4

TO‘RTINCHI AMALIY MASHG‘ULOT

Korxona ilovalarini dasturlash · 2 soat · 4-mavzu

Buyurtma – rezerv – hisob-faktura oqimi

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

Servis qatlami | DB::transaction | lockForUpdate | Invariantlar | Holat mashinasi

MASHG‘ULOT MAQSADI

Amaliy mashg‘ulot oxirida siz quyidagilarni bajara olasiz…

1. **Yozish** — `StockService`: kirim, rezerv, bo‘shatish, jo‘natish — har biri qulf va jurnal bilan
2. **Qurish** — `OrderService` da `confirm → reserve → invoice → ship` va `cancel` oqimini atomar tranzaksiyalarda
3. **Hisoblash** — `InvoiceService` da subtotal, QQS 12 % va jami summani butun tiyinda
4. **Himoyalash** — invariantlarni domen istisnolari va bazadagi CHECK cheklovi bilan
5. **Tekshirish** — oqimni Pest testi va jurnal bilan qoldiqni solishtiruvchi SQL so‘rov orqali

Qanday bajariladi va qanday baholanadi?

Ish tartibi

1. Saytga login va parolingiz bilan kiring: **emro.uz/korxona/kirish.html**
2. **4-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

StockService va ombor harakatlari jurnali

Fayllar · qulflangan qator · harakat yozish · CHECK

StockService

stock_movements

lockForUpdate

TERMINAL

Fayllarni generatorlar bilan yaratish

```
php artisan make:enum Enums/OrderStatus --string
php artisan make:class Services/StockService
php artisan make:class Services/InvoiceService
php artisan make:exception InsufficientStockException
php artisan make:exception InvalidStatusTransitionException
php artisan make:migration add_checks_to_stocks_table
php artisan make:test OrderFlowTest --pest
php artisan migrate
```

▶ NATIJA

```
INFO  Enum [app/Enums/OrderStatus.php] created successfully.
INFO  Class [app/Services/StockService.php] created successfully.
INFO  Exception [app/Exceptions/InsufficientStockException.php]
created successfully.
INFO  Test [tests/Feature/OrderFlowTest.php] created successfully.
```

`make:enum, make:class`

Laravel 11 da qo‘shilgan generatorlar

`--string`

satr qiymatli (backed) enum: `enum OrderStatus: string`

`make:exception`

yaratilgandan keyin `extends DomainException` ga o‘zgartiring

STOCKSERVICE

Rezerv va bo‘shatish: qulf → tekshiruv → yozuv

```php
public function reserve(int $warehouseId, int $productId,
    int $qty, Model $doc): void
{
    $stock = $this->lockRow($warehouseId, $productId);
    $available = $stock->quantity - $stock->reserved;
    if ($qty <= 0 || $available < $qty) {
        throw new InsufficientStockException(
            $productId, $qty, $available);
    }
    $stock->increment('reserved', $qty);
    $this->log('reserve', $stock, $qty, $doc);
}
public function release(int $warehouseId, int $productId,
    int $qty, Model $doc): void
{
    $stock = $this->lockRow($warehouseId, $productId);
    // reserved >= qty tekshiruvi, decrement, log('release', ...)
}
```

▶ NATIJA

```
select * from "stocks" where "warehouse_id" = 1 and "product_id" =
7 limit 1 for update
update "stocks" set "reserved" = "reserved" + 4 where "id" = 15
insert into "stock_movements" ("type", "quantity",
"document_type", ...) values ('reserve', 4, 'App\Models\Order',
...)
```

`lockRow()`

`where(...)->lockForUpdate()->firstOrFail()` — har bir metodda bir xil

`Model $doc`

har qanday hujjat: `Order`, `Invoice`, inventarizatsiya akti

Tartib

qulf → tekshiruv → `stocks` → jurnal

`release()`

o‘zingiz yozing: `reserved < $qty` bo‘lsa — `DomainException`

JURNAL

Harakat yozish va jo‘natish

```php
private function log(string $type, Stock $stock, int $qty,
    Model $doc): void
{
    StockMovement::create([
        'warehouse_id' => $stock->warehouse_id,
        'product_id' => $stock->product_id,
        'type' => $type,
        'quantity' => $qty,
        'document_type' => $doc->getMorphClass(),
        'document_id' => $doc->getKey(),
        'user_id' => auth()->id(),
    ]);
}
```

▶ NATIJA

```
stock_movements:
 id | type    | qty | document_type     | document_id | user_id
 41 | reserve |   4 | App\Models\Order  |         124 |       3
 42 | out     |   4 | App\Models\Order  |         124 |       5
```

`getMorphClass()`

polimorf tur; `Relation::enforceMorphMap()` bilan qisqa nom (`order`) berish mumkin

`auth()->id()`

konsol/job da `null` bo‘lishi mumkin — ustun nullable

`const UPDATED_AT = null`

jadvalda faqat `created_at` bor

`ship()`

`quantity` va `reserved` ni birga kamaytiradi, `log('out', ...)`

NAZORAT

StockService metodlari va invariantlar

Metod	stocks o‘zgarishi	Jurnal	Rad etish sharti
`receive($w, $p, $n, $doc)`	quantity += n	`in`	n ≤ 0
`reserve(...)`	reserved += n	`reserve`	n ≤ 0 yoki mavjud < n
`release(...)`	reserved −= n	`release`	reserved < n
`ship(...)`	quantity −= n, reserved −= n	`out`	reserved < n
`adjust(...)`	quantity ± n	`adjust`	natijada quantity < reserved

02

OrderService va InvoiceService: atomar oqim

Holat oʻtishi · rezerv · hisob-faktura · joʻnatish · bekor qilish

OrderService

InvoiceService

DB::transaction

ORDERSERVICE

cancel(): holat va rezervni birga qaytarish

```php
public function cancel(Order $order): Order
{
    return DB::transaction(function () use ($order) {
        $order = Order::lockForUpdate()->findOrFail($order->id);
        $wasReserved = $order->status === OrderStatus::Reserved;
        $this->moveTo($order, OrderStatus::Cancelled);

        if ($wasReserved) {
            $items = $order->items()->orderBy('product_id')->get();
            foreach ($items as $item) {
                $this->stock->release($order->warehouse_id,
                    $item->product_id, $item->quantity, $order);
            }
        }
        return $order;
    }, attempts: 3);
}
```

▶ NATIJA

```
BY-2026-000124: reserved → cancelled
stock_movements: release UN-001 × 4, release UN-007 × 2
stocks: reserved 4 → 0, 2 → 0
```

$wasReserved

holat `moveTo()` dan oldin eslab qolinadi

`=== OrderStatus::Reserved`

enum bilan taqqoslash, satr bilan emas

shipped bo‘lsa?

`moveTo()` `InvalidStatusTransitionException` tashlaydi — hech narsa o‘zgarmaydi

`orderBy('product_id')`

`reserve()` dagi tartib bilan bir xil

INVOICESERVICE

Hisob-faktura: QQS butun tiyinda

```php
public function createForOrder(Order $order): Invoice
{
    if ($order->invoice()->exists()) {
        throw new DomainException('Hisob-faktura allaqachon bor');
    }
    $subtotal = Money::toTiyin($order->total);
    $vat = intdiv($subtotal * 12 + 50, 100);
    return Invoice::create([
        'number' => $this->nextNumber(),         // HF-2026-000045
        'order_id' => $order->id,
        'counterparty_id' => $order->counterparty_id,
        'issued_at' => now(),
        'due_at' => now()->addDays(10),
        'subtotal' => Money::fromTiyin($subtotal),
        'vat_rate' => 12, 'vat_amount' => Money::fromTiyin($vat),
        'total' => Money::fromTiyin($subtotal + $vat),
    ]);
}
```

▶ NATIJA

```
HF-2026-000045  BY-2026-000124
subtotal   2 650 000.00
QQS 12 %     318 000.00
jami       2 968 000.00 so‘m
```

`invoice()->exists()`

`hasOne` munosabati; qo‘shimcha himoya — unique indeks

`Money::toTiyin()`

`decimal:2` satridan butun tiyinga

`intdiv(... + 50, 100)`

yarmidan yuqoriga yaxlitlash, `float` siz

`nextNumber()`

sequence yoki qulflangan hisoblagich — `max(id)+1` emas

TINKER

Oqimni qo‘lda sinash

1

Tayyorlash

`$o = Order::factory()->create()` + 2 qator; omborga `receive()` bilan 10 tadan kirim.

2

Tasdiqlash va rezerv

`$s = app(OrderService::class); $s->confirm($o); $s->reserve($o);`

3

Qoldiqni ko‘rish

`Stock::all(['product_id', 'quantity', 'reserved'])` — `reserved` oshgan.

4

Hisob-faktura va jo‘natish

`$s->invoice($o); $s->ship($o);` — `quantity` kamaydi, `reserved` 0.

5

Taqiqlangan o‘tish

`$s->cancel($o)` → `InvalidStatusTransitionException`.

6

Jurnal

`StockMovement::where('document_id', $o->id)->pluck('type')` — in, reserve, out.

03

Tekshirish: test va jurnal solishtiruvi

Pest testi · ROLLBACK ni isbotlash · qoldiqni jurnaldan qayta hisoblash

Pest | Reconciliation | SQL

PEST

Salbiy test: qoldiq yetmasa — hech narsa o‘zgarmaydi

```php
it('qoldiq yetmasa rezerv ROLLBACK bo‘ladi', function () {
    $order = Order::factory()
        ->has(OrderItem::factory()->state(['quantity' => 10]), 'items')
        ->create(['status' => OrderStatus::Confirmed]);
    $item = $order->items->first();
    Stock::factory()->create([
        'warehouse_id' => $order->warehouse_id,
        'product_id' => $item->product_id,
        'quantity' => 6, 'reserved' => 0,
    ]);
    expect(fn () => app(OrderService::class)->reserve($order))
        ->toThrow(InsufficientStockException::class);

    expect($order->fresh()->status)->toBe(OrderStatus::Confirmed)
        ->and(Stock::first()->reserved)->toBe(0)
        ->and(StockMovement::count())->toBe(0);
});
```

▶ NATIJA

```
PASS  Tests\Feature\OrderFlowTest
✓ qoldiq yetmasa rezerv ROLLBACK bo‘ladi                0.21s
Tests:    1 passed (4 assertions)
```

`->has(..., 'items')`

fabrika munosabat bilan birga yaratadi

`toThrow()`

closure istisno tashlashini kutadi

`fresh()`

bazadan qayta o‘qish — xotiradagi obyektga ishonmaymiz

`->and(...)`

bir nechta kutilmani bitta zanjirda

SOLISHTIRUV

stocks va jurnal mos keladimi?

```sql
SELECT s.warehouse_id, s.product_id, s.quantity, j.qty
FROM stocks s
LEFT JOIN (
    SELECT warehouse_id, product_id,
           SUM(CASE type WHEN 'in' THEN quantity
                         WHEN 'adjust' THEN quantity
                         WHEN 'out' THEN -quantity
                         ELSE 0 END) AS qty
    FROM stock_movements
    GROUP BY warehouse_id, product_id
) j ON j.warehouse_id = s.warehouse_id
   AND j.product_id = s.product_id
WHERE s.quantity <> COALESCE(j.qty, 0);
```

▶ NATIJA

```
 warehouse_id | product_id | quantity | qty
--------------+------------+----------+-----
(0 rows)
```

CASE type

`in`, `adjust` qo‘shiladi, `out` ayiriladi; `reserve`/`release` quantity ga ta’sir qilmaydi

LEFT JOIN

jurnalsiz qoldiq qatorlari ham tekshiriladi

`(0 rows)`

kutilgan natija; qator chiqsa — kimdir servisni chetlab o‘tgan

`reserved`

xuddi shunday: `reserve – release – out`

LOYIHA BOSQICHI

Natija va topshirish

Natija: servis qatlamida ishlaydigan buyurtma oqimi (draft → confirmed → reserved → invoiced → shipped, bekor qilish bilan); har bir qoldiq o‘zgarishi qulflangan qatorda, atomar tranzaksiyada bajariladi va ombor harakatlari jurnaliga yoziladi.

Baholash mezonlari

✓ `OrderService` da `confirm`, `reserve`, `invoice`, `ship`, `cancel` metodlari bor, har biri `DB::transaction` ichida

✓ Taqiqlangan o‘tish (`shipped → cancelled`, `draft → invoiced`) `InvalidStatusTransitionException` beradi

✓ `StockService` qatorni `lockForUpdate()` bilan oladi, qatorlar `product_id` tartibida qulflanadi

✓ Qoldiq yetmasa `InsufficientStockException`, holat va `stocks` o‘zgarmaydi (Pest testi bilan isbotlangan)

✓ Har bir qoldiq o‘zgarishi `stock_movements` ga hujjat va foydalanuvchi bilan yoziladi; solishtiruv so‘rovi 0 qator qaytaradi

✓ Bazada `CHECK (reserved >= 0 AND reserved <= quantity)` va `quantity >= 0` cheklovlari bor

✓ Hisob-faktura QQS 12 % bilan butun tiyinda hisoblanadi, bir buyurtmaga bitta hisob-faktura

✓ Controllerlarda `Stock::`, `DB::` yoki holat `if` lari yo‘q — faqat servis chaqiruvi

⮉ `feature/order-flow` tarmog‘idan ochilgan Pull Request havolasi. PR tavsifida: oqimni tinker yoki test bilan sinash natijasi (ekran rasmi), solishtiruv so‘rovining bo‘sh natijasi va jamoadoshingizning kod tahlili izohi bo‘lsin.

Telefon kamerasi bilan skanerlang

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

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · Oson · 10 ball

4-topshiriq: Hisob-faktura summalari

Hisob-faktura qatorlari `[miqdor, narx_tiyin]` ko‘rinishida (butun sonlar, narx — tiyinda). `subtotal` — qatorlar yig‘indisi, `vat_amount` — QQS (`qqs` foiz, standart 12), yarmidan yuqoriga yaxlitlanadi: `intdiv(subtotal × qqs + 50, 100)`. `float` ishlatmang. Bo‘sh ro‘yxat — `DomainException` (bo‘sh buyurtmaga hisob-faktura chiqmaydi); `miqdor ≤ 0` yoki `narx < 0` — `InvalidArgumentException`.

Imzo: `function hisob_faktura(array $qatorlar, int $qqs = 12): array`

Qaytaradi: `['subtotal' => int, 'vat_amount' => int, 'total' => int]` — hammasi tiyinda.

MISOL · CHAQIRUV

```
hisob_faktura([[2, 1500000], [1, 700000]])
```

MISOL · NATIJA

```
{"subtotal":3700000,"vat_amount":444000,"total":4144000}
```

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: Jurnaldan qoldiqni tiklash

Ombor harakatlari jurnali `[tur, miqdor]` ro‘yxati (bo‘sh qoldiqdan boshlanadi). Turlar: `in` (quantity += n), `reserve` (reserved += n, faqat mavjud ≥ n), `release` (reserved −= n, faqat reserved ≥ n), `out` (quantity −= n va reserved −= n, faqat reserved ≥ n), `adjust` (quantity += n, n manfiy bo‘lishi mumkin, natijada quantity ≥ reserved). `adjust` dan boshqa turlarda `n ≤ 0`, `adjust` da `n = 0` yoki noma’lum tur — `InvalidArgumentException`. Invariant buzilsa — `DomainException`.

Imzo: `function jurnal_qoldiq(array $harakatlar): array`

Qaytaradi: `['quantity' => int, 'reserved' => int, 'available' => int]`.

MISOL · CHAQIRUV

```
jurnal_qoldiq([['in', 10], ['reserve', 4], ['out', 3]])
```

MISOL · NATIJA

```
{"quantity":7,"reserved":1,"available":6}
```

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: Hujjatning keyingi raqami

Hujjat raqami formati: `PREFIKS-YYYY-NNNNNN` (masalan, `BY-2026-000123`, `HF-2026-000045`). Oxirgi raqam (`null` bo‘lishi mumkin) va joriy yil berilgan. Keyingi raqamni qaytaring: `null` yoki o‘tgan yil raqami bo‘lsa — yangi yil `000001` dan boshlanadi. Oxirgi raqam formatga yoki prefiksga mos kelmasa — `InvalidArgumentException`; oxirgi raqam yili joriy yildan katta bo‘lsa — `DomainException`; `999999` dan oshsa — `OverflowException`.

Imzo: `function keyingi_raqam(string $prefiks, int $yil, ?string $oxirgi): string`

Qaytaradi: Satr, masalan `'BY-2026-000124'`.

MISOL · CHAQIRUV	MISOL · NATIJA
`keyingi_raqam('BY', 2026, 'BY-2026-000123')`	`"BY-2026-000124"`

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: Ombor: harakatlar va jurnal

`Ombor` sinfi `StockService` ning soddalashtirilgan modeli. Metodlar: `kirim`, `rezerv`, `bekor` (rezervni bo‘shatish), `jonatish` — har biri `(string $sku, int $miqdor, string $hujjat): void`; `mavjud(string $sku): int`; `holat(string $sku): array` → `['quantity' => int, 'reserved' => int]` (noma’lum SKU — nollar); `jurnal(): array`. `miqdor ≤ 0` — `InvalidArgumentException`. Rezerv mavjuddan, bo‘shatish va jo‘natish rezervdan oshsa — `DomainException`, holat va jurnal o‘zgarmaydi. Jo‘natish `quantity` va `reserved` ni birga kamaytiradi.

Imzo: `final class Ombor`

Qaytaradi: Jurnal — satrlar ro‘yxati, faqat muvaffaqiyatli amallar: `'tur:sku:miqdor:hujjat'`, tur — `in`, `reserve`, `release`, `out`.

MISOL · CHAQIRUV

```
(function () { $o = new Ombor(); $o->kirim('UN-001', 10, 'KIR-1'); $o->rezerv('UN-001', 4, 'BY-1'); return $o->mavjud('UN-001'); })()
```

MISOL · NATIJA

```
6
```

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: Ko‘p qatorli buyurtmani atomar rezerv qilish

Qoldiq: `['SKU' => ['quantity' => int, 'reserved' => int], ...]`, buyurtma qatorlari: `[sku, miqdor]` (bir SKU bir necha marta kelishi mumkin — miqdorlar qo‘shiladi). Avval **barcha** miqdorlarni tekshiring (`≤ 0` — `InvalidArgumentException`). So‘ng SKU larni alifbo tartibida (qulflar tartibi!) rezerv qiling. Biror SKU noma’lum yoki mavjud qoldiq yetmasa — `InsufficientStockException` (`DomainException` dan meros, starter da berilgan), hech narsa o‘zgarmaydi.

Imzo: `function rezerv_barcha(array $qoldiq, array $qatorlar): array`

Qaytaradi: `['qoldiq' => yangilangan massiv (kalitlar tartibi asl holicha), 'harakatlar' => ['reserve:SKU:miqdor', ...]]` — harakatlar SKU bo‘yicha o‘sish tartibida.

MISOL · CHAQIRUV

```
rezerv_barcha(['UN-002' => ['quantity' => 5, 'reserved' => 0], 'UN-001' => ['quantity' => 10, 'reserved' => 2]], [['UN-002', 3], ['UN-001', 4], ['UN-002', 1]])
```

MISOL · NATIJA

```
{"qoldiq":{"UN-002":{"quantity":5,"reserved":4},"UN-001":{"quantity":10,"reserved":6}},"harakatlar":["reserve:UN-001:4","reserve:UN-002:4"]}
```

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

4-amaliy: 8 ta PHP topshiriq

1 **Hammasi yoki hech narsa**
Oson · 10 ball · ma’ruzada

2 **Rezerv va mavjud qoldiq**
Oson · 10 ball · ma’ruzada

3 **Buyurtma holatlari zanjiri**
O‘rta · 20 ball · ma’ruzada

4 **Hisob-faktura summalari**
Oson · 10 ball

5 **Jurnaldan qoldiqni tiklash**
O‘rta · 20 ball

6 **Hujjatning keyingi raqami**
O‘rta · 20 ball

7 **Ombor: harakatlar va jurnal**
Qiyin · 30 ball

8 **Ko‘p qatorli buyurtmani atomar rezerv qilish**
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.