

CISC/CMPE 327, Fall 2025

Assignment 2: Library Management System - Extended Unit Testing & CI/CD

Due date: 12 October, 2025, 11:59 PM

Assignment Overview

Building upon Assignment 1, you will now complete the Library Management System implementation, create comprehensive test suites, leverage AI tools for test generation, compare AI-generated tests with human-written (your) tests, and finally set up a CI/CD workflow using GitHub Actions.

Learning Objectives

By the end of this assignment, you will be able to:

- Implement complete business logic functions following specifications
- Create comprehensive test suites for both existing and new functionality
- Use Large Language Models (LLMs) to generate effective test cases
- Configure GitHub Actions workflows for automated testing and continuous integration

Tasks to Complete

You are required to complete the following tasks:

1. **Complete Function Implementation [25%]** - Based on your report in first assignment, complete all the remaining function implementations and run the project locally. Write a short paragraph in the PDF report about your experience in completing the implementation. Specify if the implementation has been completed fully or partially.
2. **Comprehensive Test Suite Development [10%]** - Add additional test case scripts **only if it is required** for newly implemented functions (We assume you have already done so in Assignment 1). List the additional test cases in the report.
3. **AI-Assisted Test Generation [20%]** - Use LLM of your choice (recommended tools from Queen's: <https://www.queensu.ca/ai/ai-applications>) to generate test cases for all the functionalities as per specifications. Report which AI tool you used, what prompt you wrote and what follow-up prompts you used to interact with the AI tool. Copy/paste the generated test cases in the report.
4. **Test-Case Comparison & Analysis [20%]** - Compare the test cases generated by AI tools with the test cases you have written and report your findings in terms of test case quality and coverage in the PDF report.
5. **CI/CD Pipeline Setup [25%]**

- If you don't have a GitHub account, create one, log in, and then create a public repository named `cisc327-library-management-a2-[last_4_digit_student_id]`.
- In your local project root directory, create a github action workflow to facilitate CI/CD integration (demo will be given to you). See the video tutorial <https://www.youtube.com/watch?v=0aEJBygCn5Q> to get the know how of how to write GitHub action workflow for automated testing.
- Now, push the local project in GitHub repository you created earlier.
- Observe GitHub action to see if the test script is executed successfully as per the workflow. If done successfully, the CI/CD test status badge will be visible as shown below

Simple Flask Calculator App

Tests passing codecov 100%

A simple Flask application that provides a REST API for basic calculator operations.

Submission Instructions

Primary Submission: Make sure you specify your GitHub repository link at the beginning of report. Upload the PDF report named `A2_LastName_last4digitID.pdf` to OnQ.

Repository Access: Ensure your GitHub repository is **public** for grading access