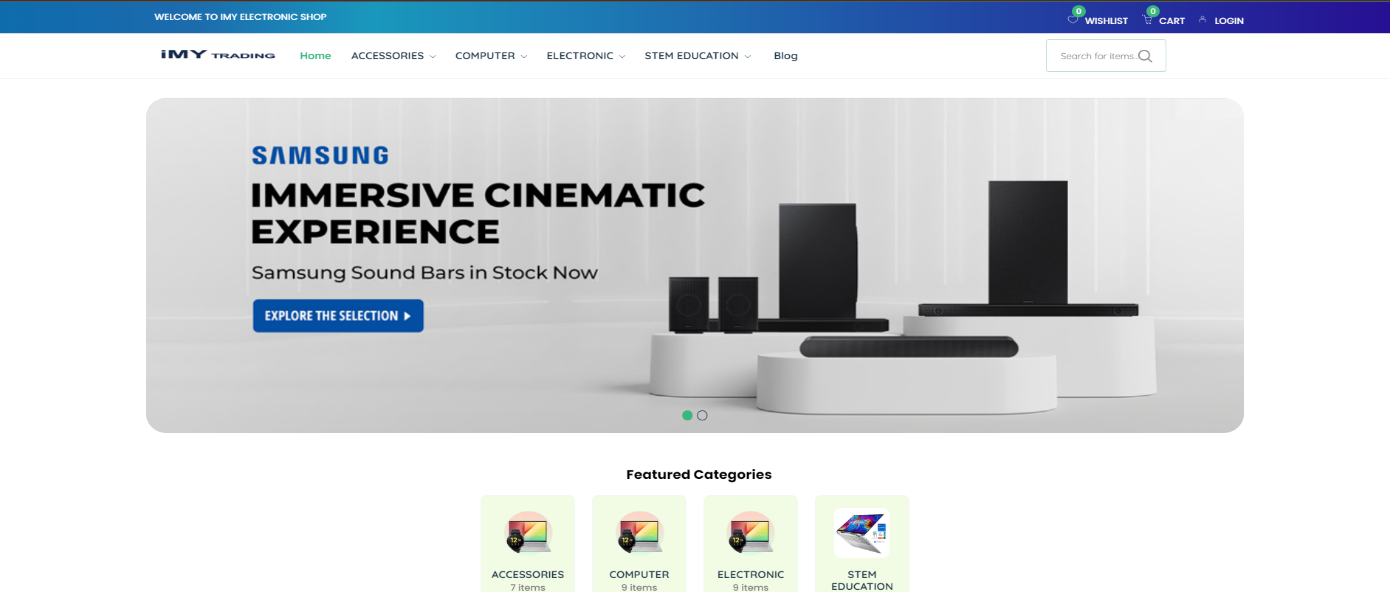
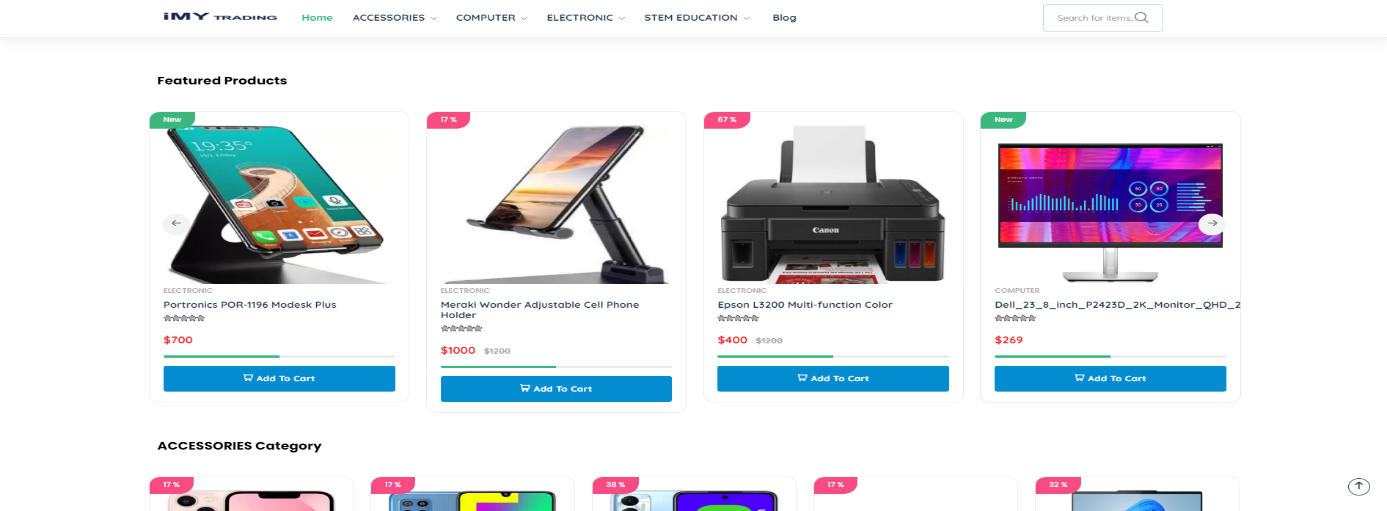
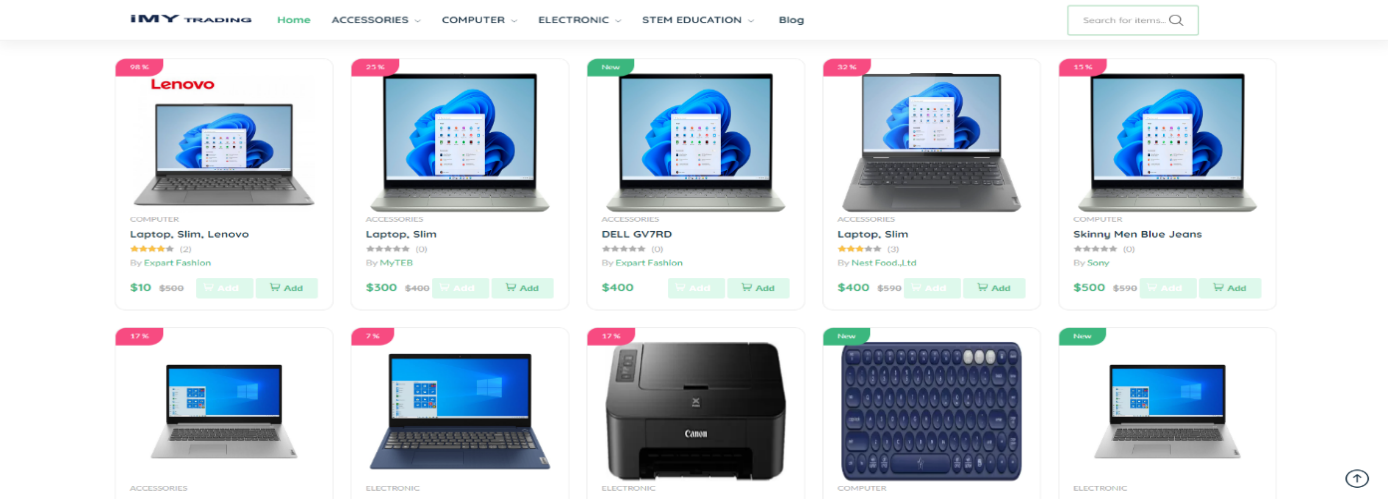
ឧបសម្ព័ន្ធ ក

Coding

ក្រុមសិក្សាស្រាវជ្រាវធ្វើការបង្ហាញអំពីការសរសេរកូដដែលមាននៅទំព័រនីមួយៗដែលក្រុមសិក្សាស្រាវជ្រាវ នឹងធ្វើការរៀបរាប់ទាំងការសរសេរ Front End និង Back End ដូចតទៅ៖

ក.១ ទំព័រដើមនៃគេហទំព័រ (Home Page)



រូបភាព ក.៥៤ បង្ហាញពី ទំព័រដើម

* IndexController.php

<?php

namespace App\Http\Controllers\Frontend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Category;

use App\Models\SubCategory;

use App\Models\MultiImg;

use App\Models\Brand;

use App\Models\Product;

use App\Models\User;

class IndexController extends Controller

{

public function Index(){

$skip\_category\_0 = Category::skip(0)->first();

$skip\_product\_0 = Product::where('status',1)->where('category\_id',$skip\_category\_0->id)->orderBy('id','DESC')->limit(5)->get();

$skip\_category\_1 = Category::skip(1)->first();

$skip\_product\_1 = Product::where('status',1)->where('category\_id',$skip\_category\_1->id)->orderBy('id','DESC')->limit(5)->get()

$skip\_category\_2 = Category::skip(2)->first();

$skip\_product\_2 = Product::where('status',1)->where('category\_id',$skip\_category\_2->id)->orderBy('id','DESC')->limit(5)->get();

$skip\_category\_3 = Category::skip(3)->first();

$skip\_product\_3 = Product::where('status',1)->where('category\_id',$skip\_category\_3->id)->orderBy('id','DESC')->limit(5)->get();

$hot\_deals = Product::where('hot\_deals',1)->where('discount\_price','!=',NULL)->orderBy('id','DESC')->limit(3)->get();

$special\_offer = Product::where('special\_offer',1)->orderBy('id','DESC')->limit(3)->get();

$new = Product::where('status',1)->orderBy('id','DESC')->limit(3)->get();

$special\_deals = Product::where('special\_deals',1)->orderBy('id','DESC')->limit(3)->get();

Return view('frontend.index',compact('skip\_category\_0','skip\_product\_0','skip\_category\_1','skip\_product\_1','skip\_category\_2','skip\_product\_2','skip\_category\_3','skip\_product\_3','hot\_deals','special\_offer','new','special\_deals'));

}

public function ProductDetails($id,$slug){

$product = Product::findOrFail($id);

$color = $product->product\_color;

$product\_color = explode(',', $color);

$size = $product->product\_size;

$product\_size = explode(',', $size);

$multiImage = MultiImg::where('product\_id',$id)->get();

$cat\_id = $product->category\_id;

$relatedProduct = Product::where('category\_id',$cat\_id)->where('id','!=',$id)->orderBy('id','DESC')->limit(4)->get();

return view('frontend.product.product\_details',compact('product','product\_color','product\_size','multiImage','relatedProduct'));

}

public function VendorDetails($id){

$vendor = User::findOrFail($id);

$vproduct = Product::where('vendor\_id',$id)->get();

return view('frontend.vendor.vendor\_details',compact('vendor','vproduct'));

}

public function VendorAll(){

$vendors = User::where('status','active')->where('role','vendor')->orderBy('id','DESC')->get();

return view('frontend.vendor.vendor\_all',compact('vendors'));

}

public function CatWiseProduct(Request $request,$id,$slug){

$products = Product::where('status',1)->where('category\_id',$id)->orderBy('id','DESC')->get();

$categories = Category::orderBy('category\_name','ASC')->get();

$breadcat = Category::where('id',$id)->first();

$newProduct = Product::orderBy('id','DESC')->limit(3)->get();

return view('frontend.product.category\_view',compact('products','categories','breadcat','newProduct'));

}

public function SubCatWiseProduct(Request $request,$id,$slug){

$products = Product::where('status',1)->where('subcategory\_id',$id)->orderBy('id','DESC')->get();

$categories = Category::orderBy('category\_name','ASC')->get();

$breadsubcat = SubCategory::where('id',$id)->first();

$newProduct = Product::orderBy('id','DESC')->limit(3)->get();

return view('frontend.product.subcategory\_view',compact('products','categories','breadsubcat','newProduct'));

}

public function ProductViewAjax($id){

$product = Product::with('category','brand')->findOrFail($id);

$color = $product->product\_color;

$product\_color = explode(',', $color);

$size = $product->product\_size;

$product\_size = explode(',', $size);

return response()->json(array(

'product' => $product,

'color' => $product\_color,

'size' => $product\_size,

));

}

public function ProductSearch(Request $request){

$request->validate(['search' => "required"]);

$item = $request->search;

$categories = Category::orderBy('category\_name','ASC')->get();

$products = Product::where('product\_name','LIKE',"%$item%")->get();

$newProduct = Product::orderBy('id','DESC')->limit(3)->get();

return view('frontend.product.search',compact('products','item','categories','newProduct'));

}

public function SearchProduct(Request $request){

$request->validate(['search' => "required"]);

$item = $request->search;

$products = Product::where('product\_name','LIKE',"%$item%")->select('product\_name','product\_slug','product\_thambnail','selling\_price','id')->limit(6)->get();

return view('frontend.product.search\_product',compact('products'));

}

}

ក.២ ផលិតផលតាមប្រភេទនីមួយៗ (Products by Category Page)

A screenshot of a computer

Description automatically generated

រូបភាព ក.៥៥ បង្ហាញពី Show product by category

* IndexController.php

<?php

namespace App\Http\Controllers\Frontend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Category;

use App\Models\SubCategory;

use App\Models\MultiImg;

use App\Models\Brand;

use App\Models\Product;

use App\Models\User;

class IndexController extends Controller

{

public function Index(){

$skip\_category\_0 = Category::skip(0)->first();

$skip\_product\_0 = Product::where('status',1)->where('category\_id',$skip\_category\_0->id)->orderBy('id','DESC')->limit(5)->get();

$skip\_category\_1 = Category::skip(1)->first();

$skip\_product\_1 = Product::where('status',1)->where('category\_id',$skip\_category\_1->id)->orderBy('id','DESC')->limit(5)->get();

$skip\_category\_2 = Category::skip(2)->first();

$skip\_product\_2 = Product::where('status',1)->where('category\_id',$skip\_category\_2->id)->orderBy('id','DESC')->limit(5)->get();

$skip\_category\_3 = Category::skip(3)->first();

$skip\_product\_3 = Product::where('status',1)->where('category\_id',$skip\_category\_3->id)->orderBy('id','DESC')->limit(5)->get();

$hot\_deals = Product::where('hot\_deals',1)->where('discount\_price','!=',NULL)->orderBy('id','DESC')->limit(3)->get();

$special\_offer = Product::where('special\_offer',1)->orderBy('id','DESC')->limit(3)->get();

$new = Product::where('status',1)->orderBy('id','DESC')->limit(3)->get();

$special\_deals = Product::where('special\_deals',1)->orderBy('id','DESC')->limit(3)->get();

return view('frontend.index',compact('skip\_category\_0','skip\_product\_0','skip\_category\_1','skip\_product\_1','skip\_category\_2','skip\_product\_2','skip\_category\_3','skip\_product\_3','hot\_deals','special\_offer','new','special\_deals'));

}

public function ProductDetails($id,$slug){

$product = Product::findOrFail($id);

$color = $product->product\_color;

$product\_color = explode(',', $color);

$size = $product->product\_size;

$product\_size = explode(',', $size);

$multiImage = MultiImg::where('product\_id',$id)->get();

$cat\_id = $product->category\_id;

$relatedProduct = Product::where('category\_id',$cat\_id)->where('id','!=',$id)->orderBy('id','DESC')->limit(4)->get();

return view('frontend.product.product\_details',compact('product','product\_color','product\_size','multiImage','relatedProduct'));

}

public function VendorDetails($id){

$vendor = User::findOrFail($id);

$vproduct = Product::where('vendor\_id',$id)->get();

return view('frontend.vendor.vendor\_details',compact('vendor','vproduct'));

}

public function VendorAll(){

$vendors = User::where('status','active')->where('role','vendor')->orderBy('id','DESC')->get();

return view('frontend.vendor.vendor\_all',compact('vendors'));

}

public function CatWiseProduct(Request $request,$id,$slug){

$products = Product::where('status',1)->where('category\_id',$id)->orderBy('id','DESC')->get();

$categories = Category::orderBy('category\_name','ASC')->get();

$breadcat = Category::where('id',$id)->first();

$newProduct = Product::orderBy('id','DESC')->limit(3)->get();

return view('frontend.product.category\_view',compact('products','categories','breadcat','newProduct'));

}

public function SubCatWiseProduct(Request $request,$id,$slug){

$products = Product::where('status',1)->where('subcategory\_id',$id)->orderBy('id','DESC')->get();

$categories = Category::orderBy('category\_name','ASC')->get();

$breadsubcat = SubCategory::where('id',$id)->first();

$newProduct = Product::orderBy('id','DESC')->limit(3)->get();

return view('frontend.product.subcategory\_view',compact('products','categories','breadsubcat','newProduct'));

}

public function ProductViewAjax($id){

$product = Product::with('category','brand')->findOrFail($id);

$color = $product->product\_color;

$product\_color = explode(',', $color);

$size = $product->product\_size;

$product\_size = explode(',', $size);

return response()->json(array(

'product' => $product,

'color' => $product\_color,

'size' => $product\_size,

));

}

public function ProductSearch(Request $request){

$request->validate(['search' => "required"]);

$item = $request->search;

$categories = Category::orderBy('category\_name','ASC')->get();

$products = Product::where('product\_name','LIKE',"%$item%")->get();

$newProduct = Product::orderBy('id','DESC')->limit(3)->get();

return view('frontend.product.search',compact('products','item','categories','newProduct'));

}

public function SearchProduct(Request $request){

$request->validate(['search' => "required"]);

$item = $request->search;

$products = Product::where('product\_name','LIKE',"%$item%")->select('product\_name','product\_slug','product\_thambnail','selling\_price','id')->limit(6)->get();

return view('frontend.product.search\_product',compact('products'));

}

}

A screenshot of a login form

Description automatically generatedក.៣ ទំព័រព័ត៌មានចុះឈ្មោះចូលក្នុងប្រព័ន្ធ (Register Page)

រូបភាព ក.៥៦ បង្ហាញពី Register Page

* RegisteredUserController.php

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;

use App\Models\User;

use App\Providers\RouteServiceProvider;

use Illuminate\Auth\Events\Registered;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Hash;

use Illuminate\Validation\Rules;

use App\Notifications\RegisterUserNotification;

use Illuminate\Support\Facades\Notification;

class RegisteredUserController extends Controller

{

public function create()

{

return view('auth.register');

}

public function store(Request $request)

{

$request->validate([

'name' => ['required', 'string', 'max:255'],

'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],

'password' => ['required', 'confirmed', Rules\Password::defaults()],

]);

$user = User::create([

'name' => $request->name,

'email' => $request->email,

'password' => Hash::make($request->password),

]);

event(new Registered($user));

Auth::login($user);

$nuser = User::where('role','admin')->get();

Notification::send($nuser, new RegisterUserNotification($request));

return redirect(RouteServiceProvider::HOME);

}

}

A screenshot of a computer

Description automatically generatedក.៤ ទំព័រព័ត៍មានគណនី (Account Details Page)

រូបភាព ក.៥៧ បង្ហាញពី Customer Profile Page

* UserController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use App\Models\User;

use Illuminate\Support\Facades\Hash;

class UserController extends Controller

{

public function UserDashboard(){

$id = Auth::user()->id;

$userData = User::find($id);

return view('index',compact('userData'));

} // End Method

public function UserProfileStore(Request $request){

$id = Auth::user()->id;

$data = User::find($id);

$data->name = $request->name;

$data->username = $request->username;

$data->email = $request->email;

$data->phone = $request->phone;

$data->address = $request->address;

if ($request->file('photo')) {

$file = $request->file('photo');

@unlink(public\_path('upload/user\_images/'.$data->photo));

$filename = date('YmdHi').$file->getClientOriginalName();

$file->move(public\_path('upload/user\_images'),$filename);

$data['photo'] = $filename;

}

$data->save();

$notification = array(

'message' => 'User Profile Updated Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

} // End Mehtod

public function UserLogout(Request $request){

Auth::guard('web')->logout();

$request->session()->invalidate();

$request->session()->regenerateToken();

$notification = array(

'message' => 'User Logout Successfully',

'alert-type' => 'success'

);

return redirect('/login')->with($notification);

} // End Mehtod

public function UserUpdatePassword(Request $request){

// Validation

$request->validate([

'old\_password' => 'required',

'new\_password' => 'required|confirmed',

]);

// Match The Old Password

if (!Hash::check($request->old\_password, auth::user()->password)) {

return back()->with("error", "Old Password Doesn't Match!!");

}

// Update The new password

User::whereId(auth()->user()->id)->update([

'password' => Hash::make($request->new\_password)

]);

return back()->with("status", " Password Changed Successfully");

} // End Mehtod

}

A screenshot of a computer

Description automatically generatedក.៥ ទំព័រផ្លាស់ប្ដូរលេខសម្ងាត់ (Change Password Page)

រូបភាព ក.៥៨ បង្ហាញពី Change Password Page

* UserController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use App\Models\User;

use Illuminate\Support\Facades\Hash;

class UserController extends Controller

{

public function UserDashboard(){

$id = Auth::user()->id;

$userData = User::find($id);

return view('index',compact('userData'));

} // End Method

public function UserProfileStore(Request $request){

$id = Auth::user()->id;

$data = User::find($id);

$data->name = $request->name;

$data->username = $request->username;

$data->email = $request->email;

$data->phone = $request->phone;

$data->address = $request->address;

if ($request->file('photo')) {

$file = $request->file('photo');

@unlink(public\_path('upload/user\_images/'.$data->photo));

$filename = date('YmdHi').$file->getClientOriginalName();

$file->move(public\_path('upload/user\_images'),$filename);

$data['photo'] = $filename;

}

$data->save();

$notification = array(

'message' => 'User Profile Updated Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

} // End Mehtod

public function UserLogout(Request $request){

Auth::guard('web')->logout();

$request->session()->invalidate();

$request->session()->regenerateToken();

$notification = array(

'message' => 'User Logout Successfully',

'alert-type' => 'success'

);

return redirect('/login')->with($notification);

} // End Mehtod public function UserUpdatePassword(Request $request){

// Validation

$request->validate([

'old\_password' => 'required',

'new\_password' => 'required|confirmed',

]);

// Match The Old Password

if (!Hash::check($request->old\_password, auth::user()->password)) {

return back()->with("error", "Old Password Doesn't Match!!");

}

// Update The new password

User::whereId(auth()->user()->id)->update([

'password' => Hash::make($request->new\_password)

]);

return back()->with("status", " Password Changed Successfully");

} // End Mehtod

}

ក.៦ ប្រវត្តិនៃការទិញឥវ៉ាន់ (Purchase History Page)

A screenshot of a computer

Description automatically generated

រូបភាព ក.៥៩ បង្ហាញពី Purchase History Page

* AllUserController.php

<?php

namespace App\Http\Controllers\User;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use App\Models\User;

use Illuminate\Support\Facades\Hash;

use App\Models\Order;

use App\Models\OrderItem;

use Barryvdh\DomPDF\Facade\Pdf;

use Carbon\Carbon;

class AllUserController extends Controller

{

public function UserAccount(){

$id = Auth::user()->id;

$userData = User::find($id);

return view('frontend.userdashboard.account\_details',compact('userData'));

} // End Method

public function UserChangePassword(){

return view('frontend.userdashboard.user\_change\_password' );

} // End Method

public function UserOrderPage(){

$id = Auth::user()->id;

$orders = Order::where('user\_id',$id)->orderBy('id','DESC')->get();

return view('frontend.userdashboard.user\_order\_page',compact('orders'));

}// End Method

public function UserOrderDetails($order\_id){

$order = Order::with('division','district','state','user')->where('id',$order\_id)->where('user\_id',Auth::id())->first();

$orderItem = OrderItem::with('product')->where('order\_id',$order\_id)->orderBy('id','DESC')->get();

return view('frontend.order.order\_details',compact('order','orderItem'));

}// End Method

public function UserOrderInvoice($order\_id){

$order = Order::with('division','district','state','user')->where('id',$order\_id)->where('user\_id',Auth::id())->first();

$orderItem = OrderItem::with('product')->where('order\_id',$order\_id)->orderBy('id','DESC')->get();

$pdf = Pdf::loadView('frontend.order.order\_invoice', compact('order','orderItem'))->setPaper('a4')->setOption([

'tempDir' => public\_path(),

'chroot' => public\_path(),

]);

return $pdf->download('invoice.pdf');

}// End Method

public function ReturnOrder(Request $request,$order\_id){

Order::findOrFail($order\_id)->update([

'return\_date' => Carbon::now()->format('d F Y'),

'return\_reason' => $request->return\_reason,

'return\_order' => 1,

]);

$notification = array(

'message' => 'Return Request Send Successfully',

'alert-type' => 'success'

);

return redirect()->route('user.order.page')->with($notification);

}// End Method

public function ReturnOrderPage(){

$orders = Order::where('user\_id',Auth::id())->where('return\_reason','!=',NULL)->orderBy('id','DESC')->get();

return view('frontend.order.return\_order\_view',compact('orders'));

}// End Method

public function UserTrackOrder(){

return view('frontend.userdashboard.user\_track\_order');

}// End Method

public function OrderTracking(Request $request){

$invoice = $request->code;

$track = Order::where('invoice\_no',$invoice)->first();

if ($track) {

return view('frontend.traking.track\_order',compact('track'));

} else{

$notification = array(

'message' => 'Invoice Code Is Invalid',

'alert-type' => 'error'

);

return redirect()->back()->with($notification);

}

}// End Method

}

A screenshot of a computer

Description automatically generatedក.៧ ទំព័រនៃការចំណាយរបស់អតិថិជន (Payment List Page)

រូបភាព ក.៦០ បង្ហាញពី Payment List Page

* AllUserController.php

<?php

namespace App\Http\Controllers\User;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use App\Models\User;

use Illuminate\Support\Facades\Hash;

use App\Models\Order;

use App\Models\OrderItem;

use Barryvdh\DomPDF\Facade\Pdf;

use Carbon\Carbon;

class AllUserController extends Controller

{

public function UserAccount(){

$id = Auth::user()->id;

$userData = User::find($id);

return view('frontend.userdashboard.account\_details',compact('userData'));

} // End Method

public function UserChangePassword(){

return view('frontend.userdashboard.user\_change\_password' );

} // End Method

public function UserOrderPage(){

$id = Auth::user()->id;

$orders = Order::where('user\_id',$id)->orderBy('id','DESC')->get();

return view('frontend.userdashboard.user\_order\_page',compact('orders'));

}// End Method

public function UserOrderDetails($order\_id){

$order = Order::with('division','district','state','user')->where('id',$order\_id)->where('user\_id',Auth::id())->first();

$orderItem = OrderItem::with('product')->where('order\_id',$order\_id)->orderBy('id','DESC')->get();

return view('frontend.order.order\_details',compact('order','orderItem'));

}// End Method

public function UserOrderInvoice($order\_id){

$order = Order::with('division','district','state','user')->where('id',$order\_id)->where('user\_id',Auth::id())->first();

$orderItem = OrderItem::with('product')->where('order\_id',$order\_id)->orderBy('id','DESC')->get();

$pdf = Pdf::loadView('frontend.order.order\_invoice', compact('order','orderItem'))->setPaper('a4')->setOption([

'tempDir' => public\_path(),

'chroot' => public\_path(),

]);

return $pdf->download('invoice.pdf');

}// End Method

public function ReturnOrder(Request $request,$order\_id){

Order::findOrFail($order\_id)->update([

'return\_date' => Carbon::now()->format('d F Y'),

'return\_reason' => $request->return\_reason,

'return\_order' => 1,

]);

$notification = array(

'message' => 'Return Request Send Successfully',

'alert-type' => 'success'

);

return redirect()->route('user.order.page')->with($notification);

}// End Method

public function ReturnOrderPage(){

$orders = Order::where('user\_id',Auth::id())->where('return\_reason','!=',NULL)->orderBy('id','DESC')->get();

return view('frontend.order.return\_order\_view',compact('orders'));

}// End Method

public function UserTrackOrder(){

return view('frontend.userdashboard.user\_track\_order');

}// End Method

public function OrderTracking(Request $request){

$invoice = $request->code;

$track = Order::where('invoice\_no',$invoice)->first();

if ($track) {

return view('frontend.traking.track\_order',compact('track'));

} else{

$notification = array(

'message' => 'Invoice Code Is Invalid',

'alert-type' => 'error'

);

return redirect()->back()->with($notification);

}

}// End Method

}

ក.៨ ទំព័រទំនិញជាទីពេញចិត្តរបស់អតិថិជន (Wishlist Page)

A screenshot of a website

Description automatically generated

រូបភាព ក.៦១ បង្ហាញពី Wishlist Page

* User/WishlistController.php

<?php

namespace App\Http\Controllers\User;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Wishlist;

use Illuminate\Support\Facades\Auth;

use Carbon\Carbon;

class WishlistController extends Controller

{

public function AddToWishList(Request $request, $product\_id){

if (Auth::check()) {

$exists = Wishlist::where('user\_id',Auth::id())->where('product\_id',$product\_id)->first();

if (!$exists) {

Wishlist::insert([

'user\_id' => Auth::id(),

'product\_id' => $product\_id,

'created\_at' => Carbon::now(),

]);

return response()->json(['success' => 'Successfully Added On Your Wishlist' ]);

} else{

return response()->json(['error' => 'This Product Has Already on Your Wishlist' ]);

}

}else{

return response()->json(['error' => 'At First Login Your Account' ]);

}

} // End Method

public function AllWishlist(){

return view('frontend.wishlist.view\_wishlist');

}// End Method

public function GetWishlistProduct(){

$wishlist = Wishlist::with('product')->where('user\_id',Auth::id())->latest()->get();

$wishQty = wishlist::count();

return response()->json(['wishlist'=> $wishlist, 'wishQty' => $wishQty]);

}// End Method

public function WishlistRemove($id){

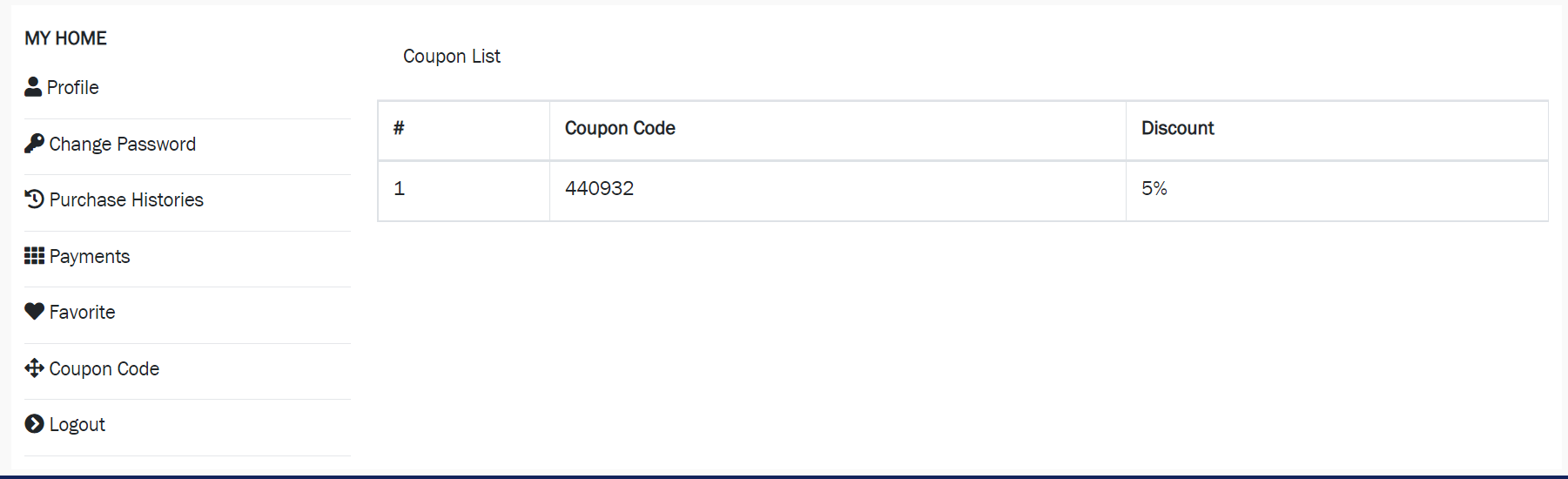
Wishlist::where('user\_id',Auth::id())->where('id',$id)->delete();

return response()->json(['success' => 'Successfully Product Remove' ]);

}// End Method

}

ក.៩ ទំព័រលេខកូដបញ្ចុះតម្លៃ (Coupon Code Page)



រូបភាព ក.៦២ បង្ហាញពី Coupon Page

* Customer/CouponController.php

<?php

namespace App\Http\Controllers\Front\Customer;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

use App\CustomerCoupon;

use App\Coupon;

use Cart;

class CouponController extends Controller

{

public function index(){

$coupon = auth()->guard('customer')

->user()

->customerCoupon()

->with('coupon')

->has('coupon')

->whereHas('coupon', function ($query) {

$query->where('expired\_date',">=",date("Y-m-d"));

})->get();

return view('frontends.customer.coupon',compact('coupon'));

}

public function store(){

if(Coupon::whereCouponCode(request()->coupon\_code)->whereDate("expired\_date",">",date('Y-m-d'))->exists()){

$coupon = auth()->guard('customer')->user()->customerCoupon()->with('coupon')->has('coupon')->latest()->first();

$coupon->coupon->save();

$discount = $coupon->coupon->discount;

// $coupon->save();

Cart::setGlobalDiscount($discount);

session()->put('coupon',request()->coupon\_code);

return back(); }

}

}

ក.១០ ទំព័រអែដធូកាត (Add to Cart Page)

A screenshot of a computer

Description automatically generated

រូបភាព ក.៦៣ បង្ហាញពី Add to Cart Page

* CartController.blade.php

<?php

namespace App\Http\Controllers\Frontend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Product;

use App\Models\Coupon;

use App\Models\ShipDivision;

use Gloudemans\Shoppingcart\Facades\Cart;

use Illuminate\Support\Facades\Session;

use Carbon\Carbon;

use Auth;

class CartController extends Controller

{

public function AddToCart(Request $request, $id){

if(Session::has('coupon')){

Session::forget('coupon');

}

$product = Product::findOrFail($id);

if ($product->discount\_price == NULL) {

Cart::add([

'id' => $id,

'name' => $request->product\_name,

'qty' => $request->quantity,

'price' => $product->selling\_price,

'weight' => 1,

'options' => [

'image' => $product->product\_thambnail,

'color' => $request->color,

'size' => $request->size,

'vendor' => $request->vendor,

],

]);

return response()->json(['success' => 'Successfully Added on Your Cart' ]);

}else{

Cart::add([

'id' => $id,

'name' => $request->product\_name,

'qty' => $request->quantity,

'price' => $product->discount\_price,

'weight' => 1,

'options' => [

'image' => $product->product\_thambnail,

'color' => $request->color,

'size' => $request->size,

'vendor' => $request->vendor,

],

]);

return response()->json(['success' => 'Successfully Added on Your Cart' ]);

}

}// End Method

public function AddToCartDetails(Request $request, $id){

if(Session::has('coupon')){

Session::forget('coupon');

}

$product = Product::findOrFail($id);

if ($product->discount\_price == NULL) {

Cart::add([

'id' => $id,

'name' => $request->product\_name,

'qty' => $request->quantity,

'price' => $product->selling\_price,

'weight' => 1,

'options' => [

'image' => $product->product\_thambnail,

'color' => $request->color,

'size' => $request->size,

'vendor' => $request->vendor,

],

]);

return response()->json(['success' => 'Successfully Added on Your Cart' ]);

}else{

Cart::add([

'id' => $id,

'name' => $request->product\_name,

'qty' => $request->quantity,

'price' => $product->discount\_price,

'weight' => 1,

'options' => [

'image' => $product->product\_thambnail,

'color' => $request->color,

'size' => $request->size,

'vendor' => $request->vendor,

],

]);

return response()->json(['success' => 'Successfully Added on Your Cart' ]);

}

}// End Method

public function AddMiniCart(){

$carts = Cart::content();

$cartQty = Cart::count();

$cartTotal = Cart::total();

return response()->json(array(

'carts' => $carts,

'cartQty' => $cartQty,

'cartTotal' => $cartTotal

));

}// End Method

public function RemoveMiniCart($rowId){

Cart::remove($rowId);

return response()->json(['success' => 'Product Remove From Cart']);

}// End Method

public function MyCart(){

return view('frontend.mycart.view\_mycart');

}// End Method

public function GetCartProduct(){

$carts = Cart::content();

$cartQty = Cart::count();

$cartTotal = Cart::total();

return response()->json(array(

'carts' => $carts,

'cartQty' => $cartQty,

'cartTotal' => $cartTotal

));

}// End Method

public function CartRemove($rowId){

Cart::remove($rowId);

if(Session::has('coupon')){

$coupon\_name = Session::get('coupon')['coupon\_name'];

$coupon = Coupon::where('coupon\_name',$coupon\_name)->first();

Session::put('coupon',[

'coupon\_name' => $coupon->coupon\_name,

'coupon\_discount' => $coupon->coupon\_discount,

'discount\_amount' => round(Cart::total() \* $coupon->coupon\_discount/100),

'total\_amount' => round(Cart::total() - Cart::total() \* $coupon->coupon\_discount/100 )

]);

}

return response()->json(['success' => 'Successfully Remove From Cart']);

}// End Method

public function CartDecrement($rowId){

$row = Cart::get($rowId);

Cart::update($rowId, $row->qty -1);

if(Session::has('coupon')){

$coupon\_name = Session::get('coupon')['coupon\_name'];

$coupon = Coupon::where('coupon\_name',$coupon\_name)->first();

Session::put('coupon',[

'coupon\_name' => $coupon->coupon\_name,

'coupon\_discount' => $coupon->coupon\_discount,

'discount\_amount' => round(Cart::total() \* $coupon->coupon\_discount/100),

'total\_amount' => round(Cart::total() - Cart::total() \* $coupon->coupon\_discount/100 )

]);

}

return response()->json('Decrement');

}// End Method

public function CartIncrement($rowId){

$row = Cart::get($rowId);

Cart::update($rowId, $row->qty +1);

if(Session::has('coupon')){

$coupon\_name = Session::get('coupon')['coupon\_name'];

$coupon = Coupon::where('coupon\_name',$coupon\_name)->first();

Session::put('coupon',[

'coupon\_name' => $coupon->coupon\_name,

'coupon\_discount' => $coupon->coupon\_discount,

'discount\_amount' => round(Cart::total() \* $coupon->coupon\_discount/100),

'total\_amount' => round(Cart::total() - Cart::total() \* $coupon->coupon\_discount/100 )

]);

}

return response()->json('Increment');

}// End Method

public function CouponApply(Request $request){

$coupon = Coupon::where('coupon\_name',$request->coupon\_name)->where('coupon\_validity','>=',Carbon::now()->format('Y-m-d'))->first();

if ($coupon) {

Session::put('coupon',[

'coupon\_name' => $coupon->coupon\_name,

'coupon\_discount' => $coupon->coupon\_discount,

'discount\_amount' => round(Cart::total() \* $coupon->coupon\_discount/100),

'total\_amount' => round(Cart::total() - Cart::total() \* $coupon->coupon\_discount/100 )

]);

return response()->json(array(

'validity' => true,

'success' => 'Coupon Applied Successfully'

));

} else{

return response()->json(['error' => 'Invalid Coupon']);

}

}// End Method

public function CouponCalculation(){

if (Session::has('coupon')) {

return response()->json(array(

'subtotal' => Cart::total(),

'coupon\_name' => session()->get('coupon')['coupon\_name'],

'coupon\_discount' => session()->get('coupon')['coupon\_discount'],

'discount\_amount' => session()->get('coupon')['discount\_amount'],

'total\_amount' => session()->get('coupon')['total\_amount'],

));

}else{

return response()->json(array(

'total' => Cart::total(),

));

}

}// End Method

public function CouponRemove(){

Session::forget('coupon');

return response()->json(['success' => 'Coupon Remove Successfully']);

}// End Method

public function CheckoutCreate(){

if (Auth::check()) {

if (Cart::total() > 0) {

$carts = Cart::content();

$cartQty = Cart::count();

$cartTotal = Cart::total();

$divisions = ShipDivision::orderBy('division\_name','ASC')->get();

return view('frontend.checkout.checkout\_view',compact('carts','cartQty','cartTotal','divisions'));

}else{

$notification = array(

'message' => 'Shopping At list One Product',

'alert-type' => 'error'

);

return redirect()->to('/')->with($notification);

}

}else{

$notification = array(

'message' => 'You Need to Login First',

'alert-type' => 'error'

);

return redirect()->route('login')->with($notification);

}

}// End Method

}

ក.១១ ទំព័របង់ប្រាក់ (Payment Page)

A screenshot of a computer

Description automatically generated

រូបភាព ក.៦៤ បង្ហាញពី Payment Page

* CheckOutController.blade.php

<?php

namespace App\Http\Controllers\User;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Product;

use App\Models\Coupon;

use App\Models\ShipDivision;

use App\Models\ShipDistricts;

use App\Models\ShipState;

use Gloudemans\Shoppingcart\Facades\Cart;

use Illuminate\Support\Facades\Session;

use Carbon\Carbon;

use Auth;

class CheckoutController extends Controller

{

public function DistrictGetAjax($division\_id){

$ship = ShipDistricts::where('division\_id',$division\_id)->orderBy('district\_name','ASC')->get();

return json\_encode($ship);

} // End Method

public function StateGetAjax($district\_id){

$ship = ShipState::where('district\_id',$district\_id)->orderBy('state\_name','ASC')->get();

return json\_encode($ship);

}// End Method

public function CheckoutStore(Request $request){

$data = array();

$data['shipping\_name'] = $request->shipping\_name;

$data['shipping\_email'] = $request->shipping\_email;

$data['shipping\_phone'] = $request->shipping\_phone;

$data['post\_code'] = $request->post\_code;

$data['division\_id'] = $request->division\_id;

$data['district\_id'] = $request->district\_id;

$data['state\_id'] = $request->state\_id;

$data['shipping\_address'] = $request->shipping\_address;

$data['notes'] = $request->notes;

$cartTotal = Cart::total();

if ($request->payment\_option == 'stripe') {

return view('frontend.payment.stripe',compact('data','cartTotal'));

}elseif ($request->payment\_option == 'card'){

return 'Card Page';

}else{

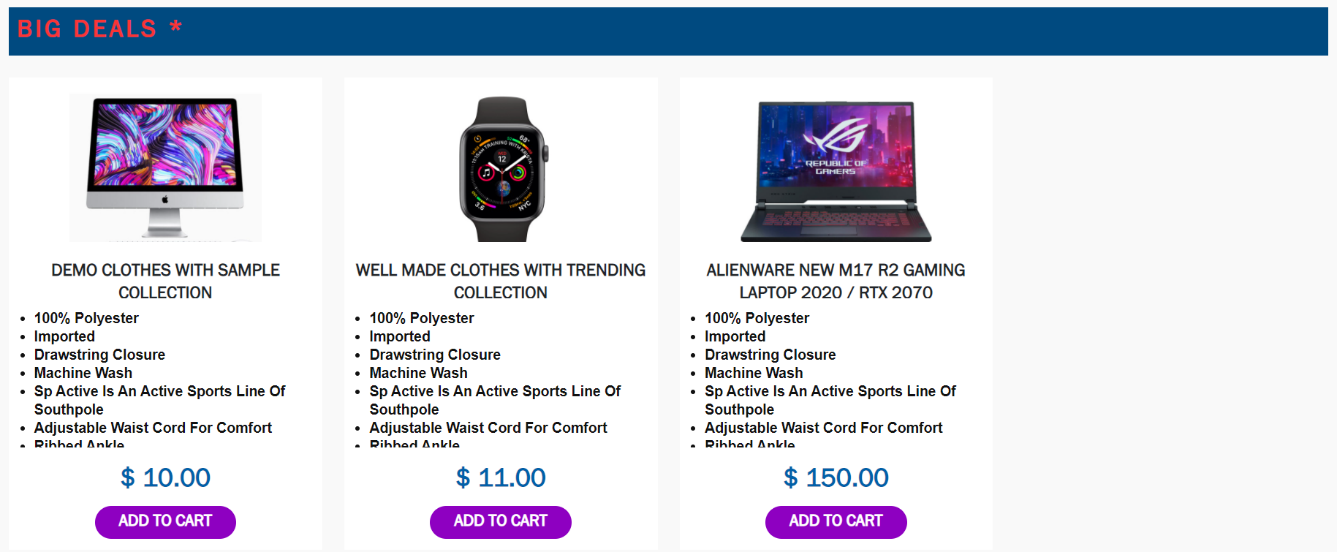
return view('frontend.payment.cash',compact('data','cartTotal'));

}

}// End Method

}

ក.១២ ទំព័ររីហ្វូរ (Review Product Page)



រូបភាព ក.៦៥ បង្ហាញពី Deal Page

* HomeController.blade.php

<?php

namespace App\Http\Controllers\Front;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

use App\Slide;

use App\Category;

use App\Product;

use App\Customer;

use App\Deals;

use Illuminate\Support\Facades\Validator;

use Illuminate\Foundation\Auth\RegistersUsers;

use Illuminate\Auth\Events\Registered;

use Twilio\Rest\Client;

class HomeController extends Controller

{

public function index(){

$slides = Slide::get();

$deals = Deals::whereDate('start\_date','<=',Date('Y-m-d'))

->whereDate('end\_date','>=',Date('Y-m-d'))

->get();

$dealLast = Deals::orderBy('end\_date','desc')

->first();

if(!empty(request()->except(['active']))){

if(request()->price != null && request()->model != null){

$categories = Category::with('product')->where("name",'like','%'.request()->model."%")->get();

}elseif(request()->price == null){

$categories = Category::with('product')->where("name",'like','%'.request()->model."%")->get();

}elseif(request()->model == null){

$categories = Category::with('product')->get();

}

}else{

if (!request()->has('search')){

$categories = Category::with('product')->whereHas('product',function($query){

$query->whereActive(1);

})->get();

}else{

$categories = Category::with('product')->whereHas('product',function($query){

$query->whereActive(1);

$query->where("slug",'like','%'.request()->search."%");

})->get();

}

}

return view('frontends.index',compact('slides','categories','deals','dealLast'));

}

// Hot Sale

public function hotSale(){

$deal = $deals = Deals::whereDate('start\_date','<=',Date('Y-m-d'))

->where('end\_date','>=',Date('Y-m-d'))

->get();

$dealLast = Deals::orderBy('end\_date','desc')->first();

return view('frontends.hot-sale',compact('deal','dealLast'));

}

public function catByName(){

$slides = Slide::get();

$categories = Category::with('product')->whereHas('product',function($query){

$query->whereActive(1);

})

->whereName(request()->cat\_name)

->get();

return view('frontends.category',compact('categories','slides'));

}

public function productDetail(){

$product = Product::where('id',request()->pro\_name)->first();

$product->views()->create();

return view('frontends.product-detail',compact('product'));

}

public function shopingCart(){

return view('frontends.shoping-cart');

}

public function categoryAll(){

return view('frontends.all-categories');

}

public function search(){ $product = Product::where("name",'like','%'.request()->search."%")->whereActive(1)->get();

return view('frontends.search',compact('product'));

}

protected function verify(Request $request)

{

$data = $request->validate([

'verification\_code' => ['required', 'numeric'],

'phone' => ['required', 'string'],

]);

/\* Get credentials from .env \*/

$token = getenv("TWILIO\_AUTH\_TOKEN");

$twilio\_sid = getenv("TWILIO\_SID");

$twilio\_verify\_sid = getenv("TWILIO\_VERIFY\_SID");

$twilio = new Client($twilio\_sid, $token);

$verification = $twilio->verify->v2->services($twilio\_verify\_sid)

->verificationChecks

->create($data['verification\_code'], array('to' => $data['phone']));

if ($verification->valid) {

$user = tap(Customer::where('phone', $data['phone']))->update(['email\_verified\_at' => date('Y-m-d H:i:s')]);

/\* Authenticate user \*/

return redirect()->route('customer.home'); }

return back()->with(['phone' => $data['phone'], 'error' => 'Invalid verification code entered!']);

}

}ក.១៣ ទំព័រព័ត៍មានលម្អិតរបស់ផលិតផល (Product Detail Page)

A computer and tablet with a screen on it

Description automatically generated with medium confidence

រូបភាព ក.៦៦ បង្ហាញពី Product Detail Page

* ProductController.php

<?php

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Category;

use App\Models\SubCategory;

use App\Models\MultiImg;

use App\Models\Brand;

use App\Models\Product;

use App\Models\User;

use Image;

use Carbon\Carbon;

class ProductController extends Controller

{

public function AllProduct(){

$products = Product::latest()->get();

return view('backend.product.product\_all',compact('products'));

} // End Method

public function AddProduct(){

$activeVendor = User::where('status','active')->where('role','vendor')->latest()->get();

$brands = Brand::latest()->get();

$categories = Category::latest()->get();

return view('backend.product.product\_add',compact('brands','categories','activeVendor'));

} // End Method

public function StoreProduct(Request $request){

$image = $request->file('product\_thambnail');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(800,800)->save('upload/products/thambnail/'.$name\_gen);

$save\_url = 'upload/products/thambnail/'.$name\_gen;

$product\_id = Product::insertGetId([

'brand\_id' => $request->brand\_id,

'category\_id' => $request->category\_id,

'subcategory\_id' => $request->subcategory\_id,

'product\_name' => $request->product\_name,

'product\_slug' => strtolower(str\_replace(' ','-',$request->product\_name)),

'product\_code' => $request->product\_code,

'product\_qty' => $request->product\_qty,

'product\_tags' => $request->product\_tags,

'product\_size' => $request->product\_size,

'product\_color' => $request->product\_color,

'selling\_price' => $request->selling\_price,

'discount\_price' => $request->discount\_price,

'short\_descp' => $request->short\_descp,

'long\_descp' => $request->long\_descp,

'hot\_deals' => $request->hot\_deals,

'featured' => $request->featured,

'special\_offer' => $request->special\_offer,

'special\_deals' => $request->special\_deals,

'product\_thambnail' => $save\_url,

'vendor\_id' => $request->vendor\_id,

'status' => 1,

'created\_at' => Carbon::now(),

]);

/// Multiple Image Upload From her //////

$images = $request->file('multi\_img');

foreach($images as $img){

$make\_name = hexdec(uniqid()).'.'.$img->getClientOriginalExtension();

Image::make($img)->resize(800,800)->save('upload/products/multi-image/'.$make\_name);

$uploadPath = 'upload/products/multi-image/'.$make\_name;

MultiImg::insert([

'product\_id' => $product\_id,

'photo\_name' => $uploadPath,

'created\_at' => Carbon::now(),

]);

} // end foreach

/// End Multiple Image Upload From her //////

$notification = array(

'message' => 'Product Inserted Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.product')->with($notification);

} // End Method

public function EditProduct($id){

$multiImgs = MultiImg::where('product\_id',$id)->get();

$activeVendor = User::where('status','active')->where('role','vendor')->latest()->get();

$brands = Brand::latest()->get();

$categories = Category::latest()->get();

$subcategory = SubCategory::latest()->get();

$products = Product::findOrFail($id);

return view('backend.product.product\_edit',compact('brands','categories','activeVendor','products','subcategory','multiImgs'));

}// End Method

public function UpdateProduct(Request $request){

$product\_id = $request->id;

Product::findOrFail($product\_id)->update([

'brand\_id' => $request->brand\_id,

'category\_id' => $request->category\_id,

'subcategory\_id' => $request->subcategory\_id,

'product\_name' => $request->product\_name,

'product\_slug' => strtolower(str\_replace(' ','-',$request->product\_name)),

'product\_code' => $request->product\_code,

'product\_qty' => $request->product\_qty,

'product\_tags' => $request->product\_tags,

'product\_size' => $request->product\_size,

'product\_color' => $request->product\_color,

'selling\_price' => $request->selling\_price,

'discount\_price' => $request->discount\_price,

'short\_descp' => $request->short\_descp,

'long\_descp' => $request->long\_descp,

'hot\_deals' => $request->hot\_deals,

'featured' => $request->featured,

'special\_offer' => $request->special\_offer,

'special\_deals' => $request->special\_deals,

'vendor\_id' => $request->vendor\_id,

'status' => 1,

'created\_at' => Carbon::now(),

]);

$notification = array(

'message' => 'Product Updated Without Image Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.product')->with($notification);

}// End Method

public function UpdateProductThambnail(Request $request){

$pro\_id = $request->id;

$oldImage = $request->old\_img;

$image = $request->file('product\_thambnail');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(800,800)->save('upload/products/thambnail/'.$name\_gen);

$save\_url = 'upload/products/thambnail/'.$name\_gen;

if (file\_exists($oldImage)) {

unlink($oldImage);

}

Product::findOrFail($pro\_id)->update([

'product\_thambnail' => $save\_url,

'updated\_at' => Carbon::now(),

]);

$notification = array(

'message' => 'Product Image Thambnail Updated Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

// Multi Image Update

public function UpdateProductMultiimage(Request $request){

$imgs = $request->multi\_img;

foreach($imgs as $id => $img ){

$imgDel = MultiImg::findOrFail($id);

unlink($imgDel->photo\_name);

$make\_name = hexdec(uniqid()).'.'.$img->getClientOriginalExtension();

Image::make($img)->resize(800,800)->save('upload/products/multi-image/'.$make\_name);

$uploadPath = 'upload/products/multi-image/'.$make\_name;

MultiImg::where('id',$id)->update([

'photo\_name' => $uploadPath,

'updated\_at' => Carbon::now(),

]);

} // end foreach

$notification = array(

'message' => 'Product Multi Image Updated Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function MulitImageDelelte($id){

$oldImg = MultiImg::findOrFail($id);

unlink($oldImg->photo\_name);

MultiImg::findOrFail($id)->delete();

$notification = array(

'message' => 'Product Multi Image Deleted Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function ProductInactive($id){

Product::findOrFail($id)->update(['status' => 0]);

$notification = array(

'message' => 'Product Inactive',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function ProductActive($id){

Product::findOrFail($id)->update(['status' => 1]);

$notification = array(

'message' => 'Product Active',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function ProductDelete($id){

$product = Product::findOrFail($id);

unlink($product->product\_thambnail);

Product::findOrFail($id)->delete();

$imges = MultiImg::where('product\_id',$id)->get();

foreach($imges as $img){

unlink($img->photo\_name);

MultiImg::where('product\_id',$id)->delete();

}

$notification = array(

'message' => 'Product Deleted Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function ProductStock(){

$products = Product::latest()->get();

return view('backend.product.product\_stock',compact('products'));

}// End Method

}

ក.១៤ ទំព័រស្វែករកផលិផល (Search Result Page)

A screenshot of a computer

Description automatically generated

រូបភាព ក.៦៧ បង្ហាញពី Search Result Page

* IndexController.blade.php

<?php

namespace App\Http\Controllers\Frontend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Category;

use App\Models\SubCategory;

use App\Models\MultiImg;

use App\Models\Brand;

use App\Models\Product;

use App\Models\User;

class IndexController extends Controller

{

public function Index(){

$skip\_category\_0 = Category::skip(0)->first();

$skip\_product\_0 = Product::where('status',1)->where('category\_id',$skip\_category\_0->id)->orderBy('id','DESC')->limit(5)->get();

$skip\_category\_1 = Category::skip(1)->first();

$skip\_product\_1 = Product::where('status',1)->where('category\_id',$skip\_category\_1->id)->orderBy('id','DESC')->limit(5)->get();

$skip\_category\_2 = Category::skip(2)->first();

$skip\_product\_2 = Product::where('status',1)->where('category\_id',$skip\_category\_2->id)->orderBy('id','DESC')->limit(5)->get();

$skip\_category\_3 = Category::skip(3)->first();

$skip\_product\_3 = Product::where('status',1)->where('category\_id',$skip\_category\_3->id)->orderBy('id','DESC')->limit(5)->get();

$hot\_deals = Product::where('hot\_deals',1)->where('discount\_price','!=',NULL)->orderBy('id','DESC')->limit(3)->get();

$special\_offer = Product::where('special\_offer',1)->orderBy('id','DESC')->limit(3)->get();

$new = Product::where('status',1)->orderBy('id','DESC')->limit(3)->get();

$special\_deals = Product::where('special\_deals',1)->orderBy('id','DESC')->limit(3)->get();

return view('frontend.index',compact('skip\_category\_0','skip\_product\_0','skip\_category\_1','skip\_product\_1','skip\_category\_2','skip\_product\_2','skip\_category\_3','skip\_product\_3','hot\_deals','special\_offer','new','special\_deals'));

}

public function ProductDetails($id,$slug){

$product = Product::findOrFail($id);

$color = $product->product\_color;

$product\_color = explode(',', $color);

$size = $product->product\_size;

$product\_size = explode(',', $size);

$multiImage = MultiImg::where('product\_id',$id)->get();

$cat\_id = $product->category\_id;

$relatedProduct = Product::where('category\_id',$cat\_id)->where('id','!=',$id)->orderBy('id','DESC')->limit(4)->get();

return view('frontend.product.product\_details',compact('product','product\_color','product\_size','multiImage','relatedProduct'));

}

public function VendorDetails($id){

$vendor = User::findOrFail($id);

$vproduct = Product::where('vendor\_id',$id)->get();

return view('frontend.vendor.vendor\_details',compact('vendor','vproduct'));

}

public function VendorAll(){

$vendors = User::where('status','active')->where('role','vendor')->orderBy('id','DESC')->get();

return view('frontend.vendor.vendor\_all',compact('vendors'));

}

public function CatWiseProduct(Request $request,$id,$slug){

$products = Product::where('status',1)->where('category\_id',$id)->orderBy('id','DESC')->get();

$categories = Category::orderBy('category\_name','ASC')->get();

$breadcat = Category::where('id',$id)->first();

$newProduct = Product::orderBy('id','DESC')->limit(3)->get();

return view('frontend.product.category\_view',compact('products','categories','breadcat','newProduct'));

}

public function SubCatWiseProduct(Request $request,$id,$slug){

$products = Product::where('status',1)->where('subcategory\_id',$id)->orderBy('id','DESC')->get();

$categories = Category::orderBy('category\_name','ASC')->get();

$breadsubcat = SubCategory::where('id',$id)->first();

$newProduct = Product::orderBy('id','DESC')->limit(3)->get();

return view('frontend.product.subcategory\_view',compact('products','categories','breadsubcat','newProduct'));

}

public function ProductViewAjax($id){

$product = Product::with('category','brand')->findOrFail($id);

$color = $product->product\_color;

$product\_color = explode(',', $color);

$size = $product->product\_size;

$product\_size = explode(',', $size);

return response()->json(array(

'product' => $product,

'color' => $product\_color,

'size' => $product\_size,

));

}

public function ProductSearch(Request $request){

$request->validate(['search' => "required"]);

$item = $request->search;

$categories = Category::orderBy('category\_name','ASC')->get();

$products = Product::where('product\_name','LIKE',"%$item%")->get();

$newProduct = Product::orderBy('id','DESC')->limit(3)->get();

return view('frontend.product.search',compact('products','item','categories','newProduct'));

}

public function SearchProduct(Request $request){

$request->validate(['search' => "required"]);

$item = $request->search;

$products = Product::where('product\_name','LIKE',"%$item%")->select('product\_name','product\_slug','product\_thambnail','selling\_price','id')->limit(6)->get();

return view('frontend.product.search\_product',compact('products'));

}

}

A screenshot of a computer

Description automatically generatedក.១៥ ទំព័រដាស់បត (Dashboard Page)

រូបភាព ក.៦៨ បង្ហាញពី Dashboard Page

* AdminController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use App\Models\User;

use Illuminate\Support\Facades\Hash;

use Spatie\Permission\Models\Role;

use Spatie\Permission\Models\Permission;

use App\Notifications\VendorApproveNotification;

use Illuminate\Support\Facades\Notification;

class AdminController extends Controller

{

public function AdminDashboard(){

return view('admin.index');

} // End Mehtod

public function AdminLogin(){

return view('admin.admin\_login');

} // End Mehtod

public function AdminDestroy(Request $request){

Auth::guard('web')->logout();

$request->session()->invalidate();

$request->session()->regenerateToken();

return redirect('/admin/login');

} // End Mehtod

public function AdminProfile(){

$id = Auth::user()->id;

$adminData = User::find($id);

return view('admin.admin\_profile\_view',compact('adminData'));

} // End Mehtod

public function AdminProfileStore(Request $request){

$id = Auth::user()->id;

$data = User::find($id);

$data->name = $request->name;

$data->email = $request->email;

$data->phone = $request->phone;

$data->address = $request->address;

if ($request->file('photo')) {

$file = $request->file('photo');

@unlink(public\_path('upload/admin\_images/'.$data->photo));

$filename = date('YmdHi').$file->getClientOriginalName();

$file->move(public\_path('upload/admin\_images'),$filename);

$data['photo'] = $filename;

}

$data->save();

$notification = array(

'message' => 'Admin Profile Updated Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

} // End Mehtod

public function AdminChangePassword(){

return view('admin.admin\_change\_password');

} // End Mehtod

public function AdminUpdatePassword(Request $request){

// Validation

$request->validate([

'old\_password' => 'required',

'new\_password' => 'required|confirmed',

]);

// Match The Old Password

if (!Hash::check($request->old\_password, auth::user()->password)) {

return back()->with("error", "Old Password Doesn't Match!!");

}

// Update The new password

User::whereId(auth()->user()->id)->update([

'password' => Hash::make($request->new\_password)

]);

return back()->with("status", " Password Changed Successfully");

} // End Mehtod

public function InactiveVendor(){

$inActiveVendor = User::where('status','inactive')->where('role','vendor')->latest()->get();

return view('backend.vendor.inactive\_vendor',compact('inActiveVendor'));

}// End Mehtod

public function ActiveVendor(){

$ActiveVendor = User::where('status','active')->where('role','vendor')->latest()->get();

return view('backend.vendor.active\_vendor',compact('ActiveVendor'));

}// End Mehtod

public function InactiveVendorDetails($id){

$inactiveVendorDetails = User::findOrFail($id);

return view('backend.vendor.inactive\_vendor\_details',compact('inactiveVendorDetails'));

}// End Mehtod

public function ActiveVendorApprove(Request $request){

$verdor\_id = $request->id;

$user = User::findOrFail($verdor\_id)->update([

'status' => 'active',

]);

$notification = array(

'message' => 'Vendor Active Successfully',

'alert-type' => 'success'

);

$vuser = User::where('role','vendor')->get();

Notification::send($vuser, new VendorApproveNotification($request));

return redirect()->route('active.vendor')->with($notification);

}// End Mehtod

public function ActiveVendorDetails($id){

$activeVendorDetails = User::findOrFail($id);

return view('backend.vendor.active\_vendor\_details',compact('activeVendorDetails'));

}// End Mehtod

public function InActiveVendorApprove(Request $request){

$verdor\_id = $request->id;

$user = User::findOrFail($verdor\_id)->update([

'status' => 'inactive',

]);

$notification = array(

'message' => 'Vendor InActive Successfully',

'alert-type' => 'success'

);

return redirect()->route('inactive.vendor')->with($notification);

}// End Mehtod

///////////// Admin All Method //////////////

public function AllAdmin(){

$alladminuser = User::where('role','admin')->latest()->get();

return view('backend.admin.all\_admin',compact('alladminuser'));

}// End Mehtod

public function AddAdmin(){

$roles = Role::all();

return view('backend.admin.add\_admin',compact('roles'));

}// End Mehtod

public function AdminUserStore(Request $request){

$user = new User();

$user->username = $request->username;

$user->name = $request->name;

$user->email = $request->email;

$user->phone = $request->phone;

$user->address = $request->address;

$user->password = Hash::make($request->password);

$user->role = 'admin';

$user->status = 'active';

$user->save();

if ($request->roles) {

$user->assignRole($request->roles);

}

$notification = array(

'message' => 'New Admin User Inserted Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.admin')->with($notification);

}// End Mehtod

public function EditAdminRole($id){

$user = User::findOrFail($id);

$roles = Role::all();

return view('backend.admin.edit\_admin',compact('user','roles'));

}// End Mehtod

public function AdminUserUpdate(Request $request,$id){

$user = User::findOrFail($id);

$user->username = $request->username;

$user->name = $request->name;

$user->email = $request->email;

$user->phone = $request->phone;

$user->address = $request->address;

$user->role = 'admin';

$user->status = 'active';

$user->save();

$user->roles()->detach();

if ($request->roles) {

$user->assignRole($request->roles);

}

$notification = array(

'message' => 'New Admin User Updated Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.admin')->with($notification);

}// End Mehtod

public function DeleteAdminRole($id){

$user = User::findOrFail($id);

if (!is\_null($user)) {

$user->delete();

}

$notification = array(

'message' => 'Admin User Deleted Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Mehtod

}

ក.១៦ ទំព័របញ្ជីប្រភេទប្រេន (Brand List Page)

A screenshot of a computer

Description automatically generated

រូបភាព ក.៦៩ បង្ហាញពី Brand List Page

* BrandController.php

<?php

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Brand;

use Image;

class BrandController extends Controller

{

public function AllBrand(){

$brands = Brand::latest()->get();

return view('backend.brand.brand\_all',compact('brands'));

} // End Method

public function AddBrand(){

return view('backend.brand.brand\_add');

} // End Method

public function StoreBrand(Request $request){

$image = $request->file('brand\_image');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(300,300)->save('upload/brand/'.$name\_gen);

$save\_url = 'upload/brand/'.$name\_gen;

Brand::insert([

'brand\_name' => $request->brand\_name,

'brand\_slug' => strtolower(str\_replace(' ', '-',$request->brand\_name)),

'brand\_image' => $save\_url,

]);

$notification = array(

'message' => 'Brand Inserted Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.brand')->with($notification);

}// End Method

public function EditBrand($id){

$brand = Brand::findOrFail($id);

return view('backend.brand.brand\_edit',compact('brand'));

}// End Method

public function UpdateBrand(Request $request){

$brand\_id = $request->id;

$old\_img = $request->old\_image;

if ($request->file('brand\_image')) {

$image = $request->file('brand\_image');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(300,300)->save('upload/brand/'.$name\_gen);

$save\_url = 'upload/brand/'.$name\_gen;

if (file\_exists($old\_img)) {

unlink($old\_img);

}

Brand::findOrFail($brand\_id)->update([

'brand\_name' => $request->brand\_name,

'brand\_slug' => strtolower(str\_replace(' ', '-',$request->brand\_name)),

'brand\_image' => $save\_url,

]);

$notification = array(

'message' => 'Brand Updated with image Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.brand')->with($notification);

} else {

Brand::findOrFail($brand\_id)->update([

'brand\_name' => $request->brand\_name,

'brand\_slug' => strtolower(str\_replace(' ', '-',$request->brand\_name)),

]);

$notification = array(

'message' => 'Brand Updated without image Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.brand')->with($notification);

} // end else

}// End Method

public function DeleteBrand($id){

$brand = Brand::findOrFail($id);

$img = $brand->brand\_image;

unlink($img );

Brand::findOrFail($id)->delete();

$notification = array(

'message' => 'Brand Deleted Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

}

A screenshot of a computer

Description automatically generatedក.១៦ ទំព័របញ្ជីប្រភេទផលិតផល (Category List Page)

រូបភាព ក.៧០ បង្ហាញពី Category List Page

* CategoryController.php

<?php

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Category;

use Image;

class CategoryController extends Controller

{

public function AllCategory(){

$categories = Category::latest()->get();

return view('backend.category.category\_all',compact('categories'));

} // End Method

public function AddCategory(){

return view('backend.category.category\_add');

}// End Method

public function StoreCategory(Request $request){

$image = $request->file('category\_image');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(120,120)->save('upload/category/'.$name\_gen);

$save\_url = 'upload/category/'.$name\_gen;

Category::insert([

'category\_name' => $request->category\_name,

'category\_slug' => strtolower(str\_replace(' ', '-',$request->category\_name)),

'category\_image' => $save\_url,

]);

$notification = array(

'message' => 'Category Inserted Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.category')->with($notification);

}// End Method

public function EditCategory($id){

$category = Category::findOrFail($id);

return view('backend.category.category\_edit',compact('category'));

}// End Method

public function UpdateCategory(Request $request){

$cat\_id = $request->id;

$old\_img = $request->old\_image;

if ($request->file('category\_image')) {

$image = $request->file('category\_image');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(120,120)->save('upload/category/'.$name\_gen);

$save\_url = 'upload/category/'.$name\_gen;

if (file\_exists($old\_img)) {

unlink($old\_img);

}

Category::findOrFail($cat\_id)->update([

'category\_name' => $request->category\_name,

'category\_slug' => strtolower(str\_replace(' ', '-',$request->category\_name)),

'category\_image' => $save\_url,

]);

$notification = array(

'message' => 'Category Updated with image Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.category')->with($notification);

} else {

Category::findOrFail($cat\_id)->update([

'category\_name' => $request->category\_name,

'category\_slug' => strtolower(str\_replace(' ', '-',$request->category\_name)),

]);

$notification = array(

'message' => 'Category Updated without image Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.category')->with($notification);

} // end else

}// End Method

public function DeleteCategory($id){

$category = Category::findOrFail($id);

$img = $category->category\_image;

if(unlink($img))

{

echo "File Deleted";

}

else

{

echo "Error Deleting File";

}

Category::findOrFail($id)->delete();

$notification = array(

'message' => 'Category Deleted Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

}

A screenshot of a computer

Description automatically generatedក.១៧ ទំព័របង្កើតប្រភេទផលិតផល (Create Category Page)

រូបភាព ក.៧១ បង្ហាញពី Create Category Page

* CategoryController.php

<?php

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Category;

use Image;

class CategoryController extends Controller

{

public function AllCategory(){

$categories = Category::latest()->get();

return view('backend.category.category\_all',compact('categories'));

} // End Method

public function AddCategory(){

return view('backend.category.category\_add');

}// End Method

public function StoreCategory(Request $request){

$image = $request->file('category\_image');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(120,120)->save('upload/category/'.$name\_gen);

$save\_url = 'upload/category/'.$name\_gen;

Category::insert([

'category\_name' => $request->category\_name,

'category\_slug' => strtolower(str\_replace(' ', '-',$request->category\_name)),

'category\_image' => $save\_url,

]);

$notification = array(

'message' => 'Category Inserted Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.category')->with($notification);

}// End Method

public function EditCategory($id){

$category = Category::findOrFail($id);

return view('backend.category.category\_edit',compact('category'));

}// End Method

public function UpdateCategory(Request $request){

$cat\_id = $request->id;

$old\_img = $request->old\_image;

if ($request->file('category\_image')) {

$image = $request->file('category\_image');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(120,120)->save('upload/category/'.$name\_gen);

$save\_url = 'upload/category/'.$name\_gen;

if (file\_exists($old\_img)) {

unlink($old\_img);

}

Category::findOrFail($cat\_id)->update([

'category\_name' => $request->category\_name,

'category\_slug' => strtolower(str\_replace(' ', '-',$request->category\_name)),

'category\_image' => $save\_url,

]);

$notification = array(

'message' => 'Category Updated with image Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.category')->with($notification);

} else {

Category::findOrFail($cat\_id)->update([

'category\_name' => $request->category\_name,

'category\_slug' => strtolower(str\_replace(' ', '-',$request->category\_name)),

]);

$notification = array(

'message' => 'Category Updated without image Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.category')->with($notification);

} // end else

}// End Method

public function DeleteCategory($id){

$category = Category::findOrFail($id);

$img = $category->category\_image;

if(unlink($img))

{

echo "File Deleted";

}

else

{

echo "Error Deleting File";

}

Category::findOrFail($id)->delete();

$notification = array(

'message' => 'Category Deleted Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

}

A screenshot of a computer

Description automatically generatedក.១៩ ទំព័រប្រភេទផលិតផលរង (Sub Category Page)

រូបភាព ក.៧២ បង្ហាញពី Sub Category Page

* Admin/SlideController.php

<?php

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Category;

use App\Models\SubCategory;

class SubCategoryController extends Controller

{

public function AllSubCategory(){

$subcategories = SubCategory::latest()->get();

return view('backend.subcategory.subcategory\_all',compact('subcategories'));

} // End Method

public function AddSubCategory(){

$categories = Category::orderBy('category\_name','ASC')->get();

return view('backend.subcategory.subcategory\_add',compact('categories'));

}// End Method

public function StoreSubCategory(Request $request){

SubCategory::insert([

'category\_id' => $request->category\_id,

'subcategory\_name' => $request->subcategory\_name,

'subcategory\_slug' => strtolower(str\_replace(' ', '-',$request->subcategory\_name)),

]);

$notification = array(

'message' => 'SubCategory Inserted Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.subcategory')->with($notification);

}// End Method

public function EditSubCategory($id){

$categories = Category::orderBy('category\_name','ASC')->get();

$subcategory = SubCategory::findOrFail($id);

return view('backend.subcategory.subcategory\_edit',compact('categories','subcategory'));

}// End Method

public function UpdateSubCategory(Request $request){

$subcat\_id = $request->id;

SubCategory::findOrFail($subcat\_id)->update([

'category\_id' => $request->category\_id,

'subcategory\_name' => $request->subcategory\_name,

'subcategory\_slug' => strtolower(str\_replace(' ', '-',$request->subcategory\_name)),

]);

$notification = array(

'message' => 'SubCategory Updated Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.subcategory')->with($notification);

}// End Method

public function DeleteSubCategory($id){

SubCategory::findOrFail($id)->delete();

$notification = array(

'message' => 'SubCategory Deleted Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function GetSubCategory($category\_id){

$subcat = SubCategory::where('category\_id',$category\_id)->orderBy('subcategory\_name','ASC')->get();

return json\_encode($subcat);

}// End Method

}

A screenshot of a computer

Description automatically generatedក.២០ ទំព័របង្កើតស្លាយដឺ (Create Slider Page)

រូបភាព ក.៧៣ បង្ហាញពី Create Slider Page

* SliderController.php

<?php

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Slider;

use Image;

class SliderController extends Controller

{

public function AllSlider(){

$sliders = Slider::latest()->get();

return view('backend.slider.slider\_all',compact('sliders'));

} // End Method

public function AddSlider(){

return view('backend.slider.slider\_add');

}// End Method

public function StoreSlider(Request $request){

$image = $request->file('slider\_image');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(2376,807)->save('upload/slider/'.$name\_gen);

$save\_url = 'upload/slider/'.$name\_gen;

Slider::insert([

'slider\_title' => $request->slider\_title,

'short\_title' => $request->short\_title,

'slider\_image' => $save\_url,

]);

$notification = array(

'message' => 'Slider Inserted Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.slider')->with($notification);

}// End Method

public function EditSlider($id){

$sliders = Slider::findOrFail($id);

return view('backend.slider.slider\_edit',compact('sliders'));

}// End Method

public function UpdateSlider(Request $request){

$slider\_id = $request->id;

$old\_img = $request->old\_image;

if ($request->file('slider\_image')) {

$image = $request->file('slider\_image');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(2376,807)->save('upload/slider/'.$name\_gen);

$save\_url = 'upload/slider/'.$name\_gen;

if (file\_exists($old\_img)) {

unlink($old\_img);

}

Slider::findOrFail($slider\_id)->update([

'slider\_title' => $request->slider\_title,

'short\_title' => $request->short\_title,

'slider\_image' => $save\_url,

]);

$notification = array(

'message' => 'Slider Updated with image Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.slider')->with($notification);

} else {

Slider::findOrFail($slider\_id)->update([

'slider\_title' => $request->slider\_title,

'short\_title' => $request->short\_title,

]);

$notification = array(

'message' => 'Slider Updated without image Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.slider')->with($notification);

} // end else

}// End Method

public function DeleteSlider($id){

$slider = Slider::findOrFail($id);

$img = $slider->slider\_image;

unlink($img );

Slider::findOrFail($id)->delete();

$notification = array(

'message' => 'Slider Deleted Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

}

A screenshot of a computer

Description automatically generatedក.២១ ទំព័របញ្ជីផលិតផល (Product List Page)

រូបភាព ក.៧៤ បង្ហាញពី Product List Page

* ProductController.php

<?php

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Category;

use App\Models\SubCategory;

use App\Models\MultiImg;

use App\Models\Brand;

use App\Models\Product;

use App\Models\User;

use Image;

use Carbon\Carbon;

class ProductController extends Controller

{

public function AllProduct(){

$products = Product::latest()->get();

return view('backend.product.product\_all',compact('products'));

} // End Method

public function AddProduct(){

$activeVendor = User::where('status','active')->where('role','vendor')->latest()->get();

$brands = Brand::latest()->get();

$categories = Category::latest()->get();

return view('backend.product.product\_add',compact('brands','categories','activeVendor'));

} // End Method

public function StoreProduct(Request $request){

$image = $request->file('product\_thambnail');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(800,800)->save('upload/products/thambnail/'.$name\_gen);

$save\_url = 'upload/products/thambnail/'.$name\_gen;

$product\_id = Product::insertGetId([

'brand\_id' => $request->brand\_id,

'category\_id' => $request->category\_id,

'subcategory\_id' => $request->subcategory\_id,

'product\_name' => $request->product\_name,

'product\_slug' => strtolower(str\_replace(' ','-',$request->product\_name)),

'product\_code' => $request->product\_code,

'product\_qty' => $request->product\_qty,

'product\_tags' => $request->product\_tags,

'product\_size' => $request->product\_size,

'product\_color' => $request->product\_color,

'selling\_price' => $request->selling\_price,

'discount\_price' => $request->discount\_price,

'short\_descp' => $request->short\_descp,

'long\_descp' => $request->long\_descp,

'hot\_deals' => $request->hot\_deals,

'featured' => $request->featured,

'special\_offer' => $request->special\_offer,

'special\_deals' => $request->special\_deals,

'product\_thambnail' => $save\_url,

'vendor\_id' => $request->vendor\_id,

'status' => 1,

'created\_at' => Carbon::now(),

]);

/// Multiple Image Upload From her //////

$images = $request->file('multi\_img');

foreach($images as $img){

$make\_name = hexdec(uniqid()).'.'.$img->getClientOriginalExtension();

Image::make($img)->resize(800,800)->save('upload/products/multi-image/'.$make\_name);

$uploadPath = 'upload/products/multi-image/'.$make\_name;

MultiImg::insert([

'product\_id' => $product\_id,

'photo\_name' => $uploadPath,

'created\_at' => Carbon::now(),

]);

} // end foreach

/// End Multiple Image Upload From her //////

$notification = array(

'message' => 'Product Inserted Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.product')->with($notification);

} // End Method

public function EditProduct($id){

$multiImgs = MultiImg::where('product\_id',$id)->get();

$activeVendor = User::where('status','active')->where('role','vendor')->latest()->get();

$brands = Brand::latest()->get();

$categories = Category::latest()->get();

$subcategory = SubCategory::latest()->get();

$products = Product::findOrFail($id);

return view('backend.product.product\_edit',compact('brands','categories','activeVendor','products','subcategory','multiImgs'));

}// End Method

public function UpdateProduct(Request $request){

$product\_id = $request->id;

Product::findOrFail($product\_id)->update([

'brand\_id' => $request->brand\_id,

'category\_id' => $request->category\_id,

'subcategory\_id' => $request->subcategory\_id,

'product\_name' => $request->product\_name,

'product\_slug' => strtolower(str\_replace(' ','-',$request->product\_name)),

'product\_code' => $request->product\_code,

'product\_qty' => $request->product\_qty,

'product\_tags' => $request->product\_tags,

'product\_size' => $request->product\_size,

'product\_color' => $request->product\_color,

'selling\_price' => $request->selling\_price,

'discount\_price' => $request->discount\_price,

'short\_descp' => $request->short\_descp,

'long\_descp' => $request->long\_descp,

'hot\_deals' => $request->hot\_deals,

'featured' => $request->featured,

'special\_offer' => $request->special\_offer,

'special\_deals' => $request->special\_deals,

'vendor\_id' => $request->vendor\_id,

'status' => 1,

'created\_at' => Carbon::now(),

]);

$notification = array(

'message' => 'Product Updated Without Image Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.product')->with($notification);

}// End Method

public function UpdateProductThambnail(Request $request){

$pro\_id = $request->id;

$oldImage = $request->old\_img;

$image = $request->file('product\_thambnail');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(800,800)->save('upload/products/thambnail/'.$name\_gen);

$save\_url = 'upload/products/thambnail/'.$name\_gen;

if (file\_exists($oldImage)) {

unlink($oldImage);

}

Product::findOrFail($pro\_id)->update([

'product\_thambnail' => $save\_url,

'updated\_at' => Carbon::now(),

]);

$notification = array(

'message' => 'Product Image Thambnail Updated Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

// Multi Image Update

public function UpdateProductMultiimage(Request $request){

$imgs = $request->multi\_img;

foreach($imgs as $id => $img ){

$imgDel = MultiImg::findOrFail($id);

unlink($imgDel->photo\_name);

$make\_name = hexdec(uniqid()).'.'.$img->getClientOriginalExtension();

Image::make($img)->resize(800,800)->save('upload/products/multi-image/'.$make\_name);

$uploadPath = 'upload/products/multi-image/'.$make\_name;

MultiImg::where('id',$id)->update([

'photo\_name' => $uploadPath,

'updated\_at' => Carbon::now(),

]);

} // end foreach

$notification = array(

'message' => 'Product Multi Image Updated Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function MulitImageDelelte($id){

$oldImg = MultiImg::findOrFail($id);

unlink($oldImg->photo\_name);

MultiImg::findOrFail($id)->delete();

$notification = array(

'message' => 'Product Multi Image Deleted Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function ProductInactive($id){

Product::findOrFail($id)->update(['status' => 0]);

$notification = array(

'message' => 'Product Inactive',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function ProductActive($id){

Product::findOrFail($id)->update(['status' => 1]);

$notification = array(

'message' => 'Product Active',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function ProductDelete($id){

$product = Product::findOrFail($id);

unlink($product->product\_thambnail);

Product::findOrFail($id)->delete();

$imges = MultiImg::where('product\_id',$id)->get();

foreach($imges as $img){

unlink($img->photo\_name);

MultiImg::where('product\_id',$id)->delete();

}

$notification = array(

'message' => 'Product Deleted Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function ProductStock(){

$products = Product::latest()->get();

return view('backend.product.product\_stock',compact('products'));

}// End Method

}

A screenshot of a computer

Description automatically generatedក.២៣ ទំព័របង្កើតផលិតផល (Create Product Page)

រូបភាព ក.៧៥ បង្ហាញពី Create Product Page

* Admin/ProductController.php

<?php

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Category;

use App\Models\SubCategory;

use App\Models\MultiImg;

use App\Models\Brand;

use App\Models\Product;

use App\Models\User;

use Image;

use Carbon\Carbon;

class ProductController extends Controller

{

public function AllProduct(){

$products = Product::latest()->get();

return view('backend.product.product\_all',compact('products'));

} // End Method

public function AddProduct(){

$activeVendor = User::where('status','active')->where('role','vendor')->latest()->get();

$brands = Brand::latest()->get();

$categories = Category::latest()->get();

return view('backend.product.product\_add',compact('brands','categories','activeVendor'));

} // End Method

public function StoreProduct(Request $request){

$image = $request->file('product\_thambnail');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(800,800)->save('upload/products/thambnail/'.$name\_gen);

$save\_url = 'upload/products/thambnail/'.$name\_gen;

$product\_id = Product::insertGetId([

'brand\_id' => $request->brand\_id,

'category\_id' => $request->category\_id,

'subcategory\_id' => $request->subcategory\_id,

'product\_name' => $request->product\_name,

'product\_slug' => strtolower(str\_replace(' ','-',$request->product\_name)),

'product\_code' => $request->product\_code,

'product\_qty' => $request->product\_qty,

'product\_tags' => $request->product\_tags,

'product\_size' => $request->product\_size,

'product\_color' => $request->product\_color,

'selling\_price' => $request->selling\_price,

'discount\_price' => $request->discount\_price,

'short\_descp' => $request->short\_descp,

'long\_descp' => $request->long\_descp,

'hot\_deals' => $request->hot\_deals,

'featured' => $request->featured,

'special\_offer' => $request->special\_offer,

'special\_deals' => $request->special\_deals,

'product\_thambnail' => $save\_url,

'vendor\_id' => $request->vendor\_id,

'status' => 1,

'created\_at' => Carbon::now(),

]);

/// Multiple Image Upload From her //////

$images = $request->file('multi\_img');

foreach($images as $img){

$make\_name = hexdec(uniqid()).'.'.$img->getClientOriginalExtension();

Image::make($img)->resize(800,800)->save('upload/products/multi-image/'.$make\_name);

$uploadPath = 'upload/products/multi-image/'.$make\_name;

MultiImg::insert([

'product\_id' => $product\_id,

'photo\_name' => $uploadPath,

'created\_at' => Carbon::now(),

]);

} // end foreach

/// End Multiple Image Upload From her //////

$notification = array(

'message' => 'Product Inserted Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.product')->with($notification);

} // End Method

public function EditProduct($id){

$multiImgs = MultiImg::where('product\_id',$id)->get();

$activeVendor = User::where('status','active')->where('role','vendor')->latest()->get();

$brands = Brand::latest()->get();

$categories = Category::latest()->get();

$subcategory = SubCategory::latest()->get();

$products = Product::findOrFail($id);

return view('backend.product.product\_edit',compact('brands','categories','activeVendor','products','subcategory','multiImgs'));

}// End Method

public function UpdateProduct(Request $request){

$product\_id = $request->id;

Product::findOrFail($product\_id)->update([

'brand\_id' => $request->brand\_id,

'category\_id' => $request->category\_id,

'subcategory\_id' => $request->subcategory\_id,

'product\_name' => $request->product\_name,

'product\_slug' => strtolower(str\_replace(' ','-',$request->product\_name)),

'product\_code' => $request->product\_code,

'product\_qty' => $request->product\_qty,

'product\_tags' => $request->product\_tags,

'product\_size' => $request->product\_size,

'product\_color' => $request->product\_color,

'selling\_price' => $request->selling\_price,

'discount\_price' => $request->discount\_price,

'short\_descp' => $request->short\_descp,

'long\_descp' => $request->long\_descp,

'hot\_deals' => $request->hot\_deals,

'featured' => $request->featured,

'special\_offer' => $request->special\_offer,

'special\_deals' => $request->special\_deals,

'vendor\_id' => $request->vendor\_id,

'status' => 1,

'created\_at' => Carbon::now(),

]);

$notification = array(

'message' => 'Product Updated Without Image Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.product')->with($notification);

}// End Method

public function UpdateProductThambnail(Request $request){

$pro\_id = $request->id;

$oldImage = $request->old\_img;

$image = $request->file('product\_thambnail');

$name\_gen = hexdec(uniqid()).'.'.$image->getClientOriginalExtension();

Image::make($image)->resize(800,800)->save('upload/products/thambnail/'.$name\_gen);

$save\_url = 'upload/products/thambnail/'.$name\_gen;

if (file\_exists($oldImage)) {

unlink($oldImage);

}

Product::findOrFail($pro\_id)->update([

'product\_thambnail' => $save\_url,

'updated\_at' => Carbon::now(),

]);

$notification = array(

'message' => 'Product Image Thambnail Updated Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

// Multi Image Update

public function UpdateProductMultiimage(Request $request){

$imgs = $request->multi\_img;

foreach($imgs as $id => $img ){

$imgDel = MultiImg::findOrFail($id);

unlink($imgDel->photo\_name);

$make\_name = hexdec(uniqid()).'.'.$img->getClientOriginalExtension();

Image::make($img)->resize(800,800)->save('upload/products/multi-image/'.$make\_name);

$uploadPath = 'upload/products/multi-image/'.$make\_name;

MultiImg::where('id',$id)->update([

'photo\_name' => $uploadPath,

'updated\_at' => Carbon::now(),

]);

} // end foreach

$notification = array(

'message' => 'Product Multi Image Updated Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function MulitImageDelelte($id){

$oldImg = MultiImg::findOrFail($id);

unlink($oldImg->photo\_name);

MultiImg::findOrFail($id)->delete();

$notification = array(

'message' => 'Product Multi Image Deleted Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function ProductInactive($id){

Product::findOrFail($id)->update(['status' => 0]);

$notification = array(

'message' => 'Product Inactive',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function ProductActive($id){

Product::findOrFail($id)->update(['status' => 1]);

$notification = array(

'message' => 'Product Active',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function ProductDelete($id){

$product = Product::findOrFail($id);

unlink($product->product\_thambnail);

Product::findOrFail($id)->delete();

$imges = MultiImg::where('product\_id',$id)->get();

foreach($imges as $img){

unlink($img->photo\_name);

MultiImg::where('product\_id',$id)->delete();

}

$notification = array(

'message' => 'Product Deleted Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Method

public function ProductStock(){

$products = Product::latest()->get();

return view('backend.product.product\_stock',compact('products'));

}// End Method

}

ក.២៣ ទំព័របង្កើតផលិតផល (Create Report Page)

A screenshot of a computer

Description automatically generated

រូបភាព ក.៧៦ បង្ហាញពី Create Report Page

* ReportController.php

<?php

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use DateTime;

use App\Models\Order;

use App\Models\User;

class ReportController extends Controller{

public function ReportView(){

return view('backend.report.report\_view');

} // End Method

public function SearchByDate(Request $request){

$date = new DateTime($request->date);

$formatDate = $date->format('d F Y');

$orders = Order::where('order\_date',$formatDate)->latest()->get();

return view('backend.report.report\_by\_date',compact('orders','formatDate'));

}// End Method

public function SearchByMonth(Request $request){

$month = $request->month;

$year = $request->year\_name;

$orders = Order::where('order\_month',$month)->where('order\_year',$year)->latest()->get();

return view('backend.report.report\_by\_month',compact('orders','month','year'));

}// End Method

public function SearchByYear(Request $request){

$year = $request->year;

$orders = Order::where('order\_year',$year)->latest()->get();

return view('backend.report.report\_by\_year',compact('orders','year'));

}// End Method

public function OrderByUser(){

$users = User::where('role','user')->latest()->get();

return view('backend.report.report\_by\_user',compact('users'));

}// End Method

public function SearchByUser(Request $request){

$users = $request->user;

$orders = Order::where('user\_id',$users)->latest()->get();

return view('backend.report.report\_by\_user\_show',compact('orders','users'));

}// End Method

}

A screenshot of a computer

Description automatically generatedក.២៧ ទំព័របញ្ជីអតិថិជន (Customer List Page)

រូបភាព ក.៧៧ បង្ហាញពី Customer List Page

* ActiveUserController.php

<?php

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\User;

class ActiveUserController extends Controller

{

public function AllUser(){

$users = User::where('role','user')->latest()->get();

return view('backend.user.user\_all\_data',compact('users'));

} // End Mehtod

public function AllVendor(){

$vendors = User::where('role','vendor')->latest()->get();

return view('backend.user.vendor\_all\_data',compact('vendors'));

} // End Mehtod

}

A screenshot of a computer

Description automatically generatedក.២៨ ទំព័របង្កើតសិទ្ធប្រើប្រាស់ (Create Admin User Page)

រូបភាព ក.៧៨ បង្ហាញពី Create Admin User Page

* AdminController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use App\Models\User;

use Illuminate\Support\Facades\Hash;

use Spatie\Permission\Models\Role;

use Spatie\Permission\Models\Permission;

use App\Notifications\VendorApproveNotification;

use Illuminate\Support\Facades\Notification;

class AdminController extends Controller

{

public function AdminDashboard(){

return view('admin.index');

} // End Mehtod

public function AdminLogin(){

return view('admin.admin\_login');

} // End Mehtod

public function AdminDestroy(Request $request){

Auth::guard('web')->logout();

$request->session()->invalidate();

$request->session()->regenerateToken();

return redirect('/admin/login');

} // End Mehtod

public function AdminProfile(){

$id = Auth::user()->id;

$adminData = User::find($id);

return view('admin.admin\_profile\_view',compact('adminData'));

} // End Mehtod

public function AdminProfileStore(Request $request){

$id = Auth::user()->id;

$data = User::find($id);

$data->name = $request->name;

$data->email = $request->email;

$data->phone = $request->phone;

$data->address = $request->address;

if ($request->file('photo')) {

$file = $request->file('photo');

@unlink(public\_path('upload/admin\_images/'.$data->photo));

$filename = date('YmdHi').$file->getClientOriginalName();

$file->move(public\_path('upload/admin\_images'),$filename);

$data['photo'] = $filename;

}

$data->save();

$notification = array(

'message' => 'Admin Profile Updated Successfully',

'alert-type' => 'success'

);

return redirect()->back()->with($notification);

} // End Mehtod

public function AdminChangePassword(){

return view('admin.admin\_change\_password');

} // End Mehtod

public function AdminUpdatePassword(Request $request){

// Validation

$request->validate([

'old\_password' => 'required',

'new\_password' => 'required|confirmed',

]);

// Match The Old Password

if (!Hash::check($request->old\_password, auth::user()->password)) {

return back()->with("error", "Old Password Doesn't Match!!");

}

// Update The new password

User::whereId(auth()->user()->id)->update([

'password' => Hash::make($request->new\_password)

]);

return back()->with("status", " Password Changed Successfully");

} // End Mehtod

public function InactiveVendor(){

$inActiveVendor = User::where('status','inactive')->where('role','vendor')->latest()->get();

return view('backend.vendor.inactive\_vendor',compact('inActiveVendor'));

}// End Mehtod

public function ActiveVendor(){

$ActiveVendor = User::where('status','active')->where('role','vendor')->latest()->get();

return view('backend.vendor.active\_vendor',compact('ActiveVendor'));

}// End Mehtod

public function InactiveVendorDetails($id){

$inactiveVendorDetails = User::findOrFail($id);

return view('backend.vendor.inactive\_vendor\_details',compact('inactiveVendorDetails'));

}// End Mehtod

public function ActiveVendorApprove(Request $request){

$verdor\_id = $request->id;

$user = User::findOrFail($verdor\_id)->update([

'status' => 'active',

]);

$notification = array(

'message' => 'Vendor Active Successfully',

'alert-type' => 'success'

);

$vuser = User::where('role','vendor')->get();

Notification::send($vuser, new VendorApproveNotification($request));

return redirect()->route('active.vendor')->with($notification);

}// End Mehtod

public function ActiveVendorDetails($id){

$activeVendorDetails = User::findOrFail($id);

return view('backend.vendor.active\_vendor\_details',compact('activeVendorDetails'));

}// End Mehtod

public function InActiveVendorApprove(Request $request){

$verdor\_id = $request->id;

$user = User::findOrFail($verdor\_id)->update([

'status' => 'inactive',

]);

$notification = array(

'message' => 'Vendor InActive Successfully',

'alert-type' => 'success'

);

return redirect()->route('inactive.vendor')->with($notification);

}// End Mehtod

///////////// Admin All Method //////////////

public function AllAdmin(){

$alladminuser = User::where('role','admin')->latest()->get();

return view('backend.admin.all\_admin',compact('alladminuser'));

}// End Mehtod

public function AddAdmin(){

$roles = Role::all();

return view('backend.admin.add\_admin',compact('roles'));

}// End Mehtod

public function AdminUserStore(Request $request){

$user = new User();

$user->username = $request->username;

$user->name = $request->name;

$user->email = $request->email;

$user->phone = $request->phone;

$user->address = $request->address;

$user->password = Hash::make($request->password);

$user->role = 'admin';

$user->status = 'active';

$user->save();

if ($request->roles) {

$user->assignRole($request->roles);

}

$notification = array(

'message' => 'New Admin User Inserted Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.admin')->with($notification);

}// End Mehtod

public function EditAdminRole($id){

$user = User::findOrFail($id);

$roles = Role::all();

return view('backend.admin.edit\_admin',compact('user','roles'));

}// End Mehtod

public function AdminUserUpdate(Request $request,$id){

$user = User::findOrFail($id);

$user->username = $request->username;

$user->name = $request->name;

$user->email = $request->email;

$user->phone = $request->phone;

$user->address = $request->address;

$user->role = 'admin';

$user->status = 'active';

$user->save();

$user->roles()->detach();

if ($request->roles) {

$user->assignRole($request->roles);

}

$notification = array(

'message' => 'New Admin User Updated Successfully',

'alert-type' => 'success'

);

return redirect()->route('all.admin')->with($notification);

}// End Mehtod

public function DeleteAdminRole($id){

$user = User::findOrFail($id);

if (!is\_null($user)) {

$user->delete();

}

$notification = array(

'message' => 'Admin User Deleted Successfully',

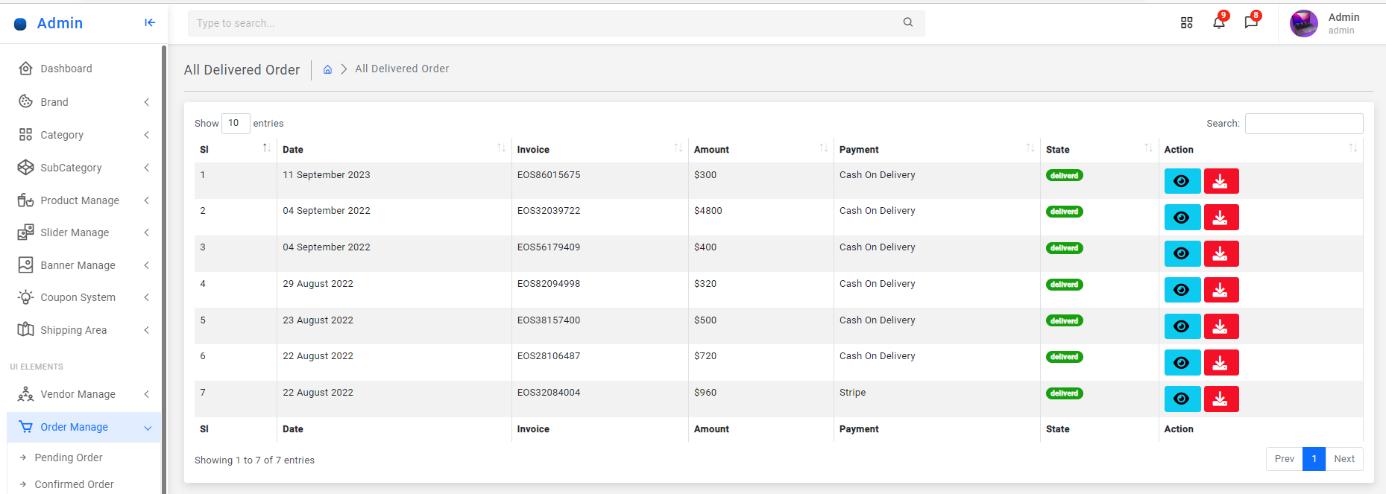
'alert-type' => 'success'

);

return redirect()->back()->with($notification);

}// End Mehtod

}

ក.៣៤ ទំព័របញ្ជាក់ការបញ្ជាទិញ (Confirm Order Page)

រូបភាព ក.៧៩ បង្ហាញពី Confirm Order Page

* OrderController.php

<?php

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Order;

use App\Models\OrderItem;

use App\Models\Product;

use Gloudemans\Shoppingcart\Facades\Cart;

use Illuminate\Support\Facades\Session;

use Carbon\Carbon;

use Auth;

use Barryvdh\DomPDF\Facade\Pdf;

use DB;

class OrderController extends Controller

{

public function PendingOrder(){

$orders = Order::where('status','pending')->orderBy('id','DESC')->get();

return view('backend.orders.pending\_orders',compact('orders'));

} // End Method

public function AdminOrderDetails($order\_id){

$order = Order::with('division','district','state','user')->where('id',$order\_id)->first();

$orderItem = OrderItem::with('product')->where('order\_id',$order\_id)->orderBy('id','DESC')->get();

return view('backend.orders.admin\_order\_details',compact('order','orderItem'));

}// End Method

public function AdminConfirmedOrder(){

$orders = Order::where('status','confirm')->orderBy('id','DESC')->get();

return view('backend.orders.confirmed\_orders',compact('orders'));

} // End Method

public function AdminProcessingOrder(){

$orders = Order::where('status','processing')->orderBy('id','DESC')->get();

return view('backend.orders.processing\_orders',compact('orders'));

} // End Method

public function AdminDeliveredOrder(){

$orders = Order::where('status','deliverd')->orderBy('id','DESC')->get();

return view('backend.orders.delivered\_orders',compact('orders'));

} // End Method

public function PendingToConfirm($order\_id){

Order::findOrFail($order\_id)->update(['status' => 'confirm']);

$notification = array(

'message' => 'Order Confirm Successfully',

'alert-type' => 'success'

);

return redirect()->route('admin.confirmed.order')->with($notification);

}// End Method

public function ConfirmToProcess($order\_id){

Order::findOrFail($order\_id)->update(['status' => 'processing']);

$notification = array(

'message' => 'Order Processing Successfully',

'alert-type' => 'success'

);

return redirect()->route('admin.processing.order')->with($notification);

}// End Method

public function ProcessToDelivered($order\_id){

$product = OrderItem::where('order\_id',$order\_id)->get();

foreach($product as $item){

Product::where('id',$item->product\_id)

->update(['product\_qty' => DB::raw('product\_qty-'.$item->qty) ]);

}

Order::findOrFail($order\_id)->update(['status' => 'deliverd']);

$notification = array(

'message' => 'Order Deliverd Successfully',

'alert-type' => 'success'

);

return redirect()->route('admin.delivered.order')->with($notification);

}// End Method

public function AdminInvoiceDownload($order\_id){

$order = Order::with('division','district','state','user')->where('id',$order\_id)->first();

$orderItem = OrderItem::with('product')->where('order\_id',$order\_id)->orderBy('id','DESC')->get();

$pdf = Pdf::loadView('backend.orders.admin\_order\_invoice', compact('order','orderItem'))->setPaper('a4')->setOption([

'tempDir' => public\_path(),

'chroot' => public\_path(),

]);

return $pdf->download('invoice.pdf');

}// End Method

}

A screenshot of a computer

Description automatically generatedក.៣៣ ទំព័របញ្ជីបញ្ជាទិញដោយអ្នកប្រើប្រាស់ (Order List Page)

រូបភាព ក.៨០ បង្ហាញពី Order by User List Page

* ReportController.php

<?php

namespace App\Http\Controllers\Backend;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use DateTime;

use App\Models\Order;

use App\Models\User;

class ReportController extends Controller

{

public function ReportView(){

return view('backend.report.report\_view');

} // End Method

public function SearchByDate(Request $request){

$date = new DateTime($request->date);

$formatDate = $date->format('d F Y');

$orders = Order::where('order\_date',$formatDate)->latest()->get();

return view('backend.report.report\_by\_date',compact('orders','formatDate'));

}// End Method

public function SearchByMonth(Request $request){

$month = $request->month;

$year = $request->year\_name;

$orders = Order::where('order\_month',$month)->where('order\_year',$year)->latest()->get();

return view('backend.report.report\_by\_month',compact('orders','month','year'));

}// End Method

public function SearchByYear(Request $request){

$year = $request->year;

$orders = Order::where('order\_year',$year)->latest()->get();

return view('backend.report.report\_by\_year',compact('orders','year'));

}// End Method

public function OrderByUser(){

$users = User::where('role','user')->latest()->get();

return view('backend.report.report\_by\_user',compact('users'));

}// End Method

public function SearchByUser(Request $request){

$users = $request->user;

$orders = Order::where('user\_id',$users)->latest()->get();

return view('backend.report.report\_by\_user\_show',compact('orders','users'));

}// End Method }