

Maak onderstaande toepassing zonder gebruik te maken van een MVC-framework. Je deelt je code wel op in Model, View, Controller zoals in `rest_fetch.zip`.

Je kiest zelf een onderwerp voor de toepassing. Mocht blijken dat je code letterlijk overgenomen hebt bijvoorbeeld uit een tutorial krijg je 0 op de opdracht.

Je maakt gebruik van een databank met minstens één tabel met 4 kolommen. Voorzie ook een id in elke tabel.

In de toepassing bied je (opgesplitst in model, view en controller) in PHP de volgende functionaliteit:

- (1) een POST-request
- (2) een GET-request
- (3) een PUT-request

Voorzie validatie in het model en maak gebruik van Exceptions indien de validatie faalt.

Voorzie voor het model ook de unit-tests via PHPUnit:

- (I) 3 unit-tests voor het model van (1)
- (II) 3 unit-tests voor het model van (2)
- (III) 3 unit-tests voor het model van (3)

In de toepassing bied je opgesplitst (in model, view en controller) in Javascript en HTML de volgende functionaliteit aan:

- (a) een fetch request voor (1)
- (b) een fetch request voor (2)
- (c) een fetch request voor (3)

Voorzie validatie in het model en maak gebruik van Errors indien de validatie faalt.

De combinatie (1), (I) & (a) is verplicht voor elke groep.

Verder maak je afhankelijk van het aantal groepsleden nog een aantal combinaties van 2 deelopdrachten.

Voor groepen van 2 maak je nog één extra combinatie, bijvoorbeeld (2) & (II) of (2) & (b).

Voor groepen van 3 maak je 2 bijkomende combinaties, bijvoorbeeld (2) & (II) en (3) & (c) of (2) & (3) en (c) & (d).

Voor groepen van 4 maak je 3 bijkomende combinaties.

Elke groep maakt gebruik van PHPMD om de PHP-code clean te houden (dit hoeft niet voor de PHPUnit-code).

Je gaat in groep aan de opdracht werken maar er is één eindverantwoordelijke per combinatie. Tijdens de verdediging komt iedereen aan het woord en bespreekt iedereen de code waar hij of zij verantwoordelijk voor is.

De opdracht wordt verdedigd tijdens de les van de week die begint op maandag 27 mei.