



E-Commerce Backend API Development



Data Modeling & Logic

Relational Schema: Created a structured database using PostgreSQL/SQLite to handle e-commerce data efficiently.

Key Entities:

Category Table: Stores product classifications (e.g., Electronics, Smartphones).

Product Table: Stores detailed information (Name, Price, Description, Timestamp).

Data Integrity: Implemented a One-to-Many relationship between Categories and Products.

Business Rules: Ensures every product is linked to a valid category using Foreign Keys, preventing orphaned data.



API Features



- Full CRUD Operations: Support for Creating, Reading, Updating, and Deleting Products and Categories.



- Advanced Filtering detargetnl :
retlif ot sresu wolla ot retlif-ognajd
ni nwohs sa) DI yrogetaC yb stcudorp
med ehto).



- Search & Ordering hcraes delbanE :
dna eman tcudorp yb ytilanoitcnuf
.etad ro ecirp yb gniredro



Security & Authentication



JWT Authentication: Implemented JSON Web Tokens (SimpleJWT) to provide a secure and stateless authentication mechanism.

Granular Permissions: Used `IsAuthenticatedOrReadOnly` to ensure that only authorized users can modify data (Create/Update/Delete), while others can view it.

Token Management: Established a secure flow for obtaining and refreshing tokens to maintain user sessions safely.

Protection against Unauthorized Access: All sensitive endpoints are protected, returning 401 Unauthorized for unauthenticated requests





API Documentation & Tools

- **Swagger & Redoc:** Integrated drf-yasg to generate interactive API documentation for easier testing and front-end integration.
- **Live Testing:** Used the Swagger UI to perform real-time requests and validate API responses (as demonstrated in the video).
- **Development Stack:**
 - Framework: Django Rest Framework (DRF).
 - Database: SQLite (Development) / PostgreSQL (Production).
 - Tools: Postman/Swagger for testing, Git for version control.



Future Improvements & Conclusion



- **Scalability:** The API is designed to be easily extendable to include Features like User Profiles, Orders, and Payments.
- **Enhanced Features:** Plans to add Image Uploads for products and Email Notifications for new orders.
- **Testing:** Future implementation of Unit Testing to ensure 100% code reliability.
- **Final Thought:** Successfully delivered a secure, documented, and fully functional E-commerce backend that meets all Project Nexus requirements.

