**Lawrence Technological University**

**Math and Computer Science Department**

**5274\_MCS7013\_Collaborative Research Proj 1**

**Student Name: Vinod Bokkisam**

**Date: 16th July 2024**

**Online Flight Booking System**

**Github Link:** [**https://github.com/vinod410/Flight-Booking**](https://github.com/vinod410/Flight-Booking)

**Introduction**

In today's fast-paced world, travel has become an integral part of our lives. Whether for business, leisure, or other purposes, the demand for air travel has surged significantly over the years. With the advent of modern technology, the airline industry has witnessed a transformation, moving towards online booking systems to simplify the travel experience for passengers. This paradigm shift has given rise to the development of the "Online Flight Booking System Project in Python and MySQL."

**Purpose**

The primary purpose of this project is to streamline the process of booking airline tickets and managing flight schedules. It is designed to offer a user-friendly, efficient, and secure platform for both passengers and administrators to interact with flight information, make bookings, and manage flight operations. The project's core objective is to provide an intuitive and convenient way for users to search for available flights, book tickets, make payments, and access essential flight information, including departure times, gate numbers, and seat details.

**Scope**

The scope of the Online Flight Booking System is extensive, covering various facets of the airline industry, including passenger booking and flight management. Here are the key aspects of the project's scope:

* Client-Side Interaction: The system facilitates passenger interaction, allowing them to search for available flights, book tickets, make payments, and access flight details.
* Admin Panel: Administrators have full control over the system, enabling them to manage flights, cities, bookings, and flight schedules.
* Passenger Registration: Passengers can register on the platform to access features like booking tickets and viewing flight information.
* Booking Airline Tickets: Passengers can book tickets for their desired flights after searching for available options.
* Simple Payment System: The system integrates a secure payment gateway for seamless transaction processing.
* Search for Flights: Passengers can search for flights based on their departure and arrival locations, dates, class, and the number of passengers.
* View E-Ticket: After booking, passengers can access and view their electronic tickets, which contain detailed flight information.
* Cancel Tickets: Passengers have the option to cancel their bookings, and the system automates the ticket generation and cancellation processes.
* Print Tickets: Users can print their tickets for reference or check-in at the airport.
* Check Flight Status: Passengers can check the real-time status of their booked flights to stay informed about any delays or changes.
* View Total Amount: The system provides a summary of the total amount paid for bookings.
* List Today’s Flights: Administrators can monitor and manage flights scheduled for the current day.
* Manage Flight’s Departure – Arrival: Admins can mark flights as departed or arrived based on their status.
* Mark Flight Issues: In case of any flight-related issues, administrators can document and track them within the system.
* Flight Management: The admin has the ability to manage flight schedules, airlines, and flight details, including departure and arrival times, locations, duration, prices, and airline information.

**Importance**

The development of this Online Flight Booking System is of significant importance for several reasons:

* Enhanced Passenger Experience: The system offers a streamlined and user-friendly way for passengers to book tickets, reducing the hassles of traditional booking processes.
* Efficient Flight Management: The project provides airlines and administrators with a comprehensive tool for managing flight schedules, bookings, and passenger information.
* Data Centralization: By centralizing flight-related data, the system enables airlines to maintain accurate records and streamline operations.
* Increased Efficiency: Automation of ticket generation, payment processing, and flight status tracking enhances operational efficiency and reduces human errors.
* Cost Reduction: The adoption of an online booking system can reduce administrative and operational costs, making it a financially beneficial solution for airlines.
* Competitive Advantage: In a competitive market, airlines that offer an efficient and user-friendly online booking system gain an edge by attracting more passengers.
* Learning Resource: Beyond its practical use, this project serves as a valuable learning resource for individuals seeking to understand and implement similar systems in the field of web development and database management.

**Technologies**

***Backend Technologies***

*Python:*

Django: Django can be used for a more robust and scalable backend, providing built-in features for authentication, database operations, and an admin interface.

APIs: RESTful APIs can be developed using Django REST Framework (DRF) to facilitate communication between the client-side and server-side components.

MySQL: A relational database management system used to store and manage data related to flights, bookings, passengers, and other essential information.

SQLAlchemy: An ORM (Object-Relational Mapping) library for Python that facilitates database operations and interactions in a more Pythonic way.

***Frontend Technologies***

HTML5: The standard markup language for creating web pages, providing the structure and content of the airline reservation system's web interface.

CSS3: Used for styling and designing the web pages to ensure a visually appealing and responsive user interface.

Bootstrap: A front-end framework that helps in creating responsive and mobile-first web pages quickly, with pre-designed components and utilities.

**System Overview**

The Online Flight Booking System is a sophisticated web-based application that offers a comprehensive solution for booking airline tickets and managing flight schedules. In this section, we will delve into the system's architecture, its core components, the interaction between the client-side and admin panel, and an overview of their respective features.

*System Architecture and Components:*

The system is built using Python and MySQL, employing a three-tier architecture, which comprises:

Presentation Layer: This layer represents the user interface, where passengers and administrators interact with the system. It includes the web pages, forms, and UI elements for making bookings, viewing flight information, and managing the system. For the presentation layer, the project utilizes a combination of HTML, CSS, JavaScript, and the Bootstrap framework to create a visually appealing and responsive design.

Application Layer: The application layer serves as the core of the system, where server-side logic is implemented. Python, a server-side scripting language, is used to handle user requests, process data, and connect with the database. This layer is responsible for managing user registration, flight booking, payment processing, and other application-specific functionalities.

Data Layer: The data layer involves a MySQL database that stores and manages all the essential data, such as flight schedules, passenger information, booking records, flight details, cities, and airlines. The database schema is designed to maintain the integrity of data and ensure efficient data retrieval and storage.

*Interaction Between Client-Side and Admin Panel:*

The system provides two distinct interfaces: the client-side and the admin panel. These interfaces interact with the same underlying database but cater to different user roles and needs.

Client-Side Interaction: The client-side is designed for passengers who wish to book flights. Users can access the client-side interface through a web browser. They can register, search for available flights, book tickets, make payments, view e-tickets, check flight status, and manage their bookings. The interaction on the client-side is primarily focused on the user experience, ensuring a seamless and user-friendly process from flight search to ticket confirmation.

Admin Panel: The admin panel is reserved for administrators who oversee and manage the system's operations. Administrators access the admin panel through a secure login. Within the admin panel, they can manage flight schedules, cities, airlines, booking records, reservations, and flight issues. Additionally, administrators can perform critical functions such as marking flights as departed or arrived. The admin panel provides a comprehensive set of tools to monitor and control the entire system, ensuring efficient flight management and user support.

*Client-Side Features:*

The client-side features are tailored to meet the needs of passengers and provide an exceptional booking experience. Key features include:

* User Registration: Passengers can create accounts by providing essential personal information.
* Booking Airline Tickets: Users can search for available flights based on criteria such as departure and arrival locations, dates, class, and the number of passengers. They can then book tickets for their preferred flights.
* Simple Payment System: The system integrates a secure payment gateway, allowing users to make payments for their bookings online.
* View E-Ticket: After booking, passengers can access electronic tickets that contain detailed flight information, including the airline name, departure and arrival locations, boarding time, seat details, gate number, and more.
* Cancel Tickets: Passengers have the option to cancel their bookings and receive automated refunds as per the system's policies.
* Print Tickets: Users can print their tickets for reference and use during check-in at the airport.
* Check Flight Status: Passengers can check the real-time status of their booked flights to stay informed about any delays, cancellations, or other changes.
* View Total Amount: The system provides a summary of the total amount paid for bookings, including the fare, taxes, and any additional charges.

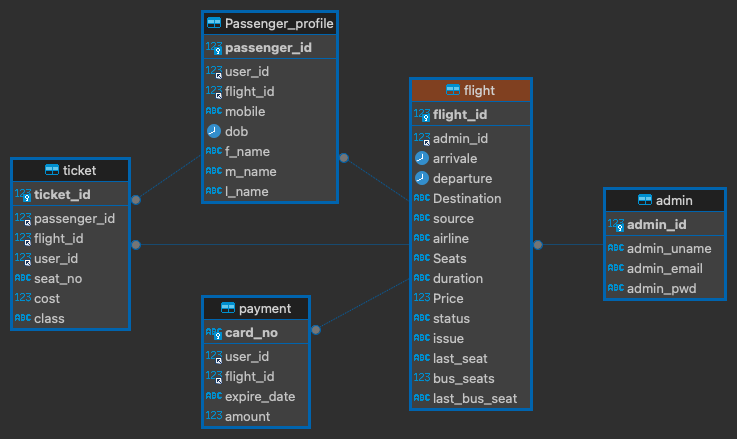
*Admin Panel Features:*

The admin panel is a robust set of tools designed to empower system administrators. Key admin-side features include:

* Admin Panel Overview: Administrators can log in securely to access the admin panel, where they have full control over the system's functions.
* Managing Flights: Administrators can manage flight schedules, including creating, updating, and deleting flight records. They can specify departure and arrival locations, date-time, duration, pricing, and the corresponding airline.
* Managing Cities and Airlines: Admins can set up city locations and airline details, providing a basis for flight scheduling.
* Listing Booking Details: The admin panel enables administrators to view and manage all booking records processed by passengers. They can access essential booking information, such as passenger names, booking amounts, destination, and airline details.
* Handling Reservations: Administrators can view comprehensive records of total passengers, amounts, flights, and airlines. They have access to detailed reservation information, helping them track and manage passenger bookings effectively.
* Managing Flight Issues: In the event of flight-related issues, administrators can document and resolve them by marking flights appropriately and ensuring passenger support.

**Entity-Relationship (ER) Diagram**

The ER diagram is a visual representation of the database schema, showing how tables are related to each other and the attributes they contain. In your Online Flight Booking System, you have several tables that store information related to passengers, flights, airlines, administrators, payments, and more. Below is an ER diagram for your system:



*Entities:*

Admin: Represents system administrators who have access to the admin panel. The admin\_id is the primary key for this entity, and it contains details such as the admin's username, email, and password.

Airline: Contains information about airlines, with airline\_id as the primary key. It includes attributes like the airline's name and the number of seats available.

Cities: Stores city names (city) used in flight schedules. It serves as a reference table.

Feedback: Captures feedback provided by users, with feed\_id as the primary key. It records user email, responses to feedback questions (q1, q2, q3), and a rating (rate).

Flight: Represents flight schedules, with flight\_id as the primary key. It contains information about flight arrival and departure times, source and destination cities, airline, seats, duration, price, status, and other details. It also references the admin table through admin\_id.

Passenger\_Profile: Stores passenger profiles, with passenger\_id as the primary key. It includes details like mobile number, date of birth (dob), and passenger names (f\_name, m\_name, l\_name). It references the users table through user\_id and the flight table through flight\_id.

Payment: Contains payment details, with card\_no as the primary key. It stores user payment information, references the users table through user\_id, and links to the flight table through flight\_id.

Ticket: Stores information about booked tickets, with ticket\_id as the primary key. It records the passenger's seat number, cost, class, and references the passenger\_profile table through passenger\_id and the flight table through flight\_id.

Users: Represents system users, with user\_id as the primary key. It contains user details, such as username, email, and password.

Packages: A table not mentioned in your original schema but added for completeness. It is used for storing information about tour packages, including package name, duration, cost, image, description, and the creation timestamp.

*Relationships:*

Admin Manages Flight: An admin can manage multiple flights. This one-to-many relationship is established through the admin\_id in the Admin entity and the admin\_id in the Flight entity.

Flight Operated By Airline: Each flight is operated by a specific airline. This one-to-one relationship is established through the airline attribute in the Flight entity and the airline\_id in the Airline entity.

User Books Ticket: A user can book multiple tickets. This one-to-many relationship is established through the user\_id in the Users entity and the user\_id in the Ticket entity.

Passenger Books Ticket: A passenger can book multiple tickets. This one-to-many relationship is established through the passenger\_id in the Passenger\_Profile entity and the passenger\_id in the Ticket entity.

Flight Has Tickets: Each flight can have multiple tickets booked. This one-to-many relationship is established through the flight\_id in the Flight entity and the flight\_id in the Ticket entity.

User Makes Payment: A user can make multiple payments. This one-to-many relationship is established through the user\_id in the Users entity and the user\_id in the Payment entity.

Flight Has Payments: Each flight can have multiple payments associated with it. This one-to-many relationship is established through the flight\_id in the Flight entity and the flight\_id in the Payment entity.

**Flight Management System**

The Flight Management System in the Online Flight Booking System is a critical component responsible for managing flight schedules, airlines, and their related details. This system empowers administrators to create, update, and delete flight schedules and airline records, ensuring that the system is up-to-date and accurate. Here, we'll provide a detailed overview of how flight schedules and airlines are managed, the process for setting up new flights, and how flights and airline records can be removed or updated.

*Managing Flight Schedules:*

Managing flight schedules involves creating, updating, and deleting flight records to maintain an accurate and efficient flight booking system.

*Creating New Flights:*

* To create a new flight, administrators log in to the admin panel and navigate to the flight management section.
* They enter the required details, including departure and arrival date and time, source and destination cities, flight duration, price, and the airline operating the flight.
* Once the details are entered, they can save the new flight record, making it available for passengers to book.

*Updating Flight Records:*

* Flight records may need updates due to various reasons, such as schedule changes or corrections.
* Administrators can select an existing flight from the system's records and edit any of the flight details as needed.
* After making the necessary changes, they save the updated record, ensuring that passengers have access to the most current information.

*Removing Flight Records:*

* Occasionally, flights need to be removed from the system, either due to cancellations or other reasons.
* Administrators can select a specific flight and choose to delete it, effectively removing it from the available flight schedules.

*Managing Airlines:*

* Managing airlines is an essential part of the Flight Management System, as it helps keep track of the airlines operating flights and their respective seat capacities.

*Adding New Airlines:*

* To add a new airline, administrators navigate to the airline management section in the admin panel.
* They input the airline's name and specify the number of seats available for flights operated by that airline.
* This information is essential for accurately managing seat availability when creating new flight schedules.

*Updating Airline Records:*

* If there are changes in the seat capacity or any other airline-related information, administrators can update the airline record.
* They select the airline they wish to modify and make the necessary changes to the seat count or any other details.
* After saving the updated record, the system reflects the changes for all related flight schedules.

*Removing Airlines:*

* In some cases, an airline might no longer operate flights within the system.
* Administrators can remove an airline record, which will affect all associated flight schedules.
* By effectively managing flight schedules and airlines, administrators ensure that the Online Flight Booking System remains up-to-date, accurate, and reliable for passengers. The ability to create, update, and remove flight records and airline details allows for flexibility in responding to changes in the airline industry and maintaining a seamless booking experience for users.

**User Guide**

*User Registration:*

* To begin using the system, visit the website and click on the "Register" or "Sign Up" button.
* Fill out the registration form with your personal details, including your username, email, and a secure password. Submit the form.
* Once registered, you can log in using your credentials.

*Searching for Flights:*

* After logging in, you will be directed to the main user dashboard. Here, you can find a flight search tool.
* Select your departure and arrival locations, along with your preferred dates, class, and the number of passengers.
* Click the "Search" button to view available flight options.

*Booking Airline Tickets:*

* Browse through the available flight options presented to you after searching.
* Select a flight that suits your preferences and click on it.
* You will be prompted to provide passenger details, including name, contact information, and date of birth.
* After entering passenger details, proceed to the payment process.

*Payment Process:*

* Enter your payment card information, including the card number and expiration date.
* Confirm the payment amount.
* After successful payment, you will receive a confirmation of your booking.

*Viewing E-Tickets:*

* In the user dashboard, navigate to the "My Tickets" or a similar section.
* You will find a list of your booked flights. Click on a flight to access your e-ticket, which contains essential flight details.

*Canceling Tickets:*

* In the user dashboard, locate the "My Tickets" or similar section.
* Find the flight you wish to cancel and click on it.
* Follow the cancellation process, which may involve confirming the cancellation.

*Printing Tickets:*

* In the user dashboard, navigate to the "My Tickets" section.
* Select the flight for which you want to print a ticket.
* Look for a "Print" or "Download" option to obtain a printable version of your ticket.

*Checking Flight Status:*

* Visit the user dashboard and find the "Check Flight Status" or similar section.
* Enter the details of the flight you want to check (e.g., flight number, date).
* The system will provide you with the real-time status of the flight.

*Viewing Total Amount:*

* On the user dashboard, check the "Total Amount" or a similar section.
* It will display the total amount you have spent on flight bookings, including fare, taxes, and other charges.

*List of Today's Flights:*

* Navigate to the "Today's Flights" or similar section in the user dashboard.
* You will find a list of all flights scheduled for the current date.
* Managing Flight Departures and Arrivals:

**Admin Guide**

*Admin Panel Overview:*

* Access the admin panel by logging in with your administrator username and password.
* The admin panel offers an array of features, including flight management, city and airline management, booking details, reservations, and more.

*Managing Flights:*

* In the admin panel, navigate to the "Manage Flights" section.
* Create new flight records by providing details such as departure and arrival times, cities, airlines, duration, price, and other relevant information.
* Update existing flight records as needed to reflect changes in schedules, pricing, or other details.
* Delete flights that are no longer relevant or have been canceled.

*Managing Cities and Airlines:*

* In the admin panel, access the "Manage Cities" and "Manage Airlines" sections.
* Add new city and airline records by providing their respective details.
* Update city and airline records when necessary, such as modifying the names of cities or adjusting seat capacities for airlines.

*Listing Booking Details:*

* In the admin panel, go to the "List Booking Details" section.
* This section provides an overview of all booking records processed by passengers. You can view essential booking information, such as passenger names, booking amounts, destinations, and airline details.

*Handling Reservations:*

* In the admin panel, navigate to the "Handling Reservations" section.
* Here, you can access detailed reservation information, helping you track and manage passenger bookings effectively.

*Managing Flight Issues:*

* In the admin panel, visit the "Manage Flight Issues" section.
* If a flight encounters issues, you can document and address them here by marking the flight with the appropriate status and providing details of the issue.

**Conclusion**

The Online Flight Booking System Project in Python and MySQL represents a comprehensive and efficient solution for booking airline tickets and managing flight schedules. This documentation has provided a detailed overview of the project's key components, features, and the underlying database schema. Here, we summarize the project's achievements and potential future enhancements:

*Key Achievements*

Streamlined Booking Process: The system offers passengers a user-friendly and convenient way to search for flights, book tickets, and make payments. It simplifies the traditionally complex process of airline ticket booking.

Efficient Flight Management: Administrators have access to a powerful admin panel that allows them to manage flight schedules, airlines, and booking records. This ensures accurate and up-to-date flight information for passengers.

Data Centralization: The system centralizes flight-related data, promoting efficient data management and retrieval, which is vital for airlines and administrators.

Cost Reduction: The adoption of an online booking system can reduce administrative and operational costs, making it a cost-effective solution for airlines.

Competitive Advantage: Airlines that offer an efficient online booking system gain a competitive edge by attracting more passengers.

Learning Resource: The project serves as a valuable learning resource for individuals interested in web development and database management, providing insights into creating similar systems.

*Potential Future Enhancements:*

Integration with Additional Services: Expanding the system to include hotel reservations, car rentals, and other travel-related services could offer passengers a more comprehensive travel booking experience.

Enhanced Reporting and Analytics: Developing robust reporting and analytics features for administrators could provide insights into booking trends, revenue analysis, and passenger behavior.

Mobile Application: Developing a mobile app version of the system can cater to the growing number of users who prefer to book flights and access travel information through mobile devices.

User Reviews and Ratings: Integrating a user review and rating system for flights and airlines can provide valuable feedback for both passengers and airlines.

Dynamic Pricing: Implementing dynamic pricing algorithms based on demand, time, and other factors can optimize revenue for airlines.

Multi-language and Currency Support: Expanding language and currency options can make the system accessible to a global audience.

Enhanced Security: Continuously improving and updating security measures to protect user and payment data is essential in today's digital environment.

Integration with Airlines' Reservation Systems: Integration with airlines' reservation systems can provide real-time flight information and booking options.