

TRƯỜNG ĐẠI HỌC TRÀ VINH  
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ  
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO MÔN HỌC  
CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI  
XÂY DỰNG ỨNG DỤNG WEB  
ĐĂNG KÝ HỌC NGOẠI NGỮ

Giảng viên hướng dẫn

**TS. Nguyễn Bảo Ân**

Sinh viên thực hiện:

**Nguyễn Duy Tín (110122182 – DA22TTA)**

**Nguyễn Phước Hiệp (110122005 – DA22TTA)**

**Kiều Gia Thịnh (110122167 – DA22TTC)**

*Vĩnh Long, tháng 07 năm 2025*

## LỜI CẢM ƠN

Trước hết, nhóm em xin gửi lời cảm ơn chân thành đến Thầy Nguyễn Bảo Ân, người đã trực tiếp hướng dẫn, giúp đỡ và giải đáp thắc mắc trong suốt quá trình thực hiện đề tài. Sự tận tình của Thầy đã giúp nhóm em hoàn thành bài báo cáo này một cách tốt nhất.

Nhóm cũng xin bày tỏ lòng biết ơn đến các Thầy, Cô trong Khoa Công nghệ thông tin, đã cung cấp những kiến thức nền tảng cần thiết trong suốt thời gian học tập và nghiên cứu.

Do lượng kiến thức của nhóm còn hạn hẹp, nên không tránh khỏi những sai sót trong bài báo cáo của mình, kính mong quý Thầy/Cô thông cảm! Nhóm em cũng rất mong nhận được sự nhận xét của quý Thầy/Cô để nhóm em có thể khắc phục những sai sót cũng như phát triển hơn nữa trong tương lai!

*Sinh viên ký và ghi rõ họ và tên*

**Sinh viên 1**

**Sinh viên 2**

**Sinh viên 3**

[illegible]

## This image shows a full page of a document template designed for handwritten notes or essays. It features approximately 28 evenly spaced, thin grey horizontal lines extending across the entire width of the page. The margins are consistent on all sides, providing ample space for writing. There are no pre-printed questions, headings, or other markings on the page.

# MỤC LỤC

|   | Trang |
|---|-------|
| CHƯƠNG 1: GIỚI THIỆU .....  | 1     |
| 1.1    Tổng quan về công nghệ phần mềm .....                            | 1     |
| 1.1.1    Các khái niệm cơ bản về công nghệ phần mềm .....               | 1     |
| 1.1.2    Các mô hình phát triển phần mềm .....                          | 1     |
| 1.1.3    Tổng quan về client/server .....                               | 1     |
| 1.2    Tên dự án và chủ đề .....  | 2     |
| 1.3    Lí do chọn đề tài .....  | 2     |
| 1.4    Mục tiêu của ứng dụng .....                                      | 3     |
| CHƯƠNG 2: PHÂN TÍCH YÊU CẦU .....                                       | 4     |
| 2.1    Các chức năng chính của hệ thống (Functional Requirements) ..... | 4     |
| 2.2    Các yêu cầu phi chức năng (Non-functional Requirements) .....    | 4     |
| CHƯƠNG 3: THIẾT KẾ HỆ THỐNG .....                                       | 6     |
| 3.1    Kiến trúc tổng thể .....   | 6     |
| 3.2    Cấu trúc thư mục .....   | 6     |
| 3.3    Thiết kế cơ sở dữ liệu .....                                     | 7     |
| 3.4    Thiết kế API .....   | 8     |
| 3.5    Thiết kế giao diện (UI/UX) .....                                 | 11    |
| CHƯƠNG 4: TRIỂN KHAI VÀ CÔNG NGHỆ SỬ DỤNG .....                         | 16    |
| 4.1    Các công nghệ đã sử dụng .....                                   | 16    |
| 4.2    Quy trình CI/CD với GitHub Actions .....                         | 17    |
| 4.3    Cấu hình Docker và quy trình triển khai ứng dụng .....           | 19    |
| CHƯƠNG 5: QUẢN LÝ DỰ ÁN .....   | 23    |
| 5.1    Sử dụng Jira để lập kế hoạch và theo dõi tiến độ .....           | 23    |
| 5.2    Phân công nhiệm vụ .....   | 23    |
| CHƯƠNG 6: KIỂM THỬ .....  | 27    |
| 6.1    Chiến lược kiểm thử và công cụ sử dụng .....                     | 27    |
| 6.2    Kết quả kiểm thử API với Postman .....                           | 27    |
| CHƯƠNG 7: ĐÁNH GIÁ VÀ KẾT LUẬN .....                                    | 36    |
| TÀI LIỆU THAM KHẢO .....  | 38    |
| PHỤ LỤC .....   | 39    |

| DANH MỤC TỪ VIẾT TẮT |                                   |
|----------------------|-----------------------------------|
| Từ viết tắt          | Từ gốc                            |
| API                  | Application Programming Interface |
| CI                   | Continuous Integration            |
| CD                   | Continuous Deployment             |
| SE                   | Software Engineering              |
| UI                   | User Interface                    |
| UX                   | User Experience                   |

## DANH MỤC HÌNH ẢNH

|  |    |
|--|----|
| Hình 1: Kiến trúc tổng thể .....                   | 6  |
| Hình 2: Cấu trúc thư mục .....                     | 7  |
| Hình 3: Cơ sở dữ liệu .....                        | 8  |
| Hình 4: Bản vẽ giao diện trên Figma.....           | 11 |
| Hình 5: Giao diện trang chủ (chưa đăng nhập) ..... | 12 |
| Hình 6: Giao diện xem chi tiết một khóa học.....   | 13 |
| Hình 7: Giao diện trang chủ (đã đăng nhập) .....   | 13 |
| Hình 8: Giao diện xem tất cả khóa học .....        | 14 |
| Hình 9: Giao diện xem khóa học đã đăng ký .....    | 14 |
| Hình 10: Giao diện xem thông tin tài khoản .....   | 15 |
| Hình 11: Giao diện trang quản trị.....             | 15 |

---

## CHƯƠNG 1: GIỚI THIỆU

### 1.1 Tổng quan về công nghệ phần mềm

#### 1.1.1 Các khái niệm cơ bản về công nghệ phần mềm

Công nghệ phần mềm (Software Engineering) bao gồm tất cả các pha của vòng đời phát triển phần mềm, bao gồm phân tích, thiết kế, phát triển, thử nghiệm, tích hợp, triển khai, bảo trì và thậm chí là khai tử phần mềm. [1]

SE là việc áp dụng một cách tiếp cận có hệ thống, có kỷ luật, có thể định lượng để phát triển, vận hành và bảo trì phần mềm; nghĩa là ứng dụng kỹ thuật vào phần mềm.

Hai phần của SE: Phần mềm (Software) và Kỹ thuật (Engineering):

- Phần mềm (S) là tập hợp các mã, tài liệu và trình kích hoạt thực hiện một công việc cụ thể và đáp ứng một yêu cầu cụ thể.
- Kỹ thuật (E) là sự phát triển của các sản phẩm bằng cách sử dụng các phương pháp, nguyên tắc và phương pháp tốt nhất.

#### 1.1.2 Các mô hình phát triển phần mềm

Có nhiều mô hình phát triển phần mềm khác nhau, mỗi mô hình có những ưu nhược điểm riêng và phù hợp với các loại dự án khác nhau. Một số mô hình phổ biến như sau:

- Mô hình thác nước - Waterfall model
- Mô hình Agile
- Mô hình Scrum
- Mô hình xoắn ốc - Spiral model
- Mô hình tiếp cận lặp - Iterative model
- Mô hình tăng trưởng - Incremental model
- Mô hình chữ V
- RAD model (Rapid Application Development) [2]

#### 1.1.3 Tổng quan về client/server



**Client (máy khách):** Thường là các trình duyệt web (Chrome, Firefox, Safari, v.v.) chạy trên thiết bị của người dùng. Nó gửi các yêu cầu đến server để truy cập nội dung.

**Server (máy chủ):** Là nơi lưu trữ website, xử lý logic, truy xuất cơ sở dữ liệu và phản hồi lại client bằng HTML/CSS/JS hoặc dữ liệu (JSON/XML).

**Ưu điểm:**

- Tách biệt giao diện và xử lý: Client xử lý giao diện (frontend), Server xử lý logic và dữ liệu (backend).
- Tập trung dữ liệu: Dữ liệu được quản lý tập trung tại server, dễ backup, cập nhật, phân quyền.
- Khả năng mở rộng: Có thể mở rộng server (tăng cấu hình, dùng load balancer) để phục vụ nhiều người dùng hơn.
- Bảo mật tốt hơn: Server có thể kiểm soát truy cập, xác thực người dùng, và mã hóa dữ liệu.

**Nhược điểm:**

- Phụ thuộc vào server: Nếu server lỗi hoặc bị tấn công, website sẽ không thể truy cập được.
- Yêu cầu kỹ thuật cao: Việc phát triển backend và triển khai server đòi hỏi kiến thức lập trình và quản trị hệ thống. [3]

## 1.2 Tên dự án và chủ đề

Tên dự án: Ứng dụng web đăng ký học ngoại ngữ.

Chủ đề: Phát triển hệ thống thông tin phục vụ cho việc quản lý và đăng ký khóa học ngoại ngữ tại trung tâm ngoại ngữ.

## 1.3 Lý do chọn đề tài

Hiện nay, nhu cầu học ngoại ngữ ngày càng gia tăng, đặc biệt là tại các thành phố lớn và là điều kiện cần để xin được việc làm. Tuy nhiên, một số trung tâm ngoại ngữ vẫn còn áp dụng hình thức đăng ký học truyền thống như ghi danh trực tiếp, quản lý bằng giấy tờ, ... – gây nhiều bất tiện và dễ sai sót.

Chính vì vậy, nhóm lựa chọn đề tài “Xây dựng ứng dụng web đăng ký học ngoại ngữ” nhằm giải quyết các vấn đề tồn tại trong thực tế, đồng thời áp dụng kiến thức đã học về phát triển hệ thống thông tin, công nghệ phần mềm, lập trình web, quản lý cơ sở dữ liệu,... vào một bài toán có ý nghĩa thực tiễn cao.

#### **1.4 Mục tiêu của ứng dụng**

Ứng dụng web đăng ký học ngoại ngữ được xây dựng nhằm phục vụ nhu cầu quản lý thông tin khóa học, học viên, giáo viên, ngôn ngữ giảng dạy và quá trình đăng ký học của học viên một cách tiện lợi, nhanh chóng và chính xác. Ứng dụng hướng tới các mục tiêu cụ thể như sau:

Cung cấp cho học viên khả năng xem thông tin các khóa học và đăng ký học trực tuyến.

Hỗ trợ quản trị trung tâm dễ dàng quản lý khóa học, học viên, giáo viên, ngôn ngữ, trình độ, lịch học,...

Tạo giao diện thân thiện với người dùng, dễ sử dụng, phù hợp với cả học viên và nhân viên trung tâm.

Hạn chế các sai sót và giảm tải công việc thủ công trong quá trình quản lý đăng ký học.

Tối ưu hóa quy trình xử lý thông tin và cập nhật dữ liệu.

## CHƯƠNG 2: PHÂN TÍCH YÊU CẦU

### 2.1 Các chức năng chính của hệ thống (Functional Requirements)

Hệ thống được chia thành 2 nhóm người dùng chính: Học viên và Quản trị viên (Admin). Các chức năng chính cho từng nhóm người dùng bao gồm:

#### **Đối với học viên:**

Chức năng đăng ký, đăng nhập, quên mật khẩu.

Chức năng cập nhật thông tin cá nhân.

Chức năng xem giao diện trang chủ.

Chức năng xem tất cả các khóa học.

Chức năng xem chi tiết 1 khóa học.

Chức năng đăng ký khóa học.

Chức năng xem khóa học đã đăng ký.

Chức năng hủy đăng ký khóa học.

#### **Đối với quản trị viên (Admin):**

Chức năng đăng ký, đăng nhập và truy cập vào trang quản trị, đồng thời có tất cả các chức năng mà học viên có.

Chức năng quản lý người dùng (Thêm/Xóa/Sửa).

Chức năng quản lý giảng viên (Thêm/Xóa/Sửa).

Chức năng quản lý ngôn ngữ (Thêm/Xóa/Sửa).

Chức năng quản lý trình độ ngôn ngữ (Thêm/Xóa/Sửa).

Chức năng quản lý khóa học (Thêm/Xóa/Sửa).

Chức năng quản lý khóa học đã đăng ký (Thêm/Xóa/Sửa).

### 2.2 Các yêu cầu phi chức năng (Non-functional Requirements)

#### **Tính dễ sử dụng (Usability)**

- Giao diện đơn giản, rõ ràng, dễ thao tác cho cả người dùng phổ thông và admin.
- Hỗ trợ toàn bộ bằng tiếng Việt, bố cục dễ hiểu.

### **Tính bảo mật (Security)**

- Tài khoản học viên và admin được bảo vệ bằng mật khẩu.
- Admin có quyền cao hơn học viên và không thể bị học viên truy cập trái phép.
- Dữ liệu người dùng được bảo vệ, không để lộ thông tin cá nhân.

### **Khả năng mở rộng (Scalability)**

- Hệ thống được xây dựng có khả năng mở rộng để tích hợp thêm chức năng như thanh toán học phí, gửi email xác nhận, thêm lịch học chi tiết...

### **Khả năng bảo trì (Maintainability)**

- Mã nguồn rõ ràng, tách biệt từng phân chức năng, dễ bảo trì hoặc nâng cấp trong tương lai.

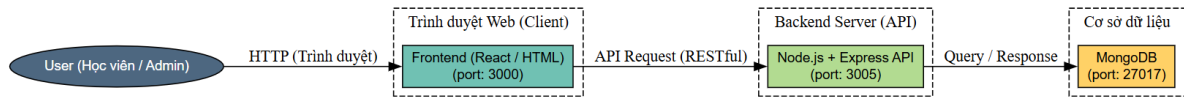
### **Hiệu năng (Performance)**

- Hệ thống phản hồi nhanh với các tác vụ trong môi trường thử nghiệm.

## CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

### 3.1 Kiến trúc tổng thể

Hệ thống được thiết kế theo mô hình Client – Server, gồm ba thành phần chính: giao diện người dùng (frontend), máy chủ API (backend) và cơ sở dữ liệu (MongoDB).



Hình 1: Kiến trúc tổng thể

Luồng hoạt động của hệ thống:

Người dùng truy cập hệ thống thông qua trình duyệt và gửi yêu cầu HTTP đến giao diện frontend tại cổng 3000.

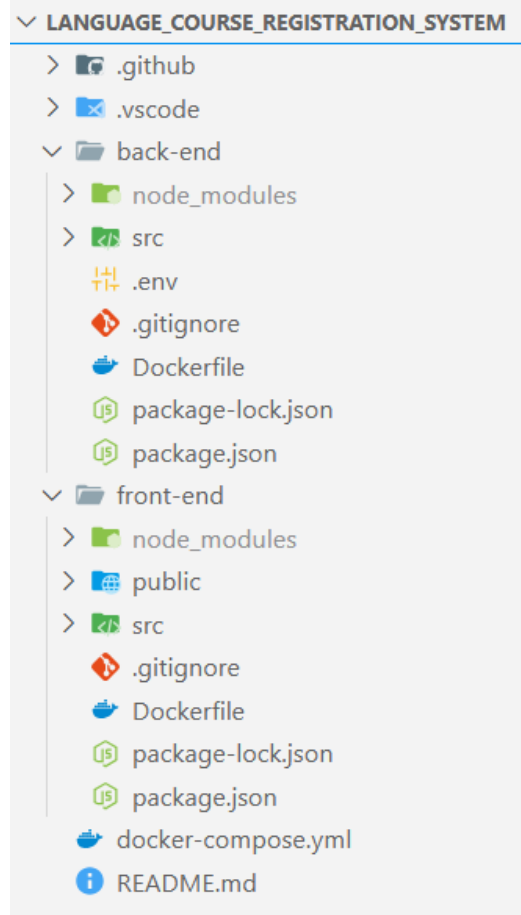
Giao diện frontend xử lý tương tác và gửi yêu cầu API đến backend qua cổng 3005.

Backend xử lý logic nghiệp vụ, xác thực quyền truy cập, và tương tác với MongoDB để truy vấn hoặc cập nhật dữ liệu.

Sau khi có dữ liệu từ MongoDB, backend gửi phản hồi về cho frontend, sau đó frontend hiển thị thông tin cho người dùng.

### 3.2 Cấu trúc thư mục

Cấu trúc thư mục được xây dựng như sau:



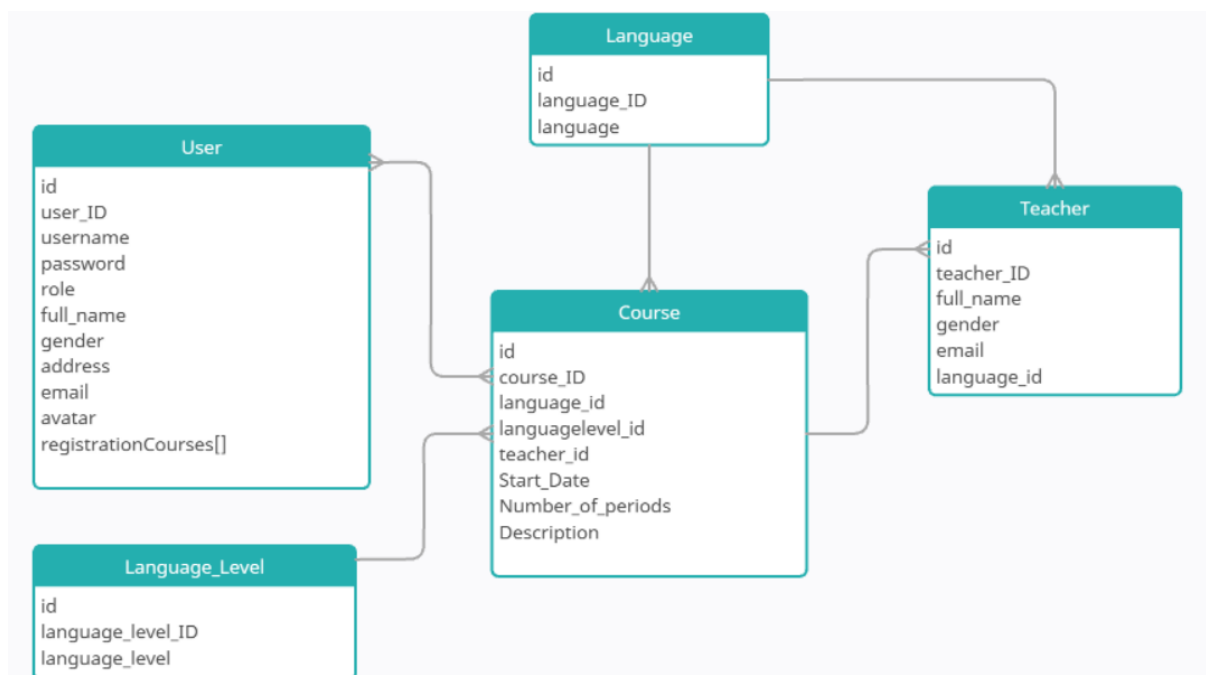
Hình 2: Cấu trúc thư mục

Thư mục backend/ chứa toàn bộ mã nguồn phía server, nơi xử lý logic nghiệp vụ, kết nối cơ sở dữ liệu, xác thực người dùng, phân quyền và cung cấp các API cho frontend gọi đến. Thư mục backend/src/ gồm các thư mục như models/ để định nghĩa dữ liệu (MongoDB), controllers/ để xử lý request, routes/ để khai báo các endpoint và services/ để xử lý logic nghiệp vụ. Backend đóng vai trò là trung tâm xử lý và giao tiếp với frontend thông qua RESTful API.

Thư mục frontend/ là nơi xây dựng giao diện người dùng bằng công nghệ React. Đây là ứng dụng client-side, nơi người dùng tương tác trực tiếp thông qua trình duyệt, thực hiện các hành động như đăng nhập, đăng ký khóa học,... Frontend gửi request tới backend qua HTTP (dùng axios), nhận dữ liệu JSON và hiển thị ra màn hình. Toàn bộ điều hướng, xử lý trạng thái và logic hiển thị đều nằm trong phần này, giúp tách biệt hoàn toàn khỏi backend.

### 3.3 Thiết kế cơ sở dữ liệu

Nhóm xây dựng cơ sở dữ liệu để quản lý việc đăng ký khóa học ngoại ngữ gồm 5 bảng như sau:



Hình 3: Cơ sở dữ liệu

User: Quản lý người dùng là học viên và admin, với user\_ID chính là mã học viên được cấp cho mỗi học viên khi đăng ký tài khoản. Ngoài ra còn có trường registrationCourses[] là một mảng để lưu khóa học mà học viên đã đăng ký.

Language: Quản lý ngôn ngữ của trung tâm ngoại ngữ. Mỗi ngôn ngữ sẽ có một language\_id.

Teacher: Quản lý giảng viên giảng dạy. Trường language\_id lấy từ bảng Language cho biết giảng viên nào dạy ngôn ngữ nào.

Language\_Level: Quản lý trình độ ngôn ngữ. Mỗi trình độ ngôn ngữ có một language\_level\_id.

Course: Quản lý khóa học. Mỗi khóa học sẽ có các thành phần chính như mã khóa học (course\_id), mã ngôn ngữ (language\_id) lấy từ bảng Language, mã trình độ (language\_level\_id) lấy từ bảng Language\_Level và mã giảng viên (teacher\_id) lấy từ bảng Teacher.

### 3.4 Thiết kế API

Ứng dụng Web đăng ký học ngoại ngữ được xây dựng với các Endpoint chính như sau (tiền tố các Endpoint: <http://3005/api>):

**Authentication:**

| Method | Endpoint           | Mô tả                 |
|--------|--------------------|-----------------------|
| POST   | /api/auth/login    | Đăng nhập             |
| POST   | /api/auth/register | Đăng ký tài khoản mới |
| GET    | /api/auth/logout   | Đăng xuất             |

**Người dùng:**

| Method | Endpoint           | Mô tả                         |
|--------|--------------------|-------------------------------|
| GET    | /api/user          | Lấy tất cả người dùng (Admin) |
| PUT    | /api/user/:id      | Cập nhật thông tin người dùng |
| DELETE | /api/user/multiple | Xóa nhiều người dùng          |

**Khóa học:**

| Method | Endpoint        | Mô tả                         |
|--------|-----------------|-------------------------------|
| GET    | /api/course     | Lấy danh sách tất cả khóa học |
| POST   | /api/course     | Tạo mới khóa học              |
| PUT    | /api/course/:id | Cập nhật khóa học             |
| DELETE | /api/course/:id | Xóa khóa học                  |

**Ngôn ngữ:**

---



### Xây dựng ứng dụng web đăng ký học ngoại ngữ

| Method | Endpoint               | Mô tả                         |
|--------|------------------------|-------------------------------|
| GET    | /api/language          | Lấy danh sách tất cả ngôn ngữ |
| POST   | /api/language/add      | Thêm ngôn ngữ mới             |
| PUT    | /api/language/:id      | Cập nhật ngôn ngữ             |
| DELETE | /api/language/multiple | Xóa nhiều ngôn ngữ            |

### Trình độ ngôn ngữ:

| Method | Endpoint                    | Mô tả               |
|--------|-----------------------------|---------------------|
| GET    | /api/languagelevel          | Lấy tất cả trình độ |
| POST   | /api/languagelevel/add      | Thêm trình độ mới   |
| PUT    | /api/languagelevel/:id      | Cập nhật trình độ   |
| DELETE | /api/languagelevel/multiple | Xóa nhiều trình độ  |

### Giảng viên:

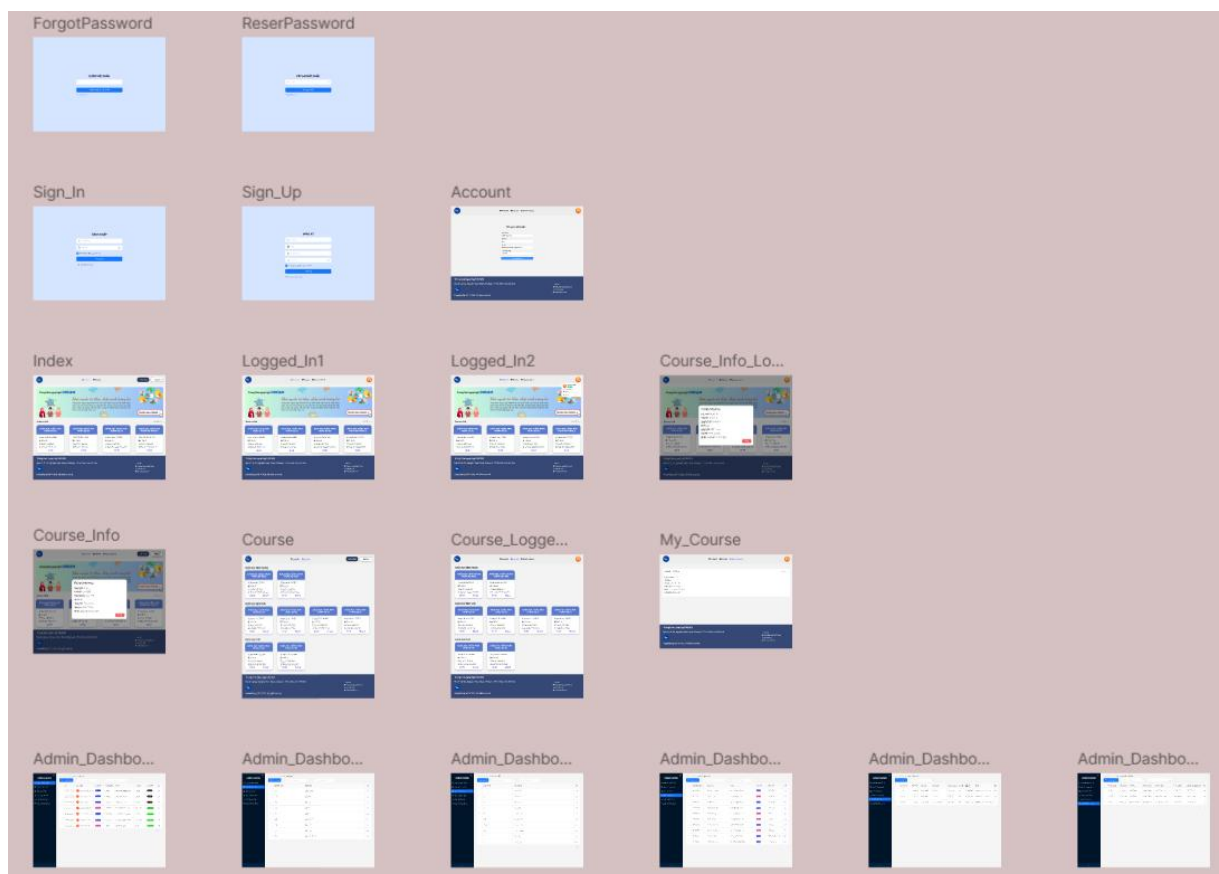
| Method | Endpoint              | Mô tả                           |
|--------|-----------------------|---------------------------------|
| GET    | /api/teacher          | Lấy danh sách tất cả giảng viên |
| POST   | /api/teacher          | Tạo mới giảng viên              |
| PUT    | /api/teacher/:id      | Cập nhật giảng viên             |
| DELETE | /api/teacher/multiple | Xóa nhiều giảng viên            |

### Đăng ký khóa học:

| Method | Endpoint  | Mô tả                                |
|--------|---|--------------------------------------|
| GET    | /api/user/:id/registered-courses                | Xem các khóa học học viên đã đăng ký |
| POST   | /api/user/:id/register-course                   | Đăng ký khóa học cho học viên        |
| PUT    | /api/user/update-registration/:userId/:courseId | Cập nhật thông tin đăng ký khóa học  |
| DELETE | /api/user/:id/unregister-course/:courseId       | Hủy đăng ký khóa học                 |

### 3.5 Thiết kế giao diện (UI/UX)

Nhóm đã thiết kế giao diện trên Figma như sau:



Hình 4: Bản vẽ giao diện trên Figma

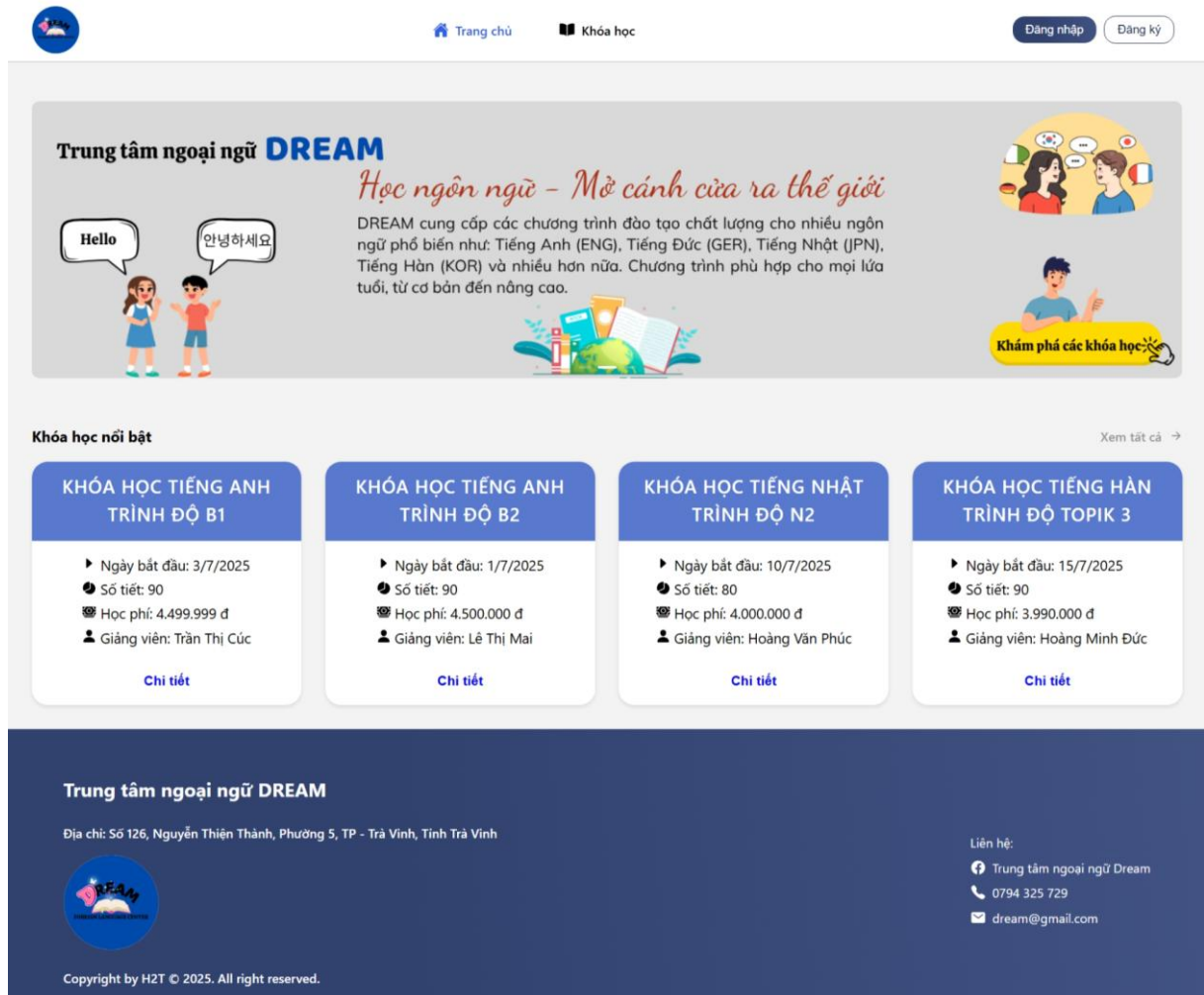
Ứng dụng web được thiết kế có tất cả 19 giao diện, bao gồm những giao diện liên quan đến đăng nhập/đăng ký, trang chủ, khóa học, trang quản trị,... đáp ứng tương đối

## Xây dựng ứng dụng web đăng ký học ngoại ngữ

đầy đủ chức năng của một ứng dụng web đăng ký khóa học ngoại ngữ với yêu cầu là học viên đăng ký tài khoản và khóa học, trang quản trị cho admin.

### Một số giao diện chính đã phát triển trên web

#### Giao diện trang chủ (chưa đăng nhập)

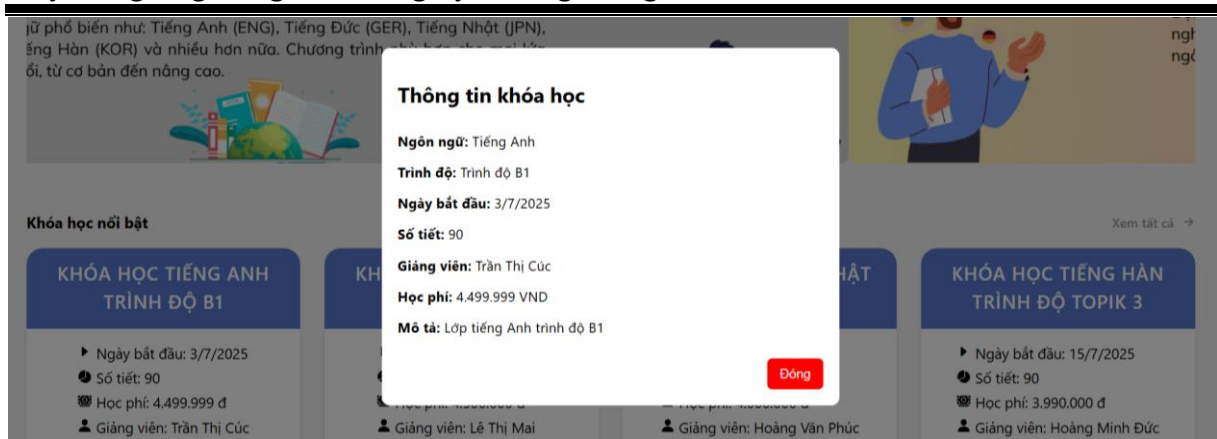


Hình 5: Giao diện trang chủ (chưa đăng nhập)

Giao diện này sẽ có một slide show giới thiệu về trung tâm ngoại ngữ, cho phép người dùng xem một vài khóa học nổi bật.

#### Giao diện xem chi tiết một khóa học

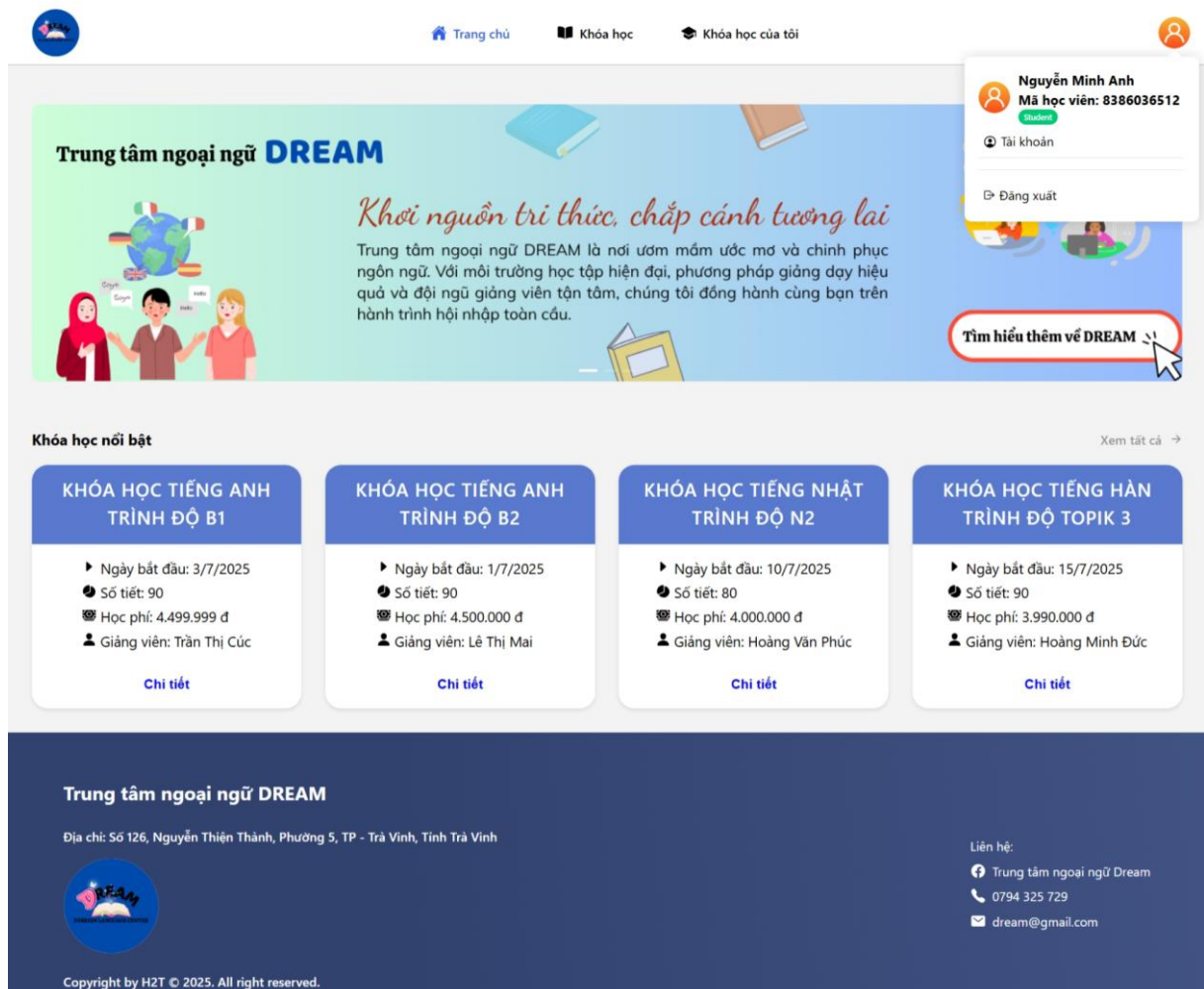
## Xây dựng ứng dụng web đăng ký học ngoại ngữ



Hình 6: Giao diện xem chi tiết một khóa học

Một khóa học sẽ có các thông tin như: ngôn ngữ, số tiết, giảng viên, học phí,...

Giao diện trang chủ (đã đăng nhập)



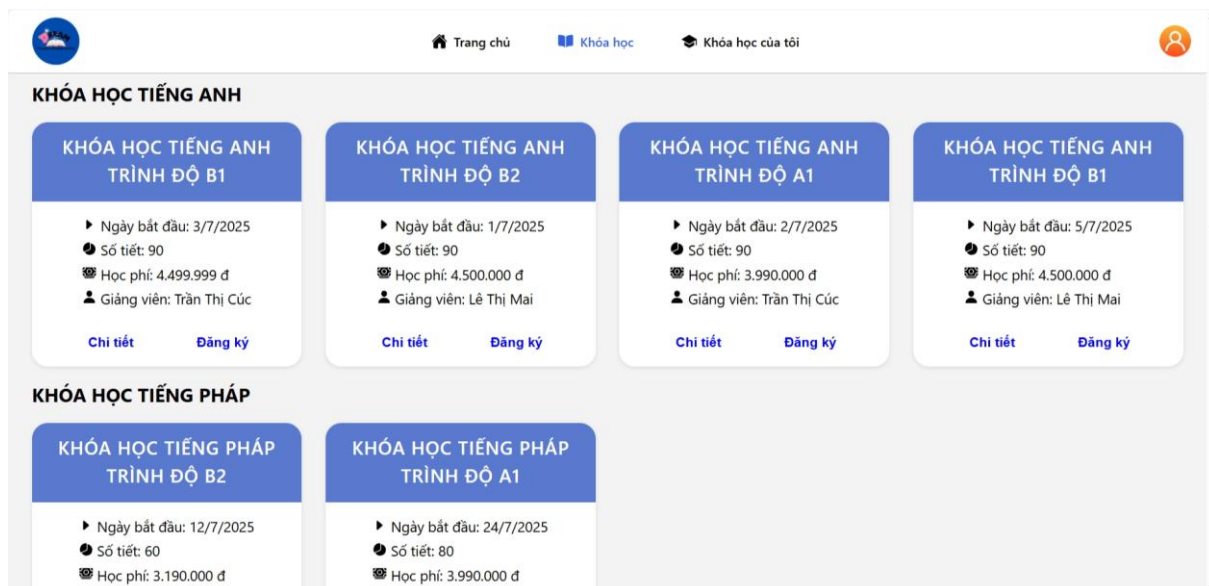
Hình 7: Giao diện trang chủ (đã đăng nhập)

Giao diện trang chủ khi đã đăng nhập tương tự như khi chưa đăng nhập nhưng có thêm một tab trên header là khóa học của tôi tức là khóa học mà học viên đã đăng ký,

## Xây dựng ứng dụng web đăng ký học ngoại ngữ

avatar popup hiển thị chức năng tùy theo phân quyền, nếu là học viên thì sẽ có mã học viên và chức năng xem thông tin tài khoản, còn nếu là admin thì sẽ có chức năng mở trang quản trị.

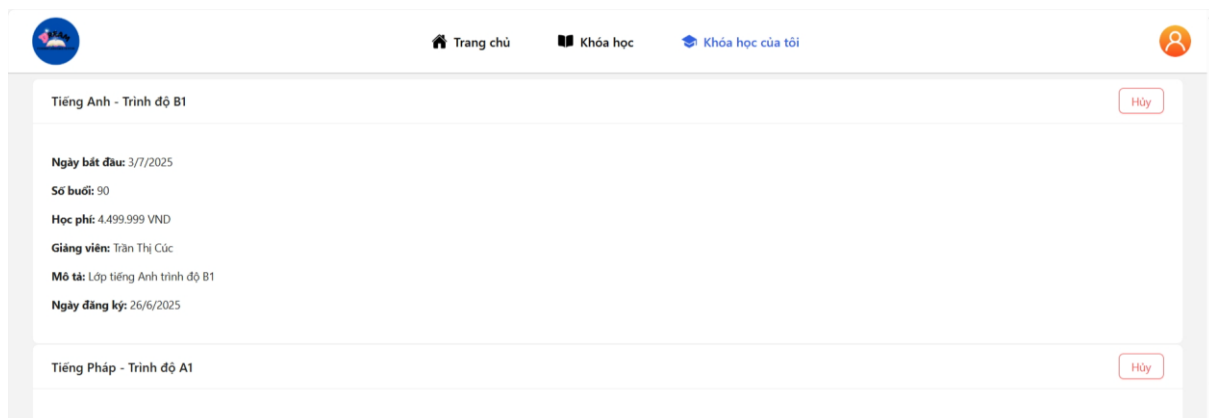
### Giao diện xem tất cả khóa học



Hình 8: Giao diện xem tất cả khóa học

Cho phép người dùng xem tất cả khóa học mà trung tâm mở và chức năng đăng ký khóa học nếu người dùng đã đăng nhập.

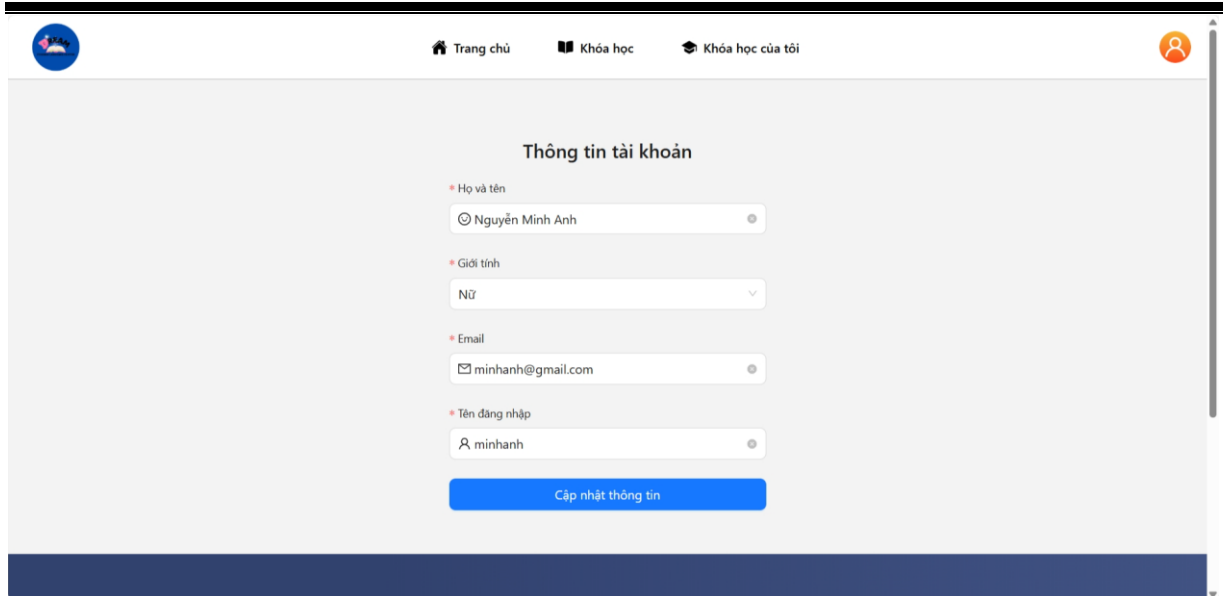
### Giao diện xem khóa học đã đăng ký



Hình 9: Giao diện xem khóa học đã đăng ký

Hiển thị tất cả các khóa học mà người dùng đã đăng ký và chức năng hủy đăng ký một khóa học bất kỳ.

### Giao diện xem thông tin tài khoản

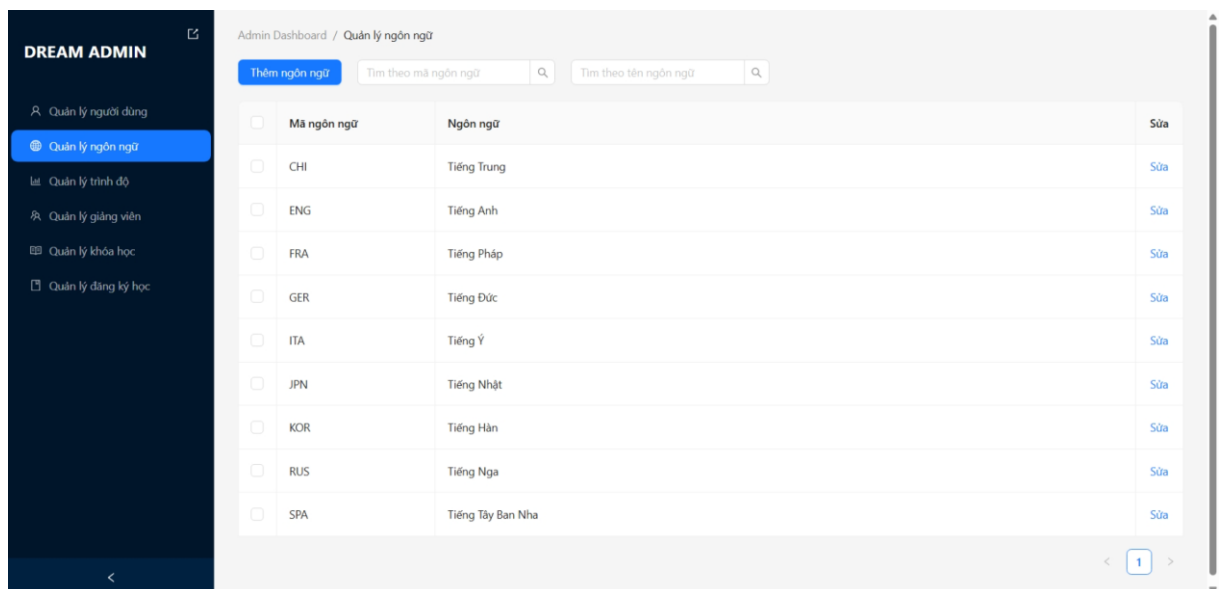


The screenshot shows a web interface for a language learning application. At the top, there is a navigation bar with a logo on the left and three links: 'Trang chủ', 'Khóa học', and 'Khóa học của tôi'. The main content area is titled 'Thông tin tài khoản'. It contains four input fields with red asterisks indicating required fields: 'Họ và tên' (filled with 'Nguyễn Minh Anh'), 'Giới tính' (a dropdown menu with 'Nữ' selected), 'Email' (filled with 'minhhanh@gmail.com'), and 'Tên đăng nhập' (filled with 'minhhanh'). Below these fields is a blue button labeled 'Cập nhật thông tin'.

Hình 10: Giao diện xem thông tin tài khoản

Hiển thị thông tin mà người dùng đã đăng ký, cho phép cập nhật lại thông tin nếu cần.

### Giao diện trang quản trị



The screenshot shows an admin dashboard titled 'Admin Dashboard / Quản lý ngôn ngữ'. On the left is a sidebar with the 'DREAM ADMIN' logo and a list of menu items: 'Quản lý người dùng', 'Quản lý ngôn ngữ' (highlighted), 'Quản lý trình độ', 'Quản lý giảng viên', 'Quản lý khóa học', and 'Quản lý đăng ký học'. The main content area has a header with a 'Thêm ngôn ngữ' button and two search filters: 'Tìm theo mã ngôn ngữ' and 'Tìm theo tên ngôn ngữ'. Below this is a table with three columns: 'Mã ngôn ngữ', 'Ngôn ngữ', and 'Sửa'. The table contains ten rows of data for different languages.

| <input type="checkbox"/> | Mã ngôn ngữ | Ngôn ngữ          | Sửa |
|--------------------------|-------------|-------------------|-----|
| <input type="checkbox"/> | CHI         | Tiếng Trung       | Sửa |
| <input type="checkbox"/> | ENG         | Tiếng Anh         | Sửa |
| <input type="checkbox"/> | FRA         | Tiếng Pháp        | Sửa |
| <input type="checkbox"/> | GER         | Tiếng Đức         | Sửa |
| <input type="checkbox"/> | ITA         | Tiếng Ý           | Sửa |
| <input type="checkbox"/> | JPN         | Tiếng Nhật        | Sửa |
| <input type="checkbox"/> | KOR         | Tiếng Hàn         | Sửa |
| <input type="checkbox"/> | RUS         | Tiếng Nga         | Sửa |
| <input type="checkbox"/> | SPA         | Tiếng Tây Ban Nha | Sửa |

Hình 11: Giao diện trang quản trị

Cho phép admin quản lý việc thêm/xóa/sửa người dùng, ngôn ngữ, trình độ, khóa học,...

## CHƯƠNG 4: TRIỂN KHAI VÀ CÔNG NGHỆ SỬ DỤNG

### 4.1 Các công nghệ đã sử dụng

#### Ngôn ngữ lập trình:

- JavaScript: Dùng cho frontend (React).
- Node.js (JavaScript): Dùng cho backend.
- HTML, CSS: Giao diện người dùng cơ bản.
- MongoDB Query Language (MQL): Truy vấn dữ liệu trong MongoDB.

#### Framework / Thư viện:

##### Frontend

- ReactJS: Xây dựng giao diện người dùng.
- Ant Design (antd): Thư viện UI component dùng cho bảng, modal, form, button,...
- Axios: Thư viện gọi API HTTP từ frontend.

##### Backend

- Express.js: Framework backend dùng với Node.js.
- Mongoose: ORM để tương tác với MongoDB.

##### Khác

- JWT (JSON Web Token): Xác thực người dùng.
- Swagger: Tài liệu hóa API.

#### Cơ sở dữ liệu:

MongoDB: NoSQL Database dùng để lưu trữ thông tin người dùng, khóa học, giảng viên,...

#### Kiểm thử:

Swagger: mô tả, thiết kế và trực quan hóa tài liệu API.

Postman: kiểm thử API.

GitHub Actions: kiểm thử khi có thay đổi mã nguồn.

## Triển khai:

Docker: đóng gói và triển khai ứng dụng.

## 4.2 Quy trình CI/CD với GitHub Actions

### Kiểm thử Backend (CI)

```
name: Test Back-end

on:
  push:
    branches: [develop]
    paths:
      - 'back-end/**'
  pull_request:
    branches: [develop]
    paths:
      - 'back-end/**'

jobs:
  test-backend:
    runs-on: ubuntu-latest

    defaults:
      run:
        working-directory: back-end

    steps:
      - name: Lấy mã nguồn
        uses: actions/checkout@v3

      - name: Cài Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '18'

      - name: Cài dependencies
        run: npm install

      - name: Chạy test
        run: npm test -- --passWithNoTests
```

Quy trình:

- Kích hoạt khi có push hoặc pull request vào nhánh develop, với các thay đổi trong thư mục back-end/.
- Thiết lập môi trường Node.js 18.
- Cài đặt dependency bằng npm install.



- Chạy lệnh kiểm thử với npm test, cho phép không có test cũng không lỗi (--passWithNoTests).

### Kiểm thử Frontend React (CI)

```
name: Test React Frontend

on:
  push:
    branches: [front-end]
    paths:
      - 'front-end/**'
  pull_request:
    branches: [front-end]
    paths:
      - 'front-end/**'

jobs:
  test:
    name: Run React (CRA) Tests
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v3

      - name: Setup Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '18'

      - name: Install dependencies
        run: npm install
        working-directory: ./front-end

      - name: Run tests
        run: npm test -- --watchAll=false
        working-directory: ./front-end
```

#### Quy trình:

- Kích hoạt khi có push hoặc pull request vào nhánh front-end, với các thay đổi trong thư mục front-end/.
- Checkout mã nguồn và cài Node.js 18.
- Chạy npm install trong thư mục ./front-end để cài các thư viện React.
- Chạy npm test với --watchAll=false để thực hiện kiểm thử giao diện.

## 4.3 Cấu hình Docker và quy trình triển khai ứng dụng

### Cấu hình Docker file ở Frontend

```
FROM node:18

WORKDIR /app

COPY package*.json ./
RUN npm install

COPY . .

EXPOSE 3000
CMD ["npm", "start"]
```

**FROM node:18:** Dùng image chính thức node:18 làm môi trường nền.

**WORKDIR /app:** Đặt thư mục làm việc trong container là /app. Các lệnh như COPY, RUN sẽ được thực thi trong thư mục này.

**COPY package\*.json .:** Copy package.json và package-lock.json vào thư mục /app trong container.

**RUN npm install:** Cài đặt tất cả dependencies được khai báo trong package.json.

**COPY . . :** Copy toàn bộ mã nguồn dự án từ máy vào trong container.

**EXPOSE 3000:** Mở cổng 3000 trong container.

**CMD ["npm", "start"]:** Chạy ứng dụng frontend

### Cấu hình Docker file ở Backend

```
FROM node:18

WORKDIR /app

COPY package*.json ./
RUN npm install

COPY . .

EXPOSE 3005

CMD ["node", "src/app.js"]
```

**FROM node:18:** Dùng image chính thức node:18 làm môi trường nền.

**WORKDIR /app:** Đặt thư mục làm việc trong container là /app. Các lệnh như COPY, RUN sẽ được thực thi trong thư mục này.

**COPY package\*.json ./:** Copy package.json và package-lock.json vào thư mục /app trong container.

**RUN npm install:** Cài đặt tất cả dependencies được khai báo trong package.json.

**COPY . . :** Copy toàn bộ mã nguồn dự án từ máy vào trong container.

**EXPOSE 3005:** Mở cổng 3005 trong container.

**CMD ["node", "src/app.js"]:** Chạy file chính của ứng dụng Node.js, là file app.js nằm trong thư mục src.

## Cấu hình Docker-compose

```
version: '3.9'

services:
  backend:
    build: ./back-end
    container_name: backend
    ports:
      - "3005:3005"
    volumes:
      - ./back-end:/app
      - /app/node_modules
    env_file:
      - ./back-end/.env
    depends_on:
      - mongo
    restart: always

  frontend:
    build: ./front-end
    container_name: frontend
    ports:
      - "3000:3000"
    volumes:
      - ./front-end:/app
      - /app/node_modules
    restart: always

  mongo:
    image: mongo:6
    container_name: mongo
    ports:
      - "27017:27017"
    volumes:
      - mongo-data:/data/db
    restart: always

volumes:
  mongo-data:
```

**version: '3.9':** Phiên bản của Docker Compose được sử dụng.

---

**services:**

**1. backend**

**build: ./back-end:** Docker sẽ build image từ thư mục ./back-end.

**container\_name: backend:** Đặt tên cho container là backend.

**ports: "3005:3005":** Mở cổng 3005 trên host và ánh xạ đến 3005 của container.

**volumes:**

**./back-end:/app:** Kết nối thư mục code ./back-end từ máy host vào /app trong container.

**/app/node\_modules:** Kết nối thư mục node\_modules bên trong container.

**env\_file: ./back-end/.env:** Nạp biến môi trường từ file .env.

**depends\_on: mongo:** Đảm bảo container mongo được khởi động trước.

**restart: always:** Container sẽ luôn được khởi động lại nếu bị lỗi hoặc sau khi máy chủ khởi động.

**2. frontend**

**build: ./front-end:** Build image từ thư mục ./front-end.

**container\_name: frontend:** Tên container là frontend.

**ports: "3000:3000":** Mở cổng 3000.

**volumes:**

**./front-end:/app:** Kết nối code frontend vào container.

**/app/node\_modules:** Bảo vệ thư mục node\_modules tương tự backend.

**restart: always:** Tự động restart khi có lỗi hoặc sau reboot.

**3. mongo**

**image: mongo:6:** Sử dụng image MongoDB phiên bản 6.

**container\_name: mongo:** Tên container là mongo.

**ports: "27017:27017":** Mở cổng MongoDB mặc định.

**volumes: mongo-data:/data/db:** Lưu trữ dữ liệu MongoDB vĩnh viễn trên volume tên mongo-data.

**restart: always:** Tự động restart khi cần thiết.

## CHƯƠNG 5: QUẢN LÝ DỰ ÁN

### 5.1 Sử dụng Jira để lập kế hoạch và theo dõi tiến độ

#### Lập kế hoạch Sprint và phân công công việc

Dự án được chia thành nhiều Sprint kéo dài 1–2 tuần, tương ứng với các giai đoạn nhỏ trong toàn bộ quy trình phát triển.

Trong mỗi Sprint, nhóm sử dụng Product Backlog để tập hợp danh sách các tính năng, yêu cầu và nhiệm vụ cần thực hiện.

Từ đó, Sprint Backlog được tạo ra để chọn các công việc ưu tiên và đưa vào thực hiện trong Sprint hiện tại.

Mỗi nhiệm vụ (task) được tạo thành một Issue trong Jira, là các Story – mô tả yêu cầu người dùng.

Các Issue được giao cho từng thành viên, kèm theo deadline và mức độ ưu tiên (Priority).

#### Theo dõi tiến độ

Nhóm sử dụng Board kiểu Scrum để trực quan hóa quy trình làm việc với các cột: To Do → In Progress → Done

Thành viên cập nhật trạng thái công việc khi bắt đầu và kết thúc 1 tính năng để mọi người đều nắm được tiến độ chung.

Jira tự động tạo Burn - down chart, Sprint report, giúp nhóm đánh giá hiệu suất làm việc và ước lượng tốt hơn cho các Sprint sau.

### 5.2 Phân công nhiệm vụ

Dự án web được hoàn thành trong 5 Sprint với mỗi nhiệm vụ phân công cho từng thành viên cụ thể như sau:

## Xây dựng ứng dụng web đăng ký học ngoại ngữ

| Summary  | Status | Comments   | Sprint         | Assignee       |
|--|--------|------------|----------------|----------------|
| Là một người dùng, tôi muốn đăng nhập và đăng xuất, nhờ đó tôi có thể b...   | DONE   | Add commen | SCRUM Sprint 1 | PH Phuoc Hiep  |
| Là một người dùng, tôi muốn xem giao diện trang chủ, nhờ đó tôi có thể d...  | DONE   | Add commen | SCRUM Sprint 1 | D DuyTinNguyen |
| Là một admin, tôi muốn đăng nhập vào hệ thống quản trị, nhờ đó tôi có th...  | DONE   | Add commen | SCRUM Sprint 1 | GT Gia Thinh   |
| Là một admin, tôi muốn có giao diện để thêm, sửa và xóa các khóa học, ...    | DONE   | Add commen | SCRUM Sprint 2 | D DuyTinNguyen |
| Là một người dùng, tôi muốn xem danh sách các khóa học ngoại ngữ, nh...      | DONE   | Add commen | SCRUM Sprint 2 | D DuyTinNguyen |
| Là một người dùng, tôi muốn xem chi tiết từng khóa học, nhờ đó tôi có thể... | DONE   | Add commen | SCRUM Sprint 2 | GT Gia Thinh   |
| Là một người dùng, tôi muốn đăng ký một khóa học, nhờ đó tôi có thể tha...   | DONE   | Add commen | SCRUM Sprint 2 | PH Phuoc Hiep  |
| Là một người dùng, tôi muốn xem danh sách các khóa học đã đăng ký, n...      | DONE   | Add commen | SCRUM Sprint 2 | PH Phuoc Hiep  |
| Là một người dùng, tôi muốn chỉnh sửa thông tin tài khoản của mình, nhờ ...  | DONE   | Add commen | SCRUM Sprint 2 | GT Gia Thinh   |

| Summary  | Status | Comments   | Sprint         | Assignee       |
|--|--------|------------|----------------|----------------|
| Là một người dùng, tôi muốn chỉnh sửa thông tin tài khoản của mình, nhờ ...  | DONE   | Add commen | SCRUM Sprint 2 | GT Gia Thinh   |
| Là một admin, tôi muốn có giao diện quản lý người dùng, để tôi có thể d...   | DONE   | Add commen | SCRUM Sprint 3 | PH Phuoc Hiep  |
| Là một admin, tôi muốn có giao diện quản lý ngôn ngữ, để tôi có thể d...     | DONE   | Add commen | SCRUM Sprint 3 | D DuyTinNguyen |
| Là một admin, tôi muốn có giao diện quản lý giảng viên, để tôi có thể d...   | DONE   | Add commen | SCRUM Sprint 4 | GT Gia Thinh   |
| Là một admin, tôi muốn có giao diện để thêm, sửa và xóa các khóa học, ...    | DONE   | Add commen | SCRUM Sprint 4 | D DuyTinNguyen |
| Tích hợp các biện pháp bảo mật cho hệ thống                                  | DONE   | Add commen | SCRUM Sprint 4 | PH Phuoc Hiep  |
| Là một admin, tôi muốn có giao diện quản lý trình độ ngôn ngữ, để tôi có ... | DONE   | Add commen | SCRUM Sprint 3 | GT Gia Thinh   |
| Thêm chức năng lấy lại mật khẩu khi quên mật khẩu                            | DONE   | Add commen | SCRUM Sprint 5 | GT Gia Thinh   |
| Tối ưu hóa hiệu suất của các API   | DONE   | Add commen | SCRUM Sprint 5 | PH Phuoc Hiep  |
| Kiểm thử tích hợp  | DONE   | Add commen | SCRUM Sprint 5 | D DuyTinNguyen |

Độ ưu tiên và Story Points được ước lượng như sau:

| User Story | Mô tả  | Story Points | Độ ưu tiên |
|------------|--|--------------|------------|
| Sprint 1   | Là một người dùng, tôi muốn đăng nhập và đăng xuất, nhờ đó tôi có thể bảo mật tài khoản.               | 5            | Cao        |
| Sprint 1   | Là một người dùng, tôi muốn xem giao diện trang chủ, nhờ đó tôi có thể dễ dàng tiếp cận các chức năng. | 2            | Cao        |
| Sprint 1   | Là một admin, tôi muốn đăng nhập vào hệ thống quản trị, nhờ đó tôi có thể quản lý nội dung.            | 3            | Cao        |

|          |   |   |            |
|----------|---|---|------------|
| Sprint 2 | Là một người dùng, tôi muốn xem danh sách các khóa học ngoại ngữ, nhờ đó tôi có thể chọn khóa học phù hợp để đăng ký.     | 5 | Cao        |
| Sprint 2 | Là một người dùng, tôi muốn xem chi tiết từng khóa học, nhờ đó tôi có thể biết nội dung, lịch học và giảng viên.          | 3 | Cao        |
| Sprint 2 | Là một người dùng, tôi muốn đăng ký một khóa học, nhờ đó tôi có thể tham gia học.   | 5 | Cao        |
| Sprint 2 | Là một người dùng, tôi muốn xem danh sách các khóa học đã đăng ký, nhờ đó tôi có thể quản lý việc học của mình.           | 3 | Trung bình |
| Sprint 2 | Là một người dùng, tôi muốn chỉnh sửa thông tin tài khoản của mình, nhờ đó tôi có thể cập nhật thông tin cá nhân.         | 3 | Trung bình |
| Sprint 2 | Là một admin, tôi muốn có giao diện để thêm, sửa và xóa các khóa học, để tôi có thể cập nhật nhanh chóng.                 | 5 | Cao        |
| Sprint 3 | Là một admin, tôi muốn có giao diện quản lý người dùng, để tôi có thể dễ dàng theo dõi và quản lý các tài khoản học viên. | 6 | Trung bình |
| Sprint 3 | Là một admin, tôi muốn có giao diện quản lý ngôn ngữ, để tôi có thể dễ dàng theo dõi và quản lý các ngôn ngữ.             | 5 | Cao        |
| Sprint 3 | Là một admin, tôi muốn có giao diện quản lý trình độ ngôn ngữ, để tôi có thể dễ dàng theo dõi và quản lý các trình độ.    | 5 | Trung bình |



|          |   |   |            |
|----------|---|---|------------|
| Sprint 4 | Tích hợp các biện pháp bảo mật cho hệ thống.  | 8 | Trung bình |
| Sprint 4 | Là một admin, tôi muốn có giao diện quản lý giảng viên, để tôi có thể dễ dàng theo dõi và quản lý các giảng viên. | 5 | Trung bình |
| Sprint 4 | Là một admin, tôi muốn có giao diện để thêm, sửa và xóa các khóa học, để tôi có thể cập nhật nhanh chóng.         | 5 | Trung bình |
| Sprint 5 | Thêm chức năng lấy lại mật khẩu khi quên mật khẩu.  | 5 | Trung bình |
| Sprint 5 | Tối ưu hóa hiệu suất các API.   | 7 | Cao        |
| Sprint 5 | Kiểm thử tích hợp.  | 7 | Cao        |

## CHƯƠNG 6: KIỂM THỬ

### 6.1 Chiến lược kiểm thử và công cụ sử dụng

Nhóm đã áp dụng chiến lược kiểm thử nhiều lớp để đảm bảo chất lượng phần mềm. Các API được kiểm thử bằng Postman, đảm bảo các chức năng backend hoạt động đúng. Ngoài ra, nhóm tích hợp kiểm thử tự động vào quy trình CI thông qua GitHub Actions, giúp kiểm tra mã nguồn tự động mỗi khi có thay đổi.

#### Kiểm thử API (Postman)

Mục tiêu: Đảm bảo các endpoint RESTful hoạt động đúng và trả về dữ liệu chính xác.

Quy trình:

- Sử dụng Postman để gửi các yêu cầu HTTP (GET, POST, PUT, DELETE).
- Kiểm tra phản hồi (response code, dữ liệu trả về, thông báo lỗi).

Áp dụng cho:

- Đăng ký, đăng nhập.
- Lấy danh sách khóa học, giảng viên,...
- Thêm/sửa/xóa các đối tượng trong hệ thống quản trị.

#### Kiểm thử tự động với GitHub Actions (CI)

Mục tiêu: Tự động kiểm tra mã nguồn mỗi khi có thay đổi, giúp phát hiện lỗi sớm trong quy trình phát triển.

Khi có push hoặc pull request lên bất kì nhánh nào, GitHub Actions sẽ:

- Tự động cài dependencies.
- Chạy lệnh kiểm thử (npm test) cho cả frontend và backend.
- Thông báo lỗi nếu có lỗi test xảy ra.

### 6.2 Kết quả kiểm thử API với Postman

#### Các API xác thực

## Xây dựng ứng dụng web đăng ký học ngoại ngữ

The image shows a Postman interface for a POST request to `http://localhost:3005/api/auth/login`. The request body is a JSON object with `username: "admin1"` and `password: "123"`. The response is a 200 OK status with a response time of 101 ms and a body size of 850 B. The response body is a JSON object containing a long JWT token.

```
POST http://localhost:3005/api/auth/login
```

```
{  "username": "admin1",  "password": "123"}
```

```
{  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjpbImlkIjoibG1Y2I5NmVhYzE5NmQ4OTMxMGnkNDJjIiwidXNlcm5hbWUiOiJhZG1pbjEiLCJyb2x1IjoiaWRTaw4ifSwiaWF0IjoxNzUxNTIxMDU5LCJleHAiOjE3NTE1NDI2NTI5LgT83uMrEK48uqk-rSmckAp0R9LGgfPE-uSt3DSabcgo"}
```

## Thử nghiệm API đăng nhập với Postman

The image shows a Postman interface for a POST request to `http://localhost:3005/api/auth/register`. The request body is a JSON object with `username: "user1"` and `password: "123"`. The response is a 201 Created status with a response time of 76 ms and a body size of 383 B. The response body is a JSON object containing a success message.

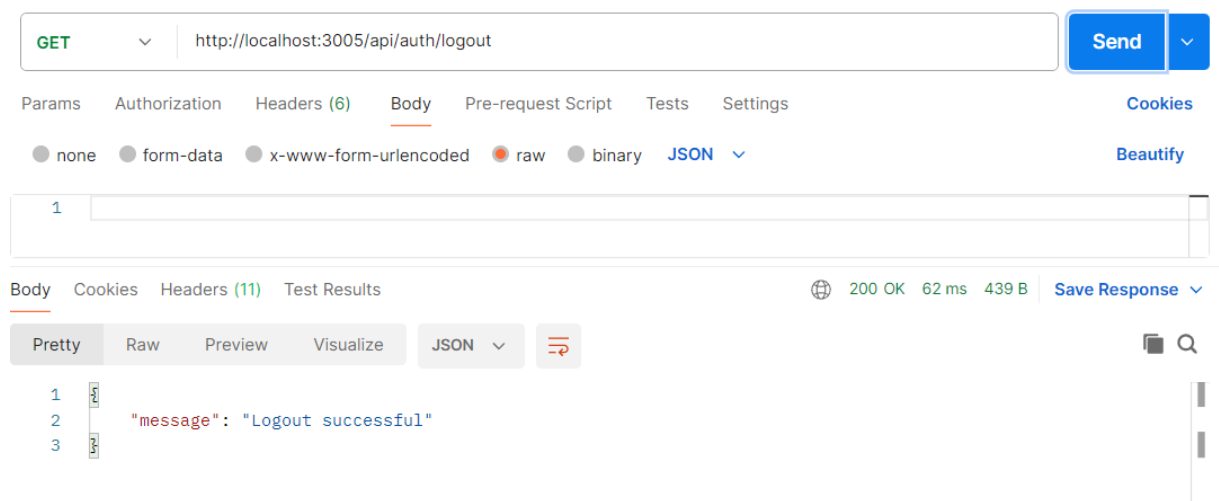
```
POST http://localhost:3005/api/auth/register
```

```
{  "username": "user1",  "password": "123"}
```

```
{  "message": "Registration successful"}
```

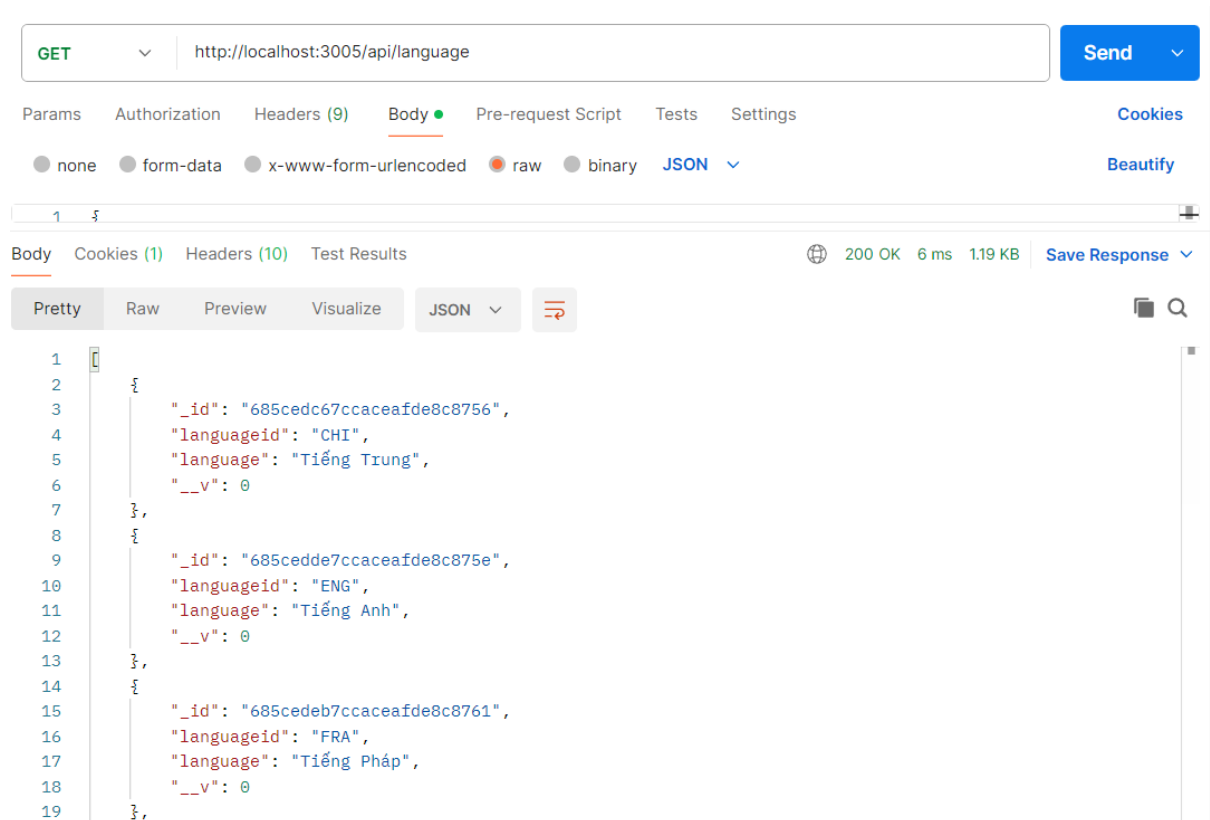
## Thử nghiệm API đăng ký tài khoản mới với Postman

## Xây dựng ứng dụng web đăng ký học ngoại ngữ



### Thử nghiệm API đăng xuất với Postman

### Các API liên quan đến ngôn ngữ



### Thử nghiệm API lấy ra danh sách các ngôn ngữ với Postman

## Xây dựng ứng dụng web đăng ký học ngoại ngữ

The screenshot shows a Postman interface for a POST request to `http://localhost:3005/api/language/add`. The request body is a JSON object: `{ "languageid": "BRA", "language": "Tiếng Brazil" }`. The response is a JSON object: `{ "message": "Language added successfully" }`. The status is 201 Created, with a response time of 7 ms and a size of 387 B.

```
POST http://localhost:3005/api/language/add

{
  "languageid": "BRA",
  "language": "Tiếng Brazil"
}
```

```
{
  "message": "Language added successfully"
}
```

Thử nghiệm API thêm ngôn ngữ với Postman

### Các API liên quan đến trình độ ngôn ngữ

The screenshot shows a Postman interface for a GET request to `http://localhost:3005/api/languagelevel`. The response is a JSON array of three objects, each representing a language level. The status is 200 OK, with a response time of 6 ms and a size of 1.98 KB.

```
GET http://localhost:3005/api/languagelevel
```

```
[
  {
    "_id": "685cee6a7ccaceafde8c877a",
    "language_levelid": "A1",
    "language_level": "Trình độ A1",
    "__v": 0
  },
  {
    "_id": "685cee717ccaceafde8c877d",
    "language_levelid": "A2",
    "language_level": "Trình độ A2",
    "__v": 0
  },
  {
    "_id": "685cee7b7ccaceafde8c8780",
    "language_levelid": "B1",
    "language_level": "Trình độ B1",
    "__v": 0
  }
]
```

Thử nghiệm API lấy ra danh sách trình độ ngôn ngữ với Postman

## Xây dựng ứng dụng web đăng ký học ngoại ngữ

The screenshot shows the Postman interface for a POST request to `http://localhost:3005/api/languagelevel/add`. The request body is a JSON object: `{ "language_levelid": "A3", "language_level": "Trình độ A3" }`. The response status is 201 Created, with a 6 ms response time and 393 B body size. The response body is a JSON object: `{ "message": "Language level added successfully" }`.

```
POST http://localhost:3005/api/languagelevel/add

{
  "language_levelid": "A3",
  "language_level": "Trình độ A3"
}
```

201 Created 6 ms 393 B Save Response

```
{
  "message": "Language level added successfully"
}
```

Thử nghiệm API thêm trình độ ngôn ngữ với Postman

### Các API liên quan đến giảng viên

The screenshot shows the Postman interface for a GET request to `http://localhost:3005/api/teacher`. The response status is 200 OK, with a 7 ms response time and 3.44 KB body size. The response body is a JSON array of two teacher objects.

```
GET http://localhost:3005/api/teacher
```

200 OK 7 ms 3.44 KB Save Response

```
[
  {
    "_id": "685cef607ccaceafde8c87ae",
    "teacherid": "GV096245",
    "full_name": "Nguyễn Văn Nam",
    "gender": "Nam",
    "email": "vannam@gmail.com",
    "language_id": {
      "_id": "685cedc67ccaceafde8c8756",
      "languageid": "CHI",
      "language": "Tiếng Trung",
      "__v": 0
    },
    "__v": 0
  },
  {
    "_id": "685cef7c7ccaceafde8c87b3",
    "teacherid": "GV042711",
    "full_name": "Trần Thị Cúc",
  }
]
```

Thử nghiệm API lấy ra danh sách giảng viên với Postman

## Xây dựng ứng dụng web đăng ký học ngoại ngữ

POST http://localhost:3005/api/teacher

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "teacherid": "GV096235",
3   "full_name": "Nguyễn Văn Tân",
4   "gender": "Nam",
5   "email": "vantan@gmail.com",
6 }
```

Body Cookies (1) Headers (10) Test Results 201 Created 9 ms 528 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "teacherid": "GV023992",
3   "full_name": "Nguyễn Văn Tân",
4   "gender": "Nam",
5   "email": "vantan@gmail.com",
6   "language_id": "685cedc67ccaceafde8c8756",
7   "_id": "68662652c2ff3929c4e31c99",
8   "__v": 0
9 }
```

Thử nghiệm API thêm giảng viên với Postman

### Các API liên quan đến khóa học

POST http://localhost:3005/api/course

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "courseid": "KH0151",
3   "language_id": "685cee2b7ccaceafde8c876d",
4   "languagelevel_id": "685ceeb57ccaceafde8c8789",
5   "teacher_id": "685cef7c7ccaceafde8c87b3",
6   "Start_Date": "2025-07-03",
7 }
```

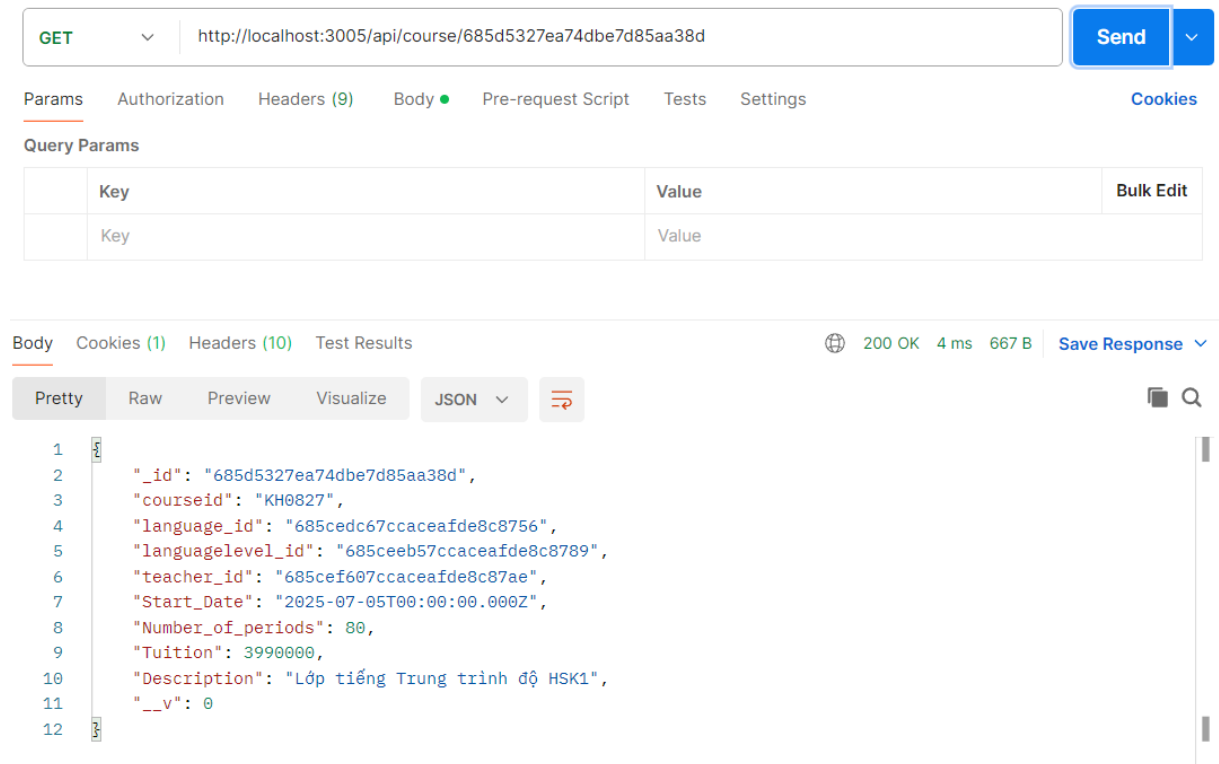
Body Cookies (1) Headers (10) Test Results 201 Created 8 ms 635 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "courseid": "KH0151",
3   "language_id": "685cee2b7ccaceafde8c876d",
4   "languagelevel_id": "685ceeb57ccaceafde8c8789",
5   "teacher_id": "685cef7c7ccaceafde8c87b3",
6   "Start_Date": "2025-07-03T00:00:00.000Z",
7   "Number_of_periods": 90,
8   "Tuition": 4499999,
9   "Description": "",
10  "_id": "686624c9c2ff3929c4e31c73",
11  "__v": 0
12 }
```

Thử nghiệm API thêm khóa học với Postman

## Xây dựng ứng dụng web đăng ký học ngoại ngữ

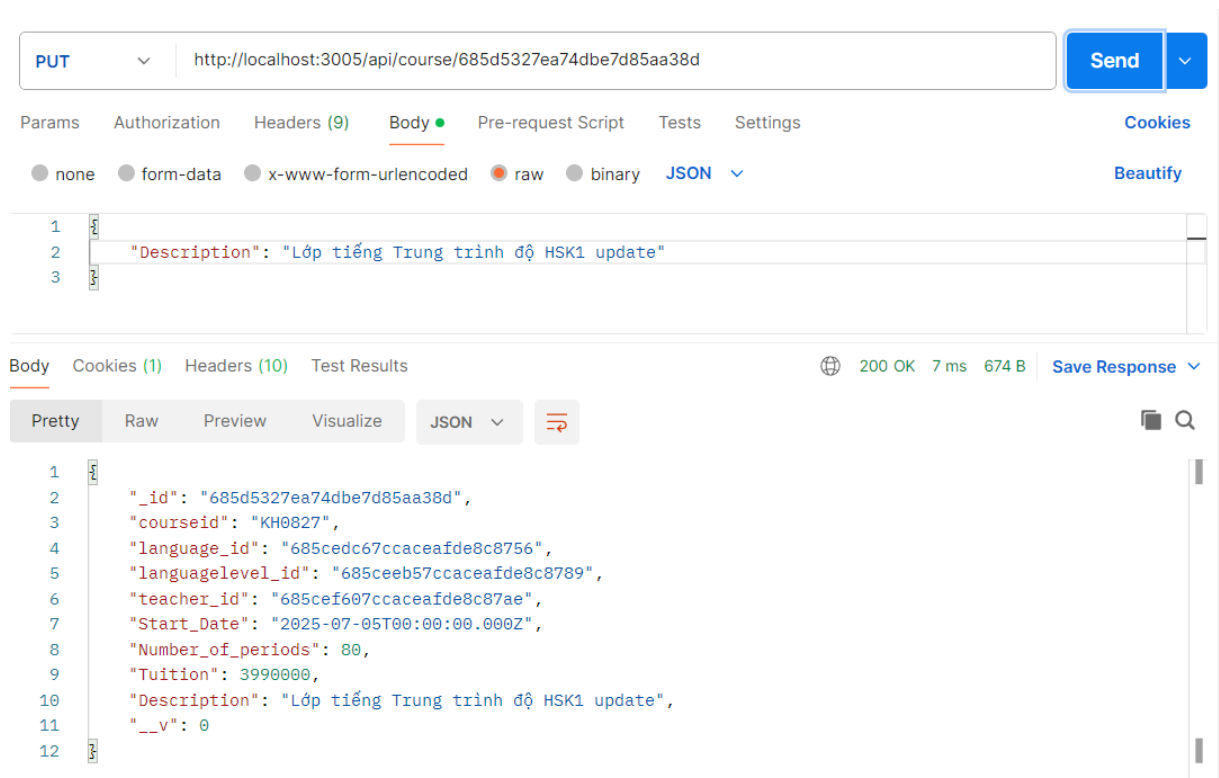


Postman interface showing a GET request to `http://localhost:3005/api/course/685d5327ea74dbe7d85aa38d`. The response is a JSON object with the following data:

| Key | Value |
|-----|-------|
| Key | Value |

```
1  {
2    "_id": "685d5327ea74dbe7d85aa38d",
3    "courseid": "KH0827",
4    "language_id": "685cedc67ccaceafde8c8756",
5    "languagelevel_id": "685ceeb57ccaceafde8c8789",
6    "teacher_id": "685cef607ccaceafde8c87ae",
7    "Start_Date": "2025-07-05T00:00:00.000Z",
8    "Number_of_periods": 80,
9    "Tuition": 3990000,
10   "Description": "Lớp tiếng Trung trình độ HSK1",
11   "__v": 0
12 }
```

## Thử nghiệm API lấy ra thông tin khóa học theo ID với Postman



Postman interface showing a PUT request to `http://localhost:3005/api/course/685d5327ea74dbe7d85aa38d`. The request body is a JSON object with the following data:

```
1  {
2    "Description": "Lớp tiếng Trung trình độ HSK1 update"
3  }
```

The response is a JSON object with the following data:

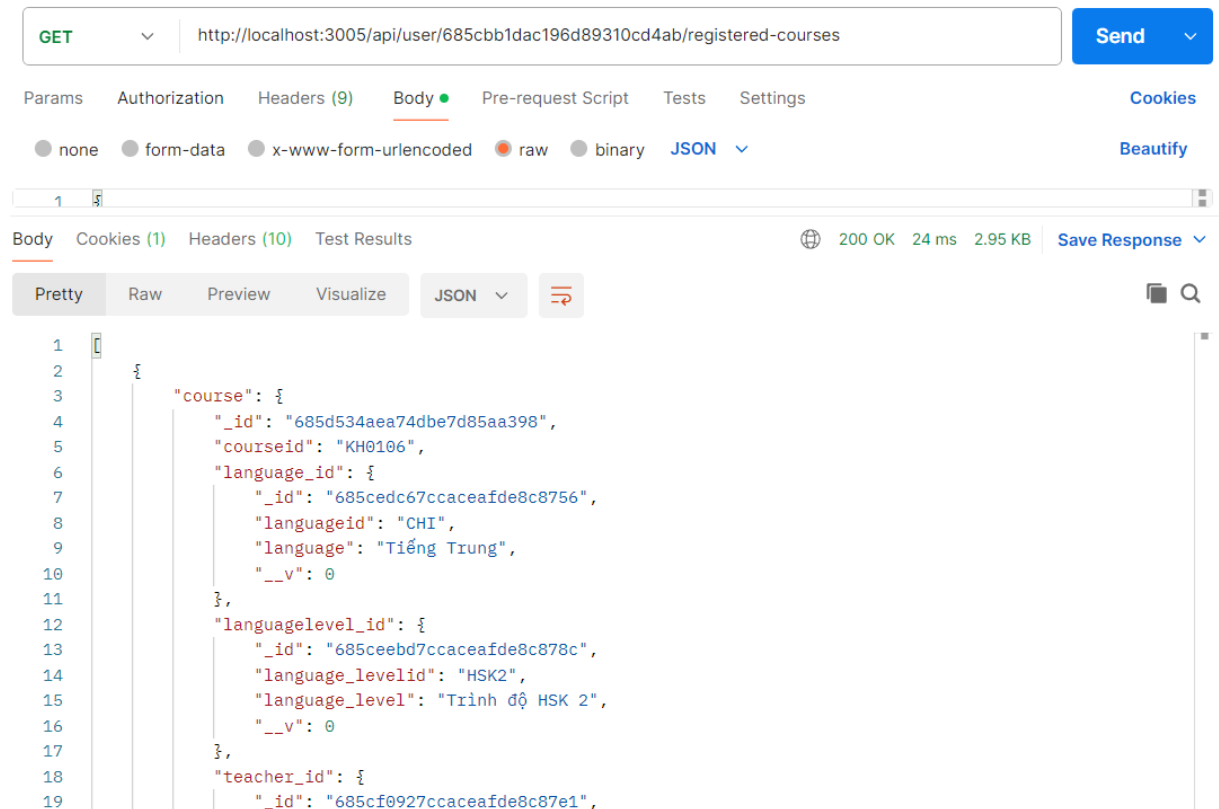
```
1  {
2    "_id": "685d5327ea74dbe7d85aa38d",
3    "courseid": "KH0827",
4    "language_id": "685cedc67ccaceafde8c8756",
5    "languagelevel_id": "685ceeb57ccaceafde8c8789",
6    "teacher_id": "685cef607ccaceafde8c87ae",
7    "Start_Date": "2025-07-05T00:00:00.000Z",
8    "Number_of_periods": 80,
9    "Tuition": 3990000,
10   "Description": "Lớp tiếng Trung trình độ HSK1 update",
11   "__v": 0
12 }
```

## Thử nghiệm API sửa khóa học với Postman

## Các API liên quan đến học viên



## Xây dựng ứng dụng web đăng ký học ngoại ngữ



GET http://localhost:3005/api/user/685cbb1dac196d89310cd4ab/registered-courses

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Beautify

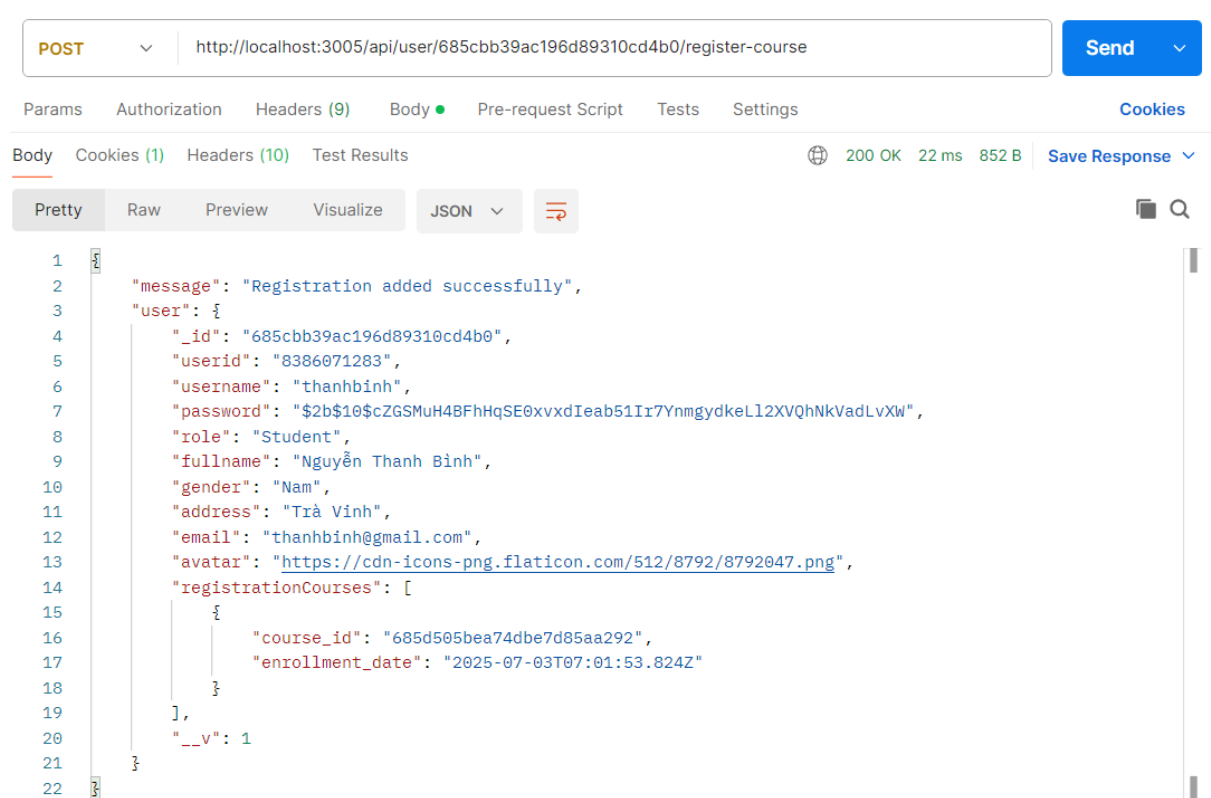
none form-data x-www-form-urlencoded raw binary JSON

Body Cookies (1) Headers (10) Test Results 200 OK 24 ms 2.95 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "course": {
3     "_id": "685d534aea74dbe7d85aa398",
4     "courseid": "KH0106",
5     "language_id": {
6       "_id": "685cedc67ccaceafde8c8756",
7       "languageid": "CHI",
8       "language": "Tiếng Trung",
9       "__v": 0
10    },
11    "languagelevel_id": {
12      "_id": "685cee7bd7ccaceafde8c878c",
13      "language_levelid": "HSK2",
14      "language_level": "Trình độ HSK 2",
15      "__v": 0
16    },
17    "teacher_id": {
18      "_id": "685cf0927ccaceafde8c87e1",
```

Thử nghiệm API lấy danh sách các khóa học mà học viên đã đăng ký theo ID với Postman



POST http://localhost:3005/api/user/685cbb39ac196d89310cd4b0/register-course

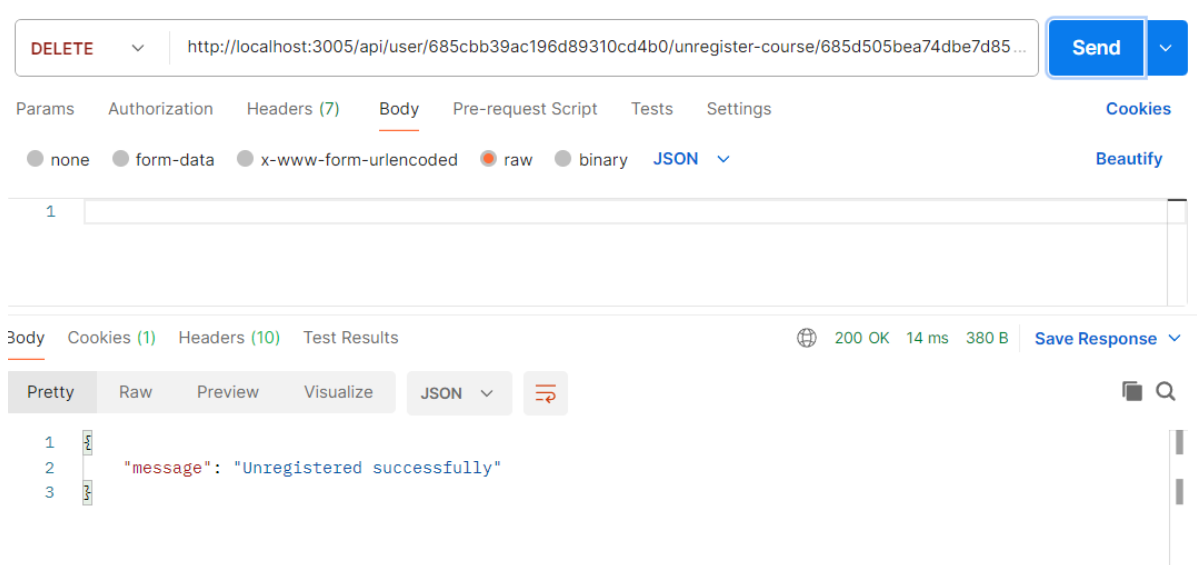
Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Save Response

Body Cookies (1) Headers (10) Test Results 200 OK 22 ms 852 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Registration added successfully",
3   "user": {
4     "_id": "685cbb39ac196d89310cd4b0",
5     "userid": "8386071283",
6     "username": "thanhbinh",
7     "password": "$2b$10$cZGSMuH4BFHqSE0xvxdIeab51Ir7YnmgydkeLl2XVQhNkVadLvXW",
8     "role": "Student",
9     "fullname": "Nguyễn Thanh Bình",
10    "gender": "Nam",
11    "address": "Trà Vinh",
12    "email": "thanhbinh@gmail.com",
13    "avatar": "https://cdn-icons-png.flaticon.com/512/8792/8792047.png",
14    "registrationCourses": [
15      {
16        "course_id": "685d505bea74dbe7d85aa292",
17        "enrollment_date": "2025-07-03T07:01:53.824Z"
18      }
19    ],
20    "__v": 1
21  }
22 }
```

Thử nghiệm API thêm đăng ký khóa học cho học viên với Postman



Thử nghiệm API xóa đăng ký khóa học cho học viên với Postman

## CHƯƠNG 7: ĐÁNH GIÁ VÀ KẾT LUẬN

### 7.1 Đánh giá

#### Kết quả đạt được

Xây dựng thành công ứng dụng web “Đăng ký học ngoại ngữ” đáp ứng tương đối đầy đủ các yêu cầu và chức năng của môn học công nghệ phần mềm như:

- Ứng dụng web áp dụng kiến trúc client/server, tách biệt rõ ràng giữa frontend và backend. Frontend và backend giao tiếp thông qua API.
- Quản lý dự án trên Jira.
- Thiết kế giao diện người dùng bằng Figma.
- Quản lý mã nguồn, phân nhánh và kiểm soát phiên bản bằng Github.
- Tự động hóa quy trình kiểm thử và triển khai (CI/CD) bằng Github Actions.
- Kiểm thử API bằng Postman.
- Đóng gói và triển khai ứng dụng trong môi trường container bằng Docker.
- Tạo tài liệu API tự động bằng Swagger.
- Áp dụng nguyên tắc REST để xây dựng API.

#### Khó khăn

Trong quá trình xây dựng hệ thống web đăng ký khóa học ngoại ngữ, nhóm đã gặp phải một số khó khăn như sau:

- Việc xây dựng cơ sở dữ liệu mất nhiều thời gian do chưa nắm được cần phải lưu những gì và quy trình đăng ký khóa học diễn ra như thế nào.
- Mặc dù frontend sử dụng ReactJS và backend sử dụng Node.js (Express) là các công nghệ phổ biến, nhưng việc kết nối API, xác thực JWT và xử lý phân quyền đôi khi gặp lỗi bất đồng bộ hoặc cấu hình sai.
- Khi thử nghiệm triển khai trên Docker, nhóm gặp khó khăn về cấu hình Dockerfile, docker-compose.
- Việc đảm bảo giao diện thân thiện, dễ sử dụng cho cả học viên và admin đòi hỏi nhiều lần chỉnh sửa và tham khảo ý kiến người dùng.

## 7.2 Kết luận

### Bài học rút ra

Qua quá trình thực hiện dự án, nhóm đã rút ra nhiều bài học quý giá:

- Làm việc theo mô hình Agile/SCRUM giúp dễ kiểm soát tiến độ và phân chia công việc hiệu quả, đặc biệt với các sprint ngắn, giúp phát hiện lỗi sớm.
- Thiết kế cơ sở dữ liệu và kiến trúc hệ thống ngay từ đầu là yếu tố then chốt, tránh phải sửa chữa về sau gây ảnh hưởng dây chuyền.
- Tầm quan trọng của kiểm thử và phản hồi người dùng: Việc kiểm thử giao diện, chức năng và thu thập phản hồi từ người dùng giúp sản phẩm cuối cùng đáp ứng đúng nhu cầu thực tế.

### Đề xuất cải thiện trong tương lai

Để nâng cao chất lượng và tính thực tiễn của hệ thống, nhóm có một số hướng cải thiện sau:

- Tích hợp hệ thống thanh toán online để học viên có thể đăng ký và thanh toán học phí trực tiếp qua website.
- Phát triển chức năng đánh giá khóa học để học viên phản hồi chất lượng sau khi kết thúc, giúp trung tâm cải tiến nội dung giảng dạy.
- Thêm tính năng thống kê và báo cáo cho admin, giúp theo dõi tình hình đăng ký.
- Triển khai ứng dụng web lên host như Google Cloud hoặc AWS để thử nghiệm.

## TÀI LIỆU THAM KHẢO

- [1] “Công nghệ phần mềm,” *Wikipedia tiếng Việt*. May 14, 2025. Accessed: Jul. 22, 2025. [Online]. Available:  
[https://vi.wikipedia.org/w/index.php?title=C%C3%B4ng\\_ngh%E1%BB%87\\_ph%E1%BA%A7n\\_m%E1%BB%81m&oldid=73510016](https://vi.wikipedia.org/w/index.php?title=C%C3%B4ng_ngh%E1%BB%87_ph%E1%BA%A7n_m%E1%BB%81m&oldid=73510016)
- [2] “Các Mô Hình Phát Triển Phần Mềm - Viblo.” Accessed: Jul. 22, 2025. [Online]. Available: <https://viblo.asia/p/cac-mo-hinh-phat-trien-phan-mem-GrLZDwbGKk0>
- [3] “Tổng quan về kiến trúc hệ thống.” Accessed: Jul. 22, 2025. [Online]. Available: <https://viblo.asia/p/tong-quan-ve-kien-truc-he-thong-4P856nAL5Y3>

## PHỤ LỤC

### Cài Đặt

#### 1. Clone dự án

`git clone https://github.com/phuchochiep131/Language\_Course\_Registration\_System.git`

#### 2. Di chuyển vào thư mục dự án

`cd Language_Course_Registration_System`

#### 3. Build và chạy dự án

`docker-compose up --build`

Mặc định hệ thống sẽ chạy trên:

Frontend chạy tại: <http://localhost:3000>

Backend chạy tại <http://localhost:3005/api/>

Swagger chạy tại <http://localhost:3005/api-docs>

#### Các Port hệ thống sử dụng sau khi đã chạy thành công

| Container | Port        | Mô tả                                    |
|-----------|-------------|--|
| mongo     | 27017:27017 | Server hệ quản trị cơ sở dữ liệu MongoDB |
| backend   | 3005:3005   | Server API, sử dụng Node.js              |
| frontend  | 3000:3000   | Giao diện người dùng, sử dụng React      |

# API Tài liệu

1.0.0

OAS 3.0

Tài liệu API cho hệ thống đăng ký khóa học ngoại ngữ

## Servers

http://localhost:3005/api

## Auth

Các API liên quan đến xác thực



POST

/auth/register Đăng ký người dùng



POST

/auth/login Đăng nhập



GET

/auth/logout Đăng xuất



POST

/auth/forgot-password Quên mật khẩu



POST

/auth/reset-password Đặt lại mật khẩu



## Courses

Quản lý khóa học



GET

/course Lấy tất cả khóa học



POST

/course Tạo mới một khóa học



GET

/course/{id} Lấy thông tin khóa học theo ID



PUT

/course/{id} Cập nhật khóa học theo ID



DELETE

/course/delete-many Xóa nhiều khóa học theo \_id



# Languages

Quản lý ngôn ngữ



GET

/language Lấy danh sách tất cả ngôn ngữ



POST

/language/add Thêm ngôn ngữ mới



DELETE

/language/multiple Xóa nhiều ngôn ngữ



GET

/language/{id} Lấy ngôn ngữ theo ID



PUT

/language/{id} Cập nhật thông tin ngôn ngữ



# LanguageLevels

Quản lý trình độ ngôn ngữ



GET

/languagelevel Lấy tất cả trình độ ngôn ngữ



POST

/languagelevel/add Thêm trình độ ngôn ngữ mới



DELETE

/languagelevel/multiple Xóa nhiều trình độ ngôn ngữ



GET

/languagelevel/{id} Lấy trình độ ngôn ngữ theo ID



PUT

/languagelevel/{id} Cập nhật thông tin trình độ ngôn ngữ



# Teachers

Quản lý giảng viên



GET

/teacher Lấy danh sách tất cả giảng viên



POST

/teacher Thêm mới giảng viên



GET

/teacher/{id} Lấy thông tin giảng viên theo ID



PUT

/teacher/{id} Cập nhật thông tin giảng viên



DELETE

/teacher/multiple Xóa nhiều giảng viên





# Users

Quản lý người dùng và đăng ký khóa học



GET

/user/registered-course Lấy tất cả đăng ký khóa học (không cần auth)



GET

/user Lấy danh sách tất cả người dùng (admin)



GET

/user/info Lấy thông tin người dùng hiện tại (token)



GET

/user/{id} Lấy thông tin người dùng theo ID



PUT

/user/{id} Cập nhật thông tin người dùng theo ID



GET

/user/{id}/registered-courses Lấy danh sách khóa học đã đăng ký của người dùng



PUT

/user/update-registration/{userId}/{courseId} Cập nhật thông tin đăng ký khóa học



DELETE

/user/{id}/unregister-course/{courseId} Hủy đăng ký khóa học



DELETE

/user/multiple Xóa nhiều người dùng (admin)



POST

/user/{id}/register-course Đăng ký khóa học cho người dùng

