Python Exercise — Designing and Implementing an Interface

In this exercise, you will design and implement a small interface-based system in Python. The goal is to understand how interfaces can be represented using Abstract Base Classes (ABCs) and to practice writing clean, extensible code that enforces a simple contract between classes.

Objective

Create a Python program that defines an interface for exporting data and at least two concrete implementations that follow this interface (for example, exporting to JSON and to CSV).

Task Description

1. Define an interface using the ABC module (e.g., `class DataExporter(ABC):`).
   - Include two abstract methods: `export(data)` and `validate(data)`.
2. Create two concrete classes implementing this interface:
   • `JSONExporter` — saves data in JSON format.
   • `CSVExporter` — saves data in CSV format.
3. Implement a function `save_report(data, exporter)` that uses the interface, not the concrete class.
4. Demonstrate usage by exporting the same sample data through both exporters.
5. Include basic validation logic (e.g., ensure data is not empty).

Additional Challenges (Optional)

• Add a new exporter class, e.g., `XMLExporter`, and show how your code handles it without modification.
• Add error handling when validation fails.
• Print or save the output to files for inspection.

Hints

• Use `from abc import ABC, abstractmethod`.
• Use `json` and `csv` libraries for export.
• Use type hints (`Iterable`, `Mapping`) for cleaner interfaces.
• Remember that your main program should not depend on concrete implementations.

Expected Result

Your program should produce readable output in both JSON and CSV formats. The goal is not just to get working code, but to demonstrate understanding of interface design and polymorphism.