**HEALTHCARE PATIENT ADMISSION RECORDS DATA TRANSFORMATION**

**Pandas-based (no visualization) mini case study**

**SUBMITTED BY**

**BHAGYASHREE HOKRANI**

**4GW23CI005**

[**bhagyashreehokrani2023@gmail.com**](mailto:bhagyashreehokrani2023@gmail.com)

**02/09/2025**

**INDEX**

* **PROJECT DESCRIPTION**
* **PROJECT OVERVIEW**
* **DETAILED EXPLANATION**
* **DATASET DESCRIPTION**
* **CODE**
* **FRONTEND INTERFACE DESIGN**

**PROJECT DESCRIPTION**

The project focuses on transforming and cleaning patient admission records in a healthcare system. Using Pandas, you will perform data wrangling tasks like handling missing values, formatting dates, filtering based on conditions, grouping for analysis, and generating new derived columns. The goal is to prepare the dataset for insights such as patient stay duration, department utilization, and admission trends.

**PROJECT OVERVIEW**

This project ingests hospital patient admission records, cleans and standardizes the data with **Pandas**, derives useful fields (like patient stay duration), filters and aggregates to produce operational insights (e.g., cardiology long-stays, admissions per department), and exports a clean CSV (transformed\_admissions.csv). It’s designed as a small ETL-style pipeline you can run standalone or call from a Flask endpoint after an upload

**DEATAILED EXPLANATION**

Step-by-step use-case explanations

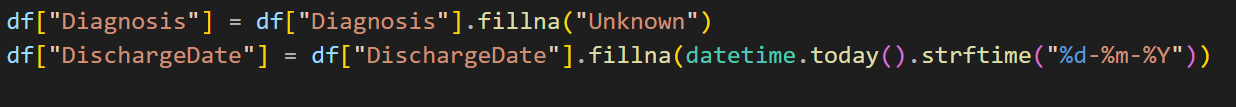
**Dataset Loading**

****

Reads the patient\_admissions.csv file into a Pandas DataFrame.

This gives us the raw dataset to work with

**Handling Missing Values**



Missing values in **Diagnosis** → filled with "Unknown".

Missing **DischargeDate** → filled with today’s date (simulating ongoing patients).

**Data Type Conversion**

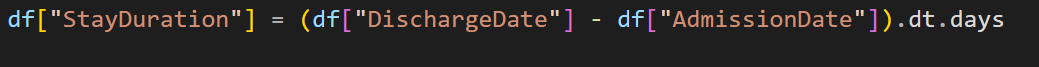
A screen shot of a computer

AI-generated content may be incorrect.

Converts admission and discharge dates into datetime objects.

This makes date arithmetic possible (e.g., calculating stay duration).

**Derived Column – Stay Duration**



Creates a new column showing patient **length of stay** in days.

Essential for analysis like longest stays, averages, and trends.

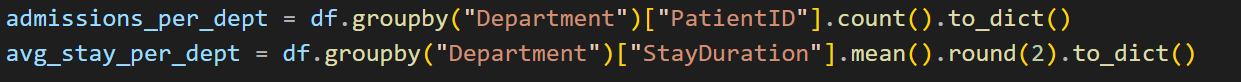
**Filter Specific Records**



Filters **Cardiology patients** who stayed for more than 5 days.

Gives deeper insight into long-term admissions.

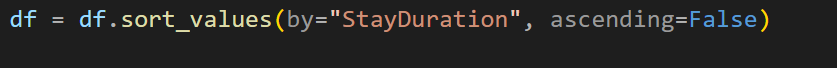
**Grouping & Aggregation**



**Admissions per department** → counts number of patients per department.

**Average stay duration per department** → calculates mean stay.

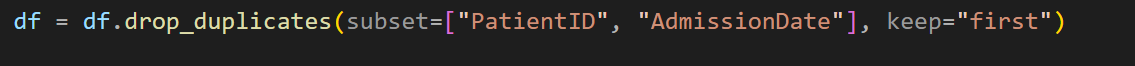
**Sorting by Stay Duration**



Sorts patients by longest hospital stay.

Helps easily spot patients with abnormal or prolonged admissions.

**Removing Duplicates**



Ensures no duplicate entries exist for the same patient with the same admission date.

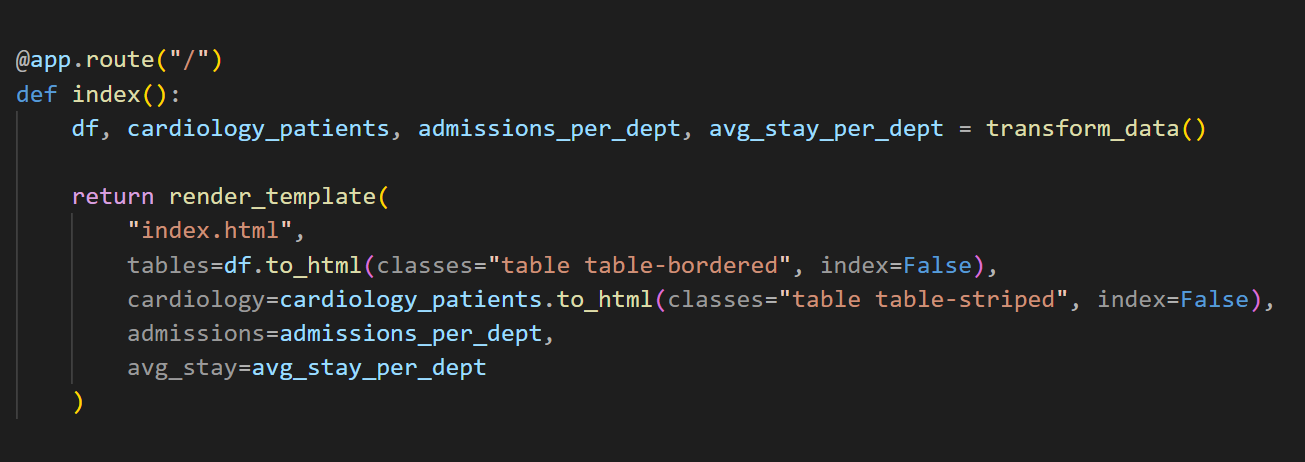
**Saving Transformed Data**

****

Saves the **cleaned & transformed dataset** into a new CSV file.

Ensures data is reusable for future analytics.

**Flask Web Integration**



Runs the transformation pipeline each time / is accessed.

Passes the results to **index.html** for frontend display:

* + Full transformed dataset (with stay duration, sorting, cleaning).
  + Cardiology patients (stay > 5 days).
  + Admissions per department.
  + Average stay duration per department.

**Dataset description**

|  |  |  |
| --- | --- | --- |
| **COLUMN NAME** | **DATA TYPE** | **DESCRIPTION** |
| Patient ID | Integer/String | Unique identifier for each patient |
| Name | String | Patient full name |
| Age | integer | Patient age in years |
| Gender | String | Patient gender (Male/Female/Other) |
| Department | String | Cardiology, Neurology, orthopaedics |
| Diagnosis | String | Diagnosis description (can be missing) |
| Admission Date | Date | Date when patient was admitted (format: DD-MM-YYYY) |
| Discharge Date | Date | Date when patient was discharged (format: DD-MM-YYYY) |
| Stay Duration | Integer | Calculated: difference between discharge and admission dates in days |

**CODE**

**BACKEND CODE : app.py**

**from flask import Flask, render\_template**

**import pandas as pd**

**from datetime import datetime**

**app = Flask(\_\_name\_\_)**

**def transform\_data():**

**# 1. Load Dataset**

**df = pd.read\_csv("patient\_admissions.csv")**

**# 2. Handle Missing Values**

**df["Diagnosis"] = df["Diagnosis"].fillna("Unknown")**

**df["DischargeDate"] = df["DischargeDate"].fillna(datetime.today().strftime("%d-%m-%Y"))**

**# 3. Data Type Conversion**

**df["AdmissionDate"] = pd.to\_datetime(df["AdmissionDate"], dayfirst=True, errors="coerce")**

**df["DischargeDate"] = pd.to\_datetime(df["DischargeDate"], dayfirst=True, errors="coerce")**

**# 4. Create Derived Column**

**df["StayDuration"] = (df["DischargeDate"] - df["AdmissionDate"]).dt.days**

**# 5. Filter Data**

**cardiology\_patients = df[(df["Department"] == "Cardiology") & (df["StayDuration"] > 5**

**# 6. Group and Aggregate**

**admissions\_per\_dept = df.groupby("Department")["PatientID"].count().to\_dict()**

**avg\_stay\_per\_dept = df.groupby("Department")["StayDuration"].mean().round(2).to\_dict()**

**# 7. Sort Data**

**df = df.sort\_values(by="StayDuration", ascending=False)**

**# 8. Remove Duplicates**

**df = df.drop\_duplicates(subset=["PatientID", "AdmissionDate"], keep="first")**

**# 9. Save Transformed Data**

**df.to\_csv("transformed\_admissions.csv", index=False)**

**return df, cardiology\_patients, admissions\_per\_dept, avg\_stay\_per\_dept**

**@app.route("/")**

**def index():**

**df, cardiology\_patients, admissions\_per\_dept, avg\_stay\_per\_dept = transform\_data()**

**return render\_template(**

**"index.html",**

**tables=df.to\_html(classes="table table-bordered", index=False),**

**cardiology=cardiology\_patients.to\_html(classes="table table-striped", index=False),**

**admissions=admissions\_per\_dept,**

**avg\_stay=avg\_stay\_per\_dept**

**)**

**if \_\_name\_\_ == "\_\_main\_\_":**

**app.run(debug=True)**

**FRONTEND CODE : templates/index.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Healthcare Admissions Dashboard</title>**

**<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">**

**</head>**

**<body class="container mt-4">**

**<h1 class="mb-4 text-center">🏥 Healthcare Admissions Dashboard</h1>**

**<h3>📊 Full Transformed Dataset</h3>**

**<div class="table-responsive">**

**{{ tables | safe }}**

**</div>**

**<h3 class="mt-5">❤️ Cardiology Patients (Stay > 5 Days)</h3>**

**<div class="table-responsive">**

**{{ cardiology | safe }}**

**</div>**

**<h3 class="mt-5">📌 Admissions per Department</h3>**

**<ul>**

**{% for dept, count in admissions.items() %}**

**<li><b>{{ dept }}:</b> {{ count }}</li>**

**{% endfor %}**

**</ul>**

**<h3 class="mt-5">⏱️ Average Stay Duration per Department</h3>**

**<ul>**

**{% for dept, avg in avg\_stay.items() %}**

**<li><b>{{ dept }}:</b> {{ avg }} days</li>**

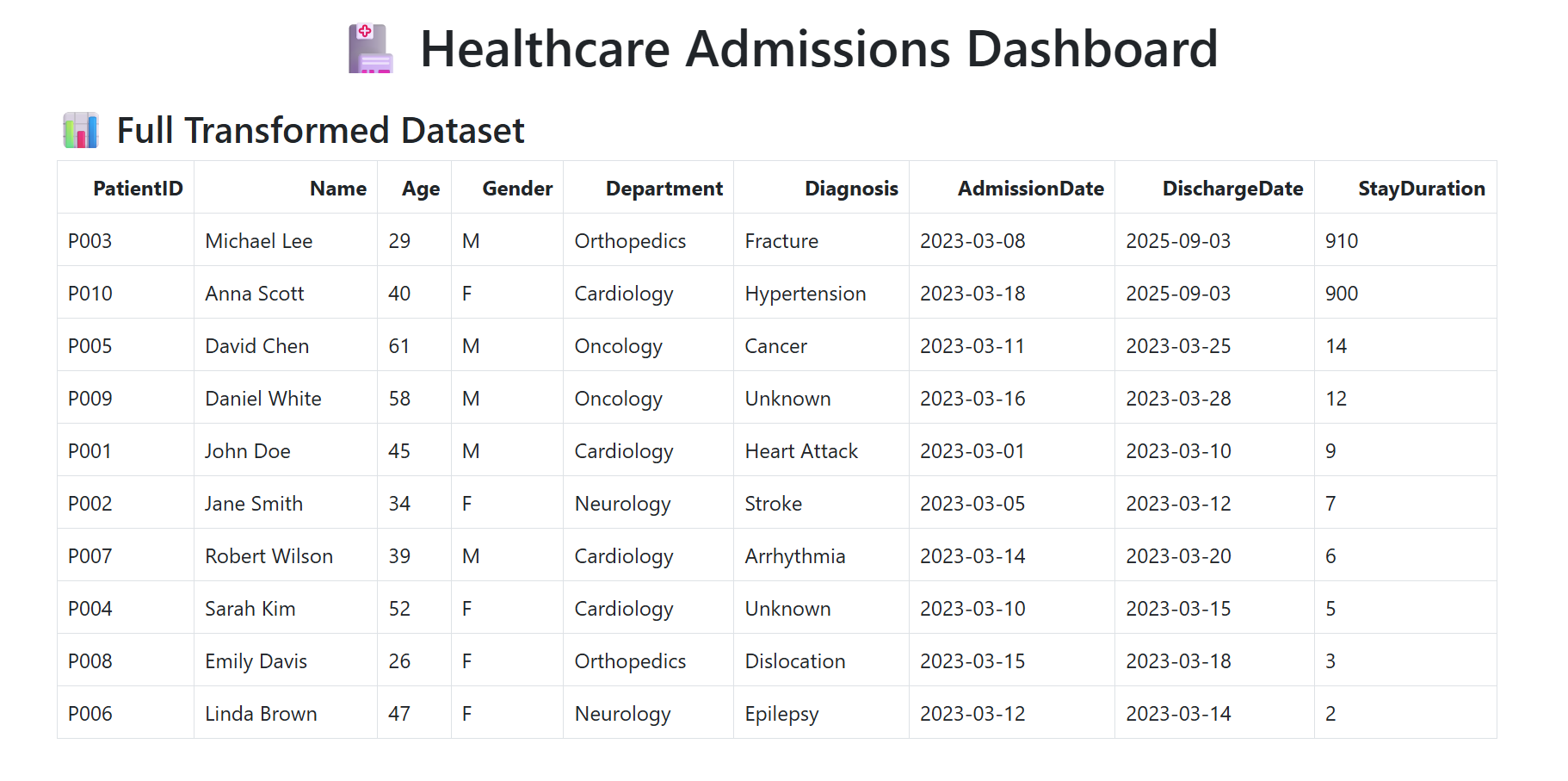
**{% endfor %}**

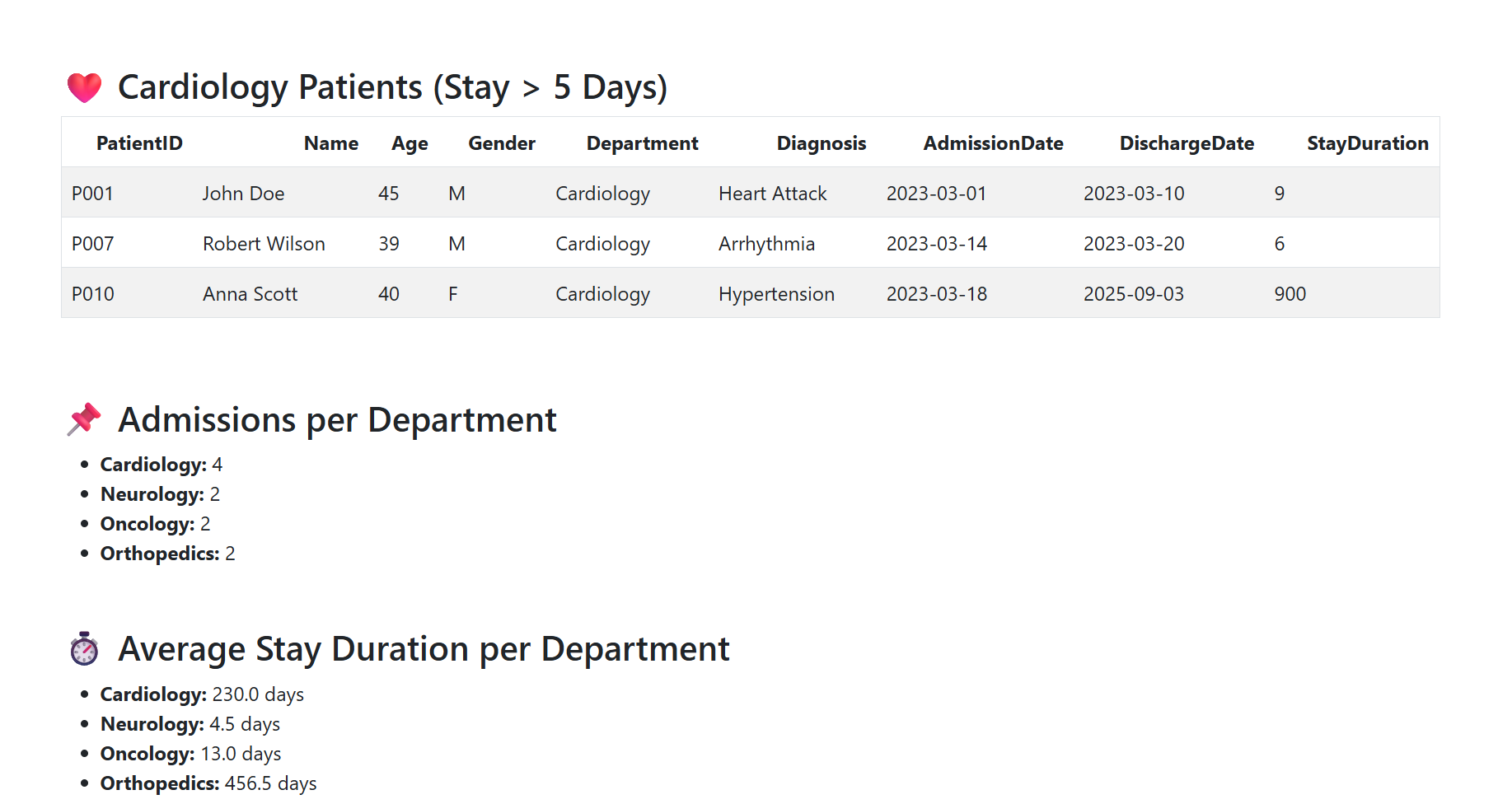
**</ul>**

**</body>**

**</html>**

**FRONTEND INTERFACE DESIGN**

****

****