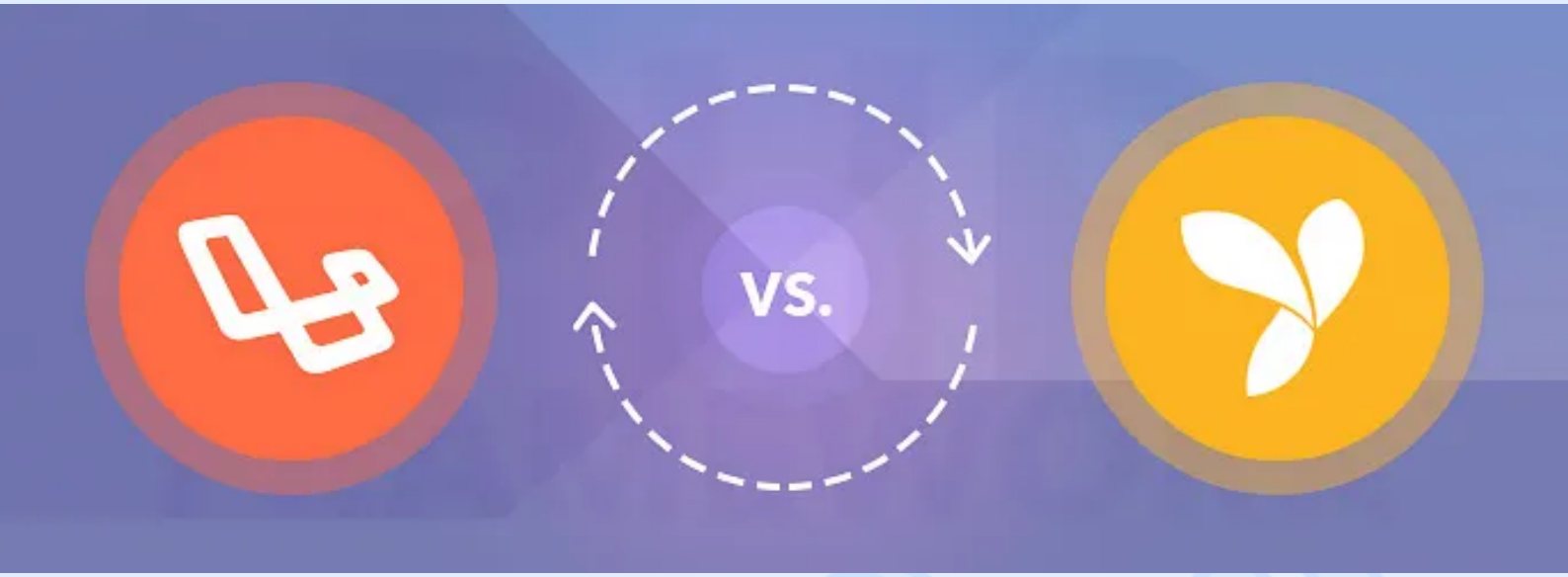

Many years ago our company was working on a big project - a Real Estate Portal. It was going to be a high-load web app, and we needed to find the best high-performance framework for it. At the time I didn't have much prior experience, I had mostly worked with CMS before. But a CMS was a bad fit for this particular project, as we were planning to use many non-standard components. Moreover, I never liked working with CMS, because it's often hard to manipulate their source code. I decided to conduct an Internet research and read many comparative posts and reviews about the cutting edge frameworks of that time. As a result, I chose Yii. The decisive factors were perfect documentation, a load of dedicated forum discussions, popularity and, therefore - low entry barrier.

I remember, as if it was yesterday, that I didn't have many difficulties and I could "google" a solution for almost any problem. Of course, my code wasn't perfect and I implemented many things roughly and improperly, but it stemmed from the fact that I lacked experience with frameworks in general. I simply had no knowledge of more elegant and proper implementation.

Now, many years later, I decided to try out a framework that is popular today - Laravel. I can tell you right away: it was uncomfortable to work with, but when it comes to dealing with something new - it's a common issue. Note, that the documentation is laconic, but I really felt like it's missing detailed descriptions of little things. Because of that, I had to go through the source code pretty often. Moreover, Laravel, unlike Yii, is rich in abstracts, so executed methods weren't easy to find. All in all, it seems like Laravel is similar to Yii in terms of usage, but implementation methods differ and it takes time to get comfortable. Something is more convenient, something - on the contrary, provokes wrath and fury. Now, let's get to our comparison.

Description

+44 20 7183 5820

info@magora.co.uk
sales@magora.co.uk

The first impression is strongly influenced by the description on the official website. Let's take a look at what these frameworks have to offer (at the moment):

Yii 2

Yii is a high-performance PHP framework that is perfect for swift development of modern web applications. Its possibilities allow implementing large scale projects like forums, portals, CMS, RESTful web services, etc.

- As most frameworks, Yii is based on MVC (Model–view–controller) pattern.
- The core of Yii philosophy is simple and elegant code.
- Yii is a full-stack framework that provides multiple turnkey solutions: a query builder, ActiveRecord for relative and NoSQL databases, RESTful API, multi-level caching support and many other things.
- Yii is very scalable. You can modify or redefine any piece of source code. You can also create your own extensions.
- The main purpose of Yii is to provide high-performance.

Laravel

Laravel is a framework designed for building web applications with exquisite and sophisticated syntax. It simplifies dealing with sore spots, such as authentication, routing, sessions and caching. Laravel was conceived to embody the best of all PHP and other frameworks (like Ruby on Rails, ASP.NET MVC and Sinatra). Laravel is accessible, yet powerful. It comes with many excellent tools for building large and reliable apps:

- Superb IoC (Inversion of control).
- Convenient migration system.
- Integrated system of module testing.

Sounds great, right? But if you draw a side-by-side comarison, you'll see that each of them has certain strengths and weaknesses. To be honest, comparing them is like comparing NetBeans IDE and

+44 20 7183 5820
info@magora.co.uk
sales@magora.co.uk

PhpStorm IDE: both have broad possibilities implemented in a different way. Until you try - you can't really understand which one works for you the best.

Yii vs Laravel

As known, in order to compare two tools you need to compare the way most common development capabilities are implemented in each of them. I really tried to take all crucial aspects into consideration:

Parameter	Yii	Laravel
Requirements	PHP 5.4 or higher	• PHP 5.4 or higher • MCrypt PHP extension • PHP JSON extension
MVC (Model–view–controller)	Yes	Yes
Extensions	Yes	Yes
ORM (Object-relational mapping)	• Data Access Objects (DAO) • Active Record Pattern (ActiveRecord) • Doctrine 2 via plugins	• DAO • ActiveRecord Pattern (Eloquent ORM) • Doctrine 2 via plugins
Testing	Out of the box: • PHPUnit • Codeception	• Out of the box PHPUnit • Symfony testing components (HttpKernel, DomCrawler and BrowserKit)

+44 20 7183 5820
info@magora.co.uk
sales@magora.co.uk

Debugging	• Comprehensive debugging console • Multi-level logging • Database requests logging	• Logging • Debugging console with a stack of callbacks Imho, pretty limited, but the problem can be solved with the *laravel-debugbar* extension.
Migration	Migration tool (Migration class)	• Migration (Migration class) • Data filling tool (Seeder class)
Security	• Feature-rich Access Control Filter (ACF) • Role-Based Access Control (RBAC) based on NIST RBAC model • OpenID, OAuth or OAuth2, etc. authorization extensions Everything off-the-shelf + plugins. Component access control is closely integrated with RBAC.	• RBAC • ACL plugins • Access Control Filters (which are really functions)
Pattern generators, working with patterns	• PHP-based • Official Twig and Smarty packages • JS, CSS, etc. connectivity assets • Plugins	• PHP-based • Blade • JS, CSS, etc. connectivity assets • Plugins • Default caching of all patterns

+44 20 7183 5820

info@magora.co.uk
sales@magora.co.uk

Caching	• APC • Database • File • Memcached • Redis • WinCache • XCache • Zend Data Cache	• APC • Database • File • Memcached (in version 5) • Redis (in version 5)
REST API	• Off-the-shelf JSON, JSONP and XML support • REST requests routing • HATEOAS support • Request caching • Speed limitations, etc.	• REST requests routing settings • JSON, JSONP support, etc.
App localization	Yes	Yes
Form validation	Yes	Yes
Scaffolding	• Gii (Yii Code Generator) - model, CRUD, controller, form, module, extension generation interface • Console	• Using a console via Artisan • Extensions

Some parts I'd like to talk more about:

Extensions

Well, both frameworks are well equipped with plenty of extensions for almost anything. It's important, because all disadvantages can be mediated with the help of one or two extensions. And they fit like a glove.

Migrations

Both frameworks offer convenient migration tools, but Laravel also enables you to add seeders for simple and smooth initial data transfer. It comes in handy for both testing (test data) and filling static directories. With Yii you only have Migrations to do that.

+44 20 7183 5820
info@magora.co.uk
sales@magora.co.uk

Security

I was, actually, surprised that Laravel doesn't have enough off-the-shelf access control tools. Nevertheless, there are many extension packages for that. I have to acknowledge that both frameworks offer instuments for working with passwords, authentication, protection from SQL injections, Cross Site Scripting (XSS), Cross Site Request Forgery (CSRF), etc.

Form validation

Both frameworks, obviously, include validation, but it's implemented differently. In Yii it's linked to the class of a form or a model. It allows you to set up rules directly in it and check it according to those rules after receiving data and filling in ActiveRecord. Let's say, there's another form that changes a portion of model data. In this case, you'll have to either create a specific class for it, or use validation scripts. It's not a problem, unless you need to validate some values directly within the controller and create your custom verification method, and not to use a ready-made solution. That's something Laravel can really be proud of. Its validator exists as a separate assisting class. You can validate data any place, any time. Plus, for convenience and unification purposes you can extend Eloquent class with your own methods.

Results

From what I've seen on multiple forums, people, who started working with Laravel and tried to master Yii later claimed that it was uncomfortable and odd to deal with. And vice versa. At the end of the day, there's no clear winner here. It's up to you to decide which one suits you best.

+44 20 7183 5820
info@magora.co.uk
sales@magora.co.uk