

Veb-ilovani yaratishda siz uchun turli xil vositalar va ramkalar mavjud. Biroq, biz Laravel zamonaviy, to'liq stekli veb-ilovalarni yaratish uchun eng yaxshi tanlov deb hisoblaymiz.

Progressiv ramka

Biz Laravelni "progressiv" ramka deb atashni yaxshi ko'ramiz. Bu bilan biz Laravel siz bilan birga o'sadi, demoqchimiz. Agar siz veb-ishlab chiqish yo'lida endigina birinchi qadamlaringizni qo'yayotgan bo'lsangiz, Laravelning hujjatlar, qo'llanmalar va [video darsliklardan](#) iborat keng kutubxonasi sizga arqonlarni chalg'itmasdan o'rganishga yordam beradi.

Agar siz katta dasturchi bo'lsangiz, Laravel sizga [qaramlikni in'ektsiya qilish](#), [birlik testi](#), [navbatlar](#), [real vaqt voqealari](#) va boshqalar uchun mustahkam vositalarni taqdim etadi. Laravel professional veb-ilovalarni yaratish uchun yaxshi sozlangan va korporativ ish yuklarini boshqarishga tayyor.

Scalable Framework

Laravel nihoyatda kengaytirilishi mumkin. PHP ning masshtablash uchun qulay tabiati va Redis kabi tez, taqsimlangan kesh tizimlari uchun Laravel-ning o'rnatilgan qo'llab-quvvatlashi tufayli Laravel bilan gorizontal masshtablash juda oson. Darhaqiqat, Laravel ilovalari

oyiga yuz millionlab so'rovlarni bajarish uchun osonlik bilan kengaytirilgan.

Haddan tashqari o'lchov kerakmi? [Laravel Vapor](#) kabi platformalar sizga Laravel ilovangizni AWS-ning so'nggi serversiz texnologiyasida deyarli cheksiz miqyosda ishlatish imkonini beradi.

Jamiyat doirasi

Laravel PHP ekosistimidagi eng yaxshi paketlarni birlashtirib, mavjud bo'lgan eng mustahkam va ishlab chiquvchilar uchun qulay tizimni taklif qiladi. Bundan tashqari, butun dunyo bo'ylab minglab iqtidorli dasturchilar [ramkaga o'z hissalarini qo'shdilar](#). Kim biladi, balki siz hatto Laravel ishtirokchisiga aylanasiz.

Sizning birinchi Laravel loyihangiz

Birinchi Laravel loyihangizni yaratishdan oldin, mahalliy kompyuteringizda PHP va [Composer](#) o'rnatilganligiga ishonch hosil qilishingiz kerak. Agar siz macOS-da ishlayotgan bo'lsangiz, PHP va Composer-ni [Homebrew](#) orqali o'rnatishingiz mumkin. Bundan tashqari, biz [Node va NPM ni o'rnatishni](#) tavsiya qilamiz.

create-project PHP va Composer-ni o'rnatganingizdan so'ng, Composer buyrug'i orqali yangi Laravel loyihasini yaratishingiz mumkin :

```
composer create-project laravel/laravel example-app
```

Yoki Laravel o'rnatuvchisini Composer orqali global o'rnatish orqali yangi Laravel loyihalarini yaratishingiz mumkin:

```
composer global require laravel/installer
```

```
laravel new example-app
```

Loyiha yaratilgandan so'ng, Laravel's Artisan CLI **serve** buyrug'i yordamida Laravelning mahalliy ishlab chiqish serverini ishga tushiring:

```
cd example-app
```

```
php artisan serve
```

Artisan ishlab chiqish serverini ishga tushirganingizdan so'ng, ilovangizga veb-brauzeringiz orqali kirish mumkin bo'ladi <http://localhost:8000>. Keyin, siz [Laravel ekotizimiga keyingi qadamlaringizni](#) qo'yishga tayyormiz. Albatta, siz [ma'lumotlar bazasini ham sozlashni](#) xohlashingiz mumkin.

Veb-ishlab chiqish uchun dasturiy ta'minot muhiti

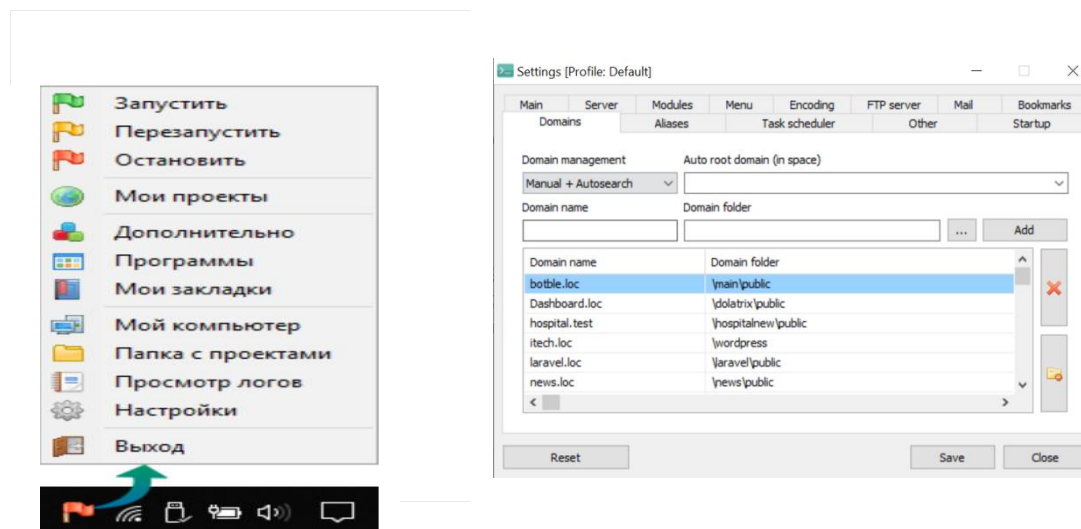
Open Server Panel - bu veb-ishlab chiquvchilar uchun, ularning tavsiyalari va istaklarini inobatga olgan holda maxsus yaratilgan ko'chma dasturiy ta'minot muhiti.

Ushbu dasturiy ta'minot to'plami diqqat bilan tanlangan server dasturlari to'plamini, shuningdek, barcha mavjud komponentlar uchun kuchli boshqaruv va konfiguratsiya imkoniyatlariga ega bo'lgan ajoyib darajada qulay va puxta o'ylangan boshqaruv yordam dasturini o'z ichiga oladi.

OSPanel veb-loyihalarni ishlab chiqish, disk raskadrovka va sinovdan o'tkazish, shuningdek, mahalliy tarmoqlarda veb-xizmatlarni taqdim etish uchun keng qo'llaniladi.

Loyiha 10 yildan ortiq vaqtdan beri mavjud bo'lib, u har bir veb-masterga kerak bo'lgan birinchi darajali, oddiy va ishonchli vosita sifatida muvaffaqiyatli o'zini namoyon qildi. Rivojlanish yillari davomida biz shunga o'xshash dasturiy mahsulotlarda mavjud bo'lgan eng yaxshi narsalarni unda amalga oshirishga harakat qildik.

Hamjamiyatga qo'shiling va veb-ishlanma olamini zabt eting!

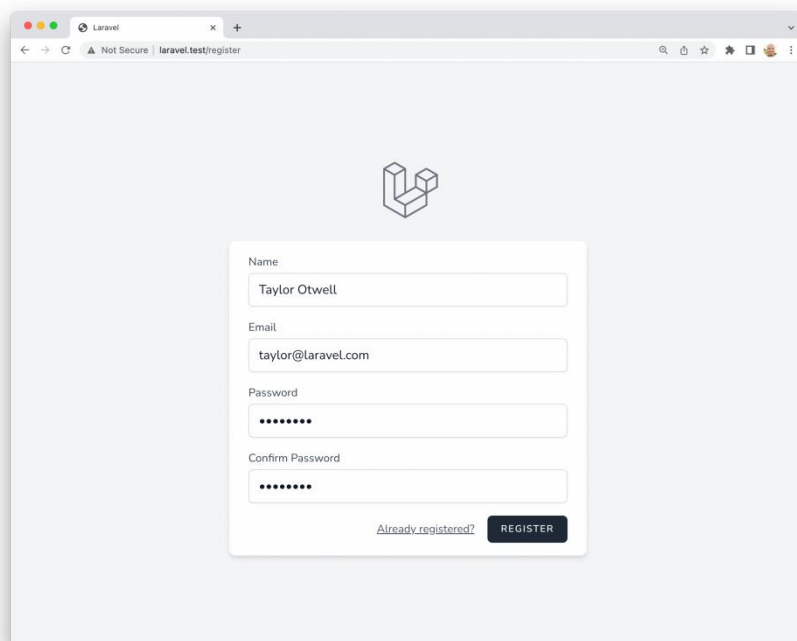


Laravel Breeze

[Laravel Breeze](#) - login, ro'yxatdan o'tish, parolni qayta o'rnatish, elektron pochta orqali tekshirish va parolni tasdiqlash kabi barcha Laravel [autentifikatsiya xususiyatlarining](#) minimal, oddiy amalga oshirilishi. Bundan tashqari, Breeze foydalanuvchi nomi, elektron pochta manzili va parolini yangilashi mumkin bo'lgan oddiy "profil" sahifasini o'z ichiga oladi.

Laravel Breeze standart ko'rinish qatlami [Tailwind CSS](#) bilan yaratilgan oddiy [Blade shablonlaridan](#) iborat. Yoki, Breeze ilovangizni Vue yoki React va [Inertia](#) -dan foydalanib qurishi mumkin.

Breeze yangi Laravel dasturini boshlash uchun ajoyib boshlanish nuqtasini taqdim etadi va shuningdek, [Laravel Livewire](#) bilan Blade shablonlarini keyingi bosqichga olib chiqishni rejalashtirgan loyihalar uchun ajoyib tanlovdir.



Laravel Bootcamp

Agar siz Laravelda yangi bo'lsangiz, [Laravel Bootcamp](#) -ga o'ting . Laravel Bootcamp sizga Breeze yordamida birinchi Laravel ilovangizni yaratishda yordam beradi. Bu Laravel va Breeze taklif qiladigan barcha narsalarga sayohat qilishning ajoyib usuli.

O'rnatish

Birinchidan, siz [yangi Laravel ilovasini yaratishingiz](#) , ma'lumotlar bazasini sozlashingiz va ma'lumotlar [bazasi migratsiyasini](#) ishga tushirishingiz kerak . Yangi Laravel ilovasini yaratganingizdan so'ng, Composer yordamida Laravel Breeze-ni o'rnatishingiz mumkin:

```
composer require laravel/breeze --dev
```

Breeze o'rnatilgandan so'ng, siz quyidagi hujjatlarda ko'rib chiqilgan Breeze "steklaridan" biri yordamida ilovangizni o'rnatishingiz mumkin.

Breeze & Blade

Composer Laravel Breeze paketini o'rnatgandan so'ng, **breeze:install** Artisan buyrug'ini ishga tushirishingiz mumkin. Ushbu buyruq ilovangiz uchun autentifikatsiya ko'rinishlarini, marshrutlarni, kontrollerlarni va boshqa resurslarni nashr etadi. Laravel Breeze o'zining barcha kodlarini ilovangizga nashr etadi, shunda siz uning xususiyatlari va amalga oshirilishi ustidan to'liq nazorat va ko'rinishga ega bo'lasiz.

Standart Breeze "stek" bu Blade stekidir, u ilovangizning old qismini ko'rsatish uchun oddiy [Blade shablonlaridan](#) foydalanadi . **breeze:install** Blade stekini boshqa qo'shimcha argumentlarsiz buyruqni chaqirish orqali

o'rnatish mumkin . Breeze iskala o'rnatilgandan so'ng, siz ilovangizning frontend aktivlarini ham kompilyatsiya qilishingiz kerak:

```
php artisan breeze:install
```

```
php artisan migrate
```

```
npm install
```

```
npm run dev
```

Keyin veb-brauzeringizda ilovangiz `/login` yoki URL manzillariga o'tishingiz mumkin. Breeze ning barcha marshrutlari fayl `/register` ichida belgilangan `.routes/auth.php`

Qorong'i rejim

Agar siz Breeze ilovangizning old tomonini iskala qilishda "qorong'i rejim" qo'llab-quvvatlashini qo'shishini istasangiz , buyruqni `--dark` bajarishda oddiygina ko'rsatma bering: `breeze:install`

```
php artisan breeze:install --dark
```

Login page

```
<?php
```

```
namespace App\Http\Controllers\Auth;
```

```
use App\Http\Controllers\Controller;
```

```
use App\Providers\RouteServiceProvider;
```

```
use Illuminate\Foundation\Auth\AuthenticatesUsers;
```

```
class LoginController extends Controller
```

```
{
```

```
    /**
```

```
    |-----
```

```

| Login Controller
|
| This controller handles authenticating users for the
application and
| redirecting them to your home screen. The controller uses a
trait
| to conveniently provide its functionality to your applications.
|
| */

```

```

use AuthenticatesUsers;

```

```

/**
 * Where to redirect users after login.
 *
 * @var string
 */
protected $redirectTo = RouteServiceProvider::HOME;

```

```

/**
 * Create a new controller instance.
 *
 * @return void
 */
public function __construct()
{
    $this->middleware('guest')->except('logout');
}
}

```

Rttm
Login

Login

E-Mail Address
admin@admin.com

Password

☐ Remember Me

Login

Register page

```
<?php
```

```
namespace App\Http\Controllers\Auth;
```

```
use App\Http\Controllers\Controller;
use App\Providers\RouteServiceProvider;
use App\Models\User;
use Illuminate\Foundation\Auth\RegistersUsers;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
```

```
class RegisterController extends Controller
```

```
{
    /**
     * The controller that registers new users, validates their input,
     * sends an email verification email, etc.
     *
     * @var RegisterController
     *
     * This controller handles the registration of new users as well
     * as their validation and creation. By default this controller uses a
     * trait to provide this functionality without requiring any additional
     * code.
     */
}
```

```
use RegistersUsers;
```

```
/**
 * Where to redirect users after registration.
 *
 * @var string
 */
```

```
protected $redirectTo = RouteServiceProvider::HOME;
```

```
/**
 * Create a new controller instance.
 */
```



```

*
* @return void
*/
public function __construct()
{
    $this->middleware('guest');
}

```

```

/**
 * Get a validator for an incoming registration request.
 *
 * @param array $data
 * @return \Illuminate\Contracts\Validation\Validator
 */
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],

```

```

        'email' => ['required', 'string', 'email', 'max:255',
'unique:users'],
        'password' => ['required', 'string', 'min:8',
'confirmed'],
    ]);
}

```

```

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return \App\Models\User
 */
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],

```

```

        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);
}
}

```

Register

Name

E-Mail Address

Password

Confirm Password

Register