

**واحد اردبیل**

**پویا دوستدار**

**استاد راهنما:**

**خوشنام عبدي قورتولموش**

**موضوع پروژه:**

**اپلودسنتر**

**زمستان۱۳۹۶**



**تقدیم به:**

**هر فردی برای به نمایش درآوردن تصاویر دلخواهش در وبلاگ/وبسایت شخصی خود و یا برای به اشترک گذاشتن دیگر فایلهای مورد علاقه اش از قبیل آهنگ، فیلم، مقالات و ... با دیگر دوستان اینترنتی خود احتیاج به فضایی بر روی اینترنت دارد. در واقع پروژه اپلود سنتر فضای اینترنتی رایگانی را برای شما فراهم میکند تا هر آنچه را که می خواهید در اینترنت با دیگران به اشتراک بگذارید در آن قرار دهید. تنها کافیست وارد سایت شده و فایلهای مورد نظر خود را از کامپیوتر شخصی خود به فضای شخصی خود در سایت قرار دهید.**

**فهرست مطالب**

1. **معرفی و نصب ابزار های برنامه نویسی**

* 1. **نرم افزار phpstorm چیست؟ .....................................................................................................2**

* + 1. ویژگی های کلیدی نرم افزار **PhpStorm..............................................................................2**

* 1. **معرفی نرم افزار lamp و نصب ان .............................................................................................۳**

* + 1. معرفی**...........................................................................................................................................................3**

* + 1. نصب**..............................................................................................................................................................4**

* + - 1. **نصب اپاچی .................................................................................................................................................4**

* + - 1. **نصب MySQL ......................................................................................................................................6**

* + - 1. نصب**php ..................................................................................................................................................6**

* + - 1. **تست php .................................................................................................................................................8**

* 1. **زبان ها و ابزار های استفاده شده در پروژه...........................................................................10**

* + 1. **php چیست؟ ........................................................................................................................................10**

* + - 1. **ویژگی های عمومی زبان php ....................................................................................................10**

* + 1. **MySQL چیست**؟ **...........................................................................................................................…15**

* + - 1. **ویژگی های اصلی MySQL ..........................................................................................................16**

* + - 1. **رقبای MySQL عبارت اند از ...................................................................................................... ۱6**

* + - 1. **مزیت های MySQL ...................................................................................................................... 16**

* + 1. **apache چیست؟ ..............................................................................................................................۱7**

* + - 1. **ویژگی های اصلی apache ......................................................................................................... 17**

1. **طراحی پایگاه داده**
   1. **جداول ………………………………………….....................................................................................………………۱۳**
   2. **جدول downloads ……………………………………….........................................................................14**
   3. **جدول files ..........................................................................................................................................16**
   4. **جدول migrations .......................................................................................................................16**
   5. **جدول password\_resets ....................................................................................................16**
   6. **جدول requests ............................................................................................................................16**
   7. **جدول statistics ...........................................................................................................................16**
   8. **جدول users .....................................................................................................................................16**
2. **Frame work**

* 1. **Frame work چیست؟**………………………………………………………….…………………………..…………...**26**

* 1. **چرا باید از فریم ورک استفاده کنیم؟**……………………………….…………….………………………….……**26**

* 1. **ORM چیست؟ .....................................................................................................................................27**

* 1. **MVC در فریم‌ورک‌ها به چه معناست؟**……………………………………………….…………………………**27**

* 1. **فریم ورک لاراول چیست؟**………..…….………………………………………….………..……………………….......**28**

* 1. **چرا لاراول ؟**…………….……………………………………………….………..………………..……………......................**28**

* 1. **کامپوزر چیست؟.................................................................................................................................29**

* 1. **نصب کامپوزر؟...................................................................................................................................32**

* 1. **نصب و راه اندازی فریم ورک لاراول** ..…….…………….……………….……..………………………………**32**

* 1. **پوشه های لاراول** …………………………..……………………………………….……………..……………………….......**33**

* + 1. **App دایرکتوری..................................................................................................................................34**

1. فایل های پروژه
   1. **مدل ها ....................................................................................................................................................23**
   2. **مایگریشن ها .........................................................................................................................................24**
   3. **Schema ها .............................................................................................................................................25**
   4. **کنترلر ها ................................................................................................................................................25**

**مقدمه:**

کسترش روز افززون شبکه اینترنت و استفاده از ان در اکثر امور زندگی از موفقیت های چشم گیر سالهای اخیر میباشد با توجه به این موضوع باید برای همگام شدن با پیشرفت سریع علوم روز دنیا بخصوص در شبکه و اینترنت و استفاده از ان در راستای تحقق اهدافی چون کاهش هزینه و زمان تلاش کنیم. هدف اصلی این سیستم بر پایی وب سایتی برای اپلود و دانلود برای فعالان این عرصه میباشد تا برای تولید محتوا و اشتراک گزاری فایل بتوانند از فضای مجازی که در اختیارشان به صورت رایگان قرار میگیرد استفاده کنند.

فصل اول

**معرفی و نصب ابزار های برنامه نویسی**

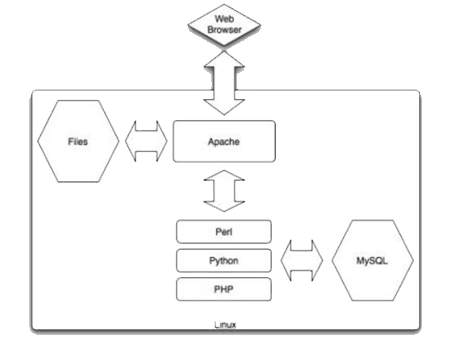
* 1. **نرم افزار phpstorm چیست؟**

**PhpStorm**یک محیط برنامه نویسی PHP ساده و هوشمند را که به هدف افزایش بهره وری توسعه دهنده ها تمرکز یافته و عمیقاً کدهای شما را درک می کند ایجاد کرده و امکان تکمیل کدها را به صورت هوشمندانه فراهم می کند، مراجعه سریع و بررسی خطاها به صورت بی درنگ از ویژگی های این نرم افزار است که همیشه برای کمک به شما در شکل دهی کدها، اجرای واحد تست و یا اشکال زدایی شهودی آماده است. این نرم افزار حاوی تمامی توابع WebStorm (HTML/CSS Editor, JavaScript Editor) بوده و پشتیبانی همه جانبه را برای PHP اضافه می کند.

* + 1. ویژگی های کلیدی نرم افزار **PhpStorm**
* ویرایشگر هوشمند PHP
* تکمیل کد PHP
* پشتیبانی از PHPDoc
* بررسی سریع
* ترکیب زبان ها (JS/SQL/XML ,...)
* ویرایشگر پیشرفته JavaScript
* ویرایشگر HTML/CSS
* محیط برنامه نویسی ساده با نصب آسان و آغاز به کار سریع
* قابل اجرا در Windows, Mac OS X, Linux
* پیکربندی راحت پروژه
* محیط هوشمند
* اشکال زدایی گرافیکی
  1. **معرفی نرم افزار lamp و نصب ان**

* + 1. معرفی:

کوتاه شده ی 4 کلمه ی PHP ,MySQL-Server ,Apache ,Linux می باشد.این 4 برنامه از معروف ترین برنامه های Open-Source در دنیای وب و اینترنت هستند که به Web Stack معروف اند. Web Stack یا بسته ی وب از لینوکس به عنوان سیستم عامل، از آپاچی به عنوان وب سرور، از MySQL به عنوان RDBMS (سیستم مدیریت پایگاه داده رابطه ای)  و از پی اچ پی به عنوان زبان برنامه نویسی شی گرا استفاده می کند.(البته ممکن است زبان های برنامه نویسی Perl یا Python جایگرین PHP شوند).



ایده استفاده از کلمه ی LAMP نخستین بار توسط آقای Michael Kunze در مجله المانی c’t در سال 1998 داده شد و در آن سال این 4 لغت برای اولین بار در یک کلمه به کار برده شدند. با وجود این نام گذاری باز هم این نتیجه را نمی توان گرفت که این برنامه ها برای کار کردن با یکدیگر ساخته شده اند. هر کدام از این برنامه ها خود به تنهایی یک برنامه ی کامل و مرجع هستند که به صورت منبع باز یا Open Source در دسترس هستند. (به این برنامه ها FOSS به معنای نرم افزار مجانی یا متن باز هم گفته می شود).برنامه های Open Source به این علت که همه کاربران به منبع آن دسترسی دارند انعطاف خوبی دارند و عیب یابی در آنها بسیار آسان تر است .

همانگونه که اشاره شد،LAMP (لامپ): لینوکس، آپاچی، مای اس کیو ال، پی اچ پی) یک مجموعه نرم افزاریست(Software Bundle) که داشتن آن برای آغاز میزبانی هر وبسایت بر روی یک کامپیوتر یا سرور ضروری است.

امروزه لمپ به یک استاندارد جهانی تبدیل گشته است، به طوری که تقریبا در تمامی لینوکس های منتشر شده عرضه می شوند تا یک برنامه ی ابر قدرت در اختیار کاربر قرار گیرد.

* + 1. نصب
       1. **نصب اپاچی**

در ابتدا ترمینال رو باز کنید و عبارت زیر وارد کرده و اینتر بزنید؛ در صورت درخواست پسورد پسورد روت خود را وارد نمایید:

sudo apt-get update

بعد از اتمام بروزسانی دستور زیر رو برای نصب وب سرور آپاچی وارد کرده و اینتر بزنید؛ در صورت درخواست پسورد، پسورد روت خود را وارد نمایید.

sudo apt-get install apache2

در صورتی که از شما سوالی پرسید عبارت Y رو به معنای Yes (بله) واردکرده و اینتر بزنید؛ صبر کنید تا بسته‌های مورد نیاز رو دانلود و نصب کنه.

بعد از اتمام نصب برای تست اینکه آپاچی به درستی کانفیگ شده دستور زیرو وارد کرده و اینتر بزنید :

sudo apache2ctl configtest

در صورتی که در خروجی دستور بالا عبارت زیر رو مشاهده کردید کانفیگ بدرستی انجام شده است :

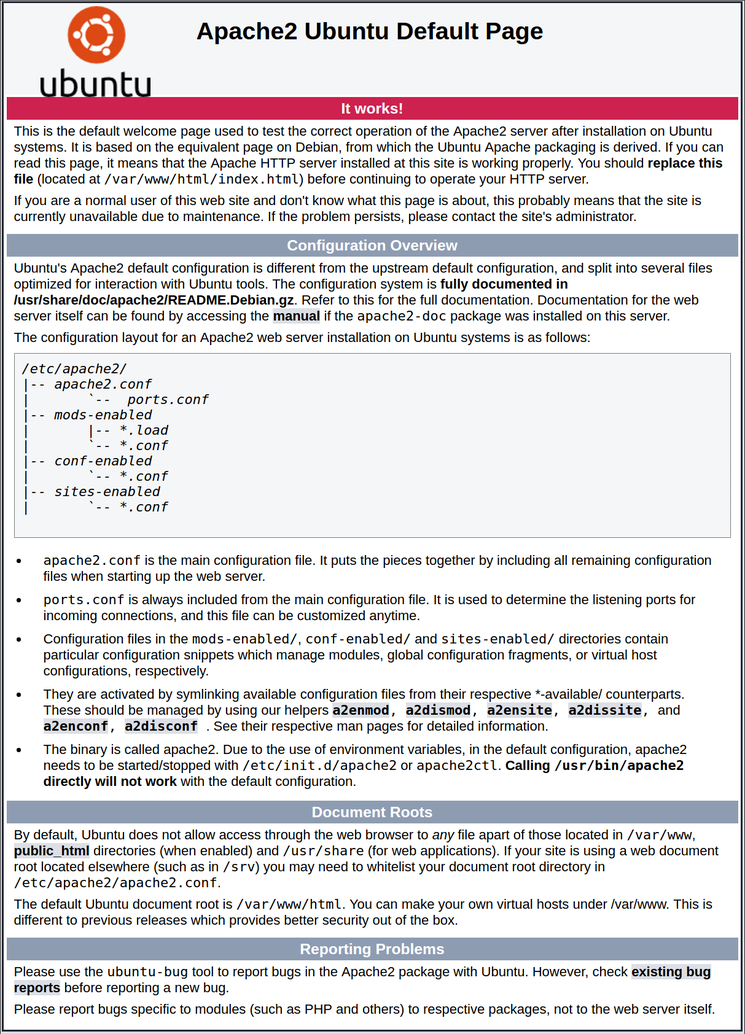
Syntax OK

خب یکبار با دستور زیر آپاچی رو برای اعمال تغییرات ریستارت میکنیم:

sudo systemctl restart apache2

حالا برای تست وب سرورمون مرورگر رو باز کرده و آدرس لوکال هاستمون رو (http://127.0.0.1) یا (http://localhost) وارد میکنیم.

در صورتی که با صفحه‌ای همانند تصویر زیر مواجه شدید همه چی درسته!



* + - 1. **نصب mysql**

بعد از نصب وب سرور آپاچی نوبت به نصب دیتابیس MYSQL میرسه؛ برای نصب دیتابیس MYSQL عبارت زیر و در ترمینال وارد کرده و اینتر بزنید.

sudo apt-get install mysql-server

بعد از وارد کردن دستور بالا می‌پرسه فلان مگابایت از فضای دیسک مورد نیاز است آیا مایل به ادامه هستید ؟! شما می‌بایست برای ادامه حرف Y وارد کرده و اینتر بزنید.

فرایند نصب دیتابیس Mysql آغاز خواهد شد و در بین مراحل نصب از شما پسوردی برای یوزر روت دیتابیس خواهد خواست که می‌بایست یک پسورد وارد کنید؛ که این پسورد؛ پسورد یورز root دیتابیس شما خواهد بود که بخاطر سپردن آن الزامی می‌باشد.

صبر کنید تا فرایند نصب کامل شود.

* + - 1. نصب**php**

خب الان دیگه نوبت به نصب خود PHP رسید؛ برای نصب پی اچ پی دستور زیر رو در ترمینال وارد کرده و اینتر بزنید (توجه داشته باشید که با دستور زیر آخرین نسخه استیبل برای شما نصب خواهد شد.

sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql

صبر کنید تا فرایند نصب کامل بشه؛ بعد از نصب شدن برای کانفیک پی اچ پی دستو زیر رو در ترمینال وارد کرده و اینتر بزنید:

sudo nano /etc/apache2/mods-enabled/dir.conf

نکته : بجای ویرایشگر nano ه یک ویرایشگر تحت ترمینال می‌باشد شما می‌توانید از ویرایشگرهایی همچمون gedit نیز استفاده کنید.توصیه میکنم اگر کاربر حرفه‌ای نیستید بجازی ویرایشگر نانو از ویرایشگر gedit با دستور زیر استفاده کنید.

sudo gedit /etc/apache2/mods-enabled/dir.conf

خب بعد از وارد کردن دستور بالا ویرایشگر متن باز می‌شود و می‌باید کد زیر رو پیدا کرده :

<IfModule mod\_dir.c>

DirectoryIndex index.html index.cgi index.pl index.php index.xhtml

index.htm

</IfModule>

و به حالت زیر تغییر دهید: (کاری که ما کردیم اینه که عبارت index.php رو قبل از index.html قرار دادیم.)

<IfModule mod\_dir.c>

DirectoryIndex index.php index.html index.cgi index.pl index.xhtml

index.htm

</IfModule>

بر روی دکمه Save کلیک کنید تا تغییرات ذخیره شود؛ سپس ویرایشگر متن رو ببنید.

خب برای اعمال تغییرات جدید بر روی وب سرور باید اونو با دستور زیر ریستارت کنیم: کافیه دستور زیر رو در ترمینال وارد کرده و اینتر بزنید :

sudo systemctl restart apache2

بعد از نصب پی اچ پی شما میتونید ماژول‌های پی اچ پی مورد نیاز خودتون رو نصب کنید برای مثال :

sudo apt-get install php-cli

* + - 1. **تست php**

تا اینجا ما مراحل کامل نصب LAMP رو باهم با موفقیت طی کردیم.اکنون زمان آن رسیده تا php که نصب کردیم رو تست کنیم و همچنین ماژول های پی اچ پی که نصب شده رو مشاهده کنیم.

دستور زیر رو برای ساخت یک فایل php وارد نمایید.

sudo gedit /var/www/html/info.php

با دستور بالا ویرایشگر متنی باز می‌شود و فایلی با عنوان info.php رو در مسیر /var/www/html/info.php ایجاد می کند.

در داخل ویرایشگر متین کدهای زیر رو قرار دهید و روی دکمه Save کلیک کنید.

<? php

phpinfo ();

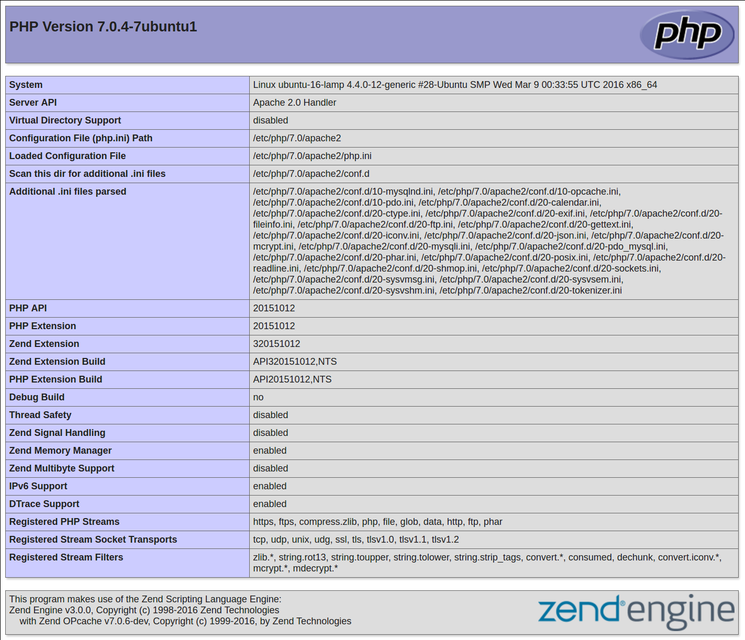
حالا مرورگر خودتون رو باز کنید و ادرس زیر رو وارد کرده و اینتر بزنید:

http://127.0.0.1/info.php

یا

http://localhost/info.php

اکنون صفحه اطلاعات پی اچ پی رو مشاهده خواهید کرد.(صفحه‌ای همانند تصویر زیر(:



خب به پایان آموزش نصب نصب Linux, Apache, MySQL, PHP (LAMP) در اوبونتو ۱۶.۰۴ رسیدیم.

* 1. **زبان ها و ابزار های استفاده شده در پروژه**
     1. **php چیست؟**

PHP مخفف PHypertext Preprocessor یک زبان قدرتمند CrossPlatform و Html Embeded برای ساخت وب سایت های پویا و داینامیک است . یک زبان سمت سرور که اسکریپت های آن بر روی سرور اجرا می شود . با استفاده از زبان php ، شما می توانید سایت ها و پورتال های بزرگ سازمانی را با قدرت طراحی و برنامه نویسی کنید . php در نسخه های مختلفی به بازار عرضه شد که در حال حاضر نسخه 7 آن توسط توسعه دهندگان php به کاربران و برنامه نویسان ارائه شده است . با آمدن php 7 تحول جدیدی در php ایجاد شد به صورتی که Microsoft سازنده زبان Asp بار دیگر از رقیب خود یعنی php عقب افتاد.

* + - 1. **ویژگی های عمومی زبان php**

هر کاربری دلایل خاص خود را برای استفاده از PHP و پیاده سازی برنامه ای حساس با آن را دارد. از جهاتی دیگر می توان نشان داد که چنین تمایلاتی به چهار گروه اصلی طبقه بندی می شوند: کاربردی بودن، قوی بودن، امکانات فراوان و هزینه مطلوب.

کاربردی بودن

از همان نسخه های اولیه، PHP با کاربردی بودن آن در ذهن، ساخته شد. هدف اولیه لردورف، طراحی زبان کاملا جدیدی نبود بلکه هدف وی، حل کردن مشکلی بود که در آن زمان، راه حل ساده برای آن وجود نداشت. بعلاوه اکثر تغییر و تحولات اولیه در این زبان، هدف اصلیشان بهبود خود زبان برنامه نویسی نبود بلکه هدف اصلی، افزودن ابزارهای جهت استفاده کاربران بود. نتیجه آن شد که زبانی بوجود آمد که به کاربران اجازه می داد حتی با کمترین دانش، برنامه هایی قدرتمند بسازند. بعنوان نمونه، یک اسکریپت مفید PHP می تواند شامل تنها یک خطPHP باشد. برخلاف C، نیازی به استفاده از اعلان برای استفاده از کتابخانه های استاندارد نیست. برای مثال، دستورات زیر، اسکریپت کالی از PHP را نشان می دهند که هدف آن، نمایش تاریخ جاری به فرمت ماه، روز، سال است:

<?php echo date("F j, Y");?>

مثالی دیگر از زیبایی این زبان در خلاصه نویسی، امکان جاسازی توابع در یکدیگر است. بعنوان نمونه، می توانید تغییرات مختلفی را با اعمال زنجیره ای توابع انجام دهید. مثال زیر رشته ای از پنج کاراکتر الفبایی نظیر a3jh8 را تولید می کند:

$randomStriong = substr(md5(microtime()), 0, 5);

PHP زبانی است که در اصطلاح برنامه نویسی به آن زبان برنامه با تایپ راحت اطلاق می شود. به این مفهوم که نیازی به بیان نوع متغیرها و یا فراخوانی نابودکننده نمی باشد، هر چند می تواند آن کارها را نیز انجام دهید. PHP این موارد را بصورت داخلی انجام می دهد، متغیرها را در مواقع لزوم و هنگامی که در اسکریپت فراخوانی می شوند می سازد و بطور خودکار از مناسبترین فرمولها استفاده میکند تا نوع متغیرها بدرستی در نظر گرفته شوند. برای نمونه، PHP دستورات زیر را کاملا معتبر می داند:

<?php

$number = "5";

$sum = 15 + $number;

$sum = "twenty";

?>

php بطور خودکار متغیرها را نابود می کند و منابع استفاده شده را در پایان اسکریپت، به سیستم بر می گرداند. این موارد و در موارد مشابه دیگری که PHP بصورت داخلی، مسائل مدیریتی و امنیتی برنامه نویسان را حل می کند، این امکان را به برنامه نویسان خواهد داد تا منحصرا تنها بر روی هدف نهایی خود تمرکز کنند که همان برنامه ای می باشد که تحت وب بدرستی کار کند.

قوی بودن

برنامه نویسان PHP بیش از 180 کتابخانه را در اختیار خود دارند که در مجموع چیزی بیش از 1.000 تابع را شامل می شود. اگرچه شما احتمالا از قابلیتهای PHP برای ارتباط با پایگاه های داده، تغییر در فرم اطلاعات و ساخت دینامیک صفحات مطلع هستید، ولی ممکن است ندانید که PHP قادر است کارهای زیر را نیز انجام دهد:

ساخت و ویرایش فایلهای در فرمت Adobe Flash و PDF

تحلیل حتی پیچیده ترین رشته ها با استفاده از کتابخانه های POSIX و عبارات قانونمند Perl

مجوز ورود به کاربران از طریق فایلهای خام، دیتابیس ها، و یا حتی Active Directory مایکروسافت

امکان برقراری ارتباط با بسیاری از پروتکلهای مختلف نظیر LDAP، IMAP، POP3، NNTP، و DNS و غیره...

امکاناتی نظیر ارتباط با بسیاری از کارتهای اعتباری

و هنوز تا به اینجا حرفی از افزونه های PHP و انبارهای برنامه نویسیApplication Repository (PEAR) به میان نیاورده ایم که صدها بسته قابل نصب را برای افزایش امکانات PHPشامل می شود.

امکانات فراوان

برنامه نویسان PHP به ندرت خود را مقید به استفاده از تنها یک روش می کنند. از طرف دیگر، کاربران معمولا از امکانات و اختیارات وسیعی که این زبان به آنها می دهد، دچار وحشت می شوند. بعنوان مثال، انتخابهایی را که PHP برای پشتیبانی از پایگاه های داده، به برنامه نویسان می دهد را در نظر بگیرید. پشتیبانی که در نسخه های اصلی موجود است، شامل پشتیبانی از 25 نوع مختلف دیتابیس است. این دیتابیس ها شامل:

Adabas D

dBase

Empress

FilePro

FrontBase

Hyperware

IBM DB2

Infomix

Ingres

InterBase

mSQL

Microsoft SQL Server

MySQL

Oracle

Ovrimos

PostgreSQL

Solid Sybase

Unix dbm

Velocis

می باشند. ضمن آنکه توابعی وجود دارند که امکان دسترسی به پایگاه های داده از نوع Berkeley DB را می دهند. چندین نوع دیگر از پایگاه های داده، به طریق مشابهی پشتیبانی می شوند که از مشهورترین آنها می توان به PDO و MDB2 اشاره کرد.

قابلیتهای انعطاف پذیر پردازش رشته ها در PHP، به کاربرانی با مجموعه مهارتهای متفاوت، امکاناتی را می دهد تا نه تنها خیلی سریع عملیات رشته ای پیچیده را انجام دهند بلکه به همان سرعت از برنامه هایشان در برنامه های مشابه دیگر نظیر Perl و Phyton استفاده نمایند. علاوه بر وجود بیش از 85 تابع برای ویرایش رشته ها، هر دو نوع شکل عبارات قانونمند Perl و POSIX پشتیبانی می شوند.

آیا ترجیح می دهید تا از زبانی استفاده کنید که از برنامه نویسی به شکل مرحله ای (Procedural programming) پشتیبانی کند؟ و همچنین از شی گرایی؟ در اینصورت PHP برای هر دوی آنها، پشتیبانی کاملی را ارائه می کند. اگرچه PHP در ابتدا زبانی کاملا متکی بر توابع بود، ولی توسعه دهندگان آن خیلی زود متوجه اهمیت شی گرایی شدند و گامهایی مناسب برای پیاده سازی آن برداشتند.

PHP به شما اجازه خواهد داد تا به سرعت و در کمترین زمان، از توانایی های فعلی خود استفاده کنید.

هزینه مطلوب

PHP بصورت رایگان در اختیار همگان قرار دارد. از زمان اولین نسخه آن تا به امروز، PHP هیچگونه محدودیتی را در استفاده، ویرایش، و توزیع مجدد آن نداشته است. در سالهای اخیر، برنامه هایی که چنین شرایطی را داشته اند، بعنوان نرم افزارهای سورس باز نام برده شده اند. پروژه های سورس باز نظیر Sendmail، Bind، Linux، و Apache همه آنها نقش بسیار بزرگی را در روند توسعه اینترنت در مقیاس وسیع داشته اند. اگرچه رایگان بودن نرم افزارهای سورس بار، مهمترین خواسته مردم است ولی چندین مشخصه دیگر نیز به همان میزان از اهمییت برخوردارند:

عدم محدودیتهایی که اکثر محصولات تجاری برای کاربران خود اعمال می کنند: کاربرانی که از نرم افزارهای سورس باز استفاده می کنند، در اکثر محدودیتهایی که با استفاده از برنامه های تجاری ملزم به رعایت آن هستند آزادی عمل بیشتری خواهند داشت. اگرچه برخی تفاوتها نیز میان مجوزهای داده شده به کاربران از برنامه ای به برنامه دیگر وجود دارد، ولی کاربران بسیار آزادی عمل دارند تا برنامه را با ویرایش و توزیع مجدد به محصول دیگری تبدیل کنند.

توسعه همگانی و فرایند رفع ایرادات: اگرچه اتفاقی نیست، ولی نرم افزارهای سورس باز، از امنیت بالاتر برخوردار می باشند. چنین استاندارد کیفی، نتیجه توسعه همگانی و فرایند رفع ایرادات است. از آنجاییکه سورس نرم افزار بطور رایگان در اختیار همگان قرار دارد تا آن را مورد بررسی قرار دهد، در نتیجه ایرادات امنیتی و مسائل آن خیلی سریع پیدا و تعمیر خواهند شد. این مزیت شاید به بهترین نحو توسط اریک س. ریموند نوشته شده باشد. وی می نویسد: "چشمان کافی در اختیار من بگذارید تا تمام باگها را بگیرم".

تشویق به مشارکت: تیمهای توسعه دهنده، به سازمان خاصی مربوط نیستند. هر کسی علاقه دارد و توانایی آن را در خود می بیند می تواند در این پروژه شرکت کند. عدم اعمال محدودیت برای اعضا، بطور وحشتناکی اعضای نخبه در یک پروژه را افزایش می دهد که در نهایت منجر به ایجاد محصولی با کیفیتی بالا خواهد شد.

* + 1. **mysql چیست**؟

MySQL یک سیستم مدیریت پایگاه داده رابطه ای (RDBMS) است که امکان ذخیره سازی، جستجو، مرتب کردن و بازیابی داده ها را فراهم می کند. سرور MySQL به چندین کاربر اجازه استفاده همزمان از داده ها را می دهد و دسترسی افراد معتبر به داده ها را تسریع می کند. بنابراین MySQL یک سرور چند کاربره و چند ریسمانه است. اگرچه تاریخچه پیدایش MySQL به سال1979 بر می گردد اما عرضه عمومی آن از سال 1996 آغاز شده است. این سیستم مدیریتی پایگاه داده، 1رایج ترین پایگاه داده کد باز (open Source) محسوب می شود.

* + - 1. **ویژگی های اصلی MySQL**

1. پشتیبانی از Sub select
2. پشتیبانی از view
3. متن باز بودن یا همان open source
4. پشتیبانی از stored procedure
5. پشتیبانی از trigger
6. پشتیبانی کامل از یونی کد (Unicode)
7. رابطه های کاربری گوناگون مانند PHPMYADMIN ,WORKBENCH , SQLMASTERO , NAVICAT و …
8. پشتیبانی از Full Text Search
9. پشتیبانی از Replication
10. پشتیبانی از Transaction
11. دارای توابعی برای کار با داده های مکانی و مختصات جغرافیایی
12. موتورهای ذخیره‌سازی مختلف

* + - 1. **رقبای mysql عبارت اند از**:

PostgreSQL

SQL Server

Oracle

* + - 1. **مزیت های mysql**

کارائی بالا

هزینه پایین

سهولت پیکربندی و یادگیری

قابلیت حمل

در دسترس بودن کد اصلی

خدمات پشتیبانی مناسب

* + 1. **apache چیست؟**

**وب سرور آپاچی(Apache) گسترده ترین و محبوب ترین سرور HTTP در دسترس بر روی اینترنت می باشد که از زبان های PHP و Perl پشتیبانی می کند و روی بیشتر سیستم عامل های قابل اجرا است یک برنامه free Open Source است که با سرور های وب برای اداره کردن درخواستها و تقاضاهای وب و منابع به کار می رود .**

**Apache HTTP Server روی سیستم عامل Unix مانند Linux یا BSD اجرا می شود همچنین می تواند روی Windows مایکروسافت و دیگر سیستم عامل ها یا Platform ها اجرا شود . یک سرور با خصوصیات منحصر به فرد با add-on های قدرتمند که به صورت مجانی در دسترس هستند .**

**Apache دارای امکانات ویژه می باشد که متداول ترین استفاده از ویژگی های این برنامه .htaccess است که طراحان حرفه ای در محیط لینوکس از آن بهره می گیرند . برای نمونه زمانی که بخواهند اولین صفحه در سایت بخصوصی باشد با یک دستور در آن پرونده این امر ممکن می گردد و یا زمانی که صاحب سایت مایل نیست که فایلهای موجود در سرور وی توسط دیگران دزدیده شود و بخواهد که مانع از پیوند مستقیم آنها شود Apache کمک می کند تا به خواستشان برسند . زمانی که برنامه نویس بخواهد که محل واقعی صفحات دیده نشود نیز این برنامه مورد استفاده قرار می گیرد .**

* + - 1. **ویژگی های اصلی apache**
* **ویژگی‌های این برنامه بسیار گسترده است؛ از پشتیبانی زبان‌های برنامه‌نویسی سمت سرور گرفته تا تصدیق هویت.**
* **تعدادی از زبان‌های برنامه‌نویسی قابل پشتیبانی عبارتند از: perl, python, TCL و PHP.**
* **ماژول‌های تصدیق هویت شامل: mod\_access, mod\_auth,... . پشتیبانی از SSL, TLS.**
* **پشتیبانی از URL rewiter یا بازنویسی URL.**
* **پشتیبانی از ثبت وقایع در فایل به‌صورت اختصاصی.**
* **پشتیبانی از روش‌های فشرده‌سازی معمول و عمومی و نیز قابل توسعه همچون mod\_gzip که برای کاهش حجم صفحات وب که در اینترنت تبادل پیدا می‌کنند استفاده می‌شود.**
* **modSecurity که یک برنامه متن‌باز برای تشخیص و جلوگیری از نفوذ به برنامه‌های تحت وب است.**
* **وقایع ثبت شده آپاچی می‌تواند توسط اسکریپت‌های رایگان همچون AWStats, W3Perl یا Visitors مورد تحلیل قرار گیرد و توسط مرورگر در اختیار تحلیل‌گر بازدید قرار گیرد.**
* **VirtualHost قابلیتی را فراهم می‌کند که یک نسخه از آپاچی برای وب سایت‌های بسیاری مورد استفاده قرار گیرد. به عنوان مثال: www.example.com , www.test.com , test47.testserver.test.com.**
* **قابلیت تنظیم و اختصاصی نمودن انواع پیام‌های خطا.**
* **تصدیق هویت براساس بانک اطلاعاتی.**
* **همچنین آپاچی توسط چندین برنامه رابط گرافیکی پشتیبانی می‌شود.**
* **contentnegotiation ارایه چند محتوای متفاوت برای یک آدرس. برای این استفاده می‌شود که چند سند مختلف برای یک آدرس ارسال شود. کاربرد آن مثلاً در زمانی است که IE4 از فرمت PNG پشتیبانی نمی‌کند و به جای آن شما نسخه gif را برای مرورگر ارسال می‌کنید. یا مثلاً براساس UserAgent نسخه موبایل برای مرورگر ارسال می‌کنید.**

فصل دوم

**طراحی پایگاه داده**

* 1. جداول:

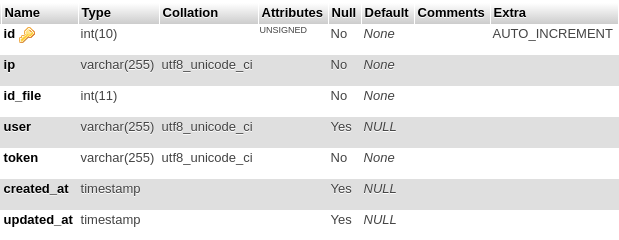
یکی از مهمترین بخش یک پروژه پایگاه داده و طراحی ان میباشد.درواقع تمام اعمالی که روی یک صفحه وب انجام میشود بر مبنای پایگاه داده است و پایگاه داده ای که با در نظر گرفتن تمام شرایط طراحی شود و حاوی فیلد های لازم باشد، بهینگی کار را بالا میبرد.

همانطور که در پیش تر بیان شد ، پایگاه دادی که برای این پروژه طراحی شده است شامل 7 جدول میباشد که در تصویر زیل مشاهده میشود.و در ادامه به شرح جداول و فیلد های هر یک میپردازیم.



* 1. جدول downloads

این جدول شامل 7 فیلد می باشد و برای ثبت و مدیریت لینک های ساخته شده برای هر فایل می باشد که هر کدام از فیلد ها به شرح زیر می باشد:



id: این فیلد به عنوان کلید اصلی در نظر گرفته شده است که به صورت اتوماتیک پر میشود.

Ip: این فیلد برای محدود کردن لینک دانلود ساخته شده برای فقط یک ایپی است تا دیگر کاربران برای خود لینک بسازند.

Id\_file: این فیلد برای تایین کردن فایل ی است که برای ان لینک ساخته شده است.

User: این فیلد تایین میکند که این لینک برای چه کاربری ساخته شده است.

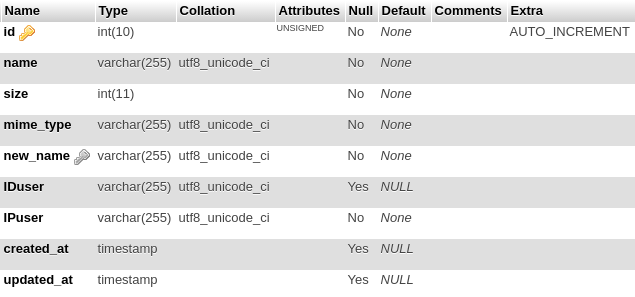
Token: این فیلد برای توکن لینک ساخته شده است که به همراه ادرس دریافت میگردد برای امنیت دانلود و تایین مشخصات فایل درخواست شده.

created\_at: برای ثبت زمان و تاریخ ایجاد رکورد است.

updated\_at: برای ثبت زمان و تاریخ اخرین ویرایش رکورد است.

* 1. جدول files:

این جدول شامل 9 فیلد میباشد که برای ثبت و مدیریت فایل ها و اپلود های انجام شده توسط کاربران و میهمانان می باشد که هر کدام از فیلد ها به شرح زیر می باشد:



id: این فیلد به عنوان کلید اصلی در نظر گرفته شده است که به صورت اتوماتیک پر میشود.

Name: نام اصلی فایل اپلود شده.

Size: ساز فایل اپلود شده.

Mime\_type: نوع فایل اپلود شده.

New\_name: برای امنیت بیشتر و جلوگیری از اپلود شل برای هر فایل یک اسم tmp ساخته میشود که فقط خود سیستم میداند و ان را در دیتابیس ذخیره کرده است. زمان دانلود به کاربر اسم اصلی فایل به فایل نوشته شده و برای دانلود ارسال میگردد.

IDuser: ایدی یوزری که فایل را اپلود کرده است. اگر یوزر ثبت نام نکرده باشد و به صورت میهمان اپلود کند در این فیلد عدد 0 نوشته میشود.

IPuser: در این فیلد ای پی کاربر ثبت میشود تا در صورت وجود مشکل در اختیار مرجع قضایی قرار داده شود.

created\_at: برای ثبت زمان و تاریخ ایجاد رکورد است.

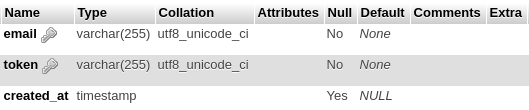
updated\_at: برای ثبت زمان و تاریخ اخرین ویرایش رکورد است.

* 1. جدول migrations :

این جدول مختص فریم ورک لاراول میباشد که در بخش مایگریشن ها توضیح خواهیم داد.

* 1. جدول password\_resets :

این جدول شامل 3 فیلد می شود که برای ثبت لینک های فرامویش میباشد که بعد از ساخته شدن به کاربر مورد نظر ایمیل میگردد که هر کدام از فیلد ها به شرح زیر می باشد:



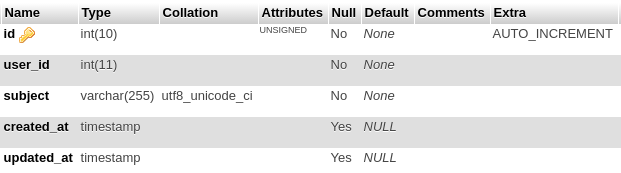
Email: ایمیل کاربری که لینک بازیابی رمز را ساخته است.

Token: توکن لینک ساخته شده برای تایین صحت و باز یابی اطلاعات لینک.

created\_at: برای ثبت زمان و تاریخ ایجاد رکورد است.

* 1. جدول requests:

این جدول شامل 5 فیلد می شود که برای ارتباط مدیر سیستم و کاربران ساخته شده است که کاربران میتوانند درخواست ها و پیشنهادات و انتقادات خود را ثبت نمایند ، هر کدام از فیلد ها به صورت زیر است:



id: این فیلد به عنوان کلید اصلی در نظر گرفته شده است که به صورت اتوماتیک پر میشود.

User\_id: کاربری که این نظر و یا درخواست را نوشته است.

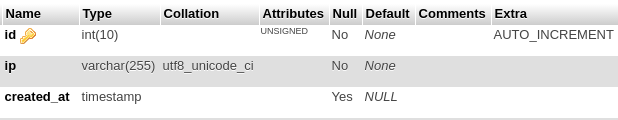
Subject: موضوع پیام و یا درخواست.

created\_at: برای ثبت زمان و تاریخ ایجاد رکورد است.

updated\_at: برای ثبت زمان و تاریخ اخرین ویرایش رکورد است.

* 1. جدول statistics:

در این جدول امار بازدید از سایت ثبت میشود که هر کدام از فیلد ها به صورت زیر است:



id: این فیلد به عنوان کلید اصلی در نظر گرفته شده است که به صورت اتوماتیک پر میشود.

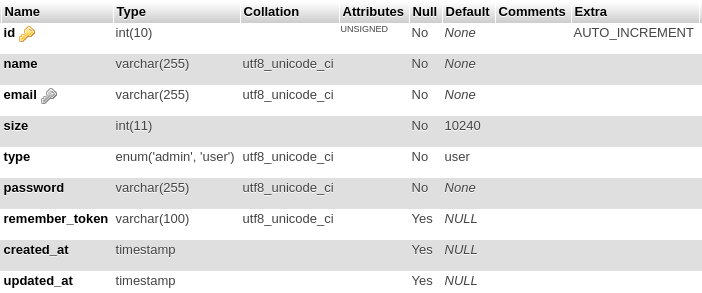
Ip: برای جلوگیری از تکراری بودن بازدید از صفحه.

created\_at: برای ثبت زمان و تاریخ ایجاد رکورد است.

updated\_at: برای ثبت زمان و تاریخ اخرین ویرایش رکورد است.

* 1. جدول users:

این جدول مشخصات مربوط به کاربران قرار میگیرد.



id: این فیلد به عنوان کلید اصلی در نظر گرفته شده است که به صورت اتوماتیک پر میشود.

Name: نام کاربر ثبت نام کرده برای نمایش در پنل.

Email: ایمیل کاربر ثبت نام کرده برای استفاده دز سیستم احراض هویت.

Size: محدودیت سایز قابل اسفاده برای هر کاربر که در صورت اتمام این محدودیت در پنل مدیر به صورت قرمز نشان داده خواهد شد و کاربر حق اپلود فایل نخواهد داشت. این فیلد به صورت دیفالت مقدار 10240 (10 مگابایت) قرار داده شده است.

Type: نوع کاربر ثبت شده است که به صورت enum وارد میشد یاadmin یا user میباشد. که به صورت دیفالت مقدار user وارد میشود.

Password: در این فیلد پس ورد کاربر نگه داری میشود و در احراص حویت از ان استفاده کنیم.

remember\_token: به خاطر جلوگیری از باگ xcf هنگام ورود یک توکن ساخته میشود و این توکن باید در هر بار ارسال فرم ارسال شود و اعتبار سنجی انجام شود.

created\_at: برای ثبت زمان و تاریخ ایجاد رکورد است.

updated\_at: برای ثبت زمان و تاریخ اخرین ویرایش رکورد است.

فصل سوم

**Frame work**

* 1. **Frame work چیست؟**

**Framework به معنای چارچوب کاری است و این واژه در حوزه‌های متنوعی مورد استفاده قرار می‌گیرد. فریم‌ورک یا چارچوب کاری در زبان‌های برنامه نویسی به مجموعه‌ای از قوانین و کتابخانه ها گفته می‌شود که کار برنامه نویسی را نظام مند و قدرتمند پیش می‌برند.**

**تقریبا هر زبان برنامه نویسی چند فریم‌ورک قدرتمند برای خود دارد، البته ذکر این نکته ضروری است که در تمام زبان‌های برنامه نویسی استفاده از فریم‌ورک‌ها هر چند ضروری است ولی بدون فریم‌ورک نیز باید بتوان کار توسعه نرم افزار را انجام داد.**

**بسیار شنیده ایم که کاربران به عنوان مثال برنامه نویسی PHP را با برنامه نویسی دات نت مقایسه می‌کنند. این مقایسه از اساس اشتباه است. چون PHP یک زبان مستقل برنامه نویسی است ولی دات نت یک فریم‌ورک است که برای زبان برنامه نویسی C# و یا ASP پیاده سازی شده است. پس درست این است که PHP را با زبان C# مقایسه کنیم.**

**در PHP فریم‌ورک‌ها بسیار متنوع و قدرتمند هستند. مانند فریم‌ورک های:**

**لاراول**

**سیمفونی**

**زند فریم‌ورک**

**کیک**

**Codeigniter**

* 1. **چرا باید از فریم ورک استفاده کنیم؟**

**به طور خلاصه فریم‌ورک‌ها کار ساخت نرم افزارهای حرفه‌ای و امن را بسیار ساده تر و قدرتمند تر می‌کنند.**

**فریم‌ورک‌ها توسط برنامه نویسان قدرتمندی نوشته می‌شوند که علاوه بر تعریف چارچوب کاری نظام مند کتابخانه‌های قدرتمندی نیز به فریم‌ورک خود متصل می‌کنند که شما را از نوشتن بسیاری از کتابخانه‌های بی نیاز می‌کند.**

**مثلا تقریبا در تمام فریم‌ورک‌ها کتابخانه‌هایی برای ارتباط با پایگاه داده وجود دارد که فرایند ارتباط با پایگاه داده شامل دریافت و ذخیره اطلاعات را تسهیل می‌کنند.**

**پایگاه‌های داده به دلیل آسیب پذیری بالایی که دارند در معرض حجم وسیعی از تهدیدها هستند که کمتر برنامه نویسی می‌تواند برنامه خود را با رعایت کلیه نکات امنیتی پیاده سازی کند.**

**متخصصانی که فریم‌ورک‌ها را پیاده می‌کنند برای حل این چالش از کتابخانه‌ای به نام ORM استفاده می‌کنند که تقریبا تمام نکات امنیتی ارتباط و استفاده از پایگاه داده را رعایت می‌کنند.**

* 1. **ORM چیست؟**

**ORM‌ها و یا Object-relational mapping ها شاید مهم ترین بخش هر فریم‌ورک باشند ولی پیاده سازی آن‌ ها نیازمند دانش، تجربه و تخصص بالاست که با استفاده از ORM در نرم افزار خود میزان زیادی از دغدغه‌های شما را حل می‌کند. ORM باعث تبدیل داده‌های ساده به Object‌ های مناسب برای استفاده در برنامه می‌شود. اطلاعات در سیستم‌ های پایگاه داده رابطه‌ای به صورت غیر شی گرا هستند و برای سهولت در استفاده از داده‌های پایگاه داده در برنامه‌های شی گرا از ORM استفاده می‌شود تا این داده‌ها تبدیل به داده‌های مناسب برای استفاده در زبان‌های شی گرا شوند. در زبان برنامه نویسی PHP شاید هیچ ORM‌ای به اندازه Doctrine معروف و مهم نباشد. ولی ORM‌ های داخلی Cake PHP و ZendFramework و Laravel نیز بازدهی خوبی دارند. یکی از مزیت‌های مهم استفاده از ORM‌ ها وجود امکان کوچ ساده در بین سیستم‌های مدیریت دیتابیس است. مثلا شما کار توسعه نرم افزار خود را با سیستم پایگاه داده MySQL شروع می‌کنید. در ادامه کار سیستم خود ممکن است مجبور باشید به جای MySQL از MongoDB استفاده کنید. در اکثر ORM‌ها تغییر سیستم مدیریت دیتابیس به راحتی امکان پذیر است. در صورتی که اگر از ORM استفاده نکنید تغییر سیستم پایگاه داده به معنای دوباره نوشتن حجم بزرگی از کدهای خود و تست‌های مکرر برای اطمینان از صحت عملکرد نرم افزار با سیستم پایگاه داده جدید می‌باشد.**

* 1. **MVC در فریم‌ورک‌ها به چه معناست؟**

**ام وی سی، در واقع یک مدل ارتباطی برای ارتباط اجزاء درونی یک فریم‌ورک است. MVC مخفف عبارات Controller، View ، Model است. model مربوط به بخش پایگاه داده، View مرتبط با لایه نمایشی و ارتباط با کاربران و Controller برای ایجاد ارتباطات درون نرم افزار بین View و Model است. البته خلاصه کردن فعالیت‌های فریم‌ورک به MVC خیلی کلی است و هر فریم‌ورک تعریف خاصی از MVC برای خود دارد.**

* 1. **فریم ورک لاراول چیست؟**

**Laravel یکی از فریم‎ ورک‎ های زبان PHP است که برای توسعه اپلیکیشن ‎های وب در نظر گرفته شده است و بر پایه MVC کار می‎ کند. فریم‎ ورک لاراول، برنامه‎ نویسی برنامه‎ های کاربردی تحت وب با زبان PHP را ساده‎تر می‎ نماید و کمک بسزایی برای انجام پروژه ‎های PHP و توسعه آسان آن‎ها می‎ کند. فریم ‎ورک Laravel بر روی اجزای مختلف فریم ورک symfony ساخته شده است و به برنامه شما پایه‎ای بزرگ از کد ‎های قابل اعتماد و تست شده می ‎دهد. لاراول مجموعه ای از بهترین راه حل ها با سینتکس پر معنا و خلاقانه را ارائه می‌کند که به درستی انجام می‌ پذیرند. لاراول توسط آسان سازی کارهای معمول مانند احراز هویت، روتینگ، sessionها ، کار با بانکهای اطلاعاتی و … که تقریبا در تمامی پروژه های تحت وب استفاده می‌شوند، مسائل و مشکلات ناشی از توسعه را هم برای توسعه دهنده و هم برای کارفرما کاهش می دهد. لاراول، سیمفونی، کیک پی اچ پی و کد ایگنایتر از محبوب ترین فریم ورک های زبان PHP هستند که بررسی میزان محبوبیت آنها در گوگل ترند حاکی از رشد روز افزون فریم ورک Laravel است.**

* 1. **چرا لاراول ؟**
* **یادگیری و بروز ‎رسانی آسان**
* **آسان‏ سازی کارهای معمول در روند برنامه ‎نویسی نظیر احراز هویت، روتینگ، جلسات و کَش**
* **راه ‎اندازی سریع و آسان محیط‎ ها و روابط چندگانه**
* **بهره ‎گیری از ریموت کامپوننت**
* **محدود‎کردن Eloquent با دستورات ساده**
* **بهبود سرعت و افزایش ۱۰۰ درصدی سرعت مسیرها**
* **دلپذیر کردن فرآیند توسعه برای توسعه ‎دهندگان بدون به خطر انداختن قابلیت‎های برنامه**
* **استفاده از شیوه استاندارد Bundle در نسخه ‎های اولیه و استفاده از پکیج ‎ها در نسخه جدید**
* **اجرای آزمایش‎ های واحد متعدد با دستورات بسیار ساده**
* **دسترسی ساده به دیتابیس و سهولت در تغییر وکنترل کدها و بروز رسانی و توسعه برنامه ‎ها**
* **استفاده از موتور تپملیت Blade**

* 1. **کامپوزر چیست؟**

**پیش از اینکه ببینیم کامپوزر composer)) چیست، می بایست با مفهومی تحت عنوان Dependency Management آشنا شویم. زمانی که ما اقدام به برنامه نویسی می کنیم، فارغ از اینکه از چه فریم ورک یا زبان برنامه نویسی برای کدنویسی استفاده می‌کنیم گاهی اوقات نیاز به استفاده از کدها، پکیج ها، و لایبرری های سایر برنامه نویسان که به صورت متن باز در اینترنت عرضه شده‌اند داریم تا بتوانیم به جای آنکه خود اقدام به نوشتن کلاس خاصی کنیم، از کدهایی که قبلاً توسط سایر توسعه دهندگان نوشته شده و آزمایش خود را پس داده‌اند استفاده کنیم که از جمله این Dependency ها می‌توان به فریم ورک های سی اس اس، پلاگین های جی کوئری و سایر پروژه اشاره کرد.**

**به طور خلاصه، کامپوزر composer)) ابزاری است که فرایند مدیریت Dependency ها را برای ما تسهیل می کند. برای آشنایی بیشتر با مفهوم مدیریت Dependency مثال ساده‌ای می زنیم. فرض کنید که شما قصد دارید یک وب سایت تک صفحه‌ای طراحی کنید و برای کدهای سی اس اس و جاوا اسکریپت خود می‌خواهید از فریم ورک فاندیشن استفاده کنید. برای استفاده از این فریم ورک دو راه‌کار پیش رو است: اول اینکه به وب سایت رسمی این فریم ورک رفته و پکیج آن را دانلود کرده و درون پروژه خود قرار دهید اما با استفاده از این روش اگر بخواهید کدهای فریم ورک خود را آپدیت کنید چه راه‌کاری به ذهن شما می رسد؟ این این فرایند دانلود، کپی و پیست کردن را مجدد تکرار می‌کنید یا راه‌کار به مراتب بهتری هم هست؟ نکته مهم اینجا است که اگر شما روی پروژه ای کار کنید که به جای یک فریم ورک، از چندین فریم ورک مختلف در آن استفاده شده باشد کار شما به مراتب دشوار تر خواهد شد.**

**این همان چالشی است که مفهوم مدیریت Dependency آن را حل می کند. در واقع، با استفاده از مفهوم Dependency Management می‌توان ماژول ها، پکیج ها و فریم ورک های مد نظر خود را به صورت به مراتب بهتری انجام داد. به عبارت دیگر، مدیریت منابع خارجی روش دانلود پکیج ها، آپدیت آن‌ها و … را استانداردسازی کرده و با استفاده از این رویکرد، مدیریت پروژه شما به مراتب آسان‌تر خواهد شد. حال که متوجه شدیم مدیریت Dependency چیست، اکنون می بایست با ابزاری آشنا شویم که این کار را به سادگی برای ما به انجام خواهد رساند و این ابزار چیزی نیست جزء Composer. کامپوزر composer)) در ابتدای ماه مارس سال ۲۰۱۲ به صورت رایگان به دنیا عرضه شد و تاکنون استقبال بسیار خوبی از این ابزار به‌خصوص از طرف برنامه نویسان زبان برنامه نویسی PHP صورت گرفته است.**

**به طور خلاصه، کامپوزرcomposer) ) یک ابزار Command Line است که در آن با استفاده از یک سری دستورات از پیش تعریف شده، می‌توان به مدیریت منابع خارجی برای استفاده در پروژه خود پرداخت. حال ممکن است این سؤال برای شما پیش بیاید که ابزار کامپوزر composer)) پکیج های مورد نیاز را از کجا دانلود می کند. در پاسخ به این سؤال باید گفت که منبع اصلی که کامپوزر از آن استفاده می‌کند سایتی تحت عنوان Packagist.org است که به عنوان فضایی می باشد که توسعه دهندگان فریم ورک هایی که می‌نویسند را در آنجا آپلود کرده و از آن پس با استفاده از کامپوزر می‌توان به کلیه پکیج های قرار گرفته روی این سایت دسترسی داشت. کامپوزر را هم روی ویندوز می‌توان نصب کرد و هم روی سیستم عامل هایی که بر پایه یونیکس هستند مثل مکینتاش و لینوکس زمانی که شما می‌خواهید از این ابزار استفاده کنید، صرفاً نیاز به یک فایل تحت عنوان composer.json خواهید داشت که کدهای قرار گرفته داخل آن چیزی شبیه به کدهای زیر است:**

**{**

**"name": "laravel/laravel",**

**"description": "The Laravel Framework.",**

**"keywords": ["framework", "laravel"],**

**"license": "MIT",**

**"type": "project",**

**"require": {**

**"php": ">=5.6.4",**

**"laravel/framework": "5.3.\*"**

**},**

**"require-dev": {**

**"fzaninotto/faker": "~1.4",**

**"mockery/mockery": "0.9.\*",**

**"phpunit/phpunit": "~5.0",**

**"symfony/css-selector": "3.1.\*",**

**"symfony/dom-crawler": "3.1.\*"**

**},**

**"autoload": {**

**"classmap": [**

**"database"**

**],**

**"psr-4": {**

**"App\\": "app/"**

**}**

**},**

**"autoload-dev": {**

**"classmap": [**

**"tests/TestCase.php"**

**]**

**},**

**"scripts": {**

**"post-root-package-install": [**

**"php -r \"file\_exists('.env') || copy('.env.example', '.env');\""**

**],**

**"post-create-project-cmd": [**

**"php artisan key:generate"**

**],**

**"post-install-cmd": [**

**"Illuminate\\Foundation\\ComposerScripts::postInstall",**

**"php artisan optimize"**

**],**

**"post-update-cmd": [**

**"Illuminate\\Foundation\\ComposerScripts::postUpdate",**

**"php artisan optimize"**

**]**

**},**

**"config": {**

**"preferred-install": "dist"**

**}**

**}**

**همان‌طور که در کد فوق ملاحظه می شود، نیازمندی های پروژه شما در بخش require آمده است. حال صرفاً نیاز است تا از طریق پنجرهCommand Line وارد فولدر پروژه مد نظر خود شده و دستور composer install را اجرا کنیم. بسته به پکیج هایی که در بخش require تعریف کرده ایم، کلیه پکیج ها دانلود شده و برخی تنظیمات اولیه هم به صورت خودکار صورت می پذیرد.**

**یکی دیگر از وظایف composer مدیریت لود کلاس ها می باشد و عملکرد آن به گونه ای است که تنها زمانی یک کلاس لود خواهد شد که شما در کد آن را فراخوانی کرده باشید که این امر تاثیر بسیار زیادی در سرعت لود پروژه شما خواهد داشت .**

**مدیریت وابستگی ها در حقیقت مفهومی جدید نمی باشد و در واقع composer از ابزار NPM در Nodjs و bandler در Roby الهام گرفته شده است . همچینین composer در حال حاضر برای php نسخه 5.3.2 به بالا قابل اجرا است ، برای جلوگیری از خطاهای احتمالی حدالمقدور از php نسخه 5.3.4 به بالا استفاده شود .**

**شما ممکن است درباره‌ی PEAR هم اطلاعاتی داشته باشید. PEAR یک مدیر بسته برای PHP می‌باشد که سال‌ها است مورد استفاده قرار می‌گیرد. اما PEAR توسط خیلی از توسعه‌دهنده‌گان PHP به چند دلیل دیگر زیاد مورد استفاده نیست. یکی از این دلیل‌ها این است که خیلی از کدها در PEAR به روز نیستند.**

* 1. **نصب کامپوزر؟**

**نصب Composer واقعا آسان است و خیلی راحت توسط خط فرمان می‌توان آن را نصب کرد. برای نصب Composer از این فرمان استفاده کنید:**

**$url -s https://getcomposer.org/installer | php**

**$sudo mv composer.phar /usr/local/bin/composer**

**اولین فرمان فایل composer.phar رو دانلود می‌کند و فرمان دوم فایل composer.phar را به داخل پوشه اجرایی منتقل می‌کند.**

**حال با نوشتن فرمان زیر می‌توانید آن را اجرا کنید:**

**$composer**

* 1. **نصب و راه اندازی فریم ورک لاراول**

**ابتدا باید یک ترمینال جدید باز کرده و اجازه دهید با استفاده از Composer اولین پروژه لاراول خود را ایجاد کنیم:**

**composer create-project laravel/laravel firstapp --prefer-dist**

**این دستور نسخه جدید از لاراول را با نام پوشه firstapp در داخلی دایرکتوری جاری ایجاد می کند. در حال حاضر لاراول شامل یک وب سرور ساخته شده در خود دارد.**

**بنابراین ما می توانیم پروژه خود را در هر نقطه نصب کنیم. هنگام اتمام نصب در آخر به شما کلید برنامه را می دهد. پس چطور کمی بوی شیرین موفقیت را احساس کنیم؟ ترمینال خود را با دستور clear پاک کنید و یا یک پنجره جدید باز کنید. آنگاه با استفاده از دستور cd به داخل پوشه FirstApp وارد شود، حال زمان آن است که وب سرور خود را با دستور زیر ایجاد کنیم:**

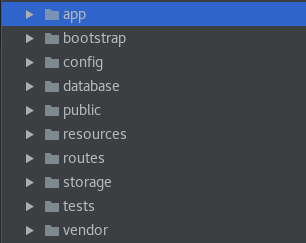
**php artisan serve**

**این دستور مخصوص محیط لاراول است. حال شما پیغامی را با Port مشخص مشاهده خواهید کرد که می توان از آن در مرورگر استفاده کرد. وب سرور را هر موقع که خواستید غیر فعال کنید با دکمه های ترکیبی Ctrl+C این کار را انجام دهید، ولی ابتدا اجازه دهید برنامه را با دستور زیر در مرورگر خود مشاهده کنیم:**

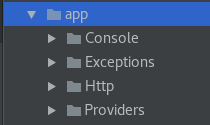
**http://localhost:8000/**

* 1. **پوشه های لاراول**

**به صورت پیش فرض ساختار دایرکتوری ها در فریم ورک لاراول طوری در نظر گرفته شده است که نقطه ی شروع مناسبی برای برنامه های کوچک و بزرگ باشد . البته شما برای سازماندهی کردن دایرکتوری های برنامه ی خودتان ,کاملا آزاد هستین و تقریبا می توان گفت هیچ محدودیتی از طرف لاراول برای محل قرار گیری کلاس ها وجود ندارد و کامپوزر می تواند آنها را بارگزاری نماید .**



* **فولدر bootstrap شامل فایل های اندکی به منظور راه اندازی فریم ورک و پیکربندی های خودکار می باشد .**
* **فولدر config همانطور که از نام آن می توان حدس زد شامل فایل های تنظیمات و پیکربندی های برنامه ی شما می باشد .**
* **فولدر database شامل تمام migration و seeds های برنامه ی شما می باشد .**
* **فولدر public شامل تمام فایل های بخش Front پروژه و assets ها می باشد (عکس ها ,استایل ها ,فایل های جاوااسکریپت و غیره)**
* **فولدر resources شامل تمام view ها , فایل های مربوط به زبان و همچنین فایل هایی مانند LESS, SASS, CoffeeScript می باشد .**
* **فولدر storage شامل تمام فایل های کامپایل شده قالب Blade , سشن ها ,کش ها و دیگر فایل های تولید شده به وسلیه فریم ورک می باشد .**
* **فولدر test شامل فایل های تست خودکار می باشد.**
* **فولدر vendor شامل تمام پکیج های مورد نیاز می باشد .**
  + 1. **App دایرکتوری**



**قسمت اصلی برنامه ی شما در فولدر app می باشد و این فولدر تحت فضای نامی App است و به صورت خودکار به وسیله ی کامپوزر و استاندارد PSR-4 autoloading standard بارگزاری می شود .شما می توانید با فرمان آرتیسان app:name این فضای نامی را به دلخواه خود عوض نمایید.**

**فولدر app شامل فولدرهای مختلفی مانند Console, Http و Providers می باشد. فولدرهای Console, Http را به منظور ارایه دهنده ی API ای داخل برنامه ی خود درنظر بگیرید و همچنین پروتوکل HTTP و CLI هر دو مکانیزمی به منظور تعامل با برنامه شما می باشند.به عبارت دیگر آنها ۲ راه ساده ی ارسال پیام در برنامه ی شما هستند. فولدر Console شامل تمام فرمان های Artisan هست و فولدر Http شامل تمام کنترلرها , فیلترها و requests ها می باشد . فولدر Commands در واقع محلی برای تمام commandsهای برنامه شما می باشد . فولدر Events همانطور که از نام اش انتظار می رود محل نگهداری کلاس های event می باشد .برای استفاده از event ها لازم نیست از این کلاس ها استفاده کنید , با این حال اگر می خواهید از آنها استفاده کنید , این فولدر محل پیشفرض event هایی است که با فرمان Artisan ایجاد می شوند .**

**فولدر Handlers شامل تمام کلاس های handler برای commands و events ها می باشد.Handlers ها یک command یا event را دریافت می کنند و زمانی که آنها فراخوانی می شوند , منطق آنها را اجرا کرده و در پاسخ به آنها ارسال می کند . فولدر Services شامل سرویس هایی است که برنامه ی شما در طول اجرا به آن نیاز دارد .برای مثال سرویس Registrar مسئول اعتبارسنجی و ایجاد یک کاربر جدید در برنامه ی شما است و یا سرویس هایی که با API های بیرونی ارتباط برقرار می کنند و یا حتی سرویس هایی که داده ها را در برنامه ی شما جمع آوری می کنند . فولدر Exceptions شامل تمام استثناها و exception های برنامه ی شما است و همچنین محل مناسبی برای رسیدگی به exception های رخ داده شده در برنامه می باشد .**

**توجه داشته باشید: بسیاری از کلاس های داخل فولدرapp را می توان با فرمان هایartisan ایجاد نمود. به منظور مرور این فرمان ها در ترمینال خود کد زیر را اجرا کنید:**

**php artisan list make**

**و با کد زیر هم میتوانید namespace پوشه App را به نام دلخواه خود تغییر دهید:**

**php artisan app:name UploadCenter**

فصل چهارم

فایل های پروژه

* 1. مدل ها
     1. مدل چیست؟

**ابزار ORM ایلوکوئنتEloquent) ) که در کنار لاراول ارائه شده، پیاده سازی ساده و زیبایی از ActiveRecord برای کار با پایگاه داده دارد. در این پیاده سازی هر جدول از پایگاه داده دارای یک مدل است. از این مدل برای تعامل با جدول در پایگاه داده استفاده میشود.**

**پیش ازآغاز، باید کانکشن به پایگاه داده را در فایل config/database.php ایجاد نمایید.**

**برای شروع، یک مدل ایلوکوئنتEloquent) ) بسازید. مدلها معمولا در دایرکتوری app قراردارند، اما میتوانید آنها را کجا که امکان بارگذاری خودکارauto-load) ) وجود دارد قرار دهید. این موضوع را از طریق فایل composer.json میتوانید بررسی کنید. تمامی مدلهای ایلوکوئنت از Illuminate\Database\Eloquent\Model ارث بری میکنند.**

**تعریف یک مدل ایلوکوئنت:**

**class User extends Model {}**

**همچنین مدلهای Eloquent را میتوانید با استفاده از فرمان make:model ایجاد نمایید:**

**php artisan make:model User**

**توجه داشته باشید ما به Eloquent در مورد اینکه از کدام جدول برای مدل User استفاده کند چیزی نگفتیم. در این مواقع حالت جمع انگلیسی نام کلاس در برای اسم جدول استفاده میشود، اما میتوانید نام جدول را هم استفاده کنید. بنابراین در این مورد، Eloquent فرض را بر این میگذارید که مدل User رکوردها را در جدول users ذخیره میکند. در صورت متفاوت بودن نام مدل با نام جدول، آن را میتوان با استفاده از خصوصیت table بر روی مدل تعریف کرد.**

**Class User extends Model {**

**protected $table = 'my\_users';**

**}**

**پس از تعریف مدل، آماده بازیابی و ایجاد رکورد در جدول خود هستید. توجه داشته باشید که باید در همه جدولها دو ستون updated\_at و created\_at را بیافزایید. اگر نمیخواهید Eloquent به طور خودکار این ستونها را مدیریت نماید، خصوصیت $timestamps را بر روی مدل خود با false مقداردهی کنید.در زمان ایجاد یک مدل جدید، آرایه ای از خصوصیتها را به متد سازنده آن میفرستید. این خصوصیتها با استفاده از ویژگی انتصاب کلی به مدل انتصاب داده میشوند. این کار بسیار ساده است؛ هر چند اگر بدون کنترل ورودیهای کاربر به مدل منتقل شود میتواند یک مشکل امنیتی جدی ایجاد کند. اگر ورودیهای کاربر بدون کنترل به مدل منتقل شود، کاربر میتواند تمامی یا هر کدام از خصوصیتهای مدل را تغییر دهد. به این دلیل، تمامی مدلهای Eloquent در حالت پیش فرض در مقابل انتصاب کلی محافظت شده اند. برای شروع، مقادیر خصوصیتهای fillable یا guarded را در مدل مقدار دهی نمایید.**

تعریف خصوصیات با امکان مقداردهی fillable) ) در مدل:

**خصوصیت fillable مشخص میکند کدام خصوصیتها را میتوان در انتصاب کلی مقداردهی کرد. این کار را میتوان در سطح کلاس یا نمونه کلاس انجام داد.**

**Class User extends Model {**

**protected $fillable = [**

**'first\_name',**

**'last\_name',**

**'email'**

**];**

**}**

**در این مثال، تنها سه خصوصیت لیست شده، در انتصاب کلی مقداردهی میشوند.**

تعریف خصوصیتهای محافظت شده (guarded) در مدل:

**guarded برعکس fillable است، و به عنوان یک "لیست سیاه" عمل میکند.**

**Class User extends Model {**

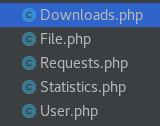
**protected $guarded = ['id', 'password'];**

**}**

**بلاک کردن تمامی خصوصیتها از انتصاب کلی:**

درمثال بالا، خصوصیتهای id و password به شکل کلی مقداردهی نمیشوند. تمامی خصوصیات دیگر میتوانند به صورت کلی انتصاب مقدار شوند. شما میتوانید تمامی خصوصیتها را با استفاده از خصوصیت guarded از انتصاب کلی محافظت نمایید:

**protected $guarded = ['\*'];**



* + 1. مدل **Downloads**

**Fillables**:

**protected $fillable = [**

**'ip',**

**'id\_file',**

**'user',**

**'token'**

**];**

* + 1. مدل **File**

**Fillables**:

**protected $fillable = [**

**'name',**

**'size',**

**'mime\_type',**

**'new\_name',**

**'IDuser',**

**'IPuser'**

**];**

* + 1. مدل **Requests**

در این مدل فیلیبل خاصی نیاز نیست.

* + 1. مدل **Statistics**

**Fillables:**

**protected $fillable=['ip'];**

* + 1. **مدل User**

**Fillables:**

**protected $fillable = [**

**'name',**

**'email',**

**'password'**

**];**

**Hidden:**

**protected $hidden = [**

**'password',**

**'remember\_token'**

**];**

* 1. **مایگریشن ها**
     1. **مایگریشن چیست؟**

Migration ها همانند سیستم های کنترل سورس کد SVN,Git ، ولی برای بانک اطلاعاتی تعبیه شده اند و به تیم کمک میکند تا تغییرات بانک اطلاعاتی را همانند سورس کد پروژه Commit کنند و اعضای تیم همیشه یک نسخه بروز از دیتابیس را برروی سیستم خود داشته باشند . Migration ها بطور کلی از Schema Builder استفاده مینمایند تا براحتی بتوانند تغییرات را بر روی ساختار (Schema) بانک اطلاعاتی اعمال و مدریرت کنند .

**ساخت Migration:**

برای ساخت یک Migration ، دستور make:migration را در Artisan CLI وارد نمایید :

**php artisan make:migration create\_users\_table**

سپس Migration شما در مسیر database/migrations تولید خواهد شد . فریموورک برای نامگذاری فایل های Migration از یک Timestamp استفاده مینماید که ترتیب اجرای آنها را مشخص میکند .

برای تعیین اینکه Migration شما بر روی کدام جدول اعمال شود ، پارامتر --table را به آخر دستور آرتیسان خود اضافه کنید . واگر Migration شما قرار است یک جدول جدید درون بانک اطلاعاتی ایجاد نماید ، --create را به پارامتر های خود اضافه نمایید :

**php artisan make:migration files --table=files**

**php artisan make:migration files --create=files**

دستور زیر تمامی Migration ها را بر روی دیتابیس اجرا میکند

**php artisan migrate**

**Schema:**

کلاس Schema لاراول روشی مستقل از پایگاه داده برای انجام تغییرات بر روی جدولها ارائه مینماید. این کلاس با تمامی پایگاه داده هایی که توسط لاراول پشتیبانی میشوند به خوبی کارمیکند، و برای تمامی این سیستم ها یک واسط توسعه یکسان دارد.

ایجاد و حذف جداول

برای ایجاد یک جدول پایگاه داده جدید، متد Schema::create استفاده می شود:

**Schema::create('users', function(Blueprint $table)}**

**$table->increments('id');**

**$table->string('name');**

**$table->string('password');**

**});**

آرگومان اولی که به متد create فرستاده می شود، اسم جدول است، و آرگومان دوم یک Closure است. این کلوژر یک شی Blueprint میپذیرد که برای تعریف جدول می توان از آن استفاده کرد.

برای تغییر نام یک جدول پایگاه داده، میتوان از متد rename استفاده کرد:

**Schema::rename($from, $to);**

برای مشخص کردن اتصال پایگاه داده مورد استفاده در انجام عملیات، از متد Schema::connection استفاده کنید:

**Schema::connection('foo')**

**->create('users', function(Blueprint $table){**

**$table->increments('id');**

**});**

برای حذف یک جدول، میتوانید از متد Schema::drop استفاده کنید:

**Schema::drop('users');**

**Schema::dropIfExists('users');**

**افزودن ستونها:**

|  |  |
| --- | --- |
| **دستور** | **تعریف** |
| **$table->increments('id');** | افزایش ID در جدول |
| **$table->string('email');** | معادل ستون VARCHAR در جدول |
| **$table->integer('size');** | معادل INTEGER در جدول |
| **$table->enum('type',['admin','user']);** | معادل ENUM در جدول |
| **$table->rememberToken();** | ستون remember\_token با نوع VARCHAR(100) و امکان NULL بودن می افزاید |
| **$table->timestamps();** | ستونهای created\_at و updated\_at را به جدول می افزاید |
| **->default('user')** | یک مقدار پیشفرض برای ستون تعیین میکند. |
| **->nullable()** | اشاره میکند که ستون امکان NULL بودن را هم دارد. |

**افزودن ایندکس**

**Schema builder چندین نوع ایندکس از چند نوع ایندکس پشتیبانی میکند. دو راه برای افزودن آنها وجود دارد. راه اول، میتوانید آن را مستقیما هنگام معرفی ستون تعریف کنید، یا در مرحله ای مجزا آنها را بیافزایید:**

**$table->string('email')->unique();**

**یا میتوانید ایندکس را در خطی مجزا بیافزایید. در ادامه لیستی از نوع ایندکسهای موجود ارائه شده اند:**

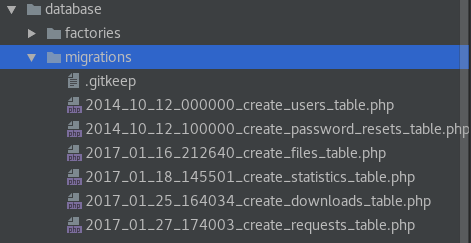
|  |  |
| --- | --- |
| **Command** | **Description** |
| **$table->unique('email');** | **افزودن ایندکس یکتا** |
| **$table->index('state');** | **افزودن ایندکس پایه ای** |

**برای به روزرسانی یک جدول، از متد Schema::table استفاده میکنیم:**

**Schema::table('users', function(Blueprint $table)}**

**$table->string('email')->unique();**

**});**



**RequestsTable:**

**public function up(){**

**Schema::create('requests', function ($table){**

**$table->increments('id');**

**$table->integer('user\_id');**

**$table->String('subject');**

**$table->timestamps();**

**});**

**}**

**public function down(){**

**Schema::dropIfExists('requests');**

**}**

**DownloadsTable:**

**public function up(){**

**Schema::create('downloads', function ($table) {**

**$table->increments('id');**

**$table->string('ip');**

**$table->integer('id\_file');**

**$table->string('user')->nullable();**

**$table->string('token');**

**$table->timestamps();**

**});**

**}**

**public function down(){**

**Schema::dropIfExists('downloads');**

**}**

**StatisticsTable:**

**public function up(){**

**Schema::create('statistics', function ($table) {**

**$table->increments('id');**

**$table->string('ip');**

**$table->timestamps();**

**});**

**}**

**public function down(){**

**Schema::dropIfExists('statistics');**

**}**

**FilesTable:**

**public function up(){**

**Schema::create('files', function ($table) {**

**$table->increments('id');**

**$table->string('name');**

**$table->integer('size');**

**$table->string('mime\_type');**

**$table->string('new\_name')->unique();**

**$table->string('IDuser')->nullable();**

**$table->string('IPuser');**

**$table->timestamps();**

**});**

**}**

**public function down(){**

**Schema::dropIfExists('files');**

**}**

**PasswordResetsTable:**

**public function up(){**

**Schema::create('password\_resets', function ($table**

**$table->string('email')->index();**

**$table->string('token')->index();**

**$table->timestamp('created\_at')->nullable();**

**});**

**}**

**public function down(){**

**Schema::drop('password\_resets');**

**}**

**UsersTable:**

**public function up(){**

**Schema::create('users', function ($table) {**

**$table->increments('id');**

**$table->string('name');**

**$table->string('email')->unique();**

**$table->integer('size')->default(10240);**

**$table->enum('type',['admin','user'])**

**->default('user');**

**$table->string('password');**

**$table->rememberToken();**

**$table->timestamps();**

**});**

**}**

**public function down(){**

**Schema::drop('users');**

**}**

* 1. کنترلر ها
     1. روتر ها در لاراول:

ما معمولا Route های پروژه را در درون فایلroutes/ web.php ذخیره میکنیم که بوسیله App\Providers\RouteServiceProvider سرویسدهی میشوند ، ساده ترین روش معرفی Route ها استفاده از URL و یک تابع (Closure) می باشد :

تعریف ساده یک Route برای درخواست GET:

**Route::get('/', function(){**

**return 'Hello World';**

**});**

درخواست های Post,Put,Delete

**Route::post('foo/bar', function(){**

**return 'Hello World';**

**});**

**Route::put('foo/bar', function(){**

**//**

**});**

**Route::delete('foo/bar', function(){**

**//**

**});**

معرفی Route برای انواع مشخصی از درخواستها

**Route::match(['get', 'post'], '/', function(){**

**return 'Hello World';**

**});**

معرفی Route برای انواع درخواست ها

**Route::any('foo', function(){**

**return 'Hello World';**

**});**

گاهی اوقات ممکن است بخواهید آدرس URL یک Route را در سایت نمایش دهید برای این منظور از تابع کمکی url استفاده میکنیم :

**$url = url('foo');**

معرفی روت resource برای 7 متد پرکاربرد:

**Route::resource('/downloads', 'DownloadsController');**

متد هایی که به کنترلر اوراید میکند به صورت زیر است:

Index که برای نمایش دادن لیستی از ریسورس هامون به کار میره

Create برای نمایش فرم برای ساخت یک ریسورس جدید به کار میره

Store که برای دریافت و پردازش اطلاعات فرم مروبط به بخشCreate به کار میرود.

Show که برای نمایش یکی از ریسورس ها به کار میرود

Edit که برای نمایش فرم برای ویرایش یکی از ریسورس های به کار میرود.

Update برای دریافت و پردازش اطلاعات فرم مربوط به بخش edit به کار میرود.

Destroy برای حذف یک ریسورس به کار میرود.

* + 1. کنترلر چیست؟

به جای تعریف تمام منطقی که درخواستها را پاسخ میدهند دریک فایل routes.php میتوانید این رفتار را با استفاده از کلاسهای کنترلر انجام دهید. کنترلرها میتوانند تمامی منطق مدیریت درخواستهای HTTP رادر یک کلاس مرتب کنند. کنترلرها میتوانند منطق مدیریت درخواستهای HTTP مرتبط را در یک کلاس دسته بندی کنند. کنترلها معمولا در دایرکتوری app/Http/Controllers ذخیره میشوند.

**کنترلرهای پایه**

درادامه مثالی از یک کلاس کنترلر ابتدایی ارائه شده است:

**<?php**

**namespace App\Http\Controllers;**

**use App\Http\Controllers\Controller;**

**class UserController extends Controller {**

**public function showProfile($id){**

**return view('user.profile',[**

**'user' => User::findOrFail($id)**

**]);**

**}**

**}**

**برای روت به اکشن متد یک کنترلر به روش زیر عمل میکنیم:**

**Route::get('user/{id}', 'UserController@showProfile');**

**نکته: تمامی کنترلرها باید کلاس کنترلر پایه را اکستند کنند.**

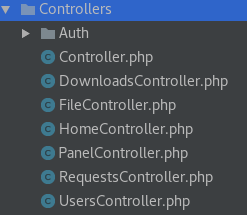
**کنترلرها و فضای نام**

**اشاره به این نکته مهم است که نیازی به ارائه کل فضای نام کنترلرنیست، تنها بخشی از نام کلاس که پس از فضای نام "ریشه" App\Http\Controllers آورده می شود. به صورت پیش فرض، RouteServiceProvider فایل routes.php را در گروه route که فضای نام کنترلر ریشه در آن قرار دارد، بارگذاری میکند.**

**اگر بخواهید کنترلرهایتان را با استفاده از namespace PHP در زیرپوشه های مجزا و دسته بندی شده در دایرکتوری App\Http\Controllers قرارگیرد، به سادگی میتوانید از نام آن کلاس نسبت به namespace ریشه App\Http\Controllers استفاده کنید. بنابراین، اگر آدرس کامل کلاس کنترلر شما App\Http\Controllers\Photos\AdminController باشد، یک route را باید به شکل زیر ایجاد کنید:**

**Route::get('foo', 'Photos\AdminController@method');**

**کنترلر های پروژه:**



**Controller:**

**این کلاس از BaseController مشتق گرفته شده است که به این معنی میباشد که متد های این کلاس در همه کنترلر ها در دست رس میباشد.**

**ما در این کلاس 1 متد کاستوم نوشتیم که در پایین کاربرد هر کدام را شرح میدهیم:**

**fileTypeFontAwesome:**

**این متد به منظور ساخت ایکن برای فایل ها بر اساس یک ارگومان ورودی که نوع فایل رو گرفته و ایکن مربوط به اون فایل رو بر میگرداند.**

**DownloadsController:**

**این کنترلر برای مدیریت دانلود های انجام شده از سمت کاربران میباشد.**

**این کنترلر دارای یک روتر میباشد به این صورت:**

**Route::resource('/downloads', 'DownloadsController');**

**ما در این کنترلر به دو متد از متد های resource نیاز داریم:**

**Index:برای نشان دادن اطلاعات لینک های ساخته شده از جمله:**

**نام فایل**

**IP**

**token**

**تاریخ ساخت**

**انقضا**

**Destroy: برای حذف توسط ایدی لینک های ساخته شده.**

**FileController:**

**این کنترلر برای مدیریت فایل های اپلود شده از سمت کاربران میباشد.**

**این کنترلر دارای یک روتر میباشد به این صورت:**

**Route::resource('/files', 'FileController');**

**ما در این کنترلر به دو متد از متد های resource نیاز داریم:**

**Index:برای نشان دادن اطلاعات لینک های ساخته شده از جمله:**

**نام فایل**

**نوع فایل**

**حجم**

**لینک دانلود فایل**

**Destroy: برای حذف توسط ایدی فایل های اپلود شده.**

**HomeController:**

**این کنترلر برای مدیریت روت های صفحه اصلی و صفحه های بیرونی سایت و هندل کردن اطلاعات در یافتی به کار میرود.**

**این کنترلر دارای 5 روتر میباشد که به صورت زیرنوشته میشود:**

**Route::get('/','HomeController@index');**

**Route::post('/upload','HomeController@uploadFiles');**

**Route::get('/download/{id}','HomeController@downloadFiles');**

**Route::post('/download/{id}','HomeController@createLink');**

**Route::get('/download/{id}/{session}','HomeController@download');**

**روتر اول برای هندل کردن صفحه اصلی به کار میرود.**

**روتر دوم برای هندل کردن درخاست های اپلود به کار میرود.**

**روتر سوم برای نمایش فایل اپلود شده بر اساس ایدی فایل به کار میرود.**

**روتر چهارم برای هندل کردن درخواست ساخت لینک به کار میرود.**

**روتر پنجم برای دانلود غیر مستقیم فایل به کار میرود.**

**در کنترلر 5 متد داریم که به صورت زیر هستند:**

**Index:**

**در این صفحه ایپی و یا نام کاربری کاربر وارد شده بررسی میگردد و اگر فایلی توسط این ایپی و یا کاربر اپلود شده بود ان ها در یک لیست نشان میدهد.**

**uploadFiles:**

**در این متد اطلاعات و فایل توسط متد پست دریافت شده و مساعل امنیتی و همچنین اطلاعات فایل پردازش شده و فایل در دیتابیس و در دیسک ذخیره میگردد.**

**downloadFiles:**

**در این صفحه که یک ارگومان ورودی توسط متد get به عنوان ای دی فایل میگیرد و اطلاعات فایل مورد نظر را از دیتابیس گرفته و برای ساخت لینک دانلود اماده میکند.**

**createLink:**

**در این متد که دو ارگومان ورودی به عنوان ایدی و اطلاعات کاربر درخواست دهنده از جمله ای پی کاربر گرفته میود و لینک دانلود غیر مستقیم توسط اطلاعات کاربر و ایدی فایل مورد نظر ساخته میشود. در خواست به این متد به صورت ایجکسی صورت میگیرد همچنین لینک دانلود ساخته شده فقط برای یک به متد محدودی معتبر میباشد.**

**Download:**

**بعد از ساخته شدن لینک دانلود توسط متد createLink لینک ساخته شده به این متد درخواست فایل را میفرستد و در این متد ما اطلاعات فایل توسط توکن دریافت شده بررسی میکنیم و در صورت معتبر بودن فایل , فایل را در اختیار درخواست دهنده دانلود فایل قرار میدهیم.**

**PanelController:**

**این کنترلر برای مدیریت روت های پنل مدیریت و صفحه های داخلی سایت به کار میرود.**

**این کنترلر دارای 3روتر میباشد که به صورت زیرنوشته میشود:**

**Route::get('/home', 'PanelController@index');**

**Route::get('/user', 'PanelController@user');**

**Route::get('/admin', 'PanelController@admin');**

**کاربر بعد از لاگین به روتر اول انتقال داده میشود و سپس در این روت تصمیم گیری میشود که کاربر عادی است و یا ادمین.بعد از تشخیص به یکی از روتر های دوم و سوم انتقال داده میشود.**

**در کنترلر 5 متد داریم که به صورت زیر هستند:**

**Index:**

**در این این متد که به روتر home متصل است هیچ کدی نوشته نمیشود زیرا عملیات تشخیص نوع کاربر در داخل متد سازنده ینی متد \_\_construct انجام میشود.**

**Admin:**

**در این متد اطلاعات و امار کلی سایت برای نمایش به مدیر اماده شده و به موتور سازنده قالب انتقال داده میشود.**

**User:**

**در این متد نیازی به اطلاعات اضافی سایت نیست و فقط ویوو را در اختیار کاربر قرار میدهیم.**

**Ip:**

**توسط این متد ما به بخش امار سایت بازدید اضافه میکنیم که ایپی کاربر را گرفته و در داخل دیتابیس جدول statistics اضافه میکند.**

**statistics:**

**این متد اطلاعات مربوط به امار سایت را از دیتابیس جمع اوری کرده و برای نمایش به مدیر اماده میکند.**

**RequestsController:**

**این کنترلر برای مدیریت درخواست های کاربران میباشد.**

**این کنترلر دارای یک روتر میباشد به این صورت:**

**Route::resource('/requests', 'RequestsController');**

**ما در این کنترلر به سه متد از متد های resource نیاز داریم:**

**Index:برای نشان دادن اطلاعات درخواست ها از جمله:**

**موضوع**

**نام کاربری**

**ایمیل**

**اولویت**

**خواندن**

**Destroy: برای حذف توسط ایدی فایل های اپلود شده.**

**Show:برای نمایش هر کدام از درخواست ها به صورت کامل.**

**UsersController:**

**این کنترلر برای مدیریت کاربران میباشد.**

**این کنترلر دارای یک روتر میباشد به این صورت:**

**Route::resource('/users', 'UsersController');**

**ما در این کنترلر به دو متد از متد های resource نیاز داریم:**

**Index:برای نشان دادن اطلاعات کاربران از جمله:**

**نام کاربری**

**ایمیل**

**حجم استفاده شده**

**حجم مانده**

**فایل های کاربر**

**Destroy: برای حذف توسط ایدی فایل های اپلود شده.**