// lib/supabase.ts

import { createClient } from '@supabase/supabase-js'

const supabaseUrl = 'https://szzymjtvozgvcsyuamfh.supabase.co'

const supabaseAnonKey = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InN6enltanR2b3pndmNzeXVhbWZoIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NTc5Njc5OTUsImV4cCI6MjA3MzU0Mzk5NX0.GY9zY-6xV7KNyjom2M1QqRS\_-fgtWlGKxLPqJtEFhDY'

export const supabase = createClient(supabaseUrl, supabaseAnonKey)

// Helper function to get the current user

export const getCurrentUser = async () => {

const { data: { user }, error } = await supabase.auth.getUser()

if (error) {

console.error('Error getting user:', error)

return null

}

return user

}

// Helper function to sign in with Google

export const signInWithGoogle = async () => {

const { data, error } = await supabase.auth.signInWithOAuth({

provider: 'google',

options: {

redirectTo: `${window.location.origin}/dashboard`

}

})

if (error) {

console.error('Error signing in with Google:', error)

throw error

}

return data

}

// Helper function to sign out

export const signOut = async () => {

const { error } = await supabase.auth.signOut()

if (error) {

console.error('Error signing out:', error)

throw error

}

}

// Helper function to get session

export const getSession = async () => {

const { data: { session }, error } = await supabase.auth.getSession()

if (error) {

console.error('Error getting session:', error)

return null

}

return session

}