2

IKKINCHI AMALIY MASHG‘ULOT

Korxona ilovalarini dasturlash · 2 soat · 2-mavzu

Loyiha skeleti: Laravel, Docker Compose va PostgreSQL

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

Qatlamlar | MVC | Service Layer | Repository | DTO

MASHG‘ULOT MAQSADI

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

1. **O‘rnatish** — Laravel 12 loyihasini Composer orqali yaratish va Git ga joylash
2. **Sozlash** — Docker Compose: php-fpm, nginx, PostgreSQL 16 va Redis 7 xizmatlari
3. **Ulash** — `.env` orqali Laravel ni PostgreSQL ga ulash va migratsiyalarni bajarish
4. **Tashkil etish** — modulli struktura: Services, DTO, Repositories, Enums, Exceptions
5. **Tekshirish** — `/up` health check, `artisan about`, Pint — ishga tushadigan skelet

Qanday bajariladi va qanday baholanadi?

Ish tartibi

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

Laravel o‘rnatish va Docker Compose

composer create-project · Dockerfile · docker-compose.yml · nginx

Laravel 12

Docker

nginx

1-QADAM

Loyiha va Git

```
composer create-project laravel/laravel:^12.0 omborpro
cd omborpro
php artisan --version
git init && git add . && git commit -m "chore: Laravel 12 skeleti"
git remote add origin git@github.com:ombor-3/omborpro.git
git push -u origin main
```

▶ NATIJA

```
Laravel Framework 11.x.x
[main (root-commit) 3f2a1c9] chore: Laravel 12 skeleti
 58 files changed, ...
```

`laravel/laravel:^12.0`

12.x versiya: >= 12.0 va < 13.0

`php artisan --version`

o‘rnatilgan versiyani tekshirish

`chore:`

Conventional Commits uslubidagi xabar

2-QADAM

Dockerfile — PHP 8.2 FPM va kengaytmalar

```
FROM php:8.2-fpm

RUN apt-get update && apt-get install -y \
        libpq-dev libzip-dev libicu-dev unzip \
    && docker-php-ext-install pdo_pgsql zip intl bcmath \
    && rm -rf /var/lib/apt/lists/*

COPY --from=composer:2 /usr/bin/composer /usr/bin/composer

WORKDIR /var/www/html
```

▶ NATIJA

```
docker compose build app
=> [app 2/3] RUN apt-get update && ...
=> naming to docker.io/library/omborpro-app
```

`php:8.2-fpm`

rasmiy obraz: PHP + FastCGI jarayon menejeri, 9000-port

`pdo_pgsql`

PostgreSQL drayveri — `libpq-dev` bilan quriladi

`COPY --from=composer:2`

Composer ni tayyor obrazdan olish

`rm -rf /var/lib/apt/lists`

obraz hajmini kamaytirish

3-QADAM

docker-compose.yml — app va web

```yaml
services:
  app:
    build: .
    volumes:
      - ./:/var/www/html
    depends_on:
      db:
        condition: service_healthy
  web:
    image: nginx:1.27-alpine
    ports:
      - "8080:80"
    volumes:
      - ./:/var/www/html
      - ./docker/nginx.conf:/etc/nginx/conf.d/default.conf
    depends_on:
      - app
```

▶ NATIJA

`http://localhost:8080 → web (nginx) → app:9000 (php-fpm)`

`build: .`

ildizdagi Dockerfile dan quriladi

`./:/var/www/html`

kod konteynerga ulanadi — qayta qurish shart emas

`service_healthy`

baza tayyor bo‘lguncha app kutadi

`"8080:80"`

xost porti : konteyner porti

docker-compose.yml — db va redis

```yaml
  db:
    image: postgres:16
    environment:
      POSTGRES_DB: omborpro
      POSTGRES_USER: omborpro
      POSTGRES_PASSWORD: secret
    volumes:
      - pgdata:/var/lib/postgresql/data
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U omborpro -d omborpro"]
      interval: 5s
      retries: 10
  redis:
    image: redis:7-alpine

volumes:
  pgdata:
```

▶ NATIJA

```
docker compose ps
NAME                SERVICE   STATUS
omborpro-db-1       db        Up (healthy)
omborpro-redis-1    redis     Up
```

postgres:16

kurs stack i: PostgreSQL 16

pgdata

nomlangan volume — ma’lumotlar konteynerdan tashqarida

pg_isready

baza ulanishga tayyorligini tekshiradi

secret

faqat lokal ishlab chiqish uchun; production da boshqa parol

4-QADAM

docker/nginx.conf

```
server {
  listen 80;
  root /var/www/html/public;
  index index.php;

  location / {
    try_files $uri $uri/ /index.php?$query_string;
  }

  location ~ \.php$ {
    fastcgi_pass app:9000;
    fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
    include fastcgi_params;
  }
}
```

▶ NATIJA

`GET /orders → try_files → /index.php?… → php-fpm (app:9000)`

`root .../public`

faqat `public/` ochiq — `.env` tashqaridan ko‘rinmaydi

`try_files`

fayl topilmasa so‘rov Laravel ga (index.php) beriladi

`app:9000`

compose xizmati nomi — ichki DNS

02

.env va PostgreSQL ulanishi

Muhit o‘zgaruvchilari · .env.example · migratsiya · tipik xatolar

.env

PostgreSQL

Migratsiya

5-QADAM

.env — muhim qatorlar

```
APP_NAME=OmborPro
APP_ENV=local
APP_DEBUG=true
APP_TIMEZONE=Asia/Tashkent
APP_URL=http://localhost:8080
APP_LOCALE=uz

DB_CONNECTION=pgsql
DB_HOST=db
DB_PORT=5432
DB_DATABASE=omborpro
DB_USERNAME=omborpro
DB_PASSWORD=secret

REDIS_HOST=redis
```

▶ NATIJA

```
config('database.default')  → "pgsql"
config('app.timezone')      → "Asia/Tashkent"
```

DB_HOST=db

compose xizmati nomi — konteyner ichida `localhost` bu ilovaning o‘zi

DB_CONNECTION=pgsql

Laravel 12 standarti `sqlite` — almashtiramiz

APP_TIMEZONE

`now()` Toshkent vaqtida

.env.example

shu fayl parolsiz holda repozitoriyga qo‘shiladi

6-QADAM

Ishga tushirish va migratsiya

```
docker compose up -d --build
docker compose exec app php artisan key:generate
docker compose exec app php artisan migrate
docker compose exec app php artisan about --only=drivers
curl -I http://localhost:8080/up
```

▶ NATIJA

```
INFO  Application key set successfully.
INFO  Running migrations.
0001_01_01_000000_create_users_table ........ DONE
Database ............................ pgsql
HTTP/1.1 200 OK
```

up -d --build

obrazni qurish va fon rejimida ishga tushirish

exec app

buyruq ilova konteyneri ichida bajariladi

about --only=drivers

baza, kesh, navbat drayverlarini ko‘rsatadi

/up

health check — 200 bo‘lsa ilova tirik

TIPIK XATOLAR

Xato xabari → sabab → yechim

Xato	Sabab	Yechim
could not find driver	`pdo_pgsql` yo‘q	Dockerfile ga qo‘shib, `up -d --build`
could not translate host name	`DB_HOST` noto‘g‘ri	`DB_HOST=db` (xizmat nomi)
Connection refused	Baza hali ishga tushmagan	`healthcheck` + `service_healthy`
502 Bad Gateway	nginx php-fpm ni topmayapti	`fastcgi_pass app:9000` va `app` ishlayotganini tekshiring
No application encryption key	`APP_KEY` bo‘sh	`php artisan key:generate`
.env o‘zgarishi ta’sir qilmayapti	Konfiguratsiya keshlangan	`php artisan config:clear`

03

Modulli struktura va yakuniy tekshiruv

Papkalar · Artisan generatorlari · konteynerda bog‘lash · Pint · README

Struktura

Artisan

Pint

7-QADAM

Papkalar va boshlang‘ich sinflar

```
php artisan make:enum Enums/OrderStatus --string
php artisan make:class DTO/OrderData
php artisan make:class DTO/OrderItemData
php artisan make:class Services/OrderService
php artisan make:class Services/StockService
php artisan make:interface Repositories/ProductRepository
php artisan make:class Repositories/EloquentProductRepository
php artisan make:exception InsufficientStockException
php artisan make:exception InvalidStatusTransitionException
```

▶ NATIJA

```
INFO  Enum [app/Enums/OrderStatus.php] created successfully.
INFO  Class [app/DTO/OrderData.php] created successfully.
INFO  Interface [app/Repositories/ProductRepository.php] created
successfully.
INFO  Exception [app/Exceptions/InsufficientStockException.php]
created successfully.
```

`Enums/OrderStatus`

nomdagi papka → `App\Enums` namespace

`make:interface`

Laravel 12 buyrug‘i

`make:exception`

`app/Exceptions` ga; merosni `DomainException` ga o‘zgartiring

7-QADAM

Enum va domen istisnosi

```php
// app/Enums/OrderStatus.php
enum OrderStatus: string
{
    case Draft = 'draft';
    case Confirmed = 'confirmed';
    case Reserved = 'reserved';
    case Invoiced = 'invoiced';
    case Shipped = 'shipped';
    case Cancelled = 'cancelled';
}

// app/Exceptions/InsufficientStockException.php
class InsufficientStockException extends DomainException
{
}
```

▶ NATIJA

```
php artisan tinker
> App\Enums\OrderStatus::from('reserved')
= App\Enums\OrderStatus {+name: "Reserved", +value: "reserved"}
```

`enum OrderStatus: string`

satr qiymatli enum — bazada `status` ustunida saqlanadi

`OrderStatus::from()`

satrdan enum; noto‘g‘ri qiymat — `ValueError`

extends DomainException

biznes-qoida buzilishi — `make:exception` standart merosini almashtiring

9-QADAM

“Ishlaydigan skelet” mezonlari

1

Ishga tushadi

`docker compose up -d --build` → `/up` 200

2

Baza

`php artisan migrate` PostgreSQL da xatosiz

3

Struktura

Enums, DTO, Services, Repositories, Exceptions papkalari

4

Uslub

`./vendor/bin/pint --test` — xato yo‘q

5

Hujjat

README bo‘yicha jamoadosh toza kompyuterda ishga tushirdi

6

PR

`feature/skeleton` → `main`, kamida bitta tasdiq

LOYIHA BOSQICHI

Natija va topshirish

Natija: Docker Compose (php-fpm, nginx, PostgreSQL 16, Redis 7) da ishga tushadigan Laravel 12 skeleti, PostgreSQL ga ulangan va migratsiyalar bajarilgan, modulli papka strukturasi va README yo‘riqnomasi bilan GitHub repozitoriysida.

Baholash mezonlari

- ✓ `Dockerfile` va `docker-compose.yml` repozitoriyda: app, web, db (postgres:16, healthcheck, volume), redis xizmatlari
- ✓ `docker compose up -d --build` dan so‘ng `http://localhost:8080/up` 200 qaytaradi
- ✓ `.env` Git da yo‘q; `.env.example` da `DB_CONNECTION=pgsql`, `DB_HOST=db` va boshqa kalitlar parolsiz bor
- ✓ `php artisan migrate` PostgreSQL da xatosiz bajariladi (`about` da drayver pgsql)
- ✓ `app/Enums/OrderStatus`, `app/DTO`, `app/Services`, `app/Repositories`, `app/Exceptions` sinflari to‘g‘ri namespace bilan
- ✓ `AppServiceProvider` da `ProductRepository` interfeysi realizatsiyaga bog‘langan
- ✓ `./vendor/bin/pint --test` xatosiz; README da ishga tushirish qadamlari yozilgan
- ✓ O‘zgarishlar Pull Request orqali qo‘shilgan va jamoadosh tomonidan tasdiqlangan

📤 GitHub repozitoriysi havolasi va `feature/skeleton` Pull Request havolasini topshiring. PR tavsifiga `docker compose ps` chiqishi va `/up` sahifasining skrinshotini qo‘shing.

Telefon kamerasi bilan skanerlang

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

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · Oson · 10 ball

4-topshiriq: .env faylini o‘qish

`.env` matnini kalit–qiymat massiviga aylantiring. Qatorlar `\n` (yoki `\r\n`) bilan ajratilgan; bo‘sh qatorlar va `#` bilan boshlanadigan izohlar o‘tkazib yuboriladi. Kalit va qiymat atrofidagi bo‘sh joylar olib tashlanadi; faqat birinchi `=` ajratuvchi hisoblanadi; qiymat qo‘sh tirnoqda bo‘lsa, tirnoqlar olib tashlanadi. Bir kalit qayta kelsa — oxirgisi olinadi. `=` siz qator yoki kalit `^[A-Z_][A-Z0-9_]*$` ga mos bo‘lmasa — `InvalidArgumentException`.

Imzo: `function env_parse(string $matn): array`

Qaytaradi: `['KALIT' => 'qiymat', ...]` — kalitlar birinchi uchragan tartibida, qiymatlar satr.



Telefon kamerasi bilan skanerlang

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

MISOL · CHAQIRUV

```
env_parse("APP_NAME=OmborPro\nDB_PORT=5432")
```

MISOL · NATIJA

```
{"APP_NAME":"OmborPro","DB_PORT":"5432"}
```

AMALIY TOPSHIRIQ

Kompyuterda mustaqil · O‘rta · 20 ball

5-topshiriq: PDO DSN satrini yasash

`.env` qiymatlaridan PDO ulanish satrini (DSN) yasang. `DB_CONNECTION`: `pgsql` → `pgsql:host=H;port=P;dbname=D` (standart port 5432); `mysql` → `mysql:host=H;port=P;dbname=D;charset=utf8mb4` (standart port 3306); `sqlite` → `sqlite:D`. `DB_HOST` berilmasa — `127.0.0.1`. `DB_CONNECTION` yoki `DB_DATABASE` yo‘q, drayver noma’lum yoki port 1–65535 oralig‘idagi butun son bo‘lmasa — `InvalidArgumentException`.

Imzo: `function pdo_dsn(array $env): string`

Qaytaradi: DSN satri.

MISOL · CHAQIRUV

```
pdo_dsn(['DB_CONNECTION' => 'pgsql', 'DB_HOST' => 'db',
'DB_PORT' => '5432', 'DB_DATABASE' => 'omborpro'])
```

MISOL · NATIJA

```
"pgsql:host=db;port=5432;dbname=omborpro"
```

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: Docker Compose: band portlar

Xizmatlar va ularning `ports` ro‘yxati berilgan. Yozuv formatlari: `"XOST:KONTEYNER"`, `"IP:XOST:KONTEYNER"` yoki faqat `"KONTEYNER"` (xostga chiqarilmaydi — hisobga olinmaydi). Bir xil xost portini ikki va undan ortiq **turli** xizmat ishlatsa — bu to‘qnashuv. Port qismlari butun son bo‘lmasa yoki qismlar soni 1–3 dan farq qilsa — `InvalidArgumentException`.

Imzo: `function port_toqnashuvi(array $servislar): array`

Qaytaradi: `[xost_port => [xizmatlar, alifbo tartibida], ...]` — portlar o‘sish tartibida; to‘qnashuv bo‘lmasa — `[]`.

MISOL · CHAQIRUV

```
port_toqnashuvi(['web' => ['8080:80'], 'db' => ['5432:5432'],
'adminer' => ['8080:8080']])
```

MISOL · NATIJA

```
{"8080":["adminer","web"]}
```

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: Xizmatlarni ishga tushirish tartibi

`depends_on` bog‘liqliklari berilgan: `['app' => ['db', 'redis'], 'web' => ['app'], 'db' => [], 'redis' => []]`. Har bir xizmat o‘zi bog‘liq bo‘lgan barcha xizmatlardan keyin ishga tushishi kerak. Bir vaqtda bir nechta xizmat tayyor bo‘lsa — alifbo bo‘yicha birinchisi oldin. Ro‘yxatda yo‘q xizmatga bog‘liqlik — `InvalidArgumentException`; aylana bog‘liqlik (o‘ziga bog‘liqlik ham) — `DomainException`.

Imzo: `function ishga_tushirish_tartibi(array $depends): array`

Qaytaradi: Xizmat nomlari ro‘yxati ishga tushirish tartibida.

MISOL · CHAQIRUV

```
ishga_tushirish_tartibi(['app' => ['db', 'redis'], 'web' => ['app'], 'db' => [], 'redis' => []])
```

MISOL · NATIJA

```
["db","redis","app","web"]
```

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: Composer versiya cheklovi

`composer.json` dagi cheklov versiyaga mosligini tekshiring. Versiya — `X.Y.Z` (yetishmagan qismlar 0). Qo‘llab-quvvatlanadigan yozuvlar: aniq `11.0.0`; `>=`, `>`, `<=`, `<` (masalan, `>=8.1`); `^X.Y.Z` — `X > 0` bo‘lsa `<(X+1).0.0` gacha, `^0.Y` → `<0.(Y+1).0`, `^0.0.Z` → `<0.0.(Z+1)`; `~X.Y` → `<(X+1).0.0`, `~X.Y.Z` → `<X.(Y+1).0`. Bo‘sh joy yoki vergul — VA, `||` — YOKI. Tushunarsiz yozuv — `InvalidArgumentException`.

Imzo: `function composer_mos(string $cheklov, string $versiya): bool`

Qaytaradi: `true` — versiya cheklovga mos, aks holda `false`.

MISOL · CHAQIRUV	MISOL · NATIJA
`composer_mos('^11.0', '11.31.0')`	`true`

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

2-amaliy: 8 ta PHP topshiriq

1 **Qatlam qoidasini buzgan bog‘liqliklar**
O‘rta · 20 ball · ma’ruzada

2 **Xotiradagi ProductRepository**
O‘rta · 20 ball · ma’ruzada

3 **OrderData DTO**
O‘rta · 20 ball · ma’ruzada

4 **.env faylini o‘qish**
Oson · 10 ball

5 **PDO DSN satrini yasash**
O‘rta · 20 ball

6 **Docker Compose: band portlar**
O‘rta · 20 ball

7 **Xizmatlarni ishga tushirish tartibi**
Qiyin · 30 ball

8 **Composer versiya cheklovi**
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.