

# Báo cáo dự án Đăng ký học phần

MÔN: KIẾN TRÚC & THIẾT KẾ PHẦN MỀM

GVHD: VÕ VĂN HẢI

Nhóm thực hiện: Nhóm 12



# M Ụ C L Ụ C

---

1. Tổng quan dự án
2. Cơ sở lý thuyết
3. Phân tích thiết kế
4. Hiện thực
5. Hoạt động kết thúc dự án

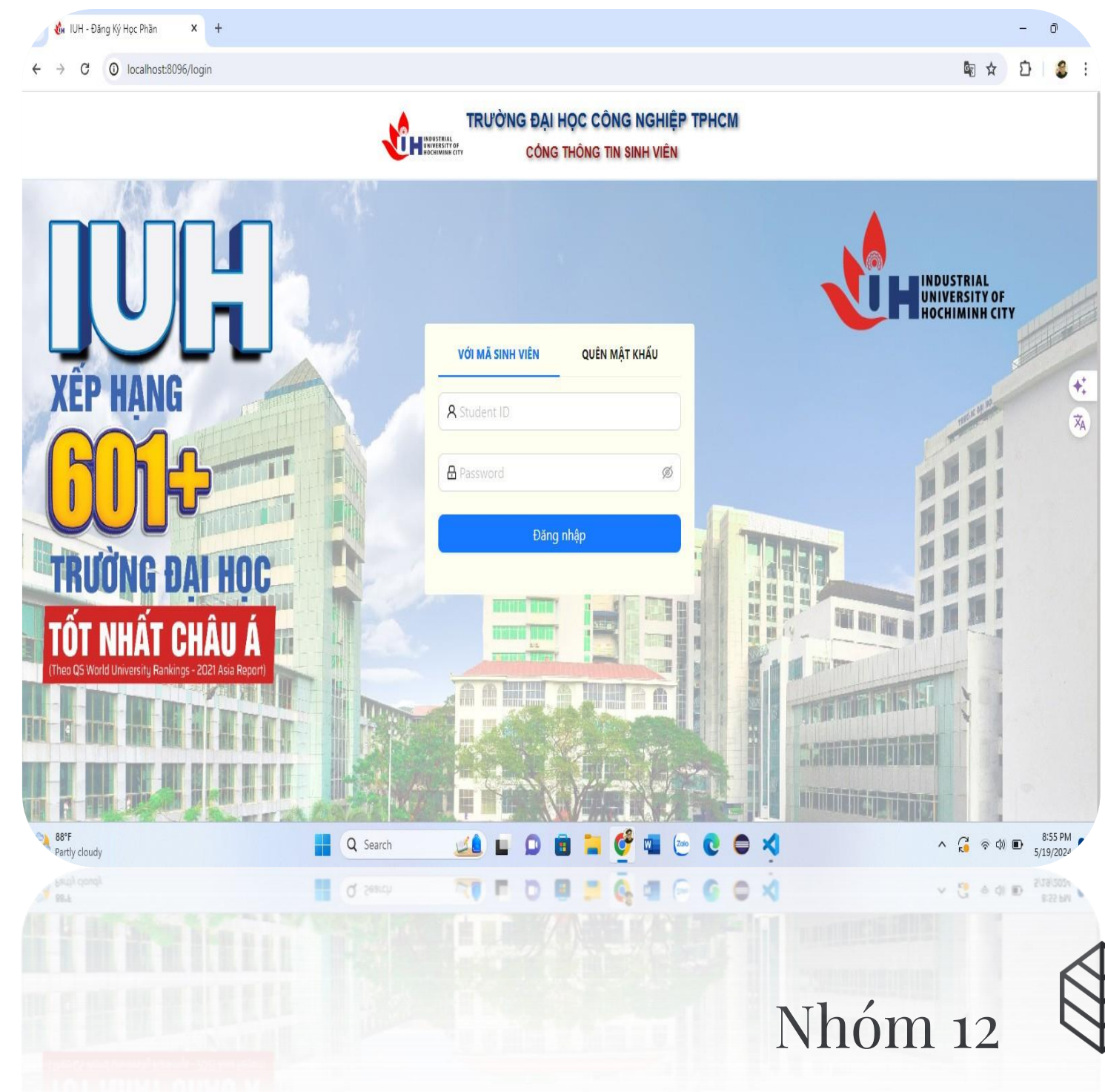


# I. TỔNG QUAN DỰ ÁN

## Mục tiêu dự án

Dự án *Đăng ký Học phần* là một nền tảng trực tuyến được phát triển nhằm tạo ra phục vụ việc đăng ký và quản lý học phần của sinh viên.

Với Dự án Đăng ký Học phần, sinh viên có thể dễ dàng đăng ký các học phần, xem lịch học, và xem tiến trình học của mình trên trang web từ bất kỳ thiết bị nào có kết nối internet.



# I. TỔNG QUAN DỰ ÁN

---

## Phạm vi dự án

1. Phát triển dự án trên nền tảng web phục vụ cho các thiết bị có kết nối Internet.
2. Cung cấp tính năng về quản lý tài khoản, thông tin học tập và đăng ký học phần.
3. Đảm bảo tính bảo mật và quyền riêng tư cho người dùng.
4. Tích hợp các tính năng phụ như xem lịch học, xem kết quả học tập.



# I. TỔNG QUAN DỰ ÁN

---

## Yêu cầu chức năng

1. Tạo tài khoản/ đăng nhập/ quản lý mật khẩu
2. Xem các môn học theo chuyên ngành, xem các lớp học tương ứng theo môn và đăng ký.
3. Xác nhận các lớp đã đăng ký.
4. Xem lại lịch học đã đăng ký.
5. Xem kết quả học tập.
6. Xem thông tin chung trên trang cá nhân sinh viên.

## Yêu cầu phi chức năng

1. Tính khả dụng
2. Tính bảo mật
3. Dễ sử dụng
4. Tính tương thích



## II. CƠ SỞ LÝ THUYẾT

---

### Cơ sở lý thuyết



Backend hệ thống



Fontend hệ thống



Chứng thực OTP



Lưu trữ dữ liệu





### III. PHÂN TÍCH & THIẾT KẾ

---

#### Phân tích các yêu cầu của hệ thống để chọn kiến trúc

**Dự án “Đăng ký học phần” yêu cầu:**

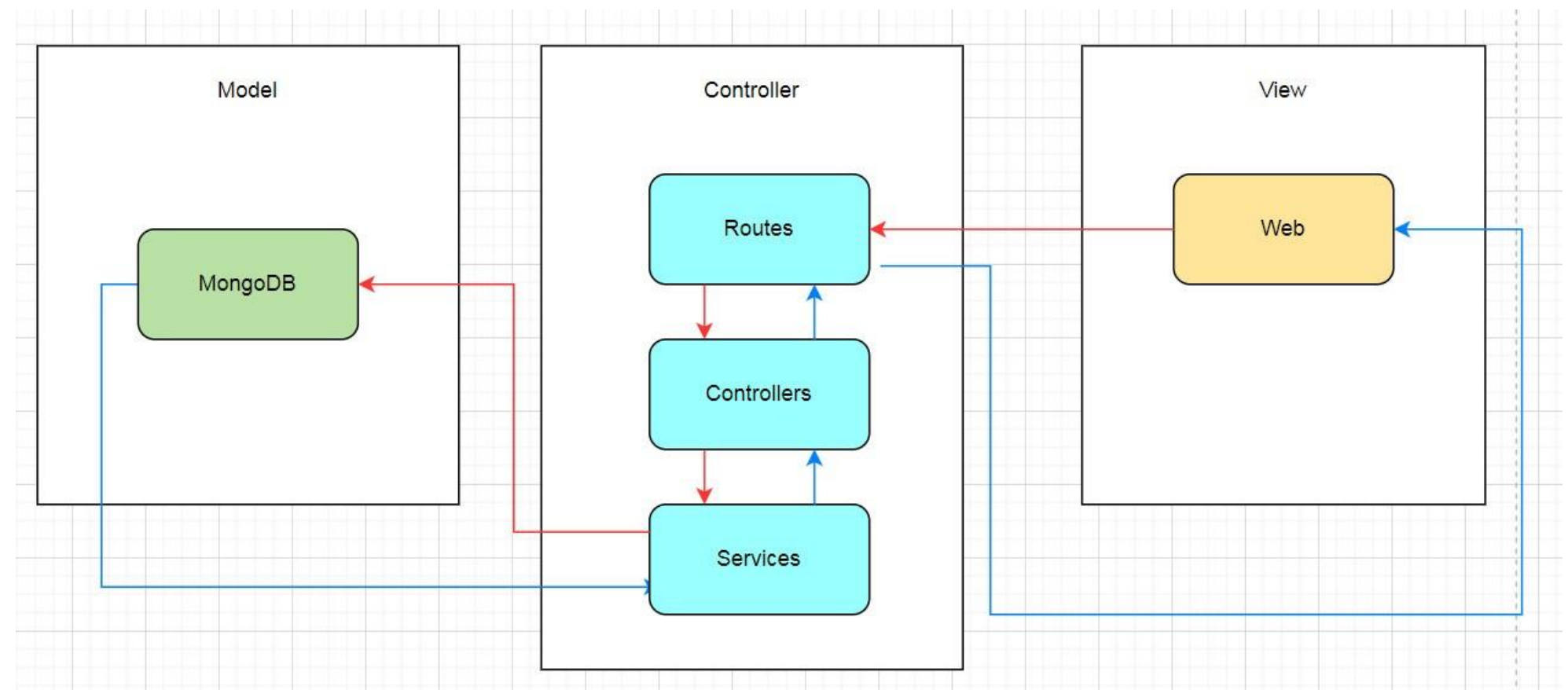
1. Các thành phần hệ thống phân tách rõ ràng, dễ bảo trì và sửa chữa.
2. Đảm bảo hoạt động ổn định, không yêu cầu tốc độ phản hồi cao nhưng cần độ sẵn sàng.
3. Các luồng được cấu hình rõ ràng.
4. Giao diện cần được phân tách để dễ dàng nâng cấp.
5. Khả năng chịu tải cao.



# III. PHÂN TÍCH & THIẾT KẾ

## Mô hình MVC cho dự án

Mô hình MVC có cấu trúc rõ ràng, có thể phân chia thành các Module và phát triển một cách độc lập. Từ đó giảm thiểu được sự xung đột trong quá trình hiện thực dự án cũng như đảm bảo cho việc mở rộng dự án về sau.





# III. PHÂN TÍCH & THIẾT KẾ

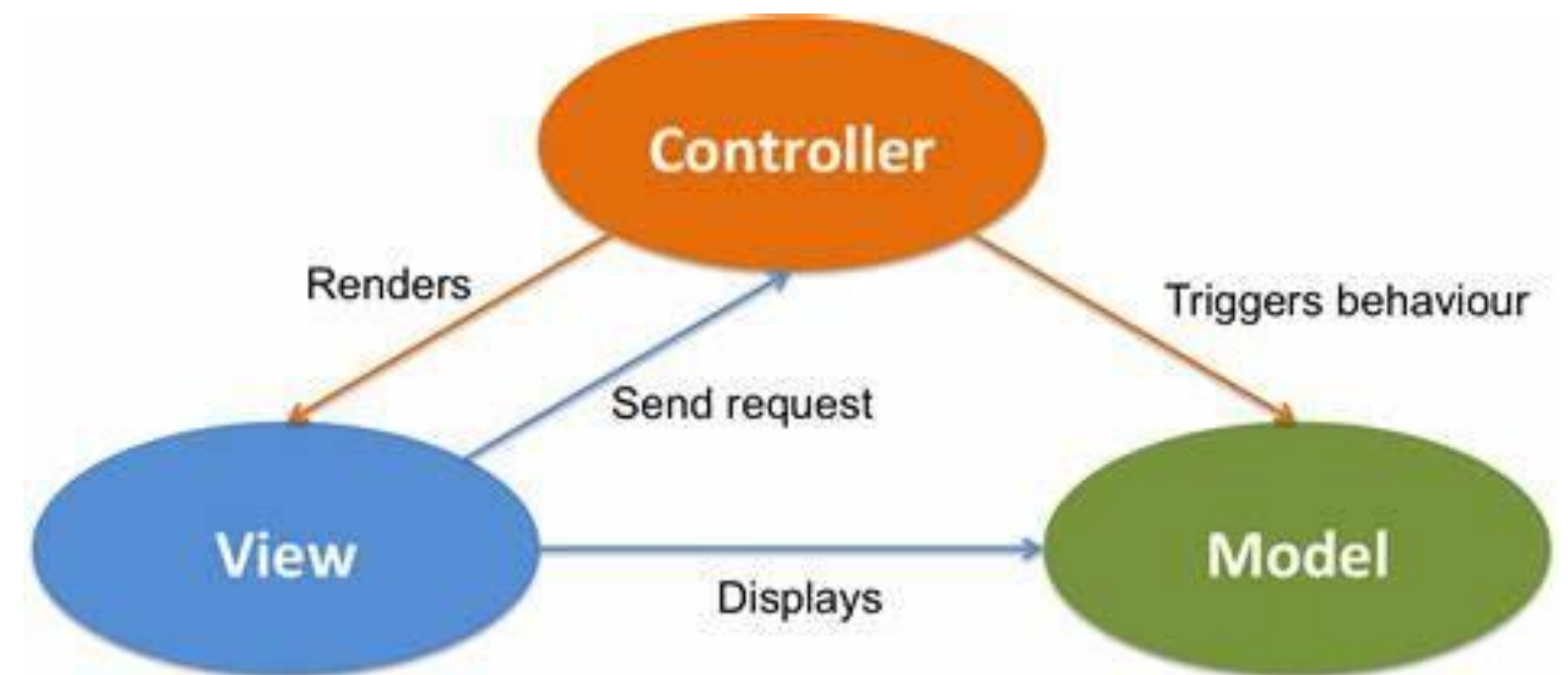
## Lý do chọn kiến trúc MVC cho dự án

Đối với mô hình MVC, ta có thể dễ dàng phân chia các thành phần của hệ thống:

Các Model được lưu trữ riêng biệt. Điều này thích hợp cho dự án DKHP luôn yêu cầu sự thay đổi về mặt cấu trúc dữ liệu.  
VD: các trường của 1 lớp học có thể được cập nhật theo thời gian

View được xây dựng tách biệt, giao tiếp với BE bằng axios. Cho nên có thể được thay đổi, nâng cấp mà không làm ảnh hưởng đến các thành phần khác của hệ thống.

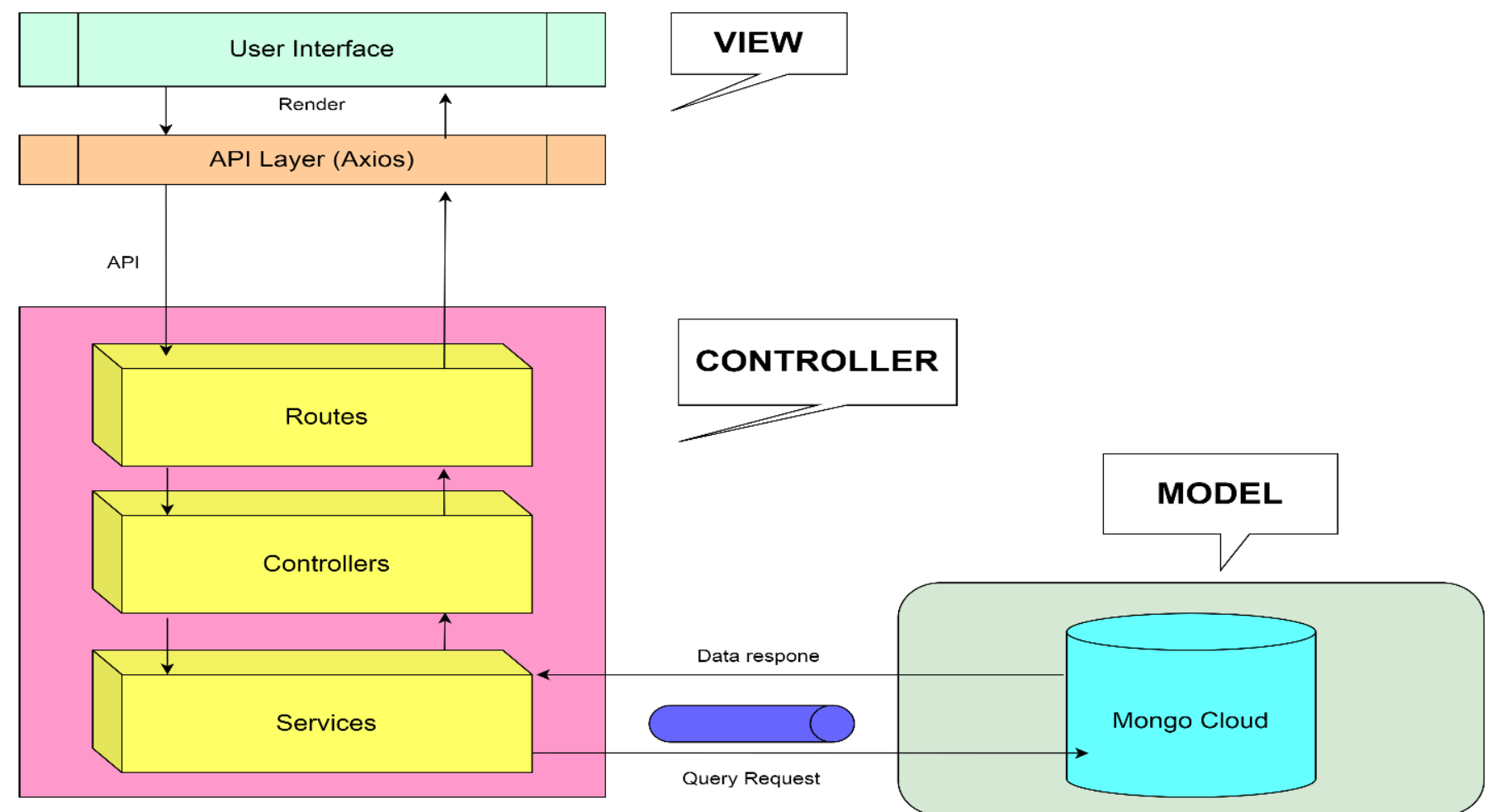
Controller được xây dựng bằng Node.js có thể dẫn đến việc các file controllers có thể trở nên rất lớn, việc tách biệt các file theo cấu trúc MVC có thể giảm tải độ lớn cho các file trong hệ thống.



# III. PHÂN TÍCH & THIẾT KẾ

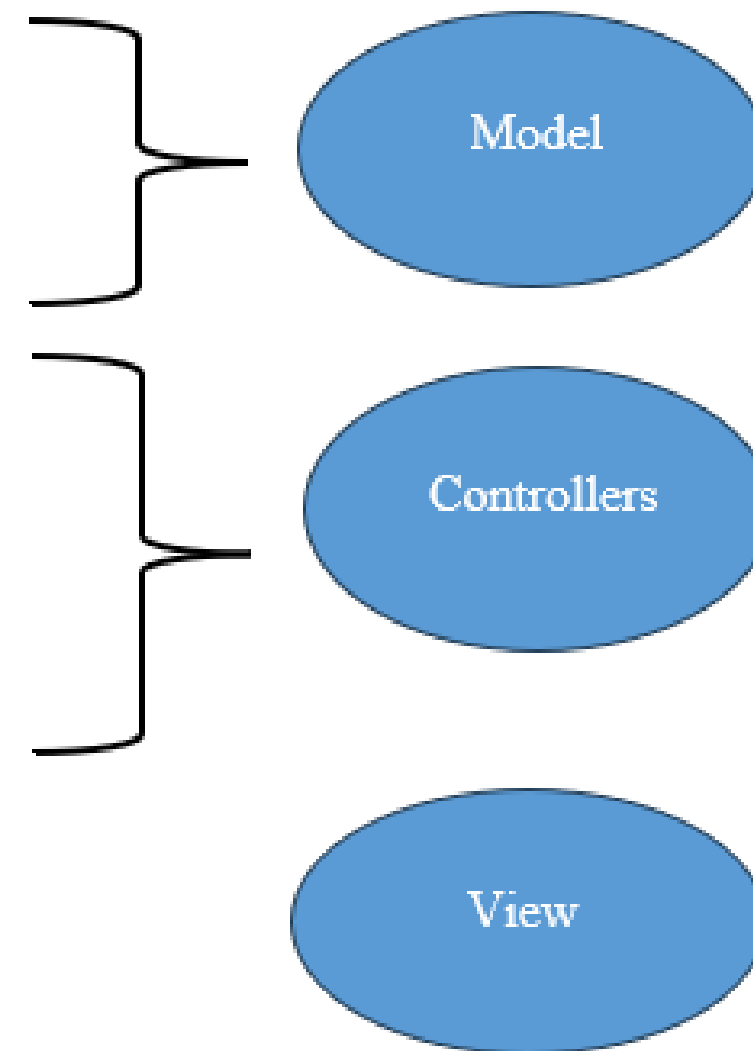
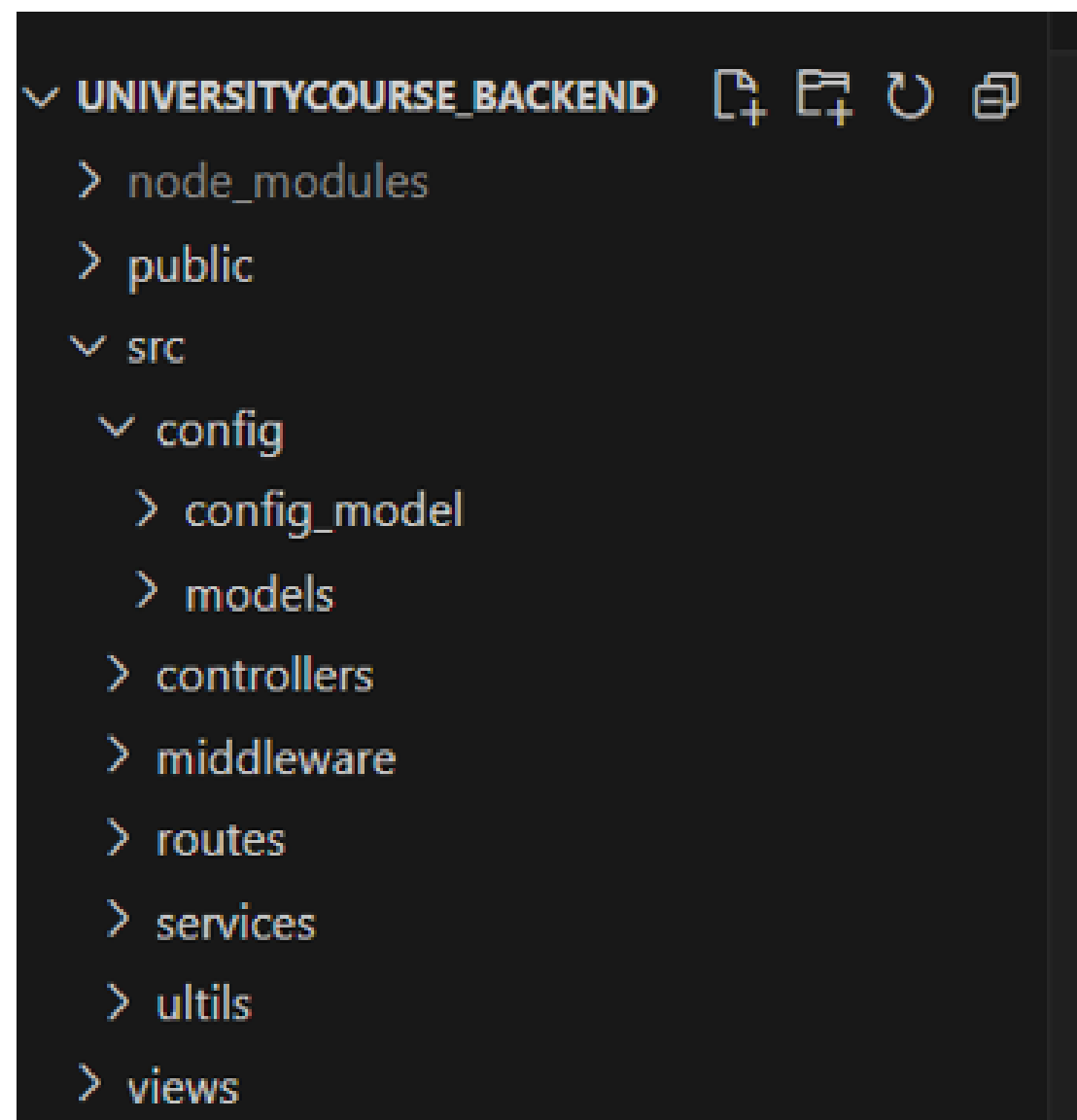
## Hiện thực kiến trúc MVC cho dự án

Giao diện người dùng được xây dựng bằng React giao tiếp với BE thông qua cách gọi API bằng Axios. Các API được định nghĩa thông qua các Routes trong BE, khi được gọi, Route sẽ gọi xuống Controller tương ứng. Controller được gọi sẽ tiếp tục gọi các Service tương ứng. Các Service thực hiện các yêu cầu từ giao diện người dùng và thực hiện update đến Cơ sở dữ liệu. Sau khi các Model được update sẽ trả kết quả truy vấn về BE, sau đó BE sẽ trả kết quả về giao diện người dùng để tiến hành hiển thị.



# III. PHÂN TÍCH & THIẾT KẾ

## Hiện thực kiến trúc MVC cho dự án



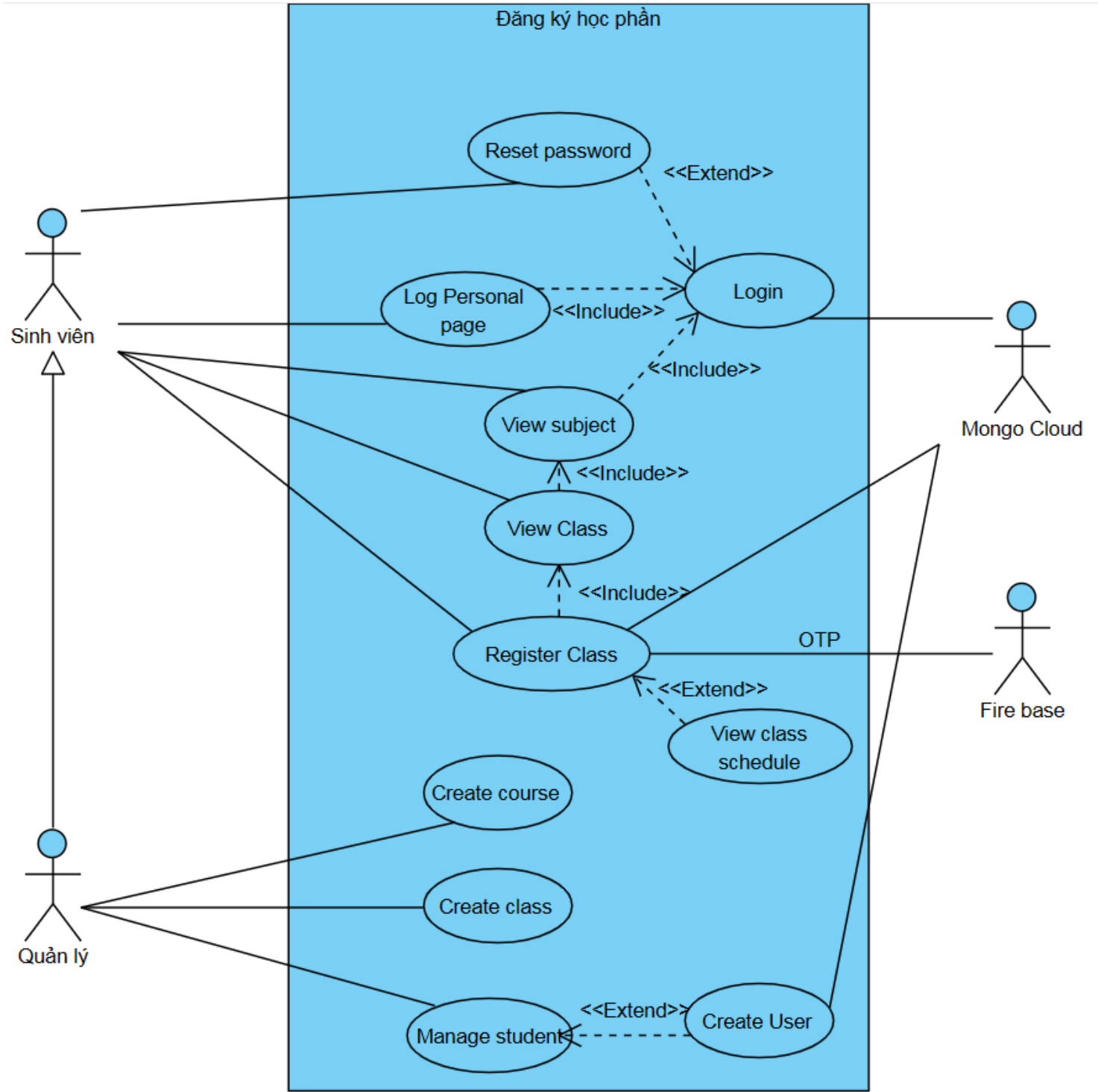
Xây dựng cấu trúc Backend theo mô hình MVC



# III. PHÂN TÍCH & THIẾT KẾ

## Sơ đồ USE\_CASE

Tác nhân	Mô tả tác nhân
Sinh viên	Người sử dụng – thao tác với các chức năng của hệ thống
Quản lý	Người quản lý các đối tượng trong hệ thống
Nodemailer	Dịch vụ chứng thực tài khoản của người dùng
Mongo Cloud	Dịch vụ lưu trữ dữ liệu không quan hệ



### III. PHÂN TÍCH & THIẾT KẾ

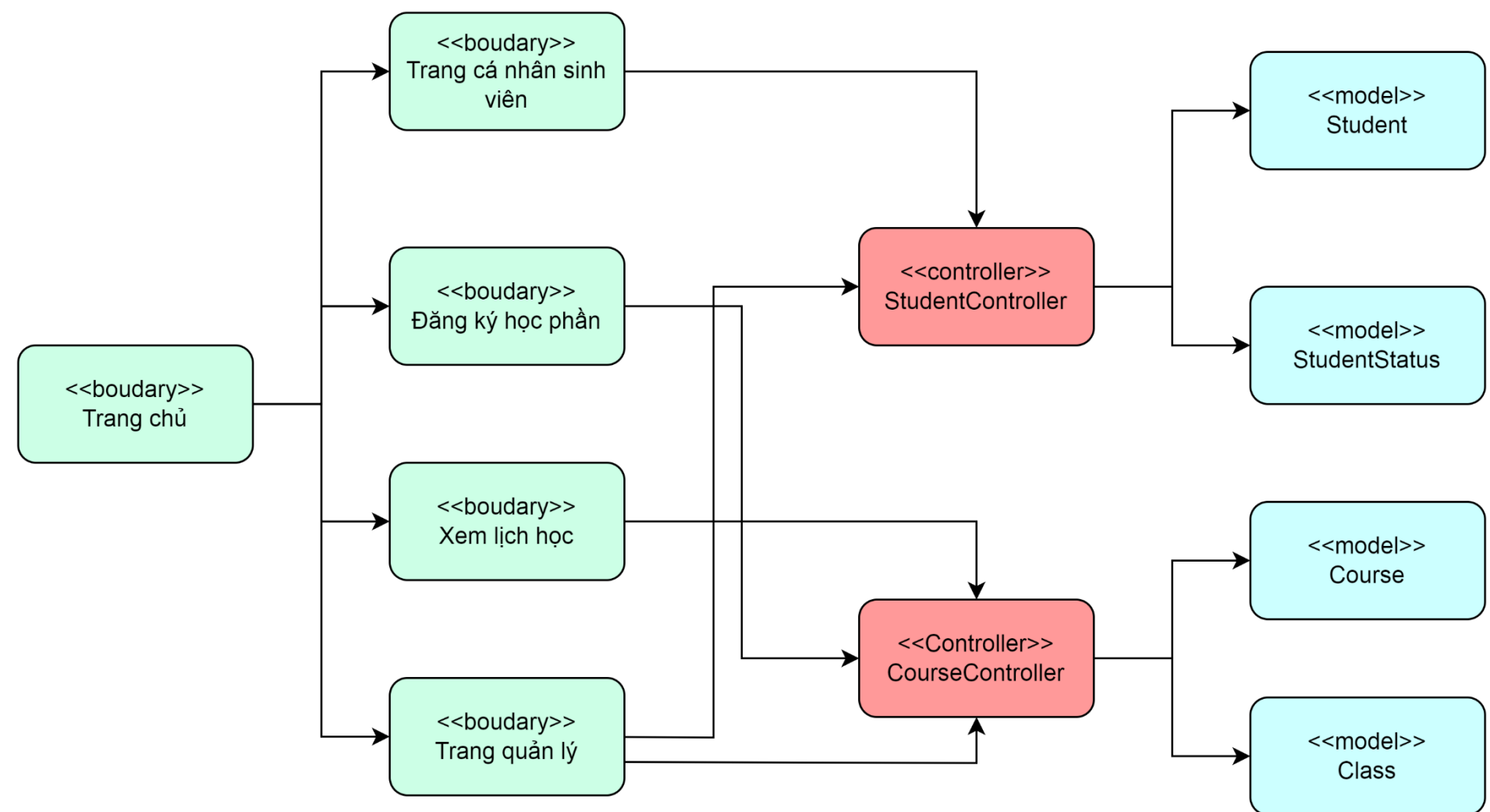
# Sơ đồ lớp tổng quát

Nhóm Boundary chính bao gồm:

- Trang chủ
- Trang đăng ký học phần
- Xem lịch học
- Quản lý

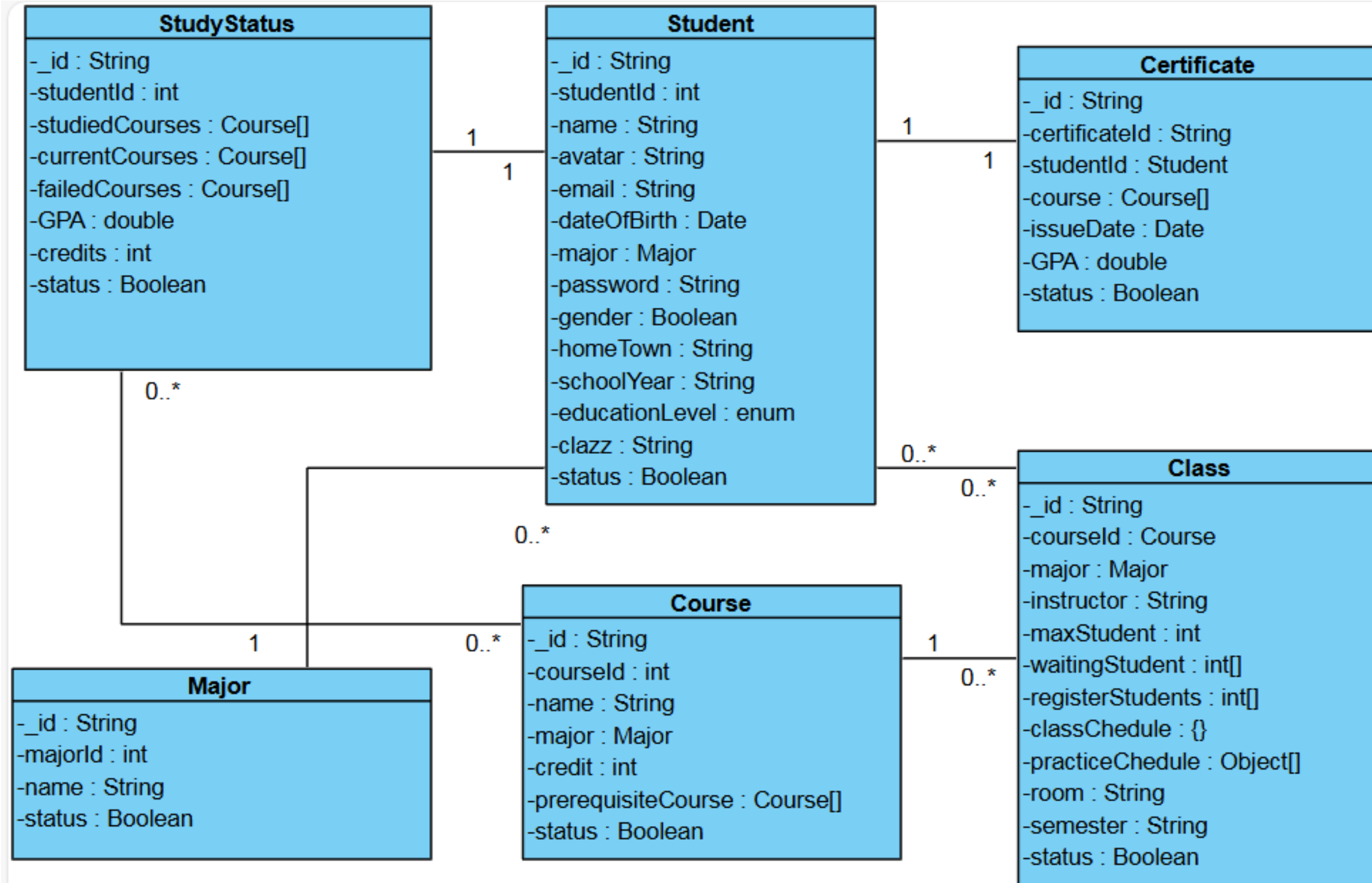
Nhóm Controller chính bao gồm:

- StudentController
- CourseController



# III. PHÂN TÍCH & THIẾT KẾ

## Mô hình Class Entity





# IV. HIỆN THỰC DỰ ÁN

---

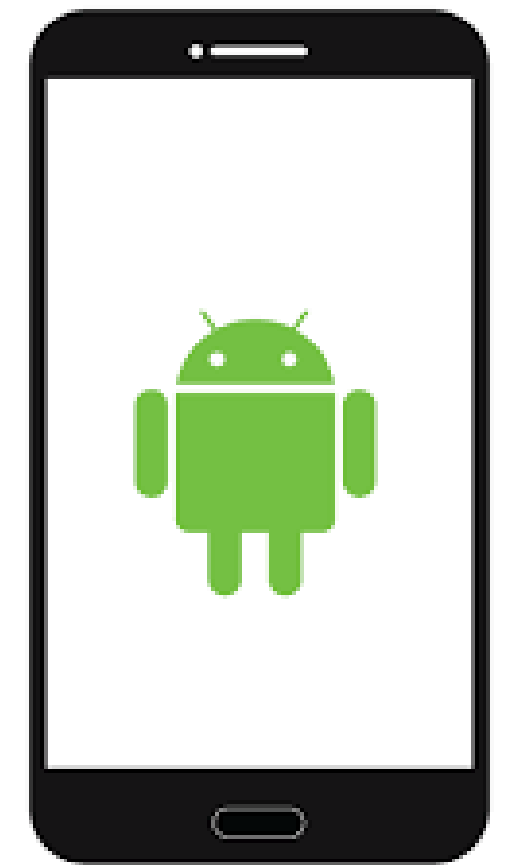
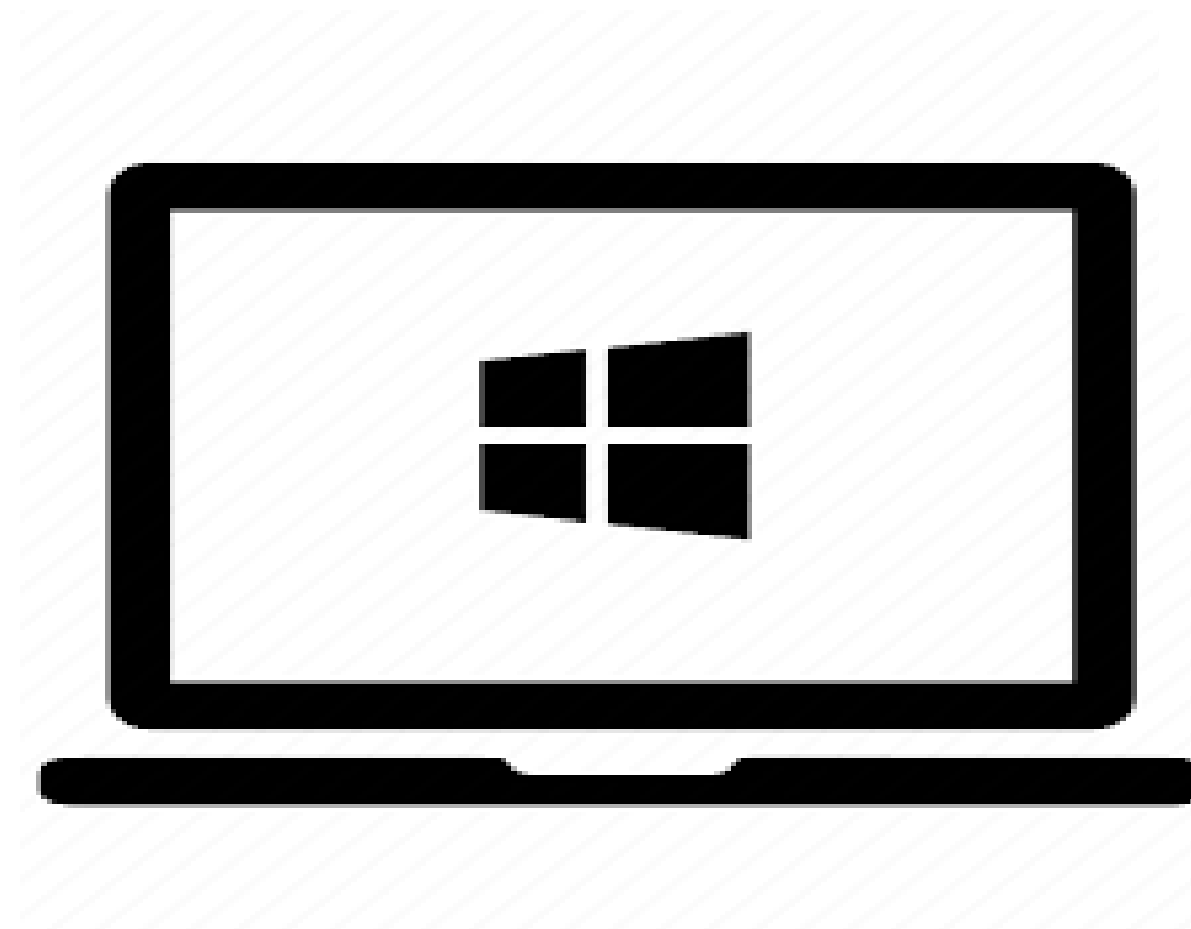
## Cấu hình hệ thống

Các dịch vụ mà dự án đã sử dụng:

- *Mongo Cloud*
- *Firebase AUTH*

Yêu cầu thiết bị kết nối Internet:

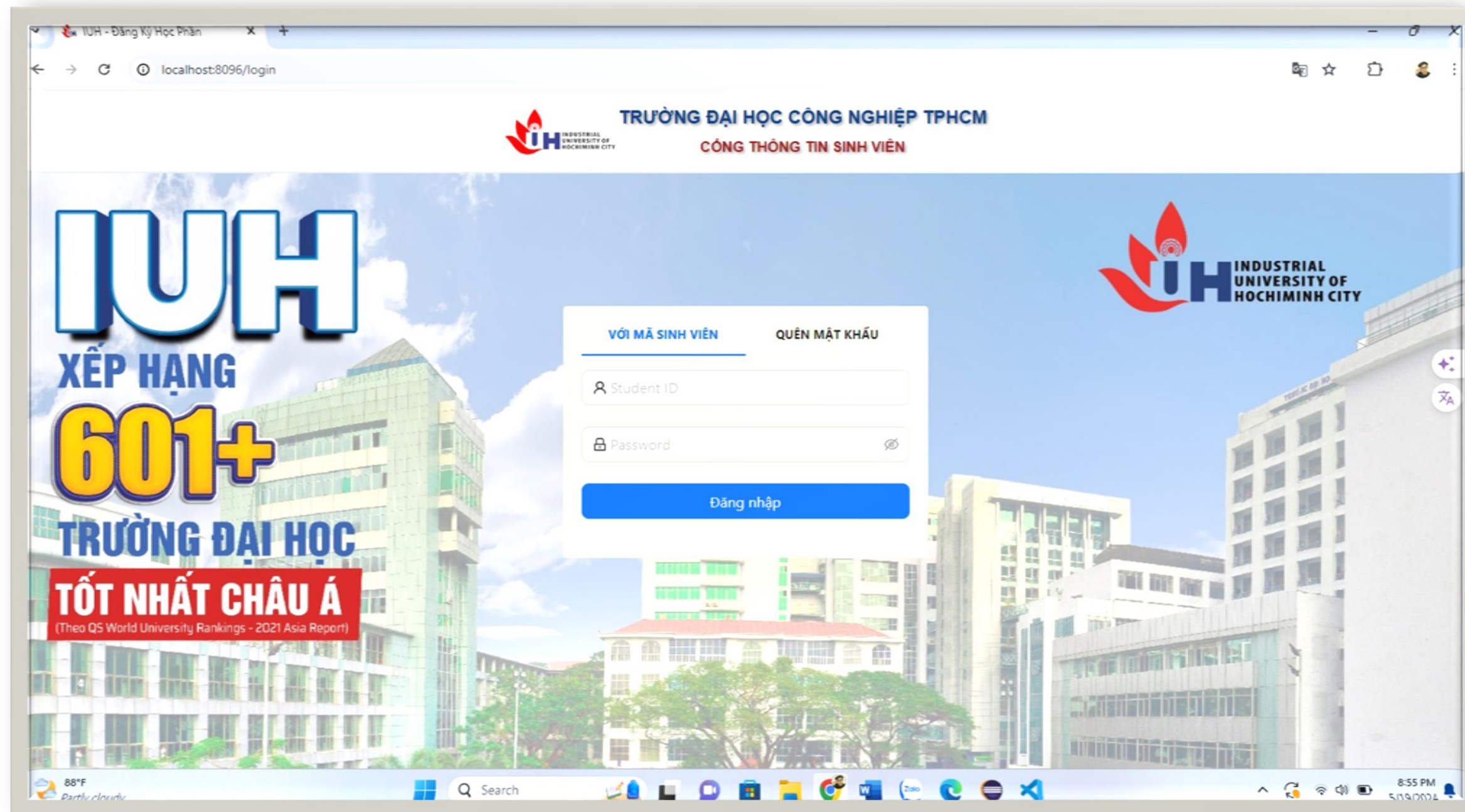
- *Điện thoại hệ điều hành Android, IOS*
- *PC hệ điều hành Window, MacOS*



# IV. HIỆN THỰC DỰ ÁN

## Các chức năng chính

---



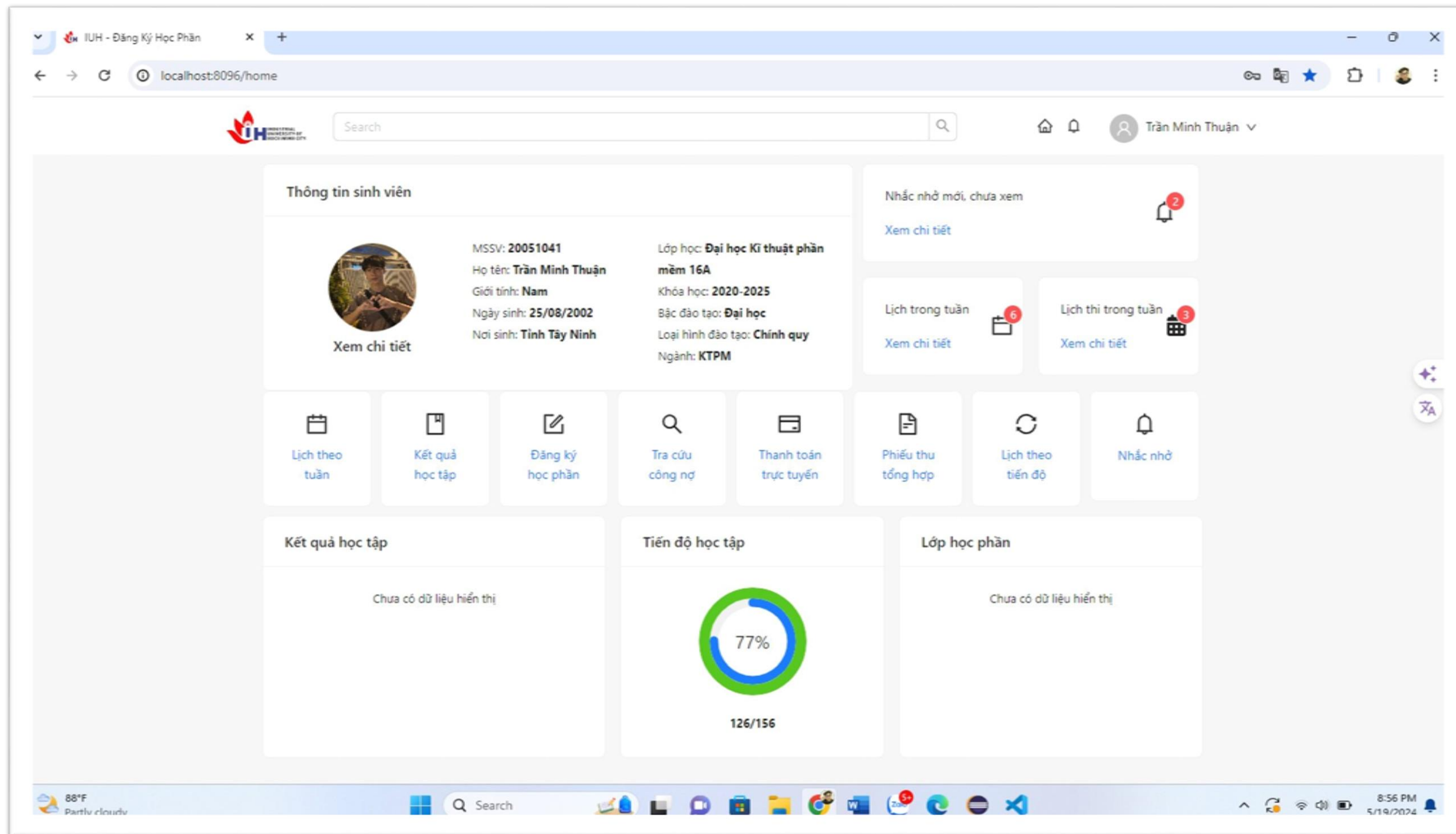
Chức năng đăng nhập

Nhóm 12



# IV. HIỆN THỰC DỰ ÁN

## Các chức năng chính



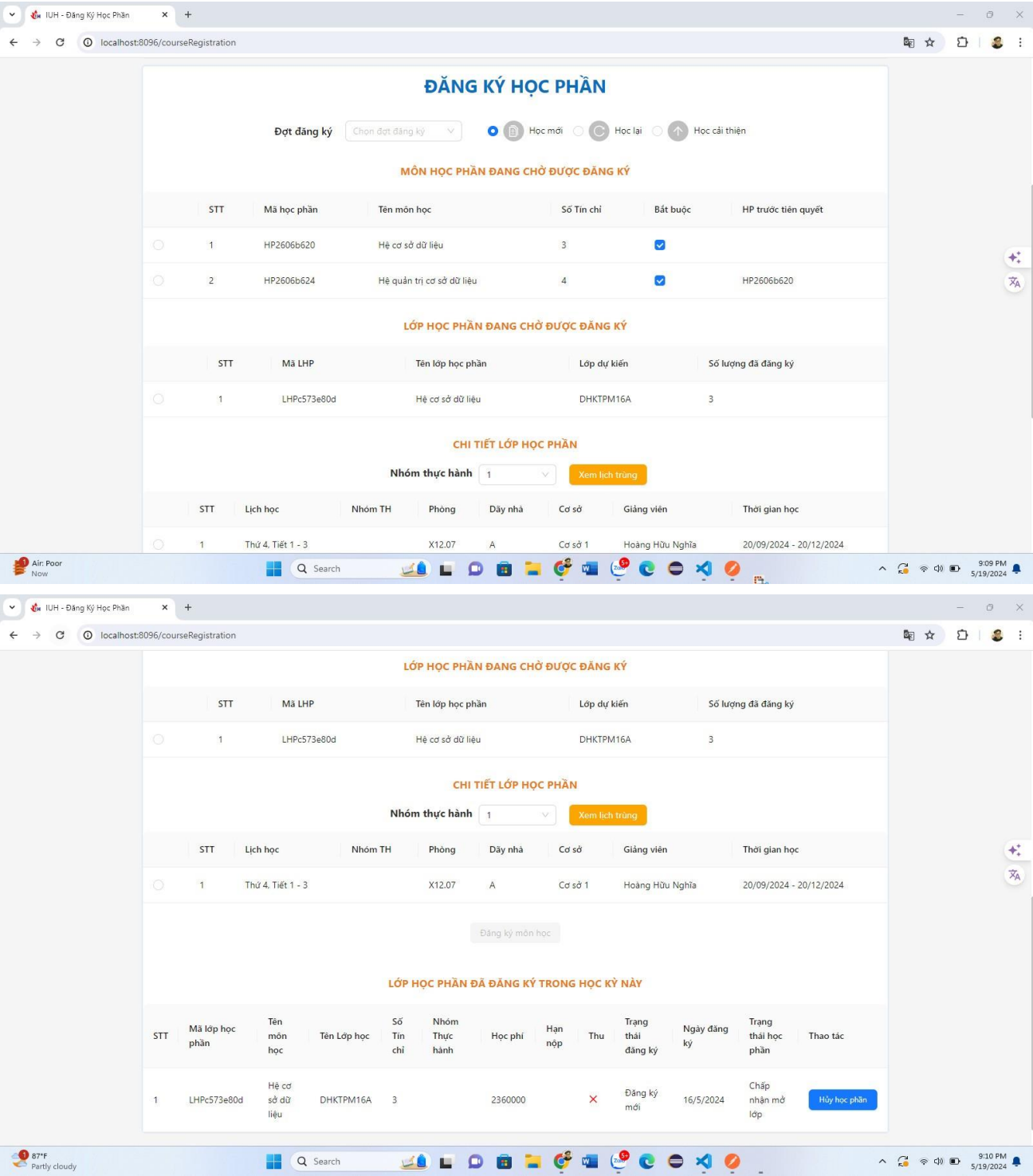
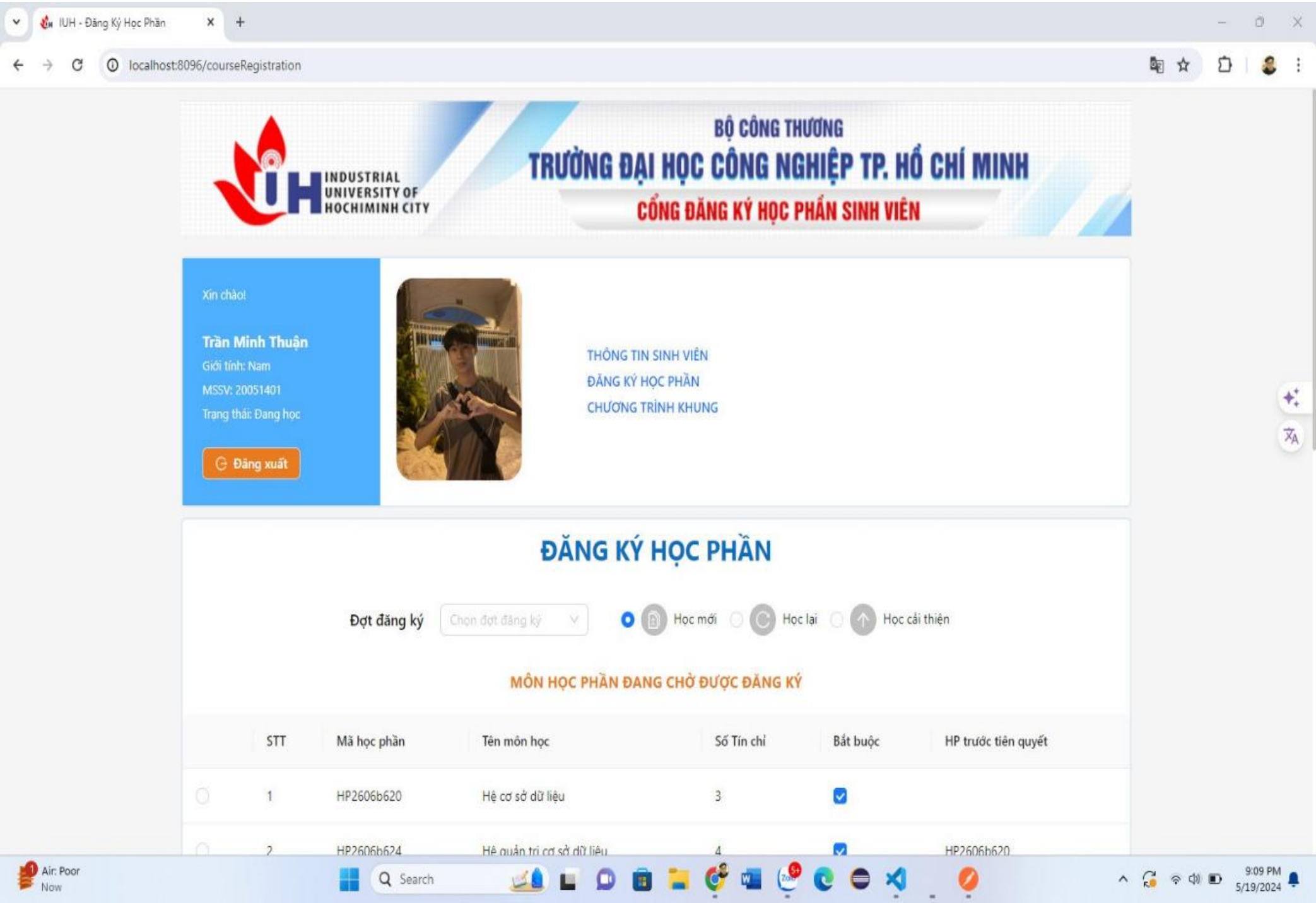
Giao diện trang cá nhân





# IV. HIỆN THỰC DỰ ÁN

## Các chức năng chính

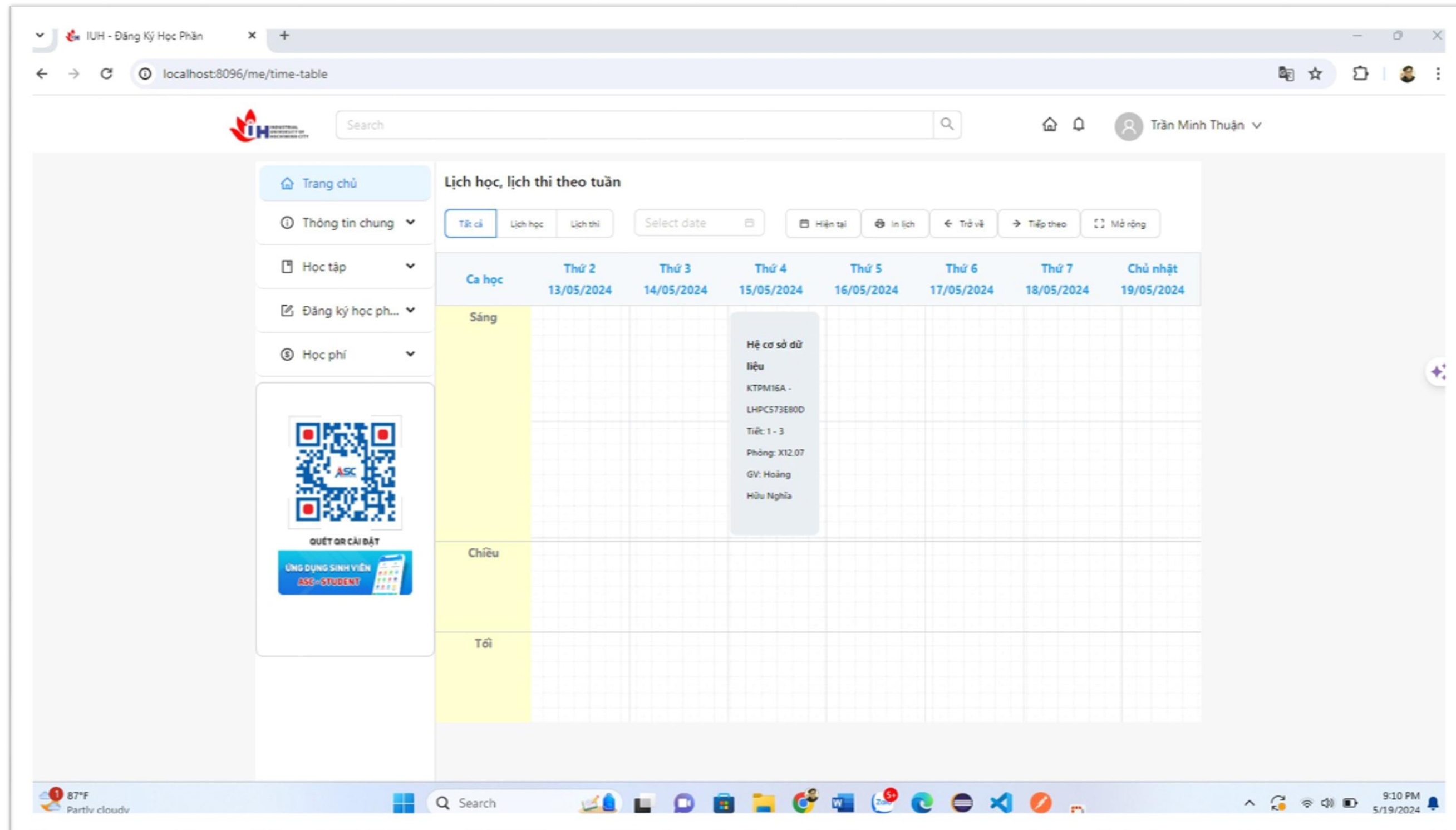


Giao diện đăng ký học phần



# IV. HIỆN THỰC DỰ ÁN

## Các chức năng chính

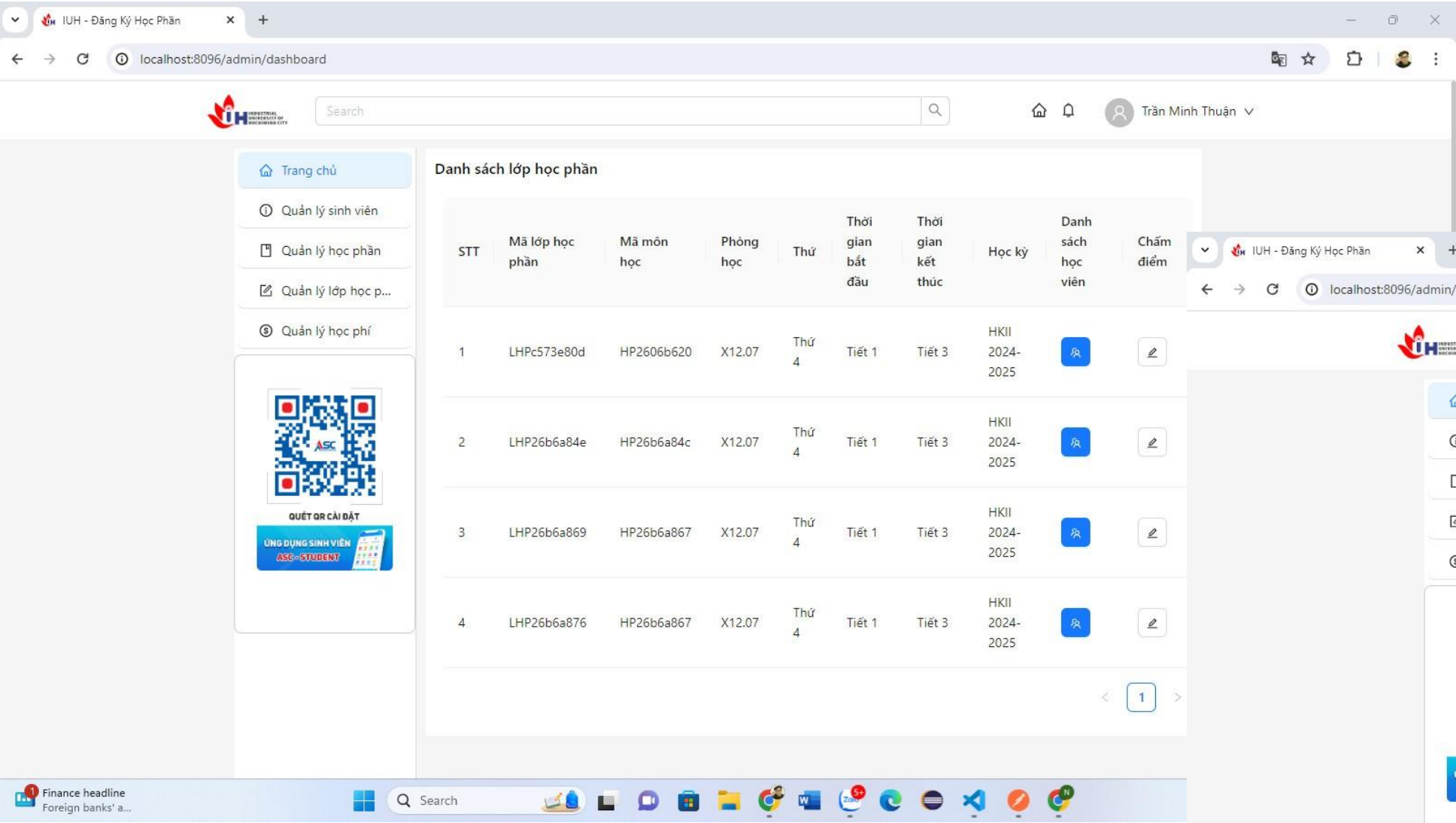


Giao diện xem lịch học



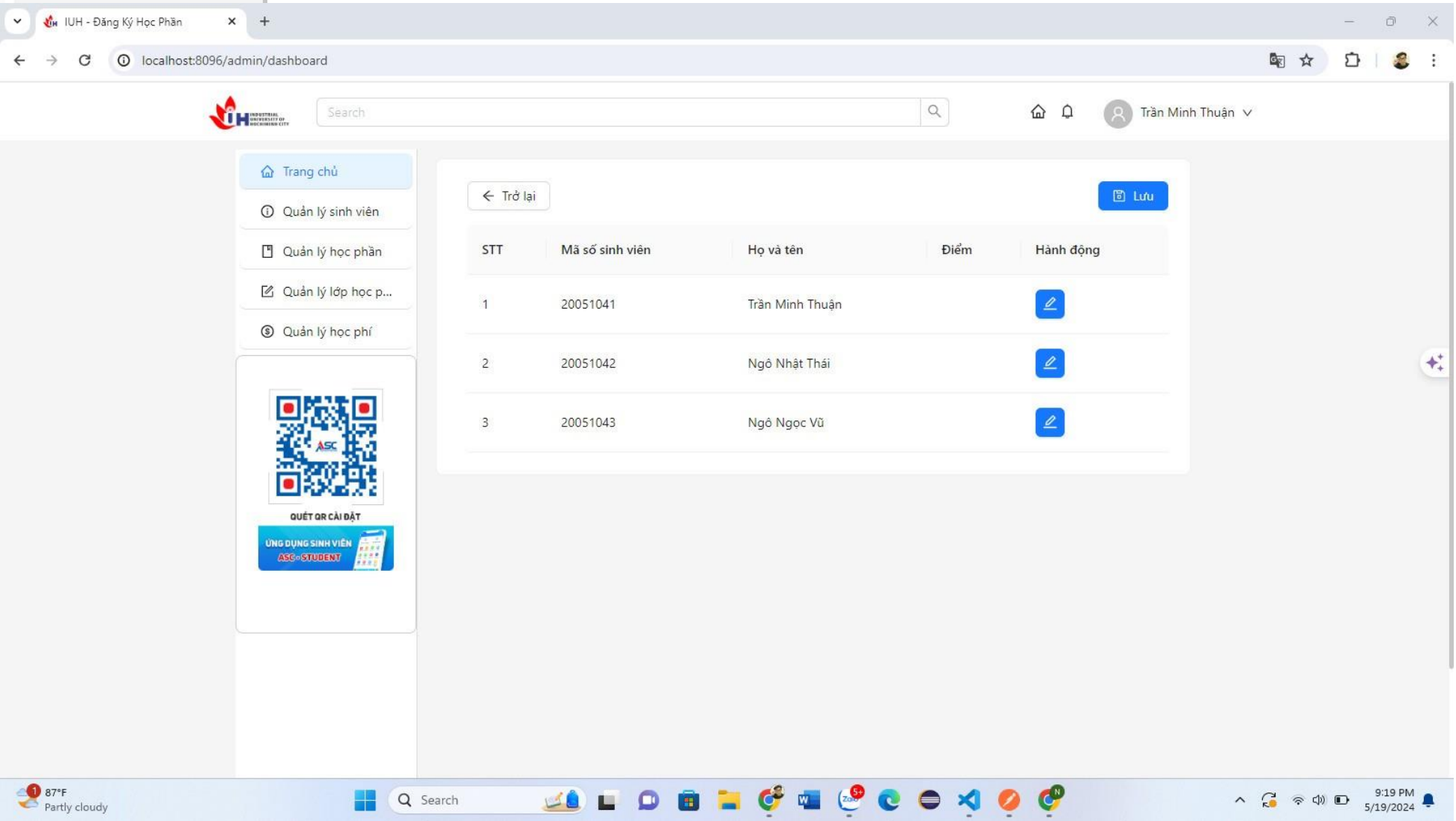
# IV. HIỆN THỰC DỰ ÁN

## Các chức năng chính



Giao diện quản lý lớp học

Giao diện quản lý sinh viên

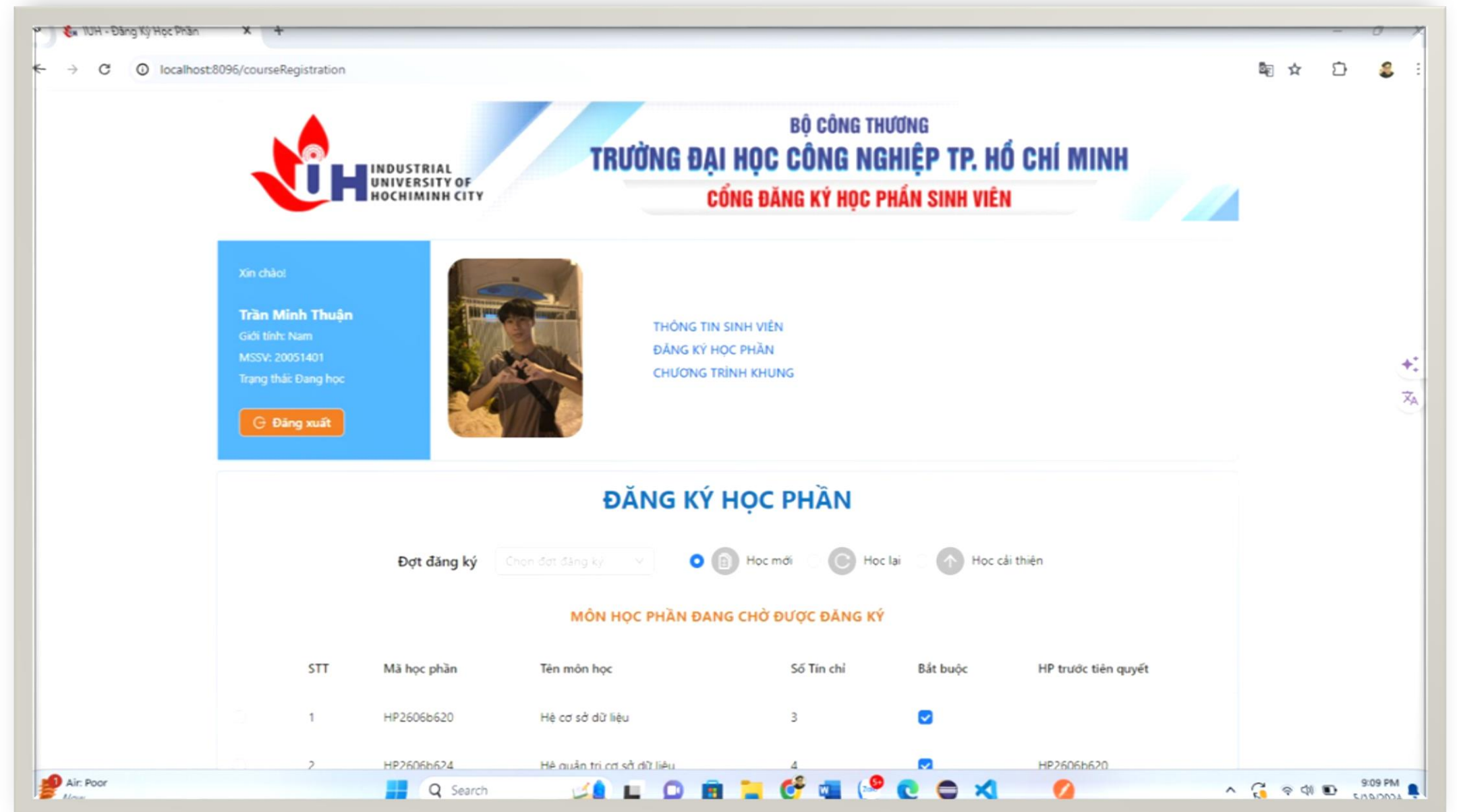




# VI. HOẠT ĐỘNG KẾT THÚC DỰ ÁN

## Các kết quả đạt được

- ❑ Đã hoàn thành đa số các chức năng theo đặc tả dự án, đảm bảo tính hoạt động và tính đúng đắn của web đăng kí học phần.
- ❑ Các chức năng chính như đăng nhập, đăng ký lớp học, xem lịch học đã được kiểm thử kỹ lưỡng và đạt được kết quả tốt.



# VI. HOẠT ĐỘNG KẾT THÚC DỰ ÁN

## Hướng phát triển dự án

---

- Tiếp tục cải thiện và bảo trì ứng dụng để đảm bảo tính ổn định và hiệu suất cao.
- Phát triển và triển khai các tính năng mới nhằm nâng cao trải nghiệm người dùng và tạo ra giá trị thêm cho ứng dụng.
- Tăng cường bảo mật và quản lý quyền riêng tư để đảm bảo an toàn thông tin của người dùng.
- Tăng cường hỗ trợ và tương tác với cộng đồng sinh viên để hiểu rõ hơn về nhu cầu và mong muốn của họ để điều chỉnh phát triển dựa trên phản hồi đó.
- Nghiên cứu và đề xuất các hướng xử lý để tăng hiệu năng web cũng như giảm thiểu hiện tượng giật lag khi quá nhiều sinh viên truy cập.



# Thank you!

---

