ĐẠI HỌC PHENIKAA KHOA CÔNG NGHỆ THÔNG TIN

**\*\*\*\*\***

****

**BÀI TẬP LỚN**

**MÔN HỌC: An toàn và bảo mật thông tin**

**ĐỀ TÀI: Bảo mật web bán hàng**

**Sinh viên thực hiện: NGUYỄN TRỌNG DƯƠNG-23010227**

**Sinh viên thực hiện : Nguyễn Hữu Huy -23010240**

**Giáo viên hướng dẫn:Đoàn Trung Sơn**

**Lớp:CNTT-3**

**MỤC LỤC**

**Mục 1. Sử dụng framework Laravel**

**Mục 2. Ứng dụng được minh họa ít nhất 03 đối tượng (Ví dụ: bài toán quản bán đồ chơi: Product, Customer, Product-Customer)**

**Mục 3. Có chức năng định danh và xác thực (User)**

**Mục 4. Xây dựng CRUD cho ít nhất 01 đối tượng (Đối tượng ngoài User) (Ví dụ: Xây CRUD cho Product-Customer)**

**Mục 5. Project phải đảm bảo các yêu cầu Security như trong phần học và trình bầy trên lớp (ví dụ csrf, xss, data validation, authentication, authorisation, session, cookies, sql statement...)**

**Mục 6. Cập nhật ReadMe.md với các nội dung trên**

**Mục 1 : web sử dụng framework laravel**

**Mục 2. Ứng dụng được minh họa ít nhất 03 đối tượng (Ví dụ: bài toán quản bán đồ chơi: Product, Customer, Product-Customer)**

1. Đối tượng: Product (Sản phẩm) Tên bảng: tbl\_product

Chức năng: Lưu trữ thông tin các sản phẩm được bán trên hệ thống. Các thuộc tính chính:

product\_id (int, khóa chính): Mã sản phẩm tự tăng. category\_id (int): Mã danh mục sản phẩm. brand\_id (int): Mã thương hiệu sản phẩm. product\_name (string): Tên sản phẩm. product\_desc (text): Mô tả ngắn về sản phẩm.

product\_content (text): Nội dung chi tiết về sản phẩm. product\_price (string): Giá sản phẩm.

product\_image (string): Đường dẫn ảnh sản phẩm. product\_status (int): Trạng thái hiển thị (1: hiển thị, 0: ẩn). created\_at, updated\_at (timestamp): Thời gian tạo và cập nhật.

1. Đối tượng: Order (Đơn hàng)

Tên bảng: tbl\_order

Chức năng: Quản lý thông tin các đơn hàng mà khách hàng đã đặt.

Các thuộc tính chính:

order\_id (int, khóa chính): Mã đơn hàng tự tăng. customer\_id (int): Mã khách hàng đặt hàng. shipping\_id (int): Mã thông tin giao hàng. payment\_id (int): Mã phương thức thanh toán. order\_total (float): Tổng giá trị đơn hàng.

order\_status (string): Trạng thái đơn hàng (ví dụ: "Đang chờ xử lý", "Đã

xác nhận", "Đã giao hàng").

created\_at, updated\_at (timestamp): Thời gian tạo và cập nhật.

1. Đối tượng: Payment (Thanh toán) Tên bảng: tbl\_payment

Chức năng: Lưu trữ thông tin về phương thức và trạng thái thanh toán của đơn hàng.

Các thuộc tính chính:

payment\_id (int, khóa chính): Mã thanh toán tự tăng.

payment\_method (string/int): Phương thức thanh toán (ví dụ: chuyển khoản, tiền mặt...).

payment\_status (string): Trạng thái thanh toán (ví dụ: "Chưa thanh toán", "Đã thanh toán").

created\_at, updated\_at (timestamp): Thời gian tạo và cập nhật.

Mối liên hệ giữa các đối tượng

Product thuộc về một Category và một Brand. Order liên kết với Customer, Shipping, Payment. Payment liên kết với Order thông qua payment\_id.

**Mục 3. Có chức năng định danh và xác thực (User)**

**Chức năng xác thực người dùng trong dự án**

1. **Xác thực người dùng quản trị (Admin)**
   * **Mục đích:**

Bảo vệ khu vực quản trị (dashboard), chỉ cho phép những người đã đăng nhập với quyền quản trị truy cập các chức năng như quản lý sản phẩm, đơn hàng, danh mục, thương hiệu, v.v.

* + **Cách hoạt động:**
    - Khi truy cập trang quản trị, hệ thống kiểm tra phiên đăng

nhập.

+ Sử dụng mã hóa MD5 trong việc xác thực mật khẩu

* + - Nếu chưa đăng nhập, người dùng sẽ được chuyển hướng tới

trang đăng nhập quản trị.

* + - Sau khi đăng nhập thành công, thông tin người dùng sẽ hiển thị ở góc trên bên phải dashboard (ảnh đại diện, tên người dùng, menu hồ sơ, cài đặt, đăng xuất).
    - Có chức năng đăng xuất để kết thúc phiên làm việc.

1. **Xác thực người dùng khách hàng (Customer)**
   * **Mục đích:**

Đảm bảo chỉ những khách hàng đã đăng ký và đăng nhập mới có

thể thực hiện các chức năng như đặt hàng, xem lịch sử mua hàng, cập nhật thông tin cá nhân.

* + **Cách hoạt động:**
    - Khách hàng cần đăng ký tài khoản với các thông tin cơ bản (họ tên, email, mật khẩu, v.v.).
    - Sử dụng mã hóa MD5 cho việc xác thực mật khẩu
    - Sau khi đăng ký, khách hàng có thể đăng nhập vào hệ thống.
    - Khi đăng nhập thành công, hệ thống lưu phiên làm việc để khách hàng có thể mua hàng, thanh toán, xem đơn hàng của mình.
    - Có chức năng đăng xuất để bảo mật tài khoản.

1. **Ý nghĩa**
   * Đảm bảo an toàn cho hệ thống, tránh truy cập trái phép vào khu vực quản trị.
   * Bảo vệ thông tin cá nhân và đơn hàng của khách hàng.
   * Cá nhân hóa trải nghiệm người dùng khi mua sắm và quản lý đơn

hàng.

**Mục 4. Xây dựng CRUD cho ít nhất 01 đối tượng (Đối tượng ngoài User) (Ví dụ: Xây CRUD cho Product-Customer)**

1. **Giao diện và layout trang quản trị (Admin Dashboard)**
   1. **Vị trí và vai trò**
      * File layout chính của trang quản trị là: [admin\_layout.blade.php](vscode-file://vscode-app/c:/Users/ACER/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)
      * File này định nghĩa giao diện tổng thể cho toàn bộ khu vực quản trị của hệ thống bán hàng.
   2. **Cấu trúc layout**
      * **Header:**
        + Hiển thị logo, thanh tìm kiếm, thông tin người dùng (ảnh đại diện, tên, menu hồ sơ, cài đặt, đăng xuất).
      * **Sidebar (Thanh điều hướng bên trái):**
        + Chứa menu quản lý các chức năng chính như:
          - Tổng quan (Dashboard)
          - Danh mục sản phẩm (thêm, liệt kê)
          - Thương hiệu sản phẩm (thêm, liệt kê)
          - Sản phẩm (thêm, liệt kê)
      * **Main Content:**
        + Khu vực hiển thị nội dung động của từng trang con, sử dụng cú pháp @yield('admin\_content') để các view con truyền nội dung vào.
      * **Footer:**
        + Hiển thị thông tin bản quyền và nhà thiết kế giao diện.
   3. **Tích hợp CSS, JS và responsive**
      * Sử dụng các file CSS như: bootstrap.min.css, style.css, style- responsive.css, font-awesome.css để đảm bảo giao diện đẹp và tương thích nhiều thiết bị.
      * Tích hợp các thư viện JavaScript như: jQuery, Morris.js (vẽ biểu đồ), Monthly.js (lịch), giúp giao diện sinh động và có nhiều chức năng quản trị trực quan.
   4. **Ý nghĩa**
      * Giúp quản trị viên thao tác dễ dàng với hệ thống: thêm, sửa, xóa, xem danh sách sản phẩm, danh mục, thương hiệu.
      * Giao diện trực quan, dễ sử dụng, thuận tiện cho việc quản lý dữ liệu và theo dõi hoạt động bán hàng.

**Mục 5. Project phải đảm bảo các yêu cầu Security như trong phần học và trình bầy trên lớp (ví dụ csrf, xss, data validation, authentication, authorisation, session, cookies, sql statement...).**

**+ Sử dụng CSRF trong dự án**

1. **Ý nghĩa**
   * CSRF (Cross-Site Request Forgery) là một kỹ thuật bảo mật giúp ngăn chặn các cuộc tấn công giả mạo yêu cầu từ bên ngoài vào hệ thống.
   * Laravel tự động bảo vệ các form khỏi tấn công CSRF bằng cách yêu cầu mỗi form gửi dữ liệu phải có một token hợp lệ.

+ Sử dụng phân quyền authentication middleware auth để bảo vệ luồng giữa customers và admin

* Mục đích
  + Ngăn truy cập trái phép vào các trang nhạy cảm (dashboard, quản trị) và phân tách quyền giữa admin và customer.
  + Bảo vệ dữ liệu và hành động nhạy cảm (quản lý sản phẩm, xem đơn hàng, thanh toán).
  + Giảm rủi ro truy cập trực tiếp bằng URL.
* Nguyên lý hoạt động
  + Middleware kiểm tra trạng thái xác thực (Auth/Session) trước khi cho phép request tiếp tục.
  + Nếu chưa xác thực, middleware chuyển hướng về trang đăng nhập (hoặc trả 401 cho API).
  + Sử dụng guards riêng (ví dụ auth:admin và auth:customer) để phân biệt loại tài khoản.
* Cấu hình chính (tóm tắt)
  + Đăng ký middleware alias trong app/Http/Kernel.php (mặc định có 'auth').
  + Định nghĩa route group bảo vệ theo guard:
    - admin: Route::middleware(['auth:admin'])->group(...)
    - customer: Route::middleware(['auth:customer'])->group(...)

1. **Cách sử dụng trong dự án Ví dụ 1: Form thêm sản phẩm**

<?php

<form action="{{URL::to('/save-

product')}}" method="POST" enctype="multipart/form-data"> @csrf

<div class="form-group">

<label for="product\_name">Tên sản phẩm</label>

<input type="text" name="product\_name" class="form- control" required>

</div>

<!-- Các trường nhập liệu khác -->

<button type="submit" class="btn btn- primary">Thêm sản phẩm</button>

</form>

**Ví dụ 2: Form đăng ký khách hàng**

<?php

<form action="{{URL::to('/register-customer')}}" method="POST"> @csrf

<div class="form-group">

<label for="customer\_name">Họ tên</label>

<input type="text" name="customer\_name" class="form- control" required>

</div>

<div class="form-group">

<label for="customer\_email">Email</label>

<input type="email" name="customer\_email" class="form- control" required>

</div>

<div class="form-group">

<label for="customer\_password">Mật khẩu</label>

<input type="password" name="customer\_password" class="form- control" required>

</div>

<button type="submit" class="btn btn-success">Đăng ký</button>

</form>



1. **Kết luận**

Trong tất cả các form gửi dữ liệu POST lên server, dự án đều sử

dụng @csrf để đảm bảo an toàn, đúng theo yêu cầu bảo mật đã học trên lớp. Nếu thiếu token này, Laravel sẽ tự động từ chối xử lý request.

