Project Title:

**E-Commerce & Inventory Tracking**

Problem statement:

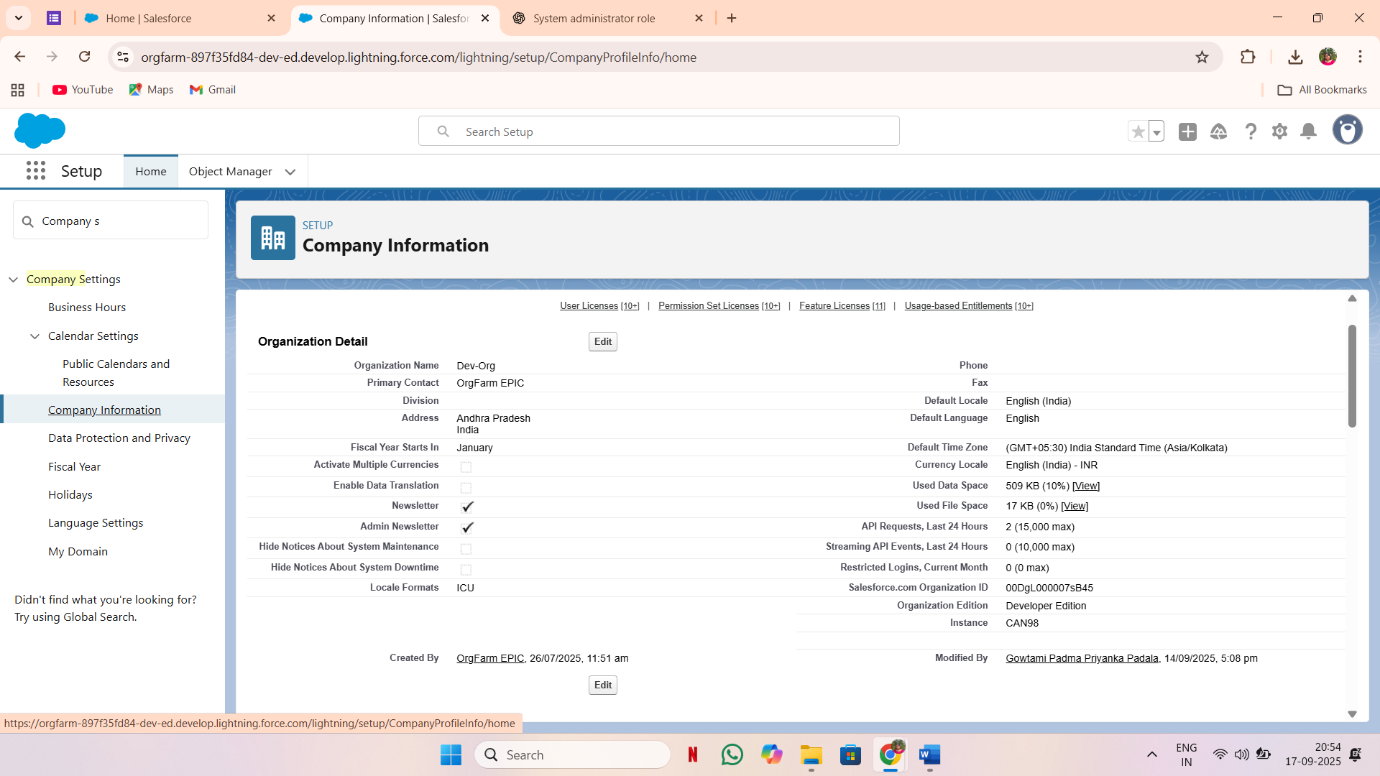
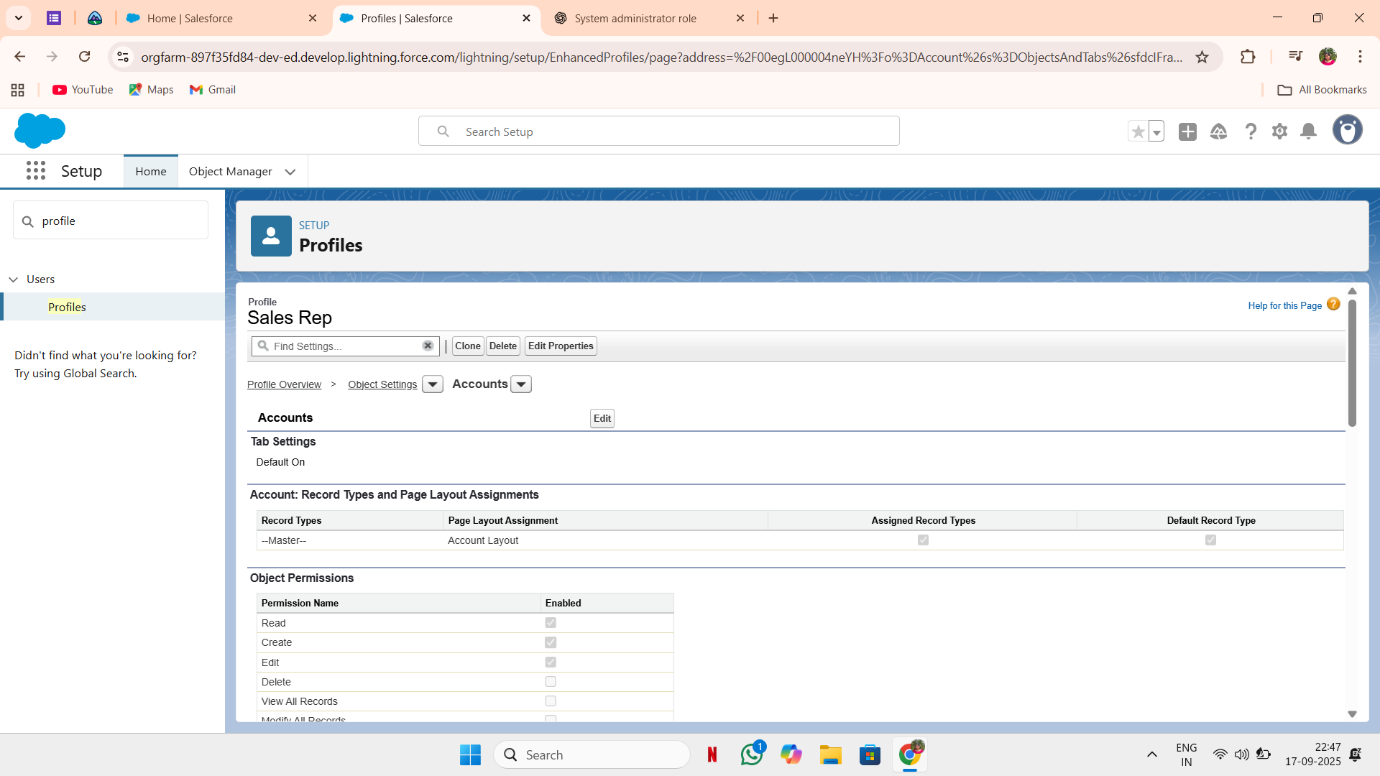
An online store struggles to track customer orders and manage stock availability. They need a Salesforce solution that integrates with e-commerce platforms, manages customer orders, updates inventory automatically, and provides reports on best-selling products.

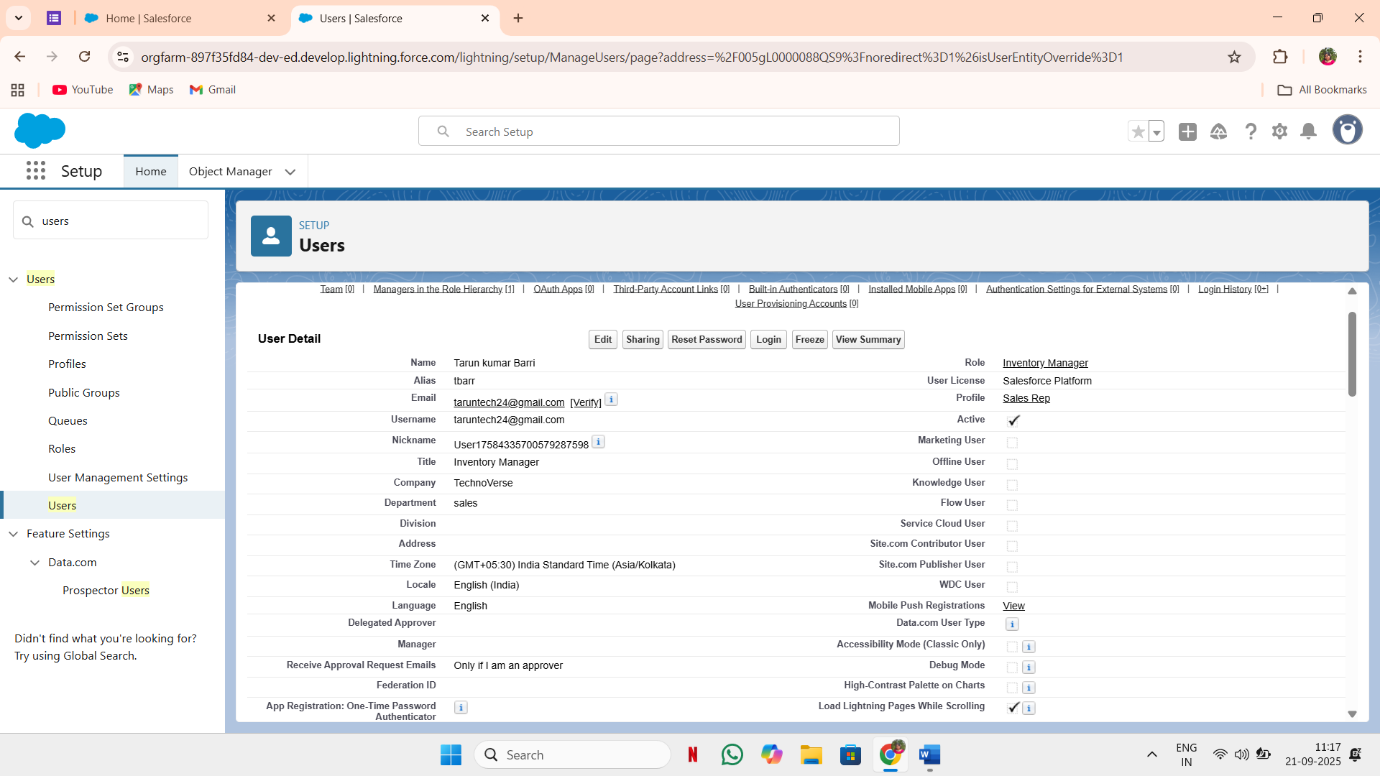
Implementation:

**Phase 1: Problem Understanding & Industry Analysis**

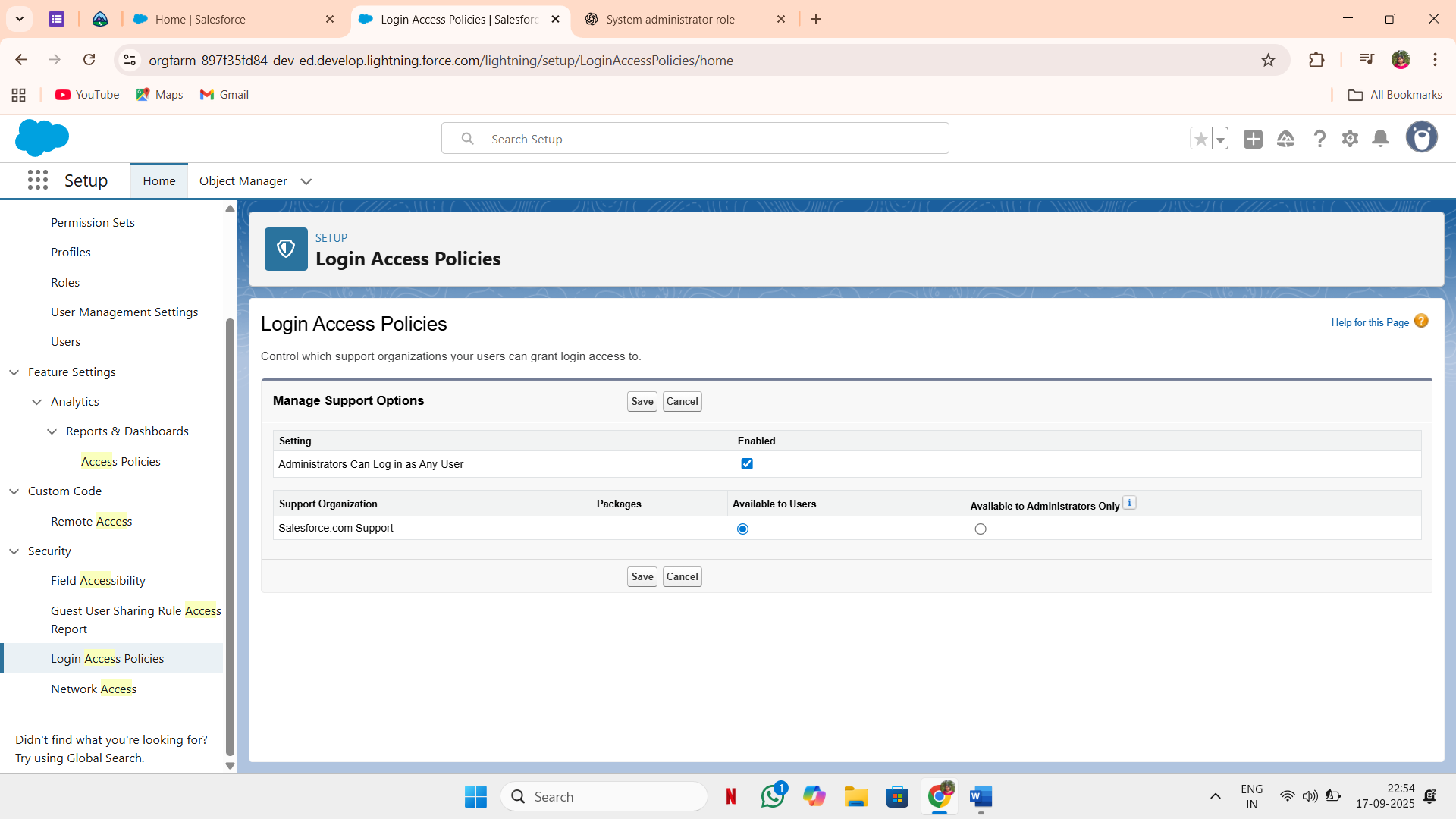
* **Requirement Gathering:**
* Track products, orders, customers, and inventory levels.
* Automate stock reduction & low-stock alert.
* Provide dashboards for sales & inventory insights.
* **Stakeholder Analysis:**
* Customers (place orders).
* Sales Team (manage orders).
* Inventory Manager (track stock).
* **System Administrator** -manages users, roles, configurations**.**
* **Business Owner/Management** - views reports, dashboards, KPIs.
* **Finance/Accounts Team** - tracks payments, refunds, invoices.
* **Business Process Mapping:**
* Customer selects product → Order created → Inventory auto-updates → Confirmation sent → Reports and dashboards generated.
* **Industry Use Case:**
* Similar to Amazon/Flipkart stock tracking.
* **AppExchange Exploration:**
* Look for prebuilt inventory/order management apps as references.

**Phase 2 — Org Setup & Configuration:**

* Setup salesforce Org
* Sign up for a **free Salesforce Developer Edition** (developer.salesforce.com).
* Choose **Sales Cloud** as the edition.
* Configure **Company Settings**
* **Company Info** → Go to *Setup* → **Company Settings → Company Information** → Fill in company name, address, currency, time zone.
* **Business Hours** → *Setup* → **Business Hours** → Create working hours (e.g., Mon–Fri 9 AM – 5 PM).
* **Holidays** → *Setup* → **Holidays** → Add public holidays (these can be used in escalation rules).
* Create **Users, Roles, Profiles**
* **Users** → *Setup* → **Users → New User**
  + Example: *Sales Rep, Inventory Manager, System Admin*.
* **Roles** → *Setup* → **Roles → Set up Roles**
  + Example hierarchy:
    - CEO/Admin (top)
      * Sales Manager
      * Inventory Manager
* **Profiles** → *Setup* → **Profiles**
  + Use *System Administrator* profile for admin.
  + Clone *Standard User* profile to create *Sales Rep Profile* and *Inventory Manager Profile* with appropriate permissions.

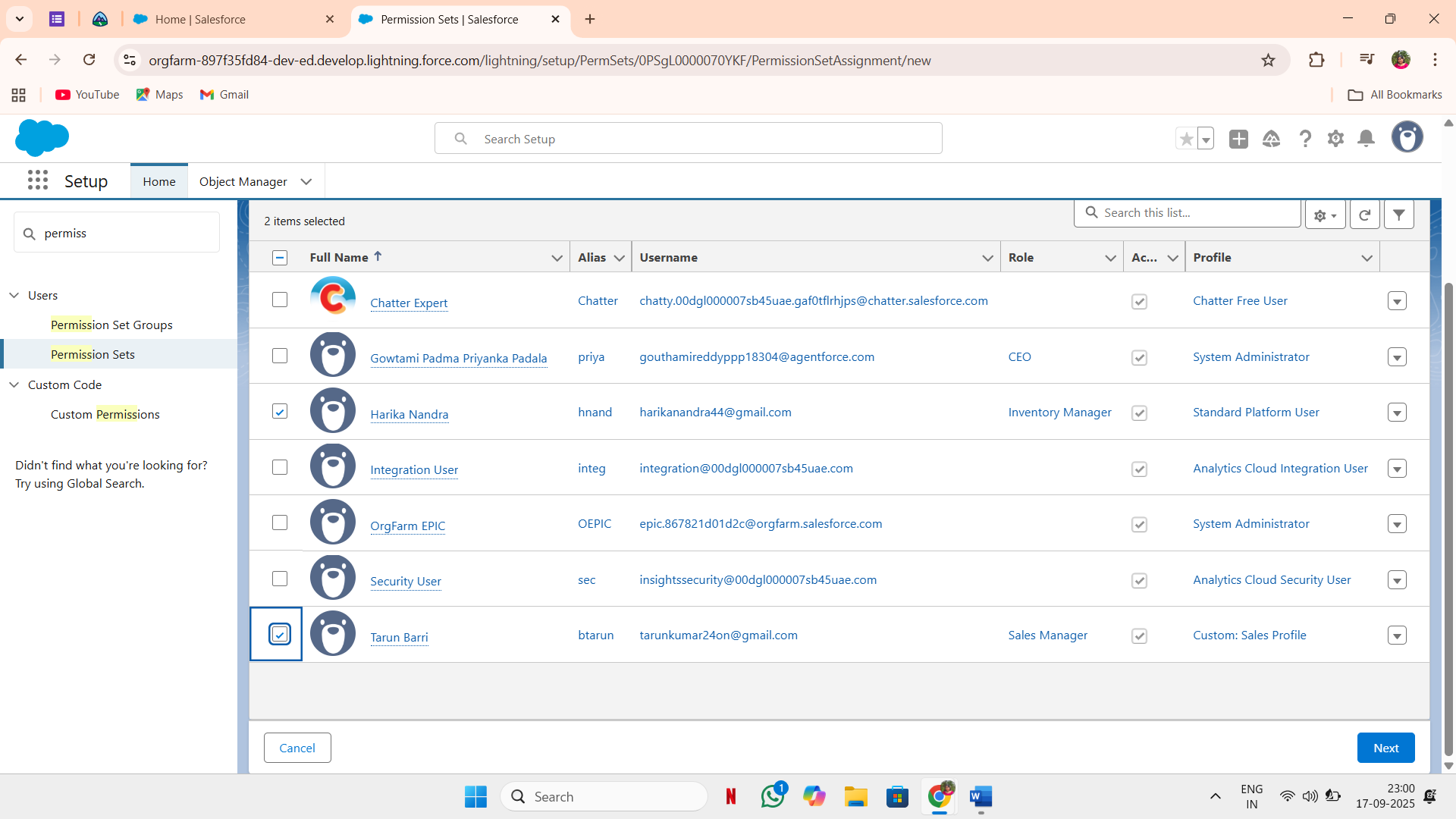


* Define **Permission Sets**
* *Setup* → **Permission Sets → New**
  + **Order Approvers** → Add “Modify All” on Orders.
  + **Inventory Editors** → Add “Edit” access on Inventory objects.
* Assign these permission sets to users who need special access without changing their profile.
* Configure Sharing Settings:
* **Organization-Wide Defaults (OWD):**
  + *Setup* → **Sharing Settings** → Set *Orders = Private* (only owner sees records).
* **Sharing Rules:**
  + Example: Create a rule to share *Orders* with *Sales Managers* so they see all team orders.
  + Use these as required by the process.
* Enable SecuritySettings
* **Login Access Policies** → Allow admins to log in as users for troubleshooting.



* **MFA (Multi-Factor Authentication):**
* Enable from *Setup* → **Identity Verification → Multi-Factor Authentication**.

Adding multi factor authentication for users using permission sets



* **Audit Trail:**

*Setup* → **View Setup Audit Trail** → Monitor configuration changes

**Phase 3: Data Modeling & Relationships**

Step 1: Identify Core Business Entities

* E-Commerce Objects:
  + Product (standard: Product2)
  + Pricebook / PricebookEntry (for pricing control)
  + Order / OrderItem
  + Customer (standard: Account & Contact)
  + Cart (custom object if required before checkout)
  + Payment (custom object)
* Inventory Objects:
  + Inventory Item (custom object linked to Product2)
  + Warehouse / Location (custom object)
  + Stock Movement (custom object for tracking stock-in/stock-out)
  + Supplier / Vendor (custom object)

Step 2: Create Standard & Custom Objects

* Use Object Manager in Salesforce Setup.
* Extend standard objects (Account, Contact, Product, Order).
* Create custom objects like:
  + Warehouse\_\_c
  + Inventory\_Item\_\_c
  + Stock\_Transaction\_\_c
  + Payment\_\_c

**A. Standard Objects to Leverage**

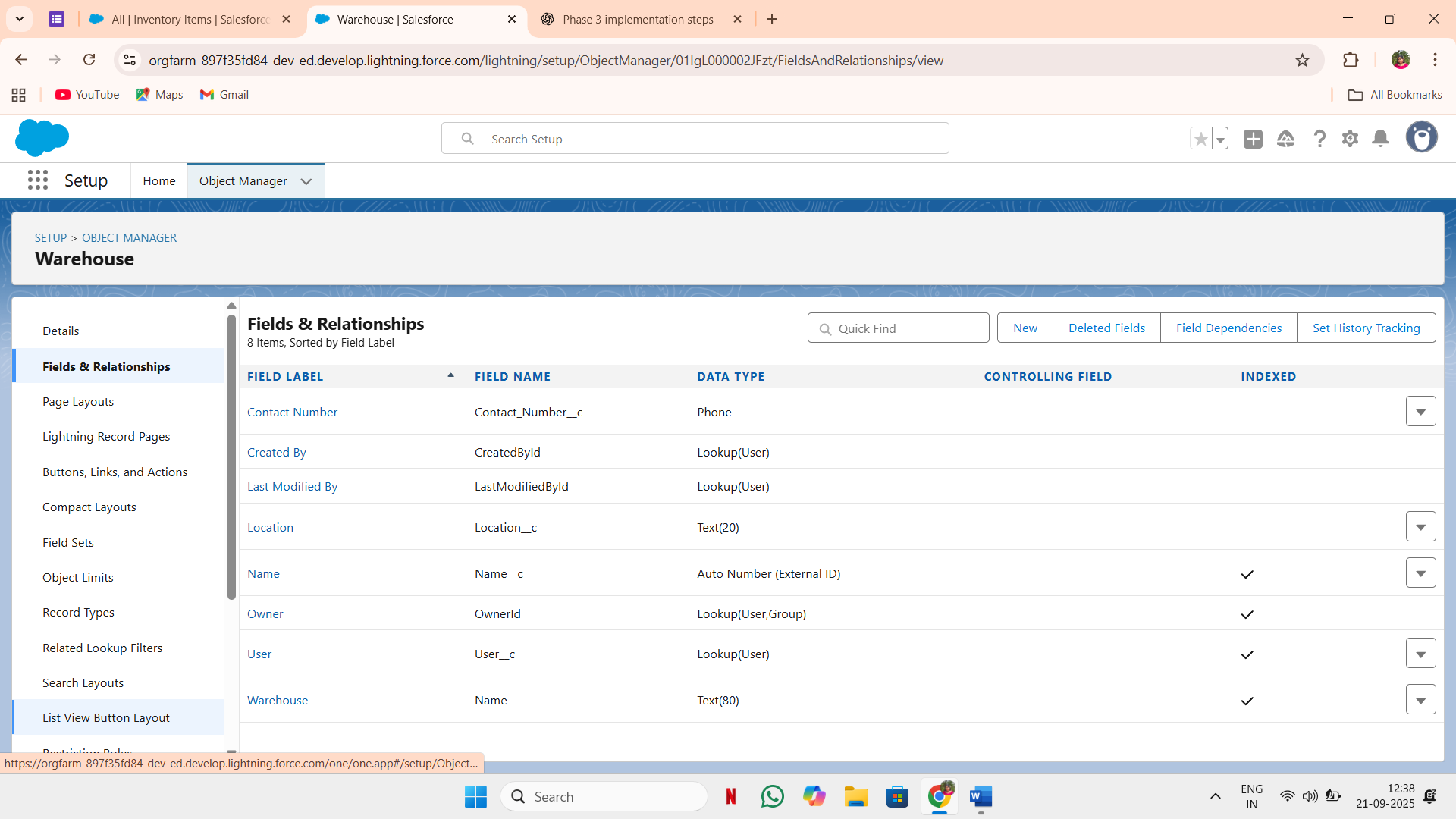
1. **Account** – Represents **customers**
   * Example: *Amazon Seller*, *Customer John Doe*.
2. **Contact** – Represents people tied to Accounts.
   * Example: *Customer’s shipping contact info*.
3. **Product2** – Represents your products.
   * Example: *iPhone 15, Samsung TV*.
4. **Pricebook / PricebookEntry** – Holds pricing.
   * Example: *Retail Pricebook, Wholesale Pricebook*.
5. **Order / OrderItem** – Captures purchase details.
   * Example: *Order #1234 with 3 OrderItems*.

**B. Custom Objects to Create**

**Create tabs for these custom objects so that they are visible to the application.**

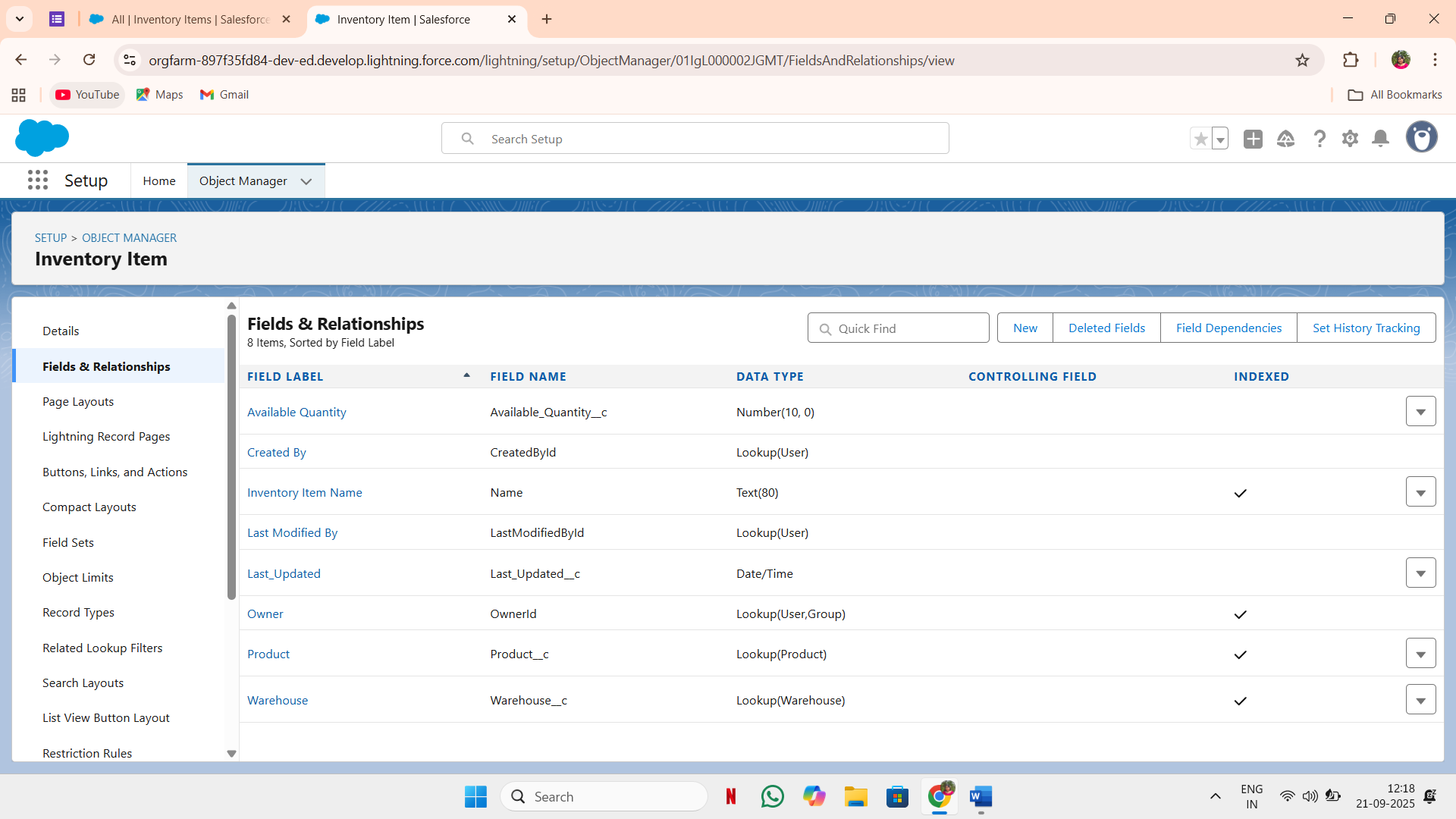
**1. Warehouse\_\_c**

* Stores warehouse locations.
* Key Fields:
  + Name (Auto-number: WH-0001, WH-0002…)
  + Location\_\_c (Text/Geolocation)
  + Manager\_\_c (Lookup → User)
  + Inventory Item(Master-Detail->Inventory Item)
  + Contact Number(number)



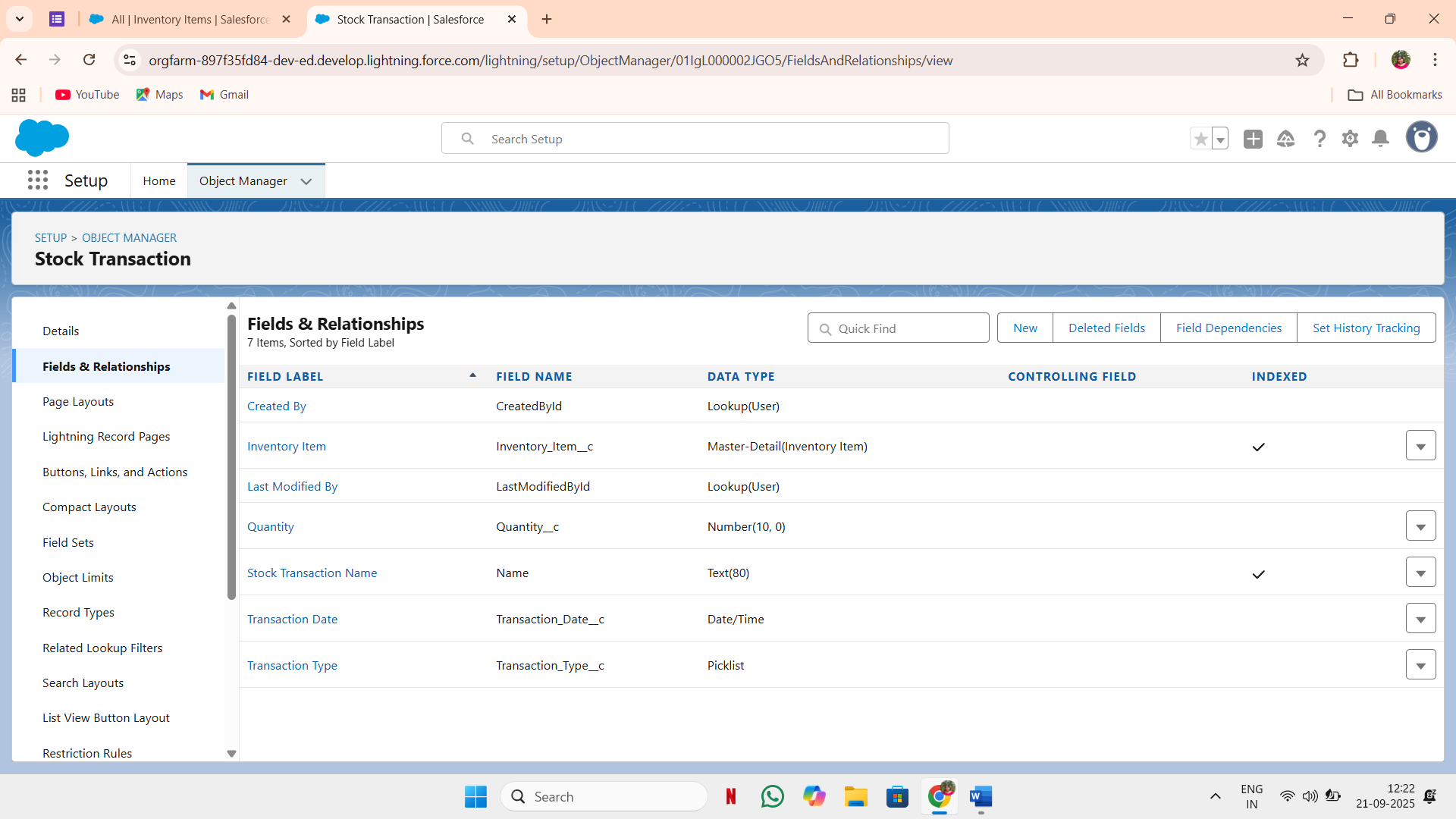
**2. Inventory\_Item\_\_c**

* Tracks stock for each product in each warehouse.
* Key Fields:
  + Product\_\_c (Lookup → Product)
  + Warehouse\_\_c (Lookup → Warehouse\_\_c)
  + Available\_Quantity\_\_c (Number)
  + Reorder\_Level\_\_c (Number)
  + Last\_Updated\_\_c (Date/Time)



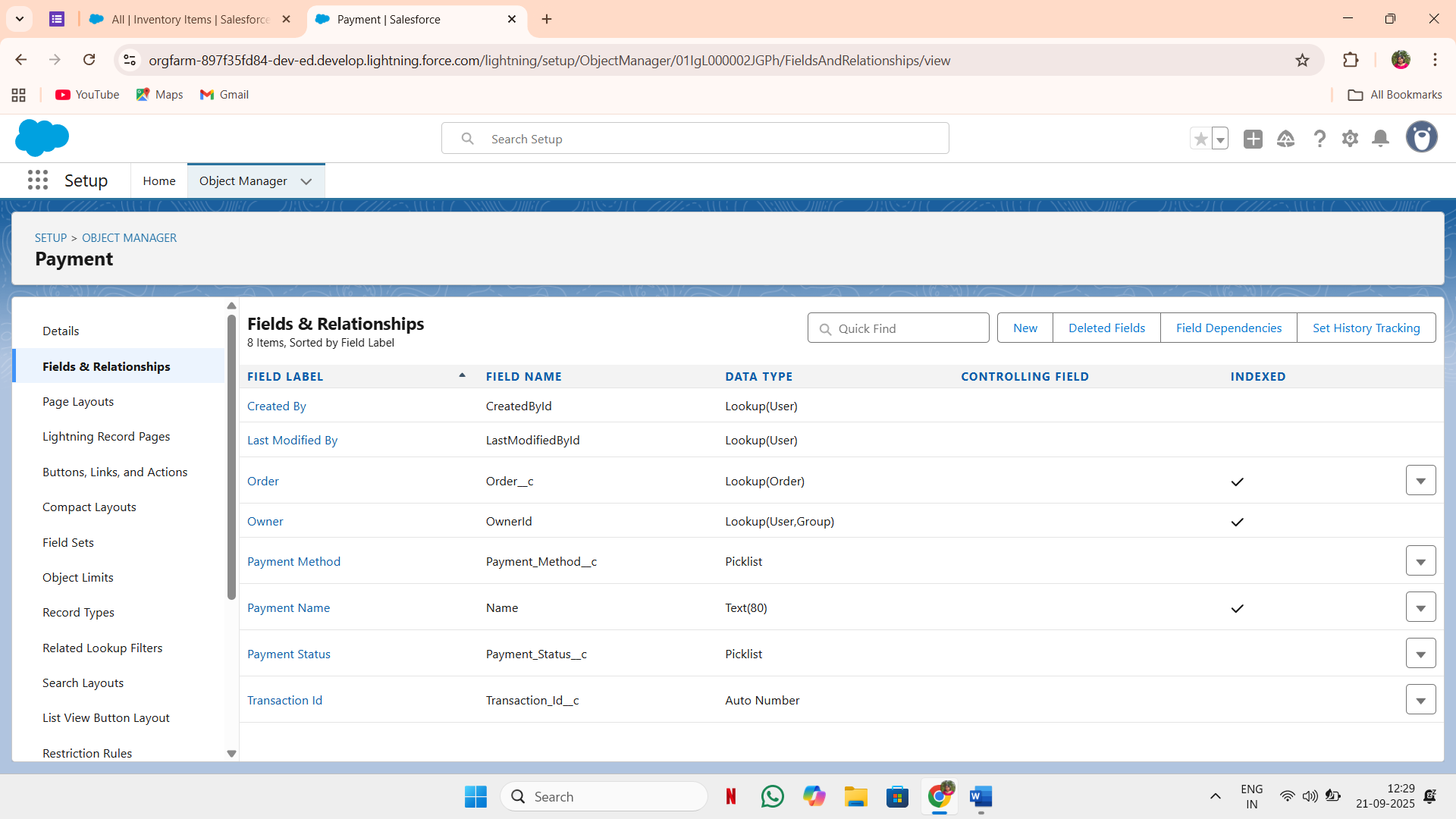
**3. Stock\_Transaction\_\_c**

* Logs movements in/out of inventory.
* Key Fields:
  + Inventory\_Item\_\_c (Master-Detail → Inventory\_Item\_\_c)
  + Transaction\_Type\_\_c (Picklist: Inbound, Outbound, Adjustment)
  + Quantity\_\_c (Number)
  + Transaction\_Date\_\_c (Date/Time)

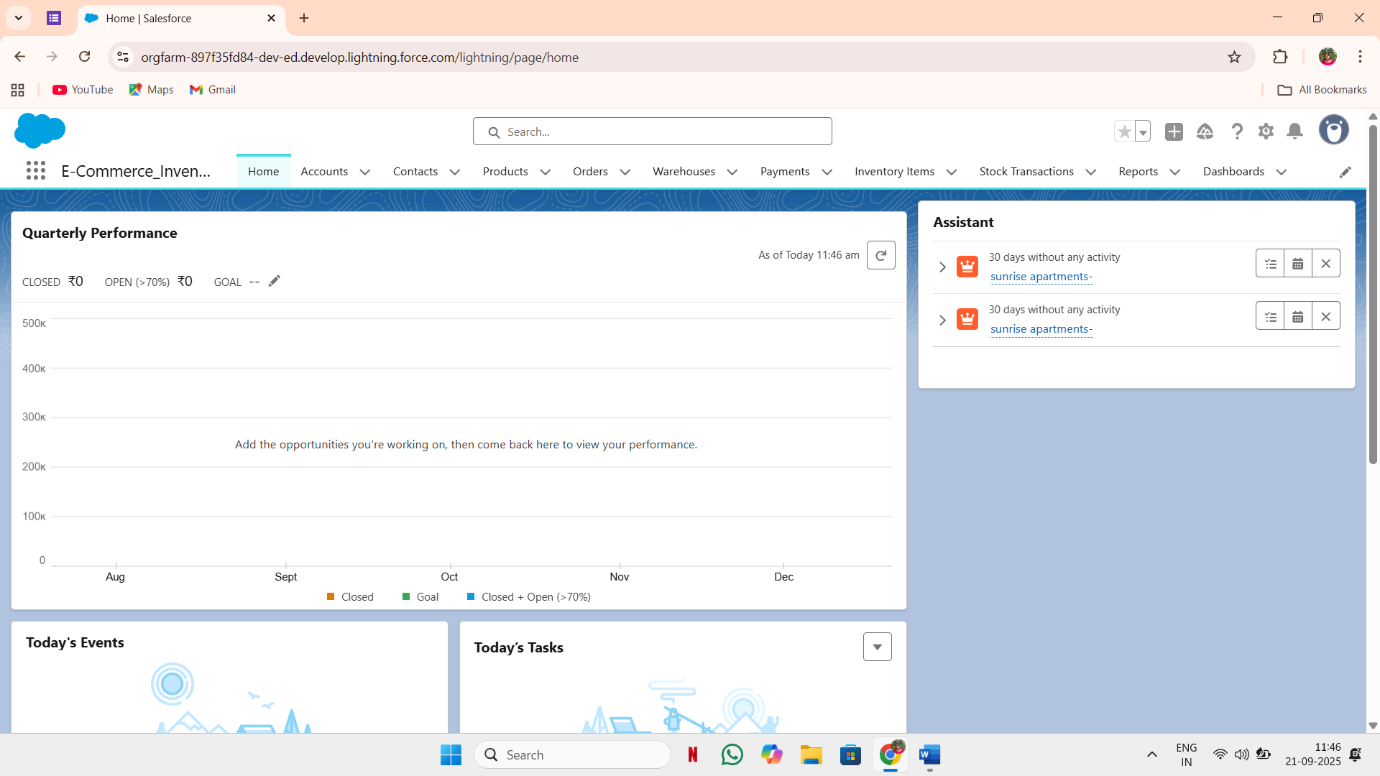


**4. Payment\_\_c**

* Captures payment details for an order.
* Key Fields:
  + Order\_\_c (Lookup → Order)
  + Payment\_Method\_\_c (Picklist: Credit Card, UPI, Net Banking, COD)
  + Payment\_Status\_\_c (Picklist: Pending, Completed, Failed)
  + Transaction\_Id\_\_c (Text)



**Create the Lightning App (Setup → App Manager → New Lightning App)**

* App Label: **ECommerceInventory**
* App Name: ECommerceInventory
* Description: E-commerce + Inventory Management
* Branding: Set logo & color
* Navigation Style: Console or Standard (Console recommended for Inventory Manager)
* Add Navigation Items (tabs): Accounts, Contacts, Products, Price Books, Orders, Payments, Warehouse, Inventory Item, Stock Transaction, Cart, Reports, Dashboards
* Assign Visibility: Choose which profiles can see the app (Sales Rep, Inventory Manager, Admin)

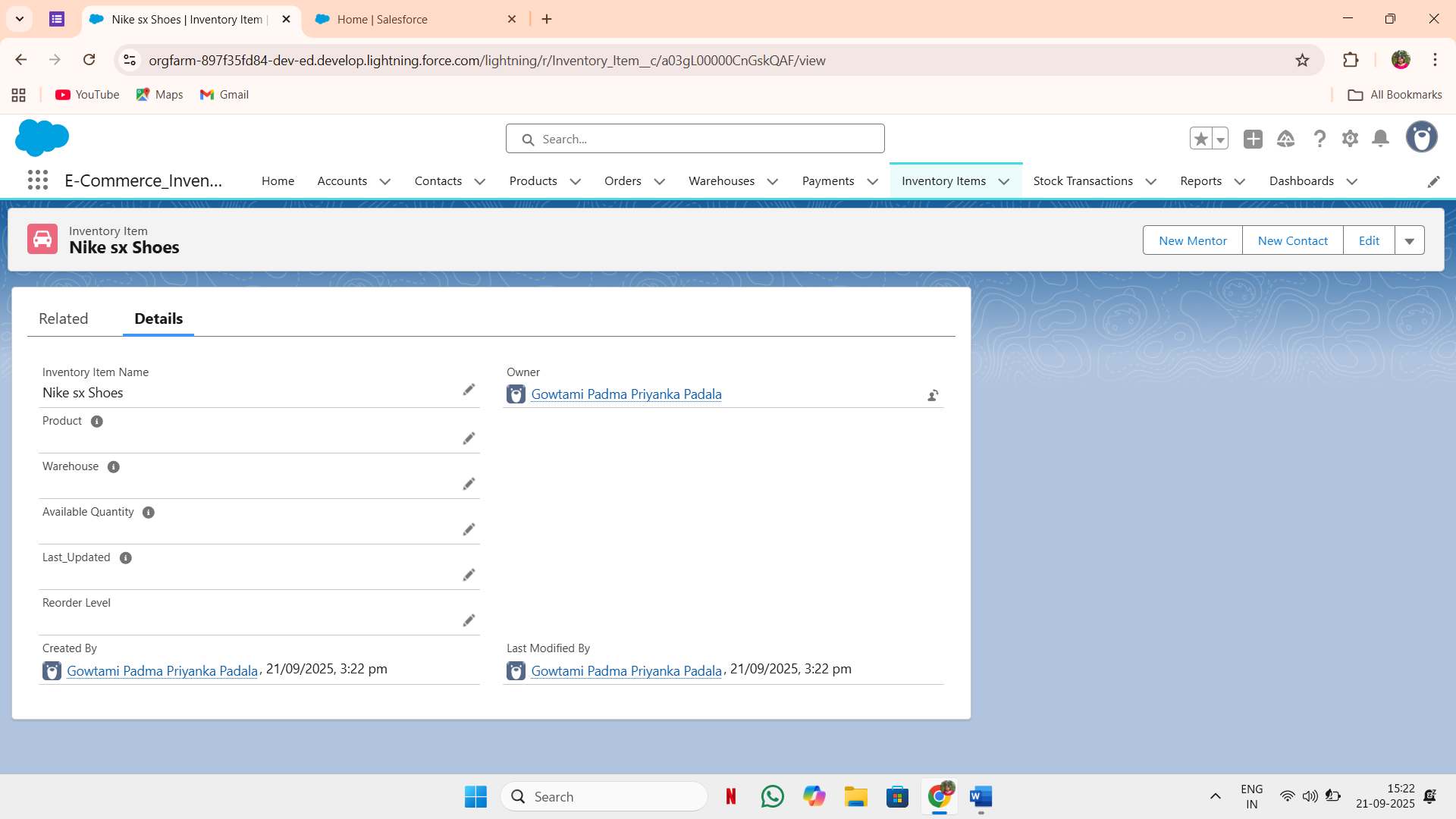
Step 4: Set Up Record Types

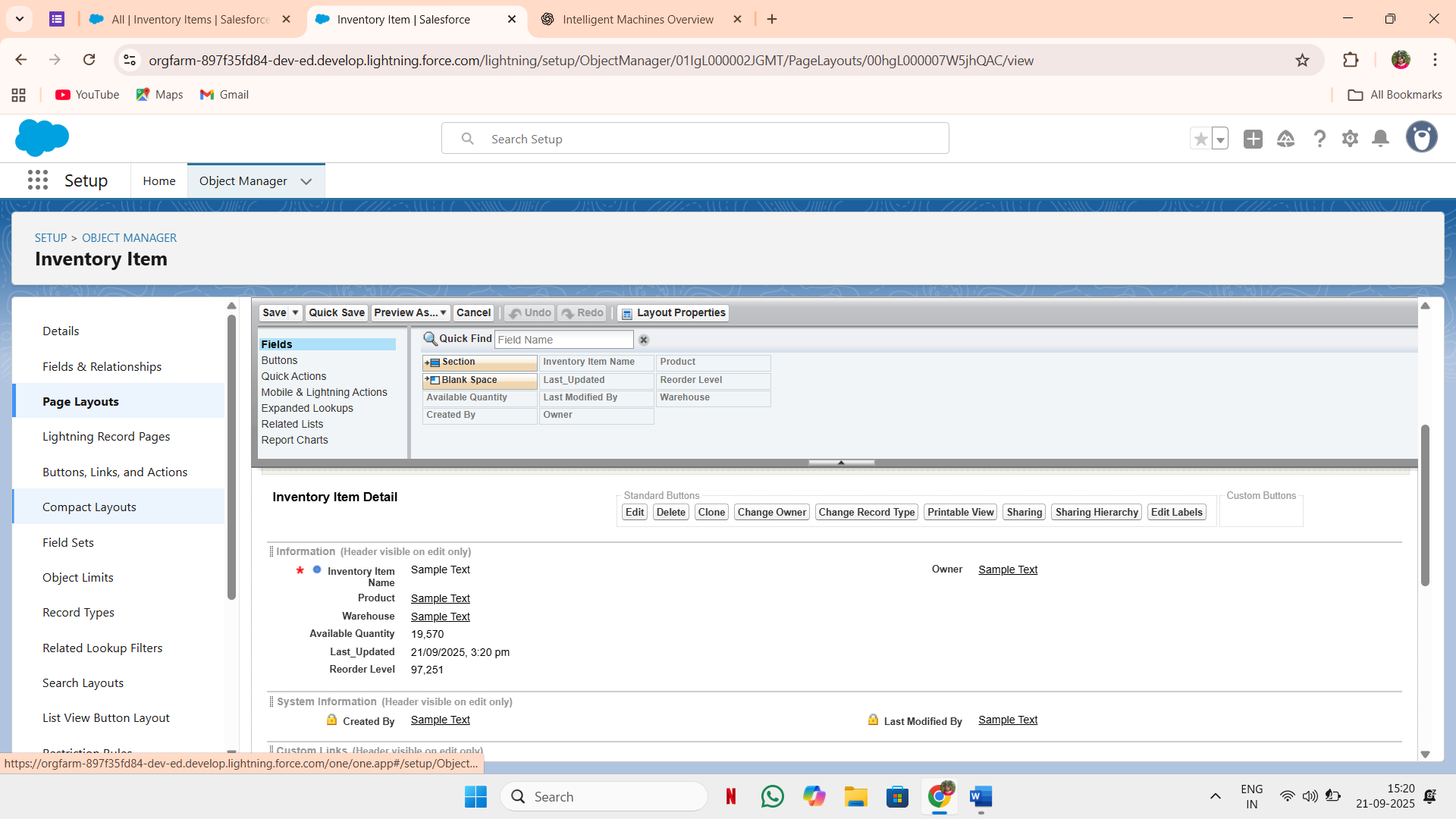
* **Purpose:** Let you create **different business processes, picklist values, and page layouts** for the same object.
* For Products:
  + Record Types: *Electronics*, *Clothing*, *Grocery* (different layouts/fields).
* For Orders:
  + Record Types: *Retail Order*, *Bulk Order*.
* Each record type can:
  + Show different **picklist values** (e.g., "Stage" values).
  + Use a different **page layout**.

✅ **Think of Record Types as “categories” of records within the same object.**

Step 5: Design Page Layouts & Compact Layouts

* **Purpose:** Control the **fields, sections, related lists, and buttons** visible on a record’s detail page.



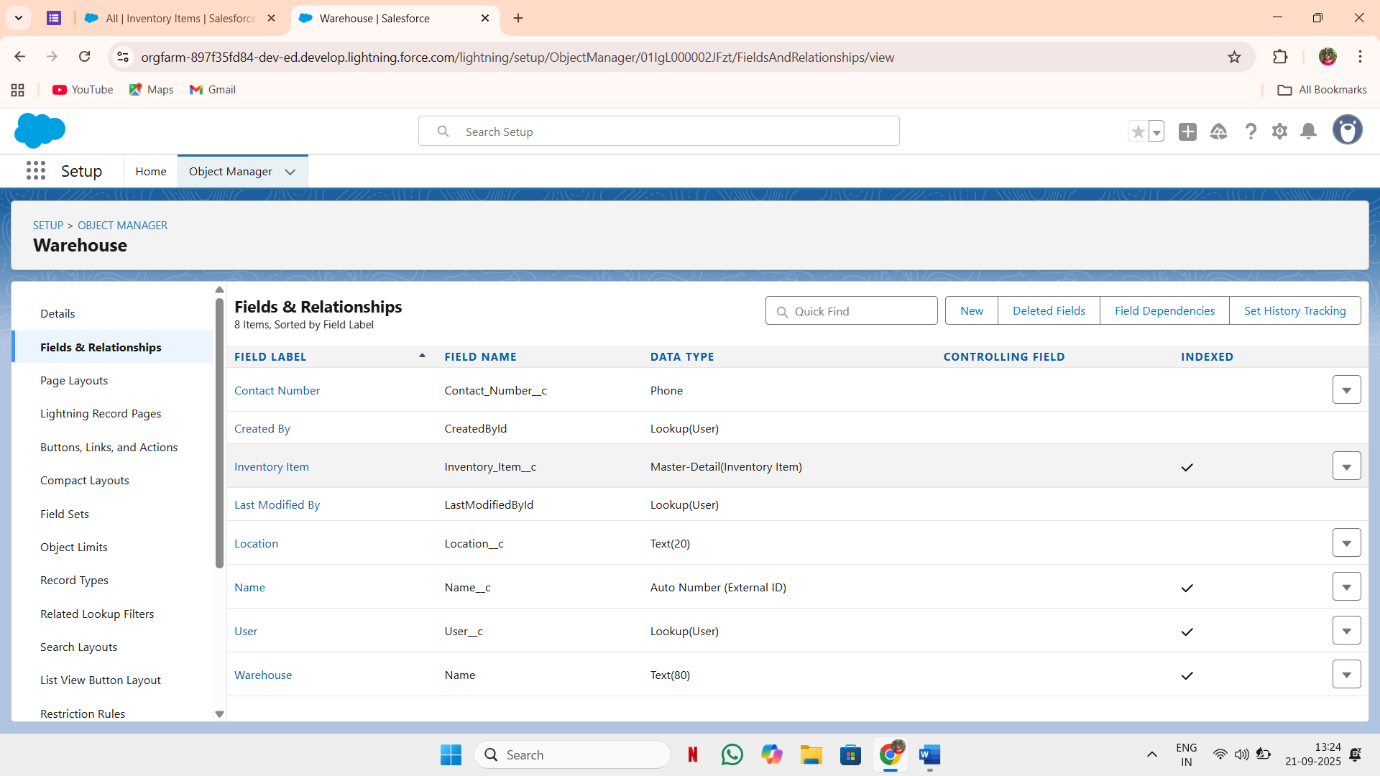


* Customize layouts for each profile (Sales Rep, Inventory Manager).
* Example:
  + Inventory Manager sees stock fields (Reorder Level, Warehouse).
  + Sales Rep sees customer/order-related fields.
* Assign **different page layouts** to profiles (or record types).

✅ **Think of Page Layouts as “what fields and sections you see and edit.”**

* **Think of Compact Layouts as “the quick info you see at the top or on mobile.”**

Step 6: Establish Relationships

* Used Schema Builder to map:
  + Account → Contact (Standard Parent-Child).
  + Product2 → PricebookEntry → OrderItem (Standard Salesforce e-commerce flow).
  + Product2 ↔ Inventory\_Item\_\_c (Lookup or Master-Detail).
  + Warehouse\_\_c ↔ Inventory\_Item\_\_c (Master-Detail).
  + Order\_\_c ↔ Payment\_\_c (Lookup).
  + Stock\_Transaction\_\_c ↔ Inventory\_Item\_\_c (Master-Detail).

Step 7: Use Junction Objects

* If needed, create junction objects:
  + Example: *Product-Warehouse Mapping* if a product is stored in multiple warehouses.
  + Example: *Customer-Cart* for handling multiple customers adding items to different carts.

Step 8: Consider External Objects

* If inventory or payment data resides in an external ERP/Payment system, set up External Objects via Salesforce Connect.

**Phase 4: Process Automation (Admin)**

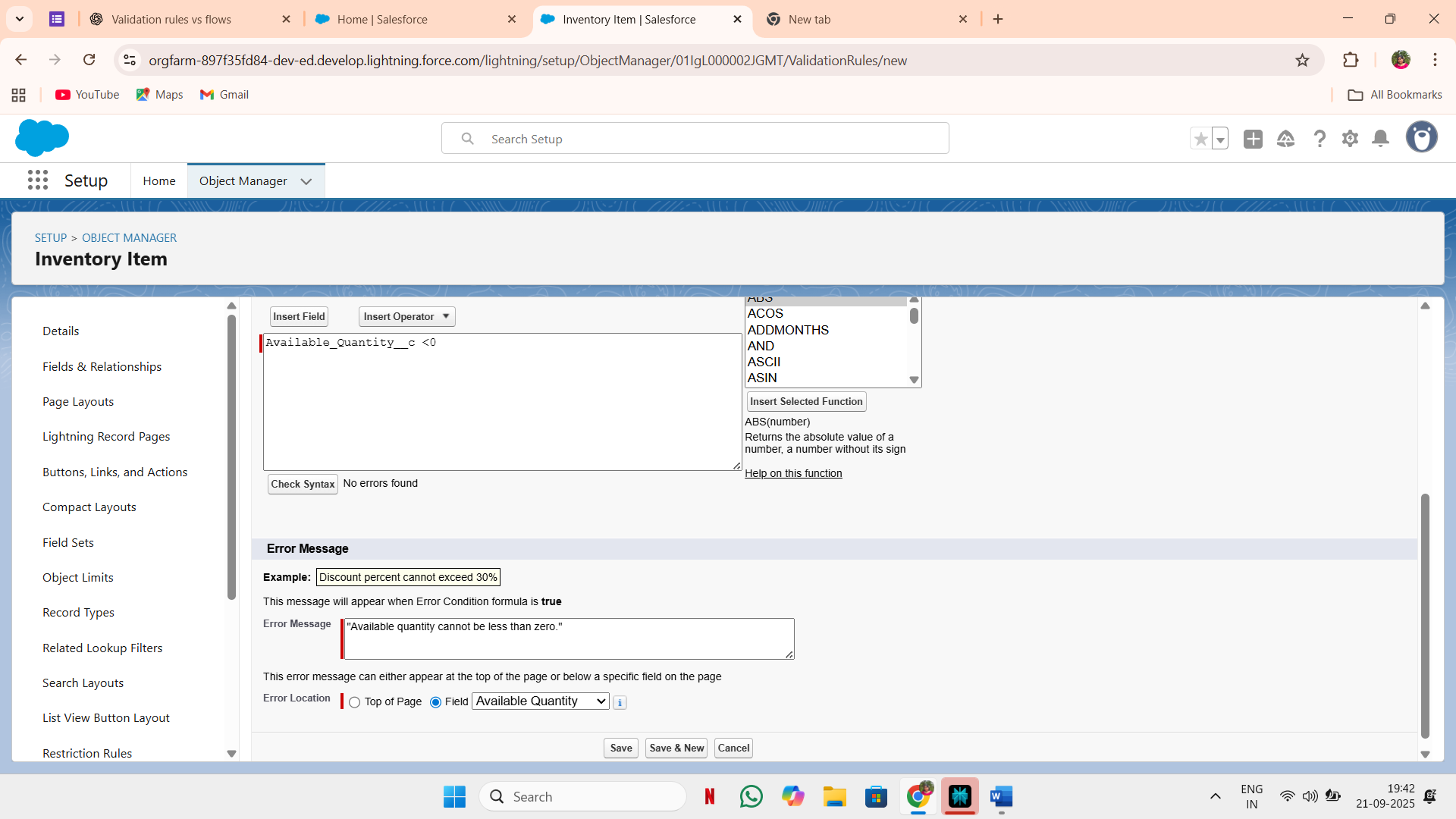
This phase focuses on reducing manual work and ensuring business processes run efficiently using Salesforce automation tools.

**1. Validation Rules**

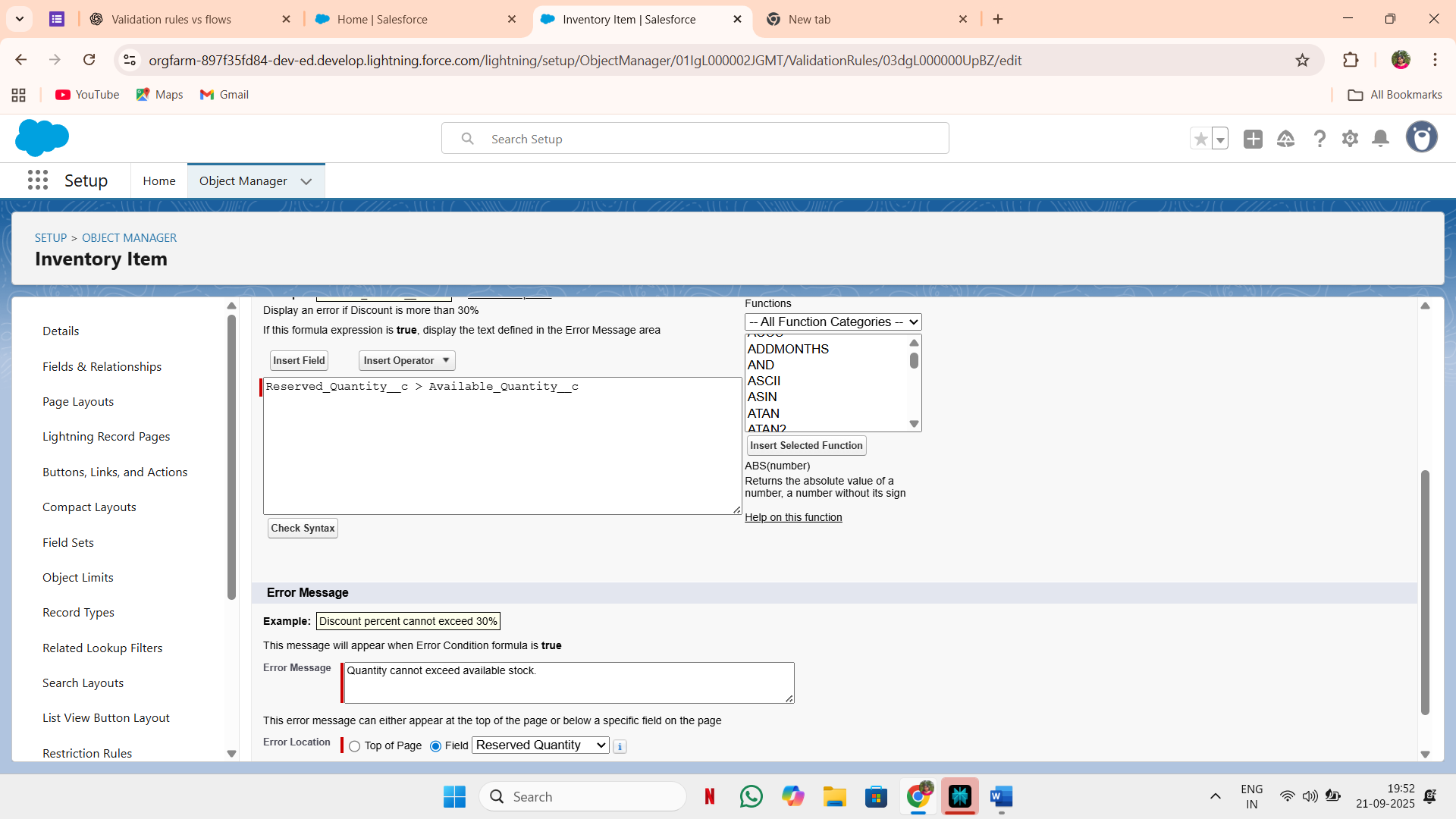
* Enforce data quality by restricting invalid entries.
* Go to **Setup → Object Manager → [Your Object, e.g., Product or Inventory] → Fields & Relationships**.
* Click **“New” → Validation Rule**.

Use cases:

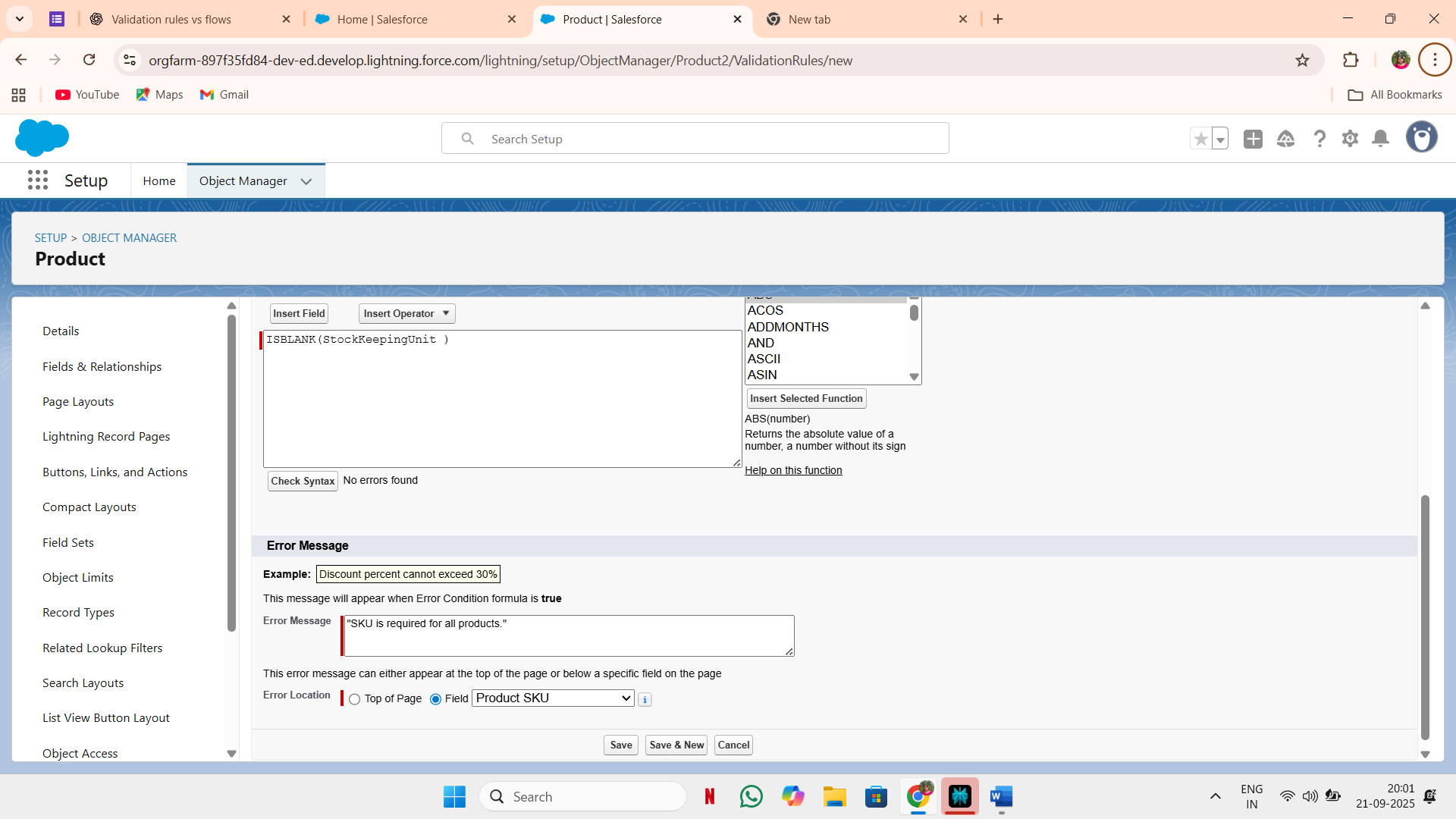
* **Inventory Control**
  + Prevent Available\_Quantity\_\_c from going below zero.



* + Prevent Quantity\_\_c from being greater than available stock.



* **Product Compliance**
  + SKU must be filled in for all products.



**2. Workflow Rules**

* Automate simple if/then actions (though being replaced by Flow).  
  **Order Confirmation Email** → Automatically send an email to the customer when an order is created.

**3. Process Builder**

* Automate multi-step logic (being replaced by Flow gradually).

**4. Approval Process**

* Route records for managerial approval.

**5. Flow Builder**

* **Record-Triggered Flow**: Auto-update fields when conditions are met.
* **Screen Flow**: Guided UI for data entry (e.g., Case resolution steps).
* **Scheduled Flow**: Automate time-based actions (e.g., send reminders for upcoming contract renewal).
* **Auto-launched Flow**: Execute behind the scenes without user interaction.

**6. Email Alerts**

* Send templated emails triggered by workflow/process/flow.
* **Order Confirmation Email** → Automatically send an email to the customer when an order is created.
* **Order Status Update Notification** → Notify the sales team or customer when the order status changes (e.g., Pending → Shipped).
* **Low Stock Alert** → Send an email or notification to inventory managers if Available\_Quantity\_\_c < Minimum\_Stock\_\_c.

**7. Field Updates**

* Auto-update field values based on logic. -> Triggered by flows

**8. Tasks**

* Create automatic tasks for users. Also done using flows.
* **Follow-up Task** → Automatically create a task for a sales rep when a customer places their first order.
* **Pending Order Reminder** → Create tasks if an order has been pending for too long.

**9. Custom Notifications**

**Use case:**

* **Welcome Email** → Automatically send a welcome email when a new customer record is created.