**Инструкция:**

1. **Зайди в Supabase** → https://supabase.com/dashboard  
   Открой свой проект.
2. В меню слева выбери:  
   **Authentication → Providers → Google**.
3. Там нужно указать:
   * **Client ID**
   * **Client Secret**  
     (берутся в Google Cloud Console).

**Как получить Client ID и Secret:**

1. Перейди в **Google Cloud Console**.
2. Создай новый проект (или выбери существующий).
3. Зайди в **APIs & Services → Credentials**.
4. Нажми **Create Credentials → OAuth Client ID**.
5. Выбери **Web application**.
6. Укажи **Authorized redirect URIs**:
7. http://localhost:3000/auth/callback

(для продакшена добавишь свой домен, например https://myblog.vercel.app/auth/callback).

1. Сохрани → получишь **Client ID** и **Client Secret**.

**В Supabase:**

* Вставь Client ID и Secret в форму Google Provider.
* Включи переключатель **Enable**.

**1. Войти в Google Cloud Console**

* Перейди: https://console.cloud.google.com
* Создай новый проект (**New Project**) или выбери существующий.

**2. Настроить OAuth Consent Screen**

1. В левом меню выбери: **APIs & Services → OAuth consent screen**.
2. Выбери **External** (если пользователи будут не только из твоей организации).
3. Заполни:
   * **App name** — название (например, My Blog).
   * **User support email** — твой email.
   * **Developer contact information** — тоже твой email.
4. Нажми **Save and Continue**.
5. На следующих шагах можно оставить пустым (Scopes и Test users) — пока просто **Continue** → **Back to Dashboard**.

**3. Создать OAuth 2.0 Credentials**

1. Перейди: **APIs & Services → Credentials**.
2. Нажми **Create Credentials → OAuth Client ID**.
3. В **Application type** выбери **Web application**.
4. Укажи имя (например, Supabase Auth).
5. Добавь **Authorized redirect URIs**:
   * Для локальной разработки:
   * http://localhost:3000/auth/callback
   * Для продакшена (например, если будешь деплоить на Vercel):
   * https://myblog.vercel.app/auth/callback
6. Нажми **Create**.

**4. Получить Client ID и Client Secret**

* После создания появится окно с:
  + **Client ID**
  + **Client Secret**
* Сохрани их → нужно вставить в **Supabase → Authentication → Providers → Google**.

**Какие Redirect URIs нужны?**

В **Google Cloud Console → Credentials → OAuth Client ID** → при создании (или редактировании) укажи такие URIs:

**Для локальной разработки:**

http://localhost:3000/auth/callback

**Для продакшена (например, если проект на Vercel):**

https://<твое-доменное-имя>/auth/callback

Например:

https://myblog.vercel.app/auth/callback

**Где это отразится?**

1. В Google Cloud Console — при настройке OAuth.
2. В **Supabase** — в разделе **Authentication → URL Configuration** (Callback URL). Там тоже нужно добавить **те же самые URI**.

⚡ Важно:

* Если хотя бы один символ отличается (например, /auth/callback vs /auth/callback/), Google выдаст ошибку.
* Для тестов можно оставить только http://localhost:3000/auth/callback.

Следующий шаг — проверить вход и выход пользователя.

🔑 Что нужно сделать:

Проверка входа

На странице /auth у тебя уже есть компонент <Auth />.

После успешного входа Supabase создаёт сессию.

Эту сессию можно получить так:

// src/app/page.tsx (пример на главной)

"use client"

import { useEffect, useState } from "react"

import { supabase } from "@/lib/supabase"

export default function HomePage() {

const [user, setUser] = useState<any>(null)

useEffect(() => {

// Получаем текущую сессию

supabase.auth.getSession().then(({ data }) => {

setUser(data.session?.user ?? null)

})

// Подписка на изменения (вход/выход)

const { data: authListener } = supabase.auth.onAuthStateChange(

(\_event, session) => {

setUser(session?.user ?? null)

}

)

return () => {

authListener.subscription.unsubscribe()

}

}, [])

return (

<div className="p-6">

<h1 className="text-2xl font-bold mb-4">Главная страница</h1>

{user ? (

<p>Привет, {user.email} 👋</p>

) : (

<p>Вы не авторизованы</p>

)}

</div>

)

}

👉 Теперь при входе будет выводиться e-mail авторизованного пользователя.

Выход из аккаунта

Кнопку выхода удобно сделать в Header.tsx:

"use client"

import { supabase } from "@/lib/supabase"

import { useRouter } from "next/navigation"

export default function Header() {

const router = useRouter()

const handleLogout = async () => {

await supabase.auth.signOut()

router.push("/auth") // после выхода перенаправляем

}

return (

<header className="flex justify-between items-center p-4 bg-gray-100">

<h1 className="text-xl font-bold">Блог</h1>

<button

onClick={handleLogout}

className="px-4 py-2 bg-red-600 text-white rounded"

>

Выйти

</button>

</header>

)

}

Что проверить

✅ Заходишь на /auth, логинишься через Google.

✅ На / видишь Привет, user@email.com.

✅ Нажимаешь "Выйти" → возвращает на /auth.

✅ Не забудь добавить Header в layout.