PROGRESS REPORT - Air Ticket Reservation System
Course Project Part 3
Due: November 20, 2025

1. Team member (working alone): Willem Neuefeind Lessig
2. Technologies

- Backend Framework: Python 3.12 with Flask 3.0.0
- Database: MySQL (using mysql-connector-python 8.2.0)
- Frontend: HTML5, CSS3, Jinja2 templating
- Session Management: Flask sessions
- Password Hashing: MD5 (as specified in requirements)
- Development Environment: Python virtual environment

3. Individual project

4. I will work on all of it early and submit early so it is over with.
The only remaining work I have to do is to add the staff systems and
then to test the system for errors.

5. Had some date formatting issues, but fixed them. Otherwise everything seems to be working
currently.

6. 6/13 use cases

Public Features (2/2):
 1. View Public Info / Search Flights - Users can search for future flights by source city/airport,
destination city/airport, and departure date

Customer Features (5/5):
 1. View My Flights - Customers can view their purchased flights with filtering options (date
range, source/destination, future/past/all)
 2. Search for Flights - Customers can search for future flights (one-way or round-trip) with same
functionality as public search
 3. Purchase Tickets - Customers can purchase tickets for flights with payment information,
seat availability checking, and validation
 4. Give Ratings and Comment on Previous Flights - Customers can rate (1-5) and comment on
flights they've purchased and already taken
 5. Logout - Session destruction with goodbye message

Airline Staff Features (0/7):
 1. View flights (enhanced)
 2. Create new flights
 3. Change Status of flights
 4. Add airplane in the system
 5. View flight ratings
 6. View reports
 7. Staff login and registration

7. Yes, all tables have been added to the database.

Tables Created:
1. Airport - Stores airport information (code, city, country, type)

2. Airline - Stores airline names
3. Airplane - Stores airplane information (weak entity, depends on Airline)
4. Flight - Stores flight information (weak entity, depends on Airline)
5. Customer - Stores customer account information
6. AirlineStaff - Stores airline staff account information
7. Phone - Stores multiple phone numbers for staff (weak entity)
8. Ticket - Stores ticket purchases with payment information
9. Review - Stores customer ratings and comments for flights

Schema Corrections Made:
- Removed duplicate `airplane_airline` column from Flight table (uses `airline_name` directly)
- Removed `review_date` column from Review table

Additional Features Planned:
- Round-trip flight booking (currently UI supports it but backend needs implementation)
- Enhanced filtering and search capabilities
- Better error handling and user feedback
- Input validation improvements

8. Data has been entered into all the tables

Data Inserted:
- 1 Airline: Jet Blue
- 10 Airports: JFK, PVG, LAX, ORD, DFW, LHR, CDG, DXB, NRT, SYD
- 3 Airplanes: Boeing and Airbus aircraft with varying seat capacities
- 26 Flights: Mix of domestic and international routes, various dates, some delayed
- 5 Customers: With MD5 hashed passwords, addresses, passport info
- 3 Airline Staff: With MD5 hashed passwords, associated with Jet Blue
- 5 Phone Numbers: Multiple numbers for staff members
- 8 Tickets: Sample ticket purchases with payment information
- 1 Review: Sample review for a past flight

All data is stored in sample_data.sql file for easy database population.

9.
Front-end Pages Completed: 11 files

Templates (10 HTML files):
1. base.html - Base template with navigation and layout
2. index.html - Home page (shows different content for logged-in vs. not logged-in)
3. login.html - Customer login form
4. register.html - Customer registration form
5. search_flights.html - Flight search form (public)
6. search_results.html - Flight search results display
7. customer_home.html - Customer dashboard with upcoming flights
8. view_my_flights.html - View and filter customer's flights
9. purchase_ticket.html - Ticket purchase form with payment information
10. customer_reviews.html - Review and rating interface for past flights

Static Files (1 CSS file):
1. style.css - Complete styling for all pages (responsive design, forms, tables, navigation)