Laravel Backend API Unit Test – Đề Thi Lại

Thời gian: 24 giờ

***Mục tiêu**: Học viên vận dụng kiến thức đã học để xây dựng ứng dụng Laravel API backend. Qua đó đánh giá năng lực của học viên trong việc phát triển API, test API bằng Postman, viết test function cho tất cả các chức năng trong controller bằng PHPUnit, và thể hiện các trường hợp đã test dưới dạng tài liệu.*

Problem Statement

Phát triển một backend API cho **Hotel Management System**. Hệ thống quản lý phòng và khách đặt phòng. API backend phải bao gồm các endpoint để thực hiện các thao tác CRUD (tạo, đọc, cập nhật, xóa) cho cả phòng và khách đặt phòng. Ngoài ra, cần triển khai đầy đủ các xác thực dữ liệu (validation) và mối quan hệ giữa các bảng (tables) trong database.

A. PHẦN BÀI KIỂM TRA CỦA HỌC VIÊN

1. Thiết lập dự án Laravel: Tạo một dự án Laravel: **library-api** và MySQL DB.

2. Tạo Models và Migrations

- Tạo các **model** và **migration** cho db:
    - Room: id, room_number, type, price, availability_status, created_at, updated_at.
    - Booking: id, room_id, customer_name, check_in_date, check_out_date, status, created_at, updated_at.
- Quan hệ giữa các tables:
    - Room có nhiều Booking.
    - Booking thuộc về một Room.

3. Controller Functions

- **RoomController:**
    - index(): Liệt kê tất cả các phòng.
    - store(Request $request): Thêm một phòng mới.
    - show(id): Lấy thông tin chi tiết của một phòng cụ thể.

    - update(Request $request, id): Cập nhật thông tin phòng.
    - destroy(id): Xóa một phòng.
- **BookController**
    - index(): Liệt kê tất cả các booking.
    - store(Request $request): Thêm một booking mới.
    - show(id): Lấy thông tin chi tiết của một booking cụ thể.
    - update(Request $request, id): Cập nhật thông tin booking.
    - destroy(id): Xóa một booking.

4. Validation: Thực hiện xác thực cho tất cả các mục bằng function validate (*$validated = $request->validate*).

5. Postman Testing: Sử dụng Postman để kiểm tra tất cả các endpoint của API đã phát triển.

6. PHPUnit Testing: Viết **PHPUnit tests** cho tất cả các functions trong controllers.

7. Viết Test Case: Tạo file **test_cases.docx** với các thông tin sau (liệt kê tất cả các function tests): **Test Name/ Time/ Details/ Status** (Pass/Fail)

8. Thư mục bài kiểm tra phải bao gồm (Zip tất cả 3 file này lại và gửi kèm theo mail**)**

- **Laravel project folder** source code.
- **Xuất ra tất cả các file test trên Postman của từng endpoint dưới dạng file Collection.** (Xem hướng dẫn bên dưới).
- **PHPUnit tests** (test_cases.docx).

B. PHẦN CHẤM ĐIỂM CỦA GIÁO VIÊN:

- Triển khai đúng models, relationships, and migrations (20 điểm).
- Chức năng CRUD đầy đủ cho cả hai controllers (20 điểm).
- Đầy đủ validation và xử lý lỗi (15 điểm).
- Kiểm thử thành công tất cả các endpoint bằng Postman (15 điểm).

- PHPUnit test đầy đủ cho tất cả các controllers (20 điểm).
- Viết tài liệu chi tiết về các trường hợp test trong file test_cases.docx (10 điểm).

C. PHẦN HƯỚNG DẪN XUẤT COLLECTION:

Postman Testing: Sử dụng Postman để kiểm tra tất cả các endpoint của API đã phát triển. Sau khi kiểm tra, xuất ra tất cả các file test trên Postman của từng endpoint dưới dạng file Collection. Để thực hiện điều này, thực hiện các bước sau:

- Tạo một Collection trên Postman và thêm tất cả các request đã sử dụng vào Collection.
- Sau khi test xong, chọn Collection và nhấp vào nút Export để xuất ra file dạng .json.
- Lưu file này trong thư mục dự án để nộp kèm theo bài kiểm tra.