

Paradigmas y Lenguajes de Programación III



Desafío de Halloween



Alumno: Solonezen Brian

DNI: 44652395

Profesor: Encina Agustín

Actividades:

Estimados alumnos, tenemos un desafío para hacerlo en la semana de halloween

El objetivo es crear una aplicación web simple que permita a los usuarios registrar y votar por sus disfraces de Halloween favoritos.

Pasos:

1. Configuración de la base de datos:

a. Crea una base de datos MySQL llamada "halloween" y configura una tabla llamada "disfraces" con las siguientes columnas:

```
-- -- Estructura de tabla para la tabla disfraces
CREATE TABLE disfraces ( id int(11) NOT NULL, nombre varchar(50) NOT NULL,
descripcion text NOT NULL, votos int(11) NOT NULL, foto varchar(20) NOT NULL,
foto_blob blob NOT NULL, eliminado int(11) NOT NULL DEFAULT 0 );
```

```
-- -- Estructura de tabla para la tabla usuarios
CREATE TABLE usuarios ( id int(11) NOT NULL, nombre varchar(50) NOT NULL,
clave text NOT NULL );
```

```
-- -- Estructura de tabla para la tabla votos
CREATE TABLE votos ( id int(11) NOT NULL, id_usuario int(11) NOT NULL,
id_disfraz int(11) NOT NULL );
```

2. Desarrollo de la aplicación web:

a. Crea una página principal (index.php) que muestre una lista de disfraces disponibles con sus nombres, descripciones y la cantidad de votos que han recibido.

b. Agrega un botón "Votar" junto a cada disfraz para permitir a los usuarios votar por su disfraz favorito. Debes prevenir votos duplicados de un mismo usuario.

c. Crea una página de registro (registro.php) que permita a los usuarios registrarse con un nombre de usuario y una contraseña.

d. Implementa un sistema de autenticación para asegurarte de que solo los usuarios registrados puedan votar.

Paradigmas y Lenguajes de Programación III



Desafío de Halloween



e. Crea una página de inicio de sesión (login.php) que permita a los usuarios iniciar sesión con su nombre de usuario y contraseña.

f. Desarrolla una página de administración (admin.php) que solo sea accesible para un usuario administrador. En esta página, el administrador puede gestionar (**CRUD**) los disfraces de la base de datos.

3. Personalización:

a. Añade estilos CSS para darle un toque de Halloween a tu aplicación.

b. Puedes permitir que los usuarios carguen imágenes de sus disfraces junto con la descripción.

¡Este desafío debería ser un proyecto interesante para desarrollar durante Halloween!

Asegúrate de investigar y aprender sobre autenticación, seguridad y buenas prácticas de desarrollo web en PHP y MySQL mientras lo construyes.

¡Diviértete programando!

ANOTACIONES, tenemos algunas cositas que debemos utilizar.

Ustedes deberán ir completando para que utilizamos cada una de ellas en el desarrollo del proyecto.

- ☒ `mysqli_connect()`
- ☒ `mysqli_query()`
- ☒ `mysqli_num_rows()`
- ☒ `mysqli_insert_id()`
- ☒ `mysqli_num_fields()`
- ☒ `mysqli_real_escape_string()`
- ☒ `$_FILES['foto']['name']`
- ☒ `explode(".", $archivo)`
- ☒ `end($extension)`
- ☒ `is_uploaded_file($_FILES['foto']['tmp_name'])`
- ☒ `time()`
- ☒ `copy($_FILES['foto']['tmp_name'], "fotos/" . $qu . "." . end($extension));`
- ☒ `mysqli_error($con)`
- ☒ `unlink("fotos/" . $_POST['foto_actual']);`

Paradigmas y Lenguajes de Programación III



Desafío de Halloween



- ☒ `isset($_POST['nombre'])`
- ☒ `file_exists("fotos/".$_POST['foto'])`
- ☒ `number_format($_POST['Precio'],2,',','.')`

Ustedes deberán ir analizando y listando los puntos críticos donde se imaginan que es necesario aplicar algún mecanismo de seguridad con el fin de que la aplicación sea segura.

Desarrollo

- `mysqli_connect()`: Conecta PHP con la base de datos MySQL.
- `mysqli_query()`: Ejecuta una consulta SQL.
- `mysqli_num_rows()`: Cuenta las filas devueltas por una consulta.
- `mysqli_insert_id()`: Obtiene el último ID autogenerado.
- `mysqli_num_fields()`: Devuelve cuántas columnas tiene el resultado.
- `mysqli_real_escape_string()`: Escapa caracteres peligrosos (previene inyección SQL).
- `$_FILES['foto']['name']`: Obtiene el nombre original del archivo subido.
- `explode(".", $archivo)`: Separa nombre y extensión del archivo.
- `end($extension)`: Obtiene la extensión del archivo.
- `is_uploaded_file($_FILES['foto']['tmp_name'])`: Verifica que el archivo se haya subido correctamente.
- `time()`: Genera un valor único (por ejemplo, para nombrar archivos).
- `copy($_FILES['foto']['tmp_name'], "fotos/".$_POST['foto'].end($extension))`: Copia la imagen subida a la carpeta "fotos".
- `mysqli_error($con)`: Devuelve el último error SQL.
- `unlink('fotos/' . $_POST['foto_actual'])`: Elimina un archivo del servidor.
- `isset($_POST['nombre'])`: Comprueba si existe un valor enviado desde el formulario.
- `file_exists("fotos/".$_POST['foto'])`: Verifica si el archivo existe.
- `number_format($_POST['Precio'],2,',','.')`: Da formato al precio (dos decimales).

Puntos críticos de seguridad a considerar

- Conexión a la base de datos: usar credenciales seguras y variables de entorno (.env).
- Consultas SQL: evitar inyección SQL usando `mysqli_real_escape_string()` o consultas preparadas (PDO).
- Subida de archivos: validar tipo, tamaño y extensión antes de copiar al servidor.
- Eliminación de archivos: verificar permisos y rutas para evitar eliminar archivos ajenos.
- Datos del formulario: usar `isset()` y sanitizar entradas para evitar ataques XSS o manipulación de datos.
- Errores: registrar errores con logs, sin mostrarlos al usuario final.
- Sesiones y autenticación: proteger con contraseñas cifradas y control de acceso.

Paradigmas y Lenguajes de Programación III



Desafío de Halloween

