



COSC3506 USER DOCUMENTATION MUHAMMAD JAMIL



Muhammad Jamil

1. Introduction

Welcome to the **Student Grade Tracker**! This application allows teachers to manage and track student academic performance efficiently. It is designed to run on **Google Colab** and features a clean, user-friendly interface powered by **Gradio**, with real-time GPA calculation, chart visualization, and CSV exporting.

2. Getting Started

To begin using the Student Grade Tracker:

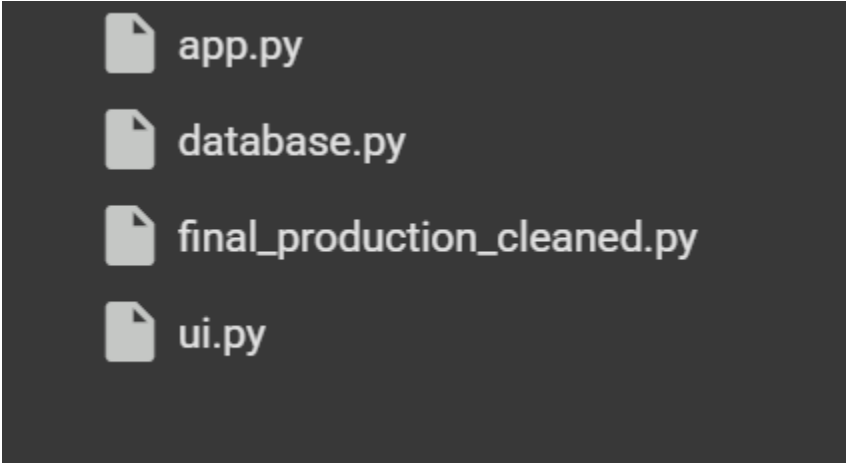
1. **Open the Colab notebook** provided with the project.

<https://colab.research.google.com/drive/1xJERZM-ttzh8GopbLvzbUwFS2iLGf2Dd#scrollTo=4fPKdc7Sf3TU>

2. Run the cells to initialize the database and launch the Gradio UI.

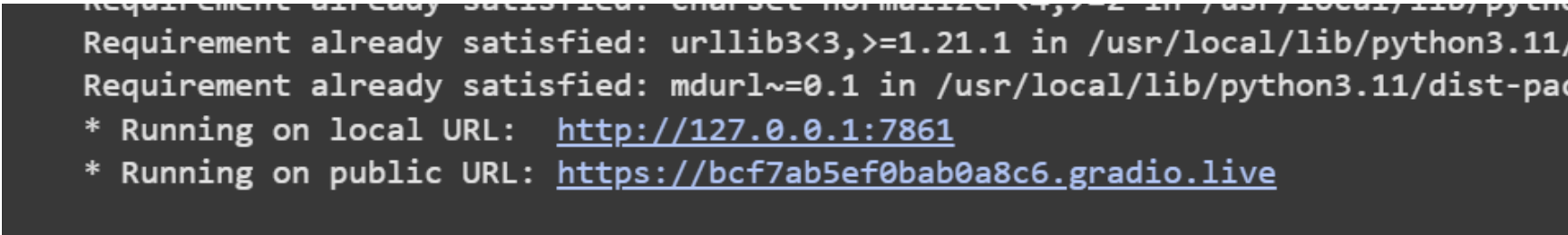
```
[ ] !pip install gradio flask matplotlib pandas
```

DOWNLOAD AND UPLOAD THESE FILES TO COLLAB NOTEBOOK(PROVIDED SEPRATELY)



- app.py
- database.py
- final_production_cleaned.py
- ui.py

3. A shareable public link (via Gradio) will appear for interacting with the app.



```
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/  
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-pac  
* Running on local URL: http://127.0.0.1:7861  
* Running on public URL: https://bcf7ab5ef0bab0a8c6.gradio.live
```

No installation is required. The software runs entirely in the cloud.

3. Application Interface Overview

The app consists of the following sections:

3.1 View Student Grades

- Select a student from the dropdown.
- Click "**View Grades**".
- You will see:
 - A table listing subject-wise grades.
 - A calculated **GPA** and **percentage average**.
 - A bar graph showing grade distribution.

Production-Ready Student Grade Tracker

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

Select Student

muhammad jamil

alice

bob

hassan

✓ muhammad jamil

English

Physics

Chemistry

Refresh List

View Grades

Grade

79

91

65

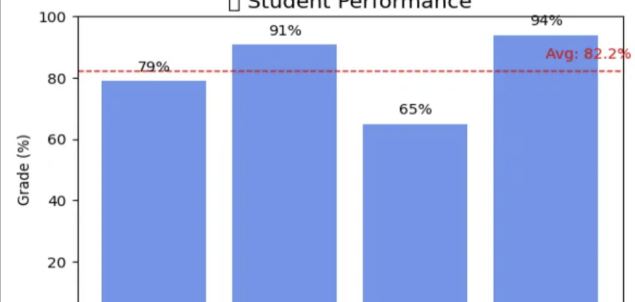
94

GPA & Average

GPA: 3.29 | Average: 82.2%

Performance Chart

Student Performance



3.2 Add / Update Grade (Teacher/Admin Role)

- Enter **Student Username** (e.g., alice, bob, sara).
- Enter or select **Subject**.
- Use the **slider** to input the grade (0 to 100).
- Click "**Add / Update**" to store the grade.
- A success or error message will confirm the operation.

Exported CSV

muhammad_jamil_grades.csv

+ Add / Update Grade

Student Username

Muhammad Jamil

Subject

Chemistry

Grade

94

↺

0

100

✓ Add / Update

Status

Grade updated.

✗ Delete Grade

Student Username

e.g. john123

Subject

e.g. History

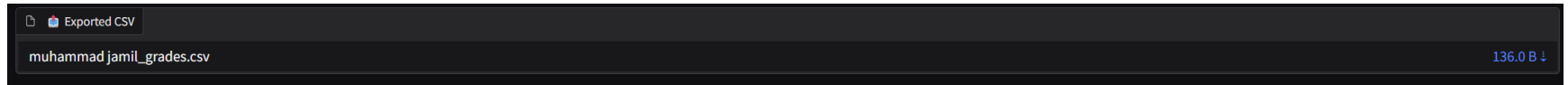
Status

Status

Grade updated.

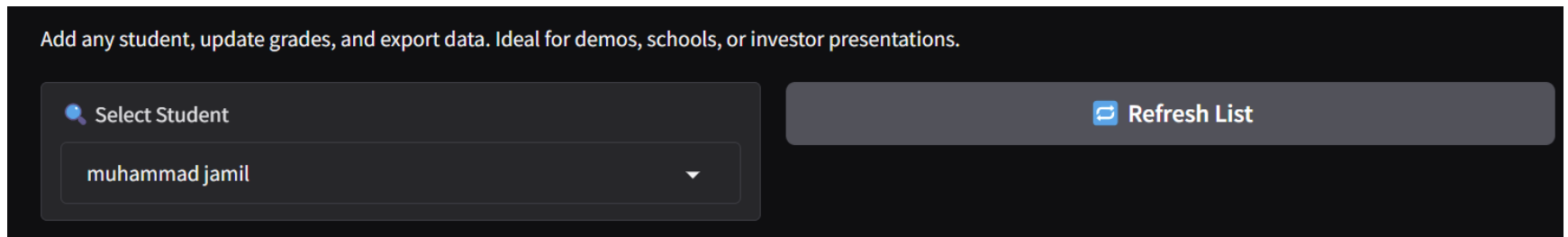
3.3 Export Grades

- After viewing a student's grades, click the **Download CSV** button.
- A .csv file will be generated with all grade data.



3.4 Refresh Grades

- Use the **Refresh** button on the student dashboard to update displayed grades in real time.



4. Features Summary

Feature	Description
GPA Calculation	GPA is computed from percentage grades and displayed in real time.

Add/Update/Delete Grades	Teachers can manage grade entries dynamically.
CSV Export	Grade data can be exported to spreadsheets for reporting.
Chart Visualization	Automatically generated bar chart for each student showing subject grades.
Dark Mode UI	Gradio blocks are styled for accessibility and night-friendly visuals.

5. FAQ / Help

Q: What if I enter a new student name?

A: The system automatically creates a record for the new student.

Q: Can I enter new subjects?

A: Yes, just type in the subject name while adding a grade.

Q: What happens if I re-enter an existing subject and student?

A: The grade will be updated.

Q: How fast does the app respond?

A: All operations (add/view/export) are executed under 2 seconds (non-functional goal).

Q: Do I need to log in?

A: No login is required. This is a demo-ready, public access app.

6. Technical Requirements

- **Platform:** Google Colab (free tier)
- **Languages:** Python
- **Frameworks:** Flask (backend), Gradio (frontend), SQLite, Pandas, Matplotlib
- **No installation required.**

7. Development Codebase

All development was version-controlled using **GitHub**. A public repository was created for tracking collaborative changes. The project uses:

- `app.py`: Backend API with Flask
- `database.py`: SQLite DB logic
- `ui.py`: Gradio-based frontend
- `requirements.txt`: Package dependencies for easy setup

Each update was committed using descriptive messages, and separate branches were used for features like CSV export and UI enhancements. This ensured safe, collaborative iteration.

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