***STUDENT GRADING SYSTEM***

This project aims to develop a robust and efficient student grading system, crucial for modern educational institutions. By leveraging functions, optional lists, and SQL, the system will streamline the process of managing student grades, calculating overall performance, and generating reports.

The primary motivations for this project include:

* **Automation of Grading:** To reduce manual effort and potential errors associated with traditional grading methods.
* **Improved Data Management:** To create a centralized and organized database for student academic records, ensuring data integrity and easy retrieval.
* **Enhanced Reporting and Analysis:** To provide educators and administrators with quick access to performance data, enabling better insights into student progress and identifying areas for improvement.
* **Scalability and Efficiency:** To build a system that can handle a large volume of student data efficiently and be easily adaptable to future requirements.
* **Practical Application of Skills:** To apply and demonstrate proficiency in programming concepts (functions, lists) and database management (SQL) learned in your CS curriculum.

In essence, this project is designed to create a practical, user-friendly tool that enhances the efficiency and accuracy of student grade management within an educational setting.

**Requirements**

To achieve its objectives, the student grading system will adhere to the following requirements:

* **Student Information Management:** Ability to add, view, update, and delete student records (e.g., student ID, name, course enrollment).
* **Grade Entry:** Functionality to input grades for various assignments, quizzes, and exams for each student.
* **Grade Calculation:** Automatic calculation of overall grades, averages, and potentially weighted averages based on predefined criteria.
* **Report Generation:** Capability to generate basic reports, such as individual student grade reports, class grade summaries, and perhaps pass/fail lists.
* **Data Persistence:** Storage of all student and grade data in an SQL database.
* **User Interface ( GUI):** graphical user interface (GUI) for interaction.

**Benefits :**

The implementation of this student grading system will yield several significant benefits:

* **Time Savings:** Automates tedious manual grading and record-keeping tasks, freeing up valuable time for educators.
* **Increased Accuracy:** Minimizes human error in grade calculation and data entry, leading to more reliable academic records.
* **Improved Data Accessibility:** Provides quick and easy access to student performance data for teachers, administrators, and potentially students themselves (if a student portal is considered for future scope).
* **Enhanced Decision-Making:** Facilitates data-driven insights into student performance trends, allowing for timely interventions and curriculum adjustments.
* **Reduced Paperwork:** Contributes to a more organized and environmentally friendly record-keeping process.
* **Consistency:** Ensures a standardized approach to grading across different courses or departments.

**Feature Scope :**

For the initial phase of this final year project, the student grading system will focus on the following core features:

* **Student Enrollment:** Basic functionality to add new students and assign them to courses.
* **Assignment/Exam Grade Input:** Manual entry of scores for individual assignments, quizzes, and exams.
* **Weighted Average Calculation:** Calculation of overall course grades based on pre-defined weights for different assessment types.
* **Simple Grade Display:** Ability to view a student's grades for all their enrolled courses.
* **Basic Class Summary:** Display of average grades for a specific course or assignment.
* **SQL Database Integration:** All data (student information, course details, grades) will be stored and retrieved from an SQL database.

Features :

1. **Student Management:**
   * add\_student(ID, name, courses): Adds a new student.
   * view\_student(ID): Shows a student's details.
   * update\_student(ID, new\_info): Modifies student information.
   * delete\_student(ID): Removes a student.
   * list\_all\_students(): Displays all registered students.
2. **Grade Management:**
   * add\_grade(student\_ID, course\_ID, assignment, score, max\_score): Records a new grade.
   * update\_grade(student\_ID, course\_ID, assignment, new\_score): Changes an existing grade.
   * view\_grades\_for\_student(student\_ID, course\_ID=None): Shows all grades for a student (optional course filter).
3. **Grade Calculation:**
   * calculate\_course\_grade(student\_ID, course\_ID): Computes a student's overall grade for a course (includes weighted averages).
   * calculate\_class\_average(course\_ID, assignment=None): Determines the average score for a class
4. **Reporting:**
   * generate\_student\_report(student\_ID): Creates a full report for one student.
   * generate\_pass\_fail\_list(course\_ID, threshold): Lists students who passed or failed a course.
5. **Database Interaction (Internal):**
   * connect\_db(): Connects to the SQL database.
   * create\_tables(): Sets up database tables.
   * execute\_query(sql, params): Runs SQL commands (insert, update, delete).
   * fetch\_data(sql, params): Retrieves data from the database (select).

USING :

* Python
* SQl
* GUI- using python In build function called as tkinter