SDA Project Phase 1

Phase 1: Functional & Data-Driven GDP Analysis

Objective

To design and implement a **data-driven GDP analysis system** using **functional programming** principles in Python, while enforcing the **Single Responsibility Principle (SRP)** and introducing configuration-based behavior.

Technical Constraints

- **Programming Language:** Python
- Students must use **functional programming constructs**, including:
  - map, filter, lambda
  - List or dictionary comprehensions
- Traditional loop-based implementations should be minimized.

Task

Build a Python-based system that analyzes World Bank GDP data and computes statistical results based on user-defined configuration.

Dataset

A CSV file containing World Bank GDP data with the following fields:

- Country Name
- Region
- Year
- Value (GDP)

Functional Requirements

1. Load GDP data from a CSV file.
2. Clean the dataset (handle missing values, correct data types).
3. Filter data based on:
    - Specific **Region**
    - Specific **Year**
    - Specific **Country**
4. Perform a statistical operation on the filtered data:
    - **Average GDP of Regions**
    - **Average GDP of a Country**
    - **Sum of GDP of Regions**
5. **Students must generate multiple visualizations** (explore python-libraries for it), including:
    - **Region-wise GDP plot** (e.g. Pie chart, Bar chart, etc).
    - **Year-specific GDP plot** (e.g. Line Graphs, Scattergram, Histogram, etc).
6. At least **two different chart types** must be used for each.
7. Each graph should be clearly labelled (title, axes).

Configuration-Driven Behavior

All filtering and computation logic must be driven by a **configuration file (config.json)**, which specifies:

- Region
- Year
- Operation to perform (average or sum)
- Dashboard output preferences

Important Constraint:
No hardcoded values for region, year, or operation are allowed in the source code.

Design & Architecture Requirements

- The system must be divided into clear modules, such as:
  - Data Loader
  - Data Processor
  - Dashboard / Presentation Layer
- Each module must have **a single responsibility**.
- Data loading, processing, and visualization must not be mixed.

Dashboard Requirements

- The dashboard acts as the **main entry point** of the application.
- It must display:
  - Selected configuration values
  - Computed statistical results
- Results must be visualized using **charts or diagrams** to resemble a real data analytics dashboard.

Error Handling

- Handle missing or invalid CSV files.
- Validate configuration fields.
- Display meaningful error messages through the dashboard.

Important Note:

- The project must be completed in **pairs** (groups of two students).
- Students must create a **GitHub repository**.
- Code must be committed **incrementally** throughout development.
- Git commit history will be considered during evaluation to assess:
  - Development consistency
  - Individual contribution
- The submitted code on **GCR will be checked** at the time of **Evaluation**, so kindly complete your work within the deadline.
- Late submissions **will not** be accepted under any circumstances.

Deliverables:

1. Complete Python Code Files.
2. Dashboard with statistical visualizations.
3. Config.json file.
4. GitHub repository link.