PHP ADVANCED

Yasar Ünlü (0996815)

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.

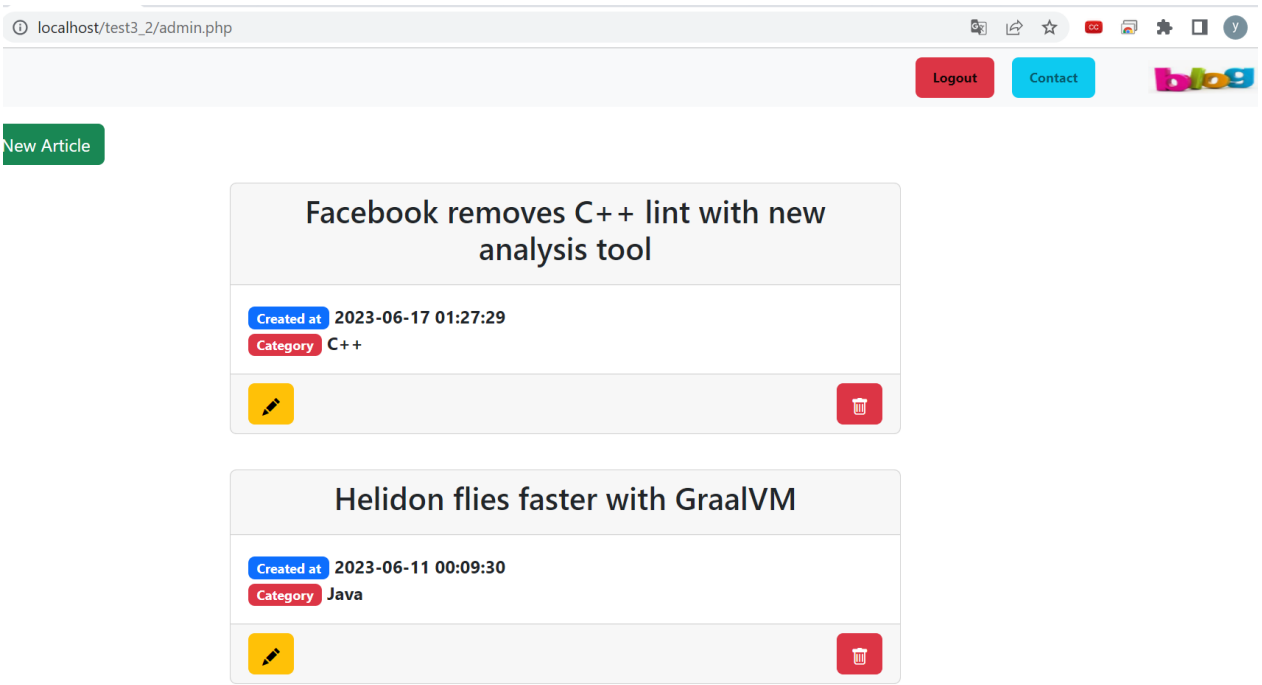
De applicatie bestaat uit de volgende paginas:

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

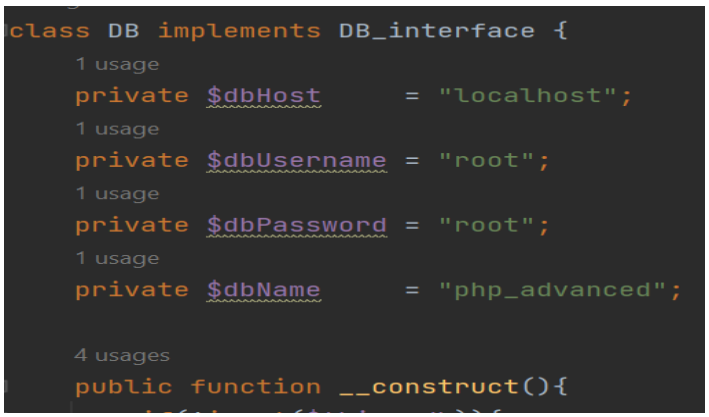
De volgende onderdelen heb ik gebruikt in mijn applicatie:

Classes, Namespaces, Interfaces

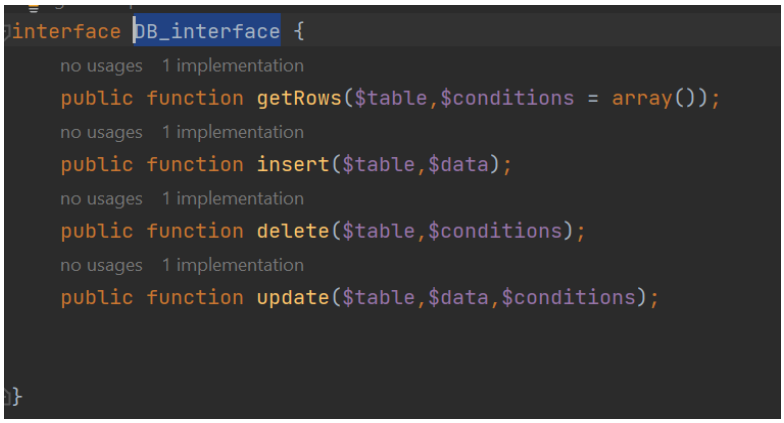
Ik heb 4 classes gebruikt:

1. DB_class.php

   Voor CRUD operations.



   De class heft 1 interface (DB_interface) en vier methodes:



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

```
interface Login_interface {
    no usages   1 implementation
    public function getUser($table,$password);
}

6 usages
class Login implements Login_interface {
```

3. **Mail_class.php**
   Voor mail sturen. De class heeft 1 methode en 1 interface

```
interface Mail_interface
{
    no usages   1 implementation
    public function send_mail($to,$subject, $msg);
}

2 usages
class Mail implements Mail_interface
{
```

4. **Template_engine.php**
   Voor template engine. De Class heft twee methodes.

Namespaces

Ik heb 3 namespaces gebruikt.

DB_namespace in **DB_class.php**

```
namespace  DB_namespace;
```

Login_namespace in **Login_class.php**

```
namespace  Login_namespace;
```

Mail_namespace in **Mail_class.php**

```
namespace Mail_namespace;
```

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**

```
$conn = new PDO( dsn: "mysql:host=".$this->dbHost.";dbname=".$this->dbName, $this->dbUsername,
$conn -> setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
```

```
$query = $this->db->prepare($sql);
$query->execute();
```

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:

```
require realpath( path: "src/Template_engine.php");
```

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

```
[15-Jun-2023 22:39:43 Europe/Berlin] PHP Fatal error:  Uncaught Error: Class "Login_namespace\TableRows" not found in C:\xampp\htdocs\test3\src\Login_
Stack trace:
#0 C:\xampp\htdocs\test3\login_action.php(13): Login_namespace\Login->getUser('test', 'test')
#1 {main}
  thrown in C:\xampp\htdocs\test3\src\Login_class.php on line 52
[15-Jun-2023 22:39:54 Europe/Berlin] PHP Fatal error:  Uncaught Error: Class "Login_namespace\TableRows" not found in C:\xampp\htdocs\test3\src\Login_
Stack trace:
#0 C:\xampp\htdocs\test3\login_action.php(13): Login_namespace\Login->getUser('test', 'test')
#1 {main}
  thrown in C:\xampp\htdocs\test3\src\Login_class.php on line 52
```

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:

```
[Xdebug]
zend_extension="C:\xampp\php\ext\php_xdebug-3.2.1-8.2-vs16-x86_64.dll"
xdebug.client_host = 127.0.0.1
xdebug.remote_enable=1
xdebug.remote_port=9000
zend_extension = xdebug
xdebug.client_port = 8888
xdebug.mode = debug
xdebug.start_with_request=trigger
xdebug.discover_client_host=yes
```

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.