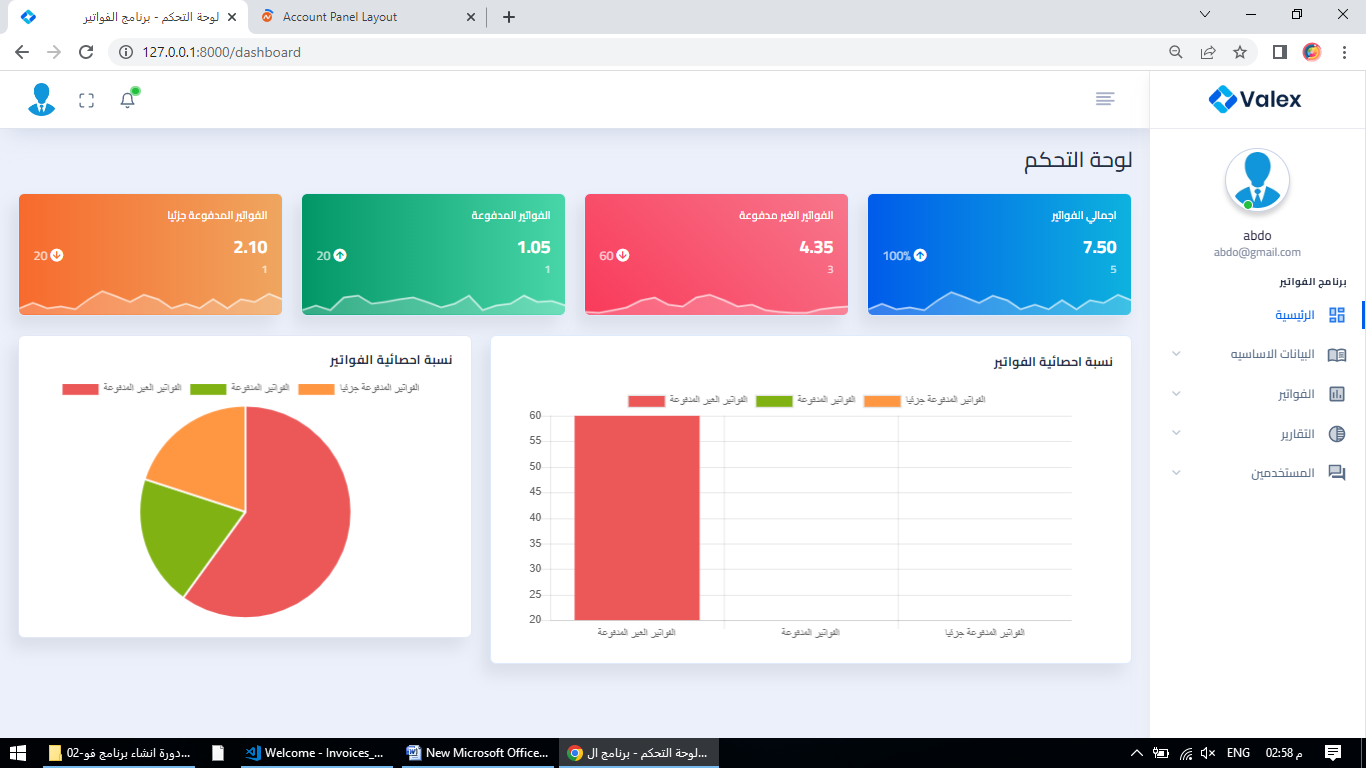
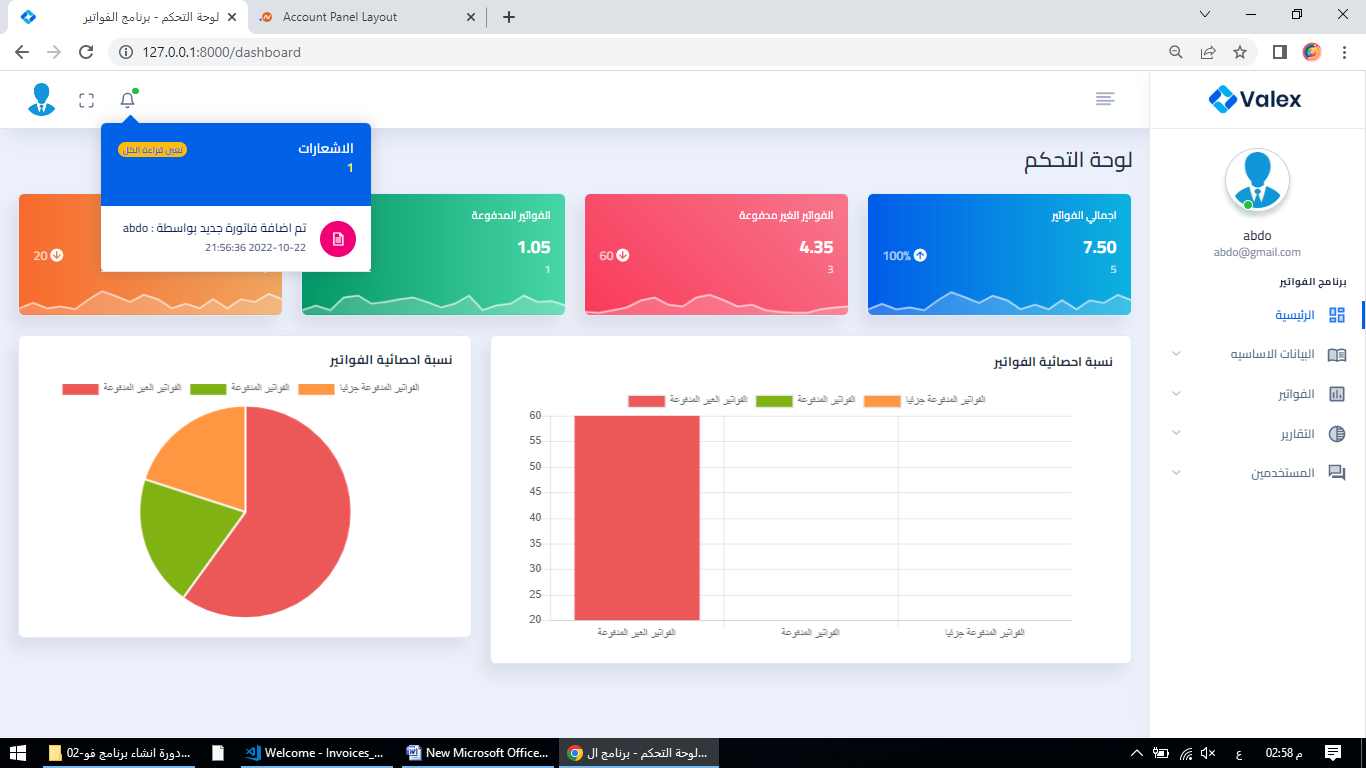
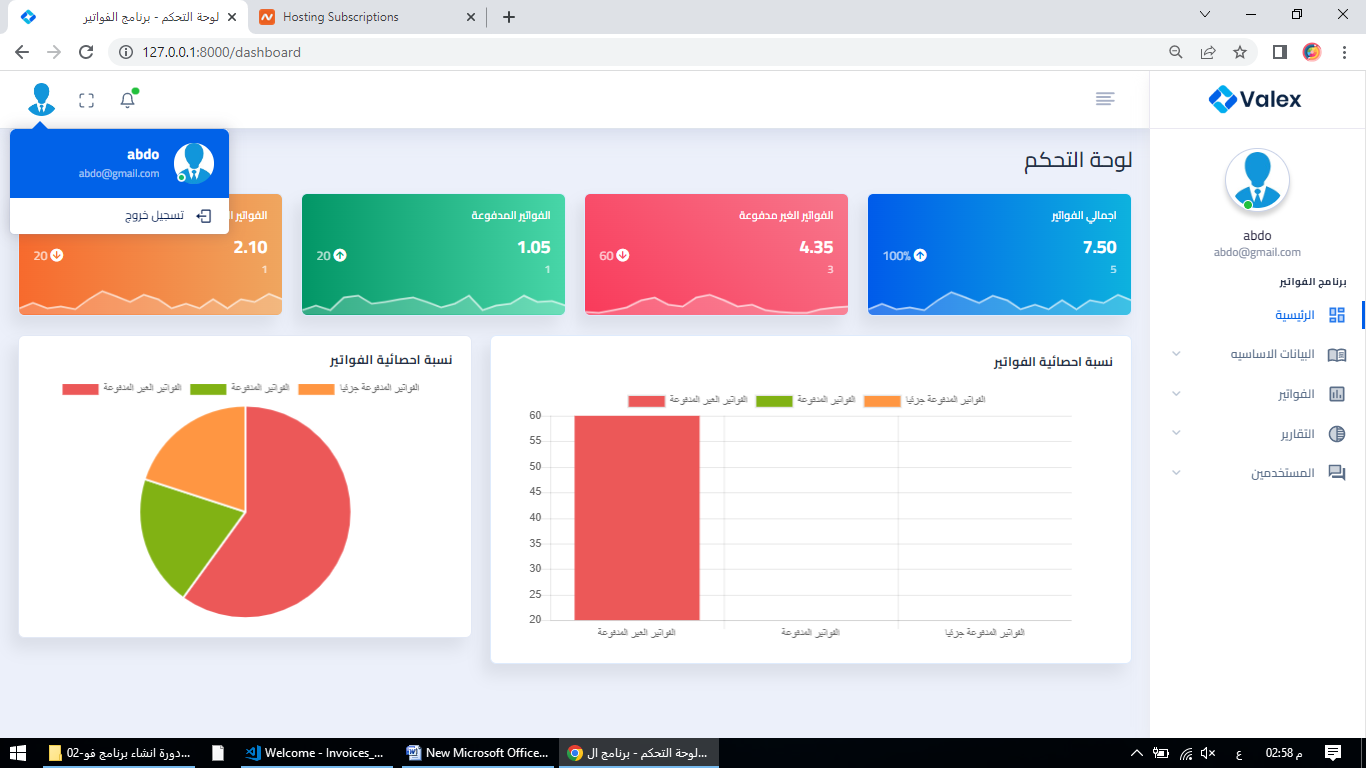
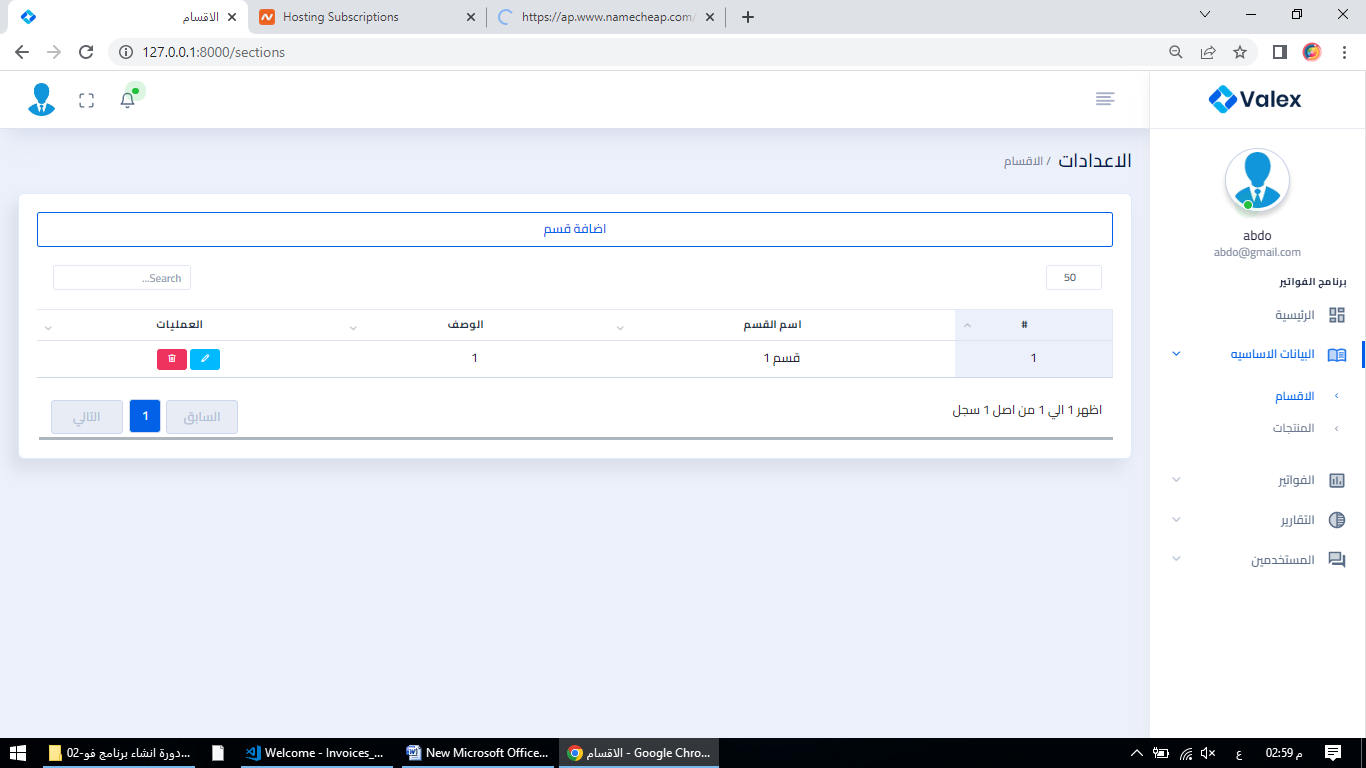
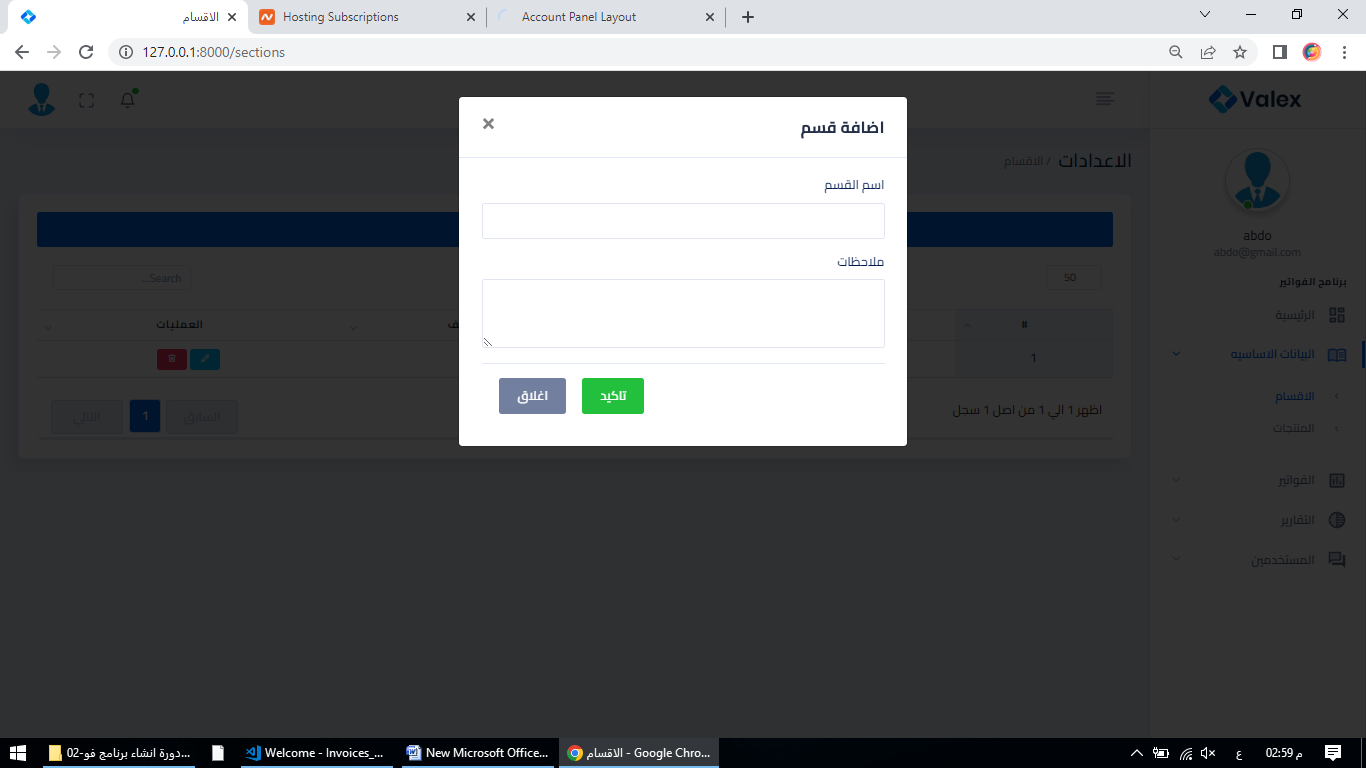
<https://invoices-project.brothersmedia.site/>

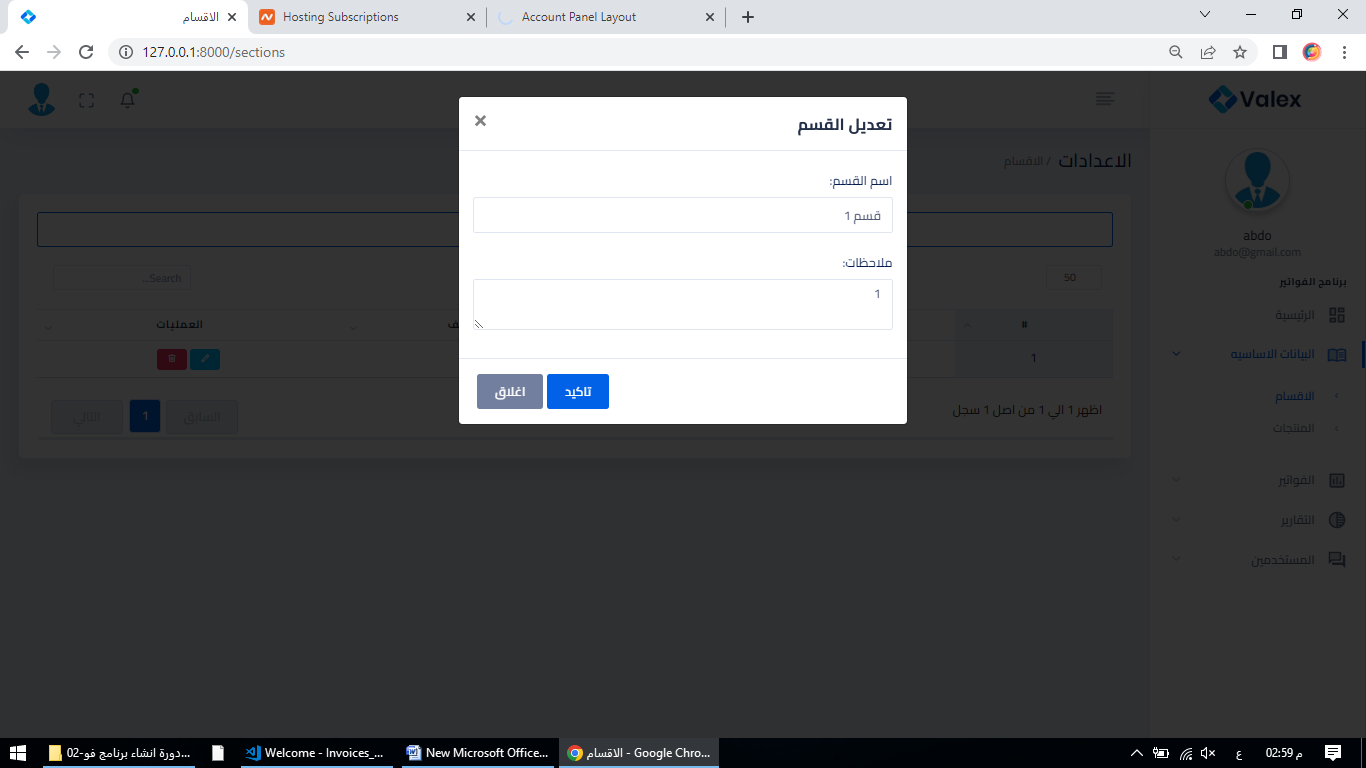


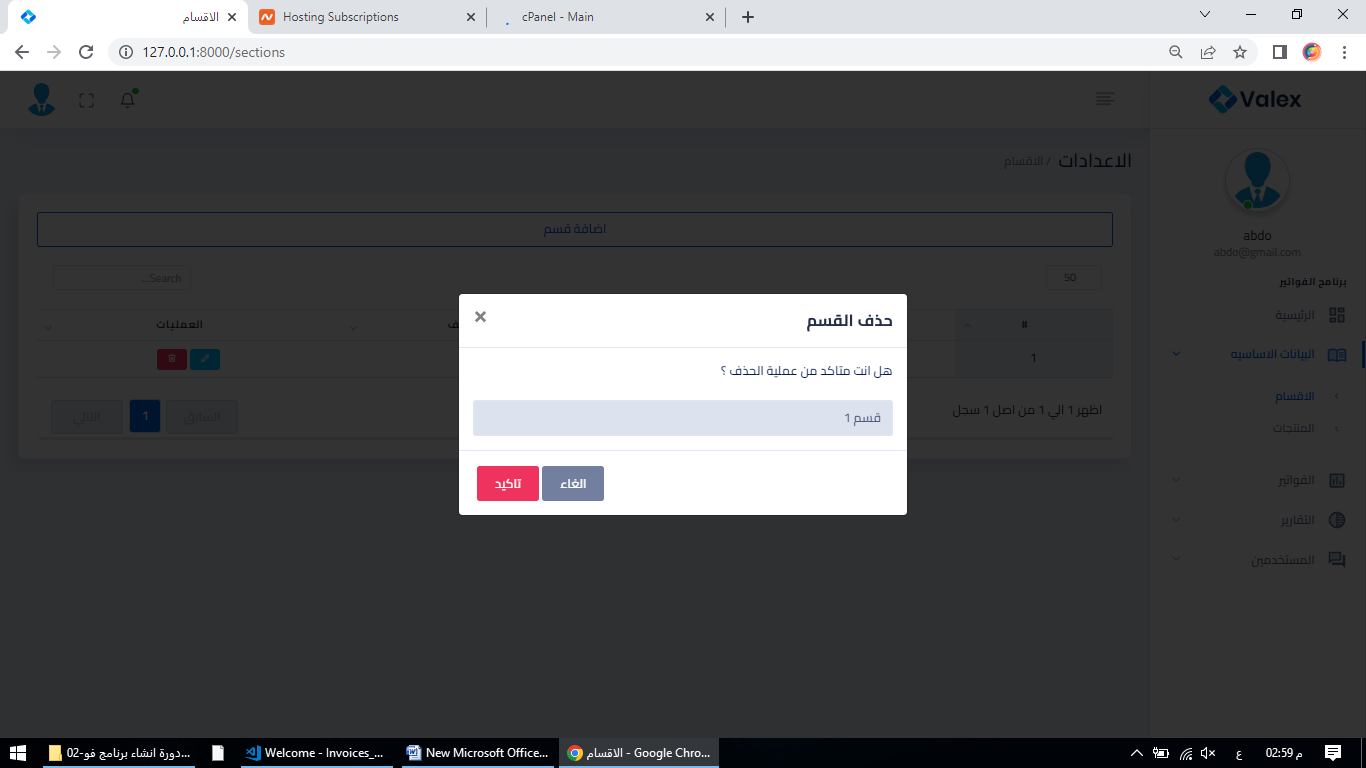


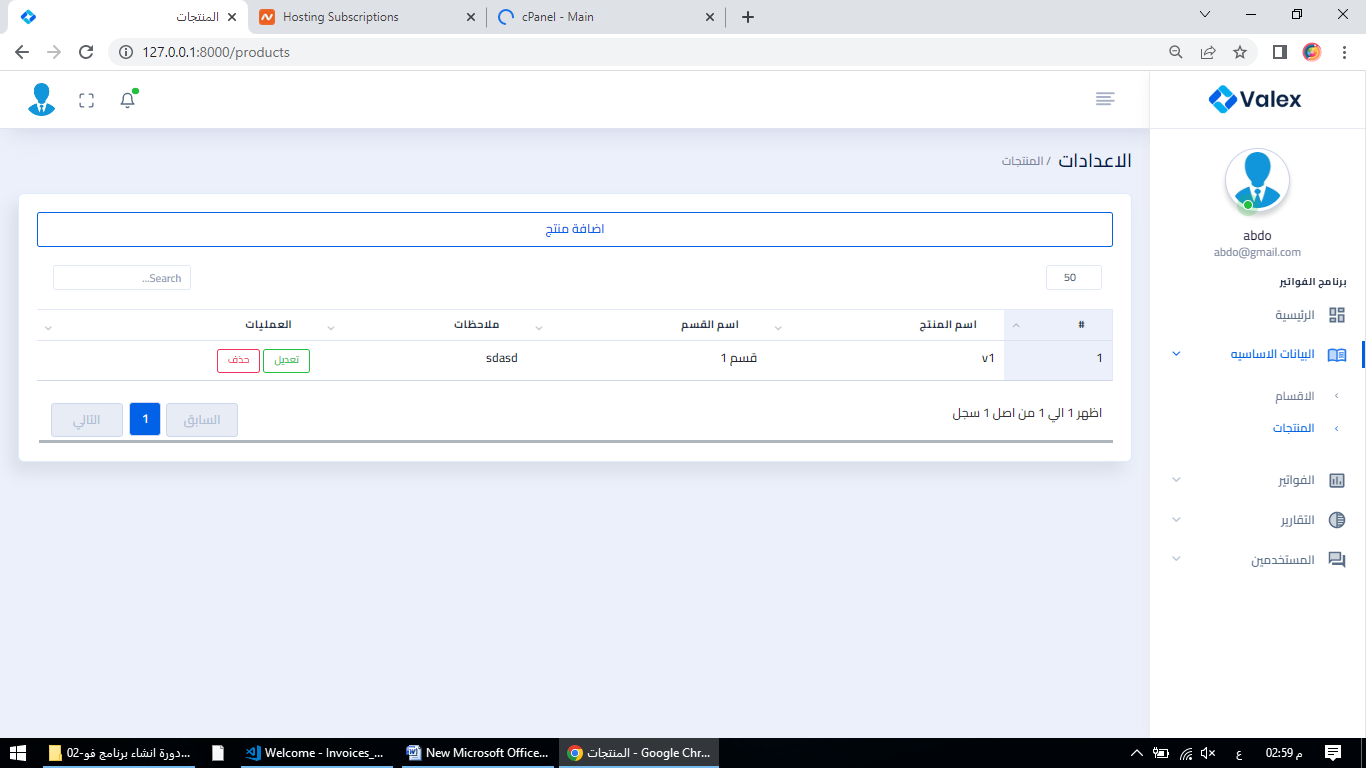


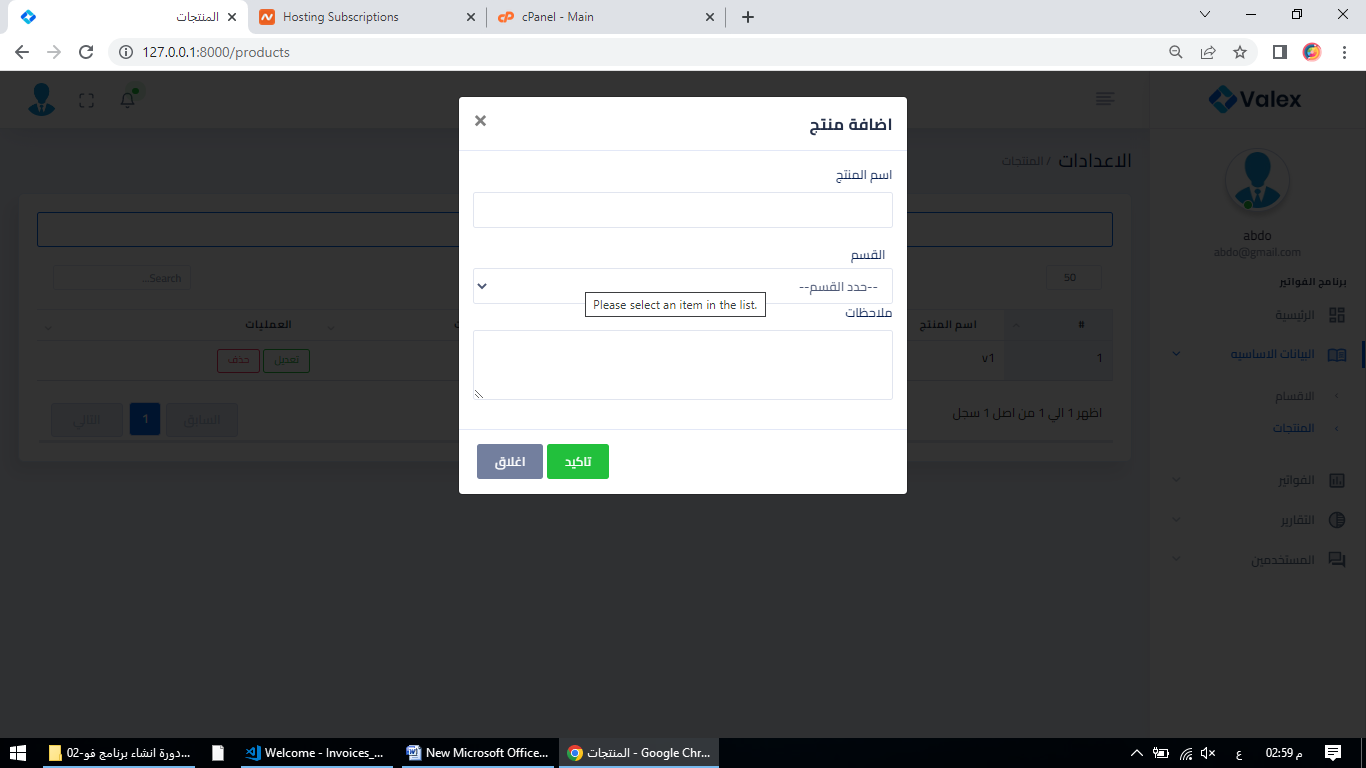


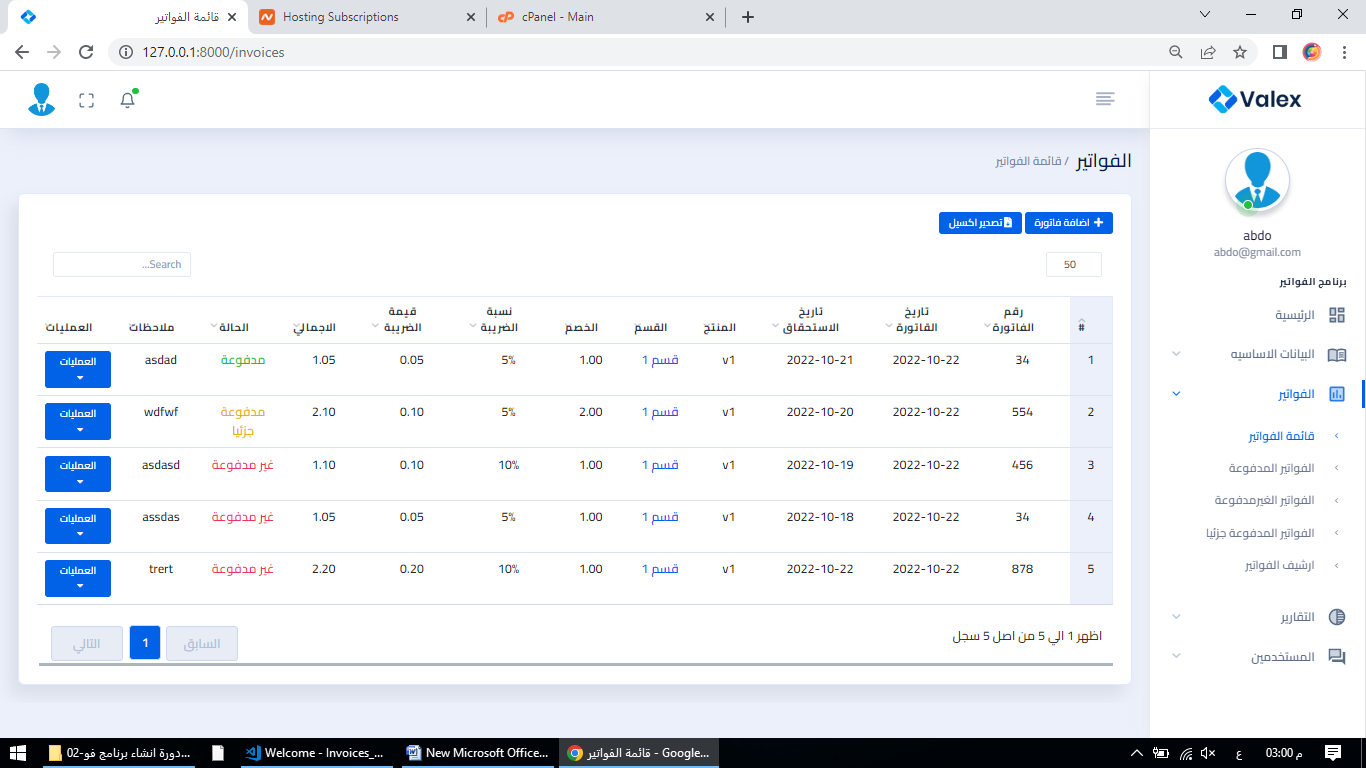


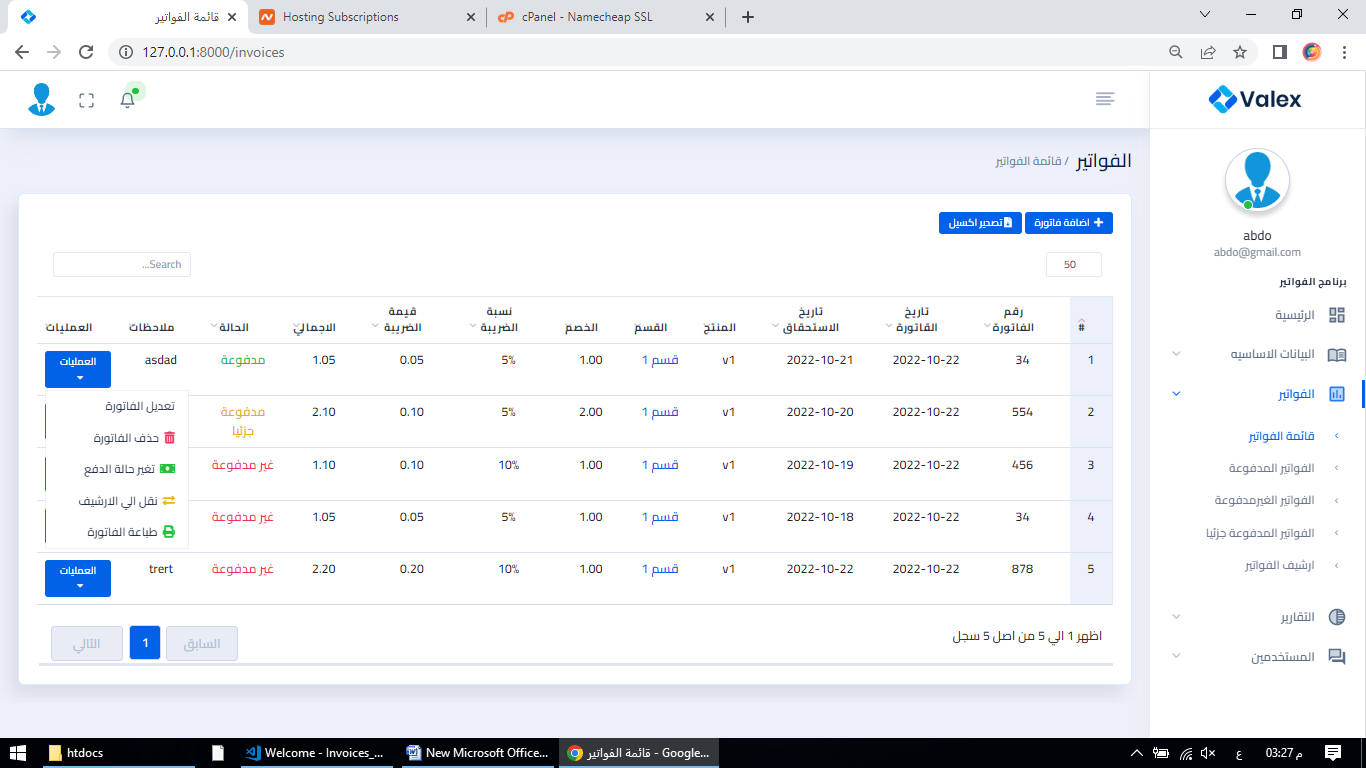


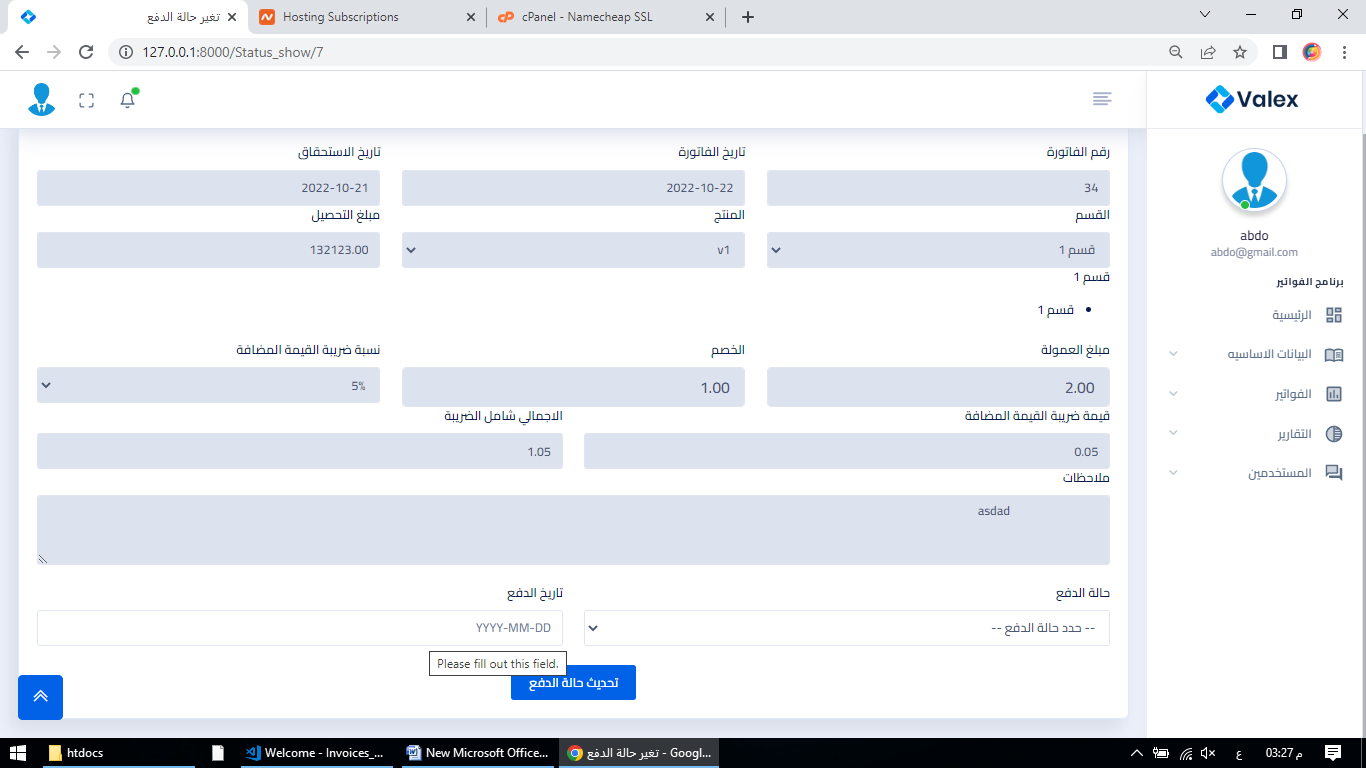


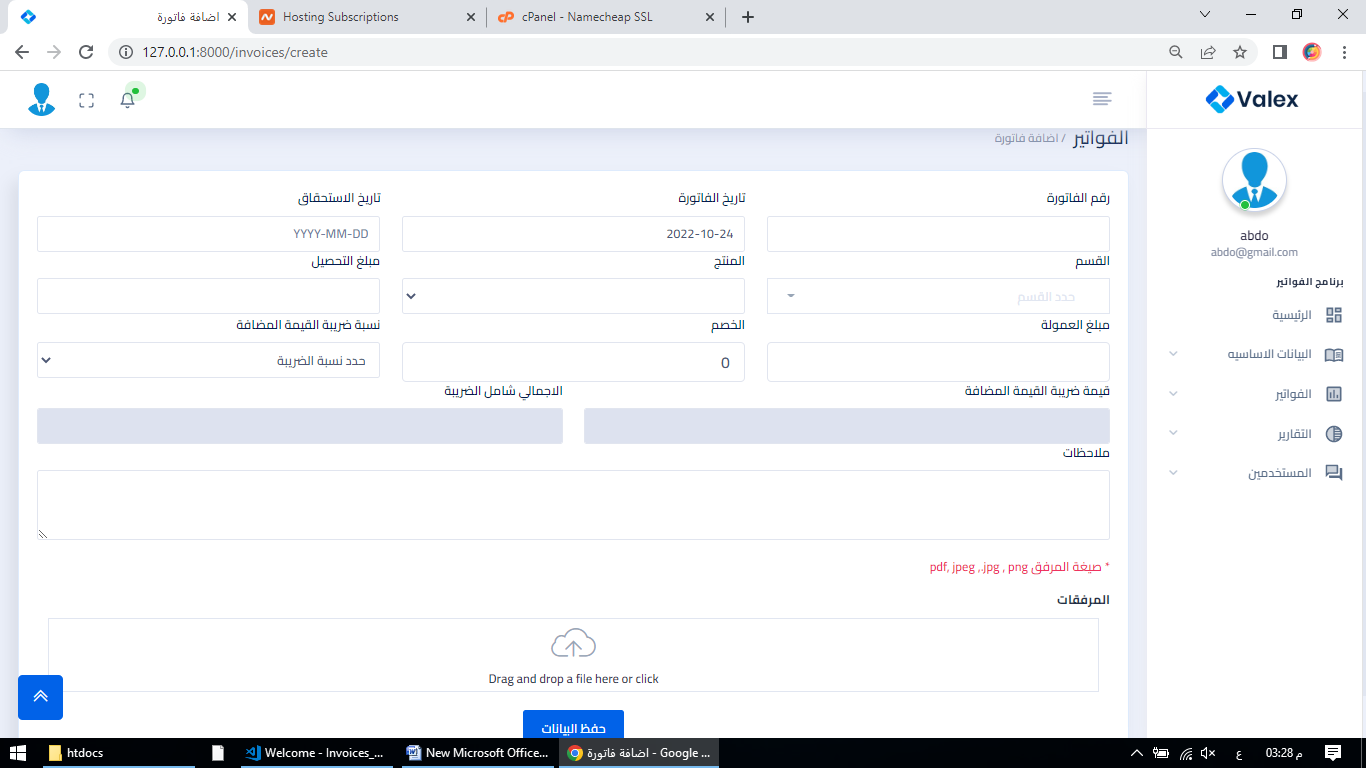


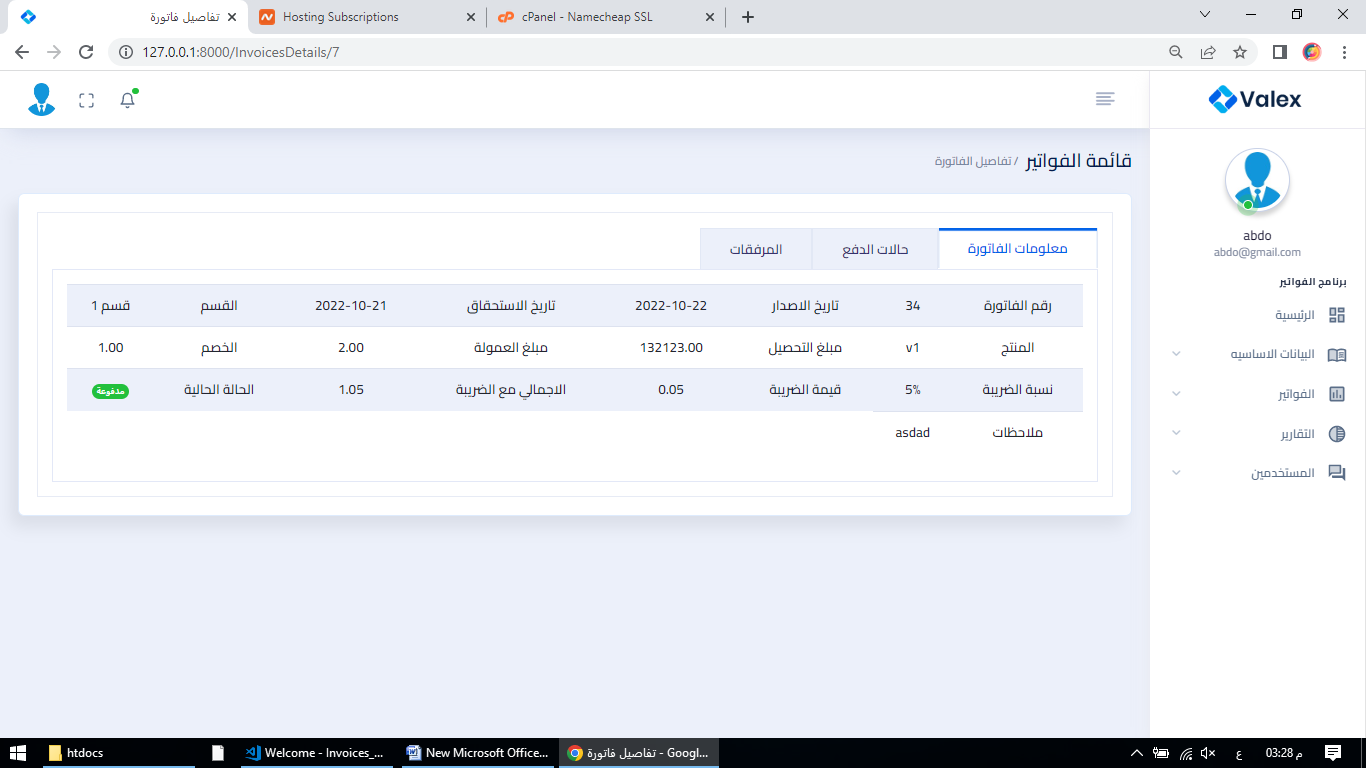


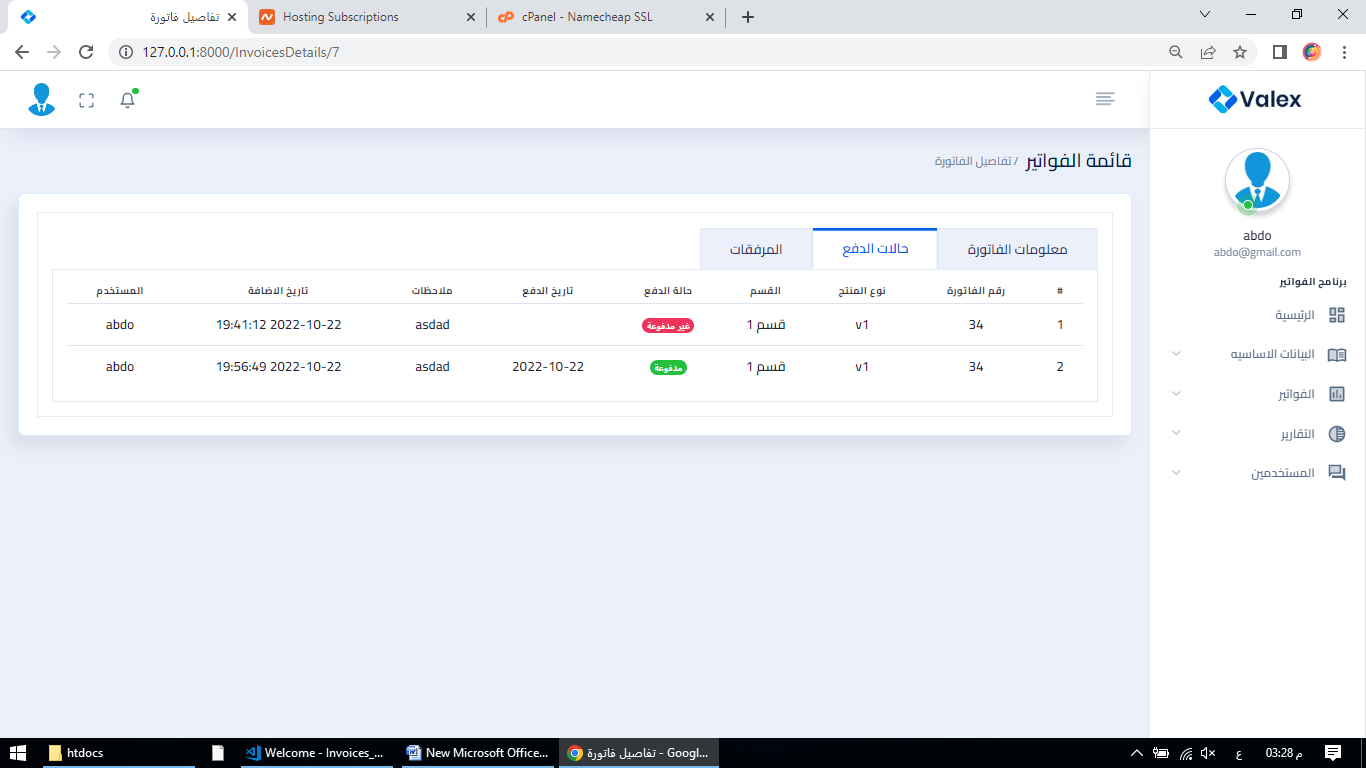


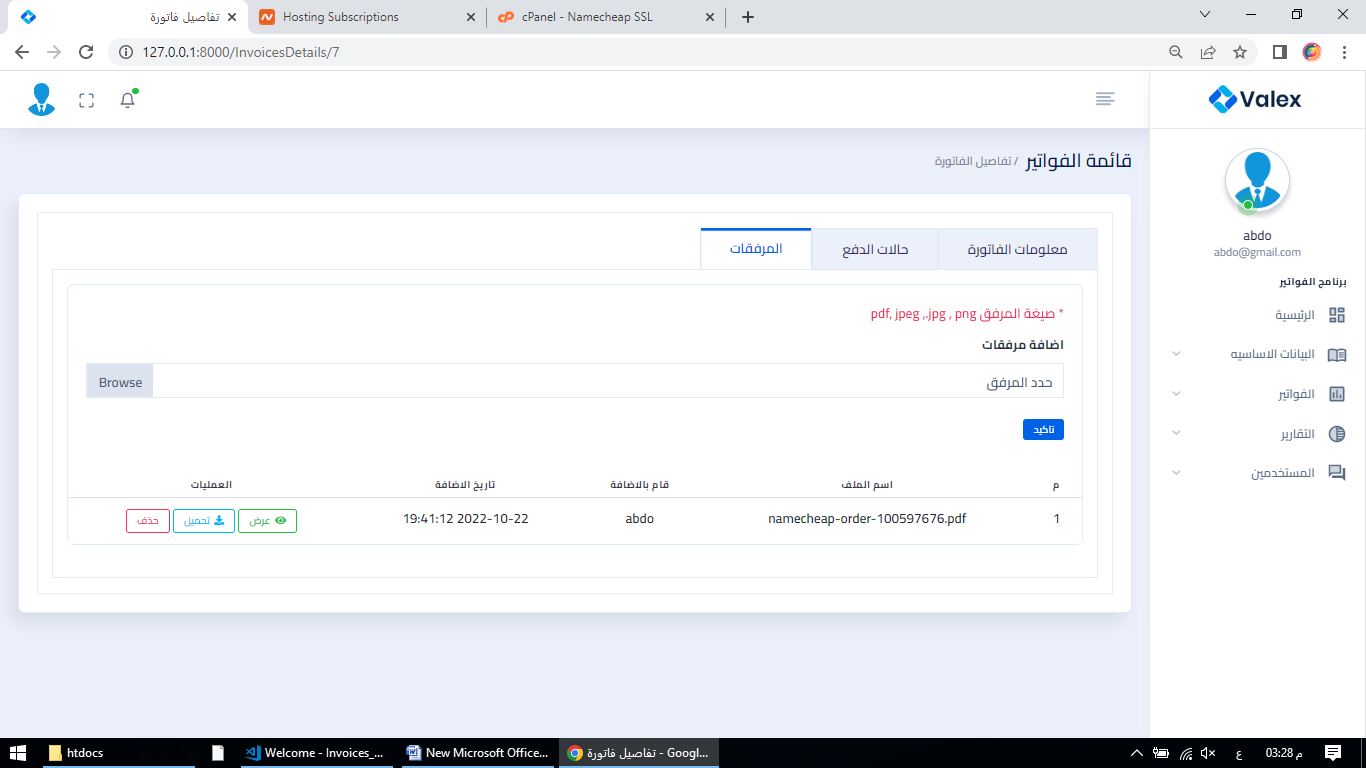


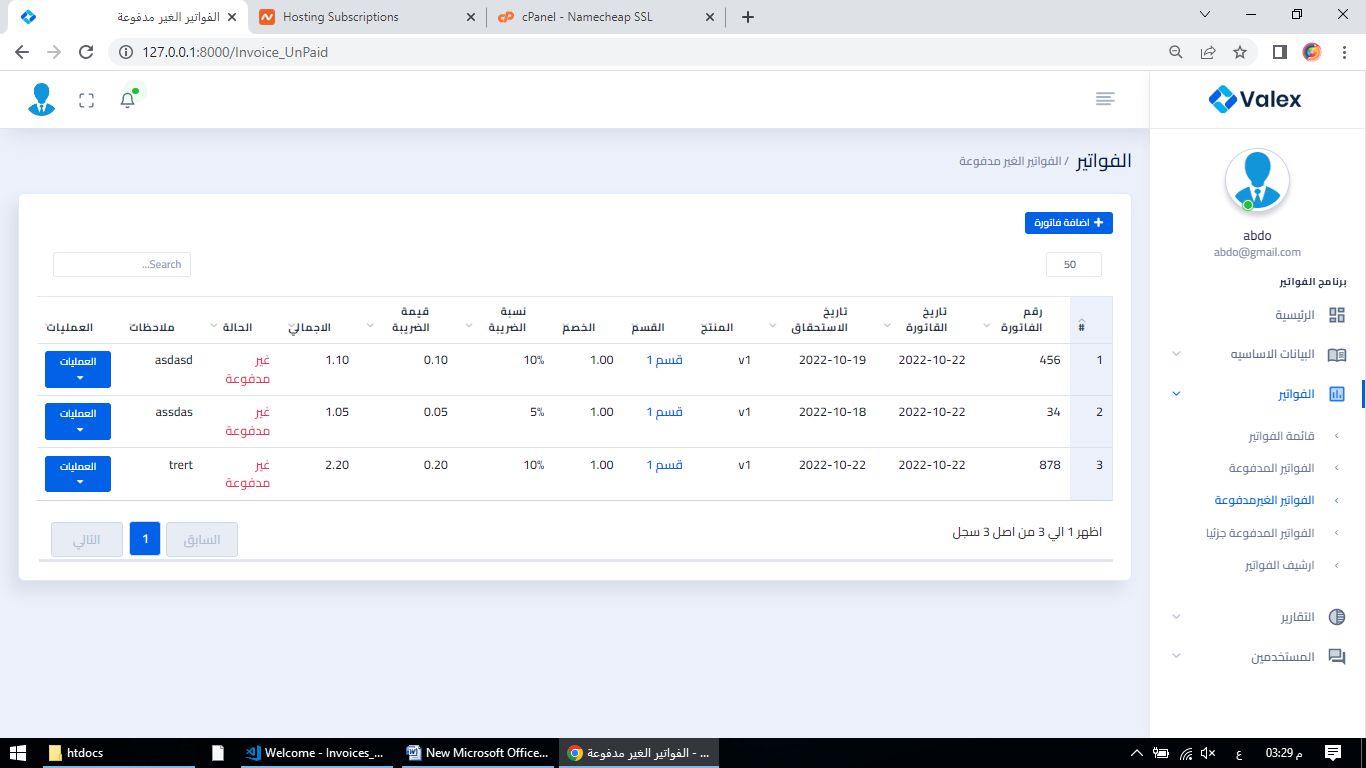


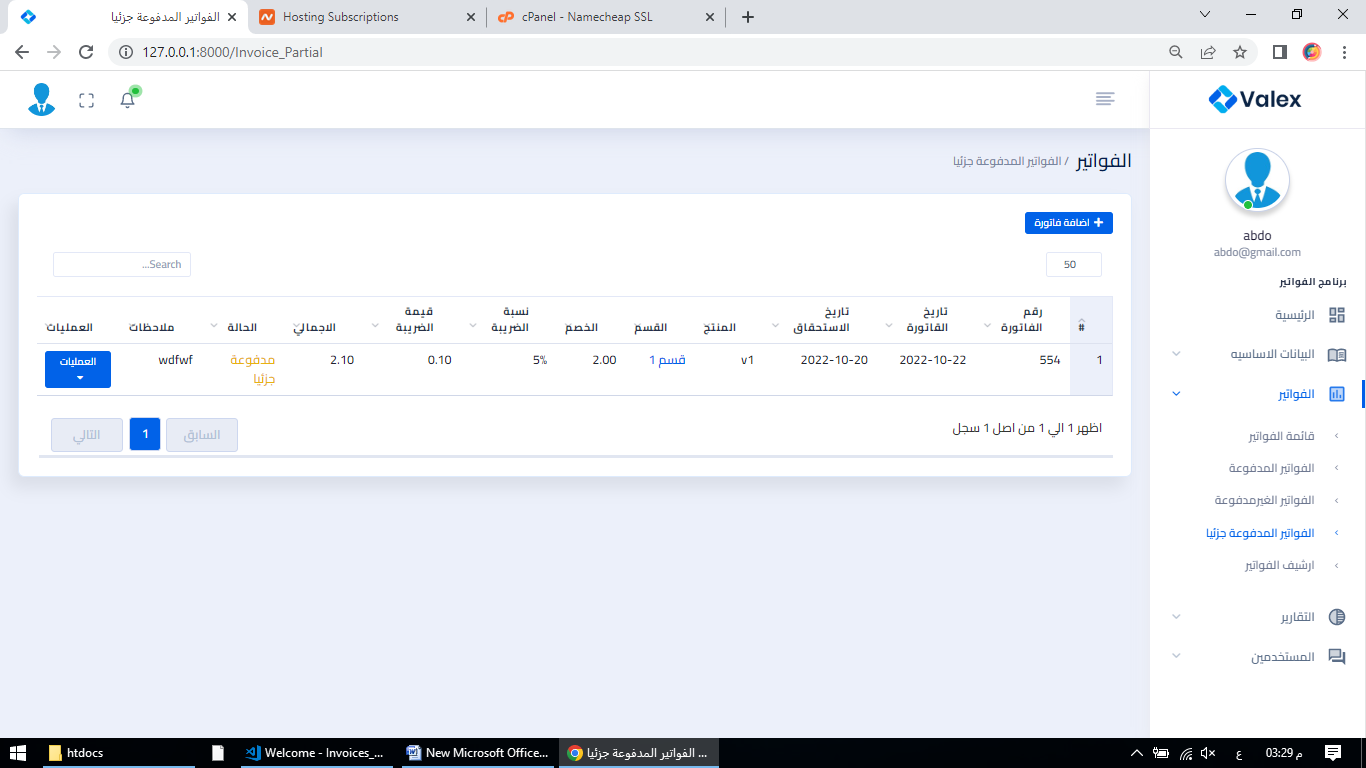


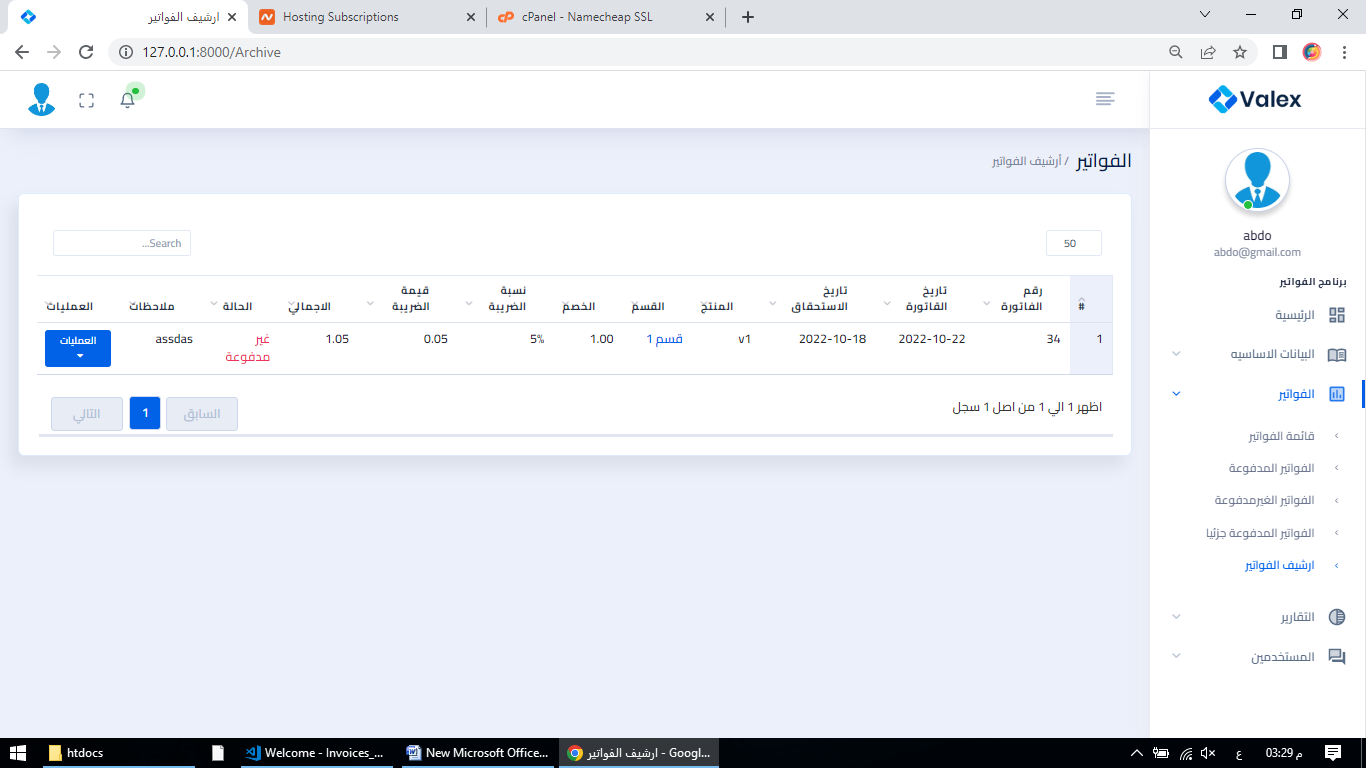


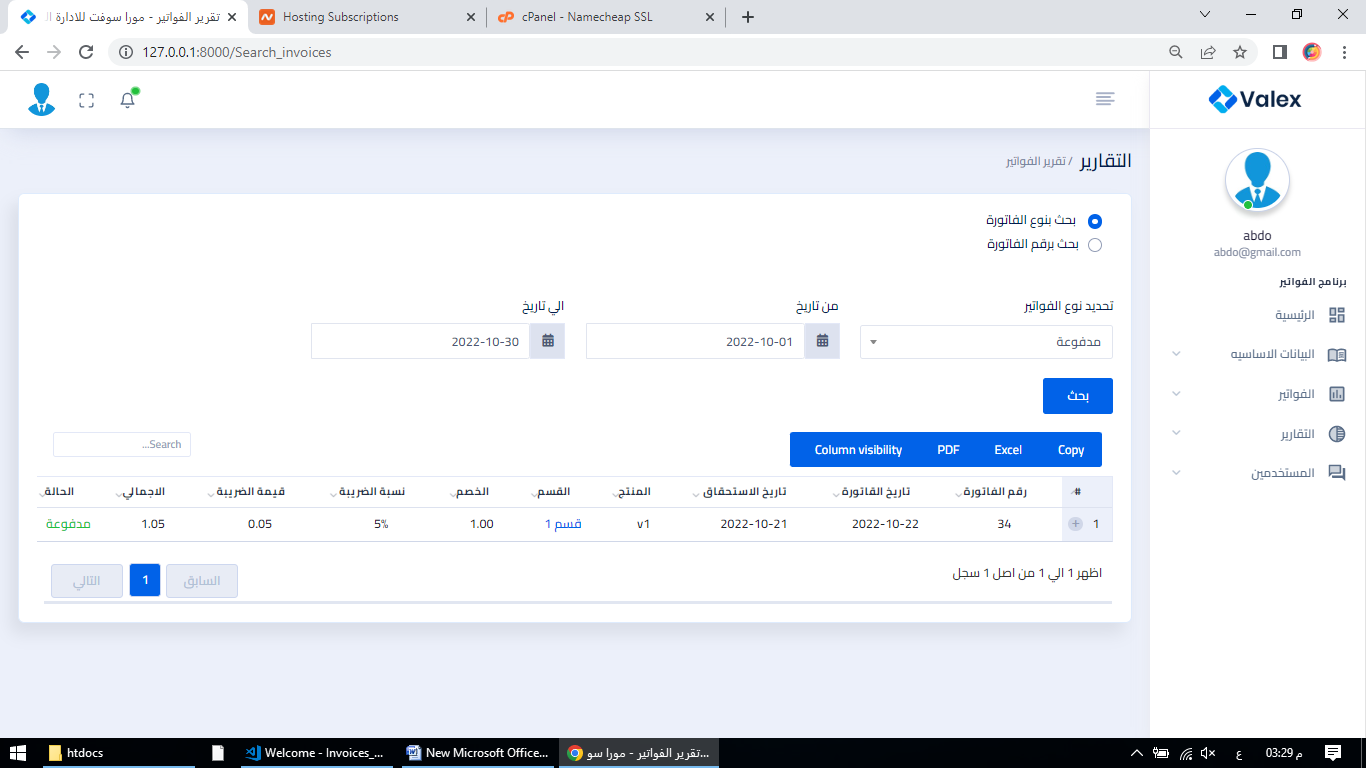


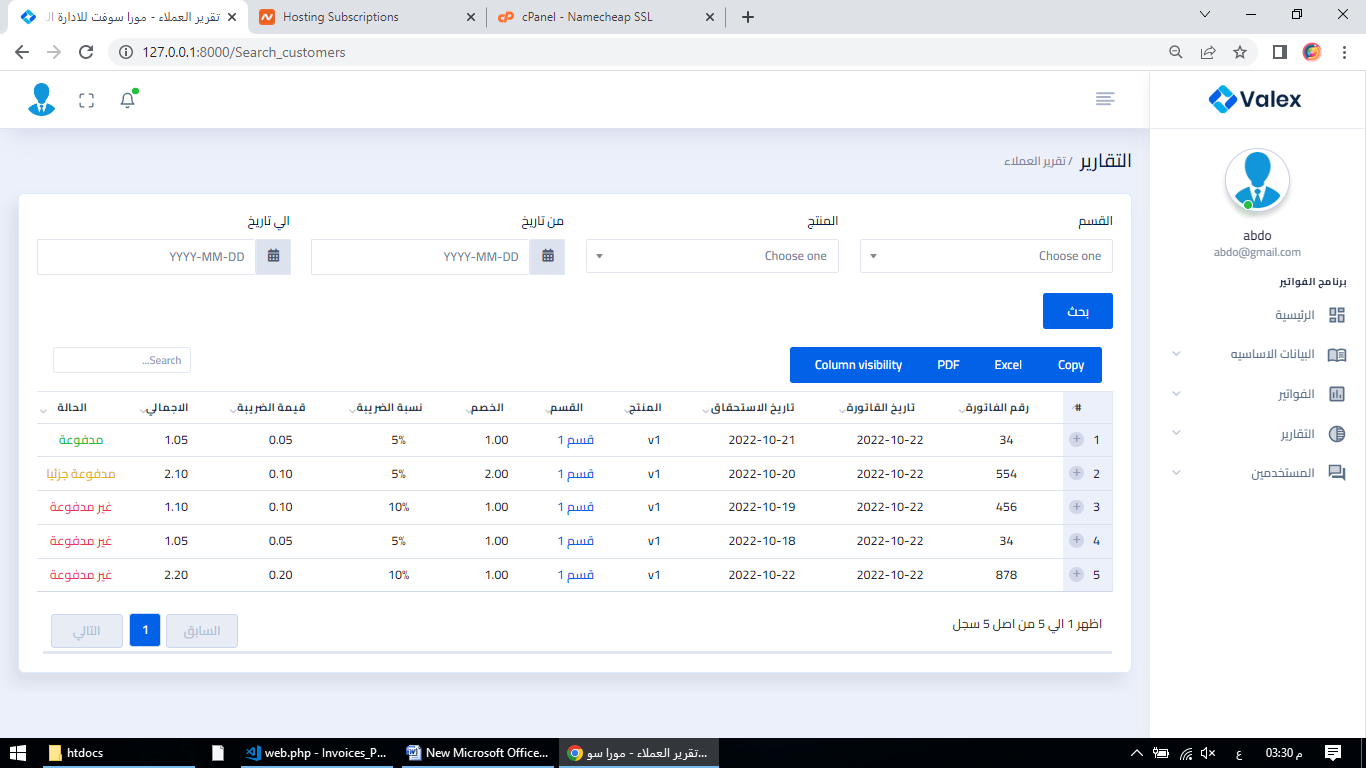


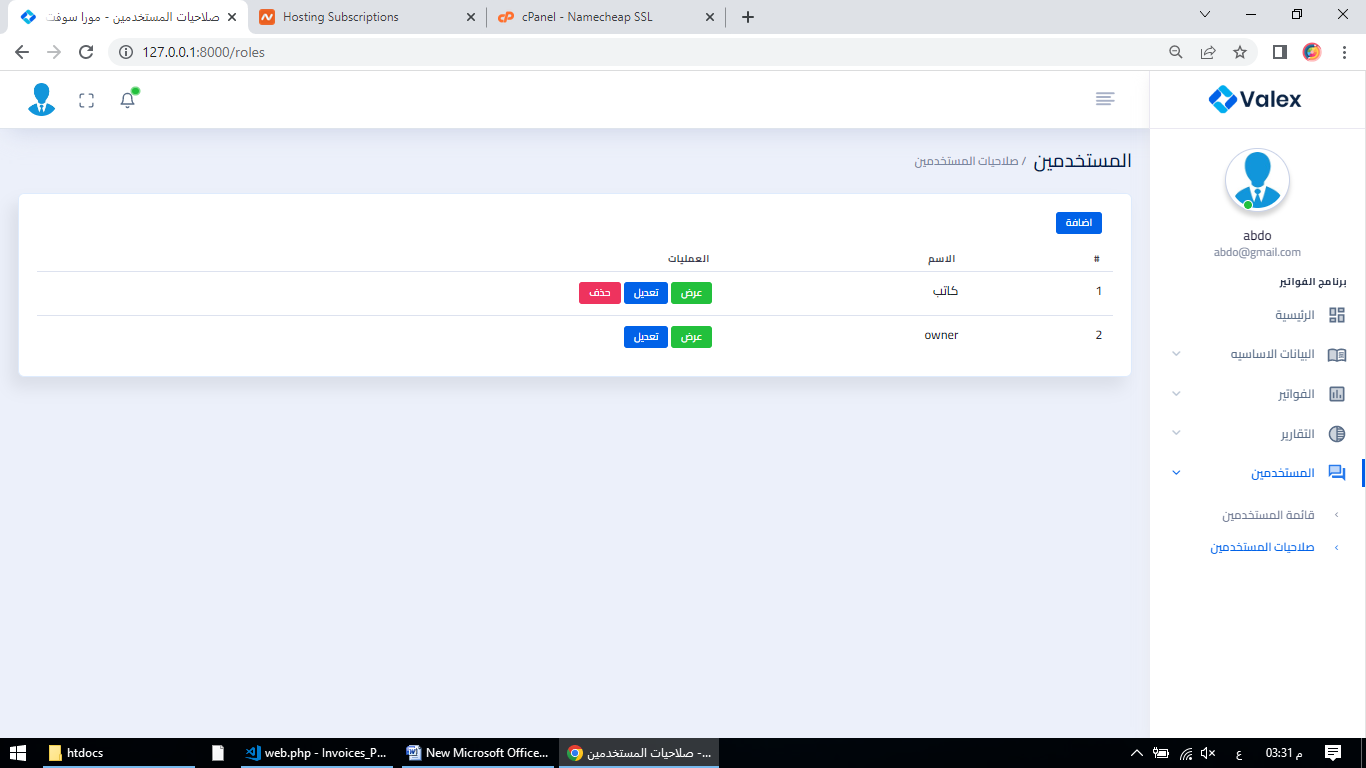


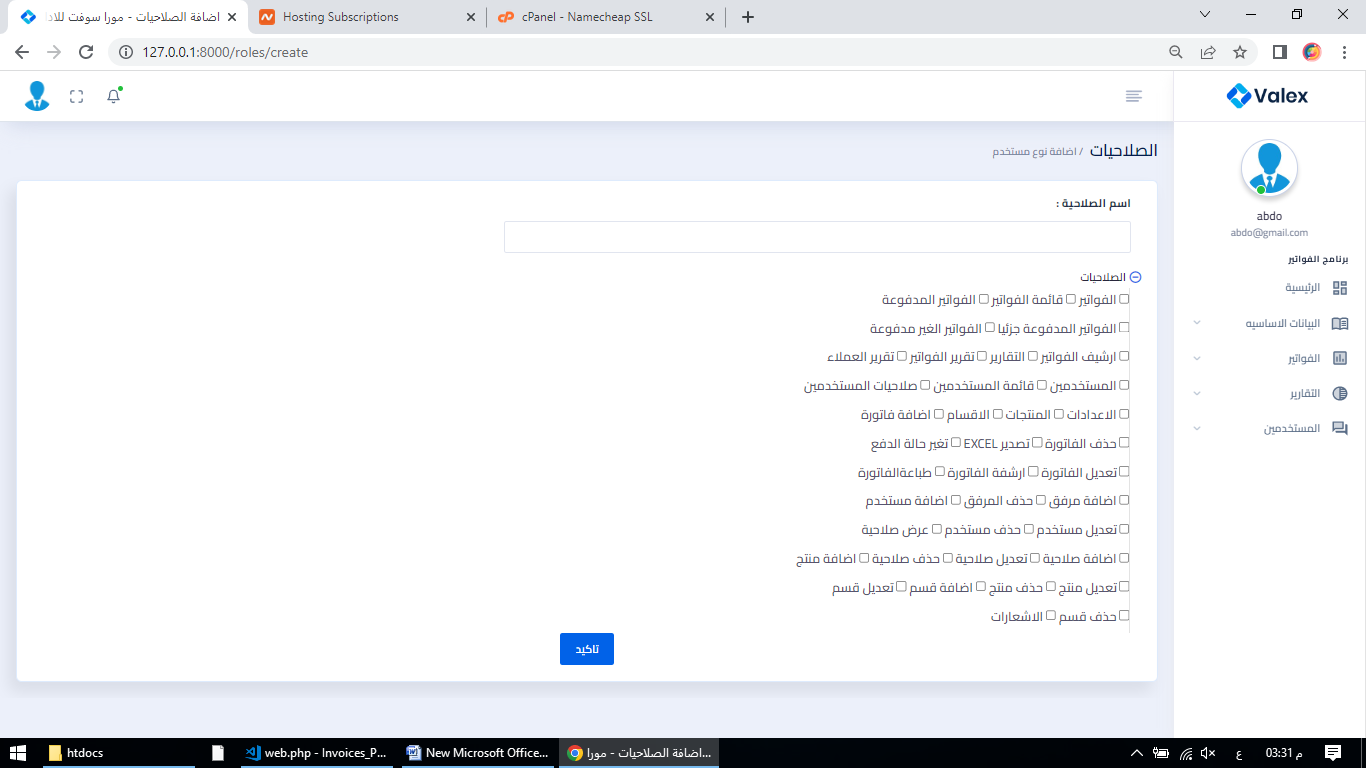


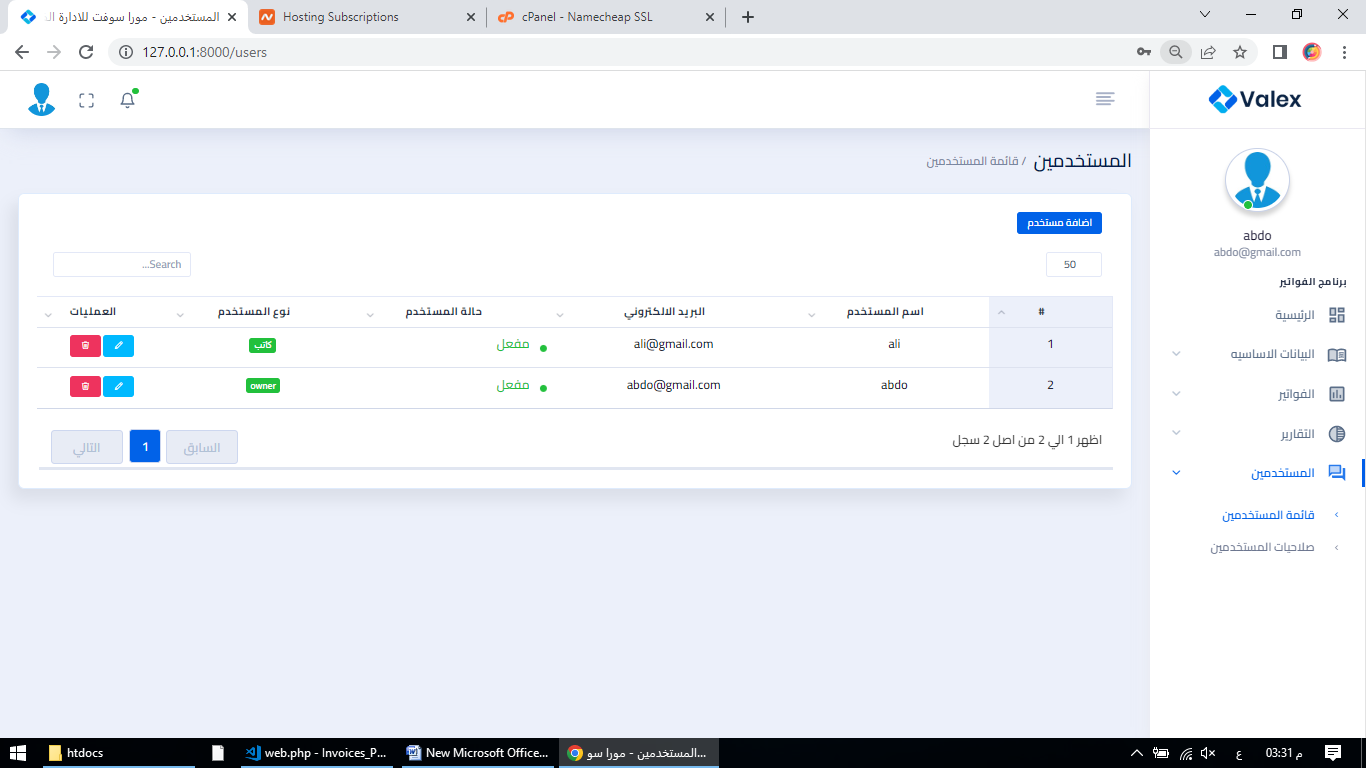


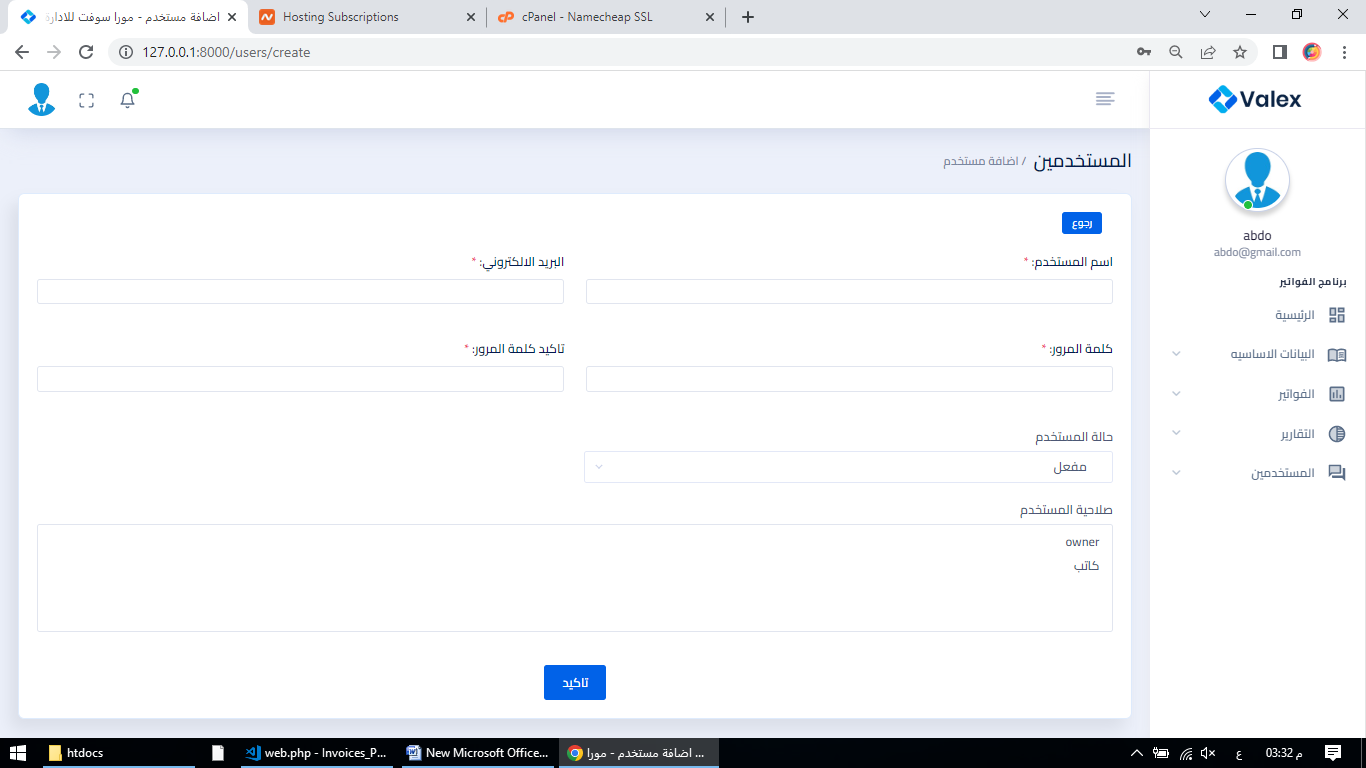












DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=invoices\_project

DB\_USERNAME=root

DB\_PASSWORD=

php artisan make:model section -m

Model created successfully.

Created Migration: 2022\_10\_21\_151503\_create\_sections\_table

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:model invoice -m

Model created successfully.

Created Migration: 2022\_10\_21\_151706\_create\_invoices\_table

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:model product -m

Model created successfully.

Created Migration: 2022\_10\_21\_151821\_create\_products\_table

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:model invoices\_detail -m

Model created successfully.

Created Migration: 2022\_10\_21\_151842\_create\_invoices\_details\_table

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:model invoice\_attachment -m

Model created successfully.

Created Migration: 2022\_10\_21\_151911\_create\_invoice\_attachments\_table

public function up()

{

Schema::create('sections', function (Blueprint $table) {

$table->bigIncrements('id');

$table->string('section\_name', 999);

$table->text('description')->nullable();

$table->string('Created\_by', 999);

$table->timestamps();

});

}

public function up()

{

Schema::create('invoices', function (Blueprint $table) {

$table->bigIncrements('id');

$table->string('invoice\_number', 50);

$table->date('invoice\_Date')->nullable();

$table->date('Due\_date')->nullable();

$table->string('product', 50);

$table->bigInteger( 'section\_id' )->unsigned();

$table->foreign('section\_id')->references('id')->on('sections')->onDelete('cascade');

$table->decimal('Amount\_collection',8,2)->nullable();;

$table->decimal('Amount\_Commission',8,2);

$table->decimal('Discount',8,2);

$table->decimal('Value\_VAT',8,2);

$table->string('Rate\_VAT', 999);

$table->decimal('Total',8,2);

$table->string('Status', 50);

$table->integer('Value\_Status');

$table->text('note')->nullable();

$table->date('Payment\_Date')->nullable();

$table->softDeletes();

$table->timestamps();

});

}

public function up()

{

Schema::create('products', function (Blueprint $table) {

$table->id();

$table->string('Product\_name', 999);

$table->text('description')->nullable();

$table->unsignedBigInteger('section\_id');

$table->foreign('section\_id')->references('id')->on('sections')->onDelete('cascade');

$table->timestamps();

});

}

public function up()

{

Schema::create('invoices\_details', function (Blueprint $table) {

$table->id();

$table->unsignedBigInteger('id\_Invoice');

$table->string('invoice\_number', 50);

$table->foreign('id\_Invoice')->references('id')->on('invoices')->onDelete('cascade');

$table->string('product', 50);

$table->string('Section', 999);

$table->string('Status', 50);

$table->integer('Value\_Status');

$table->date('Payment\_Date')->nullable();

$table->text('note')->nullable();

$table->string('user',300);

$table->timestamps();

});

}

public function up()

{

Schema::create('invoice\_attachments', function (Blueprint $table) {

$table->id();

$table->string('file\_name', 999);

$table->string('invoice\_number', 50);

$table->string('Created\_by', 999);

$table->unsignedBigInteger('invoice\_id')->nullable();

$table->foreign('invoice\_id')->references('id')->on('invoices')->onDelete('cascade');

$table->timestamps();

});

}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

use Illuminate\Database\Eloquent\SoftDeletes;

class invoices extends Model

{

use HasFactory;

use SoftDeletes;

protected $fillable = [

'invoice\_number',

'invoice\_Date',

'Due\_date',

'product',

'section\_id',

'Amount\_collection',

'Amount\_Commission',

'Discount',

'Value\_VAT',

'Rate\_VAT',

'Total',

'Status',

'Value\_Status',

'note',

'Payment\_Date',

];

protected $dates = ['deleted\_at'];

public function section()

{

return $this->belongsTo(App\Models\sections');

}

}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class invoices\_details extends Model

{

use HasFactory;

protected $fillable = [

'id\_Invoice',

'invoice\_number',

'product',

'Section',

'Status',

'Value\_Status',

'note',

'user',

'Payment\_Date',

];

}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class products extends Model

{

use HasFactory;

protected $guarded = [];

public function section()

{

return $this->belongsTo('App\Models\sections');

}

}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class sections extends Model

{

use HasFactory;

protected $fillable = [

'section\_name',

'description',

'Created\_by',

];

}

<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

use Spatie\Permission\Traits\HasRoles;

class User extends Authenticatable

{

use HasFactory, Notifiable;

use HasRoles;

protected $fillable = [

'name', 'email', 'password','roles\_name','Status'

];

protected $hidden = [

'password',

'remember\_token',

];

protected $casts = [

'email\_verified\_at' => 'datetime',

'roles\_name' => 'array',

];

}

<https://www.itsolutionstuff.com/post/laravel-6-user-roles-and-permissions-from-scratch-laravel-6-aclexample.html>

composer require spatie/laravel-permission

composer require laravelcollective/html

app.php

Spatie\Permission\PermissionServiceProvider::class,

Collective\Html\HtmlServiceProvider::class,

'Form' => Collective\Html\FormFacade::class,

'Html' => Collective\Html\HtmlFacade::class,

php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider"

php artisan migrate

Kernel.php

'role' => \Spatie\Permission\Middlewares\RoleMiddleware::class,

'permission' => \Spatie\Permission\Middlewares\PermissionMiddleware::class,

php artisan make:seeder PermissionTableSeeder

Seeder created successfully.

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

use Spatie\Permission\Models\Permission;

class PermissionTableSeeder extends Seeder

{

public function run()

{

$permissions = [

'الفواتير',

'قائمة الفواتير',

'الفواتير المدفوعة',

'الفواتير المدفوعة جزئيا',

'الفواتير الغير مدفوعة',

'ارشيف الفواتير',

'التقارير',

'تقرير الفواتير',

'تقرير العملاء',

'المستخدمين',

'قائمة المستخدمين',

'صلاحيات المستخدمين',

'الاعدادات',

'المنتجات',

'الاقسام',

'اضافة فاتورة',

'حذف الفاتورة',

'تصدير EXCEL',

'تغير حالة الدفع',

'تعديل الفاتورة',

'ارشفة الفاتورة',

'طباعةالفاتورة',

'اضافة مرفق',

'حذف المرفق',

'اضافة مستخدم',

'تعديل مستخدم',

'حذف مستخدم',

'عرض صلاحية',

'اضافة صلاحية',

'تعديل صلاحية',

'حذف صلاحية',

'اضافة منتج',

'تعديل منتج',

'حذف منتج',

'اضافة قسم',

'تعديل قسم',

'حذف قسم',

'الاشعارات',

];

foreach ($permissions as $permission) {

Permission::create(['name' => $permission]);

}

}

}

php artisan make:seeder CreateAdminUserSeeder

Seeder created successfully.

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

use App\Models\User;

use Spatie\Permission\Models\Role;

use Spatie\Permission\Models\Permission;

class CreateAdminUserSeeder extends Seeder

{

public function run()

{

$user = User::create([

'name' => 'abdo',

'email' => 'abdo@gmail.com',

'password' => bcrypt('Abdo1379'),

'roles\_name' => ["owner"],

'Status' => 'مفعل',

]);

$role = Role::create(['name' => 'owner']);

$permissions = Permission::pluck('id','id')->all();

$role->syncPermissions($permissions);

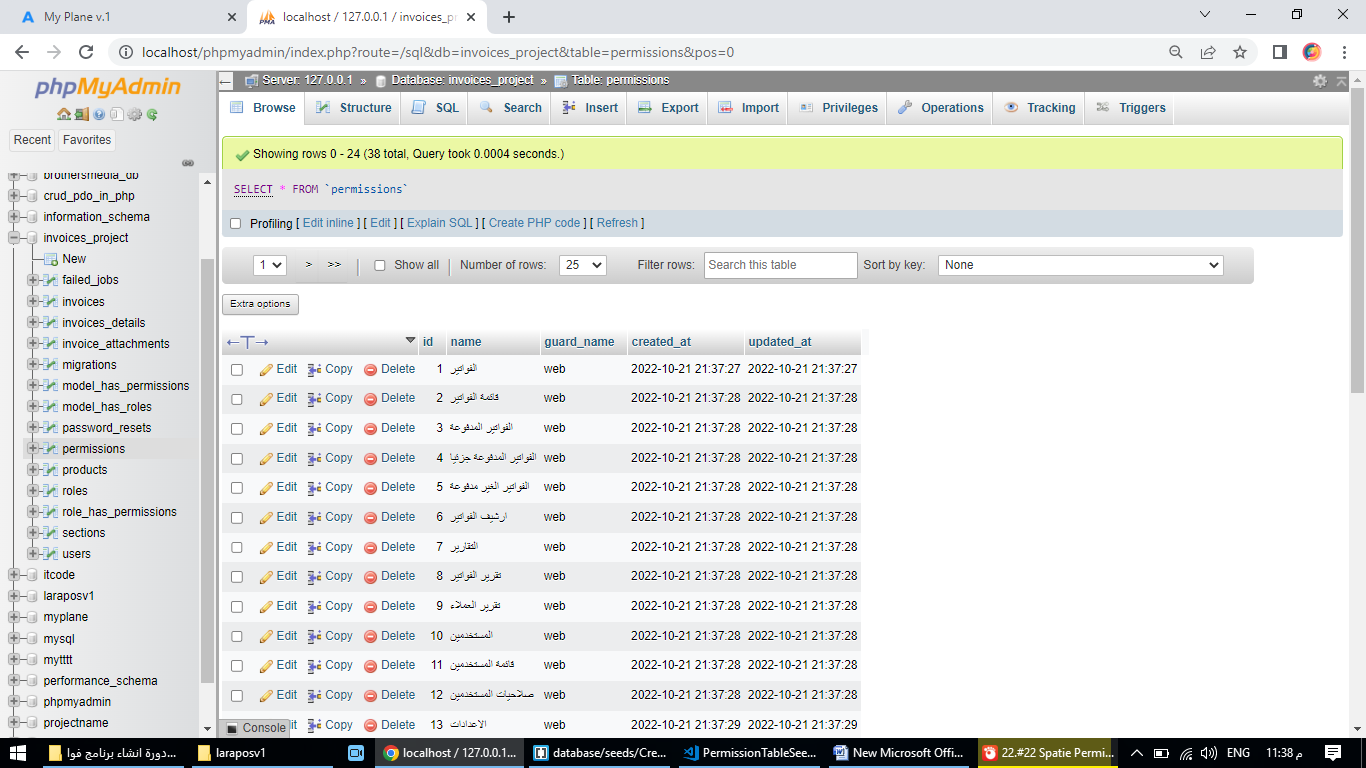
$user->assignRole([$role->id]);

}

}

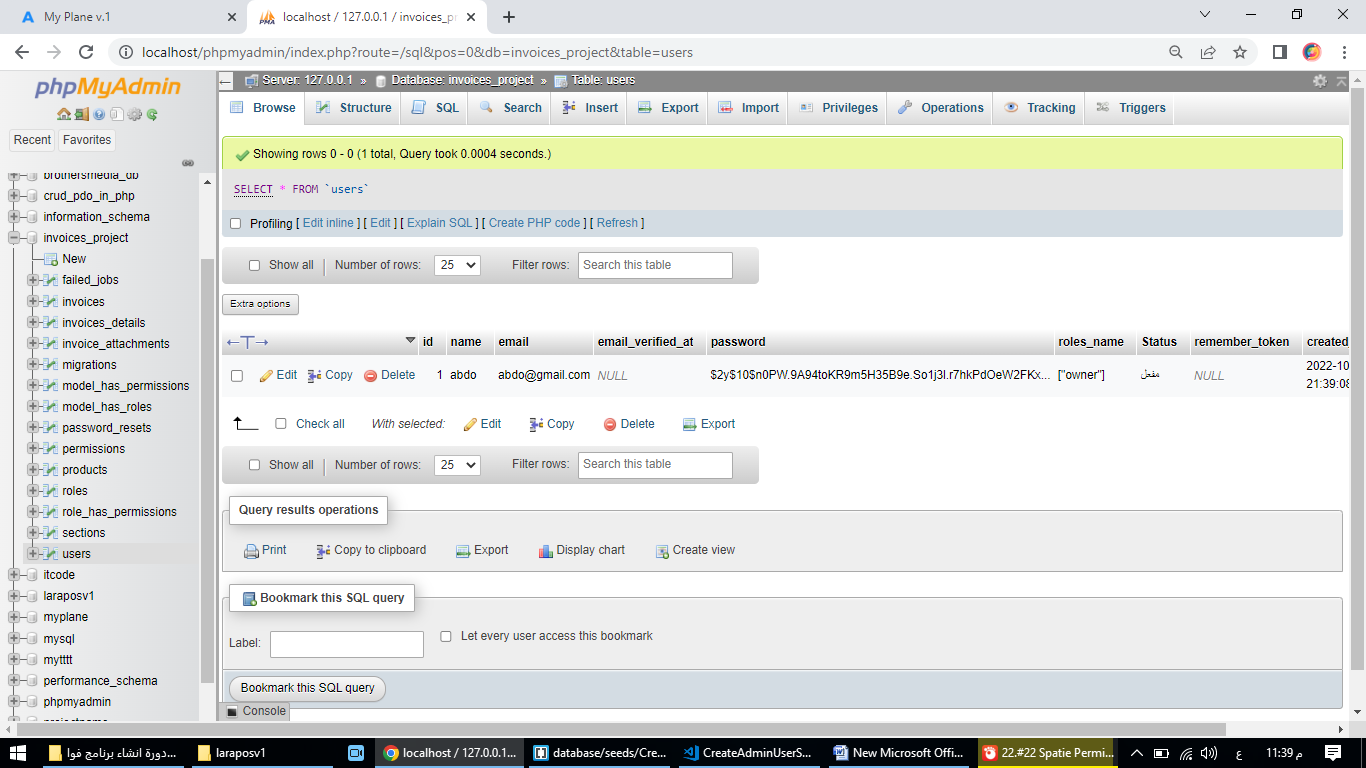
php artisan db:seed --class=PermissionTableSeeder

Database seeding completed successfully



php artisan db:seed --class=CreateAdminUserSeeder

Database seeding completed successfully.



php artisan notifications:table

Migration created successfully!

public function up()

{

Schema::create('notifications', function (Blueprint $table) {

$table->uuid('id')->primary();

$table->string('type');

$table->morphs('notifiable');

$table->text('data');

$table->timestamp('read\_at')->nullable();

$table->timestamps();

});

}

php artisan make:notification AddInvoice

Notification created successfully.

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:notification Add\_invoice\_new

Notification created successfully.

AddInvoice.php

<?php

namespace App\Notifications;

use Illuminate\Bus\Queueable;

use Illuminate\Contracts\Queue\ShouldQueue;

use Illuminate\Notifications\Messages\MailMessage;

use Illuminate\Notifications\Notification;

use App\Models\invoices;

class AddInvoice extends Notification

{

use Queueable;

private $invoice\_id;

public function \_\_construct($invoice\_id)

{

$this->invoice\_id = $invoice\_id;

}

public function via($notifiable)

{

return ['mail'];

}

public function toMail($notifiable)

{

$url = 'http://127.0.0.1:8000/InvoicesDetails/'.$this->invoice\_id;

return (new MailMessage)

->subject('اضافة فاتورة جديدة')

->line('اضافة فاتورة جديدة')

->action('عرض الفاتورة', $url)

->line('شكرا لاستخدامك مورا سوفت لادارة الفواتير');

}

public function toArray($notifiable)

{

return [

//

];

}

}

Add\_invoice\_new.php

<?php

namespace App\Notifications;

use Illuminate\Bus\Queueable;

use Illuminate\Contracts\Queue\ShouldQueue;

use Illuminate\Notifications\Messages\MailMessage;

use Illuminate\Notifications\Notification;

use Illuminate\Support\Facades\Auth;

use App\Models\invoices;

class Add\_invoice\_new extends Notification

{

use Queueable;

private $invoices;

public function \_\_construct(invoices $invoices)

{

$this->invoices = $invoices;

}

public function via($notifiable)

{

return ['database'];

}

public function toDatabase($notifiable)

{

return [

//'data' => $this->details['body']

'id'=> $this->invoices->id,

'title'=>'تم اضافة فاتورة جديد بواسطة :',

'user'=> Auth::user()->name,

];

}

}

php artisan make:controller AdminController

Controller created successfully.

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:controller Customers\_Report

Controller created successfully.

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:controller HomeController

Controller created successfully.

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:controller InvoiceAchiveController -r

Controller created successfully.

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:controller InvoiceAttachmentsController -r

Controller created successfully.

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:controller Invoices\_Report

Controller created successfully.

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:controller InvoicesController -r

Controller created successfully.

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:controller InvoicesDetailsController -r

Controller created successfully.

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:controller ProductsController -r

Controller created successfully.

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:controller SectionsController -r

Controller created successfully.

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:controller UserController -r

Controller created successfully.

PS C:\xampp\htdocs\Invoices\_Project> php artisan make:controller RoleController -r

Controller created successfully.

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

use Spatie\Permission\Models\Role;

use Spatie\Permission\Models\Permission;

use DB;

class RoleController extends Controller

{

function \_\_construct()

{

$this->middleware('permission:عرض صلاحية', ['only' => ['index']]);

$this->middleware('permission:اضافة صلاحية', ['only' => ['create','store']]);

$this->middleware('permission:تعديل صلاحية', ['only' => ['edit','update']]);

$this->middleware('permission:حذف صلاحية', ['only' => ['destroy']]);

}

public function index(Request $request)

{

$roles = Role::orderBy('id','DESC')->paginate(5);

return view('roles.index',compact('roles'))

->with('i', ($request->input('page', 1) - 1) \* 5);

}

public function create()

{

$permission = Permission::get();

return view('roles.create',compact('permission'));

}

public function store(Request $request)

{

$this->validate($request, [

'name' => 'required|unique:roles,name',

'permission' => 'required',

]);

$role = Role::create(['name' => $request->input('name')]);

$role->syncPermissions($request->input('permission'));

return redirect()->route('roles.index')

->with('success','Role created successfully');

}

public function show($id)

{

$role = Role::find($id);

$rolePermissions = Permission::join("role\_has\_permissions","role\_has\_permissions.permission\_id","=","permissions.id")

->where("role\_has\_permissions.role\_id",$id)

->get();

return view('roles.show',compact('role','rolePermissions'));

}

public function edit($id)

{

$role = Role::find($id);

$permission = Permission::get();

$rolePermissions = DB::table("role\_has\_permissions")->where("role\_has\_permissions.role\_id",$id)

->pluck('role\_has\_permissions.permission\_id','role\_has\_permissions.permission\_id')

->all();

return view('roles.edit',compact('role','permission','rolePermissions'));

}

public function update(Request $request, $id)

{

$this->validate($request, [

'name' => 'required',

'permission' => 'required',

]);

$role = Role::find($id);

$role->name = $request->input('name');

$role->save();

$role->syncPermissions($request->input('permission'));

return redirect()->route('roles.index')

->with('success','Role updated successfully');

}

public function destroy($id)

{

DB::table("roles")->where('id',$id)->delete();

return redirect()->route('roles.index')

->with('success','Role deleted successfully');

}

}

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

use App\Models\User;

use Spatie\Permission\Models\Role;

use DB;

use Hash;

class UserController extends Controller

{

public function index(Request $request)

{

$data = User::orderBy('id','DESC')->paginate(5);

return view('users.show\_users',compact('data'))

->with('i', ($request->input('page', 1) - 1) \* 5);

}

public function create()

{

$roles = Role::pluck('name','name')->all();

return view('users.Add\_user',compact('roles'));

}

public function store(Request $request)

{

$this->validate($request, [

'name' => 'required',

'email' => 'required|email|unique:users,email',

'password' => 'required|same:confirm-password',

'roles\_name' => 'required'

]);

$input = $request->all();

$input['password'] = Hash::make($input['password']);

$user = User::create($input);

$user->assignRole($request->input('roles\_name'));

return redirect()->route('users.index')

->with('success','تم اضافة المستخدم بنجاح');

}

public function show($id)

{

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

return view('users.show',compact('user'));

}

public function edit($id)

{

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

$roles = Role::pluck('name','name')->all();

$userRole = $user->roles->pluck('name','name')->all();

return view('users.edit',compact('user','roles','userRole'));

}

public function update(Request $request, $id)

{

$this->validate($request, [

'name' => 'required',

'email' => 'required|email|unique:users,email,'.$id,

'password' => 'same:confirm-password',

'roles' => 'required'

]);

$input = $request->all();

if(!empty($input['password'])){

$input['password'] = Hash::make($input['password']);

}else{

$input = array\_except($input,array('password'));

}

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

$user->update($input);

DB::table('model\_has\_roles')->where('model\_id',$id)->delete();

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

return redirect()->route('users.index')

->with('success','تم تحديث معلومات المستخدم بنجاح');

}

public function destroy(Request $request)

{

User::find($request->user\_id)->delete();

return redirect()->route('users.index')->with('success','تم حذف المستخدم بنجاح');

}

}

<?php

namespace App\Http\Controllers;

use App\Models\sections;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

class SectionsController extends Controller

{

public function index()

{

$sections = sections::all();

return view('sections.sections',compact('sections'));

}

public function create()

{

//

}

public function store(Request $request)

{

$validatedData = $request->validate([

'section\_name' => 'required|unique:sections|max:255',

],[

'section\_name.required' =>'يرجي ادخال اسم القسم',

'section\_name.unique' =>'اسم القسم مسجل مسبقا',

]);

sections::create([

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

'description' => $request->description,

'Created\_by' => (Auth::user()->name),

]);

session()->flash('Add', 'تم اضافة القسم بنجاح ');

return redirect('/sections');

}

public function show($id)

{

//

}

public function edit($id)

{

//

}

public function update(Request $request)

{

$id = $request->id;

$this->validate($request, [

'section\_name' => 'required|max:255|unique:sections,section\_name,'.$id,

'description' => 'required',

],[

'section\_name.required' =>'يرجي ادخال اسم القسم',

'section\_name.unique' =>'اسم القسم مسجل مسبقا',

'description.required' =>'يرجي ادخال البيان',

]);

$sections = sections::find($id);

$sections->update([

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

'description' => $request->description,

]);

session()->flash('edit','تم تعديل القسم بنجاج');

return redirect('/sections');

}

public function destroy(Request $request)

{

$id = $request->id;

sections::find($id)->delete();

session()->flash('delete','تم حذف القسم بنجاح');

return redirect('/sections');

}

}

<?php

namespace App\Http\Controllers;

use App\Models\products;

use App\Models\sections;

use Illuminate\Http\Request;

class ProductsController extends Controller

{

public function index()

{

$sections = sections::all();

$products = products::all();

return view('products.products', compact('sections','products'));

}

public function create()

{

//

}

public function store(Request $request)

{

Products::create([

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

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

'description' => $request->description,

]);

session()->flash('Add', 'تم اضافة المنتج بنجاح ');

return redirect('/products');

}

public function show($id)

{

//

}

public function edit($id)

{

//

}

public function update(Request $request, $id)

{

$id = sections::where('section\_name', $request->section\_name)->first()->id;

$Products = Products::findOrFail($request->pro\_id);

$Products->update([

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

'description' => $request->description,

'section\_id' => $id,

]);

session()->flash('Edit', 'تم تعديل المنتج بنجاح');

return back();

}

public function destroy(Request $request)

{

$Products = Products::findOrFail($request->pro\_id);

$Products->delete();

session()->flash('delete', 'تم حذف المنتج بنجاح');

return back();

}

}

<?php

namespace App\Http\Controllers;

use App\Models\invoices\_details;

use App\Models\invoices;

use App\Models\invoice\_attachments;

use Illuminate\Support\Facades\Storage;

use File;

use Illuminate\Http\Request;

class InvoicesDetailsController extends Controller

{

public function index()

{

//

}

public function create()

{

//

}

public function store(Request $request)

{

//

}

public function show($id)

{

//

}

public function edit($id)

{

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

$details = invoices\_Details::where('id\_Invoice',$id)->get();

$attachments = invoice\_attachments::where('invoice\_id',$id)->get();

return view('invoices.details\_invoice',compact('invoices','details','attachments'));

}

public function update(Request $request, $id)

{

//

}

public function destroy(Request $request)

{

$invoices = invoice\_attachments::findOrFail($request->id\_file);

$invoices->delete();

Storage::disk('public\_uploads')->delete($request->invoice\_number.'/'.$request->file\_name);

session()->flash('delete', 'تم حذف المرفق بنجاح');

return back();

}

public function get\_file($invoice\_number,$file\_name)

{

$contents= Storage::disk('public\_uploads')->getDriver()->getAdapter()->applyPathPrefix($invoice\_number.'/'.$file\_name);

return response()->download( $contents);

}

public function open\_file($invoice\_number,$file\_name)

{

$files = Storage::disk('public\_uploads')->getDriver()->getAdapter()->applyPathPrefix($invoice\_number.'/'.$file\_name);

return response()->file($files);

}

}

<?php

namespace App\Http\Controllers;

use Illuminate\Support\Facades\Notification;

use App\Models\invoices;

use App\Models\sections;

use App\Models\User;

use App\Models\invoices\_details;

use App\Models\invoice\_attachments;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Storage;

use Illuminate\Support\Facades\DB;

use Illuminate\Http\Request;

use App\Notifications\AddInvoice;

use App\Exports\InvoicesExport;

use Maatwebsite\Excel\Facades\Excel;

use App\Events\MyEventClass;

class InvoicesController extends Controller

{

public function index()

{

$invoices = invoices::all();

return view('invoices.invoices', compact('invoices'));

}

public function create()

{

$sections = sections::all();

return view('invoices.add\_invoice', compact('sections'));

}

public function store(Request $request)

{

invoices::create([

'invoice\_number' => $request->invoice\_number,

'invoice\_Date' => $request->invoice\_Date,

'Due\_date' => $request->Due\_date,

'product' => $request->product,

'section\_id' => $request->Section,

'Amount\_collection' => $request->Amount\_collection,

'Amount\_Commission' => $request->Amount\_Commission,

'Discount' => $request->Discount,

'Value\_VAT' => $request->Value\_VAT,

'Rate\_VAT' => $request->Rate\_VAT,

'Total' => $request->Total,

'Status' => 'غير مدفوعة',

'Value\_Status' => 2,

'note' => $request->note,

]);

$invoice\_id = invoices::latest()->first()->id;

invoices\_details::create([

'id\_Invoice' => $invoice\_id,

'invoice\_number' => $request->invoice\_number,

'product' => $request->product,

'Section' => $request->Section,

'Status' => 'غير مدفوعة',

'Value\_Status' => 2,

'note' => $request->note,

'user' => (Auth::user()->name),

]);

if ($request->hasFile('pic')) {

$invoice\_id = Invoices::latest()->first()->id;

$image = $request->file('pic');

$file\_name = $image->getClientOriginalName();

$invoice\_number = $request->invoice\_number;

$attachments = new invoice\_attachments();

$attachments->file\_name = $file\_name;

$attachments->invoice\_number = $invoice\_number;

$attachments->Created\_by = Auth::user()->name;

$attachments->invoice\_id = $invoice\_id;

$attachments->save();

// move pic

$imageName = $request->pic->getClientOriginalName();

$request->pic->move(public\_path('Attachments/' . $invoice\_number), $imageName);

}

// $user = User::first();

// Notification::send($user, new AddInvoice($invoice\_id));

$user = User::get();

$invoices = invoices::latest()->first();

Notification::send($user, new \App\Notifications\Add\_invoice\_new($invoices));

event(new MyEventClass('hello world'));

session()->flash('Add', 'تم اضافة الفاتورة بنجاح');

return back();

}

public function show($id)

{

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

return view('invoices.status\_update', compact('invoices'));

}

public function edit($id)

{

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

$sections = sections::all();

return view('invoices.edit\_invoice', compact('sections', 'invoices'));

}

public function update(Request $request)

{

$invoices = invoices::findOrFail($request->invoice\_id);

$invoices->update([

'invoice\_number' => $request->invoice\_number,

'invoice\_Date' => $request->invoice\_Date,

'Due\_date' => $request->Due\_date,

'product' => $request->product,

'section\_id' => $request->Section,

'Amount\_collection' => $request->Amount\_collection,

'Amount\_Commission' => $request->Amount\_Commission,

'Discount' => $request->Discount,

'Value\_VAT' => $request->Value\_VAT,

'Rate\_VAT' => $request->Rate\_VAT,

'Total' => $request->Total,

'note' => $request->note,

]);

session()->flash('edit', 'تم تعديل الفاتورة بنجاح');

return back();

}

public function destroy(Request $request)

{

$id = $request->invoice\_id;

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

$Details = invoice\_attachments::where('invoice\_id', $id)->first();

$id\_page =$request->id\_page;

if (!$id\_page==2) {

if (!empty($Details->invoice\_number)) {

Storage::disk('public\_uploads')->deleteDirectory($Details->invoice\_number);

}

$invoices->forceDelete();

session()->flash('delete\_invoice');

return redirect('/invoices');

}

else {

$invoices->delete();

session()->flash('archive\_invoice');

return redirect('/Archive');

}

}

public function getproducts($id)

{

$products = DB::table("products")->where("section\_id", $id)->pluck("Product\_name", "id");

return json\_encode($products);

}

public function Status\_Update($id, Request $request)

{

$invoices = invoices::findOrFail($id);

if ($request->Status === 'مدفوعة') {

$invoices->update([

'Value\_Status' => 1,

'Status' => $request->Status,

'Payment\_Date' => $request->Payment\_Date,

]);

invoices\_Details::create([

'id\_Invoice' => $request->invoice\_id,

'invoice\_number' => $request->invoice\_number,

'product' => $request->product,

'Section' => $request->Section,

'Status' => $request->Status,

'Value\_Status' => 1,

'note' => $request->note,

'Payment\_Date' => $request->Payment\_Date,

'user' => (Auth::user()->name),

]);

}

else {

$invoices->update([

'Value\_Status' => 3,

'Status' => $request->Status,

'Payment\_Date' => $request->Payment\_Date,

]);

invoices\_Details::create([

'id\_Invoice' => $request->invoice\_id,

'invoice\_number' => $request->invoice\_number,

'product' => $request->product,

'Section' => $request->Section,

'Status' => $request->Status,

'Value\_Status' => 3,

'note' => $request->note,

'Payment\_Date' => $request->Payment\_Date,

'user' => (Auth::user()->name),

]);

}

session()->flash('Status\_Update');

return redirect('/invoices');

}

public function Invoice\_Paid()

{

$invoices = Invoices::where('Value\_Status', 1)->get();

return view('invoices.invoices\_paid',compact('invoices'));

}

public function Invoice\_unPaid()

{

$invoices = Invoices::where('Value\_Status',2)->get();

return view('invoices.invoices\_unpaid',compact('invoices'));

}

public function Invoice\_Partial()

{

$invoices = Invoices::where('Value\_Status',3)->get();

return view('invoices.invoices\_Partial',compact('invoices'));

}

public function Print\_invoice($id)

{

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

return view('invoices.Print\_invoice',compact('invoices'));

}

public function export()

{

return Excel::download(new InvoicesExport, 'invoices.xlsx');

}

public function MarkAsRead\_all (Request $request)

{

$userUnreadNotification= auth()->user()->unreadNotifications;

if($userUnreadNotification) {

$userUnreadNotification->markAsRead();

return back();

}

}

public function unreadNotifications\_count()

{

return auth()->user()->unreadNotifications->count();

}

public function unreadNotifications()

{

foreach (auth()->user()->unreadNotifications as $notification){

return $notification->data['title'];

}

}

}

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\invoices;

class Invoices\_Report extends Controller

{

public function index(){

return view('reports.invoices\_report');

}

public function Search\_invoices(Request $request){

$rdio = $request->rdio;

// في حالة البحث بنوع الفاتورة

if ($rdio == 1) {

// في حالة عدم تحديد تاريخ

if ($request->type && $request->start\_at =='' && $request->end\_at =='') {

$invoices = invoices::select('\*')->where('Status','=',$request->type)->get();

$type = $request->type;

return view('reports.invoices\_report',compact('type'))->withDetails($invoices);

}

// في حالة تحديد تاريخ استحقاق

else {

$start\_at = date($request->start\_at);

$end\_at = date($request->end\_at);

$type = $request->type;

$invoices = invoices::whereBetween('invoice\_Date',[$start\_at,$end\_at])->where('Status','=',$request->type)->get();

return view('reports.invoices\_report',compact('type','start\_at','end\_at'))->withDetails($invoices);

}

}

//====================================================================

// في البحث برقم الفاتورة

else {

$invoices = invoices::select('\*')->where('invoice\_number','=',$request->invoice\_number)->get();

return view('reports.invoices\_report')->withDetails($invoices);

}

}

}

<?php

namespace App\Http\Controllers;

use App\Models\invoice\_attachments;

use Illuminate\Http\Request;

class InvoiceAttachmentsController extends Controller

{

public function index()

{

//

}

public function create()

{

//

}

public function store(Request $request)

{

$this->validate($request, [

'file\_name' => 'mimes:pdf,jpeg,png,jpg',

], [

'file\_name.mimes' => 'صيغة المرفق يجب ان تكون pdf, jpeg , png , jpg',

]);

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

$file\_name = $image->getClientOriginalName();

$attachments = new invoice\_attachments();

$attachments->file\_name = $file\_name;

$attachments->invoice\_number = $request->invoice\_number;

$attachments->invoice\_id = $request->invoice\_id;

$attachments->Created\_by = Auth::user()->name;

$attachments->save();

// move pic

$imageName = $request->file\_name->getClientOriginalName();

$request->file\_name->move(public\_path('Attachments/'. $request->invoice\_number), $imageName);

session()->flash('Add', 'تم اضافة المرفق بنجاح');

return back();

}

/\*\*

\* Display the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function show($id)

{

//

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function edit($id)

{

//

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function update(Request $request, $id)

{

//

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function destroy($id)

{

//

}

}

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\invoices;

class InvoiceAchiveController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

$invoices = invoices::onlyTrashed()->get();

return view('Invoices.Archive\_Invoices',compact('invoices'));

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function create()

{

//

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function store(Request $request)

{

//

}

/\*\*

\* Display the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function show($id)

{

//

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function edit($id)

{

//

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function update(Request $request)

{

$id = $request->invoice\_id;

$flight = Invoices::withTrashed()->where('id', $id)->restore();

session()->flash('restore\_invoice');

return redirect('/invoices');

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function destroy($id)

{

$invoices = invoices::withTrashed()->where('id',$request->invoice\_id)->first();

$invoices->forceDelete();

session()->flash('delete\_invoice');

return redirect('/Archive');

}

}

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\invoices;

class HomeController extends Controller

{

public function \_\_construct()

{

$this->middleware('auth');

}

/\*\*

\* Show the application dashboard.

\*

\* @return \Illuminate\Contracts\Support\Renderable

\*/

public function index()

{

//=================احصائية نسبة تنفيذ الحالات======================

$count\_all =invoices::count();

$count\_invoices1 = invoices::where('Value\_Status', 1)->count();

$count\_invoices2 = invoices::where('Value\_Status', 2)->count();

$count\_invoices3 = invoices::where('Value\_Status', 3)->count();

if($count\_invoices2 == 0){

$nspainvoices2=0;

}

else{

$nspainvoices2 = $count\_invoices2/ $count\_all\*100;

}

if($count\_invoices1 == 0){

$nspainvoices1=0;

}

else{

$nspainvoices1 = $count\_invoices1/ $count\_all\*100;

}

if($count\_invoices3 == 0){

$nspainvoices3=0;

}

else{

$nspainvoices3 = $count\_invoices3/ $count\_all\*100;

}

$chartjs = app()->chartjs

->name('barChartTest')

->type('bar')

->size(['width' => 350, 'height' => 200])

->labels(['الفواتير الغير المدفوعة', 'الفواتير المدفوعة','الفواتير المدفوعة جزئيا'])

->datasets([

[

"label" => "الفواتير الغير المدفوعة",

'backgroundColor' => ['#ec5858'],

'data' => [$nspainvoices2]

],

[

"label" => "الفواتير المدفوعة",

'backgroundColor' => ['#81b214'],

'data' => [$nspainvoices1]

],

[

"label" => "الفواتير المدفوعة جزئيا",

'backgroundColor' => ['#ff9642'],

'data' => [$nspainvoices3]

],

])

->options([]);

$chartjs\_2 = app()->chartjs

->name('pieChartTest')

->type('pie')

->size(['width' => 340, 'height' => 200])

->labels(['الفواتير الغير المدفوعة', 'الفواتير المدفوعة','الفواتير المدفوعة جزئيا'])

->datasets([

[

'backgroundColor' => ['#ec5858', '#81b214','#ff9642'],

'data' => [$nspainvoices2, $nspainvoices1,$nspainvoices3]

]

])

->options([]);

return view('home', compact('chartjs','chartjs\_2'));

}

}

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\sections;

use App\Models\invoices;

class Customers\_Report extends Controller

{

public function index(){

$sections = sections::all();

return view('reports.customers\_report',compact('sections'));

}

public function Search\_customers(Request $request){

// في حالة البحث بدون التاريخ

if ($request->Section && $request->product && $request->start\_at =='' && $request->end\_at=='') {

$invoices = invoices::select('\*')->where('section\_id','=',$request->Section)->where('product','=',$request->product)->get();

$sections = sections::all();

return view('reports.customers\_report',compact('sections'))->withDetails($invoices);

}

// في حالة البحث بتاريخ

else {

$start\_at = date($request->start\_at);

$end\_at = date($request->end\_at);

$invoices = invoices::whereBetween('invoice\_Date',[$start\_at,$end\_at])->where('section\_id','=',$request->Section)->where('product','=',$request->product)->get();

$sections = sections::all();

return view('reports.customers\_report',compact('sections'))->withDetails($invoices);

}

}

}

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class AdminController extends Controller

{

public function index($id)

{

if(view()->exists($id)){

return view($id);

}

else

{

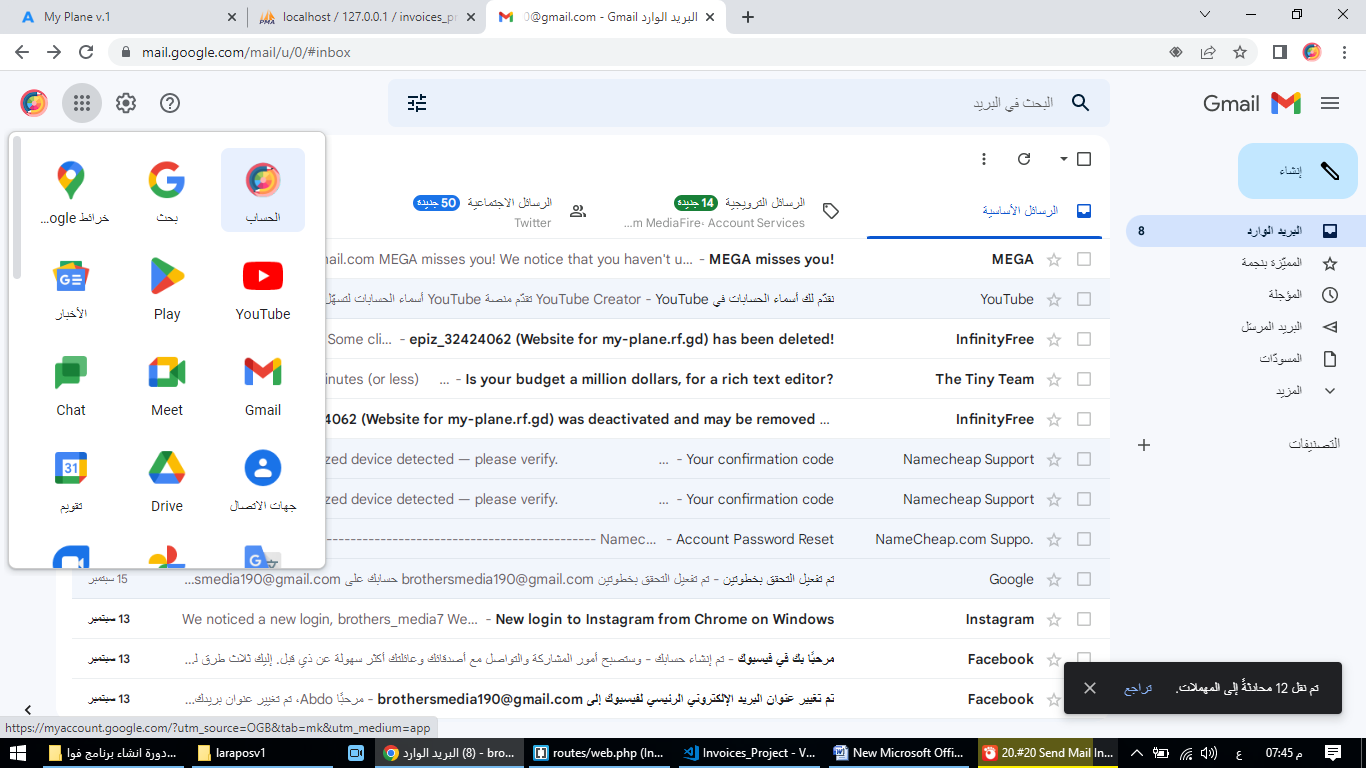
return view('404');

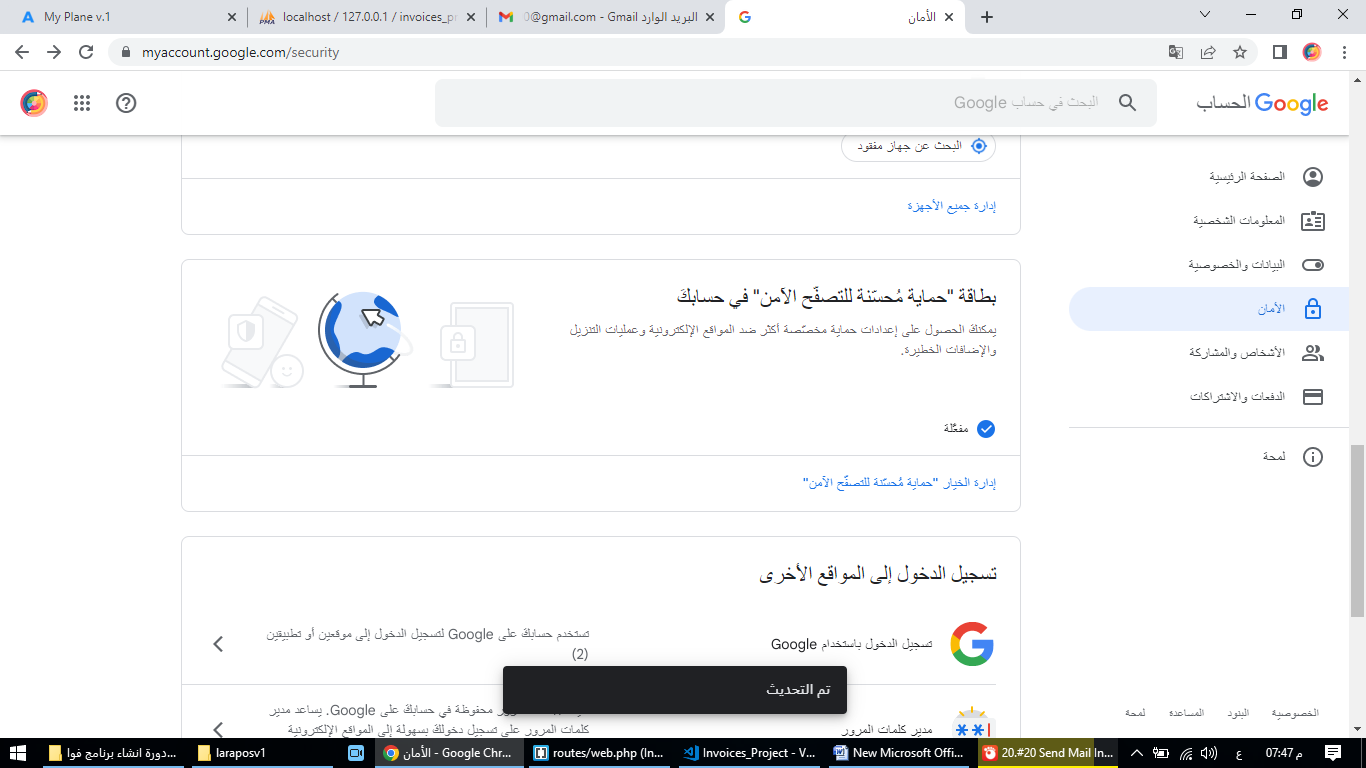
}

// return view($id);

}

}





.env

MAIL\_MAILER=smtp

MAIL\_HOST=smtp.gmail.com

MAIL\_PORT=465

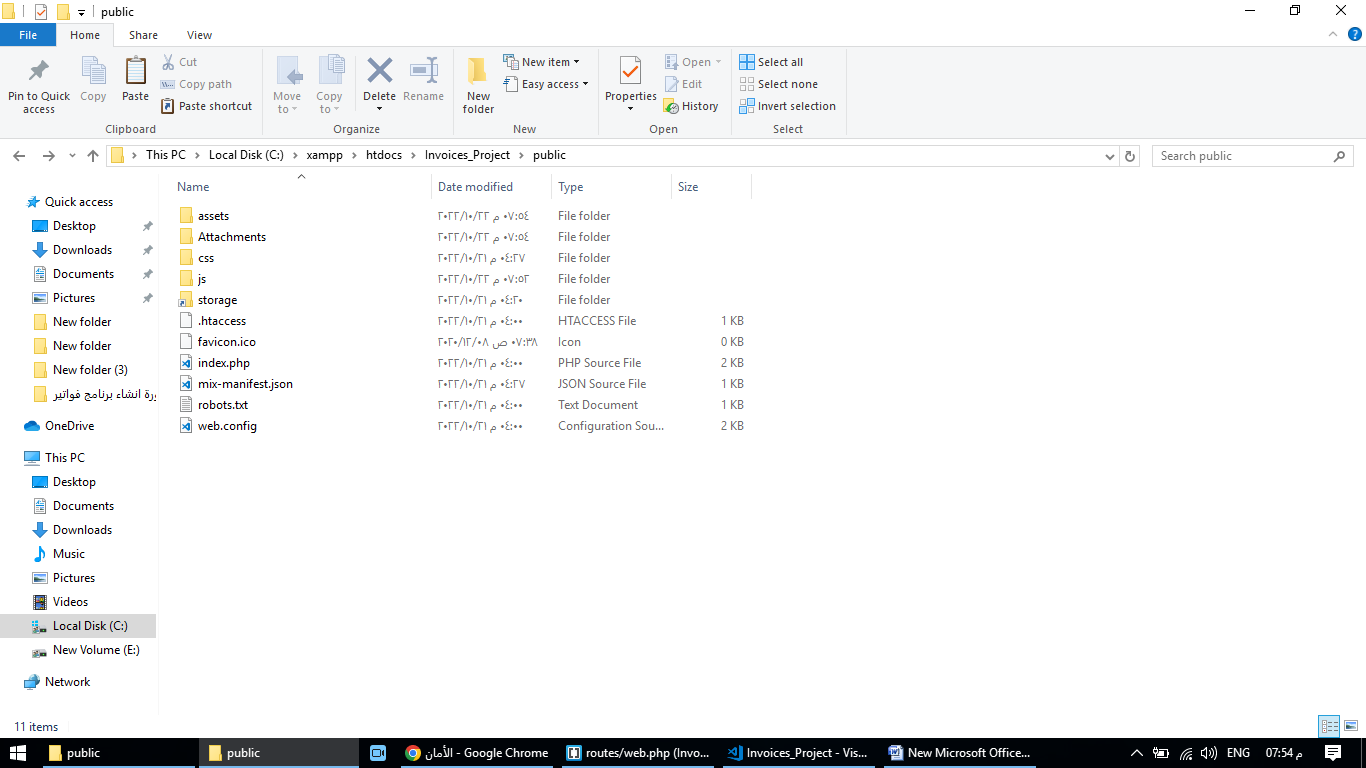
MAIL\_USERNAME=brothersmedia190@gmail.com

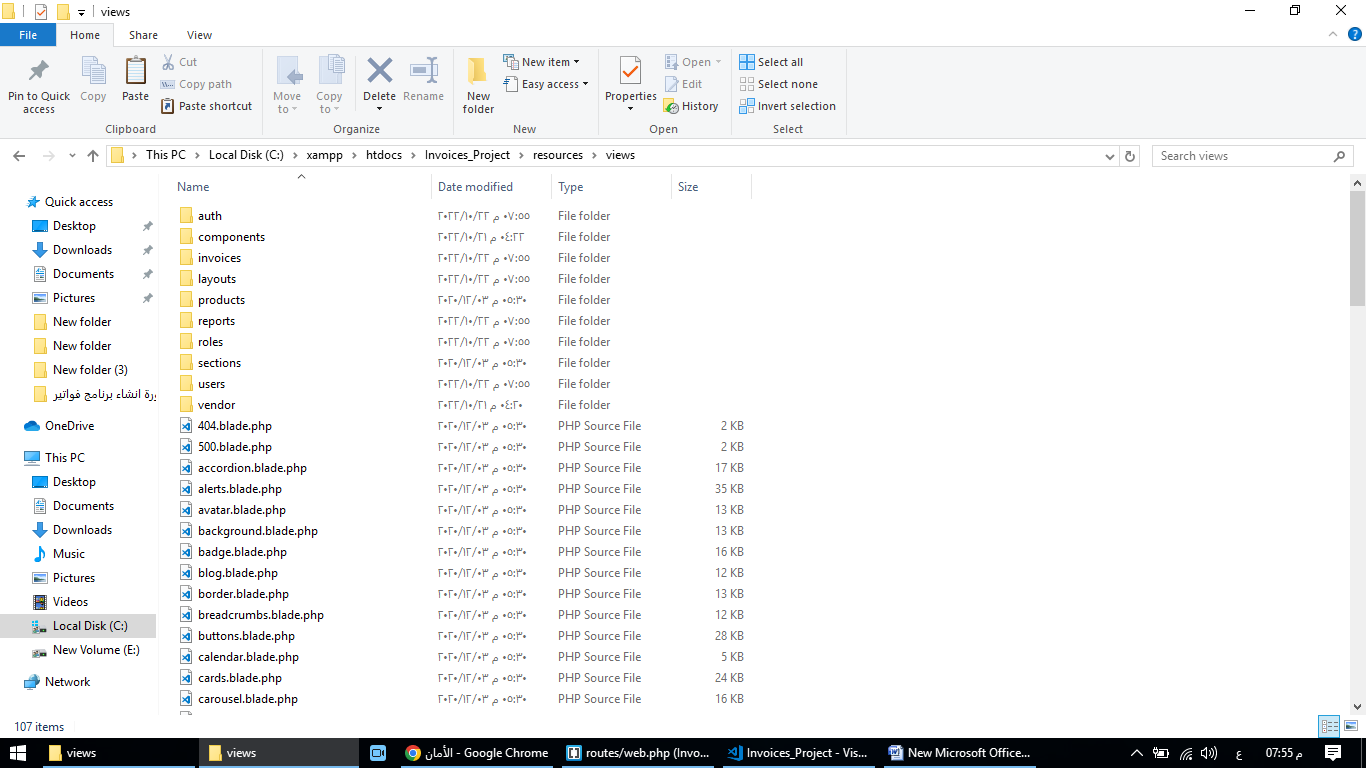
MAIL\_PASSWORD=Abdo1379

MAIL\_ENCRYPTION=ssl

MAIL\_FROM\_ADDRESS=brothersmedia190@gmail.com

MAIL\_FROM\_NAME="${APP\_NAME}"





<https://github.com/fxcosta/laravel-chartjs>

composer require fx3costa/laravelchartjs

app.php

Fx3costa\LaravelChartJs\Providers\ChartjsServiceProvider::class,

<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\AdminController;

use App\Http\Controllers\Customers\_Report;

use App\Http\Controllers\HomeController;

use App\Http\Controllers\InvoiceAchiveController;

use App\Http\Controllers\InvoiceAttachmentsController;

use App\Http\Controllers\Invoices\_Report;

use App\Http\Controllers\InvoicesController;

use App\Http\Controllers\InvoicesDetailsController;

use App\Http\Controllers\ProductsController;

use App\Http\Controllers\RoleController;

use App\Http\Controllers\SectionsController;

use App\Http\Controllers\UserController;

Route::get('/', function () {

return view('auth.login');

});

require \_\_DIR\_\_.'/auth.php';

Route::get('/dashboard', [HomeController::class, 'index'])->name('dashboard');

Route::resource('invoices', InvoicesController::class);

Route::resource('sections', SectionsController::class);

Route::resource('products', ProductsController::class);

Route::resource('InvoiceAttachments', InvoiceAttachmentsController::class);

Route::get('/section/{id}', [InvoicesController::class, 'getproducts'])->name('getproducts');

Route::get('/InvoicesDetails/{id}', [InvoicesDetailsController::class, 'edit'])->name('edit');

Route::get('download/{invoice\_number}/{file\_name}', [InvoicesDetailsController::class, 'get\_file'])->name('get\_file');

Route::get('View\_file/{invoice\_number}/{file\_name}', [InvoicesDetailsController::class, 'open\_file'])->name('open\_file');

Route::post('delete\_file', [InvoicesDetailsController::class, 'destroy'])->name('delete\_file');

Route::get('/edit\_invoice/{id}', [InvoicesController::class, 'edit'])->name('edit');

Route::get('/Status\_show/{id}', [InvoicesController::class, 'show'])->name('Status\_show');

Route::post('/Status\_Update/{id}', [InvoicesController::class, 'Status\_Update'])->name('Status\_Update');

Route::resource('Archive', InvoiceAchiveController::class);

Route::get('Invoice\_Paid', [InvoicesController::class, 'Invoice\_Paid'])->name('Invoice\_Paid');

Route::get('Invoice\_UnPaid', [InvoicesController::class, 'Invoice\_UnPaid'])->name('Invoice\_UnPaid');

Route::get('Invoice\_Partial', [InvoicesController::class, 'Invoice\_Partial'])->name('Invoice\_Partial');

Route::get('Print\_invoice/{id}', [InvoicesController::class, 'Print\_invoice'])->name('Print\_invoice');

Route::get('export\_invoices', [InvoicesController::class, 'export'])->name('export');

Route::group(['middleware' => ['auth']], function() {

Route::resource('roles', RoleController::class);

Route::resource('users', UserController::class);

});

Route::get('invoices\_report', [Invoices\_Report::class, 'index'])->name('index');

Route::get('customers\_report', [Customers\_Report::class, 'index'])->name('customers\_report');

Route::post('Search\_customers', [Customers\_Report::class, 'Search\_customers'])->name('Search\_customers');

Route::get('MarkAsRead\_all', [InvoicesController::class, 'MarkAsRead\_all'])->name('MarkAsRead\_all');

Route::get('unreadNotifications\_count', [InvoicesController::class, 'unreadNotifications\_count'])->name('unreadNotifications\_count');

Route::get('unreadNotifications', [InvoicesController::class, 'unreadNotifications'])->name('unreadNotifications');

Route::get('/{page}', [AdminController::class, 'index'])->name('index');

Route::post('Search\_invoices', [Invoices\_Report::class, 'Search\_invoices'])->name('Search\_invoices');

filesystems.php

'disks' => [

'local' => [

'driver' => 'local',

'root' => storage\_path('app'),

],

'public\_uploads' => [

'driver' => 'local',

'root' => public\_path() . '/Attachments',

],

'public' => [

'driver' => 'local',

'root' => storage\_path('app/public'),

'url' => env('APP\_URL').'/storage',

'visibility' => 'public',

],

's3' => [

'driver' => 's3',

'key' => env('AWS\_ACCESS\_KEY\_ID'),

'secret' => env('AWS\_SECRET\_ACCESS\_KEY'),

'region' => env('AWS\_DEFAULT\_REGION'),

'bucket' => env('AWS\_BUCKET'),

'url' => env('AWS\_URL'),

'endpoint' => env('AWS\_ENDPOINT'),

],

],