

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI:

.....

.....

Giảng viên hướng dẫn: ThS. TRẦN PHONG NHÃ

Sinh viên thực hiện: CHÂU QUẾ NHƠN

Lớp : CQ.61.CNTT

Khoá : 61

Tp. Hồ Chí Minh, năm 2024

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI (Bold, size 14)
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH (Bold, size 14)
BỘ MÔN CÔNG NGHỆ THÔNG TIN (Bold, size 14)



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
(Bold, size 15)

ĐỀ TÀI:
.....
.....

(Bold, size 16)

Giảng viên hướng dẫn: NGUYỄN VĂN A (size 14, chữ hoa)

Sinh viên thực hiện: TRẦN THỊ B (size 14, chữ in hoa)

Lớp : (size 14, chữ in hoa)

Khoá : (size 14, chữ in hoa)

Tp. Hồ Chí Minh, năm 2024 (size 14)

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

BỘ MÔN: CÔNG NGHỆ THÔNG TIN

-----***-----

Mã sinh viên: 6151071079 Họ tên SV: Châu Quế Nhơn

Khóa: 61..... Lớp: CQ.61.CNTT

- 1. Tên đề tài**
- 2. Mục đích, yêu cầu**
- 3. Nội dung và phạm vi đề tài**
- 4. Công nghệ, công cụ và ngôn ngữ lập trình**
- 5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng**
- 6. Giáo viên và cán bộ hướng dẫn**

Họ tên: Trần Phong Nhã

Đơn vị công tác: Bộ môn Công Nghệ Thông Tin, Trường Đại học Giao Thông Vận Tải phân hiệu Thành phố Hồ Chí Minh

Điện thoại: 0906761014

Email: tpnha@utc2.edu.vn

Ngày tháng 03 năm 2024
Trưởng BM Công nghệ Thông tin

Đã giao nhiệm vụ TKTN
Giáo viên hướng dẫn

ThS. Trần Phong Nhã

ThS. Trần Phong Nhã

Đã nhận nhiệm vụ TKTN

Sinh viên: Châu Quế Nhơn

Điện thoại: 0848611127

Ký tên:

Email: 6151071079@st.utc2.edu.vn

Size 13

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày tháng năm
Giáo viên hướng dẫn

ThS. Trần Phong Nhã

MỤC LỤC

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP	3
DANH MỤC CHỮ VIẾT TẮT	v
DANH MỤC HÌNH ẢNH	vi
DANH MỤC BẢNG BIỂU	vii
TỔNG QUAN ĐỀ TÀI	1
1. Giới thiệu về bối cảnh và cần thiết của đề tài.....	1
2. Mục tiêu của đề tài.	2
3. Phạm vi và đối tượng nghiên cứu.....	3
4. Phương pháp nghiên cứu	4
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT.....	5
1.1 Tổng quan về Laravel 11	5
1.1.1 Giới thiệu	5
1.1.2 Vòng đời và kiến trúc của Laravel	6
1.1.3 Tính năng nổi bật.....	7
1.1.4 Ưu và nhược điểm của Laravel	13
1.2 NextJS 14.....	14
1.2.1 Giới thiệu	14
1.2.2 Client Component.....	16
1.2.3 Ưu điểm, nhược điểm	16
1.3 Cơ sở dữ liệu.....	17
1.3.1 MySQL	17
1.3.2 Redis	18
1.4 Python và khoa học dữ liệu	19
1.4.1 Học máy trong Python.....	20
1.4.2 Thư viện Scikit learn	21
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ.....	23
2.1 Thiết kế hệ thống	23
2.1.1 Sơ đồ Usecase.....	23
2.1.2 Sơ đồ Class	26
2.1.3 Sơ đồ tuần tự.....	27
2.2 Thiết kế cơ sở dữ liệu	31
2.2.1 Mô tả bài toán.....	31

2.2.2 Sơ đồ ER.....	33
2.2.3 Bảng thực thể cơ sở dữ liệu	33
CHƯƠNG 3. XÂY DỰNG MÔ HÌNH DỰ ĐOÁN	38
3.1 Thu thập dữ liệu.....	38
3.1.1 Xác định các yếu tố ảnh hưởng đến điểm thi	38
3.1.2 Phân tích mối quan hệ giữa các yếu tố ảnh hưởng đến kết quả dự đoán	40
3.2 Tiền xử lý dữ liệu	43
3.3 Ứng dụng thuật toán Random Forest.....	47
3.4 Xây dựng và đánh giá mô hình.....	49
3.4.1 Xây dựng và huấn luyện mô hình.....	49
3.4.2 Đánh giá hiệu suất của mô hình	50
CHƯƠNG 4. XÂY DỰNG CHƯƠNG TRÌNH.....	52
4.1 Thết kế giao diện	52
4.2.1 Trang người dùng	52
4.2.2 Trang Admin.....	52
4.2 Tích hợp tính năng dự đoán.....	52
KẾT LUẬN & KIẾN NGHỊ.....	53
1. Kết quả đạt được.....	53
2. Hạn chế	53
3. Hướng phát triển.....	53
TÀI LIỆU THAM KHẢO	54

DANH MỤC CHỮ VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa	Ghi chú
1	THPT QG	Kỳ thi trung học phổ thông quốc gia	
2	BE	(Backend): Xử lý logic nghiệp vụ, truy xuất cơ sở dữ liệu và cung cấp API cho frontend.	
3	FE	(Frontend): Xử lý phía client	
4	ORM	(Object-Relational Mapping): Ánh xạ csdl thành các Object	
5	PK	Primary Key	
6	FK	Foreign Key	
7	SSR	Server-Side Rendering	
8	CSR	Client-Side Rendering	
9	ML	Machine Learning	
10			

DANH MỤC HÌNH ẢNH

Hình 1. 1 Sơ đồ vòng đời của Laravel.....	6
Hình 2. 1 Sơ đồ Usecase tổng quát.....	23
Hình 2. 2 Sơ đồ phân rã Usecase Thảo luận.....	23
Hình 2. 3 Sơ đồ phân rã Usecase Tham gia thi đấu	24
Hình 2. 4 Sơ đồ phân rã Usecase Quản lý thông tin cá nhân	24
Hình 2. 5 Sơ đồ phân rã Usecase Quản lý câu hỏi	25
Hình 2. 6 Sơ đồ phân rã Usecase Quản lý bài thi.....	25
Hình 2. 7 Sơ đồ phân rã Usecase Quản lý phòng thi.....	25
Hình 2. 8 Sơ đồ phân rã Usecase Quản lý người dùng.....	26
Hình 2. 9 Sơ đồ Class	26
Hình 2. 10 Sơ đồ tuần tự thi thử.....	27
Hình 2. 11 Sơ đồ tuần tự đấu trường	28
Hình 2. 12 Sơ đồ tuần tự Thêm mới bản ghi	29
Hình 2. 13 Sơ đồ tuần tự Cập nhật bản ghi	30
Hình 2. 14 Sơ đồ tuần tự xóa bản ghi.....	31
Hình 2. 15 Sơ đồ ER.....	33
Hình 3. 1 Kiểm tra dữ liệu thiếu.....	44
Hình 3. 2 Mã hóa các biến phân loại dạng text sang number.....	45
Hình 3. 3 Chuẩn hóa và loại bỏ ngoại lệ	45
Hình 3. 4 Tách đặc trưng và biến mục tiêu.....	46
Hình 3. 5 Chia tập dữ liệu.....	46
Hình 3. 6 Huấn luyện mô hình với Random Forest trong Sklearn.....	49
Hình 3. 7 Tính toán giá trị để đánh giá mô hình.....	50

DANH MỤC BẢNG BIỂU

Bảng 2. 1 Bảng thực thể User.....	33
Bảng 2. 2 Bảng thực thể Subject	34
Bảng 2. 3 Bảng thực thể Chapter.....	34
Bảng 2. 4 Bảng thực thể Practice	34
Bảng 2. 5 Bảng thực thể Lesson	34
Bảng 2. 6 Bảng thực thể Question.....	35
Bảng 2. 7 Bảng thực thể Exam.....	35
Bảng 2. 8 Bảng thực thể Topic	35
Bảng 2. 9 Bảng thực thể TopicComment	36
Bảng 2. 10 Bảng thực thể Arena	36
Bảng 2. 11 Bảng thực thể UserTarget	36
Bảng 2. 12 Bảng thực thể History	37
Bảng 2. 13 Bảng thực thể DayStatistics	37
Bảng 3. 1 Bảng các yếu tố cá nhân quyết định điểm thi	38
Bảng 3. 2 Bảng các yếu tố web quyết định điểm thi	39
Bảng 3. 3 Ví dụ về tập dữ liệu.....	47

TỔNG QUAN ĐỀ TÀI

1. Giới thiệu về bối cảnh và cần thiết của đề tài.

Kỳ thi trung học phổ thông quốc gia có thể nói là cột mốc quan trọng, có tính quyết định cao nhất trong cuộc đời học sinh. Nó là kết quả của cả một quãng đường 12 năm gian nan, vất vả và đầy nỗ lực của học sinh trong hệ thống giáo dục phổ thông. Với mỗi học sinh, kỳ thi này không chỉ đánh dấu sự hoàn thành của giai đoạn trung học mà còn là cơ hội để họ chứng tỏ khả năng và kiến thức mà mình đã tích lũy được. Tầm quan trọng của kỳ thi trung học phổ thông quốc gia nằm ở việc nó không chỉ đánh giá hiệu suất học tập của học sinh mà còn định hình tương lai của họ. Kết quả của kỳ thi này có thể ảnh hưởng trực tiếp đến cơ hội tiếp cận giáo dục đại học và sự lựa chọn nghề nghiệp sau này của học sinh. Điều này làm cho áp lực và trách nhiệm của kỳ thi trở nên nặng nề và quan trọng hơn bao giờ hết. Ngoài ra, kỳ thi trung học phổ thông quốc gia cũng là công cụ đánh giá chất lượng của hệ thống giáo dục quốc gia. Kết quả của kỳ thi này phản ánh sự hiệu quả của quá trình dạy và học ở cấp trung học, từ chất lượng giáo viên đến nội dung chương trình học. Đồng thời, nó cũng là một phần trong việc đánh giá và cải thiện chất lượng giáo dục toàn diện của đất nước.

Việc ôn luyện có phương pháp và hiệu quả đóng vai trò quan trọng trong việc chuẩn bị cho kỳ thi trung học phổ thông quốc gia. Phương pháp ôn luyện hiệu quả giúp học sinh tối ưu hóa thời gian và năng lượng để nắm vững kiến thức cũng như phát triển kỹ năng cần thiết cho kỳ thi. Việc áp dụng phương pháp ôn luyện có hệ thống giúp học sinh xác định các mục tiêu cụ thể và lập kế hoạch hợp lý cho quá trình ôn tập. Bằng cách chia nhỏ kiến thức thành các đơn vị nhỏ hơn và ưu tiên làm việc trên những phần cần thiết nhất, học sinh có thể tập trung và hiệu quả hóa quá trình học tập. Đồng thời, việc sử dụng các phương tiện và tài liệu ôn luyện chất lượng cũng đóng vai trò quan trọng. Các sách giáo khoa, tài liệu ôn thi, bài tập mẫu và đề thi thử không chỉ giúp học sinh làm quen với định dạng của kỳ thi mà còn giúp họ củng cố kiến thức và rèn luyện kỹ năng làm bài thi một cách tự tin. Việc tìm kiếm sự hỗ trợ từ giáo viên, gia đình và bạn bè cũng là một phần quan trọng của quá trình ôn luyện. Sự hỗ trợ và động viên từ người thân và cộng đồng giúp học sinh vượt qua những khó khăn trong quá trình học tập và tạo ra một môi trường tích cực để phát triển.

Trong thời đại phát triển của công nghệ thông tin hiện nay, việc sử dụng các nền tảng công nghệ như web/app để học tập, ôn luyện mang lại nhiều hiệu quả cho mỗi người. Sự tiện lợi và linh hoạt, người dùng có thể tự do học, luyện thi mọi lúc mọi nơi. Ngoài ra, việc truy cập vào nguồn tài nguyên phong phú từ internet giúp họ tiếp cận các kiến thức mới một cách dễ dàng và đa dạng hơn. Việc tham gia vào các hoạt động tương tác và hợp tác qua các nền tảng giáo dục từ xa cũng giúp họ rèn luyện kỹ năng giao tiếp và làm việc nhóm. Sử dụng công nghệ thông tin cũng giúp họ nhận được phản hồi và đánh giá nhanh chóng, từ đó điều chỉnh quá trình học tập một cách hiệu quả hơn. Cuối cùng, việc khuyến khích sự sáng tạo và khám phá qua các ứng dụng và công cụ giúp họ phát triển tư duy sáng tạo và thúc đẩy quá trình học tập. Từ đó, việc xây dựng trang web hỗ trợ học tập và luyện thi đại học có thể hỗ trợ người dùng dễ dàng tổng hợp kiến thức, thử sức với các dạng đề thi đại học là vô cùng cần thiết. Trang web này sẽ có thể giúp ích rất nhiều cho những sĩ tử sắp bước vào kì thi quan trọng nhất hoặc đối với những học sinh muốn ôn luyện sớm hơn để chuẩn bị kỹ cho kì thi này.

2. Mục tiêu của đề tài.

Mục tiêu dự kiến của đề tài này sẽ là xây dựng được một trang web với các yêu cầu:

- Về giao diện: cấu trúc trang web chuẩn SEO, giao diện hiện đại đẹp mắt dễ sử dụng. Hỗ trợ responsive trên nhiều thiết bị khác nhau.

- Về chức năng:

- + Xác thực: đăng nhập/đăng ký tài khoản.

- + Phân quyền: Sẽ có 2 loại tài khoản: học sinh và giáo viên. Ngoài ra còn có vai trò quản trị viên/người kiểm duyệt.

- + Tính năng hỏi – đáp: người dùng có thể tạo các chủ đề thảo luận, đặt các câu hỏi để mọi người có thể giúp đỡ, trao đổi, góp ý.

- + Cung cấp lý thuyết, bài giải SGK: Lý thuyết và các bài tập (có đáp án) theo chương trình học chuẩn.

- + Luyện tập giải đề: Học sinh có thể luyện tập giải các loại đề thi mẫu dưới hình thức trắc nghiệm trong thời gian quy định. Chức năng giải đề theo lộ trình học chuẩn để nâng cao hiệu quả ôn luyện.

+ Tính năng đầu trường: người dùng có thể tạo phòng để những người tham gia khác có thể cùng tham gia thi cùng lúc với nhau (realtime).

+ Thống kê tình hình ôn luyện của từng thí sinh, giúp thí sinh có thể nắm bắt được quá trình ôn luyện của bản thân.

Ngoài ra, mục tiêu quan trọng nhất là có thể xây dựng được một mô hình học máy đơn giản để dự đoán điểm số, hiệu suất học tập, hoặc khả năng làm bài thi của người dùng dựa trên dữ liệu lịch sử, đưa ra đề xuất những đề thi có thể cải thiện điểm yếu cho thí sinh, mang lại hiệu quả ôn luyện tốt nhất cho thí sinh.

3. Phạm vi và đối tượng nghiên cứu.

*** Phạm vi nghiên cứu:**

+ Yêu cầu của các sĩ tử đang ôn luyện: Đề tài này sẽ tập trung vào nghiên cứu và hiểu rõ những yêu cầu cần thiết của các sĩ tử đang ôn luyện để chuẩn bị cho kỳ thi trung học phổ thông quốc gia. Qua đó, trang web sẽ được thiết kế và phát triển để đáp ứng những yêu cầu này một cách hiệu quả nhất, bao gồm việc cung cấp các tài liệu ôn tập, bài kiểm tra mẫu, và đề thi thử.

+ Nghiên cứu về kỹ thuật lập trình: Đồng thời, nghiên cứu cũng sẽ tập trung vào các vấn đề kỹ thuật lập trình liên quan đến việc phát triển trang web. Điều này bao gồm việc tìm hiểu và áp dụng các công nghệ lập trình web như PHP Laravel và Next.js để xây dựng một trang web mạnh mẽ, linh hoạt và dễ bảo trì. Các nguyên tắc thiết kế và triển khai cũng sẽ được tìm hiểu để đảm bảo trải nghiệm người dùng tốt nhất.

+ Nghiên cứu về mô hình học máy có giám sát: Một phần quan trọng của đề tài sẽ là nghiên cứu về mô hình học máy có giám sát để xây dựng, huấn luyện và triển khai một hệ thống có khả năng dự đoán điểm số và hiệu suất học tập của các sĩ tử. Điều này đòi hỏi việc thu thập và tiền xử lý dữ liệu mẫu một cách kỹ lưỡng, sau đó áp dụng các mô hình học máy như hồi quy tuyến tính hoặc cây quyết định để tạo ra dự đoán chính xác và hữu ích cho người dùng.

*** Đối tượng nghiên cứu:**

Đối tượng nghiên cứu chính của trang web luyện thi đại học chắc chắn sẽ tập trung vào các sĩ tử đang chuẩn bị thi trung học phổ thông quốc gia hoặc những học sinh đang học cấp trung học phổ thông muốn luyện thi đại học sớm. Bằng cách nghiên cứu các cộng

đồng học sinh luyện thi đại học, các trang web luyện thi khác trên internet sẽ mang lại một cái nhìn toàn diện về nhu cầu và mong muốn của đối tượng này. Thông qua việc thu thập dữ liệu từ người dùng, bao gồm thông tin cá nhân, kết quả kiểm tra trước đó, hoạt động học tập và một số yếu tố khác, trang web có thể tạo ra những giải pháp tùy chỉnh và hữu ích nhằm nâng cao hiệu suất học tập của học sinh.

4. Phương pháp nghiên cứu

Các phương pháp nghiên cứu được sử dụng để thu thập và xử lý dữ liệu là:

- Khảo sát các yêu cầu của thí sinh chuẩn bị thi THPTQG: Khảo sát, thu thập thông tin về các yêu cầu, nhu cầu và mong muốn của thí sinh đang chuẩn bị thi Trung học Phổ thông Quốc gia (THPTQG). Điều này giúp định hình các tính năng và tài nguyên cần thiết trên trang web luyện thi.

- Tham khảo các trang web luyện thi khác: Tìm kiếm và tham khảo các trang web luyện thi khác trên internet để thu thập thông tin về các phương pháp và tài liệu ôn thi, cũng như để hiểu thêm về các nhu cầu và mong muốn của người dùng trong lĩnh vực này.

- Khảo sát các cộng đồng học sinh THPT: Tiến hành nghiên cứu và khảo sát các cộng đồng học sinh THPT trên các diễn đàn, mạng xã hội hoặc các nhóm chuyên ngành để hiểu rõ hơn về các thách thức và nhu cầu của họ trong quá trình luyện thi.

- Thu thập dữ liệu về các câu hỏi, đề thi chuẩn theo chương trình của Bộ Giáo Dục: Thu thập dữ liệu từ các nguồn đáng tin cậy về các câu hỏi và đề thi chuẩn theo chương trình của Bộ Giáo Dục, để xây dựng cơ sở dữ liệu đa dạng và chất lượng để phục vụ cho mục đích ôn thi.

- Nghiên cứu về các hàm trong thư viện sklearn của Python để xử lý dữ liệu cho mô hình máy học: Tiến hành nghiên cứu và tìm hiểu về các hàm và phương pháp trong thư viện scikit-learn của Python để tiền xử lý và chuẩn bị dữ liệu cho việc huấn luyện các mô hình máy học. Điều này bao gồm các phương pháp chuẩn hóa, mã hóa dữ liệu, và xử lý dữ liệu thiếu.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1 Tổng quan về Laravel 11

1.1.1 Giới thiệu

Laravel là gì?

Laravel là một framework phát triển ứng dụng web được viết bằng ngôn ngữ lập trình PHP. Nó cung cấp một cách tiếp cận đơn giản và linh hoạt cho việc xây dựng các ứng dụng web phức tạp. Laravel đi kèm với nhiều tính năng mạnh mẽ như hệ thống routing, ORM (Object-Relational Mapping), middleware, bảo mật, quản lý cơ sở dữ liệu và nhiều tính năng khác giúp người phát triển tạo ra các ứng dụng web chất lượng cao và dễ bảo trì. Điểm mạnh của Laravel nằm ở việc cung cấp cú pháp rõ ràng và cấu trúc dễ hiểu, làm cho việc phát triển ứng dụng trở nên nhanh chóng và hiệu quả.

Tại sao phải dùng Laravel?

Laravel là một framework phát triển web PHP mạnh mẽ và phổ biến, được thiết kế để mang lại trải nghiệm phát triển dễ dàng và hiệu quả. Một số lý do mà nhiều nhà phát triển lựa chọn Laravel là do tính nhanh chóng và thuận tiện trong quá trình phát triển ứng dụng. Với Artisan Console, một công cụ dòng lệnh mạnh mẽ, nhà phát triển có thể tự động hóa nhiều công việc lặp lại, giảm thời gian và công sức đặc biệt là trong quá trình lặp lại. Điều này giúp tối ưu hóa quá trình phát triển và giảm bớt sự lặp lại không cần thiết.

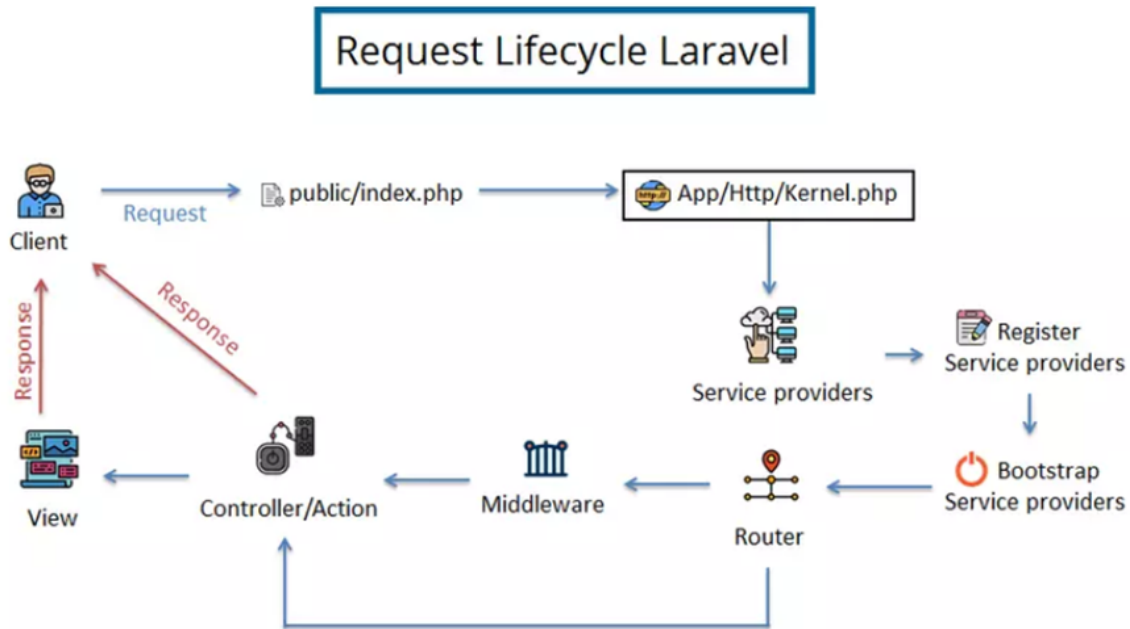
Mặc dù PHP hiện nay đã dần trở nên lạc hậu và chậm cập nhật so với những công nghệ web phía server khác, nhưng với vị thế đã được xây dựng từ lâu, nó vẫn chiếm một phần lớn trong thị phần web hiện nay trên toàn thế giới. Trong cộng đồng lập trình viên PHP, có câu đùa rằng "PHP is dead, but Laravel forever alive". Điều này có phần đúng trong khi PHP cập nhật quá chậm, thì tốc độ update và phát triển của Laravel vẫn có thể khiến các lập trình viên mới vào nghề “không kịp thích nghi”. Laravel hiện nay cung cấp hầu như mọi tính năng mà các framework hiện đại yêu cầu, như xử lý dữ liệu lớn, bảo mật, khả năng mở rộng và tích hợp tương thích cao với các dịch vụ bên thứ ba. Laravel sẽ release version mới vào quý 1 hàng năm, tính đến hiện tại đã là phiên bản Laravel 11.

Laravel 11 chạy trên phiên bản PHP mới (> 8.2) đã như một cuộc cải cách lớn của Laravel Team để bắt kịp cuộc đua công nghệ với những Framework JS mạnh mẽ hiện nay như NestJS, NextJs, ... Trong phiên bản mới này, cấu trúc thư mục đã được rút gọn, tối ưu

hơn so với những phiên bản trước đó. Ngoài ra, sự ra đời của Reverb đã giúp Laravel có một bước tiến lớn hơn khi lần đầu tiên, PHP có thể tự dùng để xây dựng một ứng dụng realtime mà không cần dùng đến bên thứ 3.

1.1.2 Vòng đời và kiến trúc của Laravel

1.1.2.1 Vòng đời hoạt động của Laravel



Hình 1. 1 Sơ đồ vòng đời của Laravel

Quá trình vòng đời hoạt động của Laravel bắt đầu khi một yêu cầu (request) được gửi từ phía client. Đối tượng nhắm đến của mọi yêu cầu từ client chính là file public/index.php, một tập tin chính của ứng dụng. Màu xanh của mũi tên biểu thị hành động di chuyển từ client đến đích đến này.

Mặc dù số lượng mã nguồn trong file index.php không nhiều, nhưng nó đóng vai trò quan trọng như là khởi nguyên cho toàn bộ framework Laravel. File này thực hiện các công việc cơ bản như bao gồm file autoload.php để load các class cần thiết, sau đó khởi tạo và chạy ứng dụng Laravel. Điều này bao gồm việc xử lý middleware, định tuyến (routing), và gọi controllers tương ứng với request.

Từ file index.php, quá trình xử lý được chuyển tiếp qua các giai đoạn tiếp theo của vòng đời hoạt động của Laravel, như chạy middleware, xử lý route, thực hiện controllers,

và cuối cùng là gửi response về phía client. Dù đơn giản nhưng file index.php tạo nên một cơ sở hạ tầng vững chắc, làm nền tảng cho tính linh hoạt và hiệu suất của toàn bộ framework Laravel.

1.1.2.2 Kiến trúc của Laravel

Laravel sử dụng một kiến trúc mô hình MVC (Model-View-Controller) để tổ chức mã nguồn của ứng dụng web một cách có tổ chức và linh hoạt. Kiến trúc này giúp tách biệt logic dữ liệu, giao diện người dùng, và quản lý điều khiển, tạo nên một cấu trúc chia làm ba thành phần chính.

Model trong Laravel đóng vai trò quan trọng trong việc xử lý dữ liệu và logic nghiệp vụ của ứng dụng. Được thiết kế để tương ứng với các bảng trong cơ sở dữ liệu, Model chịu trách nhiệm cho việc truy cập và thay đổi dữ liệu một cách hiệu quả. Sự tích hợp của Eloquent ORM giúp việc thực hiện các truy vấn dữ liệu trở nên thuận tiện và đơn giản. Model tách biệt logic dữ liệu, tạo nên một mô-đun dễ bảo trì và mở rộng.

View trong Laravel là nơi hiển thị thông tin từ Models và làm giao diện người dùng tương tác. Blade Template Engine, một công cụ quan trọng của View, giúp đơn giản hóa quá trình hiển thị dữ liệu và xây dựng giao diện trực quan. Việc tách biệt logic hiển thị giúp duy trì sự sạch sẽ của mã nguồn và thuận tiện trong việc tái sử dụng các thành phần giao diện, đồng thời mang lại trải nghiệm người dùng mượt mà và thân thiện.

Controller là bộ điều khiển chính của ứng dụng trong Laravel, nhận và xử lý yêu cầu từ người dùng thông qua routes. Controllers tương tác với Models để truy xuất và cập nhật dữ liệu, sau đó chọn View phù hợp để hiển thị kết quả. Tích hợp Middleware giúp xử lý các yêu cầu trước khi chúng đến tay Controllers, tăng tính linh hoạt và kiểm soát. Controller tách biệt logic điều khiển, tạo nên một kiến trúc dễ bảo trì và mở rộng ứng dụng.

1.1.3 Tính năng nổi bật

1.1.3.1 Eloquent ORM

Laravel cung cấp một bộ ánh xạ cơ sở dữ liệu hiện đại và vô cùng mạnh mẽ, được gọi là Eloquent ORM. Eloquent ORM (Object-Relational Mapping) là một trong những tính năng nổi bật nhất, cung cấp một cách thức đơn giản và thanh lịch để tương tác với cơ sở dữ liệu. Eloquent ORM cho phép ánh xạ các bảng trong cơ sở dữ liệu thành các đối tượng PHP (model), và các bản ghi (record) trong bảng thành các thể hiện (instance) của model đó.

Điều này giúp cho việc thao tác với cơ sở dữ liệu trở nên trực quan và dễ dàng hơn, chỉ cần có hiểu biết về bản chất của truy vấn cơ sở dữ liệu thì có thể sử dụng ORM một cách dễ dàng, nhanh chóng, đồng thời tận dụng được các tính năng của lập trình hướng đối tượng trong PHP thông qua các class model.

Các tính năng chính được hỗ trợ mạnh mẽ bởi Eloquent ORM có thể kể đến như sau:

- Xem, thêm, sửa và xóa các bảng ghi (CRUD): Eloquent cung cấp các phương thức đơn giản và nhất quán để thực hiện các thao tác cơ bản với cơ sở dữ liệu, giúp giảm thiểu lượng code boilerplate và tăng tính dễ đọc của mã nguồn.
- Mối quan hệ (Relationships): Laravel là một trong những framework hỗ trợ relationship một cách đơn giản, dễ hiểu và dễ sử dụng nhất. Eloquent hỗ trợ định nghĩa và quản lý các mối quan hệ giữa các bảng (ví dụ: một-nhiều, nhiều-nhiều) một cách dễ dàng. Nhờ đó, có thể truy xuất và thao tác với dữ liệu liên quan giữa các bảng một cách thuận tiện.
- Kiểu collections: Kết quả truy vấn từ cơ sở dữ liệu được trả về dưới dạng các tập hợp Eloquent (collections). Collections cung cấp nhiều phương thức tiện lợi để lọc, sắp xếp, biến đổi và thao tác với dữ liệu một cách linh hoạt.
- Xử lý sự kiện (Events): Eloquent cho phép "lắng nghe" và phản ứng với các sự kiện xảy ra trong quá trình thao tác với model, chẳng hạn như khi một model được tạo, cập nhật hoặc xóa. Tính năng này giúp thực hiện các tác vụ phụ trợ một cách dễ dàng và tập trung.
- Bảo mật: Eloquent tự động bảo vệ ứng dụng khỏi các cuộc tấn công SQL Injection bằng cách sử dụng prepared statements. Prepared statements là một kỹ thuật giúp ngăn chặn việc chèn các đoạn mã độc vào câu truy vấn SQL, đảm bảo tính an toàn cho dữ liệu.

```
use App\Models\User;

// Lấy tất cả người dùng

$users = User::all();

// Tìm người dùng theo ID
```

```

$user = User::find(1);

// Tạo người dùng mới

$newUser = User::create([

    'name' => 'John Doe',

    'email' => 'johndoe@example.com',

    'password' => Hash::make('password'),

]);

// Cập nhật thông tin người dùng

$user->name = 'Jane Doe';

$user->save();

// Xóa người dùng

$user->delete();

```

* API Resource

Ngoài ra, Eloquent ORM còn hỗ trợ phát triển web service với chuẩn Restful API một cách dễ dàng. API Resource là một tính năng của Laravel giúp cho việc xây dựng API trở nên dễ dàng và nhất quán hơn. API Resource cho phép chuyển đổi models và collections thành các cấu trúc JSON theo một cách định nghĩa trước, giúp cho việc trả về dữ liệu từ API trở nên dễ quản lý và bảo trì hơn.

Các tính năng chính của API Resource:

- Tùy chỉnh định dạng dữ liệu: API Resource cho phép bạn kiểm soát hoàn toàn cấu trúc và định dạng của dữ liệu JSON được trả về từ API. Bạn có thể chọn những trường dữ liệu cần hiển thị, ẩn đi những trường nhạy cảm, và thậm chí định nghĩa các mối quan hệ giữa các resource.
- Tạo resource collections: API Resource cung cấp một cách đơn giản để chuyển đổi các collections của model thành các mảng JSON, giúp cho việc trả về nhiều bản ghi cùng lúc trở nên dễ dàng hơn.
- Phân trang (Pagination): API Resource tích hợp sẵn tính năng phân trang, giúp bạn dễ dàng chia dữ liệu thành nhiều trang và trả về thông tin phân trang cho client.

- Hỗ trợ các mối quan hệ: API Resource cho phép bạn nhúng các resource khác vào trong resource hiện tại, giúp cho việc trả về dữ liệu liên quan giữa các model trở nên thuận tiện hơn.

* Xử lý dữ liệu lớn

Một tính năng ưu Việt nữa, Laravel còn cung cấp một số phương thức và công cụ để xử lý dữ liệu lớn (big data), giúp cho quá trình làm việc với các tập dữ liệu có kích thước lớn trở nên dễ dàng và hiệu quả hơn. Các phương thức xử lý dữ liệu lớn được hỗ trợ như: Phương thức chunk cho phép chia nhỏ một truy vấn lớn thành nhiều phần nhỏ hơn để xử lý, tránh tình trạng tràn bộ nhớ khi làm việc với dữ liệu lớn. Phương thức cursor cũng cho phép lặp qua một tập kết quả lớn mà không cần tải toàn bộ dữ liệu vào bộ nhớ. Lazy Collections là một tính năng mới trong Laravel, cho phép làm việc với các tập dữ liệu lớn một cách hiệu quả hơn bằng cách chỉ tải dữ liệu khi cần thiết.

Ngoài ra, các công cụ xử lý dữ liệu lớn khác như: queue cho phép đưa các tác vụ xử lý dữ liệu nặng vào hàng đợi để xử lý nền, giúp giảm tải cho máy chủ web và tăng tốc độ xử lý. Laravel Jobs cung cấp một cách để định nghĩa các tác vụ xử lý dữ liệu một cách có cấu trúc và dễ quản lý. Bạn có thể đưa các jobs vào queue để xử lý nền. Events: Laravel Events cho phép bạn tạo ra các sự kiện khi có dữ liệu mới được thêm vào hoặc cập nhật, từ đó kích hoạt các tác vụ xử lý dữ liệu khác.

1.1.3.2 Broadcasting

* Giới thiệu về Broadcasting

Broadcasting là một tính năng mạnh mẽ trong Laravel cho phép truyền tải các sự kiện trong thời gian thực từ phía máy chủ (backend) đến phía máy khách (frontend). Điều này giúp tạo ra các ứng dụng web có khả năng cập nhật thông tin tức thời mà không cần phải tải lại trang hoặc gửi yêu cầu liên tục từ phía client.

Cơ chế hoạt động: Khi có một sự kiện xảy ra trong ứng dụng (ví dụ: người dùng đăng bài mới, có thông báo mới,...), một đối tượng Event sẽ được kích hoạt. Sự kiện này sẽ được gửi đến một hoặc nhiều kênh (channel) được định nghĩa trước. Các kênh này có thể là công khai (public) hoặc riêng tư (private), tùy thuộc vào yêu cầu của ứng dụng. Laravel sử dụng các trình điều khiển để gửi sự kiện đến các kênh. Các trình điều khiển phổ biến bao gồm Pusher, Redis, và Log. Phía máy khách sẽ lắng nghe các sự kiện trên các kênh mà họ đã

đăng ký. Khi có sự kiện mới, trình duyệt sẽ nhận được thông báo và cập nhật giao diện tương ứng.

Broadcasting được sử dụng rộng rãi trong việc xây dựng các tính năng realtime cho ứng dụng web, chẳng hạn như:

- Thông báo: Gửi thông báo tức thời đến người dùng khi có sự kiện mới (ví dụ: tin nhắn mới, bình luận mới, yêu cầu kết bạn,...).
- Trò chuyện: Xây dựng các ứng dụng trò chuyện trực tuyến, nơi người dùng có thể gửi và nhận tin nhắn tức thời.
- Theo dõi trạng thái: Cập nhật trạng thái của một đối tượng trong thời gian thực (ví dụ: trạng thái đơn hàng, tiến độ công việc,...).
- Bảng tin trực tiếp: Hiển thị các cập nhật mới nhất về một chủ đề (ví dụ: tỉ số trận đấu, giá cổ phiếu,...) mà không cần phải tải lại trang.

* Reverb trong Laravel 11

Reverb là một tính năng mới đáng chú ý được giới thiệu trong Laravel 11, cho phép xây dựng các ứng dụng realtime hoàn toàn bằng PHP mà không cần phụ thuộc vào các công nghệ bên thứ ba như Pusher hay Ably. Reverb hoạt động dựa trên giao thức WebSocket và Server-Sent Events (SSE), mang đến khả năng cập nhật dữ liệu tức thời cho ứng dụng web mà không cần yêu cầu từ phía client. Một số điểm mà nhà Laravel đã chính thức giới thiệu về Reverb với những ưu điểm như sau:

- Đơn giản và dễ sử dụng: Reverb được tích hợp trực tiếp vào Laravel, giúp việc cài đặt và sử dụng trở nên dễ dàng hơn so với việc tích hợp các dịch vụ bên ngoài.
- Hiệu suất cao: Reverb tận dụng sức mạnh của Swoole, một extension PHP hiệu suất cao, để xử lý các kết nối WebSocket và SSE, đảm bảo khả năng mở rộng và đáp ứng tốt với lượng lớn người dùng.
- Linh hoạt: Reverb cung cấp các API đơn giản và linh hoạt để gửi và nhận sự kiện, cho phép tùy chỉnh theo nhu cầu của ứng dụng.
- Bảo mật: Reverb hỗ trợ xác thực và ủy quyền người dùng, đảm bảo tính bảo mật cho các kết nối và dữ liệu truyền đi.
- Giảm sự phụ thuộc: Việc sử dụng Reverb giúp giảm sự phụ thuộc vào các dịch vụ bên thứ ba, giúp tiết kiệm chi phí và tăng tính kiểm soát đối với ứng dụng.

Với Reverb, việc xây dựng các ứng dụng realtime với Laravel trở nên dễ dàng và hiệu quả hơn bao giờ hết. Tính năng này hứa hẹn sẽ mở ra nhiều cơ hội mới cho các nhà phát triển Laravel trong việc tạo ra các ứng dụng web hiện đại và hấp dẫn. Tuy nhiên, với trình độ còn hạn chế cũng như thời gian làm đồ án có giới hạn, em vẫn chưa thể ứng dụng Reverb vào đồ án của mình mà vẫn đang phải sử dụng socket.io để tạo realtime cho trang web.

1.1.3.3 Queue & Task Scheduling

Queue (hàng đợi) và Task Scheduling (lập lịch tác vụ) là hai tính năng đặc biệt làm nên sự khác biệt của Laravel. Đây là 2 tính năng rất quan trọng trong Laravel, giúp cải thiện hiệu suất và trải nghiệm người dùng của ứng dụng web.

* Queue

Queue là một cơ chế xử lý tác vụ nền (background jobs) trong Laravel, cho phép bạn trì hoãn việc thực thi các tác vụ tốn thời gian hoặc cần xử lý bất đồng bộ. Thay vì thực hiện các tác vụ này ngay lập tức trong luồng xử lý chính của ứng dụng, bạn có thể đưa chúng vào một hàng đợi (queue) để xử lý sau đó.

Lợi ích của việc sử dụng queue:

- Cải thiện tốc độ phản hồi, các tác vụ nặng được xử lý nền, giúp giảm tải cho máy chủ web và tăng tốc độ phản hồi cho người dùng.
- Hỗ trợ xử lý bất đồng bộ, các tác vụ có thể được thực hiện độc lập với luồng xử lý chính, giúp ứng dụng hoạt động trơn tru hơn.
- Dễ dàng mở rộng khả năng xử lý của ứng dụng bằng cách tăng số lượng worker.
- Cung cấp các cơ chế để xử lý lỗi trong quá trình thực thi job, đảm bảo tính ổn định của ứng dụng.
- Hỗ trợ nhiều driver khác nhau như: database, redis, Amazon SQS, ...

* Task Scheduling

Task Scheduling là một tính năng mạnh mẽ trong Laravel cho phép tự động hóa việc thực thi các tác vụ định kỳ hoặc theo lịch trình cụ thể. Bạn có thể sử dụng Task Scheduling để thực hiện nhiều loại tác vụ khác nhau, từ việc gửi email nhắc nhở, tạo báo cáo, sao lưu dữ liệu, đến việc thực hiện các tác vụ bảo trì hệ thống.

Laravel Task Scheduling hoạt động dựa trên cơ chế cron jobs của hệ điều hành. Cron job là một tiện ích cho phép lên lịch thực thi các lệnh hoặc script tại các thời điểm cụ thể.

Laravel cung cấp một giao diện thân thiện để bạn định nghĩa các tác vụ và lịch trình của chúng trong file `app/Console/Kernel.php`. Khi đến thời điểm được lên lịch, Laravel sẽ tự động gọi cron job để thực thi tác vụ tương ứng.

Ưu điểm của Task Scheduling:

- Tự động hóa: Giúp tự động hóa các tác vụ định kỳ, giảm thiểu sự can thiệp thủ công.
- Cải thiện hiệu suất: Thực hiện các tác vụ nền, giúp giảm tải cho máy chủ web và tăng tốc độ phản hồi của ứng dụng.
- Linh hoạt: Hỗ trợ nhiều loại lịch trình khác nhau, từ đơn giản đến phức tạp.
- Đáng tin cậy: Sử dụng cơ chế cron job của hệ điều hành, đảm bảo tính ổn định và độ tin cậy cao.

Task Scheduling là một công cụ vô cùng hữu ích trong Laravel, giúp tự động hóa các tác vụ và cải thiện hiệu suất của ứng dụng. Với Task Scheduling, chúng ta có thể tập trung vào việc phát triển các tính năng chính của ứng dụng mà không cần lo lắng về việc thực hiện các tác vụ định kỳ một cách thủ công. Trong đề tài này, Task Scheduling được dùng để tạo tự động tạo thống kê theo ngày cho từng học sinh vào lúc 0h đêm mỗi ngày, giúp việc thống kê hiệu quả hơn mà không bị ảnh hưởng đến hiệu suất của trang web.

1.1.4 Ưu và nhược điểm của Laravel

* Ưu điểm:

- Tài liệu chính thống dễ hiểu, có example code cho từng nội dung. Một trong số ít Framework có tài liệu từ trang chủ dễ hiểu nhất
- Có một cộng đồng rộng lớn với những chuyên gia hỗ trợ, những bài viết chia sẻ, hướng dẫn cho người dùng.
- Mã nguồn mở, cho phép lập trình viên phát triển các package để xây dựng ứng dụng web và tái sử dụng cũng như chia sẻ đến cộng đồng một cách dễ dàng thông qua composer (tương tự npm trong các framework js hiện đại).
- Tương tác với cơ sở dữ liệu một cách an toàn và dễ dàng thông qua Eloquent ORM. Eloquent cung cấp một bộ ORM mạnh mẽ, có thể dễ dàng truy vấn, đồng thời đã được thiết lập sẵn các biện pháp chống các lỗ hổng như injection.

Ngoài ra Eloquent ORM còn cung cấp phương thức Chunk hỗ trợ xử lý dữ liệu lớn.

- Bảo mật cao với các phương thức bảo mật hiện đại như OAuth2, JWT,... thông qua các package mạnh mẽ như Passport, Sanctum, ... Và cơ chế phân quyền Spatie Permissions.
- Quản lý source code dễ dàng bằng các câu lệnh artisan, cho phép tạo file, cấu hình một cách tự động thông qua terminal.
- Khả năng tương thích với các thư viện frontend giúp dễ dàng xây dựng giao diện người dùng. Hiện tại, Laravel đang phát triển một frontend engine mới tên là Livewire với các tính năng như React nhưng chỉ cần trong 1 project và 100% PHP.

* Nhược điểm:

- Laravel có nhiều tính năng và cấu hình phức tạp, vì thế nó đòi hỏi kiến thức lập trình PHP cao hơn so với các framework khác.
- Dù cung cấp nhiều tính năng và package mở rộng, nhưng khả năng mở rộng khi cần tích hợp hay phát triển các tính năng mới từ dự án cũ cũng sẽ gặp nhiều khó khăn.
- Với những tính năng và bộ package mạnh mẽ, Laravel sẽ tốn nhiều tài nguyên bộ nhớ để có thể chạy ứng dụng.

1.2 NextJS 14

1.2.1 Giới thiệu

NextJS là framework mã nguồn mở được xây dựng trên nền tảng của React, cho phép chúng ta xây dựng các trang web tĩnh có tốc độ siêu nhanh và thân thiện với người dùng, cũng như xây dựng các ứng dụng web React. NextJS được ra đời vào năm 2016, thuộc sở hữu của Vercel. NextJS bắt đầu trở nên phổ biến vào năm 2018 và tiếp tục tăng trưởng mạnh mẽ trong cộng đồng phát triển web vào những năm sau đó. Sự kết hợp của các tính năng như Server-side Rendering (SSR) với Static Site Generation (SSG) đã giúp NextJS trở thành sự lựa chọn hấp dẫn cho nhiều dự án phát triển ứng dụng web.

Next.js cung cấp một số tính năng mạnh mẽ giúp đơn giản hóa quá trình phát triển ứng dụng web như:

- Server-Side Rendering (SSR): Next.js hỗ trợ SSR, cho phép render trước các trang trên máy chủ. Điều này giúp cải thiện hiệu suất trang web, đặc biệt là trên các thiết bị di động và kết nối chậm, đồng thời tăng khả năng SEO (Search Engine Optimization) của ứng dụng.
- Static Site Generation (SSG): Next.js cũng hỗ trợ SSG, cho phép tạo ra các trang web tĩnh từ các thành phần React. Các trang web tĩnh có tốc độ tải trang cực nhanh và giảm tải cho máy chủ, vì chúng không cần phải được render lại mỗi khi có yêu cầu từ người dùng.
- Incremental Static Regeneration (ISR): Đây là một tính năng kết hợp giữa SSR và SSG, cho phép bạn tạo ra các trang tĩnh nhưng vẫn có thể cập nhật chúng một cách tăng dần theo thời gian. Điều này giúp bạn có được lợi ích của cả hai phương pháp SSR và SSG.
- Routing: Next.js có hệ thống routing tích hợp, dựa trên cấu trúc thư mục, giúp dễ dàng định nghĩa các tuyến đường (route) cho ứng dụng. Bạn chỉ cần tạo các tệp trong thư mục pages và Next.js sẽ tự động tạo các route tương ứng.
- API Routes: Next.js cho phép bạn dễ dàng tạo các API route để xây dựng các ứng dụng backend. Bạn có thể viết các hàm xử lý (handler) trong thư mục pages/api và Next.js sẽ tự động biến chúng thành các API endpoint.
- TypeScript Support: Next.js hỗ trợ TypeScript, một ngôn ngữ lập trình có kiểu dữ liệu mạnh, giúp bạn viết mã React rõ ràng, dễ bảo trì và ít lỗi hơn.

NextJS là một trong những web framework mới nhất và đang trở nên vô cùng phổ biến trong cộng đồng web hiện nay. Tốc độ cập nhật và phát triển của framework này chính là một điểm khiến framework này trở nên nổi bật và ưu dùng đối với những lập trình viên hiện đại. Tính đến nay, phiên bản mới nhất là Next.js 14, phiên bản mới nhất được phát hành vào tháng 10 năm 2023, mang đến nhiều cải tiến đáng kể như cơ chế App Router, Server Component, Streaming, ...

NextJS không chỉ đơn giản là một framework React, nó là một môi trường phát triển mạnh mẽ, mang lại hiệu suất tuyệt vời và cho trải nghiệm người dùng vượt trội. Bằng cách tận dụng các tính năng quan tích hợp sẵn, NextJS cho phép chúng ta xây dựng các ứng dụng React một cách nhanh chóng và hiệu quả.

1.2.2 Client Component

Client Components là một tính năng mới được giới thiệu trong Next.js 14, mang đến sự linh hoạt và cải thiện hiệu suất cho các ứng dụng React. Với Client Components, các nhà phát triển có thể lựa chọn render và thực thi các thành phần trực tiếp trên trình duyệt của người dùng (client-side rendering). Điều này đặc biệt hữu ích cho các thành phần có tính tương tác cao, yêu cầu cập nhật dữ liệu liên tục hoặc tận dụng các API của trình duyệt. Client Components giúp giảm tải cho máy chủ và tăng tốc độ phản hồi của ứng dụng, vì chúng không cần phải được render lại từ đầu trên máy chủ mỗi khi có thay đổi.

Một điểm mạnh của Client Components là khả năng tương tác trực tiếp với DOM (Document Object Model) của trình duyệt. Điều này cho phép các nhà phát triển thực hiện các thao tác cập nhật giao diện người dùng một cách nhanh chóng và mượt mà, tạo ra trải nghiệm người dùng tốt hơn. Client Components cũng hỗ trợ việc sử dụng các thư viện JavaScript chỉ chạy trên client, mở rộng khả năng và tính linh hoạt của ứng dụng.

Tuy nhiên, Client Components cũng có một số hạn chế cần lưu ý. Do được thực thi trên client, chúng có thể làm tăng thời gian tải trang ban đầu nếu không được tối ưu hóa đúng cách. Ngoài ra, Client Components không phù hợp cho các thành phần chứa thông tin nhạy cảm hoặc cần được bảo vệ, vì mã nguồn của chúng có thể bị xem và sửa đổi bởi người dùng. Do đó, việc lựa chọn giữa Client Components và Server Components cần được cân nhắc kỹ lưỡng dựa trên yêu cầu cụ thể của từng thành phần và ứng dụng.

1.2.3 Ưu điểm, nhược điểm

Ưu điểm:

- Hiệu suất cao: Next.js tận dụng các kỹ thuật như SSR, SSG, ISR và code splitting để tối ưu hóa hiệu suất tải trang và trải nghiệm người dùng.
- SEO thân thiện: Nhờ SSR và SSG, các trang Next.js có thể được lập chỉ mục bởi các công cụ tìm kiếm một cách dễ dàng, giúp cải thiện thứ hạng SEO của ứng dụng.
- Dễ phát triển: Next.js cung cấp một cấu trúc dự án rõ ràng, hệ thống routing đơn giản và nhiều tính năng tiện ích khác, giúp giảm thiểu thời gian và công sức phát triển ứng dụng.
- Cộng đồng lớn mạnh: Next.js có một cộng đồng người dùng và nhà phát triển đông đảo, sẵn sàng hỗ trợ và chia sẻ kiến thức.

Nhược điểm:

- Khó làm chủ: Để tận dụng hết các tính năng của Next.js, bạn cần có kiến thức vững chắc về React và các khái niệm liên quan đến SSR, SSG, ISR.
- Hạn chế tùy biến: Next.js áp đặt một số quy ước về cấu trúc dự án và cách thức hoạt động, có thể gây khó khăn nếu bạn muốn tùy biến cao.
- Khó khăn trong việc tối ưu hóa: Việc tối ưu hóa hiệu suất của các ứng dụng Next.js phức tạp có thể đòi hỏi kiến thức chuyên sâu về các kỹ thuật tối ưu hóa.

1.3 Cơ sở dữ liệu

1.3.1 MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến được sử dụng rộng rãi trên toàn thế giới. Được phát triển bởi Oracle Corporation, MySQL cung cấp một cách tiếp cận đơn giản và mạnh mẽ cho việc lưu trữ và quản lý dữ liệu trong các ứng dụng web, từ các dự án cá nhân đến các hệ thống doanh nghiệp lớn.

Một trong những điểm mạnh của MySQL là tính nhất quán và độ ổn định cao. Nó cung cấp các tính năng cơ bản của một hệ quản trị cơ sở dữ liệu quan hệ như quản lý bảng, truy vấn dữ liệu, thêm, sửa, xóa dữ liệu và các chức năng phức tạp như giao dịch, khóa và xác thực người dùng..

Đối với những người mới bắt đầu, MySQL cung cấp một giao diện dễ sử dụng và một cộng đồng lớn với nhiều tài liệu và nguồn thông tin hữu ích. Nó cũng có sẵn trong các bản phân phối mã nguồn mở như MySQL Community Edition, giúp cho việc triển khai và phát triển các ứng dụng với chi phí thấp hoặc miễn phí.

Với tính linh hoạt và khả năng tương thích cao, MySQL có thể tích hợp với nhiều ngôn ngữ lập trình và framework phổ biến như PHP, Python, Java, Node.js và Ruby on Rails. Điều này giúp cho việc phát triển và triển khai các ứng dụng web trở nên linh hoạt và dễ dàng hơn.

=> MySQL là một hệ quản trị cơ sở dữ liệu mạnh mẽ và phổ biến với nhiều tính năng và ưu điểm. Tính đơn giản, tính linh hoạt và tính nhất quán của nó đã khiến cho MySQL trở thành một lựa chọn ưa thích cho việc lưu trữ và quản lý dữ liệu trong các ứng dụng web.

1.3.2 Redis

Redis là một hệ thống cơ sở dữ liệu key-value mã nguồn mở được phát triển bởi Salvatore Sanfilippo. Nó là một trong những hệ thống cơ sở dữ liệu NoSQL phổ biến nhất hiện nay, nổi tiếng với hiệu suất cao, khả năng mở rộng tốt và các tính năng linh hoạt.

Redis được thiết kế để xử lý các tải công việc nặng về dữ liệu và các truy vấn dữ liệu đồng thời mà không ảnh hưởng đến hiệu suất. Điều này làm cho Redis trở thành một lựa chọn phổ biến cho việc lưu trữ dữ liệu tạm thời, caching, hàng đợi tin nhắn, và nhiều ứng dụng khác có yêu cầu về xử lý dữ liệu nhanh.

Một trong những tính năng nổi bật của Redis là khả năng lưu trữ dữ liệu trong bộ nhớ chính, cho phép truy xuất và cập nhật dữ liệu với tốc độ rất cao. Ngoài ra, Redis cũng hỗ trợ các loại dữ liệu phong phú như chuỗi, hash, danh sách, tập hợp, và bản đồ bit, cung cấp cho người dùng một loạt các cấu trúc dữ liệu để làm việc.

Ngoài ra, Redis còn có các tính năng bảo mật như xác thực, quyền truy cập và mật khẩu, giúp bảo vệ dữ liệu khỏi các cuộc tấn công và truy cập trái phép. Redis là một hệ thống cơ sở dữ liệu key-value mạnh mẽ, linh hoạt và hiệu quả, được sử dụng rộng rãi trong các ứng dụng web và hệ thống phân phối. Với khả năng lưu trữ dữ liệu trong bộ nhớ chính, tính năng mở rộng và các tính năng bảo mật, Redis là một lựa chọn lý tưởng cho việc xây dựng các ứng dụng có yêu cầu về hiệu suất và khả năng mở rộng.

** Redis với Laravel*

Redis là một trong những công nghệ phổ biến được sử dụng trong việc cải thiện hiệu suất và tính năng của các ứng dụng Laravel. Laravel cung cấp tích hợp sẵn với Redis thông qua các package mạnh mẽ như predis/predis hoặc illuminate/redis.

- Caching: Một trong những cách phổ biến nhất để sử dụng Redis trong Laravel là để lưu trữ dữ liệu cache. Redis cung cấp hiệu suất cao và khả năng mở rộng tốt cho việc lưu trữ dữ liệu cache, giúp tăng tốc độ truy cập dữ liệu và giảm thiểu thời gian phản hồi của ứng dụng.

- Session Storage: Laravel cũng cho phép lưu trữ phiên người dùng trong Redis thay vì lưu trữ trên ổ đĩa, giúp tăng cường tính bảo mật và khả năng mở rộng của ứng dụng.

- Queue Backend: Redis cung cấp một backend mạnh mẽ cho hệ thống hàng đợi của Laravel. Bằng cách sử dụng Redis làm đối tượng hàng đợi, bạn có thể xử lý các công việc hàng đợi một cách hiệu quả và đồng bộ.

- Real-time Data: Redis cũng được sử dụng trong Laravel để xây dựng các tính năng real-time như chat, thông báo hoặc đồng bộ dữ liệu trực tiếp giữa các ứng dụng khác nhau.

- Rate Limiting: Redis cung cấp các công cụ mạnh mẽ để triển khai các giới hạn tốc độ, giúp kiểm soát lưu lượng của các yêu cầu từ các nguồn khác nhau.

Redis là một công nghệ mạnh mẽ được sử dụng trong Laravel để cải thiện hiệu suất và tính năng của các ứng dụng. Tích hợp Redis vào Laravel giúp tăng cường hiệu suất, đồng thời cung cấp các tính năng mở rộng và linh hoạt cho các ứng dụng web.

1.4 Python và khoa học dữ liệu

Python là một ngôn ngữ lập trình bậc cao, thông dịch, hướng đối tượng và có kiểu động. Python được tạo ra bởi Guido van Rossum vào cuối những năm 1980 và được phát hành lần đầu tiên vào năm 1991. Python được thiết kế với mục tiêu dễ đọc, dễ học và dễ sử dụng, với cú pháp đơn giản và rõ ràng.

Python đã nổi lên như một ngôn ngữ lập trình hàng đầu trong lĩnh vực máy học (Machine Learning), học sâu (Deep Learning) và trí tuệ nhân tạo (Artificial Intelligence - AI) nhờ vào những ưu điểm sau:

- Thư viện phong phú: Python sở hữu một hệ sinh thái khổng lồ các thư viện chuyên dụng cho máy học, học sâu và AI.
- Cộng đồng lớn mạnh: Python có một cộng đồng người dùng và nhà phát triển đông đảo trong lĩnh vực máy học, học sâu và AI.
- Tính đơn giản và dễ đọc: Cú pháp của Python đơn giản và rõ ràng, giúp giảm thiểu thời gian và công sức cần thiết để viết mã.
- Tính linh hoạt: Python là một ngôn ngữ đa năng, có thể được sử dụng cho nhiều mục đích khác nhau, từ phân tích dữ liệu, xây dựng mô hình, đến triển khai ứng dụng.
- Nền tảng độc lập: Python có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux, giúp dễ dàng chia sẻ và cộng tác trong các dự án.

Nhờ những ưu điểm này, Python đã trở thành ngôn ngữ được lựa chọn hàng đầu cho việc phát triển các ứng dụng máy học, học sâu và AI. Các công ty công nghệ lớn như

Google, Facebook, và Microsoft đều sử dụng Python rộng rãi trong các dự án nghiên cứu và phát triển sản phẩm của mình.

1.4.1 Học máy trong Python

Python không chỉ là một ngôn ngữ lập trình phổ biến mà còn là một công cụ mạnh mẽ được sử dụng rộng rãi trong lĩnh vực máy học, học sâu và trí tuệ nhân tạo. Với cú pháp đơn giản, dễ đọc và khả năng tích hợp với nhiều thư viện chuyên dụng như NumPy, pandas, scikit-learn, TensorFlow, và PyTorch, Python đã trở thành sự lựa chọn hàng đầu của các nhà khoa học dữ liệu, kỹ sư máy học và các chuyên gia AI. Nhiều công ty công nghệ hàng đầu thế giới như Google, Facebook, Microsoft và OpenAI đều sử dụng Python làm ngôn ngữ chính để phát triển các ứng dụng và giải pháp AI tiên tiến. Sự phổ biến của Python trong cộng đồng khoa học và công nghệ đã tạo ra một hệ sinh thái phong phú với nhiều tài liệu, hướng dẫn và cộng đồng hỗ trợ, giúp việc học và áp dụng Python trở nên dễ dàng hơn bao giờ hết.

1.4.1.1 Học máy là gì

Học máy (Machine Learning - ML) là một lĩnh vực của trí tuệ nhân tạo (AI) tập trung vào việc phát triển các thuật toán và mô hình cho phép máy tính tự học từ dữ liệu và cải thiện hiệu suất của chúng trong việc thực hiện các nhiệm vụ cụ thể mà không cần lập trình rõ ràng. Mục tiêu của học máy là xây dựng các hệ thống có khả năng tự động phân tích và hiểu dữ liệu, từ đó đưa ra các quyết định hoặc dự đoán chính xác. Python, với cú pháp đơn giản, dễ đọc và hệ sinh thái thư viện phong phú, đã trở thành một trong những ngôn ngữ lập trình hàng đầu trong lĩnh vực này.

Học máy được chia ra thành 2 loại chính thường thấy như sau: Học có giám sát (Supervised Learning) và học không giám sát (Unsupervised Learning), hay đơn giản hơn có thể hiểu là nhóm máy học có hướng dẫn và không có hướng dẫn. Supervised Learning là mô hình được huấn luyện trên một tập dữ liệu có nhãn (labeled data), trong đó mỗi mẫu dữ liệu được gắn với một kết quả mong muốn. Unsupervised Learning là mô hình huấn luyện trên một tập dữ liệu không có nhãn (unlabeled data).

1.4.1.2 Nhóm học máy có giám sát

Học máy giám sát là một nhánh của học máy, trong đó mô hình được huấn luyện trên một tập dữ liệu có nhãn (labeled data). Mỗi mẫu dữ liệu trong tập huấn luyện bao gồm

một tập các đặc trưng (features) đầu vào và một nhãn (label) tương ứng, đại diện cho kết quả mong muốn. Mục tiêu của học máy giám sát là xây dựng một mô hình có khả năng học từ tập dữ liệu huấn luyện này để dự đoán nhãn cho các mẫu dữ liệu mới chưa biết trước nhãn.

Bản chất của học máy giám sát là tìm ra một hàm số hoặc một quy tắc ánh xạ từ không gian đặc trưng đầu vào sang không gian nhãn. Quá trình huấn luyện mô hình liên quan đến việc điều chỉnh các tham số của mô hình sao cho nó có thể dự đoán nhãn một cách chính xác nhất có thể trên tập dữ liệu huấn luyện. Sau khi được huấn luyện, mô hình có thể được sử dụng để dự đoán nhãn cho các mẫu dữ liệu mới.

1.4.2 Thư viện *Scikit learn*

Scikit-learn là một thư viện mã nguồn mở hàng đầu trong lĩnh vực học máy (Machine Learning) được xây dựng trên nền tảng Python. Scikit-learn cung cấp một bộ công cụ toàn diện và dễ sử dụng để thực hiện các tác vụ học máy khác nhau, từ tiền xử lý dữ liệu, lựa chọn đặc trưng, huấn luyện mô hình, đánh giá mô hình đến lựa chọn mô hình tối ưu.

Các tính năng chính của Scikit-learn:

- Thuật toán học máy đa dạng: Scikit-learn cung cấp một bộ sưu tập phong phú các thuật toán học máy, bao gồm các thuật toán phân loại (classification), hồi quy (regression), phân cụm (clustering) và giảm chiều dữ liệu (dimensionality reduction).
- Tiền xử lý dữ liệu: Scikit-learn cung cấp nhiều công cụ để xử lý dữ liệu thô, chẳng hạn như chuẩn hóa (scaling), mã hóa (encoding), và xử lý dữ liệu thiếu (missing data imputation).
- Lựa chọn đặc trưng: Scikit-learn cung cấp các phương pháp để lựa chọn các đặc trưng quan trọng nhất trong tập dữ liệu, giúp cải thiện hiệu suất của mô hình và giảm thiểu overfitting.
- Đánh giá mô hình: Scikit-learn cung cấp nhiều độ đo để đánh giá hiệu suất của mô hình, chẳng hạn như độ chính xác (accuracy), độ đo F1 (F1 score), và diện tích dưới đường cong ROC (AUC-ROC).

- Lựa chọn mô hình: Scikit-learn cung cấp các kỹ thuật như tìm kiếm tham số ngẫu nhiên (randomized search) và tìm kiếm lưới (grid search) để tìm ra các siêu tham số tối ưu cho mô hình.

Ưu điểm của Scikit-learn:

- Dễ sử dụng: Scikit-learn có một API (Application Programming Interface) đơn giản và nhất quán, giúp người dùng dễ dàng học và sử dụng thư viện.
- Hiệu quả: Scikit-learn được tối ưu hóa để hoạt động hiệu quả trên các tập dữ liệu lớn, giúp giảm thiểu thời gian và tài nguyên tính toán cần thiết.
- Linh hoạt: Scikit-learn cung cấp nhiều tùy chọn cấu hình cho các thuật toán, cho phép bạn điều chỉnh mô hình để phù hợp với bài toán cụ thể.
- Cộng đồng lớn mạnh: Scikit-learn có một cộng đồng người dùng và nhà phát triển lớn, sẵn sàng hỗ trợ và chia sẻ kiến thức.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ

2.1 Thiết kế hệ thống

2.1.1 Sơ đồ Usecase

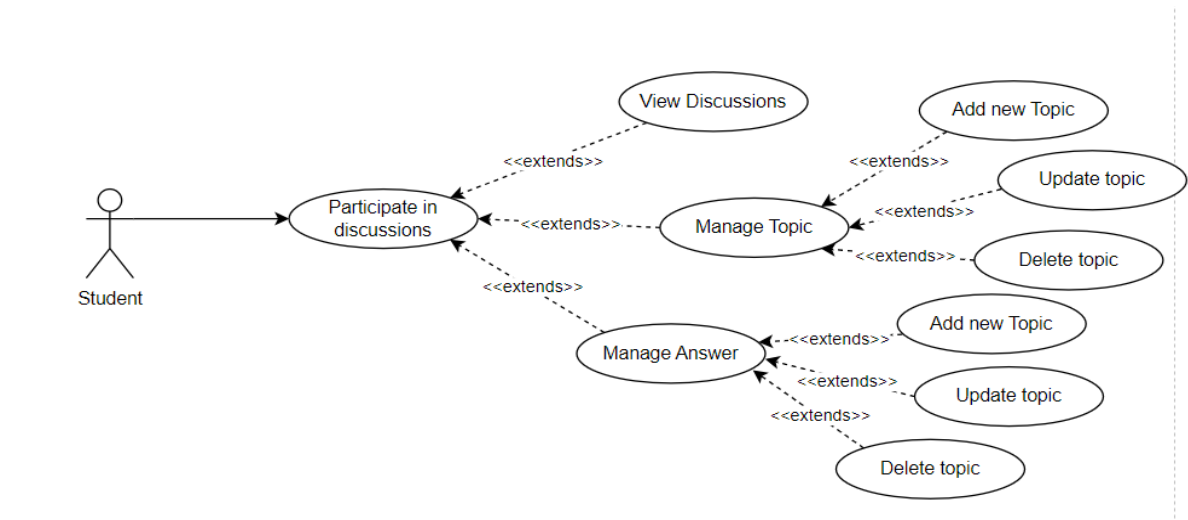
2.1.1.1 Sơ đồ Usecase tổng quát



Hình 2. 1 Sơ đồ Usecase tổng quát

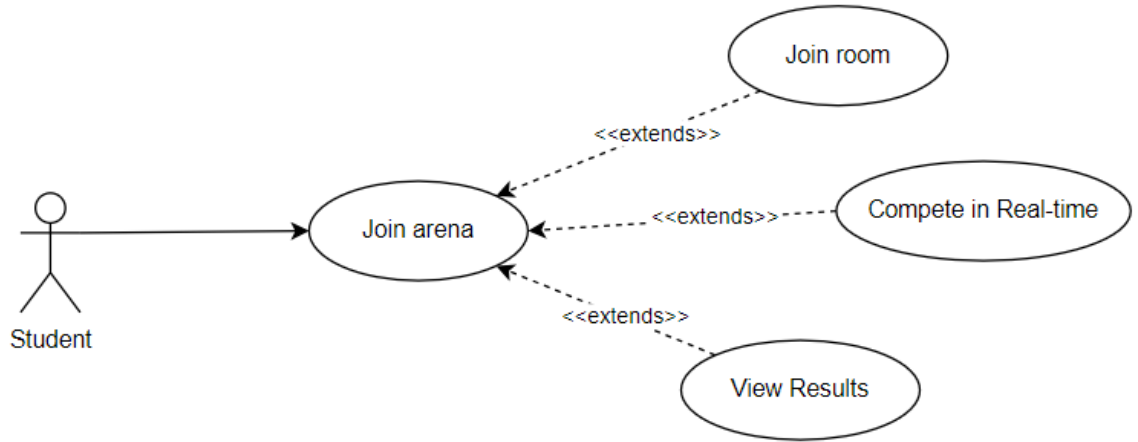
2.1.1.2 Sơ đồ Usecase phân rã

* Sơ đồ phân rã Usecase Thảo luận



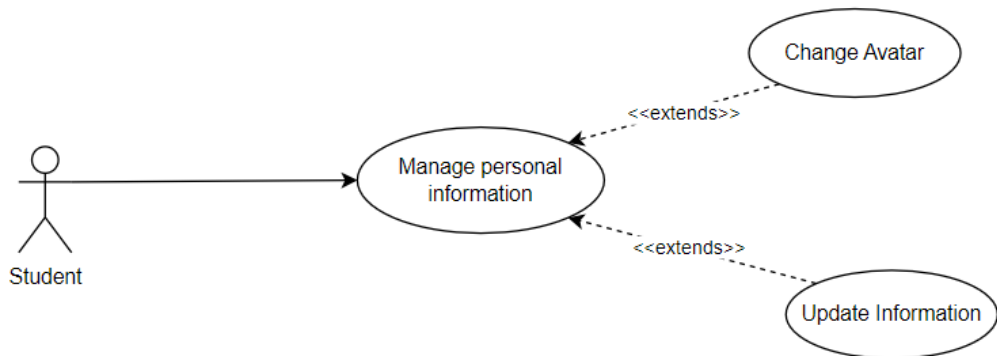
Hình 2. 2 Sơ đồ phân rã Usecase Thảo luận

* Sơ đồ phân rã Usecase Tham gia thi đấu



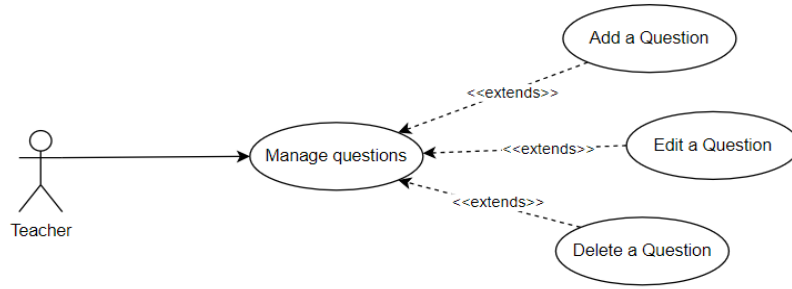
Hình 2. 3 Sơ đồ phân rã Usecase Tham gia thi đấu

* Sơ đồ phân rã Usecase Quản lý thông tin cá nhân



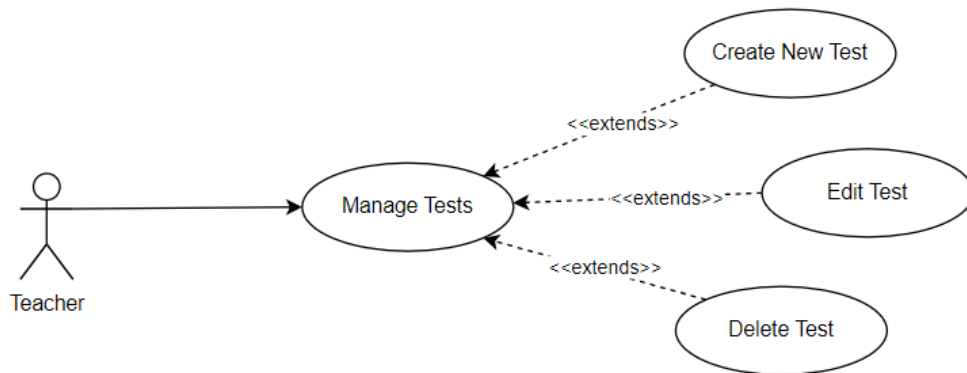
Hình 2. 4 Sơ đồ phân rã Usecase Quản lý thông tin cá nhân

* Sơ đồ phân rã Usecase Quản lý câu hỏi



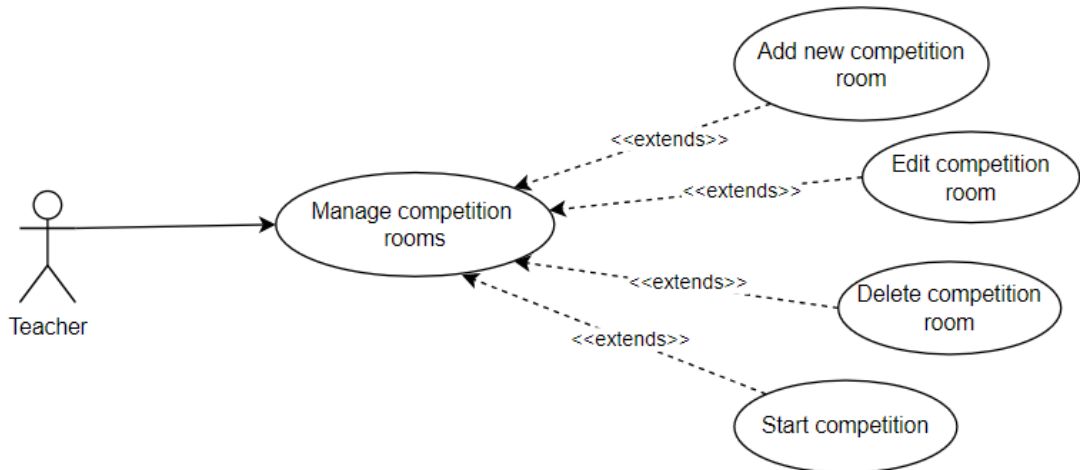
Hình 2. 5 Sơ đồ phân rã Usecase Quản lý câu hỏi

* Sơ đồ phân rã Usecase Quản lý bài thi



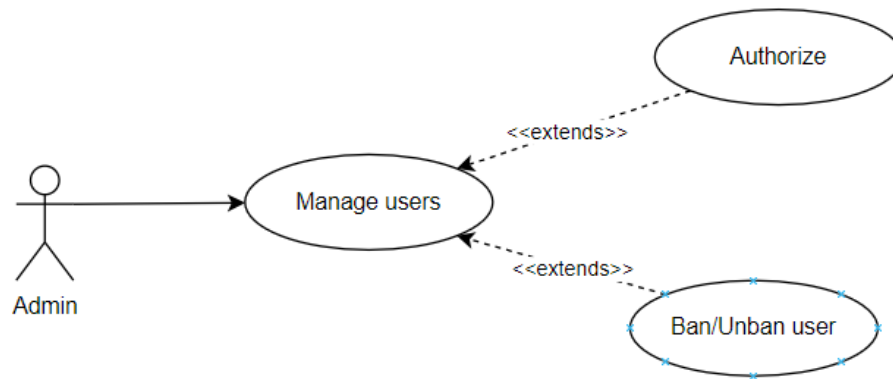
Hình 2. 6 Sơ đồ phân rã Usecase Quản lý bài thi

* Sơ đồ phân rã Usecase Quản lý phòng thi



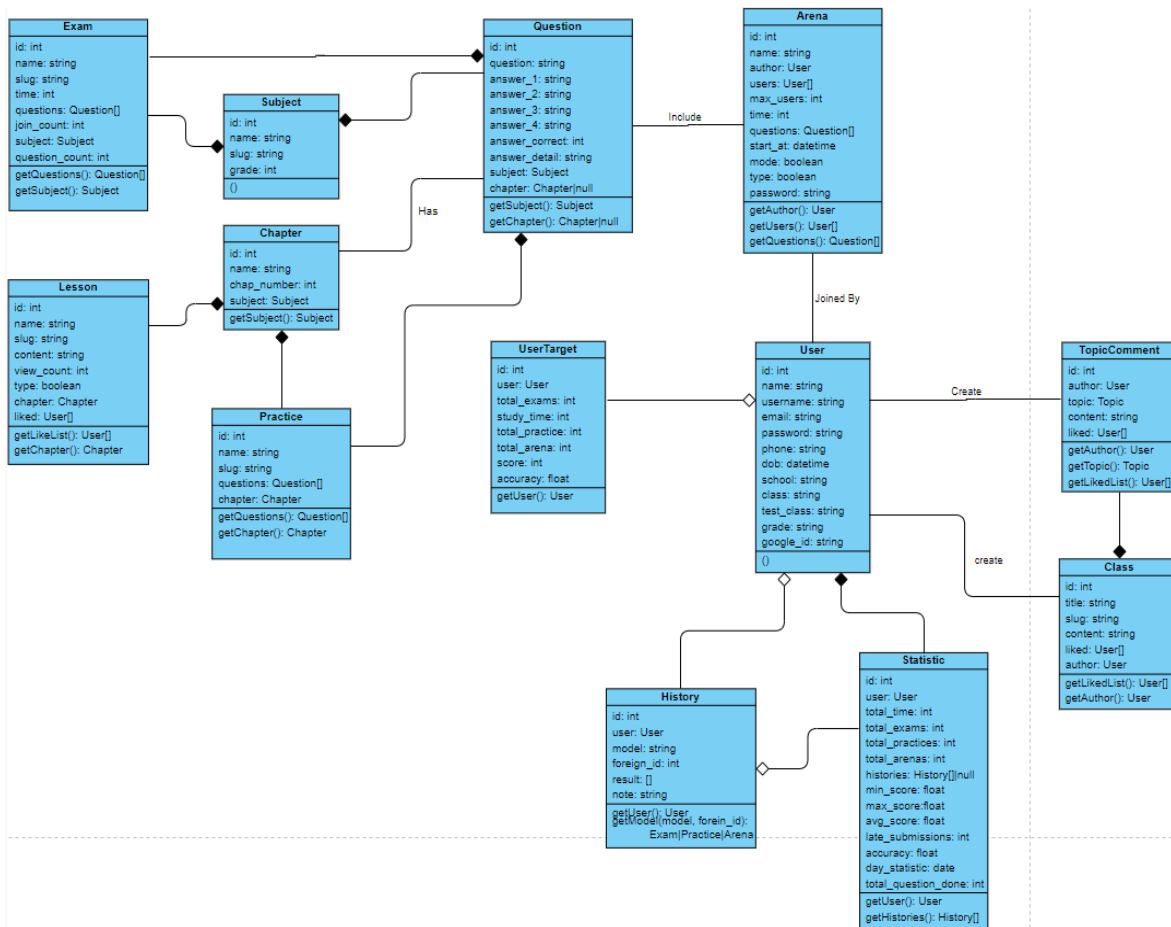
Hình 2. 7 Sơ đồ phân rã Usecase Quản lý phòng thi

* Sơ đồ phân rã Usecase Quản lý người dùng



Hình 2. 8 Sơ đồ phân rã Usecase Quản lý người dùng

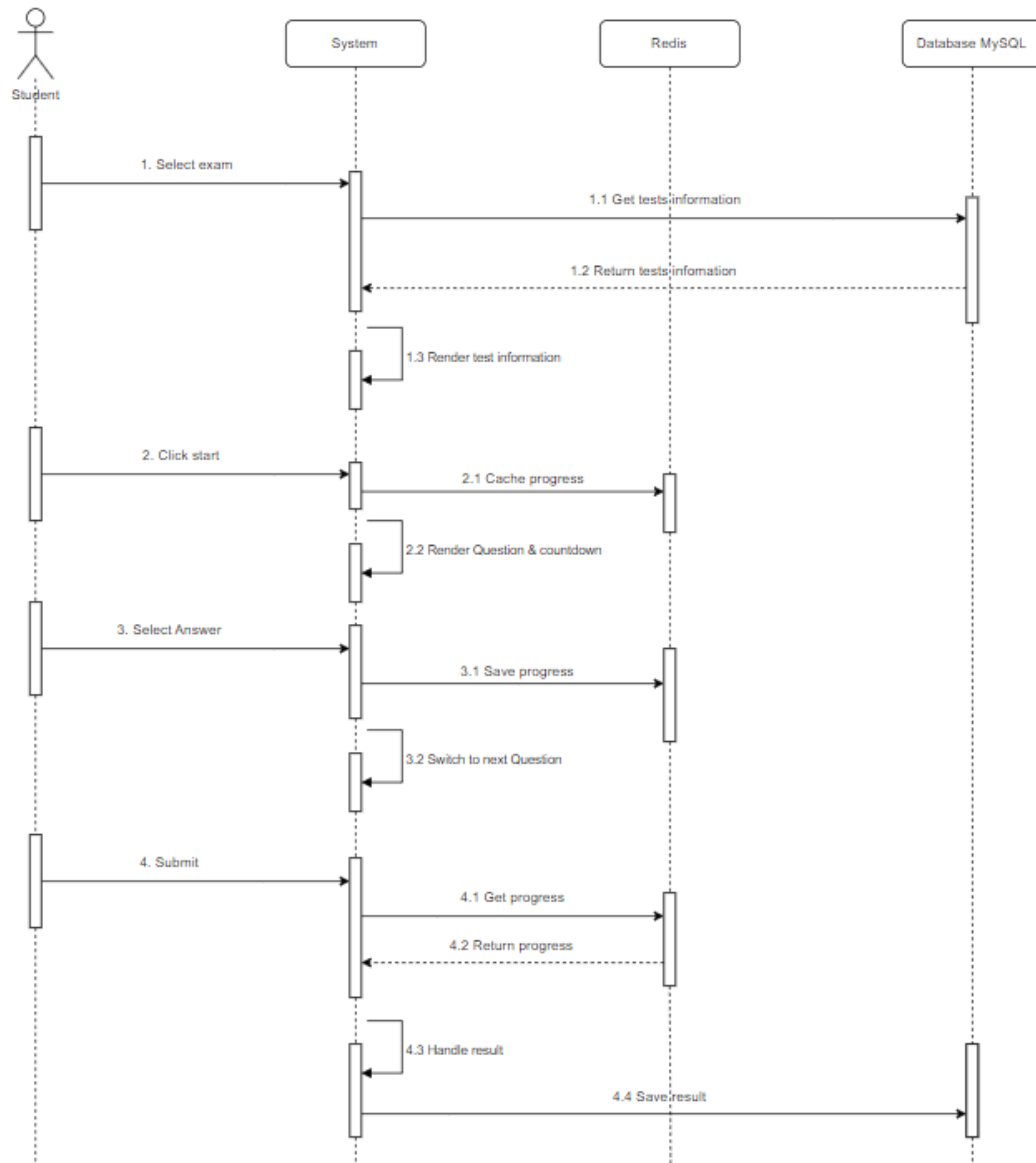
2.1.2 Sơ đồ Class



Hình 2. 9 Sơ đồ Class

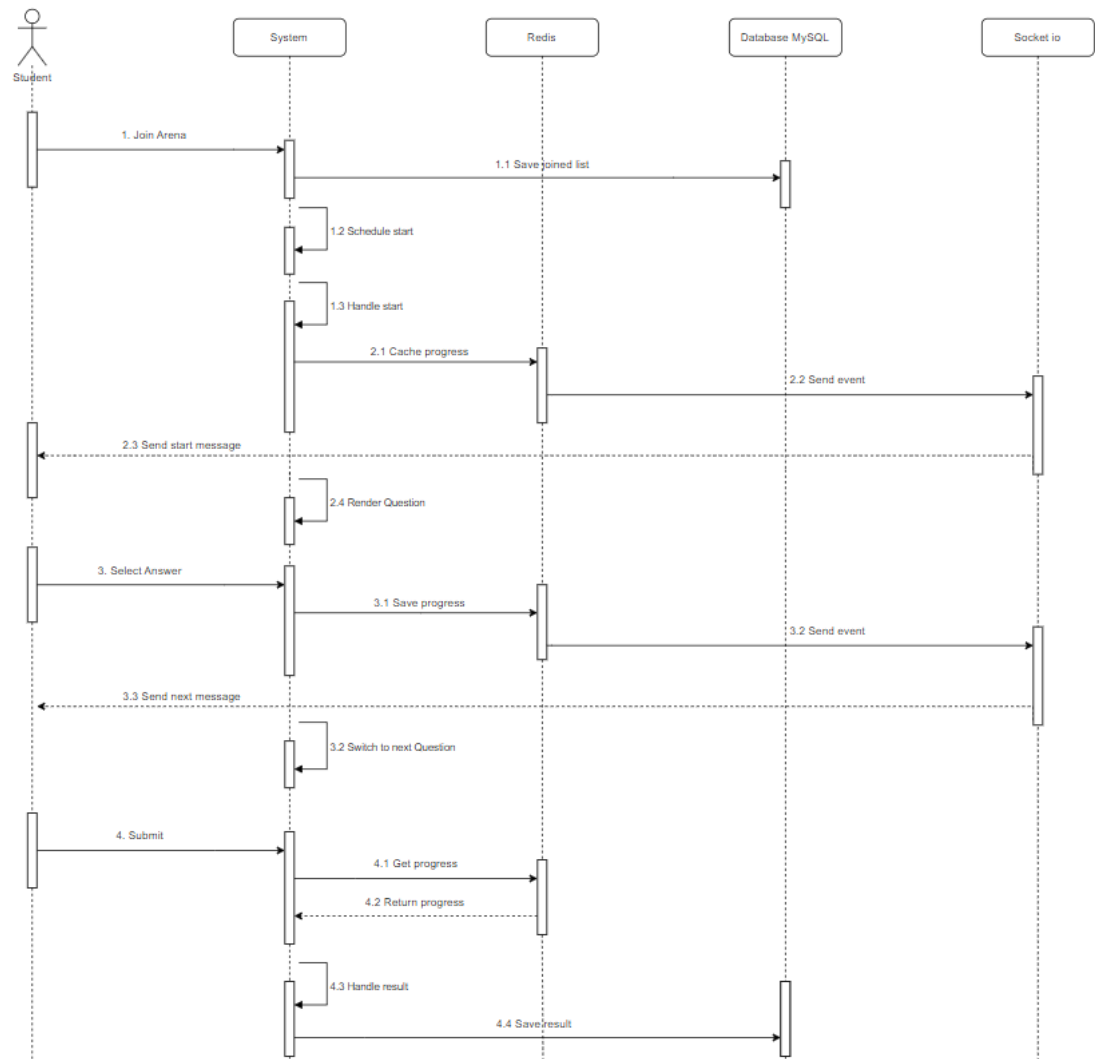
2.1.3 Sơ đồ tuần tự

2.1.3.1 Sơ đồ tuần tự thi thử



Hình 2. 10 Sơ đồ tuần tự thi thử

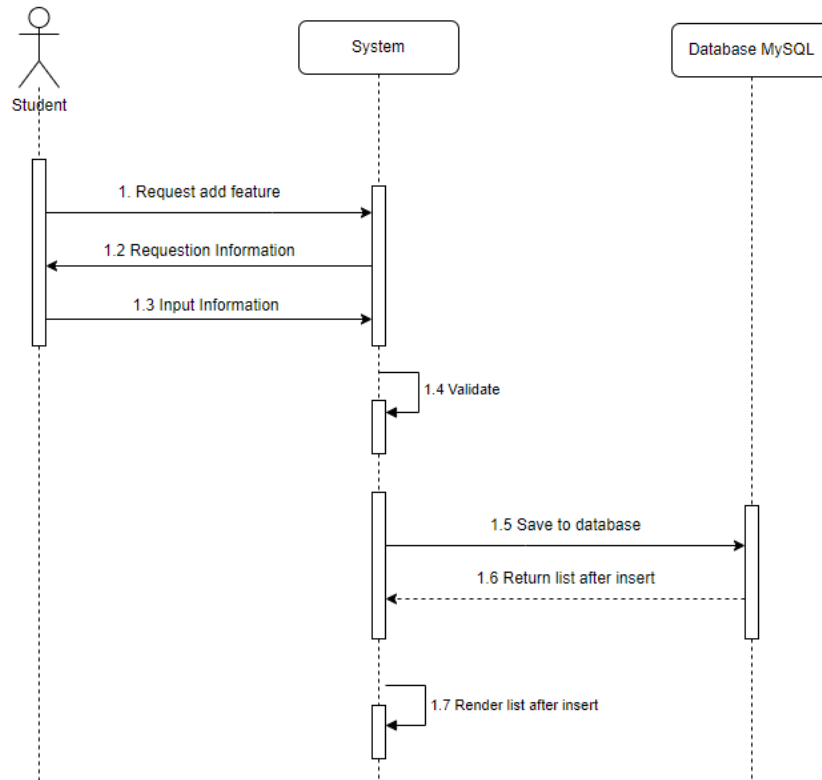
3.1.3.2 Sơ đồ tuần tự đấu trường



Hình 2. 11 Sơ đồ tuần tự đấu trường

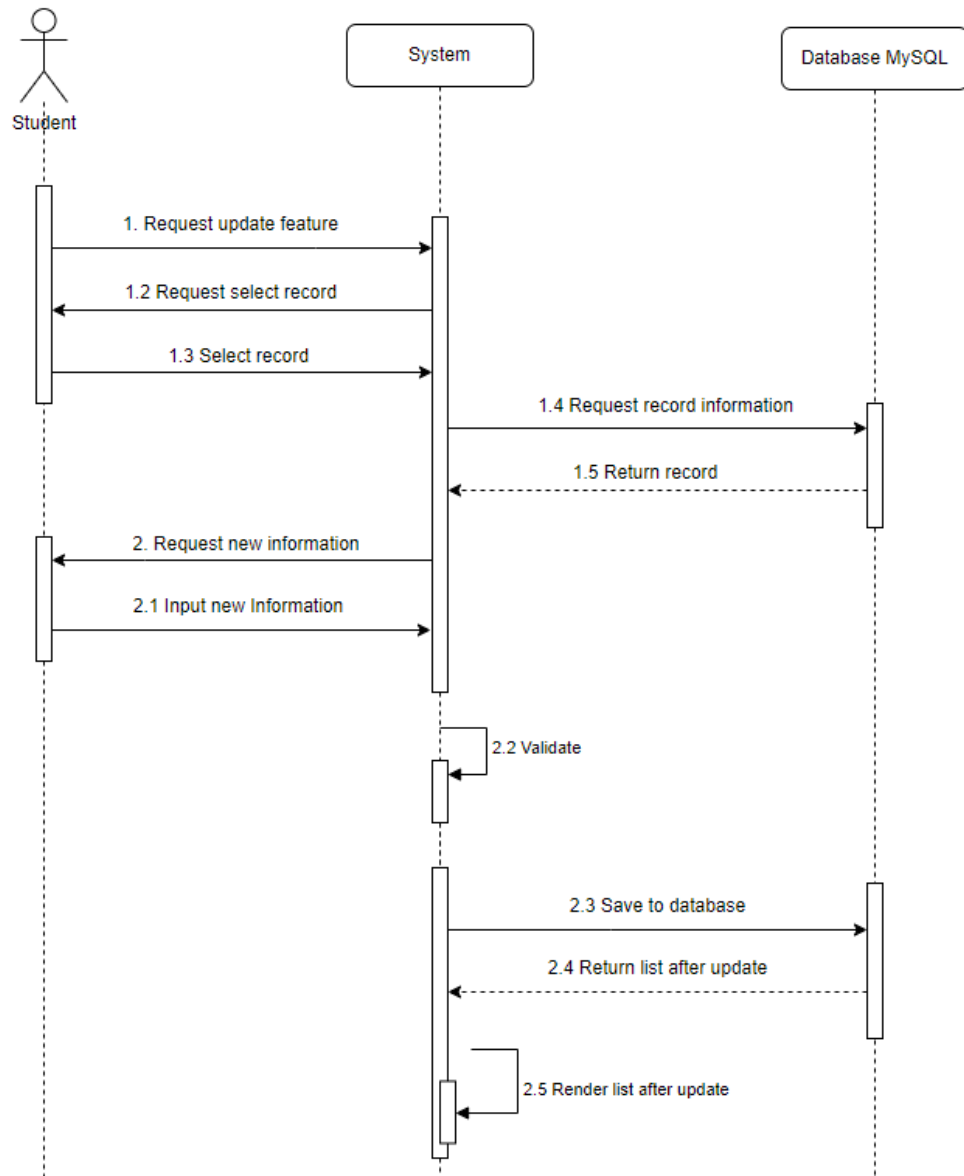
2.1.3.3 Sơ đồ tuần tự quản lý (thêm sửa xóa)

* Thêm mới bản ghi



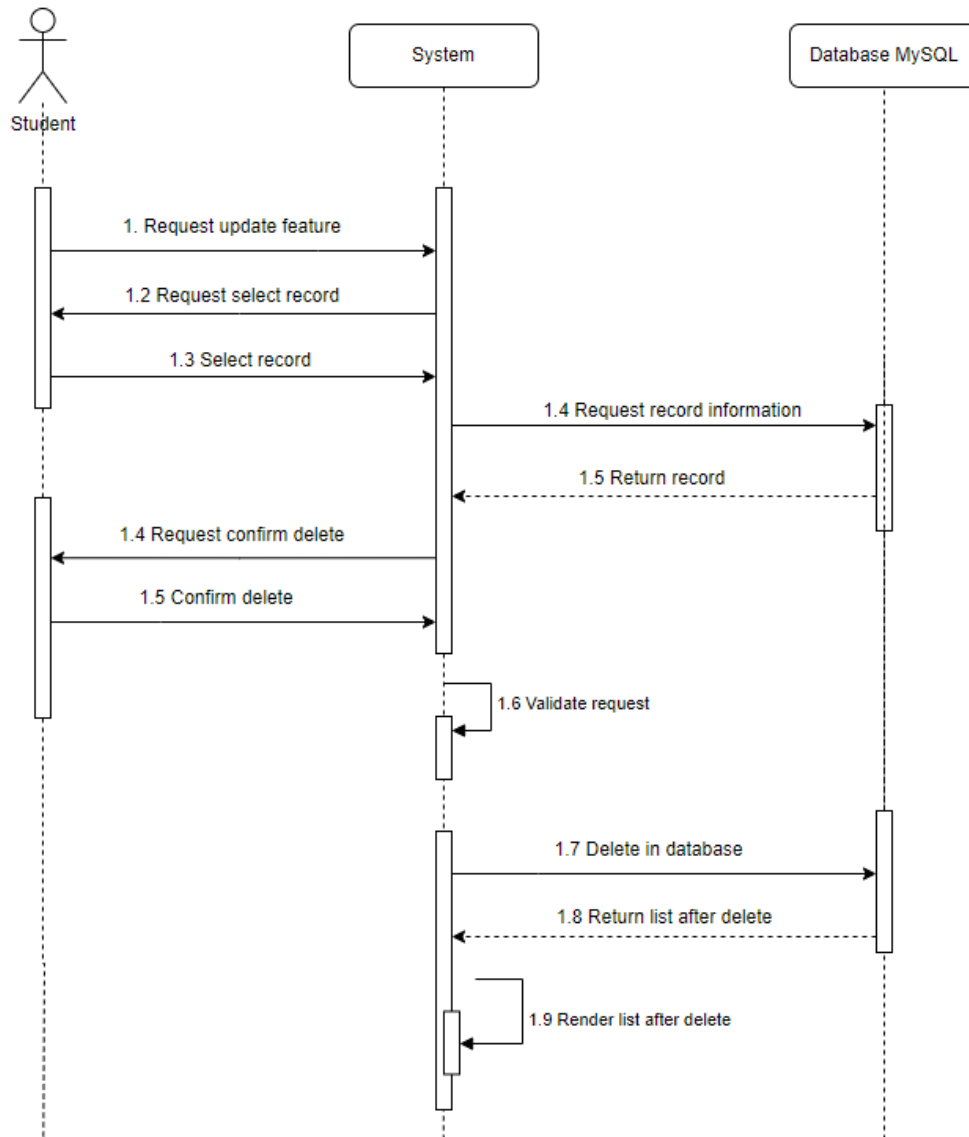
Hình 2. 12 Sơ đồ tuần tự Thêm mới bản ghi

* Cập nhật bản ghi



Hình 2. 13 Sơ đồ tuần tự Cập nhật bản ghi

* Xóa bản ghi



Hình 2. 14 Sơ đồ tuần tự xóa bản ghi

2.2 Thiết kế cơ sở dữ liệu

2.2.1 Mô tả bài toán

Trang web luyện thi Đại Học trực tuyến được thiết kế nhằm cung cấp một môi trường học tập toàn diện và hiệu quả, bao gồm các tính năng như hỏi – đáp, cung cấp lý thuyết và bài tập SGK, luyện tập giải đề, tính năng đấu trường và thống kê tình hình ôn luyện của người dùng. Cơ sở dữ liệu được xây dựng cần hỗ trợ việc quản lý người dùng, nội dung học tập, các kỳ thi và tương tác giữa người dùng một cách hiệu quả. Dữ liệu được lưu trữ:

1. Xác thực và Phân quyền: Nền tảng yêu cầu người dùng đăng ký và đăng nhập để sử dụng các tính năng. Người dùng có thể là học sinh, giáo viên hoặc quản trị viên/người kiểm duyệt. Mỗi loại tài khoản sẽ có quyền hạn khác nhau.

Bảng users: Lưu trữ thông tin của tất cả người dùng, bao gồm tên đăng nhập, email, mật khẩu và vai trò (role).

2. Tính năng hỏi – đáp: Người dùng có thể tạo các chủ đề thảo luận và đặt câu hỏi để nhận sự trợ giúp từ cộng đồng. Các bình luận sẽ được sử dụng để trao đổi và góp ý.

Bảng topics: Lưu trữ các chủ đề thảo luận được tạo bởi người dùng.

Bảng topic_comments: Lưu trữ các bình luận trên từng chủ đề, giúp người dùng tương tác với nhau.

3. Cung cấp lý thuyết và bài tập SGK: Nền tảng cung cấp nội dung học tập theo chương trình chuẩn, bao gồm lý thuyết và các bài tập có đáp án.

Bảng subjects: Lưu trữ các môn học khác nhau.

Bảng chapters: Lưu trữ các chương trong từng môn học.

Bảng lessons: Lưu trữ các bài học trong mỗi chương, bao gồm nội dung lý thuyết và bài tập.

4. Luyện tập giải đề: Học sinh có thể luyện tập các loại đề thi mẫu dưới hình thức trắc nghiệm trong thời gian quy định để chuẩn bị cho kỳ thi thật.

Bảng exams: Lưu trữ thông tin về các đề thi, bao gồm thời gian làm bài và danh sách các câu hỏi.

Bảng questions: Lưu trữ các câu hỏi trắc nghiệm, bao gồm nội dung câu hỏi và các phương án trả lời.

5. Tính năng đấu trường: Người dùng có thể tạo phòng thi đấu để thi cùng lúc với nhau (realtime), giúp tăng cường tính cạnh tranh và tương tác.

Bảng arenas: Lưu trữ thông tin về các phòng thi đấu, bao gồm người tạo, thời gian bắt đầu và kết thúc, và danh sách người tham gia.

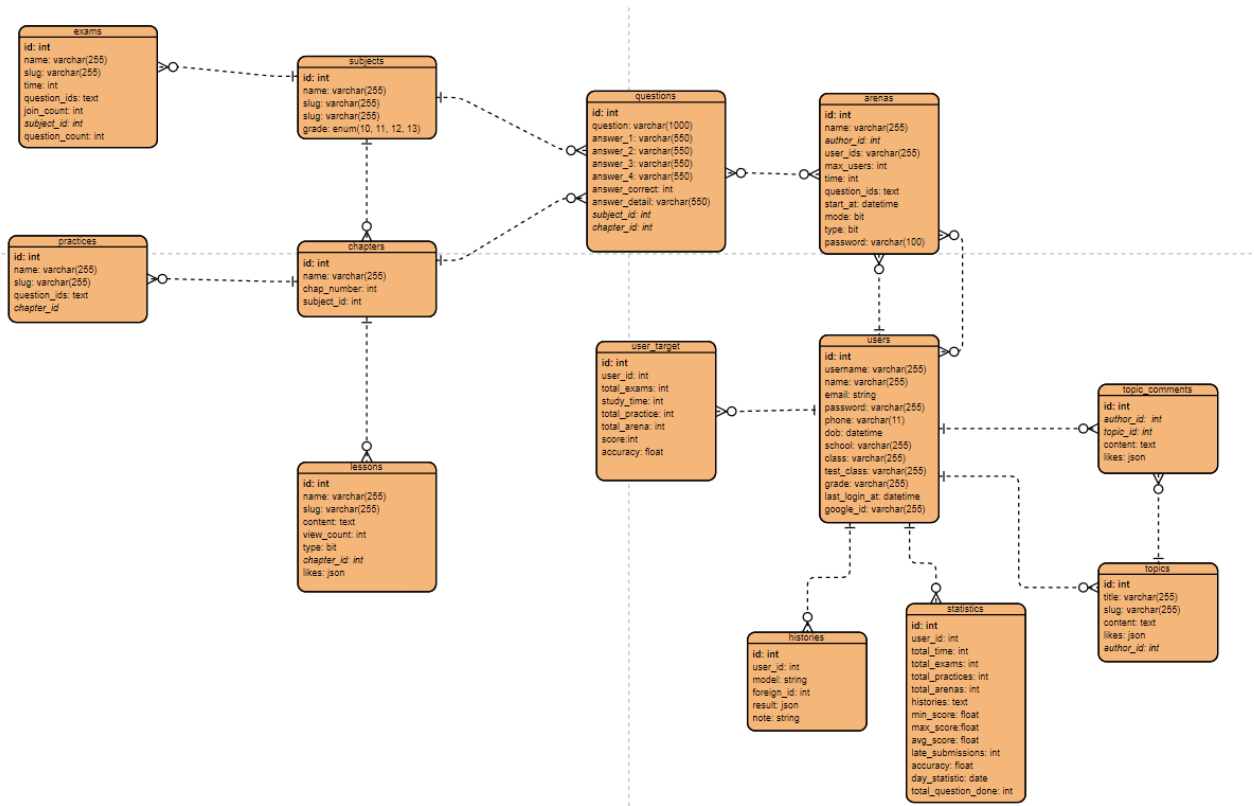
6. Thống kê tình hình ôn luyện của người dùng: Nền tảng theo dõi và cung cấp thông tin chi tiết về quá trình ôn luyện của từng người dùng, bao gồm thời gian học, số lượng bài thi và độ chính xác.

Bảng user_stats: Lưu trữ các thống kê về hoạt động học tập của người dùng.

Bảng histories: Lưu trữ lịch sử ôn luyện và kết quả từng bài thi của người dùng.

2.2.2 Sơ đồ ER

* *ERD Diagram*



Hình 2. 15 Sơ đồ ER

2.2.3 Bảng thực thể cơ sở dữ liệu

Bảng 2. 1 Bảng thực thể User

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	username	STRING	
3	name	STRING	
4	email	STRING	
5	password	STRING	
6	phone	STRING	
7	invite_code	STRING	
8	avatar	STRING	
9	gender	ENUM	
10	dob	DATE	
11	address	STRING	

12	school	STRING	
13	class	STRING	
14	test_class	STRING	
15	grade	INT	

Bảng 2. 2 Bảng thực thể Subject

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	name	STRING	
3	slug	STRING	
4	grade	INT	

Bảng 2. 3 Bảng thực thể Chapter

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	name	STRING	
3	subject_id	STRING	

Bảng 2. 4 Bảng thực thể Practice

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	name	STRING	
3	questions	STRING	
4	slug	STRING	
5	subject_id	INT	FK

Bảng 2. 5 Bảng thực thể Lesson

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	name	STRING	
3	chap_id	INT	FK
4	content	STRING	
5	view_count	INT	
6	type	STRING	
7	subject_id	INT	

8	likes	STRING	
---	-------	--------	--

Bảng 2. 6 Bảng thực thể Question

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	question	STRING	
3	answer_1	STRING	
4	answer_2	STRING	
5	answer_3	STRING	
6	answer_4	STRING	
7	answer_correct	INT	
8	answer_detail	TEXT	
9	subject_id	INT	FK
10	chap_id	INT	FK
11	level	INT	

Bảng 2. 7 Bảng thực thể Exam

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	name	STRING	
3	slug	STRING	
4	time	DATETIME	
5	questions	STRING	
6	join_count	INT	
7	complete_count	INT	
8	subject_id	INT	FK

Bảng 2. 8 Bảng thực thể Topic

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	title	STRING	
3	slug	STRING	
4	content	STRING	
5	author	INT	FK

Bảng 2. 9 Bảng thực thể TopicComment

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	topic	STRING	
3	author	INT	FK
4	content	STRING	
5	likes	STRING	

Bảng 2. 10 Bảng thực thể Arena

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	name	STRING	
3	author	INT	FK
4	users	STRING	
5	max_users	INT	
6	time	INT	
7	questions	STRING	
8	start_at	DATETIME	
9	type	BIT	
10	password	STRING	

Bảng 2. 11 Bảng thực thể UserTarget

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	user_id	INT	FK
3	total_time	INT	
4	total_exam	INT	
5	total_practice	INT	
6	total_arena	INT	
7	score	FLOAT	
8	late_submissions	INT	
9	accuracy	FLOAT	

Bảng 2. 12 Bảng thực thể History

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	user_id	INT	FK
3	model	STRING	
4	foreign_id	INT	
5	result	STRING	
6	note	STRING	

Bảng 2. 13 Bảng thực thể DayStatistics

STT	Trường	Kiểu	Ghi chú
1	id	INT	PK
2	user_id	INT	FK
3	total_time	INT	
4	total_exam	INT	
5	total_practice	INT	
6	total_arena	INT	
7	histories	STRING	
8	min_score	FLOAT	
9	max_score	FLOAT	
10	avg_score	FLOAT	
11	late_submissions	INT	
12	accuracy	FLOAT	

CHƯƠNG 3. XÂY DỰNG MÔ HÌNH DỰ ĐOÁN

3.1 Thu thập dữ liệu

3.1.1 Xác định các yếu tố ảnh hưởng đến điểm thi

Để xây dựng mô hình dự đoán kết quả học tập, việc đầu tiên cần làm là phải thu thập dữ liệu để huấn luyện mô hình. Tập dữ liệu này cần xác định và phản ánh rõ những yếu tố có thể ảnh hưởng đến việc dự đoán điểm thi và hiệu suất của mô hình. Ngoài ra tập dữ liệu này cần đảm bảo đủ độ lớn và tính chính xác để mô hình có thể hoạt động đúng.

Dựa trên dữ liệu có thể thu thập để đánh giá trên web, khảo sát ở các diễn đàn học sinh như forum hocmai, thitot, các hội nhóm trên các trang mạng xã hội như Facebook, cùng với đó là các tài liệu nghiên cứu về phương pháp học tập hiệu quả (Hattie, 2009) và các báo cáo đánh giá kết quả học tập của các tổ chức giáo dục uy tín như trung tâm luyện thi hocmai, mclass. Từ đó xác định được các yếu tố chính sau đây có thể ảnh hưởng đến kết quả học tập và điểm thi của thí sinh.

Đầu tiên là những yếu tố chủ quan của thí sinh, bao gồm các thông tin cơ bản được thu thập từ chính người dùng dựa trên thực tế:

Bảng 3. 1 Bảng các yếu tố cá nhân quyết định điểm thi

STT	Thuộc tính	Giải thích
1	Age	Độ tuổi của người dùng
2	Grade	Khối lớp hiện tại của người dùng
3	StudyTime	Thời gian trung bình học mỗi ngày (giờ) (chỉ tính thời gian học chính khóa và bổ túc trên trường)
4	OnlineCourse	Có đăng ký khóa luyện thi trên mạng không
5	SchoolType	Loại trường THPT (non-public, public, spectilized)
6	ClassType	Loại lớp (normal, gifted)
7	SelfStudy	Tần suất tự học (Từ 1 - rất ít đến 5 - rất nhiều)
8	GoingOut	Tần suất đi chơi với bạn bè (Từ 1 - rất ít đến 5 - rất nhiều)
9	Health	Tình trạng sức khỏe chung (Từ 1 - rất kém đến 5 - rất tốt)

10	LatestScore	Điểm trung bình năm học trước đó (thang 0 - 10)
----	-------------	---

Thứ hai là những yếu tố dữ liệu được thu thập từ chính trang web trong đề tài này, dựa trên kết quả ôn luyện thực tế của người dùng trên web:

Bảng 3. 2 Bảng các yếu tố web quyết định điểm thi

STT	Thuộc tính	Giải thích
1	DailyTimeSpent	Thời gian dành cho web mỗi ngày (giờ)
2	DayActive	Số ngày đã hoạt động trên web
3	ResourceAccessed	Số lượng tài nguyên học tập đã truy cập
4	ForumParticipationFrequency	Tần suất tham gia thảo luận trên diễn đàn (từ 1 - rất ít đến 5 - rất tích cực)
5	HelpfulAnswers	Số câu trả lời được đánh giá là hữu ích
6	ExercisesCompleted	Số bài tập đã làm
7	AverageExerciseScore	Điểm trung bình các bài tập
8	TestsCompleted	Số bài kiểm tra đã làm
9	AverageTestScore	Điểm trung bình các bài kiểm tra
10	AverageTestCompletionTime	Thời lượng làm bài trung bình 1 lần thi (%)
11	ArenaParticipation	Số lần tham gia đấu trường
12	ArenaQuestionsAnsweredPercentage	Phần trăm câu hỏi trả lời được trong đấu trường
13	BestOfArenaCount	Số lần top 1 trong đấu trường
14	AverageArenaRanking	Vị trí trung bình trong đấu trường (từ 1 - rất thấp 5 - rất cao)
15	MostTime	Thời gian ôn luyện thường trên web
16	UsageTrend	Xu hướng hoạt động gần đây (increasing - tăng, decreasing - giảm)

17	PredictScore	Điểm thi được dự đoán
----	--------------	-----------------------

Trong đó thì trường PredictScore sẽ là dữ liệu mục tiêu của mô hình dự đoán này. Từ đó suy ra hiệu suất ôn luyện

3.1.2 Phân tích mối quan hệ giữa các yếu tố ảnh hưởng đến kết quả dự đoán

**** Mối Quan Hệ Tích Cực***

Thời gian học tập (StudyTime): Thời gian học tập càng nhiều, khả năng đạt điểm số cao càng lớn. Việc đầu tư nhiều thời gian vào việc học giúp học sinh nắm vững kiến thức và kỹ năng cần thiết để đạt kết quả tốt.

Luyện thi trực tuyến (OnlineCourse): Tham gia các khóa học trực tuyến cung cấp thêm kiến thức và kỹ năng, từ đó cải thiện điểm số. Khóa học trực tuyến thường có nội dung phong phú và đa dạng, hỗ trợ học sinh tiếp cận với nhiều nguồn tài liệu và phương pháp học tập hiện đại.

Loại Trường và Lớp Học (SchoolType & ClassType): Loại trường và lớp học có thể ảnh hưởng đến môi trường học tập và chất lượng giảng dạy. Các trường và lớp học chất lượng cao thường cung cấp môi trường học tập tốt hơn, giáo viên có trình độ cao hơn, và tài liệu học tập phong phú hơn.

Tần suất tự học (SelfStudy): Tần suất tự học cao thể hiện sự chăm chỉ và chủ động trong việc học, có thể dẫn đến điểm số tốt hơn. Tự học giúp học sinh rèn luyện kỹ năng tự quản lý và khả năng tự giải quyết vấn đề.

Tình trạng sức khỏe của học sinh (Health): Sức khỏe tốt giúp học sinh tập trung và học tập hiệu quả hơn. Một cơ thể khỏe mạnh tạo điều kiện tốt cho việc học tập và nâng cao khả năng tiếp thu kiến thức.

Điểm số trước đó (LatestScore): Điểm số năm trước thường có mối liên hệ với khả năng học tập và điểm số hiện tại. Học sinh có thành tích tốt trong quá khứ thường duy trì được phong độ học tập và đạt điểm cao hơn.

Truy cập tài nguyên học tập (ResourceAccessed): Truy cập nhiều tài nguyên học tập giúp mở rộng kiến thức và cải thiện điểm số. Tài nguyên học tập bao gồm sách vở, bài giảng, video hướng dẫn và các nguồn thông tin trực tuyến.

Tham gia diễn đàn (ForumParticipationFrequency): Tham gia tích cực vào diễn đàn thể hiện sự quan tâm đến việc học và có thể giúp hiểu bài sâu hơn. Thảo luận và chia sẻ kiến thức trên diễn đàn giúp học sinh học hỏi từ các bạn cùng lớp và giải quyết các thắc mắc một cách hiệu quả.

Câu trả lời hữu ích (HelpfulAnswers): Câu trả lời hữu ích chứng tỏ kiến thức vững vàng và khả năng áp dụng kiến thức tốt. Học sinh có khả năng trả lời các câu hỏi một cách chính xác và hiệu quả thường có kiến thức sâu rộng và kỹ năng phân tích tốt.

Hoàn thành bài tập (ExercisesCompleted): Làm nhiều bài tập giúp luyện tập và củng cố kiến thức. Bài tập là phương pháp rèn luyện kỹ năng và kiểm tra lại kiến thức đã học, giúp học sinh làm quen với các dạng bài tập và câu hỏi khác nhau.

Điểm trung bình bài tập (AverageExerciseScore): Điểm trung bình bài tập cao cho thấy khả năng làm bài tốt và hiểu bài sâu. Học sinh đạt điểm cao trong các bài tập thường có khả năng nắm vững kiến thức và áp dụng chúng một cách hiệu quả.

Hoàn thành bài kiểm tra (TestsCompleted): Làm nhiều bài kiểm tra giúp làm quen với dạng đề và áp lực thi cử. Kiểm tra là phương pháp đánh giá khả năng học tập và sự chuẩn bị cho các kỳ thi chính thức.

Điểm trung bình bài kiểm tra (AverageTestScore): Điểm trung bình bài kiểm tra phản ánh trực tiếp khả năng làm bài thi. Học sinh có điểm trung bình cao trong các bài kiểm tra thường có khả năng ứng dụng kiến thức và kỹ năng làm bài thi tốt.

Thời gian hoàn thành bài kiểm tra (AverageTestCompletionTime): Hoàn thành bài kiểm tra nhanh với điểm số cao cho thấy khả năng làm bài tốt và tốc độ xử lý thông tin nhanh. Học sinh có kỹ năng quản lý thời gian và làm bài hiệu quả thường đạt kết quả tốt hơn.

Tham gia đấu trường (ArenaParticipation): Tham gia đấu trường giúp luyện tập kỹ năng làm bài dưới áp lực thời gian. Đấu trường là môi trường học tập cạnh tranh, giúp học sinh rèn luyện kỹ năng làm bài nhanh và chính xác.

Tỷ lệ trả lời đúng trong đấu trường (ArenaQuestionsAnsweredPercentage): Tỷ lệ trả lời đúng cao trong đấu trường cho thấy khả năng xử lý câu hỏi nhanh và chính xác. Học sinh có tỷ lệ trả lời đúng cao thường có kiến thức vững vàng và kỹ năng làm bài tốt.

Số lần đạt top 1 trong đấu trường (BestOfArenaCount): Thường xuyên đạt top 1 trong đấu trường cho thấy khả năng vượt trội so với các bạn cùng học. Học sinh đạt top 1 thường có kỹ năng làm bài xuất sắc và kiến thức sâu rộng.

Xếp hạng trung bình trong đấu trường (AverageArenaRanking): Xếp hạng trung bình cao trong đấu trường thể hiện sự ổn định trong việc làm bài và kiến thức vững vàng. Học sinh có xếp hạng trung bình cao thường duy trì được phong độ học tập và đạt kết quả tốt.

Số ngày đã hoạt động trên web (DayActive): Số ngày hoạt động trên web nhiều có thể thể hiện sự chăm chỉ và quan tâm đến việc học. Học sinh dành nhiều thời gian trên web học tập thường có ý thức tự giác và nỗ lực trong học tập.

Xu hướng hoạt động gần đây (Increase): Hoạt động gần đây tăng lên có thể cho thấy sự nỗ lực và tiến bộ trong học tập. Học sinh có xu hướng hoạt động tăng thường có động lực và quyết tâm cao trong việc đạt kết quả tốt.

** Môi Quan Hệ Tiêu Cực*

Tần suất đi chơi (GoingOut): Dành nhiều thời gian đi chơi có thể ảnh hưởng đến thời gian học tập và kết quả học tập. Việc thiếu thời gian học tập do đi chơi nhiều có thể dẫn đến việc không nắm vững kiến thức và giảm điểm số.

Xu hướng hoạt động gần đây (Decrease): Hoạt động gần đây giảm xuống có thể là dấu hiệu của việc lơ là học tập. Học sinh có xu hướng hoạt động giảm thường thiếu sự chăm chỉ và quan tâm đến việc học, dẫn đến kết quả học tập kém.

** Có Thể Tăng Hoặc Giảm*

Độ tuổi (Age): Độ tuổi có thể ảnh hưởng đến sự trưởng thành và khả năng học tập, nhưng không phải yếu tố có sức ảnh hưởng quá lớn. Học sinh lớn tuổi hơn có thể có kỹ năng học tập tốt hơn, nhưng cũng có thể chịu áp lực từ nhiều yếu tố khác.

Khối lớp (Grade): Khối lớp có thể liên quan đến độ khó của kiến thức, nhưng không phải yếu tố có sức ảnh hưởng quá lớn. Học sinh ở các khối lớp khác nhau có thể gặp các thử thách khác nhau trong việc học tập.

Thời gian sử dụng trang Web mỗi ngày (DailyTimeSpent): Thời gian dành cho web mỗi ngày có thể có cả tác động tích cực (học tập) và tiêu cực (giải trí). Học sinh sử dụng web để học tập có thể cải thiện điểm số, nhưng nếu sử dụng quá nhiều cũng sẽ không tốt.

Thời gian học chủ yếu trong ngày (MostTime): Một số nghiên cứu cho thấy con người có thể tập trung và sáng tạo tốt hơn vào ban đêm, tuy nhiên nhịp sinh học là chu kỳ tự nhiên của cơ thể nên yếu tố này sẽ không ảnh hưởng quá nhiều đến điểm thi. Học sinh cần tìm thời gian học tập phù hợp với nhịp sinh học của mình để đạt hiệu quả tốt nhất.

3.2 Tiền xử lý dữ liệu

Sau khi hoàn thành quá trình thu thập tập dữ liệu mẫu, bước tiếp theo là tiến hành xử lý tập dữ liệu này. Các bước tiền xử lý dữ liệu thông thường bao gồm: xử lý dữ liệu bị thiếu (missing values), mã hóa dữ liệu dạng chữ sang dạng số (categorical data encoding), chuẩn hóa dữ liệu (data normalization/standardization), loại bỏ các ngoại lệ (outlier removal), chia tách dữ liệu (data splitting), giảm số chiều của dữ liệu (dimensionality reduction), và áp dụng các kỹ thuật phân tích đặc trưng (feature engineering/feature scaling).

Bước 1: Xử lý dữ liệu thiếu

Xử lý dữ liệu thiếu là một bước quan trọng trong quá trình tiền xử lý dữ liệu. Các phương pháp phổ biến để xử lý dữ liệu thiếu bao gồm loại bỏ các mẫu hoặc thuộc tính chứa giá trị thiếu, hoặc áp dụng các kỹ thuật thay thế giá trị thiếu bằng phương pháp trung bình, phương pháp trung vị, hoặc các kỹ thuật suy luận tiên tiến hơn. Điều này đảm bảo rằng dữ liệu đầu vào không bị sai lệch do thiếu thông tin.

```

# Kiểm tra dữ liệu thiếu
missing_values = data.isnull().sum()
print(missing_values)

```

[3] ✓ 0.4s

Id	0
Age	0
Grade	0
StudyTime	0
OnlineCourse	0
SchoolType	0
ClassType	0
SelfStudy	0
GoingOut	0
Health	0
LatestScore	0
DailyTimeSpent	0
DayActive	0
ResourceAccessed	0
ForumParticipationFrequency	0
HelpfulAnswers	0
ExercisesCompleted	0
AverageExerciseScore	0
TestsCompleted	0
AverageTestScore	0
AverageTestCompletionTime	0
ArenaParticipation	0
ArenaQuestionsAnsweredPercentage	0
BestOfArenaCount	0
AverageArenaRanking	0
MostTime	0
PredictScore	0

dtype: int64

Hình 3. 1 Kiểm tra dữ liệu thiếu

Tập dữ liệu không chứa giá trị thiếu, bước này có thể được bỏ qua.

Bước 2: Mã hóa các biến phân loại

Mã hóa dữ liệu phân loại sang dạng số là cần thiết để các mô hình học máy có thể xử lý. Các kỹ thuật mã hóa bao gồm mã hóa nhãn (label encoding) và mã hóa một-nóng (one-hot encoding). Việc sử dụng LabelEncoder từ thư viện sklearn.preprocessing để mã hóa các biến phân loại thành các giá trị số giúp cho việc xử lý dữ liệu trở nên hiệu quả và phù hợp hơn với các mô hình học máy.

```

# Mã hóa các biến phân loại
categorical_columns = ['OnlineCourse', 'SchoolType', 'ClassType', 'MostTime']
label_encoders = {}
for col in categorical_columns:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le
print("\nDữ liệu sau khi mã hóa các biến phân loại:")
print(data[categorical_columns])

```

[32] ✓ 0.1s

...

Dữ liệu sau khi mã hóa các biến phân loại:

	OnlineCourse	SchoolType	ClassType	MostTime
0	1	1	0	1
1	0	2	1	1
2	1	0	0	1
3	0	2	0	1
4	0	2	0	0
...

Hình 3. 2 Mã hóa các biến phân loại dạng text sang number

Bước 3: Chuẩn hóa và loại bỏ ngoại lệ

Chuẩn hóa dữ liệu (normalization/standardization) là quá trình đưa các thuộc tính về cùng một thang đo nhằm cải thiện hiệu quả và tốc độ hội tụ của mô hình, đặc biệt là với các thuật toán nhạy cảm với tỷ lệ đặc trưng như Ridge Regression. Loại bỏ các giá trị ngoại lệ bằng cách sử dụng điểm z-score giúp giảm thiểu tác động của các giá trị bất thường lên mô hình, từ đó nâng cao độ chính xác của dự đoán.

```

# Chuẩn hóa các đặc trưng và xóa bỏ ngoại lệ
numerical_columns = ['Age', 'Grade', 'StudyTime', 'SelfStudy', 'GoingOut', 'Health', 'AverageTestScore', 'AverageExerciseScore', 'LatestScore', 'TestsCompleted']
scaler = StandardScaler()
data[numerical_columns] = scaler.fit_transform(data[numerical_columns])

data = data[(np.abs(zscore(data[numerical_columns]))) < 3].all(axis=1)]
print(data.head())

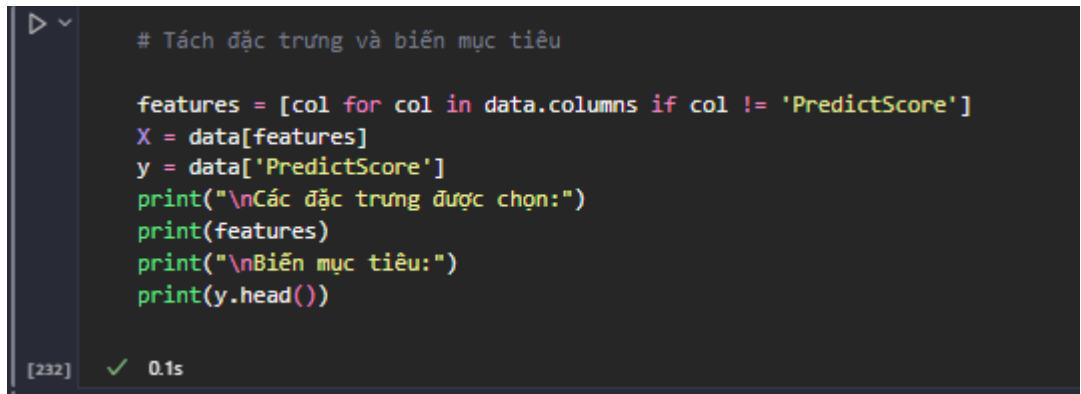
```

✓ 0.6s

Hình 3. 3 Chuẩn hóa và loại bỏ ngoại lệ

Bước 4: Tách các đặc trưng và biến mục tiêu

Việc tách các đặc trưng (features) và biến mục tiêu (target) là bước cần thiết để xác định rõ ràng các biến độc lập và biến phụ thuộc trong mô hình học máy. Các đặc trưng được lựa chọn sẽ được đưa vào mô hình, trong khi biến mục tiêu là đối tượng cần dự đoán.



```
# Tách đặc trưng và biến mục tiêu

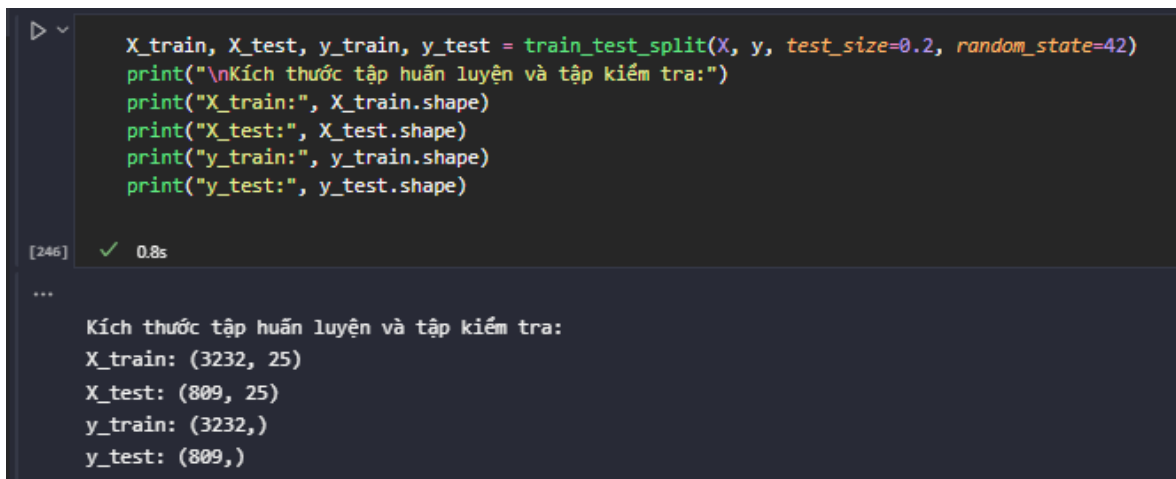
features = [col for col in data.columns if col != 'PredictScore']
X = data[features]
y = data['PredictScore']
print("\nCác đặc trưng được chọn:")
print(features)
print("\nBiến mục tiêu:")
print(y.head())
```

[232] ✓ 0.1s

Hình 3. 4 Tách đặc trưng và biến mục tiêu

Bước 5: Chia tách dữ liệu

Chia tách dữ liệu thành các tập huấn luyện và tập kiểm tra với tỷ lệ thông thường là 80-20 là một bước quan trọng để đánh giá hiệu suất của mô hình trên dữ liệu chưa từng thấy trước đó. Điều này giúp đảm bảo rằng mô hình không bị overfit vào tập huấn luyện và có thể tổng quát hóa tốt trên các dữ liệu mới.



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print("\nKích thước tập huấn luyện và tập kiểm tra:")
print("X_train:", X_train.shape)
print("X_test:", X_test.shape)
print("y_train:", y_train.shape)
print("y_test:", y_test.shape)
```

[246] ✓ 0.8s

```
...
Kích thước tập huấn luyện và tập kiểm tra:
X_train: (3232, 25)
X_test: (809, 25)
y_train: (3232,)
y_test: (809,)
```

Hình 3. 5 Chia tập dữ liệu

Mỗi bước trong quy trình tiền xử lý dữ liệu đều đóng vai trò quan trọng trong việc chuẩn bị dữ liệu cho mô hình học máy. Việc xử lý dữ liệu thiếu, mã hóa dữ liệu phân loại, chuẩn hóa và loại bỏ ngoại lệ, chọn đặc trưng và chia tách dữ liệu đều góp phần làm cho dữ liệu phù hợp hơn với các thuật toán học máy, từ đó cải thiện hiệu suất và độ chính xác của mô hình dự đoán.

3.3 Ứng dụng thuật toán Random Forest

Rừng ngẫu nhiên là một thuật toán học máy sử dụng phương pháp Bagging để kết hợp nhiều cây quyết định nhằm cải thiện độ chính xác và giảm thiểu overfitting.

Bước 1: Bootstrap Sampling

Giả sử tập dữ liệu ban đầu có N mẫu với các đặc trưng (features) như sau: Age, Grade, StudyTime, OnlineCourse, SchoolType, LatestScore, ClassType, SelfStudy, GoingOut, Health, AverageTestScore, AverageExerciseScore, DailyTimeSpent, ArenaQuestionsAnsweredPercentage, TestsCompleted, MostTime. Và biến mục tiêu là PredictScore.

Giả sử $N=100$ (số lượng mẫu) và $M=16$ (số lượng đặc trưng). Để tạo ra một tập bootstrap, chúng ta lấy mẫu ngẫu nhiên N lần với thay thế từ tập dữ liệu ban đầu. Ví dụ, giả sử sau khi bootstrap sampling, chúng ta có tập con sau:

Bảng 3. 3 Ví dụ về tập dữ liệu

Age	Grade	StudyTime	...	TestsCompleted	PredictScore
17	11	4	...	34	9.2
16	10	3	...	29	8.5
17	11	4	...	34	9.2
...
18	12	5	...	38	9.7

Bước 2: Xây Dựng Cây Quyết Định từ Bootstrap Samples

Tại mỗi nút của cây quyết định, thay vì xem xét tất cả M đặc trưng, chúng ta chỉ xem xét một tập hợp con ngẫu nhiên của các đặc trưng (giả sử $m = \sqrt{M}$).

Giả sử tại một nút, chúng ta chọn ngẫu nhiên 4 đặc trưng từ 16 đặc trưng. Để chọn đặc trưng tốt nhất để phân chia, chúng ta sử dụng các tiêu chí như Gini Impurity hoặc Information Gain.

* Gini Impurity

Giả sử chúng ta đang phân chia dựa trên đặc trưng AverageTestScore, chúng ta tính Gini Impurity cho từng giá trị ngưỡng t :

$$\text{Gini}(S) = 1 - \sum_{i=1}^e p_i^2$$

Trong đó p_i là tỷ lệ phần tử thuộc lớp i trong tập dữ liệu S .

Ví dụ, giả sử chúng ta có 4 giá trị ngưỡng để phân chia `AverageTestScore`: 7.0, 8.0, 9.0, và 10.0. Cần phải tính Gini Impurity cho từng giá trị này:

$$\text{Gini}(S \mid \text{AverageTestScore} < 8.0) = 1 - (0.6)^2 - (0.4)^2 = 0.48$$

$$\text{Gini}(S \mid \text{AverageTestScore} \geq 8.0) = 1 - (0.7)^2 - (0.3)^2 = 0.42$$

Tính tổng Gini Impurity:

$$\begin{aligned} \text{Gini}_{\text{split}} = & \left(\frac{n_{<8.0}}{n} \right) * \text{Gini}(S \mid \text{AverageTestScore} < 8.0) \\ & + \left(\frac{n_{\geq 8.0}}{n} \right) * \text{Gini}(S \mid \text{AverageTestScore} \geq 8.0) \end{aligned}$$

Chọn đặc trưng và giá trị ngưỡng có Gini Impurity nhỏ nhất để phân chia dữ liệu.

Bước 3: Xây Dựng Nhiều Cây Quyết Định

Lặp lại bước 1 và bước 2 để xây dựng nhiều cây quyết định. Giả sử chúng ta xây dựng 100 cây quyết định (tức là $n_{\text{estimators}} = 100$).

Bước 4: Tổng Hợp Kết Quả từ Nhiều Cây Quyết Định

Sau khi xây dựng xong rừng ngẫu nhiên, chúng ta dự đoán bằng cách tổng hợp kết quả từ tất cả các cây quyết định. Với bài toán hồi quy, kết quả dự đoán là trung bình của tất cả các dự đoán từ các cây quyết định:

$$\hat{y} = \sum_{t=1}^T \hat{y}_t$$

Trong đó T là số lượng cây quyết định và \hat{y}_t là dự đoán từ cây quyết định thứ t .

Ví dụ, giả sử 100 cây quyết định đưa ra các dự đoán như sau:

$$\hat{y}^1 = 9.0, \hat{y}_2 = 8.8, \dots, \hat{y}_{100} = 9.2$$

Kết quả dự đoán cuối cùng sẽ là trung bình của tất cả các dự đoán này:

$$\hat{y} = \left(\frac{1}{100} \right) * \sum_{t=1}^{100} \hat{y}_t = 9.0$$

Tóm tắt các bước để xây dựng Random Forest:

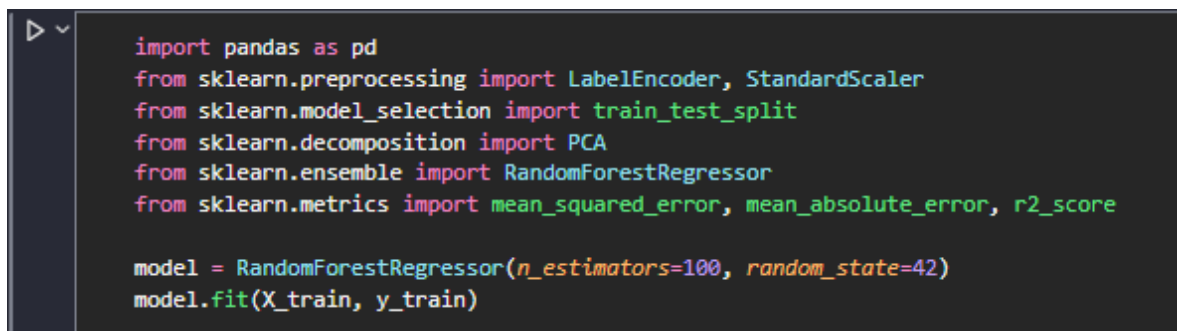
- Bootstrap Sampling: Lấy mẫu ngẫu nhiên với thay thế từ tập dữ liệu ban đầu để tạo ra nhiều tập dữ liệu con.

- Xây Dựng Cây Quyết Định: Với mỗi tập dữ liệu con, xây dựng một cây quyết định bằng cách sử dụng một tập hợp con ngẫu nhiên của các đặc trưng tại mỗi nút để chọn đặc trưng tốt nhất để phân chia.
- Tổng Hợp Kết Quả: Dự đoán điểm thi bằng cách lấy trung bình các dự đoán từ tất cả các cây quyết định.

3.4 Xây dựng và đánh giá mô hình

3.4.1 Xây dựng và huấn luyện mô hình

Scikit-learn là một công cụ mạnh mẽ và linh hoạt cho việc xây dựng mô hình học máy, đặc biệt là trong việc triển khai mô hình Random Forest. Việc sử dụng scikit-learn giúp tối ưu hóa quá trình xây dựng mô hình, từ chuẩn bị dữ liệu đến đào tạo mô hình, với tính dễ dàng và hiệu quả cao. Bên cạnh đó, việc tận dụng các tính năng linh hoạt và tài liệu phong phú của scikit-learn giúp người dùng tùy chỉnh và tinh chỉnh mô hình theo nhu cầu cụ thể nào đó.



```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

Hình 3. 6 Huấn luyện mô hình với Random Forest trong Sklearn

Đoạn mã trên là những bước cốt lõi trong quá trình xây dựng và huấn luyện mô hình Random Forest sử dụng thư viện scikit-learn, một trong những công cụ phổ biến cho việc phát triển các mô hình học máy. Mã này đóng vai trò quan trọng trong việc tạo ra một mô hình có khả năng dự đoán chính xác từ dữ liệu huấn luyện.

Đầu tiên, trong đoạn mã đầu tiên, chúng ta khởi tạo một mô hình Random Forest bằng cách sử dụng lớp RandomForestRegressor từ thư viện scikit-learn. Random Forest là một phương pháp học máy cơ bản, là một tập hợp của nhiều cây quyết định, mỗi cây đưa ra một dự đoán riêng và kết quả cuối cùng là kết hợp của các dự đoán này.

Thông qua tham số `n_estimators=100`, chúng ta xác định số lượng cây quyết định sẽ được tạo ra trong mô hình. Điều này giúp tăng tính đa dạng và ổn định của mô hình, giảm

thiếu hiện tượng quá mức phù hợp (overfitting) và nâng cao khả năng tổng quát hóa của mô hình.

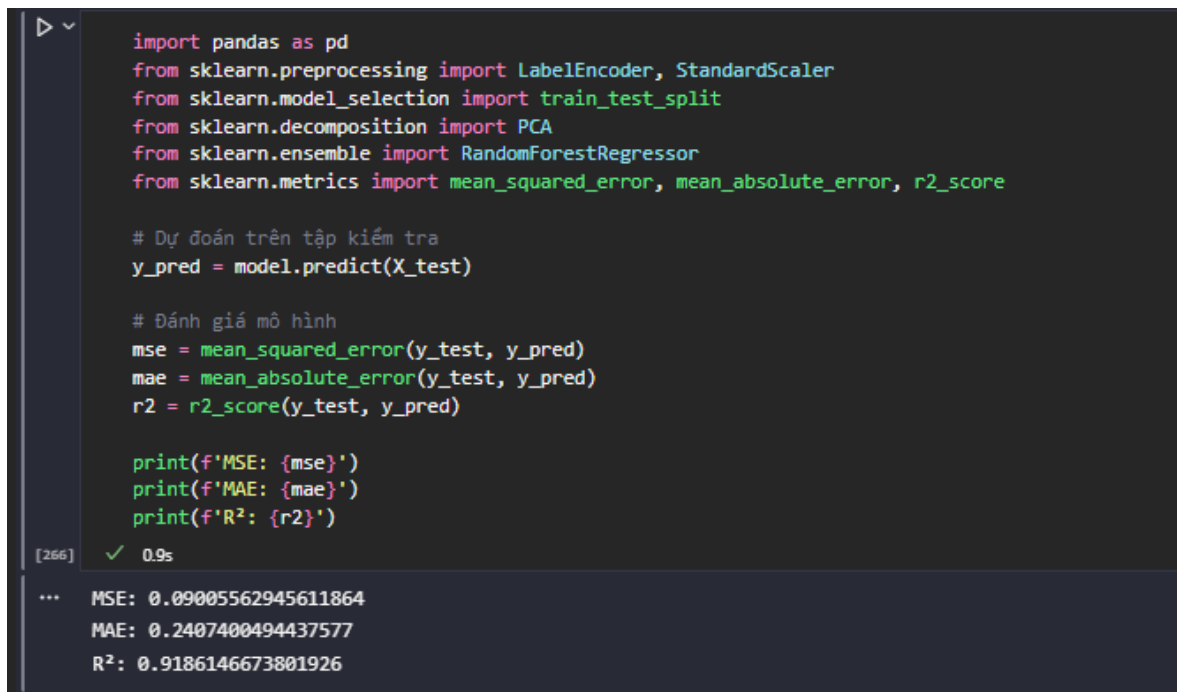
Tham số `random_state=42` đảm bảo tính tái lập của quá trình huấn luyện mô hình. Bằng cách cố định giá trị `random_state`, chúng ta có thể đảm bảo rằng mỗi lần chạy mã, chúng ta sẽ nhận được kết quả nhất quán, từ đó dễ dàng so sánh và đánh giá các mô hình khác nhau.

Tiếp theo, trong đoạn mã thứ hai, chúng ta tiến hành huấn luyện mô hình Random Forest trên tập dữ liệu huấn luyện, được đại diện bởi ma trận đặc trưng `X_train` và vector mục tiêu `y_train`. Phương thức `fit()` được gọi để bắt đầu quá trình huấn luyện.

Trong quá trình huấn luyện, scikit-learn tự động thực hiện các bước quan trọng như lấy mẫu Bootstrap và xây dựng các cây quyết định từ các tập dữ liệu con. Sau khi hoàn thành quá trình huấn luyện, mô hình Random Forest đã sẵn sàng để dự đoán giá trị mục tiêu cho các mẫu dữ liệu mới, đưa ra những dự đoán có tính chất tổng quát và chính xác.

3.4.2 Đánh giá hiệu suất của mô hình

Sau khi huấn luyện mô hình, chúng ta có thể tiến hành dự đoán trên tập kiểm tra `y_test`, sau đó sử dụng 1 số chỉ số để đánh giá hiệu suất và độ chính xác của mô hình như sai số toàn phương trung bình (MSE), sai số tuyệt đối trung bình và hệ số xác định R^2 .



```

import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# Dự đoán trên tập kiểm tra
y_pred = model.predict(X_test)

# Đánh giá mô hình
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'MSE: {mse}')
print(f'MAE: {mae}')
print(f'R²: {r2}')

```

[266] ✓ 0.9s

```

*** MSE: 0.09005562945611864
    MAE: 0.2407400494437577
    R²: 0.9186146673801926

```

Hình 3. 7 Tính toán giá trị để đánh giá mô hình

* MSE: đo lường độ chênh lệch trung bình bình phương giữa giá trị thực và giá trị dự đoán. Công thức tính MSE là:

$$MSE = \left(\frac{1}{n}\right) * \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Trong đó:

- n là số lượng mẫu.
- y_i là giá trị thực của mẫu thứ i .
- \hat{y}_i là giá trị dự đoán của mẫu thứ i .

MSE càng nhỏ thì mô hình dự đoán càng chính xác. Trong trường hợp này, $MSE = 0.09005562945611864$, tức là sai số trung bình bình phương là rất nhỏ, cho thấy mô hình dự đoán khá chính xác.

* MAE: đo lường độ chênh lệch trung bình tuyệt đối giữa giá trị thực và giá trị dự đoán. Công thức tính MAE là:

$$MAE = \left(\frac{1}{n}\right) * \sum_{i=1}^n |y_i - \hat{y}_i|$$

MAE càng nhỏ thì mô hình dự đoán càng chính xác. Trong trường hợp này, $MAE = 0.2407400494437577$, tức là sai số trung bình tuyệt đối là khá nhỏ, cho thấy mô hình dự đoán khá chính xác.

* R^2 (R-squared): đo lường mức độ giải thích được của mô hình đối với biến mục tiêu. Công thức tính R^2 là:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Trong đó:

- \bar{y} là giá trị trung bình của biến mục tiêu.

R^2 dao động trong khoảng từ 0 đến 1. Giá trị R^2 càng gần 1 thì mô hình dự đoán càng tốt. Trong trường hợp này, $R^2 = 0.9186146673801926$, tức là mô hình giải thích được 91.86% sự biến động của biến mục tiêu, cho thấy mô hình có độ chính xác cao.

CHƯƠNG 4. XÂY DỰNG CHƯƠNG TRÌNH

4.1 Thết kế giao diện

4.2.1 Trang người dùng

4.2.2 Trang Admin

4.2.2.1 Giao diện xác thực

4.2 Tích hợp tính năng dự đoán

<giới thiệu về tính năng dự đoán trong web, kèm hình chụp>

KẾT LUẬN & KIẾN NGHỊ

- 1. Kết quả đạt được**
- 2. Hạn chế**
- 3. Hướng phát triển**

TÀI LIỆU THAM KHẢO

- [1]. Tài liệu chính thức của Laravel 11, <https://laravel.com/docs/11.x/> , truy cập 03/2024
- [2]. Redis with Laravel, <https://www.squash.io/tutorial-on-laravel-redis-integration-in-php/> , truy cập 03/2024
- [3]. Broadcasting with laravel echo, <https://viblo.asia/p/broadcasting-with-laravel-echo-aWj53XB1K6m> , truy cập 05/2024
- [4]. Realtime application with Socket.IO, <https://ably.com/topic/socketio> , truy cập 05/2024
- [5]. Tài liệu chính thức của NextJS, <https://nextjs.org/docs>, truy cập 05/2024
- [6]. Bài toán dự đoán, <https://viblo.asia/p/bai-toan-du-doan-prediction-dua-tren-mo-hinh-hoi-quy-trong-machine-learning-YmjeoLgzkqa> , truy cập 03/2024
- [7]. Introduction to Machine Learning with Python (Andreas C. Müller & Sarah Guido), 2017
- [8]. Data Preprocessing in Data Mining, <https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/> , truy cập 04/2024
- [9]. Scikit-learn: Machine Learning in Python, https://scikit-learn.org/stable/supervised_learning.html, truy cập 04/2024
- [10]. "Random Forest Regression", by Towards Data Science, <https://towardsdatascience.com/random-forest-regression-209c0f354c84> , truy cập 05/2024.
- [11]. Evaluating Regression Models, <https://www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/> , truy cập 05/2024