

✓ وردپرس، no code است.

✓ لاراول code است.

✓ هرابزاری رو نمیدونستی مشابه دارد یا خیر با کلیدواژه **alternative** سرچ کن:

○ مثلا: **RSS reader alternative feedly**

○ مثلا اگر میخوای بدونی نقشه ای برای راهنمایی جاده مثل اپ ویز وجود داره یا نه سرچ کن: **Weiz alternatives**

✓ کسی که رشته اش کامپیوتره باید همه نرم افزارها رو نصب کنه و توی ذهنش، سناریوی اون برنامه ها رو و حتی دیتابیس اونها رو شبیه سازی کنه

✓ در فیدلی فقط سایتهای مهم برنامه نویسی که **top 10** دارن رو دنبال کن. یا حتی **pinterest**

✓ مهم ترین کار در شروع هر پروژه :

○ راه اندازی محیط دیباگ است

○ آماده کردن سیستم و ماشین **develop** قدم بعدی است

○ وصل کردن به گیت مرحله ی بعدی است چون بتوانی تک تک مراحل کار را تغییر بدهی و تست کنی و اگر مشکل خوردی برگردی یک قدم عقب تر و نه اینکه از اول شروع کنی.

○ ساختن برنچ مرحله ی بعدی است.

✓ جاوا اسکریپت آیونیک، **AngularJS** است. که با جاوا اسکریپت شی گرایی به نام تایپ اسکریپت ترکیب شده.

✓ **هدف اول:** راه اندازی محیط دیباگ برای **PHP** در **vscode**:

○ برای اینکار، ابتدا باید **PHP** را برای استفاده از دیباگ، راه اندازی کنیم و بعدش **vscode** را برای استفاده از این محیط دیباگ

php (که در مرحله قبل راه اندازی کردیم) راه اندازی کنیم. پس:

▪ اگر همه ی برنامه هایی که قبلا گفتیم رو نصب کردید و **vscode** و **xampp** نصب هستش و وردپرس داری. بیا شروع کنیم به درست کردن محیط دیباگ

▪ راه اندازی **php** برای استفاده از دیباگ:

○ سرچ کن: <https://www.codewall.co.uk/debug-php-in-vscode-with-xdebug/>

○ حالا مراحل بیان شده در لینک فوق رو قدم به قدم بریم:

(۱) اول گوگل کنید . همهی راهنماهایی که در اینترنت هستش رو باید بتوانید طی کنید و

همشون در گام اول میگن **php info** بگیر که با توجه به نسخه ی **php** و سیستم عامل ات بهت

بگیم چطوری و با دانلود کدوم فایل **dll** محیط دیباگ رو درست کنید

پس اول برای اینکه **phpinfo** اجرا کنیم یک فایل خالی بسازیم خب ایجاد فایل **index.php** در

root directory زمپ:

○ یه **.txt** فایل ایجاد کن.

○ بازش کن

○ کد موجود در عکس زیر را در آن **paste** کن و ذخیره کن

○ پسوند .txt را به .php تغییر بده

(۲) با یه مرورگر (ترجیحا کروم) بازش کن

(۳) کل محتوای صفحه <http://localhost:8080/info/> را کپی کن.

(۴) برو به <https://xdebug.org/wizard> و آن را پیست کن

و دکمه "Analyse my phpinfo() output" را بزن.

(۵) فایل http://xdebug.org/files/php_xdebug-2.7.2-7.3-vc15-x86_64.dll را دانلود کن.

(۶) فایل دانلود شده را در C:\xampp\php\ext قرار بده.

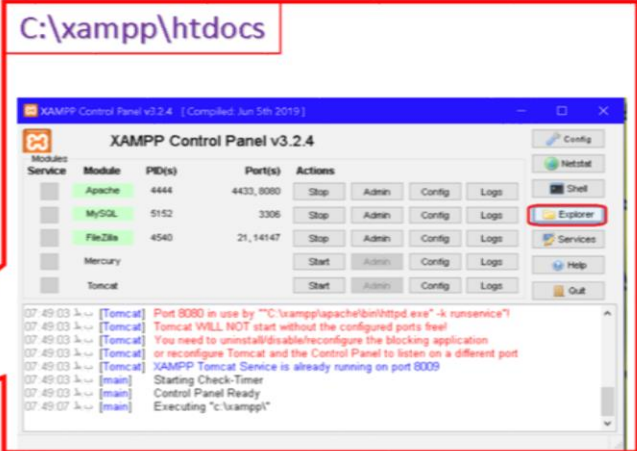
○ ext مخفف extension است.

(۷) وبسرور را restart کن.

Step 1


Create an index.php file in the root directory of your web server, and add the following code.

```
<?php
phpinfo();
?>
```



localhost:8080/info/

PHP Version 7.3.9



System	Windows NT DESKTOP-CN1C376 10.0 build 18362 (Windows 10) AMD64
Build Date	Aug 28 2019 09:21:06
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	cmd.exe /nologo configure.js "--enable-snapshot-build"--enable-debug-pack"--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\w64\instantclient_12_1\sdk_shared"--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\w64\instantclient_12_1\sdk_shared"--enable-object-out-dir=.obj"--enable-com-dotnet=shared"--without-analyzer"--with-pgsql
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731.TS.VC15
PHP Extension Build	API20180731.TS.VC15
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
Zend Signal Handling	disabled

debug PHP (aravel windows v... | Debug PHP in VSCode With X... | Xdebug: Support — Tailored In... | Xdebug: Installation Instruction... | +

https://xdebug.org/wizard

INSTALLATION WIZARD

[home](#) | [updates](#) | [download](#) | [documentation](#) | [contributing](#)

If Xdebug saves you time, please consider [supporting the project](#).

This page helps you finding which file to download, and how to configure PHP to get Xdebug running. Please paste the **full** output of `phpinfo()` (either a copy & paste of the HTML version, the HTML source or `php -i` output) and submit the form to receive tailored download and installation instructions.

```

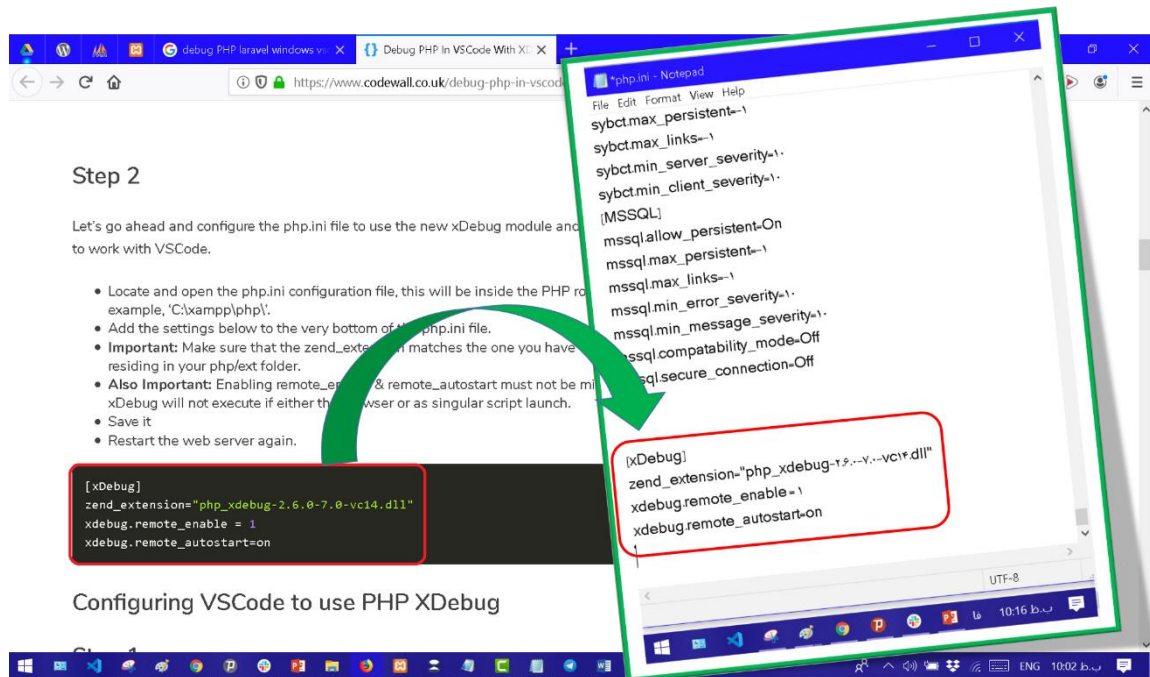
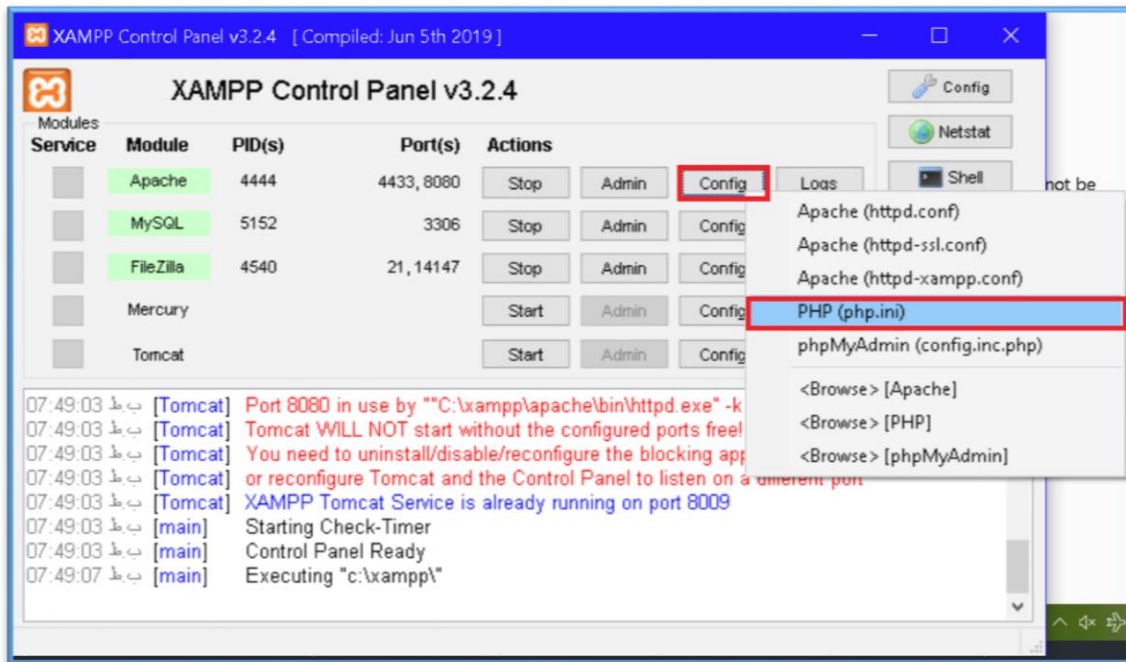
uname Linux ip1000000
SOAP Brad LaFountain, Shane Caraveo, Dmitry Stogov
Sockets Chris Vandomelen, Sterling Hughes, Daniel Beulshausen, Jason Greene
Sodium Frank Denis
SPC Marcus Baecker, Etienne Kneuss
SQLite3.x.driver for PDO Wez Furlong
SQLite3 Scott MacVicar, Ilia Alshanetsky, Brad Dewar
System V Message based IPC Wez Furlong
System V Semaphores Tom May
System V Shared Memory Christian Cartus
tidy John Coggeshall, Ilia Alshanetsky
tokenizer Andrei Zmievski, Johannes Schlueter
WDDX Andrei Zmievski
XML Stig Bakken, Thies C. Arntzen, Sterling Hughes
XMLReader Rob Richards
xmlrpc Dan Libby
XMLWriter Rob Richards, Pierre-Alain Joye
XSL Christian Stocker, Rob Richards
Zip Pierre-Alain Joye, Reini Collet
Zlib Rasmus Lerdorf, Stefan Roehrich, Zeev Suraski, Jade Nicoletti, Michael Wallner
PHP Documentation
Authors Mehdi Achour, Friedhelm Betz, Antony Dovgal, Nuno Lopes, Hannes Magnusson, Philip Olson, Georg Richter, Damien Seguy, Jakub Vrana, Adam Harvey
Editor Peter Cowburn
User Note Maintainers Daniel F. Brown, Thiago Henrique Fojda
Other Contributors Previously active authors, editors and other
  
```

The information that you upload will not be stored. The script will only use a few regular expressions to analyse the output and provide you with instructions. You can see the code [here](#).

[Analyse my phpinfo\(\) output](#)

ابزاری هست که با استفاده از به پارسر، محتوایی که پیست کردی را آنالیز میکنه

- در فایل `C:\xampp\php\php.ini` (که از دکمه Config زمپ هم قابل دسترسی است) طبق عکس های زیر پیش میریم:



The screenshot shows a Windows desktop environment. In the background, a web browser displays the Xdebug website at <https://xdebug.org/wizard>. The page includes a navigation menu (home, updates, download, documentation, contributing, support) and a summary of installed components:

- Xdebug Installed: 2.7.2
- Server API: Apache 2.0 Handler
- Windows: yes - Compiler: MS VC 15 - Architecture: x64
- Zend Server: no
- PHP Version: 7.3.9
- Zend API nr: 320180731
- PHP API nr: 20180731
- Debug Build: no
- Thread Safe Build: yes
- OPcache Loaded: no
- Configuration File Path: C:\WINDOWS
- Configuration File: C:\xampp\php\php.ini
- Extensions directory: C:\xampp\php\ext

Below the summary, there is a section titled "YOU'RE ALREADY RUNNING THE LATEST XDEBUG VERSION" with instructions on how to update Xdebug. Step 3 is highlighted with a red box:

- Download [php_xdebug-2.7.2-7.3-vc15-x86_64.dll](#)
- Move the downloaded file to C:\xampp\php\ext
- Update C:\xampp\php\php.ini and change the line `zend_extension = C:\xampp\php\ext\php_xdebug-2.7.2-7.3-vc15-x86_64.dll`
- Restart the webserver

In the foreground, a Notepad window titled "php.ini - Notepad" is open, displaying the following configuration code:

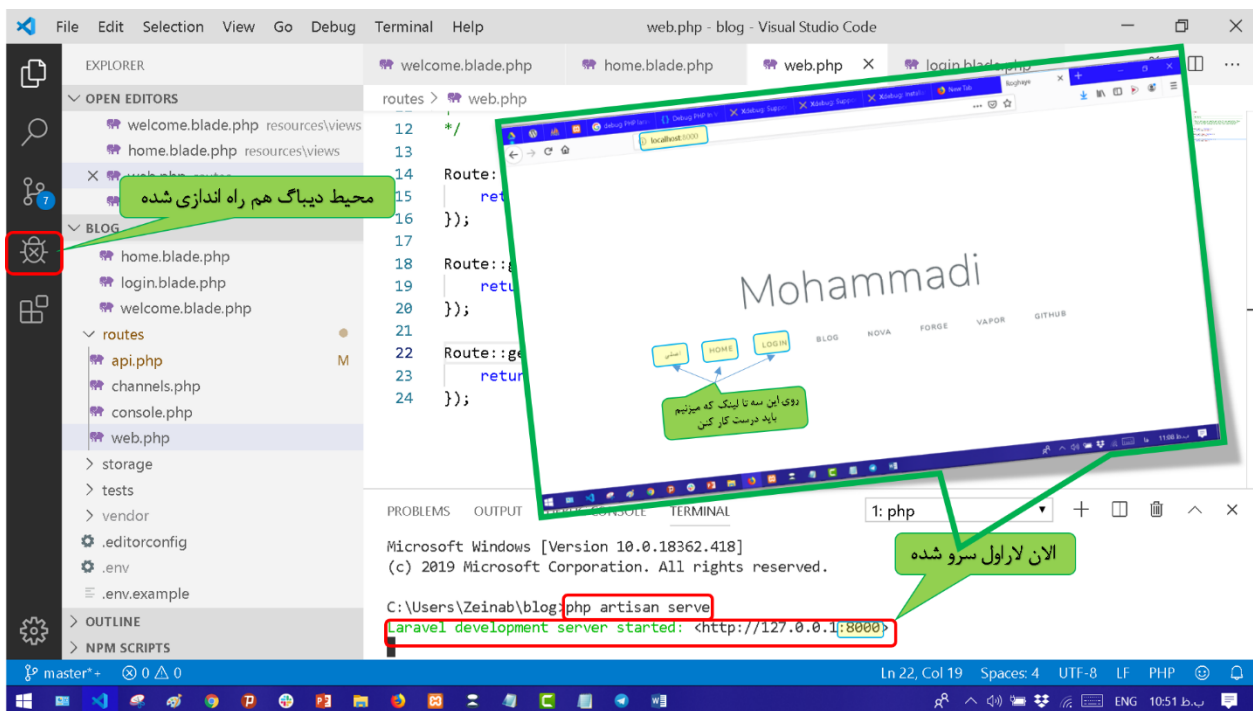
```
zend_extension=C:\xampp\php\ext\php_xdebug-2.7.2-7.3-vc15-x86_64.dll
zend_extension="php_xdebug-2.7.2-7.3-vc15-x86_64.dll"
xdebug.remote_enable=1
xdebug.remote_autostart=on
```

Handwritten Persian annotations are present: "خط سمت چپ" (left line) and "جایگزین خط سمت راست شود" (replace the right line) with an arrow pointing to the second line. Red boxes highlight the second line and the third line.

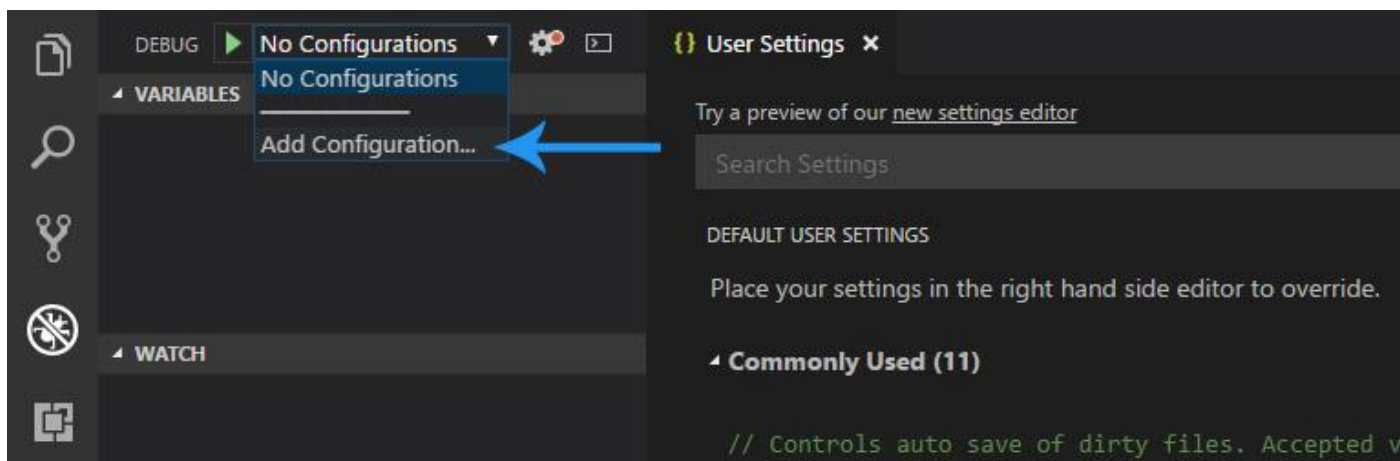
▪ راه اندازی **vscode** برای استفاده از محیط دیباگ **php** (که در مرحله قبل راه اندازی کردیم):

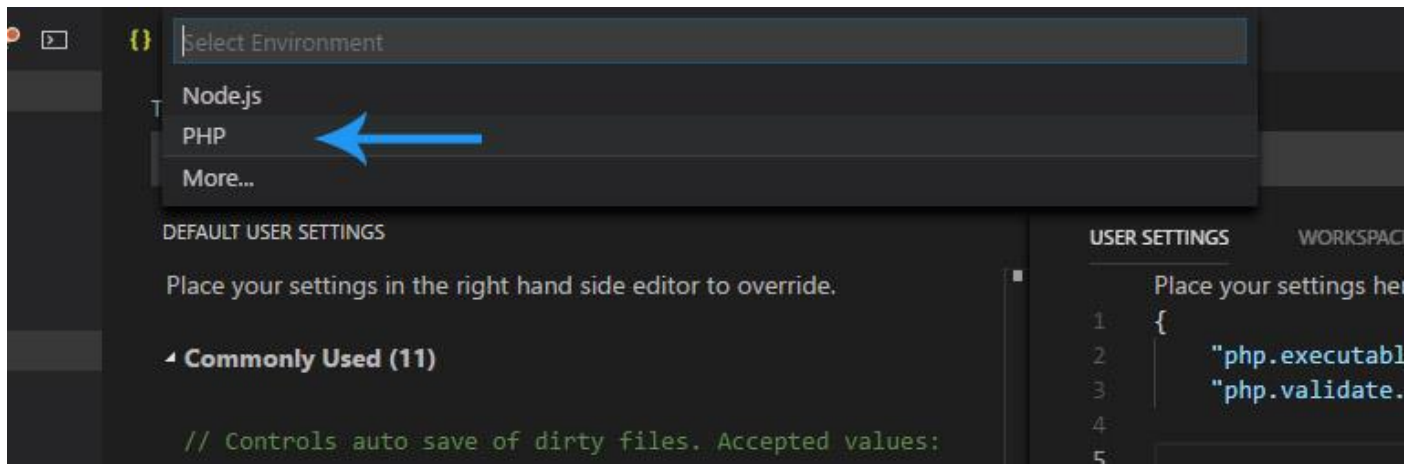
○ پوشه ای که جلسه قبل لاراول را در آن نصب کردی را با **vscode** باز کن (برای من اینه:

(C:\Users\Zeinab\blog

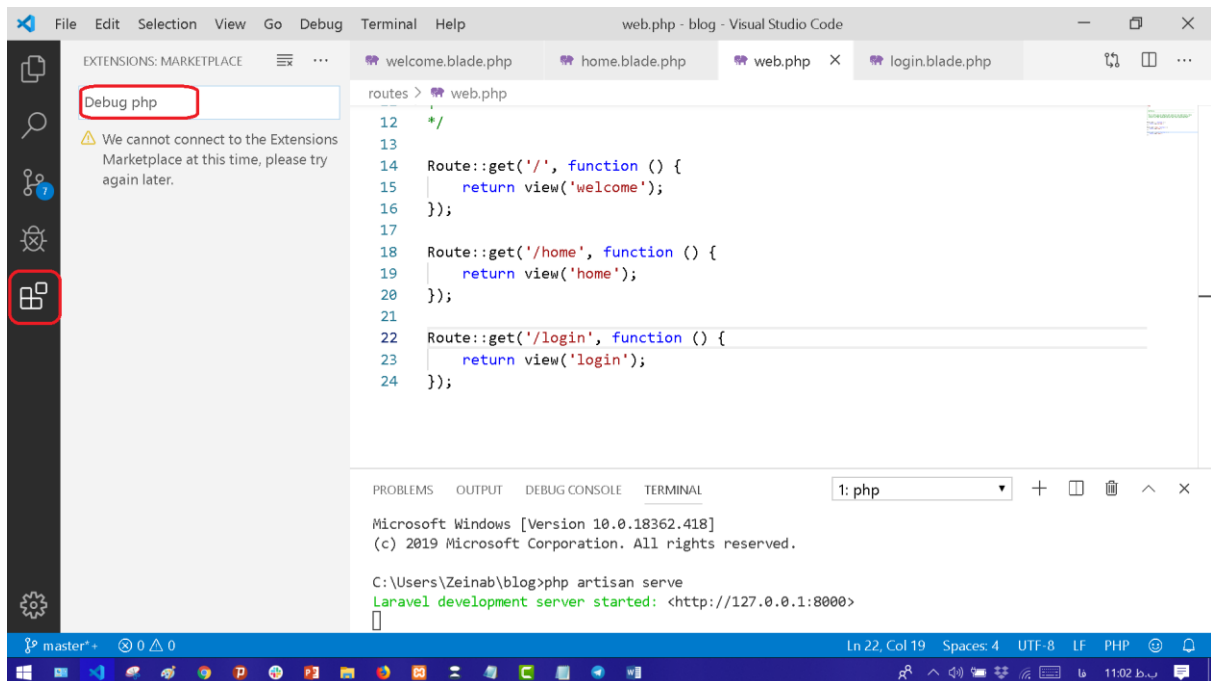


○ برای اینکه دیباگ ران بشه، باید:





اگه در عکس فوق، PHP رو نیاورد، باید از Market Place بری php Debug رو دانلود و نصب کنی (عکس زیر):



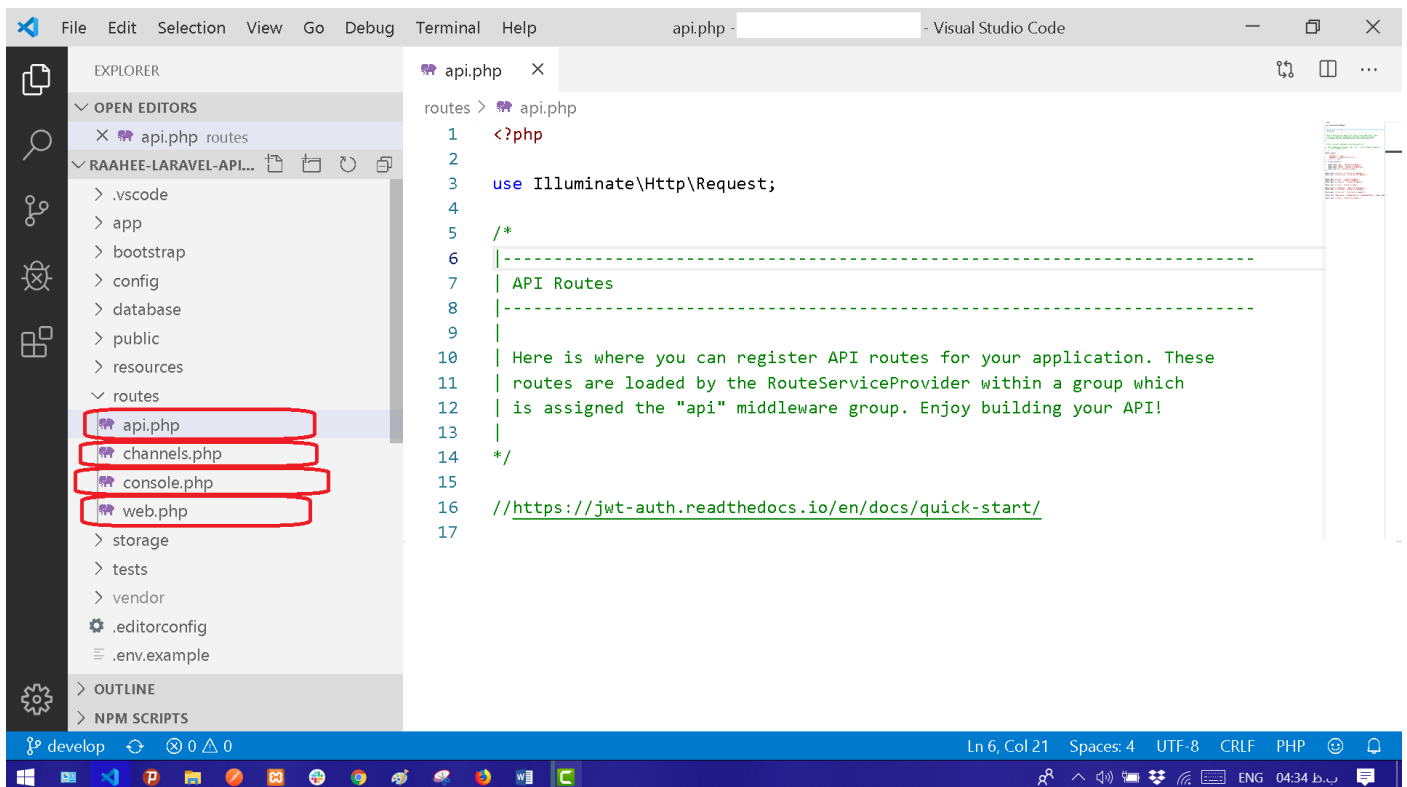
شروع کار با لاراول.

مقدمات:

✓ فایل های روتر کامپیوتر ما:

- Api.php ○
- Channels.php ○
- Console.php ○
- :Web.php ○

✓ روتر، یعنی جایی که اولین خط برنامه من اجرا میشه. یعنی لاراول، اول میره سراغ این فایلها تا ببینه چه آدرسی را براش تعریف کردی.



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a project structure with folders like .vscode, app, bootstrap, config, database, public, resources, routes, storage, tests, vendor, .editorconfig, and .env.example. The 'routes' folder is expanded, and four files are highlighted with red boxes: api.php, channels.php, console.php, and web.php. The main editor window shows the content of api.php, which includes a PHP opening tag, a use statement for Illuminate\Http\Request, a comment block for API Routes, and a URL reference.

```
1 <?php
2
3 use Illuminate\Http\Request;
4
5 /*
6 |-----
7 | API Routes
8 |-----
9 |
10 | Here is where you can register API routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | is assigned the "api" middleware group. Enjoy building your API!
13 |
14 |*/
15
16 //https://jwt-auth.readthedocs.io/en/docs/quick-start/
17
```

✓:Web.php

- هرچی ترافیک با اسم سایت شما میاد، توی این روتر مدیریت میشه.
- در عکس زیر نوشته: هر وقت route گت اجرا شد و اسلش رو زد، برو به ویوی ولکام
- در صفحات وب، get داریم و post. پست ها معمولا در پشت صحنه به سرور پست میشن.
- صفحه web.php وقتی اجرا میشه که من با مرورگر صداش بزنم. معادل همین صفحه html میتونیم یه api داشته باشیم (یه وب سرویس داشته باشیم). شبیه آن چیزی که وردپرس داشت. در وردپرس، وقتی میزدیم: "فلان /api" به کل api های وردپرس دسترسی میداد.

Visual Studio Code interface showing the file explorer on the left with 'web.php' selected under 'resources/views/routes'. The main editor displays the following PHP code:

```
10 | contains the "web" middleware group. Now create something great!
11 |
12 | */
13 |
14 | Route::get('/', function () {
15 |     return view('welcome');
16 | });
```

The terminal at the bottom shows the command `php artisan serve` and the output: `Laravel development server started: <http://127.0.0.1:8000>`

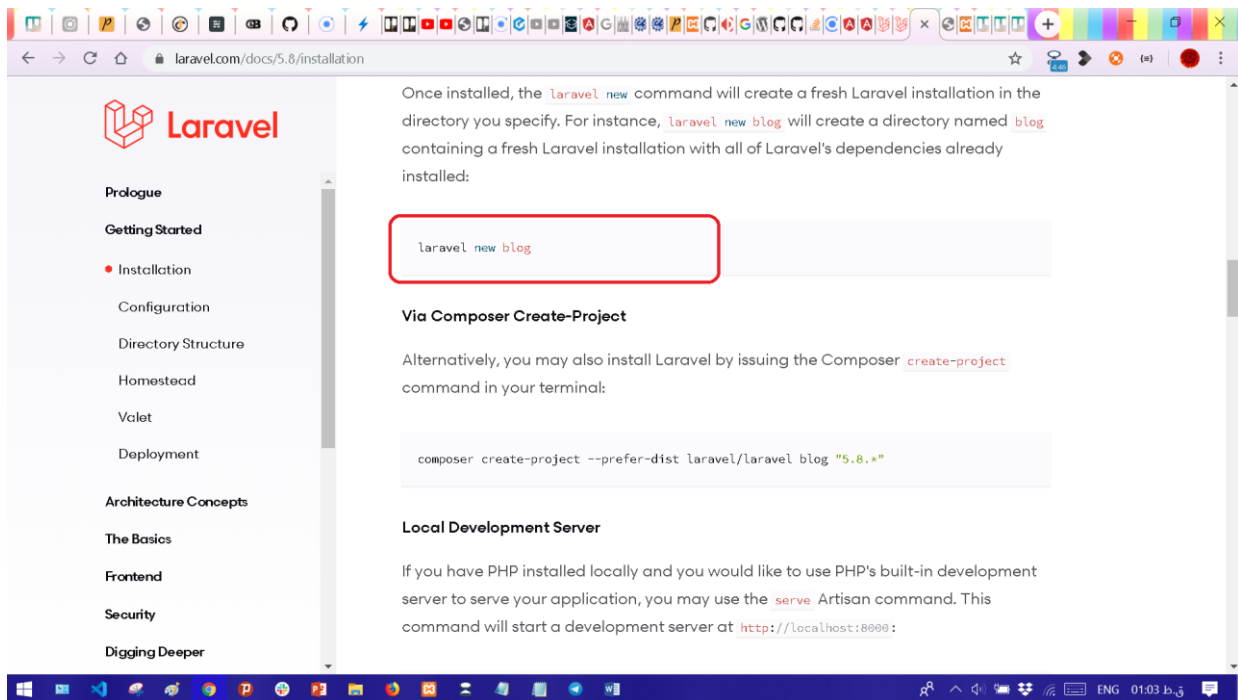
Visual Studio Code interface showing the file explorer on the left with 'welcome.blade.php' selected under 'resources/views'. The main editor displays the following Blade template code:

```
75 | @if (Route::has('register'))
76 |     <a href="{{ route('register') }}">Register</a>
77 | @endif
78 | @endauth
79 | </div>
80 | @endif
81 |
82 | <div class="content">
83 |     <div class="title m-b-md">
84 |         Mohammadi
85 |     </div>
86 |
87 |     <div class="links">
88 |         <a href="{{ url('/') }}">اصلي</a>
89 |         <a href="{{ url('/home') }}">home</a>
90 |         <a href="{{ url('/login') }}">login</a>
91 |         <a href="https://blog.laravel.com">Blog</a>
```

A green callout box points to the word 'Mohammadi' on line 84, containing the text: `جلش نوشتہ Mohammadi`. The terminal at the bottom shows the command `php artisan serve` and the output: `Laravel development server started: <http://127.0.0.1:8000>`

✓ #تمرین ۱:

- لاراوول را در یک پوشه دیگر به نام twitter نصب کنید (میخواهیم توئیتر بسازیم). یعنی در عکس زیر، به جای blog بنویس twitter:



✓ #تمرین ۲ (۵/۲۰/۲۰۲۰):

- یک پروژه جدید آیونیک به نام twitter-ionic ایجاد کنید. میخواهیم عین همین route رو که در لاراوول گفتیم، در آیونیک انجام بدیم:

- چند صفحه در پروژه جدیدتان اضافه کنید (با دستور تولید موجود در

- <https://ionicframework.com/docs/v3/cli/generate/>

- در فایل روت آیونیک به کمک گوگل صفحات رو تعریف کنید

- یعنی همان کاری که در لاراوول امروز انجام دادیم

- اضافه کردن دکمه button به هر صفحه:

- گوگل کن: How to add event when click open another page to ionic

- دستور this.NavController میادش

- اضافه کردن متد click به هر دکمه

- وصل کردن هر تابع به صفحه دیگر طوری که با کلیک روی دکمه به صفحه دیگر برود

- سناریوی هفته آینده:

- سمت لاراوول یک وب سرویس در پوشه api.php که در پوشه route هست اضافه میکنیم

- توئیتهایی که به صورت نمونه و اولیه الکی داریم به کاربر میخواهیم نمایش دهیم

- سمت آیونیک توئیتهای سمپل init اولیه را میخوانیم و نمایش میدهیم

- یک دکمه برای اضافه کردن توئیتر در صفحه اول میگذاریم

- با کلیک روی دکمه به صفحه توئیتهای جدید میرویم

- با تایپ توئیتهای جدید توسط کاربر و دکمه ارسال توئیتهای جدید به سرور ارسال میشود

- برای لاراوول یک متد جدید در api.php ایجاد میکنیم

- برای لاراوول یک متد برای مدیریت توئیتهای جدید ایجاد کرده و توئیتهای جدید را در دیتابیس ذخیره میکنیم