Multi-Tenant Ordering System

Overview:
    Build a Django-based multi-tenant application where each company manages its own orders and products.
    Implement soft-deletion and enforce appropriate access controls.

Baseline Requirements

Data Models:

    Company:
        name
    Extend User Class:
        Belongs to exactly one Company
        Role: admin | operator | viewer
    Product:
        - company, name, price, stock
        - created_by, created_at (immutable)
        - last_updated_at
        - is_active
    Order:
        - company (inferred via user)
        - product, quantity
        - created_by, created_at (immutable)
        - status (pending, success, failed)
        - shipped_at (set when status = success)

URLs & Views:
    Pages:
        / — index page
    APIs:
        GET /api/products/ — List active products for the user’s company
        POST /api/orders/ — Create one or more orders
        DELETE /api/products/ — Soft-delete one or more products
        GET /api/orders/export/ — Export company’s orders (CSV)

    Data Isolation
        Data is scoped to the authenticated user’s Company.

Django Admin:
    Products Admin:
        Bulk action: mark selected products inactive
    Orders Admin:
        Action: export orders as CSV

Simple HTML Interface (index page):
    Product creation form (auto-fill company, user, timestamp)
    Below the form, a table listing added products

Constrains:
    -Orders may include only products where is_active = True.
    -Prevent order creation if requested quantity exceeds available stock.
    -viewer users cannot place orders.
    -operator users may edit only orders created today.
    -On order success simulate send a confirmation email to the order’s creator by logging to file with appropriate details.

Tech Stack:
    project is fully dockerized
    - mysql 5.7
    - django 4.x.X
    - python 3.10 or up

- use any extra packages needed
- serve the application on 0.0.0.0

Deliverables:

A GitHub repository or ZIP of the Django project
README with setup & run instructions
add demo data for testing