



TRƯỜNG ĐẠI HỌC NGÂN HÀNG
THÀNH PHỐ HỒ CHÍ MINH

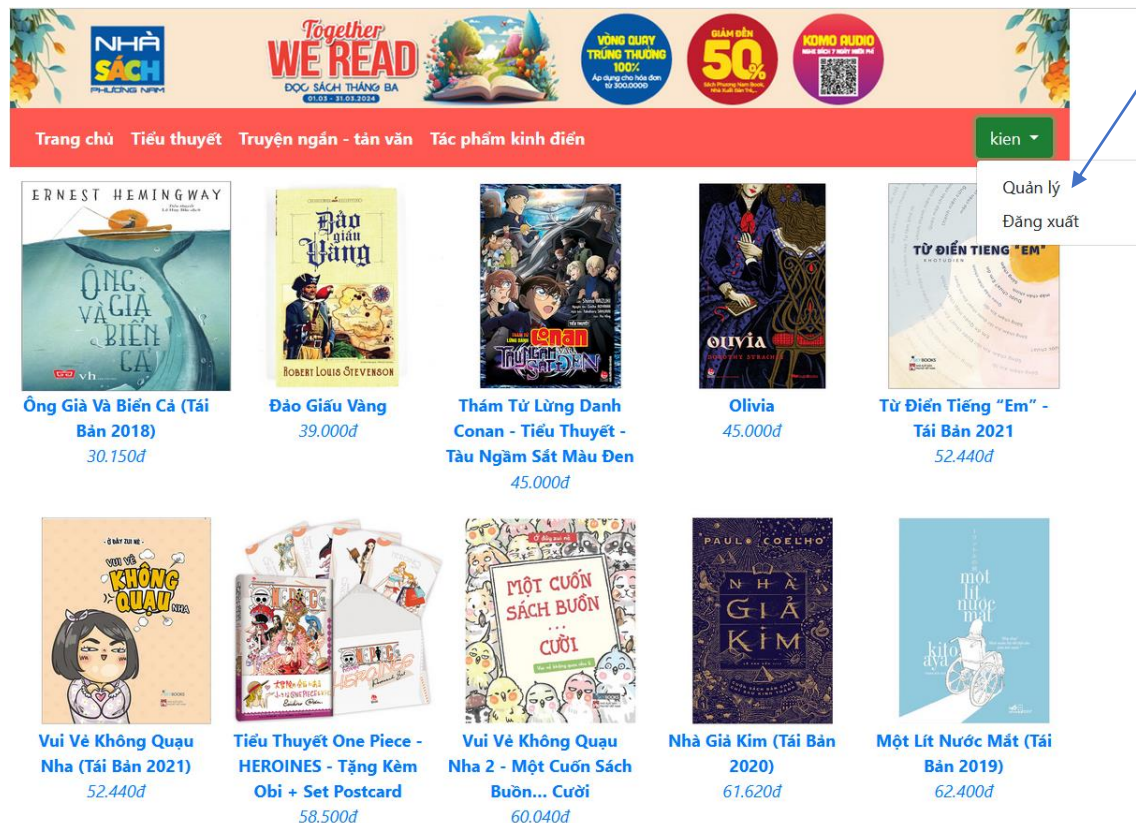
PHÁT TRIỂN ỨNG DỤNG MÃ NGUỒN MỞ

BÀI THỰC HÀNH 3

- Tạo trang cập nhật thông tin người dùng
- Tạo trang quản lý sách đơn giản

THỰC HÀNH 3

- Yêu cầu 1: Thiết kế giao diện trang chủ như hình, khi chọn vào Quản lý sẽ di chuyển đến phần cập nhật thông tin cá nhân của tài khoản



THỰC HÀNH 3

■ Bước 1: Điều chỉnh lại giao diện trang chủ

NHÀ SÁCH PHƯƠNG NHIỆM

Together WE READ
ĐỌC SÁCH THÁNG BA
01.03 - 31.03.2024

VÒNG QUAY TRUNG THUƯƠNG 100%
Áp dụng cho hóa đơn từ 300.000đ

GIẢM ĐẾN 50%
Áp dụng cho hóa đơn từ 300.000đ

KOMO AUDIO
Hàng loạt 7 cuốn audiobook miễn phí

Trang chủ | Tiểu thuyết | Truyện ngắn - tản văn | Tác phẩm kinh điển

Đăng nhập | Đăng ký

Ảnh bìa sách	Tên sách	Giá
	Ông Già và Biển Cả (Tái Bản 2018)	30.150đ
	Đảo Giấu Vàng	39.000đ
	Thám Tử Lừng Danh Conan - Tiểu Thuyết - Tàu Ngầm Sắt Màu Đen	45.000đ
	Olivia	45.000đ
	Từ Điển Tiếng "Em" - Tái Bản 2021	52.440đ
	Vui Vẻ Không Quay Nha (Tái Bản 2021)	52.440đ
	Tiểu Thuyết One Piece - HEROINES - Tặng Kèm Obi + Set Postcard	58.500đ
	Vui Vẻ Không Quay Nha 2 - Một Cuốn Sách Buồn... Cười	60.040đ
	Nhà Giả Kim (Tái Bản 2020)	61.620đ
	Một Lít Nước Mắt (Tái Bản 2019)	62.400đ

■ Bước 1: Điều chỉnh lại giao diện trang chủ

Tại file `views/components/book-layout.blade.php` điều chỉnh lại theo nội dung file `new_layout.txt`

Một số đoạn code cần lưu ý:

Kiểm tra đăng nhập tại file giao diện



THỰC HÀNH 3



- Bước 1: Điều chỉnh lại giao diện trang chủ
Tại file routes/web.php điều chỉnh lại như sau:

```
/*Route::get('/', function () {  
    return view('welcome');  
});*/
```

Chú thích lại trang mặc định (trang chủ) Laravel

Điều chỉnh trang chủ là trang Sách

```
Route::get('/', 'App\Http\Controllers\ViduLayoutController@sach');
```

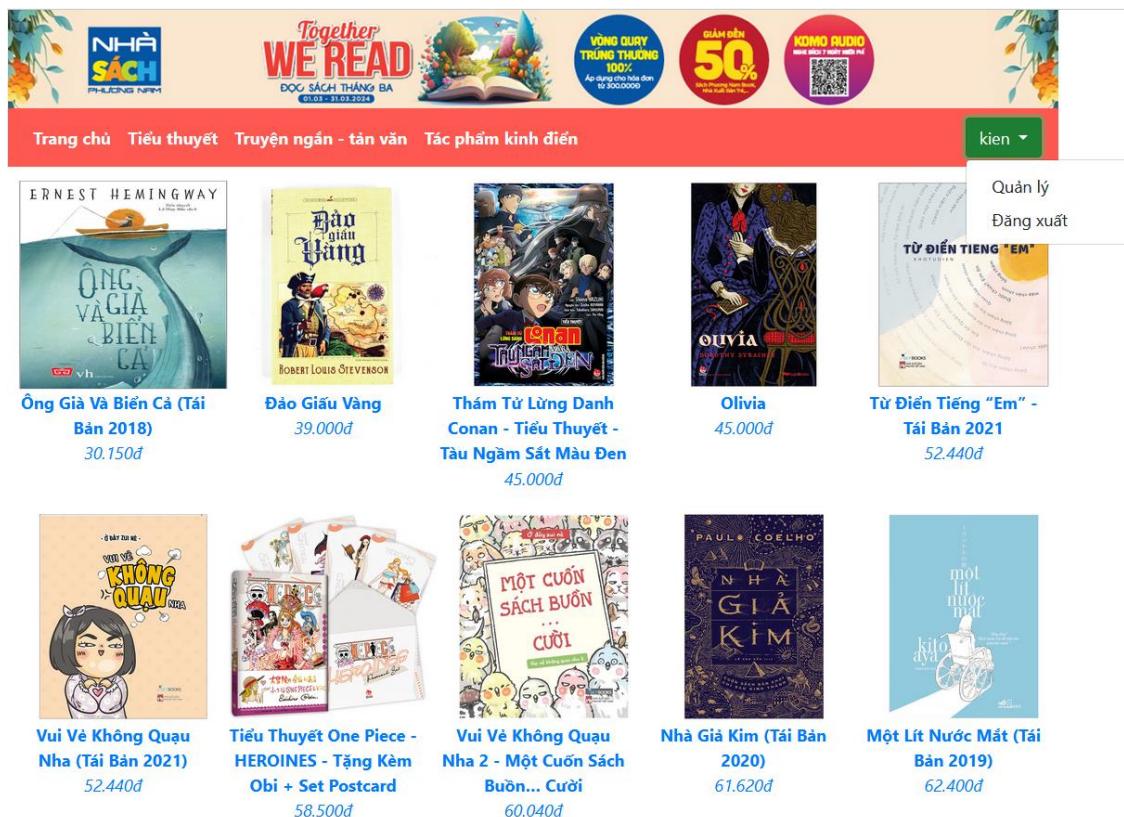

THỰC HÀNH 3

■ Bước 1: Điều chỉnh lại giao diện trang chủ

Tại file Providers/RouteServiceProvider.php điều chỉnh lại đường dẫn trang chủ như sau:

```
18
19
20
21
```

```
* @var string
*/
public const HOME = '/';
```



- Bước 2: Tạo trang cập nhật thông tin người dùng
Tạo Route cho trang cập nhật thông tin:

```
Route::get('/accountpanel', 'App\Http\Controllers\AccountController@accountpanel')  
->middleware('auth')->name("account");
```

Sử dụng `middleware('auth')` để kiểm tra tình trạng đăng nhập của người dùng trước khi tiến hành thực thi hàm xử lý của đường dẫn. Các đường dẫn cần người dùng đăng nhập phải có `middleware('auth')`.

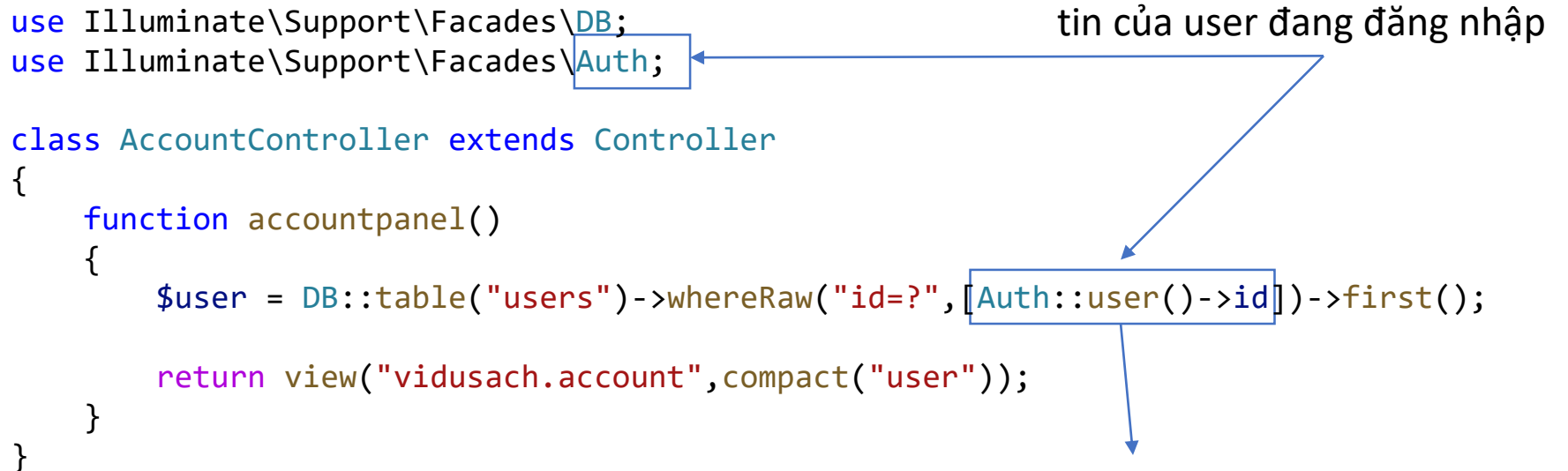
Đặt tên cho đường dẫn, khi đã đặt tên cho đường dẫn thì có thể sử dụng hàm `route` để lấy đường dẫn thông qua tên.

- Bước 2: Tạo trang cập nhật thông tin người dùng
Tạo file AccountController với hàm xử lý accountpanel:

```
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Auth;

class AccountController extends Controller
{
    function accountpanel()
    {
        $user = DB::table("users")->whereRaw("id=?", [Auth::user()->id])->first();

        return view("vidusach.account", compact("user"));
    }
}
```



The diagram illustrates the data flow in the `accountpanel()` function. A blue box highlights the `Auth` facade in the `use` statements. An arrow points from this box to the text "Facade Auth hỗ trợ lấy thông tin của user đang đăng nhập". Another arrow points from the `Auth::user()->id` expression in the database query to the text "Một số thông tin khác của user có thể lấy bằng Facade Auth: Auth::user()->name, Auth::user()->email".

Facade Auth hỗ trợ lấy thông tin của user đang đăng nhập

Một số thông tin khác của user có thể lấy bằng Facade Auth:
Auth::user()->name
Auth::user()->email

- Bước 2: Tạo trang cập nhật thông tin người dùng

Thiết kế trang cập nhật thông tin người dùng bằng cách tạo một component có tên là **account-panel.blade.php** với nội dung như trong file **account-panel.txt**

■ Bước 2: Tạo trang cập nhật thông tin người dùng

Tại file `vidusach/account.blade`, tạo form cập nhật thông tin với nội dung như sau:

Sử dụng component `account-panel` cho form cập nhật thông tin

Route xử lý cập nhật thông tin

Phải có
`csrf_field()`
khi cập
nhập form

```
<x-account-panel>
  <form method = 'post' action="{{route('saveinfo')}}" style='width:30%; margin:0 auto'>
    <div style='text-align:center;font-weight:bold;color:#15c;'>CẬP NHẬT THÔNG TIN CÁ NHÂN</div>
    <label>Tên</label>
    <input type='text' class='form-control form-control-sm' name='name' value="{{ $user->name }}">
    <label>Email</label>
    <input type='text' class='form-control form-control-sm' name='email' value="{{ $user->email }}">
    <label>Số điện thoại</label>
    <input type='text' class='form-control form-control-sm' name='phone' value="{{ $user->phone }}">
    <input type='hidden' value="{{ $user->id }}" name='id'>
    {{ csrf_field() }}
    <div style='text-align:center;'><input type='submit' class='btn btn-primary mt-1' value='Lưu'></div>
  </form>
</x-account-panel>
```

- Bước 2: Tạo trang cập nhật thông tin người dùng
Tạo Route cho trang xử lý cập nhật:

```
Route::post('/saveaccountinfo', 'App\Http\Controllers\AccountController@saveaccountinfo')  
->middleware('auth')->name('saveinfo');
```

Sử dụng middleware('auth') để kiểm tra tình trạng đăng nhập của người dùng trước khi tiến hành cập nhật thông tin

Đặt tên cho đường dẫn

Thêm cột phone kiểu chuỗi 10 ký tự vào bảng dữ liệu users

■ Bước 2: Tạo trang cập nhật thông tin người dùng

Tạo file AccountController với hàm xử lý saveaccountinfo:

```
function saveaccountinfo(Request $request)
```

```
{
```

```
    $request->validate([  
        'name' => ['required', 'string', 'max:255'],  
        'email' => ['required', 'string', 'email', 'max:255'],  
        'phone' => ['nullable', 'string']  
    ]);
```

```
    $id = $request->input('id');  
    $data["name"] = $request->input("name");  
    $data["phone"] = $request->input("phone");  
    $data["email"] = $request->input("email");
```

```
    DB::table("users")->where("id",$id)->update($data);
```

```
    return redirect()->route('account')->with('status', 'Cập  
    nhật thành công');
```

```
}
```

Kiểm tra dữ liệu người dùng nhập vào bằng phương thức validate.

Required: phải nhập dữ liệu

String: Dữ liệu nhập vào phải là chuỗi,

Max: Số lượng ký tự tối đa khi nhập

Email: Phải đúng định dạng email

Nullable: Có thể để trống

Sử dụng \$request->input() để lấy dữ liệu của người dùng gửi đến và lưu vào biến mảng \$data

Cập nhật dữ liệu vào bảng users sử dụng Query Builder

Sau khi cập nhật dữ liệu thành công, sẽ chuyển hướng về trang accountpanel thông qua tên đường dẫn (account) kèm theo thông báo cập nhật thành công được lưu trong biến Session

status

- Bước 2: Tạo trang cập nhật thông tin người dùng
Bổ sung thêm đoạn code thông báo lỗi (nếu có) và thông báo cập nhật thành công.

Tại file `vidusach/account.blade.php` bổ sung trước form cập nhật thông tin đoạn code sau:

```
@if ($errors->any())  
    <div style='color:red;width:30%; margin:0 auto'>  
        <div >  
            {{ __('Whoops! Something went wrong.') }}  
        </div>  
        <ul>  
            @foreach ($errors->all() as $error)  
                <li>{{ $error }}</li>  
            @endforeach  
        </ul>  
    </div>  
@endif  
@if (session('status'))  
    <div class="alert alert-success">  
        {{ session('status') }}  
    </div>  
@endif
```

Nếu có lỗi xảy ra

Sử dụng `$errors->all()` để lấy tất cả các lỗi và dùng vòng lặp để in ra từng lỗi

Nếu có biến session status (cập nhật thành công) thì in ra thông báo

- Bước 2: Tạo trang cập nhật thông tin người dùng
Thực hiện cập nhật thông tin cá nhân

Thông tin tài khoản
Quản lý sách

Trang chủ

Cập nhật thành công

CẬP NHẬT THÔNG TIN CÁ NHÂN

Tên

Email

Số điện thoại

Lưu

■ Bước 3: Upload ảnh đại diện

Tạo cột photo cho bảng users.

Điều chỉnh lại form cập nhật thông tin như sau:

Bổ sung thêm khi upload file

```
<form method = 'post' action="{{route('saveinfo')}}" enctype="multipart/form-data"
      style='width:30%; margin:0 auto'>
  <div style='text-align:center;font-weight:bold;color:#15c;'>CẬP NHẬT THÔNG TIN CÁ NHÂN</div>
  <label>Tên</label>
  <input type='text' class='form-control form-control-sm' name='name' value="{{ $user->name }}">
  <label>Email</label>
  <input type='text' class='form-control form-control-sm' name='email' value="{{ $user->email }}">
  <label>Số điện thoại</label>
  <input type='text' class='form-control form-control-sm' name='phone' value="{{ $user->phone }}">
  <input type='hidden' value="{{ $user->id }}" name='id'>
  <label>Ảnh đại diện</label><br>
  <input type="file" name="photo" id="photo" accept="image/*" class="form-control-file">
  {{ csrf_field() }}
  <div style='text-align:center;'><input type='submit' class='btn btn-primary mt-1' value='Lưu'></div>
</form>
```

Bổ sung thêm phần
chọn file ảnh để upload

■ Bước 3: Upload ảnh đại diện

Điều chỉnh hàm xử lý saveaccountinfo tại file AccountController như sau:

```
function saveaccountinfo(Request $request)
{
    $request->validate([
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255'],
        'phone' => ['nullable', 'string'],
        'photo' => ['nullable', 'image']
    ]);

    $id = $request->input('id');
    $data["name"] = $request->input("name");
    $data["phone"] = $request->input("phone");
    $data["email"] = $request->input("email");
    if($request->hasFile("photo"))
    {
        //Tạo tên file bằng cách lấy id của người dùng ghép với phần mở rộng của hình ảnh
        $fileName = Auth::user()->id . '.' . $request->file('photo')->extension();
        //File được lưu vào thư mục storage/app/public/profile
        $request->file('photo')->storeAs('public/profile', $fileName);
        $data['photo'] = $fileName;
    }

    DB::table("users")->where("id",$id)->update($data);
    return redirect()->route('account')->with('status', 'Cập nhật thành công');
}
```

Kiểm tra file upload.

image: file upload phải là hình ảnh

Bổ sung thêm đoạn code xử lý upload hình ảnh đại diện

- Bước 3: Upload ảnh đại diện

Chạy lệnh ***php artisan storage:link*** để liên kết đến thư mục chứa hình ảnh đại diện.

Thêm thẻ `` để hiện ảnh đại diện tại form cập nhật:

```

```

THỰC HÀNH 3



- Yêu cầu 2: Tạo một trang quản lý đơn giản sách có giao diện như sau:

DANH MỤC SÁCH

Thêm							
Tiêu đề	Nhà xuất bản	Nhà cung cấp	Tác giả	Hình thức bìa	Giá bán	Hình ảnh	Thao tác
Đảo Giấu Vàng	NXB Văn Học	Minh Long	Robert Louis Stevenson, Vũ Ngọc Phan	Bìa Mềm	39000		Sửa Xóa
Hai Số Phận (Tái Bản 2023)	Văn Học	Huy Hoang Bookstore	Jeffrey Archer	Bìa Mềm	175500		Sửa Xóa
Ông Già Và Biển Cả	Văn Học	Đông A	Ernest Hemingway	Bìa Cứng	130000		Sửa Xóa
Scarlett - Hậu Cước Theo Chiều Gió (2017)	NXB Văn Học	Huy Hoang Bookstore	Alexandra Ripley	Bìa Mềm	193500		Sửa Xóa
Kiều Hành Và Định Kiến - Bìa Cứng (Tái Bản 2023)	Văn Học	Nhà Sách Minh Thắng	Jane Austen	Bìa Cứng	115200		Sửa Xóa
Thần Thoại Bắc Âu - Văn Xúi Edda Và Thơ Edda	Thanh Niên	CÔNG TY TNHH SÁCH & TRUYỀN THÔNG VIỆT NAM	Peter Andreas Munch	Bìa Mềm	180000		Sửa Xóa
Không Gia Đình (Tái Bản 2022)	NXB Kim Đồng	Nhà Xuất Bản Kim Đồng	Hector Malot	Bìa Mềm	135000		Sửa Xóa
Olivia	NXB Kim Đồng	Nhà Xuất Bản Kim Đồng	Dorothy Strachey	Bìa Mềm	45000		Sửa Xóa
Hai Vạn Dặm Dưới Đáy Biển (Tái Bản 2020)	NXB Văn Học	Huy Hoang Bookstore	Jules Verne	Bìa Mềm	88200		Sửa Xóa
Chuyến Du Hành Vào Lòng Đất	Thanh Niên	Đinh Tị	Jules Verne	Bìa Mềm	93130		Sửa Xóa

Thực hiện các chức năng Thêm, Xóa, Sửa.

■ Yêu cầu 2:

Bước 1: Tạo trang Quản lý sách:

- Tạo Route cho trang Quản lý sách:

```
Route::get('/book/list', 'App\Http\Controllers\BookController@booklist')  
->middleware('auth')->name("booklist");
```

- Tạo hàm booklist() tại file BookController

```
public function booklist(){  
    $data = DB::table("sach")->get();  
    return view("vidusach.book_list", compact("data"));  
}
```

- Tạo file vidusach/book_list với nội dung tại file book_list.txt

■ Yêu cầu 2:

Bước 1: Tạo trang Quản lý sách

- Tạo Route cho trang xử lý thêm, xóa, sửa:

```
Route::get('/book/create', 'App\Http\Controllers\BookController@bookcreate')  
    ->middleware('auth')->name("bookcreate");
```

```
Route::get('/book/edit/{id}', 'App\Http\Controllers\BookController@bookedit')  
    ->middleware('auth')->name("bookedit");
```

```
Route::post('/book/save/{action}', 'App\Http\Controllers\BookController@booksave'  
)  
    ->middleware('auth')->name("booksave");
```

```
Route::post('/book/delete', 'App\Http\Controllers\BookController@bookdelete')  
    ->middleware('auth')->name("bookdelete");
```

KẾT QUẢ

Trang chủ

Thông tin tài khoản

Quản lý sách

QUẢN LÝ SÁCH

Thêm

Tiêu đề	Nhà xuất bản	Nhà cung cấp	Tác giả	Hình thức bìa	Giá bán	Hình ảnh	Thao tác
Cây Cam Ngọt Của Tôi	NXB Hội Nhà Văn	Nhã Nam	José Mauro de Vasconcelos	Bìa Mềm	84240		Sửa Xóa
Nhà Giả Kim (Tái Bản 2020)	NXB Hội Nhà Văn	Nhã Nam	Paulo Coelho	Bìa Mềm	61620		Sửa Xóa
Mùa Hè Không Tên - Tặng Kèm Bookmark 2 Mặt + Poster Tranh + Chữ Ký Tác Giả Ngẫu Nhiên	Trẻ	NXB Trẻ	Nguyễn Nhật Ánh	Bìa Mềm	101400		Sửa Xóa
Trường Ca Achilles	NXB Kim Đồng	Nhà Xuất Bản Kim Đồng	Madeline Miller	Bìa Mềm	140400		Sửa Xóa

■ Yêu cầu 2:

Bước 1: Tạo trang Quản lý sách

- Sử dụng thư viện DataTable:

Điều chỉnh file `account-panel.blade.php` theo nội dung của file `account-panel_2.txt` để bổ sung thêm các thư viện.

Tại file `book_list` thêm nội dung sau:

```
<script>
```

```
    $(document).ready(function(){  
        new DataTable('#book-table', {  
            responsive: true,  
            "bStateSave":true
```

```
        });
```

```
    });
```

```
</script>
```

Sử dụng thư viện
DataTable cho bảng có id
là book-table

DataTable sẽ lưu lại các
trạng thái trong quá trình
tương tác với table, ví dụ
như tìm kiếm

■ Yêu cầu 2:

Bước 1: Tạo trang quản lý sách:

- Ngoài ra, có thể sử dụng DataTable theo cách sau:

```
<script>
    $(document).ready(function(){
        $('#book-table').DataTable({
            responsive: true,
            "bStateSave":true
        });
    });
</script>
```

Sử dụng thư viện
DataTable cho bảng có id
là book-table

DataTable sẽ lưu lại các
trạng thái trong quá trình
tương tác với table, ví dụ
như tìm kiếm

THỰC HÀNH 3



KẾT QUẢ

Trang chủ

Thông tin tài khoản

Quản lý sách

QUẢN LÝ SÁCH

Thêm

10 entries per page

Search:

Tiêu đề	Nhà xuất bản	Nhà cung cấp	Tác giả	Hình thức bìa	Giá bán	Hình ảnh	Thao tác
999 Lá Thư Gửi Cho Chính Mình - Những Lá Thư Ẩn Tượng Nhất (Phiên Bản Song Ngữ Trung - Việt)	Thanh Niên	AZ Việt Nam	Miêu Công Tử	Bìa Mềm	77220		Sửa Xóa
999 Lá Thư Gửi Cho Chính Mình - Những Lá Thư Ẩn Tượng Nhất - Tập 2 (Phiên Bản Song Ngữ Trung-Việt)	Thanh Niên	AZ Việt Nam	Miêu Công Tử	Bìa Mềm	77220		Sửa Xóa

Yêu cầu 2:

Bước 2: Tạo trang Thêm dữ liệu sách

- Route cho trang Thêm dữ liệu sách (đã thực hiện ở bước 1)

```
Route::get('/book/create','App\Http\Controllers\BookController@bookcreate')  
->middleware('auth')->name("bookcreate");
```

- Tạo hàm bookcreate() tại file BookController

```
public function bookcreate(){  
    $the_loai = DB::table("dm_the_loai")->get();  
    $action = "add";  
    return view("vidusach.book_form",compact("the_loai","action"));  
}
```

Lấy dữ liệu thể loại sách và gửi đến book_form để chọn thể loại khi thêm sách

\$action = "add" →
book_form sẽ sử dụng cho Trang thêm dữ liệu sách

view book_form được sử dụng chung cho trang Thêm dữ liệu và Cập nhật dữ liệu → do vậy cần tạo biến \$action và gửi đến book_form để biết trang đang sử dụng book_form.

■ Yêu cầu 2:

Bước 2: Tạo trang Thêm dữ liệu sách

- Tạo file view vidusach/book_form.blade.php với nội dung sau:

```
<x-account-panel>
  <div class="panel panel-default" style="width:50%; margin:0 auto;">
    <div class="panel-body">
      <form action="{{route('booksave', ['action'=>$action])}}" method = "post" enctype="multipart/form-data">
        <div style='text-align:center;font-weight:bold;color:#15c;'>THÊM THÔNG TIN SÁCH</div>
        <label>Tiêu đề</label>
        <input type='text' class='form-control form-control-sm' name='tieu_de' value="">
        <label>Nhà xuất bản</label>
        <input type='text' class='form-control form-control-sm' name='nha_xuat_ban' value="">
        <label>Nhà cung cấp</label>
        <input type='text' class='form-control form-control-sm' name='nha_cung_cap' value="">
        <label>Tác giả</label>
        <input type='text' class='form-control form-control-sm' name='tac_gia' value="">
        <label>Hình thức bìa</label>
        <input type='text' class='form-control form-control-sm' name='hinh_thuc_bia' value="">
        <label>Giá bán</label>
        <input type='text' class='form-control form-control-sm' name='gia_ban' value="">
        <label>Thể loại</label>
        <select name='the_loai' class='form-control form-control-sm'>
          @foreach($the_loai as $row)
            <option value='{{ $row->id }}'>
              {{ $row->ten_the_loai }}
            </option>
          @endforeach
        </select>
        <label>Ảnh đại diện</label><br>
        <input type="file" name="file_anh_bia" accept="image/*" class="form-control-file">
        {{ csrf_field() }}
        <div style='text-align:center;'><input type='submit' class='btn btn-primary mt-1' value='Lưu'></div>
      </form>
    </div>
  </div>
</x-account-panel>
```

Truyền tham số cho route('booksave') vì tại file web.php được khai báo có tham số

Giá trị thuộc tính name tại các ô nhập thông tin tương ứng với tên các cột trong bảng dữ liệu để giúp xử lý nhanh việc lấy dữ liệu và insert dữ liệu tại file Controller

■ Yêu cầu 2:

Bước 2: Tạo trang Thêm dữ liệu sách

- Tạo hàm booksave() tại file BookController

```
public function booksave($action, Request $request)
{
    $request->validate([
        'tieu_de' => ['required', 'string', 'max:200'],
        'nha_cung_cap' => ['required', 'string', 'max:50'],
        'nha_xuat_ban' => ['required', 'string', 'max:50'],
        'tac_gia' => ['required', 'string', 'max:50'],
        'hinh_thuc_bia' => ['required', 'string', 'max:50'],
        'gia_ban' => ['required', 'numeric'],
        'the_loai' => ['required', 'max:3'],
        'file_anh_bia' => ['nullable', 'image']
    ]);

    $data = $request->except("_token");

    if($request->hasFile("file_anh_bia"))
    {
        $fileName = $request->input("tieu_de") . "_" . rand(1000000,9999999) . "." . $request->file('file_anh_bia')->extension();
        $request->file('file_anh_bia')->storeAs('public/book_image', $fileName);
        $data['file_anh_bia'] = $fileName;
    }

    $message = "";
    if($action=="add")
    {
        DB::table("sach")->insert($data);
        $message = "Thêm thành công";
    }
    return redirect()->route('booklist')->with('status', $message);
}
```

Kiểm tra dữ liệu do người dùng nhập theo các điều kiện

Lấy dữ liệu người dùng nhập, ngoại trừ dữ liệu token vì không cần lưu dữ liệu token

Lấy tên file ảnh để lưu vào cơ sở dữ liệu

Insert dữ liệu vào bảng sách

Chuyển tới trang booklist (quản lý sách) kèm thông báo

■ Yêu cầu 2:

Bước 2: Tạo trang Thêm dữ liệu sách

- File ảnh của cuốn sách khi thêm mới sẽ được lưu vào thư mục `storage/app/public/book_image` → Chuyển ảnh của các cuốn sách đã có trong dữ liệu từ thư mục `public/book_image` sang thư mục `storage/app/public/book_image`
- Tại file `vidusach/book_list.blade.php` điều chỉnh lại đoạn code hiển thị hình ảnh cuốn sách như sau:

```

```


■ Yêu cầu 2:

Bước 2: Tạo trang Thêm dữ liệu sách

- Để hiển thị thông báo sau khi thêm dữ liệu tại trang booklist → thêm tại file vidusach/book_list.blade.php:

```
@if (session('status'))  
    <div class="alert alert-success">  
        {{ session('status') }}  
    </div>  
@endif
```

- Đặt đoạn code trên tại vị trí muốn hiển thị thông báo

▪ Yêu cầu 2:

Bước 2: Tạo trang Thêm dữ liệu sách

- Để hiển thị thông báo lỗi tại trang Thêm dữ liệu sách khi người dùng nhập dữ liệu không hợp lệ → thêm tại file vidusach/book_form.blade.php đoạn code sau:

```
@if ($errors->any())
    <div style='color:red; margin:0 auto'>
        <div >
            {{ __('Whoops! Something went wrong.') }}
        </div>

        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif
```

- Đặt đoạn code trên tại vị trí muốn hiển thị thông báo

■ Yêu cầu 2:

Bước 3: Tạo trang Cập nhật dữ liệu sách

- Route cho trang Cập nhật dữ liệu sách (đã tạo ở bước 1)

```
Route::get('/book/edit/{id}', 'App\Http\Controllers\BookController@bookedit')  
->middleware('auth')->name("bookedit");
```

Route cho trang Cập nhật phải có tham số là id của cuốn sách cần cập nhật

- Tạo hàm bookedit() tại file BookController

```
public function bookedit($id){  
    $action = "edit";  
    $the_loai = DB::table("dm_the_loai")->get();  
    $sach = DB::table("sach")->where("id",$id)->first();  
    return view("vidusach.book_form",compact("the_loai","action","sach"));  
}
```

Lấy dữ liệu sách và gửi đến book_form để hiển thị dữ liệu cho người dùng cập nhật

\$action = "edit" → book_form sẽ sử dụng cho Trang Cập nhật dữ liệu sách

■ Yêu cầu 2:

Bước 3: Tạo trang Cập nhật dữ liệu sách

- Tại file vidusach/book_form.blade.php điều chỉnh lại code như sau:

```
<form action="{{route('booksave',['action'=>$action])}}" method="post" enctype="multipart/form-data">
    @if($action=="add")
        <div style='text-align:center;font-weight:bold;color:#15c;'>THÊM THÔNG TIN SÁCH</div>
    @else
        <div style='text-align:center;font-weight:bold;color:#15c;'>SỬA THÔNG TIN SÁCH</div>
    @endif
    <label>Tiêu đề</label>
    <input type='text' class='form-control form-control-sm' name='tieu_de' value="{{ $sach->tieu_de }}">
    <label>Nhà xuất bản</label>
    <input type='text' class='form-control form-control-sm' name='nha_xuat_ban' value="{{ $sach->nha_xuat_ban }}">
    <label>Nhà cung cấp</label>
    <input type='text' class='form-control form-control-sm' name='nha_cung_cap' value="{{ $sach->nha_cung_cap }}">
    <label>Tác giả</label>
    <input type='text' class='form-control form-control-sm' name='tac_gia' value="{{ $sach->tac_gia }}">
    <label>Hình thức bìa</label>
    <input type='text' class='form-control form-control-sm' name='hinh_thuc_bia' value="{{ $sach->hinh_thuc_bia }}">
    <label>Giá bán</label>
    <input type='text' class='form-control form-control-sm' name='gia_ban' value="{{ $sach->gia_ban }}">
    <label>Thể loại</label>
    <select name='the_loai' class='form-control form-control-sm'>
        @php
            $selected = isset($sach->the_loai)?$sach->the_loai:"";
        @endphp
        @foreach($the_loai as $row)
            <option value="{{ $row->id }}" {{ $selected==$row->id?'selected':'' }}>
                {{ $row->ten_the_loai }}
            </option>
        @endforeach
    </select>

    <label>Ảnh đại diện</label><br>
    @if($action=="edit")
        
        <input type='hidden' value="{{ $sach->id }}" name='id'>
    @endif
    <input type="file" name="file_anh_bia" accept="image/*" class="form-control-file">
    {{ csrf_field() }}
    <div style='text-align:center;'><input type='submit' class='btn btn-primary mt-1' value='Lưu'></div>
</form>
```

Thay đổi tiêu đề dựa vào giá trị của biến \$action.

Cú pháp để kiểm tra sự tồn tại của một biến và in ra giá trị của biến nếu biến có tồn tại.

Ví dụ:

\$sach->tieu_de??" : nếu biến \$sach->tieu_de có tồn tại thì in ra giá trị của biến ngược lại in ra "

Lấy thể loại của sách và điều chỉnh select chọn đúng thể loại của cuốn sách đang cập nhật

Hiển thị hình ảnh cuốn sách và tạo input hidden để lưu id của cuốn sách đang cập nhật nhằm mục đích xác định được id của cuốn sách khi cập nhật dữ liệu

■ Yêu cầu 2:

Bước 3: Tạo trang Cập nhật dữ liệu sách

- Điều chỉnh hàm booksave() tại file BookController:

```
public function booksave($action, Request $request)
{
    $request->validate([
        'tieu_de' => ['required', 'string', 'max:200'],
        'nha_cung_cap' => ['required', 'string', 'max:50'],
        'nha_xuat_ban' => ['required', 'string', 'max:50'],
        'tac_gia' => ['required', 'string', 'max:50'],
        'hinh_thuc_bia' => ['required', 'string', 'max:50'],
        'gia_ban' => ['required', 'numeric'],
        'the_loai' => ['required', 'max:3'],
        'file_anh_bia' => ['nullable', 'image']
    ]);

    $data = $request->except("_token");

    if($action=="edit")
        $data = $request->except("_token", "id");

    if($request->hasFile("file_anh_bia"))
    {
        $fileName = $request->input("tieu_de") . "_" . rand(1000000,9999999) . "." . $request->file('file_anh_bia')->extension();
        $request->file('file_anh_bia')->storeAs('public/book_image', $fileName);
        $data['file_anh_bia'] = $fileName;
    }

    //$data = $request->except("id","_token");
    $message = "";
    if($action=="add")
    {
        DB::table("sach")->insert($data);
        $message = "Thêm thành công";
    }
    else if($action=="edit")
    {
        $id = $request->id;
        DB::table("sach")->where("id",$id)->update($data);
        $message = "Cập nhật thành công";
    }

    return redirect()->route('booklist')->with('status', $message);
}
```

Trường hợp lưu dữ liệu từ trang cập nhật (\$action=="edit") thì loại bỏ dữ liệu _token và id

Lấy dữ liệu id cuốn sách cần cập nhật

Cập nhật dữ liệu dựa vào id cuốn sách

■ Yêu cầu 2:

Bước 4: Xóa dữ liệu sách

- Chú ý nút Xóa tại file vidusach/book_list.blade.php:

```
<form method='post' action = "{{route('bookdelete')}}"  
    onsubmit="return confirm('Bạn có chắc chắn muốn xóa cuốn sách này không?');">  
    <input type='hidden' value='{{$row->id}}' name='id'>  
    <input type='submit' class='btn btn-sm btn-danger' value='Xóa'>  
    {{ csrf_field() }}  
</form>
```

Trước khi thực hiện Xóa sẽ
hiển thị một hộp thoại để
người xác nhận

Nút xóa cần đặt trong form vì phương
thức khi xóa dữ liệu là POST. Không sử
dụng phương thức GET khi xóa dữ liệu

■ Yêu cầu 2:

Bước 4: Xóa dữ liệu sách

- Tạo hàm bookdelete tại file BookController:

```
public function bookdelete(Request $request)
{
    $id = $request->id;
    DB::table("sach")->where("id",$id)->delete();
    return redirect()->route('booklist')->with('status', "Xóa thành công");
}
```




HỎI & ĐÁP