In the next step I want to address some things I noticed at Supabase.

Warning 1:

Issue

View public.images_this_month is defined with the SECURITY DEFINER property

Description
Detects views defined with the SECURITY DEFINER property. These views enforce Postgres permissions and row level security policies (RLS) of the view creator, rather than that of the querying user

Warning 2:

Issue

Function public.monthly_image_count has a role mutable search_path

Description
Detects functions where the search_path parameter is not set.

Warning 3:

Issue

We have detected that you have enabled the email provider with the OTP expiry set to more than an hour. It is recommended to set this value to less than an hour.

Description
OTP expiry exceeds recommended threshold

Warning 4:

Issue

Supabase Auth prevents the use of compromised passwords by checking against HaveIBeenPwned.org. Enable this feature to enhance security.

Description
Leaked password protection is currently disabled.

I also show you the current schema of the database.

We use that SQL script to solve warning 1 and 2

```
-- =========================================
-- Purpose: Address Supabase warnings 1 & 2
--   (1) Make the images_this_month view run with caller permissions
--   (2) Pin function search_path for monthly_image_count
-- Notes:
--   - No table/data changes. Metadata only.
--   - Does NOT change Auth settings.
--   - Includes IF EXISTS guards and is fully reversible.
-- =========================================

BEGIN;

-- ---------- (1) View: images_this_month ----------
DO $$
BEGIN
  IF to_regclass('public.images_this_month') IS NOT NULL THEN
    -- Run the view with the querying user's permissions
    EXECUTE 'ALTER VIEW public.images_this_month SET (security_invoker = true)';

    -- Extra safety: prevent the planner from pushing predicates inside the view
    EXECUTE 'ALTER VIEW public.images_this_month SET (security_barrier = true)';

    -- ⚠️ Permissions: we leave existing GRANTs exactly as-is to avoid breaking anything.
    -- If you later want to lock it down to logged-in users only, you can uncomment:
    -- EXECUTE ''REVOKE ALL ON public.images_this_month FROM anon'';
    -- EXECUTE ''GRANT SELECT ON public.images_this_month TO authenticated'';
  END IF;
```

```
END
$$;
-- ---------- (2) Function: monthly_image_count ----------
-- We don't know the exact signature, so harden every overload named monthly_image_count.
DO $$
DECLARE
  f RECORD;
BEGIN
  FOR f IN
    SELECT p.oid,
           n.nspname,
           p.proname,
           oidvectortypes(p.proargtypes) AS argtypes
      FROM pg_proc p
      JOIN pg_namespace n ON n.oid = p.pronamespace
     WHERE n.nspname = 'public'
       AND p.proname = 'monthly_image_count'
  LOOP
    -- Pin search_path to avoid role-mutable resolution issues.
    EXECUTE format(
      'ALTER FUNCTION public.%I(%s) SET search_path = public, pg_temp',
      f.proname, f.argtypes
    );
    -- We intentionally do NOT change SECURITY INVOKER/DEFINER to avoid behavior changes.
  END LOOP;
END
$$;


COMMIT;
```