Ejercicios de Programación PHP

Álvaro Cebrian Urueña

AVISO: Programado en xampp la contraseña de la base de datos hay que ponerla

Git: https://github.com/CebrianAlvaro9/Trabajo2

ÍNDICE

BASES DE DATOS EN PHP

Una de las principales ventajas que presenta el trabajar con programación del lado del servidor es el poder trabajar con contenidos que están alojados en bases de datos. Gracias a las bases de datos ganamos muchas ventajas a la hora de desarrollar y mantener un sitio web.

- Las bases de datos permiten el tratamiento automatizado de la información para implementar funcionalidades como las búsquedas.
- Gracias a las bases de datos podemos crear cualquier número de páginas con un solo archivo PHP. Por ejemplo, podemos tener un archivo PHP que se encarga de mostrar productos y ese mismo código podrá ser el responsable de la creación de decenas, cientos o miles de páginas de productos individuales.
- Las bases de datos relacionales, como MySQL y muchas otras, además permiten relacionar los datos de una manera sencilla, por ejemplo tener facturas que pertenecen a clientes o productos que pertenecen a facturas.
- Permiten organizar los datos, actualizarlos y gestionarlos de una manera mucho más simple, mediante formularios, sin necesitar lidiar con el código, por lo que cualquier persona puede mantener la información de las páginas.

1. Dificultades encontradas

En este trabajo teníamos que crear una aplicación que se encargará de administrar el login con una base de datos para comprobar que esas credenciales existían y eran correctas. Por otro lado tenemos que gestionar una agenda de contactos que vamos a introducir un xml con contactos y por otra parte poner unas funciones sql para introducir,buscar, actualizar y eliminar un contacto.

LOGIN MÉTODO COMPROBAR CONTRASEÑA HASH

Primera parte, El primer problema que surge es que la contraseña que se encuentra dentro de la base de datos está en código hash por lo que para comprobar si es correcta o no la clave que introduces. Para ello realizó una consulta sql la que extraigo la contraseña en código hash que está asociada al nombre que introduces una vez obtenida. Comparó la contraseña de la base de datos con la introducida con la función password verify que comprueba que el código hash corresponda a esa string. Una vez accedes dropeo la tabla de contactos en el caso de que exista para que no se produzca ningún error en caso de que existiría y la crea de nuevo.

```
public function auth(){
 $nombre= $_POST['name'];
 $sql = "select password from credenciales where us
 $hash = $this->bd->query($sql);
foreach($hash as $fila){
$contra=$fila[0];
}

  if (password_verify($_POST['password'], $contra)

    $sql = "DROP TABLE IF EXISTS
    `contactos`";
    $create = $this->bd-> prepare($sql);
    $create->execute();
    $name = $_POST['name'];
    $password = $_POST['password'];
    $sql1 ="CREATE TABLE  contactos( ".
    "tipo VARCHAR(100) NOT NULL, ".
    "nombre VARCHAR(100) NOT NULL, ".
    "apellido VARCHAR(100), ".
    "email VARCHAR(100), ".
    "direccion VARCHAR(100) NOT NULL, ".
    "numero INT NOT NULL); ";

    $create = $this->bd-> prepare($sql1);
    $create->execute();
  } else {
    header('Location: ?method=login');
```

MÉTODO PARA INTRODUCIR UN XML EN LA BASE DE DATOS

```php
public function insertarXML(){

$datos = simplexml_load_file("agenda.xml");

foreach ($datos->children() as $fila) {

    $sentencia = $this->bd-> prepare("INSERT INTO contactos (tipo,nombre,apellido,email,direccion,numero) VALUES (?,?,?,?,?,?)");
   $atributo = $fila->attributes();

    $tipo1 = $atributo['tipo'];
    $nombre = $fila->nombre;
    $apellido = $fila->apellidos;
    $email=$fila->email;
    $direccion= $fila->direccion;
    $telefono= $fila->telefono;
    $sentencia->bindParam(1,$tipo1);
    $sentencia->bindParam(2,$nombre);
    $sentencia->bindParam(3,$apellido);
    $sentencia->bindParam(4,$email);
    $sentencia->bindParam(5,$direccion);
    $sentencia->bindParam(6,$telefono);
    $sentencia->execute();
```

Para este método se utiliza la función simplexml load que carga el fichero en un array asociativo, se extraen los parámetros del array. En contrato para los atributos hay que utilizar la propiedad attributes() y luego extraer el contenido de este poniendo el nombre del atributo. Una vez los datos extraídos simplemente utilizó una sentencia sql para introducir los datos dentro de un foreach para que recorra el array generado por el xml. En cuanto a la sentencia simplemente uso bindparam para la cual hay que introducir la posición que ocupa y seguidamente lo que vas a introducir.

MÉTODO PARA SUBIR UNA IMAGEN

```php
// y con move_uploaded_file subo el fichero al carpeta destino uploads
public function subirfichero(){
 $type= $_FILES["myfile"]['type'];
  if(isset($_POST["envio"])){
    if($_FILES["myfile"]['size']<5242880){
      if($type=='image/png'||$type=='image/jpg'||$type=='application/pdf'){


        $nametemp=$_FILES["myfile"]["tmp_name"];
        $destino = 'uploads/'.$_FILES["myfile"]["name"];

       $flag= move_uploaded_file($nametemp,$destino);
       $flag ? "fichero subido correctamente" : "<br>fallo en la subida";

    }
 }else{

 }
 include('views/home.php');
 }
}
```

Para este método hay que poner un par de ifs para filtrar el tamaño y la extensión del fichero y subirlo utilizando la función move_uploaded_file(nombretemporal,carpetadestino).

En la parte del home simplemente hay que crear un formulario de tipo file para subir archivos.

MÉTODOS PARA REALIZAR CONSULTAS BÁSICAS

Para las sentencias que devuelven información se utiliza la función query() que devuelve un array con la informacion, para las demas ya sean borrar, actualizar etc hay que preparar una sentencia método prepare() y seguidamente esta sentencia hay que ejecutarlas utilizando la función execute();

PROBLEMAS EN VISTAS

Para que el usuario no puede introducir una contacto de tipo persona con email o un contacto de tipo empresa con apellido he creado la siguiente función la cual al seleccionar el tipo de contacto te mostrara un formulario especifico para ese tipo de esta forma se soluciona este problema.

```
public function seleccionar(){

  if($_POST['Tipo']=='1'){
    $mensaje= '<form action="?method=insertar" method="post">
    <br>
      <select style= "visibility: hidden;" type="hidden" name="tipo">

      <option value="empresa" type="hidden" selected>EMPRESA</option>
      </select>
      <br>
      <p> TIPO EMPRESA </p>
      <br>
      <label for="">nombre </label>
      <input type="text" name="name">
      <br>
      <label for="">email  </label>
      <input type="text" name="email">
      <br>
      <label for="">direccion </label>
      <input type="text" name="direccion">
      <br>
      <label for="">telefono  </label>
      <input type="number" name="telefono">
      <br>
      <input type="submit" value="insertar">
    </form>';

  }else{

    $mensaje='<form action="?method=insertar" method="post">
    <br>
```

```
<h3>ACTUALIZAR UN CONTACTO POR NUMERO DE TELEFONO</h3>

<form action="?method=seleccionarAc" method="post">
    <select name="tipo">
  <option value="1">EMPRESA</option>
  <option value="2" >PERSONA</option>
  <input type="submit" value="seleccionar">
  </select>
  </form>
  <?php echo $mensaje1?>
```

2. Conclusiones

Este ejercicio me ha servido para aprender a utilizar las bases de datos en php. Sobre todo la parte más interesante es la del login la de comparar las contraseñas una en código hash y la otra la que introduce el usuario me parece un gran método de seguridad.