

# Project Elara

**Vision:** Project Elara is a next-generation AI-powered e-commerce platform designed to merge shopping convenience, personalization, and intelligent product discovery. By leveraging Django for the backend and React for the frontend, it provides scalability, security, and seamless user experience.

## Technical Architecture

- Backend: Django + Django REST Framework (API-driven, modular apps for products, users, orders, and AI assistant).
- Frontend: React for modern, responsive UI with a personalized shopping experience.
- Database: PostgreSQL (primary store for products, users, and orders).
- AI Assistant: Connected via Django service layer integrating with AI models (OpenAI, HuggingFace, or custom).
- Deployment: Docker containers orchestrated with Kubernetes or Railway.
- Static Content: Managed by Django staticfiles or served via CDN for performance.

## Proposed Folder Structure

```
project-elara/ ├── database/ → Database scripts and migrations ├── djangoapp/ → Django apps (products, users, orders, assistant) ├── djangoproj/ → Core Django project (settings, urls, wsgi, asgi) ├── frontend/ → React frontend (user interface) ├── static/ → Static assets (CSS, images, JS) ├── Dockerfile → Build instructions ├── deployment.yaml → Kubernetes/Railway deployment config ├── entrypoint.sh → Startup script ├── manage.py → Django management CLI ├── package.json → Frontend dependencies └── requirements.txt → Backend dependencies
```

## Django App Modules

- Products: Models, APIs, and business logic for managing catalog and categories.
- Users: Authentication, profiles, and roles.
- Orders: Cart, checkout, and payment integration.
- Assistant: AI-driven shopping helper with personalized recommendations.

## Development Roadmap

- Phase 1: Setup Django project, database, and Docker environment.
- Phase 2: Implement Products and Users apps with API endpoints.
- Phase 3: Add Orders and payment gateway integration.
- Phase 4: Develop AI Assistant with conversational interface.
- Phase 5: Integrate React frontend with Django APIs.
- Phase 6: Deployment and scaling (Kubernetes, Railway, or AWS).