Project Nexus — E■Commerce Backend (ProDev BE)

Author: [Your Name] | GitHub: https://github.com/Tsigie-beyene/alx-project-nexus-

Links: Repo • Live API • Swagger • ERD • Demo Video

Problem & Goal

- Problem: Power product discovery and management at scale
- Users: Shoppers, Admins
- Goal: Secure, performant, documented backend with clean APIs

Solution Overview

- CRUD: products, categories, users (JWT auth)
- Discovery: filtering, sorting, pagination
- Quality: indexing, query optimization, OpenAPI docs

Tech Stack & Tools

- Backend: Django, DRF
- DB: PostgreSQL
- Auth: JWT (access/refresh)
- Docs: Swagger/OpenAPI
- DevEx: pre-commit, Black, Flake8, isort
- CI/CD: GitHub Actions
- Hosting: Railway/Render/Heroku/VPS
- Observability: Admin, Logs, Health

Architecture (High Level)

- Client → Django/DRF → PostgreSQL
- Optional: Redis for caching/rate limiting
- Static/admin: Django Admin
- CI: test → lint → migrate → deploy

ERD — Core Data Model

User

- id (PK)
- email (unique)
- password_hash
- is_staff

Category

- id (PK)
- name (unique)
- slug (unique)

Product

- id (PK)
- name
- description
- price
- stock
- category_id (FK)
- created_at
- updated_at

Relationships: Category 1—N Product | User used for auth/admin

Database Design Decisions

- Normalization: 3NF; product belongs to one category
- Constraints: unique (category.name, category.slug), NOT NULL on critical fields
- Indexes: product(category_id, price, created_at)
- Optional: trigram/GIN for search on product.name

API Design (REST)

- Base: /api/v1/
- Auth: POST /auth/login → { access, refresh }
- Categories: GET/POST /categories/, GET/PATCH/DELETE /categories/{id}/
- Products: GET/POST /products/, GET/PATCH/DELETE /products/{id}/
- List params: ?category=<id>&ordering=-price&page=1&page_size=12

Filtering, Sorting, Pagination

- Filters: ?category=<id>
- Sorting: ?ordering=price or ?ordering=-price
- Pagination: ?page=2&page_size=20 (defaults and max limits)
- Example: /api/v1/products?category=3&ordering=-price&page=1&page_size=12

Authentication & Authorization

- JWT: login returns access + refresh
- Password hashing: PBKDF2/Argon2
- Permissions: staff/admin required for write; read open
- Optional: throttling for sensitive endpoints

Performance & Optimization

- ORM: select_related('category') for product lists
- Avoid N+1 with prefetch_related
- Index verification with EXPLAIN ANALYZE
- Optional caching: Redis for popular lists

Documentation & Developer Experience

- Swagger/Redoc (OpenAPI)
- README: setup, run, env, auth flow, endpoints
- Postman collection (optional)
- Pre-commit: Black, Flake8, isort

Testing & Code Quality

- Unit: serializers, utils
- Integration: endpoints (auth, products, categories)
- Coverage target: 80%+
- CI: tests, lint, migrations check

Deployment

- Hosting: Platform + Gunicorn
- Environment: secrets via env vars
- Health: /health (200 OK)
- Run migrations on release

Live Demo (≤5 minutes)

- Login: get JWT tokens
- Create Category; Create Product (with category)
- List Products: filter, sort, paginate
- Update/Delete Product (permission gate)

Rubric Mapping

- Functionality: CRUD + auth + discovery
- Code Quality: readable, docs, linters
- Design & API: normalized ERD, RESTful endpoints, DRF ORM
- Deployment & Best Practices: live, secure, documented

Challenges & Learnings

- Pagination + ordering performance
- Auth edge cases and JWT rotation
- EXPLAIN plans and index choice
- OpenAPI workflows

Roadmap

- Features: search, inventory, orders/cart, reviews
- Performance: Redis cache, selective indexes
- Security: 2FA, admin audit logs, rate limits
- Platform: staging, blue/green, observability

Links & Access

- Repo: https://github.com/Tsigie-beyene/alx-project-nexus-
- Live API: [URL]
- Swagger/Redoc: [URL]
- ERD (Google Doc): [URL – anyone with link can view]
- Demo video (≤5 min): [URL]
- Slide deck: [Google Slides link]
- Health: [URL]/health