Student Management System – Python Project (Functions Only, No File Handling)

---

1. Problem Statement

A small college wants to maintain the details of up to **8 students** (ID, name, course, and marks) using a Python program.
The system should allow adding, viewing, searching, updating, and deleting student details using only **functions** (no OOP, no database, no file storage).

---

2. Features & Functionalities

1. **Add Student**
    - Add new student details (ID, Name, Course, Marks).
    - Restrict maximum students to **8 only**.
    - Course should be selected from **CS, ECE, IT, MECH, CIVIL**.
2. **View Students**
    - Display all student records in a tabular format.
3. **Search Student**
    - Search by student ID or name.
4. **Update Student**
    - Modify details like course or marks.
5. **Delete Student**
    - Remove a student record.
6. **Exit Program**
    - End the application gracefully.

---

3. Requirements

- Python 3.8+
- Knowledge of:
    - Lists & dictionaries
    - Functions
    - Loops & conditionals

---

4. Data Structure

Students will be stored in a **list of dictionaries** (maximum 8 records).

Example:

```
students = [
   {"id": 1, "name": "Ravi", "course": "CS", "marks": 85},
   {"id": 2, "name": "Priya", "course": "ECE", "marks": 92}
]
```

---

5. Functions to Implement

- add_student(students) – Add a new student (check limit = 8, validate course).
- view_students(students) – Show all students.
- search_student(students) – Find student by ID or name.
- update_student(students) – Edit student details.
- delete_student(students) – Delete student by ID.
- menu() – Display menu and call functions.

---

6. Menu Example

```
===== Student Management System =====
1. Add Student
2. View Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice:
```

---

7. Step-by-Step Implementation Guide

Step 1 – Setup

- Create a Python file (main.py).
- Create an empty list: students = [].
- Define all required functions with just pass initially.
- Create a loop to display the menu.

Step 2 – Add Student Function

- Input details: ID, Name, Course, Marks.
- Validate course → must be from: CS, ECE, IT, MECH, CIVIL.
- Convert marks into integer.
- Check if student list has less than 8 records.
- Append dictionary to the list.
- Print success message.

Step 3 – View Students Function

- Check if list is empty → print "No students found".
- Else, loop through and display details in tabular format.

Step 4 – Search Student Function

- Ask for ID or Name.
- Loop through list to find.
- If found → display details.
- Else → print "Student not found".

Step 5 – Update Student Function

- Ask for ID.
- If ID exists → ask whether to update Course or Marks.
- Validate course again if updating course.
- Update the value.
- Print success message.

Step 6 – Delete Student Function

- Ask for ID.
- If ID exists → remove from list.
- Print success message.

Step 7 – Exit Option

- Break loop when user enters 6.

Step 8 – Testing

- Add 2–3 students with courses like CS, ECE, IT.
- Try searching, updating, deleting.
- Check limit (only 8 students allowed).
- Verify outputs.

---

8. Evaluation Criteria

- Correct implementation of CRUD operations – 50%
- Proper use of functions (no OOP) – 20%
- Course validation (CS, ECE, IT, MECH, CIVIL only) – 15%
- Code readability & comments – 15%