

# Báo cáo kết thúc môn CÔNG NGHỆ PHẦN MỀM

## XÂY DỰNG ỨNG DỤNG WEB ĐĂNG KÝ HỌC NGOẠI NGỮ

*Giáo 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



# Nội dung

1. Lí do chọn đề tài
2. Mục tiêu của ứng dụng
3. Các chức năng chính của hệ thống
4. Thiết kế hệ thống
5. Công nghệ đã sử dụng
6. Đánh giá và kết luận

# 1. Lí do chọn đề tài

- **Nhu cầu học ngoại ngữ lớn và tăng cao:** Đặc biệt tại các thành phố lớn, ngoại ngữ là yếu tố thiết yếu để tìm kiếm việc làm.
- **Hạn chế của phương pháp đăng ký truyền thống:**
  - Ghi danh trực tiếp gây bất tiện.
  - Quản lý bằng giấy tờ dễ sai sót, thiếu hiệu quả.
- **Giải pháp đề xuất:** Xây dựng ứng dụng web đăng ký học ngoại ngữ để giải quyết các vấn đề trên.
- **Ý nghĩa thực tiễn của dự án:**
  - Ứng dụng kiến thức đã học (phát triển hệ thống thông tin, công nghệ phần mềm, lập trình web, quản lý cơ sở dữ liệu) vào một bài toán thực tế.
  - Góp phần cải thiện quy trình đăng ký và quản lý tại các trung tâm ngoại ngữ.

## 2. Mục tiêu của ứng dụng

- 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.

### 3. Các chức năng chính của hệ thống

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.

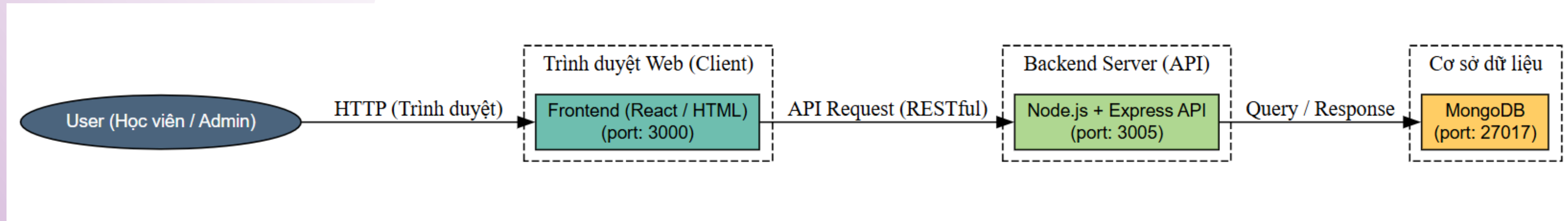
#### Đối với quản trị viên (Admin):

- 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).

# 4. Thiết kế hệ thống

## 4.1. 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).

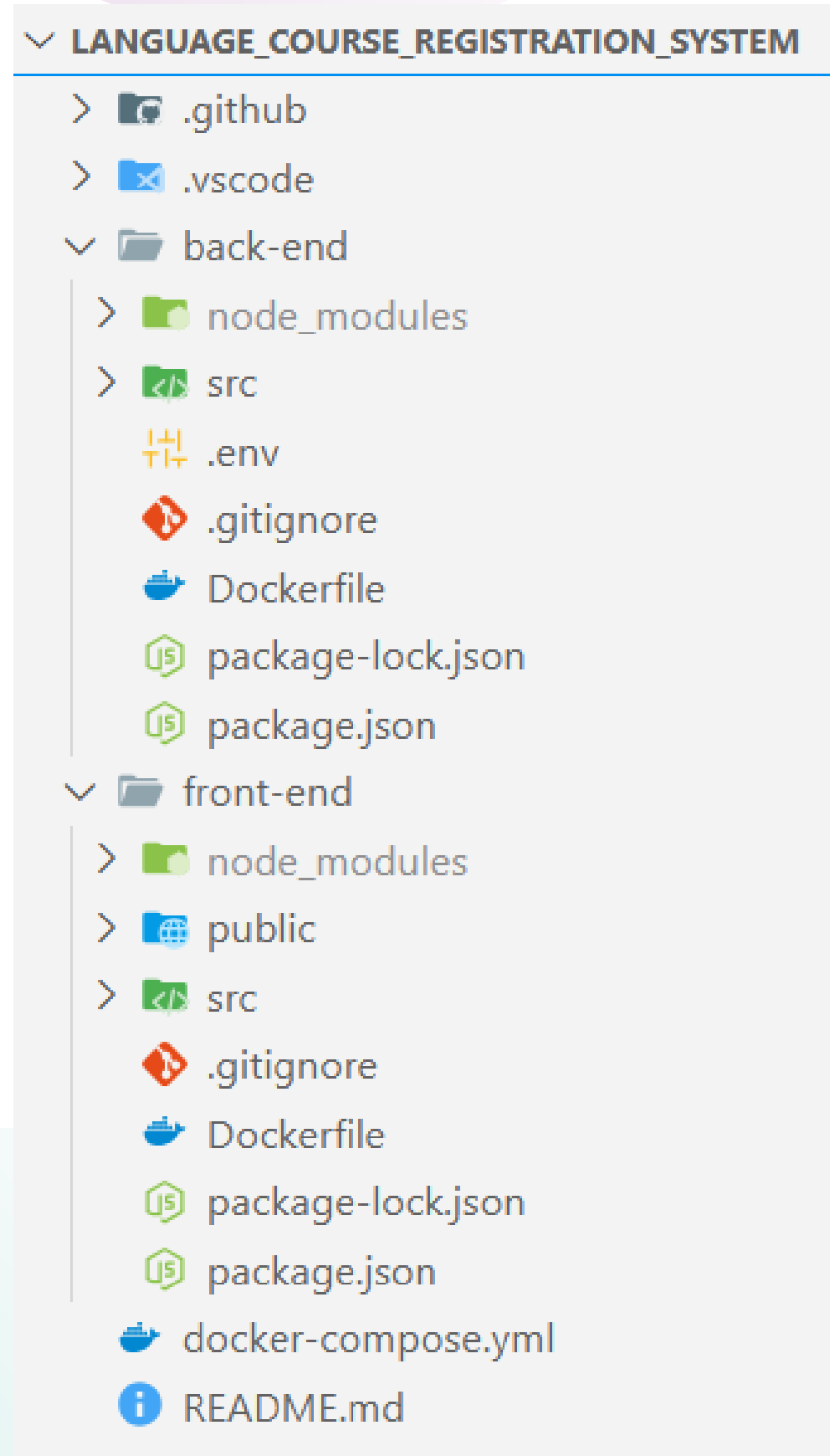


### Luồng hoạt động của hệ thống:

- 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.

# 4. Thiết kế hệ thống

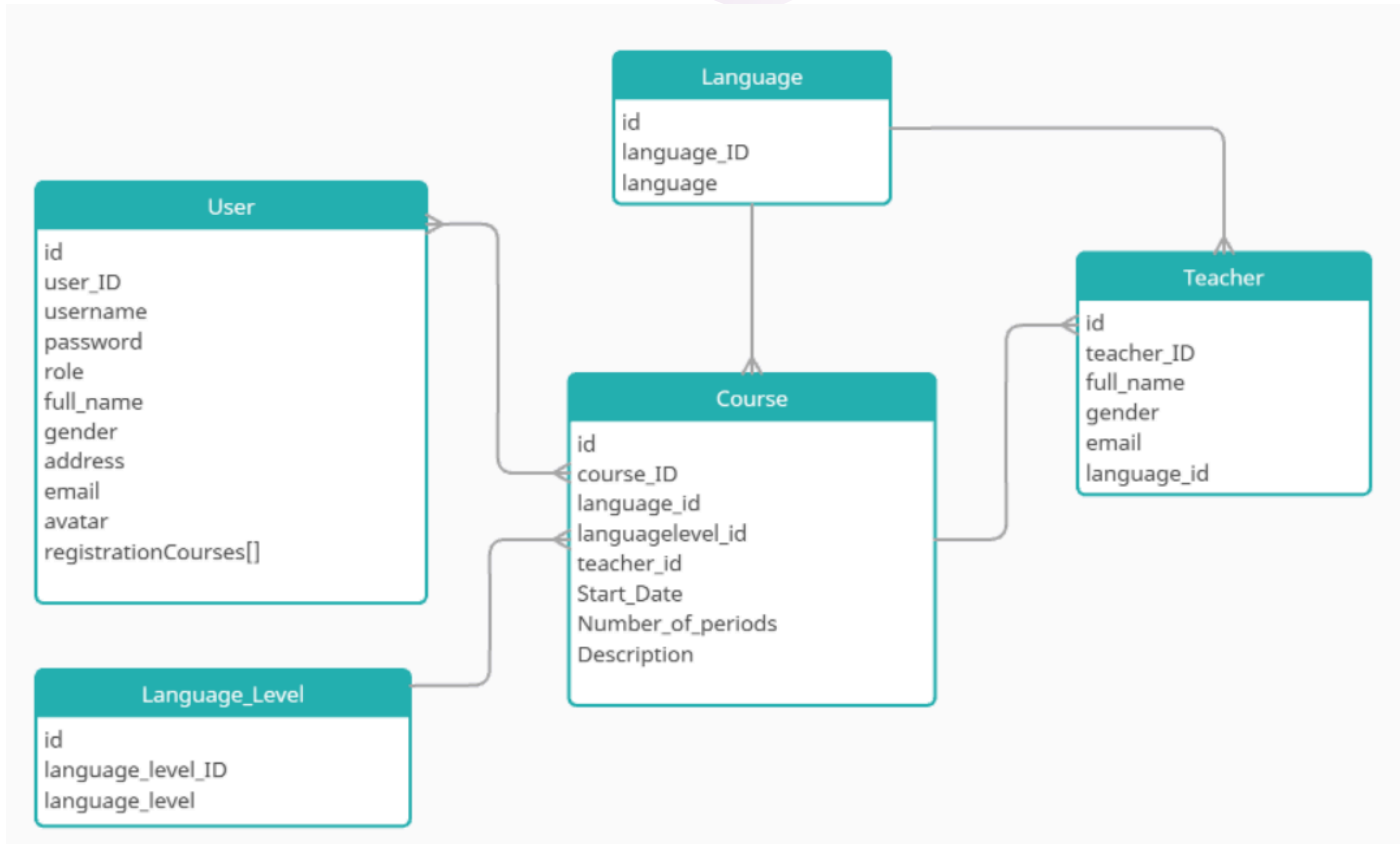
## 4.2. Cấu trúc thư mục





# 4. Thiết kế hệ thống

## 4.3. Thiết kế cơ sở dữ liệu





# 4. Thiết kế hệ thống

## 4.4. Thiết kế 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:

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

# 4. Thiết kế hệ thống

## 4.4. Thiết kế API

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:

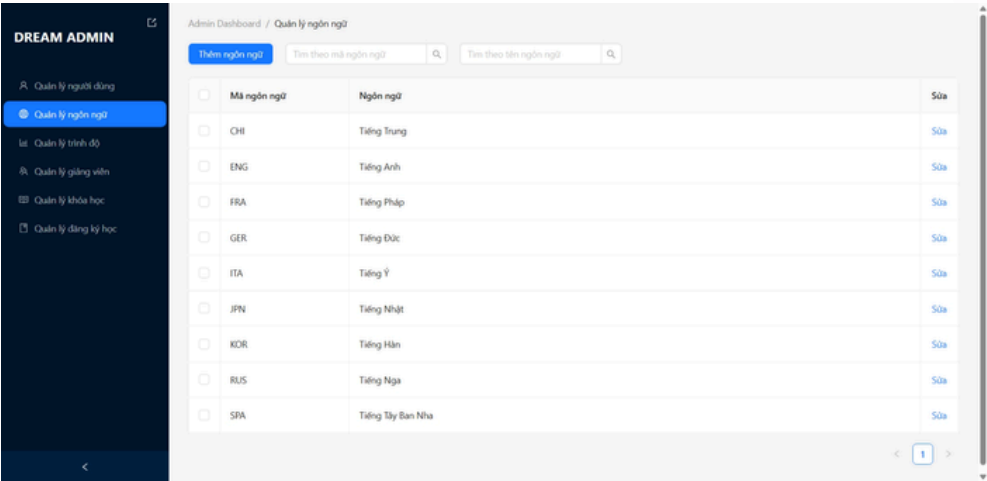
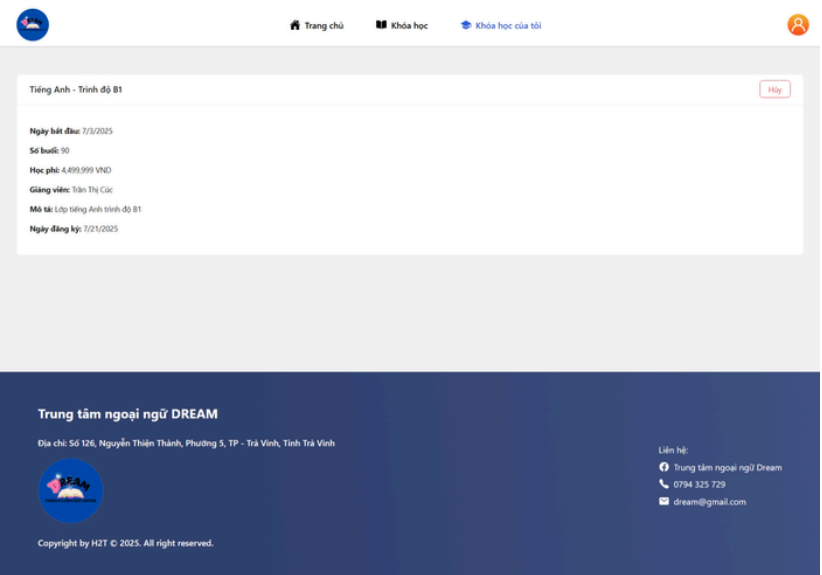
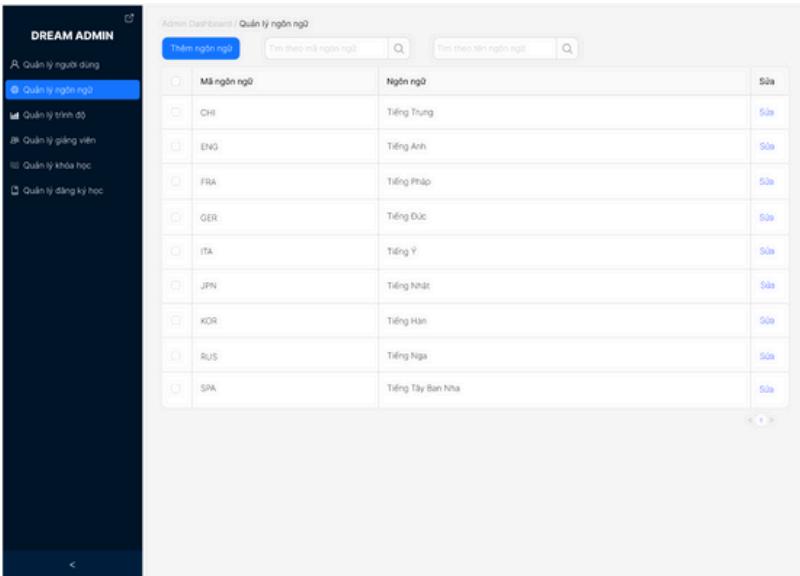
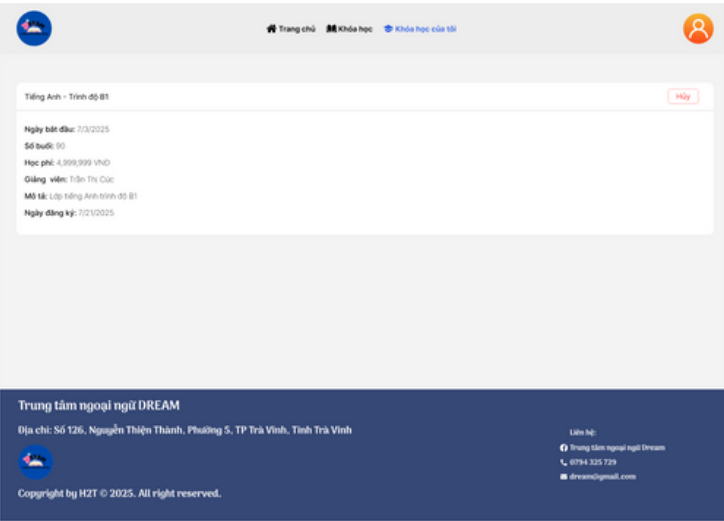
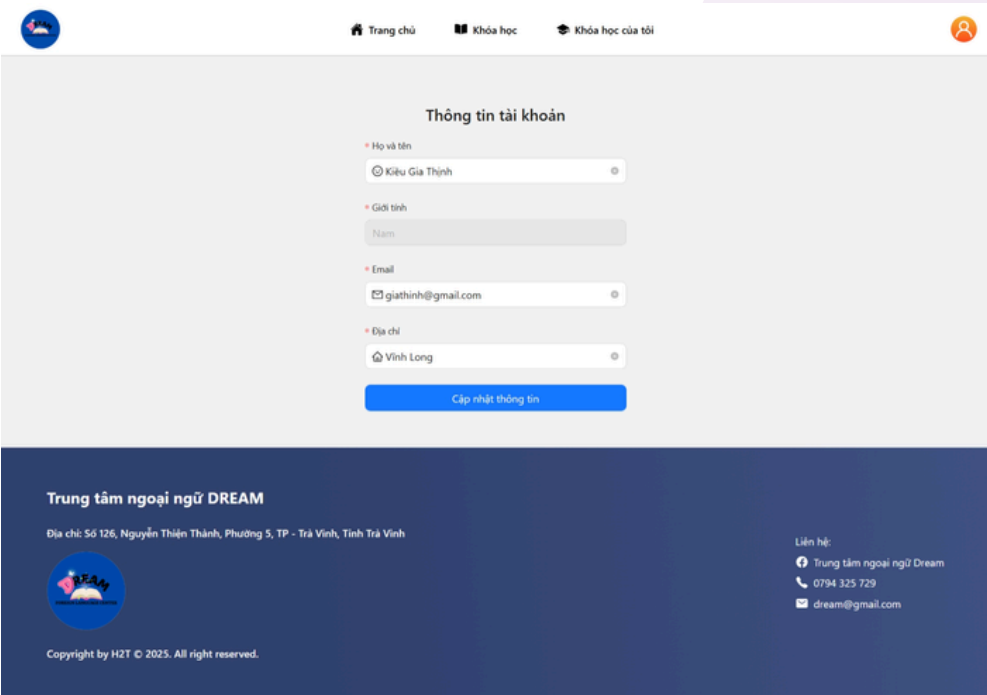
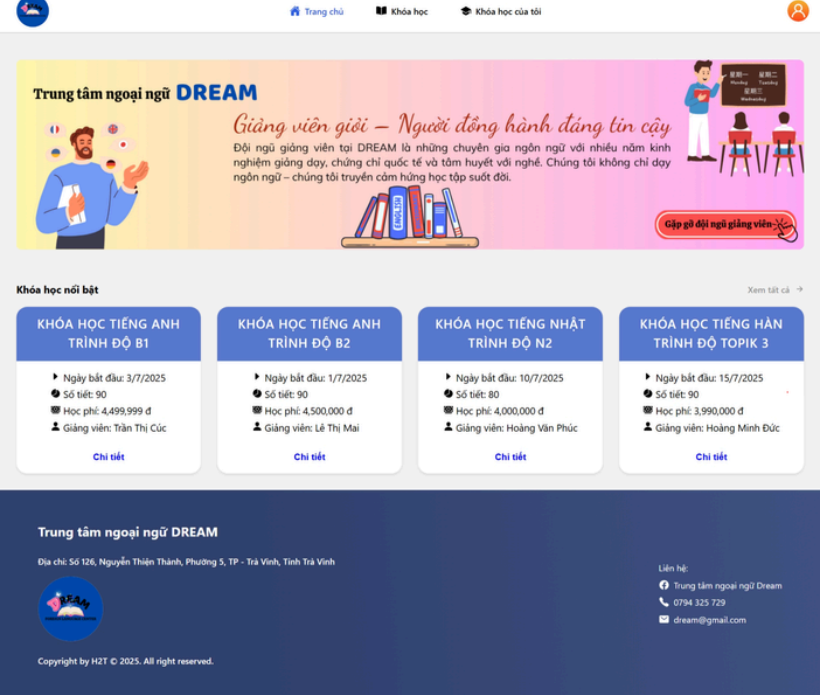
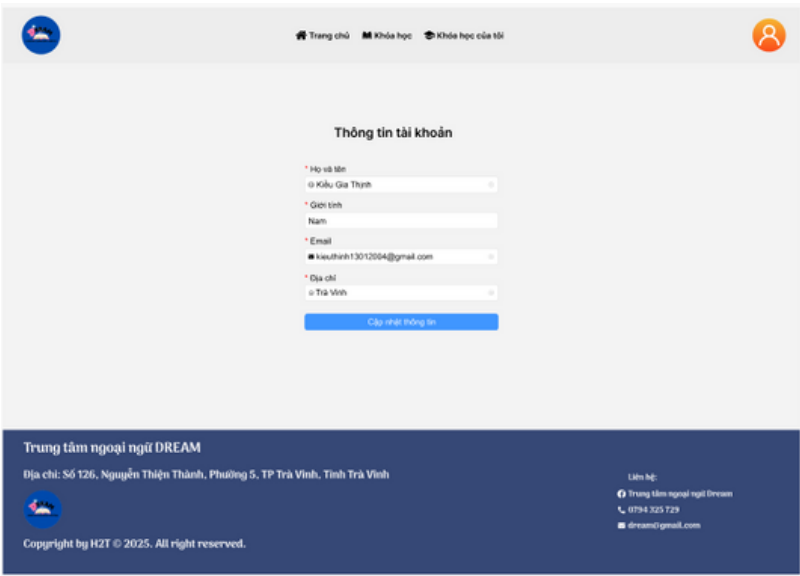
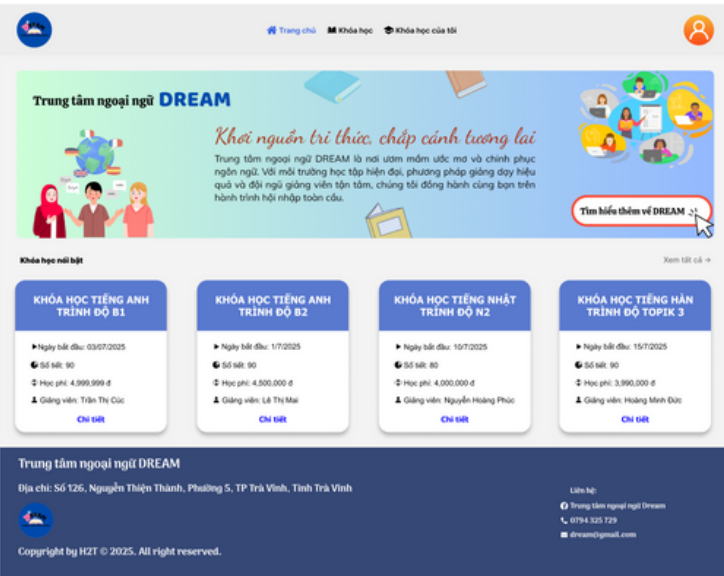
Method	Endpoint	Mô tả
GET	/api/teacher	Lấy danh sách tất cả giảng viên
POST	/api/teacher	Tạo mới giảng viên
PUT	/api/teacher/:id	Cập nhật giảng viên
DELETE	/api/teacher/multiple	Xóa nhiều giảng viên

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

# 4. Thiết kế hệ thống

## 4.5. Thiết kế giao diện (UI/UX)



Thiết kế Figma

Phát triển trên web

# 5. 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.

# 5. Công nghệ đã sử dụng

## 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.



## 6. Đánh giá và kết luận

### Kết quả đạt được

- 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:

- 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.

# 6. Đánh giá và kết luận

## Bài học rút ra

- 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

- 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.



