

**\*\*You:\*\***

```
Route::middleware('auth:admin')->prefix('admin')->name('admin.')->group(function () { Route::get('verify-email', EmailVerificationPromptController::class) ->name('verification.notice'); Route::get('verify-email/{id}/{hash}', VerifyEmailController::class) ->middleware(['signed', 'throttle:6,1']) ->name('verification.verify'); Route::post('email/verification-notification', [EmailVerificationNotificationController::class, 'store']) ->middleware('throttle:6,1') ->name('verification.send'); Route::get('confirm-password', [ConfirmablePasswordController::class, 'show']) ->name('password.confirm'); Route::post('confirm-password', [ConfirmablePasswordController::class, 'store']); Route::put('password', [PasswordController::class, 'update'])->name('password.update'); Route::post('logout', [AuthenticatedSessionController::class, 'destroy']) ->name('logout'); //all categories routes Route::get('categories', [CategoryController::class, 'index'])->name('categories.index'); Route::post('categories', [CategoryController::class, 'categoryStore'])->name('categories.store'); Route::put('categories/{id}', [CategoryController::class, 'categoryUpdate'])->name('categories.update'); Route::delete('categories/{id}', [CategoryController::class, 'categoryDestroy'])->name('categories.destroy'); Route::post('sub/categories', [CategoryController::class, 'subCategoryStore'])->name('sub.categories.store'); Route::put('sub/categories/{id}', [CategoryController::class, 'subCategoryUpdate'])->name('sub.categories.update'); Route::delete('sub/categories/{id}', [CategoryController::class, 'subCategoryDestroy'])->name('sub.categories.destroy'); Route::get('sub/categories/dropdown', [CategoryController::class, 'subCategoryDropdown'])->name('sub.categories.dropdown'); Route::post('child/categories', [CategoryController::class, 'childCategoryStore'])->name('child.categories.store'); Route::put('child/categories/{id}', [CategoryController::class, 'childCategoryUpdate'])->name('child.categories.update'); Route::delete('child/categories/{id}', [CategoryController::class, 'childCategoryDestroy'])->name('child.categories.destroy'); Route::get('child/categories/dropdown', [CategoryController::class, 'childCategoryDropdown'])->name('child.categories.dropdown'); //Product Multiimage Route::post('multi-image/store', [ProductController::class, 'multiImageStore'])->name('multi-image.store'); Route::get('multi-image/update', [ProductController::class, 'multiImageUpdate'])->name('multi-image.update'); Route::get('web-setting', [WebSettingController::class, 'index'])->name('web.setting.index'); Route::put('seo/setting', [WebSettingController::class, 'seo'])->name('seo.setting'); Route::put('smtp/setting', [WebSettingController::class, 'smtp'])->name('smtp.setting'); Route::put('site/setting', [WebSettingController::class, 'site'])->name('site.setting'); Route::resources([ 'brand' => BrandController::class, 'group-permission' => GroupPermissionController::class, 'permission' => PermissionController::class, 'role' => RoleController::class, 'supplier' => SupplierController::class, 'warehouse' => WareHouseController::class, 'pick-up-point' => PickupPointController::class, 'payment-method' => PaymentMethodController::class, 'product' => ProductController::class, 'coupon' => CouponController::class, 'advertisement' => AdvertisementController::class, 'filtering-item' => FilteringItemController::class, 'review' => ReviewController::class, 'employee' => EmployeePermissionController::class, 'rfq' => RfqController::class, 'pre-order' => PreOrderController::class, 'pre-booking' => PreBookingController::class, 'order' => OrderController::class, 'order-item' => OrderItemController::class, 'billing-address' => BillingAddressController::class, 'shipping-address' => ShippingAddressController::class, 'product-draft' => ProductDraftController::class, 'contact' => ContactController::class, 'faq' => FaqController::class, 'newsletter' => NewsletterController::class, 'feedback' => FeedbackController::class, ], [ 'except' => ['brand', 'create' . 'show' . 'edit'], 'except' => ['group-permission', 'create' . 'show' . 'edit'], 'except' => ['employee', 'create' . 'show' . 'edit'], 'except' => ['payment-method', 'create' . 'show' . 'edit'], 'except' => ['coupon', 'create' . 'show' . 'edit'], 'except' => ['advertisement', 'create' . 'show' . 'edit'], 'except' => ['filtering-item', 'show'], ]); });
```

-----  
**\*\*ChatGPT:\*\***

This code snippet represents a set of routes defined in a Laravel web application. Here's a breakdown of the routes and their corresponding controllers:

## 1. Authentication Routes:

- `verify-email`: GET route to display the email verification prompt.
- `verify-email/{id}/{hash}`: GET route to verify the user's email address.
- `email/verification-notification`: POST route to send the email verification notification.
- `confirm-password`: GET and POST routes for confirming the user's password.
- `password`: PUT route to update the user's password.
- `logout`: POST route to log out the user.

## 2. Category Routes:

- `categories`: GET and POST routes for managing categories.
- `categories/{id}`: PUT and DELETE routes to update and delete specific categories.
- `sub/categories`: POST route to store subcategories.
- `sub/categories/{id}`: PUT and DELETE routes to update and delete subcategories.
- `sub/categories/dropdown`: GET route to fetch subcategory dropdown data.
- `child/categories`: POST route to store child categories.
- `child/categories/{id}`: PUT and DELETE routes to update and delete child categories.
- `child/categories/dropdown`: GET route to fetch child category dropdown data.

## 3. Product Multiimage Routes:

- `multi-image/store`: POST route to store product multi-images.
- `multi-image/update`: GET route to update product multi-images.

## 4. Web Setting Routes:

- `web-setting`: GET route to display web settings.
- `seo/setting`: PUT route to update SEO settings.
- `smtp/setting`: PUT route to update SMTP settings.
- `site/setting`: PUT route to update site settings.

## 5. Resource Routes: These routes use the `Route::resources` method to define resourceful routes for various controllers. The controllers include:

- `BrandController`
- `GroupPermissionController`
- `PermissionController`
- `RoleController`
- `SupplierController`
- `WarehouseController`
- `PickUpPointController`
- `PaymentMethodController`
- `ProductController`
- `CouponController`
- `AdvertisementController`
- `FilteringItemController`
- `ReviewController`
- `EmployeePermissionController`
- `RfqController`
- `PreOrderController`
- `PreBookingController`
- `OrderController`

- OrderItemController
- BillingAddressController
- ShippingAddressController
- ProductDraftController
- ContactController
- FaqController
- NewsletterController
- FeedbackController

The `except` option is used to exclude certain default resource routes like `create`, `show`, and `edit` for specific controllers.

These routes are prefixed with `/admin` and are protected by the `auth:admin` middleware, ensuring that only authenticated admin users can access them. The routes are also named using the `name` method, allowing for easy route referencing in the application.

-----  
 \*\*You:\*\*

These are all of my routes. Please create a document for the tester who will be testing them and provide a detailed guide with step-by-step instructions.

-----  
 \*\*ChatGPT:\*\*

Sure! Here's a detailed guide with step-by-step instructions for testing the routes in your Laravel application:

## Route Testing Guide

### Prerequisites:

- Make sure you have a running instance of the Laravel application.
- Ensure that the necessary test data is available in your application's database.

### Testing the Authentication Routes:

1. Open a web browser and navigate to the application's homepage.
2. Click on the "Login" or "Sign In" link to access the login page.
3. Enter the credentials of an admin user and click the "Login" button.
4. Verify that you are redirected to the admin dashboard or the designated landing page for authenticated admin users.
5. Test the following routes related to email verification:
  - Visit the URL `/admin/verify-email` and ensure that the email verification prompt is displayed.
  - Visit the URL `/admin/verify-email/{id}/{hash}` (replace `{id}` and `{hash}` with valid values) to verify the email address associated with the given ID and hash.
  - Submit a POST request to `/admin/email/verification-notification` to send an email verification notification.
  - Check your email inbox for the verification email and click on the verification link.
6. Test the password confirmation routes:

- Visit the URL `/admin/confirm-password` and confirm that the password confirmation screen is displayed.
  - Enter your password and submit the form to proceed.
7. Test the password update route:
- Submit a PUT request to `/admin/password` with the necessary parameters to update your password.
8. Test the logout route:
- Submit a POST request to `/admin/logout` to log out the currently logged-in admin user.

## Testing the Category Routes:

1. Ensure that you are logged in as an authenticated admin user.
2. Test the following category routes:
  - Visit the URL `/admin/categories` to view a list of categories.
  - Submit a POST request to `/admin/categories` with the necessary parameters to create a new category.
  - Submit a PUT request to `/admin/categories/{id}` (replace `{id}` with the ID of an existing category) to update the specified category.
  - Submit a DELETE request to `/admin/categories/{id}` to delete the specified category.
  - Submit a POST request to `/admin/sub/categories` with the necessary parameters to create a new subcategory.
  - Submit a PUT request to `/admin/sub/categories/{id}` to update the specified subcategory.
  - Submit a DELETE request to `/admin/sub/categories/{id}` to delete the specified subcategory.
  - Visit the URL `/admin/sub/categories/dropdown` to fetch the dropdown data for subcategories.
  - Submit a POST request to `/admin/child/categories` with the necessary parameters to create a new child category.
  - Submit a PUT request to `/admin/child/categories/{id}` to update the specified child category.
  - Submit a DELETE request to `/admin/child/categories/{id}` to delete the specified child category.
  - Visit the URL `/admin/child/categories/dropdown` to fetch the dropdown data for child categories.

## Testing the Product Multiimage Routes:

1. Ensure that you are logged in as an authenticated admin user.
2. Test the following product multiimage routes:
  - Submit a POST request to `/admin/multi-image/store` with the necessary parameters to store product multi-images.
  - Visit the URL `/admin/multi-image/update` to update product multi-images.

## Testing the Web Setting Routes:

1. Ensure that you are logged in as an authenticated admin user.
2. Test the following web setting routes:
  - Visit the URL `/admin/web-setting` to view and modify web settings.
  - Submit a PUT request to `/admin/seo/setting` to update SEO settings.
  - Submit a PUT request to `/admin/smtp/setting` to update SMTP settings.
  - Submit a PUT request to `/admin/site/setting` to update site settings.

## Testing the Resource Routes:

1. Ensure that you are logged in as an authenticated admin user.
2. Test the following resource routes:
  - Each resource route represents a set of routes for a specific controller.
  - For each resource route, follow these steps:
    - Visit the URL `/admin/{resource}` (replace `{resource}` with the corresponding resource name) to view a list of items.
    - Optionally, submit a GET request to `/admin/{resource}/create` to access the creation form.
    - Optionally, submit a GET request to `/admin/{resource}/{id}` (replace `{id}` with the ID of an existing item) to view a specific item.
    - Optionally, submit a GET request to `/admin/{resource}/{id}/edit` to access the edit form for a specific item.
    - Optionally, submit a POST request to `/admin/{resource}` with the necessary parameters to create a new item.
    - Optionally, submit a PUT request to `/admin/{resource}/{id}` to update a specific item.
    - Optionally, submit a DELETE request to `/admin/{resource}/{id}` to delete a specific item.
3. Repeat step 2 for each of the following resource routes:
  - brand
  - group-permission
  - permission
  - role
  - supplier
  - warehouse
  - pick-up-point
  - payment-method
  - product
  - coupon
  - advertisement
  - filtering-item
  - review
  - employee
  - rfq
  - pre-order
  - pre-booking
  - order
  - order-item
  - billing-address
  - shipping-address
  - product-draft
  - contact
  - faq
  - newsletter
  - feedback

Note: For resource routes, refer to the provided route definitions to identify any specific routes that are excluded from the default resource routes.

That's it! Follow these step-by-step instructions to thoroughly test the routes in your Laravel application. Make sure to verify that each route performs as expected and produces the desired results.

-----