**TÀI LIỆU MÔ TẢ LUỒNG DỮ LIỆU & ỨNG DỤNG FLIGHT ROUTE ADVISOR**

**Mục tiêu:** Xây dựng ứng dụng đơn giản gợi ý tuyến bay trung chuyển tối ưu, hiển thị trực quan bản đồ các đường bay và phân tích trung tâm (hub) hàng không.  
**Nguồn dữ liệu:** OpenFlights Dataset – gồm 3 file chính: airports.dat, airlines.dat, routes.dat.

**1. Kiến trúc tổng thể hệ thống**

**Mô hình tổng quát**

[UI – Streamlit]

│

▼

[Python Data Pipeline]

├─ Data loading

├─ Data cleaning

├─ Distance calculation

├─ Graph analysis

│

▼

[Output Visualization –Streamlit Map]

**Frontend (Streamlit)**

* Giao diện tương tác cho người dùng:
  + Nhập sân bay xuất phát và sân bay đến.
  + Hiển thị tuyến bay ngắn nhất (Shortest Path).
  + Hiển thị các trung tâm hàng không (Hub Analysis).
* Sử dụng thư viện **Folium** để hiển thị bản đồ trực quan.

**Data Pipeline (Python Modules)**

* Thực hiện các bước xử lý dữ liệu:
  + Đọc, làm sạch (clean) dữ liệu từ file .dat.
  + Chuẩn hóa và tính khoảng cách giữa các sân bay.
  + Tạo đồ thị mạng bay bằng NetworkX.
* Kết quả lưu vào các file .csv sạch, phục vụ trực tiếp cho Streamlit.

**Storage (CSV Files)**

* 3 file đầu ra sạch:
  + airports\_cleaned.csv
  + airlines\_cleaned.csv
  + routes\_graph\_ready.csv (đã tính khoảng cách)
* Toàn bộ pipeline chỉ cần đọc/ghi CSV.

**2. Luồng hoạt động tổng quát**

**Bước 1 – Người dùng thao tác**

* Chọn **sân bay đi (source)** và **sân bay đến (destination)**.
* Chọn **chế độ hiển thị**:
  + “Shortest Route”
  + “Hub Analysis”
  + “Full Network View”

**Bước 2 – Hệ thống xử lý**

1. Đọc dữ liệu từ các file CSV sạch (/data/cleaned/).
2. Xây dựng đồ thị bay (NetworkX Graph).
3. Chạy thuật toán:
   * nx.shortest\_path() → đường đi ngắn nhất.
   * nx.betweenness\_centrality() → phân tích hub.
4. Kết hợp dữ liệu tọa độ để hiển thị trên bản đồ.

**Bước 3 – Hiển thị kết quả**

* Hiển thị đường bay trực quan trên bản đồ Folium.
* Bảng thông tin:
  + Tổng khoảng cách.
  + Số chặng.
  + Các sân bay trung chuyển.
* Xuất tùy chọn ra CSV nếu cần.

**3. Quy ước dữ liệu và pipeline**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **Mô tả** | **Trường chính** | **Output** |
| **airports.dat** | Thông tin sân bay | airport\_id, iata, city, country, latitude, longitude | airports\_cleaned.csv |
| **airlines.dat** | Danh sách hãng bay | airline\_id, name, country, active | airlines\_cleaned.csv |
| **routes.dat** | Tuyến bay (các cạnh của đồ thị) | airline\_id, source\_airport\_id, destination\_airport\_id, stops | routes\_graph\_ready.csv |

**4. Cấu trúc dữ liệu đầu ra**

**4.1 Shortest Route (JSON ví dụ)**

{

"source": "SGN",

"destination": "LHR",

"path": ["SGN", "DXB", "LHR"],

"total\_distance\_km": 10432,

"legs": [

{"from": "SGN", "to": "DXB", "distance\_km": 6110},

{"from": "DXB", "to": "LHR", "distance\_km": 4322}

]

}

**4.2 Hub Analysis (JSON ví dụ)**

{

"country": "Vietnam",

"top\_hubs": [

{"airport": "SGN", "degree\_centrality": 0.65, "betweenness": 0.42},

{"airport": "HAN", "degree\_centrality": 0.51, "betweenness": 0.33}

],

"backup\_hubs": [

{"airport": "DAD", "potential": 0.25}

]

}

**5. Pipeline xử lý dữ liệu (CSV-Based)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Bước** | **Hàm chính** | **Input** | **Output** |
| 1 | load\_raw\_data() | .dat files | DataFrames |
| 2 | clean\_airports() | airports.dat | airports\_cleaned.csv |
| 3 | clean\_airlines() | airlines.dat | airlines\_cleaned.csv |
| 4 | clean\_routes() | routes.dat | routes\_cleaned.csv |
| 5 | enrich\_distance() | routes\_cleaned.csv + airports | routes\_graph\_ready.csv |
| 6 | export\_csv() | Clean DataFrames | CSV lưu tại /data/cleaned/ |

**6. Cấu trúc thư mục dự án**

flight\_route\_advisor/

│

├── data/

│ ├── raw/

│ │ ├── airports.dat

│ │ ├── airlines.dat

│ │ └── routes.dat

│ └── cleaned/

│ ├── airports\_cleaned.csv

│ ├── airlines\_cleaned.csv

│ └── routes\_cleaned.csv

│

├── pipeline/

│ ├── loader.py

│ ├── cleaner\_airports.py

│ ├── cleaner\_airlines.py

│ ├── cleaner\_routes.py

│ ├── utils\_distance.py

│ └── main\_pipeline.py

│

├── app/

│ └── streamlit\_app.py

│

└── notebooks/

└── flight\_route\_analysis.ipynb

**7. Quy trình sử dụng thực tế**

**Ví dụ:** Tìm tuyến bay ngắn nhất từ Việt Nam đến London

1. Người dùng chọn:
2. Source = SGN
3. Destination = LHR
4. Hệ thống chạy thuật toán Dijkstra trên dữ liệu CSV.
5. Tính tổng khoảng cách & trung chuyển.
6. Vẽ tuyến bay trực tiếp trên bản đồ Folium.
7. Hiển thị thông tin chi tiết trên giao diện Streamlit.