# Airport Management System - DRF API Task

This task is designed to help you build a complete Airport Management System using Django REST Framework with function-based views. You'll implement APIs for both Admin and Passenger sides, covering functionalities like flight management, passenger registration, ticket booking, and more.

## Models to Create

### Airport

- name (CharField)  
- city (CharField)  
- country (CharField)  
- code (CharField - IATA code)  
- created\_at (DateTimeField)

### Flight

- flight\_number (CharField)  
- airline\_name (CharField)  
- from\_airport (ForeignKey to Airport)  
- to\_airport (ForeignKey to Airport)  
- departure\_time (DateTimeField)  
- arrival\_time (DateTimeField)  
- price (DecimalField)  
- available\_seats (IntegerField)

### Passenger

- user (OneToOneField to Django User)  
- name (CharField)  
- email (EmailField)  
- phone (CharField)  
- passport\_number (CharField)  
- nationality (CharField)

### Booking

- passenger (ForeignKey to Passenger)  
- flight (ForeignKey to Flight)  
- booking\_date (DateTimeField)  
- seat\_number (CharField)  
- status (choices: Confirmed/Cancelled)

## API Endpoints

### Airport APIs (Admin only)

- GET /airports/ – List all airports  
- POST /airports/ – Create airport  
- PATCH /airports/{id}/ – Update airport  
- DELETE /airports/{id}/ – Delete airport

### Flight APIs (Admin only)

- GET /flights/ – List all flights  
- POST /flights/ – Create flight  
- PATCH /flights/{id}/ – Update flight  
- DELETE /flights/{id}/ – Delete flight

### Passenger APIs

- POST /register-passenger/ – Register  
- GET /my-profile/ – Get profile  
- PATCH /update-profile/ – Update profile

### Booking APIs

- POST /book-flight/ – Book a ticket  
- GET /my-bookings/ – View my bookings  
- PATCH /cancel-booking/{id}/ – Cancel booking

## Additional Requirements

- passengers can book flights  
- Admin-only access to Airport and Flight APIs  
- Use only function-based views