**План разработки блога**

**🔹 1. Инициализация проекта**

* Создать проект Next.js (App Router, TypeScript, Tailwind).
* Установить зависимости: @supabase/supabase-js, @supabase/auth-helpers-nextjs, @supabase/auth-ui-react.
* Настроить базовую структуру: /app, /components, /lib.

**🔹 2. Настройка Supabase**

* Создать проект в Supabase.
* Настроить базу данных: таблица posts (id, title, content, author, created\_at).
* Настроить переменные окружения (NEXT\_PUBLIC\_SUPABASE\_URL, NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY).
* Создать lib/supabase.ts для клиента.

**🔹 3. Авторизация (Google OAuth)**

* Включить Google OAuth в Supabase.
* Настроить Google Cloud Console (OAuth Consent Screen + Credentials).
* Добавить Redirect URIs.
* Создать страницу /auth с готовым Auth UI компонентом.
* Проверить вход/выход.

**🔹 4. Header с навигацией и сессией**

* Компонент Header (Главная | Админка | Войти/Выйти).
* Подписка на состояние сессии (supabase.auth.getSession).
* При входе — показывать email + кнопку «Выйти».
* При выходе — возвращать кнопку «Войти».

**🔹 5. Главная страница /**

* Получать список постов из posts.
* Выводить в виде карточек (title, content, created\_at).
* Никакой авторизации не нужно.

**🔹 6. Админка /admin**

* Защита маршрута: доступ только авторизованным пользователям (через middleware или server-side проверку).
* Таблица постов: id, title, created\_at.
* CRUD: кнопки «Добавить», «Редактировать», «Удалить».
* Формы для title + content.

**🔹 7. Тестирование и отладка**

* Проверить вход/выход через Google.
* Проверить доступ в /admin.
* Добавить пост → проверить, что он появился на главной.
* Редактировать/удалять пост.

**🔹 8. (Опционально) Роли и RLS**

* Включить Row-Level Security в Supabase.
* Добавить политику: только автор может редактировать/удалять свои посты.
* Ввести роль «admin» (хранить в profiles таблице).