I am building a SAAS right now, for the sake of demonstrating how to do it in a video course.

I already built and documented everything until that current state:

-User can sign up at https://gptsweetheart.com/sign-up
-User gets a confirmation Email from Supabase
-when the user clicks on the Link he gets redirected to https://gptsweetheart.com/redirect and from there to https://gptsweetheart.com/login
-when the user logs in he comes to https://gptsweetheart.com/dashboard, sees that he is on a free subscription and has 4 credits
-when he clicks the button to subscribe to the pro subscription a stripe checkout starts and the user can pay for the subscription
-Afterwards the user gets back to the dashboard and sees that he has now a pro subscription and 20 credits available.

Attached you find the exact schema of the current Supabase data base.

The entire SAAS until now consists of the following right now:
-a Flask app running on a Vultr Server (Nginx, Green unicorn)
-data base Supabase
-Frontend static html pages, each html page has a separate .js file
-auth.js exists as well

Attached you find the exact folder structure where all the files are located right now.

What we have until here is the first working Prototype that had the goal to already set up and test the sign up, subscribe and login process.

In the next step the goal is to develop the SAAS further. In the first step I want to build the following:

The idea is that a user can sign in for free and set up an account. After the email got confirmed he is automatically redirected to the login. When he logs in, he is in his dashboard and automatically on a free account (We built this already). Now comes the new part that we have to build:

That allows him to create his AI Girl by clicking on a button on his dashboard „Create your sweetheart now“ and setting the parameters that will get used to create that character. These are the parameters that have to get stored in Supabase:
-The girls name (Supabase Table influencers/Column: name)
-The girls bio (Supabase Table influencers/Column: bio)
-The girls vibe (Supabase Table influencers/Column: vibe)

After these values get provided via the dashboard to the Flask App, they have to get stored in Supabase.

After that the idea is to send them via webhook to a n8n workflow (lets call it „Influencer Set up Workflow“) that uses the three parameters (name, bio, vibe) to create the following:
-a base prompt that is part of all other images that the user creates of that character
-an initial image of the character by reaching out to replicate via HTTP request
-the seed number of the first image (is a byproduct of the first image that gets created and can be easily accessed via the same replicate API)

When the workflow is done it sends the three parameters back to the Flask app via HTTP request and the Flask app does the following:
-It stores the base prompt in Supabase (Supabase Table influencers/Column: base_prompt)
-It stores the seed number of the initial image in Supabase (Supabase Table influencers/seed: base_prompt)
-It downloads the initial image via the url from Replicate and stores it in the users media folder on the Vultr server

-It stores the url of that image in the users media folder (not Replicate as just temporary url) in Supabase (Supabase Table influencers/Column: initial_image_url)
-It sets the parameter „is_locked“ of that users influencer to true at Supabase (Supabase Table influencers/Column: is_locked)
-It places a time stamp at that users influencers character (Supabase Table influencers/Column: created_at)

After all these parameters are stored in Supabase, the idea is that the user sees from now on that initial image of the character he created in his dashboard in a nice 2:3 window with the characters name under it, whenever he logs in to his account.

What needs to be solved on top is which id gets used to identify each user as there is the column „id“ that is present in all Supabase tables (users influencers, images, chats, messages) and the column „user_id“ that I just present in chats, images, influencers.

We can also change the Supabase Schema if recommended.

Its important, that each user can only create one influencer and that it is clear, which influencer belongs to which user.

This is the webhook URL of the n8n workflow „Influencer Set up Workflow“: **https://your_n8n_webhook_url**

This is the IP address of the Vultr Server the Flask app is running and listening on Port 5002: your_IP_adress

Here are your tasks now:
-Truly understand everything I already built and what I want to build next
-Create the Flask Routes needed and tell me what else needs to be added to the flask app (dependencies, packages, etc)
-Update my dashboard.html and my dashboard.js
-Tell me if I need to change my Supabase structure regarding the IDs

Take as much time as you need, as this is a very important and big task. Don’t make any errors and work very precise step by step. If this is too much for your context window tell me to give you missing information.