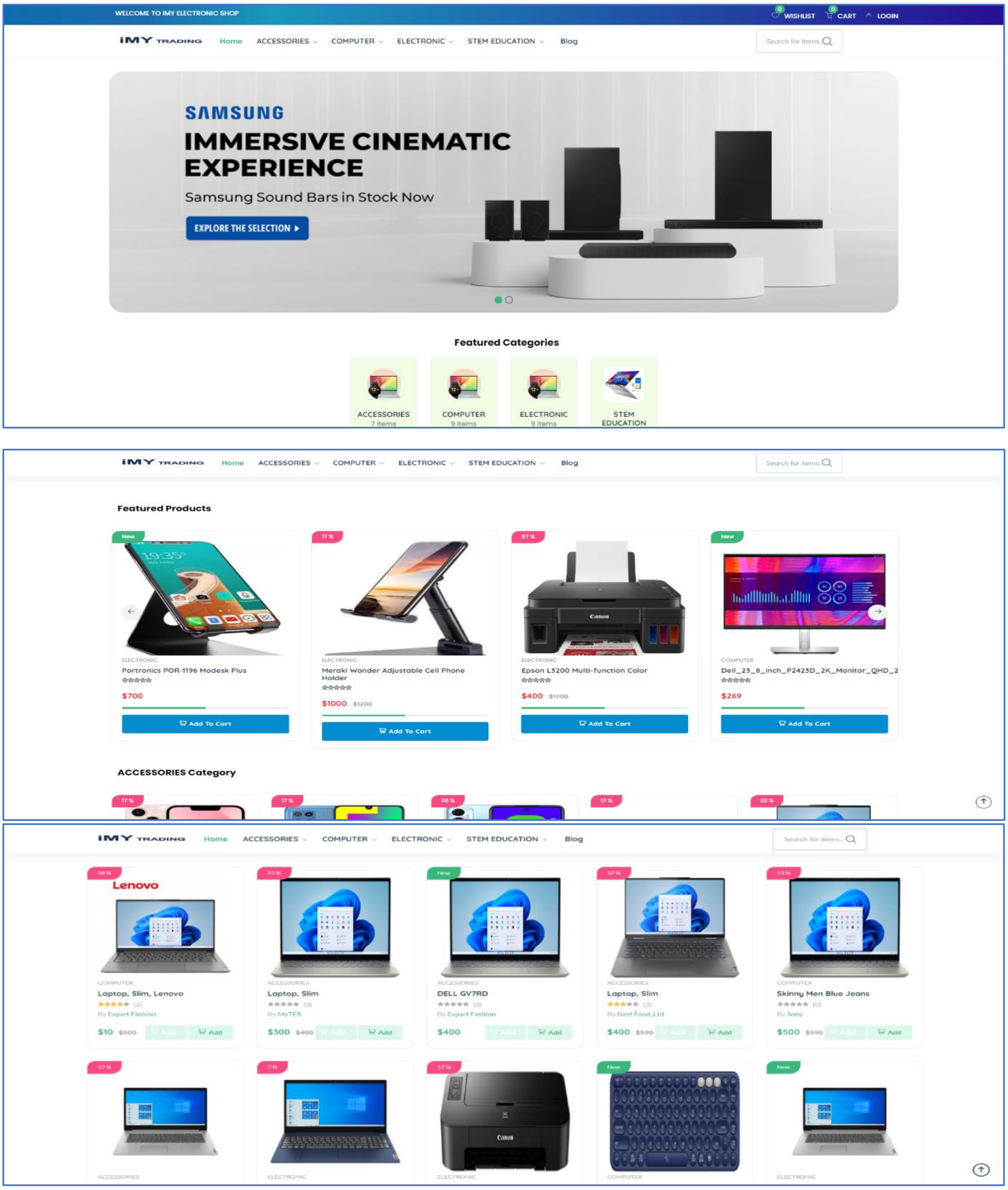
ឧបសម្ព័ន្ធ ក

Coding

ក្រុមសិក្សាស្រាវជ្រាវធ្វើការបង្ហាញអំពីការសរសេរកូដដែលមាននៅទំព័រនីមួយៗដែលក្រុមសិក្សាស្រាវជ្រាវ នឹងធ្វើការរៀបរាប់ទាំងការសរសេរ Front End និង Back End ដូចតទៅ៖

ក.១ ទំព័រដើមនៃគេហទំព័រ (Home Page)



រូបភាព ក.៥៤ បង្ហាញពី ទំព័រដើម

- IndexController.php

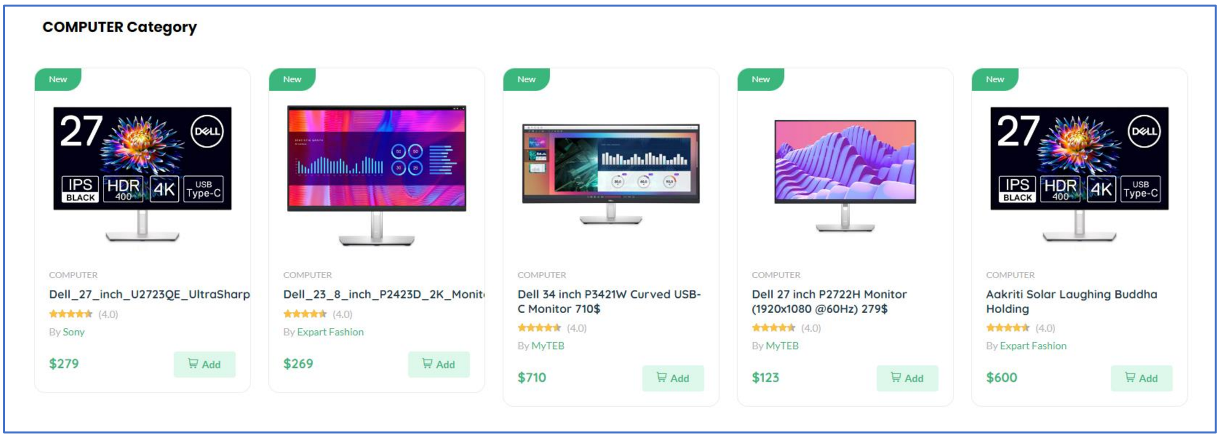
```
<?php
namespace App\Http\Controllers\Frontend;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Category;
use App\Models\SubCategory;
use App\Models\MultiImg;
use App\Models\Brand;
use App\Models\Product;
use App\Models\User;
class IndexController extends Controller
{
    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','s
kip_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_siz
e','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','newP
roduct' ) );
     }
      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','breadsubca
t','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)



COMPUTER Category

COMPUTER
Dell_27_inch_U2723QE_UltraSharp
★★★★★ (4.0)
By Sony
$279

COMPUTER
Dell_23_8_inch_P2423D_2K_Monit
★★★★★ (4.0)
By Expart Fashion
$269

COMPUTER
Dell 34 inch P3421W Curved USB-C Monitor 710$
★★★★★ (4.0)
By MyTEB
$710

COMPUTER
Dell 27 inch P2722H Monitor (1920x1080 @60Hz) 279$
★★★★★ (4.0)
By MyTEB
$123

COMPUTER
Aakriti Solar Laughing Buddha Holding
★★★★★ (4.0)
By Expart Fashion
$600

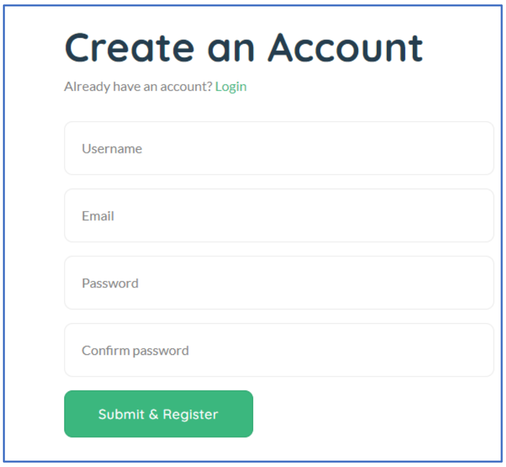
រូបភាព ក.៥៥ បង្ហាញពី Show product by category

- ➢ 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_pro
duct_1','skip_category_2','skip_product_2','skip_category_3','skip_product_3','hot_deals','spe
cial_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','multi
Image','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','new
Product' ) );
    }
    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.៣ ទំព័រព័ត៌មានចុះឈ្មោះចូលក្នុងប្រព័ន្ធ (Register Page)



Create an Account

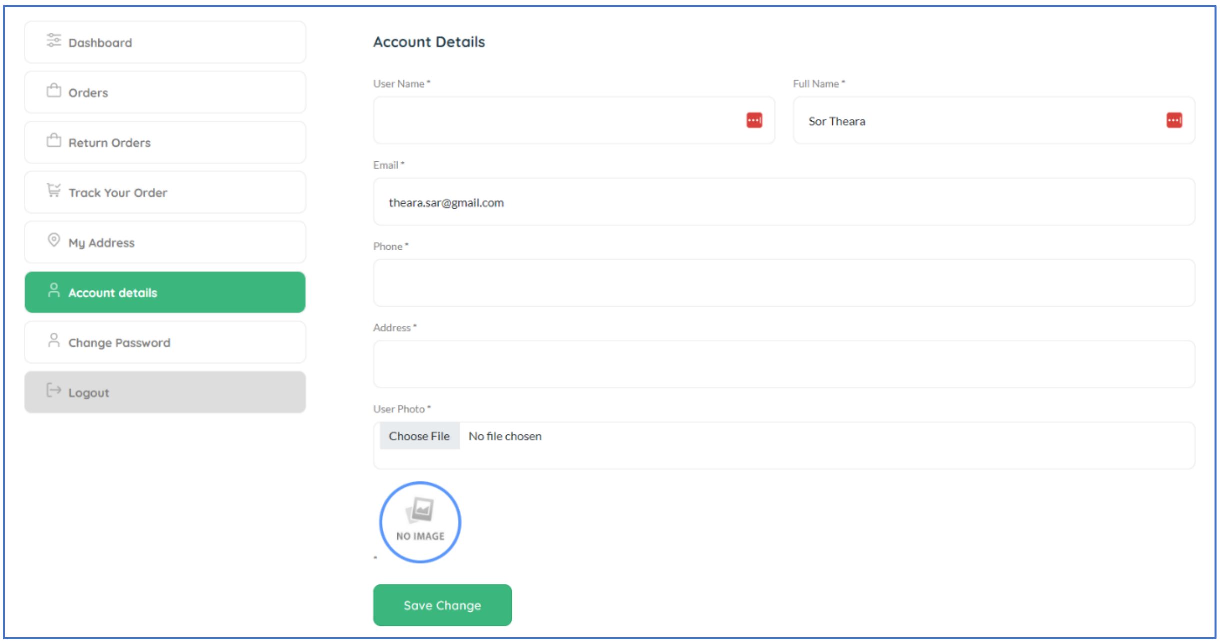
Already have an account? Login

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);
    }
}
```

ក.៤ ទំព័រព័ត៌មានគណនី (Account Details Page)



Account Details

User Name *

Full Name * Sor Theara

Email * theara.sar@gmail.com

Phone *

Address *

User Photo * Choose File No file chosen

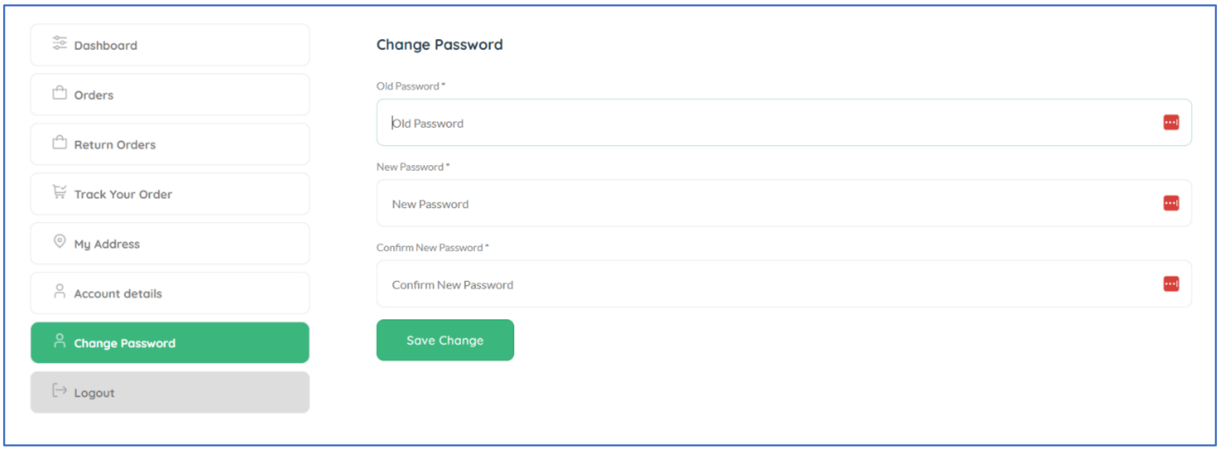
រូបភាព ក.៤៧ បង្ហាញពី Customer Profile Page

- 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
}
```

ក.៩ ទំព័រផ្លាស់ប្តូរលេខសម្ងាត់ (Change Password Page)



Change Password

Old Password *

Old Password

New Password *

New Password

Confirm New Password *

Confirm New Password

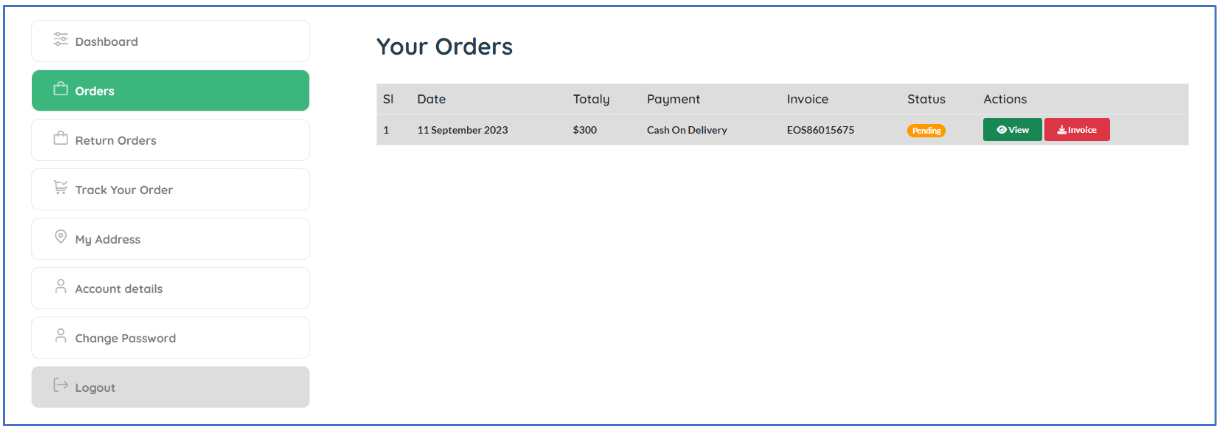
រូបភាព ក.៥៨ បង្ហាញពី Change Password Page

- ➢ 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)



Your Orders

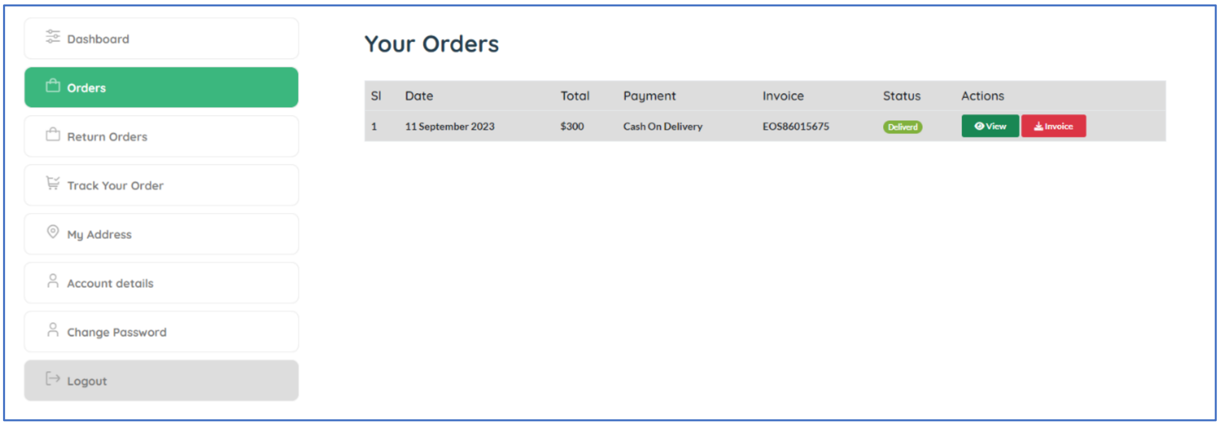
Sl	Date	Totaly	Payment	Invoice	Status	Actions
1	11 September 2023	$300	Cash On Delivery	EOS86015675	Pending	View Invoice

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
}
```

ក.៧ ទំព័រនៃការចំណាយរបស់អតិថិជន (Payment List Page)



Your Orders

Sl	Date	Total	Payment	Invoice	Status	Actions
1	11 September 2023	$300	Cash On Delivery	EOS86015675	Deliverd	View Invoice

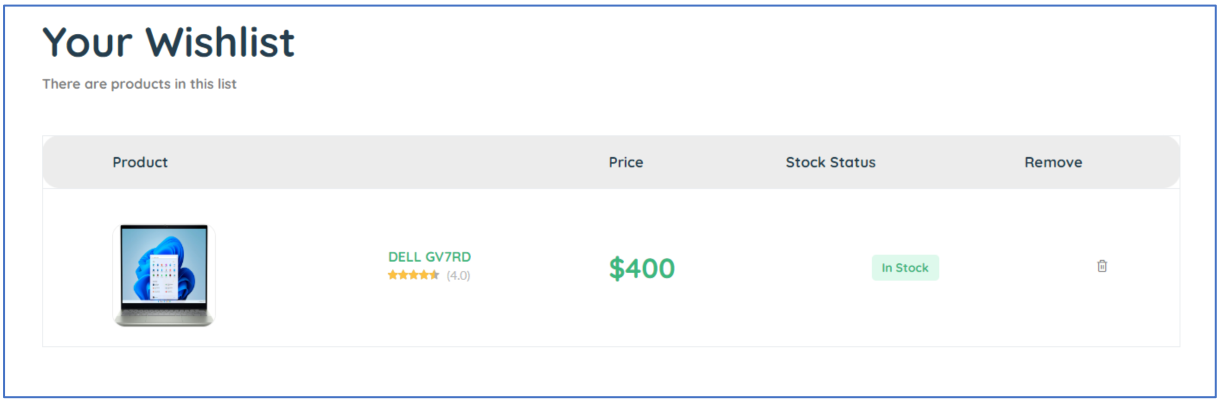
រូបភាព ក.៦០ បង្ហាញពី Payment List Page

- 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)



Your Wishlist

There are products in this list

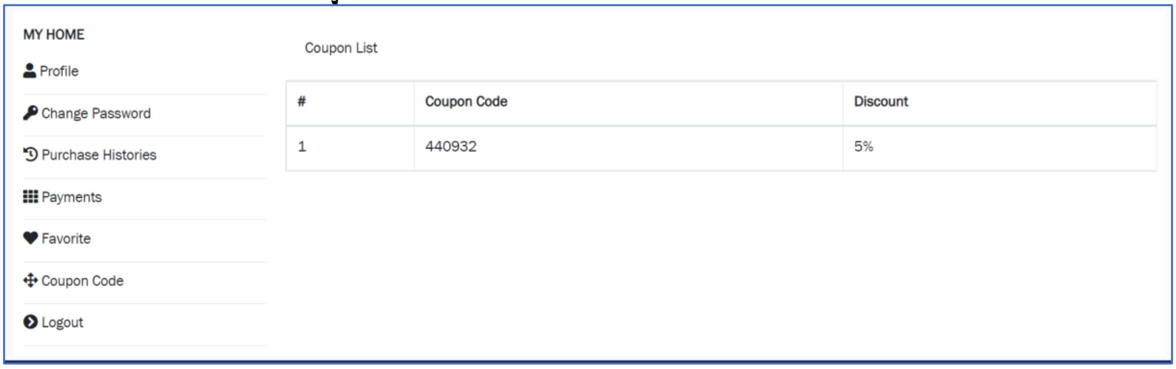
Product		Price	Stock Status	Remove
	DELL GV7RD ★★★★★ (4.0)	$400	In Stock	🗑

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)



Coupon List

#	Coupon Code	Discount
1	440932	5%

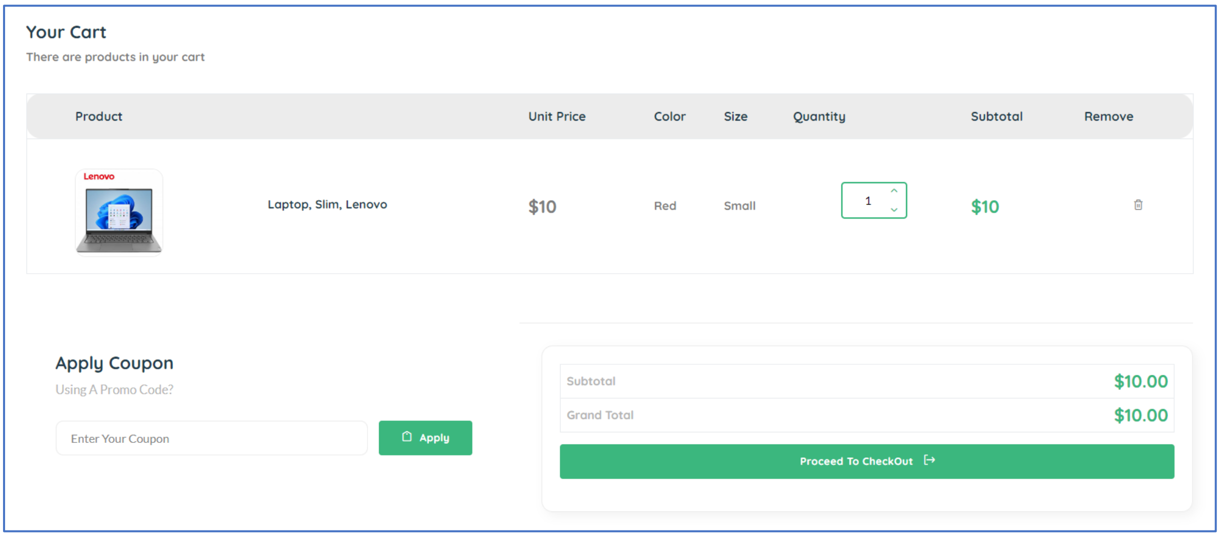
រូបភាព ក.៦២ បង្ហាញពី Coupon Page

- 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 ទំព័រអេដធូកាត (Add to Cart Page)



Your Cart

There are products in your cart

Product		Unit Price	Color	Size	Quantity	Subtotal	Remove
	Laptop, Slim, Lenovo	$10	Red	Small	1	$10	

Apply Coupon

Using A Promo Code?

Enter Your Coupon

Subtotal	$10.00
Grand Total	$10.00

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' ] );
```

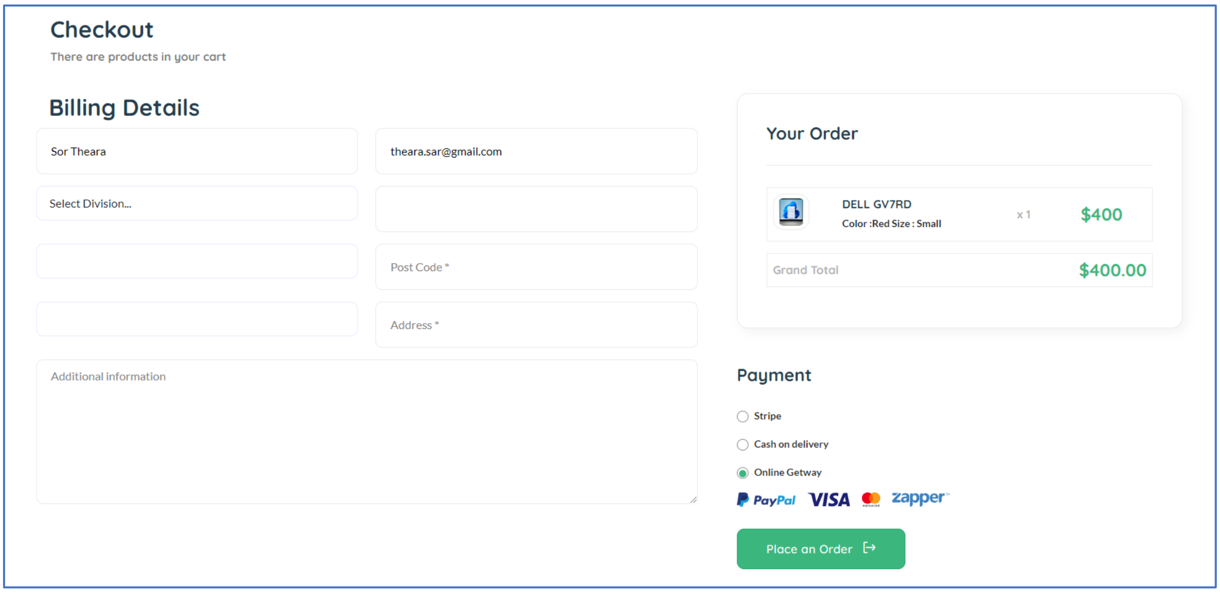
```
    }else{
        Cart::add( [
            'id' => $id,
            'name' => $request->product_name,
            'qty' => $request->quantity,
            'price' => $product->discount_price,
            'weight' => 1,
            'options' => [
                'image' => $product->product_thambnail,
                '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)



Checkout

There are products in your cart

Billing Details

Sor Theara

theara.sar@gmail.com

Select Division...

Post Code *

Address *

Additional information

Your Order

DELL GV7RD Color :Red Size : Small	x 1	$400
Grand Total		$400.00

Payment

○ Stripe

○ Cash on delivery

● Online Getway

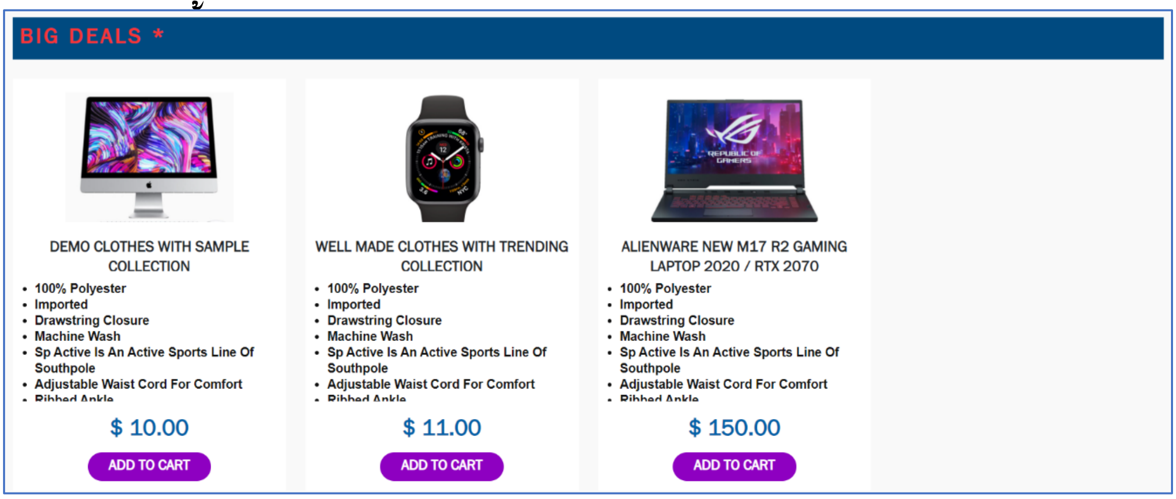
រូបភាព ក.៦៤ បង្ហាញពី Payment Page

- ➢ 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 ទំព័ររីវ្យូរ (Review Product Page)



BIG DEALS *

DEMO CLOTHES WITH SAMPLE COLLECTION

- 100% Polyester
- Imported
- Drawstring Closure
- Machine Wash
- Sp Active Is An Active Sports Line Of Southpole
- Adjustable Waist Cord For Comfort
- Ribbed Ankle

$ 10.00

ADD TO CART

WELL MADE CLOTHES WITH TRENDING COLLECTION

- 100% Polyester
- Imported
- Drawstring Closure
- Machine Wash
- Sp Active Is An Active Sports Line Of Southpole
- Adjustable Waist Cord For Comfort
- Ribbed Ankle

$ 11.00

ADD TO CART

ALIENWARE NEW M17 R2 GAMING LAPTOP 2020 / RTX 2070

- 100% Polyester
- Imported
- Drawstring Closure
- Machine Wash
- Sp Active Is An Active Sports Line Of Southpole
- Adjustable Waist Cord For Comfort
- Ribbed Ankle

$ 150.00

ADD TO CART

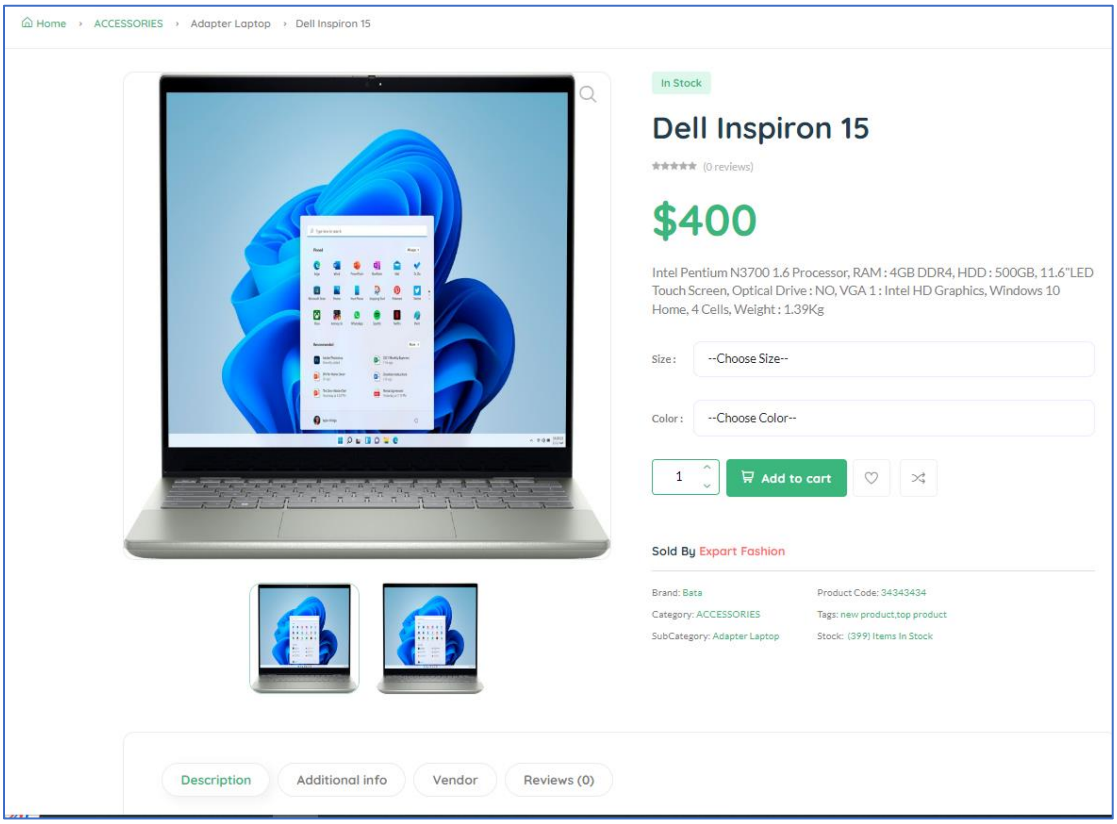
រូបភាព ក.៦៥ បង្ហាញពី Deal Page

- 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 ទំព័រព័ត៌មានលម្អិតរបស់ផលិតផល (Product Detail Page)



Home › ACCESSORIES › Adapter Laptop › Dell Inspiron 15

In Stock

Dell Inspiron 15

★★★★★ (0 reviews)

$400

Intel Pentium N3700 1.6 Processor, RAM : 4GB DDR4, HDD : 500GB, 11.6"LED Touch Screen, Optical Drive : NO, VGA 1 : Intel HD Graphics, Windows 10 Home, 4 Cells, Weight : 1.39Kg

Size : --Choose Size--

Color : --Choose Color--

1 Add to cart

Sold By Expart Fashion

Brand: Bata

Product Code: 34343434

Category: ACCESSORIES

Tags: new product,top product

SubCategory: Adapter Laptop

Stock: (399) Items In Stock

Description Additional info Vendor Reviews (0)

រូបភាព ក.៦៦ បង្ហាញពី Product Detail Page

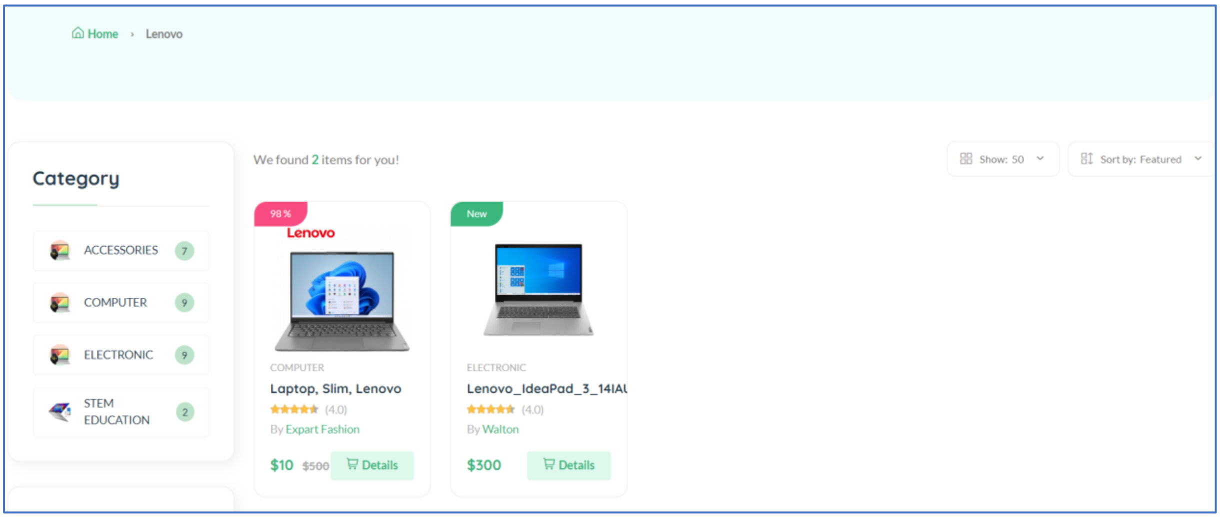
- ➢ ProductController.php

```
<?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 ទំព័រស្វែករកផលិតផល (Search Result Page)



We found 2 items for you!

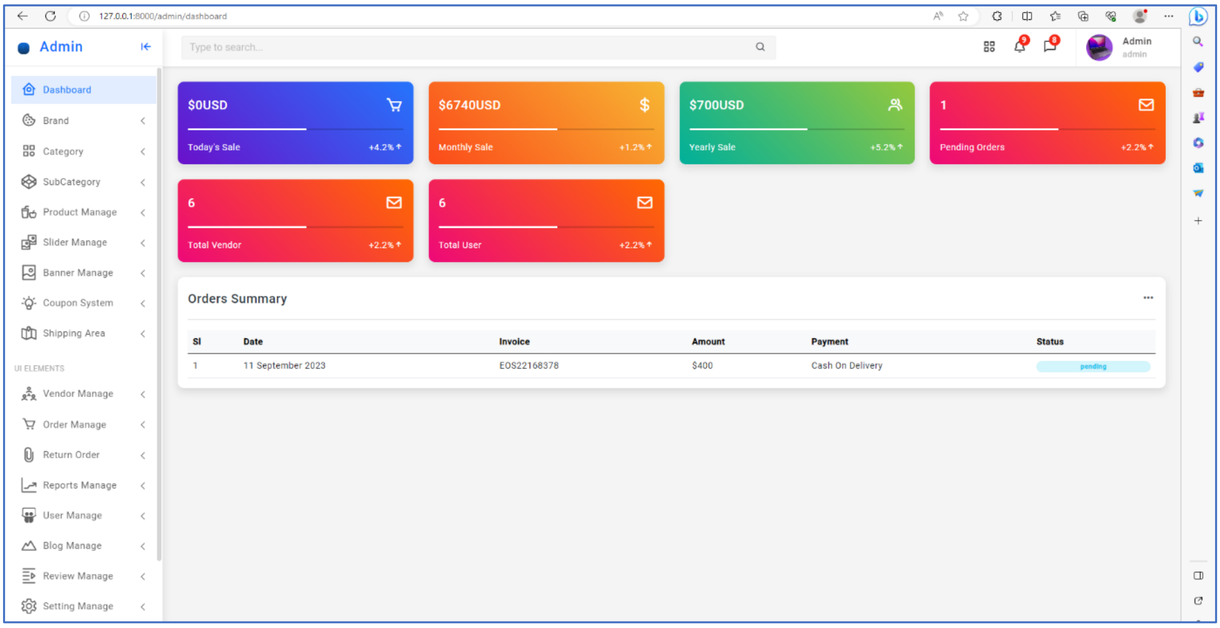
រូបភាព ក.៦៧ បង្ហាញពី Search Result Page

➢ 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' ) );
    }
}
```

ក.១៥ ទំព័រដាស់បត (Dashboard Page)


127.0.0.1:8000/admin/dashboard

Admin

Type to search...

Admin
admin

- Dashboard
- Brand
- Category
- SubCategory
- Product Manage
- Slider Manage
- Banner Manage
- Coupon System
- Shipping Area

UI ELEMENTS

- Vendor Manage
- Order Manage
- Return Order
- Reports Manage
- User Manage
- Blog Manage
- Review Manage
- Setting Manage

$0USD
Today's Sale +4.2%

$6740USD
Monthly Sale +1.2%

$700USD
Yearly Sale +5.2%

1
Pending Orders +2.2%

6
Total Vendor +2.2%

6
Total User +2.2%

Orders Summary

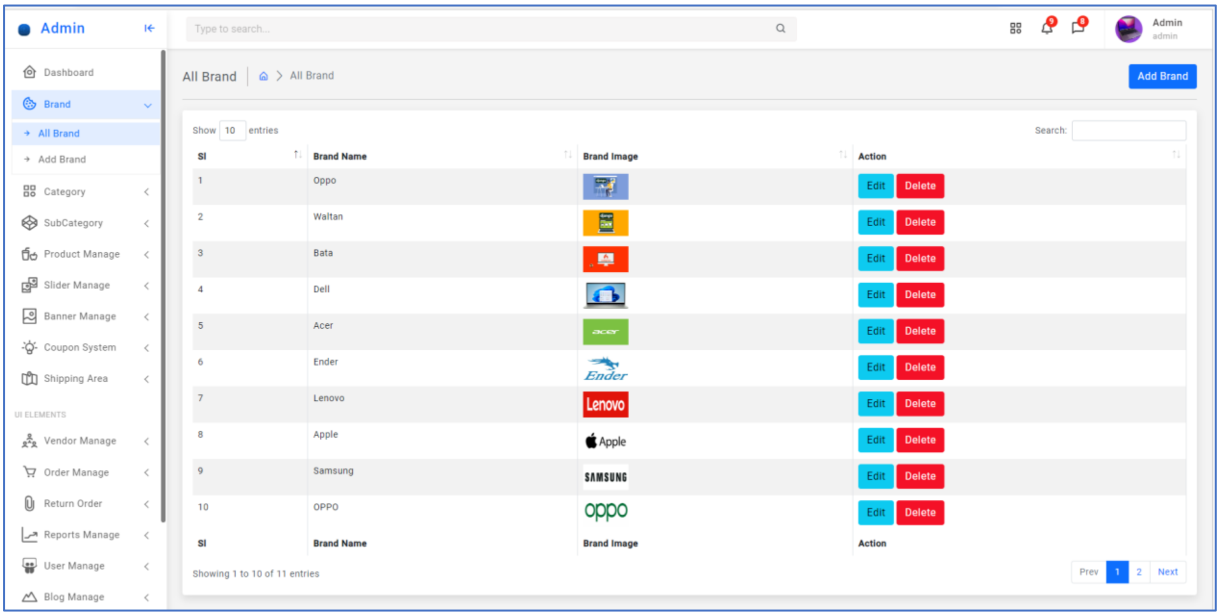
Sl	Date	Invoice	Amount	Payment	Status
1	11 September 2023	EOS22168378	$400	Cash On Delivery	pending

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)



Sl	Brand Name	Brand Image	Action
1	Oppo		Edit Delete
2	Waltan		Edit Delete
3	Bata		Edit Delete
4	Dell		Edit Delete
5	Acer	acer	Edit Delete
6	Ender	Ender	Edit Delete
7	Lenovo	Lenovo	Edit Delete
8	Apple	Apple	Edit Delete
9	Samsung	SAMSUNG	Edit Delete
10	OPPO	oppo	Edit Delete
Sl	Brand Name	Brand Image	Action

Showing 1 to 10 of 11 entries

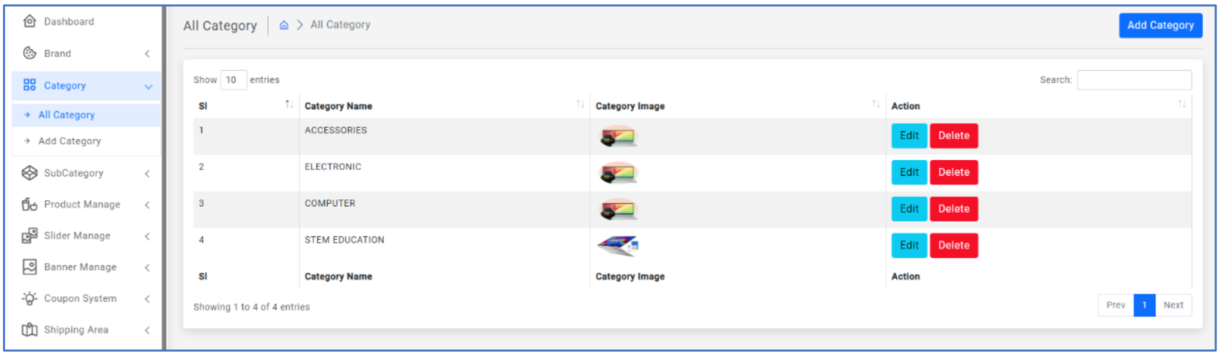
រូបភាព ក.៦៩ បង្ហាញពី Brand List Page

- 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
}
```

ក.១៦ ទំព័របញ្ជីប្រភេទផលិតផល (Category List Page)



Sl	Category Name	Category Image	Action
1	ACCESSORIES		Edit Delete
2	ELECTRONIC		Edit Delete
3	COMPUTER		Edit Delete
4	STEM EDUCATION		Edit Delete
Sl	Category Name	Category Image	Action

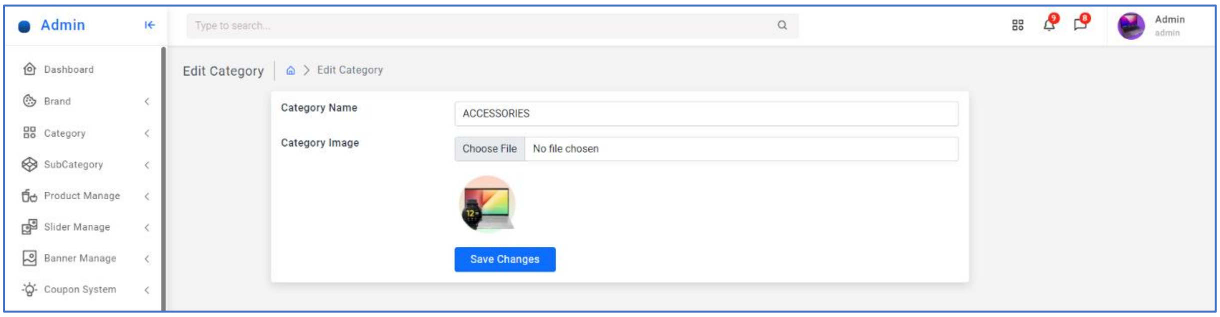
Showing 1 to 4 of 4 entries

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
}
```

ក.១៧ ទំព័របង្កើតប្រភេទផលិតផល (Create Category Page)



Category Name: ACCESSORIES

Category Image: No file chosen

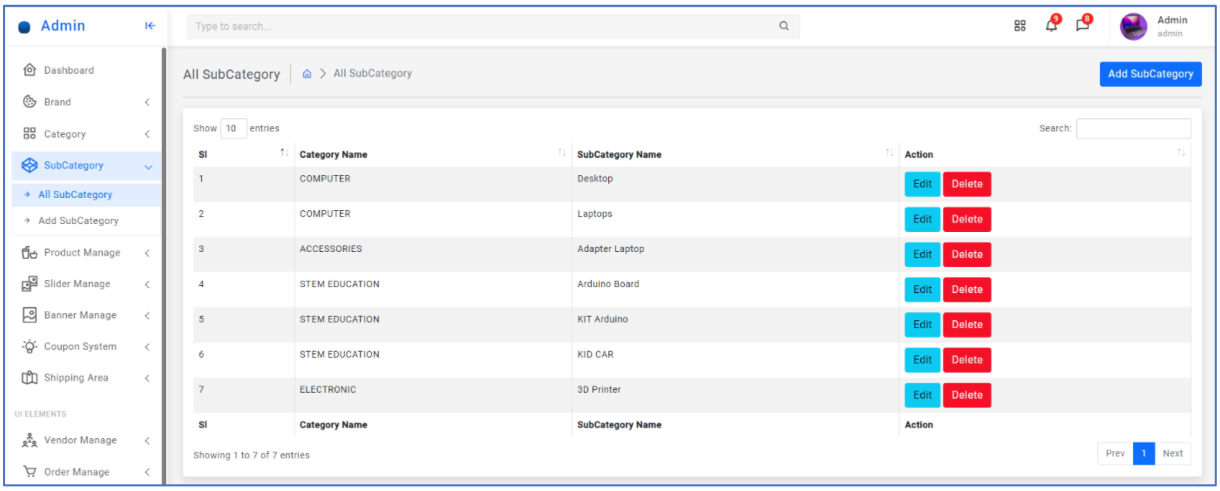
រូបភាព ក.៧១ បង្ហាញពី Create Category Page

- ➢ 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
}
```

ក.១៩ ទំព័រប្រភេទផលិតផលរង (Sub Category Page)



Sl	Category Name	SubCategory Name	Action
1	COMPUTER	Desktop	Edit Delete
2	COMPUTER	Laptops	Edit Delete
3	ACCESSORIES	Adapter Laptop	Edit Delete
4	STEM EDUCATION	Arduino Board	Edit Delete
5	STEM EDUCATION	KIT Arduino	Edit Delete
6	STEM EDUCATION	KID CAR	Edit Delete
7	ELECTRONIC	3D Printer	Edit Delete
Sl	Category Name	SubCategory Name	Action

Showing 1 to 7 of 7 entries

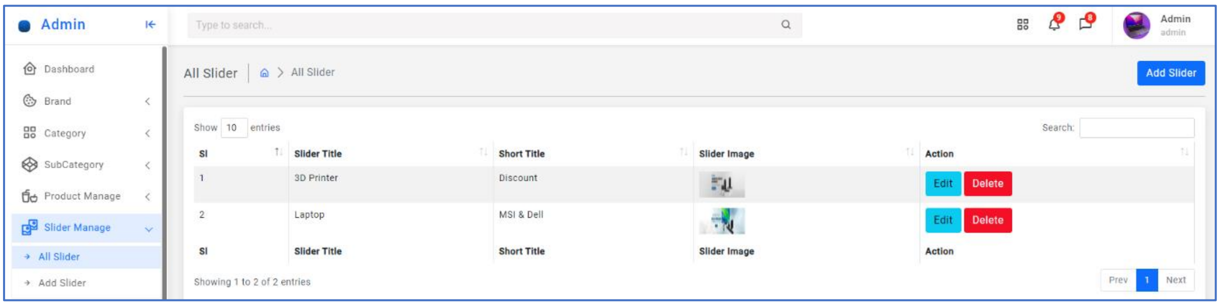
រូបភាព ក.៧២ បង្ហាញពី Sub Category Page

➢ 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
}
```

ក.២០ ទំព័របង្កើតស្លាយដឺ (Create Slider Page)



All Slider | All Slider

Show 10 entries Search:

Sl	Slider Title	Short Title	Slider Image	Action
1	3D Printer	Discount		Edit Delete
2	Laptop	MSI & Dell		Edit Delete
Sl	Slider Title	Short Title	Slider Image	Action

Showing 1 to 2 of 2 entries

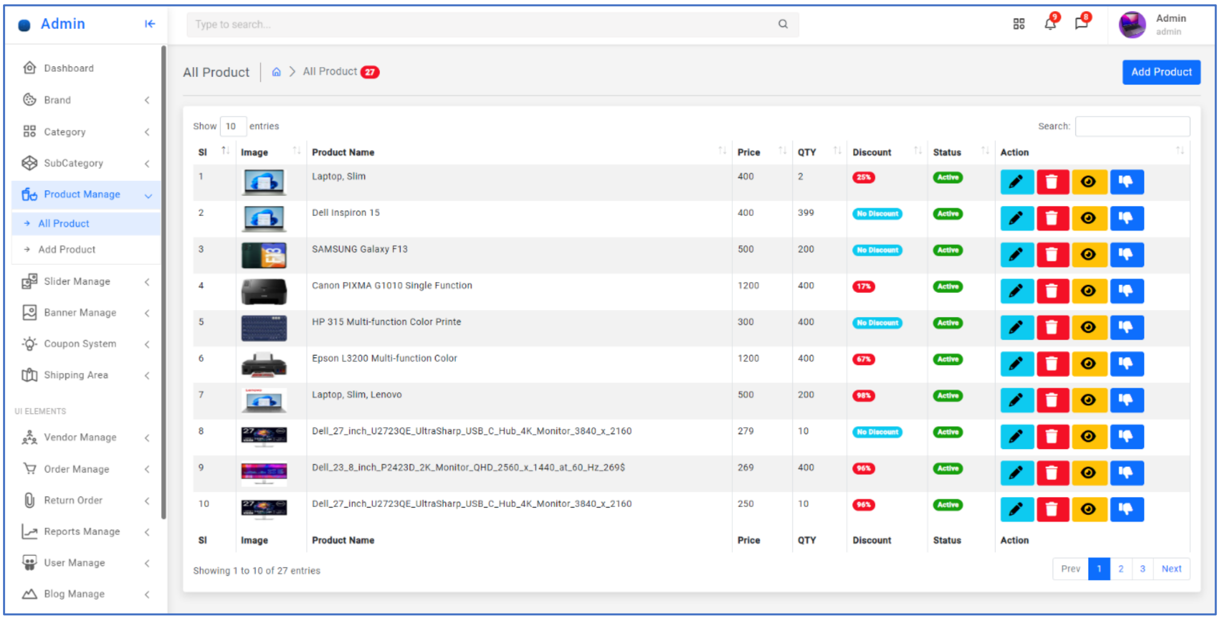
រូបភាព ក.៧៣ បង្ហាញពី Create Slider Page

- ➢ 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
}
```

ក.២១ ទំព័របញ្ជីផលិតផល (Product List Page)



Sl	Image	Product Name	Price	QTY	Discount	Status	Action
1		Laptop, Slim	400	2	25%	Active	
2		Dell Inspiron 15	400	399	No Discount	Active	
3		SAMSUNG Galaxy F13	500	200	No Discount	Active	
4		Canon PIXMA G1010 Single Function	1200	400	17%	Active	
5		HP 315 Multi-function Color Printe	300	400	No Discount	Active	
6		Epson L3200 Multi-function Color	1200	400	67%	Active	
7		Laptop, Slim, Lenovo	500	200	98%	Active	
8		Dell_27_inch_U2723QE_UltraSharp_USB_C_Hub_4K_Monitor_3840_x_2160	279	10	No Discount	Active	
9		Dell_23_8_inch_P2423D_2K_Monitor_QHD_2560_x_1440_at_60_Hz_269$	269	400	96%	Active	
10		Dell_27_inch_U2723QE_UltraSharp_USB_C_Hub_4K_Monitor_3840_x_2160	250	10	96%	Active	
Sl	Image	Product Name	Price	QTY	Discount	Status	Action

Showing 1 to 10 of 27 entries

រូបភាព ក.៧៤ បង្ហាញពី Product List Page

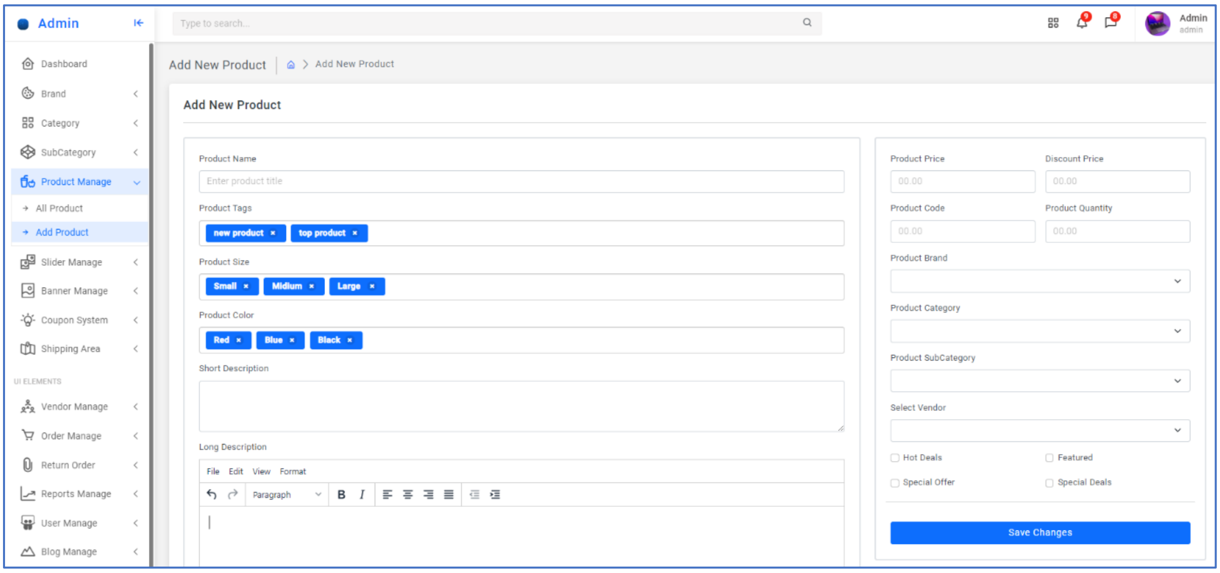
- ➢ ProductController.php

```
<?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
}
```

ក.២៣ ទំព័របង្កើតផលិតផល (Create Product Page)



Add New Product

Product Name: Enter product title

Product Tags: new product × top product ×

Product Size: Small × Midium × Large ×

Product Color: Red × Blue × Black ×

Short Description

Long Description

Product Price: 00.00 Discount Price: 00.00

Product Code: 00.00 Product Quantity: 00.00

Product Brand

Product Category

Product SubCategory

Select Vendor

□ Hot Deals □ Featured

□ Special Offer □ Special Deals

រូបភាព ក.៧៥ បង្ហាញពី Create Product Page

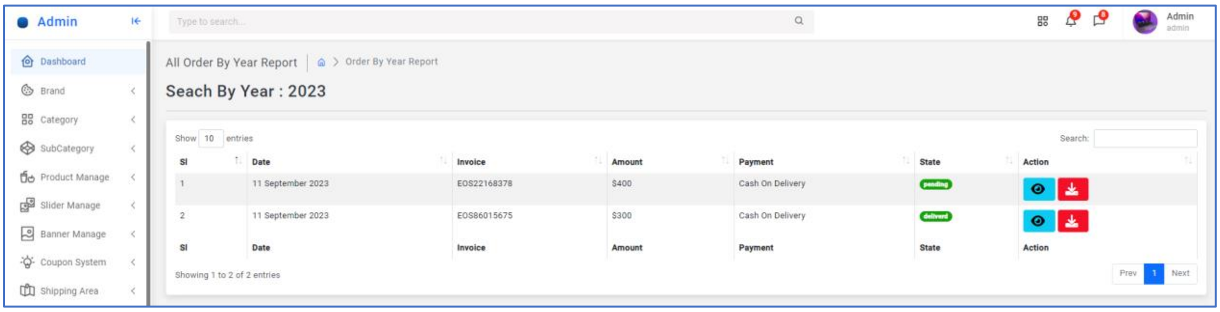
- ➢ Admin/ProductController.php

```
<?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)



All Order By Year Report > Order By Year Report

Seach By Year : 2023

Show 10 entries

Search:

Sl	Date	Invoice	Amount	Payment	State	Action
1	11 September 2023	EOS22168378	$400	Cash On Delivery	pending	
2	11 September 2023	EOS86015675	$300	Cash On Delivery	deliverd	
Sl	Date	Invoice	Amount	Payment	State	Action

Showing 1 to 2 of 2 entries

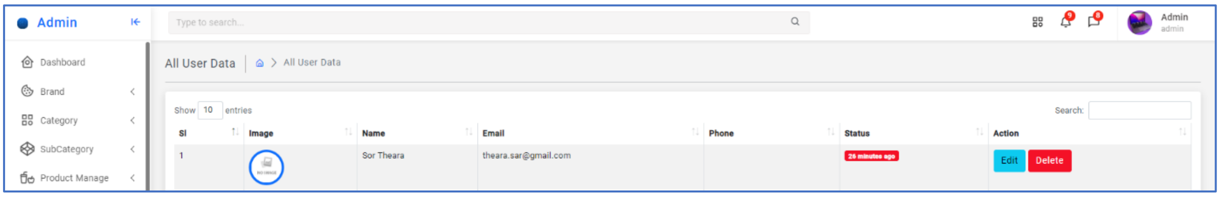
Prev 1 Next

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
}
```

ក.២៧ ទំព័របញ្ជីអតិថិជន (Customer List Page)



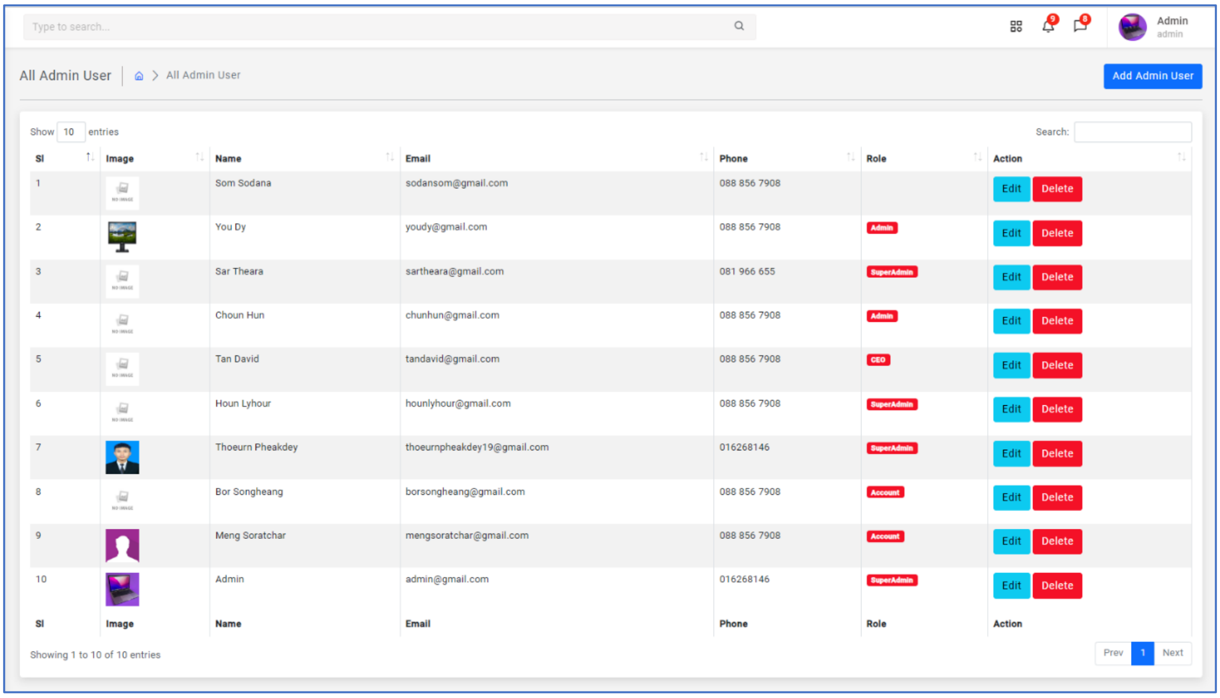
Sl	Image	Name	Email	Phone	Status	Action
1		Sor Theara	theara.sar@gmail.com		26 minutes ago	Edit Delete

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
}
```

ក.២៨ ទំព័របង្កើតសិទ្ធប្រើប្រាស់ (Create Admin User Page)



Sl	Image	Name	Email	Phone	Role	Action
1	NO IMAGE	Som Sodana	sodansom@gmail.com	088 856 7908		Edit Delete
2		You Dy	youdy@gmail.com	088 856 7908	Admin	Edit Delete
3	NO IMAGE	Sar Theara	sartheara@gmail.com	081 966 655	SuperAdmin	Edit Delete
4	NO IMAGE	Choun Hun	chunhun@gmail.com	088 856 7908	Admin	Edit Delete
5	NO IMAGE	Tan David	tandavid@gmail.com	088 856 7908	CEO	Edit Delete
6	NO IMAGE	Houn Lyhour	hounlyhour@gmail.com	088 856 7908	SuperAdmin	Edit Delete
7		Thoeurn Pheakdey	thoeurnpheakdey19@gmail.com	016268146	SuperAdmin	Edit Delete
8	NO IMAGE	Bor Songheang	borsongheang@gmail.com	088 856 7908	Account	Edit Delete
9		Meng Soratchar	mengsoratchar@gmail.com	088 856 7908	Account	Edit Delete
10		Admin	admin@gmail.com	016268146	SuperAdmin	Edit Delete
Sl	Image	Name	Email	Phone	Role	Action

Showing 1 to 10 of 10 entries

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
```

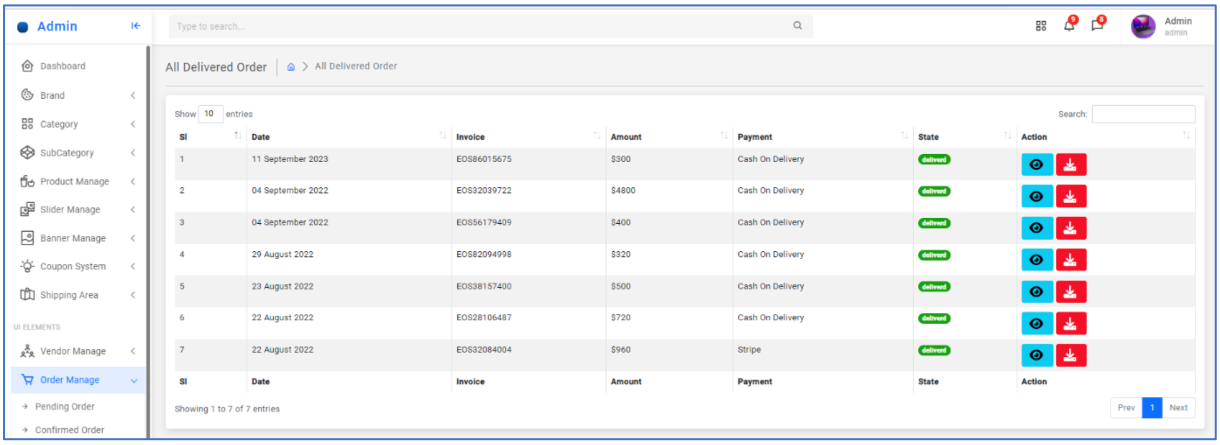
```
    public function AdminChangePassword( ){
        return view( 'admin.admin_change_password' );
    } // End Mehtod
    public function AdminUpdatePassword( Request $request ){
        // Validation
        $request->validate( [
            'old_password' => 'required',
            'new_password' => '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)



All Delivered Order | All Delivered Order

Show 10 entries Search:

Sl	Date	Invoice	Amount	Payment	State	Action
1	11 September 2023	EOS86015675	$300	Cash On Delivery	deliverd	
2	04 September 2022	EOS32039722	$4800	Cash On Delivery	deliverd	
3	04 September 2022	EOS56179409	$400	Cash On Delivery	deliverd	
4	29 August 2022	EOS82094998	$320	Cash On Delivery	deliverd	
5	23 August 2022	EOS38157400	$500	Cash On Delivery	deliverd	
6	22 August 2022	EOS28106487	$720	Cash On Delivery	deliverd	
7	22 August 2022	EOS32084004	$960	Stripe	deliverd	
Sl	Date	Invoice	Amount	Payment	State	Action

Showing 1 to 7 of 7 entries

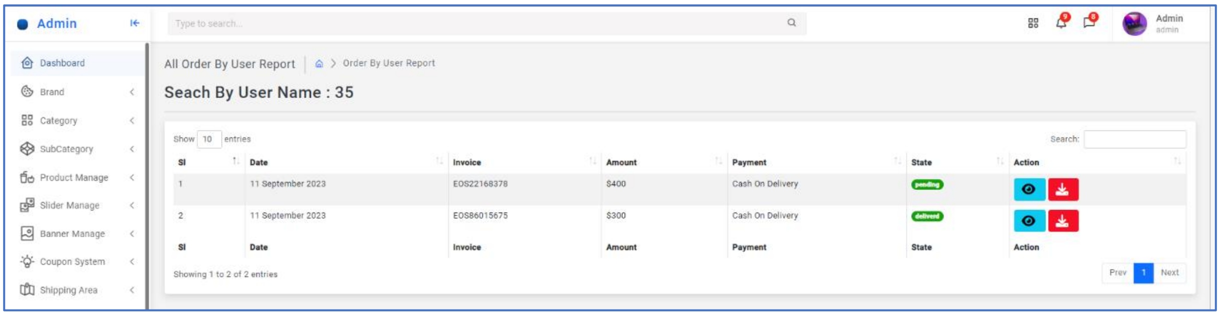
រូបភាព ក.៧៩ បង្ហាញពី Confirm Order Page

- ➢ 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
}
```

ក.៣៣ ទំព័របញ្ជីបញ្ជាទិញដោយអ្នកប្រើប្រាស់ (Order List Page)



All Order By User Report > Order By User Report

Seach By User Name : 35

Show 10 entries Search:

Sl	Date	Invoice	Amount	Payment	State	Action
1	11 September 2023	EOS22168378	$400	Cash On Delivery	pending	
2	11 September 2023	EOS86015675	$300	Cash On Delivery	deliverd	
Sl	Date	Invoice	Amount	Payment	State	Action

Showing 1 to 2 of 2 entries

Prev 1 Next

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 }
```