

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

**KHOA CÔNG NGHỆ THÔNG TIN 1**

-----🙞🙜🕮🙞🙜-----



**ĐỒ ÁN**

**TỐT NGHIỆP ĐẠI HỌC**

**ĐỀ TÀI:**

**XÂY DỰNG HỆ THỐNG QUẢN TRỊ HỌC TẬP LẬP TRÌNH ONLINE**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn** | **: TS. NGUYỄN TẤT THẮNG** |
| **Sinh viên thực hiện** | **: Nguyễn Thanh Hoàng – B20DCCN278**  **Đỗ Huyền Trang – B20DCCN686**  **Trần Khang Huy – B20DCCN326** |
|  |  |
|  |  |
| **Hệ** | **: ĐẠI HỌC CHÍNH QUY** |

***Hà Nội, tháng 12 năm 2024***

# LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành và sâu sắc tới thầy TS. Nguyễn Tất Thắng, người đã trực tiếp hướng dẫn tận tình, chu đáo, chia sẻ những ý kiến và kinh nghiệm quý báu trong suốt quá trình chúng em thực tập và thực hiện đồ án tốt.

Sau đó, chúng em xin gửi lời cảm ơn các thầy, cô trong Học Viện nói chung và khoa CNTT1 nói riêng đã luôn nhiệt huyết, tận tình trong từng bài giảng và tạo điều kiện thuận lợi nhất cho chúng em trong thời gian học tập và nghiên cứu tại trường Học Viện Công Nghệ Bưu Chính Viễn Thông.

Con xin được gửi lời cảm ơn tới bố mẹ và những người thân yêu đã luôn lo lắng, động viên, ủng hộ và tạo điều kiện cho con được học tập tốt. Là chỗ dựa tinh thân và những người tiếp sức cho con có được thành công trong cuộc sống.

Cuối cùng, tôi xin gửi lời chúc tốt đẹp nhất đến những người bạn của tôi và các thầy cô tham gia đợt bảo vệ tốt nghiệp trong khóa này. Chúc cho mọi người luôn vui vẻ và thành công trong cuộc sống.

*Chúng em xin chân thành cảm ơn!*

*Hà Nội, tháng 12 năm 2024*

Sinh viên thực hiện Sinh viên thực hiện Sinhviên thực hiện

Hoàng Huy Trang

Nguyễn Thanh Hoàng Trần Khang Huy Đồ Huyền Trang

# NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

**(Của người hướng dẫn)**

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**Điểm:**…………………… **(bằng chữ:** ……………………………………**)**

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

…………………….**, ngày**….... **tháng**…..... **năm 202…**

**CÁN BỘ GIẢNG VIÊN HƯỚNG DẪN**

# NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

**(Của giáo viên phản biện)**

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**Điểm:**…………………… **(bằng chữ:** ……………………………………**)**

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

…………………….**, ngày**….... **tháng**…..... **năm 202…**

**CÁN BỘ GIẢNG VIÊN PHẢN BIỆN**

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc187101789)

[NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM ii](#_Toc187101790)

[NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM iii](#_Toc187101791)

[MỤC LỤC iv](#_Toc187101792)

[BẢNG VIẾT TẮT VÀ THUẬT NGỮ vi](#_Toc187101793)

[DANH SÁCH HÌNH VẼ vii](#_Toc187101794)

[DANH SÁCH BẢNG ix](#_Toc187101795)

[MỞ ĐẦU 1](#_Toc187101796)

[CHƯƠNG I. GIỚI THIỆU CHUNG 3](#_Toc187101797)

[1.1 Mô tả bài toán 3](#_Toc187101798)

[1.2 Phạm vi dự án 4](#_Toc187101799)

[1.3 Tổng quan về hệ thống 4](#_Toc187101800)

[1.4 Kết luận Chương I 4](#_Toc187101801)

[CHƯƠNG II. ĐẶC TẢ, PHÂN TÍCH YÊU CẦU 6](#_Toc187101802)

[2.1 Phân tích yêu cầu 6](#_Toc187101803)

[2.1.1 Yêu cầu chức năng 6](#_Toc187101804)

[2.1.2 Yêu cầu phi chức năng 11](#_Toc187101805)

[2.2 Biểu đồ use case 13](#_Toc187101806)

[2.3 Kịch bản mô tả use case 13](#_Toc187101807)

[2.3.1 Kịch bản use case các chức năng của sinh viên 13](#_Toc187101808)

[2.3.2 Kịch bản use case các chức năng của giáo viên 18](#_Toc187101809)

[2.3.3 Kịch bản use case các chức năng của admin 22](#_Toc187101810)

[2.4 Trích lớp thực thể 29](#_Toc187101811)

[2.4.1 Trích các danh từ liên quan được mô tả trong hệ thống 29](#_Toc187101812)

[2.4.2 Đánh giá và lựa chọn các danh từ làm lớp thực thể hoặc thuộc tính 30](#_Toc187101813)

[2.5 Kết luận Chương II 31](#_Toc187101814)

[CHƯƠNG III. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG 32](#_Toc187101815)

[3.1 Kiến trúc hệ thống 32](#_Toc187101816)

[3.1.1 Hệ thống quản lý lớp học 33](#_Toc187101817)

[3.1.2 Hệ thống biên dịch thực thi code 34](#_Toc187101818)

[3.1.3 Ưu điểm của kiến trúc 37](#_Toc187101819)

[3.2 Thiết kế cơ sở dữ liệu 37](#_Toc187101820)

[3.3 Thiết kế hoạt động 42](#_Toc187101821)

[3.3.1. Thiết kế biểu đồ tuần tự cho modul xem danh sách lớp học cho tất cả người dùng 42](#_Toc187101822)

[3.3.2 Thiết kế biểu đồ tuần tự cho modul tạo lớp học của quản trị viên 43](#_Toc187101823)

[3.3.3 Thiết kế biểu đồ tuần tự cho modul xem thông tin chi tiết lớp học của tất cả người dùng 44](#_Toc187101824)

[3.4 Công nghệ sử dụng 45](#_Toc187101825)

[3.4.1. C# 45](#_Toc187101826)

[3.4.2. Công nghệ .NET 45](#_Toc187101827)

[3.4.3. SQL Server 46](#_Toc187101828)

[3.4.4. Spring Boot 47](#_Toc187101829)

[3.4.5. Docker 48](#_Toc187101830)

[3.4.6. React 49](#_Toc187101831)

[3.4.7 Javacript 50](#_Toc187101832)

[3.5 Kết luận Chương III 51](#_Toc187101833)

[CHƯƠNG IV. TRIỂN KHAI VÀ KIỂM THỬ HỆ THÔNG 53](#_Toc187101834)

[4.1 Triển khai hệ thống 53](#_Toc187101835)

[4.1.1. Môi Trường Triển Khai: 53](#_Toc187101836)

[4.1.2. Quy Trình Triển Khai: 53](#_Toc187101837)

[4.1.3. Kết Quả Triển Khai: 58](#_Toc187101838)

[4.2 Kiểm thử hệ thống 71](#_Toc187101839)

[4.2.1 Kiểm thử phi chức năng 71](#_Toc187101840)

[4.2.2 Kiểm thử test case 72](#_Toc187101841)

[4.3 Kết luận Chương IV 76](#_Toc187101842)

[CHƯƠNG V. KẾT LUẬN 77](#_Toc187101843)

[5.1 Kết quả đạt được 77](#_Toc187101844)

[5.2 Hạn chế của hệ thống 77](#_Toc187101845)

[5.3 Định hướng phát triển hệ thống 78](#_Toc187101846)

[DANH MỤC TÀI LIỆU THAM KHẢO 79](#_Toc187101847)

# BẢNG VIẾT TẮT VÀ THUẬT NGỮ

|  |  |
| --- | --- |
| **TỪ VIẾT TẮT VÀ THUẬT NGỮ** | **Ý NGHĨA** |
| API | Application Programming Interface |
| Client | Người dùng |
| JSON | JavaScript Object Notation |
| Framework | Thư viện, nền tảng |
| Cloud | Phương thức lưu trữ đám mây |
| Server | Máy chủ |

# 

# DANH SÁCH HÌNH VẼ

[Hình 2.1 Usecase chung 13](#_Toc186530496)

[Hình 3.1 Kiến trúc tổng thể hệ thống 32](#_Toc186530497)

[Hình 3.2 Thiết kế cơ sở dữ liệu của hệ thống 36](#_Toc186530498)

[Hình 4.1 Lệnh cài đặt docker-compose 50](#_Toc186530499)

[Hình 4.2 docker-compose file của hệ thống back-end 51](#_Toc186530500)

[Hình 4.3 Dockerfile của hệ thống thực thi code 53](#_Toc186530501)

[Hình 4.4 docker-compose file của hệ thống thực thi code 54](#_Toc186530502)

[Hình 4.5 Giao diện trang chủ chính 55](#_Toc186530503)

[Hình 4.6 Giao diện trang chủ admin, giao diện xem danh sách môn học 55](#_Toc186530504)

[Hình 4.7 Giao diện tạo môn học 56](#_Toc186530505)

[Hình 4.8 Giao diện xem chi tiết môn học 56](#_Toc186530506)

[Hình 4.9 Giao diện xem chi tiết lớp học 57](#_Toc186530507)

[Hình 4.10 Giao diện xem các chủ đề đã mở của lớp học 57](#_Toc186530508)

[Hình 4.11 Giao diện chỉnh sửa môn học 58](#_Toc186530509)

[Hình 4.12 Giao diện xem danh sách lớp học 58](#_Toc186530510)

[Hình 4.13 Giao diện tạo lớp học 59](#_Toc186530511)

[Hình 4.14 Giao diện xem chi tiết lớp học 59](#_Toc186530512)

[Hình 4.15 Giao diện xem danh sách bài tập 60](#_Toc186530513)

[Hình 4.16 Giao diện tạo bài tập 60](#_Toc186530514)

[Hình 4.17 Giao diện xem chi tiết bài tập của admin 61](#_Toc186530515)

[Hình 4.18 Giao diện cập nhật bài tập 61](#_Toc186530516)

[Hình 4.19 Giao diện tải danh sách sinh viên từ file excel 62](#_Toc186530517)

[Hình 4.20 Xem chi tiết một sinh viên 62](#_Toc186530518)

[Hình 4.21 Xem danh sách các lớp theo học của sinh viên 63](#_Toc186530519)

[Hình 4.22 Xem lịch sử nộp bài của sinh viên 63](#_Toc186530520)

[Hình 4.23 Sinh viên xem các lớp mình đang theo học 64](#_Toc186530521)

[Hình 4.24 Sinh viên xem chi tiết lớp học 64](#_Toc186530522)

[Hình 4.25 Sinh viên xem lịch sử nộp bài của mình 65](#_Toc186530523)

[Hình 4.26 Sinh viên xem các chủ đề đã mở 65](#_Toc186530524)

[Hình 4.27 Sinh viên xem chi tiết bài tập của chủ đề 66](#_Toc186530525)

[Hình 4.28 Sinh viên nộp bài và hiển thị kết quả 66](#_Toc186530526)

[Hình 4.29 Kết quả chạy Lighthouse 67](#_Toc186530527)

# DANH SÁCH BẢNG

[Bảng 2.1 Yêu cầu chức năng chung 6](#_Toc186534120)

[Bảng 2.2 Yêu cầu chức năng của sinh viên 7](#_Toc186534121)

[Bảng 2.3 Yêu cầu chức năng của giảng viên 8](#_Toc186534122)

[Bảng 2.4 Yêu cầu chức năng của quản trị viên 10](#_Toc186534123)

[Bảng 2.5 Yêu cầu phi chức năng 11](#_Toc186534124)

[Bảng 2.6 Kịch bản chức năng đăng nhập của sinh viên 13](#_Toc186534125)

[Bảng 2.7 Kịch bản chức năng xem danh sách các lớp theo học của sinh viên 14](#_Toc186534126)

[Bảng 2.8 Kịch bản chức năng xem chi tiết lớp học 15](#_Toc186534127)

[Bảng 2.9 Kịch bản chức năng xem danh sách các bài tập đã mở của lớp 15](#_Toc186534128)

[Bảng 2.10 Kịch bản chức năng xem chi tiết bài tập 16](#_Toc186534129)

[Bảng 2.11 Kịch bản chức năng làm và nộp bài tập 17](#_Toc186534130)

[Bảng 1.12 Kịch bản chức năng xem lịch sử nộp bài tập 17](#_Toc186534131)

[Bảng 2.13 Kịch bản chức năng xem danh sách sinh viên của lớp học 18](#_Toc186534132)

[Bảng 2.14 Kịch bản chức năng xem danh sách các môn giảng dạy 18](#_Toc186534133)

[Bảng 2.15 Kịch bản chức năng xem chi tiết một môn học 19](#_Toc186534134)

[Bảng 2.16 Kịch bản chức năng xem danh sách các lớp của môn học 19](#_Toc186534135)

[Bảng 2.17 Kịch bản chức năng xem danh sách bài tập của môn học 20](#_Toc186534136)

[Bảng 2.18 Kịch bản chức năng thêm bài tập 20](#_Toc186534137)

[Bảng 2.19 Kịch bản chức năng thêm lớp học mới 21](#_Toc186534138)

[Bảng 2.20 Kịch bản chức năng xem danh sách tất cả các môn học 22](#_Toc186534139)

[Bảng 2.21 Kịch bản chức năng xem chi tiết một môn học 22](#_Toc186534140)

[Bảng 2.22 Kịch bản chức năng xem danh sách lớp của môn học 23](#_Toc186534141)

[Bảng 2.23 Kịch bản chức năng xem danh sách bài tập của môn học 23](#_Toc186534142)

[Bảng 2.24 Kịch bản thêm bài tập của môn học 23](#_Toc186534143)

[Bảng 2.25 Kịch bản chức năng import file excel sinh viên 24](#_Toc186534144)

[Bảng 2.26 Kịch bản chức năng thêm lớp học mới 25](#_Toc186534145)

[Bảng 2.27 Kịch bản chức năng xem danh sách giáo viên 25](#_Toc186534146)

[Bảng 2.28 Kịch bản Tạo một giáo viên 26](#_Toc186534147)

[Bảng 2.29 Kịch bản xem chi tiết giáo viên 27](#_Toc186534148)

[Bảng 2.30 Kịch bản xem danh sách các lớp đang giảng dạy 27](#_Toc186534149)

[Bảng 2.31 Kịch bản thêm tài liệu môn học 28](#_Toc186534150)

[Bảng 2.32 Kịch bản tải tài liệu 28](#_Toc186534151)

[Bảng 3.1 Cấu trúc bảng User 37](#_Toc186534152)

[Bảng 2.2 Cấu trúc bảng Admin 37](#_Toc186534153)

[Bảng 3.3 Cấu trúc bảng Student 37](#_Toc186534154)

[Bảng 3.4 Cấu trúc bảng Teacher 37](#_Toc186534155)

[Bảng 3.5 Cấu trúc bảng StudentSubmission 37](#_Toc186534156)

[Bảng 3.6 Cấu trúc bảng Class 37](#_Toc186534157)

[Bảng 3.7 Cấu trúc bảng Subject 38](#_Toc186534158)

[Bảng 3.8 Cấu trúc bảng University 38](#_Toc186534159)

[Bảng 3.9 Cấu trúc bảng ClassTopicOpen 38](#_Toc186534160)

[Bảng 3.10 Cấu trúc bảng StudentClass 38](#_Toc186534161)

[Bảng 3.11 Cấu trúc bảng Department 39](#_Toc186534162)

[Bảng 3.12 Cấu trúc bảng Exercise 39](#_Toc186534163)

[Bảng 3.13 Cấu trúc bảng SubjectExercise 39](#_Toc186534164)

[Bảng 3.14 Cấu trúc bảng ClassExercise 39](#_Toc186534165)

[Bảng 3.15 Cấu trúc bảng Topic 40](#_Toc186534166)

[Bảng 3.16 Cấu trúc bảng TestCase 40](#_Toc186534167)

[Bảng 3.17 Cấu trúc bảng SubjectProgrammingLanguage 40](#_Toc186534168)

[Bảng 3.18 Cấu trúc bảng ProgrammingLanguage 40](#_Toc186534169)

[Bảng 4.1: Bảng kiểm thử testcase 68](#_Toc186534170)

# MỞ ĐẦU

Trong bối cảnh giáo dục hiện đại, việc ứng dụng công nghệ vào quá trình giảng dạy và quản lý học tập trở thành một yếu tố thiết yếu, góp phần nâng cao hiệu quả giảng dạy và sự tương tác giữa giáo viên và sinh viên. Đặc biệt, đối với các môn học lập trình, việc số hóa các quy trình như giao bài tập, nộp mã nguồn và chấm điểm tự động không chỉ giúp giảm bớt gánh nặng về công việc hành chính mà còn tạo ra một môi trường học tập thuận tiện, hiệu quả hơn.

Với sự phát triển không ngừng của công nghệ, nhu cầu về một hệ thống quản lý học lập trình trực tuyến ngày càng trở nên cấp thiết. Hệ thống này không chỉ cung cấp cho giảng viên các công cụ quản lý khóa học, tài liệu, và đánh giá sinh viên, mà còn tích hợp môi trường lập trình trực tuyến, cho phép sinh viên thực hành và nộp bài tập một cách linh hoạt. Qua đó, hệ thống tạo ra một giải pháp tối ưu trong việc hỗ trợ giảng dạy và học tập lập trình tại các trường đại học.

Đồ án này sẽ phân tích và thiết kế một hệ thống quản lý lớp học lập trình trực tuyến, nhằm mang lại giải pháp công nghệ cho việc tổ chức, theo dõi tiến độ học tập, và tự động hóa các quy trình liên quan đến học lập trình. Mục tiêu của đồ án là xây dựng một nền tảng web thân thiện với người dùng, dễ dàng sử dụng cho cả giáo viên và sinh viên, đồng thời có khả năng mở rộng trong tương lai để đáp ứng các nhu cầu phát sinh.

Nội dung của đồ án Xây dựng hệ thống quản trị học tập lập trình online bao gồm các phần sau:

**Chương I: Giới thiệu chung**

Giới thiệu tổng quan về mục đích và bối cảnh của đề tài, lý do chọn xây dựng hệ thống quản lý học tập lập trình trực tuyến. Nêu lên các thách thức hiện tại trong việc học và dạy lập trình, cùng lợi ích khi ứng dụng công nghệ vào giáo dục.

**Chương II: Đặc tả, phân tích yêu cầu**

Mô tả chi tiết các yêu cầu chức năng và phi chức năng của hệ thống. Đặc tả các yêu cầu của người dùng, đồng thời phân tích quy trình nghiệp vụ để hiểu rõ hơn về hệ thống.

**Chương III: Thiết kế và xây dựng hệ thống**

Trình bày quá trình thiết kế kiến trúc hệ thống, bao gồm các thành phần chính, mô hình cơ sở dữ liệu, và giao diện người dùng. Phác thảo quá trình triển khai hệ thống, tập trung vào xây dựng từng module.

**Chương IV: Triển khai và kiểm thử hệ thống**

Mô tả quá trình cài đặt và triển khai hệ thống trên môi trường thực tế, cùng với việc kiểm thử tính năng và hiệu suất. Đưa ra kết quả kiểm thử và đánh giá mức độ hoàn thiện của hệ thống.

**Chương V: Kết luận**

Tóm tắt các kết quả đạt được, những hạn chế của hệ thống, rút ra những bài học từ quá trình thực hiện, đồng thời đề xuất các hướng phát triển trong tương lai để hoàn thiện và mở rộng hệ thống quản lý học tập.

# CHƯƠNG I. GIỚI THIỆU CHUNG

Giới thiệu tổng quan về mục đích và bối cảnh của đề tài, lý do chọn xây dựng hệ thống quản lý học tập lập trình trực tuyến. Nêu lên các thách thức hiện tại trong việc học và dạy lập trình, cùng lợi ích khi ứng dụng công nghệ vào giáo dục.

## Mô tả bài toán

Trong quá trình giảng dạy lập trình tại các trường đại học, giáo viên thường gặp phải nhiều thách thức trong việc quản lý các lớp học và đánh giá kết quả học tập của sinh viên. Việc theo dõi tiến độ của từng sinh viên, giao bài tập lập trình, chấm điểm thủ công, và quản lý nhiều lớp học cùng lúc trở thành một nhiệm vụ phức tạp và tốn nhiều thời gian. Thêm vào đó, sinh viên gặp khó khăn trong việc nộp bài tập lập trình và cài đặt môi trường phát triển tại máy cá nhân, dẫn đến sự chậm trễ và khó khăn trong quá trình học tập.

Hệ thống hiện tại thiếu sự tích hợp các công cụ giúp tự động hóa các quy trình liên quan đến giảng dạy lập trình, chẳng hạn như giao bài, nộp bài dưới dạng mã nguồn, và chấm điểm tự động. Điều này dẫn đến việc giáo viên phải dành nhiều thời gian cho công việc hành chính và chấm điểm, thay vì tập trung vào việc hướng dẫn sinh viên. Đồng thời, sinh viên không có được môi trường lập trình thống nhất và thuận tiện để thực hành, gây khó khăn trong việc nắm bắt kiến thức và cải thiện kỹ năng lập trình.

Vì vậy, bài toán đặt ra là xây dựng một hệ thống quản lý học lập trình trực tuyến, giúp giải quyết các vấn đề trên bằng cách:

* Hỗ trợ giảng viên tạo và quản lý các khóa học lập trình một cách hiệu quả, bao gồm giao bài tập, tải tài liệu, và theo dõi tiến độ học tập của sinh viên.
* Tích hợp môi trường lập trình trực tuyến, cho phép sinh viên viết, chạy và nộp mã nguồn trực tiếp trên hệ thống mà không cần cài đặt phần mềm tại máy cá nhân.
* Cung cấp chức năng chấm điểm tự động, giúp giáo viên tiết kiệm thời gian trong việc đánh giá kết quả học tập của sinh viên thông qua các bộ kiểm thử tự động.

Việc giải quyết bài toán này không chỉ giúp tối ưu hóa quy trình giảng dạy, mà còn tạo ra một môi trường học tập linh hoạt và thuận lợi cho cả giảng viên và sinh viên trong lĩnh vực lập trình.

## 1.2 Phạm vi dự án

* Hệ thống là một chương trình trên nền tảng web, chỉ được sử dụng cho nội bộ trường đại học.
* Hệ thống chỉ hỗ trợ quản lý một trường đại học.
* Chỉ giảng viên, sinh viên và quản trị viên của một trường đại học có thể sử dụng chương trình.

## 1.3 Tổng quan về hệ thống

Ứng dụng website quản lý học tập lập trình trực tuyến giải quyết một số vấn đề thực tế trong lĩnh vực giáo dục và đào tạo lập trình, cụ thể như sau:

* **Quản lý khóa học**:  Hệ thống cho phép giảng viên dễ dàng tạo và quản lý các khóa học lập trình, bao gồm việc tải lên tài liệu học tập, tạo bài kiểm tra, bài tập lập trình, và cung cấp bài giảng trực tuyến. Các khóa học được tổ chức và quản lý một cách hệ thống, giúp giảng viên kiểm soát tốt hơn nội dung giảng dạy.
* **Tự động hóa chấm điểm và đánh giá**: Hệ thống tích hợp công cụ chấm điểm tự động giúp giảng viên tiết kiệm thời gian và công sức. Thay vì phải chấm điểm thủ công, hệ thống sử dụng các bộ kiểm thử (test case) tự động để đánh giá bài tập lập trình của sinh viên, đảm bảo tính khách quan và chính xác trong việc đánh giá năng lực.
* **Theo dõi tiến độ học tập**: Hệ thống cung cấp chức năng theo dõi và báo cáo chi tiết về quá trình học tập của từng sinh viên. Giảng viên có thể dễ dàng xem các báo cáo về tiến độ, điểm số và kết quả làm bài của sinh viên, giúp họ có cái nhìn toàn diện và chính xác về năng lực và sự tiến bộ của mỗi học viên.
* **Cung cấp môi trường lập trình trực tuyến**: Hệ thống tích hợp sẵn môi trường lập trình trực tuyến, cho phép sinh viên thực hành viết và chạy mã trực tiếp trên nền tảng mà không cần cài đặt phần mềm. Điều này giúp giải quyết các vấn đề liên quan đến cài đặt môi trường lập trình cá nhân, đồng thời tạo điều kiện thuận lợi cho việc học và thực hành lập trình trực tuyến.

## 1.4 Kết luận Chương I

Chương I đã giới thiệu tổng quan về mục đích và bối cảnh của đề tài "Xây dựng ứng dụng web quản lý học lập trình trực tuyến", làm rõ lý do chọn đề tài cũng như những thách thức hiện tại trong việc dạy và học lập trình. Việc ứng dụng công nghệ vào giáo dục, đặc biệt là xây dựng một hệ thống quản lý học tập trực tuyến, là một giải pháp thiết yếu để giải quyết các khó khăn mà giáo viên và sinh viên đang gặp phải, như quản lý lớp học, giao và chấm bài tập lập trình, theo dõi tiến độ học tập và cung cấp môi trường lập trình trực tuyến.

Qua các phân tích trên, rõ ràng rằng việc phát triển hệ thống quản lý học lập trình trực tuyến không chỉ giúp tăng cường hiệu quả giảng dạy mà còn tạo điều kiện thuận lợi cho sinh viên trong việc học tập và thực hành lập trình. Đây chính là nền tảng để đề xuất giải pháp kỹ thuật chi tiết ở các chương tiếp theo, nhằm hiện thực hóa hệ thống đáp ứng các nhu cầu đặt ra của con người.

# CHƯƠNG II. ĐẶC TẢ, PHÂN TÍCH YÊU CẦU

Chương II của đồ án tập trung vào việc đặc tả và phân tích yêu cầu cho hệ thống quản lý học tập lập trình trực tuyến. Đây là bước quan trọng nhằm đảm bảo rằng hệ thống được phát triển sẽ đáp ứng đầy đủ các nhu cầu của người dùng và hoạt động hiệu quả trong môi trường thực tế. Trong bối cảnh giáo dục hiện đại, việc xây dựng một hệ thống quản lý học tập hiệu quả đòi hỏi sự hiểu biết sâu sắc về các yêu cầu của người dùng, bao gồm giảng viên, sinh viên, và quản trị viên. Chương này sẽ đi sâu vào việc xác định và phân tích các yêu cầu đó, từ đó tạo nền tảng cho việc thiết kế và phát triển hệ thống.

## 2.1 Phân tích yêu cầu

Các tác nhân sử dụng hệ thống:

* Quản trị viên
* Giảng viên
* Sinh viên

### 2.1.1 Yêu cầu chức năng

2.1.1.1 Yêu cầu chức năng chung

Đây là các chức năng chung dành cho người dùng hệ thống, ngoại trừ các chức năng đăng ký, đăng nhập và đăng xuất thì các chức năng còn lại là tương tự giữa tác nhân khách và tác nhân người dùng đã có tài khoản.

Bảng 2.1 Yêu cầu chức năng chung

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên chức năng** | **Mô tả** |
| 1  1 | Đăng nhập | Cho phép người dùng sử dụng tài khoản đã có để đăng nhập vào hệ thống và sử dụng các chức năng theo quyền của người dùng |
| 2 | Đăng xuất | Cho phép người dùng thoát khỏi tài khoản hiện tại và quay và trang đăng nhập |
| 3 | Quên mật khẩu | Cho phép người dùng lấy lại mật khẩu khi quên |
| 4 | Xem thông tin người dùng | Cho phép người dùng xem các thông tin của tài khoản hiện tại bao gồm tên, địa chỉ, số điện thoại, email |
| 5 | Đổi mật khẩu | Cho phép người dùng đổi mật khẩu của tài khoản hiện tại |

2.1.1.2. Yêu cầu chức năng của sinh viên

Bảng 2.2 Yêu cầu chức năng của sinh viên

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên chức năng** | **Mô tả** |
| 1 | Xem danh sách các lớp đã theo học | Cho phép sinh viên xem danh sách toàn bộ các lớp đã và đang theo học. Có thể tìm kiếm, sắp xếp các lớp theo các tiêu chí khác nhau |
| 2 | Truy cập vào lớp đã theo học | Cho phép sinh viên truy cập vào lớp đã theo học. Xem thông tin chi tiết về lớp như tên, thời gian, sĩ số, … |
| 3 | Xem danh sách các bài tập đã mở của lớp học | Cho phép sinh viên xem danh sách các bài tập đã mở của môn học |
| 4 | Xem chi tiết bài tập | Cho phép sinh viên xem thông tin chi tiết của bài tập như đề bài, yêu cầu, một vài trường hợp kiểm thử và và cung cấp môi trường lập trình cho bài tập |
| 5 | Làm và nộp bài tập | Cho phép sinh viên lập trình và nộp code, sau đó trả lại kết quả bài tập cho sinh viên |
| 6 | Xem danh sách các tài liệu đã mở của môn học | Cho phép sinh viên xem danh sách các tài liệu của lớp học |
| 7 | Tải về tài liệu của lớp học | Cho phép sinh viên tải về tài liệu của lớp học |

2.1.1.3. Yêu cầu chức năng của giảng viên

Bảng 2.3 Yêu cầu chức năng của giảng viên

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên chức năng** | **Mô tả** |
| 1 | Xem danh sách các môn giảng dạy | Cho giáo viên xem danh sách các môn đã và đang giảng dạy, bao gồm các thông tin như tên môn, tên khoa, số lượng lớp đang trong thời gian giảng dạy |
| 2 | Xem thông tin chi tiết các môn | Cho giáo viên xem thông tin chi tiết của môn học như tên môn, tên khoa, số lớp hiện đang giảng dạy |
| 3 | Xem danh sách tài liệu môn học | Cho giáo viên xem danh sách tài liệu của môn học, bao gồm các thông tin như tên tài liệu, ngày thêm, người thêm, lượt xem, … |
| 4 | Tải về tài liệu của môn học | Cho phép giáo viên tải tài liệu của môn học về máy |
| 5 | Thêm tài liệu môn học | Cho phép giáo viên thêm tài liệu cho môn học |
| 6 | Chỉnh sửa tài liệu môn học | Cho phép giáo viên chỉnh sửa tài liệu do chính giáo viên tải lên |
| 6 | Xem danh sách bài tập | Cho phép giáo viên xem danh sách toàn bộ các bài tập của môn học cùng các thông tin như người thêm, ngày thêm |
| 7 | Xem thông tin chi tiết bài tập | Cho phép giáo viên xem thông tin chi tiết của bài tập như đề bài, yêu cầu, các trường hợp kiểm thử |
| 8 | Chỉnh sửa bài tập | Cho phép giáo viên thay đổi đề bài, yêu cầu, các trường hợp kiểm thử của bài tập do giáo viên tạo |
| 9 | Thêm bài tập | Cho phép giáo viên thêm bài tập cho môn học |
| 10 | Xem danh sách các lớp | Cho phép giáo viên xem các lớp đã và đang giảng dạy, có thể lọc, tìm kiếm theo các tiêu chí khác nhau |
| 11 | Xem thông tin chi tiết của lớp | Giảng viên xem thông tin chi tiết của lớp như ngày bắt đầu, ngày kết thúc, tình trạng lớp học, sĩ số lớp |
| 12 | Xem danh sách chương mở bài tập cho lớp | Giảng viên xem danh sách các chương và bài tập mở cho lớp, mỗi hàng có các thông tin như số chương, số lượng bài tập, thời gian làm |
| 13 | Xem chi tiết và sửa đổi các chương mở bài tập cho lớp | Giảng viên có thể chọn các chương và bài tập cho lớp và thay đổi thời gian làm bài |
| 14 | Mở thêm bài tập theo chương cho lớp | Giảng viên có thể chọn các chương để mở các bài tập của chương đó cho lớp |
| 15 | Xem danh sách các tài liệu đã mở | Giảng viên có thể xem thông tin các tài liệu đã mở cho lớp |
| 16 | Thêm tài liệu đã mở cho lớp | Giảng viên có thể thêm tài liệu cho lớp truy cập |
| 17 | Xem danh sách các sinh viên trong lớp | Cho phép giảng viên xem danh sách toàn bộ các sinh viên trong lớp với các thông tin như tên, mã sinh viên, số bài tập đã hoàn thành |
| 18 | Xem thông tin chi tiết của sinh viên | Cho phép giảng viên chọn một sinh viên để xem thông tin chi tiết, có thông tin về số bài tập sinh viên đã làm, số bài tập đã hoàn thành, số bài tập còn bị sai |
| 19 | Xem danh sách các lần nộp code của sinh viên | Cho phép giảng viên xem danh sách những lần nộp code của sinh viên với các thông tin như tên bài tập, thời gian nộp, kết quả nộp |
| 20 | Xem thông tin chi tiết về lần nộp code của sinh viên | Cho phép giảng viên xem thông tin chi tiết về lần nộp code của sinh viên, bao gồm cả mã nguồn |

2.1.1.4. Yêu cầu chức năng của quản trị viên

Bảng 2.4 Yêu cầu chức năng của quản trị viên

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên chức năng** | **Mô tả** |
| 1 | Thêm khoa | Cho phép quản trị viên tạo thêm khoa |
| 2 | Thêm môn học | Cho phép quản trị viên tạo thêm môn học |
| 3 | Thay đổi thông tin môn học | Cho phép quản trị viên thay đổi thông tin của môn học |
| 4 | Thêm lớp học | Cho phép quản trị viên tạo lớp học và gán giảng viên cho lớp học |
| 5 | Thay đổi thông tin lớp học | Cho phép quản trị viên thay đổi thông tin của lớp học |
| 6 | Thêm giảng viên | Cho phép quản trị viên tạo tài khoản, thông tin giảng viên mới |
| 7 | Thay đổi thông tin giảng viên | Cho phép quản trị viên thay đổi thông tin giảng viên |
| 8 | Nhập thông tin sinh viên qua file excel | Cho phép quản trị viên tạo một loạt tài khoản sinh viên mới thông qua file excel chứa thông tin sinh viên |
| 9 | Xem danh sách sinh viên | Cho phép quản trị viên xem danh sách toàn bộ thông tin sinh viên và có thể lọc, tìm kiếm theo các tiêu chí |
| 10 | Thay đổi thông tin sinh viên | Cho phép quản trị viên thay đổi thông tin sinh viên |
| 11 | Xem danh sách tài liệu của | Cho phép quản trị viên xem danh sách tài liệu cho môn học |
| 12 | Thêm tài liệu cho môn học | Cho phép quản trị viên thêm tài liệu cho môn học |
| 13 | Thay đổi tài liệu cho môn học | Cho phép quản trị viên thay đổi tài liệu bất kỳ của môn học |
| 14 | Xem danh sách bài tập của môn học | Cho phép quản trị viên xem danh sách toàn bộ các bài tập của môn học |
| 15 | Thêm bài tập cho môn học | Cho phép quản trị viên thêm bài tập cho môn học |
| 16 | Thay đổi bài tập | Cho phép quản trị viên thay đổi bài tập bất kỳ |
| 17 | Thêm tài liệu cho lớp học | Cho phép quản trị viên thêm tài liệu cho lớp học |
| 18 | Tải tài liệu lớp học | Cho phép quản trị viên tải tài liệu lớp học |
| 19 | Xem danh sách giáo viên | Cho phép quản trị viên xem danh sách giáo viên |
| 20 | Tạo giáo viên | Cho phép quản trị viên tạo giáo viên |
| 21 | Xem chi tiết giáo viên | Cho phép quản trị xem chi tiết 1 giáo viên |

### 2.1.2 Yêu cầu phi chức năng

Bảng 2.5 Yêu cầu phi chức năng

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên chức năng** | **Mô tả** |
| 1 | Tính ổn định | Hệ thống cần đảm bảo các tính năng có thể được sử dụng liên tục mà không gặp gián đoạn, đảm bảo hiệu suất hoạt động. |
| 2 | Khả năng mở rộng | Hệ thống cần được thiết kế để có thể mở rộng khi số lượng người dùng tăng lên, đảm bảo không ảnh hướng tới hiệu suất hoạt động. |
| 3 | Bảo mật | Hệ thống cần đảm bảo an toàn cho dữ liệu cá nhân của người dùng, bao gồm thông tin cá nhân, mật khẩu, email, lịch sử, đảm bảo mã hóa dữ liệu xác thực để phòng tránh trường hợp rò rỉ thông tin bảo mật. |
| 4 | Hiệu năng | Hệ thống cần đảm bảo xử lý nhanh chóng để đáp ứng nhu cầu của người dùng. |
| 5 | Giao diện | Hệ thống cần có giao diện thân thiện với người dùng, dễ học dễ sử dụng. |

## 2.2 Biểu đồ use case

A diagram of a diagram

Description automatically generated

Hình 2.1 Usecase chung

2.3 Kịch bản mô tả use case

### 2.3.1 Kịch bản use case các chức năng của sinh viên

2.3.1.1. Use case “Đăng nhập”

A diagram of an extension point

Description automatically generated

Hình 2.2 Use case “Đăng nhập”

Bảng 2.6 Kịch bản chức năng đăng nhập của sinh viên

|  |  |
| --- | --- |
| Tên ca sử dụng | Đăng nhập |
| Mô tả | Chức năng cho phép người dùng đăng nhập vào hệ thống để có thể sử dụng các chức năng cần định danh như: xem thông tin lớp học, giảng viên, sinh viên, sửa thông tin, thêm lớp, thêm bài tập... |
| Tác nhân | - Người dùng khách. |
| Tiền điều kiện | - Người dùng đã truy cập được vào trang đăng nhập. |
| Hậu điều kiện | - Người dùng đăng nhập thành công vào hệ thống. |
| Scenario | 1. Tại giao diện đăng nhập, người dùng nhập username và password để login vào hệ thống 2. Đăng nhập thành công màn hình chuyển trang tới trang chủ chính |
| Error | 1. Đối với người dùng là quản trị viên, ở bước 3 hệ thống sẽ chuyển hướng người dùng tới trang quản trị hệ thống   2. Nếu hệ thống phát hiện lỗi trong các trường mà người dùng đã nhập, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin |

2.3.1.2. Chức năng xem danh sách các lớp theo học

A diagram of a diagram

Description automatically generated

Hình 2.3 Use case xem danh sách các lớp học phần đang theo học

Bảng 2.7 Kịch bản chức năng xem danh sách các lớp theo học của sinh viên

|  |  |
| --- | --- |
| Use case | Xem danh sách các lớp theo học |
| Actor | Sinh viên |
| Tiền điều kiện | Sinh viên đăng nhập vào hệ thống |
| Hậu điều kiện | Sinh viên xem danh sách các lớp đã và đang theo học |
| Scenario | 1. Sinh viên chọn chức năng “Danh sách lớp học” trên menu 2. Hệ thống hiển thị danh sách các lớp học sinh đã và đang theo học, mỗi lớp có các thông tin như tên môn, thời gian học, tình trạng, … 3. Sinh viên nhập từ khóa tìm kiếm, chọn bộ lọc để tìm ra lớp mong muốn |
| Error | 2. 1. Sinh viên chưa tham gia lớp học nào       2. Hệ thống hiển thị “không tìm thấy lớp học”  3. 3. Hệ thống không tìm thấy lớp chứa từ khóa sinh viên nhập hệ thống hiển thị “không tìm thấy lớp học” |

2.3.1.3. Chức năng xem chi tiết lớp học

A diagram of a diagram

Description automatically generated

Hình 2.4 Use case xem chi tiết một lớp học

Bảng 2.8 Kịch bản chức năng xem chi tiết lớp học

|  |  |
| --- | --- |
| Use case | Xem chi tiết lớp học |
| Actor | Sinh viên |
| Tiền điều kiện | Sinh viên đăng nhập vào hệ thống và đã tìm được lớp mong muốn từ module xem danh sách các lớp theo học |
| Hậu điều kiện | Sinh viên vào trang chi tiết của lớp học |
| Scenario | 1. Sinh viên nhấn vào icon xem của lớp từ danh sách các lớp học đã tham gia 2. Hệ thống hiển thị trang thông tin chi tiết của lớp học bao gồm các thông tin chung như tên lớp, thời gian học, tình trạng lớp, ... Ngoài ra giao diện có 2 nút chọn “danh sách lớp” và “chủ đề đã mở ” để mở 2 tab con cho danh sách sinh viên và chủ đề đã mở của lớp |
| Error |  |

2.3.1.4. Chức năng xem danh sách các bài tập đã mở của lớp học

A diagram of a diagram

Description automatically generated

Hình 2.5 Use case xem danh sách các bài tập đã mở của lớp học

Bảng 2.9 Kịch bản chức năng xem danh sách các bài tập đã mở của lớp

|  |  |
| --- | --- |
| Use case | Xem danh sách các bài tập đã mở của lớp học |
| Actor | Sinh viên |
| Tiền điều kiện | Sinh viên đã truy cập vào trang thông tin chi tiết của lớp học |
| Hậu điều kiện | Sinh viên xem được danh sách các bài tập đã mở của lớp học |
| Scenario | 1. Từ màn hình chi tiết của lớp học, sinh viên chọn chức năng xem danh sách bài tập 2. Bảng bên dưới thông tin của lớp học hiện danh sách các bài tập đã mở của lớp học, mỗi bài tập có các thông tin như mã bài tập, tên bài tập, trạng thái, thời gian mở, ... và có phần nhập tìm kiếm, cách sắp xếp 3. Sinh viên nhập tên bài tập muốn tìm và sắp xếp danh sách theo các tiêu chí |
| Error | 2. Lớp chưa được mở bài tập nào       2.1 Hệ thống hiện “không tìm thấy bài tập”  3. Không tìm thấy bài tập theo tên hoặc mã sinh viên đã nhập       3.1 Hệ thống hiện “không tìm thấy bài tập” |

2.3.1.5. Chức năng xem chi tiết 1 bài tập

A diagram of a diagram

Description automatically generated

Hình 2.6 Use case xem chi tiết 1 bài tập

Bảng 2.10 Kịch bản chức năng xem chi tiết bài tập

|  |  |
| --- | --- |
| Use case | Xem thông tin chi tiết của bài tập |
| Actor | Sinh viên |
| Tiền điều kiện | Sinh viên đã xem danh sách các bài tập của lớp học |
| Hậu điều kiện | Sinh viên xem được chi tiết của bài tập |
| Scenario | 1. Sinh viên tìm được bài tập mong muốn từ module danh sách các bài tập đã mở của lớp 2. Sinh viên nhấn vào bài tập trên danh sách để xem chi tiết 3. Hệ thống chuyển sang trang thông tin chi tiết bài tập. Nửa bên trái là đề bài, yêu cầu của bài tập, nửa bên phải là môi trường để lập trình |
| Error | **1.Bài tập không tồn tại hoặc bị xóa**: Khi sinh viên nhấn vào một bài tập, nhưng bài tập đó đã bị xóa hoặc không còn tồn tại.  2. Màn hình Hiển thị thông báo lỗi "Bài tập không tồn tại" và quay lại danh sách bài tập. |

2.3.1.6. Chức năng làm và nộp bài tập

A diagram of a diagram

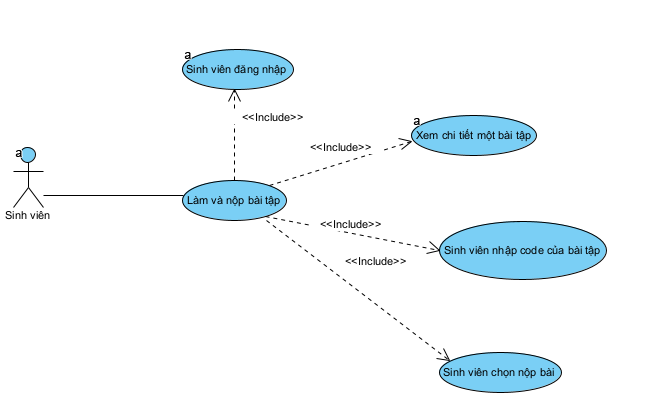
Description automatically generated

Hình 2.7 Usecase làm và nộp bài tập

Bảng 2.11 Kịch bản chức năng làm và nộp bài tập

|  |  |
| --- | --- |
| Use case | Làm và nộp bài tập |
| Actor | Sinh viên |
| Tiền điều kiện | Sinh viên đã vào trang chi tiết của bài tập |
| Hậu điều kiện | Sinh viên làm và nộp bài tập |
| Scenario | 1. Sinh viên viết mã lập trình cho bài tập tại phần môi trường lập trình trong trang chi tiết của bài tập 2. Sinh viên ấn nộp bài 3. Hệ thống trả lại kết quả của bài sinh viên đã nộp |
| Error | Nếu sinh viên viết có lỗi cú pháp hoặc không đáp ứng yêu cầu bài tập màn hình sẽ hiển thị thông báo "Mã lập trình không hợp lệ”. Vui lòng kiểm tra và thử lại |

2.3.1.7. Chức năng xem lịch sử nộp bài tập



Hình 2.8 Use case xem lịch sử nộp bài tập

Bảng2.12 Kịch bản chức năng xem lịch sử nộp bài tập

|  |  |
| --- | --- |
| Use case | Xem lịch sử nộp bài tập |
| Actor | Sinh viên |
| Tiền điều kiện | Sinh viên đã vào trang làm bài tập |
| Hậu điều kiện | Sinh viên xem lịch sử nộp bài tập thành công |
| Scenario | 1. Tại giao diện làm bài tập, sinh viên lập trình và click submit để nộp bài  2. Hệ thống sẽ hiển thị danh sách lịch sử các bài tập đã được nộp sắp xếp theo thời gian từ sớm nhất -> lâu nhất và các thông tin liên quan như: kết quả, tên bài tập, mã bài tập... |
| Error | Nếu sinh viên viết có lỗi cú pháp hoặc không đáp ứng yêu cầu bài tập màn hình sẽ hiển thị thông báo "Mã lập trình không hợp lệ”. Vui lòng kiểm tra và thử lại |

2.3.1.8. Danh sách sinh viên theo lớp học

A diagram of a diagram

Description automatically generated

Hình 2.9 Use case xem danh sách sinh viên cùng lớp

Bảng 2.13 Kịch bản chức năng xem danh sách sinh viên của lớp học

|  |  |
| --- | --- |
| Use case | Xem danh sách sinh viên của lớp học |
| Actor | Sinh viên |
| Tiền điều kiện | Sinh viên đã truy cập vào trang thông tin chi tiết của lớp học |
| Hậu điều kiện | Sinh viên xem được danh sách sinh viên của lớp học |
| Scenario | 1.Từ màn hình chi tiết của lớp học, sinh viên chọn chức năng xem danh sách sinh viên  2. Bảng bên dưới thông tin của lớp học hiện danh sách sinh viên gồm các thông tin: msv, tên, ngày sinh, email, sdt... |
| Error | 2. |

### 2.3.2 Kịch bản use case các chức năng của giáo viên

2.3.2.1. Chức năng xem danh sách các môn giảng dạy

A diagram of a diagram

Description automatically generated

Hình 2.10 Use case xem danh sách các môn đang giảng dạy

Bảng 2.14 Kịch bản chức năng xem danh sách các môn giảng dạy

|  |  |
| --- | --- |
| Use case | Xem danh sách các môn giảng dạy |
| Actor | Giáo viên |
| Tiền điều kiện | Giáo viên đã đăng nhập thành công vào hệ thống, tại trang chủ giáo viên chọn “Môn học” |
| Hậu điều kiện | Giáo viên xem được danh sách các môn học đang giảng dạy |
| Scenario | 1. Từ màn hình trang chủ của giáo viên, giáo viên chọn “Môn học” 2. Màn hình sẽ hiển thị danh sách các môn đang giảng dạy bao gồm các thông tin như: khoa, tên môn, số lượng lớp và action xem chi tiết từng môn |
| Error |  |

2.3.2.2 Xem chi tiết 1 môn học

A diagram of a diagram

Description automatically generated

Hình 2.11 Use case xem chi tiết một môn học đang giảng dạy

Bảng 2.15 Kịch bản chức năng xem chi tiết một môn học

|  |  |
| --- | --- |
| Use case | Xem thông tin chi tiết của môn học |
| Actor | Giáo viên |
| Tiền điều kiện | Giáo viên đã truy cập vào trang danh sách môn học |
| Hậu điều kiện | Sinh viên xem được chi tiết của môn học |
| Scenario | 1. Giáo viên truy cập vào danh sách môn học, chọn xem chi tiết 1 môn học bất kỳ  2. Màn hình sẽ hiển thị chi tiết môn học đó bao gồm các thông tin như tên môn, số tín chỉ, khoa, mô tả, bên dưới sẽ có 2 nút : các lớp học và danh sách bài tập |
|  |  |

2.3.2.2. Chức năng xem danh sách lớp của môn học

A diagram of a diagram

Description automatically generated

Hình 2.12 Use case xem danh sách lớp của môn học

Bảng 2.16 Kịch bản chức năng xem danh sách các lớp của môn học

|  |  |
| --- | --- |
| Use case | Xem danh sách lớp của môn học |
| Actor | Giáo viên |
| Tiền điều kiện | Giáo viên đã vào trang chi tiết môn học |
| Hậu điều kiện | Giáo viên xem được danh sách lớp của môn học |
| Scenario | 1. Tại màn hình chi tiết môn học , giáo viên click chọn “Các lớp học” 2. Màn hình sẽ hiển thị danh sách các lớp học, gồm các thông tin như tên lớp, tên môn học, tên giáo viên, sĩ số, ngày bắt đầu, ngày kết thúc... |
| Error |  |

2.3.2.3. Chức năng xem danh sách bài tập của môn học

A diagram of a diagram

Description automatically generated

Hình 2.13 Usecase xem danh sách bài tập của môn học

Bảng 2.17 Kịch bản chức năng xem danh sách bài tập của môn học

|  |  |
| --- | --- |
| Use case | Xem danh sách bài tập của môn học |
| Actor | Giáo viên |
| Tiền điều kiện | Giáo viên đã vào trang danh sách các môn học đang giảng dạy |
| Hậu điều kiện | Giáo viên xem được danh sách bài tập của môn học |
| Scenario | 1. Tại màn hình danh sách chi tiết môn học, giáo viên click chọn xem danh sách bài tập 2. Màn hình sẽ hiển thị danh sách bài tập và nút thêm bài tập ở bên trên. 3. Danh sách bài tập sẽ hiển thị các thông tin như tên bài tập, chủ đề, ngày thêm... |
| Error |  |

2.3.2.4. Chức năng thêm bài tập của lớp học

A diagram of a diagram

Description automatically generated

Hình 2.14 Usecase thêm bài tập của lớp học

Bảng 2.18 Kịch bản chức năng thêm bài tập

|  |  |
| --- | --- |
| Use case | Thêm bài tập của môn học |
| Actor | Giáo viên |
| Tiền điều kiện | Giáo viên đã vào trang danh sách bài tập của môn học |
| Hậu điều kiện | Giáo viên thêm được bài tập của môn học |
| Scenario | 1. Tại màn hình danh sách các bài tập môn học, giáo viên click chọn thêm bài tập cho môn học 2. Màn hình sẽ hiển thị form để nhập các thông tin về bài tập của môn học: tên, mô tả, yêu cầu, giới hạn thời gian và nút Tạo   3. Giáo viên điền form và click “Tạo”, bài tập sẽ được thêm vào danh sách bài tập |
| Error |  |

2.3.2.5. Chức năng thêm lớp học mới

Bảng 2.19 Kịch bản chức năng thêm lớp học mới

|  |  |
| --- | --- |
| Use case | Thêm bài tập của môn học |
| Actor | Giáo viên |
| Tiền điều kiện | Giáo viên đã vào trang chủ chính |
| Hậu điều kiện | Giáo viên thêm được lớp học mới |
| Scenario | 1. Tại màn hình trang chủ chính, giáo viên click chọn Lớp học bên trái màn hình 2. Màn hình sẽ hiển thị danh sách các lớp học và nút tạo lớp học mới   3. Giáo viên click chọn “Tạo lớp học mới”, màn hình sẽ hiển thị form tạo lớp với các thông tin như tên, ngày bắt đầu, kết thúc, chọn giáo viên và chọn môn học, nút import file excel và nút tạo lớp  4. Giáo viên điền form và click tạo lớp |
| Error |  |

### 2.3.3 Kịch bản use case các chức năng của admin

2.3.3.1. Chức năng xem danh sách tất cả các môn học

A diagram of a diagram

Description automatically generated

Hình 2.15 Use case xem danh sách tất cả các môn học

Bảng 2.20 Kịch bản chức năng xem danh sách tất cả các môn học

|  |  |
| --- | --- |
| Use case | Xem danh sách tất cả các môn học |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã đăng nhập thành công vào hệ thống, tại trang chủ admin chọn “Môn học” |
| Hậu điều kiện | Quản trị viên xem được danh sách các môn học |
| Scenario | 1. Từ màn hình trang chủ của giáo viên, giáo viên chọn “Môn học” 2. Màn hình sẽ hiển thị danh sách các môn đang giảng dạy bao gồm các thông tin như: khoa, tên môn, số lượng lớp và action xem chi tiết từng môn |
| Error |  |

2.3.3.2 Xem chi tiết 1 môn học

A diagram of a diagram

Description automatically generated

Hình 2.16 Use case xem chi tiết 1 môn học

Bảng 2.21 Kịch bản chức năng xem chi tiết một môn học

|  |  |
| --- | --- |
| Use case | Xem thông tin chi tiết của môn học |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã truy cập vào trang danh sách môn học |
| Hậu điều kiện | Quản trị viên xem được chi tiết của môn học |
| Scenario | 1. Quản trị viên truy cập vào danh sách môn học, chọn xem chi tiết 1 môn học bất kỳ  2. Màn hình sẽ hiển thị chi tiết môn học đó bao gồm các thông tin như tên môn, số tín chỉ, khoa, mô tả, bên dưới sẽ có 2 nút : các lớp học và danh sách bài tập |
|  |  |

2.3.3.2. Chức năng xem danh sách lớp của môn học

A diagram of a diagram

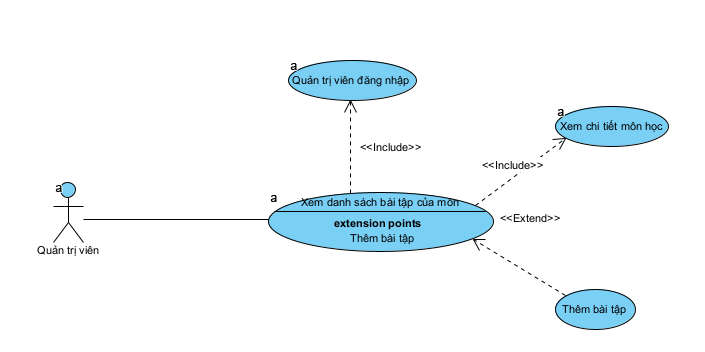
Description automatically generated

Hình 2.17 Use case xem danh sách lớp của môn học

Bảng 2.22 Kịch bản chức năng xem danh sách lớp của môn học

|  |  |
| --- | --- |
| Use case | Xem danh sách lớp của môn học |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã vào trang chi tiết môn học |
| Hậu điều kiện | Quản trị viên xem được danh sách lớp của môn học |
| Scenario | 1. Tại màn hình chi tiết môn học, sẽ có 3 nút để admin lựa chọn, admin click “Các lớp học”  2. Màn hình sẽ hiển thị danh sách các lớp học, gồm các thông tin như tên lớp, tên môn học, tên giáo viên, sĩ số, ngày bắt đầu, ngày kết thúc... |
| Error |  |

2.3.3.3. Chức năng xem danh sách bài tập của môn học



Hình 2.18 Use case xem danh sách bài tập của môn học

Bảng 2.23 Kịch bản chức năng xem danh sách bài tập của môn học

|  |  |
| --- | --- |
| Use case | Xem danh sách bài tập của môn học |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã vào trang chi tiết môn học |
| Hậu điều kiện | Quản trị viên xem được danh sách bài tập của môn học |
| Scenario | 1. Tại màn hình chi tiết môn học, quản trị viên click chọn xem danh sách bài tập 2. Màn hình sẽ hiển thị danh sách bài tập và nút thêm bài tập ở bên trên.Danh sách bài tập sẽ hiển thị các thông tin như tên bài tập, chủ đề, ngày thêm... |
| Error |  |

2.3.3.4. Chức năng thêm bài tập của môn học

A diagram of a diagram

Description automatically generated

Hình 2.19 Use case thêm bài tập cho môn học

Bảng 2.24 Kịch bản thêm bài tập của môn học

|  |  |
| --- | --- |
| Use case | Thêm bài tập của môn học |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã vào trang danh sách bài tập của môn học |
| Hậu điều kiện | Quản trị viên thêm được bài tập của môn học |
| Scenario | 1. Tại màn hình danh sách các bài tập môn học, quản trị viên click chọn thêm bài tập cho môn học 2. Màn hình sẽ hiển thị form để nhập các thông tin về bài tập của môn học: tên, mô tả, yêu cầu, giới hạn thời gian và nút Tạo 3. Quản trị viên điền form và click “Tạo”, bài tập sẽ được thêm vào danh sách bài tập |
| Error |  |

2.3.3.5. Chức năng import file excel sinh viên

A diagram of a diagram

Description automatically generated

Hình 2.20 Use case import file excel sinh viên

Bảng 2.25 Kịch bản chức năng import file excel sinh viên

|  |  |
| --- | --- |
| Use case | Import file excel danh sách sinh viên |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã vào trang chủ chính |
| Hậu điều kiện | Quản trị viên import được file excel sinh viên |
| Scenario | 1. Tại giao diện trang chủ chính, quản trị viên click chọn Sinh viên ở bên trái màn hình 2. Màn hình sẽ hiển thị danh sách sinh viên gồm các thông tin như: tên, msv, sdt... và nút Import Students (excel)   3. Quản trị viên click chọn Import Students (excel) và chọn 1 file excel từ máy tính  4. File được tải lên và thêm vào danh sách sinh viên |
| Error |  |

2.3.3.5. Chức năng thêm lớp học mới cho môn học

A diagram of a diagram

Description automatically generated

Hình 2.21 Use case thêm lớp học mới cho môn học

Bảng 2.26 Kịch bản chức năng thêm lớp học mới cho môn học

|  |  |
| --- | --- |
| Use case | Thêm lớp học mới cho môn học |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã vào trang chủ chính |
| Hậu điều kiện | Quản trị viên thêm được lớp học mới |
| Scenario | 1. Tại màn hình trang chủ chính, quản trị viên click chọn Lớp học bên trái màn hình 2. Màn hình sẽ hiển thị danh sách các lớp học và nút tạo lớp học mới 3. Quản trị viên click chọn “Tạo lớp học mới”, màn hình sẽ hiển thị form tạo lớp với các thông tin như tên, ngày bắt đầu, kết thúc, chọn quản trị viên và chọn môn học, nút import file excel và nút tạo lớp 4. Quản trị viên điền form và click tạo lớp |
| Error |  |

2.3.3.6 Xem danh sách giáo viên

A diagram of a diagram

Description automatically generated

Hình 2.22 Use case xem danh sách giáo viên

Bảng 2.27 Kịch bản chức năng xem danh sách giáo viên

|  |  |
| --- | --- |
| Use case | Xem danh sách giáo viên |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã vào trang chủ chính |
| Hậu điều kiện | Quản trị viên xem được danh sách giáo viên |
| Scenario | 1. Tại màn hình trang chủ chính, quản trị viên click “Giáo viên” 2. Màn hình sẽ hiển thị danh sách các giáo viên trên hệ thống gồm các thông tin như tên, email , sdt… và action xem chi tiết |
| Error |  |

2.3.3.7 Tạo một giáo viên

A diagram of a diagram

Description automatically generated

Hình 2.23 Use case thêm mới một giáo viên

Bảng 2.28 Kịch bản Tạo một giáo viên

|  |  |
| --- | --- |
| Use case | Thêm một giáo viên |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã vào trang chủ chính |
| Hậu điều kiện | Quản trị viên thêm được giáo viên |
| Scenario | 1. Tại màn hình trang chủ chính, quản trị viên click “Giáo viên” 2. Màn hình sẽ hiển thị danh sách các giáo viên và nút tạo giáo viên 3. Quản trị viên click chọn “Tạo giáo viên”, màn hình sẽ hiển thị form tạo lớp với các thông tin như tên, email, sdt, khoa và nút Tạo 4. Quản trị viên điền form và click tạo giáo viên |
| Error |  |

2.3.3.8 Xem chi tiết một giáo viên

A diagram of a diagram

Description automatically generated

Hình 2.24 Use case xem chi tiết một giáo viên

Bảng 2.29 Kịch bản xem chi tiết giáo viên

|  |  |
| --- | --- |
| Use case | Xem thông tin chi tiết của giáo viên |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã truy cập vào trang danh sách giáo viên |
| Hậu điều kiện | Quản trị viên xem được chi tiết của giáo viên |
| Scenario | 1. Quản trị viên truy cập vào danh sách giáo viên, chọn xem chi tiết 1 giáo viên bất kì  2. Màn hình sẽ hiển thị chi tiết giáo viên đó bao gồm các thông tin như tên, email, sdt… |
| Error |  |

2.3.3.9 Xem danh sách các lớp giáo viên đang giảng dạy

Bảng 2.30 Kịch bản xem danh sách các lớp đang giảng dạy

|  |  |
| --- | --- |
| Use case | Xem danh sách các lớp đang giảng dạy của giáo viên |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã truy cập vào trang danh sách giáo viên |
| Hậu điều kiện | Quản trị viên xem danh sách các lớp đang giảng dạy |
| Scenario | 1. Quản trị viên truy cập vào danh sách giáo viên, chọn xem chi tiết 1 giáo viên bất kì  2. Màn hình sẽ hiển thị 2 tab, 1 tab gồm các thông tin chi tiết giáo viên đó bao gồm các thông tin như tên, email, sdt… và 1tab hiển thị danh sách các lớp giáo viên giảng dạy  3. Quản trị viên click chọn Danh sách các lớp đang giảng dạy  4. Màn hình sẽ hiển thị các lớp giáo viên đang giảng dạy |
| Error |  |

2.3.3.10 Thêm tài liệu cho môn học

Bảng 2.31 Kịch bản thêm tài liệu môn học

|  |  |
| --- | --- |
| Use case | Thêm tài liệu cho môn học |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã truy cập vào trang danh sách môn học |
| Hậu điều kiện | Quản trị thêm tài liệu cho môn học thành công |
| Scenario | 1. Quản trị viên truy cập vào danh môn học, chọn xem chi tiết 1 môn học bất kì  2. Màn hình sẽ hiển thị 3 tab, 1 tab gồm các thông tin về các lớp học, 1tab hiển thị danh sách các bài tập và 1tab hiển thị danh sách tài liệu của môn học  3. Quản trị viên click chọn Danh sách tài liệu  4. Màn hình sẽ hiển thị các danh sách tài liệu của lớp học và nút thêm tài liệu  5. Quản trị viên click Thêm tài liệu  6. Màn hình hiển thị form nhập tên tài liệu và nút tải tài liệu lên. Quản trị viên nhập tên và chọn tài liệu từ máy rồi click Tạo  7. Hệ thống hiển thị thông báo tạo thành công |
| Error |  |

2.3.3.11 Tải tài liệu môn học

Bảng 2.32 Kịch bản tải tài liệu

|  |  |
| --- | --- |
| Use case | Tải tài liệu cho môn học |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã truy cập vào trang chi tiết 1 môn học |
| Hậu điều kiện | Quản trị tải tài liệu thành công |
| Scenario | 1. Quản trị viên truy cập vào chi tiết 1 môn học, màn hình sẽ hiển thị 3 tab, 1 tab gồm các thông tin về các lớp học, 1tab hiển thị danh sách các bài tập và 1tab hiển thị danh sách tài liệu của môn học  3. Quản trị viên click chọn Danh sách tài liệu  4. Màn hình sẽ hiển thị các danh sách tài liệu của lớp học và nút thêm tài liệu  5. Quản trị viên click button tải tài liệu  6. Tài liệu được tải xuống |
| Error |  |

## 2.4 Trích lớp thực thể

### 2.4.1 Trích các danh từ liên quan được mô tả trong hệ thống

* Các danh từ liên quan đến người: thành viên hệ thống, giảng viên, quản trị viên, sinh viên,
* Các danh từ liên quan đến vật: trường đại học
* Các danh từ thông tin: khoa, khóa học, môn học, lớp học, tài liệu, bài tập, bộ kiểm thử, điểm số, môi trường lập trình, ngôn ngữ lập trình, lịch sử làm bài, chương

### 2.4.2 Đánh giá và lựa chọn các danh từ làm lớp thực thể hoặc thuộc tính

* Các danh từ liên quan đến người
  + Thành viên hệ thống -> lớp User(trừu tượng): tên, tên đăng nhập, mật khẩu, ngày sinh, địa chỉ, điện thoại, vai trò, ghi chú
  + Sinh viên -> lớp Student: kế thừa lớp User, thêm thuộc tính: mã sinh viên
  + Giảng viên -> lớp Teacher: kế thừa User
  + Quản trị viên -> lớp Admin: kế thừa User
* Các danh từ chỉ vật
  + Trường đại học -> lớp University: tên, địa chỉ, mô tả
* Các danh từ thông tin:
  + Khoa -> lớp Department: tên, mô tả
  + Môn học -> lớp Subject: tên, mô tả
  + Lớp học -> lớp Class: tên, mô tả, thời gian, trạng thái
  + Tài liệu -> lớp StudyMaterial: tên, thời gian tạo, nội dung
  + Bài tập -> lớp Exercise: tên, yêu cầu, mức độ, thời gian tạo, timeLimit, memoryLimit
  + Bộ kiểm thử -> lớp TestCase: tên bài tập, đầu vào, đầu ra
  + Môi trường lập trình -> ngoài phạm vi
  + Ngôn ngữ lập trình -> lớp ProgrammingLanguage: tên, mô tả
  + Lịch sử làm bài -> lớp SubmissionHistory: thời gian, kết quả, mã nguồn
  + Chương + chủ đề -> lớp Topic: tên, mô tả

2.4.3. Xác định quan hệ các thực thể

* Một trường có nhiều khoa -> quan hệ giữa University và Department là 1-n
* Một khoa có nhiều môn học, một môn học có thể thuộc nhiều khoa quan hệ giữa Department và Subject là n-n -> đề xuất lớp DepartmentSubject để biểu diễn quan hệ giữa 2 lớp
* Một bộ môn có thể có nhiều lớp, một lớp chỉ dạy 1 môn -> quan hệ giữa Subject và Class là 1 - n
* Một giảng viên có thể dạy nhiều lớp, một lớp chỉ được dạy bởi 1 giảng viên -> quan hệ giữa Teacher và Class là 1-n
* Một lớp có thể có nhiều sinh viên, một sinh viên có thể học nhiều lớp -> quan hệ giữa Class và Student là n-n -> đề xuất lớp StudentClass để kết nối giữa lớp học và sinh viên
* Một môn có thể có nhiều chủ đề, một chủ đề thuộc một môn -> Quan hệ giữa 2 lớp Subject và Topic là 1-n
* Một môn học có thể có nhiều bài tập, một bài tập có thể thuộc nhiều môn học -> quan hệ giữa Subject và Exercise là n-n -> Đề xuất tạo một lớp trung gian SubjectExercise để thể hiện mối quan hệ giữa Subject và Exercise
* Một chủ đề của môn học có thể chứa nhiều bài tập của môn học, một bài tập của môn học chỉ thuộc một chủ đề của môn học -> mối quan hệ giữa Topic và SubjectExercise là 1-n
* Một lớp học có thể sử dụng nhiều ngôn ngữ lập trình, một ngôn ngữ lập trình có thể được dùng trong nhiều môn học -> mối quan hệ giữa Subject và ProgrammingLanguage là n-n -> đề xuất lớp SubjectProgrammingLanguage  để biểu diễn quan hệ giữa 2 lớp
* Một lớp có có thể mở nhiều chủ đề bài tập của môn, một chủ đề bài tập có thể mở trong nhiều lớp -> mối quan hệ giữa Class và Topic là n-n -> đề xuất lớp ClassTopicOpen để biểu diễn các chủ đề bài tập được mở trong một lớp
* Một chủ đề được mở trong lớp có thể chứa nhiều bài tập của môn, một bài tập của môn có thể nằm trong chủ đề được mở trong nhiều lớp khác nhau -> mối quan hệ giữa ClassTopicOpen và SubjectExercise là một nhiều -> đề xuất lớp ClassExercise để biểu diễn các bài tập được mở theo chủ đề trong một lớp
* Một sinh viên có thể làm nhiều bài tập trong lớp, một bài tập trong lớp có thể được làm bởi nhiều sinh viên -> mối quan hệ giữa Student và ClassExercise là n-n -> đề xuất lớp StudentSubmission để biểu diễn các lần nộp bài của sinh viên

2.4.5. Biểu đồ lớp thực thể

A diagram of a company

Description automatically generated

## 2.5 Kết luận Chương II

Chương II đã cung cấp một cái nhìn chi tiết về các yêu cầu của hệ thống thông qua việc phân tích và đặc tả các chức năng cần thiết. Các biểu đồ Use Case và tuần tự giúp minh họa rõ ràng các tương tác và luồng hoạt động trong hệ thống, tạo nền tảng vững chắc cho việc thiết kế và phát triển hệ thống trong các chương tiếp theo. Việc phân tích kỹ lưỡng các yêu cầu đảm bảo rằng hệ thống sẽ đáp ứng được nhu cầu của người dùng và hoạt động hiệu quả trong môi trường thực tế.

# CHƯƠNG III. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

Chương này trình bày chi tiết về quá trình thiết kế và xây dựng hệ thống "Hệ thống quản trị học tập lập trình online ". Đầu tiên, chương tập trung vào kiến trúc tổng thể của hệ thống, làm rõ cách thức các thành phần tương tác và phối hợp với nhau để đảm bảo tính ổn định, bảo mật và khả năng mở rộng. Tiếp theo, chương giới thiệu cấu trúc cơ sở dữ liệu, bao gồm việc thiết kế các bảng, mối quan hệ và ràng buộc cần thiết để đáp ứng yêu cầu lưu trữ và truy xuất dữ liệu hiệu quả. Cuối cùng, chương liệt kê và phân tích các công nghệ được lựa chọn để triển khai dự án, từ đó giải thích lý do chọn lựa và lợi ích mà các công nghệ này mang lại.

Mục tiêu của chương là cung cấp cái nhìn rõ ràng và toàn diện về cách hệ thống được thiết kế và xây dựng, từ ý tưởng ban đầu đến các quyết định kỹ thuật thực tế, ...

## 3.1 Kiến trúc hệ thống

Hệ thống "Xây dựng hệ thống quản trị học tập lập trình online " được chia thành hai phần chính: Hệ thống quản lý lớp học và Hệ thống biên dịch thực thi code, mỗi phần đảm nhận vai trò riêng biệt nhằm đảm bảo hiệu quả, tính bảo mật và khả năng mở rộng.

A diagram of a data flow

Description automatically generated

Hình 3.1 Kiến trúc tổng thể hệ thống

### 3.1.1 Hệ thống quản lý lớp học

Hệ thống quản lý lớp học là phần chính của ứng dụng, bao gồm các chức năng quản lý thông tin lớp học, lịch học và tiến độ học tập của học viên. Hệ thống được xây dựng dựa trên kiến trúc ba lớp:

* Frontend (Lớp giao diện người dùng):
  + Vai trò: Cung cấp giao diện trực quan và thân thiện để người dùng tương tác với hệ thống.
  + Chức năng:
    - Hiển thị danh sách các khóa học, lớp học, bài tập, tài liệu, thông tin cá nhân, và các chức năng khác của hệ thống.
    - Cho phép người dùng thực hiện các thao tác như đăng nhập, đăng ký, xem thông tin, nộp bài tập, ...
    - Xử lý các sự kiện từ người dùng (click, nhập liệu, ...), và gửi các yêu cầu đến backend.
  + Công nghệ: Sử dụng ReactJS để xây dựng giao diện người dùng linh hoạt và có tính tái sử dụng cao.
  + Giao tiếp: Giao tiếp với Backend thông qua các RESTful API.
* Backend (Lớp xử lý nghiệp vụ):
  + Vai trò: Xử lý logic nghiệp vụ của hệ thống, điều phối các hoạt động và tương tác với cơ sở dữ liệu.
  + Chức năng:
    - Xác thực người dùng, phân quyền truy cập.
    - Quản lý thông tin khóa học, lớp học, bài tập, tài liệu.
    - Xử lý các yêu cầu từ frontend (thêm, sửa, xóa, truy vấn dữ liệu, ...).
    - Thực thi các nghiệp vụ liên quan đến quản lý lớp học.
    - Tương tác với cơ sở dữ liệu để lưu trữ và truy xuất dữ liệu.
    - Cung cấp các API cho Frontend để giao tiếp.
  + Công nghệ: Sử dụng C# và .NET Core (hoặc .NET) để xây dựng backend.
  + Giao tiếp: Cung cấp RESTful API cho Frontend. Truy vấn dữ liệu từ Database.
* Database (Lớp cơ sở dữ liệu):
  + Vai trò: Lưu trữ và quản lý tất cả dữ liệu của hệ thống một cách an toàn và hiệu quả.
  + Chức năng:
    - Lưu trữ thông tin người dùng (sinh viên, giảng viên, quản trị viên), thông tin khóa học, lớp học, bài tập, tài liệu, kết quả nộp bài, ...
    - Cung cấp các chức năng truy vấn, cập nhật, xóa dữ liệu.
    - Công nghệ: Sử dụng SQL Server để quản lý cơ sở dữ liệu quan hệ.

### Hệ thống biên dịch thực thi code

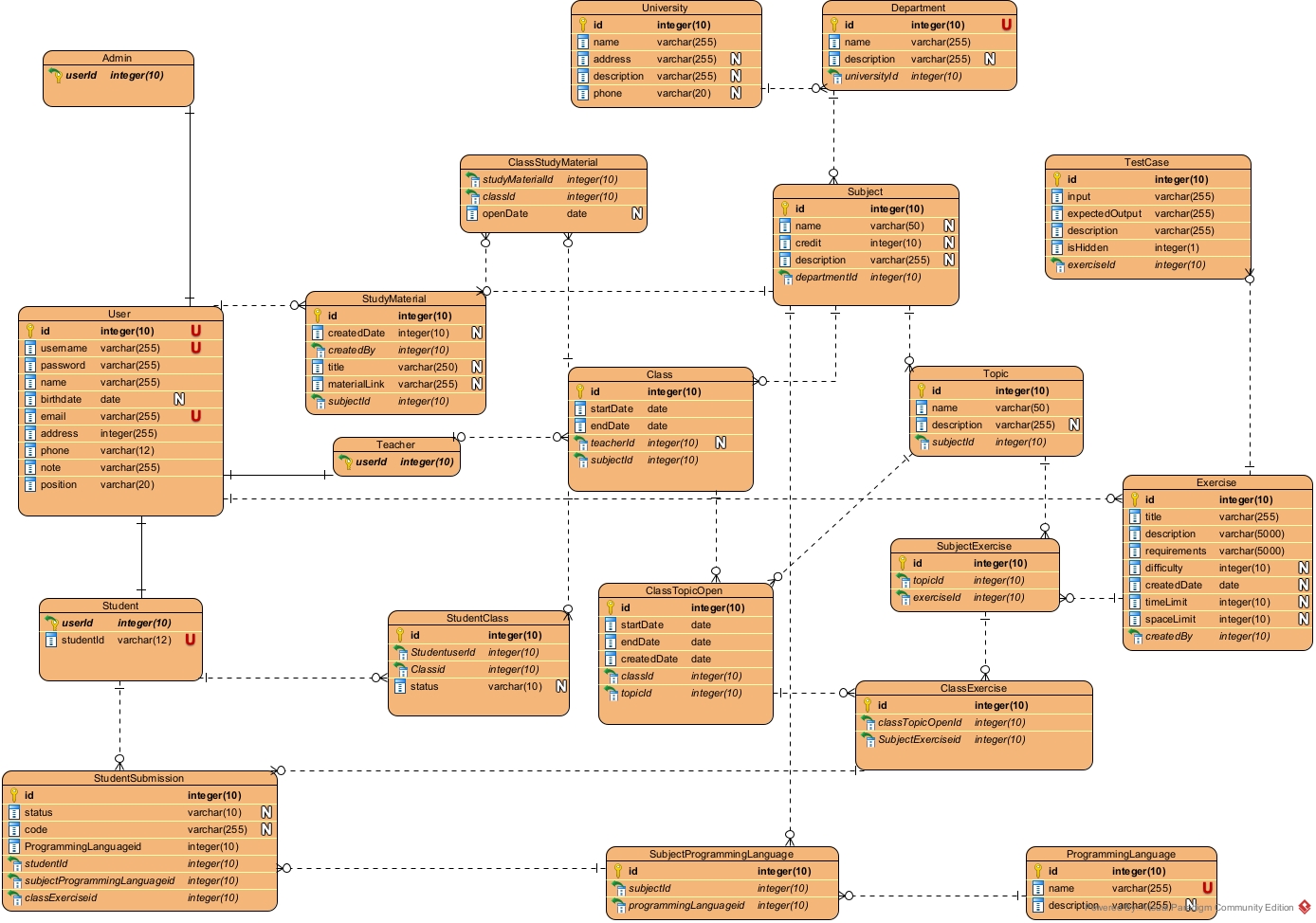
Hệ thống biên dịch thực thi code là một module độc lập, được thiết kế để cung cấp dịch vụ biên dịch và thực thi mã nguồn lập trình trực tuyến cho sinh viên, đóng vai trò như một dịch vụ hỗ trợ cho hệ thống quản lý lớp học. Hệ thống này được tách biệt để đảm bảo tính linh hoạt, bảo mật và khả năng tái sử dụng.

* Mục tiêu:
* Khả năng tích hợp cao: Hệ thống biên dịch có thể được sử dụng bởi các ứng dụng khác nhau, không chỉ giới hạn ở hệ thống quản lý lớp học.
* Tăng cường bảo mật: Bằng cách tách biệt, hệ thống biên dịch không có quyền truy cập vào cơ sở dữ liệu hoặc thông tin nhạy cảm của hệ thống quản lý lớp học. Điều này giúp hạn chế nguy cơ xâm nhập và bảo vệ tài nguyên của hệ thống chính.
* Độc lập và tối ưu hóa: Hệ thống biên dịch được thiết kế để hoạt động độc lập, có khả năng mở rộng và nâng cấp mà không ảnh hưởng đến các chức năng khác của hệ thống.
* Hệ thống thực thi code chỉ cung cấp dịch vụ biên dịch và thực thi mã, qua đó đóng vai trò như một dịch vụ hỗ trợ chuyên biệt cho hệ thống chính.
* Kiến trúc:
  + Hệ thống hoạt động theo mô hình dịch vụ, tương tác với backend của hệ thống quản lý lớp học thông qua RESTful API.
  + Không có client frontend trực tiếp.
  + Hệ thống quản lý lớp học (Backend):
    - Đóng vai trò như “client”, gửi yêu cầu biên dịch/thực thi mã nguồn đến hệ thống biên dịch thông qua API.
    - Gửi các thông tin bao gồm: sourceCode, language và testCases trong request body JSON.
    - Nhận kết quả từ hệ thống biên dịch và trả về cho frontend của hệ thống quản lý lớp học.
  + Hệ thống biên dịch thực thi code (Backend):
    - Đóng vai trò như “server” cung cấp dịch vụ biên dịch/thực thi mã nguồn.
    - Nhận yêu cầu từ hệ thống quản lý lớp học thông qua RESTfulAPI (method POST).
    - Xác định ngôn ngữ lập trình dựa trên trường language trong request.
    - Tạo môi trường thực thi an toàn bằng cách:
      * Tạo một thư mục tạm thời riêng biệt để chứa mã nguồn và các file cần thiết cho mỗi yêu cầu.
      * Tạo một process mới để gọi trình biên dịch/thực thi tương ứng (javac, gcc, python, dotnet).
      * Sử dụng ProcessBuilder để tạo và quản lý process.
      * Chạy container với user app không phải root.
      * Mount root filesystem ở chế độ chỉ đọc.
    - Giới hạn tài nguyên sử dụng cho container (CPU, RAM, disk I/O).
    - Thực hiện biên dịch/thực thi code trong process.
    - Thực hiện các test case, so sánh output với kết quả mong đợi.
    - Ghi lại output và error streams.
    - Trả về kết quả dưới dạng JSON qua API cho hệ thống quản lý lớp học.
    - Xử lý timeout bằng cách kiểm tra thời gian thực thi và destroy process nếu quá thời gian quy định.
* Luồng Hoạt Động:
  + Sinh viên nộp bài: Sinh viên viết mã nguồn tại Frontend của hệ thống quản lý lớp học.
  + Frontend gửi code cho Backend: Frontend gửi mã nguồn cùng với thông tin bài tập và ngôn ngữ sử dụng đến Backend của hệ thống quản lý lớp học.
  + Backend gửi yêu cầu cho hệ thống biên dịch: Backend của hệ thống quản lý lớp học gửi yêu cầu biên dịch/thực thi đến hệ thống biên dịch thông qua POST request đến API /api/execute.
  + Hệ thống biên dịch xử lý:
    - Hệ thống biên dịch nhận yêu cầu qua API, xác thực, ủy quyền (nếu có).
    - Tạo môi trường chạy code an toàn (một thư mục và process riêng trong Docker container).
    - Biên dịch/thực thi mã nguồn.
    - Chạy các test cases.
    - Lưu lại output và error stream.
    - Thực hiện các biện pháp bảo mật như đã nêu ở phần trên.
  + Hệ thống biên dịch trả kết quả: Hệ thống biên dịch trả về kết quả dưới dạng JSON thông qua API.
  + Backend trả kết quả cho Frontend: Backend của hệ thống quản lý lớp học xử lý và trả kết quả cho Frontend để hiển thị cho sinh viên.
* Các Biện Pháp Bảo Mật:
  + User không phải root: Container chạy với user không phải root (app):
  + Không lưu thông tin nhạy cảm trong image: Không lưu trực tiếp thông tin nhạy cảm vào image, sử dụng các secret manager hoặc inject biến môi trường để truyền các thông tin này.
  + Tạo môi trường thực thi an toàn: Mỗi yêu cầu thực thi code sẽ chạy trong một thư mục và process riêng.
  + Giới hạn thời gian thực thi: Sử dụng timeout để ngăn code chạy vô hạn.

### 3.1.3 Ưu điểm của kiến trúc

* Tính linh hoạt: Các phần của hệ thống có thể hoạt động độc lập, dễ dàng bảo trì và nâng cấp.
* Tăng hiệu quả phát triển: Việc tách riêng hệ thống biên dịch giúp nhóm phát triển tập trung tối ưu hóa từng phần, không gây ảnh hưởng chéo.
* An toàn dữ liệu: Giảm thiểu nguy cơ truy cập trái phép vào cơ sở dữ liệu của hệ thống chính thông qua các lỗ hổng từ hệ thống thực thi code.

## 3.2 Thiết kế cơ sở dữ liệu



Hình 3.2 Thiết kế cơ sở dữ liệu của hệ thống

**Dưới đây là mô tả cấu trúc cụ thể từng bảng theo hệ quản trị CSDL Microsoft SQL Server:**

- Bảng User: Lưu trữ thông tin về tất cả người dùng trong hệ thống, bao gồm cả người dùng thường và quản trị viên

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | ID tài khoản người dùng. |
| username | NVARCHAR(255) | Tên đăng nhập của người dùng. Phải là duy nhất trong hệ thống. |
| password | NVARCHAR(255) | Mật khẩu của người dùng, được mã hóa để đảm bảo bảo mật. |
| name | NVARCHAR(255) | Tên của người dùng. |
| birthdate | DATETIME | Ngày sinh của người dùng |
| email | NVARCHAR(255) | Email của người dùng |
| address | NVARCHAR(255) | Địa chỉ của người dùng |
| phone | NVARCHAR(12) | Sdt của người dùng |
| note | NVARCHAR(255) | Ghi chú của người dùng (nếu có) |
| position | NVARCHAR(20) | Phân quyền của người dùng, có thể là admin, user.... |

Bảng 3.1 Cấu trúc bảng User

- Bảng Admin: Lưu trữ thông tin về quản trị viên như tên, email, địa chỉ, ...

Bảng 2.2 Cấu trúc bảng Admin

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| userId | UNIQUEIDENTIFIER | Khóa ngoại đến id của người dùng |

- Bảng Student: Lưu trữ thông tin về sinh viên như tên, email, địa chỉ, ...

Bảng 3.3 Cấu trúc bảng Student

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| userId | UNIQUEIDENTIFIER | Khóa ngoại đến id của người dùng |
| studentId | NVARCHAR(12) | Id của học sinh |

- Bảng Teacher: Lưu trữ thông tin về giáo viên như tên, email, địa chỉ, ...

Bảng 3.4 Cấu trúc bảng Teacher

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| userId | UNIQUEIDENTIFIER | Khóa ngoại đến id của người dùng |

- Bảng StudentSubmission: Lưu trữ thông tin về bài tập đã nộp của sinh viên

Bảng 3.5 Cấu trúc bảng StudentSubmission

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | ID bài nộp của học sinh |
| status | NVARCHAR(255) | Trạng thái của bài tập đã nộp |
| code | NVARCHAR(255) | Mô tả phần code của học sinh |
| studentId | UNIQUEIDENTIFIER | Khóa ngoại đến id của học sinh |
| subjectProgrammingLanguagedid | UNIQUEIDENTIFIER | Khóa ngoại đến ngôn ngữ lập trình |
| classExerciseid | UNIQUEIDENTIFIER | Khóa ngoại đến bài tập của lớp học |

- Bảng Class: Lưu trữ thông tin về lớp học

Bảng 3.6 Cấu trúc bảng Class

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | ID của lớp học |
| startDate | DATETIME | Ngày bắt đầu của lớp học |
| endDate | DATETIME | Ngày kết thúc của lớp học |
| teacherId | UNIQUEIDENTIFIER | Khóa ngoại đến id của giáo viên |
| subjectId | UNIQUEIDENTIFIER | Khóa ngoại đến id của môn học |

- Bảng Subject: Lưu trữ thông tin về môn học

Bảng 3.7 Cấu trúc bảng Subject

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | ID của môn học |
| name | NVARCHAR(50) | Tên của môn học |
| credit | INT(10) | Số tín chỉ của môn học |
| description | NVARCHAR(255) | Mô tả những thông tin liên quan tới môn học |
| departmentId | UNIQUEIDENTIFIER | Khóa ngoại đến khoa của môn học |

- Bảng University: Lưu trữ thông tin về trường đại học

Bảng 3.8 Cấu trúc bảng University

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | ID của trường đại học |
| name | NVARCHAR(255) | Tên của đại học |
| address | NVARCHAR(255) | Địa chỉ của đại học |
| description | NVARCHAR(255) | Mô tả những thông tin liên quan tới đại học, bao gồm các thông tin về tên trường, địa chỉ... |
| phone | NVARCHAR(20) | SDT của đại học |

- Bảng ClassTopicOpen: Lưu trữ thông tin về các chủ đề đã mở của lớp học

Bảng 3.9 Cấu trúc bảng ClassTopicOpen

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | ID chủ đề đã mở của lớp học |
| startDate | DATETIME | Ngày bắt đầu của chủ đề đã mở của lớp học |
| endDate | DATETIME | Ngày bắt đầu của chủ đề đã mở của lớp học |
| createdDate | DATETIME | Ngày tạo chủ đề của lớp học |
| Classid | UNIQUEIDENTIFIER | Khóa ngoại đến id lớp học |
| topicId | UNIQUEIDENTIFIER | Khóa ngoại đến chủ đề lớp học |

- Bảng StudentClass: Lưu trữ thông tin về học sinh của lớp học

Bảng 3.10 Cấu trúc bảng StudentClass

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | ID lớp của học sinh |
| StudentuserId | UNIQUEIDENTIFIER | Khóa ngoại đến id của sinh viên |
| classId | UNIQUEIDENTIFIER | Khóa ngoại đến id của lớp học |
| Status | NVARCHAR(10) | Tình trạng của sinh viên trong lớp học |

- Bảng Department: Lưu trữ các thông tin về khoa học

Bảng 3.11 Cấu trúc bảng Department

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | ID lớp của khoa học |
| name | NVARCHAR(255) | Tên của khoa |
| description | NVARCHAR(255) | Mô tả những thông tin liên quan đến khoa |
| universityId | UNIQUEIDENTIFIER | Khóa ngoại đến id của đại học |

- Bảng Exercise: Lưu trữ các thông tin về bài tập

Bảng 3.12 Cấu trúc bảng Exercise

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | ID của bài tập |
| title | NVARCHAR(255) | Tiêu đề của bài tập |
| description | NVARCHAR(5000) | Mô tả đề bài của bài tập |
| requirements | NVARCHAR(5000) | Yêu cầu cho bài tập đó |
| difficulty | INT(10) | Độ khó của bài tập |
| createdDate | DATETIME | Thời gian được tạo của bài tập |
| timeLimit | INT(10) | Giới hạn thời gian của bài tập |
| spaceLimit | INT(10) | Giới hạn khoảng trống của bài tập |
| createdBy | INT(10) | Khóa ngoại đến người tạo bài tập |

- Bảng SubjectExercise: Lưu trữ thông tin về bài tập của môn học

Bảng 3.13 Cấu trúc bảng SubjectExercise

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | ID bài tập của môn học |
| topicId | UNIQUEIDENTIFIER | Khóa ngoại đến id của chủ đề |
| exerciseId | UNIQUEIDENTIFIER | Khóa ngoại đến id của bài tập |

- Bảng ClassExercise: Lưu trữ thông tin về bài tập của lớp học

Bảng 3.14 Cấu trúc bảng ClassExercise

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | ID bài tập của lớp học |
| classTopicOpenid | UNIQUEIDENTIFIER | Khóa ngoại đến id chủ đề đã mở của lớp học |
| subjectExercisedid | UNIQUEIDENTIFIER | Khóa ngoại đến id bài tập của môn học |

- Bảng Topic: Lưu trữ thông tin về chủ đề bài tập

Bảng 3.15 Cấu trúc bảng Topic

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | ID của chủ đề môn học |
| name | NVARCHAR(50) | Tên chủ đề |
| description | NVARCHAR(255) | Mô tả thông tin về chủ đề của môn học |
| subjectId | UNIQUEIDENTIFIER | Khóa ngoại đến id của môn học |

- Bảng TestCase: Lưu trữ thông tin về test case

Bảng 3.16 Cấu trúc bảng TestCase

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | Id của testcase |
| input | NVARCHAR(255) | Mô tả đầu vào của test case |
| expectedOutput | NVARCHAR(255) | Mô tả đầu ra mong muốn của test case |
| description | NVARCHAR(255) | Mô tả những thông tin liên quan về test case |
| isHidden | INT(1) | Trạng thái ẩn/hiện của test case với học sinh |
| exerciseId | UNIQUEIDENTIFIER | Khóa ngoại đến id của bài tập |

- Bảng SubjectProgrammingLanguage: Lưu trữ thông tin về ngôn ngữ lập trình được sử dụng trong môn học

Bảng 3.17 Cấu trúc bảng SubjectProgrammingLanguage

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | Id ngôn ngữ lập trình được phép sử dụng trong môn học |
| subjectId | UNIQUEIDENTIFIER | Khóa ngoại đến id môn học |
| programmingLanguageid | UNIQUEIDENTIFIER | Khóa ngoại đến ngôn ngữ lập trình |

- Bảng ProgrammingLanguage: Lưu trữ thông tin về ngôn ngữ lập trình

Bảng 3.18 Cấu trúc bảng ProgrammingLanguage

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| id | UNIQUEIDENTIFIER | Id ngôn ngữ lập trình |
| name | NNVARCHAR(255) | Tên của ngôn ngữ lập trình |
| description | NNVARCHAR(255) | Mô tả các thông tin liên quan về ngôn ngữ lập trình |

## 3.3 Thiết kế hoạt động

### 3.3.1. Thiết kế biểu đồ tuần tự cho modul xem danh sách lớp học cho tất cả người dùng

A screenshot of a computer

Description automatically generated

### 3.3.2 Thiết kế biểu đồ tuần tự cho modul tạo lớp học của quản trị viên

A screenshot of a computer

Description automatically generated

### 3.3.3 Thiết kế biểu đồ tuần tự cho modul xem thông tin chi tiết lớp học của tất cả người dùng

A screenshot of a computer screen

Description automatically generated

### 3.3.4. Thiết kế biểu đồ tuần tự cho module xem danh sách các môn học của quản trị viên và giáo viên

A computer screen shot of a computer program

Description automatically generated with medium confidence

## 3.4 Công nghệ sử dụng

### 3.4.1. C#

C# (hay còn gọi là C Sharp) là một ngôn ngữ lập trình được phát triển bởi Microsoft. Nó thuộc về họ ngôn ngữ C và được thiết kế đặc biệt cho việc phát triển ứng dụng trên nền tảng .NET Framework và .NET Core (nay được gọi là .NET) của Microsoft.

C# được kế thừa những ưu điểm của C/C++, bỏ đi những dư thừa không cần thiết gây khó khăn cho người sử dụng. Câu lệnh C# đơn giản, ít từ khóa. Tính năng debug rất thuận tiện. Ngôn ngữ lập trình C# được đánh giá là dễ học và kết hợp được nhiều ưu điểm của các ngôn ngữ lập trình hướng đối tượng Java và C++. Do đó ngôn ngữ C# đang được ưu chuộng nhất.

Đặc điểm nổi bật của C#: Hướng đối tượng, An toàn bộ nhớ, Tính năng hiện đại. Nhờ những điều đó, nên C# được sử dụng rộng rãi trong việc phát triển các ứng dụng Windows dành cho máy tính để bàn, game, ứng dụng web và các ứng dụng dành cho thiết bị di động.

### 3.4.2. Công nghệ .NET

a. Tìm hiểu về công nghệ .NET

.NET là một nền tảng nguồn mở, miễn phí và được Microsoft phát triển để xây dựng, cũng như có thể chạy nhiều loại ứng dụng khác nhau, bao gồm ứng dụng web, phần mềm máy tính để bàn, ứng dụng dành cho thiết bị di động, game…

Nền tảng .NET bao gồm hai thành phần chính, là:

* .NET Framework: Là một nền tảng trưởng thành đã xuất hiện trong nhiều năm, chủ yếu được sử dụng để xây dựng các ứng dụng máy tính để bàn Windows và ứng dụng web phía Server.
* .NET Core: Là một Framework nguồn mở và đa nền tảng có thể được sử dụng để xây dựng các ứng dụng cho Windows, Linux và macOS. Nó cũng được tối ưu hóa cho các ứng dụng dựa trên đám mây.

Một trong những lợi ích chính mà .NET mang lại đó là về mặt hiệu suất và khả năng mở rộng của nó. Các ứng dụng .NET có thể xử lý lưu lượng truy cập cao và các hoạt động phức tạp mà không ảnh hưởng đến tốc độ hoặc độ ổn định. Thêm vào đó, .NET còn cung cấp các tính năng bảo mật nâng cao, làm cho nó trở thành một nền tảng an toàn trong việc xây dựng các ứng dụng xử lý dữ liệu nhạy cảm.

b. Tại sao lại chọn .NET?

Dễ phát triển: Các nhà phát triển thích sử dụng .NET vì nền tảng này chứa nhiều công cụ giúp họ làm việc dễ dàng hơn. Ví dụ: nhờ sử dụng bộ Visual Studio, các nhà phát triển có thể viết mã nhanh hơn, cộng tác hiệu quả, đồng thời kiểm thử và sửa mã của họ một cách hiệu quả.  Khả năng tái sử dụng mã giữa các kiểu triển khai giúp giảm chi phí phát triển.

Ứng dụng hiệu năng cao: Các ứng dụng .NET cung cấp thời gian phản hồi nhanh hơn và đòi hỏi công suất điện toán thấp hơn. Những ứng dụng này sở hữu các biện pháp bảo mật tích hợp mạnh mẽ và thực hiện hiệu quả các tác vụ phía máy chủ như truy cập cơ sở dữ liệu.

Hỗ trợ cộng đồng: .NET là nền tảng nguồn mở, tức là bất kỳ ai cũng có thể truy cập để tự do sử dụng, đọc và sửa đổi .NET. Một cộng đồng gồm các nhà phát triển hoạt động tích cực trong việc duy trì và cải thiện phần mềm .NET. .NET Foundation là tổ chức phi lợi nhuận độc lập, được thành lập để hỗ trợ cộng đồng .NET. Tổ chức này cung cấp các tài nguyên học tập, dự án .NET nguồn mở và nhiều sự kiện khác nhau dành cho các nhà phát triển .NET.

### 3.4.3. SQL Server

a) Tìm hiểu về SQL Server

SQL Server là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft**,** cho phép lưu trữ, quản lý và truy vấn dữ liệu một cách hiệu quả. SQL Server sử dụng T-SQL (Transact-SQL)**,** một mở rộng của SQL (Structured Query Language), giúp tăng cường khả năng xử lý dữ liệu với các tính năng như biến, hàm, và thủ tục. SQL Server được xây dựng một cách tối ưu để có thể hoạt động trên hệ sơ sở dữ liệu rất lớn, lên đến hàng Terabyte. SQL Server cung cấp cho người dùng đầy đủ các công cụ, phục vụ việc quản lý dữ liệu từ giao diện [GUI](https://vietnix.vn/gui-la-gi/) đến ngôn ngữ truy vấn SQL. Điểm mạnh của SQL Server là có thể kết hợp được với nhiều nền tảng phổ biến như [ASP.NET](https://vietnix.vn/asp-net-la-gi/), [C#](https://vietnix.vn/c-sharp-la-gi/) để xây dựng Winform hoặc là SQL Server cũng có thể hoạt động độc lập.

b) Các tính năng nổi bật:

* Duy trì việc lưu trữ bền vững
* Tạo nhiều cơ sở dữ liệu
* Khả năng bảo mật cao
* Tạo báo cáo bằng SSRS (SQL Server Reporting Services)
* Các quá trình được thực hiện bằng SSIS (SQL Server Integration Services)

c) Tại sao nên chọn SQL Server?

* Xây dựng một đường cơ sở cho hiệu suất.
* Xác định những thay đổi về hiệu suất theo thời gian.
* Chẩn đoán các vấn đề hiệu suất cụ thể.
* Xác định các thành phần hoặc quy trình để tối ưu hóa.
* So sánh hiệu quả của các ứng dụng khách khác nhau đối với hiệu suất.
* Giúp việc kiểm tra các hoạt động của người dùng.
* Kiểm tra một máy chủ dưới các tải khác nhau.
* Kiểm tra kiến trúc cơ sở dữ liệu.
* Kiểm tra lịch trình bảo trì.
* Kiểm tra các kế hoạch sao lưu và khôi phục.
* Xác định thời điểm sửa đổi cấu hình phần cứng của bạn.

Với giao diện trực quan, dễ sử dụng, SQL Server là một giải pháp quản trị cơ sở dữ liệu toàn diện, lý tưởng cho cả các ứng dụng nhỏ và doanh nghiệp lớn, mang lại hiệu suất cao, tính bảo mật mạnh mẽ và khả năng mở rộng vượt trội.

### 3.4.4. Spring Boot

Spring Boot là một dự án con của framework Spring, được thiết kế để giúp phát triển ứng dụng Java một cách nhanh chóng và dễ dàng. Dưới đây là một số đặc điểm nổi bật và tính ưu việt của Spring Boot:

* Thuận tiện cấu hình (Convenient configuration): Spring Boot giúp tự động cấu hình môi trường ứng dụng một cách đơn giản thông qua việc sử dụng các giá trị mặc định và các cấu hình thông minh. Điều này giảm đáng kể khối lượng công việc cần thiết cho việc cấu hình.
* Embeddable web server: Spring Boot đi kèm với các web server như Tomcat, Jetty, hoặc Undertow được tích hợp sẵn trong ứng dụng, giảm thiểu sự phức tạp trong việc triển khai ứng dụng.
* Dependency Injection (DI): Spring Boot sử dụng cơ chế DI mạnh mẽ của Spring Framework, giúp quản lý và tự động kết nối các thành phần của ứng dụng.
* Standalone: Ứng dụng Spring Boot có thể chạy độc lập mà không cần các cấu hình phức tạp, điều này giúp tiết kiệm thời gian và công sức khi triển khai.
* Tích hợp tốt với Spring Ecosystem: Spring Boot tương thích và tích hợp tốt với nhiều dự án khác của Spring như Spring Data, Spring Security, Spring Cloud, giúp phát triển ứng dụng một cách linh hoạt và mạnh mẽ.
* Tự động cập nhật Dependency: Spring Boot hỗ trợ tính năng tự động cập nhật các phiên bản dependency, giúp dễ dàng duy trì và cập nhật ứng dụng.
* Annotation-Based configuration: Sử dụng các chú thích (annotation) để cấu hình thay vì sử dụng các file cấu hình XML, giúp mã nguồn trở nên gọn gàng và dễ đọc.
* Microservices development: Spring Boot được sử dụng rộng rãi trong phát triển ứng dụng dạng Microservices do tính linh hoạt và dễ triển khai
* Những đặc điểm này khiến Spring Boot trở thành một lựa chọn phổ biến trong cộng đồng phát triển Java, đặc biệt là cho việc xây dựng các ứng dụng web, dịch vụ và các hệ thống phức tạp

### 3.4.5. Docker

Docker là một công cụ đóng gói phần mềm và triển khai ứng dụng trong các container. Các ứng dụng chạy trong các container được gọi là các container Docker.

Docker giúp các lập trình viên chỉ cần chạy các container lên mà không cần phải lo về việc cài các thư viện hoặc môi trường, điều này giúp các lập trình viên phát triển ứng dụng 1 cách nhanh chóng, đảm bảo tính nhất quán giữa môi trường phát triển và môi trường triển khai.

Cấu trúc của một mô hình docker: Docker sử dụng kiến trúc client – server. Docker client sử dụng một REST API (thông qua UNIX socket, hoặc cổng mạng) để có thể giao tiếp với Docker daemon, tiến trình này thực hiện công việc tạo, chạy và phân phối các Docker container. Docker client và daemon có thể chạy trên cùng một hệ thống hoặc có thể kết nối Docker client với Docker daemon từ xa. Còn Docker registry sẽ là nơi lưu trữ các Docker image, ngoài ra Docker Hub là nơi lưu trữ Docker image công khai (public registry) mà bất kỳ ai cũng có thể sử dụng và Docker được định cấu hình mặc định để tìm image trên Docker Hub, người dùng có thể cấu hình các registry riêng tư khác để lưu trữ Docker image.

Môi trường thực thi Docker (hay còn gọi là runtime environment): Là một môi trường cô lập, nhẹ và di động, được tạo ra từ một image Docker để chạy ứng dụng của bạn. Nó cung cấp cho ứng dụng tất cả các tài nguyên cần thiết để hoạt động, bao gồm hệ điều hành, thư viện, công cụ và các cấu hình. Cách hoạt động của nó là đầu tiên tạo một image Docker bằng cách viết một Dockerfile. Dockerfile chứa các lệnh để xây dựng một môi trường hoàn chỉnh cho ứng dụng. Sau đó khởi chạy container, từ image tạo một container. Container này là một lớp (instance) đang chạy của image và là nơi ứng dụng của bạn sẽ thực thi.

Ứng dụng của docker:

* Triển khai phần mềm
* Triển khai ứng dụng web
* Chạy các bản phân phối Linux nhẹ
* Máy chủ lưu trữ
* Tạo hình ảnh Linux tùy chỉnh
* Tạo môi trường phát triển, phát triển ứng dụng
* Kiểm thử phần mềm, ứng dụng

Tại sao nên chọn Docker ?

* Có thể kích hoạt nhiều hệ điều hành trên cùng một máy.
* Nó rẻ hơn so với các phương pháp trước đây, do thiết lập cơ sở hạ tầng ít hơn,
* nhỏ gọn hơn.
* Nếu có bất kỳ trạng thái thất bại nào cũng sẽ dễ dàng để phục hồi và bảo trì.
* Cung cấp nhanh hơn các ứng dụng và tài nguyên cần thiết cho các nhiệm vụ.
* Tăng năng suất, tiện lợi hiệu quả và đáp ứng cho ngành công nghệ thông tin là rất nhiều.

### 3.4.6. React

a) Giới thiệu về React

React (Hay ReactJS, React.js) là một thư viện Javascript mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó được tạo ra bởi Jordan Walke, một kỹ sư phần mềm tại Facebook. Người bị ảnh hưởng bởi XHP (Một nền tảng thành phần HTML cho PHP). React lần đầu tiên được triển khai cho ứng dụng Newsfeed của Facebook năm 2011, sau đó được triển khai cho Instagram.com năm 2012. Nó được mở mã nguồn (open-sourced) tại JSConf US tháng 5 năm 2013.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động

trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên DOM.

b) Đặc điểm

Reactjs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều

Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể them vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi.

Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về.

Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.

Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

### 3.4.7 Javacript

a) Javascript là gì?

Javascript chính là một ngôn ngữ lập trình với khả năng đem tới sự sinh động khi thiết kế website. Đây là dạng ngôn ngữ theo kịch bản, dựa trên chính đối trượng phát triển có sẵn, hoặc là tự định nghĩa ra. Chính vì tính tiện lợi, hiệu quả ứng dụng cao mà ngôn ngữ lập trình này được tin dùng ngày càng nhiều, ứng dụng rộng rãi trong các website hiệu quả.

Việc sử dụng ngôn ngữ JS có thể ứng dụng cho mọi trình duyệt khác nhau, hiện được sử dụng phổ biến như Chrome, hay Firefox,… Hơn nữa, đây còn là ngôn ngữ lập trình hoạt động hiệu quả, được hỗ trợ đầy đủ trên các trình duyệt của thiết bị di động. Bởi thế mà việc sử dụng đa dạng, có thể đáp ứng tốt cho nhiều nhu cầu, những đòi hỏi khác nhau của người dùng

b) Javascript là ngôn ngữ lập trình dễ dàng sử dụng

Đơn giản, dễ dàng tìm hiểu và sử dụng là đặc điểm nổi bật của ngôn ngữ lập trình này. Với những cú pháp khá tương đồng với tiếng Anh thì việc sử dụng Javascript trở nên dễ dàng, dễ tiếp cận hơn rất nhiều. Thông qua mô hình DOM được sử dụng, cung cấp tới nhiều tính năng hữu ích, được viết sẵn đem lại khả năng đáp ứng tốt cho những nhu cầu, những đòi hỏi khác nhau từ phía người dùng.

Với nhiều tính năng hữu ích mà ngôn ngữ lập trình Javascript mang lại thì việc có thể phát triển kịch bản nhằm giải quyết yêu cầu, mục đích nào đó trở nên dễ dàng hơn. Những mục đích tùy chính của con người được đáp ứng hiệu quả với sự hỗ trợ của JS. Sử dụng JS đảm bảo dễ học, dễ dàng phát hiện lỗi khi xuất hiện, đồng thời nó hoạt động nhanh chóng và nhẹ nhàng hơn nhiều ngôn ngữ lập trình khác. Bởi thế, quá trình sử dụng có thể thích hợp với nhiều đối tượng người dùng, đem lại hiệu quả cao và tránh những ảnh hưởng không mong muốn có thể xảy ra

c) Ưu điểm

**-** Đối với lập trình viên: Đây là ngôn ngữ dễ học, dễ để phát hiện và sửa lỗi hơn. Thông qua JavaScript thì lập trình viên cũng có thể kiểm tra dữ liệu đầu vào, nhằm giảm bớt công việc kiểm tra thủ công. JavaScript cũng khá linh hoạt, và nó có thể được sử dụng ở nhiều nền tảng, trình duyệt, và không cần những công cụ quá phức tạp bởi chúng có thể được biên dịch bởi HTML từ trình duyệt web.

**-** Đối với khách truy cập: Ta có thể truy cập và tương tác với website hiệu quả hơn. Nhờ đặc tính gọn nhẹ mà chúng sẽ cho phép thực hiện các tác vụ trên trang web nhanh hơn.

## 3.5 Kết luận Chương III

Chương III đã trình bày chi tiết về kiến trúc hệ thống, thiết kế cơ sở dữ liệu và các công nghệ được sử dụng trong dự án. Thông qua việc phân tích kiến trúc, chương đã làm rõ cách các thành phần trong hệ thống phối hợp với nhau để đạt được các mục tiêu về hiệu quả, bảo mật và khả năng mở rộng. Phần thiết kế cơ sở dữ liệu đảm bảo hệ thống có khả năng quản lý dữ liệu một cách chặt chẽ, hỗ trợ tốt cho các chức năng nghiệp vụ. Bên cạnh đó, việc lựa chọn các công nghệ tiên tiến và phù hợp đã tạo nền tảng vững chắc cho quá trình triển khai và phát triển hệ thống.

Những nội dung trình bày trong chương này không chỉ đặt nền móng cho việc triển khai hệ thống mà còn thể hiện sự chuẩn bị kỹ lưỡng, hợp lý để đạt được các yêu cầu và mục tiêu của dự án. Chương tiếp theo sẽ tập trung vào việc triển khai và thử nghiệm hệ thống dựa trên các thiết kế đã được đề xuất.

# CHƯƠNG IV. TRIỂN KHAI VÀ KIỂM THỬ HỆ THÔNG

Sau quá trình phân tích, thiết kế và xây dựng hệ thống, chương IV tập trung vào giai đoạn triển khai và kiểm thử, một bước quan trọng để đưa ứng dụng vào hoạt động thực tế. Chương này sẽ mô tả chi tiết quá trình cài đặt hệ thống, cấu hình môi trường, và thực hiện các bài kiểm thử để đảm bảo hệ thống hoạt động ổn định, hiệu quả và đáp ứng đúng các yêu cầu đã đề ra.

## 4.1 Triển khai hệ thống

### 4.1.1. Môi Trường Triển Khai:

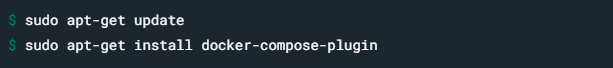
* Frontend**:**
  + Nền tảng: Vercel.
  + Các công nghệ frontend: ReactJS, JavaScript.
* Backend (Hệ Thống Quản Lý Lớp Học):
  + Môi trường: VPS (Virtual Private Server)
  + Hệ điều hành: Ubuntu 20.04 Server.
* Hệ Thống Biên Dịch Thực Thi Code:
  + Môi trường: VPS (Virtual Private Server).
  + Hệ điều hành: Ví dụ, Ubuntu 20.04 Server.

### 4.1.2. Quy Trình Triển Khai:

* Triển khai Frontend (Vercel):
  1. Chuẩn bị code: Đảm bảo code frontend đã được commit lên GitHub.
  2. Tạo tài khoản Vercel: Đăng ký tài khoản và tạo project trên Vercel.
  3. Kết nối với GitHub: Kết nối Vercel với repository chứa code frontend.
  4. Cấu hình build settings: Cấu hình các lệnh build và thư mục output cho dự án React.
  5. Deploy: Deploy ứng dụng lên Vercel.
  6. Kiểm tra: Kiểm tra ứng dụng trên Vercel.
* Triển khai Backend (Hệ Thống Quản Lý Lớp Học) trên VPS:
  1. Build image api bằng cách truy cập vào folder chứa dockerfile của api và chạy câu lệnh: docker build -t huytk5/lmsapi:latest
  2. Đẩy image api tạo ở bước 1 lên dockerhub bằng 2 câu lệnh docker tag huytk5/lmsapi huytk5/lmsapi:latest và docker push huytk5/lmsapi
  3. ThuêVPS**:** Chọn nhà cung cấp VPS và cấu hình VPS.
     + Nhóm đã lựa chọn dịch vụ VPS của H2Cloud vì chi phí rẻ, có thể thuê theo tháng, phù hợp với việc triển khai hệ thống trong thời gian ngắn
  4. Truy cập VPS và cài đặt các phần mềm cần thiết
     + Cài đặt docker bằng các câu lệnh



* + - Cài đặt dockercompose bằng các câu lệnh



Hình 4.1 Lệnh cài đặt docker-compose

* 1. Kéo image api được đẩy lên ở xử lý 2 từ dockerhub về vps sử dụng câu lệnh: *docker pull huytk5/lmsapi:latest*
  2. Kéo image SQL Server do chính microsoft phát hành sử dụng câu lệnh: *docker pull mcr.microsoft.com/mssql/server:2022-latest*
  3. Gửi file docker-compose từ máy local sang vps
     + File docker-compose.yml để quản lý continer cho api và sql server

A screen shot of a computer

Description automatically generated

Hình 4.2 docker-compose file của hệ thống back-end

* + - Gửi file docker-compose.yml trên lên vps bằng cách truy cập folder chứa file và gọi câu lệnh scp docker-compose.yml [root@36.50.135.228:/root](mailto:root@36.50.135.228:/root)
  1. Tạo và khởi chạy các container trên vps sử dụng docker-compose.yml file bằng câu lệnh docker-compose up --build -d
  2. Khôi phục cơ sở dữ liệu từ máy local lên vps
     + Tạo file bak từ dữ liệu local bằng cách chọn Database -> Task -> Backup -> SQL Server sẽ tạo file bak để backup
     + Truy cập vào folder C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER2022\MSSQL\Backup lưu trữ file .bak vừa generate
     + Gửi file .bak đến vps bằng câu lệnh: scp LearningManagementSystem.bak [root@36.50.135.228:/root](mailto:root@36.50.135.228:/root)
     + Xóa file .bak hiện tại trong cơ sở dữ liệu (nếu có) bằng cách vào: docker exec -u root -it <containerid> bash. Truy cập vào foler var/opt/mssql/data/ và remove LearningManagementSystem.bak
     + Chuyển file bak từ root vps sang dữ liệu của docker container với câu lệnh: *docker cp /root/LearningManagementSystem.bak sqlservercontainer:/var/opt/mssql/data/LearningManagementSystem.bak*
     + Đăng nhập vào sql container với câu lệnh: */opt/mssql-tools/bin/sqlcmd -S localhost -U sa -P <Password>*
     + Khôi phục dữ liệu bằng câu lệnh: *RESTORE DATABASE LearningManagementSystem FROM DISK = '/var/opt/mssql/data/LearningManagementSystem.bak' WITH MOVE 'LearningManagementSystem' TO '/var/opt/mssql/data/LearningManagementSystem.mdf', MOVE 'LearningManagementSystem\_log' TO '/var/opt/mssql/data/LearningManagementSystem\_log.ldf';*
* Triển khai Hệ Thống Biên Dịch Thực Thi Code trên VPS:
  1. ThuêVPS**:** Chọn nhà cung cấp VPS và cấu hình VPS.
  2. Cài đặt Docker, docker-compose và Git: Cài đặt Docker, docker-compose và Git trên hệ điều hành của VPS.
  3. Kéo code dự án về VPS: Dùng lệnh git clone để lấy code dự án trên github về VPS
  4. Build Docker Image: Sử dụng lệnh docker compose build để tạo image từ Dockerfile của backend.
  5. Triển khai Docker Container và khởi động ứng dụng: Dùng docker compose up để chạy các container và truy cập các API endpoint để đảm bảo hệ thống biên dịch code đã chạy
* A screenshot of a computer program

  Description automatically generated

Hình 4.3 Dockerfile của hệ thống thực thi code

A screenshot of a computer program

Description automatically generated

Hình 4.4 docker-compose file của hệ thống thực thi code

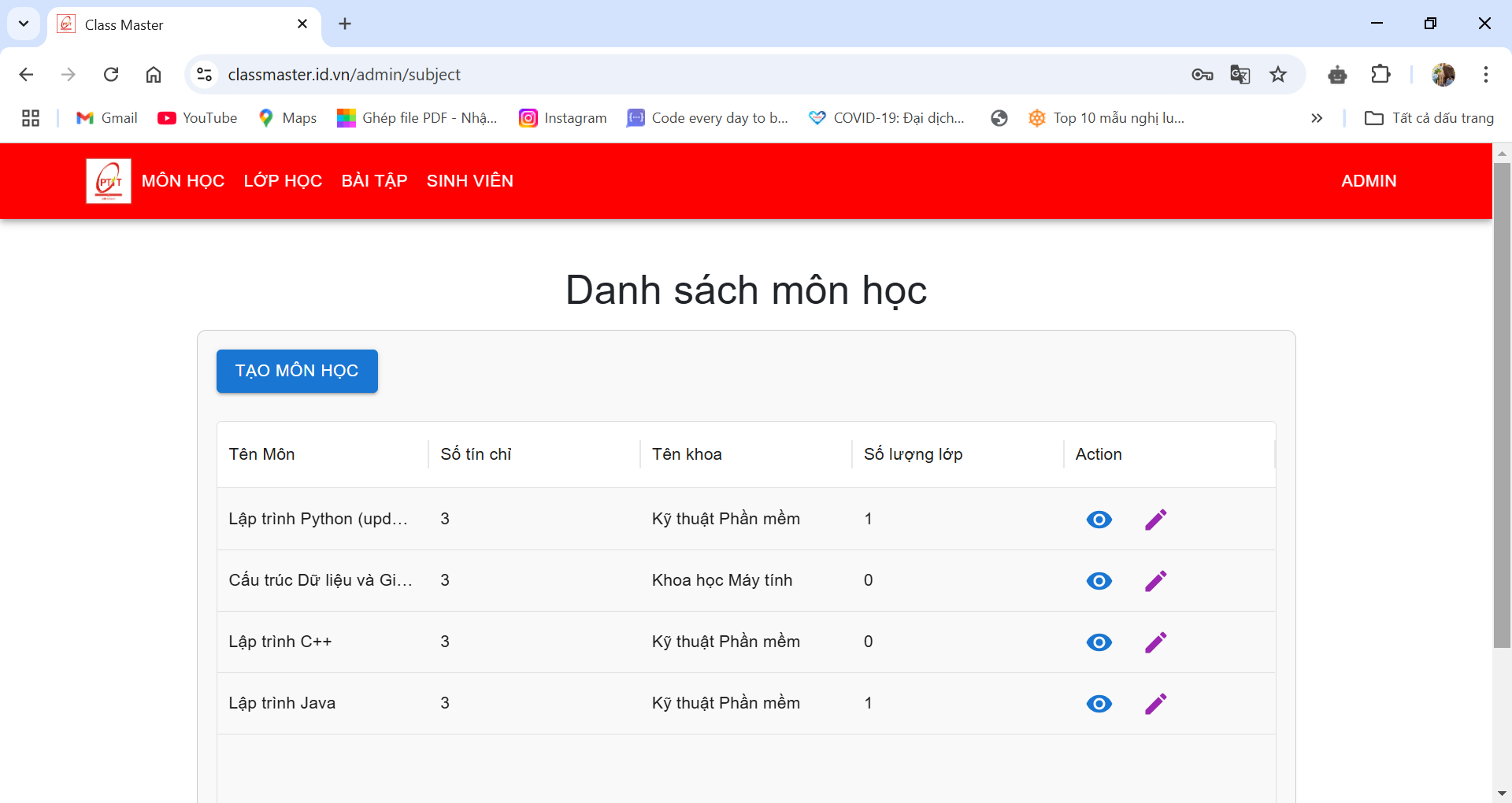
### 4.1.3. Kết Quả Triển Khai:

* Hệ thống đã được deploy thành công trên các nền tảng (Vercel và VPS).
* Các thành phần (frontend, backend, hệ thống biên dịch) hoạt động tốt, kết nối và giao tiếp với nhau.
* Hệ thống có thể sử dụng được ở mức cơ bản.
* Hệ thống đang được triển khai tại đường dẫn https://classmaster.id.vn/
* Dưới đây là các hình ảnh giao diện hệ thống:

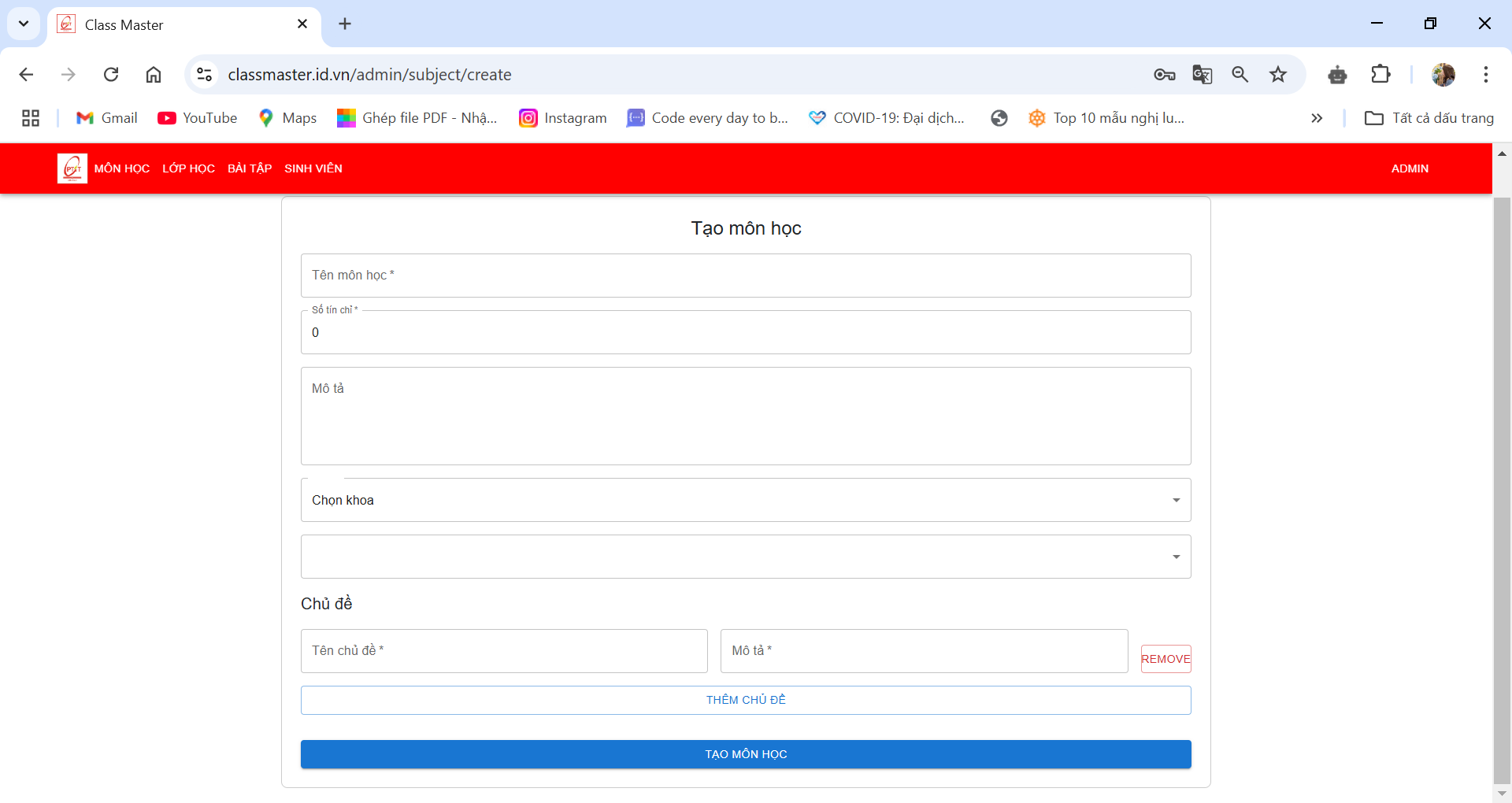
Hình 4.5 Giao diện trang chủ chính



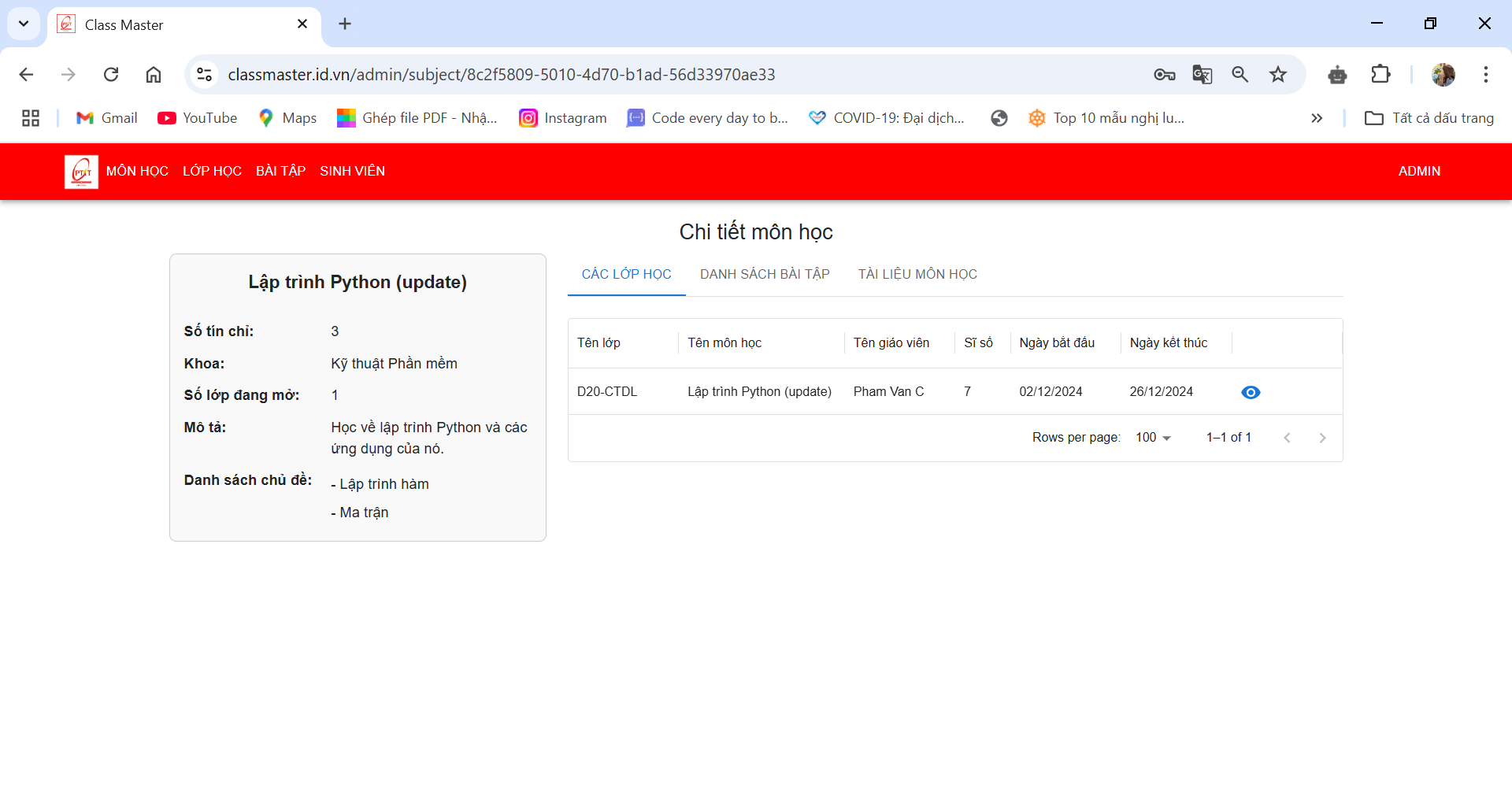
Hình 4.6 Giao diện trang chủ admin, giao diện xem danh sách môn học



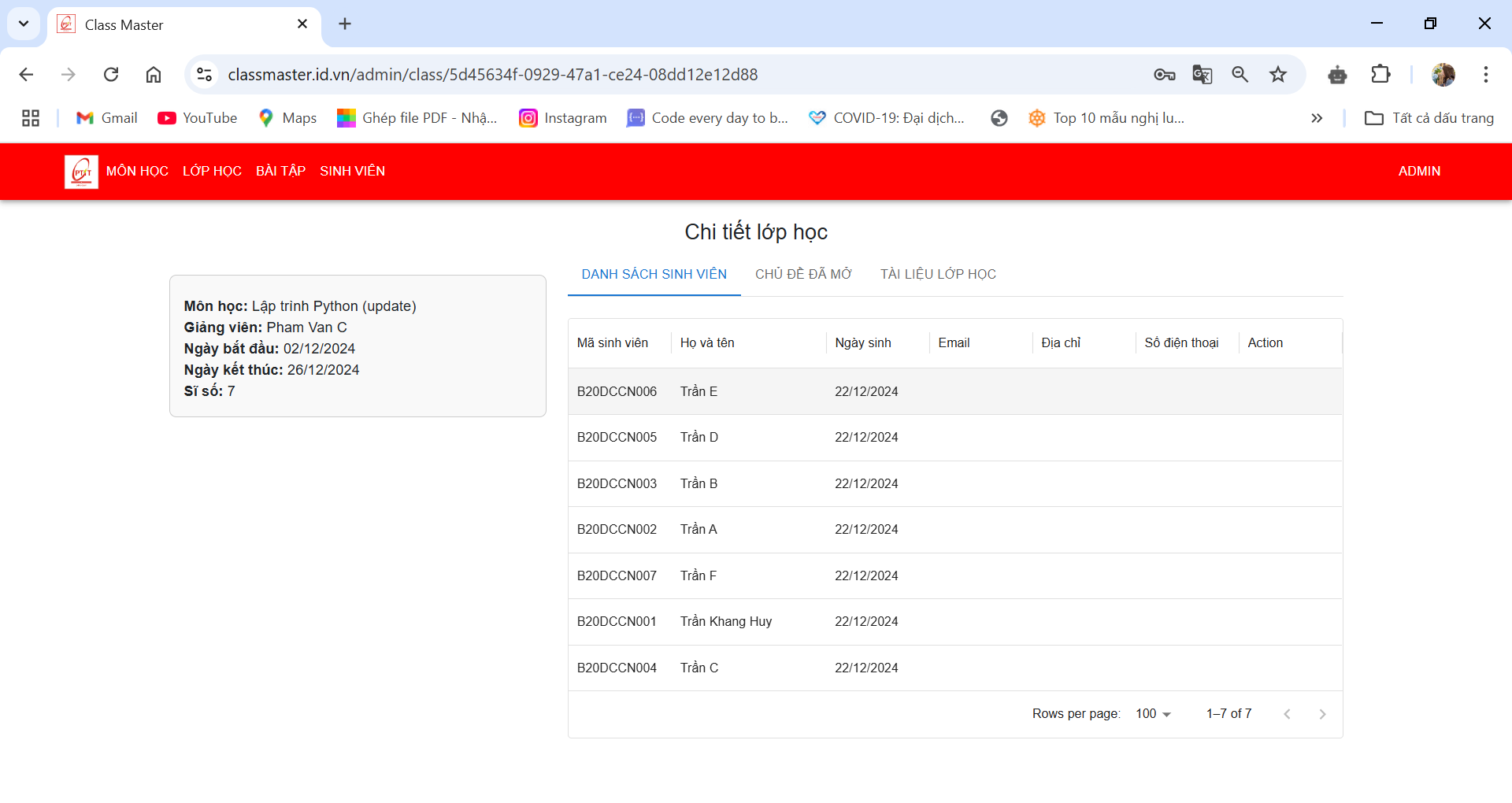
Hình 4.7 Giao diện tạo môn học



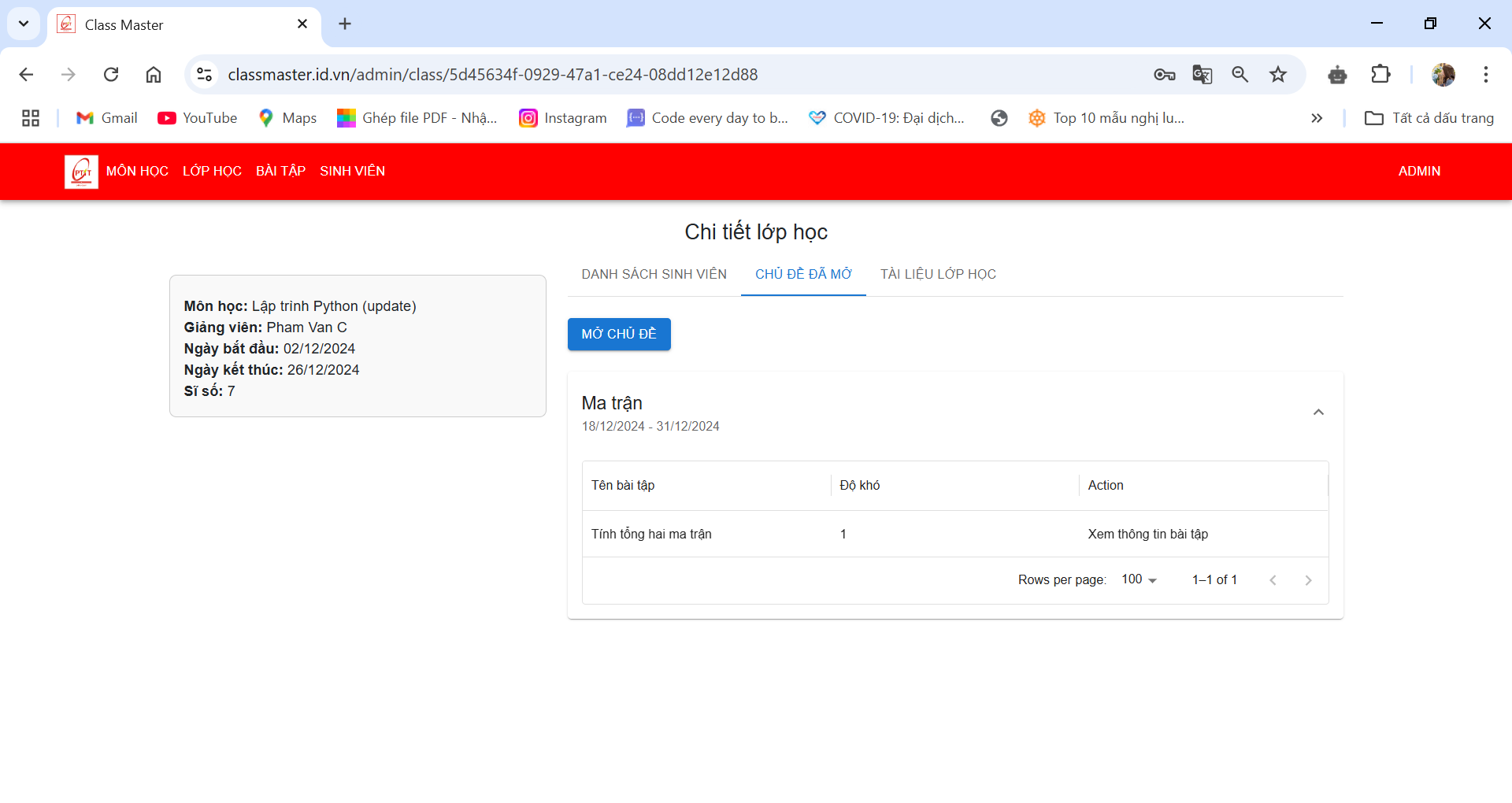
Hình 4.8 Giao diện xem chi tiết môn học



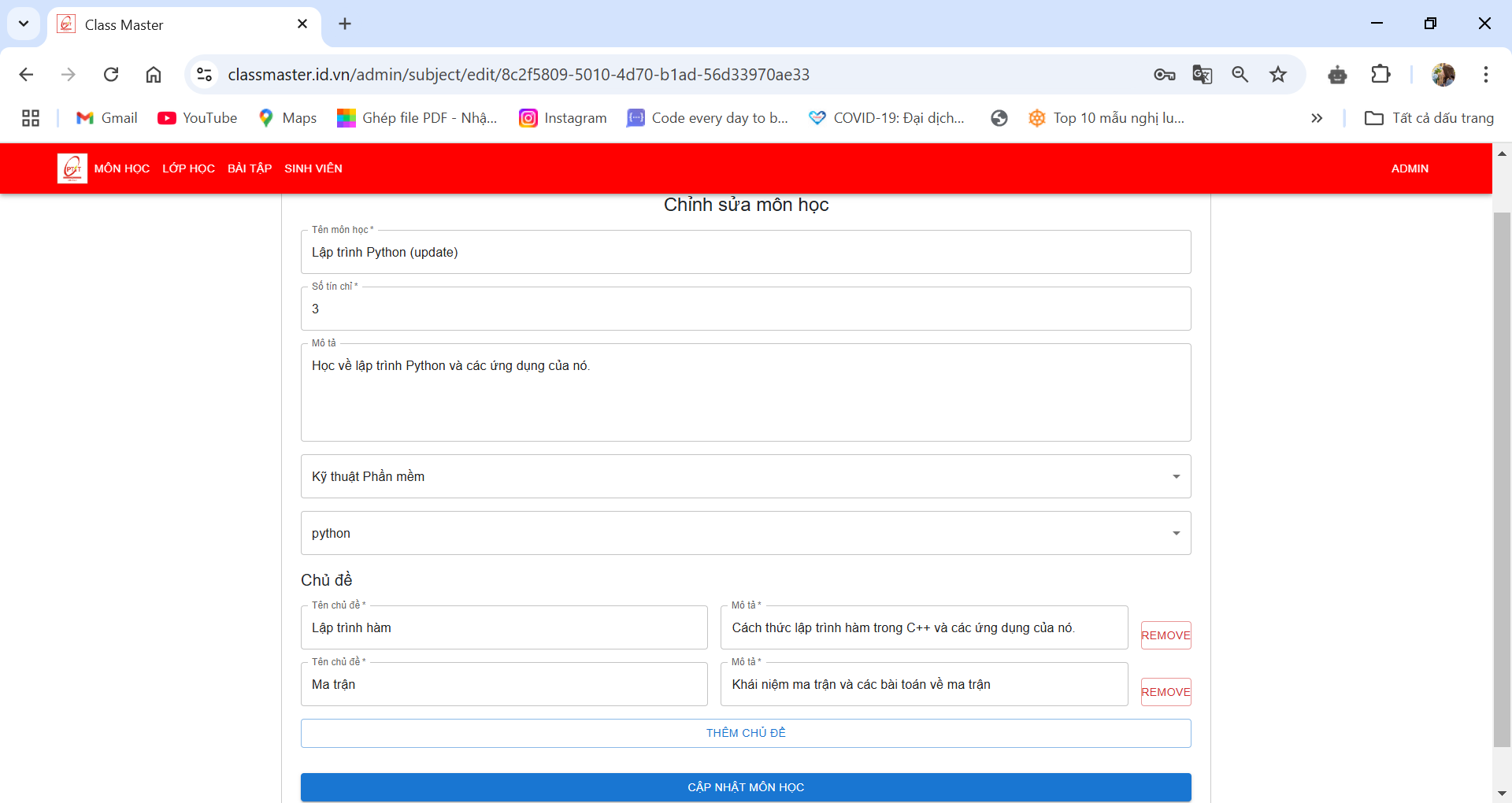
Hình 4.9 Giao diện xem chi tiết lớp học



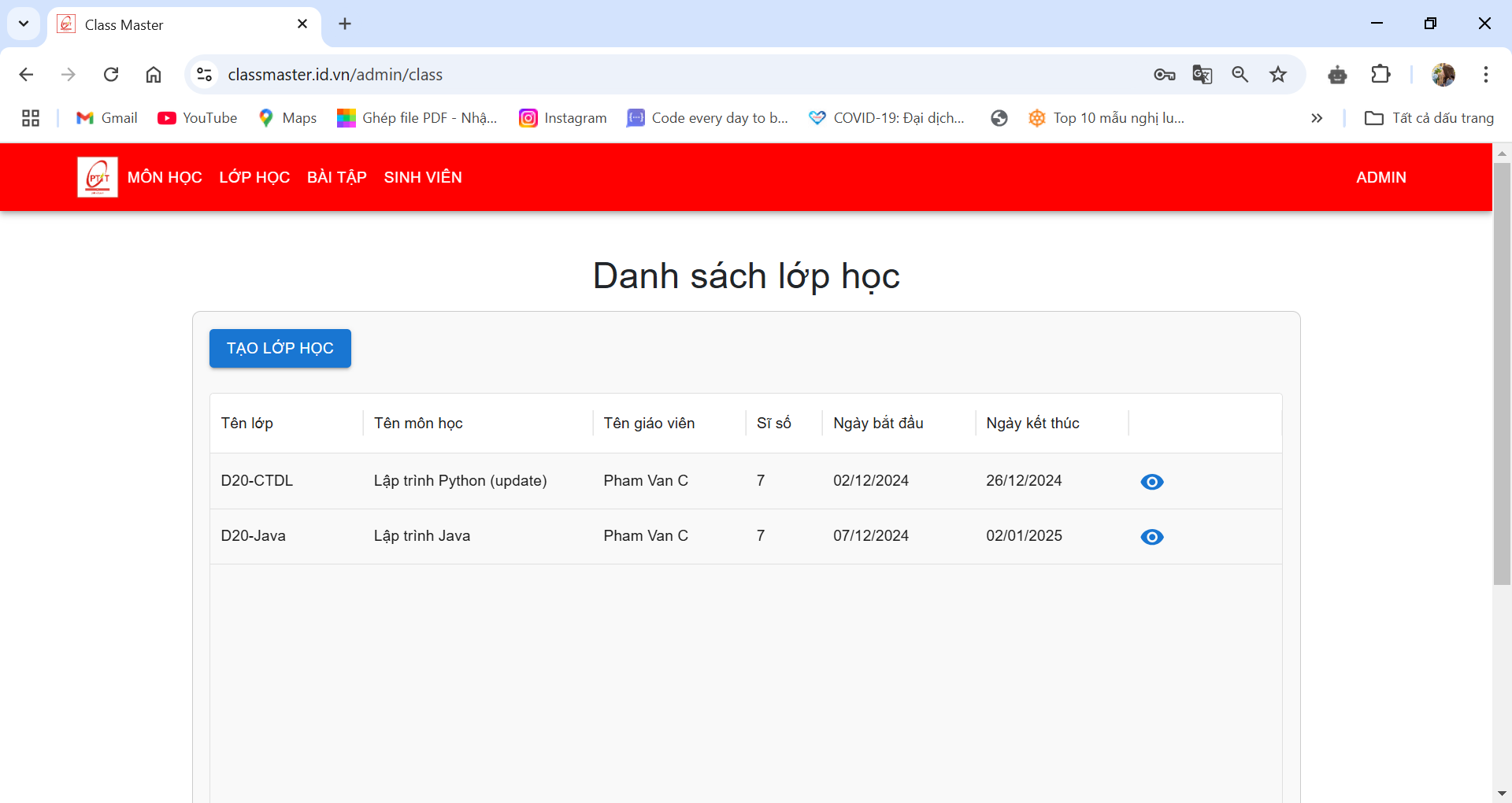
Hình 4.10 Giao diện xem các chủ đề đã mở của lớp học



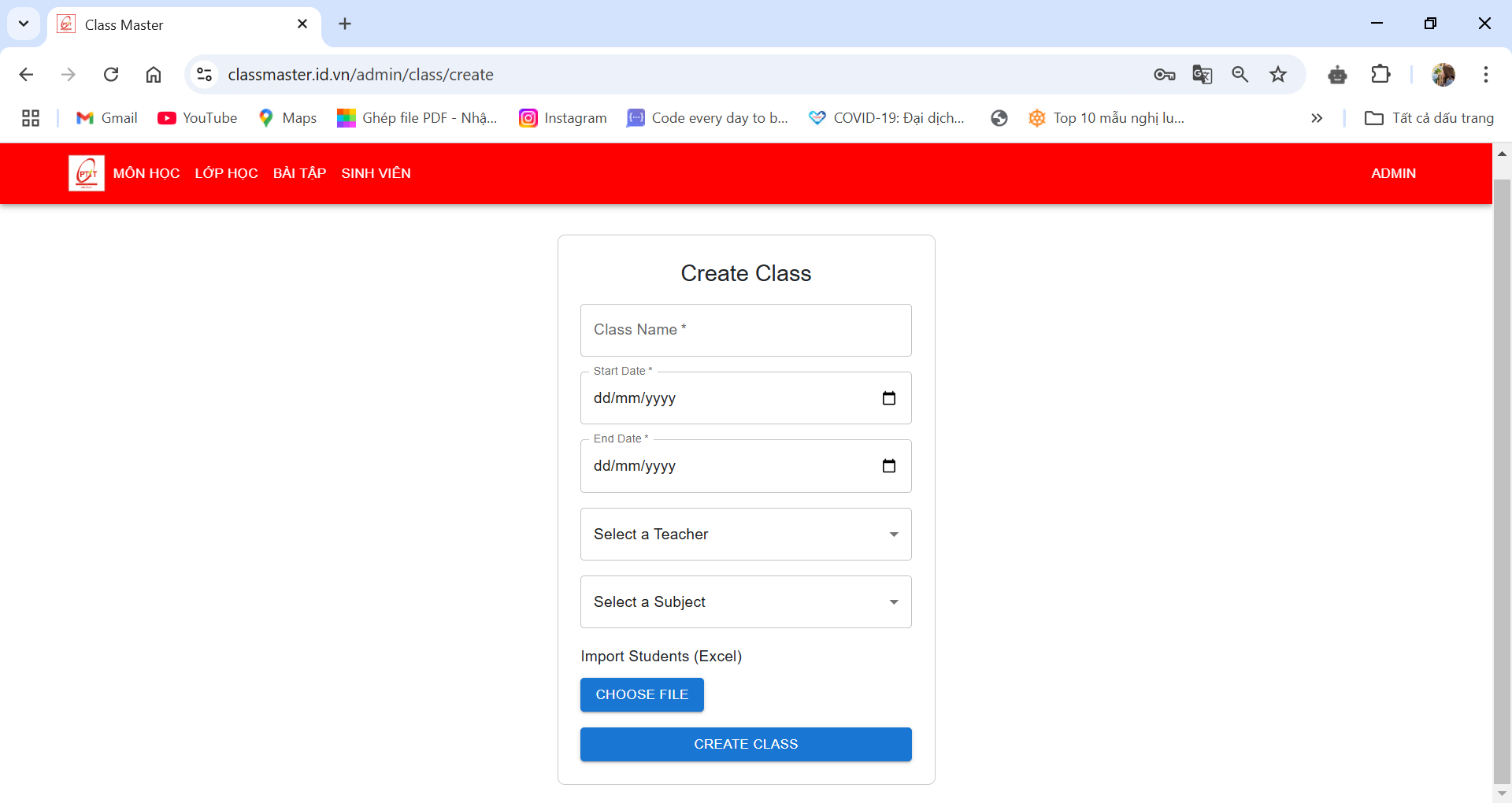
Hình 4.11 Giao diện chỉnh sửa môn học



Hình 4.12 Giao diện xem danh sách lớp học



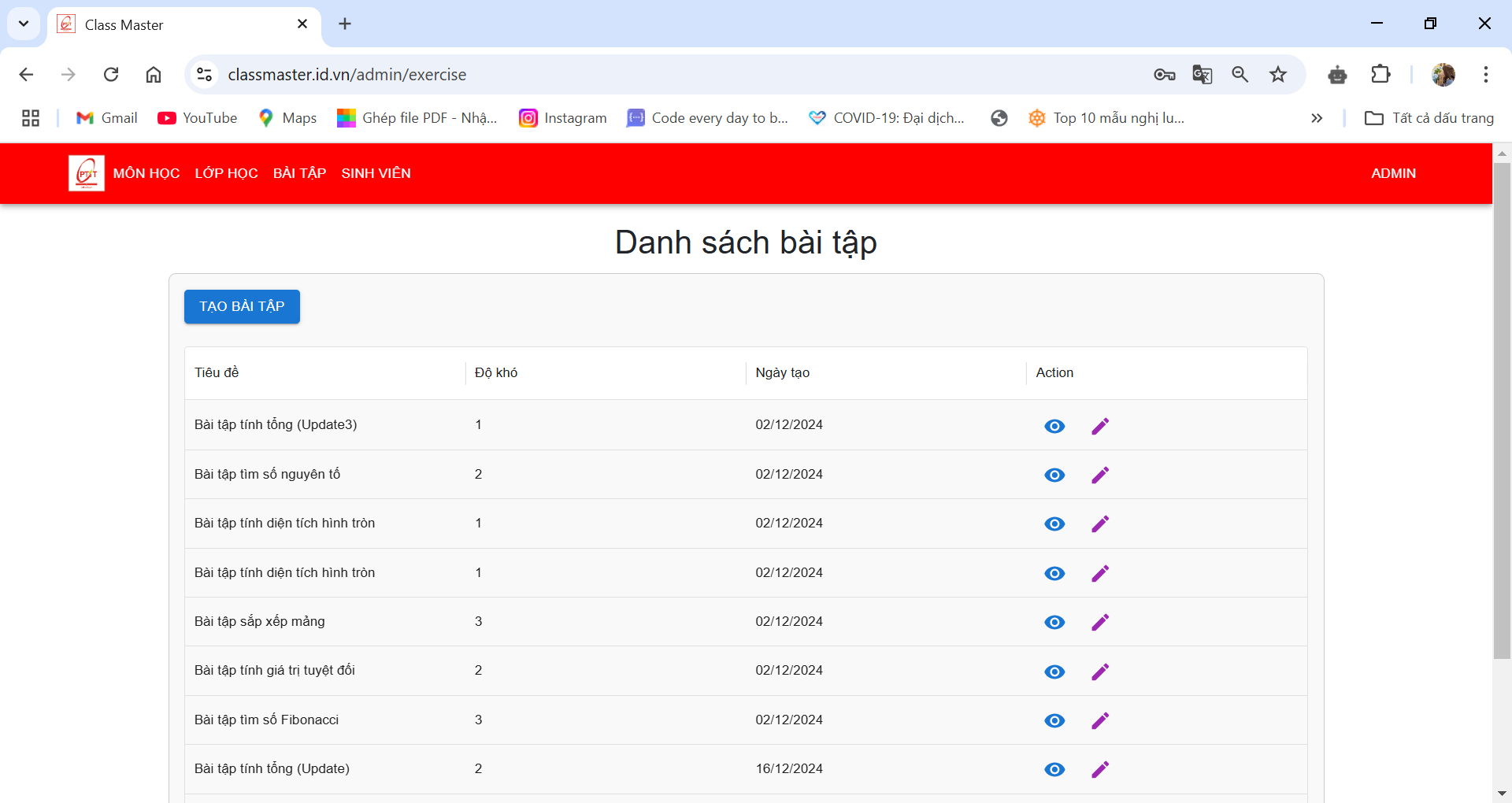
Hình 4.13 Giao diện tạo lớp học



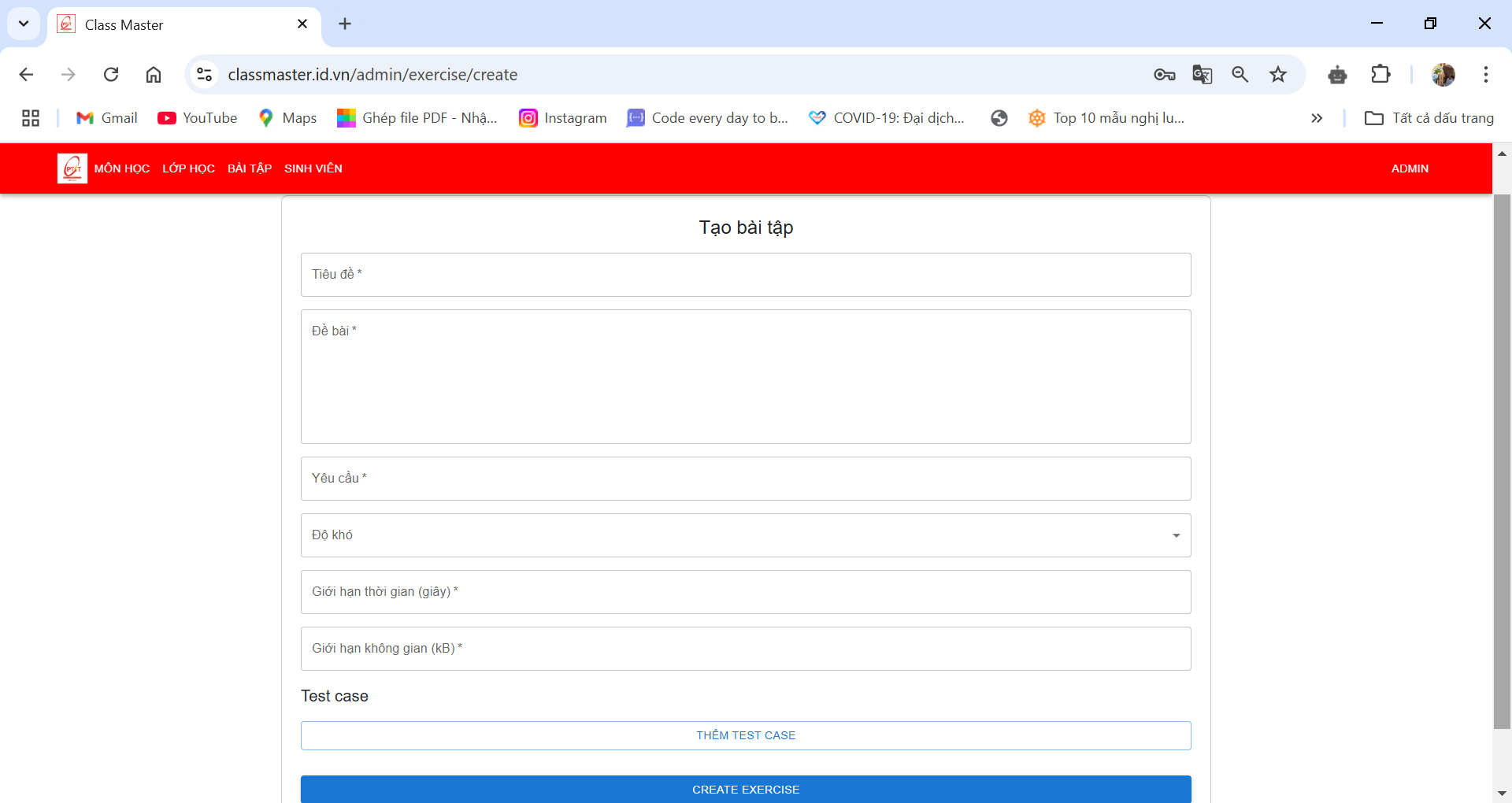
Hình 4.14 Giao diện xem chi tiết lớp học



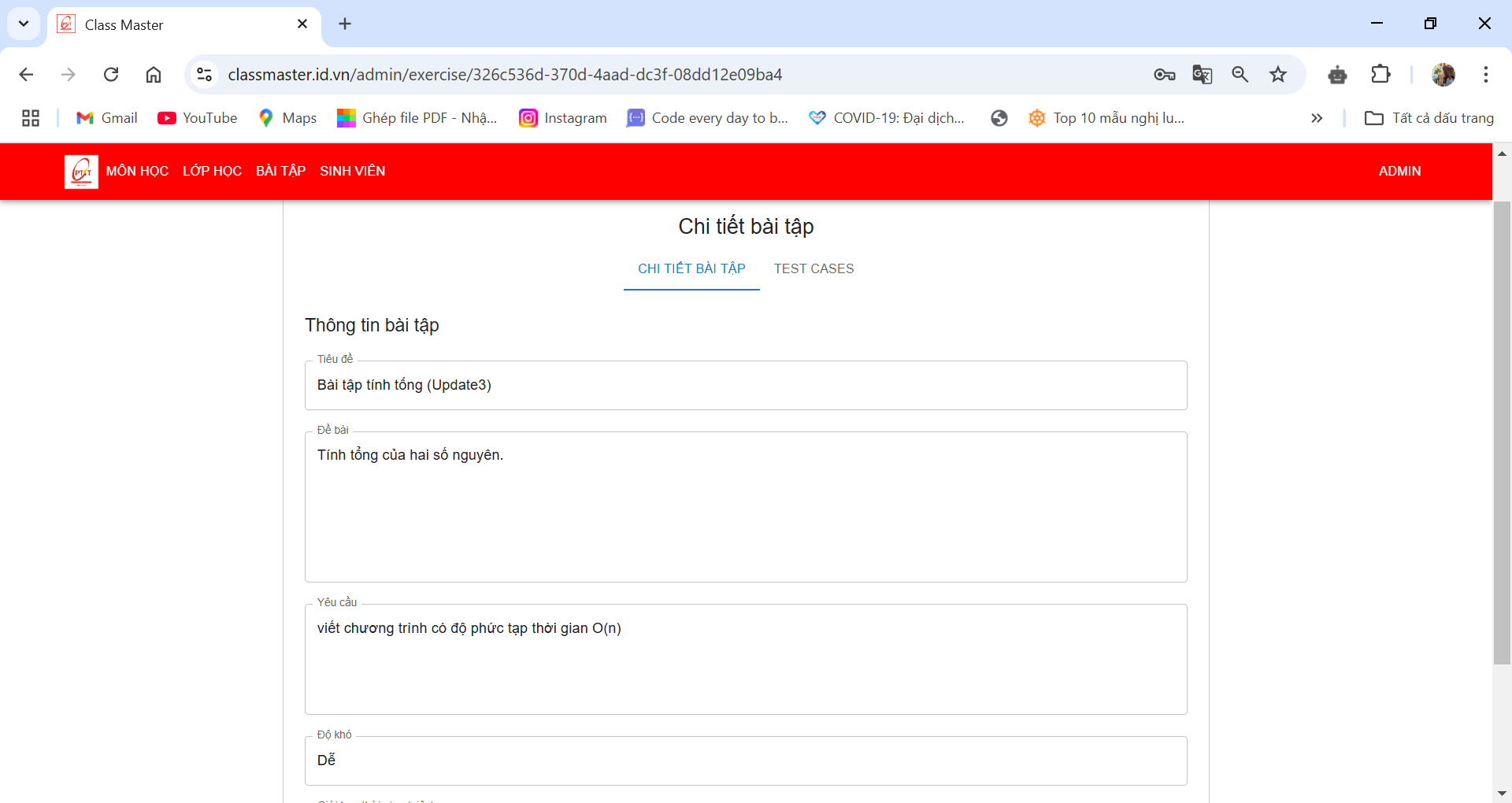
Hình 4.15 Giao diện xem danh sách bài tập



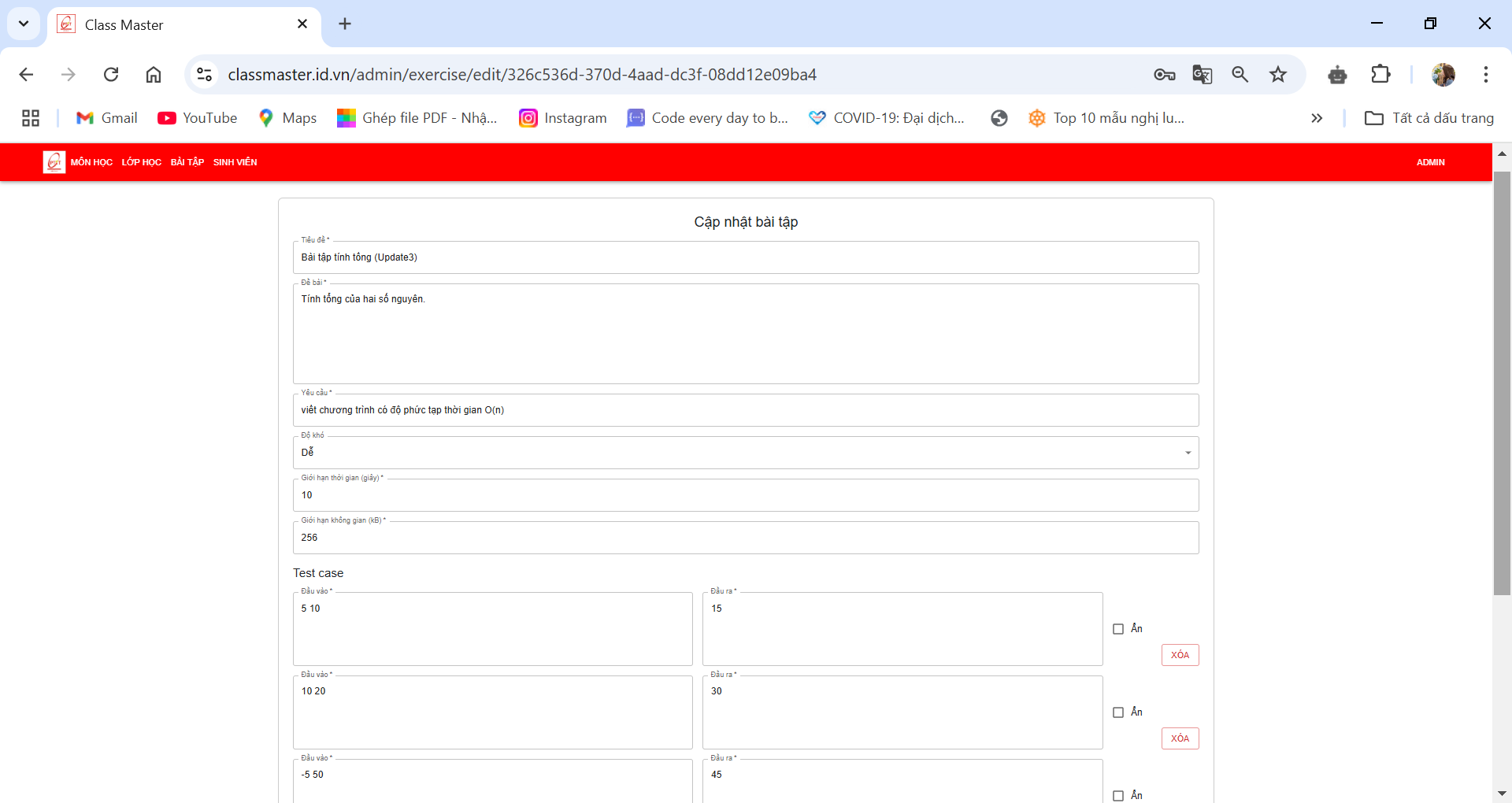
Hình 4.16 Giao diện tạo bài tập



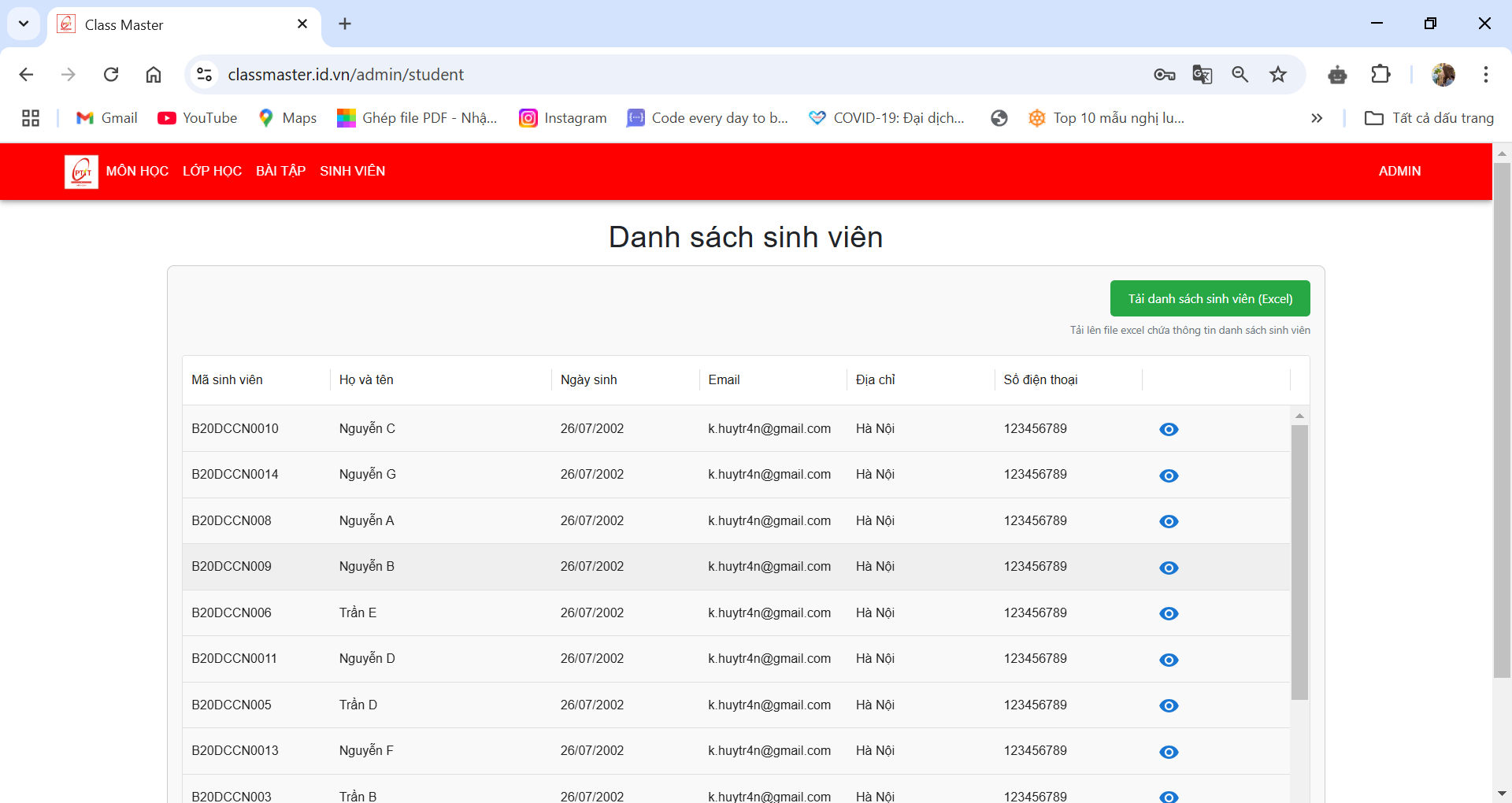
Hình 4.17 Giao diện xem chi tiết bài tập của admin



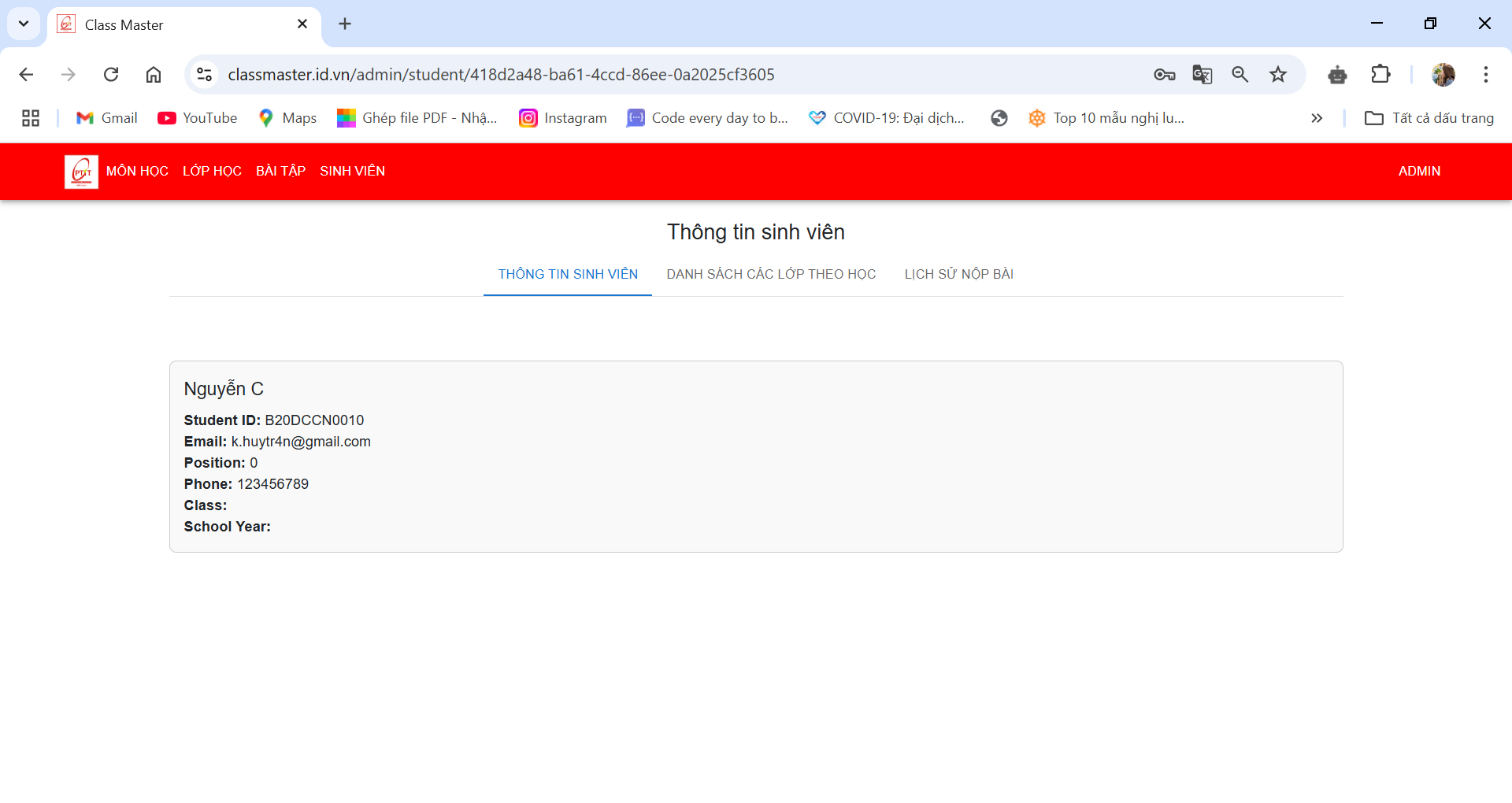
Hình 4.18 Giao diện cập nhật bài tập



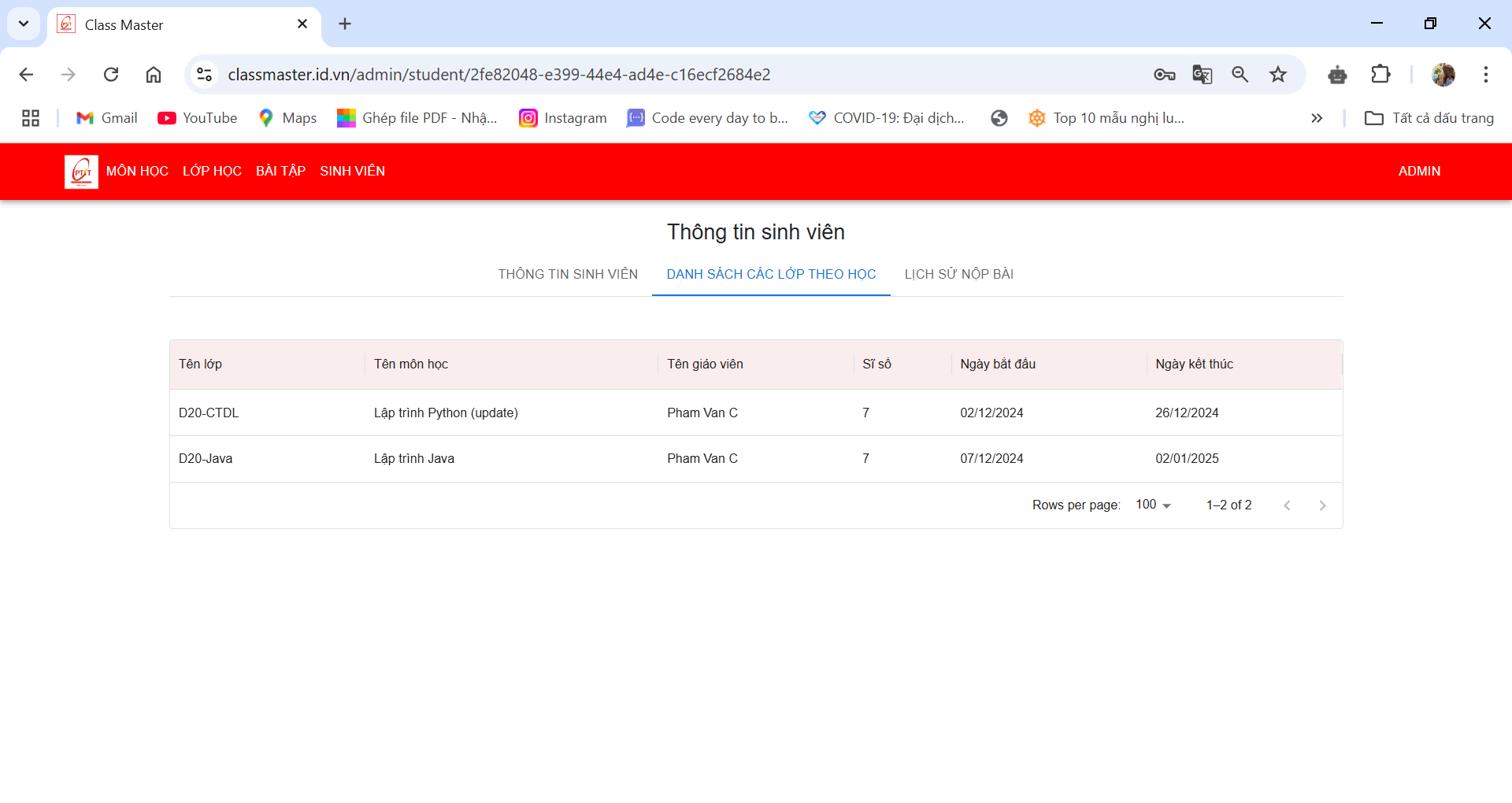
Hình 4.19 Giao diện tải danh sách sinh viên từ file excel



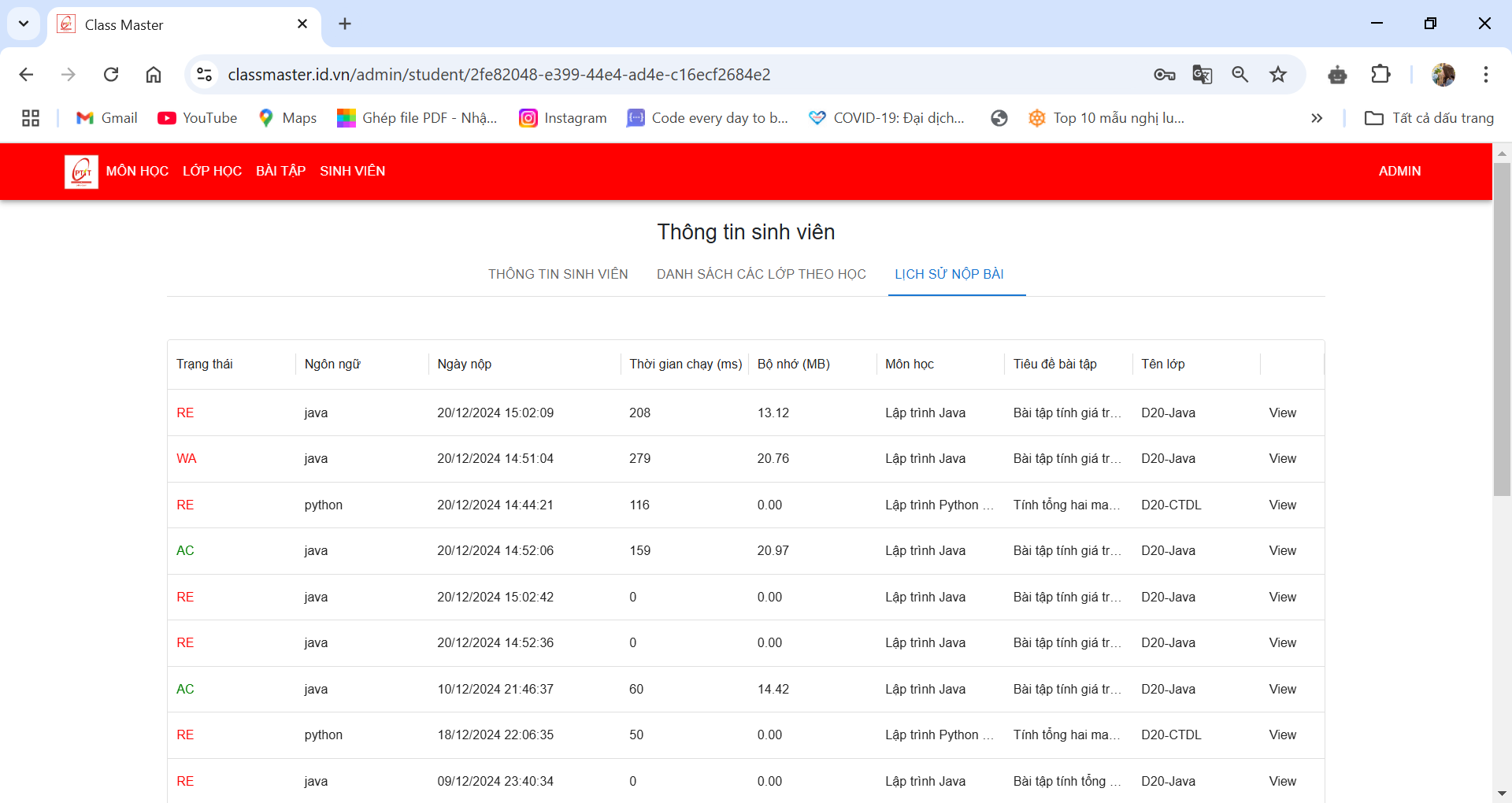
Hình 4.20 Xem chi tiết một sinh viên



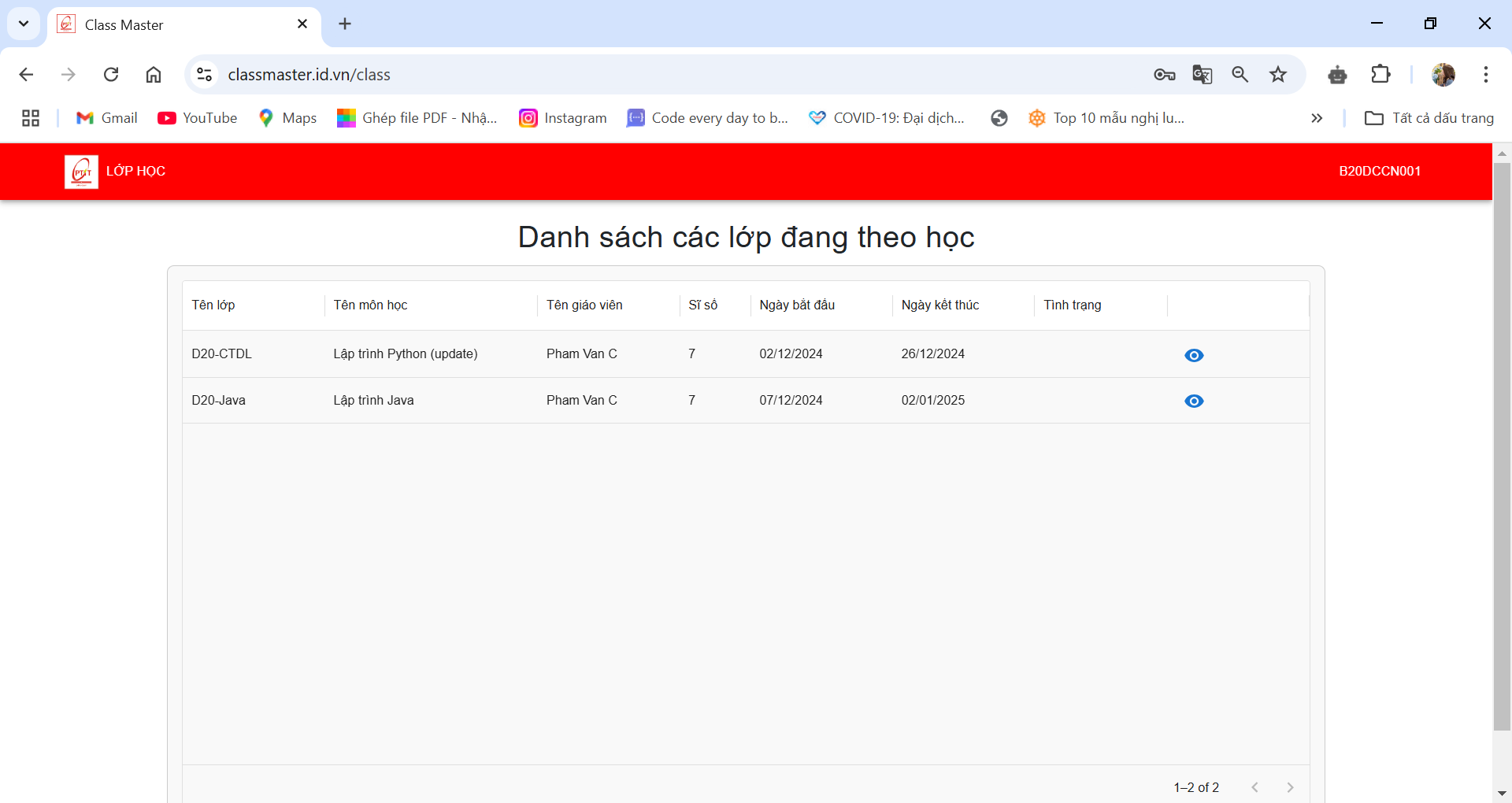
Hình 4.21 Xem danh sách các lớp theo học của sinh viên



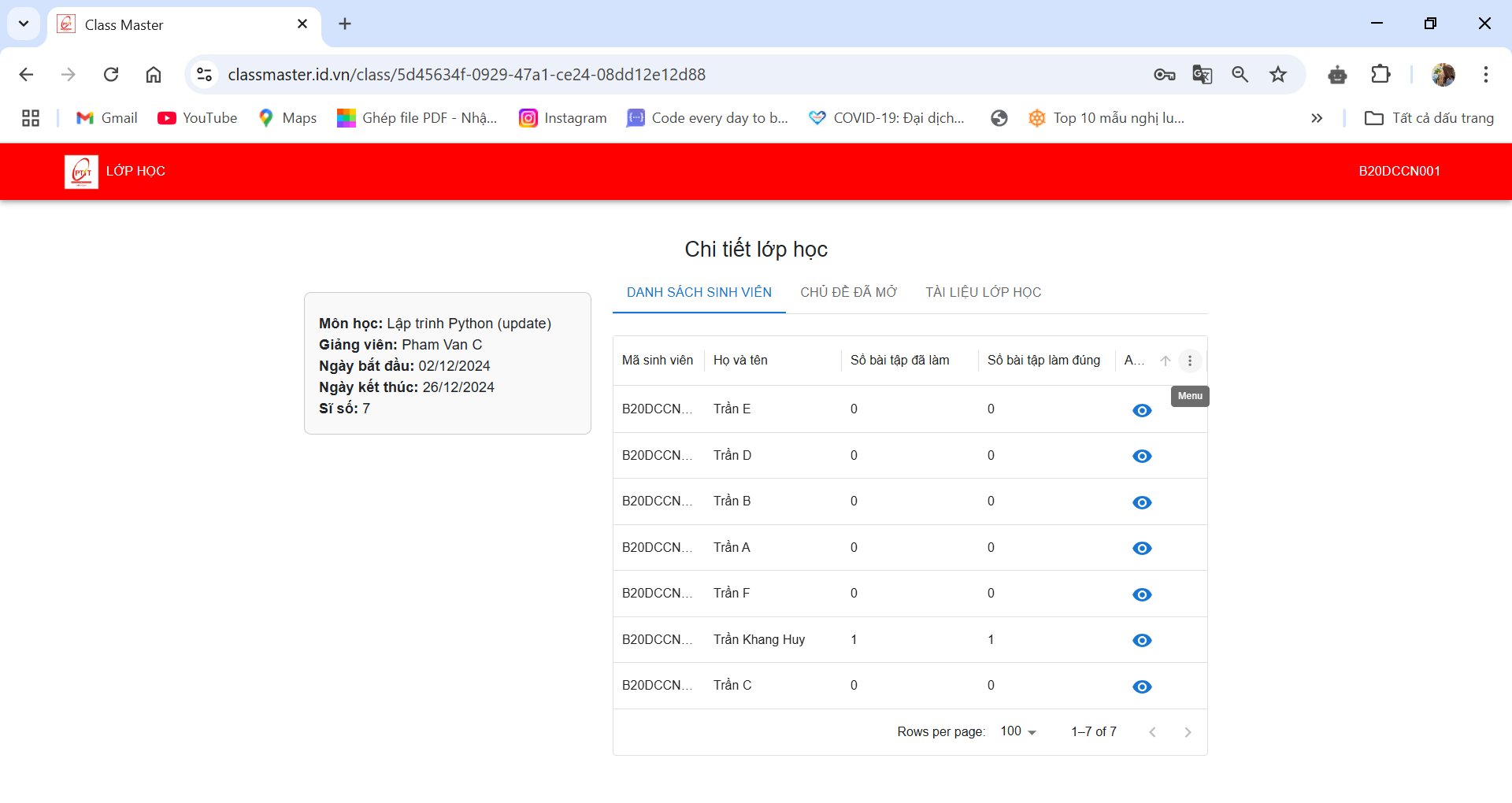
Hình 4.22 Xem lịch sử nộp bài của sinh viên



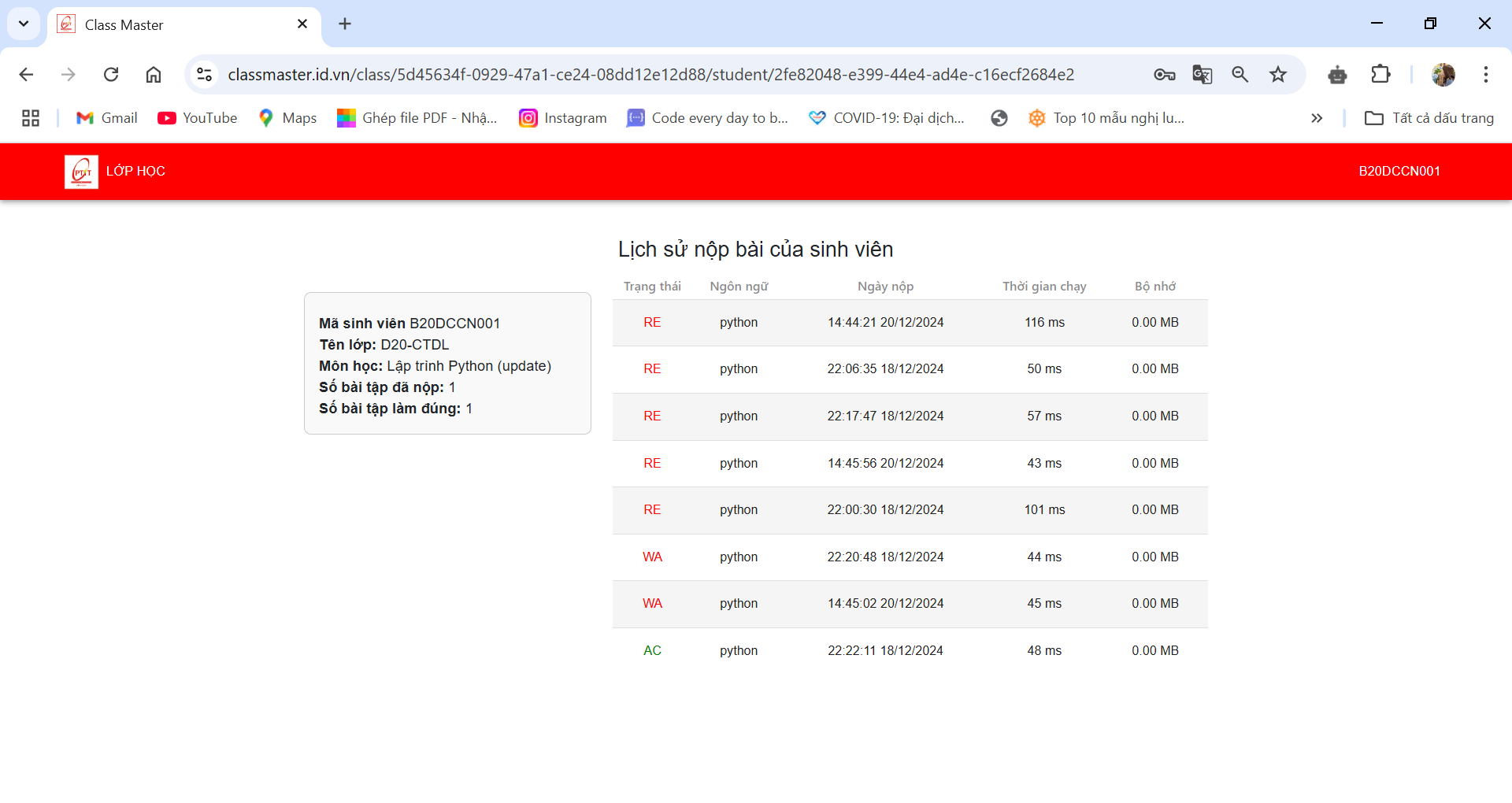
Hình 4.23 Sinh viên xem các lớp mình đang theo học



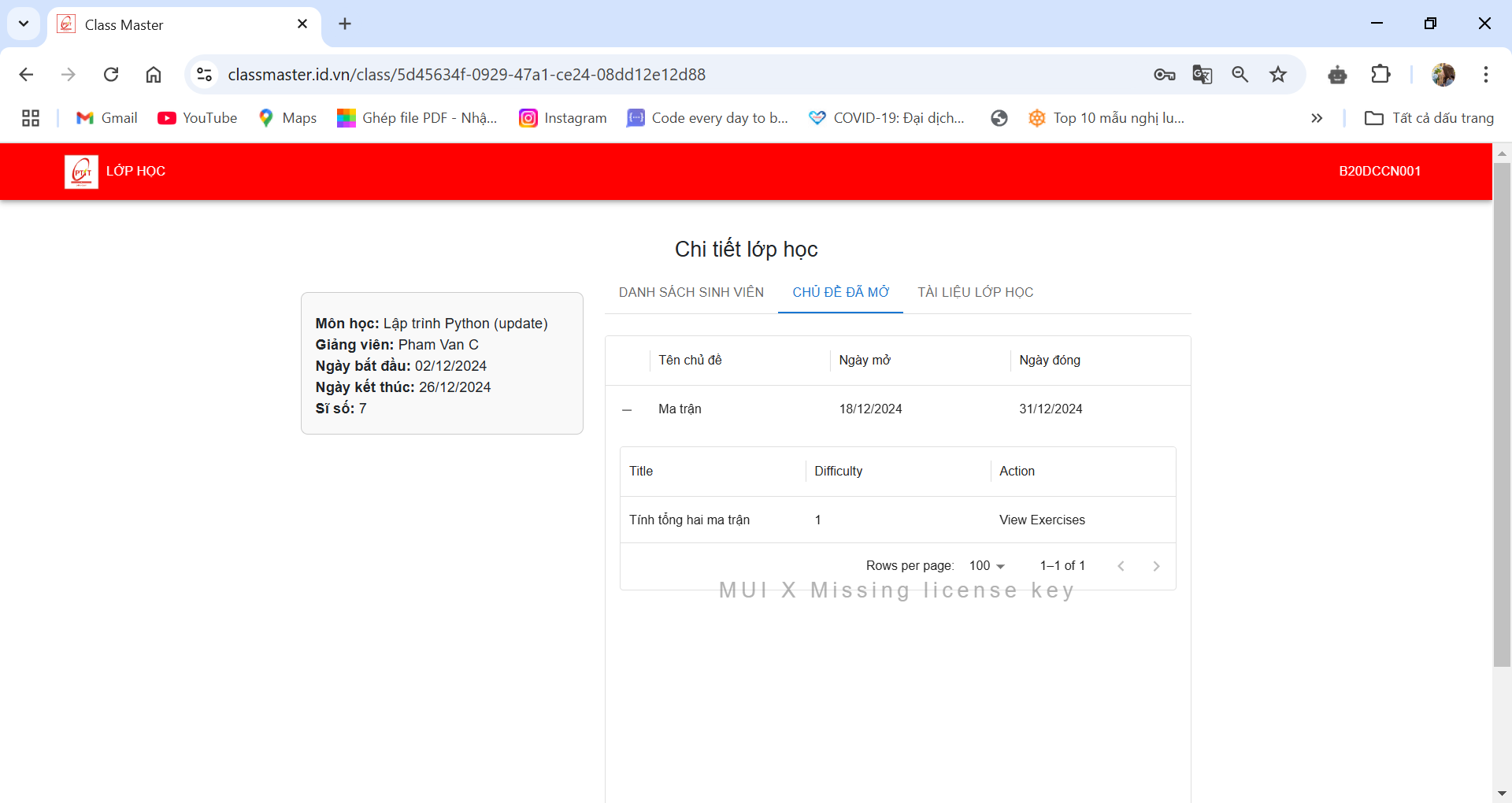
Hình 4.24 Sinh viên xem chi tiết lớp học



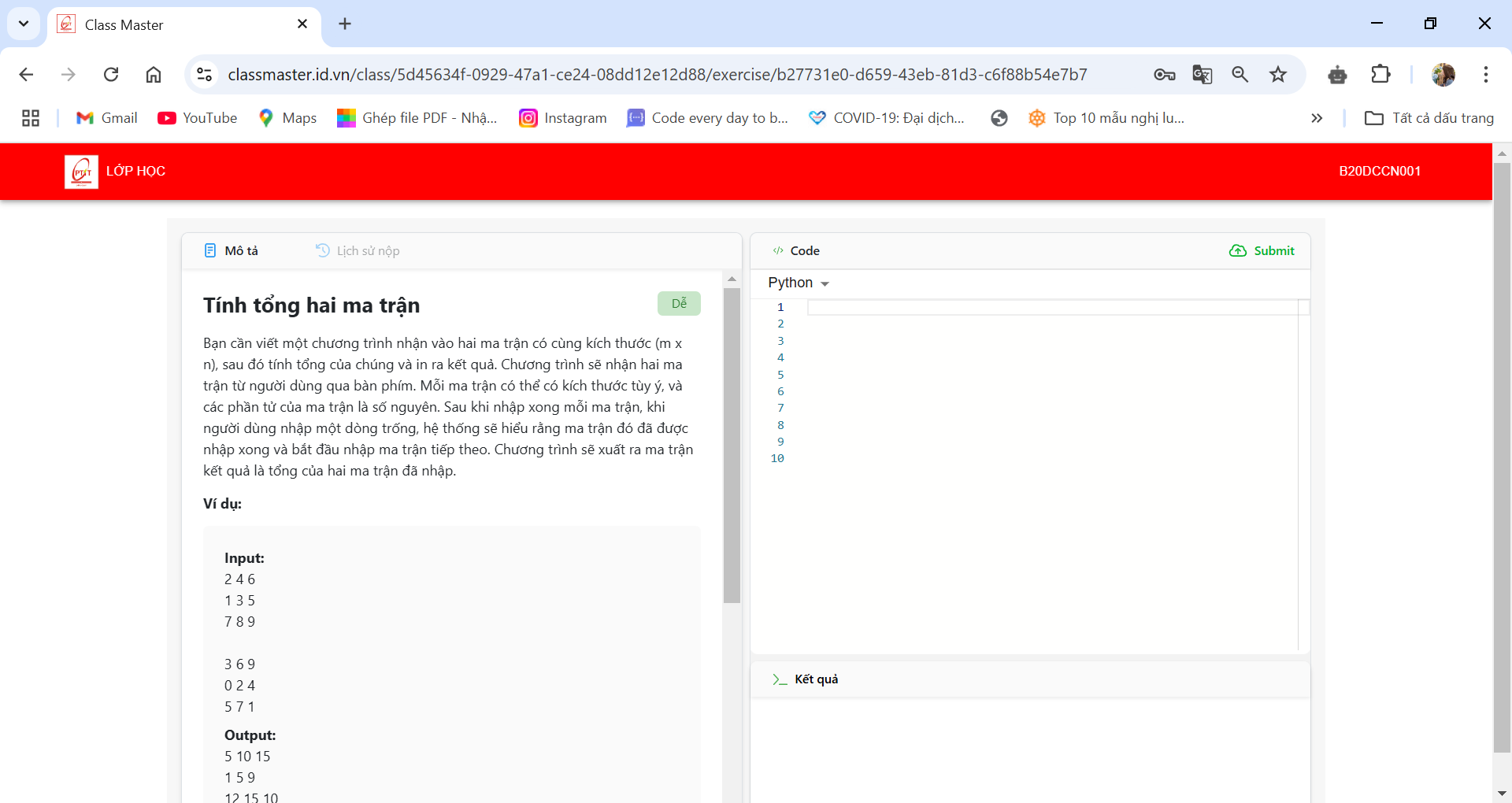
Hình 4.25 Sinh viên xem lịch sử nộp bài của mình



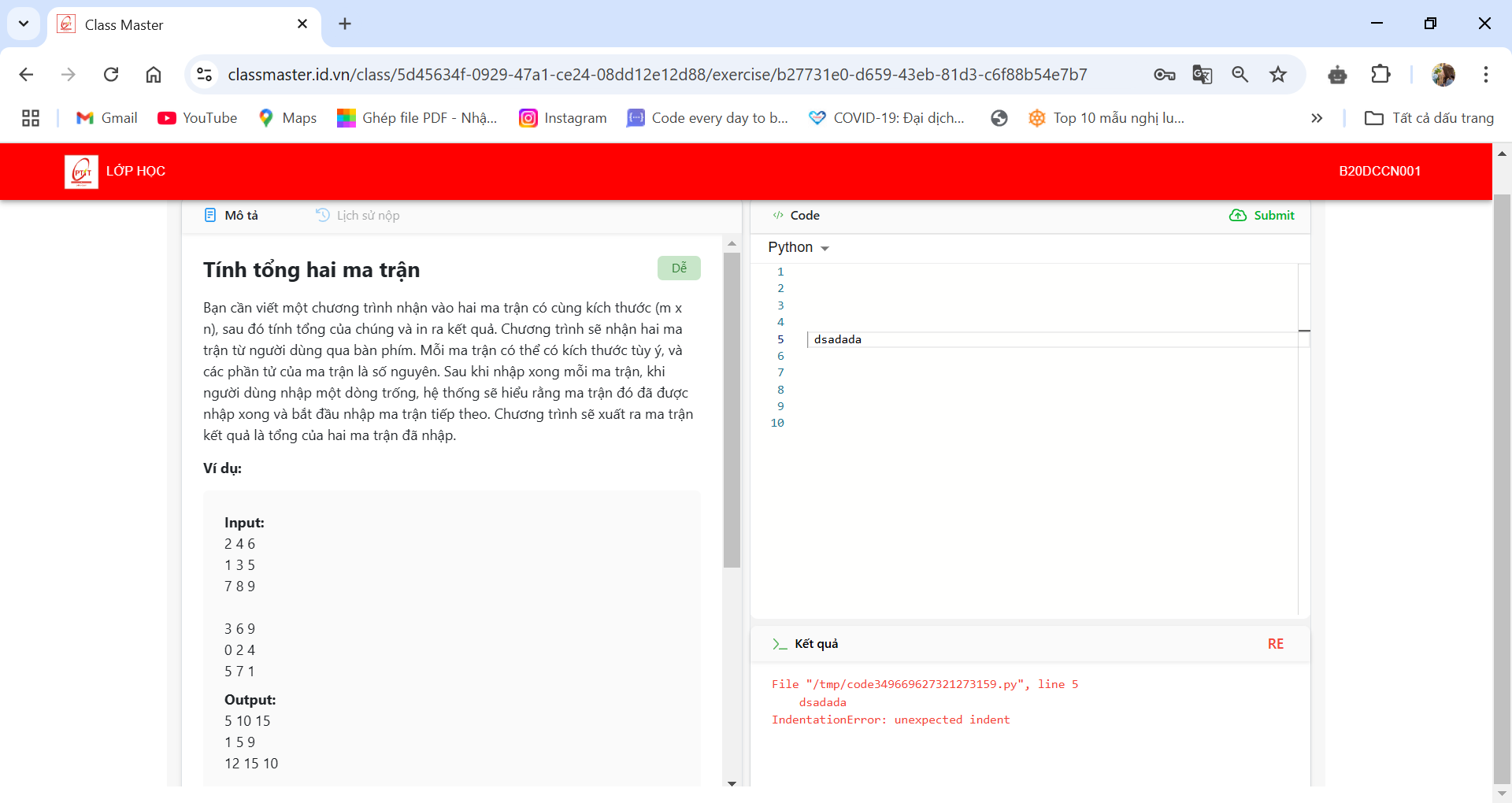
Hình 4.26 Sinh viên xem các chủ đề đã mở



Hình 4.27 Sinh viên xem chi tiết bài tập của chủ đề



Hình 4.28 Sinh viên nộp bài và hiển thị kết quả



## 4.2 Kiểm thử hệ thống

* Kiểm thử hệ thống là một quá trình quan trọng nhằm đảm bảo hệ thống hoạt động đúng đắn về cả chức năng và hiệu suất, nâng cao trải nghiệm của người dùng cuối. Dưới đây là các nội dung kiểm thử được thực hiện đối với hệ thống

### 4.2.1 Kiểm thử phi chức năng

* Kiểm thử phi chức năng được thực hiện sử dụng công cụ Lighthouse được cung cấp sẵn trong trình duyệt Chrome. Môi trường thực hiện kiểm thử là ở trong trình duyệt ẩn danh, không bật extension nào ảnh hưởng tới tốc độ của hệ thống.

A screenshot of a computer

Description automatically generated

Hình 4.29 Kết quả chạy Lighthouse

* Công cụ Lighthouse[2] đánh giá một website thông qua bốn tiêu chí chính là  
  hiệu năng, trợ năng, các thực hành tốt nhất, và tối ưu công cụ tìm kiếm. Dưới đây là các đầu mục mà hệ thống tổng hợp tin tức đã pass khi chạy công cụ Lighthouse
  1. Về các chỉ số hiệu năng của trang web
  2. Về mục trợ năng
  3. Về mục các thực hành tốt nhất
  4. Về mục tối ưu công cụ tìm kiếm (SEO)

### 4.2.2 Kiểm thử test case

Bảng 4.1: Bảng kiểm thử testcase

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Nội dung kiểm thử | Kết quả mong muốn | Kết quả thực tế | Kết quả |
|  |  | GIAO DIỆN CỦA ADMIN |  |  |
| 1 | Đăng nhập với tài khoản đúng | Đăng nhập thành công, chuyển tới giao diện chính của admin | Đăng nhập thành công, chuyển sang trang admin | PASS |
| 2 | Đăng nhập với tài khoản sai | Đăng nhập thất bại, hiển thị thông báo, sai tài khoản, mật khẩu | Đăng nhập thất bại, hiển thị thông báo | PASS |
| 3 | Đăng xuất | Đăng xuất tài khoản thành công, quay về trang login | Đăng xuất tài khoản thành công, quay về trang login | PASS |
| 4 | Xem danh sách môn khi click “Môn học” | Xem danh sách môn học đang dạy thành công | Hiển thị danh sách môn học | PASS |
| 5 | Tạo môn học | Hiển thị form tạo, điền các thông tin và ấn “Tạo” | Hiển thị form tạo để điền các thông tin và ấn “Tạo” | PASS |
| 6 | Chọn tín chỉ khi tạo môn học | Tín chỉ bắt đầu từ 0, chỉ được tăng | Tín chỉ bắt đầu từ 0, chỉ tăng, ko giảm | PASS |
| 7 | Chọn khoa và ngôn ngữ lập trình khi tạo môn học | Hiển thị các option khoa và ngôn ngữ lập trình đang có trên hệ thống để chọn | Hiển thị các option khoa và ngôn ngữ lập trình có trên hệ thống để chọn | PASS |
| 8 | Thêm chủ đề khi tạo môn | Thêm chủ đề khi tạo môn thành công | Thêm chủ đề khi tạo môn thành công | PASS |
| 9 | Xoá chủ đề khi tạo môn | Xoá chủ đề khi tạo môn thành công | Chưa xoá được chủ đề, báo lỗi | FAIL |
| 10 | Xoá môn học | Xoá môn học thành công | Chưa thể xoá môn học | FAIL |
| 11 | Không điền hết các trường khi tạo form | Hiển thị thông báo yêu cầu điền đủ các thông tin | Hiển thị thông báo yêu cầu điền đủ các thông tin | PASS |
| 12 | Cập nhật môn học | Sửa thông tin và Cập nhật môn học thành công | Không thành công | FAIL |
| 13 | Xem chi tiết môn học | Xem chi tiết môn học thành công, hiển thị thêm các option xem lớp và bài tập | Xem chi tiết môn học thành công, hiển thị thêm các option xem lớp và bài tập | PASS |
| 14 | Xem chi tiết lớp học theo môn | Hiển thị chi tiết lớp học theo môn và các option xem danh sách sinh viên và chủ đề đã mở | Hiển thị chi tiết lớp học theo môn và các option xem danh sách sinh viên và chủ đề đã mở | PASS |
| 15 | Xem danh sách sinh viên theo lớp | Hiển thị danh sách sinh viên theo lớp thành công | Hiển thị danh sách sinh viên theo lớp thành công | PASS |
| 16 | Mở thêm chủ đề theo lớp | Mở thêm chủ đề theo lớp thành công | Chưa thành công, không lưu chủ đề vừa thêm | PASS |
| 17 | Xem bài tập chủ đề đã mở | Xem bài tập chủ đề đã mở thành công | Chưa xem được bài tập theo chủ đề | PASS |
| 18 | Xem danh sách lớp khi click “Lớp học” | Xem danh sách lớp học thành công | Hiển thị các danh sách lớp học | PASS |
| 19 | Tạo lớp học | Tạo lớp học thành công | Chưa thành công, hiển thị fail | PASS |
| 20 | Xem chi tiết lớp học | Xem chi tiết lớp học thành công | Xem chi tiết lớp học thành công | PASS |
| 21 | Xem danh sách bài tập | Hiển thị danh sách bài tập trên hệ thống | Hiển thị danh sách bài tập trên hệ thống thành công | PASS |
| 22 | Tạo bài tập | Hiển thị form điền thông tin, báo thành công khi tạo | Hiển thị form điền thông tin, báo thành công khi tạo | PASS |
| 23 | Xem chi tiết bài tập | Hiển thị thông tin chi tiết bài tập | Hiển thị thông tin chi tiết bài tập thành công | PASS |
| 24 | Edit bài tập | Sửa bài tập và hiển thị thông báo sửa thành công khi lưu | Sửa được bài tập và hiển thị thông báo sửa thành công khi lưu | PASS |
| 25 | Xem danh sách sinh viên | Xem danh sách sinh viên thành công | Hiển thị danh sách sinh viên thành công | PASS |
| 26 | Xem chi tiết 1 sinh viên | Hiển thị thông tin chi tiết của sinh viên và option xem danh sách các lớp theo học, lịch sử nộp bài | Hiển thị thông tin chi tiết của sinh viên và option xem danh sách các lớp theo học, lịch sử nộp bài | PASS |
| 27 | Tải file excel sinh viên lên | Tải file excel sinh viên lên thành công | Tải file excel sinh viên lên thành công | PASS |
| 28 | Xem danh sách các lớp theo học của sinh viên | Xem danh sách các lớp theo học của sinh viên thành công | Hiển thị danh sách các lớp theo học của sinh viên thành công | PASS |
| 29 | Xem chi tiết lịch sử nộp bài của sinh viên | Xem chi tiết lịch sử nộp bài của sinh viên thành công | Hiển thị chi tiết lịch sử từng bài nộp của sinh viên | PASS |
| 30 | Xem danh sách giáo viên | Xem danh sách các giáo viên trên hệ thống | Hiển thị danh sách các giáo viên | PASS |
| 31 | Tạo giáo viên | Tạo giáo viên mới | Tạo giáo viên thành công | PASS |
| 32 | Xem danh sách các lớp giáo viên giảng dạy | Hiển thị danh sách các lớp giáo viên đang dạy | Hiển thị danh sách các lớp giáo viên đang dạy | PASS |
| 33 | Thêm tài liệu cho môn học | Thêm tài liệu thành công cho môn học | Thêm tài liệu thành công cho môn học và hiển thị thông báo | PASS |
| 34 | Tải tài liệu môn học | Tải xuống thành công tài liệu môn học | Tải xuống thành công tài liệu môn học | PASS |
|  |  | GIAO DIỆN SINH VIÊN |  |  |
| 1 | Đăng nhập với tài khoản đúng | Đăng nhập thành công, chuyển tới giao diện chính của admin | Đăng nhập thành công, chuyển sang trang admin | PASS |
| 2 | Đăng nhập với tài khoản sai | Đăng nhập thất bại, hiển thị thông báo, sai tài khoản, mật khẩu | Đăng nhập thất bại, hiển thị thông báo | PASS |
| 3 | Đăng xuất | Đăng xuất tài khoản thành công, quay về trang login | Đăng xuất tài khoản thành công, quay về trang login | PASS |
| 4 | Xem danh sách các lớp đang học | Xem danh sách các lớp đang học thành công | Hiển thị danh sách các lớp đang học | PASS |
|  | Xem chi tiết 1 lớp đang học | Xem chi tiết 1 lớp đang học thành công và các option xem danh sách sinh viên, chủ đề đã mở | Xem chi tiết 1 lớp đang học thành công và các option xem danh sách sinh viên, chủ đề đã mở | PASS |
|  | Xem lịch sử nộp bài của sinh viên | Xem lịch sử nộp bài của sinh viên thành công | Hiển thị lịch sử nộp bài của sinh viên khi click xem chi tiết | PASS |
|  | Xem chủ đề đã mở | Xem chủ đề đã mở thành công | Hiển thị các chủ đề đã mở | PASS |
|  | Xem chi tiết bài tập theo chủ đề | Xem chi tiết bài tập theo chủ đề thành công | Hiển thị chi tiết bài tập theo chủ đề | PASS |
|  | Code và nộp bài tập đúng | Code và nộp bài tập thành công, hiển thị kết quả | Code và nộp bài tập thành công, hiển thị kết quả | PASS |
|  | Code và nộp bài tập sai | Hiển thị số test case đúng/ tổng | Hiển thị các test case pass, chưa pass | PASS |
|  | Xem lịch sử nộp bài | Lịch sử hiển thị theo thứ tự mới nhất -> cũ nhất | Chưa hiển thị theo thứ tự mới -> cũ | PASS |

## 4.3 Kết luận Chương IV

Chương IV đã trình bày chi tiết quá trình triển khai và kiểm thử hệ thống quản trị học tập lập trình online. Qua quá trình này, hệ thống đã được đưa vào hoạt động trong môi trường thực tế và được kiểm tra kỹ lưỡng để đảm bảo tính ổn định, hiệu quả và đáp ứng các yêu cầu đặt ra. Chương IV đã chứng minh rằng hệ thống không chỉ được thiết kế tốt trên lý thuyết, mà còn có khả năng hoạt động ổn định và hiệu quả trong môi trường thực tế. Những kinh nghiệm thu được từ quá trình triển khai và kiểm thử sẽ là nền tảng quan trọng để tiếp tục phát triển và cải tiến hệ thống trong tương lai.

# CHƯƠNG V. KẾT LUẬN

## 5.1 Kết quả đạt được

Sau thời gian học tập, nghiên cứu và phát triển hệ thống, đồ án của chúng em về cơ bản đã đạt được các mục tiêu ban đầu đặt ra. Đồ án đã đưa ra được cái giải pháp mới rồi đi sâu vào phân tích, thiết kế và cuối cùng đã tạo ra được một website hoàn chỉnh.

Về hệ thống quản lý lớp học lập trình trực tuyến:

* Hệ thống biên dịch và thực thi hoạt động trơn tru, ổn định, hiệu năng tốt
* Hoàn thiện tốt các chức năng bảo mật cho hệ thống
* API hoạt động, tương tác tốt với frontend

Về mặt giao diện website:

* Màu sắc và thiết kế bắt mắt cho website học tập lập trình
* Bố cục giao diện website cân đối, hài hoà giúp người dùng có thể thao tác dễ dàng

Về chức năng:

* Đáp ứng được phần lớn các chức năng của website quản lý lớphọc tập lập trình
* Các chức năng bổ trợ đã hoạt động tốt

Là người phát triển hệ thống, chúng em thấy rằng việc xây dựng website học tập lập trình đã giúp chúng em hiểu được toàn cảnh một hệ thống lớn hoạt động từ bước phát triển tới bước vận hành. Ngoài ra, chúng em còn được nắm rõ hơn cách Docker và hệ điều hành Linux hoạt động, cách xây dựng website bằng ReactJS, cách sử dụng các framework .Net, Springboot, cách deploy một website…

## 5.2 Hạn chế của hệ thống

Đồ án còn một số hạn chế cần nghiên cứu và cải thiện như:

* Do vấn đề về thời gian nên còn nhiều chức năng có thể bổ sung giúp hệ thống có thể hoàn thiện hơn nữa
* Giao diện chưa responsive: Hệ thống mới có giao diện cho desktop chưa hỗ trợ cho các giao diện khác
* Chưa có hệ thống chấm điểm tự động độ phức tạp: Hệ thống chấm điểm mới chỉ dừng lại ở việc so sánh output của code với output mong đợi trong test case. Việc tích hợp các hệ thống chấm điểm phức tạp hơn, đánh giá nhiều tiêu chí (ví dụ, độ phức tạp của code, tính tối ưu, ...) cần được nghiên cứu thêm.
* Chưa có frontend thân thiện cho người dùng: Frontend của hệ thống tuy đã đáp ứng được yêu cầu về chức năng, nhưng cần được đầu tư thêm về mặt giao diện, trải nghiệm người dùng để trở nên thân thiện và dễ sử dụng hơn.
* Hệ thống biên dịch còn hạn chế về việc hỗ trợ nhiều ngôn ngữ lập trình: Hiện tại mới chỉ hỗ trợ ngôn ngữ C++, Java, Python
* Hệ thống bảo mật có thể cần nâng cấp thêm: Các biện pháp bảo mật hiện tại mới chỉ đáp ứng ở mức cơ bản. Hệ thống cần được nghiên cứu và bổ sung thêm các biện pháp bảo mật nâng cao để chống lại các cuộc tấn công.

## 5.3 Định hướng phát triển hệ thống

* Mở rộng hỗ trợ ngôn ngữ lập trình: Nghiên cứu và tích hợp thêm các ngôn ngữ lập trình mới để đáp ứng nhu cầu đa dạng của người dung
* Nâng cấp các biện pháp bảo mật: Tìm hiểu và áp dụng các kỹ thuật bảo mật nâng cao, chống lại các cuộc tấn công mạng.
* Cải thiện trải nghiệm người dùng: Đầu tư vào thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX) để tạo sự thân thiện và dễ sử dụng cho hệ thống.
* Phát triển riêng hệ thống thực thi code thành một API miễn phí, có thể sử dụng được bởi các nhà phát triển khác và dễ dàng sử dụng trong ứng dụng của họ.

# DANH MỤC TÀI LIỆU THAM KHẢO

**Tài liệu, giáo trình:**

[1] PGS.TS Trần Đình Quế,Giáo trình Phân tích và thiết kế hệ thống thông tin, Học viện Công nghệ Bưu chính Viễn thông, 2014.

[2] PGS.TS Nguyễn Mạnh Hùng, Giáo trình Công nghệ phần mềm, Học viện Công nghệ Bưu chính Viễn thông, 2011.

[3] PGS.TS Trần Đình Quế, PGS.TS Nguyễn Mạnh Hùng, Nhập môn công nghệ phần mềm, Học viện Công nghệ Bưu chính Viễn thông, 2020.

[4] PGS.TS Nguyễn Mạnh Hùng, TS. Đỗ Thị Bích Ngọc Giáo trình Phân tích thiết kế đảm bảo chất lượng phần mềm, Học viện Công nghệ Bưu chính Viễn thông.

**Trang web:**

[1] Microsoft Documentation, <https://docs.microsoft.com/>

[2] Spring Documentation, <https://spring.io/projects/spring-boot>

[3] React Documentation, [https://reactjs.org/docs/](https://www.google.com/url?sa=E&q=https%3A%2F%2Freactjs.org%2Fdocs%2F)

[4] Docker Documentation, <https://docs.docker.com/>

[5] Medium, “How does the computer understands programming languages”, https:// medium.com/the-ui-girl/some-computer-science-questions-7c1056909757

[6] Medium, “System Design — Online Coding / Judge Platform”, https://medium. com/@jnu\_saurav/system-design-online-coding-judge-platform-5b39380818fc