Backend - Chat Features Documentation

Overview

The MeetMe backend supports real-time chat via WebSocket, enabling instant communication between students and professors.

This is backed by MongoDB for persistent message storage and FastAPI's WebSocket support for live messaging.

WebSocket Endpoint

The primary WebSocket route for chat is:

- ‘/ws/chat/{user_id}’

When a user connects to this endpoint:

- Their WebSocket connection is stored in the ‘connected_users’ dictionary.
- Incoming messages are JSON-formatted and include ‘sender_id’, ‘receiver_id’, and ‘message’.
- The backend saves the message to MongoDB and forwards it live if the recipient is connected.

Connected Users Dictionary

A global dictionary ‘connected_users: Dict[str, WebSocket]’ maps user IDs to their open WebSocket

connections.

This is used to:

- Route messages to the appropriate recipient
- Remove users upon disconnection

Message Handling Flow

1. User connects via WebSocket to ‘/ws/chat/{user_id}’
2. WebSocket accepts connection and stores it in ‘connected_users’
3. User sends a JSON message:

```
{
"sender_id": "prof@example.com",
"receiver_id": "student@example.com",
"message": "Let's meet at 3PM."
```

}

4. Server:
   - Inserts the message into ‘chat_messages_collection’
   - Forwards it to the receiver via their WebSocket if connected

Chat History Endpoint

GET ‘/chat/history?user1=prof@example.com&user2=student@example.com’

This fetches the entire conversation (from both directions) sorted by timestamp.

The messages are returned in the following format:

- ✓ sender_id
- ✓ receiver_id
- ✓ message
- ✓ timestamp

MongoDB Schema

Collection: ‘chat_messages’

Each document:

{

"sender_id": str,

"receiver_id": str,

"message": str,

"timestamp": str (ISO 8601)

}

Indexes should be considered on sender_id + receiver_id + timestamp for optimal performance.

Scalability Notes

The current system supports 1-on-1 chat and:

- ✓ Does not require polling
- ✓ Works with any number of users as long as server memory permits
- ✓ Could be extended to support group chats or typing indicators with minor adjustments

All messages are stored and retrieved efficiently, supporting a consistent user experience across sessions.