API Documentation: Register User API

Overview

This PHP API handles user registration by accepting `user_id`, `user_name`, and `user_password` as input. It checks if the `user_id` is already taken and then either inserts the new user into the database or returns an error message.

Base URL

The API can be accessed at your server endpoint, e.g., `http://localhost/register_user.php`.

HTTP Methods

- **POST**: Used for registering a new user by providing the required fields: `user_id`, `user_name`, and `user_password`.

Input Parameters

The API expects a JSON payload with the following fields:

Field	Type	Description	Required
`user_id`	`string`	A unique identifier for the user.	Yes
`user_name`	`string`	The name of the user.	Yes
`user_password`	`string`	The password for the user's account.	Yes

Sample Request

To register a new user, make a `POST` request with a JSON body like the following:

```
json
Copy code
{
    "user_id": "user123",
    "user_name": "John Doe",
    "user_password": "securepassword123"
}
```

Responses

Success Response

If the registration is successful, the API returns the following JSON response:

```
json
Copy code
{
    "user_id": "user123",
    "user_name": "John Doe",
    "message": "registration successful"
}
```

Error Responses

1. **User ID Already Exists** If the `user_id` already exists in the database, the API returns this error message:

   ```
   json
   Copy code
   {
       "error": true,
       "message": "User ID already exists. Please choose another one."
   }
   ```

2. **Missing Input Parameters** If any of the required fields (`user_id`, `user_name`, or `user_password`) are missing from the request body, the API returns this error:

   ```
   json
   Copy code
   {
       "error": true,
       "message": "Missing user_id, user_name, or user_password"
   }
   ```

3. **Registration Failure** If the registration fails due to any reason (e.g., database issues), the API will return:

   ```
   json
   Copy code
   {
       "error": true,
       "message": "Failed to complete registration. Try again."
   }
   ```

Code Explanation

1. Database Connection

The PHP code connects to a MySQL database with the following credentials:

```
php
Copy code
$host = "localhost";
$username = "root";
$password = "";
$db_name = "attendance_system";

$conn = new mysqli($host, $username, $password, $db_name);
```

- **$host**: Database host (localhost in this case).
- **$username**: Database username (root by default).
- **$password**: Database password (empty by default for localhost).
- **$db_name**: The name of the database to connect to (`attendance_system`).

---

2. Input Handling

The JSON payload is read from the request body using `file_get_contents("php://input")` and then decoded into an associative array:

```
php
Copy code
$data = json_decode(file_get_contents("php://input"), true);
```

---

3. User Registration Logic

The API handles the `POST` request by performing the following steps:

- **Check if required fields are provided**: If `user_id`, `user_name`, and `user_password` are present, the code proceeds. Otherwise, an error response is returned.
- **Check if the `user_id` already exists**: A SQL query checks the database for an existing `user_id`:

  ```
  php
  Copy code
  $checkUserSql = "SELECT * FROM users WHERE user_id = '$user_Id'";
  ```

  If the `user_id` exists, an error message is returned:

  ```
  json
  Copy code
  {
      "error": true,
  ```

```
    "message": "User ID already exists. Please choose another one."
}
```

- **Insert the new user into the database**: If the `user_id` does not exist, a new user is added with the provided details:

  ```
  php
  Copy code
  $sql = "INSERT INTO users (user_id, user_name, user_password) VALUES
  ('$user_Id', '$user_Name', '$user_password')";
  ```

  If the insert is successful, the user's details are returned:

  ```
  json
  Copy code
  {
      "user_id": "user123",
      "user_name": "John Doe",
      "message": "registration successful"
  }
  ```

  If the registration fails, an error message is returned:

  ```
  json
  Copy code
  {
      "error": true,
      "message": "Failed to complete registration. Try again."
  }
  ```

---

4. Method Not Allowed

If the request method is not `POST`, the API responds with a `405 Method Not Allowed` status code:

```
php
Copy code
http_response_code(405);
echo json_encode(["error" => "Method not allowed"]);
```

Security Considerations

1. **SQL Injection Prevention**: The code uses `real_escape_string()` to prevent SQL injection attacks by sanitizing the input data before using it in SQL queries. However, using **prepared statements** would be more secure.

   Example using prepared statements:

```
php
Copy code
$stmt = $conn->prepare("SELECT * FROM users WHERE user_id = ?");
$stmt->bind_param("s", $user_Id);
$stmt->execute();
```

2. **Password Hashing**: For better security, passwords should be hashed before storing them in the database:

   ```
   php
   Copy code
   $hashedPassword = password_hash($user_password, PASSWORD_BCRYPT);
   ```

   And when verifying the password:

   ```
   php
   Copy code
   if (password_verify($user_password, $row['user_password'])) {
       // Password matches
   }
   ```

3. **Input Validation**: Ensure that all input parameters are validated and sanitized to prevent malicious input.

Usage Example

Request Example

```
bash
Copy code
curl -X POST http://localhost/register_user.php \
-H "Content-Type: application/json" \
-d '{
    "user_id": "user123",
    "user_name": "John Doe",
    "user_password": "securepassword123"
}'
```

Response Example

```
json
Copy code
{
    "user_id": "user123",
    "user_name": "John Doe",
    "message": "registration successful"
}
```