

## مستند پروژه فروشگاه آنلاین (ShopX)

### 1. هدف پروژه

- ایجاد یک فروشگاه آنلاین ساده و پایدار با قابلیت مدیریت محصولات و سفارش‌ها.
- مناسب برای کاربران غیر فنی (مثلاً یک فروشنده کوچک یا نوجوان) که بتوانند بدون دانش برنامه‌نویسی فروشگاه خود را راه‌اندازی کنند.
- تمرکز بر سادگی استفاده برای کاربر نهایی و تمیزی معماری برای توسعه‌دهندگان.

### 2. پیشرفت‌های پروژه تا اینجا

- مرحله اول: ساخت صفحات اصلی (محصولات، سبد خرید، تسویه حساب).
- مرحله دوم: افزودن پنل ادمین برای مدیریت محصولات.
- مرحله سوم: پیاده‌سازی API داخلی با Next.js برای محصولات.
- مرحله چهارم: اتصال به دیتابیس واقعی. (Prisma + SQLite)
- مرحله پنجم: افزودن سیستم سفارش‌ها. (Order + OrderItem)
- مرحله ششم: ساخت پنل ادمین برای مدیریت سفارش‌ها و تغییر وضعیت آن‌ها.

### 3. تکنولوژی‌ها و سیستم‌های استفاده‌شده

- Next.js 14 (App Router) برای ساخت Frontend و Backend در یک پروژه.
- Prisma ORM برای مدیریت دیتابیس و مدل‌سازی داده‌ها.
- SQLite دیتابیس پیش‌فرض برای توسعه محلی (قابل ارتقا به PostgreSQL/MySQL).
- TailwindCSS برای طراحی رابط کاربری مدرن و واکنش‌گرا.
- LocalStorage برای ذخیره‌سازی سبد خرید در سمت کاربر.
- API داخلی: Next.js برای مدیریت محصولات و سفارش‌ها (CRUD کامل).

### 4. معماری پروژه

shopx-frontend/

prisma/

schema.prisma  
src/  
lib/  
prisma.ts  
cart.ts  
app/  
products/page.tsx  
cart/page.tsx  
checkout/page.tsx  
admin/  
login/page.tsx  
register/page.tsx  
products/page.tsx  
orders/page.tsx  
api/  
products/route.ts  
products/[id]/route.ts  
orders/route.ts  
orders/[id]/route.ts  
checkout/route.ts

---

#### 5. مدل داده‌ها

- **Product** شامل نام، قیمت، تصویر، تاریخ ایجاد/ویرایش.
  - **Order** شامل مشتری، شماره تماس، آدرس، مجموع مبلغ، وضعیت (جدید، پرداخت شده، ارسال شده، لغو).
  - **OrderItem** شامل محصول، تعداد، قیمت واحد.
-

## 6. گردش کار (Workflow)

1. ادمین وارد پنل می‌شود و محصولات را اضافه/ویرایش/حذف می‌کند.
2. مشتری وارد سایت می‌شود و محصولات را در صفحه `/products` می‌بیند.
3. مشتری محصول را به سبد خرید اضافه می‌کند.
4. در صفحه سبد خرید، تعداد محصولات را مدیریت می‌کند.
5. در صفحه Checkout، اطلاعات خود را وارد کرده و سفارش ثبت می‌شود.
6. سفارش در دیتابیس ذخیره می‌شود و در پنل ادمین `/admin/orders` نمایش داده می‌شود.
7. ادمین می‌تواند وضعیت سفارش را تغییر دهد (پرداخت شد، ارسال شد، لغو شد).

---

## 7. نحوه توسعه پروژه

- راه‌اندازی اولیه :
- `npm install`
- `npx prisma migrate dev --name init`
- `npm run dev`
- مدیریت دیتابیس:
- با دستور `npx prisma studio` می‌توان داده‌ها را مشاهده و مدیریت کرد.
- توسعه ماژولار:
- هر بخش (محصولات، سفارش‌ها، سبد خرید) در پوشه جداگانه توسعه داده شده است.
- Client/Server Components:
  - Server Components برای دریافت داده‌ها.
  - Client Components برای تعامل (دکمه‌ها، فرم‌ها).

---

## 8. امکانات فعلی

- مدیریت محصولات (CRUD) کامل.
- نمایش محصولات به مشتری.
- سبد خرید با `LocalStorage`.
- ثبت سفارش واقعی در دیتابیس.

- پنل ادمین برای مدیریت سفارش‌ها و تغییر وضعیت.
- 

## 9. مسیرهای اصلی

- مشتری :

○ → `products`/لیست محصولات

○ → `cart`/سبد خرید

○ → `checkout`/ثبت سفارش

- ادمین :

○ → `admin/register`/ثبت‌نام مدیر

○ → `admin/login`/ورود مدیر

○ → `admin/products`/مدیریت محصولات

○ → `admin/orders`/مدیریت سفارش‌ها

---

## 10. پیشنهاد های آینده (Roadmap)

- اتصال به درگاه پرداخت زرین‌پال یا PayPal
  - سیستم احراز هویت واقعی + JWT نقش‌ها.
  - آپلود تصویر محصولات روی فضای ابری.
  - نوتیفیکیشن لحظه‌ای برای سفارش‌های جدید.
  - بهینه‌سازی SEO و چندزبانه‌سازی.
- 

## 🔮 نتیجه‌گیری

این پروژه یک فروشگاه آنلاین کامل و پایدار است که از صفر تا صد مسیر خرید و فروش را پوشش می‌دهد. ساختار ماژولار و استفاده از ابزارهای مدرن باعث می‌شود توسعه‌دهندگان بتوانند به راحتی آن را نگهداری و ارتقا دهند.

---

♦ محمد، این متن رو مستقیم کپی کن داخل Word چون با تیتراهای شماره‌دار و بولت‌پوینت نوشته شده، توی Word خیلی مرتب و تمیز درمیا.

می‌خواهی من همین متن رو بهت یک چک‌لیست اجرایی هم تبدیل کنم) مثلاً برای (Developer Onboarding) تا هر برنامه‌نویسی که پروژه رو می‌گیره دقیق بدونه از کجا شروع کنه؟