

**BÁO CÁO MÔN HỌC**

**Thực tập cơ sởĐề tài: Xây dựng web quản lí khóa học**

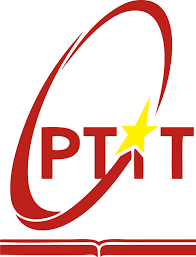
Họ tên: Lê Đức Thắng

Mã sinh viên: B21DCCN664

Nhóm thực tập cơ sở

Giảng viên: Thầy Nguyễn Hoàng Anh

***Hà Nội 2024***



**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Mục lục

[**Giới thiệu** 3](#_Toc165744598)

[**Chương 1 : Tổng quan** 5](#_Toc165744599)

[**I.Đặt vấn đề** 5](#_Toc165744600)

[**II.Mô tả vấn đề và cách giải quyết** 6](#_Toc165744601)

[***1.*** ***Phân tích*** 6](#_Toc165744602)

[***2.*** ***Chức năng chính*** 6](#_Toc165744603)

[**Chương 2: Cơ sở lí thuyết** 7](#_Toc165744604)

[**I.Hệ thống** 7](#_Toc165744605)

[**II, Ngôn ngữ lập trình Python** 7](#_Toc165744606)

[***1, Giới thiệu chung về ngôn ngữ lập trình Python*** 7](#_Toc165744607)

[***2, Các đặc trưng của Python*** 8](#_Toc165744608)

[***3, Framework Django*** 9](#_Toc165744609)

[**III. Cơ sở dữ liệu : MySQL Workbench** 11](#_Toc165744610)

[**IV, Giao diện** 12](#_Toc165744611)

[***1, HTML , CSS và JavaScript*** 12](#_Toc165744612)

[***2, Bootstrap*** 14](#_Toc165744613)

[***3, AJAX*** 15](#_Toc165744614)

[**Chương 3: Phân tích thiết kế và xây dựng hệ thống** 16](#_Toc165744615)

[**I, Xác định hệ thống** 16](#_Toc165744616)

[***1, Xác định yêu cầu*** 16](#_Toc165744617)

[***2, Phân tích yêu cầu*** 16](#_Toc165744618)

[**II, Phân tích hệ thống** 17](#_Toc165744619)

[***1, Xây dựng các thực thể*** 17](#_Toc165744620)

[***2, Chi tiết các thực thể*** 17](#_Toc165744621)

[**III, Xây dựng mô hình** 23](#_Toc165744622)

[***1, Thiết kế mô hình dữ liệu*** 23](#_Toc165744623)

[***2, Thiết kế cơ sở dữ liệu*** 23](#_Toc165744624)

[**Chương 4 : Chương trình** 24](#_Toc165744625)

[**I, Giới thiệu chương trình** 24](#_Toc165744626)

[***1, Giao diện đăng nhập*** 24](#_Toc165744627)

[***2, Giao diện chính của chương trình*** 25](#_Toc165744628)

[**II, Các chức năng** 27](#_Toc165744629)

[***1, Người dùng Admin*** 27](#_Toc165744630)

[***2, Người dùng Staff*** 35](#_Toc165744631)

[***3, Người dùng Student*** 39](#_Toc165744632)

[**Tổng kết** 41](#_Toc165744633)

**DANH MỤC HÌNH ẢNH**

[Hình 1 23](#_Toc165747746)

[Hình 2 23](#_Toc165747747)

[Hình 3 : Giao diện đăng nhập của web 24](#_Toc165747748)

[Hình 4: GIao diện đăng nhập thất bại 25](#_Toc165747749)

[Hình 5: Giao diện chính của tài khoản admin 25](#_Toc165747750)

[Hình 6 : Giao diện chính Staff 26](#_Toc165747751)

[Hình 7 : Giao diện Student 26](#_Toc165747752)

[Hình 8 27](#_Toc165747753)

[Hình 9 27](#_Toc165747754)

[Hình 10 : Thông báo username, email không hợp lệ 28](#_Toc165747755)

[Hình 11: Thông báo addstaff thành công 28](#_Toc165747756)

[Hình 12 29](#_Toc165747757)

[Hình 13 29](#_Toc165747758)

[Hình 14 30](#_Toc165747759)

[Hình 15 30](#_Toc165747760)

[Hình 16 31](#_Toc165747761)

[Hình 17 31](#_Toc165747762)

[Hình 18 31](#_Toc165747763)

[Hình 19 : Feedback của Student 32](#_Toc165747764)

[Hình 20 : Feedback của Staff 32](#_Toc165747765)

[Hình 21: Giao diện Reply feedback 33](#_Toc165747766)

[Hình 22: Đồng ý 33](#_Toc165747767)

[Hình 23 : Từ chối 34](#_Toc165747768)

[Hình 24 : Xem tình trạng điểm danh 34](#_Toc165747769)

[Hình 25 : Thông báo khi không có dữ liệu điểm danh 35](#_Toc165747770)

[Hình 26 35](#_Toc165747771)

[Hình 27 36](#_Toc165747772)

[Hình 28 36](#_Toc165747773)

[Hình 29 37](#_Toc165747774)

[Hình 30 37](#_Toc165747775)

[Hình 31: Giao diện cập nhật điểm số 38](#_Toc165747776)

[Hình 32: Thông báo khi sinh viên chưa có điểm 38](#_Toc165747777)

[Hình 33 39](#_Toc165747778)

[Hình 34 39](#_Toc165747779)

[Hình 35 39](#_Toc165747780)

[Hình 36 40](#_Toc165747781)

[Hình 37 40](#_Toc165747782)

# **Giới thiệu**

Hiện nay trong hầu hết các lĩnh vực, các tổ chức và doanh nghiệp đều đã áp dụng công nghệ vào quản lí công việc thường ngày để thay thế cho việc quản lí công việc một cách thủ công bằng sức người vốn dễ nhầm lẫn , độ chính xác và tin cậy không cao. Trong lĩnh vực giáo dục cũng không ngoại lệ, Học viện Công nghệ Bưu chính Viễn thông mỗi năm lại chào đón hàng nghìn sinh viên mới, do đó cần phải quản lí một cách nhanh và chính xác nhất. Xuất phát từ ý tưởng đó, bản thân em đã quyết định xây dựng một trang web *Quản lí sinh viên .* Đề tài tập trung xây dựng vào các chức năng cơ bản cho sinh viên, bao gồm : đăng kí khóa học, môn học, xem điểm, xin nghỉ,… Về hướng phát triển, cần cập nhật thêm nhiều chức năng liên kết với các phần mềm khác…

# **Chương 1: Tổng quan**

## **I.Đặt vấn đề**

Trong những năm gần đây, cách chương trình quản lí không còn xa lạ với mỗi chúng ta. Quản lí đã có ở hầu hết các lĩnh vực : giáo dục, kinh doanh, …Đặc biệt với lĩnh vực giáo dục và đào tạo, công tác quản lí sinh viên trong quá trình học tập và giảng dạy là vô cùng quan trọng, đòi hỏi sự chính xác cao. Ứng dụng công nghệ thông tin vào trong quản lí giúp con người giảm đi sự vất vả vì phải quản lí thủ công, qua đó còn rút ngắn thời gian và tăng độ chính xác, từ đó hiệu quả công việc và chất lượng sản phẩm đều được tăng cao.

Nắm bắt được xu thế đó, nên em đã tìm hiểu và nghiên cứu, qua đó xây dựng lên web quản lí khóa học của sinh viên nhằm hỗ trợ quản lí sinh viên trong các trung tâm đào tạo, giảng dạy

* Đối tượng nghiên cứu :
  + Hệ thống quản lí khóa học của các trung tâm giảng dạy
  + Các công cụ : MySQL Workbench để thiết kế cơ sở dữ liệu, Pycharm để lập trình Python
  + Giao diện : HTML, CSS, JavaScript
* Mục tiêu :
  + Quản lí sinh viên bao gồm điểm, các khóa học, môn học
  + Đưa ra các báo cáo, gửi phản hồi
* Phương pháp :
  + Khảo sát thực tế hệ thống
  + Lựa chọn công cụ lập trình và tiến hành xây dựng các module xử lí
  + Cài đặt và chạy chương trình trên hệ điều hành Windows
* Ý nghĩa :
  + Giúp hiểu thêm về nghiệp vụ quản lí sinh viên
  + Áp dụng kiến thức để thiết kế hệ thống
  + Hiểu thêm về ngôn ngữ lập trình Python

## **II.Mô tả vấn đề và cách giải quyết**

1. ***Phân tích***

Phòng quản lí sinh viên cần quản lí toàn bộ sinh viên đang theo học tại trường, mỗi sinh viên được gán một mã sinh viên duy nhất, thông tin của sinh viên bao gồm họ tên, ngày sinh,giới tính, quê quán, số điện thoại

Sinh viên có thể đăng kí nhiều khóa học khác nhau, trong mỗi khóa học thì có nhiều môn học khác nhau, các môn học có nhiều khung giờ giảng dạy khác nhau.

Nếu trong ngày học mà sinh viên có lí do không thể đi học, sinh viên có thể xin nghỉ bằng chức năng ở trên web

Vào mỗi cuối kì, điểm sẽ được nhập và được update trên web. Sinh viên có thể xem, nếu có thắc mắc về điểm sinh viên có thể gửi báo cáo lại với quản trị viên hoặc giáo viên của mình

1. ***Chức năng chính***

* Quản lí cập nhật thông tin của sinh viên
* Đăng kí khóa học, môn học cho sinh viên
* Gửi đơn xin nghỉ
* Gửi báo cáo đối với quản trị viên

# **Chương 2: Cơ sở lí thuyết**

## **I.Hệ thống**

Hệ thống là tập hợp các phần tử có mối quan hệ ràng buộc lẫn nhau cùng hoạt động với chung mục tiêu nào đó. Trong hoạt động sẽ có trao đổi vào ra với mới trường bên ngoài

Hệ thống quản lí : Là hệ thống có mục đích mang lại lợi nhuận hoặc lợi ích nào đó, đặc điểm của hệ thống quản lí là có sự tham gia của con người và trao đổi thông tin

## **II, Ngôn ngữ lập trình Python**

### ***1, Giới thiệu chung về ngôn ngữ lập trình Python***

Python là một ngôn ngữ lập trình thông dịch, đa mục đích và dễ học. Được phát triển bởi Guido van Rossum và ra mắt lần đầu vào năm 1991, Python nhanh chóng trở thành một trong những ngôn ngữ phổ biến nhất trên thế giới, được sử dụng trong nhiều lĩnh vực khác nhau từ phát triển phần mềm đến khoa học dữ liệu và trí tuệ nhân tạo.

Một số đặc điểm nổi bật của Python bao gồm cú pháp đơn giản, dễ đọc, dễ hiểu, cộng đồng lớn mạnh với nhiều thư viện và framework hỗ trợ, khả năng tích hợp tốt với các ngôn ngữ và nền tảng khác nhau, cũng như tính linh hoạt và mở rộng. Python cung cấp một môi trường phát triển phong phú với nhiều công cụ như IPython, Jupyter Notebook, và các IDE như PyCharm, VSCode.

Python được sử dụng rộng rãi trong các lĩnh vực như phát triển web (với các framework như Django, Flask), khoa học dữ liệu và máy học (với các thư viện như NumPy, Pandas, scikit-learn), automation, scripting, game development, và nhiều ứng dụng khác.

Với sự phát triển liên tục và sự hỗ trợ từ cộng đồng lập trình, Python tiếp tục là một trong những lựa chọn hàng đầu cho các nhà phát triển trên toàn thế giới.

### ***2, Các đặc trưng của Python***

Dưới đây là một số đặc trưng nổi bật của ngôn ngữ lập trình Python:

* Cú pháp đơn giản: Python có cú pháp rất đơn giản và dễ đọc, giúp người lập trình tập trung vào logic của chương trình thay vì phải lo lắng về cú pháp phức tạp.
* Thông dịch (Interpreted): Python là một ngôn ngữ thông dịch, điều này có nghĩa là mã nguồn được dịch và thực thi từng dòng lệnh một, không cần biên dịch trước như các ngôn ngữ lập trình khác như C++ hoặc Java.
* Đa mục đích: Python được thiết kế để phục vụ nhiều mục đích khác nhau. Từ phát triển web, ứng dụng desktop, đến xử lý dữ liệu và trí tuệ nhân tạo, Python có thể được sử dụng trong nhiều lĩnh vực khác nhau.
* Cộng đồng lớn mạnh: Python có một cộng đồng lập trình viên lớn mạnh và nhiệt tình. Điều này có nghĩa là có rất nhiều tài liệu, thư viện và framework được phát triển bởi cộng đồng để giúp người dùng giải quyết các vấn đề khác nhau.
* Thư viện và framework phong phú: Python có một số thư viện và framework mạnh mẽ cho nhiều mục đích khác nhau. Ví dụ, Django và Flask cho phát triển web, NumPy và Pandas cho xử lý dữ liệu, TensorFlow và PyTorch cho máy học và trí tuệ nhân tạo.
* Tích hợp tốt: Python có thể tích hợp dễ dàng với các ngôn ngữ lập trình khác như C/C++, Java và .NET, cho phép người lập trình sử dụng các thư viện và module có sẵn trong các ngôn ngữ này.
* Tính linh hoạt và mở rộng: Python cho phép người lập trình sử dụng nhiều phong cách lập trình khác nhau, từ lập trình hướng đối tượng đến lập trình hàm. Ngoài ra, Python cũng hỗ trợ mô-đun hóa tốt, giúp dễ dàng mở rộng chương trình.
* Hỗ trợ đa nền tảng: Python có sẵn trên nhiều nền tảng khác nhau như Windows, macOS và Linux, giúp đảm bảo tính di động và khả năng chạy trên nhiều loại hệ thống khác nhau.

### ***3, Framework Django***

Django là một framework phát triển web mạnh mẽ được viết bằng ngôn ngữ Python. Nó cung cấp một cơ sở hạ tầng để xây dựng các ứng dụng web phức tạp và hiệu quả, giúp các nhà phát triển tập trung vào việc xây dựng các tính năng của ứng dụng mà không phải lo lắng quá nhiều về các chi tiết kỹ thuật như xử lý HTTP, quản lý cơ sở dữ liệu, và bảo mật.

Django được phát triển bởi Django Software Foundation và đã có một lịch sử dài, với phiên bản đầu tiên được phát hành vào năm 2005. Tính đến thời điểm hiện tại, Django vẫn là một trong những framework phát triển web phổ biến nhất trong cộng đồng Python.

Một số đặc điểm của Django bao gồm:

* Mô hình MVC (Model-View-Controller): Django áp dụng mô hình MVC cho việc phát triển web, tuy nhiên, trong Django, nó được gọi là mô hình "MTV" (Model-Template-View). Điều này giúp tổ chức mã nguồn một cách dễ dàng và rõ ràng.
* ORM (Object-Relational Mapping): Django cung cấp một ORM mạnh mẽ, cho phép người phát triển tương tác với cơ sở dữ liệu thông qua các đối tượng Python thay vì sử dụng SQL trực tiếp.
* Admin Interface tự động: Django cung cấp một giao diện quản trị tự động cho các ứng dụng, giúp quản trị viên dễ dàng quản lý dữ liệu mà không cần phải viết mã nguồn.
* URL Routing: Django cho phép định tuyến các URL của ứng dụng một cách dễ dàng thông qua file `urls.py`, giúp phân chia và quản lý các đường dẫn của ứng dụng.
* Bảo mật tích hợp: Django tích hợp các công cụ bảo mật như CSRF (Cross-Site Request Forgery) protection và XSS (Cross-Site Scripting) protection để giúp bảo vệ ứng dụng khỏi các cuộc tấn công.
* Thư viện đa dạng: Django đi kèm với một loạt các thư viện và tiện ích tích hợp sẵn, giúp việc phát triển ứng dụng web nhanh chóng và hiệu quả.
* Templates mạnh mẽ: Django cung cấp hệ thống template mạnh mẽ, cho phép người phát triển tạo các giao diện người dùng linh hoạt và dễ dàng tái sử dụng. Templates của Django hỗ trợ kỹ thuật kế thừa, các biểu thức điều kiện và vòng lặp, giúp tạo ra các trang web động và linh hoạt.
* Phát triển nhanh chóng: Django cung cấp các công cụ và tiện ích giúp người phát triển xây dựng ứng dụng web nhanh chóng. Tính năng như migrations (chuyển đổi cơ sở dữ liệu), forms (biểu mẫu), và authentication (xác thực) giúp giảm thiểu thời gian và công sức cần thiết để phát triển ứng dụng.
* Phát triển API: Django không chỉ hỗ trợ việc phát triển ứng dụng web truyền thống mà còn cho phép xây dựng các API (Application Programming Interface) thông qua Django REST Framework. Điều này giúp Django trở thành một lựa chọn phổ biến cho việc xây dựng cả ứng dụng web và ứng dụng di động hoặc dịch vụ web.
* Khả năng mở rộng: Mặc dù Django cung cấp một số tính năng và tiện ích tích hợp sẵn, nhưng nó cũng cho phép mở rộng và tùy chỉnh thông qua việc sử dụng các ứng dụng (apps) và các thư viện bên ngoài. Điều này giúp người phát triển tạo ra các ứng dụng đa dạng và phức tạp theo nhu cầu cụ thể của dự án.
* Tài liệu phong phú: Django có một tài liệu phong phú và cộng đồng hỗ trợ mạnh mẽ. Tài liệu chính thức của Django cung cấp hướng dẫn chi tiết từ cơ bản đến nâng cao, cùng với ví dụ minh họa và hướng dẫn sử dụng các tính năng cụ thể.
* Cộng đồng lớn mạnh: Django có một cộng đồng người dùng và phát triển đông đảo, với hàng ngàn người dùng và các nhà phát triển đóng góp vào việc cải thiện và phát triển framework. Điều này đảm bảo rằng Django luôn cập nhật và có những giải pháp cho các vấn đề mới phát sinh.

## **III. Cơ sở dữ liệu : MySQL Workbench**

MySQL Workbench là một công cụ thiết kế cơ sở dữ liệu và quản lý cơ sở dữ liệu MySQL được phát triển bởi MySQL, hiện nay thuộc sở hữu của Oracle Corporation. Đây là một ứng dụng GUI (Giao diện người dùng đồ họa) mạnh mẽ, được thiết kế để hỗ trợ các nhà thiết kế cơ sở dữ liệu, nhà phát triển, và quản trị cơ sở dữ liệu trong việc thiết kế, quản lý và tối ưu hóa cơ sở dữ liệu MySQL của họ.

Dưới đây là một số đặc điểm của MySQL Workbench:

- Thiết kế Cơ sở dữ liệu:

* MySQL Workbench cung cấp một giao diện đồ họa để thiết kế cơ sở dữ liệu, cho phép người dùng tạo, chỉnh sửa và quản lý các bảng, quan hệ và các yếu tố cơ sở dữ liệu khác.
* Người dùng có thể vẽ biểu đồ quan hệ thực thể (ERD) để minh họa cấu trúc của cơ sở dữ liệu.

- SQL Development:

* MySQL Workbench cung cấp một môi trường phát triển SQL tích hợp, cho phép người dùng viết và thực thi các truy vấn SQL trực tiếp từ giao diện người dùng.
* Công cụ này cung cấp các tính năng như cú pháp kiểm tra, gợi ý mã, và hiển thị kết quả truy vấn một cách dễ hiểu.

- Quản trị cơ sở dữ liệu:

* MySQL Workbench cung cấp các công cụ quản trị cơ sở dữ liệu cho phép người dùng quản lý máy chủ MySQL, người dùng, và quyền truy cập.
* Các tính năng quản trị bao gồm sao lưu/ phục hồi cơ sở dữ liệu, quản lý biến thể, và giám sát hoạt động cơ sở dữ liệu.

- Tối ưu hóa hiệu suất:

* MySQL Workbench cung cấp các công cụ giúp người dùng tối ưu hóa hiệu suất cơ sở dữ liệu MySQL bằng cách giám sát và phân tích các câu lệnh SQL và các yếu tố khác ảnh hưởng đến hiệu suất.
* Công cụ này cung cấp gợi ý về cách cải thiện cấu trúc cơ sở dữ liệu, chỉ ra các chỉ số chậm, và giúp tối ưu hóa truy vấn SQL.

- Di cư cơ sở dữ liệu:

* MySQL Workbench cung cấp một trình hướng dẫn di cư cơ sở dữ liệu cho phép người dùng di chuyển dữ liệu từ các nguồn khác nhau sang MySQL.
* Công cụ này hỗ trợ di chuyển cấu trúc cơ sở dữ liệu, dữ liệu, và các đối tượng khác một cách dễ dàng.

- Tích hợp công cụ bên thứ ba:

* MySQL Workbench có thể tích hợp với các công cụ bên thứ ba như các hệ thống quản lý phiên bản (VCS) để quản lý mã nguồn SQL và cấu trúc cơ sở dữ liệu.

MySQL Workbench là một công cụ đa năng và mạnh mẽ cho việc thiết kế, phát triển và quản lý cơ sở dữ liệu MySQL, giúp tăng cường hiệu suất và tiết kiệm thời gian cho các nhà phát triển và quản trị viên cơ sở dữ liệu.

## **IV, Giao diện**

### ***1, HTML , CSS và JavaScript***

HTML, CSS và JavaScript là ba thành phần cơ bản và quan trọng nhất trong phát triển web hiện đại. Dưới đây là một số điểm chính về mỗi thành phần:

HTML (Hypertext Markup Language):

* HTML là ngôn ngữ định dạng cấu trúc cho các trang web. Nó được sử dụng để xác định và tổ chức nội dung của trang web, bao gồm văn bản, hình ảnh, video, liên kết, biểu mẫu, và nhiều thành phần khác.
* HTML sử dụng các thẻ (tags) để đánh dấu các phần của trang web. Ví dụ, `<p>` là thẻ để đánh dấu một đoạn văn bản, `<img>` là thẻ để đánh dấu một hình ảnh, và `<a>` là thẻ để tạo ra một liên kết.
* HTML cung cấp một cấu trúc cơ bản cho trang web, nhưng nó không quản lý về cách trang web được hiển thị hoặc trông như thế nào. Để điều chỉnh về giao diện và kiểu dáng của trang web, chúng ta cần sử dụng CSS.

CSS (Cascading Style Sheets):

* CSS là ngôn ngữ được sử dụng để mô tả cách các phần tử HTML được hiển thị trên trình duyệt web. Nó định dạng và trang trí các phần tử HTML bằng cách áp dụng các quy tắc kiểu dáng như màu sắc, kích thước, font chữ, đường viền, và vị trí.
* CSS giúp tách biệt nội dung của trang web (được định dạng bởi HTML) và kiểu dáng của trang web (được định dạng bởi CSS), làm cho mã nguồn trở nên dễ đọc hơn, dễ bảo trì hơn, và tăng khả năng tái sử dụng.
* CSS có thể được áp dụng vào các phần tử HTML thông qua các cách khác nhau, bao gồm thông qua các lớp (classes), các định danh (IDs), hoặc các phần tử HTML trực tiếp.

JavaScript:

* JavaScript là một ngôn ngữ lập trình được sử dụng để thêm tính năng tương tác và động vào trang web.
* JavaScript cho phép xử lý sự kiện như nhấp chuột, nhập liệu từ bàn phím, hoặc thời gian, và phản hồi lại các sự kiện đó một cách động.
* JavaScript cũng được sử dụng để thay đổi nội dung của trang web và kiểu dáng của các phần tử trên trang, tạo ra các hiệu ứng động và tương tác người dùng.

Khi kết hợp cả ba thành phần này, ta có thể tạo ra các trang web đa dạng, đẹp mắt và tương tác. HTML xác định cấu trúc và nội dung của trang, CSS kiểm soát kiểu dáng và trang trí của trang, trong khi JavaScript thêm tính năng tương tác và động vào trang web. Sự kết hợp của cả ba thành phần này làm cho việc phát triển trang web trở nên linh hoạt và mạnh mẽ.

### ***2, Bootstrap***

Bootstrap là một framework CSS và JavaScript mã nguồn mở được phát triển bởi nhóm phát triển tại Twitter. Nó cung cấp một bộ công cụ để thiết kế giao diện web đẹp mắt, linh hoạt và hiệu quả. Bootstrap giúp tăng tốc quá trình phát triển web bằng cách cung cấp các thành phần giao diện sẵn có, lớp CSS được chuẩn hóa, và các kịch bản JavaScript để tạo ra các trang web phản hồi và dễ sử dụng.

Dưới đây là một số đặc điểm chính của Bootstrap:

* Grid System: Bootstrap cung cấp một hệ thống lưới linh hoạt dựa trên cột, giúp phân chia giao diện web thành các cột và hàng. Điều này giúp tạo ra giao diện đáp ứng, tự động điều chỉnh kích thước và bố cục của trang web trên các thiết bị khác nhau.
* CSS Components: Bootstrap đi kèm với một loạt các thành phần giao diện như nút, biểu mẫu, thanh điều hướng, thẻ, bảng, và nhiều hơn nữa. Các thành phần này được thiết kế để giúp xây dựng giao diện web một cách nhanh chóng và dễ dàng.
* Responsive Design: Bootstrap được xây dựng với sự chú trọng đặc biệt vào việc tạo ra các trang web phản hồi, tức là các trang web có thể hiển thị một cách tốt trên các thiết bị khác nhau như máy tính, điện thoại thông minh và máy tính bảng.
* Theming: Bootstrap cho phép người dùng tùy chỉnh và thay đổi giao diện của trang web thông qua việc áp dụng các chủ đề (themes) có sẵn hoặc tạo ra các chủ đề tùy chỉnh.
* JavaScript Plugins: Bootstrap đi kèm với một số plugin JavaScript như carousel, modal, tab, tooltip, và popover. Những plugin này giúp thêm các tính năng tương tác và hiệu ứng động vào trang web.

Bootstrap đã trở thành một trong những framework phổ biến nhất cho phát triển giao diện web, được sử dụng rộng rãi trong cộng đồng phát triển web để tạo ra các trang web một cách nhanh chóng và chuyên nghiệp.

### ***3, AJAX***

Ajax (Asynchronous JavaScript and XML) là một kỹ thuật lập trình web cho phép trao đổi dữ liệu giữa trình duyệt và máy chủ một cách bất đồng bộ, mà không cần làm mới lại toàn bộ trang web. Điều này tạo ra trải nghiệm người dùng mượt mà và tương tác hơn, vì trang web có thể cập nhật một phần của nội dung mà không gây ra sự gián đoạn cho người dùng.

Dưới đây là các bước và chi tiết về cách kỹ thuật Ajax hoạt động:

Tạo yêu cầu XMLHttpRequest (XHR):

* Ajax sử dụng đối tượng XMLHttpRequest để gửi yêu cầu đến máy chủ và nhận phản hồi từ máy chủ một cách bất đồng bộ. Đối tượng XMLHttpRequest cho phép gửi yêu cầu HTTP (như GET, POST) đến một URL cụ thể.

Gửi yêu cầu đến máy chủ:

* Khi người dùng thực hiện một hành động trên trang web (ví dụ: nhấp vào một nút), JavaScript sẽ tạo một yêu cầu XMLHttpRequest và gửi nó đến máy chủ. Yêu cầu này thường là một yêu cầu HTTP GET hoặc POST.

Xử lý phản hồi từ máy chủ:

* Khi máy chủ nhận được yêu cầu, nó sẽ xử lý và trả về phản hồi. Phản hồi này có thể là một đoạn văn bản, XML, JSON hoặc bất kỳ dạng dữ liệu nào khác.
* JavaScript sẽ xử lý phản hồi từ máy chủ và cập nhật nội dung của trang web một cách động thông qua manipulation của DOM.

Cập nhật giao diện người dùng:

* Sau khi nhận được phản hồi từ máy chủ, JavaScript sẽ sử dụng dữ liệu đó để cập nhật nội dung của trang web một cách động. Điều này thường được thực hiện bằng cách thay đổi nội dung của một phần tử HTML hoặc thêm mới các phần tử vào trang.

Ajax là một công nghệ mạnh mẽ trong phát triển web, cho phép tạo ra các ứng dụng web động, tương tác và phản hồi mà không cần tải lại toàn bộ trang.

# **Chương 3: Phân tích thiết kế và xây dựng hệ thống**

## **I, Xác định hệ thống**

### ***1, Xác định yêu cầu***

* Đối tượng phục vụ: Học viên, giảng viên các trung tâm đào tạo
* Đối tượng quản lý: Sinh viên, Giảng viên.
* Yêu cầu đối với hệ thống:
  + Quản lý:
    - Quản lý giảng viên
    - Quản lý sinh viên
    - Quản lý môn học và khóa học
  + Xem thống kê (số người tham gia,...)
  + Gửi thông báo, phản hồi

### ***2, Phân tích yêu cầu***

* Quản lý giảng viên: Để sử dụng được phần mềm giảng viên sẽ được cấp một tài khoản bao gồm có Username và Password để đăng nhập hệ thống/
* Quản lý sinh viên: Sinh viên sẽ được cấp một tài khoản bao gồm có Username và Password để đăng nhập hệ thống
* Quản lý khóa học, môn học: Hỗ trợ giảng viên quản lý được danh sách các khóa học, các môn học của từng khóa học
* Xem thống kê: Thống kê số lượng sinh viên của 1 course
* Gửi phản hồi: Giảng viên và Sinh viên có thể gửi feedback đến admin - tức người quản trị hệ thống để đóng góp ý kiến và admin có thể phản hồi lại. Sinh viên có thể gửi thông báo đến giảng viên (ví dụ về xin nghỉ học ), giảng viên có thể xem và phản hồi lại

## **II, Phân tích hệ thống**

### ***1, Xây dựng các thực thể***

* Thực thể ADMIN (admin): ID, email, password, creat\_at, update\_at
* Thực thể STAFF (giảng viên): ID, email, password, address, creat\_at, update\_at
* Thực thể STUDENT (sinh viên): ID, email, name, gender, password, course ID, creat\_at, update\_at
* Thực thể COURSE (khóa học): ID, course name, creat\_at, update\_at
* Thực thể SUBJECT (môn học): ID, subject name, course ID, staff ID, creat\_at, update\_at
* Thực thể NOTIFICATION (thông báo): ID, staff ID, course ID, message, creat\_at, update\_at
* Thực thể FEEDBACK (phản hồi): ID, student\_staff\_ID, message, creat\_at, update\_at
* Thực thể ATTENDANCE (người tham gia): ID, student ID, attendance\_ID, status, creat\_at, update\_at

### ***2, Chi tiết các thực thể***

* ADMIN:

Class AdminHOD(models.Model):  
 id = models.AutoField(primary\_key=True)  
 admin = models.OneToOneField(CustomUser, on\_delete=models.CASCADE)  
 created\_at = models.DateTimeField(auto\_now\_add=True)  
 updated\_at = models.DateTimeField(auto\_now\_add=True)  
 objects = models.Manager()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thực thể | Kiểu | Khóa chính | Khóa ngoại | Diễn giải |
| ID | AutoField | x |  | Mã admin |
| CREAT\_AT | DatetimeField |  |  | Ngày tạo |
| UPDATE\_AT | DatetimeField |  |  | Ngày sửa |

*Bảng 1*

* STAFF:

class Staffs(models.Model):  
 id = models.AutoField(primary\_key=True)  
 admin = models.OneToOneField(CustomUser, on\_delete=models.CASCADE)  
 address = models.TextField()  
 created\_at = models.DateTimeField(auto\_now\_add=True)  
 updated\_at = models.DateTimeField(auto\_now\_add=True)  
 object = models.Manager()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thực thể | Kiểu | Khóa chính | Khóa phụ | Diễn giải |
| ID | AutoField | x |  | Mã nhân viên (giảng viên) |
| ADDRESS | TextField |  |  | Địa chỉ |
| CREATE\_AT | DatetimeField |  |  | Ngày tạo |
| UPDATE\_AT | DatetimeField |  |  | Ngày sửa |

*Bảng 2*

* STUDENT:

class Students(models.Model):  
 id = models.AutoField(primary\_key=True)  
 gender = models.CharField(max\_length=255)  
 profile\_pic=models.FileField()  
 address=models.TextField()  
 course\_id = models.ForeignKey(Courses, on\_delete=models.DO\_NOTHING)  
 session\_start\_year = models.DateField()  
 session\_end\_year = models.DateField()  
 created\_at = models.DateTimeField(auto\_now\_add=True)  
 updated\_at = models.DateTimeField(auto\_now\_add=True)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thực thể | Kiểu | Khóa chính | Khóa phụ | Diễn giải |
| ID | AutoField | x |  | Mã sinh viên |
| GENDER | CharField |  |  | Giới tính |
| PROFILE\_PIC | FileField |  |  | ảnh profile cá nhând |
| ADDRESS | TextField |  |  | Địa chỉ |
| COURSE\_ID | AutoField |  | x | Mã môn học |
| SESSION\_START\_YEAR | DateField |  |  | Phiên bắt đầu |
| SESSION\_END\_YEAR | DateField |  |  | Phiên kết thúc |
| CREATE\_AT | DatetimeField |  |  | Ngày tạo |
| UPDATE\_AT | DatetimeField |  |  | Ngày sửa |

Bảng 3

* COURSE:

class Courses(models.Model):  
 id=models.AutoField(primary\_key=True)  
 course\_name=models.CharField(max\_length=255)  
 created\_at=models.DateTimeField(auto\_now\_add=True)  
 updated\_at=models.DateTimeField(auto\_now\_add=True)  
 object = models.Manager()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thực thể | Kiểu | Khóa chính | Khóa phụ | Diễn giải |
| ID | AutoField | x |  | Mã khóa học |
| COURSE\_ID | AutoField |  | x | Mã khóa học |
| CREATE\_AT | DatetimeField |  |  | Ngày tạom |
| UPDATE\_AT | DatetimeField |  |  | Ngày sửa |

Bảng 4

* SUBJECT:

class Subjects(models.Model):  
 id=models.AutoField(primary\_key=True)  
 admin = models.OneToOneField(CustomUser, on\_delete=models.CASCADE)  
 subject\_name=models.CharField(max\_length=255)  
 course\_id=models.ForeignKey(Courses,on\_delete=models.CASCADE)  
 staff\_id = models.ForeignKey(Staffs, on\_delete=models.CASCADE)  
 created\_at = models.DateTimeField(auto\_now\_add=True)  
 updated\_at = models.DateTimeField(auto\_now\_add=True)  
 object = models.Manager()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thực thể | Kiểu | Khóa chính | Khóa phụ | Diễn giải |
| ID | AutoField | x |  | Mã môn họct |
| SUBJECT\_NAME | CharField |  |  | Tên môn học |
| COURSE\_ID | AutoField |  | x | Mã khóa học |
| STAFF\_ID | AutoField |  | x | Mã nhân viên (giảng viên) |
| CREATE\_AT | DatetimeField |  |  | Ngày tạo |
| UPDATE\_AT | DatetimeField |  |  | Ngày sửa |

Bảng 5

* FEEDBACK (staff / student):

class FeedBackStudent(models.Model):  
 id = models.AutoField(primary\_key=True)  
 student\_id = models.ForeignKey(Students, on\_delete=models.CASCADE)  
 feedback = models.TextField()  
 feedback\_reply = models.TextField()  
 created\_at = models.DateTimeField(auto\_now\_add=True)  
 updated\_at = models.DateTimeField(auto\_now\_add=True)  
 objects = models.Manager()

class FeedBackStaffs(models.Model):  
 id = models.AutoField(primary\_key=True)  
 staff\_id = models.ForeignKey(Staffs, on\_delete=models.CASCADE)  
 feedback = models.TextField()  
 feedback\_reply=models.TextField()  
 created\_at = models.DateTimeField(auto\_now\_add=True)  
 updated\_at = models.DateTimeField(auto\_now\_add=True)  
 objects = models.Manager()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thực thể | Kiểu | Khóa chính | Khóa phụ | Diễn giải |
| ID | AutoField | x |  | Mã phản hồi |
| STAFF\_ID | AutoField |  | x | Mã nhân viên (giảng viên) |
| STUDENT\_ID | AutoField |  | x | Mã sinh viên |
| FEEDBACK | TextField |  |  | Phản hồi |
| FEEBACK\_REPLY | TextField |  |  | Trả lời phản hồi |
| CREATE\_AT | DatetimeField |  |  | Ngày tạo |
| UPDATE\_AT | DatetimeField |  |  | Ngày sửa |

Bảng 6

* ATTENDANCE (staff / student):

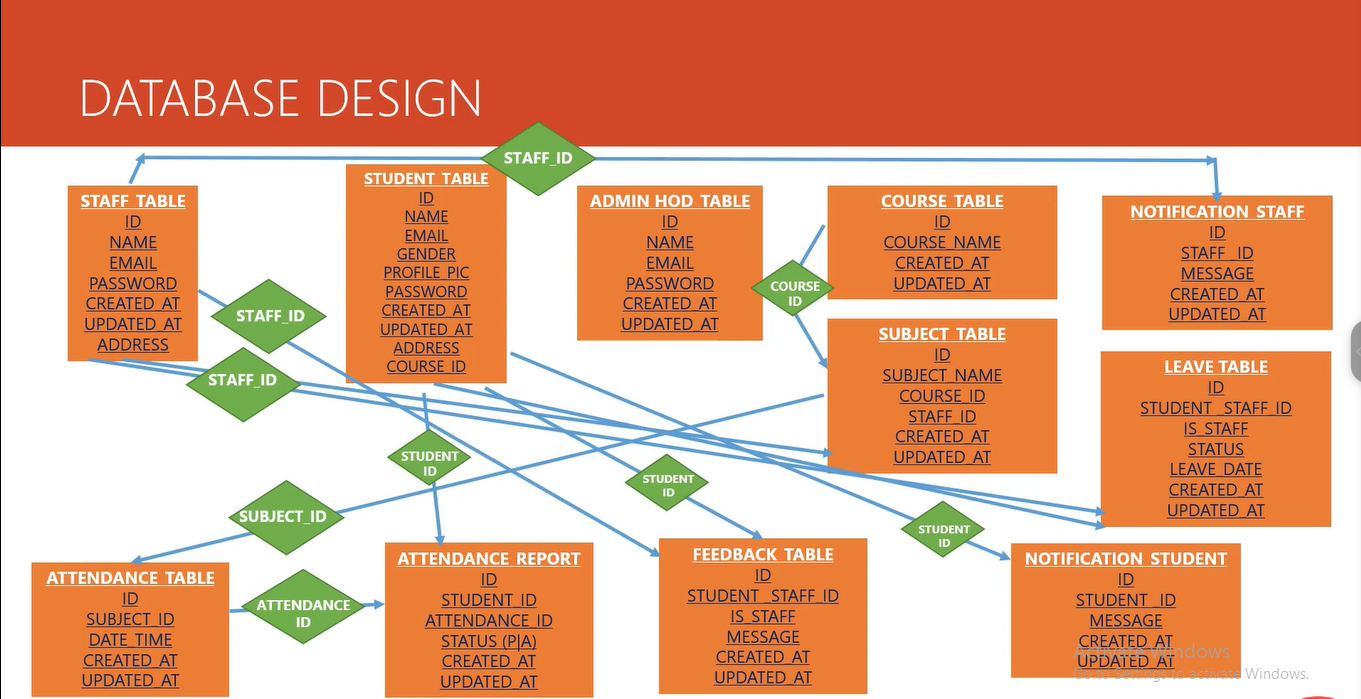
class Attendance(models.Model):  
 id = models.AutoField(primary\_key=True)  
 subject\_id = models.ForeignKey(Subjects, on\_delete=models.DO\_NOTHING)  
 attendance\_date = models.DateTimeField(auto\_now\_add=True)  
 created\_at = models.DateTimeField(auto\_now\_add=True)  
 updated\_at = models.DateTimeField(auto\_now\_add=True)  
 objects = models.Manager()  
  
class AttendanceReport(models.Model):  
 id=models.AutoField(primary\_key=True)  
 student\_id=models.ForeignKey(Students,on\_delete=models.DO\_NOTHING)  
 attendance\_id=models.ForeignKey(Attendance,on\_delete=models.CASCADE)  
 status=models.BooleanField(default=False)  
 created\_at=models.DateTimeField(auto\_now\_add=True)  
 updated\_at=models.DateTimeField(auto\_now\_add=True)  
 objects=models.Manager()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thực thể | Kiểu | Khóa chính | Khóa phụ | Diễn giải |
| ID | AutoField | x |  | Mã tham gia |
| SUBJECT\_ID | AutoField |  | x | Mã môn học |
| STUDENT\_ID | AutoField |  | x | Mã sinh viên |
| STATUS | BooleanField |  |  | Trạng thái |
| CREATE\_AT | DatetimeField |  |  | Ngày tạo |
| UPDATE\_AT | DatetimeField |  |  | Ngày sửa |

Bảng 7

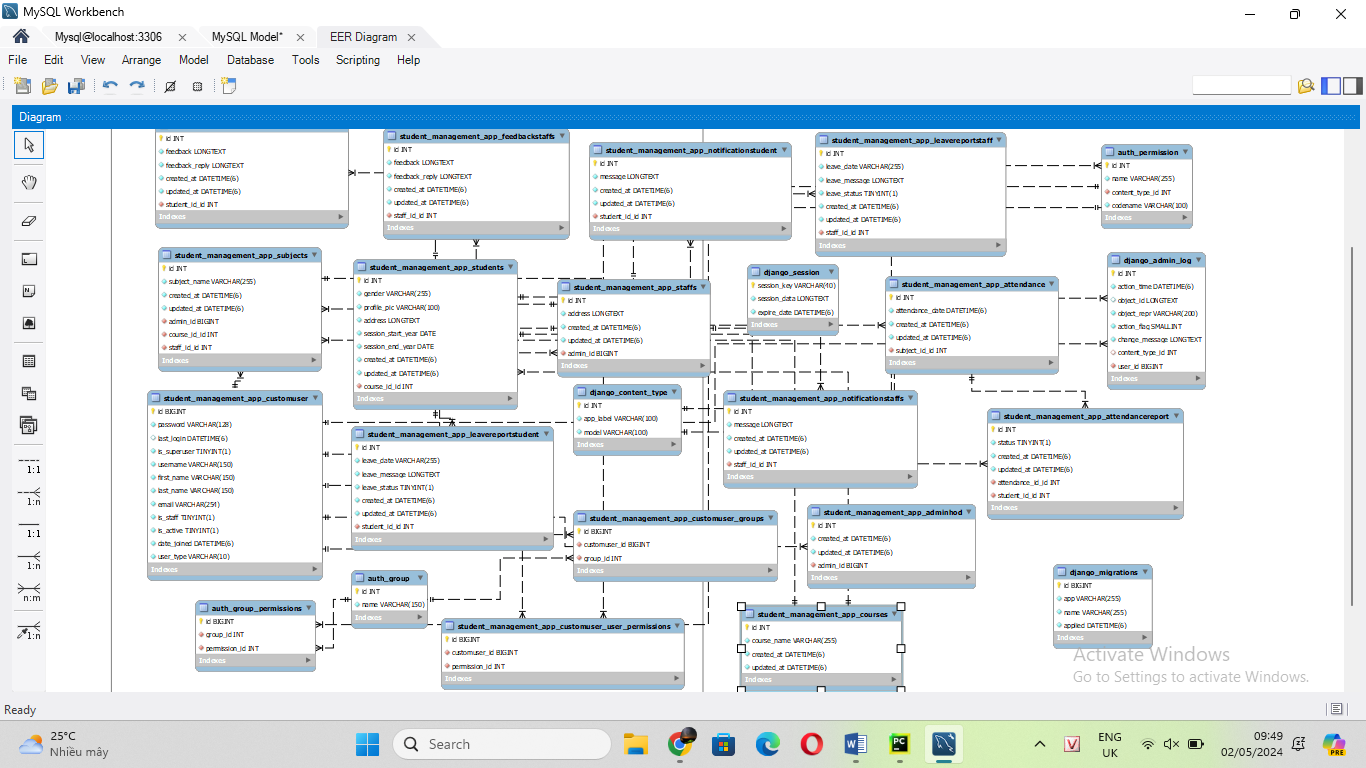
## **III, Xây dựng mô hình**

### ***1, Thiết kế mô hình dữ liệu***



Hình 1

### ***2, Thiết kế cơ sở dữ liệu***



Hình 2

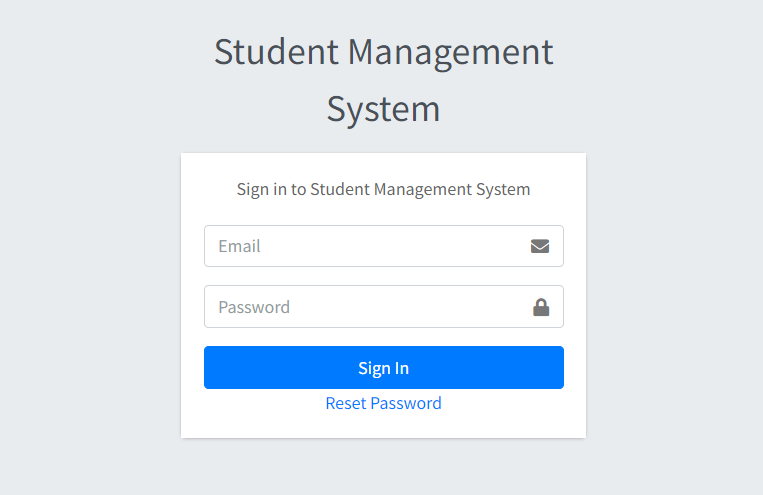
# **Chương 4 : Chương trình**

## **I, Giới thiệu chương trình**

Do website chưa đi vào hoạt động nên sau khi chạy chương trình ta truy cập từ ip localhost <http://127.0.0.1:7000/>

### ***1, Giao diện đăng nhập***

Đăng nhập với email và password đã được cấp tùy theo loại tài khoản người dùng nếu đúng sẽ chuyển tới giao diện chính của chương trình ngược lại sẽ hiện ra thông báo “Invalid Login Detail” và tiến hành nhập thông tin đăng nhập lại như các hình dưới đây



Hình 3 : Giao diện đăng nhập của web

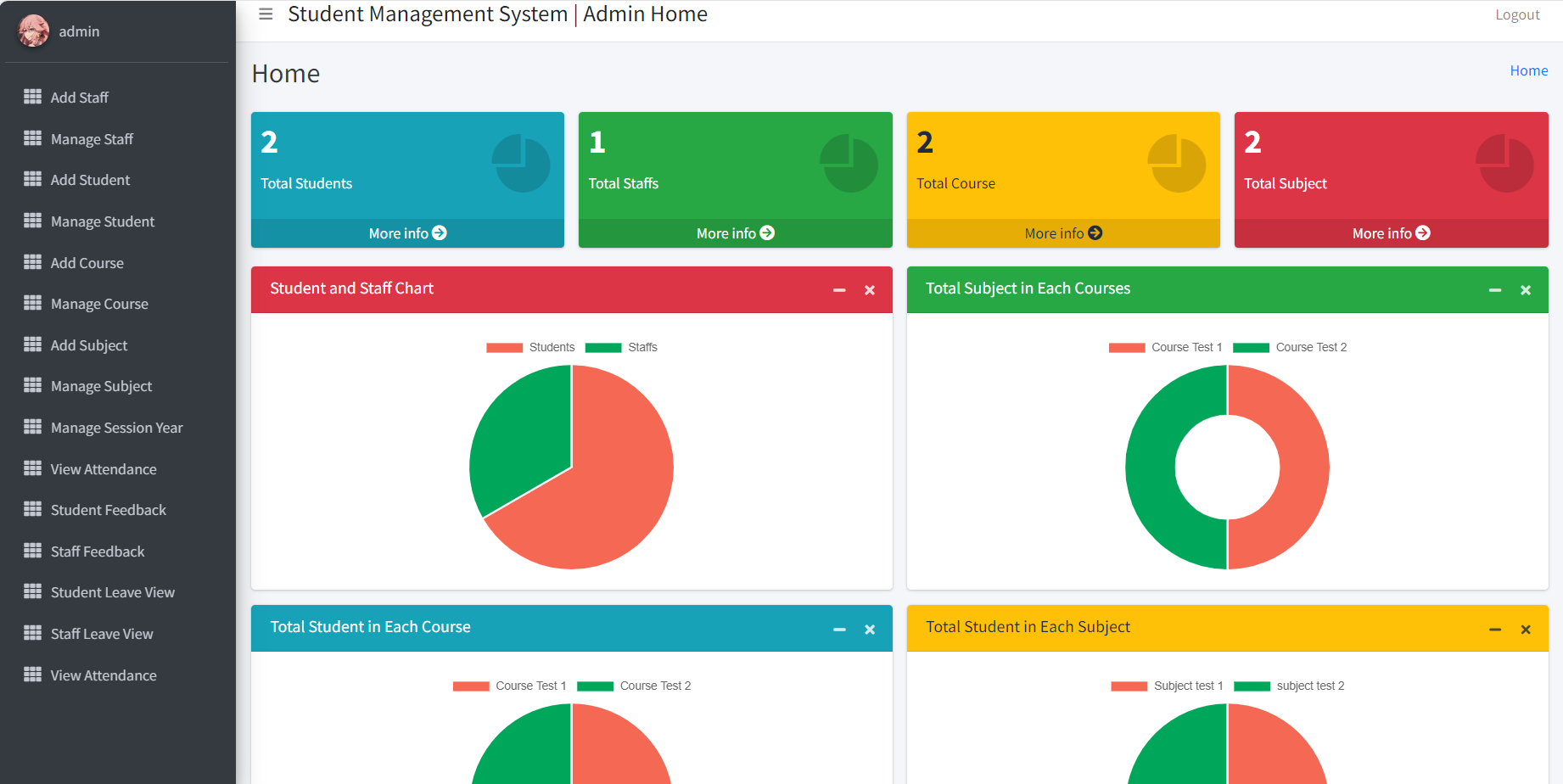


Hình 4: GIao diện đăng nhập thất bại

### ***2, Giao diện chính của chương trình***

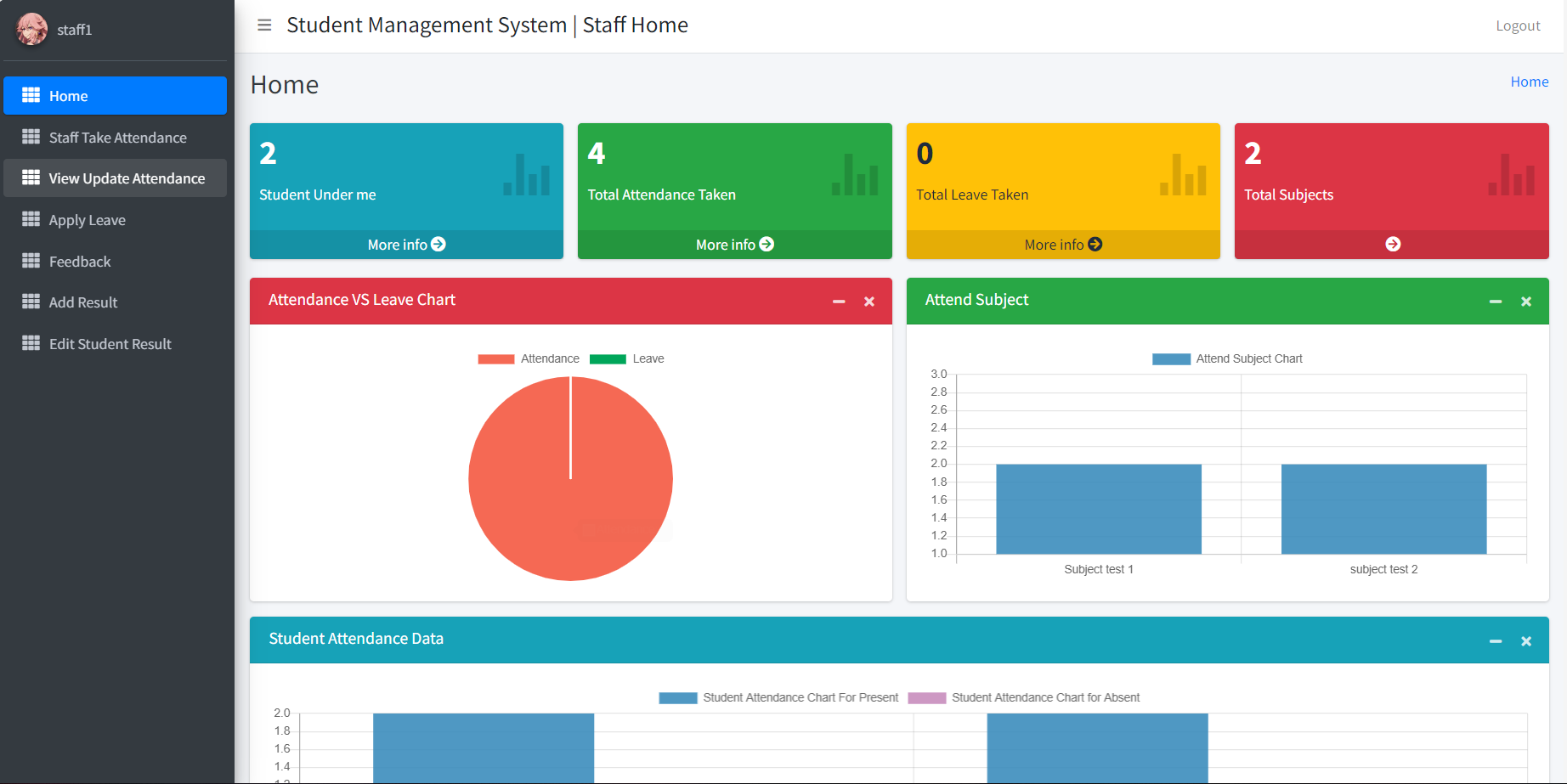
Sau khi đăng nhập tùy theo loại tài khoản người dùng sẽ dẫn đến trang giao diện khác nhau

* Admin : gồm một số thống kê các dữ liệu cùng với danh mục các chức năng chính của chương trình



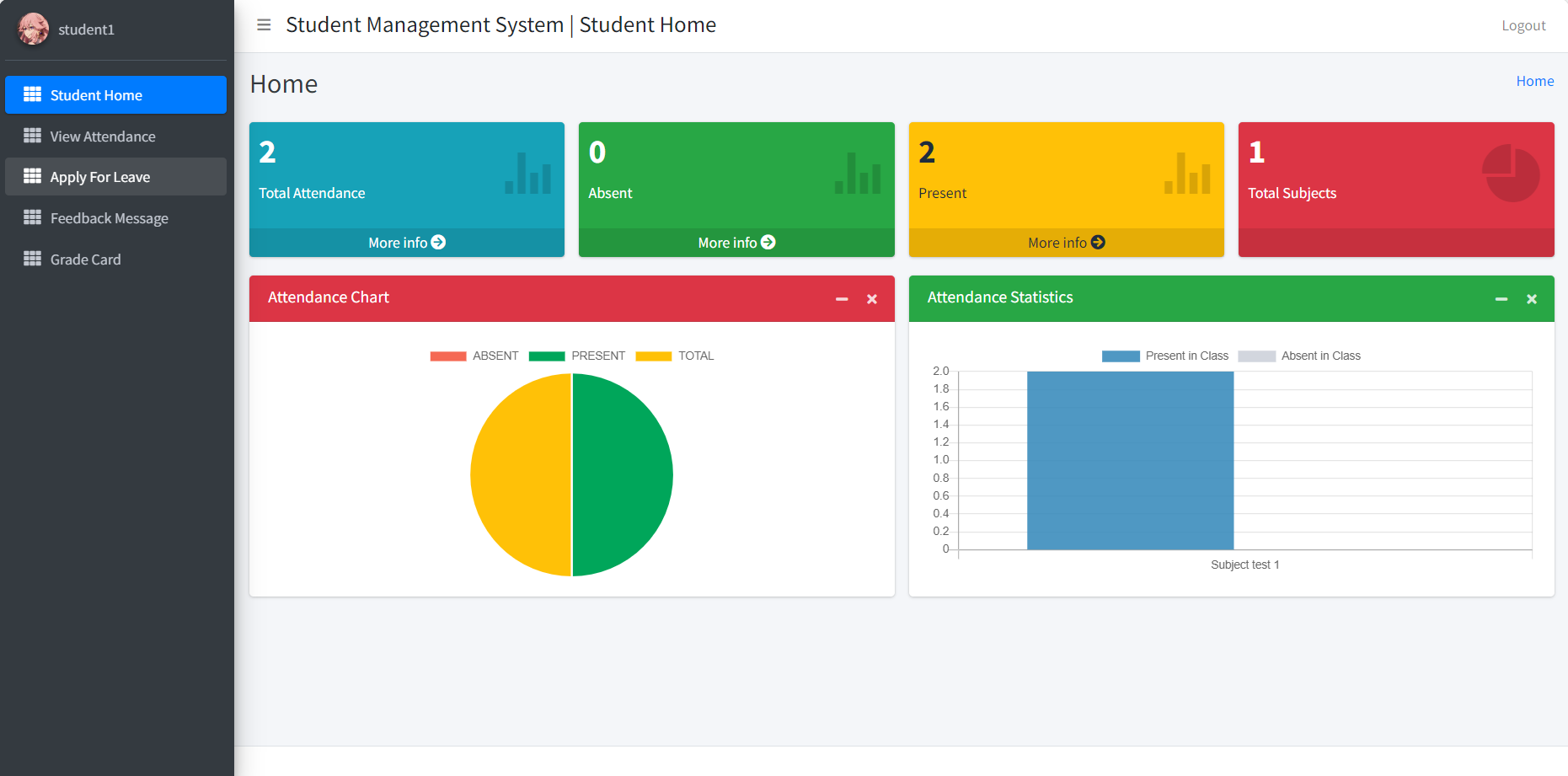
Hình 5: Giao diện chính của tài khoản admin

* Staff : Giao diện tương tự với của người dùng Admin với một số thống kê và danh mục chức năng



Hình 6 : Giao diện chính Staff

* Student :



Hình 7 : Giao diện Student

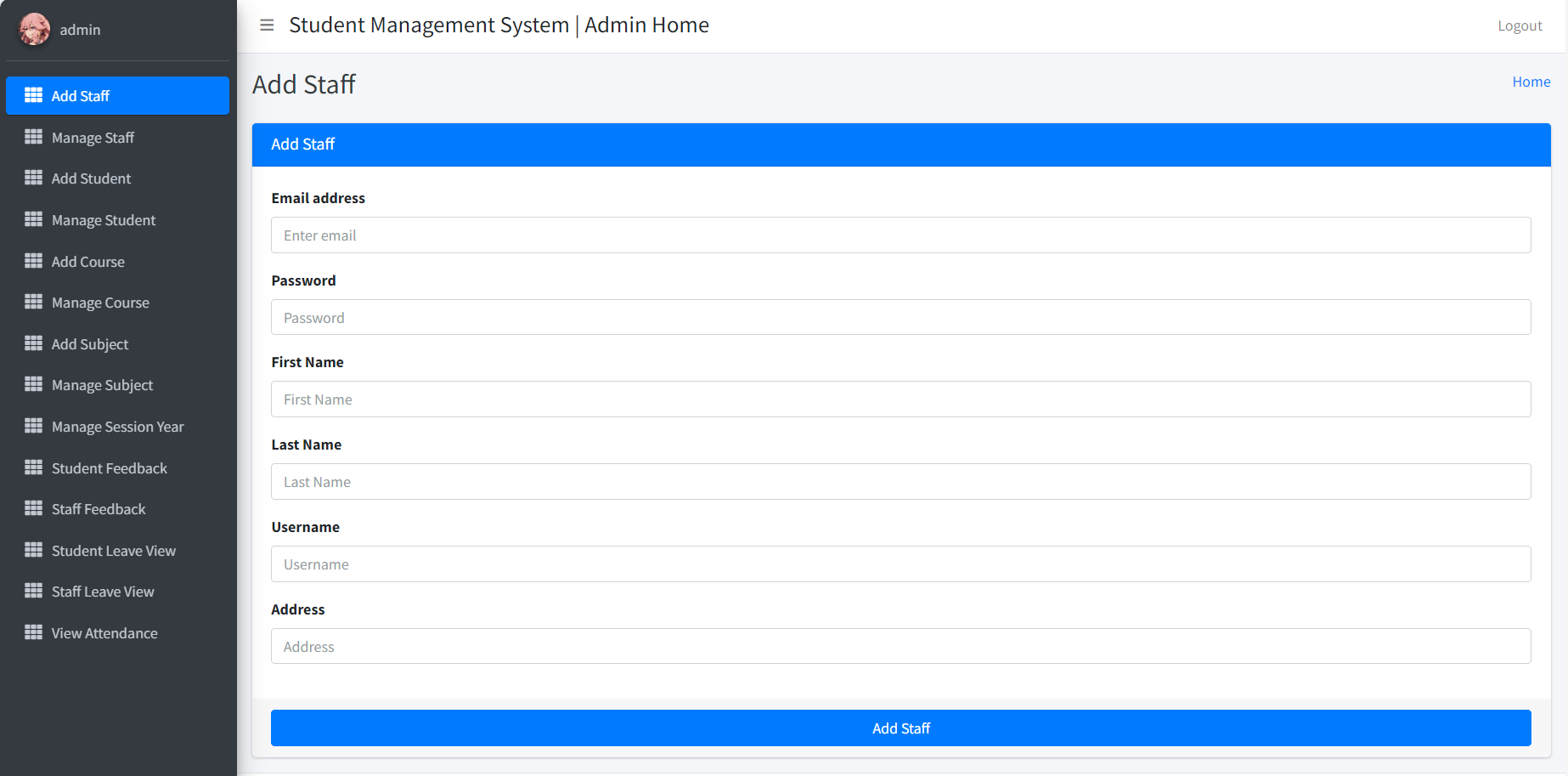
## 

## **II, Các chức năng**

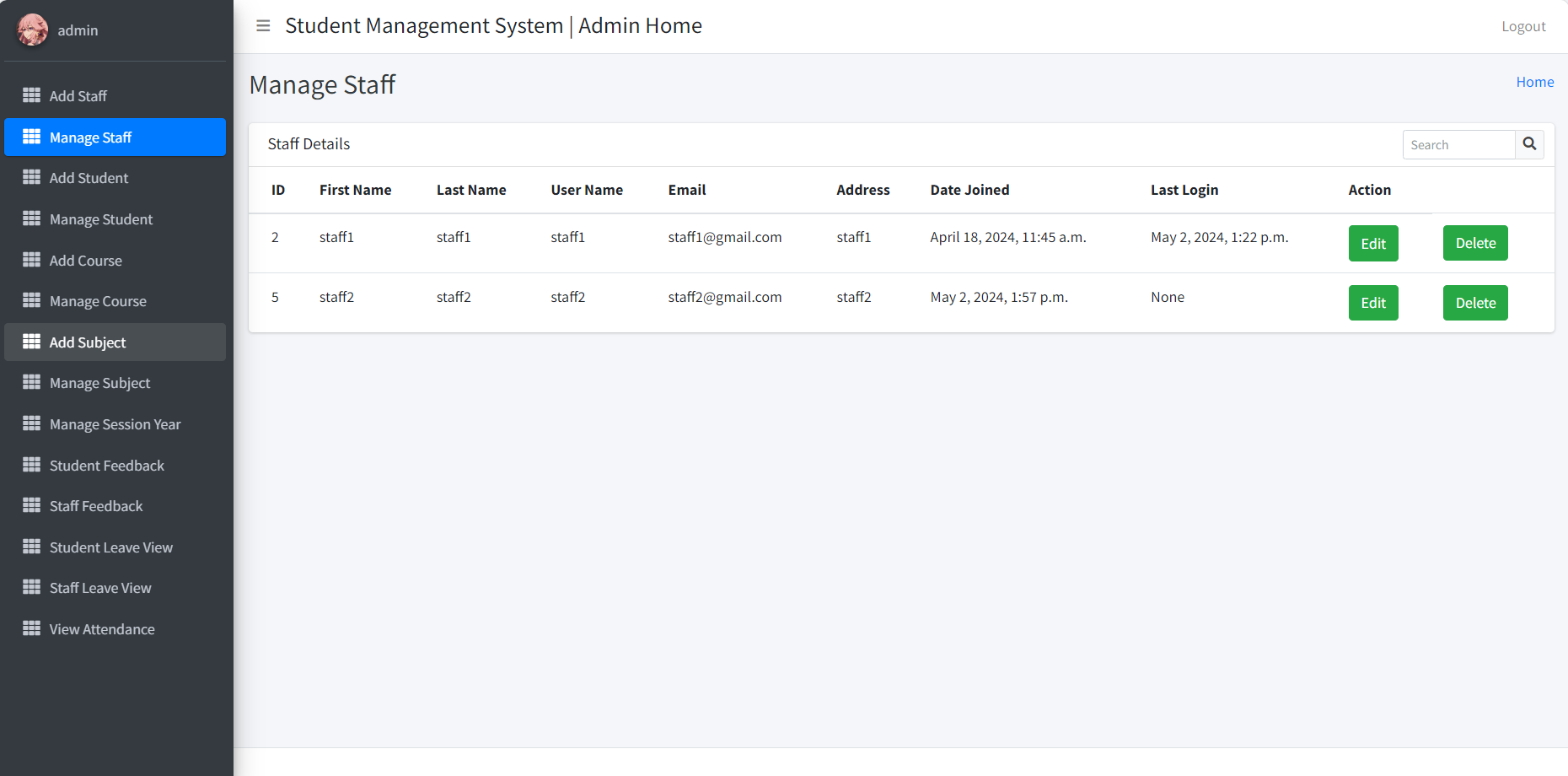
### ***1, Người dùng Admin***

*a, Quản lý Staff*

Chức năng quản lý Staff với các tùy chọn như thêm, sửa thông tin, xóa , tìm kiếm Staff như các hình dưới

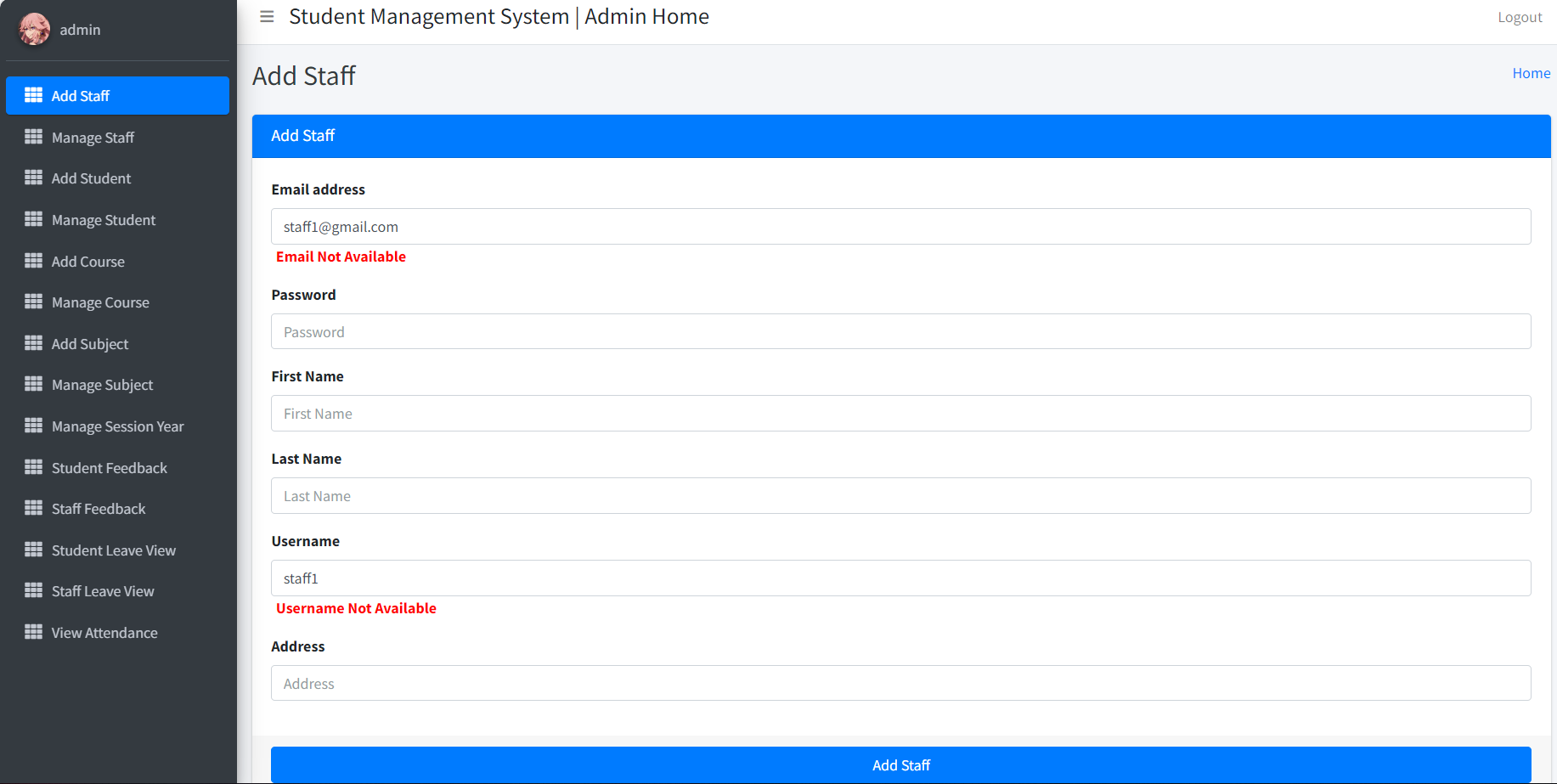


Hình 8



Hình 9

Khi thêm Staff nếu email, username đã tồn tại sẽ có cảnh báo hiện lên



Hình 10 : Thông báo username, email không hợp lệ

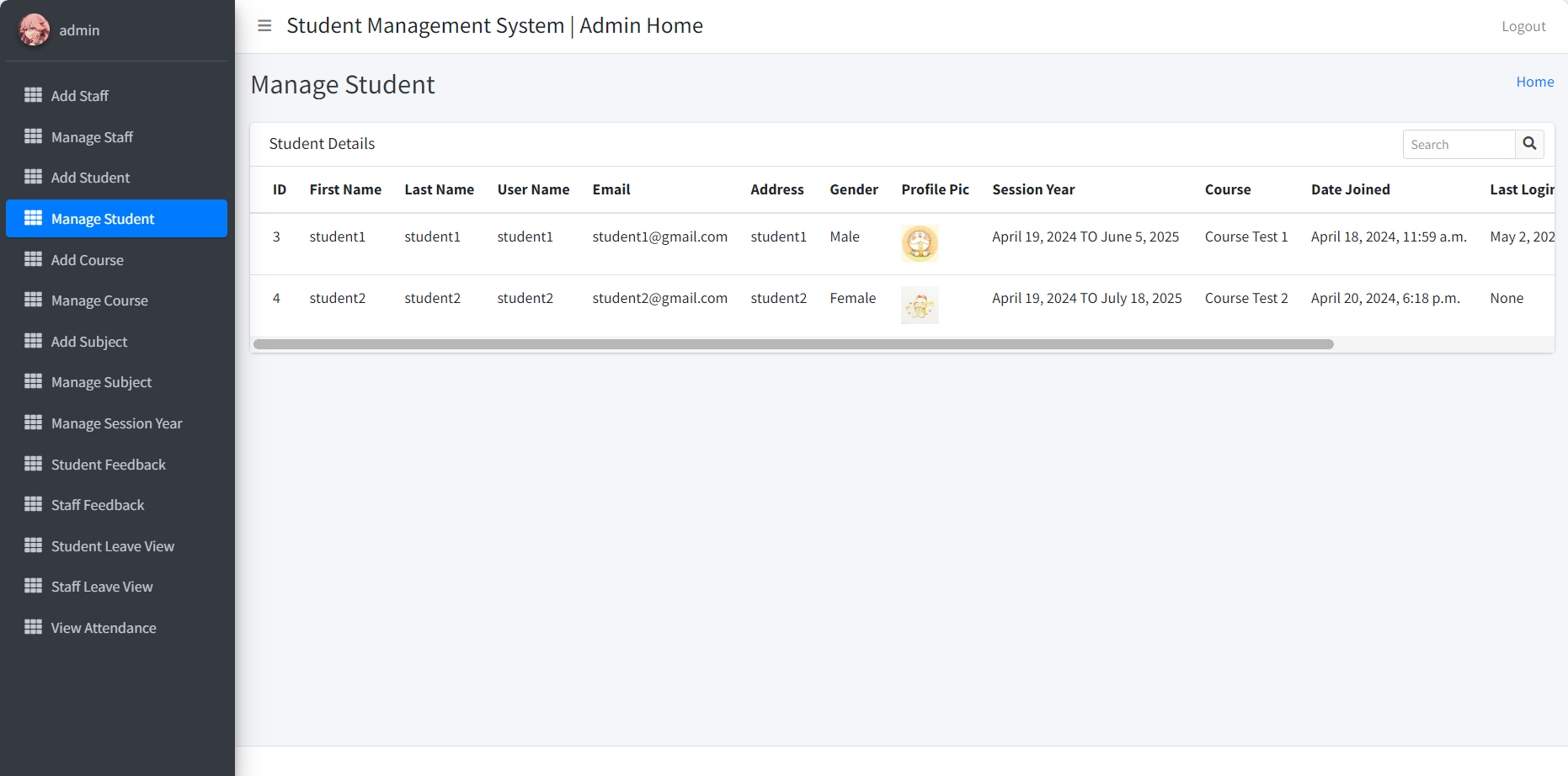
Sau khi thêm Staff thành công sẽ hiện thông báo



Hình 11: Thông báo addstaff thành công

*b, Quản lý Student*

Chức năng quản lý Student tương tự với quản lý Staff nhưng khi thêm học sinh sẽ có thêm các trường khóa học, giới tính, niên khóa, ảnh đại diện



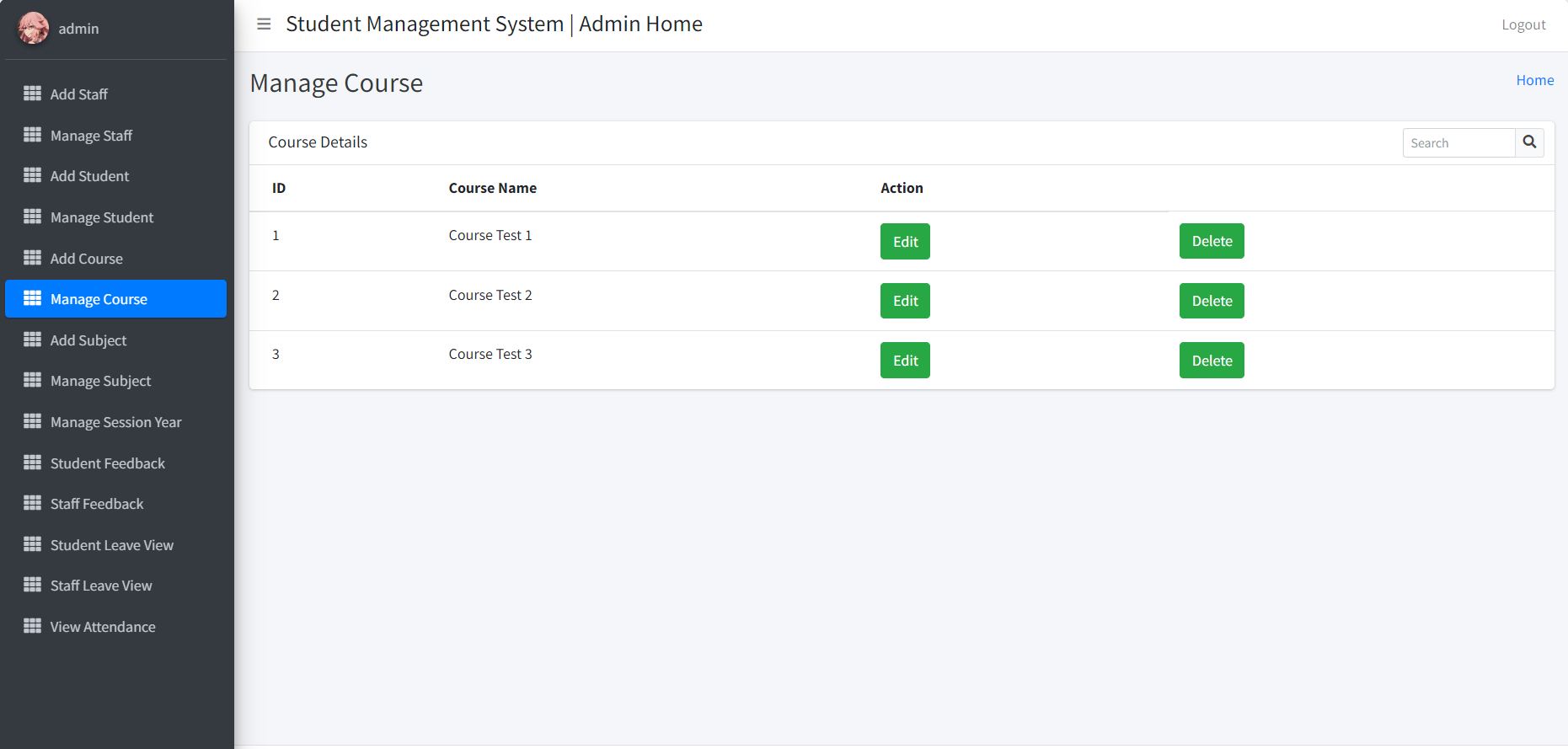
Hình 12



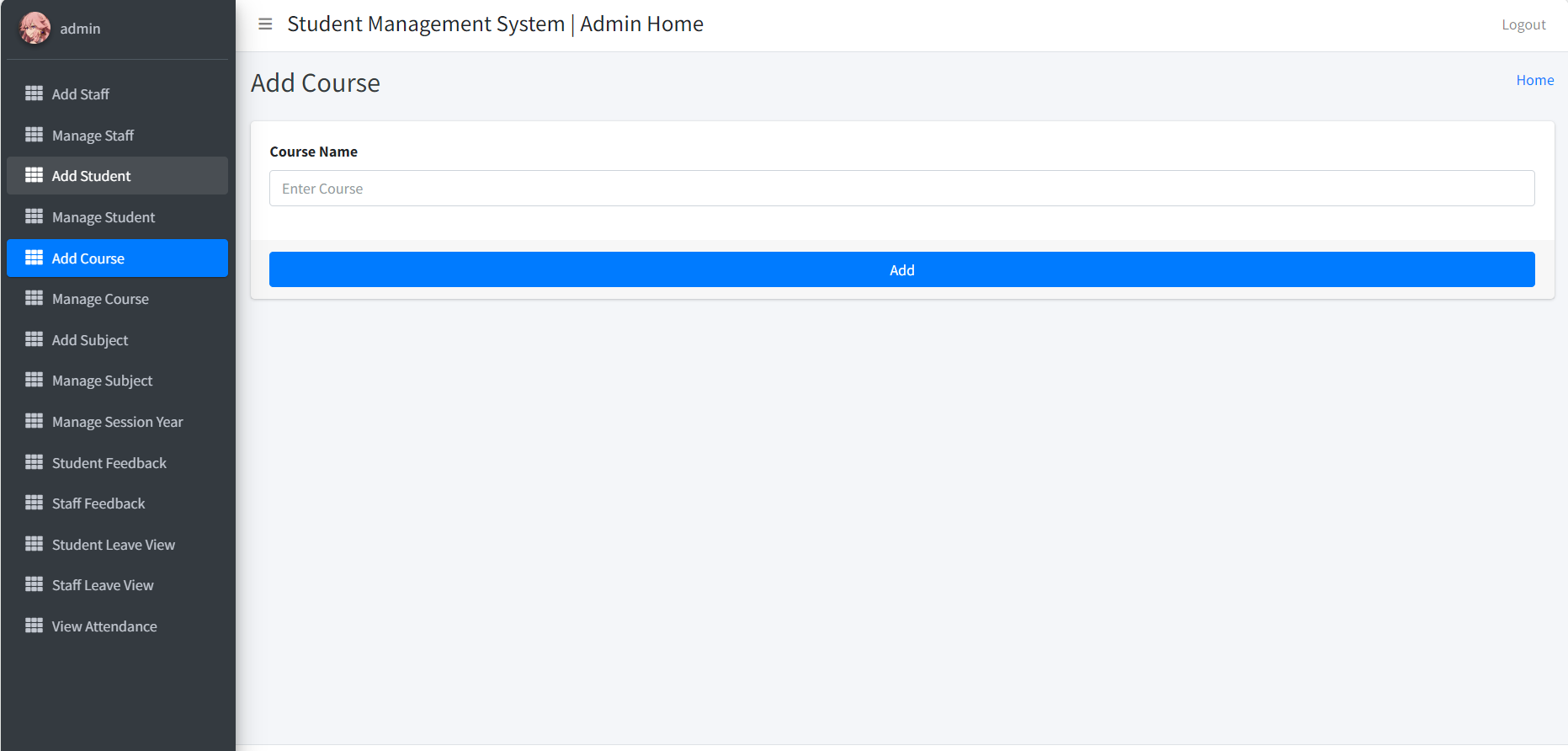
Hình 13

*c, Quản lí khóa học, môn học, năm học*

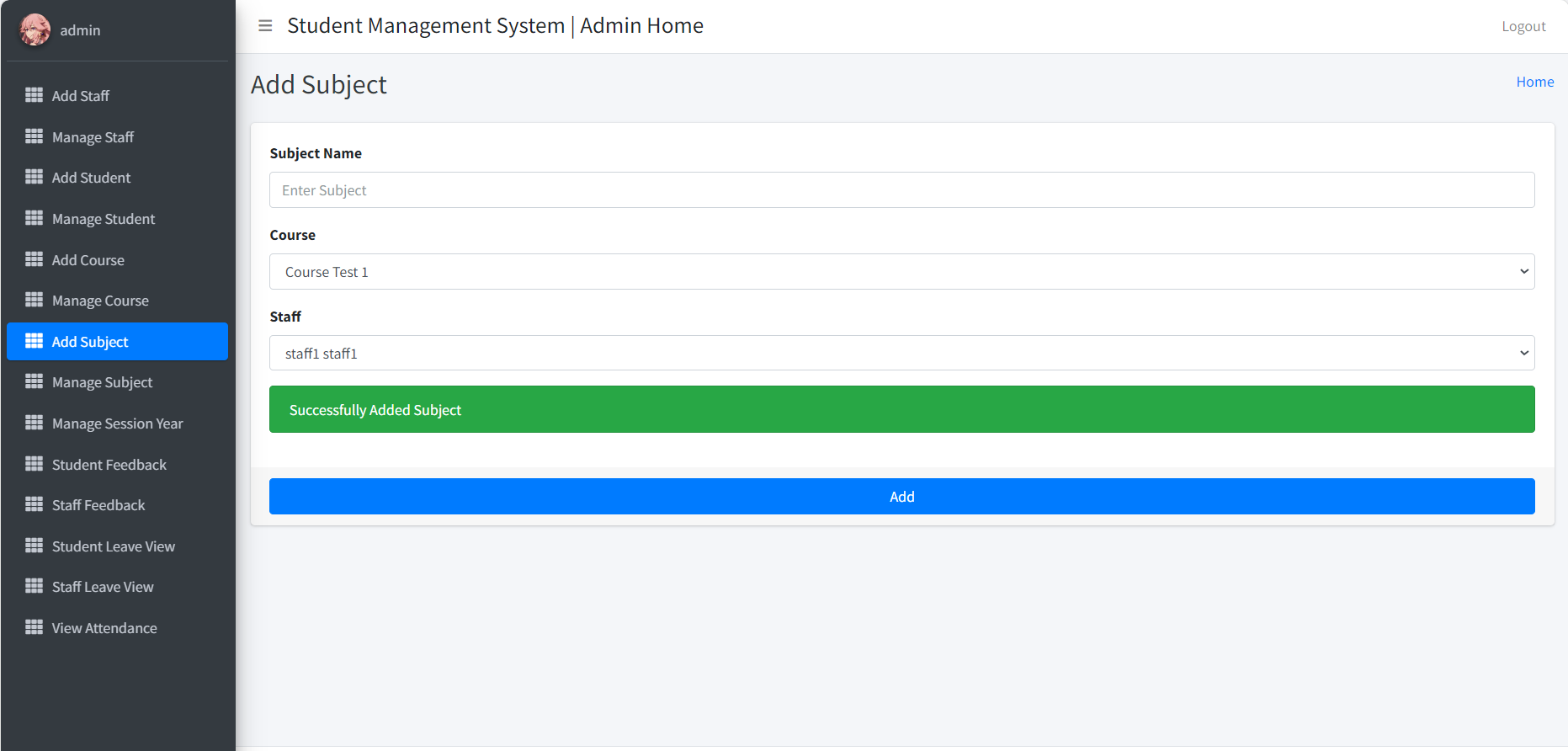
Giao diện với các chức năng thêm, sửa, xóa, tìm kiếm



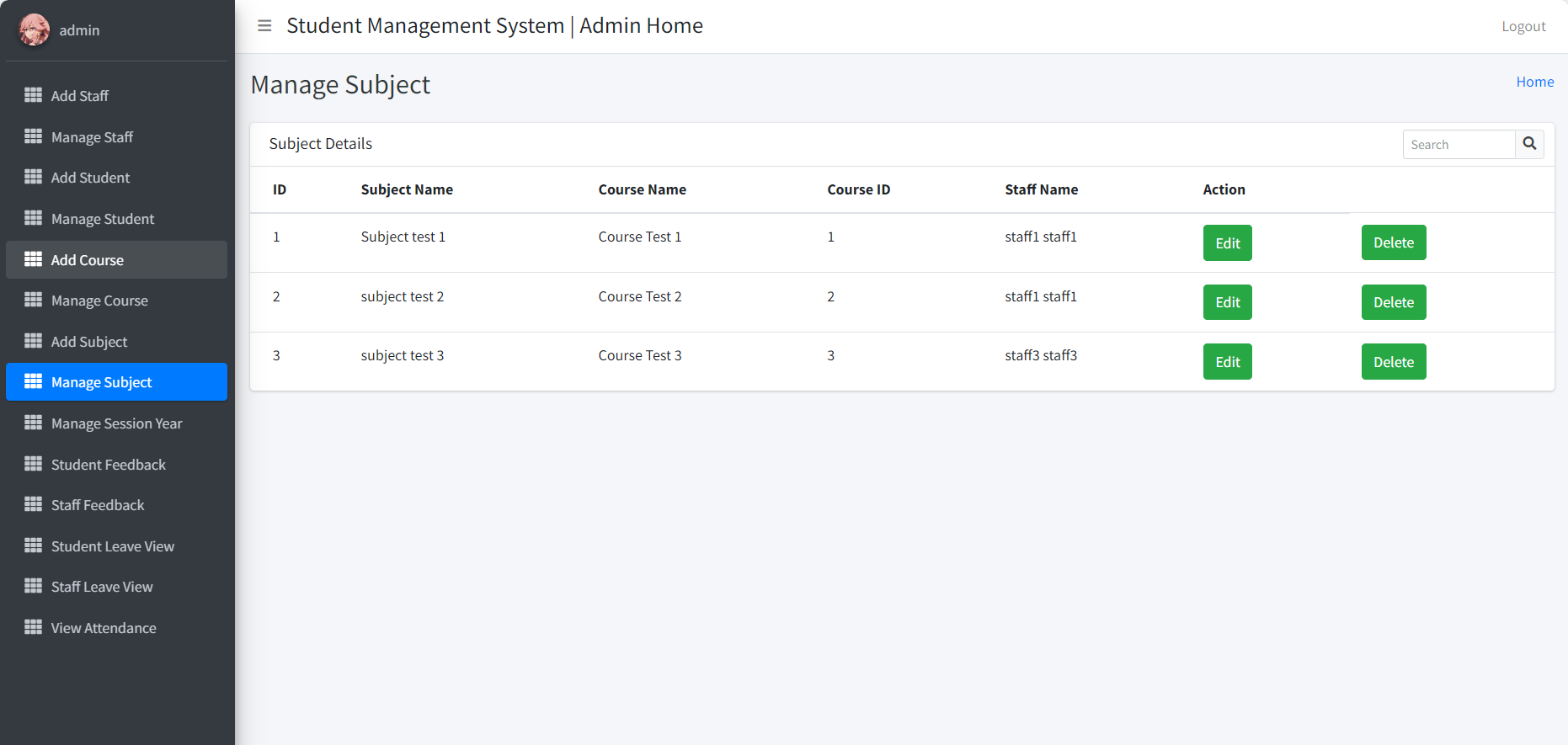
Hình 14



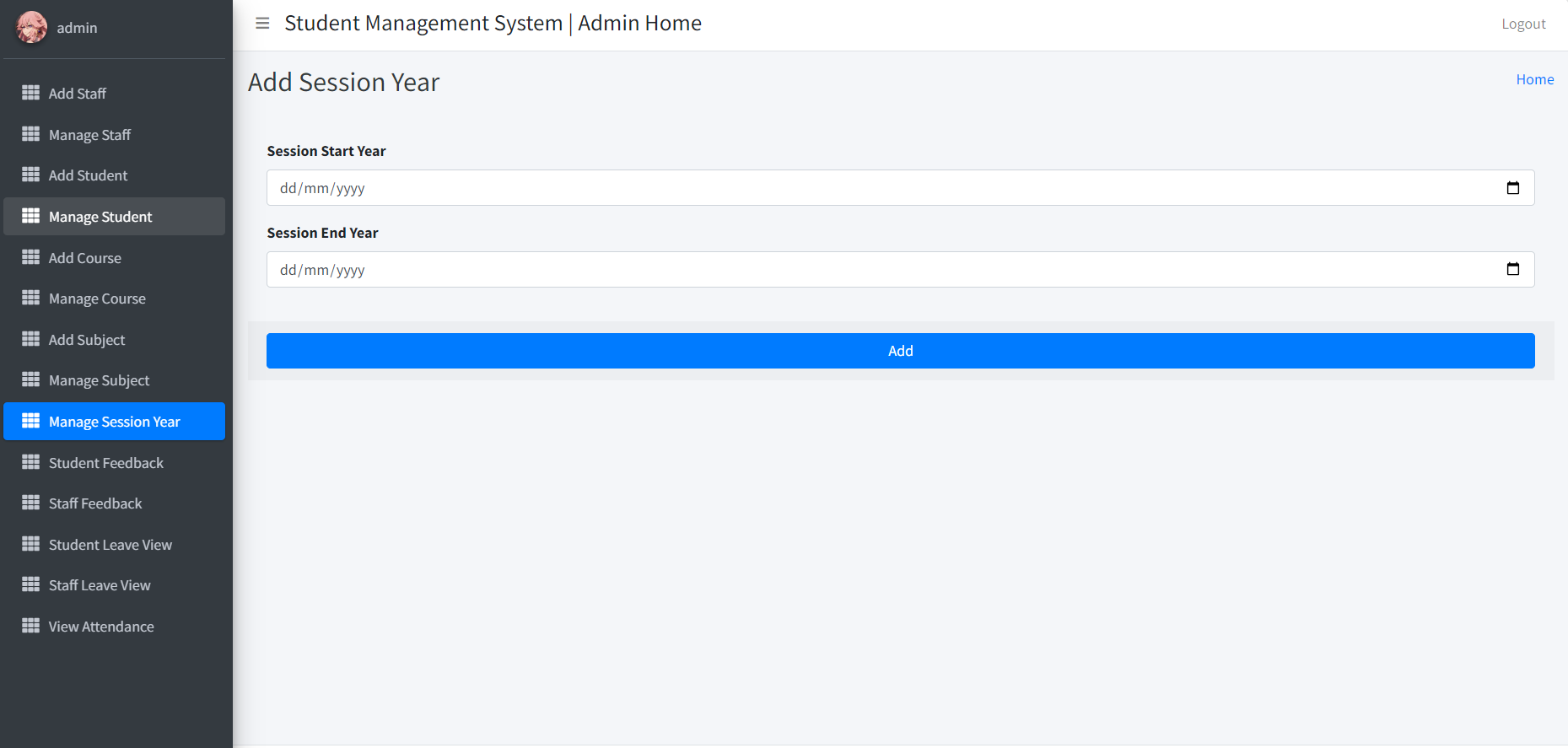
Hình 15



Hình 16



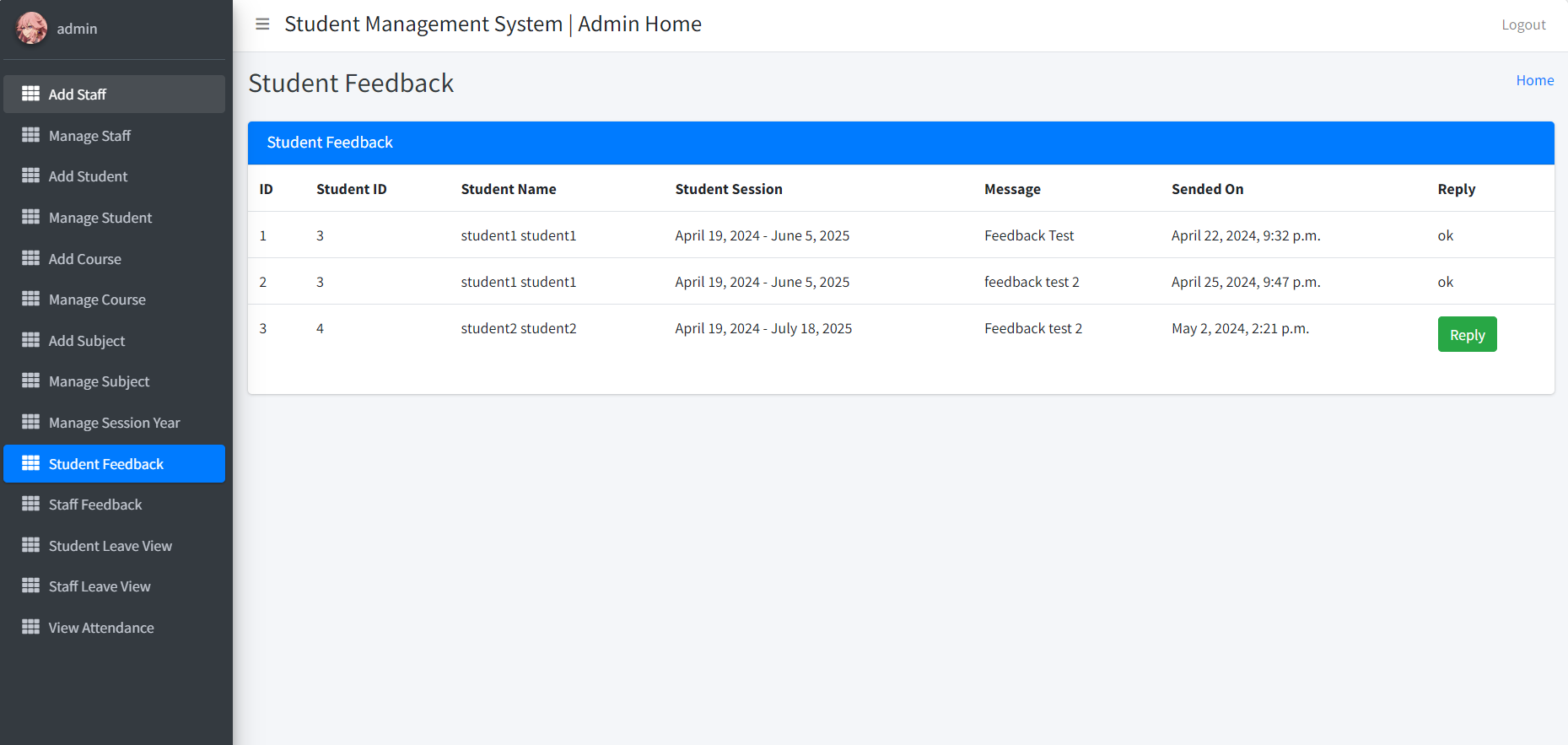
Hình 17



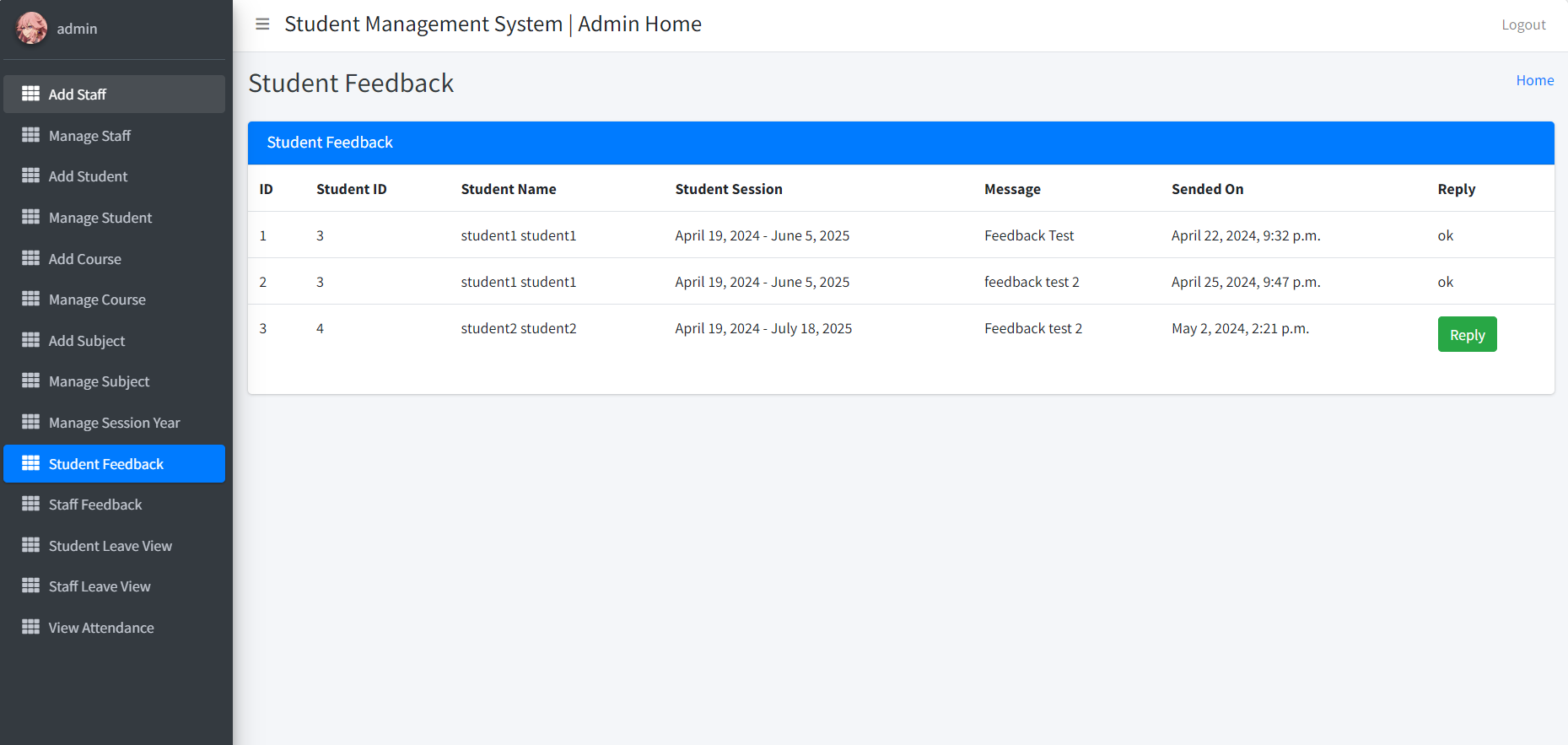
Hình 18

*d, Feedback của Staff và Student*

Hiển thị feedback của Staff và Student

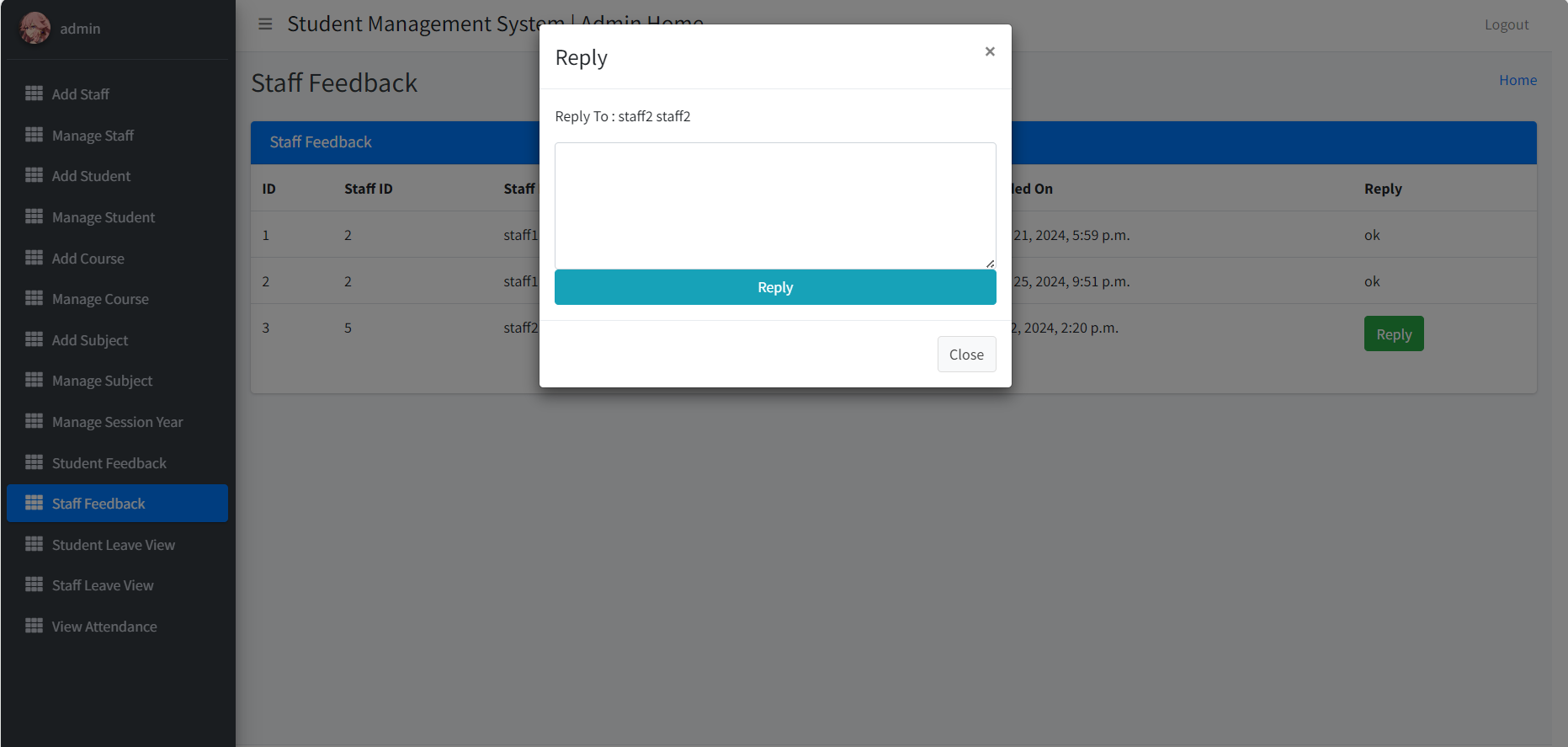


Hình 19 : Feedback của Student



Hình 20 : Feedback của Staff

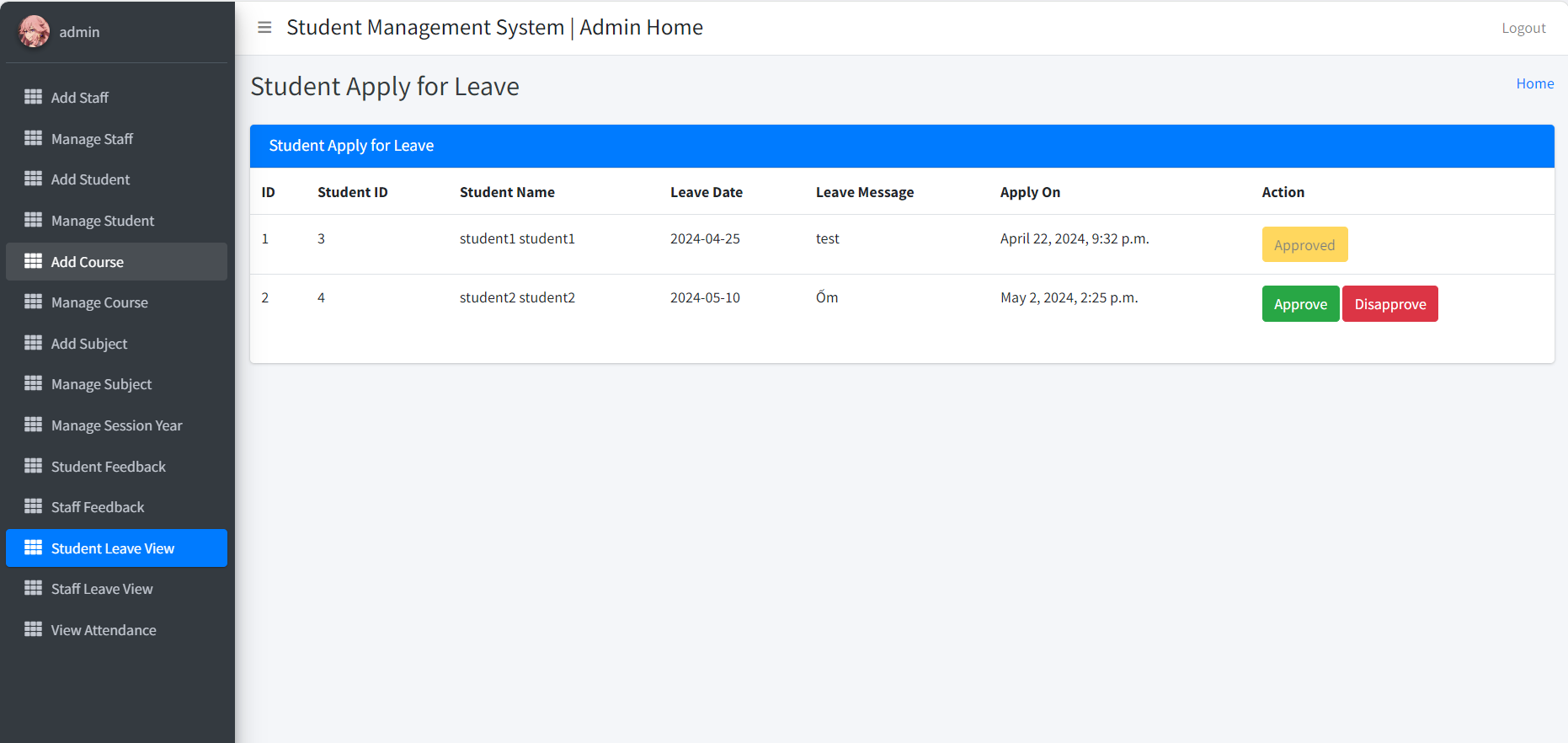
Với feedback chưa phản hồi có lựa chọn reply sau khi ấn sẽ hiện lên khung chat phản hồi cho người dùng



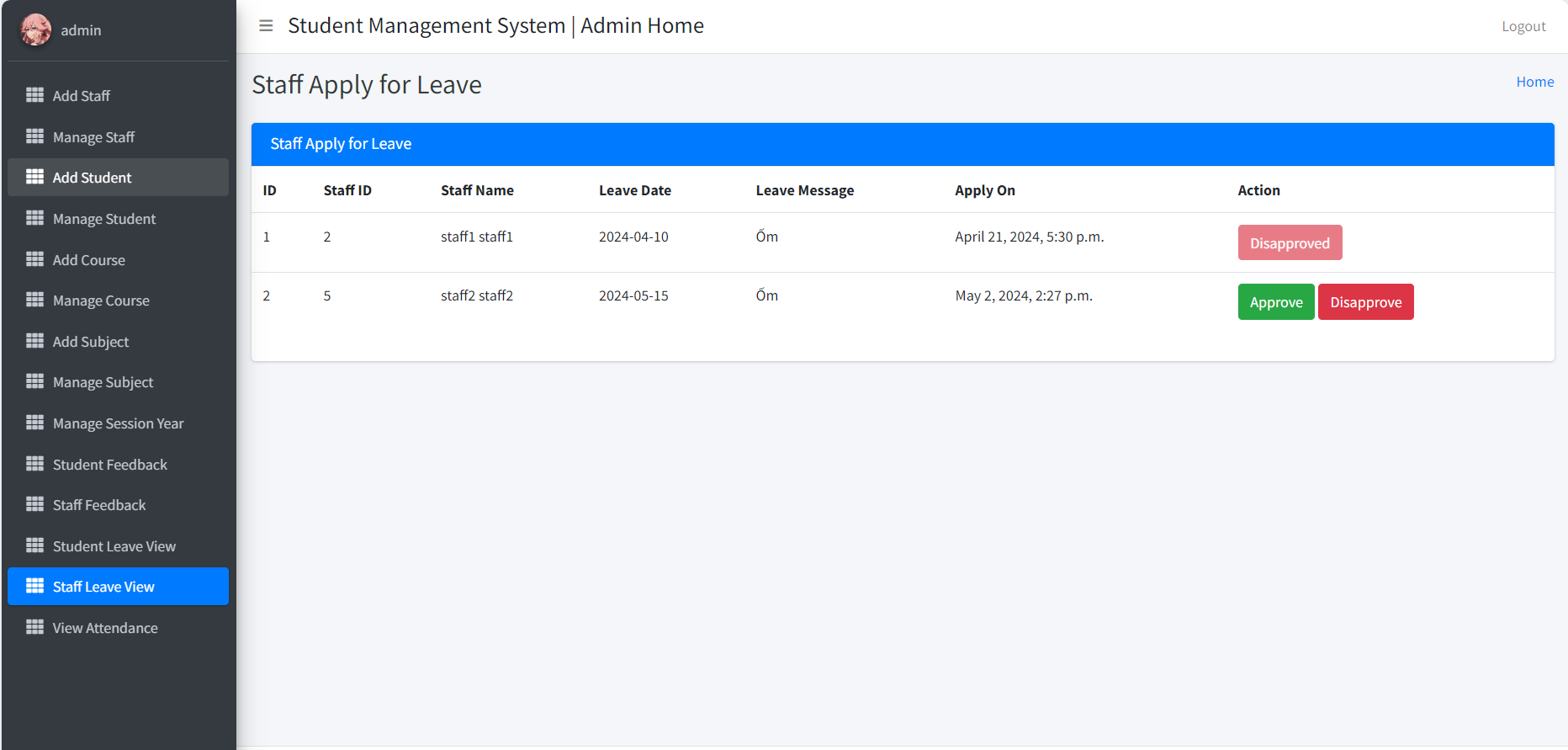
Hình 21: Giao diện Reply feedback

*e, Nghỉ phép của Staff và Student*

Hiển thị yêu cầu xin nghỉ phép của Staff và Student với các tùy chọn đồng ý hoặc không đồng ý



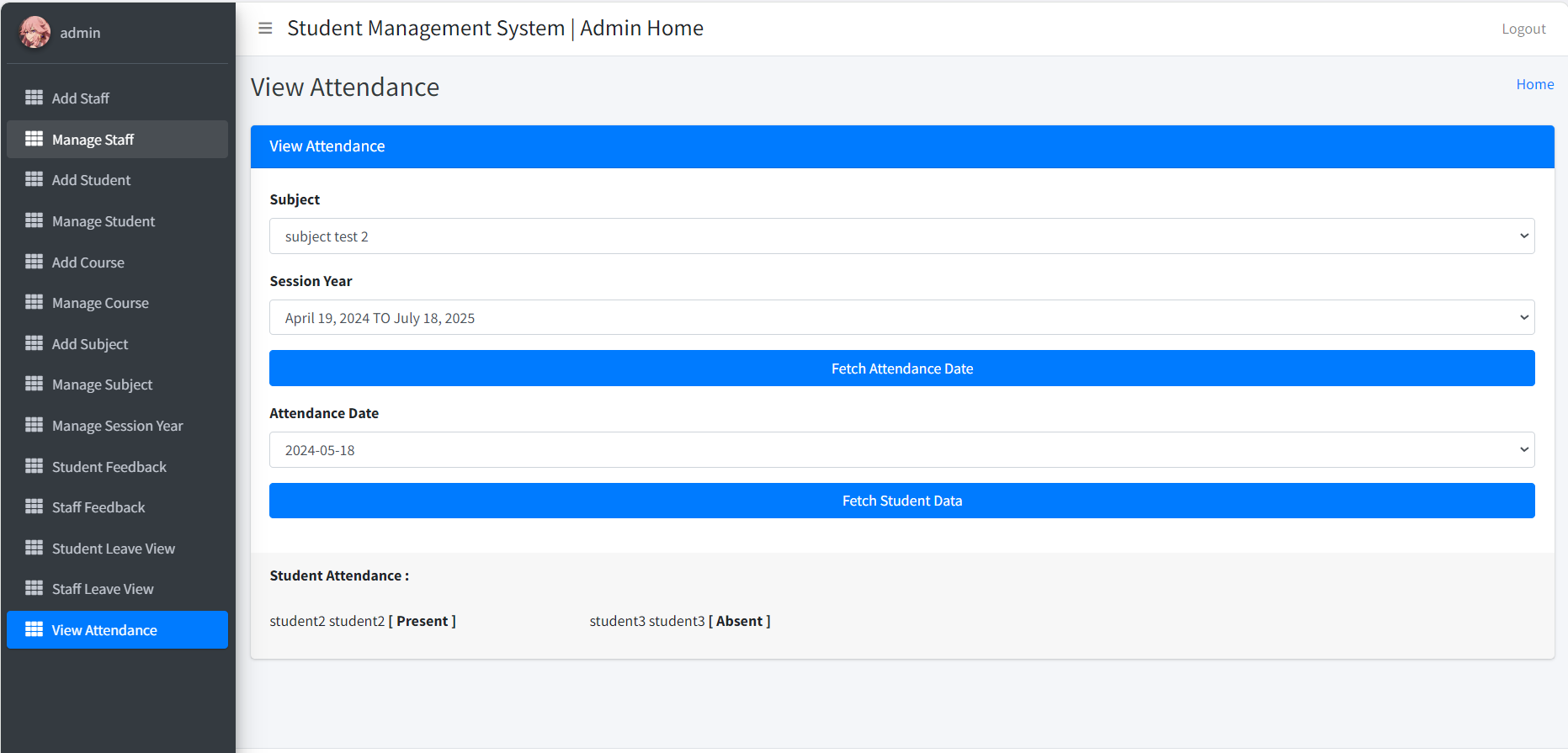
Hình 22: Đồng ý



Hình 23 : Từ chối

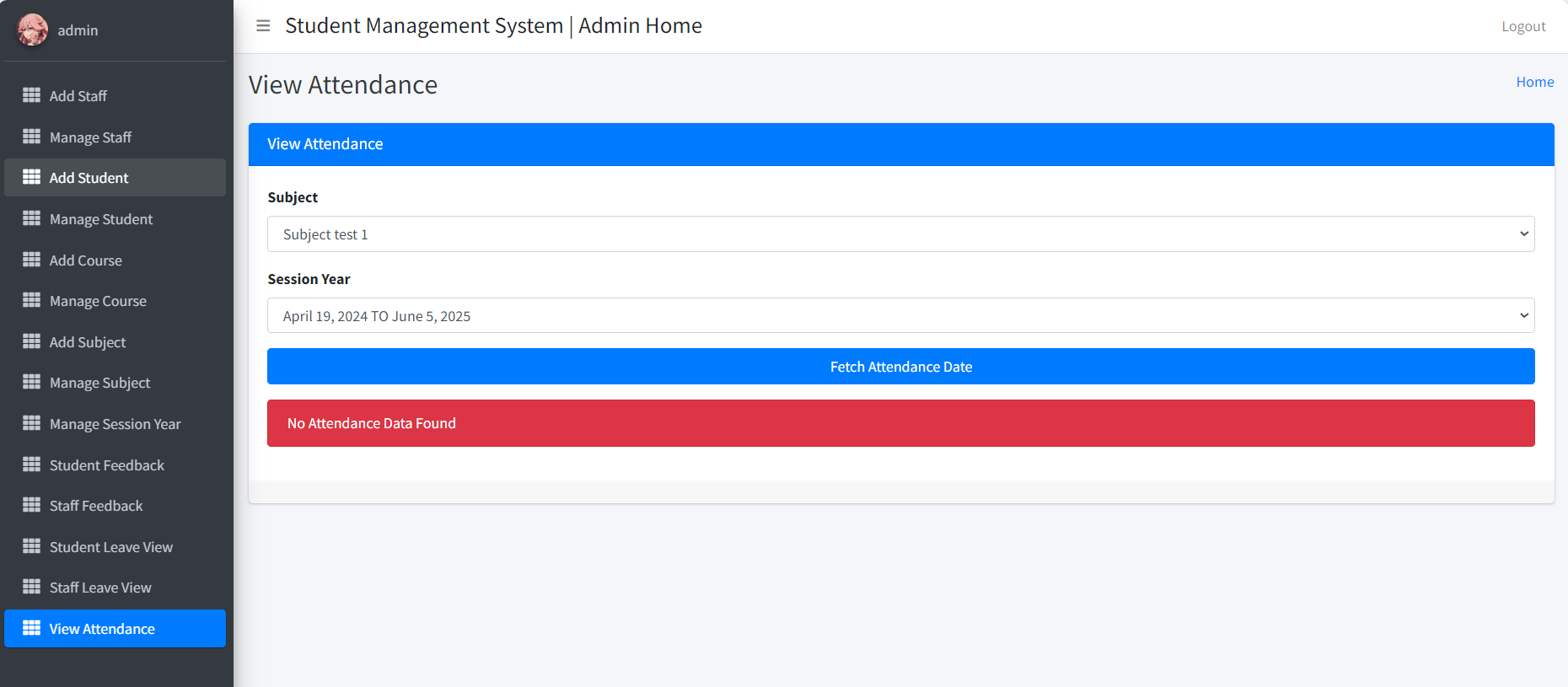
*f, Xem tính trạng điểm danh*

Giao diện thực hiện chọn môn học và năm học để hiển thị các ngày điểm danh tiến hành chọn ngày để xem dữ liệu điểm danh.



Hình 24 : Xem tình trạng điểm danh

Hiển thị thông báo khi không có dữ liệu điểm danh

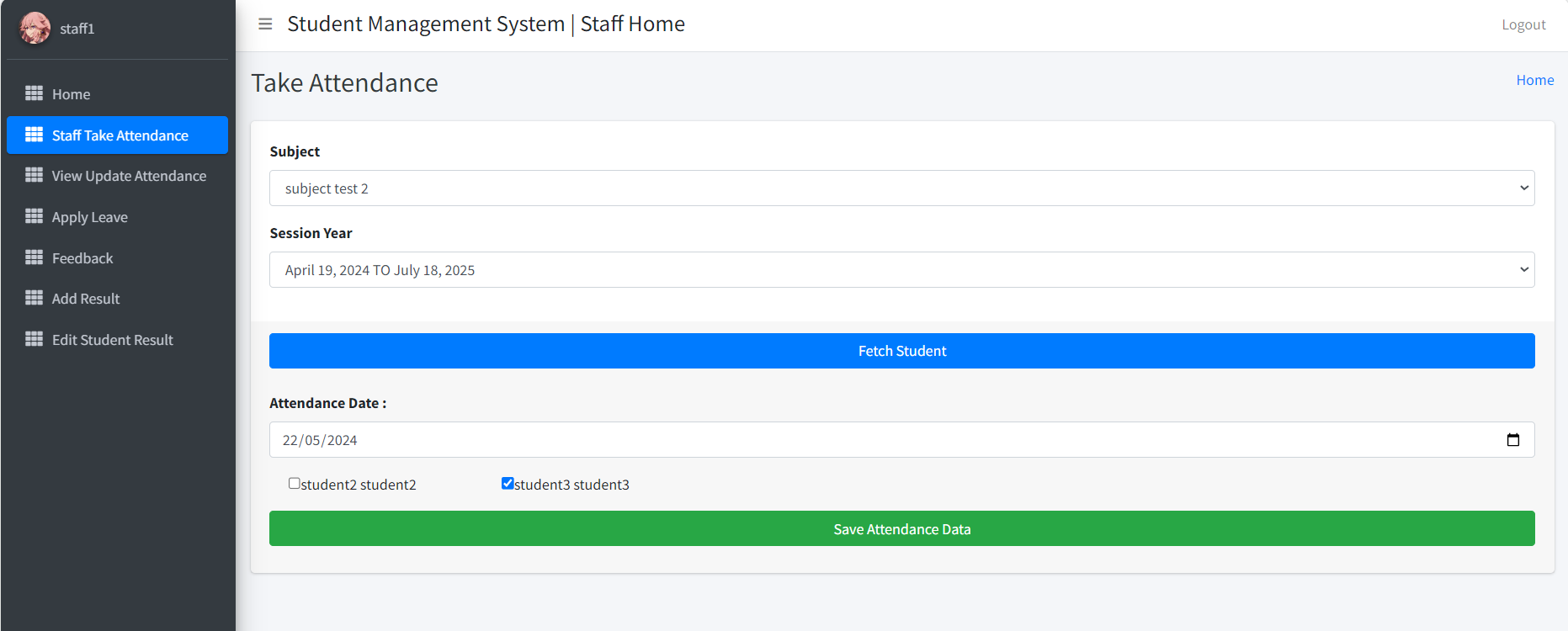


Hình 25 : Thông báo khi không có dữ liệu điểm danh

### ***2, Người dùng Staff***

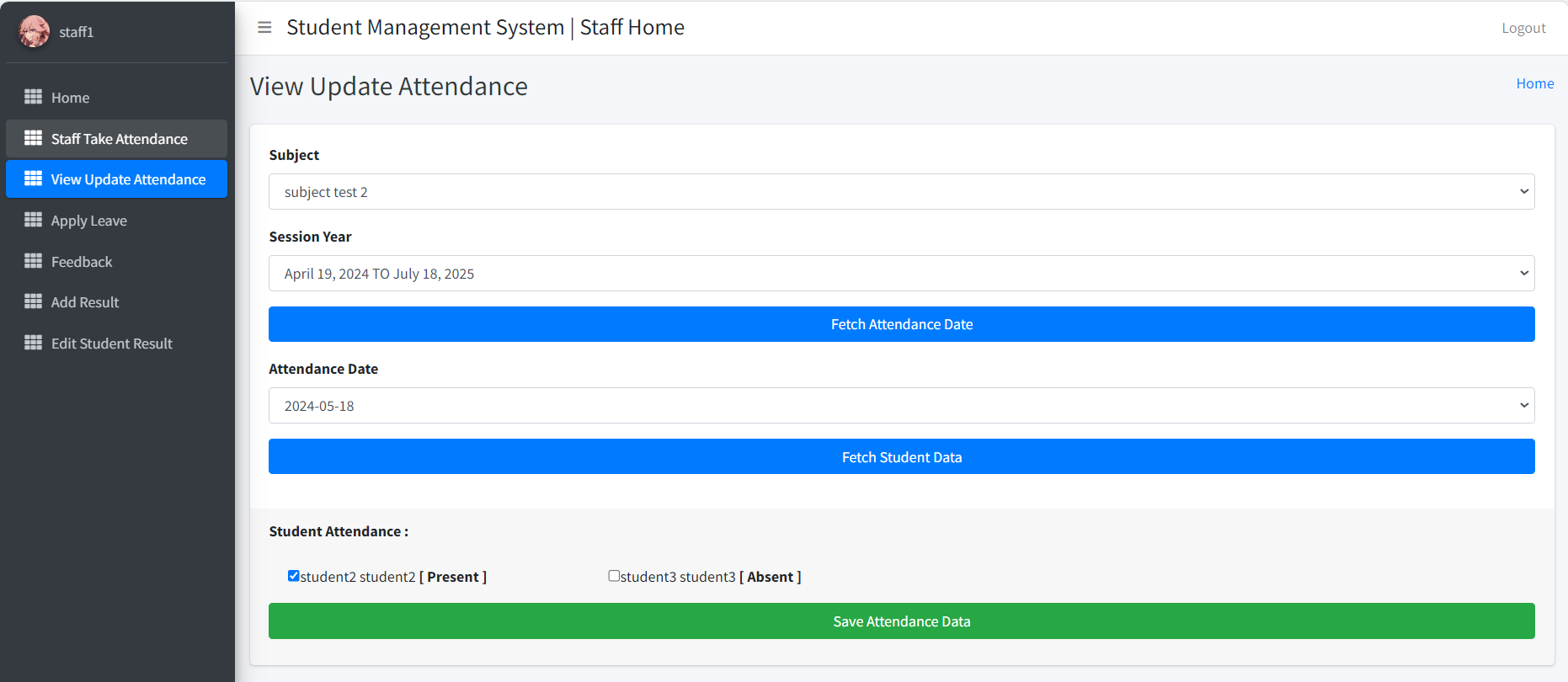
*a, Điểm danh sinh viên*

Sau khi Staff chọn môn học và niên khóa có thể chọn ngày và tiến hành điểm danh sinh viên



Hình 26

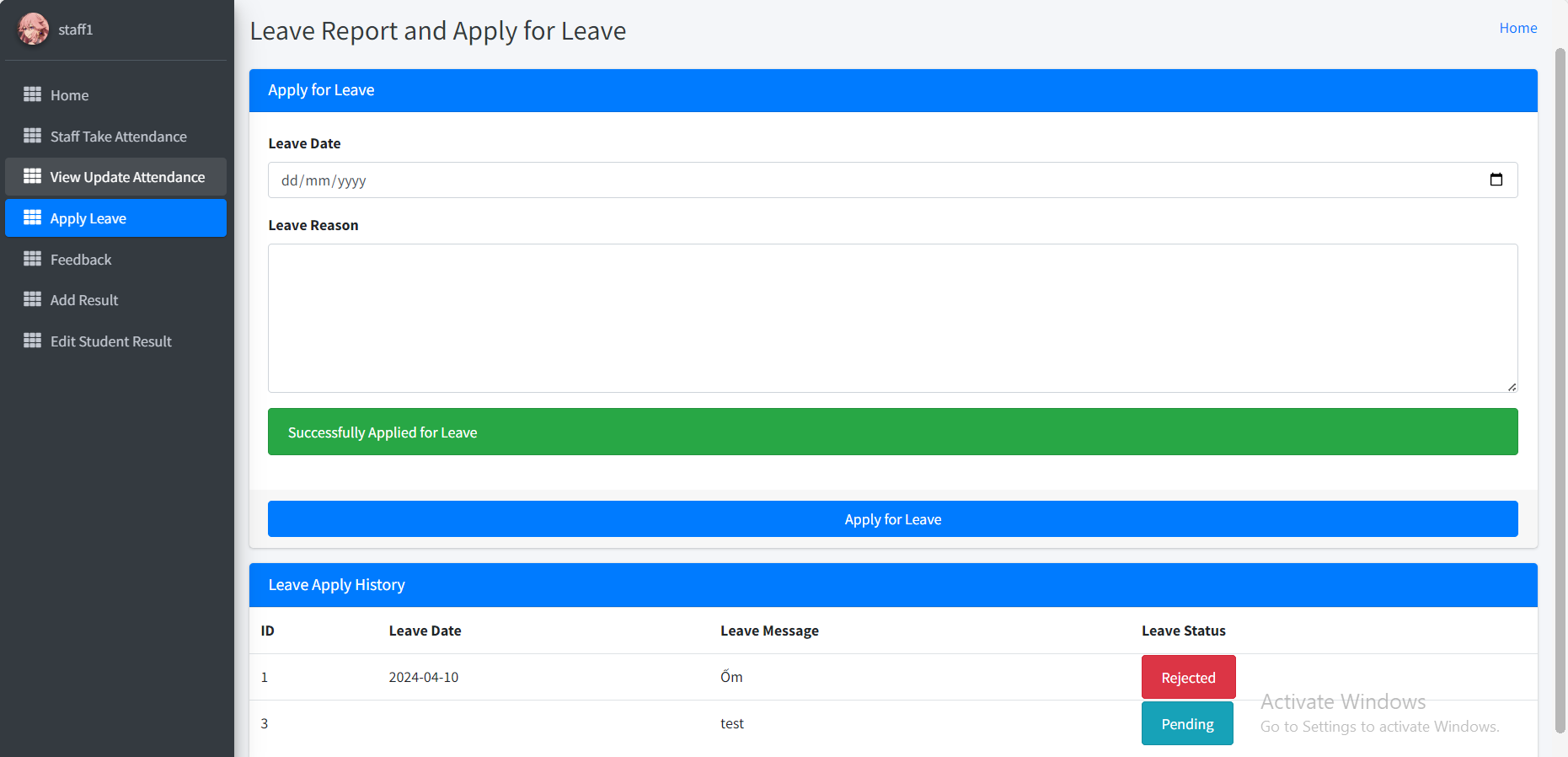
Trạng thái điểm danh có thể thay đổi được



Hình 27

*b, Xin nghỉ phép*

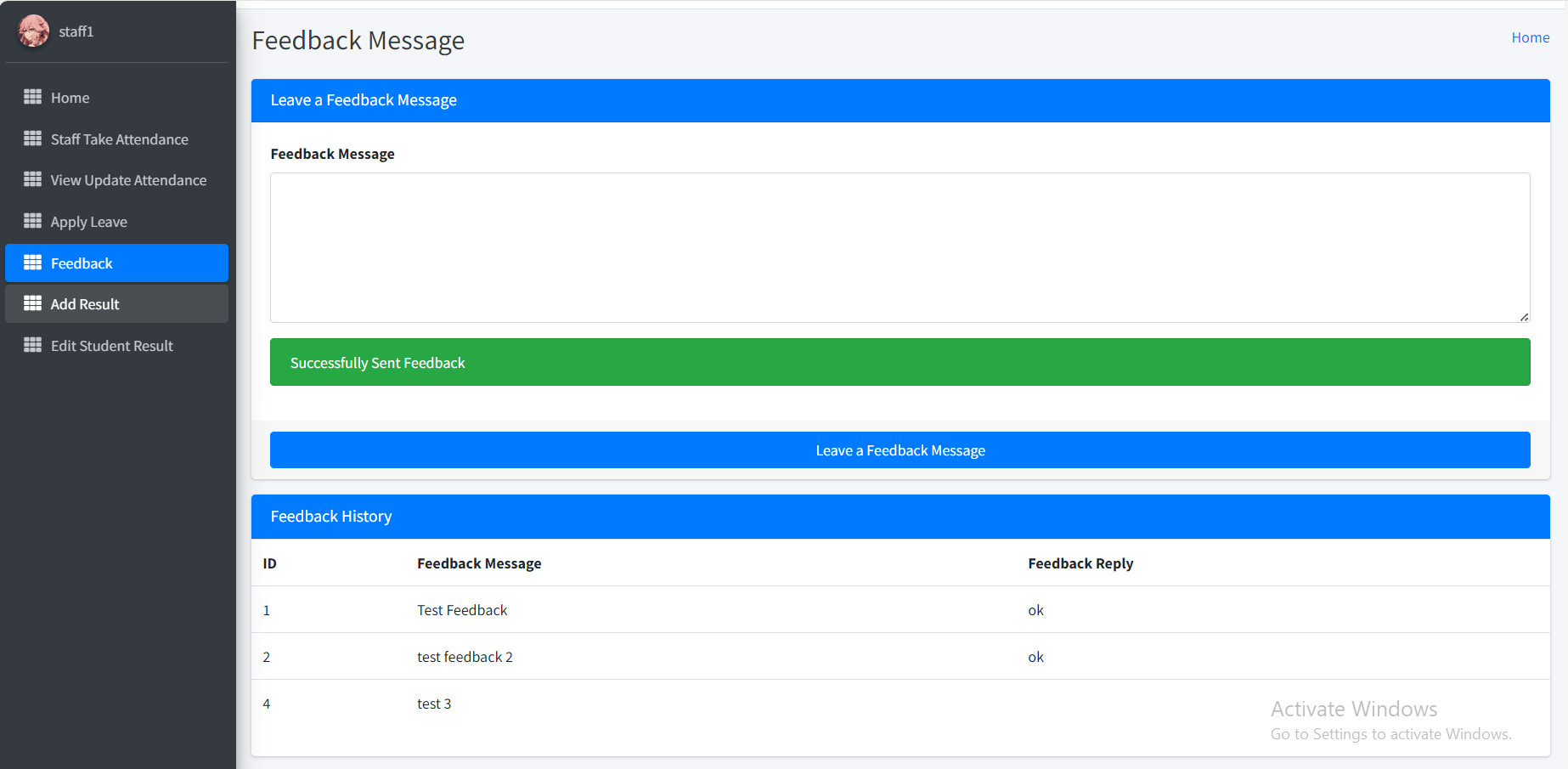
Staff có thể chọn ngày và nêu lý do để gửi yêu cầu xin nghỉ phép tới Admin



Hình 28

*c, Gửi feedback*

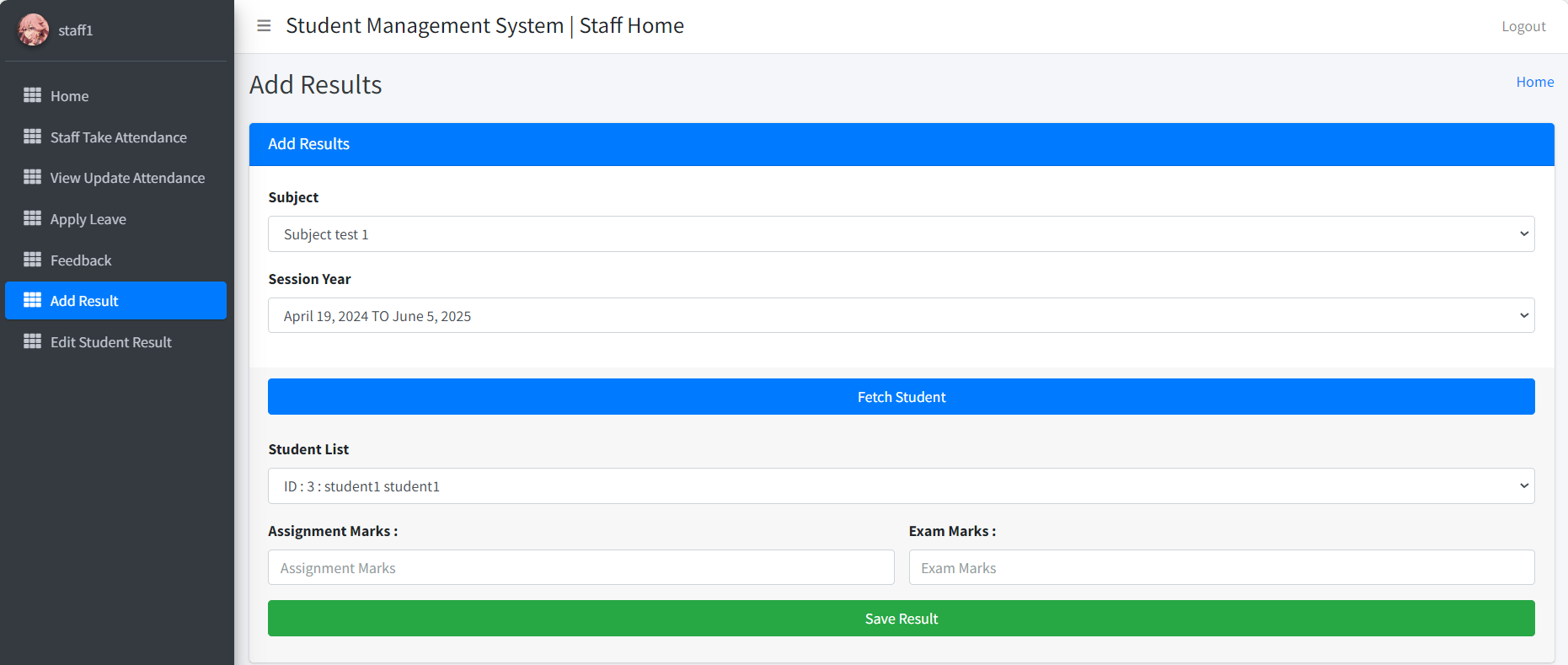
Staff có thể gửi nhận xét, góp ý tới Admin và xem phản hồi



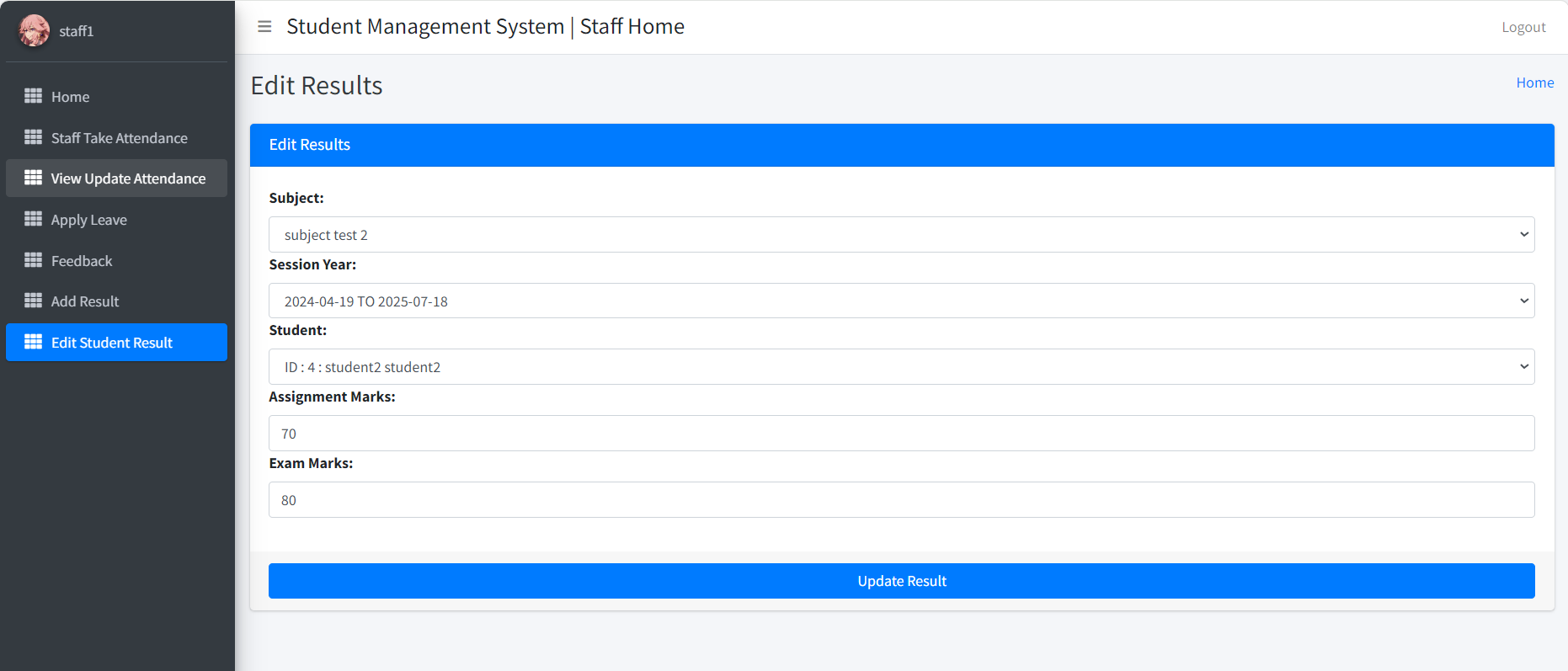
Hình 29

*d, Chấm điểm môn học*

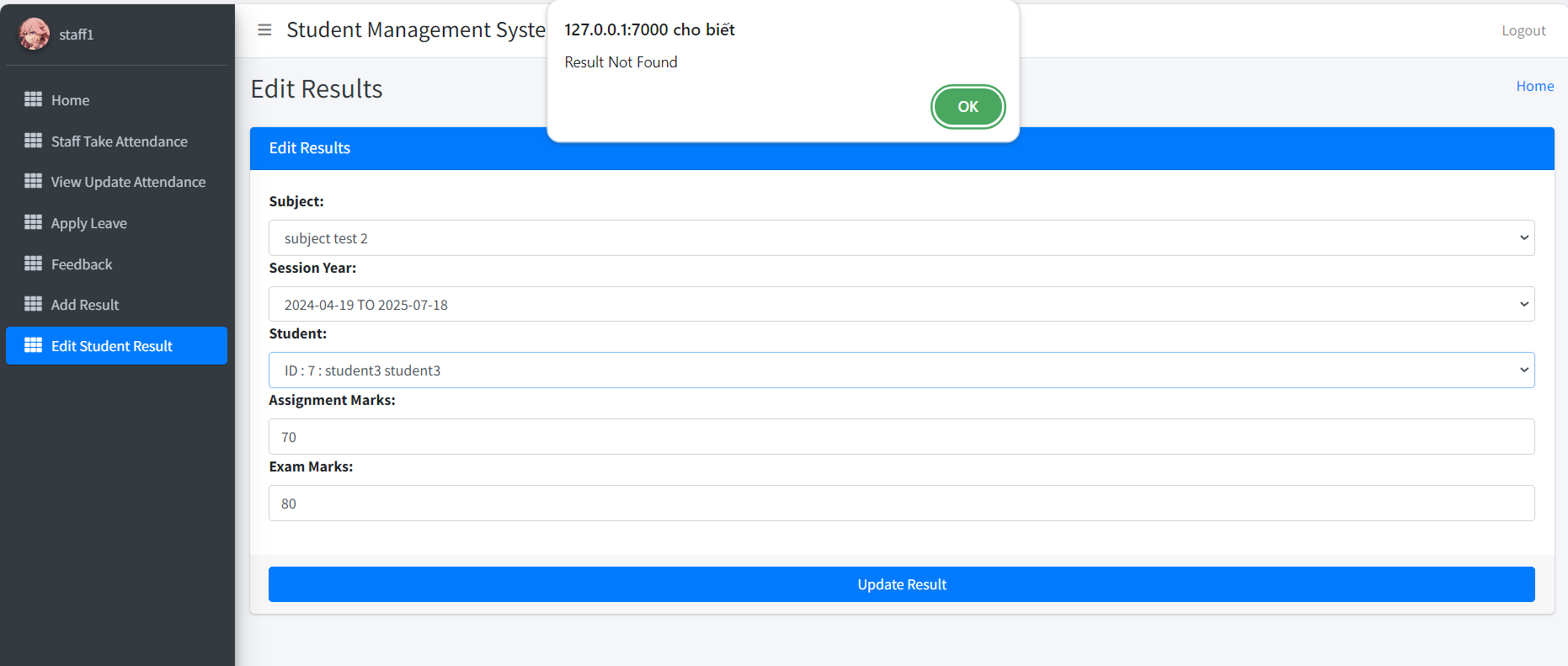
Sau khi chọn môn học và niên khóa Staff có thể tiến hành chấm điểm bài tập và điểm kiểm tra cho học sinh



Hình 30



Hình 31: Giao diện cập nhật điểm số

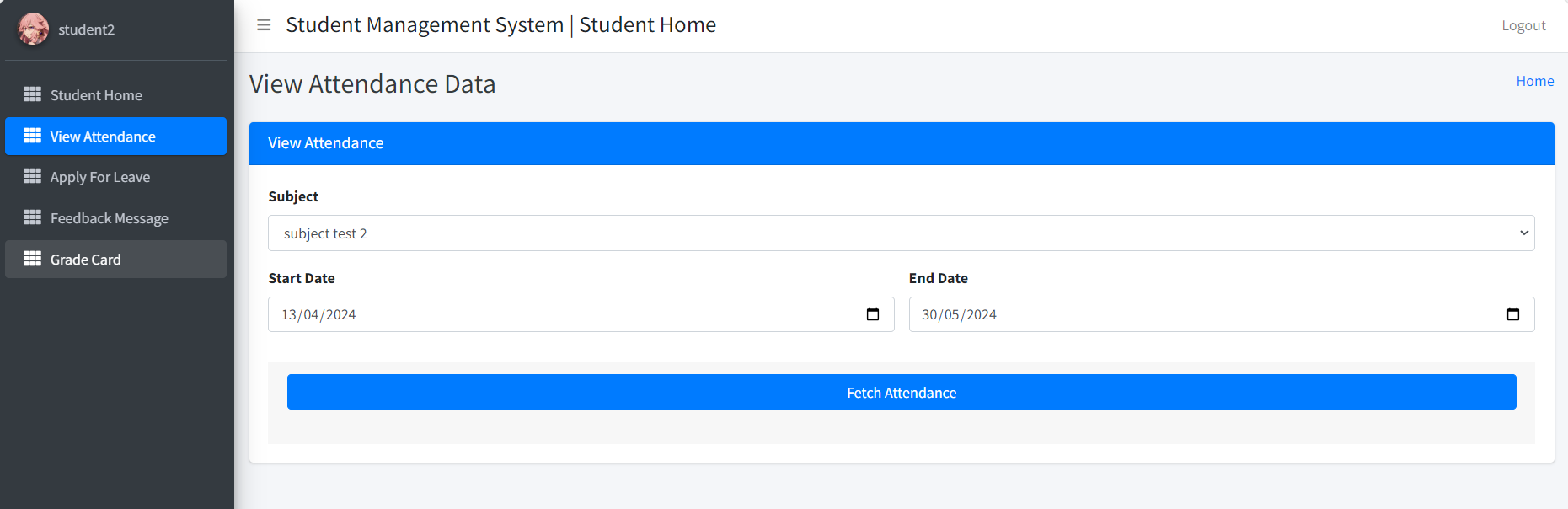


Hình 32: Thông báo khi sinh viên chưa có điểm

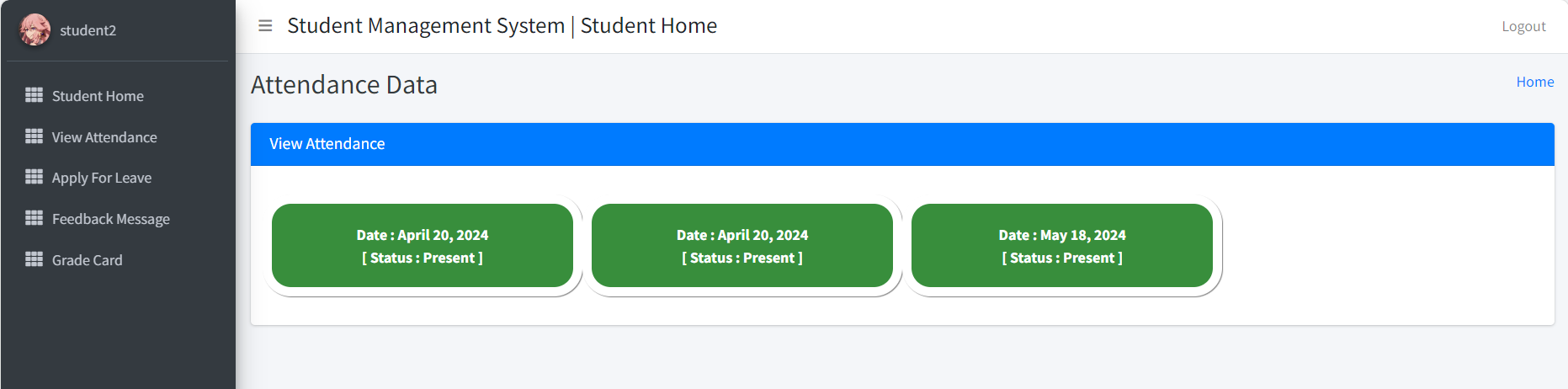
### ***3, Người dùng Student***

*a, Xem trạng thái điểm danh*

Student có thể chọn môn và khoảng thời gian để xem thống kê dữ liệu điểm danh



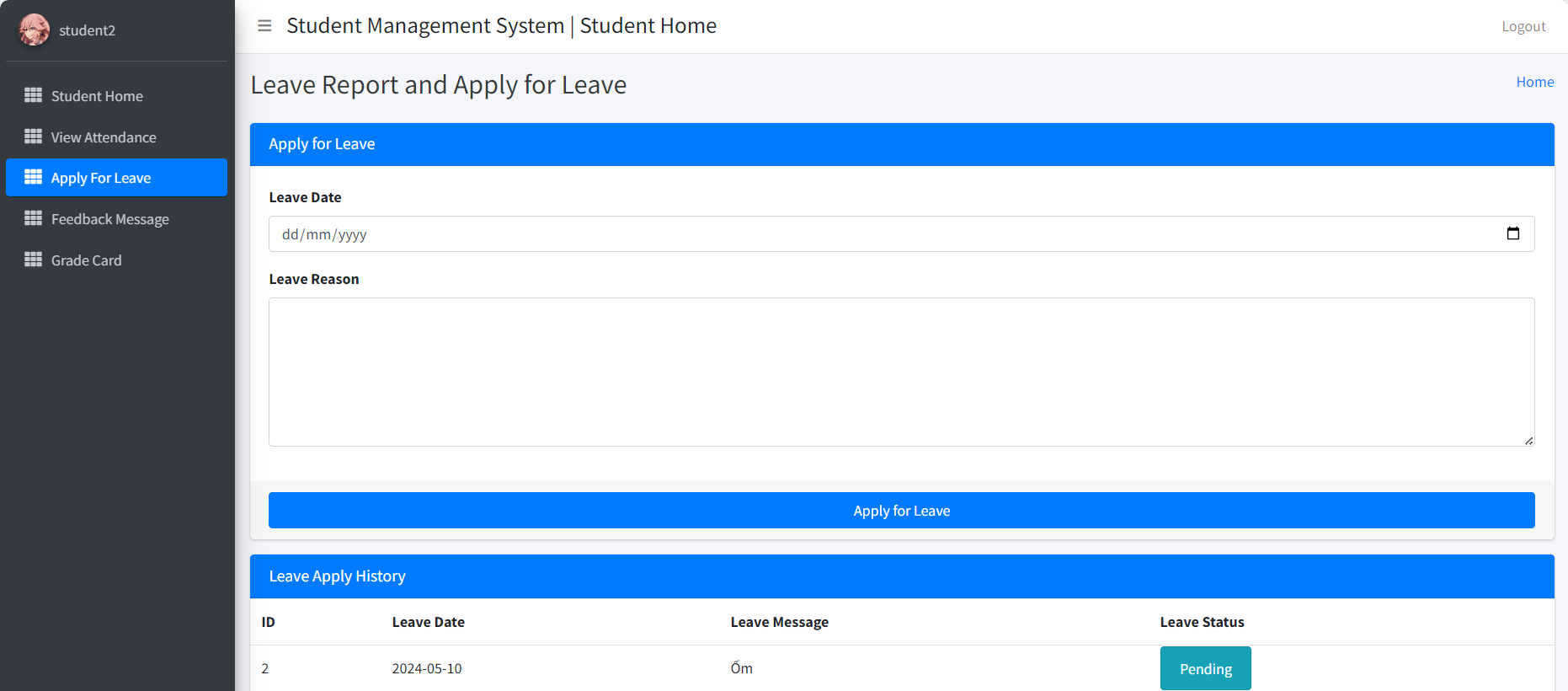
Hình 33



Hình 34

*b, Xin nghỉ phép*

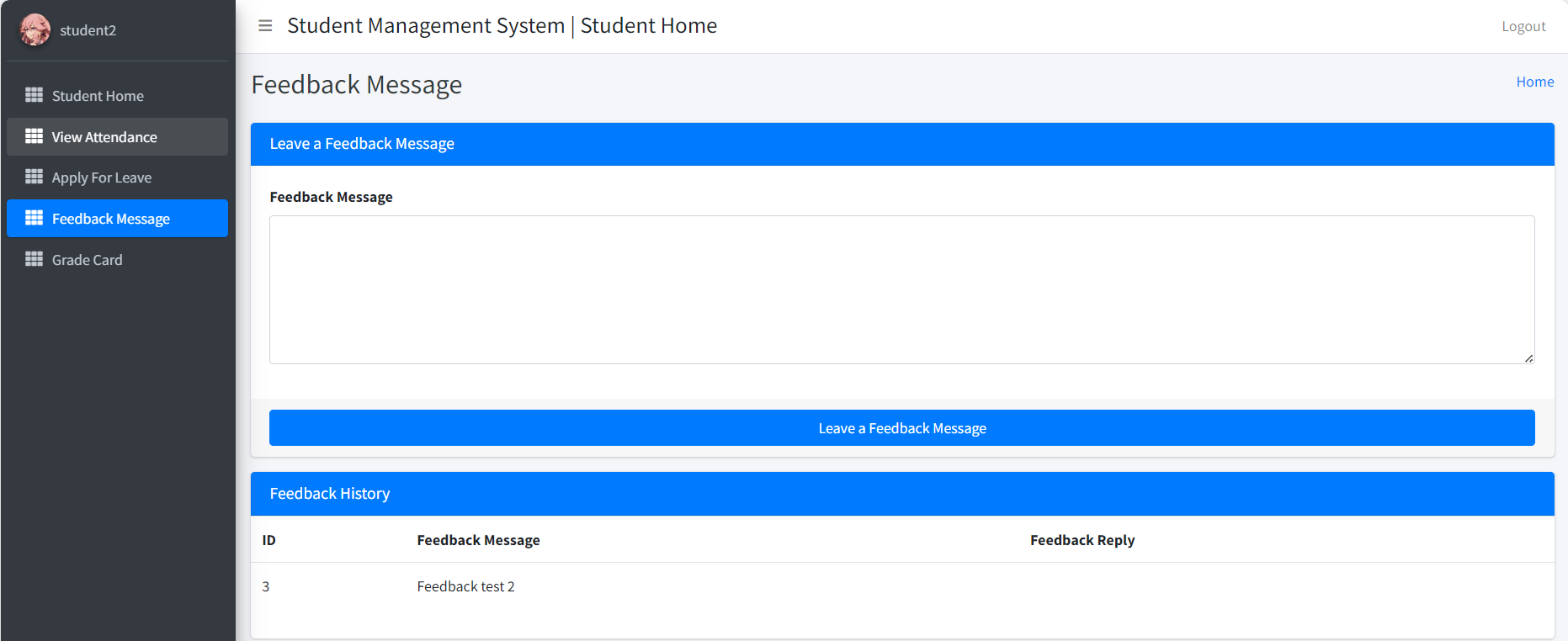
Sau khi chọn ngày và nêu lý do Student gửi yêu cầu nghỉ phép đến Admin



Hình 35

*c, Gửi feedback*

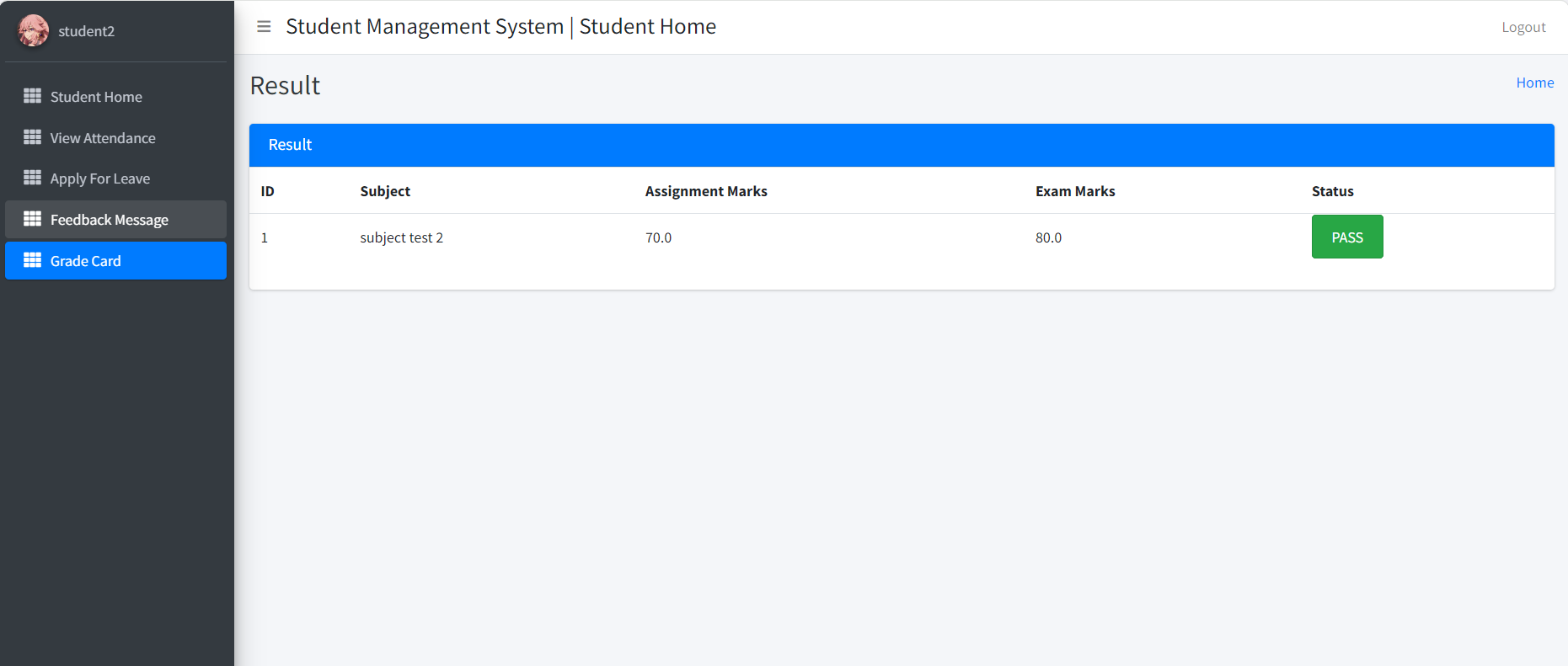
Tương tự Staff Student có thể gửi nhận xét và góp ý tới Admin



Hình 36

*d, Xem điểm các môn học*

Student có thể xem điểm và trạng thái môn học



Hình 37

# **Tổng kết**

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến Giảng viên phụ trách giảng dạy môn: Thực Tập cơ sở - Thầy Nguyễn Hoàng Anh đã truyền đạt ý tưởng, kiến thức và hướng dẫn em hoàn thành đề tài này.

Sau một khoảng thời gian tìm hiểu và thực hiện, cuối cùng sản phẩm “Web quản lí khóa khọc của sinh viên” của em đã hoàn thành. Tuy đã hoàn thành giao diện và các chức năng cơ bản của một web quản lí, bên cạnh đó vẫn còn nhiều thiếu sót, cần được bổ sung và cải tiến. Vì vậy em rất mong nhận được những nhận xét khách quan của thầy về đề tài này.