

# INF1002 Final Report and Presentation Guidelines

## 1. Project Report (10%)

Your report should clearly describe **how your CMS program works, how it was developed, and how your team contributed.**

It should be **concise, well-structured, and professional**, showing clear understanding of your design, logic, and testing.

**Length:** 10-15 pages (single column, font size 11 or greater, standard margins).

The page count excludes the title page, table of contents, and appendix.

The report must be **self-contained** - if the examiner needs to look into the appendix to understand your work, that content will count toward the main page limit.

The structure below is to help you structure your report; it is not an exact template.

### Recommended Structure

#### 1. Title Page

Include:

- Project title
- Team ID and team name
- Team members (name, student ID, email)
- Date of submission

#### 2. Introduction (1 to 2 pages)

Briefly introduce the system you built:

- What the CMS does and why it was created.
- A short description of the main features / Enhancement Features (e.g., open, insert, update, delete, query, save).
- A short description of your additional unique feature.
- The scope and goals of your project.

#### 3. System Design and Implementation (6–8 pages)

Describe **how your system is structured and how it works internally.**

Include:

- **Overall design / program flow:**
  - A simple diagram or flowchart showing how commands are processed.
  - Description of the main modules and their relationships (e.g., main loop, file operations, data manipulation functions).
- **Data structure design:**

- Explain the key structures (e.g., struct StudentRecord) and why you chose them.
- **Key functions:**
  - For each major operation (OPEN, INSERT, QUERY, UPDATE, DELETE, SAVE, ETC), explain:
    - Purpose of the function
    - Key parameters and return values
    - How it works logically (in words, not full code)
    - Any special error handling or validation you added
- **Enhancement features:**
  - Describe any additional features like sorting, statistics, or validation improvements.
  - Explain briefly how you implemented them and why you chose that design.

#### 4. Test Cases and Validation (3-4 pages)

Document the **test cases** your team designed to verify that the CMS works correctly.

Include a **test table** with:

Test Case ID	Description	Input Command(s)	Expected Output	Reason for Test	Actual Result
--------------	-------------	------------------	-----------------	-----------------	---------------

You should:

- Cover **all main functions** and **edge cases** (e.g., duplicate IDs, invalid input).
- Include **at least 8–10 meaningful/quality test cases**.
- Comment briefly on what you learned from testing (e.g., which bugs you discovered and fixed).

#### 5. Team Contributions (1 page)

Summarize what each team member worked on. Remember, here you must also reflect on contributions from your AI-Teammate

**For example:**

##### Team Member Major Contributions

- A File handling and SAVE/OPEN functions
- B Query, Update, and data validation
- C Command parser and user interface
- D Test case design and report integration

### Team Member Major Contributions

AI              Logical flow, ideation for unique features

### 6. Reflection (No pages!)

This will be a separate form sent to you closer to the end of the module

---

### Presentation and Demonstration Recommended Structure (5%)

Your team must submit a **video presentation** (15 total) that includes:

#### Participation / Presentation

- Every member must speak briefly to practice presentation skills.  
You may divide sections as you prefer (e.g., each person covers 1–2 functions).

#### System Demonstration

- Show your CMS running live.
- Demonstrate all required functions (OPEN, INSERT, QUERY, UPDATE, DELETE, SHOW, SAVE), enhancement features and unique feature.
- Include a key/crucial test cases to prove correctness.

#### Code Structure Overview

- Give a clear, high-level walkthrough of your program:
  - File organization
  - Key data structures (struct, arrays, etc.)
  - How commands are processed
  - How file reading/writing is implemented
- Focus on the **logic flow** — don't explain every line.

#### Time Management

- Keep it concise but informative — Max of **15 minutes total**.
- Ensure transitions between presenters are smooth and clear.