FINAL PROJECT - STUDENT MANAGEMENT SYSTEM

You will create a Student Management System program. This program helps manage student information using TWO LISTS.

Your program will store:

1- Student IDs in one list

2- Student Names in another list

IMPORTANT: The two lists work together. This means:

1- The first ID belongs to the first Name

2- The second ID belongs to the second Name

Example:

student_ids = [101, 102, 103]

student_names = ["Ahmed Ali", "Sara Mohammed", "Omar Hassan"]

This means:

1- Student with ID 101 is named Ahmed Ali

2- Student with ID 102 is named Sara Mohammed

3- Student with ID 103 is named Omar Hassan

IMPORTANT RULES:

1- Both lists must always have the SAME LENGTH (same number of items)

2- When you add a student, you add to BOTH lists at the same time

3- When you delete a student, you delete from BOTH lists at the same position

Your program will allow users to:

1- Add new students (ID and Name)

2- View all students

3- Search for a student by ID

4- Search for a student by Name

5- Delete a student

6- Save all data to a file

7- Load data from a file

8- Exit the program

DETAILED FUNCTION REQUIREMENTS

FUNCTION 1: show_menu() :

What it does:

- Displays the main menu to the user

- Shows all 8 options in a clear format

- Makes it easy for the user to see all choices

Parameters: None

Returns: Nothing

What to print: Print the menu exactly as shown above with all 8 options.

══

FUNCTION 2: add_student(ids_list, names_list)

What it does:

- Asks the user for a student ID (number)

- Checks if this ID already exists in the ids_list

- If the ID already exists, show an error message and stop

- If the ID is new, ask for the student name

- Add the ID to the ids_list

- Add the Name to the names_list

- Show a success message

Parameters: ids_list , names_list

Returns: Nothing (but it changes the two lists by adding new items)

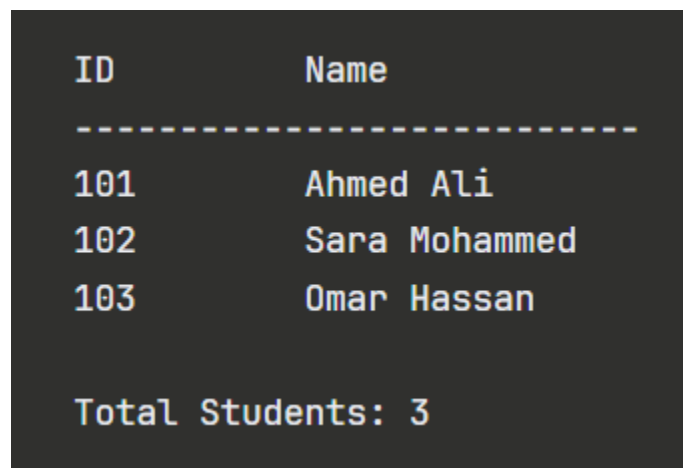══

What it does:

- Shows all students in a nice table format

- If there are no students, show "No students found"

- Display the total number of students at the end

Parameters: ids_list , names_list

Returns: Nothing

Example output:

```
ID              Name
----------------------------
101             Ahmed Ali
102             Sara Mohammed
103             Omar Hassan


Total Students: 3
```

FUNCTION 4: search_by_id(ids_list, names_list) :

What it does:

- Asks the user for a student ID to search for

- Looks through the ids_list to find that ID

- If found: shows the student's ID and Name

- If not found: shows "Student not found!"

Parameters: ids_list , names_list

Returns: Nothing

FUNCTION 5: search_by_name(ids_list, names_list)

What it does:

- Asks the user for a student name to search for

- Looks through the names_list to find matching names

- Shows all students whose name contains the search text

- If no match found: shows "Student not found!"

Parameters: ids_list , names_list

Returns:Nothing

═══

FUNCTION 6: delete_student(ids_list, names_list)

What it does:

- Asks the user for a student ID to delete

- Finds that student

- Asks for confirmation (yes or no)

- If yes: deletes the student from both lists

- If no: cancels the deletion

- Shows appropriate message

Parameters: ids_list, names_list

Returns: Nothing (but it changes the lists)

═══

FUNCTION 7: save_to_file(ids_list, names_list)

What it does:

- Opens a file called "students.txt"

- Writes all student IDs and Names to the file

- Each student is one line in the format: ID,Name

- Closes the file

- Shows success message

Parameters: ids_list , names_list

Returns: Nothing

═══

What it does:

- Opens the file "students.txt"

 - Reads all student data from the file

- Clears the current lists (removes all existing students)

- Adds all students from the file to the two lists

- Closes the file

- Shows success message

- If file doesn't exist, shows error message

Parameters: ids_list , names_list

Returns: Nothing (but it changes the lists)

═══

FUNCTION 9: main()

What it does:

This is the MAIN function that runs your entire program. It creates the two lists, shows the menu, gets user input, and calls the other functions.

Parameters : None

Returns: Nothing

═══

Important Notes

1- Clear and meaningful variable names should be used.

2- The program must handle invalid inputs gracefully (no crashes).

3- All menus must loop correctly until the user chooses to exit.

4- The project can be completed by a maximum of 3 students per group.

5- Each group member should understand every part of the code.

6- Final submission deadline: 28 / 2 / 2026

7- Any copy-paste or plagiarism is strictly forbidden.

8- The use of Artificial Intelligence tools (including ChatGPT) to generate code or project solutions is strictly prohibited.

9- The project must be submitted as one PDF file only. The PDF must start with the complete Python code added by copy–paste (no screenshots of code). After the code, the PDF must include clear screenshots for all required cases and scenarios (such as Add Student, Delete Student, Search for a student using id). For each screenshot, a short explanation is mandatory, describing what action was taken, what input was entered, and why the system produced that result.

10- only one student from the group must submit the complete project

Best Wishes 😎 ☕

Aya Nabil Alharazin

5 Feb 2026