

Word Creator Uygulaması Dökümantasyonu

Bu proje, Laravel kullanarak belirli bir zaman diliminde çalışan bir Cron servisi oluşturmayı amaçlamaktadır. Servis, her çalışma döngüsünde 10 karakterlik büyük ve küçük harflerden oluşan kelime üretecektir. Üretilen kelimeler bir veritabanı tablosunda görüntülenecek ve tabloda her kelimenin büyük ve küçük harf duysız olarak kullanılan harf sayıları yer alacaktır. Ayrıca, verilerin sayfalama işlemi için uygun bir arayüz sağlanacaktır.

Kurulum ve Yapılandırma

1. Öncelikle laravel projemizi composer create-project laravel/laravel word-creator komutuyla kuruyoruz.
2. Sonra ki adımda ise proje dizinine gidip composer install komutunu çalıştırıyoruz.
3. .env dosyamızı oluşturmak için .env.example dosyasını kopyalıyoruz.
4. Php artisan key:generate fonksiyonu ile app_key tanımlıyoruz.
5. Laravel saili kurmak için composer require laravel/sail --dev komutunu çalıştırıyoruz.Ardından php artisan sail:install komutunu çalıştırıp saili indiriyoruz. (Bu projede postgresql i docker da ayağa kaldırdım ve laravel sail kullandım).
6. .env dosyasında veritabanımızı yapılandırıyoruz.

```
DB_CONNECTION=pgsql
DB_HOST=localhost
DB_PORT=5432
DB_DATABASE=word-creator
DB_USERNAME=sail
DB_PASSWORD=password
```

Tablo , Model Ve Controller Oluşturma

1. Php artisan make:migration create_words_table komutunu kullanarak kelimeler tablosunu oluşturuyoruz. Ve oluşturacağımız kelimeyi tutabilmek için string tipinde 'word' değişkenimizi ekliyoruz.

```
public function up(): void
{
    Schema::create( table: 'words', function (Blueprint $table) {
        $table->id();
        $table->string( column: 'word');
        $table->timestamps();
    });
}
```

2. Php artisan make:model Word komutu ile modelimi oluşturuyoruz.

```

class Word extends Model
{
    use HasFactory;
    no usages
    protected $fillable = [
        'word'
    ];

    alpereninci *
    public function getLetterCounts() // Harf adetlerini hesaplayan fonksiyon
    {
        $counts = array_count_values(str_split(strtolower($this->word)));
        ksort( &array: $counts);

        return implode( separator: ', ', $counts);
    }
}

```

3. Php artisan migrate fonksiyonu ile tablolarımızı veritabanımıza aktarıyoruz.
4. Php artisan make:controller WordGeneratorController dosyasını oluşturuyoruz.

```

alpereninci
public function index()
{
    $words = Word::paginate(10);
    return view( view: 'words.index', compact( var_name: 'words'));
}

```

Kelime Oluşturma

Cron servisi ile belirli aralıkta 10 farklı harf ile 10 adet kelime oluşturmak için /app/console/kernel.php dosyasının içine kodumu yazıyoruz.

```

/**
 * Define the application's command schedule.
 */
no usages alpereninci
protected function schedule(Schedule $schedule): void
{
    $schedule->call(function () {
        $letters = 'abcdefghijklmnopqrstuvwxyz';
        for ($j = 0; $j < 10; $j++) {
            $word = '';
            for ($i = 0; $i < 10; $i++) {
                $letter = $letters[rand(0, 25)];
                $word .= $i % 2 === 0 ? strtoupper($letter) : strtolower($letter);
            }
            Word::create(['word' => $word]);
        }
    }->everyMinute());
}

```

Blade Dosyası Oluşturma

1. Resources/views/layouts/app.blade.php dizininde dosyayı oluşturuyoruz.
2. Resources/views/words/index.blade.php dizininde dosyayı oluşturuyoruz.

Route Oluşturma

```
Route::get( uri: '/', function () {  
    return view( view: 'welcome' );  
});  
  
Route::get( uri: '/words', [WordGeneratorController::class, 'index']->name( name: 'words.index' );
```

Cron Servisi Ayarları

1. Cron servisini başlatmak için crontab -e komutunu yazıyoruz.
2. * * * * * php /path/to/artisan schedule:run >> /dev/null 2>&1 komutunu yazıp kaydediyoruz.
3. Php artisan schedule:run komutunu çalıştırarak sistemimizi çalıştırıyoruz.

Projenin Başlatılması

1. ./vendor/bin/sail up komutu ile postgresql i ayağa kaldırıyoruz.
2. Php artisan serve ile projemizi başlatıyoruz.

Versiyonlar

- Laravel 10.10.1
- PHP 8.1
- Postgresql 15.3
- Laravel Sail 1.22