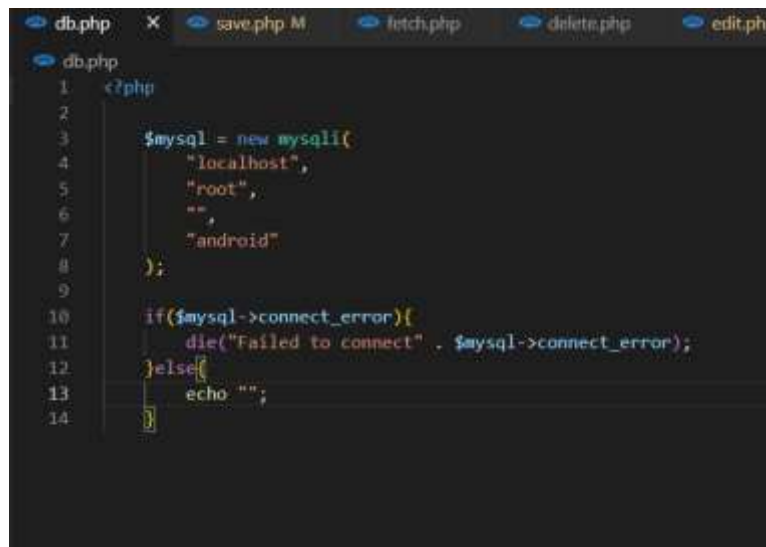


db.php

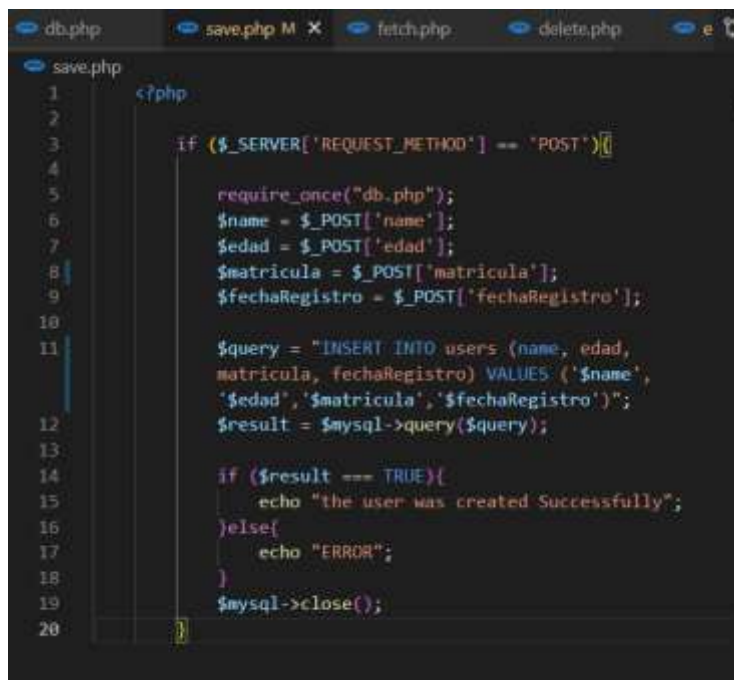
A screenshot of a code editor with a dark theme. The editor has several tabs at the top: 'db.php' (active), 'save.php M', 'fetch.php', 'delete.php', and 'edit.php'. The 'db.php' tab is selected, showing the following PHP code:

```
1 <?php
2
3 $mysql = new mysqli(
4     "localhost",
5     "root",
6     "",
7     "android"
8 );
9
10 if($mysql->connect_error){
11     die("Failed to connect" . $mysql->connect_error);
12 }else{
13     echo "";
14 }
```

Este es un script PHP que crea una nueva instancia de la clase `mysqli` para conectarse a una base de datos MySQL. El script intenta conectarse a una base de datos llamada "android" en un servidor local utilizando el nombre de usuario "root" por defecto y una contraseña vacía.

Si la conexión no tiene éxito, el script terminará con un mensaje de error que indica la razón del fallo. De lo contrario, el script no hará nada y simplemente imprimirá una cadena vacía.

save.php



```
1 <?php
2
3 if ($_SERVER['REQUEST_METHOD'] == 'POST'){
4
5     require_once("db.php");
6     $name = $_POST['name'];
7     $edad = $_POST['edad'];
8     $matricula = $_POST['matricula'];
9     $fechaRegistro = $_POST['fechaRegistro'];
10
11     $query = "INSERT INTO users (name, edad,
12 matricula, fechaRegistro) VALUES ('$name',
13 '$edad', '$matricula', '$fechaRegistro')";
14     $result = $mysql->query($query);
15
16     if ($result === TRUE){
17         echo "the user was created Successfully";
18     }else{
19         echo "ERROR";
20     }
21     $mysql->close();
22 }
```

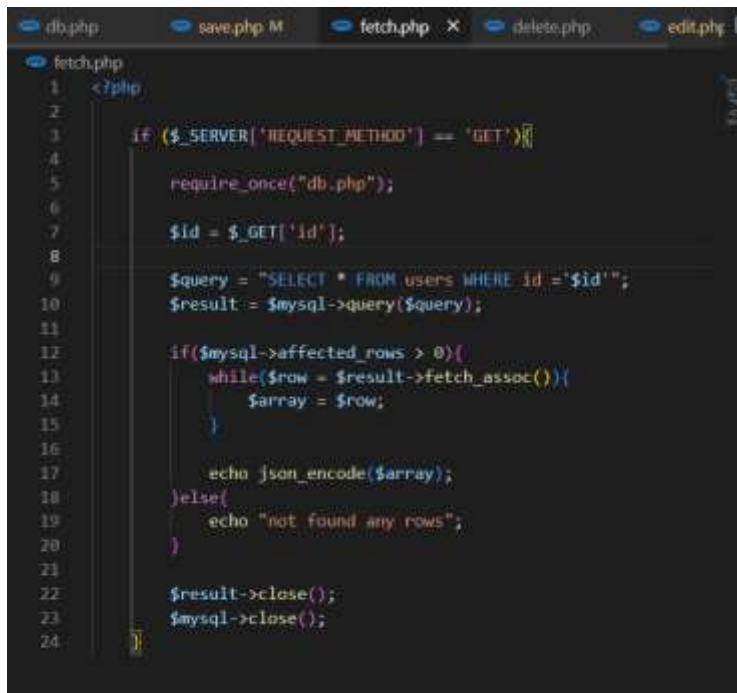
Este es un script PHP que maneja una solicitud POST y que depende del archivo db.php para conectarse a una base de datos MySQL.

El script recupera los valores enviados por la solicitud POST, es decir, los valores de name, edad, matricula y fechaRegistro, y los almacena en variables PHP.

A continuación, se construye una consulta SQL INSERT utilizando los valores recuperados de la solicitud POST y se ejecuta en la base de datos. Si la consulta se ejecuta correctamente, el script imprime un mensaje indicando que el usuario se ha creado correctamente. De lo contrario, se imprime un mensaje de error.

Finalmente, se cierra la conexión a la base de datos utilizando el método close() de la instancia de la clase mysqli.

Fetch.php



```
1 <?php
2
3 if ($_SERVER['REQUEST_METHOD'] == 'GET'){
4
5     require_once("db.php");
6
7     $id = $_GET['id'];
8
9     $query = "SELECT * FROM users WHERE id = '$id'";
10    $result = $mysql->query($query);
11
12    if($mysql->affected_rows > 0){
13        while($row = $result->fetch_assoc()){
14            $array = $row;
15        }
16
17        echo json_encode($array);
18    }else{
19        echo "not found any rows";
20    }
21
22    $result->close();
23    $mysql->close();
24 }
```

Este es un script PHP que maneja una solicitud GET y que depende del archivo `db.php` para conectarse a una base de datos MySQL.

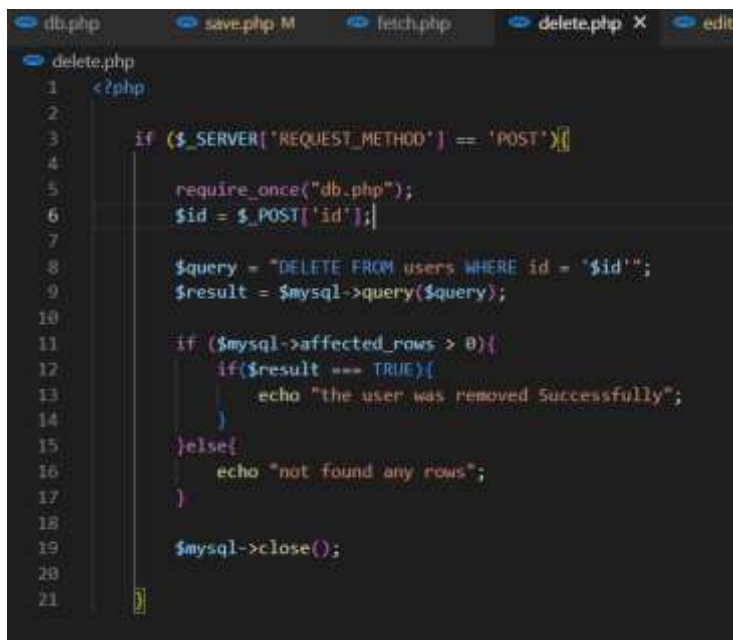
El script recupera el valor `id` enviado por la solicitud GET y lo almacena en una variable PHP. A continuación, se construye una consulta SQL SELECT para recuperar los datos correspondientes al usuario con ese `id` de la tabla `users`.

La consulta se ejecuta en la base de datos utilizando el método `query()` de la instancia de la clase `mysqli`. Si la consulta devuelve algún resultado, se utiliza un bucle `while` para recorrer los resultados y almacenarlos en un array PHP.

El array resultante se convierte en formato JSON utilizando la función `json_encode()`, que se utiliza para devolver los datos recuperados al cliente en forma de una respuesta HTTP. Si la consulta no devuelve ningún resultado, se imprime un mensaje indicando que no se encontraron filas.

Finalmente, se cierra el objeto `result` utilizando el método `close()` de la instancia de la clase `mysqli`, y se cierra la conexión a la base de datos utilizando el método `close()` de la misma instancia.

Delete.php

A screenshot of a code editor with a dark background. The editor shows a file named 'delete.php' with PHP code. The code starts with a conditional check for a POST request. If it is a POST request, it includes 'db.php', gets the 'id' from the POST data, and constructs a SQL DELETE query for the 'users' table. It then executes the query using the mysqli query method. After execution, it checks if any rows were affected and if the result was true, it echoes a success message. Otherwise, it echoes a message indicating no rows were found. Finally, it closes the mysqli connection. The code is as follows:

```
1 <?php
2
3 if ($_SERVER['REQUEST_METHOD'] == 'POST'){
4
5     require_once("db.php");
6     $id = $_POST['id'];
7
8     $query = "DELETE FROM users WHERE id = '$id'";
9     $result = $mysqli->query($query);
10
11     if ($mysqli->affected_rows > 0){
12         if($result === TRUE){
13             echo "the user was removed Successfully";
14         }
15     }else{
16         echo "not found any rows";
17     }
18
19     $mysqli->close();
20
21 }
```

El script recupera el valor id enviado por la solicitud POST y lo almacena en una variable PHP. A continuación, se construye una consulta SQL DELETE para eliminar el usuario correspondiente al id de la tabla users.

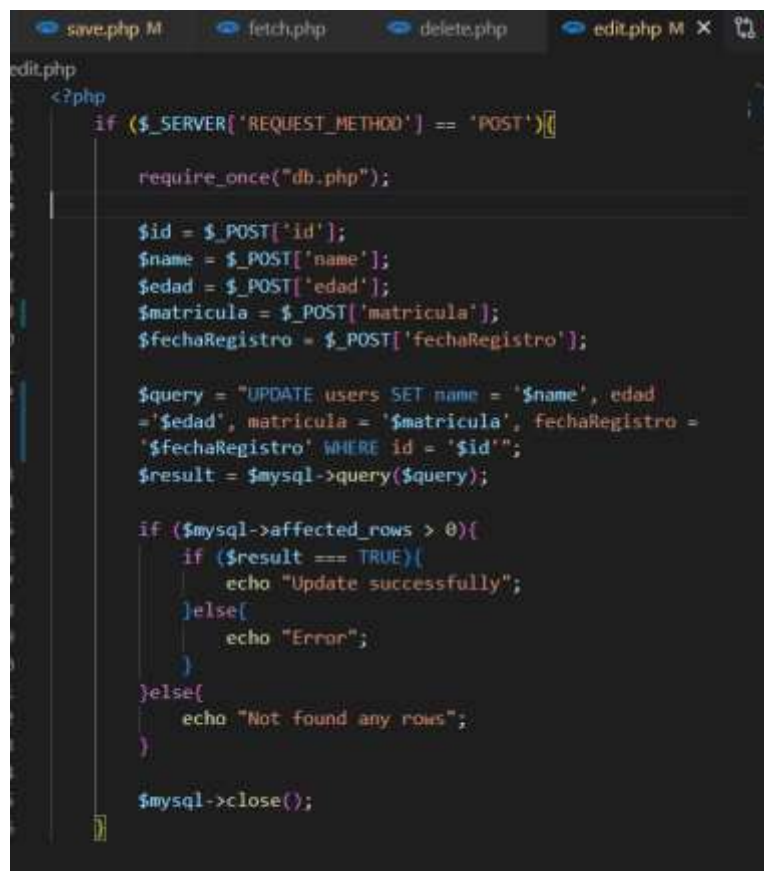
La consulta se ejecuta en la base de datos utilizando el método `query()` de la instancia de la clase `mysqli`. Si la consulta afecta alguna fila, lo que significa que se ha eliminado el usuario, se comprueba si la consulta se ha ejecutado correctamente, y si es así,

se imprime un mensaje indicando que el usuario se ha eliminado correctamente.

Si la consulta no afecta ninguna fila, lo que significa que no se ha encontrado ningún usuario con ese id, se imprime un mensaje indicando que no se encontraron filas.

Finalmente, se cierra la conexión a la base de datos utilizando el método `close()` de la instancia de la clase `mysqli`.

Edit.php



```
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST'){
    require_once("db.php");

    $id = $_POST['id'];
    $name = $_POST['name'];
    $edad = $_POST['edad'];
    $matricula = $_POST['matricula'];
    $fechaRegistro = $_POST['fechaRegistro'];

    $query = "UPDATE users SET name = '$name', edad = '$edad', matricula = '$matricula', fechaRegistro = '$fechaRegistro' WHERE id = '$id'";
    $result = $mysql->query($query);

    if ($mysql->affected_rows > 0){
        if ($result === TRUE){
            echo "Update successfully";
        }else{
            echo "Error";
        }
    }else{
        echo "Not found any rows";
    }

    $mysql->close();
}
```

El script recupera los valores id, name, edad, matricula y fechaRegistro enviados por la solicitud POST y los almacena en variables PHP correspondientes.

A continuación, se construye una consulta SQL UPDATE para actualizar el usuario correspondiente al id en la tabla users. La consulta establece nuevos valores para name, edad, matricula y fechaRegistro.

La consulta se ejecuta en la base de datos utilizando el método query() de la instancia de la clase mysqli. Si la consulta afecta alguna fila, lo que significa que se ha actualizado el usuario, se comprueba si la consulta se ha ejecutado correctamente, y si es así, se imprime un mensaje indicando que el usuario se ha actualizado correctamente.

Si la consulta no afecta ninguna fila, lo que significa que no se ha encontrado ningún usuario con ese id, se imprime un mensaje indicando que no se encontraron filas.

Finalmente, se cierra la conexión a la base de datos utilizando el método close() de la instancia de la clase mysqli

MainActivity

Este es un código de la aplicación Android que permite crear y leer datos de una base de datos MySQL a través de una API PHP. La aplicación tiene una actividad principal (MainActivity) con dos botones: uno para crear un nuevo usuario y otro para buscar un usuario existente.

Cuando se hace clic en el botón para crear un nuevo usuario, se recogen los datos de entrada del usuario, se crea una solicitud de cadena y se envía a la URL de la API PHP a través de una cola de solicitudes de Volley. La API PHP recibirá los datos, los procesará y los insertará en la base de datos MySQL. Si la operación se realiza correctamente, se muestra un mensaje de confirmación en la aplicación.

Cuando se hace clic en el botón para buscar un usuario existente, se recoge el ID del usuario introducido por el usuario, se crea una intención para iniciar una nueva actividad (MainActivity2) y se envía el ID como un extra. La nueva actividad mostrará los datos del usuario correspondiente a ese ID.

En general, este código ilustra cómo una aplicación Android puede interactuar con una base de datos MySQL a través de una API PHP.

MainActivity2

La función `readUser` se encarga de enviar una solicitud GET a una URL específica, que incluye un parámetro `id` que se utiliza para recuperar datos específicos del usuario del servidor. El servidor devuelve un objeto JSON que contiene los datos del usuario, que se analizan y se muestran en algunos campos de texto en la actividad.

La función `onClick` se activa cuando el usuario hace clic en un botón de la actividad. Dependiendo del botón que se haya presionado, la función llama a la función `updateUser` o a la función `removeUser`.

La función `updateUser` se encarga de enviar una solicitud POST a una URL específica, que incluye los nuevos datos del usuario que se han introducido en los campos de texto de la actividad. Los datos se envían en forma de parámetros en el cuerpo de la solicitud POST. El servidor recibe estos datos y actualiza el registro correspondiente en su base de datos. Si todo funciona correctamente, la función muestra un mensaje en forma de Toast que indica que la actualización se ha realizado correctamente.

La función `removeUser` se encarga de enviar una solicitud POST a una URL específica para eliminar un usuario específico de la base de datos del servidor. La solicitud incluye el ID del usuario que se va a eliminar. Si la solicitud tiene éxito, la función finaliza la actividad actual.

En resumen, este código se encarga de realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en una base de datos remota utilizando solicitudes HTTP en una aplicación Android.