

HƯỚNG DẪN ĐỌC HIỂU CODE - PHẦN ADMIN

MỤC LỤC

1. Tổng Quan
 2. Cấu Trúc File Admin
 3. Session Management
 4. **Chi Tiết Login** (Tóm tắt : import function khác và check xem có hoặc không có session chưa, có thì vào còn không có thì check tiếp xem có cookie không, nếu có thì vào mà không có thì ở yên đấy, rồi tạo biến thông báo msg, tạo function đăng nhập, bấm hàm password rồi gửi form đăng nhập cho query để tra xem có trên sql không ?, nếu có thì hiện admin_login = yes và lưu email đăng nhập vào admin_email để hiện ở đầu trang đồng thời điều hướng sang trang categories : danh mục) password temp dư thừa
 5. **Chi Tiết Logout**(Tóm tắt : truy cập phiên đang hoạt động và xóa 2 dòng LOGIN VÀ EMAIL RỒI CHUYỂN HƯỚNG PAGE)
 6. **Chi Tiết TopNav** (Tóm tắt: Check đã đăng nhập chưa, rồi thêm topnav)
 7. **Chi Tiết Users** (Tóm tắt: Function hoạt động theo hướng khi ấn type thì type mới có và type khác 0, thì hàm điều kiện chạy, clean type và chạy xuống type = delet vì tôi set nó thế, rồi chạy delete cái cột id user(ép về int) rồi load trang để xóa type)
 8. **Chi Tiết Orders** (Tóm tắt : check bảo mật -> nếu có trạng thái là đã Hủy hoặc Trả Hàng -> Tự động +1 số lượng sách vào kho sách dùng lệnh query rồi reload lại trang rồi thoát lệnh.....-> Tiếp theo hiển thị đơn hàng -> join các bảng data để lấy dữ liệu, thêm bảng order_detail để xem giá sách + số lượng ngày thuê, thêm bảng book để lấy thông tin sách gì, + bảng order_status để lấy trạng thái đơn hàng, sắp xếp từ mới cho đến cũ..-> tiếp theo function nếu trạng thái đơn không phải là Hủy hay Trả Hàng thì -> canchange = true, nếu phải thì canchange = false(boolean), nếu false thì ngay tiếp theo có hàm không cho render form -> không thay đổi được)
 9. **Chi Tiết Categories** (Tóm tắt:Function giống User nhưng có 2 tính năng chính -> 1 Thay đổi trạng thái 0 - 1 bật tắt danh mục. 2 xóa danh mục i hết xóa user. Còn 2 tính năng khác là Edit và Add categories -> gửi đến trang manageCategories.php)
 10. **Chi Tiết Manage Categories** (Tóm tắt : Function check xem id được lấy có id không ? nếu là số từ 1->xx thì xác định là edit function trong edit tìm dùng lệnh sql-> nếu tìm thấy thì cập nhật , 0 tìm thấy thì đá về lại categories , còn id = 0 -> add categories. Cả 2 cái điều liên quan đến danh mục nên sẽ check trùng, chạy lệnh query xem có danh mục trùng với danh mục đang edit/add không check >0 xem có mục nào trùng không, dùng if !\$id cho thêm và check logic id != id trong sql cho sửa, sau đó hàm chạy SQL if else)
 11. **Chi Tiết Books**
 12. **Chi Tiết ManageBooks**
-

TỔNG QUAN

Hệ thống Admin Panel cho phép quản trị viên quản lý toàn bộ hệ thống Book Rental:

-  Đăng nhập vào hệ thống quản trị
-  Quản lý sách (thêm, sửa, xóa, thay đổi trạng thái)
-  Quản lý danh mục sách
-  Quản lý đơn hàng
-  Quản lý người dùng
-  Xem phản hồi từ khách hàng
-  Quản lý ngày trả sách

Công nghệ sử dụng: - **Backend:** PHP - **Database:** MySQL - **Frontend:** HTML, CSS, JavaScript, Bootstrap 5, MDB (Material Design for Bootstrap)

CẤU TRÚC FILE ADMIN

Thư Mục Admin

Book-rental/

└─ Admin/	
├─ login.php	# Đăng nhập admin ★ (Đã giải thích)
├─ logout.php	# Đăng xuất admin ★ (Đã giải thích)
├─ topNav.php	# Header/Navigation chung cho admin ★ (Đã giải thích)
├─ categories.php	# Quản lý danh mục sách ★ (Đã giải thích)
├─ manageCategories.php	# Thêm/Sửa danh mục ★ (Đã giải thích)
├─ books.php	# Danh sách sách ★ (Đã giải thích)
├─ manageBooks.php	# Thêm/Sửa sách ★ (Đã giải thích)
├─ orders.php	# Quản lý đơn hàng ★ (Đã giải thích)
├─ returnDate.php	# Quản lý ngày trả sách ★ (Đã giải thích)
├─ users.php	# Quản lý người dùng ★ (Đã giải thích)
├─ feedback.php	# Xem phản hồi từ khách hàng ★ (Đã giải thích)
├─ css/	
├─ admin.css	# CSS riêng cho admin
├─ js/	
├─ admin.js	# JavaScript riêng cho admin

Lưu ý: Tất cả các file PHP chính đã được giải thích chi tiết trong tài liệu này.

SESSION MANAGEMENT

Các Session Variable

`$_SESSION['ADMIN_LOGIN']`

- **Giá trị:** 'yes' (string)
- **Set khi:** Đăng nhập thành công
- **Dùng để:** Kiểm tra admin đã đăng nhập chưa
- **Unset khi:** Logout
- **Kiểm tra trong:** Tất cả các trang admin (trừ login.php)

`$_SESSION['ADMIN_email']`

- **Giá trị:** Email của admin (string)
- **Set khi:** Đăng nhập thành công
- **Dùng để:** Hiện thị email admin trên navigation bar
- **Unset khi:** Logout

Cơ Chế Bảo Vệ Trang Admin

Tất cả các trang admin (trừ login.php) đều được bảo vệ bằng cách kiểm tra session:

Cách 1: Kiểm tra trực tiếp trong file

```
if (!isset($_SESSION['ADMIN_LOGIN']) || $_SESSION['ADMIN_LOGIN'] == ' ') {  
    header('Location: login.php');  
    exit;  
}
```

Cách 2: Kiểm tra trong topNav.php

```
// topNav.php được require ở đầu mỗi trang admin  
if (isset($_SESSION['ADMIN_LOGIN']) && $_SESSION['ADMIN_LOGIN'] != ' ') {  
    // Cho phép truy cập  
} else {  
    header('location:login.php');  
    die();  
}
```

Lưu ý: - Nếu chưa đăng nhập → Tự động redirect về login.php - Session được kiểm tra ở cả 2 nơi để đảm bảo an toàn

CHI TIẾT LOGIN

File: Admin/login.php

Mục đích: Xử lý đăng nhập cho quản trị viên

Flow Hoạt Động

[Admin truy cập login.php]



[Hiển thị form đăng nhập]



[Admin nhập email và password]



[Submit form (POST)]



[Kiểm tra email và password trong database]



[Nếu đúng → Set session và redirect đến categories.php]

[Nếu sai → Hiển thị lỗi "Invalid Username/Password"]

Code Chi Tiết

1. Include các file cần thiết

```
require(__DIR__ . '/../config/connection.php');
```

```
require(__DIR__ . '/../includes/function.php');
```

Giải thích: - connection.php: Kết nối database và khởi động session - function.php: Các function hỗ trợ (getSafeValue, ...)

2. Khởi tạo biến

```
$msg = $passwordTemp = '';
```

Giải thích: - \$msg: Lưu thông báo lỗi (nếu có) - \$passwordTemp: Biến tạm (có thể không dùng)

3. Xử lý form đăng nhập

```
if (isset($_POST['submit'])) {  
    $email = getSafeValue($con, $_POST['email']);  
    $password = getSafeValue($con, $_POST['password']);  
    $sql = "select * from admin where email='$email' and  
password='$password'";  
    $res = mysqli_query($con, $sql);  
    $count = mysqli_num_rows($res);  
  
    if ($count > 0) {  
        $_SESSION['ADMIN_LOGIN'] = 'yes';  
        $_SESSION['ADMIN_email'] = $email;  
        header('location:categories.php');  
        die();  
    } else {  
        $msg = "Invalid Username/Password";  
    }  
}
```

Giải thích từng bước:

1. **Kiểm tra form đã submit:**

```
if (isset($_POST['submit']))
```

- Kiểm tra nút "Login" đã được click chưa

2. **Lấy và làm sạch dữ liệu:**

```
$email = getSafeValue($con, $_POST['email']);  
$password = getSafeValue($con, $_POST['password']);
```

- Lấy email và password từ form
- Sử dụng getSafeValue() để bảo mật (chống SQL injection, XSS)

3. **Query database:**

```
$sql = "select * from admin where email='$email' and  
password='$password'";  
$res = mysqli_query($con, $sql);  
$count = mysqli_num_rows($res);
```

- Tìm admin trong database với email và password khớp
- Đếm số dòng kết quả

4. **Nếu tìm thấy (đăng nhập thành công):**

```
if ($count > 0) {  
    $_SESSION['ADMIN_LOGIN'] = 'yes';  
    $_SESSION['ADMIN_email'] = $email;  
    header('location:categories.php');  
    die();  
}
```

- Set 2 session variables:
 - \$_SESSION['ADMIN_LOGIN'] = 'yes': Đánh dấu đã đăng nhập
 - \$_SESSION['ADMIN_email'] = \$email: Lưu email để hiển thị
- Redirect đến categories.php (trang quản lý danh mục)
- die(): Dừng script để đảm bảo không có code nào chạy sau redirect

5. **Nếu không tìm thấy (đăng nhập thất bại):**

```
else {  
    $msg = "Invalid Username/Password";  
}
```

- Gán thông báo lỗi vào \$msg
- Thông báo này sẽ được hiển thị trên form

4. *Hiển thị form đăng nhập*

```
<form class="mx-1 mx-md-4" method="post">  
    <div class="d-flex align-items-center mb-4">  
        <i class="fas fa-envelope fa-lg me-3 fa-fw"></i>
```

```

        <div class="form-floating flex-fill">
            <input type="email" name="email" class="form-control" id="email"
                placeholder="name@example.com" required />
            <label for="email">Email address</label>
        </div>
    </div>
    <div class="d-flex align-items-center mb-1">
        <i class="fas fa-lock fa-lg me-3 fa-fw"></i>
        <div class="form-floating flex-fill">
            <input type="password" name="password" class="form-control"
id="Password"
                placeholder="Password" required />
            <label for="Password">Password</label>
        </div>
    </div>
    <div class="mt-2 mb-1 d-flex justify-content-center field_error">
        <?php echo $msg ?>
    </div>
    <div class="d-flex justify-content-center mt-3 mb-3 mb-lg-4">
        <button type="submit" name="submit" class="btn btn-primary
btn-lg">Login</button>
    </div>
</form>

```

Giải thích: - Form sử dụng method POST - Có 2 input: - email: Type email (có validation tự động) - password: Type password (ẩn ký tự) - Hiện thị thông báo lỗi \$msg (nếu có) - Nút submit có name="submit" để trigger xử lý PHP

Database Schema

Bảng admin

Lưu thông tin quản trị viên

Cột	Kiểu	Mô tả
id	int(11)	ID admin (Primary Key, Auto Increment)
email	varchar(50)	Email đăng nhập (Unique)
password	varchar(255)	Mật khẩu (lưu dạng plain text - không an toàn)

Lưu ý bảo mật: - ⚠ Password hiện tại được lưu dạng plain text (không hash) - ⚠ Nên sử dụng password hashing (md5, password_hash, ...) trong tương lai - ⚠ Query sử dụng string concatenation (có nguy cơ SQL injection nếu không dùng getSafeValue)

Sử dụng trong: - login.php: SELECT (kiểm tra đăng nhập)

So Sánh Với Customer Login

Đặc điểm	Admin Login	Customer Login
File	Admin/login.php	pages/SignIn.php
Bảng database	admin	users
Session variables	ADMIN_LOGIN, ADMIN_email	USER_LOGIN, USER_ID, USER_NAME
Password hash	✗ Plain text	✓ MD5
Remember Me	✗ Không có	✓ Có
Redirect sau login	categories.php	index.php hoặc checkout
Validation	Chỉ kiểm tra email/password	Có validation đầy đủ

Lưu Ý Quan Trọng

- Bảo mật:**
 - ✓ Sử dụng `getSafeValue()` để chống SQL injection
 - ⚠ Password không được hash (nên cải thiện)
 - ✓ Session được kiểm tra ở nhiều nơi
- Redirect:**
 - Sau khi đăng nhập thành công → Redirect đến `categories.php`
 - Sử dụng `header()` và `die()` để đảm bảo redirect hoạt động
- Error Handling:**
 - Hiển thị thông báo lỗi rõ ràng
 - Không tiết lộ thông tin chi tiết về lỗi (bảo mật)

CHI TIẾT LOGOUT

File: Admin/logout.php

Mục đích: Xóa session và đăng xuất admin

Flow Hoạt Động

```
[Admin click Logout]
↓
[logout.php được gọi]
↓
[Xóa session variables]
↓
[Redirect về login.php]
```

Code Chi Tiết

```
<?php
    session_start();
```

```
unset($_SESSION['ADMIN_LOGIN']);
unset($_SESSION['ADMIN_EMAIL']);
header('location:login.php');
die();
?>
```

Giải thích từng dòng:

1. Khởi động session:

```
session_start();
```

- Bắt đầu hoặc tiếp tục session hiện tại
- Cần thiết để có thể unset session variables

2. Xóa session variables:

```
unset($_SESSION['ADMIN_LOGIN']);
unset($_SESSION['ADMIN_EMAIL']);
```

- Xóa ADMIN_LOGIN: Đánh dấu đã logout
- Xóa ADMIN_EMAIL: Xóa thông tin email
- **Lưu ý:** Có sự khác biệt tên biến:
 - Khi set: `$_SESSION['ADMIN_email']` (chữ thường 'email')
 - Khi unset: `$_SESSION['ADMIN_EMAIL']` (chữ hoa 'EMAIL')
 - ⚠️ Nên thống nhất tên biến để tránh lỗi

3. Redirect về trang login:

```
header('location:login.php');
die();
```

- Redirect về login.php
- `die()`: Dừng script để đảm bảo không có code nào chạy sau redirect

So Sánh Với Customer Logout

Đặc điểm	Admin Logout	Customer Logout
File	Admin/logout.php	pages/logout.php
Session variables	ADMIN_LOGIN, ADMIN_EMAIL	USER_LOGIN, USER_ID, USER_NAME, BeforeCheckoutLogin
Cookie	✗ Không xóa cookie	✓ Xóa cookie Remember Me
Redirect	login.php	index.php
Database	✗ Không thao tác	✓ Xóa token Remember Me

Vấn Đề Và Cải Thiện

1. Inconsistency trong tên session variable

Vấn đề: - Khi set: `$_SESSION['ADMIN_email']` (chữ thường) - Khi unset: `$_SESSION['ADMIN_EMAIL']` (chữ hoa)

Giải pháp:

```
// Nên thống nhất:  
unset($_SESSION['ADMIN_email']); // Thay vì ADMIN_EMAIL
```

2. Thiếu require connection và function

Vấn đề: - File không require `connection.php` và `function.php` - Nếu cần xóa cookie hoặc thao tác database sẽ không có sẵn

Giải pháp (nếu cần):

```
require(__DIR__ . '/../config/connection.php');  
require(__DIR__ . '/../includes/function.php');
```

Lưu ý: Hiện tại file `logout` đơn giản nên không cần thiết, nhưng nếu muốn thêm tính năng (như xóa cookie) thì cần require.

Lưu Ý Quan Trọng

1. Đơn giản:

- File `logout` rất đơn giản, chỉ xóa session và redirect
- Không có logic phức tạp

2. Bảo mật:

- ☒ Xóa tất cả session variables
- ☒ Redirect về trang login
- ☒ Sử dụng `die()` để đảm bảo không có code nào chạy sau

3. Session:

- Session được xóa hoàn toàn
- Admin phải đăng nhập lại để truy cập

CHI TIẾT TOPNAV

File: `Admin/topNav.php`

Mục đích: Header/Navigation chung cho tất cả trang admin

Flow Hoạt Động

[Trang admin được load]



[require topNav.php]



[Kiểm tra session ADMIN_LOGIN]
└─> Có session → Hiển thị navigation
└─> Không có session → Redirect đến login.php

Code Chi Tiết

1. Include và Kiểm Tra Session

```
require_once(__DIR__ . '/../config/connection.php');
require_once(__DIR__ . '/../includes/function.php');
if (isset($_SESSION['ADMIN_LOGIN']) && $_SESSION['ADMIN_LOGIN'] != ' ') {
} else {
    header('location:login.php');
    die();
}
```

Giải thích: - require_once: Include connection và function (chỉ 1 lần, tránh duplicate) - Kiểm tra session ADMIN_LOGIN: - Nếu có và không rỗng → Cho phép tiếp tục - Nếu không có hoặc rỗng → Redirect về login.php - Đây là cơ chế bảo vệ chính cho tất cả trang admin

2. HTML Head

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no" />
    <title>Admin Panel</title>
    <!-- CSS files -->
</head>
```

Giải thích: - Cấu trúc HTML chuẩn - Include các CSS: - Font Awesome (icons) - Google Fonts Roboto - MDB (Material Design for Bootstrap) - Custom admin.css

3. Navigation Bar

```
<nav class="navbar navbar-expand-lg sticky-top navbar-light bg-light">
    <div class="container-fluid">
        <!-- Logo -->
        <a class="navbar-brand" href="../pages/index.php">
            
        </a>

        <!-- Menu Items -->
        <ul class="navbar-nav me-auto mb-2 mb-lg-0">
            <li class="nav-item">
                <a class="nav-link" href="categories.php">Categories</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="books.php">Books list</a>
            </li>
```

```

        <li class="nav-item">
            <a class="nav-link" href="orders.php">Orders</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="returnDate.php">Return Date</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="users.php">Users</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="feedback.php">Feedbacks</a>
        </li>
    </ul>

    <!-- User Dropdown -->
    <div class="d-flex align-items-center nav-item">
        <?php
            $userName = $_SESSION['ADMIN_email'];
            echo '<div class="btn-group shadow-0">
                <button type="button" class="btn btn-light
dropdown-toggle"
                    data-mdb-toggle="dropdown">' . $userName .
'</button>
                    <ul class="dropdown-menu">
                        <li><a class="dropdown-item"
href="logout.php">Logout</a></li>
                    </ul>
                </div>';
        ?>
    </div>
</div>
</nav>

```

Giải thích: - **Logo:** Link về trang chủ customer (../pages/index.php) - **Menu Items:** 6 menu chính: - Categories: Quản lý danh mục - Books list: Danh sách sách - Orders: Quản lý đơn hàng - Return Date: Ngày trả sách - Users: Quản lý người dùng - Feedbacks: Phản hồi - **User Dropdown:** - Hiện thị email admin từ \$_SESSION['ADMIN_email'] - Dropdown menu với option Logout

Cách Sử Dụng

Trong các trang admin:

```

// Xử lý logic trước (nếu có)
require('topNav.php');
// HTML content

```

Lưu ý: - topNav.php tự động kiểm tra session - Nếu chưa đăng nhập → Tự động redirect - Các trang admin chỉ cần require('topNav.php') là đã có navigation

Styling

File có CSS inline để tùy chỉnh: - Gradient background cho navbar - Hover effects cho menu items - Responsive design với Bootstrap

CHI TIẾT USERS

File: Admin/users.php

Mục đích: Quản lý danh sách người dùng (customers)

Flow Hoạt Động

[Admin truy cập users.php]



[require topNav.php - Kiểm tra session]



[Xử lý xóa user (nếu có)]



[Lấy danh sách users từ database]



[Hiển thị bảng danh sách users]

Code Chi Tiết

1. Include topNav

```
require('topNav.php');
```

Giải thích: - topNav.php tự động kiểm tra session và hiển thị navigation - Không cần kiểm tra session riêng vì topNav đã làm

2. Xử Lý Xóa User

```
if (isset($_GET['type']) && $_GET['type'] != ' ') {  
    $type = getSafeValue($con, $_GET['type']);  
  
    if ($type == 'delete') {  
        $id = getSafeValue($con, $_GET['id']);  
        $deleteSql = "delete from users where id='$id'";  
        mysqli_query($con, $deleteSql);  
    }  
}
```

Giải thích: - Kiểm tra có action delete không - Lấy ID user cần xóa - Xóa user khỏi database -

Lưu ý: Không có redirect sau khi xóa, trang sẽ reload và hiển thị danh sách mới

3. Lấy Danh Sách Users

```
$sql = "select * from users order by id desc";  
$res = mysqli_query($con, $sql);
```

Giải thích: - Lấy tất cả users - Sắp xếp theo ID giảm dần (user mới nhất trước)

4. Hiện Thị Bảng

```
<table class="table">
  <thead>
    <tr>
      <th>ID</th>
      <th>Name</th>
      <th>Email</th>
      <th>Mobile</th>
      <th>Date of Joining</th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    <?php while ($row = mysqli_fetch_assoc($res)): ?>
      <tr>
        <td><?php echo $row['id'] ?></td>
        <td><?php echo $row['name'] ?></td>
        <td><?php echo $row['email'] ?></td>
        <td><?php echo $row['mobile'] ?></td>
        <td><?php echo $row['doj'] ?></td>
        <td>
          <a class='link-white btn btn-danger px-2 py-1'
            href='?type=delete&id=<?php echo $row['id'] ?>'>Delete</a>
        </td>
      </tr>
    <?php endwhile; ?>
  </tbody>
</table>
```

Giải thích: - Hiện thị bảng với các cột: ID, Name, Email, Mobile, Date of Joining - Mỗi dòng có nút Delete - Click Delete → Gọi lại trang với ?type=delete&id=X

Database Schema

Bảng users

Cột	Kiểu	Mô tả
id	int(11)	ID user (Primary Key)
name	varchar(80)	Tên user
email	varchar(50)	Email
mobile	bigint(20)	Số điện thoại
doj	datetime	Ngày tham gia (Date of Join)
password	varchar(255)	Mật khẩu

Sử dụng trong: - users.php: SELECT (lấy danh sách), DELETE (xóa user)

Lưu Ý Quan Trọng

1. **Bảo mật:**
 -  Sử dụng `getSafeValue()` để bảo mật
 -  Không có xác nhận trước khi xóa (có thể thêm confirm dialog)
 -  Xóa user sẽ xóa luôn các đơn hàng liên quan? (Cần kiểm tra foreign key)
 2. **Chức năng:**
 - Chỉ có chức năng xem và xóa
 - Không có chức năng sửa user
 - Không có chức năng thêm user (user tự đăng ký)
 3. **UI/UX:**
 - Bảng đơn giản, dễ đọc
 - Nút Delete màu đỏ để cảnh báo
-

CHI TIẾT ORDERS

File: `Admin/orders.php`

Mục đích: Quản lý đơn hàng - Xem và cập nhật trạng thái đơn hàng

Flow Hoạt Động

[Admin truy cập `orders.php`]



[Kiểm tra session]



[Xử lý cập nhật trạng thái đơn hàng (nếu có POST)]

- Nếu hủy/trả → Tăng lại số lượng sách
- Cập nhật `order_status`



[Lấy danh sách đơn hàng từ database]



[Hiển thị bảng với form cập nhật trạng thái]

Code Chi Tiết

1. Kiểm Tra Session

```
if (!isset($_SESSION['ADMIN_LOGIN']) || $_SESSION['ADMIN_LOGIN'] == ' ') {  
    header('Location: login.php');  
    exit;  
}
```

Giải thích: - Kiểm tra session trước khi xử lý logic - Đảm bảo chỉ admin mới truy cập được

2. Xử Lý Cập Nhật Trạng Thái

```
if (isset($_POST['status_id'])) {  
    $orderId = (int)$_POST['orderId'];  
    $statusId = (int)$_POST['status_id'];
```

```

// Nếu đơn hàng bị hủy hoặc trả lại, tăng lại số lượng sách
if (in_array($statusId, [4, 6])) {
    $qtyRes = mysqli_query($con, "SELECT books.id FROM orders
                                JOIN order_detail ON
orders.id=order_detail.order_id
                                JOIN books ON
order_detail.book_id=books.id
                                WHERE
order_detail.order_id=$orderId");
    if ($qtyRow = mysqli_fetch_assoc($qtyRes)) {
        mysqli_query($con, "UPDATE books SET qty = qty + 1 WHERE
id={$qtyRow['id']}");
    }
}

mysqli_query($con, "UPDATE orders SET order_status=$statusId WHERE
id=$orderId");
header('Location: orders.php');
exit;
}

```

Giải thích: - Nhận orderId và statusId từ form POST - **Logic đặc biệt:** Nếu status là 4 (Cancelled) hoặc 6 (Returned): - Tìm sách trong đơn hàng - Tăng lại số lượng sách (qty + 1) - Cập nhật order_status trong bảng orders - Redirect về orders.php để hiển thị cập nhật

3. Lấy Danh Sách Đơn Hàng

```

$res = mysqli_query($con, "SELECT orders.*, name, status_name FROM orders
                        JOIN order_detail ON
orders.id=order_detail.order_id
                        JOIN books ON order_detail.book_id=books.id
                        JOIN order_status ON
orders.order_status=order_status.id
                        ORDER BY date DESC");

```

Giải thích: - JOIN 3 bảng: - orders: Thông tin đơn hàng - order_detail: Chi tiết đơn hàng (để lấy book_id) - books: Thông tin sách (để lấy tên sách) - order_status: Trạng thái đơn hàng (để lấy tên trạng thái) - Sắp xếp theo ngày giảm dần (đơn hàng mới nhất trước)

4. Hiện Thị Bảng Với Form

```

<table class="table">
    <thead>
        <tr>
            <th>OrderID</th>
            <th>Order Date</th>
            <th>Book Name</th>
            <th>Book Price</th>
            <th>Rent Duration</th>
            <th>Address</th>
            <th>Payment Method</th>
            <th>Payment Status</th>

```

```

        <th>Order Status</th>
        <th>Change Order Status</th>
    </tr>
</thead>
<tbody>
    <?php while ($row = mysqli_fetch_assoc($res)):
        $canChange = !in_array($row['status_name'], ['Returned',
'Cancelled']);
        ?>
        <tr>
            <td><?php echo $row['id'] ?></td>
            <td><?php echo $row['date'] ?></td>
            <td><?php echo $row['name'] ?></td>
            <td><?php echo $row['total'] ?></td>
            <td><?php echo $row['duration'] ?> days</td>
            <td><?php echo $row['address'] ?><?php echo $row['address2'] ? ' ',
' . $row['address2'] : ' ' ?></td>
            <td><?php echo $row['payment_method'] ?></td>
            <td><?php echo $row['payment_status'] ?></td>
            <td><?php echo $row['status_name'] ?></td>
            <td>
                <?php if ($canChange): ?>
                <form method="post">
                    <input type="hidden" name="orderId" value="<?php echo
$row['id'] ?>">
                    <select class="form-select" name="status_id">
                        <option value="">Select Status</option>
                        <?php
                            $statusSql = mysqli_query($con, "SELECT * FROM
order_status ORDER BY status_name");
                            while ($statusRow = mysqli_fetch_assoc($statusSql)):
                                ?>
                                <option value="<?php echo $statusRow['id'] ?>"><?php
echo $statusRow['status_name'] ?></option>
                                <?php endwhile; ?>
                            </select>
                            <input type="submit" value="Submit" class="btn
btn-primary mt-2">
                        </form>
                        <?php endif; ?>
                    </td>
                </tr>
            <?php endwhile; ?>
        </tbody>
    </table>

```

Giải thích: - Hiện thị đầy đủ thông tin đơn hàng - **Logic đặc biệt:** Chỉ cho phép thay đổi trạng thái nếu đơn hàng chưa “Returned” hoặc “Cancelled” - Form cập nhật: - Hidden input chứa orderId - Dropdown chọn trạng thái mới (lấy từ bảng order_status) - Nút Submit để cập nhật

Database Schema

Bảng orders

Cột	Kiểu	Mô tả
id	int(11)	ID đơn hàng
user_id	int(11)	ID user
address	varchar(100)	Địa chỉ
address2	varchar(100)	Địa chỉ 2
pin	int(11)	Mã pin
payment_method	varchar(20)	Phương thức thanh toán
total	int(11)	Tổng tiền
payment_status	varchar(20)	Trạng thái thanh toán
order_status	int(11)	Trạng thái đơn hàng (FK → order_status.id)
date	datetime	Ngày đặt hàng
duration	int(11)	Số ngày thuê

Bảng order_status

Cột	Kiểu	Mô tả
id	int(11)	ID trạng thái
status_name	varchar(15)	Tên trạng thái

Các trạng thái: 1. Pending (id=1) 2. Processing (id=2) 3. Shipped (id=3) 4. Cancelled (id=4) 5. Delivered (id=5) 6. Returned (id=6)

Sử dụng trong: - orders.php: SELECT (lấy danh sách), UPDATE (cập nhật trạng thái) - books: UPDATE (tăng số lượng khi hủy/trả)

Lưu Ý Quan Trọng

- Logic nghiệp vụ:**
 - ✓ Tự động tăng số lượng sách khi hủy/trả đơn
 - ✓ Không cho phép thay đổi trạng thái đơn đã hủy/trả
 - ✓ Hiện thị đầy đủ thông tin đơn hàng
 - Bảo mật:**
 - ✓ Type casting cho ID: (int)\$_POST['orderId']
 - ✓ Kiểm tra session trước khi xử lý
 - UI/UX:**
 - Form inline trong bảng, dễ sử dụng
 - Dropdown chọn trạng thái rõ ràng
 - Ẩn form nếu đơn đã hủy/trả
-

CHI TIẾT RETURNDATE

File: Admin/returnDate.php

Mục đích: Xem danh sách đơn hàng đã hủy hoặc đã trả, tính ngày trả dự kiến

Flow Hoạt Động

```
[Admin truy cập returnDate.php]
↓
[Kiểm tra session]
↓
[Xử lý cập nhật trạng thái (nếu có POST)]
↓
[Lấy danh sách đơn hàng đã hủy/trả]
↓
[Tính ngày trả dự kiến cho mỗi đơn]
↓
[Hiển thị bảng]
```

Code Chi Tiết

1. Kiểm Tra Session

```
if (!isset($_SESSION['ADMIN_LOGIN']) || $_SESSION['ADMIN_LOGIN'] == ' ') {
    header('Location: login.php');
    exit;
}
```

Giải thích: - Tương tự orders.php, kiểm tra session trước

2. Xử Lý Cập Nhật Trạng Thái

```
if (isset($_POST['status_id'])) {
    $orderId = (int)$_POST['orderId'];
    $statusId = (int)$_POST['status_id'];

    // Nếu đơn hàng bị hủy hoặc trả lại, tăng lại số lượng sách
    if (in_array($statusId, [4, 6])) {
        $qtyRes = mysqli_query($con, "SELECT books.id FROM orders
                                      JOIN order_detail ON
                                      orders.id=order_detail.order_id
                                      JOIN books ON
                                      order_detail.book_id=books.id
                                      WHERE
                                      order_detail.order_id=$orderId");
        if ($qtyRow = mysqli_fetch_assoc($qtyRes)) {
            mysqli_query($con, "UPDATE books SET qty = qty + 1 WHERE
id={$qtyRow['id']}");
        }
    }

    mysqli_query($con, "UPDATE orders SET order_status=$statusId WHERE
```

```

id=$orderId");
    header('Location: returnDate.php');
    exit;
}

```

Giải thích: - Logic tương tự orders.php - Cập nhật trạng thái và tăng số lượng sách nếu cần

3. Lấy Danh Sách Đơn Hàng Đã Hủy/Trả

```

$sql = "SELECT orders.*, name, status_name
        FROM orders
        JOIN order_detail ON orders.id=order_detail.order_id
        JOIN books ON order_detail.book_id=books.id
        JOIN order_status ON orders.order_status=order_status.id
        WHERE status_name IN ('Cancelled', 'Returned')
        ORDER BY date DESC";
$res = mysqli_query($con, $sql);

```

Giải thích: - Chỉ lấy đơn hàng có status là "Cancelled" hoặc "Returned" - JOIN các bảng để lấy thông tin đầy đủ - Sắp xếp theo ngày giảm dần

4. Tính Ngày Trả Dự Kiến

```

<?php while ($row = mysqli_fetch_assoc($res)):
    // Tính ngày trả dự kiến (Order Date + Duration)
    $orderDate = new DateTime($row['date']);
    $returnDate = clone $orderDate;
    $returnDate->modify('+ ' . $row['duration'] . ' days');
?>
<tr>
    <td>#<?php echo $row['id'] ?></td>
    <td><?php echo $row['date'] ?></td>
    <td><?php echo $returnDate->format('Y-m-d') ?></td>
    <td><?php echo htmlspecialchars($row['name']) ?></td>
    <td><?php echo number_format($row['total']) ?></td>
    <td><?php echo $row['duration'] ?> days</td>
    <td><?php echo htmlspecialchars($row['address']) ?><?php echo
    $row['address2'] ? ' , ' . htmlspecialchars($row['address2']) : '' ?></td>
    <td><?php echo $row['status_name'] ?></td>
</tr>
<?php endwhile; ?>

```

Giải thích: - **Tính toán ngày trả:** - Tạo DateTime từ date (ngày đặt hàng) - Clone để không ảnh hưởng biến gốc - Cộng thêm duration ngày - Format thành Y-m-d - Hiển thị thông tin đơn hàng - Sử dụng htmlspecialchars() để bảo mật output

5. Xử Lý Không Có Dữ Liệu

```

<?php if ($res && mysqli_num_rows($res) > 0): ?>
    <!-- Hiển thị bảng -->
<?php else: ?>
    <tr>
        <td colspan="8" class="text-center">No returned or cancelled orders

```

```
found.</td>
</tr>
<?php endif; ?>
```

Giải thích: - Kiểm tra có dữ liệu không - Nếu không có → Hiển thị thông báo

Database Schema

Sử dụng các bảng giống orders.php: - orders - order_detail - books - order_status

Sử dụng trong: - returnDate.php: SELECT (lấy đơn đã hủy/trả)

So Sánh Với Orders.php

Đặc điểm	returnDate.php	orders.php
Mục đích	Xem đơn đã hủy/trả	Quản lý tất cả đơn hàng
Filter	Chỉ hiển thị Cancelled/Returned	Hiển thị tất cả
Tính ngày trả	✓ Có	✗ Không
Cập nhật trạng thái	✓ Có (giống orders.php)	✓ Có
Form cập nhật	✗ Không có trong code hiện tại	✓ Có

Lưu Ý Quan Trọng

- Tính toán ngày trả:**
 - Sử dụng PHP DateTime để tính toán chính xác
 - Công thức: Order Date + Duration = Return Date
 - Format Y-m-d để dễ đọc
- Bảo mật:**
 - ✓ Sử dụng htmlspecialchars() cho output
 - ✓ Type casting cho ID
- UI/UX:**
 - Hiển thị rõ ràng ngày trả dự kiến
 - Thông báo khi không có dữ liệu
 - Bảng dễ đọc

CHI TIẾT CATEGORIES

File: Admin/categories.php

Mục đích: Quản lý danh sách danh mục sách - Xem, thay đổi trạng thái, xóa

Flow Hoạt Động

[Admin truy cập categories.php]



[Kiểm tra session]
↓
[Xử lý action (nếu có GET)]
├─→ Thay đổi status (active/deactive)
└─→ Xóa category
↓
[Lấy danh sách categories từ database]
↓
[Hiển thị bảng danh sách]

Code Chi Tiết

1. Kiểm Tra Session

```
if (!isset($_SESSION['ADMIN_LOGIN']) || $_SESSION['ADMIN_LOGIN'] == ' ') {  
    header('Location: login.php');  
    exit;  
}
```

Giải thích: - Kiểm tra session trước khi xử lý logic

2. Xử Lý Action

```
if (isset($_GET['type']) && $_GET['type'] != ' ') {  
    $type = getSafeValue($con, $_GET['type']);  
    $id = (int)$_GET['id'];  
  
    if ($type == 'status') {  
        $operation = getSafeValue($con, $_GET['operation']);  
        $status = ($operation == 'active') ? 1 : 0;  
        mysqli_query($con, "UPDATE categories SET status=$status WHERE  
id=$id");  
    } elseif ($type == 'delete') {  
        mysqli_query($con, "DELETE FROM categories WHERE id=$id");  
    }  
  
    header('Location: categories.php');  
    exit;  
}
```

Giải thích: - **Thay đổi status:** - Nhận operation (active/deactive) - Chuyển thành số: active = 1, deactive = 0 - Cập nhật trong database - **Xóa category:** - Xóa category khỏi database - Redirect về categories.php sau khi xử lý

3. Lấy Danh Sách Categories

```
$sql = "select * from categories order by category asc";  
$res = mysqli_query($con, $sql);
```

Giải thích: - Lấy tất cả categories - Sắp xếp theo tên tăng dần (A-Z)

4. Hiển Thị Bảng

```
<table class="table">
  <thead>
    <tr>
      <th>Categories</th>
      <th>Status</th>
      <th></th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    <?php while ($row = mysqli_fetch_assoc($res)): ?>
      <tr>
        <td><?php echo htmlspecialchars($row['category']) ?></td>
        <td>
          <?php if ($row['status'] == 1): ?>
            <a href="?type=status&operation=deactive&id=?php echo $row['id'] ?>">Active</a>
          <?php else: ?>
            <a href="?type=status&operation=active&id=?php echo $row['id'] ?>">Inactive</a>
          <?php endif; ?>
        </td>
        <td>
          <a href="manageCategories.php?id=?php echo $row['id'] ?>">Edit</a>
        </td>
        <td>
          <a href="?type=delete&id=?php echo $row['id'] ?>"
            onclick="return confirm('Are you sure you want to delete this category?')">Delete</a>
        </td>
      </tr>
    <?php endwhile; ?>
  </tbody>
</table>
```

Giải thích: - Hiển thị tên category (dùng htmlspecialchars() để bảo mật) - **Status:** Hiển thị link để toggle status - Nếu Active → Link “Active” (click để deactive) - Nếu Inactive → Link “Inactive” (click để active) - **Edit:** Link đến manageCategories.php?id=X - **Delete:** Link xóa với confirm dialog JavaScript

Database Schema

Bảng categories

Cột	Kiểu	Mô tả
id	int(11)	ID category (Primary Key)

Cột	Kiểu	Mô tả
category	varchar(50)	Tên danh mục
status	int(11)	Trạng thái (1 = active, 0 = inactive)

Sử dụng trong: - categories.php: SELECT (lấy danh sách), UPDATE (status), DELETE (xóa)
 - manageCategories.php: SELECT (lấy 1 category), INSERT, UPDATE

Lưu Ý Quan Trọng

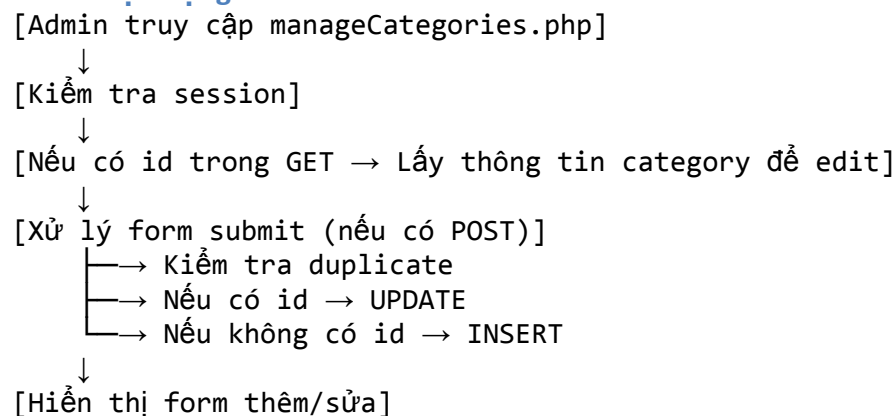
- Bảo mật:**
 - ✓ Sử dụng `getSafeValue()` và type casting
 - ✓ Sử dụng `htmlspecialchars()` cho output
 - ✓ Có confirm dialog trước khi xóa
- Chức năng:**
 - Xem danh sách categories
 - Toggle status (active/inactive)
 - Xóa category
 - Link đến trang thêm/sửa
- UI/UX:**
 - Bảng đơn giản, dễ đọc
 - Status có thể click để toggle
 - Có confirm dialog khi xóa

CHI TIẾT MANAGECATEGORIES

File: Admin/manageCategories.php

Mục đích: Thêm mới hoặc sửa danh mục sách

Flow Hoạt Động



Code Chi Tiết

1. Kiểm Tra Session

```
if (!isset($_SESSION['ADMIN_LOGIN']) || $_SESSION['ADMIN_LOGIN'] == ' ') {  
    header('Location: login.php');  
    exit;  
}
```

2. Lấy Thông Tin Category (Nếu Đang Edit)

```
$id = isset($_GET['id']) ? (int)$_GET['id'] : 0;  
$categories = '';  
$msg = '';  
$res = '';
```

// Lấy thông tin category nếu đang edit

```
if ($id > 0) {  
    $sql = mysqli_query($con, "SELECT * FROM categories WHERE id=$id");  
    if ($row = mysqli_fetch_assoc($sql)) {  
        $categories = $row['category'];  
    } else {  
        header('Location: categories.php');  
        exit;  
    }  
}
```

Giải thích: - Lấy id từ GET (nếu có) - Nếu có id: - Query database lấy thông tin category - Gán vào biến \$categories để hiển thị trong form - Nếu không tìm thấy → Redirect về categories.php

3. Xử Lý Form Submit

```
if (isset($_POST['submit'])) {  
    $category = getSafeValue($con, $_POST['category']);  
  
    // Check duplicate (trừ category hiện tại nếu đang edit)  
    $checkSql = mysqli_query($con, "SELECT id FROM categories WHERE  
category='$category'");  
    if (mysqli_num_rows($checkSql) > 0) {  
        $existing = mysqli_fetch_assoc($checkSql);  
        if (!$id || $existing['id'] != $id) {  
            $msg = "Category already exists";  
        }  
    }  
  
    if (empty($msg)) {  
        if ($id > 0) {  
            $sql = "UPDATE categories SET category='$category' WHERE id=$id";  
        } else {  
            $sql = "INSERT INTO categories(category, status)  
VALUES('$category', 1)";  
        }  
    }  
}
```



```

        if (mysqli_query($con, $sql)) {
            header('Location: categories.php');
            exit;
        } else {
            $res = "Error: " . mysqli_error($con);
        }
    }
}

```

Giải thích: - **Kiểm tra duplicate:** - Tìm category có tên trùng - Nếu đang edit (\$id > 0): Cho phép trùng với chính nó - Nếu đang thêm mới: Không cho phép trùng - **Thêm mới hoặc cập nhật:** - Nếu có id → UPDATE - Nếu không có id → INSERT (status mặc định = 1) - Redirect về categories.php sau khi thành công

4. Hiện Thị Form

```

<form method="post">
    <div class="form-outline mb-4 mx-5">
        <input type="text" name="category" value="<?php echo $categories ?>"
id="category" class="form-control" required />
        <label class="form-label" for="category">Enter category name</label>
    </div>
    <div class="mb-1 d-flex justify-content-center field_error">
        <?php echo $msg ?>
    </div>
    <div class="mb-1 d-flex justify-content-center">
        <?php echo $res ?>
    </div>
    <div class="text-center">
        <button type="submit" name="submit" class="btn btn-primary
mx-5">Submit</button>
    </div>
</form>

```

Giải thích: - Form đơn giản với 1 input: tên category - Auto-fill giá trị nếu đang edit - Hiện thị thông báo lỗi nếu có

Database Schema

Sử dụng bảng categories (đã giải thích ở phần Categories)

Sử dụng trong: - manageCategories.php: SELECT (lấy 1 category), INSERT (thêm mới), UPDATE (sửa)

Lưu Ý Quan Trọng

1. **Logic thêm/sửa:**
 - Cùng 1 form cho cả thêm và sửa
 - Phân biệt bằng cách kiểm tra \$id
 - Kiểm tra duplicate thông minh (cho phép trùng với chính nó khi edit)

2. Bảo mật:

- ☒ Sử dụng `getSafeValue()` để bảo mật
- ☒ Type casting cho ID

3. UI/UX:

- Form đơn giản, dễ sử dụng
- Auto-fill khi edit
- Hiển thị thông báo lỗi rõ ràng

CHI TIẾT BOOKS

File: `Admin/books.php`

Mục đích: Quản lý danh sách sách - Xem, thay đổi trạng thái, xóa

Flow Hoạt Động

[Admin truy cập `books.php`]



[Kiểm tra session]



[Xử lý action (nếu có GET)]

- Thay đổi status (active/inactive)
- Thay đổi best_seller (Most Viewed/Normal)
- Xóa sách



[Lấy danh sách sách từ database]



[Hiển thị bảng danh sách]

Code Chi Tiết

1. Kiểm Tra Session

```
if (!isset($_SESSION['ADMIN_LOGIN']) || $_SESSION['ADMIN_LOGIN'] == ' ') {  
    header('Location: login.php');  
    exit;  
}
```

2. Xử Lý Action

```
if (isset($_GET['type']) && $_GET['type'] != ' ') {  
    $type = getSafeValue($con, $_GET['type']);  
    $id = (int)$_GET['id'];  
  
    if ($type == 'status') {  
        $status = ($_GET['operation'] == 'active') ? 1 : 0;  
        mysqli_query($con, "UPDATE books SET status=$status WHERE id=$id");  
    } elseif ($type == 'best_seller') {  
        $bestSeller = ($_GET['operation'] == 'active') ? 1 : 0;  
        mysqli_query($con, "UPDATE books SET best_seller=$bestSeller WHERE  
id=$id");  
    }  
}
```

```

    } elseif ($type == 'delete') {
        mysqli_query($con, "DELETE FROM books WHERE id=$id");
    }

    header('Location: books.php');
    exit;
}

```

Giải thích: - **Thay đổi status:** - Active/Deactive sách (hiển thị/ẩn trên website) - **Thay đổi best_seller:** - Đánh dấu sách là “Most Viewed” (hiển thị trên trang chủ) - **Xóa sách:** - Xóa sách khỏi database

3. Lấy Danh Sách Sách

```

$sql = "SELECT books.*, categories.category
        FROM books
        LEFT JOIN categories ON books.category_id=categories.id
        ORDER BY books.name ASC";
$res = mysqli_query($con, $sql);

```

Giải thích: - JOIN với bảng categories để lấy tên danh mục - LEFT JOIN: Vẫn hiển thị sách dù không có category - Sắp xếp theo tên sách tăng dần

4. Hiển Thị Bảng

```

<table class="table">
    <thead>
        <tr>
            <th>ISBN</th>
            <th>Category</th>
            <th>img</th>
            <th>Name</th>
            <th>Author</th>
            <th>Security</th>
            <th>Rent</th>
            <th>Price</th>
            <th>Qty</th>
            <th>Status</th>
            <th>Actions</th>
        </tr>
    </thead>
    <tbody>
        <?php while ($row = mysqli_fetch_assoc($res)): ?>
            <tr>
                <td><?php echo htmlspecialchars($row['ISBN']) ?></td>
                <td><?php echo htmlspecialchars($row['category']) ?? 'N/A'
?></td>
                <td>
                    
                </td>
            </tr>
        </tbody>
    </table>

```

```

<td><?php echo htmlspecialchars($row['name']) ?></td>
<td><?php echo htmlspecialchars($row['author']) ?></td>
<td><?php echo number_format($row['security']) ?></td>
<td><?php echo number_format($row['rent']) ?>/day</td>
<td><?php echo number_format($row['price']) ?></td>
<td><?php echo $row['qty'] ?></td>
<td>
    <?php if ($row['best_seller'] == 1): ?>
        <a href="?type=best_seller&operation=deactive&id=<?php
echo $row['id'] ?>">Most Viewed</a>
    <?php else: ?>
        <a href="?type=best_seller&operation=active&id=<?php echo
$row['id'] ?>">Normal</a>
    <?php endif; ?>
</td>
<td>
    <a href="manageBooks.php?id=<?php echo $row['id']
?>">Edit</a> |
    <a href="?type=delete&id=<?php echo $row['id'] ?>"
onclick="return confirm('Are you sure you want to delete
this book?')">Delete</a>
</td>
</tr>
<?php endwhile; ?>
</tbody>
</table>

```

Giải thích: - Hiển thị đầy đủ thông tin sách - **Ảnh sách:** Hiển thị thumbnail - **Status:** Hiển thị “Most Viewed” hoặc “Normal” (có thể click để toggle) - **Actions:** Edit và Delete - Sử dụng htmlspecialchars() và number_format() để format dữ liệu

Database Schema

Bảng books

Cột	Kiểu	Mô tả
id	int(11)	ID sách (Primary Key)
ISBN	varchar(20)	ISBN sách
category_id	int(11)	ID danh mục (FK → categories.id)
name	varchar(200)	Tên sách
author	varchar(100)	Tác giả
security	float	Tiền đặt cọc
rent	float	Giá thuê/ngày
price	float	Giá bán (nếu có)
qty	int(11)	Số lượng

Cột	Kiểu	Mô tả
status	int(11)	Trạng thái (1 = active, 0 = inactive)
best_seller	int(11)	Bán chạy (1 = Most Viewed, 0 = Normal)
img	varchar(200)	Tên file ảnh
short_desc	text	Mô tả ngắn
description	text	Mô tả chi tiết

Sử dụng trong: - books.php: SELECT (lấy danh sách), UPDATE (status, best_seller), DELETE (xóa) - manageBooks.php: SELECT (lấy 1 sách), INSERT, UPDATE

Lưu Ý Quan Trọng

- Chức năng:**
 - Xem danh sách sách đầy đủ
 - Toggle status (active/inactive)
 - Toggle best_seller (Most Viewed/Normal)
 - Xóa sách
 - Link đến trang thêm/sửa
- Bảo mật:**
 - ☒ Sử dụng `getSafeValue()` và type casting
 - ☒ Sử dụng `htmlspecialchars()` cho output
 - ☒ Có confirm dialog khi xóa
- UI/UX:**
 - Hiển thị ảnh thumbnail
 - Format số tiền dễ đọc
 - Status có thể click để toggle

CHI TIẾT MANAGEBOOKS

File: Admin/manageBooks.php

Mục đích: Thêm mới hoặc sửa sách

Flow Hoạt Động

```

[Admin truy cập manageBooks.php]
↓
[require topNav.php - Kiểm tra session]
↓
[Nếu có id trong GET → Lấy thông tin sách để edit]
↓
[Xử lý form submit (nếu có POST)]
|→ Kiểm tra duplicate

```

- Upload ảnh (nếu thêm mới)
- Nếu có id → UPDATE
- Nếu không có id → INSERT

↓
[Hiện thị form thêm/sửa]

Code Chi Tiết

1. Include topNav

```
require('topNav.php');
```

2. Lấy Thông Tin Sách (Nếu Đang Edit)

```
$id = isset($_GET['id']) ? (int)$_GET['id'] : 0;
$category_id = '';
$ISBN = '';
$name = '';
// ... các biến khác

if ($id > 0) {
    $sql = mysqli_query($con, "SELECT * FROM books WHERE id=$id");
    if ($row = mysqli_fetch_assoc($sql)) {
        $category_id = $row['category_id'];
        $ISBN = $row['ISBN'];
        $name = $row['name'];
        // ... gán các giá trị khác
    } else {
        header('Location: books.php');
        exit;
    }
}
```

Giải thích: - Khởi tạo các biến để lưu thông tin sách - Nếu có id: Lấy thông tin sách và gán vào các biến - Nếu không tìm thấy → Redirect về books.php

3. Xử Lý Form Submit

```
if (isset($_POST['submit'])) {
    $category_id = (int)$_POST['category_id'];
    $ISBN = getSafeValue($con, $_POST['ISBN']);
    $name = getSafeValue($con, $_POST['name']);
    $author = getSafeValue($con, $_POST['author']);
    $security = (int)$_POST['security'];
    $rent = (int)$_POST['rent'];
    $qty = (int)$_POST['qty'];
    $short_desc = getSafeValue($con, $_POST['short_desc']);
    $description = getSafeValue($con, $_POST['description']);

    // Check if book name already exists (except current book)
    $checkSql = mysqli_query($con, "SELECT id FROM books WHERE
name='$name'");
    if (mysqli_num_rows($checkSql) > 0) {
```

```

        $existing = mysqli_fetch_assoc($checkSql);
        if (! $id || $existing['id'] != $id) {
            $msg = "Book already exists";
        }
    }

    if (empty($msg)) {
        if ($id > 0) {
            // Update existing book
            $sql = "UPDATE books SET category_id=$category_id, ISBN='$ISBN',
name='$name', author='$author',
            security=$security, rent=$rent, qty=$qty,
short_desc='$short_desc',
            description='$description' WHERE id=$id";
        } else {
            // Insert new book
            if (!empty($_FILES['img']['name'])) {
                $img = time() . '_' . $_FILES['img']['name'];
                move_uploaded_file($_FILES['img']['tmp_name'],
BOOK_IMAGE_SERVER_PATH . $img);
            } else {
                $msg = "Please upload book image";
            }

            if (empty($msg)) {
                $sql = "INSERT INTO books(category_id, ISBN, name, author,
security, rent, qty, short_desc, description, status, img)
                VALUES ($category_id, '$ISBN', '$name', '$author',
                $security, $rent, $qty, '$short_desc', '$description', 1, '$img')";
            }
        }

        if (empty($msg) && mysqli_query($con, $sql)) {
            header('Location: books.php');
            exit;
        } else {
            $error = "Error: " . mysqli_error($con);
        }
    }
}
}

```

Giải thích: - **Lấy dữ liệu từ form:** - Type casting cho số: (int)\$_POST[...] -
getSafeValue() cho chuỗi - **Kiểm tra duplicate:** - Kiểm tra tên sách đã tồn tại chưa - Cho
phép trùng với chính nó khi edit - **Upload ảnh (chỉ khi thêm mới):** - Kiểm tra có file upload
không - Đặt tên file: time() . '_' . tên file gốc (tránh trùng) - Lưu vào
BOOK_IMAGE_SERVER_PATH - **Lưu ý:** Khi edit không bắt buộc upload ảnh mới - **Thêm mới
hoặc cập nhật:** - Nếu có id → UPDATE (không cập nhật ảnh) - Nếu không có id → INSERT
(bắt buộc có ảnh)

4. Hiện Thị Form

```
<form method="post" enctype="multipart/form-data">
  <!-- ISBN và Category -->
  <div class="row g-3">
    <div class="col-sm-8">
      <input type="text" name="ISBN" value="<?php echo $ISBN ?>"
required />
    </div>
    <div class="col-sm">
      <select class="form-select" name="category_id">
        <option>Select Category</option>
        <?php
          $categorySql = mysqli_query($con, "select id, category from
categories order by category asc");
          while ($row = mysqli_fetch_assoc($categorySql)) {
            if ($row['id'] == $category_id) {
              echo "<option selected value=" . $row['id'] . ">" .
$row['category'] . "</option>";
            } else {
              echo "<option value=" . $row['id'] . ">" .
$row['category'] . "</option>";
            }
          }
        </select>
      </div>
    </div>

    <!-- Các trường khác: name, author, security, rent, qty, img, short_desc,
description -->

    <button type="submit" name="submit" class="btn
btn-primary">Submit</button>
  </form>
```

Giải thích: - Form có enctype="multipart/form-data" để upload file - **Category**

dropdown: - Lấy danh sách categories từ database - Auto-select category hiện tại khi edit -

Các trường: - ISBN, Name, Author: Text input - Security, Rent, Qty: Number input - Image: File input (chỉ bắt buộc khi thêm mới) - Short Description, Description: Textarea - Auto-fill giá trị khi edit

Database Schema

Sử dụng bảng books (đã giải thích ở phần Books)

Sử dụng trong: - manageBooks.php: SELECT (lấy 1 sách), INSERT (thêm mới), UPDATE (sửa) - categories: SELECT (lấy danh sách categories cho dropdown)

Lưu Ý Quan Trọng

1. Upload ảnh:

- Chỉ bắt buộc khi thêm mới
 - Tên file: time() . '_' . tên_gốc (tránh trùng)
 - Lưu vào BOOK_IMAGE_SERVER_PATH
2. **Logic thêm/sửa:**
- Cùng 1 form cho cả thêm và sửa
 - Khi edit: Không cập nhật ảnh (giữ nguyên ảnh cũ)
 - Khi thêm: Bắt buộc upload ảnh
3. **Bảo mật:**
- ☒ Sử dụng `getSafeValue()` và type casting
 - ☒ Kiểm tra duplicate
 - ☒ ⚠ Không validate file type (nên thêm)
4. **UI/UX:**
- Form đầy đủ các trường
 - Auto-fill khi edit
 - Dropdown category dễ chọn
-

🗨 CHI TIẾT FEEDBACK

File: Admin/feedback.php

Mục đích: Xem và xóa phản hồi từ khách hàng

Flow Hoạt Động

```
[Admin truy cập feedback.php]
↓
[require topNav.php - Kiểm tra session]
↓
[Xử lý xóa feedback (nếu có GET)]
↓
[Lấy danh sách feedback từ database]
↓
[Hiện thị bảng danh sách]
```

Code Chi Tiết

1. Include topNav

```
require('topNav.php');
```

2. Xử Lý Xóa Feedback

```
if (isset($_GET['type']) && $_GET['type'] != ' ') {
    $type = getSafeValue($con, $_GET['type']);

    if ($type == 'delete') {
        $id = getSafeValue($con, $_GET['id']);
        $deleteSql = "delete from contact_us where id='$id'";
        mysqli_query($con, $deleteSql);
    }
}
```

```
}  
}
```

Giải thích: - Kiểm tra có action delete không - Lấy ID feedback cần xóa - Xóa feedback khỏi database - **Lưu ý:** Không có redirect, trang sẽ reload và hiển thị danh sách mới

3. Lấy Danh Sách Feedback

```
$sql = "select * from contact_us order by id desc";  
$res = mysqli_query($con, $sql);
```

Giải thích: - Lấy tất cả feedback - Sắp xếp theo ID giảm dần (feedback mới nhất trước)

4. Hiển Thị Bảng

```
<table class="table">  
  <thead>  
    <tr>  
      <th>ID</th>  
      <th>Name</th>  
      <th>Email</th>  
      <th>Mobile</th>  
      <th>Message</th>  
      <th>Date</th>  
      <th></th>  
    </tr>  
  </thead>  
  <tbody>  
    <?php while ($row = mysqli_fetch_assoc($res)): ?>  
      <tr>  
        <td><?php echo $row['id'] ?></td>  
        <td><?php echo $row['name'] ?></td>  
        <td><?php echo $row['email'] ?></td>  
        <td><?php echo $row['mobile'] ?></td>  
        <td><?php echo $row['message'] ?></td>  
        <td><?php echo $row['date'] ?></td>  
        <td>  
          <a class='link-white btn btn-danger px-2 py-1'  
            href='?type=delete&id=<?php echo $row['id'] ?>'>Delete</a>  
        </td>  
      </tr>  
    <?php endwhile; ?>  
  </tbody>  
</table>
```

Giải thích: - Hiển thị bảng với các cột: ID, Name, Email, Mobile, Message, Date - Mỗi dòng có nút Delete màu đỏ - **Lưu ý:** Không sử dụng htmlspecialchars() cho output (nên thêm để bảo mật)

Database Schema

Bảng *contact_us*

Cột	Kiểu	Mô tả
id	int(11)	ID feedback (Primary Key)
name	varchar(70)	Tên người gửi
email	varchar(70)	Email người gửi
mobile	bigint(10)	Số điện thoại
message	text	Nội dung phản hồi
date	datetime	Ngày gửi

Sử dụng trong: - feedback.php: SELECT (lấy danh sách), DELETE (xóa) - pages/contactUs.php: INSERT (khách hàng gửi phản hồi)

Lưu Ý Quan Trọng

- Bảo mật:**
 - ✓ Sử dụng `getSafeValue()` để bảo mật
 - ⚠ Không sử dụng `htmlspecialchars()` cho output (nên thêm)
 - ⚠ Không có xác nhận trước khi xóa (có thể thêm confirm dialog)
- Chức năng:**
 - Chỉ có chức năng xem và xóa
 - Không có chức năng trả lời feedback (có thể thêm trong tương lai)
- UI/UX:**
 - Bảng đơn giản, dễ đọc
 - Nút Delete màu đỏ để cảnh báo
 - Hiển thị đầy đủ thông tin feedback



TÓM TẮT

Authentication

- Login:** Xác thực admin, set session, redirect đến categories.php
- Logout:** Xóa session, redirect về login.php
- TopNav:** Navigation chung, tự động kiểm tra session

Quản Lý Người Dùng

- Users:** Xem danh sách users, xóa user

Quản Lý Đơn Hàng

- Orders:** Xem danh sách đơn hàng, cập nhật trạng thái, tự động tăng số lượng sách khi hủy/trả
- ReturnDate:** Xem đơn đã hủy/trả, tính ngày trả dự kiến

Quản Lý Danh Mục

- **Categories:** Xem danh sách categories, toggle status, xóa
- **ManageCategories:** Thêm mới hoặc sửa category, kiểm tra duplicate

Quản Lý Sách

- **Books:** Xem danh sách sách, toggle status/best_seller, xóa
- **ManageBooks:** Thêm mới hoặc sửa sách, upload ảnh (chỉ khi thêm mới)

Phản Hồi

- **Feedback:** Xem danh sách phản hồi từ khách hàng, xóa

Điểm Chung

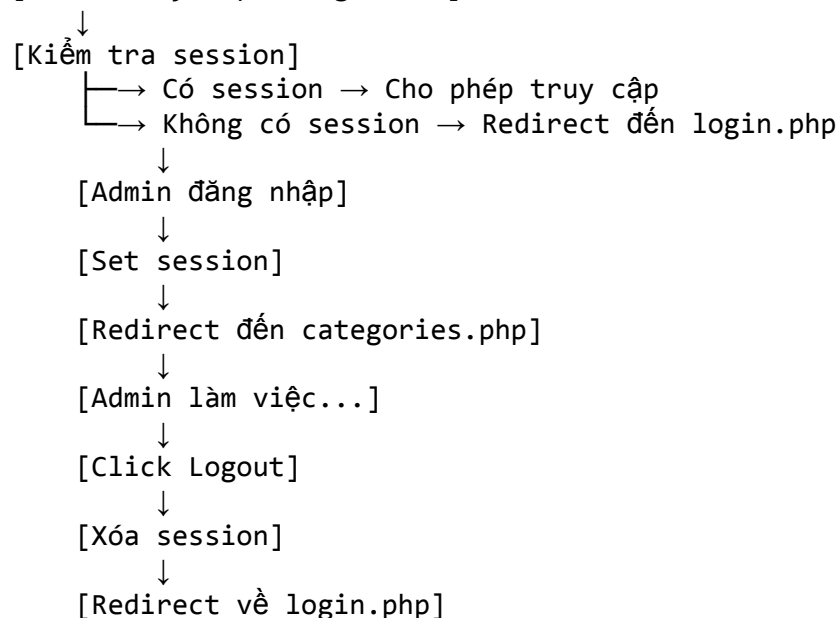
Bảo mật: - ☒ Kiểm tra session ở tất cả trang (trừ login.php) - ☒ Sử dụng `getSafeValue()` để chống SQL injection - ☒ Type casting cho ID: `(int)$_GET['id']` - ☒ Sử dụng `htmlspecialchars()` cho output (hầu hết các file)

Pattern chung: - Xử lý action (GET/POST) TRƯỚC KHI require topNav - Redirect sau khi xử lý thành công - Hiện thị bảng với các action (Edit, Delete, Toggle status)



FLOW TỔNG QUAN

[Admin truy cập trang admin]



GHI CHÚ

- Tài liệu này đã giải thích đầy đủ:
 - ☒ Login và Logout

- ☒ TopNav (Navigation)
- ☒ Users (Quản lý người dùng)
- ☒ Orders (Quản lý đơn hàng)
- ☒ ReturnDate (Ngày trả sách)
- ☒ Categories (Quản lý danh mục)
- ☒ ManageCategories (Thêm/Sửa danh mục)
- ☒ Books (Quản lý sách)
- ☒ ManageBooks (Thêm/Sửa sách)
- ☒ Feedback (Phản hồi khách hàng)

Tài liệu này giúp bạn hiểu rõ cách hoạt động của các phần chính trong hệ thống Admin.
Chúc bạn code vui vẻ! 🚀