**TUẦN 1**

**nguoi\_dung (User Table)**

id (INT, PRIMARY KEY)

mat\_khau (VARCHAR): Mật khẩu của người dùng (được mã hóa)

email (VARCHAR, UNIQUE): Email đăng nhập

vai\_tro (ENUM): Phân biệt giữa admin và user

ho\_ten (VARCHAR): Tên đầy đủ của người dùng

dia\_chi (TEXT): Địa chỉ người dùng

trang\_thai (BOOLEAN): Trạng thái tài khoản

ngay\_tao (DATETIME): Ngày tạo tài khoản

**san\_pham (Product Table)**

id (INT, PRIMARY KEY)

ten\_san\_pham (VARCHAR): Tên sản phẩm

gia (FLOAT): Giá sản phẩm

mo\_ta (TEXT): Mô tả sản phẩm

hinh\_anh (VARCHAR): Tên file hình ảnh sản phẩm

so\_luong (INT): Số lượng sản phẩm

id\_thuong\_hieu (INT, FOREIGN KEY): Liên kết với bảng thuong\_hieu

ngay\_tao (DATETIME): Ngày tạo sản phẩm

**thuong\_hieu (Brand Table)**

id (INT, PRIMARY KEY)

ten\_thuong\_hieu (VARCHAR): Tên thương hiệu

trang\_thai (BOOLEAN): Trạng thái của thương hiệu

**gio\_hang (Cart Table)**

id (INT, PRIMARY KEY)

id\_nguoi\_dung (INT, FOREIGN KEY): Liên kết với nguoi\_dung

so\_luong (INT): Số lượng sản phẩm trong giỏ

ngay\_cap\_nhat (DATETIME): Ngày cập nhật giỏ hàng

**don\_hang (Order Table)**

id (INT, PRIMARY KEY)

id\_nguoi\_dung (INT, FOREIGN KEY): Người đặt hàng

tong\_tien (FLOAT): Tổng tiền của đơn hàng

ghi\_chu (TEXT): Ghi chú của khách hàng

trang\_thai (ENUM): Trạng thái đơn hàng (pending, completed, cancelled)

ngay\_tao (DATETIME): Ngày tạo đơn hàng

**chi\_tiet\_hoa\_don (Order Details Table)**

id (INT, PRIMARY KEY)

id\_don\_hang (INT, FOREIGN KEY): Đơn hàng liên quan

id\_san\_pham (INT, FOREIGN KEY): Sản phẩm trong đơn hàng

so\_luong (INT): Số lượng sản phẩm

gia (FLOAT): Giá của sản phẩm tại thời điểm đặt hàng

**thanh\_toan (Payment Table)**

id (INT, PRIMARY KEY)

id\_don\_hang (INT, FOREIGN KEY)

phuong\_thuc (ENUM): Phương thức thanh toán (COD, Online)

ngay\_thanh\_toan (DATETIME)

**Xây Dựng Backend (Node.js, Express, MySQL)**

**Cấu trúc thư mục dự án**

project-folder/

│

├── config/

│ └── db.js # Kết nối cơ sở dữ liệu MySQL

├── controllers/

│ ├── authController.js # Xử lý xác thực người dùng

│ ├── productController.js # Xử lý quản lý sản phẩm

│ ├── userController.js # Xử lý quản lý người dùng

│ └── orderController.js # Xử lý đơn hàng (dự kiến)

├── middleware/

│ └── authMiddleware.js # Xác thực token và phân quyền

├── routes/

│ ├── authRoutes.js # Đường dẫn cho xác thực

│ ├── productRoutes.js # Đường dẫn cho sản phẩm

│ └── userRoutes.js # Đường dẫn cho người dùng (dự kiến)

├── server.js # File chính khởi chạy server

├── .env # Biến môi trường (chuỗi kết nối DB, secret key)

└── package.json

**Đã hoàn thành các API:**

**Xác thực người dùng (Authentication)**

**Đăng ký** (POST /api/auth/register)

**Đăng nhập** (POST /api/auth/login)

**Quản lý sản phẩm (Admin Only)**

**Thêm sản phẩm mới** (POST /api/products)

**Lấy danh sách sản phẩm** (GET /api/products)

**Lấy chi tiết sản phẩm** (GET /api/products/:id)

**Cập nhật sản phẩm** (PUT /api/products/:id)

**Xóa sản phẩm** (DELETE /api/products/:id)

**Middleware cho phân quyền**

**verifyToken:** Xác thực token JWT.

**isAdmin:** Kiểm tra vai trò admin.

**Các Chức Năng Backend Cần Xây Dựng Tiếp Theo**

**Quản lý Người Dùng (Admin)**

Xem danh sách người dùng (GET /api/users)

Cập nhật thông tin người dùng (PUT /api/users/:id)

Xóa người dùng (DELETE /api/users/:id)

**Quản lý Giỏ Hàng (User)**

Thêm sản phẩm vào giỏ hàng (POST /api/cart)

Xem giỏ hàng của người dùng (GET /api/cart/:userId)

Cập nhật giỏ hàng (PUT /api/cart/:cartId)

Xóa sản phẩm khỏi giỏ hàng (DELETE /api/cart/:cartId)

**Quản lý Đơn Hàng (User)**

Tạo đơn hàng mới (POST /api/orders)

Xem danh sách đơn hàng của người dùng (GET /api/orders/:userId)

Cập nhật trạng thái đơn hàng (Admin) (PUT /api/orders/:id)

**Thanh Toán (User)**

Xử lý thanh toán (POST /api/payment)

Lấy thông tin thanh toán (GET /api/payment/:orderId)

**Kế Hoạch Tiếp Theo**

Hoàn thiện các API cho **giỏ hàng**, **đơn hàng**, và **thanh toán**.

Tích hợp frontend với các API đã xây dựng.

Tối ưu hóa hiệu suất và bảo mật.

**TUẦN 2**

**1. Quản lý Người Dùng (User Management)**

* Cập nhật thông tin người dùng:
  + Đã cho phép cập nhật họ tên, địa chỉ, vai trò, và cả email và mật khẩu cho người dùng.
  + Xử lý kiểm tra email trùng lặp trước khi cập nhật.
  + Mã hóa mật khẩu mới khi cập nhật.
* Xóa người dùng:
  + Sửa lỗi không thể xóa người dùng trước đó.
  + Đảm bảo xóa người dùng chỉ khi có quyền admin.

**2. Quản lý Giỏ Hàng (Cart Management)**

* Thiết kế và xây dựng lại các API cho giỏ hàng dựa trên cấu trúc bảng gio\_hang mà bạn cung cấp:
  + Lấy giỏ hàng của người dùng.
  + Thêm sản phẩm vào giỏ hàng.
  + Cập nhật số lượng sản phẩm trong giỏ hàng.
  + Xóa sản phẩm khỏi giỏ hàng.
* Sửa các lỗi liên quan đến cột trong bảng gio\_hang:
  + Điều chỉnh truy vấn để phù hợp với cấu trúc bảng hiện tại của bạn.
* Cấu hình lại cartRoutes.js và cập nhật server.js để tích hợp các route mới.

**3. Kiểm tra và Debug**

* Sử dụng Postman để test các API:
  + Kiểm tra các API liên quan đến người dùng và giỏ hàng.
  + Đảm bảo các API yêu cầu token xác thực hoạt động đúng cách.
* Khắc phục các lỗi 500 Internal Server Error, lỗi truy vấn SQL, và các vấn đề khác.

**19/11/2024**

Sửa lỗi, test lại các chức năng, api.

**22/11/2024**

Chức năng quản lý hóa đơn thanh toán, đưa giỏ hàng xuống hóa đơn thanh toán của người dùng

Cơ bản hoàn thành backend, sẽ sữa đổi bổ sung sau.

Kết thúc tuần 2.

**TUẦN 3**

Làm giao diện trang chủ, lấy api tất cả các sản phẩm lên thử trang chủ, sữa đổi bổ sung.

Làm trng chủ giao diện sản phẩm.

Lỗi khá nhiều phần đăng nhập đăng ký.  
Làm lại phần đăng nhập đăng ký

TUẦN 4

Sữa chữa lại Api các chức năng.

Cập nhật

**TUẦN 5**

Cập nhật được một số chức năng, sữa chữa hoàn thiện.