**API Documentation: Check User API**

**Description**

This API checks if a user exists in the database based on the provided user\_id. If the user exists, it returns the user's details. If not, it returns an error message indicating that the user doesn’t exist.

**Endpoint**

arduino

Copy code

http://<your-server-host>/check\_user.php

**HTTP Method**

POST

**Headers**

| **Key** | **Value** |
| --- | --- |
| Content-Type | application/json |

**Request Parameters**

The API expects a JSON payload with the following fields:

| **Parameter Name** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| user\_id | string | Yes | Unique ID of the user. |

**Example Request Body**:

json

Copy code

{

"user\_id": "12345"

}

**Response Structure**

The API returns a JSON response. The structure depends on whether the user exists or not.

**1. Successful Response (User Exists)**

If the user exists in the database, the API returns the following:

| **Field Name** | **Type** | **Description** |
| --- | --- | --- |
| user\_id | string | The ID of the user. |
| user\_name | string | The name of the user. |

**Example Response**:

json

Copy code

{

"user\_id": "12345",

"user\_name": "John Doe"

}

**2. Error Response (User Does Not Exist)**

If the user is not found in the database, the API returns an error message.

| **Field Name** | **Type** | **Description** |
| --- | --- | --- |
| error | boolean | Indicates an error (true). |
| message | string | Description of the error. |

**Example Response**:

json

Copy code

{

"error": true,

"message": "User doesn’t exist"

}

**3. Error Response (Missing user\_id)**

If user\_id is not provided in the request, the API returns the following:

| **Field Name** | **Type** | **Description** |
| --- | --- | --- |
| error | boolean | Indicates an error (true). |
| message | string | Description of the error. |

**Example Response**:

json

Copy code

{

"error": true,

"message": "Invalid input data"

}

**4. Error Response (Invalid HTTP Method)**

If a method other than POST is used, the API responds with a method not allowed error:

| **Field Name** | **Type** | **Description** |
| --- | --- | --- |
| error | boolean | Indicates an error (true). |
| message | string | Description of the error. |

**Example Response**:

json

Copy code

{

"error": true,

"message": "Method not allowed"

}

**Error Codes**

| **HTTP Status Code** | **Description** |
| --- | --- |
| 200 | Success |
| 400 | Bad Request (e.g., missing user\_id). |
| 405 | Method Not Allowed |

**Example Usage**

**Using Postman**

1. Set the HTTP method to POST.
2. Enter the endpoint URL: http://<your-server-host>/check\_user.php.
3. In the Headers section, set Content-Type: application/json.
4. In the Body section, select raw and enter:

json

Copy code

{

"user\_id": "12345"

}

1. Send the request and check the response.

**Using cURL**

bash

Copy code

curl -X POST \

-H "Content-Type: application/json" \

-d '{"user\_id": "12345"}' \

http://<your-server-host>/check\_user.php

**Implementation Details**

This API connects to the attendance\_system database and performs the following steps:

1. Reads the request body and extracts user\_id.
2. Queries the database to check if a user with the given user\_id exists.
3. Responds with user details if the user exists.
4. Returns an error if the user does not exist, the input is invalid, or the method is not POST.

**Improvements**

* **Authentication**: Add an API key or token mechanism for security.
* **Validation**: Validate user\_id to prevent SQL injection.
* **Rate Limiting**: Prevent abuse by limiting the number of API calls per user.