Yasar Ünlü (0996815)

PHP ADVANCED

# De web applicatie

Demo link: https://youtu.be/f0XGeEBOM2w

Voor advanced PHP opdracht heb ik een Blog website gemaakt.

De website heeft twee soorten gebruikers**:**

1. **Normale gebruiker:** Kan alleen blogs op website lezen
2. **Admin gebruiker:** Kan blogs lezen, schrijven, updaten en verwijderen.

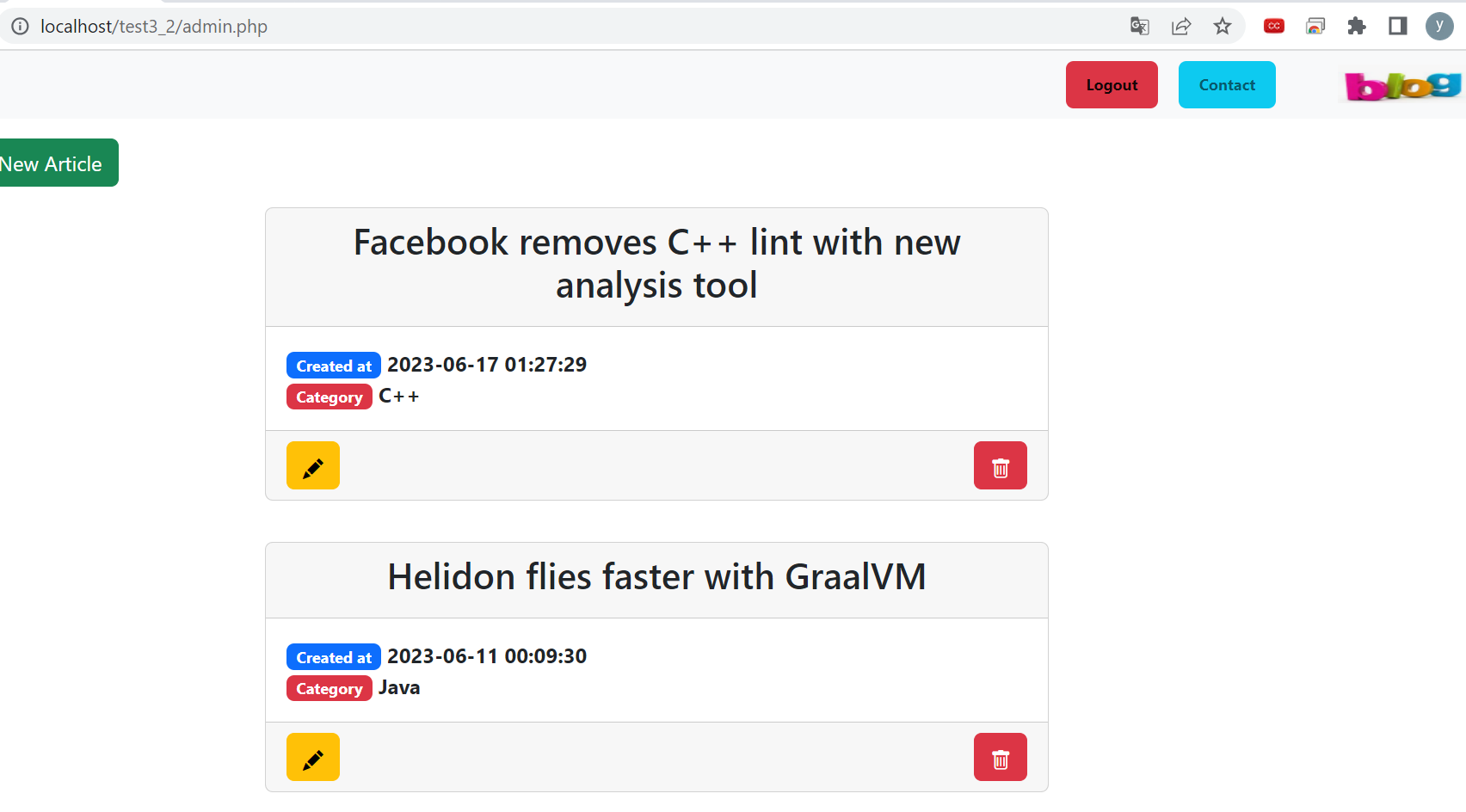
**De applicatie bestaat uit de volgende paginas:**

***Index.php***

Hier staan alle blogs.

***Admin.php***

Hier kan admin gebruiker blogs toevoegen, updaten en verwijderen



**Login.php**

Hier kan admin gebruiker inloggen

**Add.php**

Hier kan admin gebruiker nieuw blog toevoegen

**Edit.php**

Hier kan de admin blog updaten

**Contact.php**

Hier kan gewone gebruiker contact formulier invullen

# Minimaal vereiste onderdelen

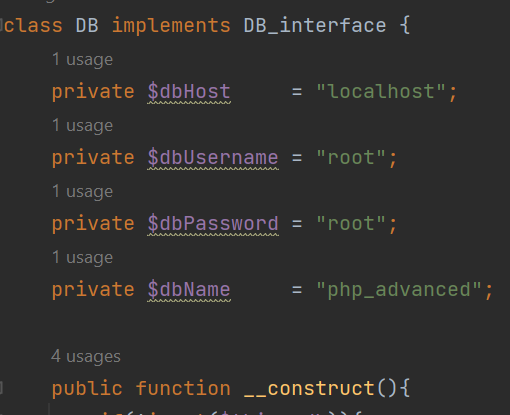
**De volgende onderdelen heb ik gebruikt in mijn applicatie:**

**Classes, Namespaces, Interfaces**

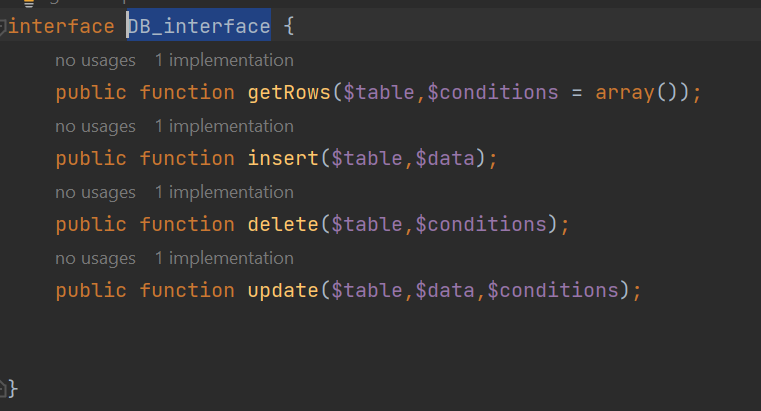
Ik heb 4 classes gebruikt:

1. DB\_class.php

Voor CRUD operations.

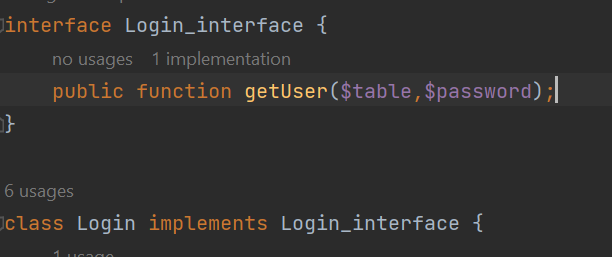


De class heft 1 interface (DB\_interface) en vier methodes:



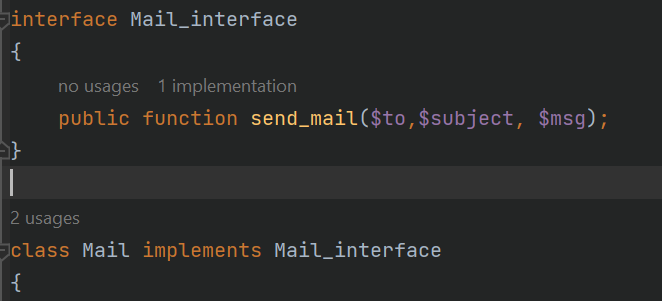
1. **Login\_class.php**

Voor admin login. De class checkt of de admin juiste inlog gegevens verstrekt. De class heeft 1 methode en 1 interface



1. **Mail\_class.php**

Voor mail sturen. De class heeft 1 methode en 1 interface



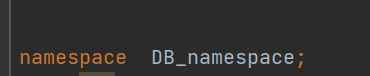
1. **Template\_engine.php**

Voor template engine. De Class heft twee methodes.

**Namespaces**

Ik heb 3 namespaces gebruikt.

**DB\_namespace** in **DB\_class.php**



**Login\_namespace** in **Login\_class.php**



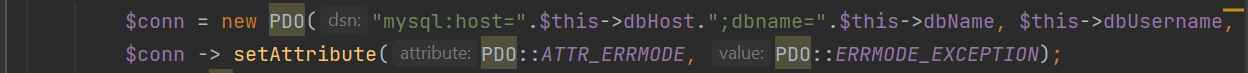
**Mail\_namespace** in **Mail\_class.php**

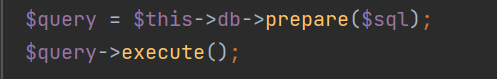


Deze classes heb ik in verschillende pagina’s gebruikt met **Use** statement.

**PDO & prepared statements**

Voor snelle en veilige database operations heb ik PDO & prepare statements gebruikt in **DB\_class.php** en **Login\_class.php**





**Autoloading via composer**

Via composer-init command heb ik een composer.json file aangemaakt. Hier heb ik geschreven welke files moeten automatisch ingeladen worden. Daarna via command *composer dump-autoload* heb ik **vendor** folder en autoloading php files aangemaakt.

Voor autoloading heb ik in elke pagina de volgende code gebruikt:



**Templateengine**

Hier heb ik een TemplateEngine class aangemaakt. De class heeft twee methodes.

assign(): Om variables te assignen.

render(): Om template page te renderen.

Hier heb ik contact\_template.php file als template gebruikt. Als de gebruiker wil contact page te gaan, wordt contact\_template.php gerenderd.

**Error logging**

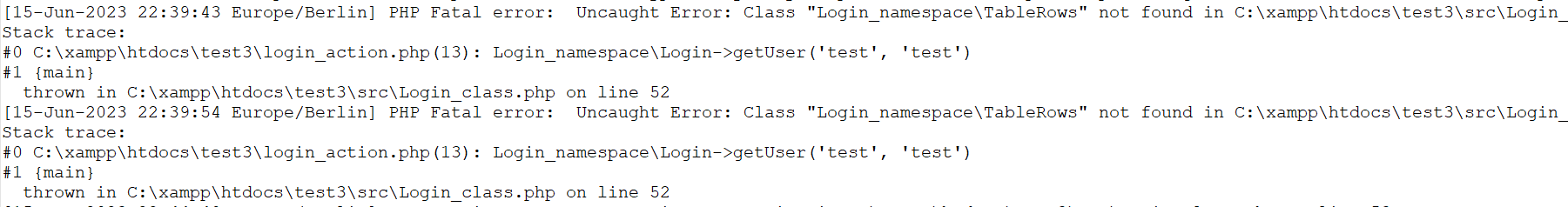
In php.ini heb ik de volgende regels aangepast:

error\_reporting = E\_ALL

display\_errors = off (zodat de gebruiker de errors niet kan zien)

error\_log="C:\xampp\php\logs\php\_error\_log" (log file path)

screenshoot van php\_error\_log file

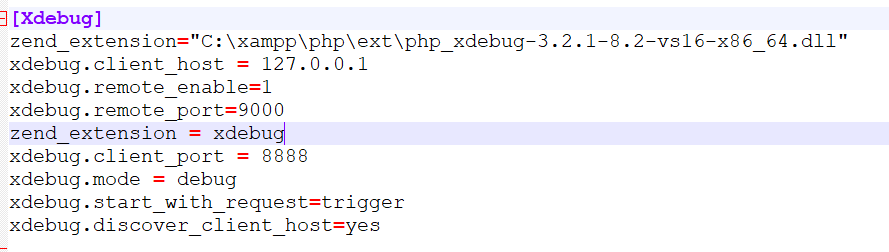


**xDebug**

Ik heb eerst Xdebug geinstalleerd from <https://xdebug.org/download>.

Daarna heb ik de php\_xdebug-3.2.1-8.2-vs16-x86\_64.dll file geinstalleerd in de path C:\xampp\php\ext.

In php.ini heb ik de volgende configuratie gedaan:



# Reflectie

Met uitzondering van databases (PDO) heb ik alle andere onderdelen op basisniveau toegepast in mijn applicatie. De volgende keer wil ik vooral de onderdelen namespaces, classes en template engines op een hoger niveau toepassen. Bijvoorbeeld, ik heb de template engine momenteel alleen toegepast op één pagina, maar ik zou deze op alle pagina's willen toepassen.