**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**Kết Quả Thực Hiện**

Yêu cầu nhóm sinh viên hoàn thành tài liệu Kết Quả Thực Hiện cho đồ án đã được giao theo biểu mẫu đính kèm.

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**

1. Môi trường phát triển và Môi trường triển khai 4

2. Kết quả đạt được 8

3. Hướng phát triển 12

# Bảng đánh giá thành viên

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ Tên** | **% đóng góp (tối đa 100%)** | **Chữ ký** |
| 23880008 | Nguyễn Quốc Bình | 25% |  |
| 23880094 | Võ Xuân Tú | 25% |  |
| 23880047 | Võ Vĩnh Nghi | 25% |  |
| 21810226 | Phạm Hoàng Nhật Trường | 25% |  |

# Môi trường phát triển và Môi trường triển khai

* ***Môi trường phát triển ứng dụng:***
* ***Hệ điều hành:***

*+ Windows 10/11 hoặc Linux (Ubuntu 20.04 trở lên), vì dự án sử dụng Node.js và FFmpeg, vốn tương thích tốt với cả hai hệ điều hành này.*

*+ Phát triển chủ yếu trên môi trường cục bộ (localhost).*

* ***Hệ quản trị cơ sở dữ liệu (nếu có dùng):***

*+ PostgreSQL (phiên bản 13 hoặc mới hơn), được cấu hình trong config.json cho môi trường development và production.*

* ***Công cụ dùng để phân tích, thiết kế (ví dụ Rational Rose…):***

*+ Lucidchart hoặc Draw.io để vẽ sơ đồ khối và luồng dữ liệu (dựa trên kiến trúc client-server và tích hợp API bên thứ ba).*

*+ Postman để thiết kế và kiểm tra các API (/chat, /upload-audio).*

* ***Công cụ đã dùng để xây dựng ứng dụng:***

*+ Visual Studio Code (VS Code): Trình soạn thảo mã nguồn chính, hỗ trợ extension cho JavaScript, Node.js, và Handlebars.*

*+ Node.js (phiên bản 16.x hoặc 18.x) và npm (phiên bản 8.x) để quản lý thư viện và chạy server.*

*+ pgAdmin 4 để quản lý cơ sở dữ liệu PostgreSQL.*

*+ Google Cloud Console để cấu hình và kiểm tra Google Cloud Speech-to-Text.*

*+ Terminal (Command Prompt trên Windows hoặc Bash trên Linux) để chạy lệnh npm, cấu hình FFmpeg, và triển khai cục bộ.*

* ***Các thư viện đã sử dụng:***
* ***Backend****:*

*+ express: Framework web cho Node.js.*

*+ sequelize: ORM cho PostgreSQL.*

*+ pg, pg-hstore: Driver PostgreSQL cho Sequelize.*

*+ socket.io: Hỗ trợ chat thời gian thực.*

*+ multer: Xử lý upload file âm thanh.*

*+ fluent-ffmpeg, ffmpeg-static: Chuyển đổi định dạng âm thanh.*

*+ openai: Gọi API OpenAI (gpt-4o-mini).*

*+ express-handlebars: Template engine cho giao diện.*

*+ dotenv: Quản lý biến môi trường.*

* ***Frontend****:*

*+ jquery: Xử lý DOM và AJAX (main.js).*

*+ slick-carousel: Tạo slider cho trang chủ và sản phẩm.*

*+ font-awesome: Icon cho giao diện.*

*+ socket.io-client: Kết nối Socket.IO từ client (trong client.js).*

* ***Dịch vụ bên thứ ba****:*

*+ Google Cloud Speech-to-Text SDK.*

*+ OpenAI API.*

* ***Môi trường triển khai ứng dụng:***

*Để chạy được ứng dụng, cần thiết lập cấu hình máy tính/server như sau:*

* ***Hệ điều hành****:*

*+ Linux (Ubuntu 20.04 hoặc CentOS 8) cho server production, hoặc Windows Server*

*nếu cần.*

*+ Khuyến nghị Linux vì FFmpeg và PostgreSQL hoạt động ổn định hơn.*

* ***Cấu hình phần cứng tối thiểu****:*

*+ CPU: 2 cores, 2.0 GHz.*

*+ RAM: 4GB (8GB khuyến nghị để xử lý âm thanh và API).*

*+ Dung lượng ổ cứng: 20GB (bao gồm file âm thanh tạm thời trong thư mục uploads).*

*+ Kết nối internet ổn định để gọi OpenAI và Google Cloud APIs*

* ***Phần mềm và thư viện cần cài đặt****:*

*+ Node.js (v16.x hoặc v18.x) và npm (v8.x).*

*+ PostgreSQL (v13 hoặc mới hơn), cấu hình user/password như trong config.json.*

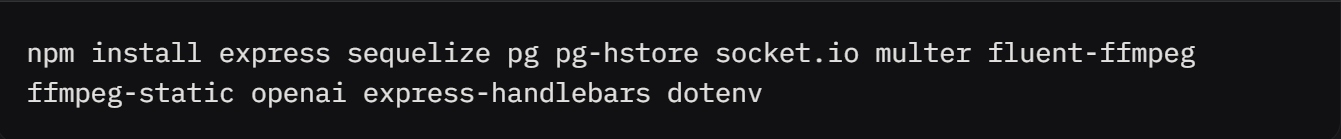
*+ FFmpeg: Cài đặt trên server để chuyển đổi âm thanh (sudo apt-get install ffmpeg trên Ubuntu).*

*+ Google Cloud SDK: Cài đặt và cấu hình credentials (GOOGLE\_APPLICATION\_CREDENTIALS).*

*+ PM2 (khuyến nghị): Quản lý tiến trình Node.js trong production (npm install -g pm2).*

*+ Nginx hoặc Apache: Proxy ngược để xử lý yêu cầu HTTP và HTTPS.*

*+ SSL Certificate (Let’s Encrypt): Cấu hình HTTPS cho production.*

*+ Các thư viện Node.js: Chạy npm install để cài các thư viện từ package.json:*

# *+ Biến môi trường: Tạo file .env với các giá trị:*

# *Các bước triển khai:*

# *Clone repository từ GitHub.*

# *Cài đặt Node.js, PostgreSQL, FFmpeg, và Google Cloud SDK.*

# *Chạy npm install trong thư mục dự án.*

# *Cấu hình .env và config.json.*

# *Khởi tạo cơ sở dữ liệu PostgreSQL và chạy migrations (nếu có) qua Sequelize:*

1. *Chạy server:*
2. *Cấu hình Nginx để proxy yêu cầu đến localhost:3000 (port mặc định của Express).*
3. *Kiểm tra kết nối OpenAI và Google Cloud APIs.*

# Kết quả đạt được

# *Phân tích và thiết kế:*

# *Chức năng đã phân tích và thiết kế chi tiết:*

# *Chatbot học tiếng Anh:*

# *Giao tiếp văn bản qua HTTP (/chat) và Socket.IO (chat message).*

# *Gửi tin nhắn đến OpenAI để nhận phản hồi và gợi ý tiếp theo.*

# *Lưu trữ tin nhắn trong cơ sở dữ liệu (conversation, message).*

# *Xử lý giọng nói:*

# *Ghi âm từ browser (MediaRecorder), upload file .webm.*

# *Chuyển đổi .webm sang .flac và sử dụng Google Cloud Speech-to-Text để lấy transcription.*

# *Tích hợp transcription vào giao diện chat như tin nhắn văn bản.*

# *Quản lý lịch sử trò chuyện:*

# *Lưu trữ server-side (PostgreSQL) và client-side (localStorage).*

# *Hiển thị danh sách cuộc trò chuyện với khả năng xóa.*

# *Giao diện người dùng:*

# *Giao diện chat responsive với sidebar, input văn bản, nút ghi âm, và gợi ý.*

# *Trang chủ và trang sản phẩm với slider, navbar, và checkout form.*

# *Tích hợp API:*

# *OpenAI API (gpt-4o-mini) cho phản hồi thông minh.*

# *Google Cloud Speech-to-Text cho xử lý giọng nói (hỗ trợ en-US và vi-VN).*

# *Chức năng đã cài đặt hoàn chỉnh:*

# *Giao diện chat (chat.hbs, index.hbs, script.js):*

# *Hiển thị welcome message, tin nhắn người dùng và bot.*

# *Input văn bản, nút gửi, nút ghi âm, và nút đính kèm (chưa hoạt động).*

# *Gợi ý tĩnh và động từ OpenAI.*

# *Text-to-speech cho phản hồi bot.*

# *Xử lý tin nhắn văn bản (chatController.js, chatRouter.js):*

# *Gửi tin nhắn qua POST /chat, nhận phản hồi từ OpenAI.*

# *Lưu tin nhắn vào message (PostgreSQL).*

# *Hỗ trợ Socket.IO (trong client.js, nhưng bị comment).*

# *Xử lý âm thanh (audioController.js, audioRouter.js):*

# *Upload file âm thanh qua POST /upload-audio.*

# *Chuyển đổi .webm sang .flac và lấy transcription từ Google Cloud.*

# *Trả về transcription và URL file âm thanh.*

# *Lịch sử trò chuyện client-side (script.js):*

# *Lưu vào localStorage, hỗ trợ tạo mới, chọn, và xóa cuộc trò chuyện.*

# *Giao diện ngoài chat (main.js, style2.css):*

# *Navbar với dropdown hover.*

# *Slider cho trang chủ và sản phẩm.*

# *Form checkout với số lượng và điều kiện hiển thị.*

# *Chức năng đã cài đặt nhưng chưa hoàn chỉnh*

# *Socket.IO (client.js, index.js):*

# *Đã triển khai nhưng bị comment trong client.js, chưa được tích hợp vào script.js.*

# *Cần kích hoạt để hỗ trợ chat thời gian thực.*

# *Xác thực người dùng (layout.hbs):*

# *Navbar hiển thị isLoggedIn, nhưng chưa có route /login, /logout, /history, /newChat.*

# *Hardcode userId: 1 và conversationId trong chatController.js và indexController.js.*

# *Đính kèm file (script.js):*

# *Nút #attach-btn hiển thị nhưng chỉ hiện alert, chưa xử lý upload file.*

# *Lịch sử trò chuyện server-side:*

# *Lưu vào conversation và message, nhưng script.js dùng localStorage thay vì đồng bộ với server.*

# *Chức năng chỉ có giao diện nhưng chưa xử lý*

# *Error page (error.hbs):*

# *Giao diện hiển thị thông báo lỗi, nhưng chưa có route hoặc controller render view này.*

# *Sidebar và lịch sử trò chuyện (chat.hbs):*

# *Thiếu #chat-history-list và #suggestions-fixed trong HTML, dù được xử lý trong script.js.*

# *Các route trong navbar (layout.hbs):*

# */history, /newChat, /logout, /login hiển thị nhưng chưa có router xử lý.*

* ***Điểm đặc sắc của đề tài***
* ***Tích hợp AI và giọng nói:***
* *Sử dụng OpenAI (gpt-4o-mini) để tạo phản hồi thông minh và gợi ý tiếp theo, hỗ trợ luyện tập tiếng Anh.*
* *Google Cloud Speech-to-Text hỗ trợ đa ngôn ngữ (en-US, vi-VN) với tự động chèn dấu câu.*
* ***Kiến trúc linh hoạt:***
* *Sử dụng MVC (Model: Sequelize, View: Handlebars, Controller: Express controllers).*
* *Tích hợp Socket.IO cho khả năng mở rộng chat thời gian thực.*
* *Hỗ trợ thay đổi cơ sở dữ liệu (PostgreSQL/MySQL) qua config.json.*
* ***Giao diện người dùng hiện đại:***
* *Responsive với Flexbox (style.css), slider (main.js), và hiệu ứng động (shake-anim).*
* *Text-to-speech tích hợp để đọc phản hồi bot, tăng trải nghiệm học tập.*
* ***Khả năng mở rộng:***
* *Dễ dàng thêm ngôn ngữ mới cho Speech-to-Text hoặc model AI khác từ OpenAI.*
* *Có thể tích hợp plugin xác thực (Passport.js) hoặc cache (Redis).*

## Hướng phát triển

1. ***Tích hợp xác thực người dùng****:* 
   * *Thêm hệ thống đăng nhập/đăng ký với Passport.js hoặc JWT.*
   * *Đồng bộ userId và conversationId với phiên đăng nhập.*
   * *Triển khai route /history, /newChat, /logout, /login.*
2. ***Hoàn thiện Socket.IO****:* 
   * *Kích hoạt Socket.IO trong script.js để chat thời gian thực.*
   * *Đồng bộ lịch sử trò chuyện giữa localStorage và PostgreSQL.*
3. ***Cải thiện xử lý âm thanh****:* 
   * *Thêm timer và nút cancel cho ghi âm trong script.js (như client.js).*
   * *Tối ưu hóa upload âm thanh bằng cách nén file trước khi gửi.*
   * *Hỗ trợ nhiều định dạng âm thanh đầu vào (MP3, WAV).*
4. ***Tăng cường giao diện****:* 
   * *Hợp nhất chat.hbs và index.hbs, thêm sidebar với lịch sử trò chuyện.*
   * *Triển khai chức năng đính kèm file (hình ảnh, PDF) qua Multer.*
   * *Thêm theme tùy chỉnh (light/dark mode).*
5. ***Hiệu suất và bảo mật****:* 
   * *Tích hợp Redis để cache phản hồi OpenAI.*
   * *Thêm rate limiting cho API (/chat, /upload-audio) để tránh lạm dụng.*
   * *Loại bỏ rejectUnauthorized: false trong config.json và sử dụng chứng chỉ SSL hợp lệ.*
6. ***Mở rộng chức năng học tập****:* 
   * *Thêm bài kiểm tra tiếng Anh với câu hỏi từ OpenAI.*
   * *Tích hợp video/hình ảnh minh họa từ Google Cloud Vision API.*
   * *Hỗ trợ học từ vựng qua flashcard hoặc trò chơi.*
7. ***Triển khai trên cloud****:* 
   * *Sử dụng AWS/GCP với autoscaling để xử lý nhiều người dùng.*
   * *Tích hợp CI/CD với GitHub Actions để tự động triển khai.*