

Khusan Davlatov

Dr. Sarvar Abdullaev

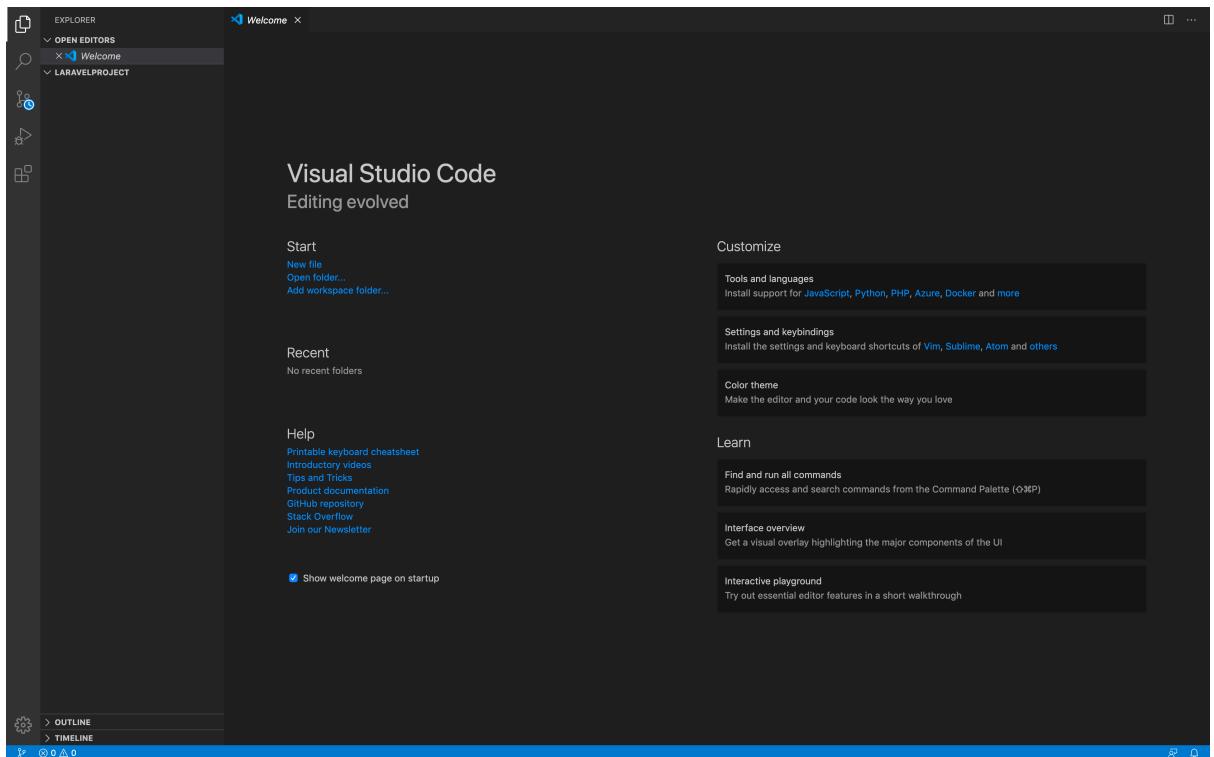
Web Programming

10.05.20

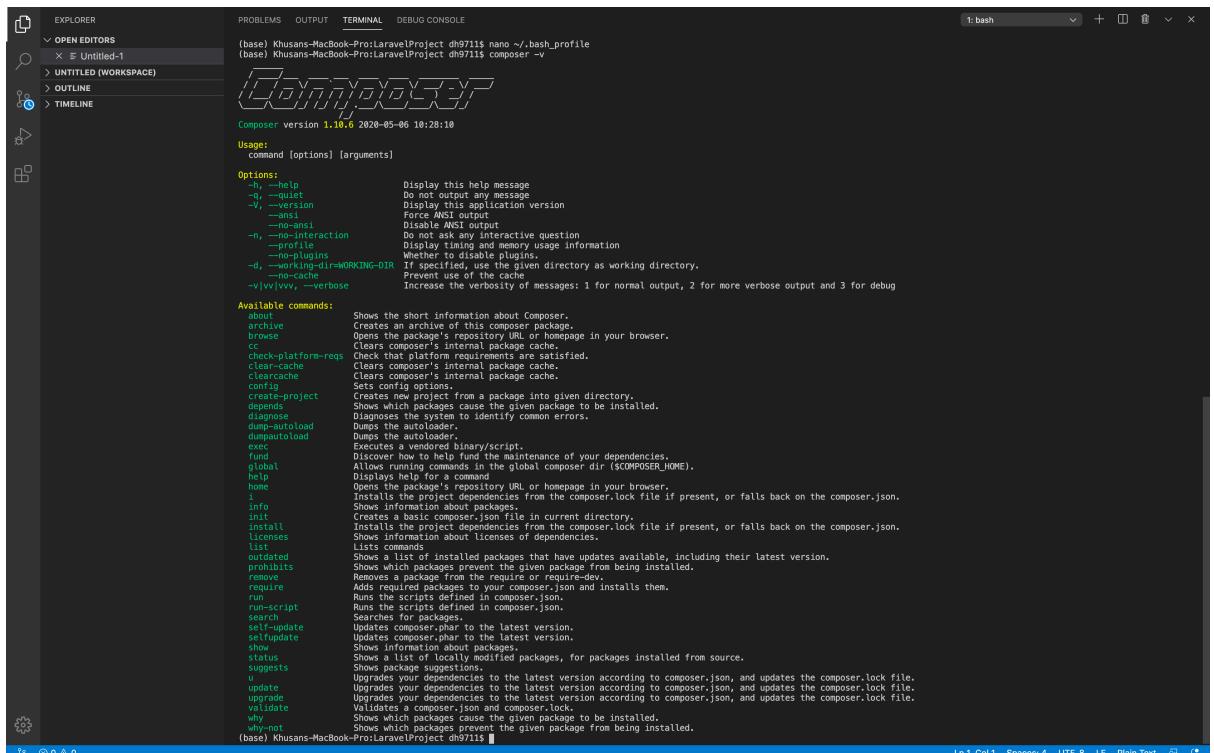


Web Programming Project Report

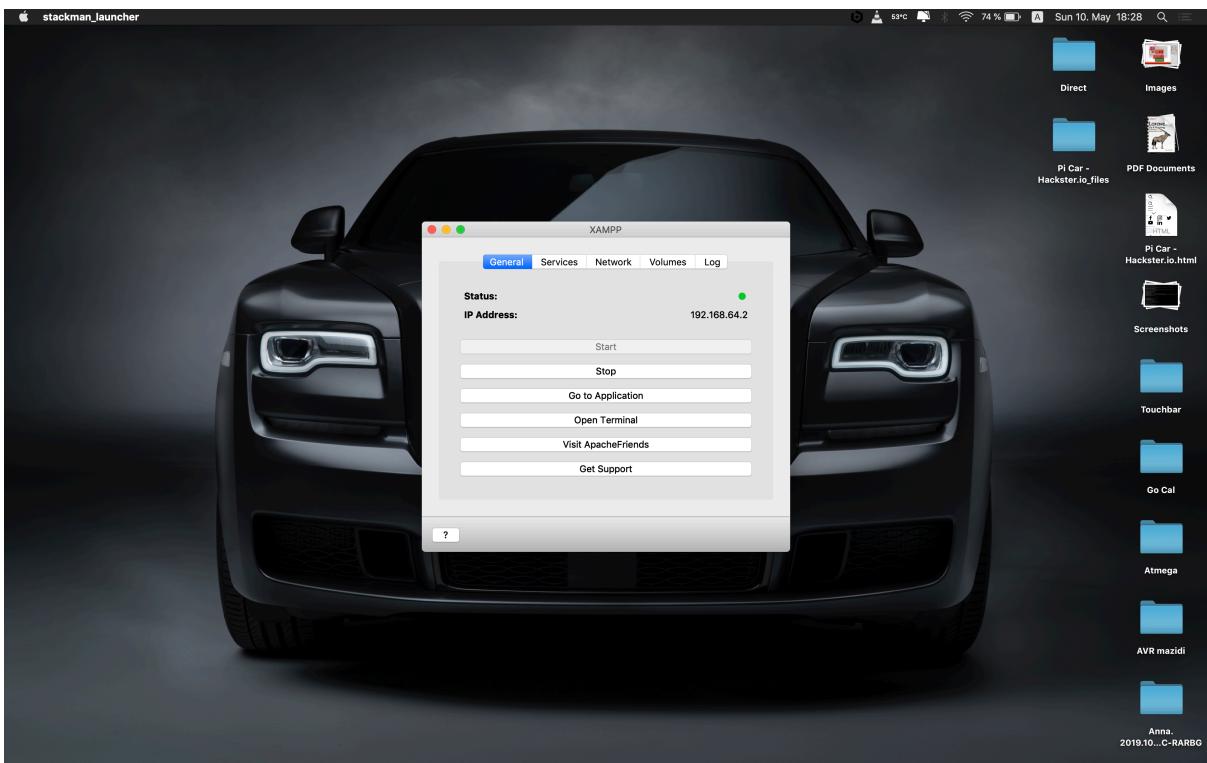
The very concept of this project is to show student's ability to program a fully developed Web application a website on a running server. The idea behind this project is making a fully functioning online website. In this case, I chose a blog post architecture website. Where people can leave their blog. And I intend to use Laravel framework, including Eloquent.



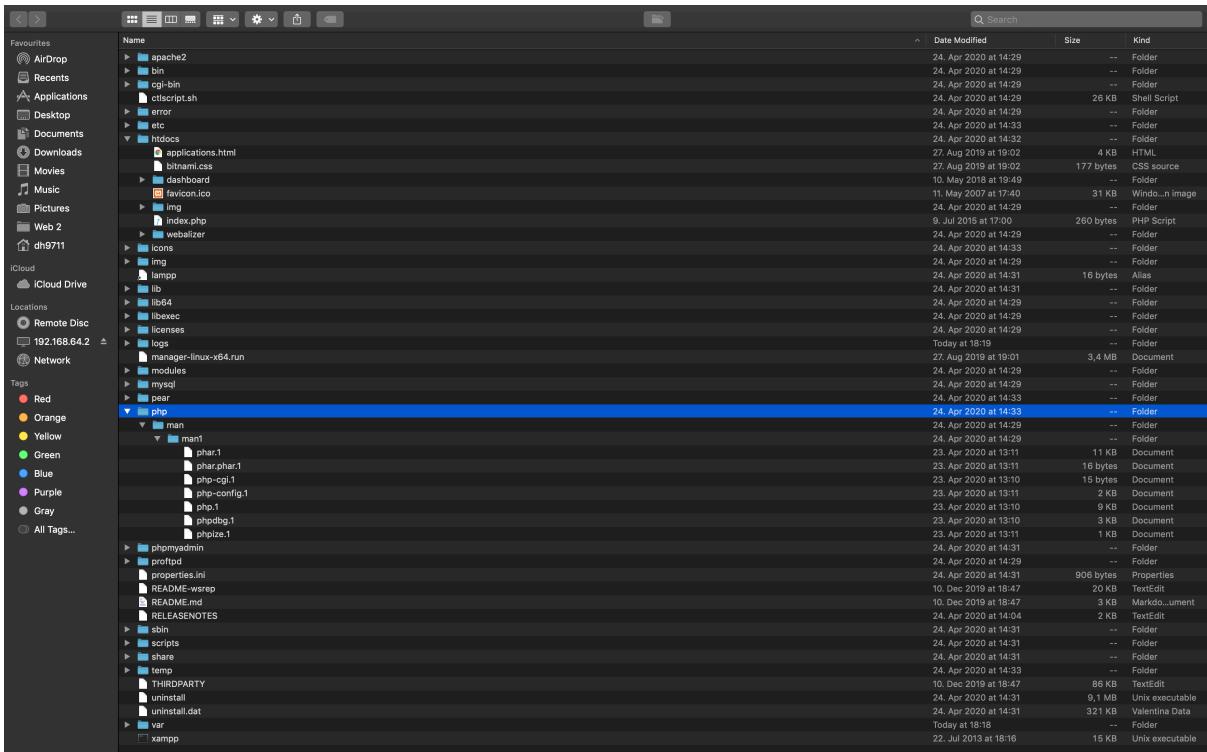
Environment preparation. Using Visual Studio Code.



Using Composer, installing Composer in terminal.



Using XAMPP



XAMPP as localhost. Empty yet!

Creating my web project online.

```

composer -v Untitled-1
 1  composer -v
 2

monolog/monolog suggests installing doctrine/doctrine-logger-bridge (Allow sending log messages to a Doctrine database)
monolog/monolog suggests installing swiftmailer/swiftmailer (Allows sending log messages to an Elasticsearch server)
monolog/monolog suggests installing phpsaml/phpsaml (Allow sending log messages to an AMQP server (1.0+ required))
monolog/monolog suggests installing ext-mongo (Allow sending log messages to a MongoDB server)
monolog/monolog suggests installing aws/aws-sdk-php (Allow sending log messages to AWS services like Amazon S3 via PHP Driver)
monolog/monolog suggests installing rollbar/rollbar (Allow sending log messages to Rollbar)
monolog/monolog suggests installing league/uri (Allow sending log messages to Google Chrome)
league/flysystem suggests installing league/flysystem-event (Allows you to use EventableFilesystem)
league/flysystem suggests installing league/flysystem-rackspace (Allows you to use Rackspace Cloud Files)
league/flysystem suggests installing league/flysystem-azure (Allows you to use Windows Azure Blob storage)
league/flysystem suggests installing league/flysystem-s3 (Allows you to use S3 storage with AWS SDK v2)
league/flysystem suggests installing league/flysystem-aws-s3-v3 (Allows you to use S3 storage with AWS SDK v3)
league/flysystem suggests installing spatie/laravel-droushelper (Allows you to use Drush style commands)
league/flysystem suggests installing league/flysystem-ropbox-v2 (Allows you to use BoxDropbox storage for PHP 5 applications)
league/flysystem suggests installing league/flysystem-cached-adapter (Flysystem adapter decorator for metadata caching)
league/flysystem suggests installing league/flysystem-sftp (Allows you to use SFTP server storage via phpsclib)
league/flysystem suggests installing league/flysystem-s3 (Allows you to use S3 storage with AWS SDK v2)
laravel/framework suggests installing ext-pcntl (Required to use all features of the queue worker)
laravel/framework suggests installing aws/aws-sdk-php (Required to use the SQS queue driver and SES mail driver (^3.0).)
laravel/framework suggests installing doctrine/doctrine-lexer (Required to use the XML and JSON documental drivers (^2.6).)
laravel/framework suggests installing league/flysystem-aws-s3 (Required to use the Amazon S3 and AWS Lambda (^6.0).)
laravel/framework suggests installing league/flysystem-s3 (Required to use the Flysystem S3 driver (^1.0).)
laravel/framework suggests installing league/flysystem-cached-adapter (Required to use the Flysystem cache (^1.0).)
laravel/framework suggests installing league/flysystem-sftp (Required to use the Flysystem SFTP driver (^1.0).)
laravel/framework suggests installing league/flysystem-mysql (Required to use the MySQL Rackspace driver (^4.2).)
laravel/framework suggests installing monostack/math (Required to use ordered UUIDs (^1.1).)
laravel/framework suggests installing nexo/client (Required to use the Nexo transport (^1.0).)
laravel/framework suggests installing league/flysystem-redis (Required to use the Redis queue driver (^4.0).)
laravel/framework suggests installing predis/predis (Required to use the Redis cache and queue drivers (^1.0).)
laravel/framework suggests installing pusher/pusher-php-server (Required to use the Pusher broadcast driver (^3.0).)
laravel/framework suggests installing symfony/monolog-handler (Required to use the Monolog Handler tools (^4.2).)
laravel/framework suggests installing symfony/http-message-bridge (Required to use PSR-7 bridging features (^1.1).)
laravel/framework suggests installing wildbit/swiftmailer-postmark (Required to use Postmark mail driver (^3.0).)
psy/psysh suggests installing ext-pcntl (Enabling the PCNTL extension makes Psysh a lot happier :))
psy/psysh suggests installing libxml2 (Psysh needs libxml2 to work)
psy/psysh suggests installing hoa/console (A pure PHP readline implementation. You'll want this if your PHP install doesn't already support readline or libedit.)
seth/doctrine-coverage suggests installing doctrine-coverage (^1.0.0)
seth/doctrine-coverage suggests installing psalm (^1.0.0)
phunit/phunit suggests installing ext-debug (^2.6.0)
phunit/phunit suggests installing phunit/php-invoker (^2.0)
phunit/phunit suggests installing ext-dl (^1.0.0)
psalm/psalm suggests installing ext-dl (^1.0.0)
composer/composer suggests installing composer/composer (^1.0.0)
composer/composer suggests installing composer/composer-highlighter (^1.0.0)
Package jakub-onderka/php-console-highlighter is abandoned, you should avoid using it. Use php-parallel-lint/php-console-color instead.
Package lock file
Warning: lock file was not generated successfully.
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
@php artisan package:discover --ansi
Discovered Package: beyondcode/laravel-dump-server
Discovered Package: beyondcode/laravel-monolog-bridge
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: numerus/numerus
Discovered Package: nunoester/laravel-validation
Validation has been performed successfully.
22 packages you are using are looking for funding!
Use the `composer fund` command to find out more!
> @php artisan key:generate
Application key set successfully.
(base) Khusan-MacBook-Pro:htdocs dh9711$
```

Ln 1, Col 1 (11 selected) Spaces: 4 UTF-8 LF Plain Text

```

Untitled-1
 1

Available commands:
  about           Show the short information about Composer.
  archive         Creates an archive of this composer package.
  browse          Opens the package's repository URL or homepage in your browser.
  clear-cache    Clears composer's internal package cache.
  check-platform-reqs Checks if the system meets requirements specified.
  clear-cache    Clears composer's internal package cache.
  clear-cache    Clears composer's internal package cache.
  config          Sets configuration options.
  create-project Creates a project from a package into given directory.
  depends         Shows which packages cause the given package to be installed.
  diagnose        Diagnoses the system to identify common errors.
  dumpautoload   Dumps the autoloader.
  dumpautoload   Dumps the autoloader.
  exec            Executes a vendored binary/script.
  funds           Discovers new to help fund the maintenance of your dependencies.
  global          Allows running commands in the global composer dir ($COMPOSER_HOME).
  help            Displays help for a command.
  home            Opens the package's repository URL or homepage in your browser.
  info             Shows information about packages.
  init             Creates a basic composer.json file in current directory.
  install          Installs the project dependencies from the composer.lock file if present, or falls back on the composer.json.
  licenses        Shows information about licenses of dependencies.
  list             Lists commands.
  outdated        Shows a list of installed packages that have updates available, including their latest version.
  remove          Removes a package from the require or require-dev section.
  require         Adds required packages to your composer.json and installs them.
  run-script      Runs the script defined in composer.json.
  run-script      Runs user scripts defined in composer.json.
  search          Searches for packages.
  self-update     Updates composer.phar to the latest version.
  set              Updates composer.json to the latest version.
  show             Shows information about packages.
  status           Shows a list of locally modified packages, for packages installed from source.
  suggest         Shows package suggestions.
  update          Upgrades your dependencies to the latest version according to composer.json, and updates the composer.lock file.
  upgrade         Upgrades your dependencies to the latest version according to composer.json, and updates the composer.lock file.
  validate        Validates the composer.json file.
  why             Shows which packages cause the given package to be installed.
  why-not        Shows which packages prevent the given package from being installed.
(base) Khusan-MacBook-Pro:LaravelProject dh9711$ composer create-project laravel/laravel LaravelProject
Creating a "laravel/laravel" project at ".\LaravelProject"
Installing laravel/laravel (v5.8.33)
  - Installation successful in 0.835s: Downloading (100%)
Created project in /Users/dh9711music/LaravelProject/LaravelProject
@php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading Composer repositories with package information
Updating dependencies (including require-dev)

```

Ln 1, Col 1 (11 selected) Spaces: 4 UTF-8 LF Plain Text

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure for a Laravel application named "healthapp". The "views" folder contains "welcome.blade.php".
- EDITOR:** Displays the content of "welcome.blade.php". The code defines a full-height flex container with a centered title.
- TERMINAL:** Shows the command line output of "composer -v" and "php artisan key:generate --ansi". It also shows the creation of a ".env.example" file and the generation of a "PagesController.php" controller.
- STATUS BAR:** Shows "Ln 1, Col 1" and "Spaces: 4, UTF-8, LF, PHP".

```

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Laravel</title>
        <!-- Fonts -->
        <link href="https://fonts.googleapis.com/css?family=Nunito:200,600" rel="stylesheet">
        <!-- Styles -->
        <style>
            html, body {
                background-color: #fff;
                color: #636b6f;
                font-family: 'Nunito', sans-serif;
                font-weight: 200;
                height: 100vh;
                margin: 0;
            }
            .full-height {
                height: 100vh;
            }
            .flex-center {
                align-items: center;
                display: flex;
                justify-content: center;
            }
            .position-ref {
                position: relative;
            }
        </style>
    </head>
    <body class="flex-center position-ref full-height">
        <h1>Laravel</h1>
    </body>
</html>

```

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure for a Laravel application named "healthapp". The "Http\Controllers" folder contains "PagesController.php".
- EDITOR:** Displays the content of "PagesController.php". It defines a controller extending the Illuminate\Http\Controller class.
- TERMINAL:** Shows the command line output of "composer -v", "php artisan key:generate --ansi", and the creation of a "PagesController.php" controller using the "make:controller" command.
- STATUS BAR:** Shows "Ln 1, Col 1" and "Spaces: 4, UTF-8, LF, PHP".

```

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class PagesController extends Controller
{
}

```

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure under "UNTITLED (WORKSPACE)". Key files visible include `bootstrap`, `cache`, `config`, `database`, `public` (containing `css`, `js`, `index.php`, `robots.txt`, `web.config`), `resources` (containing `js`, `lang`, `sass`), and `views` (containing `welcome.blade.php`).
- Editor View:** Displays the `web.php` file content:

```

1 <?php
2
3 /*
4 | Web Routes
5 |
6 |
7 |
8 | Here is where you can register web routes for your application. These
9 | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13
14 Route::get('/', function () {
15     return view('Hello World!');
16 });
    
```
- Terminal View:** Shows the command `composer make:controller PagesController` being run in the terminal.
- Status Bar:** Shows "Ln 17, Col 1" and "Spaces: 4" along with other standard status bar icons.

Creating controller

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure under "UNTITLED (WORKSPACE)". Key files visible include `bootstrap`, `cache`, `config`, `database`, `public` (containing `css`, `js`, `index.php`, `robots.txt`, `web.config`), `resources` (containing `js`, `lang`, `sass`), and `views` (containing `welcome.blade.php`).
- Editor View:** Displays the `.env` file content:

```

1 APP_NAME=HealthApp
2 APP_ENV=local
3 APP_KEY=base64:jDIKDeEMLDVbxMqG6TT2IXxa6ZJlj6SHullTDECLw=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=127.0.0.1
11 DB_PORT=3306
12 DB_DATABASE=laravel
13 DB_USERNAME=root
14 DB_PASSWORD=
15
16 BROADCAST_DRIVER=log
17 CACHE_DRIVER=file
18 QUEUE_CONNECTION=sync
19 SESSION_DRIVER=file
20 SESSION_LIFETIME=120
21
22 REDIS_HOST=127.0.0.1
23 REDIS_PASSWORD=null
24 REDIS_PORT=6379
25
26 MAIL_DRIVER=smtp
27 MAIL_HOST=smtp.mailtrap.io
28 MAIL_PORT=525
    
```
- Terminal View:** Shows the command `php artisan make:controller PagesController` being run in the terminal, followed by the response "Controller created successfully".
- Status Bar:** Shows "Ln 17, Col 21" and "Spaces: 4" along with other standard status bar icons.

Installing npm
\$ npm run dev
\$ npm run watch

The screenshot shows a code editor interface with a dark theme. On the left is the file explorer, displaying a project structure for a Laravel application. The main area shows the content of the `.env.example` file. The file contains environment variables for a MySQL database, Redis, and Mailtrap. Below the file content is a terminal window showing the output of a compilation process, which completed successfully in 6845ms. The bottom right corner shows status information like 'Ln 1, Col 1' and file encoding.

```

APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost
LOG_CHANNEL=stack
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379
MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525

```

I wanted to make a full report of everything about this project. Apparently, I do not seem to manage to finish the report fully. But mostly, I really loved this framework. I do remember my Dr. Sarvar Abduallev mentioning that this is a cool framework. And I bet he is right. It does have very cool features, and easy to use. Meaningful.