

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN
THÔNG**



NGUYỄN CHÍNH MINH

**NGHIÊN CỨU PHÁT TRIỂN ỨNG DỤNG HỖ
TRỢ HỌC NGOẠI NGỮ DỰA TRÊN MÔ HÌNH
NGÔN NGỮ LỚN**

CHUYÊN NGÀNH : KHOA HỌC MÁY TÍNH

MÃ SỐ: 8.48.01.01

**TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ
THUẬT**

HÀ NỘI - 2025

MỞ ĐẦU

1. Lý do chọn đề tài

Sự bùng nổ của trí tuệ nhân tạo, đặc biệt là các mô hình ngôn ngữ lớn, mang đến tiềm năng cải thiện chất lượng học tập tiếng Anh thông qua phản hồi tự động và cá nhân hóa. Người học thường thiếu môi trường tương tác và hướng dẫn kịp thời, trong khi LLM có khả năng hiểu ngữ cảnh và tạo ra phản hồi phù hợp dựa trên tài liệu chuẩn. Do đó, việc phát triển một ứng dụng tích hợp LLM nhằm hỗ trợ người học tự học tiếng Anh là cần thiết, giúp tối ưu hóa trải nghiệm và hiệu quả học tập.

2. Tổng quan về vấn đề nghiên cứu

Các nghiên cứu trước đây cho thấy AI có thể cá nhân hóa trải nghiệm học tập, nhưng việc tích hợp LLM vào ứng dụng di động gặp nhiều thách thức liên quan đến xử lý ngữ cảnh phức tạp và đảm bảo tính mượt mà. In-Context Learning (ICL) và Retrieval-Augmented Generation (RAG) được xem là hai kỹ thuật then chốt để cải thiện khả năng tương tác và độ chính xác của hệ thống mà không tốn nhiều tài nguyên huấn luyện. Nghiên cứu sẽ tập trung vào

cách kết hợp ICL và RAG với LLM để cung cấp thông tin chính xác, kịp thời dựa trên bộ tài liệu tiếng Anh chuẩn.

3. Mục đích nghiên cứu

Về lý luận, đề tài nhằm xây dựng cơ sở kiến thức về In-Context Learning và RAG, từ đó chỉ ra cách làm giàu prompt và truy xuất thông tin cho LLM trong bối cảnh giáo dục ngoại ngữ. Về thực tiễn, mục tiêu là phát triển một ứng dụng di động cho phép người dùng tương tác với chatbot AI qua văn bản và giọng nói, đảm bảo mọi phản hồi đều dựa trên tài liệu chuẩn hóa, giúp người học xây dựng và củng cố kiến thức một cách liên tục và tự động.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu là ứng dụng hỗ trợ học ngoại ngữ tích hợp LLM, tập trung vào việc sử dụng In-Context Learning để tối ưu tương tác giữa người dùng và chatbot. Phạm vi bao gồm thiết kế, phát triển và thử nghiệm trên nền tảng di động, sử dụng ngôn ngữ Java/Kotlin cho Android hoặc React Native, cùng với việc kết nối GPT-4 qua LangChain để đảm bảo chất lượng câu trả lời dựa trên tài liệu tiếng Anh chuẩn.

5. Phương pháp nghiên cứu

Phương pháp nghiên cứu kết hợp phân tích lý thuyết và thực nghiệm: đầu tiên tổng hợp, phân tích các công trình liên quan đến ICL, RAG và ứng dụng LLM trong giáo dục; sau đó áp dụng phương pháp Agile để thiết kế, phát triển, kiểm thử và cải tiến ứng dụng. Các thử nghiệm chức năng, phi chức năng và thu thập phản hồi người dùng sẽ được thực hiện để đánh giá hiệu quả. Công cụ chính bao gồm React Native/Android, LangChain và GPT-4.

CHƯƠNG 1 : CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN

1.1 Tổng quan về học tập ngoại ngữ

1.1.1 Khái niệm về học tập ngoại ngữ

Học ngoại ngữ được xem là quá trình tương tác đa chiều giữa các yếu tố nhận thức, cảm xúc và xã hội, không chỉ là việc thuần túy nắm bắt ngữ pháp hay từ vựng mà còn bao gồm sự thấu hiểu bối cảnh văn hóa. Nhiều khung lý thuyết – từ tiếp cận hành vi đến quan điểm nhận thức hiện đại – đã đưa ra những cách giải thích khác nhau về cơ chế tiếp thu ngôn ngữ, trong đó Ngữ pháp Phổ quát khẳng định khả năng bẩm sinh của con người để học ngôn ngữ.

1.1.2 Lợi ích của việc học ngoại ngữ dựa trên các bộ tài liệu chuẩn hóa

Việc học theo tài liệu chuẩn hóa đem lại cho người học một lộ trình chặt chẽ, đã được kiểm chứng về mặt nội dung và phương pháp. Những tài liệu này tích hợp đầy đủ ngữ pháp, từ vựng và yếu tố văn hóa, giúp học viên xây dựng kiến thức một cách liên tục và mạch lạc. Đồng thời, tài liệu chuẩn hóa thường thiết kế theo khung quốc tế như CEFR,

giúp đánh giá trình độ một cách chính xác và chuẩn bị cho các mục tiêu học tập hoặc nghề nghiệp. Qua các bài tập và ví dụ thực tiễn, người học không chỉ củng cố kiến thức mà còn rèn luyện kỹ năng giao tiếp tự nhiên trên cả bốn lĩnh vực: nghe, nói, đọc, viết.

1.2 Mô hình ngôn ngữ lớn

1.2.1 Giới thiệu về các mô hình ngôn ngữ lớn

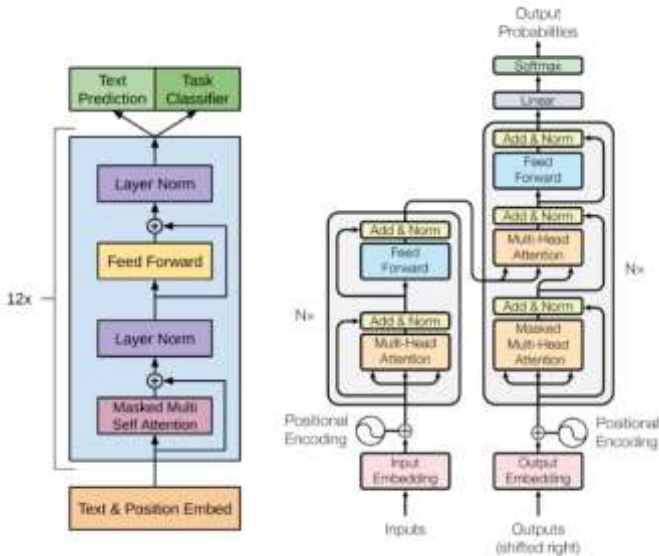
Transformer là kiến trúc cốt lõi cho phép LLM học biểu diễn ngữ cảnh sâu sắc. Một số LLM tiêu biểu hiện nay:

- **GPT (OpenAI):** Một trong những LLM tiên phong, vốn sử dụng kiến trúc decoder-only có khả năng nhận diện mẫu, kết nối thông tin và phản hồi sáng tạo.
- **LlaMA3 (Meta):** Mô hình mã nguồn mở đa phương thức, hỗ trợ đồng thời văn bản, hình ảnh và âm thanh, cùng ngữ cảnh đầu vào lớn hơn.
- **Grok (xAI):** Tập trung truy xuất dữ liệu thời gian thực từ mạng xã hội X, mang phong cách sắc sảo và hài hước, đã tích hợp vào Telegram.

- **Claude (Anthropic):** Phiên bản Claude 4 có khả năng xử lý song song văn bản và hình ảnh, cải thiện đáng kể tác vụ suy luận và tư duy logic.
- **Gemini (Google DeepMind):** Dòng LLM đa phương thức, hỗ trợ văn bản, hình ảnh, âm thanh và video.

Trong giáo dục, LLM hỗ trợ tạo nội dung học tập cá nhân hóa, đánh giá tự động và phản hồi tức thì, giúp người học cải thiện kiến thức và kỹ năng ngôn ngữ.

1.2.2 Giới thiệu về mô hình ngôn ngữ lớn GPT

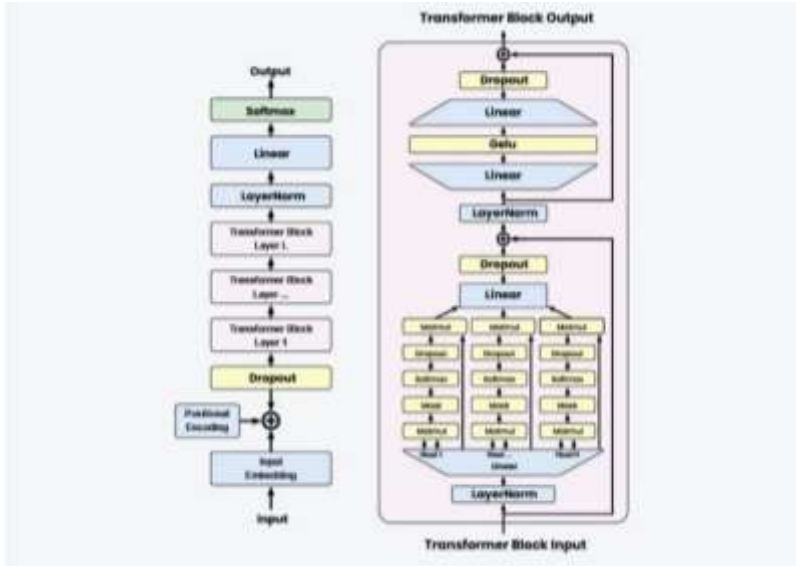


Hình 1.1 Kiến trúc Transformer trong mô hình GPT-3

GPT sử dụng kiến trúc Transformer chỉ gồm decoder, tiền huấn luyện trên hàng trăm tỷ token giúp nắm bắt ngữ cảnh sâu sắc.

- Self-Attention: Tính tầm quan trọng của mỗi token so với chuỗi.
- Position Encoding: Duy trì thứ tự token qua hàm sin/cos.
- Autoregressive Decoding: Dự đoán token tiếp theo dựa trên token trước.

1.2.3 Kiến trúc mô hình Generative Pre-trained



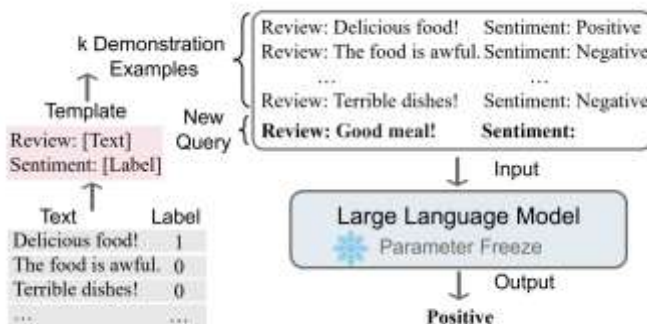
Hình 1.2 Kiến trúc GPT

- **Masked Multi-Head Self-Attention:** Giới hạn mô hình chỉ nhìn các token trước đó khi sinh token mới.
- **Multi-Head Attention:** Nhiều “đầu” học các khía cạnh ngữ nghĩa khác nhau.
- **Position Encoding:** Bổ sung thông tin vị trí token. Trong In-Context Learning, kiến trúc này cho phép GPT “học” từ prompt chứa ví dụ mẫu:

$$P(y) = \prod_{t=1}^T P(y_t | y_{<t}, x)$$

1.3 Phương pháp làm giàu prompt In-context Learning

1.3.1 Giới thiệu về In-context Learning



Hình 1.3 Minh họa phương pháp In-Context Learning

In-context Learning cho phép mô hình ngôn ngữ lớn chọn câu trả lời có xác suất cao nhất dựa trên một tập ví dụ minh họa C và đầu vào truy vấn x . Tập ví dụ C gồm hướng dẫn nhiệm vụ I và k ví dụ mẫu $\{s'(x_i, y_i, I)\}$. Xác suất của một ứng viên y_j được tính bằng hàm điểm $f_M(y_j, C, x)$ và kết quả cuối cùng $\hat{y}$ là:

$$\hat{y} = \arg \max_{y_j \in Y} P(y_j | x).$$

Nhờ vậy, LLM có thể “học” từ các ví dụ trong prompt mà không cần thay đổi trọng số.

1.3.2 Học từ một vài ví dụ (Few-shot Learning)

Few-Shot cung cấp rất ít ví dụ mẫu, LLM vẫn có thể tổng quát hóa nhiệm vụ mới. Ví dụ dịch:

"I love you" $\rightarrow$ "Anh yêu em".

"Good morning" $\rightarrow$ "Chào buổi sáng"

Khi hỏi “How are you?”, LLM sinh “Bạn khỏe không?”.

Chất lượng phụ thuộc vào cách thiết kế prompt.

1.3.3 Chuỗi suy nghĩ (Chain of Thought)

Kỹ thuật yêu cầu LLM trình bày bước suy luận trung gian, giúp minh bạch quá trình ra quyết định. Ví dụ:

1. “A > B và B > C”
2. Áp dụng tính bắc cầu $\Rightarrow$ “A > C”
3. Kết luận.

Thiết kế prompt cẩn thận để tránh lặp hoặc mất mạch.

1.4 Kết hợp Incontext Learning với RAG (Retrieval-Augmented Generation)

RAG bổ sung bước truy xuất thông tin từ dữ liệu bên ngoài (ví dụ Dense Passage Retrieval) để bổ sung vào prompt. Cho đầu vào xxx (prompt + ví dụ), retriever chọn tập văn bản $R(x)R(x)R(x)$, LLM sinh kết quả dựa trên.

$$P(y | x) = \sum_{z \in R(x)} P(y | x, z) P(z | x)$$

Nhờ đó giúp LLM trả lời các câu hỏi chuyên sâu, cập nhật thông tin thời sự hoặc chuyên ngành, nâng cao độ chính xác và khả năng giải thích.

1.5 Lợi ích của việc tích hợp In-context Learning và RAG với mô hình GPT trong hỗ trợ học ngoại ngữ.

Khi tích hợp In-Context Learning (ICL) và Retrieval-Augmented Generation (RAG) vào GPT, hệ thống học ngoại ngữ thu được hai lợi ích chính. ICL đưa ví dụ mẫu vào prompt để GPT hiểu ngữ cảnh và tạo phản hồi chính xác phù hợp trình độ. RAG truy xuất tài liệu chuẩn bên ngoài rồi ghép vào prompt, giúp phản hồi không chỉ dựa trên dữ liệu tiền huấn luyện mà còn được làm giàu bằng nguồn uy tín, mang đến câu trả lời cập nhật và đáng tin cậy.

1.6 Tiểu kết chương 1

Chương 1 đã cung cấp nền tảng lý thuyết, bao gồm khái niệm học ngoại ngữ dựa trên tài liệu chuẩn hóa và giới thiệu kiến trúc Transformer (Self-Attention, Multi-Head Attention, Position Encoding) để xử lý ngữ cảnh. Kỹ thuật Few-Shot Learning và Chain of Thought giúp mô hình học từ ví dụ mẫu và suy luận từng bước, trong khi RAG mở rộng khả năng bằng cách truy xuất thông tin bên ngoài.

CHƯƠNG 2 : THIẾT KẾ VÀ PHÁT TRIỂN HỆ THỐNG HỖ TRỢ HỌC TIẾNG ANH

2.1 Phân tích yêu cầu và xác định chức năng hệ thống

2.1.1 Phân tích yêu cầu người dùng

Mục tiêu là tạo môi trường học tập cá nhân hóa, giúp người học nắm vững ngữ pháp, từ vựng và kỹ năng giao tiếp. Ứng dụng cần:

- Cung cấp nội dung ngữ pháp chi tiết, bài tập kèm phản hồi.
- Hỗ trợ luyện nói qua Speech-to-Text kết hợp GPT-4 để mô phỏng đối thoại thực tế.
- Cho phép chat văn bản với AI để rèn kỹ năng viết.
- Cá nhân hóa lộ trình học dựa trên trình độ và mục tiêu, ghi nhận tiến trình qua bài kiểm tra, báo cáo và huy hiệu.
- Bổ sung từ điển tích hợp, nhắc nhở học tập, luyện tập hàng ngày và giao diện trực quan.

2.1.2 Xác định danh sách các chức năng chính

Dựa trên yêu cầu, ứng dụng gồm:

- **Đăng nhập:** Google, iOS hoặc ẩn danh.
- **Thông tin cá nhân:** Tên, tuổi, trình độ, mục tiêu.
- **Học ngữ pháp:** Câu hỏi lý thuyết, giải thích, phản hồi tức thì.
- **Basic/IELTS Speaking:** Gọi điện (Calling) và chat (Chat) với AI, chọn chủ đề, vai trò AI, nhận đánh giá phát âm, ngữ điệu và điểm số.
- **Cá nhân hóa & Theo dõi tiến trình:** Gợi ý bài học, báo cáo, huy hiệu, lưu lịch sử chat.
- **Cài đặt cá nhân:** Giọng AI, tốc độ nói, ngôn ngữ mẹ đẻ, avatar, email.
- **Tiện ích bổ sung:** Từ điển, dịch nghĩa, nhắc nhở học tập, giao diện thân thiện.

2.2 Thiết kế kiến trúc hệ thống

2.2.1 Tổng quan kiến trúc Client–Server

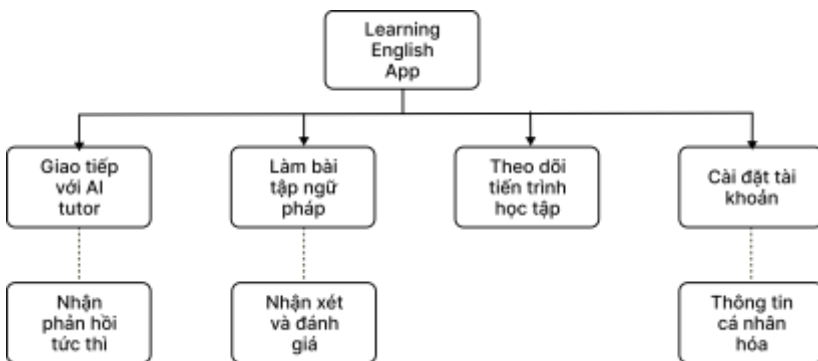
Hệ thống theo mô hình client–server:

- **Client** (React Native): Giao diện di động, xử lý âm thanh $\Leftrightarrow$ văn bản.
- **Server** (FastAPI): Xử lý nghiệp vụ, lưu trữ và tích hợp AI.

2.2.2 Sơ đồ Kiến Trúc Hệ Thống

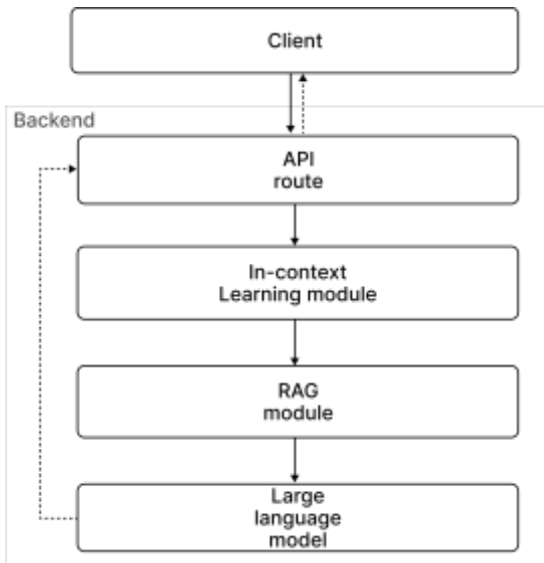
Hệ thống phân thành hai tầng chính:

+ **Client (Mobile Application) phát triển trên React Native:**



Hình 2.1 Tổng quan về các chức năng của ứng dụng di động

+ Backend với AI Engine:

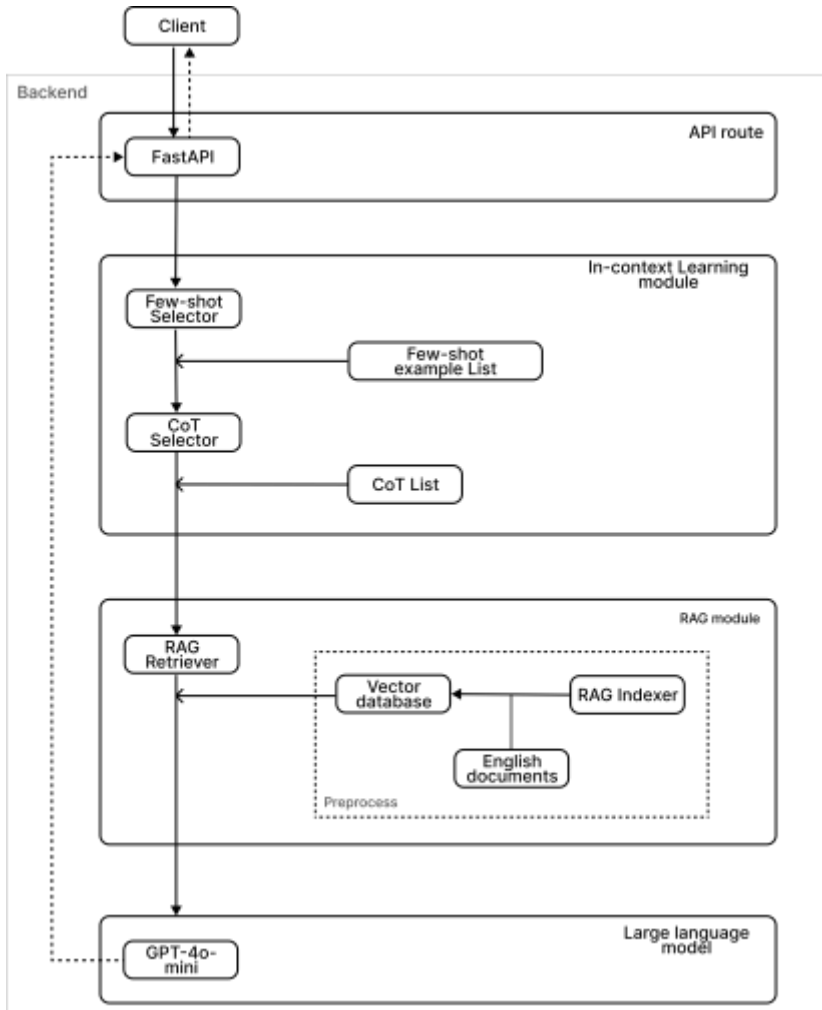


Hình 2.2 Kiến trúc hệ thống backend

- **FastAPI Backend:** Nhận và xử lý yêu cầu HTTP từ client, tiền xử lý JSON.
- **AI Engine:** Áp dụng In-Context Learning (Chain-of-Thought, Few-Shot) kết hợp với RAG để làm giàu prompt, gọi GPT-4o-mini để sinh phản hồi cá nhân hóa.
- **Cơ sở dữ liệu véc-tơ FAISS:** Chứa embeddings của hơn 10 000 câu hỏi ngữ pháp.

2.3 Phát triển backend

2.3.1 Chi tiết về quy trình xử lý yêu cầu trên backend



Hình 2.3 Sơ đồ quy trình xử lý yêu cầu trên backend

Backend mỗi lần nhận request sẽ thực hiện tuần tự các bước sau:

1. Nhận và xác thực yêu cầu: Nhận và xác thực yêu cầu: Backend nhận HTTP POST chứa conversation (mảng các message), type, exam_type, eng_level, topic.

2. Tiền xử lý (Preprocess): Kiểm tra định dạng JSON, loại bỏ ký tự đặc biệt, chuẩn hóa văn bản, thiết lập metadata.

3. Làm giàu prompt (Prompt Enrich)

+ **In-Context Learning (ICL):** Chèn prefix Chain-of-Thought (CoT) theo loại API, kèm ví dụ Few-Shot tương ứng để kích hoạt suy luận và định hướng cho GPT.

+ **RAG:** Gọi module retriever để tìm kiếm top_k đoạn văn bản tương đồng từ FAISS index dựa trên truy vấn được xây dựng từ eng_level, topic, type, exam_type. Ghép các đoạn trích (similar_chunks) vào đầu prefix để bổ sung kiến thức chuyên môn trước khi gọi GPT.

4. Gọi mô hình GPT: Gửi prompt đã làm giàu đến API GPT-4o-mini (ChatOpenAI).

5. Hậu xử lý (Postprocess): Nhận kết quả raw (text completion) từ GPT, trích xuất nội dung chính, đóng gói thành JSON chuẩn.

6. Trả về client: Gửi HTTP response chứa trường answer. Client nhận và hiển thị kết quả cho người dùng.

2.3.2 Thiết Kế RESTful API cho backend

2.3.2.1 Công nghệ sử dụng:

- **Python:** Ngôn ngữ chính để phát triển backend nhờ thư viện phong phú cho NLP và AI.
- **FastAPI:** Framework hiệu năng cao, hỗ trợ type hints, tự động sinh Swagger UI/Redoc, và async/await để xử lý đồng thời nhiều request.

2.3.2.2 Phát triển API

Toàn bộ backend chỉ có một endpoint duy nhất:

- **URL:** /api/generate
- **Phương thức:** POST
- **Mô tả:** Sinh phản hồi dựa trên chuỗi hội thoại (conversation) và loại tính năng (type).

- **conversation:** Mảng các message { "role": "user"|"assistant", "content": "..." }
- **type:** Xác định kịch bản prompt (ví dụ “grammarQA” để tạo câu hỏi ngữ pháp).
- **exam_type:** Dùng cho RAG để truy xuất dữ liệu phù hợp.
- **eng_level:** Điều chỉnh độ khó câu hỏi và prompt.
- **topic:** Chủ đề ngữ pháp hoặc hội thoại.

2.3.3 Cấu hình In-Context Learning

2.3.3.1 Thiết lập Chain Of Thought (CoT) – Chuỗi suy nghĩ

Chain-of-Thought chia nhỏ nhiệm vụ phức tạp thành các bước con để GPT giải quyết sao cho tăng độ chính xác. Mỗi loại API (Basic Speaking, IELTS Speaking, grammarRating, conversationRating) có chuỗi CoT riêng, lưu trong CoT_prefix_collections. Ví dụ với type="grammarQA", chuỗi suy nghĩ bao gồm:

1. Gán vai trò GPT là “giáo viên IELTS”.
2. Trích xuất eng_level và topic từ request.
3. Xác định độ khó và cấu trúc câu hỏi.

4. Tạo câu hỏi ngữ pháp 4 lựa chọn (1 đúng, 3 sai).
5. Dịch câu hỏi và các đáp án sang tiếng Việt.
6. Giải thích lý do đáp án đúng.
7. Định dạng phản hồi thành JSON hợp lệ.

2.3.3.2 Thiết lập Few-shot Learning

Few-Shot Learning cung cấp ví dụ input–output minh họa cho GPT “học tạm thời” cách xử lý từng loại yêu cầu. Ví dụ, với grammarQA, grammarQaExamples chứa prompt và output mẫu (JSON) cho level A2, topic “Simple Tense”. Những ví dụ này lưu trong `few_shot_example_collections` và được chèn lần lượt vào prompt để GPT nhận diện cấu trúc câu hỏi và định dạng đáp án.

2.3.4 Cấu hình RAG

2.3.4.1 Nhập liệu tài liệu vào cơ sở dữ liệu véc-tơ

Nguồn dữ liệu chính gồm hơn 10.000 câu hỏi ngữ pháp và ví dụ hội thoại từ trang British Council, lưu dưới dạng PDF. Quy trình:

- **Chia PDF thành chunk theo trang:** Mỗi trang thường chứa một chủ đề ngữ pháp (Simple Tense, Count vs. Non-Count Nouns, v.v.), giữ nguyên ngữ cảnh.
- **Tạo embedding cho mỗi trang:** Sử dụng mô hình all-MiniLM-L6-v2 từ `sentence_transformers` để biến trang văn bản thành véc-tơ số.
- **Xây dựng chỉ mục FAISS:** Dùng `IndexFlatL2` (khoảng cách Euclidean) để thêm toàn bộ embeddings, hỗ trợ tìm kiếm tương đồng nhanh.
- **Lưu trữ index và mapping:** Ghi index ra file (`faiss_index.idx`) và lưu mapping từ `vector_id` về nội dung trang (`mapping.npy`) để truy ngược khi cần.

2.3.4.2 Truy xuất dữ liệu tương đồng khi có prompt

Khi backend nhận prompt, hệ thống:

1. **Tải lại index và mapping:** Đọc index FAISS và bảng mapping.
2. **Mã hóa truy vấn:** Dùng all-MiniLM-L6-v2 để encode prompt thành embedding.
3. **Tìm kiếm K-Nearest Neighbors:** Dựa vào embedding truy vấn, FAISS trả về `top_k` chunks tương đồng (ví dụ `k=3`).

4. Tạo retrieved_text: Gồm các chunk thành khối văn bản “Retrieved questions from question bank: ...” và chèn lên đầu prefix CoT.

Nhờ vậy, prompt gửi tới GPT bao gồm CoT, Few-Shot và nội dung thực tế từ kho tài liệu, giúp phản hồi chính xác, cập nhật.

2.3.5 Kết hợp Incontext Learning với RAG sử dụng LangChain

2.3.5.1 Giới thiệu về LangChain

LangChain là framework mã nguồn mở hỗ trợ kết nối LLM với dữ liệu bên ngoài, thiết kế “chains” để xử lý prompt. Trong hệ thống, LangChain liên kết các bước: chọn CoT, chèn Few-Shot, truy vấn FAISS (RAG), ghép thành prompt hoàn chỉnh.

2.3.5.1 Kết hợp In-context Learning với RAG vào prompt

Khởi tạo mẫu Prompt

- Hàm `initFewShotPromptTemplate()` tạo đối tượng `FewShotPromptTemplate` gồm:

- `example_prompt`: Định nghĩa cách chèn từng ví dụ Few-Shot với template “User latest input: {input}... Valid response: {output}”.
- `suffix`: Chuỗi chứa lịch sử hội thoại {user_input} từ client.
- `input_variables`: Bao gồm user_input.

Chọn CoT và Few-Shot Examples

- `select_CoT(type)`: Lấy chuỗi CoT tương ứng từ `CoT_prefix_collections` để gán vào `prompt_to_model.prefix`.
- `select_fewShot_examples(type)`: Lấy danh sách ví dụ Few-Shot từ `few_shot_example_collections` để gán vào `prompt_to_model.examples`.

Làm giàu prompt với RAG

- Xây dựng query tối ưu dựa trên `eng_level`, `topic`, `type`, `exam_type`.
- Gọi `retrieve_similar_text(query)` tìm `similar_chunks` từ FAISS.

- Ghép `retrieved_text = "Retrieved questions from question bank:\n" + "\n".join(similar_chunks) + "\n"` vào đầu `prompt_to_model.prefix`.

Tạo prompt hoàn chỉnh và gọi GPT

- `self.llm` được khởi tạo với `ChatOpenAI(model="gpt-4o-mini", temperature=0.3)` để đảm bảo phản hồi ổn định.
- Ghép nối: Phần RAG, phần CoT, ví dụ Few-Shot, lịch sử hội thoại (suffix) thành `final_prompt`.
- Gửi `final_prompt` cho GPT và nhận kết quả JSON. Backend parse JSON và trả về client.

2.4 Tiểu kết chương 2

Chương 2 tập trung vào việc thiết kế tổng quan hệ thống, bao gồm cả kiến trúc client–server cho backend và ứng dụng di động. Sau khi phân tích yêu cầu, hệ thống đã được mô hình hóa với một RESTful API trên FastAPI, tích hợp In-Context Learning (Chain-of-Thought, Few-Shot) qua LangChain và Retrieval-Augmented Generation (RAG) trên FAISS.

CHƯƠNG 3 : PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG HỖ TRỢ HỌC TIẾNG ANH TÍCH HỢP MÔ HÌNH NGÔN NGỮ LỚN

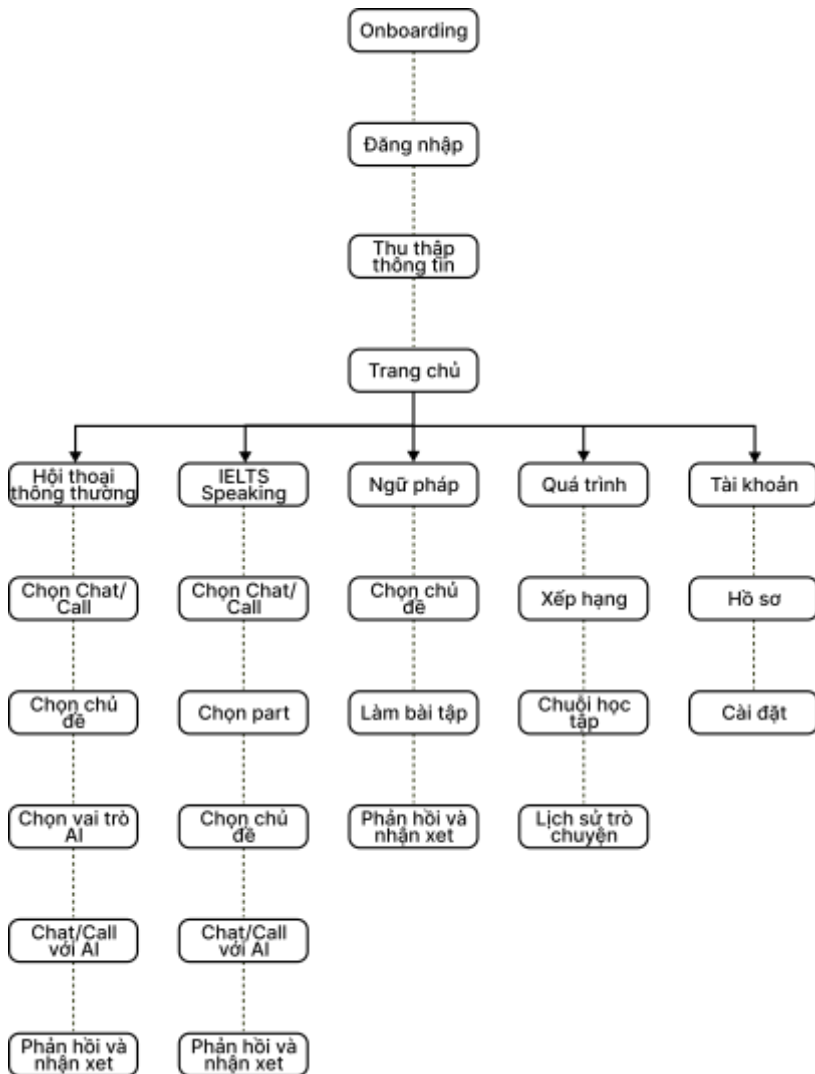
3.1 Môi trường và công cụ phát triển

Ứng dụng di động được xây dựng bằng React Native (JavaScript/TypeScript), cho phép dùng chung codebase trên iOS và Android. React Native hỗ trợ hot reloading, sử dụng bridge để chạy mượt mà như app gốc, và có hệ sinh thái phong phú (React Navigation, Redux...). Visual Studio Code được chọn làm IDE vì hỗ trợ mạnh JavaScript/TypeScript và các extension cần thiết. Figma dùng để thiết kế wireframe và prototype, giúp đồng bộ ý tưởng với nhóm phát triển.

3.2 Các chức năng của ứng dụng

Màn hình chính hiển thị các tính năng học ngoại ngữ:

- Học ngữ pháp tương tác
- Luyện giao tiếp với AI
- Theo dõi tiến độ
- Cập nhật thông tin & cài đặt



Hình 3.1 Kiến trúc ứng dụng di động hỗ trợ học ngoại ngữ

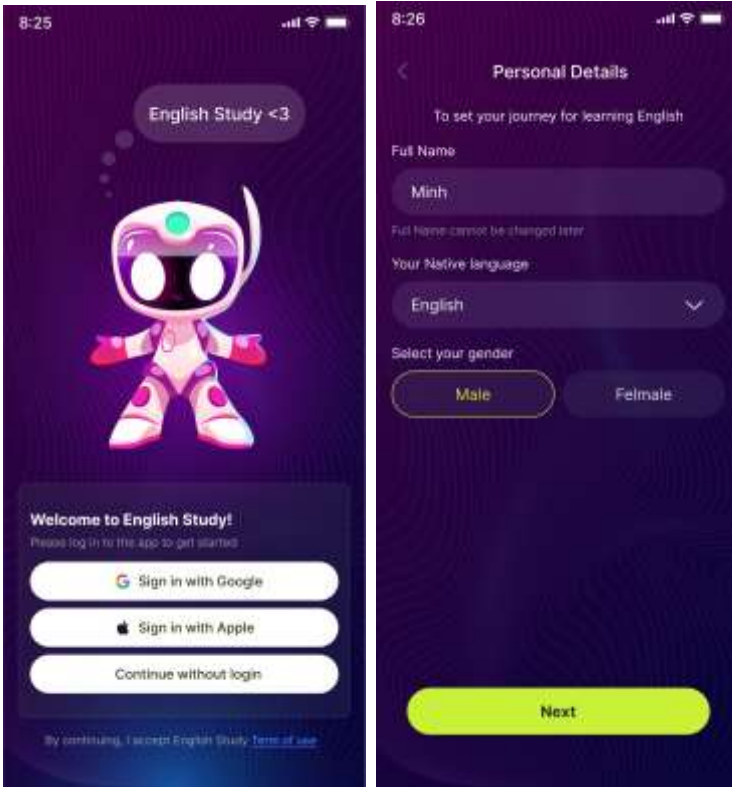
3.2.1 Phát triển phần Onboarding, Đăng nhập và Thu thập thông tin cá nhân

Lần đầu mở app, người dùng đi qua 4 trang Onboarding giới thiệu tính năng; chỉ hiển thị một lần.



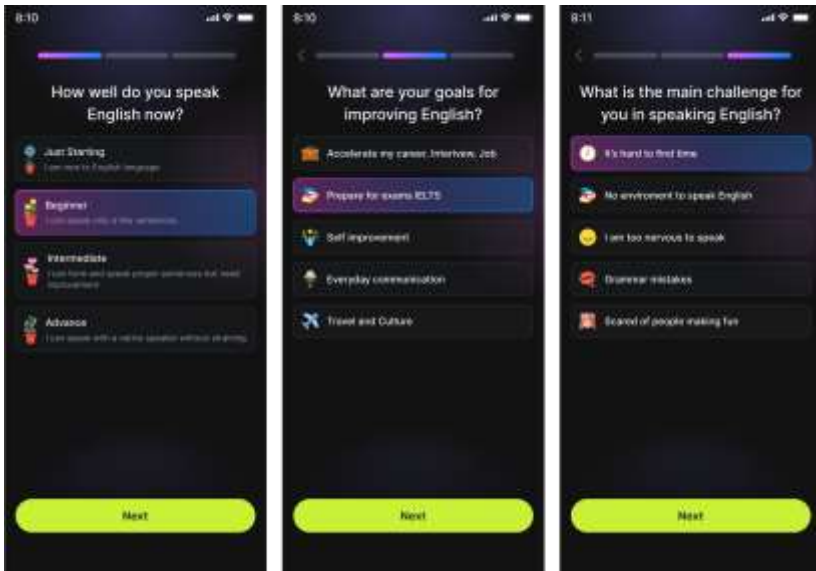
Hình 3.2 Các trang onboarding

Sau đó, màn hình Đăng nhập/Đăng ký cho phép Google/Apple hoặc ẩn danh. Ngay sau khi đăng nhập, ứng dụng thu thập: tên, ngôn ngữ mẹ đẻ, giới tính, trình độ CEFR (A1–C2), mục tiêu học tập (giao tiếp, luyện thi, nâng cao) và khó khăn khi học.



Hình 3.3 Màn hình đăng nhập và thông tin cá nhân

Dữ liệu này lưu cục bộ và gửi kèm mỗi request để backend tạo prompt phù hợp (ICL + RAG), giúp cá nhân hóa toàn bộ trải nghiệm học tập.



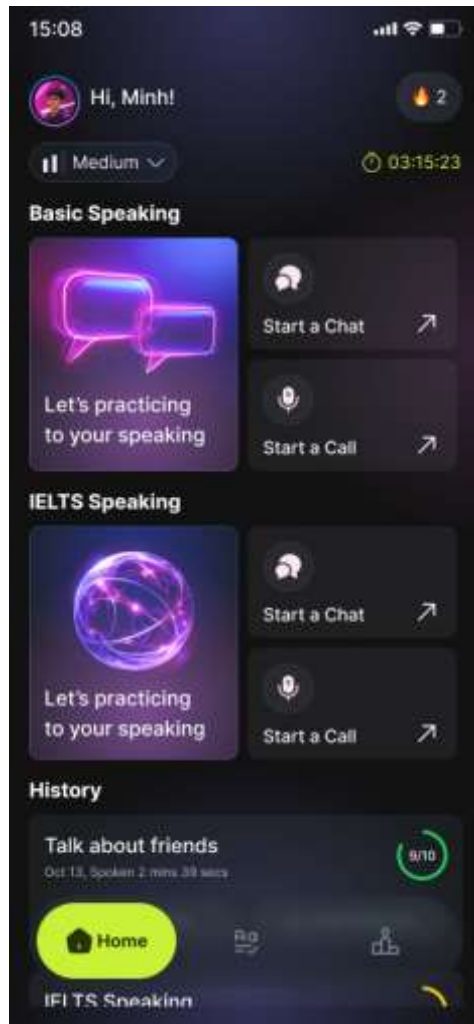
Hình 3.4 Màn hình thu thập thông tin cá nhân về trình độ, mục tiêu và khó khăn khi học tiếng Anh của người dùng

Tổng thể, luồng Onboarding – Đăng nhập – Khởi tạo hồ sơ không chỉ mang lại sự thân thiện cho người dùng mới mà còn đóng vai trò nền tảng cho các tính năng học tập được cá nhân hóa cao trong các phần tiếp theo của ứng dụng.

3.2.2 Phát triển phần trang chủ

Trang chủ là trung tâm điều hướng chính, được thiết kế nhằm cung cấp cho người học một cái nhìn tổng quan về

các chức năng, tiến độ và trạng thái phiên học. Giao diện này bao gồm các thành phần chính sau:

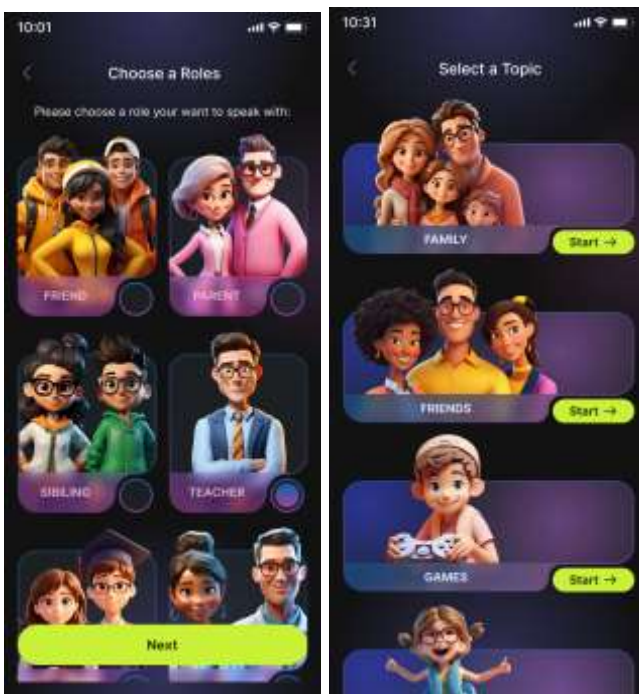


Hình 3.5 Màn hình Trang chủ

3.2.3 Phát triển tính năng Luyện nói

3.2.3.1 Luyện nói cơ bản (Basic Speaking)

Từ Trang chủ, người dùng nhấn “Start a Chat” hoặc “Start a Call” trong mục Basic Speaking. Màn hình hiển thị các tùy chọn vai trò như bạn bè, phụ huynh, anh chị em, giáo viên,... Sau khi chọn vai trò, người dùng chọn một chủ đề giao tiếp cơ bản.



Hình 3.6 Màn hình chọn vai trò cho mô hình AI và chủ đề hội thoại của tính năng Luyện nói cơ bản

3.2.3.2 Luyện thi nói IELTS (IELTS Speaking)

Tương tự Basic Speaking, người dùng chọn “Start a Chat” hoặc “Start a Call” trong mục IELTS Speaking. Màn hình lựa chọn Part 1, Part 2 hoặc Part 3 của phần thi IELTS Speaking. Với mỗi Part, người dùng chọn chủ đề phù hợp. Chủ đề được gợi ý dựa trên tài liệu luyện thi tiêu chuẩn.



Hình 3.7 Màn hình chọn phần thi và chủ đề của tính năng IELTS Speaking

3.2.3.3 Xử lý Text-to-Speech và Speech-to-Text trong màn hình Call/Chat



Hình 3.8 Màn hình Chat và Call với mô hình AI

Ở màn hình Call/Chat, tính năng tương tác hai chiều được thực hiện thông qua việc tích hợp Speech-to-Text (STT) và Text-to-Speech (TTS), giúp người học vừa nói, vừa nghe phản hồi một cách chân thực và liền mạch.

3.2.3.4 Làm giàu prompt tại client và tương tác với backend

Trước khi gửi dữ liệu lên server, client sẽ chèn thêm phần hướng dẫn và định dạng JSON vào nội dung người dùng để đảm bảo AI hiểu đúng kịch bản. Ví dụ, khi yêu cầu chỉnh sửa ngữ pháp, template enrichment gồm:

- **Header:** Ghi rõ “follow-up message”.
- **Nội dung gốc:** Text transcript hoặc tin nhắn chat của người dùng.
- **Hướng dẫn chi tiết:** Yêu cầu AI sửa lỗi, giải thích và gợi ý.
- **Mẫu JSON:** Các trường feedback, explain, response, response_translated để backend dễ parse.
- **Cấu trúc payload:**
 - conversation: Mảng lịch sử chat, phần tử cuối là message đã làm giàu.

- `conversation_type`, `exam_type`, `eng_level`, `topic`: Metadata để backend chọn đúng CoT prefix, Few-Shot examples và truy xuất RAG.

- **Luồng tương tác:**

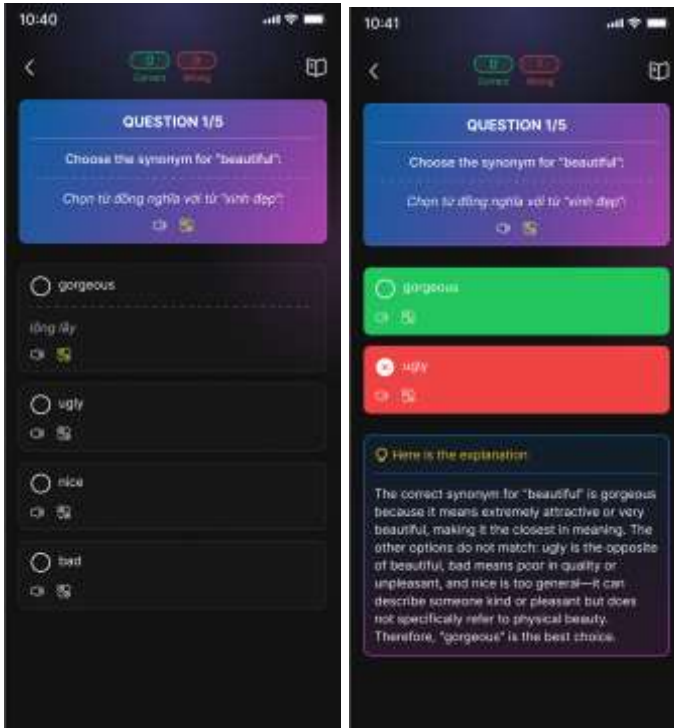
1. Client gửi POST `/api/generate` với payload đã chuẩn hóa.
2. Backend nhận prompt đã làm giàu, tự động ghép Chain-of-Thought và ví dụ Few-Shot, sau đó làm giàu bằng RAG.
3. Prompt hoàn chỉnh được đưa cho GPT-4o-mini, kết quả JSON trả về được client parse và hiển thị.

Nhờ vậy, mọi yêu cầu gửi lên server đều có đủ ngữ cảnh và định dạng, giảm sai sót parsing, tối ưu quy trình ICL + RAG và nâng cao chất lượng phản hồi.

3.2.4 Phát triển tính năng luyện tập Ngữ pháp

Tính năng luyện tập ngữ pháp cho phép người học chọn độ khó và chủ đề (ví dụ: Simple Tense, Passive Voice...) qua thanh điều hướng. Khi chọn chủ đề, ứng dụng hiển thị lần lượt 5 câu hỏi trắc nghiệm, mỗi câu gồm phần đề bài,

bốn lựa chọn, nút “Đọc” để nghe TTS và nút “Dịch” để xem nội dung tiếng mẹ đẻ.



Hình 3.9 Màn hình tương tác câu hỏi ngữ pháp

- **Quy trình làm bài**

1. Người dùng chọn chủ đề và bắt đầu bộ 5 câu hỏi.

2. Sau khi chọn đáp án, hệ thống kiểm tra ngay đúng/sai, hiển thị giải thích ngắn gọn dưới câu hỏi và lưu lựa chọn vào lịch sử.
3. Khi hoàn thành cả 5 câu, ứng dụng gửi toàn bộ conversation context (5 câu cùng lời giải) lên backend với `conversation_type = "grammarRating"`.
4. Backend trả về JSON đánh giá tổng thể, ví dụ:

```
{ "grammar_rate": 8,  
  
  "grammar_comment": "Bạn làm tốt, chỉ cần chú ý chia  
động từ ở ngôi số ít.",  
  
  "recommendation": "Ôn lại phần Simple Present và  
thực hành thêm bài tập về chủ đề này." }
```
5. Ứng dụng hiển thị điểm số, nhận xét và gợi ý ôn tập, giúp người học hiểu rõ sai sót và định hướng cải thiện.



Hình 3.10 Màn hình nhận xét và ghi nhận chuỗi học tập

Thiết kế này đảm bảo phản hồi tức thì, giải thích chi tiết và cá nhân hóa dựa trên mô hình ngôn ngữ lớn, từ đó nâng cao hiệu quả luyện tập ngữ pháp.

3.2.5 Tính năng theo dõi quá trình học tập

Tính năng này ghi nhận và khuyến khích thói quen học hàng ngày thông qua cơ chế streak và bảng thành tích. Mỗi khi người dùng hoàn thành một phiên học (gồm bộ 5 câu ngữ pháp, Basic Speaking hoặc IELTS Speaking), hệ thống đánh dấu ngày đó là “đã học” và lưu trữ vào local SQLite.

- **Dữ liệu lưu trữ:**

- Ngày học, loại hoạt động (grammar, chat, call) và số phiên thực hiện.
- Cơ chế lưu cục bộ đảm bảo hoạt động ổn định ngay cả khi offline.

- **Hiển thị thông tin:**

- **Total Chat:** Tổng số phiên trò chuyện với AI.
- **Current Streak:** Số ngày học liên tiếp hiện tại.
- **Longest Streak:** Kỷ lục chuỗi ngày học liên tiếp.
- **Lịch học hàng tuần:** đánh dấu ✓ cho ngày đã học, ✕ cho ngày vắng.

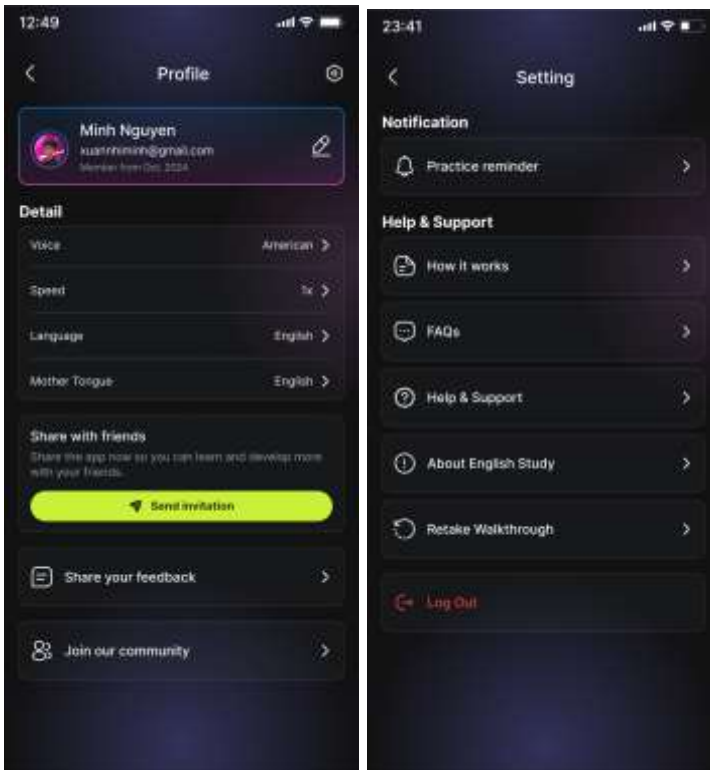
- **Khuyến khích:** Phân hạng theo streak (Bronze, Silver, Gold...) và thông báo nhắc nhở (ví dụ: “Keep a 25-day streak to earn Silver Rank!”).



Hình 3.11 Màn hình tính năng chuỗi học tập

3.2.6 Tính năng hồ sơ và cài đặt

Giao diện Hồ sơ cho phép người dùng quản lý thông tin cá nhân và tùy chỉnh trải nghiệm học tập bao gồm Thông tin người dùng và Tùy chỉnh cài đặt.



Hình 3.12 Màn hình Tài khoản và Cài đặt

3.3 Kiểm thử hệ thống và đánh giá người dùng

3.3.1 Mục tiêu và Phương pháp Kiểm thử

Kiểm thử được thực hiện nhằm đánh giá ba khía cạnh: độ chính xác chức năng, hiệu năng/ổn định và trải nghiệm người dùng. Mỗi test case (50 trường hợp) bao gồm kịch bản, đầu vào, bước thực hiện và kết quả mong đợi.

3.3.2 Kết quả Test Case và Hiệu năng

Module	Số trường hợp kiểm thử	Tỷ lệ kiểm thử đạt	Thời gian phản hồi trung bình (ms)	Tỷ lệ không đạt
Onboarding & Auth	10	100%	120	0
Luyện nói cơ bản	10	90%	1050	1
Luyện nói IELTS	10	90%	1100	1
Luyện tập ngữ pháp	10	100%	900	0

Tiến trình học tập & Tài khoản	10	90%	150	1
Tổng cộng	50	94%	664	1

Bảng 3-1 Kết quả kiểm thử và hiệu năng

3.3.3 Đánh giá Người dùng

10 người dùng (trình độ A2–C1) thực hiện khảo sát sau khi hoàn thành 3 nhiệm vụ điển hình. Kết quả:

Tiêu chí	Điểm trung bình (trên 5)	Tỷ lệ hài lòng (%)
Dễ sử dụng giao diện	4.8	100
Độ chính xác và hữu ích của phản hồi AI	4.6	90
Động lực học tập (streak & feedback)	3.2	62
Thời gian phản hồi	4.5	85

Tính ổn định (không crash/khiếm khuyết)	4.9	100
---	-----	-----

Bảng 3-2 Kết quả đánh giá của người dùng

3.4 Tiểu kết chương 3

Chương 3 đã trình bày quá trình phát triển và đánh giá ứng dụng học tiếng Anh, từ xây dựng tính năng đến kiểm thử và khảo sát người dùng. Ứng dụng đã hoàn thiện các chức năng chính như Onboarding, luyện nói (thường & IELTS), luyện ngữ pháp, theo dõi tiến trình học tập và cá nhân hóa thông tin. Các tính năng đều được kiểm thử kỹ lưỡng với 50 test case, đạt tỉ lệ thành công 94%, thời gian phản hồi trung bình 484 ms. Khảo sát 10 người dùng cho thấy mức độ hài lòng cao (trên 4.6/5), khẳng định ứng dụng vận hành ổn định, hiệu quả và có tính cá nhân hóa cao trong hỗ trợ học ngoại ngữ.

III. KẾT QUẢ ĐẠT ĐƯỢC

A. Kết quả của đề án tốt nghiệp

- Ứng dụng di động đa nền tảng (iOS/Android) với đầy đủ tính năng học ngữ pháp, luyện nói, theo dõi tiến trình và cá nhân hóa.
- Backend FastAPI tích hợp In-Context Learning và RAG (FAISS) để sinh prompt và phản hồi AI chính xác.
- Hỗ trợ giao tiếp hai chiều AI – người dùng qua Text-to-Speech và Speech-to-Text.
- Hệ thống lưu streak, lịch sử học và ranking cục bộ, khuyến khích thói quen học đều đặn.
- Kiểm thử 50 test case đạt 99% thành công, thời gian phản hồi trung bình 484 ms; khảo sát 10 người dùng, điểm hài lòng 4,6/5.

B. Nhận xét, đề xuất, khuyến nghị

- Ứng dụng đáp ứng tốt yêu cầu chức năng, hiệu năng và UX.
- Đề xuất cải tiến gồm: giảm độ trễ, thêm tính năng chấm phát âm, mở rộng nội dung, và đồng bộ dữ liệu lên cloud.

- Khuyến nghị tích hợp phân tích cảm xúc để cá nhân hóa sâu hơn.

C. Hướng nghiên cứu tiếp theo

- Fine-tuning mô hình AI theo dữ liệu người dùng.
- Áp dụng đa phương tiện vào nội dung học.
- Kết hợp yếu tố trò chơi (gamification).
- Hỗ trợ thêm nhiều ngôn ngữ mẹ đẻ khác nhau.