🚀 PHASE 1: Project Setup & Core Setup

Step What We'll Do

1. Create the Django project and core app

2. Set up base templates (base.html, navbar, etc.)

3. Add static files and global layout styling

4. Configure URLs and settings (templates, static, etc.)

✅ We can start here or skip if already done.

🔐 PHASE 2: User Authentication

Feature Includes

Register Username, email, password

Login/Logout Login form, logout button

Roles Admin vs. Employee permissions

Access Control Certain views only available to admin

We’ll use Django's built-in User model and optionally extend it.

📦 PHASE 3: Inventory Management

Feature Includes

Product Model Name, category, stock, reorder level, unit cost

Category Model (optional) for grouping

CRUD Views Add/Edit/Delete/View

Low Stock Alerts Show on dashboard

👥 PHASE 4: Customer Management

Feature Includes

Customer Model Name, email, phone, company, address

Interaction Logs (Optional) notes or messages

CRUD Views Add/Edit/Delete/View

🏭 PHASE 5: Supplier Management

This is the one we already started! We'll:

Finalize the Supplier model

Add proper forms for Add/Edit

Connect with inventory later (purchase orders)

🧾 PHASE 6: Sales & Purchase Orders

Feature Includes

Sales Order Linked to customer, product(s), date, total

Purchase Order Linked to supplier, product(s), date, total

Order Tracking Status: pending, paid, delivered

CRUD Full control over orders

🧾 PHASE 7: Invoice Generation

Feature Includes

PDF Invoice Use xhtml2pdf or WeasyPrint

Fields Items, quantity, price, total, date, client

Download/Print Button to export invoice

📊 PHASE 8: Dashboard

Feature Includes

KPIs Total sales, top-selling products, low stock

Charts Bar/line/pie using Chart.js

Role-Based Show only what the user should see

📤 PHASE 9: Reports & Export

Feature Includes

Export Sales, inventory, customer data

Format CSV or Excel using pandas or csv

Filters Date range, product, etc.