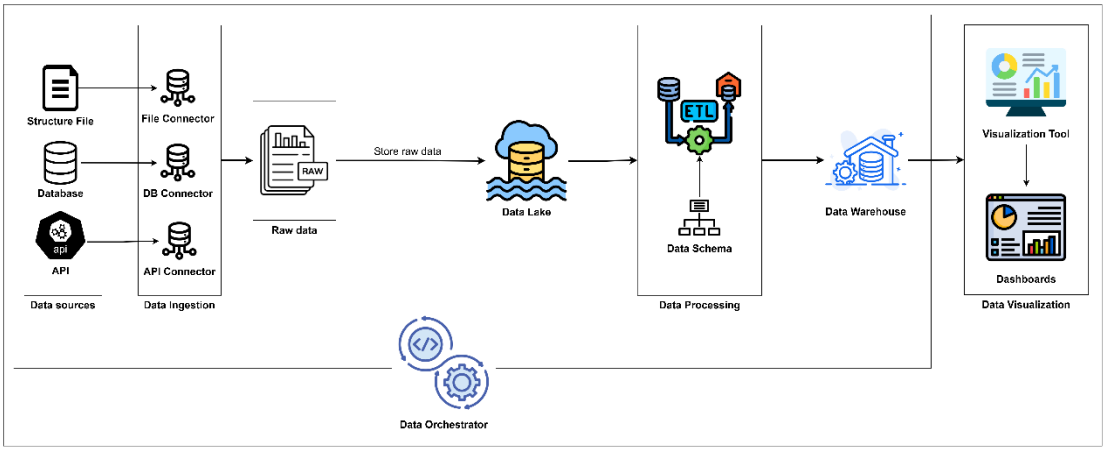
Quy trình ETL gồm ba giai đoạn chính. Giai đoạn Extract thực hiện trích xuất dữ liệu từ nhiều nguồn khác nhau, ưu tiên phương pháp trích xuất tăng cường nhằm giảm tải cho hệ thống nguồn và tối ưu thời gian xử lý, đồng thời yêu cầu đảm bảo an toàn kết nối và toàn vẹn dữ liệu. Giai đoạn Transform tập trung vào làm sạch, tích hợp và chuyển đổi dữ liệu, bao gồm xử lý dữ liệu thiếu hoặc trùng lặp, chuẩn hóa định dạng, tính toán các chỉ số nghiệp vụ và bổ sung thông tin cần thiết. Đây là giai đoạn quyết định chất lượng dữ liệu đầu ra. Giai đoạn Load thực hiện nạp dữ liệu đã xử lý vào hệ thống đích, thường là Data Warehouse, với các chiến lược tải phù hợp nhằm đảm bảo hiệu năng, tính ổn định và khả năng phục hồi của hệ thống.

Bên cạnh ETL, mô hình ELT ngày càng được sử dụng phổ biến trong các hệ thống kho dữ liệu hiện đại nhờ khả năng tận dụng năng lực xử lý của nền tảng đích và tăng tính linh hoạt trong phân tích.

Trong đề tài này, ETL được áp dụng để xử lý dữ liệu học tập và rèn luyện của sinh viên, từ trích xuất dữ liệu thô trong Data Lake, chuyển đổi và tính toán các chỉ số học vụ, đến nạp dữ liệu vào Data Warehouse theo mô hình star schema. Quy trình được tự động hóa và điều phối bằng Apache Airflow, đảm bảo tính ổn định và khả năng giám sát cho hệ thống hỗ trợ cố vấn học tập.

3.2. Phân tích và thiết kế hệ thống

3.2.1. Kiến trúc hệ thống tổng quan



Hình 1. Kiến trúc tổng quan hệ thống

Kiến trúc của hệ thống tuân theo mô hình hiện đại (Modern Data Architecture), bao gồm các tầng: Nguồn dữ liệu (Data Sources), Tiếp nhận dữ liệu (Data Ingestion), Lưu trữ dữ liệu thô (Data Lake), Xử lý – chuyển đổi dữ liệu (Data Processing / ETL), Kho dữ liệu (Data Warehouse) và Trực quan hóa dữ liệu (Data Visualization). Toàn bộ quy trình được điều phối bởi Data Orchestrator nhằm đảm bảo dữ liệu được di chuyển và xử lý tự động, nhất quán và hiệu quả.

- Hệ thống hỗ trợ đa nguồn dữ liệu:
  - File dữ liệu cấu trúc như CSV, Excel, JSON…
  - Cơ sở dữ liệu như MySQL, PostgreSQL, SQL Server…
  - API cung cấp dữ liệu theo thời gian thực hoặc theo yêu cầu.

Tầng này sử dụng các connector để thu thập dữ liệu từ từng loại nguồn:

- File Connector giúp đọc dữ liệu từ hệ thống file.
- DB Connector kết nối trực tiếp tới các hệ quản trị cơ sở dữ liệu.
- API Connector tiếp nhận dữ liệu thông qua REST API/HTTP.

Dữ liệu sau khi được ingest sẽ được lưu trữ trong Data Lake ở dạng nguyên bản. Data Lake đóng vai trò nơi tập trung dữ liệu, giúp hệ thống dễ dàng mở rộng và tái sử dụng.

Tầng xử lý thực hiện các nhiệm vụ:

- Làm sạch dữ liệu (cleaning).

– Biến đổi – chuẩn hóa dữ liệu (transformation).

– Ánh xạ vào mô hình dữ liệu (Data Schema) (không bắt buộc).

– Tải dữ liệu vào kho dữ liệu theo quy trình ETL/ELT.

Sau khi được chuẩn hóa, dữ liệu được lưu vào Data Warehouse, đóng vai trò:

– Lưu trữ dữ liệu sạch, có cấu trúc.

– Tối ưu cho truy vấn phân tích.

– Là nền tảng cho việc xây dựng báo cáo và dashboard.

Kho dữ liệu có thể theo mô hình star schema, snowflake, hoặc các mô hình hiện đại như Data Vault tùy theo yêu cầu hệ thống.

Tầng cuối cùng sử dụng các công cụ trực quan hóa để:

– Kết nối với Data Warehouse.

– Tạo dashboard.

– Hiển thị chỉ số và phân tích dữ liệu phục vụ nghiệp vụ và ra quyết định.

Công cụ Orchestrator đảm nhiệm:

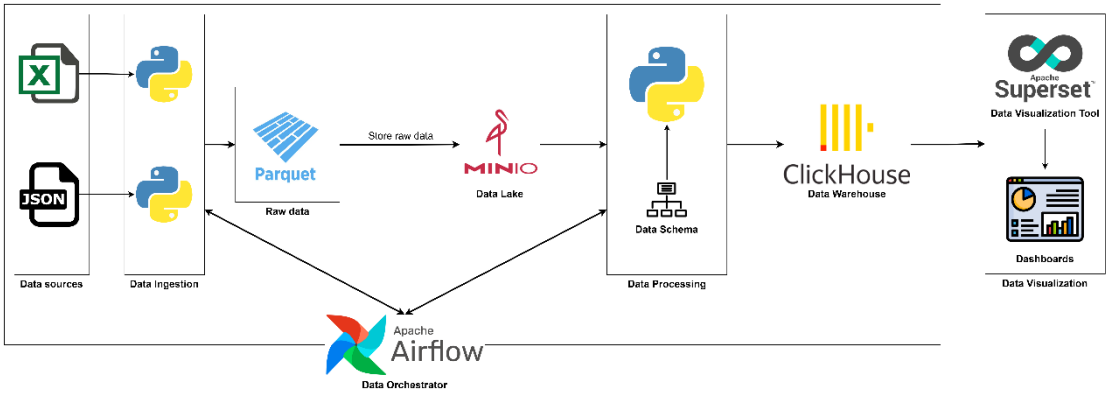
– Lập lịch công việc (scheduler).

– Điều phối toàn bộ pipeline ingest → xử lý → load → kho dữ liệu.

– Giám sát luồng dữ liệu và xử lý lỗi.

– Tự động hóa toàn bộ hệ thống.

3.2.2. Thiết kế pipeline



Hình 2. Thiết kế pipeline chi tiết

Hệ thống được xây dựng theo kiến trúc xử lý dữ liệu hiện đại, cho phép tự động hóa toàn bộ quy trình thu thập, lưu trữ, xử lý và trực quan hóa dữ liệu theo một luồng khép kín. Kiến trúc bao gồm các thành phần chính: nguồn dữ liệu, tầng tiếp nhận dữ liệu, Data Lake, tầng xử lý dữ liệu (ETL/ELT), Data Warehouse, công cụ trực quan hóa và tầng điều phối dữ liệu, đảm bảo khả năng mở rộng, tính nhất quán và sẵn sàng cho phân tích.

Nguồn dữ liệu đầu vào chủ yếu là các tệp Excel và JSON, phản ánh đặc trưng dữ liệu phân tán và thiếu chuẩn hóa trong môi trường giáo dục. Dữ liệu được tiếp nhận và xử lý sơ bộ bằng các script Python, sau đó chuyển sang định dạng Parquet để tối ưu lưu trữ và hiệu năng phân tích. Dữ liệu thô được lưu trữ tập trung trong Data Lake triển khai trên MinIO, đảm bảo tính toàn vẹn và khả năng tái xử lý.

Tầng xử lý dữ liệu thực hiện các chức năng ETL/ELT, bao gồm làm sạch, chuẩn hóa, tính toán chỉ số và tổ chức dữ liệu theo mô hình phân tích. Dữ liệu sau xử lý được nạp vào kho dữ liệu ClickHouse – hệ quản trị cơ sở dữ liệu dạng cột có hiệu năng cao, phù hợp cho các truy vấn phân tích và tổng hợp dữ liệu lớn.

Tầng trực quan hóa sử dụng Apache Superset để xây dựng các dashboard và báo cáo tương tác, phục vụ theo dõi kết quả học tập, rèn luyện và cảnh báo học vụ cho công tác cố vấn học tập. Toàn bộ pipeline được điều phối và giám sát bởi Apache Airflow thông qua các DAG, cho phép tự động hóa, lập lịch linh hoạt và xử lý lỗi hiệu quả.

Tổng thể kiến trúc đảm bảo dữ liệu được xử lý theo một chu trình thống nhất từ ingest đến trực quan hóa, đáp ứng tốt yêu cầu quản lý sinh viên trong giáo dục đại học, đồng thời tạo nền tảng cho việc mở rộng các chức năng phân tích nâng cao trong tương lai.

3.2.3. Luồng dữ liệu

Luồng dữ liệu của hệ thống được thiết kế theo mô hình tuyến tính – phân tầng, đảm bảo dữ liệu được thu thập, lưu trữ, xử lý và trực quan hóa theo một quy trình khép kín và thống nhất. Toàn bộ pipeline được tự động hóa và điều phối bởi Apache Airflow, giúp hệ thống vận hành ổn định và hạn chế thao tác thủ công.

Dữ liệu đầu vào chủ yếu đến từ các tệp Excel và JSON, phản ánh đặc trưng dữ liệu phân tán trong môi trường giáo dục đại học. Dữ liệu được tiếp nhận thông qua tầng

Data Ingestion, nơi các chương trình Python thực hiện kiểm tra cấu trúc và chuẩn hóa ban đầu nhằm đảm bảo chất lượng dữ liệu trước khi đưa vào pipeline.

Sau khi ingest, dữ liệu được chuyển sang định dạng Parquet và lưu trữ trong Data Lake triển khai trên MinIO. Data Lake đóng vai trò lưu trữ dữ liệu thô, đảm bảo tính toàn vẹn, hỗ trợ kiểm soát phiên bản và cho phép tái xử lý khi cần thiết.

Từ Data Lake, dữ liệu được xử lý thông qua các bước ETL/ELT, bao gồm làm sạch, chuẩn hóa, tính toán các trường phát sinh và tổ chức dữ liệu theo mô hình phân tích. Dữ liệu sau xử lý được nạp vào kho dữ liệu ClickHouse – hệ quản trị cơ sở dữ liệu hướng cột có hiệu năng cao, phù hợp cho các truy vấn phân tích và tổng hợp.

Cuối cùng, Apache Superset kết nối trực tiếp với ClickHouse để trực quan hóa dữ liệu thông qua các dashboard và báo cáo tương tác, hỗ trợ theo dõi các chỉ số học tập, rèn luyện và tình trạng học vụ. Toàn bộ luồng dữ liệu được Apache Airflow quản lý bằng các DAG, đảm bảo thực thi đúng trình tự, chạy định kỳ và xử lý lỗi hiệu quả.

3.3. Công nghệ sử dụng

3.3.1. Apache Airflow

Apache Airflow là nền tảng mã nguồn mở được phát triển bởi Airbnb nhằm lập lịch, điều phối và giám sát các workflow xử lý dữ liệu phức tạp. Airflow cho phép định nghĩa các pipeline dữ liệu dưới dạng mã Python thông qua mô hình DAG (Directed Acyclic Graph), trong đó các tác vụ và mối quan hệ phụ thuộc được mô tả một cách rõ ràng và linh hoạt.

Kiến trúc của Airflow gồm các thành phần chính như Scheduler, Executor, Web Server và Metadata Database, phối hợp để lập lịch, thực thi, giám sát và lưu trữ trạng thái của các workflow. Nhờ khả năng định nghĩa pipeline bằng code, Airflow hỗ trợ xây dựng các pipeline động, dễ mở rộng và dễ bảo trì. Hệ thống cung cấp giao diện web trực quan để theo dõi trạng thái thực thi, xem log chi tiết và hỗ trợ cơ chế tự động chạy lại khi xảy ra lỗi.

Trong hệ thống pipeline hỗ trợ công tác cố vấn học tập, Apache Airflow được sử dụng làm công cụ điều phối trung tâm cho quy trình ETL. Các DAG được thiết kế để tự động hóa các bước trích xuất dữ liệu từ Data Lake, chuyển đổi và kiểm tra chất lượng dữ liệu, sau đó nạp dữ liệu vào Data Warehouse. Việc ứng dụng Airflow giúp

đảm bảo quy trình xử lý dữ liệu được thực hiện ổn định, có khả năng giám sát và giảm thiểu tối đa các thao tác thủ công.

3.3.2. MinIO

MinIO là hệ thống lưu trữ đối tượng (object storage) mã nguồn mở, được thiết kế để lưu trữ dữ liệu phi cấu trúc và tương thích hoàn toàn với Amazon S3 API. Nhờ khả năng triển khai linh hoạt trên hạ tầng on-premise hoặc đám mây, MinIO cho phép xây dựng các hệ thống lưu trữ dữ liệu quy mô lớn với chi phí thấp và dễ quản lý.

Kiến trúc của MinIO dựa trên nguyên tắc hệ thống phân tán, hỗ trợ mở rộng theo chiều ngang và cung cấp hiệu năng đọc/ghi cao. Hệ thống hỗ trợ các tính năng quan trọng như quản lý phiên bản dữ liệu, bảo mật truy cập và giao diện web trực quan để quản trị. Việc tương thích S3 giúp MinIO dễ dàng tích hợp với các công cụ xử lý dữ liệu và các ứng dụng hiện có mà không cần thay đổi nhiều về mã nguồn.

Trong phạm vi đề tài, MinIO được lựa chọn làm giải pháp Data Lake, sử dụng để lưu trữ dữ liệu thô dưới dạng các tệp Excel từ các đơn vị chức năng. Dữ liệu được tổ chức theo tầng Bronze, với quy ước đặt tên rõ ràng nhằm hỗ trợ truy vết, quản lý phiên bản và phục vụ hiệu quả cho các bước xử lý ETL tiếp theo trong pipeline hỗ trợ công tác cố vấn học tập.

3.3.3. Clickhouse

ClickHouse là hệ quản trị cơ sở dữ liệu mã nguồn mở dạng cột (columnar), được phát triển nhằm phục vụ các bài toán phân tích dữ liệu trực tuyến (OLAP) với hiệu năng cao. Nhờ kiến trúc lưu trữ theo cột và cơ chế nén dữ liệu hiệu quả, ClickHouse có khả năng xử lý khối lượng dữ liệu lớn và trả về kết quả truy vấn nhanh với thông lượng cao.

Hệ thống hỗ trợ nhiều engine lưu trữ khác nhau, trong đó MergeTree và các biến thể của nó được sử dụng phổ biến cho các kịch bản phân tích, cho phép phân vùng, sắp xếp và xử lý dữ liệu thay đổi theo thời gian. ClickHouse cung cấp cú pháp SQL mở rộng, hỗ trợ materialized view để tiền tính toán dữ liệu, giúp giảm đáng kể thời gian truy vấn trong các bài toán phân tích thường xuyên. Ngoài ra, hệ thống hỗ trợ triển khai phân tán, sao chép và giao tiếp thông qua các giao thức HTTP và TCP.

Trong đề tài, ClickHouse được lựa chọn làm Data Warehouse để lưu trữ dữ liệu đã được xử lý ở các tầng Silver và Gold. Lược đồ dữ liệu được thiết kế theo mô hình star schema với các bảng fact và dimension, kết hợp materialized views để tiền tính toán các chỉ số học vụ như GPA theo lớp, theo kỳ và số lượng sinh viên cảnh báo. Việc sử dụng ClickHouse giúp tối ưu hiệu năng truy vấn và đáp ứng hiệu quả các yêu cầu phân tích phục vụ công tác cố vấn học tập.

3.3.4. Apache Superset

Apache Superset là nền tảng mã nguồn mở hiện đại phục vụ khám phá và trực quan hóa dữ liệu, được thiết kế cho cả người dùng không chuyên kỹ thuật và các nhà phân tích dữ liệu. Superset cung cấp giao diện web trực quan để tạo biểu đồ, dashboard và thực hiện truy vấn dữ liệu một cách linh hoạt.

Kiến trúc của Superset gồm Web Server, Metadata Database, lớp cache và công cụ SQL Lab, cho phép kết nối với nhiều hệ quản trị cơ sở dữ liệu thông qua SQLAlchemy, bao gồm ClickHouse. Superset hỗ trợ đa dạng các loại biểu đồ, công cụ xây dựng truy vấn không cần lập trình, dashboard động với bộ lọc linh hoạt và khả năng truy vấn SQL trực tiếp. Bên cạnh đó, nền tảng này cung cấp cơ chế bảo mật chi tiết theo vai trò, hỗ trợ báo cáo định kỳ và cảnh báo tự động.

Trong hệ thống trình bày dữ liệu của đề tài, Apache Superset được sử dụng để xây dựng các dashboard phục vụ công tác cố vấn học tập. Các dashboard cung cấp cả góc nhìn tổng quan và chi tiết về tình trạng học tập, rèn luyện của sinh viên, cho phép theo dõi xu hướng, so sánh giữa các nhóm và nhận diện sớm sinh viên có nguy cơ học vụ. Việc kết nối trực tiếp Superset với Data Warehouse ClickHouse và khai thác các materialized view giúp đảm bảo hiệu năng truy vấn cao và khả năng tương tác thời gian thực cho người dùng.

3.4. Môi trường triển khai

Môi trường triển khai hệ thống pipeline xử lý dữ liệu phục vụ cố vấn học tập được thiết kế để đảm bảo tính ổn định, khả năng mở rộng và dễ dàng bảo trì. Việc lựa chọn công nghệ và cấu hình phần cứng phù hợp đóng vai trò quan trọng trong việc đảm bảo hiệu năng và độ tin cậy của hệ thống.

Thành phần	Cấu hình tối thiểu	Cấu hình khuyến nghị
CPU	12 vCPU	24 vCPU
RAM	8 GB	32 GB
Bộ nhớ	150 GB	Tùy theo lượng dữ liệu

Bảng 1. Cấu hình máy chủ thử nghiệm

Hệ điều hành: Ubuntu Server 22.04 LTS được chọn làm platform chính do tính ổn định, long-term support, và compatibility tốt với các công nghệ open-source. Containerization: Docker Engine và Docker Compose được sử dụng để đóng gói các dịch vụ, đảm bảo tính gọn nhẹ và dễ triển khai. Ngôn ngữ lập trình Python là ngôn ngữ chính cho ETL và xử lý dữ liệu.

3.5. Xây dựng các module hệ thống

3.6. Đánh giá hệ thống

Về tính tự động hóa: Hệ thống với Apache Airflow làm trung tâm orchestration đã đáp ứng tốt nhu cầu tự động hóa quy trình ETL. Cố vấn học tập không cần can thiệp thủ công vào việc thu thập, xử lý dữ liệu - dữ liệu điểm rèn luyện và học tập được cập nhật định kỳ và sẵn sàng trên dashboard.

Về hiệu năng truy vấn: ClickHouse là lựa chọn xuất sắc cho use case này. Với dữ liệu sinh viên (thường từ vài nghìn đến vài chục nghìn records mỗi học kỳ), ClickHouse có thể xử lý các truy vấn phân tích phức tạp trong vài milliseconds, đảm bảo dashboard load nhanh ngay cả khi nhiều cố vấn truy cập đồng thời.

Về khả năng mở rộng: Kiến trúc phân tầng rõ ràng giúp hệ thống dễ dàng scale khi số lượng sinh viên tăng hoặc cần thêm nguồn dữ liệu mới (ví dụ: dữ liệu điểm danh, kết quả thi). Minio làm data lake có thể lưu trữ lượng lớn dữ liệu lịch sử với chi phí thấp.

Về trực quan hóa: Superset cung cấp đầy đủ các loại biểu đồ cần thiết (line chart cho xu hướng điểm, bar chart so sánh, table chi tiết sinh viên) và có khả năng drill-down tốt, giúp cố vấn dễ dàng phân tích từ tổng quan đến chi tiết.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Thông qua đề tài này, chúng tôi đã thành công trong việc thiết kế và xây dựng một hệ thống Data Pipeline hoàn chỉnh, tích hợp các công nghệ hiện đại để tự động hóa quy trình xử lý dữ liệu sinh viên từ giai đoạn thu thập cho đến trực quan hóa. Hệ thống được xây dựng theo kiến trúc Medallion (Bronze-Silver-Gold), cho phép quản lý dữ liệu một cách hiệu quả qua các tầng xử lý khác nhau.

Về khía cạnh tính tự động hóa, Apache Airflow đã chứng tỏ là một lựa chọn xuất sắc làm trung tâm orchestration. Nhờ đó, cố vấn học tập không còn phải can thiệp thủ công vào việc thu thập và xử lý dữ liệu. Dữ liệu điểm rèn luyện và học tập được cập nhật định kỳ theo lịch biểu được lên lịch, và luôn sẵn sàng trên dashboard để sử dụng. Điều này đáng kể tiết kiệm thời gian, giảm khả năng sai sót do lỗi con người, và cho phép đội ngũ cố vấn tập trung vào việc phân tích và can thiệp hỗ trợ sinh viên.

Về khía cạnh hiệu năng truy vấn, ClickHouse tỏ ra là lựa chọn phù hợp cho use case này. Với khối lượng dữ liệu sinh viên thường từ vài nghìn đến vài chục nghìn records mỗi học kỳ, ClickHouse có khả năng xử lý các truy vấn phân tích phức tạp trong vài milliseconds. Điều này đảm bảo dashboard tải nhanh ngay cả khi nhiều cố vấn truy cập đồng thời, cung cấp trải nghiệm người dùng mượt mà và đáp ứng kịp thời.

Về khía cạnh khả năng mở rộng (scalability), kiến trúc phân tầng rõ ràng của hệ thống cho phép dễ dàng scale khi số lượng sinh viên tăng lên hoặc cần thêm các nguồn dữ liệu mới (chẳng hạn dữ liệu điểm danh, kết quả thi, hoặc dữ liệu từ các LMS khác). MinIO làm data lake có khả năng lưu trữ lượng lớn dữ liệu lịch sử với chi phí thấp, đáp ứng được nhu cầu tăng trưởng trong tương lai.

Về khía cạnh trực quan hóa dữ liệu, Apache Superset cung cấp một bộ công cụ đầy đủ với các loại biểu đồ đa dạng (line chart, bar chart, pie chart, table, heatmap, v.v.) để đáp ứng nhu cầu phân tích khác nhau. Khả năng drill-down và tương tác của Superset cho phép cố vấn dễ dàng chuyển đổi giữa các mức độ chi tiết, từ cái nhìn tổng quan đến các chi tiết cụ thể về từng sinh viên, hỗ trợ quá trình ra quyết định hiệu quả hơn.

Tuy nhiên, đề tài này vẫn còn một số hạn chế cần được cải thiện trong tương lai. Thứ nhất, hệ thống hiện tại được thử nghiệm với dữ liệu giả lập (dummy data) thay vì dữ liệu thực từ các hệ thống quản lý học tập (LMS) hiện hành. Trong bước tiếp theo, cần thiết kế và triển khai các connector để kết nối trực tiếp với các LMS phổ biến như Moodle, Canvas, hoặc các hệ thống nội bộ của trường, đảm bảo dữ liệu được cập nhật real-time từ các nguồn chính thức.

Thứ hai, hiện tại hệ thống cung cấp chủ yếu các chức năng báo cáo và trực quan hóa mô tả. Để tăng giá trị hơn nữa, có thể mở rộng hệ thống bằng cách tích hợp các mô hình Machine Learning để dự báo rủi ro học vụ, phân nhóm sinh viên có nhu cầu can thiệp, hoặc cá nhân hóa lộ trình đào tạo. Nền tảng dữ liệu đã được xây dựng là tiền đề tốt cho các ứng dụng nâng cao này.

Thứ ba, bảo mật dữ liệu và quản lý quyền hạn là những yếu tố cần được chú ý thêm. Mặc dù dữ liệu sinh viên là thông tin nhạy cảm, hệ thống hiện tại cần được bổ sung thêm các cơ chế xác thực nâng cao (authentication), phân quyền chi tiết (fine-grained authorization), mã hóa dữ liệu (encryption), và log audit để đảm bảo tuân thủ các quy định về bảo vệ dữ liệu cá nhân.

Cuối cùng, về mặt triển khai tại các trường đại học khác, cần phát triển một tài liệu hướng dẫn chi tiết, một template cấu hình có thể tái sử dụng, và các công cụ để dễ dàng triển khai hệ thống này tại các cơ sở giáo dục khác nhau. Đồng thời, cần đào tạo đội ngũ kỹ thuật tại các trường để họ có thể duy trì, mở rộng, và tối ưu hóa hệ thống theo nhu cầu riêng của từng cơ sở.

Tóm lại, đề tài này đã chứng minh được khả năng của một kiến trúc Data Pipeline hiện đại trong việc hỗ trợ công tác cố vấn học tập tại các trường đại học. Hệ thống không chỉ giải quyết được các bài toán quản lý dữ liệu hiện tại mà còn tạo nền tảng vững chắc cho các phát triển tiếp theo. Với việc tiếp tục cải thiện và mở rộng, chúng tôi tin rằng giải pháp này sẽ đóng góp tích cực vào quá trình chuyển đổi số trong giáo dục, nâng cao chất lượng quản lý và hỗ trợ sinh viên tại các trường đại học Việt Nam.