

Universidad Tecnológica de Tijuana



Carrera: Tecnologías de la Información Área desarrollo de software multiplataforma.

Grupo: 3-D.

Nombre del alumno:

Herrera Luevano Jesús Arturo

Materia: Páginas web

Nombre del profesor: Ray Brunett Parra Galaviz

Trabajo: Súper Globales \$GET Y \$POST

Fecha: 18 de octubre de 2024

Para enviar ciertos tipos de información al servidor, el protocolo HTTP provee diferentes métodos de petición. Los dos más importantes son GET y POST, los cuales, aunque entregan los mismos resultados, revelan algunas diferencias entre ellos. Lee a continuación cuáles son estas diferencias y cuándo conviene utilizar uno u otro.

Métodos de petición HTTP: GET

Con el método GET, los datos que se envían al servidor se escriben en la misma dirección URL. En la ventana del navegador, lo encontrarás así:

www.ejemplo.com/registrarse.php?nombre=pedro&apellido=perez&edad=55&genero=homubre

Toda la información introducida por el usuario (los llamados “parámetros URL”) se transmiten tan abiertamente como el URL en sí mismo. Esto tiene ventajas y desventajas.

Ventajas de GET

Los parámetros URL se pueden guardar junto a la dirección URL como marcador. De esta manera, puedes introducir una búsqueda y más tarde consultarla de nuevo fácilmente. También se puede volver a acceder a la página a través del historial del navegador.

Esto resulta práctico, por ejemplo, si visitas con asiduidad un mismo lugar en Google Maps o si guardas páginas web con configuraciones de filtro determinadas.

Desventajas de GET

La mayor desventaja del método GET es su débil protección de los datos. Los parámetros URL que se envían quedan visibles en la barra de direcciones del navegador y son accesibles sin clave en el historial de navegación, en el caché y en el log de los servidores.

Otra desventaja es que su capacidad es limitada: dependiendo del servidor y del navegador, no es posible introducir más de 2000 caracteres. Además, los parámetros URL solo pueden contener caracteres ASCII (letras, números, signos, etc.) y no datos binarios como archivos de audio o imágenes.

Métodos de petición HTTP: POST

El método POST introduce los parámetros en la solicitud HTTP para el servidor. Por ello, no quedan visibles para el usuario. Además, la capacidad del método POST es ilimitada.

Ventajas de POST

En lo relativo a los datos, como, por ejemplo, al rellenar formularios con nombres de usuario y contraseñas, el método POST ofrece mucha discreción. Los datos no se muestran en el caché ni tampoco en el historial de navegación. La flexibilidad del método POST también resulta muy útil: no solo se pueden enviar textos cortos, sino también otros tipos de información, como fotos o vídeos.

Desventajas de POST

Cuando una página web que contiene un formulario se actualiza (por ejemplo, cuando se retrocede a la página anterior) los datos del formulario deben transferirse de nuevo (puede que alguna vez hayas recibido una de estas advertencias). Por este motivo, existe el riesgo de que los datos se envíen varias veces por error, lo que, en el caso de una tienda online, puede dar lugar a pedidos duplicados. No obstante, las webs modernas de las tiendas suelen estar preparadas para evitar este tipo de problemas.

Además, los datos transferidos con el método POST no pueden guardarse junto al URL como marcador.

FUENTE

- Equipo editorial de IONOS. (2020, 11 agosto). GET vs. POST: los dos métodos de petición HTTP más conocidos cara a cara. IONOS Digital Guide. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/get-vs-post/>

Universidad Tecnológica de Tijuana



Carrera: Tecnologías de la Información Área desarrollo de software multiplataforma.

Grupo: 3-D.

Nombre del alumno:

Herrera Luevano Jesús Arturo

Materia: Páginas web

Nombre del profesor: Ray Brunett Parra Galaviz

Trabajo: PHP y MySQL

Fecha: 19 de octubre de 2024

Hay dos métodos para conectar con PHP a una base de datos MySQL: MySQLi y PDO.

MySQLi son las siglas de MySQL Improved. Es una extensión exclusiva de MySQL que agrega nuevas funciones a la interfaz de una base de datos MySQL. MySQLi es tanto procedimental como orientada a objetos, siendo el primer atributo heredado de la versión anterior de MySQL.

El MySQL original divide una tarea en procedimientos lineales, paso a paso, lo que dificulta la modificación porque tienes que editar el código desde la parte superior. Mientras tanto, MySQLi ve los datos como un conjunto de objetos intercambiables con funciones, lo que permite a los usuarios agregar o eliminar datos fácilmente.

PDO son las siglas de PHP Data Object. A diferencia de MySQLi, PDO solo está orientado a objetos y admite varios tipos de bases de datos diferentes que usan PHP, como MySQL, MSSQL, Informix y PostgreSQL.

Cualquiera que sea el método que utilices, necesitarás la información correcta para poder conectarte a la base de datos MySQL que has creado. Aquí es donde los detalles de la base de datos MySQL que has guardado previamente son útiles.

También necesitas el nombre de servidor o nombre de host correcto para la configuración. Hostinger usa “localhost” como el nombre de host de su servidor MySQL. En general, este es el nombre que querrás usar si cargaste tu secuencia de comandos PHP en el mismo servidor que la base de datos.

Uso de MySQLi para conectar un script PHP a MySQL

```
<?php

$servername = "localhost";

$dbname = "databasename";

$username = "username";

$password = "password";

// Create connection

$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check connection

if (!$conn) {

    die("Connection failed: " . mysqli_connect_error());

}

echo "Connected successfully";

mysqli_close($conn);

?>
```

Usar PDO para conectar un script PHP a MySQL

En public_html, crea un archivo llamado pdoconfig.php e inserta el siguiente código. Como siempre, no olvides reemplazar los valores del marcador de posición con la información de tu base de datos. Guárdalo y ciérralo una vez que hayas terminado.

```
<?php

$host = 'localhost';

$dbname = 'databasename';

$username = 'username';

$password = 'password';
```

Crea otro archivo llamado databaseconnect.php en el mismo directorio, pero con el siguiente código. Si nombraste el archivo anterior de manera diferente, asegúrese de cambiar el valor de require_once.

```
<?php

require_once 'pdoconfig.php';

try {

    $conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);

    echo "Connected to $dbname at $host successfully.";

} catch (PDOException $pe) {

    die("Could not connect to the database $dbname :". $pe->getMessage());

}
```


Referencia

- B, G., & B, G. (2023, 6 mayo). *Cómo conectar PHP con Bases de datos MySQL*. Tutoriales Hostinger. <https://www.hostinger.mx/tutoriales/conectar-php-mysql>

Universidad Tecnológica de Tijuana



Carrera: Tecnologías de la Información Área desarrollo de software multiplataforma.

Grupo: 3-D.

Nombre del alumno:

Herrera Luevano Jesús Arturo

Materia: Páginas web

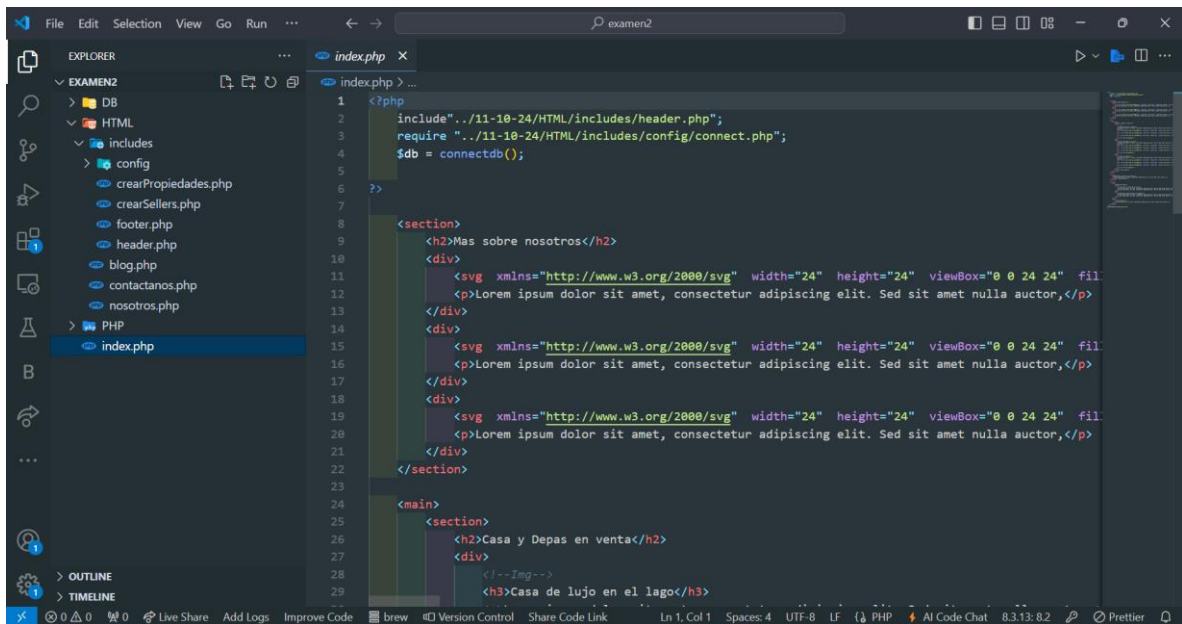
Nombre del profesor: Ray Brunett Parra Galaviz

Trabajo: Pagina web de la clase

Fecha: 31 de octubre del 2024

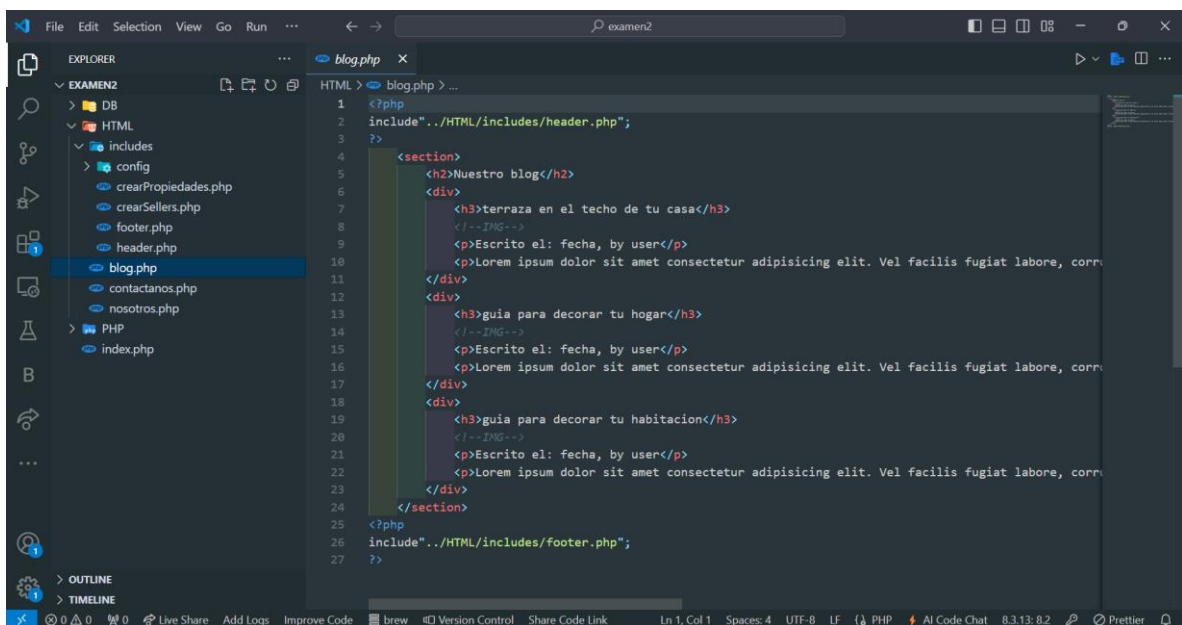
En este apartado solo se despliega el header de la página, ya que todas las pantallas comparten los mismos datos, y este se creó para su reutilización

En este apartado solo se despliega el footer de la página, ya que todas las pantallas comparten los mismos datos, y este se creó para su reutilización y evitarnos reescribir código



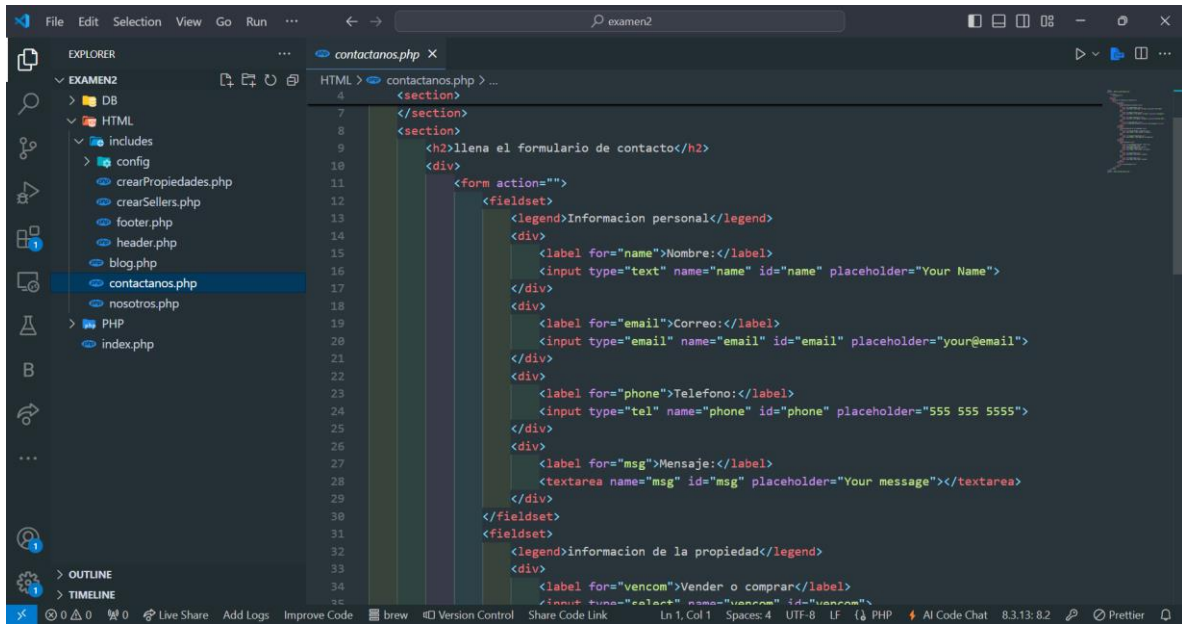
```
1 <?php
2 include "../11-10-24/HTML/includes/header.php";
3 require "../11-10-24/HTML/includes/config/connect.php";
4 $db = connectdb();
5
6 ?>
7
8 <section>
9 <h2>Mas sobre nosotros</h2>
10 <div>
11 <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="black" stroke-width="2"></svg>
12 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sit amet nulla auctor,</p>
13 </div>
14 <div>
15 <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="black" stroke-width="2"></svg>
16 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sit amet nulla auctor,</p>
17 </div>
18 <div>
19 <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="black" stroke-width="2"></svg>
20 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sit amet nulla auctor,</p>
21 </div>
22 </section>
23
24 <main>
25 <section>
26 <h2>Casa y Depas en venta</h2>
27 <div>
28 <!--img-->
29 <h3>Casa de lujo en el lago</h3>
```

Aquí en el index se puede observar que se utilizo el header y después de incluirlo con php se agregan secciones para esta pagina que nos permiten ver la información y datos que contiene este sitio web



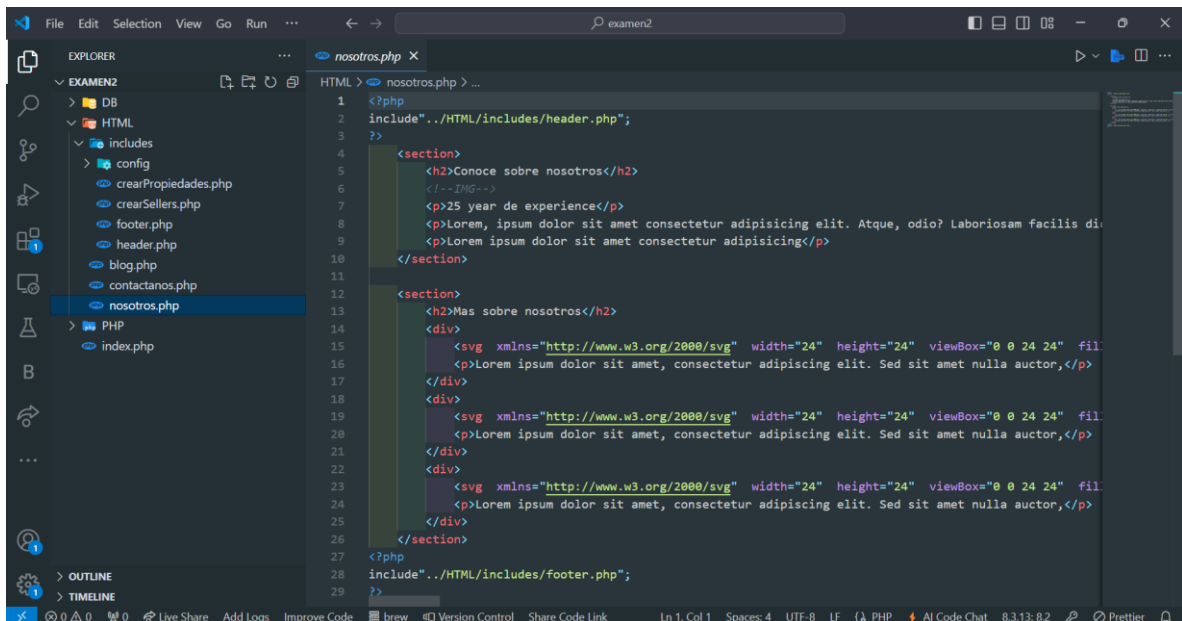
```
1 <?php
2 include "../HTML/includes/header.php";
3 ?>
4
5 <section>
6 <h2>Nuestro blog</h2>
7 <div>
8 <h3>terraza en el techo de tu casa</h3>
9 <!--img-->
10 <p>Escrito el: fecha, by user</p>
11 <p>Lorem ipsum dolor sit amet consectetur adipiscing elit. Vel facilis fugiat labore, corr<
12 </div>
13 <div>
14 <h3>guia para decorar tu hogar</h3>
15 <!--img-->
16 <p>Escrito el: fecha, by user</p>
17 <p>Lorem ipsum dolor sit amet consectetur adipiscing elit. Vel facilis fugiat labore, corr<
18 </div>
19 <div>
20 <h3>guia para decorar tu habitacion</h3>
21 <!--img-->
22 <p>Escrito el: fecha, by user</p>
23 <p>Lorem ipsum dolor sit amet consectetur adipiscing elit. Vel facilis fugiat labore, corr<
24 </div>
25 </section>
26
27 <?php
28 include "../HTML/includes/footer.php";
29 ?>
```

Este apartado simplemente es un pequeño blog del sitio web, despliega información y contiene el header y footer creado anteriormente.



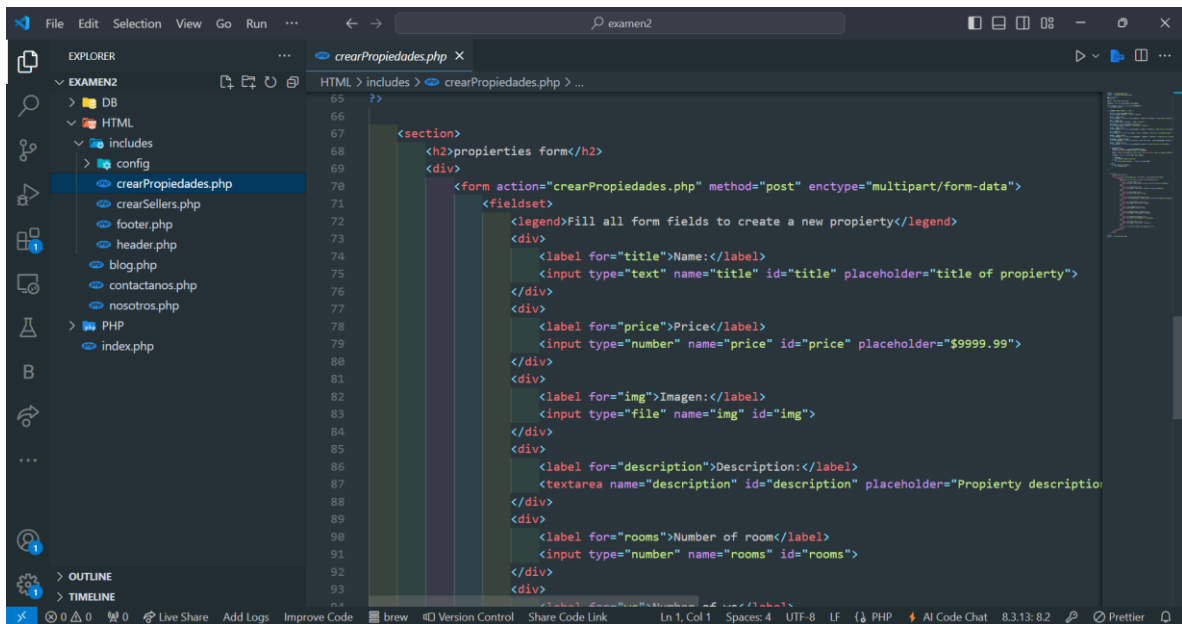
```
4 <section>
7 </section>
8 <section>
9 <h2>llena el formulario de contacto</h2>
10 <div>
11 <form action="">
12 <fieldset>
13 <legend>Informacion personal</legend>
14 <div>
15 <label for="name">Nombre:</label>
16 <input type="text" name="name" id="name" placeholder="Your Name">
17 </div>
18 <div>
19 <label for="email">Correo:</label>
20 <input type="email" name="email" id="email" placeholder="your@email">
21 </div>
22 <div>
23 <label for="phone">Telefono:</label>
24 <input type="tel" name="phone" id="phone" placeholder="555 555 5555">
25 </div>
26 <div>
27 <label for="msg">Mensaje:</label>
28 <textarea name="msg" id="msg" placeholder="Your message"></textarea>
29 </div>
30 </fieldset>
31 <fieldset>
32 <legend>informacion de la propiedad</legend>
33 <div>
34 <label for="vencom">Vender o comprar</label>
35 </div>
36 </fieldset>
37 </div>
38 </form>
39 </div>
40 </section>
41 </html>
```

Pequeño formulario que permite enviar un mensaje para contactar a la persona ante cualquier duda o malentendido.



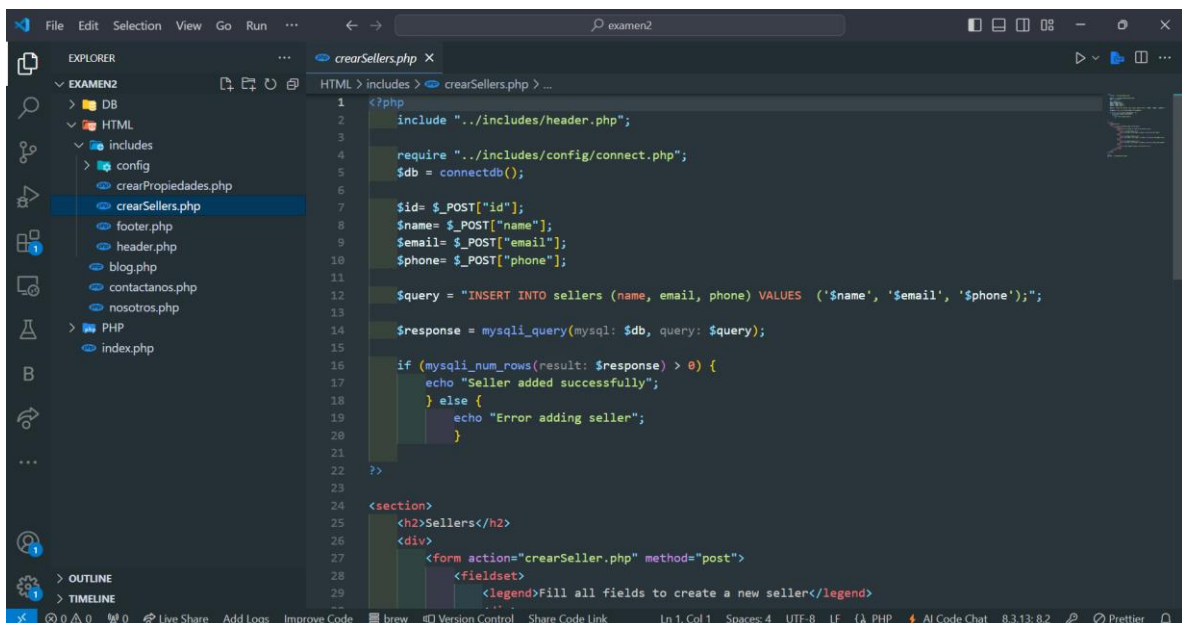
```
1 <?php
2 include "../HTML/includes/header.php";
3 <?>
4 <section>
5 <h2>Conoce sobre nosotros</h2>
6 <!-- IMG -->
7 <p>25 year de experience</p>
8 <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Atque, odio? Laboriosam facilis di.
9 <p>Lorem ipsum dolor sit amet consectetur adipisicing</p>
10 </section>
11
12 <section>
13 <h2>Mas sobre nosotros</h2>
14 <div>
15 <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="black" stroke-width="2">
16 <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Sed sit amet nulla auctor,</p>
17 </div>
18 <div>
19 <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="black" stroke-width="2">
20 <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Sed sit amet nulla auctor,</p>
21 </div>
22 <div>
23 <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="black" stroke-width="2">
24 <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Sed sit amet nulla auctor,</p>
25 </div>
26 </section>
27 <?php
28 include "../HTML/includes/footer.php";
29 <?>
```

Pequeña área sobre la información de la empresa con iconos y texto.



```
65 <?php
66
67 <section>
68 <h2>propiedades form</h2>
69 <div>
70 <form action="crearPropiedades.php" method="post" enctype="multipart/form-data">
71 <fieldset>
72 <legend>Fill all form fields to create a new propriety</legend>
73 <div>
74 <label for="title">Name:</label>
75 <input type="text" name="title" id="title" placeholder="title of propriety">
76 </div>
77 <div>
78 <label for="price">Price:</label>
79 <input type="number" name="price" id="price" placeholder="$9999.99">
80 </div>
81 <div>
82 <label for="img">Imagen:</label>
83 <input type="file" name="img" id="img">
84 </div>
85 <div>
86 <label for="description">Description:</label>
87 <textarea name="description" id="description" placeholder="Property description">
88 </div>
89 <div>
90 <label for="rooms">Number of room</label>
91 <input type="number" name="rooms" id="rooms">
92 </div>
93 </div>
94 </form>
95 </div>
96 </section>
```

Formulario que permite crear propiedades, el cual cuenta con su propio codigo php incluido y validado por cualquier error



```
1 <?php
2 include "../includes/header.php";
3
4 require "../includes/config/connect.php";
5 $db = connectdb();
6
7 $id= $_POST["id"];
8 $name= $_POST["name"];
9 $email= $_POST["email"];
10 $phone= $_POST["phone"];
11
12 $query = "INSERT INTO sellers (name, email, phone) VALUES ('$name', '$email', '$phone')";
13
14 $response = mysqli_query(mysql: $db, query: $query);
15
16 if (mysqli_num_rows(result: $response) > 0) {
17 echo "Seller added successfully";
18 } else {
19 echo "Error adding seller";
20 }
21
22 >?
23
24 <section>
25 <h2>Sellers</h2>
26 <div>
27 <form action="crearSeller.php" method="post">
28 <fieldset>
29 <legend>Fill all fields to create a new seller</legend>
```

Formulario para crear un vendedor el cual como se aprecia en la imagen, cuenta con el header y requiere la conexión a la base de datos por la función que tiene.