College Management System FastAPI Exercises

Objective: Build a FastAPI application using 3 JSON files: `students.json`, `courses.json`, `enrollments.json`. Implement CRUD operations and additional custom routes for sorting and filtering.

JSON Files:

1. students.json [{"id": 1, "name": "Alice", "age": 20, "gpa": 3.5, "department": "CS"}, {"id": 2, "name": "Bob", "age": 22, "gpa": 3.8, "department": "EE"}, {"id": 3, "name": "Charlie", "age": 21, "gpa": 3.2, "department": "ME"}]

2. courses.json [{"id": 1, "name": "Data Structures", "credits": 4, "department": "CS"}, {"id": 2, "name": "Circuits", "credits": 3, "department": "EE"}, {"id": 3, "name": "Thermodynamics", "credits": 3, "department": "ME"}]

3. enrollments.json [{"id": 1, "student_id": 1, "course_id": 1, "grade": "A"}, {"id": 2, "student_id": 2, "course_id": 2, "grade": "B"}, {"id": 3, "student_id": 3, "course_id": 3, "grade": "C"}, {"id": 4, "student_id": 1, "course_id": 2, "grade": "B"}]

Exercises / Routes:

Students Router: 1. GET `/students` → Get all students. 2. GET `/students/{id}` → Get student by ID. 3. POST `/students` → Add a new student. 4. DELETE `/students/{id}` → Delete student by ID. 5. GET `/students/sort/gpa` → Return students sorted by GPA descending. 6. GET `/students/department/{department}` → Get students by department. 7. GET `/students/top/{n}` → Get top `n` students by GPA. 8. GET `/students/average/gpa` → Return average GPA of all students.

Courses Router: 1. GET `/courses` → Get all courses. 2. GET `/courses/{id}` → Get course by ID. 3. POST `/courses` → Add a new course. 4. DELETE `/courses/{id}` → Delete course by ID. 5. GET `/courses/department/{department}` → Get courses by department. 6. GET `/courses/sort/credits` → Sort courses by credits. 7. GET `/courses/student-count` → Return courses with number of students enrolled.

Enrollments Router: 1. GET `/enrollments` → Get all enrollments. 2. GET `/enrollments/{id}` → Get enrollment by ID. 3. POST `/enrollments` → Add new enrollment (validate `student_id` and `course_id`). 4. DELETE `/enrollments/{id}` → Delete enrollment by ID. 5. GET `/enrollments/student/{student_id}` → Get all courses a student is enrolled in. 6. GET `/enrollments/course/{course_id}` → Get all students enrolled in a course. 7. GET `/enrollments/sort/grade` → Sort enrollments by grade. 8. GET `/enrollments/student/{student_id}/gpa` → Compute GPA for a particular student.

Instructions: - Organize the project using routers. - Use functions to load and save JSON data. - Test sorting, filtering, and CRUD operations using Swagger UI or Postman. - Optional: Add data validation using Pydantic models.