6

OLTINCHI AMALIY MASHG‘ULOT

Korxona ilovalarini dasturlash · 2 soat · 6-mavzu

CRUD ekranlari: filtr, sahifalash, validatsiya

Jamoa loyihasi bosqichi · 5 ta PHP topshiriq · o‘qituvchi bahosi

Blade | Livewire 3 | Sahifalash | Formalar | Dashboard

MASHG‘ULOT MAQSADI

Amaliy mashg‘ulot oxirida siz quyidagilarni bajara olasiz...

1. **Yaratish** — Livewire 3 `ProductTable` komponentini: qidiruv, kategoriya filtri, saralash va sahifalash bilan

2. **Qurish** — buyurtmalar ro‘yxati va `OrderForm` hujjat formasini dinamik qatorlar bilan

3. **Qo‘llash** — FormRequest va Livewire validatsiyasini, xato xabarlarini o‘zbek tilida ko‘rsatishni

4. **Bog‘lash** — tugmalarni rollarga (`@can`, Policy) va saqlashni servis qatlamiga (`OrderService`)

5. **Tekshirish** — ekranlarni `Livewire::test()` bilan va UX nazorat ro‘yxati bo‘yicha

Qanday bajariladi va qanday baholanadi?

Ish tartibi

1. Saytga login va parolingiz bilan kiring: **emro.uz/korxona/kirish.html**
2. **6-amaliy** sahifasini oching (oxirgi slayddagi QR kod).
3. Yo‘riqnoma bo‘yicha jamoa loyihasida bosqichni bajaring, **commit** va **push** qiling.
4. PHP topshiriqlarni yeching: ▶ **Sinab ko‘rish** → ✓ **Topshirish** (yashirin testlar).
5. Loyiha havolasini (GitHub repo, PR yoki tag) topshiring — o‘qituvchi **baho** qo‘yadi.

Baholash mezonlari

- **Natija** — loyiha bosqichi mezonlar ro‘yxati bo‘yicha bajarilgan.
- **Topshiriqlar** — PHP topshiriqlar testlardan o‘tgan.
- **Kod sifati** — tushunarli nomlar, PSR-12, kichik va mazmunli commitlar.
- **Jamoaviy hissa** — har bir a’zo o‘z qismini tushuntira oladi.
- **Muddat** — muddatdan keyin topshirilgan ish “kechikkan” deb belgilanadi.

01

Mahsulotlar katalogi: ProductTable

Livewire o‘rnatish · marshrutlar · qidiruv scope · saralash

Livewire 3

ProductTable

scopeSearch

TERMINAL

O‘rnatish va komponent yaratish

```
composer require livewire/livewire
php artisan livewire:layout        # components/layouts/app.blade.php
php artisan make:livewire ProductTable
php artisan make:livewire OrderForm
php artisan make:request StoreProductRequest
php artisan lang:publish           # lang/ papkasi (keyin lang/uz)
```

▶ NATIJA

```
app/Livewire/ProductTable.php
resources/views/livewire/product-table.blade.php
app/Livewire/OrderForm.php
resources/views/livewire/order-form.blade.php
app/Http/Requests/StoreProductRequest.php
```

`livewire:layout`

to‘liq sahifali komponentlar uchun standart maket

`make:livewire`

sinf + Blade ko‘rinish juftligi

`lang:publish`

`lang/en` ni chiqaradi; `lang/uz/validation.php` ni tarjima qilib qo‘shing

ROUTES/WEB.PHP

Komponentlarni marshrutga ulash

```php
use App\Livewire\{OrderForm, ProductTable};

Route::middleware('auth')->group(function () {
    Route::get('/products', ProductTable::class)
        ->middleware('can:products.view')->name('products.index');
    Route::get('/orders/create', OrderForm::class)
        ->middleware('can:orders.create')->name('orders.create');
    Route::resource('orders', OrderController::class)
        ->only(['index', 'show']);
});
```

▶ NATIJA

```
GET /products       → ProductTable (products.view)
GET /orders/create  → OrderForm (orders.create)
GET /orders, /orders/{order} → OrderController
```

ProductTable::class

to‘liq sahifali Livewire komponenti — controller kerak emas

can:products.view

ruxsat serverda tekshiriladi (5-mavzu)

tartib

`/orders/create` — `orders/{order}` dan oldin

MODEL VA KOMPONENT

scopeSearch va xavfsiz sortBy

```php
// app/Models/Product.php
public function scopeSearch(Builder $q, string $term): Builder
{
    $term = trim($term);
    return $term === '' ? $q : $q->where(fn ($w) => $w
        ->where('name', 'ilike', "%{$term}%")
        ->orWhere('sku', 'ilike', "%{$term}%"));
}

// app/Livewire/ProductTable.php
public function sortBy(string $col): void
{
    abort_unless(in_array($col, ['sku', 'name', 'price'], true), 400);
    $this->dir = $this->sort === $col && $this->dir === 'asc'
        ? 'desc' : 'asc';
    $this->sort = $col;
    $this->resetPage();
}
```

▶ NATIJA

“Narx” bosildi: sort=price, dir=asc
“Narx” yana bosildi: sort=price, dir=desc
sortBy('password') → 400 Bad Request

scopeSearch

`Product::search($s)` yoki `->search($s)` ko‘rinishida chaqiriladi

where(fn ...)

OR shart qavs ichida — boshqa filtrlar bilan to‘g‘ri birlashadi

abort_unless(..., 400)

brauzerdan kelgan metod argumenti ham — foydalanuvchi kiritishi

resetPage()

saralash o‘zgardi → 1-sahifa

02

Hujjatlar interfeysi: buyurtmalar ro‘yxati va OrderForm

Buyurtmalar filtrlari · dinamik qatorlar · massiv validatsiyasi · servisga topshirish

OrderForm

Hujjat

Validatsiya

Filtrlar, ustunlar va so‘rov

Element	Qiymat / qoida	Laravel
Holat filtri	`draft … shipped, cancelled`	`Rule::enum(OrderStatus::class), where('status', ...)`
Sana oralig‘i	`from, to` (Y-m-d)	`whereDate('created_at', '>=', $from)`
Raqam qidiruvi	`BY-2026-000123`	`where('number', 'ilike', "%$s%")`
Summa ustuni	o‘ngga, `so‘m`, ming ajratgich	`number_format(..., 0, '.', ' ')`
Holat belgisi	rang + matn	`$order->status->label(), color()`
Eager loading	mijoz va ombor nomi	`with(['counterparty:id,name', 'warehouse:id,code'])`

Saralash standart: `created_at desc`, keyin `id desc`; sahifalash — `paginate(25)->withQueryString()`.

ORDER-FORM.BLADE.PHP

Dinamik qatorlar va xatolar

```
<form wire:submit="save">
    <select wire:model="counterpartyId" class="form-select">
        <option value="">— mijozni tanlang —</option>
        @foreach ($customers as $c)
            <option value="{{ $c->id }}">{{ $c->name }}</option>
        @endforeach
    </select>
    @foreach ($lines as $i => $line)
    <div class="row g-2" wire:key="line-{{ $i }}">
        <select wire:model="lines.{{ $i }}.product_id">…</select>
        <input type="number" wire:model.blur="lines.{{ $i }}.quantity">
        @error("lines.$i.quantity")<em>{{ $message }}</em>@enderror
        <button wire:click.prevent="removeLine({{ $i }})">✕</button>
    </div>
    @endforeach
    <button type="button" wire:click="addLine">+ Qator</button>
    <button type="submit" wire:loading.attr="disabled">Saqlash</button>
</form>
```

▶ NATIJA

```
Mijoz: “Baraka Savdo” MChJ
1) UN-001 × 10   2) GR-007 × 0 → “The lines.1.quantity field must be at least 1.”
[+ Qator]  [Saqlash] (so‘rov paytida o‘chiq)
```

`wire:submit="save"`

forma yuborilganda `save()` metodi

`lines.{{ $i }}.quantity`

massiv elementiga bog‘lanish

`@error("lines.$i.quantity")`

har bir qator uchun alohida xato

`wire:key`

qatorlar o‘chirilganda DOM adashmasligi uchun; yaxshisi — har qatorga noyob `uid`

ORDERFORM.PHP

Qator qo‘shish, o‘chirish va ma’lumotnomalar

```php
public function addLine(): void
{
    $this->lines[] = ['product_id' => null, 'quantity' => 1];
}

public function removeLine(int $i): void
{
    unset($this->lines[$i]);
    $this->lines = array_values($this->lines);
}
public function render()
{
    return view('livewire.order-form', [
        'customers' => Counterparty::query()
            ->whereIn('type', ['customer', 'both'])
            ->orderBy('name')->get(['id', 'name', 'inn']),
    ]);
}
```

▶ NATIJA

```
addLine()      → lines: 2 ta qator
removeLine(0)  → lines: [0 => ...] (indekslar qayta tartiblandi)
```

`array_values()`

indekslar 0 dan qayta — `lines.0`, `lines.1` xatolari to‘g‘ri bog‘lanadi

`whereIn('type', …)`

faqat mijoz bo‘la oladigan kontragentlar

`get(['id', 'name', 'inn'])`

kerakli ustunlar — komponent holati yengil

03

Ruxsatlar, UX va tekshirish

@can va Policy · tasdiqlash dialoglari · UX nazorat ro‘yxati · Livewire testi

@can

UX

Livewire::test

ORDERS/SHOW.BLADE.PHP

Holat va ruxsatga bog‘liq tugmalar

```blade
<x-status-badge :status="$order->status" />
@can('confirm', $order)
    <form method="POST" action="{{ route('orders.confirm', $order) }}">
        @csrf
        <button class="btn btn-sm btn-success">Tasdiqlash</button>
    </form>
@endcan
@can('cancel', $order)
    <form method="POST" action="{{ route('orders.cancel', $order) }}"
          onsubmit="return confirm('Buyurtma bekor qilinsinmi?')">
        @csrf
        <button class="btn btn-sm btn-danger">Bekor qilish</button>
    </form>
@endcan
```

▶ NATIJA

```
Menejer, holat = draft     → [Tasdiqlash] [Bekor qilish]
Menejer, holat = shipped   → tugmalar yo‘q
Omborchi                   → tugmalar yo‘q (ruxsat yo‘q)
```

`@can('confirm', $order)`

OrderPolicy: rol + holat (5-mavzu)

`<x-status-badge>`

`OrderStatus::label()` va `color()` dan foydalanadi

`onsubmit="return confirm(...)"`

xavfli amal oldidan tasdiqlash; Livewire da `wire:confirm`

Server tomonda

marshrut ham `can:` yoki `authorize()` bilan himoyalangan

LIVEWIRE TEST

Qidiruv filtrini avtomatik tekshirish

```php
use App\Livewire\ProductTable;
use App\Models\{Product, User};
use Livewire\Livewire;

it('qidiruv bo‘yicha filtrlaydi', function () {
    $user = User::factory()->create()->assignRole('omborchi');
    Product::factory()->create(['sku' => 'UN-001', 'name' => 'Un']);
    Product::factory()->create(['sku' => 'GR-007', 'name' => 'Guruch']);

    Livewire::actingAs($user)
        ->test(ProductTable::class)
        ->set('search', 'UN-0')
        ->assertSee('UN-001')
        ->assertDontSee('GR-007');
});
```

▶ NATIJA

```
PASS  Tests\Feature\ProductTableTest
✓ qidiruv bo‘yicha filtrlaydi   0.21s
Tests: 1 passed (2 assertions)
```

`Livewire::actingAs($user)`

test foydalanuvchi nomidan

`->set('search', ...)`

xossani o‘zgartirish — `wire:model` ga teng

`assertSee / assertDontSee`

render qilingan HTML ni tekshiradi

`assignRole('omborchi')`

spatie/laravel-permission (5-mavzu)

UX

Ekran sifatini tekshiring

✗ Yomon ekran

- Filtr natijasiz — bo‘sh oq maydon
- Filtr o‘zgardi, 7-sahifada qolib ketdi
- Narx `1250000.00` ko‘rinishida, chapga tekislangan
- “Saqlash” ni ikki marta bosish — ikki buyurtma
- Xato: “The given data was invalid.”

✓ Yaxshi ekran

- “Hech narsa topilmadi — filtrni tozalang”
- Filtr o‘zgarganda `resetPage()` — 1-sahifa
- `1 250 000 so‘m`, o‘ngga tekislangan
- `wire:loading.attr="disabled"` — bitta so‘rov
- Maydon ostida o‘zbekcha aniq xabar

Bu ro‘yxatni topshirishdan oldin jamoa ichida o‘zaro tekshiring (har kim boshqa a’zoning ekranini).

LOYIHA BOSQICHI

Natija va topshirish

Natija: ikki modul (mahsulotlar katalogi va buyurtmalar) uchun ishlaydigan CRUD ekranlari — filtrlar URL da saqlanadi, saralash xavfsiz, sahifalash va forma validatsiyasi (Blade + Livewire) ishlaydi, tugmalar rollarga mos ko‘rinadi.

Baholash mezonlari

✓ `ProductTable`: qidiruv (debounce), kategoriya filtri, 3 ustun bo‘yicha saralash, 20 tadan sahifalash; filtrlar URL da (`#[Url]`) va filtr o‘zgarganda 1-sahifa

✓ Saralash ustunlari oq ro‘yxat bilan tekshiriladi; `?sort=password` 500 xato bermaydi

✓ Ro‘yxatlarda N+1 yo‘q (`with(...)`), bo‘sh holat va yuklanish holati bor

✓ Mahsulot formasi: `StoreProductRequest`, `old()`, `@error`, tahrirda `Rule::unique(...)->ignore()`, saqlashdan keyin flash xabar

✓ `OrderForm`: dinamik qatorlar, `lines.*` qoidalari, saqlash `OrderService` orqali, servis xatolari formada ko‘rinadi

✓ Tugmalar `@can` / Policy bilan yashiriladi va marshrutlar serverda ham himoyalangan

✓ Xato xabarlari o‘zbek tilida (`lang/uz/validation.php`), pul `so‘m` formatida, sana `d.m.Y`

✓ Kamida 2 ta `Livewire::test()` testi o‘tadi

📤 GitHub repozitoriyangizdagi `a06-crud-ui` tegi yoki Pull Request havolasi. README da ikkala modulning ekran rasmlari, rollar bo‘yicha tugmalar jadvali va testlarni ishga tushirish buyrug‘i bo‘lsin.

Telefon kamerasi bilan skanerlang

Loyiha havolasini amaliy sahifasidagi «Loyiha bosqichi» bo‘limida topshiring.

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · Oson · 10 ball

4-topshiriq: Filtrlarni saqlovchi query string

Ro‘yxat ekranida filtr, saralash va sahifa URL da saqlanadi. Joriy parametrlar `$current` ga `$changes` qo‘llanadi (`$changes` ustun). Qiymati `null` yoki `''` bo‘lgan kalitlar olib tashlanadi; `page = 1` ham yozilmaydi (standart). Agar `$changes` da `page` dan boshqa kalit bo‘lsa va `page` ning o‘zi berilmagan bo‘lsa — `page` olib tashlanadi (filtr o‘zgardi → 1-sahifa). Kalitlar alifbo tartibida saralanib, `http_build_query` bilan satrga aylantiriladi.

Imzo: `function qs_build(array $current, array $changes): string`

Qaytaradi: Satr, masalan `'page=4&search=un'`; parametr qolmasa — `''`.

MISOL · CHAQIRUV

```
qs_build(['search' => 'un', 'page' => 3], ['page' => 4])
```

MISOL · NATIJA

```
"page=4&search=un"
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · O‘rta · 20 ball

5-topshiriq: Filtrdan xavfsiz WHERE

Filtrlar paneli qiymatlaridan parametrli SQL sharti yasang. Ruxsat etilgan kalitlar va ularning tartibi: `search` → `(name ILIKE ? OR sku ILIKE ?)` (ikki marta `%matn%`, matndagi `%` va `_` oldiga `\` qo‘yiladi), `category_id` → `category_id = ?` (musbat butun son yoki raqamli satr, int ga aylantiriladi), `is_active` → `is_active = ?` (`true`/`false`/`'1'`/`'0'`, bool ga), `price_min` → `price >= ?`, `price_max` → `price <= ?` (0 yoki musbat butun son yoki raqamli satr). Qiymati `null` yoki bo‘sh joydan iborat bo‘lgan filtrlar e’tiborsiz. Noma’lum kalit, noto‘g‘ri qiymat yoki `price_min > price_max` — `InvalidArgumentException`.

Imzo: `function filtr_where(array $f): array`

Qaytaradi: `['sql' => 'WHERE ... AND ...', 'bindings' => [...]]`; shart bo‘lmasa `['sql' => '', 'bindings' => []]`. Shartlar yuqoridagi kalitlar tartibida.

MISOL · CHAQIRUV

```
filtr_where(['search' => 'un', 'category_id' => 3])
```

MISOL · NATIJA

```
{"sql":"WHERE (name ILIKE ? OR sku ILIKE ?) AND category_id = ?","bindings":["%un%","%un%",3]}
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · O‘rta · 20 ball

6-topshiriq: Mahsulot formasini validatsiya qilish

Mahsulot formasidan kelgan `$d` massivini tekshiring va xatolar xaritasini qaytaring (har maydon uchun birinchi xato). `sku`: majburiy → `SKU majburiy`; `/^[A-Z]{2}-\d{3}$/` ga mos emas → `SKU formati: AA-000`; `$bandSku` ichida bo‘lsa → `Bu SKU band`. `name`: majburiy → `Nom majburiy`; uzunligi (belgilarda, UTF-8) 2–100 emas → `Nom 2–100 belgi`. `unit`: majburiy → `Birlik majburiy`; `dona`, `kg`, `litr`, `quti` dan biri emas → `Noma’lum birlik`. `price`: majburiy → `Narx majburiy`; int yoki `/^\d+(\.\d{1,2})?$/` satr emas (float ham qabul qilinmaydi) → `Narx formati noto‘g‘ri`; 0 dan katta emas → `Narx 0 dan katta bo‘lsin`. `min_stock` ixtiyoriy; berilsa — 0 yoki musbat int yoki raqamli satr, aks holda `Minimal qoldiq noto‘g‘ri`. Majburiy: kalit yo‘q, `null` yoki bo‘sh joydan iborat satr.

Imzo: `function mahsulot_validate(array $d, array $bandSku = []): array`

Qaytaradi: Xatolar: `['sku' => '...', 'name' => '...', 'unit' => '...', 'price' => '...', 'min_stock' => '...']` — faqat xatoli maydonlar, aynan shu tartibda; xato yo‘q — `[]`.

MISOL · CHAQIRUV	MISOL · NATIJA
`mahsulot_validate(['sku' => 'UN-001', 'name' => 'Un oliy nav', 'unit' => 'kg', 'price' => '385000.00'])`	`[]`

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · Qiyin · 30 ball

7-topshiriq: Dashboard ko‘rsatkichlari

Buyurtmalar ro‘yxati berilgan: har biri `['status' => ..., 'total' => int, 'date' => 'Y-m-d']`. Bugungi sana `$today` (`Y-m-d`). Savdo deb faqat `invoiced` va `shipped` holatidagi buyurtmalar summasi hisoblanadi. Qaytaring: `sales_today` — bugungi savdo; `orders_today` — bugungi `cancelled` bo‘lmagan buyurtmalar soni; `by_status` — barcha buyurtmalar bo‘yicha 6 ta holat soni (`draft, confirmed, reserved, invoiced, shipped, cancelled` tartibida, 0 lari bilan); `series` — `today - 6` dan `today` gacha 7 kunlik savdo (`'Y-m-d' => summa`, o‘sish tartibida). Noma’lum holat — `DomainException`; `$today` haqiqiy `Y-m-d` sana bo‘lmasa (masalan `2026-02-30`) — `InvalidArgumentException`.

Imzo: `function dashboard_kpi(array $orders, string $today): array`

Qaytaradi: `['sales_today' => int, 'orders_today' => int, 'by_status' => [6 ta holat => int], 'series' => [7 ta sana => int]]`.

MISOL · CHAQIRUV

```
dashboard_kpi([['status' => 'shipped', 'total' => 500000, 'date' => '2026-11-10'], ['status' => 'draft', 'total' => 90000, 'date' => '2026-11-10'], ['status' => 'invoiced', 'total' => 250000, 'date' => '2026-11-08']], '2026-11-10')
```

MISOL · NATIJA

```
{"sales_today":500000,"orders_today":2,"by_status":{"draft":1,"confirmed":0,"reserved":0,"invoiced":1,"shipped":1,"cancelled":0},"series":{"2026-11-04":0,"2026-11-05":0,"2026-11-06":0,"2026-11-07":0,"2026-11-08":250000,"2026-11-09":0,"2026-11-10":500000}}
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · Qiyin · 30 ball

8-topshiriq: Ma’lumotlar jadvali: filtr + saralash + sahifa

Xotiradagi jadval qatorlari `[sku, nom, narx, qoldiq]` ko‘rinishida. So‘rov `$q` ixtiyoriy kalitlari: `search` (sku yoki nomda katta-kichik harfsiz qidiruv, `trim`), `sort` (`sku`, `name`, `price`, `qty`, boshida `-` bo‘lsa kamayish; standart `sku`), `page` (standart 1), `per_page` (standart 10, 1..50). Teng qiymatlar har doim `sku` bo‘yicha o‘sish tartibida joylashadi. Ruxsat etilmagan `sort`, `page < 1` yoki `per_page` oraliqdan tashqarida — `InvalidArgumentException`. Sahifa oxirgisidan katta bo‘lsa — `data` bo‘sh.

Imzo: `function grid(array $rows, array $q): array`

Qaytaradi: `['data' => [sahifadagi sku lar], 'meta' => ['total' => int, 'current_page' => int, 'last_page' => int]]`; `last_page` kamida 1.

MISOL · CHAQIRUV

```
grid([['UN-001', 'Un', 385, 40], ['GR-007', 'Guruch', 41, 0], ['SH-010', 'Shakar', 41, 12], ['YG-002', 'Yog', 120, 5], ['UN-002', 'Un 1-nav', 300, 8]], ['sort' => '-price', 'per_page' => 2])
```

MISOL · NATIJA

```
{"data":["UN-001","UN-002"],"meta":{"total":5,"current_page":1,"last_page":3}}
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

TOPSHIRIQLAR VA TOPSHIRISH

6-amaliy: 8 ta PHP topshiriq

1 **Qoldiq belgisi (badge)**
Oson · 10 ball · ma’ruzada

2 **Saralash parametrini tahlil qilish**
O‘rta · 20 ball · ma’ruzada

3 **Sahifalash havolalari oynasi**
O‘rta · 20 ball · ma’ruzada

4 **Filtrlarni saqlovchi query string**
Oson · 10 ball

5 **Filtrdan xavfsiz WHERE**
O‘rta · 20 ball

6 **Mahsulot formasini validatsiya qilish**
O‘rta · 20 ball

7 **Dashboard ko‘rsatkichlari**
Qiyin · 30 ball

8 **Ma’lumotlar jadvali: filtr + saralash + sahifa**
Qiyin · 30 ball

Telefon kamerasi bilan skanerlang

PHP kod saytda yoziladi va server tomonidan avtomatik tekshiriladi.

Loyiha havolasini ham shu sahifada topshirasiz; o‘qituvchi baho qo‘yadi.