Assignment-2C: Building a Student Performance Tracker REST API using FastAPI and Postgres

---

🎯 Objective

Convert the existing **console-based Student Performance Tracker (Assignment-2A)** into a fully functional **REST API** using **FastAPI** and integrate with initially with SQLite and then to Postgres by:

- Reusing and organizing existing modules
- Defining routes using FastAPI decorators
- Using Pydantic models for input/output
- Handling validation, response messages, and error cases
- Connecting to in-memory or file-based DB like SQLite (extendable to real DBs like PostgreSQL later)

Expected Endpoints (REST API)

Student APIs (Mandatory)

Method	Endpoint	Description
POST	/students/	Add a new student
GET	/students/	List all students
GET	/students/{student_id}	Get student details by ID
DELETE	/students/{student_id}	Remove student by ID
GET	/students/search/	Search students by name

Scores APIs (Optional)

Method	Endpoint	Description
POST	/students/{student_id}/scores/	Add/update subject score for a student
GET	/students/{student_id}/average-score/	Get average score for a student

Method	Endpoint	Description
GET	/students/top-scorer/{subject}	Get top scorer in a given subject
GET	/departments/{department}/average-score/	Get average score for a department

---

Testing with Postman/Bruno or Swagger UI

- Launch app: uvicorn main:app –reload/ fastapi dev main.py
- Visit docs: http://localhost:8000/docs

📮 **Submission Guidelines**

1. GitHub repo with:
    - All source code under proper folder structure
    - Postman collection (if used)
    - Sample curl/Postman calls in README.md
2. Include screenshots or curl examples for:
    - Adding student
    - Adding score
    - Listing all students
    - Top scorer and department average endpoints

Assessment Criteria

Area	Criteria
Completeness	All expected endpoints implemented and tested
Code Structure	Modular code with FastAPI conventions (routers/, schemas/, etc.)
Validation	Input validation using Pydantic models
OOP Use	Student class logic reused cleanly in API

Area	Criteria
Extendability	Code easily extendable to use real DB (e.g. Postgres)
Documentation	README with run instructions, screenshots or Postman calls

---

Optional Extensions (Bonus Exercise)

- Add pagination to GET /students/
- Add basic JWT authentication for protected routes
