

**Challenge Description:**

Your task is to create a command-line tool or a web application that aggregates insurance policy data from multiple brokers, each providing data in different formats. The goal is to normalise the data into a consistent format, provide basic insights into the aggregated policies, and ensure the solution is well-tested.

**Requirements:**

1. **Data Ingestion:**
   * Assume you have two mock data sources, each representing an insurance broker, providing data in CSV format (sample data provided below).
   * Build a data ingestion mechanism to read data from these sources. You can use a hard-coded dataset or file paths.
2. **Data Normalisation:**
   * Parse the data from each source and transform it into a common data structure that can be easily processed and analysed.
   * Standardise common fields such as policy number, insured amount, customer, start date, end date, etc.
   * Handle differences in data structure across different formats.
3. **Data Aggregation:**
   * Aggregate the normalised data from all sources into a single collection.
4. **Basic Reporting:**
   * Calculate and display the total count of policies, count of customers, the sum of insured amounts, and the average policy duration (in days) across the two brokers for active policies. The broker considers an active policy to be where the start date has passed and the renewal date is in the future.
   * Allow the user to input a broker's name or ID and display the policies associated with that broker.
5. **Unit Testing:**
   * Use a testing framework of your choice (e.g. PHPUnit, JUnit, NUnit, pytest) to create and run tests.
   * Aim to achieve good coverage of your codebase with meaningful tests.

**Submission:**

Share your solution from a source control repository such as GitHub with any necessary instructions to run the application.

**Evaluation Criteria:**

You will be evaluated based on the quality of your code, data parsing and transformation logic, data aggregation accuracy, reporting functionality, unit testing coverage, and overall code organisation.