Student Grade Tracker

A full-stack academic performance tracking application designed for teachers and students. Built using Python (Flask + Gradio + SQLite), this tool allows real-time grade management, GPA calculation, data visualization, and CSV export — all running in the cloud via Google Colab.

Features

Add, update, delete student grades

GPA and percentage calculation

Grade visualization (bar charts)

CSV export

No login required (demo mode)

Accessible dark-mode UI with Gradio

File Structure

├── app.py # Flask API backend ├── ui.py # Gradio frontend interface ├── database.py # SQLite CRUD operations ├── final_production_cleaned.py # Backup version for final deployment ├── requirements.txt # Python dependencies

Getting Started (Google Colab Setup)

This application is designed for cloud execution via Google Colab.

1. Upload Files to Colab

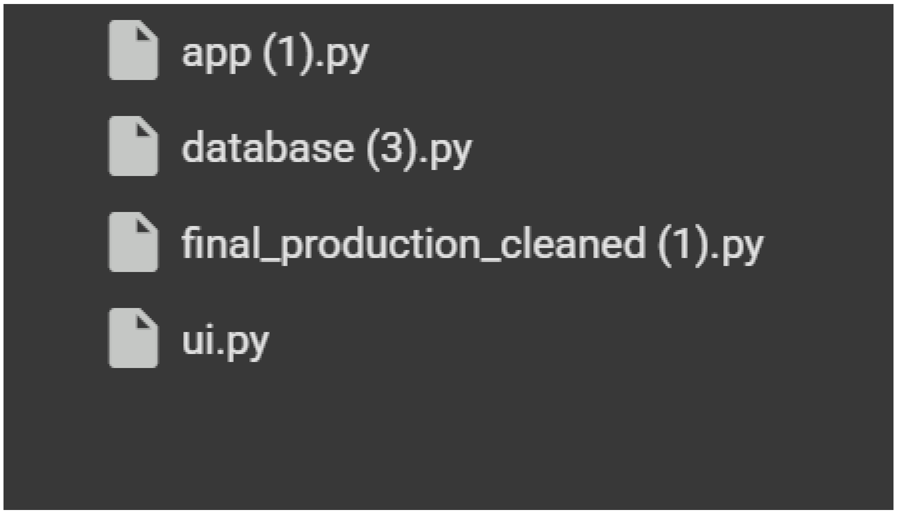
Upload the following .py files manually:

app.py

database.py

ui.py

(optionally) final_production_cleaned.py



app (1).py

database (3).py

final_production_cleaned (1).py

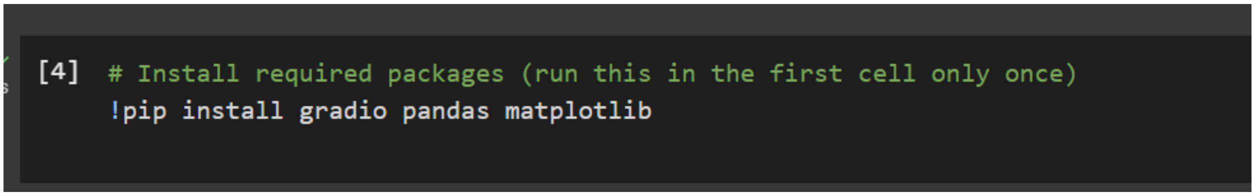
ui.py

PAST IN COLLAB NOTEBOOK:

RUN THIS CELL

Install required packages (run this in the first cell only once)

!pip install gradio pandas matplotlib

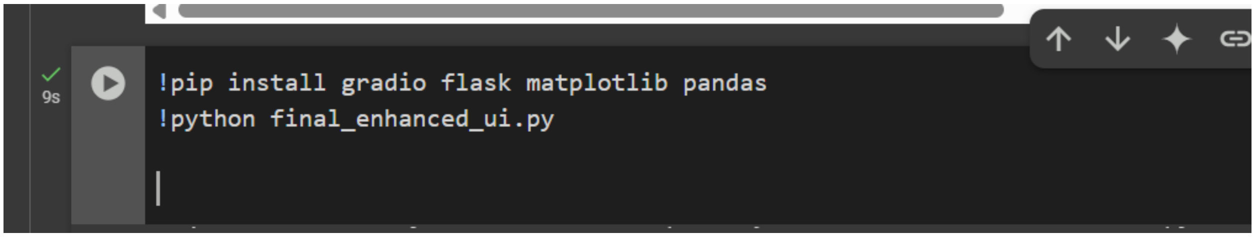


```
[4]  # Install required packages (run this in the first cell only once)
     !pip install gradio pandas matplotlib
```

THEN RUN THIS CELL:

!pip install gradio flask matplotlib pandas

!python final_enhanced_ui.py



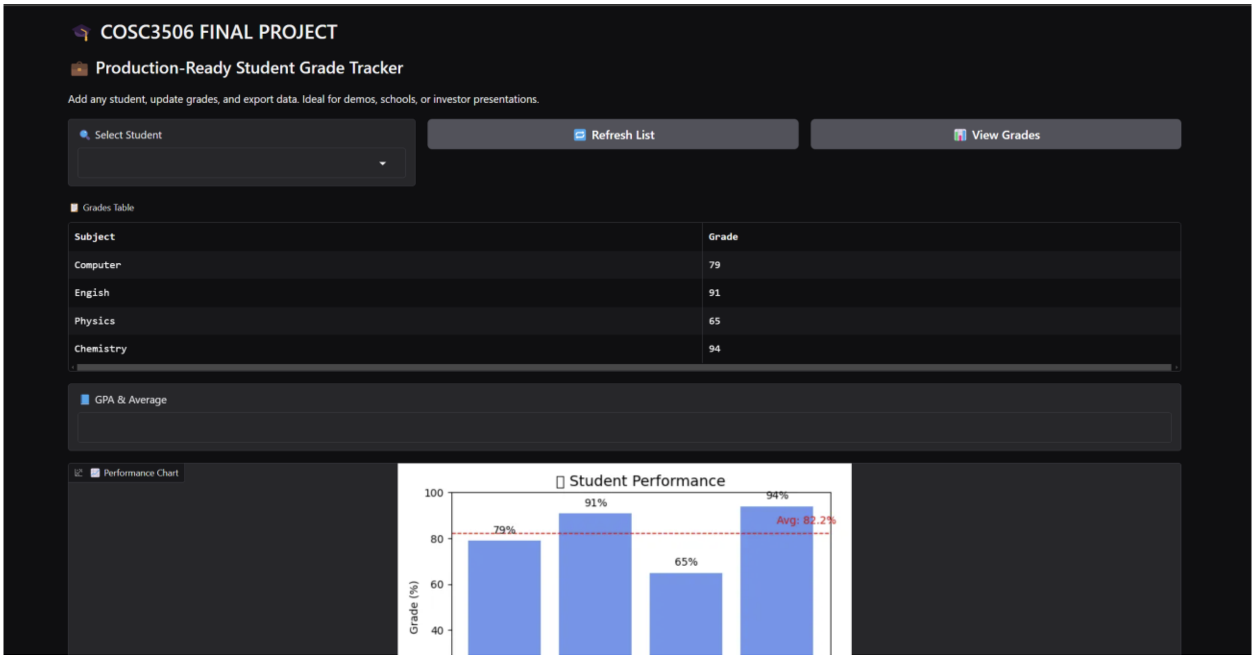
```
!pip install gradio flask matplotlib pandas
!python final_enhanced_ui.py
```

THE PAST CODE IN NEXT CELL AND RUN:

import gradio as gr

```
import pandas as pd
import matplotlib.pyplot as plt
from database import init_db, get_grades, add_grade, delete_grade

# Reinitialize or create database
init_db()

# Helper: Get all unique students from DB
def get_all_students():
    records = get_grades()
    students = sorted(set([r[0] for r in records]))
    return students if students else ["alice", "bob"]

# Plot bar chart for student grades
def plot_grades(grades_list):
    subjects = [subj for (subj, _) in grades_list]
    scores = [float(s) for (_, s) in grades_list]
    fig = plt.figure(figsize=(6,4))
    ax = fig.add_subplot(1,1,1)
    if not scores:
        ax.text(0.3, 0.5, "No grades to display", fontsize=12)
        ax.axis('off')
        return fig
    bars = ax.bar(subjects, scores, color='cornflowerblue')
    ax.set_ylim(0, 100)
    ax.set_title("📊 Student Performance")
    ax.set_xlabel("Subjects")
    ax.set_ylabel("Grade (%)")
    for bar in bars:
        height = bar.get_height()
        ax.text(bar.get_x() + bar.get_width()/2, height + 2, f'{height:.0f}%', ha='center')
    avg = sum(scores) / len(scores)
    ax.axhline(avg, color='red', linestyle='--', linewidth=1)
    ax.text(len(subjects)-1, avg + 4, f"Avg: {avg:.1f}%", color='red')
    return fig

# Fetch student data from DB
def get_student_data(student):
    records = get_grades(student)
    df = pd.DataFrame(records, columns=["Subject", "Grade"])
    if records:
        grades = [float(g) for (_, g) in records]
        avg = sum(grades) / len(grades)
        gpa = (avg / 100) * 4.0
        stats = f"GPA: {gpa:.2f} | Average: {avg:.1f}%"
        chart = plot_grades(records)
    else:
        stats = "No data available."
        chart = plot_grades([])
    csv_path = f"{student}_grades.csv"
    df["Student"] = student
    df = df[["Student", "Subject", "Grade"]]
    df.to_csv(csv_path, index=False)
    return df, stats, chart, csv_path

# Add grade
def add_student_grade(student, subject, score):
    try:
        msg = add_grade(student.strip().lower(), subject.strip(), float(score))
        return msg
    except Exception as e:
        return f"❌ Error: {str(e)}"

# Delete grade
def delete_student_grade(student, subject):
    try:
        msg = delete_grade(student.strip().lower(), subject.strip())
        return msg
    except Exception as e:
        return f"❌ Error: {str(e)}"

# 🖥️ Gradio UI
with gr.Blocks(title="COSC3506 FINAL PROJECT - Student Grade Tracker") as demo:
    gr.Markdown("""
    # 🎓 COSC3506 FINAL PROJECT
    ## 💼 Student Grade Tracker (Demo Mode)
    Add students, visualize grades, and export reports.
    """)

    with gr.Row():
        student_input = gr.Dropdown(label="Select Student", choices=get_all_students())
        refresh_students = gr.Button("🔁 Refresh")
        view_button = gr.Button("📊 View Grades")

    grades_table = gr.Dataframe(label="Grades Table")
    grade_stats = gr.Textbox(label="GPA & Average", interactive=False)
    grade_chart = gr.Plot(label="Performance Chart")
    export_button = gr.File(label="Export CSV")

    with gr.Row():
        with gr.Column():
            gr.Markdown("### ➕ Add / Update Grade")
            add_student = gr.Textbox(label="Student", placeholder="e.g. sara")
            add_subject = gr.Textbox(label="Subject", placeholder="e.g. Chemistry")
            grade_slider = gr.Slider(0, 100, step=1, label="Grade")
            add_button = gr.Button("✅ Add / Update")
            add_status = gr.Textbox(label="Status", interactive=False)

        with gr.Column():
            gr.Markdown("### ❌ Delete Grade")
            del_student = gr.Textbox(label="Student")
            del_subject = gr.Textbox(label="Subject")
        del_button = gr.Button("🗑️ Delete")
        del_status = gr.Textbox(label="Status", interactive=False)

    # Actions
    view_button.click(get_student_data, inputs=student_input, outputs=[grades_table, grade_stats, grade_chart, export_button])
    add_button.click(add_student_grade, inputs=[add_student, add_subject, grade_slider], outputs=add_status)
    del_button.click(delete_student_grade, inputs=[del_student, del_subject], outputs=del_status)
    refresh_students.click(lambda: gr.update(choices=get_all_students()), None, student_input)

# 🔗 Launch the app
demo.launch(share=True)
```

A public Gradio URL will appear.(FOR EASY I HAVE MODIFIED IT SO YOU CAN DIERECTLY VIEW IN COLLAB) Open it in a new tab.(SCREENSHOTS ATTACHED IN OTHER DOCUMENT HOW IT WILL LOOK IF YOU OPEN IT SEPRATELY)

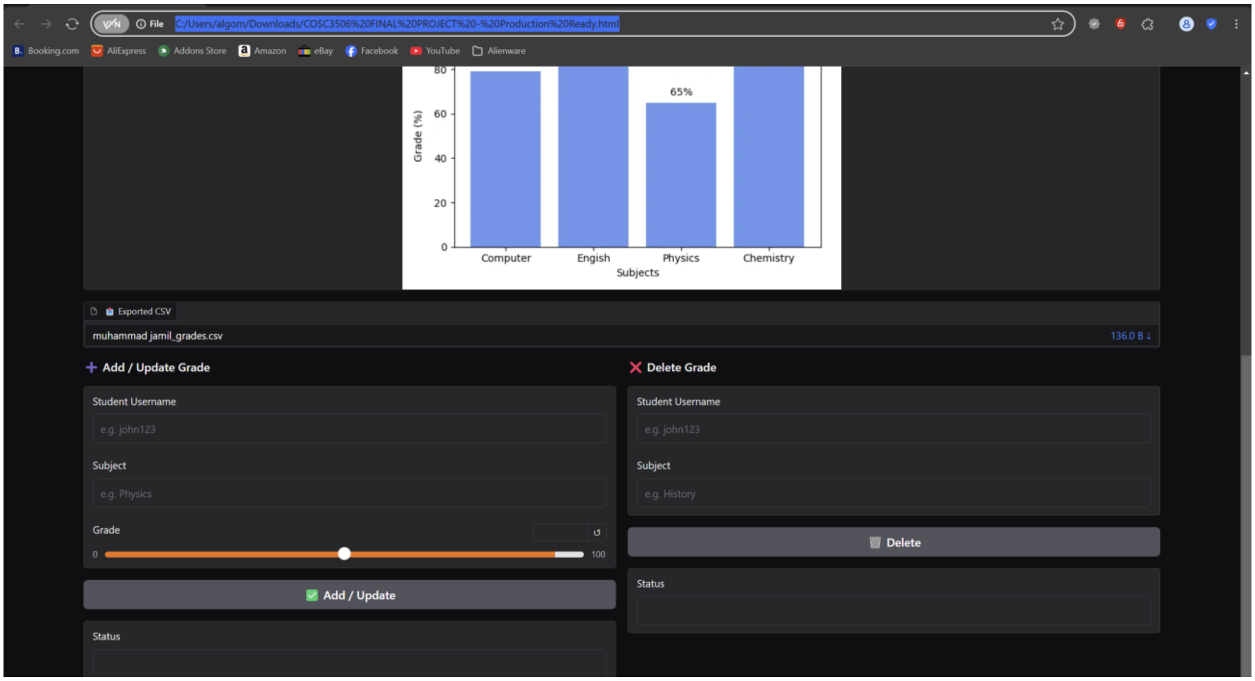


COSC3506 FINAL PROJECT

Production-Ready Student Grade Tracker

Add any student, update grades, and export data. Ideal for demos, schools, or investor presentations.

Subject	Grade
Computer	79
Engish	91
Physics	65
Chemistry	94



C:/Users/algom/Downloads/COSC3506%20FINAL%20PROJECT%20-%20Production%20Ready.html

Usage

Select or enter a student name.

Enter or select subject.

Use slider to input grade.

Click Add/Update to save it.

Click View Grades to fetch all data and visualization.

Use Download CSV to export grade report.

Developer

Name: Muhammad JamilCourse: COSC3506 – Software EngineeringInstitution: Algoma University

Repository Link

https://github.com/Jamie8788/MuhammadJamil_student-grade-tracker

License