⍰ Low-Level Design (LLD) – Hospital Visit Analysis

Difficulty Level: Medium | **Total Marks:** 20
Standards Followed: 6 Functions | 6 Visible Test Cases

□ Summary of Corrections (Based on SME Feedback)

- □ Used Pandas `read_csv()` and merging with `pd.merge()`
- □ Ensured visit duration and group operations follow clean logic
- □ Output structures strictly follow test case expectations
- □ Split logic into independently testable functions
- □ Sample results aligned with driver expectations

□ Concepts Tested

□ Reading CSVs into DataFrames

□ Merging using inner joins

□ Grouping and aggregation with `groupby()` and `count()`

□ Sorting using `.sort_values()`

□ Conditional filtering with `.isin()` and `.isnull()`

□ Aggregation with `.mean()` or `.sum()/len()`

□ Problem Statement

You are given two CSV files containing hospital records:

- `patients.csv` – Patient demographics
- `visits.csv` – Patient visits and their durations

Perform core data analysis using Pandas to extract key metrics:

- Load and merge the data
- Identify patients with most or no visits
- Calculate visit durations and averages

□ Operations

□ 1. Load Data

□ Load both CSVs (`patients.csv` and `visits.csv`) into Pandas DataFrames.
□ **Function Prototype:**

```
def load_data(patients_path: str, visits_path: str) -> tuple:
```

□ Input: "patients.csv", "visits.csv"
□ Output: Tuple – (patients_df, visits_df)
□ **Implementation Flow:**
• Use `pd.read_csv()` for both files
• Return both DataFrames as a tuple

□ 2. Join DataFrames

□ Perform an inner join on `patient_id` between patients and visits.
□ **Function Prototype:**

```
def join_data(patients_df: pd.DataFrame, visits_df: pd.DataFrame) ->
pd.DataFrame:
```

□ Output: Merged DataFrame
□ **Implementation Flow:**
• Use `pd.merge()` with `how="inner"`
• Join on `"patient_id"`
• Return the merged DataFrame

□ 3. Most Frequent Visitor

□ Find the patient_id who visited the hospital most often.
□ **Function Prototype:**

```
def most_frequent_visitor(df: pd.DataFrame) -> int:
```

□ Output: patient_id (int)
□ **Implementation Flow:**

• Group by `"patient_id"`
• Use `.count()` on `"visit_id"`
• Sort descending and take top result

□ 4. Longest Visit

□ Get the visit_id of the longest visit.
□ **Function Prototype:**

```
def longest_visit_id(df: pd.DataFrame) -> int:
```

□ Output: visit_id (int)
□ **Implementation Flow:**
• Sort DataFrame by `"duration"` descending
• Use `.iloc[0]` to get the top row
• Return `"visit_id"` from that row

□ 5. Patients with No Visits

□ Return list of patient_ids who never had a visit.
□ **Function Prototype:**

```
def patients_with_no_visits(patients_df: pd.DataFrame, visits_df:
pd.DataFrame) -> list:
```

□ Output: List of integers
□ **Implementation Flow:**
• Use `isin()` or left merge with `isnull()`
• Filter unmatched records
• Return patient_ids as a list

□ 6. Average Visit Duration

□ Calculate and return average visit duration.
□ **Function Prototype:**

```
def average_visit_duration(visits_df: pd.DataFrame) -> float:
```

□ Output: Float
□ **Implementation Flow:**
- Use `.mean()` on `"duration"` column
- Return result as float

□ Implementation Code

```
# □ Implementation Hints for Hospital Visit Analysis
import pandas as pd

class HospitalAnalyzer:

    def load_data(self, patients_path: str, visits_path: str) -> tuple:
        """Load two CSV files and return as DataFrames."""
        pass  # TODO

    def join_data(self, patients_df: pd.DataFrame, visits_df: pd.DataFrame) -> pd.DataFrame:
        """Inner join on patient_id and return merged data."""
        pass  # TODO

    def most_frequent_visitor(self, df: pd.DataFrame) -> int:
        """Return patient_id with most visits."""
        pass  # TODO

    def longest_visit_id(self, df: pd.DataFrame) -> int:
        """Return visit_id of visit with longest duration."""
        pass  # TODO

    def patients_with_no_visits(self, patients_df: pd.DataFrame, visits_df: pd.DataFrame) -> list:
        """Return list of patients with no visits."""
        pass  # TODO

    def average_visit_duration(self, visits_df: pd.DataFrame) -> float:
        """Return average visit duration."""
        pass  # TODO
```

□ Test Cases & Marks Allocation

Test Case ID	Description	Associated Function	Marks
TC1	Load both CSVs	load_data()	□ 3
TC2	Join on patient_id	join_data()	□ 3

Test Case ID	Description	Associated Function	Marks
TC3	Patient with most visits	most_frequent_visitor()	□ 4
TC4	Longest visit ID	longest_visit_id()	□ 3
TC5	Patients without visits	patients_with_no_visits()	□ 3
TC6	Average duration calculation	average_visit_duration()	□ 4

□ **Total Marks: 20**

□ Visible Test Cases (6)

□ **TC1: Load DataFrames**
□ Input: `"patients.csv", "visits.csv"`
□ Output: Tuple of valid DataFrames

□ **TC2: Join Data**
□ Input: two valid DataFrames
□ Output: merged DataFrame with combined fields

□ **TC3: Most Frequent Visitor**
□ Input: merged DataFrame
□ Output: patient_id with most entries

□ **TC4: Longest Visit ID**
□ Input: merged DataFrame
□ Output: visit_id with max duration

□ **TC5: No Visit Patients**
□ Input: patients_df and visits_df
□ Output: list of patient_ids with no match in visits

- **TC6: Average Visit Duration**
- Input: visits_df with duration column
- Output: average duration as float