

ĐỀ KIỂM TRA

HỌC PHẦN: LẬP TRÌNH MOBILE ĐỀ SỐ: 05 - XÂY DỰNG WEB API

Thời gian làm bài: 120 phút

Tổng điểm: 100 điểm

Được phép: Sử dụng ChatGPT, tài liệu, Internet

Công nghệ: Tùy chọn (Laravel, Node.js/Express, Django, ASP.NET Core, Spring Boot, v.v.)

MÔ TẢ DỰ ÁN

⚠ YÊU CẦU BẮT BUỘC:

- Tên folder source code project phải là: `web_api_<MaSinhVien>` (Ví dụ: `web_api_2151061234`).
- Tên Cơ sở dữ liệu (Database Name) PHẢI là: `db_exam_<MaSinhVien>` (Ví dụ: `db_exam_2151061234`). Sinh viên phải cấu hình kết nối CSDL đúng với tên này.

Bạn cần xây dựng một RESTful API cho hệ thống quản lý nhà hàng và một ứng dụng Flutter (Client App) cho khách hàng. API cho phép quản lý đặt bàn, thực đơn. App cho phép khách hàng xem menu.

📋 YÊU CẦU CƠ SỞ DỮ LIỆU

Database Schema:

Bảng 1: customers

- `id` (Primary Key, Auto Increment)
- `email` (String, Unique, Not Null)
- `password` (String, Not Null)
- `full_name` (String, Not Null)
- `phone_number` (String, Nullable)
- `address` (String, Nullable)
- `loyalty_points` (Integer, Default: 0)
- `created_at` (Timestamp)
- `updated_at` (Timestamp)
- `is_active` (Boolean, Default: true)

Bảng 2: menu_items

- `id` (Primary Key, Auto Increment)
- `name` (String, Not Null)
- `description` (Text, Nullable)
- `category` (Enum/String, Not Null): "Appetizer", "Main Course", "Dessert", "Beverage", "Soup"
- `price` (Decimal, Not Null)
- `image_url` (String, Nullable)
- `preparation_time` (Integer): Thời gian chế biến (phút)

- `is_vegetarian` (Boolean, Default: false)
- `is_spicy` (Boolean, Default: false)
- `is_available` (Boolean, Default: true)
- `rating` (Decimal, Default: 0.0)
- `created_at` (Timestamp)
- `updated_at` (Timestamp)

Bảng 3: reservations

- `id` (Primary Key, Auto Increment)
- `customer_id` (Foreign Key → customers.id, Not Null)
- `reservation_number` (String, Unique, Not Null): Mã đặt bàn tự sinh
- `reservation_date` (Timestamp, Not Null)
- `number_of_guests` (Integer, Not Null)
- `table_number` (String, Nullable): Số bàn được phân
- `status` (Enum/String, Default: "pending"): "pending", "confirmed", "seated", "completed", "cancelled", "no_show"
- `special_requests` (Text, Nullable)
- `subtotal` (Decimal, Default: 0)
- `service_charge` (Decimal, Default: 0): Phí phục vụ (10% subtotal)
- `discount` (Decimal, Default: 0): Giảm giá từ loyalty points
- `total` (Decimal, Default: 0)
- `payment_method` (String, Nullable): "cash", "card", "online"
- `payment_status` (String, Default: "pending"): "pending", "paid", "refunded"
- `created_at` (Timestamp)
- `updated_at` (Timestamp)

Bảng 4: reservation_items (Bảng trung gian)

- `id` (Primary Key, Auto Increment)
- `reservation_id` (Foreign Key → reservations.id, Not Null)
- `menu_item_id` (Foreign Key → menu_items.id, Not Null)
- `quantity` (Integer, Not Null)
- `price` (Decimal, Not Null): Giá tại thời điểm đặt
- `created_at` (Timestamp)

Bảng 5: tables (Quản lý bàn)

- `id` (Primary Key, Auto Increment)
- `table_number` (String, Unique, Not Null)
- `capacity` (Integer, Not Null): Sức chứa (số người)
- `is_available` (Boolean, Default: true)
- `created_at` (Timestamp)
- `updated_at` (Timestamp)

Ràng buộc:

- Foreign Key: `reservations.customer_id` → `customers.id` (ON DELETE RESTRICT)
- Foreign Key: `reservation_items.reservation_id` → `reservations.id` (ON DELETE CASCADE)

- Foreign Key: `reservation_items.menu_item_id` → `menu_items.id` (ON DELETE RESTRICT)
- Unique: `reservations.reservation_number`
- Unique: `tables.table_number`
- Khi đặt bàn: tạo reservation với status = "pending"
- Khi xác nhận: phân table_number, set status = "confirmed"
- Khi thanh toán: tính loyalty_points (1% total), cộng vào customer, trừ discount đã dùng

YÊU CẦU CHỨC NĂNG

Phần 1: Thiết lập Project và Database (10 điểm)

1. **Khởi tạo Project** (5 điểm)
2. **Tạo Database Migration** (7 điểm) - 5 bảng
3. **Seeder Data** (3 điểm) - 5 customers, 20 menu_items, 10 reservations với items, 8 tables

Phần 2: Authentication & Authorization (20 điểm)

Yêu cầu đặc thù: Trong response của API Đăng nhập ([/login](#)), ngoài token và user info, BẮT BUỘC phải trả về thêm trường `student_id` có giá trị là Mã Sinh Viên của bạn (hardcode).

2.1. Đăng ký Customer (5 điểm)

Endpoint: [POST /api/auth/register](#)

Request Body:

```
{  
    "email": "customer@example.com",  
    "password": "password123",  
    "full_name": "Nguyễn Văn A",  
    "phone_number": "0123456789",  
    "address": "123 Đường ABC"  
}
```

2.2. Đăng nhập (5 điểm)

Endpoint: [POST /api/auth/login](#)

2.3. Middleware Authentication (5 điểm)

2.4. Authorization (5 điểm) - Admin cho quản lý menu, tables

Phần 3: Customer Management API (5 điểm)

3.1. Lấy danh sách Customers (2 điểm)

Endpoint: GET /api/customers - Chỉ admin

3.2. Lấy Customer theo ID (2 điểm)

Endpoint: GET /api/customers/:id - Admin hoặc chính customer đó

3.3. Cập nhật Customer (2 điểm)

Endpoint: PUT /api/customers/:id

3.4. Lấy thông tin Customer hiện tại (2 điểm)

Endpoint: GET /api/auth/me

Phần 4: Menu Management API (12 điểm)

4.1. Lấy danh sách Menu Items (3 điểm)

Endpoint: GET /api/menu-items

Query Parameters:

- `page, limit`: Pagination
- `search`: Tìm kiếm trong name, description
- `category`: Lọc theo category
- `vegetarian_only`: true/false
- `spicy_only`: true/false
- `available_only`: true/false

4.2. Lấy Menu Item theo ID (2 điểm)

Endpoint: GET /api/menu-items/:id

4.3. Thêm Menu Item (4 điểm)

Endpoint: POST /api/menu-items - Chỉ admin

Request Body:

```
{  
    "name": "Phở Bò",  
    "description": "Phở bò truyền thống",  
    "category": "Main Course",  
    "price": 80000,  
    "preparation_time": 15,  
    "is_vegetarian": false,  
    "is_spicy": false,  
    "image_url": "https://example.com/pho.jpg"  
}
```

4.4. Cập nhật Menu Item (4 điểm)

Endpoint: PUT /api/menu-items/:id - Chỉ admin

4.5. Xóa Menu Item (2 điểm)

Endpoint: DELETE /api/menu-items/:id - Chỉ admin

- Kiểm tra item có trong reservation_items không (reservation chưa completed)

4.6. Tìm kiếm Menu Items (2 điểm)

Endpoint: GET /api/menu-items/search

Phần 5: Reservation Management API (25 điểm)

5.1. Đặt Bàn (6 điểm)

Endpoint: POST /api/reservations

Request Body:

```
{  
    "reservation_date": "2024-01-15T19:00:00Z",  
    "number_of_guests": 4,  
    "special_requests": "Bàn gần cửa sổ"  
}
```

Yêu cầu:

- Cần authentication
- Tạo reservation với reservation_number tự sinh (VD: "RES-20240115-001")
- Status = "pending"

5.2. Thêm Món vào Đơn (6 điểm)

Endpoint: POST /api/reservations/:id/items

Request Body:

```
{  
    "menu_item_id": 1,  
    "quantity": 2  
}
```

Yêu cầu:

- Kiểm tra menu_item is_available
- Tạo reservation_item
- Tính lại subtotal, service_charge (10%), total

5.3. Xác nhận Đặt Bàn (4 điểm)

Endpoint: PUT /api/reservations/:id/confirm

Request Body:

```
{  
    "table_number": "T01"  
}
```

Yêu cầu:

- Chỉ admin
- Kiểm tra table có available và đủ capacity không
- Set status = "confirmed", table_number
- Set table.is_available = false

4.4. Lấy Đặt Bàn theo ID (2 điểm)

Endpoint: GET /api/reservations/:id

- Include reservation_items với menu_item info

5.5. Lấy Đặt Bàn của Customer (3 điểm)

Endpoint: GET /api/customers/:id/reservations

- Pagination
- Filter by status
- Include reservation_items

5.6. Thanh toán (7 điểm)

Endpoint: POST /api/reservations/:id/pay

Request Body:

```
{  
    "payment_method": "card",  
    "use_loyalty_points": true,  
    "loyalty_points_to_use": 500  
}
```

Yêu cầu:

- Validate: reservation status = "seated"
- Tính discount từ loyalty_points (1 point = 1000đ, tối đa 50% total)
- Tính lại total sau discount
- Set payment_status = "paid", status = "completed"
- Cộng loyalty_points cho customer (1% total)
- Trừ loyalty_points đã dùng
- Set table.is_available = true

5.7. Hủy Đặt Bàn (2 điểm)

Endpoint: `DELETE /api/reservations/:id`

- Customer có thể hủy (status = "pending" hoặc "confirmed")
- Admin có thể hủy bất kỳ
- Set status = "cancelled"
- Nếu đã confirm: set table.is_available = true

Phần 6: Table Management API (5 điểm)

6.1. Lấy danh sách Tables (2 điểm)

Endpoint: `GET /api/tables`

- Filter: `available_only=true`

6.2. Thêm Table (1 điểm)

Endpoint: `POST /api/tables` - Chỉ admin

6.3. Cập nhật Table (1 điểm)

Endpoint: `PUT /api/tables/:id` - Chỉ admin

6.4. Xóa Table (1 điểm)

Endpoint: `DELETE /api/tables/:id` - Chỉ admin

Phần 7: Error Handling & Validation (3 điểm)

1. **Global Error Handler** (1 điểm)
 2. **Input Validation** (1 điểm)
 3. **Database Transaction** (1 điểm) - Cho pay reservation
-

Phần 8: Ứng dụng Client với Flutter (20 điểm)

Lưu ý: Bạn cần tạo một project Flutter riêng biệt (nằm trong thư mục `flutter_app_<MaSinhVien>`).

1. Màn hình Đăng nhập (5 điểm)

- UI: Form đăng nhập.
- Logic: Gọi API [/api/auth/login](#).

2. Màn hình Danh sách Món ăn (Menu) (10 điểm)

- Gọi API [/api/menu-items](#).
- Hiển thị danh sách món: Hình ảnh, Tên, Giá.
- Hiển thị món chay/mặn, cay/không cay.

3. Màn hình Chi tiết Món (5 điểm)

- Hiển thị chi tiết món ăn.
- Nguyên liệu, thời gian chế biến.

YÊU CẦU KỸ THUẬT

- RESTful API Design
- Response Format nhất quán
- Security
- Code Organization

API DOCUMENTATION

Tạo file [API_DOCUMENTATION.md](#) hoặc Swagger/Postman Collection

CHECKLIST NỘP BÀI

- Project Web API hoàn chỉnh, project Flutter hoàn chỉnh
- Database đã migrate và có seeder data
- Tất cả endpoints hoạt động đúng
- Authentication và Authorization hoạt động
- Validation đầy đủ
- Error handling toàn cục
- Response format nhất quán
- API Documentation đầy đủ
- CORS đã cấu hình
- Code tổ chức rõ ràng
- File README.md
- File .env.example

THANG ĐIỂM CHI TIẾT

Phần	Điểm
Thiết lập Project và Database	10

Phần	Điểm
Authentication & Authorization	20
Customer Management API	5
Menu Management API	12
Reservation Management API	25
Table Management API	5
Error Handling & Validation	3
Flutter Client App	20
Tổng	100

Bonus điểm (tối đa +10):

- Unit/Integration Tests: +5
 - API Documentation chi tiết (Swagger): +3
 - Code quality cao: +2
-

Chúc các bạn làm bài tốt! 🎉