CommerCast – Django E-commerce Project

Project Overview

CommerCast is a professional e-commerce web application built using Django. It aims to provide users with a seamless online shopping experience similar to real-world e-commerce platforms. The project demonstrates key aspects of web application development, including authentication, database modeling, secure payments (conceptual), user-friendly interfaces, and robust backend management.

Key Features

User Authentication:

Register, login, and manage accounts securely.

Password hashing with Django's built-in authentication system.

Product Catalog:

Browse products with categories, descriptions, stock, and pricing.

Product detail pages show all relevant information, images, and availability.

Shopping Cart:

Add, update, or remove products from cart.

Cart persists for logged-in users.

Checkout and Order Placement:

Enter/select delivery address and phone number.

Final order review before confirming purchase.

Checkout success page with order summary/confirmation.

Order History:

View a list of past orders for every user.

Order details include products, address, and status.

Admin Product Management:

Superusers can add, edit, or remove products and categories using Django admin.

Bulk product upload options supported via Django admin or custom scripts.

Responsive, Clean UI:

Uses Django's template system and Bootstrap for a modern look.

Mobile-friendly and accessible design.

How It Works:

Users can browse products on the homepage or by category.

Product Detail View: Clicking a product shows all details.

Add to Cart: User adds items, cart total automatically calculated.

Checkout:

User selects delivery address (can edit/add address).

Confirms phone number.

Reviews and submits the order.

Order Confirmation: Shows “Thank you for your order!” message and order details.

Order History: Users see all previous orders from their account page.

Technology Stack
Backend: Django (Python)

Frontend: HTML, CSS, Bootstrap (via Django templates)

Database: SQLite (for local dev, easily switchable to PostgreSQL/MySQL)

Authentication: Django built-in User model

Admin Interface: Django admin (robust back-office product management)