**📝 TÀI LIỆU BACKEND POS**

**1. Giới thiệu tổng quan**

**POS Cashier System** là API backend (NodeJS, Express, Prisma, MySQL) dành cho ứng dụng bán hàng/quản lý bán tại quầy thu ngân.

* Frontend có thể dùng React, Axios hoặc Fetch gọi API JSON.
* Bảo mật dựa vào JWT – login để lấy token, chèn vào header các API.
* Sản phẩm/tài khoản ban đầu đã có sẵn dữ liệu mẫu (prisma/seed.js).

**2. Cách chạy backend**

**Chuẩn bị môi trường**

* Đảm bảo cài đặt Node.js ≥ 16, đã có MySQL

**Cài đặt dépendencies:**

npm install

**Cấu hình database**

* Tạo file .env (nếu chưa có):

text

DATABASE\_URL="mysql://root:@localhost:3306/employee\_management"

PORT=3000

NODE\_ENV=development

JWT\_SECRET=employee\_pos\_secret\_key\_2024

JWT\_EXPIRE=8h

FRONTEND\_URL=http://localhost:3001

**Chạy migration, seed dữ liệu:**

text

npm run db:generate

npm run db:migrate

npm run db:seed

**Khởi động server:**

text

npm run dev

Server mặc định chạy tại: [**http://localhost:3000**](http://localhost:3000/)

**3. Danh sách API – sử dụng cho Frontend React**

**GHI CHÚ CHUNG:**

* Tất cả các API bảo vệ (trừ login) phải gửi kèm header:

text

Authorization: Bearer <token>

* Tất cả body gửi/nhận và responses đều dùng JSON.

**A. AUTH - Đăng nhập, đăng xuất**

**1. Đăng nhập**

text

POST /api/auth/login

* Body:

json

{

"tai\_khoan": "cashier1",

"mat\_khau": "123456"

}

* Response thành công:

json

{

"success": true,

"data": {

"token": "eyJhbGciOiJIUzI1... (JWT)",

"employee": {

"id": 1, "ho\_ten": "Nguyễn Thu Ngân", "tai\_khoan": "cashier1", "vai\_tro": "thu\_ngan", "ca\_lam\_viec": "CA\_SANG"

}

}

}

**2. Lấy thông tin nhân viên**

text

GET /api/auth/profile

* Header: Authorization: Bearer <token>

**3. Đăng xuất/Kết thúc ca**

text

POST /api/auth/logout

* Header: Authorization: Bearer <token>

**B. SẢN PHẨM – tìm kiếm & quét mã vạch**

**1. Tìm kiếm sản phẩm**

text

GET /api/pos/products/search?q=<keyword>

* Ví dụ: /api/pos/products/search?q=coca
* Trả về danh sách sản phẩm còn hàng, chứa keyword trong tên/mã/mã vạch.

**2. Quét mã vạch**

text

GET /api/pos/products/barcode/:barcode

* Ví dụ: /api/pos/products/barcode/8935049001001

**3. Lấy sản phẩm bán chạy**

text

GET /api/pos/products/popular?limit=8

* Trả về các sản phẩm nổi bật hoặc được bán nhiều.

**C. HÓA ĐƠN – tạo đơn hàng và thống kê**

**1. Tạo đơn hàng**

text

POST /api/pos/orders

* Body:

json

{

"items": [

{ "id\_san\_pham": 1, "so\_luong": 2 },

{ "id\_san\_pham": 4, "so\_luong": 1 }

],

"tien\_khach\_dua": 50000,

"phuong\_thuc\_tt": "TIEN\_MAT", *// hoặc "THE\_NGAN\_HANG", "CHUYEN\_KHOAN"*

"ghi\_chu": "Khách mang đi"

}

* trả về: tổng tiền, tiền thối, chi tiết đơn hàng vừa tạo.

**D. ĐƠN ĐÃ BÁN – quản lý & thống kê ca**

**1. Lịch sử đơn hàng của thu ngân**

text

GET /api/orders/my-orders?page=1&limit=10

**2. Xem chi tiết đơn hàng**

text

GET /api/orders/:id

**3. Thống kê ca làm việc**

text

GET /api/orders/shift-stats

GET /api/orders/shift-stats?date=YYYY-MM-DD

* trả về số đơn đã bán, tổng doanh thu trong ca.

**4. Tài khoản test**

| **Tài khoản** | **Mật khẩu** | **Ghi chú** |
| --- | --- | --- |
| cashier1 | 123456 | Ca sáng |
| cashier2 | 123456 | Ca chiều |
| cashier3 | 123456 | Ca tối |

**5. Error format**

Dữ liệu lỗi luôn có dạng:

json

{

"success": false,

"message": "Giải thích lỗi bằng tiếng Việt",

"errors": [ ... ] *// nếu là validation*

}

**6. Code React mẫu**

**Sử dụng Axios:**

js

**import** axios **from** 'axios';

**export** **const** api = axios.create({

baseURL: 'http://localhost:3000',

});

*// Thêm token vào mọi request*

api.interceptors.request.use((config) => {

**const** token = localStorage.getItem('pos\_token');

**if** (token) config.headers.Authorization = `Bearer ${token}`;

**return** config;

});

**export** **async** **function** loginPOS({ tai\_khoan, mat\_khau }) {

**return** api.post('/api/auth/login', { tai\_khoan, mat\_khau });

}

**export** **async** **function** searchProducts(q) {

**return** api.get(`/api/pos/products/search?q=${encodeURIComponent(q)}`);

}

**7. Lưu ý**

* Luôn kiểm tra mã trạng thái HTTP trả về, đặc biệt là 401/403 để redirect về trang đăng nhập nếu token hết hạn.
* Có thể test các API này bằng Postman như trong bản hướng dẫn chi tiết ở trên.

**Dùng đúng flow login -> lưu token -> các API còn lại phải đính kèm header Authorization.**  
Nếu gặp lỗi bất kỳ, backend sẽ trả về thông báo rõ ràng để frontend hiển thị hoặc xử lý hợp lý.

**Bạn chỉ cần follow các endpoint trên là giao tiếp frontend-react với backend này sẽ thuận tiện tối đa.**