TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ**

**MÔN CÔNG NGHỆ .NET**

**MÃ MÔN HỌC: 503112**

*Người hướng dẫn*: **TS. ĐẶNG NGỌC VŨ**

*Người thực hiện*: **NGUYỄN BẢO PHÚC – 52200173**

**NGUYỄN ĐÌNH HOÀNG – 52200178**

**NGUYỄN GIA HUY - 52200181**

*Lớp*: **22050301**

*Khoá****:* 26**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ**

**MÔN CÔNG NGHỆ .NET**

**MÃ MÔN HỌC: 503112**

*Người hướng dẫn*: **TS. ĐẶNG NGỌC VŨ**

*Người thực hiện*: **NGUYỄN BẢO PHÚC – 52200173**

**NGUYỄN ĐÌNH HOÀNG – 52200178**

**NGUYỄN GIA HUY - 52200181**

*Lớp*: **22050301**

*Khoá****:* 26**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025**

# 

# LỜI CẢM ƠN

Lời đầu tiên, nhóm em xin chân thành cảm ơn các thầy, cô giảng viên trong Khoa Công nghệ thông tin Đại học Tôn Đức Thắng đã luôn quan tâm, giúp đỡ sinh viên.

Nhóm em xin gửi lời cảm ơn đến thầy **Đặng Ngọc Vũ**. Trong quá trình học bộ môn công nghệ .NET, chúng em đã nhận được nhiều sự quan tâm giúp đỡ, hướng dẫn và giảng dạy rất tận tình bằng cả tâm huyết của thầy. Nhờ đó mà tụi em có thể tích lũy thêm nhiều kiến thức và vận dụng kiến thức đó vào bài báo cáo giữa kì này.

Nhóm em xin cảm ơn thầy **Đặng Ngọc Vũ** đã giúp chúng em củng cố lại những kiến thức thông qua các buổi lý thuyết và thực hành với các ví dụ thực tiễn, bao quát hết tất cả để chúng em có cái nhìn rộng hơn về công nghệ .net, có thể ứng dụng được những gì học được và tiếp thu vào trong thực tế.

Do chưa có nhiều kinh nghiệm làm báo cáo cũng như những hạn chế về kiến thức và khả năng lý luận của bản thân, trong bài báo cáo này chắc chắn sẽ không tránh khỏi những thiếu sót. Kính mong nhận được lời nhận xét và sự chỉ dẫn của thầy để bài báo cáo của nhóm em có thể hoàn thiện hơn.

Nhóm em xin trân trọng cảm ơn!

# BÁO CÁO ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm báo cáo của riêng tôi và được sự hướng dẫn của **TS**. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 14 tháng 11 năm 2024*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Bảo Phúc*

*Nguyễn Đình Hoàng*

*Nguyễn Gia Huy*

# PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

# TÓM TẮT

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc197908776)

[BÁO CÁO ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG ii](#_Toc197908777)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc197908778)

[TÓM TẮT iv](#_Toc197908779)

[MỤC LỤC v](#_Toc197908780)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ vii](#_Toc197908781)

[CHƯƠNG 1 - GIỚI THIỆU ĐỀ TÀI 1](#_Toc197908782)

[1.1 Mục tiêu đề tài 1](#_Toc197908783)

[1.1.1 Phạm vi chức năng 2](#_Toc197908784)

[1.1.2 Phạm vi kỹ thuật 3](#_Toc197908785)

[1.1.3 Phạm vi đối tượng người dùng 4](#_Toc197908786)

[1.2 Công nghệ sử dụng 4](#_Toc197908787)

[1.2.1 Backend: .NET 8.0 và Entity Framework Core 4](#_Toc197908788)

[1.2.2 Cơ sở dữ liệu: SQL Server 5](#_Toc197908789)

[1.2.3 Frontend: ReactJS 5](#_Toc197908790)

[1.2.4 Bảo mật: JSON Web Tokens (JWT) 6](#_Toc197908791)

[1.2.5 Ghi log: Serilog 6](#_Toc197908792)

[1.2.6 Công cụ phát triển và hỗ trợ 6](#_Toc197908793)

[1.2.7 Các thư viện bổ trợ 7](#_Toc197908794)

[CHƯƠNG 2 - PHÂN TÍCH HỆ THỐNG 9](#_Toc197908795)

[2.1 Khảo sát hệ thống 9](#_Toc197908796)

[2.1.1 Thực trạng 9](#_Toc197908797)

[2.1.2 Mô hình hoạt động của hệ thống 10](#_Toc197908798)

[2.2 Yêu cầu hệ thống 12](#_Toc197908799)

[2.2.1 Yêu cầu chức năng 12](#_Toc197908800)

[2.2.2 Yêu cầu phi chức năng 15](#_Toc197908801)

[2.3 Mô hình ERD (ERD – Entity Relationship Diagram) 16](#_Toc197908802)

[CHƯƠNG 3 - WEBSITE THỰC NGHIỆM 17](#_Toc197908803)

[3.1 17](#_Toc197908804)

[TÀI LIỆU THAM KHẢO 18](#_Toc197908805)

# BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| MSSV-Họ tên | Công việc | Đánh giá |
| 52200178-Nguyễn Đình Hoàng | Code giao diện  Các chức năng liên quan đăng nhập | 100% |
| 52200173-Nguyễn Bảo Phúc | Code phần admin | 100% |
| 52200181-Nguyễn Gia Huy | Code các chức năng liên quan đến tour và đặt tour  Code giao diện | 100% |

# Chương 1: GIỚI THIỆU ĐỀ TÀI

## Mục tiêu đề tài

Đề tài Website đặt tour du lịch được thực hiện nhằm xây dựng một hệ thống trực tuyến hiện đại, hỗ trợ người dùng trong việc tìm kiếm, lựa chọn và đặt tour từ các công ty một cách nhanh chóng, tiện lợi. Các mục tiêu cụ thể của đề tài bao gồm:

* Phát triển một website đầy đủ chức năng dựa trên công nghệ .NET
* Sử dụng .NET 8.0 và Entity Framework Core để xây dựng backend, đảm bảo xử lý logic nghiệp vụ hiệu quả và quản lý dữ liệu an toàn với cơ sở dữ liệu SQL Server.
* Triển khai các tính năng cốt lõi như đăng ký, đăng nhập, xác thực tài khoản bằng JSON Web Tokens (JWT), tìm kiếm tour, phân trang danh sách, và quản lý tour, đáp ứng đầy đủ 16 tiêu chí đánh giá trong rubric của môn học.
* Đảm bảo website hoạt động ổn định, bảo mật cao, và dễ dàng mở rộng trong tương lai.
* Cung cấp trải nghiệm người dùng thân thiện và linh hoạt
* Thiết kế giao diện người dùng (UI) trực quan, thẩm mỹ, sử dụng các công nghệ frontend là React.
* Đảm bảo website hỗ trợ responsive, hoạt động tốt trên cả máy tính và thiết bị di động, mang lại trải nghiệm liền mạch cho người dùng.
* Tích hợp chức năng tìm kiếm thông minh và phân trang, giúp người dùng dễ dàng truy cập danh sách tour và quản lý đơn hàng.
* Xây dựng hệ thống quản trị hiệu quả
* Phát triển trang quản trị (admin page) cho phép quản lý danh sách tour , đơn hàng, và tài khoản người dùng một cách dễ dàng.
* Đảm bảo tính bảo mật và phân quyền, chỉ admin mới có thể truy cập và thực hiện các thao tác quản lý.
* Ứng dụng các kỹ thuật hiện đại và thực tiễn
* Tích hợp ghi log hệ thống bằng Serilog hoặc Nlog để theo dõi và debug các sự kiện quan trọng (đăng nhập, đặt hàng, lỗi hệ thống).
* Áp dụng các kỹ thuật kiểm tra đầu vào (input validation) để đảm bảo dữ liệu người dùng nhập là hợp lệ, tăng cường độ tin cậy của hệ thống.
* Thiết kế cơ sở dữ liệu tối ưu với các bảng và mối quan hệ rõ ràng, đảm bảo tính toàn vẹn và hiệu quả truy xuất dữ liệu.

### Phạm vi chức năng

Website đặt tour được thiết kế để cung cấp các tính năng cốt lõi, đáp ứng nhu cầu của người dùng cuối và quản trị viên, bao gồm:

* Chức năng dành cho người dùng
* Đăng ký và quản lý tài khoản: Người dùng có thể dễ dàng tạo tài khoản mới, xác thực qua email, đăng nhập vào hệ thống, thay đổi mật khẩu, cũng như khôi phục mật khẩu khi quên. Toàn bộ quá trình đều được bảo mật và tối ưu trải nghiệm người dùng.
* Tìm kiếm và lựa chọn tour du lịch: Website hỗ trợ tìm kiếm tour nhanh chóng theo tên, địa điểm, hoặc danh mục du lịch (ví dụ: nghỉ dưỡng, khám phá, văn hóa). Danh sách tour được hiển thị kèm phân trang rõ ràng, giúp người dùng dễ dàng duyệt và lựa chọn theo nhu cầu.
* Đặt tour và theo dõi đơn đặt: Người dùng có thể thêm tour vào giỏ hàng, tiến hành đặt tour một cách thuận tiện và theo dõi tình trạng đơn đặt (chờ xác nhận, đã xác nhận, hoàn thành) ngay trên hệ thống.
* Giao diện thân thiện và linh hoạt: Website được thiết kế hiện đại, trực quan và thân thiện với người dùng. Giao diện responsive giúp trải nghiệm luôn mượt mà trên mọi thiết bị, từ điện thoại thông minh đến máy tính để bàn.
* Chức năng dành cho quản trị viên
* Quản lý nội dung tour: Quản trị viên có thể thêm, chỉnh sửa hoặc xóa thông tin tour du lịch, bao gồm tên tour, điểm đến, giá cả, lịch trình, hình ảnh và danh mục liên quan, nhằm đảm bảo nội dung luôn đầy đủ và chính xác.
* Quản lý đơn đặt tour và người dùng: Hệ thống cho phép quản trị viên theo dõi, cập nhật trạng thái các đơn đặt tour (đang chờ xử lý, đã xác nhận, hoàn tất) và quản lý tài khoản người dùng như khóa hoặc mở khóa khi cần thiết.
* Các tính năng bổ trợ
* Kiểm tra tính hợp lệ của dữ liệu (input validation): Đảm bảo dữ liệu người dùng nhập vào như email, số điện thoại, hoặc số lượng người đăng ký tour đều đúng định dạng và hợp lệ, giúp tăng tính ổn định và bảo mật cho hệ thống.
* Ghi log hệ thống: Sử dụng Serilog để ghi lại các sự kiện quan trọng như đăng nhập, đặt tour, thay đổi trạng thái đơn hàng hoặc các lỗi xảy ra trong quá trình sử dụng, hỗ trợ hiệu quả cho việc kiểm tra, giám sát và bảo trì hệ thống.
* Xác thực và bảo mật truy cập: Áp dụng JWT (JSON Web Tokens) để bảo mật quá trình đăng nhập và truy cập các tài nguyên trên hệ thống, đảm bảo chỉ người dùng đã xác thực mới có thể sử dụng các chức năng phù hợp.

### Phạm vi kỹ thuật

Dự án tập trung vào việc ứng dụng các công nghệ hiện đại nhằm xây dựng một hệ thống web tour du lịch hiệu quả, bảo mật và dễ mở rộng. Cụ thể:

* Backend: Sử dụng nền tảng .NET 8.0 để phát triển các API RESTful, chịu trách nhiệm xử lý logic nghiệp vụ như đặt tour, quản lý người dùng, và cập nhật trạng thái đơn hàng. Entity Framework Core được áp dụng để ánh xạ dữ liệu và tương tác với hệ quản trị cơ sở dữ liệu SQL Server, giúp giảm thiểu mã SQL thủ công và tăng tính linh hoạt trong quản lý dữ liệu.
* Frontend: Giao diện người dùng được xây dựng bằng ReactJS với thiết kế hiện đại, responsive và tối ưu trải nghiệm người dùng. Hệ thống hỗ trợ hiển thị danh sách tour, tìm kiếm, đặt tour, theo dõi đơn đặt và quản lý tài khoản một cách trực quan.
* Cơ sở dữ liệu: Cơ sở dữ liệu được thiết kế với các bảng chính như Users, Tours, Orders, Destinations, và Categories. Cấu trúc dữ liệu đảm bảo tính liên kết hợp lý giữa các bảng, đồng thời đáp ứng yêu cầu toàn vẹn và hiệu suất truy vấn.
* Bảo mật: Hệ thống sử dụng JWT Authentication để bảo vệ các tuyến API, đảm bảo chỉ người dùng hợp lệ mới có quyền truy cập tài nguyên. Ngoài ra, mật khẩu được mã hóa và dữ liệu đầu vào được kiểm tra kỹ lưỡng nhằm ngăn chặn các lỗ hổng bảo mật phổ biến như SQL Injection hay XSS.
* Ghi log: Serilog hoặc NLog được tích hợp để ghi lại các hoạt động quan trọng như đăng nhập, đặt tour, lỗi hệ thống,... giúp quản trị viên dễ dàng theo dõi, phân tích và bảo trì hệ thống.

### Phạm vi đối tượng người dùng

Website tour du lịch được thiết kế nhằm phục vụ hai nhóm đối tượng chính:

* Khách hàng: Là những cá nhân có nhu cầu tìm kiếm và đặt tour du lịch trực tuyến, bao gồm học sinh, sinh viên, nhân viên văn phòng, gia đình, hoặc bất kỳ ai yêu thích khám phá và nghỉ dưỡng. Người dùng chỉ cần thiết bị có kết nối internet (máy tính, điện thoại thông minh, máy tính bảng) và trình duyệt web để truy cập và sử dụng các chức năng của hệ thống.
* Quản trị viên: Là những người chịu trách nhiệm quản lý và vận hành nội dung của website, bao gồm nhân viên của công ty du lịch hoặc quản trị hệ thống. Quản trị viên cần có tài khoản đặc biệt để đăng nhập vào trang quản lý (admin), từ đó thực hiện các thao tác như cập nhật tour, theo dõi đơn đặt, và quản lý người dùng.

## Công nghệ sử dụng

### Backend: .NET 8.0 và Entity Framework Core

* NET 8.0:
* Mô tả: .NET 8.0 là phiên bản mới nhất của nền tảng phát triển phần mềm mã nguồn mở từ Microsoft, cung cấp các tính năng mạnh mẽ để xây dựng ứng dụng web hiệu suất cao.
* Vai trò trong dự án: Được sử dụng để phát triển các API RESTful, xử lý logic nghiệp vụ của website như đăng ký, đăng nhập, đặt hàng, và quản lý dữ liệu.
* Lý do chọn: Đáp ứng yêu cầu bắt buộc của đề bài, cung cấp hiệu suất cao, tích hợp tốt với các thư viện bảo mật và cơ sở dữ liệu, đồng thời hỗ trợ cộng đồng phát triển mạnh mẽ.
* Entity Framework Core (EF Core):
* Mô tả: EF Core là một ORM (Object-Relational Mapping) mã nguồn mở, cho phép ánh xạ dữ liệu giữa cơ sở dữ liệu và các đối tượng trong mã nguồn.
* Vai trò trong dự án: Quản lý tương tác với cơ sở dữ liệu SQL Server, bao gồm việc tạo, đọc, cập nhật, và xóa (CRUD) dữ liệu cho các bảng như Users, Tours, Customers, và Schedules. EF Core giúp đơn giản hóa việc truy vấn dữ liệu và đảm bảo tính toàn vẹn dữ liệu thông qua các mối quan hệ được định nghĩa.
* Lý do chọn: Là công cụ bắt buộc trong đề bài, giúp giảm thiểu mã SQL thủ công, tăng tốc độ phát triển, và hỗ trợ các tính năng như migrations để quản lý lược đồ cơ sở dữ liệu.

### Cơ sở dữ liệu: SQL Server

* Mô tả: Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ, hỗ trợ lưu trữ và quản lý dữ liệu quy mô lớn.
* Vai trò trong dự án: Lưu trữ tất cả dữ liệu của website, bao gồm thông tin người dùng, tour, đơn hàng,... Cơ sở dữ liệu được thiết kế với các bảng và mối quan hệ hợp lý, đảm bảo hiệu quả truy xuất và tính toàn vẹn dữ liệu.
* Lý do chọn: Là yêu cầu bắt buộc của đề bài, SQL Server tích hợp tốt với .NET và EF Core, cung cấp các tính năng như khóa ngoại, ràng buộc, và truy vấn phức tạp để hỗ trợ các chức năng như tìm kiếm và phân trang.

### Frontend: ReactJS

* Mô tả: ReactJS là một thư viện JavaScript phổ biến để xây dựng giao diện người dùng động và tương tác. (Nếu nhóm bạn chọn Razor Pages, Tailwind CSS, AngularJS, hoặc VueJS, hãy thay thế mô tả tương ứng. Ví dụ: Razor Pages là công nghệ frontend tích hợp sẵn trong ASP.NET Core, hỗ trợ xây dựng giao diện server-side).
* Vai nhớ trong dự án: Xây dựng giao diện người dùng cho các trang như trang chủ, danh sách tour, giỏ hàng, và trang quản trị. ReactJS (hoặc công nghệ tương tự) cho phép tạo giao diện responsive, tương tác mượt mà, và tích hợp với backend qua các API RESTful.
* Lý do chọn: Đề bài cho phép tùy chọn công nghệ frontend. ReactJS được chọn vì tính linh hoạt, cộng đồng lớn, và khả năng tạo giao diện hiện đại, hỗ trợ responsive tốt trên các thiết bị.

### Bảo mật: JSON Web Tokens (JWT)

* Mô tả: JWT là một tiêu chuẩn mở để tạo token truy cập, được sử dụng để xác thực và phân quyền trong các ứng dụng web.
* Vai trò trong dự án: Đảm bảo bảo mật cho quá trình đăng nhập và truy cập các tài nguyên được bảo vệ (protected resources) như giỏ hàng hoặc trang quản trị. JWT được tạo khi người dùng đăng nhập thành công và được gửi kèm trong các yêu cầu API để xác minh danh tính.
* Lý do chọn: Là yêu cầu bắt buộc trong rubric, JWT cung cấp cơ chế bảo mật mạnh mẽ, dễ triển khai trong .NET, và phù hợp với các ứng dụng web hiện đại.

### Ghi log: Serilog

* Mô tả: Serilog là một thư viện ghi log mạnh mẽ cho .NET, hỗ trợ cấu hình linh hoạt và lưu trữ log có cấu trúc.
* Vai trò trong dự án: Ghi lại các sự kiện quan trọng như đăng nhập, đăng ký, đặt hàng, hoặc lỗi hệ thống. Log được lưu vào tệp hoặc cơ sở dữ liệu, giúp nhóm dễ dàng theo dõi và debug khi cần.
* Lý do chọn: Đáp ứng yêu cầu ghi log trong rubric, Serilog tích hợp tốt với .NET, cung cấp log có cấu trúc dễ đọc, và hỗ trợ nhiều định dạng đầu ra (file, console, v.v.).

### Công cụ phát triển và hỗ trợ

* Visual Studio 2022
* Mô tả: Môi trường phát triển tích hợp (IDE) chính thức của Microsoft, hỗ trợ lập trình .NET và quản lý dự án.
* Vai trò: Được sử dụng để viết mã, debug, và triển khai ứng dụng. Visual Studio cung cấp các công cụ như IntelliSense, tích hợp Git, và hỗ trợ EF Core migrations.
* Lý do chọn: Là IDE tiêu chuẩn cho phát triển .NET, miễn phí (phiên bản Community) và tích hợp tốt với các công nghệ trong dự án.
* SQL Server Management Studio (SSMS)
* Mô tả: Công cụ quản lý cơ sở dữ liệu SQL Server, hỗ trợ tạo bảng, truy vấn, và quản lý dữ liệu.
* Vai trò: Được sử dụng để thiết kế và kiểm tra cơ sở dữ liệu, chạy các truy vấn SQL, và kiểm tra tính toàn vẹn dữ liệu.
* Lý do chọn: Là công cụ chuẩn để làm việc với SQL Server, dễ sử dụng và miễn phí.
* Postman
* Mô tả: Công cụ kiểm thử API, cho phép gửi yêu cầu HTTP và kiểm tra phản hồi.
* Vai trò: Kiểm tra các API RESTful (đăng ký, đăng nhập, đặt hàng) để đảm bảo hoạt động đúng trước khi tích hợp với frontend.
* Lý do chọn: Dễ sử dụng, hỗ trợ kiểm thử nhanh các API trong quá trình phát triển.

### Các thư viện bổ trợ

* Microsoft.AspNetCore.Authentication.JwtBearer (Phiên bản 8.0.4)
* Mô tả: Thư viện middleware của ASP.NET Core, hỗ trợ xác thực bằng JSON Web Token (JWT) trong các ứng dụng web.
* Vai trò trong dự án: Cho phép hệ thống xác thực người dùng và quản trị viên khi gửi yêu cầu API, cấp và kiểm tra token JWT để bảo vệ các tuyến đường nhạy cảm (như đặt hàng, trang quản trị).
* Lý do chọn: Là yêu cầu trong rubric, thư viện này tích hợp liền mạch với .NET 8.0, đảm bảo bảo mật và tuân thủ tiêu chuẩn OpenID Connect.
* CloudinaryDotNet (Phiên bản 1.27.5)
* Mô tả: Thư viện client chính thức của Cloudinary, hỗ trợ tích hợp dịch vụ quản lý hình ảnh và video dựa trên đám mây.
* Vai trò trong dự án: Được sử dụng để tải lên, quản lý, và tối ưu hóa hình ảnh món ăn, cung cấp URL cho frontend ReactJS hiển thị.
* Lý do chọn: Tích hợp dễ dàng với .NET, hỗ trợ xử lý hình ảnh tự động (nén, thay đổi kích thước), và giảm tải cho server.
* Microsoft.AspNetCore.Identity.EntityFrameworkCore, Microsoft.EntityFrameworkCore.Design, Microsoft.EntityFrameworkCore.SqlServer, Microsoft.EntityFrameworkCore.Tools (Phiên bản 9.0.3-9.0.4)
* Mô tả: Bộ thư viện Entity Framework Core, bao gồm Identity (hỗ trợ xác thực), Design (thành phần thiết kế), SqlServer (hỗ trợ SQL Server), và Tools (công cụ quản lý).
* Vai trò trong dự án: Hỗ trợ quản lý danh tính người dùng (Identity), ánh xạ cơ sở dữ liệu (SqlServer), và thực hiện migrations (Tools) để quản lý lược đồ cơ sở dữ liệu.
* Lý do chọn: Là yêu cầu bắt buộc trong rubric, tích hợp chặt chẽ với .NET 8.0 và SQL Server, đảm bảo hiệu quả quản lý dữ liệu.
* AutoMapper
* Mô tả: Thư viện hỗ trợ ánh xạ tự động giữa các đối tượng trong .NET, đặc biệt hữu ích khi chuyển đổi giữa các lớp Entity và DTO.
* Vai trò trong dự án: Được sử dụng để ánh xạ dữ liệu giữa các lớp domain model (như User, Food, Order) và các lớp Data Transfer Object (DTO), giúp giảm thiểu mã lặp khi xử lý dữ liệu giữa backend và frontend.
* Lý do chọn: Cấu hình đơn giản, dễ sử dụng, giảm thiểu lỗi khi ánh xạ thủ công và tăng tốc độ phát triển API trong môi trường .NET.
* Dapper
* Mô tả: Thư viện ORM nhẹ dành cho .NET, phát triển bởi StackOverflow, cho phép thực thi truy vấn SQL thuần túy một cách nhanh chóng và hiệu quả.
* Vai trò trong dự án: Được sử dụng trong một số thao tác truy vấn tùy chỉnh cần hiệu năng cao hoặc vượt ra ngoài khả năng của Entity Framework, như join nhiều bảng, thống kê đơn hàng, hoặc truy vấn phức tạp.
* Lý do chọn: Hiệu năng rất cao, dễ tích hợp song song với Entity Framework, cho phép viết truy vấn SQL trực tiếp và kiểm soát tốt hơn việc truy xuất dữ liệu.

# Chương 2: PHÂN TÍCH HỆ THỐNG

## Khảo sát hệ thống

### Thực trạng

Trong những năm gần đây, nhu cầu đặt tour du lịch trực tuyến tại Việt Nam và trên thế giới đã tăng trưởng mạnh mẽ, nhờ vào sự bùng nổ của công nghệ số và sự thay đổi trong thói quen tiêu dùng. Các nền tảng như Traveloka, Klook, Booking.com hay Vietravel đã trở thành những ví dụ điển hình, cung cấp dịch vụ đặt tour, khách sạn và vé tham quan một cách nhanh chóng và tiện lợi. Tuy nhiên, thực tế triển khai và nhu cầu của người dùng cho thấy vẫn còn nhiều vấn đề cần cải thiện:

* Ưu điểm của các hệ thống hiện tại:
* Tiện lợi: Người dùng có thể tìm kiếm và đặt tour du lịch trong nước và quốc tế chỉ với vài thao tác đơn giản qua điện thoại hoặc máy tính.
* Đa dạng lựa chọn: Các nền tảng cung cấp hàng loạt gói tour đa dạng từ nhiều công ty du lịch với các chủ đề hấp dẫn như khám phá thiên nhiên, nghỉ dưỡng, văn hóa – lịch sử, v.v.
* Tích hợp công nghệ: Hỗ trợ thanh toán trực tuyến, theo dõi trạng thái đơn đặt tour, và xem đánh giá từ những người dùng trước.
* Giao diện thân thiện: Các ứng dụng và website thường có thiết kế hiện đại, dễ sử dụng và tương thích trên nhiều thiết bị.
* Hạn chế của các hệ thống hiện tại:
* Chi phí dịch vụ cao: Một số nền tảng tính phí hoa hồng cao đối với công ty du lịch, làm tăng giá thành tour cho người dùng cuối.
* Phụ thuộc vào bên thứ ba: Các công ty lữ hành nhỏ thường phải dựa vào nền tảng lớn để tiếp cận khách hàng, khiến họ mất quyền kiểm soát dữ liệu và thương hiệu.
* Vấn đề bảo mật: Một số nền tảng chưa chú trọng đầy đủ đến bảo vệ dữ liệu người dùng, dẫn đến nguy cơ rò rỉ thông tin cá nhân.
* Hiệu suất tìm kiếm và lọc chưa tối ưu: Một số website không cung cấp bộ lọc hiệu quả theo ngân sách, thời gian khởi hành hoặc điểm đến, khiến người dùng mất thời gian tìm kiếm.
* Khó tùy chỉnh thương hiệu: Các công ty du lịch nhỏ không thể cá nhân hóa giao diện hoặc chức năng theo định hướng kinh doanh riêng.
* Nhu cầu thực tế:
* Đối với người dùng: Cần một hệ thống đặt tour đơn giản, chi phí hợp lý, bảo mật cao, giao diện dễ sử dụng và hoạt động tốt trên cả thiết bị di động lẫn máy tính. Tính năng tìm kiếm nhanh và rõ ràng theo tiêu chí như giá tour, lịch trình, hoặc khu vực là rất cần thiết.
* Đối với doanh nghiệp du lịch nhỏ và vừa: Mong muốn có một nền tảng độc lập, dễ triển khai và quản lý, cho phép kiểm soát toàn bộ dữ liệu khách hàng, đơn đặt tour, và nội dung quảng bá tour mà không bị ràng buộc bởi nền tảng trung gian.
* Đối với quản trị viên: Cần một hệ thống quản trị mạnh mẽ, hỗ trợ kiểm soát toàn diện tour, người dùng, và đơn đặt, đồng thời ghi lại nhật ký hệ thống để phục vụ cho việc giám sát và bảo trì.

### Mô hình hoạt động của hệ thống

Mô hình hoạt động của Website đặt tour du lịch được thiết kế để đảm bảo sự tương tác hiệu quả giữa ba thành phần chính: người dùng cuối (khách hàng), quản trị viên, và hệ thống backend. Hệ thống hoạt động theo kiến trúc client-server, với các luồng hoạt động chính được mô tả như sau:

#### Các thành phần chính

* Người dùng cuối (khách hàng): Truy cập website thông qua trình duyệt web trên máy tính hoặc thiết bị di động để thực hiện các chức năng như đăng ký, đăng nhập, tìm kiếm tour du lịch, đặt tour, và theo dõi tình trạng đơn đặt.
* Quản trị viên: Sử dụng giao diện trang quản trị (admin page) để thực hiện các tác vụ như thêm/sửa/xóa tour, quản lý danh mục và điểm đến, theo dõi đơn đặt tour, và kiểm soát tài khoản người dùng.
* Hệ thống backend: Đảm nhiệm việc xử lý các yêu cầu từ phía client (bao gồm người dùng và quản trị viên) thông qua các API RESTful được xây dựng bằng .NET 8.0. Backend sử dụng Entity Framework Core để tương tác với cơ sở dữ liệu SQL Server, phục vụ việc lưu trữ và truy vấn dữ liệu một cách hiệu quả.
* Cơ sở dữ liệu: Lưu trữ các thông tin quan trọng như người dùng, tour du lịch, đơn đặt tour, điểm đến, và danh mục tour. Hệ thống được thiết kế để đảm bảo tính toàn vẹn dữ liệu, hỗ trợ truy xuất nhanh chóng và chính xác.

#### Luồng hoạt động chính

* Đăng ký và xác thực
* Người dùng truy cập website, điền thông tin (email, mật khẩu, họ tên, số điện thoại) để đăng ký tài khoản.
* Hệ thống gửi email xác thực (hoặc OTP) để kích hoạt tài khoản người dùng.
* Sau khi xác thực thành công, người dùng có thể đăng nhập bằng email và mật khẩu. Backend sẽ tạo ra một JWT (JSON Web Token) để xác minh danh tính người dùng trong các yêu cầu tiếp theo.
* Tìm kiếm và đặt tour
* Người dùng sử dụng thanh tìm kiếm để tìm tour theo tên, danh mục, điểm đến hoặc thời gian khởi hành.
* Danh sách tour được hiển thị kèm phân trang (paging), cho phép người dùng dễ dàng duyệt nội dung.
* Người dùng chọn tour, xem chi tiết, thêm vào giỏ đặt tour (nếu có), nhập thông tin liên hệ và xác nhận đặt tour.
* Hệ thống lưu thông tin đơn đặt tour vào cơ sở dữ liệu và gửi thông báo xác nhận qua email hoặc giao diện.
* Quản lý đơn đặt tour
* Người dùng có thể xem lịch sử đặt tour và kiểm tra trạng thái đơn hàng (chờ xác nhận, đã xác nhận, hoàn tất, hủy).
* Quản trị viên nhận thông báo khi có đơn đặt mới, có thể cập nhật trạng thái, xử lý các yêu cầu chỉnh sửa hoặc hủy đơn của khách.
* Quản trị hệ thống
* Quản trị viên đăng nhập vào trang quản trị (admin page), thực hiện các thao tác như thêm/sửa/xóa tour, quản lý danh mục tour và điểm đến, cũng như khóa/mở khóa tài khoản người dùng.
* Giao diện admin được thiết kế trực quan, hỗ trợ thao tác nhanh chóng và được bảo vệ bằng cơ chế phân quyền truy cập dựa trên JWT.
* Ghi log và bảo mật
* Hệ thống sử dụng Serilog để ghi lại các sự kiện quan trọng như đăng nhập, đặt tour, cập nhật đơn hàng hoặc lỗi hệ thống, giúp theo dõi và phục vụ quá trình bảo trì.
* Tất cả các yêu cầu đến API đều được xác thực bằng JWT, đồng thời dữ liệu nhập được kiểm tra kỹ lưỡng (input validation) để phòng tránh các lỗ hổng bảo mật như SQL injection hoặc nhập liệu sai định dạng.

#### Sơ đồ luồng hoạt động

* Người dùng: Gửi yêu cầu HTTP (GET/POST) qua trình duyệt tới API backend.
* Backend: Xử lý yêu cầu, xác thực JWT, tương tác với cơ sở dữ liệu qua EF Core, và trả về phản hồi (JSON).
* Cơ sở dữ liệu: Lưu trữ và cung cấp dữ liệu theo yêu cầu từ backend.
* Quản trị viên: Truy cập trang admin, gửi yêu cầu quản lý nội dung tới backend, nhận phản hồi và cập nhật giao diện.

## Yêu cầu hệ thống

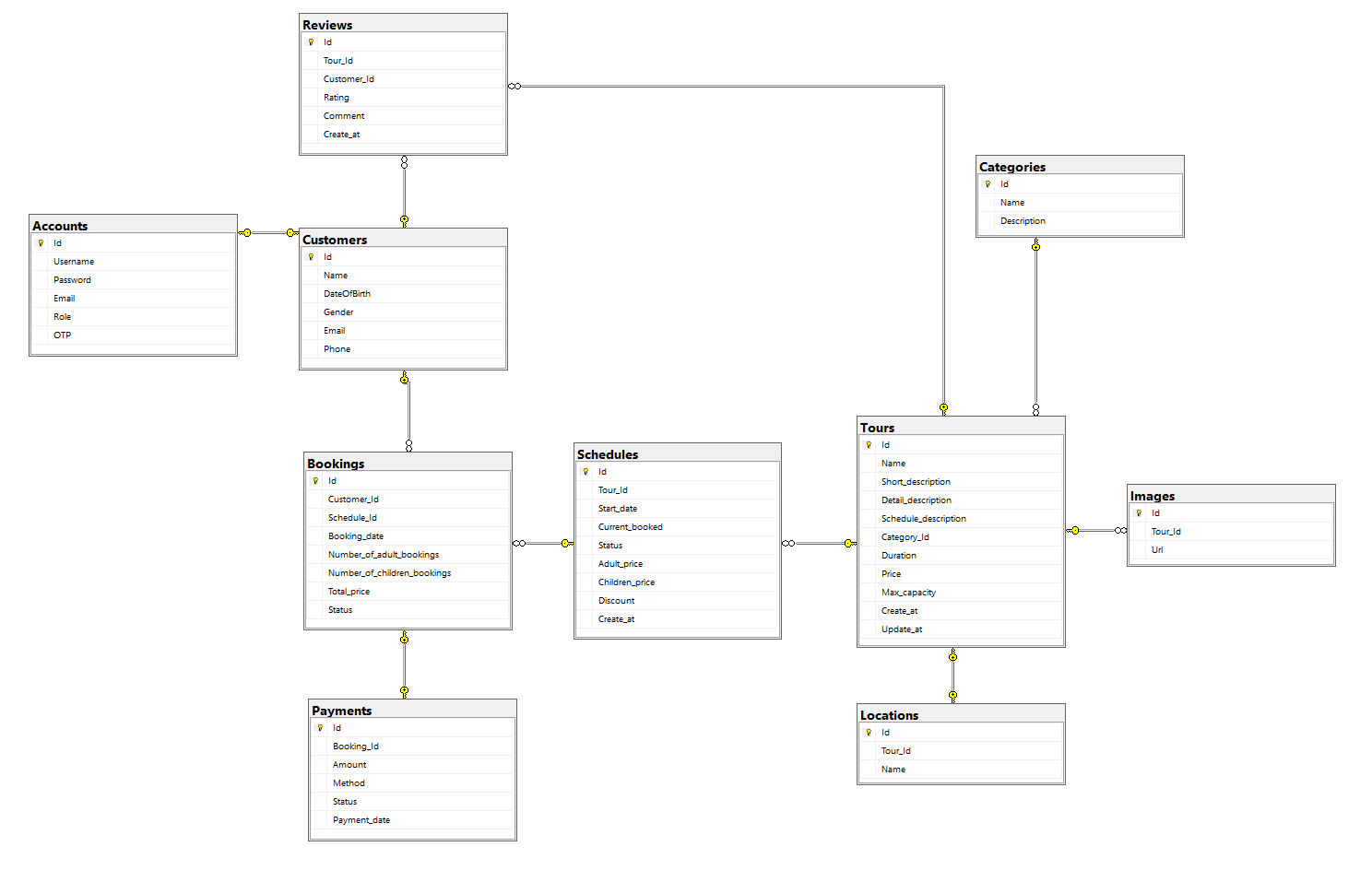
### Yêu cầu chức năng

* Đăng ký tài khoản (Registration)
* Người dùng có thể tạo tài khoản mới bằng cách nhập thông tin như email, mật khẩu, tên, và số điện thoại.
* Hệ thống kiểm tra tính hợp lệ của dữ liệu nhập (ví dụ: email đúng định dạng, mật khẩu đủ độ dài).
* Hệ thống gửi email xác thực để kích hoạt tài khoản.
* Xác thực tài khoản (Account Verification)
* Người dùng nhận email chứa liên kết xác thực hoặc mã OTP sau khi đăng ký.
* Hệ thống xác nhận tài khoản sau khi người dùng nhấp vào liên kết hoặc nhập mã OTP đúng.
* Đăng nhập (Login)
* Người dùng đăng nhập bằng email và mật khẩu.
* Hệ thống xác thực thông tin đăng nhập và cấp JSON Web Token (JWT) để truy cập các tài nguyên được bảo vệ.
* Hiển thị thông báo lỗi nếu thông tin đăng nhập không chính xác.
* Đổi mật khẩu (Change Password)
* Người dùng đã đăng nhập có thể thay đổi mật khẩu trong trang quản lý tài khoản.
* Hệ thống yêu cầu nhập mật khẩu cũ và mật khẩu mới, kiểm tra tính hợp lệ trước khi cập nhật.
* Quên mật khẩu (Forgot Password)
* Người dùng có thể yêu cầu khôi phục mật khẩu bằng cách nhập email.
* Hệ thống gửi email chứa liên kết đặt lại mật khẩu, cho phép người dùng tạo mật khẩu mới.
* Trang quản trị (Admin Page)
* Quản trị viên có thể đăng nhập vào trang quản trị để thực hiện các thao tác quản lý nội dung và người dùng.
* Trang quản trị cung cấp các chức năng chính như sau:
* Thêm, sửa, xóa tour du lịch (bao gồm: tên tour, giá, mô tả, hình ảnh, thời gian khởi hành, địa điểm đến).
* Quản lý danh mục tour (ví dụ: du lịch trong nước, du lịch nước ngoài, tour theo mùa, tour khám phá...).
* Xem và cập nhật trạng thái đơn đặt tour (chờ xác nhận, đã xác nhận, hoàn tất, hủy).
* Quản lý tài khoản người dùng: cho phép khóa hoặc mở khóa tài khoản nếu phát hiện vi phạm hoặc có yêu cầu.
* Giao diện quản trị được thiết kế trực quan, dễ sử dụng và có phân quyền rõ ràng, giúp đảm bảo rằng chỉ những quản trị viên có quyền mới có thể thực hiện các thao tác tương ứng.
* Xác thực bằng JWT (Authentication - JWT)
* Hệ thống sử dụng JWT để xác thực và phân quyền cho các yêu cầu API.
* Token được tạo khi đăng nhập thành công và được gửi kèm trong các yêu cầu đến các tuyến API được bảo vệ (ví dụ: đặt hàng, truy cập trang quản trị).
* Hệ thống kiểm tra tính hợp lệ của token và xử lý lỗi nếu token hết hạn hoặc không hợp lệ.
* Tìm kiếm tour du lịch (Tour Searching)
* Người dùng có thể tìm kiếm tour theo từ khóa (tên tour), danh mục (ví dụ: du lịch biển, khám phá, nghỉ dưỡng...), hoặc địa điểm đến.
* Kết quả trả về sẽ được phân trang để giúp người dùng dễ dàng duyệt qua nhiều lựa chọn.
* Phân trang (Paging)
* Danh sách tour, đơn đặt tour, và kết quả tìm kiếm đều hiển thị dưới dạng phân trang (ví dụ: 10 tour/trang).
* Hệ thống hỗ trợ điều hướng giữa các trang để nâng cao trải nghiệm người dùng.
* Đặt tour (Tour Booking):
* Người dùng chọn tour mong muốn, điền thông tin cá nhân và thông tin liên hệ để đặt tour.
* Hệ thống tính tổng chi phí, xác nhận lại thông tin đặt chỗ, và lưu đơn đặt tour vào cơ sở dữ liệu.
* Người dùng nhận được thông báo xác nhận đặt tour qua hệ thống (hoặc qua email).
* Quản lý đơn đặt tour (Order Management):
* Người dùng có thể theo dõi lịch sử đặt tour, kiểm tra trạng thái tour (chờ xác nhận, đã xác nhận, hoàn tất, đã hủy).
* Quản trị viên có thể cập nhật trạng thái đơn đặt tour, xử lý yêu cầu hủy, và quản lý dữ liệu từ trang quản trị.
* Kiểm tra đầu vào (Input Validation):
* Hệ thống thực hiện kiểm tra định dạng dữ liệu nhập vào (email, số điện thoại, số người đi, ngày khởi hành...) để đảm bảo hợp lệ.
* Các thông báo lỗi cụ thể sẽ được hiển thị để hướng dẫn người dùng sửa lỗi kịp thời.
* Trang chủ (Homepage):
* Hiển thị các tour nổi bật, danh mục phổ biến, hoặc chương trình khuyến mãi hấp dẫn.
* Thiết kế giao diện thẩm mỹ, hiện đại, giúp người dùng dễ dàng tiếp cận thông tin và bắt đầu hành trình đặt tour.
* Điều hướng (Navigation - Routing):
* Website có hệ thống điều hướng rõ ràng, cho phép người dùng di chuyển mượt mà giữa các trang (trang chủ, tour, giỏ hàng, tài khoản, v.v.).
* Tối ưu trải nghiệm người dùng bằng các đường dẫn hợp lý và dễ hiểu.
* Ghi log hệ thống (Logging - Serilog):
* Hệ thống ghi lại các hoạt động quan trọng như đăng nhập, đăng ký, đặt tour, hoặc lỗi phát sinh bằng thư viện Serilog.
* Log có cấu trúc rõ ràng, phục vụ công tác giám sát, bảo trì và phân tích hệ thống hiệu quả.

### Yêu cầu phi chức năng

* Hiệu suất (Performance):
* Hệ thống phản hồi các yêu cầu của người dùng (tìm kiếm, đặt hàng, đăng nhập) trong vòng dưới 1 phút dưới điều kiện tải bình thường.
* Tìm kiếm và phân trang được tối ưu để xử lý danh sách món ăn lớn mà không ảnh hưởng đến tốc độ.
* Bảo mật (Security):
* Thông tin người dùng (email, mật khẩu, số điện thoại) được lưu trữ an toàn, với mật khẩu mã hóa.
* Sử dụng JWT để bảo vệ các API, đảm bảo chỉ người dùng đã xác thực mới truy cập được các tài nguyên nhạy cảm.
* Khả năng sử dụng (Usability):
* Giao diện người dùng trực quan, dễ sử dụng, với bố cục rõ ràng và các nút bấm dễ nhận biết.
* Người dùng không cần đào tạo vẫn có thể sử dụng các chức năng cơ bản (tìm kiếm, đặt hàng) sau khi truy cập.
* Trang quản trị được thiết kế để quản trị viên thực hiện các thao tác quản lý (thêm tour, cập nhật đơn hàng) một cách nhanh chóng và hiệu quả.

## Mô hình ERD (ERD – Entity Relationship Diagram)



Mô hình ERD

# WEBSITE THỰC NGHIỆM

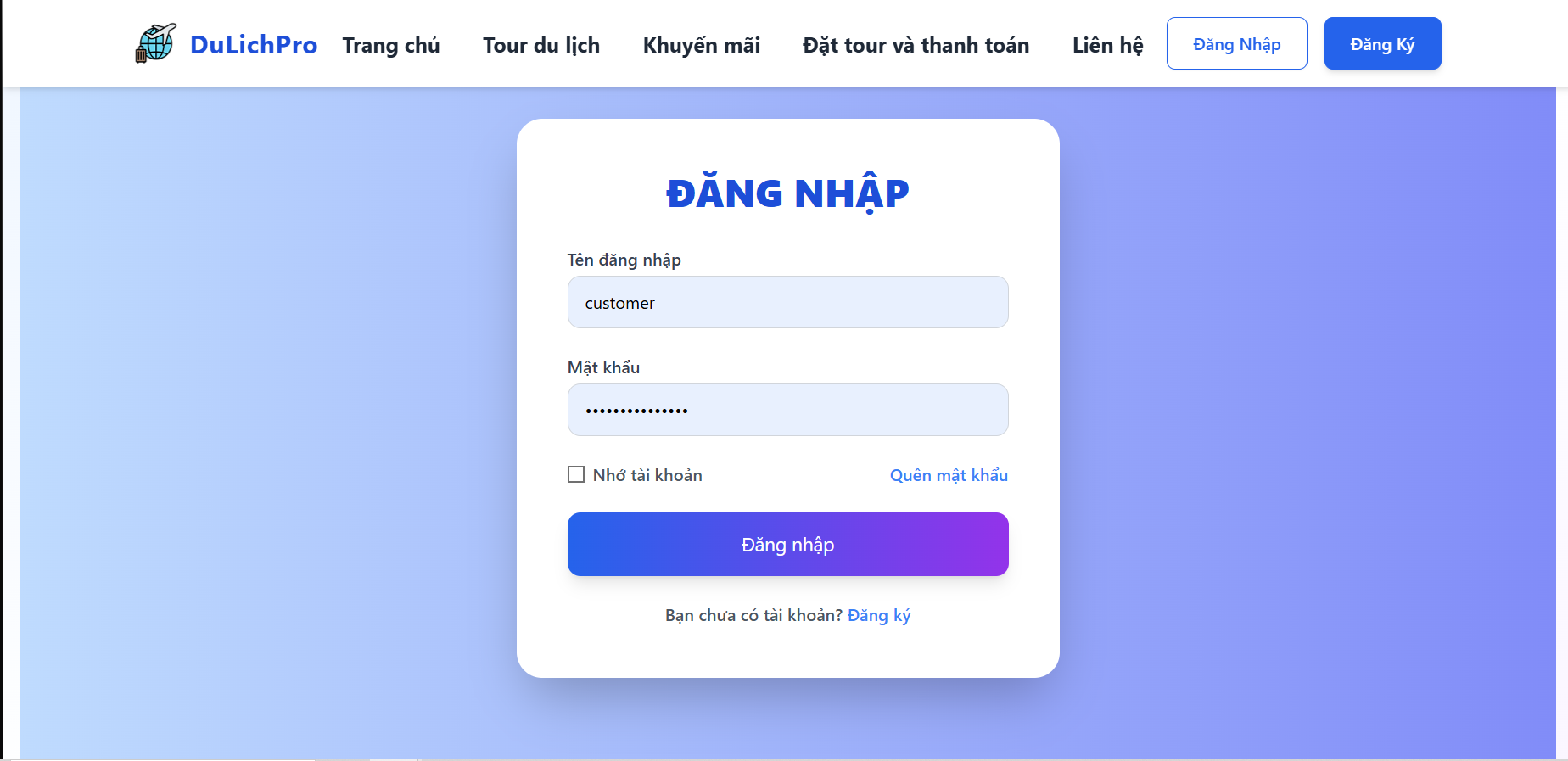
## Sinh seed data:

* Seed data cho các bảng được được sinh bằng mô hình Germini 2.5 dưới dạng json, sau đó dùng code python chuyển object về code để migration vào database.
* Đối với hình ảnh lấy từ pexels, hình ảnh chỉ mang tính chất minh họa và để test chức năng là chính.
* Lưu trữ hình ảnh: Sử dụng dịch vụ cloud cloundinary để lưu trữ hình ảnh cho server.
* Ngoài ra còn có thêm các procedure hỗ trợ:
* CreateBooking: giúp tạo booking
* CreateCustomer: giúp tạo khách hàng
* CreateAccount: giúp tạo account
* CreateReview: giúp tạo review
* CreateTour: giúp tạo tour
* CreateImage: giúp thêm dữ liệu hình ảnh
* CreateSchedule: giúp tạo lịch trình
* UpdateScheduleDiscount: cập nhật giá lịch trình
* CreatePayment: giúp tạo giao dịch.

## Các chức năng chính của khách hàng:

### Đăng nhập:

**Giao diện:**



**Mô tả ngắn:** Dùng để đăng nhập tài khoản của khách hàng.

**Các api sử dụng:**

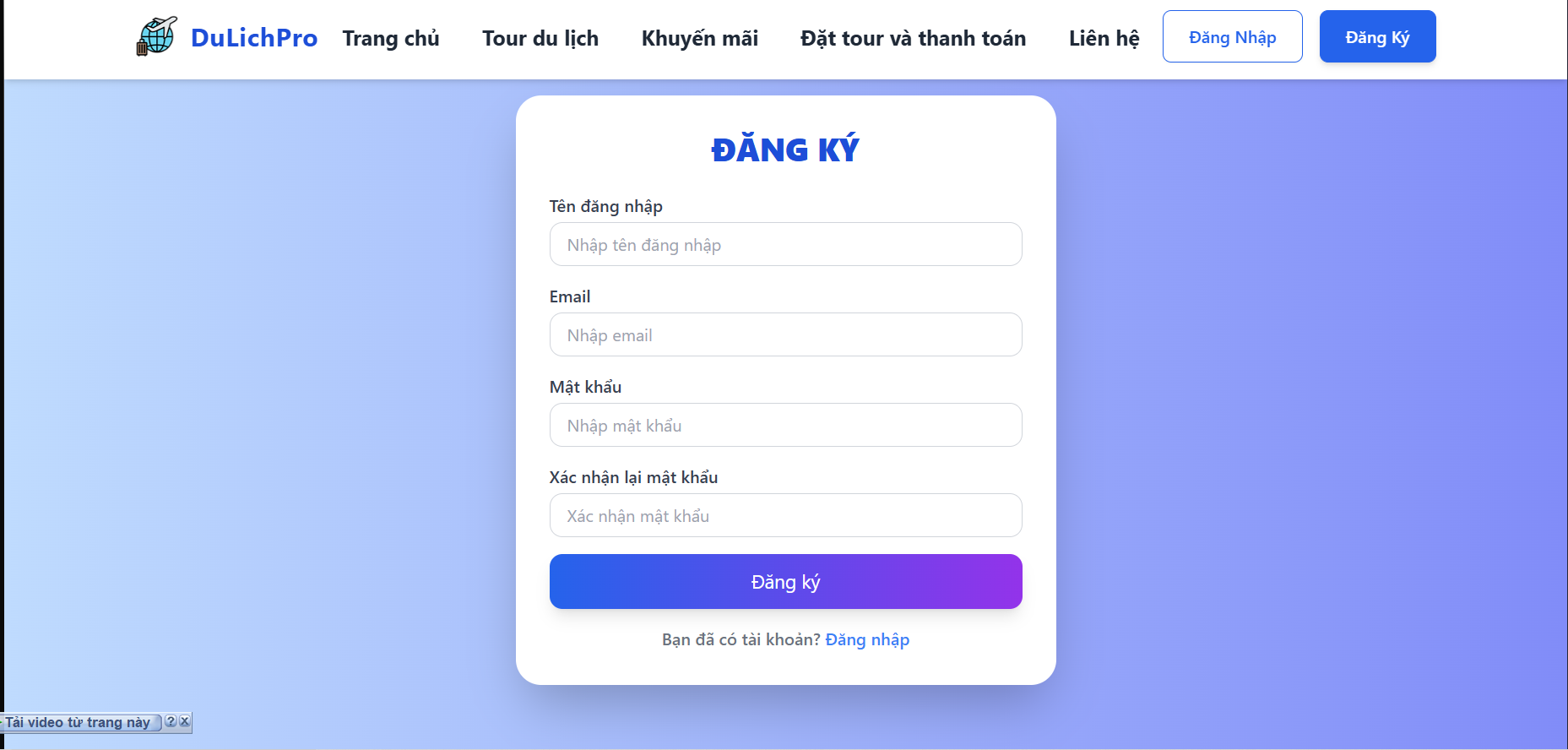
* POST api/Account/login: kiểm tra xem tài khoản và mật khẩu có đúng không, đồng thời kiểm tra role phải là customer hay không. Nếu đúng trả về JWT token và các thông tin tài khoản đó trừ mật khẩu.
* GET: api/Customer/email: lấy thông tin customer theo email và token.

**Luồng hoạt động:**

* Người dùng nhập tên đăng nhập và mật khẩu rồi bấm đăng nhập.
* Client gọi POST api/Account/login nếu kết quả trả về là đúng, lưu token và thông tin account lại. Nếu sai thì báo lỗi.
* Client gọi GET: api/Customer/email để lấy thông tin khách hàng, lưu thông tin khách hàng dựa trên kết quả trả về.
* Chuyển đến trang chủ.

### Đăng ký

**Giao diện:**



**Mô tả ngắn:** Dùng để đăng ký tài khoản cho khách hàng mới.

**Các api sử dụng:**

* POST api/Account/register: kiểm tra xem email và tên đăng nhập có trùng không , nếu không thì tạo một account mới kèm theo tạo 1 otp ngẫu nhiên cho chức năng xác nhận tài khoản.
* POST api/Customer/create: tạo một đối tượng khách hàng mới với các thông tin rỗng và email người dùng tạo tài khoản.

**Luồng hoạt động:**

* Người dùng nhập các trường trên và web client kiểm tra xem có hợp lệ không.
* Nếu hợp lệ hệ thống gọi POST api/Account/register để tạo tài khoản và server đồng thời gửi email xác nhận.
* Client dùng POST api/Customer/create để lấy tạo khách hàng mới.
* Chuyển đến trang đăng nhập tuy nhiên vẫn chưa thể đăng nhập mà cần thêm bước xác nhận.

### Xác nhận tài khoản

**Giao diện:**

Email được gửi về mẫu:

A screenshot of a computer

AI-generated content may be incorrect.

Giao diện khi xác nhận thành công

A screenshot of a computer

AI-generated content may be incorrect.

**Mô tả ngắn:** Xác nhận tài khoản qua email thì tài khoản mới có thể dùng và đăng nhập.

**Các api sử dụng:**

* POST api/Account/confirm: chuyển trạng thái isConfirm tài khoản trong data về đã xác nhận, người dùng có thể đăng nhập từ đó.

**Luồng hoạt động:**

* Người dùng nhận được email yêu cầu xác nhận kèm 1 đường link xác nhận
* Khi bấm vào liên kết sẽ đến trang confirm của web client, kèm theo query gồm email và otp. Đồng thời client cũng gọi api confirm.
* Server kiểm tra otp và email có hợp lệ không nếu có thì xác nhận tài khoản để người dùng đăng nhập.

### Quên mật khẩu

**Giao diện:**

Gửi email:

A screenshot of a computer

AI-generated content may be incorrect.

Email kèm otp

A screenshot of a web page

AI-generated content may be incorrect.

Giao diện sau khi gửi otp thành công

A screenshot of a computer

AI-generated content may be incorrect.

**Mô tả ngắn:** Lấy lại mật khẩu bằng email trong trường hợp quên.

**Các api sử dụng:**

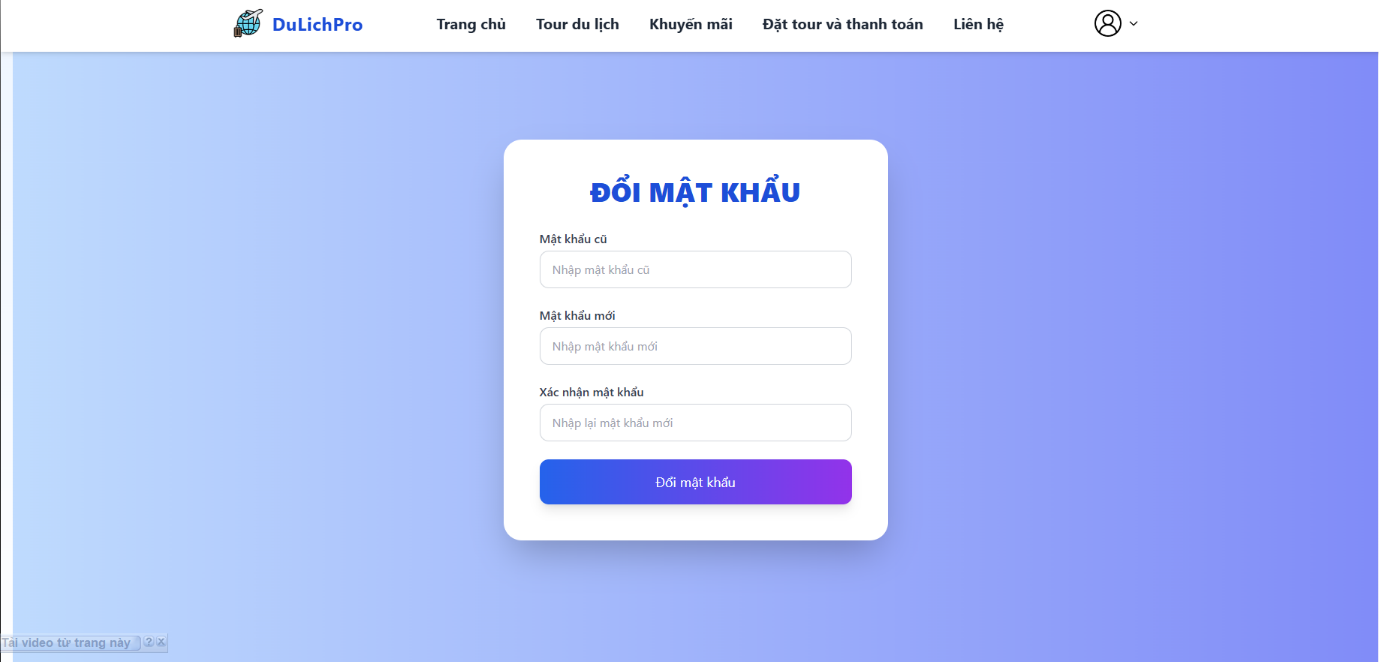
* POST api/Account/SendOTPResetPass: khởi tạo một otp ngẫu nhiên tương ứng với tài khoản của email người dùng nhập sau đó gửi về email cho người dùng otp đó.
* POST api/Account/ResetPassword: gửi otp, email và mật khẩu mới về server nếu hợp lệ thì cho phép đổi mật khẩu

**Luồng hoạt động:**

* Người dùng nhập email đã đăng ký tài khoản rồi bấm gửi mã xác thực. Client gọi api để gửi otp.
* Nếu tồn tại tài khoản, sau đó server sẽ gửi mã xác thực tới email đó.
* Trang resetPass hiện ra, người dùng nhập mã xác thực đã được gửi và mật khẩu mới, sau đó gọi api để kiểm tra các tính hợp lệ nếu mã xác thực hợp lệ thì đổi mật khẩu mới.

### Đổi mật khẩu

**Giao diện**



**Mô tả ngắn:** Đổi mật khẩu tài khoản của người dùng.

**Các api sử dụng:**

* GET api/Account/CheckToken: Kiểm tra token người dùng còn hạn không.
* PUT api/Account/update-password: Cập nhật mật khẩu mới cho tài khoản, có authourize với bearer token.

**Luồng hoạt động:**

* Người dùng chọn chức năng đổi mật khẩu. Client sẽ gọi api để kiểm tra token xem hết hạn chưa nếu hết hạn về lại trang đăng nhập
* Người dùng nhập mật khẩu cũ và mật khẩu mới rồi bấm đổi mật khẩu
* Gọi api api/Account/update-password để cập nhật mật khẩu với token và các thông tin trên. Server kiểm tra tính hợp lệ của thông tin rồi đổi mật khẩu cho người dùng.

### Cập nhật thông tin

**Giao diện**

A screenshot of a computer

AI-generated content may be incorrect.

**Mô tả ngắn:** Thay đổi thông tin khách hàng.

**Các api sử dụng:**

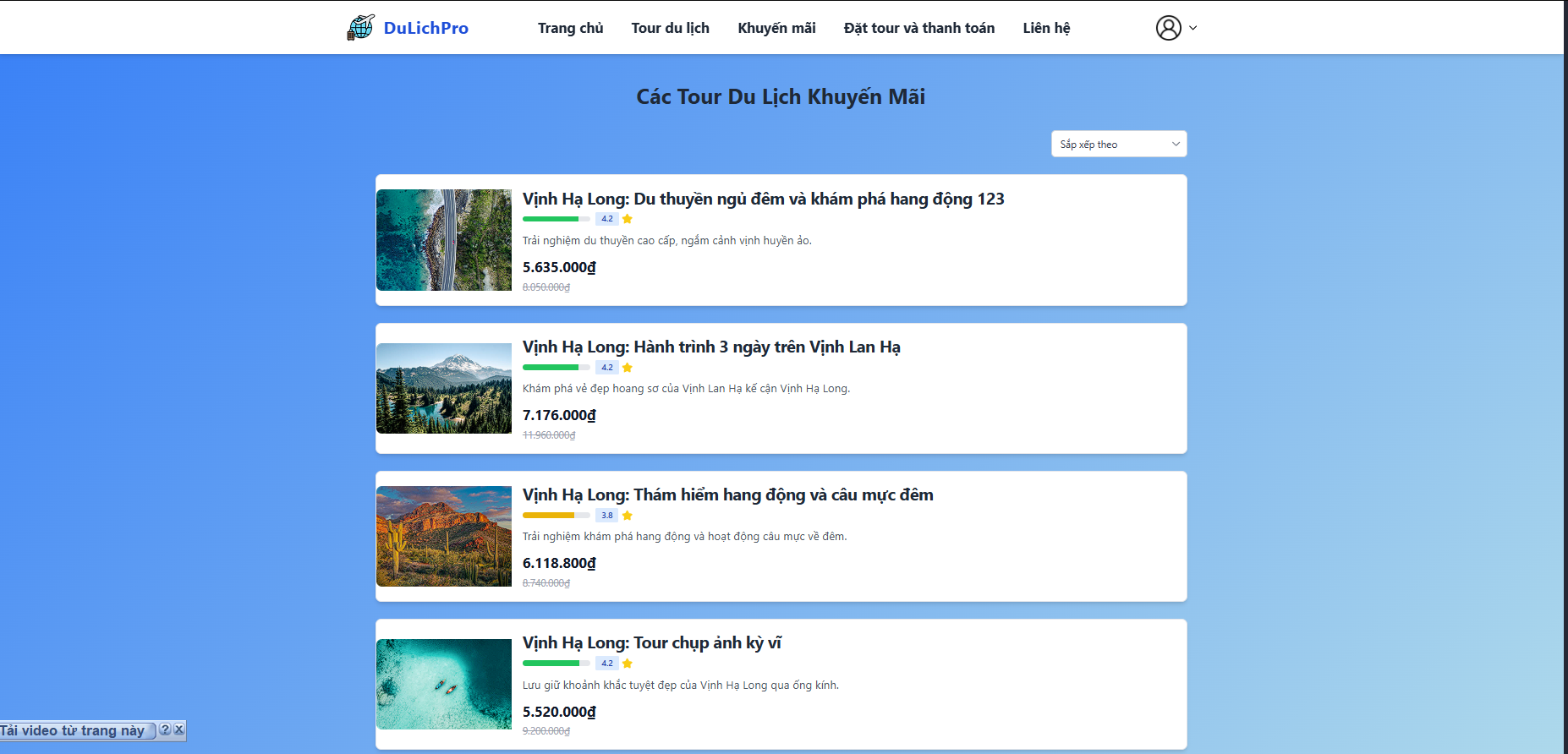
* GET api/Account/CheckToken: Kiểm tra token người dùng còn hạn không.
* PUT api/Customer/update: Cập nhật thông tin mới cho người dung, có authourize với bearer token.

**Luồng hoạt động:**

* Người dùng chọn chức năng xem thông tin khách hàng. Client gọi api kiểm tra token xem hết hạn chưa nếu hết hạn về lại trang đăng nhập
* Người dùng nhập có thể chỉnh sửa các trường mà client cho phép rồi bấm lưu thông tin.
* Gọi api api/Customer/update để cập nhật thông tin người dùng với token và các thông tin trên. Server kiểm tra tính hợp lệ của thông tin rồi cập nhật cho người dùng.

### Xem các tour khuyến mãi

**Giao diện**



**Mô tả ngắn**: Khi bấm vào hiện ra danh sách các tour du lịch khuyến mãi để người dùng chọn.

**Các api sử dụng:**

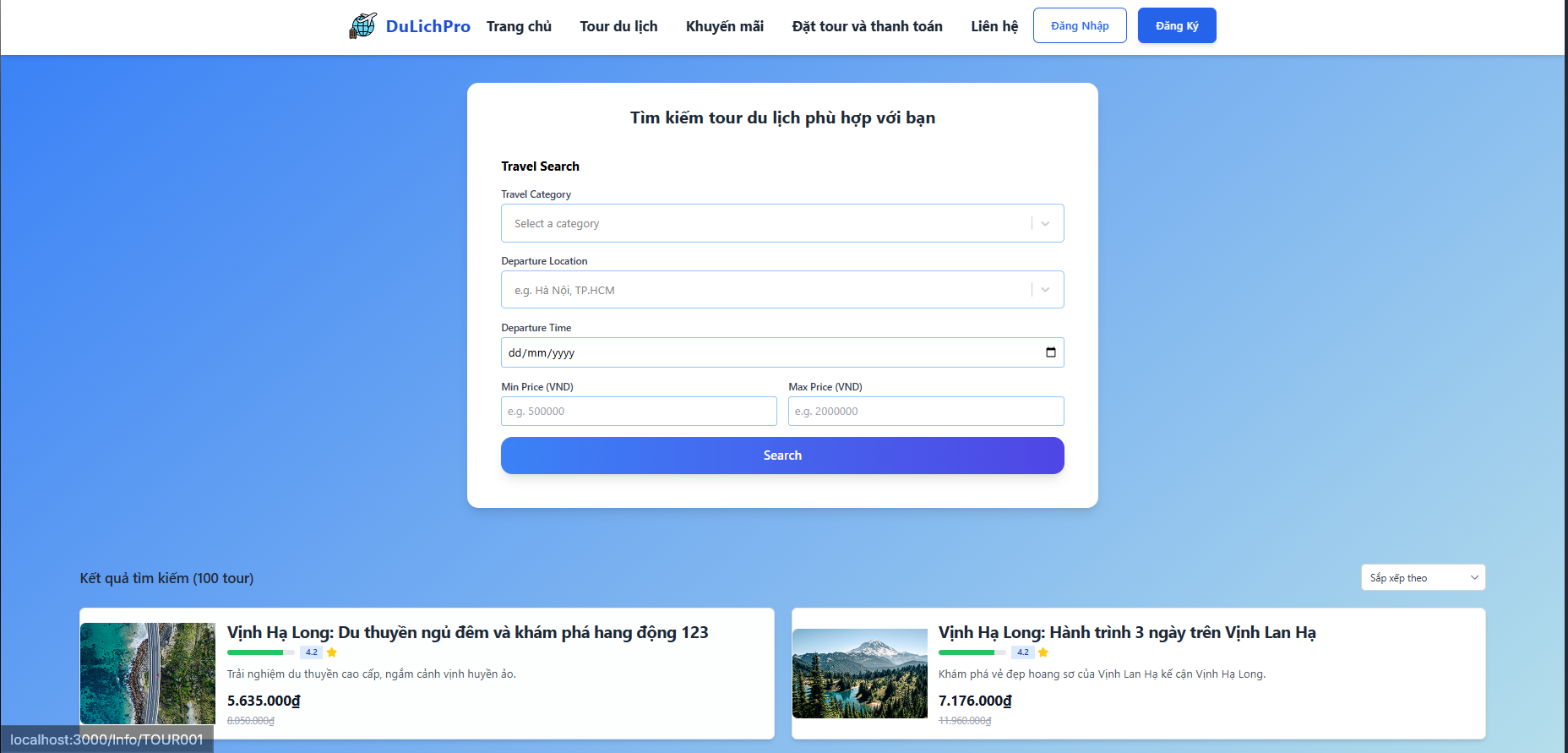
* GET /api/Tour/promotions: Lấy về các thông tin cần thiết của tour khuyến mãi gồm: tên tour, mô tả ngắn, số tiền ban đầu, số tiền đã giảm, đánh giá trung bình và url hình ảnh của tour trên cloud.

**Luồng hoạt động:**

* Người dùng chọn phần khuyến mãi trên navbar.
* Client sẽ gọi api này để lấy thông tin cần thiết.
* Client hiển thị kết quả ra màn hình, thực hiện phân trang và cũng có thể sort theo đánh giá và giá tour nếu người dùng cần.

### Tìm kiếm Tour

**Giao diện**



**Mô tả ngắn**: Giúp tìm kiếm tour phù hợp các tiêu chí.

**Các api sử dụng:**

GET api/Tour/search: Lấy về các thông tin cần thiết của tour gồm: tên tour, mô tả ngắn, số tiền ban đầu, số tiền đã giảm, đánh giá trung bình và url hình ảnh của tour trên cloud dựa trên các tiêu chí mà người dùng tìm kiếm.

GET api/Location/AllLocation: Lấy về tất cả địa điểm du lịch có trong hệ thống.

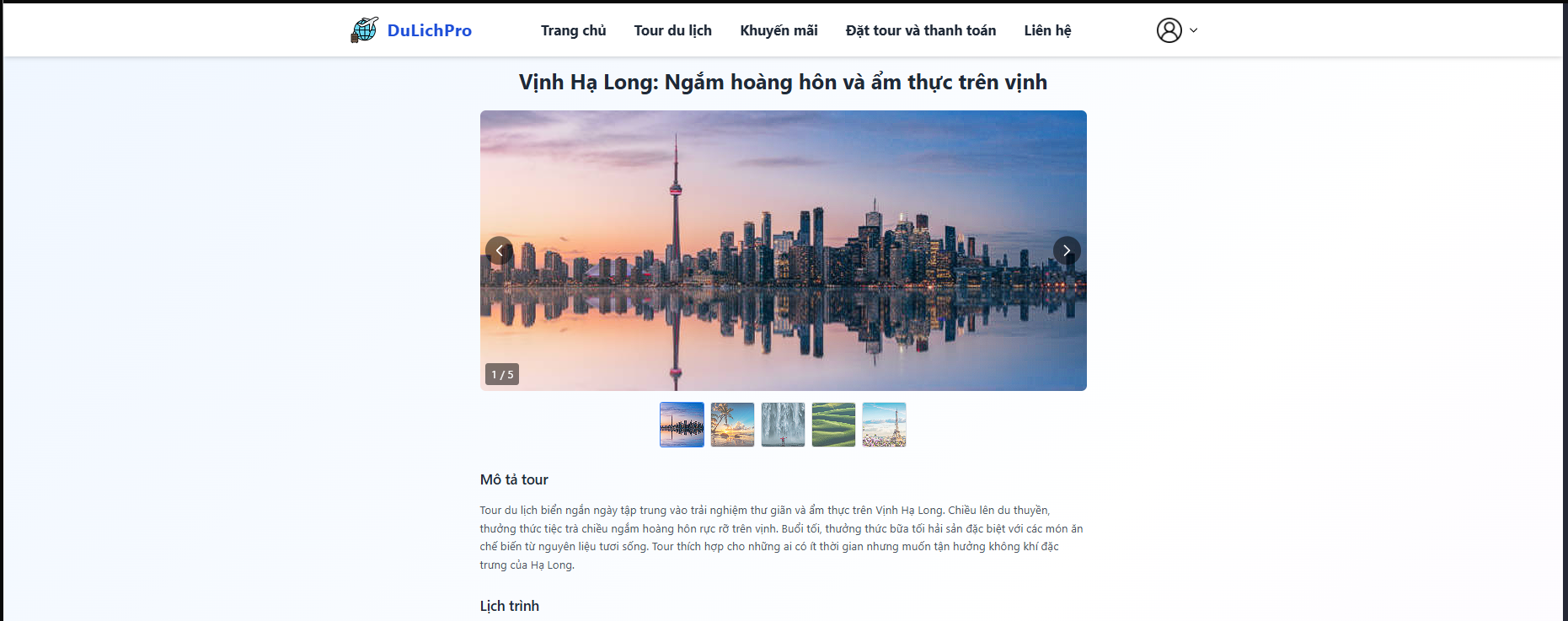
GET api/Category/all-categories: Lấy về tất cả thể loại du lịch có trong hệ thống.

**Luồng hoạt động:**

* Người dùng chọn phần tour du lịch trên navbar.
* Client sẽ gọi api này để tìm kiếm với các trường để trống khi mới vào trang và hiển thị vào result.
* Client gọi tiếp 2 api lấy địa điểm và thể loại du lịch rồi truyền data vào 2 trường tìm kiếm.
* Người dùng nhập các trường tìm kiếm sau đó ấn tìm kiếm.
* Clietn gọi api search để tìm kiếm rồi nhận về kết quả.
* Client hiển thị kết quả ra màn hình, thực hiện phân trang và cũng có thể sort theo đánh giá và giá tour nếu người dùng cần.

### Hiển thị thông tin tour

**Giao diện:**



A screenshot of a computer

AI-generated content may be incorrect.

**Mô tả ngắn:** Hiển thị thông tin của tour du lịch.

**Các api sử dụng:**

GET api/Tour/{id}: Trả về thông tin của Tour theo TourId.

GET api/Schedule/by-tour: Trả về tất cả lịch trình của 1 tour.

GET api/Image/get-tour: Trả về tất cả url hình ảnh liên quan đến tour

GET api/Review/get-tour: Trả về các review liên quan đến tour.

**Luồng hoạt động:**

* Người dùng chọn 1 tour bất kỳ trong khuyến mãi hay tìm kiếm.
* Gọi các api để lấy các thông tin liên quan trong tour đó.
* Client lấy thông tin đó truyền vào các component rồi hiển thị lên cho người dùng

**3.2.10 Đặt tour**

**Giao diện:**

A screenshot of a computer

AI-generated content may be incorrect.

**Mô tả ngắn:** Giúp người dùng đặt tour theo lịch trình đã chọn.

**Các api sử dụng:**

GET api/Schedule/booking-data: Giúp lấy các thông tin của lịch trình theo mã lịch trình đó.

POST api/Booking/create: Tạo mới 1 yêu cầu đặt tour.

**Luồng hoạt động:**

* Người dùng chọn 1 lịch trình còn chỗ bất kỳ trong tour.
* Client gọi api kiểm tra token của người dùng nếu không hợp lệ hoặc hết hạn thì qua trang đăng nhập.
* Client gọi api để lấy dữ liệu cho lịch trình đó.
* Người dùng có thể tăng giảm số lượng người lớn và trẻ em, hệ thống tự tính tiền cần trả rồi chọn đặt tour.
* Client gọi api đặt tour rồi server tạo 1 yêu cầu booking mới.
* Nếu đặt thành công hệ thống thông báo và chuyển về trang các tour đã đặt.

### Xem danh sách đặt tour và thanh toán:



A screenshot of a computer

AI-generated content may be incorrect.

**Mô tả ngắn:** Hiển thị các tour người dùng đã đặt và trạng thái của chúng.

**Các api sử dụng**

PUT api/Booking/cancel: Để yêu cầu hủy đặt tour nếu cần

GET api/Booking/customer: Lấy dữ liệu đặt tour của 1 customer.

**Luồng hoạt động:**

* Người dùng chọn đặt tour và thanh toán trên navbar, client kiểm tra token xem hợp lệ không, nếu không thì quay lại trang login.
* Client gọi api để lấy dữ liệu đặt tour của khách hàng và show ra màn hình.
* Các đặt tour có 4 trạng thái: hủy, chờ thanh toán, đã thanh toán, đã đánh giá.
* Trạng thái chờ thanh toán người dùng có thể bấm hủy. khi đó client sẽ gọi api để hủy tour cho người dùng.
* Trạng thái chờ thanh toán có thể bấm hiện mã, sẽ hiện mã qr thanh toán.
* Trạng thái đã thanh toán thì người dùng có thể đánh giá tour.

### Đánh giá tour

A screenshot of a computer

AI-generated content may be incorrect.

**Mô tả ngắn**: để lại đánh giá cho một tour du lịch

**Các api sử dụng**

POST /api/Review: dùng để upload 1 review của người dùng

**Luồng hoạt động:**

* Người dùng chọn vào đánh giá khi đó một cửa sổ để ghi review hiện lên.
* Người dùng nhập các trường và chọn gửi đánh giá.
* Client gọi api để gửi review đó lên server.

## 3.3 Các chức năng chính của admin

### 3.3.1 Đăng nhập

**Giao diện**

A login screen with blue and white text

AI-generated content may be incorrect.

**Mô tả ngắn**: Đăng nhập để vào trang admin

**Các api sử dụng**

POST api/Account/login-admin: Dùng để xem account của admin có tồn tại hay không.

**Luồng hoạt động:**

* Admin nhập tên đăng nhập và mật khẩu rồi bấm sign in.
* Client sẽ gọi api để kiểm tra tài khoản admin có hợp lệ không nếu có thì set lại trạng thái đăng nhập

### 3.3.2 Quản lý tour

**Giao diện:**

Danh sách tour

A screenshot of a computer

AI-generated content may be incorrect.

Thông tin tour:

A screenshot of a computer

AI-generated content may be incorrect.

Thêm lịch trình:

A screenshot of a computer

AI-generated content may be incorrect.

Thêm tour mới

A screenshot of a computer

AI-generated content may be incorrect.

**Mô tả ngắn**: admin có thể thêm tour, xem và chỉnh sửa thông tin tour, thêm lịch trình tour.

**Các api sử dụng:**

GET api/Tour/tourTableData: dùng để lấy data cho bảng tour.

GET api/Location/AllLocation: Lấy về tất cả địa điểm du lịch có trong hệ thống.

GET api/Category/all-categories: Lấy về tất cả thể loại du lịch có trong hệ thống.

GET api/Tour/${tourId}: Lấy data cho tour mà người dùng chọn xem thông tin

PUT api/Tour/${tourId}: Cập nhật data cho tour mà người dùng chọn xem thông tin

POST api/Tour/create: Giúp tạo tour mới.

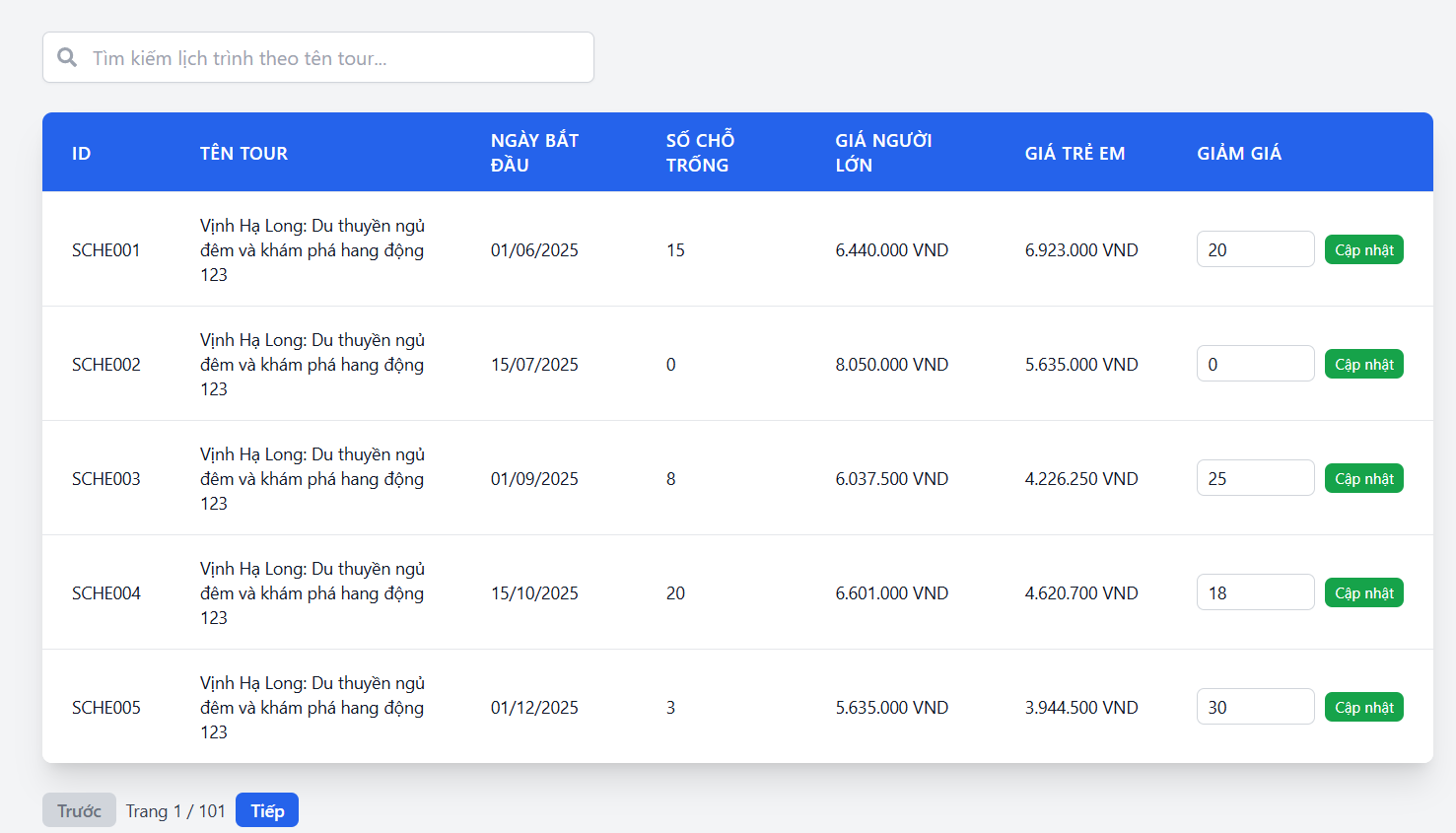
GET api/Schedule/create: Giúp tạo lịch trình mới cho tour chọn.

**Luồng hoạt động:**

* Admin chọn vào chức năng quản lý Tour.hệ thống sẽ kiểm tra đã đăng nhập chưa nếu chưa thì chuyển sang trang đăng nhập
* Client gọi api lấy tour data rồi hiển thị lên giao diện
* Admin có thể chọn xem chi tiết để xem thông tin của tour đó và có thể chỉnh sủa các trường cho phép bấm lưu thay đổi để lưu thay đổi. Sau đó client sẽ gọi api để cập nhật tour đó.
* Admin có thể thêm tour bằng cách chọn “Thêm tour mới” cần nhập đủ các trường yêu cầu sau đó bấm thêm tour. Client gọi api để thêm tour đó.
* Admin có thể thêm lịch trình bằng cách chọn thêm lịch trình.

### 3.3.3 Quản lý lịch trình

**Giao diện:**



**Mô tả ngắn:** Người dùng có thể cập nhật giá cho lịch trình.

**Các api sử dụng:**

GET api/Schedule/table-data: Lấy dữ liệu tất cả lịch trình để hiển thị cho table.

PUT api/Schedule/{id}/discount: điều chỉnh giá khuyến mãi cho schedule

**Luồng sử dụng:**

* Admin chọn vào chức năng quản lý lịch trình, hệ thống sẽ kiểm tra đã đăng nhập chưa nếu chưa thì chuyển sang trang đăng nhập
* Client gọi api lấy dữ liệu lịch trình rồi hiển thị dưới dạng table
* Khi admin cần cập nhật giá lịch trình có thể tự chỉnh trong ô giảm giá, rồi bấm cập nhật. Client sẽ gọi api để server cập nhật giá cho lịch trình đó.

### 3.3.4 Quản lý thanh toán

**Giao diện**

A screenshot of a computer

AI-generated content may be incorrect.

Thêm giao dịch:

A screenshot of a phone

AI-generated content may be incorrect.

**Mô tả ngắn**: giúp xem các thanh toán đã lưu, đồng thời thêm thanh toán mới nếu cần.

**Các api sử dung:**

GET api/Payment/all-payment: Lấy tất cả dữ liệu thanh toán để hiển thị lên giao diện.

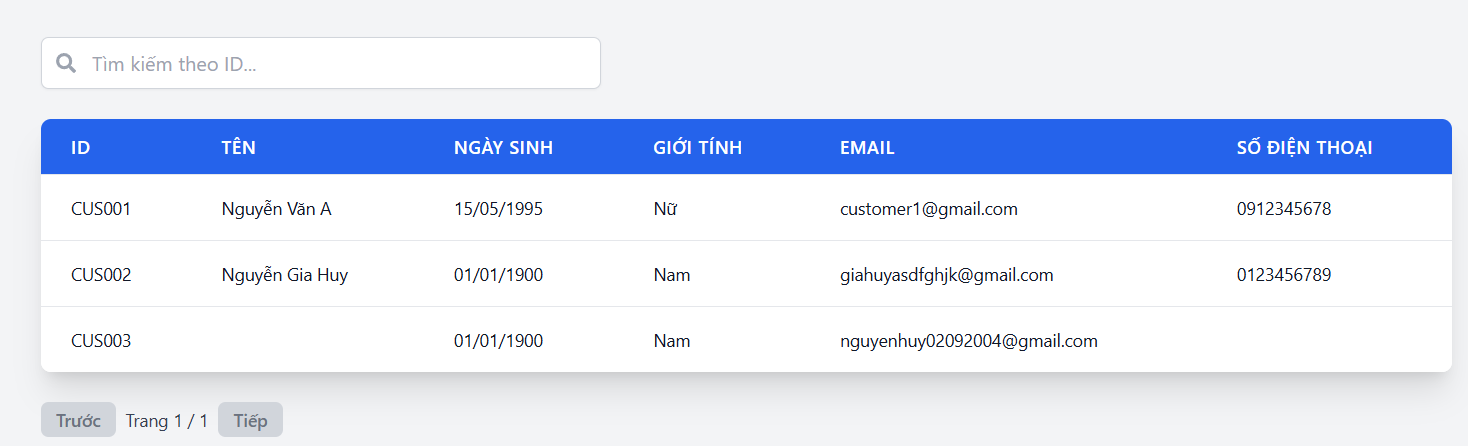
POST api/Payment/create: Để thêm giao dịch.

**Luồng hoạt động:**

* Admin chọn vào chức năng quản lý giao dịch, hệ thống sẽ kiểm tra đã đăng nhập chưa nếu chưa thì chuyển sang trang đăng nhập
* Client gọi api lấy dữ liệu giao dịch rồi hiển thị dưới dạng table
* Khi admin cần thêm giao dịch thì chọn thêm giao dịch, điền đủ thông tin vào forrm rồi chọn thêm thanh toán. Client sẽ gọi api để server thêm giao dịch đó.

### 3.3.5 Danh sách khách hàng

**Giao diện**



**Mô tả ngắn:** Hiển thị danh sách khách hàng

**Api sử dụng**

GET api/Customer/all-customers: Lấy dữ liệu của tất cả khách hàng.

**Luồng sử dụng:**

* Admin chọn vào chức năng danh sách khách hàng, hệ thống sẽ kiểm tra đã đăng nhập chưa nếu chưa thì chuyển sang trang đăng nhập
* Client gọi api lấy dữ liệu khách hàng rồi hiển thị dưới dạng table

### 3.3.6 Danh sách đặt chỗ

**Giao diện:**

A screenshot of a computer

AI-generated content may be incorrect.

**Mô tả ngắn:** Xem danh sách booking, có thể thay đổi trạng thái thanh toán.

**Các api sử dụng:**

GET api/Booking/all-bookings: Lấy tất cả dữ liệu booking.

PUT api/Booking/update-statusL Cập nhật trạng thái giao dịch của booking.

Luồng hoạt động:

* Admin chọn vào chức năng danh sách đặt chỗ, hệ thống sẽ kiểm tra đã đăng nhập chưa nếu chưa thì chuyển sang trang đăng nhập
* Client gọi api lấy dữ liệu đặt chỗ rồi hiển thị dưới dạng table
* Đối với các hàng dữ liệu mà trạng thái là thanh toán/chưa thanh toán admin có thể thay đổi linh hoạt. Sau khi thay đổi, client gọi api để cập nhật lại.

### 3.3.7 Quản lý hình ảnh Tour:

**Giao diện:**

A screenshot of a computer

AI-generated content may be incorrect.

Thêm xóa hình ảnh cho tour

A screenshot of a computer

AI-generated content may be incorrect.

**Mô tả ngắn:** Quản lý hình ảnh của các tour du lịch

**Các api sử dụng:**

GET api/Tour/tourTableData: dùng để lấy data tour cho bảng quản lý hình ảnh.

GET api/Image/get-image-data: lấy dữ liệu hình ảnh cho 1 tour cụ thể.

POST api/Image/create: upload hình ảnh cho 1 tour.

DELETE api/Image/${imageId}: xóa hình ảnh của theo id của hình ảnh.

Luồng hoạt động:

* Admin chọn vào chức năng quản lý hình ảnh tour, hệ thống sẽ kiểm tra đã đăng nhập chưa nếu chưa thì chuyển sang trang đăng nhập.
* Client gọi api lấy dữ liệu tour rồi hiển thị dưới dạng table
* Khi admin chọn xem hình ảnh trên một hàng bất kỳ, client gọi api lấy tất cả hình ảnh của 1 tour một form chứa các url hình ảnh của tour đó hiện lên.
* Người dùng có thể chọn Xóa để xóa hình ảnh khỏi tour đó, khi đó client sẽ gọi api server sẽ xóa thông tin hình ảnh trong cơ sở dữ liệu và hình ảnh trên cloud.
* Người dùng có thể upload thêm hình ảnh, khi đó client sẽ gọi api, server sẽ thêm hình ảnh trên cloud và lưu thông tin hình ảnh trong cơ sở dữ liệu.