API Automation Challenge

# Objective

You are tasked with automating the testing of the RESTful API for a bookstore using the provided demo API (Books) from FakeRestAPI. The assessment will evaluate your ability to design a clean and maintainable API automation test framework, implement reusable code, handle various scenarios, and provide clear test reports.  
In addition to testing, you will be required to automate the process through a CI/CD tool like GitHub Actions, GitLab, Jenkins, etc.

# API Overview

The assessment includes the following Books API endpoints:

Books API:  
- **GET /api/v1/Books** – Retrieve a list of all books  
- **GET /api/v1/Books/{id}** – Retrieve details of a specific book by its ID  
- **POST /api/v1/Books** – Add a new book to the system  
- **PUT /api/v1/Books/{id}** – Update an existing book by its ID  
- **DELETE /api/v1/Books/{id}** – Delete a book by its ID

# Assessment Requirements

**1. Project Structure:**  
- Set up a well-structured API test automation project using a Python and testing framework of your choice (e.g., pytest, pyunit etc.)  
- Organize the project with a clean folder structure that promotes maintainability and reusability.

**2. Test Case Implementation:**  
- Cover both happy paths and edge cases for the Books API endpoints.

**3. Reporting:**  
- Generate a test report that provides a clear summary of the test executions, showing the status (pass/fail) of each test case.

**4. Code Quality:**  
- Ensure clean code practices: clear naming conventions, proper use of variables, functions, and classes.  
- Follow SOLID principles or similar best practices for maintainable test code.

**6. CI/CD (GitHub Actions, GitLab, Jenkins, etc.):**  
- Select a CI/CD tool and create a job (YAML) that:  
 - Build the code of the automated test cases  
 - Runs the tests  
 - Publish the test execution report as an artifact  
 - Is triggered on every push and pull request on selected branch

# Deliverables

Please provide the following via a Git repository:

- **Source Code:** Complete source code for your API automation test project  
- **Test Report**: A generated test execution report showing test case results (success/fail)  
- **CI/CD:** A YAML file to build and run the project in CI  
- **README File:** Clear setup and usage instructions:  
 - How to run tests locally  
 - How the CI/CD job is triggered and where to find the test results

# Additional Notes

- You are free to choose any testing framework you are comfortable with in python.  
- Ensure that your project is well-documented so that anyone reviewing it can easily set up and run the tests.  
- The assessment focuses not only on test coverage, but also on structure, maintainability, automation, and modern delivery practices.  
- Use FakeRestAPI (<https://fakerestapi.azurewebsites.net>).

# Time Limit

You have **3 calendar days** to complete the task from the moment it is assigned. This includes development of test cases, CI setup, and documentation.

# Evaluation Criteria

- Correctness and completeness of API test automation  
- Test coverage for both happy paths and edge cases  
- Project structure, code clarity, reusability, and documentation  
- CI/CD (correct, automated, usable)  
- Clarity and completeness of the README file