

اول مشروع لارافيل

اترك تعليقاً, [Blog, Laravel](#) / [مشاريع لارافيل](#)

قبل البدء في عمل اول مشروع لارافيل `Laravel project one` ، من المهم تثبيت بعض البرامج والأدوات الضرورية. هذه بعض البرامج التي يُفضل تثبيتها:

1. PHP :

تأكد من تثبيت PHP على جهازك وتكوينه بالطريقة المناسبة. يفضل استخدام إصدار PHP الموصى به من قبل [Laravel](#)

او يمكنك استبداله ببرنامج [xampp](#) وسيكون غني عن ادوات كثيرة انت في حاجة اليها و يمكنك الرجوع [للدروس](#) لمعرفة المزيد عنه

2. Composer :

هو أداة لإدارة الاعتماديات في `PHP. Laravel` يعتمد بشكل كبير على `Composer` لإدارة تبعياته. يمكنك تثبيت [Composer](#) من الموقع الرسمي.

3. قاعدة بيانات:

يمكنك استخدام `MySQL` أو `PostgreSQL` أو `SQLite` أو أي قاعدة بيانات تفضلها. تأكد من تثبيت وتكوين قاعدة البيانات التي تناسب مشروعك.

او يمكنك استبداله ببرنامج [xampp](#) وسيكون غني عن ادوات كثيرة انت في حاجة اليها

4. Node.js (و npm للمشاريع التي تحتوي على الواجهة الأمامية):

إذا كنت تعمل على تطوير جزء الواجهة الأمامية باستخدام `Vue.js` أو `React` أو أي إطار عمل آخر، فسيكون من الضروري تثبيت `Node.js` و `npm` لإدارة حزم `JavaScript`.

5. visual studio code :

فيجوال ستوديو كود ([Visual Studio Code](#)) هو برنامج تحرير النصوص بيئة تطوير مفتوحة المصدر وخفيفة الوزن تم تطويرها بواسطة مايكروسوفت. تعتبر أحد أشهر وأكثر الأدوات شعبية بين المطورين لتطوير البرمجيات وتحرير النصوص بفضل ميزاتها القوية والمرونة التي توفرها.

6. Git :

يفيد Git في تتبع التغييرات وإدارة إصدارات البرمجيات. يمكنك تثبيت [Git](#) للمساهمة في مشروع Laravel أو لإدارة الكود المصدري الخاص بك.

بعد تثبيت هذه البرامج، ستكون جاهزًا للبدء في إنشاء وتشغيل مشروعك في Laravel بكل يسر وسهولة.

اول مشروع لارافيل

حسنًا، يمكنك استخدام الأمر التالي في الطرفية لإنشاء مشروع لارافيل Laravel جديد:

```
composer create-project --prefer-dist laravel/laravel
```

استبدل اسم-المشروع بالاسم الذي تريده لمشروعك في Laravel. هذا الأمر يقوم بتنزيل أحدث إصدار من Laravel وإعداد مشروع لارافيل جديد في مجلد يحمل اسم المشروع. وفي حالة تنزيل إصدار أقدم مثل laravel 9 استخدم الأمر التالي

```
composer create-project --prefer-dist laravel/laravel project-name "9.*"
```

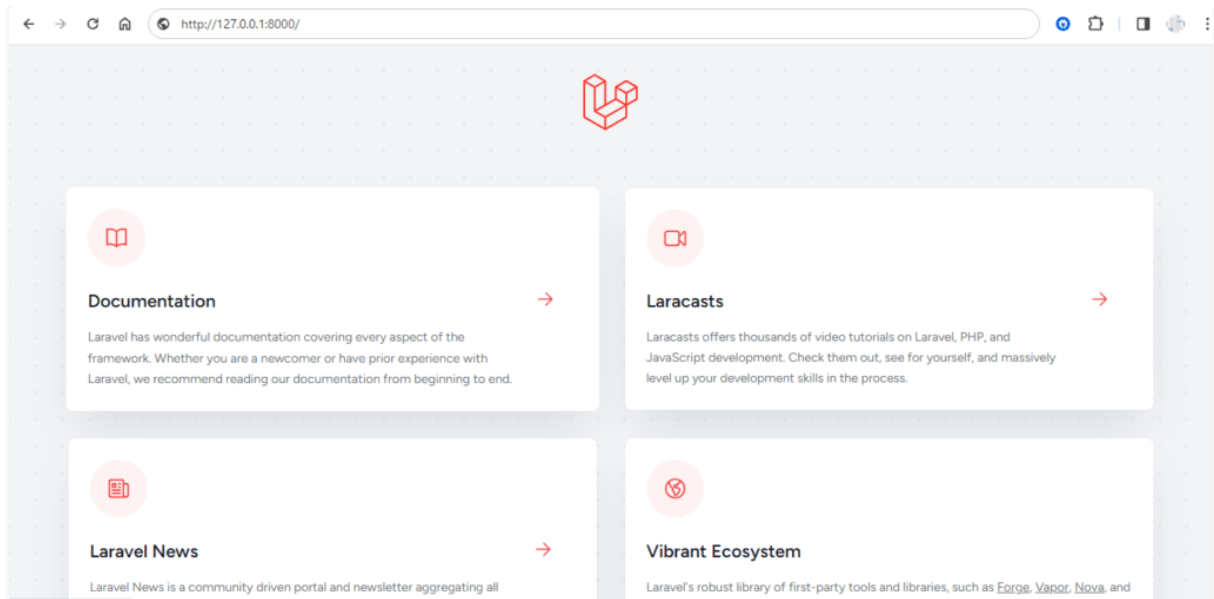
استبدل **project-name** بالاسم الذي تريده لمشروعك في Laravel 8 بعد اكتمال التنصيب، انتقل إلى مجلد المشروع باستخدام:

```
cd اسم-المشروع
```

يمكنك الآن البدء في العمل على مشروع Laravel الخاص بك بتشغيل خادم التطوير باستخدام الأمر التالي:

```
php artisan serve
```

سيقوم هذا الأمر بتشغيل خادم التطوير، ويمكنك الوصول إلى تطبيق Laravel الخاص بك عبر <http://localhost:8000> باستخدام google chrome.



```
composer require laravel/ui
```

لارافيل UI

لارافيل UI هو حزمة تساعد في تنشيط واجهة تطبيق لارافيل الأمامية باستخدام عروض وموارد مسبقة لإطارات العمل الأمامية الشهيرة مثل Bootstrap و Vue.js و React. يبسط هذا الأمر عملية إعداد مكونات الواجهة الأمامية ويوفر نقطة انطلاق لبناء واجهة المستخدم لتطبيق الويب الخاص بك.

سابقًا كانت حزمة Laravel UI مدمجة في Laravel بدءًا من الإصدار 6.x، ولكن اعتبارًا من Laravel 7.x، تم استخراجها إلى حزمة منفصلة. يمكنك استخدامها عن طريق تثبيتها باستخدام Composer:

```
composer require laravel/ui
```

توفر هذه الحزمة الشيفرة المصدرية الأساسية والموارد اللازمة للبدء بسرعة. بعد إنشاء الهيكل، يمكنك تخصيصه وبناءه لإنشاء الهيكل والتصميم الأمامي المحدد المطلوب لتطبيقك.

```
php artisan ui bootstrap --auth
```

bootstrap and auth

أمر `php artisan ui bootstrap --auth` يستخدم لتثبيت حزمة Bootstrap وإنشاء هيكل مبدئي لنظام المصادقة (Authentication) في Laravel. هذا الأمر يقوم بإعداد الجوانب الأساسية لنظام المصادقة مثل تسجيل الدخول (Login) وتسجيل المستخدمين (Registration)، وإعادة تعيين كلمة المرور (Password Reset)، وتأكد البريد الإلكتروني (Email Verification)، باستخدام واجهة Bootstrap كمظهر افتراضي.

بمجرد تشغيل هذا الأمر، ستقوم بتوليد الملفات والمجلدات اللازمة لنظام المصادقة مع Bootstrap كواجهة للمستخدم. يمكنك بعد ذلك تخصيص وتعديل هذه الملفات لتناسب مع احتياجات مشروعك وتصميم الواجهة الخاصة بك.

```
npm install
```

npm install

أمر `npm install` يستخدم لتنصيب جميع الوحدات النمطية (modules) الموجودة في ملف `package.json` في مشروع Node.js. هذا الأمر يقوم بتنزيل وتنصيب جميع الوحدات النمطية المذكورة كمتبعيات (dependencies) في مشروعك.

عندما تقوم بتشغيل `npm install` في مجلد مشروعك، ستقوم npm مدير حزم (Node.js) بالبحث عن ملف `package.json`، ومن ثم قراءة الوحدات النمطية المدرجة هناك وتنصيبها في مجلد `node_modules`.

يمكن أن يكون هذا الأمر مهمًا بشكل خاص بعد تنزيل مشروع جديد أو بعد تحديثات كبيرة في المشروع، حيث يتطلب استخدام مكتبات جديدة أو محدثة.

note : يجب تنزيل وتنصيب اخر اصدار من برنامج `node.js` قبل تنفيذ هذا الامر

npm run dev

npm run dev

`npm run dev` هو أمر يُستخدم في تطوير تطبيقات الويب باستخدام npm ويعتبر جزءًا من سكربتس المشروع المُعرّفة في ملف `package.json`.

ويتم استخدامه لتنفيذ سلسلة من الأوامر التي تقوم بعمليات مثل:

– تجميع الملفات الـ CSS والـ JavaScript والملفات الأصلية للمشروع.

– تحويل وتحسين الملفات (مثل تقليص الحجم أو ضغط الصور).

– توحيد وتنظيم الأصول والملفات لجعلها جاهزة للاستخدام في الويب.

على سبيل المثال، في مشاريع Vue.js أو Laravel التي تحتوي على ملف `package.json`، قد تجد سكربت مُعرّف باسم `dev` والذي يُشغّل عمليات تطويرية مثل تجميع الملفات للوضع التطويري.

لا يمكن تحديد بالضبط ماذا يفعل `npm run dev` في مشروع معين دون معرفة المحتوى الدقيق لسكربت الـ `dev` في `package.json` لهذا المشروع.

npm run build

npm run build

`npm run build` هو أمر يستخدم عادةً في سياق تطوير التطبيقات ويُستخدم لبناء (Build) التطبيق للإصدار النهائي أو الإنتاجي. هذا الأمر يقوم بتجميع وتجهيز الملفات والأصول الخاصة بالتطبيق بحيث يمكن نشرها بشكل مناسب واستخدامها في بيئة الإنتاج.

عادةً ما يتم استخدام `npm run build` قبل نشر التطبيق على الخادم الحقيقي أو قبل إطلاق الإصدار النهائي من التطبيق لضمان أنه يعمل بشكل سليم وفعال في بيئة الإنتاج.