

Memoria del Proyecto: Tienda Virtual en PHP

1. Introducción

Este proyecto es una tienda virtual desarrollada en PHP, que permite a los usuarios registrarse, iniciar sesión, comprar productos, gestionar un carrito de compras y, para el administrador, gestionar productos y usuarios. Utiliza PDO para la conexión a una base de datos MySQL y Bootstrap para la interfaz.

2. Estructura del Proyecto

- **index.php**: Lógica principal de la aplicación y vistas.
 - **conexion.php**: Conexión a la base de datos.
 - **conexion.properties**: Parámetros de conexión a la base de datos.
-

3. Conexión a la Base de Datos

La conexión se realiza mediante PDO, cargando los parámetros desde `conexion.properties` para mayor seguridad y flexibilidad.

```
1 <?php
2 //Loading database connection properties
3 $properties = parse_ini_file('conexion.properties');
4
5 //Database connection parameters
6 $servername = $properties['servername'];
7 $username = $properties['username'];
8 $password = $properties['password'];
9 $dbname = $properties['dbname'];
10
11 try{
12     //Creating a PDO connection
13     $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
14     //Set PDO error mode to exception
15     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
16     //Set charset to UTF-8
17     $conn->exec("SET NAMES utf8");
18 } catch(PDOException $e){
19     //Display error message and stop execution
20     die("Error de conexión: " . $e->getMessage());
21 }
22 ?>
```

4. Gestión de Sesiones y Carrito

Se utiliza `session_start()` para mantener la sesión del usuario y almacenar el carrito de compras en `$_SESSION['cart']`.

5. Registro de Usuarios

El registro valida que todos los campos estén completos, que el usuario y el correo no existan previamente, y almacena la contraseña de forma segura usando `password_hash`.

```

122 .....}
123 .....}
124 .....elseif($action==='admin'&&isset($_SESSION['admin']))·{
125 .....if(isset($_POST['add_product']))·{
126 .....$referencia=trim($_POST['referencia']);
127 .....$nombre=trim($_POST['nombre']);
128 .....$precio=floatval($_POST['precio']);
129 .....if(empty($referencia)||empty($nombre)||$precio<=0)·{
130 .....$error="All product fields are required and price must be positive.";
131 .....}·else·{
132 .....$stmt=$conn->prepare("SELECT·referencia·FROM·productos·WHERE·referencia=?");
133 .....$stmt->execute([$referencia]);
134 .....if($stmt->fetch())·{
135 .....$error="Product reference already exists.";
136 .....}·else·{
137 .....$stmt=$conn->prepare("INSERT·INTO·productos·(referencia,nombre,precio)·VALUES·(?,?,?)");
138 .....if($stmt->execute([$referencia,$nombre,$precio]))·{
139 .....$success="Product added successfully.";
140 .....}·else·{
141 .....$error="Failed to add product.";
142 .....}
143 .....}
144 .....}
145 .....}·elseif(isset($_POST['edit_product']))·{
146 .....$referencia=$_POST['referencia'];

```

6. Inicio de Sesión

El usuario puede iniciar sesión con su nombre de usuario y contraseña. Se verifica la contraseña usando `password_verify`.

```

108 .....try·{
109 .....foreach($_SESSION['cart']·as·$referencia=>$cantidad)·{
110 .....for($i=0;$i<$cantidad;$i++)·{
111 .....$stmt=$conn->prepare("INSERT·INTO·compras·(usuario,referencia_producto,fecha_compra)·VALUES·(?,?,NOW())");
112 .....$stmt->execute([$usuario,$referencia]);
113 .....}
114 .....}
115 .....$conn->commit();
116 .....$_SESSION['cart']=[];
117 .....$success="Purchase completed successfully!";
118 .....$action='store';

```

7. Panel de Administración

El administrador accede con credenciales fijas y puede:

- Añadir, editar y eliminar productos.
- Ver y eliminar usuarios.

```

260 .....<label·class="form-label">Name</label>
261 .....<input·type="text"·name="nombre"·class="form-control"·required>
262 .....</div>
263 .....<div·class="mb-3">
264 .....<label·class="form-label">Surname</label>
265 .....<input·type="text"·name="apellidos"·class="form-control"·required>
266 .....</div>
267 .....<div·class="mb-3">
268 .....<label·class="form-label">Email</label>
269 .....<input·type="email"·name="correo"·class="form-control"·required>
270 .....</div>
271 .....<div·class="mb-3">
272 .....<label·class="form-label">Birth·Date</label>
273 .....<input·type="date"·name="fecha_nacimiento"·class="form-control">
274 .....</div>
275 .....<div·class="mb-3">
276 .....<label·class="form-label">Gender</label>
277 .....<select·name="genero"·class="form-control"·required>
278 .....<option·value="M">Male</option>
279 .....<option·value="F">Female</option>
280 .....<option·value="Otro">Other</option>
281 .....</select>
282 .....</div>
283 .....<button·type="submit"·class="btn·btn-primary">Register</button>
284 .....</form>
285
286 .....<?php·elseif·($action===·'store'·&&·isset($_SESSION['usuario'])):·?>
287 .....<h2>Store</h2>
288 .....<div·class="row">
289 .....<?php
290 .....$stmt·=·$conn->query("SELECT·*·FROM·productos");
291 .....while·($product·=·$stmt->fetch(PDO::FETCH_ASSOC)):
292 .....?>
293 .....<div·class="col-md-4·mb-4">
294 .....<div·class="card">
295 .....<div·class="card-body">
296 .....<h5·class="card-title"><?php·echo·htmlspecialchars($product['nombre']);·?></h5>
297 .....<p·class="card-text">Price:·$<?php·echo·number_format($product['precio'],·2);·?></p>
298 .....<form·method="POST"·action=""?action=add_to_cart">
299 .....<input·type="hidden"·name="referencia"·value="<?php·echo·$product['referencia'];·?>">
300 .....<button·type="submit"·class="btn·btn-primary">Add·to·Cart</button>

```

```

315 .....<tr>
316 .....<th>Product</th>
317 .....<th>Price</th>
318 .....<th>Quantity</th>
319 .....</tr>
320 .....</thead>
321 .....<tbody>
322 .....<?php
323 .....$total:=0;
324 .....foreach($_SESSION['cart'] as $referencia=>$cantidad):
325 .....$stmt=$conn->prepare("SELECT nombre, precio FROM productos WHERE referencia=?");
326 .....$stmt->execute([$referencia]);
327 .....$product=$stmt->fetch(PDO::FETCH_ASSOC);
328 .....$subtotal=$product['precio']*$cantidad;
329 .....$total+=$subtotal;
330 .....?>
331 .....<tr>
332 .....<td><?php echo htmlspecialchars($product['nombre']); ?></td>
333 .....<td>$<?php echo number_format($product['precio'], 2); ?></td>
334 .....<td><?php echo $cantidad; ?></td>
335 .....</tr>
336 .....<?php endforeach; ?>
337 .....<tr>
338 .....<td colspan="2"><strong>Total</strong></td>
339 .....<td><strong>$<?php echo number_format($total, 2); ?></strong></td>
340 .....</tr>
341 .....</tbody>
342 .....</table>
343 .....<form method="POST" action="?action=checkout">
344 .....<button type="submit" class="btn btn-success">Complete Purchase</button>
345 .....</form>
346 .....<?php endif; ?>
347
348 .....<?php elseif($action==='admin_login'): ?>
349 .....<h2>Admin Login</h2>
350 .....<form method="POST" action="?action=admin_login">
351 .....<div class="mb-3">
352 .....<label class="form-label">Username</label>
353 .....<input type="text" name="usuario" class="form-control" required>
354 .....</div>
355 .....<div class="mb-3">
356 .....<label class="form-label">Password</label>
357 .....<input type="password" name="contrasena" class="form-control" required>
358 .....</div>
359 .....<button type="submit" class="btn btn-primary">Login</button>
360 .....</form>

```

8. Gestión de Productos

Los productos se pueden añadir, editar y eliminar desde el panel de administración. Se valida que los datos sean correctos y que la referencia no exista previamente.

```

260 .....<label class="form-label">Name</label>
261 .....<input type="text" name="nombre" class="form-control" required>
262 .....</div>
263 .....<div class="mb-3">
264 .....<label class="form-label">Surname</label>
265 .....<input type="text" name="apellidos" class="form-control" required>
266 .....</div>
267 .....<div class="mb-3">
268 .....<label class="form-label">Email</label>
269 .....<input type="email" name="correo" class="form-control" required>
270 .....</div>
271 .....<div class="mb-3">
272 .....<label class="form-label">Birth Date</label>
273 .....<input type="date" name="fecha_nacimiento" class="form-control">
274 .....</div>
275 .....<div class="mb-3">
276 .....<label class="form-label">Gender</label>
277 .....<select name="genero" class="form-control" required>
278 .....<option value="M">Male</option>
279 .....<option value="F">Female</option>
280 .....<option value="Otro">Other</option>
281 .....</select>
282 .....</div>
283 .....<button type="submit" class="btn btn-primary">Register</button>
284 .....</form>
285
286 .....<?php elseif ($action === 'store' && isset($_SESSION['usuario'])): ?>
287 .....<h2>Store</h2>
288 .....<div class="row">
289 .....<?php
290 .....$stmt = $conn->query("SELECT * FROM productos");
291 .....while ($product = $stmt->fetch(PDO::FETCH_ASSOC)):
292 .....?>
293 .....<div class="col-md-4 mb-4">
294 .....<div class="card">
295 .....<div class="card-body">
296 .....<h5 class="card-title"><?php echo htmlspecialchars($product['nombre']); ?></h5>
297 .....<p class="card-text">Price: $<?php echo number_format($product['precio'], 2); ?></p>
298 .....<form method="POST" action="?action=add_to_cart">
299 .....<input type="hidden" name="referencia" value="<?php echo $product['referencia']; ?>">
300 .....<button type="submit" class="btn btn-primary">Add to Cart</button>

```

9. Gestión de Usuarios

El administrador puede ver todos los usuarios y eliminarlos. Al eliminar un usuario, se eliminan también sus compras y datos asociados.

```

302 .....</div>
303 .....</div>
304 .....</div>
305 .....<?php endwhile;?>
306 .....</div>
307
308 .....<?php elseif ($action === 'cart' && isset($_SESSION['usuario'])):?>
309 .....<h2>Shopping Cart</h2>
310 .....<?php if (empty($_SESSION['cart'])):?>
311 .....<p>Your cart is empty.</p>
312 .....<?php else:~?>
313 .....<table class="table">
314 .....<thead>
315 .....<tr>
316 .....<th>Product</th>
317 .....<th>Price</th>
318 .....<th>Quantity</th>
319 .....</tr>
320 .....</thead>
321 .....<tbody>
322 .....<?php
323 .....$total := 0;
324 .....foreach ($_SESSION['cart'] as $referencia => $cantidad):
325 .....$stmt = $conn->prepare("SELECT nombre, precio FROM productos WHERE referencia = ?");
326 .....$stmt->execute([$referencia]);
327 .....$product = $stmt->fetch(PDO::FETCH_ASSOC);
328 .....$subtotal = $product['precio'] * $cantidad;
329 .....$total += $subtotal;
330 .....?>
331 .....<tr>
332 .....<td><?php echo htmlspecialchars($product['nombre']);?></td>
333 .....<td><?php echo number_format($product['precio'], 2);?></td>
334 .....<td><?php echo $cantidad;?></td>
335 .....</tr>
336 .....<?php endforeach;?>
337 .....<tr>
338 .....<td colspan="2"><strong>Total</strong></td>
339 .....<td><strong><?php echo number_format($total, 2);?></strong></td>
340 .....</tr>

```

10. Tienda y Carrito de Compras

Los usuarios pueden ver los productos, añadirlos al carrito y realizar compras. El carrito se almacena en la sesión y al finalizar la compra se registra en la base de datos.

```

148 .....$precio=floatval($_POST['precio']);
149 .....if (empty($nombre) || $precio <= 0) {
150 .....$error="Product name and positive price are required.";
151 .....} else {
152 .....$stmt=$conn->prepare("UPDATE productos SET nombre=?, precio=? WHERE referencia=?");
153 .....if ($stmt->execute([$nombre, $precio, $referencia])) {
154 .....$success="Product updated successfully.";
155 .....} else {
156 .....$error="Failed to update product.";
157 .....}
158 .....}
159 .....} elseif (isset($_POST['delete_product'])) {
160 .....$referencia=$_POST['referencia'];
161 .....$stmt=$conn->prepare("DELETE FROM productos WHERE referencia=?");
162 .....if ($stmt->execute([$referencia])) {
163 .....$success="Product deleted successfully.";
164 .....} else {
165 .....}
166 .....}
167 .....} elseif (isset($_POST['delete_user'])) {
168 .....$usuario=$_POST['usuario'];
169 .....$conn->beginTransaction();
170 .....try {
171 .....$stmt=$conn->prepare("DELETE FROM compras WHERE usuario=?");
172 .....$stmt->execute([$usuario]);
173 .....$stmt=$conn->prepare("DELETE FROM clientes WHERE usuario=?");
174 .....$stmt->execute([$usuario]);
175 .....$stmt=$conn->prepare("DELETE FROM usuarios WHERE usuario=?");
176 .....$stmt->execute([$usuario]);
177 .....$conn->commit();
178 .....$success="User deleted successfully.";
179 .....} catch (Exception $e) {
180 .....$conn->rollBack();

```

11. Seguridad

- Contraseñas cifradas con `password_hash`.
- Uso de sentencias preparadas para evitar inyección SQL.
- Validación de datos en formularios.