Like:  
index.php

Consulta SQL para contar "likes": Se ejecuta una consulta SQL para contar cuántos "me gusta" tiene una publicación específica (id\_publicacion). Esto se hace mediante la consulta SELECT count(id\_megusta) as total\_megusta FROM megusta WHERE id\_publicacion = $id\_publicacion.

Checkbox de "like": Se muestra un checkbox que representa si al usuario le gusta o no la publicación actual. El estado de este checkbox (checked o no) se determina mediante la variable $ya\_le\_gusta, que indica si el usuario ya le ha dado "me gusta" a esa publicación.

JavaScript

Evento de cambio del checkbox: Se utiliza el evento change para detectar cuando el usuario marca o desmarca el checkbox de "like".

Solicitud AJAX: Cuando el usuario cambia el estado del checkbox, se envía una solicitud AJAX al archivo like.php. Esta solicitud incluye el ID de la publicación y la acción a realizar (add para agregar "me gusta" o remove para eliminarlo).

Actualización de la UI: Después de que se procesa la solicitud AJAX y se recibe una respuesta del servidor, se actualiza dinámicamente la interfaz de usuario para reflejar el cambio en el número total de "me gusta" de la publicación.

like.php

Verificación del usuario: Se verifica si el usuario está autenticado. Si no lo está, se redirige a la página de inicio de sesión.

Obtención de datos: Se obtienen los datos necesarios de la solicitud POST, como el ID de la publicación, la acción a realizar y el ID del usuario.

Lógica de "like": Dependiendo de la acción especificada (add o remove), se agrega o elimina un registro en la tabla megusta para indicar si al usuario le gusta o no la publicación.

Notificación: Si se agrega un nuevo "me gusta", también se inserta una notificación en la tabla notificaciones para informar al autor de la publicación sobre el nuevo "me gusta".

Comentar:  
Recuperación de datos de la base de datos: En index.php, se recuperan las publicaciones de la base de datos junto con sus respectivos comentarios. Esto se realiza mediante consultas SQL que obtienen tanto las publicaciones como los comentarios asociados a cada una.

Bucle de publicaciones: Se utiliza un bucle para iterar sobre todas las publicaciones recuperadas de la base de datos. Para cada publicación, se muestra su contenido, incluyendo el autor, la imagen, la descripción y los comentarios asociados.

Formulario de comentarios: Para cada publicación, se muestra un formulario de comentarios que permite a los usuarios agregar nuevos comentarios. Este formulario contiene un campo de texto para el comentario y un botón de envío.

Envío de comentarios mediante AJAX: Cuando un usuario envía un comentario utilizando el formulario, se utiliza JavaScript para capturar el evento de envío y enviar los datos del comentario al archivo comentar.php mediante una solicitud AJAX.

Actualización de la interfaz de usuario: Después de enviar el comentario con éxito, se utiliza JavaScript para actualizar dinámicamente la interfaz de usuario y mostrar el nuevo comentario sin tener que recargar toda la página.

Notificaciones: Además de agregar el comentario, comentar.php también inserta una notificación en la tabla notificaciones para informar al autor de la publicación sobre el nuevo comentario.