Practical Exercise: Implementing ETL Using Python for a Healthcare Application

Problem Statement

You are tasked with building an ETL pipeline for a healthcare clinic. The goal is to extract data from a CSV file containing patient information and a simulated API providing diagnostic details. You will transform the data by cleaning it and enriching diagnostic information with patient details, then load the data into MongoDB for further reporting and analysis.

Step-by-Step Instructions

1) Extract Data

Patient data (CSV file):

You have a CSV file named patients.csv that contains basic patient information, such as ID, name, age, and gender.

This is a sample CSV file:

```
patient_id,name,age,gender
P001,John Doe,45,Male
P002,Jane Smith,30,Female
P003,Robert Taylor,50,Male
```

Code to extract patient data from the CSV file:

```
import pandas as pd

# Extract patient data from CSV file
patients_df = pd.read_csv('patients.csv')
print("Extracted Patient Data:")
print(patients_df)
```

Diagnostics data (simulated API):

Diagnostics data is retrieved from a simulated API that provides information about medical tests and results.

python code:

```
# Simulated API response for diagnostic data
diagnostic_data = [
    {"diagnostic_id": "D001", "patient_id": "P001", "test": "Blood
    Test", "result": "Normal"},
```

```
    {"diagnostic_id": "D002", "patient_id": "P002", "test": "X-Ray
   ", "result": "Fracture"},
    {"diagnostic_id": "D003", "patient_id": "P003", "test": "MRI",
    "result": "Normal"}
]

print("Extracted Diagnostic Data:")
print(diagnostic_data)
```

2) Transform Data

Clean patient data: Let's assume you need to filter out patients who are younger than 40 years old for a specific study.

Enrich diagnostic data with patient information: Join the diagnostics data with patient details (name, age, gender) to provide context for the test results.

3) Load Data into MongoDB

- Connect to MongoDB
- Load Patient Data into MongoDB
- Load Diagnostic Data into MongoDB

4) Automate the ETL Process

To make the ETL process reusable, create functions for each step and run the complete ETL pipeline.