FLIGHT EXPLORER

Objective:
To create a flight booking system that allows users to search for available flights, check seat availability, make bookings, and process payments.

Scope:
- The system will handle user interactions for booking flights, checking seat availability, and managing passenger details.
- It will connect to a MySQL database to retrieve flight details and update seat availability

Requirements Analysis

- Connect to a MySQL database.
- Display available flights based on user input.
- Check seat availability.
- Update seats after successful payment.
- Manage passenger details and trip summary.
- Handle user inputs and validations.
- The system should be user-friendly.
- It should handle errors gracefully.
- The system must ensure data consistency and integrity.

Design

- ★ Architecture:
  - ○ The system follows a modular design with separate functions for each task:
    - ■ connecting to the database,
    - ■ showing available flights,
    - ■ checking seat availability,
    - ■ updating seats after payment, and
    - ■ retrieving flight details.

- ★ Database Schema:
  - ○ `available_flights`: Stores information about available flights including:

- flight_id,
- departure_city,
- destination_city, and
- departure_month.

- flight_details: Stores detailed flight information including:
  - flight_id,
  - departure_time,
  - arrival_time,
  - airline, and
  - seats.

User Interface:

- Command-line interface for user interaction.
- Options for booking flights, viewing destinations, selecting departure city, etc.

Technologies:

- Python for programming.
- MySQL for the database.
- mysql.connector for connecting Python with MySQL.
- tabulate for displaying data in a tabular format.

Code Structure:

- connect_to_mysql(): Establishes a connection to the MySQL database.
- show_available_flights(): Displays available flights based on user inputs.
- check_seats_availability(): Checks if the requested number of seats are available.
- update_seats_after_payment(): Updates the number of available seats after payment.
- get_flight_details(): Retrieves detailed information about a specific flight.
- Main program logic handles user interactions and invokes these functions as needed.