

TRƯỜNG ĐẠI HỌC TRÀ VINH
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ



ISO 9001:2015

DƯƠNG TRUNG HIẾU

XÂY DỰNG HỆ THỐNG TỰ ĐỘNG CHẤM ĐIỂM
VÀ HỖ TRỢ SỬA LỖI TIẾNG ANH
KỸ NĂNG WRITING

KHÓA LUẬN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

VĨNH LONG, NĂM 2025

TRƯỜNG ĐẠI HỌC TRÀ VINH
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ

XÂY DỰNG HỆ THỐNG TỰ ĐỘNG CHẤM ĐIỂM
VÀ HỖ TRỢ SỬA LỖI TIẾNG ANH
KỸ NĂNG WRITING

KHÓA LUẬN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên thực hiện: **DƯƠNG TRUNG HIẾU**

Mã số sinh viên: **110121139**

Lớp: **DA21TTA**

GVHD: **ThS. PHAN THỊ PHƯƠNG NAM**

VĨNH LONG, NĂM 2025

LỜI MỞ ĐẦU

Trong thời đại toàn cầu hóa và chuyển đổi số, tiếng Anh đã trở thành ngôn ngữ giao tiếp phổ biến trên toàn thế giới, đóng vai trò quan trọng trong học tập, công việc và nghiên cứu. Trong bốn kỹ năng ngôn ngữ (Nghe – Nói – Đọc – Viết), kỹ năng Viết (Writing) được xem là một trong những kỹ năng khó đối với người học, đặc biệt là người học tiếng Anh như ngôn ngữ thứ hai (EFL – English as a Foreign Language). Việc viết đúng ngữ pháp, sử dụng từ vựng phù hợp và trình bày mạch lạc là một thách thức không nhỏ.

Hiện nay, việc chấm điểm và sửa lỗi bài viết tiếng Anh được thực hiện trực tiếp bởi giáo viên hoặc chuyên gia ngôn ngữ, điều này không chỉ tiêu tốn nhiều thời gian mà còn khó mở rộng cho số lượng lớn người học. Trong bối cảnh đó, sự phát triển vượt bậc của các mô hình ngôn ngữ lớn (Large Language Models – LLMs), tiêu biểu là ChatGPT do OpenAI phát triển, đã mở ra cơ hội mới cho việc ứng dụng trí tuệ nhân tạo vào giáo dục, đặc biệt là trong việc tự động hóa quá trình đánh giá và sửa lỗi bài viết tiếng Anh. Từ thực tiễn đó, đề án “Xây dựng hệ thống tự động chấm điểm và hỗ trợ sửa lỗi tiếng Anh kỹ năng Writing” được thực hiện với mục tiêu ứng dụng mô hình ChatGPT được tinh chỉnh lại (retrained hoặc fine-tuned) để xây dựng một hệ thống có khả năng phân tích, phát hiện lỗi ngôn ngữ, gợi ý sửa lỗi và đánh giá chất lượng bài viết tiếng Anh một cách tự động, khách quan và hiệu quả. Hệ thống được kỳ vọng sẽ trở thành công cụ hỗ trợ đắc lực cho người học trong quá trình luyện viết, giúp cải thiện khả năng sử dụng tiếng Anh một cách chính xác và tự nhiên.

Đề án không chỉ có ý nghĩa học thuật mà còn hướng đến khả năng ứng dụng thực tiễn trong các nền tảng học tiếng Anh trực tuyến hoặc các hệ thống học tập thông minh (Intelligent Tutoring Systems), góp phần thúc đẩy việc áp dụng công nghệ AI vào giáo dục hiện đại tại Việt Nam.

LỜI CẢM ƠN

Trước tiên, em xin bày tỏ lòng biết ơn chân thành và sâu sắc đến quý thầy cô Khoa Công nghệ Thông Tin – Trường Kỹ Thuật và Công nghệ, Trường Đại học Trà Vinh, những người đã tận tình giảng dạy, truyền đạt kiến thức và tạo điều kiện thuận lợi cho em trong suốt quá trình học tập và thực hiện đồ án tốt nghiệp. Đặc biệt, em xin gửi lời cảm ơn chân thành đến Cô Phan Thị Phương Nam, người đã trực tiếp hướng dẫn em thực hiện đề tài “Xây dựng hệ thống tự động chấm điểm và hỗ trợ sửa lỗi tiếng Anh kỹ năng Writing”. Cô đã dành nhiều thời gian, công sức để định hướng, góp ý chuyên môn và động viên em trong suốt quá trình thực hiện đồ án. Những nhận xét quý báu của Cô không chỉ giúp em hoàn thiện đồ án mà còn là hành trang quý giá cho quá trình học tập và làm việc sau này. Em cũng xin gửi lời cảm ơn đến bạn bè, người thân đã luôn đồng hành, chia sẻ và hỗ trợ em cả về tinh thần lẫn học thuật trong suốt thời gian thực hiện đồ án.

Mặc dù đã cố gắng hết sức, nhưng do kiến thức và kinh nghiệm còn hạn chế, đồ án chắc chắn không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý, nhận xét của quý thầy cô để em có thể hoàn thiện hơn trong những nghiên cứu sau này.

Vĩnh Long, ngày tháng năm 2025

Sinh viên thực hiện

DƯƠNG TRUNG HIẾU

NHẬN XÉT

(Của giảng viên hướng dẫn trong đề án, khoá luận của sinh viên)

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Giảng viên hướng dẫn

(ký và ghi rõ họ tên)

UBND TỈNH VĨNH LONG
TRƯỜNG ĐẠI HỌC TRÀ VINH

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP

(Của giảng viên hướng dẫn)

Họ và tên sinh viên: Dương Trung Hiếu

MSSV: 110121139

Ngành: Công nghệ Thông tin

Khóa: 2021

Tên đề tài: Xây dựng hệ thống tự động chấm điểm và hỗ trợ sửa lỗi tiếng Anh kỹ năng Writing

Họ và tên Giảng viên hướng dẫn: Phan Thị Phương Nam

Chức danh: Giảng viên chính

Học vị: Thạc sĩ

NHẬN XÉT

1. Nội dung đề tài:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Ưu điểm:

.....

.....

.....

.....

3. Nhược điểm:

.....

.....

.....

.....

.....

4. Điểm mới đề tài:

.....

.....

.....

.....

5. Giá trị thực trên đề tài:

.....

.....

.....

.....

7. Đề nghị sửa chữa bổ sung:

.....

.....

.....

.....

.....

8. Đánh giá:

.....

.....

.....

.....

....., ngày tháng năm 20...

Giảng viên hướng dẫn

(Ký & ghi rõ họ tên)

NHẬN XÉT

(Của giảng viên chấm trong đề án, khoá luận của sinh viên)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Giảng viên chấm
(ký và ghi rõ họ tên)

BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP

(Của cán bộ chấm đồ án, khóa luận)

Họ và tên người nhận xét:

Chức danh: Học vị:

Chuyên ngành:

Cơ quan công tác:

Tên sinh viên:

Tên đề tài đồ án, khóa luận tốt nghiệp:

.....
.....

I. Ý KIẾN NHẬN XÉT

1. Nội dung:

.....
.....
.....
.....
.....
.....
.....
.....
.....

2. Điểm mới các kết quả của đồ án, khóa luận:

.....
.....
.....

3. Ứng dụng thực tế:

.....
.....
.....
.....
.....

.....

.....

II. CÁC VẤN ĐỀ CẦN LÀM RÕ

(Các câu hỏi của giáo viên phản biện)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

III. KẾT LUẬN

(Ghi rõ đồng ý hay không đồng ý cho bảo vệ đề án khóa luận tốt nghiệp)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

....., ngày tháng năm 20...

Người nhận xét

(Ký & ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	1
1.1. Đặt vấn đề.....	1
1.2. Mục đích.....	1
1.3. Nội dung nghiên cứu	2
1.4. Phương pháp nghiên cứu.....	4
1.5. Phạm vi nghiên cứu.....	5
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	6
2.1. Tổng quan về công nghệ phát triển frontend được sử dụng.....	6
2.1.1. HTML	6
2.1.2. CSS	7
2.1.3. Javascript.....	8
2.1.4. TypeScript.....	9
2.1.5. ReactJS.....	10
2.2. Tổng quan về công nghệ phát triển backend được sử dụng.....	15
2.2.1. Python	15
2.2.2. Flask.....	16
2.3. Tổng quan về Cơ sở dữ liệu được sử dụng	18
2.3.1. MySQL	18
2.4. OpenAI SDK	22
2.4.1. Khái niệm.....	22
2.4.2. Cách thức hoạt động của mô hình GPT trong OpenAI SDK.....	22
2.5. Tổng quan về mô hình sử dụng – mô hình MVC	24
2.5.1. Tổng quan	24
2.5.2. Các thành phần trong mô hình MVC.....	24
2.5.3. Chức năng	25
2.5.4. Cách hoạt động của MVC.....	26
2.5.5. Ưu và nhược điểm của MVC.....	27
2.6. Tổng quan về Writing và các tiêu chuẩn đánh giá	27
2.6.1. Tổng quan	27
2.6.2. Các loại Writing.....	28
2.6.3. Các tiêu chuẩn đánh giá Writing.....	29

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU	31
3.1. Mô tả bài toán.....	31
3.2. Yêu cầu chức năng của hệ thống.....	31
3.2.1. Đối với người quản trị.....	31
3.2.2. Đối với người dùng	32
3.3. Yêu cầu phi chức năng của hệ thống	33
3.3.1. Hiệu năng	33
3.3.2. Bảo mật	34
3.3.3. Khả năng mở rộng.....	34
3.3.4. Dễ bảo trì.....	34
3.3.5. Tính thân thiện với người dùng	34
3.3.6. Khả năng tích hợp	35
3.4. Phân tích thiết kế hệ thống	35
3.4.1. Đặc tả hệ thống	35
3.4.2. API chính giữa frontend và backend	35
3.4.3. Mô hình thực thể quan hệ (ERD).....	36
3.4.4. Sơ đồ use-case.....	42
3.5. Cài đặt môi trường và công cụ sử dụng	45
3.5.1. Cài đặt công cụ lập trình chính - Visual Studio Code	45
3.5.2. Cài đặt môi trường chạy frontend	45
3.5.3. Cài đặt môi trường chạy backend	46
3.5.4. Xampp và MySQL.....	46
3.6. Thiết kế layout cho giao diện trang web	47
3.6.1. Layout giao diện người dùng	47
3.6.2. Layout giao diện quản trị	53
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU	58
4.1. Giao diện người dùng.....	58
4.1.1. Giao diện trang chủ.....	58
4.1.2. Giao diện trang đăng ký.....	59
4.1.3. Giao diện trang đăng nhập	59
4.1.4. Giao diện trang lựa chọn dạng bài luyện viết	60
4.1.5. Giao diện trang làm bài luyện viết.....	62

4.1.6. Giao diện trang chọn nhiệm vụ (Task) và nhập bài viết.....	63
4.1.7. Trang lịch sử điểm số.....	64
4.1.8. Giao diện trang điểm tổng hợp (Combined Score).....	65
4.1.9. Giao diện trang chọn gói nâng cấp	66
4.1.10. Trang hồ sơ cá nhân	67
4.2. Giao diện trang quản trị.....	68
4.2.1. Dashboard chính	68
4.2.2. Trang quản lý người dùng.....	69
4.2.3. Trang quản lý đơn hàng	70
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	71
5.1. Kết luận	71
5.1.1. Kết quả và đóng góp của đề án.....	71
5.1.2. Hạn chế	72
5.2. Hướng phát triển	72

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa (Tiếng Anh)	Ý nghĩa (Tiếng Việt)
ACID	Atomicity, Consistency, Isolation, Durability	Tính nguyên tử, nhất quán, cô lập, bền vững (đảm bảo toàn vẹn dữ liệu trong CSDL)
AI	Artificial Intelligence	Trí tuệ nhân tạo
API	Application Programming Interface	Giao diện lập trình ứng dụng
CSS	Cascading Style Sheets	Tập tin định kiểu tầng
DOM	Document Object Model	Mô hình đối tượng tài liệu
GPT	Generative Pre-trained Transformer	Mô hình ngôn ngữ được huấn luyện trước
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
IoT	Internet of Things	Internet vạn vật
JSX	JavaScript Syntax Extension	Phần mở rộng cú pháp của JavaScript
MVC	Model - View - Controller	Mô hình phần mềm gồm: Dữ liệu - Giao diện - Bộ điều khiển
MySQL	My Structured Query Language	Ngôn ngữ truy vấn có cấu trúc của tôi (Hệ quản trị cơ sở dữ liệu quan hệ)
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
RDBMS	Relational Database Management System	Hệ quản trị cơ sở dữ liệu quan hệ
SDK	Software Development Kit	Bộ công cụ phát triển phần mềm
SPA	Single Page Application	Ứng dụng đơn trang
SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
UI	User Interface	Giao diện người dùng

DANH MỤC BẢNG BIỂU

Bảng 1: So sánh ReactJS và các thư viện Javascript khác	13
Bảng 2: Liệt kê và mô tả các API.....	35
Bảng 3: Mô tả các thực thể và mối quan h	36
Bảng 4: Mô tả cấu trúc bảng User	38
Bảng 5: Mô tả cấu trúc bảng Essay	38
Bảng 6: Mô tả cấu trúc bảng EssaySuggestion	39
Bảng 7: Mô tả cấu trúc bảng AIChat.....	39
Bảng 8: Mô tả cấu trúc bảng Export.....	40
Bảng 9: Mô tả cấu trúc bảng Payment.....	40
Bảng 10: Mô tả cấu trúc bảngSubscription	41
Bảng 11: Mô tả cấu trúc bảng UserCredit.....	41
Bảng 12: Mô tả cấu trúc bảng WritingScore	42

DANH MỤC HÌNH ẢNH

Hình 2.1: Cơ chế cập nhật Virtual DOM và Real DOM [14]	11
Hình 2.2: Cơ chế mã hóa dữ liệu trong MySQL [15]	19
Hình 2.3. Cấu trúc của MySQL [15]	21
Hình 2.4: Mô hình hoạt động của MySQL [15]	22
Hình 2.5: Mô hình MVC - Cách dữ liệu và giao diện tương tác với nhau.....	25
Hình 2.6: Luồng hoạt động của mô hình MVC trong ứng dụng web	26
Hình 2.7: Tiêu chí chấm bài Writing (IELTS) [23]	30
Hình 3.1: Mô hình ERD	36
Hình 3.2: Sơ đồ use case - actor: Admin.....	43
Hình 3.3: Sơ đồ use case - actor: Người dùng.....	45
Hình 3.4: Giao diện khởi động Xampp	47
Hình 3.5: Giao diện cài đặt thành công xampp	47
Hình 3.6: Layout giao diện trang chủ.....	48
Hình 3.7: Layout trang đăng ký.....	49
Hình 3.8: Layout trang đăng nhập.....	49
Hình 3.9: Layout trang nhập bài viết và gửi chấm điểm	50
Hình 3.10: Layout trang nâng cấp gói	51
Hình 3.11: Layout trang luyện thi VSTEP hoặc IELTS.....	51
Hình 3.12: Layout trang bắt đầu luyện thi.....	52
Hình 3.13: Layout trang xem lịch sử các bài đã nộp.....	53
Hình 3.14: Layout trang Dashboard	54
Hình 3.15: Layout trang quản lý tài khoản người dùng	55
Hình 3.16: Layout trang quản lý các gói nâng cấp của người dùng.....	56
Hình 3.17: Layout trang thống kê số lượng đăng ký tài khoản người dùng	57
Hình 4.1: Giao diện trang chủ BandBoost	58
Hình 4.2: Giao diện trang đăng ký	59
Hình 4.3: Giao diện trang đăng nhập.....	60
Hình 4.4: Giao diện trang luyện viết VSTEP và IELTS	61
Hình 4.5: Giao diện trang làm bài luyện thi	63
Hình 4.6 Giao diện trang viết và nộp bài.....	64
Hình 4.7: Giao diện trang lịch sử nộp bài	65

Hình 4.8: Giao diện trang sửa lỗi hiển thị chi tiết lịch sử các bài chấm	65
Hình 4.9: Giao diện trang cộng điểm Writing	66
Hình 4.10: Giao diện trang nâng cấp gói.....	67
Hình 4.11: Giao diện trang thông tin cá nhân	68
Hình 4.12: Giao diện trang dashboard.....	69
Hình 4.13: Giao diện trang quản lý người dùng.....	69
Hình 4.14: Giao diện trang quản lý các gói người dùng đăng ký	70

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1. Đặt vấn đề

Trong bối cảnh xã hội hiện đại không ngừng phát triển và hội nhập, sự bùng nổ của công nghệ thông tin đã tạo ra những chuyển biến sâu sắc trong lĩnh vực giáo dục, đặc biệt là trong việc học và kiểm tra ngoại ngữ. Việc đánh giá kỹ năng Writing tiếng Anh đang trở thành một thách thức lớn do nhu cầu về sự chính xác, khách quan và nhanh chóng trong chấm điểm bài viết ngày càng cao. Các phương pháp chấm điểm truyền thống vừa tốn thời gian, vừa phụ thuộc nhiều vào năng lực và sự chủ quan của giám khảo, gây ảnh hưởng đến tính nhất quán và hiệu quả của quá trình đánh giá.

Do đó, xây dựng một hệ thống tự động chấm điểm và hỗ trợ sửa lỗi tiếng Anh kỹ năng Writing không chỉ đáp ứng yêu cầu hiện đại về tự động hóa, mà còn nâng cao chất lượng giảng dạy và học tập, đồng thời tối ưu hóa trải nghiệm người học. Hệ thống này cần cung cấp khả năng phân tích ngữ pháp, cấu trúc câu, từ vựng, ý tưởng và sự mạch lạc trong bài viết một cách thông minh, tự động gợi ý sửa lỗi và cải thiện, giúp người học nhanh chóng nhận diện điểm yếu và nâng cao trình độ.

Về mặt kỹ thuật, hệ thống sẽ được xây dựng dựa trên kiến trúc linh hoạt, dễ mở rộng và tích hợp các mô hình trí tuệ nhân tạo (AI) tiên tiến như xử lý ngôn ngữ tự nhiên (NLP). Backend sử dụng FlaskAPI kết hợp MySQL quản lý dữ liệu, đồng thời tận dụng. Phía frontend, ReactJS được lựa chọn để xây dựng giao diện người dùng trực quan, thân thiện, hỗ trợ tương tác hiệu quả cho người dùng.

Trên cơ sở đó, đề tài "Xây dựng hệ thống tự động chấm điểm và hỗ trợ sửa lỗi tiếng Anh kỹ năng Writing" được đề xuất thực hiện với mục tiêu nghiên cứu và phát triển một nền tảng đánh giá Writing tự động chính xác, hiệu quả, hỗ trợ người học góp phần nâng cao chất lượng ngoại ngữ trong kỷ nguyên số.

1.2. Mục đích

Mục đích của đề tài là nghiên cứu và triển khai một hệ thống tự động chấm điểm và hỗ trợ sửa lỗi tiếng Anh kỹ năng Writing, tích hợp các công nghệ AI nhằm nâng cao trải nghiệm người học và tối ưu hóa hiệu quả đánh giá. Hệ thống sẽ đáp ứng đầy đủ các chức năng cơ bản của một nền tảng đánh giá ngoại ngữ hiện đại, đồng thời ứng dụng AI để mang lại những cải tiến vượt trội trong các khía cạnh sau:

- Đối với người học: Với vai trò là đối tượng chính sử dụng hệ thống, người học cần được cung cấp một môi trường luyện tập và đánh giá Writing tiện lợi, thông minh và cá nhân hóa. Cụ thể, hệ thống sẽ đảm bảo các chức năng sau:

+ Trải nghiệm đánh giá chính xác và nhanh chóng: Người học có thể dễ dàng nộp bài viết để nhận điểm số tự động dựa trên các tiêu chí chuẩn mực như ngữ pháp, từ vựng, cấu trúc câu và sự mạch lạc trong bài. Hệ thống cũng cung cấp phản hồi chi tiết và gợi ý sửa lỗi giúp người học hiểu rõ điểm mạnh, điểm yếu và hướng cải thiện.

+ Tương tác và phản hồi liên tục: Người học được khuyến khích đóng góp phản hồi về chất lượng chấm điểm và đề xuất các tính năng mới. Hệ thống còn cho phép lưu lại lịch sử bài viết và tiến trình học tập, giúp theo dõi sự tiến bộ qua thời gian.

+ Quản lý luyện tập linh hoạt: Hệ thống hỗ trợ người học bằng các đề thi mẫu, luyện tập theo từng dạng kỹ năng viết. Ngoài ra, hệ thống tích hợp các bài tập sửa lỗi, phân tích ngữ pháp và từ vựng theo mức độ cá nhân hóa.

- Đối với quản trị hệ thống: Hệ thống cung cấp bộ công cụ quản lý linh hoạt nhằm kiểm soát dữ liệu, giám sát chất lượng chấm điểm và vận hành tổng thể. Cụ thể bao gồm: Quản lý người dùng và quyền truy cập Hệ thống hỗ trợ quản lý tài khoản học viên, bao gồm phân quyền, chỉnh sửa thông tin.

Mục đích của đề tài không chỉ dừng lại ở việc xây dựng một hệ thống đánh giá Writing tự động với các chức năng cơ bản, mà còn hướng tới việc phát triển một nền tảng thông minh, bảo mật và có khả năng mở rộng, tận dụng tối đa sức mạnh của AI nhằm nâng cao trải nghiệm học tập cá nhân hóa, tối ưu hóa quy trình đánh giá và thúc đẩy hiệu quả đào tạo kỹ năng viết tiếng Anh trong kỷ nguyên số.

1.3. Nội dung nghiên cứu

Đề tài “Xây dựng hệ thống tự động chấm điểm và hỗ trợ sửa lỗi tiếng Anh kỹ năng Writing” được triển khai dựa trên việc kết hợp các công nghệ AI tiên tiến và kiến trúc phần mềm hiện đại nhằm đáp ứng nhu cầu đánh giá kỹ năng viết tiếng Anh một cách chính xác, nhanh chóng và khách quan. Nội dung nghiên cứu tập trung vào các nội dung sau:

- Khảo sát và phân tích yêu cầu hệ thống:

+ Thu thập yêu cầu từ giáo viên và tham khảo các hệ thống luyện kỹ năng Writing hiện có thông qua việc nghiên cứu tài liệu, tìm hiểu thông tin từ website.

- + Xác định các tiêu chí chấm điểm Writing như: ngữ pháp, từ vựng, tính mạch lạc và bố cục bài viết dựa trên chuẩn đánh giá quốc tế.

- + Xác định các tính năng cốt lõi của hệ thống như: chấm điểm tự động, gợi ý sửa lỗi, quản lý người dùng, theo dõi tiến độ học tập.

- Nghiên cứu và lựa chọn công nghệ chấm điểm tự động:

- + Tìm hiểu các mô hình xử lý ngôn ngữ tự nhiên (NLP) dựa trên kiến trúc Transformer, đặc biệt là GPT-3.5 Turbo.

- + Phân tích ưu, nhược điểm của các phương pháp chấm điểm truyền thống so với phương pháp AI.

- + Xác định cách tích hợp GPT-3.5 Turbo để phân tích ngữ pháp, từ vựng, cấu trúc câu và tính mạch lạc.

- Thiết kế kiến trúc hệ thống và cơ sở dữ liệu:

- + Áp dụng phương pháp UML để mô hình hóa yêu cầu và chức năng hệ thống.

- + Thiết kế cơ sở dữ liệu quản lý thông tin người dùng, bài viết, lịch sử chấm điểm, phản hồi và gợi ý cải thiện.

- Phát triển mô-đun chấm điểm và gợi ý sửa lỗi:

- + Xây dựng API backend sử dụng Flask để giao tiếp với mô hình GPT-3.5 Turbo.

- + Tính toán điểm số dựa trên các tiêu chí chuẩn, đồng thời đưa ra gợi ý cải thiện cụ thể.

- Xây dựng giao diện người dùng (UI/UX):

- + Thiết kế frontend bằng ReactJS đảm bảo trực quan, thân thiện, hỗ trợ các thao tác như nộp bài, nhận điểm, xem lịch sử và phản hồi.

- + Tích hợp các chức năng luyện thi, quản lý bài viết và theo dõi tiến độ học tập.

- Kiểm thử

- + Kiểm thử chức năng, hiệu năng và độ chính xác chấm điểm của hệ thống.

- + Thu thập phản hồi từ người dùng thực tế để cải tiến hệ thống.

Việc triển khai các nội dung nghiên cứu trên nhằm đảm bảo hệ thống không chỉ thực hiện chức năng chấm điểm và gợi ý sửa lỗi chính xác, mà còn mang lại trải nghiệm học tập cá nhân hóa, góp phần nâng cao hiệu quả học tập trong lĩnh vực tiếng Anh Writing.

1.4. Phương pháp nghiên cứu

Để triển khai đề tài “Xây dựng hệ thống tự động chấm điểm và hỗ trợ sửa lỗi tiếng Anh kỹ năng Writing”, các phương pháp nghiên cứu bao gồm:

- Nghiên cứu lý thuyết:

+ Tìm hiểu các tài liệu, bài báo khoa học và tiêu chuẩn đánh giá kỹ năng Writing của các kỳ thi quốc tế (IELTS, VSTEP) để xây dựng tiêu chí chấm điểm.

+ Nghiên cứu các mô hình xử lý ngôn ngữ tự nhiên (NLP) hiện đại, đặc biệt là kiến trúc Transformer và mô hình GPT-3.5 Turbo, để ứng dụng trong việc phân tích và đánh giá bài viết.

- Khảo sát yêu cầu và phân tích hệ thống:

+ Thu thập yêu cầu từ giảng viên thông qua trao đổi trực tiếp và tìm hiểu các nền tảng luyện Writing hiện có (tham khảo website, tài liệu công khai).

+ Phân tích yêu cầu chức năng, xác định các tính năng chính: chấm điểm tự động, gợi ý sửa lỗi, quản lý người dùng, theo dõi tiến trình học tập.

- Phân tích và thiết kế hệ thống:

+ Sử dụng UML (Unified Modeling Language) để mô hình hóa hệ thống: Áp dụng biểu đồ Use Case để mô tả các tác nhân và chức năng của hệ thống, đồng thời sử dụng biểu đồ quan hệ thực thể (ERD) để thiết kế cấu trúc cơ sở dữ liệu.

+ Thiết kế kiến trúc theo mô hình MVC (Model – View – Controller) kết hợp tích hợp mô-đun AI.

- Xây dựng và triển khai hệ thống:

+ Phát triển backend bằng Flask API, tích hợp GPT-3.5 Turbo để xử lý bài viết, chấm điểm và gợi ý sửa lỗi, tối ưu GPT-3.5 để giải quyết các yêu cầu từ hệ thống, qua đó tích hợp GPT-3.5 để phát triển thêm ChatAI.

+ Thiết kế và phát triển frontend bằng ReactJS để xây dựng giao diện trực quan, thân thiện với người dùng.

+ Xây dựng cơ sở dữ liệu để quản lý người dùng, bài viết và kết quả chấm điểm.

- Kiểm thử và đánh giá hệ thống:

+ Tiến hành kiểm thử chức năng và kiểm thử tính chính xác của việc chấm điểm, gợi ý sửa lỗi.

1.5. Phạm vi nghiên cứu

❖ Về chức năng hệ thống:

- Tập trung vào kỹ năng Writing của tiếng Anh, không bao gồm các kỹ năng khác như Listening, Reading hoặc Speaking.
- Hệ thống hỗ trợ hai bài thi phổ biến là VSTEP và IELTS với các dạng bài chính:
 - + VSTEP: Task 1 (Email Writing), Task 2 (Essay Writing)
 - + IELTS: Task 1, Task 2 (Essay Writing).
- Các chức năng chính gồm:
 - + Luyện kỹ năng Writing của các bài thi VSTEP hoặc IELTS
 - + Nộp bài viết và nhận điểm số tự động dựa trên tiêu chí: ngữ pháp, từ vựng, mạch lạc và bố cục.
 - + Nhận gợi ý sửa lỗi chi tiết từ AI.
 - + Quản lý thông tin người dùng, lịch sử bài viết và tiến độ luyện tập.

❖ Về công nghệ sử dụng:

- Sử dụng GPT-3.5 Turbo (mô hình thuộc kiến trúc Transformer) để phân tích bài viết và sinh phản hồi.
- Backend được phát triển bằng Flask API, kết nối với cơ sở dữ liệu.
- Frontend sử dụng ReactJS để xây dựng giao diện tương tác.

❖ Về đối tượng nghiên cứu:

- Người dùng: Người có nhu cầu luyện kỹ năng Writing, sử dụng hệ thống để nộp bài, nhận chấm điểm và gợi ý sửa lỗi.
- Quản trị viên: Người quản lý hệ thống, chịu trách nhiệm theo dõi dữ liệu, quản lý người dùng và đảm bảo vận hành ổn định.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về công nghệ phát triển frontend được sử dụng

2.1.1. HTML

2.1.1.1. Khái niệm

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản, được sử dụng để xây dựng cấu trúc cơ bản của các trang web. HTML mô tả cách hiển thị nội dung trên trình duyệt thông qua các thẻ (tags). Các thẻ này xác định các thành phần như tiêu đề, đoạn văn, hình ảnh, liên kết, biểu mẫu, ... HTML không phải là ngôn ngữ lập trình, mà là ngôn ngữ đánh dấu dùng để sắp xếp và tổ chức nội dung hiển thị trên website [5].

2.1.1.2. Ưu và nhược điểm

❖ Ưu điểm

- Cú pháp dễ hiểu, thân thiện với người mới bắt đầu.
- Là chuẩn mở, không cần cài đặt phần mềm riêng.
- Hỗ trợ trên hầu hết các trình duyệt hiện nay.
- Dễ dàng tích hợp với CSS, JavaScript, các framework khác (React, Angular, ...).
- Hỗ trợ các thẻ ngữ nghĩa giúp tối ưu công cụ tìm kiếm.

❖ Nhược điểm

- Không thể xử lý logic như điều kiện, vòng lặp.
- Giao diện thô sơ, khó tạo giao diện đẹp nếu không kết hợp CSS.
- Mã HTML hiển thị công khai, không nên chứa thông tin nhạy cảm.
- Phụ thuộc vào các công nghệ khác, muốn có chức năng động cần kết hợp thêm JavaScript hoặc backend.

2.1.1.3. Ứng dụng

- Là nền tảng cơ bản để tạo ra mọi trang web.
- Xây dựng bố cục, định dạng nội dung trang.
- Tạo email HTML để gửi đến người dùng với định dạng phong phú.
- Là phần hiển thị của nhiều ứng dụng chạy trên trình duyệt

- Hỗ trợ những đa phương tiện như video, audio, và các yếu tố tương tác như form giúp web sinh động hơn

2.1.2. CSS

2.1.2.1. Khái niệm

CSS (Cascading Style Sheets – Tập tin định kiểu tầng) là ngôn ngữ dùng để mô tả cách hiển thị của các thành phần HTML trên trình duyệt như màu sắc, bố cục, kích thước, font chữ, hiệu ứng, ... Giúp tạo giao diện đẹp mắt và nhất quán cho website [4].

2.1.2.2. Ưu và nhược điểm

❖ Ưu điểm:

- Tách biệt nội dung và hình thức, giúp mã HTML gọn gàng, dễ bảo trì.
- Tái sử dụng dễ dàng vì một file CSS có thể áp dụng cho nhiều trang web.
- Giảm dung lượng HTML, trình duyệt tải nhanh hơn.
- Có thể thiết kế giao diện hiện đại, responsive (thích ứng nhiều thiết bị).
- Dễ dàng thay đổi giao diện toàn bộ trang web chỉ bằng cách sửa một file CSS.

❖ Nhược điểm:

- Một số thuộc tính CSS có thể không hiển thị giống nhau trên các trình duyệt khác nhau.
- CSS chỉ định kiểu hiển thị, không xử lý tương tác hay dữ liệu.
- Khó kiểm soát trong dự án lớn nếu không tổ chức tốt, dễ bị rối nếu nhiều lớp (class) chồng chéo nhau.

2.1.2.3. Ứng dụng

- Giúp website hiển thị tốt trên nhiều loại thiết bị (máy tính, điện thoại, máy tính bảng).
- Tạo nhiều giao diện cho cùng một nội dung.
- Dùng hoạt ảnh, hiệu ứng để tạo trải nghiệm người dùng sinh động.
- Là nền tảng của các thư viện như Tailwind CSS, Bootstrap,...

2.1.3. Javascript

2.1.3.1. Khái niệm

JavaScript là một ngôn ngữ lập trình phía client (chạy trên trình duyệt), được dùng để tạo ra các chức năng tương tác, xử lý logic và nâng cao trải nghiệm người dùng trên website [6].

2.1.3.2. Ưu và nhược điểm

❖ **Ưu điểm:**

- Giúp website phản hồi với hành động của người dùng (như click, nhập dữ liệu,...).
- Không cần gửi yêu cầu đến server nhanh hơn, giảm tải cho server.
- Dễ học, cú pháp gần gũi, phù hợp với người mới bắt đầu học lập trình web.
- Khả năng mở rộng cao, kết hợp với các thư viện và framework mạnh mẽ như React, Angular, Vue,...
- Có thể dùng trên web, mobile app (React Native), server (Node.js), IoT,...

❖ **Nhược điểm**

- Dễ bị khai thác bảo mật, mã JavaScript chạy trên trình duyệt có thể bị người dùng sửa đổi.
- Một số tính năng cũ có thể hoạt động không nhất quán trên các trình duyệt khác nhau.
- Hiệu năng thấp hơn so với các ngôn ngữ server-side khi xử lý lượng dữ liệu lớn.

2.1.3.3. Ứng dụng

- Thực hiện các hiệu ứng như dropdown, slideshow, form validation,...
- Xử lý logic trên trình duyệt như kiểm tra dữ liệu nhập, phản hồi sự kiện,...
- Là ngôn ngữ chính trong các SPA (Single Page Application) như Gmail, Facebook,...
- Phát triển ứng dụng mobile thông qua React Native, Ionic,...
- Lập trình backend với Node.js, có thể xây dựng API, máy chủ,...
- Tích hợp AI, vẽ đồ thị, game nhỏ,... dùng thư viện như TensorFlow.js, Chart.js, Three.js,...

2.1.4. TypeScript

2.1.4.1. Khái niệm

TypeScript là một ngôn ngữ lập trình được phát triển bởi Microsoft, là phiên bản mở rộng của JavaScript, bổ sung thêm tính năng kiểu tĩnh (static type) và các tính năng hướng đối tượng như interface, enum, generic,... TypeScript được biên dịch (transpile) sang JavaScript trước khi chạy trên trình duyệt [7]. Khi dùng với React, TypeScript giúp kiểm soát kiểu dữ liệu tốt hơn trong props, state, hooks, v.v..., từ đó giúp phát triển ứng dụng an toàn, rõ ràng và dễ bảo trì hơn.

2.1.4.2. Ưu và nhược điểm

❖ Ưu điểm:

- Giúp phát hiện lỗi sớm trước khi chạy.
- Dễ dàng nâng cấp hoặc sửa code mà không lo lỗi kiểu dữ liệu.
- Trình soạn thảo như VSCode hỗ trợ auto-complete, gợi ý kiểu.
- Rõ ràng về cấu trúc và hợp đồng (interface, types). Đặc biệt hữu ích khi làm việc nhóm.
- Hỗ trợ tốt cho dự án lớn với tính năng chia module, namespace, generic...

❖ Nhược điểm:

- Tốn thời gian cài đặt ban đầu. Cần cấu hình và làm quen với công cụ biên dịch (tsconfig).
- Cú pháp phức tạp hơn JavaScript. Do đó, người mới bắt đầu có thể thấy khó tiếp cận.
- Cần transpile sang JavaScript trước khi chạy.
- Khó khăn khi dùng thư viện không hỗ trợ TypeScript tốt.

2.1.4.3. Ứng dụng của TypeScript trong React

- Xây dựng ứng dụng web hiện đại, tăng tính ổn định, giúp kiểm soát rõ props, state, context,...
- Tạo component tái sử dụng an toàn. Gán kiểu dữ liệu rõ ràng cho props và state.
- Làm việc nhóm chuyên nghiệp do dễ đọc hiểu, dễ debug, dễ maintain.
- Tích hợp với backend do dễ map kiểu dữ liệu từ API vào frontend.

2.1.5. ReactJS

2.1.5.1. Khái niệm

ReactJS là một thư viện JavaScript mã nguồn mở được dùng để xây dựng giao diện người dùng (UI) cho website. ReactJS là một thư viện dựa trên components, khai báo cho phép lập trình viên xây dựng UI component có thể tái sử dụng và tuân theo phương pháp Virtual DOM. Điều này giúp tối ưu hóa hiệu suất render bằng cách giảm thiểu các bản cập nhật DOM [12].

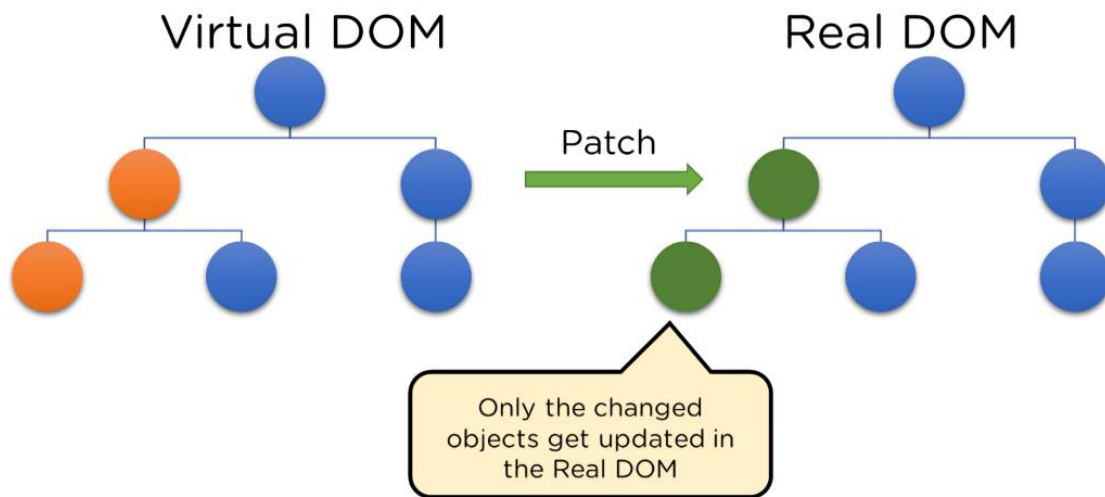
ReactJS được phát triển và vận hành bởi Facebook, bằng cách triển khai DOM ảo, ReactJS đã giải quyết cải thiện vấn đề DOM chậm và trở nên phổ biến hơn. Bên cạnh đó, các phiên bản của ReactJS luôn được phát triển và cập nhật các tính năng mới phù hợp với xu hướng hiện nay. Các phiên bản mới của ReactJS không chỉ cập nhật tính năng như Hooks, Concurrent Mode, Suspense mà còn cải thiện hiệu năng, giúp lập trình viên dễ dàng tạo ra giao diện tương tác mượt mà và thân thiện với người dùng.

2.1.5.2. Một số tính năng nổi bật của React

- Kiến trúc Component-Based: ReactJS cung cấp tính năng chia nhỏ UI thành các component nhỏ hơn và có tính độc lập. Mỗi components có trạng thái và thuộc tính (props) riêng biệt.

- JSX (JavaScript Syntax Extension): JSX là phần mở rộng cú pháp cho JavaScript cho phép lập trình viên viết mã giống HTML trong các tệp JavaScript của nó. Đồng thời, JSX làm cho components ReactJS dễ đọc và thu hút hơn.

- Virtual DOM: ReactJS duy trì một biểu diễn nhẹ của DOM thực tế trong bộ nhớ. Khi có thay đổi, ReactJS chỉ cập nhật hiệu quả của các phần cần thiết trong DOM.



Hình 2.1: Cơ chế cập nhật Virtual DOM và Real DOM [14]

- Liên kết dữ liệu 1 chiều (One-way Data Binding): Dữ liệu trong ReactJS chỉ chạy theo một hướng, tức là dữ liệu được truyền từ trên xuống dưới, từ thành phần lớn đến thành phần nhỏ. Các thuộc tính (props) trong components con không thể trả về dữ liệu cho components lớn hơn nó, nhưng nó có thể giao tiếp với components cha để sửa đổi các trạng thái theo đầu vào được cung cấp.

- Performance: ReactJS sử dụng DOM ảo và chỉ cập nhật các phần đã sửa đổi. Do đó, điều này làm cho DOM chạy nhanh hơn. DOM thực thi trong bộ nhớ để bạn có thể tạo components giúp DOM chạy nhanh hơn.

- Components: ReactJS chia trang web thành nhiều component khác nhau, bởi vì chúng hoạt động dựa trên các components. Mỗi component là một phần của UI có logic và thiết kế riêng. Do đó, logic component được viết bằng JavaScript sẽ giúp khởi chạy dễ dàng, nhanh chóng và có thể tái sử dụng.

- Dữ liệu một trang (Single-Page Applications – SPA): ReactJS được khuyến khích sử dụng trong việc tạo SPA, cho phép cập nhật nội dung mượt mà mà không cần tải lại trang. Tập trung vào các thành phần có thể tái sử dụng khiến nó trở nên thích hợp cho các ứng dụng thời gian thực.

- Props: Props là từ viết tắt của Properties trong ReactJS Props, cho phép người dùng truyền tham số hoặc dữ liệu cho component. Props giúp component trở nên năng động hơn và một thành phần chỉ đọc, không thể thay đổi.

2.1.5.3. Ưu và nhược điểm của React

❖ Ưu điểm:

- Xây dựng DOM ảo tùy chỉnh, bởi vì DOM ảo JavaScript hoạt động nhanh hơn do với DOM thông thường, điều này sẽ giúp nâng cao hiệu suất của ứng dụng.
- ReactJS tạo ra giao diện người dùng UI cho website hấp dẫn và ấn tượng.
- Thân thiện với công cụ tìm kiếm.
- Module và dữ liệu giúp quản lý ứng dụng lớn dễ dàng hơn bằng cách cải thiện khả năng đọc.
- Tích hợp nhiều kiến trúc khác nhau.
- Tối ưu và đơn giản hóa quy trình môi trường tập lệnh.
- Bảo trì dễ dàng và tăng cường đầu ra.
- Đảm bảo render nhanh hơn.
- Có thể sử dụng nhiều hệ thống khác nhau, trên cả máy khách (client-side) và máy chủ (server-side).

❖ **Nhược điểm:**

- Chỉ giải quyết góc và khoảng cách của ứng dụng, do đó phải chọn các kỹ thuật bổ sung nếu muốn có bộ công cụ phát triển đầy đủ.
- Sử dụng tập lệnh nội tuyến và JSX có thể khiến một số lập trình viên cảm thấy không thoải mái hoặc phù hợp với nhu cầu.

2.1.5.4. Cách hoạt động của React

ReactJS có thể cho phép truyền mã HTML với JavaScript với những lợi ích như:

- DOM là một cấu trúc cây biểu diễn tài liệu HTML, và JavaScript có thể tương tác với DOM để thay đổi nội dung và cấu trúc của trang web.
- ReactJS sử dụng một thuật toán diff hiệu quả để tìm ra những phần tử cần thay đổi và chỉ cập nhật những phần đó trên DOM thực tế, tránh việc cập nhật lại toàn bộ DOM.
- Khi có nhu cầu đọc và ghi vào DOM, JSX sẽ sử dụng DOM ảo của nó. Sau đó DOM ảo sẽ cố gắng tìm cách hiệu quả để cập nhật DOM của trình duyệt.

Trong ReactJS, khi tạo ra các phần tử React bằng cách sử dụng các hàm JSX như `<div>`, `<button>`,... Các phần tử này không phải là DOM thực, mà là các đối tượng đơn giản được tạo ra dễ dàng.

ReactJS sử dụng một DOM ảo (Virtual DOM) để tối ưu quá trình cập nhật DOM thực. Khi cập nhật một phần tử React, ReactJS sẽ so sánh phần tử mới với phần tử cũ

trong DOM ảo, sau đó chỉ cập nhật những phần cần thiết trong DOM thực, nhờ tốc độ xử lý nhanh của JavaScript.

Bảng 1: So sánh ReactJS và các thư viện Javascript khác

STT	Tiêu chí	Angular	ReactJS	Vue.js
1	Phát triển bởi	Google	Facebook	Evan You, a former Google engineer
2	Thành lập	2010	2013	2014
3	Được viết bởi	TypeScript	JavaScript (có thể dùng JSX)	JavaScript
4	Category	Framework	Thư viện (có thể dùng như framework)	Framework
5	Cú pháp	HTML	JSX	HTML (mặc định) và JSX
6	Kiến trúc	Model-View-Control (MVC), Component-based (dựa trên thành phần)	Component-based (dựa trên thành phần)	Component-based (dựa trên thành phần)
7	Data Binding	Liên kết dữ liệu hai chiều	Liên kết dữ liệu một chiều	Liên kết giao diện người dùng của ứng dụng với tệp dữ liệu thích hợp
8	Performance	Có khả năng nhanh hơn đối với các SPA phức tạp	Có khả năng nhanh hơn đối với các ứng dụng đơn giản vì sử dụng DOM ảo	Hiệu suất vượt trội trong cả các dự án nhỏ hơn hoặc ít phức tạp hơn.

STT	Tiêu chí	Angular	ReactJS	Vue.js
9	Tính năng chính	TypeScript, Dependency Injection, Module, Routing, Services, Directives, Forms	Virtual DOM, Components, JSX	Virtual DOM, Documentation, Conditional Rendering, Interpolation, Iteration
10	Mục đích sử dụng	Tạo ứng dụng web di động, xây dựng các ứng dụng web lớn, phức tạp, yêu cầu cấu trúc chặt chẽ.	Tạo các thành phần UI tương tác	Phát triển giao diện người dùng của website và các ứng dụng một trang (SPA)
11	Cấu trúc	MVC phức tạp	Chế độ xem dựa trên thành phần linh hoạt	Chế độ xem dựa trên thành phần linh hoạt
12	Hiệu suất	Cao	Rất Cao (nhờ DOM ảo)	Cao
13	Ngôn ngữ được hỗ trợ	TypeScript (chính), JavaScript	JavaScript (có thể dùng JSX, TypeScript)	NativeScript, JavaScript
14	Ứng dụng nổi bật	Google, Gmail, YouTube, Microsoft Office, Xbox, Mixer, Weather.com, Forbes	Atlassian, Instagram, Airbnb, Pinterest, Netflix, Dropbox, Uber, Reddit	Behance, Google, Facebook, Wizz Air, Nintendo, Upwork, Alibaba, Vice, Trustpilot, Netflix, Euronews

2.1.5.5. Ứng dụng

Các ứng dụng của ReactJS bao gồm:

- Xây dựng UI cho ứng dụng web.
- Phát triển ứng dụng đơn trang (SPA), nơi mà toàn bộ ứng dụng được tải một lần và các thành phần giao diện được cập nhật động khi người dùng tương tác, mang lại trải nghiệm người dùng mượt mà.
- Xây dựng các ứng dụng di động (React Native), một framework khác của Facebook, sử dụng ReactJS để phát triển các ứng dụng di động cho cả iOS và Android, tận dụng khả năng tái sử dụng code và kiến trúc của ReactJS.
- Tạo các thành phần giao diện người dùng có thể tái sử dụng.
- Cải thiện hiệu suất do ReactJS sử dụng Virtual DOM để tối ưu hóa quá trình render, chỉ cập nhật những phần thay đổi trên giao diện người dùng, giúp cải thiện hiệu suất của ứng dụng.

2.2. Tổng quan về công nghệ phát triển backend được sử dụng

2.2.1. Python

2.2.1.1. Khái niệm

Python là một ngôn ngữ lập trình thông dịch, bậc cao, hướng đối tượng, được phát triển bởi Guido van Rossum và ra mắt lần đầu vào năm 1991. Python nổi bật với cú pháp đơn giản, dễ đọc, dễ học [11][20], giúp lập trình viên tập trung vào giải quyết vấn đề hơn là cú pháp ngôn ngữ. Python hiện là một trong những ngôn ngữ phổ biến nhất trên thế giới, được sử dụng rộng rãi trong nhiều lĩnh vực như khoa học dữ liệu, trí tuệ nhân tạo, lập trình web, tự động hóa, phân tích dữ liệu, ...

2.2.1.2. Ưu và nhược điểm

❖ Ưu điểm:

- Cú pháp đơn giản, dễ đọc vì Python gần giống ngôn ngữ tự nhiên, thuận tiện cho người mới bắt đầu học lập trình.
- Có sẵn hàng nghìn thư viện và framework hỗ trợ các lĩnh vực khác nhau như NumPy, Pandas, TensorFlow, Flask, Django, ...
- Chạy được trên nhiều hệ điều hành như Windows, macOS, Linux.

- Là ngôn ngữ phổ biến nhất trong lĩnh vực trí tuệ nhân tạo và học máy.
- Hỗ trợ tốt từ cộng đồng giúp giải quyết nhanh các vấn đề khi lập trình.

❖ **Nhược điểm:**

- So với các ngôn ngữ như C/C++ hoặc Java, Python chậm hơn do là ngôn ngữ thông dịch.
- Không thích hợp để phát triển các ứng dụng cần xử lý thời gian thực, như trò chơi đồ họa cao cấp hoặc phần mềm hệ thống.
- Python sử dụng cơ chế quản lý bộ nhớ tự động (garbage collection) nên đôi khi thiếu kiểm soát chi tiết trong các ứng dụng lớn.

2.2.1.3. Ứng dụng

Python được sử dụng trong nhiều lĩnh vực khác nhau, bao gồm:

- Trí tuệ nhân tạo (AI) và học máy (Machine Learning): Dùng để xây dựng và huấn luyện các mô hình học máy, xử lý ngôn ngữ tự nhiên (NLP), thị giác máy tính, v.v.
- Phân tích và trực quan hóa dữ liệu như sử dụng để thu thập, xử lý, phân tích dữ liệu lớn.
- Lập trình web: Framework như Flask và Django giúp xây dựng các ứng dụng web nhanh chóng.
- Phát triển trò chơi và ứng dụng desktop.
- Viết các đoạn script để tự động hóa công việc như xử lý file, gửi email, kiểm tra hệ thống.

2.2.2. Flask

2.2.2.1. Khái niệm

Flask là một micro web framework được viết bằng ngôn ngữ Python. Flask được phát triển bởi Armin Ronacher và lần đầu ra mắt vào năm 2010. Dù là một “micro framework”, Flask vẫn cung cấp đầy đủ các tính năng cần thiết để xây dựng ứng dụng web, chẳng hạn như routing (định tuyến), xử lý request/response, template engine (Jinja2), và hỗ trợ mở rộng qua các tiện ích (extension). Với thiết kế đơn giản, dễ học, dễ mở rộng, Flask là lựa chọn phổ biến cho các dự án nhỏ, vừa hoặc các hệ thống cần tính linh hoạt cao.

2.2.2.2. Ưu và nhược điểm

❖ Ưu điểm:

- Cấu trúc đơn giản, không bắt buộc quy ước nghiêm ngặt như một số framework khác.
- Lập trình viên có toàn quyền kiểm soát cấu trúc và các thành phần ứng dụng.
- Hỗ trợ tốt cho RESTful API: Phù hợp để xây dựng backend phục vụ frontend hiện đại như React, Vue,...
- Dễ dàng kết hợp với các thư viện như TensorFlow, PyTorch [22], Scikit-learn để triển khai mô hình học máy dưới dạng web API [13].
- Thư viện mở rộng phong phú. Có thể tích hợp với SQLAlchemy, Flask-Login, Flask-WTF, v.v.

❖ Nhược điểm:

- Không tích hợp sẵn nhiều chức năng. Phải tự thêm nhiều tính năng thủ công (xác thực, ORM, quản lý người dùng...) so với framework lớn như Django.
- Không thích hợp cho dự án lớn phức tạp nếu không có tổ chức chặt chẽ.
- Bảo mật cần tự xử lý kỹ lưỡng hơn.

2.2.2.3. Ứng dụng

Flask được sử dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là:

- Xây dựng RESTful API vì Flask rất thích hợp để phát triển các dịch vụ web backend cung cấp dữ liệu cho frontend (React, Angular, mobile apps...).
- Triển khai mô hình học máy (Machine Learning Deployment) nhờ khả năng tích hợp dễ dàng với các thư viện như TensorFlow, PyTorch, Scikit-learn để tạo API phục vụ dự đoán và phân tích dữ liệu.
- Xây dựng website nhỏ và vừa, blog cá nhân, hoặc ứng dụng web đơn giản vì Flask có cấu trúc gọn nhẹ và dễ mở rộng.
- Phát triển nguyên mẫu (prototype) nhanh cho các dự án nghiên cứu hoặc khởi nghiệp do tính linh hoạt và tốc độ triển khai cao.
- Xây dựng hệ thống quản lý (CMS, quản lý người dùng) bằng cách tích hợp cơ sở dữ liệu (MySQL, PostgreSQL, MongoDB) và mở rộng với các tiện ích Flask-Login, Flask-Admin.

- Tạo cổng kết nối với AI hoặc chatbot vì Flask dễ dàng triển khai API cho mô hình AI, chatbot, hoặc xử lý ngôn ngữ tự nhiên (NLP).
- Ứng dụng IoT và xử lý dữ liệu thời gian thực bằng cách kết hợp Flask với WebSocket hoặc MQTT để quản lý dữ liệu từ thiết bị thông minh.
- Tích hợp với dịch vụ đám mây (Cloud) như AWS, Heroku, Google Cloud để triển khai ứng dụng quy mô nhỏ đến trung bình một cách nhanh chóng.

2.3. Tổng quan về Cơ sở dữ liệu được sử dụng

2.3.1. MySQL

2.3.1.1. Khái niệm

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mã nguồn mở, được phát triển bởi công ty MySQL AB và hiện thuộc sở hữu của Oracle Corporation. MySQL sử dụng ngôn ngữ truy vấn có cấu trúc SQL (Structured Query Language) để thao tác và quản lý dữ liệu trong cơ sở dữ liệu.

Với khả năng xử lý nhanh, đáng tin cậy và dễ sử dụng, MySQL đã trở thành một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất thế giới, đặc biệt trong các ứng dụng web và hệ thống có quy mô từ nhỏ đến trung bình. MySQL cho phép lưu trữ và quản lý dữ liệu một cách có tổ chức thông qua các bảng (tables), với mối quan hệ rõ ràng giữa các bảng thông qua khóa chính (primary key) và khóa ngoại (foreign key). Hệ thống hỗ trợ các thao tác như thêm, sửa, xóa, truy vấn dữ liệu hiệu quả và bảo mật cao.

2.3.1.2. Các tính năng của MySQL

MySQL cung cấp nhiều tính năng mạnh mẽ, bao gồm:

- Hỗ trợ đa ngôn ngữ lập trình: MySQL hỗ trợ nhiều ngôn ngữ lập trình khác nhau như C, C++, Python, Perl, Java, PHP,... giúp các nhà phát triển dễ dàng tích hợp với nhiều nền tảng và môi trường phát triển khác nhau.
- Tính năng ACID: MySQL tuân thủ các nguyên tắc ACID (Atomicity, Consistency, Isolation, Durability), đảm bảo tính toàn vẹn và nhất quán của dữ liệu trong quá trình thực hiện giao dịch, giảm thiểu rủi ro sai sót dữ liệu khi có sự cố.
- Tính năng sao lưu và phục hồi: Hệ thống cho phép người quản trị thực hiện sao lưu định kỳ và phục hồi dữ liệu dễ dàng, giúp giảm thiểu mất mát dữ liệu trong trường hợp lỗi phần cứng hoặc tấn công mạng.

- Hỗ trợ khóa ngoại: MySQL cho phép định nghĩa khóa ngoại để duy trì mối quan hệ giữa các bảng trong cơ sở dữ liệu, đảm bảo tính toàn vẹn tham chiếu và hỗ trợ thiết kế cơ sở dữ liệu chuẩn hóa.

- Khả năng mở rộng cao: MySQL hỗ trợ xử lý dữ liệu với dung lượng lớn, tối ưu cho các ứng dụng từ nhỏ đến quy mô doanh nghiệp, đồng thời dễ dàng mở rộng bằng cách phân mảnh (sharding) hoặc nhân bản (replication).

- Replication (Sao chép dữ liệu): MySQL hỗ trợ cơ chế nhân bản dữ liệu master-slave hoặc master-master, giúp phân tải truy vấn, tăng tính sẵn sàng và khả năng dự phòng khi hệ thống gặp sự cố.

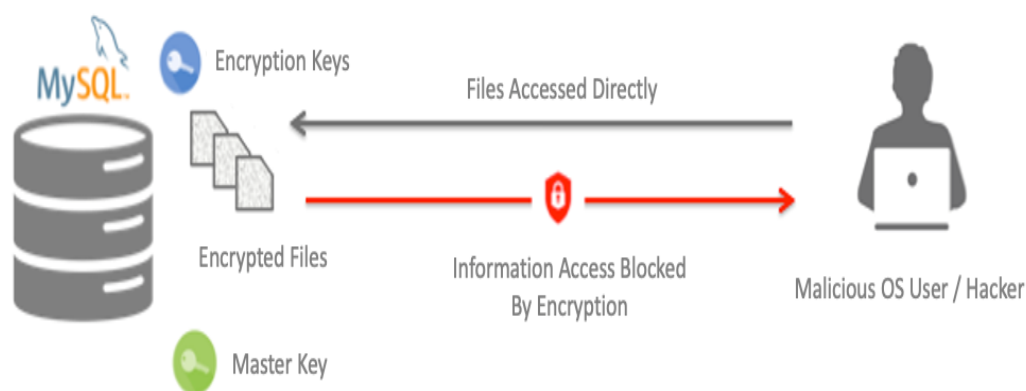
- Hỗ trợ chỉ mục (Indexing): MySQL cung cấp nhiều loại chỉ mục như B-Tree, Hash Index,... giúp tăng tốc độ truy vấn dữ liệu và cải thiện hiệu suất của hệ thống.

- Bảo mật dữ liệu: MySQL hỗ trợ cơ chế phân quyền chi tiết (GRANT, REVOKE), mã hóa kết nối SSL/TLS, xác thực người dùng, đảm bảo an toàn dữ liệu trước các truy cập trái phép.

- Khả năng tương thích đa nền tảng: MySQL có thể chạy trên nhiều hệ điều hành khác nhau như Windows, Linux, macOS, giúp triển khai linh hoạt trong nhiều môi trường.

- Hỗ trợ giao dịch (Transactions) và lưu trữ tạm thời: Với các engine như InnoDB, MySQL hỗ trợ giao dịch phức tạp, rollback, và bảng tạm (temporary tables) để xử lý dữ liệu hiệu quả.

- Cộng đồng và hệ sinh thái lớn: MySQL được hỗ trợ bởi một cộng đồng phát triển mạnh mẽ và có nhiều công cụ quản trị đi kèm như MySQL Workbench, phpMyAdmin giúp quản lý cơ sở dữ liệu trực quan và tiện lợi..



Hình 2.2: Cơ chế mã hóa dữ liệu trong MySQL [15]

2.3.1.3. Cấu trúc

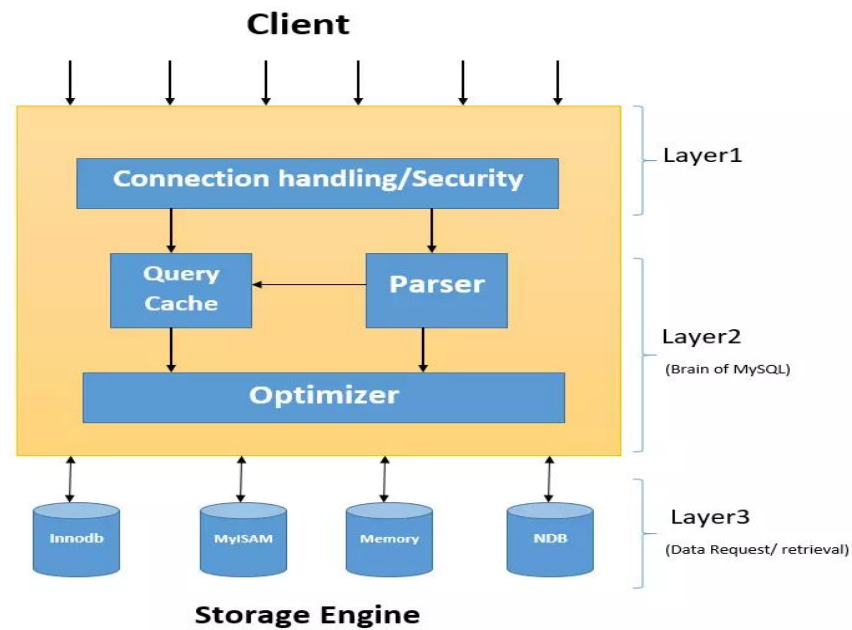
MySQL Server (Máy chủ MySQL): Đây là thành phần trung tâm thực thi các thao tác trên cơ sở dữ liệu. Nó bao gồm nhiều bộ phận quan trọng như:

- Connection Manager quản lý các kết nối từ client tới server [10].
- SQL Parser and Optimizer: Phân tích câu lệnh SQL, kiểm tra cú pháp, tối ưu truy vấn để thực thi hiệu quả nhất.
- Query Execution Engine: Thực thi các truy vấn đã được tối ưu, lấy dữ liệu từ storage engine [15].
- Storage Engine Interface: Giao tiếp với các engine lưu trữ dữ liệu vật lý (InnoDB, MyISAM, Memory, vv).

Client (Máy khách): là các ứng dụng hoặc công cụ kết nối tới server để gửi yêu cầu, nhận kết quả và tương tác với cơ sở dữ liệu. Ví dụ: MySQL Workbench, ứng dụng web, hoặc ứng dụng backend.

Storage Engines (Bộ máy lưu trữ): MySQL hỗ trợ nhiều storage engine khác nhau, mỗi engine có cách lưu trữ và xử lý dữ liệu riêng:

- InnoDB: Engine mặc định, hỗ trợ transaction, khóa hàng (row-level locking), và tính toàn vẹn dữ liệu.
- MyISAM: Engine truyền thống, nhanh cho các truy vấn đọc nhiều, nhưng không hỗ trợ transaction.
- Memory: Lưu dữ liệu trên RAM, tốc độ truy cập rất nhanh nhưng dữ liệu mất khi server tắt.
- CSV, Archive, NDB Cluster: Các engine đặc biệt phục vụ mục đích khác nhau như lưu trữ dạng CSV, lưu trữ nén hoặc dùng trong cluster.



Hình 2.3. Cấu trúc của MySQL [15]

2.3.1.4. Cách hoạt động

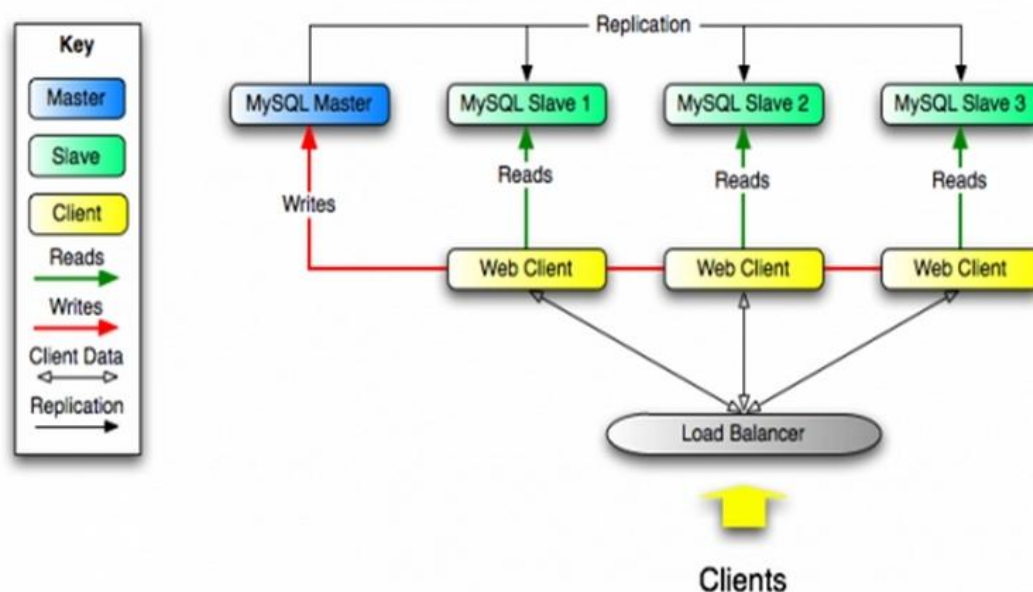
MySQL hoạt động theo mô hình client-server, trong đó client gửi các truy vấn đến server để thực thi các câu lệnh SQL và lấy dữ liệu. Dữ liệu được lưu trữ trong các cơ sở dữ liệu, trong đó mỗi cơ sở dữ liệu có thể chứa nhiều bảng với quan hệ khác nhau.

MySQL cung cấp các công cụ quản lý cơ sở dữ liệu để tạo, chỉnh sửa và xóa các cơ sở dữ liệu và bảng.

Các câu lệnh SQL được sử dụng để truy xuất và xử lý dữ liệu trong MySQL. Các lệnh này bao gồm:

- SELECT để lấy dữ liệu
- INSERT để chèn dữ liệu mới
- UPDATE để cập nhật dữ liệu đã có
- DELETE để xóa dữ liệu

Ngoài ra, MySQL cũng cung cấp các câu lệnh để quản lý cơ sở dữ liệu như CREATE để tạo mới cơ sở dữ liệu hoặc bảng, ALTER để thay đổi cấu trúc của bảng và DROP để xóa cơ sở dữ liệu hoặc bảng [19].



Hình 2.4: Mô hình hoạt động của MySQL [15]

2.4. OpenAI SDK

2.4.1. Khái niệm

OpenAI SDK là bộ thư viện chính thức giúp lập trình viên dễ dàng tích hợp các mô hình trí tuệ nhân tạo mạnh mẽ như GPT-3.5, GPT-4, DALL·E, Whisper... vào ứng dụng thông qua các lời gọi API đơn giản và linh hoạt. SDK này được hỗ trợ chính thức bởi OpenAI và có thể tích hợp với Python, Node.js và nhiều ngôn ngữ khác.

2.4.2. Cách thức hoạt động của mô hình GPT trong OpenAI SDK

Mô hình GPT (viết tắt của Generative Pre-trained Transformer) là một loại mô hình ngôn ngữ lớn (Large Language Model – LLM) được huấn luyện trước trên tập dữ liệu văn bản khổng lồ nhằm học cách dự đoán và sinh ra ngôn ngữ tự nhiên một cách mạch lạc và hợp ngữ cảnh. Đây là một sản phẩm tiên tiến của công nghệ trí tuệ nhân tạo, được phát triển bởi OpenAI nhằm mục đích mô phỏng khả năng hiểu và phản hồi ngôn ngữ như con người.

Trong môi trường phát triển ứng dụng, OpenAI cung cấp SDK (Software Development Kit) chính thức, giúp các nhà lập trình tích hợp trực tiếp mô hình GPT vào hệ thống của họ thông qua việc gọi API. Mô hình GPT hoạt động dưới dạng mô hình sinh ngôn ngữ, tức là khi người dùng cung cấp một đoạn văn bản đầu vào gọi là prompt, mô hình sẽ tiếp nhận ngữ cảnh từ đoạn prompt này và sinh ra phần nội dung tiếp theo một cách logic, mạch lạc và hợp lý về mặt ngữ pháp, cấu trúc và ngữ nghĩa [8].

Mô hình GPT không chỉ đơn thuần tạo ra văn bản mà còn có thể được điều chỉnh để phù hợp với từng bối cảnh cụ thể như giám khảo chấm điểm, trợ lý học tập, chuyên gia tư vấn hay chatbot dịch vụ khách hàng. Tính linh hoạt trong cách xây dựng prompt và ngữ cảnh hội thoại là một trong những yếu tố then chốt tạo nên hiệu quả vượt trội của GPT khi được tích hợp thông qua OpenAI SDK.

❖ **Ưu điểm**

Việc tích hợp mô hình GPT thông qua OpenAI SDK mang lại nhiều lợi thế rõ rệt cho các hệ thống xử lý ngôn ngữ tự nhiên, đặc biệt trong các ứng dụng như chấm điểm bài viết, tạo nội dung, hoặc hỗ trợ người dùng qua chatbot. Các ưu điểm nổi bật bao gồm:

- Hiệu quả ngôn ngữ vượt trội: Mô hình GPT-3.5-turbo thể hiện khả năng hiểu và sinh văn bản rất gần với cách viết của con người. Nó có thể phân tích, tổng hợp, và phản hồi lại các yêu cầu phức tạp bằng ngôn ngữ tự nhiên một cách logic, chính xác và mạch lạc.

- Khả năng tùy biến cao: Thông qua cấu trúc prompt linh hoạt, người phát triển có thể điều chỉnh vai trò, giọng điệu, mục tiêu của mô hình (ví dụ: giám khảo IELTS, trợ lý viết văn, giáo viên...) một cách chi tiết để phù hợp với từng bài toán cụ thể.

- Tích hợp đơn giản và nhanh chóng: SDK của OpenAI cung cấp giao diện lập trình rõ ràng, cho phép tích hợp trực tiếp vào các ứng dụng Python, backend (Flask, FastAPI), hoặc frontend thông qua API REST. Việc triển khai không đòi hỏi nhiều cấu hình phức tạp [9].

- Khả năng mở rộng tốt: Do OpenAI xử lý phía server, người dùng không cần triển khai mô hình cục bộ. Điều này cho phép mở rộng quy mô sử dụng mà không phải đầu tư hạ tầng phần cứng lớn.

- Tạo phản hồi có chiều sâu: Mô hình không chỉ đưa ra kết quả, mà còn kèm theo phân tích, gợi ý cải thiện, và ví dụ minh họa – điều đặc biệt hữu ích trong các ứng dụng giáo dục, đào tạo hoặc đánh giá kỹ năng viết.

- Liên tục cập nhật và cải tiến: Với các phiên bản mới như GPT-4 Turbo, OpenAI liên tục nâng cấp tốc độ, chi phí và khả năng của mô hình, giúp nhà phát triển hưởng lợi từ sự cải tiến công nghệ mà không cần can thiệp vào kiến trúc hệ thống cũ.

❖ **Nhược điểm:**

Mặc dù đem lại nhiều lợi ích, việc sử dụng mô hình GPT thông qua OpenAI SDK cũng tồn tại những hạn chế cần cân nhắc:

- Chi phí sử dụng cao: Mỗi lần gọi API sẽ tiêu tốn token, và với các mô hình như GPT-4, chi phí có thể tăng nhanh nếu không được kiểm soát kỹ. Điều này có thể trở thành rào cản trong các ứng dụng miễn phí hoặc quy mô lớn với ngân sách hạn chế.

- Không truy cập dữ liệu real-time: Mô hình GPT hoạt động dựa trên kiến thức đã được huấn luyện, không có khả năng tra cứu trực tiếp Internet hay truy cập cơ sở dữ liệu thời gian thực (trừ khi tích hợp thêm plugin hoặc công cụ bên ngoài).

- Khả năng “bịa” thông tin (hallucination): Trong một số trường hợp, mô hình có thể sinh ra thông tin không chính xác hoặc không có thật nhưng vẫn diễn đạt một cách rất tự tin. Đây là rủi ro tiềm ẩn nếu ứng dụng yêu cầu độ chính xác tuyệt đối.

- Phụ thuộc vào hạ tầng của OpenAI: Vì toàn bộ xử lý diễn ra phía server của OpenAI, nếu API bị gián đoạn hoặc server quá tải, hệ thống sẽ không hoạt động bình thường. Điều này làm giảm tính chủ động và kiểm soát của doanh nghiệp.

- Giới hạn độ dài văn bản (token limit): Mỗi mô hình có giới hạn số token tối đa cho một lần xử lý (ví dụ 128k token với GPT-4 Turbo), gây bất tiện khi xử lý văn bản dài hoặc hội thoại phức tạp nhiều lượt.

- Thiếu minh bạch trong quá trình suy luận: GPT là mô hình hộp đen (black-box), không giải thích rõ lý do tại sao lại đưa ra kết quả cụ thể nào đó. Điều này gây khó khăn khi cần kiểm tra logic nội bộ hoặc lý do chấm điểm cụ thể trong các ứng dụng kiểm định chất lượng.

2.5. Tổng quan về mô hình sử dụng – mô hình MVC

2.5.1. Tổng quan

MVC là viết tắt của Model-View-Controller. Cấu trúc Model-View-Controller (MVC) là một mẫu thiết kế (design pattern) tách ứng dụng thành ba thành phần logic chính: Model, View và Controller. Mỗi thành phần kiến trúc được xây dựng để xử lý các khía cạnh phát triển cụ thể của một ứng dụng.

2.5.2. Các thành phần trong mô hình MVC

❖ Model

Model là các thành phần của ứng dụng tương ứng với tất cả logic liên quan đến miền dữ liệu (data domain), hoặc nói ngắn gọn đây là phần back-end chứa tất cả logic

dữ liệu của ứng dụng. Dữ liệu ở đây có thể là dữ liệu đang được truyền giữa các thành phần View và Controller hoặc bất kỳ dữ liệu nào khác liên quan đến logic của doanh nghiệp. Nếu trạng thái của dữ liệu này thay đổi thì Model thường sẽ thông báo cho View (để màn hình có thể thay đổi khi cần) và đôi khi là Controller (nếu cần logic khác để cập nhật View). Thông thường, các đối tượng Model có thể truy xuất từ cơ sở dữ liệu, thao tác và lưu trữ trạng thái Model trong cơ sở dữ liệu.

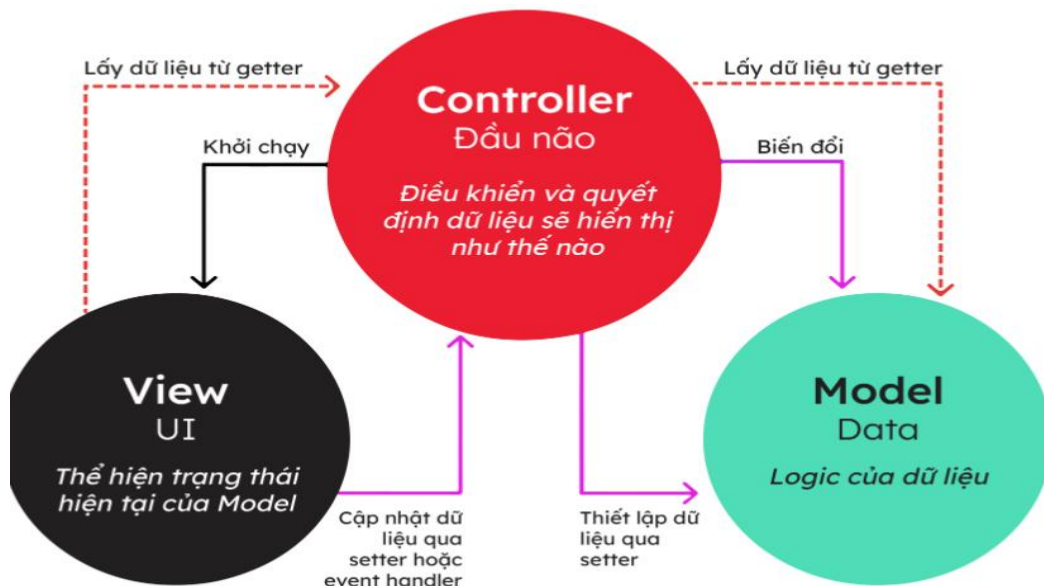
❖ View

View là các thành phần hiển thị giao diện người dùng (UI) của ứng dụng. Thông thường, giao diện người dùng này được tạo từ dữ liệu Model.

❖ Controller

Controller là các thành phần xử lý tương tác của người dùng để làm việc với Model (cập nhật logic dữ liệu) hoặc/ và với View (cập nhật hiển thị giao diện người dùng).

Trong ứng dụng MVC, Controller xử lý các giá trị chuỗi truy vấn và chuyển các giá trị này cho Model, từ đó Model sẽ truy vấn cơ sở dữ liệu bằng cách sử dụng các giá trị đó. View hiển thị thông tin do Controller xử lý và phản hồi đầu vào từ tương tác của người dùng.



Hình 2.5: Mô hình MVC - Cách dữ liệu và giao diện tương tác với nhau

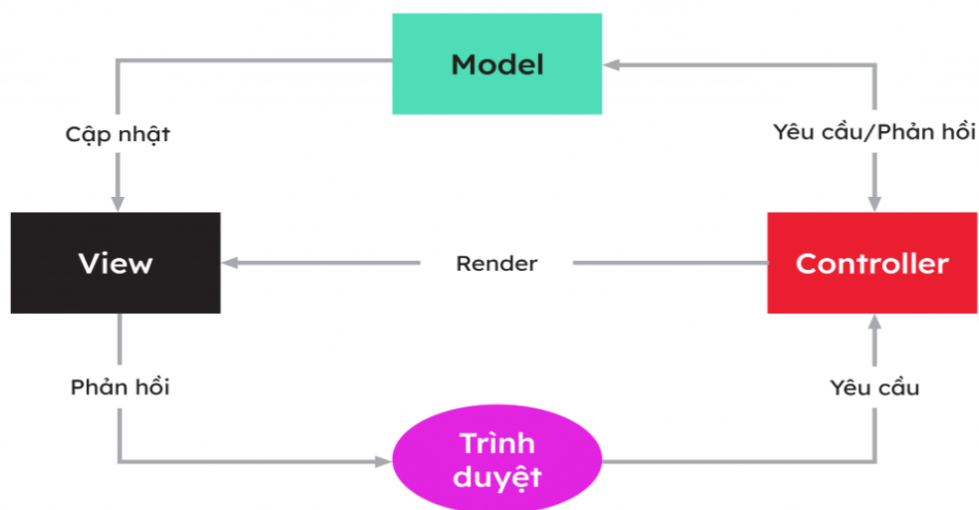
2.5.3. Chức năng

Mô hình MVC (Model – View – Controller) là một kiến trúc phần mềm phổ biến, giúp phân tách rõ ràng ba thành phần chính trong ứng dụng: dữ liệu, giao diện và điều

phối xử lý [3]. Thành phần Model chịu trách nhiệm quản lý dữ liệu, thực hiện các thao tác với cơ sở dữ liệu như thêm, sửa, xóa và chứa các logic nghiệp vụ cốt lõi. View là lớp giao diện người dùng, dùng để hiển thị thông tin từ Model và nhận các thao tác từ phía người dùng, như nhập bài viết, xem kết quả chấm điểm. Trong khi đó, Controller đóng vai trò trung gian, tiếp nhận yêu cầu từ View, xử lý điều hướng, gọi đến Model để xử lý dữ liệu và trả kết quả ngược lại cho View. Việc áp dụng kiến trúc MVC giúp hệ thống dễ dàng bảo trì, mở rộng và tách biệt rõ ràng giữa frontend và backend, từ đó nâng cao hiệu quả phát triển và khả năng kiểm thử từng phần của hệ thống.

2.5.4. Cách hoạt động của MVC

Mô hình MVC hoạt động dựa trên nguyên tắc phân chia trách nhiệm rõ ràng giữa ba thành phần: Model, View và Controller. Khi người dùng thực hiện một hành động trên giao diện (View), chẳng hạn như nhập bài viết hoặc nhấn nút gửi bài, yêu cầu này sẽ được chuyển đến Controller. Tại đây, Controller có nhiệm vụ xử lý logic điều hướng, xác thực yêu cầu và gọi đến Model để truy xuất hoặc cập nhật dữ liệu tương ứng. Sau khi Model xử lý xong dữ liệu (ví dụ: lưu bài viết, tính điểm, truy xuất phản hồi từ AI...), kết quả sẽ được gửi lại cho Controller. Cuối cùng, Controller truyền dữ liệu đó sang View, nơi dữ liệu được hiển thị lại cho người dùng dưới dạng giao diện trực quan. Nhờ đó, quá trình xử lý được tách biệt rõ ràng: giao diện chỉ hiển thị, Model tập trung vào dữ liệu, còn Controller điều phối toàn bộ luồng xử lý giữa hai thành phần còn lại. Cách hoạt động này giúp hệ thống dễ bảo trì, nâng cấp và mở rộng chức năng trong quá trình phát triển phần mềm.



Hình 2.6: Luồng hoạt động của mô hình MVC trong ứng dụng web

2.5.5. Ưu và nhược điểm của MVC

❖ Ưu điểm:

- Phân tách rõ ràng các thành phần: Giúp tổ chức mã nguồn rõ ràng: Model (xử lý dữ liệu), View (giao diện người dùng), Controller (điều khiển luồng dữ liệu). Dễ bảo trì và mở rộng ứng dụng.
- Tăng khả năng tái sử dụng: Một Model có thể được sử dụng lại với nhiều View khác nhau mà không cần viết lại logic xử lý dữ liệu.
- Dễ dàng kiểm thử (Test): Việc tách biệt các thành phần giúp dễ dàng viết unit test cho từng phần riêng biệt, đặc biệt là Model và Controller.
- Hỗ trợ làm việc nhóm hiệu quả: Các lập trình viên backend có thể làm việc với Model và Controller, trong khi lập trình viên frontend có thể làm việc độc lập với View.

❖ Nhược điểm:

- Phức tạp khi xây dựng dự án nhỏ: Với các dự án đơn giản, việc tách nhỏ theo MVC có thể làm tăng độ phức tạp và tốn công tổ chức code.
- Yêu cầu lập trình viên hiểu rõ mô hình: Cần hiểu cách vận hành và phân tách đúng vai trò từng thành phần, nếu không sẽ dễ nhầm lẫn hoặc viết sai kiến trúc.
- Giao tiếp giữa các thành phần có thể rối: Khi ứng dụng phát triển lớn, luồng dữ liệu giữa Model - View - Controller có thể trở nên phức tạp và khó kiểm soát.

2.6. Tổng quan về Writing và các tiêu chuẩn đánh giá

2.6.1. Tổng quan

Kỹ năng Writing (viết) là một trong bốn kỹ năng quan trọng (nghe, nói, đọc, viết) được đánh giá trong các kỳ thi chuẩn hóa tiếng Anh như IELTS, TOEFL, VSTEP (tiếng Anh theo khung năng lực ngoại ngữ 6 bậc của Việt Nam) và nhiều kỳ thi quốc tế khác. Đây là kỹ năng đòi hỏi người học phải vận dụng đồng thời nhiều yếu tố: ngữ pháp, từ vựng, cấu trúc câu, tư duy logic, khả năng triển khai ý tưởng và tính mạch lạc trong bài viết.

- Trong IELTS, Writing chiếm 25% tổng điểm và bao gồm 2 task: Task 1 (viết báo cáo mô tả số liệu hoặc quy trình) và Task 2 (viết bài luận nêu quan điểm, phân tích, hoặc tranh luận về một vấn đề xã hội).

- Trong VSTEP, kỹ năng Writing cũng được đánh giá qua hai phần: Phần đầu là viết thư hoặc email theo yêu cầu, phần thứ hai là viết bài luận ngắn về một chủ đề nhất định.

- Ngoài ra, trong các kỳ thi chuẩn hóa khác như TOEFL hay Cambridge English, kỹ năng viết cũng đóng vai trò then chốt để đánh giá khả năng diễn đạt và tư duy ngôn ngữ của thí sinh.

Kỹ năng này đặc biệt quan trọng trong học thuật và công việc, vì nó phản ánh khả năng truyền đạt thông tin, trình bày luận điểm rõ ràng, thuyết phục và chuyên nghiệp bằng tiếng Anh.

2.6.2. Các loại Writing

Trong các kỳ thi và môi trường học thuật, Writing thường được phân thành nhiều dạng khác nhau. Một số loại phổ biến bao gồm:

- Writing Task 1 (IELTS Academic): Yêu cầu viết báo cáo mô tả, so sánh, phân tích biểu đồ, bảng số liệu, hoặc mô tả quy trình, sơ đồ.

- Writing Task 1 (IELTS General): Viết thư (thư xin việc, thư phản nản, thư cảm ơn, thư yêu cầu thông tin).

- Writing Task 2 (IELTS): Viết bài luận (essay) về một vấn đề xã hội, bao gồm các dạng:

+ Opinion Essay (nêu quan điểm và bảo vệ quan điểm).

+ Discussion Essay (thảo luận hai quan điểm và đưa ra ý kiến cá nhân).

+ Problem-Solution Essay (nêu vấn đề và giải pháp).

+ Advantages-Disadvantages Essay (phân tích ưu và nhược điểm).

- Writing trong VSTEP: Bao gồm viết thư/email và viết bài luận ngắn từ 120 đến 250 từ.

- Các dạng Writing khác:

+ Academic Writing: Viết nghiên cứu, bài luận học thuật, báo cáo khoa học.

+ Creative Writing: Viết sáng tạo như truyện ngắn, thơ.

+ Business Writing: Viết email chuyên nghiệp, báo cáo công việc, đề xuất kinh doanh.

2.6.3. Các tiêu chuẩn đánh giá Writing

Mỗi kỳ thi có tiêu chí đánh giá riêng, nhưng nhìn chung, việc chấm điểm kỹ năng Writing dựa trên một số tiêu chí chính sau:

❖ IELTS

IELTS sử dụng 4 tiêu chí

- Task Achievement / Task Response:
 - + Task 1: Mức độ hoàn thành yêu cầu mô tả số liệu hoặc quy trình.
 - + Task 2: Trả lời đúng chủ đề, phát triển ý tưởng đầy đủ, lập luận rõ ràng.
- Coherence and Cohesion (Tính mạch lạc và liên kết): Sắp xếp ý tưởng hợp lý, sử dụng liên từ, câu nối tự nhiên.
- Lexical Resource (Từ vựng): Đa dạng từ vựng, sử dụng từ phù hợp với ngữ cảnh.
- Grammatical Range and Accuracy (Ngữ pháp): Cấu trúc câu đa dạng, chính xác, ít lỗi sai.

❖ VSTEP

VSTEP cũng chấm điểm Writing dựa trên các yếu tố tương tự IELTS, bao gồm:

- Hoàn thành yêu cầu đề bài: Đúng dạng bài (thư, email, bài luận), không lạc đề.
- Tổ chức và mạch lạc: Có mở bài, thân bài, kết bài rõ ràng; ý tưởng logic.
- Ngôn ngữ: Đa dạng từ vựng, sử dụng ngữ pháp chuẩn.
- Độ chính xác: Hạn chế lỗi chính tả, ngữ pháp.

Các kỳ thi khác (TOEFL, Cambridge, ...)

❖ TOEFL iBT:

- Writing có 2 phần: Integrated Writing (kết hợp đọc + nghe + viết) và Independent Writing (viết bài luận). Chấm dựa trên khả năng tích hợp thông tin và diễn đạt rõ ràng.
- Cambridge English: Đánh giá qua độ chính xác ngữ pháp, tính mạch lạc, và sự phù hợp với phong cách viết yêu cầu.

Nhìn chung, tiêu chí quan trọng trong tất cả các hệ thống đánh giá Writing là: đúng yêu cầu đề bài, triển khai ý tưởng hợp lý, sử dụng từ vựng và ngữ pháp chính xác, bài viết mạch lạc và có tính logic.



Hình 2.7: Tiêu chí chấm bài Writing (IELTS) [23]

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Mô tả bài toán

Trong bối cảnh kỳ thi IELTS ngày càng phổ biến, kỹ năng Writing được đánh giá là một trong những kỹ năng khó nâng cao nhất đối với người học tiếng Anh tại Việt Nam. Việc luyện viết đòi hỏi phản hồi chất lượng từ giám khảo, nhưng chi phí và thời gian chấm bài truyền thống lại không phù hợp với đại đa số người học.

Đề tài đặt ra bài toán xây dựng một hệ thống web có khả năng tiếp nhận bài viết IELTS (Task 1 và Task 2), phân tích và chấm điểm tự động theo tiêu chí chính thức của kỳ thi (Task Achievement, Coherence & Cohesion, Lexical Resource, Grammatical Range) [1], đồng thời đưa ra gợi ý cải thiện. Hệ thống cũng phải có khả năng quản lý tài khoản người dùng, phân quyền truy cập, và cung cấp giao diện quản trị cho người điều hành hệ thống.

3.2. Yêu cầu chức năng của hệ thống

3.2.1. Đối với người quản trị

Người quản trị có vai trò kiểm soát toàn bộ hệ thống, với các chức năng chính sau:

- Dashboard tổng quan:

- + Hiển thị tổng số người dùng.
- + Số lượng người dùng mới trong tháng.
- + Doanh thu từ các gói trả phí.
- + Tổng số lượng người dùng đang có gói active.
- + Danh sách hoạt động đăng ký mới nhất.

- Quản lý người dùng (User Management):

- + Danh sách tất cả người dùng kèm thông tin email, ngày đăng ký, trạng thái tài khoản.
- + Chức năng lọc, tìm kiếm theo tên/email.
- + Cập nhật vai trò (User/Admin), thay đổi trạng thái tài khoản (Active/Inactive), chỉnh sửa hoặc xóa tài khoản.

- Quản lý gói sử dụng (Order Management):

- + Xem danh sách tất cả các đơn đăng ký sử dụng gói (Free, Student, Unlimited).

+ Hiện thị Subscription ID, email người dùng, loại gói, trạng thái và thời gian sử dụng.

+ Chức năng xem chi tiết đơn đăng ký.

3.2.2. Đối với người dùng

Người dùng là đối tượng chính sử dụng hệ thống để luyện kỹ năng IELTS Writing. Họ có thể thực hiện các chức năng sau:

- Trang chủ hệ thống:

+ Hiện thị giới thiệu về BandBoost, mục tiêu sử dụng, và lời kêu gọi hành động (CTA).

+ Có nút truy cập nhanh vào viết bài, kiểm tra ngữ pháp, xem điểm.

- Viết bài IELTS Writing Task 1 và Task 2:

+ Người dùng có thể chọn loại đề (Task 1 hoặc Task 2) để nhập bài viết.

+ Hệ thống hiển thị bộ đếm thời gian (timer), cảnh báo giới hạn từ (word count), và thông báo về việc bị trừ điểm nếu dưới chuẩn (dưới 150 hoặc 250 từ).

+ Sau khi viết xong, người dùng nhấn nút "Continue to Analysis" để hệ thống chấm điểm bằng AI.

- Xem điểm và phản hồi chi tiết:

+ Kết quả chấm điểm hiển thị đầy đủ 4 tiêu chí của IELTS: Task Achievement, Coherence & Cohesion, Lexical Resource, Grammatical Range & Accuracy

+ Mỗi tiêu chí đi kèm nhận xét cụ thể, highlight điểm mạnh – điểm yếu.

+ Bài viết được hiển thị lại với các đoạn bị sai được highlight theo loại lỗi: sai ngữ pháp (Grammar), từ vựng chưa tốt (Vocabulary), vấn đề cấu trúc câu, đoạn (Structure)

+ Hệ thống cũng hiển thị điểm ban đầu, điểm sau khi trừ penalty (nếu có).

- Lịch sử bài viết & cộng điểm tổng hợp:

+ Người dùng có thể xem lại toàn bộ bài viết đã nộp theo thời gian.

+ Mỗi bài hiển thị đầy đủ điểm từng phần và điểm tổng.

+ Có một trang riêng để tính Combined Score từ điểm Task 1 và Task 2 theo công thức: $\text{Score} = \text{Task 1} \times 1/3 + \text{Task 2} \times 2/3$

+ Dữ liệu được trình bày rõ ràng, có thể so sánh sự tiến bộ theo thời gian.

- Trang lựa chọn & nâng cấp gói sử dụng:

❖ Các gói gồm:

- Free (0đ – giới hạn số bài),
- VIP (99,000đ/tháng – 50 bài, email hỗ trợ, gợi ý cải thiện),
- ProPlus (199,000đ – 200 bài, email hỗ trợ, gợi ý cải thiện, chatbox AI),
- Ultimate (499,000đ – không giới hạn, email hỗ trợ, gợi ý cải thiện, chatbox AI, hỗ trợ 24/7)

Mỗi gói liệt kê rõ quyền lợi đi kèm: AI scoring, feedback chi tiết, luyện không giới hạn, sample essay...

- Thanh toán qua VNPAY:
 - + Sau khi chọn gói, người dùng được chuyển tới giao diện thanh toán qua VNPAY.
 - + Hỗ trợ thanh toán bằng thẻ nội địa.
 - + Hiện thị đầy đủ thông tin đơn hàng, hạn thời gian thanh toán, mã đơn, tên người nhận.
- Trang quản lý tài khoản & đổi mật khẩu:
 - + Người dùng có thể cập nhật thông tin cá nhân, thay đổi mật khẩu (yêu cầu xác minh mật khẩu cũ).
 - + Cho phép bật/tắt thông báo qua email.
 - + Giao diện đơn giản, dễ sử dụng.

3.3. Yêu cầu phi chức năng của hệ thống

Bên cạnh các yêu cầu chức năng liên quan đến bài toán nghiệp vụ và tính năng, hệ thống BandBoost cũng cần đáp ứng các yêu cầu phi chức năng nhằm đảm bảo hiệu năng, bảo mật, tính mở rộng và trải nghiệm người dùng trong quá trình sử dụng. Các yêu cầu phi chức năng cụ thể bao gồm:

3.3.1. Hiệu năng

- Hệ thống phải có khả năng xử lý đồng thời nhiều người dùng truy cập và gửi bài viết mà không làm gián đoạn hoạt động.
- Thời gian phản hồi trung bình khi chấm điểm một bài viết (bao gồm gọi OpenAI GPT-3.5-turbo và xử lý kết quả) phải dưới 10 giây.
- Thời gian tải giao diện người dùng không vượt quá 2 giây đối với kết nối ổn định.

3.3.2. Bảo mật

- Hệ thống xác thực người dùng bằng JSON Web Token (JWT) và lưu trữ token an toàn trên trình duyệt (qua HTTP-only cookie hoặc localStorage mã hóa).
- Mật khẩu người dùng được mã hóa bằng bcrypt trước khi lưu trữ trong cơ sở dữ liệu.
- Truyền dữ liệu giữa frontend và backend qua HTTPS để đảm bảo an toàn dữ liệu.
- Hệ thống phân quyền truy cập: tài khoản người dùng chỉ có quyền gửi bài, xem kết quả cá nhân; tài khoản quản trị viên mới có quyền truy cập thống kê, quản lý tài khoản, đơn hàng.

3.3.3. Khả năng mở rộng

- Hệ thống được thiết kế tách biệt frontend (React) và backend (Flask) theo mô hình RESTful API, giúp dễ dàng mở rộng tính năng mới hoặc tích hợp thêm hệ thống bên ngoài (ví dụ: thanh toán, gửi email tự động...).
- Có thể nâng cấp số lượng bài viết mỗi ngày, số người dùng truy cập đồng thời, hoặc tích hợp thêm mô hình AI khác mà không ảnh hưởng đến cấu trúc tổng thể.

3.3.4. Dễ bảo trì

- Mã nguồn backend được chia thành các module controller rõ ràng (auth_controller.py, writing_controller.py, admin_controller.py...), giúp dễ dàng kiểm tra, cập nhật và mở rộng.
- Frontend sử dụng kiến trúc thư mục rõ ràng (pages, services, contexts, components...), thuận tiện cho teamwork và bảo trì sau này.

3.3.5. Tính thân thiện với người dùng

- Giao diện được thiết kế hiện đại, đơn giản, có chỉ dẫn rõ ràng cho người dùng không chuyên.
- Các cảnh báo về giới hạn từ vựng, thời gian, trạng thái gói... được hiển thị rõ ràng, dễ hiểu.
- Người dùng có thể dễ dàng xem lại kết quả, điểm số cũ, cập nhật thông tin cá nhân hoặc nâng cấp gói mà không cần hỗ trợ kỹ thuật.

3.3.6. Khả năng tích hợp

- Hệ thống hỗ trợ tích hợp với VNPAY để xử lý thanh toán qua thẻ nội địa.
- Các API thiết kế theo chuẩn RESTful, dễ tích hợp vào hệ thống khác như mobile app, chatbot hoặc dashboard nội bộ.
- Có thể mở rộng thêm API kết nối với tổ chức giáo dục.

3.4. Phân tích thiết kế hệ thống

3.4.1. Đặc tả hệ thống

Hệ thống BandBoost được xây dựng theo kiến trúc client-server hiện đại, gồm hai phần chính:

- Frontend: được phát triển bằng ReactJS + TypeScript, sử dụng Vite làm bundler. Đây là giao diện người dùng tương tác trực tiếp.
- Backend: được xây dựng bằng Flask RESTful API, dùng Flask-JWT để xác thực, SQLAlchemy để thao tác với cơ sở dữ liệu MySQL, và tích hợp OpenAI GPT để chấm điểm bài viết.

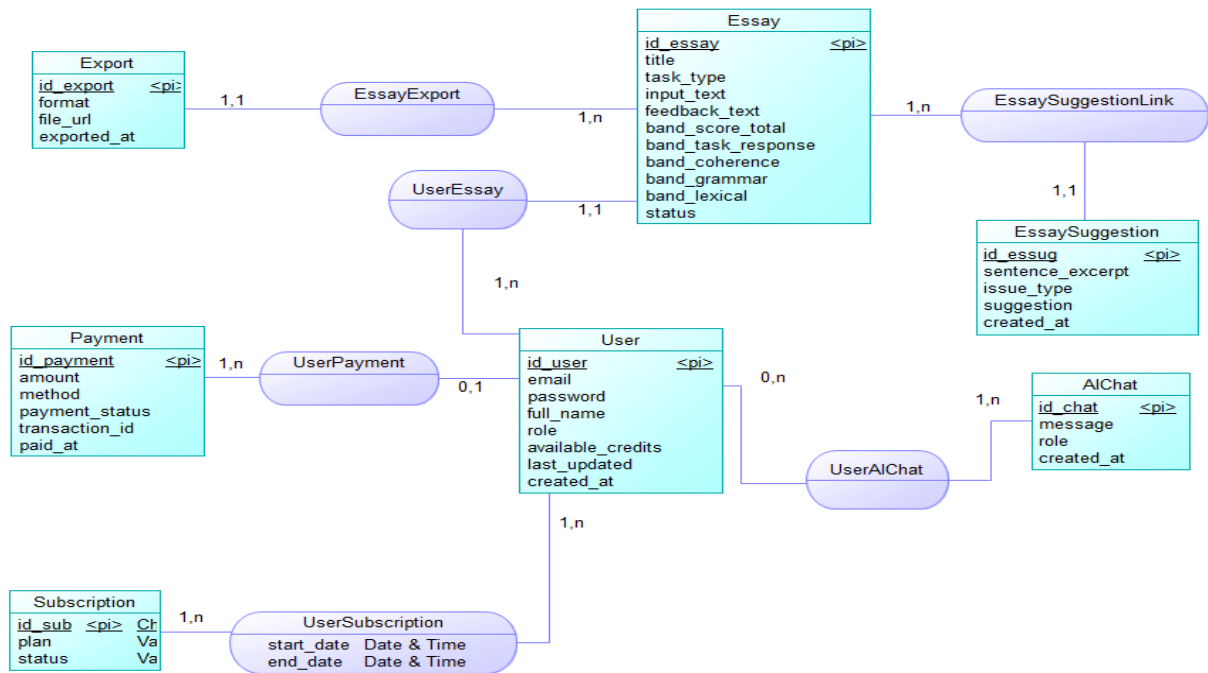
Ngoài ra, hệ thống tích hợp với cổng thanh toán VNPAY để xử lý các giao dịch nâng cấp gói sử dụng.

3.4.2. API chính giữa frontend và backend

Bảng 2: Liệt kê và mô tả các API

API Endpoint	Phương Thức	Mô tả
http://localhost:5173/api/auth/register	POST	Đăng ký tài khoản
http://localhost:5173/api/auth/login	POST	Đăng nhập
http://localhost:5173/writing	POST	Gửi bài viết, gọi GPT và nhận điểm
http://localhost:5173/writing/scores	GET	Lấy danh sách các bài đã nộp
http://localhost:5173/plans	POST	Nâng cấp gói
http://localhost:5173/history	GET	Xem lịch sử làm bài
http://localhost:5173/mocktest	GET	Xem các đề luyện thi
http://localhost:5173/mocktest/start	POST	Bắt đầu thi thử
http://localhost:5173/mocktest/submit	POST	Nộp bài và chấm điểm
http://localhost:5173/mocktest/results	GET	Xem kết quả thi

3.4.3. Mô hình thực thể quan hệ (ERD)



Hình 3.1: Mô hình ERD

❖ Mô tả thực thể và mối quan hệ:

Bảng 3: Mô tả các thực thể và mối quan hệ

STT	Tên thực thể/ mối quan hệ	Chức năng
1	User	<ul style="list-style-type: none"> - Thực thể trung tâm, lưu thông tin người dùng (email, mật khẩu, tên, vai trò, credit...). - Một User có thể viết nhiều bài luận (Essay), thực hiện nhiều thanh toán (Payment), đăng ký nhiều gói (Subscription), tương tác với AI (AIChat) và nhận nhiều kết quả chấm điểm bài viết (EssayScore).
2	Essay	<ul style="list-style-type: none"> - Lưu bài viết của người dùng, gồm tiêu đề, đề bài, nội dung bài viết, phản hồi, điểm tổng và điểm từng tiêu chí chấm

STT	Tên thực thể/ mối quan hệ	Chức năng
		IELTS. - Mỗi Essay thuộc về một User và có thể liên kết với nhiều Export, Suggestion và EssayScore.
3	EssaySuggestion	- Gợi ý chỉnh sửa từng lỗi trong bài viết. - Mỗi Suggestion liên kết với một Essay thông qua EssaySuggestionLink.
4	EssayScore	- Lưu kết quả chấm điểm chi tiết cho mỗi bài viết IELTS, sử dụng AI. - Gồm điểm số từng tiêu chí (Task Achievement, Coherence & Cohesion, Lexical Resource, Grammatical Range), điểm trung bình (overall_score), phản hồi chi tiết từng mục, và thời gian chấm.
5	Export	Ghi lại các lần xuất bài viết ra file (PDF, DOCX...), gắn với mỗi bài Essay thông qua EssayExport.
6	Payment	- Lưu thông tin thanh toán của người dùng (số tiền, phương thức, trạng thái, mã giao dịch...), liên kết với User thông qua UserPayment.
7	Subscription	- Ghi nhận thông tin về gói dịch vụ (plan), tình trạng gói, liên kết với User thông qua - UserSubscription kèm thời gian bắt đầu và kết thúc.

STT	Tên thực thể/ mối quan hệ	Chức năng
8	AiChat	Ghi lại các cuộc hội thoại giữa người dùng và hệ thống AI, gắn với User qua bảng UserAiChat.

- Mô tả thông tin bảng dữ liệu

❖ **Bảng User**

- Chức năng: quản lý thông tin người dùng bao gồm tài khoản, vai trò (admin hoặc user), và thời gian tạo tài khoản.

- Thông tin lưu trữ:

Bảng 4: Mô tả cấu trúc bảng User

STT	Tên trường	Kiểu dữ liệu	Mô tả	Loại khóa
1	id_user	char(10)	Mã định danh người dùng	PK
2	email	varchar(255)	Địa chỉ email người dùng	Unique
3	password	text	Mật khẩu đã mã hóa	
4	full_name	varchar(50)	Họ tên đầy đủ	
5	role	varchar(20)	Vai trò (admin/user)	
6	created_at	timestamp	Thời điểm tạo tài khoản	

❖ **Bảng Essay**

- Chức năng: lưu trữ nội dung các bài viết IELTS của người dùng để chấm điểm và nhận phản hồi
- Thông tin lưu trữ:

Bảng 5: Mô tả cấu trúc bảng Essay

STT	Tên trường	Kiểu dữ liệu	Mô tả	Loại khóa
1	id_essay	char(10)	Mã định danh bài viết	PK
2	id_user	char(10)	Mã người dùng viết bài	FK
3	title	varchar(255)	Tiêu đề bài viết	
4	task_type	varchar(10)	Loại task (task1/task2)	
5	input_text	text	Nội dung bài viết gốc	

STT	Tên trường	Kiểu dữ liệu	Mô tả	Loại khóa
6	feedback_text	text	Phản hồi tổng quát	
7	band_score_total	float	Điểm tổng	
8	band_task_response	float	Điểm tiêu chí Task Response	
9	band_coherence	float	Điểm tiêu chí Coherence & Cohesion	
10	band_grammar	float	Điểm tiêu chí Grammar	
11	band_lexical	float	Điểm tiêu chí Lexical Resource	
12	status	varchar(20)	Trạng thái xử lý bài viết	

❖ **Bảng EssaySuggestion**

- Chức năng: lưu gợi ý chỉnh sửa chi tiết cho từng bài viết gồm lỗi ngữ pháp, từ vựng, mạch lạc và logic.
- Thông tin lưu trữ:

Bảng 6: Mô tả cấu trúc bảng EssaySuggestion

STT	Tên trường	Kiểu dữ liệu	Mô tả	Loại khóa
1	id_essug	char(10)	ID góp ý	PK
2	id_essay	char(10)	Liên kết bài viết	FK
3	sentence_excerpt	text	Trích đoạn cần góp ý	
4	issue_type	varchar(20)	Loại lỗi (ngữ pháp, từ vựng...)	
5	suggestion	text	Gợi ý cải thiện	

❖ **Bảng AIChat:**

- Chức năng: lưu trữ nội dung hội thoại giữa người dùng và trợ lý AI (OpenAI GPT) để tư vấn học tập.
- Thông tin lưu trữ:

Bảng 7: Mô tả cấu trúc bảng AIChat

STT	Tên trường	Kiểu dữ liệu	Mô tả	Loại khóa
1	id_chat	char(10)	Mã định danh tin nhắn	PK
2	id_user	char(10)	Người gửi	FK
3	message	text	Nội dung tin nhắn	
4	role	varchar(20)	Vai trò: user/assistant	
5	created_at	timestamp	Thời điểm gửi	

❖ **Bảng Export:**

- Chức năng: lưu thông tin xuất file bài viết sang định dạng PDF hoặc DOCX.
- Thông tin lưu trữ:

Bảng 8: Mô tả cấu trúc bảng Export

STT	Tên trường	Kiểu dữ liệu	Mô tả	Loại khóa
1	id_export	char(10)	Mã export	PK
2	id_essay	char(10)	Liên kết bài viết	FK
3	format	varchar(10)	Định dạng file (pdf, docx)	
4	file_url	text	Đường dẫn file export	

❖ **Bảng Payment:**

- Chức năng: lưu trữ thông tin giao dịch thanh toán của người dùng cho các gói nâng cấp hoặc mua điểm.
- Thông tin lưu trữ

Bảng 9: Mô tả cấu trúc bảng Payment

STT	Tên trường	Kiểu dữ liệu	Mô tả	Loại khóa
1	id_payment	char(10)	Mã giao dịch	PK
2	id_user	char(10)	Người thanh toán	FK
3	amount	decimal(10,2)	Số tiền giao dịch	
4	method	varchar(20)	Phương thức thanh toán	
5	payment_status	varchar(50)	Trạng thái thanh toán	

6	transaction_id	varchar(255)	Mã giao dịch duy nhất	Unique
7	paid_at	timestamp	Thời điểm thanh toán	

❖ **Bảng Subscription:**

- Chức năng: Quản lý thông tin đăng ký gói sử dụng của người dùng (Free, Pro, ProPlus, VIP).
- Thông tin lưu trữ:

Bảng 10: Mô tả cấu trúc bảngSubscription

STT	Tên trường	Kiểu dữ liệu	Mô tả	Loại khóa
1	id_sub	char(10)	Mã đăng ký	PK
2	id_user	char(10)	Người đăng ký	FK
3	plan	varchar(20)	Tên gói (free/student...)	
4	status	varchar(20)	Trạng thái gói	
5	start_date	datetime	Ngày bắt đầu	
6	end_date	datetime	Ngày kết thúc	
7	created_at	timestamp	Ngày tạo đăng ký	

❖ **Bảng UserCredit**

- Chức năng: Quản lý số lượng điểm (credits) hiện có của mỗi người dùng dùng để chấm bài viết.
- Thông tin lưu trữ:

Bảng 11: Mô tả cấu trúc bảng UserCredit

STT	Tên trường	Kiểu dữ liệu	Mô tả	Loại khóa
1	id_user	char(10)	ID người dùng	PK/FK
2	available_credits	int	Số lượng credit còn lại	
3	last_updated	timestamp	Ngày cập nhật	

❖ **Bảng WritingScore**

- Chức năng: Lưu kết quả chấm điểm chi tiết cho mỗi bài viết IELTS sử dụng AI.

- Thông tin lưu trữ:

Bảng 12: Mô tả cấu trúc bảng WritingScore

STT	Tên trường	Kiểu dữ liệu	Mô tả	Loại khóa
1	id	char(36)	Mã điểm viết	PK
2	user_id	char(36)	Người viết bài	FK
3	task_type	varchar(20)	Loại bài viết: task1 hoặc task2	
4	essay_text	text	Nội dung bài viết gốc	
5	task_achievement	float	Điểm tiêu chí: Task Achievement	
6	coherence_cohesion	float	Điểm tiêu chí: Coherence & Cohesion	
7	lexical_resource	float	Điểm tiêu chí: Lexical Resource	
8	grammatical_range	float	Điểm tiêu chí: Grammatical Range	
9	overall_score	float	Điểm trung bình các tiêu chí	
10	task_achievement_feedback	text	Phản hồi chi tiết tiêu chí Task Achievement	
11	coherence_cohesion_feedback	text	Phản hồi chi tiết tiêu chí Coherence & Cohesion	
12	lexical_resource_feedback	text	Phản hồi chi tiết tiêu chí Lexical Resource	
13	grammatical_range_feedback	text	Phản hồi chi tiết tiêu chí Grammatical Range	
14	created_at	timestamp	Thời điểm chấm điểm	

3.4.4. Sơ đồ use-case

Actor: Admin (Người quản trị hệ thống): Là người có toàn quyền kiểm soát hệ thống bao gồm: quản lý người dùng, theo dõi hoạt động đăng ký và giám sát.

❖ Mô tả:

- Quản lý người dùng

+ Mục đích: Cho phép admin thực hiện các thao tác liên quan đến người dùng hệ thống.

+ Bao gồm các hành động phụ (<<include>>) như:

+ Xem danh sách người dùng: Hiển thị toàn bộ danh sách người dùng kèm thông tin cơ bản.

+ Tìm kiếm / lọc: Cho phép admin lọc người dùng theo tên, trạng thái.

+ Sửa thông tin người dùng: Cập nhật thông tin cá nhân.

+ Thay đổi vai trò người dùng: Chuyển đổi giữa vai trò User/Admin.

+ Thay đổi trạng thái tài khoản: Kích hoạt hoặc vô hiệu hóa tài khoản người dùng.

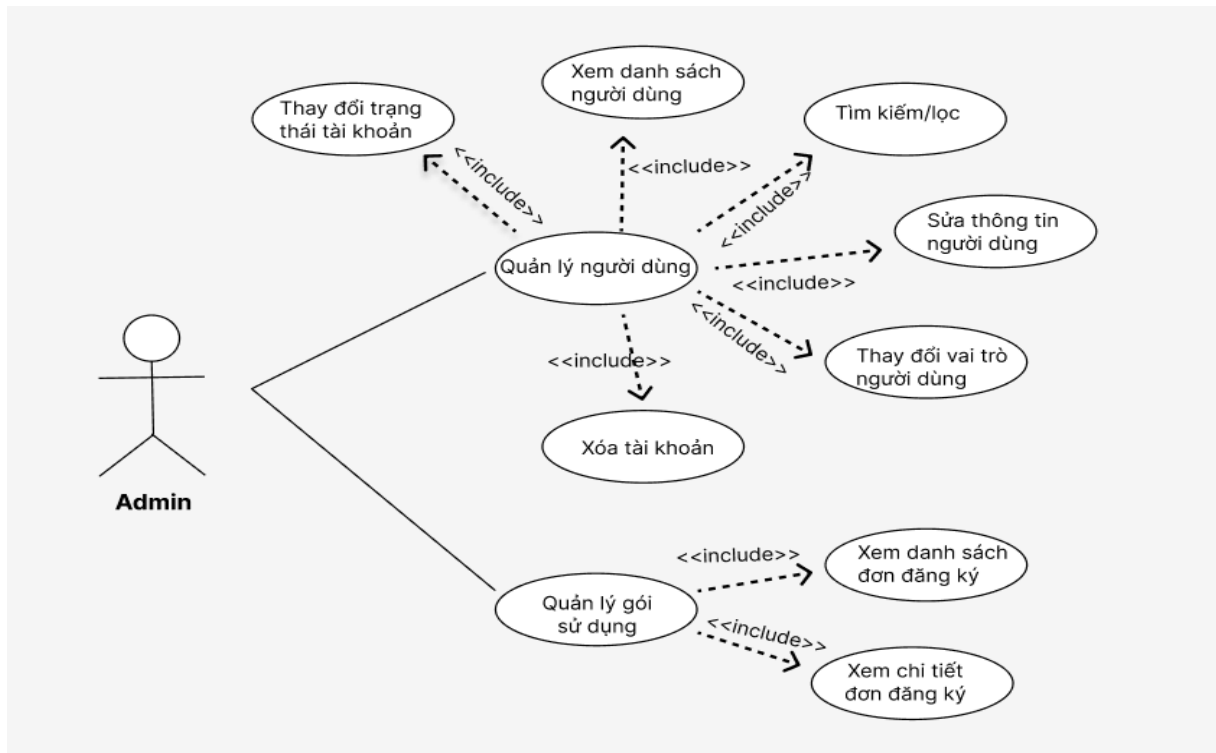
+ Xóa tài khoản: Xóa tài khoản khỏi hệ thống.

- Quản lý gói sử dụng

+ Mục đích: Giúp admin theo dõi và quản lý các gói đăng ký người dùng đã mua.

+ Xem danh sách đơn đăng ký: Hiển thị toàn bộ danh sách các đơn mua gói (Free, Pro, Vip...).

+ Xem chi tiết đơn đăng ký: Hiển thị thông tin đầy đủ của từng đơn: loại gói, thời gian bắt đầu – kết thúc, trạng thái, người mua...



Hình 3.2: Sơ đồ use case - actor: Admin

Actor: User (Người dùng): Là người sử dụng chính của hệ thống để luyện kỹ năng IELTS Writing, xem phản hồi chấm điểm, quản lý tài khoản và thực hiện thanh toán gói dịch vụ.

❖ **Mô tả:**

Đăng nhập/Đăng ký: Cho phép người dùng truy cập hệ thống bằng tài khoản cá nhân hoặc tạo tài khoản mới.

- Quản lý tài khoản, bao gồm:

+ Cập nhật thông tin cá nhân: Thay đổi tên, email, ảnh đại diện...

- Luyện viết IELTS Task 1 & Task 2:

+ Chọn đề bài: Chọn đề viết thuộc Task 1 hoặc Task 2.

+ Nhập bài viết: Soạn thảo bài viết có hỗ trợ đếm từ và đếm giờ.

+ Gửi bài viết: Gửi bài cho hệ thống chấm điểm tự động bằng AI.

- Xem điểm và phản hồi: Cho phép người dùng xem kết quả chấm điểm với 4 tiêu chí của IELTS bao gồm:

+ Xem danh sách các bài viết đã nộp: Hiện thị tất cả các bài đã gửi.

+ Xem chi tiết: Hiện thị feedback AI, điểm từng tiêu chí, lỗi sai được highlight.

- Quản lý lịch sử và tổng hợp điểm

+ Xem và tính điểm: Tính điểm tổng hợp từ Task 1 và Task 2.

+ Theo dõi tiến bộ: So sánh sự cải thiện theo thời gian.

- Xem và nâng cấp gói sử dụng

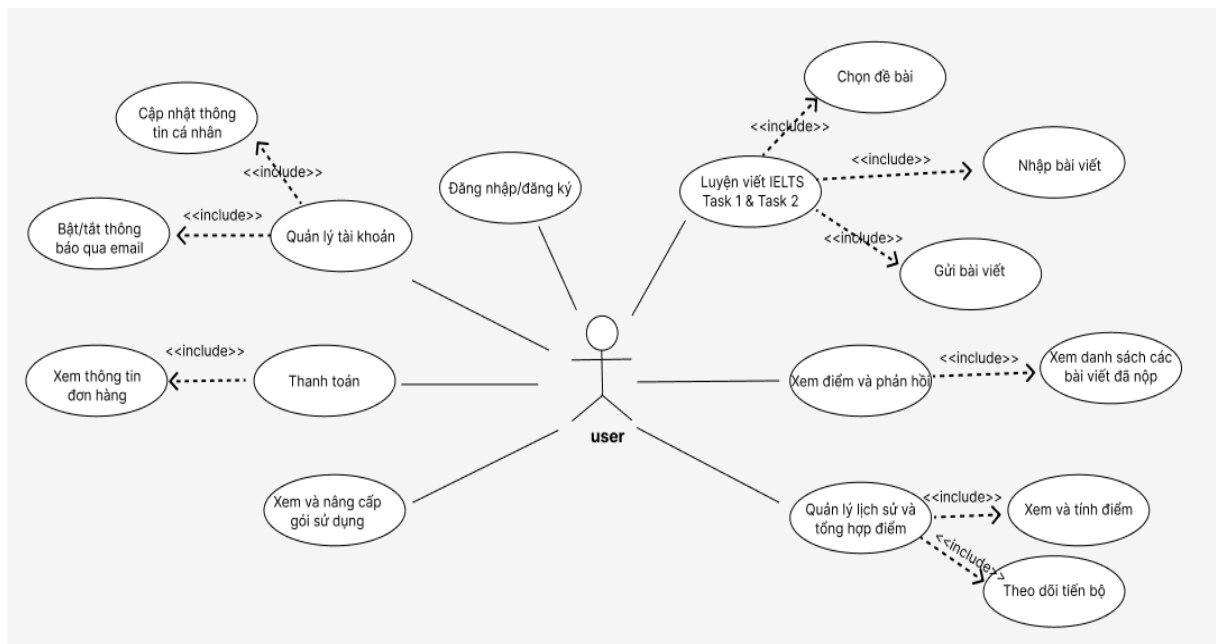
+ Hiện thị thông tin các gói (Free, Pro, Vip...) và quyền lợi tương ứng.

+ Cho phép chọn gói muốn nâng cấp.

- Thanh toán: thực hiện thanh toán qua VNPAY bao gồm xem thông tin đơn hàng (xem chi tiết đơn, trạng thái thanh toán, mã đơn)

Mô tả sơ đồ use case – actor chức năng dành cho Admin tại Hình 3. 3

(Trang sau)



Hình 3.3: Sơ đồ use case - actor: Người dùng

3.5. Cài đặt môi trường và công cụ sử dụng

3.5.1. Cài đặt công cụ lập trình chính - Visual Studio Code

- Bước 1: Cài VSCode tại trang: <https://code.visualstudio.com/>
- Bước 2: Cài đặt
 - + Mở file .exe vừa tải về (Windows) hoặc .dmg (macOS).
 - + Trong quá trình cài đặt, nên **tick các tùy chọn sau**: Add "Open with Code" to right-click menu, Add to PATH (giúp gọi code . từ terminal)
 - + Nhấn Install và chờ hoàn tất.
- Bước 3: Cài đặt các Extension cần thiết: Sau khi mở VSCode, truy cập tab Extensions (hoặc nhấn Ctrl + Shift + X) và cài các tiện ích: ESLint, Prettier - Code formatter...

3.5.2. Cài đặt môi trường chạy frontend

- Bước 1: Cài Node.js (có kèm npm) tại trang web: <https://nodejs.org>
- Bước 2: Tạo project React với TypeScript, ta dùng lệnh sau:

```
npx create-react-app bandboost-client --template typescript
cd bandboost-client
npm install
```

- Bước 3: Cài thêm thư viện cần thiết:

```
npm install axios react-router-dom @types/react-router-dom
npm install tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

- Bước 4: Cấu hình TailwindCSS trong tailwind.config.js:

```
content: ["./src/**/*..{js,ts,jsx,tsx}"]
```

- Bước 5: Khởi tạo môi trường React bằng lệnh:

```
npm start
```

3.5.3. Cài đặt môi trường chạy backend

- Bước 1: Cài Python tại trang web: <https://www.python.org/downloads/>
- Bước 2: Tạo virtual environment và kích hoạt:

```
python -m venv venv
venv\Scripts\activate # (Windows)
```

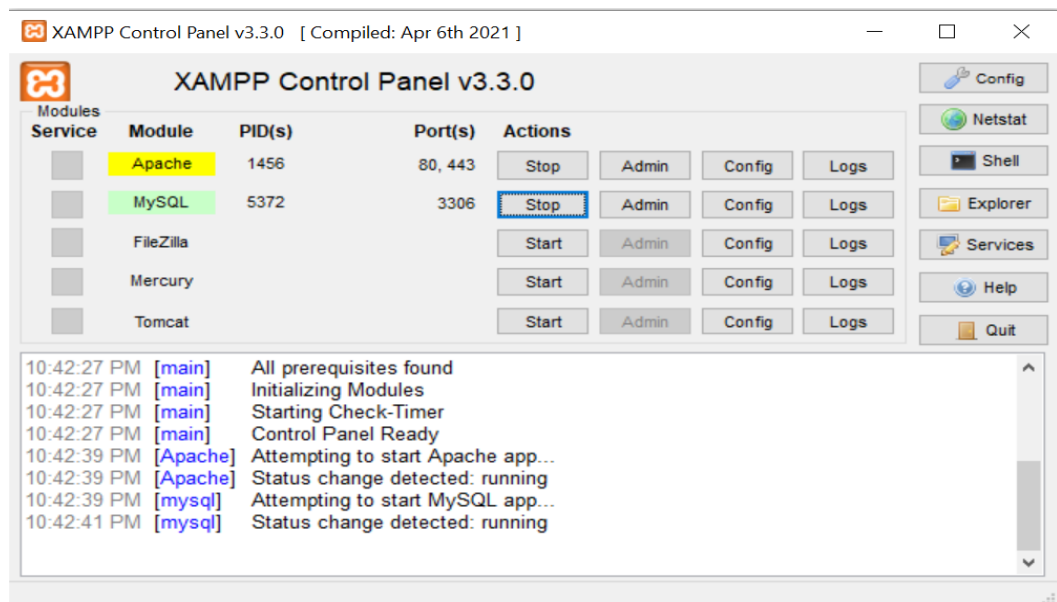
- Bước 3: Cài Flask và thư viện cần thiết:

```
pip install flask flask-cors mysql-connector-python openai python-dotenv
```

3.5.4. Xampp và MySQL

- Download Xampp tại <https://www.apachefriends.org>
- Start Apache và MySQL trong control panel:

Giao diện khởi động Xampp Control Panel tại Hình 3.4 (Trang sau)



Hình 3.4: Giao diện khởi động Xampp

- Gõ vào trình duyệt địa chỉ localhost. Hình sau sẽ hiện ra nếu cài đặt thành công:



Hình 3.5: Giao diện cài đặt thành công xampp

3.6. Thiết kế layout cho giao diện trang web

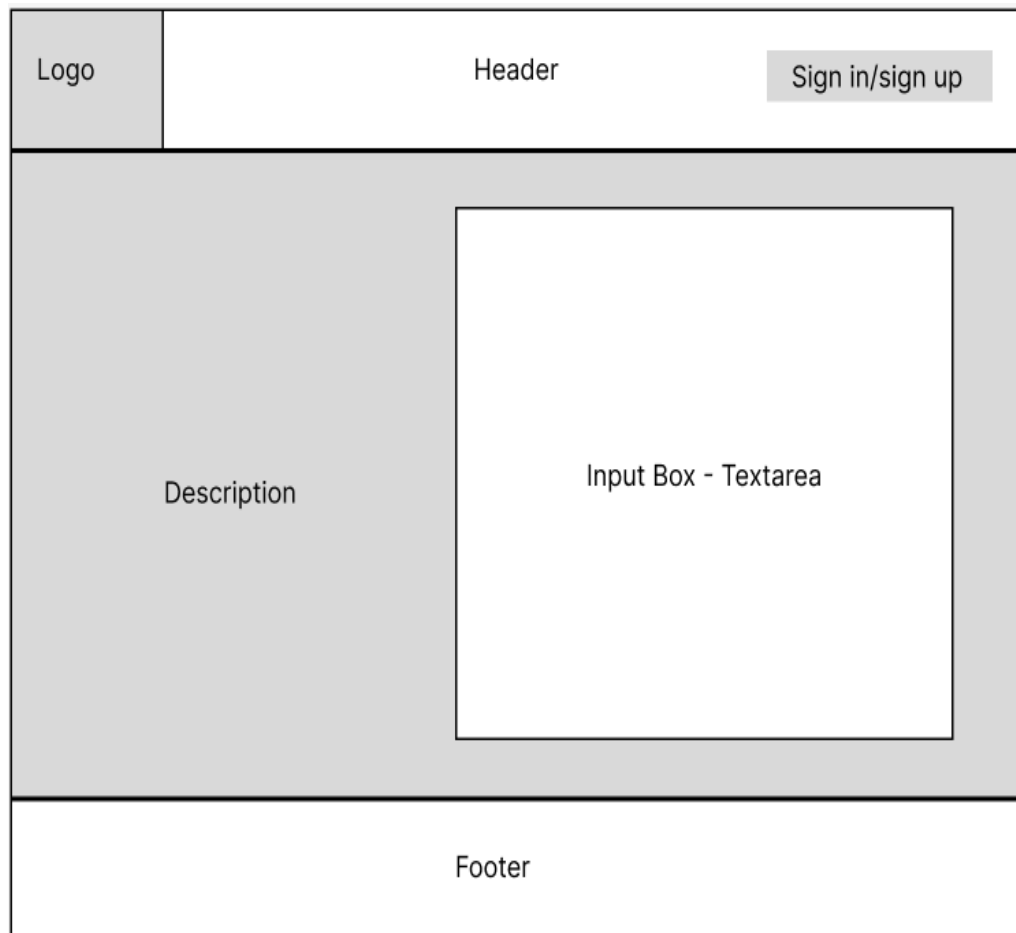
3.6.1. Layout giao diện người dùng

❖ Layout trang chủ:

Trang chủ được chia thành 3 phần chính:

- Header ở trên cùng gồm logo bên trái, tiêu đề ở giữa và nút Sign in/Sign up bên phải.

- Phần nội dung chính chia làm 2 cột: bên trái là phần Description mô tả, bên phải là Input Box (Textarea) để người dùng nhập nội dung.
- Footer ở cuối trang để chứa thông tin bổ sung.

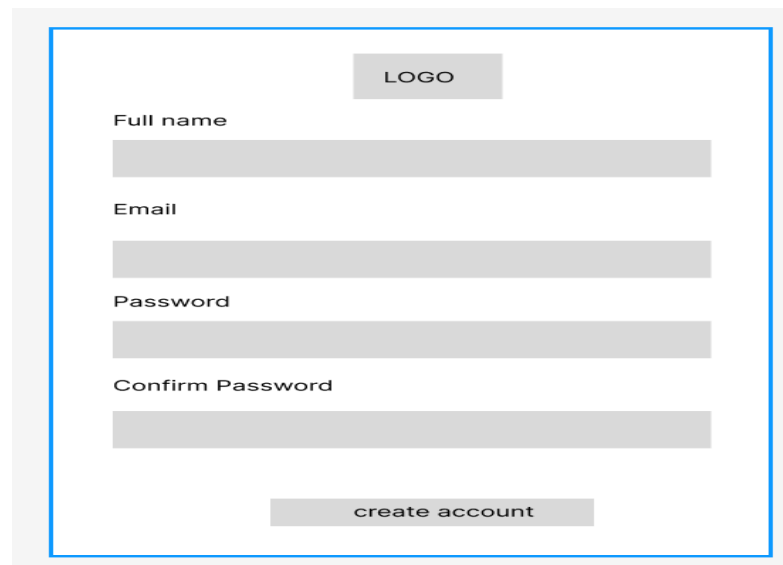


Hình 3.6: Layout giao diện trang chủ

❖ **Layout trang đăng ký:**

Trang đăng ký gồm:

- Logo đặt ở phía trên cùng, giúp nhận diện thương hiệu.
- Các trường nhập thông tin gồm:
 - + Full name: họ tên người đăng ký
 - + Email: để nhập gmail người đăng ký
 - + Password: để người dùng nhập mật khẩu
 - + Confirm Password để xác nhận lại mật khẩu.
- Các title được đặt trên các textbox để cho người dùng dễ dàng nhập thông tin
- Bên dưới là nút Create account để hoàn tất quá trình đăng ký.

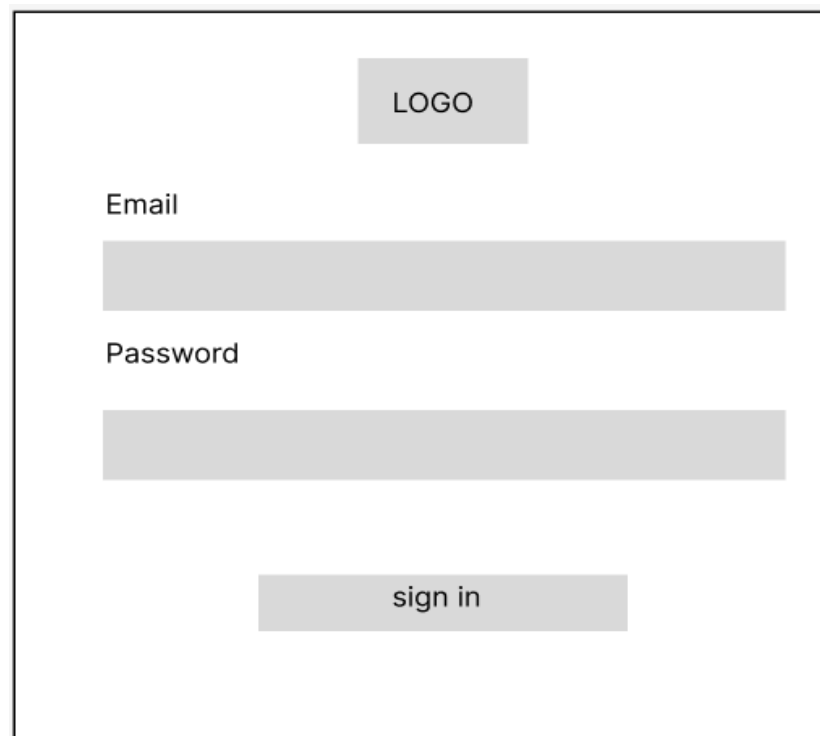


Hình 3.7: Layout trang đăng ký

❖ **Layout trang đăng nhập:**

Trang đăng nhập gồm:

- Logo đặt ở đầu trang để nhận diện thương hiệu.
- Hai trường nhập thông tin: Email và Password cho người dùng điền thông tin tài khoản.
- Nút Sign in bên dưới để gửi yêu cầu đăng nhập.



Hình 3.8: Layout trang đăng nhập

❖ **Layout trang chọn nhiệm vụ (Task) và nhập bài viết**

Trang nhiệm vụ để viết bài task 1 và task 2 gồm:

- Header phía trên có logo bên trái, tiêu đề ở giữa và tên tài khoản ở bên phải.
- Khu vực nội dung gồm:
 - + Tiêu đề Submit writing task.
 - + Phân chọn loại bài viết với hai tùy chọn task 1 và task 2.
 - + Textarea để người dùng nhập nội dung bài viết.
 - + Nút Button để gửi bài và nhận chấm điểm.
 - + Footer ở cuối trang chứa thông tin bổ sung.

The diagram illustrates the layout of the writing task submission page. It is structured as follows:

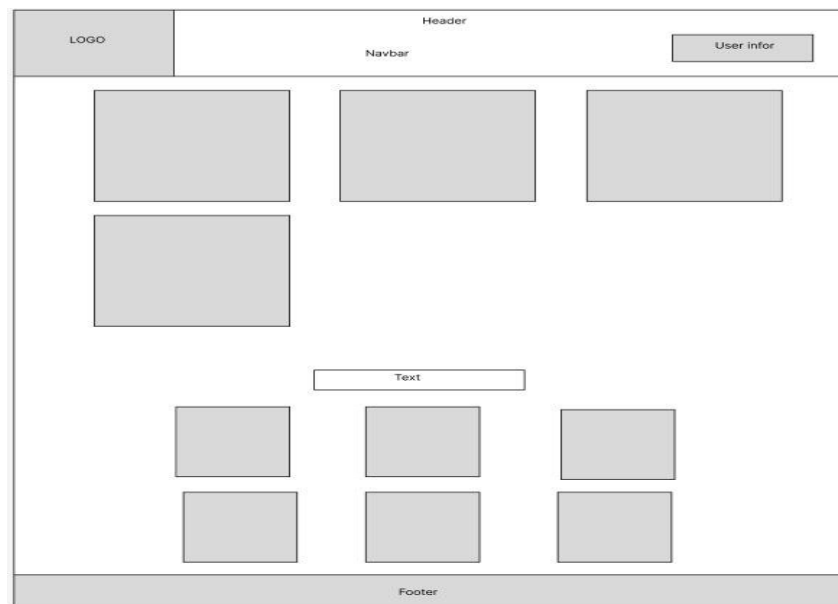
- Header:** A horizontal bar at the top containing a 'Logo' on the left, a 'Header' title in the center, and an 'account name' on the right.
- Main Content Area:** A large central box with a light gray background.
 - At the top of this area is the title 'Submit writing task'.
 - Below the title is a white box containing a 'Select task type' label and two buttons labeled 'task 1' and 'task 2'.
 - Below the task selection box is a large white 'textarea' for writing.
 - At the bottom right of the main content area is a 'Button' for submitting the task.
- Footer:** A horizontal bar at the bottom of the page labeled 'Footer'.

Hình 3.9: Layout trang nhập bài viết và gửi chấm điểm

❖ **Layout trang nâng cấp các gói dịch vụ:**

Trang nâng cấp các gói để mở thêm tính năng gồm:

- Header ở trên cùng với logo bên trái, thanh điều hướng (Navbar) ở giữa và thông tin người dùng bên phải.
- Khu vực nội dung chính hiển thị:
 - + Ba ô lớn ở phần trên, là các gói nâng cấp chính với các tính năng chỉ khi nâng cấp mới có thể sử dụng được.
 - + Một thanh tiêu đề cho phần bên dưới.
 - + Sáu ô nhỏ hơn, có thể là tiện ích bổ sung.
- Footer ở cuối trang chứa thông tin bổ sung hoặc liên hệ.

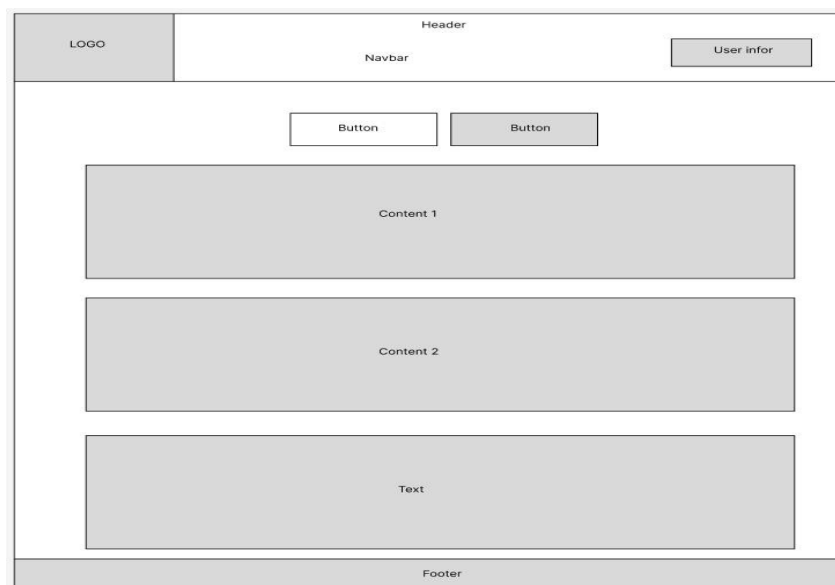


Hình 3.10: Layout trang nâng cấp gói

❖ Layout trang luyện thi

Trang luyện thi gồm:

- Header với logo bên trái, thanh điều hướng (Navbar) ở giữa và thông tin người dùng bên phải.
- Phần nội dung chính:
 - + Hai nút Button để chọn dạng kỳ thi (IELTS hoặc VSTEP).
 - + Content 1, Content 2, và Content 3 tương ứng với danh sách đề thi được phân loại (ví dụ: Free và VIP).
- Footer ở cuối trang chứa thông tin bổ sung.

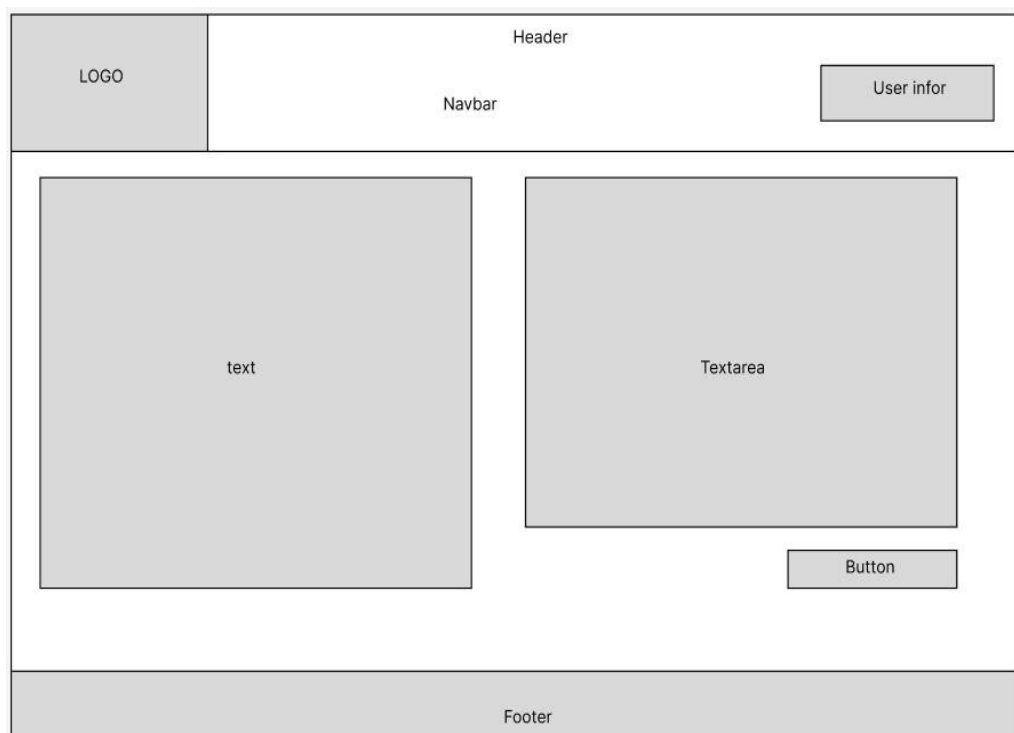


Hình 3.11: Layout trang luyện thi VSTEP hoặc IELTS

❖ Layout trang khi bắt đầu luyện thi

Trang bắt đầu luyện thi gồm:

- Header với logo bên trái, thanh điều hướng ở giữa và thông tin người dùng bên phải.
- Nội dung chính chia thành 2 phần:
 - + Bên trái: ô text hiển thị đề thi hoặc nội dung yêu cầu.
 - + Bên phải: Textarea để thí sinh nhập câu trả lời, kèm nút Button bên dưới để nộp bài.
- Footer ở cuối trang hiển thị thông tin bổ sung hoặc liên hệ.



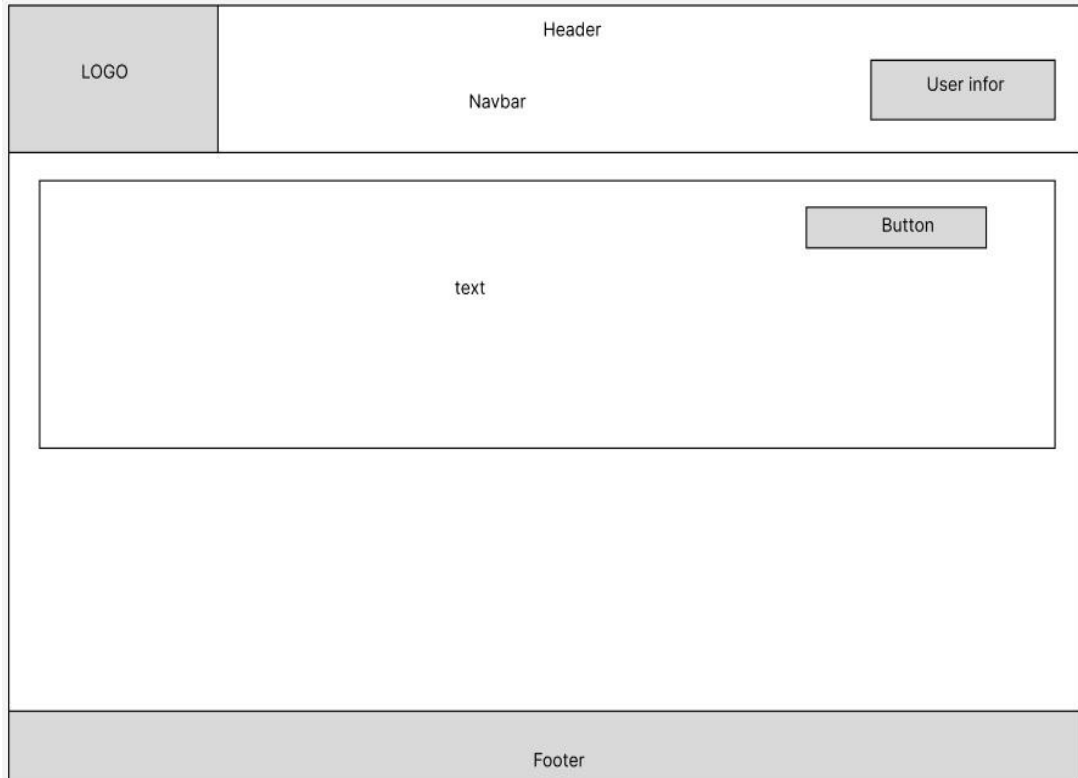
Hình 3.12: Layout trang bắt đầu luyện thi

❖ Layout trang xem các bài đã nộp chấm điểm

Trang xem lại các bài đã nộp chấm điểm gồm:

- Header với logo bên trái, thanh điều hướng (Navbar) ở giữa và thông tin người dùng bên phải.
- Phần nội dung chính hiển thị:
 - + Một khung text ở giữa chứa các bài thi đã nộp
 - + Nút Button ở góc phải khung để thực hiện thao tác xem chi tiết
 - + Một Button ở góc trái để hiển thị thông tin cá nhân

- Footer ở cuối trang hiển thị thông tin bổ sung.



Hình 3.13: Layout trang xem lịch sử các bài đã nộp

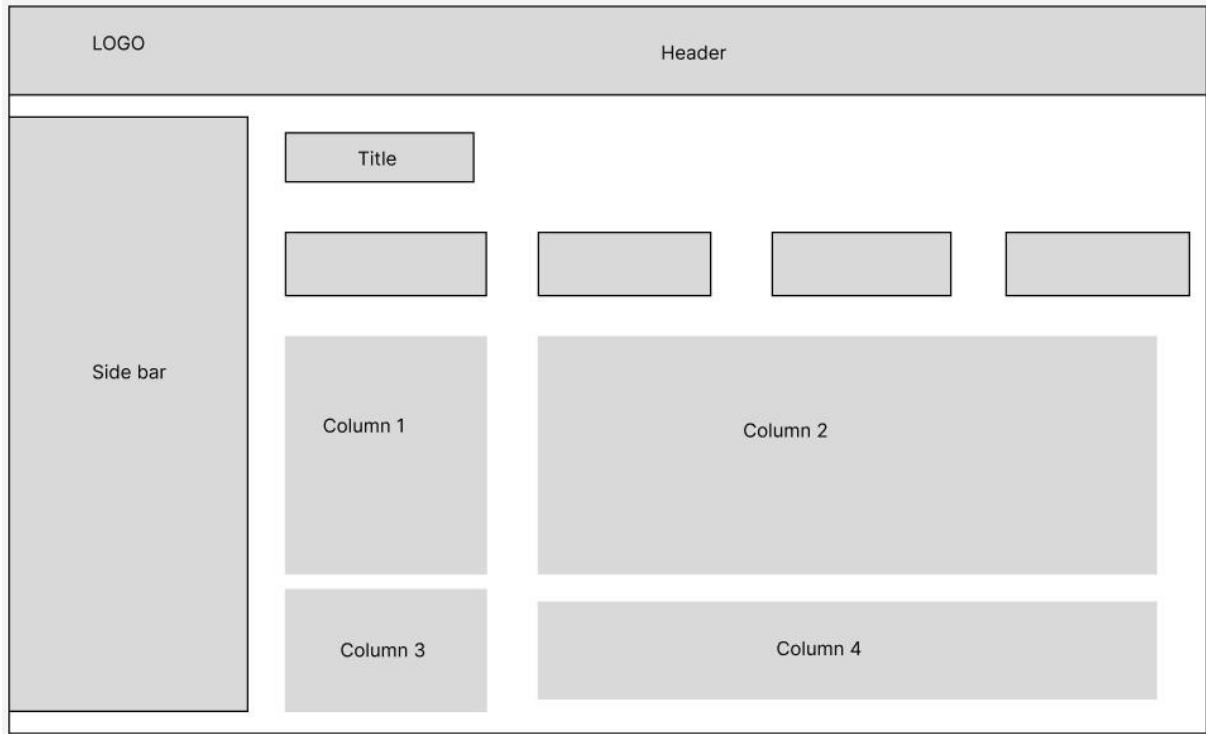
3.6.2. Layout giao diện quản trị

❖ Layout trang dashboard

Trang Dashboard quản trị bao gồm:

- Header trên cùng với logo bên trái và tiêu đề trang ở giữa.
- Sidebar bên trái chứa menu điều hướng để quản lý các mục (ví dụ: quản lý sinh viên, thống kê...).
- Phần nội dung chính gồm:
 - + Title hiển thị tên mục hiện tại.
 - + Hàng các ô nhỏ bên dưới title, có thể dùng làm nút chức năng hoặc hiển thị số liệu nhanh.
 - + Column 1: Thống kê số lượng các gói đã được đăng ký trên hệ thống
 - + Column 2: Thống kê các account vừa được tạo gần nhất
 - + Column 3: Thống kê số lượng account tạo trong các tháng
 - + Column 4: Hiển thị số lượng người dùng hoặc các thống kê liên quan.

Ngoài ra còn có các ô thống kê khác như: thống kê số tiền từ các gói người dùng đăng ký, thống kê các gói còn và chưa được đăng ký

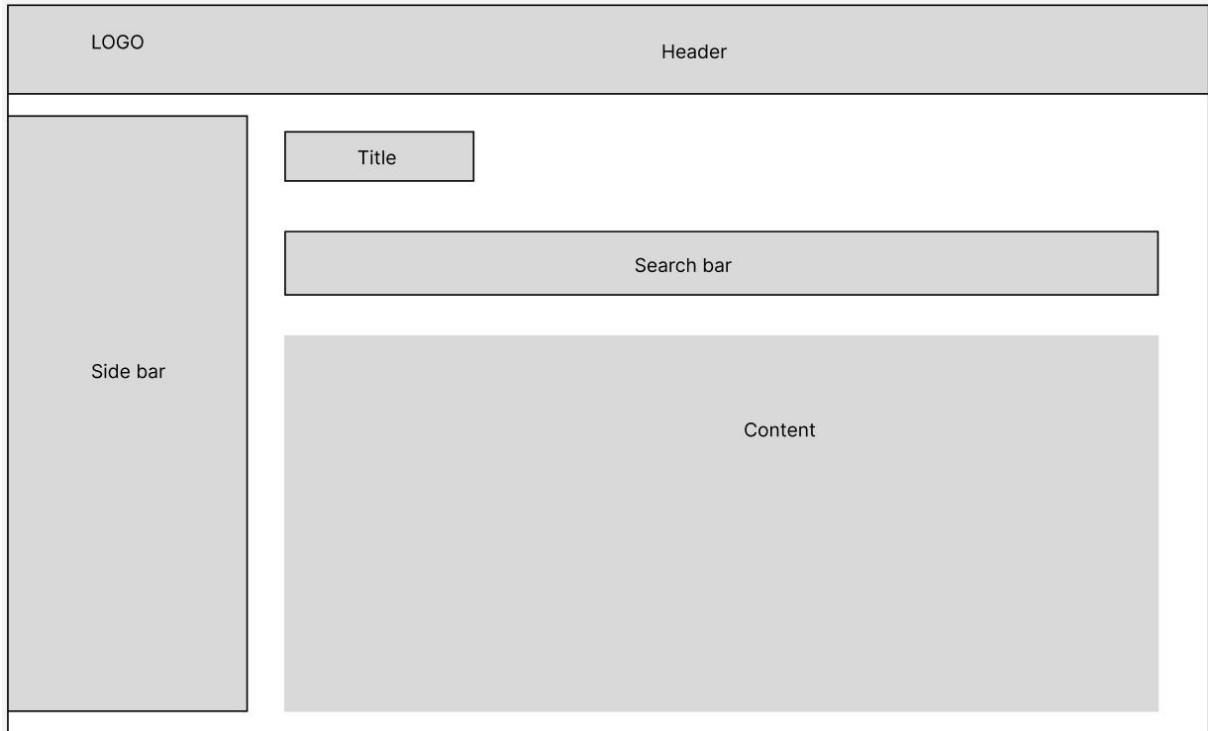


Hình 3.14: Layout trang Dashboard

❖ Layout trang quản lí người dùng

Trang quản lí người dùng bao gồm:

- Header phía trên cùng với logo bên trái và tiêu đề ở giữa.
- Sidebar bên trái chứa menu điều hướng đến các chức năng quản trị khác.
- Nội dung chính gồm:
 - + Title để hiển thị tiêu đề của phần đang quản lý.
 - + Search bar để tìm kiếm người dùng nhanh chóng
 - + Content bên dưới hiển thị danh sách người dùng
 - + Hiển thị thông tin người dùng
 - + Xem trạng thái người dùng
- Ngoài ra còn có các nút Button để thực hiện một số chức năng như:
 - + Chỉnh sửa thông tin cá nhân người dùng
 - + Xóa tài khoản người dùng
 - + Chỉnh sửa vai trò người dùng
 - + Một Button để thoát khỏi trang
 - + Một Button để xem trang profile
- Footer ở cuối trang hiển thị thông tin bổ sung.



Hình 3.15: Layout trang quản lý tài khoản người dùng

❖ Layout trang quản lý các gói nâng cấp của người dùng

Trang quản lý các gói nâng cấp của người dùng bao gồm:

- Header phía trên cùng:

- + Chứa logo đặt góc trái, giúp nhận diện thương hiệu hoặc hệ thống
- + Tiêu đề tổng thể: giúp người dùng nhận biết mình đang ở trang nào

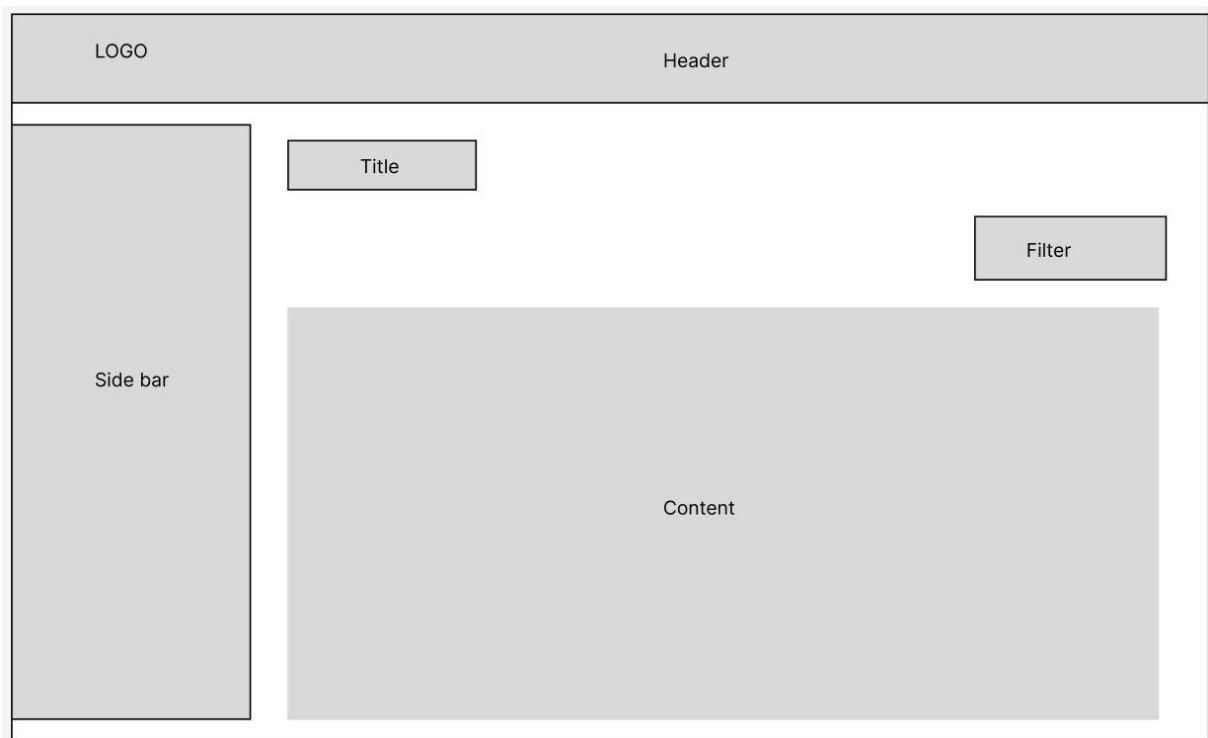
- Sidebar:

- + Vị trí bên trái màn hình
- + Chức năng điều hướng đến các module quản trị khác như: Dashboard, quản lý người dùng, quản lý gói nâng cấp, lịch sử giao dịch,...

- Khu vực nội dung chính gồm:

- + Title (tiêu đề) ở phía trên trái để hiển thị tên trang hoặc danh mục gói nâng cấp.
- + Filter (bộ lọc) ở phía trên phải để lọc dữ liệu theo tiêu chí (ví dụ: loại gói, trạng thái, thời gian nâng cấp...).
- + Content bên dưới hiển thị danh sách các gói nâng cấp, thông tin chi tiết, hoặc các tùy chọn quản lý.
- + Thanh tìm kiếm và phân trang

- + Nút xem trạng thái các gói
- + Nút Button xem chi tiết gói

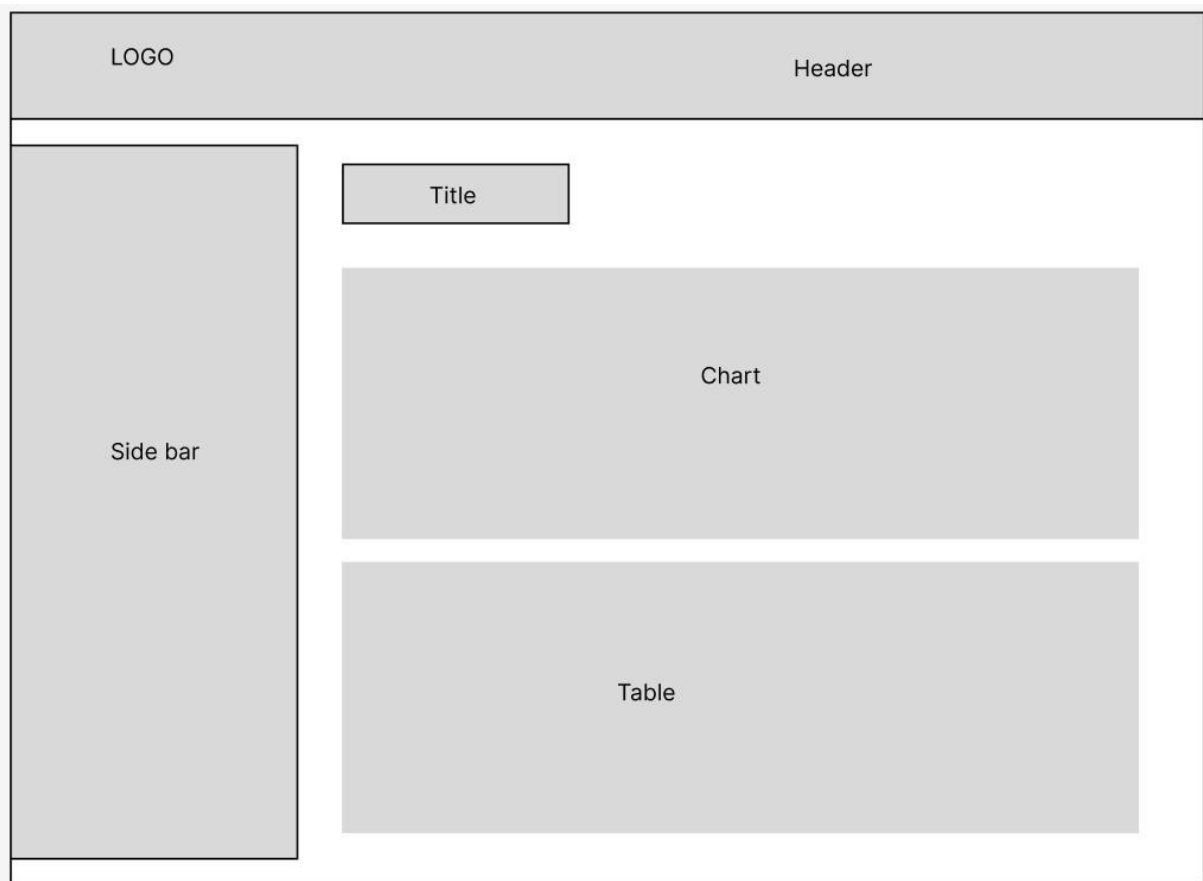


Hình 3.16: Layout trang quản lý các gói nâng cấp của người dùng

❖ Layout trang thống kê số lượng người dùng đăng ký tài khoản

Trang thống kê số lượng người dùng đăng ký tài khoản bao gồm:

- Header phía trên cùng: chứa logo và tiêu đề tổng thể.
- Sidebar:
 - + Vị trí bên trái
 - + Chức năng điều hướng đến các module quản trị khác như: Dashboard, quản lý người dùng, quản lý gói nâng cấp, lịch sử giao dịch,...
- Khu vực nội dung chính gồm:
 - + Title (tiêu đề): Nằm phía trên khu vực nội dung chính, hiển thị tên chức năng hiện tại
 - + Chart (biểu đồ): Nằm giữa, hiển thị biểu đồ thống kê
 - + Table (bảng): Nằm dưới biểu đồ, hiển thị dữ liệu chi tiết dạng bảng



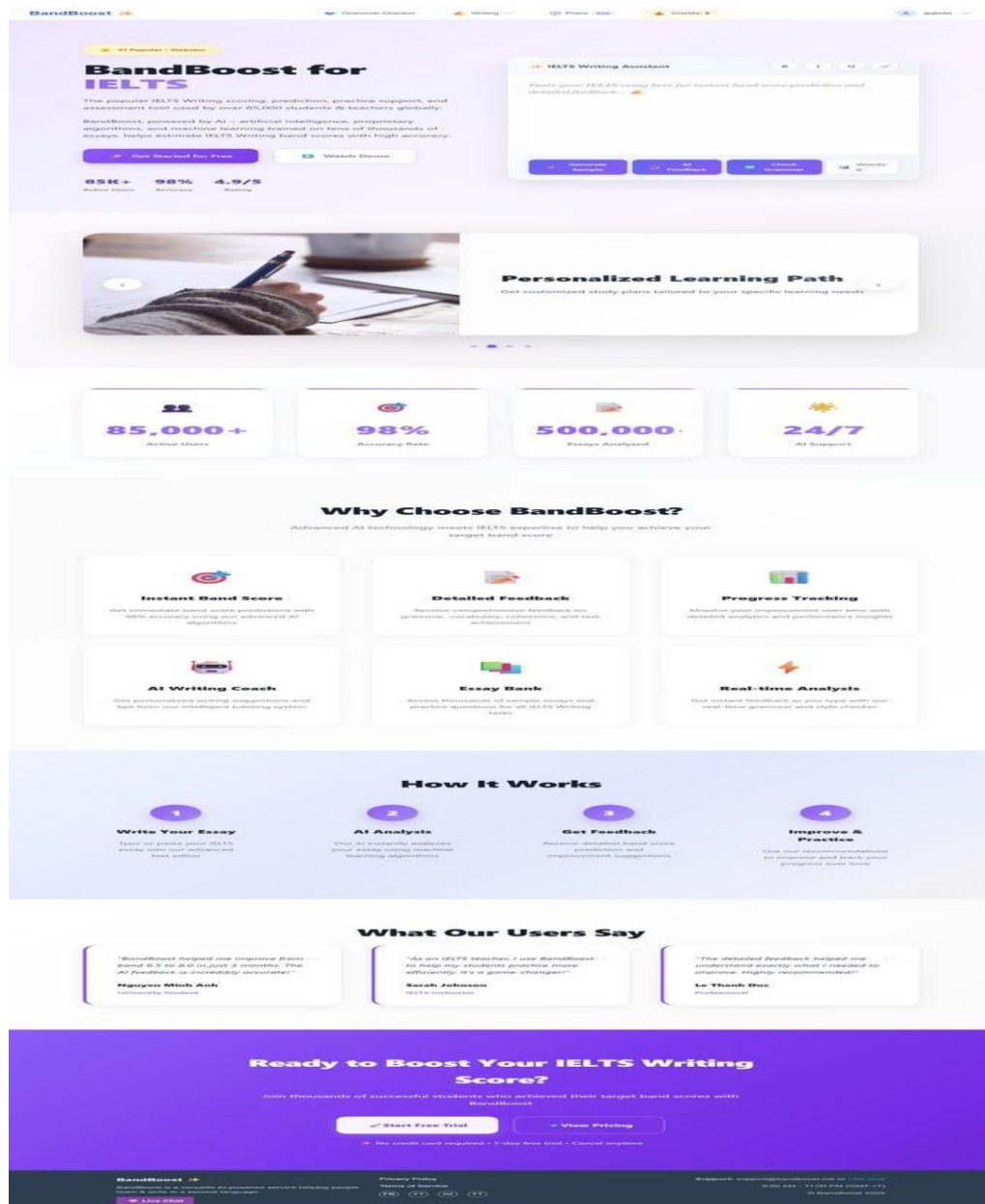
Hình 3.17: Layout trang thống kê số lượng đăng ký tài khoản người dùng

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

4.1. Giao diện người dùng

4.1.1. Giao diện trang chủ

Trang chủ BandBoost được thiết kế trực quan, hiển thị rõ thông điệp chính về công cụ chấm điểm và luyện viết IELTS bằng AI. Người dùng có thể dán bài viết vào khung nhập liệu, nhận phản hồi chi tiết và điểm số ước lượng theo thang IELTS. Giao diện đơn giản, dễ sử dụng, hỗ trợ đăng ký nhanh và cung cấp hỗ trợ trực tuyến, phù hợp cho học sinh luyện thi IELTS.



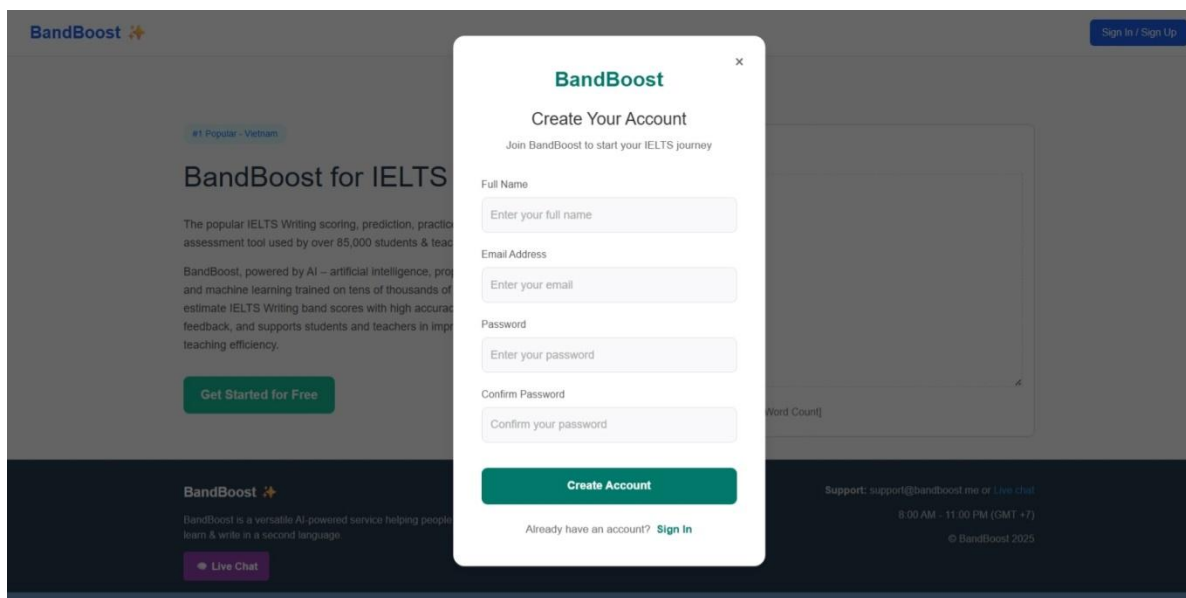
Hình 4.1: Giao diện trang chủ BandBoost

4.1.2. Giao diện trang đăng ký

Trang đăng ký tài khoản được thiết kế dưới dạng popup nổi bật ở trung tâm màn hình, giúp người dùng dễ dàng tập trung vào thao tác đăng ký mà không bị phân tán bởi nội dung nền. Ở phía trên cùng là tên thương hiệu “BandBoost” được hiển thị rõ ràng bằng màu xanh, bên dưới là tiêu đề “Create Your Account” cùng một câu mô tả ngắn gọn khuyến khích người dùng tham gia để bắt đầu hành trình học IELTS.

Bên trong popup có bốn trường nhập liệu cơ bản gồm họ và tên, địa chỉ email, mật khẩu và xác nhận mật khẩu. Các ô nhập liệu được thiết kế đơn giản, bo tròn nhẹ với placeholder gợi ý giúp người dùng dễ hiểu chức năng của từng trường. Ngay bên dưới là nút “Create Account” màu xanh lá nổi bật, đây là nút hành động chính để tạo tài khoản sau khi người dùng nhập đầy đủ thông tin.

Phía dưới nút tạo tài khoản có dòng chữ “Already have an account? Sign In” với phần “Sign In” được thiết kế dưới dạng liên kết, cho phép người dùng nhanh chóng chuyển sang màn hình đăng nhập nếu đã có tài khoản. Góc trên bên phải của popup có biểu tượng dấu “X” để đóng cửa sổ đăng ký và quay lại trang trước. Giao diện tổng thể đơn giản, rõ ràng, trực quan, giúp người dùng thao tác nhanh chóng và dễ dàng.

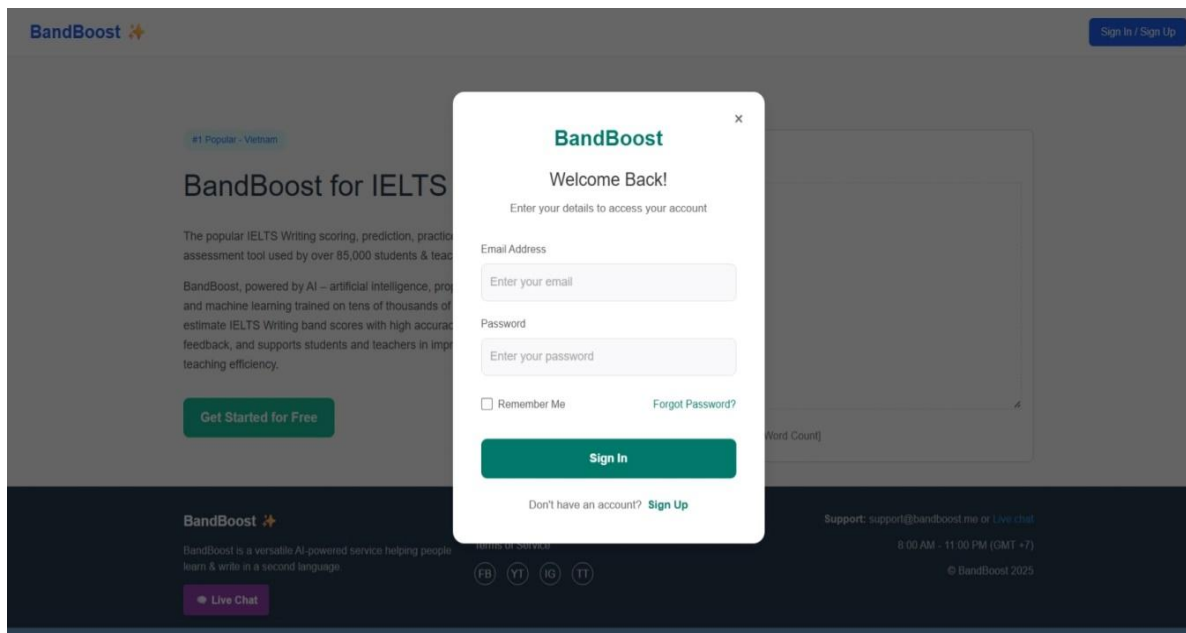


Hình 4.2: Giao diện trang đăng ký

4.1.3. Giao diện trang đăng nhập

Trang đăng nhập được thiết kế dưới dạng cửa sổ nổi (popup), yêu cầu người dùng nhập địa chỉ email và mật khẩu để truy cập hệ thống. Giao diện tích hợp tùy chọn

“Remember Me” và đường dẫn “Forgot Password?” để lấy lại mật khẩu khi cần. Bên dưới có liên kết chuyển nhanh sang trang đăng ký cho người dùng mới. Thiết kế tối giản, rõ ràng, giúp người dùng đăng nhập nhanh chóng và thuận tiện.



Hình 4.3: Giao diện trang đăng nhập

4.1.4. Giao diện trang lựa chọn dạng bài luyện viết

Trang Luyện Thi (Writing Practice) được thiết kế để giúp người học rèn luyện kỹ năng viết thông qua các bài tập có cấu trúc rõ ràng và bám sát đề thi thực tế. Người dùng có thể lựa chọn giữa các dạng bài luyện thi VSTEP và IELTS, với bốn nhóm chính:

- VSTEP Task 1: Email Writing – Luyện viết email trang trọng hoặc thân mật theo tình huống cho trước.
- VSTEP Task 2: Essay Writing – Luyện viết bài luận lập luận hoặc thảo luận về các chủ đề khác nhau.
- IELTS Task 1: Academic/General Training Writing – Luyện viết mô tả, phân tích biểu đồ, bảng, quy trình (Academic) hoặc viết thư (General Training) theo tình huống thực tế.
- IELTS Task 2: Essay Writing – Luyện viết bài luận nghị luận, nêu quan điểm, phân tích vấn đề xã hội, giáo dục, môi trường,...

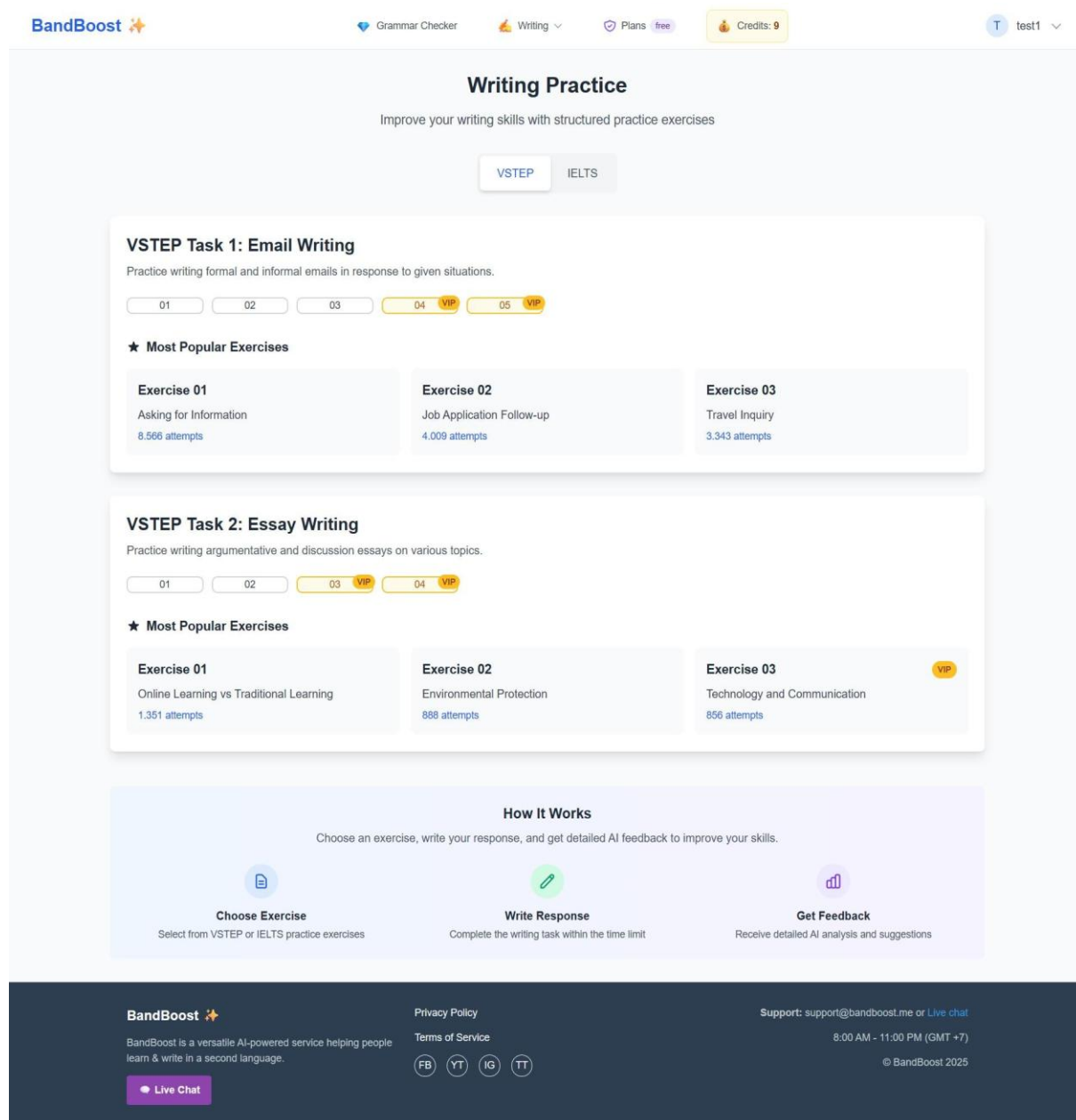
Mỗi nhóm bài tập được chia thành nhiều Exercise với mã số và tiêu đề riêng, kèm số lượt làm bài. Một số bài được gắn nhãn VIP, yêu cầu người dùng nâng cấp gói để truy cập.

Ngoài ra, trang cũng hiển thị Most Popular Exercises giúp người học dễ dàng tiếp cận các bài được nhiều người luyện tập nhất.

Phần How It Works ở cuối trang mô tả quy trình luyện thi gồm 3 bước:

- Bước 1: Choose Exercise – Chọn bài luyện từ danh sách VSTEP hoặc IELTS.
- Bước 2: Write Response – Thực hiện viết bài trong thời gian giới hạn.
- Bước 3: Get Feedback – Nhận phản hồi chi tiết và gợi ý cải thiện từ AI.

Giao diện được tối ưu trực quan, bố cục rõ ràng, giúp người học nhanh chóng tìm được bài luyện phù hợp và theo dõi tiến trình luyện tập.



Hình 4.4: Giao diện trang luyện viết VSTEP và IELTS

4.1.5. Giao diện trang làm bài luyện viết

Trang làm bài luyện thi khi người dùng chọn một đề luyện cụ thể từ danh sách, hệ thống sẽ chuyển sang giao diện làm bài. Trang được chia làm hai khu vực chính:

- Khu vực đề bài (bên trái):
 - + Hiển thị yêu cầu chi tiết của đề, ví dụ: viết email, bài luận hoặc mô tả biểu đồ.
 - + Cung cấp bối cảnh, nội dung hoặc tình huống (thường đặt trong khung nền xanh nhạt).
 - + Có hướng dẫn bổ sung về yêu cầu độ dài, định dạng và tiêu chí chấm điểm (Task Fulfillment, Organization, Vocabulary, Grammar).
 - + Thông tin về thời gian làm bài được hiển thị ở góc trên bên phải, giúp người học theo dõi tiến độ.
- Khu vực viết bài (bên phải):
 - + Ô nhập văn bản cho người dùng gõ câu trả lời trực tiếp.
 - + Bộ đếm số từ ở phía dưới giúp kiểm soát yêu cầu tối thiểu/tối đa.
 - + Nút Submit Response để nộp bài khi hoàn thành.

Giao diện được thiết kế đơn giản, tập trung vào nội dung bài làm, đồng thời vẫn giữ đồng hồ đếm ngược và tiêu chí chấm điểm để người học dễ dàng bám sát yêu cầu đề thi.

The screenshot displays the BandBoost writing practice interface. At the top, there's a navigation bar with 'BandBoost' logo, 'Grammar Checker', 'Writing', 'Plans free', and 'Credits: 9'. Below this, the main area is divided into two columns. The left column contains a task prompt: 'You should spend about 20 minutes on this task. Write an email to respond to Brianna.' followed by a sample email response from Brianna. The right column is titled 'Your Response:' and contains a large text input field with the placeholder 'Type your response here...'. Below the input field, it says 'Words: 0 / 150' and a 'Submit Response' button. The footer includes the BandBoost logo, a 'Live Chat' button, 'Privacy Policy', 'Terms of Service', and social media icons (FB, YT, IG, TT). It also shows support contact information and the copyright notice '© BandBoost 2025'.

Hình 4.5: Giao diện trang làm bài luyện thi

4.1.6. Giao diện trang chọn nhiệm vụ (Task) và nhập bài viết

Trang nộp bài viết cho phép người dùng lựa chọn giữa hai dạng đề chính của IELTS Writing là Task 1 (mô tả biểu đồ, bảng, sơ đồ) và Task 2 (viết bài luận phản hồi quan điểm, lập luận hoặc vấn đề). Người dùng có thể nhập bài trực tiếp vào ô soạn thảo lớn bên dưới, kèm gợi ý về số từ tối thiểu và bộ đếm từ giúp kiểm soát độ dài bài viết. Ở phía dưới có nút “Continue to Analysis” để gửi bài cho hệ thống AI phân tích và chấm điểm. Giao diện được bố trí gọn gàng, trực quan với phần chọn dạng bài nổi bật, phần soạn thảo rộng rãi, tạo cảm giác giống trải nghiệm làm bài thật, hỗ trợ luyện tập hiệu quả.

BandBoost Grammar Checker Writing Plans Credits 2 admin

Submit Writing Task

[View Previous Scores](#)

Select Task Type

Task 1
 Describe visual information (graph, table, chart, or diagram)
 Time limit: 20 minutes | Min words: 150

Task 2
 Write an essay in response to a point of view, argument, or problem
 Time limit: 40 minutes | Min words: 250

Your Essay

Enter your essay text

Start writing your essay here...

Word count: 0 Recommended: 150+ words

[Continue to Analysis](#)

BandBoost BandBoost is a versatile AI-powered service helping people learn & write in a second language. [Live Chat](#)

Privacy Policy Terms of Service (FB) (YT) (IG) (TT)

Support: support@bandboost.me or [Live chat](#) 8:00 AM - 11:00 PM (GMT +7) © BandBoost 2025

Hình 4.6 Giao diện trang viết và nộp bài

4.1.7. Trang lịch sử điểm số

Trang sửa lỗi hiển thị chi tiết lịch sử các bài viết đã chấm điểm, kèm theo điểm số từng tiêu chí (Task Achievement, Coherence, Lexical, Grammar). Bên dưới là bài viết gốc có đánh dấu lỗi, cùng phần sửa lỗi được gợi ý rõ ràng gồm ba nhóm: ngữ pháp, từ vựng và cấu trúc. Mỗi đề xuất đều có giải thích và ví dụ minh họa, giúp người học hiểu rõ điểm cần cải thiện. Trang cũng hỗ trợ xuất kết quả ra Word, thuận tiện cho việc lưu trữ và ôn tập.

BandBoost Grammar Checker Writing Plans Credits 0 test1

Writing Scores History

Track your IELTS writing progress over time

[+ New Writing Task](#)

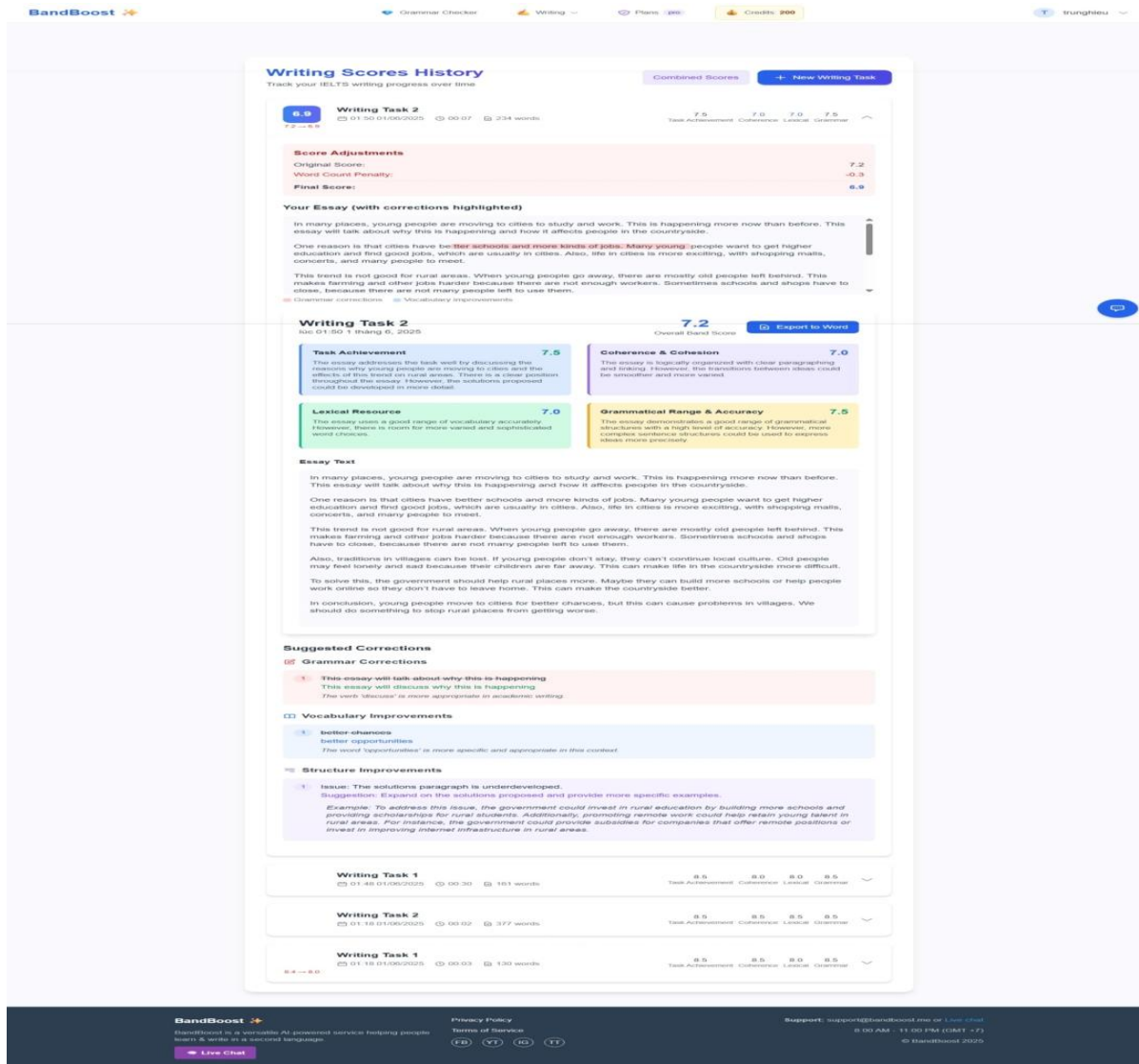
Task	Score	Date	Time	Words	Task Achievement	Coherence	Lexical	Grammar
Writing Task 2	4	20:49:00/20/2025	00:24	310 words	0	4	5	6

BandBoost BandBoost is a versatile AI-powered service helping people learn & write in a second language. [Live Chat](#)

Privacy Policy Terms of Service (FB) (YT) (IG) (TT)

Support: support@bandboost.me or [Live chat](#) 8:00 AM - 11:00 PM (GMT +7) © BandBoost 2025

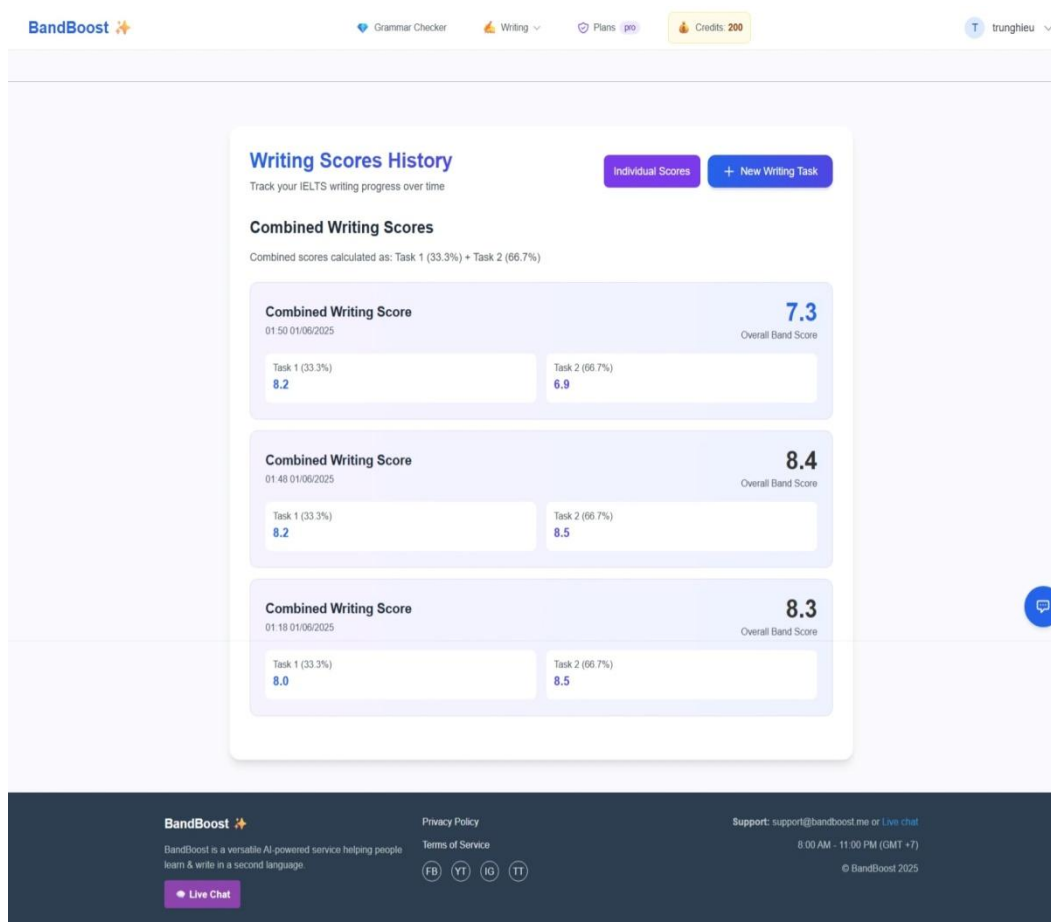
Hình 4.7: Giao diện trang lịch sử nộp bài



Hình 4.8: Giao diện trang sửa lỗi hiển thị chi tiết lịch sử các bài chấm

4.1.8. Giao diện trang điểm tổng hợp (Combined Score)

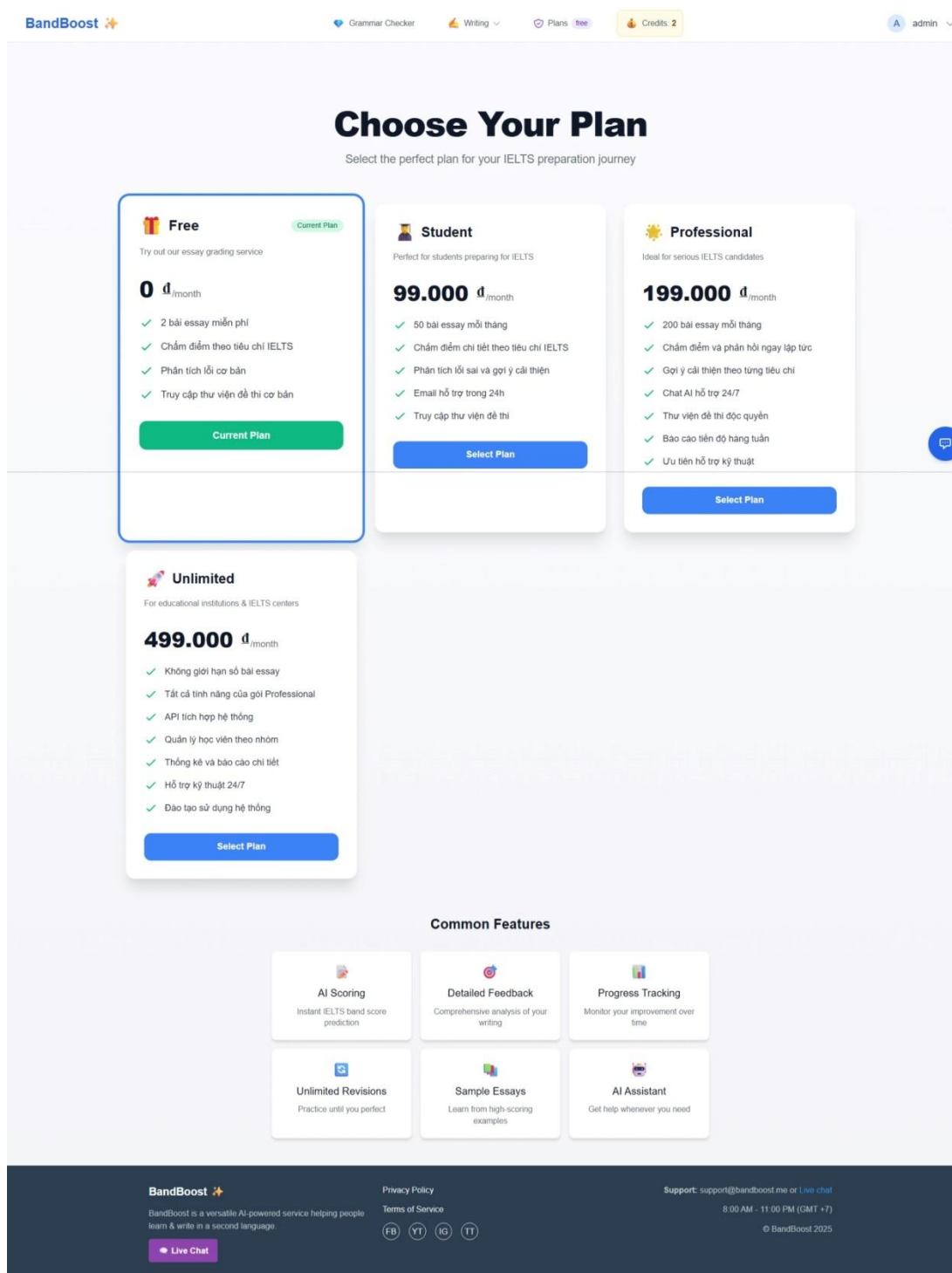
Trang cộng điểm Writing hiển thị điểm tổng hợp từ hai task của bài thi IELTS Writing, với Task 1 chiếm 33.3% và Task 2 chiếm 66.7% trong tổng điểm. Người dùng có thể theo dõi điểm từng phần và band điểm trung bình theo từng thời gian gửi bài. Giao diện rõ ràng, sắp xếp theo thứ tự thời gian, hỗ trợ theo dõi quá trình tiến bộ và so sánh hiệu suất giữa các lần luyện tập.



Hình 4.9: Giao diện trang cộng điểm Writing

4.1.9. Giao diện trang chọn gói nâng cấp

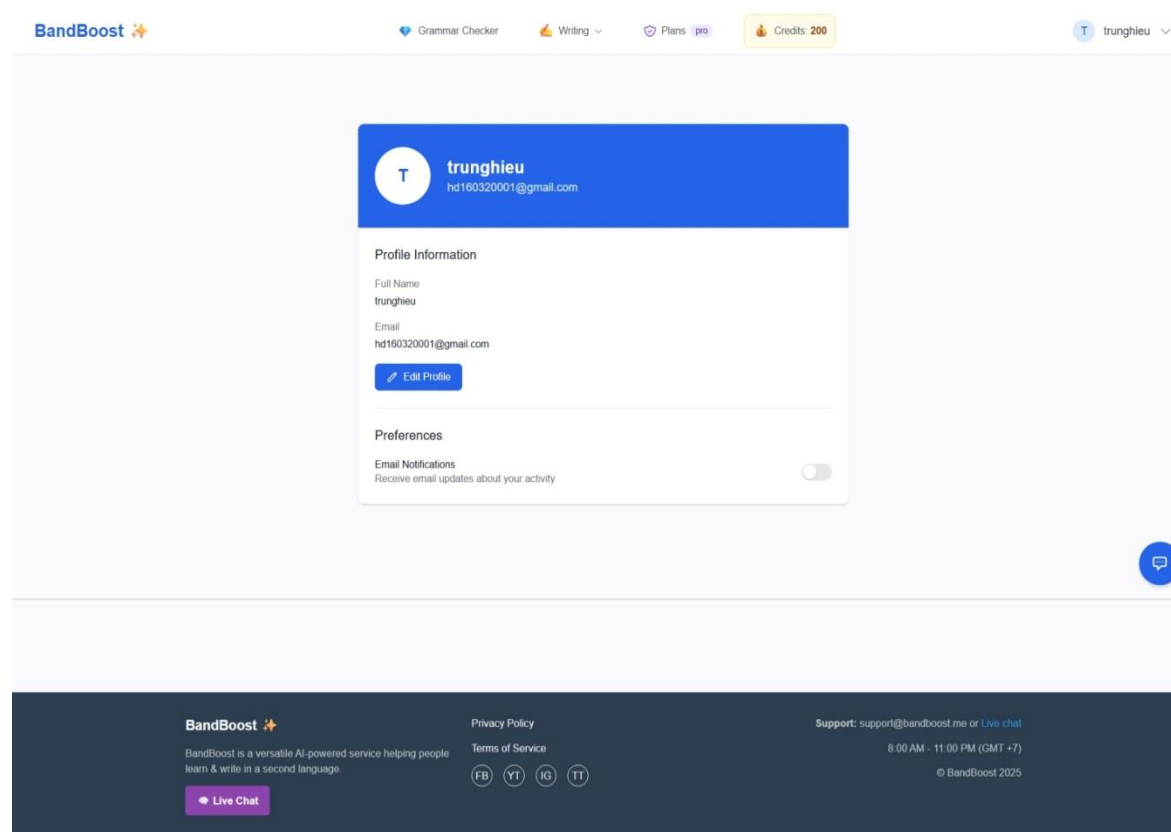
Trang chọn gói sử dụng hiển thị rõ ràng các gói dịch vụ theo tháng gồm: Free, Student, Professional và Unlimited, với mức giá và tính năng tăng dần. Mỗi gói được trình bày dưới dạng thẻ, dễ so sánh về số lượng bài chấm, mức độ hỗ trợ, và quyền truy cập tính năng. Bên dưới là các tiện ích chung như chấm điểm AI, phản hồi chi tiết, theo dõi tiến độ, và trợ lý AI. Phần so sánh tính năng chung bên dưới hỗ trợ người dùng hiểu rõ lợi ích mà tất cả các gói đều mang lại. Giao diện hiện đại, trực quan, giúp người dùng dễ dàng chọn gói phù hợp với nhu cầu luyện thi IELTS của mình.



Hình 4.10: Giao diện trang nâng cấp gói

4.1.10. Trang hồ sơ cá nhân

Trang hồ sơ người dùng (Profile) hiển thị thông tin tài khoản bao gồm tên đầy đủ, địa chỉ email và tùy chọn chỉnh sửa hồ sơ. Người dùng có thể cập nhật thông tin cá nhân bằng nút "Edit Profile" và bật/tắt nhận thông báo email. Giao diện tối giản, rõ ràng, tập trung vào tính cá nhân hóa và dễ quản lý thông tin tài khoản.



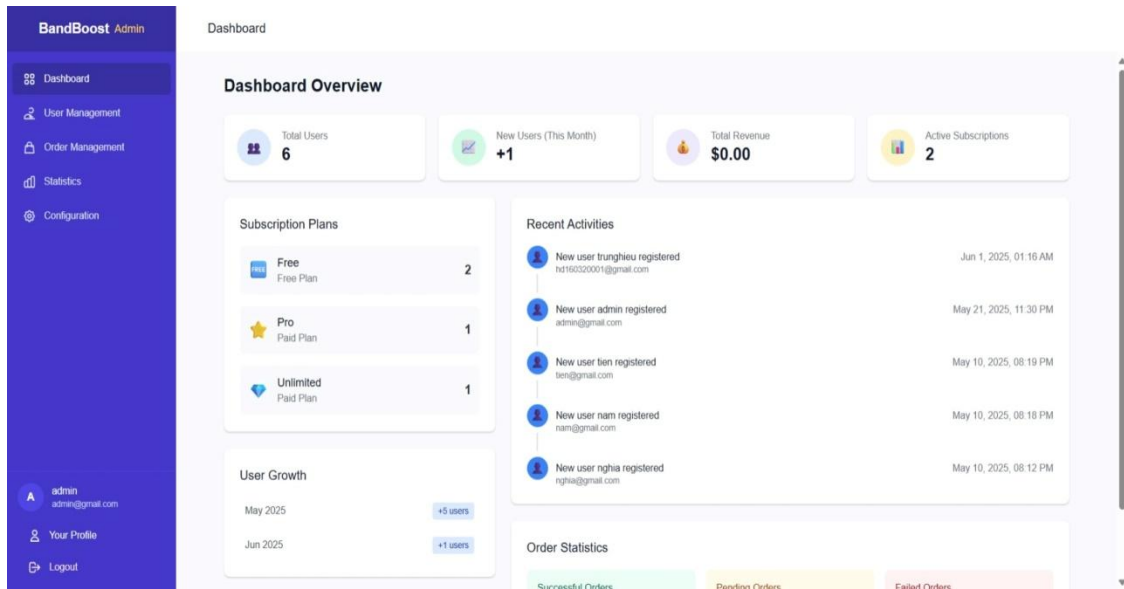
Hình 4.11: Giao diện trang thông tin cá nhân

4.2. Giao diện trang quản trị

4.2.1. Dashboard chính

Trang Dashboard của admin cung cấp cái nhìn tổng quan về hệ thống, bao gồm tổng số người dùng, số người đăng ký mới trong tháng, doanh thu và số lượng gói đang hoạt động. Giao diện còn hiển thị chi tiết số lượng người dùng theo từng gói (Free, Pro, Unlimited), các hoạt động gần đây như đăng ký tài khoản mới, cùng thống kê tăng trưởng người dùng theo từng tháng. Thanh điều hướng bên trái giúp quản trị viên truy cập nhanh vào các mục quản lý người dùng, đơn hàng, thống kê và cấu hình hệ thống. Giao diện tối ưu cho việc giám sát và điều hành hệ thống một cách trực quan và hiệu quả. Ngoài ra, phần thống kê đơn hàng (thành công, chờ xử lý, thất bại) được bố trí khoa học, giúp quản trị viên dễ dàng theo dõi tình trạng giao dịch. Các thông tin được trình bày gọn gàng với bảng màu trực quan, hỗ trợ việc quản lý nhanh chóng và chính xác.

Hệ thống còn hỗ trợ hiển thị chi tiết các chỉ số theo thời gian thực, giúp quản trị viên kịp thời đưa ra quyết định. Thiết kế hiện đại, bố cục rõ ràng giúp tăng trải nghiệm sử dụng và hiệu quả quản lý.



Hình 4.12: Giao diện trang dashboard

4.2.2. Trang quản lý người dùng

Trang quản lý người dùng hiển thị danh sách tài khoản đã đăng ký, gồm các thông tin: tên người dùng, email, ngày đăng ký, vai trò (User/Admin) và trạng thái hoạt động (Active/Inactive). Quản trị viên có thể tìm kiếm người dùng theo tên hoặc email, chỉnh sửa vai trò, đổi trạng thái hoặc xóa tài khoản trực tiếp. Giao diện rõ ràng, dễ thao tác, hỗ trợ kiểm soát hệ thống người dùng một cách hiệu quả và nhanh chóng.

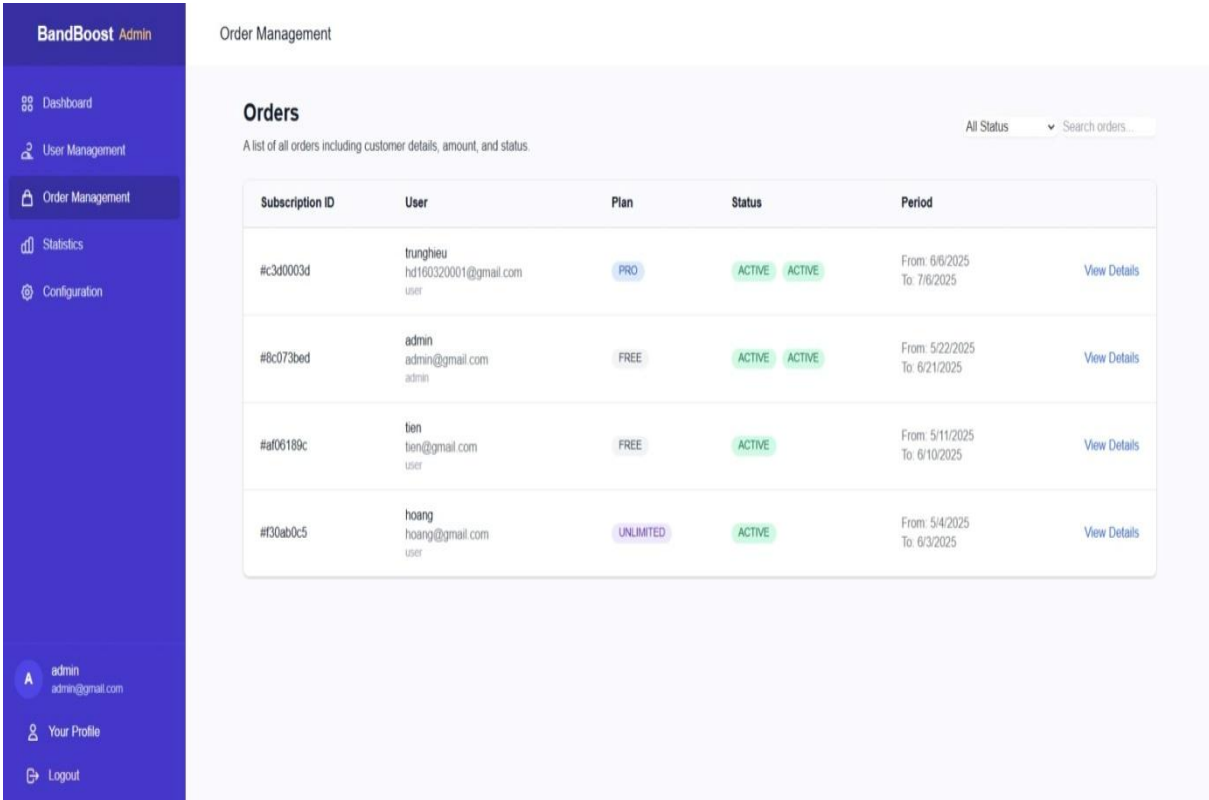
The screenshot shows the BandBoost Admin User Management page. The left sidebar contains navigation links: Dashboard, User Management, Order Management, Statistics, and Configuration. The main content area is titled 'User Management' and includes a search bar and a table of users.

User	Email	Role	Status	
T trungieu Joined 6/1/2025	hd160320001@gmail.com	User	Inactive	Edit Delete
T tien Joined 5/10/2025	tien@gmail.com	User	Inactive	Edit Delete
H hoang Joined 5/2/2025	hoang@gmail.com	User	Inactive	Edit Delete
A admin Joined 5/21/2025	admin@gmail.com	Admin	Inactive	Edit Delete
N nam Joined 5/10/2025	nam@gmail.com	User	Inactive	Edit Delete
N nghia Joined 5/10/2025	nghia@gmail.com	User	Inactive	Edit Delete

Hình 4.13: Giao diện trang quản lý người dùng

4.2.3. Trang quản lý đơn hàng

Trang quản lý đơn hàng hiển thị đầy đủ thông tin về các đăng ký dịch vụ của người dùng, bao gồm mã đơn hàng (Subscription ID), tên và email người dùng, loại gói đăng ký (Free, Pro, Unlimited), trạng thái hoạt động và thời hạn sử dụng. Các trạng thái được thể hiện bằng nhãn màu sắc nổi bật giúp phân biệt nhanh giữa các gói và tình trạng dịch vụ. Giao diện cho phép quản trị viên dễ dàng tìm kiếm đơn hàng bằng tên hoặc mã đơn, cũng như lọc theo trạng thái để quản lý hiệu quả hơn. Mỗi đơn hàng đều có thời gian bắt đầu và kết thúc rõ ràng, giúp việc theo dõi chu kỳ gia hạn hoặc hết hạn trở nên trực quan. Nút “View Details” ở cuối mỗi dòng cho phép xem chi tiết hoặc thực hiện thao tác quản trị chuyên sâu với đơn hàng. Bảng danh sách được thiết kế gọn gàng, bố cục hợp lý giúp tối ưu trải nghiệm ngay cả khi xử lý nhiều dữ liệu. Trang này đóng vai trò quan trọng trong việc đảm bảo tính minh bạch và kiểm soát toàn bộ quá trình đăng ký dịch vụ trong hệ thống.



Subscription ID	User	Plan	Status	Period
#c3d0003d	trungheio hd160320001@gmail.com user	PRO	ACTIVE ACTIVE	From: 6/6/2025 To: 7/6/2025 View Details
#8c073bed	admin admin@gmail.com admin	FREE	ACTIVE ACTIVE	From: 5/22/2025 To: 6/21/2025 View Details
#af06189c	tien tien@gmail.com user	FREE	ACTIVE	From: 5/11/2025 To: 6/10/2025 View Details
#f30ab0c5	hoang hoang@gmail.com user	UNLIMITED	ACTIVE	From: 5/4/2025 To: 6/3/2025 View Details

Hình 4.14: Giao diện trang quản lý các gói người dùng đăng ký

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

5.1.1. Kết quả và đóng góp của đồ án

Sau quá trình thiết kế, xây dựng và triển khai, hệ thống tự động chấm điểm và hỗ trợ sửa lỗi tiếng Anh kỹ năng Writing đã hoàn thiện đầy đủ các chức năng cốt lõi, đáp ứng được yêu cầu của một nền tảng hỗ trợ học tiếng Anh hiệu quả và thông minh. Hệ thống cho phép người dùng dễ dàng tạo tài khoản, đăng nhập và sử dụng các tính năng như: nhập bài viết, nhận phản hồi về điểm số, phát hiện lỗi ngữ pháp, từ vựng, cấu trúc câu, cũng như nhận các gợi ý cải thiện trực tiếp từ mô hình AI.

Phần giao diện người dùng được xây dựng bằng ReactJS, được thiết kế theo hướng tối giản và trực quan, giúp người học dễ dàng thao tác như nhập bài, xem phản hồi AI, và lưu lại lịch sử các lần luyện tập trước đó. Tính năng hỗ trợ hiển thị điểm số tổng quan và chi tiết theo các tiêu chí (ngữ pháp, từ vựng, logic, cấu trúc) giúp người học có cái nhìn toàn diện và có định hướng cải thiện rõ ràng.

Về phía kỹ thuật, hệ thống backend được xây dựng trên nền tảng Flask (Python) với kiến trúc RESTful API. Mô hình ngôn ngữ ChatGPT-3.5 Turbo được tích hợp trực tiếp thông qua OpenAI API nhằm đảm nhiệm vai trò đánh giá nội dung và đề xuất cải thiện một cách tự nhiên, chính xác và có tính cá nhân hóa cao. Các yêu cầu xử lý từ frontend được gửi đến server Flask, sau đó được tiền xử lý và chuyển đến API ChatGPT, rồi trả kết quả về cho người dùng theo thời gian thực.

Một điểm nổi bật trong hệ thống là khả năng tự động gợi ý sửa lỗi và nâng cao câu bằng cách sử dụng prompt chuyên biệt cho ChatGPT, mô phỏng vai trò của một giám khảo IELTS có kinh nghiệm. Hệ thống không chỉ chỉ ra lỗi sai, mà còn giải thích nguyên nhân và đưa ra cách viết lại phù hợp theo tiêu chuẩn học thuật.

Về mặt bảo mật, hệ thống sử dụng cơ chế JWT Authentication với hai cấp quyền chính: người dùng và quản trị viên. Quản trị viên có thể theo dõi lịch sử hoạt động của người dùng, kiểm tra dữ liệu thống kê về mức độ tiến bộ, cũng như can thiệp chỉnh sửa nội dung nếu cần thiết.

5.1.2. Hạn chế

Mặc dù hệ thống đã triển khai và hoàn thiện phần lớn các chức năng cốt lõi phục vụ cho việc chấm điểm và hỗ trợ sửa lỗi tiếng Anh kỹ năng Writing, vẫn còn một số hạn chế cần được cải thiện trong các giai đoạn phát triển tiếp theo.

Trước hết, dù hệ thống đã phân tích bài viết dựa trên các tiêu chí chấm điểm chuẩn IELTS bao gồm Task Achievement, Coherence & Cohesion, Lexical Resource và Grammatical Range & Accuracy, các phân đánh giá hiện vẫn mang tính mô tả tổng quan. Việc phân tích lỗi và gợi ý cải thiện chi tiết tương ứng với từng tiêu chí cụ thể chưa được tự động hóa sâu, chẳng hạn như phân loại lỗi ngữ pháp theo dạng (thì, cấu trúc câu, mạo từ...) hay từ vựng theo cấp độ học thuật. Điều này khiến người học chưa thể hiểu rõ điểm yếu chính nằm ở kỹ năng nào để tập trung cải thiện hiệu quả.

Bên cạnh đó, hệ thống hiện tại chưa có chức năng so sánh trực quan giữa bài viết trước và sau khi sửa một tính năng quan trọng giúp người học nhận thấy mức độ tiến bộ của bản thân theo thời gian. Thay đổi trước/sau chỉ được thể hiện ở dạng chỉnh sửa trực tiếp trong văn bản, chưa có biểu đồ hoặc chế độ so sánh song song.

Ngoài ra, phần hiển thị phản hồi từ AI, mặc dù đã có các khối tách biệt cho sửa lỗi, gợi ý từ vựng và cải thiện cấu trúc, vẫn còn đơn giản về mặt giao diện. Việc tách rõ lỗi nghiêm trọng, lỗi nhẹ và gợi ý nâng cao sẽ giúp người học dễ dàng ưu tiên xử lý và học tập hiệu quả hơn.

5.2. Hướng phát triển

Trong giai đoạn tiếp theo, hệ thống sẽ tập trung phát triển theo hướng cá nhân hóa trải nghiệm học tập và nâng cao tính thông minh của AI, nhằm đáp ứng tốt hơn nhu cầu luyện kỹ năng Writing của người dùng ở nhiều cấp độ khác nhau.

Trước tiên, hệ thống sẽ được mở rộng khả năng phân tích lỗi theo từng tiêu chí IELTS Writing Band Descriptor, không chỉ dừng ở điểm số và mô tả chung, mà sẽ tự động nhận diện các lỗi ngữ pháp cụ thể (ví dụ: lỗi chia thì, lỗi mạo từ, lỗi chủ-vị, câu phức...), các lỗi về tổ chức ý (coherence), cũng như cách sử dụng từ vựng chưa chính xác hoặc chưa học thuật. Nhờ đó, người học có thể nắm rõ điểm yếu của mình theo từng tiêu chí và tập trung cải thiện đúng trọng tâm.

Bên cạnh đó, giao diện phản hồi từ AI sẽ được nâng cấp theo hướng phân loại gợi ý theo mức độ ưu tiên, bao gồm: lỗi nghiêm trọng cần sửa gấp, lỗi cải thiện chất

lượng bài viết, và gợi ý nâng cao để đạt band cao hơn. Các gợi ý cũng sẽ được liên kết với phân lý thuyết ngữ pháp/từ vựng tương ứng, giúp người học hiểu rõ nguyên nhân và cách khắc phục.

Về mặt công nghệ, hệ thống sẽ tiếp tục tối ưu tốc độ xử lý và giảm độ trễ phản hồi thông qua caching hợp lý và tái sử dụng phiên làm việc với AI khi cần thiết.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] IDP IELTS, "IELTS Writing band descriptors," IELTS IDP Vietnam, [Online]. Available: <https://ielts.idp.com/vietnam/prepare/article-ielts-writing-band-descriptors>. [Accessed: 15-Apr-2025].
- [2] Microsoft, "TypeScript: JavaScript With Syntax for Types," TypeScript Handbook, [Online]. Available: <https://www.typescriptlang.org/>. [Accessed: 15-Apr-2025].
- [3] Microsoft Learn, "Model-View-Controller (MVC) architecture," [Online]. Available: <https://learn.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview>. [Accessed: 15-Apr-2025].
- [4] Mozilla Developer Network, "CSS: Cascading Style Sheets," MDN Web Docs, [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/CSS>. [Accessed: 15-Apr-2025].
- [5] Mozilla Developer Network, "HTML: HyperText Markup Language," MDN Web Docs, [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTML>. [Accessed: 15-Apr-2025].
- [6] Mozilla Developer Network, "JavaScript," MDN Web Docs, [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [Accessed: 15-Apr-2025].
- [7] Mozilla Developer Network, "JSX – JavaScript XML," MDN Web Docs, [Online]. Available: <https://developer.mozilla.org/en-US/docs/Glossary/JSX>. [Accessed: 15-Apr-2025].
- [8] OpenAI, "OpenAI API Reference," [Online]. Available: <https://platform.openai.com/docs/>. [Accessed: 15-Apr-2025].
- [9] OpenAI, "OpenAI Python Library," [Online]. Available: <https://github.com/openai/openai-python>. [Accessed: 15-Apr-2025].
- [10] Oracle Corporation, "MySQL Documentation," [Online]. Available: <https://dev.mysql.com/doc/>. [Accessed: 15-Apr-2025].

- [11] Python Software Foundation, "The Python Tutorial," [Online]. Available: <https://docs.python.org/3/tutorial/>. [Accessed: 15-Apr-2025].
- [12] ReactJS, "React – A JavaScript library for building user interfaces," React Official Documentation, [Online]. Available: <https://reactjs.org/>. [Accessed: 15-Apr-2025].
- [13] Real Python, "Flask by Example," [Online]. Available: <https://realpython.com/tutorials/flask/>. [Accessed: 15-Apr-2025].
- [14] Simplilearn, "What is ReactJS? Introduction to ReactJS and Its Features," [Online]. Available: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>. [Accessed: 15-Apr-2025].
- [15] V. Hồ, "Tìm hiểu MySQL Architecture," Viblo, Aug. 13, 2020. [Online]. Available: <https://viblo.asia/p/tim-hieu-mysql-architecture-RnB5pj9JZPG>. [Accessed: 15-Apr-2025].
- [16] W3Schools, "CSS Tutorial," [Online]. Available: <https://www.w3schools.com/css/>. [Accessed: 15-Apr-2025].
- [17] W3Schools, "HTML Tutorial," [Online]. Available: <https://www.w3schools.com/html/>. [Accessed: 15-Apr-2025].
- [18] W3Schools, "JavaScript Tutorial," [Online]. Available: <https://www.w3schools.com/js/>. [Accessed: 15-Apr-2025].
- [19] W3Schools, "MySQL Tutorial," [Online]. Available: <https://www.w3schools.com/mysql/>. [Accessed: 15-Apr-2025].
- [20] W3Schools, "Python Tutorial," [Online]. Available: <https://www.w3schools.com/python/>. [Accessed: 15-Apr-2025].
- [21] W3Schools, "TypeScript Tutorial," [Online]. Available: <https://www.w3schools.com/typescript/>. [Accessed: 15-Apr-2025].

[22] Flask Documentation, “Welcome to Flask,” [Online]. Available: <https://flask.palletsprojects.com/>. [Accessed: 15-Apr-2025].

[23] MC IELTS, “CÁCH TÍNH ĐIỂM IELTS WRITING TASK 1,” MCIELTS, 22-Dec-2023. [Trực tuyến]. Available: <https://mcielts.com/cach-tinh-diem-ielts-writing-task-1/>. [Accessed: 15-Apr-2025].