Home Assignment 24: Mini Test Management Tool using Python OOP and CSV

Learning Objective

In this assignment, you will build a **simple Test Management Tool** using Python.
You will learn how to:

- Create classes and objects
- Use constructors (`__init__`)
- Apply inheritance
- Store test execution results in a CSV file

This assignment is designed to resemble **real-world testing tools** like TestRail or JIRA.

Expected Time

- **Best Case:** 60 minutes
- **Average Case:** 90 minutes

Problem Statement

Create a Python program that:

- Stores test case details
- Executes test cases
- Saves the execution results into a **CSV file**
- Displays a test execution summary

Step-by-Step Requirements

Step 1: Create a `TestCase` class

This class represents **one test case**.

Constructor (`__init__`) should accept:

- `test_id`
- `test_name`
- `module`
- `status` (default value should be `"Not Executed"`)

Methods to implement:

1. `execute_test(result)`
   - Updates the test status (`Pass` or `Fail`)
2. `display_test_case()`
   - Prints test case details
3. `to_csv_row()`
   - Returns test details in list format (for CSV writing)

Step 2: Create an `AutomatedTestCase` class

This class represents **automated test cases**.

- It should **inherit from `TestCase`**
- Add one extra attribute:
  - `automation_tool` (example: Selenium, Playwright)

Override:

- `display_test_case()` to show automation tool
- `to_csv_row()` to include automation tool

Step 3: Create a `TestSuite` class

This class manages **multiple test cases**.

Constructor should accept:

- `suite_name`

Methods to implement:

1. `add_test(test_case)`
   - Adds a test case to the suite
2. `run_all_tests()`
   - Execute all tests (ask user to enter Pass/Fail)
3. `save_results_to_csv(file_name)`
   - Save all test results into a CSV file
4. `summary_report()`
   - Display:

        - Total tests
        - Passed tests
        - Failed tests
        - Not executed tests

Step 4: CSV File Format

The CSV file should contain:

```
Test ID, Test Name, Module, Status, Automation Tool
```

- For manual tests, automation tool can be `"NA"`.

Step 5: Main Program Execution

In the `main` block:

- Create:
    - 2 manual test cases
    - 2 automated test cases
- Add all test cases to a test suite
- Execute the tests
- Save results to `test_results.csv`
- Print the execution summary

Hints

- Use Python's built-in `csv` module
- Use `super().__init__()` in the automated test class
- Store test cases as **objects**, not dictionaries
- Do not write CSV logic inside `TestCase`

Expected Outcome

After completing this assignment, you should be able to:

- Understand how OOP applies to real testing problems
- Use constructors properly
- Apply inheritance in Python
- Save data into CSV files
- Build a simple but realistic test management system

💡 Tip for Learners

Think of this assignment as:

- `TestCase` → One test in TestRail
- `TestSuite` → A test cycle
- CSV file → Execution report