5

BESHINCHI AMALIY MASHG‘ULOT

Korxona ilovalarini dasturlash · 2 soat · 5-mavzu

Foydalanuvchilar, rollar va audit jurnali

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

Autentifikatsiya | RBAC | Policy | Audit | OWASP

MASHG‘ULOT MAQSADI

Amaliy mashg‘ulot oxirida siz quyidagilarni bajara olasiz…

1. **O‘rnatish** — spatie/laravel-permission paketini va uning jadvallarini
2. **Yaratish** — admin, menejer, omborchi, buxgalter rollari va 14 ta modul ruxsatini seeder orqali
3. **Himoyalash** — marshrutlarni `permission:` middleware, buyurtmalarni `OrderPolicy` bilan
4. **Moslashtirish** — menyu va tugmalarni `@can` / `@role` direktivalari bilan rolga qarab
5. **Yozish** — mahsulot, buyurtma va login hodisalari uchun audit jurnalini va uni ko‘rish sahifasini

Qanday bajariladi va qanday baholanadi?

Ish tartibi

1. Saytga login va parolingiz bilan kiring: **emro.uz/korxona/kirish.html**
2. **5-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

Paket, rollar va ruxsatlar

composer · migratsiya · HasRoles · seeder · test foydalanuvchilar

spatie/laravel-permission

Seeder

HasRoles

TERMINAL

Paketni o‘rnatish

```
git switch -c feature/rbac-audit
composer require spatie/laravel-permission
php artisan vendor:publish \
    --provider="Spatie\Permission\PermissionServiceProvider"
php artisan migrate
php artisan make:seeder RolesAndPermissionsSeeder
php artisan make:policy OrderPolicy --model=Order
php artisan migrate:fresh --seed
```

▶ NATIJA

```
INFO  Publishing assets.
  config/permission.php ....................... DONE
  database/migrations/..._create_permission_tables.php ... DONE
INFO  Policy [app/Policies/OrderPolicy.php] created successfully.
```

`vendor:publish`

`config/permission.php` va jadvallar migratsiyasini loyihaga nusxalaydi

`make:policy --model`

CRUD metodlari tayyor Policy skeletini yaratadi

`migrate:fresh --seed`

faqat lokal bazada: hammasini qayta yaratib, seeder larni ishga tushiradi

MODEL

User: HasRoles va maxfiy maydonlar

```php
namespace App\Models;

use Illuminate\Foundation\Auth\User as Authenticatable;
use Spatie\Permission\Traits\HasRoles;

class User extends Authenticatable
{
    use HasRoles;

    protected $fillable = ['name', 'email', 'password', 'is_active'];
    protected $hidden = ['password', 'remember_token'];

    protected function casts(): array
    {
        return ['password' => 'hashed', 'is_active' => 'boolean'];
    }
}
```

▶ NATIJA

```
>>> User::firstWhere('email', 'menejer@omborpro.test')
...   ->getPermissionNames()
= ["orders.view", "orders.create", "orders.confirm",
   "orders.cancel", "products.view", "stock.view", "reports.view"]
```

HasRoles

`assignRole`, `hasRole`, `getRoleNames`, `getPermissionNames` va boshqalar

`$hidden`

`toArray()`/JSON da parol va token chiqmaydi

`'hashed'`

parol saqlanishdan oldin avtomatik xeshlanadi

`casts()`

Laravel 12 uslubi — `$casts` xossasi o‘rniga metod

TEST FOYDALANUVCHILAR

Seeder dan keyin bazada

E-mail	Rol	is_active	Kutilgan xatti-harakat
admin@omborpro.test	admin	true	hamma modul, foydalanuvchilar va audit
menejer@omborpro.test	menejer	true	buyurtmalar: yaratish, tasdiqlash, bekor qilish
omborchi@omborpro.test	omborchi	true	ombor harakatlari, mahsulot katalogi
buxgalter@omborpro.test	buxgalter	true	hisob-faktura, to‘lovlar, hisobotlar
blocked@omborpro.test	menejer	false	login rad etiladi

02

Marshrutlar, Policy va interfeys

middleware aliaslari · marshrut guruhlari · OrderPolicy · @can

Middleware

Policy

@can

BOOTSTRAP/APP.PHP

spatie middleware aliaslari

```php
return Application::configure(basePath: dirname(__DIR__))
    ->withRouting(
        web: __DIR__ . '/../routes/web.php',
        commands: __DIR__ . '/../routes/console.php',
    )
    ->withMiddleware(function (Middleware $middleware) {
        $middleware->alias([
            'role' => RoleMiddleware::class,
            'permission' => PermissionMiddleware::class,
            'role_or_permission' => RoleOrPermissionMiddleware::class,
        ]);
    })
    ->withExceptions(function (Exceptions $exceptions) {
        //
    })
    ->create();
```

▶ NATIJA

```
php artisan route:list --path=invoices
GET|HEAD  invoices ......... invoices.index › InvoiceController@index
          ⇂ web, auth, permission:invoices.view
```

RoleMiddleware

to‘liq nomi `Spatie\Permission\Middleware\RoleMiddleware` — faylning boshida `use` bilan

`alias([...])`

marshrutlarda qisqa nom bilan ishlatish uchun

`withExceptions`

xatolarni sozlash — 403 sahifasini shu yerda moslash mumkin

ROUTES/WEB.PHP

Modullar bo‘yicha himoyalangan guruhlar

```php
Route::middleware('auth')->group(function () {
    Route::middleware('permission:orders.view')->group(function () {
        Route::resource('orders', OrderController::class);
        Route::post('orders/{order}/confirm', [OrderController::class,
            'confirm'])->name('orders.confirm');
    });
    Route::middleware('permission:stock.view')
        ->get('stock', [StockController::class, 'index']);
    Route::middleware('permission:invoices.view')
        ->resource('invoices', InvoiceController::class);
    Route::middleware('permission:audit.view')
        ->get('audit', [AuditLogController::class, 'index'])
        ->name('audit.index');
});
```

▶ NATIJA

```
omborchi  GET /invoices      → 403 Forbidden
buxgalter GET /invoices      → 200 OK
menejer   GET /audit         → 403 Forbidden
mehmon    GET /orders        → 302 /login
```

`middleware('auth')`

login qilmagan — login sahifasiga yo‘naltirish

`permission:orders.view`

butun modulga kirish; aniq amallar Policy da

`Route::resource`

CRUD marshrutlari bitta qatorda — guruh middleware i hammasiga qo‘llanadi

BLADE

Menyu va tugmalar rolga qarab

```blade
<nav>
    @can('orders.view')
        <a href="{{ route('orders.index') }}">Buyurtmalar</a>
    @endcan
    @can('invoices.view')
        <a href="{{ route('invoices.index') }}">Hisob-fakturalar</a>
    @endcan
    @role('admin')
        <a href="{{ route('audit.index') }}">Audit jurnali</a>
    @endrole
</nav>

@can('confirm', $order)
    <form method="POST" action="{{ route('orders.confirm', $order) }}">
        @csrf
        <button type="submit">Tasdiqlash</button>
    </form>
@endcan
```

▶ NATIJA

```
menejer  → Buyurtmalar · [Tasdiqlash]
buxgalter → Buyurtmalar · Hisob-fakturalar
admin    → Buyurtmalar · Hisob-fakturalar · Audit jurnali · [Tasdiqlash]
```

`@can('orders.view')`

spatie ruxsati — Gate orqali tekshiriladi

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

Policy metodi — obyekt va holat bilan

`@role('admin')`

spatie direktivasi; lekin imkon qadar ruxsatni tekshiring

`@csrf`

POST forma uchun majburiy token

QO‘LDA TEKSHIRISH

Har bir rol bilan sinov

1

Kirish

Har bir test foydalanuvchi bilan login; bloklangan foydalanuvchi kira olmasligini tekshiring

2

Menyu

Faqat ruxsat etilgan bandlar ko‘rinadi

3

Manzil satri

Yashirilgan sahifani URL bilan ochish — 403

4

Obyekt

Boshqa menejerning qoralamasini `/orders/{id}/edit` orqali ochish — 403

5

Natija

Rol × sahifa jadvalini PR tavsifiga yozish

03

Audit jurnali

migratsiya · AuditLog modeli · Auditable trait · login hodisalari · ko‘rish sahifasi

audit_logs

Auditable

Login events

MIGRATSIYA

audit_logs jadvali

```php
Schema::create('audit_logs', function (Blueprint $table) {
    $table->id();
    $table->foreignId('user_id')->nullable()->constrained();
    $table->string('action', 40);
    $table->nullableMorphs('auditable');
    $table->jsonb('old_values')->nullable();
    $table->jsonb('new_values')->nullable();
    $table->ipAddress('ip')->nullable();
    $table->timestamp('created_at')->useCurrent()->index();
});
```

▶ NATIJA

```
INFO  Running migrations.
  2026_10_27_090000_create_audit_logs_table ........ 18ms DONE
\d audit_logs → ip: inet, old_values: jsonb,
  index audit_logs_auditable_type_auditable_id_index
```

`nullableMorphs('auditable')`

`auditable_type`, `auditable_id` va ikkalasiga birgalikdagi indeks

`jsonb`

PostgreSQL da JSON ichidan qidirish va indekslash mumkin

`ipAddress`

PostgreSQL da `inet` turi

`updated_at` yo‘q

yozuv faqat qo‘shiladi

AuditLog: faqat qo‘shish

```php
class AuditLog extends Model
{
    const UPDATED_AT = null;

    protected $guarded = ['id'];

    protected $casts = [
        'old_values' => 'array',
        'new_values' => 'array',
    ];

    protected static function booted(): void
    {
        $deny = fn () => throw new LogicException('Audit o‘zgarmaydi');
        static::updating($deny);
        static::deleting($deny);
    }
}
```

▶ NATIJA

```
>>> AuditLog::first()->update(['action' => 'x'])
LogicException  Audit o‘zgarmaydi
>>> AuditLog::latest('id')->first()->action
= "login_failed"
```

`UPDATED_AT = null`

Eloquent `updated_at` ni yozishga urinmaydi — ustun yo‘q

`$casts 'array'`

`jsonb` ustunlar PHP massivi sifatida o‘qiladi va yoziladi

throw ifodasi

PHP 8 da `fn () => throw ...` mumkin — istisno `update()`/`delete()` ni to‘xtatadi

TOPSHIRISHDAN OLDIN

Audit: tekshiruv ro‘yxati

✗ Qabul qilinmaydi

- Jurnalda parol xeshi yoki `remember_token` ko‘rinadi
- Narx ommaviy `update()` bilan o‘zgartirilgan va iz qolmagan
- `/audit` sahifasini menejer ham ocha oladi
- Audit yozuvini tinker da `update()` qilish mumkin

✓ Qabul qilinadi

- `Product`, `Order`, `Invoice` da `created`/`updated`/`deleted` yoziladi
- `login` va `login_failed` IP bilan yoziladi
- `/audit` faqat `audit.view` (admin) uchun, filtr va sahifalash bilan
- `AuditLog` o‘zgartirilganda `LogicException`

LOYIHA BOSQICHI

Natija va topshirish

Natija: OmborPro da to‘rtta rol va 14 ta modul ruxsati seeder orqali yaratilgan, marshrutlar va buyurtma amallari middleware hamda `OrderPolicy` bilan himoyalangan, asosiy modellardagi o‘zgarishlar va login urinishlari o‘zgartirib bo‘lmaydigan audit jurnaliga yoziladi.

Baholash mezonlari

✓ `spatie/laravel-permission` o‘rnatilgan, `User` modelida `HasRoles`; `RolesAndPermissionsSeeder` takroriy ishga tushirilganda ham bir xil natija beradi.

✓ 4 ta rol (admin, menejer, omborchi, buxgalter) va 14 ta `modul.amal` ruxsati ma’ruzadagi matritsaga mos.

✓ Har bir modul marshrut guruhi `auth` va `permission:` middleware bilan himoyalangan; yashirilgan sahifa URL orqali ochilganda 403.

✓ `OrderPolicy` (`update`, `confirm`, `cancel`) ruxsat va buyurtma holatini tekshiradi; controller da `Gate::authorize` chaqiriladi.

✓ Menyu va tugmalar `@can` bilan rolga moslashgan; barcha POST formalarda `@csrf`.

✓ `audit_logs` ga `Product`, `Order`, `Invoice` o‘zgarishlari va `login`/`login_failed` yoziladi; parol va tokenlar yo‘q.

✓ `AuditLog` yozuvini o‘zgartirish yoki o‘chirish `LogicException` beradi; `/audit` sahifasi faqat `audit.view` uchun.

✓ README da test foydalanuvchilar jadvali va rol × sahifa tekshiruvi natijalari bor.

⬆ `feature/rbac-audit` tarmog‘idan `main` ga ochilgan Pull Request havolasini yuboring. PR tavsifida rol × sahifa natijalari jadvali va `/audit` sahifasining skrinshoti (narx o‘zgarishi va `login_failed` yozuvi ko‘ringan) bo‘lsin.

Telefon kamerasi bilan skanerlang

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

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · Oson · 10 ball

4-topshiriq: Ruxsatlarni modullar bo‘yicha guruhlash

Admin panelda rol ruxsatlari modullar bo‘yicha jadval ko‘rinishida chiqadi. `modul.amal` ko‘rinishidagi ruxsat nomlari ro‘yxatini modul → amallar lug‘atiga aylantiring. Nom faqat kichik lotin harflari va `_` dan iborat ikki qismdan (nuqta bilan ajratilgan) tashkil topishi kerak, aks holda `InvalidArgumentException`. Takrorlar olib tashlanadi.

Imzo: `function modul_ruxsatlari(array $ruxsatlar): array`

Qaytaradi: `['modul' => ['amal1', 'amal2'], ...]` — modullar alifbo tartibida, har bir modul ichidagi amallar ham alifbo tartibida. Bo‘sh ro‘yxat — `[]`.

MISOL · CHAQIRUV

```
modul_ruxsatlari(['orders.view', 'stock.move', 'orders.create'])
```

MISOL · NATIJA

```
{"orders":["create","view"],"stock":["move"]}
```

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: Audit: o‘zgargan maydonlar

Eloquent `updated` hodisasida audit yozuvi uchun eski va yangi qiymatlar farqini hisoblang. Faqat `$yangi` dagi kalitlar ko‘riladi, `updated_at` e’tiborga olinmaydi. Kalit `$eski` da yo‘q bo‘lsa yoki qiymat qat’iy (`!==`) farq qilsa — o‘zgargan; eski qiymat yo‘q bo‘lsa `null` yoziladi. `$yashirin` ro‘yxatidagi maydonlar (masalan, `password`) o‘zgargan bo‘lsa, ikkala tomonda ham `'***'` bilan almashtiriladi.

Imzo: `function audit_farq(array $eski, array $yangi, array $yashirin): ?array`

Qaytaradi: `['old' => [...], 'new' => [...]]` — ikkala qismda kalitlar alifbo tartibida. Hech narsa o‘zgarmagan bo‘lsa — `null`.

MISOL · CHAQIRUV

```
audit_farq(['price' => '125000.00', 'name' => 'Un'], ['price' => '119000.00', 'name' => 'Un'], [])
```

MISOL · NATIJA

```
{"old":{"price":"125000.00"},"new":{"price":"119000.00"}}
```

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 · O‘rta · 20 ball

6-topshiriq: Login urinishlarini cheklash

Brute-force hujumidan himoya: oxirgi `$oyna` sekund ichidagi muvaffaqiyatsiz urinishlar soni `$limit` ga yetsa, login bloklanadi. `$urinishlar` — muvaffaqiyatsiz urinishlar vaqti (Unix sekund, tartibsiz bo‘lishi mumkin). Oynaga `t > $hozir - $oyna` bo‘lgan urinishlar kiradi. Bloklanganda `retry_after` — oynadagi urinishlar soni `limit - 1` gacha kamayishi uchun kutish kerak bo‘lgan sekundlar. `$limit < 1`, `$oyna < 1` yoki kelajakdagi vaqt (`t > $hozir`) bo‘lsa — `InvalidArgumentException`.

Imzo: `function login_cheklov(array $urinishlar, int $hozir, int $limit = 5, int $oyna = 60): array`

Qaytaradi: `['blocked' => bool, 'remaining' => int, 'retry_after' => int]`. Bloklanmagan bo‘lsa `remaining = limit - son`, `retry_after = 0`; bloklangan bo‘lsa `remaining = 0`.

MISOL · CHAQIRUV

```
login_cheklov([100, 130, 150, 155, 170], 175, 3)
```

MISOL · NATIJA

```
{"blocked":true,"remaining":0,"retry_after":35}
```

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: Imzolangan havola (signed URL)

Hisob-faktura PDF faylini login qilmagan mijozga vaqtinchalik havola orqali berish kerak (Laravel `URL::temporarySignedRoute` g‘oyasi). `sign($url, $expires)` havolaga `expires=<vaqt>` parametrini qo‘shadi (`?` yoki `&` bilan) va oxiriga `&signature=<hash_hmac('sha256', shu_havola, kalit)>` ni yozadi. `check($url, $now)` imzoni qayta hisoblaydi: `signature` parametri yo‘q yoki imzo mos kelmasa — `'invalid'`; imzo to‘g‘ri, lekin `$now > expires` — `'expired'`; aks holda `'ok'`. Imzoni `hash_equals` bilan solishtiring. Bo‘sh kalit bilan obyekt yaratilsa — `InvalidArgumentException`.

Imzo: `final class SignedUrl`

Qaytaradi: `sign()` — imzolangan havola satri; `check()` — `'ok'`, `'expired'` yoki `'invalid'`.

MISOL · CHAQIRUV

```
(new SignedUrl('k1'))->sign('https://omborpro.test/invoices/45/pdf', 1800000000)
```

MISOL · NATIJA

```
"https://omborpro.test/invoices/45/pdf?expires=1800000000&signature=d983232e8047e6ac7291798e337e4a0aca66fb70b9e0f06921bd1afb44b88d0f"
```

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: O‘zgartirib bo‘lmaydigan audit zanjiri

Audit jurnali yozuvlari keyinchalik bazada jimgina o‘zgartirilmaganini isbotlash uchun hash-zanjir quring. `append($data)` yangi yozuv qo‘shadi: `prev` — oldingi yozuv `hash` i (birinchisi uchun 64 ta `'0'`), `hash = hash('sha256', prev . json_encode($data))`, va hashni qaytaradi; bo‘sh `$data` — `InvalidArgumentException`. `entries()` — `['data' => ..., 'prev' => ..., 'hash' => ...]` yozuvlar ro‘yxati. Statik `verify($entries)` zanjirni tekshiradi: yozuvning `prev` i oldingi hashga teng emas yoki `hash` qayta hisoblanganiga mos kelmasa — o‘sha yozuv indeksini qaytaradi; hammasi to‘g‘ri bo‘lsa `-1`. Yozuvda `data`, `prev` yoki `hash` kaliti bo‘lmasa — `InvalidArgumentException`.

Imzo: `final class AuditChain`

Qaytaradi: `append()` — 64 belgili hex hash; `entries()` — massiv; `verify()` — buzilgan birinchi yozuv indeksi yoki `-1`.

MISOL · CHAQIRUV

```
(function () { $c = new AuditChain(); return $c->append(['user_id' => 1, 'action' => 'login']); })()
```

MISOL · NATIJA

```
"2823c6cd5b5fade3ff0a9d857d1772a8f05a4cb35ca35df1f169b817de80849b"
```

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

5-amaliy: 8 ta PHP topshiriq

1. **Parol mustahkamligi**
   Oson · 10 ball · ma’ruzada

2. **OrderPolicy qarori**
   O‘rta · 20 ball · ma’ruzada

3. **Ruxsat: aniq nom va wildcard**
   O‘rta · 20 ball · ma’ruzada

4. **Ruxsatlarni modullar bo‘yicha guruhlash**
   Oson · 10 ball

5. **Audit: o‘zgargan maydonlar**
   O‘rta · 20 ball

6. **Login urinishlarini cheklash**
   O‘rta · 20 ball

7. **Imzolangan havola (signed URL)**
   Qiyin · 30 ball

8. **O‘zgartirib bo‘lmaydigan audit zanjiri**
   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.