**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**TÌM HIỂU THUẬT TOÁN XGBOOST ĐỂ DỰ ĐOÁN GIÁ PHÒNG CHO WEBSITE ĐẶT PHÒNG KHÁCH SẠN**

Sinh viên thực hiện: Hồ Thị Kiều Linh

MSSV: 6151071066

Lớp : Công nghệ thông tin K61

Khoá :K61

Giáo viên hướng dẫn: Ths. Trần Phong Nhã

Tp. Hồ Chí Minh, năm 2024

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**TÌM HIỂU THUẬT TOÁN XGBOOST ĐỂ DỰ ĐOÁN GIÁ PHÒNG CHO WEBSITE ĐẶT PHÒNG KHÁCH SẠN**

Sinh viên thực hiện: Hồ Thị Kiều Linh

MSSV: 6151071066

Lớp : Công nghệ thông tin K61

Khoá :K61

Giáo viên hướng dẫn: Ths. Trần Phong Nhã

Tp. Hồ Chí Minh, năm 2024

**LỜI CẢM ƠN**

Lời đầu tiên, em không biết nói gì hơn ngoài bày tỏ sự biết ơn sâu sắc đến các thầy cô. Trong suốt chặng đường học tập và làm đồ án tốt nghiệp em đã luôn nhận được sự hướng dẫn, giúp đỡ tận tình của thầy cô.

Đặc biệt, em xin bày tỏ sự kính trọng và lòng biết ơn sâu sắc nhất đến thầy giáo hướng dẫn thầy Trần Phong Nhã, thầy là người đã trực tiếp hướng dẫn, giúp đỡ cho em để em có thể hoàn thành đồ án này. Trong quá trình học tập và nghiên cứu, nếu em có những sai sót gì, kính mong thầy cô bỏ qua cho em!

Em xin kính chúc các thầy cô luôn luôn khỏe mạnh và ngày một thành công hơn trên con đường giảng dạy của mình.

Em xin trân trọng cảm ơn

***Tp. Hồ Chí Minh, ngày ….… tháng ….… năm ….…***

**Sinh viên thực hiện**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

|  |
| --- |
| ***Tp. Hồ Chí Minh, ngày ….… tháng ….… năm ….…***  **Giáo viên hướng dẫn**  ThS Trần Phong Nhã |

**MỤC LỤC**

[CHƯƠNG 1. TỔNG QUAN 6](#_Toc168495804)

[1.1 Khái quát về đề tài 6](#_Toc168495805)

[1.1.1 Bối cảnh đề tài: 6](#_Toc168495806)

[1.1.2 Cần thiết của đề tài: 6](#_Toc168495807)

[1.2 Mục tiêu nghiên cứu 7](#_Toc168495808)

[1.3 Phạm vi và đối tượng nghiên cứu 7](#_Toc168495809)

[1.3.1.Phạm vi nghiên cứu 8](#_Toc168495810)

[1.3.2.Đối tượng nghiên cứu 8](#_Toc168495811)

[1.4 Phương pháp nghiên cứu 8](#_Toc168495812)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 10](#_Toc168495813)

[2.1 Tổng quan về XGBoost 10](#_Toc168495814)

[2.1.1 Giới thiệu về thuật toán XGBoost và cách hoạt động 10](#_Toc168495815)

[2.1.3 Ứng dụng của XGBoost trong các lĩnh vực khác nhau 13](#_Toc168495816)

[2.2 Hệ thống đặt phòng trực tuyến trong ngành du lịch 14](#_Toc168495817)

[2.2.1 Phân loại và tính năng của hệ thống đặt phòng 14](#_Toc168495818)

[2.3 Laravel: Framework phát triển ứng dụng web 18](#_Toc168495819)

[2.3.1 Giới thiệu về Laravel 18](#_Toc168495820)

[2.3.2 Các tính năng và ưu điểm của Laravel 18](#_Toc168495821)

[CHƯƠNG 3. THIẾT KẾ VÀ PHÁT TRIỂN ĐỀ TÀI 22](#_Toc168495822)

[3.1 Thiết kế và mô hình hóa 22](#_Toc168495823)

[3.1.1 Phân tích yêu cầu của website và xây dựng use case 22](#_Toc168495824)

[3.1.2 Thiết kế cơ sở dữ liệu 48](#_Toc168495825)

[3.2 Phát triển website 52](#_Toc168495826)

[3.2.1 Thiết kế giao diện người dùng và các chức năng chính 52](#_Toc168495827)

[CHƯƠNG 4. NGHIÊN CỨU VÀ ỨNG DỤNG 60](#_Toc168495828)

[4.1. Áp dụng XGBoost vào dự đoán Booking Hotel 60](#_Toc168495829)

[4.1.1. Thu thập dữ liệu và tiền xử lý cho mô hình 60](#_Toc168495830)

[4.1.2. Xây dựng mô hình dự đoán sử dụng XGBoost 60](#_Toc168495831)

[4.2. Tích hợp mô hình vào website 60](#_Toc168495832)

[4.2.1. Triển khai mô hình dự đoán trên website Booking Hotel 60](#_Toc168495833)

[CHƯƠNG 5. KẾT QUẢ VÀ KIẾN NGHỊ 63](#_Toc168495834)

[5.1 Kết quả đạt được 63](#_Toc168495835)

[5.1.1. Tóm tắt kết quả đạt được 63](#_Toc168495836)

[5.1.2. Đánh giá ưu điểm và hạn chế 63](#_Toc168495837)

[5.2 Kiến nghị 63](#_Toc168495838)

[5.2.1. Mở rộng tính năng và khả năng của hệ thống 63](#_Toc168495839)

[5.2.2. Nghiên cứu và tích hợp các công nghệ mới 64](#_Toc168495840)

[TÀI LIỆU THAM KHẢO 65](#_Toc168495841)

# CHƯƠNG 1. TỔNG QUAN

## Khái quát về đề tài

1.1.1 Bối cảnh đề tài:

Sự phát triển của công nghệ và internet trong ngành du lịch:

Trong những năm gần đây, sự phát triển của công nghệ thông tin và internet đã thay đổi cách mà người tiêu dùng tìm kiếm và đặt phòng khách sạn. Việc truy cập internet thông qua các thiết bị di động đã trở thành một phần không thể thiếu trong việc lên kế hoạch và thực hiện các chuyến du lịch. Người dùng mong đợi có thể tìm kiếm, so sánh giá cả và đặt phòng khách sạn một cách thuận tiện từ bất kỳ đâu và bất kỳ khi nào.

Tăng cường sự cạnh tranh:

Thị trường đặt phòng khách sạn trực tuyến ngày nay đang chứng kiến sự cạnh tranh khốc liệt giữa các công ty lớn và nhỏ. Sự cạnh tranh này thúc đẩy các doanh nghiệp phải không ngừng cải thiện dịch vụ của mình, từ quy trình đặt phòng cho đến trải nghiệm của khách hàng.

Tính quan trọng của dự đoán và tối ưu hóa:

Trong ngành du lịch, việc dự đoán và đáp ứng nhu cầu của khách hàng là rất quan trọng. Các trang web đặt phòng cần phải có khả năng dự đoán các yêu cầu và sở thích của khách hàng, từ đó tối ưu hóa quy trình đặt phòng và cung cấp những gợi ý phù hợp nhất.

Sức mạnh của Machine Learning và XGBoost:

Machine Learning, đặc biệt là các thuật toán như XGBoost, đã chứng minh sức mạnh của mình trong việc dự đoán và phân loại trong nhiều lĩnh vực. Áp dụng Machine Learning vào ngành du lịch có thể cung cấp những dự đoán chính xác hơn về hành vi của khách hàng, giúp tối ưu hóa các quy trình kinh doanh và cải thiện trải nghiệm của người dùng.

Xu hướng tăng trưởng của ngành du lịch:

Ngành du lịch vẫn đang tiếp tục phát triển mạnh mẽ, với số lượng du khách quốc tế và nội địa tăng lên mỗi năm. Điều này tạo ra một cơ hội lớn cho các doanh nghiệp trong ngành du lịch mở rộng hoạt động của họ và thu hút một lượng lớn khách hàng mới.

* + 1. Cần thiết của đề tài:
* Hiệu quả và tính cạnh tranh: Việc nghiên cứu thuật toán XGBoost và áp dụng nó vào việc dự đoán nhu cầu và xu hướng của khách hàng có thể cung cấp cho các doanh nghiệp trong ngành du lịch một lợi thế cạnh tranh lớn.
* Tối ưu hóa trải nghiệm của người dùng: Việc phát triển website đặt phòng khách sạn với sự tích hợp của XGBoost có thể cung cấp cho người dùng những trải nghiệm tìm kiếm và đặt phòng tốt nhất, từ đó tăng sự hài lòng và trung thành của họ.
* Dự đoán giá phòng: Sử dụng XGBoost để dự đoán nhu cầu giá phòng có thể giúp các doanh nghiệp du lịch tối ưu hóa quản lý nguồn cung, từ việc đặt giá cho các phòng trống đến kế hoạch tiếp thị và quảng cáo.
* Tích hợp công nghệ và ngành du lịch: Việc kết hợp giữa công nghệ thông tin và ngành du lịch là không thể tránh khỏi trong thời đại số hóa ngày nay, và đề tài này cung cấp một cơ hội tuyệt vời để làm điều đó.

## Mục tiêu nghiên cứu

1. Tìm hiểu và nắm vững kiến thức về thuật toán XGBoost: Nghiên cứu chi tiết về nguyên lý hoạt động, cấu trúc, và các tham số quan trọng của thuật toán XGBoost.

2. Phân tích dữ liệu giá phòng: Thu thập và xử lý dữ liệu liên quan đến giá phòng từ các website đặt phòng khách sạn. Xác định các yếu tố ảnh hưởng đến giá phòng như: loại phòng, thời gian đặt phòng, số lượng người đặt phòng.

3. Ứng dụng thuật toán XGBoost để dự đoán giá phòng: Xây dựng mô hình dự đoán giá phòng sử dụng thuật toán XGBoost. Đánh giá hiệu quả của mô hình qua các chỉ số đo lường như MAE (Mean Absolute Error), RMSE (Root Mean Square Error), và R² (Coefficient of Determination)

4. So sánh với các thuật toán khác: So sánh hiệu quả của mô hình XGBoost với các thuật toán dự đoán khác như Linear Regression, Decision Tree, và Random Forest để xác định ưu và nhược điểm của từng phương pháp.

5. Ứng dụng thực tế: Đưa ra các ứng dụng thực tế của mô hình dự đoán giá phòng vào việc tối ưu hóa chiến lược giá cho các website đặt phòng khách sạn, giúp cải thiện trải nghiệm người dùng và tăng doanh thu.

1.3 Phạm vi và đối tượng nghiên cứu

### 1.3.1.Phạm vi nghiên cứu

1. Thuật toán XGBoost: Nghiên cứu sâu về cách XGBoost hoạt động, các thành phần của thuật toán và cách áp dụng nó vào bài toán dự đoán nhu cầu đặt phòng khách sạn.

2. Phát triển website Booking Hotel: Phạm vi này bao gồm quá trình phát triển và thiết kế website đặt phòng khách sạn, từ giao diện người dùng đến cơ sở dữ liệu và tích hợp các tính năng của thuật toán XGBoost.

3. Tối ưu hóa trải nghiệm người dùng: Tìm hiểu về cách cải thiện trải nghiệm người dùng thông qua việc sử dụng thuật toán XGBoost trong việc tìm kiếm và đề xuất các phòng khách sạn phù hợp.

4. Đánh giá và thử nghiệm: Đánh giá hiệu suất của website booking được phát triển, đặc biệt là trong việc dự đoán và đề xuất khách sạn, thông qua các thử nghiệm và phân tích kết quả.

1.3.2.Đối tượng nghiên cứu

1. Thuật toán và Machine Learning Researchers: Những người quan tâm đến việc nghiên cứu và phát triển các thuật toán học máy, đặc biệt là XGBoost, và ứng dụng chúng vào các bài toán thực tế.

2. Web Developers: Những nhà phát triển web chuyên về việc xây dựng các trang web đặt phòng khách sạn và quan tâm đến cách tích hợp các thuật toán học máy vào các ứng dụng của họ.

3. Người làm trong ngành du lịch: Những người quản lý hoặc làm việc trong ngành du lịch, đặc biệt là những người quan tâm đến cách tối ưu hóa quy trình đặt phòng và cải thiện trải nghiệm của khách hàng.

4. Người tiêu dùng: Những người dùng cuối cùng của website đặt phòng khách sạn, quan tâm đến việc có được trải nghiệm tìm kiếm và đặt phòng thuận tiện và hiệu quả.

## 1.4 Phương pháp nghiên cứu

1. Thu thập dữ liệu:

- Crawler và Scraping: Sử dụng các công cụ tự động (crawler) để thu thập dữ liệu từ các trang web khác nhau, hoặc sử dụng kỹ thuật scraping để trích xuất thông tin từ các trang web có cấu trúc.

- Khảo sát trực tuyến (Online Surveys): Tạo các khảo sát trực tuyến để thu thập dữ liệu từ một mẫu ngẫu nhiên hoặc một nhóm cụ thể của người dùng trên internet.

- Dữ liệu công cộng (Public Datasets): Sử dụng các bộ dữ liệu công cộng có sẵn trên internet hoặc từ các nguồn như OpenStreetMap, Kaggle, hoặc Google Dataset Search.

2. Xử lý dữ liệu:

- Tiền xử lý (Preprocessing): Bao gồm các bước như loại bỏ dữ liệu trùng lặp, điền các giá trị thiếu, chuẩn hóa dữ liệu, và mã hóa biến phân loại thành dạng số.

- Phân tích dữ liệu (Data Analysis): Sử dụng các công cụ và kỹ thuật phân tích dữ liệu như thống kê mô tả, phân tích hồi quy, và phân tích phân loại để hiểu rõ hơn về dữ liệu thu thập.

- Trích xuất đặc trưng (Feature Extraction): Tạo ra các đặc trưng mới từ dữ liệu gốc để cải thiện hiệu suất của mô hình machine learning như XGBoost.

3. Phương pháp thống kê và machine learning:

- Mô hình hóa dữ liệu (Modeling): Sử dụng các phương pháp machine learning như XGBoost để xây dựng các mô hình dự đoán dựa trên dữ liệu đã thu thập và được xử lý.

1.5 Cấu trúc báo cáo thực tập chuyên môn

Chương 1 Tổng quan

Chương 2: Cơ sở lý thuyết

Chương 3: Thiết kế và phát triển

Chương 4: Nghiên cứu và ứng dụng

Chương 5: Kết luận và kiến nghị

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## 2.1 Tổng quan về XGBoost

### 2.1.1 Giới thiệu về thuật toán XGBoost và cách hoạt động

* 1. Thuật toán Xgboost

XGBoost (eXtreme Gradient Boosting) là một thư viện mã nguồn mở được sử dụng rộng rãi trong các cuộc thi học máy và thực tế để giải quyết các bài toán về phân loại và hồi quy. Nó được xây dựng dựa trên thuật toán Gradient Boosting nhưng cải tiến hơn để tăng hiệu suất và độ chính xác.

XGBoost được phát triển bởi Tianqi Chen và là một phần của dự án Distributed (Deep) Machine Learning Community (DMLC). Nó được viết bằng ngôn ngữ C++, do đó có thể đạt được tốc độ tính toán rất nhanh.

* 1. Cách hoạt động của Xgboost

XGBoost thực chất là một kỹ thuật ensemble học máy, cụ thể là kỹ thuật boosting, nơi mà các mô hình yếu (weak learners, thường là cây quyết định) được xây dựng tuần tự, mỗi mô hình cố gắng sửa các lỗi của mô hình trước đó.

Các Bước Chính của XGBoost:

Khởi Tạo Mô Hình Đầu Tiên:

Bắt đầu với một mô hình cơ sở, thường là một cây quyết định đơn giản.

Tính Toán Residuals (phần dư):

Sau khi mô hình đầu tiên dự đoán, phần dư (residuals) được tính toán như là sự khác biệt giữa giá trị thực tế và giá trị dự đoán.

Xây Dựng Mô Hình Mới để Dự Đoán Residuals:

Một mô hình cây quyết định mới được huấn luyện để dự đoán các residuals từ mô hình trước.

Cập Nhật Dự Đoán:

Dự đoán cuối cùng là tổng của dự đoán từ tất cả các mô hình trước đó và dự đoán từ mô hình mới nhất (với một hệ số học nhỏ - learning rate).

Lặp Lại Quá Trình:

Tiếp tục lặp lại các bước 2-4 cho đến khi số lượng mô hình đạt tới ngưỡng đã định hoặc cho đến khi không có sự cải thiện nào đáng kể về độ chính xác.

Các Thành Phần Chính của XGBoost:

Loss Function (Hàm Mất Mát):

Để tối ưu hóa mô hình, XGBoost sử dụng một hàm mất mát để đo lường sự khác biệt giữa giá trị dự đoán và giá trị thực tế. Hàm mất mát phổ biến bao gồm hàm mất mát trung bình bình phương (Mean Squared Error) cho các bài toán hồi quy và hàm logloss cho các bài toán phân loại.

Regularization (Chuẩn Hóa):

XGBoost thêm vào các điều khoản chuẩn hóa trong hàm mất mát để giảm thiểu tình trạng overfitting. Điều này giúp mô hình tổng quát hóa tốt hơn trên dữ liệu chưa từng thấy.

Tree Pruning (Cắt Tỉa Cây):

XGBoost sử dụng một kỹ thuật gọi là "pruning" để loại bỏ các nhánh không đóng góp nhiều vào việc cải thiện độ chính xác của mô hình.

Learning Rate (Hệ Số Học):

Để giảm thiểu ảnh hưởng của mỗi cây đối với dự đoán cuối cùng, XGBoost sử dụng một hệ số học nhỏ, điều này giúp mô hình học dần dần và tránh việc fitting quá mức.

Parallel Processing (Xử Lý Song Song):

XGBoost có khả năng thực hiện tính toán song song, giúp tăng tốc độ huấn luyện và dự đoán.

2.1.2 Các tính năng và ưu điểm của XGBoost

1. Các tính năng của XGBoost

Hiệu Suất Cao:

Tốc Độ: XGBoost được viết bằng C++ và có thể xử lý dữ liệu rất nhanh.

Xử Lý Song Song: Hỗ trợ tính toán song song, giúp tăng tốc độ huấn luyện và dự đoán.

Khả Năng Tổng Quát Hóa Tốt:

Regularization: Bao gồm các kỹ thuật chuẩn hóa L1 (Lasso) và L2 (Ridge) để tránh overfitting.

Tree Pruning: Kỹ thuật cắt tỉa cây để loại bỏ các nhánh không cần thiết.

Tối Ưu Hóa Bộ Nhớ:

Block Structure: Sử dụng cấu trúc khối để tối ưu hóa bộ nhớ và cải thiện hiệu suất tính toán.

Sparsity Awareness: Hỗ trợ dữ liệu thưa, tự động xử lý các giá trị thiếu (missing values).

Tính Linh Hoạt:

Đa Dạng Các Loại Mất Mát (Loss Functions): Hỗ trợ nhiều hàm mất mát như logistic, linear, và hàm đặc biệt cho ranking.

Custom Objective and Evaluation Functions: Người dùng có thể tùy chỉnh hàm mục tiêu và hàm đánh giá.

Công Cụ Chẩn Đoán và Kiểm Tra:

Early Stopping: Ngừng huấn luyện sớm nếu không có cải thiện về hiệu suất trên tập kiểm tra.

Cross Validation: Tích hợp sẵn công cụ cross-validation để đánh giá mô hình.

Khả Năng Xử Lý Dữ Liệu Lớn:

Out-of-Core Computing: Hỗ trợ xử lý dữ liệu lớn không vừa bộ nhớ bằng cách xử lý từng khối dữ liệu nhỏ.

1. Ưu điểm của XGBoost

Hiệu Suất Cao và Tốc Độ Nhanh:

XGBoost được tối ưu hóa cho cả tốc độ và hiệu suất, giúp xử lý nhanh các tập dữ liệu lớn và phức tạp.

Độ Chính Xác Cao:

Nhờ các kỹ thuật như gradient boosting và regularization, XGBoost thường đạt được độ chính xác rất cao trên nhiều loại bài toán.

Giảm Thiểu Overfitting:

Các kỹ thuật regularization và pruning giúp giảm thiểu nguy cơ overfitting, làm cho mô hình tổng quát hóa tốt hơn trên dữ liệu mới.

Khả Năng Giải Thích:

XGBoost cung cấp công cụ để đánh giá và giải thích tầm quan trọng của các tính năng, giúp người dùng hiểu rõ mô hình hơn.

Linh Hoạt và Dễ Sử Dụng:

Hỗ trợ nhiều loại hàm mất mát và có khả năng tùy chỉnh hàm mục tiêu, làm cho XGBoost phù hợp với nhiều loại bài toán khác nhau.

Hỗ Trợ Mạnh Mẽ Cộng Đồng và Tài Liệu Phong Phú:

Là một trong những thuật toán phổ biến nhất trong cộng đồng học máy, XGBoost có tài liệu phong phú và cộng đồng hỗ trợ mạnh mẽ.

### 2.1.3 Ứng dụng của XGBoost trong các lĩnh vực khác nhau

1. Tài Chính

Phát Hiện Gian Lận: XGBoost được sử dụng để phát hiện các giao dịch gian lận trong các hệ thống tài chính. Với khả năng xử lý nhanh chóng và độ chính xác cao, XGBoost có thể phân tích các mẫu giao dịch và phát hiện ra các hoạt động bất thường.

Dự Báo Giá Cổ Phiếu: XGBoost được áp dụng để phân tích các yếu tố thị trường và dự báo giá cổ phiếu. Các mô hình này thường sử dụng dữ liệu lịch sử về giá cổ phiếu, khối lượng giao dịch, và các chỉ số tài chính khác để đưa ra dự đoán chính xác.

2. Y Tế

Chẩn Đoán Bệnh: XGBoost được sử dụng để xây dựng các mô hình chẩn đoán bệnh dựa trên dữ liệu y tế của bệnh nhân. Ví dụ, nó có thể giúp phát hiện sớm các bệnh như ung thư, tiểu đường, và các bệnh tim mạch.

Phân Loại Hình Ảnh Y Tế: Trong lĩnh vực xử lý hình ảnh y tế, XGBoost được sử dụng để phân loại các hình ảnh như X-quang, MRI, và CT scan nhằm phát hiện các dấu hiệu bệnh lý.

3. Tiếp Thị và Bán Hàng

Phân Tích Hành Vi Khách Hàng: XGBoost có thể phân tích hành vi mua sắm của khách hàng và dự đoán các xu hướng mua sắm. Điều này giúp các công ty tiếp thị sản phẩm một cách hiệu quả hơn và tối ưu hóa chiến dịch quảng cáo.

Dự Báo Doanh Thu: XGBoost được sử dụng để dự báo doanh thu dựa trên các yếu tố như mùa vụ, xu hướng thị trường, và các chiến dịch quảng cáo.

4. Công Nghệ Thông Tin

Phát Hiện Anomalies trong Hệ Thống: XGBoost được sử dụng để phát hiện các hành vi bất thường trong các hệ thống CNTT, chẳng hạn như phát hiện sự xâm nhập hoặc lỗi hệ thống.

Xử Lý Ngôn Ngữ Tự Nhiên (NLP): XGBoost có thể được sử dụng trong các ứng dụng NLP để phân loại văn bản, phân tích cảm xúc, và gán nhãn văn bản.

5. Nông Nghiệp

Dự Báo Năng Suất Cây Trồng: XGBoost được sử dụng để dự báo năng suất cây trồng dựa trên các yếu tố như điều kiện thời tiết, loại đất, và các biện pháp canh tác.

Phát Hiện Bệnh Hại Cây Trồng: XGBoost có thể giúp phát hiện sớm các bệnh hại cây trồng thông qua phân tích dữ liệu từ các cảm biến và hình ảnh.

6. Giao Thông và Vận Tải

Dự Báo Lưu Lượng Giao Thông: XGBoost được sử dụng để dự báo lưu lượng giao thông và tình trạng kẹt xe dựa trên dữ liệu lịch sử và các yếu tố như thời tiết và sự kiện địa phương.

Quản Lý Đội Xe: XGBoost có thể giúp tối ưu hóa quản lý đội xe bằng cách dự báo nhu cầu vận tải và lên kế hoạch bảo trì xe.

7. Giáo Dục

Phân Tích Hiệu Suất Học Tập: XGBoost có thể phân tích dữ liệu học tập của sinh viên để dự đoán hiệu suất học tập và xác định những sinh viên cần hỗ trợ thêm.

Cá Nhân Hóa Học Tập: XGBoost được sử dụng để cá nhân hóa lộ trình học tập của từng sinh viên dựa trên khả năng và sở thích học tập của họ.

## 2.2 Hệ thống đặt phòng trực tuyến trong ngành du lịch

### 2.2.1 Phân loại và tính năng của hệ thống đặt phòng

1. Phân loại

Đặt Phòng Khách Sạn: Các hệ thống như Booking.com, Hotels.com, và Agoda.

Đặt Vé Máy Bay: Các hệ thống như Expedia, Skyscanner, và Kayak.

Thuê Xe: Các hệ thống như Rentalcars.com, Hertz, và Avis.

Đặt Tour Du Lịch: Các hệ thống như Viator, GetYourGuide, và Klook.

Cho Thuê Nhà Riêng: Các hệ thống như Airbnb và Vrbo.

Theo Đối Tượng Khách Hàng:

Khách Hàng Cá Nhân: Hệ thống dành cho người dùng cá nhân, tập trung vào sự tiện lợi và đa dạng dịch vụ.

Khách Hàng Doanh Nghiệp: Hệ thống dành cho các doanh nghiệp, tập trung vào quản lý và báo cáo chi phí, ưu đãi nhóm, và hỗ trợ doanh nghiệp.

Theo Công Nghệ Sử Dụng:

Web-Based Systems: Các hệ thống hoạt động trên trình duyệt web.

Mobile Apps: Các ứng dụng đặt phòng trên điện thoại di động.

Hybrid Systems: Kết hợp cả ứng dụng di động và trang web.

1. Tính năng của hệ thống

1.Tìm Kiếm và Lựa Chọn

Tìm Kiếm Nâng Cao: Cho phép khách hàng tìm kiếm theo nhiều tiêu chí như giá, vị trí, tiện nghi, và đánh giá.

Bản Đồ Tương Tác: Hiển thị vị trí của các khách sạn hoặc dịch vụ trên bản đồ, giúp khách hàng dễ dàng chọn lựa dựa trên địa điểm.

2. Đặt Chỗ và Thanh Toán

Quy Trình Đặt Chỗ Đơn Giản: Giao diện trực quan và quy trình đặt chỗ dễ hiểu, giảm thiểu các bước không cần thiết.

Nhiều Phương Thức Thanh Toán: Hỗ trợ thanh toán qua thẻ tín dụng, PayPal, chuyển khoản ngân hàng, và các phương thức khác.

3. Quản Lý Đặt Chỗ

Xác Nhận và Quản Lý Đặt Chỗ: Gửi email hoặc SMS xác nhận đặt chỗ, cho phép khách hàng xem và quản lý các đặt chỗ của mình.

Hủy và Thay Đổi Đặt Chỗ: Cho phép khách hàng dễ dàng hủy hoặc thay đổi đặt chỗ nếu cần.

4. Gợi Ý Cá Nhân Hóa

Gợi Ý Dịch Vụ: Dựa trên lịch sử tìm kiếm và đặt chỗ, hệ thống có thể gợi ý các dịch vụ phù hợp với sở thích của khách hàng.

Thông Báo Khuyến Mãi: Gửi thông báo về các chương trình khuyến mãi và ưu đãi đặc biệt dựa trên sở thích của người dùng.

5. Hỗ Trợ Khách Hàng

Hỗ Trợ Trực Tuyến: Cung cấp hỗ trợ khách hàng qua chat trực tuyến, email, và điện thoại.

FAQ và Hướng Dẫn: Cung cấp tài liệu hướng dẫn và câu hỏi thường gặp để hỗ trợ người dùng.

6. Dữ Liệu và Phân Tích

Báo Cáo và Phân Tích: Cung cấp các báo cáo chi tiết về hành vi người dùng, doanh thu, và các chỉ số khác.

Theo Dõi Hiệu Suất: Theo dõi hiệu suất của các dịch vụ và chiến dịch marketing.

Lợi Ích Của Các Tính Năng

Tối Ưu Hóa Trải Nghiệm Người Dùng: Các tính năng tìm kiếm nâng cao, bản đồ tương tác, và gợi ý cá nhân hóa giúp nâng cao trải nghiệm người dùng.

Tiết Kiệm Thời Gian và Công Sức: Quy trình đặt chỗ đơn giản và khả năng quản lý đặt chỗ dễ dàng giúp tiết kiệm thời gian và công sức cho khách hàng.

Tăng Độ Tin Cậy: Đánh giá và bình luận từ người dùng khác giúp tăng độ tin cậy và quyết định mua sắm.

Hỗ Trợ Khách Hàng Tốt Hơn: Hỗ trợ trực tuyến và các tài liệu hướng dẫn giúp giải quyết các vấn đề của khách hàng nhanh chóng và hiệu quả.

Phân Tích và Cải Tiến: Dữ liệu và phân tích giúp các doanh nghiệp hiểu rõ hơn về hành vi khách hàng và tối ưu hóa dịch vụ.

2.2.2 Tầm quan trọng của việc cải thiện hệ thống

Cải thiện hệ thống đặt phòng trực tuyến đóng vai trò rất quan trọng trong việc nâng cao trải nghiệm người dùng, tăng cường hiệu quả kinh doanh, và duy trì sự cạnh tranh trên thị trường. Dưới đây là những lý do cụ thể tại sao việc cải thiện hệ thống là cần thiết:

1. Nâng Cao Trải Nghiệm Người Dùng

Giao Diện Thân Thiện và Dễ Sử Dụng: Một hệ thống đặt phòng với giao diện thân thiện và dễ sử dụng giúp khách hàng dễ dàng tìm kiếm và đặt chỗ, từ đó nâng cao sự hài lòng và trung thành của khách hàng.

Tốc Độ và Hiệu Suất: Cải thiện tốc độ tải trang và hiệu suất hệ thống giúp giảm thiểu thời gian chờ đợi của khách hàng, tăng cường trải nghiệm người dùng.

2. Tăng Tỷ Lệ Chuyển Đổi

Quy Trình Đặt Chỗ Đơn Giản: Đơn giản hóa quy trình đặt chỗ, giảm số bước cần thiết và loại bỏ các bước phức tạp không cần thiết giúp tăng tỷ lệ chuyển đổi từ người xem thành người đặt chỗ.

Tích Hợp Thanh Toán An Toàn: Đảm bảo tích hợp các phương thức thanh toán an toàn và thuận tiện, giảm thiểu tỷ lệ từ chối thanh toán và tăng số lượng giao dịch thành công.

3. Tối Ưu Hóa Hoạt Động Kinh Doanh

Quản Lý Hiệu Quả: Các tính năng quản lý đặt chỗ và dữ liệu khách hàng được cải thiện giúp doanh nghiệp theo dõi và quản lý hiệu quả hơn các giao dịch và khách hàng.

Phân Tích và Báo Cáo: Hệ thống phân tích và báo cáo mạnh mẽ giúp doanh nghiệp hiểu rõ hơn về hành vi của khách hàng và xu hướng thị trường, từ đó đưa ra các chiến lược kinh doanh phù hợp.

4. Duy Trì và Mở Rộng Thị Phần

Cạnh Tranh Trên Thị Trường: Cải thiện liên tục hệ thống giúp doanh nghiệp duy trì lợi thế cạnh tranh và không bị tụt hậu so với các đối thủ.

Thu Hút Khách Hàng Mới: Những tính năng mới và cải tiến trong hệ thống có thể thu hút thêm nhiều khách hàng mới, mở rộng thị phần và tăng doanh thu.

5. Tăng Cường Bảo Mật và Tin Cậy

Bảo Vệ Dữ Liệu: Cải thiện các biện pháp bảo mật giúp bảo vệ dữ liệu cá nhân và thông tin thanh toán của khách hàng, xây dựng niềm tin và tăng cường độ tin cậy của hệ thống.

Đáp Ứng Yêu Cầu Pháp Lý: Đảm bảo hệ thống tuân thủ các quy định và tiêu chuẩn pháp lý liên quan đến bảo mật và quyền riêng tư dữ liệu.

6. Phản Hồi và Hỗ Trợ Khách Hàng Tốt Hơn

Hỗ Trợ Trực Tuyến: Tích hợp các công cụ hỗ trợ trực tuyến như chatbots và hỗ trợ khách hàng 24/7 giúp giải quyết nhanh chóng các vấn đề của khách hàng.

Cải Tiến FAQ và Tài Liệu Hướng Dẫn: Cập nhật và cải thiện các tài liệu hướng dẫn và mục FAQ giúp khách hàng tự giải quyết các vấn đề thường gặp mà không cần liên hệ hỗ trợ.

7. Tích Hợp Công Nghệ Mới

AI và Machine Learning: Sử dụng AI và Machine Learning để cá nhân hóa trải nghiệm người dùng, gợi ý các dịch vụ phù hợp và dự đoán xu hướng.

IoT và Smart Devices: Tích hợp với các thiết bị thông minh và công nghệ IoT để cung cấp các dịch vụ tiên tiến như mở khóa phòng khách sạn bằng điện thoại di động.

## 2.3 Laravel: Framework phát triển ứng dụng web

### 2.3.1 Giới thiệu về Laravel

1. Laravel là gì?

Laravel là một framework web PHP mã nguồn mở phổ biến được sử dụng để xây dựng các ứng dụng web. Nó giúp các nhà phát triển xây dựng các ứng dụng web bằng cách sử dụng mẫu kiến ​​trúc Model-View-Controller (MVC).

Laravel được biết đến với cú pháp tinh gọn, cách tiếp cận thân thiện với nhà phát triển và bộ tính năng phong phú giúp đơn giản hóa các tác vụ phát triển web phổ biến. Nó tận dụng nhiều tính năng nâng cao của PHP, chẳng hạn như lập trình hướng đối tượng (OOP) để cung cấp cú pháp rõ ràng, dễ dàng viết mã và có thể duy trì, mở rộng.

1. Tại sao nên chọn Laravel?

Cú Pháp Dễ Đọc và Dễ Hiểu:

Laravel được thiết kế với cú pháp rõ ràng và dễ hiểu, giúp các nhà phát triển dễ dàng học và làm việc với framework này.

Hỗ Trợ Cộng Đồng Mạnh Mẽ:

Laravel có một cộng đồng lớn và tích cực, cung cấp nhiều tài liệu, video hướng dẫn và các gói mở rộng (package) để hỗ trợ quá trình phát triển.

Tích Hợp Công Cụ Phát Triển:

Các công cụ như Laravel Mix (hỗ trợ biên dịch CSS và JavaScript), Horizon (quản lý hàng đợi công việc), và Telescope (giám sát ứng dụng) giúp cải thiện năng suất và quản lý dự án dễ dàng hơn.

Bảo Mật:

Laravel cung cấp nhiều tính năng bảo mật tích hợp như mã hóa dữ liệu, bảo vệ CSRF (Cross-Site Request Forgery), và bảo vệ XSS (Cross-Site Scripting), giúp bảo vệ ứng dụng khỏi các lỗ hổng bảo mật phổ biến.

Khả Năng Mở Rộng và Tùy Biến:

Laravel cho phép mở rộng và tùy biến dễ dàng nhờ hệ thống package và service container, giúp bạn có thể tích hợp và sử dụng các thư viện bên ngoài hoặc viết các package tùy chỉnh theo nhu cầu.

### 2.3.2 Các tính năng và ưu điểm của Laravel

1. Các tính năng của Laravel

Eloquent ORM (Object-Relational Mapping):

ORM mạnh mẽ: Eloquent cung cấp một cách tiếp cận đơn giản và đẹp mắt để tương tác với cơ sở dữ liệu. Nó sử dụng các mô hình để đại diện cho các bảng trong cơ sở dữ liệu và cung cấp các phương thức để thực hiện các thao tác CRUD (Create, Read, Update, Delete).

Blade Templating Engine:

Templating linh hoạt: Blade là engine templating mạnh mẽ của Laravel, cho phép sử dụng các cấu trúc điều khiển như if, for, và foreach trong HTML. Blade hỗ trợ template inheritance, giúp việc tạo và quản lý các layout trở nên dễ dàng hơn.

Routing:

Hệ thống định tuyến mạnh mẽ: Laravel cung cấp một hệ thống định tuyến đơn giản và trực quan, cho phép định nghĩa các route ứng với các URI và phương thức HTTP cụ thể. Hỗ trợ route groups, middleware và route parameter binding.

Middleware:

Xử lý yêu cầu trung gian: Middleware trong Laravel là các lớp trung gian để xử lý các yêu cầu HTTP trước khi chúng đến controller. Middleware có thể được sử dụng để kiểm tra quyền truy cập, xác thực người dùng, hoặc xử lý các tác vụ trước và sau khi yêu cầu được xử lý.

Artisan Console:

Công cụ dòng lệnh mạnh mẽ: Artisan là công cụ dòng lệnh tích hợp của Laravel, cung cấp nhiều lệnh hữu ích để hỗ trợ phát triển ứng dụng. Bạn có thể tạo mô hình, controller, middleware và thực hiện nhiều tác vụ khác thông qua Artisan.

Authentication & Authorization:

Xác thực và ủy quyền: Laravel cung cấp các hệ thống xác thực và ủy quyền tích hợp, giúp dễ dàng quản lý người dùng và quyền truy cập vào các phần khác nhau của ứng dụng. Bao gồm sẵn các tính năng như đăng ký, đăng nhập, quên mật khẩu, và reset mật khẩu.

Task Scheduling:

Lập lịch công việc: Laravel cung cấp một API để định nghĩa và quản lý các tác vụ theo lịch trình. Tính năng này giúp tự động hóa các tác vụ định kỳ như gửi email, dọn dẹp dữ liệu hoặc sao lưu cơ sở dữ liệu.

Testing:

Kiểm thử mạnh mẽ: Laravel tích hợp PHPUnit và cung cấp các công cụ để kiểm thử đơn vị và kiểm thử tích hợp dễ dàng. Bạn có thể viết các test case cho các phần khác nhau của ứng dụng và kiểm tra chúng một cách tự động.

Event Broadcasting:

Phát sự kiện thời gian thực: Laravel hỗ trợ phát sự kiện qua WebSocket bằng cách sử dụng thư viện như Pusher. Tính năng này rất hữu ích để xây dựng các ứng dụng thời gian thực như chat, thông báo và cập nhật dữ liệu trực tiếp.

Database Migrations & Seeders:

Quản lý cơ sở dữ liệu dễ dàng: Migrations và Seeders giúp quản lý và phiên bản hóa cấu trúc cơ sở dữ liệu. Bạn có thể dễ dàng thay đổi cấu trúc cơ sở dữ liệu và tạo dữ liệu mẫu.

Queueing:

Xử lý hàng đợi công việc: Laravel cung cấp hệ thống xử lý hàng đợi để quản lý các tác vụ nền như gửi email, xử lý hình ảnh, giúp tăng hiệu suất và khả năng đáp ứng của ứng dụng.

File Storage:

Lưu trữ file: Laravel cung cấp một hệ thống lưu trữ file thống nhất, cho phép lưu trữ file cục bộ hoặc trên các dịch vụ lưu trữ đám mây như Amazon S3, Google Cloud Storage.

1. Ưu điểm của Lavarel

Cú Pháp Dễ Đọc và Dễ Hiểu:

Laravel được thiết kế với cú pháp rõ ràng và dễ hiểu, giúp các nhà phát triển dễ dàng học và làm việc với framework này. Điều này giúp giảm thiểu thời gian và công sức cần thiết để phát triển và bảo trì ứng dụng.

Hỗ Trợ Cộng Đồng Mạnh Mẽ:

Laravel có một cộng đồng lớn và tích cực, cung cấp nhiều tài liệu, video hướng dẫn và các gói mở rộng (package) để hỗ trợ quá trình phát triển. Laravel cũng có nhiều sự kiện và hội nghị để kết nối và học hỏi từ các nhà phát triển khác.

Tích Hợp Công Cụ Phát Triển:

Laravel đi kèm với nhiều công cụ phát triển hữu ích như Laravel Mix (hỗ trợ biên dịch CSS và JavaScript), Horizon (quản lý hàng đợi công việc), và Telescope (giám sát ứng dụng). Các công cụ này giúp cải thiện năng suất và quản lý dự án dễ dàng hơn.

Bảo Mật:

Laravel cung cấp nhiều tính năng bảo mật tích hợp như mã hóa dữ liệu, bảo vệ CSRF (Cross-Site Request Forgery), và bảo vệ XSS (Cross-Site Scripting). Điều này giúp bảo vệ ứng dụng khỏi các lỗ hổng bảo mật phổ biến.

Khả Năng Mở Rộng và Tùy Biến:

Laravel cho phép mở rộng và tùy biến dễ dàng nhờ hệ thống package và service container. Bạn có thể tích hợp và sử dụng các thư viện bên ngoài hoặc viết các package tùy chỉnh theo nhu cầu của mình.

Hiệu Suất Cao:

Laravel được tối ưu hóa để mang lại hiệu suất cao, giúp ứng dụng của bạn chạy nhanh và hiệu quả. Các tính năng như caching, queueing, và công cụ profiling giúp tối ưu hóa hiệu suất ứng dụng.

Quản Lý Dự Án Dễ Dàng:

Các công cụ và tính năng của Laravel giúp quản lý dự án dễ dàng, từ việc tổ chức mã nguồn đến triển khai và bảo trì. Laravel Forge và Envoyer là các dịch vụ hỗ trợ triển khai và quản lý ứng dụng trên các máy chủ.

# CHƯƠNG 3. THIẾT KẾ VÀ PHÁT TRIỂN ĐỀ TÀI

## 3.1 Thiết kế và mô hình hóa

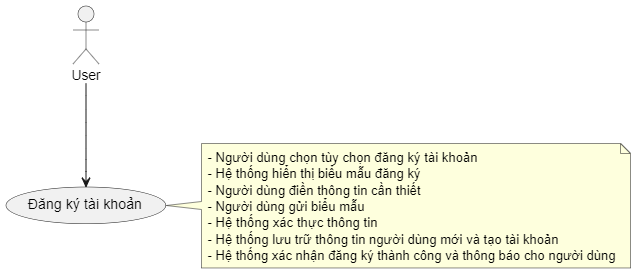
### 3.1.1 Phân tích yêu cầu của website và xây dựng use case

1. Sơ đồ use case

Vẽ lại

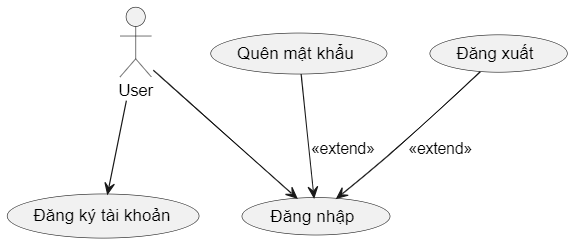
Đặc tả Use case của chức năng chính của người dùng

* Use case: Đăng ký tài khoản



|  |  |
| --- | --- |
| Tên Use case | Đăng ký tài khoản |
| Tác nhân chính | Người dùng |
| Mô tả | Người dùng có thể đăng ký tài khoản mới để sử dụng |
| Tiền điều kiện | Người dùng chưa có tài khoản |
| Hậu điều kiện | Người dùng có tài khoản mới trong hệ thống |
| Luồng sự kiện chính | 1. Người dùng chọn tùy chọn đăng ký tài khoản.  2. Hệ thống hiển thị biểu mẫu đăng ký.  3. Người dùng điền thông tin cần thiết (tên, email, mật khẩu, v.v.).  4. Người dùng gửi biểu mẫu.  5. Hệ thống xác thực thông tin.  6. Hệ thống lưu trữ thông tin người dùng mới và tạo tài khoản.  7. Hệ thống xác nhận đăng ký thành công và thông báo cho người dùng. |
| Luồng sự kiện phụ | 5a. Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin. |

* Use case: Đăng nhập tài khoản



|  |  |
| --- | --- |
| Tên Use case | Đăng nhập tài khoản |
| Tác nhân chính | Người dùng |
| Mô tả | Người dùng có thể đăng nhập vào hệ thống bằng tài khoản đã đăng ký trước đó để sử dụng các tính năng của hệ thống. |
| Tiền điều kiện | Người dùng đã có tài khoản trong hệ thống. |
| Hậu điều kiện | Người dùng đã đăng nhập thành công vào hệ thống. |
| Luồng sự kiện chính | 1. Người dùng truy cập vào trang đăng nhập trên hệ thống.  2. Hệ thống hiển thị biểu mẫu đăng nhập (yêu cầu nhập tên người dùng và mật khẩu).  3. Người dùng nhập thông tin đăng nhập (tên người dùng và mật khẩu).  4. Người dùng nhấn nút "Đăng nhập".  5. Hệ thống xác thực thông tin đăng nhập.  6. Nếu thông tin đúng, hệ thống cho phép người dùng truy cập vào hệ thống và chuyển hướng đến trang chính của ứng dụng.  7. Hệ thống thông báo cho người dùng biết rằng họ đã đăng nhập thành công. |
| Luồng sự kiện phụ | 5a. Nếu thông tin đăng nhập không chính xác, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin đăng nhập. |

Use case: Tìm kiếm phòng



|  |  |
| --- | --- |
| Tên Use case | Tìm kiếm phòng |
| Tác nhân chính | Người dùng |
| Mô tả | Khách hàng có thể tìm kiếm phòng khách sạn dựa trên các tiêu chí dựa vào loại phòng. |
| Tiền điều kiện | Khách hàng muốn tìm kiếm và đặt phòng khách sạn.  Khách hàng đã truy cập vào hệ thống tìm kiếm phòng khách sạn. |
| Hậu điều kiện | Khách hàng đã xem được kết quả tìm kiếm và chọn được phòng phù hợp. |
| Luồng sự kiện chính | 1. Khách hàng truy cập vào giao diện tìm kiếm phòng khách sạn.  2. Hệ thống hiển thị các tùy chọn tìm kiếm như loại phòng.  3. Hệ thống thực hiện tìm kiếm dựa trên thông tin đã chọn.  4. Hệ thống hiển thị kết quả tìm kiếm cho khách hàng.  5. Khách hàng thực hiện đặt phòng và hoàn tất giao dịch. |
| Luồng sự kiện phụ | 3a. Nếu không có kết quả tìm kiếm phù hợp, hệ thống thông báo cho khách hàng biết và đề xuất các tùy chọn khác. |

Use case: Xem chi tiết phòng



|  |  |
| --- | --- |
| Tên Use case | Xem chi tiết phòng |
| Tác nhân chính | Người dùng |
| Mô tả | Khách hàng có thể xem chi tiết về một phòng khách sạn, bao gồm các thông tin như loại phòng, tiện nghi, giá cả, hình ảnh, vv. |
| Tiền điều kiện | Khách hàng đã tìm kiếm và chọn được một phòng khách sạn để xem chi tiết.Khách hàng đã truy cập vào hệ thống tìm kiếm phòng khách sạn. |
| Hậu điều kiện | Khách hàng đã xem thông tin chi tiết về phòng khách sạn và có thể quay lại màn hình trước đó. |
| Luồng sự kiện chính | 1. Khách hàng chọn một phòng khách sạn để xem chi tiết.  2. Hệ thống hiển thị thông tin chi tiết về phòng khách sạn, bao gồm loại phòng, tiện nghi, giá cả, hình ảnh, vv.  3. Khách hàng duyệt qua thông tin và có thể xem hình ảnh chi tiết của phòng.  4. Sau khi đã xem thông tin đầy đủ, khách hàng có thể quay lại màn hình trước đó. |
| Luồng sự kiện phụ |  |

Use case: Đặt phòng



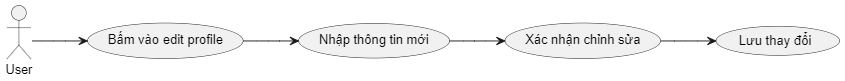
|  |  |
| --- | --- |
| Tên Use case | Đặt phòng |
| Tác nhân chính | Người dùng |
| Mô tả | Khách hàng có thể tìm kiếm, chọn loại phòng, nhập thông tin đặt phòng, thêm vào giỏ hàng, thực hiện thanh toán và xác nhận thanh toán. |
| Tiền điều kiện | Khách hàng đã truy cập vào hệ thống đặt phòng và thanh toán trực tuyến của khách sạn. |
| Hậu điều kiện | Hệ thống đã ghi nhận thông tin đặt phòng và thanh toán.  Khách hàng đã nhận được xác nhận thanh toán. |
| Luồng sự kiện chính | 1. Khách hàng truy cập vào giao diện tìm kiếm phòng khách sạn.  2. Hệ thống hiển thị các tùy chọn tìm kiếm như loại phòng, ngày đến, ngày đi, số lượng phòng, vv.  3. Khách hàng chọn loại phòng, nhập thông tin đặt phòng và nhấn nút "Tìm kiếm".  4. Hệ thống hiển thị kết quả tìm kiếm và cho phép khách hàng chọn phòng và thêm vào giỏ hàng.  5. Khách hàng kiểm tra thông tin giỏ hàng và thực hiện thanh toán.  6. Hệ thống xử lý thanh toán và gửi xác nhận thanh toán cho khách hàng. |
| Luồng sự kiện phụ | 5a. Nếu thanh toán không thành công, hệ thống thông báo lỗi và yêu cầu khách hàng thử lại hoặc sử dụng phương thức thanh toán khác. |

Use case: Xem lịch sử đặt phòng

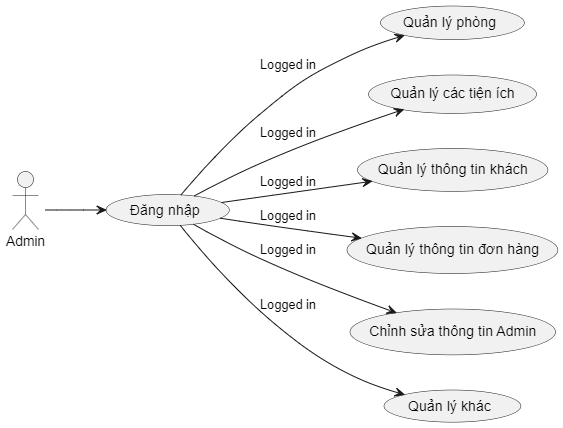


|  |  |
| --- | --- |
| Tên Use case | Xem lịch sử đặt phòng |
| Tác nhân chính | Người dùng |
| Mô tả | Khách hàng có thể xem lịch sử đặt phòng của mình cũng như chi tiết của mỗi đơn đặt phòng trong lịch sử. |
| Tiền điều kiện | Khách hàng đã đăng nhập vào hệ thống đặt phòng khách sạn. |
| Hậu điều kiện | Khách hàng đã xem được lịch sử đặt phòng và chi tiết của mỗi đơn đặt phòng trong lịch sử. |
| Luồng sự kiện chính | 1. Khách hàng truy cập vào giao diện xem lịch sử và chi tiết đặt phòng.  2. Hệ thống hiển thị lịch sử đặt phòng của khách hàng.  3. Khách hàng chọn một đơn đặt phòng từ lịch sử để xem chi tiết.  4. Hệ thống hiển thị chi tiết của đơn đặt phòng được chọn. |
| Luồng sự kiện phụ | 3a. Nếu không có đơn đặt phòng nào trong lịch sử, hệ thống thông báo cho khách hàng biết và không hiển thị chi tiết nào |

Use case: Chỉnh sửa thông tin cá nhân



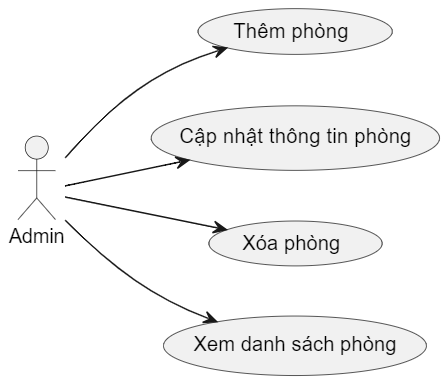
|  |  |
| --- | --- |
| Tên Use case | Chỉnh sửa thông tin cá nhân |
| Tác nhân chính | Người dùng |
| Mô tả | Người dùng có thể chỉnh sửa thông tin cá nhân của mình trong hồ sơ người dùng, bao gồm thông tin như tên, địa chỉ, số điện thoại, vv. |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống và có quyền truy cập vào chức năng chỉnh sửa hồ sơ. |
| Hậu điều kiện | Thay đổi trong hồ sơ người dùng đã được lưu và cập nhật thành công trong hệ thống. |
| Luồng sự kiện chính | 1. Người dùng truy cập vào giao diện chỉnh sửa hồ sơ.  2. Hệ thống hiển thị thông tin cá nhân hiện tại của người dùng.  3. Người dùng thay đổi thông tin cần chỉnh sửa (ví dụ: tên, địa chỉ, số điện thoại, vv.).  4. Người dùng xác nhận các thay đổi đã thực hiện.  5. Hệ thống lưu các thay đổi và cập nhật hồ sơ người dùng. |
| Luồng sự kiện phụ | 3a. Nếu người dùng quyết định không thay đổi bất kỳ thông tin nào, họ có thể hủy bỏ quá trình chỉnh sửa và quay lại giao diện trước đó. |

Đặc tả Use case của chức năng chính của admin

Use case: Đăng nhập

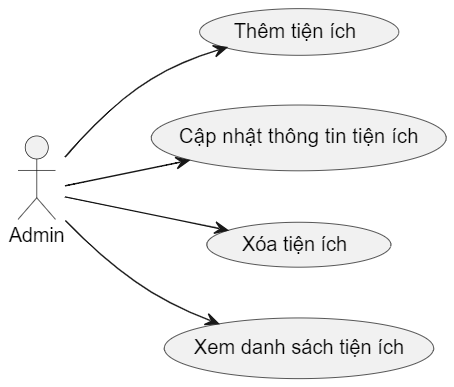
|  |  |
| --- | --- |
| Tên Use case | Đăng nhập tài khoản |
| Tác nhân chính | Admin |
| Mô tả | Người dùng có thể đăng nhập vào hệ thống bằng cách cung cấp thông tin đăng nhập hợp lệ. |
| Tiền điều kiện | Hệ thống đã được khởi động và hiển thị giao diện đăng nhập. |
| Hậu điều kiện | Người dùng đã đăng nhập thành công vào hệ thống và được chuyển hướng đến giao diện chính. |
| Luồng sự kiện chính | 1. Người dùng truy cập vào giao diện đăng nhập của hệ thống.  2. Người dùng nhập tên người dùng (username) và mật khẩu (password).  3. Người dùng nhấn nút "Đăng nhập".  4. Hệ thống xác nhận thông tin đăng nhập.  5. Nếu thông tin đăng nhập hợp lệ, hệ thống chuyển hướng người dùng đến giao diện chính của hệ thống.  6. Nếu thông tin đăng nhập không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin đăng nhập. |
| Luồng sự kiện phụ | 5a. Nếu thông tin đăng nhập không hợp lệ, người dùng có thể chọn "Quên mật khẩu" để thiết lập lại mật khẩu hoặc liên hệ với quản trị viên để hỗ trợ. |

Use case: Quản lý phòng



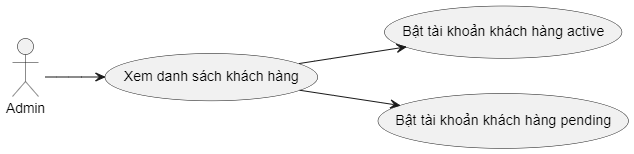
|  |  |
| --- | --- |
| Tên Use case | Quản lý phòng |
| Tác nhân chính | Admin |
| Mô tả | Quản trị viên có thể quản lý thông tin về các phòng trong hệ thống, bao gồm thêm phòng mới, cập nhật thông tin phòng, xóa phòng và xem danh sách phòng. |
| Tiền điều kiện | Quản trị viên đã đăng nhập vào hệ thống. |
| Hậu điều kiện | Các thông tin về phòng được quản lý đúng cách trong hệ thống. |
| Luồng sự kiện chính | 1. Quản trị viên đăng nhập vào hệ thống.  - Quản trị viên chọn tùy chọn "Quản lý phòng" từ giao diện quản lý.  2. Thêm phòng:  1. Quản trị viên chọn tùy chọn "Thêm phòng".  2. Hệ thống hiển thị biểu mẫu nhập thông tin phòng mới.  3. Quản trị viên nhập thông tin phòng (số phòng, loại phòng, giá, tiện nghi, vv.).  4. Quản trị viên xác nhận thêm phòng.  5. Hệ thống lưu trữ thông tin phòng mới vào cơ sở dữ liệu.  6. Hệ thống hiển thị thông báo thành công và cập nhật danh sách phòng.  3. Cập nhật thông tin phòng:  1. Quản trị viên chọn tùy chọn "Cập nhật thông tin phòng" từ danh sách phòng.  2. Hệ thống hiển thị biểu mẫu thông tin phòng hiện tại.  3. Quản trị viên thay đổi các thông tin cần cập nhật.  4. Quản trị viên xác nhận cập nhật thông tin.  5. Hệ thống lưu trữ thông tin phòng đã cập nhật vào cơ sở dữ liệu.  6. Hệ thống hiển thị thông báo thành công và cập nhật danh sách phòng.  4. Xóa phòng:  1. Quản trị viên chọn tùy chọn "Xóa phòng" từ danh sách phòng.  2. Hệ thống hiển thị yêu cầu xác nhận xóa phòng.  3. Quản trị viên xác nhận xóa phòng.  4. Hệ thống xóa thông tin phòng khỏi cơ sở dữ liệu.  5. Hệ thống hiển thị thông báo thành công và cập nhật danh sách phòng.  5. Xem danh sách phòng:  1. Quản trị viên chọn tùy chọn "Xem danh sách phòng" từ giao diện quản lý.  2. Hệ thống hiển thị danh sách tất cả các phòng hiện có trong hệ thống, bao gồm thông tin chi tiết của từng phòng. |
| Luồng sự kiện phụ | Thêm phòng:  - Nếu thông tin nhập vào không hợp lệ hoặc thiếu, hệ thống hiển thị thông báo lỗi và yêu cầu quản trị viên nhập lại thông tin.  - Cập nhật thông tin phòng:  - Nếu thông tin cập nhật không hợp lệ hoặc thiếu, hệ thống hiển thị thông báo lỗi và yêu cầu quản trị viên nhập lại thông tin.  - Xóa phòng:  - Nếu quản trị viên hủy bỏ yêu cầu xóa, hệ thống không thực hiện hành động xóa và quay lại danh sách phòng |

Use case: Quản lý các tiện ích



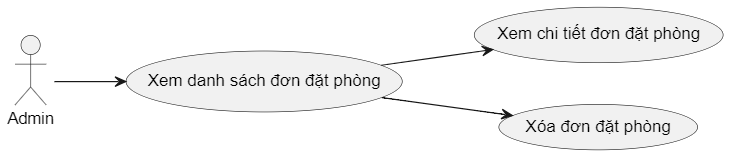
|  |  |
| --- | --- |
| Tên Use case | Quản lý các tiện ích |
| Tác nhân chính | Admin |
| Mô tả | Quản trị viên có thể quản lý thông tin về các tiện ích trong hệ thống, bao gồm thêm tiện ích mới, cập nhật thông tin tiện ích, xóa tiện ích và xem danh sách tiện ích. |
| Tiền điều kiện | Quản trị viên đã đăng nhập vào hệ thống. |
| Hậu điều kiện | Các thông tin về tiện ích được quản lý đúng cách trong hệ thống. |
| Luồng sự kiện chính | 1. Quản trị viên đăng nhập vào hệ thống.  - Quản trị viên chọn tùy chọn "Quản lý tiện ích" từ giao diện quản lý.  2. Thêm tiện ích:  1. Quản trị viên chọn tùy chọn "Thêm tiện ích".  2. Hệ thống hiển thị biểu mẫu nhập thông tin tiện ích mới.  3. Quản trị viên nhập thông tin tiện ích (tên, mô tả, loại, v.v.).  4. Quản trị viên xác nhận thêm tiện ích.  5. Hệ thống lưu trữ thông tin tiện ích mới vào cơ sở dữ liệu.  6. Hệ thống hiển thị thông báo thành công và cập nhật danh sách tiện ích.  3. Cập nhật thông tin tiện ích:  1. Quản trị viên chọn tùy chọn "Cập nhật thông tin tiện ích" từ danh sách tiện ích.  2. Hệ thống hiển thị biểu mẫu thông tin tiện ích hiện tại.  3. Quản trị viên thay đổi các thông tin cần cập nhật.  4. Quản trị viên xác nhận cập nhật thông tin.  5. Hệ thống lưu trữ thông tin tiện ích đã cập nhật vào cơ sở dữ liệu.  6. Hệ thống hiển thị thông báo thành công và cập nhật danh sách tiện ích.  4. Xóa tiện ích:  1. Quản trị viên chọn tùy chọn "Xóa tiện ích" từ danh sách tiện ích.  2. Hệ thống hiển thị yêu cầu xác nhận xóa tiện ích.  3. Quản trị viên xác nhận xóa tiện ích.  4. Hệ thống xóa thông tin tiện ích khỏi cơ sở dữ liệu.  5. Hệ thống hiển thị thông báo thành công và cập nhật danh sách tiện ích.  5. Xem danh sách tiện ích:  1. Quản trị viên chọn tùy chọn "Xem danh sách tiện ích" từ giao diện quản lý.  2. Hệ thống hiển thị danh sách tất cả các tiện ích hiện có trong hệ thống, bao gồm thông tin chi tiết của từng tiện ích. |
| Luồng sự kiện phụ | - Thêm tiện ích:  - Nếu thông tin nhập vào không hợp lệ hoặc thiếu, hệ thống hiển thị thông báo lỗi và yêu cầu quản trị viên nhập lại thông tin.  - Cập nhật thông tin tiện ích:  - Nếu thông tin cập nhật không hợp lệ hoặc thiếu, hệ thống hiển thị thông báo lỗi và yêu cầu quản trị viên nhập lại thông tin.  - Xóa tiện ích:  - Nếu quản trị viên hủy bỏ yêu cầu xóa, hệ thống không thực hiện hành động xóa và quay lại danh sách tiện ích. |

Use case: Quản lý thông tin khách hàng



|  |  |
| --- | --- |
| Tên Use case | Quản lý thông tin khách hàng |
| Tác nhân chính | Admin |
| Mô tả | Chức năng này cho phép quản trị viên quản lý thông tin khách hàng trong hệ thống bao gồm xem danh sách khách hàng và quản lý trạng thái tài khoản của khách hàng. |
| Tiền điều kiện | Quản trị viên đã đăng nhập vào hệ thống. |
| Hậu điều kiện |  |
| Luồng sự kiện chính | 1. Quản trị viên chọn một trong các chức năng quản lý thông tin khách hàng từ giao diện quản lý.  2. Hệ thống thực hiện các hành động tương ứng theo yêu cầu của quản trị viên và cập nhật thông tin khách hàng trong hệ thống. |
| Luồng sự kiện phụ | Nếu quản trị viên hủy bỏ một trong các hành động thực hiện, hệ thống không thực hiện hành động. |

Use case: Quản lý thông tin đơn đặt phòng



|  |  |
| --- | --- |
| Tên Use case | Quản lý đơn đặt phòng |
| Tác nhân chính | Admin |
| Mô tả | - Admin có thể xem danh sách các đơn đặt phòng.  - Admin có thể xem chi tiết của một đơn đặt phòng cụ thể.  - Admin có thể xóa một đơn đặt phòng nếu cần. |
| Tiền điều kiện | Quản trị viên đã đăng nhập vào hệ thống. |
| Hậu điều kiện |  |
| Luồng sự kiện chính | 1. Admin chọn tùy chọn "Xem danh sách đơn đặt phòng".  2. Hệ thống hiển thị danh sách các đơn đặt phòng.  3. Admin có thể chọn một đơn đặt phòng cụ thể để xem chi tiết.  4. Hệ thống hiển thị chi tiết của đơn đặt phòng được chọn.  5. Admin có thể chọn tùy chọn "Xóa đơn đặt phòng" để xóa đơn đặt phòng đó.  6. Hệ thống yêu cầu xác nhận xóa đơn đặt phòng từ Admin.  7. Sau khi xác nhận, hệ thống xóa đơn đặt phòng và cập nhật danh sách. |
| Luồng sự kiện phụ | - Nếu không có đơn đặt phòng nào tồn tại trong hệ thống, thông báo tương ứng được hiển thị.  - Nếu Admin không muốn xóa đơn đặt phòng, quá trình sẽ kết thúc và không có thay đổi nào được thực hiện. |

Use case: Chỉnh sửa thông tin admin

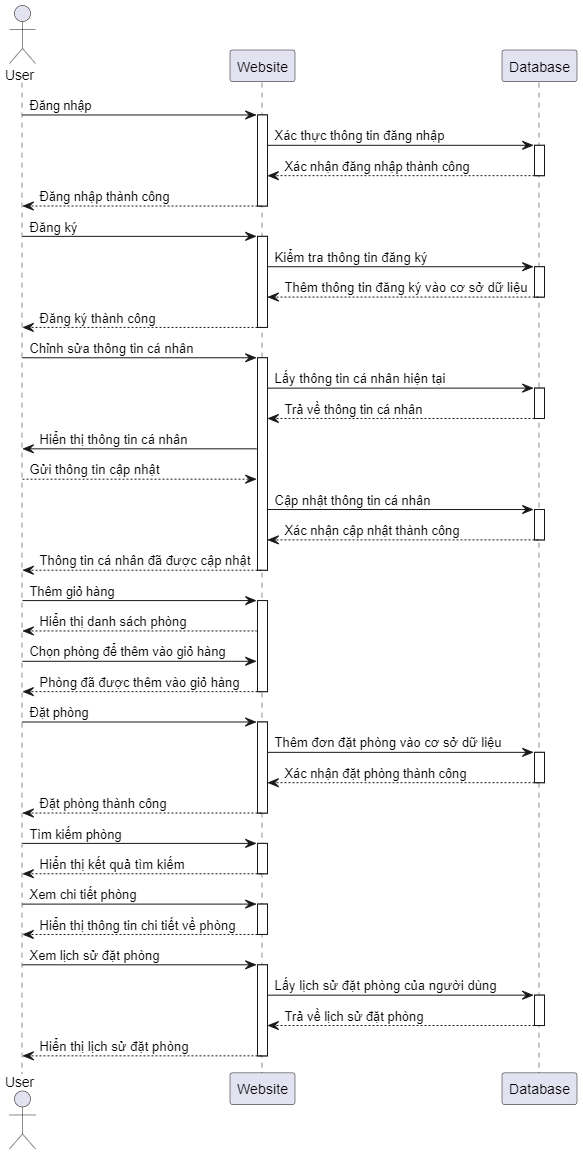


|  |  |
| --- | --- |
| Tên Use case | Chỉnh sửa thông tin Admin |
| Tác nhân chính | Admin |
| Mô tả | - Admin có thể chỉnh sửa thông tin cá nhân của mình.  - Sau khi chỉnh sửa, Admin có thể lưu thông tin đã thay đổi. |
| Tiền điều kiện | Quản trị viên đã đăng nhập vào hệ thống. |
| Hậu điều kiện |  |
| Luồng sự kiện chính | 1. Admin chọn tùy chọn "Chỉnh sửa thông tin Admin".  2. Hệ thống hiển thị một giao diện cho phép Admin chỉnh sửa thông tin cá nhân.  3. Admin cập nhật thông tin cá nhân theo ý muốn.  4. Sau khi đã hoàn tất việc chỉnh sửa, Admin chọn tùy chọn "Lưu thông tin".  5. Hệ thống lưu lại thông tin đã được chỉnh sửa và cập nhật trong cơ sở dữ liệu. |
| Luồng sự kiện phụ | - Nếu Admin không muốn lưu thông tin đã chỉnh sửa, có thể chọn tùy chọn "Hủy" hoặc "Quay lại" để thoát khỏi giao diện chỉnh sửa mà không có thay đổi nào được lưu lại. |

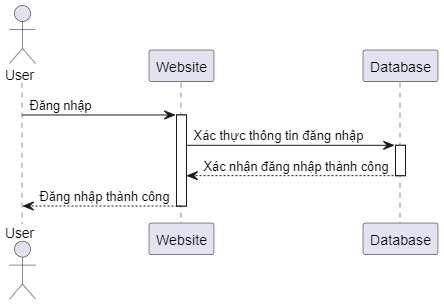
Use case: Quản lý khác

1. Sơ đồ tuần tự

* Sơ đồ tuần tự cho người dùng



1. Đăng nhập:

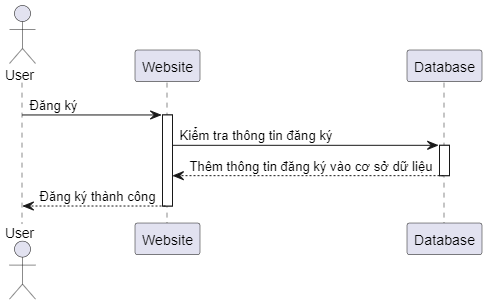


- Người dùng bắt đầu bằng cách nhập thông tin đăng nhập vào trang web.

- Trang web gửi thông tin đăng nhập tới cơ sở dữ liệu để xác thực.

- Sau khi thông tin đăng nhập được xác thực, trang web hiển thị thông báo đăng nhập thành công cho người dùng.

2. Đăng ký:



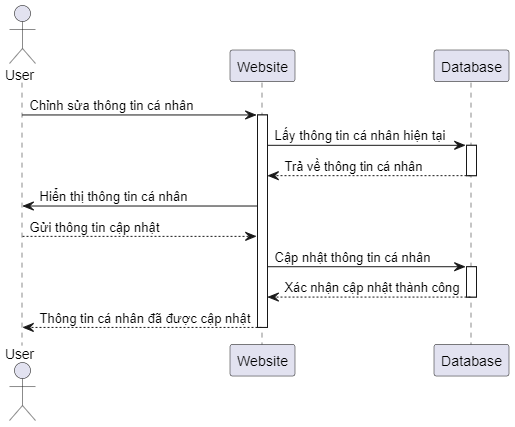
- Nếu người dùng chưa có tài khoản, họ có thể chọn tùy chọn "Đăng ký".

- Trang web yêu cầu người dùng cung cấp thông tin cần thiết cho việc đăng ký.

- Thông tin đăng ký được gửi tới cơ sở dữ liệu để kiểm tra và thêm vào cơ sở dữ liệu nếu thông tin hợp lệ.

- Sau khi đăng ký thành công, trang web hiển thị thông báo đăng ký thành công cho người dùng.

3. Chỉnh sửa thông tin cá nhân:



- Người dùng có thể chọn tùy chọn để chỉnh sửa thông tin cá nhân của mình trên trang web.

- Trang web gửi yêu cầu để lấy thông tin cá nhân hiện tại từ cơ sở dữ liệu.

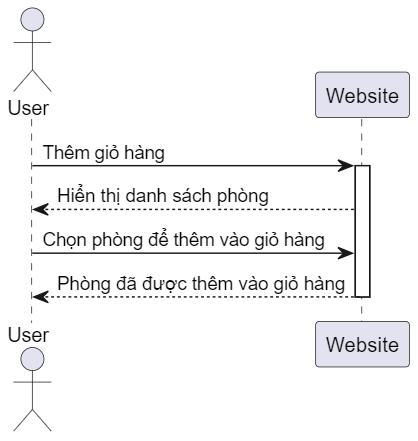
- Thông tin cá nhân được trả về và hiển thị cho người dùng.

- Người dùng có thể cập nhật thông tin cá nhân và gửi lại cho trang web.

- Trang web gửi thông tin cập nhật tới cơ sở dữ liệu để lưu trữ thông tin mới.

- Sau khi thông tin được cập nhật thành công, trang web thông báo cho người dùng biết rằng thông tin cá nhân đã được cập nhật thành công.

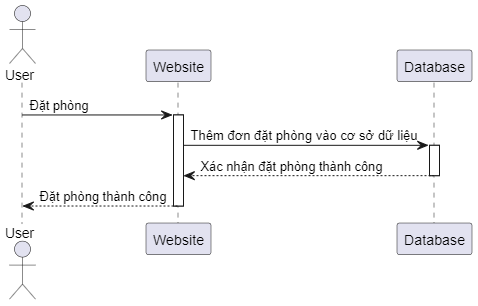
4. Thêm giỏ hàng:



- Người dùng có thể duyệt qua danh sách phòng trên trang web và chọn phòng muốn thêm vào giỏ hàng.

- Sau khi chọn, trang web hiển thị thông báo xác nhận rằng phòng đã được thêm vào giỏ hàng của người dùng.

5. Đặt phòng:

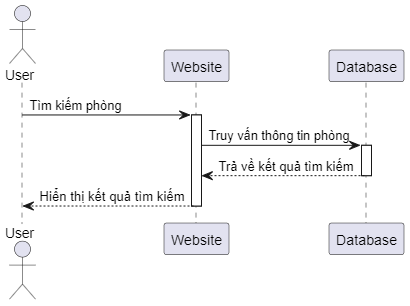


- Người dùng chọn phòng muốn đặt trên trang web.

- Thông tin đặt phòng được gửi tới cơ sở dữ liệu để lưu trữ.

- Sau khi đặt phòng thành công, trang web thông báo cho người dùng biết rằng đặt phòng đã thành công.

6. Tìm kiếm phòng:



- Người dùng có thể sử dụng chức năng tìm kiếm trên trang web để tìm kiếm các phòng khách sạn theo các tiêu chí nhất định.

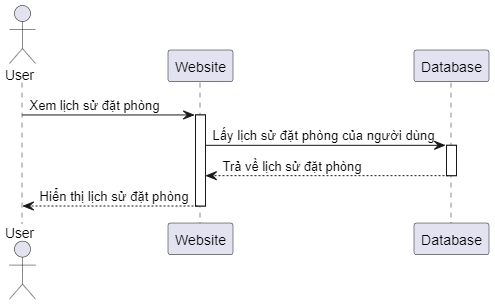
7. Xem chi tiết phòng:



- Người dùng có thể chọn xem thông tin chi tiết về một phòng cụ thể trên trang web.

- Thông tin chi tiết về phòng được hiển thị cho người dùng.

8. Xem lịch sử đặt phòng:

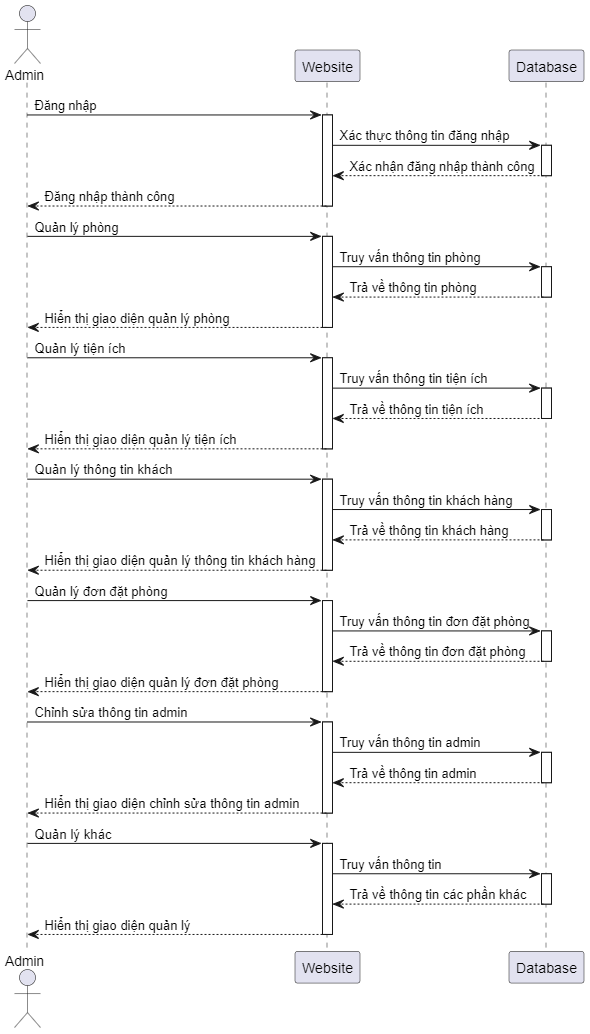


- Người dùng có thể xem lịch sử các đơn đặt phòng đã thực hiện trước đó trên trang web.

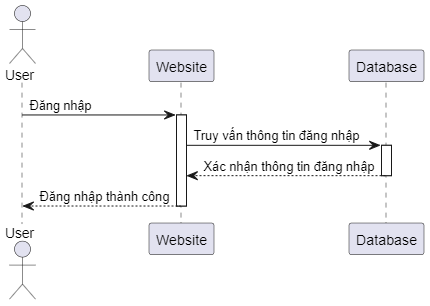
- Trang web gửi yêu cầu để lấy lịch sử đặt phòng từ cơ sở dữ liệu.

- Thông tin lịch sử đặt phòng được trả về và hiển thị cho người dùng.

* Sơ đồ tuần tự cho admin



1. Đăng nhập:

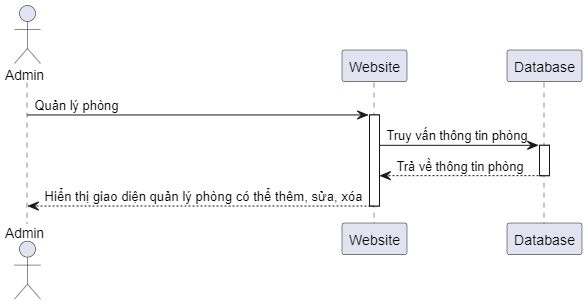


- Admin gửi yêu cầu đăng nhập thông qua giao diện web.

- Trang web chuyển yêu cầu đăng nhập đến cơ sở dữ liệu để xác thực thông tin.

- Sau khi xác thực thành công, trang web thông báo cho Admin về việc đăng nhập thành công.

2. Quản lý phòng:



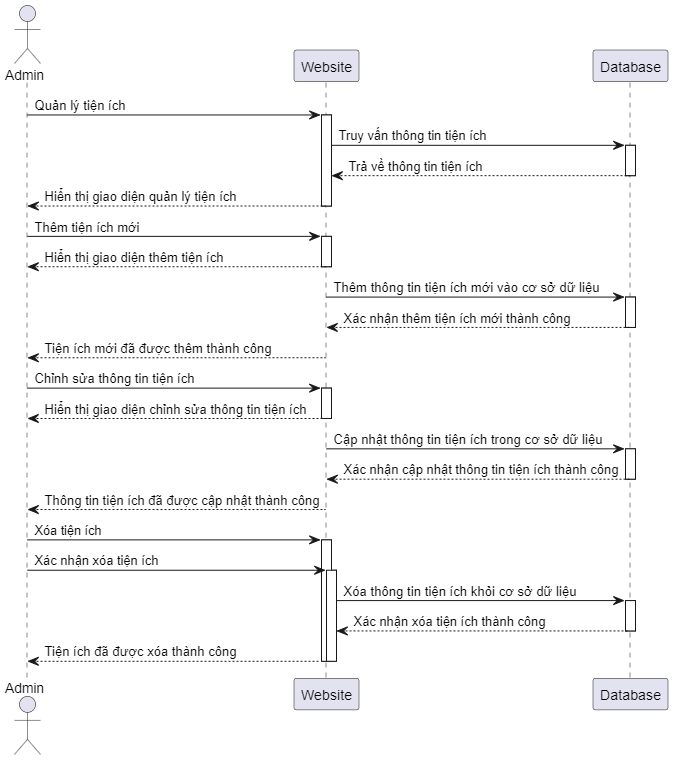
- Admin yêu cầu quản lý thông tin phòng thông qua giao diện web.

- Trang web gửi yêu cầu truy vấn thông tin phòng đến cơ sở dữ liệu.

- Cơ sở dữ liệu trả về thông tin phòng cho trang web.

- Trang web hiển thị giao diện quản lý phòng cho Admin.

3. Quản lý tiện ích:

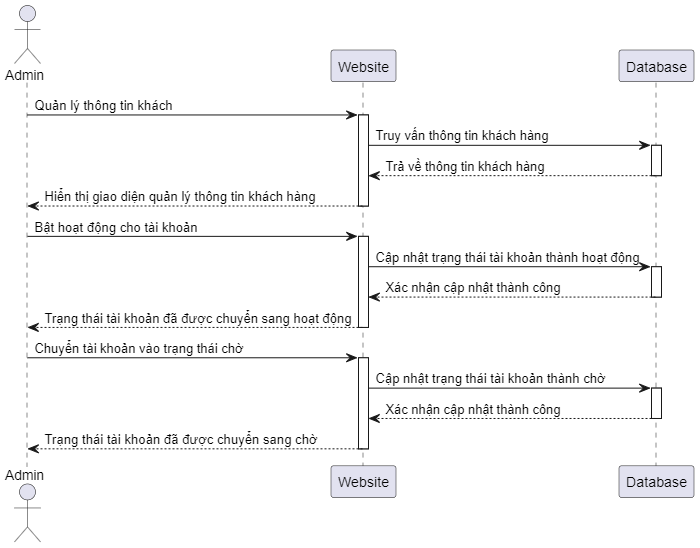


- Tương tự như quản lý phòng, Admin yêu cầu quản lý thông tin tiện ích.

- Trang web truy vấn thông tin tiện ích từ cơ sở dữ liệu và hiển thị kết quả cho Admin.

- Trang web có thể được thêm,sửa, xóa thông tin các tiện ích của phòng.

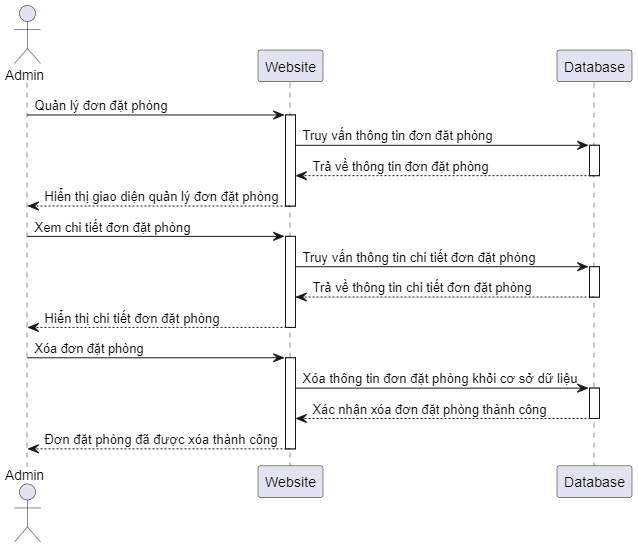
4. Quản lý thông tin khách:



- Admin yêu cầu quản lý thông tin khách hàng.

- Trang web truy vấn thông tin khách hàng từ cơ sở dữ liệu và hiển thị kết quả cho Admin.

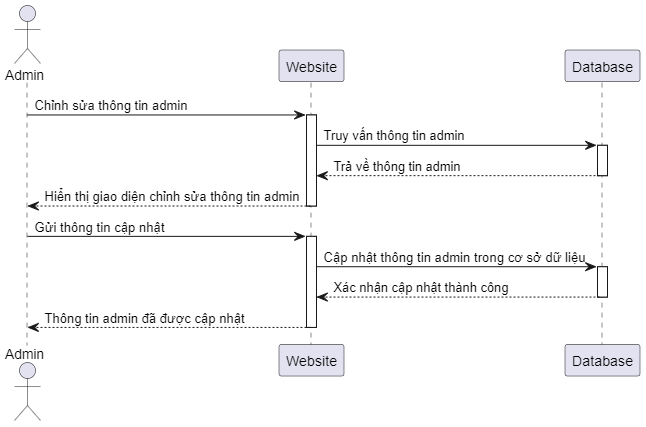
5. Quản lý đơn đặt phòng:



- Admin yêu cầu quản lý thông tin đơn đặt phòng.

- Trang web truy vấn thông tin đơn đặt phòng từ cơ sở dữ liệu và hiển thị kết quả cho Admin.

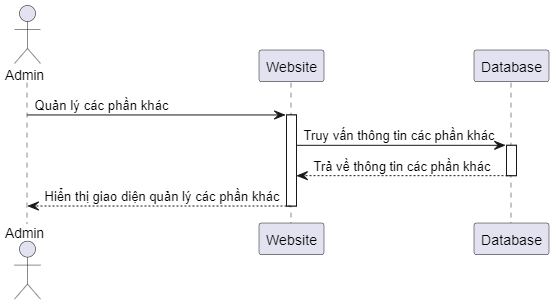
6. Chỉnh sửa thông tin admin:



- Admin yêu cầu chỉnh sửa thông tin cá nhân của mình.

- Trang web truy vấn thông tin admin từ cơ sở dữ liệu và hiển thị giao diện cho Admin để chỉnh sửa.

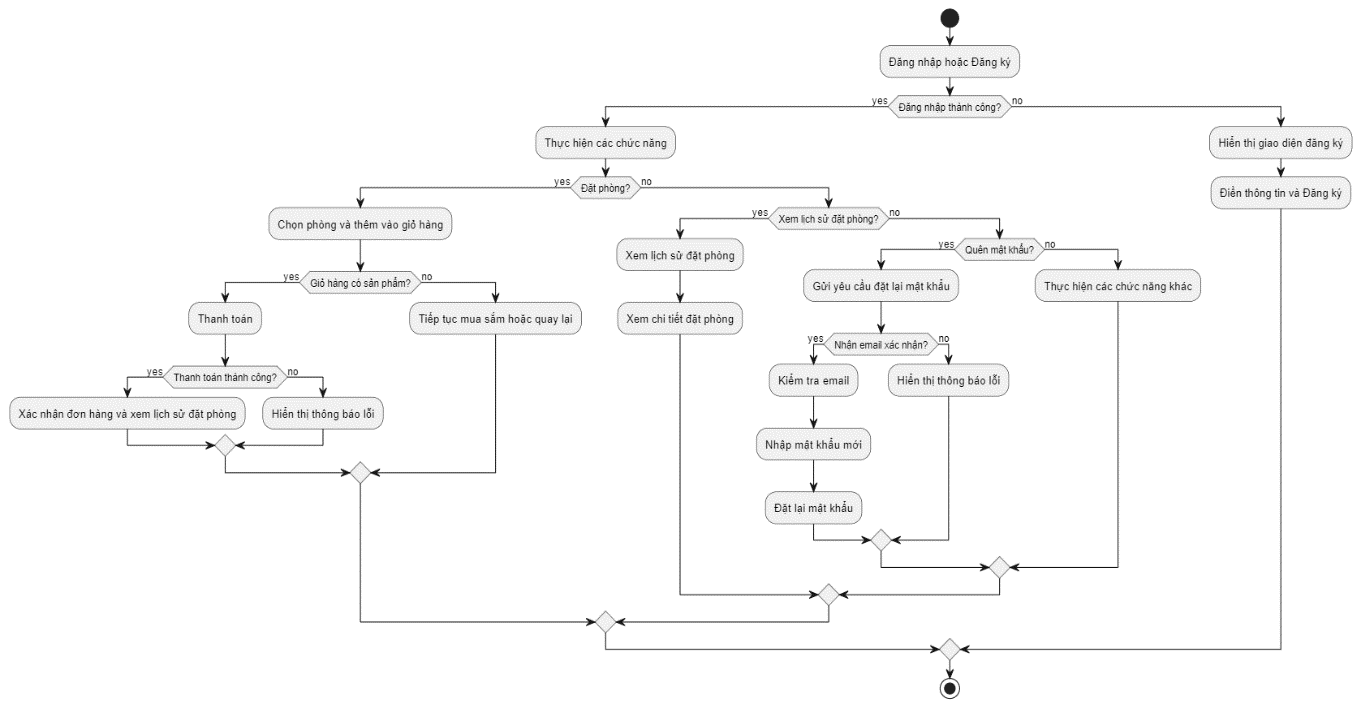
7. Quản lý các phần khác:



- Admin yêu cầu quản lý thông tin các phần khác của trang web như: video, hình ảnh, faq, blog, contact,....

- Trang web truy vấn thông tin từ cơ sở dữ liệu và hiển thị giao diện quản lý cho Admin.

1. Sơ đồ hoạt động

* Sơ đồ hoạt động của người dùng

1. Bắt đầu: Người dùng bắt đầu bằng việc đăng nhập hoặc đăng ký vào hệ thống.

2. Đăng nhập hoặc Đăng ký:

- Nếu người dùng đã có tài khoản, họ đăng nhập bằng cách nhập thông tin đăng nhập.

- Nếu không, họ chọn tùy chọn đăng ký và điền thông tin cần thiết để tạo tài khoản mới.

3. Thực hiện các chức năng:

- Sau khi đăng nhập thành công, người dùng có thể thực hiện các chức năng khác nhau như đặt phòng, xem lịch sử đặt phòng, quên mật khẩu hoặc thực hiện các chức năng khác.

4. Đặt phòng:

- Người dùng chọn phòng và thêm vào giỏ hàng.

- Nếu giỏ hàng đã đầy, họ tiến hành thanh toán.

- Nếu thanh toán thành công, hệ thống xác nhận đơn hàng và gửi email xác nhận.

5. Xem lịch sử đặt phòng:

- Nếu người dùng muốn xem lịch sử đặt phòng của mình, họ có thể truy cập vào chức năng này.

- Họ có thể xem chi tiết đặt phòng để biết thông tin cụ thể về từng đơn hàng.

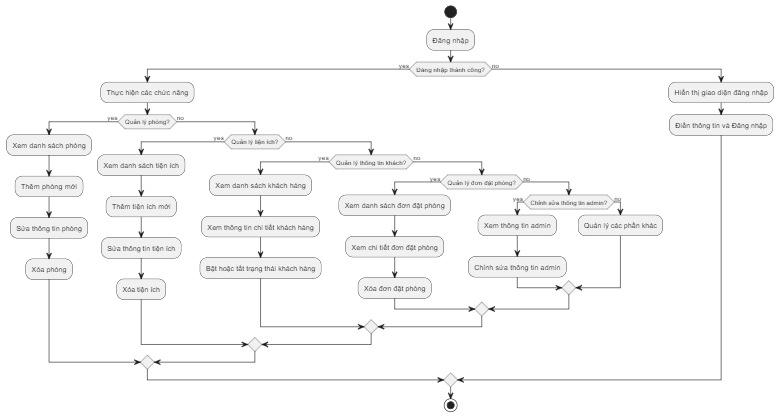
6. Quên mật khẩu:

- Nếu người dùng quên mật khẩu, họ có thể yêu cầu đặt lại mật khẩu.

- Sau khi yêu cầu, hệ thống gửi email xác nhận cho người dùng để đặt lại mật khẩu mới.

7. Kết thúc: Quy trình kết thúc sau khi người dùng hoàn thành các hoạt động mong muốn hoặc gặp lỗi và cần hỗ trợ.

* Sơ đồ hoạt động Admin



1. Bắt đầu: Quá trình bắt đầu khi admin truy cập vào hệ thống bằng cách đăng nhập hoặc nhập thông tin đăng nhập.

2. Đăng nhập hoặc Đăng ký:

- Admin nhập thông tin đăng nhập.

- Nếu thông tin hợp lệ, họ được phép truy cập vào hệ thống.

3. Thực hiện các chức năng:

- Sau khi đăng nhập thành công, admin có thể thực hiện nhiều chức năng khác nhau, bao gồm quản lý phòng, quản lý tiện ích, quản lý thông tin khách hàng, quản lý đơn đặt phòng, chỉnh sửa thông tin admin hoặc quản lý các phần khác của trang web.

4. Quản lý phòng:

- Admin có thể xem danh sách phòng hiện có.

- Họ có thể thêm phòng mới, sửa thông tin phòng hoặc xóa phòng khỏi hệ thống.

5. Quản lý tiện ích:

- Admin có thể xem danh sách các tiện ích có sẵn.

- Họ có thể thêm mới, sửa thông tin hoặc xóa tiện ích khỏi hệ thống.

6. Quản lý thông tin khách hàng:

- Admin có thể xem danh sách khách hàng đang đăng ký hoặc đã đăng ký.

- Họ có thể xem thông tin chi tiết của từng khách hàng và quản lý trạng thái của họ.

7. Quản lý đơn đặt phòng:

- Admin có thể xem danh sách các đơn đặt phòng hiện có.

- Họ có thể xem chi tiết đơn đặt phòng và quản lý chúng, bao gồm việc xóa đơn đặt phòng.

8. Chỉnh sửa thông tin admin:

- Admin có thể xem thông tin cá nhân của mình và thực hiện chỉnh sửa nếu cần thiết.

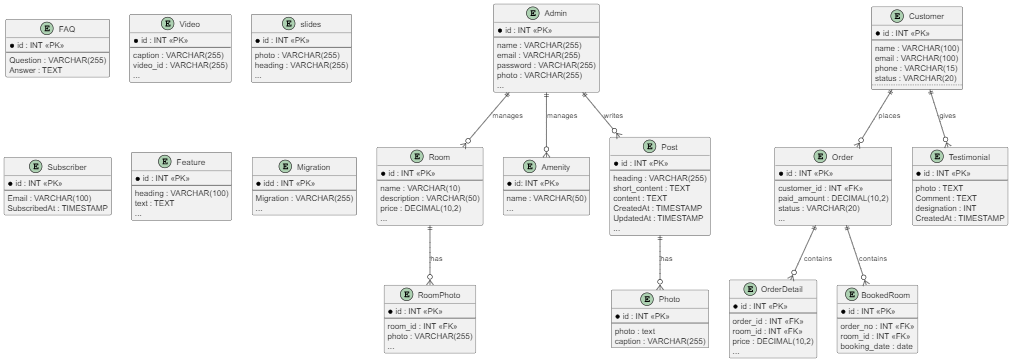
9. Quản lý các phần khác:

- Nếu cần, admin có thể quản lý các phần khác của trang web như hình ảnh, video, faq, blog, contacs.

10. Kết thúc: Quá trình kết thúc sau khi admin hoàn thành các hoạt động cần thiết hoặc sau khi họ đăng xuất khỏi hệ thống.

### 3.1.2 Thiết kế cơ sở dữ liệu

1. Sơ đồ ERD



Mô tả bảng chi tiết

**Bảng Admin**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Name | Varchar(255) | NOT NULL |
| Email | Varchar(255) | NOT NULL |
| Password | Varchar(255) | NOT NULL |
| Photo | Varchar(255) | NOT NULL |
| token | Varchar(255) | NOT NULL |
| Created\_at | timestamp | NOT NULL |
| Update\_at | timestamp | NOT NULL |

**Bảng Customers**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Name | Text | NOT NULL |
| Email | Text | NOT NULL |
| Password | Text | NOT NULL |
| Photo | Text |  |
| token | Text |  |
| Phone | Text |  |
| Country | Text |  |
| Address | Text |  |
| State | Text |  |
| City | Text |  |
| Zip | Text |  |
| Satus | Int(11) | NOT NULL |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Booked\_rooms**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Booking\_date | Text | NOT NULL |
| Order\_no | Text | NOT NULL |
| Room\_id | Text | NOT NULL |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Rooms**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Name | Text | NOT NULL |
| Description | Text | NOT NULL |
| Price | Text | NOT NULL |
| Total\_rooms | Text | NOT NULL |
| Amenities | Text | NOT NULL |
| Size | Text |  |
| Total\_beds | Text | NOT NULL |
| Total\_bathrooms | Text | NOT NULL |
| Total\_balconies | Text | NOT NULL |
| Total\_guests | Text | NOT NULL |
| Featured\_photo | Text | NOT NULL |
| Video\_id | Int(11) | NOT NULL |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Room\_photos**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Photo | Text | NOT NULL |
| room\_id | Int(11) | NOT NULL |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Orders**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Customer\_id | int(11) | NOT NULL |
| Order\_no | Text | NOT NULL |
| Transaction\_id | Text | NOT NULL |
| Payment\_method | Text | NOT NULL |
| Card\_last\_digit | Text |  |
| Paid\_amount | Text | NOT NULL |
| Booking\_date | Text | NOT NULL |
| Status | Text | NOT NULL |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Order\_details**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Order\_id | Int(11) | NOT NULL |
| Room\_id | Int(11) | NOT NULL |
| Order\_no | Text | NOT NULL |
| Checkin\_date | Text | NOT NULL |
| Checkout\_date | Text | NOT NULL |
| Adult | Text | NOT NULL |
| Children | Text | NOT NULL |
| subtotal | Text | NOT NULL |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Faqs**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Question | Text | NOT NULL |
| Answer | Text | NOT NULL |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Features**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Icon | Text | NOT NULL |
| Heading | Text | NOT NULL |
| Text | Text |  |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Photos**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Photo | Text | NOT NULL |
| Caption | Text |  |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Posts**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Photo | Text | NOT NULL |
| Heading | Text | NOT NULL |
| Short\_content | Text | NOT NULL |
| Content | Text | NOT NULL |
| Total\_view | Int(11) | NOT NULL |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Slides**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Photo | Text | NOT NULL |
| Heading | Text |  |
| Text | Text |  |
| Button\_url | Text |  |
| Button\_text | Text |  |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Subscribers**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Email | Text | NOT NULL |
| token | Text | NOT NULL |
| Satus | Int(11) | NOT NULL |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Testimonials**

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Photo | Text | NOT NULL |
| Name | Text | NOT NULL |
| Designation | Text | NOT NULL |
| Comment | Text | NOT NULL |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

**Bảng Videos**

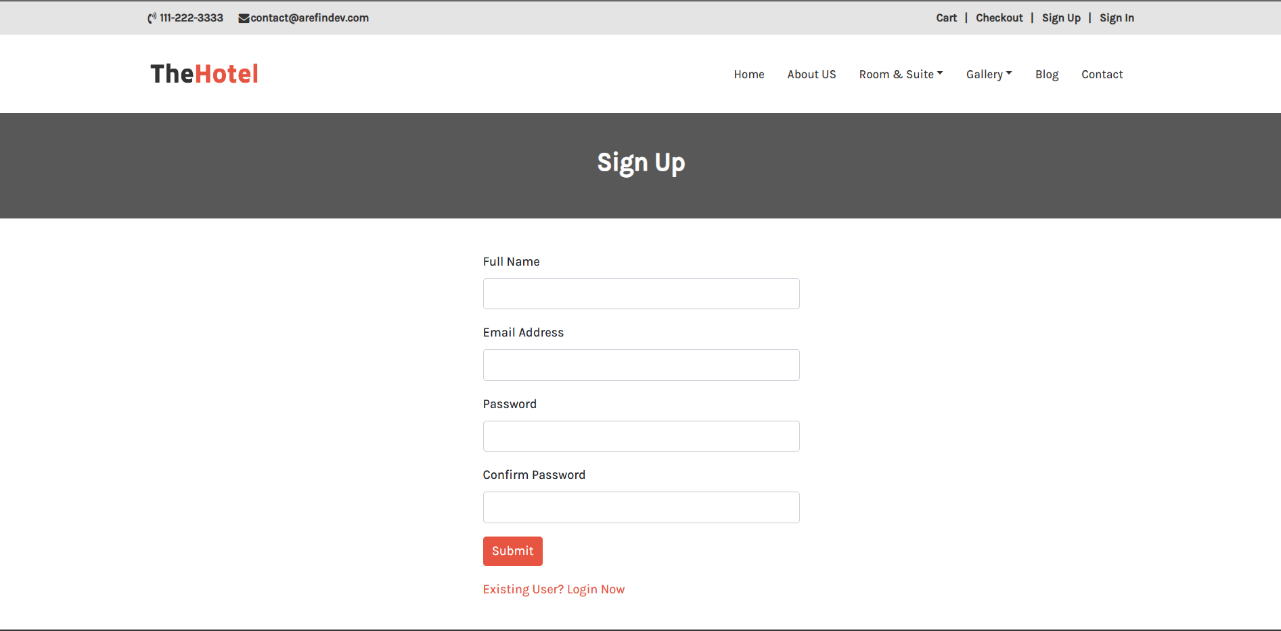
|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Ràng buộc |
| Id | Int(11) | PK, AUTO\_INCREMENT, NOT NULL |
| Video\_id | Text | NOT NULL |
| Caption | Text | NOT NULL |
| Created\_at | timestamp |  |
| Update\_at | timestamp |  |

## 3.2 Phát triển website

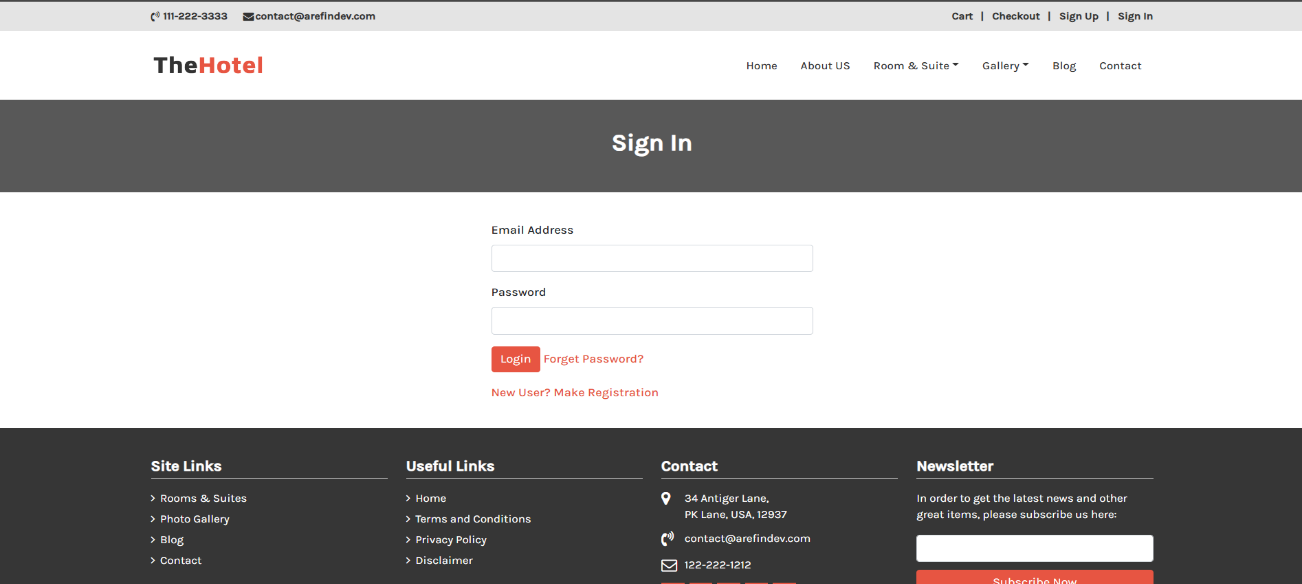
### 3.2.1 Thiết kế giao diện người dùng và các chức năng chính

1. Giao diện người dùng

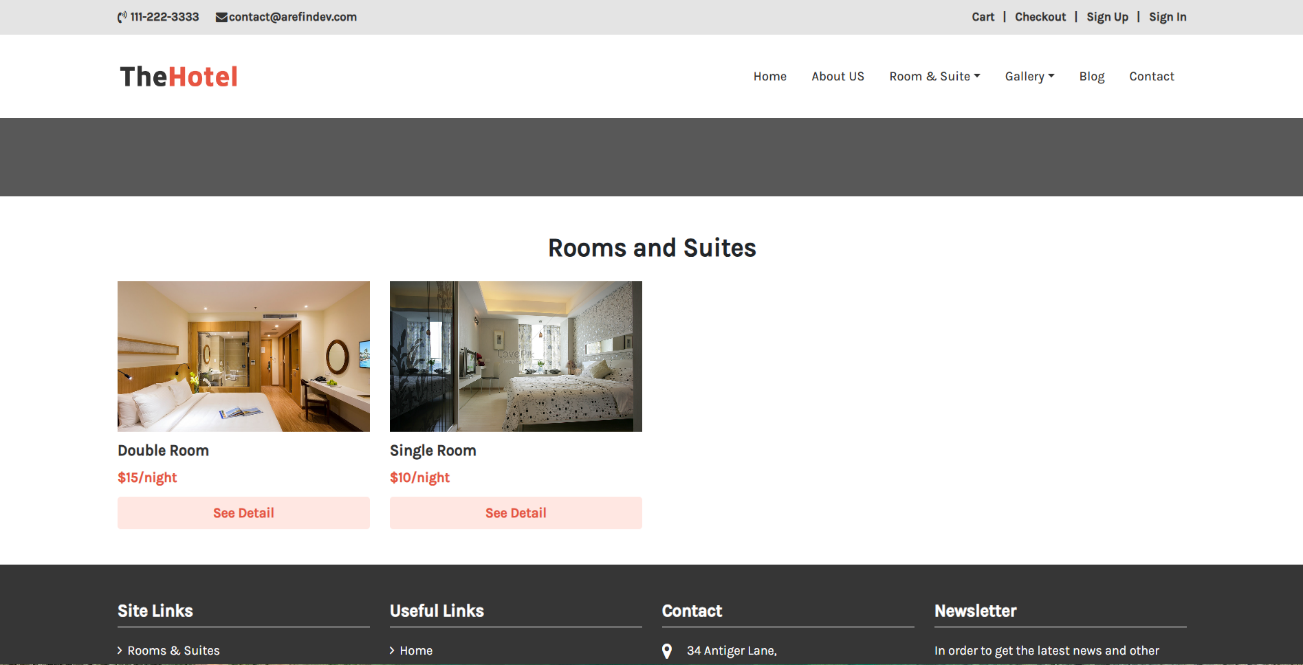
* Giao diện đăng ký



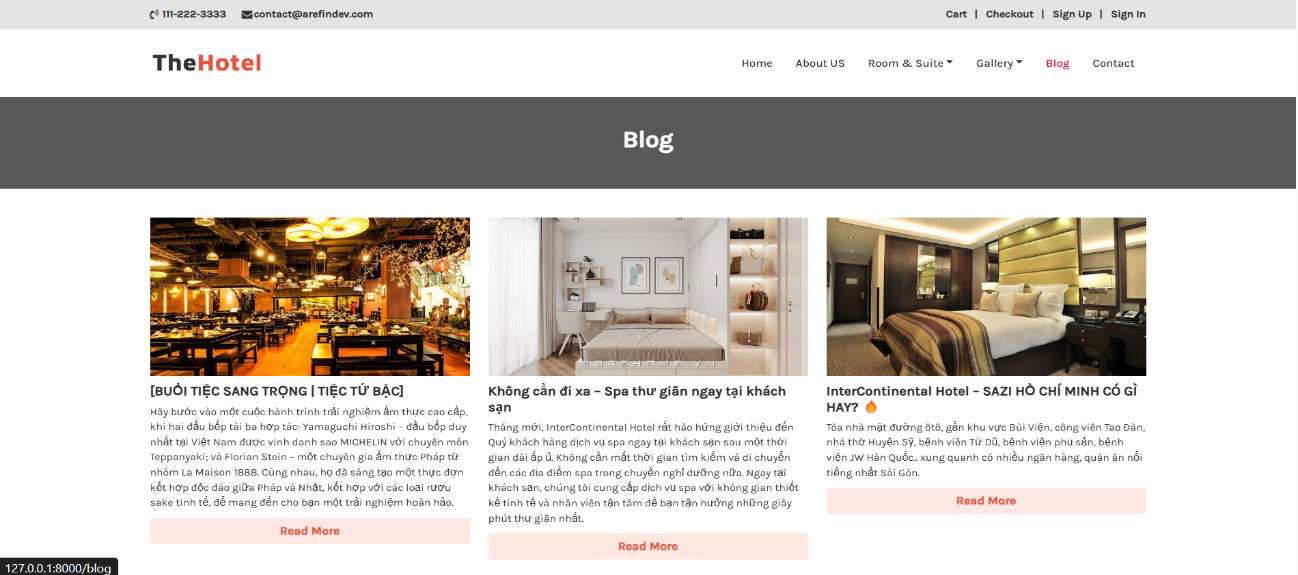
* Giao diện đăng nhập



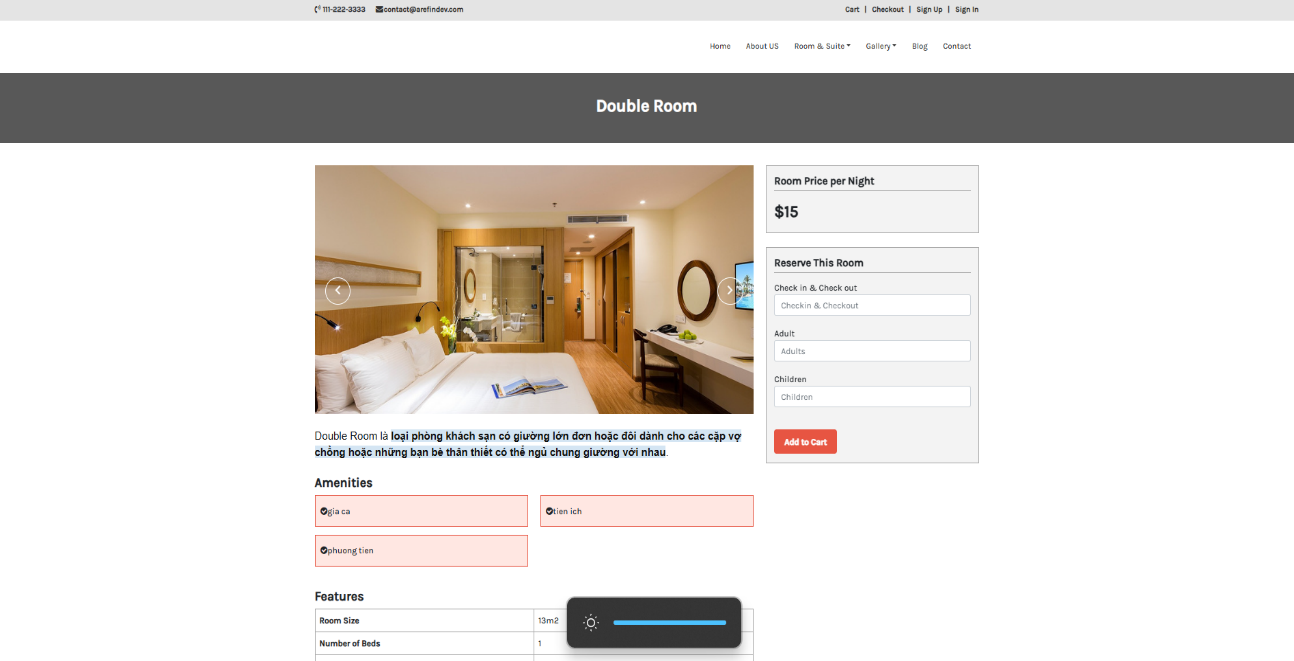
* Giao diện tìm danh sách phòng



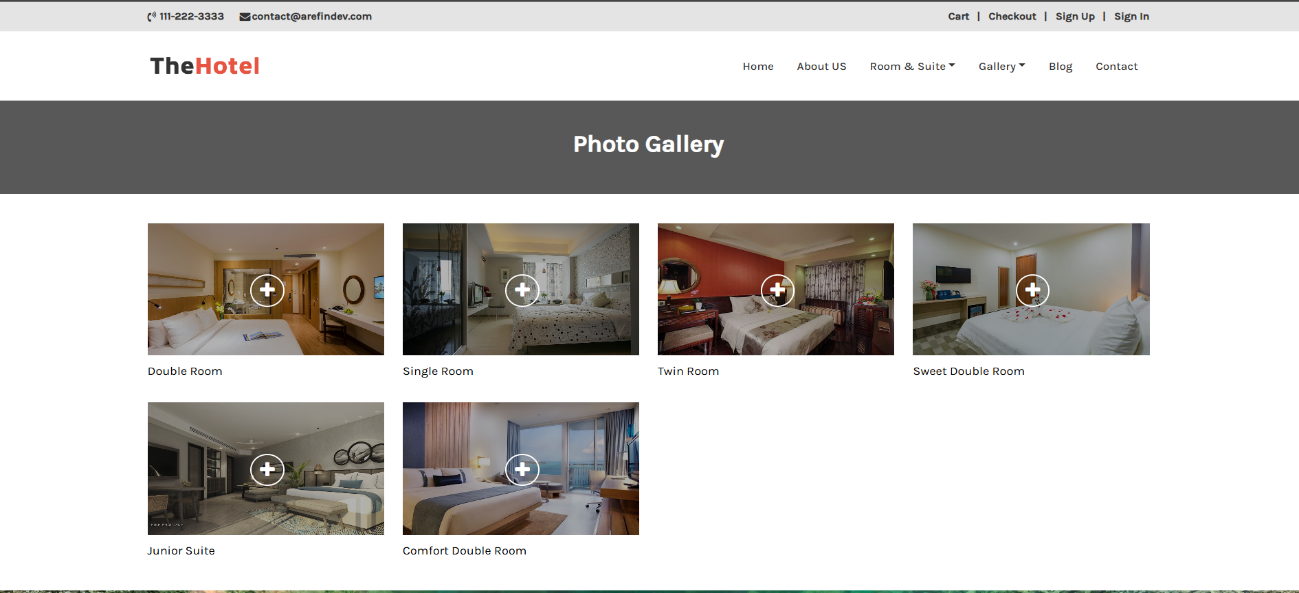
* Giao diện trang blog



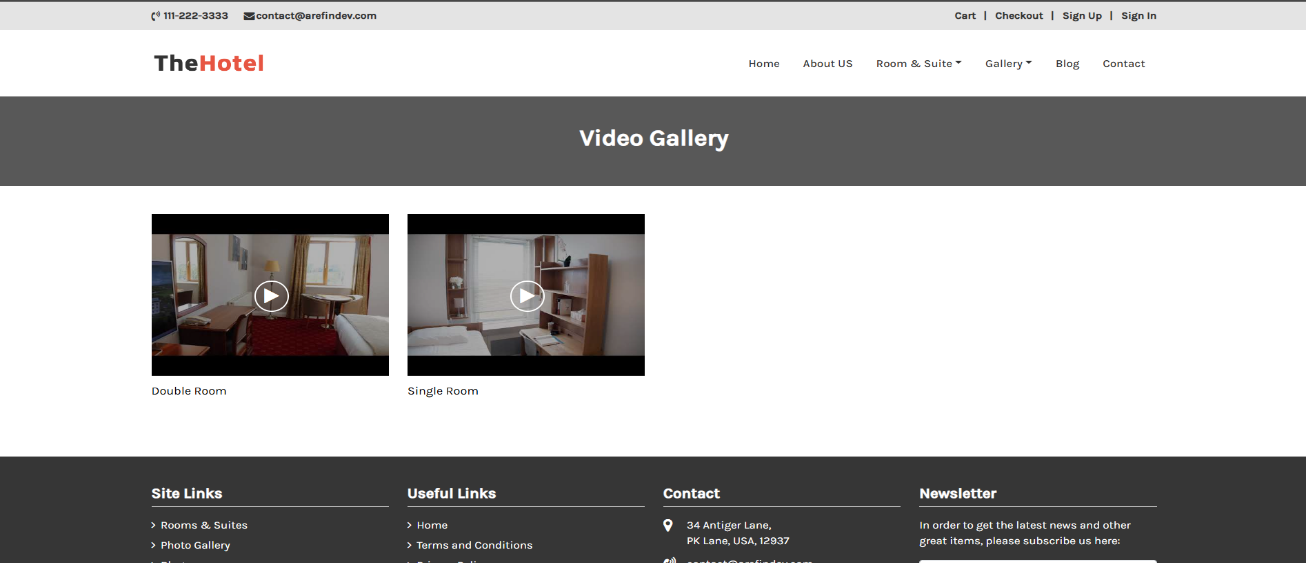
* Giao diện chi tiết phòng



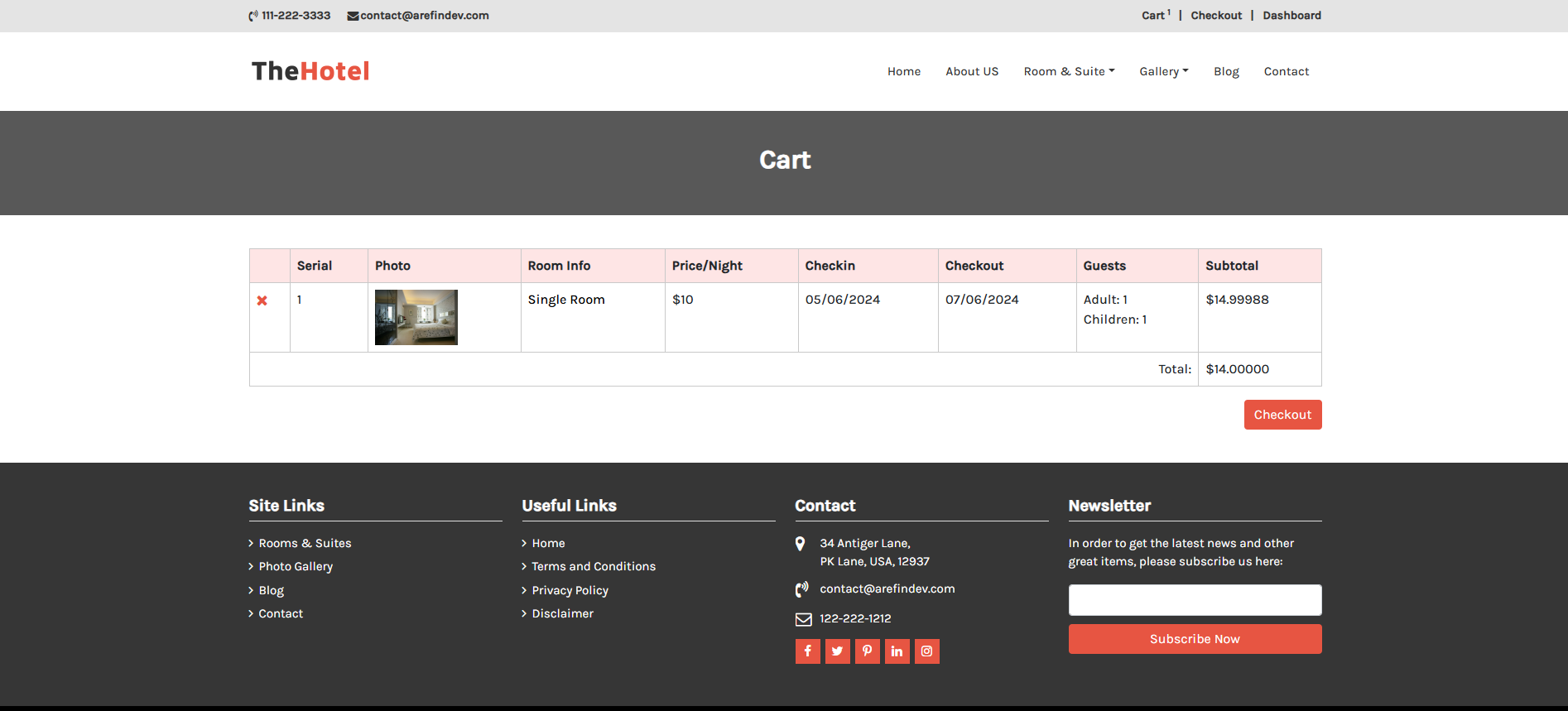
* Giao diện xem ảnh



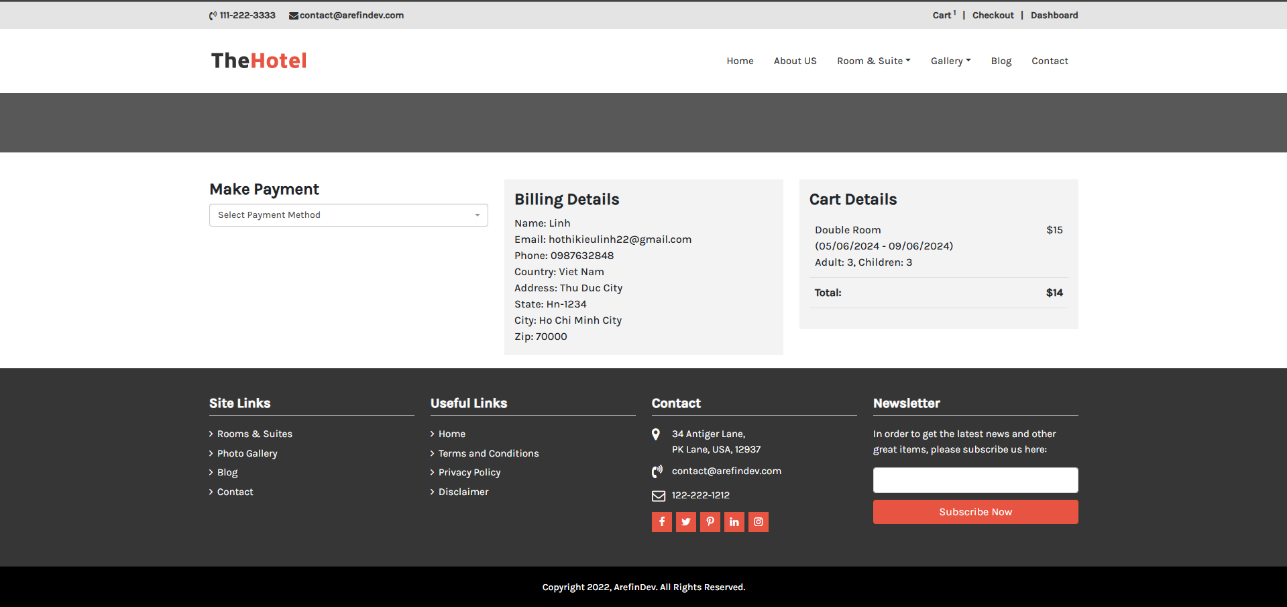
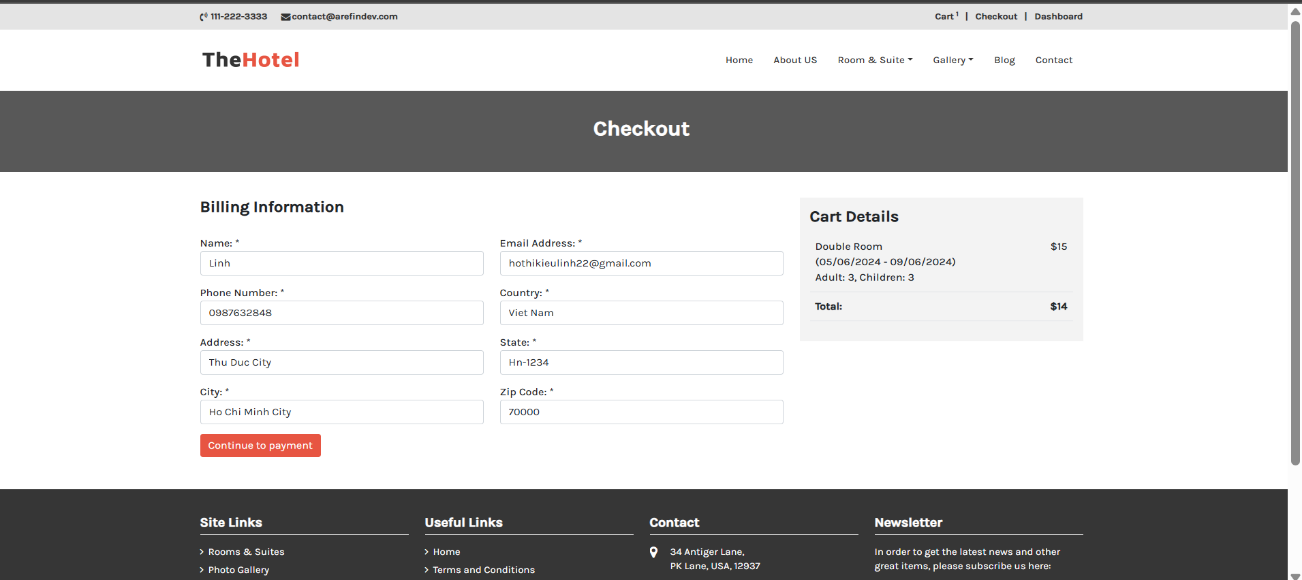
* Giao diện xem video về khách sạn



* Giao diện giỏ hàng



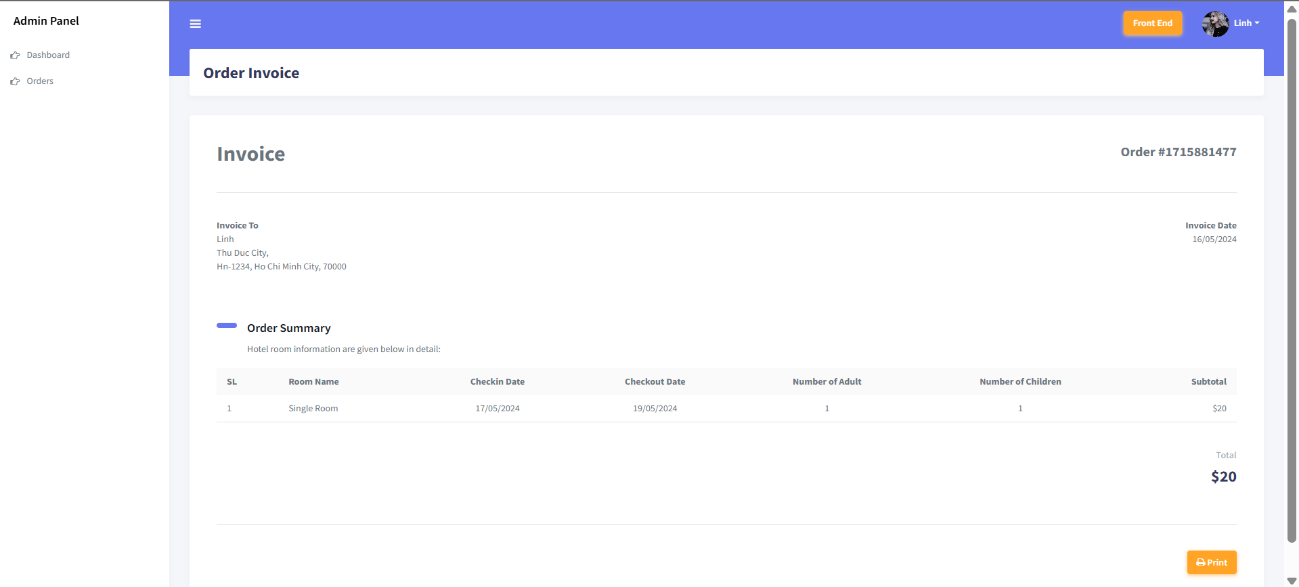
* Giao diện thanh toán



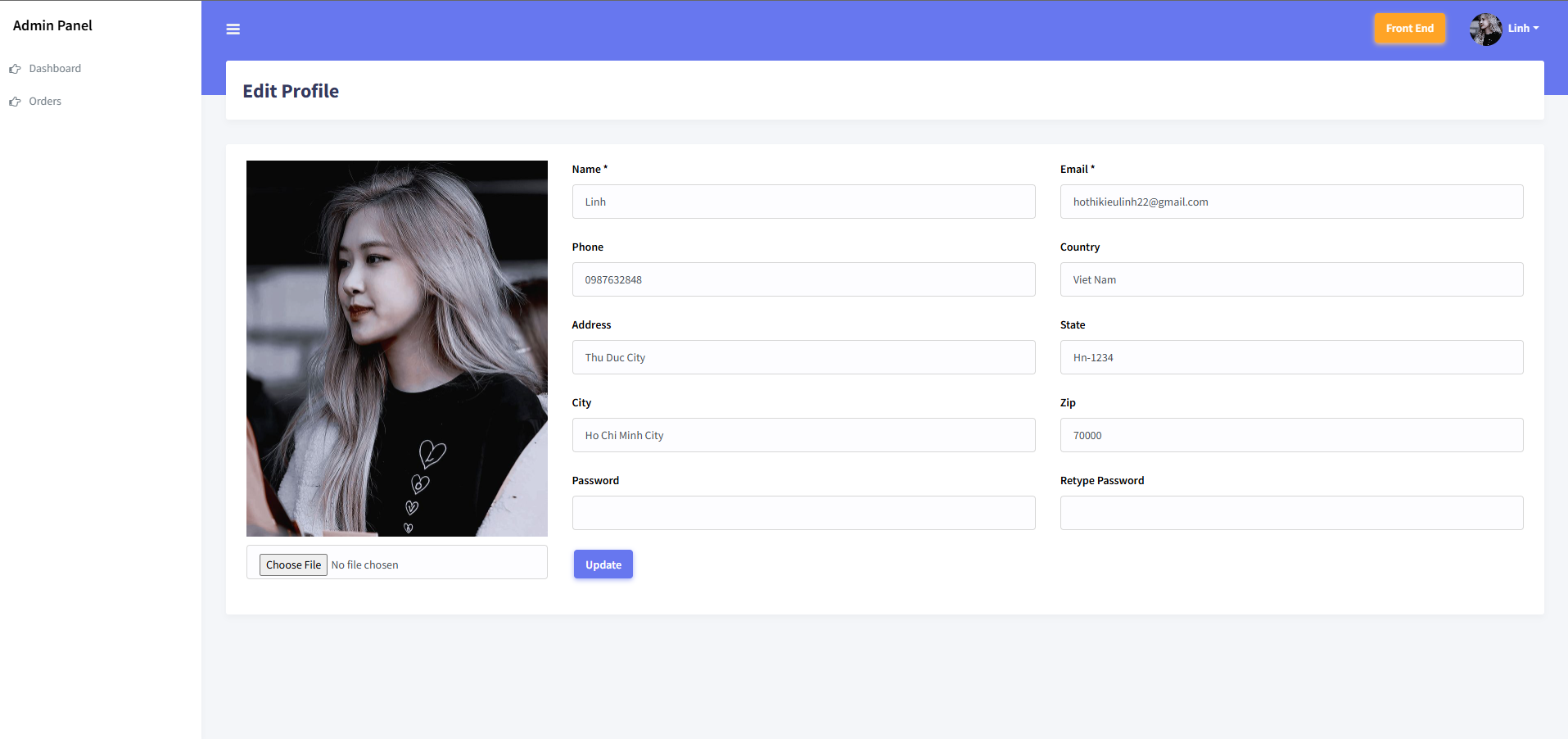
* Giao diện xem chi tiết đơn hàng



* Giao diện trang hóa đơn

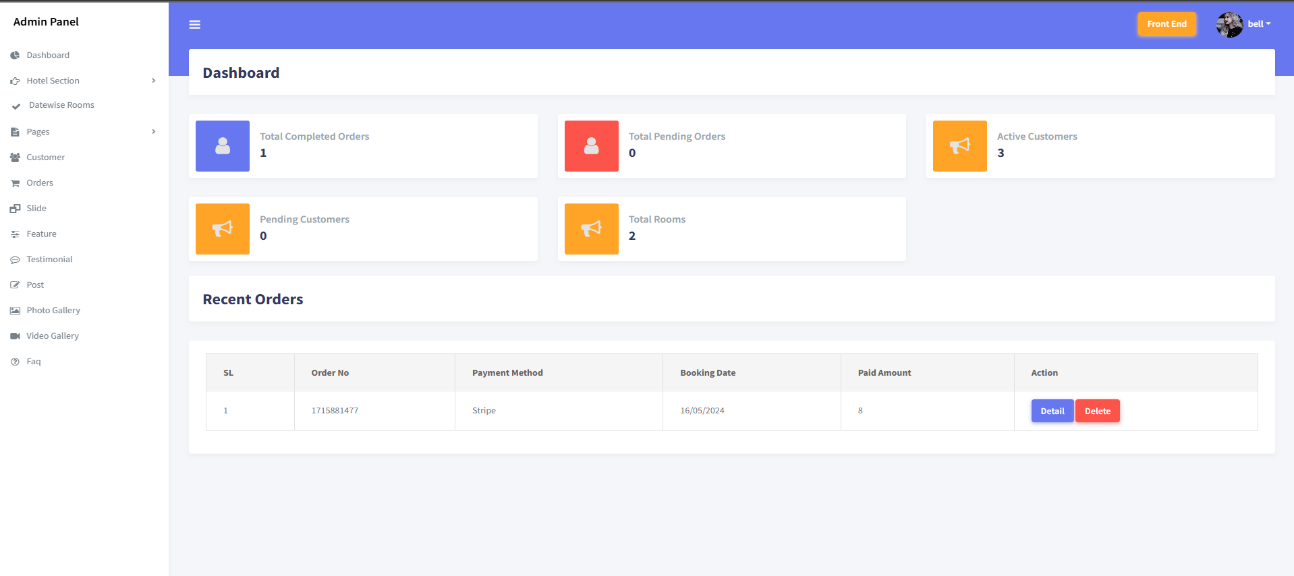


* Giao diện trang thông tin khách

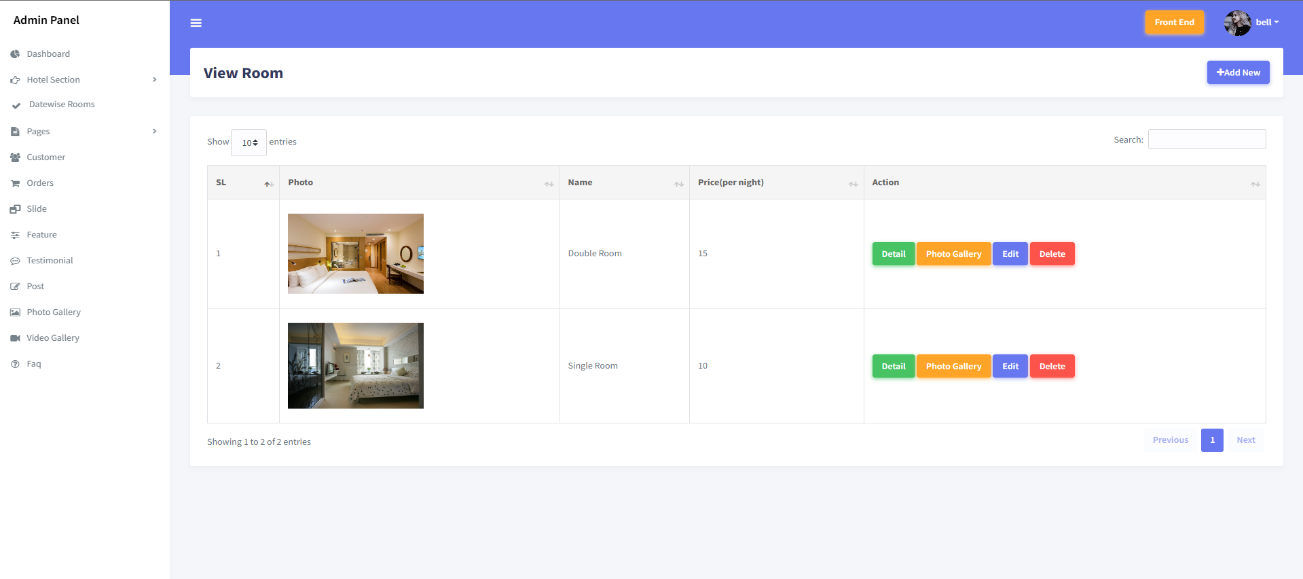


1. Giao diện admin

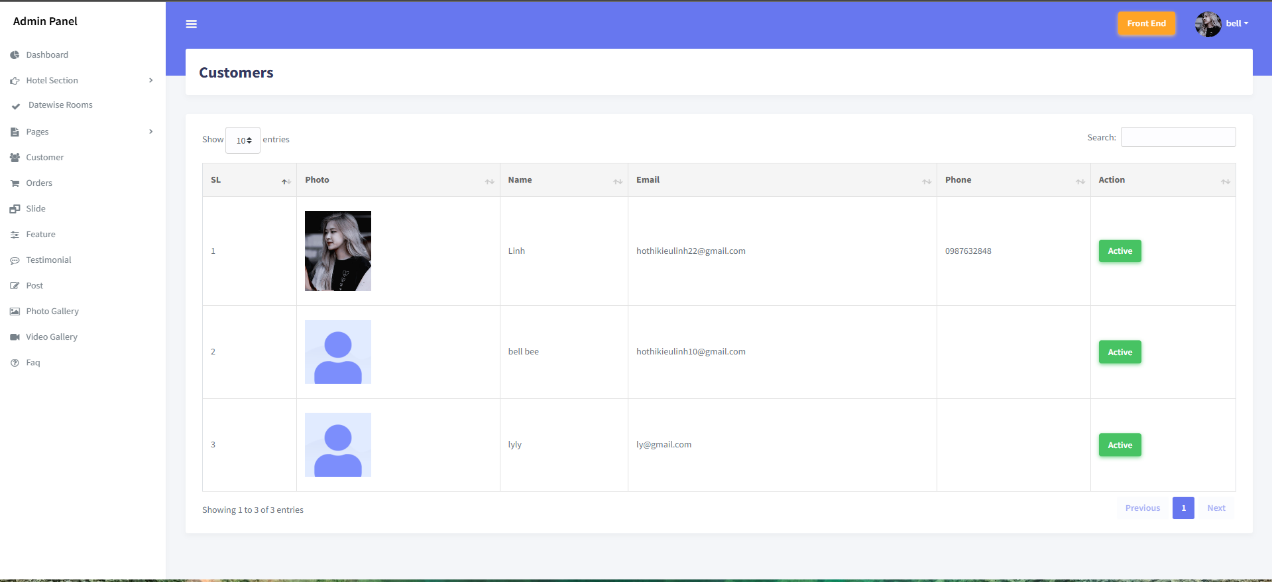
* Giao diện trang quản lý



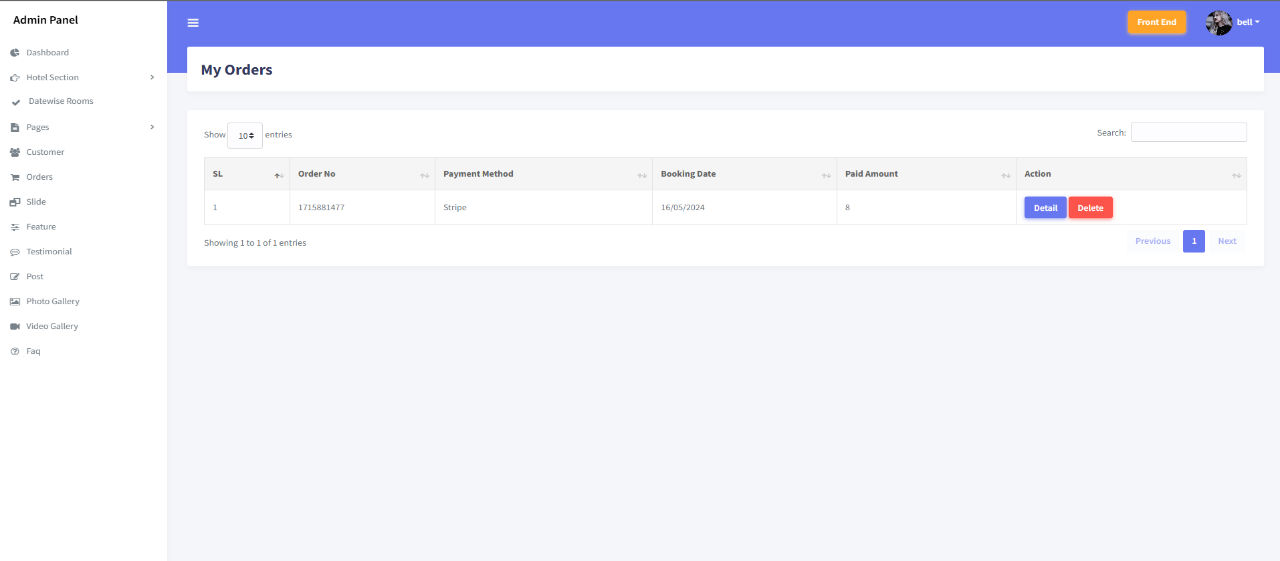
* Giao diện quản lý phòng



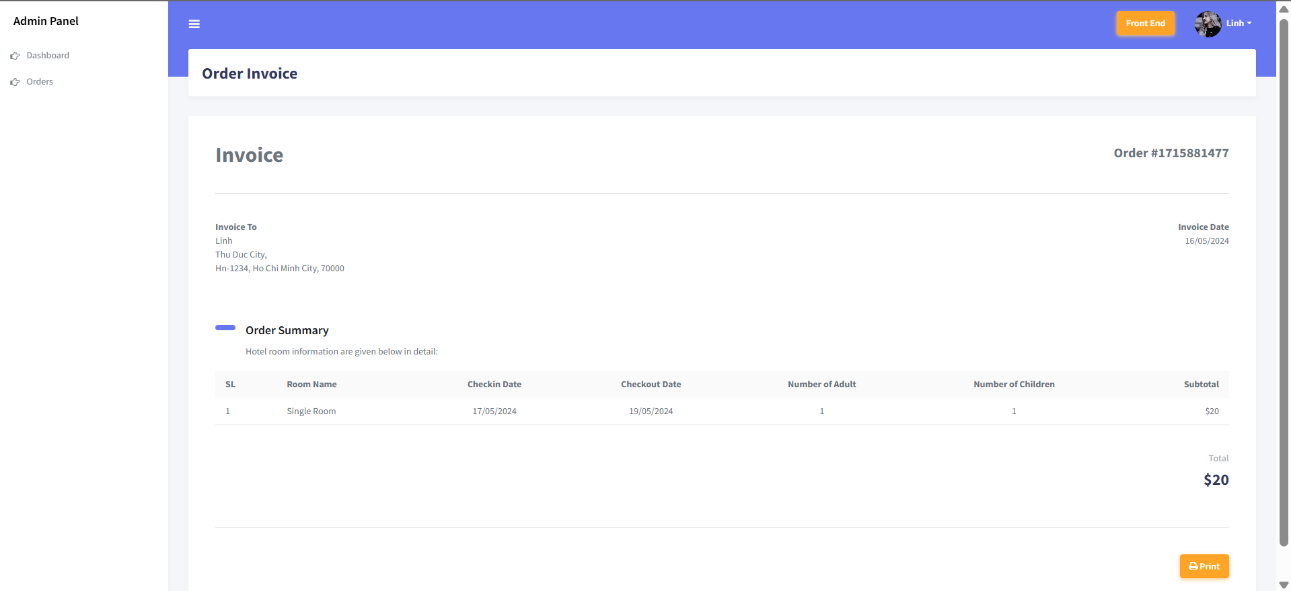
* Giao diện quản lý thông tin khách



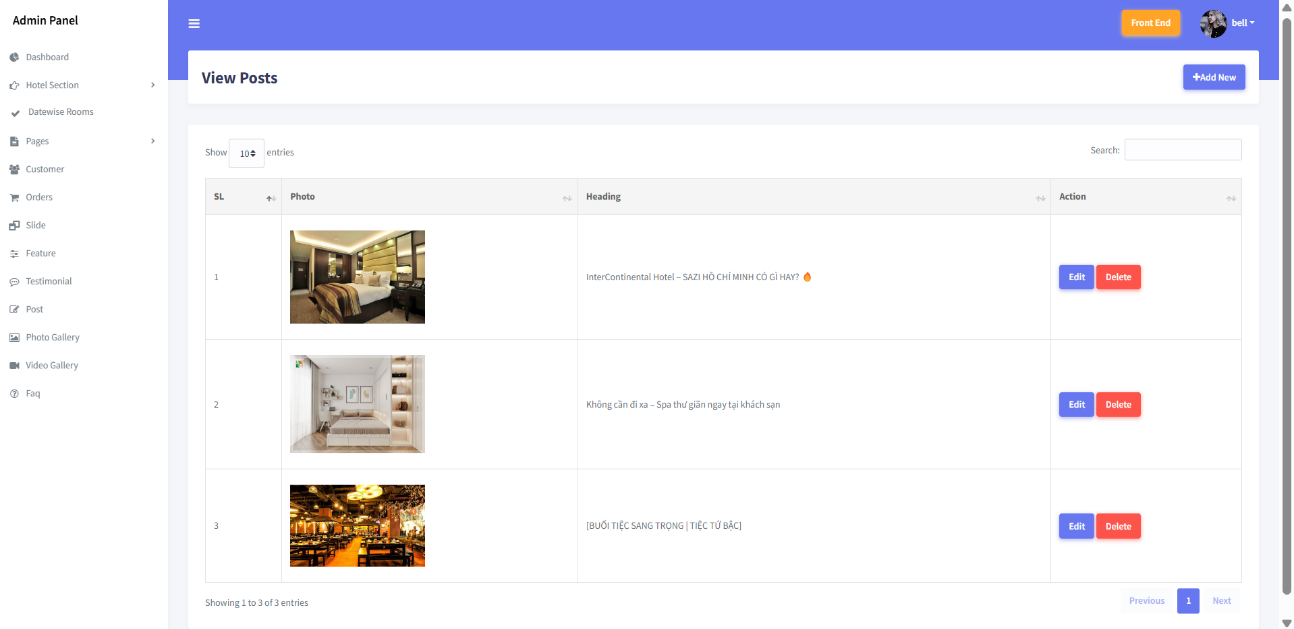
* Giao diện quản lý đơn hàng



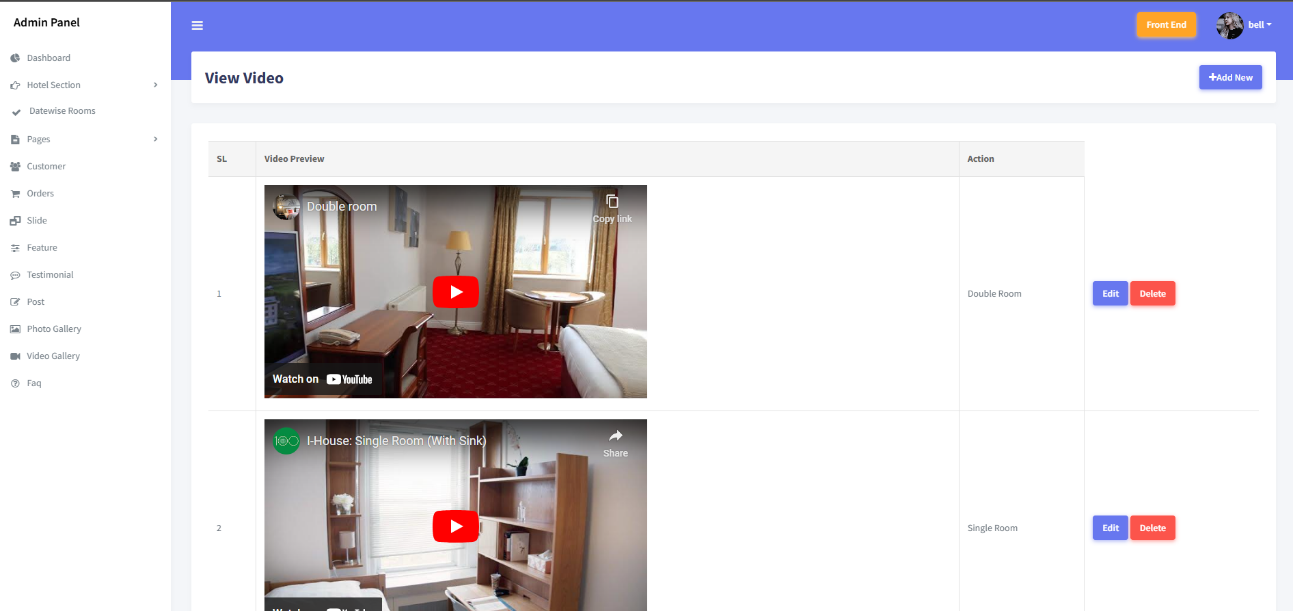
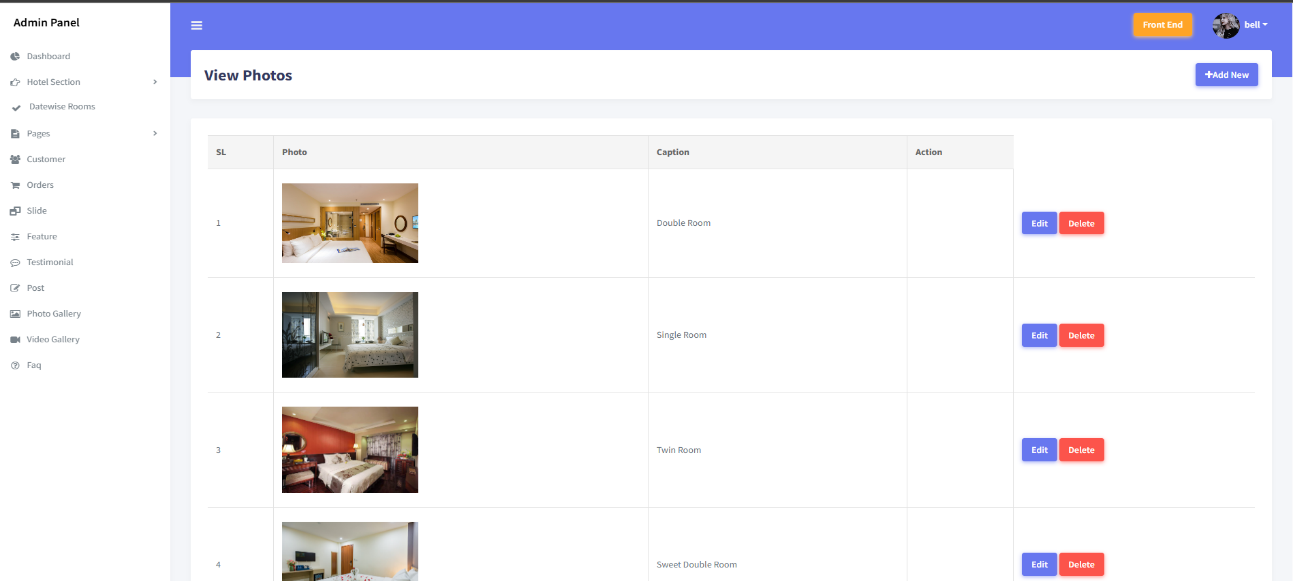
* Giao diện hóa đơn



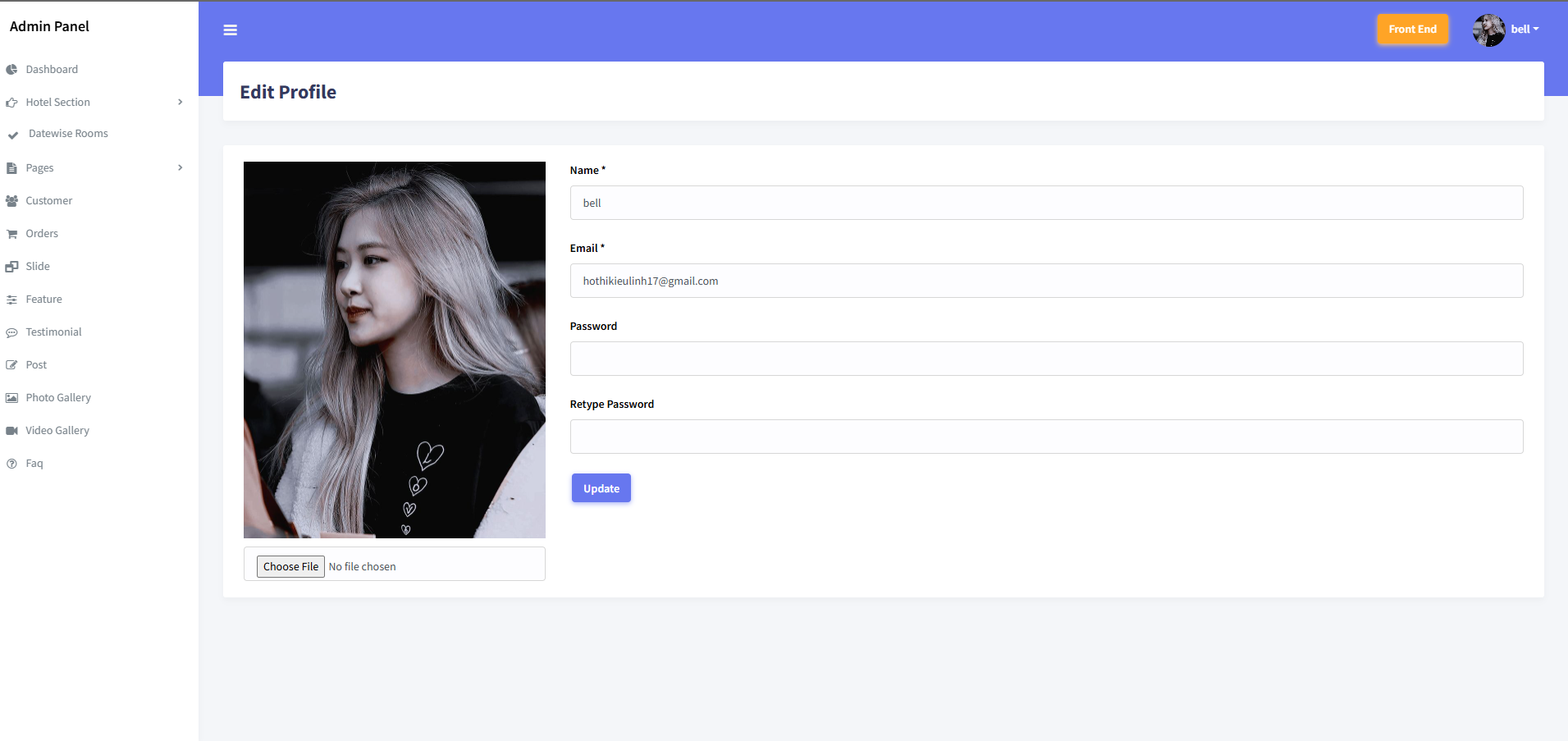
* Giao diện bài blog



* Giao diện quản lý hình ảnh , video



* Giao diện chỉnh sửa thông tin admin



# CHƯƠNG 4. NGHIÊN CỨU VÀ ỨNG DỤNG

## 4.1. Áp dụng XGBoost vào dự đoán Booking Hotel

### 4.1.1. Thu thập dữ liệu và tiền xử lý cho mô hình

- Làm sạch dữ liệu: Loại bỏ các giá trị thiếu, các bản ghi trùng lặp, và xử lý các giá trị ngoại lệ.

- Biến đổi dữ liệu: Chuyển đổi các biến số dạng chuỗi thành dạng số hoặc danh mục phù hợp cho mô hình. Ví dụ, chuyển đổi ngày tháng thành các biến số dạng số biểu thị thời gian.

- Tính toán các đặc trưng mới: Tạo ra các đặc trưng mới từ dữ liệu gốc để cải thiện hiệu suất của mô hình, chẳng hạn như tính toán số ngày lưu trú từ ngày check-in và check-out.

- Chuẩn hóa và chuẩn bị dữ liệu: Chuẩn hóa dữ liệu để đảm bảo tất cả các biến số đều có thang đo tương đương, và chia dữ liệu thành tập huấn luyện và tập kiểm tra.

### 4.1.2. Xây dựng mô hình dự đoán sử dụng XGBoost

- Chia dữ liệu: Chia dữ liệu đã tiền xử lý thành các tập huấn luyện, tập kiểm tra, và tập xác thực.

- Huấn luyện mô hình: Sử dụng tập huấn luyện để huấn luyện mô hình XGBoost, tinh chỉnh các siêu tham số của mô hình để đạt được hiệu suất tốt nhất.

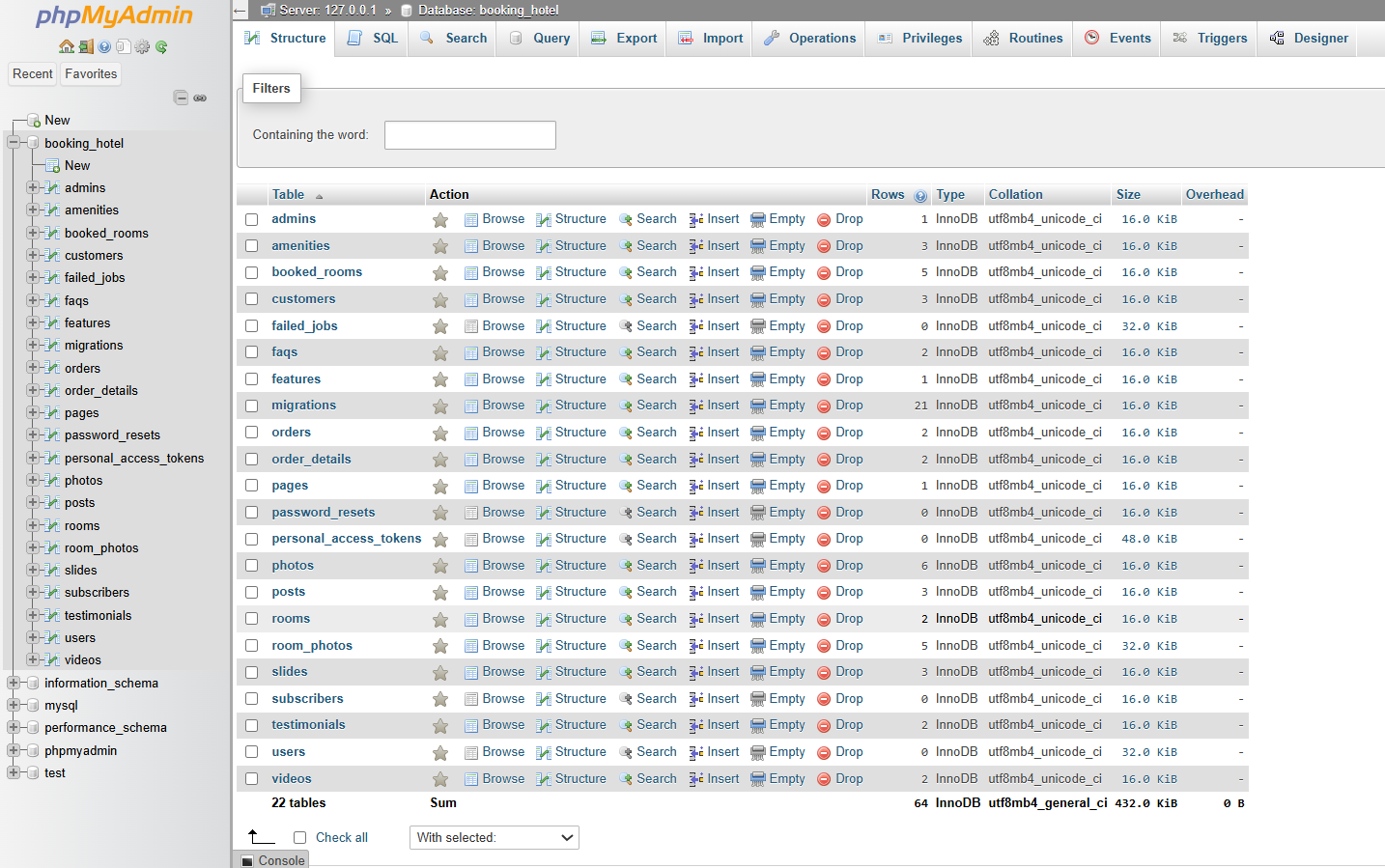
## 4.2. Tích hợp mô hình vào website

### 4.2.1. Triển khai mô hình dự đoán trên website Booking Hotel

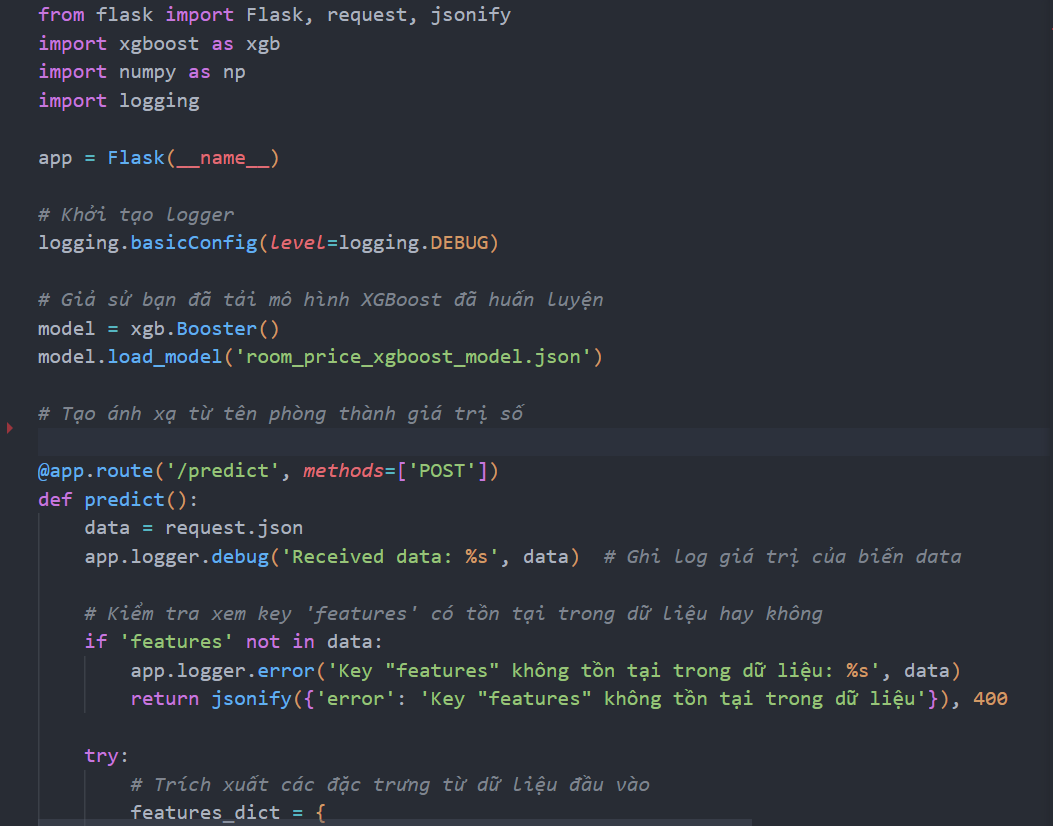
- Chuẩn bị môi trường: Thiết lập môi trường triển khai bao gồm máy chủ, cơ sở dữ liệu, và các dịch vụ cần thiết để hỗ trợ mô hình.

- Máy chủ: Cài đặt và cấu hình máy chủ để chạy mô hình dự đoán. Máy chủ cần đủ mạnh để xử lý các yêu cầu từ website một cách hiệu quả.

- Cơ sở dữ liệu: Thiết lập cơ sở dữ liệu để lưu trữ dữ liệu liên quan đến mô hình và các dự đoán. Đảm bảo cơ sở dữ liệu có khả năng mở rộng và bảo mật tốt.



- Dịch vụ hỗ trợ: Cài đặt các dịch vụ cần thiết như web server (Apache), framework (Flask) để triển khai API của mô hình. Các dịch vụ này sẽ chịu trách nhiệm xử lý các yêu cầu từ website và giao tiếp với mô hình dự đoán.

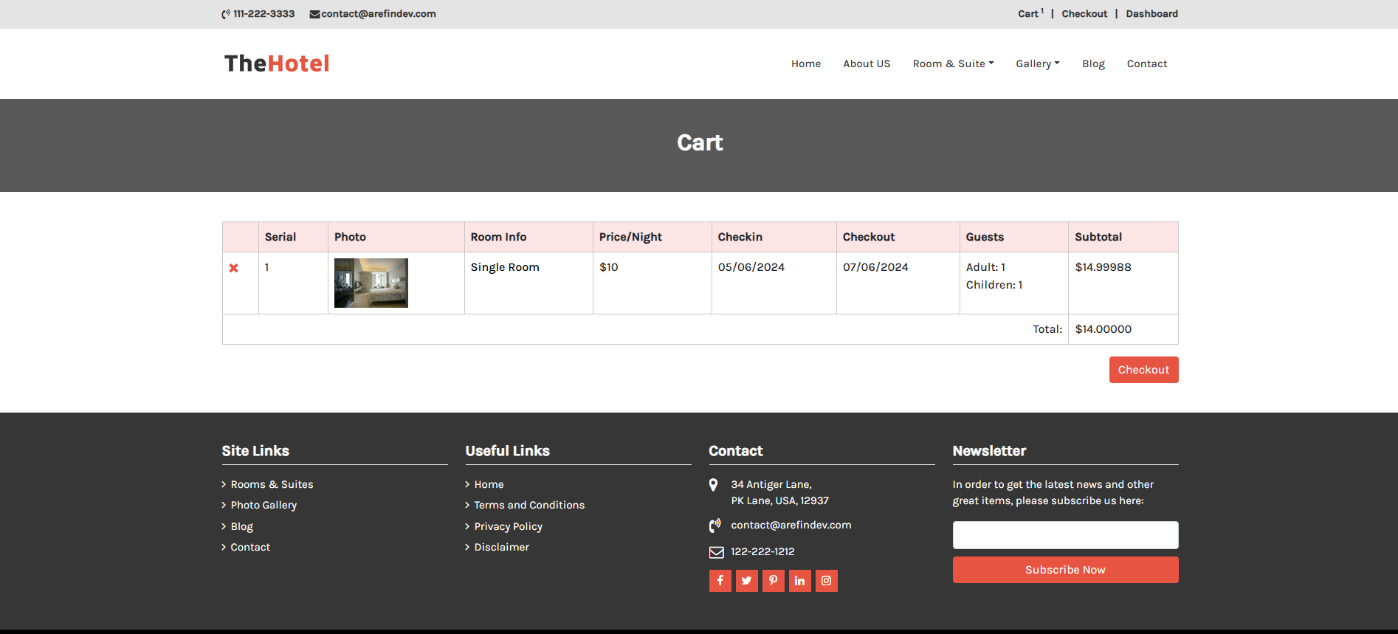


- Triển khai mô hình: Triển khai mô hình XGBoost dưới dạng một API có thể gọi và sử dụng.

- API triển khai: Xây dựng API cho mô hình dự đoán bằng các framework như Flask. API sẽ nhận dữ liệu từ website, xử lý thông qua mô hình XGBoost và trả về kết quả dự đoán.

- Kết nối với website PHP: Tích hợp API của mô hình vào hệ thống website PHP, đảm bảo rằng các yêu cầu dự đoán từ người dùng được gửi đến mô hình và kết quả được trả về và hiển thị một cách chính xác.

- Giao diện người dùng: Cập nhật giao diện người dùng của website để tương tác với API của mô hình. Đảm bảo rằng người dùng có thể dễ dàng nhập dữ liệu và nhận kết quả dự đoán.



- Xử lý kết quả: Hiển thị kết quả dự đoán trên website một cách thân thiện và dễ hiểu cho người dùng. Đảm bảo các kết quả được trình bày rõ ràng và có thể hành động.

# CHƯƠNG 5. KẾT QUẢ VÀ KIẾN NGHỊ

## 5.1 Kết quả đạt được

### 5.1.1. Tóm tắt kết quả đạt được

- Phát triển Web Booking Phòng Khách Sạn: Hệ thống web booking đã được xây dựng và triển khai thành công, bao gồm các chức năng như đăng ký, đăng nhập, tìm kiếm phòng, đặt phòng, quản lý đặt phòng, và thanh toán trực tuyến.

- Giao Diện Người Dùng Thân Thiện: Giao diện người dùng được thiết kế trực quan, dễ sử dụng với khả năng tương thích trên nhiều thiết bị khác nhau (desktop, tablet, mobile).

- Tích Hợp XGBoost: Thuật toán XGBoost đã được tích hợp thành công vào hệ thống để dự đoán giá phòng khách sạn dựa trên các yếu tố như thời gian đặt phòng, loại phòng, số lượng người lớn, số lượng trẻ em.

### 5.1.2. Đánh giá ưu điểm và hạn chế

1. Ưu điểm:

- Tính Năng Đa Dạng: Hệ thống cung cấp nhiều tính năng hữu ích, từ đặt phòng đến dự đoán giá phòng.

- Thuật Toán Dự Đoán Hiệu Quả: XGBoost cho kết quả dự đoán chính xác cao, giúp người dùng có thể lên kế hoạch chi phí tốt hơn.

1. Hạn chế:

- Yêu Cầu Về Dữ Liệu: Để dự đoán chính xác, hệ thống cần một lượng lớn dữ liệu lịch sử, điều này có thể gây khó khăn trong giai đoạn đầu.

- Bảo Mật: Cần cải thiện thêm về các biện pháp bảo mật để bảo vệ thông tin người dùng và giao dịch trực tuyến.

## 5.2 Kiến nghị

### 5.2.1. Mở rộng tính năng và khả năng của hệ thống

- Tích Hợp Đánh Giá và Bình Luận: Cho phép người dùng đánh giá và bình luận về các khách sạn và phòng đã đặt, giúp cải thiện quyết định của những người dùng sau.

- Chương Trình Khách Hàng Thân Thiết: Xây dựng các chương trình khách hàng thân thiết để khuyến khích người dùng quay lại sử dụng dịch vụ.

- Tích Hợp Nhiều Phương Thức Thanh Toán: Mở rộng các phương thức thanh toán để người dùng có nhiều lựa chọn hơn.

- Phân Tích Dữ Liệu Người Dùng: Sử dụng dữ liệu người dùng để cá nhân hóa trải nghiệm, đề xuất phòng hoặc khuyến mãi phù hợp.

### 5.2.2. Nghiên cứu và tích hợp các công nghệ mới

- Trí Tuệ Nhân Tạo và Học Máy: Tiếp tục nghiên cứu và ứng dụng các công nghệ trí tuệ nhân tạo và học máy để cải thiện tính năng dự đoán và tối ưu hóa giá phòng.

- Blockchain: Xem xét tích hợp công nghệ blockchain để tăng cường bảo mật cho các giao dịch trực tuyến.

- Phân Tích Dữ Liệu Lớn: Áp dụng các phương pháp phân tích dữ liệu lớn để hiểu rõ hơn về hành vi người dùng và xu hướng thị trường, từ đó tối ưu hóa dịch vụ và chiến lược kinh doanh.

# TÀI LIỆU THAM KHẢO

[1]. [Cloud Firestore Flutter] [Online] Aviliable <https://viblo.asia/p/flutter-su-dung-cloud-firestore-tren-flutter-Az45bL8gZxY> Access[1/11/2023].

[2]. [Ngôn ngữ dart] [Online] Aviliable <https://baoflutter.com/category/kien-thuc-ve-ngon-ngu-dart-dung-trong-flutter/> Access[11/1/2023].

[3]. [Flutter] [Online] Avaliable <https://blog.flutter.wtf/flutter-web-the-web-development-trend-you-cant-ignore/> Access[11/1/2023].

[4]. [Firebase] [Online] Avalibale <https://magz.techover.io/2022/04/09/firebase-dung-trong-flutter-phan-2/> by Access[11/1/2023].

[5]. [Flutter] [Online] Avaliable <https://baoflutter.com/flutter-web-nhung-kien-thuc-can-co-trong-lap-trinh/> Access[11/1/2023].

[6]. [Firebase] [Online] Avalibale <https://www.youtube.com/watch?v=aKgEEnVhU1I> Access[11/1/2023].