E■Commerce Store (Cart + Payments + Admin Panel)

Overview

This project is a complete e■commerce web application built with Django. It allows users to browse products, add them to a shopping cart, complete orders through a payment gateway, and lets administrators manage inventory, orders, and customers from an integrated admin panel.

Key Features

- User authentication (login, signup, profile).
- Product catalog with search and filters.
- Shopping cart with quantity updates.
- Order checkout and payment integration (test mode).
- Admin panel for managing products, categories, and orders.
- Order history and email notifications.

Project Structure

```
project_root/
  manage.py
  store/
    models.py
    views.py
    urls.py
  cart/
  orders/
  templates/
  static/
```

Installation & Setup

1 Install Python 3.10+ and pip.

2 Create a virtual environment: python -m venv venv

3 Activate it: (Windows) venv\Scripts\activate | (Mac/Linux) source venv/bin/activate

4 Install dependencies: pip install -r requirements.txt

5 Run migrations: python manage.py migrate

6 Create superuser: python manage.py createsuperuser

7 Start server: python manage.py runserver

8 Open http://127.0.0.1:8000 in your browser.

How Payments Work (Test Mode)

Payments are processed using a sandbox (test) payment gateway integration. When users check out, the order is created, and a simulated payment page confirms the transaction. No real money is involved in development mode.

Admin Panel

Visit */admin* and log in with your superuser account. From there, you can add products, manage categories, review customer orders, and mark orders as shipped or delivered.

Example Use Case

- A user browses the catalog and adds two items to the cart.
- They proceed to checkout, provide address details, and place the order.
- The payment gateway confirms payment, and the user receives an order summary.
- The admin later views the order and updates its status.

Screenshots (Mock Layout Preview)

[Home Page Preview]
[Product Details Page Preview]
[Shopping Cart Preview]
[Admin Panel Preview]

What This Project Teaches

- Building real■world Django apps with multiple modules.
- Working with models, views, templates, and forms.
- Implementing authentication and authorization.
- Integrating payments and handling orders securely.
- Using Django Admin as a powerful CMS-like backend.

Next Improvements

- Add product reviews and ratings.
- Implement coupons and discounts.
- Add real payment gateway keys in production.
- Deploy to a cloud platform (Render, Railway, AWS, etc.).