**نمای کلی پروژه**

این پروژه یک برنامه وب ساده شبکه اجتماعی است که با استفاده از **Next.js** برای فرانت‌اند و بک‌اند نوشته شده و به زبان **TypeScript** توسعه یافته است. از **Prisma** به عنوان ORM و **PostgreSQL** به عنوان پایگاه داده استفاده می‌شود که بر روی **Supabase** میزبانی می‌شود. احراز هویت از طریق **Auth.js (**قبلاً با نام NextAuth.js شناخته می‌شد) انجام می‌شود و از کتابخانه **Shadcn UI** در کنار **Tailwind CSS** برای طراحی رابط کاربری استفاده می‌شود.

**فناوری‌های استفاده‌شده**

* **فرانت‌اند:**  Next.js (TypeScript)
* **بک‌اند:**  Next.js API Routes (TypeScript)
* **پایگاه داده:** PostgreSQL (via Supabase)
* **ORM**: Prisma
* **احراز هویت:** Auth.js (GitHub OAuth)
* **رابط کاربری:** Tailwind CSS, Shadcn UI

**ساختار پوشه‌ها**

ساختار پوشه‌ها بر اساس کنوانسیون‌های Next.js طراحی شده است.

**فناوری‌های استفاده‌شده**

**1. Next.js**

Next.js برای رندرینگ سمت سرور (SSR) و ایجاد بک‌اند API استفاده می‌شود. این فریم‌ورک امکانات توسعه سریعی مانند بارگذاری داغ و ایجاد آسان API‌ها از طریق مسیرهای API را فراهم می‌کند.

**2. TypeScript**

TypeScript تایپینگ قوی، بررسی خطا و تجربه بهتر برای توسعه‌دهندگان را تضمین می‌کند و امنیت نوعی بیشتری نسبت به جاوا اسکریپت فراهم می‌آورد.

**3. Prisma ORM**

Prisma برای تعاملات پایگاه داده استفاده می‌شود. این ORM کوئری‌های تایپ‌محور و مدیریت اسکیما برای پایگاه داده PostgreSQL را فراهم می‌کند.

**4. PostgreSQL (از طریق Supabase)**

PostgreSQL پایگاه داده رابطه‌ای است که برای ذخیره‌سازی داده‌های کاربر، پست‌ها و تعاملات استفاده می‌شود و **Supabase** نقش میزبان پایگاه داده را دارد.

**5. Auth.js (NextAuth.js)**

Auth.js برای احراز هویت از طریق OAuth استفاده می‌شود و GitHub تنها ارائه‌دهنده احراز هویت شخص ثالث است. این سیستم مدیریت sessionها و فرآیند احراز هویت کاربران را برعهده دارد.

**6. Tailwind CSS و Shadcn UI**

**Tailwind CSS** برای استایل‌دهی UI به روش utility-first استفاده می‌شود **Shadcn UI** مجموعه‌ای از کامپوننت‌های آماده و دسترس‌پذیر را ارائه می‌دهد که با روش Tailwind به خوبی همخوانی دارند.

**اسکیما پایگاه داده**

اسکیما پایگاه داده با استفاده از **Prisma** تعریف شده است. در زیر نمای کلی از مدل‌های اصلی ارائه شده است:

* **مدل کاربر: (User Model)** نمایانگر موجودیت کاربر است که شامل فیلدهایی مانند id، name،email و روابطی با پست‌ها، لایک‌ها و بوکمارک‌ها می‌باشد.
* **مدل پست: (Post Model)** نمایانگر پست‌های شبکه اجتماعی است که شامل فیلدهایی مانند title، content،author و روابطی با کاربرانی که پست‌ها را لایک یا بوکمارک می‌کنند، است.
* **مدل لایک: (Like Model)** تعاملات لایک را با ایجاد ارتباط میان کاربران و پست‌ها از طریق یک رابطه چند به چند ثبت می‌کند.
* **مدل بوکمارک:(Bookmark Model)** بوکمارک‌های پست‌ها را با ایجاد ارتباط میان کاربران و پست‌ها ثبت می‌کند.

**احراز هویت با GitHub (Auth.js)**

این اپلیکیشن از **Auth.js** برای احراز هویت مبتنی بر OAuth GitHub استفاده می‌کند. این سیستم مدیریت session و ورود کاربران از طریق جریان OAuth GitHub را بر عهده دارد. تنظیمات اصلی شامل پیکربندی GitHub به عنوان ارائه‌دهنده OAuth و مدیریت توکن‌های session می‌باشد.

**مسیرهای API**

مسیرهای API در Next.js برای مدیریت بک‌اند، پست‌ها، مدیریت کاربران و تعاملات استفاده می‌شوند. این مسیرها عملیاتی مانند ایجاد پست، دریافت داده‌های کاربران و مدیریت لایک‌ها یا بوکمارک‌ها را انجام می‌دهند. هر مسیر API در یک مسیر مشخص فعالیت می‌کند و با استفاده از Prisma برای ذخیره‌سازی یا بازیابی داده‌ها با پایگاه داده تعامل دارد.

**استایل‌ها و UI**

رابط کاربری با استفاده از کامپوننت‌های**Shadcn UI**  و با استایل‌دهی **Tailwind CSS** ساخته شده است.

**انتشار (Deployment)**

* **پایگاه داده:** میزبانی شده بر روی **Supabase** (PostgreSQL).
* **اپلیکیشن**: منتشر شده بر روی **Vercel** که برای پروژه‌های**Next.js** بهینه است.
* **متغیرهای محیطی:** در محیط انتشار (داشبورد Vercel یا فایل .env) تنظیم می‌شوند.

**نتیجه‌گیری**

این مستندات ساختار، تکنولوژی‌های استفاده‌شده و جزئیات طراحی برنامه وب شبکه اجتماعی را شرح می‌دهد. این برنامه با **TypeScript**،**Next.js** و**Prisma** ساخته شده است، با استفاده از **Tailwind CSS** و **Shadcn UI** برای طراحی رابط کاربری. احراز هویت از طریق GitHub انجام می‌شود و پایگاه داده بر روی **Supabase** میزبانی می‌شود. این ساختار برای مقیاس‌پذیری، امنیت و نگهداری آسان طراحی شده است.