ДОДАТОК А

Код модуля ERP

```
<?php
namespace App\Http\Controllers;

use App\Models\Company;
use App\Models\Employee;
use App\Models\OrdersService;
use App\Models\Service;
use App\Services\CompanyService;
use App\Services\OrderService;
use Asantibanez\LivewireCharts\Models\ColumnChartModel;
use Asantibanez\LivewireCharts\Models\LineChartModel;
use Asantibanez\LivewireCharts\Facades\LivewireCharts;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class CompanyController extends Controller
{


    public function __invoke($id)
    {
        setlocale(LC_TIME, 'ru_RU.UTF–8');
        $company = Company::with('employees', 'services')–>findOrFail($id);
        $employees = $company–>employees;

        $lineData =  $company–>orderServices–>sortBy('created_at')–
```

```
>groupBy(function ($order) {
            return Carbon::parse($order–>created_at)–>format('Y–m');
        })–>map(function ($orders) {
            return [
                'month' => Carbon::parse($orders–>first()–>created_at)–
>formatLocalized('%B'),
                'sum' => $orders–>sum('price'),
                'count' => $orders–>count(),
            ];
        });

        $key = 0;


        $moneyChartModel =  $lineData
            –>reduce(function ($lineChartModel, $data) use ($lineData, $key) {
                $key++;
                return $lineChartModel–>addPoint($data['month'],  $data['sum'], ['id' =>
$key]);
            },  LivewireCharts::lineChartModel()
                –>setAnimated(true)
                –>setSmoothCurve()
                –>setThemePalette('palette10')
                –>setXAxisVisible(true)
                –>setYAxisVisible(false)
                –>withDataLabels()
                –>setColors(['#6100FF','#6630BF','#3F00A6','#8840FF','#A873FF'])
            );
```

```
        $lineChartModel =  $lineData
            –>reduce(function ($lineChartModel, $data) use ($lineData, $key) {
                $key++;
                return $lineChartModel–>addPoint($data['month'], $data['count'], ['id'
=> $key]);
            },  LivewireCharts::lineChartModel()
                –>setAnimated(true)
                –>setSmoothCurve()
                –>setThemePalette('palette10')
                –>setXAxisVisible(true)
                –>setYAxisVisible(false)
                –>withDataLabels()
                –>setColors(['#6100FF','#6630BF','#3F00A6','#8840FF','#A873FF'])
            );

        $pieData = Employee::query()–>company($company–>id)–
>withCount('orderServices')–>get();

        $pieChartModel = $pieData
            –>reduce(function ($pieChartModel, $data) {

                $value = $data–>order_services_count;

                return $pieChartModel–>addSlice($data–>name,$value,"#272727");
            }, LivewireCharts::pieChartModel()
                –>enableShades()
                –>setAnimated(true)
                –>setType('donut')
```

```
            –>legendPositionBottom()
            –>legendPositionLeft()
            –>legendHorizontallyAlignedCenter()
            –>setDataLabelsEnabled(true)
            –>setColors(['#6100FF','#6630BF','#3F00A6','#8840FF','#A873FF'])
        );

    $columnData = Service::query()–>company($company–>id)–
>withCount('orders')–>get();
    $columnChartModel = $columnData
        –>reduce(function ($columnChartModel, $data) {

            return $columnChartModel–>addColumn($data–>title, $data–
>orders_count, '#f1f1f1');
        }, LivewireCharts::columnChartModel()
            –>setAnimated(true)
            –>setLegendVisibility(false)
            –>setDataLabelsEnabled(true)
            //–>setOpacity(0.25)
            –>setColors(['#6100FF','#6630BF','#3F00A6','#8840FF','#A873FF'])
            –>setColumnWidth(90)
            –>withGrid()
        );
    ;
    return view('client.user.company', compact('company',
'lineChartModel','employees', 'columnChartModel','pieChartModel',
'moneyChartModel'));
  }
  public function store(Request $request, CompanyService $service) : Company
```

```
{
    return $service–>store($request);
}
public function getClients($id)
{
     $company = Company::findOrFail($id);
     return view('client.user.customers', compact('company'));
}
public function getOrders($id){
    $company = Company::findOrFail($id);
    return view('client.user.orders', compact('company'));
}
public function getServices($id){
    $company = Company::findOrFail($id);
    return view('client.user.services', compact('company'));
}
public function getStanki($id){

    return view('client.user.stanki');
}
public function getEmployees($id){
    $company = Company::findOrFail($id);
    return view('client.user.employees', compact('company'));
}

public function updateOrder(Request $request, OrderService $service){
    $service–>update($request–>post('id'), [
        'status_id' => $request–>post('status_id')
    ]);
```

```
        return 'OK';
    }
}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Illuminate\Database\Eloquent\Relations\HasManyThrough;

class Company extends Model
{
    use HasFactory;

    protected $fillable = [

        'title', 'user_id'
    ];

    protected $with = [
        'orderStatuses',
        'employees',
        'services',
        'orders',
        'customers'
    ];
```

```
public function orderStatuses() : HasMany
{
    return $this–>hasMany(OrderStatus::class);
}

public function orderServices() : HasManyThrough
{
    return $this–>through('orders')–>has('services');
}

public function employees() : HasMany
{
    return $this–>hasMany(Employee::class);
}

public function services() : HasMany
{
    return $this–>hasMany(Service::class);
}

public function orders() : HasMany
{
    return $this–>hasMany(Order::class)–>with('customer');
}

public function customers() : HasMany
{
    return $this–>hasMany(Customer::class);
```

```
    }


}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Builder;
use Illuminate\Database\Eloquent\Casts\Attribute;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Relations\BelongsToMany;
use Illuminate\Database\Eloquent\Relations\HasMany;

class Order extends Model
{
    use HasFactory;

    protected $fillable = [
        'title',
        'uid',
        'description',
        'company_id',
        'customer_id',
        'status_id',
    ];
```

```
    protected $with = [
        'customer',
        'services',
        'employees'
    ];

    public function customer() : BelongsTo
    {
        return $this–>belongsTo(Customer::class);
    }

    public function companyInfo() : BelongsTo
    {
        return $this–>belongsTo(Company::class);
    }

    public function services() : hasMany
    {
        return $this–>hasMany(OrdersService::class, 'order_id', 'id')–>with('service');
    }

    public function employees() : hasMany
    {
        return $this–>hasMany(OrdersService::class, 'order_id', 'id')–
>with('employee');
    }

    public function scopeCompany(Builder $query,$id) : Builder
```

```
    {
        return $query–>where('company_id', $id);
    }
}

<?php

namespace App\Services;

use App\Models\Order;
use App\Models\OrdersService;

/**
 * Class OrderService.
 */
class OrderService
{
    public function deleteService($rowId){
        $service = OrdersService::findOrFail($rowId);
        $service–>delete();
        return 'deleted';
    }
    public function getPrintOrder($id){
        $order = Order::find($id);

        $print = "<h1>$order–>uid</h1>";

        return $print;
    }
```

```
public function store(array $args)
{
    $order = new Order();
    $order–>uid = rand(000000,999999);
    $order–>fill($args);
    if(array_key_exists('services', $args)){
        $order–>services()
            –>attach($args['services']);
    }
    if(array_key_exists('employees', $args)){
        $order–>employees()
            –>attach($args['employees']);
    }
    $order–>save();

    return $order;
}

public function update($id, array $arr)
{
    $order = Order::findOrFail($id);
    $order–>fill($arr)–>save();
}
public function destroy($id)
{
    $order = Order::findOrFail($id);
    $order–>delete();
}
```

```
    public function addService(array $args)
    {
        $service = new OrdersService();
        $service–>fill($args);
        $service–>save();
        return response()–>json($service);
    }
}
```

ДОДАТОК Б

Демонстраційний матеріал

Формат	Позиція.	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кільк.	Примітки
			Текстові документи		
А4	1	ГЮІК 505300.005 ПЗ	Пояснювальна записка	63	с.
			Інші документи		
	2		Код модуля ERP	12	с.
	3		Демонстраційний матеріал	10	с.

					ГЮІК 505300.005 ВД			
Змін.	Арк.	Номер докум.	Підпис	Дата				
Розроб.		Залуцький Д. Д.			Розроблення автоматичного програмного модуля адміністрування підприємством	Літера	Арку	Аркуш
Перевір.		Гурін Д. В.				Н	1	1
						Кафедра КІТАР ХНУРЕ		
Н.контр.		Стародубцев М.Г.						
Затв.		Невлюдов І.Ш.						