Thats the ChatGPT prompt I used for adding the chatting feature

Now we are already pretty far in the process of development our SAAS. The next thing we implement now is the chatting feature.

The goal is to implement a functionality, so that a user can send chat messages and an imaginary character (the AI Influencer the user created) is answering.

The chatting feature is completely independent of the subscription or the amount of credits usable. The only limitation we implement is, that a user can maximally send 20 messages per day. After that, the button to send another message should disappear. As this is not so critical we don't have to block that at backend level, we just make the button disappear at the dashboard.

Speaking of the dashboard. The chatting menu has to look like in the two example images I provide and it should be visible together with the image creation menu as soon as the AI Influencer got created.

You will see where the chatting menu has to be located, how the chatting menu should look like with an empty chat and when the chat is used. Pay attention to all the little details how I want the chat and the messages to look like. User messages are always right, AI Avatar messages always left. There is a time stamp and the avatar message should always have this little pink heart on top as indicator that this is a message from the AI sweetheart. As I said, the send message button has to disappear after 20 messages per day, showing up next day again.

For the chatting functionality we will use the following Supabase Tables and parameters:

(table) messages
- id
- chat_id
- role
- content
- created_at

(table) chats
- id
- user_id
- influencer_id
- created_at

The idea for the chatting experience is the following:

-as soon as the user created his AI Influencer (e.g. parameter influencers/is_locked = TRUE) a 10 second timer starts ticking.
-when the 10 seconds are over in the table „chats" the „influencer_id" gets stored and a chat gets started proactively by the app to incentivize the usage of that feature.
-the first message gets send automatically by the AI Influencer (the app) and should go like that: „Hi sweetheart, how are you today? Why don't you tell me your name and how your day is going?;)"
-Also that message has to get stored in the messages table together with the id, role (who said it/ user or Influencer), content and when it was created
-When the user replies that reply gets stored in the same message table as well, so that over time the entire conversation is stored there.

-The goal is to be able to have all messages and responses of a certain user available, so that always a big part of that conversation can be loaded and provided to a n8n workflow that is responsible for creating the AI Influencer responses via a LLM.

-With big part I mean that I always want to provide the last 20 user and AI Influencer messages to n8n. By that the conversation feels kind of natural and the AI Influencer remembers at least a while some details.

Here is the description what we send to n8n and what is coming back:

We send always to that webhook url to provide the data to the n8n workflow responsible for creating the AI Influencer response: https://course.germanaicreator.com/webhook/05594dd1-91e1-456c-a802-22c824fa1d09

We send the following values (all available in Supabase tables, take out what we don't need to send in your opinion)

table) messages
- id
- chat_id

The last twenty messages of user and Influencer with these Information
- role
- content
- created_at

(table) chats
- id
- user_id

(table) influencers
- vibe
- bio
- name

The workflow then creates out of that information a response message that gets send back. We probably also need to send the n8n secret etc.

The response message has to get stored immediately in the right supabase table and of course the latest message has to show up in the chatting window of the dashboard.

It would be amazing to have a small notification showing up under the chatting window somewhere like „Sweetheart is typing…" three seconds after the user message was sent, until the message arrives. We know that feature from WhatsApp and similar chats.

I provide you the current Supabase Schema, app.py, dashboard.html and dashboard.js and the images how the dashboard should look like.

Regarding the app.py only give me back entire new functions and if an existing functions needs an update also give me back that entire updated function, together with the reworked dashboard.html and dashboard.js.

Take as much time as you need to truly think this through and create a fantastic high quality solution. This is very important so think step by step and work very precisely. If this task is to big for your context window demand files separately and we solve that task step by step.