**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**Kết Quả Thực Hiện**

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Bộ môn Công nghệ phần mềm

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

**MỤC LỤC**

[Bảng đánh giá thành viên 3](#_Toc200745011)

[Môi trường phát triển và Môi trường triển khai 4](#_Toc200745012)

[Kết quả đạt được 5](#_Toc200745013)

[Hướng phát triển 7](#_Toc200745014)

# Bảng đánh giá thành viên

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ Tên** | **% đóng góp (tối đa 100%)** | **Chữ ký** |
| 22810209 | Lâm Trọng Nghĩa | 25% |  |
| 23880002 | Nguyễn Tuấn Anh | 25% |  |
| 23880085 | Trương Ngọc Tin | 25% |  |
| 23880288 | Trần Thanh Tùng | 25% |  |

# Môi trường phát triển và Môi trường triển khai

* ***Môi trường phát triển ứng dụng:***
  + Hệ điều hành: Windows 10/11
  + Hệ quản trị cơ sở dữ liệu: MySQL Server (Phiên bản 5.7 trở lên)
  + Công cụ quản lý dữ liệu: DBeaver
  + Công cụ phân tích thiết kế: Draw.io, Visual paradigm (phiên bản Community), dbdesigner.net
  + Công cụ xây dựng ứng dụng: Visual Studio Code
  + Các thư viện đã dùng:
    - Các thư viện sử dụng chính:
    - Backend: express, mysql2, dotenv, openai, uuid, cors, body-parser
    - Frontend: React, react-boostrap, axios, react-router-dom, openai, dotenv, react-icons
* ***Môi trường triển khai ứng dụng:***
  + Hệ điều hành: Windows 10/11
  + Cần cài đặt:
    - Node.js (Khuyến nghị phiên bản 18.x trở lên),
    - MySQL Server (nếu không sử dụng Docker)
    - Docker (nếu chạy bằng container, không cần cài MySQL thủ công)

# Kết quả đạt được

**1. Kết quả đã đạt được**

**a. Phân tích & Thiết kế chi tiết**

* **Phân tích yêu cầu**: Đã thực hiện phân tích nghiệp vụ, xác định các chức năng chính như: đăng ký/đăng nhập, chat với AI, lưu lịch sử hội thoại, lưu từ vựng, quản lý người dùng, v.v.
* **Thiết kế hệ thống**:
  + Thiết kế kiến trúc client-server, sử dụng ReactJS cho frontend, NodeJS/Express cho backend, MySQL cho database.
  + Thiết kế API RESTful rõ ràng, áp dụng nguyên tắc SOLID trong phát triển backend.
  + Thiết kế giao diện người dùng hiện đại, responsive, sử dụng Bootstrap và React-Bootstrap.

**b. Các chức năng đã cài đặt hoàn chỉnh**

* **Đăng ký/Đăng nhập người dùng**:
  + Đã hoàn thiện backend (API /api/user/register, /api/user/login) và frontend (form, xác thực, lưu trạng thái đăng nhập).
* **Nhận diện giọng nói và phản hồi bằng AI (OpenAI API)**:
  + Người dùng có thể tương tác với AI thông qua chức năng nhận diện giọng nói, nhận phản hồi từ AI thông qua văn bản và âm thanh   
    (Giao tiếp liên tục - chế độ Conversation / Giao tiếp thủ công có chỉnh sửa - chế độ Manual)
* **Chỉnh sửa văn bản sau khi nhận dạng giọng nói:** 
  + Sau khi nhận diện giọng nói từ người dùng, nội dung được hiển thị trong khung chat và cho phép chỉnh sửa để phù hợp trước gửi.   
    (Chế độ Giao tiếp thủ công có chỉnh sửa - Manual)
* **Gợi ý câu hỏi tiếp theo nhằm thúc đẩy cuộc đối thoại**
  + Hiển thị nội dung gợi ý sau mỗi lần phản hồi, người dùng có thể chọn để tiếp tục triển khai ý tưởng, cuộc đối thoại phù hợp với ngữ cảnh hiện tại.
* **Lưu và xem lại lịch sử hội thoại**:
  + Hiển thị danh sách các cuộc hội thoại đã lưu.
  + Cho phép chọn để xem lại chi tiết các cuộc hội thoại.
* **Lưu và quản lý từ vựng**:
  + Cho phép lưu các từ mới, hiển thị danh sách từ đã lưu, xem chi tiết nghĩa của từng từ.
* **Giao diện người dùng**:
  + Đã hoàn thiện giao diện chính, header, chat UI, responsive tốt trên PC/Laptop.

**2. Điểm đặc sắc của đề tài**

* **Tích hợp AI (OpenAI API)**:
  + Cho phép người dùng trò chuyện với AI, hỗ trợ học tiếng Anh theo ngữ cảnh thực tế.
* **Kiến trúc hiện đại, dễ mở rộng**:
  + Ứng dụng xây dựng theo mô hình client-server, frontend tách biệt backend.
  + Áp dụng các nguyên tắc SOLID (đặc biệt ở phần backend sử dụng Node.js/Express).
  + Dễ dàng mở rộng hoặc tích hợp thêm tính năng mới.
* **Sử dụng Docker**:
  + Hỗ trợ triển khai nhanh chóng qua Docker Compose, thuận tiện cho việc đóng gói và triển khai.
* **Giao diện thân thiện, responsive**:
  + Sử dụng React, Bootstrap, tối ưu trải nghiệm người dùng trên nhiều thiết bị.
* **Quản lý từ vựng cá nhân**:
  + Cho phép lưu, xem lại, tra nghĩa từ vựng đã học trong quá trình chat với AI.
* **Triển khai hoàn chỉnh trên nền tảng Render:**
  + Ứng dụng được deploy và vận hành ổn định trên nền tảng host Render, hỗ trợ trình diễn trực tiếp và thử nghiệm tính năng.
* **Có thể mở rộng thêm các tính năng động**:
  + Kiến trúc backend và frontend đều hỗ trợ dễ dàng bổ sung các module mới (ví dụ: plug-in, đổi loại CSDL, thêm API...).

# Hướng phát triển

**1. Nâng cao trải nghiệm người dùng**

* Bổ sung chức năng đổi giao diện (skin/theme): Cho phép người dùng lựa chọn giao diện sáng/tối hoặc các chủ đề màu sắc khác nhau.
* Tối ưu giao diện trên thiết bị di động: Đảm bảo ứng dụng hoạt động mượt mà trên smartphone, tablet.

**2. Mở rộng chức năng học tập**

* Tích hợp chức năng luyện phát âm: Sử dụng AI để đánh giá phát âm của người học, đưa ra nhận xét và hướng dẫn sửa lỗi.
* Thêm các mini-game học từ vựng: Giúp người dùng học từ mới qua trò chơi, tăng tính tương tác.
* Tạo hệ thống nhắc nhở học tập: Gửi thông báo nhắc nhở người dùng ôn tập từ vựng hoặc luyện hội thoại hàng ngày.

**3. Phát triển hệ thống quản trị**

* Phân quyền người dùng: Thêm chức năng quản trị viên, giáo viên, học viên với các quyền truy cập khác nhau.
* Thống kê, báo cáo tiến độ học tập: Hiển thị biểu đồ, số liệu về quá trình học của từng người dùng.

**4. Mở rộng tích hợp công nghệ**

* Hỗ trợ đa ngôn ngữ: Không chỉ tiếng Anh, có thể mở rộng sang các ngôn ngữ khác như tiếng Nhật, Hàn, Trung...
* Tích hợp thêm các API AI khác: Ví dụ như Google Speech-to-Text, Text-to-Speech để nâng cao khả năng giao tiếp bằng giọng nói.
* Cho phép bổ sung tính năng động qua plug-in: Xây dựng hệ thống plug-in để cộng đồng có thể phát triển thêm các tính năng mới.

**5. Nâng cao bảo mật và hiệu năng**

* Bảo mật thông tin người dùng: Mã hóa dữ liệu nhạy cảm, xác thực hai lớp (2FA).
* Tối ưu hiệu năng hệ thống: Sử dụng cache, tối ưu truy vấn database, hỗ trợ mở rộng theo chiều ngang (scaling).

**6. Triển khai thực tế và thương mại hóa**

* Triển khai trên cloud (AWS, Azure, GCP): Đảm bảo khả năng mở rộng và vận hành ổn định.
* Xây dựng mô hình kinh doanh: Cung cấp các gói học tập trả phí, tài khoản premium với nhiều tính năng nâng cao.