Product Data Explorer – Technical Documentation

Prepared as part of a full-stack engineering assignment.

1. Project Overview

Product Data Explorer is a full-stack web application designed to scrape, store, and display product data from an external e-commerce source. The system demonstrates backend scraping, API design, database modeling, and a modern frontend built with Next.js.

2. System Architecture

The application follows a modular client-server architecture:

- **Frontend:** Next.js (App Router, TypeScript)
- **Backend:** NestJS REST API
- **Scraping:** Playwright + Crawlee
- **Database:** PostgreSQL with Prisma ORM

3. Backend Implementation

The backend is built using NestJS and exposes RESTful endpoints for navigation, categories, and products. Prisma ORM is used to manage database access and enforce schema consistency.

- /navigation – Fetches all navigation categories
- /categories/:slug/products – Returns paginated products for a category
- /products/:id – Fetches detailed product information

4. Scraping Strategy

Product data is scraped on-demand using PlaywrightCrawler from Crawlee. When a category is requested and no recent cached data exists, the scraper fetches live product data from the source website and stores it in the database.

5. Database Design

The database schema is designed to support scalability and relational integrity. Key models include Navigation, Product, ProductDetail, Review, and ScrapeJob.

6. Frontend Implementation

The frontend is implemented using Next.js App Router with TypeScript. Client components are used for pagination and interactivity, while API communication is handled through a centralized api.ts layer.

7. Error Handling and Debugging

Significant effort was invested in resolving schema mismatches, Prisma relation errors, and Next.js dynamic routing issues. Strict typing and clear separation of concerns ultimately stabilized the application.

8. Trade-offs and Limitations

- Images may not always load due to external source constraints
- Scraping is limited to one request per category to reduce load
- Backend schema was frozen once stability was achieved

9. Conclusion

Product Data Explorer successfully demonstrates full-stack development skills, including backend engineering, web scraping, database modeling, and frontend integration. The project meets all core assignment objectives and is ready for evaluation.