L2.8. PHP Frameworks

# Inleiding:

Hoe meer je programmeert, hoe meer je eigenlijk een 'bibliotheek' opbouwt van handige stukken code, die je in volgende projecten weer gebruikt, zodat je niet telkens het wiel opnieuw hoeft uit te vinden. Frameworks nemen op dezelfde manier werk uit handen, en helpen je ook op andere manieren. Er bestaan veel PHP frameworks, dus het kan nodig zijn om onderzoek te doen welke het beste is voor wat jij wilt bouwen. Hierbij is het werken met Composer (PHP dependency manager) bijna onmisbaar of zelfs vereist.

Het onderzoeken van PHP frameworks bestaat uit drie fasen:

## Fase 1: Onderzoek

In de onderzoeksfase ga jij op zoek naar de volgende elementen die belangrijk zijn voor jou als programmeur:

* Wat is MVC/MVVM/MVP?

MVC: Model View controller model.

MVVM: Model View ViewModel.

MVP: Minimum Viable Product.

* Hoe past CRUD hierin?

CRUD: Create, Read, Update and Delete.

* Welke PHP frameworks kun je vinden?

Laravel & Symfony.

* Op welke manier verschillen de frameworks van elkaar?

Ze gebruiken andere template systems laravel heeft blade en symfony heeft twig.

* Welke PHP versie hebben je gevonden frameworks nodig?

Symfony gebruikt 4.1 en laravel gebruikt 7.1 of hoger.

* Wat zijn de belangrijkste redenen om een framework te gebruiken?

Als je website update veranderd er niks aan de website en het is al zo ver ontwikkeld dat alle functies die je wilt en nodig hebt er al standaard inzitten.

* Is er goede documentatie beschikbaar voor de frameworks die je gevonden hebt?

Ja er is een goede documentatie er van te vinden vind ik.

* Wat is Composer en waar wordt het voor gebruikt?

Composer deelt een standard indeling aan voor je php framework, het beheerd je bibliotheek of je bestands map waar php in zit. En je kan er mee controlleren als alles correct gaat of niet.

* Hoe kun je Composer gebruiken voor het werken met frameworks?

Om een standard indeling in te delen voor je framework.

* Wat is Yarn/NPM en wat kun je er mee?

NPM: is een pakketbeheerder net zoals composer maar NPM is niet voor PHP maar voor JavaScript.

* Wat doet encore/mix voor jou?

Zorgt voor snellere laadtijden op je websites of games die je codeert???

## Fase 2: Analyseren

In de analysefase ga jij ontdekken wat jij, met behulp van jouw onderzoek, zou willen maken. Hiervoor ga je de volgende stappen zetten:

* Pas bestaande tekeningen aan of maak een nieuwe tekening van jouw webomgeving die jij hebt gemaakt of wil maken. Beschrijf hierbij hoe jij een PHP framework wil gebruiken.  
  Tip: Kijk ook naar andere websites, misschien kun je bestaande ideeën gebruiken op jouw eigen pagina.
* Pas een bestaand plan van eisen aan of maak een nieuwe PvE waarin jij ook een PHP framework meeneemt. Houd er rekening mee dat het in jouw PvE niet 'framework' heet.
* Pas ook een bestaande planning aan of maak een nieuwe planning.
* Zoek ook eens naar mogelijke templates, v.b. AdminLTE.

## Fase 3: Realisatie

Tijdens de realisatiefase ga je zelf het programma maken. Dit houdt in, dat je de schermen maakt en ook laat werken.

* Pas een bestaande website aan of maak een nieuwe website (v.b. portfolio website).
* Vergeet niet dat alles in een GitHub repository meegenomen wordt.
* Optioneel: als jij het programma in opdracht van een klant/opdrachtgever maakt, zorg er dan voor dat de klant (en docenten) van de nieuwe opmaak op de hoogte worden gebracht en dat de klant het mogelijk nieuwe ontwerp wel goed moet keuren.

# Pitch:

Op basis van een PHP framework kan ik een website maken die het doel treft.

# Praktijkopdracht:

Maak een website met behulp van een PHP framework. Het is verstandig dit met Composer te doen, scheelt later veel tijd. Doorloop elke fase zorgvuldig.

Lever kwaliteit op techniek, inhoud en creativiteit. Vraag daarom regelmatig om feedback!

Maak het eindproduct geschikt voor het beoogde platform(en). Zorg ervoor dat je het eindproduct regelmatig test!

Zorg voor een netjes gedocumenteerd bestandsbeheer (mediamanagement).

Als je werkt voor een opdrachtgever, zorg je voor een doorlopende samenwerking met [de opdrachtgever.](https://fclive.sharepoint.com/sites/Bronnen/_layouts/15/Doc.aspx?sourcedoc=%7B7627EA31-FE67-4A55-AB54-8C8128C844D6%7D&file=Werken%20voor%20een%20opdrachtgever.docx&action=default&mobileredirect=true&CT=1593575469990&OR=ItemsView)

Als je werkt in een team, zorg je voor een doorlopende samenwerking met je team.

# Resultaat; dit laat ik zien:

* Ik kan het eindproduct netjes en op een professionele wijze opleveren
* Ik kan kwaliteit leveren op elke fase van het prototype (onderzoek, analyse, realisatie)
* Ik kan bewust keuzes maken over welke werkvorm ik toepas op techniek, inhoud en creativiteit
* Ik kan samenwerken (met opdrachtgever/in teamverband/…)
* Ik kan een realistische, bruikbare planning maken waarin doel, benodigdheden en tijd op elkaar zijn afgestemd en werk volgens deze planning
* Ik kan zelfstandig inschatten en uitzoeken hoe ik tot oplossingen kan komen
* Ik heb een brede kennis van ethiek rond software en houd mij aan regel- en wetgeving   
  (o.a. AVG, plagiaat, fact-checking, integriteit, enz.)
* Ik werk netjes: apparatuur, middelen & bestanden zijn zorgvuldig behandeld en gearchiveerd
* Mijn portfolio is up-to-date en zichtbaar voor de markt, ik ben online vindbaar als professional
* Ik kan feedback vragen en ontvangen
* Ik kan reflecteren op mijn werk, gedrag en houding

# Bewijs:

* Een website met verschillende pagina’s
* Implementatie van een PHP framework
* I.g.v. Composer: Een kloppend composer.json bestand
* Projectdocumentatie
* Geordende bestandsbeheer
* Up-to-date portfolio
* Feedback kijkers, vakgenoten en/of opdrachtgever

# Ondersteunende workshops:

* Research training
* Plannen, organiseren, bestandsbeheer
* Wet & regelgeving
* Planning maken
* PHP
* Composer
* CodeIgniter framework
* Pakkettraining
* Design thinking
* Mediamanagement

# Bronnen

* Docenten
* Lessen/workshops
* Medestudenten
* Opdrachtgevers
* Kennisbank
* LinkedIn Learning

|  |  |
| --- | --- |
| **Exameneisen Applicatieontwikkelaar tot 2020** | **Exameneisen Softwaredeveloper vanaf 2020** |
| **B1-K1: Levert een bijdrage aan het ontwikkeltraject**  B1-K1-W1: Stelt de opdracht vast  B1-K1-W2: Levert een bijdrage aan het projectplan  B1-K1-W3: Levert een bijdrage aan het ontwerp  B1-K1-W4: Bereidt de realisatie voor  **B1-K2: Realiseert en test (onderdelen van) een product**  B1-K2-W1: Realiseert (onderdelen van) een product  B1-K2-W2: Test het ontwikkelde product  **B1-K3: Levert een product op**  B1-K3-W1: Optimaliseert het product  B1-K3-W2: Levert het product op  B1-K3-W3: Evalueert het opgeleverde product  **P1-K1: Onderhoudt en beheert de applicatie**  P1-K1: Onderhoudt een applicatie  P1-K2: Beheert gegevens | **B1-K1: Realiseert software**  B1-K1-W1: Plant werkzaamheden en bewaakt de voortgang  B1-K1-W2: Ontwerpt software  B1-K1-W3: Realiseert (onderdelen van) software  B1-K1-W4: Test software  B1-K1-W5: Doet verbetervoorstellen voor de software  **B1-K2: Werkt in een ontwikkelteam**  B1-K2-W1: Voert overleg  B1-K2-W2: Presenteert het opgeleverde werk  B1-K2-W3: Reflecteert op het werk |