

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN

UTH

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

NGHIÊN CỨU VÀ XÂY DỰNG HỆ  
THỐNG QUẢN LÝ HỌC TẬP THÔNG  
MINH (SMART LMS) DỰA TRÊN  
KIẾN TRÚC RAG VÀ MÔ HÌNH  
NGÔN NGỮ LỚN

Giảng viên hướng dẫn: TS. Nguyễn Văn B

Sinh viên thực hiện: Nguyễn Quốc Tùnggggg

Mã số sinh viên: 2251120259

Lớp: CN22E

TP. HỒ CHÍ MINH, NĂM 2025

## **LỜI CAM ĐOAN**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi dược sự hướng dẫn của giảng viên hướng dẫn. Các kết quả nêu trong luận văn là trung thực và chưa từng được công bố trong bất kỳ công trình nào khác. Mọi sự giúp đỡ cho việc thực hiện luận văn này đã được cảm ơn và các thông tin trích dẫn trong luận văn đã được chỉ rõ nguồn gốc.

## LỜI MỞ ĐẦU

Sự bùng nổ của Trí tuệ nhân tạo tạo sinh (Generative AI) đã mở ra những cơ hội chưa từng có trong việc cá nhân hóa giáo dục. Tuy nhiên, các hệ thống Quản lý học tập (LMS) hiện nay tại Việt Nam chủ yếu vẫn đóng vai trò là kho lưu trữ tài liệu tĩnh, thiếu đi tính tương tác thông minh và khả năng hỗ trợ học sinh học tập chủ động.

Dự án này được thực hiện với mong muốn xây dựng một hệ sinh thái học tập "Smart LMS" dành cho cấp trung học. Điểm đột phá của hệ thống là việc ứng dụng kỹ thuật Retrieval-Augmented Generation (RAG) để tạo ra một Trợ lý ảo (AI Tutor) có khả năng hiểu sâu sắc nội dung bài giảng của giáo viên, từ đó phản hồi chính xác thắc mắc của học sinh. Bên cạnh đó, hệ thống tích hợp các công cụ giám sát thi cử dựa trên AI nhằm đảm bảo tính liêm chính trong môi trường học tập trực tuyến.

# MỤC LỤC

<b>Lời cam đoan</b>	i
<b>Lời mở đầu</b>	ii
<b>1 MỞ ĐẦU</b>	1
1.1 Lý do chọn đề tài . . . . .	1
1.2 Mục tiêu nghiên cứu . . . . .	1
1.3 Đối tượng và phạm vi nghiên cứu . . . . .	1
<b>2 CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ</b>	2
2.1 Tổng quan nền tảng công nghệ . . . . .	2
2.2 Kiến trúc tổng thể hệ thống . . . . .	2
2.3 Kiến trúc Web: Next.js App Router và React . . . . .	3
2.4 Xác thực và phân quyền: NextAuth.js . . . . .	3
2.4.1 Phân quyền theo vai trò (RBAC) và bảo vệ truy cập . . . . .	4
2.5 Tầng dữ liệu: Prisma ORM và PostgreSQL . . . . .	4
2.6 Cơ sở dữ liệu vector: pgvector . . . . .	4
2.7 Lưu trữ tệp: Supabase Storage . . . . .	4
2.8 Mô hình ngôn ngữ lớn (LLM) và Gemini API . . . . .	5
2.9 Kiến trúc RAG (Retrieval-Augmented Generation) . . . . .	5
2.10 Ứng dụng AI trong giám sát thi . . . . .	5
<b>3 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG</b>	6
3.1 Giới thiệu chương . . . . .	6
3.2 Phân tích hệ thống . . . . .	6
3.2.1 Đối tượng sử dụng (Actor) và vai trò (Role) . . . . .	6
3.2.2 Yêu cầu chức năng . . . . .	6
3.2.3 Yêu cầu phi chức năng . . . . .	7
3.3 Thiết kế hệ thống . . . . .	7
3.3.1 Thiết kế kiến trúc . . . . .	7
3.3.2 Thiết kế Use Case . . . . .	8
3.3.3 Thiết kế cơ sở dữ liệu . . . . .	10
<b>4 HIỆN THỰC HÓA HỆ THỐNG</b>	13
4.1 Xây dựng trợ lý ảo AI Tutor . . . . .	13
4.2 Hệ thống đánh giá và giám sát thi cử . . . . .	13
4.2.1 Cơ chế tự động chấm bài . . . . .	13
4.2.2 Phát hiện hành vi đáng ngờ . . . . .	13
4.3 Giao diện người dùng (UI/UX) . . . . .	13

<b>5 ĐÁNH GIÁ VÀ THỬ NGHIỆM</b>	<b>14</b>
5.1 Đánh giá độ chính xác của AI Tutor . . . . .	14
5.2 Đánh giá hiệu năng hệ thống . . . . .	14
<b>6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN</b>	<b>15</b>
6.1 Kết quả đạt được . . . . .	15
6.2 Hướng phát triển tương lai . . . . .	15
<b>Tài liệu tham khảo</b>	<b>16</b>

## **DANH SÁCH HÌNH VẼ**

3.1	Sơ đồ bối cảnh hệ thống EduVerse . . . . .	7
3.2	Kiến trúc tổng thể hệ thống EduVerse . . . . .	8
3.3	Use Case tổng thể hệ thống EduVerse . . . . .	9
3.4	Use Case tổng quan module giám sát thi và chống gian lận . . . . .	10
3.5	Sơ đồ cơ sở dữ liệu cốt lõi của hệ thống EduVerse . . . . .	11

## **DANH SÁCH BẢNG**

2.1	Tóm tắt công nghệ sử dụng trong hệ thống EduVerse . . . . .	2
2.2	So sánh RAG và LLM thuần trong bối cảnh hệ thống học tập . . . . .	3
3.1	Các bảng dữ liệu cốt lõi trong hệ thống EduVerse . . . . .	12

# CHƯƠNG 1

## MỞ ĐẦU

### 1.1 Lý do chọn đề tài

Trong kỷ nguyên giáo dục 4.0, vai trò của giáo viên đang dịch chuyển từ người truyền thụ kiến thức sang người điều phối học tập. Tuy nhiên, khối lượng công việc hành chính và việc theo sát từng học sinh trong lớp học trực tuyến là một thách thức lớn. Hệ thống LMS thông minh cần phải giải quyết được bài toán giảm tải cho giáo viên và tăng tính tự học cho học sinh.

### 1.2 Mục tiêu nghiên cứu

- Thiết kế kiến trúc hệ thống LMS đa vai trò (Admin, Teacher, Student, Parent).
- Hiện thực hóa module AI Tutor hỗ trợ học tập dựa trên cơ sở dữ liệu vector.
- Xây dựng quy trình tự động hóa đánh giá năng lực học sinh qua AI.
- Đảm bảo an ninh và tính liêm chính học thuật qua hệ thống chống gian lận.

### 1.3 Đối tượng và phạm vi nghiên cứu

- **Đối tượng:** Quy trình quản lý và hỗ trợ học tập tại các trường trung học.
- **Phạm vi:** Xây dựng ứng dụng Web dựa trên Next.js và tích hợp Google Gemini API.

## CHƯƠNG 2

### CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

#### 2.1 Tổng quan nền tảng công nghệ

Hệ thống **EduVerse** được hiện thực hóa dưới dạng ứng dụng Web theo mô hình full-stack, trong đó lớp giao diện và lớp xử lý nghiệp vụ (API) được tổ chức thống nhất. Nhóm công nghệ trọng tâm gồm: Next.js (App Router) cho kiến trúc web hiện đại [1], NextAuth.js cho xác thực và quản lý phiên đăng nhập [2], Prisma ORM cho truy cập dữ liệu [3], PostgreSQL cho lưu trữ quan hệ [4], pgvector cho truy xuất tương đồng vector phục vụ RAG [5], Supabase Storage cho lưu trữ tệp nộp bài [6], và Gemini API cho các tác vụ AI (embedding và sinh nội dung) [8].

Công nghệ	Vai trò trong hệ thống	Lý do lựa chọn
Next.js (App Router) [1]	Full-stack Web framework; tổ chức routing, rendering, API Routes	Hỗ trợ kiến trúc hiện đại, tối ưu hiệu năng và bảo mật khi triển khai logic phía server
NextAuth.js [2]	Xác thực người dùng, quản lý phiên đăng nhập	Tích hợp đăng nhập linh hoạt, đồng thời hỗ trợ quản lý session/role cho phân quyền
Prisma ORM [3]	Truy cập dữ liệu type-safe, migrations, ánh xạ mô hình dữ liệu	Giảm lỗi truy vấn, tăng tính nhất quán và hỗ trợ mở rộng schema có kiểm soát
PostgreSQL [4]	Cơ sở dữ liệu quan hệ lưu trữ dữ liệu nghiệp vụ	Đảm bảo ACID, ổn định và phù hợp dữ liệu có quan hệ phức tạp (khóa học, bài tập, sự kiện thi)
pgvector [5]	Lưu embedding và truy vấn tương đồng vector phục vụ RAG	Cho phép truy xuất ngữ nghĩa ngay trên Postgres, thông nhất hạ tầng dữ liệu
Supabase Storage [6]	Lưu trữ tệp nộp bài, cấp signed URL	Tách tài liệu khỏi DB, hỗ trợ truy cập an toàn theo thời hạn
Gemini API [8]	Tạo embedding và sinh nội dung (AI Tutor, tóm tắt hỗ trợ)	Khả năng sinh ngôn ngữ tự nhiên và tạo embedding phục vụ truy xuất/giải thích

Bảng 2.1: Tóm tắt công nghệ sử dụng trong hệ thống EduVerse

#### 2.2 Kiến trúc tổng thể hệ thống

Về mặt kiến trúc, EduVerse có thể xem như một hệ thống gồm bốn lớp chính:

- **Lớp giao diện (Presentation):** các trang dashboard cho học sinh/giáo viên, tập trung vào thao tác học tập, làm bài và theo dõi kết quả.
- **Lớp dịch vụ ứng dụng (Application/API):** các API Routes xử lý nghiệp vụ (xác thực, nộp bài, ghi sự kiện thi, gọi AI Tutor, tóm tắt chống gian lận), đóng vai trò ranh giới

Tiêu chí	RAG [9]	LLM thuần
Nguồn tri thức	Dựa trên tài liệu nội bộ (bài giảng) được truy xuất theo câu hỏi	Chủ yếu dựa trên tri thức đã học trong quá trình huấn luyện
Tính kiểm chứng	Có thể truy vết theo đoạn truy xuất; dễ giải thích tại sao trả lời như vậy	Khó truy vết nguồn; dễ sinh nội dung không có trong tài liệu
Rủi ro "ảo giác"	Thấp hơn do bị ràng buộc bởi ngữ cảnh truy xuất	Cao hơn, đặc biệt với câu hỏi đặc thù nội dung lớp học
Chi phí cập nhật tri thức	Cập nhật bằng cách thay tài liệu/embedding (không cần huấn luyện lại)	Thường cần fine-tune hoặc chờ model mới để cập nhật tri thức
Phù hợp giáo dục	Phù hợp khi cần bám sát học liệu và chuẩn hóa nội dung	Phù hợp cho gợi ý chung, nhưng cần kiểm soát chặt khi dùng như nguồn kiến thức

Bảng 2.2: So sánh RAG và LLM thuần trong bối cảnh hệ thống học tập

bảo mật và kiểm soát truy cập.

- **Lớp dữ liệu (Data):** PostgreSQL lưu trữ dữ liệu quan hệ, kết hợp pgvector để truy vấn vector; Supabase Storage lưu tệp đính kèm.
- **Lớp dịch vụ AI (External AI Service):** Gemini API đảm nhiệm embedding và sinh phản hồi.

Trong bối cảnh Next.js, các API Routes và logic truy xuất dữ liệu được thực thi phía server, giúp đảm bảo các tác vụ nhạy cảm (khoá API, truy vấn dữ liệu) không bị lộ ra phía client [1].

### 2.3 Kiến trúc Web: Next.js App Router và React

Next.js App Router cung cấp cơ chế định tuyến theo thư mục và hỗ trợ Server Components, giúp tách bạch phần render phía server và tương tác phía client. Trong EduVerse, API Routes được dùng để đóng gói các nghiệp vụ nhạy cảm (ví dụ gọi AI, truy cập cơ sở dữ liệu), qua đó hạn chế lộ khóa dịch vụ và giảm tải cho phía client [1].

Ở cấp độ triển khai, App Router giúp thống nhất các luồng:

- **SSR/Server Components:** phù hợp khi cần render dữ liệu theo người dùng (ví dụ thông tin lớp học, bài tập), giảm thời gian chờ trả về HTML đã có dữ liệu.
- **Client Components:** xử lý tương tác trực tiếp (làm bài kiểm tra, thao tác nộp bài, theo dõi trạng thái) và các hành vi cần bắt sự kiện theo thời gian thực.
- **API Routes:** cung cấp endpoint thống nhất cho client, đồng thời là nơi áp dụng các chính sách bảo mật (xác thực, phân quyền, giới hạn tần suất).

### 2.4 Xác thực và phân quyền: NextAuth.js

Hệ thống sử dụng NextAuth.js để hiện thực hóa đăng nhập và quản lý phiên làm việc của người dùng. Cách tiếp cận này cho phép tích hợp nhiều nhà cung cấp đăng nhập (ví dụ

OAuth) và đồng thời hỗ trợ cơ chế Credentials khi cần. Dữ liệu phiên (session) được sử dụng để thực thi phân quyền theo vai trò (giáo viên, học sinh, quản trị, phụ huynh) trong các API và luồng điều hướng [2].

Về mô hình triển khai, NextAuth.js cung cấp cơ chế quản lý phiên dựa trên JWT hoặc session lưu trong cơ sở dữ liệu. Khi người dùng đăng nhập, hệ thống tạo thông tin phiên để:

- định danh người dùng (id/email);
- gắn vai trò (role) để kiểm soát truy cập chức năng;
- đảm bảo tính nhất quán khi truy cập các API Routes.

#### 2.4.1 Phân quyền theo vai trò (RBAC) và bảo vệ truy cập

EduVerse áp dụng RBAC (Role-Based Access Control) để tách biệt quyền truy cập giữa học sinh và giáo viên, đặc biệt với các chức năng nhạy cảm như xem sự kiện thi, tạo cấu hình chống gian lận, hoặc gọi API tóm tắt chống gian lận. Trong kiến trúc Next.js, phân quyền có thể được thực thi đồng thời ở:

- **Tầng điều hướng (middleware):** chặn truy cập sớm vào các tuyến không phù hợp với vai trò.
- **Tầng API Routes:** kiểm tra phiên/role trước khi truy vấn dữ liệu hoặc trả kết quả.

Cách tiếp cận phòng thủ nhiều lớp (defense in depth) giúp giảm rủi ro truy cập trái phép ngay cả khi client bị giả mạo [1, 2].

### 2.5 Tầng dữ liệu: Prisma ORM và PostgreSQL

Prisma đóng vai trò lớp ánh xạ đối tượng-quan hệ (ORM), giúp truy cập dữ liệu thông qua client kiểu tĩnh (type-safe) trong TypeScript, giảm lỗi truy vấn và tăng tính nhất quán khi phát triển [3]. Dữ liệu nghiệp vụ cốt lõi (người dùng, khóa học/bài học, bài tập, lần làm bài, sự kiện thi) được lưu trong PostgreSQL nhờ đặc tính ACID, khả năng mở rộng và hệ sinh thái phong phú [4].

### 2.6 Cơ sở dữ liệu vector: pgvector

Để phục vụ truy xuất ngữ nghĩa cho AI Tutor, EduVerse lưu trữ embedding dưới dạng vector trong PostgreSQL thông qua phần mở rộng pgvector [5]. Việc lưu vector cùng hệ quản trị quan hệ giúp thống nhất hạ tầng dữ liệu, đồng thời cho phép truy vấn tương đồng (similarity search) trực tiếp trên cùng một nguồn dữ liệu.

### 2.7 Lưu trữ tệp: Supabase Storage

Đối với các bài nộp dạng tệp (ví dụ bài tự luận đính kèm), hệ thống sử dụng Supabase Storage để lưu trữ và cấp phát liên kết truy cập an toàn theo thời hạn (signed URL) [6].

## 2.8 Mô hình ngôn ngữ lớn (LLM) và Gemini API

Gemini API được dùng cho hai nhóm tác vụ: (i) **tạo embedding** cho tài liệu bài giảng và câu hỏi của học sinh, và (ii) **sinh câu trả lời** dựa trên ngữ cảnh được truy xuất [8].

## 2.9 Kiến trúc RAG (Retrieval-Augmented Generation)

RAG là nền tảng kỹ thuật của AI Tutor, kết hợp truy xuất tri thức từ nguồn dữ liệu nội bộ và sinh câu trả lời bằng mô hình ngôn ngữ. So với cách chỉ dùng LLM thuần túy, RAG giúp giảm hiện tượng "ảo giác" và tăng khả năng kiểm chứng nhờ bám theo tài liệu bài giảng [9].

## 2.10 Úng dụng AI trong giám sát thi

Ngoài AI Tutor, Gemini còn có thể được sử dụng để **tóm tắt tín hiệu chống gian lận** dựa trên nhật ký sự kiện thi, giúp giáo viên giám sát khi phải xem nhiều log.

## CHƯƠNG 3

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 3.1 Giới thiệu chương

Chương 3 trình bày phân tích và thiết kế hệ thống **EduVerse** dựa trên hiện thực triển khai. Nội dung tập trung làm rõ đối tượng sử dụng, yêu cầu chức năng/phi chức năng và thiết kế các thành phần cốt lõi. Trọng tâm thiết kế được nhấn mạnh ở hai mô-đun: **AI Tutor** theo **kiến trúc RAG** và **giám sát thi/chống gian lận** dựa trên event log kết hợp AI tổng hợp.

### 3.2 Phân tích hệ thống

#### 3.2.1 *Đối tượng sử dụng (Actor) và vai trò (Role)*

EduVerse phục vụ các tác nhân chính:

- **Học sinh (STUDENT):** tham gia lớp, học bài, làm bài tập/kiểm tra, tương tác AI Tutor.
- **Giáo viên (TEACHER):** tạo lớp, quản lý khóa học/bài học, tạo bài tập (Quiz/Essay), theo dõi tiến độ và giám sát thi.
- **Phụ huynh (PARENT):** liên kết với học sinh, theo dõi tiến độ/kết quả và nhận tóm tắt định kỳ.
- **Quản trị viên (ADMIN):** quản trị người dùng và cấu hình hệ thống; theo dõi nhật ký hoạt động.

#### 3.2.2 *Yêu cầu chức năng*

Các yêu cầu chức năng được mô tả theo tác nhân sử dụng chính như sau:

- **Khách (chưa đăng nhập):** đăng ký, đăng nhập, quên/đặt lại mật khẩu.
- **Học sinh:** tham gia lớp; xem bài học và học liệu; làm bài (quiz/essay); nộp bài; nhận thông báo; nhắn tin; tương tác AI Tutor.
- **Giáo viên:** tạo và quản lý lớp học; quản lý khóa học/bài học; tạo bài tập/đề kiểm tra; theo dõi tiến độ và kết quả; giám sát thi dựa trên event log và nhận hỗ trợ tổng hợp bằng AI.
- **Phụ huynh:** liên kết với học sinh; theo dõi tiến độ/kết quả; trao đổi với giáo viên.
- **Quản trị viên:** quản trị người dùng và cấu hình hệ thống; theo dõi nhật ký hoạt động.

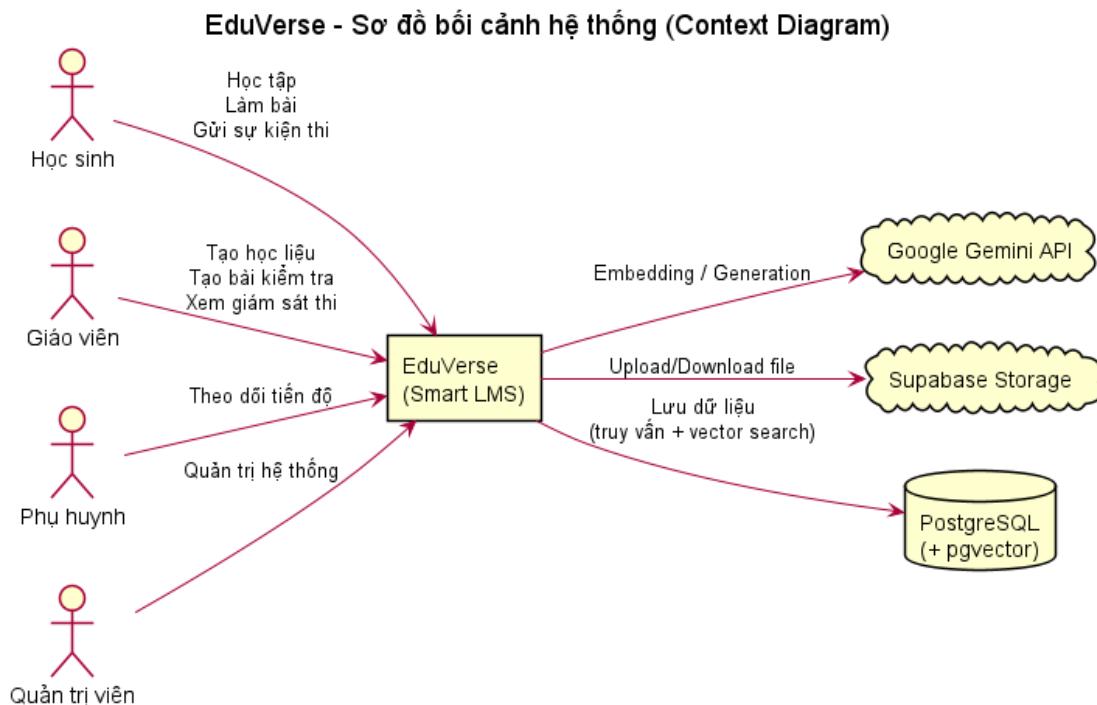
### 3.2.3 Yêu cầu phi chức năng

- **Bảo mật và phân quyền:** hệ thống dùng NextAuth (JWT strategy) và middleware điều hướng/phân quyền theo role; các API nhạy cảm giới hạn theo vai trò (ví dụ: teacher-only, admin-only) [2].
- **Kiểm soát lạm dụng (rate limit):** các API nhạy cảm (reset password, AI Tutor, AI anti-cheat, cập nhật cấu hình hệ thống) áp dụng rate limit theo IP và/hoặc user.
- **Khả năng truy vết:** hệ thống ghi audit log cho các hành vi quan trọng và lưu event log cho giám sát thi.

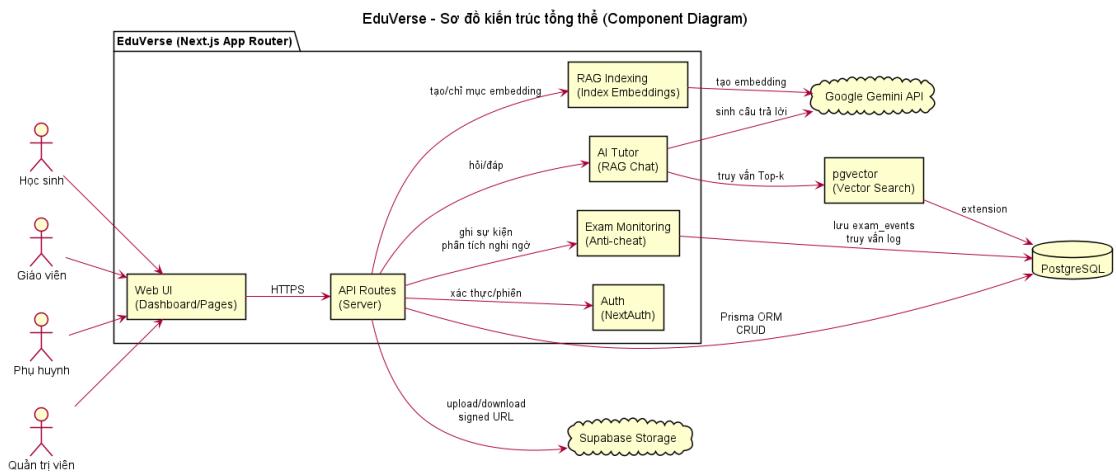
## 3.3 Thiết kế hệ thống

### 3.3.1 Thiết kế kiến trúc

EduVerse triển khai theo mô hình **Client–Server** trong một codebase Next.js: client UI gửi request đến API Routes; API Routes thực thi nghiệp vụ, truy cập dữ liệu qua Prisma và gọi dịch vụ AI.



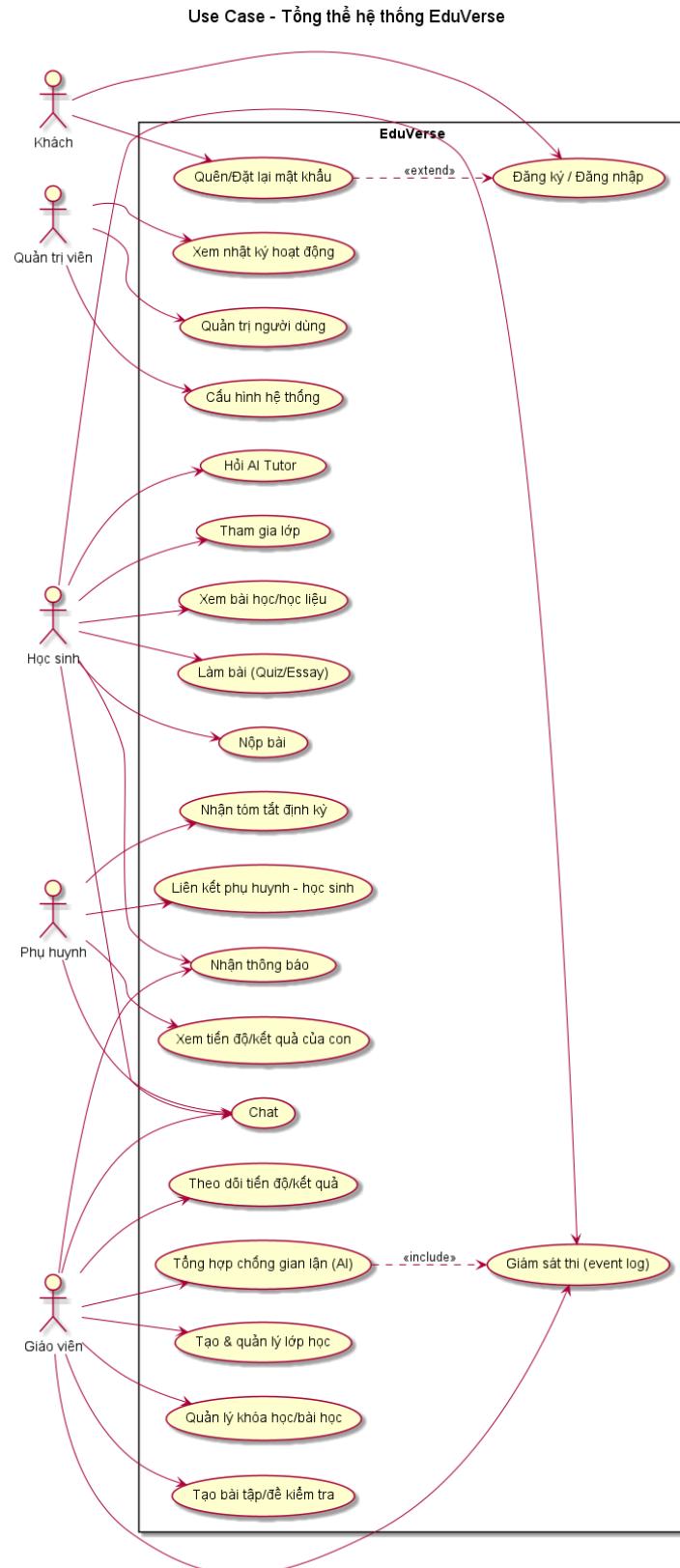
Hình 3.1: Sơ đồ bối cảnh hệ thống EduVerse



Hình 3.2: Kiến trúc tổng thể hệ thống EduVerse

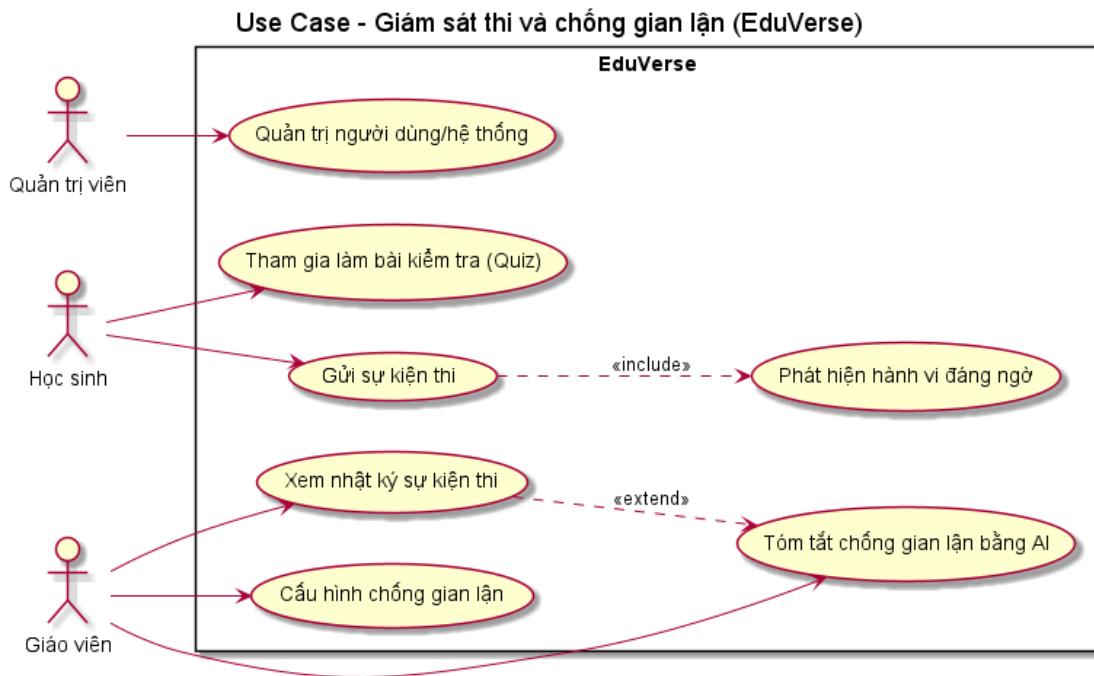
### 3.3.2 Thiết kế Use Case

## Use Case tổng thể hệ thống



Hình 3.3: Use Case tổng thể hệ thống EduVerse

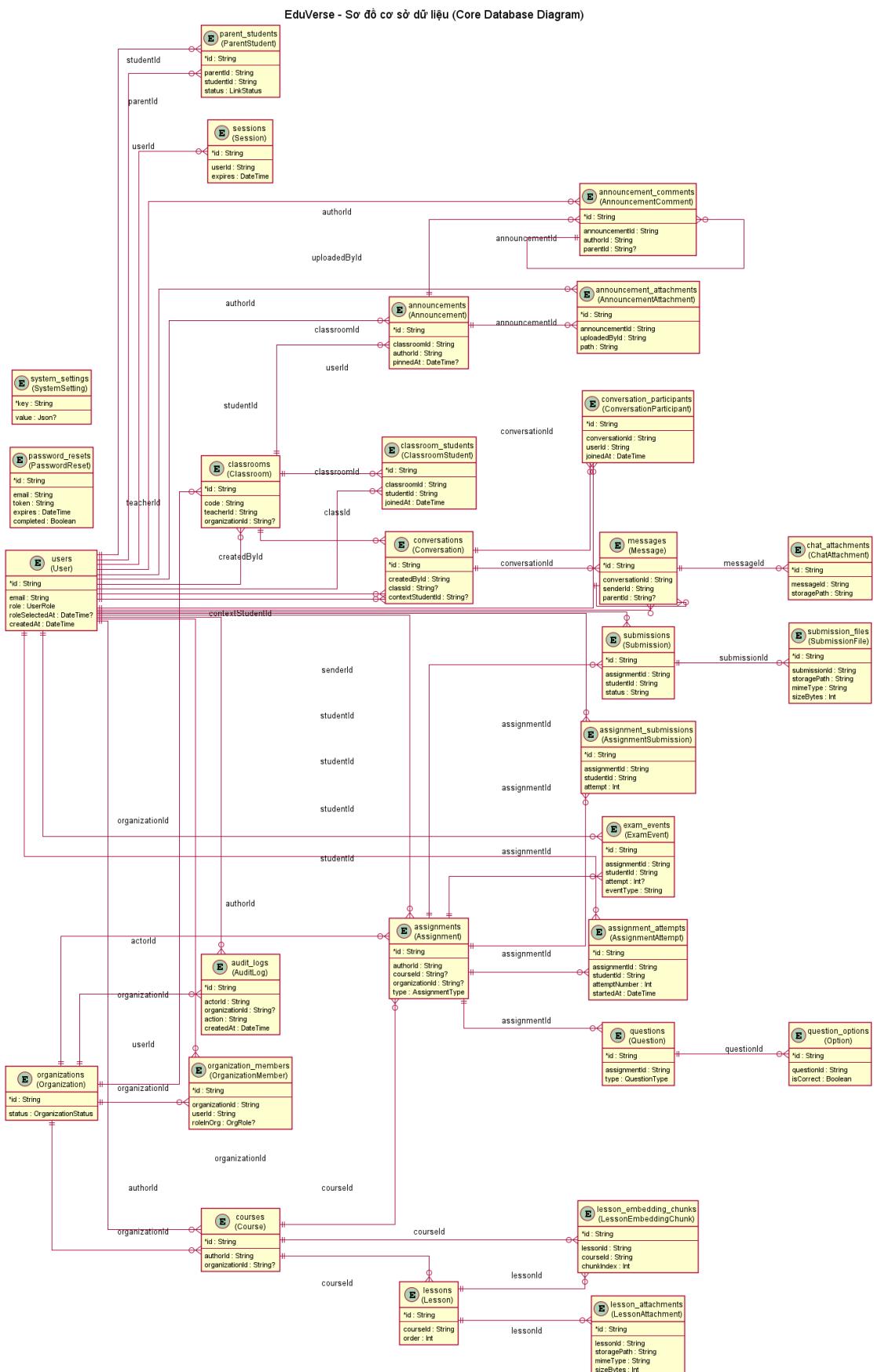
## Use Case module giám sát thi và chống gian lận



Hình 3.4: Use Case tổng quan module giám sát thi và chống gian lận

### 3.3.3 Thiết kế cơ sở dữ liệu

Hệ thống sử dụng Prisma ORM kết nối PostgreSQL [3, 4]. Hình 3.5 minh họa mối quan hệ giữa các thực thể dữ liệu cốt lõi; Bảng 3.1 tóm tắt các nhóm dữ liệu và vai trò tương ứng.



Hình 3.5: Sơ đồ cơ sở dữ liệu cốt lõi của hệ thống EduVerse

Nhóm dữ liệu	Vai trò
Người dùng	Các bảng: users (User). Lưu thông tin tài khoản, vai trò (UserRole), thời điểm chọn vai trò (roleSelectedAt).
Khôi phục mật khẩu	Các bảng: password_resets (PasswordReset). Lưu mã xác nhận đặt lại mật khẩu, thời hạn (expires) và trạng thái hoàn thành (completed).
Lớp học	Các bảng: classrooms, classroom_students. Tổ chức lớp học, liên kết học sinh-lớp.
Khóa học/bài học	Các bảng: courses, lessons, lesson_attachments. Lưu nội dung bài học và tệp đính kèm; là nguồn tri thức cho RAG Tutor.
RAG embedding	Bảng: lesson_embedding_chunks. Lưu các đoạn (chunk) đã embedding bằng pgvector để truy vấn tương đồng theo Top-k [5].
Bài tập/quiz	Các bảng: assignments, questions, question_options. Lưu bài tập dạng ESSAY/QUIZ, cấu trúc câu hỏi và đáp án.
Nộp bài (ESSAY)	Bảng: assignment_submissions. Lưu bài tự luận dạng nội dung, điểm và phản hồi.
Nộp bài (tệp)	Các bảng: submissions, submission_files. Lưu metadata tệp nộp bài (storagePath, mimeType, sizeBytes).
Giám sát thi	Các bảng: exam_events, assignment_attempts. Lưu nhật ký sự kiện thi (event log) và thông tin attempt.
Chat	Các bảng: conversations, conversation_participants, messages, chat_attachments. Lưu hội thoại và tin nhắn.
Hệ thống/Audit	Các bảng: system_settings, audit_logs. Lưu cấu hình hệ thống và nhật ký hoạt động phục vụ truy vết.

Bảng 3.1: Các bảng dữ liệu cốt lõi trong hệ thống EduVerse

## CHƯƠNG 4

# HIỆN THỰC HÓA HỆ THỐNG

### 4.1 Xây dựng trợ lý ảo AI Tutor

Mô tả cơ chế chỉ mục hóa bài học và truy vấn ngữ nghĩa phục vụ AI Tutor. Cơ chế này khai thác pgvector trên PostgreSQL để lưu trữ và truy vấn vector embedding.

### 4.2 Hệ thống đánh giá và giám sát thi cử

#### 4.2.1 Cơ chế tự động chấm bài

Kỹ thuật Prompt Engineering được sử dụng để hướng dẫn Gemini chấm bài tự luận dựa trên Rubric do giáo viên cung cấp.

#### 4.2.2 Phát hiện hành vi đáng ngờ

Phân tích logic trong suspicious-behavior.ts: Theo dõi chuyển đổi Tab, phím tắt cảm và mất kết nối.

### 4.3 Giao diện người dùng (UI/UX)

Sử dụng Tailwind CSS và các thành phần từ Shadcn UI để tạo ra một giao diện tối giản, tập trung vào trải nghiệm học tập (Learning-centric design).

## CHƯƠNG 5

### ĐÁNH GIÁ VÀ THỬ NGHIỆM

#### **5.1 Đánh giá độ chính xác của AI Tutor**

Thử nghiệm với các bộ câu hỏi mẫu và so sánh câu trả lời của AI với nội dung bài giảng thực tế.

#### **5.2 Đánh giá hiệu năng hệ thống**

Phân tích thời gian phản hồi khi truy xuất Vector và khả năng chịu tải của API Routes.

## **CHƯƠNG 6**

### **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

#### **6.1 Kết quả đạt được**

Hệ thống đã giải quyết được vấn đề cá nhân hóa hỗ trợ học tập và giúp giáo viên tự động hóa các công việc lặp đi lặp lại.

#### **6.2 Hướng phát triển tương lai**

Tích hợp nhận diện khuôn mặt để xác thực danh tính học sinh khi thi và mở rộng ứng dụng trên nền tảng di động.

## TÀI LIỆU THAM KHẢO

- [1] Vercel, "Next.js 14 Documentation," 2024. [Online]. Available: <https://nextjs.org/docs>.
- [2] NextAuth.js, "NextAuth.js Documentation," 2024. [Online]. Available: <https://next-auth.js.org>.
- [3] Prisma, "Prisma Documentation," 2024. [Online]. Available: <https://www.prisma.io/docs>.
- [4] The PostgreSQL Global Development Group, "PostgreSQL Documentation," 2024. [Online]. Available: <https://www.postgresql.org/docs/>.
- [5] pgvector, "pgvector: Open-source vector similarity search for Postgres," 2024. [Online]. Available: <https://github.com/pgvector/pgvector>.
- [6] Supabase, "Supabase Documentation," 2024. [Online]. Available: <https://supabase.com/docs>.
- [7] PlantUML, "PlantUML Documentation," 2024. [Online]. Available: <https://plantuml.com/>.
- [8] Google DeepMind, "Gemini: A Family of Highly Capable Multimodal Models," 2023.
- [9] Lewis, P., et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," NeurIPS, 2020.