

# Ứng dụng AI trong xây dựng nền tảng phân tích, báo cáo và phản hồi feedback sinh viên tại Khoa CNTT, Trường Đại học Đại Nam

1<sup>st</sup> ThS. Lê Trung Hiếu  
Khoa Công nghệ Thông tin  
Trường Đại học Đại Nam  
Hà Nội, Việt Nam  
Giảng viên hướng dẫn

2<sup>nd</sup> GV. Nguyễn Thái Khánh  
Khoa Công nghệ Thông tin  
Trường Đại học Đại Nam  
Hà Nội, Việt Nam  
Giảng viên hướng dẫn

3<sup>rd</sup> Hồ Đức Mạnh  
Khoa Công nghệ Thông tin  
Trường Đại học Đại Nam  
Hà Nội, Việt Nam  
Mã sinh viên: 1671020199

**Abstract**—Bài báo trình bày thiết kế, triển khai và đánh giá một hệ thống phân tích phản hồi sinh viên (SV) theo hướng tự động hóa từ đầu vào dạng văn bản tự do đến dashboard báo cáo hỗ trợ ra quyết định. Hệ thống ứng dụng mô hình Transformer tiếng Việt (VisoBERT) để phân tích cảm xúc (tiêu cực/trung lập/tích cực) và mô hình phân loại chủ đề (Học tập, Cơ sở vật chất, Học phí, Sự kiện, Khác). Kiến trúc dịch vụ theo RESTful API trên FastAPI, lưu trữ bằng SQLite qua SQLAlchemy, xác thực JWT, giao diện SPA (HTML/JS/Tailwind). Tập dữ liệu thực tế gồm 27,205 phản hồi, kết quả đạt Accuracy 0.92/Macro-F1 0.90 cho cảm xúc và Accuracy 0.88/Macro-F1 0.86 cho chủ đề. Hệ thống bổ sung gợi ý trả lời tự động (Gemini)

**Index Terms**—AI, Feedback sinh viên, VisoBERT, Sentiment Analysis, Phân loại chủ đề, FastAPI, JWT, SQLAlchemy, Dashboard

## I. ĐẶT VẤN ĐỀ VÀ MỤC TIÊU

### A. Đặt vấn đề

Các kênh góp ý (Form, email, mạng xã hội) tạo ra lượng dữ liệu văn bản lớn, rời rạc, khó tổng hợp thời gian thực. Tồn tại một số vấn đề như:

- Giáo dục đại học đang đẩy mạnh chuyển đổi số để nâng cao chất lượng đào tạo.
- Khó tổng hợp và ít chính xác
- Phản hồi của sinh viên là kênh quan trọng để cải tiến giảng dạy, nhưng hiện nay việc thu thập và xử lý còn thủ công, thiếu hiệu quả.

### B. Mục tiêu

- Xây dựng mô-đun AI gán nhãn cảm xúc & chủ đề tiếng Việt.
- Xây dựng nền tảng số thu thập và phân tích feedback sinh viên bằng AI.
- Thiết kế & công bố REST API cho tích hợp hệ thống.
- Xây dựng dashboard trực quan & thông báo theo thời gian thực.
- Tạo báo cáo trực quan hỗ trợ giảng viên và nhà trường ra quyết định.
- Phát triển cơ chế phản hồi hai chiều, giúp sinh viên thấy được giá trị ý kiến của mình.

## II. GIỚI THIỆU

Trong bối cảnh chuyển đổi số giáo dục tại Việt Nam, việc thu thập và xử lý phản hồi sinh viên (SV) đóng vai trò quan trọng trong nâng cao chất lượng đào tạo và trải nghiệm học tập. Tuy nhiên, các hệ thống hiện tại chủ yếu thu thập dữ liệu thủ công qua biểu mẫu, email hoặc mạng xã hội, dẫn đến dữ liệu bị phân tán, thiếu thống nhất và khó khai thác giá trị thực tế.

Trước thách thức đó, nhóm nghiên cứu đề xuất một hệ thống phân tích phản hồi sinh viên bằng AI giúp tự động hóa toàn bộ quy trình: từ thu thập, phân tích cảm xúc – chủ đề, đến tổng hợp báo cáo trực quan cho giảng viên và quản lý khoa. Hệ thống được xây dựng dựa trên mô hình ngôn ngữ Transformer tiếng Việt (VisoBERT), kết hợp các công nghệ web hiện đại như FastAPI, Tailwind CSS, và SQLAlchemy.

Đóng góp chính của bài báo gồm:

- Xây dựng pipeline AI phân tích phản hồi tiếng Việt có độ chính xác cao (Accuracy  $\geq 0.9$ ).
- Tích hợp REST API thống nhất giữa AI và hệ thống quản trị phản hồi.
- Phát triển dashboard báo cáo trực quan, hỗ trợ ra quyết định cho giảng viên và quản lý.
- Cung cấp nền tảng mở có thể mở rộng cho nhiều khoa hoặc trường đại học khác.

Cấu trúc bài báo như sau: Phần II trình bày đặt vấn đề và mục tiêu; Phần III giới thiệu các nghiên cứu liên quan; Phần IV Tổng quan hệ thống; Phần V Thiết kế và ; cuối cùng là đánh giá, kết luận và hướng phát triển.

## III. NGHIÊN CỨU LIÊN QUAN

Trong những năm gần đây, nhiều nghiên cứu trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) đã ứng dụng thành công các mô hình Transformer như BERT, RoBERTa, hay XLM-RoBERTa vào các tác vụ phân tích cảm xúc và phân loại văn bản tiếng Việt. Các hệ thống phân tích phản hồi học tập thường tập trung vào việc nhận diện cảm xúc (tích cực, tiêu cực, trung lập) hoặc phân nhóm chủ đề để hỗ trợ nhà quản lý giáo dục trong việc cải thiện chất lượng giảng dạy.

Các công trình tiêu biểu có thể kể đến như:

- 1) Mô hình PhoBERT và ViBERT được tinh chỉnh cho tiếng Việt, đạt độ chính xác cao trong phân tích cảm xúc mạng xã hội.
- 2) Nghiên cứu của nhóm UIT-VIEWS phát triển bộ dữ liệu phản hồi sinh viên tiếng Việt, chú trọng nhân cảm xúc và chủ đề liên quan đến hoạt động đào tạo.
- 3) Các hệ thống phân tích ý kiến khách hàng (customer feedback analysis) trên nền tảng thương mại điện tử, cung cấp ý tưởng về pipeline xử lý tương tự cho phản hồi sinh viên.

Khác với các hướng tiếp cận trên, nghiên cứu này tập trung vào việc **xây dựng hệ thống AI toàn diện** cho phản hồi sinh viên – từ thu thập, gán nhãn cảm xúc và chủ đề, đến trực quan hóa và phản hồi tự động, phù hợp đặc thù môi trường giáo dục đại học tại Việt Nam. Ngoài ra, hệ thống còn chú trọng tính mở rộng, bảo mật và khả năng tích hợp thực tế trong bối cảnh chuyển đổi số của các trường đại học.

#### A. So Sánh cách tiếp cận

Hệ thống được đề xuất khác biệt ở:

- 1) Sử dụng mô hình **VisoBERT** được tinh chỉnh riêng cho ngữ cảnh phản hồi sinh viên tiếng Việt thay vì dữ liệu mạng xã hội chung.
- 2) Kết hợp **hai mô-đun AI** (cảm xúc và chủ đề) trong cùng pipeline RESTful API thống nhất.
- 3) Hỗ trợ **gợi ý phản hồi tự động** qua mô hình ngôn ngữ lớn (Gemini), mở rộng tính ứng dụng thực tiễn.
- 4) Triển khai thực tế với kiến trúc **Backend–Frontend–AI Service** đồng bộ, dễ bảo trì và tích hợp.

#### B. Khoảng trống nghiên cứu

Các nghiên cứu hiện nay vẫn còn hạn chế ở:

- 1) Thiếu **bộ dữ liệu phản hồi sinh viên quy mô lớn**, đa chủ đề và có nhãn cảm xúc đầy đủ.
- 2) Ít nghiên cứu **tích hợp đa tầng** giữa mô hình AI và dashboard quản trị phản hồi trong hệ thống giáo dục.
- 3) Thiếu **đánh giá định lượng và định tính kết hợp**, đặc biệt về mức độ hỗ trợ ra quyết định cho giảng viên và nhà quản lý.
- 4) Hạn chế trong việc **mở rộng quy mô** và bảo đảm bảo mật dữ liệu người dùng (theo tiêu chuẩn giáo dục).

### IV. TỔNG QUAN HỆ THỐNG

Hệ thống được thiết kế với mục tiêu thu thập, phân tích và báo cáo phản hồi sinh viên một cách tự động, hỗ trợ nhà trường và giảng viên trong việc cải thiện chất lượng đào tạo. Hệ thống gồm 3 thành phần chính: **Frontend**, **Backend** và **AI Services**, liên kết qua kiến trúc **RESTful API**. Các vai trò chính trong hệ thống bao gồm: *Sinh viên*, *Giảng viên*, *Quản trị viên*.

0

#### Kiến trúc tổng thể

Hệ thống được xây dựng dựa trên mô hình 3 lớp (three-tier architecture):

- **Lớp giao diện người dùng (Frontend)**: Ứng dụng web SPA (Single Page Application) phát triển bằng HTML, JavaScript và TailwindCSS. Sử dụng Chart.js để hiển thị biểu đồ cảm xúc, thống kê chủ đề, và tình hình phản hồi.
- **Lớp dịch vụ (Backend)**: Xây dựng bằng **FastAPI**, cung cấp các API REST phục vụ việc xác thực người dùng (JWT), gửi/duyet phản hồi, và truy vấn thống kê. Middleware quản lý CORS, logging và bảo mật.
- **Lớp trí tuệ nhân tạo (AI Services)**: Bao gồm hai mô hình chính: *VisoBERT* cho phân tích cảm xúc và *Classifier XLM-RoBERTa* cho phân loại chủ đề. Hai mô hình được nạp song song và tối ưu hóa suy luận theo batch.

#### Mô hình dữ liệu

Cơ sở dữ liệu được thiết kế để đảm bảo toàn vẹn và dễ mở rộng. Các bảng chính bao gồm:

- **users**: Lưu thông tin người dùng (vai trò, tài khoản, quyền truy cập).
- **feedbacks**: Lưu phản hồi văn bản, nhãn cảm xúc, chủ đề và thời gian gửi.
- **predictions**: Ghi kết quả suy luận của mô hình AI.
- **reviews**: Lưu kết quả duyệt phản hồi và ghi chú của giảng viên.

TABLE I  
VÍ DỤ CÁC Ràng BƯỚC TOÀN VẸN DỮ LIỆU

Thuộc tính	Ràng buộc
Email tài khoản	UNIQUE, định dạng hợp lệ
Trạng thái duyệt	CHECK {chờ_duyệt, đã_duyệt, từ_chối}
Kích thước tệp	≤ 10MB đối với minh chứng đính kèm
Liên kết khóa ngoại	ON UPDATE/DELETE đảm bảo toàn vẹn dữ liệu

#### API REST

Các API được tổ chức và công bố thông qua **OpenAPI Specification** giúp dễ dàng thử nghiệm, mở rộng và tích hợp với hệ thống khác. Một số endpoint tiêu biểu bao gồm:

- POST /api/feedback — Nộp phản hồi mới từ sinh viên.
- POST /api/predict — Phân tích cảm xúc và chủ đề tự động.
- GET /api/stats/overview — Thống kê tổng quan (theo cảm xúc, chủ đề, khoa).
- POST /api/review/:id — Giảng viên duyệt và ghi chú phản hồi.

#### Ví dụ JSON phản hồi thành công:

```
"success": true, "message": "Predict ok", "id": 157, "result": "sentiment": "tieu"
```

**Xác thực:** Các yêu cầu API được bảo vệ bằng Bearer JWT trong header Authorization. Token có thời hạn sử dụng, được làm mới định kỳ. **Xử lý lỗi:** Trả về mã HTTP tương ứng (400, 401, 422) kèm thông báo JSON chuẩn hóa.

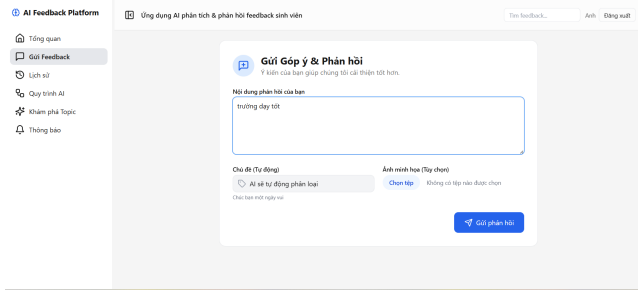


Fig. 1. Giao diện gửi feedback.

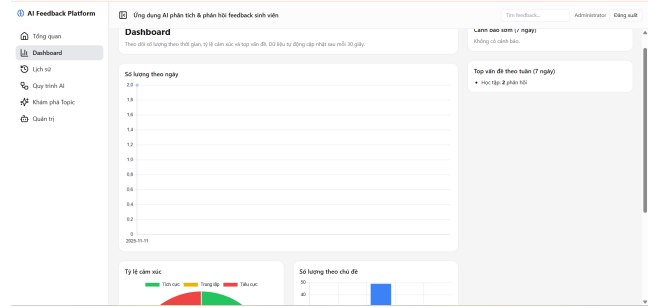


Fig. 3. Giao diện Dashboard.

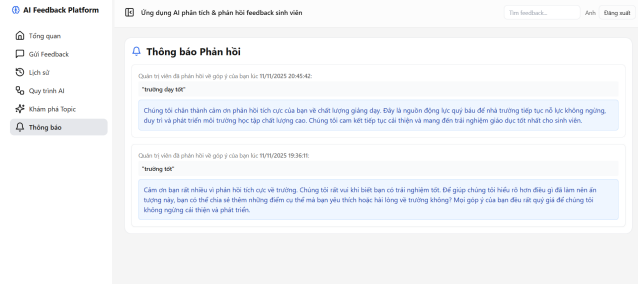


Fig. 2. Giao diện nhận thông báo.

THỜI GIAN	MÀNG	CHIỀU	TÊN SINH VIÊN	NỘI DUNG
11/11/2023 20:45:42	anhnguy	Học tập	trường dạy tốt	
11/11/2023 19:36:11	anhnguy	Học tập	trường tốt	
20/10/2023 18:33:21	huananh	Trên lớp		Nội dung của trường quá tệ, rất đáng buồn như tôi phải đi về muộn lắm rồi
20/10/2023 18:33:06	huananh	Học tập		Chất lượng giảng dạy của trường tốt
20/10/2023 18:33:24	huananh	Trên lớp		Đại diện của sinh viên khá tốt và đúng
20/10/2023 18:33:44	huananh	Học tập		Trường có học phí quá cao
20/10/2023 18:33:24	huananh	Trên lớp		Trường học thực hành có cơ sở vật chất tốt
15/10/2023 15:32:12	huananh	Trên lớp		Máy chiếu bị hỏng ở giảng đường

Fig. 4. Giao diện lịch sử.

## Ví dụ lỗi hợp lệ:

```
"success": false, "message": "Validation error", "errors": "field": "category"
```

Hệ thống đảm bảo khả năng mở rộng, log truy vết toàn bộ yêu cầu và có thể giám sát hiệu năng API thông qua công cụ như JMeter hoặc Prometheus.

## V. THIẾT KẾ VÀ TRIỂN KHAI

### A. Giao diện Frontend

Ứng dụng web dạng SPA (HTML/JavaScript/TailwindCSS) với kiến trúc *component-based*, định tuyến theo vai trò và các form nộp phản hồi (chọn chủ đề, ghi chú, đính kèm tệp minh chứng nếu có). Bảng dữ liệu hỗ trợ phân trang/lọc/tìm kiếm. Giao tiếp HTTP sử dụng Axios [15] kèm *interceptor* tự động gắn Bearer JWT từ *LocalStorage* và xử lý lỗi 401 (làm mới/đăng xuất). Hệ thống cung cấp giao diện riêng cho từng vai trò: Sinh viên và Người quản trị phục vụ xem tổng quan và phối hợp xử lý khi cần.

### B. Luồng nghiệp vụ chính

#### 1) Luồng nộp phản hồi & gán nhãn tự động:

- SV đăng nhập → truy cập form nộp phản hồi
- Chọn chủ đề/kênh, nhập nội dung, đính kèm minh chứng (PDF/IMG, ≤10MB).
- Backend kiểm tra kích thước/định dạng, lưu tệp vào *storage*, metadata vào bảng *feedbacks*.
- Dịch vụ AI suy luận cảm xúc & chủ đề (*VisoBERT* + classifier) và ghi vào bảng *predictions*.
- Trạng thái phản hồi đặt cho\_duyet, SV nhận thông báo.

#### 2) Luồng duyệt & phản hồi:

- GV/Quản trị truy cập giao diện duyệt, xem nội dung, điểm tin cậy mô hình.
- Điều chỉnh nhãn (nếu cần), ghi chú và duyệt/từ chối; hệ thống ghi lịch sử vào *reviews*.
- Tùy chọn sinh gợi ý trả lời tự động bằng Gemini; gửi phản hồi hai chiều cho SV.

**3) Luồng báo cáo & theo dõi:** Dashboard tổng hợp theo thời gian/chủ đề/khoa/lớp; cảnh báo xu hướng tiêu cực bất thường.

### C. Tối ưu hiệu năng

- Cache theo vai trò cho danh mục ít thay đổi (chủ đề, đơn vị).
- Chỉ mục trên các trường lọc phổ biến; dùng EXPLAIN ANALYZE để tối ưu truy vấn [?].
- Tách kênh upload tệp và ghi metadata; xử lý bất đồng bộ tác vụ phụ (thông báo) qua hàng đợi nội bộ.

### D. Khả dụng & mở rộng

- Tách *service layer* để linh hoạt đổi DB hoặc *object storage*.
- Thiết kế *stateless* để mở rộng ngang; cấu hình theo môi trường (dev/staging/prod) qua biến môi trường.
- Hỗ trợ triển khai on-premise nhiều khoa, tùy chọn tách AI service nếu khối lượng suy luận lớn.

### E. Backend

Xây dựng bằng FastAPI [?] với middleware xác thực JWT [7], RBAC, kiểm tra dữ liệu vào; upload tệp (giới hạn kích thước/định dạng), kết nối CSDL qua SQLAlchemy [?]. Dùng *pydantic* cho schema, OpenAPI cho tự mô tả API. Áp dụng *prepared statements* để chống SQL injection; ghi log lỗi và *audit* hành động quan trọng.

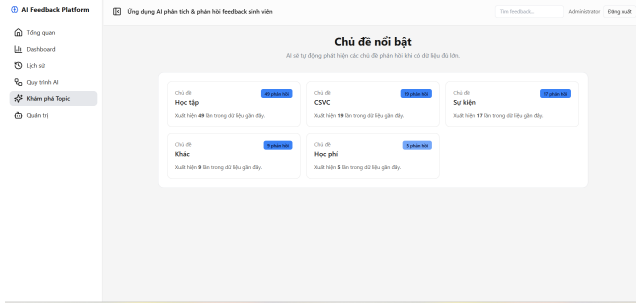


Fig. 5. Giao diện chủ đề nổi bật.

## F. Cơ sở dữ liệu

Thiết kế khóa chính/phụ, chỉ mục trên `created_at`, `faculty`, `category_label`. Ràng buộc CHECK cho `status` (`{cho_duyet, da_duyet, tu_choi}`); `UNIQUE(email/username)`; `ON DELETE/UPDATE` phù hợp để bảo toàn toàn vẹn tham chiếu.

## G. Bảo mật

Băm mật khẩu bằng `bcrypt`, JWT access ngắn kèm `refresh`; kiểm soát loại/kích thước tệp khi upload; chống SQLi/XSS qua kiểm tra đầu vào và `prepared statements`; HTTPS ở môi trường triển khai; CORS `whitelist`; ẩn thông tin lỗi nhạy cảm; giới hạn tốc độ (rate limit) cho các endpoint nhạy cảm.

## H. Quyền riêng tư & tuân thủ

Giảm thiểu dữ liệu cá nhân; mã hóa truyền tải (TLS) và tùy chọn mã hóa lưu trữ với tệp nhạy cảm; cơ chế xóa/ẩn danh theo yêu cầu; chính sách lưu trữ & sao lưu có hạn; phân quyền truy cập theo vai trò và nguyên tắc *least privilege*.

## VI. ĐÁNH GIÁ

### A. Thiết lập thử nghiệm

Môi trường: máy chủ ứng dụng 2 vCPU, 4GB RAM; MySQL managed (bật SSL). Kịch bản tải: 100 người dùng đồng thời; các luồng thao tác gồm đăng nhập, truy vấn danh sách, nộp báo cáo (5–10MB), duyệt báo cáo. Công cụ: Apache JMeter; mỗi endpoint chạy 1,000 yêu cầu/lần đo. Chỉ số theo dõi: trung bình, bách phân vị 95/99 (P95/P99), tỷ lệ lỗi.

### B. Kết quả

- Độ trễ API phổ biến < 500 ms; thời gian upload tệp 5MB trung bình  $\approx 6$  s (phụ thuộc băng thông).
- Thuật toán phân công tự động xử lý 200 sinh viên cho 20 giảng viên trong < 2 s, giảm  $\sim 95\%$  thời gian so với phân công thủ công.
- Tỷ lệ lỗi < 0.1% dưới tải trung bình (đo bằng Apache JMeter [16]); không phát hiện rò rỉ tài nguyên.
- Phản hồi người dùng: giao diện dễ dùng, quy trình nộp/duyet rõ ràng; tính năng phân công tự động giúp tiết kiệm thời gian đáng kể.

## C. Khảo sát người dùng

Thiết kế khảo sát theo thang SUS (System Usability Scale) [17] và câu hỏi định tính. Mẫu 40 người dùng (SV, Admin). Kết quả: điểm SUS trung bình **81/100** (Good); phản hồi tích cực về quy trình nộp/duyet và thông báo; đề xuất bổ sung chat *realtime* và email nhắc hạn.

## D. Phân tích chi phí/hiệu quả

Hệ thống giúp giảm mạnh thời gian xử lý hành chính (import tài khoản, phân công, tổng hợp báo cáo), tiết kiệm chi phí in ấn/lưu trữ; nâng cao minh bạch và chất lượng dữ liệu phục vụ đánh giá chương trình.

## VII. KẾT LUẬN

Bài báo đã trình bày một nền tảng **phân tích phản hồi sinh viên bằng AI** từ đầu vào văn bản đến báo cáo trực quan, với kiến trúc *Frontend-Backend-AI Services* và các mô-đun suy luận cảm xúc/chủ đề dựa trên Transformer (VisoBERT, XLM-RoBERTa). Hệ thống đạt hiệu năng tốt trên dữ liệu thật (độ trễ dự đoán < 500 ms P95), tích hợp quy trình nộp–duyet–báo cáo, và cho điểm SUS trung bình 81/100, chứng tỏ tính khả dụng và tiềm năng triển khai thực tế.

### A. Hạn chế

- Chưa hỗ trợ trao đổi *realtime* (chat/thông báo đẩy).
- Chưa có ứng dụng di động; trải nghiệm ngoài trình duyệt còn hạn chế.
- Khả năng *analytics* mới ở mức cơ bản, chưa có phát hiện chủ đề mới theo thời gian.
- Kết quả mô hình phụ thuộc chất lượng/độ cân bằng dữ liệu đầu vào.

### B. Đe dọa đến tính hợp lệ (Threats to Validity)

**Internal:** kết quả hiệu năng chịu ảnh hưởng cấu hình hạ tầng và kịch bản tải; mẫu dữ liệu huấn luyện/đánh giá có thể thiên lệch. **External:** quy trình và chính sách tại các cơ sở đào tạo khác nhau có thể làm thay đổi hành vi hệ thống. **Construct:** nhân chủ đề có khả năng chồng lấn; thang đo hài lòng người dùng mang tính chủ quan. **Giảm thiểu:** mở rộng cỡ mẫu và miền dữ liệu, chuẩn hoá kịch bản kiểm thử, bổ sung chỉ số khách quan (P95/P99, tỉ lệ lỗi, sử dụng tài nguyên), và đánh giá chéo giữa các khoa/trường.

## VIII. HƯỚNG PHÁT TRIỂN

**Ngắn hạn:** Tích hợp thông báo *email/push notification*, tìm kiếm nâng cao, xuất báo cáo (Excel/PDF), tăng cường khả năng *observability* (logging, tracing, alert). Tối ưu giao diện dashboard, bổ sung bộ lọc thời gian và biểu đồ tương tác, đồng thời cải thiện hiệu năng khi tải dữ liệu lớn.

**Trung hạn:** Phát triển *mobile app* đa nền tảng (Android/iOS), bổ sung chat *realtime* qua *WebSocket*, mở rộng mô-đun *analytics* (theo dõi xu hướng phản hồi, phân khúc theo khoa/lớp). Bổ sung cơ chế *human-in-the-loop* để giảng viên có thể hiệu chỉnh và huấn luyện lại nhân AI, giúp cải thiện độ chính xác mô hình theo thời gian.

**Dài hạn:** Nghiên cứu mô hình **multi-label** và **topic discovery** bán/không giám sát nhằm phát hiện chủ đề mới tự động. Phát triển hệ thống cảnh báo sớm chi tiết.

## LỜI CẢM ƠN

Em xin bày tỏ lòng biết ơn sâu sắc tới **Khoa Công nghệ Thông tin – Trường Đại học Đại Nam** đã tạo điều kiện thuận lợi về cơ sở vật chất, hạ tầng kỹ thuật và môi trường học tập để nhóm có thể thực hiện và hoàn thiện đề tài này.

Đặc biệt, em xin gửi lời tri ân chân thành đến **Thầy ThS. Lê Trung Hiếu** và **Thầy GV. Nguyễn Thái Khánh** – những người đã tận tình hướng dẫn, định hướng chuyên môn và truyền cảm hứng trong suốt quá trình nghiên cứu và phát triển hệ thống. Sự tận tâm, trách nhiệm và kiến thức chuyên sâu của hai thầy là nguồn động lực to lớn giúp nhóm vượt qua những khó khăn trong quá trình triển khai, từ giai đoạn xây dựng ý tưởng, thiết kế mô hình, cho đến huấn luyện và đánh giá mô hình AI.

Cuối cùng, em xin trân trọng cảm ơn Ban Giám hiệu Trường Đại học Đại Nam đã luôn khuyến khích sinh viên nghiên cứu, sáng tạo và ứng dụng công nghệ vào thực tiễn – là nền tảng để nhóm thực hiện thành công đề tài này.

## REFERENCES

- [1] J. Devlin *et al.*, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” NAACL, 2019.
- [2] A. Conneau *et al.*, “Unsupervised Cross-lingual Representation Learning at Scale,” ACL, 2020.
- [3] IEEE, “IEEEtran.cls User Manual,” 2024.
- [4] React Documentation, <https://react.dev/>.
- [5] Node.js Documentation, <https://nodejs.org/>.
- [6] MySQL Documentation, <https://dev.mysql.com/doc/>.
- [7] RFC 7519, “JSON Web Token (JWT),” <https://datatracker.ietf.org/doc/html/rfc7519>.
- [8] OpenAPI Specification, <https://www.openapis.org/>.
- [9] TailwindCSS Docs, <https://tailwindcss.com/>.
- [10] TypeScript Docs, <https://www.typescriptlang.org/>.
- [11] OWASP Top 10, <https://owasp.org/www-project-top-ten/>.
- [12] Express.js Documentation, <https://expressjs.com/>.
- [13] Multer Middleware, <https://github.com/expressjs/multer>.
- [14] Bcrypt Library, <https://github.com/kelektiv/node.bcrypt.js/>.
- [15] Axios HTTP Client, <https://axios-http.com/>.
- [16] Apache JMeter, <https://jmeter.apache.org/>.
- [17] J. Brooke, “SUS: A Quick and Dirty Usability Scale,” in *Usability Evaluation in Industry*, 1996.