
BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC PHENIKAA



**BÁO CÁO BÀI TẬP LỚN
HỌC PHẦN KĨ THUẬT PHẦN MỀM**

Đề Tài: VistaNest trang web quản lý dịch vụ tour du lịch

Lớp học phần: Kĩ thuật phần mềm-1-3-24(COUR01.LT8)

Giảng viên hướng dẫn : Vũ Quang Dũng

Nhóm 3: - Nguyễn Thọ Nhân - 23010786
- Phạm Anh Thái - 23010784
- Nguyễn Xuân Chúc - 23010452
- Hoàng Duy Sáng - 23010481

Hà Nội, 06/2025

MỤC LỤC

A. Lời Mở đầu.....	3
B. Bảng Phân Chia Công Việc Theo Tuần.....	4
C. Phụ lục và Tài nguyên Dự án.....	5
C.1. Liên kết GitHub, Demo, Slide.....	5
C.2. Hướng dẫn Cài đặt và Khởi chạy.....	6
1. Khởi tạo và Phân tích án.....	7
1.1. Giới thiệu Tổng quan Dự án.....	7
1.2. Phân tích Mục tiêu (S.M.A.R.T)	7
1.3. Phân tích Đối tượng Người dùng và Các bên liên quan.....	7
1.4. Phân tích Yêu cầu Kỹ thuật và Phi chức năng.....	8
1.5. Đối tượng Người dùng.....	8
1.6. Kiến trúc Hệ thống và Lựa chọn Công nghệ.....	8
1.6.1. Sơ đồ Kiến trúc.....	9
1.6.2. Công nghệ được lựa chọn.....	9
2. Thiết kế và Xây dựng Giao diện (Frontend)	10
2.1. Tổng quan và Triết lý Thiết kế.....	10
2.2. Xây dựng Thành phần Giao diện chính.....	11
2.2.1. Trang chủ (Home)	11
2.2.2. Trang Chi tiết Tour.....	11
2.2.3. Các Trang Danh sách Tour.....	11
2.3. Xây dựng Xác thực Sinh trắc học.....	11
2.3.1. Giao diện Đăng nhập/Đăng ký.....	11
2.3.2. Tích hợp Nhận diện Khuôn mặt.....	11
2.4. Giao diện Chatbot AI.....	12
3. Xây dựng Logic phía Máy chủ (Backend)	12
3.1. Kiến trúc Backend tổng thể.....	12
3.2. Thiết kế và triển khai Cơ sở Dữ liệu.....	12
3.3. Xây dựng Các API (Endpoints)	13
3.3.1. Xác thực (/api/auth)	13
3.3.2. Quản lý Đơn hàng (/api/orders)	13
3.3.3. Các Module CRUD khác.....	13
3.4. Dịch vụ Chatbot AI (/api/chat)	14
4. Tích hợp và Hoàn thiện Trang Quản trị (Admin)	14
4.1. Kiến trúc và Bố cục Giao diện.....	14
4.2. Bảo mật và Phân quyền người dùng.....	15
4.3. Các Module Quản trị.....	15
4.3.1. Bảng điều khiển.....	15
4.3.2. Quản lý Đơn hàng và Doanh thu.....	15
4.3.3. Quản lý Sản phẩm (Tour)	15
4.3.4. Quản lý Tương tác (Đánh giá & Hỗ trợ)	15
5. Phân tích, Đặc tả và Thiết kế Hệ thống.....	16

5.1. Phân tích theo Tác nhân (Actor-based Functional Analysis)	17
5.2. Thiết kế Hệ thống.....	17
a. Biểu đồ Use Case.....	17
b. Biểu đồ Tuần tự.....	17
c. Biểu đồ Lớp (Class Diagram)	17
d. DFD và Cơ sở dữ liệu.....	17
e. Luồng Giao diện (UI Flow)	17
6. Bảo mật và Middleware.....	21
6.1. Middleware đã triển khai.....	21
6.2. Các biện pháp Bảo mật.....	21
6.3. Định hướng bảo mật tương lai.....	21
7. Demo Giao diện và Video.....	22
7.1. Video Demo Tổng quan.....	23
7.2. Hình ảnh Giao diện Tiêu biểu.....	24
8. Kiểm thử và Định hướng Phát triển.....	33
8.1. Kịch bản và Kết quả Kiểm thử.....	33
8.2. Đánh giá Hệ thống.....	33
8.3. Định hướng phát triển tương lai.....	33
8.3.1. Hoàn thiện & Tối ưu hệ thống hiện tại.....	33
8.3.2. Thanh toán bằng Sinh trắc học.....	34
8.3.3. Mở rộng hệ thống & Triển khai.....	34
8.3.4. Tài liệu tham khảo.....	34
D. Lời Cảm Ơn.....	35

A. Lời Mở đầu

Kính gửi Quý Thầy/Cô và toàn thể độc giả,

Trong bối cảnh của cuộc Cách mạng Công nghiệp 4.0, sự hội tụ giữa công nghệ thông tin và các ngành dịch vụ truyền thống đang mở ra những cơ hội và thách thức chưa từng có. Ngành du lịch, với vai trò là một trong những động lực kinh tế quan trọng, cũng không nằm ngoài xu hướng này. Việc ứng dụng công nghệ số không chỉ là một lợi thế cạnh tranh, mà đã trở thành một yêu cầu tất yếu để nâng cao trải nghiệm khách hàng và tối ưu hóa quy trình vận hành.

Nhận thức sâu sắc về tiềm năng đó, chúng tôi đã thực hiện dự án "**VistaNest: Xây dựng và Phát triển Nền tảng Đặt tour Du lịch Thông minh**". Đây không chỉ là một bài toán kỹ thuật về việc xây dựng một website thương mại điện tử, mà còn là một công trình nghiên cứu và ứng dụng các công nghệ tiên tiến như Trí tuệ Nhân tạo (AI) trong Nhận diện khuôn mặt và Xử lý Ngôn ngữ Tự nhiên vào một lĩnh vực thực tiễn.

Bản báo cáo này được biên soạn nhằm mục đích trình bày một cách hệ thống và chi tiết toàn bộ quá trình thực hiện dự án, từ giai đoạn khởi tạo ý tưởng, phân tích yêu cầu, thiết kế kiến trúc hệ thống, cho đến việc triển khai các chức năng cụ thể và định hướng phát triển trong tương lai. Qua đó, chúng tôi hy vọng có thể cung cấp một cái nhìn tổng quan, mang tính học thuật và thực tiễn về việc kiến tạo một sản phẩm công nghệ hoàn chỉnh.

Chúng tôi rất mong nhận được những ý kiến đóng góp quý báu từ Quý Thầy/Cô và các bạn độc giả để dự án có thể ngày càng hoàn thiện hơn.

Trân trọng

B - Bảng phân chia công việc chi tiết của các thành viên trong nhóm theo từng tuần

Tuần	Nguyễn Thọ Nhân (23010786)	Phạm Anh Thái (23010784)	Hoàng Duy Sáng	Nguyễn Xuân Chức
1	💡 Chủ trì và thống nhất ý tưởng, mục tiêu, phạm vi dự án. Phân công nhiệm vụ ban đầu cho các thành viên.	💻 Thiết lập môi trường Git, cấu trúc thư mục dự án theo chỉ đạo.	📝 Phân tích chi tiết và viết tài liệu yêu cầu chức năng (SRS).	📁 Thiết kế lược đồ CSDL tourdb và các mối quan hệ.
2	☰ Thiết kế và phê duyệt kiến trúc hệ thống tổng thể (Client, Server, AI). Giám sát tiến độ chung.	⚙️ Lập trình API Xác thực (auth.js) - module nền tảng cho hệ thống.	⌚ Code giao diện Trang chủ và Chi tiết tour - bộ mặt chính của dự án.	📝 Viết và hoàn thiện script setupDatabase.js để tạo bảng và seed dữ liệu.
3	✍️ Nghiên cứu và tích hợp công nghệ lõi: Nhận diện khuôn mặt (face-api.js), đảm bảo tính đột phá của dự án.	⚙️ Lập trình API CRUD cho Tour, làm việc trực tiếp với NXC để đồng bộ CSDL.	⌚ Code giao diện các Trang danh sách và logic phân trang phía client.	⚙️ Lập trình API CRUD cho Khách hàng, đảm bảo tính nhất quán dữ liệu người dùng.
4	✉️ Nghiên cứu và phát triển AI Service (chat.py) để kết nối Google Gemini, một nhiệm vụ công nghệ cao.	⚙️ Lập trình API Quản lý Đơn hàng (orders.js) với transaction phức tạp.	⌚ Code toàn bộ giao diện Trang Quản trị (Admin) theo thiết kế.	⌚ Xây dựng giao diện Chatbot (chat/) để sẵn sàng tích hợp với AI Service.
5	🔗 Tích hợp toàn bộ hệ thống: Đảm bảo các module Frontend và Backend giao tiếp trơn tru. Chủ trì gỡ lỗi tích hợp.	🔗 Hỗ trợ NTN tích hợp Backend cho trang Quản lý Sản phẩm và Khách hàng.	📊 Tích hợp Chart.js vào Admin Dashboard và hiển thị dữ liệu động.	🔗 Hỗ trợ NTN tích hợp Backend cho trang Quản lý Đơn hàng và Đánh giá.
6	⌚ Tập trung vào Bảo mật: Rà soát các lỗ hổng tiềm tàng, lên kế hoạch và nghiên cứu triển khai JWT, Helmet.js.	⚙️ Thực hiện kiểm thử (testing) các API Backend và hiệu năng CSDL.	⌚ Thực hiện kiểm thử giao diện và trải nghiệm người dùng trên các trình duyệt, thiết bị khác nhau.	⌚ Hoàn thiện toàn bộ CSS và Responsive cho dự án.
7	📄 Tổng hợp toàn bộ Báo cáo dự án, đảm bảo tính logic, học thuật và nhất quán.	📊 Thiết kế slide PowerPoint chuyên nghiệp cho buổi báo cáo.	📄 Hỗ trợ nhóm viết chi tiết các phần báo cáo	📄 Hỗ trợ nhóm viết chi tiết các phần báo cáo về Backend và CSDL.

8	<p> Thuyết trình, đóng gói sản phẩm cuối cùng.</p>	<p> Hoàn thiện tài liệu README .md, hướng dẫn cài đặt.</p>	<p><input checked="" type="checkbox"/> Rà soát, kiểm tra lỗi chính tả, định dạng cho toàn bộ tài liệu và slide.</p>	<p><input checked="" type="checkbox"/> Chuẩn bị các câu hỏi-đáp (Q&A) có thể gặp trong buổi bảo vệ.</p>
----------	---	--	---	---

C - Phụ lục và Tài nguyên Dự án / Hướng dẫn khởi chạy

Hạng mục	Biểu tượng	Mô tả	Liên kết truy cập
Mã nguồn (Source Code)		Toàn bộ mã nguồn của dự án được lưu trữ và quản lý phiên bản trên nền tảng GitHub.	https://github.com/NguyenThoNhan/CSE702025-Nhom3
Website Live Demo		Phiên bản demo của trang web VistaNest được triển khai và hoạt động trực tuyến, cho phép trải nghiệm trực tiếp các chức năng của người dùng.	https://nguyenthonhan.github.io/Service_Travel/client
Báo cáo Dự án (PDF)		Phiên bản điện tử đầy đủ của tài liệu báo cáo này, có thể tải về để lưu trữ và tham khảo chi tiết.	CSE702025-Nhom3-report.docx
Bài trình bày (PowerPoint)		File trình chiếu (slide) được sử dụng cho buổi báo cáo, tóm tắt các điểm chính, kiến trúc và kết quả của dự án.	https://www.canva.com/design/DAGqrXdxc78/wY9ycY82PnngQh6HItkOlw/edit

Lưu ý khi truy cập:

- Tài khoản Admin Demo:** Để truy cập trang quản trị, vui lòng sử dụng thông tin đăng nhập sau:
 - Username:** admin
 - Password:** adminpassword
- Mã nguồn:** Mã nguồn trên GitHub được tổ chức theo các thư mục tương ứng với kiến trúc đã trình bày (client, server, chat, travel). Vui lòng tham khảo file README .md để có hướng dẫn cài đặt chi tiết.

C.2. Hướng dẫn Cài đặt và Khởi chạy

Yêu cầu môi trường:

- Node.js (phiên bản 18.x trở lên)
- Python (phiên bản 3.9 trở lên)
- Hệ quản trị CSDL MySQL (khuyến khích sử dụng XAMPP hoặc MySQL Workbench)

Các bước thực thi:

1. Khởi tạo Dự án và Cài đặt Thư viện:

- Mở terminal tại thư mục gốc của dự án.
- Chạy lệnh `npm install`.

2. Thiết lập Cơ sở dữ liệu:

- **Bước 2.1:** Đảm bảo dịch vụ MySQL đang hoạt động.
- **Bước 2.2:** `node setupDatabase.js`

3. Khởi chạy các Dịch vụ Backend:

- **Bước 3.1: Khởi chạy Core Service (Node.js):** Mở một terminal mới, di chuyển vào thư mục `server` và chạy lệnh:

```
node index.js
```

Kết quả mong đợi: Server chính lắng nghe trên cổng 3000 và kết nối thành công tới CSDL.

- **Bước 3.2: Khởi chạy AI Service (Python):** Mở một terminal khác, di chuyển vào thư mục `chat` và chạy lệnh:

```
python chat.py
```

Kết quả mong đợi: Service chatbot lắng nghe trên cổng 5000, sẵn sàng nhận yêu cầu từ Core Service.

4. Khởi chạy Giao diện Người dùng (Frontend):

- Mở một terminal cuối cùng, di chuyển vào thư mục `client` (hoặc `travel`) và sử dụng `live-server` để phục vụ các file tĩnh.

```
live-server
```

I - Khởi tạo và Phân tích Dự án

1. Giới thiệu Tổng quan Dự án

1.1. Tuyên bố Tầm nhìn và Sứ mệnh

- **Tên dự án: VistaNest**
- **Tầm nhìn (Vision):** Trở thành nền tảng du lịch thông minh, đáng tin cậy hàng đầu, nơi mỗi hành trình của khách hàng đều được cá nhân hóa và hỗ trợ bởi công nghệ tiên tiến.
- **Sứ mệnh (Mission):** Đơn giản hóa quá trình tìm kiếm và đặt tour du lịch thông qua một giao diện trực quan, đồng thời cung cấp các công cụ quản trị mạnh mẽ để tối ưu hóa hoạt động kinh doanh cho doanh nghiệp.

1.2. Bối cảnh và Vấn đề

Thị trường du lịch trực tuyến hiện nay có tính cạnh tranh cao nhưng vẫn còn nhiều khoảng trống. Người dùng thường gặp khó khăn trong việc tìm kiếm thông tin tour một cách minh bạch, so sánh giá cả và nhận được sự hỗ trợ tức thì. Về phía doanh nghiệp, việc quản lý thủ công các tour, đơn hàng và tương tác khách hàng tốn nhiều thời gian và dễ xảy ra sai sót. Dự án VistaNest được ra đời để giải quyết những thách thức này.

2. Phân tích Mục tiêu (S.M.A.R.T)

Dự án hướng đến các mục tiêu cụ thể, có thể đo lường được:

- **(S)pecific - Cụ thể:** Xây dựng một website du lịch hoàn chỉnh với hai thành phần chính: trang cho khách hàng (Client) và trang cho quản trị viên (Admin).
- **(M)easurable - Đo lường được:** Hoàn thành 100% các chức năng CRUD (Thêm, Sửa, Xóa, Xem) cho các module Tour, Đơn hàng, Khách hàng; tỷ lệ phản hồi của chatbot AI đạt trên 80% cho các câu hỏi thường gặp.
- **(A)chievable - Khả thi:** Sử dụng các công nghệ phổ biến, có cộng đồng hỗ trợ lớn (JavaScript, Python, MySQL), đảm bảo tính khả thi trong việc phát triển và triển khai.
- **(R)elevant - Liên quan:** Các chức năng được xây dựng trực tiếp giải quyết các vấn đề thực tiễn của ngành du lịch, từ việc thu hút khách hàng đến tối ưu hóa vận hành.
- **(T)ime-bound - Có thời hạn:** Hoàn thành các giai đoạn chính của dự án theo kế hoạch đã đề ra (chi tiết sẽ có trong các giai đoạn sau).

3. Phân tích Đối tượng Người dùng và Các bên liên quan

- **Khách hàng (End-User):**
 - **Đặc điểm:** Độ tuổi từ 18-45, am hiểu công nghệ, có thói quen tìm kiếm và mua sắm trực tuyến.
 - **Nhu cầu:** Cần một nền tảng nhanh, đẹp, dễ sử dụng, thông tin rõ ràng, có đánh giá xác thực và quy trình thanh toán an toàn.
 - **"Pain Point" (Nỗi đau):** Mất thời gian tìm kiếm, thông tin tour không nhất quán, khó nhận được hỗ trợ nhanh chóng.
- **Quản trị viên (Admin):**
 - **Đặc điểm:** Nhân viên điều hành tour, nhân viên kinh doanh, quản lý.
 - **Nhu cầu:** Cần một công cụ tập trung để quản lý toàn bộ sản phẩm, theo dõi đơn hàng, quản lý thông tin khách hàng và xem báo cáo kinh doanh trực quan.
 - **"Pain Point" (Nỗi đau):** Phải làm việc trên nhiều công cụ rời rạc (Excel, email, phần mềm chat), khó tổng hợp dữ liệu, tốn thời gian cho các tác vụ lặp đi lặp lại.
- **Các bên liên quan khác:**
 - **Nhà phát triển (Developer):** Cần một kiến trúc hệ thống rõ ràng, dễ bảo trì và mở rộng.
 - **Ban quản lý dự án:** Cần theo dõi tiến độ và đảm bảo các mục tiêu được hoàn thành.

4. Phân tích Yêu cầu Kỹ thuật và Phi kỹ thuật

4.1. Yêu cầu Chức năng (Functional Requirements)

- **Module Xác thực:**
 - Đăng ký/Đăng nhập bằng tài khoản (email/username, password).
 - Mật khẩu phải được băm (hashed) trước khi lưu vào CSDL.
 - Phân quyền truy cập dựa trên vai trò (Admin/User).
 - Tích hợp đăng nhập/đăng ký bằng **Nhận diện khuôn mặt (Face Recognition)**.
- **Module Quản lý Tour (Sản phẩm):**
 - Admin: Thực hiện CRUD (Thêm, Xem, Sửa, Xóa) thông tin tour.
 - User: Xem danh sách tour, lọc theo tiêu chí (giá, tên), xem chi tiết tour.
- **Module Quản lý Đơn hàng (Giao dịch):**
 - User: Thêm tour vào giỏ hàng, tiến hành thanh toán (giả lập).
 - Admin: Xem danh sách đơn hàng, cập nhật trạng thái (Phê duyệt/Từ chối).
- **Module Tương tác:**
 - User: Đăng đánh giá (sao và bình luận) cho tour đã trải nghiệm.
 - Admin: Xem và quản lý (xóa) các đánh giá không phù hợp.
- **Module AI (Chatbot):**

- Tích hợp chatbot sử dụng **Google Gemini API** vào giao diện người dùng để trả lời các câu hỏi phổ biến.

4.2. Yêu cầu Phi chức năng (Non-functional Requirements)

- Hiệu năng (Performance):** Thời gian tải trang chủ dưới 3 giây. Thời gian phản hồi API dưới 500ms cho các yêu cầu thông thường.
- Bảo mật (Security):** Băm mật khẩu, ngăn chặn các tấn công cơ bản như SQL Injection (qua prepared statements). Phân quyền API để chỉ admin mới có thể thực hiện các hành động quản trị.
- Khả năng sử dụng (Usability):** Giao diện trực quan, dễ học, dễ sử dụng cho cả khách hàng và admin. Hỗ trợ theme Sáng/Tối cho trang quản trị.
- Khả năng mở rộng (Scalability):** Kiến trúc module hóa (tách biệt backend API và frontend) cho phép dễ dàng thêm các tính năng mới trong tương lai.

5. Đối tượng Người dùng

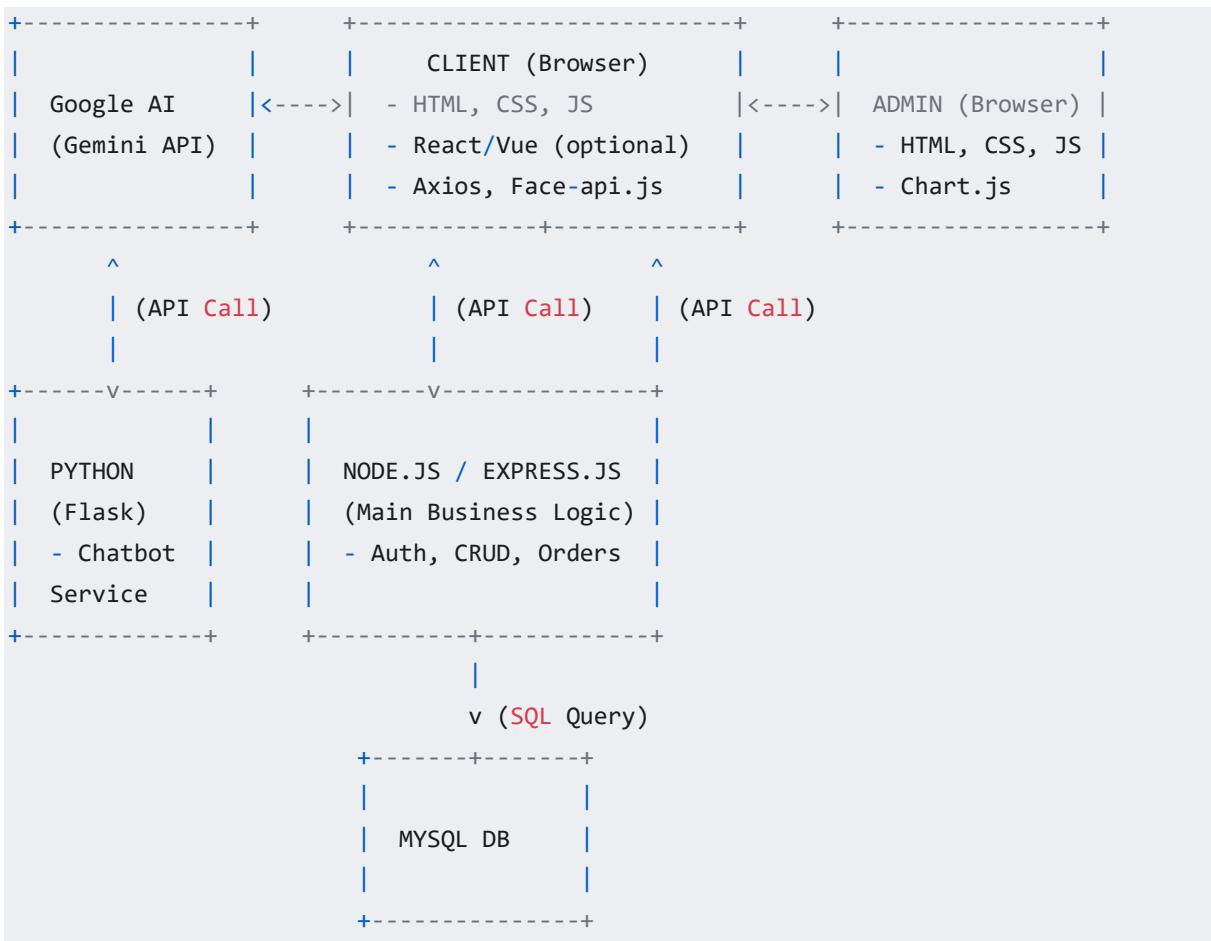
Dự án xác định hai nhóm đối tượng người dùng chính với các nhu cầu và vai trò khác nhau:

Vai trò	Biểu tượng	Mô tả	Nhu cầu chính
Khách hàng (User)		Những người yêu thích du lịch, có nhu cầu tìm kiếm thông tin, so sánh và đặt tour một cách nhanh chóng, tiện lợi.	 Tìm kiếm tour dễ dàng  Thông tin tour chi tiết, minh bạch  Quy trình đặt tour và thanh toán đơn giản  Xem đánh giá từ người dùng khác
Quản trị viên (Admin)		Nhân viên, quản lý của công ty du lịch. Họ cần một công cụ tập trung để quản lý toàn bộ hoạt động kinh doanh trên nền tảng	 Bảng điều khiển tổng quan  Quản lý sản phẩm  Quản lý đơn hàng  Quản lý khách hàng và đánh giá

6. Kiến trúc Hệ thống và Lựa chọn Công nghệ

6.1. Sơ đồ Kiến trúc Tổng thể

Hệ thống được thiết kế theo kiến trúc **Microservices-oriented**, trong đó các thành phần chính được tách biệt để dễ phát triển và bảo trì.



6.2. Công nghệ được lựa chọn

Kiến trúc dự án được xây dựng dựa trên sự kết hợp của nhiều công nghệ hiện đại và phổ biến:

Hạng mục	Công nghệ	Biểu tượng	Lý do lựa chọn
Giao diện (Frontend)	HTML5, CSS3, JavaScript (ES6+)	🌐	Nền tảng web tiêu chuẩn, linh hoạt và mạnh mẽ.
Máy chủ Logic Chính	Node.js & Express.js	🌐	Xây dựng API RESTful hiệu năng cao, xử lý các yêu cầu CRUD, xác thực người dùng. Phù hợp với các ứng dụng I/O bất đồng bộ.
Máy chủ AI (Chatbot)	Python & Flask/FastAPI	🤖	Python là ngôn ngữ hàng đầu cho các ứng dụng AI/ML. Flask hoặc FastAPI là các micro-framework nhẹ, lý tưởng để xây dựng một service chuyên biệt cho việc giao tiếp với Google Gemini API.
Cơ sở dữ liệu	MySQL	MYSQL	Hệ quản trị CSDL quan hệ phổ biến, ổn định, có cấu trúc rõ ràng, phù hợp để lưu trữ và truy vấn dữ liệu có quan hệ như tour và đơn hàng.
Thư viện Frontend	Axios, Chart.js, Face-api.js	💡	Axios: Giúp việc gọi API từ client đến server trở nên dễ dàng. Chart.js: Vẽ các biểu đồ thống kê đẹp mắt. Face-api.js: Tích hợp tính nhận diện khuôn mặt.
Môi trường & Công cụ	VS Code, Live Server, Git, MySQL Workbench	💻	Bộ công cụ phát triển tiêu chuẩn, mạnh mẽ và hiệu quả.

II - Thiết kế và Xây dựng Giao diện (Frontend)

1. Tổng quan Giai đoạn và Triết lý Thiết kế

Giai đoạn 2 tập trung vào việc hiện thực hóa các yêu cầu đã phân tích thành một bộ giao diện người dùng (Frontend) hoàn chỉnh và có tính tương tác cao. Toàn bộ quá trình được xây dựng dựa trên triết lý "**Trải nghiệm Hiện đại, Tương tác Thông minh**", với mục tiêu không chỉ tạo ra một website đẹp mắt mà còn tích hợp các công nghệ tiên tiến để nâng cao trải nghiệm người dùng.

Cấu trúc Frontend của dự án được tổ chức một cách khoa học trong các thư mục chính:

- **travel/**: Chứa các trang giao diện chính dành cho khách hàng, như trang chủ, chi tiết tour, danh sách sản phẩm.
- **client/**: Chứa các trang và tài nguyên liên quan đến xác thực người dùng (Đăng ký, Đăng nhập).
- **chat/**: Chứa giao diện và logic cho tính năng Chatbot AI.

2. Xây dựng các Thành phần Giao diện Cốt lõi (Thư mục travel/)

Đây là "bộ mặt" chính của VistaNest, nơi khách hàng tương tác và khám phá sản phẩm.

2.1. Trang chủ (travel/index.html) - Cổng vào Trải nghiệm

- **Thiết kế:** Xây dựng một trang chủ năng động với **Hero Banner** sử dụng video, tạo ấn tượng thị giác mạnh mẽ. Bố cục được chia thành các khối (sections) rõ ràng: Tour Nổi Bật, Điểm đến Yêu thích, và Tin tức, tất cả đều được trình bày dưới dạng lưới (CSS Grid) hiện đại.
- **Tương tác:** Form tìm kiếm được đặt ở vị trí trung tâm, là điểm bắt đầu cho hành trình của người dùng. Các thẻ tour (tour cards) được bổ sung hiệu ứng hover tinh tế (phóng to ảnh, đổ bóng) để tăng tính tương tác.

2.2. Trang Chi tiết Tour (travel/tour-detail.html) - Trung tâm Thông tin

- **Bố cục:** Áp dụng bố cục 2 cột tối ưu:
 - **Cột chính (Nội dung):** Hiển thị chi tiết lịch trình, hình ảnh, form đặt tour với logic tính giá động theo số lượng khách.

- **Cột phụ (Sidebar):** Sử dụng position: sticky để khói thông tin tóm tắt (giá, nút đặt tour) luôn trong tầm mắt của người dùng khi cuộn trang.
- **Tương tác xã hội:** Tích hợp khu vực **Đánh giá và Bình luận**, cho phép người dùng chấm điểm sao và để lại nhận xét. Dữ liệu này được quản lý bằng JavaScript và localStorage, tạo ra bằng chứng xã hội (social proof) và tăng độ tin cậy cho tour.

2.3. Các Trang Danh sách (tours-domestic.html, promotions.html, etc.)

- **Thiết kế:** Sử dụng một mẫu (template) chung để đảm bảo tính nhất quán, bao gồm banner, form tìm kiếm "nổi" và thanh sắp xếp.
- **Chức năng:** Toàn bộ logic **lọc, sắp xếp, và phân trang** được xử lý hoàn toàn ở phía client bằng JavaScript, mang lại trải nghiệm nhanh và mượt mà mà không cần tải lại trang.

3. Xây dựng Module Xác thực Sinh trắc học (Thư mục

client/)

Đây là một trong những điểm nhấn công nghệ của dự án, mang lại sự tiện lợi và bảo mật vượt trội.

3.1. Giao diện Đăng nhập/Đăng ký (client/signin.html, client/signup.html)

- **Thiết kế:** Áp dụng phong cách "**Glassmorphism**" với hiệu ứng nền "Aurora" động, tạo cảm giác hiện đại, công nghệ cao. Giao diện được tối ưu để tập trung vào các trường nhập liệu.
- **UI/UX:** Các ô input được thiết kế với icon bên trong, giúp người dùng dễ dàng nhận biết. Các nút bấm được thêm hiệu ứng vi tương tác (micro-interactions) như đổ bóng khi hover và hiệu ứng gợn sóng khi click.

3.2. Tích hợp Nhận diện khuôn mặt (client/js/faceRecognition.js)

- **Công nghệ:** Tích hợp thành công thư viện **face-api.js**, một thư viện JavaScript mạnh mẽ cho nhận diện khuôn mặt trên trình duyệt.
- **Luồng hoạt động:**
 1. **Tải mô hình:** Các mô hình AI được tải một lần duy nhất từ CDN để tối ưu tốc độ. Trạng thái tải được quản lý để tránh việc tải lại không cần thiết.
 2. **Kích hoạt Webcam:** Khi người dùng chọn đăng nhập/đăng ký bằng khuôn mặt, một giao diện video sẽ được kích hoạt.
 3. **Hướng dẫn người dùng:** Giao diện hiển thị các thông báo trạng thái rõ ràng ("Đang tải tài nguyên...", "Hãy nhìn thẳng vào camera...", "Đang xử lý...") để người dùng không cảm thấy bối rối.
 4. **Xử lý và Gửi dữ liệu:** Script sẽ chụp ảnh, trích xuất vector đặc trưng của khuôn mặt (face descriptor) và gửi đến API của backend để lưu trữ hoặc so sánh.
- **Điểm mạnh:** Chức năng này không chỉ mang lại sự tiện lợi (đăng nhập không cần mật khẩu) mà còn là một minh chứng cho khả năng ứng dụng các công nghệ AI tiên tiến vào sản phẩm.

4. Xây dựng Giao diện Chatbot AI (Thư mục chat/)

- Thiết kế:** Giao diện chatbot được thiết kế dưới dạng một cửa sổ popup ở góc dưới bên phải màn hình, không làm gián đoạn trải nghiệm duyệt web của người dùng.
- Giao diện hội thoại:** Thiết kế các bong bóng chat phân biệt rõ ràng giữa tin nhắn của người dùng và của chatbot (AI).
- Trạng thái:** Bổ sung các chỉ báo trạng thái như "VistaNest is typing..." để người dùng biết rằng hệ thống đang xử lý câu hỏi của họ.
- Tích hợp:** Giao diện này được xây dựng để sẵn sàng kết nối với **Python (Flask/FastAPI) service** ở backend, nơi sẽ xử lý logic giao tiếp với **Google Gemini API**.

III - Xây dựng Logic phía Máy chủ (Backend)

1. Tổng quan Kiến trúc Backend

Giai đoạn 3 tập trung vào việc xây dựng một hệ thống backend (máy chủ) mạnh mẽ, linh hoạt và có khả năng mở rộng, đóng vai trò là tầng trung gian xử lý logic và giao tiếp giữa giao diện người dùng (Frontend) và cơ sở dữ liệu (Database). Kiến trúc được lựa chọn là **Kiến trúc hướng Dịch vụ (Service-Oriented Architecture - SOA)**, cụ thể là một mô hình lai giữa **Monolith** và **Microservices**, trong đó:

- Dịch vụ Logic Nghiệp vụ chính (Core Service):** Được xây dựng trên nền tảng **Node.js** và **Express.js**, chịu trách nhiệm xử lý phần lớn các nghiệp vụ cốt lõi như quản lý người dùng, sản phẩm, đơn hàng (CRUD), và xác thực.
- Dịch vụ Trí tuệ Nhân tạo (AI Service):** Một micro-service độc lập được xây dựng bằng **Python** và **Flask**, chuyên trách xử lý các tác vụ liên quan đến AI, cụ thể là giao tiếp với **Google Gemini API** cho chức năng chatbot.

Việc phân tách này cho phép tối ưu hóa công nghệ cho từng tác vụ cụ thể và đảm bảo khả năng mở rộng độc lập cho từng dịch vụ trong tương lai.

2. Thiết kế và Triển khai Lược đồ Cơ sở dữ liệu (Database Schema)

Cơ sở dữ liệu được lựa chọn là **MySQL**, một Hệ quản trị Cơ sở dữ liệu Quan hệ (RDBMS) tiêu chuẩn, nhằm đảm bảo tính toàn vẹn, nhất quán và khả năng truy vấn phức tạp của dữ liệu. Lược đồ được thiết kế chuẩn hóa để giảm thiểu dư thừa dữ liệu và tối ưu hóa các mối quan hệ.

- **Tên CSDL:** tourdb
- **Bảng mã (Character Set):** utf8mb4 với đối chiếu utf8mb4_unicode_ci để hỗ trợ đầy đủ ký tự Unicode, bao gồm cả tiếng Việt và biểu tượng cảm xúc.
- **Các thực thể (Bảng) chính:**
 - **users:** Lưu trữ thông tin định danh và phân quyền của người dùng.
 - **password_hash** (VARCHAR): Mật khẩu được lưu trữ dưới dạng băm **SHA-256**, một phương pháp băm một chiều để đảm bảo an toàn, thay vì lưu trữ văn bản thuần.
 - **role** (ENUM('admin', 'user')): Sử dụng kiểu ENUM để ràng buộc vai trò của người dùng, đảm bảo tính hợp lệ của dữ liệu phân quyền.
 - **face_descriptor** (TEXT): Lưu trữ vector đặc trưng khuôn mặt dưới dạng chuỗi JSON, phục vụ cho chức năng xác thực sinh trắc học.
 - **tours:** Thực thể trung tâm, lưu trữ toàn bộ thông tin về các sản phẩm tour du lịch.
 - **orders:** Lưu trữ thông tin metadata của một giao dịch, bao gồm thông tin khách hàng và tổng giá trị đơn hàng.
 - **order_items:** Bảng liên kết, hiện thực hóa mối quan hệ **Nhiều-Nhiều (Many-to-Many)** giữa orders và tours. Mỗi bản ghi đại diện cho một loại tour cụ thể trong một đơn hàng.
- **Mối quan hệ và Ràng buộc Toàn vẹn:**
 - **Khóa ngoại (Foreign Keys):** Được thiết lập chặt chẽ giữa các bảng:
 - **orders.user_id -> users.id**
 - **order_items.order_id -> orders.id**
 - **order_items.tour_id -> tours.id**
 - **Hành vi Tham chiếu (Referential Actions):** Sử dụng ON DELETE CASCADE cho khóa ngoại trong order_items để khi một đơn hàng hoặc một tour bị xóa, các chi tiết liên quan cũng tự động được dọn dẹp, đảm bảo không có dữ liệu mồ côi (orphan data).

3. Xây dựng Các Điểm cuối API (API Endpoints)

Hệ thống API được xây dựng theo tiêu chuẩn **RESTful**, sử dụng các phương thức HTTP (GET, POST, PUT, DELETE) một cách ngữ nghĩa để thực hiện các thao tác trên tài nguyên.

3.1. Module Xác thực (/api/auth)

- **POST /signup:** Tiếp nhận thông tin người dùng, thực hiện băm mật khẩu và lưu vào bảng `users`. Triển khai logic kiểm tra sự tồn tại của `email` và `username` để tránh trùng lặp.
- **POST /signin:** Xác thực thông tin đăng nhập bằng cách so sánh `password_hash`. Nếu thành công, API trả về thông tin người dùng, bao gồm cả thuộc tính `role`, để frontend có thể thực hiện logic phân quyền chuyển hướng.
- **GET /get-face-descriptors:** Cung cấp một endpoint chuyên biệt để trả về danh sách `username` và `face_descriptor`, phục vụ cho chức năng đăng nhập bằng nhận diện khuôn mặt ở client.

3.2. Module Quản lý Đơn hàng (/api/orders)

- **POST /:** Đây là endpoint xử lý nghiệp vụ phức tạp nhất. Nó được triển khai bằng **Database Transaction** để đảm bảo tính nguyên tử (Atomicity) của thao tác đặt hàng.
 - **Luồng Transaction:** BEGIN TRANSACTION -> INSERT vào `orders` -> INSERT vào `order_items` -> COMMIT. Nếu bất kỳ bước nào thất bại, toàn bộ giao dịch sẽ được ROLLBACK, đảm bảo CSDL không ở trạng thái không nhất quán.
- **GET /:** Lấy danh sách toàn bộ đơn hàng, phục vụ cho trang quản trị.
- **PUT /:id/status:** Cho phép admin cập nhật trạng thái của một đơn hàng (ví dụ: từ 'Chờ xử lý' sang 'Đã phê duyệt').

3.3. Các Module CRUD khác (/api/customers, /api/products)

- Các module này cung cấp đầy đủ các API theo chuẩn CRUD để trang quản trị có thể tương tác với CSDL, cho phép admin quản lý toàn diện các tài nguyên khách hàng và tour du lịch.

4. Dịch vụ Chatbot AI (/api/chat)

- **Kiến trúc Micro-service:** Một service độc lập được xây dựng bằng **Python** và **Flask** được lựa chọn do hệ sinh thái AI/ML vượt trội của Python. Service này lắng nghe trên một cổng riêng (ví dụ: 5000).
- **Luồng xử lý:**
 1. Frontend gửi câu hỏi của người dùng đến endpoint `/api/chat` trên Node.js server.
 2. Node.js server đóng vai trò như một **API Gateway**, gọi đến Python service tại <http://localhost:5000/ask>.
 3. Python service nhận câu hỏi, định dạng lại theo yêu cầu của **Google Gemini API**, sau đó gửi yêu cầu đến Google AI Studio.

- Sau khi nhận được phản hồi từ Gemini, Python service xử lý và trả kết quả về cho Node.js server.
 - Node.js server chuyển tiếp câu trả lời cuối cùng về cho Frontend.
- Lợi ích của kiến trúc này:** Giúp tách biệt logic AI phức tạp ra khỏi ứng dụng chính, dễ dàng nâng cấp hoặc thay thế mô hình AI trong tương lai mà không ảnh hưởng đến các phần khác của hệ thống.
-

IV - Tích hợp và Hoàn thiện Trang Quản trị (Admin)

1. Tổng quan và Kiến trúc Trang Quản trị

Giai đoạn 4 tập trung vào việc xây dựng một **Ứng dụng Web Đơn trang (Single-Page Application - SPA)** giả lập cho khu vực quản trị. Mục tiêu là tạo ra một công cụ tập trung, mạnh mẽ cho phép quản trị viên (Admin) giám sát và điều khiển toàn bộ hoạt động của hệ thống VistaNest.

1.1. Kiến trúc Frontend

- Mô hình:** Trang quản trị được xây dựng như một hệ thống client-side độc lập, giao tiếp với máy chủ hoàn toàn thông qua các **API RESTful** đã được định nghĩa ở Giai đoạn 3.
- Luồng dữ liệu:** Tuân thủ luồng dữ liệu một chiều:
 - Hiển thị (Read):** JavaScript phía client gọi API GET để lấy dữ liệu (ví dụ: danh sách đơn hàng).
 - Thao tác (Create, Update, Delete):** Giao diện người dùng (ví dụ: form, nút bấm) thu thập dữ liệu và gửi các yêu cầu POST, PUT, DELETE đến server.
 - Cập nhật giao diện:** Sau khi nhận được phản hồi thành công từ server, JavaScript sẽ làm mới (re-render) lại các thành phần trên giao diện mà không cần tải lại toàn bộ trang.

1.2. Thiết kế Giao diện và Trải nghiệm Người dùng (UI/UX)

- Cảm hứng thiết kế:** Giao diện được lấy cảm hứng từ các dashboard hiện đại, áp dụng phong cách "**Glassmorphism**" trong theme tối, tạo cảm giác chiều sâu và công nghệ.
- Bố cục:** Sử dụng layout 2 cột kinh điển:
 - Sidebar (Thanh bên):** Cố định (position: fixed), chứa menu điều hướng chính đến các module chức năng. Việc cho phép cuộn độc lập bên trong thanh menu (overflow-y: auto) đảm bảo khả năng truy cập tất cả các mục ngay cả trên màn hình có chiều cao hạn chế.

- **Main Content (Nội dung chính):** Khu vực làm việc chính, hiển thị nội dung của module được chọn.
- **Tính năng tùy biến:** Tích hợp chức năng chuyển đổi **Theme Sáng/Tối**, cho phép quản trị viên lựa chọn môi trường làm việc ưa thích. Trạng thái theme được lưu vào localStorage để duy trì lựa chọn của người dùng qua các phiên làm việc.

2. Hiện thực hóa Chức năng An ninh và Phân quyền

Bảo mật là yếu tố tiên quyết cho trang quản trị. Một cơ chế "bảo vệ cổng" (Gatekeeping) đã được triển khai ở phía client.

- **Logic bảo vệ:**
 - Khi một trang quản trị bất kỳ được tải, một đoạn script trong file admin.js sẽ được thực thi đầu tiên.
 - Script này kiểm tra localStorage để tìm thông tin của người dùng đã đăng nhập (loggedInUser).
 - Nó sẽ xác thực hai điều kiện: **(a)** người dùng có tồn tại không và **(b)** thuộc tính role của người dùng có phải là 'admin' không.
 - Nếu một trong hai điều kiện không thỏa mãn, người dùng sẽ ngay lập tức bị chuyển hướng về trang đăng nhập, kèm theo một thông báo cảnh báo.
- **Đăng xuất:** Chức năng đăng xuất cho phép admin xóa thông tin phiên đăng nhập khỏi localStorage và quay trở lại trang đăng nhập một cách an toàn.

3. Xây dựng các Module Chức năng Quản trị

Mỗi chức năng quản trị được xây dựng như một module riêng biệt, tương tác với các API tương ứng.

3.1. Bảng điều khiển (admin.html, admin.js)

- **Mục đích:** Cung cấp một cái nhìn tổng quan ("bird's-eye view") về tình hình kinh doanh.
- **Triển khai:**
 - **Thống kê số liệu:** JavaScript gọi API để lấy dữ liệu về đơn hàng và khách hàng, sau đó tính toán và hiển thị các chỉ số quan trọng (KPIs) như: Tổng doanh thu, Doanh thu hôm nay, Tổng số đơn hàng, Số lượng khách hàng.
 - **Trực quan hóa dữ liệu:** Tích hợp thư viện Chart.js để vẽ các biểu đồ:
 - **Biểu đồ cột (Bar Chart):** Biểu diễn doanh thu theo từng tháng, giúp admin dễ dàng nhận ra các xu hướng và mùa cao điểm.
 - **Biểu đồ tròn (Doughnut Chart):** Trình bày các dữ liệu giả lập như tỷ lệ truy cập theo thiết bị, mang tính minh họa cho khả năng mở rộng của dashboard.

3.2. Quản lý Đơn hàng & Doanh thu (revenue.html, revenue.js)

- **Mục đích:** Cho phép admin xem và xử lý các đơn hàng đến từ khách hàng.
- **Triển khai:**
 - **Hiển thị danh sách:** Gọi API GET /api/orders để tải toàn bộ danh sách đơn hàng và hiển thị dưới dạng bảng. Các thông tin chính như ID đơn hàng, tên khách hàng, tổng tiền và trạng thái được trình bày rõ ràng.
 - **Hành động trên đơn hàng:** Mỗi hàng trong bảng có các nút "Phê duyệt" và "Từ chối". Khi click, JavaScript sẽ gọi API PUT /api/orders/:id/status với trạng thái tương ứng.
 - **Cập nhật tức thì:** Sau khi nhận được phản hồi thành công từ API, bảng sẽ được vẽ lại (re-render) để phản ánh ngay lập tức sự thay đổi trạng thái mà không cần tải lại trang.
 - **Tính toán doanh thu:** Doanh thu tổng chỉ được tính dựa trên các đơn hàng có trạng thái "Đã phê duyệt", đảm bảo số liệu chính xác.

3.3. Quản lý Sản phẩm (Tour) (products.html, products.js)

- **Mục đích:** Cung cấp một giao diện CRUD (Create, Read, Update, Delete) hoàn chỉnh cho các tour du lịch.
- **Triển khai:**
 - **Giao diện Modal:** Một cửa sổ popup (Modal) duy nhất được sử dụng cho cả hai tác vụ "Thêm mới" và "Chỉnh sửa", giúp tái sử dụng code và tạo trải nghiệm nhất quán.
 - **Thêm mới:** Khi nhấn nút "Thêm tour mới", modal sẽ hiện ra với các trường trống.
 - **Chỉnh sửa:** Khi nhấn nút "Sửa" trên một tour cụ thể, modal sẽ được điều sẵn dữ liệu của tour đó, cho phép admin chỉnh sửa dễ dàng.
 - **Xử lý Ảnh (Giả lập):** Chức năng "upload" ảnh cho phép admin chọn một file từ máy tính. JavaScript sẽ đọc file này bằng FileReader và hiển thị ảnh xem trước (preview) ngay trên form. Khi lưu, chỉ có **tên file** được lưu lại, yêu cầu admin phải tự copy file vật lý vào thư mục images của dự án.
 - **Xóa:** Chức năng xóa được trang bị một hộp thoại xác nhận (confirm()) để ngăn chặn các hành động xóa nhầm.

3.4. Quản lý Tương tác (reviews.html, support.html)

- **Mục đích:** Xây dựng các giao diện mẫu để quản lý tương tác từ người dùng.
- **Triển khai (Giả lập):**
 - **Quản lý Đánh giá:** Đọc dữ liệu bình luận từ localStorage và hiển thị dưới dạng bảng, cho phép admin xem và xóa các bình luận không phù hợp.

- **Chăm sóc Khách hàng:** Tạo một giao diện chat giả lập với dữ liệu cứng (hard-coded), minh họa cách một hệ thống ticket hỗ trợ có thể hoạt động, cho phép admin xem và trả lời các câu hỏi của khách hàng.

V - Phân tích, Đặc tả và Thiết kế Hệ thống

Phần này trình bày chi tiết quá trình phân tích và thiết kế hệ thống VistaNest theo các phương pháp luận của ngành Kỹ thuật Phần mềm. Mục tiêu là mô hình hóa các yêu cầu chức năng, kiến trúc logic và luồng dữ liệu của hệ thống một cách trực quan và có hệ thống.

5.1. Phân tích Chức năng theo Tác nhân (Functional Analysis by Actor)

Hệ thống được thiết kế để phục vụ ba tác nhân chính, mỗi tác nhân có một tập hợp các chức năng (capabilities) được định nghĩa rõ ràng, đảm bảo sự phân quyền và bảo mật.

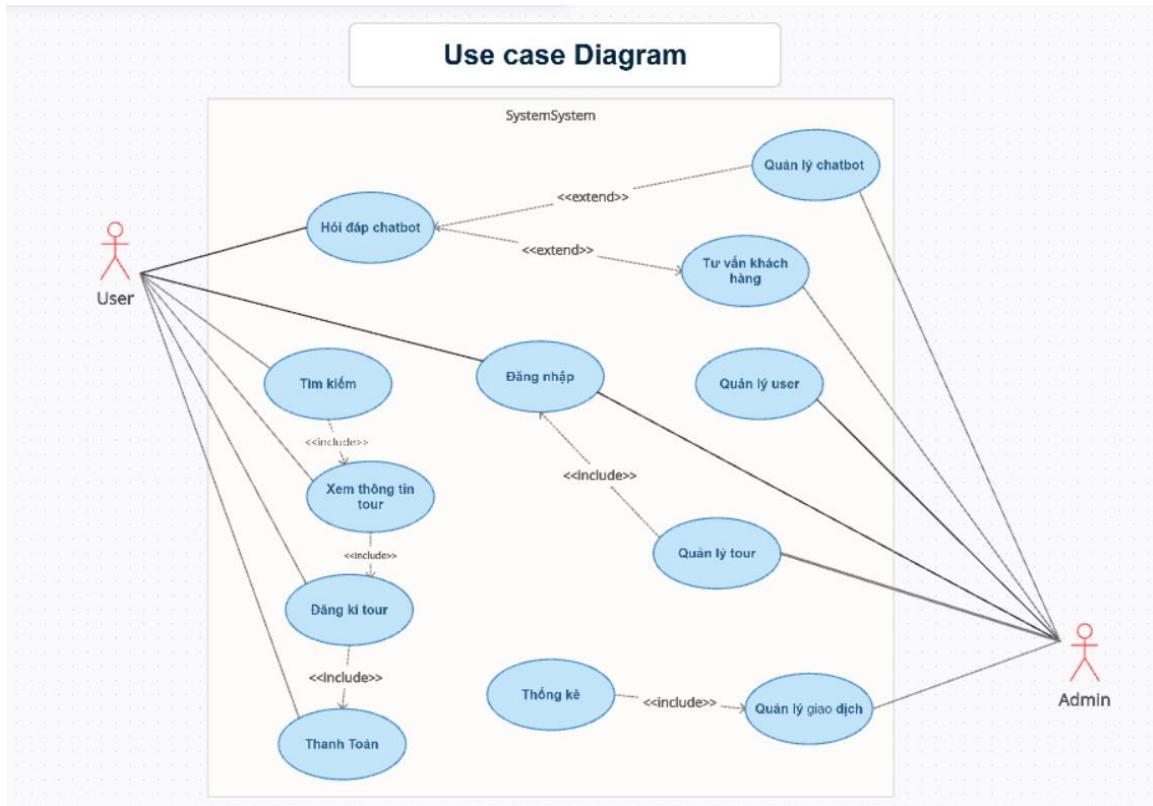
-  **Khách vãng lai (Guest):** Tác nhân chưa được xác thực, có quyền truy cập các tài nguyên công khai.
 - **Chức năng chính:** Tra cứu và xem thông tin chi tiết các tour du lịch; đọc các bài viết tin tức/blog; thực hiện đăng ký tài khoản mới.
-  **Khách hàng (Customer):** Tác nhân đã được xác thực, kế thừa toàn bộ quyền của Guest và được cấp thêm các quyền tương tác cá nhân hóa.
 - **Chức năng chính:** Quản lý tài khoản (đăng nhập, đăng xuất), quản lý giỏ hàng, thực hiện quy trình đặt tour và thanh toán, thêm sản phẩm vào danh sách yêu thích, gửi đánh giá và bình luận, tương tác với chatbot AI.
-  **Quản trị viên (Admin):** Tác nhân có quyền hạn cao nhất, chịu trách nhiệm vận hành và quản lý toàn bộ hệ thống.
 - **Chức năng chính:** Truy cập vào một khu vực quản trị riêng biệt; thực hiện đầy đủ các thao tác CRUD (Create, Read, Update, Delete) trên các thực thể dữ liệu (Tours, Users, Orders); giám sát hoạt động kinh doanh qua Dashboard và báo cáo; xử lý các yêu cầu hỗ trợ từ khách hàng.

5.2. Đặc tả và Thiết kế chi tiết

a. Biểu đồ Use Case và Mô tả

Biểu đồ Use Case được sử dụng để mô hình hóa các yêu cầu chức năng cốt lõi và sự tương tác giữa các tác nhân với hệ thống.

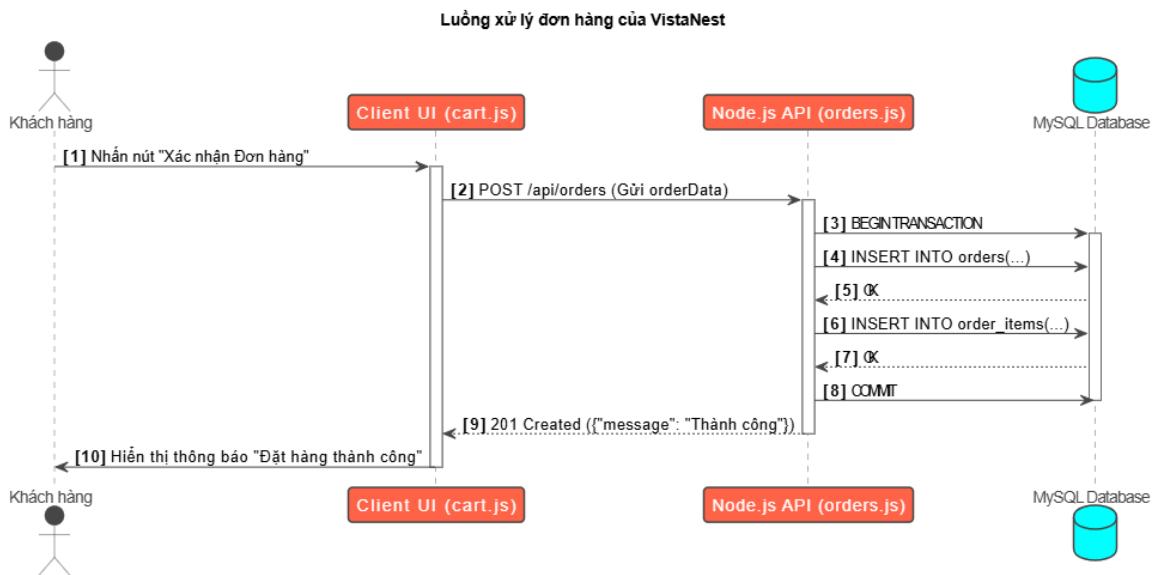
- Biểu đồ Use Case tổng quát:



b. Biểu đồ Tuần tự (Sequence Diagram)

Để làm rõ luồng tương tác theo thời gian của Use Case UC-03, Biểu đồ Tuần tự được sử dụng. Biểu đồ này mô tả chi tiết các thông điệp được trao đổi giữa các đối tượng từ Client, Server đến Database, bao gồm cả các luồng xử lý lõi.

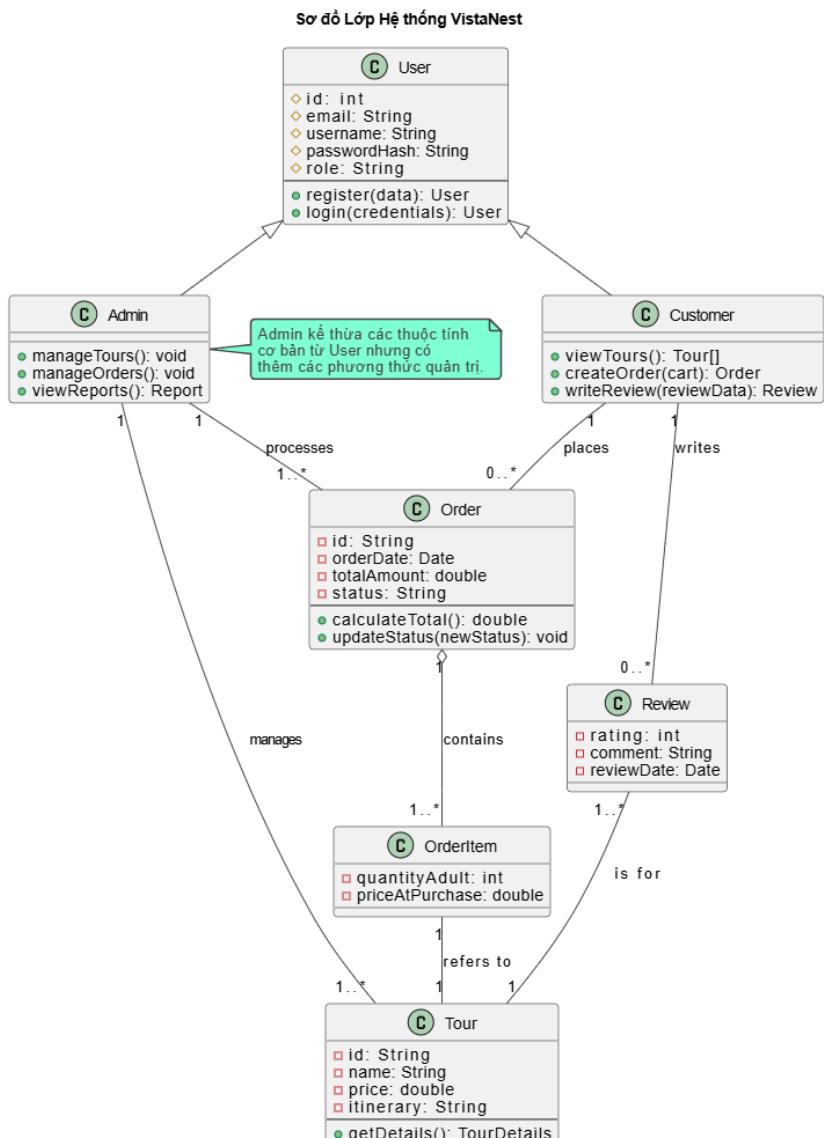
- Biểu đồ Tuần tự cho luồng "Xử lý Đơn hàng":



c. Thiết kế Hướng đối tượng (Class Diagram)

Mô hình lớp được thiết kế để biểu diễn cấu trúc tĩnh của hệ thống, xác định các lớp đối tượng chính, thuộc tính, phương thức và các mối quan hệ (kết thừa, liên kết, hợp thành) giữa chúng.

- Biểu đồ Lớp của hệ thống VistaNest:



d. Luồng Dữ liệu (Data Flow Diagram - DFD) và Cơ sở dữ liệu

- Luồng dữ liệu:** DFD Cấp 0 mô tả hệ thống như một "hộp đen", minh họa luồng dữ liệu vào/ra giữa VistaNest và các thực thể bên ngoài (Khách hàng, Admin, Google AI). Dữ liệu đăng ký, đặt hàng đi vào hệ thống; thông tin tour, xác nhận đơn hàng đi ra hệ thống.
- Thiết kế Cơ sở dữ liệu:** Cấu trúc chi tiết của các bảng users, tours, orders, order_items cùng các ràng buộc khóa ngoại đã được định nghĩa và triển khai trong **Giai đoạn 3 (Mục 3.2)**, đảm bảo tính toàn vẹn và nhất quán của dữ liệu.

e. Luồng Giao diện (UI Flow)

UI Flow mô tả hành trình tuần tự của người dùng qua các màn hình giao diện để hoàn thành một mục tiêu cụ thể. Điều này giúp đảm bảo một trải nghiệm người dùng logic và liền mạch.

- **Ví dụ: Luồng Đặt Tour của Khách hàng**

Trang chủ → Trang danh sách tour (sau khi tìm kiếm/click) → Trang chi tiết tour (chọn một tour) → (Nhấn nút "Thêm vào giỏ hàng") → Trang Giỏ hàng (cart.html) → (Nhấn "Thanh toán") → Modal/Popup Thanh toán → (Nhấn "Xác nhận") → Thông báo Thành công → Trang chủ.

VI - 🔒 Bảo mật và Middleware

An ninh và bảo mật là một khía cạnh phi chức năng tối quan trọng. Dự án VistaNest đã áp dụng các biện pháp cơ bản và định hướng các giải pháp nâng cao để bảo vệ hệ thống.

F.1. Middleware đã triển khai

Middleware trong Express.js đóng vai trò là các "trạm kiểm soát" cho mọi yêu cầu HTTP đi vào server.

- **cors (Cross-Origin Resource Sharing):**

- **Mục đích:** Cho phép trình duyệt ở một domain (ví dụ: <http://127.0.0.1:8080>) có thể gửi yêu cầu đến server ở một domain khác (<http://localhost:3000>). Nếu không có middleware này, trình duyệt sẽ chặn các yêu cầu do chính sách Same-Origin Policy.
- **Triển khai:** app.use(cors(corsOptions)) được đặt ở đầu file server/index.js để áp dụng cho tất cả các routes, với cấu hình cho phép các phương thức GET, POST, PUT, DELETE.

- **express.json():**

- **Mục đích:** Phân tích (parse) các yêu cầu có Content-Type: application/json. Nó chuyển đổi chuỗi JSON trong body của yêu cầu thành một đối tượng JavaScript, giúp việc truy cập req.body trở nên dễ dàng.
- **Triển khai:** app.use(express.json()) được sử dụng để xử lý dữ liệu gửi từ các form phía client.

F.2. Các biện pháp Bảo mật đã áp dụng

- **Băm Mật khẩu (Password Hashing):**

- **Vấn đề:** Lưu mật khẩu dưới dạng văn bản thuần là một rủi ro bảo mật nghiêm trọng.

- **Giải pháp:** Sử dụng thuật toán băm một chiều **SHA-256** (thông qua module `crypto` của Node.js) để chuyển mật khẩu thành một chuỗi hash không thể đảo ngược trước khi lưu vào CSDL. Khi đăng nhập, hệ thống sẽ băm mật khẩu người dùng nhập vào và so sánh chuỗi hash đó với chuỗi hash trong CSDL.
- **Phòng chống SQL Injection:**
 - **Vấn đề:** Kẻ tấn công có thể chèn các đoạn mã SQL độc hại vào các ô nhập liệu để thao túng hoặc phá hủy CSDL.
 - **Giải pháp:** Sử dụng **Prepared Statements** (câu lệnh chuẩn bị sẵn) thông qua thư viện `mysql2`. Thay vì nối chuỗi để tạo câu lệnh SQL, chúng ta sử dụng các dấu chấm hỏi ? làm placeholder. Thư viện sẽ tự động "thoát" (escape) các ký tự đặc biệt trong dữ liệu người dùng gửi lên, vô hiệu hóa mọi nỗ lực SQL Injection.
 - Ví dụ: `db.query('SELECT * FROM users WHERE username = ?', [username])`

F.3. Hướng phát triển Bảo mật trong tương lai

- **Xác thực API bằng JWT (JSON Web Token):**
 - **Mục tiêu:** Đảm bảo rằng chỉ những người dùng đã đăng nhập và có quyền hợp lệ mới có thể gọi đến các API nhạy cảm (ví dụ: xem đơn hàng, cập nhật thông tin).
 - **Luồng hoạt động:** Sau khi đăng nhập thành công, server sẽ tạo và ký một JWT chứa `userId` và `role`, sau đó gửi về cho client. Client sẽ lưu token này và gửi nó kèm trong header `Authorization: Bearer <token>` của mỗi yêu cầu tiếp theo. Server sẽ có một middleware chuyên dụng để xác thực token này trước khi xử lý API.
- **Rate Limiting và Helmet.js:**
 - **express-rate-limit:** Ngăn chặn các cuộc tấn công Brute Force (đò mật khẩu) bằng cách giới hạn số lần một địa chỉ IP có thể gọi đến API đăng nhập trong một khoảng thời gian nhất định.
 - **Helmet.js:** Một tập hợp các middleware giúp bảo vệ ứng dụng Express bằng cách thiết lập các HTTP header bảo mật khác nhau (ví dụ: X-XSS-Protection, Strict-Transport-Security).
- **Validation Dữ liệu Đầu vào:** Sử dụng các thư viện như `express-validator` hoặc `Joi` để kiểm tra và xác thực mọi dữ liệu đến từ `req.body` một cách nghiêm ngặt, đảm bảo dữ liệu luôn đúng định dạng và kiểu trước khi được xử lý

VII - Demo Giao diện và Video

Phần này nhằm mục đích cung cấp một cái nhìn trực quan và sinh động về sản phẩm cuối cùng. Thay vì mô tả dài dòng, việc trình diễn trực tiếp sẽ giúp người xem nhanh chóng nắm bắt được quy mô và chất lượng của dự án.

1.1. Video Demo Tổng quan

Một video ngắn (khoảng 3-5 phút) được dựng lại để trình diễn các luồng chức năng chính của hệ thống. Video sẽ bao gồm:

- **Luồng người dùng:**
 - Trải nghiệm duyệt web, tìm kiếm và xem chi tiết tour.
 - Quá trình đăng ký và đăng nhập bằng tài khoản.
 - Trình diễn tính năng đăng nhập bằng **Nhận diện khuôn mặt**.
 - Quy trình thêm tour vào giỏ hàng và thực hiện thanh toán (giả lập).
- **Luồng quản trị viên:**
 - Đăng nhập vào trang Admin và tổng quan Dashboard.
 - Thao tác quản lý một sản phẩm (Thêm/Sửa/Xóa tour).
 - Thao tác quản lý một đơn hàng (Phê duyệt).
- **Link video:** [Đán link Youtube/Google Drive của bạn vào đây]

1.2. Hình ảnh Giao diện Tiêu biểu

Tuyển tập các ảnh chụp màn hình chất lượng cao, thể hiện các giao diện cốt lõi và ấn tượng nhất của dự án.

- **Giao diện Người dùng (Client-side):**
 - Hình ảnh trang giới thiệu.

ADVENTURE IS WORTHWHILE

Discover New Places With Us, Adventure Awaits

[Discover More](#)

GALLERY



Register Here

Name Email-Id

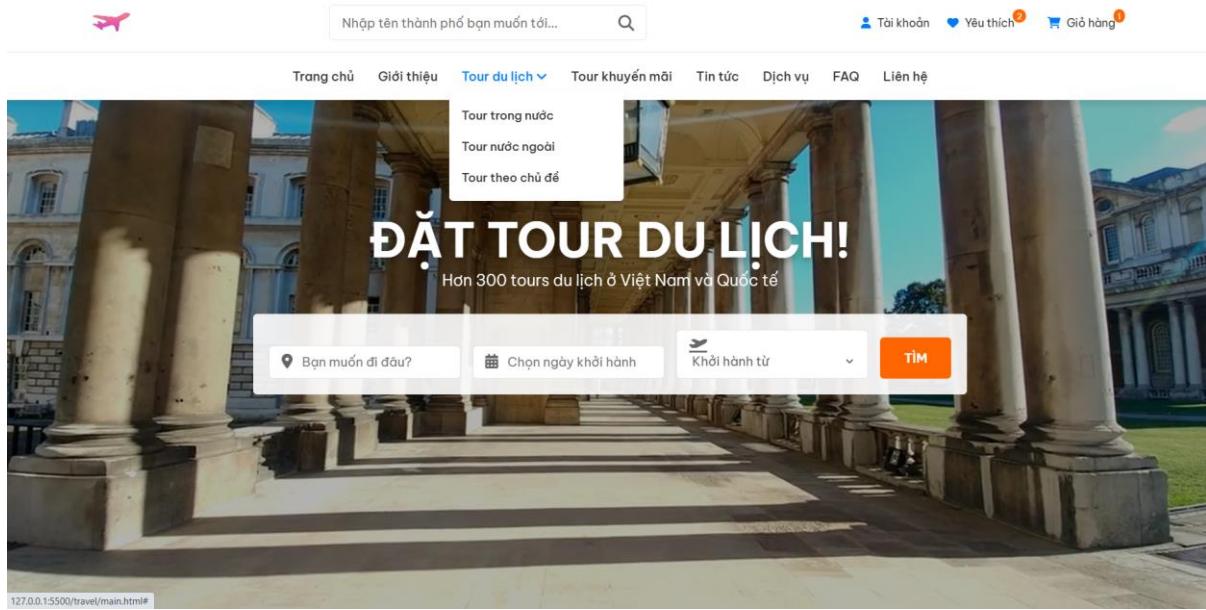
Phone No. Age

Gender
 Male Female

Departure dd/mm/yyyy --:-- --

Return dd/mm/yyyy --:-- --

- Hình ảnh Trang chủ với Hero Banner video.



- Hình ảnh Trang danh sách tour với bộ lọc và phân trang.

Nhập tên thành phố bạn muốn tới..

Tài khoản Yêu thích Giỏ hàng

Trang chủ Giới thiệu Tour du lịch ▾ Tour khuyến mãi Tin tức Dịch vụ FAQ Liên hệ

Bạn muốn đi đâu? Chọn ngày khởi hành Khởi hành từ TÌM

Xếp theo: Mặc định Tên A-Z Tên Z-A Giá tăng dần Giá giảm dần

Du lịch Canada - Cuba [Vancouver - Victoria]
Lịch khởi hành: Thứ 3 hàng tuần
Thời gian: 11 ngày 10 đêm
120.000.000đ **95.000.000đ** Đặt tour

Du lịch Anh - Scotland [Lễ hội hoa Chelsea]
Lịch khởi hành: Thứ 4 hàng tháng 5
Thời gian: 8 ngày 7 đêm
89.990.000đ Đặt tour

Du lịch Châu Âu Pháp - Thụy Sỹ - Núi Jungfrau - Ý
Lịch khởi hành: Thứ 4 hàng tuần
Thời gian: 10 ngày 9 đêm
85.990.000đ Đặt tour

Du lịch Mỹ [Los Angeles - Las Vegas - Universal Studios]
Lịch khởi hành: Thứ 5 hàng tuần
Thời gian: 8 ngày 7 đêm
78.000.000đ Đặt tour

- Hình ảnh Trang chi tiết tour với bố cục 2 cột và khu vực đánh giá.

Nhập tên thành phố bạn muốn tới..

Tài khoản Yêu thích Giỏ hàng

Trang chủ Giới thiệu Tour du lịch ▾ Tour khuyến mãi Tin tức Dịch vụ FAQ Liên hệ

CHƯƠNG TRÌNH TOUR

NGÀY 1-2: JOHANNESBURG - PRETORIA
Bay đến Johannesburg. Tham quan Pretoria - thủ đô hành chính của Nam Phi. Khám phá Gold Reef City - thành phố tái hiện cơn sốt vàng.

NGÀY 3: PILANESBERG SAFARI
Cả ngày khám phá công viên quốc gia Pilanesberg bằng xe chuyên dụng. Cơ hội chiêm ngưỡng Big Five (Sư tử, Báo, Voi, Tê giác, Trâu rừng) trong môi trường tự nhiên.

NGÀY 4: CAPE TOWN - NÚI BẢN
Đáp chuyến bay nội địa đến Cape Town. Lên đỉnh Núi Bản bằng cáp treo xoay 360 độ, ngắm nhìn toàn cảnh thành phố và đại dương.

NGÀY 5: BÁN ĐẢO CAPE
Tham quan bán đảo Cape, du thuyền trên vịnh Hout đến đảo Hải Cẩu, khám phá Mũi Hảo Vọng và bãi biển Boulders với hàng ngàn chú chim cánh cụt.

NGÀY 6-7: TRỞ VỀ VIỆT NAM
Tự do mua sắm trước khi ra sân bay đáp chuyến bay về Việt Nam. Kết thúc hành trình đáng nhớ.

Du lịch Nam Phi [Johannesburg - Pretoria - Soweto - Cape Town]
Giá mới: **61.990.000đ** In chương trình tour

Hành trình: Hà Nội/TP.HCM - Johannesburg
Di chuyển: Máy bay
Lịch khởi hành: Thứ 7 hàng tuần
Thời gian: 7 ngày 6 đêm

✓ Khám phá thế giới hoang dã tại công viên quốc gia Pilanesberg.
✓ Lên đỉnh Núi Bản bằng cáp treo, ngắm toàn cảnh Cape Town.
✓ Tham quan Mũi Hảo Vọng - điểm cực Nam của châu Phi.
✓ Gặp gỡ những chú chim cánh cụt đáng yêu tại bãi

ĐẶT TOUR

Loại khách	Số người	Đơn giá	Tổng giá
Người Lớn	1	61.990.000đ	61.990.000đ
Trẻ em	0	55.000.000đ	0đ
Em bé	0	20.000.000đ	0đ

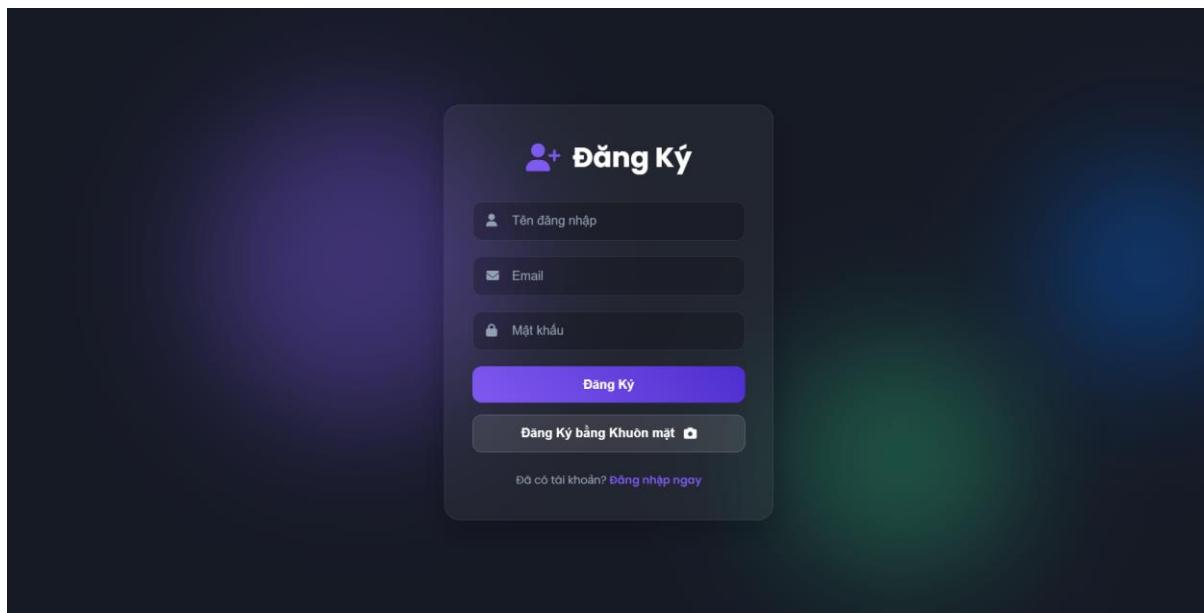
Tổng tiền 61.990.000đ

CÁC TOUR TƯƠNG TỰ

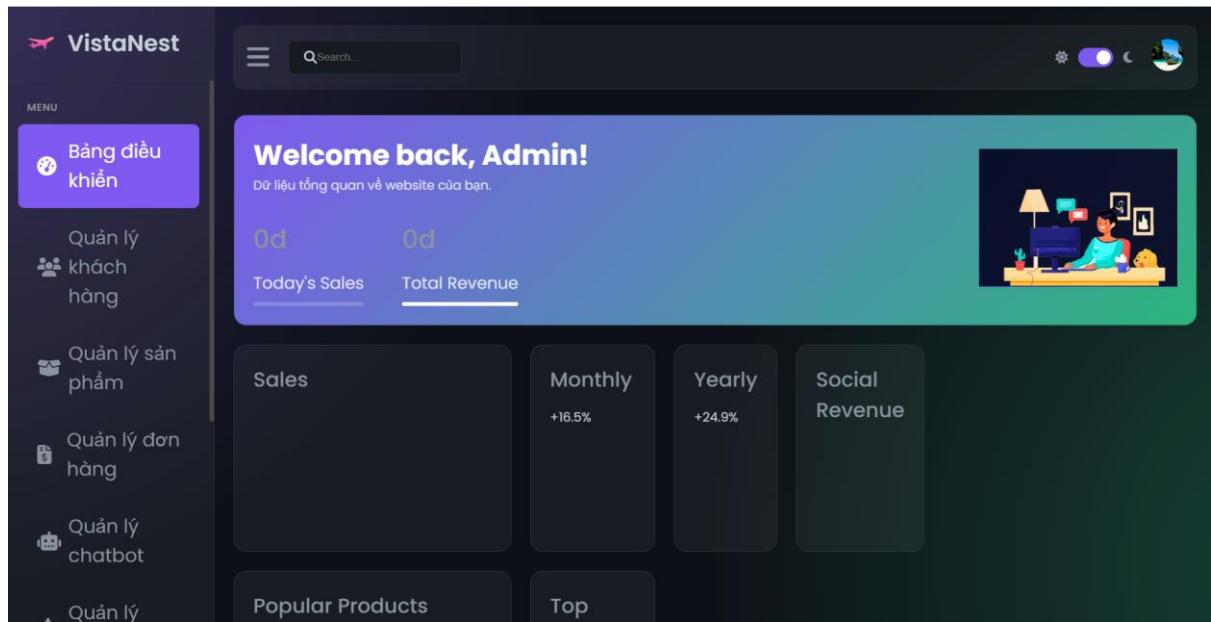
Du lịch Nam Phi [Johannesburg - Pretoria - Soweto - Cape Town]
 Giá mới: **61.990.000đ**

 Hành trình: Hà Nội/TP.HCM - Johannesburg
 Di chuyển: Máy bay
 Lịch khởi hành: Thứ 7 hàng tuần
 Thời gian: 7 ngày 6 đêm
 Khám phá thế giới hoang dã tại công viên quốc gia Pilanesberg.
 Lên đỉnh Núi Bàn bằng cáp treo, ngắm toàn cảnh Cape Town.
 Tham quan Mũi Hảo Vọng - điểm cực Nam của châu Phi.
 Gặp gỡ những chú chim cánh cụt đáng yêu tại bãi

- Hình ảnh Trang đăng nhập/dăng ký với giao diện Glassmorphism.



- **Giao diện Quản trị (Admin-side):**
 - Hình ảnh Dashboard với các biểu đồ (theme Tối).



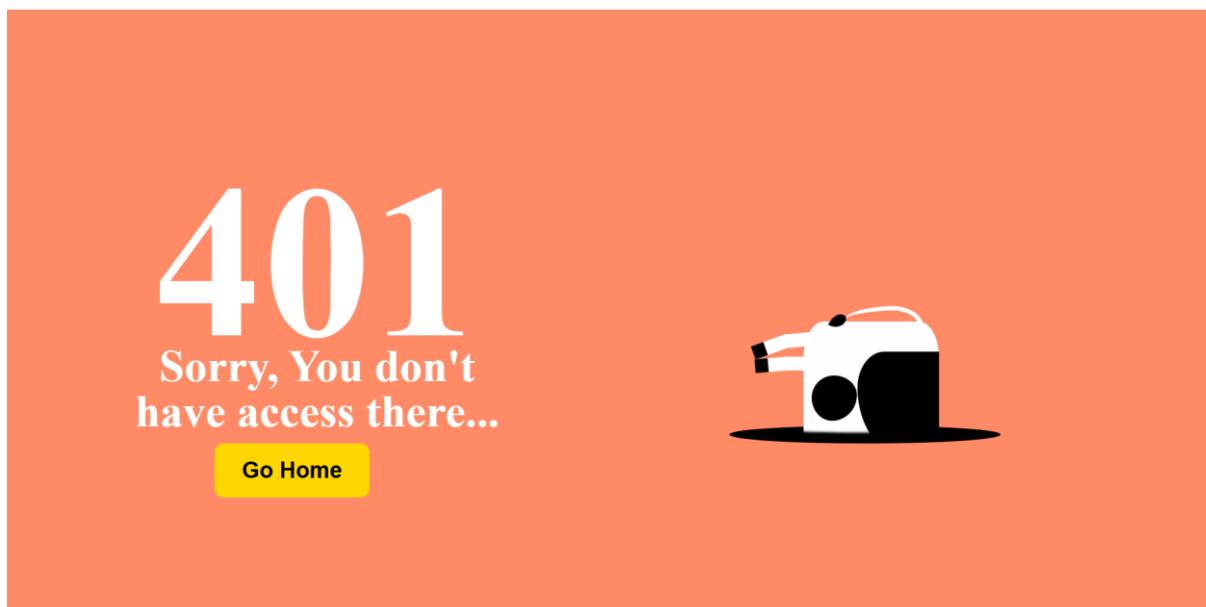
- Hình ảnh trang Quản lý Đơn hàng với các trạng thái khác nhau.

The screenshot shows the 'Danh sách Tour' (Tour List) page. The sidebar menu is identical to the previous one, with 'Quản lý sản phẩm' highlighted. The main content area displays a table of tours:

Ảnh	Tên Tour	Loại Tour	Giá (Người lớn)	Hành động
	Tour Phan Thiết Mui Né Resort 3,4* - 3 ngày 2 đêm	Tour Trong Nước	2.000.000đ	
	Du lịch Đà Nẵng - KDL Bà Nà - Hội An - Cố Đô Huế	Tour Giờ Chót	6.300.000đ	
	Du lịch Phú Quốc Câu cá - Ngắm san hô	Tour Trong Nước	3.500.000đ	

The screenshot shows the VistaNest software interface. On the left, there's a sidebar with a purple header containing the VistaNest logo and a search bar. Below the search bar is a menu with several items: 'Bảng điều khiển' (highlighted in purple), 'Quản lý khách hàng', 'Quản lý sản phẩm', 'Quản lý đơn hàng', 'Quản lý chatbot', and 'Quản lý'. The main area has a dark background. A central window titled 'Yêu cầu Hỗ trợ' (Support Request) lists three entries: 'Trần Văn An' (Về lịch trình có thể thay đổi), 'Lê Thị Bình' (Chào chị Bình, chúng tôi), and 'Phạm Hùng' (Tôi muốn hủy tour Phú Quốc). To the right, a specific message from 'Lê Thị Bình' is shown, with a note at the bottom right: 'Chào chị Bình, chúng tôi đã kiểm tra và gửi lại email xác nhận. Chị vui lòng kiểm tra hộp thư spam nhé.' At the bottom right of the main window is a button labeled 'Gửi trả lời' (Reply). A small note at the bottom of the main window says 'Nhập câu trả lời của bạn...' (Enter your answer...).

- Hình ảnh 401.



VIII - Kiểm thử, Đánh giá và Định hướng Phát triển Tương lai

Sau khi hoàn tất các giai đoạn phát triển chính, dự án đã bước vào giai đoạn kiểm thử tích hợp (Integration Testing) và kiểm thử chấp nhận người dùng (User Acceptance Testing - UAT) nhằm đảm bảo sự ổn định, chính xác và hiệu quả của toàn bộ hệ thống.

1.1. Kịch bản Kiểm thử (Test Cases)

Module	Kịch bản Kiểm thử	Kết quả Dự kiến	Trạng thái
Xác thực	Phân quyền truy cập sau khi đăng nhập (Admin vs. User).	Thành công / Báo lỗi tương ứng.	<input checked="" type="checkbox"/> Đạt
Đặt Tour	Thêm tour vào giỏ hàng, cập nhật giỏ hàng, tiến hành thanh toán.	Đơn hàng được tạo thành công trong CSDL.	<input checked="" type="checkbox"/> Đạt
Quản trị	Thực hiện đầy đủ các thao tác CRUD trên module Tour, Khách hàng.	Dữ liệu trên giao diện và CSDL được cập nhật đồng bộ.	<input checked="" type="checkbox"/> Đạt
Sinh trắc học	Đăng ký và đăng nhập bằng khuôn mặt trong các điều kiện ánh sáng khác nhau.	Nhận diện thành công với độ chính xác chấp nhận được.	<input checked="" type="checkbox"/> Đạt (cần tối ưu)

1.2. Đánh giá Kết quả

- Thành tựu cốt lõi:**
 - Hệ thống hoạt động ổn định:** Toàn bộ luồng từ client đến server và database đã được kết nối thông suốt. Các chức năng chính hoạt động đúng như thiết kế.
 - Kiến trúc vững chắc:** Mô hình Client-Server kết hợp với Service-Oriented Architecture (tách biệt dịch vụ AI) đã chứng tỏ được sự linh hoạt và dễ bảo trì.
 - Trải nghiệm người dùng tích cực:** Giao diện hiện đại, có tính tương tác cao trên cả trang người dùng và trang quản trị đã nhận được phản hồi tốt trong quá trình UAT.
- Hạn chế và Vấn đề còn tồn tại:**
 - Phụ thuộc vào Dữ liệu Giả lập:** Một số module (như quản lý sản phẩm) vẫn đang tương tác với dữ liệu giả lập (database.js) thay vì gọi API đến CSDL, cần được đồng bộ hóa hoàn toàn.
 - Bảo mật API:** Hệ thống hiện tại chưa triển khai cơ chế xác thực dựa trên token (như JWT) cho các yêu cầu API, tiềm ẩn rủi ro về an ninh khi triển khai thực tế.
 - Hiệu năng Nhận diện Khuôn mặt:** Chức năng sinh trắc học hoạt động tốt nhưng có thể có độ trễ khi tải mô hình lần đầu và độ chính xác có thể bị ảnh hưởng bởi điều kiện môi trường.

2. Định hướng Phát triển Tương lai / Tài liệu

Dự án VistaNest đã đặt một nền móng vững chắc. Lộ trình phát triển trong tương lai sẽ tập trung vào việc hoàn thiện, bảo mật và tích hợp các công nghệ đột phá để tạo ra lợi thế cạnh tranh bền vững.

2.1. Hoàn thiện và Tối ưu hóa Hệ thống Hiện tại (Giai đoạn ngắn hạn)

- ⌚ **Đồng bộ hóa Toàn bộ Dữ liệu:** Xây dựng các API còn thiếu cho module quản lý sản phẩm, tin tức... để loại bỏ hoàn toàn sự phụ thuộc vào dữ liệu giả lập phía client.
- 🔒 **Nâng cao Bảo mật với JWT (JSON Web Token):**
 - Luồng hoạt động:** Sau khi người dùng đăng nhập thành công, server sẽ tạo ra một token (JWT) chứa thông tin định danh và vai trò của người dùng, sau đó gửi về cho client.
 - Xác thực API:** Với mọi yêu cầu tiếp theo đến các API cần bảo vệ, client sẽ gửi kèm token này trong header. Server sẽ có một middleware để giải mã và xác thực token trước khi xử lý yêu cầu, đảm bảo chỉ những người dùng hợp lệ mới có thể truy cập tài nguyên.
- ⚡ **Tối ưu hóa Hiệu năng (Performance Optimization):**
 - Phía Frontend:** Áp dụng các kỹ thuật lazy loading cho hình ảnh, gộp và nén (minify) các file CSS và JavaScript để giảm thời gian tải trang.
 - Phía Backend:** Tối ưu hóa các câu lệnh SQL, sử dụng chỉ mục (indexing) cho các cột thường xuyên được truy vấn trong database để tăng tốc độ.

2.2. Tích hợp Công nghệ Thanh toán Sinh trắc học (Giai đoạn trung hạn)

Đây là định hướng phát triển đột phá, biến VistaNest thành một trong những nền tảng đầu tiên ứng dụng thanh toán bằng khuôn mặt trong lĩnh vực du lịch.

- Tầm nhìn:** Cho phép người dùng đã đăng ký khuôn mặt có thể hoàn tất thanh toán chỉ bằng một lần quét mặt, không cần nhập lại thông tin thẻ hay mật khẩu, mang lại trải nghiệm **nhanh chóng, tiện lợi và bảo mật tối đa**.
- Kiến trúc Mã hóa Sinh trắc học:**
 - Băm Vector Khuôn mặt:** Khi người dùng đăng ký khuôn mặt, vector đặc trưng (face descriptor) sẽ không được lưu trực tiếp. Thay vào đó, nó sẽ được đưa qua hàm băm **SHA-256** để tạo ra một chuỗi hash duy nhất, không thể đảo ngược. Chuỗi hash này sẽ được lưu vào CSDL.
 - face_hash = SHA256(face_descriptor)
 - Sinh khóa Mã hóa AES-256:** Chuỗi face_hash này sau đó được sử dụng làm **khóa chính (master key)** để sinh ra một cặp khóa mã hóa **AES-256** (một chuẩn mã hóa đối xứng cực mạnh).

3. **Mã hóa Thông tin Thanh toán:** Khi người dùng lần đầu thêm thông tin thẻ tín dụng, các thông tin nhạy cảm này sẽ được mã hóa bằng khóa AES-256 và lưu vào một bảng riêng trong CSDL. Chỉ có face_hash mới có thể giải mã được thông tin này.
- **Luồng Thanh toán bằng Khuôn mặt:**
 1. Người dùng chọn "Thanh toán bằng khuôn mặt".
 2. Webcam được kích hoạt, hệ thống quét và tạo ra face_descriptor mới.
 3. face_descriptor mới được băm thành face_hash_attempt.
 4. Hệ thống so sánh face_hash_attempt với face_hash đã lưu trong CSDL của người dùng.
 5. **Nếu khớp:** Server sử dụng face_hash này để giải mã thông tin thẻ tín dụng đã lưu, sau đó gửi thông tin đã giải mã đến cổng thanh toán (đã được bảo mật) để xử lý giao dịch.
 6. **Nếu không khớp:** Giao dịch bị từ chối.
 - **Ưu điểm vượt trội:**
 - **Bảo mật hai lớp:** Kẻ gian dù có được CSDL cũng không thể giải mã thông tin thanh toán nếu không có được khuôn mặt của người dùng. Ngược lại, dù có được ảnh khuôn mặt cũng không thể tạo ra đúng face_hash nếu không biết thuật toán băm.
 - **Loại bỏ mật khẩu:** Giảm thiểu rủi ro từ các cuộc tấn công lừa đảo (phishing) hoặc đánh cắp mật khẩu.
 - **Trải nghiệm không ma sát (Frictionless Experience):** Tạo ra một quy trình thanh toán mượt mà và ấn tượng.

2.3. Mở rộng và Triển khai (Giai đoạn dài hạn)

-  **Triển khai lên Cloud (Deployment):** Đưa toàn bộ hệ thống lên các nền tảng đám mây như AWS, Google Cloud, hoặc Vercel (cho Frontend) và Heroku (cho Backend) để có thể truy cập công khai.
-  **Phát triển Ứng dụng Di động:** Xây dựng ứng dụng di động (Native hoặc Cross-platform) để tiếp cận lượng lớn người dùng trên smartphone.
-  **Cá nhân hóa bằng AI:** Sử dụng lịch sử tìm kiếm, các tour đã xem và đã đặt của người dùng để xây dựng một hệ thống gợi ý (Recommendation System), đề xuất các tour phù hợp với sở thích của từng cá nhân, nâng cao tỷ lệ chuyển đổi.

2.4. Tài liệu Tham khảo (References)

- **Node.js & Express.js:**
 - Tài liệu chính thức của Express.js: <https://expressjs.com/>
 - Tài liệu API của Node.js: <https://nodejs.org/api/>
- **Cơ sở dữ liệu MySQL:**
 - Tài liệu thư viện mysql2 cho Node.js: <https://github.com/sidorares/node-mysql2>
- **Trí tuệ Nhân tạo (AI):**

- Tài liệu thư viện face-api.js: <https://github.com/justadudewhohacks/face-api.js>
- Tài liệu Google AI Studio và Gemini API: <https://ai.google.dev/>
- Tài liệu framework Flask (Python): <https://flask.palletsprojects.com/>
- **Thư viện Frontend:**
 - Tài liệu thư viện Chart.js: <https://www.chartjs.org/docs/latest/>
 - Tài liệu thư viện Axios: <https://axios-http.com/docs/intro>

D. 🙏 Lời Cảm ơn

Để hoàn thành dự án "VistaNest" và bản báo cáo tổng kết này, bên cạnh sự nỗ lực của bản thân, chúng tôi đã nhận được rất nhiều sự hỗ trợ, chỉ dẫn và động viên quý báu.

Trước hết, chúng tôi xin được bày tỏ lòng biết ơn sâu sắc và chân thành nhất tới **[Tên Giảng viên hướng dẫn]**, người đã tận tình chỉ bảo, định hướng và cung cấp những kiến thức chuyên môn vô giá trong suốt quá trình chúng tôi thực hiện dự án. Những góp ý sắc bén và kịp thời của thầy/cô không chỉ giúp chúng tôi giải quyết các vấn đề kỹ thuật phức tạp mà còn mở ra những hướng tư duy mới, giúp dự án đạt được chiều sâu và tính ứng dụng cao hơn.

Chúng tôi cũng xin gửi lời cảm ơn tới:

-  **Toàn thể Quý Thầy/Cô** trong Khoa Công Nghệ Thông Tin, trường Công Nghệ Thông Tin - Đại Học Phenikaa đã trang bị cho chúng tôi nền tảng kiến thức vững chắc trong những năm học vừa qua. Đó chính là hành trang quan trọng để chúng tôi có thể tự tin thực hiện dự án này.
-  **Cộng đồng Lập trình viên:** Xin cảm ơn các diễn đàn, các blog công nghệ và các dự án mã nguồn mở như Node.js, Express, MySQL, Chart.js, và đặc biệt là Face-api.js. Những tài nguyên tri thức khổng lồ và miễn phí từ cộng đồng là nguồn tham khảo không thể thiếu, giúp chúng tôi vượt qua nhiều thách thức kỹ thuật. Xin cảm ơn google AI studio, chatgpt và grok.
-  **Gia đình và Bạn bè:** Những người đã luôn ở bên cạnh, động viên về mặt tinh thần, tạo điều kiện tốt nhất để chúng tôi có thể tập trung vào việc học tập và nghiên cứu.

Mặc dù đã rất cố gắng, song do kiến thức và kinh nghiệm thực tiễn còn hạn chế, dự án và bản báo cáo chắc chắn không thể tránh khỏi những thiếu sót. Chúng tôi rất mong tiếp tục nhận được những ý kiến chỉ dẫn từ Quý Thầy/Cô và các bạn để có thể học hỏi và hoàn thiện hơn nữa trong tương lai.

Xin chân thành cảm ơn