**Art Gallery CRM Project**

# Phase 1: Problem Understanding & Industry Analysis

👉 **Goal:** Understand what we’re building, why it’s needed, and how it fits into the art gallery industry.

### **1. Requirement Gathering**

To build a CRM that actually solves the challenges faced by an art gallery, requirements should be collected from different stakeholders involved in daily operations:

* **Curators / Gallery Managers**
  + Need to organize exhibitions with start/end dates, assign artworks, and ensure proper documentation of participating artists.
  + Require visibility into artwork inventory and availability (which artworks are sold, reserved, or displayed).
  + Want performance metrics (top-selling artists, exhibition attendance).
* **Sales/Finance Teams**
  + Must record artwork sales with accuracy.
  + Need automatic calculation of commissions (artist vs gallery share).
  + Require invoicing and financial reporting for sales and revenue.
* **Visitors / Event Attendees**
  + Want a simple registration process for exhibitions and events.
  + Expect confirmation emails and reminders.
  + Prefer personalized communication for upcoming events.
* **Artists**
  + Expect transparency in sales (notifications when their artwork is sold).
  + Need clarity on commission amounts.
  + Want a record of which exhibitions feature their work.

**Example requirements gathered:**

* Maintain a digital catalogue of all artworks with images, prices, and status.
* Allow curators to schedule and manage exhibitions.
* Automate artwork status updates after a sale.
* Calculate and record artist commissions without manual errors.
* Capture and manage visitor registrations for exhibitions.
* Generate reports on sales trends, artwork inventory, and visitor engagement.

### **2. Stakeholder Analysis**

Different users will interact with the CRM in different roles.

* **Admin (You / Gallery IT Manager)**
  + Configures Salesforce objects, fields, and automations.
  + Ensures data security, access control, and system customization.
* **Curators / Gallery Managers**
  + Create and manage exhibitions.
  + Assign artworks to exhibitions.
  + Monitor artwork availability and sales reports.
* **Sales / Finance Staff**
  + Record sales and track payments.
  + Verify commission calculations.
  + Generate financial and sales dashboards.
* **Visitors / Customers**
  + Register for exhibitions/events.
  + Receive confirmation and follow-up emails.
  + Provide feedback on exhibitions.
* **Artists** (indirect stakeholders)
  + Receive automated notifications when artworks are sold.
  + Benefit from transparent tracking of their sales and commissions.

### **3. Business Process Mapping**

We can map the gallery’s operations as a simple workflow:

1. **Exhibition Planning**
   * Curator creates an exhibition → assigns artworks and artists → publishes details.
2. **Visitor Registration**
   * Visitor registers online → record created in CRM → email confirmation sent.
3. **Artwork Sale**
   * Sales agent records sale → system validates reserve price → sale is confirmed.
4. **Post-Sale Updates**
   * Artwork status changes to Sold.
   * Commission calculation is triggered → artist and gallery notified.
5. **Reporting**
   * Manager reviews dashboards (sales, visitor registrations, top artists).

This ensures smooth transitions from planning → execution → reporting, without manual errors.

### **4. Industry-Specific Use Case Analysis**

The art gallery industry has unique challenges compared to generic retail or event businesses:

* **Artwork Inventory is Limited & Unique**
  + Each artwork is one-of-a-kind; once sold, it cannot be resold.
  + Tracking availability and preventing duplicate sales is critical.
* **Artist-Gallery Financial Relationship**
  + Commission-based revenue sharing is standard.
  + Galleries must ensure transparent and fair calculation of commissions.
* **Exhibition as Core Revenue Driver**
  + Success depends on visitor engagement and sales during exhibitions.
  + Visitor registration data helps forecast attendance and measure success.
* **Visitor Experience**
  + Personalized communication (invitations, reminders) encourages loyalty.
  + Repeat visitors and VIP customers can be identified for special events.

Thus, our CRM must handle **inventory management, automated commission workflows, event/visitor tracking, and visitor engagement** seamlessly.

### **5. AppExchange Exploration**

* A quick search on **Salesforce AppExchange** shows event management and inventory tracking apps, but very few specifically for **art galleries**.
* Some “museum” or “event ticketing” apps exist, but they are too generic or overly complex.
* Instead, we will build a **custom, lightweight Art Gallery CRM** tailored specifically for:
  + Artwork inventory (one-of-a-kind items).
  + Commission-based sales tracking.
  + Exhibition-focused visitor engagement.

This ensures the project is **unique, practical, and a strong fit for a resume**, while also giving you hands-on Salesforce customization experience.

# Phase 2: Org Setup & Configuration

### 👉 **Goal:** Prepare Salesforce environment for Art Gallery CRM.

This phase ensures the foundational Salesforce setup (users, roles, permissions, org-wide defaults, etc.) matches the business needs of an **Art Gallery** (tracking exhibitions, artists, artwork sales, visitor registrations).

### Z4**1. Salesforce Edition**

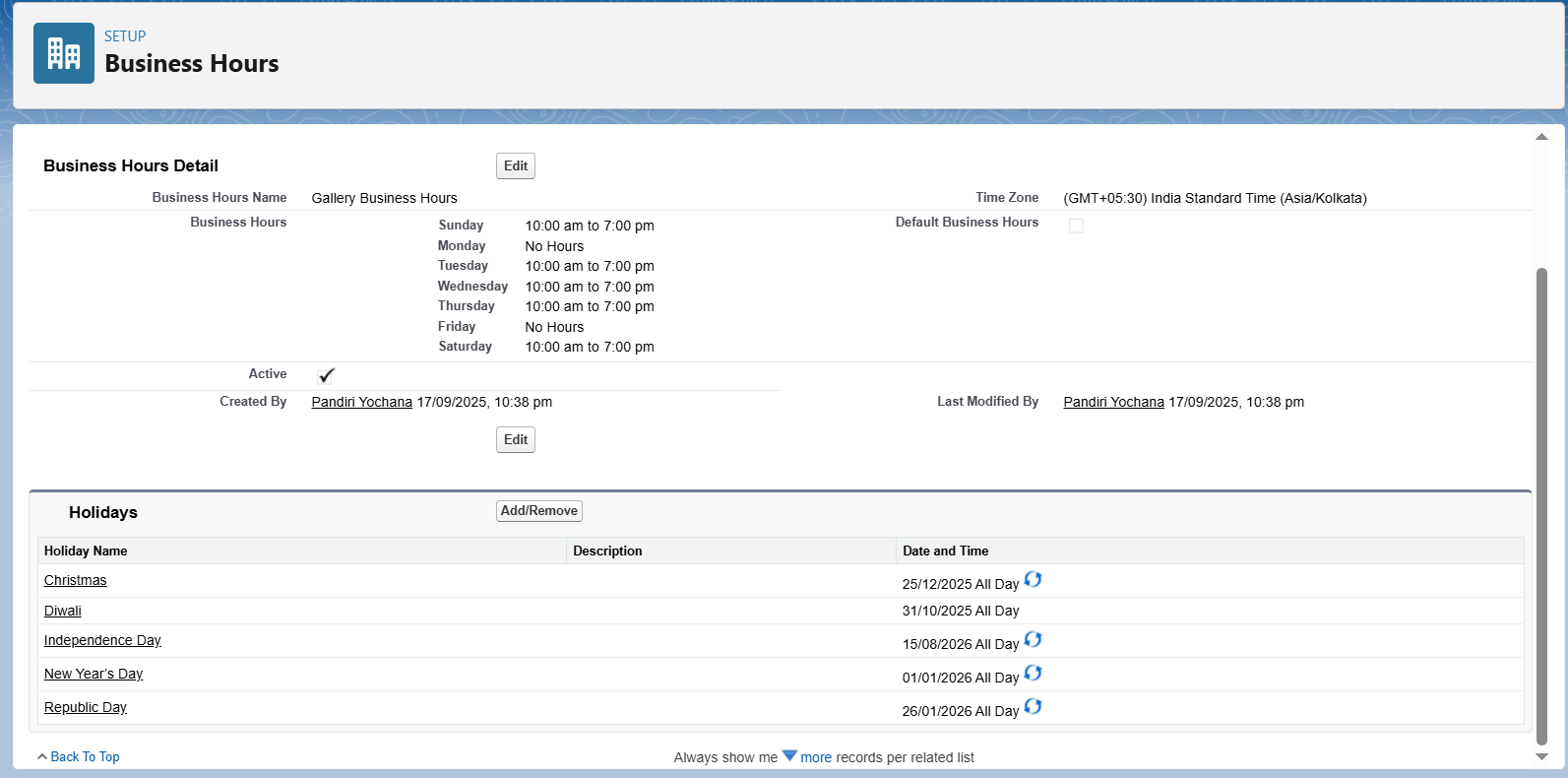
* Used **Salesforce Developer Edition (Free Dev Org)** to build the project.
* This edition provides standard CRM features, custom objects, workflow automation, reports, and dashboards – enough to model gallery processes.

### **2. Company Profile Setup**

* Navigated to **Setup → Company Information**.
* Configured company as:
  + **Company Name:** Modern Art & Exhibitions Pvt. Ltd.
  + **Primary Language:** English.
  + **Time Zone:** Asia/Kolkata (IST).
  + **Currency:** INR (₹), since sales reporting and artwork pricing in India is mostly in rupees.

### **3. Business Hours & Holidays**

* Defined **Gallery Business Hours: 10 AM – 7 PM** (Tuesday–Sunday).
* Added **Holidays** like Independence Day, Diwali, New Year, when exhibitions remain closed.
* This ensures workflows like approvals or visitor event notifications won’t trigger on closed days.



### **4. Fiscal Year Settings**

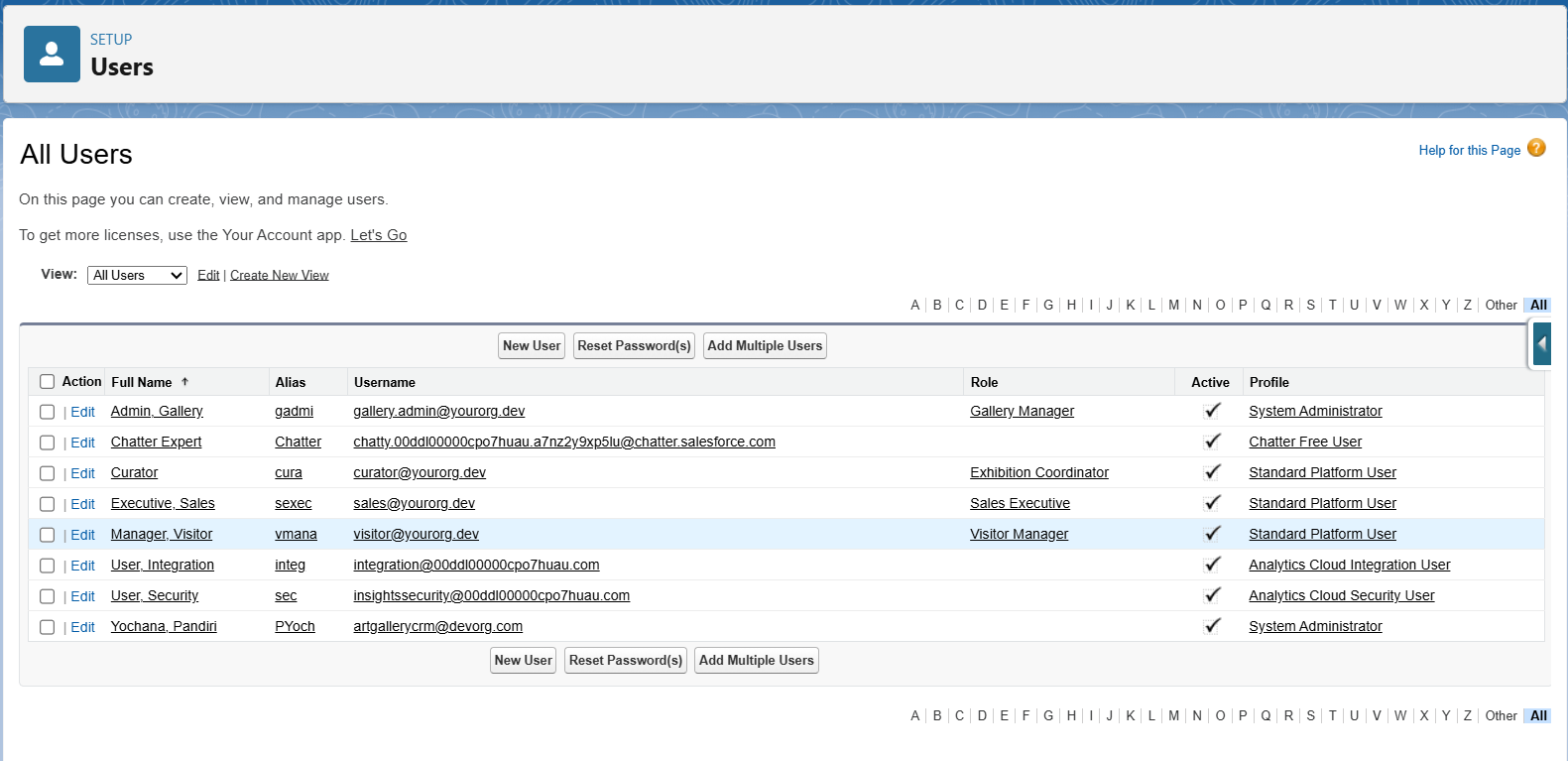
* Kept **Standard Fiscal Year (Jan–Dec)** → suitable for generating sales and revenue reports per quarter.
* Example: Quarterly sales of artworks per exhibition.

### **5. User Setup & Licenses**

Created users with different roles:

1. **Gallery Admin** – Full system control.
2. **Exhibition Coordinator** – Manages exhibitions and assigns artists.
3. **Sales Executive** – Handles artwork sales, generates invoices.
4. **Visitor Manager** – Manages visitor registrations & event RSVPs.

Each user assigned **Salesforce Platform License** for custom CRM usage.



### **6. Profiles**

* **Exhibition Coordinator Profile:** Can create/manage exhibitions and assign artists but restricted from altering sales data.
* **Sales Executive Profile:** Can create/edit artwork sales, but limited read-only access to exhibitions.
* **Visitor Manager Profile:** Manages visitor objects but no access to revenue reports.
* **Admin Profile:** Full CRUD access across all objects.

### **7. Roles**

Defined role hierarchy to maintain data visibility:

* **Gallery Manager (Top Role)**
  + **Exhibition Coordinator**
  + **Sales Executive**
  + **Visitor Manager**

This ensures managers can see all data, while lower roles see only their relevant records.

### **8. Permission Sets**

Instead of editing profiles repeatedly:

* Created **"Reports Access" Permission Set** → assigned to Coordinators who need to analyse exhibitions.
* Created **"Artwork Pricing Update" Permission Set** → given to selected Sales Executives.

### **9. Org-Wide Defaults (OWD)**

Set record-level security:

* **Exhibitions:** Public Read Only → all users can view exhibitions but only Coordinator/Admin can edit.
* **Artists:** Public Read Only.
* **Artworks & Sales Records:** Private → only Sales Executives/Managers can view/edit their sales.
* **Visitor Registrations:** Private → only assigned Visitor Manager can see/edit.

### **10. Sharing Rules**

Added rules for collaboration:

* Allowed **Sales Executives** to view artwork sales created by other executives (for team collaboration).
* Coordinators can view **visitor registrations** for exhibitions they manage.

### **11. Login Access Policies**

* Restricted **Visitor Manager & Sales Executives login access** → 10 AM – 7 PM (business hours).
* Admin has unrestricted login.

### **12. Developer Org Setup**

This Dev Org acted as the **sandbox environment** where I configured:

* Custom Objects (Exhibition, Artist, Artwork, Visitor Registration, Sales).
* Roles, Profiles, and Sharing Rules.



### **13. Sandbox Usage (Conceptual)**

* In real implementation, all configs would be tested in **Sandbox** before pushing to production.
* Since Developer Edition was used, sandbox acted as the test environment.

### **14. Deployment Basics**

* For real org:
  + Move components (Objects, Workflows, Reports, Dashboards) using **Change Sets** to Production.
  + Versioning ensures stability before release.

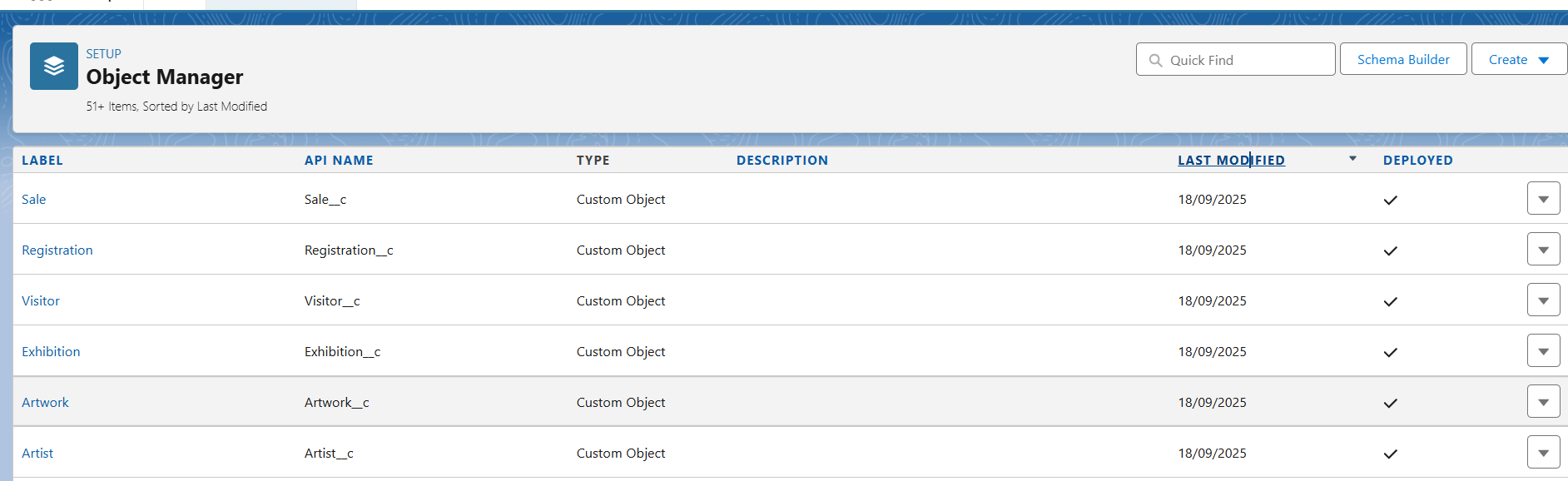
# Phase 3: Data Modeling & Relationships

👉 **Goal**: Define the data model (objects, fields, relationships) that forms the foundation of the Art Gallery CRM.

## ****1. Standard & Custom Objects****

We’ll use a mix of Salesforce **standard objects** and **custom objects** to capture all gallery operations.

* **Standard Objects**
  + **Contact** → used for Visitors (name, phone, email, etc.).
  + **User** → system users (Admin, Sales Executive, Curator, Visitor Manager).
  + **Account (optional)** → could be used if gallery wants to group corporate clients or sponsors.
* **Custom Objects**
  + **Artist\_\_c** → stores details of artists.
  + **Artwork\_\_c** → each piece of art, linked to Artist.
  + **Exhibition\_\_c** → gallery exhibitions/events.
  + **Sale\_\_c** → records sales of artworks.
  + **Visitor\_\_c** → visitor information (beyond standard contact fields).
  + **Registration\_\_c** → links Visitors with Exhibitions (attendance tracking).



## ****2. Fields****

Each custom object will have its own set of custom fields.

* **Artist\_\_c**
  + Name (Text)
  + Nationality (Picklist)
  + Specialty (Picklist: Painting, Sculpture, Photography, Digital Art, etc.)
  + Biography (Long Text Area)
* **Artwork\_\_c**
  + Title (Text)
  + Medium (Picklist: Oil, Acrylic, Marble, Bronze, etc.)
  + Price (Currency)
  + Status (Picklist: Available, Reserved, Sold, On Loan)
  + Artist (Lookup → Artist\_\_c)
* **Exhibition\_\_c**
  + Name (Text)
  + Start Date (Date)
  + End Date (Date)
  + Location (Text)
  + Theme (Text)
  + Coordinator (Lookup → User)
* **Sale\_\_c**
  + Sale Date (Date)
  + Sale Price (Currency)
  + Buyer (Lookup → Contact)
  + Artwork (Lookup → Artwork\_\_c)
  + Artist Commission (Formula: % of Sale Price)
* **Visitor\_\_c**
  + Name (Text)
  + Email (Email)
  + Phone (Phone)
  + Membership Status (Picklist: Regular, Premium, Guest)
* **Registration\_\_c**
  + Visitor (Lookup → Visitor\_\_c)
  + Exhibition (Lookup → Exhibition\_\_c)
  + Registration Date (Date)
  + Ticket Type (Picklist: Free, VIP, Student, Early Bird)

## ****3. Record Types****

Record Types let us handle different scenarios within one object.

* **Exhibition\_\_c** →
  + **Solo Exhibition** (single artist show)
  + **Group Exhibition** (multiple artists)
* **Sale\_\_c** →
  + **Direct Sale** (sold at gallery)
  + **Auction Sale** (sold at event/auction)

## ****4. Page Layouts****

Page layouts define what users see.

* **Artist\_\_c Page Layout**
  + Artist Info → Name, Nationality, Specialty, Bio
  + Related List: Artworks
* **Artwork\_\_c Page Layout**
  + Artwork Info → Title, Medium, Year, Status
  + Related Lists: Artist, Sales History, Exhibitions shown in
* **Exhibition\_\_c Page Layout**
  + Exhibition Info → Name, Dates, Theme, Location
  + Related Lists: Assigned Artworks, Visitor Registrations
* **Sale\_\_c Page Layout**
  + Sale Info → Sale Date, Price, Buyer
  + Related: Artwork details
* **Visitor\_\_c Page Layout**
  + Visitor Info → Name, Membership Status
  + Related Lists: Registrations
* **Registration\_\_c Page Layout**
  + Registration Info → Exhibition, Ticket Type
  + Related: Visitor Details

## ****5. Compact Layouts (for mobile & list views)****

Compact layouts summarize key fields.

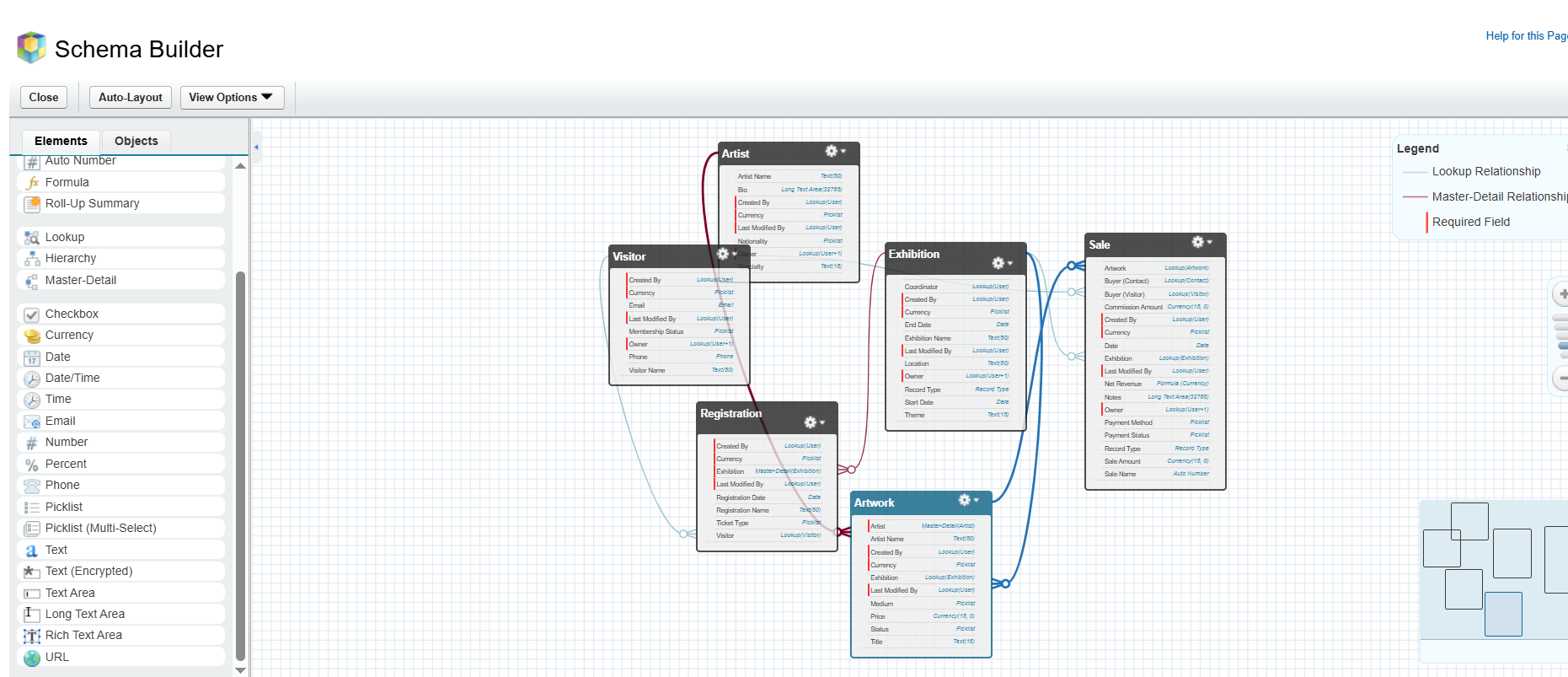
* **Artwork\_\_c Compact Layout**
  + Title, Medium, Price, Status
* **Exhibition\_\_c Compact Layout**
  + Name, Start Date, End Date, Location
* **Sale\_\_c Compact Layout**
  + Sale Date, Sale Price, Artwork

## ****6. Schema Builder****

Use Schema Builder (Setup → Schema Builder) to **visualize the relationships**.

* Artist\_\_c → Artwork\_\_c (Lookup, one Artist to many Artworks).
* Artwork\_\_c → Sale\_\_c (Lookup, one Artwork to one Sale).
* Exhibition\_\_c → Registration\_\_c (Lookup, one Exhibition to many Registrations).
* Visitor\_\_c → Registration\_\_c (Lookup, one Visitor can register for many Exhibitions).

This gives a clear **entity-relationship diagram (ERD)** for the project.



## ****7. Lookup vs Master-Detail vs Hierarchical****

Choose the right relationship type:

* **Artist\_\_c ↔ Artwork\_\_c** → Master-Detail (if Artist is deleted, their artworks should be deleted).
* **Artwork\_\_c ↔ Sale\_\_c** → Lookup (Sale record exists only if artwork is sold, so artwork may exist independently).
* **Exhibition\_\_c ↔ Registration\_\_c** → Master-Detail (a Registration cannot exist without an Exhibition).
* **Visitor\_\_c ↔ Registration\_\_c** → Lookup (a Visitor can exist even if they haven’t registered for any exhibition).

## ****8. Junction Objects****

If an **Artwork** could appear in multiple Exhibitions, we’d need a **junction object** (e.g., ArtworkExhibition\_\_c).  
For now, assume each artwork belongs to one exhibition at a time, so no junction object is needed.

## ****9. External Objects****

If the gallery integrates external systems:

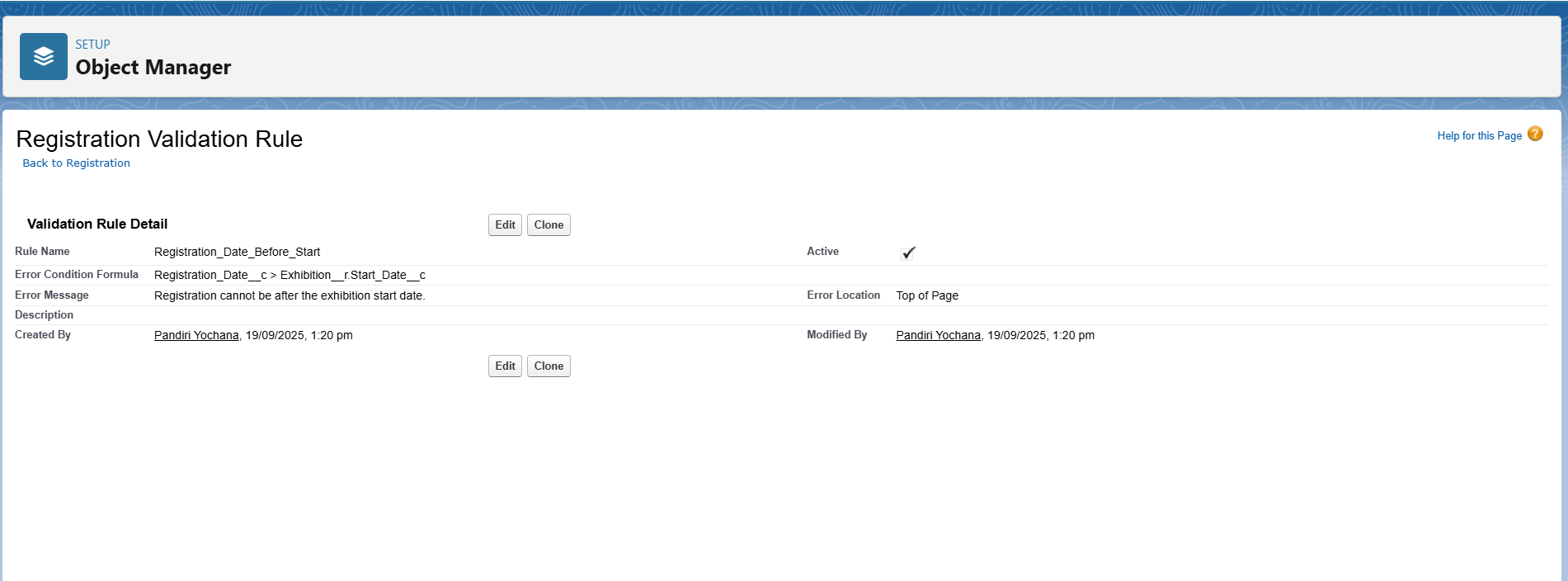
* Example: Insurance database for artwork insurance policies.
* Use **External Objects** in Salesforce (via Salesforce Connect) to reference data without storing it locally.

# Phase 4: Process Automation (Admin)

## ****1. Validation Rules****

Validation rules ensure **data quality**.  
Examples for your objects:

* **Exhibition\_\_c**
  + End\_Date\_\_c > Start\_Date\_\_c (Exhibition cannot end before it starts).
* **Artwork\_\_c**
  + If **Status = "Sold"**, then **Price must not be blank**.
  + If **Medium = Marble or Bronze**, then **Price > 10,000** (to enforce realistic pricing).
* **Sale\_\_c**
  + Sale\_Price\_\_c > 0.
* **Registration\_\_c**
  + **Registration\_Date\_\_c ≤ Exhibition.Start\_Date\_\_c** (visitors cannot register after the exhibition starts).



## ****2. Workflow Rules (legacy, optional)****

⚠️ Salesforce recommends **Flow**, but you can still note workflows.  
Examples:

* When a **Visitor\_\_c** record is created, send a **Welcome Email**.
* When an **Artwork\_\_c** status changes to **Sold**, notify the Curator.

## ****3. Process Builder (legacy, replaced by Flow)****

Process Builder is being retired, but for completeness:

* Could auto-update **Artwork\_\_c.Status → Sold** when a **Sale\_\_c** record is created.
* Could auto-update **Visitor Membership Status** if they purchase art above ₹1,00,000 (upgrade to Premium).

👉 Instead, we’ll implement these in **Flow**.

## ****4. Approval Process****

Approvals are critical in galleries for **high-value transactions**.  
Example:

* **Sale\_\_c → Approval Process**
  + Condition: If Sale\_Price\_\_c > 50,000.
  + Approval goes to **Gallery Manager**.
  + Actions:
    - **Before Approval**: Status = "Pending".
    - **After Approval**: Status = "Confirmed", trigger email to Buyer.
    - **If Rejected**: Status = "Declined", send rejection email to Buyer.

## 

## ****5. Flow Builder (Recommended Tool)****

Flows will be the main automation engine.

### **a) Record-Triggered Flows**

* **Artwork Sale Flow**
  + When a **Sale\_\_c** is created, automatically update the related **Artwork\_\_c.Status → Sold**.
* **Exhibition Capacity Flow**
  + Count Registrations for an Exhibition. If Registrations exceed 200, update Exhibition.Status = "Full".
* **Commission Calculation Flow**
  + Automatically calculate **Artist Commission = Sale Price × 30%** (or dynamic % based on Artist profile).

### **b) Scheduled Flows**

* Daily check for **Exhibitions starting in 7 days** → Send reminders to registered Visitors.

### **c) Screen Flows**

* **Exhibition Registration Screen Flow**
  + A form where a Visitor can pick an Exhibition and Ticket Type.
  + Automatically creates **Registration\_\_c** and sends confirmation email.
* **Artwork Intake Form**
  + For Curators to add a new Artwork, selecting Artist, Medium, Price, etc.

## ****6. Email Alerts****

Examples:

* When a **Sale\_\_c** is approved, send an **Invoice Email** to the Buyer.
* When a Visitor registers for an Exhibition, send a **Ticket Confirmation Email**.
* Send **Reminder Emails** to Visitors 2 days before an Exhibition.

## ****7. Field Updates****

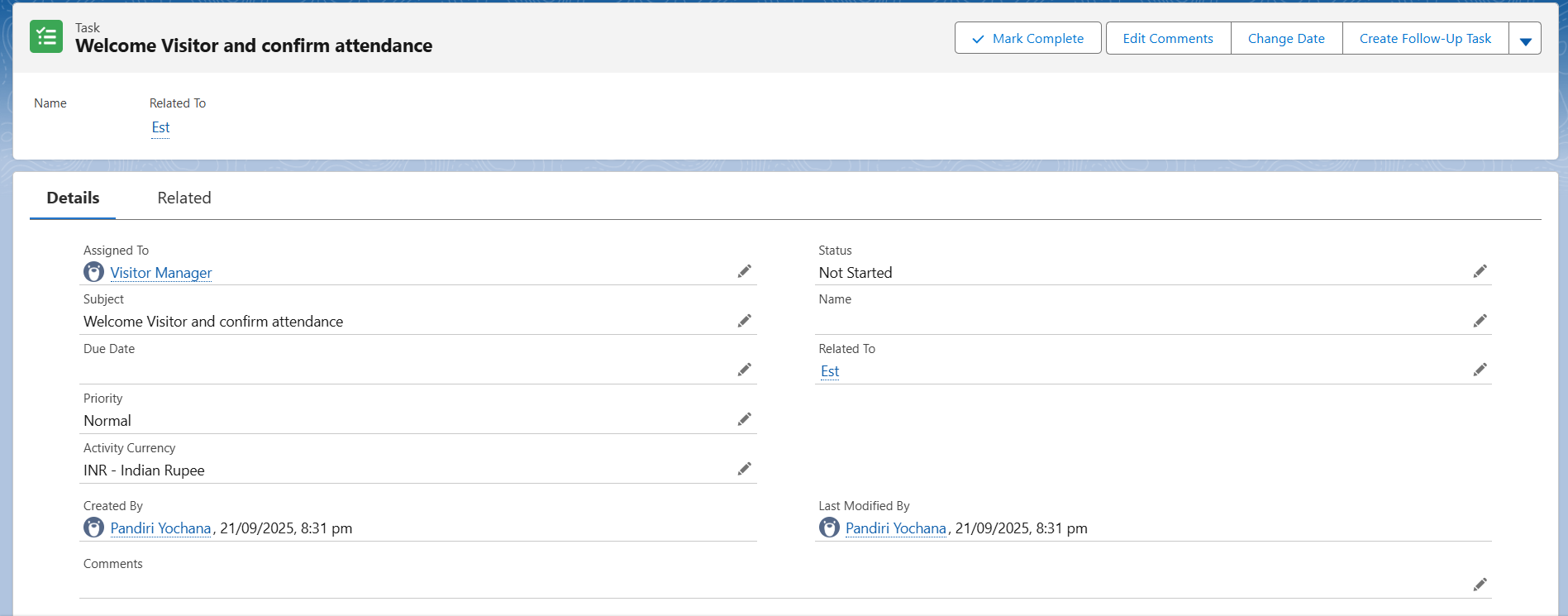
Examples:

* After a **Sale\_\_c** is approved, set **Sale\_Status\_\_c = Confirmed**.
* After **Exhibition End Date**, auto-update **Exhibition.Status\_\_c = Completed**.
* When a Visitor purchases > 5 Artworks, auto-update **Visitor.Membership\_Status = Premium**.

## ****8. Tasks****

Create automatic follow-up tasks:

* After a **Sale\_\_c** is confirmed → Create a task for **Finance Team**: "Prepare Invoice".
* After a **Registration\_\_c** is created → Create a task for **Visitor Manager**: "Welcome Visitor and confirm attendance".
* After a new **Artwork\_\_c** is added → Task for **Curator**: "Verify authenticity and details".



## ****9. Custom Notifications****

Push **real-time in-app notifications** to users.  
Examples:

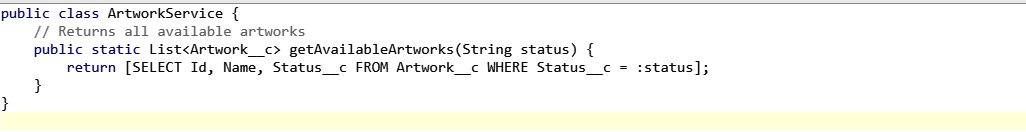
* Notify Curators when a **new Artwork** is submitted.
* Notify Sales Executive when a **Visitor upgrades to Premium**.
* Notify Exhibition Coordinator when **Registrations exceed 80% capacity**.

# ****Phase 5: Apex Programming (Developer) – Art Gallery CRM****

# ****Goal:**** Add advanced logic and automation to the Art Gallery CRM to handle bookings, sales, exhibitions, and related processes using Apex programming.

## ****1️. Classes & Objects****

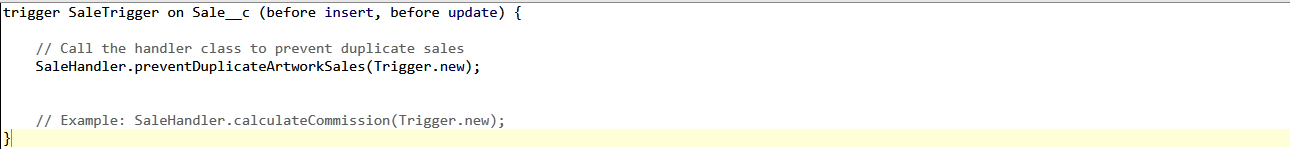
* **Purpose:** Centralize reusable logic in Apex classes.
* **Example:**
  + Create a class ArtworkService to handle common operations on Artwork, Sale, and Exhibition records.



* **Benefit:** Avoid duplicating SOQL queries across triggers and flows.

## ****2️. Apex Triggers****

* **Example:** Trigger to prevent duplicate or overlapping exhibition bookings for a single Artwork.
* **Logic:**
  + On Sale\_\_c insert/update → check if the Artwork is already sold or booked for another exhibition in the same date range.



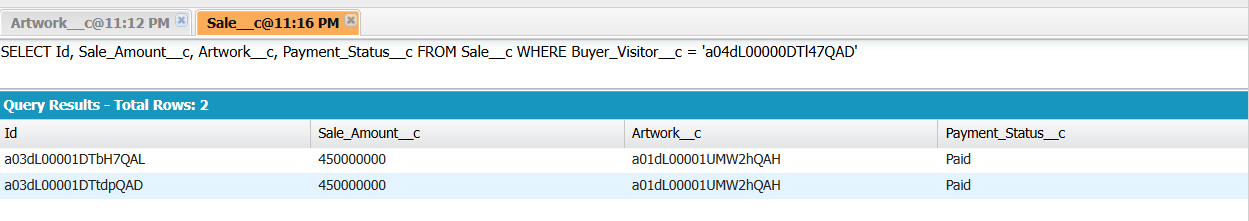
## ****3️. Trigger Design Pattern****

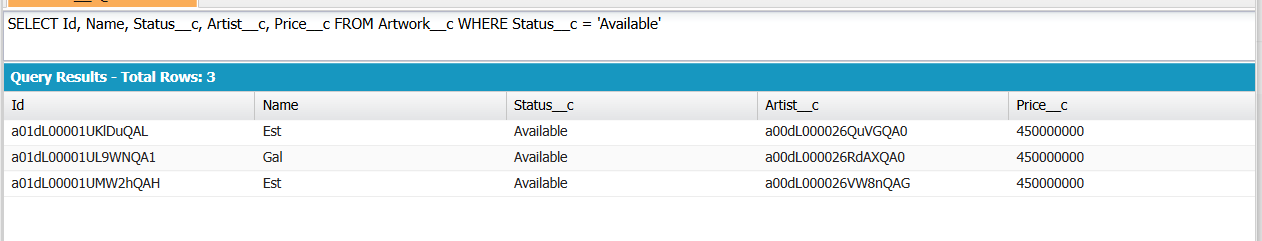
* **Handler Class:** Keep triggers clean; delegate logic to a class.



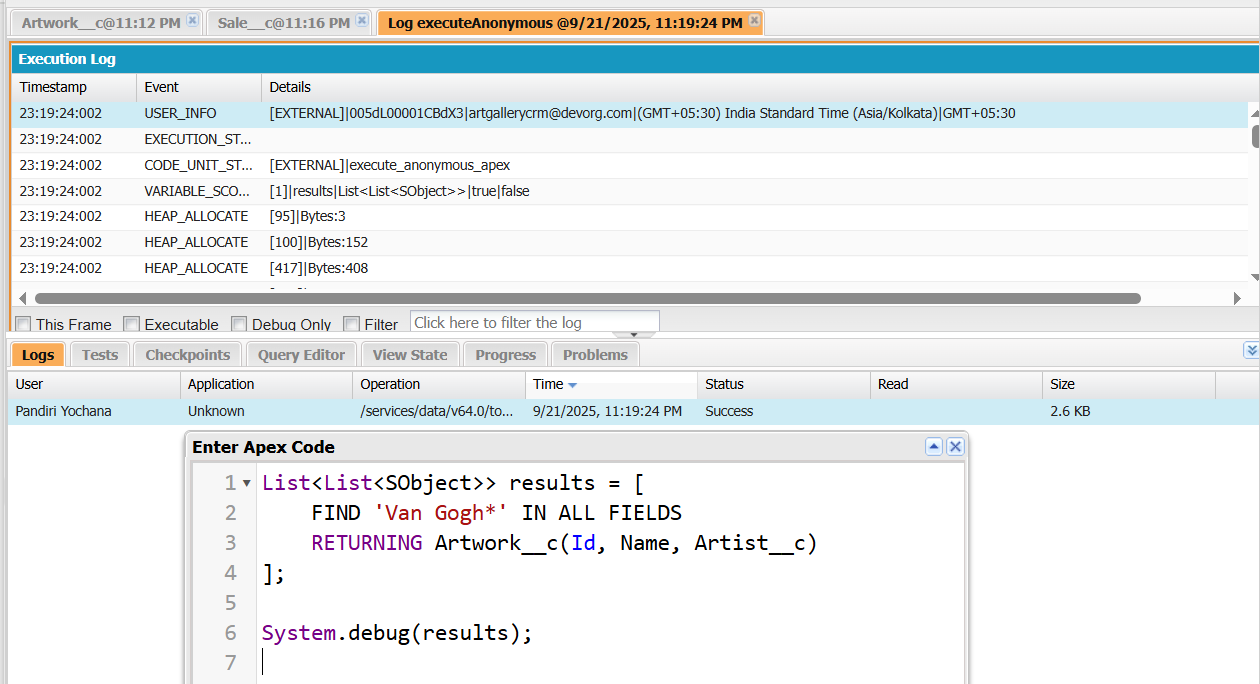
## ****4️. SOQL & SOSL Queries****

* **SOQL Example:** Fetch all available artworks for sale or exhibitions.
* List<Artwork\_\_c> availableArtworks = [SELECT Id, Name FROM Artwork\_\_c WHERE Status\_\_c = 'Available'];



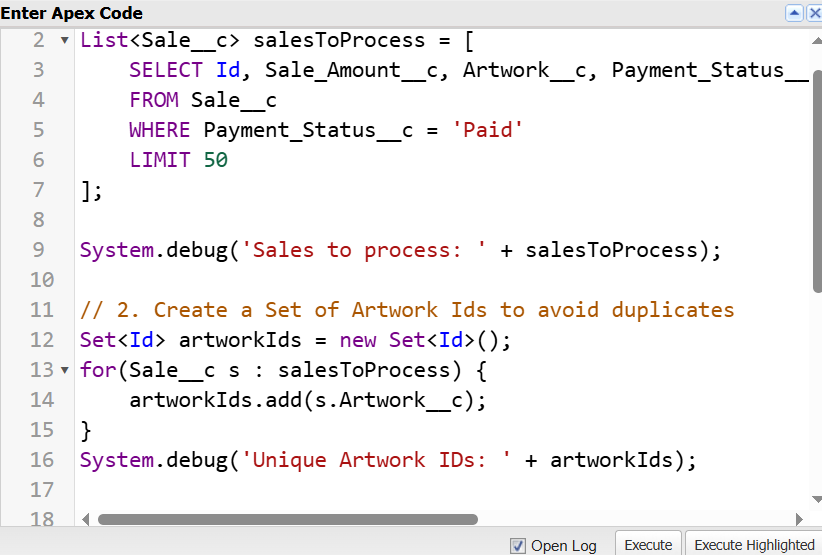


* **SOSL Example:** Search for artworks or artists by keyword.
* List<List<SObject>> searchResults = [FIND 'Van Gogh\*' IN ALL FIELDS RETURNING Artwork\_\_c(Id, Name), Artist\_\_c(Id, Name)];

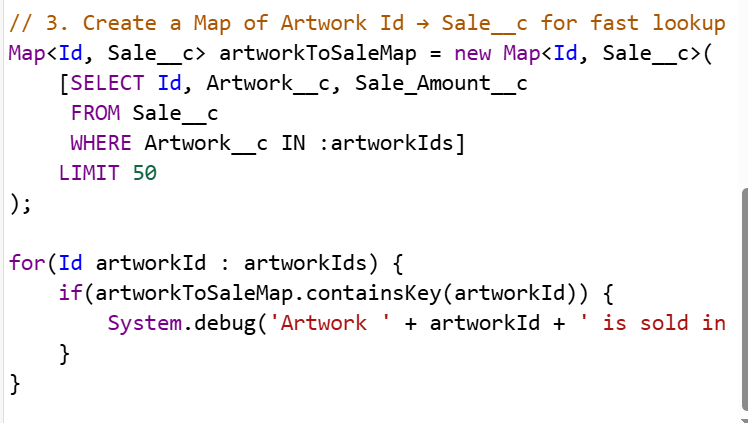


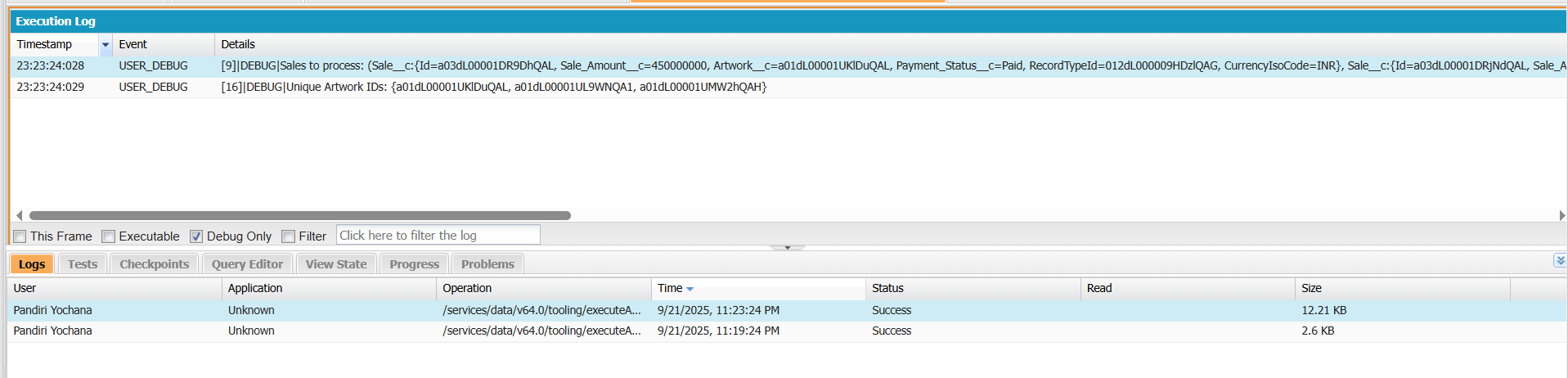
## ****5️. Collections: List, Set, Map****

* **Set Example:** Avoid duplicate Artwork Ids when processing sales or exhibition bookings.

**Map Example:** Map Artwork Id → Sale record for faster lookups.

* Map<Id, Sale\_\_c> artworkToSaleMap = new Map<Id, Sale\_\_c>([SELECT Id, Artwork\_\_c FROM Sale\_\_c WHERE Artwork\_\_c IN :artworkIds]);



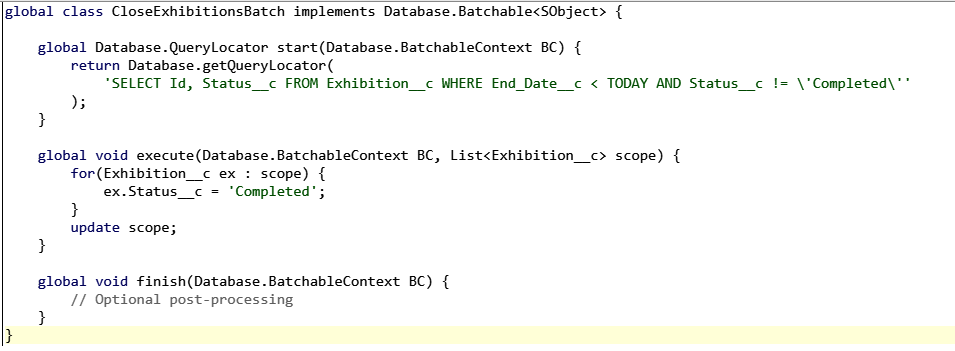


## ****6.Control Statements****

* **Example:** Validate booking or sale rules.

**7️. Batch Apex**

* **Purpose:** Handle large datasets asynchronously.
* **Example:** Nightly job to mark **exhibitions as completed** if their end date has passed.



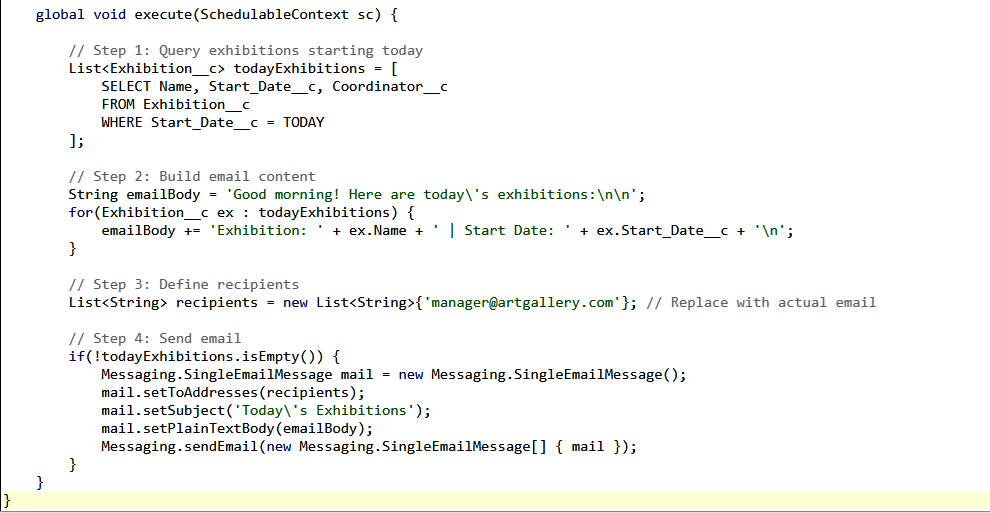
## ****8️. Queueable Apex****

* **Purpose:** Async processing for bulk operations.
* **Example:** Calculate and update discounts for bulk art sales.



**9️. Scheduled Apex**

* **Purpose:** Automate daily tasks.
* **Example:** Email gallery manager the list of **today’s exhibitions or sales** every morning.



**10. Future Methods**

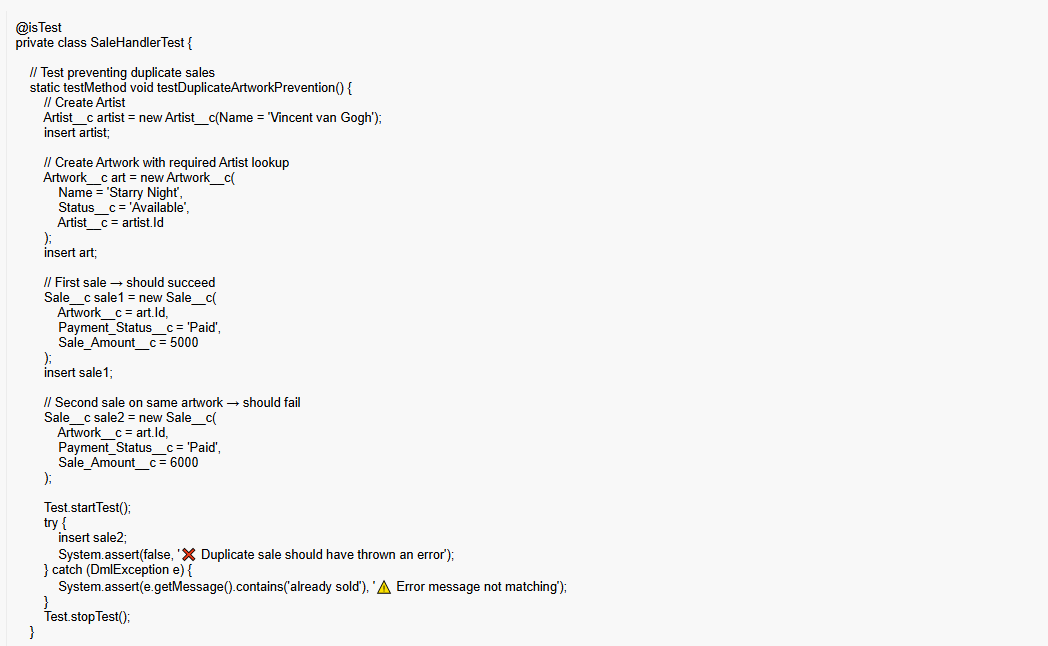
* **Purpose:** Call external APIs asynchronously.
* **Example:** Call an external **insurance API** to insure newly sold artworks.

## ****11️. Exception Handling****

* **Purpose:** Catch and handle errors gracefully.
* **Example:** Prevent overlapping bookings or sales.

## ****12️. Test Classes****

* **Purpose:** Ensure code coverage and validate logic.



## ****13️. Asynchronous Processing Combined****

* Batch + Queueable + Future = full async automation:
  + Batch to close old exhibitions
  + Queueable to calculate discounts
  + Future to call external insurance API

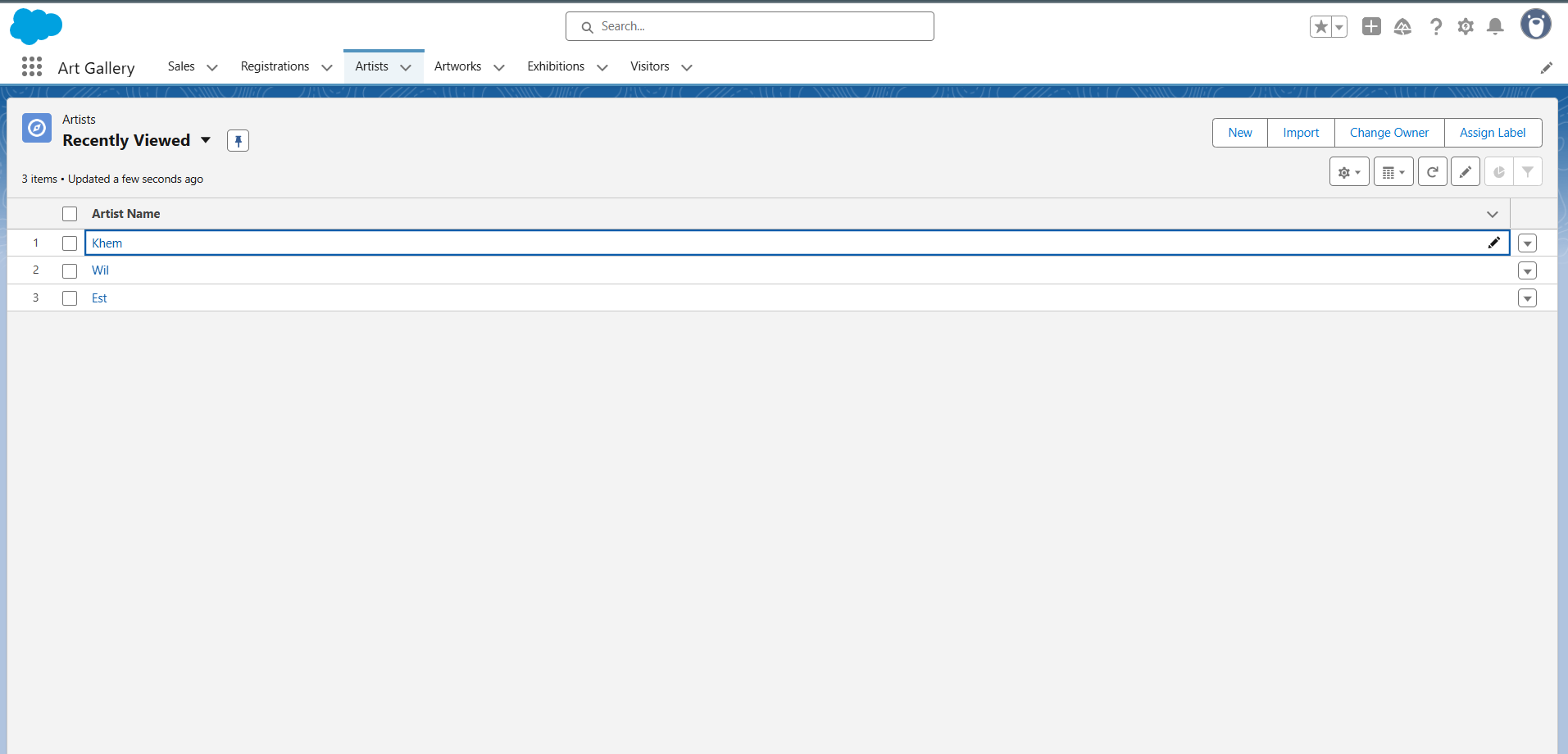
This **ensures high-performance processing** without hitting governor limits.

# Phase 6: User Interface Development (Art Gallery CRM)

👉 **Goal:** Build a user-friendly Salesforce Lightning experience for Curators, Managers, and Visitors.

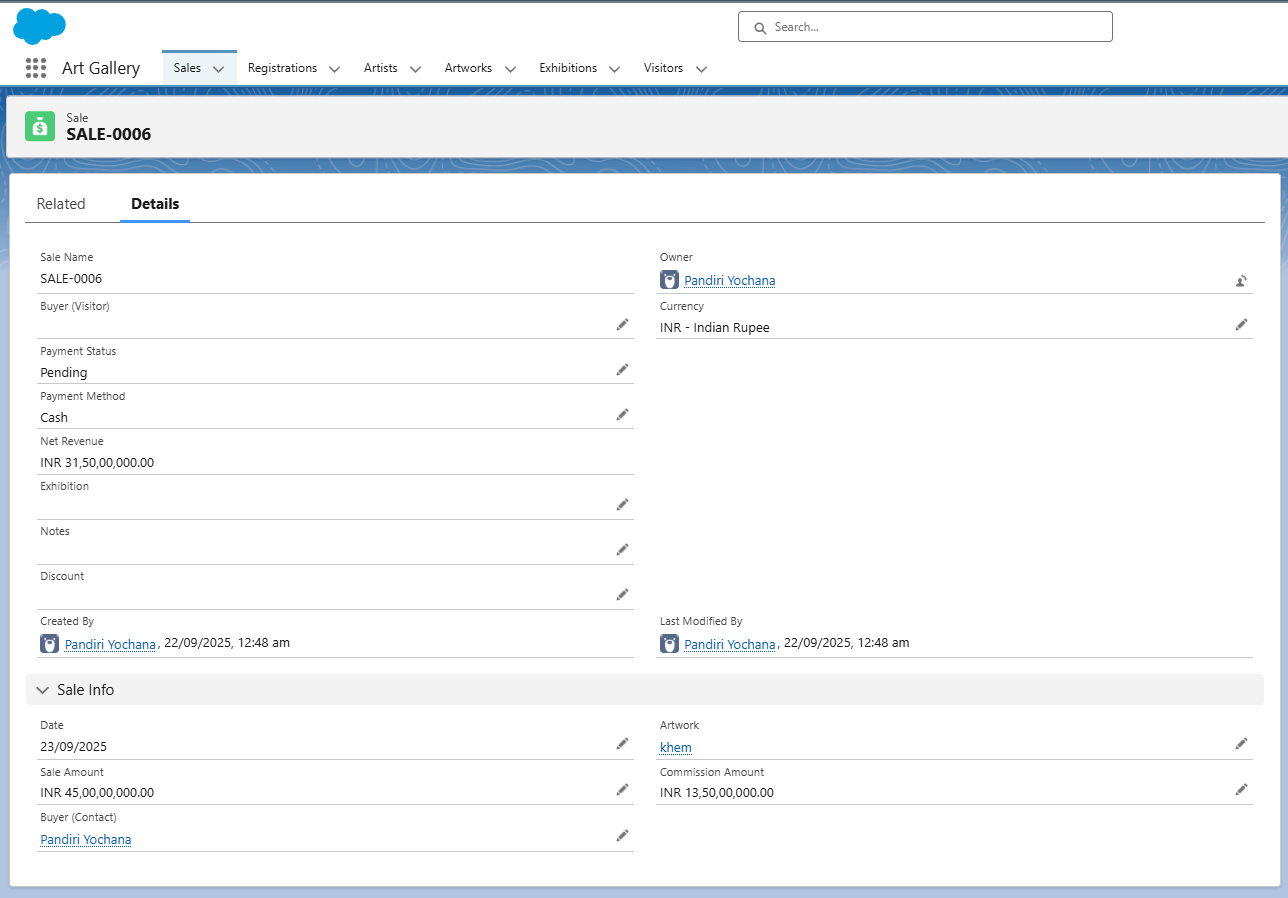
## 1️. Lightning App Builder → Create "Art Gallery CRM" App

* Go to **Setup → App Manager → New Lightning App**.
* Name: **Art Gallery CRM**
* Add **Navigation Items**: Artworks, Exhibitions, Sales, Visitors, Artists.
* Branding: Add gallery logo + theme color.



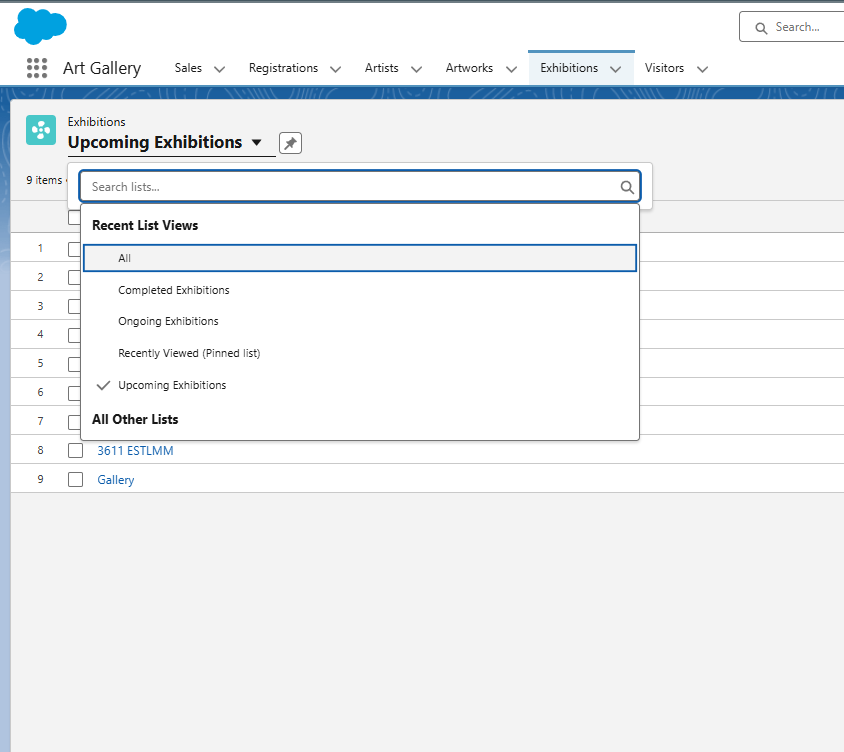
## 2️. Record Pages (Lightning Record Pages)

* **Artwork Page**
  + Show artwork details (Name, Artist, Price, Status).
  + Related List → Sales (if artwork sold), Exhibitions (where artwork was displayed).
* **Exhibition Page**
  + Show event details (Name, Venue, Start/End Date, Status).
  + Related List → Participating Artworks, Visitor Registrations.
* **Sale Page**
  + Display Buyer details, Sale Amount, Payment Status.
  + Related List → Invoice, Artwork.



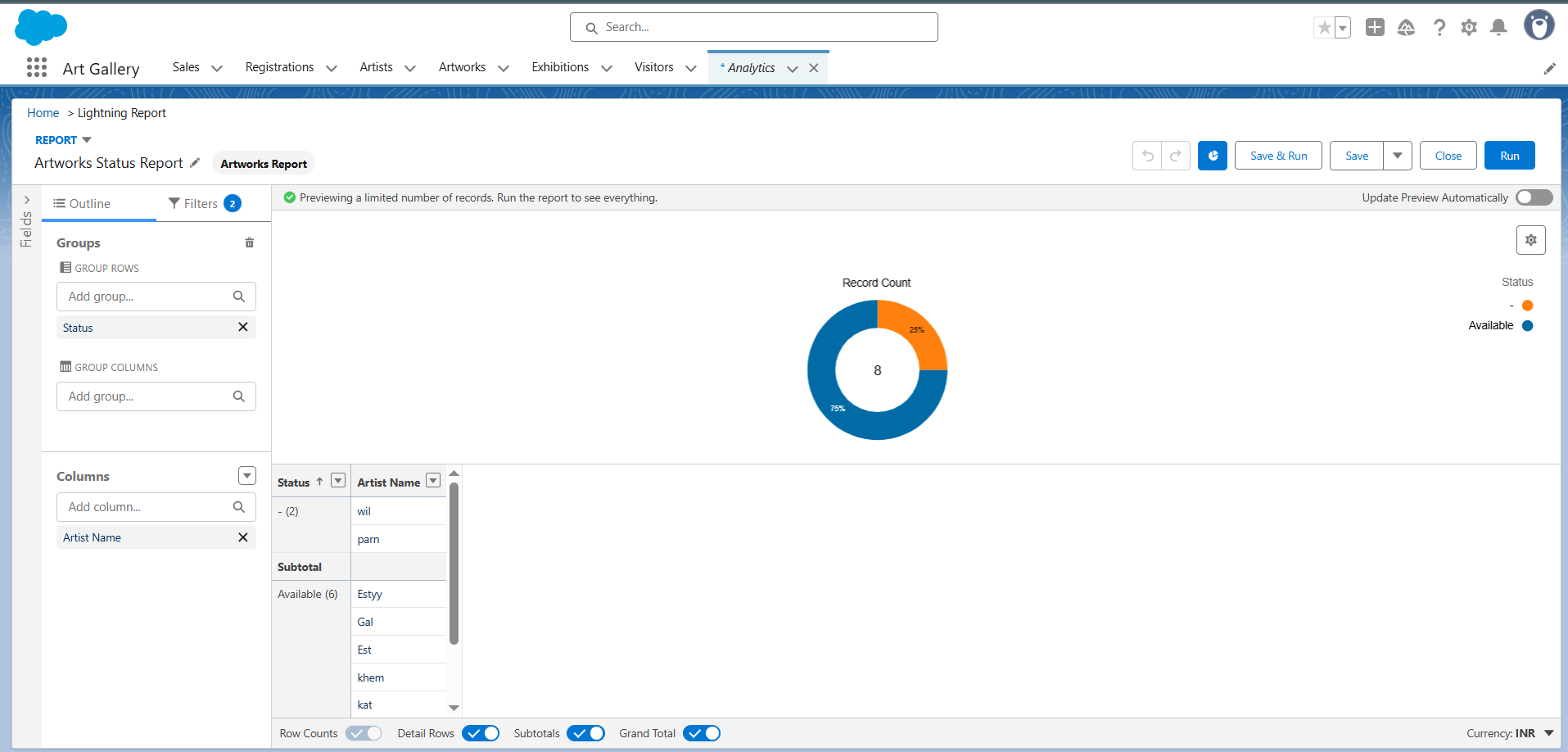
## 3️. Tabs

* **Artworks Tab** → list of all artworks.
* **Exhibitions Tab** → list of exhibitions.
* **Sales Tab** → track completed and pending sales.
* **Visitors Tab** → track buyers & exhibition attendees.
* **Artists Tab** → list of all artists with linked artworks.



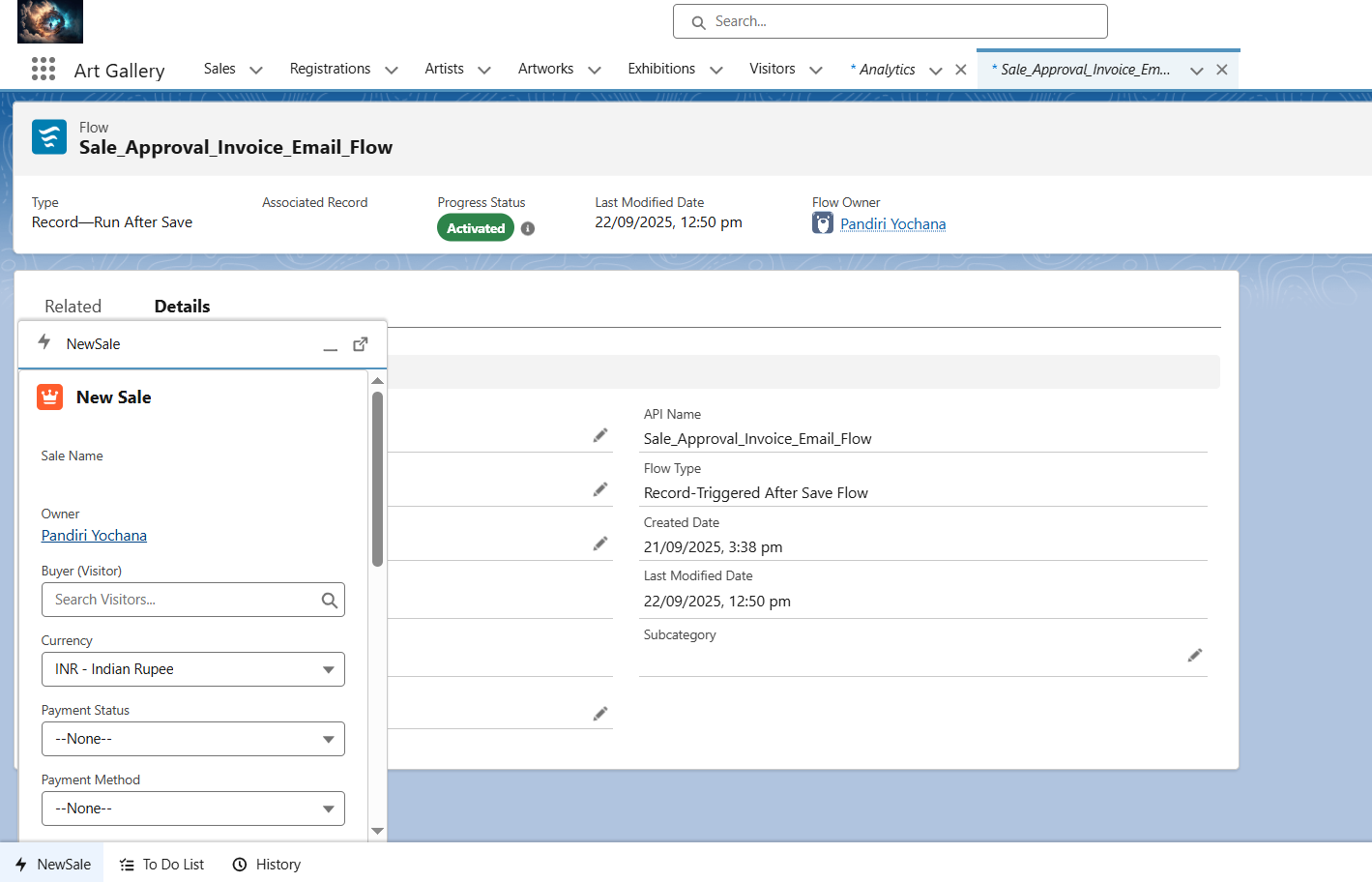
## 4️. Home Page Layout

* Create a **Gallery Dashboard** with components:
  + **KPI Tiles:**
    - Total Artworks
    - Total Sales (this month)
    - Ongoing Exhibitions
    - Visitors Registered
  + **Report Charts:**
    - Pie chart: Artwork Status (Available vs Sold)
    - Bar chart: Sales by Artist
    - Line chart: Visitors trend by Exhibition



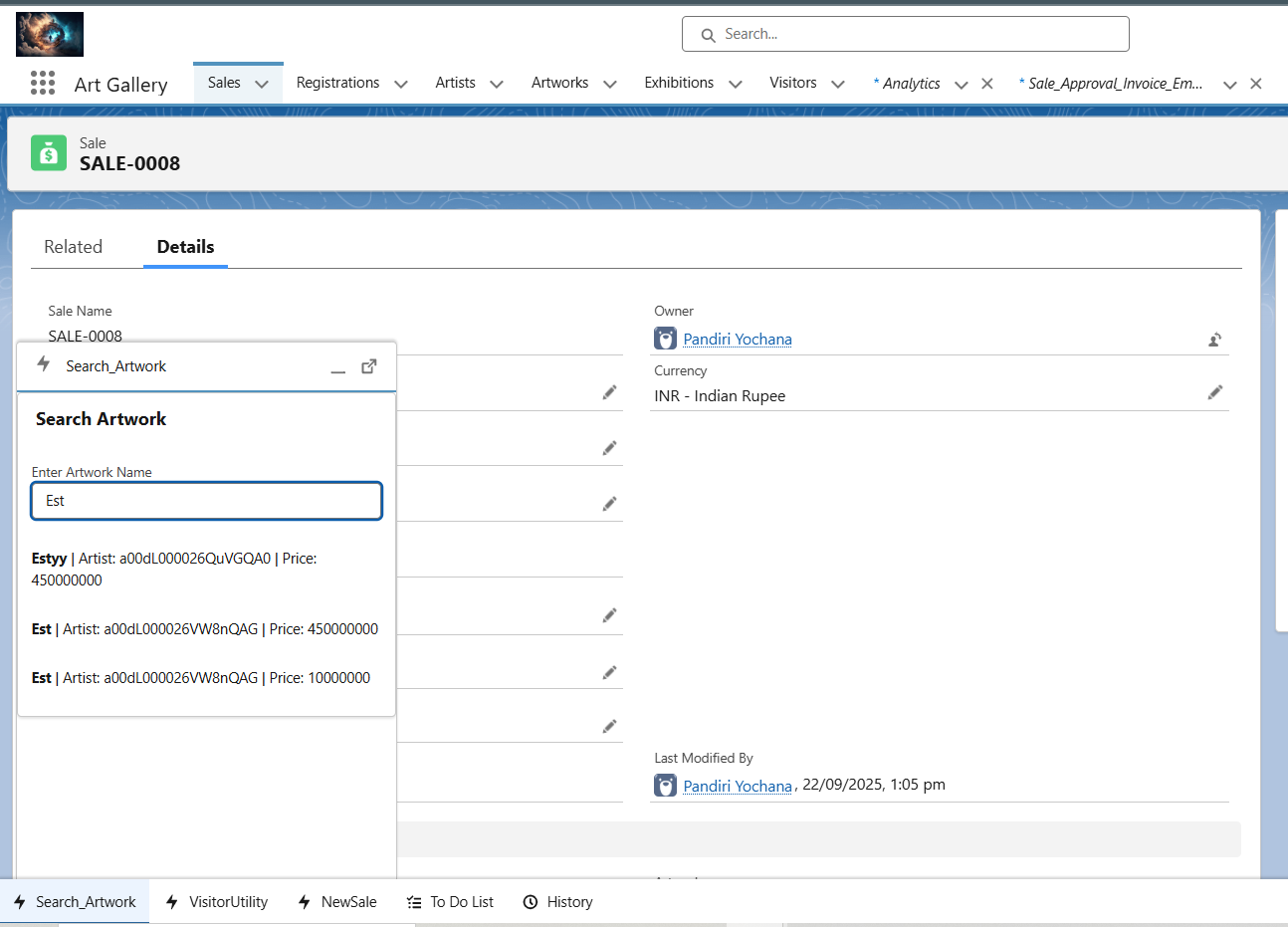
## 5️. Utility Bar

* Add quick actions for users:
  + **New Sale** (quickly log a new sale).
  + **Register Visitor** (add visitor to an exhibition).
  + **Search Artwork** (find available artworks by name or artist).
  + **Send Email** (to buyer or visitor directly).



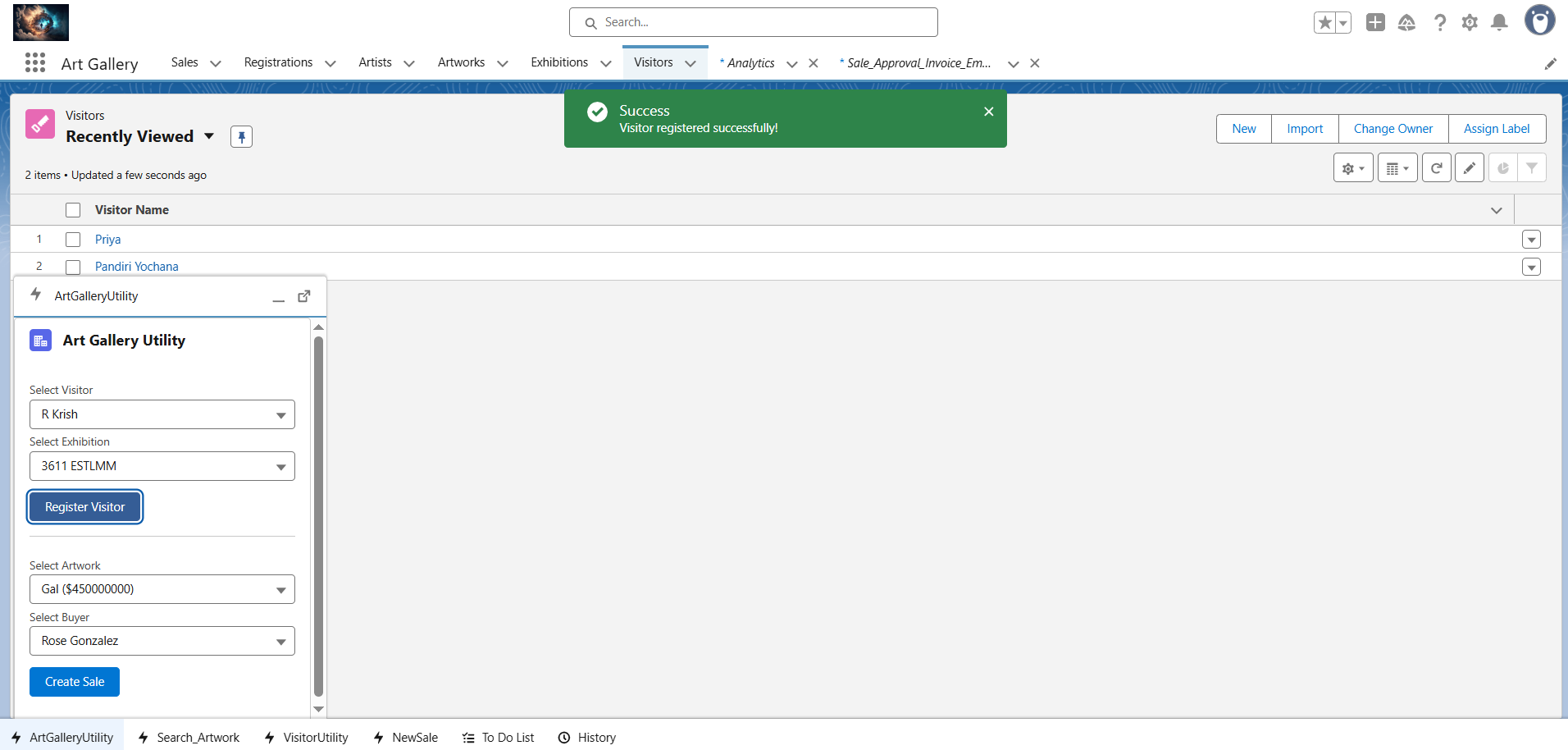
## 6️. Lightning Web Components (LWC)

* **Artwork Search Component**
  + Input: Artwork name or Artist.
  + Output: List available artworks in a **datatable** with fields (Name, Price, Status, Artist).
* **Exhibition Registration Component**
  + Input: Visitor details + Exhibition selection.
  + Button: "Register".
  + Output: Confirmation message.



## 7️. Apex with LWC

* Create Apex methods for LWC:
  + getAvailableArtworks(status) → return artworks with given status.
  + registerVisitor(visitorId, exhibitionId) → create registration record.
  + createSale(artworkId, buyerId) → insert Sale record and trigger email flow.



## 8️. Events in LWC

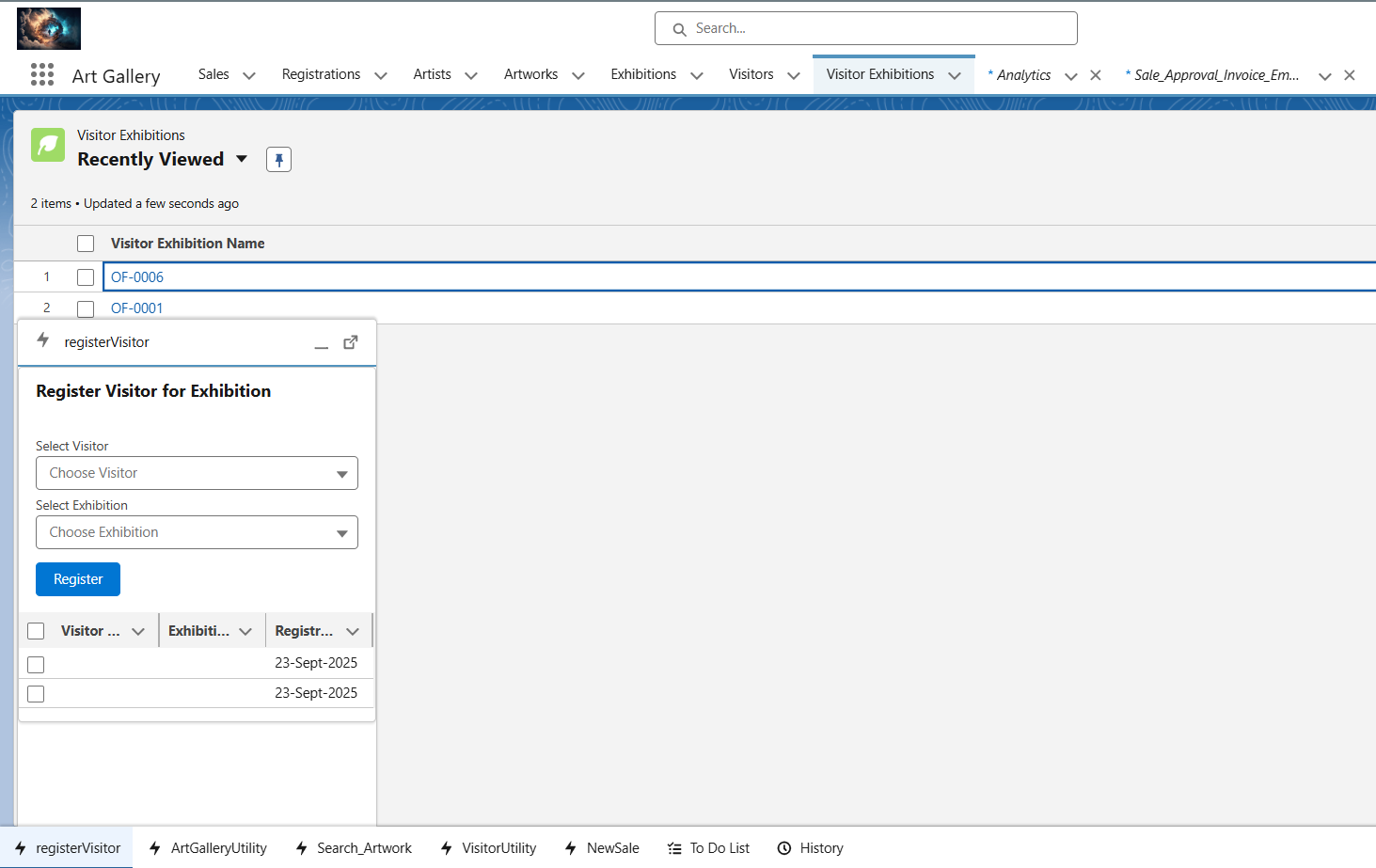
* **Child Component (Search Form):** Takes user input for search keyword (Artwork/Artist).
* **Parent Component (Results):** Displays results in a datatable.
* Event → when user hits **Search**, child sends keyword to parent, parent calls Apex to fetch data.

## 9️. Wire Adapters

* Use **@wire** with SOQL to auto-fetch data:
  + Example: Show all **Available Artworks** without refreshing.
  + Example: Show **Ongoing Exhibitions** for Visitors.

## 10. Imperative Apex Calls

* When user clicks **Book Artwork / Register for Exhibition / Buy Now**, call Apex imperatively:



## 11. Navigation Service

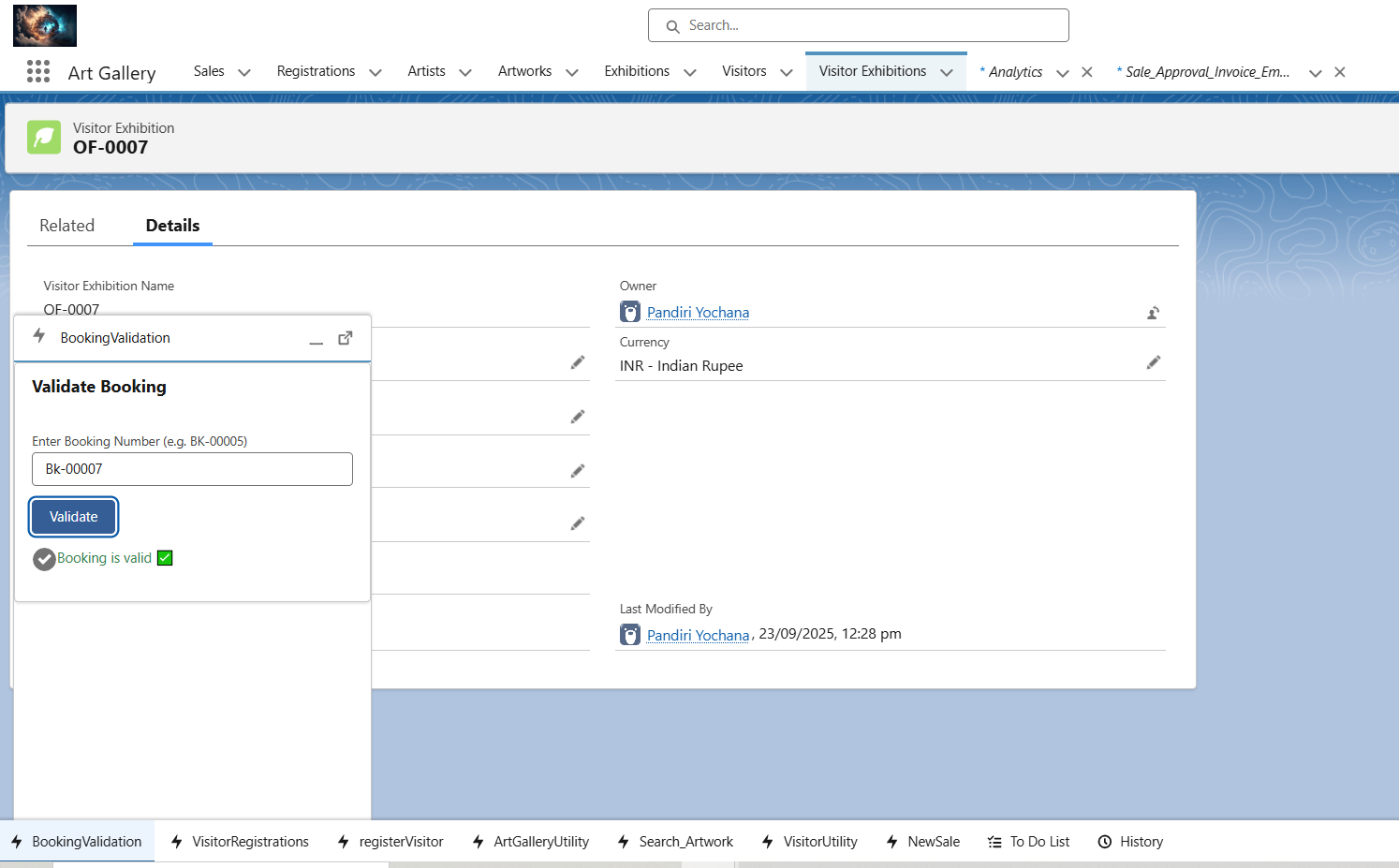
* After actions like "New Sale" or "Exhibition Registration":
  + Automatically redirect user to the **record page**.  
    Example: After creating a sale → navigate to Sale\_\_c record detail page.

# ****Phase 7: Integration & External Access****

**Goal:** Connect the Art Gallery CRM to external systems, APIs, and ensure secure, seamless data exchange for better operational efficiency.

## ****1️. Named Credentials****

* **Purpose:** Store credentials (API keys, usernames, passwords, OAuth tokens) securely for external services.
* **Example Use Case:**
  + Connect to a **ticketing platform API** to validate visitor bookings.
  + Connect to **payment gateways** (like Stripe or PayPal) to verify completed transactions.
* **Salesforce Action:**
  + Setup → Security → Named Credentials.
  + Store the endpoint and authentication type (e.g., OAuth 2.0 or Basic Auth).
* **Benefit:** Developers don’t hardcode sensitive information in Apex.



## ****2️. External Services****

* **Purpose:** Easily connect to external REST or SOAP services using declarative tools.
* **Example Use Case:**
  + Connect to an **external art valuation service** to fetch estimated artwork prices automatically.
  + Check artist information from an **artist registry API**.
* **Salesforce Action:**
  + Setup → External Services → Import API schema (Swagger/OpenAPI).
  + Generate Apex actions for declarative Flows or Apex integration.

## ****3️. Web Services (REST/SOAP)****

* **Purpose:** Integrate via programmatic endpoints for sending/receiving data.
* **Example Use Case:**
  + **REST callout:** Check ticket availability in a third-party exhibition booking system.
  + **SOAP callout:** Fetch artwork authenticity certificates from external systems.
* **Salesforce Action:**
  + Use @future(callout=true) or Queueable Apex to make asynchronous callouts.
  + Parse JSON/XML responses and store them in Salesforce objects (like Artwork\_\_c or Visitor\_\_c).

## ****4️. Callouts Triggered by Records****

* **Purpose:** Automate integration when certain records are created/updated.
* **Example Use Case:**
  + When a **new exhibition is created**, send its details to the **marketing portal**.
  + When a **visitor registers**, validate ticket via external system.
* **Salesforce Action:**
  + Trigger → Call Apex class → Call external API.
  + Ensure error handling and retries for reliable integration.



## ****5️. Platform Events****

* **Purpose:** Publish real-time events to external systems or internal processes.
* **Example Use Case:**
  + Publish a **Visitor Registered** event → external **email system** sends welcome email automatically.
  + Publish **Artwork Sold** event → triggers inventory system update.
* **Salesforce Action:**
  + Define Platform Event object (e.g., Visitor\_Registered\_\_e).
  + Apex triggers or Process Builder/Flow publish events.
  + Subscribers (external systems or Salesforce processes) listen and act on events.

## ****6️. Change Data Capture****

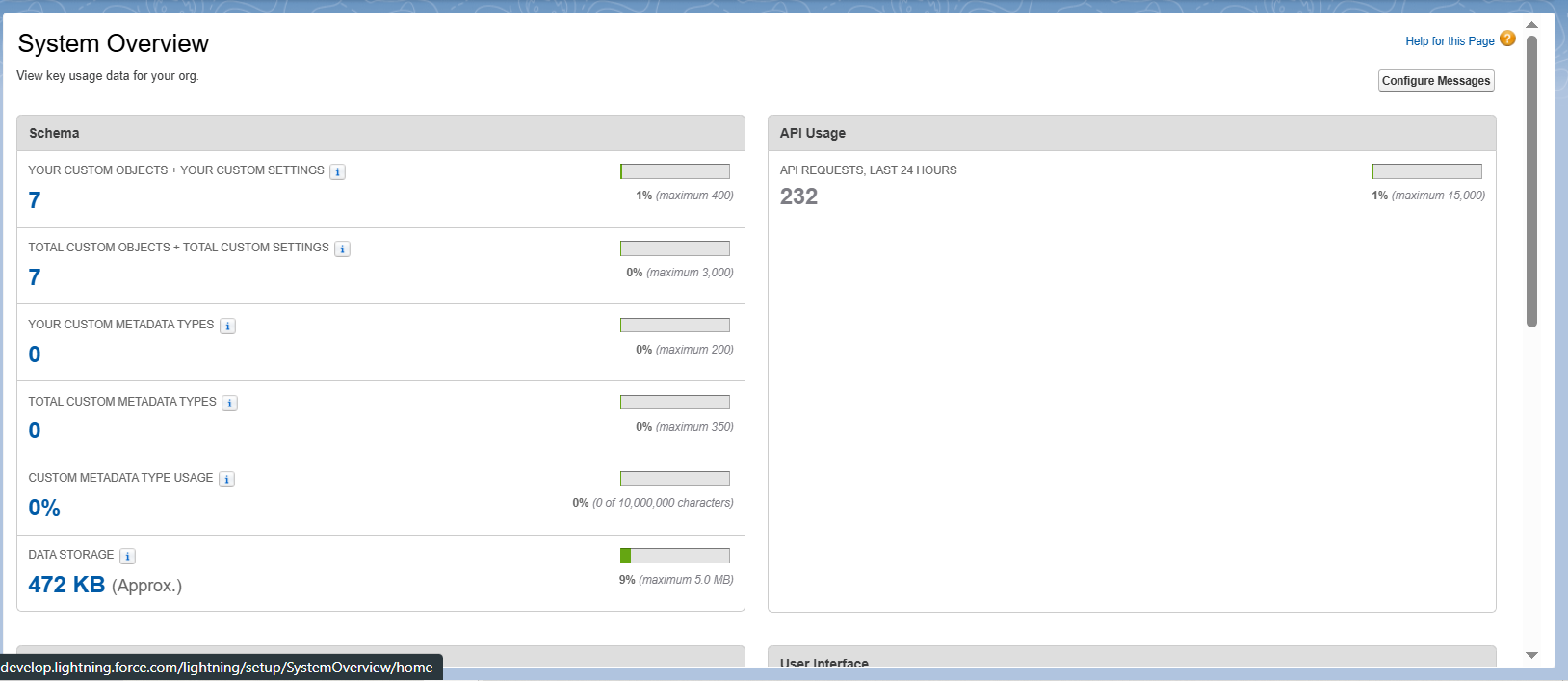
* **Purpose:** Track changes in Salesforce objects and notify external systems.
* **Example Use Case:**
  + When Exhibition\_\_c is updated → send updated schedule to **partner websites**.
  + When Artwork\_\_c price changes → update **pricing system** automatically.
* **Salesforce Action:**
  + Enable **Change Data Capture** for objects like Visitor\_\_c, Exhibition\_\_c, Sale\_\_c.
  + External system subscribes to these events using CometD API.

## ****7️. Salesforce Connect****

* **Purpose:** Access data stored outside Salesforce without copying it.
* **Example Use Case:**
  + Art gallery inventory is managed in an **external database**.
  + Use **Salesforce Connect** (OData) to view artworks in Salesforce without syncing.
* **Benefit:** Real-time data access, no storage duplication.

## ****8️. API Limits****

* **Purpose:** Monitor and manage Salesforce API usage to avoid hitting limits.
* **Example Use Case:**
  + Daily callouts to **ticketing system** should be tracked to avoid exceeding Salesforce API limits.
* **Salesforce Action:**
  + Setup → System Overview → API Usage.
  + Use custom monitoring or alerts for high callout usage.

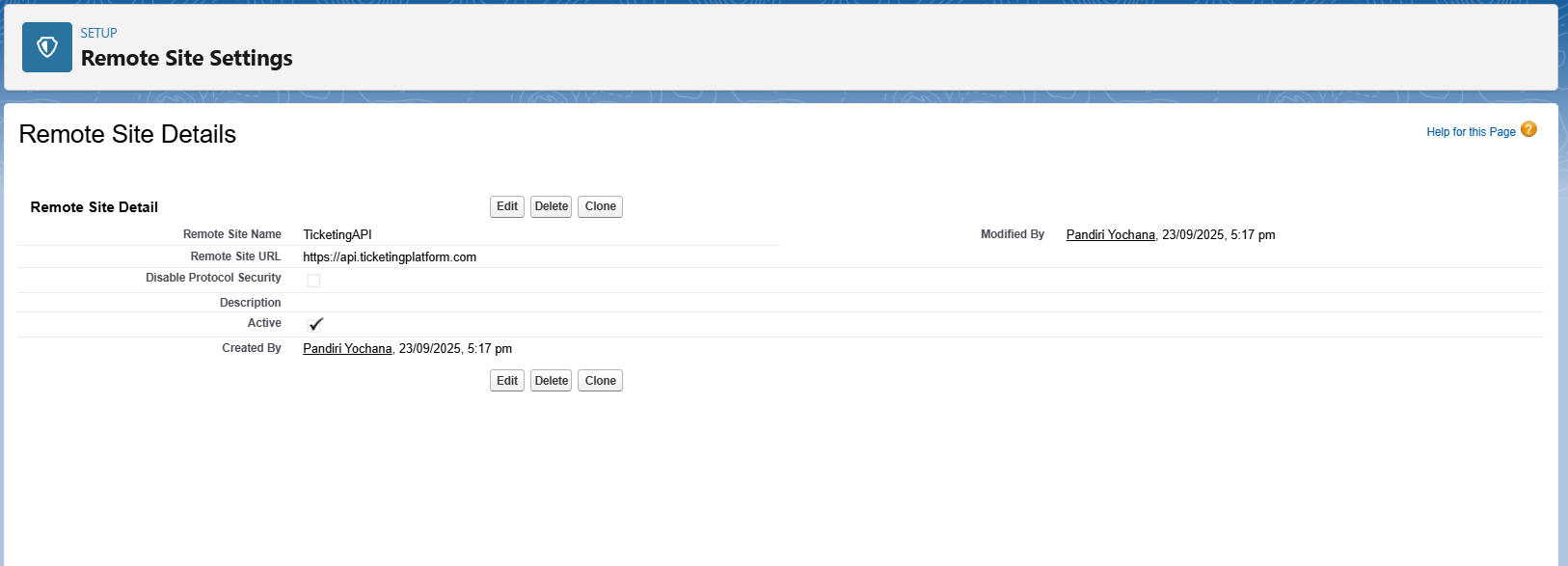


## ****9️. OAuth & Authentication****

* **Purpose:** Allow secure login and integration for external users or apps.
* **Example Use Case:**
  + Customers log in via **community portal** to view tickets, artwork, or exhibition info.
  + Integration with **artist portals** via OAuth for secure data access.
* **Salesforce Action:**
  + Configure **Connected Apps** for OAuth.
  + Use JWT or OAuth 2.0 flow for third-party apps.

## ****10. Remote Site Settings****

* **Purpose:** Authorize external domains for Apex callouts.
* **Example Use Case:**
  + Allow callouts to **ticketing APIs** or **payment gateways**.
* **Salesforce Action:**
  + Setup → Security → Remote Site Settings → Add external endpoint URLs.

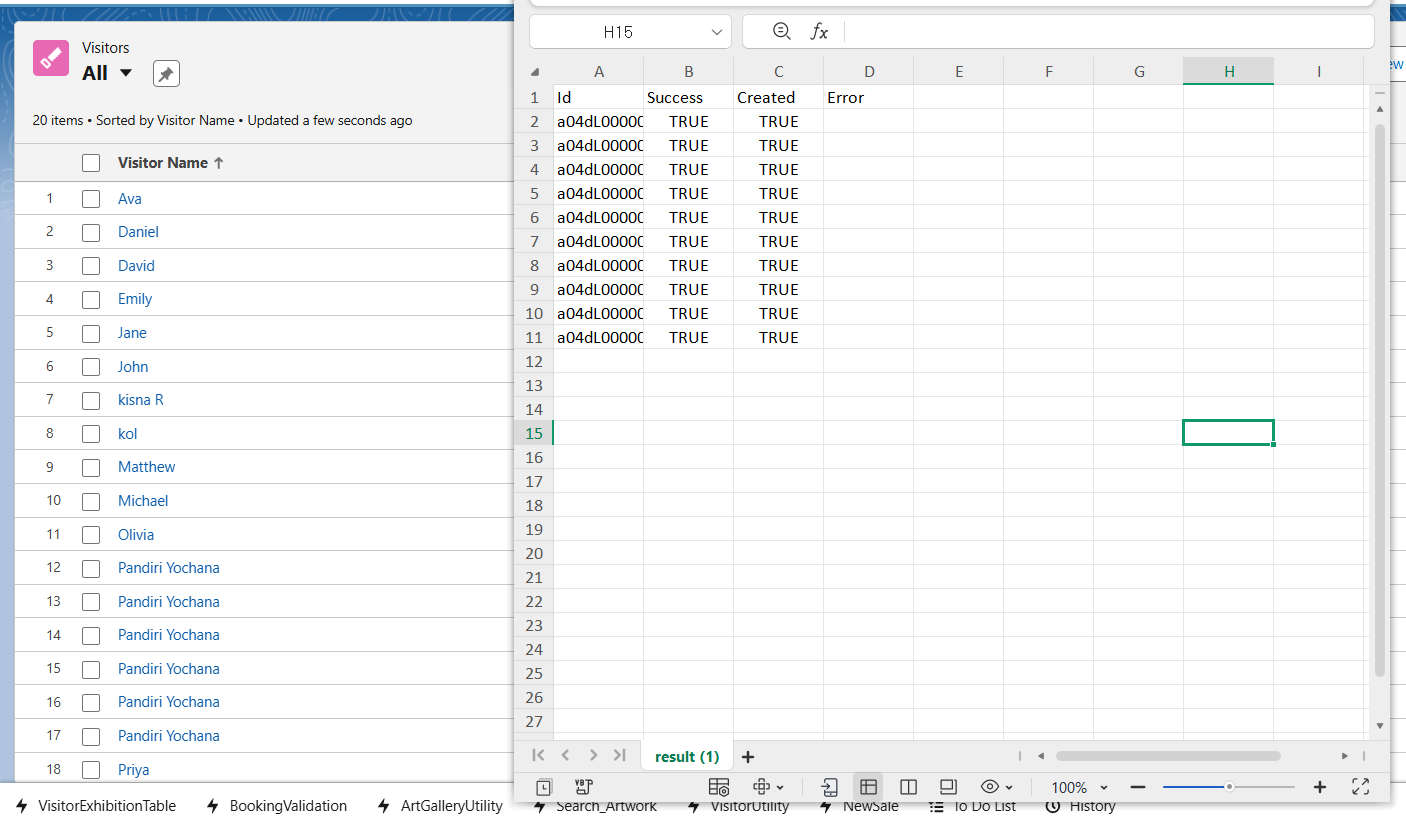


## ****Phase 8: Data Management & Deployment****

**Goal:** Efficiently manage gallery data and deploy changes between environments.

### **1️. Data Import Wizard**

* **Purpose:** Quickly import small datasets.
* **Use Case for Art Gallery:**
  + Import demo **Visitors** or **Exhibitions** (e.g., 50 demo visitors, 10 exhibitions).
* **How:**
  + Setup → Data → Data Import Wizard.
  + Select object (**Contact** for Visitors, **Exhibition\_\_c** for exhibitions).
  + Map CSV fields to Salesforce fields.
  + Start import and verify records in Salesforce.

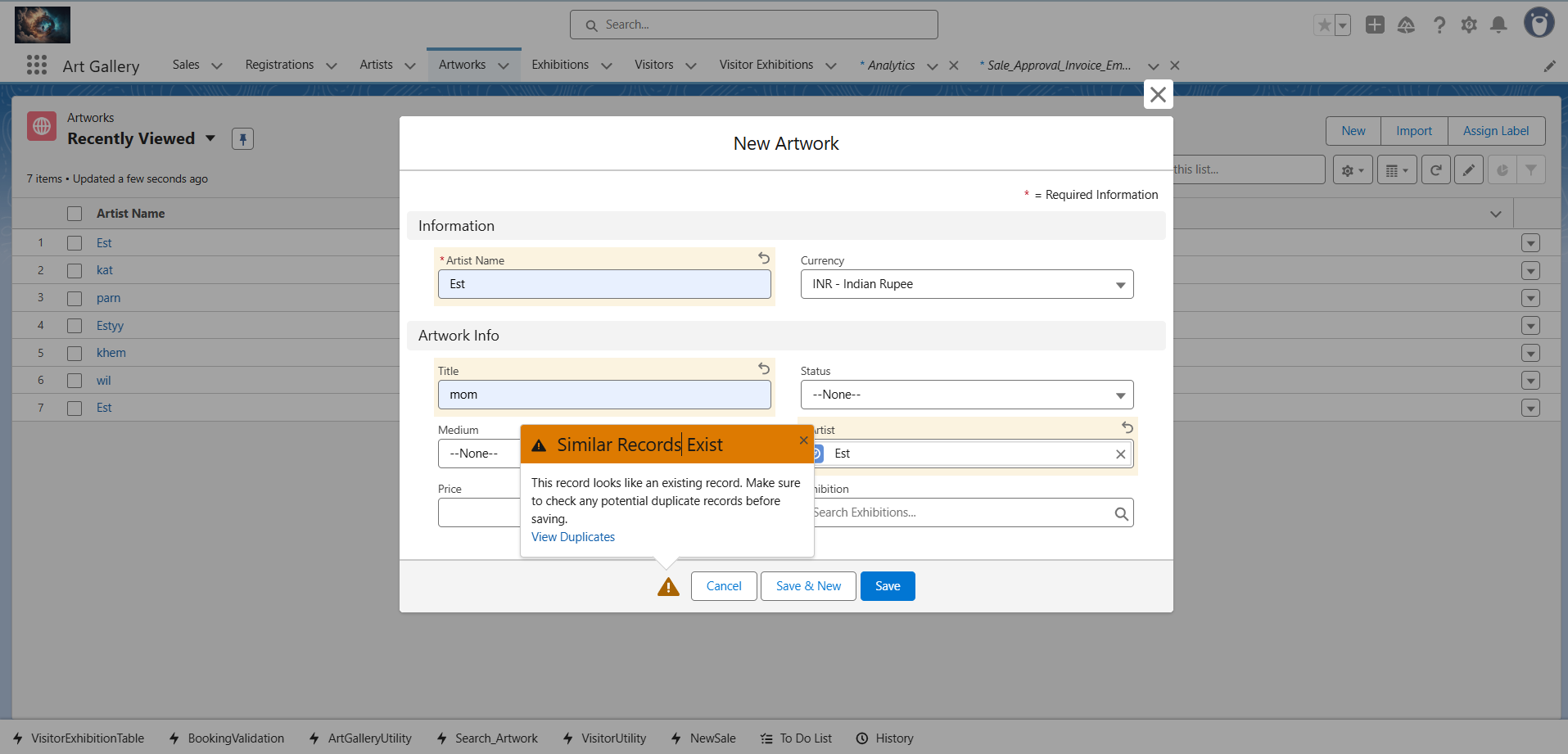


### **2️. Data Loader**

* **Purpose:** Bulk import/export for large datasets.
* **Use Case:**
  + Import hundreds of **Visitor\_Exhibition\_\_c registrations** or **Artwork\_\_c records**.
* **How:**
  + Install Salesforce Data Loader.
  + Connect using credentials or OAuth.
  + Select **Insert/Update/Upsert** operation.
  + Map CSV columns to Salesforce fields.
  + Run and verify imported data.

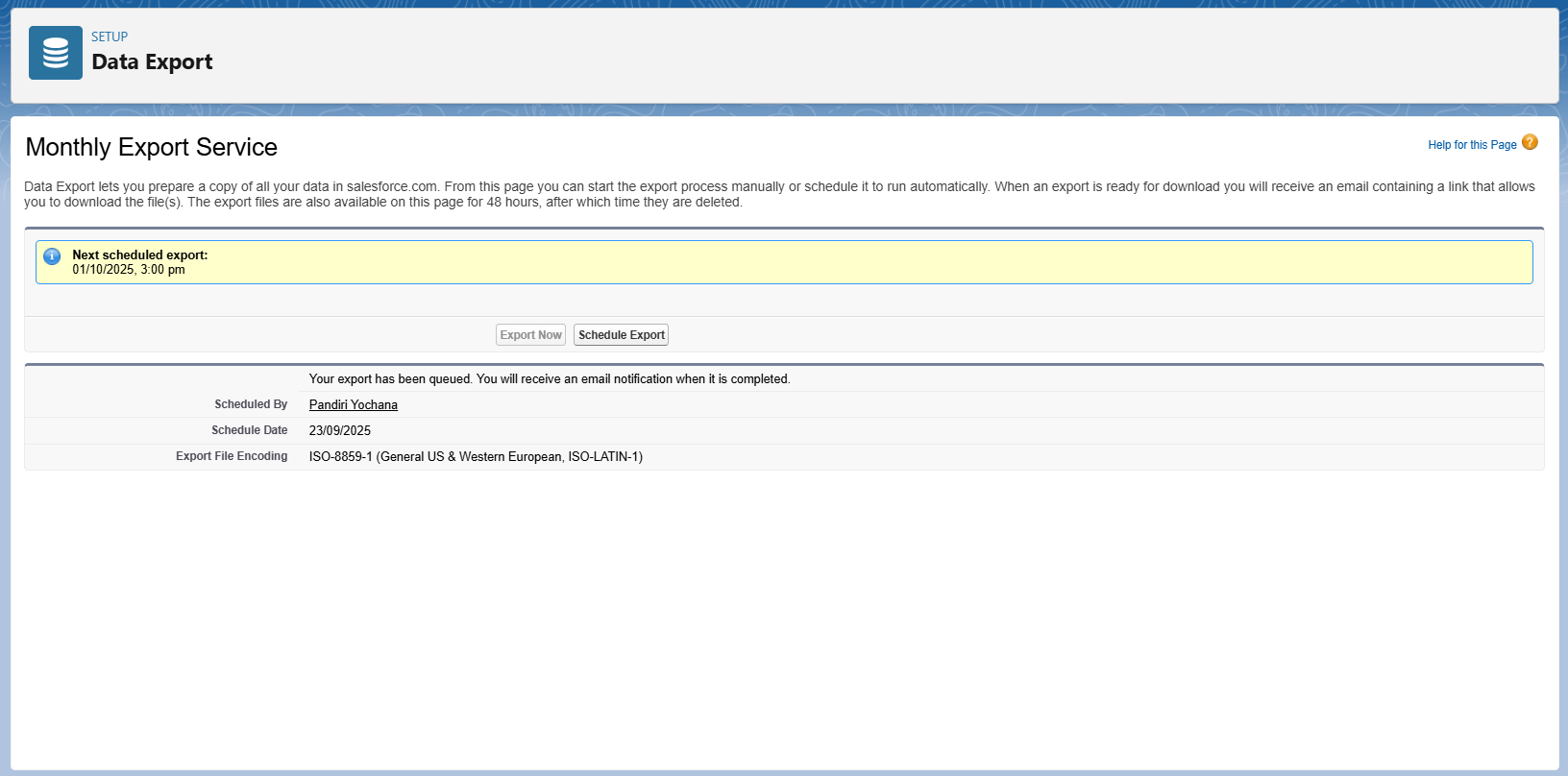
### **3️. Duplicate Rules**

* **Purpose:** Prevent duplicate data entry.
* **Use Case:**
  + Prevent duplicate **Visitors** (Contact) or **Artworks**.
* **How:**
  + Setup → Duplicate Management → Duplicate Rules.
  + Create a new rule for **Contact** and **Artwork\_\_c**.
  + Define **matching criteria** (like Name + Email for Visitors, Name + Artist for Artwork).



### **4️. Data Export & Backup**

* **Purpose:** Maintain backups for safety.
* **Use Case:**
  + Export all **Visitor, Exhibition, Artwork, Sales, and Registrations** weekly.
* **How:**
  + Setup → Data → Data Export.
  + Schedule weekly export.
  + Download .zip files for local backup or cloud storage.



### **5️. Deployment Strategies (Theoretical)**

Since Developer Edition doesn’t have Sandbox → Production:

**Options in real orgs:**

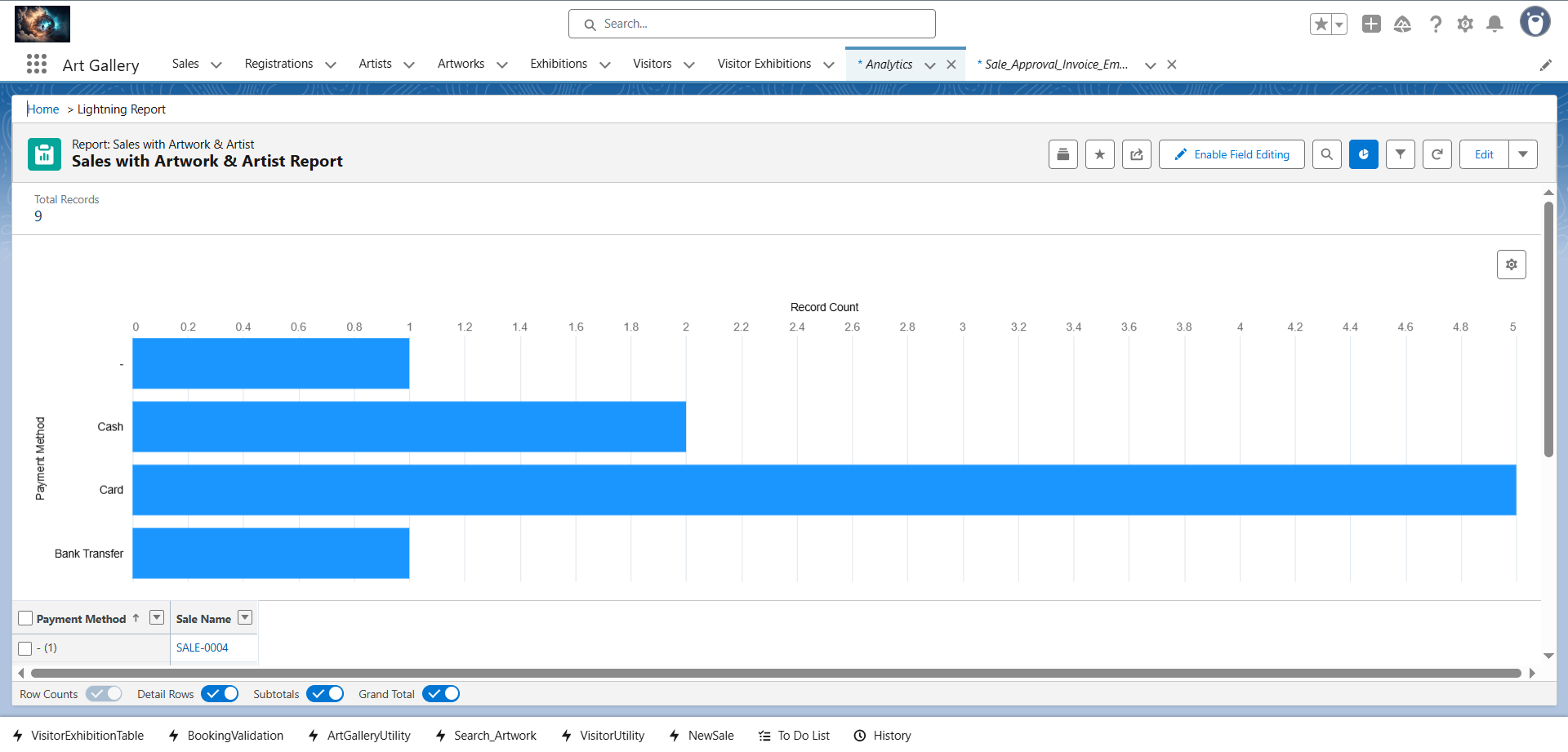
1. **Change Sets**
   * Move configurations, Apex, LWC, Flows from Sandbox → Production.
2. **Unmanaged / Managed Packages**
   * Create package → include components → upload/install.
   * Managed packages are for AppExchange.
3. **ANT Migration Tool / Salesforce CLI**
   * Command-line deployments using package.xml.
   * Automates Apex Classes, LWC, Flows deployment.

## ****Phase 9: Reporting, Dashboards & Security Review****

👉 **Goal:** Monitor gallery operations & ensure data security.

### **1. Reports**

○ **Visitor Registrations per Exhibition:** Shows how many visitors have registered for each exhibition. Useful for understanding exhibition popularity and planning visitor capacity.  
○ **Artwork Sales by Artist or Price Range:** Helps track which artists or artworks are generating the most revenue. Useful for inventory and pricing decisions.  
○ **Revenue by Exhibition:** Summarizes total sales per exhibition, helping management monitor financial performance.



### **2. Report Types**

○ **Custom report: Visitor + Visitor\_Exhibition\_\_c:** Combines visitor details with their exhibition registrations. Allows filtering by exhibition, registration date, or visitor type (guest, regular, premium).  
○ **Custom report: Artwork\_\_c + Sale\_\_c + Buyer:** Combines artworks with sales and buyer information. Useful for tracking which buyers purchased which artworks and total revenue per sale.



### **3. Dashboards**

○ **Visitor Registration Trends Dashboard:** Visualizes visitor registration data over time using charts (line, bar, or donut). Helps identify busy periods or trends in visitor engagement.  
○ **Revenue by Exhibition Dashboard:** Displays total sales revenue per exhibition using charts or tables. Useful for comparing exhibitions’ financial success.

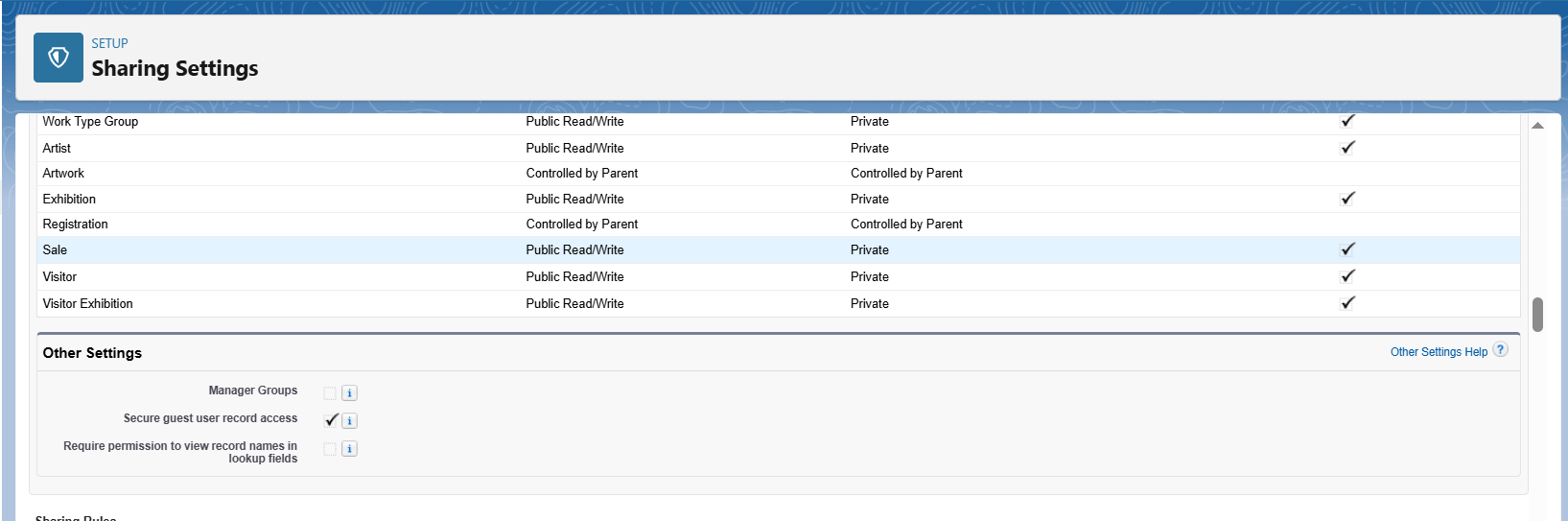
### **4.Dynamic Dashboards**

○ **Each Curator sees only their exhibitions:** Restrict dashboard visibility so each curator can monitor visitor registrations and artworks for their own exhibitions.  
○ **Each Sales agent sees only their handled artworks:** Allows agents to track their own sales and revenue performance without accessing others’ data.



### **5. Sharing Settings**

○ **Visitor\_Exhibition\_\_c records private, Exhibition\_\_c public:** Ensures sensitive registration data is restricted to authorized users while exhibitions remain visible to all relevant staff.  
○ **Artwork\_\_c public to Sales, restricted for Visitors:** Sales team can access artwork details for sales purposes; general visitors cannot view sensitive pricing or inventory details.



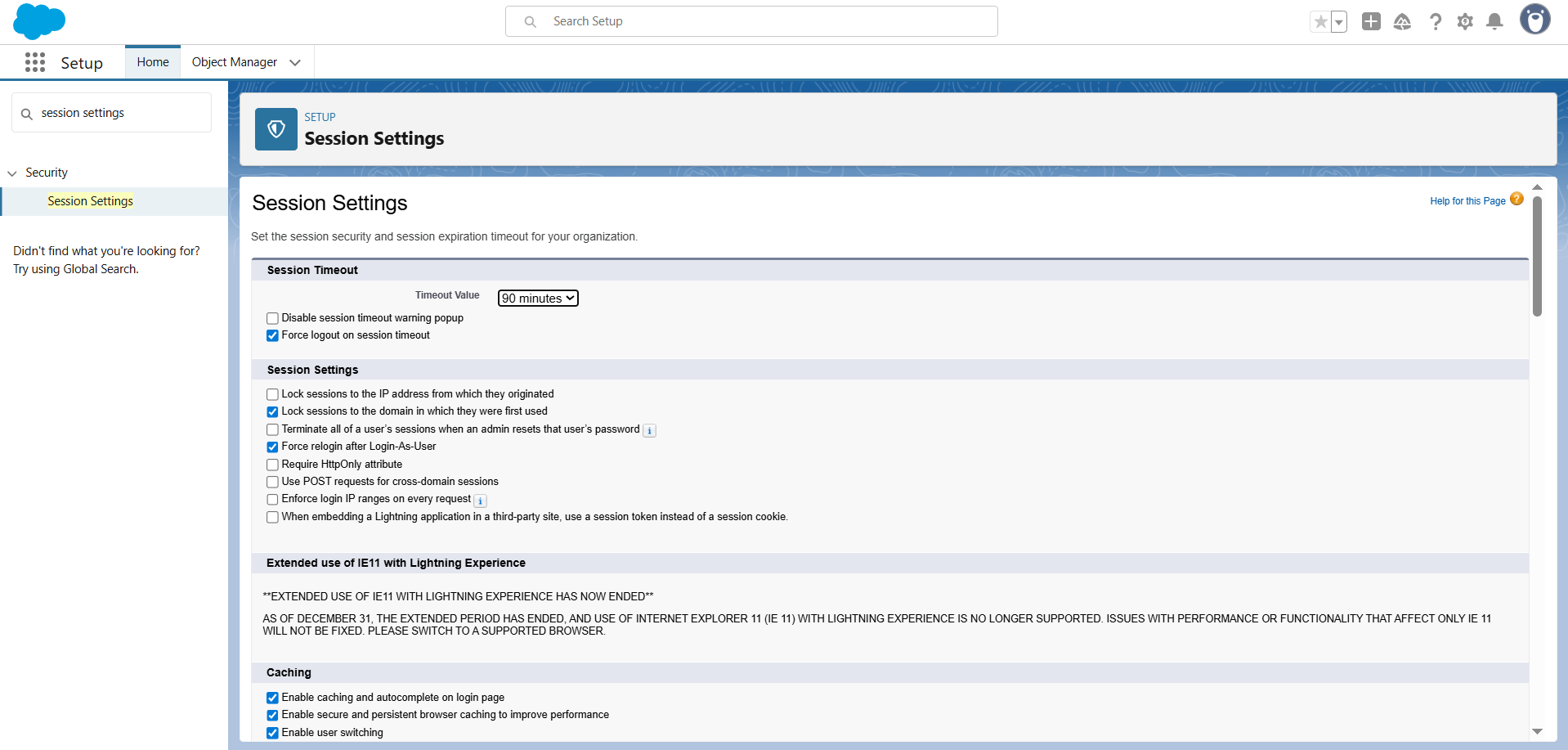
### **6. Field Level Security**

○ **Hide sensitive fields like visitor email/contact from general users:** Protects personal information while still allowing internal staff to perform their roles.



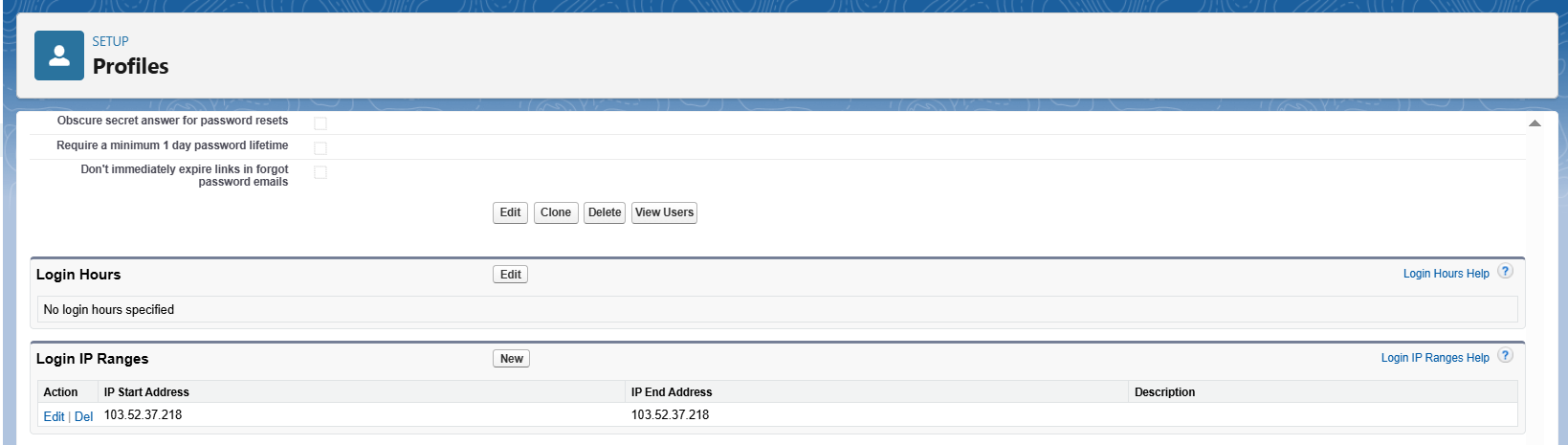
### **7. Session Settings**

○ **Timeout after 90 mins of inactivity:** Helps secure the system by automatically logging out inactive users, preventing unauthorized access.



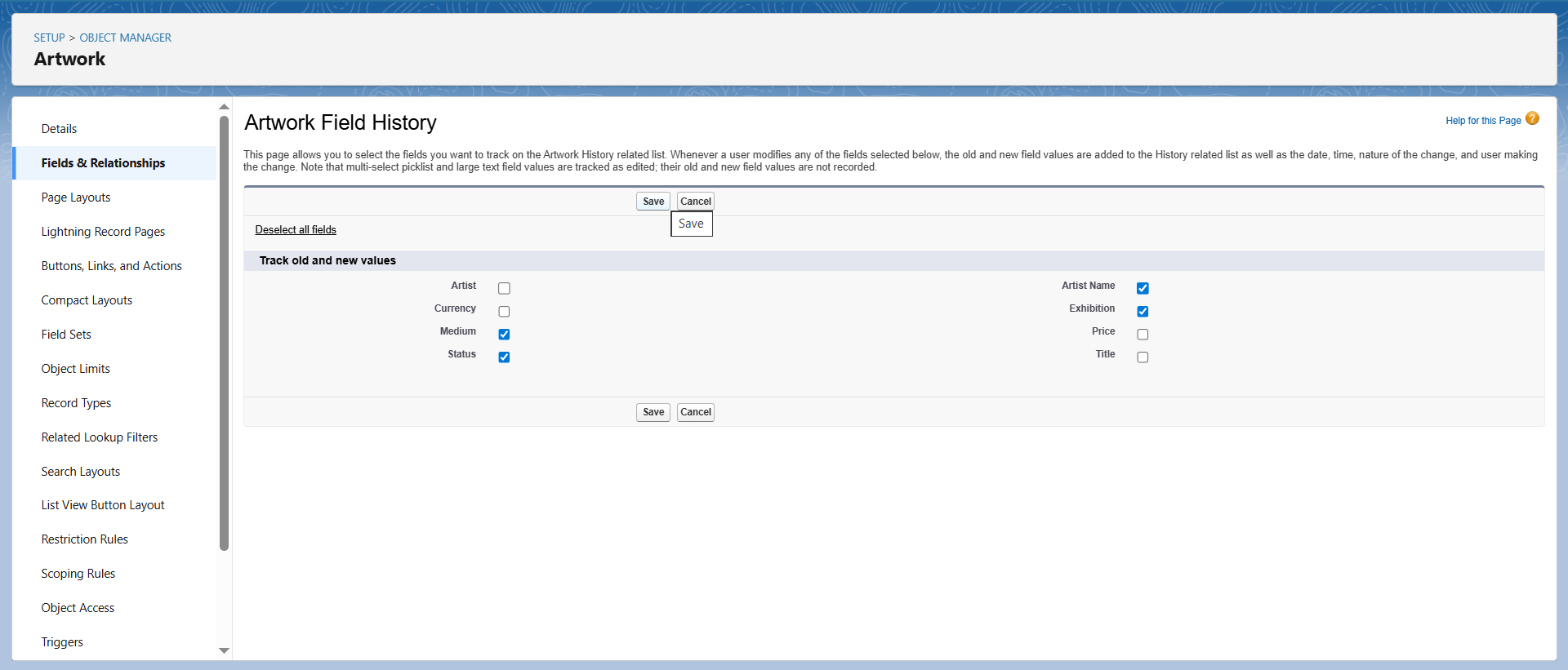
### **8. Login IP Ranges**

○ **Restrict Curators and Sales users to office IPs:** Enhances security by limiting access to known and trusted networks, reducing the risk of external attacks.



### **9. Audit Trail**

○ **Track who updated Visitor, Exhibition, Artwork, Sale, and Registration records:** Maintain accountability and transparency for all critical operations. Helps detect unauthorized changes or mistakes.



### **Phase 10: Final Presentation & Demo Day**

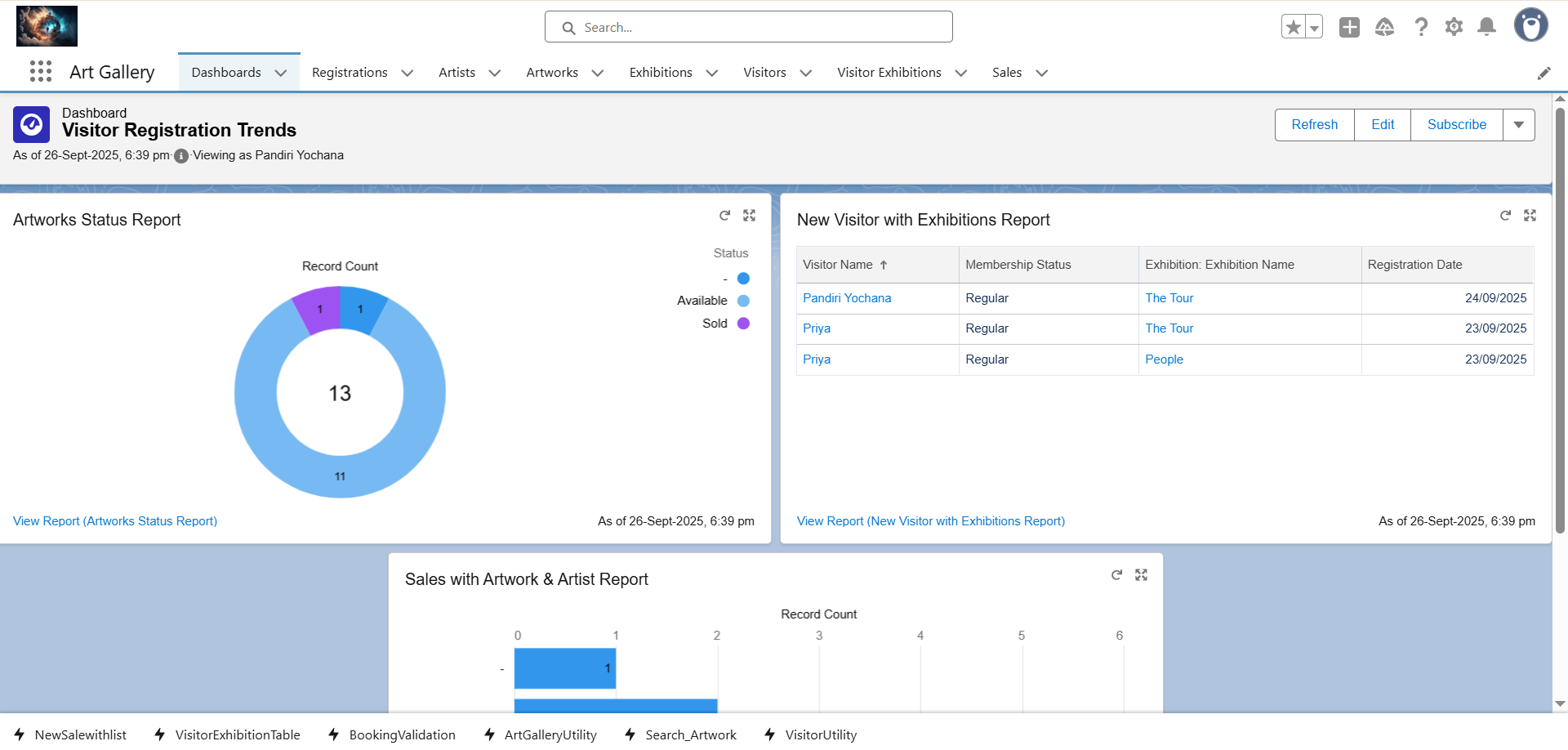
**👉 Goal:** Wrap up the project professionally, demonstrating functionality, benefits, and usability of your Art Gallery CRM system.

### **1. Pitch Presentation**

* **Problem Statement:**
  + Manual tracking of visitors, exhibitions, and artwork sales is inefficient.
  + Difficult to analyze visitor engagement or revenue per exhibition.
  + Risk of duplicate registrations or missing sales data.
* **Solution Overview:**
  + Salesforce-based Art Gallery CRM automates visitor registrations, exhibition management, artwork sales, and reporting.
  + Real-time dashboards for managers and curators.
  + Integration with external systems like Ticketing API for booking validation.
* **Benefits:**
  + Streamlined visitor registration and exhibition tracking.
  + Better decision-making with reports and dashboards.
  + Secure handling of sensitive data (visitor info, artwork pricing).
  + Improved collaboration among curators, sales agents, and management.

### **2. Demo Walkthrough**

* **Step 1: Visitor & Exhibition Management**
  + Show creating new Visitors and Exhibitions.
  + Demonstrate registration of visitors for multiple exhibitions via Visitor\_Exhibition\_\_c.
  + Highlight auto-generated booking IDs and external API validation (if implemented).
* **Step 2: Artwork Sales**
  + Demonstrate selecting artwork, creating a Sale, and linking the Buyer.
  + Show real-time update in Artwork\_\_c status.
* **Step 3: Notifications & Automation**
  + Publish Platform Events like Visitor\_Registered\_\_e.
  + Trigger email notifications or other flows.
  + Show any callouts to external Ticketing APIs or validation services.
* **Step 4: Reporting & Dashboards**
  + Open **Visitor Registration Trends Dashboard**.
  + Demonstrate filtering and dynamic dashboard visibility (curator sees only their exhibitions).
  + Highlight audit trail for registrations and sales.



* **Step 5: Security & Access**
  + Show field-level security (hide sensitive visitor fields).
  + Demonstrate profile-based access to artworks, exhibitions, and registrations.
  + Mention session timeout and IP restrictions.

### **3. Handoff Documentation**

* **System Design Document:**
  + Objects, fields, relationships, Apex classes, LWC/Aura components.
  + Flow diagrams for visitor registration, artwork sales, and event handling.
* **User Guide:**
  + Step-by-step instructions for curators, sales agents, and admin users.
  + Screenshots of registration, sale creation, and dashboards.
  + Tips on handling external API validations and error scenarios.
* **Deployment Notes:**
  + Named credentials, remote site settings, and external services configuration.
  + How to import demo data via Data Import Wizard or Data Loader.

### **LinkedIn/Portfolio Project Showcase**

