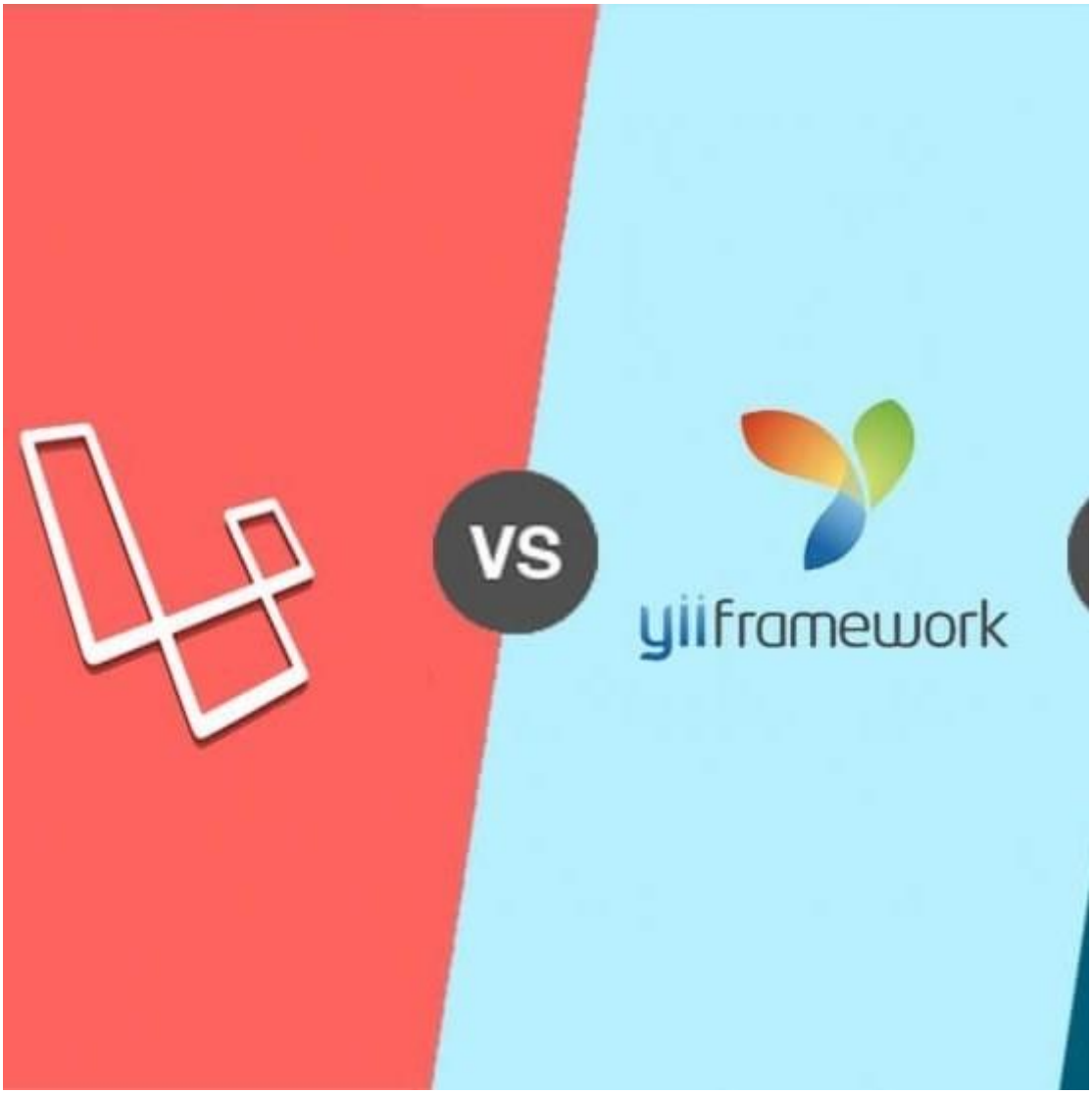
CODING WEB APPLICATION FRAMEWORKS

ONDERZOEKSVERSLAG



Inleiding

Als creatieve techneut is het belangrijk om flexibel te zijn en zul je snel moeten kunnen inspelen op veranderingen in de markt. Technische veranderingen gaan razensnel en daarmee ook de tools waarin de producten gebouwd worden. Het is dus zaak om deze tools te kennen en er in te leren werken. Een belangrijk voorbeeld hiervan zijn MVC frameworks. MVC staat voor: Model View Controller, waarin verschillende verantwoordelijkheden gescheiden zijn van elkaar. Deze frameworks hebben een aantal belangrijke basisonderdelen gemeenschappelijk welke we behandelen in de theorielessen.

Ook de keuze voor het framework is belangrijk. Bij de start van de module wordt door jou onderzoek gedaan naar het best passende framework (voor jou) en dit wordt in een verslag uitgewerkt. Daarnaast in projectbeheersing een belangrijke vaardigheid die ook in deze module wordt aangeleerd en/of verfijnd.

Vanuit school is een lijst gemaakt met bestaande MVC-framworks:

PHP: Laravel, Symfony, Phalcon, Cake, Yii**, C#:** .NET Core MVC**, JavaScript:** Sails**,** Locomotive**, Elixir:** Phoenix**, Go:** Revel**, Kotlin:** Spring**, Python:** Flask**, Rust:** Iron

Persoonlijke ervaring

Aangezien dit de tweede keer is dat ik het vak volg, en ik vorig jaar mijn praktijkopdracht probeerde te maken in C#, is mijn persoonlijke ervaring lichtelijk veranderd ten opzichte van de vorige keer dat ik dit verslag schreef.

Ik heb sinds de vorige keer dat ik dit vak volgde niet meer met MVC gewerkt, en mijn ervaring op het gebied van MVC is hierdoor niet verbeterd. Wel heb ik heel veel kennis opgedaan op het gebied van programmeren; voornamelijk JavaScript. Daarnaast heb ik ook ondervonden dat de drie keuzes die ik vorig jaar maakte nog relevanter zijn geworden dan ze al waren.

De drie frameworks die mijn voorkeur hebben te onderzoeken zijn .NET Core, Laravel en Phoenix.

.NET Core – C#

Installatie

De installatie van .NET Core is erg eenvoudig, en je kunt vrijwel meteen beginnen met je eerste project. Wel heb ik meteen de Visual Studio IDE geïnstalleerd. Deze IDE gaat bijna hand-in-hand met C#.

'Hello world!'

Om met C# te werken heb ik de Visual Studio IDE gedownload. Deze IDE heeft een groot aantal features ingebouwd in de UI. Ook is Visual Studio voorzien van een grote library aan templates en extensies, dit scheelt een hoop werk iedere keer dat je een nieuw project begint.

Om deze reden was het maken van een kleine demo erg eenvoudig; template inladen, klein tekstje aanpassen en in de browser staat: "Hello world!". Ik heb daarna nog kort gespeeld met de code en routes, door de bestaande code aan te passen.

Laravel - PHP

Installatie

Voor het installeren van Laravel heb ik een tutorial gevolgd. De stappen in de tutorial zijn via de command line. Door een composer te installeren, en via een command Laravel te installeren, een nieuwe directory aan te maken en je lokale server te starten kun je meteen beginnen met je project.

'Hello world!'

Ook voor de demo heb ik een tutorial gevolgd. Deze tutorial toont kort hoe routes worden bepaald in Laravel. Ik heb in de controller en view code geschreven die in mijn browser mijn voor- en achternaam toont als ik op mijn localhost naar /hello route.

Phoenix - Elixir

Installatie

In vergelijking met .NET en Laravel is het opzetten van een Phoenix project een stuk lastiger. Dit lag voornamelijk aan het in moeten stellen van een wachtwoord binnen de configuratie van je project. Wel is de documentatie van Elixir erg goed.

'Hello world!'

Net als bij de andere frameworks worden de view en controller apart ingeladen en vervolgens gekoppeld. Wat me opviel bij Phoenix is dat in de tutorial meteen een variable route werd toegpast, in plaats van hard-coded.

Brede verkenning

Laravel

Volwassenheid

De eerste versie van Laravel werd uitgebracht in 2011. En is sindsdien erg veranderd. Er zijn veel nieuwe versies uitgebracht, en development aan het framework is constant gebleven. In 2015 is het framework uitgeroepen tot populairste PHP-framework.

Levendigheid

Er komen continue nieuwe versies uit voor Laravel, binnen een paar jaar is het framework erg veranderd en uitgebreid. De laatste versie van het framework kwam uit in 2018 in september. Het framework wordt regelmatig verbeterd.

Kosten en hosting

Het framework heeft een MIT-licentie, hierdoor kan het gebruikt worden zonder extra kosten. Laravel kan ook gebruikt worden zonder speciale hosting.

Leercurve en ondersteuning

De documentatie van Laravel op de website is zeer uitgebreid, en volledig. Ook zijn er talloze tutorials te vinden op YouTube e.d.

Features en mogelijkheden

Op packalyst.com staat een lijst met extensies die door Laravel ondersteund worden. Het is niet moeilijk om een extensie te vinden voor Laravel.

Installatiegemak

Het installeren kan makkelijker, voornamelijk doordat gebruik wordt gemaakt van externe partijen voordat Laravel geïnstalleerd kan worden, maar het gaat snel en de guide op de website is makkelijk te volgen.

Schaalbaarheid

Volgens blogs en experts is Laravel goed schaalbaar, maar het heeft een paar complicaties.

Beveiliging

In de documentatie is veel te vinden over de verschillende mogelijkheden voor security en hoe deze is toe te voegen aan je project.

Performance

In vergelijking tot andere PHP-frameworks is Laravel een van de snelsten.

Test mogelijkheden

In de Laravel documentatie wordt testing erg uitvoerig beschreven.

Supportmogelijkheden

Er wordt nog veel door ontwikkeld aan Laravel en het framework groeit nog steeds erg snel.

.NET Core 2.0

Volwassenheid

ASP.NET bestaat al meer dan 10 jaar, en is eigenlijk zo goed als af. De nieuwere versie: .NET Core is redelijk nieuw, maar wel gebaseerd op ASP.NET. Het veschil tussen de twee is dat .NET Core crossplatform is.

Levendigheid

Er wordt hard door ontwikkeld aan dit framework en er komen constant nieuwe versies uit.

Kosten en hosting

Het framework heeft een MIT-licentie, hierdoor kan het gebruikt worden zonder extra kosten. .NET Core kan ook gebruikt worden zonder speciale hosting.

Leercurve en ondersteuning

De documentatie op de website is in mijn ogen erg rommelig, moeilijk te vinden of te lezen en vaak niet uitgelegd via voorbeelden, al is er wel echt enorm veel te vinden. Gelukkig zijn er wel veel guides op YouTube te vinden.

Features en mogelijkheden

In Visual Studio kun je extensies e.d. binnen de editor vinden, installeren, en beheren. Enorm simpel en snel.

Installatiegemak

Installeren van dit framework is erg eenvoudig en snel.

Schaalbaarheid

Het is erg schaalbaar, het gaat al meer dan 10 jaar mee en is nog steeds erg up-to-date, gewild, en populair.

Beveiliging

Er is een enorme lijst in de documentatie van .NET die alle vormen en risico's van beveiliging uitgebreid beschrijven.

Performance

Er is een grote en volledige documentatie over hoe het performance verhoogd kan worden, en guides op YouTube die hier dieper op ingaan.

Test mogelijkheden

In de .NET Core documentatie wordt veel beschreven en uitgelegd met betrekking tot testing.

Supportmogelijkheden

Er wordt nog steeds erg veel door ontwikkeld op het framework, en doordat ze al zolang meedraaien is het duidelijk dat ze weten hoe ze met verandering moeten omgaan.

Phoenix

Volwassenheid

Elixir, de taal waarin Phoenix geschreven is, is erg nieuw. Dit brengt een beetje een risico met zich mee, doordat de kans bestaat dat het nooit groot zal worden, daarnaast is het daardoor ook weer aantrekkelijk, door er vroeg bij te zijn, en het vroegtijdig te leren, kun je snel inspelen op ontwikkelingen wanneer het echt groot wordt.

Levendigheid

Het is nog heel erg in opkomst, op Stack Overflow heb je maar 500 resultaten, en op YouTube zijn niet extreem veel guides te vinden. Wel is de documentatie erg uitgebreid en volledig.

Kosten en hosting

Ook dit framework heeft een MIT-licentie. Er is ook geen speciale hosting nodig.

Leercurve en ondersteuning

Op Stack Overflow is vrijwel niets te vinden, maar op YouTube staan een paar goede tutorials. Daarnaast is de documentatie op de website erg goed en duidelijk en volledig. Wel heb ik het idee dat het lastig kan worden om hele ingewikkelde dingen te kunnen vinden.

Features en mogelijkheden

Na kort zoeken heb ik een aantal extensies en plug-ins kunnen vinden voor Phoenix.

Installatiegemak

Het installeren ging best lastig, en zou zeker makkelijker moeten worden.

Schaalbaarheid

Op blogs en op de website vertellen de developers van Phoenix hoe zei zich bewust zijn dat het belangrijk is goed up-to-date te blijven, ze vertellen hoe zei denken dit te kunnen bereiken en zich hiervoor inzetten. Het is lastig te zeggen hoe schaalbaar het framework zal blijken aangezien het extreem jong is en nog niet veel veranderingen heeft meegemaakt.

Beveiliging

In de documentatie wordt hier vrijwel niets van beschreven. Dit is een beetje zorgwekkend.

Performance

In de benchmarks op medium blijkt dat Phoenix het niet heel goed deed in vergelijking met anderen.

Test mogelijkheden

De Phoenix documentatie heeft een uitgebreide testing sectie.

Supportmogelijkheden

Het framework en ook Elixir is nog erg in ontwikkeling en erg nieuw, het is moeilijk te zeggen hoe dit er in de toekomst uit gaat zien, maar voor nu gaat het de goede kant op.

Persoonlijke invalshoek

Naar mijn persoonlijke ervaring vond ik het maken van de 'Hello world!' opdracht in Phoenix het fijnst en meest overzichtelijk. Ik ben erg geïnteresseerd in de programmeertaal en de mogelijke potentie hiervan, maar ik vind het op dit moment een iets te groot risico, tijd en moeite te steken in het leren hiervan.

.NET Core heeft een redelijke leercurve, en ik heb vorig jaar gemerkt dat ik de IDE niet fijn vind. Ook de manier van schrijven in C# vind ik niet prettig als ik het vergelijk met JavaScript en TypeScript. PHP is ook niet de fijnste manier, maar naar mijn mening wel beter dan C#. Ik heb heel veel over C# en C++ gehoord en hoe goede programmeurs in deze programmeertaal erg efficiënt applicaties en programma's kunnen schrijven. Maar ik denk niet dat in een keer instappen met een MVC-project de juiste manier is om C# te leren.

Mijn persoonlijke voorkeur voor het uitwerken van deze opdracht gaat hierdoor naar Laravel.

Commerciële invalshoek

Qua kosten zijn alle drie ongeveer gelijk. C# en zijn voorgangers C++ en C is een van de oudste programmeertalen en eingelijk de enige die nog op grote schaal gebruikt wordt. C gaat al zolang mee, dat de kans groot is dat een goede kennis van de programmeertaal op langere termijn werk zal blijven bieden. Daarnaast is PHP ook vrij oud, de vraag naar PHP-developers is groot, mede doordat programmeurs niet met PHP willen werken. Maar het blijkt wel goed te werken, dus ik denk dat de vraag naar applicaties die met PHP werken nog een tijdje zullen blijven bestaan.

Conclusie

Mijn voorkeur gaat naar Laravel, omdat mijn persoonlijke voorkeur hiernaar gaat, en het bovenaan staat bij de commerciële invalshoek. PHP-developers zijn erg gewild, en ik denk dat het een goede eigenschap zal zijn PHP-kennis te hebben.