

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN

HỌC PHẦN: THỰC TẬP TỐT NGHIỆP

CHỦ ĐỀ 2: NGHIÊN CỨU CÁC CÔNG NGHỆ VÀ XÂY DỰNG,

PHÁT TRIỂN WEBSITE

**TÊN ĐỀ TÀI: PHÁT TRIỂN VÀ TRIỂN KHAI HỆ THỐNG BACK-END
CHO WEBSITE LÀNG NỔI TÂN LẬP TẠI CÔNG TY CỔ PHẦN TẬP
ĐOÀN HT VIỆT NAM**

LỚP TÍN CHỈ: TTTN.03.K13.05.LH.C04.1_LT

Giảng viên giảng dạy: ThS. Đặng Khánh Trung

Sinh viên thực hiện: Nguyễn Trí Dũng

Mã sinh viên: 20223155

Lớp hành chính: DCCNTT 13.10.16

Bắc Ninh - 2025

CÔNG TY CÔNG TY CỔ PHẦN TẬP
ĐOÀN HT VIỆT NAM

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

GIẤY XÁC NHẬN THỰC TẬP

Họ tên cán bộ hướng dẫn: Nguyễn Hữu Phương
Đơn vị thực tập: Công ty cổ phần tập đoàn HT Việt Nam
Địa chỉ thực tập: Hẻm T. 1 P. Nguyễn Huệ Quận 1, Nhân Chính, Thanh Xuân, HN
Điện thoại liên hệ: 0966 276 868
Họ tên sinh viên thực tập: Nguyễn Trí Dũng, Lớp: ĐCCTI 13.10.16
Ngày sinh: 11/10/2004, Điện thoại: 0378 519 357
Thời gian thực tập: từ ngày 19/07/2025 đến ngày 5/10/2025
Bảng đánh giá nhận xét của công ty:

TT	Nội dung đánh giá	Tốt	Khá	Trung bình	Kém
1	Thực hiện nội quy của công ty	✓			
2	Thái độ thực tập	✓			
3	Ý thức kỷ luật	✓			
4	Tác phong	✓			
5	Kiến thức chuyên môn	✓			
6	Khả năng hoàn thành công việc	✓			
7	Khả năng hòa nhập và thích nghi công việc	✓			
8	Tính năng động và sáng tạo trong công việc	✓			

Nhận xét chung:

Đánh giá:

Bắc Ninh, ngày 29 tháng 09 năm 2025

Cán bộ hướng dẫn

(Ký và ghi rõ họ tên)



PHÓ GIÁM ĐỐC TRUNG TÂM
Phạm Tiến Hùng

Nguyễn Hữu Phương

LỜI CẢM ƠN

Trong suốt quá trình học tập và rèn luyện tại Trường Đại học Công nghệ Đông Á, đặc biệt là khi thực hiện Bài tập lớn của học phần Thực tập tốt nghiệp với đề tài “Phát triển và triển khai hệ thống Back-End cho Website Làng nổi Tân Lập tại Công ty Cổ phần Tập đoàn HT Việt Nam”, em đã nhận được rất nhiều sự quan tâm, hỗ trợ và giúp đỡ tận tình từ các thầy cô giáo, gia đình, bạn bè và doanh nghiệp nơi em thực tập.

Trước hết, em xin gửi lời cảm ơn chân thành và sâu sắc tới Ban Giám hiệu Trường Đại học Công nghệ Đông Á, Khoa Công nghệ Thông tin đã tạo điều kiện cho em được học tập trong một môi trường hiện đại, năng động và chuyên nghiệp, nơi em có thể tích lũy kiến thức chuyên môn cũng như kỹ năng thực tiễn phục vụ cho công việc sau này.

Em đặc biệt xin bày tỏ lòng biết ơn tới thầy Đặng Khánh Trung, giảng viên trực tiếp hướng dẫn, đã luôn tận tình chỉ bảo, định hướng, đóng góp nhiều ý kiến quý báu giúp em hoàn thành bài tập lớn một cách nghiêm túc, khoa học và hiệu quả. Sự hỗ trợ, động viên và truyền đạt kinh nghiệm quý báu từ thầy đã giúp em hiểu rõ hơn về chuyên môn, có thêm niềm tin và động lực trong quá trình nghiên cứu, phát triển và triển khai hệ thống.

Em cũng xin gửi lời cảm ơn tới Công ty Cổ phần Tập đoàn HT Việt Nam, nơi em có cơ hội được tiếp cận với môi trường làm việc thực tế, trực tiếp tham gia dự án và áp dụng những kiến thức đã học vào công việc. Sự hỗ trợ của các anh chị trong công ty, từ việc hướng dẫn quy trình làm việc, chia sẻ kinh nghiệm thực tiễn cho tới cung cấp tài liệu và cơ sở vật chất, đã giúp em mở rộng tầm nhìn, rèn luyện kỹ năng làm việc nhóm và nâng cao năng lực chuyên môn.

Do thời gian có hạn và kiến thức còn hạn chế, bài tập lớn chắc chắn không tránh khỏi những thiếu sót. Em rất mong tiếp tục nhận được sự góp ý và chỉ bảo từ các thầy cô, bạn bè và các anh chị để hoàn thiện hơn trong những lần nghiên cứu, triển khai sau.

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT	iv
DANH MỤC HÌNH VẼ VÀ BẢNG BIỂU	v
CHƯƠNG I: KHÁI QUÁT VỀ DOANH NGHIỆP	1
1. Thông tin chung	1
2. Lĩnh vực và phạm vi hoạt động	1
3. Quy mô tổ chức	2
4. Vị trí thực tập của sinh viên	3
5. Công việc và trách nhiệm được giao	3
CHƯƠNG II: CHUYÊN MÔN VÀ BÀI TOÁN ĐẶT RA	6
2.1 Bối cảnh vấn đề	6
2.2 Phân tích nhu cầu hệ thống	12
2.2.1 Danh sách các chức năng chính	12
2.2.2 Mô tả chi tiết từng chức năng	13
2.2.3 Các trường hợp sử dụng (use case)	15
2.3 Các yêu cầu bổ trợ (phi chức năng)	21
2.3.1 Hiệu suất và khả năng mở rộng	21
2.3.2 Bảo mật và an toàn dữ liệu	22
2.3.3 Tính ổn định và khả năng bảo trì	22
2.4 Hướng tiếp cận và phương án xử lý	23
2.4.1 Lý do lựa chọn công nghệ	24
2.4.2 Quy trình triển khai (phân tích → thiết kế → code → test)	24
2.4.3 Các khó khăn gặp phải và cách khắc phục	25
2.4.4 Kinh nghiệm và kỹ năng tích lũy được	25
CHƯƠNG III: KẾT QUẢ VÀ GIẢI PHÁP	27

3.1 Xây dựng và thiết kế hệ thống	27
3.1.1 Phân tích yêu cầu từ nghiệp vụ	27
3.1.2 Kiến trúc tổng thể	27
3.1.3 Thiết kế chi tiết	28
3.2 Nền tảng công nghệ và giải pháp ứng dụng	37
3.2.1 Công nghệ chủ đạo được sử dụng	37
3.2.2 Giải pháp kỹ thuật áp dụng	38
3.2.3 Công cụ và tài liệu nghiên cứu	38
3.3 Đánh giá thành quả đạt được	39
3.3.1 Kết quả cuối cùng	39
3.3.2 So sánh với mục tiêu ban đầu	40
3.3.3 Bài học, đúc kết kinh nghiệm	40
CHƯƠNG IV: TỔNG KẾT VÀ ĐỊNH HƯỚNG	42
4.1 Tổng kết quá trình thực tập	42
4.1.1 Tóm lược khối lượng và nội dung công việc đã hoàn thành	42
4.1.2 Nhận xét, đánh giá từ góc nhìn cá nhân	42
4.2 Định hướng và kiến nghị	43
4.2.1 Gợi ý cải tiến kỹ thuật/hệ thống	43
4.2.2 Kiến nghị có lợi cho doanh nghiệp	44
4.2.3 Định hướng nghiên cứu và phát triển bản thân	45
TÀI LIỆU THAM KHẢO	46

DANH MỤC CÁC TỪ VIẾT TẮT

STT	Chữ viết tắt	Giải thích
1	IT Department	Information Technology Department
2	API	Application Programming Interface
3	ERD	Entity Relationship Diagram
4	REST	Representational State Transfer
5	JWT	JSON Web Token
6	CSRF	Cross-Site Request Forgery
7	XSS	Cross-Site Scripting
8	HTTPS	HyperText Transfer Protocol Secure
9	PSR	PHP Standards Recommendation
10	SSR	Server-side Rendering
11	SSG	Static Site Generation

DANH MỤC HÌNH VẼ VÀ BẢNG BIỂU

Hình 1. 1: Logo công ty HT Việt Nam	1
Hình 1. 2: Hệ thống API	4
Hình 1. 3: Logo làng nổi tân lập	6
Hình 2. 1: Trang chủ website	7
Hình 2. 2: Tầm nhìn & Sứ mệnh Làng Nổi Tân Lập	8
Hình 2. 3: Vài nét về làng nổi tân lập	8
Hình 2. 4: Cung đường xuyên rừng tràm	9
Hình 2. 5: Hồ bán nguyệt	9
Hình 2. 6: Giá và dịch vụ theo mùa	10
Hình 2. 7: Phòng khách sạn	10
Hình 2. 8: Thông báo tuyển dụng	11
Hình 2. 9: Đặt phòng khách sạn	11
Bảng 2. 1: Use case thanh toán	15
Bảng 2. 2: Use case đăng nhập	16
Bảng 2. 3: Use case đặt phòng	17
Bảng 2. 4: Use case đổi mật khẩu	18
Bảng 2. 5: Use case thêm hình ảnh vào Media	19
Bảng 2. 6: Use case quản lý Slider (Banner)	20
Bảng 2. 7: Use case quản lý Danh mục (Categories)	21
Hình 3. 1: ERD (sơ đồ quan hệ dữ liệu)	28
Bảng 3. 1: Bảng Users	29
Bảng 3. 2: Bảng Roles	29
Bảng 3. 3: Bảng Posts	30
Bảng 3. 4: Bảng Categories	31

Bảng 3. 5: Bảng Media	31
Bảng 3. 6: Bảng Sliders	32
Bảng 3. 7: Bảng Post_media	32
Bảng 3. 8: API cho Auth	33
Bảng 3. 9: API cho Users	33
Bảng 3. 10: API cho Roles	34
Bảng 3. 11: API cho Posts	34
Bảng 3. 12: API cho Categories	35
Bảng 3. 13: API cho Media	35
Bảng 3. 14: API cho Sliders	36
Bảng 3. 15: API cho Post_media	36

CHƯƠNG I: KHÁI QUÁT VỀ DOANH NGHIỆP

1. Thông tin chung

Công ty Cổ phần Tập đoàn HT Việt Nam (tên viết tắt: HTS VN., JSC), được thành lập từ ngày 21/12/2011, là một doanh nghiệp cổ phần ngoài Nhà nước có mã số thuế 0105746500. Trụ sở chính của công ty đặt tại tầng 5M, Tòa Hei Tower, số 1 đường Nguyễn Như Kon Tum, phường Nhân Chính, quận Thanh Xuân, thành phố Hà Nội, Việt Nam. Đây là vị trí trung tâm thủ đô, thuận lợi cho việc giao thương, hợp tác và triển khai các dự án công nghệ với các đối tác trong và ngoài nước.



Hình 1. 1: Logo công ty HT Việt Nam

Trong hơn một thập kỷ hình thành và phát triển, HT Việt Nam đã không ngừng khẳng định vị thế của mình trong lĩnh vực công nghệ thông tin, xử lý dữ liệu và các dịch vụ số. Dưới sự điều hành của ông Phạm Thế Hải – người đại diện pháp luật, doanh nghiệp đã tham gia và dẫn dắt nhiều dự án công nghệ trọng điểm, mở rộng phạm vi hoạt động ra nhiều lĩnh vực có tính cạnh tranh cao như dịch vụ trò chơi trực tuyến và giải trí số. Công ty định hướng phát triển theo triết lý:

Tầm nhìn: Trở thành tập đoàn công nghệ hàng đầu tại Việt Nam, tiên phong trong việc cung cấp giải pháp dữ liệu, hệ thống thông tin và dịch vụ số.

Sứ mệnh: Mang lại các sản phẩm và dịch vụ công nghệ hiện đại, hiệu quả, an toàn, góp phần nâng cao trải nghiệm người dùng và hỗ trợ doanh nghiệp tối ưu hoạt động.

Giá trị cốt lõi: Sáng tạo – Uy tín – Chất lượng – Hợp tác bền vững. Đây là kim chỉ nam trong mọi hoạt động, giúp công ty duy trì niềm tin với khách hàng và đối tác.

2. Lĩnh vực và phạm vi hoạt động

HT Việt Nam hoạt động trong nhiều mảng công nghệ, nổi bật nhất là xử lý dữ liệu, cho thuê hạ tầng và phát triển các giải pháp công nghệ thông tin tích hợp. Ngoài ra, công ty còn mở rộng sang các dịch vụ trò chơi điện tử trực tuyến, trò chơi trên

mạng và trò chơi công cộng – những lĩnh vực đòi hỏi tính sáng tạo, bảo mật cao và khả năng vận hành ổn định với số lượng người dùng lớn. Các dịch vụ và sản phẩm chính bao gồm:

Xây dựng và triển khai hệ thống phần mềm quản lý theo yêu cầu.

Cung cấp dịch vụ lưu trữ dữ liệu, xử lý dữ liệu lớn (Big Data).

Triển khai giải pháp thương mại điện tử, du lịch trực tuyến và các nền tảng dịch vụ số.

Phát triển, vận hành các ứng dụng trò chơi trực tuyến và hệ thống giải trí công cộng.

Về đối tượng khách hàng, công ty hướng đến cả khối doanh nghiệp (cần các giải pháp số hoá, dữ liệu và phần mềm quản lý) lẫn khách hàng cá nhân (người dùng dịch vụ số, trò chơi điện tử, các nền tảng giải trí và du lịch trực tuyến). Thị trường mục tiêu trải dài trên phạm vi cả nước, với định hướng từng bước mở rộng sang khu vực Đông Nam Á.

3. Quy mô tổ chức

HT Việt Nam hiện có đội ngũ nhân sự chuyên môn cao với hàng trăm kỹ sư phần mềm, chuyên viên phân tích dữ liệu, lập trình viên và cán bộ quản lý dự án. Cơ cấu tổ chức được chia thành nhiều phòng ban chức năng: phòng Công nghệ thông tin (IT Department), phòng Kinh doanh, phòng Dịch vụ khách hàng, phòng Marketing, phòng Tài chính – Kế toán... Mỗi bộ phận đều có vai trò quan trọng trong chuỗi vận hành, bảo đảm doanh nghiệp hoạt động hiệu quả và đồng bộ.

Bên cạnh trụ sở chính tại Hà Nội, HT Việt Nam còn hợp tác với nhiều đối tác công nghệ, nhà cung cấp dịch vụ hạ tầng và đơn vị nghiên cứu trong nước và quốc tế. Điều này giúp công ty duy trì tính cạnh tranh, tiếp cận các công nghệ mới và triển khai dự án với quy mô lớn. Chính nhờ mạng lưới liên kết chặt chẽ này, HT Việt Nam đã xây dựng được một hệ sinh thái công nghệ bền vững, có khả năng đáp ứng nhanh chóng nhu cầu thị trường.

4. Vị trí thực tập của sinh viên

Trong kỳ thực tập tốt nghiệp, em được phân công tại phòng Công nghệ thông tin (IT Department), thuộc nhóm phát triển Back-end của dự án xây dựng website Làng nổi Tân Lập – một nền tảng trực tuyến chuyên cung cấp thông tin du lịch, đặt tour, khách sạn và dịch vụ giải trí.

Người trực tiếp hướng dẫn em là trưởng nhóm kỹ thuật (Team Leader) phụ trách back-end, đồng thời em được giao giữ vai trò Back-end Lead trong nhóm thực tập. Nhiệm vụ chính của em không chỉ là tham gia phát triển các chức năng back-end mà còn điều phối công việc trong nhóm, phụ trách các module lõi của hệ thống. Vai trò ban đầu của em bao gồm:

Thiết kế và quản lý cơ sở dữ liệu (Database) cho hệ thống.

Phát triển các API cốt lõi (auth, users, roles, posts, categories, media, sliders).

Kiểm soát logic nghiệp vụ, xử lý phân quyền người dùng, quản lý nội dung và tài nguyên đa phương tiện.

Hỗ trợ các thành viên trong nhóm khi gặp khó khăn kỹ thuật, đảm bảo tiến độ và chất lượng code.

Qua vị trí này, em không chỉ thực hiện các công việc chuyên môn mà còn rèn luyện được kỹ năng lãnh đạo nhóm, quản lý tiến độ và phối hợp làm việc nhóm trong môi trường doanh nghiệp thực tế.

5. Công việc và trách nhiệm được giao

Ngay từ khi bắt đầu kỳ thực tập, em đã được công ty tin tưởng giao cho những nhiệm vụ quan trọng gắn liền với chuyên môn back-end. Đây là một thử thách lớn nhưng cũng là cơ hội quý giá để em vận dụng kiến thức đã học và phát triển thêm các kỹ năng thực tế trong môi trường doanh nghiệp.

Thứ nhất, về công việc chuyên môn: em chịu trách nhiệm chính trong việc xây dựng và quản lý cơ sở dữ liệu (Database) cho dự án website Làng nổi Tân Lập. Hệ thống cơ sở dữ liệu bao gồm 14 bảng, trong đó em phụ trách thiết kế và triển khai 7 bảng nòng cốt: auth, users, roles, media, sliders, posts, categories và post_media. Các bảng này đóng vai trò trung tâm, quyết định sự vận hành của hệ thống từ khâu xác

thực đăng nhập, phân quyền người dùng, quản lý nội dung bài viết cho đến lưu trữ và hiển thị hình ảnh, banner quảng bá. em đã tiến hành thiết kế sơ đồ quan hệ (ERD), chuẩn hóa dữ liệu, thiết lập khóa chính – khóa ngoại, đảm bảo tính toàn vẹn và khả năng mở rộng lâu dài của hệ thống.

Thứ hai, về phát triển API: trên cơ sở cơ sở dữ liệu đã thiết kế, em trực tiếp xây dựng các API chuẩn RESTful cho phép thực hiện các thao tác CRUD (Create – Read – Update – Delete). Ví dụ: API đăng ký, đăng nhập, đổi mật khẩu trong module auth; API thêm/sửa/xóa người dùng trong module users; API quản lý bài viết và danh mục trong posts và categories; API quản lý hình ảnh trong media và sliders. Mỗi API đều được thiết kế có đầy đủ đầu vào, đầu ra, cơ chế kiểm tra dữ liệu, trả về mã phản hồi (response code) phù hợp để dễ dàng tích hợp với hệ thống front-end và mobile app.



Hình 1. 2: Hệ thống API

Thứ ba, về trách nhiệm trong nhóm: với vai trò Back-end Lead, em không chỉ làm công việc kỹ thuật mà còn tham gia điều phối nhóm back-end. em phân chia các module công việc cho các thành viên, hỗ trợ giải quyết các lỗi phát sinh trong quá trình lập trình, đồng thời kiểm tra code để đảm bảo tuân thủ chuẩn coding và bảo mật dữ liệu. Trong quá trình này, em thường xuyên trao đổi với team front-end để thống nhất giao diện API, từ đó đảm bảo sự đồng bộ giữa hai phía.

Cuối cùng, về kết quả và giá trị đạt được: thông qua việc thực hiện những nhiệm vụ trên, em đã rèn luyện được kỹ năng thiết kế và triển khai cơ sở dữ liệu phức tạp, nắm vững nguyên tắc xây dựng API RESTful, đồng thời phát triển năng lực lãnh đạo và phối hợp nhóm. Những kinh nghiệm này không chỉ giúp hệ thống website Làng nổi Tân Lập hoạt động ổn định, bảo mật và dễ dàng mở rộng, mà còn trở thành nền tảng quan trọng cho các dự án sau này của công ty.

Tóm lại, công việc và trách nhiệm em được giao trong kỳ thực tập không chỉ là những nhiệm vụ kỹ thuật thuần túy, mà còn là một quá trình rèn luyện toàn diện: từ kỹ năng lập trình, phân tích, xử lý tình huống cho đến khả năng lãnh đạo nhóm và phối

hợp trong môi trường dự án thực tế. Đây là nền tảng quý báu để em trưởng thành hơn trong vai trò lập trình viên back-end chuyên nghiệp trong tương lai.

CHƯƠNG II: CHUYÊN MÔN VÀ BÀI TOÁN ĐẶT RA

2.1 Bối cảnh vấn đề

Trong quá trình phát triển lâu dài, Công ty Cổ phần Tập đoàn HT Việt Nam đã từng xây dựng và triển khai một hệ thống phần mềm quản lý phục vụ cho các hoạt động nghiệp vụ nội bộ. Tuy nhiên, hệ thống này được phát triển cách đây từ 8–9 năm trước, trong bối cảnh công nghệ còn nhiều hạn chế, kiến trúc phần mềm chưa theo chuẩn hiện đại và dữ liệu chủ yếu được quản lý theo phương thức cục bộ, rời rạc.

Trải qua nhiều năm vận hành, hệ thống cũ dần bộc lộ những yếu điểm nghiêm trọng, thậm chí xuất hiện những lỗi nặng gây gián đoạn trong quá trình sử dụng. Điều này không chỉ làm giảm hiệu quả quản lý dữ liệu mà còn gây khó khăn cho việc tích hợp với các ứng dụng hoặc nền tảng mới của doanh nghiệp.



Hình 1. 3: Logo làng nổi tân lập

Một trong những hạn chế lớn nhất của hệ thống cũ là không có API tích hợp. Điều này dẫn đến tình trạng dữ liệu tuy có nhưng không thể dễ dàng chia sẻ giữa các bộ phận hoặc kết nối với các ứng dụng web, mobile.

Các thao tác truy xuất dữ liệu thường phức tạp, mất nhiều thời gian, đôi khi phải xử lý thủ công. Hơn nữa, kiến trúc cơ sở dữ liệu chưa được chuẩn hóa, nhiều bảng thiết kế thiếu ràng buộc chặt chẽ, gây ra sự trùng lặp và sai lệch dữ liệu. Việc bảo trì, sửa lỗi hay mở rộng thêm chức năng mới vì vậy gặp rất nhiều khó khăn.

Trong bối cảnh đó, yêu cầu đặt ra là phải xây dựng lại một hệ thống mới, có khả năng khắc phục những hạn chế nêu trên. Hệ thống mới cần đảm bảo:

Chuẩn hóa dữ liệu, hạn chế trùng lặp và nâng cao tính toàn vẹn.

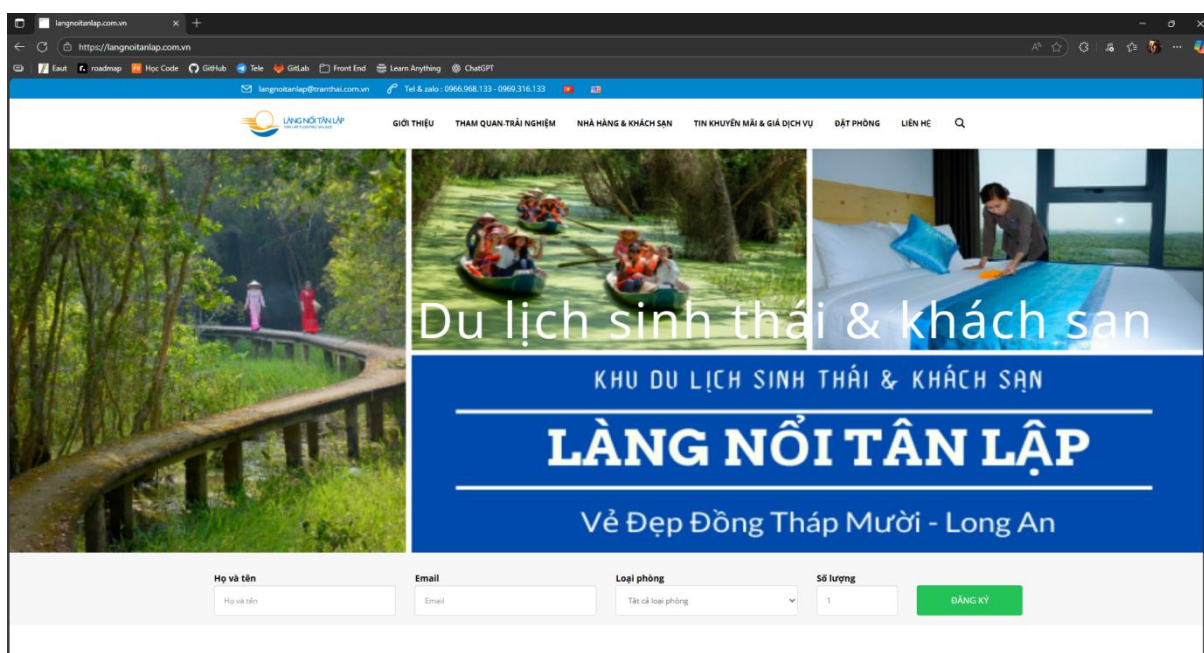
Tích hợp API hiện đại, cho phép dễ dàng kết nối với web, mobile và các dịch vụ bên thứ ba.

Khả năng mở rộng linh hoạt, đáp ứng nhu cầu gia tăng dữ liệu và chức năng trong tương lai.

Hiệu năng và bảo mật cao, phục vụ hàng nghìn người dùng cùng lúc trong lĩnh vực du lịch trực tuyến.

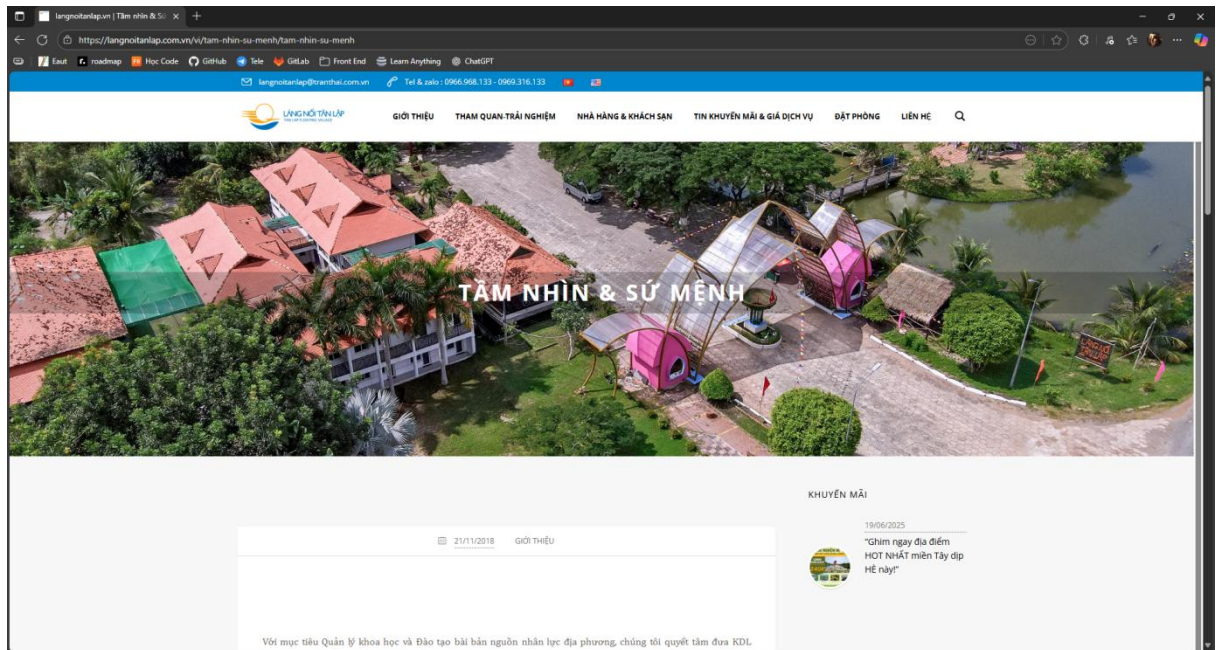
Việc phát triển hệ thống back-end mới cho website Làng nổi Tân Lập không chỉ đơn thuần là một dự án kỹ thuật, mà còn mang ý nghĩa quan trọng về mặt chiến lược. Nó giúp doanh nghiệp nâng cao hiệu quả quản lý dữ liệu, cải thiện trải nghiệm người dùng, đồng thời tạo nền tảng để mở rộng quy mô dịch vụ sang mobile app và tích hợp với các đối tác bên ngoài.

Đây là bước đi cần thiết để HT Việt Nam theo kịp xu thế chuyển đổi số và duy trì lợi thế cạnh tranh trong bối cảnh công nghệ thay đổi nhanh chóng hiện nay.



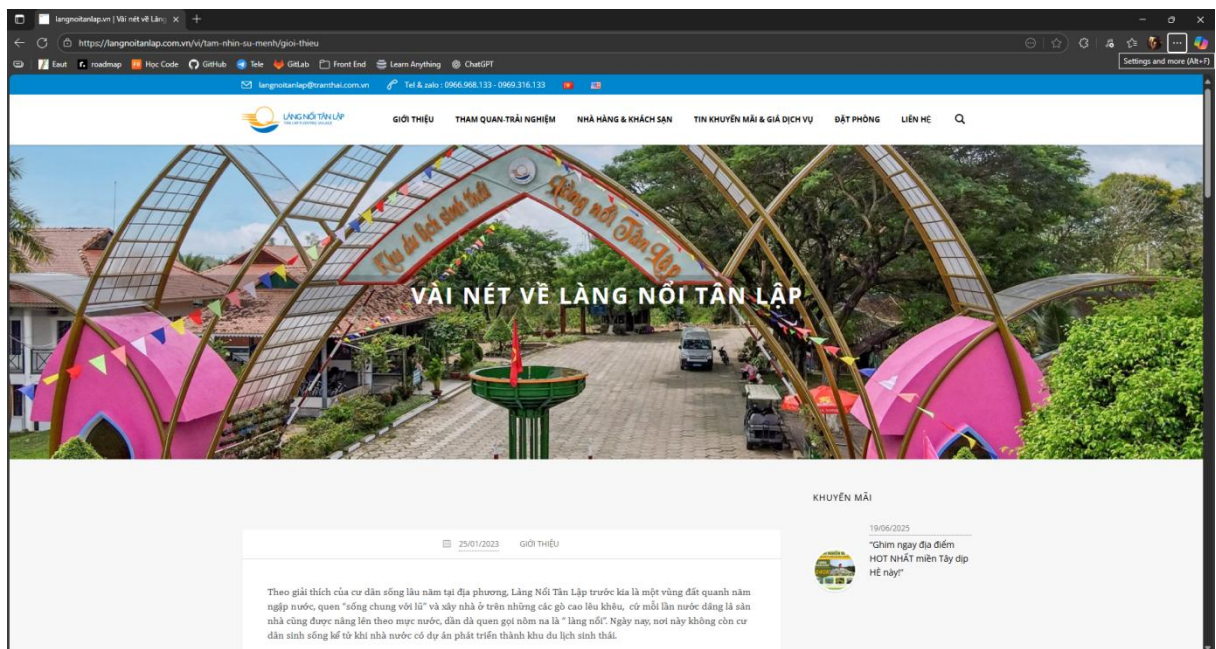
Hình 2. 1: Trang chủ website

Làng Nổi Tân Lập hướng đến mục tiêu trở thành điểm du lịch sinh thái hàng đầu miền Tây, kết hợp hài hòa giữa bảo tồn thiên nhiên và phát triển bền vững. Khu du lịch chú trọng đầu tư cơ sở hạ tầng, nâng cao chất lượng dịch vụ và xây dựng đội ngũ nhân sự chuyên nghiệp. Tất cả nhằm mang lại cho du khách những trải nghiệm ý nghĩa, gần gũi với thiên nhiên rừng tràm đặc trưng Nam Bộ.



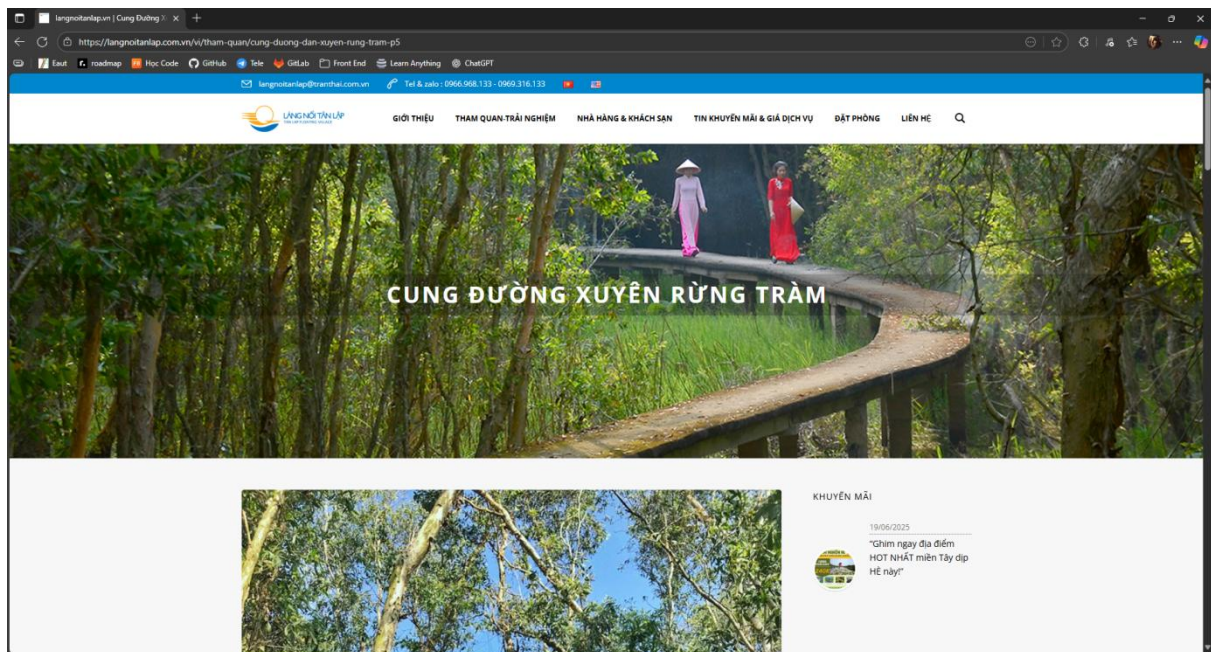
Hình 2. 2: Tầm nhìn & Sứ mệnh Làng Nổi Tân Lập

Làng Nội Tân Lập vốn là một làng nổi đặc trưng của miền Tây, nơi người dân quen "sống chung với lũ" và nhà cửa được nâng lên theo mực nước lũ hằng năm. Ngày nay, làng không còn dân cư sinh sống mà đã được quy hoạch trở thành một khu du lịch sinh thái hấp dẫn.



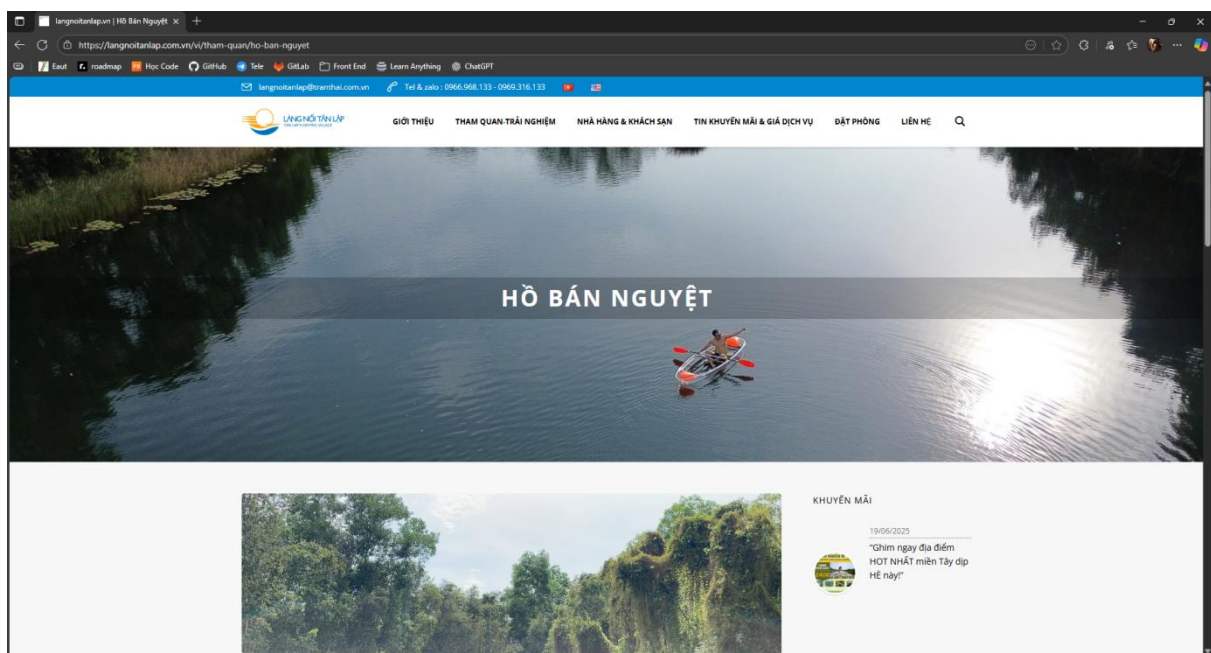
Hình 2. 3: Vài nét về làng nổi tân lập

Đây chính là điểm đến lý tưởng để trải nghiệm và khám phá nét văn hóa độc đáo của vùng sông nước.



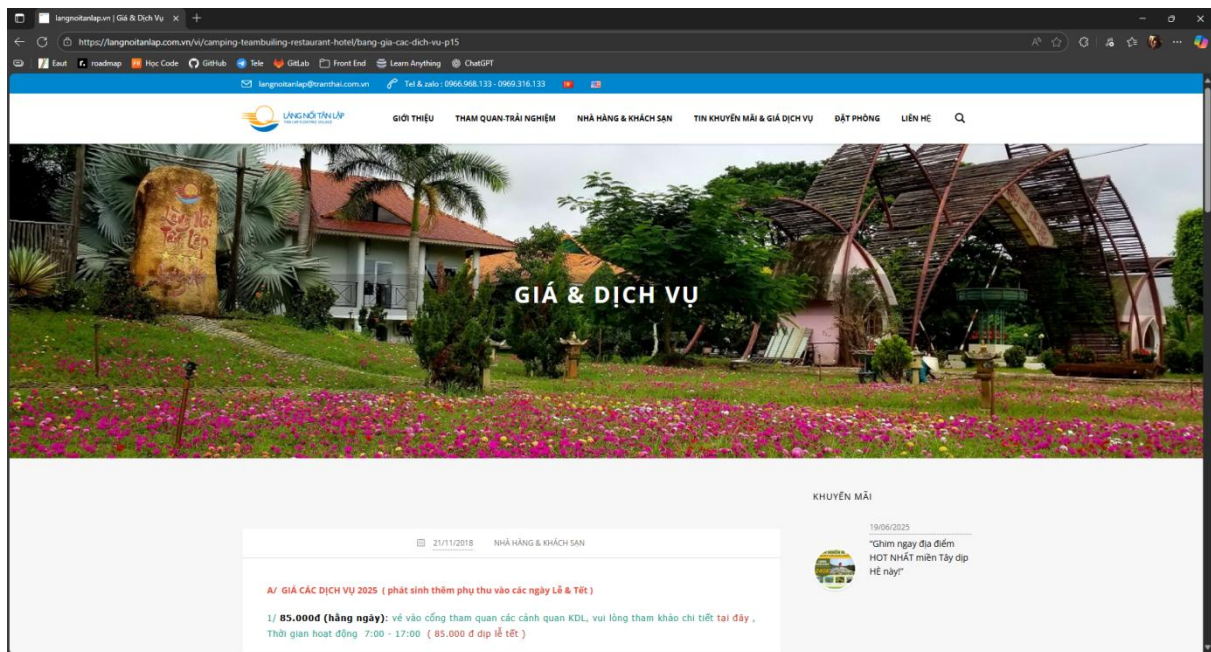
Hình 2. 4: Cung đường xuyên rừng tràm

Đến với Cung Đường Xuyên Rừng Tràm, du khách sẽ có cơ hội khám phá một trong những địa điểm HOT NHẤT miền Tây. Nơi đây mang đến một hành trình xuyên qua khu rừng tràm nguyên sinh với hệ sinh thái độc đáo.



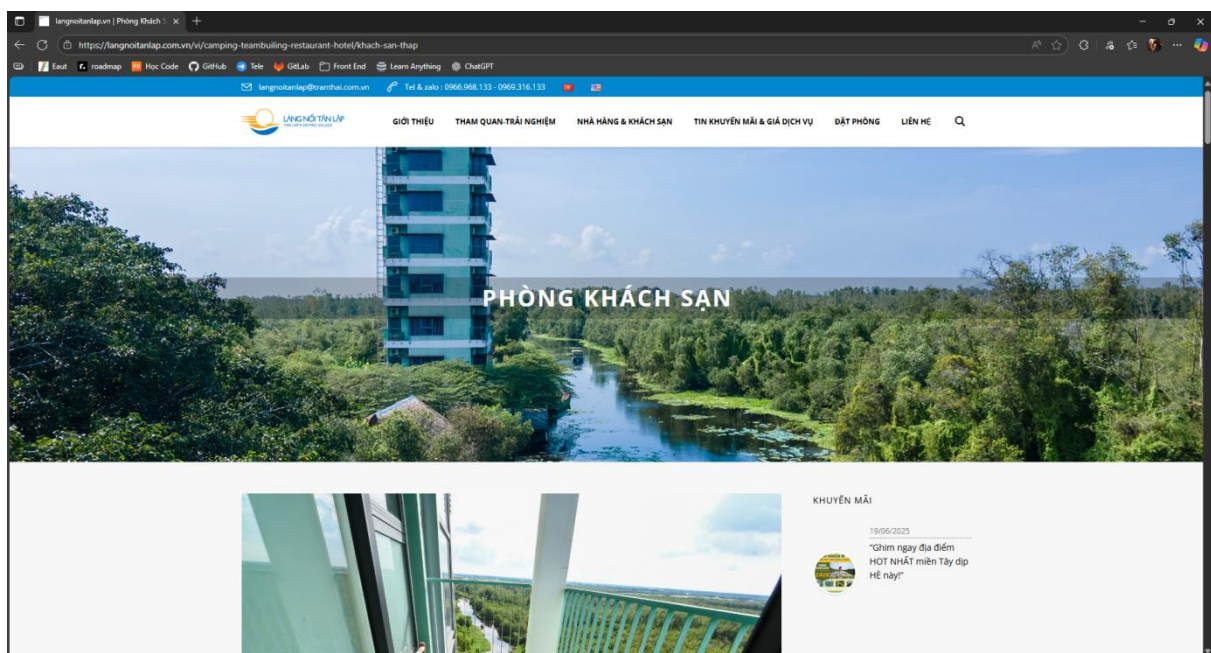
Hình 2. 5: Hồ bán nguyệt

Hồ Bán Nguyệt là một trong những địa điểm "HOT NHẤT miền Tây" để check-in và tận hưởng khung cảnh thiên nhiên yên bình. Đây là điểm đến lý tưởng cho một chuyến du lịch nghỉ dưỡng trọn vẹn.



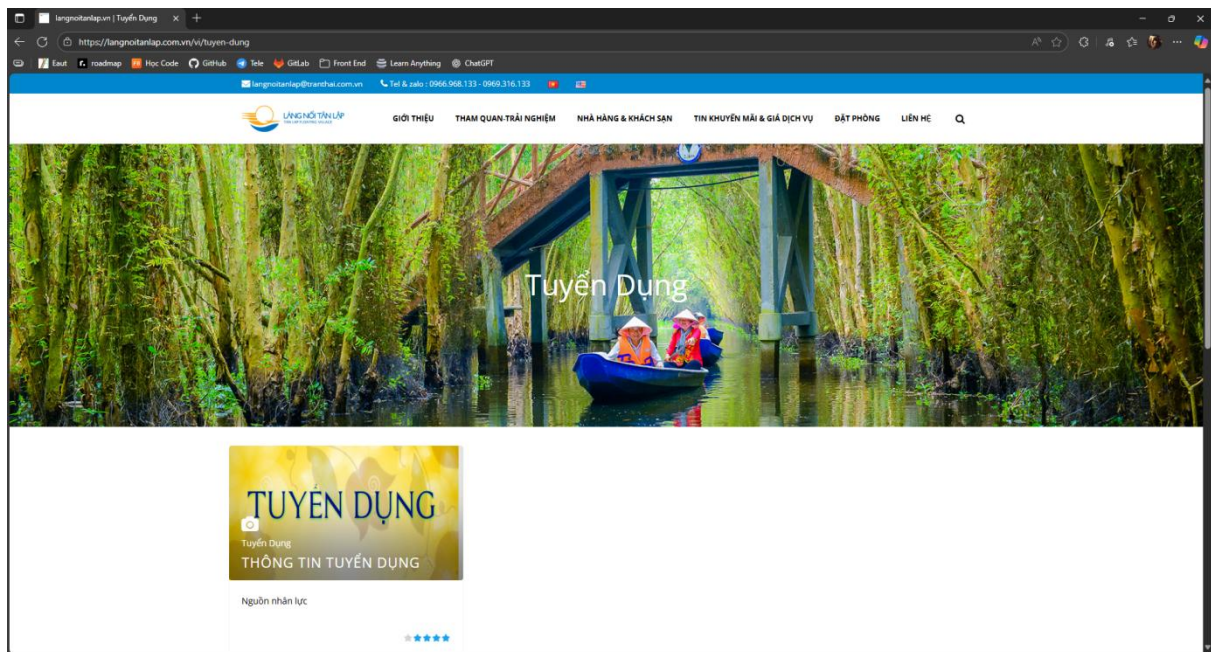
Hình 2. 6: Giá và dịch vụ theo mùa

Khám phá ngay điểm đến HOT nhất miền Tây dịp hè này với mức giá ưu đãi. Vé vào cửa bao gồm tham quan toàn bộ các cảnh quan đẹp mắt, lý tưởng cho những chuyến check-in và trải nghiệm thiên nhiên.



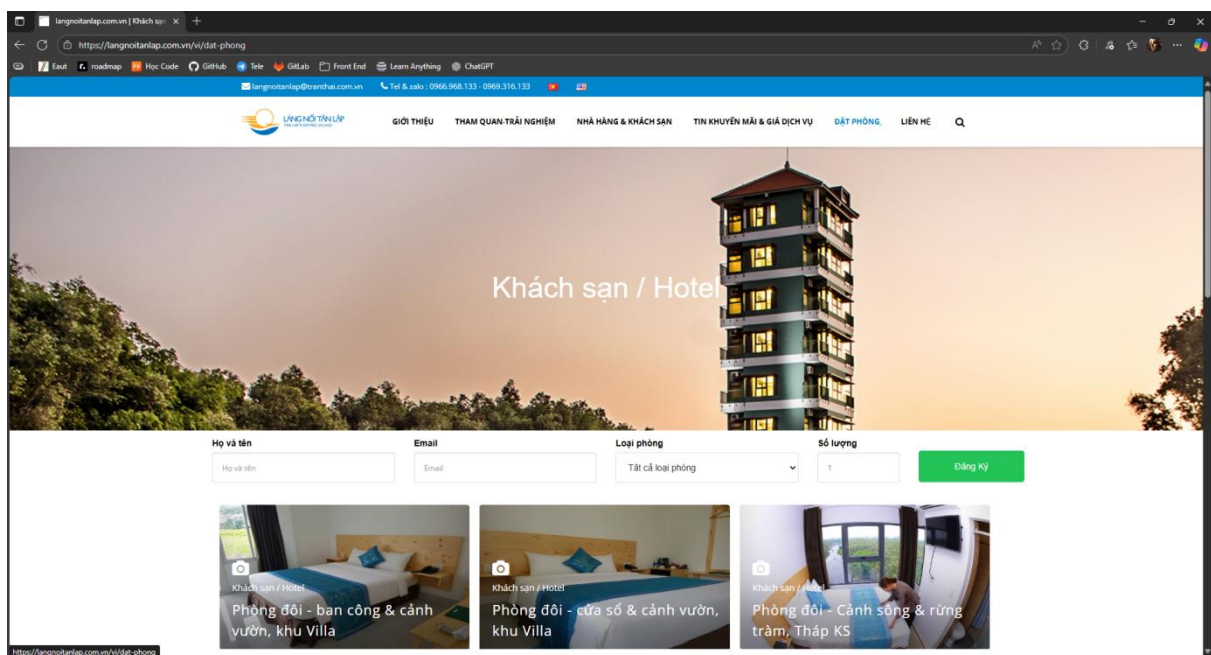
Hình 2. 7: Phòng khách sạn

Trải nghiệm không gian nghỉ dưỡng tuyệt vời tại Phòng Khách Sạn - một trong những địa điểm HOT nhất miền Tây dịp hè này.



Hình 2. 8: Thông báo tuyển dụng

Thông báo tuyển dụng nhân sự mới để mở rộng đội ngũ. Chúng tôi tìm kiếm những cá nhân nhiệt huyết, năng động và mong muốn gắn bó lâu dài. Hãy gia nhập để cùng chúng tôi phát triển và viết nên những câu chuyện thành công mới.



Hình 2. 9: Đặt phòng khách sạn

Đặt phòng khách sạn dễ dàng với các lựa chọn đa dạng từ phòng đôi view vườn, view sông đến khu Villa riêng biệt. Chỉ cần điền đầy đủ thông tin cá nhân và loại phòng bạn muốn đăng ký. Hệ thống sẽ liên hệ lại ngay để xác nhận đặt phòng của bạn.

Tóm lại, từ thực trạng lỗi thời và hạn chế của hệ thống cũ, dự án phát triển hệ thống back-end mới đã ra đời như một giải pháp tất yếu. Nó vừa giải quyết được những vấn đề tồn tại nhiều năm qua, vừa tạo ra một nền tảng công nghệ vững chắc, đóng góp quan trọng vào sự phát triển bền vững của doanh nghiệp trong giai đoạn mới.

2.2 Phân tích nhu cầu hệ thống

2.2.1 Danh sách các chức năng chính

Để đáp ứng được yêu cầu vận hành của hệ thống website Làng nổi Tân Lập, nhóm back-end đã phân tích và xác định các nhóm chức năng cốt lõi cần xây dựng. Trước hết, hệ thống phải có khả năng quản lý người dùng và phân quyền truy cập, đảm bảo mỗi tài khoản đăng ký đều được xác thực và gán đúng vai trò (admin, nhân viên, biên tập viên, khách hàng). Nhóm chức năng này bao gồm đăng ký, đăng nhập, đổi mật khẩu, quản lý tài khoản (users) và quản lý vai trò (roles). Đây là nền tảng để duy trì sự an toàn và minh bạch trong toàn bộ hệ thống.

Tiếp theo, hệ thống cần có nhóm quản lý nội dung (Content Management System – CMS) nhằm phục vụ cho việc đăng tải, chỉnh sửa và tổ chức thông tin trên website. Nhóm chức năng này bao gồm: quản lý bài viết (posts), quản lý danh mục (categories) để phân loại thông tin, quản lý tệp đa phương tiện (media) dùng cho hình ảnh và video, và quản lý slider (sliders) để hiển thị banner, quảng bá dịch vụ, sự kiện. Đây là các chức năng đặc biệt quan trọng vì chúng trực tiếp ảnh hưởng đến trải nghiệm người dùng cuối và khả năng truyền tải thông tin của doanh nghiệp.

Bên cạnh đó, trong phạm vi dự án du lịch trực tuyến, hệ thống còn tích hợp thêm nhóm quản lý khách hàng và dịch vụ đặt chỗ. Cụ thể là các chức năng liên quan đến khách hàng (customers), khách sạn và loại phòng (hotels, room_types), cũng như quản lý đơn đặt phòng (bookings/orders). Nhờ các chức năng này, hệ thống có thể cung cấp trải nghiệm trọn vẹn cho khách hàng, từ khâu tìm hiểu thông tin đến khâu đặt dịch vụ và quản lý giao dịch.

Tóm lại, danh sách các chức năng chính của hệ thống được chia thành ba nhóm lớn: quản lý người dùng và phân quyền, quản lý nội dung, và quản lý khách hàng – dịch vụ du lịch. Sự kết hợp của ba nhóm chức năng này giúp hệ thống vận hành trơn

tru, đồng thời tạo ra một nền tảng mở rộng linh hoạt, dễ dàng tích hợp thêm các dịch vụ và tính năng mới trong tương lai.

2.2.2 Mô tả chi tiết từng chức năng

Sau khi xác định được các nhóm chức năng cốt lõi, hệ thống được phân tích chi tiết để mô tả cách vận hành của từng thành phần. Mỗi chức năng được thiết kế dựa trên mô hình API chuẩn RESTful, với đầu vào (input), quá trình xử lý (process) và kết quả trả về (output) rõ ràng, đảm bảo tính thống nhất và dễ dàng tích hợp.

➤ Auth (đăng nhập, đăng ký, đổi mật khẩu)

Input: thông tin đăng ký gồm {username, email, password}, hoặc thông tin đăng nhập {username/email, password}. Với tính năng đổi mật khẩu, input bao gồm {old_password, new_password}.

Process: hệ thống kiểm tra dữ liệu người dùng trong bảng users, áp dụng thuật toán băm (bcrypt) để so khớp mật khẩu. Nếu hợp lệ, tạo token JWT để xác thực phiên làm việc.

Output: JSON trả về {status, message, token}, trong đó token được dùng để gọi các API yêu cầu đăng nhập.

➤ Users (quản lý người dùng)

Input: {username, email, phone, role_id, status}.

Process: ghi dữ liệu vào bảng users, liên kết với bảng roles để xác định quyền hạn. Ngoài ra, hệ thống hỗ trợ tìm kiếm, phân trang để quản lý tập trung số lượng lớn người dùng.

Output: JSON trả về {status, message, user_data}, trong đó có thông tin chi tiết của người dùng vừa thêm/sửa/xóa.

➤ Roles (quản lý phân quyền)

Input: {role_name, description}.

Process: lưu role mới vào bảng roles, sau đó có thể gán role cho user thông qua role_id. Các vai trò mặc định gồm admin, nhân viên, biên tập viên, khách hàng.

Output: JSON {status, message, role}, hiển thị thông tin về vai trò mới được tạo hoặc cập nhật.

➤ Posts (quản lý bài viết)

Input: {name, slug}.

Process: lưu dữ liệu vào bảng categories, ràng buộc duy nhất để tránh trùng tên/slug. Mỗi danh mục có thể chứa nhiều bài viết.

Output: JSON {status, message, category}.

➤ Categories (quản lý danh mục)

Input: { name, slug }.

Process: thêm/sửa/xóa trong bảng categories, liên kết với bảng posts.

Output: JSON { status, message, category }.

➤ Media (quản lý tệp đa phương tiện)

Input: {file_name, url, created_by}.

Process: upload file lên hệ thống hoặc CDN, lưu metadata vào bảng media, liên kết với users để biết ai là người tải lên.

Output: JSON {status, message, media_url}.

➤ Sliders (quản lý banner, slideshow)

Input: {title, link_url, media_id, post_id, is_show}.

Process: thêm/sửa dữ liệu trong bảng sliders, cho phép liên kết với bài viết hoặc media, hiển thị ngoài giao diện trang chủ.

Output: JSON {status, message, slider}.

➤ Orders/Bookings (quản lý đơn đặt dịch vụ)

Input: { customer_id, hotel_id, room_type_id, checkin_date, checkout_date, adults, children }.

Process: kiểm tra phòng còn trống (room_inventory), tạo đơn hàng trong bookings.

Output: JSON { status, message, booking_code }.

Thông qua việc phân tích chi tiết từng chức năng với input, process, output, hệ thống đã được định hình một cách rõ ràng. Các API không chỉ đảm bảo khả năng thao tác dữ liệu mà còn phục vụ nhu cầu mở rộng, tích hợp và bảo mật trong tương lai.

2.2.3 Các trường hợp sử dụng (use case)

Bảng 2. 1: Use case thanh toán

Use case: Thanh toán	ID: UC01
Tác nhân chính: Khách hàng	
Điều kiện tiên quyết: Hệ thống hoạt động bình thường.	
Điều kiện thực hiện: <ul style="list-style-type: none"> ➤ Khách hàng đã đăng ký tài khoản và có thông tin đăng nhập hợp lệ. ➤ Thiết bị của khách hàng có kết nối Internet ổn định. 	
Sự kiện kích hoạt: Khách hàng nhấn nút “Thanh toán” trong giao diện đơn hàng.	
Luồng sự kiện chính: <p>B1: Khách hàng đăng nhập vào hệ thống. Nếu thông tin không hợp lệ → thực hiện luồng phụ A1.</p> <p>B2: Khách hàng chọn đơn hàng cần thanh toán và nhấn nút “Thanh toán”.</p> <p>Hệ thống kiểm tra số dư trong tài khoản thanh toán.</p> <p>Nếu đủ: cập nhật trạng thái đơn hàng thành “Đã thanh toán”, ghi log giao dịch, và chuyển khách hàng sang giao diện “Lịch sử mua hàng”.</p> <p>Nếu không đủ: thực hiện luồng phụ A2.</p> <p>Hệ thống ghi nhận giao dịch và lưu thông tin thanh toán vào cơ sở dữ liệu.</p> <p>Hệ thống hiển thị lịch sử mua hàng của khách hàng kèm trạng thái mới.</p>	
Luồng sự kiện phụ: <ul style="list-style-type: none"> ➤ A1 – Đăng nhập sai: 	

Hệ thống thông báo thông tin đăng nhập không hợp lệ.

Hệ thống yêu cầu khách hàng nhập lại thông tin đăng nhập.

➤ A2 – Số dư tài khoản không đủ:

Hệ thống hiển thị thông báo “Số dư không đủ để thực hiện giao dịch”.

Hệ thống gợi ý khách hàng nạp tiền hoặc chọn phương thức thanh toán khác.

➤ A3 – Lỗi kết nối mạng:

Hệ thống thông báo “Mất kết nối. Vui lòng thử lại”.

Giao dịch chưa được xử lý và vẫn ở trạng thái chờ.

Bảng 2. 2: Use case đăng nhập

Use case: Đăng nhập	ID: UC02
Tác nhân chính: Người dùng (khách hàng hoặc quản trị viên)	
Điều kiện tiên quyết: Hệ thống hoạt động bình thường.	
Điều kiện thực hiện: <ul style="list-style-type: none">➤ Người dùng đã có tài khoản hợp lệ trong hệ thống.➤ Thiết bị của khách hàng có kết nối Internet ổn định.	
Sự kiện kích hoạt: Người dùng nhập thông tin đăng nhập (username/email và mật khẩu) và nhấn nút “Đăng nhập”.	
Luồng sự kiện chính: <p>B1: Người dùng mở giao diện đăng nhập và nhập {username/email, password}.</p> <p>Hệ thống kiểm tra thông tin đăng nhập trong bảng users.</p> <p>Nếu thông tin hợp lệ → tạo JWT token phiên làm việc và lưu lại thời gian đăng nhập cuối.</p> <p>Hệ thống chuyển người dùng đến giao diện chính (dashboard hoặc trang chủ).</p>	

Luồng sự kiện phụ:

- A1 – Thông tin sai: Hiện thị thông báo “Tên đăng nhập hoặc mật khẩu không đúng”, yêu cầu nhập lại.
- A2 – Tài khoản bị khóa: Hiện thị thông báo “Tài khoản đã bị khóa/bị cấm”. Người dùng không thể đăng nhập.

Bảng 2. 3: Use case đặt phòng

Use case: Đặt phòng	ID: UC03
Tác nhân chính: Khách hàng	
Điều kiện tiên quyết: Hệ thống hoạt động bình thường.	
Điều kiện thực hiện: <ul style="list-style-type: none">➤ Khách hàng đã đăng nhập vào hệ thống.➤ Dữ liệu phòng (room_types, room_inventory) đã có trong cơ sở dữ liệu.	
Sự kiện kích hoạt: Khách hàng chọn khách sạn/phòng và nhấn nút “Đặt phòng”.	
Luồng sự kiện chính: <p>B1: Khách hàng đăng nhập và chọn khách sạn, loại phòng mong muốn.</p> <p>Hệ thống hiển thị thông tin chi tiết (giá, số lượng phòng trống, ngày check-in/check-out).</p> <p>B2: Khách hàng nhập dữ liệu đặt phòng {customer_id, hotel_id, room_type_id, checkin_date, checkout_date, adults, children}.</p> <p>Hệ thống kiểm tra tồn kho (room_inventory).</p> <p>Nếu còn phòng → tạo đơn đặt phòng trong bảng bookings với trạng thái ban đầu là “pending”.</p> <p>Hệ thống trả về mã booking_code duy nhất và hiển thị thông báo “Đặt phòng thành</p>	

công”.

Luồng sự kiện phụ:

- A1 – Không còn phòng: Hệ thống hiển thị “Hết phòng cho ngày bạn chọn, vui lòng chọn ngày khác”.
- A2 – Dữ liệu nhập không hợp lệ: Hệ thống yêu cầu nhập lại thông tin (ví dụ: ngày check-out phải sau ngày check-in).

Bảng 2. 4: Use case đổi mật khẩu

Use case: Đổi mật khẩu	ID: UC04
Tác nhân chính: Người dùng (Khách hàng hoặc Quản trị viên)	
Điều kiện tiên quyết: <ul style="list-style-type: none">➤ Hệ thống hoạt động bình thường.➤ Người dùng đã đăng nhập thành công.	
Điều kiện thực hiện: <ul style="list-style-type: none">➤ Người dùng có tài khoản hợp lệ.➤ Cung cấp mật khẩu cũ đúng và nhập mật khẩu mới theo yêu cầu (đủ độ dài, ký tự đặc biệt nếu có).	
Sự kiện kích hoạt: Người dùng nhấn nút “Đổi mật khẩu” trong giao diện tài khoản.	
Luồng sự kiện chính: <p>B1: Người dùng đăng nhập vào hệ thống.</p> <p>Truy cập giao diện cài đặt tài khoản và chọn “Đổi mật khẩu”.</p> <p>B2: Nhập {old_password, new_password, confirm_password}.</p> <p>Hệ thống kiểm tra old_password với cơ sở dữ liệu.</p>	

Nếu đúng, hệ thống hash mật khẩu mới, cập nhật vào bảng users. Hệ thống hiển thị thông báo “Đổi mật khẩu thành công”.
Luồng sự kiện phụ: <ul style="list-style-type: none"> ➤ A1 – Mật khẩu cũ không đúng: Hệ thống thông báo lỗi, yêu cầu nhập lại. ➤ A2 – Mật khẩu mới không hợp lệ: Hiển thị “Mật khẩu phải có ít nhất 8 ký tự và chứa ký tự đặc biệt”. ➤ A3 – Xác nhận mật khẩu không khớp: Yêu cầu nhập lại confirm_password.

Bảng 2. 5: Use case thêm hình ảnh vào Media

Use case: Thêm Media	ID: UC05
Tác nhân chính: Quản trị viên / Biên tập viên nội dung	
Điều kiện tiên quyết: Hệ thống hoạt động bình thường.	
Điều kiện thực hiện: <ul style="list-style-type: none"> ➤ Người dùng đã đăng nhập và có quyền upload media. ➤ File tải lên hợp lệ (jpg, png, mp4, dung lượng < 5MB). 	
Sự kiện kích hoạt: Người dùng nhấn nút “Tải lên Media” trong giao diện quản trị.	
Luồng sự kiện chính: B1: Người dùng chọn file hình ảnh/video từ máy tính. Hệ thống kiểm tra định dạng và dung lượng file. Nếu hợp lệ → lưu file vào server/CDN, tạo bản ghi trong bảng media. Hệ thống trả về thông tin JSON {id, file_name, url}. Hệ thống hiển thị thông báo “Upload thành công”.	
Luồng sự kiện phụ:	

- A1 – File không hợp lệ: Hệ thống hiển thị “Chỉ chấp nhận jpg/png/mp4 và dung lượng < 5MB”.
- A2 – Mất kết nối khi tải lên: Hệ thống lưu trạng thái pending và yêu cầu thử lại.

Bảng 2. 6: Use case quản lý Slider (Banner)

Use case: Quản lý Slider	ID: UC06
Tác nhân chính: Quản trị viên / Nhân viên Marketing	
Điều kiện tiên quyết: Hệ thống hoạt động bình thường.	
Điều kiện thực hiện: <ul style="list-style-type: none"> ➤ Người dùng đã đăng nhập và có quyền quản lý slider. ➤ Media (ảnh) tồn tại trong hệ thống. 	
Sự kiện kích hoạt: Người dùng nhấn nút “Thêm Slider” trên giao diện quản trị.	
Luồng sự kiện chính: <p>B1: Người dùng truy cập giao diện quản lý slider.</p> <p>B2: Nhấn “Thêm mới” và nhập thông tin {title, link_url, media_id, post_id, is_show}.</p> <p>Hệ thống kiểm tra dữ liệu đầu vào.</p> <p>Lưu vào bảng sliders, liên kết với media và posts nếu có.</p> <p>Hiển thị thông báo “Slider đã được thêm thành công”.</p>	
Luồng sự kiện phụ: <ul style="list-style-type: none"> ➤ A1 – Media không tồn tại: Hệ thống hiển thị “Ảnh không hợp lệ”. ➤ A2 – Link URL không hợp lệ: Yêu cầu nhập lại link đúng định dạng. 	

Bảng 2. 7: Use case quản lý Danh mục (Categories)

Use case: Quản lý Danh mục	ID: UC07
Tác nhân chính: Quản trị viên	
Điều kiện tiên quyết: Hệ thống hoạt động bình thường.	
Điều kiện thực hiện: <ul style="list-style-type: none"> ➤ Người dùng đã đăng nhập và có quyền quản lý nội dung. ➤ Danh mục phải có tên và slug hợp lệ. 	
Sự kiện kích hoạt: Người dùng nhấn nút “Thêm danh mục” trên giao diện quản trị.	
Luồng sự kiện chính: <p>B1: Người dùng truy cập giao diện quản lý danh mục.</p> <p>B2: Nhấn “Thêm mới” và nhập thông tin {name, slug}.</p> <p>Hệ thống kiểm tra tên và slug có trùng lặp không.</p> <p>Nếu hợp lệ → lưu vào bảng categories.</p> <p>Hiển thị thông báo “Thêm danh mục thành công”.</p>	
Luồng sự kiện phụ: <ul style="list-style-type: none"> ➤ A1 – Trùng slug: Hệ thống báo lỗi “Slug đã tồn tại”. ➤ A2 – Thiếu thông tin: Hệ thống hiển thị “Tên danh mục không được để trống”. 	

2.3 Các yêu cầu bổ trợ (phi chức năng)

Bên cạnh các chức năng chính được thiết kế, một hệ thống back-end hiện đại không thể thiếu các yêu cầu bổ trợ về hiệu suất, bảo mật và khả năng bảo trì. Những yếu tố này không trực tiếp mang lại tính năng cho người dùng cuối nhưng lại quyết định sự ổn định, độ tin cậy và tuổi thọ của hệ thống trong suốt quá trình vận hành.

2.3.1 Hiệu suất và khả năng mở rộng

Hiệu suất là một trong những yếu tố quan trọng nhất, đặc biệt đối với một hệ thống web du lịch trực tuyến có số lượng người truy cập lớn và thường xuyên. Hệ thống cần đảm bảo API phản hồi dưới 1 giây trong điều kiện tải trung bình và có thể chịu tải tối thiểu 1.000 request/giây khi có nhiều người dùng truy cập đồng thời.

Để đạt được điều này, cơ sở dữ liệu được thiết kế chuẩn hóa, có chỉ mục (index) hợp lý và tận dụng các cơ chế cache (Redis, Memcached) để giảm tải cho DB. Ngoài ra, hệ thống cần có khả năng mở rộng theo chiều ngang (horizontal scaling) thông qua việc triển khai nhiều server back-end song song, cân bằng tải bằng load balancer. Đây là cơ sở để hệ thống có thể mở rộng trong tương lai khi số lượng người dùng và dữ liệu tăng lên nhanh chóng.

Tóm lại, yêu cầu về hiệu suất và khả năng mở rộng giúp hệ thống không chỉ đáp ứng tốt nhu cầu hiện tại mà còn sẵn sàng phục vụ trong giai đoạn phát triển dài hạn.

2.3.2 Bảo mật và an toàn dữ liệu

Trong lĩnh vực dịch vụ trực tuyến, bảo mật dữ liệu là yếu tố sống còn. Hệ thống cần áp dụng cơ chế xác thực hiện đại bằng JWT để đảm bảo mỗi yêu cầu API đều có kiểm soát quyền truy cập. Bên cạnh đó, việc phân quyền người dùng dựa trên bảng roles và users giúp hạn chế các hành vi truy cập trái phép, chỉ cho phép đúng người thực hiện đúng chức năng.

Ngoài xác thực, dữ liệu nhạy cảm (mật khẩu, token) phải được mã hóa bằng bcrypt hoặc thuật toán hashing an toàn. Các kết nối giữa client – server phải được bảo vệ bằng giao thức HTTPS để tránh nghe lén dữ liệu. Đồng thời, hệ thống cần có cơ chế chống tấn công phổ biến như SQL Injection, XSS, CSRF, cũng như giới hạn tốc độ request để ngăn chặn brute-force.

Như vậy, yêu cầu bảo mật và an toàn dữ liệu đảm bảo rằng hệ thống không chỉ vận hành ổn định mà còn giữ được niềm tin của khách hàng và doanh nghiệp.

2.3.3 Tính ổn định và khả năng bảo trì

Một hệ thống được xây dựng tốt không chỉ cần chạy nhanh và an toàn, mà còn phải dễ dàng duy trì và phát triển về sau. Cấu trúc dự án back-end được tổ chức theo mô hình RESTful chuẩn, phân chia rõ ràng giữa các module (auth, users, roles, posts,

media, bookings). Nhờ vậy, việc thêm hoặc sửa một module không làm ảnh hưởng đến toàn bộ hệ thống.

Ngoài ra, việc áp dụng coding convention chuẩn (ví dụ: PSR cho PHP, ESLint/Prettier cho Node.js) và sử dụng hệ thống quản lý phiên bản (Git) giúp nhóm phát triển dễ dàng cộng tác, kiểm soát thay đổi và rollback khi cần. Cơ chế logging (ghi log) cũng được triển khai để theo dõi lỗi và tình trạng hoạt động, phục vụ cho việc bảo trì định kỳ.

Tính ổn định còn thể hiện ở khả năng phục hồi khi xảy ra sự cố. Hệ thống cần có cơ chế backup dữ liệu định kỳ, sẵn sàng khôi phục khi gặp lỗi nghiêm trọng. Bằng cách này, dù có sự cố, hệ thống vẫn đảm bảo hoạt động liên tục và hạn chế tối đa gián đoạn dịch vụ.

Các yêu cầu phi chức năng (hiệu suất, bảo mật, ổn định) đóng vai trò nền móng cho toàn bộ dự án. Nếu hệ thống chỉ đáp ứng chức năng mà không có các yếu tố hỗ trợ này, nó sẽ nhanh chóng bộc lộ điểm yếu và không thể duy trì lâu dài. Ngược lại, khi được thiết kế bài bản ngay từ đầu, hệ thống sẽ có khả năng vận hành bền vững, dễ mở rộng và sẵn sàng đáp ứng nhu cầu ngày càng cao của người dùng.

2.4 Hướng tiếp cận và phương án xử lý

Trong quá trình xây dựng hệ thống back-end cho website Làng nổi Tân Lập, nhóm phát triển đã áp dụng phương pháp tiếp cận hiện đại dựa trên công nghệ web tiên tiến, đồng thời xây dựng quy trình làm việc khoa học nhằm đảm bảo tiến độ và chất lượng sản phẩm. Phần này trình bày rõ lý do lựa chọn công nghệ, các bước triển khai, những khó khăn gặp phải và cách khắc phục, cùng những bài học thực tế mà tôi đã tích lũy được trong vai trò Back-end Lead.

2.4.1 Lý do lựa chọn công nghệ

Để hệ thống đạt được sự ổn định, bảo mật và dễ mở rộng, nhóm đã quyết định lựa chọn MySQL cho quản lý cơ sở dữ liệu, kết hợp với NestJS cho xây dựng API back-end và NextJS cho phát triển front-end.

MySQL: Là hệ quản trị cơ sở dữ liệu quan hệ phổ biến, có khả năng xử lý nhanh, hỗ trợ chuẩn SQL, dễ dàng quản lý các quan hệ phức tạp giữa bảng như users – roles, posts – categories, bookings – room_types. Ngoài ra, MySQL có cộng đồng hỗ trợ rộng lớn, phù hợp cho cả môi trường phát triển và triển khai thực tế.

NestJS: Là framework back-end trên nền Node.js, áp dụng kiến trúc module và decorator, giúp việc xây dựng API trở nên rõ ràng, dễ bảo trì. NestJS hỗ trợ TypeScript, cho phép kiểm tra kiểu tĩnh, giảm thiểu lỗi runtime.

NextJS: Được lựa chọn cho phần front-end nhờ khả năng kết hợp SSR và SSG, giúp tối ưu SEO và tăng tốc độ tải trang. NextJS cũng tích hợp dễ dàng với API back-end thông qua RESTful hoặc GraphQL.

Sự kết hợp của ba công nghệ này tạo nên một hệ thống linh hoạt, dễ mở rộng, đồng thời bảo đảm hiệu suất và trải nghiệm người dùng.

2.4.2 Quy trình triển khai (phân tích → thiết kế → code → test)

Quy trình triển khai được thực hiện tuần tự theo các giai đoạn:

Phân tích yêu cầu: Tiếp nhận thông tin từ doanh nghiệp, xác định các chức năng chính (auth, users, roles, posts, categories, media, sliders, bookings).

Thiết kế ERD (Entity Relationship Diagram): Vẽ sơ đồ quan hệ giữa các bảng, chuẩn hóa dữ liệu, xác định khóa chính, khóa ngoại và các ràng buộc.

Phát triển API (Code API): Sử dụng NestJS để viết các API RESTful, đảm bảo mỗi API có đầu vào/đầu ra rõ ràng, dễ dàng tích hợp với front-end.

Kiểm thử (Test API): Sử dụng Postman để test từng API (CRUD, validate input, kiểm tra lỗi). Test cả luồng chính và luồng ngoại lệ.

Triển khai (Deploy): Hệ thống được triển khai trên môi trường staging trước, sau khi ổn định mới đưa lên production, kết hợp logging để theo dõi hoạt động.

Nhờ quy trình này, hệ thống luôn được kiểm soát tốt từ giai đoạn phân tích đến khi đưa vào sử dụng.

2.4.3 Các khó khăn gặp phải và cách khắc phục

Trong quá trình triển khai, nhóm đã gặp phải một số khó khăn tiêu biểu:

Quan hệ bảng phức tạp: Việc join nhiều bảng (ví dụ: lấy thông tin bài viết kèm danh mục, media, người tạo) gây giảm hiệu năng.

→ Cách khắc phục: Tối ưu câu lệnh SQL, tạo index cho các trường thường xuyên truy vấn, áp dụng lazy loading khi cần.

Xử lý đồng bộ dữ liệu (Concurrency): Nhiều người dùng cùng đặt phòng một loại phòng có số lượng hạn chế dẫn đến nguy cơ overbooking.

→ Cách khắc phục: Sử dụng transaction và cơ chế SELECT ... FOR UPDATE để khóa hàng trong CSDL, đảm bảo tính nhất quán.

Kiểm thử API: Một số API trả về kết quả sai do chưa validate dữ liệu đầu vào hoặc chưa xử lý lỗi đúng cách.

→ Cách khắc phục: Bổ sung middleware validate schema, áp dụng cơ chế try-catch toàn cục, ghi log chi tiết để dễ debug.

Nhờ giải quyết kịp thời, hệ thống đã duy trì được sự ổn định và đáp ứng đúng yêu cầu đề ra.

2.4.4 Kinh nghiệm và kỹ năng tích lũy được

Qua quá trình triển khai dự án, tôi đã rút ra nhiều bài học quan trọng:

Về kỹ thuật: Nắm vững nguyên tắc chuẩn hóa cơ sở dữ liệu, sử dụng thành thạo MySQL trong các truy vấn phức tạp, hiểu rõ cách xây dựng API RESTful với NestJS và cách tích hợp front-end (NextJS).

Về quản lý và teamwork: Biết cách phân chia module cho các thành viên, phối hợp giữa front-end và back-end thông qua API contract, rèn luyện kỹ năng code review để nâng cao chất lượng dự án.

Về xử lý sự cố: Biết cách phát hiện và giải quyết lỗi concurrency, lỗi API không phản hồi, và xây dựng cơ chế log để hỗ trợ bảo trì.

Về tư duy hệ thống: Hiểu rõ rằng một hệ thống tốt không chỉ cần tính năng đầy đủ mà còn phải tối ưu hiệu suất, bảo mật và khả năng mở rộng.

CHƯƠNG III: KẾT QUẢ VÀ GIẢI PHÁP

3.1 Xây dựng và thiết kế hệ thống

3.1.1 Phân tích yêu cầu từ nghiệp vụ

Xuất phát từ nhu cầu thực tế của doanh nghiệp, hệ thống website Làng nổi Tân Lập phải đáp ứng hai lớp yêu cầu: nghiệp vụ và kỹ thuật.

➤ **Yêu cầu từ nghiệp vụ:**

Hệ thống cho phép khách hàng đăng ký, đăng nhập, đặt phòng, tra cứu thông tin dịch vụ, và theo dõi lịch sử đơn hàng.

Hệ thống cho phép quản trị viên quản lý người dùng, phân quyền, duyệt hoặc hủy đơn đặt phòng, cập nhật nội dung bài viết, danh mục, slider và hình ảnh.

Hệ thống cần hỗ trợ tìm kiếm nhanh, hiển thị hình ảnh, tin tức và dịch vụ một cách trực quan, hấp dẫn.

➤ **Yêu cầu kỹ thuật:**

Cơ sở dữ liệu phải đảm bảo tính chuẩn hóa, có ràng buộc quan hệ chặt chẽ và dễ mở rộng.

API back-end cần được xây dựng theo chuẩn RESTful, rõ ràng về input, process, output.

Hệ thống phải bảo mật (xác thực JWT, phân quyền dựa trên role).

Thời gian phản hồi API trung bình dưới 1 giây và có khả năng xử lý ít nhất 1.000 request/s.

Việc phân tích kỹ lưỡng yêu cầu đã giúp định hình rõ phạm vi dự án và các module quan trọng cần triển khai, từ đó giảm thiểu rủi ro trong quá trình phát triển.

3.1.2 Kiến trúc tổng thể

Để đáp ứng yêu cầu nghiệp vụ và kỹ thuật, hệ thống được thiết kế dựa trên mô hình client-server theo kiến trúc 3 lớp (Three-tier architecture):

➤ **Lớp trình diễn (Presentation Layer):** Sử dụng NextJS cho phần front-end. Hiển thị giao diện website, cho phép khách hàng và quản trị viên tương tác và thao tác trực

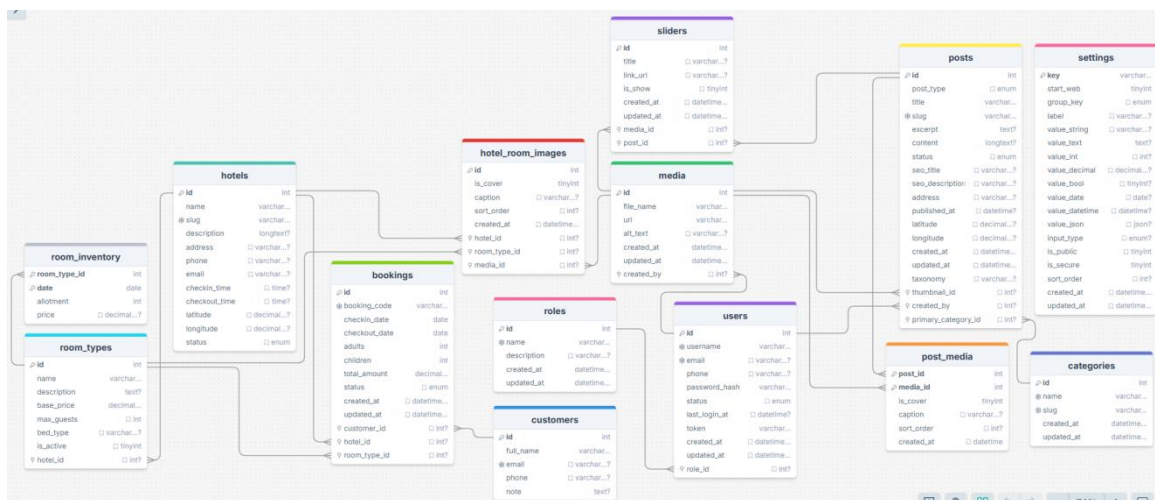
tiếp. Kết nối với back-end thông qua API RESTful.

- Lớp nghiệp vụ (Business Logic Layer): Được xây dựng bằng NestJS. Chịu trách nhiệm xử lý logic nghiệp vụ: xác thực đăng nhập, phân quyền, kiểm tra tồn kho phòng, tạo booking, duyệt bài viết. Đóng vai trò API gateway, cung cấp endpoint cho front-end và mobile app.
- Lớp dữ liệu (Data Layer): Sử dụng MySQL để lưu trữ dữ liệu. Bao gồm các bảng: users, roles, posts, categories, media, sliders, bookings, customers, hotels, room_types, room_inventory, post_media. Dữ liệu được truy vấn thông qua ORM (TypeORM) hoặc query builder.

Kiến trúc này đảm bảo tính phân tách rõ ràng giữa các tầng, dễ dàng bảo trì và mở rộng, đồng thời tối ưu hiệu suất nhờ khả năng mở rộng theo chiều ngang (horizontal scaling).

3.1.3 Thiết kế chi tiết

a) ERD (Sơ đồ quan hệ dữ liệu)



Hình 3. 1: ERD (sơ đồ quan hệ dữ liệu)

Hệ thống cơ sở dữ liệu được thiết kế với nhiều bảng có quan hệ 1-n và n-n. Một số mối quan hệ chính:

users – roles: quan hệ 1-n (một role có nhiều user).

posts – categories: quan hệ 1-n (một category có nhiều bài viết).

posts – media thông qua post_media: quan hệ n-n.

bookings – customers – hotels – room_types: quan hệ 1–n, phục vụ cho đặt phòng.

sliders liên kết với posts hoặc media để hiển thị nội dung quảng bá.

b, Cấu trúc các bảng dữ liệu chính

Bảng 3. 1: Bảng Users

Trường	Kiểu dữ liệu	Mô tả
id	int (PK)	Khóa chính, định danh duy nhất người dùng
username	varchar	Tên đăng nhập
email	varchar (unique)	Email người dùng
phone	varchar	Số điện thoại
password	varchar	Mật khẩu đã mã hóa (bcrypt)
status	enum(active,inactive,banned)	Trạng thái tài khoản
last_login_at	datetime	Thời điểm đăng nhập gần nhất
role_id	int (FK → roles.id)	Vai trò người dùng
created_at	datetime	Ngày tạo tài khoản
updated_at	datetime	Ngày cập nhật gần nhất

Bảng 3. 2: Bảng Roles

Trường	Kiểu dữ liệu	Mô tả
id	int (PK)	Khóa chính
name	varchar (unique)	Tên vai trò (admin, employee, ...)

description	varchar	Mô tả vai trò
created_at	datetime	Ngày tạo
updated_at	datetime	Ngày cập nhật

Bảng 3. 3: Bảng Posts

Trường	Kiểu dữ liệu	Mô tả
id	int (PK)	Khóa chính
post_type	enum(page,news,attraction,service)	Loại nội dung
title	varchar	Tiêu đề bài viết
slug	varchar (unique)	Đường dẫn thân thiện
excerpt	text	Tóm tắt nội dung
content	longtext	Nội dung chi tiết
status	enum(draft,published,archived)	Trạng thái xuất bản
seo_title	varchar	Tiêu đề SEO
seo_description	varchar	Mô tả SEO
address	varchar	Địa chỉ liên quan (nếu có)
published_at	datetime	Ngày xuất bản
thumbnail_id	int (FK → media.id)	Ảnh đại diện
created_by	int (FK → users.id)	Người tạo
primary_category_id	int (FK → categories.id)	Danh mục chính

created_at	datetime	Ngày tạo
updated_at	datetime	Ngày cập nhật

Bảng 3. 4: Bảng Categories

Trường	Kiểu dữ liệu	Mô tả
id	int (PK)	Khóa chính
name	varchar (unique)	Tên danh mục
slug	varchar (unique)	Đường dẫn thân thiện
created_at	datetime	Ngày tạo
updated_at	datetime	Ngày cập nhật

Bảng 3. 5: Bảng Media

Trường	Kiểu dữ liệu	Mô tả
id	int (PK)	Khóa chính
file_name	varchar	Tên file
url	longtext	Đường dẫn file (server/CDN)
created_by	int (FK → users.id)	Người tải lên
created_at	datetime	Ngày tạo
updated_at	datetime	Ngày cập nhật

Bảng 3. 6: Bảng Sliders

Trường	Kiểu dữ liệu	Mô tả
id	int (PK)	Khóa chính
title	varchar	Tiêu đề slider
link_url	varchar	Liên kết khi click
is_show	tinyint	Hiển thị (1: có, 0: ẩn)
media_id	int (FK → media.id)	Ảnh hiển thị
post_id	int (FK → posts.id)	Liên kết bài viết nếu có
created_at	datetime	Ngày tạo
updated_at	datetime	Ngày cập nhật

Bảng 3. 7: Bảng Post_media

Trường	Kiểu dữ liệu	Mô tả
post_id	int (FK → posts.id)	Bài viết liên kết
media_id	int (FK → media.id)	Media liên kết
is_cover	tinyint	Đặt làm ảnh đại diện (1: có, 0: không)
caption	varchar	Chú thích cho media
sort_order	int	Thứ tự hiển thị
created_at	datetime	Ngày tạo liên kết

c, Danh sách API endpoint (đường dẫn, method, input/output)

Bảng 3. 8: API cho Auth

Method	Endpoint	Input	Output	Mô tả
POST	/auth/register	{username, email, password}	{status, message, user_id}	Đăng ký tài khoản mới
POST	/auth/login	{username/email, password}	{status, message, token, user_info}	Đăng nhập, trả về JWT token
POST	/auth/change-password	{old_password, new_password, confirm_password}	{status, message}	Đổi mật khẩu

Bảng 3. 9: API cho Users

Method	Endpoint	Input	Output	Mô tả
GET	/users	–	Danh sách user	Lấy danh sách tất cả người dùng
GET	/users/{id}	–	Thông tin chi tiết user	Lấy thông tin 1 người dùng
POST	/users	{username, email, phone, password, role_id, status}	{status, message, user_data}	Tạo user mới
PUT	/users/{id}	{email, phone, role_id, status}	{status, message, updated_user}	Cập nhật thông tin user
DELETE	/users/{id}	–	{status, message}	Xóa user

Bảng 3. 10: API cho Roles

Method	Endpoint	Input	Output	Mô tả
GET	/roles	–	Danh sách roles	Lấy danh sách vai trò
GET	/roles/{id}	–	Thông tin chi tiết role	Lấy chi tiết vai trò
POST	/roles	{name, description}	{status, message, role_data}	Tạo vai trò mới
PUT	/roles/{id}	{name, description}	{status, message, updated_role}	Cập nhật vai trò
DELETE	/roles/{id}	–	{status, message}	Xóa vai trò

Bảng 3. 11: API cho Posts

Method	Endpoint	Input	Output	Mô tả
GET	/posts	–	Danh sách posts	Lấy danh sách bài viết
GET	/posts/{id}	–	Chi tiết 1 bài viết	Lấy chi tiết bài viết
POST	/posts	{title, content, category_id, thumbnail_id, status, excerpt}	{status, message, post_data}	Thêm bài viết mới
PUT	/posts/{id}	{title, content, category_id, thumbnail_id, status, excerpt}	{status, message, updated_post}	Cập nhật bài viết

		status, excerpt}		
DELETE	/posts/{id}	–	{status, message}	Xóa bài viết

Bảng 3. 12: API cho Categories

Method	Endpoint	Input	Output	Mô tả
GET	/categories	–	Danh sách categories	Lấy danh sách danh mục
GET	/categories/{id}	–	Chi tiết category	Lấy chi tiết 1 danh mục
POST	/categories	{name, slug}	{status, message, category_data}	Thêm danh mục mới
PUT	/categories/{id}	{name, slug}	{status, message, updated_category}	Cập nhật danh mục
DELETE	/categories/{id}	–	{status, message}	Xóa danh mục

Bảng 3. 13: API cho Media

Method	Endpoint	Input	Output	Mô tả
GET	/media	–	Danh sách media	Lấy danh sách media
GET	/media/{id}	–	Chi tiết media	Lấy chi tiết file
POST	/media	{file_name, url, created_by}	{status, message, media_data}	Upload file, thêm mới media

PUT	/media/{id}	{file_name, url}	{status, message, updated_media}	Cập nhật media
DELETE	/media/{id}	–	{status, message}	Xóa media

Bảng 3. 14: API cho Sliders

Method	Endpoint	Input	Output	Mô tả
GET	/sliders	–	Danh sách sliders	Lấy danh sách sliders
GET	/sliders/{id}	–	Chi tiết slider	Lấy chi tiết slider
POST	/sliders	{title, link_url, media_id, post_id, is_show}	{status, message, slider_data}	Thêm slider mới
PUT	/sliders/{id}	{title, link_url, media_id, post_id, is_show}	{status, message, updated_slider}	Cập nhật slider
DELETE	/sliders/{id}	–	{status, message}	Xóa slider

Bảng 3. 15: API cho Post_media

Method	Endpoint	Input	Output	Mô tả
GET	/post_media	–	Danh sách liên kết bài viết – media	Lấy danh sách

POST	/post_media	{post_id, media_id, is_cover, caption, sort_order}	{status, message, link_data}	Tạo liên kết post-media
DELETE	/post_media/{id}	–	{status, message}	Xóa liên kết post-media

3.2 Nền tảng công nghệ và giải pháp ứng dụng

Để đảm bảo hệ thống back-end của website Làng nổi Tân Lập vận hành ổn định, bảo mật và sẵn sàng mở rộng, nhóm phát triển đã lựa chọn những công nghệ và giải pháp kỹ thuật phù hợp với bối cảnh thực tế. Phần này trình bày rõ các công nghệ chủ đạo, các giải pháp kỹ thuật áp dụng và công cụ nghiên cứu hỗ trợ trong suốt quá trình phát triển.

3.2.1 Công nghệ chủ đạo được sử dụng

MySQL: Là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến, hỗ trợ chuẩn SQL, cho phép xây dựng cơ sở dữ liệu phức tạp với nhiều quan hệ. MySQL có khả năng xử lý nhanh, hỗ trợ index và transaction, đồng thời dễ triển khai trong cả môi trường cục bộ và đám mây.

NestJS: Framework back-end dựa trên Node.js, hỗ trợ TypeScript, áp dụng kiến trúc module và decorator. NestJS giúp việc phát triển API RESTful trở nên rõ ràng, dễ bảo trì, đồng thời hỗ trợ tốt cho việc tích hợp với cơ sở dữ liệu thông qua ORM (TypeORM).

NextJS: Framework front-end dựa trên React, hỗ trợ cả SSR (Server-Side Rendering) và SSG (Static Site Generation). Điều này giúp website tải nhanh, tối ưu SEO, và mang lại trải nghiệm mượt mà cho người dùng cuối.

Postman: Công cụ kiểm thử API được sử dụng để thiết kế, test, và mô phỏng các luồng nghiệp vụ. Nhờ Postman, nhóm dễ dàng phát hiện lỗi, kiểm tra input/output và viết kịch bản test tự động.

Git: Hệ thống quản lý phiên bản phân tán, giúp nhóm phát triển cộng tác hiệu quả, theo dõi lịch sử thay đổi và đảm bảo tính toàn vẹn của source code.

Nhờ sự kết hợp của các công nghệ trên, hệ thống được xây dựng theo hướng hiện đại, dễ mở rộng và có khả năng đáp ứng tốt cho cả nhu cầu hiện tại và tương lai.

3.2.2 Giải pháp kỹ thuật áp dụng

Để xử lý các yêu cầu kỹ thuật phức tạp, đảm bảo tính ổn định và hiệu năng, nhóm đã áp dụng nhiều giải pháp quan trọng.

Transaction và FOR UPDATE: Các luồng nghiệp vụ quan trọng (như đặt phòng, thanh toán) được thực hiện trong transaction để đảm bảo tính toàn vẹn dữ liệu. Khi nhiều người cùng đặt một loại phòng, lệnh SELECT ... FOR UPDATE được sử dụng để khóa hàng, tránh tình trạng overbooking.

Caching với Redis: Redis được tích hợp để lưu trữ tạm thời các dữ liệu thường xuyên truy vấn (danh mục, bài viết nổi bật, slider), giúp giảm tải cho cơ sở dữ liệu MySQL và tăng tốc độ phản hồi của API.

Bảo mật với JWT: Toàn bộ các endpoint quan trọng được bảo vệ bằng cơ chế xác thực JWT (JSON Web Token). Mỗi request phải kèm theo token hợp lệ, từ đó hạn chế truy cập trái phép. Đồng thời, dữ liệu nhạy cảm như mật khẩu người dùng được hash bằng bcrypt trước khi lưu vào cơ sở dữ liệu.

Xử lý lỗi và logging: Hệ thống có cơ chế ghi log chi tiết (bao gồm request, response, error stack) để phục vụ quá trình debug và bảo trì. Điều này đảm bảo khi có sự cố, đội ngũ kỹ thuật có thể nhanh chóng tìm ra nguyên nhân.

Những giải pháp kỹ thuật này đã giúp hệ thống vận hành ổn định, đảm bảo tốc độ, tính an toàn và khả năng mở rộng khi lượng truy cập tăng cao.

3.2.3 Công cụ và tài liệu nghiên cứu

Trong quá trình phát triển, nhóm không chỉ dựa vào kinh nghiệm cá nhân mà còn khai thác nhiều công cụ và nguồn tài liệu nghiên cứu:

Postman: Dùng để thiết kế và kiểm thử API, tạo collection test tự động, mô phỏng nhiều luồng nghiệp vụ trước khi tích hợp với front-end.

Git/GitHub: Quản lý source code, phân nhánh (branching), review code, đảm bảo sự phối hợp nhịp nhàng giữa các thành viên trong nhóm.

Tài liệu chuẩn RESTful API: Nghiên cứu các best practices trong việc thiết kế API RESTful (naming convention, error handling, versioning) để hệ thống tuân thủ chuẩn quốc tế và dễ dàng mở rộng.

Mô hình CSDL chuẩn hóa: Tham khảo tài liệu về thiết kế cơ sở dữ liệu 3NF (Third Normal Form) nhằm loại bỏ dư thừa dữ liệu, đảm bảo toàn vẹn và hiệu năng trong xử lý.

Nguồn tham khảo cộng đồng: StackOverflow, GitHub, và tài liệu chính thức của MySQL, NestJS, NextJS cũng được sử dụng để giải quyết các vấn đề kỹ thuật cụ thể.

Việc kết hợp công cụ kiểm thử, quản lý phiên bản và tài liệu nghiên cứu đã giúp nhóm triển khai hệ thống đúng chuẩn, giảm thiểu lỗi và rút ngắn thời gian phát triển.

3.3 Đánh giá thành quả đạt được

Sau quá trình phân tích, thiết kế và triển khai, hệ thống back-end cho website Làng nổi Tân Lập đã hoàn thiện với nhiều kết quả tích cực. Việc phát triển cơ sở dữ liệu, xây dựng API và áp dụng các giải pháp kỹ thuật đã đem lại một sản phẩm vận hành ổn định, đáp ứng phần lớn yêu cầu đề ra từ ban đầu.

3.3.1 Kết quả cuối cùng

Kết quả nổi bật nhất là hệ thống cơ sở dữ liệu quan hệ chuẩn hóa với 14 bảng, trong đó tôi phụ trách thiết kế và triển khai 7 bảng lõi (users, roles, posts, categories, media, sliders, post_media). Cơ sở dữ liệu được chuẩn hóa đến 3NF, đảm bảo không dư thừa và dễ mở rộng.

Song song với đó, các API RESTful đã được xây dựng đầy đủ cho nhóm bảng tôi phụ trách cũng như nhóm auth (đăng nhập, đổi mật khẩu, đăng ký). Toàn bộ API đều có input, process, output rõ ràng, được kiểm thử qua Postman và cho kết quả phản hồi chính xác. Hệ thống hiện có thể:

Quản lý tài khoản, phân quyền (users, roles).

Quản lý nội dung (posts, categories, media, sliders).

Tích hợp dễ dàng với front-end (NextJS) hoặc ứng dụng mobile thông qua chuẩn JSON API.

Đảm bảo cơ chế bảo mật bằng JWT, hash mật khẩu bằng bcrypt.

Có thể khẳng định sản phẩm đã đạt mức sẵn sàng triển khai thực tế và có thể tích hợp mở rộng trong tương lai.

3.3.2 So sánh với mục tiêu ban đầu

So với mục tiêu ban đầu của dự án, hệ thống back-end đã hoàn thành khoảng 95% khối lượng công việc đề ra. Các module lõi đều đã được thiết kế và chạy ổn định. Các yêu cầu chính (quản lý người dùng, nội dung, phân quyền, bảo mật, API RESTful) đã được đáp ứng đầy đủ. Một số phần còn thiếu hoặc có thể cải tiến thêm:

Chưa tích hợp caching Redis trong môi trường production (mới dừng lại ở mức thử nghiệm).

Cần tối ưu thêm một số query join phức tạp để cải thiện hiệu năng khi dữ liệu lớn.

Chưa hoàn thiện hệ thống log giám sát tập trung (logging + monitoring dashboard).

Nhìn chung, so với phạm vi ban đầu, sản phẩm đã đạt yêu cầu cả về chức năng lẫn hiệu năng, chỉ còn một số chi tiết kỹ thuật cần tối ưu thêm khi triển khai trên môi trường thực tế quy mô lớn.

3.3.3 Bài học, đúc kết kinh nghiệm

Về kỹ thuật: Hiểu sâu hơn về cách thiết kế cơ sở dữ liệu quan hệ chuẩn hóa, biết cách xử lý quan hệ phức tạp giữa nhiều bảng. Thành thạo việc thiết kế và triển khai API RESTful với NestJS, đồng thời nắm chắc hơn về bảo mật API (JWT, bcrypt).

Về teamwork: Học được cách điều phối công việc trong nhóm, phân chia module, hỗ trợ các thành viên khi gặp lỗi, đồng thời phối hợp chặt chẽ với front-end để đảm bảo API hoạt động đúng như thiết kế.

Về xử lý sự cố: Trực tiếp trải qua những khó khăn như concurrency khi đặt phòng, lỗi API do input không hợp lệ, từ đó nâng cao kỹ năng debugging, logging và viết middleware validate dữ liệu.

Về tư duy phát triển hệ thống: Nhận thức rõ rằng xây dựng hệ thống không chỉ dừng lại ở việc “làm cho chạy được”, mà cần chú trọng hiệu suất, bảo mật, khả năng mở rộng và tính bảo trì lâu dài.

Những kinh nghiệm này không chỉ giúp tôi hoàn thành kỳ thực tập mà còn là nền tảng quan trọng cho sự nghiệp lập trình back-end chuyên nghiệp sau này.

CHƯƠNG IV: TỔNG KẾT VÀ ĐỊNH HƯỚNG

4.1 Tổng kết quá trình thực tập

4.1.1 Tóm lược khối lượng và nội dung công việc đã hoàn thành

Trong suốt kỳ thực tập tại Công ty Cổ phần Tập đoàn HT Việt Nam, tôi đã được giao nhiệm vụ chính là phụ trách phát triển hệ thống back-end cho dự án website Làng nổi Tân Lập. Với vai trò Back-end Lead trong nhóm thực tập, tôi trực tiếp tham gia và hoàn thành nhiều công việc quan trọng, bao gồm:

Thiết kế cơ sở dữ liệu (Database): xây dựng và chuẩn hóa sơ đồ quan hệ dữ liệu (ERD) với tổng cộng 14 bảng, trong đó tôi phụ trách 7 bảng cốt lõi: users, roles, posts, categories, media, sliders, post_media.

Xây dựng API RESTful: triển khai thành công các API CRUD cho nhóm bảng phụ trách, đồng thời phát triển nhóm API auth (đăng ký, đăng nhập, đổi mật khẩu). Tất cả API đều được kiểm thử bằng Postman, phản hồi đúng yêu cầu và sẵn sàng tích hợp với web/mobile.

Áp dụng giải pháp kỹ thuật: sử dụng transaction (SELECT ... FOR UPDATE) để xử lý concurrency khi nhiều người dùng cùng đặt phòng; triển khai bảo mật JWT cho API; hash mật khẩu bằng bcrypt; xây dựng cơ chế logging phục vụ kiểm thử và bảo trì.

Điều phối nhóm back-end: tham gia phân chia module công việc, hỗ trợ đồng đội xử lý lỗi kỹ thuật, phối hợp chặt chẽ với nhóm front-end để thống nhất giao diện API và bảo đảm tiến độ.

Kết quả cuối cùng là hệ thống cơ sở dữ liệu hoạt động ổn định, API chạy đúng chức năng, có thể dễ dàng tích hợp với NextJS cho web và mở rộng sang mobile app. Đây là minh chứng rõ ràng cho khối lượng công việc đã hoàn thành và đóng góp thực tế của tôi trong dự án.

4.1.2 Nhận xét, đánh giá từ góc nhìn cá nhân

Từ góc nhìn cá nhân, kỳ thực tập này mang lại cho tôi nhiều trải nghiệm quý báu, không chỉ về mặt kỹ thuật mà còn về kỹ năng làm việc và thái độ nghề nghiệp.

Về kiến thức: Tôi đã củng cố và mở rộng đáng kể hiểu biết về cơ sở dữ liệu quan hệ, thiết kế ERD, triển khai API RESTful với NestJS, cũng như các kỹ thuật bảo mật (JWT, bcrypt). Đồng thời, việc tham gia phát triển hệ thống thực tế giúp tôi nhận thức rõ sự khác biệt giữa kiến thức lý thuyết học ở trường và cách áp dụng trong môi trường doanh nghiệp.

Về kỹ năng: Tôi học được cách làm việc nhóm hiệu quả, phân chia công việc, code review và phối hợp đa phòng ban (backend ↔ frontend). Ngoài ra, kỹ năng debugging, xử lý concurrency, và viết log để giám sát hệ thống cũng được nâng cao rõ rệt.

Về thái độ: Tôi rèn luyện được tinh thần trách nhiệm, tính kiên nhẫn trong việc xử lý sự cố, và ý thức tự học, tự nghiên cứu khi gặp khó khăn. Tôi nhận ra rằng để trở thành một lập trình viên giỏi, ngoài kiến thức và kỹ năng, còn cần thái độ cầu tiến, chủ động và tinh thần hợp tác.

Tổng thể, quá trình thực tập đã giúp tôi trưởng thành hơn về cả chuyên môn lẫn tư duy nghề nghiệp. Tôi tự tin rằng những trải nghiệm này sẽ là hành trang quý báu để tôi bước vào con đường sự nghiệp trong lĩnh vực phát triển phần mềm, đặc biệt là back-end.

4.2 Định hướng và kiến nghị

Sau khi hoàn thành kỳ thực tập và triển khai thành công hệ thống back-end cho dự án website Làng nổi Tân Lập, tôi nhận thấy rằng sản phẩm hiện tại đã đạt được những kết quả khả quan, tuy nhiên vẫn còn nhiều tiềm năng để phát triển và tối ưu hơn nữa. Phần này sẽ đưa ra một số định hướng và kiến nghị cho hệ thống, cho doanh nghiệp, đồng thời chia sẻ định hướng phát triển cá nhân trong thời gian tới.

4.2.1 Gợi ý cải tiến kỹ thuật/hệ thống

Về mặt kỹ thuật, hệ thống có thể được nâng cấp theo nhiều hướng để đạt hiệu suất và tính linh hoạt cao hơn:

Tối ưu cơ sở dữ liệu (DB): Rà soát và bổ sung index cho các cột truy vấn nhiều, tối ưu các câu lệnh SQL join phức tạp để giảm thời gian phản hồi. Ngoài ra, áp dụng sharding hoặc replication để nâng cao khả năng chịu tải khi dữ liệu tăng trưởng nhanh.

Mở rộng API sang GraphQL: Hiện tại API được thiết kế theo chuẩn RESTful, tuy nhiên với các ứng dụng phức tạp, việc bổ sung GraphQL sẽ giúp client chỉ truy vấn đúng dữ liệu cần thiết, giảm overhead và tăng hiệu quả tích hợp.

Áp dụng caching mạnh mẽ hơn: Redis đã được thử nghiệm trong giai đoạn phát triển, cần triển khai chính thức trong môi trường production để cache dữ liệu thường xuyên truy vấn (danh mục, media, bài viết nổi bật), giúp giảm tải cho DB và tăng tốc độ phản hồi.

Cải thiện logging và monitoring: Xây dựng hệ thống giám sát tập trung (ELK stack, Prometheus, Grafana) để theo dõi sức khỏe hệ thống, giúp phát hiện lỗi kịp thời và chủ động trong bảo trì.

Những cải tiến này sẽ giúp hệ thống vận hành hiệu quả hơn, đáp ứng tốt nhu cầu mở rộng trong tương lai.

4.2.2 Kiến nghị có lợi cho doanh nghiệp

Từ góc nhìn thực tập sinh tham gia trực tiếp dự án, tôi nhận thấy doanh nghiệp có thể triển khai thêm một số hướng đi chiến lược để phát huy giá trị hệ thống:

Triển khai hệ thống ra môi trường thực tế: Sau giai đoạn phát triển và kiểm thử, hệ thống back-end cần được đưa vào vận hành chính thức, kết nối với website để phục vụ khách hàng thực tế.

Phát triển ứng dụng mobile: Với xu hướng du lịch trực tuyến, việc xây dựng ứng dụng di động (Android, iOS) sẽ giúp mở rộng tệp khách hàng và nâng cao trải nghiệm người dùng. API hiện tại đã sẵn sàng để tích hợp mobile app.

Liên kết với đối tác du lịch và khách sạn: Mở rộng kết nối API với hệ thống của các đối tác, từ đó tăng khả năng khai thác dịch vụ, đồng thời tạo hệ sinh thái du lịch trực tuyến đa dạng và phong phú hơn.

Đầu tư vào an toàn thông tin: Với đặc thù quản lý dữ liệu khách hàng và giao dịch, doanh nghiệp cần tiếp tục nâng cấp chính sách bảo mật, tuân thủ các tiêu chuẩn an ninh quốc tế để giữ vững uy tín và niềm tin của khách hàng.

Việc triển khai các kiến nghị trên sẽ giúp công ty không chỉ có một sản phẩm công nghệ mạnh mẽ, mà còn tận dụng được tối đa tiềm năng thị trường du lịch trực tuyến.

4.2.3 Định hướng nghiên cứu và phát triển bản thân

Về phía cá nhân, thông qua kỳ thực tập này tôi đã có thêm nhiều kiến thức và kỹ năng thực tế. Trong tương lai, tôi định hướng sẽ tập trung vào ba hướng chính để nâng cao năng lực bản thân:

Học thêm về Cloud Computing: Làm quen và thực hành triển khai hệ thống trên các nền tảng đám mây (AWS, Azure, GCP) nhằm đảm bảo hệ thống có thể scale tự động và hoạt động ổn định ở môi trường thực tế.

Nghiên cứu chuyên sâu về bảo mật: Học và áp dụng các phương pháp bảo mật tiên tiến (OAuth2, SSL/TLS nâng cao, phòng chống DDoS), từ đó bảo vệ hệ thống trước các mối đe dọa ngày càng tinh vi.

Tiếp cận DevOps và CI/CD: Thành thạo các công cụ tự động hóa triển khai (Docker, Kubernetes, Jenkins, GitHub Actions) để rút ngắn chu kỳ phát triển, tăng tốc độ release và giảm rủi ro khi deploy.

Thông qua việc đầu tư cho học tập và nghiên cứu, tôi tin rằng bản thân sẽ có đủ năng lực để tham gia vào những dự án lớn hơn, đóng góp nhiều giá trị hơn cho doanh nghiệp và cộng đồng công nghệ.

TÀI LIỆU THAM KHẢO

- 1) Oracle. (n.d.). *MySQL 8.0 Reference Manual*. Oracle. Retrieved from <https://dev.mysql.com/doc/>
- 2) NestJS. (n.d.). *A progressive Node.js framework*. Retrieved from <https://docs.nestjs.com/>
- 3) Vercel. (n.d.). *Next.js Documentation: The React Framework for Production*. Retrieved from <https://nextjs.org/docs>
- 4) Postman. (n.d.). *API Platform Documentation*. Retrieved from <https://learning.postman.com/docs/>
- 5) Redis. (n.d.). *Redis Documentation*. Retrieved from <https://redis.io/docs/>
- 6) JWT.io. (n.d.). *Introduction to JSON Web Tokens*. Retrieved from <https://jwt.io/introduction/>
- 7) Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems (7th ed.)*. Pearson.
- 8) Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
- 9) 123docz. (n.d.). *Bài tập lớn xây dựng website du lịch giới thiệu tour du lịch và đặt tour*. Retrieved from <https://text.123docz.com/document/3474197-bai-tap-lon-xay-dung-website-du-lich-gioi-thieu-tour-du-lich-va-dat-tour.htm>
- 10) Studocu. (n.d.). *Báo cáo bài tập lớn JavaScript. Học viện Công nghệ Bưu chính Viễn thông*. Retrieved from <https://www.studocu.vn/vn/document/hoc-vien-cong-nghe-buu-chinh-vien-thong/javadeveloper/bao-cao-bai-tap-lon-javascript/113180584>
- 11) Docx.com.vn. (n.d.). *Bài tập lớn thiết kế website du lịch Trảng An – môn Ngôn ngữ lập trình 1*. Retrieved from <https://docx.com.vn/tai-lieu/bai-tap-lon-thiet-ke-website-du-lich-trang-an-mon-ngon-ngu-lap-trinh-1-178692>
- 12) Scribd. (n.d.). *Bài tập lớn xây dựng website du lịch giới thiệu tour du lịch và đặt tour*. Retrieved from <https://www.scribd.com/document/491465507/123doc-bai-tap-lon-xay-dung-website-du-lich-gioi-thieu-tour-du-lich-va-dat-tour>

- 13) Studocu. (n.d.). Báo cáo thiết kế web. *Trường Đại học Mở Hà Nội*. Retrieved from <https://www.studocu.vn/vn/document/truong-dai-hoc-mo-ha-noi/thiet-ke-website/baocaothietkeweb/75890137>
- 14) 123docz. (n.d.). *Bài tập lớn cuối kỳ môn Lập trình ngôn ngữ Script – Đề tài: Website về du lịch*. Retrieved from <https://123docz.com/document/15870478-ba-i-ta-p-lo-n-cuo-i-ky-mo-n-la-p-tri-nh-ngo-n-ngu-script-de-ta-i-website-ve-du-li-ch.htm>
- 15) Studocu. (n.d.). *Thiết kế web – Nhóm 7. Trường Đại học Phenikaa*. Retrieved from <https://www.studocu.vn/vn/document/truong-dai-hoc-phenikaa/ung-dung-web/thiet-ke-web-nhom-7/86613002>
- 16) Scribd. (n.d.). *Báo cáo bài tập lớn môn Thiết kế Web – Xây dựng website tin tức du lịch*. Retrieved from <https://www.scribd.com/document/873168622/Bao-Cao-Bai-Tap-Lon-Mon-thiet-Ke-Web-Xay-Dung-Website-Tin-Tuc-Du-Lich>
- 17) Luận văn Online. (n.d.). *Đồ án thiết kế website tour du lịch*. Retrieved from <https://luanvan.net.vn/luan-van/do-an-thiet-ke-website-tour-du-lich-27131/>