

# ■ README — E-Commerce Pets Backend

## 1. Overview

This backend project provides a REST API to manage users, products, carts, and orders for a pet e-commerce platform. It uses Flask, PostgreSQL, Redis (for caching), and JWT for authentication.

## 2. Requirements

- Python 3.11+
- PostgreSQL 14+
- Redis 6+ (optional, for caching)
- Git

## 3. Setup the Database

1. Create a PostgreSQL database and schema:

```
CREATE DATABASE ecommerce;  
\c ecommerce;  
CREATE SCHEMA "ECommercePets";
```

2. Configure environment variable for DB connection:

```
export DATABASE_URL=postgresql://user:password@localhost:5432/ecommerce
```

3. Apply migrations:

```
alembic upgrade head
```

## 4. Environment Variables

Create a `.env`` file with the following keys:

```
FLASK_APP=app:app  
FLASK_ENV=development  
DATABASE_URL=postgresql://postgres:postgres@localhost:5432/ecommerce  
JWT_SECRET_KEY=super_secret_key  
REDIS_URL=redis://localhost:6379/0
```

## 5. Run the Server

Start the Flask server:

```
flask run
```

The API will be available at:

```
http://127.0.0.1:5000
```

## 6. Verify Database Connection

To confirm the DB is connected:

- Ensure PostgreSQL is running locally.
- Use `psql` to connect manually with the same credentials as in DATABASE\_URL.
- If connection fails, check credentials and schema name.

## 7. Testing

Run tests with:

```
pytest -q
```

Run tests with coverage:

```
pytest --cov=.
```

## 8. Common Endpoints

- POST /register → Create a new user
- POST /login → Authenticate and get JWT
- GET /products → Retrieve product list (cached)
- POST /carts/{id}/checkout → Complete a purchase

## 9. Project Structure

Proyecto\_Final/

■■■ app.py

■■■ models.py

■■■ config.py

■■■ requirements.txt

■■■ /auth

■■■ /products

■■■ /carts

■■■ /tests

■■■ /migrations