Project Report

ProSensia

19th August—2024

Relative Grading GPA Calculator Using CSV file

—

Abdul Moiz Arsalan (Team Leader)

Sibgha Manzoor

Muzammil Khan

Proposal: Student GPA Calculation Program

Objective:

To create a Python program that reads a CSV file containing student names, subject names, grades, and credit hours. The program will calculate the Grade Point Average (GPA) for each student based on their grades and credit hours, output the results, and save the updated GPAs back into the original CSV file.

Key Features:

1.**CSV File Handling:**

- Read the input CSV file containing student data.
- Write the calculated GPAs back to the same CSV file.

2. GPA Calculation:

- Convert letter grades to grade points.
- Calculate the GPA for each student by considering the grade points and credit hours for each subject.

3. Output:

- Display the GPA for each student.
- Save the calculated GPAs in the input CSV file.

4. User-Friendly:

- Clear and concise code with comments for easy understanding.
- Written in Python and executed in Visual Studio Code (VS Code).

ProSensia

Tools & Technologies:

- **Programming Language:** Python
- **IDE:** Visual Studio Code (VS Code)
- **Excel:** To read CSV files

Libraries:

- `csv` for handling CSV file operations.

Implementation Steps:

1. Input File Handling:

- Read the CSV file and extract student data.

2. GPA Calculation:

- Implement a function to convert grades to grade points.
- Calculate the GPA for each student using a weighted average of grade points based on credit hours.

3. Output & File Update:

- Display each student's GPA.
- Write the calculated GPAs back into the original CSV file.

Project Plan:

Day 1: Requirements Gathering and Initial Setup

Tasks:

1. **Understand Requirements**:
    - o Clarify the project's scope and deliverables.
    - o Identify the structure of the input CSV file.
    - o Define the grading scale and the formula for GPA calculation.
2. **Setup Development Environment**:
    - o Ensure Python and required libraries (e.g., csv) are installed.
    - o Set up a project directory structure.
3. **Plan Program Structure**:
    - o Outline the key functions: get_grade_point, calculate_gpa, process_csv, and main.
    - o Decide on the input format and output expectations.

Day 2: Core Functionality Development

Tasks:

1. **Develop Core Functions:**
    - o Implement get_grade_point to convert letter grades to grade points.
    - o Implement calculate_gpa to compute the GPA based on grades and credit hours.
2. **File Processing**:
    - o Implement process_csv to read the input CSV file, calculate GPAs, and append the results to each student's record.
    - o Handle any potential errors, such as missing or incorrect data.

Day 3: Testing and Validation

Tasks:

1. **Test Functionality**:
    - o Test the program with various CSV files to ensure accuracy in GPA calculations.
    - o Check edge cases, such as empty grades, incorrect input formats, and handling of file read/write errors.
2. **User Interaction**:
    - o Implement the main function to interact with the user, allowing them to input the CSV file name and manage the flow of the program.

Pr@Sensia

3. **Validation**:
    - Validate that the GPA calculations are correct and that the program is robust against various input scenarios.

Day 4: Finalization and Documentation

Tasks:

1. **Finalize the Program**:
    - Make any necessary adjustments based on testing feedback.
    - Ensure the program runs smoothly and outputs the correct data format.
2. **Documentation**:
    - Write clear documentation explaining how the program works, including instructions on how to use it.
    - Comment the code for readability and maintainability.
3. **Review and Delivery**:
    - Review the entire project to ensure all requirements are met.
    - Prepare the project for submission, including packaging the code and documentation.

Pr@Sensia

Expected Outcome:

A fully functional Python program that efficiently reads student data from a CSV file, calculates GPAs based on grades and credit hours, and updates the original CSV file with the calculated GPAs.

Note: the program can read and calculate multiple CSV files in one Run
This proposal outlines a clear plan to develop a simple yet effective GPA calculator that will assist in automating the process of GPA calculation and management for student records.