STUDENT_PERFORMANCE_TRACKER

I’ve built the full project (CLI + Flask web app), wired to SQLite with clean OOP design, input validation, and a menu loop. It’s packaged and ready to run or deploy.

How to run

CLI app

1. Unzip → cd student_performance_tracker/cli_app

2. Run: python3 main.py

Options: add student, add grades, view details, calculate average, list students.

Data persists in students.db.

Web app (Flask)

1. cd student_performance_tracker/web_app

```
python3 -m venv .venv
. .venv/bin/activate
```

Windows (PowerShell):

```
python -m venv .venv
.venv\Scripts\Activate.ps1
```

3. Install deps: pip install -r requirements.txt

4. Run locally: python app.py → open http://localhost:5000

Add students, add subject-wise grades, view student page with average.

Deploy (quick start)

The web_app has Procfile and requirements.txt. For Heroku-like platforms:

```
cd student_performance_tracker/web_app
git remote add origin https://github.com/bharathbharath12102004-coder/student_performance_tracker.git
git init
git add .
git commit -m "initial deploy"
git push origin Main
```

What’s included

Object-Oriented core: Student (name, roll_number, grades dict) and StudentTracker (add_student, add_grades, view_student_details, calculate_average, list_students).

Validation: grade range 0–100; duplicate roll numbers blocked; graceful “not found” handling.

SQLite persistence: students & grades tables with constraints.

Menu-driven CLI: continuous loop for all operations.

Flask UI: pages to list students, add student, view student details with average, and add multiple grades at once.

Styling: simple responsive CSS.