**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."

}

1. **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"

}

1. **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();

1. **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

}

1. **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"

}