

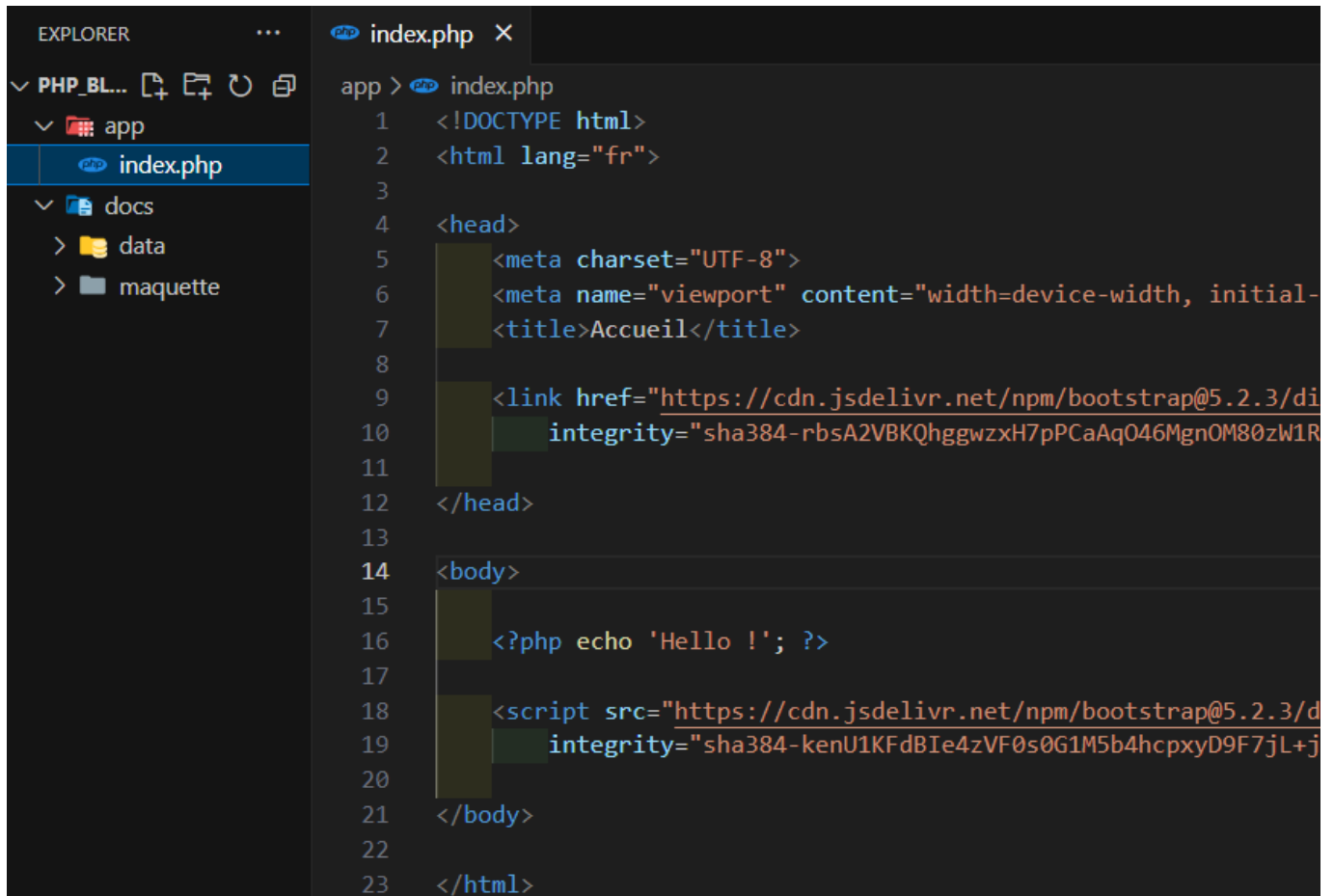
# PHP Blog Project

## 2. Premières pages en PHP

### Première structure du projet

Dans le répertoire www de WAMP, créez un dossier (php\_blog\_project) qui contiendra le projet puis ouvrez le avec VS Code.

Créez l'arborescence ci-dessous ainsi qu'une première page index.php qui intègre Bootstrap (index.php se trouve dans le répertoire app)



The screenshot shows the Visual Studio Code interface. On the left, the 'EXPLORER' sidebar displays the project structure: a folder named 'PHP\_BL...' containing a sub-folder 'app', which in turn contains 'index.php'. Below 'app' are folders 'docs', 'data', and 'maquette'. The main editor area shows the 'index.php' file with the following code:

```
1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-
7     <title>Accueil</title>
8
9     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/di
10     integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1R
11
12 </head>
13
14 <body>
15
16     <?php echo 'Hello !'; ?>
17
18     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/d
19     integrity="sha384-kenU1KFdBIE4zVF0s0G1M5b4hcpyD9F7jL+j
20
21 </body>
22
23 </html>
```

doc :

<https://www.php.net/manual/fr/language.basic-syntax.phptags.php>

<https://www.php.net/manual/en/function.echo.php>

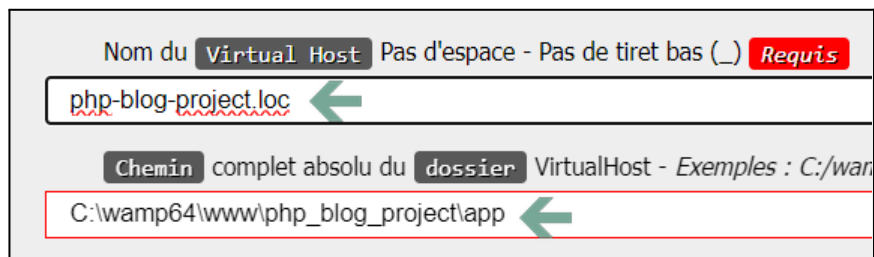
Le répertoire docs sert à stocker l'ensemble des documents relatifs au projet (maquettes, mcd, fichiers SQL, etc ...)

Le répertoire app contiendra le code de notre application (Front-Office)

**Toute la doc php officielle se trouve ici -> <https://www.php.net/manual/en/index.php>**

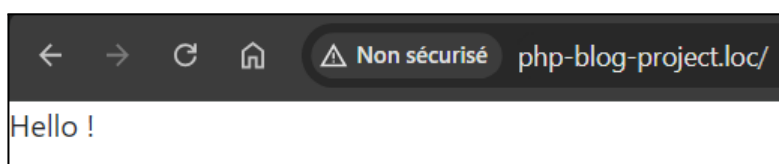
## Virtual host

Dans l'interface localhost de WAMP, créer un virtual host *php-blog-project.loc* qui pointe vers le répertoire app. Pensez à redémarrer les services (ou WAMP) ensuite.



The screenshot shows the 'Virtual Host' configuration window in WAMP. It has two main sections. The top section is for the 'Nom du Virtual Host' (Name of Virtual Host), with a label 'Pas d'espace - Pas de tiret bas (\_) Requis' (No space - No underscore - Required). The text 'php-blog-project.loc' is entered in the field, with a green arrow pointing to it. The bottom section is for the 'Chemin complet absolu du dossier VirtualHost' (Full absolute path of the VirtualHost folder), with a label 'Exemples : C:/wamp'. The text 'C:\wamp64\www\php\_blog\_project\app' is entered in the field, with a green arrow pointing to it.

Vous pouvez tester la page index.php dans le navigateur de votre choix avec l'url : <http://php-blog-project.loc>

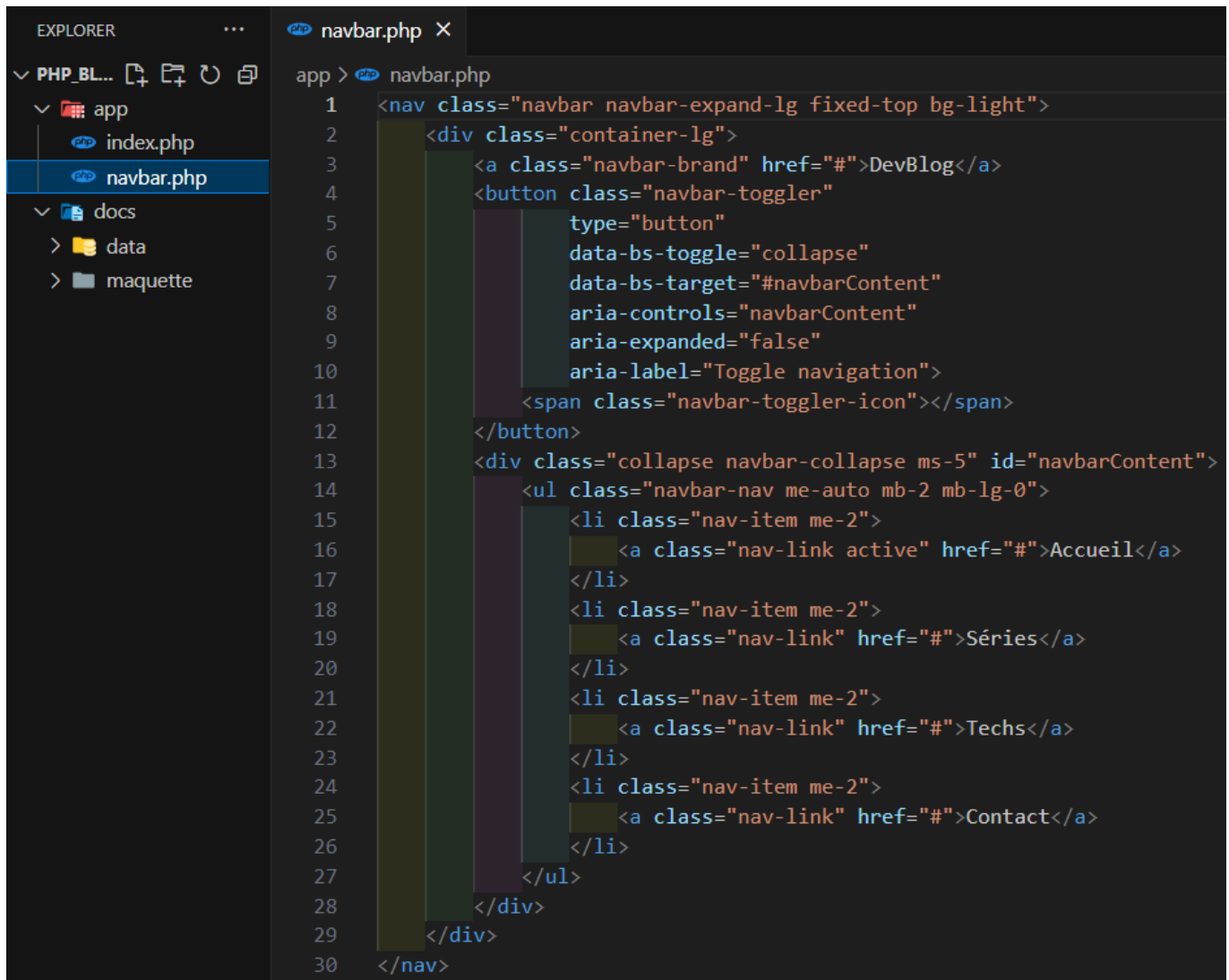


Le virtual host créé est également accessible dans l'interface localhost de WAMP



## Navbar

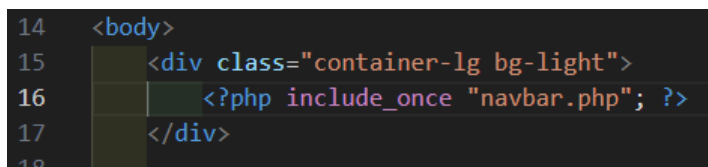
Nous créons notre navbar dans un fichier navbar.php



```
EXPLORER
  PHP_BL...
    app
      index.php
      navbar.php
    docs
    data
    maquette

app > navbar.php
1 <nav class="navbar navbar-expand-lg fixed-top bg-light">
2   <div class="container-lg">
3     <a class="navbar-brand" href="#">DevBlog</a>
4     <button class="navbar-toggler"
5       type="button"
6       data-bs-toggle="collapse"
7       data-bs-target="#navbarContent"
8       aria-controls="navbarContent"
9       aria-expanded="false"
10      aria-label="Toggle navigation">
11       <span class="navbar-toggler-icon"></span>
12     </button>
13     <div class="collapse navbar-collapse ms-5" id="navbarContent">
14       <ul class="navbar-nav me-auto mb-2 mb-lg-0">
15         <li class="nav-item me-2">
16           <a class="nav-link active" href="#">Accueil</a>
17         </li>
18         <li class="nav-item me-2">
19           <a class="nav-link" href="#">Séries</a>
20         </li>
21         <li class="nav-item me-2">
22           <a class="nav-link" href="#">Techs</a>
23         </li>
24         <li class="nav-item me-2">
25           <a class="nav-link" href="#">Contact</a>
26         </li>
27       </ul>
28     </div>
29   </div>
30 </nav>
```

Puis nous l'importons dans index.php (à la place de l'instruction echo)



```
14 <body>
15   <div class="container-lg bg-light">
16     <?php include_once "navbar.php"; ?>
17   </div>
18
```

doc :

<https://www.php.net/manual/en/function.include.php>

<https://www.php.net/manual/en/function.require.php>

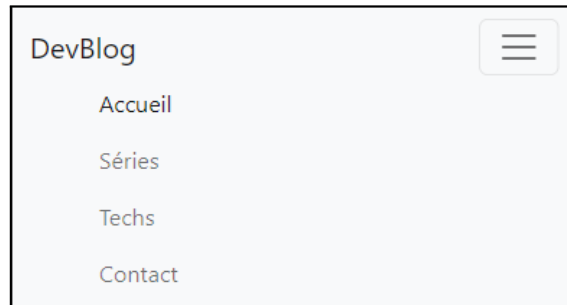
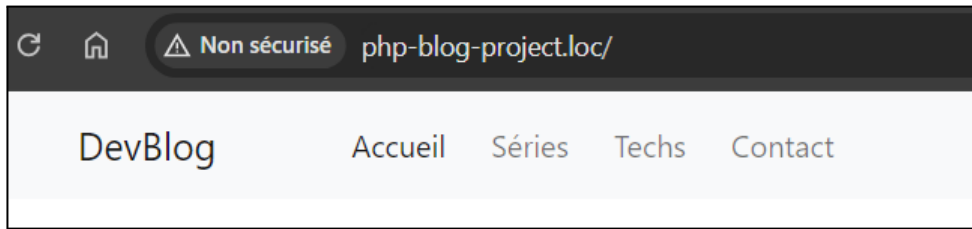
<https://www.php.net/manual/en/function.include-once.php>

<https://www.php.net/manual/en/function.require-once.php>

ressources :

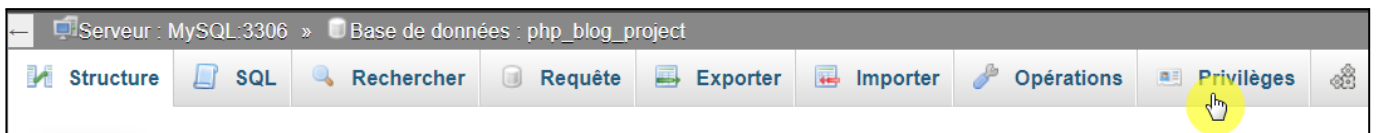
[https://www.w3schools.com/php/php\\_includes.asp](https://www.w3schools.com/php/php_includes.asp)

## Résultat

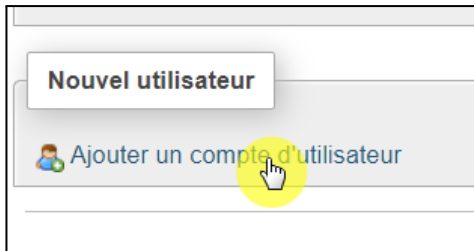


## Création d'un utilisateur en DB pour l'application (Sécurité !)

Sélectionnez l'onglet Privèges



Et cliquez sur Ajouter un compte d'utilisateur



Saisissez ensuite un nom d'utilisateur et un mot de passe (2 fois)

Mémorez ces informations (copiez les sur un bloc note)

A screenshot of the 'Ajouter un compte d'utilisateur' form in the MySQL interface. The form is titled 'Ajouter un compte d'utilisateur' and has a section 'Informations pour la connexion'. It contains the following fields:

- Nom d'utilisateur : A dropdown menu with 'Saisir une valeur' and a text input field containing 'zyQuFmFP07jJoDTT' (highlighted with a green box).
- Nom d'hôte : A dropdown menu with 'Tout hôte' and a text input field containing '%'.
- Mot de passe : A dropdown menu with 'Saisir une valeur' and two text input fields for password confirmation (both highlighted with green boxes).
- Force : A progress bar showing 'Fort'.
- Extension d'authentification : A dropdown menu with 'Authentification MySQL native'.

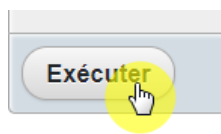
Ne cochez que la section Données

Privilèges globaux ☐ Tout cocher

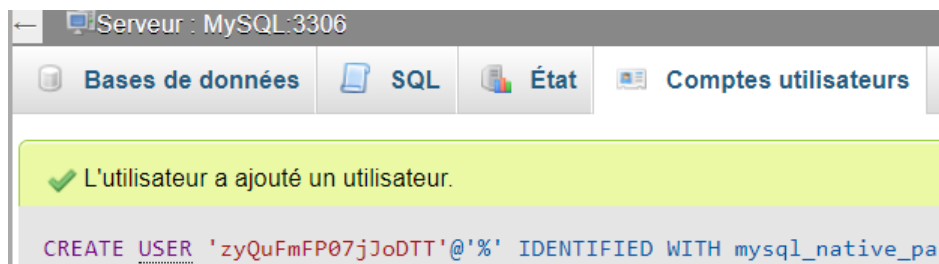
NB : les noms de privilèges sont exprimés en anglais.

<input checked="" type="checkbox"/> <b>Données</b>	<input type="checkbox"/> <b>Structure</b>
<input checked="" type="checkbox"/> SELECT	<input type="checkbox"/> CREATE
<input checked="" type="checkbox"/> INSERT	<input type="checkbox"/> ALTER
<input checked="" type="checkbox"/> UPDATE	<input type="checkbox"/> INDEX
<input checked="" type="checkbox"/> DELETE	<input type="checkbox"/> DROP
<input checked="" type="checkbox"/> FILE	<input type="checkbox"/> CREATE TEMPORARY TABLES
	<input type="checkbox"/> SHOW VIEW
	<input type="checkbox"/> CREATE ROUTINE
	<input type="checkbox"/> ALTER ROUTINE
	<input type="checkbox"/> EXECUTE
	<input type="checkbox"/> CREATE VIEW
	<input type="checkbox"/> EVENT
	<input type="checkbox"/> TRIGGER

Puis cliquez sur Exécuter



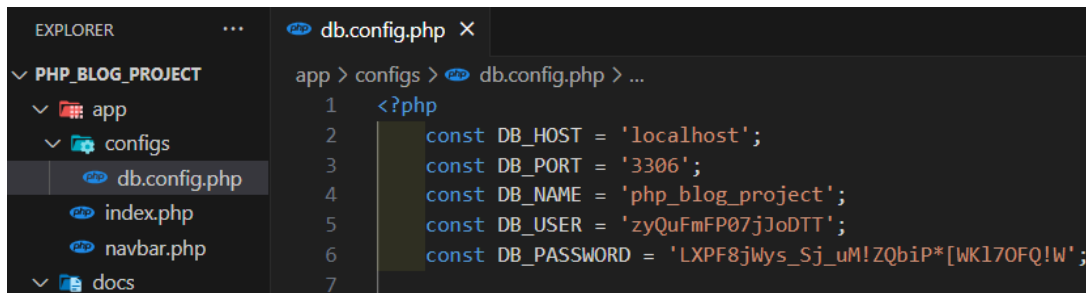
Nous avons créé le nouvel utilisateur



Notre application va se servir de cet utilisateur pour établir une connexion à la DB et interagir avec celle-ci.

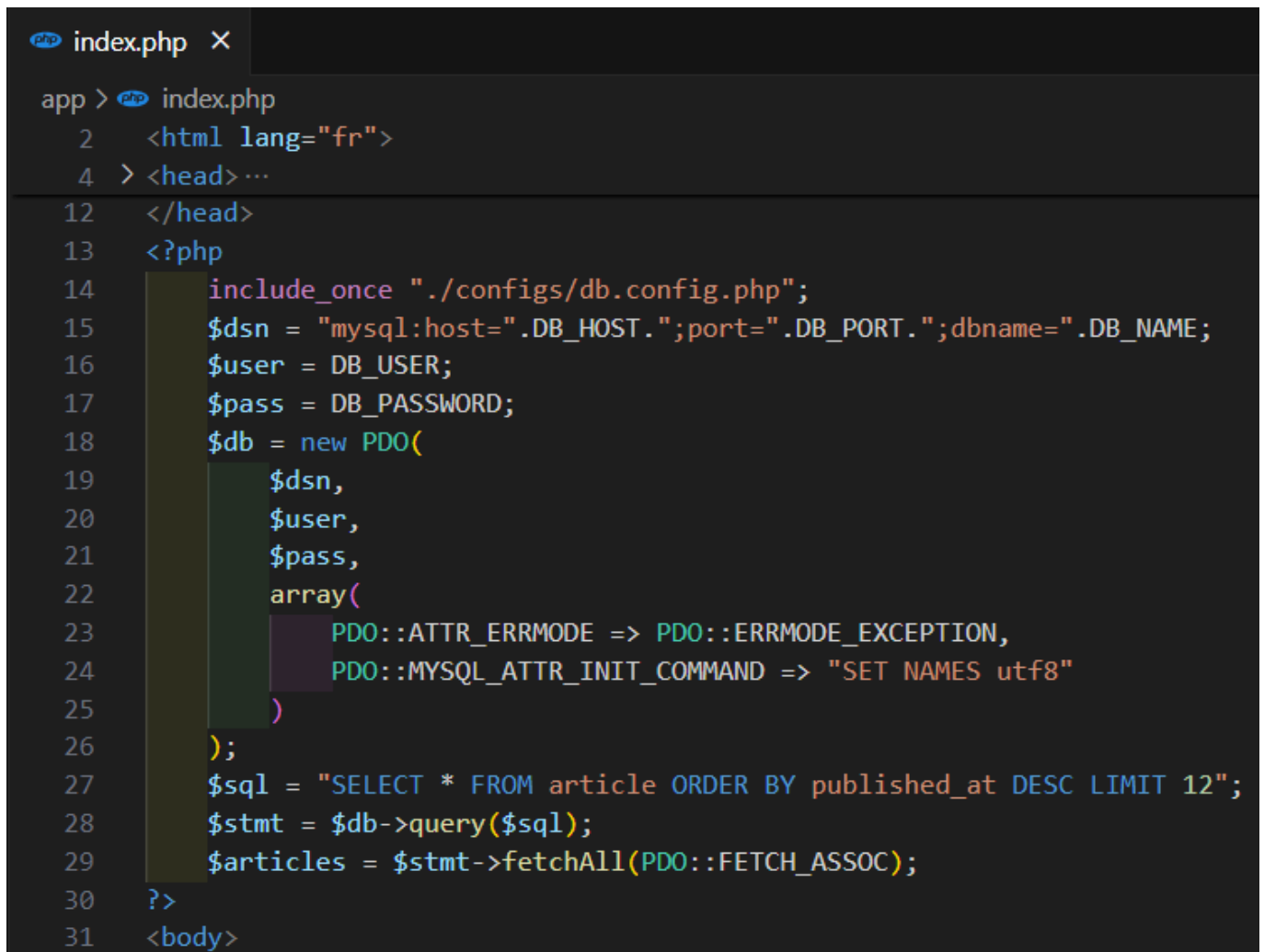
## Première connexion à la DB et première requête

Nous ajoutons un répertoire configs dans l'application et y créons un fichier *db.config.php* afin d'y stocker les informations nécessaires pour la connexion à la DB (notamment le nom d'utilisateur et mot de passe que vous avez saisi précédemment)



```
1 <?php
2     const DB_HOST = 'localhost';
3     const DB_PORT = '3306';
4     const DB_NAME = 'php_blog_project';
5     const DB_USER = 'zyQuFmFP07jJoDTT';
6     const DB_PASSWORD = 'LXPF8jWys_Sj_uM!ZQbiP*[WK170FQ!W';
7
```

Nous ajoutons le code PHP dans index.php pour établir la connexion et exécuter une première requête



```
1 <html lang="fr">
2 <head>...
3
4 </head>
5
6 <?php
7     include_once "../configs/db.config.php";
8     $dsn = "mysql:host=".DB_HOST.";port=".DB_PORT.";dbname=".DB_NAME;
9     $user = DB_USER;
10    $pass = DB_PASSWORD;
11    $db = new PDO(
12        $dsn,
13        $user,
14        $pass,
15        array(
16            PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
17            PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"
18        )
19    );
20    $sql = "SELECT * FROM article ORDER BY published_at DESC LIMIT 12";
21    $stmt = $db->query($sql);
22    $articles = $stmt->fetchAll(PDO::FETCH_ASSOC);
23    ?>
24    <body>
```

doc :

<https://www.php.net/manual/en/class.pdo.php>

<https://www.php.net/manual/en/pdo.construct.php>

<https://www.php.net/manual/en/pdo.query.php>

<https://www.php.net/manual/en/class.pdostatement.php>

<https://www.php.net/manual/en/pdostatement.fetchall.php>

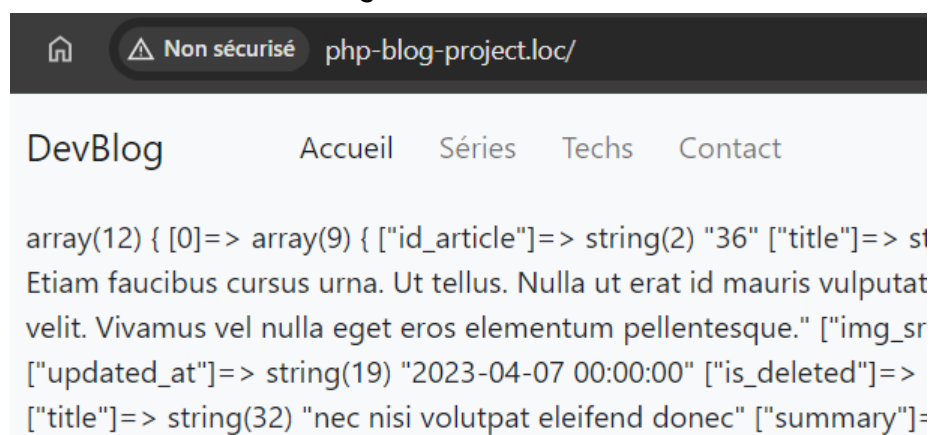
Dans un premier temps, pour visualiser les données récupérées, nous ajoutons le code suivant sous la navbar

```
<body>
  <div class="container-lg bg-light">
    <?php include_once "navbar.php"; ?>
    <main class="mt-5 pt-3">
      <?php var_dump($articles); ?>
    </main>
  </div>
```

doc :

<https://www.php.net/manual/en/function.var-dump.php>

Nous obtenons bien 12 lignes



## Mise en page des données

Plutôt que d'afficher les données brutes nous allons créer des cards Bootstrap et boucler sur le tableau \$articles. Commençons par créer une card et y afficher les données du premier article

```
$articles = $stmt->fetchAll(PDO::FETCH_ASSOC);  
$article = $articles[0];  
?  
<body>  
  <div class="container-lg bg-light">  
    <?php include_once "navbar.php"; ?>  
    <main class="mt-5 pt-3 row">  
      <div class="col-12 col-md-6 col-lg-3 d-flex align-items-stretch justify-content-center">  
        <div class="card mb-3" style="width: 18rem;">  
          ">  
          <div class="card-body text-center">  
            <h5 class="card-title"><?= $article['title']?></h5>  
            <p class="card-text" style="text-align: justify;">  
              <?= $article['summary']?>  
            </p>  
            <a href="#" class="btn btn-primary">Lire</a>  
          </div>  
        </div>  
      </div>  
    </main>
```

## Résultat





Puis ajoutons la boucle foreach

```
$articles = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>
<body>
<div class="container-lg bg-light">
  <?php include_once "navbar.php"; ?>
  <main class="mt-5 pt-3 row">
    <?php foreach($articles as $article){ ?>
      <div class="col-12 col-md-6 col-lg-3 d-flex align-items-stretch justify-content-center">
        <div class="card mb-3" style="width: 18rem;">
          ">
          <div class="card-body text-center">
            <h5 class="card-title"><?= $article['title']?></h5>
            <p class="card-text" style="text-align: justify;">
              <?= $article['summary']?>
            </p>
            <a href="#" class="btn btn-primary">Lire</a>
          </div>
        </div>
      </div>
    <?php } ?>
  </main>
</div>
```

la variable `$article = $articles[0];` déclarée précédemment n'est plus nécessaire puisque nous la déclarons à chaque tour du foreach.

doc :

<https://www.php.net/manual/en/control-structures.foreach.php>

## Résultat



git :

[https://github.com/DWWM-23526/PHP\\_BLOG\\_PROJECT/tree/Step02](https://github.com/DWWM-23526/PHP_BLOG_PROJECT/tree/Step02)