

PHP CRUD

PHP CRUD MySQL



Autor: Aramis Sarmiento Díaz

ÍNDICE

1.Introducción	3
2.Diseño e implementación	4
3.Problemas encontrados	5
4.Trabajo futuro y conclusiones	7
5.Documentación	8

1.Introducción

Este proyecto tiene como objetivo crear un blog donde se publicarán artículos de distinta índole. Las publicaciones contarán con un título, una imagen, el contenido de la noticia y la categoría de la misma. Además, se proporciona un sistema de autenticación a través de sesiones y un sistema de registro de usuarios. Para el almacenamiento de los datos, se utiliza una base de datos MySQLi. Además, se han implementado diferentes roles de usuario para permitir diferentes operaciones dependiendo del rol asignado en la base de datos.

En resumen, este proyecto combina la publicación de artículos, la autenticación y el registro de usuarios, así como la gestión de roles de usuario y el almacenamiento de datos en una base de datos, es decir implementa operaciones crud sobre todas las tablas de la base de datos.

2.Diseño e implementación

La implementación del proyecto consiste en la utilización de funciones CRUD (crear, leer, actualizar y eliminar) para la gestión de los artículos, categorías y usuarios en la base de datos. Estas funciones se han implementado mediante consultas SQL y el uso de la librería MySQLi para conectarse a la base de datos. También se ha realizado el código de las funciones de manera que sea seguro ante inyecciones SQL. Para esto se han usando consultas preparadas con MySQLi

Para garantizar la seguridad de las credenciales de la base de datos, se han usado variables de entorno mediante la dependencia composer "phpdotenv". Esta dependencia permite cargar variables de entorno desde un archivo ".env" y así evitar la exposición de información sensible en el código fuente.

Además, se ha implementado un sistema de autenticación mediante sesiones y un sistema de registro de usuarios para controlar el acceso a la aplicación. Los usuarios disponen de distintos roles que les permiten realizar distintas operaciones dentro de la aplicación, dependiendo de su rol asignado en la base de datos.

En resumen, la implementación de este proyecto consiste en la conexión a la base de datos y la gestión de los datos mediante funciones CRUD, la utilización de variables de entorno para garantizar la seguridad de las credenciales, y un sistema de autenticación y registro de usuarios con diferentes roles para controlar el acceso a la aplicación.

3.Problemas encontrados

Uno de los problemas encontrados, fue securizar el código ante las inyecciones SQL, para ello tuve que modificar el código de todas las funciones para usarlo con consultas preparadas de MySQLi, ya que había empezado a hacerlo sin tener en cuenta este requerimiento.

Otro de los problemas encontrados fue securizar la contraseña en la base de datos, para ello tuve que cifrarla mediante un hash, este aspecto no lo tuve en cuenta al empezar a implementar el proyecto por lo cual modifiqué el código de tal manera que al registrar un usuario, la contraseña se cifre con un hash y al iniciar sesión se compruebe que el hash coincide con el base de datos.

Otro problema encontrado fue el uso de las variables para conectar la aplicación a la base de datos. Al principio definí las variables sin usar las variables de entorno, las cuales dan seguridad a la aplicación, era necesario hacer esto ya que la aplicación va a ser subida a heroku y github y no nos interesa que estas variables sean visibles. Para implementar las variables de entorno hice uso del composer para el mantenimiento de las dependencias, luego instalé la dependencia “phpdotenv” para finalmente poder hacer uso de las variables de entorno.

Otro problema encontrado fue la estructura que iba a tener el proyecto. Al principio no estaba usando carpetas para organizarlo por lo cual el proyecto no estaba organizado de una manera correcta. Al final estructuré el proyecto de la siguiente manera: una carpeta public, que incluía la página index.php, que es la página principal del aplicativo, una carpeta parts: donde incluiremos la cabecera y el footer que usaremos para los demás archivos haciendo uso de la función include. Una carpeta views, dónde defino las distintas vistas que va a ver el usuario administrador ,una carpeta controllers donde se controla el cierre de sesión ,una carpeta data con los datos sql de la base de datos.

4.Trabajo futuro y conclusiones

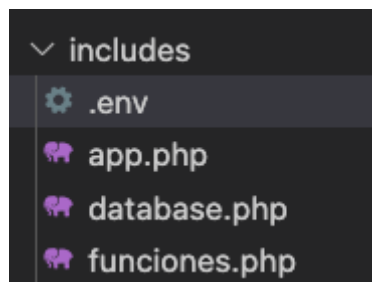
En el trabajo futuro, se pueden llevar a cabo varias mejoras en el proyecto, tales como:

1. Implementación de un sistema de búsqueda de artículos por categorías y/o títulos.
2. Implementación de un sistema de comentarios en los artículos, con la posibilidad de moderarlos antes de su publicación.
3. Mejora en el sistema de autenticación mediante la utilización de autenticación por token.
4. Implementación de un sistema de recomendación de artículos basados en los gustos y preferencias de cada usuario.

En conclusión, el proyecto que se ha implementado cumple con los requerimientos solicitados, ofreciendo una plataforma fácil de usar y con muchas posibilidades para su crecimiento en el futuro.

5.Documentación

Para acceder al proyecto y que se pueda usar correctamente, se debe tener instalado composer, además se debe ejecutar el comando `composer require vlucas/phpdotenv` en el directorio raíz del proyecto para instalar la herramienta PHP Dotenv. Al instalarla se añadirá la dependencia al archivo `composer.json`. Aquí se muestra como se usaría el dotenv y las variables de entorno y cómo quedaría el archivo `composer.json` luego de instalar las dependencias necesarias, en este caso solo dotenv. Para usar la aplicación se ejecuta el archivo `index.php` y muestra la página principal con las publicaciones, si no existe la sesión se redirige al usuario al formulario de login



```
<?php

require __DIR__ . '/../vendor/autoload.php';
$dotenv = Dotenv\Dotenv::createImmutable(__DIR__);
$dotenv->safeLoad();
require 'database.php';
require 'funciones.php';

?>
```

```

includes > database.php > ...
1  <?php
2  $db = mysqli_connect(
3      $_ENV['DB_HOST'],
4      $_ENV['DB_USER'],
5      $_ENV['DB_PASS'],
6      $_ENV['DB_DB']
7  );
8
9
10 if (!$db) {
11     echo "Error: No se pudo conectar a MySQL.";
12     echo "error de depuración: " . mysqli_connect_errno();
13     echo "error de depuración: " . mysqli_connect_error();
14     exit;
15 }
16
17 ?>

```

```

index.php — Proyecto_DSW_DPL
EXPLORADOR
PROYECTO_DSW_DPL
  router.php
  header_index.php
  header_login.php
  header.php
  public
    build
    index.php M
  vendor
  views
    auth
      login.php U
      registro.php
    gestion categorias
public > index.php > ...
1  <?php
2  ob_start();
3  session_start();
4  if (isset($_SESSION['rol'])){
5
6      $_SESSION['start'] = time();
7      $_SESSION['expire'] = $_SESSION['start'] + (40 * 60) ;
8  } else {
9      header('Location: ../views/auth/login.php');
10
11
12      die() ;
13  }
14
15  ?>

```

Enlace del vídeo:

https://drive.google.com/file/d/1w-WKO14C1w3etqiWVhbLKDyPAO8Dq4Zu/view?usp=share_link