

ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG ĐIỆN - ĐIỆN TỬ



ĐỒ ÁN  
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

TRIỂN KHAI HỆ THỐNG HỌC LẬP TRÌNH  
TRỰC TUYẾN LÊN CLOUD AWS

Tạ Hữu Hào

Hao.th182492@sis.hust.edu.vn

Giảng viên hướng dẫn: Th.S Đinh Thị Nhung

Chữ ký của GVHD

HÀ NỘI, 08/2023



ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG ĐIỆN - ĐIỆN TỬ



ĐỒ ÁN  
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

TRIỂN KHAI HỆ THỐNG HỌC LẬP TRÌNH  
TRỰC TUYẾN LÊN CLOUD AWS

Tạ Hữu Hào

Hao.th182492@sis.hust.edu.vn

Giảng viên hướng dẫn: Th.S Đinh Thị Nhung

Chữ ký của GVHD

Hội đồng phản biện:

HÀ NỘI, 08/2023



## ĐÁNH GIÁ QUYỀN ĐỒ ÁN TỐT NGHIỆP

(Dùng cho giảng viên hướng dẫn)

Giảng viên đánh giá: .....

Họ và tên sinh viên: ..... MSSV: .....

Tên đồ án: .....

**Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:**

STT	Tiêu chí (Điểm tối đa)	Hướng dẫn đánh giá tiêu chí	Điểm tiêu chí
1	<b>Thái độ làm việc (2,5 điểm)</b>	Nghiêm túc, tích cực và chủ động trong quá trình làm ĐATN  Hoàn thành đầy đủ và đúng tiến độ các nội dung được GVHD giao	
2	<b>Kỹ năng viết quyền ĐATN (2 điểm)</b>	Trình bày đúng mẫu quy định, bô cục các chương logic và hợp lý: Bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đê cập đến trong đồ án, có căn lè, dấu cách sau dấu chấm, dấu phẩy, có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn, v.v.  Kỹ năng diễn đạt, phân tích, giải thích, lập luận: Cấu trúc câu rõ ràng, văn phong khoa học, lập luận logic và có cơ sở, thuật ngữ chuyên ngành phù hợp, v.v.	
3	<b>Nội dung và kết quả đạt được (5 điểm)</b>	Nêu rõ tính cấp thiết, ý nghĩa khoa học và thực tiễn của đề tài, các vấn đề và các giả thuyết, phạm vi ứng dụng của đề tài. Thực hiện đầy đủ quy trình nghiên cứu: Đặt vấn đề, mục tiêu đề ra, phương pháp nghiên cứu/ giải quyết vấn đề, kết quả đạt được, đánh giá và kết luận.  Nội dung và kết quả được trình bày một cách logic và hợp lý, được phân tích và đánh giá thỏa đáng. Biện luận phân tích kết quả mô phỏng/ phần mềm/ thực nghiệm, so sánh kết quả đạt được với kết quả trước đó có liên quan.  Chỉ rõ phù hợp giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. Hàm lượng khoa học/ độ phức tạp cao, có tính mới/tính sáng tạo trong nội dung và kết quả đồ án.	
4	<b>Điểm thành tích (1 điểm)</b>	Có bài báo KH được đăng hoặc chấp nhận đăng/ đạt giải SV NCKH giải 3 cấp Trường trở lên/ Các giải thưởng khoa học trong nước, quốc tế từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế. <b>(1 điểm)</b>	

		Được báo cáo tại hội đồng cấp Trường trong hội nghị SV NCKH nhưng không đạt giải từ giải 3 trở lên/ Đạt giải khuyến khích trong cuộc thi khoa học trong nước, quốc tế/ Kết quả đồ án là sản phẩm ứng dụng có tính hoàn thiện cao, yêu cầu khối lượng thực hiện lớn. <b>(0,5 điểm)</b>	
<b>Điểm tổng các tiêu chí:</b>			
<b>Điểm hướng dẫn:</b>			

*Nhận xét khác (về thái độ và tinh thần làm việc của sinh viên)*

.....

.....

.....

.....

.....

Ngày: ... / ... / 20...

**Người nhận xét**

(Ký và ghi rõ họ tên)

## ĐÁNH GIÁ QUYỀN ĐỒ ÁN TỐT NGHIỆP

(Dùng cho cán bộ phản biện)

Giảng viên đánh giá: .....

Họ và tên sinh viên: ..... MSSV: .....

Tên đồ án: .....

**Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:**

STT	Tiêu chí (Điểm tối đa)	Hướng dẫn đánh giá tiêu chí	Điểm tiêu chí
1	<b>Trình bày quyền ĐATN (4 điểm)</b>	<p>Đồ án trình bày đúng mẫu quy định, bô cục các chương logic và hợp lý: Bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến trong đồ án, có cẩn lè, dấu cách sau dấu chấm, dấu phẩy, có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn, v.v.</p> <p>Kỹ năng diễn đạt, phân tích, giải thích, lập luận: cấu trúc câu rõ ràng, văn phong khoa học, lập luận logic và có cơ sở, thuật ngữ chuyên ngành phù hợp, v.v.</p>	
2	<b>Nội dung và kết quả đạt được (5,5 điểm)</b>	<p>Nêu rõ tính cấp thiết, ý nghĩa khoa học và thực tiễn của đề tài, các vấn đề và các giả thuyết, phạm vi ứng dụng của đề tài. Thực hiện đầy đủ quy trình nghiên cứu: Đặt vấn đề, mục tiêu đề ra, phương pháp nghiên cứu/ giải quyết vấn đề, kết quả đạt được, đánh giá và kết luận.</p> <p>Nội dung và kết quả đạt được trình bày một cách logic và hợp lý, được phân tích và đánh giá thỏa đáng. Biện luận phân tích kết quả mô phỏng/ phần mềm/ thực nghiệm, so sánh kết quả đạt được với kết quả trước đó có liên quan.</p> <p>Chỉ rõ phù hợp giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai. Hành lượng khoa học/ độ phức tạp cao, có tính mới/ sáng tạo trong nội dung và kết quả đồ án.</p>	
3	<b>Điểm thành tích (1 điểm)</b>	Có bài báo KH được đăng hoặc chấp nhận đăng/ đạt giải SV NCKH giải 3 cấp Trường trở lên/ Các giải thưởng khoa học trong nước, quốc tế từ giải 3 trở lên/ Có đăng ký bằng phát minh sáng chế.(1 điểm)	

		Được báo cáo tại hội đồng cấp Trường trong hội nghị SV NCKH nhưng không đạt giải từ giải 3 trở lên/ Đạt giải khuyến khích trong cuộc thi khoa học trong nước, quốc tế/ Kết quả đồ án là sản phẩm ứng dụng có tính hoàn thiện cao, yêu cầu khối lượng thực hiện lớn. <b>(0,5 điểm)</b>	
<b>Điểm tổng các tiêu chí:</b>			
<b>Điểm phản biện:</b>			

*Nhận xét khác của cán bộ phản biện*

.....

.....

.....

.....

.....

Ngày: ... / ... / 20...

**Người nhận xét**  
(Ký và ghi rõ họ tên)

## LỜI NÓI ĐẦU

Trong thời đại công nghệ số ngày càng phát triển mạnh mẽ, việc ứng dụng công nghệ thông tin vào các lĩnh vực trong đời sống ngày càng trở nên quan trọng và hữu ích. Sự phát triển bùng nổ của công nghệ thông tin làm tăng nhu cầu học lập trình trong mỗi người. Để đáp ứng được nhu cầu học công nghệ thông tin, việc xây dựng hệ thống học lập trình trực tuyến sẽ đáp ứng được nhu cầu tìm hiểu công nghệ thông tin trong mỗi người.

Để đáp ứng nhu cầu trên, hệ thống học lập trình trực tuyến được xây dựng bằng một website giúp người dùng có thể học tập trực tuyến mọi lúc mọi nơi thông qua các video bài giảng. Hệ thống sẽ được triển khai trên máy chủ AWS.

Để đồ án tốt nghiệp với đề tài: “**Triển khai hệ thống học lập trình trực tuyến lên Cloud AWS**” được hoàn thành. Ngoài sự nỗ lực của bản thân, em đã được sự chỉ dạy, hướng dẫn của cô giáo Th.S Đinh Thị Nhung cũng như các thầy cô giáo trong trường Điện – Điện tử của Đại học Bách Khoa Hà Nội.

Thông qua đây, em xin được gửi lời cảm ơn sâu sắc, trước tiên là tới cô giáo Th.S Đinh Thị Nhung, người trực tiếp hướng dẫn và giúp đỡ em hoàn thành đồ án tốt nghiệp. Bên cạnh đó, em xin chân thành cảm ơn Ban lãnh đạo trường Điện - Điện tử, các thầy cô trong trường đã luôn tận tình, quan tâm, dạy bảo giúp em có cơ hội nâng cao kiến thức, mở rộng hiểu biết trong suốt quá trình học tập tại trường vừa qua. Em rất biết ơn vì những điều đó.



## LỜI CAM ĐOAN

Em là Tạ Hữu Hào, mã số sinh viên 20182492, sinh viên lớp Điện tử viễn thông 03, khoá 63, người hướng dẫn là cô giáo Th.S Đinh Thị Nhụng. Em xin cam đoan toàn bộ nội dung trình bày trong đồ án “**Triển khai hệ thống học lập trình trực tuyến lên Cloud AWS**” là kết quả của quá trình tìm kiếm và nghiên cứu của em. Các dữ liệu nêu trong đồ án hoàn toàn là trung thực, phản ánh đúng kết quả thực tế. Mọi thông tin trích dẫn đều tuân thủ các quy định về sở hữu trí tuệ; các tài liệu tham khảo được liệt kê rõ ràng. Em xin chịu hoàn toàn trách nhiệm với những nội dung viết trong đồ án này.

Hà Nội, Ngày 30 tháng 07 năm 2023

**Người cam đoan**

**Tạ Hữu Hào**



# MỤC LỤC

DANH MỤC HÌNH ẢNH.....	i
DANH MỤC BẢNG BIỂU.....	iii
TÓM TẮT NỘI DUNG ĐỒ ÁN.....	iv
Abstract.....	v
CHƯƠNG 1: TỔNG QUAN .....	1
1.1    Đặt vấn đề .....	1
1.2    Mục tiêu và ý nghĩa .....	1
1.2.1    Mục tiêu .....	1
1.2.2    Ý nghĩa .....	2
1.3    Đối tượng sử dụng .....	2
1.4    Môi trường sử dụng & điều kiện sử dụng.....	7
1.5    Thị trường.....	8
1.6    Kết luận .....	8
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....	9
2.1    Học trực tuyến và xu hướng hiện đại .....	9
2.2.    Các phương pháp học trực tuyến .....	10
2.3.    Mô hình giảng dạy trực tuyến .....	11
2.4.    Quản lý người dùng và khoá học .....	12
2.5.    Công nghệ và công cụ sử dụng.....	13
2.6.    Cơ sở lý thuyết về công nghệ sử dụng .....	14
2.6.1.    Nx Monorepo.....	15
2.6.2.    Docker .....	17
2.6.3.    AWS .....	18
2.6.4.    Next.js.....	20
2.6.5.    Nest.js .....	20
2.6.6.    Postgresql.....	21
2.6.7.    CI/CD .....	23
2.7.    Kết luận .....	25
CHƯƠNG 3: THIẾT KẾ & TRIỂN KHAI HỆ THỐNG .....	26
3.1.    Thiết kế hệ thống .....	26
3.1.1.    Thu thập yêu cầu.....	26
3.1.2.    Thiết kế kiến trúc hệ thống .....	27
3.1.3.    Sơ đồ chức năng của hệ thống .....	28

3.1.4. Sơ đồ Usecase.....	30
3.1.5. Thiết kế cơ sở dữ liệu .....	34
3.2. Thiết kế phần mềm.....	43
3.2.1. Thiết lập môi trường, cấu trúc dự án .....	43
3.2.2. Xây dựng server .....	44
3.2.3. Xây dựng giao diện người dùng.....	47
3.3. Triển khai hệ thống lên Cloud AWS .....	56
3.4. Kết luận .....	61
<b>CHƯƠNG 4: KIỂM THỬ &amp; HƯỚNG PHÁT TRIỂN .....</b>	<b>62</b>
4.1. Kiểm thử.....	62
4.2. Hướng phát triển .....	63
<b>KẾT LUẬN.....</b>	<b>64</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>65</b>

## DANH MỤC HÌNH ẢNH

Hình 2.1. Logo Nx monorepo .....	15
Hình 2.2. Quy trình hoạt động của Nx .....	16
Hình 2.3. Logo docker .....	17
Hình 2.4. Logo AWS .....	18
Hình 2.5. Logo Next.js .....	20
Hình 2.6. Logo Nest.js .....	20
Hình 2.7. Logo Posgresql .....	21
Hình 2.8. Quy trình CI/CD .....	23
Hình 3.1. Sơ đồ kiến trúc hệ thống .....	27
Hình 3.2. Hình ảnh sơ đồ chức năng của hệ thống .....	29
Hình 3.3. Hình ảnh sơ đồ usecase ché độ khách .....	31
Hình 3.4. Hình ảnh usecase xác thực người dùng .....	31
Hình 3.5. Hình ảnh usecase quản lý tài khoản .....	32
Hình 3.6. Hình ảnh usecase quản lý thông báo .....	32
Hình 3.7. Hình ảnh usecase quản lý khoá học .....	33
Hình 3.8. Hình ảnh usecase quản lý tham gia khoá học .....	33
Hình 3.9. Hình ảnh usecase quản lý video bài giảng .....	34
Hình 3.10. Hình ảnh sơ đồ quan hệ dữ liệu của hệ thống .....	39
Hình 3.11. Hình ảnh cơ sở dữ liệu của hệ thống .....	42
Hình 3.12. Hình ảnh cấu trúc dự án .....	44
Hình 3.13. Hình ảnh các url api của server .....	45
Hình 3.14. Hình ảnh các url api của server (p2) .....	46
Hình 3.15. Hình ảnh các dto, response data của server .....	47
Hình 3.16. Hình ảnh giao diện trang chủ của hệ thống .....	48
Hình 3.17. Hình ảnh giao diện danh sách chủ đề theo danh mục .....	49
Hình 3.18. Hình ảnh giao diện danh sách khoá học .....	49
Hình 3.19. Hình ảnh giao diện chi tiết khoá học .....	50
Hình 3.20. Hình ảnh giao diện tạo mới khoá học .....	50
Hình 3.21. Hình ảnh giao diện quản lý chi tiết khoá học .....	51
Hình 3.22. Hình ảnh giao diện tạo video bài giảng khoá học .....	51
Hình 3.23. Hình ảnh giao diện cập nhật video bài giảng khoá học .....	52
Hình 3.24. Hình ảnh giao diện danh sách yêu cầu cấp phép khoá học .....	52
Hình 3.25. Hình ảnh giao diện danh sách thông báo .....	53
Hình 3.26. Hình ảnh giao diện chi tiết yêu cầu cấp phép khoá học .....	53
Hình 3.27. Hình ảnh giao diện chi tiết video bài giảng .....	54
Hình 3.28. Hình ảnh giao diện danh sách yêu cầu tham gia khoá học .....	54
Hình 3.29. Hình ảnh giao diện tạo bài tập .....	55
Hình 3.30. Hình ảnh giao diện cập nhật thông tin người dùng .....	55
Hình 3.31. Hình ảnh giao diện đánh giá khoá học .....	56
Hình 3.32. Hình ảnh source code được đưa lên gitlab .....	57
Hình 3.33. Hình ảnh tạo máy chủ ec2 ubuntu trên AWS .....	57
Hình 3.34. Hình ảnh truy cập vào máy chủ .....	58
Hình 3.35. Hình ảnh các biến môi trường được sử dụng trong ci/cd .....	58
Hình 3.36. Hình ảnh file gitlab-ci.yml để thực thi ci/cd trên gitlab .....	59
Hình 3.37. Hình ảnh file ci.yml thực hiện stage test, build .....	59
Hình 3.38. Hình ảnh file build-image.yml để thực hiện stages build-docker-image .....	60

Hình 3.39. Hình ảnh file deploy.yml thực hiện stage deploy .....60

## **DANH MỤC BẢNG BIỂU**

Bảng 1.1. Bảng câu hỏi phỏng vấn .....	2
Bảng 1.2. Bảng khách hàng tham gia phỏng vấn.....	3
Bảng 1.3. Bảng trả lời câu hỏi phỏng vấn của khách hàng(Kh) .....	5
Bảng 2.1. Bảng tính năng của postgresql.....	22
Bảng 3.1. Bảng thuộc tính các đối tượng trong hệ thống .....	35
Bảng 3.2. Bảng các cặp dữ liệu quan hệ 1-1 .....	39
Bảng 3.3. Bảng các cặp dữ liệu quan hệ N-N .....	39
Bảng 3.4. Bảng các cặp dữ liệu quan hệ 1-N .....	39
Bảng 3.5. Bảng danh sách các bảng trong cơ sở dữ liệu .....	40
Bảng 4.1. Bảng danh sách testcase .....	62

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Nội dung chính của đề tài: Thiết kế hệ thống học lập trình trực tuyến và triển khai hệ thống tự động lên cloud AWS.

Mục đích: Xây dựng website hệ thống học lập trình trực tuyến triển khai lên máy chủ AWS. Đưa hệ thống và sử dụng giúp người dùng có thể học lập trình trực tuyến thông qua việc tham gia các khóa học với các video bài giảng.

**Cụ thể nội dung được chia thành các chương chính như sau:**

- Chương 1: Tổng quan đề tài
- Chương 2: Cơ sở lý thuyết
- Chương 3: Thiết kế & triển khai hệ thống
- Chương 4: Kiểm thử & hướng phát triển
- Kết luận

## Phương pháp thực hiện

Một số phương pháp chính được sử dụng chính được sử dụng trong đề tài

- Phương pháp thu thập và tổng hợp tài liệu
- Phương pháp tổng kết, phân tích, đánh giá
- Phương pháp thiết kế kiến trúc phần mềm
- Phương pháp kiểm thử

## Công cụ sử dụng

Một số công cụ sử dụng: IDE Visual studio code, Dbeaver, DbDiagram, Diagram.io

## Kết luận

Qua bản đồ án này, em đã hiểu và nắm bắt được các bước thiết kế hệ thống phần mềm. Đồ án tốt nghiệp giúp em củng cố kiến thức chuyên ngành và bồi dưỡng thêm cho em các kỹ năng mềm như kỹ năng đọc hiểu, kỹ năng tin học văn phòng... Mong rằng đồ án này sẽ trở thành một tài liệu tham khảo và có thể vận dụng vào thực tiễn.

**Sinh viên thực hiện**  
(Ký và ghi rõ họ tên)

## **Abstract**

The main content of the topic is: Designing an online programming learning system and deploying the system to the AWS cloud.

Objective: To build a website for an online programming learning system and deploy it on the AWS server. The system aims to allow users to learn programming online through participating in courses with instructional videos.

**The specific content is divided into main chapters as follows:**

- Chapter 1: Introduction to the topic
- Chapter 2: Theoretical foundations
- Chapter 3: System design & deployment
- Chapter 4: Testing & development direction
- Conclusion

**Implementation Method:**

Some main methods used in the topic include:

- Method of collecting and synthesizing materials
- Summarization, analysis, and evaluation method
- Software architecture design method
- Testing method

**Tools Used:**

Some tools used include: Visual Studio Code IDE, DBeaver, DbDiagram, Diagram.io.

**Conclusion:**

Through this project, I have understood and grasped the steps of software system design. This graduation thesis has not only provided me with specialized knowledge but also enhanced my soft skills such as reading comprehension and proficiency in office computer skills. I hope that this thesis will become a valuable reference document and can be applied in practical scenarios.



# CHƯƠNG 1: TỔNG QUAN

## 1.1 Đặt vấn đề

Ngày nay, công nghệ thông tin đã trở thành một phần không thể thiếu của cuộc sống hiện đại. Sự phát triển nhanh chóng của công nghệ này đã mang lại nhiều tiện ích và cơ hội mới, mở ra một tương lai đầy triển vọng. Một lĩnh vực quan trọng và phổ biến trong lĩnh vực này là lập trình.

Công nghệ thông tin đã thúc đẩy sự phát triển của lập trình từ việc tạo ra các ứng dụng di động, phần mềm, website cho đến trí tuệ nhân tạo và blockchain. Lập trình không chỉ là một công việc chuyên môn, mà còn là một kỹ năng cần thiết trong nhiều lĩnh vực khác nhau như khoa học, kinh doanh, y tế và giáo dục.

Chính sự phát triển bùng nổ của công nghệ thông tin, nhu cầu học lập trình ngày càng tăng cao. Công nghệ ngày càng trở nên phổ biến và phức tạp, đòi hỏi nguồn nhân lực có kiến thức chuyên môn vững vàng để đáp ứng nhu cầu của xã hội. Các công ty công nghệ và tổ chức đang tìm kiếm những lập trình viên có khả năng sáng tạo, giải quyết vấn đề và xây dựng những sản phẩm đột phá.

Học lập trình không chỉ mang lại cơ hội nghề nghiệp tốt, mà còn giúp mở ra một thế giới mới đầy khám phá và sáng tạo. Các nguồn tài liệu và khóa học trực tuyến phong phú đã được phát triển, giúp mọi người có thể học lập trình một cách linh hoạt và tiện lợi.

Chính vì những nhu cầu trên, một hệ thống học lập trình trực tuyến được xây dựng để tạo ra môi trường cho những người có nhu cầu, đam mê về công nghệ thông tin học tập và phát triển bản thân và có cơ hội làm việc với công nghệ thông tin phục vụ cuộc sống trong thời đại chuyển đổi số ngày nay.

## 1.2 Mục tiêu và ý nghĩa

### 1.2.1 Mục tiêu

Dựa trên nhu cầu, hệ thống học lập trình được phát triển với mục tiêu xây dựng một môi trường học tập đa dạng, linh hoạt và thú vị cho mọi người. Hệ thống tập trung vào các mục tiêu sau:

- Đa dạng hóa khóa học: Hệ thống đảm bảo rằng mọi người có cơ hội tham gia vào khóa học miễn phí và trả phí. Điều này tạo ra sự công bằng và kết nối giữa những người có khả năng tài chính khác nhau. Mọi người có thể tìm thấy các khóa học phù hợp với mục tiêu học tập và ngân sách của mình.
- Khả năng tạo và chia sẻ khóa học: Hệ thống khuyến khích mọi người tự tạo ra các khóa học và chia sẻ kiến thức của mình với cộng đồng. Người dùng có thể tạo nội dung giảng dạy, bài giảng, bài tập và chia sẻ chúng miễn phí hoặc bán với mức

giá hợp lý. Điều này thúc đẩy sự phát triển và trao đổi kiến thức trong cộng đồng lập trình.

3. Truy cập linh hoạt và học mọi lúc, mọi nơi: Hệ thống học lập trình được thiết kế để người dùng có thể tiếp cận nội dung học tập từ mọi nơi và mọi thiết bị. Người học có thể truy cập vào khóa học bằng máy tính, điện thoại di động hoặc máy tính bảng, cho phép họ học tập theo lịch trình và nhu cầu cá nhân của mình.
4. Tạo môi trường học tập tương tác: Hệ thống tạo ra một môi trường trực tuyến giống như một lớp học offline, cho phép người học trao đổi thông tin, đặt câu hỏi và giải đáp khó khăn, thắc mắc với giảng viên và cả cộng đồng học viên khác. Điều này tạo ra một cộng đồng học tập đồng nghiệp và giúp người học cảm thấy được sự hỗ trợ và động viên trong quá trình học tập.

### 1.2.2 Ý nghĩa

Hệ thống học lập trình trực tuyến xây dựng với ý nghĩa tạo ra một môi trường học tập thân thiện, đa dạng và linh hoạt, khuyến khích mọi người tham gia, chia sẻ kiến thức và phát triển kỹ năng lập trình của mình một cách hiệu quả và tiện lợi. Giúp những người có nhu cầu, đam mê công nghệ thông tin mà chưa biết bắt đầu như thế nào có một lộ trình và bước đi đầu tiên đúng đắn. Là nơi đặt những viên gạch đầu tiên trong sự phát triển của bản thân mỗi người trên con đường phát triển công nghệ thông tin.

## 1.3 Đối tượng sử dụng

Hệ thống học lập trình này dành cho tất cả mọi người, mọi lứa tuổi, từ học sinh, sinh viên, lập trình viên, những người muốn chuyển ngành, từ người mới bắt đầu muốn khám phá lập trình cho đến những lập trình viên có kinh nghiệm muốn nâng cao kỹ năng.

Bảng 1.1. Bảng câu hỏi phỏng vấn

STT	Câu hỏi
1	Bạn có quan tâm và theo dõi sự phát triển của công nghệ thông tin hiện nay không? (Có/Không)
2	Bạn đã sử dụng các sản phẩm công nghệ thông tin như ứng dụng di động, phần mềm, trang web trong cuộc sống hàng ngày chưa? (Có/Không)
3	Bạn có cảm thấy nhu cầu học lập trình là cần thiết trong thời đại công nghệ thông tin hiện nay không? (Có/Không)
4	Bạn đã từng học lập trình trước đây không? (Có/Không)
5	Nếu bạn đã học lập trình trước đây, bạn cảm thấy nó đã giúp ích cho cuộc sống và công việc của bạn không? (Có/Không)
6	Bạn đã từng tham gia khoá học lập trình trực tuyến hay offline chưa? (Có/Không)
7	Bạn nghĩ gì về việc tự tạo khoá học để chia sẻ kiến thức lập trình với người khác? (Rất tốt/Tốt/Bình thường/Kém)

8	Bạn có muốn học lập trình miễn phí hay có sẵn các khóa học trả phí với chất lượng cao hơn? (Miễn phí/Trả phí/Không quan tâm)
9	Bạn nghĩ gì về khả năng học lập trình ở mọi lúc, mọi nơi thông qua các nền tảng học trực tuyến? (Rất thuận tiện/Thuận tiện/Bình thường/Không thuận tiện)
10	Bạn có ưu tiên tham gia môi trường học tập tương tác, nơi bạn có thể trao đổi, đặt câu hỏi và nhận giải đáp từ giảng viên và cộng đồng học viên khác không? (Có/Không)

Danh sách khách hàng tham gia khảo sát bộ câu hỏi trên.

*Bảng 1.2. Bảng khách hàng tham gia phỏng vấn*

TT	Khách hàng	Thông tin cá nhân	Chia sẻ
1	 <p><b>Bùi Gia Long</b></p> <p>Thêm bạn bè Nhắn tin ...</p> <p>Bài viết Ành</p> <p>Chi tiết</p> <ul style="list-style-type: none"> <li>Sống tại Phúc Lý, Ha Noi, Vietnam</li> <li>Đến từ Hà Nội</li> <li>Đã kết hôn</li> <li>Xem thông tin giới thiệu của Bùi Gia</li> </ul>	<p>Tuổi: thế hệ 9x          Quê quán: Hà Nội          Tình trạng: Đã kết hôn          Nghề nghiệp: Lập trình viên</p>	<p>Tôi là một lập trình viên đã có 8 năm kinh nghiệm. Nhu cầu về nhân lực công nghệ thông tin chất lượng cao ngày càng lớn. Tôi hy vọng có một hệ thống học tập lập trình trực tuyến có có một lộ trình rõ ràng cho người học ngay từ đầu. Tôi cũng muốn sau này không làm lập trình viên nữa sẽ quay lại dạy, chia sẻ kiến thức, kinh nghiệm của mình cho các thế hệ sau.</p>

2	 <p><b>Đức Thành</b></p> <p>Ngày Em Đẹp Nhất 1967</p> <p>Bạn bè Nhắn tin ...</p> <p>Bài viết Ảnh Reels</p> <p>Chi tiết</p> <ul style="list-style-type: none"> <li>Học Công Nghệ Thông Tin tại Trường Đại học Giao thông Vận tải, Hà nội</li> <li>Đã học tại THPT Việt Đức</li> <li>Đã học tại THCS Quỳnh Mai</li> <li>Sống tại Hà Nội</li> <li>Độc thân</li> </ul>	<p>Tuổi: 22 Quê quán: Hà Nội Tình Trạng: Độc thân Nghề nghiệp: Sinh viên</p>	<p>Tôi hiện đang là sinh viên học chuyên ngành công nghệ thông tin. Tôi có tham gia các khoá học lập trình trên mạng nhưng ở rất nhiều nơi và không có lộ trình cụ thể. Tôi muốn có một website học lập trình có lộ trình để cho những người mới có thể tiếp cận công nghệ thông tin một cách dễ dàng và có định hướng.</p>
3	 <p><b>Mạnh Sơn</b></p> <p>Bạn bè Nhắn tin ...</p> <p>Làm việc tại Học viên Tài Chính (Academy Of Finance)</p> <p>Đã làm việc tại Trường THPT B Kim Bảng</p> <p>Đã học tại Trường THPT B Kim Bảng</p> <p>Sống tại Ha-Nam, Hà Nam, Vietnam</p> <p>Đến từ Kim Bảng, Hà Nam Ninh, Vietnam</p> <p>Có 102 người theo dõi</p> <p>Xem thông tin giới thiệu của Sơn</p>	<p>Tuổi: 23 Quê quán: Hà Nam Tình trạng: Độc thân Nghề nghiệp: Kế toán</p>	<p>Tôi đang là kế toán. Công việc của tôi rất nhiều, tôi có mong muốn học lập trình để có thể ứng dụng vào công việc để giảm tải những công việc lặp đi lặp lại.</p>

4	 <p><b>Kiên Vũ</b></p> <p>Bạn bè Nhắn tin</p> <ul style="list-style-type: none"> <li>Hacking (computers) tại ĐQ. Trường THPT Tứ Kỳ - Niềm Tự Hào Của Tôi và Bạn.</li> <li>Đã làm việc tại Tứ Kỳ Confessions</li> <li>Sống tại Hải Dương</li> <li>Đến từ Hải Dương</li> <li>Tham gia vào Tháng 12 năm 2016</li> <li>Xem thông tin giới thiệu của Kien</li> </ul>	Tuổi : 17 Quê quán: Hải Dương Tình trạng: Độc thân Nghề nghiệp: Học sinh lớp 12	Em đang là học sinh và chuẩn bị thi đại học. Em yêu thích lập trình và mong muốn thành lập trinh viên trong tương lai. Em hiện tại chưa biết lập trình và muốn có một nơi dạy lập trình cho những người mới như em để biết mình có thích lập trình hay không để có lựa chọn đúng đắn.
5	 <p><b>Hoàng Mạnh Cường (Merlin)</b></p> <p>Bạn bè Nhắn tin</p> <ul style="list-style-type: none"> <li>Sống tại Hà Nội</li> <li>Xem thông tin giới thiệu của Cường</li> <li>Bộ sưu tập</li> <li>Bạn bè 3 bạn chung</li> </ul>	Tuổi 26 Quê quán: Hà Nội Tình trạng: Độc thân Nghề nghiệp: Tester phần mềm	Tôi đang là một tester phần mềm cho công ty IT, tôi đã từng học lập trình trong thời đại học. Tôi muốn học về lập trình để giúp ích thêm trong công việc của tôi.

Bảng 1.3. Bảng trả lời câu hỏi phỏng vấn của khách hàng(Kh)

STT	Câu hỏi	Kh1	Kh2	Kh3	Kh4	Kh5
1	Bạn có quan tâm và theo dõi sự phát triển của công nghệ thông tin hiện nay không? (Có/Không)	Có	Có	Có	Có	Có
2	Bạn đã sử dụng các sản phẩm công nghệ thông tin như ứng dụng di động, phần mềm, trang web trong cuộc sống hàng ngày chưa? (Có/Không)	Có	Có	Có	Có	Có
3	Bạn có cảm thấy nhu cầu học lập trình là cần thiết trong thời đại công nghệ thông tin hiện nay không? (Có/Không)	Có	Có	Có	Có	Có

4	Bạn đã từng học lập trình trước đây không? (Có/Không)	Có	Có	Không	Không	Có
5	Nếu bạn đã học lập trình trước đây, bạn cảm thấy nó đã giúp ích cho cuộc sống và công việc của bạn không? (Có/Không)	Có	Có			Có
6	Bạn đã từng tham gia khoá học lập trình trực tuyến hay offline chưa? (Có/Không)	Có	Có	Không	Không	Không
7	Bạn nghĩ gì về việc tự tạo khoá học để chia sẻ kiến thức lập trình với người khác? (Rất tốt / Tốt / Bình thường / Kém)	Rất tốt	Tốt	Tốt	Tốt	Tốt
8	Bạn có muốn học lập trình miễn phí hay có sẵn các khóa học trả phí với chất lượng cao hơn? (Miễn phí / Trả phí / Không quan tâm)	Trả phí	Miễn phí và trả phí	Miễn phí	Miễn phí	Miễn phí
9	Bạn nghĩ gì về khả năng học lập trình ở mọi lúc, mọi nơi thông qua các nền tảng học trực tuyến? (Rất thuận tiện / Thuận tiện / Bình thường / Không thuận tiện)	Rất thuận tiện	Rất thuận tiện	Rất thuận tiện	Rất thuận tiện	Rất thuận tiện
10	Bạn có ưu tiên tham gia môi trường học tập tương tác, nơi bạn có thể trao đổi, đặt câu hỏi và nhận giải đáp từ giảng viên và cộng đồng học viên khác không? (Có/Không)	Có	Có	Có	Có	Có

Dựa trên khảo sát và phân tích về bài khảo sát trên, hệ thống được xây dựng hướng đến các đối tượng:

1. Người mới bắt đầu: Hệ thống cung cấp những khóa học cơ bản giúp người mới bắt đầu hiểu về lập trình và hướng dẫn các ngôn ngữ lập trình phổ biến. Người mới bắt đầu có thể tự học theo tốc độ của mình và nhờ vào tính linh hoạt của hệ thống, họ có thể nắm bắt cơ bản mà không cần kiến thức hay kinh nghiệm trước đó.
2. Sinh viên và học sinh: Hệ thống học lập trình cung cấp các khóa học phù hợp với nhu cầu giảng dạy trong các trường đại học, cao đẳng và trung học. Sinh viên và học sinh có thể truy cập vào tài liệu học tập, bài giảng và bài tập để nâng cao kiến thức và kỹ năng lập trình của mình.
3. Lập trình viên chuyên nghiệp: Đối với những người đã có kinh nghiệm trong lập trình, hệ thống cung cấp những khóa học tiên tiến và chuyên sâu để giúp họ mở

rộng kiến thức và nắm bắt các công nghệ mới. Họ có thể tìm hiểu về các lĩnh vực chuyên môn như trí tuệ nhân tạo, khoa học dữ liệu, phát triển ứng dụng di động và nhiều hơn nữa.

4. Những người muốn chuyển nghề: Hệ thống học lập trình cũng phục vụ những người muốn chuyển sang lĩnh vực công nghệ thông tin và lập trình. Hệ thống cung cấp các khóa học định hướng nghề nghiệp và hỗ trợ trong việc chuẩn bị cho công việc mới trong lĩnh vực này.

Dù bạn là người mới bắt đầu, sinh viên, lập trình viên chuyên nghiệp hay người muốn chuyển nghề, hệ thống học lập trình sẽ cung cấp tài liệu học tập phù hợp, môi trường tương tác và cơ hội phát triển kỹ năng lập trình cho bạn.

## 1.4 Môi trường sử dụng & điều kiện sử dụng

Môi trường và điều kiện sử dụng của hệ thống học lập trình có thể được mô tả như sau:

1. Thiết bị: Hệ thống học lập trình có thể truy cập từ các thiết bị điện tử như máy tính (desktop hoặc laptop), điện thoại di động và máy tính bảng. Điều này cho phép người dùng truy cập và học tập từ bất kỳ đâu và bất kỳ khi nào.
2. Kết nối internet: Để truy cập và sử dụng hệ thống học lập trình, người dùng cần có kết nối internet ổn định. Điều này đảm bảo họ có thể tải nội dung học tập, xem video, tham gia diễn đàn và tương tác trực tiếp với cộng đồng học viên.
3. Trình duyệt web: Hệ thống học lập trình hoạt động trên các trình duyệt web phổ biến như Google Chrome, Mozilla Firefox, Safari, và Microsoft Edge. Người dùng cần cài đặt một trình duyệt web tương thích để truy cập vào hệ thống và tận hưởng các tính năng của nó.
4. Tài khoản người dùng: Để sử dụng hệ thống học lập trình, người dùng cần tạo một tài khoản đăng nhập. Tài khoản này cho phép họ truy cập vào các khóa học, lưu trữ tiến trình học tập và tương tác với giảng viên và cộng đồng học viên khác.
5. Phần mềm hỗ trợ: Một số khóa học lập trình có thể yêu cầu người dùng cài đặt và sử dụng các phần mềm hỗ trợ cụ thể. Điều này có thể bao gồm các trình biên dịch, trình thông dịch, trình chỉnh sửa mã nguồn và các công cụ phát triển phụ trợ. Thông tin về các phần mềm này sẽ được cung cấp rõ ràng trong từng khóa học cụ thể.

Điều kiện sử dụng hệ thống học lập trình là người dùng cần có trách nhiệm cá nhân và cam kết học tập. Họ nên dành thời gian và nỗ lực để nắm bắt nội dung học tập, hoàn thành bài tập và tham gia tích cực trong các hoạt động tương tác trong cộng đồng học viên.

## 1.5 Thị trường

Trên thị trường hiện nay, có nhiều hệ thống học lập trình khác nhau cạnh tranh với nhau. Để có thể cạnh tranh được với các hệ thống học lập trình khác hệ thống cần phải tập trung vào chất lượng và đưa ra các điểm mới lạ hơn so với những hệ thống học đã có trên thị trường. Hiện nay có một số cộng đồng lập trình có hệ thống online như F8, Udemy,... Mỗi hệ thống đều có một đặc điểm riêng biệt

- F8: Khoá học có thể chọn học miễn phí, trả phí. Do chính F8 xuất bản, người bên ngoài không chia sẻ được. Việc này làm hạn chế khoá học, hạn chế đến đối tượng khách hàng.

- Udemy: Khoá học trả phí, đăng ký mở bán khoá học trên Udemy với đa dạng tất cả các loại khoá học. Thanh toán bằng thẻ tín dụng để học. Các khoá học đa phần là tiếng Anh. Không có khoá học miễn phí.

Hệ thống học lập trình trên đã có trước và có lượng khách hàng lớn, để cạnh tranh được và bước chân vào thị trường thì hệ thống mới này đã đưa ra mục tiêu cụ thể tập trung chủ yếu vào công nghệ thông tin, cho phép tất cả mọi người đều có thể chia sẻ khoá học do chính mình tạo ra một cách dễ dàng, các khoá học phải phù hợp với nội dung cho phép. Mọi người có thể tham gia khoá học dễ dàng, không giới hạn.

## 1.6 Kết luận

Dựa trên xu thế công nghệ thông tin và nhu cầu về lập trình của mọi người trong xã hội ngày nay, hệ thống học lập trình trực tuyến sẽ được xây dựng để giải quyết các nhu cầu học lập trình, tìm hiểu về công nghệ thông tin cho mọi người từ những người chưa có kinh nghiệm đến những người có kinh nghiệm có thể học tập và chia sẻ những kiến thức của bản thân cho mọi người. Chương tiếp theo là tìm hiểu về cơ sở lý thuyết xây dựng lên hệ thống học lập trình trực tuyến.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1 Học trực tuyến và xu hướng hiện đại

Cơ sở lý thuyết của học trực tuyến dựa trên một số nguyên lý và xu hướng học tập hiện đại. Dưới đây là một số cơ sở lý thuyết và xu hướng quan trọng:

1. Nguyên lý học tập linh hoạt: Học trực tuyến tạo ra một môi trường học tập linh hoạt, cho phép người học tự điều chỉnh thời gian và địa điểm học tập. Người học có thể tiếp cận nội dung học tập từ bất kỳ nơi nào và bất kỳ khi nào phù hợp với lịch trình và khả năng của mình. Điều này tạo điều kiện thuận lợi cho việc học tập tự chủ và phù hợp với nhu cầu cá nhân.
2. Nguyên lý học tập theo nhu cầu: Học trực tuyến đặt người học vào trung tâm quá trình học tập. Nó cung cấp cho người học khả năng lựa chọn các khóa học và tài liệu phù hợp với nhu cầu và mục tiêu học tập của mình. Người học có thể tùy chỉnh chương trình học tập của mình và chọn những khóa học có tính ứng dụng cao trong lĩnh vực mong muốn.
3. Nguyên lý học tập tương tác: Học trực tuyến tạo ra một môi trường học tập tương tác, trong đó người học có thể tương tác với giảng viên và cộng đồng học viên khác. Thông qua các công nghệ truyền thông như diễn đàn, trò chuyện trực tuyến và hỗ trợ qua email, người học có thể thảo luận, chia sẻ ý kiến và nhận được phản hồi từ giảng viên và đồng học viên. Tương tác này tạo ra một môi trường học tập sôi nổi và thúc đẩy việc học tập thông qua việc giao tiếp và chia sẻ kiến thức.
4. Xu hướng học tập tự học: Học trực tuyến thúc đẩy xu hướng học tập tự học, nơi người học tự chủ và tự quản lý quá trình học tập. Nó khuyến khích người học đảm nhận vai trò chủ động trong việc xác định mục tiêu, lựa chọn tài liệu học tập và quản lý thời gian học tập. Điều này giúp người học phát triển kỹ năng tự học và nâng cao khả năng tự phát triển trong cuộc sống và sự nghiệp.
5. Sử dụng công nghệ thông tin và internet: Học trực tuyến tận dụng sự phát triển của công nghệ thông tin và internet để mang lại những tiện ích và ưu điểm cho quá trình học tập. Nó sử dụng các công nghệ truyền thông như video, trực tuyến, bài giảng ghi âm, diễn đàn trực tuyến và các ứng dụng di động để cung cấp nội dung học tập và tương tác giữa giảng viên và người học.

Học trực tuyến mang lại nhiều lợi ích quan trọng cho người học. Dưới đây là một số lợi ích chính của học trực tuyến:

1. Linh hoạt về thời gian và địa điểm: Học trực tuyến cho phép người học tự chủ về thời gian và địa điểm học tập. Người học có thể tiếp cận nội dung học tập từ bất kỳ nơi đâu và vào bất kỳ thời điểm nào phù hợp với lịch trình cá nhân và cam kết

khác. Không cần phải di chuyển đến một địa điểm cụ thể, học trực tuyến giúp tiết kiệm thời gian và công sức.

2. Tiết kiệm thời gian và chi phí: Học trực tuyến tiết kiệm thời gian và chi phí cho người học. Không cần phải di chuyển đến trường học hoặc tổ chức giáo dục truyền thống, người học có thể tiết kiệm thời gian di chuyển và các chi phí phát sinh như đi lại, chỗ ở, và tiền vé. Họ có thể tập trung vào việc học tập mà không phải lo lắng về các yếu tố khác như giao thông và tiện ích.
3. Tiếp cận kiến thức từ mọi nơi: Học trực tuyến cho phép người học tiếp cận kiến thức từ mọi nơi trên thế giới. Bất kể bạn ở đâu, chỉ cần có kết nối internet, bạn có thể truy cập vào các khóa học, tài liệu và nguồn kiến thức trực tuyến. Điều này mở rộng phạm vi tiếp cận kiến thức và tạo điều kiện cho việc học tập toàn cầu.
4. Tự chủ và tự học: Học trực tuyến khuyến khích người học trở nên tự chủ và tự học. Người học có thể tự quản lý thời gian và tiến độ học tập của mình. Họ có thể lựa chọn các khóa học phù hợp với mục tiêu và mong muốn cá nhân và tự điều chỉnh tốc độ học tập. Điều này phát triển kỹ năng tự học và sự độc lập trong việc tiếp thu kiến thức.
5. Tương tác và phản hồi: Hệ thống học trực tuyến cung cấp khả năng tương tác và phản hồi từ giảng viên và cộng đồng học viên. Thông qua các công nghệ truyền thông như diễn đàn, thảo luận trực tuyến và email, người học có thể giao tiếp trực tiếp với giảng viên và đồng học viên, chia sẻ ý kiến, đặt câu hỏi và nhận phản hồi. Điều này tạo ra một môi trường học tập tương tác và hỗ trợ.

Tóm lại, học trực tuyến mang lại lợi ích lớn như sự linh hoạt về thời gian và địa điểm, tiết kiệm thời gian và chi phí, khả năng tiếp cận kiến thức từ mọi nơi, sự tự chủ và tự học, và khả năng tương tác và phản hồi. Đây là những lợi ích quan trọng đáng xem xét khi lựa chọn hình thức học tập hiện đại.

## 2.2. Các phương pháp học trực tuyến

Có nhiều phương pháp học trực tuyến được sử dụng để cung cấp trải nghiệm học tập tốt nhất cho người học. Dưới đây là một số phương pháp phổ biến trong học trực tuyến:

1. Video học tập: Phương pháp này sử dụng video để cung cấp nội dung học tập cho người học. Giảng viên tạo và ghi lại các bài giảng, giải thích, bài thực hành và ví dụ trên video, sau đó chia sẻ chúng với người học. Video học tập cho phép người học học tập theo tốc độ riêng và xem lại các khái niệm và bài giảng khi cần thiết.
2. Học qua trực tuyến: Đây là phương pháp sử dụng nền tảng học trực tuyến hoặc phần mềm để cung cấp các khóa học và tài liệu học tập. Người học có thể truy cập vào các tài liệu học tập, bài giảng, bài kiểm tra và bài tập thông qua trang web hoặc ứng dụng. Hình thức này cung cấp sự linh hoạt cho người học để tiếp thu kiến thức theo tốc độ và lịch trình riêng.

3. Diễn đàn và thảo luận trực tuyến: Phương pháp này tạo ra một môi trường tương tác và thảo luận trực tuyến cho người học. Người học có thể tham gia vào các diễn đàn, nhóm thảo luận, hoặc trò chuyện trực tuyến để chia sẻ ý kiến, trao đổi thông tin và thảo luận vấn đề liên quan đến nội dung học tập. Điều này khuyến khích sự tương tác giữa người học, giảng viên và cộng đồng học viên.
4. Bài tập và kiểm tra trực tuyến: Học trực tuyến cung cấp cơ hội cho người học thực hành và kiểm tra kiến thức thông qua các bài tập và bài kiểm tra trực tuyến. Người học có thể hoàn thành các bài tập và kiểm tra trực tuyến để áp dụng kiến thức đã học và đánh giá tiến trình học tập của mình. Hình thức này cung cấp phản hồi tức thì và giúp người học đo lường mức độ hiểu biết và tiến bộ của mình.
5. Hướng dẫn từ xa: Phương pháp này sử dụng các công cụ truyền thông như video cuộc gọi, trò chuyện video, hoặc hệ thống quản lý học tập để cung cấp hướng dẫn từ xa cho người học. Giảng viên và người học có thể giao tiếp trực tiếp, giải đáp câu hỏi và nhận sự hỗ trợ qua các công nghệ truyền thông trực tuyến.
6. Học tập hỗn hợp: Phương pháp này kết hợp học trực tuyến và học truyền thống trong một khóa học. Người học có thể tham gia vào các buổi học trực tuyến và cũng có thể tham gia vào các buổi học tại một địa điểm vật lý. Hình thức này tận dụng sự linh hoạt của học trực tuyến và đồng thời tạo ra cơ hội tương tác trực tiếp và học tập nhóm.

Tóm lại, học trực tuyến có nhiều phương pháp học tập khác nhau như video học tập, học qua trực tuyến, diễn đàn và thảo luận trực tuyến, bài tập và kiểm tra trực tuyến, hướng dẫn từ xa và học tập hỗn hợp. Sự kết hợp của các phương pháp này mang lại sự linh hoạt và tiện lợi cho người học trong quá trình tiếp thu kiến thức. Dựa theo các tiêu chí trên, đánh giá theo hình thức học tập phù hợp, hệ thống học tập này được lựa chọn phương pháp học tập qua video kết hợp hướng dẫn từ xa, làm bài tập và thảo luận trực tuyến.

### **2.3. Mô hình giảng dạy trực tuyến**

Có nhiều mô hình giảng dạy trực tuyến được sử dụng để tạo ra trải nghiệm học tập tốt nhất cho người học. Dưới đây là một số mô hình phổ biến trong giảng dạy trực tuyến:

1. Mô hình học tự chủ: Mô hình này tập trung vào việc tự học của người học. Người học có quyền tự chủ về thời gian và tốc độ học tập. Họ có thể truy cập vào các tài liệu học tập và nguồn kiến thức, hoàn thành các bài tập và kiểm tra, và tiếp thu kiến thức theo tốc độ riêng. Giảng viên chủ yếu đóng vai trò là nguồn cung cấp tài liệu và hỗ trợ qua các hệ thống truyền thông trực tuyến.
2. Mô hình học nhóm: Mô hình này tập trung vào việc học tập trong nhóm. Người học được phân vào các nhóm học tập nhỏ, trong đó họ tương tác và hợp tác trong việc tiếp thu kiến thức. Họ có thể thảo luận, chia sẻ ý kiến, và làm việc cùng nhau

để hoàn thành các bài tập và dự án. Giảng viên đóng vai trò là người hướng dẫn và điều phối các hoạt động học tập của nhóm.

3. Mô hình học trực tuyến theo thời gian thực: Mô hình này đưa người học và giảng viên vào một môi trường trực tuyến trong thời gian thực. Thông qua các công nghệ truyền thông như video cuộc gọi, trò chuyện video và hộp thư thoại, người học và giảng viên có thể giao tiếp trực tiếp và tương tác như trong một lớp học truyền thống. Giảng viên có thể giảng dạy, giải đáp câu hỏi và cung cấp hướng dẫn cho người học trong thời gian thực.
4. Mô hình học kết hợp: Mô hình này kết hợp giữa học trực tuyến và học truyền thống. Người học có thể tham gia vào các buổi học trực tuyến và cũng có thể tham gia vào các buổi học tại một địa điểm vật lý. Các buổi học trực tuyến có thể được sử dụng để cung cấp nội dung học tập, thảo luận và tương tác trực tuyến. Các buổi học truyền thống có thể sử dụng để thực hiện các hoạt động thực hành, thảo luận trực tiếp và kiểm tra.
5. Mô hình học tương tác: Mô hình này tập trung vào việc tạo ra một môi trường học tập tương tác và thảo luận. Thông qua các công nghệ truyền thông như diễn đàn, thảo luận trực tuyến và email, người học có thể giao tiếp với giảng viên và đồng học viên. Họ có thể chia sẻ ý kiến, đặt câu hỏi và nhận được phản hồi từ giảng viên và đồng học viên, tạo ra một môi trường học tập sôi nổi và tương tác.

Tóm lại, có nhiều mô hình giảng dạy trực tuyến được sử dụng như học tự chủ, học nhóm, học trực tuyến theo thời gian thực, học kết hợp và học tương tác. Mỗi mô hình có đặc điểm riêng và phù hợp với các mục tiêu học tập và ngữ cảnh cụ thể. Sự linh hoạt và tiện lợi của học trực tuyến cho phép giảng viên và người học lựa chọn mô hình phù hợp nhằm đảm bảo việc học tập hiệu quả và tạo ra trải nghiệm học tập tích cực.

Dựa trên khái niệm về các mô hình học tập từ đó ánh xạ sang hệ thống học lập trình trực tuyến này, hệ thống sẽ áp dụng kết hợp giữa các mô hình với nhau cụ thể như mô hình tự chủ, mô hình học nhóm, mô hình học tương tác. Đối với mô hình tự chủ, người dùng có thể tự chọn video học tập mà mình muốn học, xem tài liệu mình cần. Đối với mô hình học nhóm, người dùng sẽ tham gia vào từng khóa học riêng biệt, với mô hình tương tác, các bạn học có thể trao đổi với nhau thông qua kênh chat chung của từng bài học.

## 2.4. Quản lý người dùng và khoá học

1. Đăng ký và đăng nhập người dùng: Hệ thống sẽ sử dụng đăng nhập thông qua tài khoản Google của người dùng sau đó sử dụng trình xác thực JWT để mã hoá thông tin và bảo mật thông tin.

2. Quản lý hồ sơ người dùng: Hồ sơ người dùng gồm có ảnh đại diện, họ tên, email, link mạng xã hội và mô tả về bản thân, công việc.

3. Quản lý khóa học: Khoá học được tạo bởi bất kì người dùng nào đều phải đăng kí được admin chấp nhận mới được công khai cho người dùng đăng kí học tập và mua bán. Khoá học được công khai người dùng mới được tham gia học tập, đối với khoá học miễn phí, người dùng phải có được sự cho phép của chủ khoá học mới có thể tham gia vào khoá học.

4. Tiến độ học tập và quản lý khóa học của người dùng: Lưu trữ khoá học của đã tham gia của mỗi người, thể hiện tiến độ phần trăm học tập của học viên trên mỗi khoá học mình tham gia

5. Giao tiếp và tương tác giữa người dùng và giảng viên và bạn học khác: Trong mỗi bài học sẽ có bài tập và phần thảo luận, các học viên có thể hỏi đáp và nộp bài tập trong phần thảo luận của mỗi bài và các bạn học viên khác và giáo viên sẽ tương tác tại đó hoặc học viên có thể hỏi thông qua email hoặc mạng xã hội riêng của giảng viên hay các bạn học viên khác.

## 2.5. Công nghệ và công cụ sử dụng

Để xây dựng một hệ thống công nghệ phần mềm thì có rất nhiều công nghệ và công cụ được sử dụng. Dưới đây là một số ví dụ:

1. Ngôn ngữ lập trình và framework:

- Ngôn ngữ lập trình phổ biến như JavaScript, Python, Java, Ruby, PHP, vv.
- Framework phát triển ứng dụng web như Node.js, Django, Ruby on Rails, Laravel, vv.

2. Cơ sở dữ liệu:

- Cơ sở dữ liệu quan hệ như MySQL, PostgreSQL, Oracle, SQL Server.
- Cơ sở dữ liệu NoSQL như MongoDB, Firebase, Cassandra.

3. Công cụ phát triển và triển khai:

- Visual Studio Code: Môi trường phát triển tích hợp (IDE) cho nhiều ngôn ngữ lập trình.
- Git/GitHub/Gitlab: Quản lý phiên bản mã nguồn và hợp tác phát triển.
- Docker: Platform để đóng gói và chạy ứng dụng trong một môi trường cô lập.
- Amazon Web Services (AWS): Nền tảng điện toán đám mây cung cấp dịch vụ triển khai và quản lý ứng dụng.

4. Công nghệ web:

- HTML, CSS và JavaScript: Ngôn ngữ và công nghệ cơ bản để xây dựng giao diện và tương tác trên web.
- React, Angular, Vue.js: Framework JavaScript phổ biến cho phát triển ứng dụng web động.

Dựa trên thế mạnh và kiến thức của bản thân, sau khi phân tích giữa các công cụ và phát triển, hệ thống học lập trình được xây dựng dựa trên ngôn ngữ lập trình

Javascript kết hợp Typescript. Dưới đây là các công nghệ và công cụ dùng để phát triển phần mềm:

1. Nx Monorepo:

- Sử dụng Nx Monorepo để quản lý mã nguồn và triển khai các ứng dụng và thư viện.
- Nx Monorepo cung cấp các công cụ và quy trình tối ưu hóa cho phát triển và triển khai ứng dụng trong môi trường monorepo.

2. OpenAPI Generator:

- Sử dụng OpenAPI Generator để tạo mã và tài liệu API dựa trên các tệp OpenAPI.
- OpenAPI Generator giúp tự động hóa việc tạo mã từ mô tả API và giúp tạo ra tài liệu API theo cấu trúc chuẩn.

3. Docker:

- Sử dụng Docker để đóng gói ứng dụng vào các container.
- Docker cho phép đảm bảo sự độc lập và di động của ứng dụng khi triển khai trên môi trường khác nhau.

4. AWS (Amazon Web Services):

- Triển khai hệ thống lên AWS, nền tảng điện toán đám mây.
- AWS cung cấp các dịch vụ và giải pháp để triển khai, quản lý và mở rộng ứng dụng của bạn trên nền tảng điện toán đám mây.

5. Frontend:

- Sử dụng Next.js để phát triển giao diện người dùng.
- Sử dụng bộ thư viện MUI (Material-UI) để xây dựng giao diện với các thành phần giao diện chuẩn Material Design.

6. Backend:

- Sử dụng Nest.js để phát triển backend.
- Sử dụng trình xác thực JWT (JSON Web Tokens).

7. Firebase.

- Sử dụng FCM firebase để nhận thông báo.

8. Database

- Cơ sở dữ liệu postgresql.
- MikroOrm.

Hệ thống sử dụng các công nghệ và công cụ tiên tiến để xây dựng một hệ thống phân tán, linh hoạt và bảo mật.

## 2.6. Cơ sở lý thuyết về công nghệ sử dụng

## 2.6.1. Nx Monorepo



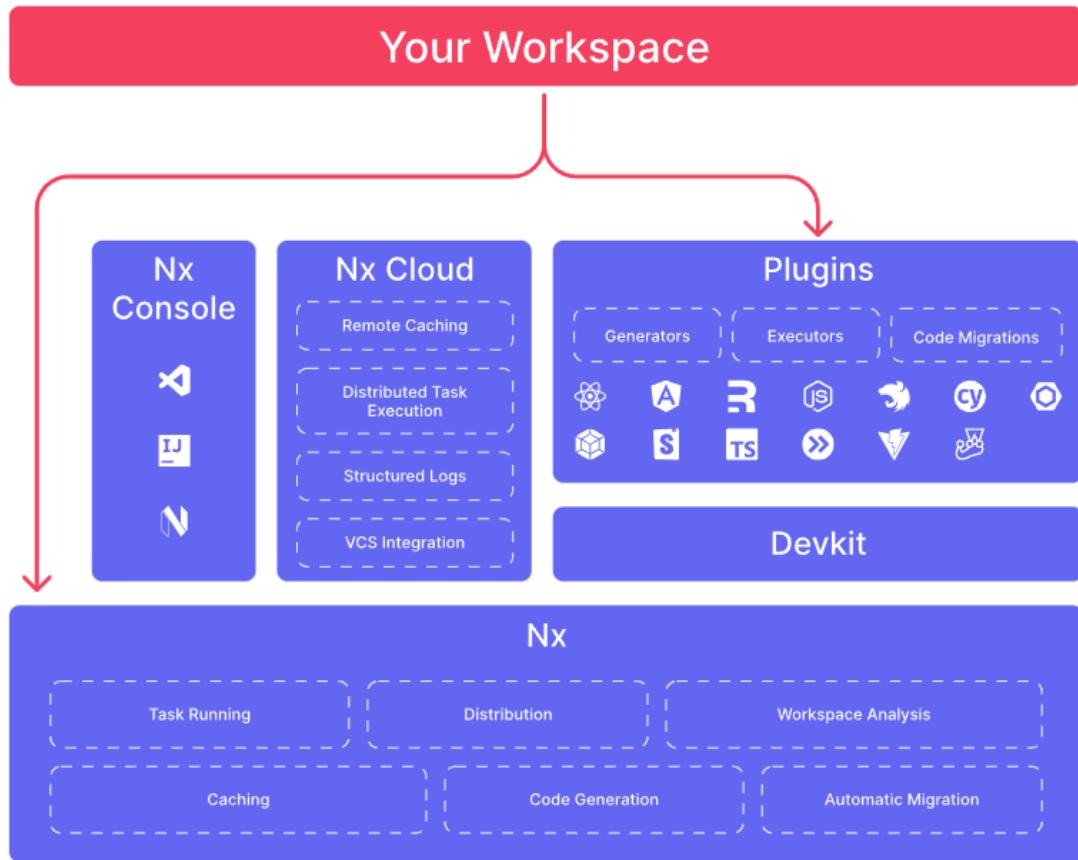
Hình 2.1. Logo Nx monorepo

Nx là một hệ thống xây dựng mã nguồn mở mạnh mẽ cung cấp các công cụ và kỹ thuật để nâng cao năng suất của nhà phát triển, tối ưu hóa hiệu suất CI và duy trì chất lượng mã. Nx có thể nhanh chóng dàn dựng một dự án mới hoặc thậm chí toàn bộ một monorepo. Nó có thể được áp dụng dần dần và sẽ phát triển cùng với dự án của bạn khi nó mở rộng quy mô.

Nx giúp: tăng tốc tính toán của bạn (ví dụ: bản dựng, thử nghiệm, v.v.), cục bộ và trên CI, đồng thời tích hợp và tự động hóa công cụ của bạn thông qua các plugin của nó. Tất cả điều này có thể được thông qua dần dần. Bạn có thể sử dụng plugin, nhưng bạn không cần phải làm vậy. Xem kiến trúc Nx trong phần tiếp theo để tìm hiểu thêm. Dưới đây là lí do sử dụng Nx:

- Tăng tốc quá trình xây dựng và kiểm tra dự án hiện tại của bạn, cục bộ và trên CI (cho dù đó là ứng dụng đơn lẻ hay ứng dụng độc lập).
- Nhanh chóng xây dựng một dự án mới (sử dụng plugin Nx) mà không cần phải định cấu hình bất kỳ công cụ xây dựng cấp thấp nào.
- Dễ dàng tích hợp công cụ mới (ví dụ: Storybook, Tailwind, v.v.) vào dự án của bạn.
- Đảm bảo tính nhất quán và chất lượng mã với trình tạo tùy chỉnh và quy tắc lint.
- Cập nhật các khung và công cụ của bạn và giữ cho không gian làm việc của bạn luôn xanh tươi bằng cách sử dụng tính năng “migration” tự động.

Nx hoạt động theo quy trình sau:



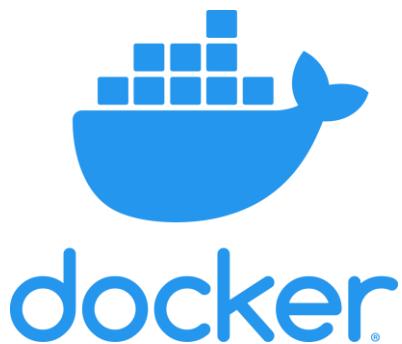
Hình 2.2. Quy trình hoạt động của Nx

- Gói Nx cung cấp các khả năng cơ bản về công nghệ chặng hạn như: **workspace analysis, task running, caching, distribution, code generation and automated code migrations**.
- Plugin là các gói NPM được xây dựng dựa trên các khả năng cơ bản do gói Nx cung cấp. Các plugin Nx chứa trình tạo mã, trình thực thi (để trù trừ tượng hóa công cụ xây dựng cấp thấp hơn) và di chuyển mã tự động để giữ cho công cụ của bạn luôn cập nhật. Trái ngược với gói Nx, hoạt động theo cùng một cách với bất kỳ dự án JS hoặc không phải JS nào, các plugin thường dành riêng cho công nghệ. Chẳng hạn, @nx/react thêm hỗ trợ để xây dựng ứng dụng React và lib, @nx/cypress thêm khả năng thử nghiệm e2e với Cypress. Các plugin giúp các nhà phát triển làm việc hiệu quả hơn bằng cách loại bỏ mọi trở ngại trong việc tích hợp các công cụ khác nhau với nhau và bằng cách cung cấp các tiện ích để luôn cập nhật chúng. Nhóm Nx duy trì các plugin cho React, Next, Remix, Angular, Jest, Cypress, Storybook, v.v. Bạn có thể sử dụng @nx/plugin gói này để dễ dàng tạo một plugin mới hoặc thậm chí chỉ tự động hóa không gian làm việc cục bộ của mình. Ngoài ra còn có hơn 80 plugin cộng đồng.
- Devkit là một tập hợp các tiện ích để xây dựng các plugin Nx .

- Nx Cloud giúp mở rộng quy mô dự án của bạn trên CI bằng cách thêm bộ nhớ đệm từ xa và thực thi tác vụ phân tán . Nó cũng cải thiện công thái học của nhà phát triển bằng cách tích hợp với GitHub, GitLab và BitBucket và cung cấp nhật ký có cấu trúc có thể tìm kiếm được. Tìm hiểu thêm tại nx.app.
- Bảng điều khiển Nx là phần mở rộng cho VSCode, IntelliJ và VIM . Nó cung cấp tính năng tự động hoàn thành mã, trình tạo tương tác, trực quan hóa không gian làm việc, tái cấu trúc mạnh mẽ.

Ứng dụng của Nx monorepo chính là xây dựng cấu trúc của dự án, xây dựng base dự án một cách nhanh chóng chuẩn cấu trúc. Quản lý tất cả các app nhỏ, chia sẻ code chung giữa các app nhỏ với nhau. Nx cung cấp các quy trình xây dựng và quản lý chặt chẽ giúp code của bạn luôn chuẩn định dạng với 3 quy trình format\_check, test, build.

### 2.6.2. Docker



Hình 2.3. Logo docker

Docker là nền tảng phần mềm cho phép bạn dựng, kiểm thử và triển khai ứng dụng một cách nhanh chóng. Docker đóng gói phần mềm vào các đơn vị tiêu chuẩn hóa được gọi là container có mọi thứ mà phần mềm cần để chạy, trong đó có thư viện, công cụ hệ thống, mã và thời gian chạy. Bằng cách sử dụng Docker, bạn có thể nhanh chóng triển khai và thay đổi quy mô ứng dụng vào bất kỳ môi trường nào và biết chắc rằng mã của bạn sẽ chạy được.

Docker hoạt động bằng cách cung cấp phương thức tiêu chuẩn để chạy mã của bạn. Docker là hệ điều hành dành cho container. Cũng tương tự như cách máy ảo hóa (loại bỏ nhu cầu quản lý trực tiếp) phần cứng máy chủ, các container sẽ ảo hóa hệ điều hành của máy chủ. Docker được cài đặt trên từng máy chủ và cung cấp các lệnh đơn giản mà bạn có thể sử dụng để dựng, khởi động hoặc dừng container.

Việc sử dụng Docker cho phép bạn vận chuyển mã nhanh hơn, tiêu chuẩn hóa hoạt động của ứng dụng, di chuyển mã một cách trọn vẹn và tiết kiệm tiền bằng cách cải thiện khả năng tận dụng tài nguyên. Với Docker, bạn sẽ được nhận một đối tượng duy nhất có khả năng chạy ổn định ở bất kỳ đâu. Cú pháp đơn giản và không phức tạp của Docker sẽ cho bạn quyền kiểm soát hoàn toàn. Việc đưa vào áp dụng rộng rãi

nền tảng này đã tạo ra một hệ sinh thái bền vững các công cụ và ứng dụng có thể sử dụng ngay đã sẵn sàng sử dụng với Docker.

- Vận chuyển phần mềm nhiều hơn và nhanh hơn: Người dùng sử dụng Docker vận chuyển phần mềm nhanh hơn trung bình 7 lần so với người dùng không sử dụng Docker. Docker đem đến cho bạn khả năng vận chuyển dịch vụ được tách riêng với tần suất mong muốn.
- Tiêu chuẩn hóa quá trình vận hành: Ứng dụng được đóng gói vào container nhỏ sẽ khiến cho việc triển khai, xác định vấn đề và đảo ngược để khắc phục trở nên dễ dàng.
- Di chuyển trọn vẹn: Ứng dụng trên nền tảng Docker có thể được di chuyển trọn vẹn từ các máy phát triển cục bộ đến đơn vị triển khai sản xuất trên AWS.
- Tiết kiệm tiền bạc: Container Docker giúp cho việc chạy nhiều mã hơn trên từng máy chủ trở nên dễ dàng hơn, cải thiện khả năng tận dụng và tiết kiệm tiền bạc cho bạn.

Ứng dụng của docker trong dự án chính là việc xây dựng các service một cách dễ dàng, không yêu cầu phải cài đặt quá nhiều các ứng dụng, tạo ra môi trường ảo hóa, chạy các container độc lập không phụ thuộc lẫn nhau giúp giảm thiểu thời gian, chi phí xây dựng hệ thống.

### 2.6.3. AWS



Hình 2.4. Logo AWS

Amazon Web Services (AWS) là giải pháp đám mây toàn diện và được sử dụng rộng rãi nhất, cung cấp trên 200 dịch vụ đầy đủ tính năng từ các trung tâm dữ liệu trên toàn thế giới. Hàng triệu khách hàng – bao gồm các công ty khởi nghiệp tăng trưởng nhanh nhất, các tập đoàn lớn nhất cũng như các cơ quan hàng đầu của chính phủ – đều tin tưởng vào AWS để giảm chi phí, trở nên linh hoạt hơn và đổi mới nhanh hơn.

Amazon Elastic Compute Cloud (Amazon EC2) cung cấp khả năng điện toán theo yêu cầu, có thể thay đổi quy mô trong Đám mây Amazon Web Services (AWS). Sử dụng Amazon EC2 giúp giảm chi phí phần cứng để bạn có thể phát triển và triển khai ứng dụng nhanh hơn. Bạn có thể sử dụng Amazon EC2 để khởi chạy số lượng hoặc số lượng máy chủ ảo tùy theo nhu cầu của mình, định cấu hình bảo mật và kết nối mạng cũng như quản lý dung lượng lưu trữ. Bạn có thể thêm dung lượng (tăng

quy mô) để xử lý các tác vụ nặng về điện toán, chẳng hạn như quy trình hàng tháng hoặc hàng năm hoặc lưu lượng truy cập trang web tăng đột biến. Khi mức sử dụng giảm, bạn có thể giảm lại dung lượng (giảm quy mô).

Các tính năng của Amazon EC2:

- Instances: Máy chủ ảo.
- Amazon Machine Images (AMIs): Các mẫu được cấu hình sẵn cho các phiên bản của bạn đóng gói các thành phần bạn cần cho máy chủ của mình (bao gồm cả hệ điều hành và phần mềm bổ sung).
- Instance types: Các cấu hình khác nhau của CPU, bộ nhớ, bộ lưu trữ, dung lượng mạng và phần cứng đồ họa cho các phiên bản của bạn.
- Key pairs: Thông tin đăng nhập an toàn cho các phiên bản của bạn. AWS lưu trữ khóa chung và bạn lưu trữ khóa riêng ở một nơi an toàn.
- Instance store volumes: Dung lượng lưu trữ cho dữ liệu tạm thời sẽ bị xóa khi bạn dừng, ngủ đông hoặc chấm dứt phiên bản của mình.
- Amazon EBS volumes: Khối lượng lưu trữ liên tục cho dữ liệu của bạn bằng cách sử dụng Amazon Elastic Block Store (Amazon EBS).
- Regions, Availability Zones, Local Zones, AWS Outposts, and Wavelength Zones: Nhiều vị trí thực cho tài nguyên của bạn, chẳng hạn như phiên bản và ổ đĩa Amazon EBS.
- Security groups: Tường lửa ảo cho phép bạn chỉ định các giao thức, cổng và dải IP nguồn có thể tiếp cận các phiên bản của bạn và dải IP đích mà các phiên bản của bạn có thể kết nối.
- Elastic IP addresses: Địa chỉ IPv4 tĩnh cho điện toán đám mây động.
- Tags: Siêu dữ liệu mà bạn có thể tạo và gán cho tài nguyên Amazon EC2 của mình.
- Virtual private clouds (VPCs): Các mạng ảo mà bạn có thể tạo được tách biệt về mặt logic với phần còn lại của Đám mây AWS. Bạn có thể tùy ý kết nối các mạng ảo này với mạng riêng của mình.

Ứng dụng của AWS đối với hệ thống là xây dựng một máy chủ từ xa như một máy chủ vật lý có thể chạy 24/24 để vận hành hệ thống liên tục. Cho phép người dùng có thể truy cập vào hệ thống ở bất cứ đâu, tại mọi thời điểm.

#### 2.6.4. Next.js



Hình 2.5. Logo Next.js

Next.js là framework React cung cấp cho bạn các khối xây dựng để tạo các ứng dụng web. Next.js chủ yếu được sử dụng để xây dựng các ứng dụng web phía máy chủ và phía máy khách.

Ưu điểm của Next.js:

- Cho phép tối ưu hóa SEO, giúp tăng khả năng tìm thấy trên các công cụ tìm kiếm.
- Hỗ trợ server-side rendering, giúp tăng tốc độ tải trang và trải nghiệm người dùng.
- Có nhiều tùy chọn để quản lý trạng thái ứng dụng.

Next.js và Nuxt.js đều là các framework JavaScript để phát triển ứng dụng web trên nền tảng Node.js và React. Tuy nhiên Nuxt.js chủ yếu được sử dụng để xây dựng các ứng dụng web tĩnh và ứng dụng web phía máy khách.

Ứng dụng của Next.js chính là xây dựng giao diện website giúp người dùng tương tác với hệ thống một cách dễ dàng, đơn giản và trực quan.

#### 2.6.5. Nest.js



Hình 2.6. Logo Nest.js

Nest (NestJS) là một khung để xây dựng các ứng dụng phía máy chủ Node.js hiệu quả, có thể mở rộng. Nó sử dụng JavaScript lũy tiến, được xây dựng và hỗ trợ

đầy đủ TypeScript (nhưng vẫn cho phép các nhà phát triển viết mã bằng JavaScript thuận tủy) và kết hợp các yếu tố của OOP (Lập trình hướng đối tượng), FP (Lập trình chức năng) và FRP (Lập trình phản ứng chức năng).

Về cơ bản, Nest sử dụng các khung Máy chủ HTTP mạnh mẽ như Express (mặc định) và có thể được định cấu hình tùy chọn để sử dụng cả Fastify!

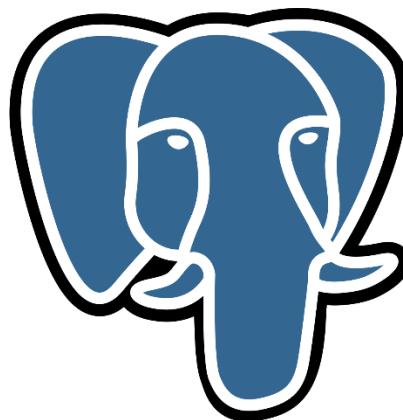
Nest cung cấp mức độ trùu tượng cao hơn các khung Node.js phổ biến này (Express/Fastify), nhưng cũng hiển thị trực tiếp các API của chúng cho nhà phát triển. Điều này cho phép các nhà phát triển tự do sử dụng vô số mô-đun của bên thứ ba có sẵn cho nền tảng cơ bản.

Trong những năm gần đây, nhờ có Node.js, JavaScript đã trở thành “ngôn ngữ chung” của web cho cả ứng dụng front-end và back-end. Điều này đã tạo ra các dự án tuyệt vời như Angular, React và Vue, giúp cải thiện năng suất của nhà phát triển và cho phép tạo các ứng dụng giao diện người dùng nhanh, có thể kiểm tra và có thể mở rộng. Tuy nhiên, trong khi có rất nhiều thư viện, trình trợ giúp và công cụ tuyệt vời dành cho Node (và JavaScript phía máy chủ), không có thư viện nào giải quyết hiệu quả vấn đề chính của - Kiến trúc.

Nest cung cấp một kiến trúc ứng dụng vượt trội cho phép các nhà phát triển và nhóm tạo ra các ứng dụng có thể kiểm tra cao, có thể mở rộng, liên kết lỏng lẻo và dễ bảo trì. Kiến trúc được lấy cảm hứng rất nhiều từ Angular.

Ứng dụng của Nest.Js chính là xây dựng một server giúp tiếp nhận các thông tin nhận được từ người dùng gửi đến và kết nối tới cơ sở dữ liệu để truy xuất thông tin phản hồi lại yêu cầu của khách hàng.

#### 2.6.6. Postgresql



Hình 2.7. Logo Posgresql

PostgreSQL là một hệ thống cơ sở dữ liệu quan hệ đối tượng mã nguồn mở, mạnh mẽ, sử dụng và mở rộng ngôn ngữ SQL kết hợp với nhiều tính năng giúp lưu trữ và thay đổi quy mô khối lượng công việc dữ liệu phức tạp nhất một cách an toàn. Nguồn gốc của PostgreSQL bắt đầu từ năm 1986 như một phần của dự án

POSTGRES tại Đại học California ở Berkeley và đã có hơn 35 năm phát triển tích cực trên nền tảng cốt lõi.

PostgreSQL đã đạt được danh tiếng mạnh mẽ về kiến trúc, độ tin cậy, tính toàn vẹn của dữ liệu, bộ tính năng mạnh mẽ, khả năng mở rộng và sự công hiến của cộng đồng nguồn mở đăng sau phần mềm để cung cấp các giải pháp sáng tạo và hiệu quả một cách nhất quán. PostgreSQL chạy trên tất cả các hệ điều hành chính, tuân thủ ACID từ năm 2001 và có các tiện ích bổ sung mạnh mẽ như bộ mở rộng cơ sở dữ liệu không gian địa lý PostGIS phổ biến. Không có gì ngạc nhiên khi PostgreSQL đã trở thành cơ sở dữ liệu quan hệ nguồn mở được nhiều người và tổ chức lựa chọn.

Các tính năng của postgresql:

*Bảng 2.1. Bảng tính năng của postgresql*

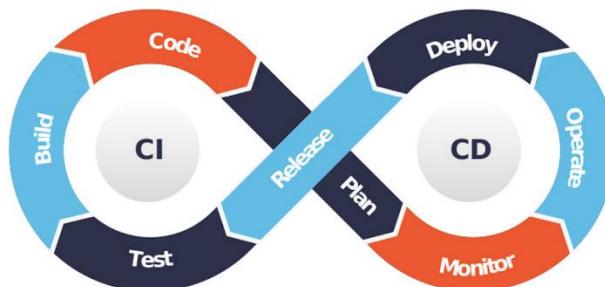
Tính năng	Chi tiết
Data Types (Loại dữ liệu)	Nguyên thủy: Integer, Numeric, String, Boolean. Structured: Date/Time, Array, Range / Multirange, UUID. Document: JSON/JSONB, XML, Key-value (Hstore). Hình học: Point, Line, Circle, Polygon. Tùy chỉnh: Tổng hợp, Loại tùy chỉnh.
Data Integrity (Toàn vẹn dữ liệu)	UNIQUE, NOT NULL. Primary Keys. Foreign Keys. Exclusion Constraints. Explicit Locks, Advisory Locks.
Concurrency, Performance (Đồng thời, Hiệu suất)	Lập chỉ mục: B-tree, Nhiều cột, Biểu thức, Một phần. Lập chỉ mục nâng cao: GiST, SP-Gist, KNN Gist, GIN, BRIN, Chỉ mục bao gồm, Bộ lọc Bloom. Công cụ lập kế hoạch / trình tối ưu hóa truy vấn tinh vi, quét chỉ mục, thống kê nhiều cột. Giao dịch, Giao dịch lồng nhau (qua điểm lưu trữ). Kiểm soát đồng thời nhiều phiên bản (MVCC). Song song hóa các truy vấn đọc và xây dựng chỉ mục cây. Phân vùng bảng. Tất cả các mức độ giao dịch được xác định trong tiêu chuẩn SQL, bao gồm cả Serializable. Biên dịch các biểu thức đúng lúc (JIT).
Reliability, Disaster Recovery ( Độ tin cậy, Khôi phục)	Ghi nhật ký ghi trước (WAL). Sao chép: Không đồng bộ, Đồng bộ, Hợp lý. Phục hồi tại thời điểm (PITR), chế độ chờ tích cực. Không gian bảng.
Security (Bảo vệ)	Xác thực: GSSAPI, SSPI, LDAP, SCRAM-SHA-256, Chứng chỉ, v.v. Hệ thống kiểm soát truy cập mạnh mẽ. Bảo mật cột và hàng. Xác thực đa yếu tố với chứng chỉ và một phương pháp bổ sung.
Extensibility (Khả năng mở rộng)	Các hàm và thủ tục lưu trữ.

	<p>Ngôn ngữ thủ tục: PL/pgSQL, Perl, Python và Tcl. Có các ngôn ngữ khác có sẵn thông qua các tiện ích mở rộng, ví dụ: Java, JavaScript (V8), R, Lua và Rust.</p> <p>Biểu thức đường dẫn SQL/JSON.</p> <p>Trình bao bọc dữ liệu nước ngoài: kết nối với các cơ sở dữ liệu hoặc luồng khác bằng giao diện SQL tiêu chuẩn.</p> <p>Giao diện lưu trữ có thể tùy chỉnh cho các bảng.</p> <p>Nhiều tiện ích mở rộng cung cấp chức năng bổ sung, bao gồm cả PostGIS.</p>
Internationalisation, Text Search (Quốc tế hóa, tìm kiếm văn bản)	<p>Hỗ trợ các bộ ký tự quốc tế, ví dụ như thông qua bộ sưu tập ICU.</p> <p>Các đối chiếu không phân biệt chữ hoa chữ thường và phân biệt dấu.</p> <p>Tìm kiếm toàn văn.</p>

PostgreSQL và MySQL đều là các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến được sử dụng trong các ứng dụng web và doanh nghiệp. Tuy nhiên MySQL không hỗ trợ tất cả các kiểu dữ liệu phức tạp như PostgreSQL, ví dụ như GIS. Bảo mật có thể kém hơn PostgreSQL và không cung cấp nhiều tính năng bảo mật và phân quyền trên từng bảng và truy vấn.

Ứng dụng của PostgreSQL trong hệ thống chính là nơi lưu trữ tất cả các thông tin dữ liệu từ người dùng gửi yêu cầu lên server.

#### 2.6.7. CI/CD



Hình 2.8. Quy trình CI/CD

CI/CD là Tích hợp liên tục (CI) và Phân phối liên tục (CD) là các quy trình nhằm tạo ra một quy trình phát triển, thử nghiệm và triển khai. Với CI/CD, tất cả các nhà phát triển có thể làm việc đồng thời trên mã của họ với ít chi phí. Điều này có nghĩa là có ít lỗi hơn trong sản phẩm cuối cùng vì mỗi lần lặp trình viên luôn làm việc với phiên bản cập nhật của cơ sở mã dự án.

Lợi ích của CI/CD:

- Thời gian chờ đợi kiểm tra thủ công ít hơn do các kiểm tra tự động chạy liên tục (CI) hoặc sau mỗi lần xác nhận (CD).

- Ít xung đột hơn giữa các nhà phát triển vì tất cả họ đều làm việc trên bản sao kho lưu trữ của riêng họ trong suốt quá trình phát triển không có "của tôi" so với "của bạn".
- Triển khai nhanh hơn nhờ các công cụ tự động viết và thực thi các tập lệnh khi đáp ứng các điều kiện nhất định trong quá trình xây dựng.
- Nâng cao chất lượng phần mềm.
- Tăng tốc độ phát triển phần mềm.

Ứng dụng của CI/CD đối với hệ thống là việc sử dụng gitlab tự động kết nối với máy chủ ec2 để thực hiện xây dựng image docker và từ máy chủ ec2 kéo ảnh docker từ gitlab về thực thi chạy phần mềm.

## **2.7. Kết luận**

Trong chương hai này em đã đi tìm hiểu cơ sở lý thuyết để xây dựng hệ thống học lập trình trực tuyến. Tìm hiểu về học trực tuyến và xu hướng hiện nay, các phương pháp học trực tuyến , các mô hình giảng dạy trực tuyến và tìm hiểu lý thuyết các công nghệ sử dụng để xây dựng hệ thống học lập trình trực tuyến này. Qua tìm hiểu để nắm sơ lược về hệ thống và đi đến chương tiếp theo “Thiết kế và Triển khai hệ thống”.

## CHƯƠNG 3: THIẾT KẾ & TRIỂN KHAI HỆ THỐNG

### 3.1. Thiết kế hệ thống

#### 3.1.1. Thu thập yêu cầu

Dựa trên yêu cầu đối với hệ thống học lập trình trực tuyến, ta có thể xác định các yêu cầu chính như sau:

##### 1. Quản lý khoá học:

- Hệ thống cho phép người dùng tạo, chỉnh sửa và xóa khoá học.
- Khoá học sẽ ở trạng thái "chờ" cho đến khi được admin duyệt để công khai.
- Khoá học có thể là miễn phí hoặc có phí.
- Chủ khoá học có quyền thay đổi thông tin của khoá học của mình.
- Khoá học gồm nhiều chương, mỗi chương gồm nhiều bài giảng.
- Khoá học được quản lý bởi chính người tạo ra khoá học.

##### 2. Xác thực người dùng:

- Người dùng đăng nhập bằng tài khoản Google để xác thực.
- Sử dụng JWT (JSON Web Tokens) để mã hoá thông tin xác thực người dùng.
- Cho phép xem thông tin các khoá học khi chưa đăng nhập, nhưng chỉ được tham gia khi đã đăng nhập.

##### 3. Quản lý tài khoản người dùng:

- Mỗi người dùng khi đăng nhập lần đầu sẽ được tạo một ví cá nhân với số dư ban đầu là 0 đồng.
- Người dùng có thể nạp tiền vào ví để mua khoá học với những khoá học trả phí.
- Rút tiền ra khỏi ví của mình.
- Cập nhật thông tin cá nhân của mình trừ email.

##### 4. Quản lý tham gia khoá học:

- Người dùng có thể xin tham gia khoá học miễn phí và chờ sự cho phép của chủ khoá học.
- Người dùng có thể tham gia khoá học sau khi được chấp thuận.
- Người dùng có quyền đánh giá khoá học mà họ đã tham gia.

##### 5. Quản lý video bài giảng và thảo luận:

- Các video bài giảng sẽ được tải lên dưới dạng video cá nhân hoặc liên kết từ YouTube.
- Giáo viên có thể giao bài tập và học viên có thể tương tác với giáo viên và các học viên khác trong phần thảo luận của mỗi video bài giảng.

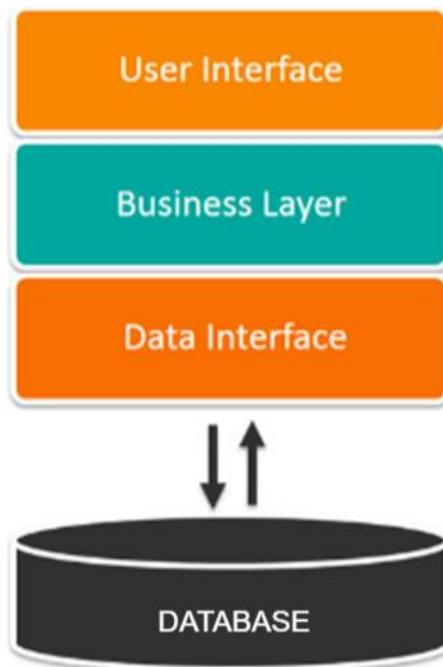
##### 6. Quản lý yêu cầu và thông báo:

- Cung cấp dịch vụ thông báo của FCM của firebase đến với người dùng

- Hệ thống gửi thông báo đến người dùng khi có yêu cầu duyệt khoá học, đồng ý hoặc từ chối yêu cầu, yêu cầu tham gia lớp học, đồng ý hoặc từ chối yêu cầu tham gia.
7. Chủ đề và danh mục cho khoá học:
- Hệ thống có phân chia theo danh mục chủ đề để mọi người dễ dàng tìm hiểu và tìm kiếm theo nhu cầu của mình.
  - Các khoá học sẽ gắn với các chủ đề để người dùng tìm kiếm dễ dàng.
8. Yêu cầu phi chức năng:
- Hiệu suất và đáp ứng: Hoạt động 24/24, tốc độ truy cập và khả năng xử lý nhanh.
  - Bảo mật & Quyền riêng tư: Đảm bảo dữ liệu và thông tin người dùng được bảo vệ khỏi việc truy cập trái phép.
  - Khả năng mở rộng: Dễ dàng mở rộng, tích hợp thêm các tính năng mới.
  - Tính khả dụng: Trang web dễ sử dụng, trải nghiệm người dùng tốt, giao diện thiết kế dễ nhìn, dễ đọc và thân thiện.

### 3.1.2. Thiết kế kiến trúc hệ thống

Hệ thống được xây dựng dựa trên kiến trúc “Monolithic”. Monolithic, hay còn được gọi là kiến trúc monolithic, là một mô hình kiến trúc phần mềm nguyên khối trong đó toàn bộ ứng dụng được xây dựng và triển khai như một hệ thống duy nhất.



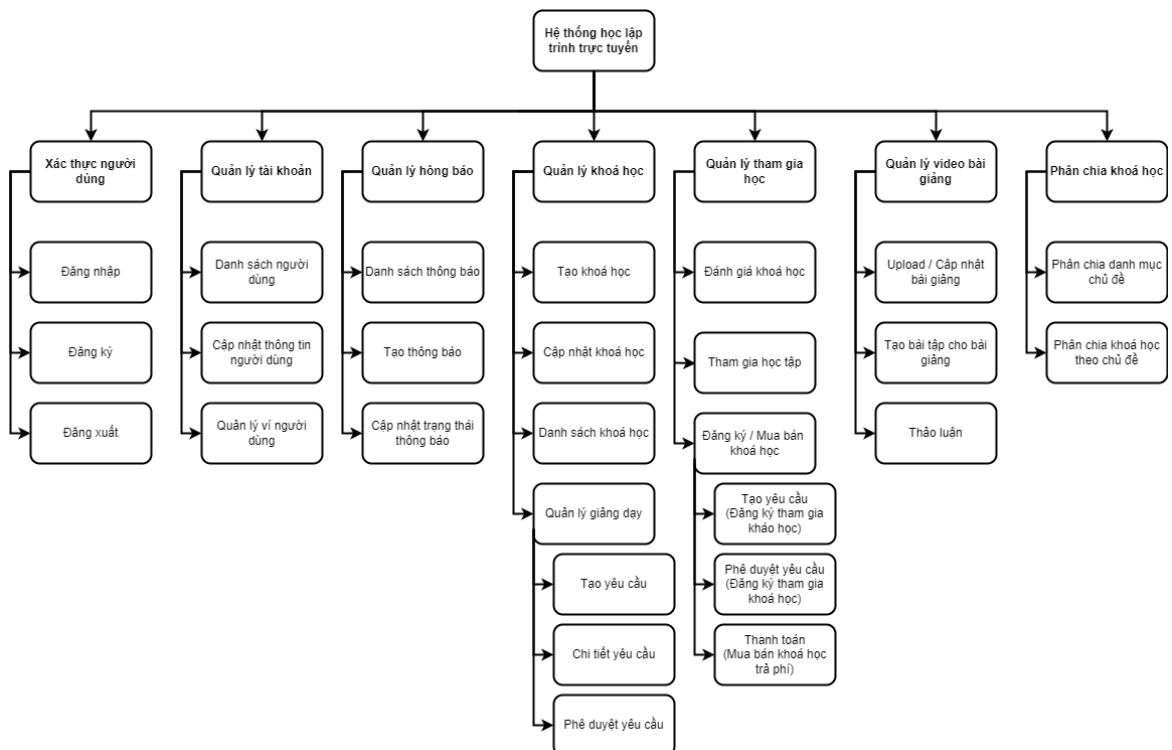
*Hình 3.1. Sơ đồ kiến trúc hệ thống*

Hình 3.1 là sơ đồ kiến trúc monothilic được hệ thống sử dụng. Sơ đồ gồm bốn tầng đó là “User Interface”, “Business Layer”, “Data Interface”, “Database”.

- User Interface (Giao diện người dùng): Tầng User Interface chịu trách nhiệm tạo giao diện tương tác với người dùng, cho phép họ tương tác và gửi các yêu cầu đến hệ thống. Khi người dùng tương tác với ứng dụng (chẳng hạn như truy cập trang web), các yêu cầu của họ được gửi đến tầng Business Layer.
- Business Layer (Logic): Tầng Business Layer thực hiện logic kinh doanh của ứng dụng. Đây là nơi xử lý các yêu cầu từ giao diện người dùng và thực hiện các tính toán, xử lý dữ liệu và logic kinh doanh liên quan. Sau khi xử lý, tầng Business Layer gửi các yêu cầu tương ứng đến tầng Data Interface.
- Data Interface (Giao diện dữ liệu): Tầng Data Interface giúp tương tác với cơ sở dữ liệu. Nó thực hiện việc truy vấn, cập nhật và xử lý dữ liệu từ và đến cơ sở dữ liệu. Khi tầng Data Interface nhận yêu cầu từ tầng Business Layer, nó sẽ thực hiện truy vấn cơ sở dữ liệu để lấy hoặc cập nhật thông tin cần thiết.
- Database (Cơ sở dữ liệu): Tầng Database là nơi lưu trữ dữ liệu của ứng dụng. Nó chứa các bảng, dòng và cột chứa thông tin cần thiết cho ứng dụng. Khi tầng Data Interface yêu cầu thực hiện truy vấn cơ sở dữ liệu, tầng Database sẽ trả về dữ liệu tương ứng cho tầng Data Interface để sau đó truyền lại kết quả cho tầng Business Layer.

Tóm lại, kiến trúc monolithic với 4 tầng hoạt động bằng cách tương tác giữa các tầng để xử lý yêu cầu người dùng, thực hiện logic, truy vấn và cập nhật dữ liệu từ cơ sở dữ liệu. Tất cả các tầng này hoạt động cùng nhau để tạo ra một ứng dụng hoàn chỉnh và chức năng. Ưu điểm của nó là tiết kiệm được thời gian phát triển và triển khai đơn giản, hiệu suất ứng dụng cũng cao hơn do các thành phần giao tiếp với nhau 1 cách trực tiếp, đồng thời bộ nhớ được chia sẻ. Tuy nhiên, nhược điểm của kiến trúc này là càng theo thời gian, khi lượng code, chức năng càng lớn thì hệ thống càng trở lên cồng kềnh và kém linh hoạt; chưa kể nếu một thành phần nhỏ trong cả hệ thống bị lỗi sẽ kéo theo việc toàn bộ hệ thống không thể hoạt động được.

### **3.1.3. Sơ đồ chức năng của hệ thống**



*Hình 3.2. Hình ảnh sơ đồ chức năng của hệ thống*

Dựa trên các yêu cầu, mục đích của hệ thống được đưa ra, hệ thống sẽ đưa ra chi tiết cụ thể về chức năng của hệ thống. Dưới đây là bản mô tả chi tiết từng chức năng của hệ thống:

1. Xác thực người dùng: Hệ thống sử dụng tài khoản google của người dùng để đăng nhập và đăng kí tài khoản và mã hoá thông tin người dùng bằng trình bảo mật JWT (jsonwebtoken). Sử dụng cookie để lưu trữ token, khi đăng xuất sẽ xoá bỏ cookie.
2. Quản lý tài khoản
  - Cập nhật thông tin tài khoản: Chức năng dành cho người dùng đăng nhập vào hệ thống muốn chỉnh sửa lại thông tin của mình, tuy nhiên người dùng sẽ không thể chỉnh sửa được email
  - Quản lý ví tiền: Quản lý dòng tiền vào ra của ví.
  - Danh sách người dùng: Danh sách người dùng trong hệ thống
3. Quản lý thông báo: Hệ thống cung đưa ra thông báo đến với người dùng để thông báo các hoạt động người dùng tham gia đã được xử lý giúp người dùng thuận tiện sử dụng dịch vụ. Đối với thông báo, hệ thống sử dụng thêm thông báo FCM đến từ nhà phát triển “Firebase Google” để người dùng có thể nhận được thông báo ngay cả khi người dùng không hoạt động trong trang hệ thống.
4. Quản lý khóa học:
  - Tạo khóa học: Chức năng yêu cầu người dùng phải đăng nhập hệ thống, không giới hạn bất kì thành viên nào có thể tạo khóa học với hình thức

miễn phí hoặc trả phí, mọi người đều có thể tạo khoá học và xin phép cấp quyền khoá học đưa vào giảng dạy.

- Cập nhật khoá học: Chức năng chỉ cho phép người tạo khoá học được cập nhật thông tin khoá học của chính mình.
- Quản lý giảng dạy: Chức năng dành cho admin để quản lý các yêu cầu cấp phép khoá học đến từ người dùng

## 5. Quản lý tham gia học

- Tham gia học tập khoá học: Xem nội dung khoá học.
- Đánh giá khoá học: Người dùng tham gia khoá học được đánh giá khoá học một lần duy nhất.
- Đăng ký / Mua bán khoá học: Tạo yêu cầu tham gia học đối với khoá học miễn phí, xem chi tiết yêu cầu, chủ khoá học phê duyệt yêu cầu xin tham gia. Đối với khoá học trả tiền, thanh toán mua bán khoá học bằng tiền trong ví tạo ra đơn giao dịch mua khoá học.

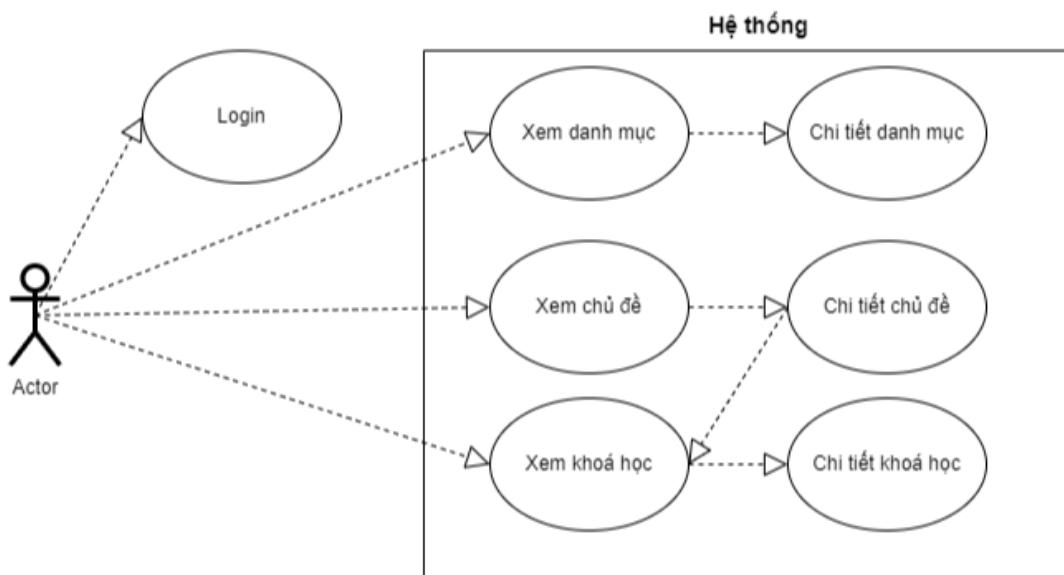
## 6. Quản lý video bài giảng

- Tạo bài giảng: Chủ khoá học tạo video bài giảng cho khoá học có thể dưới dạng tải video lên lưu trữ trên cloud aws của hệ thống hoặc gắn link video youtube của mình.
- Cập nhật bài giảng: Thay đổi nội dung hoặc video bài giảng.
- Bài tập: Với mỗi video bài giảng chủ khoá học có thể tạo 1 hoặc nhiều bài tập để học viên thực hành, chỉnh sửa bài tập đã tạo. Học viên chỉ xem được danh sách bài tập và thực hành nộp bài vào phần thảo luận của bài học.
- Thảo luận: Học viên tham gia khoá học có thể thảo luận trao đổi kiến thức bài học với nhau dưới dạng text trong phần thảo luận trong mỗi bài giảng.

## 7. Phân loại khoá học: Hệ thống xây dựng các khoá học được chia theo các danh mục, chủ đề giúp người dùng dễ dàng phân loại.

### 3.1.4. Sơ đồ Usecase

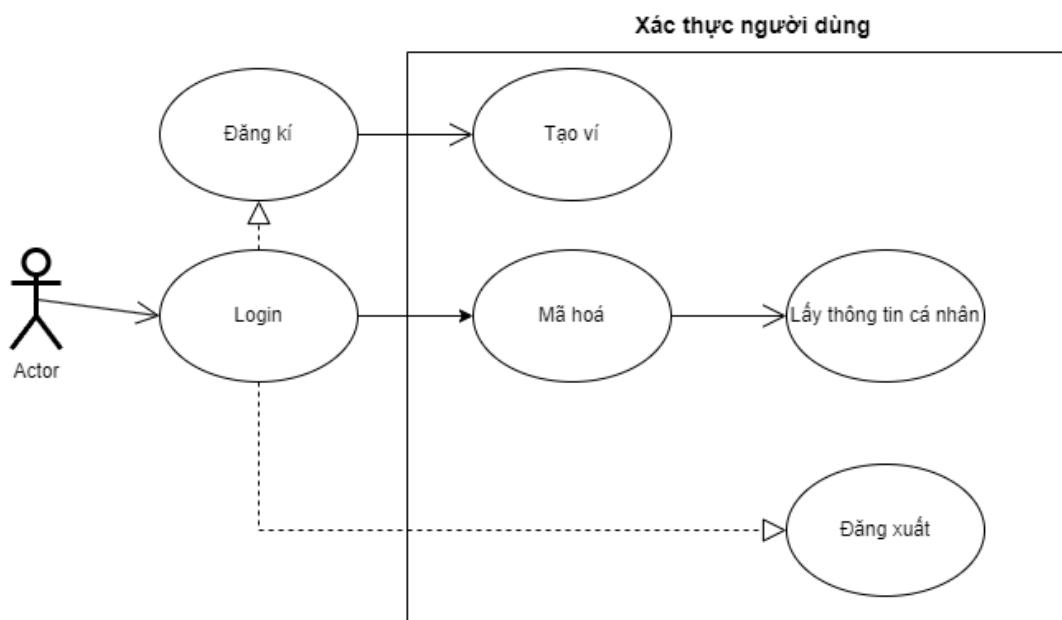
#### 1. Sơ đồ usecase dành cho chế độ khách



Hình 3.3. Hình ảnh sơ đồ usecase chế độ khách

Đặc tả: Đối với người dùng ở chế độ khách, không đăng nhập vào hệ thống có thể xem được tổng quan hệ thống, danh sách danh mục, chủ đề, chi tiết chủ đề, danh sách khoá học, chi tiết khoá học. Có thể đăng nhập vào hệ thống để thực hiện các tác vụ khác.

## 2. Sơ đồ usecase xác thực người dùng

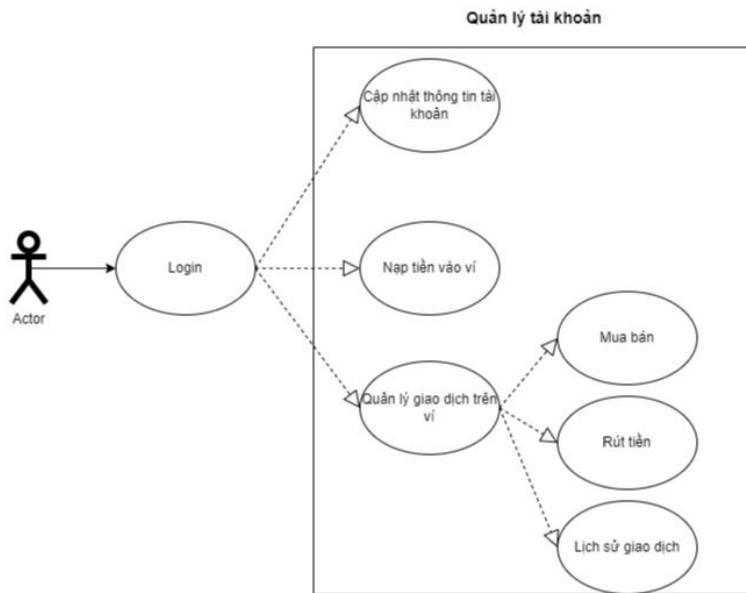


Hình 3.4. Hình ảnh usecase xác thực người dùng

Đặc tả: Người dùng đăng nhập vào hệ thống thông qua tài khoản của google, nếu người dùng đăng nhập lần đầu thì sẽ tự động đăng kí tài khoản và được tạo ví sau

đó quay lại đăng nhập và được mã hoá thông tin. Khi đã đăng nhập người dùng có thể đăng xuất tài khoản của mình trở về chế độ khách.

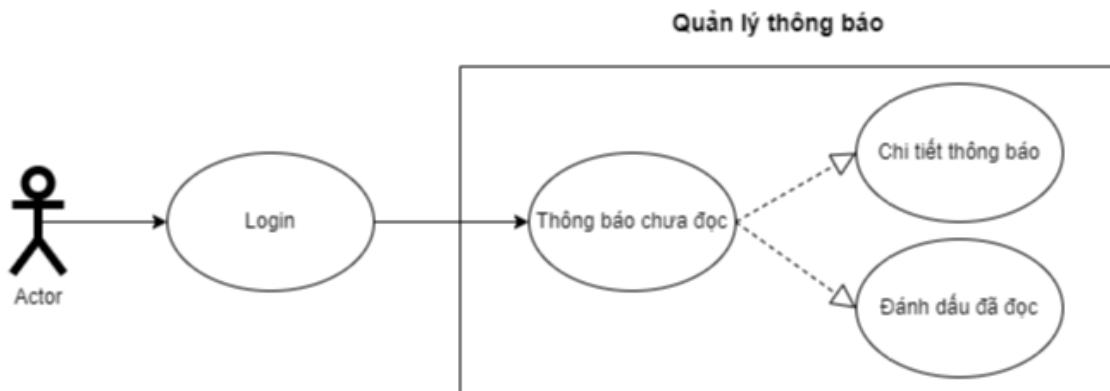
### 3. Sơ đồ usecase quản lý tài khoản



*Hình 3.5. Hình ảnh usecase quản lý tài khoản*

Đặc tả: Người dùng đăng nhập vào hệ thống có thể thay đổi, cập nhật thông tin cá nhân của mình. Thực hiện nạp tiền vào ví nếu có nhu cầu mua bán, trao đổi khoá học, quản lý các giao dịch trên ví của mình. Rút tiền về nếu không muốn sử dụng dịch vụ nữa.

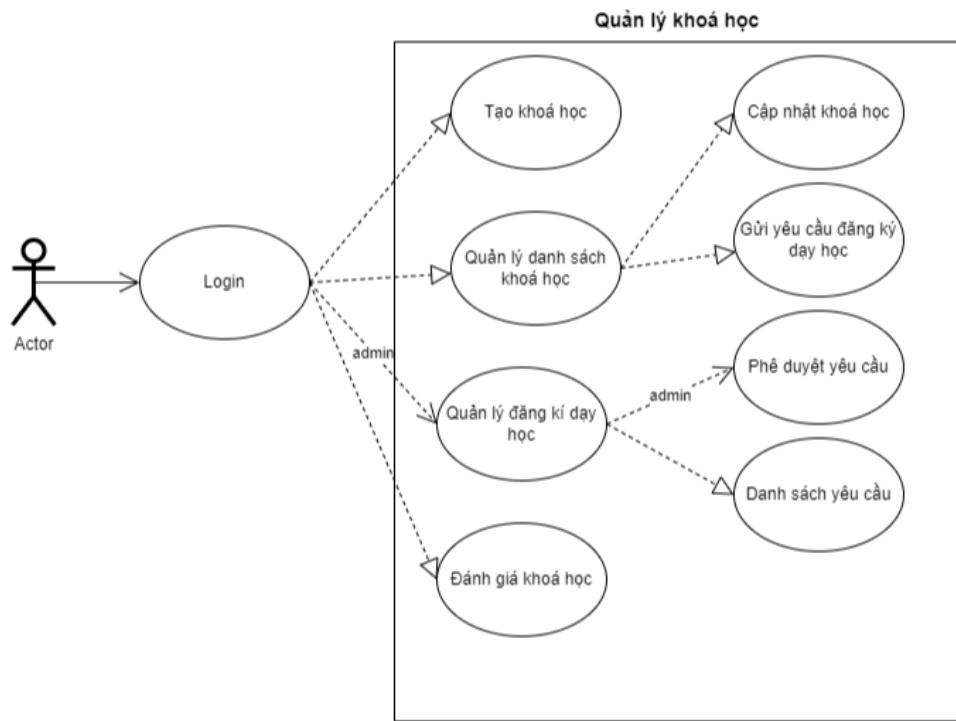
### 4. Sơ đồ usecase quản lý thông báo



*Hình 3.6. Hình ảnh usecase quản lý thông báo*

Đặc tả: Người dùng đăng nhập vào hệ thống có thể nhìn thấy các thông báo của mình chưa đọc và có thể xem thông báo, khi xem và click vào thông báo sẽ chuyển thành thông báo đã đọc. Người dùng có thể đánh dấu toàn bộ thông báo chưa đọc sang đã đọc.

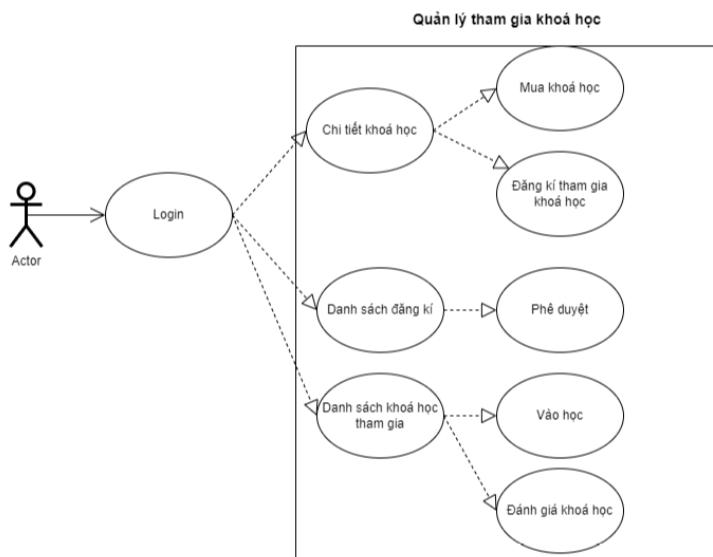
## 5. Sơ đồ usecase quản lý khoá học



Hình 3.7. Hình ảnh usecase quản lý khoá học

**Đặc tả:** Người dùng đăng nhập vào hệ thống, hệ thống phân quyền người dùng để thực hiện các chức năng khác nhau, người dùng có thể tạo khoá học, quản lý khoá học mình đã tạo, tạo yêu cầu cấp phép khoá học được giảng dạy, cập nhật thông tin khoá học, đánh giá khoá học mình tham gia. Admin phê duyệt yêu cầu cấp phép cho khoá học được giảng dạy của chủ khoá học.

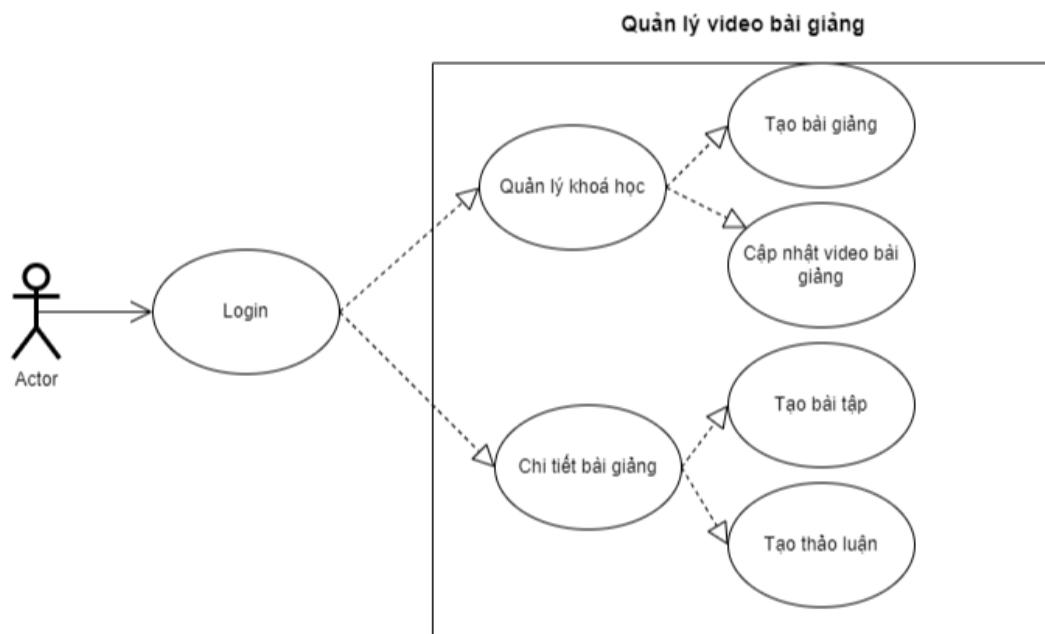
## 6. Sơ đồ usecase quản lý tham gia khoá học



Hình 3.8. Hình ảnh usecase quản lý tham gia Khoa học

Đặc tả: Người dùng đăng nhập vào hệ thống có thể xem chi tiết khoá học và thực hiện đăng ký tham gia học tập hoặc mua khoá học đối với những khoá học trả phí. Quản lý danh sách người dùng đăng ký học đối với những khoá học mà mình sở hữu và phê duyệt yêu cầu tham gia. Xem danh sách khoá học mình tham gia và vào học tập. Đánh giá chất lượng khoá học mình đã tham gia một lần duy nhất cho mỗi người dùng tham gia khoá học.

## 7. Sơ đồ usecase quản lý video bài giảng



Hình 3.9. Hình ảnh usecase quản lý video bài giảng

Đặc tả: Người dùng đăng nhập vào hệ thống có thể quản lý khoá học của chính mình, tạo bài giảng, cập nhật video bài giảng cho học viên học tập. Đi đến chi tiết bài giảng để tạo bài tập, tạo thảo luận trong bài giảng.

### 3.1.5. Thiết kế cơ sở dữ liệu

#### 1. Xây dựng đối tượng thực thể

Dựa trên sơ đồ chức năng, sơ đồ usecase, các yêu cầu chức năng đã đề ra, hệ thống xây dựng các đối tượng thực thể và liệt kê dưới bảng sau:

Bảng 3.1. Bảng thuộc tính các đối tượng trong hệ thống

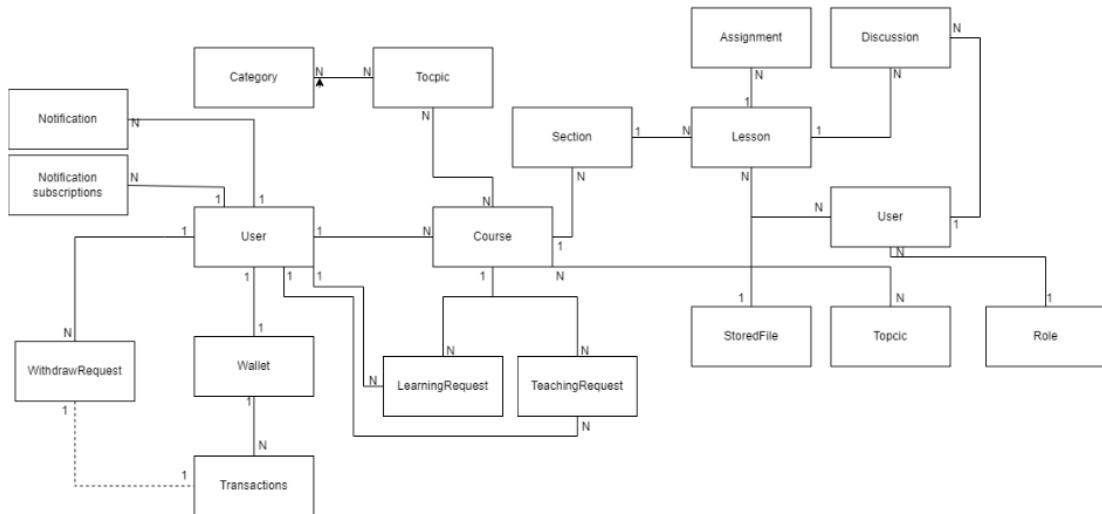
STT	Đối tượng	Thuộc tính	Mô tả																						
1	User	<table border="1"> <thead> <tr> <th colspan="2">users</th> </tr> </thead> <tbody> <tr> <td><code>id</code> </td><td>integer</td></tr> <tr> <td><code>name</code></td><td>varchar</td></tr> <tr> <td><code>email</code></td><td>varchar</td></tr> <tr> <td><code>facebook</code></td><td>varchar</td></tr> <tr> <td><code>github</code></td><td>varchar</td></tr> <tr> <td><code>twitter</code></td><td>varchar</td></tr> <tr> <td><code>avatar</code></td><td>varchar</td></tr> <tr> <td><code>description</code></td><td>varchar</td></tr> <tr> <td><code>created_at</code></td><td>timestamp</td></tr> <tr> <td><code>updated_at</code></td><td>timestamp</td></tr> </tbody> </table>	users		<code>id</code>	integer	<code>name</code>	varchar	<code>email</code>	varchar	<code>facebook</code>	varchar	<code>github</code>	varchar	<code>twitter</code>	varchar	<code>avatar</code>	varchar	<code>description</code>	varchar	<code>created_at</code>	timestamp	<code>updated_at</code>	timestamp	<ol style="list-style-type: none"> <li>1. Id: mã người dùng</li> <li>2. Name: Tên</li> <li>3. Email</li> <li>4. Facebook: link facebook</li> <li>5. Github: Link github</li> <li>6. Twiter: Link Twiter</li> <li>7. Avatar: Link ảnh đại diện</li> <li>8. Description: Mô tả</li> </ol>
users																									
<code>id</code>	integer																								
<code>name</code>	varchar																								
<code>email</code>	varchar																								
<code>facebook</code>	varchar																								
<code>github</code>	varchar																								
<code>twitter</code>	varchar																								
<code>avatar</code>	varchar																								
<code>description</code>	varchar																								
<code>created_at</code>	timestamp																								
<code>updated_at</code>	timestamp																								
2	Wallet	<table border="1"> <thead> <tr> <th colspan="2">wallet</th> </tr> </thead> <tbody> <tr> <td><code>id</code> </td><td>integer</td></tr> <tr> <td><code>balance</code></td><td>double</td></tr> <tr> <td><code>type</code></td><td>integer</td></tr> <tr> <td><code>created_at</code></td><td>timestamp</td></tr> <tr> <td><code>updated_at</code></td><td>timestamp</td></tr> </tbody> </table>	wallet		<code>id</code>	integer	<code>balance</code>	double	<code>type</code>	integer	<code>created_at</code>	timestamp	<code>updated_at</code>	timestamp	<ol style="list-style-type: none"> <li>1. Id: Mã của ví</li> <li>2. Balance: Số dư ví</li> <li>3. Type: Loại của ví</li> </ol>										
wallet																									
<code>id</code>	integer																								
<code>balance</code>	double																								
<code>type</code>	integer																								
<code>created_at</code>	timestamp																								
<code>updated_at</code>	timestamp																								
3	Category	<table border="1"> <thead> <tr> <th colspan="2">Category</th> </tr> </thead> <tbody> <tr> <td><code>id</code> </td><td>integer</td></tr> <tr> <td><code>name</code></td><td>varchar</td></tr> <tr> <td><code>slug</code></td><td>varchar</td></tr> <tr> <td><code>created_at</code></td><td>timestamp</td></tr> <tr> <td><code>updated_at</code></td><td>timestamp</td></tr> </tbody> </table>	Category		<code>id</code>	integer	<code>name</code>	varchar	<code>slug</code>	varchar	<code>created_at</code>	timestamp	<code>updated_at</code>	timestamp	<ol style="list-style-type: none"> <li>1. Id: Mã danh mục</li> <li>2. Name: Tên danh mục</li> <li>3. Slug: Tên không dấu</li> </ol>										
Category																									
<code>id</code>	integer																								
<code>name</code>	varchar																								
<code>slug</code>	varchar																								
<code>created_at</code>	timestamp																								
<code>updated_at</code>	timestamp																								
4	Topic	<table border="1"> <thead> <tr> <th colspan="2">Topic</th> </tr> </thead> <tbody> <tr> <td><code>id</code> </td><td>integer</td></tr> <tr> <td><code>name</code></td><td>varchar</td></tr> <tr> <td><code>slug</code></td><td>varchar</td></tr> <tr> <td><code>created_at</code></td><td>timestamp</td></tr> <tr> <td><code>updated_at</code></td><td>timestamp</td></tr> </tbody> </table>	Topic		<code>id</code>	integer	<code>name</code>	varchar	<code>slug</code>	varchar	<code>created_at</code>	timestamp	<code>updated_at</code>	timestamp	<ol style="list-style-type: none"> <li>1. Id: Mã chủ đề</li> <li>2. Name: Tên chủ đề</li> <li>3. Slug: Tên không dấu</li> </ol>										
Topic																									
<code>id</code>	integer																								
<code>name</code>	varchar																								
<code>slug</code>	varchar																								
<code>created_at</code>	timestamp																								
<code>updated_at</code>	timestamp																								

5	Course	<b>Courses</b> id ⚡ integer name varchar slug varchar cover varchar description text mode integer status integer type integer updated_at timestamp created_at timestamp	1. Id: Mã khoá học 2. Name: Tên khoá học 3. Slug: Tên không dấu 4. Cover: Ảnh bìa 5. Description: Mô tả khoá học 6. Status: Trạng thái khoá học 7. Mode: Hình thức khoá học 8. Type: Loại khoá học
6	Section	<b>Section</b> id ⚡ integer title varchar description varchar order integer created_at timestamp updated_at timestamp	1. Id: Mã của chương học 2. Title: Tiêu đề chương 3. Description: Mô tả chương 4. Order: Thứ tự chương trong khoá học
7	Lesson	<b>Lesson</b> id ⚡ integer title varchar description varchar video varchar time integer order integer type integer created_at timestamp updated_at timestamp	1. Id: Mã bài giảng 2. Title: Tiêu đề bài giảng 3. Description: Mô tả bài giảng 4. Video: Video của bài giảng 5. Time: Thời lượng bài giảng 6. Order: Thứ tự bài giảng 7. Type: Loại bài giảng
8	Assignment	<b>Assignment</b> id ⚡ uuid title varchar description varchar created_at timestamp updated_at timestamp	1. Id: Mã bài tập 2. Title: Tiêu đề bài tập 3. Description: Mô tả chi tiết
9	Discussion	<b>Discussion</b> id ⚡ uuid description varchar created_at timestamp updated_at timestamp	1. Id: Mã thảo luận 2. Description: Nội dung thảo luận

		<table border="1"> <thead> <tr> <th colspan="2">Notification</th> </tr> </thead> <tbody> <tr><td><code>id</code></td><td>integer</td></tr> <tr><td><code>payload</code></td><td>json</td></tr> <tr><td><code>fcm_message</code></td><td>json</td></tr> <tr><td><code>read</code></td><td>integer</td></tr> <tr><td><code>status</code></td><td>interger</td></tr> <tr><td><code>type</code></td><td>interger</td></tr> <tr><td><code>updated_at</code></td><td>timestamp</td></tr> <tr><td><code>created_at</code></td><td>timestamp</td></tr> </tbody> </table>	Notification		<code>id</code>	integer	<code>payload</code>	json	<code>fcm_message</code>	json	<code>read</code>	integer	<code>status</code>	interger	<code>type</code>	interger	<code>updated_at</code>	timestamp	<code>created_at</code>	timestamp	<ol style="list-style-type: none"> <li>1. Id: Mã thông báo</li> <li>2. Payload: Dữ liệu từ thông báo</li> <li>3. Fcm_message: Tin nhắn thông báo</li> <li>4. Read: Chưa đọc/ Đã đọc</li> <li>5. Status: Trạng thái</li> <li>6. Type: Loại thông báo</li> </ol>		
Notification																							
<code>id</code>	integer																						
<code>payload</code>	json																						
<code>fcm_message</code>	json																						
<code>read</code>	integer																						
<code>status</code>	interger																						
<code>type</code>	interger																						
<code>updated_at</code>	timestamp																						
<code>created_at</code>	timestamp																						
10	Notification	<table border="1"> <thead> <tr> <th colspan="2">Notification_Subscriptions</th> </tr> </thead> <tbody> <tr><td><code>id</code></td><td>integer</td></tr> <tr><td><code>status</code></td><td>interger</td></tr> <tr><td><code>token</code></td><td>varchar</td></tr> <tr><td><code>last_active</code></td><td>timestamp</td></tr> <tr><td><code>updated_at</code></td><td>timestamp</td></tr> <tr><td><code>created_at</code></td><td>timestamp</td></tr> </tbody> </table>	Notification_Subscriptions		<code>id</code>	integer	<code>status</code>	interger	<code>token</code>	varchar	<code>last_active</code>	timestamp	<code>updated_at</code>	timestamp	<code>created_at</code>	timestamp	<ol style="list-style-type: none"> <li>1. Id: Mã đăng ký</li> <li>2. Status: Trạng thái</li> <li>3. Token: Mã đăng ký nhận thông báo về máy tính</li> <li>4. Last_active: Lần cuối hoạt động</li> </ol>						
Notification_Subscriptions																							
<code>id</code>	integer																						
<code>status</code>	interger																						
<code>token</code>	varchar																						
<code>last_active</code>	timestamp																						
<code>updated_at</code>	timestamp																						
<code>created_at</code>	timestamp																						
11	Notification Subscription	<table border="1"> <thead> <tr> <th colspan="2">Transaction</th> </tr> </thead> <tbody> <tr><td><code>id</code></td><td>uuid</td></tr> <tr><td><code>amount</code></td><td>double</td></tr> <tr><td><code>message</code></td><td>json</td></tr> <tr><td><code>status</code></td><td>integer</td></tr> <tr><td><code>type</code></td><td>interger</td></tr> <tr><td><code>from_wallet</code></td><td>int</td></tr> <tr><td><code>to_wallet</code></td><td>int</td></tr> <tr><td><code>updated_at</code></td><td>timestamp</td></tr> <tr><td><code>created_at</code></td><td>timestamp</td></tr> </tbody> </table>	Transaction		<code>id</code>	uuid	<code>amount</code>	double	<code>message</code>	json	<code>status</code>	integer	<code>type</code>	interger	<code>from_wallet</code>	int	<code>to_wallet</code>	int	<code>updated_at</code>	timestamp	<code>created_at</code>	timestamp	<ol style="list-style-type: none"> <li>1. Id: Mã giao dịch</li> <li>2. Amount: Tổng tiền giao dịch</li> <li>3. Message: Nội dung</li> <li>4. Status: Trạng thái</li> <li>5. Type: Loại giao dịch</li> <li>6. From_wallet: Ví người gửi</li> <li>7. To_Wallet: Ví người nhận</li> </ol>
Transaction																							
<code>id</code>	uuid																						
<code>amount</code>	double																						
<code>message</code>	json																						
<code>status</code>	integer																						
<code>type</code>	interger																						
<code>from_wallet</code>	int																						
<code>to_wallet</code>	int																						
<code>updated_at</code>	timestamp																						
<code>created_at</code>	timestamp																						
12	Transaction	<table border="1"> <thead> <tr> <th colspan="2">Withdraw_requests</th> </tr> </thead> <tbody> <tr><td><code>id</code></td><td>uuid</td></tr> <tr><td><code>amount</code></td><td>varchar</td></tr> <tr><td><code>status</code></td><td>varchar</td></tr> <tr><td><code>requester</code></td><td>varchar</td></tr> <tr><td><code>accepter</code></td><td>text</td></tr> <tr><td><code>transaction</code></td><td>uuid</td></tr> <tr><td><code>created_at</code></td><td>timestamp</td></tr> <tr><td><code>updated_at</code></td><td>timestamp</td></tr> </tbody> </table>	Withdraw_requests		<code>id</code>	uuid	<code>amount</code>	varchar	<code>status</code>	varchar	<code>requester</code>	varchar	<code>accepter</code>	text	<code>transaction</code>	uuid	<code>created_at</code>	timestamp	<code>updated_at</code>	timestamp	<ol style="list-style-type: none"> <li>1. Id: Mã yêu cầu rút tiền</li> <li>2. Amount: Số tiền yêu cầu</li> <li>3. Status: Trạng thái</li> <li>4. Requester: người yêu cầu</li> <li>5. Acceptor: người chấp thuận</li> <li>6. Transaction: Mã giao dịch</li> </ol>		
Withdraw_requests																							
<code>id</code>	uuid																						
<code>amount</code>	varchar																						
<code>status</code>	varchar																						
<code>requester</code>	varchar																						
<code>accepter</code>	text																						
<code>transaction</code>	uuid																						
<code>created_at</code>	timestamp																						
<code>updated_at</code>	timestamp																						
13	WithdrawRequest																						

14	Storedfile	<b>Stored_file</b> id ⚡ integer name varchar hash varchar key varchar <b>path</b> varchar created_at timestamp updated_at timestamp	1. Id: Mã của file 2. Name: Tên file 3. Hash: Tên mã hoá file 4. Key: Khoá file lưu 5. Path: Đường dẫn file
15	LearningRequest	<b>LearningRequest</b> id ⚡ uuid course integer requester integer status integer	1. Id: Mã yêu cầu tham gia học 2. Course: Khoá học yêu cầu 3. Requester: Người yêu cầu 4. Status: Trạng thái
16	TeachingRequest	<b>TeachingRequest</b> id ⚡ uuid course integer requester integer status integer	1. Id: Mã yêu cầu tham gia học 2. Course: Khoá học yêu cầu 3. Requester: Người yêu cầu 4. Status: Trạng thái
17	Roles	<b>Role</b> id ⚡ integer <b>name</b> varchar type integer created_at timestamp updated_at timestamp	1. Id: Mã của quyền 2. Name: Tên quyền 3. Type: Loại quyền

## 9. Xây dựng sơ đồ quan hệ



Hình 3.10. Hình ảnh sơ đồ quan hệ dữ liệu của hệ thống

Dựa trên các yêu cầu về chức năng, xác định đối tượng, hệ thống xây dựng sơ đồ quan hệ giữa các đối tượng theo quan hệ các cặp quan hệ 1-1, 1-N, N-N.

Các cặp dữ liệu quan hệ 1 – 1 trong hệ thống được liệt kê sau đây:

Bảng 3.2. Bảng các cặp dữ liệu quan hệ 1-1

STT	Cặp khoá	Mô tả
1	User - Wallet	Một người dùng có một ví duy nhất
2	WithdrawRequest – Transactions	Một yêu cầu rút tiền được chấp nhận sẽ có một giao dịch chuyển tiền duy nhất.

Các cặp dữ liệu quan hệ N - N trong hệ thống được liệt kê sau đây:

Bảng 3.3. Bảng các cặp dữ liệu quan hệ N-N

STT	Cặp khoá	Mô tả
1	Category – Topic	Một danh mục có nhiều chủ đề và một chủ đề có nhiều danh mục
2	Topic – Course	Một chủ đề có nhiều khóa học và một khóa học có nhiều chủ đề

Các cặp dữ liệu quan hệ 1 – N trong hệ thống được liệt kê sau đây:

Bảng 3.4. Bảng các cặp dữ liệu quan hệ 1-N

STT	Cặp khoá	Mô tả
1	User-Course	Một người dùng có thể sở hữu nhiều khóa học và một khóa học chỉ có 1 chủ sở hữu
2	User-Withdraw Request	Một người dùng có thể tạo nhiều yêu cầu rút tiền. Một yêu cầu chỉ được xác định bởi một người
3	User-Notification	Một người có thể nhận nhiều thông báo, một thông báo chỉ định đến một người

4	User-Notification Subscriptions	Một người dùng dùng có nhiều token thông báo đăng ký, một token chỉ dành cho một người.
5	User-Learning Request	Một người dùng sẽ tạo được nhiều yêu cầu tham gia khoá học, một yêu cầu chỉ được xác định bởi một người tạo nó.
6	Course-Learning Request	Một khoá học có thể có nhiều yêu cầu tham gia học, một yêu cầu chỉ được xác định bởi một khoá học duy nhất
7	User-Teaching Request	Một người có thể tạo nhiều yêu cầu dạy học, một yêu cầu chỉ được xác định bởi một người
8	Course - Teaching Request	Một khoá học có nhiều yêu cầu giảng dạy, một yêu cầu giảng dạy chỉ dành cho một khoá học
9	Wallet – Transactions	Một ví có thể có nhiều giao dịch, một giao dịch chỉ xác định được một ví nhận và một ví gửi
10	Course-Section	Một khoá học có nhiều chương, một chương chỉ thuộc về một khoá học
11	Section-Lesson	Một chương có nhiều bài giảng, một bài giảng thuộc về một chương
12	Lesson-Assignment	Một bài giảng có nhiều bài tập, một bài tập chỉ thuộc về một bài giảng
13	Lesson-Discussion	Một bài giảng có nhiều thảo luận, một thảo luận thuộc về một bài giảng
14	User-Discussion	Một người dùng tạo nhiều thảo luận, một thảo luận chỉ thuộc về một người

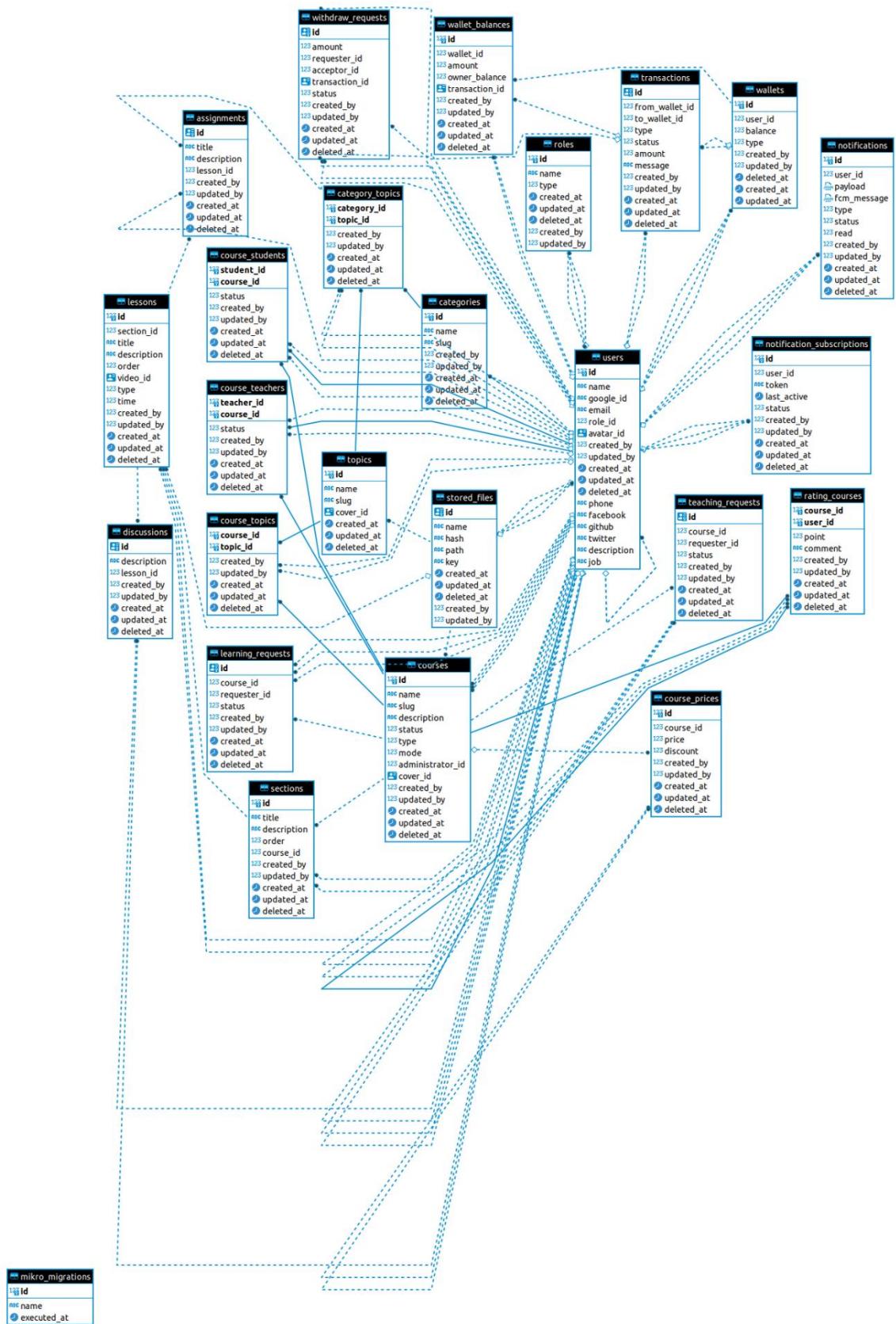
## 10. Xây dựng database

Dựa trên các chức năng, đối tượng thực thể và quan hệ giữa các thực thể, hệ thống xây dựng cơ sở dữ liệu của mình với các bảng như sau

Bảng 3.5. Bảng danh sách các bảng trong cơ sở dữ liệu

STT	Tên bảng	Ý nghĩa
1	Roles	Phân quyền dành cho người dùng
2	Users	Thông tin người dùng
3	Wallet	Ví tiền của người dùng
4	Categories	Các danh mục cho chủ đề khoá học
5	Topics	Các chủ đề khoá học
6	Courses	Các khoá học của người dùng
7	Sections	Các chương của mỗi khoá học
8	Lessons	Các bài giảng của mỗi chương trong khoá học
9	Assignments	Bài tập cho mỗi bài giảng
10	Dicussions	Các thảo luận trong bài giảng
11	Notifications	Các thông báo gửi đến từng người dùng
12	Learning_requests	Các yêu cầu tham gia khoá học của người dùng
13	Teaching_requests	Các yêu cầu cấp phép khoá học của chủ khoá học
14	Course_teacher	Bảng giáo viên của khoá học
15	Course_students	Bảng học viên của khoá học

16	Category_topics	Bảng các danh mục và chủ đề liên quan đến nhau
17	Course_topics	Bảng các khoá học và chủ đề liên quan đến nhau
18	Rating_courses	Các đánh giá của học viên tham gia khoá học
19	Course_prices	Bảng thông tin giá khoá học
20	Transactions	Bảng thông tin giao dịch liên quan đến tiền của người dùng
21	Withdraw_requests	Bảng liên quan đến các yêu cầu rút tiền của người dùng
22	Wallet_balance	Bảng số dư của ví tại thời điểm giao dịch
23	Stored_files	Bảng liên quan đến các fileUpload, image, video, ...của người dùng đối với từng đối tượng cụ thể.
24	Notification_subscriptions	Đăng ký nhận thông báo FCM message từ google



Hình 3.11. Hình ảnh cơ sở dữ liệu của hệ thống

Hình ảnh 3.11 trên là hình ảnh cơ sở dữ liệu đầy đủ được sử dụng trong hệ thống sau khi thiết lập đối tượng, xác định các chủ đề và mối quan hệ giữa các đối tượng từ sơ đồ quan hệ để xây dựng lên một cơ sở dữ liệu hoàn chỉnh

### 3.2. Thiết kế phần mềm.

#### 3.2.1. Thiết lập môi trường, cấu trúc dự án

Để xây dựng được hệ thống học lập trình trực tuyến máy tính cần phải cài đặt môi trường phát triển và thiết lập một số các cài đặt công cụ để lập trình. Hệ thống được phát triển bằng NestJs và NextJs nên máy tính cần phải cài đặt môi trường NodeJs.

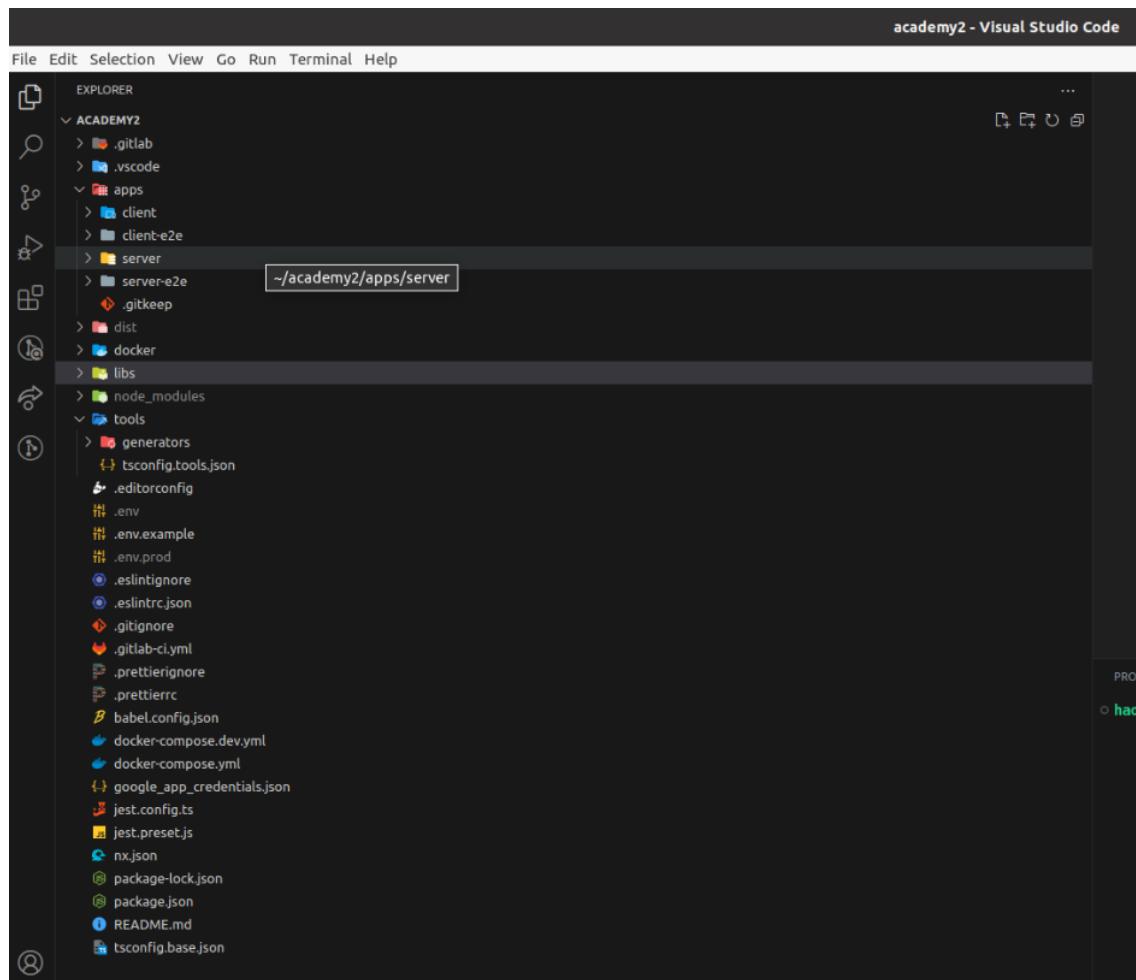
Hệ thống sử dụng cơ sở dữ liệu Postgresql, sử dụng redis, vv. Để thuận tiện cho việc phát triển và dễ dàng triển khai, không mất công cài đặt nhiều phần mềm xung quanh việc phát triển hệ thống, Docker sẽ được lựa chọn để tiết kiệm thời gian và dễ dàng sử dụng.

Đối với công cụ lập trình, Visual studio code là một IDE rất phổ biến và thuận tiện, hỗ trợ mạnh mẽ đối với ngôn ngữ Javascript và Typescript.

Hệ thống quản lý mã nguồn bằng gitlab, do đó ở máy tính phát triển sẽ cài đặt git để đẩy code lên gitlab.

1. Cài đặt NodeJs: Cài đặt NodeJs về máy tính dựa trên hướng dẫn cài đặt từ chính trang chủ của NodeJs <https://nodejs.org/en/download> để cài đặt phiên bản cho phù hợp với máy tính.
2. Cài đặt Docker: Cài đặt về máy tính của mình từ chính trang chủ của Docker <https://docs.docker.com/get-docker/>
3. Cài đặt Visual studio code: Tải về máy của mình trên chính trang chủ <https://code.visualstudio.com/> và làm theo hướng dẫn.
4. Cài đặt Git: Cài đặt git về máy của mình từ trang chủ <https://git-scm.com/downloads> và tạo tài khoản gitlab (nếu chưa có) để tạo repository trên <https://gitlab.com/>

Sau khi cài đặt xong, bước tiếp theo là tiến hành tạo project và setup cho nó. Đối với hệ thống học lập trình sử dụng “Nx monorepo” để quản lý cấu trúc của dự án. Thực hiện cài đặt “nx monorepo” theo hướng dẫn từ <https://nx.dev/getting-started/installation>. Dưới đây là cấu trúc của dự án



Hình 3.12. Hình ảnh cấu trúc dự án

Cấu trúc dự án được chia thành các phần chính:

- Apps: Nơi chứa code chính của dự án bao gồm frontend và backend
- Dist: Nơi chứa thư mục build sau khi build project để đưa lên sử dụng
- Docker: Nơi chứa các Dockerfile của client và server
- Libs: Nơi lưu trữ code dùng chung giữa frontend và backend
- Node\_modules: Nơi chứa mã nguồn các thư viện cài đặt trong dự án
- Package.json: file cấu hình của npm, giúp npm cần cài đặt và thực thi lệnh, chứa các thông tin ứng dụng, phiên bản.

### 3.2.2. Xây dựng server

Backend của hệ thống sử dụng ngôn ngữ lập trình Javascript và framework NestJs của NodeJS để lập trình.

1. Server xây dựng dưới dạng WebAPI cung cấp các URL API để bên thứ 3 có thể gửi request dữ liệu đến máy chủ thông qua giao thức HTTP hoặc HTTPS.
2. Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.

3. Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.
4. Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động , dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu, vv.

Dưới đây là kết quả sau sau khi thực hiện xây dựng server

The screenshot displays the Swagger UI for the 'Academy Zero3' API. The API is organized into several categories:

- app**: Contains a single endpoint: `GET /api`.
- users**: Contains four endpoints: `GET /api/users`, `POST /api/users` (highlighted in green), `GET /api/users/public`, and `GET /api/users/teacher`.
- authGoogle**: Contains a single endpoint: `POST /api/auth/google`.
- auth**: Contains a single endpoint: `GET /api/auth/me`.
- categories**: Contains a single endpoint: `GET /api/categories`.
- topics**: Contains two endpoints: `GET /api/topics` and `GET /api/topics/{slug}`.
- COURSES**: Contains seven endpoints: `GET /api/courses`, `POST /api/courses` (highlighted in green), `GET /api/courses/manage`, `GET /api/courses/stats`, `GET /api/courses/learnings`, `GET /api/courses/{slug}`, `PUT /api/courses/{id}` (highlighted in orange), and `DELETE /api/courses/{id}` (highlighted in red).
- sections**: Contains two endpoints: `POST /api/sections` and `DELETE /api/sections/{id}`.

Hình 3.13. Hình ảnh các url api của server

GET	/api/discussions
POST	/api/discussions
assignments	
DELETE	/api/discussions/{id}
GET	/api/assignments
POST	/api/assignments
DELETE	/api/assignments/{id}
teaching-requests	
GET	/api/teaching-requests
POST	/api/teaching-requests
GET	/api/teaching-requests/{id}
PUT	/api/teaching-requests/accept/{id}
PUT	/api/teaching-requests/reject/{id}
learning-requests	
GET	/api/learning-requests
POST	/api/learning-requests
GET	/api/learning-requests/{id}
PUT	/api/learning-requests/accept/{id}
PUT	/api/learning-requests/reject/{id}
notification-subscriptions	
POST	/api/notification-subscriptions
notifications	
GET	/api/notifications
POST	/api/notifications/mark-as-read
POST	/api/notifications/{id}/mark-as-read
rating-courses	
GET	/api/rating-courses
POST	/api/rating-courses
GET	/api/rating-courses/{id}/point
Schemas	
ValidationExceptionErrorResponse >	

Hình 3.14. Hình ảnh các url api của server (p2)

Hai hình trên là hình ảnh các api từ server cung cấp cho các bên thứ ba gọi đến truy xâu vào hệ thống lấy dữ liệu.



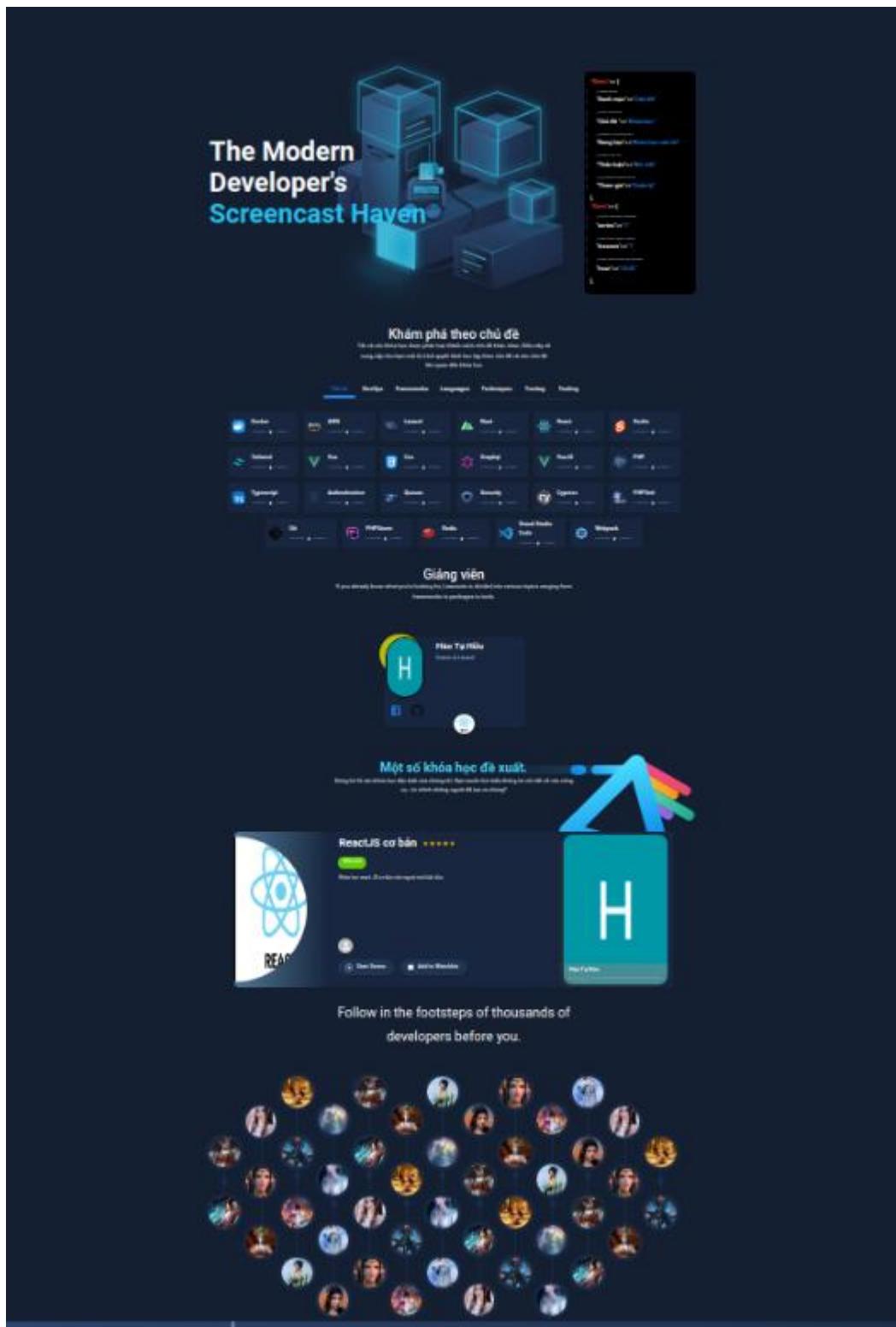
Hình 3.15. Hình ảnh các dto, response data của server

Hình 3.15 là hình ảnh các Dto dùng để kiểm tra đầu vào dữ liệu từ bên thứ ba gửi đến, các responsive là các kiểu thông tin được trả về khi lấy dữ liệu thành công.

### 3.2.3. Xây dựng giao diện người dùng

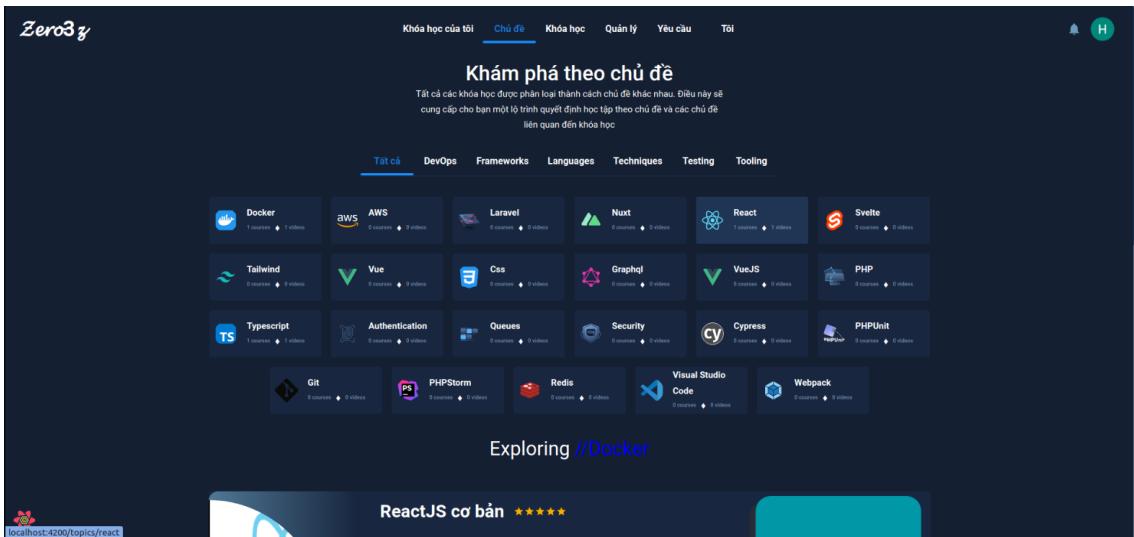
Hệ thống xây dựng giao diện người dùng sử dụng ngôn ngữ Javascript và dùng framework NextJS để xây dựng website frontend. Website được kết nối với server thông qua các API do server cung cấp.

Hệ thống xây dựng website với các trang giao diện thân thiện với người dùng, giúp người dùng sử dụng website thực hiện các chức năng để tương tác với server một cách dễ dàng hơn. Giúp người dùng có trải nghiệm mượt mà, dễ dàng và tiện lợi, đáp ứng các chức năng hệ thống đưa ra. Dưới đây là giao diện người dùng đã được xây dựng.



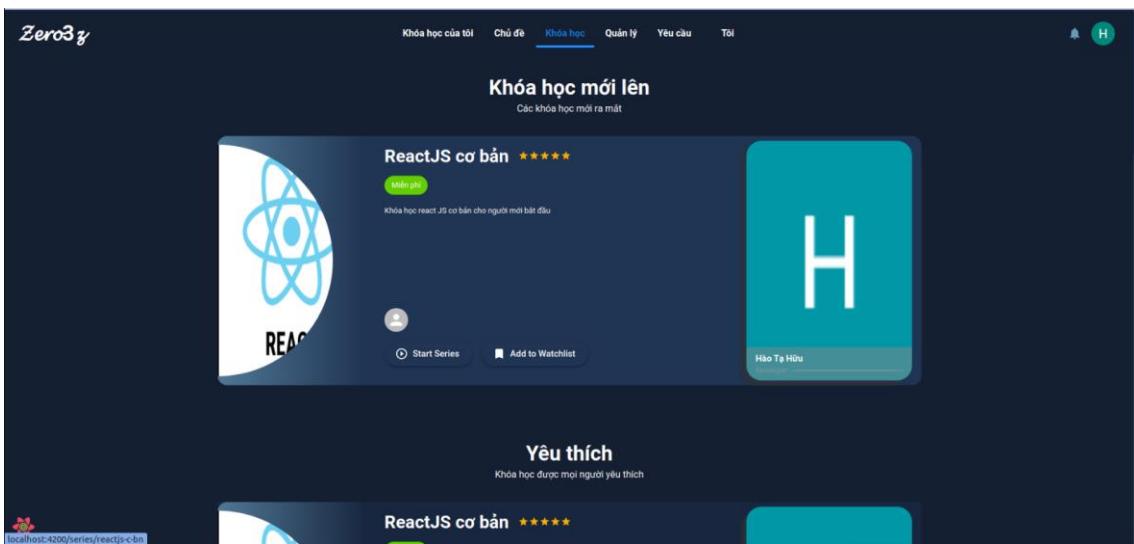
Hình 3.16. Hình ảnh giao diện trang chủ của hệ thống

Hình 3.16 hiển thị màn hình trang chủ khi truy cập vào website, người dùng sẽ đứng ở trang chủ và có thể xem các danh mục, chủ đề, các khoá học, giảng viên, hình ảnh các học viên của hệ thống. Người dùng có thể đi đến chi tiết các chủ đề, danh mục, danh sách các khoá học thông qua menu.



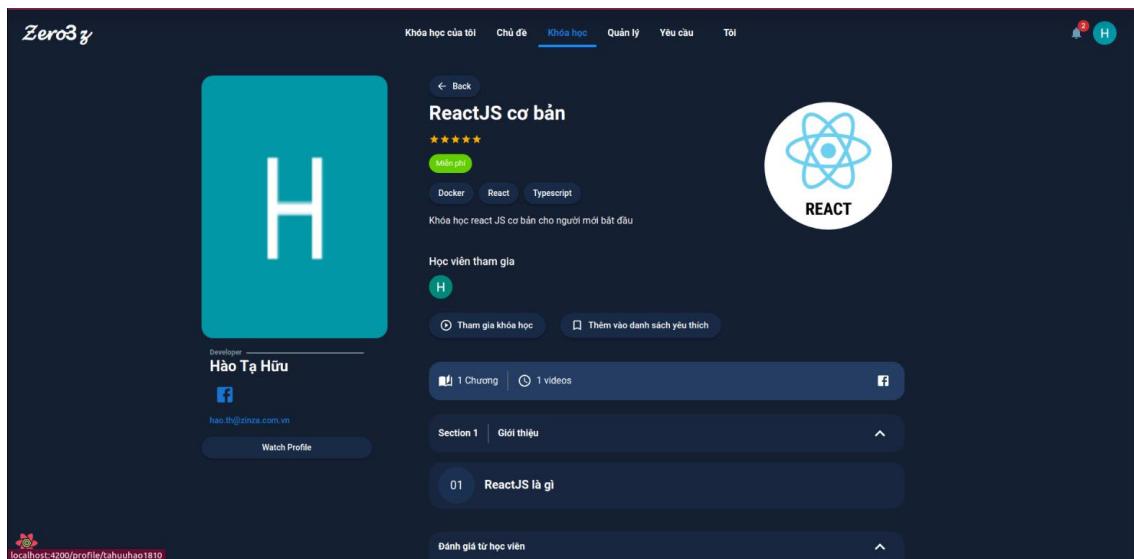
Hình 3.17. Hình ảnh giao diện danh sách chủ đề theo danh mục

Hình 3.17 hiển thị danh sách các danh mục, chủ đề, thông tin số lượng các khóa học có trong chủ đề, người dùng chọn bất kì chủ đề nào sẽ xuất hiện các khóa học của chủ đề đó bên dưới.



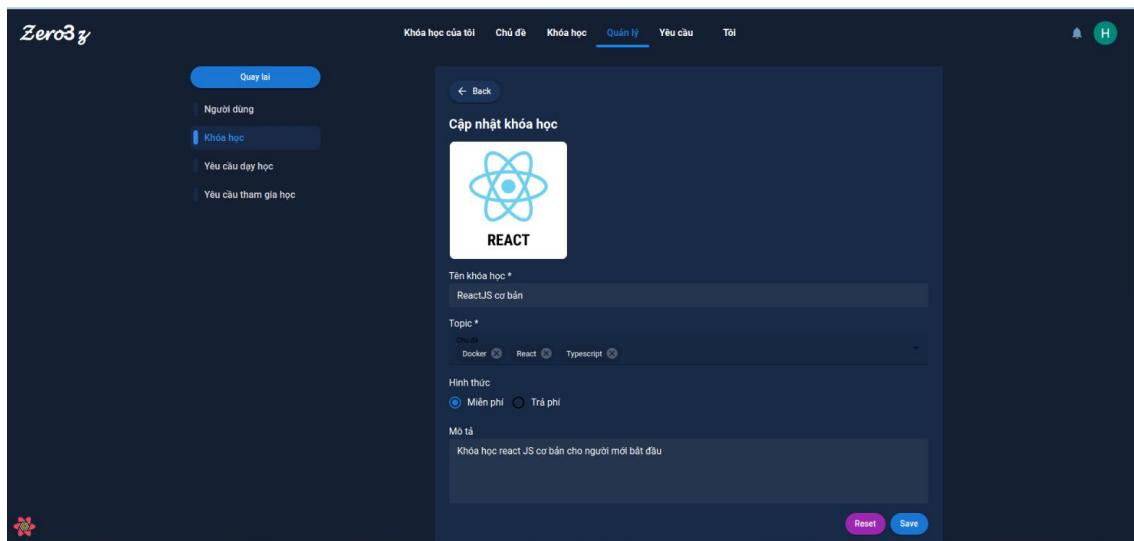
Hình 3.18. Hình ảnh giao diện danh sách khoá học

Hình 3.18 hiển thị trang danh sách khoá học, giao diện hiển thị danh sách các khoá học đã được cấp phép bởi admin đi vào hoạt động. Có thể chọn bất kì khoá học nào để đi đến chi tiết khoá học đó.



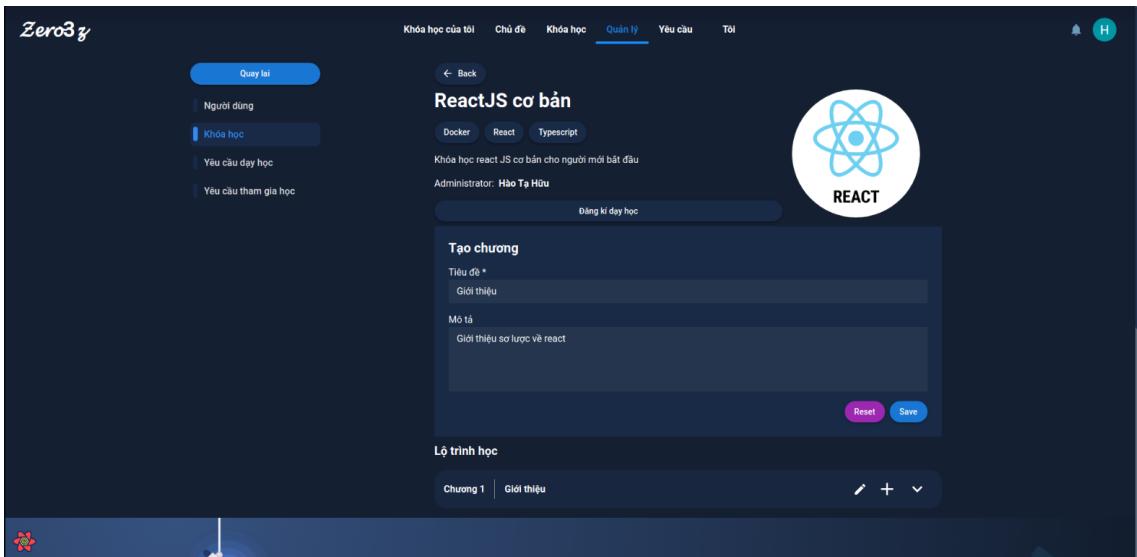
Hình 3.19. Hình ảnh giao diện chi tiết khóa học

Hình 3.19 hiển thị giao diện chi tiết khóa học, giao diện hiển thị thông tin chi tiết của khóa học, số lượng bài giảng có trong khóa học, nội dung tiêu đề các bài giảng và có thể tham gia khóa học.



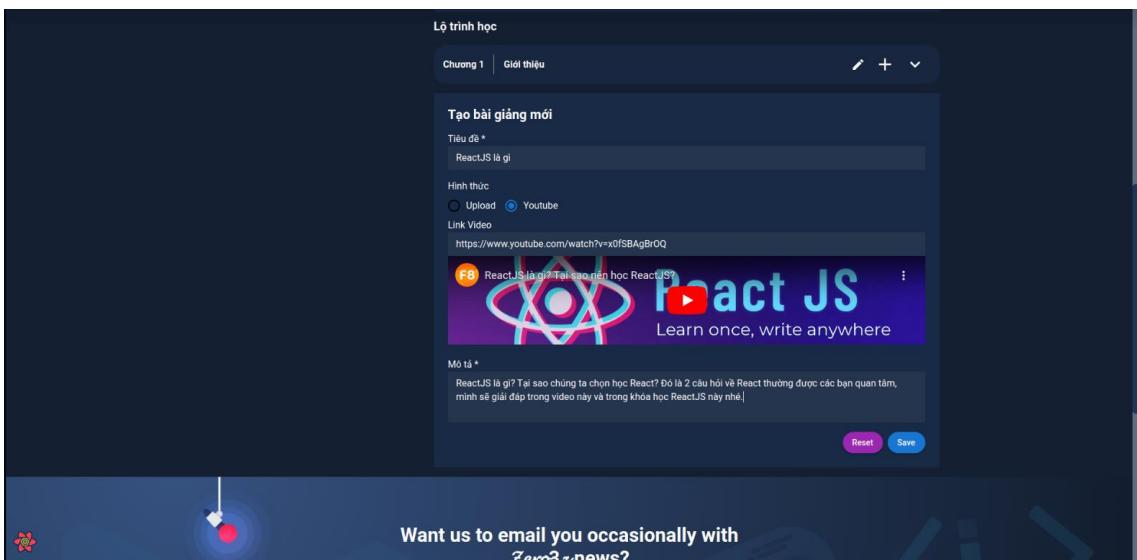
Hình 3.20. Hình ảnh giao diện tạo mới khóa học

Hình 3.20 hiển thị form tạo mới khóa học, giao diện hiển thị form tạo khóa học. Tại đây có thể tải ảnh từ máy của mình hoặc kéo thả từ trên mạng về để làm ảnh bìa cho khóa học, lựa chọn các chủ đề phù hợp với khóa học, chọn hình thức miễn phí hoặc khóa học có phí, nhập mô tả cho khóa học. Nếu khóa học không đầy đủ thông tin sẽ không cho upload.



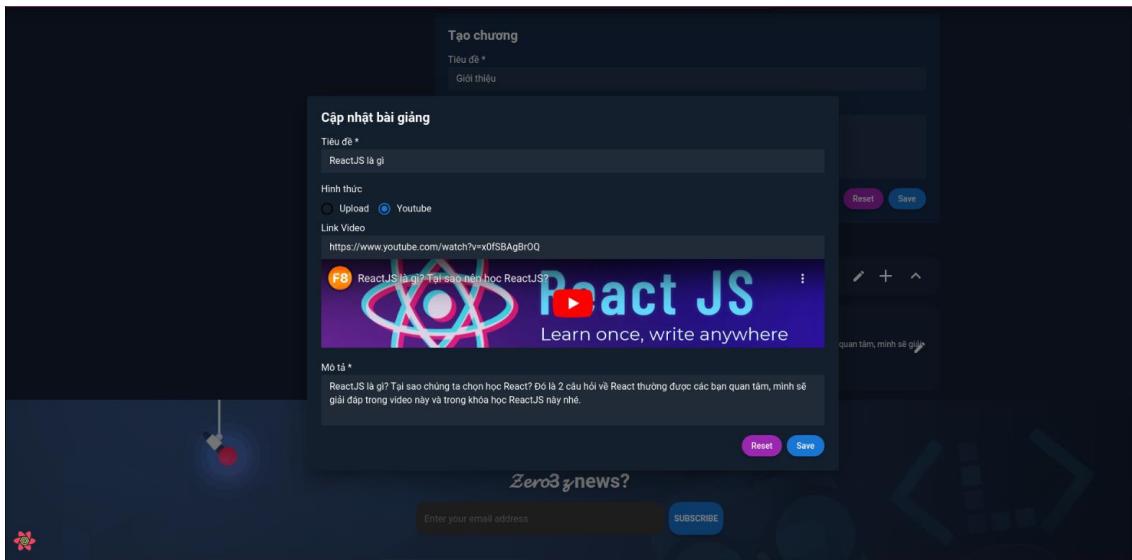
Hình 3.21. Hình ảnh giao diện quản lý chi tiết khóa học

Hình 3.21 hiển thị màn quản lý chi tiết khóa học, giao diện hiển thị thông tin khóa học và có thể thêm các chương, bài giảng cho khóa học. Đăng kí cấp phép cho khóa học được sử dụng và chờ phê duyệt yêu cầu từ admin.



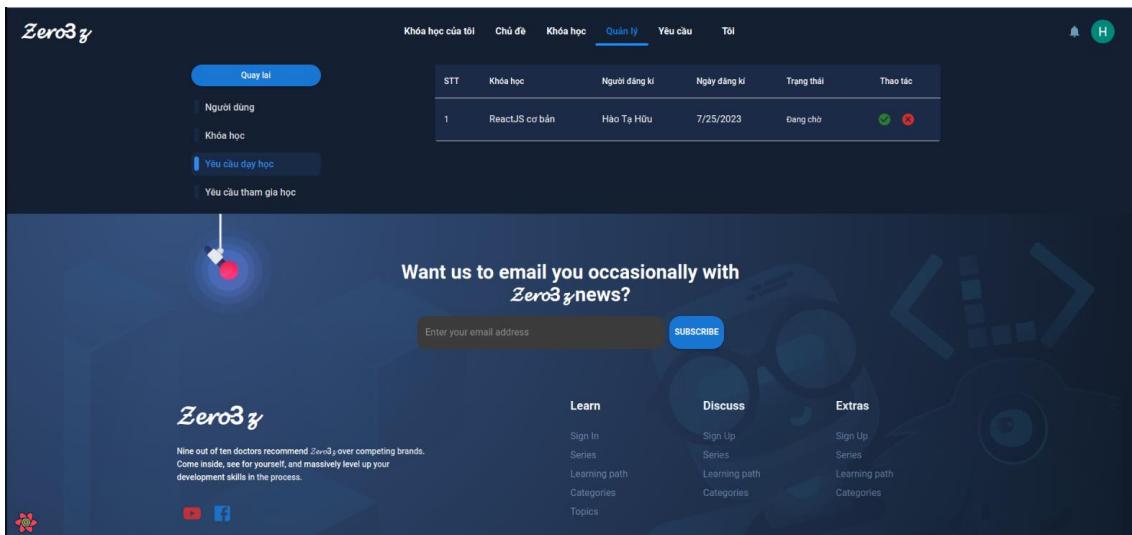
Hình 3.22. Hình ảnh giao diện tạo video bài giảng khóa học

Hình 3.22 hiển thị form tạo video bài giảng, người dùng nhập tên bài học, chọn hình thức tải video từ máy của mình hoặc sử dụng video youtube, nhập mô tả cho video. Đây đủ thông tin hệ thống sẽ cho phép upload video.



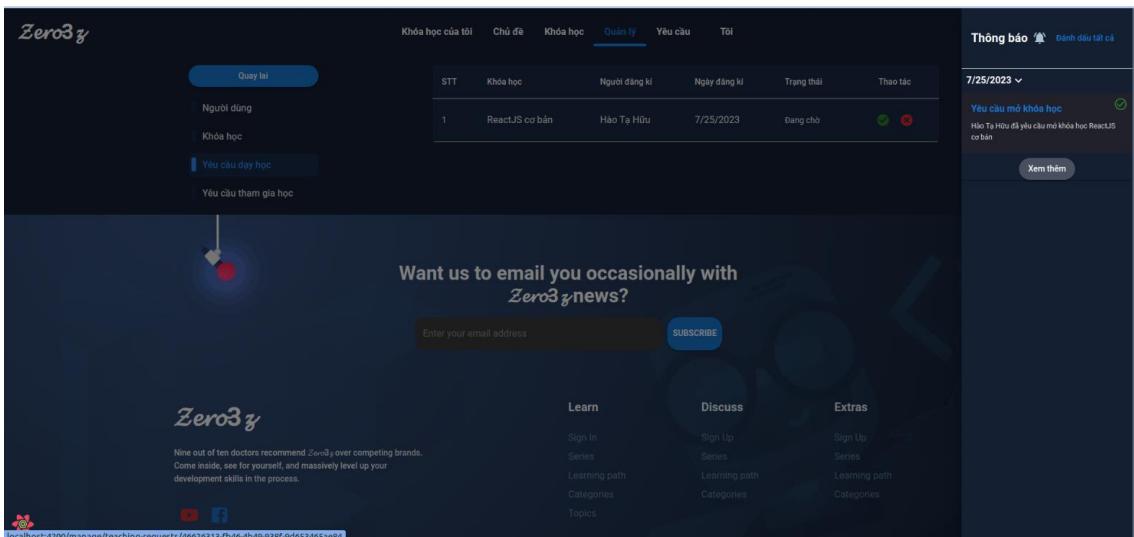
Hình 3.23. Hình ảnh giao diện cập nhật video bài giảng khoá học

Hình 3.23 hiển thị form cập nhật thông tin của video bài giảng đã tạo trước đó, người dùng có thể thay đổi nếu cần có bất kì sự thay đổi nào ví dụ như thay đổi tên, nội dung video hay mô tả video.



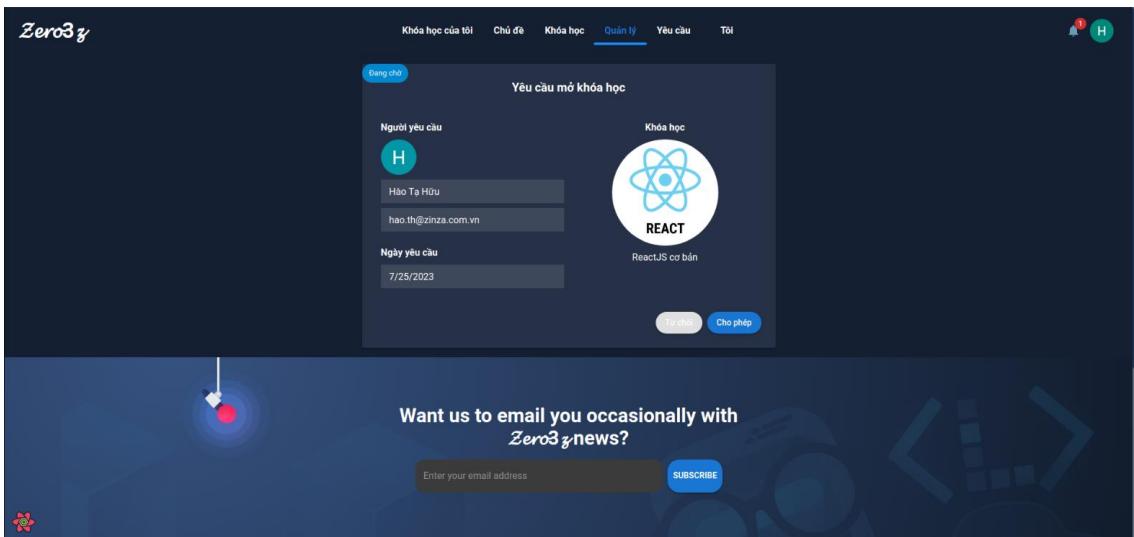
Hình 3.24. Hình ảnh giao diện danh sách yêu cầu cấp phép khoá học

Hình 3.24 hiển thị danh sách yêu cầu cấp phép khoá học hiển thị các yêu cầu đến từ chủ khoá học yêu cầu cấp phép cho khoá học của mình được sử dụng. Tại màn này chỉ admin được quyền thực hiện xác nhận cấp phép hay không.



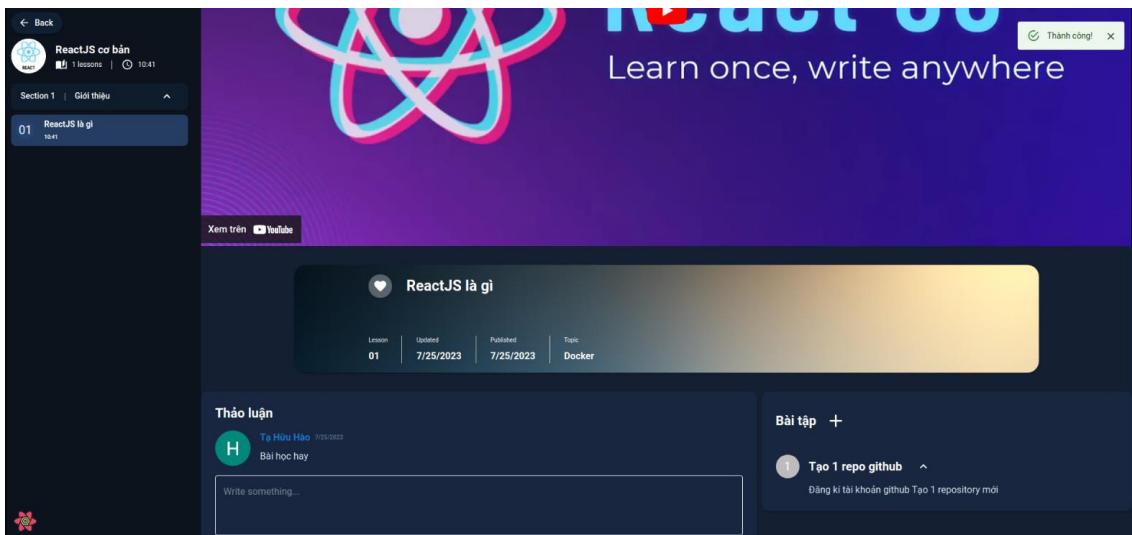
Hình 3.25. Hình ảnh giao diện danh sách thông báo

Hình 3.25 là giao diện hiển thị các thông báo chưa đọc của người dùng, khi click vào tích xanh đánh dấu đọc thì thông báo sẽ mất đi, khi chọn vào thông báo sẽ chuyển người dùng đến nội dung của thông báo. Người dùng có thể chọn nhanh bằng cách đánh dấu tất cả các thông báo nếu người dùng đã biết hết thông tin của thông báo.



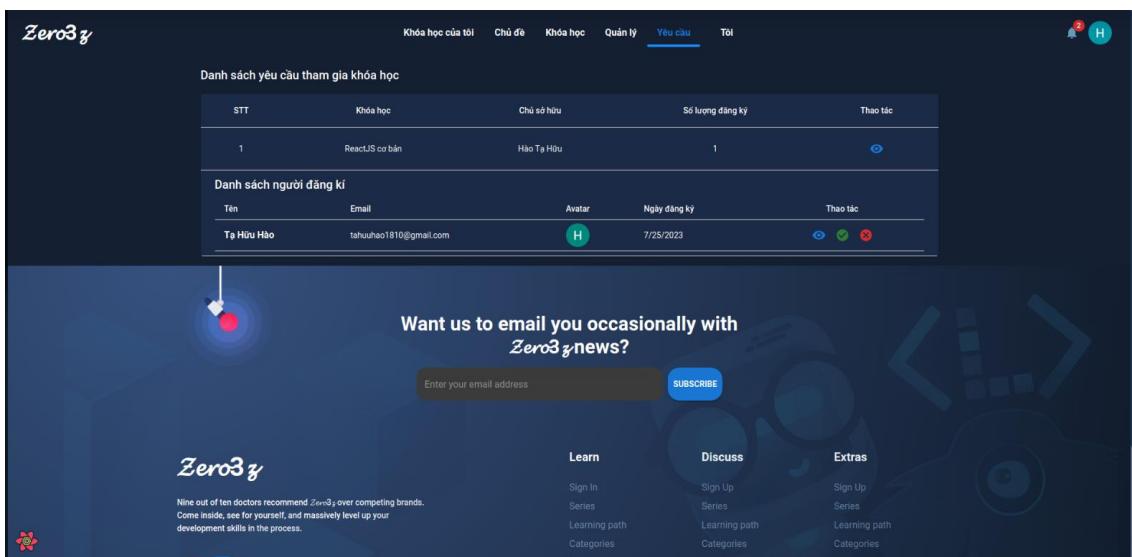
Hình 3.26. Hình ảnh giao diện chi tiết yêu cầu cấp phép khóa học

Hình 3.26 hiển thị chi tiết yêu cầu mở khóa. Nội dung hiển thị là người đã yêu cầu, thời gian tạo yêu cầu, thông tin về khóa học, trạng thái yêu cầu đã được phê duyệt hay chưa. Admin sẽ là người được quyền thao tác cấp phép cho khóa học hay không.



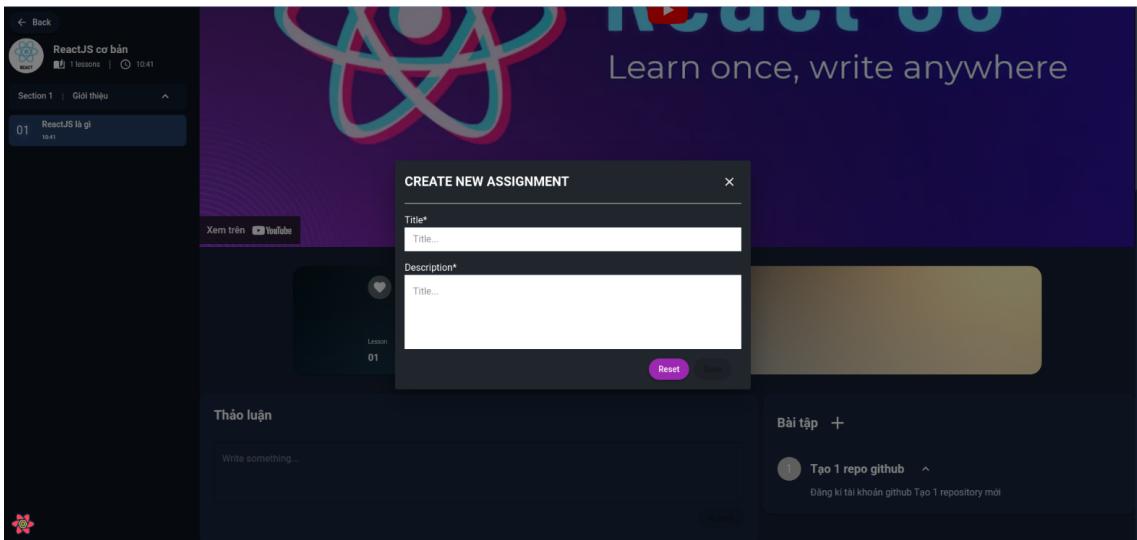
Hình 3.27. Hình ảnh giao diện chi tiết video bài giảng

Hình 3.27 hiển thị chi tiết video bài giảng. Tại đây người dùng sẽ học tập qua video của bài giảng và trao đổi thảo luận về bài giảng bên dưới. Trong bài giảng học viên có thể nhìn thấy bài tập do giáo viên giao và giáo viên tạo bài tập tại đây.



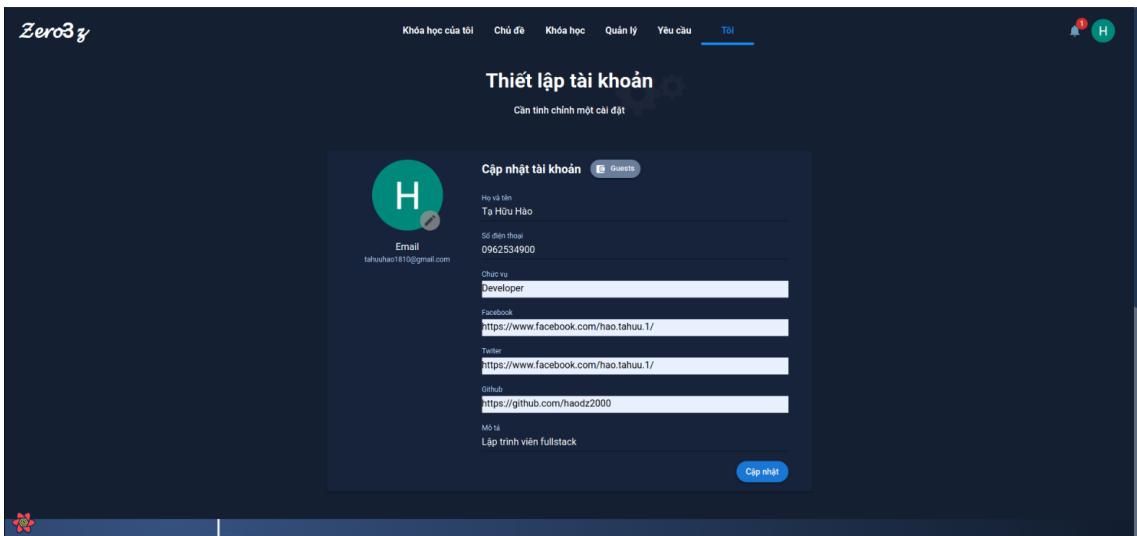
Hình 3.28. Hình ảnh giao diện danh sách yêu cầu tham gia khóa học

Hình 3.28 hiển thị danh sách những khoá học của mình và danh sách người đăng kí của mỗi khoá học. Chỉ chủ khoá học được phép chấp nhận hay từ chối người đăng kí tham gia khoá học.



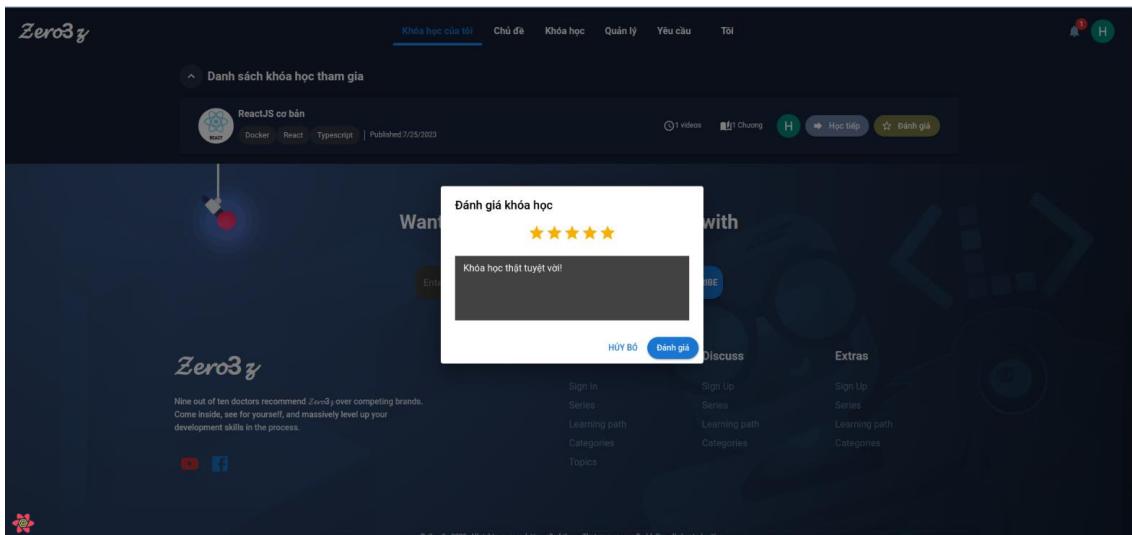
Hình 3.29. Hình ảnh giao diện tạo bài tập

Hình 3.29 hiển thị form tạo bài tập cho mỗi bài giảng. Form tạo bài tập sẽ gồm tên bài tập và mô tả chi tiết bài tập.



Hình 3.30. Hình ảnh giao diện cập nhật thông tin người dùng

Hình 3.30 là giao diện form cập nhật thông tin cá nhân của mỗi người. Mọi người có thể cập nhật thông tin của mình để mọi người khác có thể nhìn thấy thông tin của mình.



Hình 3.31. Hình ảnh giao diện đánh giá khóa học

Hình 3.31 là hình ảnh có danh sách các khóa học mà mình đã tham gia và hệ thống cho phép người dùng tham gia khóa học có thể đánh giá khóa học một lần duy nhất.

Các hình trên đây là các hình thể hiện chức năng chính của hệ thống, ngoài ra còn các giao diện khác mà chưa được đề cập đến.

### 3.3. Triển khai hệ thống lên Cloud AWS

Sau khi phát triển sản phẩm trên môi trường development, sản phẩm sẽ được đưa lên production để triển khai tới người dùng cuối. Hệ thống được lựa chọn đưa lên máy chủ EC2 bằng phương pháp ci/cd gitlab.

Quy trình triển khai phần mềm lên chạy trên máy chủ EC2.

Bước 1: Tạo tài khoản gitlab, tạo một repository trên git lab, đẩy source code dự án lên gitlab.

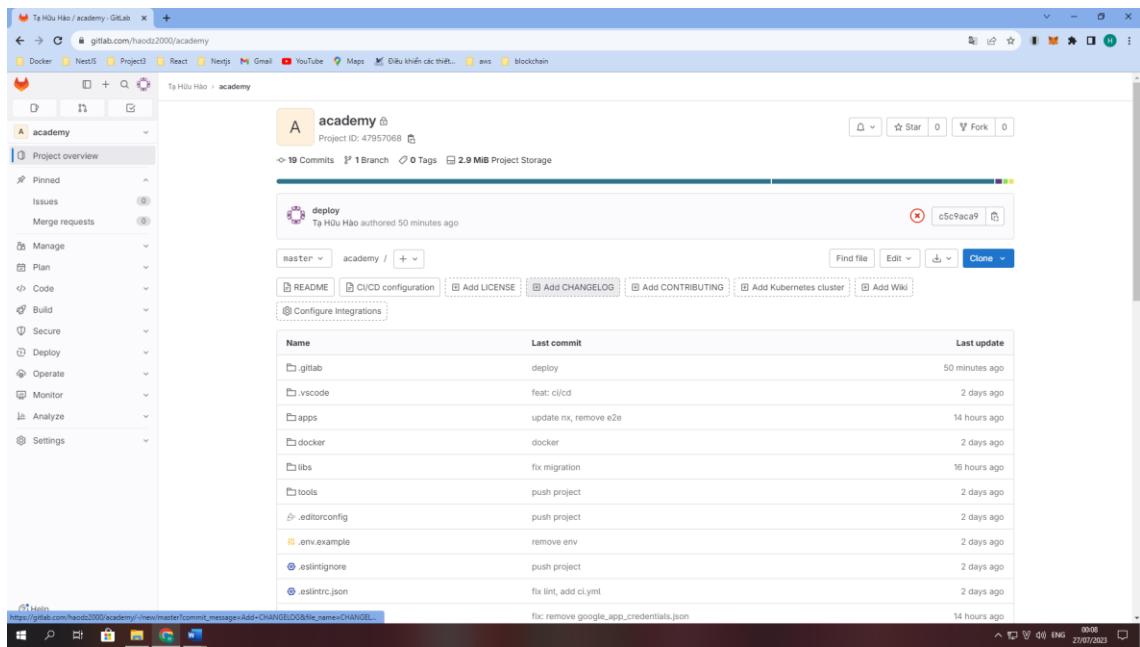
Bước 2: Tạo tài khoản AWS.

Bước 3: Tạo một máy chủ EC2 ubuntu 20.04, sinh ra file .pem lưu trữ cần thận file này để truy cập vào máy chủ.

Bước 4: Truy cập vào máy chủ, cài đặt môi trường cho máy chủ, cài đặt git, cài đặt nginx, cài đặt docker.

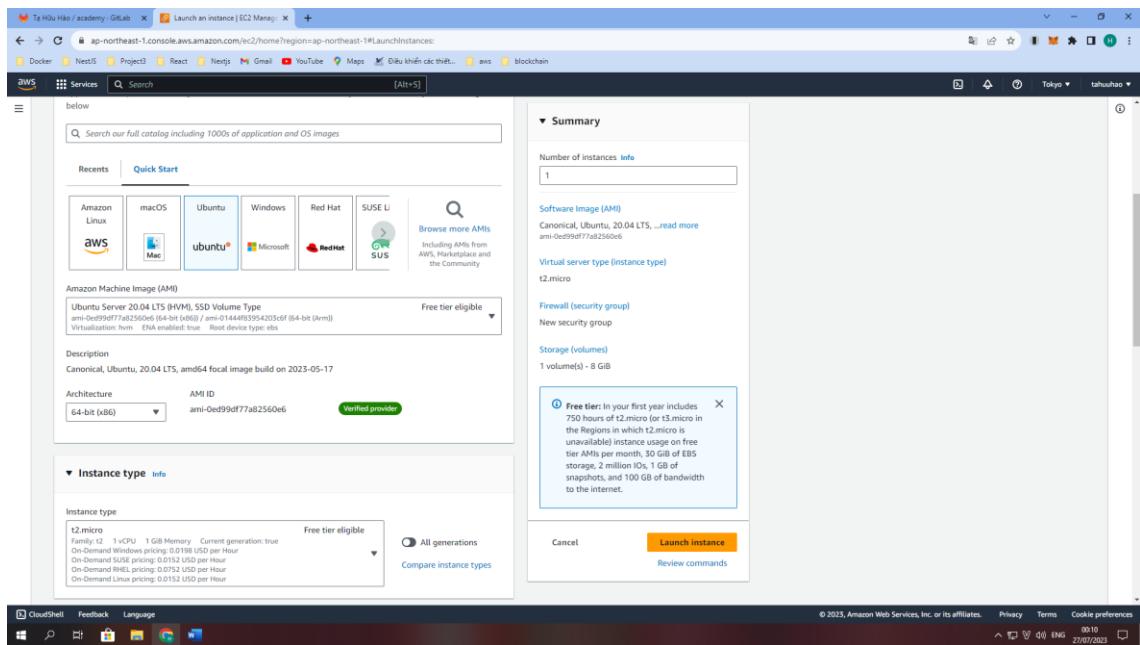
Bước 5. Thiết lập CI/CD ở gitlab, tạo các biến môi trường trên Gitlab cần thiết, viết nội dung file gitlab-ci.yml để tạo ci/cd cho project.

Dưới đây là một số hình ảnh thực hiện deploy code:



Hình 3.32. Hình ảnh source code được đưa lên gitlab

Source code khi được đưa lên gitlab sẽ không đưa những thư mục build, node\_modules để giảm tải kích thước project trên gitlab, không đưa file config .env lên đây để bảo mật thông tin hệ thống.



Hình 3.33. Hình ảnh tạo máy chủ ec2 ubuntu trên AWS

Sau khi có tài khoản AWS, truy cập vào AWS, tìm kiếm dịch vụ EC2, chọn launch instance để tạo máy chủ trên ec2.

```

Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1036-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Wed Jul 26 17:11:09 UTC 2023

System load: 0.0      Processes:           117
Usage of /: 36% of 7.57GB  Users logged in: 1
Memory usage: 31%          IPv4 address for docker0: 172.17.0.1
Swap usage: 0%            IPv4 address for eth0: 172.31.33.31

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run sudo pro status

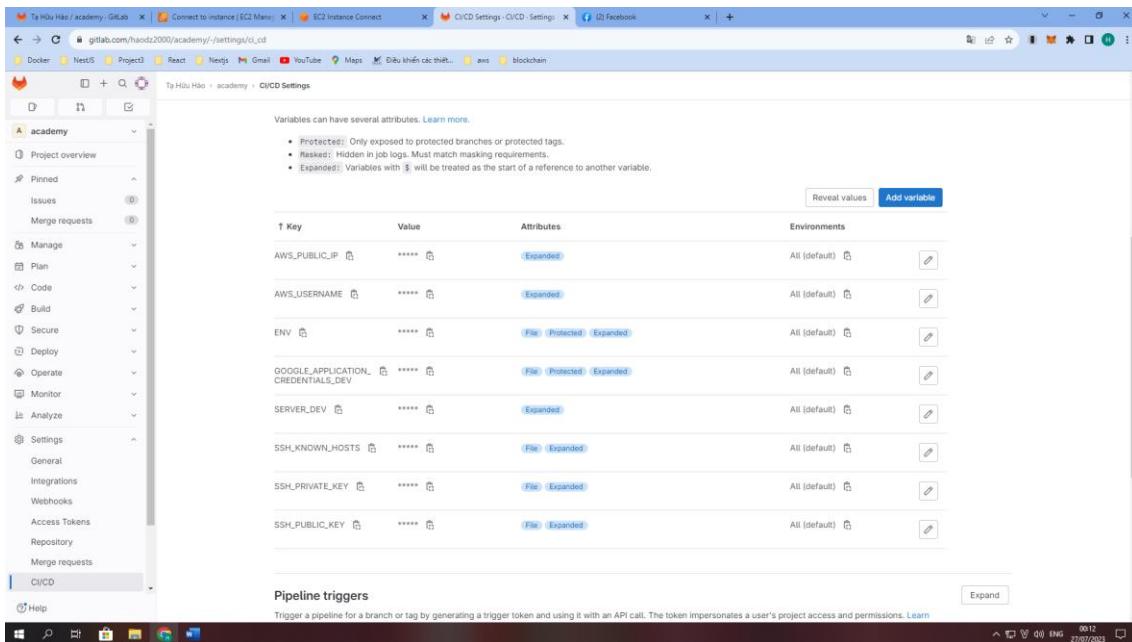
New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Wed Jul 26 16:00:46 2023 from 183.80.56.73
ubuntu@ip-172-31-33-31:~ $ 
```

i-0cf885881f3a19a25 (academy)  
PublicIP: 52.192.84.209 PrivateIP: 172.31.33.31

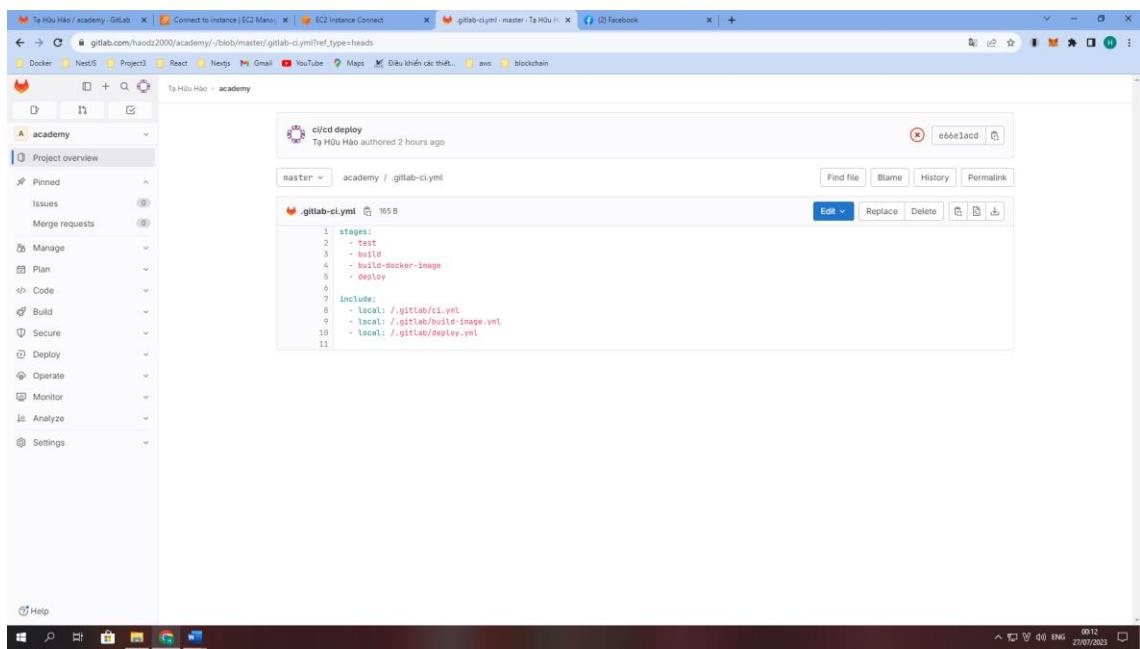
Hình 3.34. Hình ảnh truy cập vào máy chủ

Truy cập thành công vào hệ thống, hệ thống hiển thị terminal, thao tác, giao tiếp với máy chủ EC2 qua terminal bằng các lệnh.



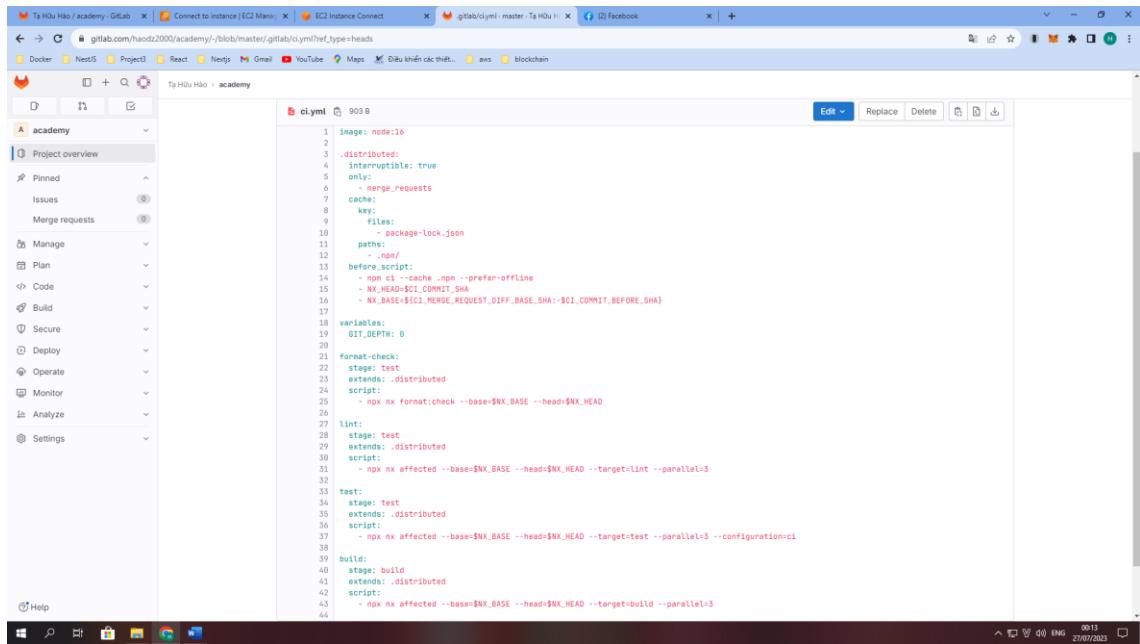
Hình 3.35. Hình ảnh các biến môi trường được sử dụng trong ci/cd

Thiết lập các biến môi trường, các biến này cần được bảo mật không được tiết lộ nên được đưa vào các phần CI/CD Variables của gitlab, chỉ người dùng có quyền cao nhất được xem các biến này trên gitlab.



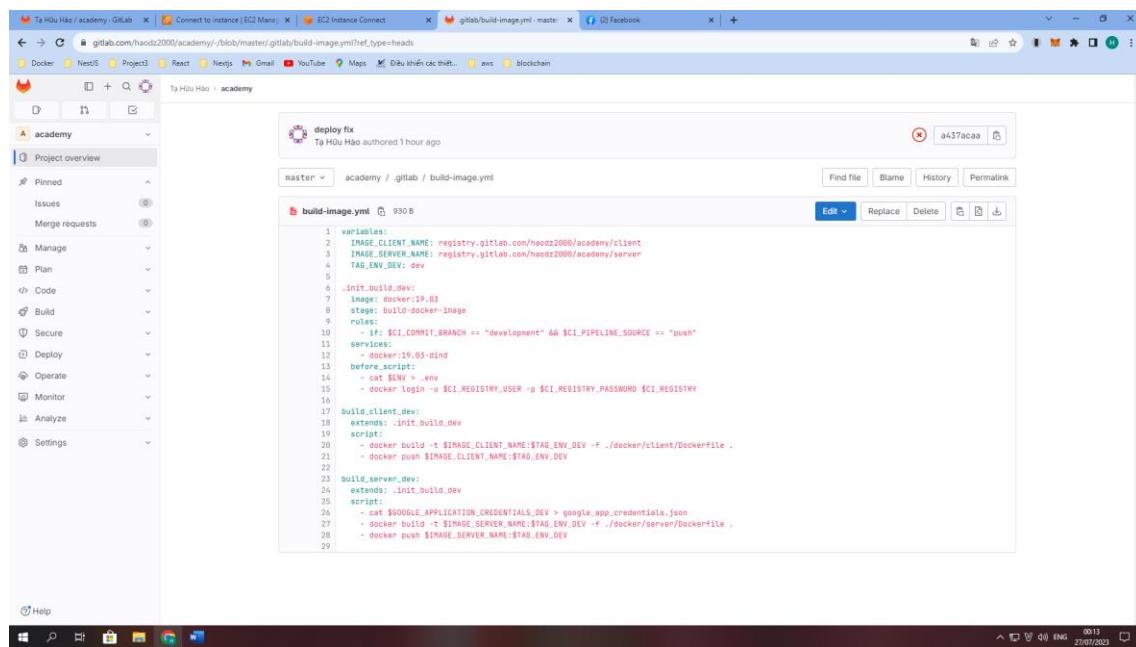
Hình 3.36. Hình ảnh file gitlab-ci.yml để thực thi ci/cd trên gitlab

Mã code để thực hiện CI/CD gitlab, khi có file gitlab-ci.yml, gitlab runner trên gitlab có sẵn hoặc mình tạo sẽ tìm file này và thực thi quá trình CI/CD, trong các bước thực thi có bước nào thất bại thì quá trình CI/CD sẽ thất bại nếu chúng ta không có ép buộc bỏ qua lỗi đó.



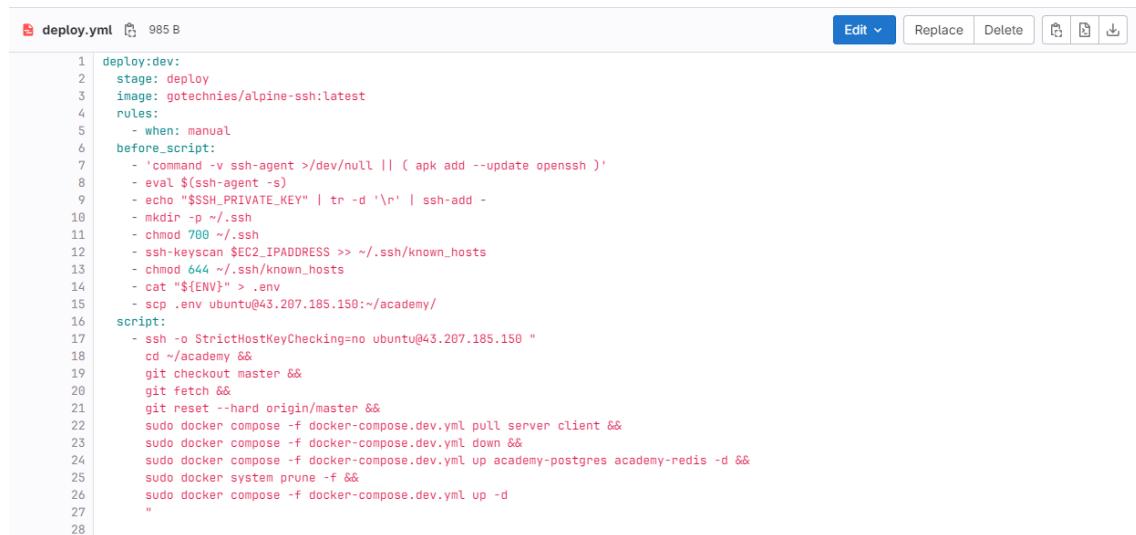
Hình 3.37. Hình ảnh file ci.yml thực hiện stage test, build

Quá trình này để test các chuẩn của code trong project khi đưa lên gitlab đã đúng format theo yêu cầu của project đặt ra chua.



Hình 3.38. Hình ảnh file build-image.yml để thực hiện stages build-docker-image

Quá trình này thực hiện nhiệm vụ đóng gói project thành image docker và đẩy image lên kho lưu trữ container registry của gitlab.



Hình 3.39. Hình ảnh file deploy.yml thực hiện stage deploy

Sau khi thực hiện các bước trên chúng ta truy cập vào máy chủ trên cloud thiết lập cài đặt cần thiết trên máy chủ. Trên máy chủ chúng ta cần cài đặt docker, git, nginx, tạo ssh key cho phép truy cập vào gitlab. Từ máy chủ clone project từ gitlab về sau đó chúng ta bắt đầu chạy cicd trên gitlab. Đầu tiên chúng ta đưa “image” lên kho lưu trữ “container registry” của git lab, sau khi đưa “image” lên thành công chúng ta sẽ tiến hành “deploy” sản phẩm. Truy cập website qua địa chỉ ip.

Sau khi deploy đã thành công, chúng ta sẽ mua tên miền, cấu hình DNS cho tên miền, tạo chúng chỉ TLS cho tên miền trên CloudFlare. Trỏ tên miền về server,

cấu hình nginx cho server trả đến tên miền. Sau khi thành công, chúng ta sẽ truy cập website qua địa chỉ tên miền để bảo mật và không bị lộ ip, chống tấn công máy chủ.

### **3.4. Kết luận**

Trong chương 3 này, em đã đi thực hiện các bước để xây dựng 1 hệ thống học trực tuyến. Thu thập các yêu cầu xây dựng chức năng của hệ thống, lựa chọn mô hình kiến trúc hệ thống, xây dựng chức năng hoàn chỉnh cho hệ thống, vẽ các sơ đồ mô usecase cho các chức năng của hệ thống. Tiến hành xây dựng cơ sở dữ liệu, xây dựng server, giao diện người dùng, đưa sản phẩm chạy trên máy chủ aws ec2. Qua đó đã giúp em hiểu về cách xây dựng một hệ thống từ ý tưởng đến xây dựng một sản phẩm phần mềm hoàn chỉnh.

## CHƯƠNG 4: KIỂM THỬ & HƯỚNG PHÁT TRIỂN

### 4.1. Kiểm thử

Kiểm thử là một quá trình vô cùng quan trọng trong các bước phát triển hệ thống phần mềm. Trước khi đưa sản phẩm đến với người dùng hệ thống cần được kiểm tra từ môi trường “developer” đến môi trường “production” để đảm bảo hệ thống chạy đúng yêu cầu đưa ra, tránh gây sai sót dữ liệu, gây ra lỗi khi khách hàng sử dụng dịch vụ. Chính vì điều đó em đã tiến hành kiểm thử từng yêu cầu của hệ thống đề ra. Dưới đây là các chức năng đã được kiểm thử

Bảng 4.1. Bảng danh sách testcase

Chức năng	Yêu cầu	Kết quả
Đăng nhập	Hệ thống yêu cầu đăng nhập bằng tài khoản google mã hoá thông tin người dùng bằng JWT và trả về token xác thực người dùng , nếu người dùng đăng nhập lần đầu thì sẽ đăng ký người dùng với quyền người dùng, tạo ví với số dư 0.	Đạt
Tạo khoá học	Yêu cầu người dùng đăng nhập, bất kì ai cũng tạo được khoá học nhưng khoá học ở dạng chờ, cần xin cấp phép tới admin để có thể công khai khoá học.	Đạt
Cập nhật khoá học	Chỉ chủ khoá học được cập nhật thông tin khoá học.	Đạt
Tạo chương, video bài giảng	Chỉ chủ khoá học được tạo chương, video bài giảng cho khoá học của mình. Đối với tạo video bài giảng, cho phép đăng tải video theo hai lựa chọn là video từ máy tính hoặc dùng link youtube.	Đạt
Cập nhật chương, video bài giảng	Chỉ chủ khoá học được thực hiện. Cập nhật lại thông tin của chương và video bài giảng.	Đạt
Xem ,tạo mới, cập nhật bài tập	Chủ khoá học có thể tạo mới, cập nhật bài tập. Mỗi học viên tham gia khoá học có thể xem thông tin bài tập, Bài tập của bài giảng nào chỉ xuất hiện trong bài giảng đó.	Đạt
Tạo thảo luận, xem thảo luận	Chỉ các học viên tham gia khoá học mới xem được thảo luận trong mỗi bài học. Thảo luận của bài giảng nào chỉ xuất hiện trong bài giảng đó. Học viên tạo bình luận trong bài giảng.	Đạt
Danh sách khoá học, chủ đề , danh mục	Các khoá học, danh mục, chủ đề hiển thị liên kết với nhau. Mỗi danh mục có các chủ đề tương ứng. Mỗi chủ đề sẽ có các khoá học ứng với mỗi chủ đề đó.	Đạt
Yêu cầu cấp phép	Chủ khoá học xin cấp phép cho khoá học của mình đến admin, chỉ admin được quyền cấp phép cho khoá học, khoá học được cấp phép sẽ được đưa vào sử dụng cho mọi tham gia người học, yêu cầu bị từ chối chủ khoá học phải yêu cầu lại đến khi được sự cho phép.	Đạt

Yêu cầu tham gia khoá học	Đối với khoá học miễn phí, học viên gửi yêu cầu xin tham gia khoá học đến chủ khoá học, chờ được phê duyệt sẽ được tham gia học tập. Chủ khoá học xem được danh sách yêu cầu tham gia học cho từng khoá học của mình. Chỉ chủ khoá học được phê duyệt nó.	Đạt
	Đối với khoá học mất phí, ví số dư của người dùng phải đủ so với phí khoá học và khi tham gia thì ví sẽ bị trừ và được tham gia vào khoá học luôn.	Đang thực hiện
Khoá học tham gia	Hiển thị danh sách các khoá học của người dùng tham gia. Cho phép đánh giá khoá học một lần.	Đạt
Tài khoản	Cập nhật thông tin cá nhân.	Đạt
	Nạp tiền vào ví.	Đang thực hiện
	Quản lý giao dịch.	Đang thực hiện
Thông báo	Tạo thông báo khi có sự kiện xảy ra liên quan đến khoá học, gửi thông báo đến người dùng sử dụng Fcm của google firebase.	Đạt
	Hiển thị thông báo chưa đọc đến với người dùng.	Đạt

## 4.2. Hướng phát triển

Trong tương lai, hệ thống sẽ không ngừng phát triển và xây dựng thêm các tính năng mới phục vụ nhu cầu mọi người. Hệ thống sẽ xây dựng mô hình thanh toán tự động để mọi người có thể nạp tiền tự động, thanh toán hoá đơn khoá học tự động. Xây dựng thêm các tính năng như đăng tải bài viết, chia sẻ kiến thức, ứng dụng api chat GPT để làm chatbot, nâng cao bảo mật người dùng, mã hoá các video khoá học để không bị lấy cắp dữ liệu, vv.

## KẾT LUẬN

Trên đây là toàn bộ nội dung đồ án tốt nghiệp của em với đề tài: “Triển khai hệ thống học lập trình trực tuyến lên cloud AWS”.

Sau khi trải qua quá trình học tập tại trường và đặc biệt là một số môn học cơ sở ngành và chuyên ngành như: Kỹ thuật phần mềm ứng dụng, Cấu trúc dữ liệu và giải thuật, hệ điều hành, Kỹ thuật lập trình, Lập trình nâng cao, Tư duy công nghệ và thiết kế kỹ thuật..., cùng với quá trình thực tập kĩ thuật, thực tập tốt nghiệp tại các doanh nghiệp công ty, em đã tích luỹ và trang bị được cho mình một hệ thống kiến thức chuyên môn cơ bản về thiết kế phần mềm. Cùng với sự tìm tòi học hỏi, nghiên cứu các tài liệu tham khảo, cũng như sự nỗ lực không ngừng của bản thân và đặc biệt dưới sự hướng dẫn tận tình của cô giáo Th.S Đinh Thị Nhụng, em đã hoàn thành đồ án theo đúng thời gian quy định.

Qua bản đồ án này, em hiểu và nắm bắt được các bước thiết kế hệ thống phần mềm: Tìm hiểu thị trường, mục tiêu, xây dựng kiến trúc hệ thống, quy trình phát triển sản phẩm, tìm hiểu về các công nghệ mới trong phát triển phần mềm, tìm hiểu về dựng máy chủ, cấu hình máy chủ trên AWS để chạy phần mềm. Dựa sản phẩm từ suy nghĩ đến sản phẩm thật để mọi người sử dụng.

Trong quá trình làm đồ án tốt nghiệp về “Triển khai hệ thống học lập trình trực tuyến lên cloud AWS”, em đã dựa trên nền tảng kiến thức lý thuyết được học tại trường và áp dụng các kiến thức đã được học hỏi, trải nghiệm qua thời gian thực tập kĩ thuật và thực tập tốt nghiệp tại công ty. Tuy nhiên đồ án vẫn không thể tránh khỏi những thiếu sót nên em mong sẽ nhận được những nhận xét, góp ý từ các thầy cô giáo trong bộ môn để em có thể rút kinh nghiệm, hoàn thiện hơn cũng như tích luỹ thêm kiến thức cho bản thân để có thể ứng dụng vào công việc sau khi ra trường.

Đồ án tốt nghiệp không chỉ giúp em củng cố kiến thức chuyên ngành mà còn bồi dưỡng thêm cho em các kỹ năng mềm như đọc hiểu, kỹ năng tin học văn phòng,... Mong rằng đồ án này sẽ trở thành một tài liệu tham khảo và có thể ứng dụng trong thực tiễn.

Thông qua đây, một lần nữa em xin gửi lời cảm ơn chân thành đến các thầy cô giáo trong trường Điện - Điện Tử và đặc biệt là cô giáo Th.S Đinh Thị Nhụng đã tận tâm chỉ dạy, hướng dẫn giúp đỡ chúng em rất nhiều trong suốt quá trình học tập cũng như thực hiện đồ án tại trường Điện-Điện Tử Đại học Bách Khoa Hà Nội.

Em xin chân thành cảm ơn !

**Sinh viên thực hiện**  
(Ký và ghi rõ họ tên)

## TÀI LIỆU THAM KHẢO

- [1] [https://docs.gitlab.com/ee/ci/cloud\\_deployment/](https://docs.gitlab.com/ee/ci/cloud_deployment/)
- [2] <https://nx.dev/getting-started/installation>
- [3] <https://docs.docker.com/>
- [4] <https://aws.amazon.com/ec2/>
- [5] <https://aws.amazon.com/s3/>
- [6] <https://aws.amazon.com/rds/>
- [7] <https://docs.nestjs.com/>
- [8] <https://nextjs.org/docs>
- [9] <https://mikro-orm.io/docs/>
- [10] <https://socket.io/>
- [11] <https://www.postgresql.org/docs/>
- [12] [https://viblo.asia/p/quy-trinh-phat-trien-phan-mem-va-cac-mo-hinh-phat-trien-phan-mem-gAm5ybP8Kdb](https://viblo.asia/p/quy-trinh-phat-trien-phan-mem-vacac-mo-hinh-phat-trien-phan-mem-gAm5ybP8Kdb)
- [13] <https://hub.docker.com/r/openapitools/openapi-generator-cli>
- [14] <https://www.n-ix.com/microservices-vs-monolith-which-architecture-best-choice-your-business/>
- [15] <https://viblo.asia/p/setup-gitlab-cicd-reactjs-app-len-aws-ec2-6J3ZgRPqKmB>
- [16] <https://stackoverflow.com/>
- [17] <https://www.typescriptlang.org/docs/>
- [18] <https://nginx.org/en/docs/>