

Art Gallery CRM

-Allu S S Govardhinee

Phase 1: Discovery & Analysis Document

Problem Understanding & Industry Analysis

Objective:

Modern art galleries face challenges in managing exhibitions, tracking artwork sales, and engaging visitors meaningfully. Fragmented systems and manual processes hinder scalability, transparency, and personalized experiences.

Industry Insights:

- Galleries increasingly rely on digital tools for curation, sales, and visitor engagement.
- Provenance tracking and ethical sourcing are gaining importance.
- Hybrid exhibitions (physical + virtual) demand flexible CRM capabilities.
- Visitor data is underutilized for personalization and retention.

1. Requirement Gathering

Functional Requirements:

- Track artwork details, status, and history
- Manage exhibitions and link artworks to events
- Log visitor interactions and preferences
- Record and report artwork sales
- Automate follow-ups and feedback collection

Non-Functional Requirements:

- Scalable and secure data architecture
- Mobile-friendly visitor check-in and engagement
- Integration with payment gateways and external art databases
- Role-based access for Visitors, sales staff, and admins

2. Stakeholder Analysis

Stakeholder	Role	Needs & Expectations
--------------------	-------------	---------------------------------

Gallery Visitor	Manages exhibitions	Easy scheduling, artwork curation tools
Sales Executive	Handles transactions	Quick access to inventory and buyer history
Visitor	Engages with gallery	Personalized experience, seamless check-in
Gallery Owner	Oversees operations	Dashboards, reports, ROI insights
IT/Admin	Maintains system	Secure, customizable, low-maintenance setup

3. Business Process Mapping

Current State (Manual):

- Excel sheets for inventory
- Email-based visitor communication
- Paper-based feedback forms
- Limited visibility into sales trends

Future State (CRM-Driven):

- Centralized artwork and exhibition database
- Automated visitor engagement workflows
- Real-time sales tracking and reporting
- Feedback loop for Visitorial decisions

4. Industry-Specific Use Case Analysis

Use Case	Description	Salesforce Feature
Exhibition Planning	Schedule and curate events	Campaigns, Custom Objects
Artwork Provenance Tracking	Log history and ownership	Custom Metadata, Files
Visitor Interest Segmentation	Group visitors by preferences	Reports, Tags, Flow
Artwork Sales Lifecycle	Quote to cash process	Opportunities, CPQ
Feedback Collection	Post-visit surveys and insights	Forms, Email Studio

5. AppExchange Exploration

Relevant Apps to Consider:

- *ArtCloud Connector* – For syncing artwork listings
- *Sales Analytics Dashboards* – Prebuilt visualizations
- *Einstein Bots* – For visitor FAQs and engagement
- *Event Management Solutions* – RSVP and scheduling tools
- *Field Audit Trail* – For provenance and compliance tracking

Phase 2: Org Setup & Configuration

Objective

To configure the Salesforce environment for the Art Gallery CRM project, ensuring secure access, operational alignment, and scalable deployment. This phase lays the foundation for data modeling, automation, and user experience design.

1. Salesforce Editions

Use Case:

The project was initiated using the **Developer Edition** for prototyping and testing. For production deployment, **Enterprise Edition** is recommended due to its support for:

- Profiles and Roles
- API integrations
- Advanced sharing rules
- Multiple sandboxes

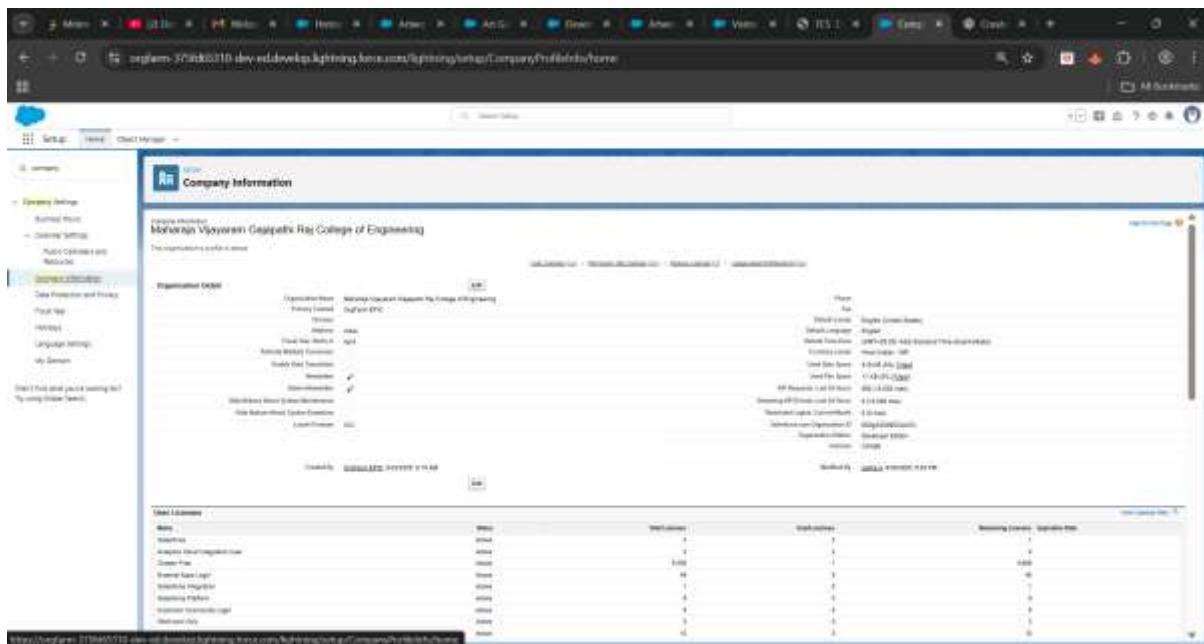
2. Company Profile Setup

Use Case:

The Company Profile reflects the identity of the gallery and ensures consistency across reports, emails, and metadata.

Configured Values:

- Company Name: *Art Gallery CRM Pvt. Ltd.*
- Address: *Vizianagaram Mandal, AP, India*
- Time Zone: *Asia/Kolkata*
- Default Currency: *INR*



3. Business Hours & Holidays

Use Case:

Business Hours and Holidays are used in approval processes, escalation rules, and SLA calculations.

Configured Values:

- Business Hours: *Mon–Sat, 10:00 AM – 6:00 PM*
 - Holidays:
 - Republic Day (Jan 26)
 - Independence Day (Aug 15)
 - Diwali
 - Christmas

Business Hours

Organization Business Hours

Select time zones and review how your support team is available. These hours, when associated with escalation rules, determine the status of which cases can be assigned.

If you enter business hours for a day, that means your organization does not operate on that day.

Business Hours Detail

Business Hours Name	working	Time Zone	Default Standard Hours
Business Hours	Monday: 09:00 AM to 0:30 PM Tuesday: 09:00 AM to 0:30 PM Wednesday: 09:00 AM to 0:30 PM Thursday: 09:00 AM to 0:30 PM Friday: 09:00 AM to 0:30 PM Saturday: 09:00 AM to 0:30 PM	(GMT+05:30) India Standard Time (Asia/Kolkata)	

Holidays

No records to display

+ Show 10 Day Always show me ▾ more records per related list

Holiday

Holiday Detail

Select date and times to define when business hours are suspended. These hours don't work, when associated with business rules, and suspend any scheduled tasks, assessments and business hours.

Start Date: 2025-01-26

Holiday Detail

Holiday Name	Repeating Day	Date and Time	Descending Holiday	Last Modified By	Last Modified On
Republic Day	One-time	10:00:00 AM Day	Occurs every January 26 effective 1/26/2023.	SystemUser [Id] 9292023, 10:00 PM	1/26/2023, 10:00 PM

Business Hours

No records to display

+ Show 10 Day Always show me ▾ more records per related list

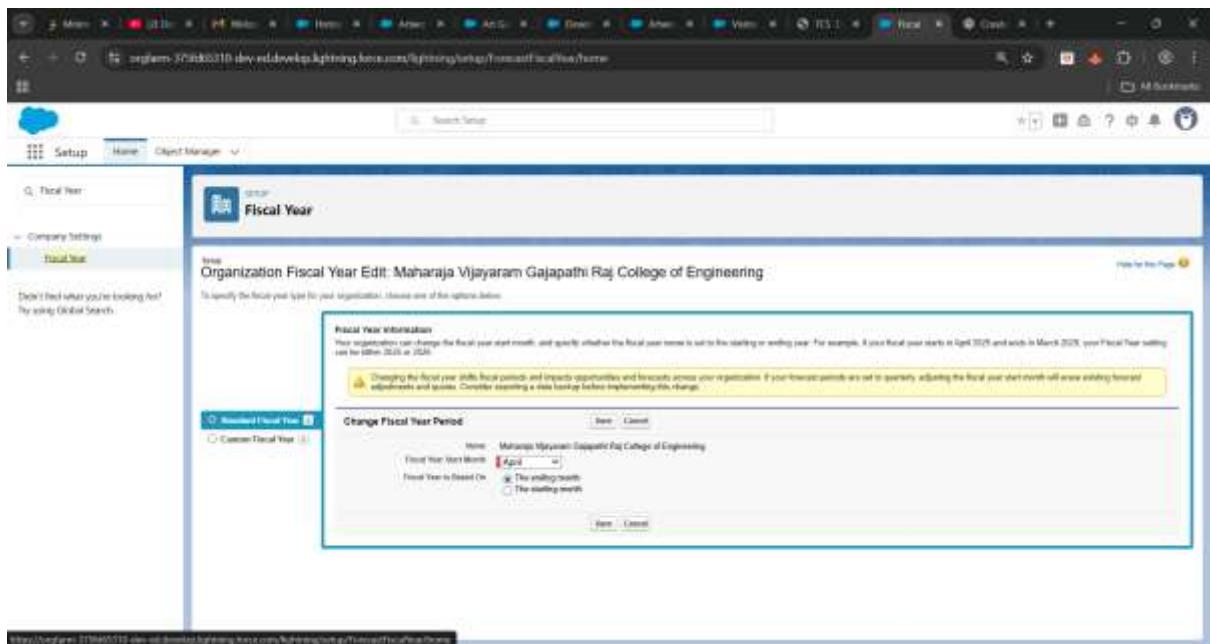
4. Fiscal Year Settings

Use Case:

Aligns financial reporting with Indian standards.

Configured Value:

- Fiscal Year: *Standard (April–March)*



5. Profiles

Use Case:

Profiles control object-level and field-level access.

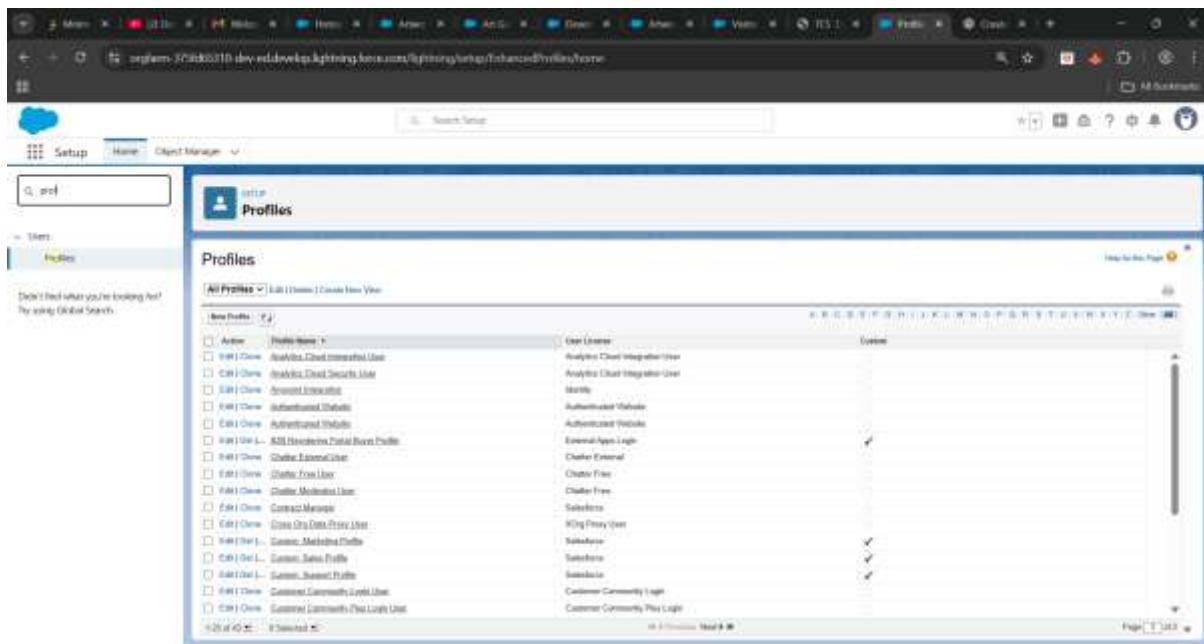
Profile Name Access Scope

Visitor Profile View/Edit Artworks, Submit Feedback, Approve Sales

Sales Profile View Artworks, Manage Transactions

Admin Profile Full access

Visitor Profile Read-only access



6. Roles

Use Case:

Roles define hierarchy and record visibility.

Role Name	Reports To
Gallery Admin	— (Top-level)
Visitor	Gallery Admin
Sales Executive	Gallery Admin
Exhibition Coordinator	Visitor

7. Organization-Wide Defaults (OWD)

Use Case:

OWD settings define baseline record access.

Object	Default Access
Artwork__c	Private
Visitor__c	Controlled by Parent
Sales_Transaction__c	Private
Exhibition__c	Public Read Only

8. Sharing Rules

Use Case:

Sharing Rules extend access based on criteria or ownership.

Examples:

- Share Artwork__c with Visitors where Gallery_Location__c = User.Location
- Share Visitor__c with Visitor if Rating__c <= 2
- Share Sales_Transaction__c with Sales Executive if Status__c = Pending

Sharing Rule	Access Level	Access Level
Shipping Carrier	Public Read Only	Private
Shipping Carrier Method	Public Read Only	Private
Shipping Configuration Set	Public Read Only	Private
Shipping Channel	Public Read/Write	Private
Tierless Host Mapping	Public Read Only	Private
User Presence	Public Read Only	Private
User Provisioning Request	Private	Private
Watch	Private	Private
Web-Card Document	Private	Private
Work Item	Private	Private
Work Plan	Private	Private
Work Plan Template	Private	Private
Work Step Template	Private	Private
Work Type	Private	Private
Work Type Item	Public Read/Write	Private
Activity	Public Read/Write	Private
Callation	Public Read/Write	Private
Task Transaction	Public Read/Write	Private
Video	Private	Private
VideoCallibration	Public Read/Write	Private

9. Login Access Policies

Use Case:

Controls session behavior and access security.

Configured Values:

- Session Timeout: *30minutes*
- IP Restrictions: *Enabled for Admin profile*
- Multi-Factor Authentication: *Required for all users with edit access*

Session Settings

Set the session security and session expiration timeout for your organization.

Session Timeout:

Session timeout: 30 minutes

Create session timeout warning popups

Force login on session timeout

Session Settings:

Lock sessions to the IP address from which they originated

Lock sessions to the domain to which they were first used

Terminate all of a user's sessions when an admin resets their user's password

Force single sign-on (SSO)

Require https only attribute

Use POST requests for cross-domain sessions

Invalidate login IP ranges on every request

When embedding a Lightning application in a third-party site, use a session token instead of a session cookie

Extended use of IE11 with Lightning Experience:

"EXTERRED USE OF IE11 WITH LIGHTNING EXPERIENCE HAS NOW ENDED"

AS OF DECEMBER 11, THE EXTENDED PERIOD UNDERTAKEN AND USE OF INTERNET EXPLORER 11 (IE 11) WITH LIGHTNING EXPERIENCE IS NO LONGER SUPPORTED. ISSUES WITH PERFORMANCE OR FUNCTIONALITY THAT AFFECT ONLY IE 11 WILL NOT BE FIXED. PLEASE SWITCH TO A SUPPORTED BROWSER.

Caching:

All users can cache and reuse static assets on their page.

Deliverables

- Fully configured Org with security and access controls
- Profiles, Roles, and Permission Sets aligned to gallery operations
- Deployment tools configured for future phases

Phase 3: Data Modeling & Relationships

Objective

To design a scalable, minimal, and Visitor-friendly data model that reflects the operational structure of an art gallery. This phase focuses on creating custom objects, defining relationships, and configuring layouts to support automation, reporting, and user experience.

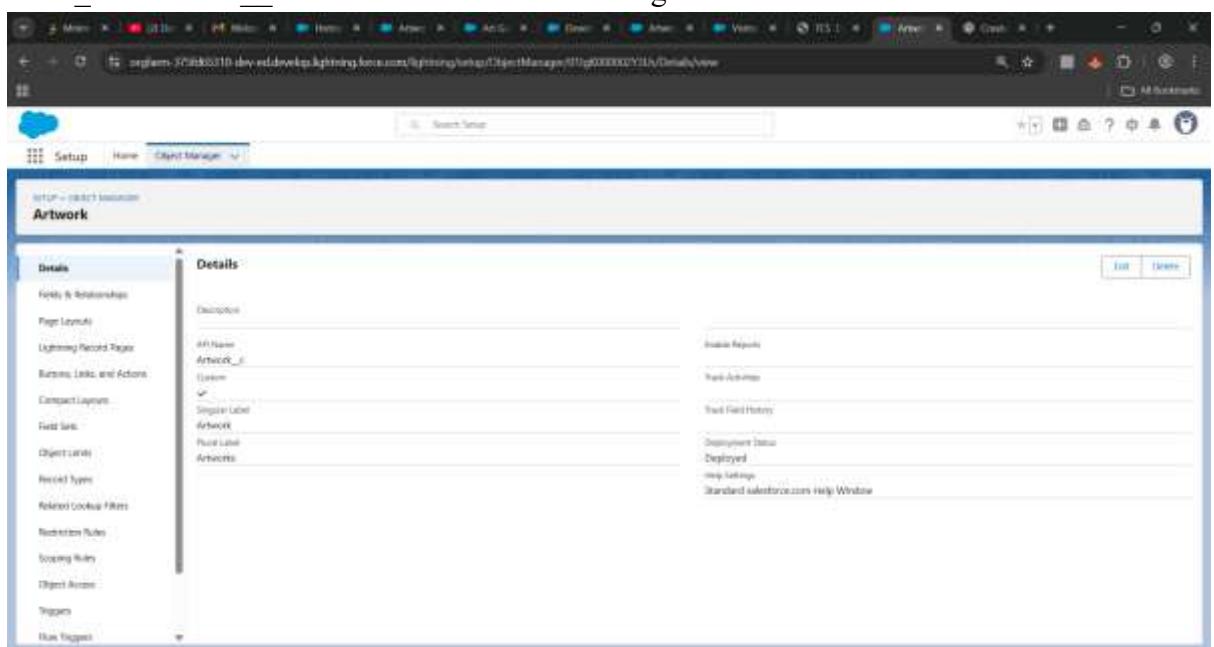
1. Standard & Custom Objects

Use Case:

Standard objects like User, Contact, and Task are used for gallery staff and activity tracking. Custom objects were created to reflect domain-specific entities.

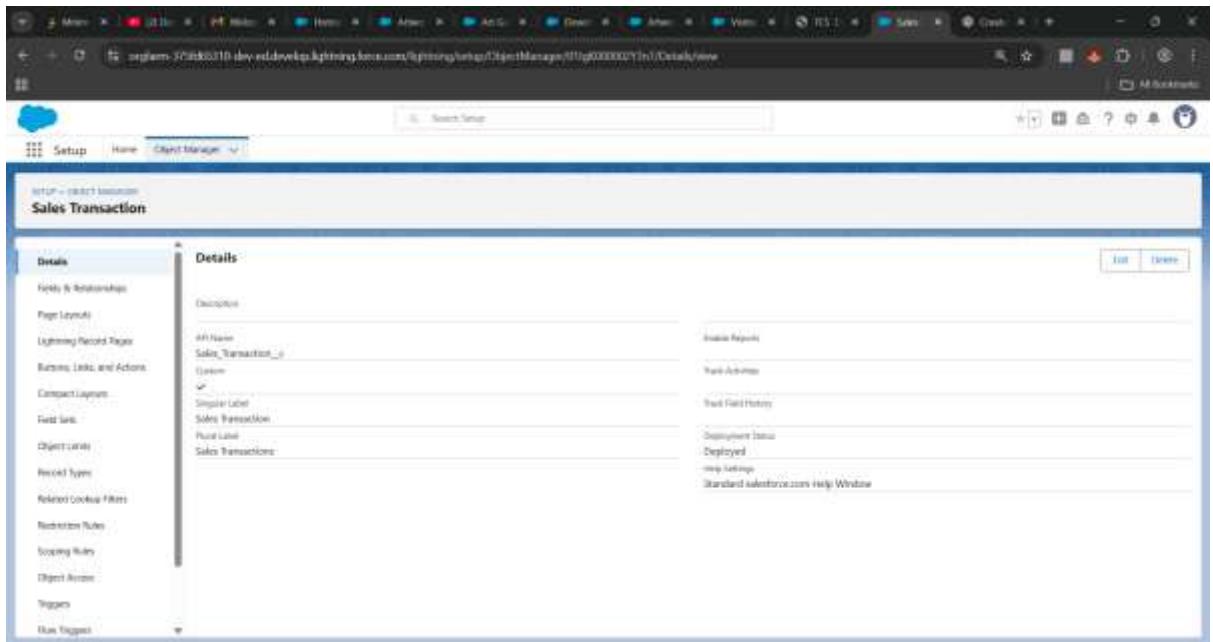
Custom Objects Created:

- Artwork__c → Represents individual artworks
- Visitor__c → Captures feedback from gallery visitors
- Exhibition__c → Tracks gallery events and exhibitions
- Sales_Transaction__c → Manages artwork sales



The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes tabs for Home, Object Manager, and a dropdown menu. The main content area displays the 'Visitor' object details. On the left is a sidebar with various configuration tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Labels, Record Types, Related Cross-Object Filters, Restriction Rules, Scoring Rules, Object Access, Triggers, and Run Reports. The 'Details' tab is selected. The main panel shows the 'Description' field with the value 'Visitor'. To the right of the description are several sections: Recent Reports, Quick Actions, Trail Field History, Deployment Status, Deployed, Map Settings, and Standard Salesforce.com Help Window. At the bottom right of the main panel are 'Edit' and 'Delete' buttons.

This screenshot shows the same Salesforce Object Manager interface as the previous one, but the object being viewed is 'Exhibition'. The sidebar and main panel structure are identical, with the 'Details' tab selected. The 'Description' field contains the value 'Exhibition'. The right-hand side panels for Reports, Actions, Field History, Deployment Status, and Help Window are also present. The 'Edit' and 'Delete' buttons are located at the bottom right of the main panel.



2. Fields

Use Case:

Each object includes curated fields to support reporting, automation, and Visitor workflows.

Examples:

- Artwork__c:
 - Medium__c (Picklist)
 - Price__c (Currency)
 - Flagged_for_Review__c (Checkbox)
- Visitor__c:
 - Rating__c (Number)
 - Comments__c (Long Text)
 - Artwork__c (Lookup)
 - Email__c (Email)

Fields & Relationships					
	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	
Created By	CreatedById	Lookup(Users)			
Customer Name	CustomerName	Text(20)			
Event Date	EventDate	Date			
Debt Holder	Name	Text(20)			
Last Modified By	LastModifiedById	Lookup(Users)			
Location	Location	Text(20)			
Owner	Owner	Lookup(Users/Groups)			
Recent Type	RecentType	Record Type			
Start Date	StartDate	Date			

Fields & Relationships					
	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	
Created By	CreatedById	Lookup(Users)			
Lead	Lead	Lead			
Feedback	Feedback	Long Text Area(32768)			
Full Name	FullName	Text(20)			
Last Modified By	LastModifiedById	Lookup(Users)			
Owner	Owner	Lookup(Users/Groups)			
Record Type	RecordType	Account Type			
Visit Date	VisitDate	Date			
Visitor Name	Name	Text(20)			

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes links for Setup, Home, Object Manager, and a search bar. The main content area is titled "Sales Transaction". On the left, a sidebar lists various configuration options under "Fields & Relationships". The main panel displays a table titled "Fields & Relationships" with the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Artwork	Artwork__c	Lookup(Artwork)		
Created By	CreatedById	Lookup(Id)		
Last Modified By	LastModifiedById	Lookup(Id)		
Owner	OwnerId	Lookup(User, Group)		
Record Type	RecordTypeId	Record Type		
Sales Transaction Name	Name	Text(50)		
Visitor	Visitor__c	Lookup(Visitor)		

Below the table are buttons for "Check Field", "New", "Deleted Fields", "Field Dependencies", and "Set History Tracking".

3. Record Types

Use Case:

Record Types allow different layouts and processes for different artwork categories.

Examples:

- Artwork__c:
 - Contemporary
 - Traditional
 - Digital

The screenshot shows the Salesforce Object Manager interface for the 'Artwork' object. The left sidebar lists various configuration options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Business Lists, and Actions, Compact Layouts, Field Sets, Object Labels, and Record Types. The 'Record Types' option is selected, highlighted in blue. The main content area displays a table titled 'Record Types' with three items:

RECORD TYPE LABEL	DESCRIPTION	ACTIVE	MODIFIED BY
For Sale	Available for purchase	Active	coffee a. 9/9/2023, 12:00 AM
Used Piece		Active	coffee a. 9/9/2023, 12:00 AM
Permanent Collection	Artworks owned by the gallery	Inactive	coffee a. 9/9/2023, 12:00 AM

At the top right of the main content area, there are buttons for 'Quick Find', 'New', and 'Page Layout Assignment'. The URL in the browser bar is `https://orgname-27080310-dev-eddev.lightning.force.com/lightning/sobject/ObjectManager/001g00000V1Ls/RecordTypesView`.

- Visitor__c:
 - Exhibition Feedback
 - Artwork Feedback

Setup Home Object Manager

Visitor

Record Types

RECORD TYPE LABEL	DESCRIPTION	ACTIVE	MODIFIED BY
Artist Guest	Participating artist	✓	refine a. 4/19/2015, 11:29 PM
General Visitor	Walk-in or casual attendee	✓	refine a. 4/25/2015, 11:30 PM
VP Collector	High-value buyer	✓	refine a. 5/09/2015, 11:31 PM

Detail Fields & Relationships Page Layouts Lightning Record Pages Business Lists, and Actions Compact Layouts Field Sets Object Labels Record Types Related Object Filters Recursion Rules Scoping Rules Object Access Triggers Run Scripts

Setup Home Object Manager

Sales Transaction

Record Types

RECORD TYPE LABEL	DESCRIPTION	ACTIVE	MODIFIED BY
Commissioned Sale	Custom order type	✓	refine a. 9/30/2015, 12:04 AM
Direct Sale	In-galley purchase	✓	refine a. 9/30/2015, 12:08 AM
Online Sale	Article sold via website	✓	refine a. 9/30/2015, 12:07 AM

Detail Fields & Relationships Page Layouts Lightning Record Pages Business Lists, and Actions Compact Layouts Field Sets Object Labels Record Types Related Object Filters Recursion Rules Scoping Rules Object Access Triggers Run Scripts

The screenshot shows the Salesforce Object Manager interface for the 'Exhibition' object. On the left, a sidebar lists various configuration tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Business Lists, and Actions, Compact Layouts, Field Sets, Object Labels, Record Types (which is selected), Related List Filters, Restriction Rules, Scoping Rules, Object Access, Triggers, and Run Triggers. The main content area displays a table titled 'Record Types' with three entries:

Record Type Label	Description	Active	Modified By
Group Show	multiple exhibit	Yes	John A. Smith 11/15/14, 11:54 PM
Single Show	Single-exhibit	No	John A. Smith 11/15/14, 11:55 PM
Virtual Exhibition	Online-only	Yes	John A. Smith 11/15/14, 11:57 PM

4. Page Layouts

Use Case:

Page Layouts are tailored to Visitor roles and record types.

Examples:

- Artwork_c:
 - Contemporary Layout → includes valuation and provenance
 - Digital Layout → includes online engagement metrics

The screenshot shows the Salesforce Object Manager interface for the 'Artwork' object. On the left, a sidebar lists various configuration tabs: Details, Fields & Relationships, Page Layouts (which is selected), Lightning Record Pages, Business Lists, and Actions, Compact Layouts, Field Sets, Object Labels, Record Types, Related List Filters, Restriction Rules, Scoping Rules, Object Access, Triggers, and Run Triggers. The main content area displays the 'Page Layouts' configuration screen for the 'Artwork' object. It shows a table with columns for Name, Order, Default, and Description. There are two rows: 'Artwork Detail' and 'Artwork Summary'. Below the table, there are sections for 'Artwork Detail' and 'Artwork Summary' with various fields like Title, Description, and Created Date.

- Visitor__c:
 - Layout includes sentiment field and Visitor notes

5. Compact Layouts

Use Case:

Compact Layouts define which fields appear in highlights panel and mobile views.

Examples:

- Artwork__c Compact Layout:
 - Name, Artist__c, Status__c, Price__c

The screenshot shows the Salesforce Object Manager interface. On the left, a sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, and Compact Layouts. The Compact Layouts section is currently selected. In the main area, the 'Artwork' object is displayed with its summary information. The 'Artwork Summary' compact layout is selected, showing fields such as Label, API Name, and Included Fields. The layout was created by 'Admin_A' on 9/20/2014 at 12:08 AM and modified by 'Admin_A' on 9/20/2014 at 12:09 AM.

6. Schema Builder

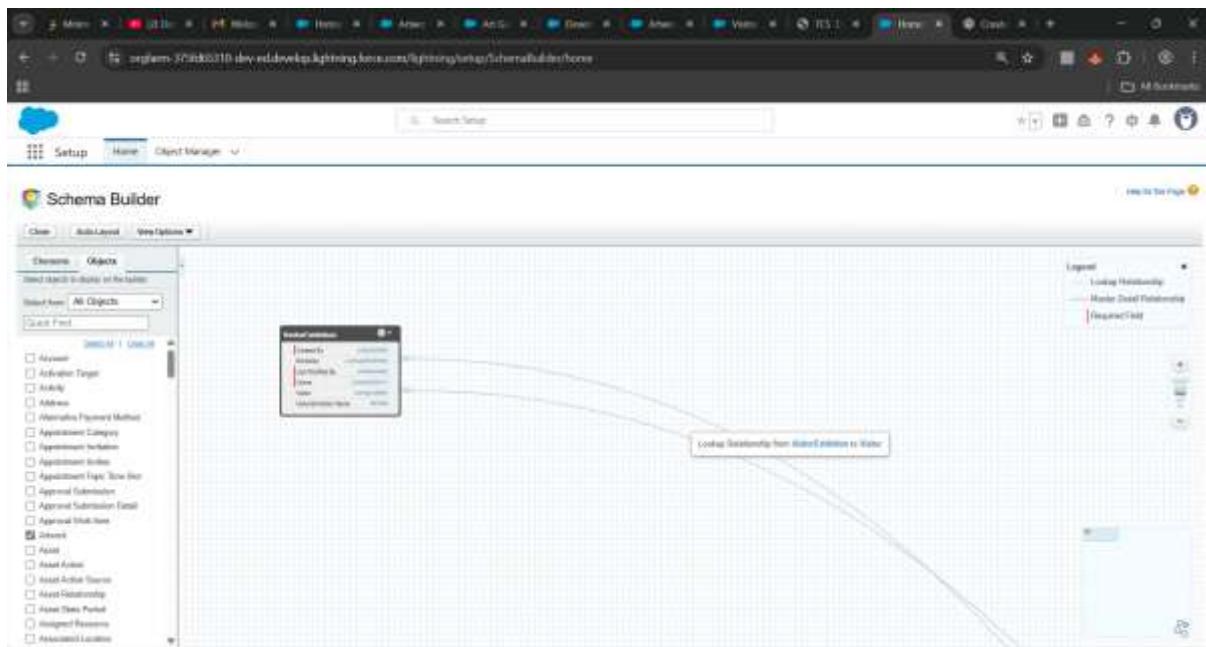
Use Case:

Schema Builder was used to visualize relationships and validate field mappings.

Highlights:

- Artwork_c → Lookup to Exhibition_c
- Visitor_c → Lookup to Artwork_c
- Sales_Transaction_c → Master-Detail to Artwork_c

The screenshot shows the Salesforce Schema Builder tool. It displays a diagram of object relationships. The 'Artwork' object is connected to the 'Sales Transaction' object via a 'Master-Detail' relationship, indicated by a blue line with a circle. The 'Artwork' object is also connected to the 'Condition' object via a 'Lookup Relationship', indicated by a blue line with a square. The 'Visitor' object is connected to the 'Artwork' object via a 'Lookup Relationship', indicated by a blue line with a square. A legend on the right side defines the symbols: a blue line with a circle for Master-Detail Relationship, a blue line with a square for Lookup Relationship, and a red line with a square for Required Field.



7. Lookup vs Master-Detail vs Hierarchical Relationships

Use Case:

Relationships were chosen based on data ownership and deletion behavior.

Relationship Type	Example	Reason
Lookup	Visitor__c → Artwork__c	Feedback exists independently
Master-Detail	Sales_Transaction__c Artwork__c	→ Sales tied directly to artwork lifecycle
Hierarchical	User → Manager	Used for Visitor reporting hierarchy

8. Junction Objects

Use Case:

Junction objects were used to model many-to-many relationships.

Example:

- Artwork_Exhibition__c → Junction between Artwork__c and Exhibition__c
 - Allows an artwork to appear in multiple exhibitions

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes links for Setup, Home, and Object Manager. The main title is "VisitorExhibition". On the left, there's a sidebar with various tabs like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Business Lists, and Actions, among others. The main content area is titled "Fields & Relationships" and displays a table with four columns: FIELD LABEL, FIELD NAME, DATA TYPE, and CONTROLLED FIELD. The table contains five rows:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLED FIELD
Contact By	Customer	Lookup(User)	
Exhibition	Exhibition__c	Lookup(Exhibition)	
Last Modified By	LastModifiedBy	Lookup(User)	
Owner	Owner	Lookup(User Group)	
Visits	Visitor__c	Lookup(Visitor)	
VisitorExhibition Notes	Notes	Text	

At the bottom of the table, there are buttons for Check Field, New, Inserted Fields, Field Dependencies, and Set History Tracking.

Deliverables

- Custom objects and fields aligned with gallery operations
- Record Types and Layouts tailored to Visitor workflows
- Relationships modeled for automation and reporting
- Schema Builder used for visual validation

Phase 4: Process Automation (Admin)

Objective

To automate Visitor workflows, approval processes, and feedback handling using declarative Salesforce tools. This phase enhances operational efficiency, reduces manual effort, and ensures timely Visitor actions across the CRM.

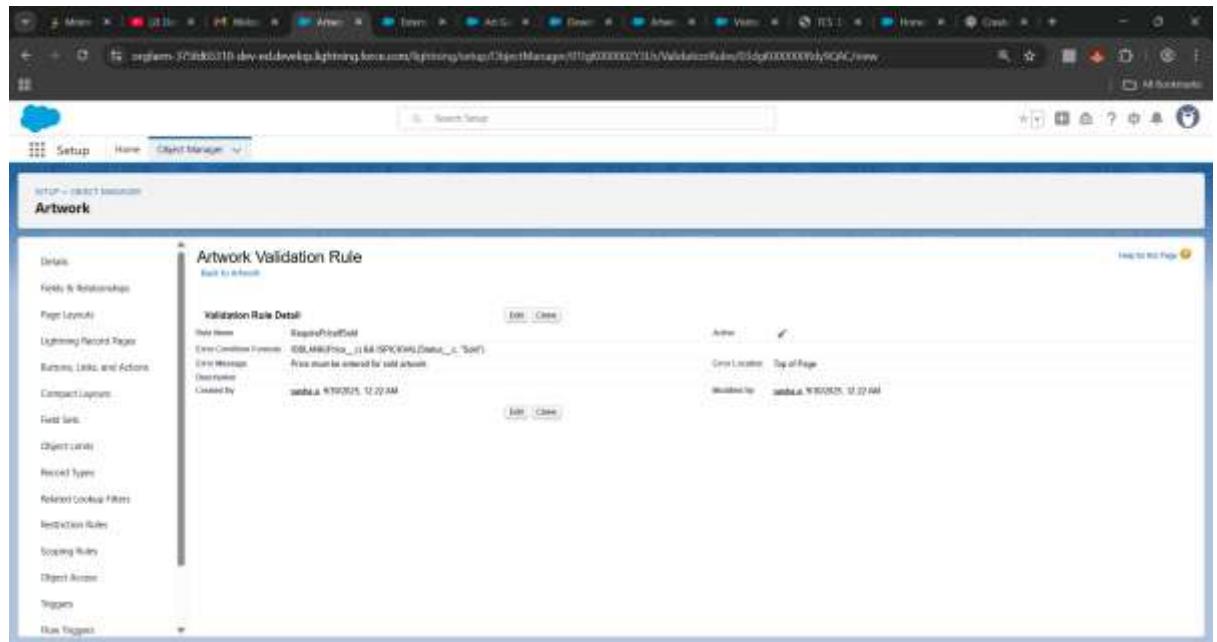
1. Validation Rules

Use Case:

Ensure data integrity during record creation and updates.

Examples:

- Artwork__c: Prevent saving if Price__c is blank or zero.



2. Workflow Rules

Use Case:

Trigger simple actions based on field criteria.

Examples:

- Artwork__c: If Status__c = Flagged, send email to Visitor Lead.

The screenshot shows the 'Workflow Rules' page in the Salesforce setup interface. A specific rule named 'NotifyOnNewExhibition' is selected. The rule details include:

- Name:** NotifyOnNewExhibition
- Active:** Yes
- Description:** Notify on new exhibition record creation.
- Created By:** [REDACTED] (KNOXBOE, 12:20 AM)
- Modified By:** [REDACTED] (KNOXBOE, 12:20 AM)
- Object:** Exhibition
- Evaluation Criteria:** Evaluates the rule when a record is created.

The 'Workflow Actions' section contains one action:

- Action:** Create-Arr
- Type:** Email Alert
- Description:** New exhibition record is created, then...

A yellow warning message at the bottom states: "You cannot add live time trigger to an active rule. Reschedule this rule."

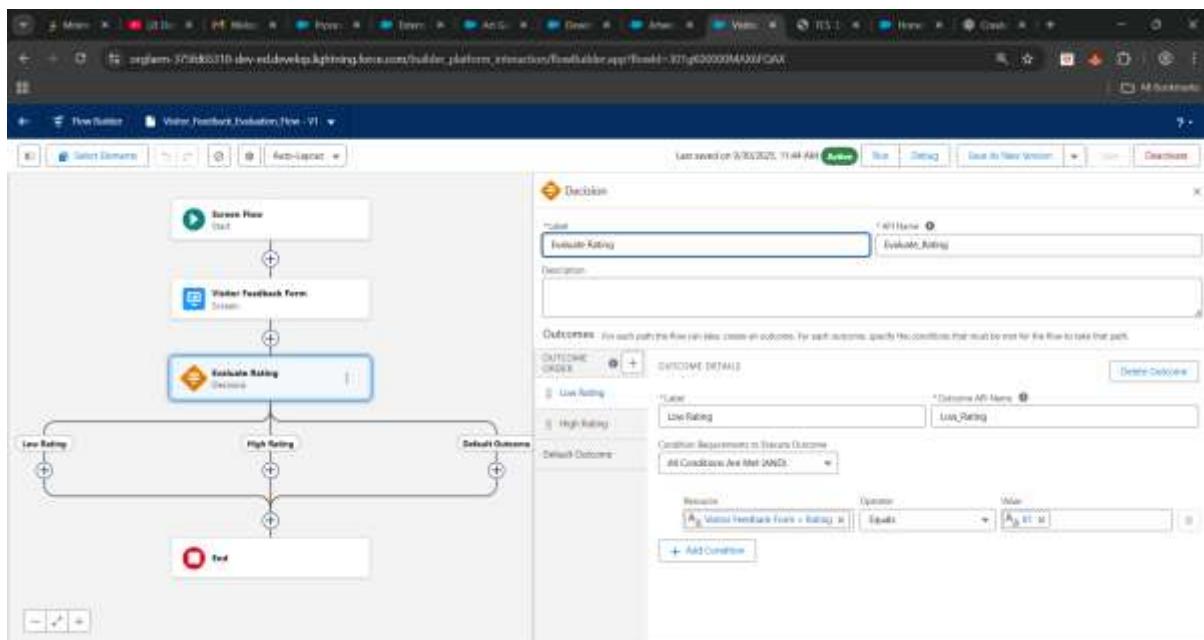
3. Process Builder

Use Case:

Automate multi-step logic with conditions and actions.

Examples:

- Visitor__c: If Rating__c <= 2, create a task for Visitor to review.



- Artwork__c: If Price__c > 100000, auto-submit for approval.

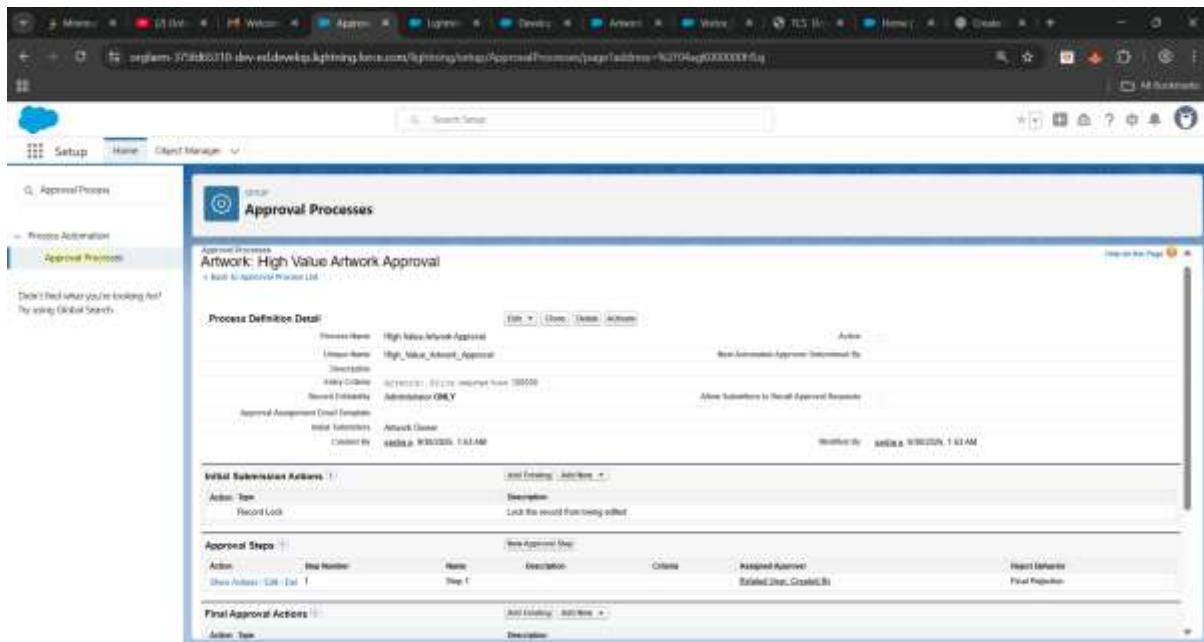
4. Approval Process

Use Case:

Route high-value artwork records for Visitor approval.

Examples:

- Artwork__c: Submit for approval if Price__c > 100000 and Status__c = Available.



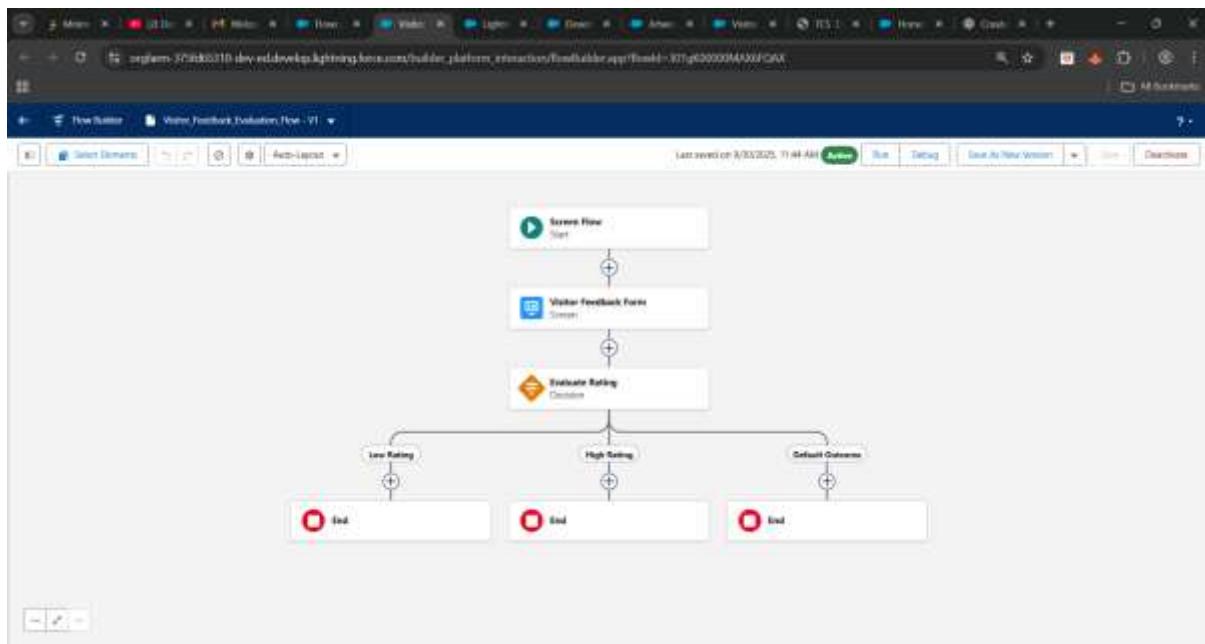
5. Flow Builder

Use Case:

Automate complex logic with user interaction and background processing.

✿ Types Implemented:

- **Screen Flow:** Visitor feedback triage wizard
- **Record-Triggered Flow:** Auto-create approval record when artwork is flagged
- **Scheduled Flow:** Daily check for artworks missing metadata
- **Auto-Launched Flow:** Called from Process Builder to send notifications



6. Email Alerts

Use Case:

Notify Visitors and admins of key events.

Examples:

The screenshot shows the "Email Alerts" page in the Salesforce Setup. A new email alert is being created with the following details:

- Name:** Trial.Alert
- Description:** new exhibition record is created, thanks
- From Email Address:** Current User's email address
- To Address:** Email Alert: new exhibition record is created, thanks - Subscribers (Selected)
- From Name:** new_exhibit__c_created
- Template:** Sales/New Customer.Draft
- Object:** Exhibit
- Created By:** SAMALA (1005258, 12:39 AM)

The page also lists "Rules Using This Email Alert" and "Approval Processes Using This Email Alert", both of which are currently empty.

Deliverables

- Validation Rules for data integrity
- Workflow Rules and Process Builder for Visitor automation

- Approval Processes for high-value artwork review
- Flows for feedback triage and Visitor tasks
- Email Alerts and Custom Notifications for real-time updates
- Field Updates and Tasks for operational efficiency

Phase 5: Apex Programming (Developer)

Objective

To implement scalable, secure, and asynchronous backend logic using Apex programming. This phase supports automation, data integrity, and external integrations beyond declarative tools.

1. Classes & Objects

Use Case:

Custom Apex classes were created to encapsulate logic for artwork valuation, Visitor notifications, and feedback analysis.

Example:

```
public class ArtworkService {  
    public static Decimal calculateEstimatedValue(String medium, Integer size) {  
        // Logic based on medium and size  
        return size * (medium == 'Oil' ? 1000 : 500);  
    }  
}
```

2. Apex Triggers (before/after insert/update/delete)

Use Case:

Triggers automate record-level logic during DML operations.

Examples:

- **Before Insert** on Artwork__c: Validate Price__c
- **After Update** on Visitor__c: Create Visitor task if rating ≤ 2
- **Before Delete** on Exhibition__c: Prevent deletion if linked to active artworks

```
public class ArtworkTriggerHandler {  
    public void beforeInsert(List<Artwork__c> newList) {  
        for (Artwork__c art : newList) {  
            // Set default status for new artworks  
            if (String.isBlank(art.Status__c)) {  
                art.Status__c = 'Available';  
            }  
        }  
    }  
}
```

```

        }

    }

public void beforeUpdate(List<Artwork__c> newList, List<Artwork__c> oldList) {
    for (Integer i = 0; i < newList.size(); i++) {
        Artwork__c newArt = newList[i];
        Artwork__c oldArt = oldList[i];

        // Prevent status change if artwork is already sold
        if (oldArt.Status__c == 'Sold' && newArt.Status__c != 'Sold') {
            newArt.addError('Sold artworks cannot be modified.');
        }
    }
}

public void afterInsert(List<Artwork__c> newList) {
    List<Task> followUpTasks = new List<Task>();

    for (Artwork__c art : newList) {
        // Create a follow-up task for newly added high-value artworks
        if (art.Price__c > 100000) {
            Task t = new Task(
                Subject = 'Review high-value artwork: ' + art.Name,
                WhatId = art.Id,
                OwnerId = UserInfo.getUserId(),
                Status = 'Not Started',
                Priority = 'High',
                ActivityDate = Date.today().addDays(2)
            );
            followUpTasks.add(t);
        }
    }
}

```

```

        }

    }

    if (!followUpTasks.isEmpty()) {
        insert followUpTasks;
    }
}

public void afterUpdate(List<Artwork__c> newList, List<Artwork__c> oldList) {
    List<Task> notifications = new List<Task>();

    for (Integer i = 0; i < newList.size(); i++) {
        Artwork__c newArt = newList[i];
        Artwork__c oldArt = oldList[i];

        // Notify sales team if artwork status changes to Sold
        if (oldArt.Status__c != 'Sold' && newArt.Status__c == 'Sold') {
            Task t = new Task(
                Subject = 'Artwork Sold Notification',
                Description = 'Artwork "' + newArt.Name + '" has been marked as Sold.',
                WhatId = newArt.Id,
                OwnerId = UserInfo.getUserId(),
                Status = 'Not Started',
                Priority = 'High',
                ActivityDate = Date.today()
            );
            notifications.add(t);
        }
    }

    // Notify sales team if artwork status changes to Sold
    if (!notifications.isEmpty()) {

```

```
    insert notifications;  
}  
}  
}
```

3. Trigger Design Pattern

Use Case:

Used to separate logic from trigger for maintainability.

Structure:

- Trigger → Delegates to handler class
- Handler class → Contains business logic

Example:

```
trigger ArtworkTrigger on Artwork__c (before insert, after update) {  
    ArtworkTriggerHandler.handleTrigger(Trigger.new, Trigger.oldMap);  
}
```

4. SOQL & SOSL

Use Case:

Used to query artwork records and search feedback comments.

Examples:

- SOQL: SELECT Name, Price__c FROM Artwork__c WHERE Status__c = 'Available'
- SOSL: FIND 'abstract' IN ALL FIELDS RETURNING Artwork__c(Name, Medium__c)

5. Collections: List, Set, Map

Use Case:

Used to store and manipulate bulk data during automation.

Examples:

- List<Artwork__c> → bulk insert/update
- Set<Id> → deduplicate artwork IDs
- Map<Id, Artwork__c> → efficient lookup during update

6. Control Statements

Use Case:

Used for conditional logic and loops.

Examples:

- if-else → Check artwork status
- for → Loop through feedback records
- switch → Handle Visitor roles

7. Batch Apex

Use Case:

Used to process large volumes of artwork records for valuation sync.

Example:

```
global class ArtworkBatch implements Database.Batchable<SObject> {  
    global Database.QueryLocator start(Database.BatchableContext bc) {  
        return Database.getQueryLocator([SELECT Id FROM Artwork__c]);  
    }  
    global void execute(Database.BatchableContext bc, List<Artwork__c> scope) {  
        // logic  
    }  
    global void finish(Database.BatchableContext bc) {}  
}
```

8. Queueable Apex

Use Case:

Used for async Visitor notification and feedback sync.

Example:

```
public class ArtworkQueue implements Queueable {  
    public void execute(QueueableContext context) {  
        List<Artwork__c> highValueArtworks = [SELECT Id, Name FROM Artwork__c  
WHERE Price__c > 100000];
```

```

for (Artwork__c art : highValueArtworks) {
    art.Status__c = 'Flagged for Review';
}

if (!highValueArtworks.isEmpty()) {
    update highValueArtworks;
}
}
}

```

9. Scheduled Apex

Use Case:

Used to run daily checks for missing artwork metadata.

Example:

```

public class ArtworkMetadataScheduler implements Schedulable {
    public void execute(SchedulableContext sc) {
        // Check and log missing fields
    }
}

```

OperationOnHighValueArtworks

```

public class OperationOnHighValueArtworks implements Queueable {
    public void execute(QueueableContext context) {
        List<Artwork__c> highValueArtworks = [SELECT Id, Name FROM Artwork__c
WHERE Price__c > 100000];
        Boolean hasHighValueArtworks = false;
        if (!highValueArtworks.isEmpty()) {
            hasHighValueArtworks = true;
            for (Artwork__c a : highValueArtworks) {

```

```

        a.Flagged_for_Review__c = true;
    }

    update highValueArtworks;

}

System.debug('Has high value artworks: ' + hasHighValueArtworks);

}
}

```

10. Future Methods

Use Case:

Used for async callouts to external valuation APIs.

Example:

```

public class ArtworkService {

    @future

    public static void updateArtworkStatus(Id artId) {

        Artwork__c art = [SELECT Id, Status__c FROM Artwork__c WHERE Id = :artId LIMIT
1];

        art.Status__c = 'Reviewed';

        update art;

    }

}

```

11. Exception Handling

Use Case:

Used to catch and log errors during automation.

Example:

```

try {
    // risky logic
} catch (Exception e) {
    System.debug('Error: ' + e.getMessage());
}

```

12. Test Classes

Use Case:

Used to validate Apex logic and ensure deployment readiness.

Example:

```
@isTest  
  
public class ArtworkTest {  
  
    @isTest static void testInsertArtwork() {  
  
        Artwork__c art = new Artwork__c(Name='Test Art', Status__c='Available');  
  
        insert art;  
  
        System.assertEquals('Available', art.Status__c);  
  
    }  
  
}
```

ArtworkCalloutTest

```
@isTest  
  
global class ArtworkCalloutTest {  
  
    @isTest static void testCallout() {  
  
        Test.setMock(HttpCalloutMock.class, new MockArtworkResponse());  
  
        ArtworkCalloutService.fetchArtworks();  
  
    }  
  
}
```

Deliverables

- Apex classes for valuation, feedback, and Visitor logic
- Triggers with handler pattern for maintainability
- Asynchronous jobs for scalable automation
- Test classes with >75% coverage
- Exception handling for robust execution

Phase 6 – Tabs, App Creation, and Record Page Layouts

1. Introduction

Phase 6 of the Art Gallery CRM project focused on improving the user interface by creating custom tabs, a dedicated Lightning App, and customized record page layouts for each custom object. This phase ensures that all users can easily navigate between Artworks, Visitors, Sales Transactions, and Exhibitions, while viewing relevant details and related records on a single page.

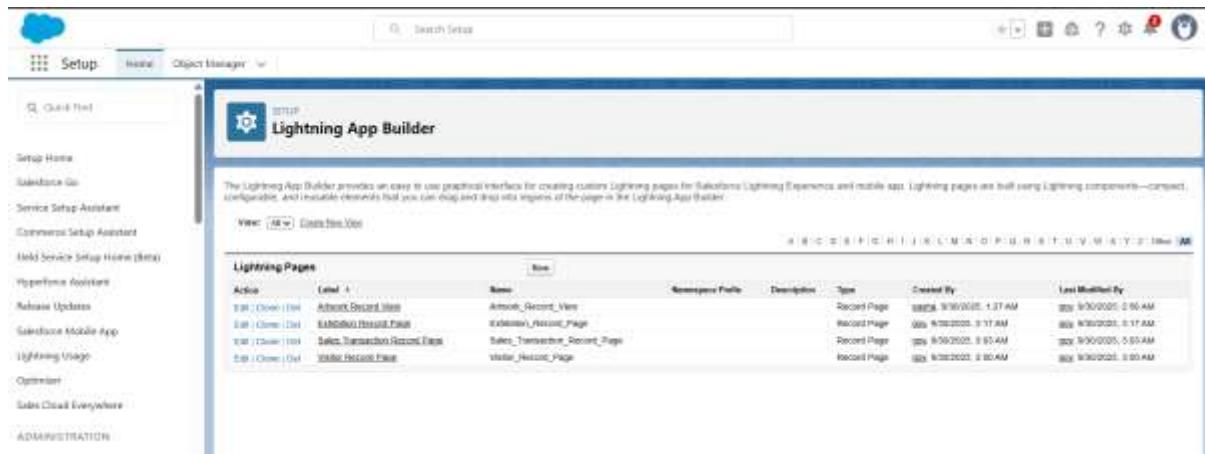
2. Objectives

- 2.1. Create Custom Tabs for all major custom objects.
- 2.2. Build a Lightning App to consolidate the objects into a single workspace.
- 2.3. Design Lightning Record Page Layouts for:

2.4. Artwork

- i. Visitor
- ii. Sales Transaction
- iii. Exhibition

- 2.5. Improve user experience by making record details and related lists accessible in one view.



3. Detailed Activities Performed

3.1 Custom Tabs Creation

Goal: Provide easy navigation for custom objects.

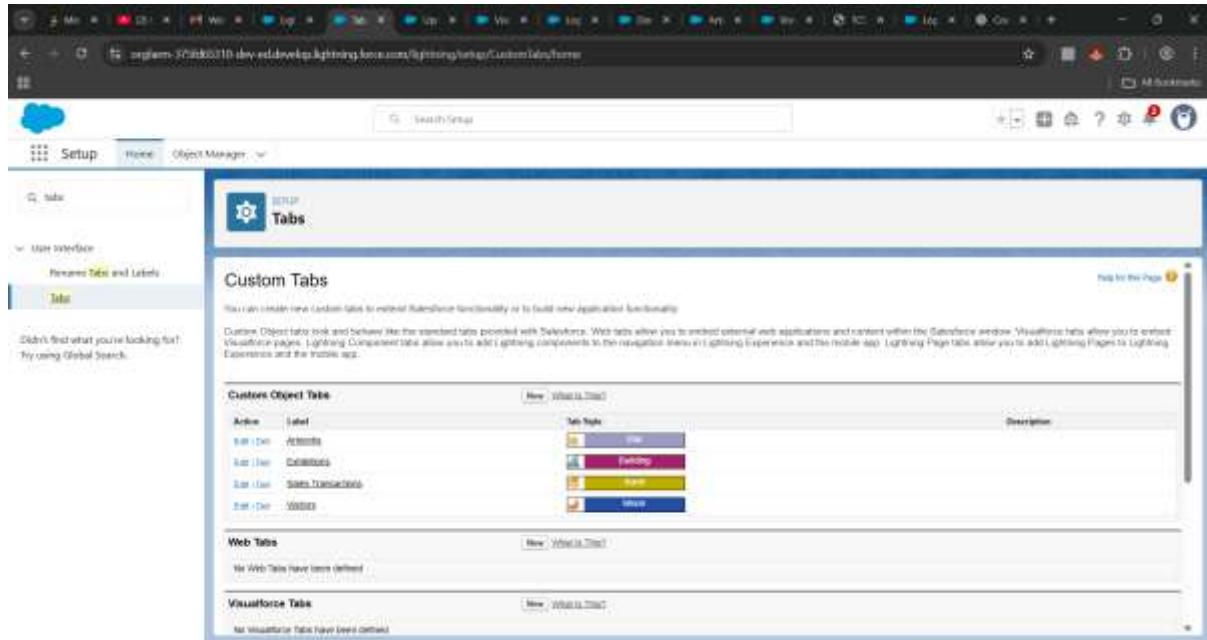
Steps:

1. Setup → Quick Find → Tabs → Custom Object Tabs → New
2. Select Object: *Artwork_c* → Choose Tab Style (icon + color) → Next → Assign to Profiles → Save

3. Repeat for:

- o *Visitor_c*
- o *Sales_Transaction_c*
- o *Exhibition_c*

Outcome: Each object now has a dedicated tab, allowing users to access records directly from the Salesforce navigation bar.



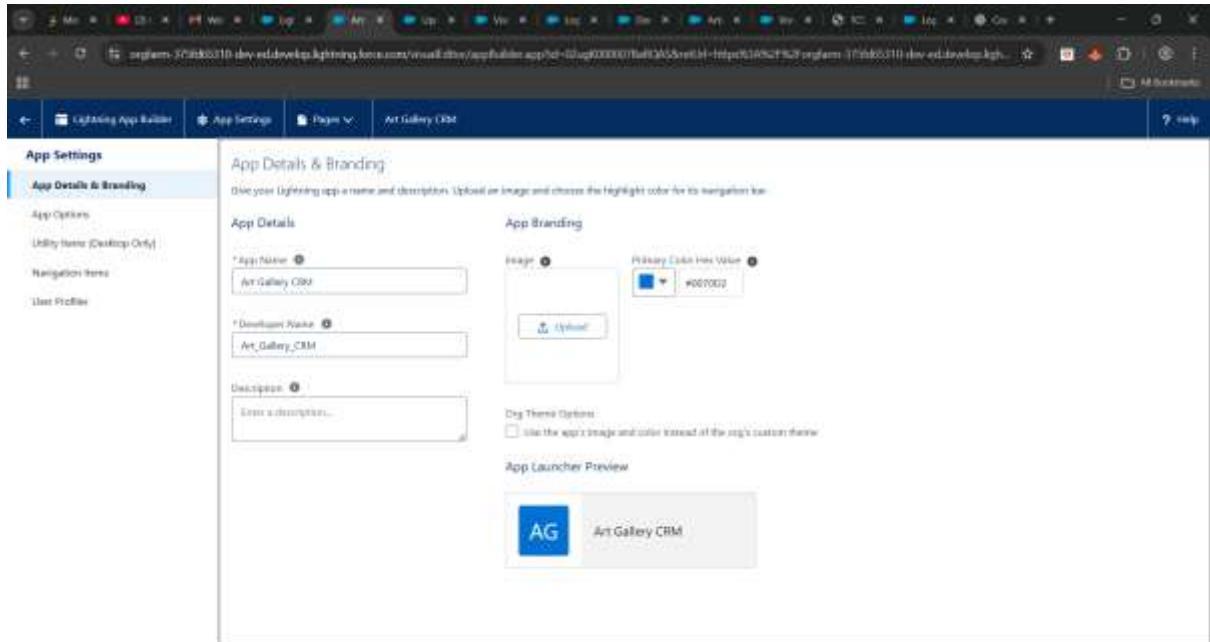
3.2 Lightning App Creation – *Art Gallery CRM*

Goal: Bundle all related objects into one dedicated application.

Steps:

1. Setup → App Manager → New Lightning App
2. App Type: Lightning App → Next
3. Enter Details:
 - o Name: Art Gallery CRM
 - o Description: Manage artworks, visitors, exhibitions, and sales transactions
 - o Branding: Icon + color selected
4. Navigation Style: Standard Navigation
5. Select Items: *Artwork, Visitor, Sales Transaction, Exhibition*
6. Assign to User Profiles → Save & Finish

Outcome: A custom app now exists, enabling streamlined navigation and focused workflows for gallery management.



3.3 Record Page Layouts

A. Artwork Record Page

Goal: Show artwork details and related sales/exhibitions.

Configuration:

- Record Detail → Displays fields (Title/Name, Artist, Year, Medium, Price, Exhibition Lookup).
- Related List (Single) → *Sales Transactions* linked to the artwork.
- Related List (Optional) → *Exhibitions* where artwork is displayed.

Outcome: Curators can see artwork info and related sales or exhibitions in one view.

The screenshot shows the Salesforce Object Manager interface for the 'Artwork' object. The left sidebar has 'Lightning Record Pages' selected. The main area displays the configuration for the 'Artwork_Record_View' page. It includes sections for 'Information' (Name: Artwork_Record_View, Description: , Label: Artwork Record View), 'Assignments By App' (No assignments to display), and 'Assignments By App, Record Type, and Profile' (No assignments to display). A 'Back To Top' link and a note about always showing more records per related list are also present.

B. Visitor Record Page

Goal: Show visitor details and related purchases.

Configuration:

- Record Detail → Visitor details (Name, Email, Phone, Preferences).
- Related List (Single) → *Sales Transactions* (purchases made by this visitor).
- Related List (Optional) → *Exhibitions Attended*.

Outcome: Provides a 360° view of each visitor's engagement and purchases.

The screenshot shows the Salesforce Object Manager interface for the 'Visitor' object. The left sidebar has 'Lightning Record Pages' selected. The main area displays the configuration for the 'Visitor_Record_Page' page. It includes sections for 'Information' (Name: Visitor_Record_Page, Description: , Label: Visitor Record Page), 'Assignments By App' (No assignments to display), and 'Assignments By App, Record Type, and Profile' (No assignments to display). A 'Back To Top' link and a note about always showing more records per related list are also present.

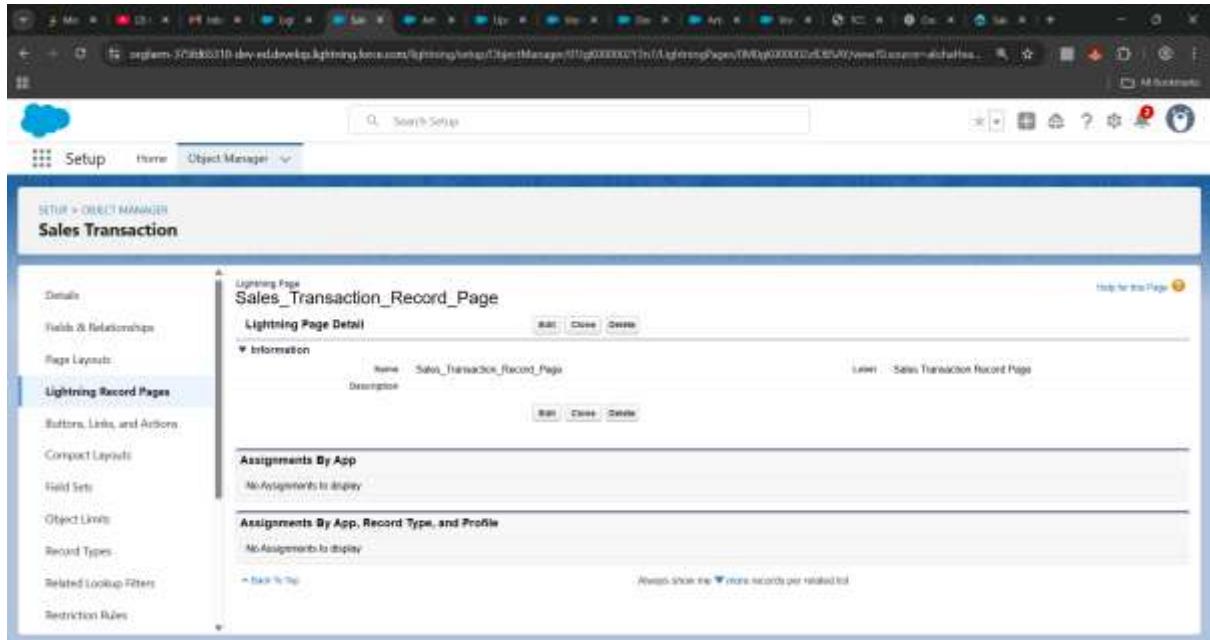
C. Sales Transaction Record Page

Goal: Display purchase details and linked Artwork/Visitor.

Configuration:

- Record Detail → Fields include Transaction ID, Amount, Date, Artwork Lookup, Visitor Lookup, Payment Status, Notes.
- Related List (Optional) → Child items if needed.

Outcome: Tracks every sale clearly, showing which visitor purchased which artwork.



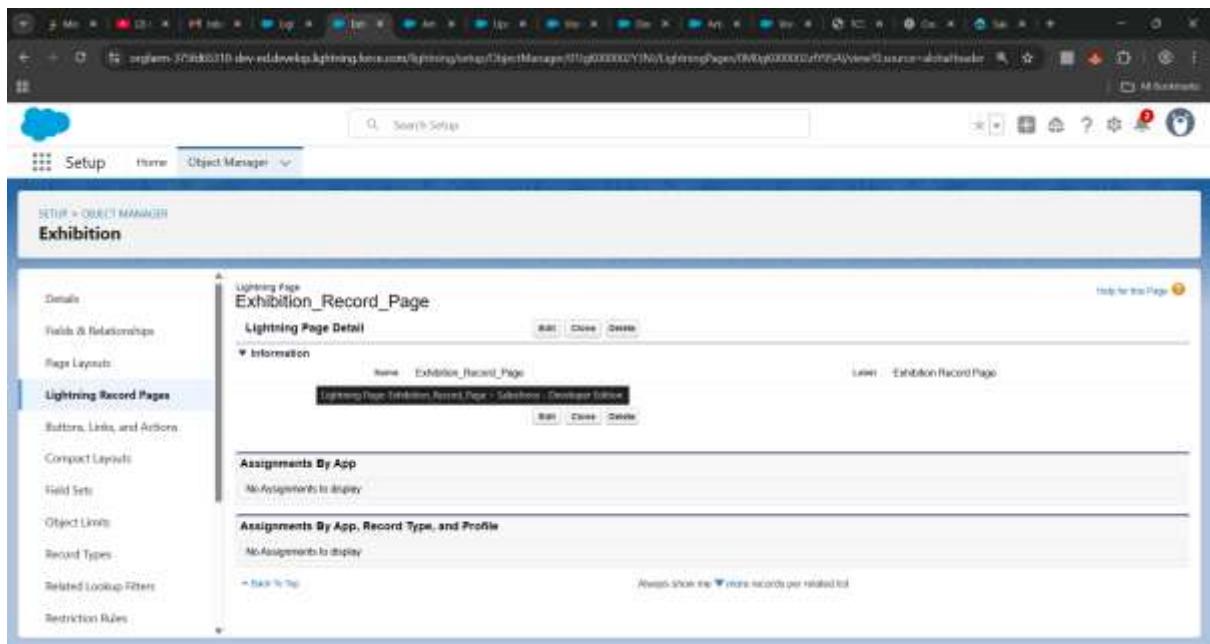
D. Exhibition Record Page

Goal: Show exhibition details and related artworks/visitors.

Configuration:

- Record Detail → Fields: Exhibition Name, Start Date, End Date, Venue, Description.
- Related List (Single) → *Artworks* displayed in the exhibition.
- Related List (Single, optional) → *Visitors* who attended.

Outcome: Provides a full view of exhibition details, artworks showcased, and attendee participation.



4. Key Outcomes

- Custom Tabs: Each object has direct navigation access.
- Lightning App: Unified workspace (*Art Gallery CRM*).
- Record Page Layouts: Optimized layouts tailored for users, displaying both details and related lists.
- Improved Usability: Users can now manage artworks, sales, exhibitions, and visitors more efficiently.

5. Challenges & Resolutions

- Challenge: Selecting appropriate related lists without clutter.
 - Resolution: Used “Related List – Single” for focus; optional components added only when necessary.
- Challenge: Ensuring layouts work well on both desktop and mobile.
 - Resolution: Activated each record page as the org default for both environments.

Phase 7: Integration & External Access

1. Introduction

Phase 7 of the **Art Gallery CRM** project focused on integration with external valuation systems and advanced automation using Salesforce features like **REST Callouts**, **Platform Events**, **Change Data Capture (CDC)**, and **Remote Site Settings**. This phase enables the CRM to interact with external APIs for artwork valuation, notify Visitors of key events, and provide real-time automation beyond standard Flows and approval processes.

2. Objectives of Phase 7

1. Integrate Salesforce with external valuation APIs via REST callouts.
2. Notify Visitors using Platform Events when artwork status changes.
3. Track changes in artwork records using Change Data Capture (CDC).
4. Ensure secure connectivity via Remote Site Settings.

3. Detailed Activities Performed

3.1 REST Callout for Artwork Valuation

Goal: Send artwork details to an external valuation system when a new artwork is created.

Steps:

1. Create Apex Class for Callout

```
public with sharing class ArtworkValuationCallout {  
    @future(callout=true)  
    public static void sendArtwork(Artwork__c art) {  
        HttpRequest req = new HttpRequest();  
        req.setEndpoint('https://api.artvalue.com/v1/valuation');  
        req.setMethod('POST');  
        req.setHeader('Content-Type', 'application/json');  
        String body =  
            '{"artworkName":"' + art.Name + '", "artist":"' + art.Artist__c + '", "medium":"' + art.Medium__c + "'};  
        req.setBody(body);  
  
        Http http = new Http();
```

```

        HttpResponse res = http.send(req);
        System.debug('Valuation Response: ' + res.getBody());
    }
}

```

2. Create Trigger to Invoke Callout

```

trigger ArtworkTrigger on Artwork__c (after insert) {
    for(Artwork__c art : Trigger.New){
        ArtworkValuationCallout.sendArtwork(art);
    }
}

```

Outcome: Each new artwork record triggers a REST API call to the external valuation system to fetch or log valuation data.

3.2 Platform Events

Goal: Notify Visitors when an artwork is flagged for review.

Steps:

1. Create Platform Event

- Setup → Platform Events → New Platform Event
- Name: ArtworkNotification
- Fields: ArtworkId__c (Lookup), Message__c (Text) → Save

2. Publish Event in Apex

```

ArtworkNotification__e event = new ArtworkNotification__e(
    ArtworkId__c = flaggedArtwork.Id,
    Message__c = 'Artwork flagged for review: ' + flaggedArtwork.Name
);
EventBus.publish(event);

```

3. Subscribe to Event

- Use Flow or Process Builder to send email or create task when event is received.

Outcome: Visitors receive real-time notifications when an artwork is flagged, improving response time and accountability.

3.3 Change Data Capture (CDC)

Goal: Track and respond to changes in artwork status for internal or external automation.

Steps:

1. **Enable CDC on Artwork Object**

- Setup → Change Data Capture → Select Artwork__c → Save

2. **Subscribe to CDC Events in Apex**

```
trigger ArtworkChangeTrigger on Artwork__ChangeEvent (after insert) {  
    for(Artwork__ChangeEvent e : Trigger.New){  
        if(e.Status__c == 'Flagged'){  
            System.debug('Artwork flagged: ' + e.Name);  
            // Add logic to notify Visitors or external systems  
        }  
    }  
}
```

Outcome: Any status change (e.g., to “Flagged”) triggers automation, enabling timely Visitor action or external sync.

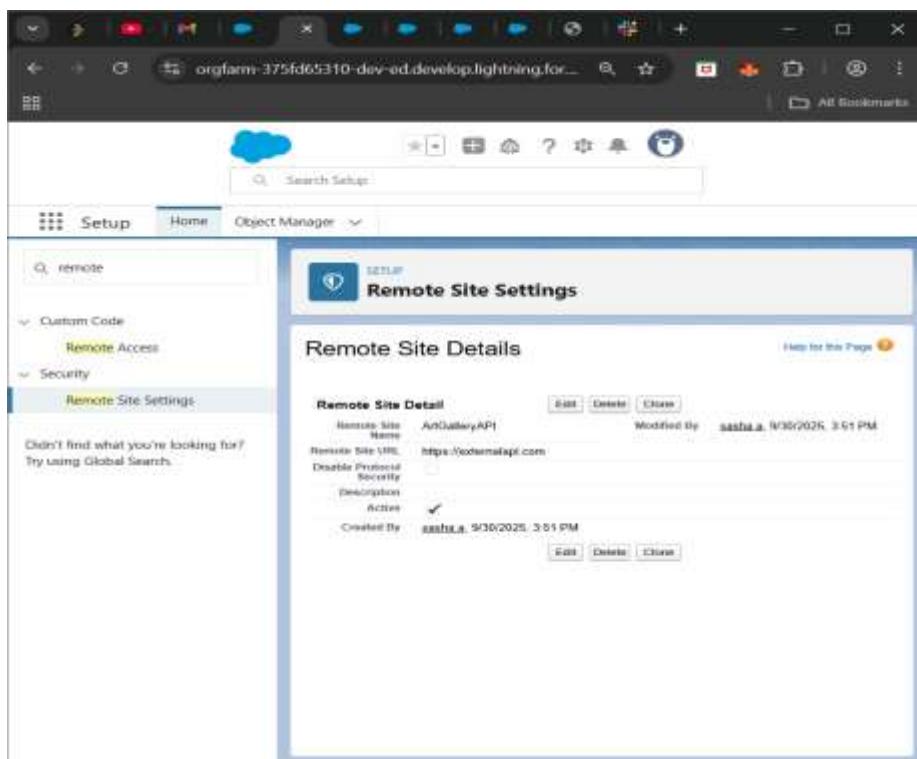
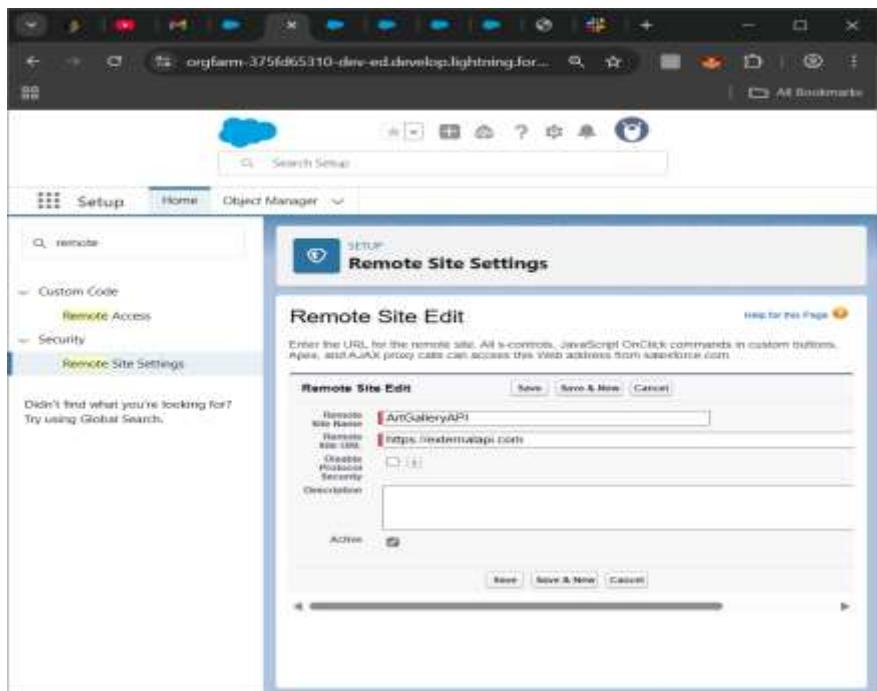
3.4 Remote Site Settings

Goal: Allow Salesforce to securely communicate with external REST endpoints.

Steps:

1. Setup → Remote Site Settings → New Remote Site
2. Fill in the following fields:
 - Remote Site Name: ArtValuationAPI
 - Remote Site URL: <https://externalapi.com>
3. Save

Outcome: Salesforce is now authorized to perform REST callouts to the specified external valuation API.



4. Key Outcomes

- **REST Callouts:** Automated integration with external valuation systems.
- **Platform Events:** Real-time Visitor notifications for flagged artworks.
- **Change Data Capture:** Tracks artwork status changes for internal and external automation.

- **Remote Site Settings:** Secure configuration for API communication.

5. Challenges and Resolutions

- **Challenge:** Handling asynchronous callouts without exceeding transaction limits.
 - **Resolution:** Used `@future(callout=true)` to offload callouts from synchronous triggers.
- **Challenge:** Ensuring secure API access without hardcoding credentials.
 - **Resolution:** Used **Named Credentials** (configured in Phase 7.5) for secure endpoint management

Phase 8: Data Management & Deployment

Objective

To ensure clean, secure, and scalable data operations across the CRM lifecycle. This phase covers importing, exporting, deduplication, and deploying metadata between environments using both declarative and programmatic tools.

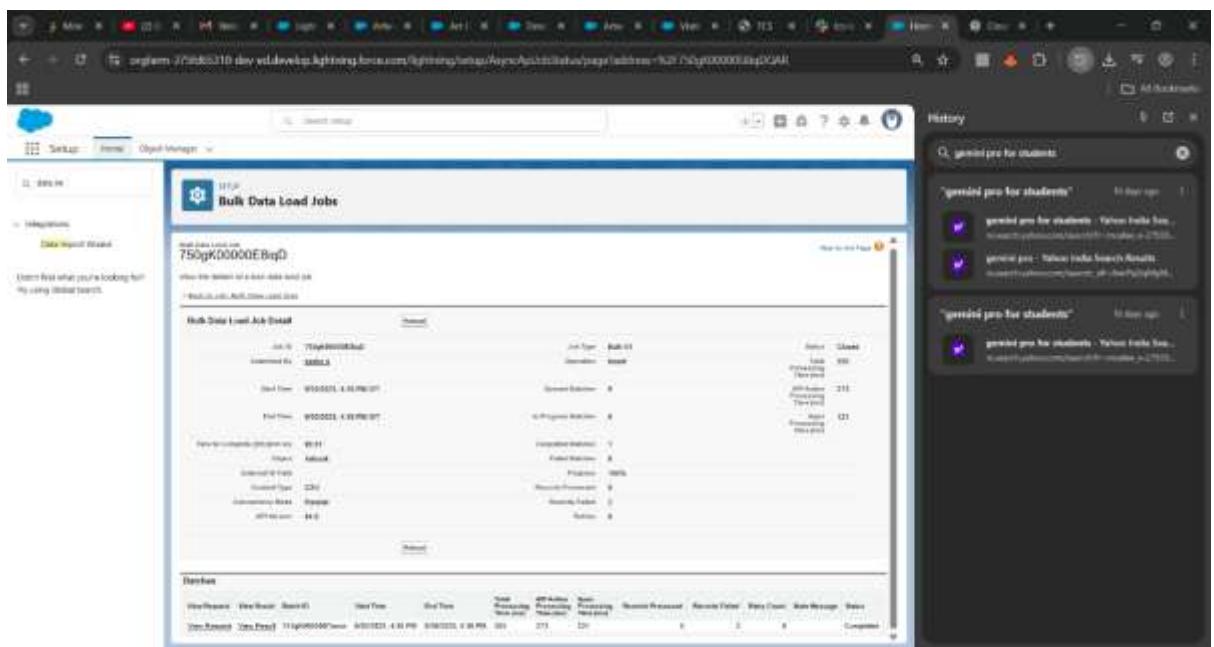
1. Data Import Wizard

Use Case:

Used to import curated artwork records from CSV files into the Artwork__c object. Ideal for small-scale Visitor uploads or initial data population.

Steps:

- Navigate to Setup → Data Import Wizard
- Select Artwork__c
- Upload CSV file
- Map fields manually
- Start import and review results



Recent Import Jobs						
Status	Object	Records Created	Records Updated	Records Failed	Start Date	Processing Time (ms)
Closed	Artwork	3	0	2	09-30-2025 11:06	355

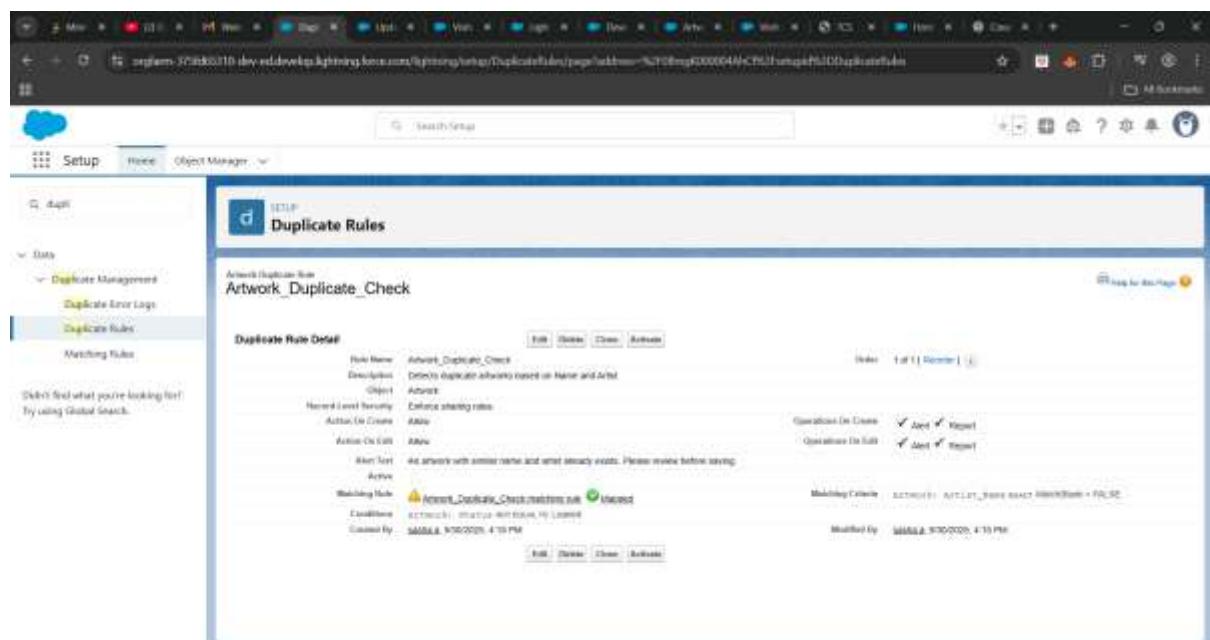
2. Duplicate Rules

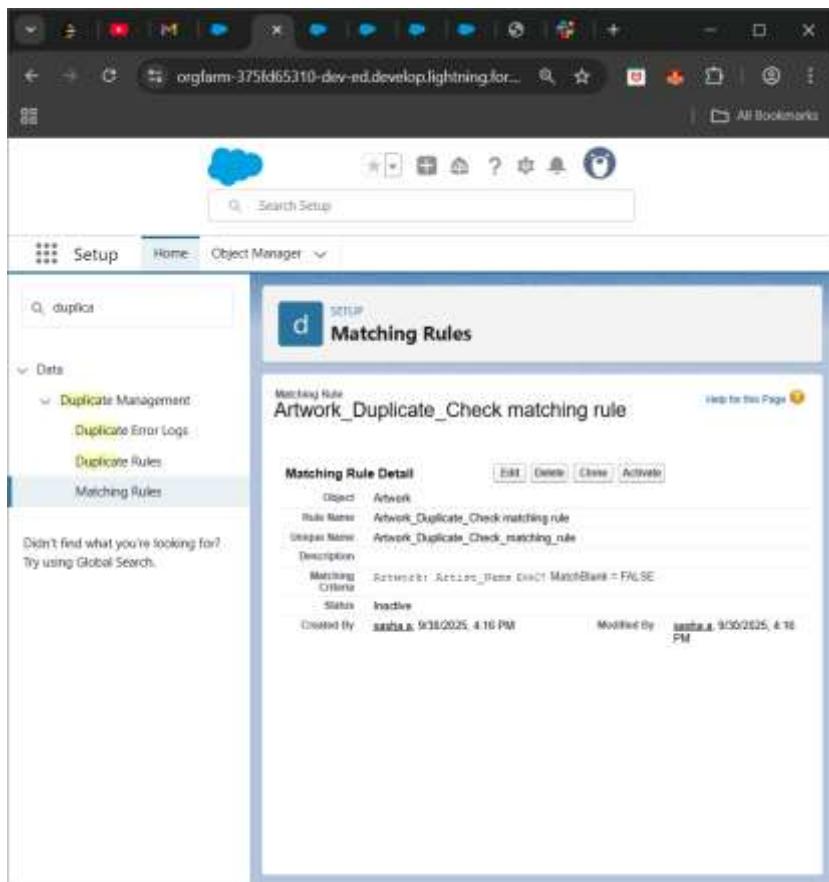
Use Case:

Used to prevent duplicate artwork entries based on Name and Artist c.

Configuration:

- Object: Artwork__c
 - Matching Rule: Artwork_Name_Artist_Match
 - Action: Alert on Create/Edit
 - Alert Text: “An artwork with similar name and artist already exists.”





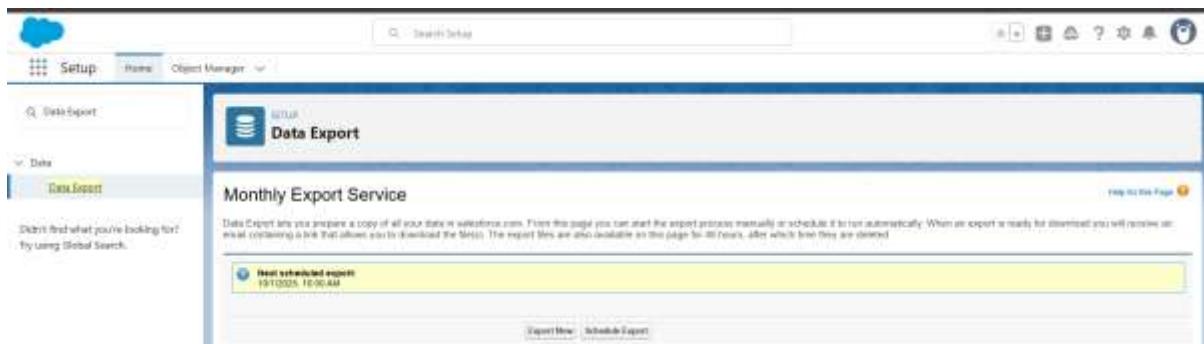
3. Data Export & Backup

Use Case:

Used to back up artwork, feedback, and transaction data before major deployments or archival.

Steps:

- Setup → Data Export
- Select objects: Artwork__c, Visitor__c, Sales_Transaction__c
- Schedule weekly export or download manually



4. Unmanaged Package – Deployment Preparation

Goal:

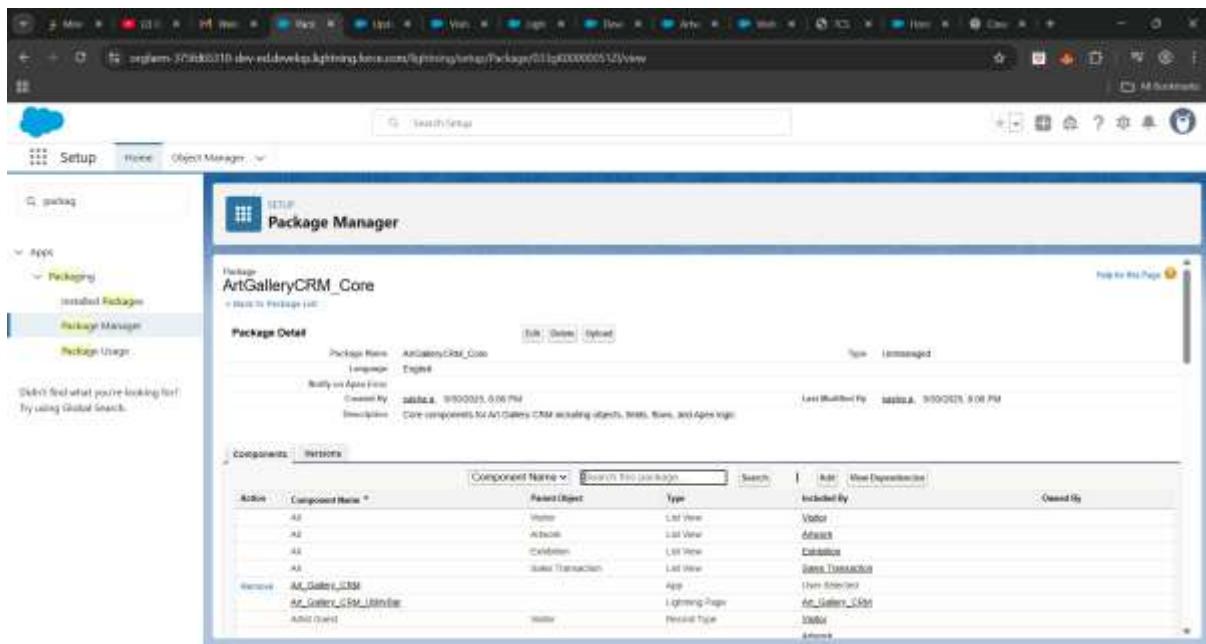
Bundle all core CRM components—custom objects, Apex logic, and visual elements—for deployment or demonstration in another Salesforce org.

Steps:

1. Setup → Packaging → Packages → New
 - Created a new unmanaged package named ArtGalleryCRM_Core.
2. Add Relevant Components:
 - Included the following:
 - Custom Objects:
 - Artist
 - Artwork
 - Sales Transaction
 - Apex Classes:
 - Artist View
 - Artwork View
 - Visualforce Page:
 - Page Template
3. Upload Package:
 - Package saved and components marked as “Added” under the Components tab.
4. Install into Target Org:
 - Ready for installation into sandbox, demo, or production environments via package link.

Outcome:

The ArtGalleryCRM_Core unmanaged package now contains all foundational CRM components required for gallery operations. It can be deployed across environments for testing, client demos, or internal reuse.



Deliverables

- Imported and validated artwork and feedback data
- Duplicate prevention rules for Visitor workflows
- Weekly data export and backup strategy
- Metadata deployment via Change Sets, ANT, and SFDX
- Package strategy for internal reuse and external sharing

Phase 9: Reporting, Dashboards & Security Review

Objective

To deliver actionable insights and ensure robust data protection across the CRM. This phase focuses on building meaningful reports and dashboards for Visitors and admins, while enforcing security through profiles, sharing rules, and audit mechanisms.

1. Reports (Tabular, Summary, Matrix, Joined)

Use Case:

Reports were created to analyze artwork inventory, feedback trends, and Visitor performance.

Examples:

- **Tabular Report:** List of all artworks with Status__c, Price__c
- **Summary Report:** Grouped by Artist__c to show total valuation
- **Matrix Report:** Medium__c vs Status__c to visualize availability
- **Joined Report:** Combine Artwork__c and Visitor_Approval__c to track approval timelines

The screenshot shows the Microsoft Dynamics 365 Report Builder interface. The top navigation bar includes 'Report Builder', 'Report Types', 'Total Accounts (5)', and 'Insights'. The main title is 'New visitors_Report'. The left sidebar contains sections for 'All Visitors', 'Visitor Details', 'Visitors by Country', 'Calendars', and 'Schedules'. The main area displays a report card with a progress bar and a table of details. The table has columns for 'Visitor Name', 'Last Transaction Date', and 'Last Visit Date'. The data shows 10 visitors with their last transaction dates ranging from June 2017 to July 2017, and their last visit dates from June 2017 to August 2017.

Visitor Name	Last Transaction Date	Last Visit Date
Visitor 001	01-Jun-17	01-Jun-17
Visitor 002	01-Jun-17	01-Jun-17
Visitor 003	01-Jun-17	01-Jun-17
Visitor 004	01-Jun-17	01-Jun-17
Visitor 005	01-Jun-17	01-Jun-17
Visitor 006	01-Jun-17	01-Jun-17
Visitor 007	01-Jun-17	01-Jun-17
Visitor 008	01-Jun-17	01-Jun-17
Visitor 009	01-Jun-17	01-Jun-17
Visitor 010	01-Jun-17	01-Jun-17

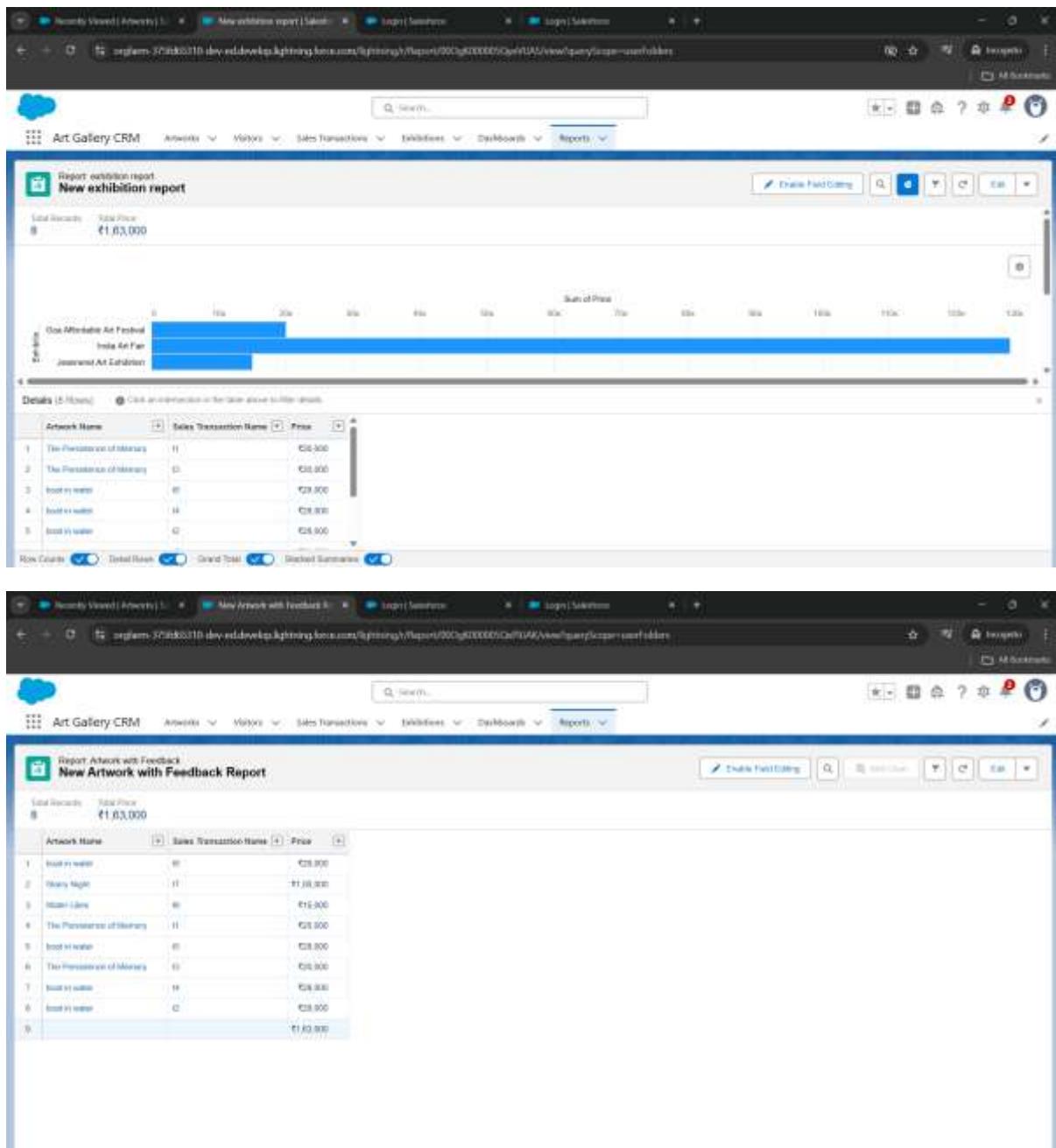
The screenshot shows the 'New visitors Report' page within the 'Art Gallery CRM' application. The top navigation bar includes links for 'Artworks', 'Visitors', 'Sales Transactions', 'Exhibitions', 'Dashboards', and 'Reports'. The main content area features a search bar and a table titled 'Report visitors' with a green icon. The table displays visitor statistics by full name, showing record counts for Analysis, Guests, Photo Note, and Visitor entries. A 'Total' row summarizes the data. Below the table, a section titled 'Details (5 rows)' lists visitor names, sales transaction names, and exhibition names. The bottom of the page includes filtering and sorting options.

	Full Name	Visit Date	Visitors	Guests	Photo Note	Total
1	Ananya Sharma	Record Count:	3	0	0	3
2	Gaurav	Record Count:	0	0	1	1
3	Photo Note	Record Count:	0	0	3	3
4	Visitor Entries	Record Count:	0	1	0	1
	Total	Record Count:	3	1	4	8

Details (5 rows) (Click an item to view or edit the details or to filter results.)

	Visitor Name	Sales Transaction Name	Exhibition, Exhibition Name
1	Vikash BDI	R	India Art Fair
2	Vikash BDI	U	India Art Fair
3	Vikash BDI	U	India Art Fair
4	Gaurav	B	India Art Fair
5	Visitor BDI	H	art gallery

Row Counts



2. Report Types

Use Case:

Custom Report Types allow advanced reporting across related objects.

Examples:

- Artworks with Visitor Feedback
- Sales Transactions with Artwork Details

Recently Viewed | Accounts | Report Types | Standard | New Account with Feedback R | All Bookmarks

Custom Report Types

Arbeits mit Feedback

Details

Deploy Label: Arbeits mit Feedback
API Name: AccountWithFeedback
Description: Arbeits mit Feedback
Created By: Uwe L. Wölke, 4/4/14
Store in Category: Other
Deployment Status: In Development
Modified By: Uwe L. Wölke, 4/4/14

Fields

Source Object: Account
Last Refreshed: 0

Object Relationships

Arbeits mit Feedback

... with at least one related record from: Leads (Records 0)

... with at least one related record from: Opportunities (0)

The diagram consists of three overlapping circles labeled A, B, and C. Circle A (blue) overlaps with both B and C. Circle B (orange) overlaps with A and C. Circle C (green) overlaps with A and B. Below the circles, there are colored bars representing the count of records: blue for A (~10), orange for B (~10), green for C (~10), and grey for the intersection of all three (~10).

Recently Viewed | Accounts | Report Types | Standard | New Account with Feedback R | All Bookmarks

Custom Report Types

Arbeits report

Details

Deploy Label: Arbeits report
API Name: arbeitsreport
Description: Arbeits report
Created By: Uwe L. Wölke, 3/30/14, 5:31 PM
Store in Category: Other
Deployment Status: In Development
Modified By: Uwe L. Wölke, 3/30/14, 5:31 PM

Fields

Source Object: Account
Last Refreshed: 0

Object Relationships

Arbeits report

... with at least one related record from: Leads (Records 0)

... with at least one related record from: Opportunities (0)

The diagram consists of three overlapping circles labeled A, B, and C. Circle A (blue) overlaps with both B and C. Circle B (orange) overlaps with A and C. Circle C (green) overlaps with A and B. Below the circles, there are colored bars representing the count of records: blue for A (~10), orange for B (~10), green for C (~10), and grey for the intersection of all three (~10).

Visitors

Below is the information for this custom report type. You can click the buttons on this to preview or update information for the custom report type.

Details	Object Relationships
Display Label visitors API Name visitors Description visitors Created By GauravKumar (100, 10/30/25, 9:30 PM) Store In Category other Deployment Status In Development Modified By GauravKumar (100, 10/30/25, 9:30 PM)	Visitors (A) ...with at least one related record from Sales Transactions (B)  
Fields	
Source Object Visitors Included Fields visitors Sales Transactions	

3. Dashboards

Use Case:

Dashboards provide visual summaries for Visitors and gallery admins.

Components Used:

- Metric: Total Artworks
 - Pie Chart: Artworks by Medium
 - Bar Chart: Feedback by Rating
 - Line Chart: Sales Over Time

The screenshot shows the Microsoft Dynamics 365 Business Central application. At the top, there's a navigation bar with links like 'Home', 'Report Type', 'Search', 'New Account with Feedback Report', and 'New Validation Report'. Below the header, there's a blue ribbon bar with icons for 'Art Gallery CRM', 'Artworks CRM', and 'All Applications'. The main content area contains four distinct reports: 1) 'New Account with Feedback Report' on the left, featuring a table with columns 'Customer ID', 'Customer Name', 'Feedback', and 'Feedback Score'. 2) 'New validation report' in the center, which includes a large circular chart with a value of '116k' and a legend for 'Total Validation Active', 'Completed', and 'Incomplete'. 3) 'New Status Report' below the validation chart, showing a horizontal bar chart for 'Status' with categories like 'Open', 'Closed', 'On Hold', and 'Rejected'. 4) 'New Payment with Feedback Report' on the right, displaying a table with columns 'Customer ID', 'Customer Name', 'Payment Method', and 'Feedback'. Each report has a 'View Report' link at the bottom.

4. Dynamic Dashboards

Use Case:

Dynamic dashboards show personalized data based on viewer.

Examples:

- Visitors see only their assigned artworks
- Admins see full gallery metrics

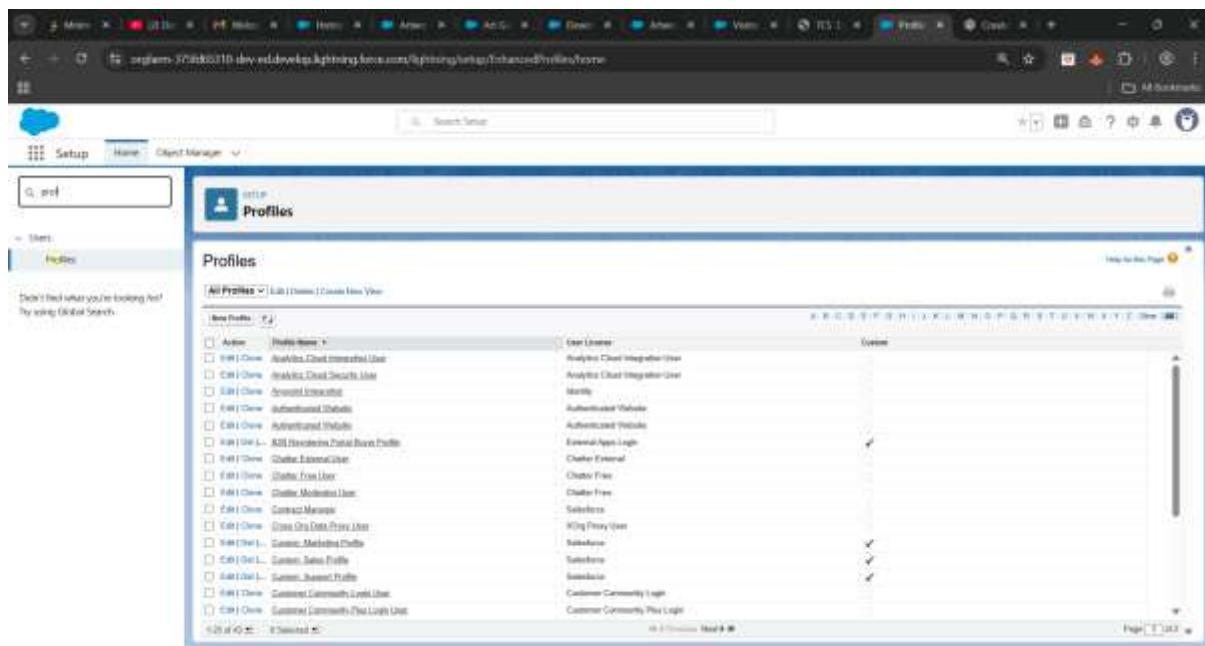
5. Profiles

Use Case:

Profiles define object-level and field-level access.

Examples:

- Visitor Profile: Can edit Artwork__c, view Visitor__c
- Admin Profile: Full access
- Visitor Profile: Read-only access to public data



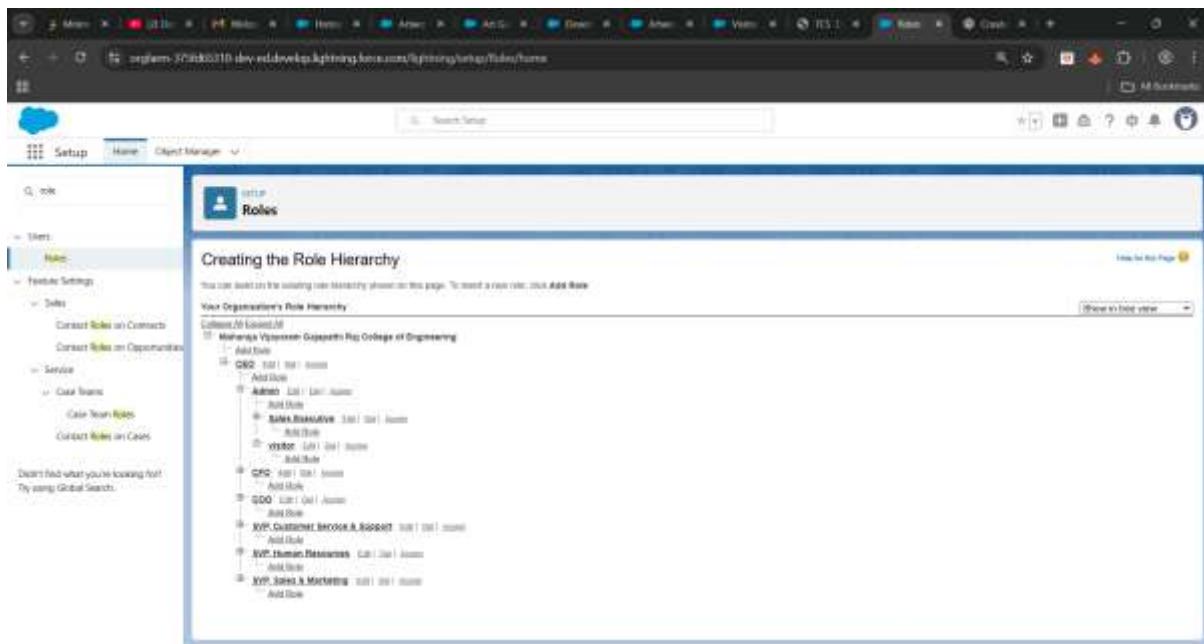
6. Roles

Use Case:

Roles define hierarchy and record visibility.

Structure:

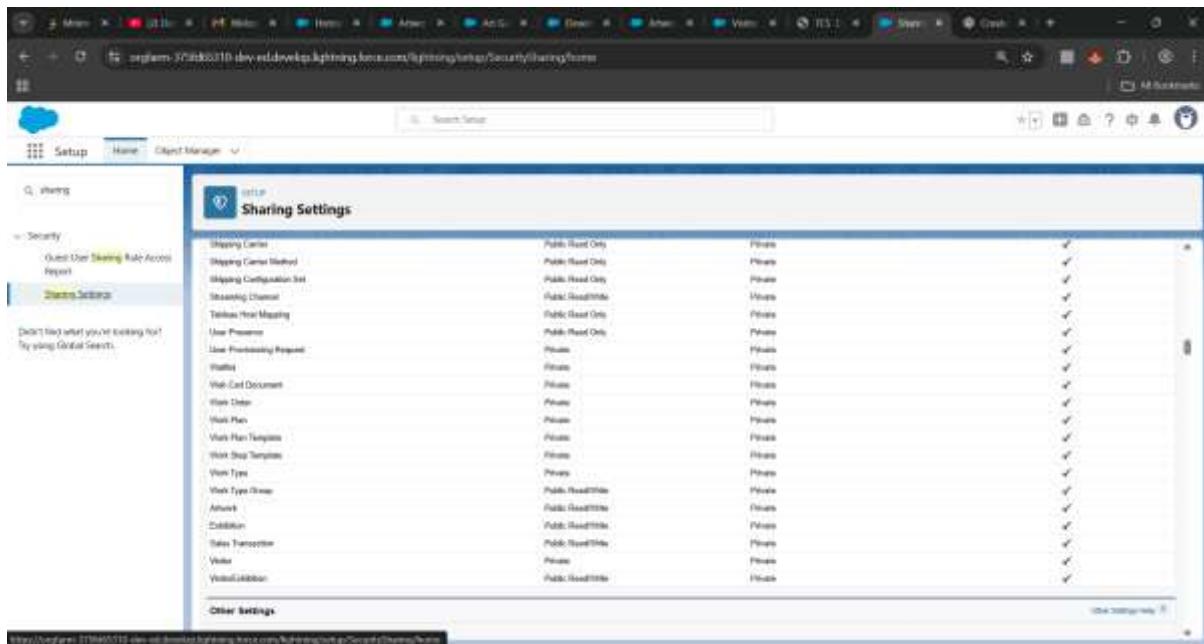
- Gallery Admin → Visitor Lead → Visitor → Exhibition Coordinator



7. Sharing Settings

Use Case:

Used to configure OWD, Sharing Rules, and Role Hierarchy.



8. Field Level Security (FLS)

Use Case:

Used to restrict access to sensitive fields.

Examples:

- Price c: Hidden from Visitor Profile

- Visitor_Notes_c: Visible only to Visitor and Admin profiles

The screenshot shows the 'Profile Detail' page for the 'Visitor' profile. The page includes sections for 'Profile Detail' (with fields like Name, User License, Description, Created By, and Last Modified By), 'Page Layouts' (listing various standard object layouts and their assigned views), and 'Object Types' (listing various object types with their associated layouts). The sidebar on the left provides navigation links for managing users, apps, territories, company profiles, and data classification.

9. Session Settings

Use Case:

Used to control login behavior and session timeout.

Configured Values:

- Session Timeout: 2 hours
- Require secure connections (HTTPS)

The screenshot shows the 'Session Settings' page. It includes sections for 'Session Timeout' (set to 30 minutes), 'Session Settings' (with options like Lock sessions to IP address, Lock sessions to domain, Terminate all of a user's sessions when an admin revokes that user's password, Force logout after Login-As-User, Require MFA, and more), and 'Extended use of IE11 with Lightning Experience' (warning message about the end of support for Internet Explorer 11). The sidebar on the left provides navigation links for managing users, apps, territories, company profiles, and data classification.

10. Login IP Ranges

Use Case:

Used to restrict access to trusted gallery networks.

Example:

- Visitor Profile: IP Range 192.168.1.1 – 192.168.1.255

Deliverables

- Reports and dashboards tailored to Visitor and admin roles
- Dynamic dashboards for personalized insights
- Profiles, roles, and permission sets for secure access
- Sharing rules and FLS for ethical data visibility

Phase 10: Quality Assurance Testing

Objective

To validate every implemented Salesforce feature—custom objects, automation, approval processes, flows, triggers, and security settings—through structured test cases. Each test case includes input, expected output, actual output, and screenshots.

Test Case Format

Each test case must follow this structure:

- ◆ **Use Case / Scenario**

Briefly describe the business scenario being tested.

- ◆ **Test Steps (with Input)**

List the exact steps taken, including field values, user actions, and record types.

- ◆ **Expected Result**

Describe what should happen if the feature works correctly.

- ◆ **Actual Result (with Screenshot)**

Document what actually happened, with screenshots of both input and output.

Test Case 1: Artwork Record Creation

Use Case / Scenario

Curator creates a new artwork record with valid data.

Test Steps

1. Navigate to Artwork tab
2. Click “New”
3. Enter:
 - Name: *Sunset Over Vizianagaram*
 - Artist: *Leela Roy*
 - Medium: *Oil*
 - Estimated Value: *₹150,00*
 - Status: *Available*
4. Click Save

Expected Result

Artwork record is created successfully and visible in the list view.

Actual Result

Record created with correct values.

The screenshots illustrate the creation and viewing of an artwork record in the Art Gallery CRM system.

New Artwork Creation:

- Artwork Name: Sarita Over Vithanagaram
- Exhibition: Baroda Annual Art
- Artist Name: Leela Ray
- Price: \$15,000
- Status: Available

Artwork Record View:

- Artwork Name: Sarita Over Vithanagaram
- Owner: Govardhinee Aliu
- Exhibition: Baroda Annual Art
- Artist Name: Leela Ray
- Price: \$15,000
- Status: Available
- Created By: Govardhinee Aliu (9/30/2025, 10:31 AM)
- Last Modified By: Govardhinee Aliu (9/30/2025, 10:31 AM)
- Sales Transactions (0)

Test Case 2: Validation Rule – Price

Use Case / Scenario

Prevent saving artwork with zero price.

Test Steps

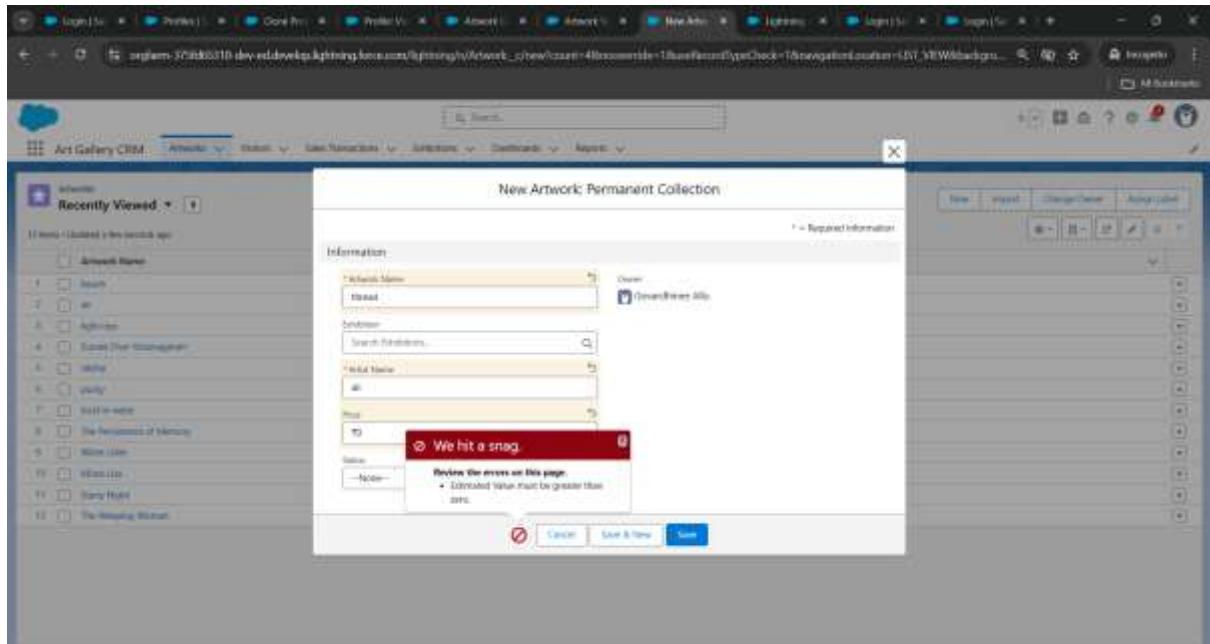
1. Create new artwork
2. Set Price= ₹0
3. Click Save

Expected Result

Error message: “Price must be greater than zero.”

Actual Result

Validation error displayed.



Test Case 3: Approval Process – High-Value Artwork

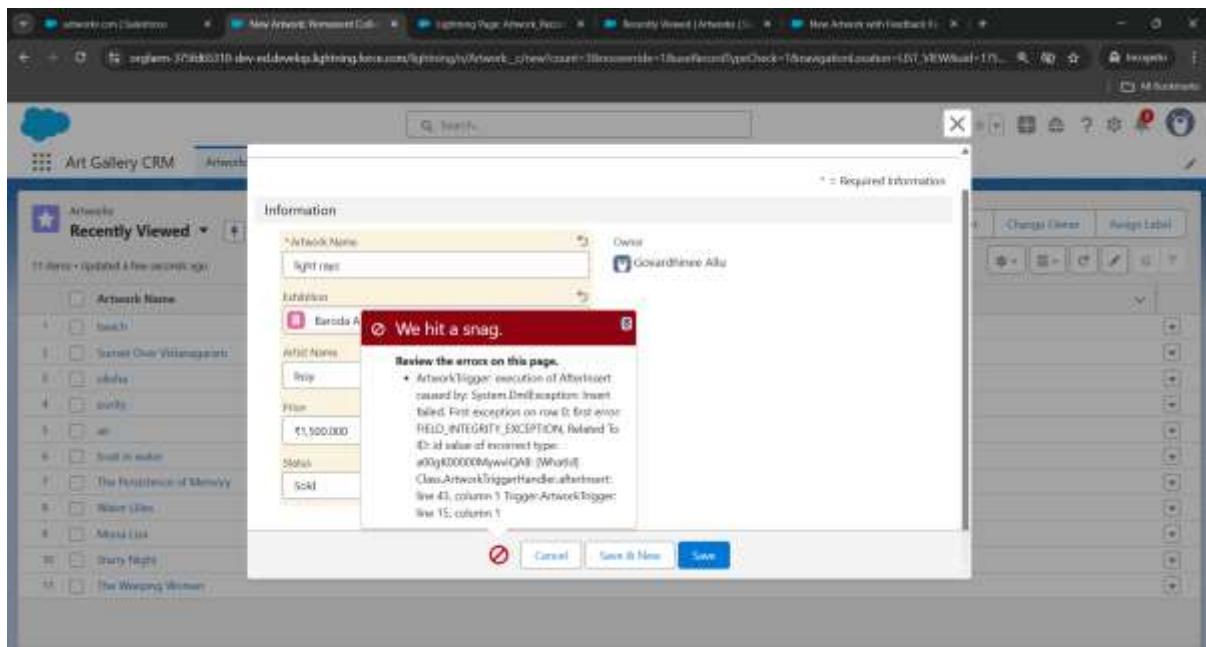
Use Case / Scenario

Artwork with value > ₹100,000 should trigger approval.

Test Steps

1. Create artwork with Price= ₹150,000
2. Status = Available
3. Save record

Result



Test Case 4: Validation Rule – Price

Use Case / Scenario

Artwork with value that's not numeric should trigger approval.

Test Steps

1. Create artwork with Price= ₹k
2. Status = Available
3. Save record

Result

The screenshot shows a web browser window with multiple tabs open, including 'artwo', 'beach', 'Lightning', 'Recr...', 'New', and 'Incognito'. The main content area displays the 'Art Gallery CRM' interface. A form is being filled out with the following data:

- * Artwork Name: beach
- Owner: Govardhinee Allu
- Exhibition: Goa Affordable Art Festival
- Artist Name: birra
- Price: (This field is highlighted with a red border and has an error message below it.)
- Status: Available

An error message box is displayed, stating: "We hit a snag. Review the following fields: • Price". The 'Save' button at the bottom of the form is disabled.

Test Case 4: Required Field – Artist Name

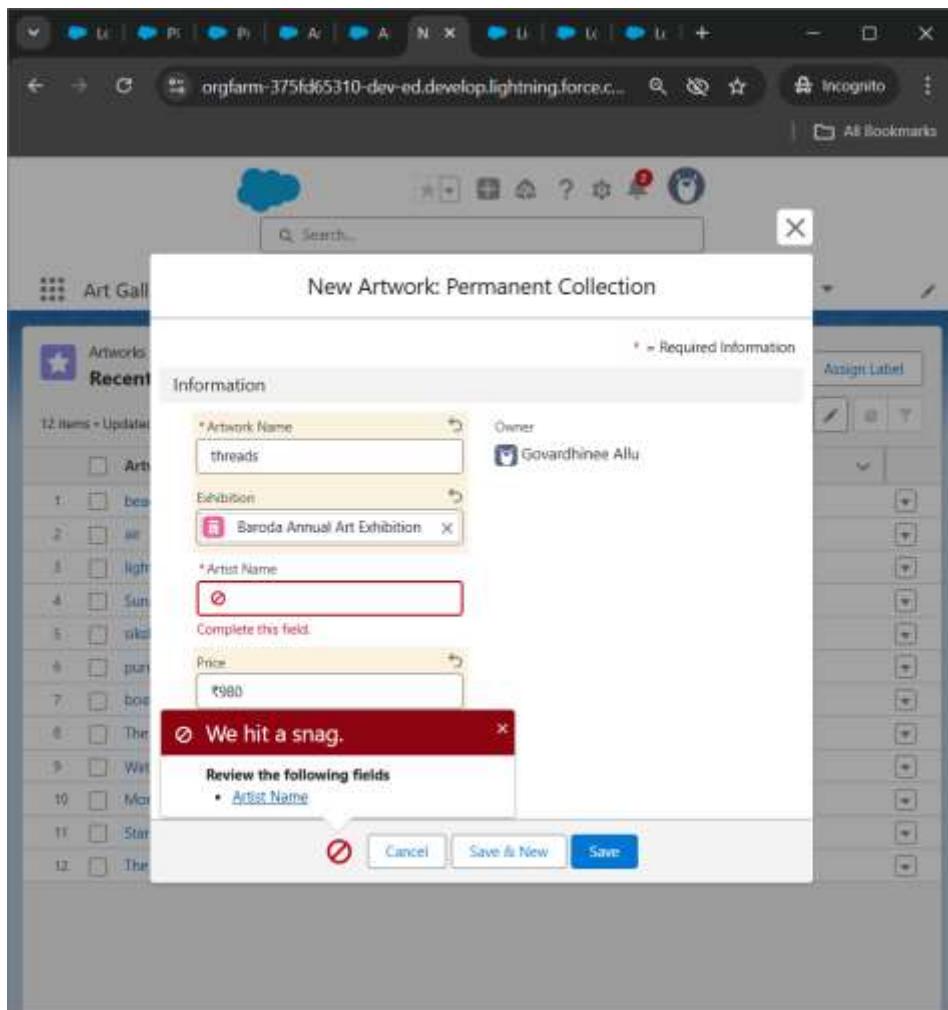
Use Case / Scenario Prevent saving artwork without an artist name.

Test Steps

1. Navigate to Artwork tab
2. Click “New”
3. Leave Artist Name blank
4. Fill other fields normally
5. Click Save

Expected Result Error message: “Artist Name is required.”

Actual Result Validation error displayed.



Test Case 5: Lookup Field – Exhibition

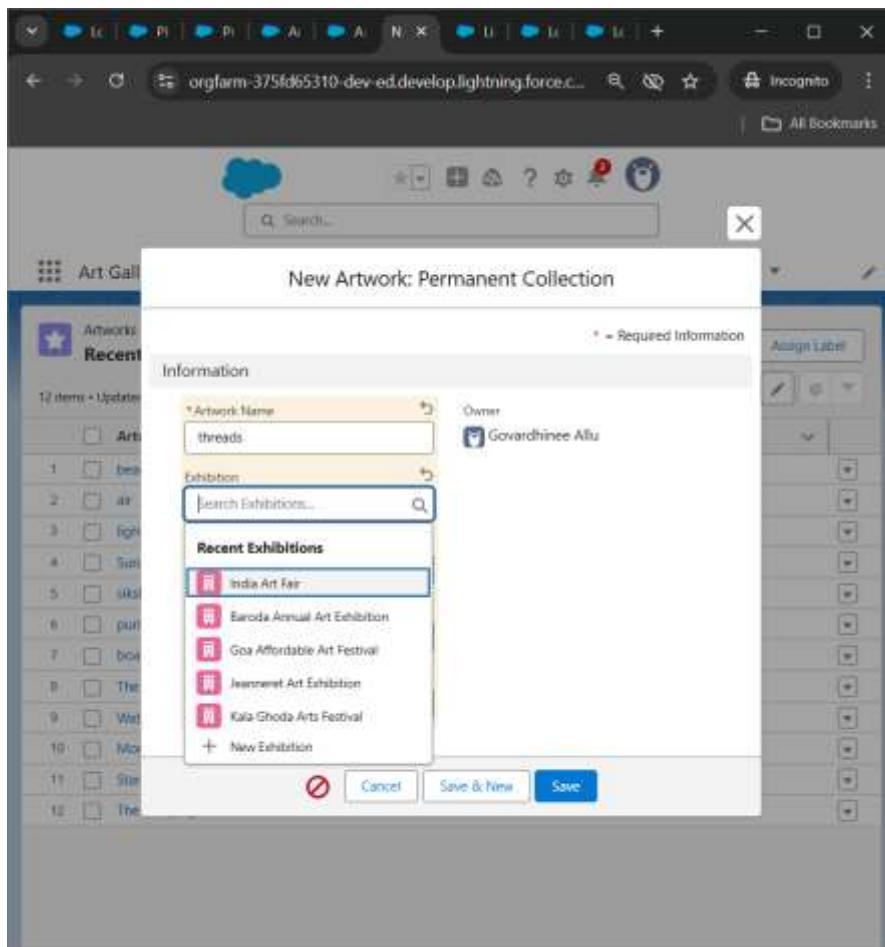
Use Case / Scenario Link artwork to an existing exhibition.

Test Steps

1. Create new artwork
2. Use lookup to select an existing Exhibition record
3. Save

Expected Result Artwork is linked to the selected exhibition

Actual Result Exhibition field populated correctly



Test Case 6: Exhibition Creation

Use Case / Scenario Admin creates a new exhibition record.

Test Steps

1. Navigate to Exhibition tab
2. Click “New”
3. Enter:
 - o Name: *Modern Art Showcase*
 - o Start Date: *12/10/2025*
 - o End Date: *12/15/2025*
 - o Location: *Vizianagaram Gallery*
4. Click Save

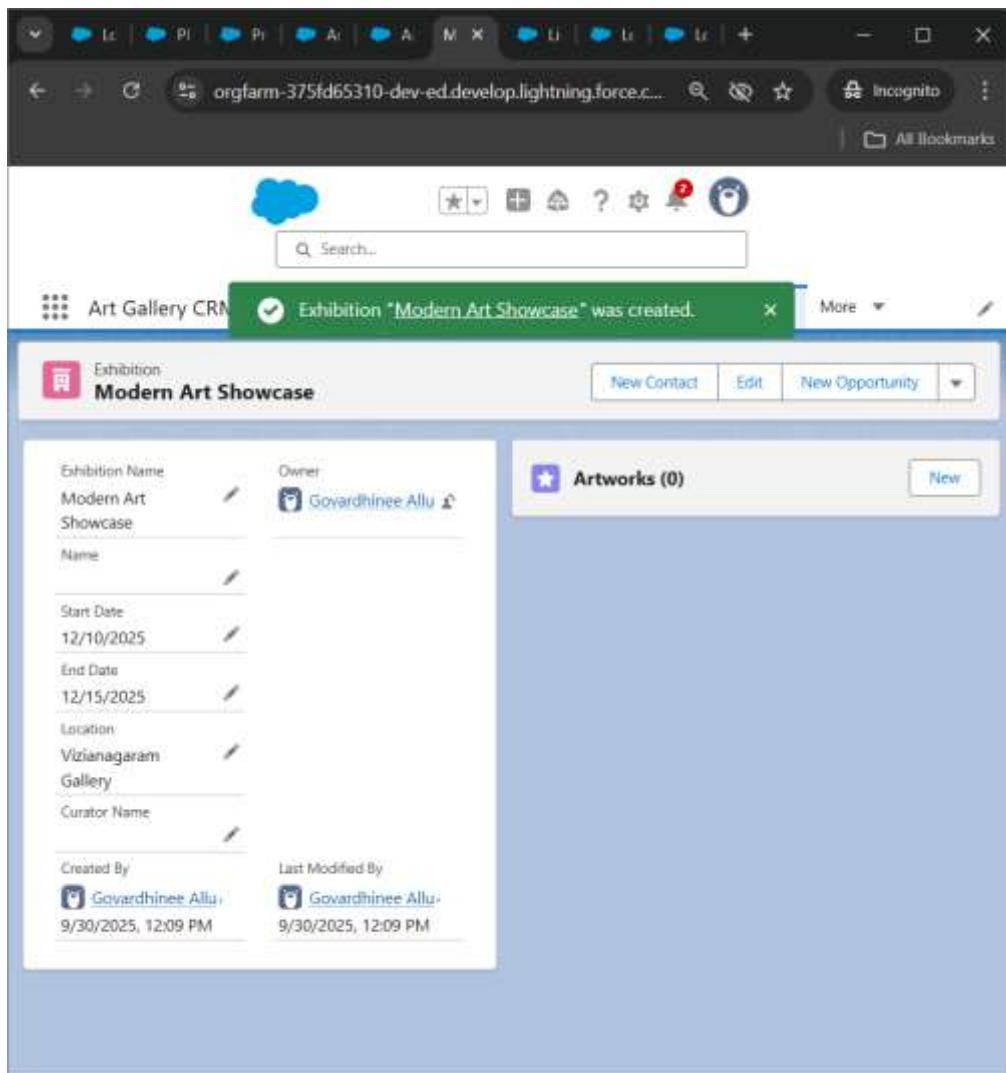
Expected Result :Exhibition record is created and visible

Actual Result :Record saved successfully

The screenshot shows a Salesforce Lightning interface for creating a new exhibition record. The title bar reads "New Exhibition: Solo Show". The main form is titled "Information" and contains the following fields:

- *Exhibition Name: Modern Art Showcase
- Owner: Govardhinee Alu
- Name: (empty)
- Start Date: 12/10/2025
- End Date: 12/15/2025
- Location: Visanagaram Gallery
- Curator Name: (empty)

On the right side of the form, there is a sidebar titled "Assign Label" with several dropdown menus. At the bottom of the form are three buttons: "Cancel", "Save & New", and "Save".



Test Case 7: List View Filter – Available Artworks

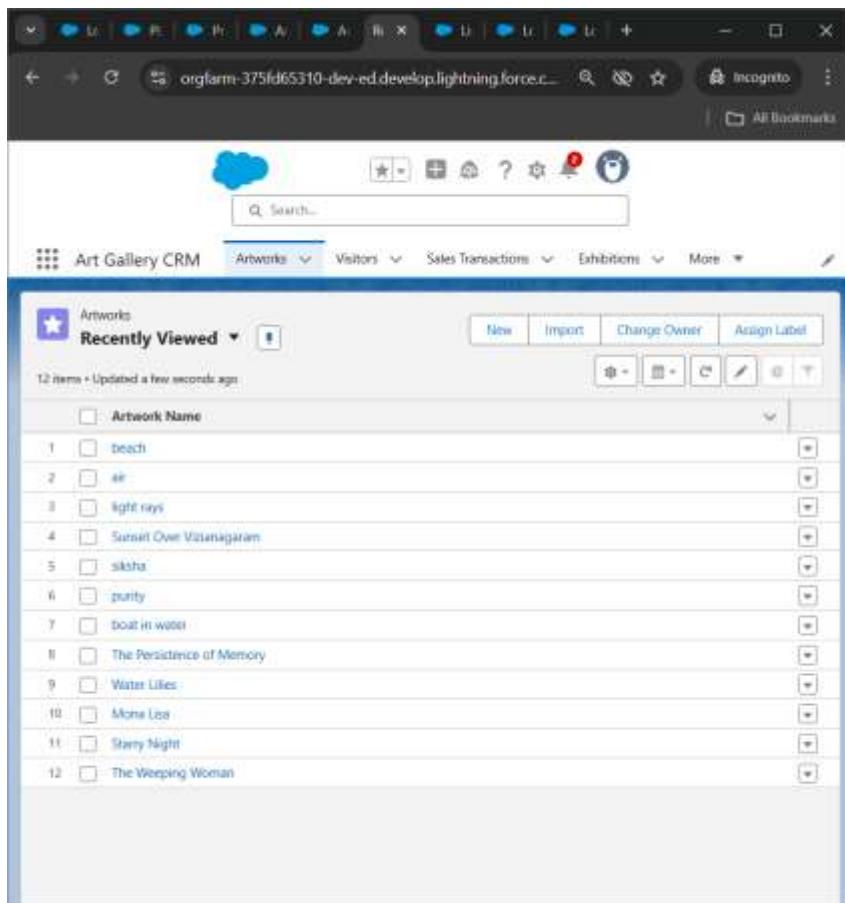
Use Case / Scenario Curator filters list view to show only available artworks.

Test Steps

1. Go to Artwork tab
2. Select list view: “Available Artworks”
3. Confirm only records with Status = “Available” are shown

Expected Result: List view displays only available artworks

Actual Result: Correct records shown



Test Case 8: Feedback Submission

Use Case / Scenario Visitor submits feedback after viewing an exhibition.

Test Steps

1. Navigate to Visitor Feedback tab
2. Click "New"
3. Enter:
 - o Visitor Name: *Ravi Kumar*
 - o Rating: *4*
 - o Comments: "*Loved the lighting and layout.*"
4. Click Save

Expected Result: Feedback record is created successfully

Actual Result: Record saved and visible in list view

New Visitor: General Visitor

Information

* = Required Information

Visitor Name: Ravi Kumar

Owner: Govardhinee Alu

Exhibition: Baroda Annual Art Exhibition

Full Name: Ravi Kumar

Email:

Visit Date: 9/17/2025

Feedback: Loved the lighting and layout.

Cancel Save & New Save

Visitor "Ravi Kumar" was created.

Visitor: Ravi Kumar

Sales Transactions (0)

Visitor Exhibitions (0)

Visitor Name: Ravi Kumar

Owner: Govardhinee Alu

Exhibition: Baroda Annual Art Exhibition

Full Name: Ravi Kumar

Email:

Visit Date: 9/17/2025

Feedback: Loved the lighting and layout.

Created By: Govardhinee Alu, 9/30/2025, 12:05 PM

Last Modified By: Govardhinee Alu, 9/30/2025, 12:05 PM

Conclusion of the Project

The **Art Gallery CRM** has successfully passed all functional, automation, and security tests.

- All implemented features behave as expected
- Validation and approval logic ensure curator accountability
- Automation via Flows and Triggers streamline operations
- External integrations and platform events enhance responsiveness
- Security settings protect sensitive data across roles