

LISTING PROGRAM

```
<?php
namespace App\Console;
use
Illuminate\Console\Scheduling\
Schedule;
use
Illuminate\Foundation\Console\
Kernel as ConsoleKernel;
class Kernel extends
ConsoleKernel
{
    /**
     * Define the application's
     command schedule.
     */

    protected function
schedule(Schedule $schedule):
void
    {
        // $schedule-
>command('inspire')->hourly();
    }
    /**
     * Register the commands for
     the application.
     */

    protected function
commands(): void
    {
        $this-
>load(__DIR__.'/Commands');
        require
base_path('routes/console.php');
    }
}
<?php

namespace App\Exceptions;
use
Illuminate\Foundation\Exceptio
ns\Handler as
ExceptionHandler;
use Throwable;
```

```
* A list of exception types with
their corresponding custom log
levels.
*
* @var array<class-
string<\Throwable>,
\Psr\Log\LogLevel::*>
*/
protected $levels = [
    //
];

/**
 * A list of the exception types
that are not reported.
*
* @var array<int, class-
string<\Throwable>>
*/
protected $dontReport = [
    //
];
/**
 * A list of the inputs that are
never flashed to the session on
validation exceptions.
*
* @var array<int, string>
*/
protected $dontFlash = [
    'current_password',
    'password',
    'password_confirmation',
];

/**
 * Register the exception
handling callbacks for the
application.
*/
public function register(): void
{
    $this->reportable(function
(Throwable $e) {
```

```
<?php
namespace
App\Http\Controllers;

use
Illuminate\Foundation\Auth\Ac
cess\AuthorizesRequests;
use
Illuminate\Foundation\Validatio
n\ValidatesRequests;
use
Illuminate\Routing\Controller
as BaseController;

class Controller extends
BaseController
{
    use AuthorizesRequests,
ValidatesRequests;
}
<?php

namespace
App\Http\Controllers;

use App\Models\Pengaduan;
use App\Models\Tanggapan;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Aut
h;
use Illuminate\Support\Str;

date_default_timezone_set('Asi
a/Jakarta');

class HomeController extends
Controller
{
    public function index()
    {
        if (Auth::check()) {
            return route('dashboard');
        }

        $ratings = [];
        $totalRatings = 0;
```

```

        $counts = [
            'PBB' =>
Pengaduan::where('bidang',
'Pajak bumi dan bangunan
(PBB)')->count(),
            'Umum' =>
Pengaduan::where('bidang',
'Umum')->count(),
            'Piutang' =>
Pengaduan::where('bidang',
'Piutang')->count(),
        ];

        // Menghitung jumlah
        penilaian untuk setiap bintang
        for ($i = 1; $i <= 5; $i++) {
            $ratings[$i] =
Pengaduan::join('tb_tanggaps'
, 'tb_tanggaps.id_pengaduan'
, '=', 'tb_pengaduans.id')
->where('tb_tanggaps.rating'
, $i)
->count();
            $totalRatings +=
$ratings[$i]; // Menjumlahkan
            total penilaian
        }

        $jumlah =
Pengaduan::count();

        return view('index', [
            'pengaduan' =>
Pengaduan::join('tb_tanggaps'
, 'tb_tanggaps.id_pengaduan'
, '=', 'tb_pengaduans.id')-
>orderBy('tgl_waktutanggapan'
, 'desc')->get(['tb_tanggaps.id'
as id_tanggapan'
, 'tb_tanggaps.*'
, 'tb_pengaduans.*']),
            'no' => 1,
            'ratings' => $ratings,
            'jumlah' => $jumlah,
            'totalRatings' =>
$totalRatings,
            'counts' => $counts,

```

```

        ]);
    }
    public function login()
    {
        return view('login');
    }

    public function
prosesLogin(Request $request)
    {
        if($request->captcha !=
session('captcha')){
            return back()-
>with('error','Captcha yang anda
masukan salah, coba lagi')-
>withInput();
        }
        $kredensial = $request-
>only('nip', 'password');
        if
(Auth::attempt($kredensial)) {
            $request->session()-
>regenerate();
            return
redirect(Route('dashboard'));
        }

        return back()->with('error',
'Nik atau password anda salah')-
>withInput();
    }

    public function dashboard()
    {
        if (in_array(Auth::user()-
>jabatan, ['Kepala BAPENDA',
'Admin'])) {
            $pengaduan =
Pengaduan::whereRaw('YEAR(
tgl_waktu) = ?', [date('Y')])-
>count();
            $tanggapan =
Pengaduan::join('tb_tanggaps'
, 'tb_tanggaps.id_pengaduan'
, '=', 'tb_pengaduans.id')
->whereRaw('YEAR(tgl_waktu)
= ?', [date('Y')])
->count();
            $grafikPengaduan =
Pengaduan::selectRaw('MONT

```

```

H(tgl_waktu) as bulan,
COUNT(id) as total')
->whereRaw('MONTH(tgl_wakt
u) >= ?', [1])
->whereRaw('MONTH(tgl_wakt
u) <= ?', [12])
->groupBy('bulan')
->orderBy('bulan'
, 'asc')
->limit(12)
->get();
            $grafikTanggapan =
Pengaduan::join('tb_tanggaps'
, 'tb_tanggaps.id_pengaduan'
, '=', 'tb_pengaduans.id')
->selectRaw('MONTH(tgl_wakt
u) as bulan,
COUNT(tb_tanggaps.id) as
total')
->whereRaw('MONTH(tgl_wakt
u) >= ?', [1])
->whereRaw('MONTH(tgl_wakt
u) <= ?', [12])
->groupBy('bulan')
->orderBy('bulan'
, 'asc')
->limit(12)
->get();
        } else {
            $pengaduan =
Pengaduan::where('bidang',
Auth::user()->bidang)
->whereRaw('YEAR(tgl_waktu)
= ?', [date('Y')])
->count();
            $tanggapan =
Pengaduan::join('tb_tanggaps'
, 'tb_tanggaps.id_pengaduan'
, '=', 'tb_pengaduans.id')
->where('bidang',
Auth::user()->bidang)
->whereRaw('YEAR(tgl_waktu)
= ?', [date('Y')])

```

```

->count();
$grafikPengaduan =
Pengaduan::selectRaw('MONTH(tgl_waktu) as bulan,
COUNT(id) as total')
->where('bidang',
Auth::user()->bidang)
->whereRaw('MONTH(tgl_waktu) >= ?', [1])
->whereRaw('MONTH(tgl_waktu) <= ?', [12])
->groupBy('bulan')
->orderBy('bulan',
'asc')
->limit(12)
->get();
$grafikTanggapan =
Pengaduan::join('tb_tanggapan', 'tb_tanggapan.id_pengaduan',
'=', 'tb_pengaduans.id')
->selectRaw('MONTH(tgl_waktu) as bulan,
COUNT(tb_tanggapan.id) as total')
->where('bidang',
Auth::user()->bidang)
->whereRaw('MONTH(tgl_waktu) >= ?', [1])
->whereRaw('MONTH(tgl_waktu) <= ?', [12])
->groupBy('bulan')
->orderBy('bulan',
'asc')
->limit(12)
->get();
}
return view('kantor.index', [
'jumlahpengaduan' =>
$pengaduan,
'jumlahtanggapan' =>
$tanggapan,
'grafikPengaduan' =>
$grafikPengaduan,
'grafikTanggapan' =>
$grafikTanggapan,

```

```

]);
}
public function
logout(Request $request)
{
Auth::logout();

$request->session()-
>invalidate();

$request->session()-
>regenerateToken();

return
redirect(Route('public'));
}
public function
create(Request $request,
Tanggapan $tanggapan)
{
if ($request->kode !==
$tanggapan->kode) {
return redirect()->back()-
>with('alert',
'<script>alert("Kode yang anda
masukan salah")</script>');
}

$tanggapan->update([
'rating' => $request-
>rating,
]);
return redirect()->back()-
>with('alert',
'<script>alert("Rating berhasil
dikirim")</script>');
}
}
<?php

namespace
App\Http\Controllers;

use App\Models\Pengaduan;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Aut
h;

```

```

use
Illuminate\Support\Facades\Val
idator;

date_default_timezone_set('Asi
a/Jakarta');
class PengaduanController
extends Controller
{
public function index()
{
if (Auth::user()->jabatan
!== 'Kepala BAPENDA') {
return
view('kantor.pengaduan', [
'pengaduan' =>
Pengaduan::where('bidang',
Auth::user()->bidang)
->orderByRaw("FIELD(status,
'Belum Ditanggapi') DESC")
->orderBy('status',
'asc')
->get(),
'no' => 1,
]);
return redirect()->back();
}

public function
create(Request $request)
{
if($request->captcha !==
session('captcha')){
return back()-
>with('error','Captcha yang anda
masukan salah, coba lagi')-
>withInput();
}

$validator =
Validator::make($request-
>all(), [
'upload_bukti' =>
'file|mimes:pdf,jpg|max:10000',
]);
if ($validator->fails()) {
return redirect()->back()-
>with('error', $validator-
>errors()->first())->withInput();
}
}
}

```

```

        $file = $request-
>upload_bukti;
        $extension = $file-
>getClientOriginalExtension();
        $fileName = time() . '_' .
rand() . '.' . $extension;
        $uploadPath =
'uploads/upload_bukti/';

        Pengaduan::create([
            'tgl_waktu' => date('Y-m-
d H:i:s'),
            'bidang' => $request-
>bidang,
            'nama_pengadu' =>
$request->nama_pengadu,
            'kategori' => $request-
>kategori,
            'perihal' => $request-
>perihal,
            'keterangan' => $request-
>keterangan,
            'upload_bukti' =>
$fileName,
            'email' => $request-
>email,
            'alamat' => $request-
>alamat,
            'status' => 'Belum
Ditanggapi',
        ]);
        $file->move($uploadPath,
$fileName);
        session(['captcha' => ""]);
        return redirect()->back()-
>with('pesan', 'Pengaduan
berhasil dikirim');
    }

    public function
laporan(Request $request)
    {
        $pengaduan =
Pengaduan::whereRaw('Date(tg
l_waktu) >= ?', [$request->tgl1])
-
>whereRaw('Date(tgl_waktu)
<= ?', [$request->tgl2]);

```

```

        (!empty($request->status) ?
        $pengaduan->where('status',
        $request->status) : "");

        return view('kantor.laporan-
pengaduan', [
            'pengaduan' =>
        $pengaduan->get(),
            'no' => 1,
        ]);
    }
}

<?php

namespace
App\Http\Controllers;

use App\Models\Pengaduan;
use App\Models>Tanggapan;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Aut
h;

date_default_timezone_set('Asi
a/Jakarta');
class TanggapanController
extends Controller
{
    public function index()
    {
        if (Auth::user()->jabatan
!=='Kepala BAPENDA') {
            return
view('kantor.tanggapan', [
                'pengaduan' =>
Pengaduan::where('bidang',
Auth::user()->bidang)
->where('status',
'Belum Ditanggapi')
->get(),
                'tanggapan' =>
Pengaduan::join('tb_tanggapan'
s', 'tb_tanggapan'
s.id_pengaduan',
'=', 'tb_pengaduans.id')
->where('id_user',
Auth::user()->id)
->get(),
                'no' => 1,

```

```

        ]);
    }
    return redirect()->back();
}

    public function
create(Request $request)
    {
        $kode = rand(000000,
999999);
        $create =
Tanggapan::create([
            'id_pengaduan' =>
$request->id_pengaduan,
            'id_user' => Auth::user()-
>id,
            'tgl_waktutanggapan' =>
date('Y-m-d H:i:s'),
            'tanggapan' => $request-
>tanggapan,
            'kode' => $kode,
            // 'rating' => $request-
>rating,
        ]);

        if ($create) {
            $pengaduan =
Pengaduan::find($request-
>id_pengaduan);
            $pengaduan-
>update(['status' => 'Sudah
Ditanggapi']);
            // Mengatur subjek dan
headers email
            $subjek = 'Notifikasi
Email';
            $headers = "From: Badan
Pendapatan Daerah
<email@bapenda.com>\r\n";
            $headers .= "MIME-
Version: 1.0\r\n";
            $headers .= "Content-
Type: text/html; charset=UTF-
8\r\n";

            // Konten email dengan
format HTML
            $pesan = "
<html>
<head>

```

```

        <title>Notifikasi
Pengaduan</title>
    </head>
    <body>
        <p>Halo, " .
htmlspecialchars($pengaduan-
>nama_pengadu,
ENT_QUOTES, 'UTF-8') .
".</p>
        <p>Terima kasih atas
pengaduan yang Anda laporkan.
Pengaduan Anda telah selesai
ditanggapi. Silakan lihat
tanggapan di website, dan
berikan rating berdasarkan
pendapat Anda.</p>
        <p>Berikan penilaian
dengan menggunakan kode di
bawah ini:</p>
        <br>
        <center><h2>" .
htmlspecialchars($kode,
ENT_QUOTES, 'UTF-8') .
"</h2></center>
    </body>
</html>";

```

```

// Menampilkan pesan di
layar (untuk debugging, dapat
dihapus jika tidak diperlukan)
    echo $pesan;

// Mengirim email
    if (mail($pengaduan-
>email, $subjek, $pesan,
$headers)) {
        // Email berhasil
        dikirim, arahkan kembali
        dengan pesan sukses
        return redirect()-
>back()->with('pesan',
'Pengaduan berhasil
ditanggapi');
    } else {
        // Email gagal dikirim,
        arahkan kembali dengan pesan
        error
        return redirect()-
>back()->with('error', 'Gagal
mengirim email notifikasi.');
```

```

    }
    }
    return redirect()->back()-
>with('error', 'Pengaduan gagal
ditanggapi');
    }
}

<?php
namespace
App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Aut
h;

class UserController extends
Controller
{
    public function index()
    {
        if (Auth::user()->jabatan ==
'Kepala BAPENDA') {
            return view('kantor.user',
[
                'user' =>
User::orderBy('jabatan', 'asc')-
>get(),
                'no' => 1,
            ]);
        }
        return redirect()->back();
    }

    public function
create(Request $request)
    {
        if (User::where('nip',
$request->nip)->exists()) {
            return redirect()->back()-
>with('error', 'Nip telah
digunakan');
        }
        user::create([
            'nip' => $request->nip,
            'nama_pegawai' =>
$request->nama_pegawai,
```

```

            'bidang' => $request-
>bidang,
            'jabatan' => $request-
>jabatan,
            'password' =>
bcrypt($request->password),
        ]);
        return redirect()->back()-
>with('pesan', 'Data berhasil
disimpan');
    }

    public function
update(Request $request, User
$user)
    {
        if ($request->nip !== $user-
>nip && User::where('nip',
$request->nip)->exists()) {
            return redirect()->back()-
>with('error', 'Nip telah
digunakan');
        }
        $user->update([
            'nip' => $request->nip,
            'nama_pegawai' =>
$request->nama_pegawai,
            'bidang' => $request-
>bidang,
            'jabatan' => $request-
>jabatan,
        ]);

        return redirect()->back()-
>with('pesan', 'Data berhasil
diubah');
    }

    public function
delete(User
$user)
    {
        $user->delete();
        return redirect()->back()-
>with('pesan', 'Data berhasil
dihapus');
    }

    public function
ubahPassword(Request
$request, User $user)
    {
        if ($request->p1 !==
$request->p2) {
```



```

use
Symfony\Component\HttpFoundation\Response;

class RedirectIfAuthenticated
{
    /**
     * Handle an incoming
     request.
     *
     *
     * @param \Closure(\Illuminate\
    Http\Request):
    (\Symfony\Component\HttpFoundation\Response) $next
    */

    public function
    handle(Request $request,
    Closure $next, string ...$guards):
    Response
    {
        $guards = empty($guards)
        ? [null] : $guards;

        foreach ($guards as $guard)
        {
            if (Auth::guard($guard)-
            >check()) {
                return
                redirect(RouteServiceProvider::
                HOME);
            }
        }

        return $next($request);
    }
}

<?php

namespace
App\Http\Middleware;

use
Illuminate\Foundation\Http\Mid
dleware\TrimStrings as
Middleware;

class TrimStrings extends
Middleware

```

```

{
    /**
     * The names of the attributes
     that should not be trimmed.
     *
     * @var array<int, string>
     */
    protected $except = [
        'current_password',
        'password',
        'password_confirmation',
    ];
}

<?php

namespace
App\Http\Middleware;

use
Illuminate\Http\Middleware\Tr
ustHosts as Middleware;

class TrustHosts extends
Middleware
{
    /**
     * Get the host patterns that
     should be trusted.
     *
     * @return array<int,
    string|null>
     */
    public function hosts(): array
    {
        return [
            $this-
            >allSubdomainsOfApplication
            Url(),
        ];
    }
}

<?php

namespace
App\Http\Middleware;

```

```

use
Illuminate\Http\Middleware\Tr
ustProxies as Middleware;
use Illuminate\Http\Request;

class TrustProxies extends
Middleware
{
    /**
     * The trusted proxies for this
     application.
     *
     * @var array<int,
    string>|string|null
     */
    protected $proxies;

    /**
     * The headers that should be
     used to detect proxies.
     *
     * @var int
     */
    protected $headers =
        Request::HEADER_X_FO
        RWARDED_FOR |
        Request::HEADER_X_FO
        RWARDED_HOST |
        Request::HEADER_X_FO
        RWARDED_PORT |
        Request::HEADER_X_FO
        RWARDED_PROTO |
        Request::HEADER_X_FO
        RWARDED_AWS_ELB;
}

<?php

namespace
App\Http\Middleware;

use
Illuminate\Routing\Middleware
\ValidateSignature as
Middleware;

class ValidateSignature extends
Middleware
{
    /**

```

* The names of the query string parameters that should be ignored.

```
*
* @var array<int, string>
*/
protected $except = [
    // 'fbclid',
    // 'utm_campaign',
    // 'utm_content',
    // 'utm_medium',
    // 'utm_source',
    // 'utm_term',
];
}
```

<?php

```
namespace
App\Http\Middleware;

use
Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as
Middleware;
```

```
class VerifyCsrfToken extends
Middleware
{
    /**
     * The URIs that should be
     excluded from CSRF
     verification.
```

```
*
* @var array<int, string>
*/
protected $except = [
    //
];
}
```

<?php

```
namespace App\Http;

use
Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
```

```
{
    /**
     * The application's global
     HTTP middleware stack.
     *
     * These middleware are run
     during every request to your
     application.
```

```
*
* @var array<int, class-string|string>
*/
```

```
protected $middleware = [
    //
    \App\Http\Middleware\TrustHosts::class,
    \App\Http\Middleware\TrustProxies::class,
    \Illuminate\Http\Middleware\HandleCors::class,
    \App\Http\Middleware\PreventRequestsDuringMaintenance::class,
    \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
    \App\Http\Middleware\TrimStrings::class,
    \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
];
```

```
/**
* The application's route
middleware groups.
```

```
*
* @var array<string, array<int, class-string|string>>
*/
```

```
protected $middlewareGroups
= [
    'web' => [
        \App\Http\Middleware\EncryptCookies::class,
        \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
        \Illuminate\Session\Middleware\StartSession::class,
```

```
\Illuminate\View\Middleware\ShareErrorsFromSession::class,
        \App\Http\Middleware\VerifyCsrfToken::class,
        \Illuminate\Routing\Middleware\SubstituteBindings::class,
    ],
```

```
'api' => [
    //
    \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,
    \Illuminate\Routing\Middleware\ThrottleRequests::class . ':api',
    \Illuminate\Routing\Middleware\SubstituteBindings::class,
],
];
```

```
/**
* The application's
middleware aliases.
```

```
*
* Aliases may be used to
conveniently assign middleware
to routes and groups.
```

```
*
* @var array<string, class-string|string>
*/
```

```
protected $middlewareAliases
= [
    'auth' =>
    \App\Http\Middleware\Authenticate::class,
    'auth.basic' =>
    \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'auth.session' =>
    \Illuminate\Session\Middleware\AuthenticateSession::class,
    'cache.headers' =>
    \Illuminate\Http\Middleware\SetCacheHeaders::class,
```

```

        'can' =>
\Illuminate\Auth\Middleware\A
uthorize::class,
        'guest' =>
\App\Http\Middleware\Redirect
IfAuthenticated::class,
        'password.confirm' =>
\Illuminate\Auth\Middleware\R
equirePassword::class,
        'signed' =>
\App\Http\Middleware\Validate
Signature::class,
        'throttle' =>
\Illuminate\Routing\Middlewar
e\ThrottleRequests::class,
        'verified' =>
\Illuminate\Auth\Middleware\E
nsureEmailIsVerified::class,
        'cekUserLogin' =>
\App\Http\Middleware\CekUser
Login::class,
    ];
}

```

```
<?php
```

```
namespace App\Models;
```

```

use
Illuminate\Database\Eloquent\F
actories\HasFactory;
use
Illuminate\Database\Eloquent\
Model;

```

```

class Pengaduan extends Model
{
    use HasFactory;
    protected $table =
'tb_pengaduans';
    protected $primaryKey = 'id';
    protected $fillable = [
        'tgl_waktu',
        'bidang',
        'nama_pengadu',
        'kategori',
        'perihal',
        'keterangan',
        'upload_bukti',
        'email',

```

```

        'alamat',
        'status',
    ];
}

```

```
<?php
```

```
namespace App\Models;
```

```

use
Illuminate\Database\Eloquent\F
actories\HasFactory;
use
Illuminate\Database\Eloquent\
Model;

```

```

class Tanggapan extends Model
{
    use HasFactory;
    protected $table =
'tb_tanggapan';
    protected $primaryKey = 'id';
    protected $fillable = [
        'id_pengaduan',
        'id_user',
        'tgl_waktutanggapan',
        'tanggapan',
        'kode',
        'rating',
    ];
}

```

```
<?php
```

```
namespace App\Models;
```

```

//
use
Illuminate\Contracts\Auth\Must
VerifyEmail;
use
Illuminate\Database\Eloquent\F
actories\HasFactory;
use
Illuminate\Foundation\Auth\Us
er as Authenticatable;
use
Illuminate\Notifications\Notifia
ble;

```

```

use
Laravel\Sanctum\HasApiToken
s;

```

```

class User extends
Authenticatable
{
    use HasApiTokens,
    HasFactory, Notifiable;

```

```

    protected $table = 'tb_users';
    protected $fillable = [
        'nip',
        'nama_pegawai',
        'bidang',
        'jabatan',
        'password',
    ];

```

```

    protected $hidden = [
        'password',
        'remember_token',
    ];
}

```

```
<?php
```

```
namespace App\Providers;
```

```

use
Illuminate\Support\ServiceProv
ider;

```

```

class AppServiceProvider
extends ServiceProvider
{
    /**
     * Register any application
services.
     */
    public function register(): void
    {
        //
    }

    /**
     * Bootstrap any application
services.
     */
}

```

```

        public function boot(): void
        {
            //
        }
    }
<?php

namespace App\Providers;

// use
Illuminate\Support\Facades\Gate;
use
Illuminate\Foundation\Support\
Providers\AuthServiceProvider
as ServiceProvider;

class AuthServiceProvider
extends ServiceProvider
{
    /**
     * The model to policy
     mappings for the application.
     *
     * @var array<class-string,
     class-string>
     */
    protected $policies = [
        // 'App\Models\Model' =>
        'App\Policies\ModelPolicy',
    ];

    /**
     * Register any authentication
     / authorization services.
     */
    public function boot(): void
    {
        $this->registerPolicies();

        //
    }
}
<?php

namespace App\Providers;

```

```

use
Illuminate\Support\Facades\Bro
adcast;
use
Illuminate\Support\ServiceProv
ider;

class BroadcastServiceProvider
extends ServiceProvider
{
    /**
     * Bootstrap any application
     services.
     */
    public function boot(): void
    {
        Broadcast::routes();

        require
        base_path('routes/channels.php'
        );
    }
}
<?php

namespace App\Providers;

use
Illuminate\Auth\Events\Registe
red;
use
Illuminate\Auth\Listeners\Send
EmailVerificationNotification;
use
Illuminate\Foundation\Support\
Providers\EventServiceProvider
as ServiceProvider;
use
Illuminate\Support\Facades\Eve
nt;

class EventServiceProvider
extends ServiceProvider
{
    /**
     * The event to listener
     mappings for the application.
     *

```

```

     * @var array<class-string,
     array<int, class-string>>
     */
    protected $listen = [
        Registered::class => [
            SendEmailVerificationN
            otification::class,
        ],
    ];

    /**
     * Register any events for your
     application.
     */
    public function boot(): void
    {
        //
    }

    /**
     * Determine if events and
     listeners should be automatically
     discovered.
     */
    public function
    shouldDiscoverEvents(): bool
    {
        return false;
    }
}
<?php

namespace App\Providers;

use
Illuminate\Cache\RateLimiting\
Limit;
use
Illuminate\Foundation\Support\
Providers\RouteServiceProvider
as ServiceProvider;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Rat
eLimiter;
use
Illuminate\Support\Facades\Ro
ute;

```

```

class RouteServiceProvider
extends ServiceProvider
{
    /**
     * The path to the "home"
     route for your application.
     *
     * Typically, users are
     redirected here after
     authentication.
     *
     * @var string
     */
    public const HOME = '/home';

    /**
     * Define your route model
     bindings, pattern filters, and
     other route configuration.
     */
    public function boot(): void
    {
        $this->configureRateLimiting();

        $this->routes(function () {
            Route::middleware('api')
                ->prefix('api')
                -
        >group(base_path('routes/api.ph
        p'));

            Route::middleware('web')
            -
        >group(base_path('routes/web.p
        hp'));
        });
    }

    /**
     * Configure the rate limiters
     for the application.
     */
    protected function
    configureRateLimiting(): void
    {
        RateLimiter::for('api',
        function (Request $request) {
            return
            Limit::perMinute(60)-

```

```

>by($request->user()?->id    ?:
$request->ip());
        });
    }

<?php return array (
    'laravel/sail' =>
        array (
            'providers' =>
                array (
                    0 =>
'Laravel\Sail\SailServiceProvi
der',
                ),
            'laravel/sanctum' =>
                array (
                    'providers' =>
                        array (
                            0 =>
'Laravel\Sanctum\SanctumSer
viceProvider',
                        ),
                    ),
                    'laravel/tinker' =>
                        array (
                            'providers' =>
                                array (
                                    0 =>
'Laravel\Tinker\TinkerService
Provider',
                                ),
                                ),
                                'nesbot/carbon' =>
                                    array (
                                        'providers' =>
                                            array (
                                                0 =>
'Carbon\Laravel\ServiceProvid
er',
                                            ),
                                            ),
                                            'nunomaduro/collision' =>
                                                array (
                                                    'providers' =>
                                                        array (
                                                            0 =>
'NunoMaduro\Collision\Adapt

```

```

ers\Laravel\CollisionServicePr
ovider',
                ),
                ),
                'nunomaduro/termwind' =>
                    array (
                        'providers' =>
                            array (
                                0 =>
'Termwind\Laravel\Termwind
ServiceProvider',
                            ),
                            ),
                            'spatie/laravel-ignition' =>
                                array (
                                    'providers' =>
                                        array (
                                            0 =>
'Spatie\Laravel\Ignition\Ignitio
nServiceProvider',
                                        ),
                                        'aliases' =>
                                            array (
                                                'Flare' =>
'Spatie\Laravel\Ignition\Facade
s\Flare',
                                            ),
                                            ),
                                        );

<?php return array (
    'providers' =>
        array (
            0 =>
'Illuminate\Auth\AuthServiceP
rovider',
            1 =>
'Illuminate\Broadcasting\Broad
castServiceProvider',
            2 =>
'Illuminate\Bus\BusServicePro
vider',
            3 =>
'Illuminate\Cache\CacheServic
eProvider',
            4 =>
'Illuminate\Foundation\Provid
ers\ConsoleSupportServicePro
vider',

```

5 =>
'Illuminate\\Cookie\\CookieServiceProvider',

6 =>
'Illuminate\\Database\\DatabaseServiceProvider',

7 =>
'Illuminate\\Encryption\\EncryptionServiceProvider',

8 =>
'Illuminate\\Filesystem\\FilesystemServiceProvider',

9 =>
'Illuminate\\Foundation\\Providers\\FoundationServiceProvider',

10 =>
'Illuminate\\Hashing\\HashServiceProvider',

11 =>
'Illuminate\\Mail\\MailServiceProvider',

12 =>
'Illuminate\\Notifications\\NotificationServiceProvider',

13 =>
'Illuminate\\Pagination\\PaginationServiceProvider',

14 =>
'Illuminate\\Pipeline\\PipelineServiceProvider',

15 =>
'Illuminate\\Queue\\QueueServiceProvider',

16 =>
'Illuminate\\Redis\\RedisServiceProvider',

17 =>
'Illuminate\\Auth\\Passwords\\PasswordResetServiceProvider',

18 =>
'Illuminate\\Session\\SessionServiceProvider',

19 =>
'Illuminate\\Translation\\TranslationServiceProvider',

20 =>
'Illuminate\\Validation\\ValidationServiceProvider',

21 =>
'Illuminate\\View\\ViewServiceProvider',

22 =>
'Laravel\\Sail\\SailServiceProvider',

23 =>
'Laravel\\Sanctum\\SanctumServiceProvider',

24 =>
'Laravel\\Tinker\\TinkerServiceProvider',

25 =>
'Carbon\\Laravel\\ServiceProvider',

26 =>
'NunoMaduro\\Collision\\Adapters\\Laravel\\CollisionServiceProvider',

27 =>
'Termwind\\Laravel\\TermwindServiceProvider',

28 =>
'Spatie\\LaravelIgnition\\IgnitionServiceProvider',

29 =>
'App\\Providers\\AppServiceProvider',

30 =>
'App\\Providers\\AuthServiceProvider',

31 =>
'App\\Providers\\EventServiceProvider',

32 =>
'App\\Providers\\RouteServiceProvider',
)
,
'eager' =>
array (
0 =>
'Illuminate\\Auth\\AuthServiceProvider',
1 =>
'Illuminate\\Cookie\\CookieServiceProvider',
2 =>
'Illuminate\\Database\\DatabaseServiceProvider',

3 =>
'Illuminate\\Encryption\\EncryptionServiceProvider',

4 =>
'Illuminate\\Filesystem\\FilesystemServiceProvider',

5 =>
'Illuminate\\Foundation\\Providers\\FoundationServiceProvider',

6 =>
'Illuminate\\Notifications\\NotificationServiceProvider',

7 =>
'Illuminate\\Pagination\\PaginationServiceProvider',

8 =>
'Illuminate\\Session\\SessionServiceProvider',

9 =>
'Illuminate\\View\\ViewServiceProvider',

10 =>
'Laravel\\Sanctum\\SanctumServiceProvider',

11 =>
'Carbon\\Laravel\\ServiceProvider',

12 =>
'NunoMaduro\\Collision\\Adapters\\Laravel\\CollisionServiceProvider',

13 =>
'Termwind\\Laravel\\TermwindServiceProvider',

14 =>
'Spatie\\LaravelIgnition\\IgnitionServiceProvider',

15 =>
'App\\Providers\\AppServiceProvider',

16 =>
'App\\Providers\\AuthServiceProvider',

17 =>
'App\\Providers\\EventServiceProvider',

18 =>
'App\\Providers\\RouteServiceProvider',

```

),
'deferred' =>
array (
    'Illuminate\\Broadcasting\\Broad-
    castManager' =>
    'Illuminate\\Broadcasting\\Broad-
    castServiceProvider',
    'Illuminate\\Contracts\\Broad-
    casting\\Factory' =>
    'Illuminate\\Broadcasting\\Broad-
    castServiceProvider',
    'Illuminate\\Contracts\\Broad-
    casting\\Broadcaster' =>
    'Illuminate\\Broadcasting\\Broad-
    castServiceProvider',
    'Illuminate\\Bus\\Dispatcher'
=>
'Illuminate\\Bus\\BusServicePro-
vider',
    'Illuminate\\Contracts\\Bus\\
    Dispatcher' =>
    'Illuminate\\Bus\\BusServicePro-
    vider',
    'Illuminate\\Contracts\\Bus\\
    QueueingDispatcher' =>
    'Illuminate\\Bus\\BusServicePro-
    vider',
    'Illuminate\\Bus\\BatchRepos-
    itory' =>
    'Illuminate\\Bus\\BusServicePro-
    vider',
    'Illuminate\\Bus\\DatabaseBa-
    tchRepository' =>
    'Illuminate\\Bus\\BusServicePro-
    vider',
    'cache' =>
    'Illuminate\\Cache\\CacheServic-
    eProvider',
    'cache.store' =>
    'Illuminate\\Cache\\CacheServic-
    eProvider',
    'cache.psr6' =>
    'Illuminate\\Cache\\CacheServic-
    eProvider',
    'memcached.connector' =>
    'Illuminate\\Cache\\CacheServic-
    eProvider',
    'Illuminate\\Cache\\RateLimit-
    er' =>

```

```

'Illuminate\\Cache\\CacheServic-
eProvider',
    'Illuminate\\Foundation\\Con-
    sole\\AboutCommand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Cache\\Console\\
    ClearCommand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Cache\\Console\\
    ForgetCommand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Foundation\\Con-
    sole\\ClearCompiledCommand'
=>
'Illuminate\\Foundation\\Provid-
ers\\ConsoleSupportServicePro-
vider',
    'Illuminate\\Auth\\Console\\C-
    learResetsCommand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Foundation\\Con-
    sole\\ConfigCacheCommand'
=>
'Illuminate\\Foundation\\Provid-
ers\\ConsoleSupportServicePro-
vider',
    'Illuminate\\Foundation\\Con-
    sole\\ConfigClearCommand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Foundation\\Con-
    sole\\ConfigShowCommand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Database\\Consol-
    e\\DbCommand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',

```

```

'Illuminate\\Database\\Consol-
e\\MonitorCommand' =>
'Illuminate\\Foundation\\Provid-
ers\\ConsoleSupportServicePro-
vider',
    'Illuminate\\Database\\Consol-
    e\\PruneCommand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Database\\Consol-
    e\\ShowCommand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Database\\Consol-
    e\\TableCommand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Database\\Consol-
    e\\WipeCommand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Foundation\\Con-
    sole\\DownCommand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Foundation\\Con-
    sole\\EnvironmentCommand'
=>
'Illuminate\\Foundation\\Provid-
ers\\ConsoleSupportServicePro-
vider',
    'Illuminate\\Foundation\\Con-
    sole\\EnvironmentDecryptCom-
    mand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Foundation\\Con-
    sole\\EnvironmentEncryptCom-
    mand' =>
    'Illuminate\\Foundation\\Provid-
    ers\\ConsoleSupportServicePro-
    vider',
    'Illuminate\\Foundation\\Con-
    sole\\EventCacheCommand' =>

```

```

'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\EventClearCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\EventListCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\KeyGenerateCommand'
=>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\OptimizeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\OptimizeClearCommand'
=>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\PackageDiscoverCommand'
=>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Cache\\Console\\PruneStaleTagsCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Queue\\Console\\ClearCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Queue\\Console\\ListFailedCommand' =>
'Illuminate\\Foundation\\Provid

```

```

ers\\ConsoleSupportServiceProvider',
    'Illuminate\\Queue\\Console\\FlushFailedCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Queue\\Console\\ForgetFailedCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Queue\\Console\\ListenCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Queue\\Console\\MonitorCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Queue\\Console\\PruneBatchesCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Queue\\Console\\PruneFailedJobsCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Queue\\Console\\RestartCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Queue\\Console\\RetryCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Queue\\Console\\RetryBatchCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Queue\\Console\\WorkCommand' =>
'Illuminate\\Foundation\\Provid

```

```

ers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\RouteCacheCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\RouteClearCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Foundation\\Console\\RouteListCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Database\\Console\\DumpCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Database\\Console\\Seeds\\SeedCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Console\\Scheduling\\ScheduleFinishCommand'
=>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Console\\Scheduling\\ScheduleListCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Console\\Scheduling\\ScheduleRunCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Console\\Scheduling\\ScheduleClearCacheCommand'
=>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',

```

'Illuminate\Console\Scheduling\ScheduleTestCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Console\Scheduling\ScheduleWorkCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Console\Scheduling\ScheduleInterruptCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Database\Console\ShowModelCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\StorageLinkCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\StorageUnlinkCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\UpCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\ViewCacheCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\ViewClearCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Cache\Console\CacheTableCommand' =>

'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\CastMakeCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\ChannelListCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\ChannelMakeCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\ComponentMakeCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\ConsoleMakeCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Routing\Console\ControllerMakeCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\DocsCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\EventGenerateCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\EventMakeCommand' =>
 'Illuminate\Foundation\Provid

ers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\ExceptionMakeCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Database\Console\Factories\FactoryMakeCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\JobMakeCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\LangPublishCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\ListenerMakeCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\MailMakeCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Routing\Console\MiddlewareMakeCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\ModelMakeCommand' =>
 'Illuminate\Foundation\Providers\ConsoleSupportServiceProvider',
 'Illuminate\Foundation\Console\NotificationMakeCommand' =>
 'Illuminate\Foundation\Provid

```

ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Notifications\\Co
nsole\\NotificationTableComma
nd' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Foundation\\Con
sole\\ObserverMakeCommand'
=>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Foundation\\Con
sole\\PolicyMakeCommand' =>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Foundation\\Con
sole\\ProviderMakeCommand'
=>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Queue\\Console\\
FailedTableCommand' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Queue\\Console\\
TableCommand' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Queue\\Console\\
BatchesTableCommand' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Foundation\\Con
sole\\RequestMakeCommand'
=>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Foundation\\Con
sole\\ResourceMakeCommand'
=>
'Illuminate\\Foundation\\Provid

```

```

ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Foundation\\Con
sole\\RuleMakeCommand' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Foundation\\Con
sole\\ScopeMakeCommand' =>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Database\\Consol
e\\Seeds\\SeederMakeComman
d' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Session\\Console
\\SessionTableCommand' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Foundation\\Con
sole\\ServeCommand' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Foundation\\Con
sole\\StubPublishCommand' =>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Foundation\\Con
sole\\TestMakeCommand' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Foundation\\Con
sole\\VendorPublishCommand'
=>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Foundation\\Con
sole\\ViewMakeCommand' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',

```

```

    'migrator' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'migration.repository' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'migration.creator' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Database\\Consol
e\\Migrations\\MigrateComman
d' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Database\\Consol
e\\Migrations\\FreshCommand'
=>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Database\\Consol
e\\Migrations\\InstallCommand'
=>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Database\\Consol
e\\Migrations\\RefreshComman
d' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Database\\Consol
e\\Migrations\\ResetCommand'
=>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Database\\Consol
e\\Migrations\\RollbackComma
nd' ==>
'Illuminate\\Foundation\\Provid
ers\\ConsoleSupportServicePro
vider',
    'Illuminate\\Database\\Consol
e\\Migrations\\StatusCommand'

```

```

=>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'Illuminate\\Database\\Console\\Migrations\\MigrateMakeCommand' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'composer' =>
'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider',
    'hash' =>
'Illuminate\\Hashing\\HashServiceProvider',
    'hash.driver' =>
'Illuminate\\Hashing\\HashServiceProvider',
    'mail.manager' =>
'Illuminate\\Mail\\MailServiceProvider',
    'mailer' =>
'Illuminate\\Mail\\MailServiceProvider',
    'Illuminate\\Mail\\Markdown' =>
'Illuminate\\Mail\\MailServiceProvider',
    'Illuminate\\Contracts\\Pipeline\\Hub' =>
'Illuminate\\Pipeline\\PipelineServiceProvider',
    'pipeline' =>
'Illuminate\\Pipeline\\PipelineServiceProvider',
    'queue' =>
'Illuminate\\Queue\\QueueServiceProvider',
    'queue.connection' =>
'Illuminate\\Queue\\QueueServiceProvider',
    'queue.failer' =>
'Illuminate\\Queue\\QueueServiceProvider',
    'queue.listener' =>
'Illuminate\\Queue\\QueueServiceProvider',

```

```

    'queue.worker' =>
'Illuminate\\Queue\\QueueServiceProvider',
    'redis' =>
'Illuminate\\Redis\\RedisServiceProvider',
    'redis.connection' =>
'Illuminate\\Redis\\RedisServiceProvider',
    'auth.password' =>
'Illuminate\\Auth\\Passwords\\PasswordResetServiceProvider',
    'auth.password.broker' =>
'Illuminate\\Auth\\Passwords\\PasswordResetServiceProvider',
    'translator' =>
'Illuminate\\Translation\\TranslationServiceProvider',
    'translation.loader' =>
'Illuminate\\Translation\\TranslationServiceProvider',
    'validator' =>
'Illuminate\\Validation\\ValidationServiceProvider',
    'validation.presence' =>
'Illuminate\\Validation\\ValidationServiceProvider',
    'Illuminate\\Contracts\\Validation\\UncompromisedVerifier' =>
'Illuminate\\Validation\\ValidationServiceProvider',
    'Laravel\\Sail\\Console\\InstallCommand' =>
'Illuminate\\Sail\\SailServiceProvider',
    'Laravel\\Sail\\Console\\PublishCommand' =>
'Illuminate\\Sail\\SailServiceProvider',
    'command.tinker' =>
'Illuminate\\Tinker\\TinkerServiceProvider',
    'when' =>
    array (
        'Illuminate\\Broadcasting\\BroadcastServiceProvider' =>
        array (

```

```

        'Illuminate\\Bus\\BusServiceProvider' =>
        array (
        ),
        'Illuminate\\Cache\\CacheServiceProvider' =>
        array (
        ),
        'Illuminate\\Foundation\\Providers\\ConsoleSupportServiceProvider' =>
        array (
        ),
        'Illuminate\\Hashing\\HashServiceProvider' =>
        array (
        ),
        'Illuminate\\Mail\\MailServiceProvider' =>
        array (
        ),
        'Illuminate\\Pipeline\\PipelineServiceProvider' =>
        array (
        ),
        'Illuminate\\Queue\\QueueServiceProvider' =>
        array (
        ),
        'Illuminate\\Redis\\RedisServiceProvider' =>
        array (
        ),
        'Illuminate\\Auth\\Passwords\\PasswordResetServiceProvider' =>
        array (
        ),
        'Illuminate\\Translation\\TranslationServiceProvider' =>
        array (
        ),
        'Illuminate\\Validation\\ValidationServiceProvider' =>
        array (
        ),
        'Laravel\\Sail\\SailServiceProvider' =>
        array (
        ),

```

```
'Laravel\\Tinker\\TinkerServiceProvider' =>
    array (
    ),
);
```

```
<?php
```

```
/*
|-----
|-----
| Create The Application
|-----
|-----
|
| The first thing we will do is
| create a new Laravel application
| instance
| which serves as the "glue" for
| all the components of Laravel,
| and is
| the IoC container for the system
| binding all of the various parts.
|
*/
```

```
$app = new
Illuminate\Foundation\Application(
    $_ENV['APP_BASE_PATH']
    ?? dirname(__DIR__)
);
```

```
/*
|-----
|-----
| Bind Important Interfaces
|-----
|-----
|
| Next, we need to bind some
| important interfaces into the
| container so
| we will be able to resolve them
| when needed. The kernels serve the
```

```
| incoming requests to this
| application from both the web
| and CLI.
```

```
|
*/
```

```
$app->singleton(
    Illuminate\Contracts\Http\Kernel::class,
    App\Http\Kernel::class
);
```

```
$app->singleton(
    Illuminate\Contracts\Console\Kernel::class,
    App\Console\Kernel::class
);
```

```
$app->singleton(
    Illuminate\Contracts\Debug\ExceptionHandler::class,
    App\Exceptions\Handler::class
);
```

```
/*
|-----
|-----
```

```
| Return The Application
```

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

```
|
| This script returns the
| application instance. The
| instance is given to
| the calling script so we can
| separate the building of the
| instances
| from the actual running of the
| application and sending
| responses.
|
*/
```

```
return $app;
```

```
<?php
```

```
use
Illuminate\Support\Facades\Facade;
```

```
return [
```

```
    /*
    |-----
    |-----
    | Application Name
    |-----
    |-----
```

```
|
| This value is the name of
| your application. This value is
| used when the
| framework needs to place the
| application's name in a
| notification or
| any other location as required
| by the application or its
| packages.
```

```
|
*/
```

```
'name' => env('APP_NAME',
'Laravel'),
```

```
/*
|-----
|-----
```

```
| Application Environment
|-----
|-----
```

```
|
| This value determines the
| "environment" your application
| is currently
| running in. This may
| determine how you prefer to
| configure various
| services the application
| utilizes. Set this in your ".env"
| file.
```

```
|
*/
```

```
'env' => env('APP_ENV',
'production'),
```

```

/*
|-----
|
| Application Debug Mode
|-----
|
| When your application is in
debug mode, detailed error
messages with
| stack traces will be shown on
every error that occurs within
your
| application. If disabled, a
simple generic error page is
shown.
|
*/

```

```

'debug' => (bool)
env('APP_DEBUG', false),

```

```

/*
|-----
|
| Application URL
|-----
|
| This URL is used by the
console to properly generate
URLs when using
| the Artisan command line
tool. You should set this to the
root of
| your application so that it is
used when running Artisan
tasks.
|
*/

```

```

'url' => env('APP_URL',
'http://localhost'),

```

```

'asset_url' =>
env('ASSET_URL', '/'),

```

```

/*
|-----
|

```

```

| Application Timezone
|-----
|
| Here you may specify the
default timezone for your
application, which
| will be used by the PHP date
and date-time functions. We
have gone
| ahead and set this to a
sensible default for you out of
the box.
|
*/

```

```

'timezone' => 'UTC',

```

```

/*
|-----
|
| Application Locale
Configuration
|-----
|

```

```

| The application locale
determines the default locale
that will be used
| by the translation service
provider. You are free to set this
value
| to any of the locales which
will be supported by the
application.
|
*/

```

```

'locale' => 'en',

```

```

/*
|-----
|

```

```

| Application Fallback Locale
|-----
|

```

```

| The fallback locale
determines the locale to use
when the current one

```

| is not available. You may
change the value to correspond
to any of
| the language folders that are
provided through your
application.
|

```

*/

```

```

'fallback_locale' => 'en',

```

```

/*
|-----
|

```

```

| Faker Locale
|-----
|

```

```

| This locale will be used by
the Faker PHP library when
generating fake
| data for your database seeds.
For example, this will be used to
get
| localized telephone numbers,
street address information and
more.
|
*/

```

```

'faker_locale' => 'en_US',

```

```

/*
|-----
|

```

```

| Encryption Key
|-----
|

```

```

'faker_locale' => 'en_US',

```

```

/*
|-----
|

```

```

| Encryption Key
|-----
|

```

```

| This key is used by the
Illuminate encrypter service and
should be set
| to a random, 32 character
string, otherwise these encrypted
strings
| will not be safe. Please do
this before deploying an
application!
|
*/

```

```

| Application Fallback Locale
|-----
|

```

```

| The fallback locale
determines the locale to use
when the current one

```

```

*/

```

```

'key' => env('APP_KEY'),

'cipher' => 'AES-256-CBC',

/*
|-----
| Maintenance Mode Driver
|-----
|
| These configuration options
determine the driver used to
determine and
| manage Laravel's
"maintenance mode" status. The
"cache" driver will
| allow maintenance mode to
be controlled across multiple
machines.
|
| Supported drivers: "file",
"cache"
|
*/

'maintenance' => [
    'driver' => 'file',
    // 'store' => 'redis',
],

/*
|-----
| Autoloaded Service
Providers
|-----
|
| The service providers listed
here will be automatically
loaded on the
| request to your application.
Feel free to add your own
services to
| this array to grant expanded
functionality to your
applications.
|

```

```

*/

'providers' => [

    /*
        * Laravel Framework
        Service Providers...
    */
    Illuminate\Auth\AuthServiceProvider::class,
    Illuminate\Broadcasting\BroadcastServiceProvider::class,
    Illuminate\Bus\BusServiceProvider::class,
    Illuminate\Cache\CacheServiceProvider::class,
    Illuminate\Foundation\Providers\ConsoleSupportServiceProvider::class,
    Illuminate\Cookie\CookieServiceProvider::class,
    Illuminate\Database\DatabaseServiceProvider::class,
    Illuminate\Encryption\EncryptionServiceProvider::class,
    Illuminate\Filesystem\FilesystemServiceProvider::class,
    Illuminate\Foundation\Providers\FoundationServiceProvider::class,
    Illuminate\Hashing\HashServiceProvider::class,
    Illuminate\Mail\MailServiceProvider::class,
    Illuminate\Notifications\NotificationServiceProvider::class,
    Illuminate\Pagination\PaginationServiceProvider::class,
    Illuminate\Pipeline\PipelineServiceProvider::class,
    Illuminate\Queue\QueueServiceProvider::class,
    Illuminate\Redis\RedisServiceProvider::class,
    Illuminate\Auth\Passwords\PasswordResetServiceProvider::class,
    Illuminate\Session\SessionServiceProvider::class,

```

```

    Illuminate\Translation\TranslationServiceProvider::class,
    Illuminate\Validation\ValidationServiceProvider::class,
    Illuminate\View\ViewServiceProvider::class,

    /*
        * Package Service
        Providers...
    */

    /*
        * Application Service
        Providers...
    */
    App\Providers\AppServiceProvider::class,
    App\Providers\AuthServiceProvider::class,
    //
    App\Providers\BroadcastServiceProvider::class,
    App\Providers\EventServiceProvider::class,
    App\Providers\RouteServiceProvider::class,

    ],

    /*
    |-----
    |-----
    | Class Aliases
    |-----
    |
    | This array of class aliases
    will be registered when this
    application
    | is started. However, feel free
    to register as many as you wish
    as
    | the aliases are "lazy" loaded
    so they don't hinder
    performance.
    |
    */

```

```

        'aliases' =>
Facade::defaultAliases()-
>merge([
    // 'ExampleClass' =>
App\Example\ExampleClass::cl
ass,
])->toArray(),
];

<?php

return [

    /*
    |-----
    | Authentication Defaults
    |-----
    |

    | This option controls the
    default authentication "guard"
    and password
    | reset options for your
    application. You may change
    these defaults
    | as required, but they're a
    perfect start for most
    applications.
    |
    */

    'defaults' => [
        'guard' => 'web',
        'passwords' => 'users',
    ],

    /*
    |-----
    | Authentication Guards
    |-----
    |

    | Next, you may define every
    authentication guard for your
    application.

```

| Of course, a great default configuration has been defined for you

| here which uses session storage and the Eloquent user provider.

| All authentication drivers have a user provider. This defines how the

| users are actually retrieved out of your database or other storage

| mechanisms used by this application to persist your user's data.

| Supported: "session"

*/

'guards' => [
 'web' => [
 'driver' => 'session',
 'provider' => 'users',
],
],

/*
|-----
| User Providers
|-----
|

| All authentication drivers have a user provider. This defines how the

| users are actually retrieved out of your database or other storage

| mechanisms used by this application to persist your user's data.

| If you have multiple user tables or models you may configure multiple

| sources which represent each model / table. These sources may then

| be assigned to any extra authentication guards you have defined.

| Supported: "database", "eloquent"

*/

'providers' => [
 'users' => [
 'driver' => 'eloquent',
 'model' =>
App\Models\User::class,
],

 // 'users' => [
 // 'driver' => 'database',
 // 'table' => 'users',
 //],

 /*
 |-----
 | Resetting Passwords
 |-----
 |

 | You may specify multiple password reset configurations if you have more
 | than one user table or model in the application and you want to have
 | separate password reset settings based on the specific user types.
 |
 | The expire time is the number of minutes that each reset token will be
 | considered valid. This security feature keeps tokens short-lived so

| they have less time to be guessed. You may change this as needed.

| The throttle setting is the number of seconds a user must wait before

| generating more password reset tokens. This prevents the user from

| quickly generating a very large amount of password reset tokens.

|
*/

```
'passwords' => [
    'users' => [
        'provider' => 'users',
        'table' =>
'password_reset_tokens',
        'expire' => 60,
        'throttle' => 60,
    ],
],
```

```
/*
|-----
| Password Confirmation
Timeout
|-----
```

| Here you may define the amount of seconds before a password confirmation

| times out and the user is prompted to re-enter their password via the

| confirmation screen. By default, the timeout lasts for three hours.

|
*/

```
'password_timeout' => 10800,
```

```
];
```

```
<?php
```

```
return [
```

```
    /*
    |-----
```

| Default Broadcaster

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

| This option controls the default broadcaster that will be used by the

| framework when an event needs to be broadcast. You may set this to

| any of the connections defined in the "connections" array below.

```
    | Supported: "pusher", "ably",
'redis', "log", "null"
    */
```

```
        'default' =>
env('BROADCAST_DRIVER',
'null'),
```

```
    /*
    |-----
```

| Broadcast Connections

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

| Here you may define all of the broadcast connections that will be used

| to broadcast events to other systems or over websockets. Samples of

| each available type of connection are provided inside this array.

|
*/

```
'connections' => [
```

```
    'pusher' => [
        'driver' => 'pusher',
        'key' =>
env('PUSHER_APP_KEY'),
        'secret' =>
env('PUSHER_APP_SECRET')
    ,
        'app_id' =>
env('PUSHER_APP_ID'),
        'options' => [
            'host' =>
env('PUSHER_HOST') ?: 'api-
'.env('PUSHER_APP_CLUSTER', 'mt1').'.pusher.com',
            'port' =>
env('PUSHER_PORT', 443),
            'scheme' =>
env('PUSHER_SCHEME', 'https'),
            'encrypted' => true,
            'useTLS' =>
env('PUSHER_SCHEME', 'https') === 'https',
        ],
        'client_options' => [
            // Guzzle client options:
https://docs.guzzlephp.org/en/stable/request-options.html
        ],
    ],
    'ably' => [
        'driver' => 'ably',
        'key' =>
env('ABLY_KEY'),
    ],
    'redis' => [
        'driver' => 'redis',
        'connection' => 'default',
    ]
]
```