**Bantang, Shaira Mae C.** **BSCS-1B**

# Documentation: Grade Tracker

## 1. Description of the Study

The Grade Tracker is like a digital assistant designed specifically for college students to help them stay on top of their grades. It's a handy tool that allows students to keep track of their performance in different classes by easily recording and organizing their grades. With this tool, students can say goodbye to messy spreadsheets or notebooks and effortlessly manage their academic progress in one place.

## 2. Benefits to the User

- **Simplified Grade Management:** No more juggling between multiple notebooks or Excel sheets. The Grade Tracker provides a simple and organized way to manage grades for each class, making it easier to stay organized and focused on studying.
- **Instant GPA Calculations:** Wondering what your GPA is? With just a click of a button, the Grade Tracker instantly calculates your GPA based on the grades you've entered, saving you time and mental math headaches.
- **Insights into Progress:** Seeing your grades over time can provide valuable insights into your academic performance. The Grade Tracker's visualization feature allows you to track your progress and identify areas where you might need to put in extra effort.
- **Quick Reset:** Made a mistake or want to start fresh for a new semester? The Grade Tracker lets you delete all your grades with a single click, giving you a clean slate to begin again.

## 3. Objective

The Grade Tracker project aims to provide college students with a user-friendly tool to streamline the process of managing their grades. By offering features such as easy grade input, GPA calculation, and visual progress tracking, the objective is to empower students to take control of their academic journey and achieve their goals with confidence.

## 4. Purpose of the Code

The purpose of the Grade Tracker code is to create a practical solution to a common challenge faced by college students: staying organized and informed about their grades. By using Python and Tkinter, the code develops a user-friendly interface that simplifies grade management, making it accessible and intuitive for students of all levels.

## 5. Significance of the Code / Features of the Code

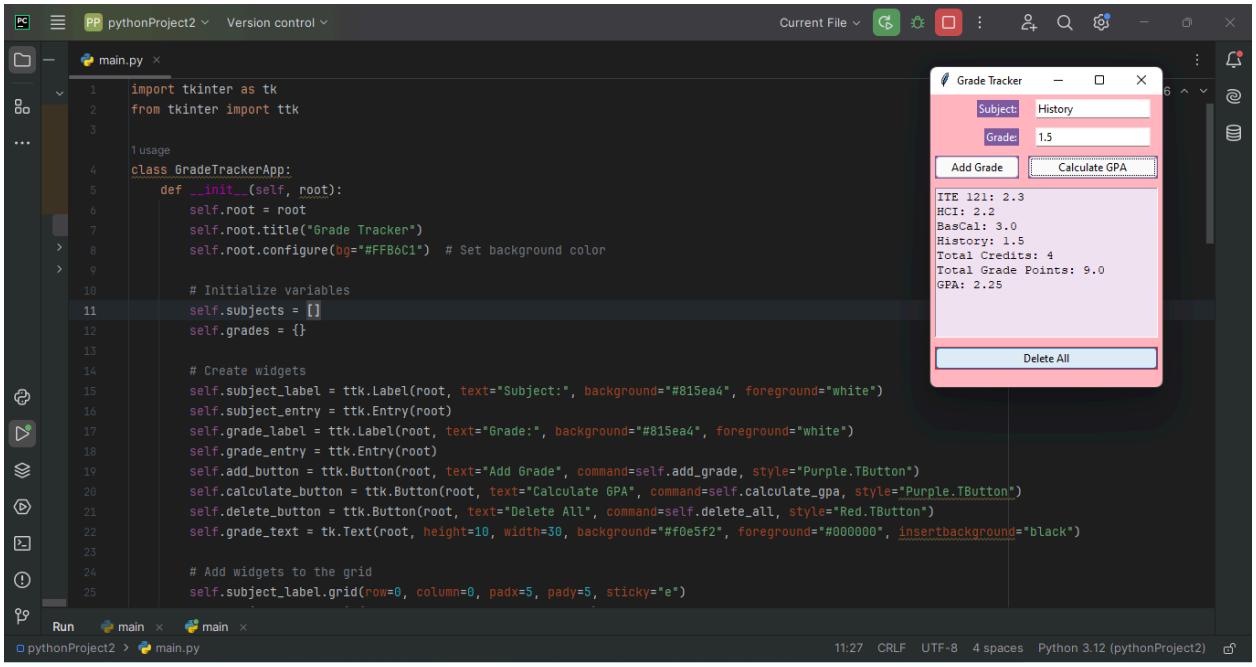
- **Grade Input:** Users can input grades for different subjects or courses, allowing them to keep track of their performance across multiple academic areas.
- **GPA Calculation:** The code calculates the GPA based on the entered grades, providing users with a standardized measure of their overall academic achievement.
- **Grade Visualization:** Users can visualize their grades and GPA trends over time, helping them identify patterns, set goals, and make informed decisions about their academic progress.
- **User-Friendly Interface:** The application features a user-friendly interface with intuitive controls and clear instructions, making it accessible and easy to use for students of all levels.
- **Data Management:** The code includes functions to add, view, and delete grades, ensuring efficient data management and organization for users.

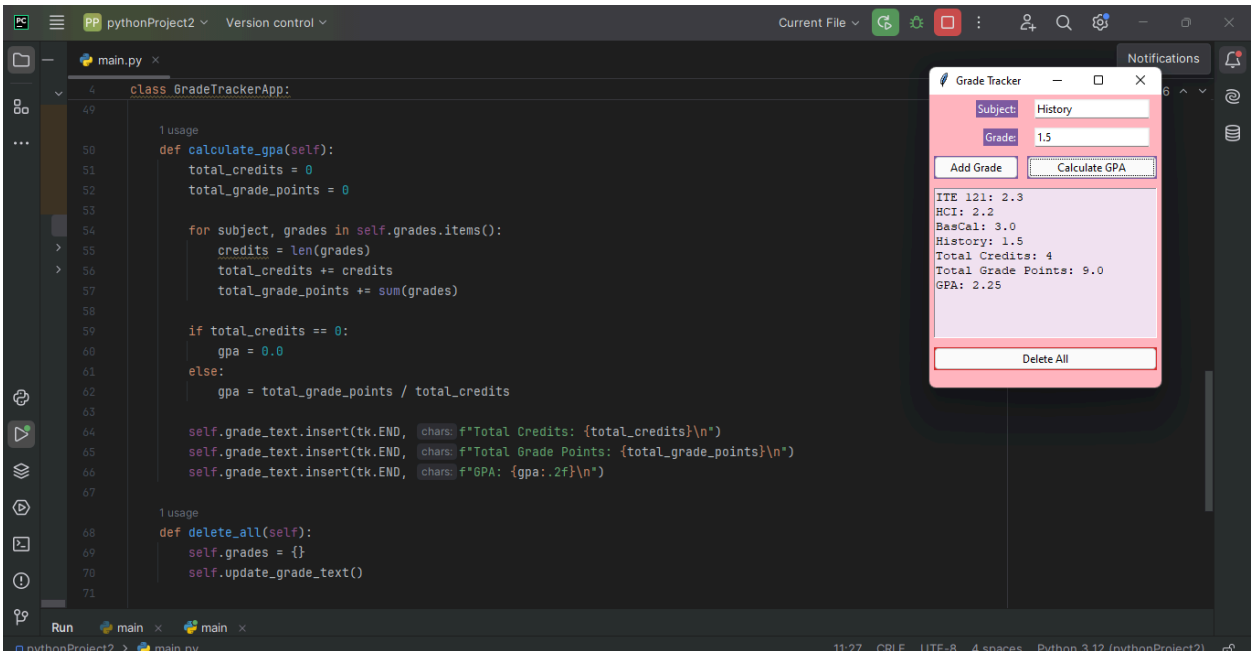
## 6. Code

```
import tkinter as tk
from tkinter import ttk

class GradeTrackerApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Grade Tracker")
        self.root.configure(bg="#FFB6C1")  # Set background color

        # Initialize variables
        self.subjects = []
        self.grades = {}

        # Create widgets
        self.subject_label = ttk.Label(root, text="Subject:",
background="#815ea4", foreground="white")
        self.subject_entry = ttk.Entry(root)
        self.grade_label = ttk.Label(root, text="Grade:", background="#815ea4",
foreground="white")
        self.grade_entry = ttk.Entry(root)
        self.add_button = ttk.Button(root, text="Add Grade",
command=self.add_grade, style="Purple.TButton")
        self.calculate_button = ttk.Button(root, text="Calculate GPA",
command=self.calculate_gpa, style="Purple.TButton")
        self.delete_button = ttk.Button(root, text="Delete All",
command=self.delete_all, style="Red.TButton")
        self.grade_text = tk.Text(root, height=10, width=30,
background="#f0e5f2", foreground="#000000", insertbackground="black")

        # Add widgets to the grid
        self.subject_label.grid(row=0, column=0, padx=5, pady=5, sticky="e")
        self.subject_entry.grid(row=0, column=1, padx=5, pady=5)
        self.grade_label.grid(row=1, column=0, padx=5, pady=5, sticky="e")
        self.grade_entry.grid(row=1, column=1, padx=5, pady=5)
        self.add_button.grid(row=2, column=0, padx=5, pady=5, sticky="ew")
        self.calculate_button.grid(row=2, column=1, padx=5, pady=5,
sticky="ew")
```

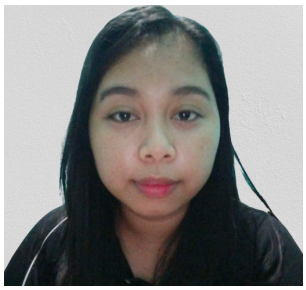
```
        self.delete_button.grid(row=5, column=0, columnspan=2, padx=5, pady=5,
sticky="ew")
        self.grade_text.grid(row=3, column=0, columnspan=2, padx=5, pady=5,
sticky="nsew")

        # Create a style for buttons
        self.root.style = ttk.Style()
        self.root.style.configure("Purple.TButton", foreground="black",
background="#815ea4")
        self.root.style.configure("Red.TButton", foreground="black",
background="#d32f2f")

    def add_grade(self):
        subject = self.subject_entry.get()
        grade = float(self.grade_entry.get())

        if subject in self.grades:
            self.grades[subject].append(grade)
        else:
            self.grades[subject] = [grade]

        self.update_grade_text()

    def calculate_gpa(self):
        total_credits = 0
        total_grade_points = 0

        for subject, grades in self.grades.items():
            credits = len(grades)
            total_credits += credits
            total_grade_points += sum(grades)

        if total_credits == 0:
            gpa = 0.0
        else:
            gpa = total_grade_points / total_credits

        self.grade_text.insert(tk.END, f"Total Credits: {total_credits}\n")
        self.grade_text.insert(tk.END, f"Total Grade Points:
{total_grade_points}\n")
        self.grade_text.insert(tk.END, f"GPA: {gpa:.2f}\n")

    def delete_all(self):
        self.grades = {}
        self.update_grade_text()

    def update_grade_text(self):
        self.grade_text.delete(1.0, tk.END)

        for subject, grades in self.grades.items():
            self.grade_text.insert(tk.END, f"{subject}: ")
            self.grade_text.insert(tk.END, ", ".join(map(str, grades)) + "\n")

if __name__ == "__main__":
    root = tk.Tk()
    app = GradeTrackerApp(root)
    root.mainloop()
```

## 7. Screenshots




## 8. BIO DATA



**PERSONAL INFORMATION**

Name: Shaira Mae C. Bantang

Contact number: 09945090856

Email Address: bantangshairamae@gmail.com

Date of Birth: January 28, 1998

Place of Birth: Surigao City

Address: 00169 Rizal Street, Surigao City, Surigao del Norte, Caraga, Philippines

Age: 26

Nationality: Filipino

Religion: pentocost

Civil status: Single

Father's Name: Sancholito L. Bantang

Mother's Name: Jesusa C. Bantang

**EDUCATIONAL BACKGROUND**

- **Elementary**: C.V. Diez Memorial Central Elementary School, Surigao del Norte

- **High School**: Surigao Del Norte National High School

- **College**: Surigao Norte State University