Table for messages

CREATE TABLE Messages (

message\_id INT AUTO\_INCREMENT PRIMARY KEY,

sender\_id INT NOT NULL,

receiver\_id INT DEFAULT NULL, -- Used for individual messages

group\_receiver\_id INT DEFAULT NULL, -- Used for group messages

message\_content TEXT NOT NULL,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (sender\_id) REFERENCES Users(user\_id),

FOREIGN KEY (receiver\_id) REFERENCES Users(user\_id),

FOREIGN KEY (group\_receiver\_id) REFERENCES Groups(group\_id)

);

APIs

Certainly! Here are the refined functions for sending a message (function 1), retrieving individual messages (function 2), and retrieving group messages (function 3).

Database Connection (Common to All Functions)

Assuming your database connection is stored in $conn:

$conn = new mysqli(‘hostname’, ‘username’, ‘password’, ‘database’);

If ($conn->connect\_error) {

Die(“Connection failed: “ . $conn->connect\_error);

}

1. Send Message Function

This function sends a message to either a user or a group, based on the provided parameters.

Function sendMessage($conn, $sender\_id, $message\_content, $receiver\_id = null, $group\_receiver\_id = null) {

If ($receiver\_id || $group\_receiver\_id) {

$stmt = $conn->prepare(“INSERT INTO Messages (sender\_id, receiver\_id, group\_receiver\_id, message\_content, timestamp) VALUES (?, ?, ?, ?, NOW())”);

$stmt->bind\_param(“iiis”, $sender\_id, $receiver\_id, $group\_receiver\_id, $message\_content);

$stmt->execute();

If ($stmt->affected\_rows > 0) {

Return json\_encode([“status” => “success”, “message” => “Message sent successfully”]);

} else {

Return json\_encode([“status” => “error”, “message” => “Failed to send message”]);

}

$stmt->close();

} else {

Return json\_encode([“status” => “error”, “message” => “Specify either receiver\_id or group\_receiver\_id”]);

}

}

Usage:

Echo sendMessage($conn, $sender\_id, $message\_content, $receiver\_id, $group\_receiver\_id);

1. Get Individual Messages Function

This function retrieves messages between two users (sender and receiver) in both directions.

Function getMessages($conn, $user\_id, $other\_user\_id) {

$stmt = $conn->prepare(“SELECT \* FROM Messages WHERE (sender\_id = ? AND receiver\_id = ?) OR (sender\_id = ? AND receiver\_id = ?) ORDER BY timestamp ASC”);

$stmt->bind\_param(“iiii”, $user\_id, $other\_user\_id, $other\_user\_id, $user\_id);

$stmt->execute();

$result = $stmt->get\_result();

$messages = [];

While ($row = $result->fetch\_assoc()) {

$messages[] = $row;

}

$stmt->close();

Return json\_encode($messages);

}

Usage:

Echo getMessages($conn, $user\_id, $other\_user\_id);

1. Get Group Messages Function

This function retrieves all messages sent to a specified group.

Function getGroupMessages($conn, $group\_id) {

$stmt = $conn->prepare(“SELECT \* FROM Messages WHERE group\_receiver\_id = ? ORDER BY timestamp ASC”);

$stmt->bind\_param(“i”, $group\_id);

$stmt->execute();

$result = $stmt->get\_result();

$messages = [];

While ($row = $result->fetch\_assoc()) {

$messages[] = $row;

}

$stmt->close();

Return json\_encode($messages);

}

Usage:

Echo getGroupMessages($conn, $group\_id);

Summary

sendMessage($conn, $sender\_id, $message\_content, $receiver\_id, $group\_receiver\_id): Sends a message to either a user or a group.

getMessages($conn, $user\_id, $other\_user\_id): Retrieves messages between two users.

getGroupMessages($conn, $group\_id): Retrieves all messages for a specified group.

These functions can now be called to send or retrieve messages based on user or group specifications.