Perfect 👍 This is a **serious, production-level architecture** — not just a hackathon demo.

I'll give you a **clear execution blueprint** so you can:

- Divide work across your 7-member team
- Create models correctly
- Design CrewAI agents properly
- Build it production-ready
- Avoid chaos

🔷 OVERALL SYSTEM ARCHITECTURE

Backend:

- Django + PostgreSQL
- Celery (for background AI + email monitoring)
- CrewAI (agent orchestration)
- Email integration (IMAP + SMTP)

Frontend:

- React (Dashboard + AI Assistance Panel)

AI Layer:

- CrewAI Agents
- Tools (Email tool, DB tool, Web search tool, Recommendation tool)

🔷 PHASE WISE EXECUTION PLAN

✅ PHASE 1 – CORE FOUNDATION (Day 1–2)

1️⃣ Authentication System

Models

🔹 User

(Use Django default user with email login)

Fields:

- email (unique)
- is_verified
- role (ADMIN / OWNER / QUALITY / LOGISTICS)

2️⃣ Company Model

```
Company
- id
- user (FK)
- name
- address
- location
- business_type
- business_description
```

```
- created_at
```

✅ PHASE 2 – SUPPLY CHAIN CORE DATABASE STRUCTURE

This is CRITICAL.

3️⃣ Inventory Model

```
InventoryItem
- id
- company (FK)
- name
- description
- category
- current_stock
- max_stock
- safety_stock
- unit (kg, pcs, ton)
- reorder_threshold_percentage (default 25)
- created_at
```

4️⃣ Vendor Model

```
Vendor
- id
- company (FK)
- name
- email (unique)
- phone
- address
- rating
```

```
- supplies_items (ManyToMany -> InventoryItem)
- delivery_capability
- average_delivery_days
- is_active
- created_at
```

5️⃣ VendorQuotation Model

```
VendorQuotation
- id
- vendor (FK)
- inventory_item (FK)
- price_per_unit
- discount
- delivery_days
- payment_terms
- email_thread_id
- status (PENDING / ACCEPTED / REJECTED)
- created_at
```

6️⃣ PurchaseOrder Model

```
PurchaseOrder
- id
- vendor (FK)
- inventory_item (FK)
- quantity
- total_cost
- expected_delivery_date
- status (CREATED / CONFIRMED / CANCELLED / DELIVERED)
- email_thread_id
- created_at
```

7️⃣ ReceivedInventory Model (Quality Check)

```
ReceivedInventory
- id
- purchase_order (FK)
- received_date
- quantity_received
- quality_status (PENDING / APPROVED / REJECTED)
- rejection_reason
- replacement_requested (boolean)
```

8️⃣ FinishedProduct Model

```
FinishedProduct
- id
- company (FK)
- name
- current_stock
- safety_stock
- unit
```

9️⃣ CustomerOrder Model

```
CustomerOrder
- id
- customer_email
- product (FK FinishedProduct)
- quantity
- requested_delivery_date
- status (PENDING / CONFIRMED / REJECTED)
- email_thread_id
```

🔷 PHASE 3 – CREW AI AGENT DESIGN

This is where your project becomes powerful.

🎯 AGENT ARCHITECTURE

You should create 8–10 agents.

1️⃣ Inventory Monitoring Agent

Responsibility:

- Monitor stock levels
- Trigger vendor sourcing when below 25%
- Notify admin

Trigger:

- Celery scheduled job every 10 minutes

2️⃣ Vendor Discovery Agent

Responsibility:

- Search internet
- Extract vendor details
- Check Google ratings
- Add to DB

Tools:

- Web search tool

- DB write tool

3️⃣ Vendor Communication Agent

Responsibility:

- Send inquiry emails
- Generate structured RFQ email
- Track email thread ID

4️⃣ Email Reader & Analyzer Agent

Responsibility:

- Read incoming emails
- Extract:
    - Price
    - Delivery days
    - Discount
    - Rejection
- Update VendorQuotation table

Tool:

- IMAP tool
- LLM extraction tool

5️⃣ Negotiation Agent

Responsibility:

- Compare vendors
- Negotiate price via email
- Recommend best vendor

Decision parameters:

- Delivery time (Priority 1)
- Cost
- Rating
- Discount

6️⃣ Purchase Order Agent

Responsibility:

- Generate PO
- Send final order email
- Ask for payment details
- Await confirmation

7️⃣ Quality Monitoring Agent

Responsibility:

- Monitor ReceivedInventory
- If REJECTED:
    - Email vendor
    - Inform admin
    - Restart vendor sourcing

8️⃣ Finished Product Monitoring Agent

Responsibility:

- Monitor stock
- Notify admin if low
- Suggest increase production

9️⃣ Customer Order Handling Agent

Responsibility:

- Read order emails
- Check stock
- Recommend:
    - Accept
    - Reject
- Send confirmation/rejection email

🔷 CREW STRUCTURE

You should create 3 CREWS.

🔹 Crew 1 – Procurement Crew

Agents:

- Inventory Monitoring Agent
- Vendor Discovery Agent
- Vendor Communication Agent
- Email Analyzer Agent
- Negotiation Agent
- Purchase Order Agent

🔹 Crew 2 – Quality Control Crew

Agents:

- Quality Monitoring Agent
- Vendor Communication Agent

🔹 Crew 3 – Sales Crew

Agents:

- Finished Product Monitor
- Customer Order Agent

🔷 TASK STRUCTURE IN CREWAI

Each agent should have structured tasks.

Example:

Task: Analyze Vendor Reply

Input:

- Raw email content

Output:

```
{
  "price": 120,
  "delivery_days": 5,
  "discount": 10,
  "response_type": "positive"
}
```

Always use structured JSON output.

🔷 REACT PAGES STRUCTURE

You already defined:

- Dashboard
- AI Assistance

- Company Details
- Inventory
- Vendors
- Logistics

Add:

- Purchase Orders
- Quotations
- Quality Reports
- Email Threads

🔷 TEAM DIVISION (7 PEOPLE)

You can divide like this:

👨‍💻 Backend Core (2 People)

- Django models
- APIs
- Authentication
- DB relations

🤖 AI Team (2 People)

- CrewAI agent design

- Prompt engineering
- Tool creation
- JSON structured outputs

📩 Email & Automation (1 Person)

- SMTP + IMAP
- Email thread tracking
- Celery scheduling

⚛️ React UI (2 People)

- Dashboard
- AI Assistance panel
- Vendor comparison table
- Real-time updates

🔷 PRODUCTION LEVEL IMPROVEMENTS

To impress judges:

- Use Celery for async processing
- Maintain email thread mapping
- Maintain audit logs

- Add vendor performance scoring system
- Add retry mechanism for failed vendors
- Add recommendation scoring formula:

Example:

```
Score = (DeliveryWeight * delivery_days_score) +
        (CostWeight * price_score) +
        (RatingWeight * rating_score)
```

🔷 LOGISTICS (PART 2 – HIGH LEVEL)

You will need:

Models:

```
TransportVendor
- name
- vehicle_type
- cost_per_km
- rating

Shipment
- purchase_order
- transport_vendor
- pickup_date
- delivery_status
```

Agents:

- Route Optimization Agent
- Transport Vendor Selection Agent

🔷 EXECUTION ORDER (IMPORTANT)

Day 1:

- DB Models
- Auth
- Company
- Inventory
- Vendor

Day 2:

- Email integration
- Celery
- Inventory monitoring agent

Day 3:

- Vendor sourcing agent
- Email analyzer

Day 4:

- Negotiation
- PO agent
- UI integration

Day 5:

- Quality

- Customer orders
- Recommendation engine

🔥 WHAT WILL MAKE YOU WIN

Judges will love:

- Structured Agent workflow
- Real email automation
- Live vendor comparison dashboard
- Smart recommendation logic
- Rejection handling automation
- Fully functional demo

If you want next:

I can now give you:

- 🔹 Exact CrewAI agent prompt templates
- 🔹 Email parsing structured schema
- 🔹 Celery architecture plan
- 🔹 Full system architecture diagram explanation
- 🔹 Database ER diagram design

Tell me which one you want next 👌