**1. Nên lựa chọn giữa đăng nhập thường và đăng nhập bằng Google hoặc Facebook**

Phụ thuộc vào nhu cầu và đối tượng người dùng Web

***- Đăng nhập Thường:***

+ Không phụ thuộc vào bên thứ ba để lưu trữ dữ liệu, vì vậy có thể kiểm soát toàn bộ thông tin người dùng.

+ Một số người dùng thích đăng ký và đăng nhập bằng email và mật khẩu để tránh chia sẻ thông tin với bên thứ ba.

+ Có thể dễ dàng điều chỉnh cách xác thực như xác thực 2 lớp, yêu cầu đặt lại mật khẩu,…

🡪 Tuy nhiên, Người dùng có thể quên mật khẩu, dẫn đến việc phải thực hiện quy trình đặt lại mật khẩu, làm mất thời gian.

***- Đăng nhập bằng Google hoặc Facebook***: Tiện lợi và nhanh chóng, không cần phải tạo tài khoản mới, ít quên thông tin đăng nhập

+ Google:

* Người dùng Google thường xuyên sử dụng các dịch vụ như Gmail, YouTube, Google Photos, nên tài khoản Google được nhiều người có sẵn và dễ dàng truy cập.
* Google cung cấp các tính năng bảo mật mạnh mẽ như xác thực hai lớp và bảo mật OAuth 2.0, giúp bảo vệ tài khoản của người dùng tốt hơn.

+ Facebook:

* Dữ liệu người dùng phong phú: Facebook cho phép truy cập vào nhiều dữ liệu xã hội hơn như danh sách bạn bè, hình ảnh, sở thích,…
* Phổ biến với người dùng mạng xã hội: đăng nhập bằng Facebook giúp kết nối dễ dàng hơn.

🡪 Tuy nhiên, sẽ phụ thuộc vào Google hoặc Facebook để xử lý đăng nhập, Google và Facebook kiểm soát dữ liệu mà bạn có thể truy cập, có thể sẽ không lấy được thông tin như email hay số điện thoại người dùng.

🡺 ***Ưu tiên đăng nhập bằng Google/Facebook nếu web có tính chất mạng xã hội hoặc yêu cầu thao tác nhanh*.** Tích hợp cả đăng nhập thường và đăng nhập bằng Google/Facebook là một cách tốt để đáp ứng nhu cầu đa dạng của người dùng.

**2. Các công dụng và kỹ thuật của các chức năng**

***2.1 Đăng ký tài khoản và đăng nhập***

- **Công dụng**: Cho phép khách hàng đăng ký tài khoản và đăng nhập, giúp lưu trữ thông tin cá nhân và lịch sử mua sắm.

- **Kỹ thuật**:

+ Đăng ký:

* Người dùng cung cấp tên đăng nhập, mật khẩu, email, tên đầy đủ,...
* Xác thực các thông tin này (kiểm tra email đã tồn tại hay chưa, định dạng email đúng, mật khẩu đủ mạnh).
* Nếu tất cả thông tin hợp lệ, tạo tài khoản mới và trả về mã 201 (Created), kèm theo thông tin người dùng.
* Nếu không hợp lệ (ví dụ: email đã tồn tại), trả về mã 400 (BadRequest).

+ Đăng nhập:

* Người dùng cung cấp tên đăng nhập và mật khẩu.
* Hệ thống kiểm tra thông tin đăng nhập, nếu đúng thì cấp một JWT token để xác thực các yêu cầu API sau này.
* Trả về mã 200 (OK) với token nếu đăng nhập thành công, mã 400 nếu thông tin sai hoặc 500 nếu có lỗi hệ thống.

***2.2 Đăng nhập bằng tài khoản Google và Facebook***

- **Công dụng**: Cho phép người dùng đăng nhập nhanh qua các dịch vụ bên ngoài như Google và Facebook, giúp tiết kiệm thời gian và không phải tạo tài khoản mới.

- **Kỹ thuật**:

+ Google: Sử dụng OAuth2.0 để xác thực người dùng. Sau khi xác thực qua Google, hệ thống nhận mã thông báo và lấy thông tin người dùng (như tên, email, ảnh đại diện).

+ Facebook: Sử dụng Graph API của Facebook để nhận token xác thực và lấy thông tin người dùng. Tương tự Google, hệ thống sẽ yêu cầu xác thực từ Facebook và xử lý thông tin trả về.

***2.3 Đăng nhập trang quản trị***

- **Công dụng**: Chỉ cho phép quản trị viên (Admin) đăng nhập vào trang quản lý, đảm bảo quyền truy cập chỉ dành cho người có vai trò quản trị.

- **Kỹ thuật**:

+ Phân quyền thông qua vai trò người dùng (Admin).

+ Sử dụng Cookie Authentication hoặc JWT để duy trì trạng thái đăng nhập của quản trị viên và kiểm tra quyền truy cập.

***2.4 Quên mật khẩu***

- **Công dụng**: Giúp người dùng lấy lại mật khẩu khi quên, thông qua email xác nhận.

- **Kỹ thuật**:

+ Người dùng cung cấp email để hệ thống gửi một liên kết khôi phục mật khẩu.

+ Liên kết này thường có thời gian hết hạn để đảm bảo bảo mật.

+ Hệ thống gửi email chứa mã thông báo xác thực và cho phép người dùng nhập mật khẩu mới.

***2.5 Đổi mật khẩu***

- **Công dụng**: Phương thức này cho phép người dùng thay đổi mật khẩu tài khoản của họ. Việc đổi mật khẩu giúp tăng cường bảo mật tài khoản, đặc biệt khi người dùng cho rằng mật khẩu cũ có thể bị lộ hoặc bị tấn công.

- **Kỹ thuật**:

+ Người dùng phải nhập mật khẩu cũ và mật khẩu mới.

+ Mật khẩu mới được mã hóa (hashing) trước khi lưu vào cơ sở dữ liệu.

+ Kiểm tra tính hợp lệ của mật khẩu mới (độ dài, ký tự đặc biệt,...) và cập nhật cơ sở dữ liệu nếu hợp lệ.

***2.6 Đăng xuất***

- **Công dụng**: Giúp người dùng thoát khỏi hệ thống và bảo mật tài khoản. Khi đăng xuất, các token xác thực của người dùng sẽ bị vô hiệu hóa hoặc xóa bỏ khỏi hệ thống, giúp đảm bảo rằng người dùng không còn quyền truy cập vào hệ thống nữa.

- **Kỹ thuật**: Hệ thống xóa hoặc vô hiệu hóa token JWT hoặc cookie xác thực để người dùng không thể tiếp tục sử dụng ứng dụng mà không đăng nhập lại.

***2.6 Quản lý sản phẩm***

- **Công dụng**:

+ Thêm mới sản phẩm: Quản trị viên có thể bổ sung các sản phẩm mới vào hệ thống.

+ Chỉnh sửa thông tin sản phẩm: Cho phép cập nhật các thông tin sản phẩm như giá, mô tả, hình ảnh, và trạng thái (còn hàng, hết hàng).

+ Xóa sản phẩm: Loại bỏ các sản phẩm không còn được kinh doanh.

+ Xem chi tiết sản phẩm: Xem toàn bộ thông tin sản phẩm để quản lý tốt hơn.

- **Kỹ thuật**:

*+ API CRUD:*

* **Create (Thêm mới)**:
  + Sử dụng API POST /api/products để gửi dữ liệu sản phẩm (tên, giá, danh mục, mô tả, hình ảnh).
  + Validation dữ liệu phía server trước khi lưu.
* **Read (Xem)**:
  + GET /api/products để lấy danh sách sản phẩm có hỗ trợ phân trang và lọc.
  + GET /api/products/{id} để xem chi tiết sản phẩm.
* **Update (Chỉnh sửa)**:
  + PUT /api/products/{id} để cập nhật thông tin sản phẩm.
* **Delete (Xóa)**:
  + DELETE /api/products/{id} để xóa sản phẩm khỏi hệ thống.

*+ Giao diện quản trị:*

* Tạo trang quản lý với bảng danh sách sản phẩm, có các thao tác: Thêm, Sửa, Xóa, Xem chi tiết

***2.7 Quản lý danh mục sản phẩm***

- **Công dụng**:

+ Thêm mới danh mục: Giúp phân loại các sản phẩm theo danh mục.

+ Chỉnh sửa danh mục: Cập nhật thông tin danh mục như tên, mô tả.

+ Xóa danh mục: Xóa các danh mục không còn sử dụng.

+ Xem danh sách danh mục: Dễ dàng theo dõi và quản lý các danh mục hiện có

- **Kỹ thuật**:

*+ API CRUD:*

* **Create (Thêm mới)**:
  + POST /api/categories để thêm danh mục mới.
* **Read (Xem)**:
  + GET /api/categories để lấy danh sách danh mục (có phân trang).
  + GET /api/categories/{id} để xem chi tiết danh mục.
* **Update (Chỉnh sửa)**:
  + PUT /api/categories/{id} để cập nhật thông tin danh mục.
* **Delete (Xóa)**:
  + DELETE /api/categories/{id} để xóa danh mục.

*+ Giao diện quản trị:*

* Bảng danh sách danh mục với các chức năng: Thêm, Sửa, Xóa, Xem chi tiết

***2.8 Tìm kiếm***

- **Công dụng**:

+ Tăng trải nghiệm người dùng: Giúp người dùng nhanh chóng tìm được sản phẩm, đơn hàng hoặc thông tin cần thiết mà không cần duyệt qua toàn bộ danh sách.

+ Hỗ trợ đa dạng nhu cầu tìm kiếm:

* Tìm kiếm theo **từ khóa** (tên sản phẩm, mã đơn hàng,...).
* Tìm kiếm nâng cao dựa trên **các tiêu chí** như giá cả, danh mục, trạng thái đơn hàng,...

- **Kỹ thuật**:

+ Sử dụng các thuật toán tìm kiếm (có thể là tìm kiếm theo từ khóa, lọc theo giá, danh mục,...).

+ Tìm kiếm có thể hỗ trợ phân trang và sắp xếp kết quả.

***2.9 Quản lý người dùng***

- **Công dụng**:

+ Xem thông tin tài khoản: Theo dõi thông tin chi tiết của từng tài khoản.

+ Chỉnh sửa tài khoản: Thay đổi thông tin như vai trò, trạng thái hoạt động.

+ Xóa tài khoản: Loại bỏ những tài khoản vi phạm hoặc không còn hoạt động.

+ Thêm tài khoản Admin: Cho phép quản trị viên cấp cao tạo tài khoản Admin mới.

+ Phân quyền vai trò: Đảm bảo mỗi tài khoản có quyền hạn phù hợp (Admin, User)

- **Kỹ thuật**:

*+ API CRUD:*

* + **Create (Thêm)**:
    - POST /api/users để thêm người dùng mới, yêu cầu các thông tin như username, email, role.
  + **Read (Xem)**:
    - GET /api/users để lấy danh sách người dùng (có hỗ trợ lọc theo vai trò).
    - GET /api/users/{id} để xem chi tiết tài khoản.
  + **Update (Chỉnh sửa)**:
    - PUT /api/users/{id} để cập nhật thông tin tài khoản, bao gồm vai trò.
  + **Delete (Xóa)**:
    - DELETE /api/users/{id} để xóa tài khoản.

*+ Phân quyền và bảo mật:*

* + Chỉ Admin có quyền thêm, sửa, xóa tài khoản.
  + Sử dụng **JWT Authentication** để xác thực và kiểm soát quyền truy cập.

*+ Giao diện quản trị:*

* + Tạo trang danh sách người dùng với các chức năng: Thêm, Sửa, Xóa, Xem chi tiết

***2.10 Quản lý mã giảm giá***

- **Công dụng**:

+ Thêm mới mã giảm giá: Tăng doanh số bằng cách cung cấp ưu đãi cho khách hàng.

+ Chỉnh sửa thông tin mã giảm giá: Điều chỉnh giá trị hoặc ngày hết hạn của mã.

+ Xóa mã giảm giá: Xóa bỏ các mã không còn hiệu lực hoặc bị lỗi.

+ Xác thực mã khi thanh toán: Đảm bảo chỉ áp dụng mã hợp lệ, tránh gian lận.

- **Kỹ thuật**:

+ **API CRUD**:

* **Create (Thêm mới)**:
* POST /api/coupons để tạo mã giảm giá mới.
* **Read (Xem)**:
  + GET /api/coupons để lấy danh sách mã giảm giá.
  + GET /api/coupons/{code} để kiểm tra tính hợp lệ của mã.
* **Update (Chỉnh sửa)**:
* PUT /api/coupons/{id} để cập nhật thông tin.
* **Delete (Xóa)**:
  + DELETE /api/coupons/{id} để xóa mã giảm giá.

+ **Áp dụng mã giảm giá khi thanh toán**:

* Xác thực mã giảm giá thông qua API
* Nếu hợp lệ, giảm giá và cập nhật số lượng sử dụng mã

+ **Giao diện quản trị**:

* Trang quản lý danh sách mã giảm giá với các chức năng: Thêm, Sửa, Xóa, Xem chi tiết

***2.11 Giỏ hàng và Thanh toán***

- **Công dụng**: Quản lý các sản phẩm trong giỏ hàng của người dùng và xử lý thanh toán khi người dùng mua hàng.

- **Kỹ thuật**:

+ Giỏ hàng thường lưu trữ trong session hoặc cơ sở dữ liệu (nếu người dùng đăng nhập).

+ Sử dụng các dịch vụ thanh toán như Cash, VNPay, MoMo để xử lý thanh toán.

***2.12 Lịch sử và Trạng thái đơn hàng***

- **Công dụng**: Người dùng có thể xem lại lịch sử mua hàng và theo dõi trạng thái của đơn hàng.

- **Kỹ thuật**: Lưu trữ thông tin đơn hàng trong cơ sở dữ liệu và cung cấp các API để người dùng có thể tra cứu trạng thái đơn hàng.

***2.13 Cập nhật số lượng sản phẩm trong giỏ hàng***

- **Công dụng**: Cho phép người dùng điều chỉnh số lượng sản phẩm trong giỏ hàng.

- **Kỹ thuật**:

+ Sử dụng API để cập nhật số lượng sản phẩm trong giỏ hàng.

+ Kiểm tra xem số lượng có hợp lệ (>= 0), nếu số lượng bằng 0 thì sản phẩm sẽ bị xóa khỏi giỏ hàng.

***2.14 Phân trang***

- **Công dụng**:

+ Tăng hiệu suất hiển thị: Tránh việc tải toàn bộ dữ liệu một lần, giảm thời gian tải và tài nguyên.

+ Tăng trải nghiệm người dùng: Giúp hiển thị dữ liệu dễ theo dõi hơn khi chia nhỏ thành từng trang.

+ Hỗ trợ phân trang linh hoạt: Cải thiện khả năng tìm kiếm và lọc dữ liệu.

- **Kỹ thuật**: Sử dụng kỹ thuật phân trang (pagination) trong API để chia nhỏ dữ liệu và dễ dàng hiển thị.

***2.15 Cập nhật thông tin cá nhân***

- **Công dụng**: Cho phép người dùng cập nhật các thông tin cá nhân như email, số điện thoại, mật khẩu,…

- **Kỹ thuật**:

+ Tạo form cập nhật thông tin cá nhân.

+ Kiểm tra tính hợp lệ của dữ liệu (định dạng email, số điện thoại,…).

+ Cập nhật thông tin vào cơ sở dữ liệu nếu hợp lệ.

***2.16 AI Chat Box***

- **Công dụng**:

+ Gợi ý món ăn phù hợp: Dựa vào các thông tin cá nhân như độ tuổi, giới tính, thể trạng, dị ứng, hoặc sở thích, AI sẽ đưa ra các món ăn phù hợp, tối ưu hóa dinh dưỡng và sức khỏe cho người dùng.

+ Cải thiện trải nghiệm khách hàng: Cung cấp hỗ trợ nhanh chóng 24/7, tăng sự hài lòng của khách hàng bằng cách cung cấp thông tin món ăn hoặc hỗ trợ đặt món.

**- Kỹ thuật:**

+ Xử lý ngôn ngữ tự nhiên (NLP):

* Sử dụng thư viện AI như **ChatGPT** để xử lý và phản hồi ngôn ngữ tự nhiên của người dùng.
* Phân tích câu hỏi để nhận diện yêu cầu (ví dụ: "Tôi muốn món ít calo cho người 50 tuổi").

+ Tích hợp dữ liệu người dùng:

* Thu thập thông tin cá nhân từ hồ sơ khách hàng (tuổi, giới tính, dị ứng, thể trạng).
* Lưu trữ và quản lý dữ liệu này trong cơ sở dữ liệu an toàn, sử dụng API để tương tác với dữ liệu.

+ Hệ thống gợi ý (Recommendation System):

* Xây dựng mô hình AI dựa trên các thuật toán gợi ý như **Collaborative Filtering** hoặc **Content-Based Filtering**.
* Tích hợp thêm dữ liệu dinh dưỡng, phân loại món ăn, và phân tích dinh dưỡng từ cơ sở dữ liệu về món ăn.

***2.17 Bình luận và đánh giá cho từng món thức ăn (chỉ khách mua hàng mới được đánh giá)***

**- Công dụng:**

**+** Cải thiện chất lượng dịch vụ: Nhận phản hồi từ khách hàng để cải tiến món ăn và dịch vụ.

**+** Tăng độ tin cậy: Những đánh giá từ khách hàng thực sẽ tăng sự tin tưởng của khách hàng mới.

**+** Hỗ trợ khách hàng khác: Đánh giá giúp khách hàng khác lựa chọn món ăn dễ dàng hơn.

**- Kỹ thuật:**

+ Xác thực người dùng:

* Chỉ cho phép người dùng đã đặt hàng và nhận món ăn mới có thể đánh giá.
* Xác minh qua mã đơn hàng hoặc trạng thái đơn hàng (nếu đơn hàng đã giao thành công).

+ Tạo hệ thống đánh giá: Tạo giao diện người dùng để nhập bình luận và đánh giá (ví dụ: 1-5 sao).

+ Hiển thị đánh giá:

* + Tích hợp API để hiển thị đánh giá theo món ăn.
  + Sắp xếp đánh giá theo thời gian, mức độ hữu ích, hoặc điểm đánh giá cao.

***2.18 Mã QR Code cho phần đăng nhập, đăng ký, theo dõi đơn hàng***

**- Công dụng:**

**+** Tiện lợi và nhanh chóng: Người dùng có thể đăng nhập, đăng ký hoặc theo dõi đơn hàng chỉ bằng cách quét mã QR, không cần nhập thông tin thủ công.

**+** Tăng tính bảo mật: Sử dụng mã QR mã hóa để giảm nguy cơ lộ thông tin tài khoản.

**- Kỹ thuật:**

+ Tạo mã QR (QR Code Generation):

* + Sử dụng thư viện như **QRCode.NET**, **ZXing.NET** (cho .NET) để tạo mã QR.
  + Mã hóa thông tin cần thiết:
    - **Đăng nhập**: Mã hóa token xác thực JWT.
    - **Đăng ký**: Chứa liên kết dẫn đến form đăng ký trên ứng dụng hoặc web.
    - **Theo dõi đơn hàng**: Chứa mã đơn hàng và liên kết đến API tra cứu đơn hàng.

+ Quét mã QR (QR Code Scanning):

* + Tích hợp thư viện đọc mã QR (**ZXing.NET**, **Google ML Kit**) trên trình duyệt.
  + Giải mã dữ liệu từ mã QR và thực hiện hành động tương ứng (đăng nhập, mở form đăng ký, hoặc tra cứu đơn hàng).

**+** Bảo mật mã QR: Mã hóa thông tin bên trong mã QR bằng thuật toán AES hoặc SHA-256 để bảo vệ dữ liệu.

**+** Tích hợp hệ thống:

* + Kết nối mã QR với API:
    - API xác thực đăng nhập.
    - API đăng ký tài khoản mới.
    - API tra cứu trạng thái đơn hàng.

***2.19 Quản lý đánh giá món ăn (Đã duyệt thì mới hiển thị lên)***

**- Công dụng:**

+ Kiểm soát chất lượng đánh giá: Đảm bảo chỉ những đánh giá hợp lệ, không vi phạm nội quy (spam, ngôn ngữ không phù hợp) mới được hiển thị.

+ Tăng tính minh bạch và tin cậy: Ngăn chặn các đánh giá không trung thực hoặc có mục đích xấu, từ đó xây dựng lòng tin với khách hàng.

+ Quản lý nội dung: Cho phép quản trị viên kiểm duyệt các đánh giá trước khi công khai, bảo vệ hình ảnh của cửa hàng.

**- Kỹ thuật:**

**+** Hệ thống phê duyệt đánh giá:

* + Thêm thuộc tính trạng thái đánh giá (IsApproved) trong cơ sở dữ liệu, với giá trị mặc định là false.
  + Mỗi đánh giá mới sẽ được lưu trữ nhưng không hiển thị cho đến khi quản trị viên phê duyệt.

**+** Giao diện quản trị:

* + Tạo bảng hiển thị danh sách đánh giá chưa duyệt, bao gồm thông tin: nội dung, số sao, tên người dùng, và món ăn.
  + Hiển thị nút **Duyệt** hoặc **Xóa** để quản trị viên xử lý đánh giá.

**+** API hiển thị đánh giá: Khi lấy danh sách đánh giá, chỉ truy vấn những đánh giá được duyệt

**+** Gửi thông báo cho người dùng: Gửi thông báo khi đánh giá được phê duyệt hoặc bị từ chối, kèm lý do nếu cần.

***2.20 Lọc tài khoản người dùng (Khóa tài khoản liên quan đến hủy đơn nhiều và đặt không nhận, sau đó thông báo qua email)***

**- Công dụng:**

* Ngăn chặn hành vi không hợp lệ: Hạn chế tình trạng người dùng gây thất thoát cho cửa hàng (hủy đơn nhiều, đặt nhưng không nhận).
* Tăng hiệu quả kinh doanh: Bảo vệ cửa hàng khỏi các hành vi tiêu cực làm lãng phí tài nguyên.
* Thông báo minh bạch: Thông báo lý do khóa tài khoản giúp khách hàng hiểu rõ hành vi sai phạm.

- **Kỹ thuật:**

**+** Theo dõi lịch sử đơn hàng:

* Lưu trạng thái đơn hàng (hủy, không nhận) trong cơ sở dữ liệu

**+** Phân tích hành vi người dùng:

* Định nghĩa ngưỡng vi phạm, ví dụ:
  + Hủy từ 3 đơn trở lên trong 30 ngày.
  + Không nhận hàng 2 lần liên tiếp.

+ Tự động khóa tài khoản:

* Thêm thuộc tính IsLocked vào bảng tài khoản
* Tự động cập nhật trạng thái IsLocked = true nếu phát hiện vi phạm.

+ Gửi email thông báo:

* Sử dụng dịch vụ email như SendGrid, SMTP, hoặc MailKit để gửi email thông báo lý do khóa tài khoản.
* Tạo hệ thống mẫu email (template) để tùy chỉnh nội dung dễ dàng.

+ Hiển thị thông báo ngay trên giao diện khi người dùng cố gắng đăng nhập vào tài khoản bị khóa.

***2.21 Đơn hàng thanh toán với 3 hình thức (Momo, VNPay, tiền mặt)***

**- Công dụng:**

**+** Đa dạng hình thức thanh toán: Tăng tiện lợi và trải nghiệm cho khách hàng khi cho phép lựa chọn phương thức thanh toán phù hợp.

**+** Hỗ trợ thanh toán trực tuyến: Thanh toán qua Momo và VNPay giúp khách hàng giao dịch nhanh chóng, an toàn.

**+** Thanh toán tiền mặt: Phù hợp với khách hàng chưa quen thanh toán trực tuyến hoặc muốn thanh toán khi nhận hàng.

**- Kỹ thuật:**

**+ Tích hợp Momo**:

* *API Momo:*
  + - Gửi yêu cầu tạo giao dịch đến API của Momo
    - Nhận liên kết thanh toán và chuyển hướng khách hàng đến trang thanh toán của Momo.
* *Quy trình:*
  + - Tạo mã đơn hàng và tính tổng số tiền.
    - Gửi yêu cầu đến Momo API với các tham số: mã đơn hàng, số tiền, URL callback.
    - Nhận kết quả và lưu trạng thái thanh toán (Paid, Failed,...).

**+ Tích hợp VNPay**:

* + *API VNPay*:
    - Sử dụng SDK hoặc API của VNPay để tạo URL thanh toán.
    - Xử lý callback từ VNPay để xác nhận trạng thái thanh toán.
  + *Quy trình:*
    - Gửi yêu cầu thanh toán qua URL VNPay với thông tin giao dịch (merchant\_id, tổng tiền, đơn hàng).
    - Nhận phản hồi trạng thái từ VNPay thông qua callback.
    - Lưu trạng thái thanh toán vào cơ sở dữ liệu.

**+ Thanh toán tiền mặt (COD)**:

* + *Quy trình:*
    - Chọn tùy chọn "Thanh toán khi nhận hàng" (COD) trong giao diện đặt hàng.
    - Lưu trạng thái đơn hàng là Pending Payment và cập nhật thành Paid khi giao hàng thành công.
    - Xử lý logic thay đổi trạng thái giao dịch dựa trên phản hồi từ shipper.

***2.22 Xem chi tiết đơn hàng***

**- Công dụng:**

**+** Thông tin minh bạch: Cho phép khách hàng theo dõi chi tiết về đơn hàng, bao gồm sản phẩm, giá cả, trạng thái, và phương thức thanh toán.

**+** Hỗ trợ quản lý: Khách hàng và quản trị viên có thể kiểm tra lịch sử đơn hàng, xử lý các vấn đề liên quan.

**- Kỹ thuật:**

*+ API chi tiết đơn hàng:*

* + Tạo endpoint API để lấy thông tin đơn hàng dựa trên OrderId
  + Kết hợp bảng Orders và OrderDetails trong cơ sở dữ liệu để truy xuất dữ liệu

***+ Giao diện người dùng:***

* Tạo trang hiển thị chi tiết đơn hàng với các thông tin:
  + Mã đơn hàng.
  + Danh sách sản phẩm (tên, số lượng, giá).
  + Tổng tiền.
  + Trạng thái (đang xử lý, đã giao, đã hủy).
  + Phương thức thanh toán.
* Thiết kế giao diện dễ đọc, có nút **Mua lại**.

***+ Xác thực bảo mật:***

* Chỉ cho phép người dùng sở hữu đơn hàng hoặc quản trị viên xem chi tiết.

***2.23 Mua lại đơn hàng sau khi hủy đơn hoặc giao đơn thành công***

**- Công dụng:**

* Tiện ích cho khách hàng: Cho phép khách hàng đặt lại đơn hàng đã hủy hoặc đã giao mà không cần chọn từng sản phẩm. Tiết kiệm thời gian và tăng tiện lợi cho khách hàng.
* Tăng doanh số: Khuyến khích khách hàng quay lại mua sắm, cải thiện trải nghiệm người dùng.

**- Kỹ thuật:**

*+ API tạo đơn hàng từ đơn hàng cũ:*

* + Tạo endpoint API để tạo đơn hàng mới từ đơn hàng cũ
* Quy trình trong service:
* Truy xuất danh sách sản phẩm từ đơn hàng cũ (OrderDetails).
* Tạo đơn hàng mới với các thông tin:
  + Danh sách sản phẩm.
  + Thông tin người nhận.
  + Phương thức thanh toán (cho phép chỉnh sửa).
* Lưu đơn hàng mới và trả kết quả.

*+ Giao diện người dùng:*

* Thêm nút **Mua lại** bên cạnh chi tiết đơn hàng.
* Khi nhấn nút, hiển thị giỏ hàng được tự động thêm các sản phẩm từ đơn hàng cũ.

*+ Kiểm tra trạng thái đơn hàng:*

* Chỉ cho phép mua lại nếu đơn hàng:
  + Đã bị hủy.
  + Đã giao thành công.
* Thực hiện kiểm tra trạng thái trước khi xử lý.

*+ Tùy chỉnh trước khi mua lại:*

* Cho phép khách hàng thay đổi số lượng hoặc loại bỏ sản phẩm không cần thiết trước khi đặt lại.