1. **Bảng đánh giá thành viên**
2. **Phát biểu bài toán sơ lược**

Là trưởng nhóm phát triển của một công ty phần mềm, bạn được yêu cầu phát triển một ứng dụng học tiếng Anh nâng cao khả năng giao tiếp. Hệ thống này giúp kết nối người học với một AI đối thoại, cho phép họ thực hành kỹ năng nói và phản xạ tiếng Anh một cách liên tục thông qua môi trường mô phỏng giao tiếp một-kèm-một. Toàn bộ quá trình luyện tập từ việc nhận dạng giọng nói, phản hồi theo ngữ cảnh, đến hỗ trợ từ vựng đều được thực hiện thông qua nền tảng số.

Người dùng bắt đầu bằng cách nói câu tiếng Anh, hệ thống sử dụng công nghệ nhận dạng giọng nói (speech-to-text) để chuyển đổi giọng nói thành văn bản. Sau đó, người học có thể chỉnh sửa văn bản nếu cần và gửi lại cho hệ thống để tiếp tục cuộc hội thoại. AI sẽ phản hồi bằng văn bản phù hợp với ngữ cảnh và có thể chuyển đổi sang giọng nói để giúp người học nghe cách phát âm chuẩn.

Để hỗ trợ nâng cao kỹ năng ngôn ngữ, hệ thống cung cấp các tính năng như tra cứu từ điển tích hợp, giải thích từ vựng ngay trong văn bản, và hướng dẫn phát âm. Ngoài ra, hệ thống sử dụng mô hình ngôn ngữ lớn để đảm bảo phản hồi tự nhiên và mang tính đối thoại.

Phần mềm có thể được phát triển trên nền tảng web hoặc ứng dụng di động, sử dụng công nghệ tiên tiến như ReactJS, Angular, Swift hoặc Kotlin. Hệ thống tích hợp API từ các nhà cung cấp dịch vụ như Google Cloud, IBM Watson hoặc Microsoft Azure để thực hiện nhận dạng giọng nói và chuyển đổi văn bản thành giọng nói.

Mục tiêu của phần mềm là mang đến một nền tảng học tiếng Anh tương tác cao, giúp người dùng cải thiện kỹ năng giao tiếp bằng công nghệ hiện đại và trải nghiệm thực tế.

1. **Giải pháp đề xuất**
   1. ***Phần mềm***
      1. ***Danh sách các chức năng phần mềm***

|  |  |
| --- | --- |
| **Nhu cầu** | **Yêu cầu** |
| Là người học tiếng Anh, tôi muốn có thể nói chuyện với AI để thực hành phản xạ và giao tiếp | Chức năng nhận dạng giọng nói và phản hồi bằng AI |
| Là người dùng, tôi muốn có thể chỉnh sửa câu nói trước khi gửi đi để đảm bảo nội dung chính xác | Chức năng chỉnh sửa văn bản sau khi nhận dạng giọng nói |
| Là người dùng, tôi muốn AI có thể phản hồi một cách tự nhiên và phù hợp với ngữ cảnh | Chức năng mô hình ngôn ngữ lớn (LLM) tạo phản hồi theo ngữ cảnh |
| Là người học, tôi muốn nghe giọng nói phát âm đúng chuẩn để học cách nói chuẩn | Chức năng chuyển văn bản thành giọng nói (text-to-speech) |
| Là người học, tôi muốn có thể tra nghĩa và cách phát âm của từ khó ngay khi gặp | Chức năng hỗ trợ từ vựng, tích hợp từ điển |
| Là người dùng, tôi muốn giao diện dễ sử dụng, không gây khó khăn khi thao tác | Yêu cầu về thiết kế giao diện thân thiện với người dùng |
| Là người dùng, tôi muốn thời gian xử lý mỗi thao tác không quá 3 giây để đảm bảo trải nghiệm mượt mà | Yêu cầu tối ưu hiệu suất hệ thống |
| Là người dùng, tôi muốn hệ thống bảo mật dữ liệu cá nhân và không chia sẻ thông tin với bên thứ ba | Yêu cầu bảo mật và quyền riêng tư |

***Danh sách chức năng phần mềm***

1. **Nhận dạng giọng nói**: Chuyển đổi giọng nói của người dùng thành văn bản.
2. **Chỉnh sửa văn bản:** Cho phép người dùng chỉnh sửa câu trước khi gửi.
3. **Phản hồi bằng mô hình ngôn ngữ lớn (LLM):** Tạo phản hồi phù hợp với ngữ cảnh cuộc hội thoại.
4. **Chuyển văn bản thành giọng nói (TTS):** Đọc phản hồi bằng giọng nói để người học nghe cách phát âm chuẩn.
5. **Tra cứu từ vựng:** Cho phép người dùng tra nghĩa và cách phát âm của từ ngay trong ứng dụng.
6. **Gợi ý câu hỏi tiếp theo**: Thúc đẩy cuộc đối thoại liên tục.
7. **Giao diện trực quan, dễ sử dụng:** Đảm bảo người dùng thao tác dễ dàng.
8. **Tích hợp API từ Google Cloud, IBM Watson, Microsoft Azure**: Hỗ trợ các chức năng nhận dạng giọng nói và chuyển văn bản thành giọng nói.
9. **Bảo mật dữ liệu:** Mã hóa thông tin người dùng, đảm bảo quyền riêng tư.
10. **Tối ưu hiệu suất**: Đảm bảo phản hồi nhanh chóng, không gây chậm trễ.
    * 1. ***Kiến trúc tổng thể phần mềm***

Để đảm bảo đáp ứng đầy đủ danh sách các chức năng đã liệt kê, hệ thống phần mềm sẽ được thiết kế theo ***kiến trúc*** ***Client – server*** kết hợp ***kiến trúc đa tầng*** với các thành phần:

|  |  |
| --- | --- |
| **Tầng** | **Chức năng** |
| **Presentation Tier** | Giao diện người dùng (React, Web, App mobile...) |
| **Application/ Logic Tier** | Xử lý logic, gọi API như speech-to-text, GPT, kiểm tra đăng nhập... |
| **Data Tier** | Lưu trữ dữ liệu người dùng, lịch sử hội thoại, tài khoản (MySQL, Firebase...) |

Ngoài ra, backend cũng được tổ chức theo mô hình **đa lớp**, bao gồm lớp controller, service, và data access để dễ bảo trì và mở rộng.

* 1. ***Phần cứng***

Chỉ cần một máy tính (pc) hoặc 1 laptop là phần mềm có thể hoạt động.

1. **Kế hoạch phát triển**
   1. ***Phân tích yêu cầu***
   2. ***Thiết kế phần mềm***
   3. ***Cài đặt phần mềm***
   4. ***Kiếm thử phần mềm***
   5. ***Triển khai, bảo trì***
2. **Kế hoạch nhân sự và chi phí**