**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**Project Proposal**

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

**MỤC LỤC**

[Các nội dung chính 1](#_Toc200360567)

[1. Bảng đánh giá thành viên 2](#_Toc200360568)

[2. Phát biểu bài toán sơ lược 3](#_Toc200360569)

[1.1 Giới thiệu chung dự án 3](#_Toc200360570)

[1.2 Môi trường hoạt động 3](#_Toc200360571)

[3. Giải pháp đề xuất 4](#_Toc200360572)

[1.3 Phần mềm 4](#_Toc200360573)

[1.3.1 Danh sách các chức năng phần mềm 4](#_Toc200360574)

[1.3.2 Kiến trúc tổng thể phần mềm 4](#_Toc200360575)

[1.3.3 Phần cứng 6](#_Toc200360576)

[4.1. Kế hoạch phát triển 7](#_Toc200360577)

[1.1 Phân tích yêu cầu 7](#_Toc200360578)

[1.2 Thiết kế phần mềm 8](#_Toc200360579)

[1.2.1. Kiến trúc tổng thể 8](#_Toc200360580)

[1.3 Cài đặt (implement) phần mềm 8](#_Toc200360581)

[1.4 Kiểm thử phần mềm 8](#_Toc200360582)

[1.5 Triển khai, bảo trì 8](#_Toc200360583)

[2 Kế hoạch nhân sự & chi phí 9](#_Toc200360584)

**Project Proposal**

# Các nội dung chính

Mục tiêu tài liệu tập trung vào các chủ đề:

* Tạo ra tài liệu Project Proposal.
* Hoàn chỉnh tài liệu Project Proposal với các nội dung:

Hiển thị dữ liệu phức tạp

* DataGridView
  + Phát biểu bài toán sơ lược
  + Giải pháp đề xuất
  + Kế hoạch phát triển
  + Kế hoạch nhân sự & chi phí
* Đọc hiểu tài liệu Project Proposal.

# Bảng đánh giá thành viên

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ Tên** | **% đóng góp (tối đa 100%)** | **Chữ ký** |
| 23880262 | Nguyễn Thanh Sang | 100% |  |
| 23880249 | Trịnh Tiến Long | 100% |  |
| 23880204 | Nguyễn Thanh Bảo | 100% |  |

# Phát biểu bài toán sơ lược

## Giới thiệu chung dự án

Ứng dụng “LearnSpeak” là nền tảng web hỗ trợ người dùng luyện tập tiếng Anh qua chat với AI, chuyển giọng nói thành văn bản (speech-to-text) khi người dùng nói, và dùng giọng nói đọc câu trả lời AI (text-to-speech). Mục tiêu là xây dựng môi trường học tập tương tác đa phương thức, tăng khả năng phản xạ ngôn ngữ, phù hợp với mọi đối tượng từ tự học đến luyện thi.

## Môi trường hoạt động

* **Frontend**:
  + Framework: React.js (JSX, hooks – React 16.8+) trên trình duyệt hiện đại (Chrome, Firefox, Edge…) hỗ trợ HTML5 & CSS3.
  + Thư viện hỗ trợ giọng nói: sử dụng Web Speech API (SpeechRecognition, SpeechSynthesis) hoặc thư viện như react-speech-recognition [npmjs.com](https://www.npmjs.com/package/react-speech-recognition?utm_source=chatgpt.com).
* **Backend**:
  + Chạy trên Node.js (≥v14) với Express.js để phục vụ REST API.
  + Xử lý logic chat: nhận tin nhắn từ frontend qua endpoint /api/message, gửi đến mô hình AI Gemini (có thể GPT, hoặc tự build đơn giản), sau đó trả về phản hồi.
* **Lưu trữ dữ liệu**:
  + Dùng file JSON (ví dụ /data/users.json, /data/messages.json) làm “database”, đọc/ghi bằng Node.js fs module.
  + Dữ liệu lưu gồm: người dùng, lịch sử chat, phản hồi AI.
* **Triển khai**:
  + Local dev: frontend chạy localhost:3000, backend localhost:5000.
  + Có thể deploy trên VPS hoặc cloud chỉ với Node.js, không cần DB server phức tạp.

# Giải pháp đề xuất

## Phần mềm

### Danh sách các chức năng phần mềm

|  |  |
| --- | --- |
| **Nhu cầu** | **Yêu cầu** |
| Là người học tiếng Anh, tôi muốn giao tiếp với AI qua cửa sổ chat để luyện phản xạ ngôn ngữ. | Hệ thống phải cung cấp giao diện chat với AI, cho phép người dùng gửi tin nhắn và nhận phản hồi. |
| Là người học tiếng Anh, tôi muốn chuyển giọng nói thành văn bản để nhập vào ô chat mà không cần gõ. | Hệ thống phải tích hợp chức năng speech-to-text, sử dụng Web Speech API hoặc thư viện như react-speech-recognition. |
| Là người học tiếng Anh, tôi muốn nghe câu trả lời của AI được đọc bằng giọng nói để luyện nghe. | Hệ thống phải tích hợp chức năng text-to-speech, sử dụng Web Speech API hoặc thư viện như react-speakup. |
| Là người học tiếng Anh, tôi muốn hệ thống lưu lại lịch sử chat của tôi để xem lại sau này. | |  | | --- | |  |  |  | | --- | | Hệ thống phải lưu trữ lịch sử chat của người dùng trong file JSON, bao gồm: tên người dùng, thời gian, nội dung tin nhắn. | |
| Là người học tiếng Anh, tôi muốn giao diện chat có thể tự động cuộn xuống khi có tin nhắn mới. | Hệ thống phải cung cấp tính năng tự động cuộn danh sách tin nhắn xuống dưới cùng khi có tin nhắn mới. |
| Là người học tiếng Anh, tôi muốn hệ thống ghi nhớ trạng thái đăng nhập của tôi để không phải đăng nhập lại mỗi lần truy cập. | Hệ thống phải sử dụng JSON Web Token (JWT) để xác thực người dùng và lưu trữ token trong localStorage hoặc sessionStorage của trình duyệt. |
| Là người học tiếng Anh, tôi muốn hệ thống tự động đăng xuất khi token hết hạn hoặc khi tôi chủ động đăng xuất. | Hệ thống phải kiểm tra thời gian hết hạn của token và tự động đăng xuất người dùng khi token hết hạn. |
| Là người học tiếng Anh, tôi muốn giao tiếp trực tiếp với AI bằng giọng nói mà không cần thao tác nhập hoặc gửi tin nhắn, để tăng phản xạ giao tiếp tự nhiên. | Hệ thống phải cho phép người dùng nói trực tiếp với AI (giao tiếp hai chiều bằng giọng nói) thông qua microphone và speaker mà không cần gõ hay bấm gửi. Hệ thống cần tích hợp song song chức năng **speech-to-text** và **text-to-speech** để tạo trải nghiệm hội thoại liên tục. |
| Là người học tiếng Anh, tôi muốn hệ thống tự động lưu lại các từ mới mà tôi chưa biết để tôi có thể ôn tập sau. | Hệ thống phải:   * Cho phép người dùng đánh dấu các từ là “mới” hoặc “cần học” trong nội dung hội thoại. * Lưu trữ các từ mới đã đánh dấu vào một file JSON hoặc trong cơ sở dữ liệu (gắn với tài khoản người dùng). * Hiển thị danh sách từ mới đã lưu trong một giao diện riêng để người dùng ôn tập. |

### Kiến trúc tổng thể phần mềm

Dưới đây là mô tả kiến trúc tổng thể của hệ thống phần mềm nhằm đáp ứng các yêu cầu chức năng đã được liệt kê.

* **Giao Diện Người Dùng (Frontend)**
* **Công nghệ sử dụng**: React.js
* **Chức năng**:
  + Cung cấp giao diện chat với AI.
  + Cho phép người dùng gửi tin nhắn và nhận phản hồi.
  + Tích hợp chức năng speech-to-text để nhập văn bản bằng giọng nói.
  + Tích hợp chức năng text-to-speech để đọc câu trả lời của AI.
  + Tích hợp tính năng trò chuyện trực tiếp với AI.
  + Lưu từ vựng mới để user có thể dễ dàng tham khảo.
  + Tự động cuộn xuống khi có tin nhắn mới.
  + Hiển thị lịch sử chat từ file JSON.
* **Backend**
* **Công nghệ sử dụng**: Node.js với Express
* **Chức năng**:
  + Xử lý các yêu cầu từ frontend.
  + Tích hợp API cho chức năng chat với AI (Gemini API).
  + Lưu trữ lịch sử chat vào file JSON.
  + Xác thực người dùng bằng JSON Web Token (JWT).
  + Kiểm tra và quản lý thời gian hết hạn của token.
  + Lưu trữ và quản lý các từ vựng mới.
* **Tích Hợp API**
* **Speech-to-Text**:
  + Sử dụng Web Speech API hoặc thư viện **react-speech-recognition** để chuyển giọng nói thành văn bản.
* **Text-to-Speech**:
  + Sử dụng Web Speech API hoặc thư viện **react-speakup** để đọc câu trả lời của AI.
* **Lưu Trữ Dữ Liệu**
* **Hệ thống lưu trữ**:
  + Sử dụng file JSON để lưu trữ lịch sử chat, bao gồm:
    - Tên người dùng
    - Thời gian
    - Nội dung tin nhắn
    - Từ vựng mới
* **Duy trì đăng nhập:**
* **Xác thực**:
  + Sử dụng JWT để xác thực người dùng.
  + Lưu trữ token trong **localStorage** hoặc **sessionStorage** của trình duyệt.
* **Đăng xuất**:
  + Tự động đăng xuất khi token hết hạn hoặc khi người dùng chủ động đăng xuất.
* **Tính Năng Tự Động Cuộn**
* **Công nghệ sử dụng**: React Hooks
* **Chức năng**:
  + Sử dụng **useRef** và **scrollTo** trong component chat để tự động cuộn xuống khi có tin nhắn mới.
* **Bảo Mật và Xử Lý Lỗi:**
* **Xử lý lỗi**:
  + Sử dụng **async/await** để xử lý bất đồng bộ và quản lý lỗi rõ ràng.

### Phần cứng

Dưới đây là mô tả các yêu cầu về máy móc và thiết bị mà phần mềm cần để hoạt động hiệu quả:

* **Máy Tính (PC hoặc Laptop)**
* **Cấu hình tối thiểu**:
  + **Bộ vi xử lý**: Intel Core i3 hoặc tương đương
  + **RAM**: 8GB
  + **Ổ cứng**: 256GB SSD hoặc HDD
  + **Card đồ họa**: Tích hợp (hoặc card rời cho hiệu suất tốt hơn)
* **Thiết Bị Âm Thanh**
* **Microphone**:
  + Microphone chất lượng tốt để thu âm giọng nói rõ ràng.
  + Có thể sử dụng microphone tích hợp trên laptop hoặc microphone rời (USB hoặc XLR).
* **Loa hoặc Tai Nghe**:
  + Loa hoặc tai nghe có chất lượng âm thanh tốt để nghe phản hồi từ AI.
  + Tai nghe có mic tích hợp sẽ là lựa chọn tốt cho việc giao tiếp.
* **Kết Nối Internet**
* **Băng thông**:
  + Kết nối Internet ổn định với tốc độ tối thiểu 5 Mbps để đảm bảo trải nghiệm chat mượt mà.
* **Công nghệ**:
  + Có thể sử dụng Wi-Fi hoặc kết nối Ethernet để đảm bảo độ ổn định.
* **Phần Mềm Hỗ Trợ**
* **Trình duyệt web**:
  + Cần sử dụng trình duyệt web hiện đại (Chrome, Firefox, Edge, Safari) để đảm bảo tương thích với các tính năng của ứng dụng.

# Kế hoạch phát triển

## Phân tích yêu cầu

Hệ thống được thiết kế để phục vụ nhu cầu học tiếng Anh của người dùng bằng cách cung cấp một môi trường giao tiếp tự nhiên với AI thông qua cửa sổ chat. Với mục tiêu này, các yêu cầu chính được phân tích như sau:

* **Giao tiếp AI – người dùng:** Hệ thống phải cho phép người dùng gửi tin nhắn và nhận phản hồi từ AI một cách nhanh chóng, mượt mà, giúp người dùng luyện phản xạ ngôn ngữ hiệu quả.
* **Nhập liệu bằng giọng nói:** Để hỗ trợ tiện lợi hơn trong quá trình giao tiếp, hệ thống tích hợp chức năng chuyển đổi giọng nói thành văn bản (speech-to-text), giúp người dùng nhập liệu mà không cần gõ phím, nâng cao trải nghiệm người dùng.
* **Phát âm câu trả lời:** Hỗ trợ chức năng đọc lại câu trả lời của AI bằng giọng nói (text-to-speech), giúp người dùng luyện kỹ năng nghe và phát âm chuẩn xác.
* **Lưu trữ lịch sử chat:** Hệ thống cần lưu lại toàn bộ lịch sử hội thoại của người dùng, bao gồm tên người dùng, thời gian, và nội dung tin nhắn dưới dạng file JSON, nhằm giúp người dùng có thể xem lại và theo dõi tiến trình học tập.
* **Tự động cuộn chat:** Giao diện chat phải tự động cuộn xuống dưới khi có tin nhắn mới để người dùng luôn nhìn thấy nội dung cập nhật mới nhất mà không phải thao tác thủ công.
* **Xác thực người dùng và bảo mật:** Hệ thống cần quản lý trạng thái đăng nhập của người dùng bằng JWT, đảm bảo bảo mật và sự tiện lợi bằng cách lưu trữ token trong bộ nhớ trình duyệt và tự động đăng xuất khi token hết hạn hoặc theo yêu cầu của người dùng.
* **Lưu các từ vựng mới:** Trong quá trình hội thoại, người dùng có thể gặp các từ vựng mới hoặc chưa hiểu rõ. Hệ thống cần cho phép người dùng đánh dấu những từ này, sau đó tự động lưu trữ để người dùng có thể xem lại và ôn tập sau.

Phân tích yêu cầu này là cơ sở để xây dựng kiến trúc và các thành phần chức năng của hệ thống, đảm bảo vừa đáp ứng yêu cầu người dùng vừa đảm bảo tính hiệu quả, bảo mật và mở rộng trong tương lai.

## Thiết kế phần mềm

Để xây dựng hệ thống học tiếng Anh qua giao tiếp với AI đáp ứng toàn bộ các nhu cầu, phần mềm được thiết kế theo kiến trúc tinh gọn và dễ mở rộng, đảm bảo tối ưu trải nghiệm người dùng.

### Kiến trúc tổng thể

* Hệ thống được xây dựng theo kiến trúc Client-Server.
* **Frontend**: React.js chịu trách nhiệm giao diện và tương tác người dùng.
* **Backend**: Node.js với Express xử lý logic nghiệp vụ, xác thực, lưu trữ và tích hợp AI.
* Giao tiếp giữa client và server qua REST API sử dụng JSON làm định dạng trao đổi dữ liệu.
* Xác thực người dùng qua JWT (JSON Web Token)

### Các thành phần chính và chức năng

#### Giao diện người dùng (Frontend)

* **Trang Đăng Nhập (Login)**:
  + Form đăng nhập gồm các trường nhập liệu: tên đăng nhập (username), mật khẩu.
  + Nút chuyển sang trang đăng ký (Register).
  + Hiển thị lỗi trực quan khi nhập sai thông tin.
  + Thiết kế giao diện rất sạch sẽ, tối giản theo phong cách Default Design Guidelines: nền trắng (#ffffff), các thành phần có bo góc nhẹ (0.75rem), bóng đổ mờ tinh tế.
  + Font chữ tiêu đề lớn (>48px), đậm, dễ nhìn, chữ thường dùng màu xám trung tính (#6b7280).
  + Button có hiệu ứng hover nhẹ nhàng, chuyển màu trơn tru.
* **Trang Đăng Ký (Register)**:
  + Form đăng ký với các trường: tên người dùng, email, mật khẩu, xác nhận mật khẩu.
  + Kiểm tra hợp lệ đầu vào (email đúng định dạng, mật khẩu đủ mạnh, xác nhận mật khẩu khớp).
  + Hiển thị thông báo lỗi hoặc thành công rõ ràng.
  + Giao diện và trải nghiệm đồng nhất với trang Login, khoáng đạt, dễ thao tác.
* **Giao diện Chat với AI**:
  + Thành phần chat gồm:
    - Danh sách tin nhắn: hiển thị tin nhắn của người dùng và AI.
    - Danh sách các từ vựng mới đã được lưu.
    - Ô nhập liệu có hỗ trợ nhập bằng bàn phím và bằng giọng nói (speech-to-text).
    - Nút gửi tin nhắn và nút bật/tắt ghi âm.
  + Tự động cuộn xuống dưới khi có tin nhắn mới để giữ cho người dùng luôn nhìn thấy nội dung mới nhất.
  + Tích hợp tính năng đọc to câu trả lời của AI (text-to-speech).
  + Giao diện đáp ứng, rộng tối đa 1200px, căn giữa tr

#### Backend

* **Xác thực người dùng**:
  + Endpoint **/login** nhận thông tin đăng nhập, xác thực và trả về token JWT.
  + Endpoint **/register** tiếp nhận dữ liệu đăng ký, lưu thông tin người dùng hợp lệ.
  + Middleware xác thực JWT bảo vệ các API nhạy cảm như gửi/nhận tin nhắn, lấy lịch sử chat.
  + Tự động kiểm tra token hết hạn, trả về trạng thái yêu cầu đăng xuất.
* **Xử lý chức năng Chat**:
  + API nhận tin nhắn người dùng, gọi Gemini AI API nhận phản hồi.
  + Lưu lịch sử chat theo user vào file JSON, ghi đầy đủ tên người dùng, thời gian và nội dung tin nhắn.
* **Lưu trữ**:
  + Lịch sử chat, danh sách từ vựng được lưu trữ trên hệ thống file dạng JSON để thuận tiện đọc, ghi và phân tách theo từng người dùng.
  + Dữ liệu người dùng đăng ký cũng có thể lưu trong file hoặc mở rộng với database nhẹ nếu cần.

## Cài đặt (implement) phần mềm

* Tạo file **.env** ở thư mục gốc dự án với nội dung:

GEMINI\_API\_KEY=AIzaSyBPNS252sb4q88pmwMcAs0meU7Vqe5f6Js

* Mở terminal và chạy lệnh cài đặt dependencies:

npm install

* Chạy thử trên mode dev:

npm run dev

## Kiểm thử phần mềm

Kiểm thử phần mềm là một giai đoạn quan trọng trong quy trình phát triển, nhằm đảm bảo rằng hệ thống hoạt động đúng như mong đợi và đáp ứng các yêu cầu đã đề ra. Dưới đây là kế hoạch kiểm thử cho hệ thống học tiếng Anh qua giao tiếp với AI.

**Kiểm thử chức năng (Functional Testing)**

* **Kiểm thử đăng nhập và đăng ký**:
  + Xác minh rằng người dùng có thể đăng ký tài khoản mới với thông tin hợp lệ.
  + Kiểm tra thông báo lỗi khi nhập thông tin không hợp lệ (ví dụ: mật khẩu không khớp, user đã tồn tại,..).
  + Đảm bảo rằng người dùng có thể đăng nhập thành công với thông tin đã đăng ký.
* **Kiểm thử chức năng chat**:
  + Gửi tin nhắn từ người dùng và xác minh rằng tin nhắn được hiển thị trong chatbox.
  + Kiểm tra phản hồi từ AI và đảm bảo rằng nó được hiển thị đúng cách.
  + Xác minh rằng chức năng ghi âm và chuyển đổi giọng nói thành văn bản hoạt động chính xác.
  + Đảm bảo rằng chức năng đọc lại câu trả lời của AI hoạt động mượt mà.
* **• Giao tiếp trực tiếp với AI:** 
  + Hệ thống phải đảm bảo khả năng nghe – trả lời liên tục một cách mượt mà. Khi người dùng bật chế độ giao tiếp tự động, hệ thống sẽ bắt đầu lắng nghe, gửi nội dung đến AI và phát lại phản hồi bằng giọng nói.
  + Trong lúc AI đang nói, hệ thống phải tạm ngừng nhận giọng nói để tránh thu lại phản hồi. Sau khi AI nói xong, hệ thống tự động quay lại trạng thái lắng nghe, giúp người dùng luyện phản xạ ngôn ngữ liền mạch.
* **• Lưu lại các từ vựng mới:** 
  + Hệ thống phải hỗ trợ người dùng lưu lại những từ vựng mới phát sinh trong quá trình giao tiếp.
  + Khi người dùng chọn lưu một từ hoặc cụm từ, hệ thống sẽ ghi lại từ đó kèm theo nghĩa tiếng Việt (nếu có), thời điểm lưu và ngữ cảnh sử dụng vào một file JSON hoặc hệ thống lưu trữ nội bộ, giúp người dùng dễ dàng xem lại và ôn tập từ vựng.
* **Kiểm thử lưu trữ lịch sử chat**:
  + Kiểm tra xem lịch sử chat có được lưu trữ đúng cách trong file JSON.
  + Đảm bảo rằng người dùng có thể xem lại lịch sử chat khi cần.

**Kiểm thử hiệu suất (Performance Testing)**

* **Kiểm thử tải (Load Testing)**:
  + Đánh giá khả năng xử lý của hệ thống khi có nhiều người dùng cùng truy cập và gửi tin nhắn.
  + Kiểm tra thời gian phản hồi của hệ thống khi có nhiều yêu cầu đồng thời.
* **Kiểm thử tốc độ (Speed Testing)**:
  + Đo thời gian cần thiết để gửi và nhận tin nhắn từ AI.
  + Đánh giá tốc độ tải trang và thời gian khởi động ứng dụng.

**Kiểm thử bảo mật (Security Testing)**

* **Kiểm thử xác thực**:
  + Đảm bảo rằng chỉ những người dùng đã đăng nhập mới có thể truy cập vào các chức năng nhạy cảm.
* **Kiểm thử mã hóa**:
  + Đảm bảo rằng dữ liệu nhạy cảm (như mật khẩu) được mã hóa khi lưu trữ và truyền tải.

## Triển khai, bảo trì

# Kế hoạch nhân sự & chi phí

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Tên** | **Khối lượng công việc** | **Thời gian** |
| **23880262** | **Nguyễn Thanh Sang** | **-Tính năng speech to text.**  **-Tính năng text to speak.**  **-UI/UX cho ứng dụng**  **-Duy trì đăng nhập bằng token.**  **-API login.**  **-Tính năng trò chuyện với API** | **40-50 giờ** |
| **23880249** | **Trịnh Tiến Long** | **-API để trò chuyện với AI (Gemini)**  **-Backend core.**  **-API lưu dữ liệu trò chuyện của user.**  **-API lấy dữ liệu trò chuyện của user.**  **-Tối ưu code.** | **40-50 giờ** |
| **23880204** | **Nguyễn Thanh Bảo** | **-UI/UX cho register.**  **-API cho register.**  **-API từ vựng**  **-UI/UX cho phần từ điển**  **-Tính năng từ vựng.**  **-Tính năng dịch từ/câu.**  **-Căn chỉnh CSS cho toàn ứng dụng.** | **40-50 giờ** |