

Explicación del código traducido al español

- **Login.php**

- La estructura HTML básica se define, incluyendo los encabezados, el título de la página y los enlaces a las bibliotecas Bootstrap y jQuery.
- El formulario de inicio de sesión. Se utiliza el atributo "action" para indicar la ubicación del archivo PHP que procesará los datos del formulario.
- Dentro del formulario, se utilizan etiquetas HTML para crear un diseño con columnas y filas que contienen campos de entrada de texto para el nombre de usuario y la contraseña.
- El campo de entrada de texto para el nombre de usuario se define, utilizando la etiqueta "input" con el atributo "type" establecido en "text".
- El campo de entrada de texto para la contraseña se define, utilizando la etiqueta "input" con el atributo "type" establecido en "password". También hay un campo oculto definido que se utiliza para indicar la acción que se debe realizar en el archivo PHP que procesa el formulario.
- El botón "Sign In" se define utilizando la etiqueta "input" con el atributo "type" establecido en "submit".

- **Functions.php**

- Requiere un archivo llamado "database.php" que se encuentra en la carpeta "controllers" en el nivel superior al archivo actual. Es probable que este archivo contenga información de conexión a una base de datos MySQL.
- Verifica si el campo "action" está presente en la matriz \$_POST, lo que indica que se ha enviado el formulario. Si se envió el formulario, se utiliza un switch para determinar qué función debe ser llamada. En este caso, la única opción es "access_user".
- La función "access_user()". Esta función recupera el nombre de usuario y la contraseña enviados desde el formulario y los almacena en variables.
- La sesión se inicia utilizando la función "session_start()". A continuación, se almacena el nombre de usuario en la variable \$_SESSION['name'].
- Se establece una conexión a una base de datos MySQL utilizando la función "mysqli_connect()". Se utiliza el servidor "localhost", el nombre de usuario "root", la contraseña vacía y la base de datos "r_user". Es probable que estos valores deban ajustarse para que se ajusten a la configuración de la base de datos local.
- Define una consulta SQL que busca un registro en la tabla "user" que coincida con el nombre de usuario y la contraseña proporcionados.
- Se utiliza la función "mysqli_query()" para ejecutar la consulta y almacenar el resultado en la variable \$result.

- La función "mysqli_fetch_array()" para obtener el primer registro del resultado y almacenarlo en la variable \$rows. Esto nos permite verificar el valor del campo "rol" para determinar qué página debe cargarse después del inicio de sesión.
- Una serie de declaraciones "if" para determinar qué página debe cargarse después del inicio de sesión, según el valor del campo "rol". Si el valor es 1, se carga la página "admin.php"; si es 2, se carga la página "accounting.php"; si es 3, se carga la página "marketing.php".
- Si el valor del campo "rol" no coincide con ninguno de los valores esperados, se muestra un mensaje de alerta. El mensaje indica que el nombre de usuario o la contraseña no son válidos y redirige al usuario a la página de inicio de sesión. También se utiliza la función "session_destroy()" para destruir la sesión y evitar que el usuario acceda a páginas posteriores a través del historial del navegador.

- **Database.php**
- Se definen cuatro variables: \$host, \$user, \$password y \$database, que contienen la información necesaria para conectarse a la base de datos.
- Se utiliza la función mysqli_connect() para establecer la conexión con la base de datos. Se le pasan como argumentos las variables definidas anteriormente. El resultado de esta función se almacena en la variable \$connect.
- Se verifica si la conexión se realizó correctamente mediante un condicional if. Si no se realizó la conexión, se imprime un mensaje de error utilizando la función mysqli_connect_error().

- **Logout.php**
- Este código inicia una sesión en PHP usando la función session_start() y luego destruye todas las variables de sesión almacenadas en ella usando session_destroy(). Luego, redirige al usuario a la página de inicio de sesión con header("Location:../views/login.php"). Esto se utiliza comúnmente para cerrar la sesión del usuario y redirigirlo a la página de inicio de sesión.

- **Admin.php**
- Inicia una sesión usando `session_start()` y desactiva los mensajes de error usando `error_reporting(0)`.
- Comprueba si hay un nombre de usuario almacenado en la sesión. Si el valor almacenado es nulo o vacío, se redirige a la página de inicio de sesión usando `header("Location: ../views/login.php")` y se detiene el proceso usando `die()`.
- Si hay un nombre de usuario almacenado en la sesión, se muestra la página de administrador. Esta página muestra la lista de usuarios almacenados en una base de datos.
- La página se encarga de cargar los estilos y scripts necesarios, usando varias etiquetas `link` y `script`.
- La página muestra un encabezado de bienvenida que incluye el nombre del usuario almacenado en la sesión, y un botón de cierre de sesión.
- La página también muestra una tabla que contiene los datos de todos los usuarios almacenados en la base de datos.
- Para mostrar los datos de los usuarios, el código realiza una consulta en la base de datos utilizando el comando `SELECT` y la cláusula `LEFT JOIN`. Los datos se recuperan de la tabla `user` y se une con la tabla `permissions` para recuperar el rol del usuario. Los resultados se almacenan en la variable `$fact`.
- Si la consulta devuelve uno o más resultados, los datos se muestran en una tabla utilizando un bucle `while`. Si no hay resultados, se muestra un mensaje indicando que no hay usuarios en la tabla.
- La página también incluye algunos scripts al final para habilitar la funcionalidad de paginación y búsqueda en la tabla mediante la librería DataTables y un archivo JavaScript llamado `user.js`.

- **Accounting.php**
- Este código comienza con la llamada a la función `session_start()` para iniciar la sesión y luego establece la opción `error_reporting()` para desactivar la notificación de errores en la página.
- Luego, se realiza una validación de la variable de sesión `name` para comprobar si existe o no un valor. Si la variable no tiene un valor, la página se redirige a la página de inicio de sesión.
- Después de eso, se crea el código HTML y se utiliza la función `curl_init()` de PHP para hacer una solicitud HTTP al API de "reqres.in" que devuelve una lista de usuarios.
- La respuesta de la solicitud HTTP se decodifica utilizando la función `json_decode()` y se muestra en una tabla con la ayuda de un bucle `foreach`.
- Por último, se incluyen algunos archivos de JavaScript para agregar funcionalidades adicionales a la página, como la búsqueda y ordenamiento de datos en la tabla, entre otras.

- **Marketing.php**
- El código comienza con la función `session_start()` para iniciar la sesión.
- Luego, se utiliza la variable `$_SESSION['name']` para comprobar si el usuario ha iniciado sesión o no. Si no hay una sesión activa o la variable `$_SESSION['name']` es nula o vacía, el código redirige al usuario a la página de inicio de sesión `login.php` utilizando la función `header()`.
- Si se encuentra una sesión activa, se muestra la página web con el título "Marketing" y se muestra un saludo personalizado que incluye el nombre del usuario recuperado de la variable `$_SESSION['name']`.
- También se muestra un mensaje que indica que el usuario no tiene ninguna acción asignada y se le sugiere que se ponga en contacto con el administrador para solucionar el problema.
- Por último, se incluyen algunos archivos de JavaScript para agregar funcionalidades adicionales a la página, como la búsqueda y ordenamiento de datos en la tabla, entre otras.