# **React + Django Assignment Documentation**

## **🧩 Project Overview**

This assignment involves building a **React Dashboard** with a Django backend. The dashboard will visualize order and product analytics data and include interaction features such as importing orders, approving selected inventory items, and viewing analytics with an assistant bot.

## **🔧 Local Setup**

### **1. Clone the Repository**

git clone <your-github-repo-url>

cd <project-directory>

### **2. Backend (Django)**

#### **Install Dependencies**

cd backend

python -m venv venv

source venv/bin/activate # On Windows: venv\Scripts\activate

pip install -r requirements.txt

#### **Run Migrations**

python manage.py migrate

#### **Start the Server**

python manage.py runserver

API Base URL: http://localhost:8000/api/

### **3. Frontend (React)**

cd frontend

npm install

npm start

## **📊 Dashboard UI Structure**

### **🔝 Navbar**

* Logo
* Date display
* Refresh button (triggers data refresh)
* User profile icon

### **🔢 Overview Boxes**

* Fulfilled Orders (e.g. ✔️ 52 Fulfilled Orders)
* Items Sold (e.g. 📦 1169 Items Sold)
* Pending Orders (e.g. ⏳ 16 Pending Orders)

### **📝 Pending Orders Section**

* **API**: Uses /getorders endpoint
* **Filters**: Company Name and Priority
* **Refresh Button**: Triggers fetch for updated order data
* **Table Columns**:
  + Order ID (clickable to show order details)
  + Order Date
  + Priority
* **Import Orders CTA**: File input + API call to import orders

### **📌 Priority Items Section**

* **Description**: Two-line overview
* **Table Columns**:
  + Inventory ID
  + Product Name
  + Stock Status
  + Analytics Dashboard (CTA redirects to /analytics/inventory\_id)

### **📦 Order Details Section**

* **API**: Uses /getorderdetails endpoint
* Hidden by default
* Visible on row click in Pending Orders
* Maps full order data
* Includes a nested table with:
  + Checkbox (for approving)
  + Inventory ID
  + Product Name
  + Status
  + Analytics (CTA redirects to /analytics/inventory\_id)
  + **Additional Fields**: The API provides many parameters; include extra columns in the table for relevant fields like Analytics and checkbox.
* **Approve Selected Orders**:
  + Visible when one or more checkboxes selected
  + Calls the /approveorders API with selected items
  + Shows toast on success/error

## **📈 /analytics/:id Page**

### **📋 Overview**

* **APIs Used**:
  + /forcast: For fetching forecast analytics data
  + /chat: Backend endpoint used by the assistant bot for product-related queries
* Route: /analytics/:inventory\_id
* Page Title: "Product Analytics"

### **🔍 Features**

* **Dropdown**: To switch products
* **Data Display**:
  + Current Stock
  + Additional Forecasted Demands
  + Next Reorder Date
* **Line Chart**:
  + X-Axis: Month/Year
  + Y-Axis: Quantity
  + Lines: Manufacturer, Current Stock, Sales

### **🤖 Assistant Bot Panel**

* Users can ask questions related to **vendors/suppliers** and **products**.
* If the user requests: "Generate email for me to send vendor", the chatbot generates the email body directly in the chat.
* Once the email is generated in the response, the **Generate Email** button becomes enabled.
* Clicking the **Generate Email** button:
  + Displays the generated email
  + Redirects the user to their mailbox with the email content pre-filled
  + Visible when email info is present in bot response
  + Generates and displays an email
  + Redirects to the user mailbox with email ready to send