CENG-3502 : Dynamic Web Programming

Final Assignment: FlyTicket Project - Full Stack Final Project

---

Project Overview

Build a web application called **FlyTicket** for an airline company where users can:

- View available flights
- Search for flights
- Book (buy) tickets
- Admins can create, update, and delete flights

Students must create both **Frontend** and **Backend** parts.

Technologies to Use

- **Frontend**: React / Vue.js / Plain HTML-CSS-JS
- **Backend**: Node.js + Express / Django / Flask
- **Database**: MongoDB / PostgreSQL / MySQL

Key Features

User Side (Customer)

- View all available flights
- Search flights by origin, destination, date
- Book a ticket by entering passenger information
- View booking confirmation

Admin Side (Admin Panel)

- Admin login system (simple password login acceptable)
- Add, edit, delete flights
- View all ticket bookings

Database Models

1. City

- **city_id**: String
- **city_name**: String (81 Cities of Türkiye)

2. Flight

- **flight_id**: String
- **from_city**: Foreign key to City
- **to_city**: Foreign key to City
- **departure_time**: DateTime
- **arrival_time**: DateTime
- **price**: Number
- **seats_total**: Number
- **seats_available**: Number

3. Ticket

- **ticket_id**: String
- **passenger_name**: String
- **passenger_surname**: String
- **passenger_email**: String
- **flight_id**: Foreign key to Flight
- **seat_number**: String (optional)

4. Admin

- **username**: String
- **password**: String (hashed)

Flight Scheduling Special Rules

Flight Rules:

1. **81 Cities** of Türkiye must be available.
2. **No two flights from the same city** can **depart at the same hour**.
    - Example: If a flight departs from Istanbul at 10:00 AM, another flight cannot depart from Istanbul at 10:00 AM.
3. **No two flights can arrive at the same city at the same arrival time**.
    - Example: If a flight arrives in Ankara at 2:00 PM, no other flight can arrive in Ankara at 2:00 PM.

Note: `Please check all other rules.`

Pages to Develop

- Home page: Flight search and results
- Flight detail page: Ticket booking form
- Booking confirmation page
- Admin dashboard (Login protected)
- Admin create/update/delete flight pages

Deliverables

- Frontend code (GitHub repository)
- Backend code (GitHub repository)
- Database export (MongoDB dump or SQL file)
- README file including:
    - How to run the project
    - Admin login credentials
    - Technologies used

Bonus Features (Optional)

- Seat selection during ticket booking
- Send e-ticket by email (SMTP or Email API)
- Payment simulation
- User authentication system (Register/Login)
- Mobile responsive design

Important Reminders

- Validate flight rules on the **backend**, not only frontend!
- Provide clear and friendly error messages.
- Keep UI clean and functional.

Submission Deadline

- **Deadline**: 26/05/2025

Example API Endpoints (Suggestion)

- `GET /flights` - List all flights
- `POST /flights` - Create a new flight (Admin only)
- `PUT /flights/:id` - Update a flight (Admin only)
- `DELETE /flights/:id` - Delete a flight (Admin only)
- `POST /tickets` - Book a ticket
- `GET /tickets/:email` - List tickets by user email

Example Concept Sketches:

1. Admin Dashboard

- Title: "Flight Management"
- Table columns: Flight ID, From, To, Departure Time, Arrival Time, Price, Actions
- Buttons: [Edit] [Delete] [Add New Flight]

---

2. User Flight Search Page

- Form: From (Dropdown) - To (Dropdown) - Date (Date Picker) - [Search Flights]
- Flight Cards: Show each flight with Price, Time, Available Seats
- Book Button on each flight

---

3. Booking Confirmation Page

- "Success" badge
- Ticket Details: Passenger Name, Flight Details, Seat Number
- Option to Download E-Ticket

Let's Fly High with FlyTicket!

Good Luck Students!