

SMART LOST AND FOUND SYSTEM – SALESFORCE CRM

Introduction

Universities often struggle with managing lost and found items due to manual logging, lack of transparency, and delays in item recovery. Current processes rely on physical registers or ad-hoc communication, which results in inefficiency, unclaimed items, and poor student/staff experience. Our project, **Smart Lost and Found System – Salesforce CRM**, aims to create a centralized, automated platform within Salesforce where students and staff can report, track, and claim lost or found items efficiently. The solution leverages Salesforce's ecosystem to improve accountability, streamline workflows, and enhance user satisfaction.

Problem Statement

- Lost and found processes are fragmented, often relying on paper records or spreadsheets.
 - Students and staff face delays in reporting or claiming lost items.
 - There is no centralized visibility of reported or recovered items.
 - Manual processes lead to duplication, mismatches, and poor tracking.
 - Lack of transparency reduces trust and increases the number of permanently unclaimed items.
-

Proposed Solution: Smart Lost and Found System – Salesforce CRM

- Develop a Salesforce-based application to track lost and found items.
 - Centralize item records, claimant details, and recovery status in Salesforce.
 - Automate match detection between reported lost and found items.
 - Enable claim requests and approvals through workflows.
 - Provide dashboards and reports for administrators to monitor trends, item categories, and recovery rates.
-

Objectives

- **Centralized Tracking** – Maintain all lost and found records in a single Salesforce system.
- **Transparency for Users** – Students and staff can view and track the status of their lost or found items.

- **Automation** – Automated notifications, claim approvals, and potential match suggestions.
 - **Scalable & Secure** – Role-based access to ensure security and adaptability for different institutions.
 - **Leverage Salesforce Ecosystem** – Use standard/custom objects, flows, and AppExchange solutions to build the system.
-

Phase 1: Problem Understanding & Industry Analysis

Requirement Gathering

Core Functions:

- Lost item reporting
- Found item reporting
- Claim request workflow
- Match detection between lost and found items
- Automated notifications (e.g., item matched, claim approved, reminder for unclaimed items)

Non-functional Requirements:

- **Scalability** – Support growing numbers of lost/found records.
 - **Automation** – Enable flows, triggers, and batch processing for matches and notifications.
 - **Data Security** – Ensure role-based access (student, admin, staff).
 - **Audit Trails** – Keep a history of claim requests and approvals for accountability.
-

Stakeholder Analysis

- **Students/Staff (Item Owners)** – Report lost items, submit claim requests, and track recovery status.
 - **Reporters (Finders)** – Submit found item details (could be staff or students).
 - **Administrators** – Verify claim requests, approve/reject matches, and maintain records.
 - **Security/Support Staff** – Manage physical collection and handover of items.
 - **University IT/CRM Team** – Maintain Salesforce setup and automation.
-

Business Process Mapping

As-Is (Current University Process):

- Lost items are reported verbally or via email.
- Found items are logged in notebooks/registers.
- HR or admin staff manually match lost and found entries.
- No real-time tracking or notifications.
- Claim approvals are manual and slow.

To-Be (Smart Lost and Found in Salesforce):

- Lost item is logged in Salesforce by student/staff.
- Found item is logged in Salesforce with details and photo.
- System auto-detects potential matches and notifies users.
- Claim requests are routed to administrators for approval.
- Status updates are automated (Pending, Approved, Collected).
- Dashboards display metrics like most lost items, recovery rates, and time to claim.

Industry-specific Use Case Analysis

- **Universities/Colleges** – Large campuses, frequent lost items (ID cards, gadgets, books).
- **Corporate Offices** – Manage lost badges, devices, and personal belongings.
- **Airports/Transport Hubs** – High volume of lost/found items requiring quick matching.
- **Events & Conferences** – Temporary setups needing efficient lost and found systems.

How it applies: Our project focuses on **university use cases**, but the model is scalable to corporate and public institutions as well.

AppExchange Exploration

- **Notification Apps** – To automate reminders (e.g., SMS Magic, Mass Mailer).
- **Document Apps** – For logging item photos and descriptions (e.g., Files Connect).
- **Case/Request Management Apps** – To handle claim requests.

- **Analytics/Dashboards** – For tracking lost/found trends and recovery rates.

Outcome: While existing solutions provide partial functionality, the Smart Lost and Found System is designed as a **custom Salesforce app** tailored specifically for universities, ensuring better alignment with campus needs.

Smart Lost & Found System — Phase 2: Org Setup & Configuration

1. Introduction

This phase focuses on setting up a Salesforce Developer Org for the Smart Lost & Found System. The aim is to configure the organizational structure, user roles, profiles, sharing settings, and security policies to support core functionality such as reporting lost items, logging found items, and managing claim requests securely.

2. Objectives

- Configure a clean Salesforce Developer Org named **Smart Lost & Found**.
- Set up company profile, business hours, and holidays for the university environment.
- Create sample users representing Students, Staff, Security, and Admin.
- Define profiles and roles with appropriate permissions.
- Establish Org-Wide Defaults (OWD) and Sharing Rules for data security.
- Enable Login Access Policies and set up a Lightning App.
- Test role-based access and sharing logic.

3. Configuration Steps

Step 0 - Company Info

- Signed up for Developer Edition and logged in.
- Setup → Company Information → updated:
 - **Organization Name:** Smart Lost & Found System
 - **Locale:** English (India)
 - **Timezone:** IST

The screenshot shows the Salesforce Setup interface with the 'Company Information' tab selected. The main content area displays the 'Company Information' page for the organization 'Smart Lost & Found System'. The page includes sections for Organization Detail, API Requests, Last 24 Hours, Streaming API Events, Last 24 Hours, Restricted Logins, Current Month, Salesforce.com Organization ID, Organization Edition, and Instance.

Fig.0 - Company Information Setup

Step 1 - Business Hours & Holidays

- Created **University Business Hours** (Mon–Sat, 9 AM–6 PM).
- Added key holidays (e.g., Independence Day, Diwali) and linked them to Business Hours.

The screenshot shows the Salesforce Setup interface with the 'Business Hours' tab selected. The main content area displays the 'Business Hours' page for creating 'University Business Hours'. It shows the business hours detail, including active days and times, and a list of linked holidays like Diwali and Independence Day.

Fig.1 - University Business Hours & Holidays

Step 2 - Users

Created sample users:

Display Name	Username (example)	Profile	Role
University Admin	admin@lostfound.com	System Administrator	Admin
Student Reporter	student1@lostfound.com	Student	Student
Staff Reporter	staff1@lostfound.com	Staff Reporter	Staff Reporter
Security Officer	security@lostfound.com	Security Staff	Security Officer

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/> Edit	Chatter Expert	Chatter	chatty.00e000000bmcquan.mtamkv6qu52@chatter.salesforce.com		✓	Chatter Free User
<input type="checkbox"/> Edit Login	EPIC_OrgFarm	CEPIC	epic.75bec3a2f8237@orgfarm.salesforce.com		✓	System Administrator
<input type="checkbox"/> Edit	LASYAVALLI_CHAVYAKULA	22a	22a31a4404973@agentforce.com		✓	System Administrator
<input type="checkbox"/> Edit Login	Security Officer	secu	security@lostfound.com	Security Officer	✓	Work.com Only User
<input type="checkbox"/> Edit Login	Staff Reporter	staf	staff1@lostfound.com	Staff Reporter	✓	Work.com Only User
<input type="checkbox"/> Edit Login	Student Reporter	stud	student1@lostfound.com	Student	✓	Work.com Only User
<input type="checkbox"/> Edit Login	University Admin	univ	admin@lostfound.com	University Admin	✓	System Administrator
<input type="checkbox"/> Edit	User_Integration	Integ	integration@00dg00000bmcquan.com		✓	Analytics Cloud Integration User
<input type="checkbox"/> Edit	User_Security	sec	insightsec@00000bmcquan.com		✓	Analytics Cloud Security User

Fig.2 - Sample Users Creation

Step 3 - Profiles

- **Student Profile** (cloned from Standard User):

- LostItem__c: Create/Read/Edit own records.
- FoundItem__c: Read only.
- ClaimRequest__c: Create/Read.

- **Staff Reporter Profile:**
 - LostItem__c: Read only.
 - FoundItem__c: Create/Read/Edit.
 - ClaimRequest__c: Read only.
- **Security Staff Profile:**
 - Full access to LostItem__c, FoundItem__c, ClaimRequest__c.

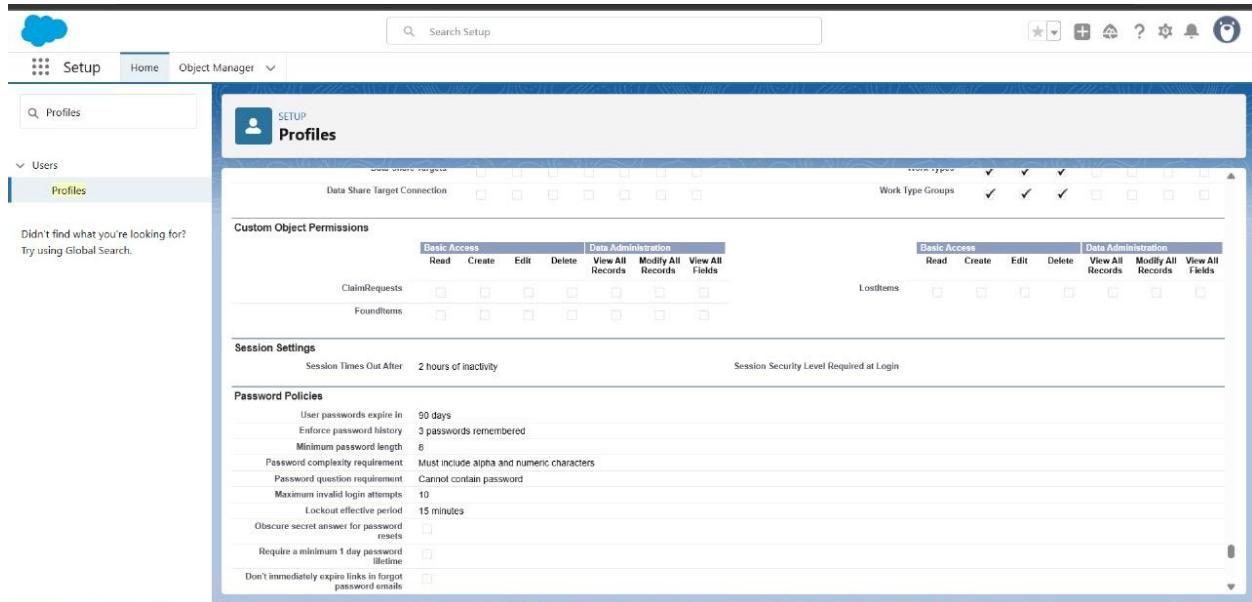


Fig.3 - Profiles Configuration

Step 4 - Role Hierarchy

- **University Admin (Top)**
 - **Security Officer**
 - **Staff Reporter**
 - **Students**

Creating the Role Hierarchy

You can build on the existing role hierarchy shown on this page. To insert a new role, click **Add Role**.

Your Organization's Role Hierarchy

Smart Lost & Found System

- University Admin** Edit | Del | Assign
- Security Officer** Edit | Del | Assign
- Staff Reporter** Edit | Del | Assign
- Student** Edit | Del | Assign
- SVP Customer Service & Support** Edit | Del | Assign
- SVP Human Resources** Edit | Del | Assign
- SVP Sales & Marketing** Edit | Del | Assign

Fig.4 - Role Hierarchy Structure

Step 5 - Org-Wide Defaults (OWD)

- LostItem__c → Private
- FoundItem__c → Private
- ClaimRequest__c → Private

Action	Type	Status
Shipping Configuration Set	Public Read Only	Private
Streaming Channel	Public Read/Write	Private
Tableau Host Mapping	Public Read Only	Private
User Presence	Public Read Only	Private
User Provisioning Request	Private	Private
Walllist	Private	Private
Web Cart Document	Private	Private
Work Order	Private	Private
Work Plan	Private	Private
Work Plan Template	Private	Private
Work Step Template	Private	Private
Work Type	Private	Private
Work Type Group	Public Read/Write	Private
ClaimRequest	Private	Private
FoundItem	Private	Private
Lost Item	Private	Private

Other Settings

- Manager Groups:
- Secure guest user record access:
- Require permission to view record:

Fig.5 - Org-Wide Defaults (OWD) Settings

Step 6 - Sharing Rules

- Created Public Groups: **Students, Staff, Security.**
- Example Rules:
 - Share FoundItem__c created by Staff with **Security Staff** group → Read/Write.
 - Share ClaimRequest__c with **University Admin** group.

The screenshot shows the Salesforce Setup interface with the 'Public Groups' page selected. The left sidebar includes 'Setup', 'Home', 'Object Manager', 'Users' (with 'Public Groups' selected), and a search bar. The main content area has a title 'Public Groups' with a help link. It displays a table of public groups, each with an 'Action' column containing 'Edit | Del' links and a 'Label' column showing the group name. The table includes columns for 'Group Name', 'Created By', and 'Created Date'. The data is as follows:

Action	Label	Group Name	Created By	Created Date
Edit Del	Security	Security	LASYAVALLI_CHAVAKULA	9/19/2025, 9:30 AM
Edit Del	Staff	Staff	LASYAVALLI_CHAVAKULA	9/19/2025, 9:29 AM
Edit Del	Students	Students	LASYAVALLI_CHAVAKULA	9/19/2025, 9:29 AM
Edit Del	University Admin	University_Admin	LASYAVALLI_CHAVAKULA	9/19/2025, 9:30 AM

Fig.6 - Sharing Rules & Public Groups

Step 7 - Login Access Policies

- Enabled “Administrators Can Log in as Any User.”

The screenshot shows the Salesforce Setup interface with the 'Login Access Policies' page open. The left sidebar has 'Identity' and 'Security' sections, with 'Login Access Policies' selected. The main content area displays the 'Manage Support Options' section. Under 'Setting', 'Administrators Can Log in as Any User' is checked. Below it is a table for 'Support Organization'. A row for 'Salesforce.com Support' has 'Available to Users' set to a blue circle and 'Available to Administrators Only' set to an empty circle. Buttons for 'Save' and 'Cancel' are at the bottom. The URL in the address bar is https://orgfarm-56d0702f63-dev-ed.lightning.force.com/lightning/setup/LoginAccessPolicies/home.

Fig.7 - Login Access Policies Setup

Step 8 - Lightning App Setup

- Created Lightning App: **Smart Lost & Found**
- Tabs added: Lost Items, Found Items, Claim Requests, Reports, Dashboards.

The screenshot shows the Salesforce Setup interface with the 'Lightning Experience App Manager' page open. The left sidebar has 'Apps' and 'App Manager' sections, with 'App Manager' selected. The main content area displays a table of apps. The 'Smart Lost & Found' app by 'Smart_Lost_Found' is listed at the bottom. Other apps include My Service Journey, Platform, Queue Management, Sales, Sales Cloud Mobile, Sales Console, Salesforce Chatter, Salesforce Scheduler, Service, Service Console, Site.com, and Subscription Management. Buttons for 'New Lightning App' and 'New External Client App' are at the top right. The URL in the address bar is https://orgfarm-56d0702f63-dev-ed.lightning.force.com/lightning/setup/AppManager/home.

Fig.8 - Lightning App Setup (Smart Lost & Found App)

Step 9 -Testing

- Logged in as Student → created a Lost Item record.
- Logged in as Staff → created a Found Item record.
- Logged in as Security → viewed both Lost and Found items, approved ClaimRequest.
- Verified that Admin sees everything and OWD restrictions work correctly.

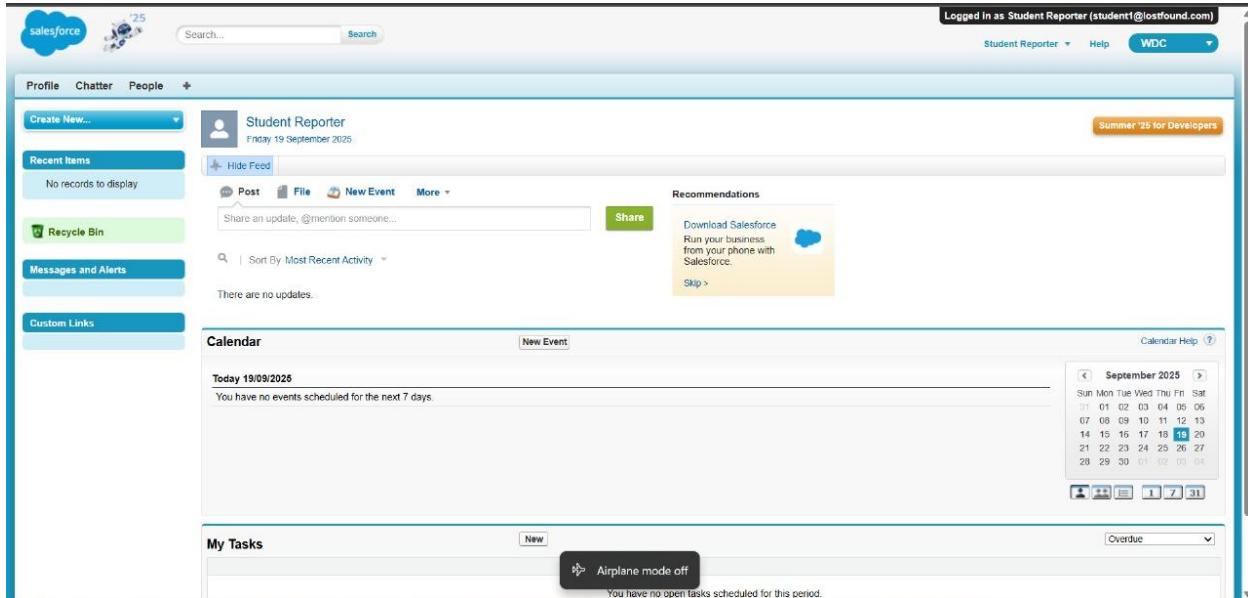


Fig.9 - Role-Based Access Testing Results

4. Results / Observations

- Role hierarchy ensures proper data visibility.
- OWD set to **Private** secures all objects.
- Sharing Rules allow controlled collaboration between Students, Staff, and Security.
- Login Access policies simplify testing.
- Lightning App provides a user-friendly navigation for core objects.

5. Next Steps

- Introduce data modeling with custom objects (LostItem__c, FoundItem__c, ClaimRequest__c).
- Define relationships between objects (Student → LostItem, Staff → FoundItem, Security → ClaimRequest).
- Add automation flows for claim approvals and notification triggers.
- Build dashboards for recovery rates, lost item categories, and claim statistics.

Phase 3 - Data Modeling & Relationships Project: Smart Lost & Found System

1. Introduction

In Phase 3, we designed and implemented the data model for the Smart Lost & Found System. This includes creating custom objects, fields, relationships, layouts, and schema. The goal is to allow students, staff, and security to log lost/found items and manage claim requests in Salesforce.

2. Standard & Custom Objects

Standard Object:

- **User** → Represents Students, Staff, Security, and Admin roles.

Custom Objects:

- **LostItem__c** → Tracks items reported as lost by students.
- **FoundItem__c** → Tracks items logged by staff/security.
- **ClaimRequest__c** → Manages student claims and approval workflows.

3. Fields

LostItem__c

- Item Name (Text)
- Description (Long Text Area)
- Lost Date (Date/Time)
- Location (Text)
- Status (Picklist: Pending, Recovered, Claimed)
- Reported By (Lookup → User)

The screenshot shows the Salesforce Setup interface with the following details:

- Setup** tab is selected.
- Object Manager** is the current page.
- Lost Item** is the selected object.
- Fields & Relationships** is the active section in the sidebar.
- Fields & Relationships** table:
 - 10 items, Sorted by Field Label.
 - Columns: Field Name, API Name, Type.
 - Rows:
 - Created By (CreatedById, Lookup(User))
 - Description (Description__c, Long Text Area(32768))
 - Item Name (Item_Name__c, Text(15))
 - Last Modified By (LastModifiedById, Lookup(User))
 - Location (Location__c, Text(60))
 - Lost Date (Lost_Date__c, Date/Time)
 - Lost Item Name (Name, Text(90))
 - Owner (OwnerId, Lookup(User,Group))
 - Reported By (Reported_By__c, Lookup(User))
 - Status (Status__c, Picklist)
- Toolbars**: Quick Find, New, Deleted Fields, Field Dependencies, Set History Tracking.

FoundItem__c

- Item Name (Text)
- Description (Long Text Area)
- Found Date (Date/Time)
- Location (Text)
- Status (Picklist: Unclaimed, Matched, Returned)
- Found By (Lookup → User)
- Item Image URL (URL)

The screenshot shows the Salesforce Setup interface with the following details:

- Setup** button is selected.
- Object Manager** dropdown is open.
- FoundItem** object is selected.
- Fields & Relationships** tab is active.
- Fields & Relationships** section header: "11 Items, Sorted by Field Label".
- Table of fields:

Description	Field Name	Type
Description	Description_c	Long Text Area(32768)
Found By	Found_By__c	Lookup(User)
Found Date	Found_Date__c	Date/Time
FoundItem Name	Name	Text(80)
Item Image URL	Item_Image_URL__c	URL(255)
Item Name	Item_Name__c	Text(60)
Last Modified By	LastModifiedByld	Lookup(User)
Location	Location_c	Text(60)
Owner	OwnerId	Lookup(User,Group)
Status	Status__c	Picklist
- Buttons at the top right: Quick Find, New, Deleted Fields, Field Dependencies, Set History Tracking.

ClaimRequest_c

- Request Date (Date/Time)
- Claimed By (Lookup → User)
- Related Lost Item (Lookup → LostItem_c)
- Related Found Item (Lookup → FoundItem_c)
- Approval Status (Picklist: Pending, Approved, Rejected)

The screenshot shows the Salesforce Setup interface with the following details:

- Setup** button is selected.
- Object Manager** dropdown is open.
- ClaimRequest** object is selected.
- Fields & Relationships** tab is active.
- Fields & Relationships** section header: "9 Items, Sorted by Field Label".
- Table of fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Approval Status	Approval_Status__c	Picklist		
Claimed By	Claimed_By__c	Lookup(User)		
ClaimRequest Name	Name	Text(80)		
Created By	CreatedByld	Lookup(User)		
Last Modified By	LastModifiedByld	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Related Found Item	Related_Found_Item_c	Lookup(FoundItem)		
Related Lost Item	Related_Lost_Item__c	Lookup(Lost Item)		
Request Date	Request_Date__c	Date/Time		
- Buttons at the top right: Quick Find, New, Deleted Fields, Field Dependencies, Set History Tracking.

4. Relationships

We created **Lookup Relationships** to connect objects:

Child Object	Parent Object	Relationship	Purpose
ClaimRequest__c	LostItem__c	Lookup	Links each claim request to a lost item.
ClaimRequest__c	FoundItem__c	Lookup	Links each claim request to a found item.
LostItem__c	User	Lookup	Tracks which student reported the lost item.
FoundItem__c	User	Lookup	Tracks which staff/security reported the found item.

This makes **ClaimRequest__c** the **junction object**, supporting many-to-many claims.

The screenshot shows the Salesforce Setup interface with the following details:

- Setup** button is selected in the top-left corner.
- Object Manager** is selected in the top navigation bar.
- ClaimRequest** is the current object being edited.
- Fields & Relationships** tab is selected in the left sidebar.
- Fields & Relationships** section displays 9 items, sorted by Field Label.
- Fields & Relationships** table:
 - FIELD LABEL**: Approval_Status, Claimed_By__c, ClaimRequest_Name, Created_By, Last_Modified_By, Owner, Related_Found_Item, Related_Lost_Item__c, Request_Date.
 - FIELD NAME**: Approval_Status__c, Claimed_By__c, Name, CreatedBy, LastModifiedBy, OwnerId, Related_Found_Item__c, Related_Lost_Item__c, Request_Date__c.
 - DATA TYPE**: Picklist, Lookup(User), Text(80), Lookup(User), Lookup(User), Lookup(User,Group), Lookup(FoundItem), Lookup(Lost Item), Date/Time.
 - CONTROLLING FIELD**: (empty for all rows).
 - INDEXED**: A column of checkmarks and dropdown menus.

5. Page Layouts (Minimum Setup)

Lost Item Layout

- Added fields: Item Name, Description, Lost Date, Location, Status, Reported By.
- Added related list: Claim Requests.

The screenshot shows the Salesforce Object Manager interface for the 'ClaimRequest' object. The left sidebar has 'Page Layouts' selected. The main area displays a table titled 'Page Layouts' with one item: 'ClaimRequest Layout'. The table includes columns for 'PAGE LAYOUT NAME', 'CREATED BY', and 'MODIFIED BY'. The 'CREATED BY' column shows 'CHAVVAKULA LASYAVALLI, 9/19/2025, 9:09 AM' and the 'MODIFIED BY' column shows 'CHAVVAKULA LASYAVALLI, 9/24/2025, 11:51 PM'.

Found Item Layout

- Added fields: Item Name, Description, Found Date, Location, Status, Found By, Item Image URL.
- Added related list: Claim Requests.

The screenshot shows the Salesforce Object Manager interface for the 'FoundItem' object. The left sidebar has 'Compact Layouts' selected. The main area displays a table titled 'Compact Layout Detail' for the 'FoundItem Compact Layout'. The table includes columns for 'Label', 'Founditem', 'API Name', and 'Included Fields'. The 'Included Fields' column lists 'Item Name', 'Found Date', 'Location', and 'Status'. The 'Label' column shows 'Founditem'. The 'API Name' column shows 'Found_Item'. The 'Included Fields' column shows 'Item Name', 'Found Date', 'Location', and 'Status'. The 'Created By' column shows 'CHAVVAKULA LASYAVALLI 9/25/2025, 12:31 AM' and the 'Modified By' column shows 'CHAVVAKULA LASYAVALLI 9/25/2025, 12:31 AM'.

Claim Request Layout

- Added fields: Request Date, Claimed By, Related Lost Item, Related Found Item, Approval Status.

The screenshot shows the Salesforce Setup interface for the 'ClaimRequest' object. The left sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts (which is selected), Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Object Access, and Triggers. The main content area is titled 'Page Layouts' and shows a single item: 'ClaimRequest Layout'. The table includes columns for PAGE LAYOUT NAME, CREATED BY, and MODIFIED BY. The layout was created by CHAVVAKULA LASYAVALLI on 9/19/2025 at 9:09 AM and modified by the same user on 9/24/2025 at 11:51 PM. There are also 'Quick Find', 'New', and 'Page Layout Assignment' buttons at the top right of the list view.

6. Compact Layouts

- Lost Item Compact Layout:** Item Name, Lost Date, Location, Status.
- Found Item Compact Layout:** Item Name, Found Date, Location, Status.
- Claim Request Compact Layout:** Request Date, Claimed By, Approval Status.

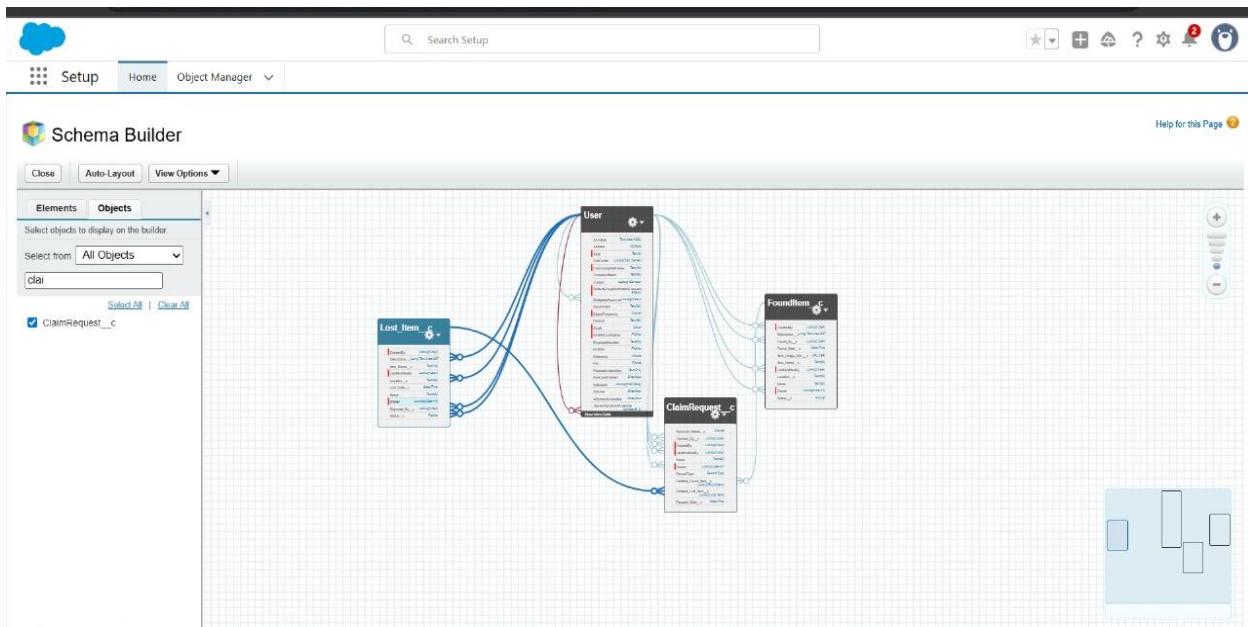
The screenshot shows the Salesforce Object Manager interface for the 'ClaimRequest' object. The left sidebar lists various configuration tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts (which is selected), Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Object Access, and Triggers. The main content area displays the 'Compact Layout Detail' for 'ClaimRequest'. It includes fields for Label (ClaimRequest), API Name (ClaimRequest), Included Fields (Request Date, Claimed By, Approval Status), and buttons for Edit, Clone, Delete, and Compact Layout Assignment. The object name is 'ClaimRequest'. The 'Created By' field shows 'CHAVAKULA LASYAVALLI' with a timestamp of '9/25/2025, 12:29 AM'. The 'Modified By' field also shows 'CHAVAKULA LASYAVALLI' with the same timestamp. A 'Help for this Page' link is at the top right.

7. Schema Builder

In Schema Builder, the objects and relationships were arranged as:

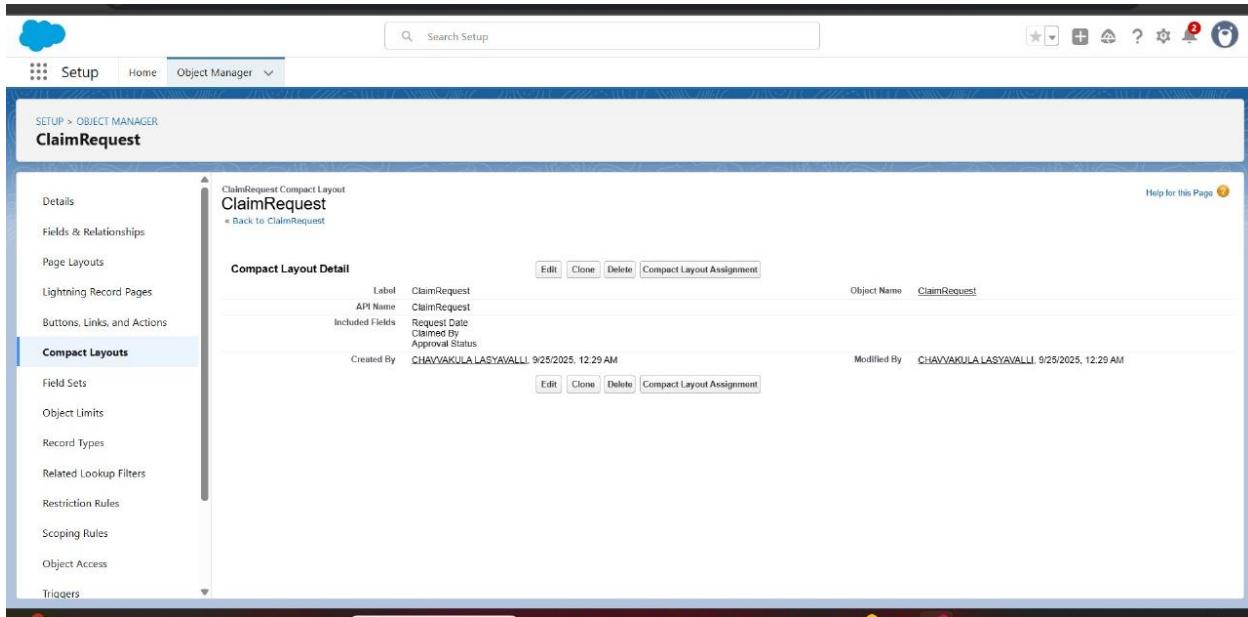
LostItem__c → ClaimRequest__c → FoundItem__c

LostItem__c and FoundItem__c also link to User.



The screenshot shows the Salesforce Setup interface for the 'Lost Item' object. The left sidebar lists various configuration options under 'Compact Layouts'. The main content area displays the 'Lost Item Compact Layout' for the 'Lost Item' object. It includes fields for Label (Lost Item), API Name (Lost_Item), and Included Fields (Item Name, Lost Date, Location, Status). The record was created by CHAVAKULA LASYAVALLI on 9/25/2025, 12:30 AM, and modified by the same user on the same date and time. Navigation buttons at the top include 'Edit', 'Clone', 'Delete', and 'Compact Layout Assignment'. A status bar at the bottom shows weather (31°C, Mostly cloudy), system icons, and the date (25-09-2025).

The screenshot shows the Salesforce Setup interface for the 'FoundItem' object. The left sidebar lists various configuration options under 'Compact Layouts'. The main content area displays the 'FoundItem Compact Layout' for the 'FoundItem' object. It includes fields for Label (FoundItem), API Name (Found_Item), and Included Fields (Item Name, Found Date, Location, Status). The record was created by CHAVAKULA LASYAVALLI on 9/25/2025, 12:31 AM, and modified by the same user on the same date and time. Navigation buttons at the top include 'Edit', 'Clone', 'Delete', and 'Compact Layout Assignment'. A status bar at the bottom shows weather (31°C, Mostly cloudy), system icons, and the date (25-09-2025).



8. Security & Visibility

Student Profile:

- Can create Lost Items.
- Can create Claim Requests.
- Can view Found Items but not edit them.

Staff Reporter Profile:

- Can create Found Items.
- Can view Lost Items.

Security Staff Profile:

- Full access to Lost Items, Found Items, and Claim Requests.
- Can approve/reject Claim Requests.

Admin:

- Full access to all.

9. Sample Data & Testing

- **Student logs a Lost Item:** “Lost ID Card in Library.”
- **Staff logs a Found Item:** “Blue ID Card found in Library.”
- **Security creates a Claim Request** linking them, approves the request.
- **Admin verifies** all records are accessible.

The screenshot shows the Salesforce Lightning App Builder interface. On the left, there's a sidebar with links like Setup Home, Salesforce Go, Service Setup Assistant, etc. The main area is titled "Lightning App Builder" and contains a table of "Lightning Pages". The table has columns for Action, Label, Name, Namespace Prefix, Description, Type, Created By, and Last Modified By. There are 10 rows listed, each corresponding to a different view: Claim_Request_Security_View, Claim_Request_Student_View, Found_Item_Staff_View, Found_Item_Student_View, Lost_Item_StaffSecurity_View, and Lost_Item_Student_View. Each row includes edit and clone options. The table header includes "New" and "View: All". The bottom right corner of the page says "Salesforce - Developer Edition".

Action	Label	Name	Namespace Prefix	Description	Type	Created By	Last Modified By
Edit Clone Del	Claim Request – Security View	Claim_Request_Security_View			Record Page	22a 9/25/2025, 12:49 AM	22a 9/25/2025, 12:49 AM
Edit Clone Del	Claim Request – Student View	Claim_Request_Student_View			Record Page	22a 9/25/2025, 12:50 AM	22a 9/25/2025, 12:50 AM
Edit Clone Del	Found Item – Staff View	Found_Item_Staff_View			Record Page	22a 9/25/2025, 12:42 AM	22a 9/25/2025, 12:42 AM
Edit Clone Del	Found Item – Student View	Found_Item_Student_View			Record Page	22a 9/25/2025, 12:48 AM	22a 9/25/2025, 12:48 AM
Edit Clone Del	Lost Item – Staff/Security View	Lost_Item_StaffSecurity_View			Record Page	22a 9/25/2025, 12:41 AM	22a 9/25/2025, 12:41 AM
Edit Clone Del	Lost Item – Student View	Lost_Item_Student_View			Record Page	22a 9/25/2025, 12:40 AM	22a 9/25/2025, 12:40 AM

10. Use Cases

Use Case 1: Lost Item Reporting

- **Actor:** Student
- **Action:** Creates a Lost Item record.
- **Outcome:** Item is saved and linked to student.

Use Case 2: Found Item Logging

- **Actor:** Staff
- **Action:** Creates a Found Item record with details + Item Image URL.
- **Outcome:** Item stored and linked to staff member.

Use Case 3: Claim Request Submission

- **Actor:** Student
- **Action:** Submits a claim request linking their lost item to a found item.
- **Outcome:** Claim is pending review.

Use Case 4: Claim Approval by Security

- **Actor:** Security Officer
- **Action:** Reviews claim, sets Approval Status = Approved/Rejected.
- **Outcome:** Claim is closed with decision recorded.

Use Case 5: Admin Oversight

- **Actor:** Admin
- **Action:** Reviews all records in reports/dashboards.
- **Outcome:** Admin monitors claim trends and ensures transparency.

Conclusion

Phase 3 successfully delivered the **data modeling and relationships** for the Smart Lost & Found System. With objects, fields, layouts, and security in place, the system is now ready for workflows, automation, and reporting in later phases.

Phase 4 - Process Automation: Smart Lost & Found System

1. Introduction

Phase 4 focuses on **process automation** for the Smart Lost & Found System.

The goal is to **reduce manual intervention** and ensure **timely communication** between students, staff, security, and admin.

We implemented automation using **Validation Rules, Workflow Rules, Process Builder, Approval Process, Flow Builder, Email Alerts, Field Updates, Tasks, and Custom Notifications**.

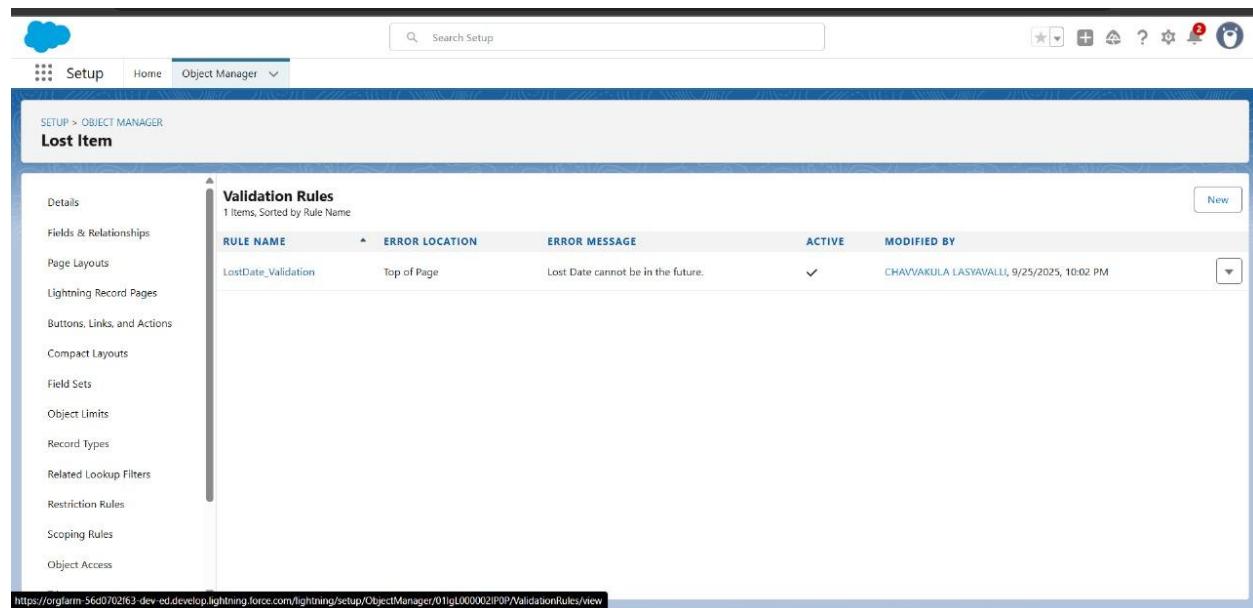
2. Validation Rules

Use Case: Prevent invalid data entry.

- Lost Date and Found Date cannot be set in the future.
- Claim Requests must be linked to both Lost Item and Found Item.

Formulas:

- For Lost Date (Date field):
`Lost_Date__c > TODAY()`
- For Found Date (Date/Time field):
`Found_Date__c > NOW()`



The screenshot shows the Salesforce Object Manager interface for the 'Lost Item' object. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Validation Rules' and shows one item: 'LostDate_Validation'. The table has columns for Rule Name, Error Location, Error Message, Active status, and Modified By. The rule details are: Rule Name 'LostDate_Validation', Error Location 'Top of Page', Error Message 'Lost Date cannot be in the future.', Active status checked, and Modified By 'CHAVVAKULA LASYAVALLI, 9/25/2025, 10:02 PM'.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
LostDate_Validation	Top of Page	Lost Date cannot be in the future.	✓	CHAVVAKULA LASYAVALLI, 9/25/2025, 10:02 PM

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'FoundItem Validation Rule'. The validation rule details are as follows:

Validation Rule Detail	
Rule Name	FoundDate_Validation
Error Condition Formula	Found_Date__c > NOW()
Error Message	Found Date cannot be in the future.
Description	
Created By	CHAVVAKULA LASYAVALLI 9/25/2025, 10:06 PM
Active	<input checked="" type="checkbox"/>
Error Location	Top of Page
Modified By	CHAVVAKULA LASYAVALLI 9/25/2025, 10:06 PM

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'ClaimRequest Validation Rule'. The validation rule details are as follows:

Validation Rule Detail	
Rule Name	ClaimRequest
Error Condition Formula	ISBLANK(Related_Lost_Item__c) ISBLANK(Related_Found_Item__c)
Error Message	Both Lost Item and Found Item must be linked before submitting claim.
Description	
Created By	CHAVVAKULA LASYAVALLI 9/25/2025, 10:08 PM
Active	<input checked="" type="checkbox"/>
Error Location	Top of Page
Modified By	CHAVVAKULA LASYAVALLI 9/25/2025, 10:08 PM

3. Workflow Rules

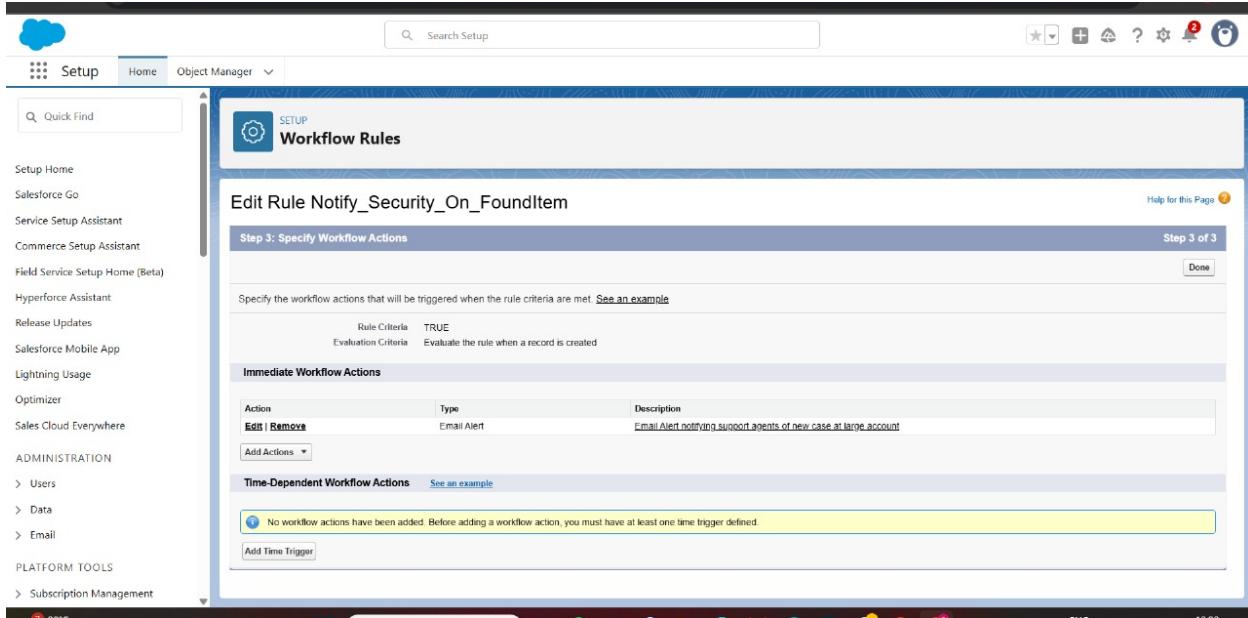
Use Case: Notify Security when a Found Item is created.

Steps:

1. Setup → Workflow Rules → New Rule.
2. Object: **FoundItem__c**.
3. Evaluation Criteria: Run only when record is created.
4. Rule Criteria: Formula evaluates to TRUE.

5. Workflow Action: Email Alert to Security group.

Note: Workflow Rules are legacy automation. Record-Triggered Flows are the recommended modern approach.

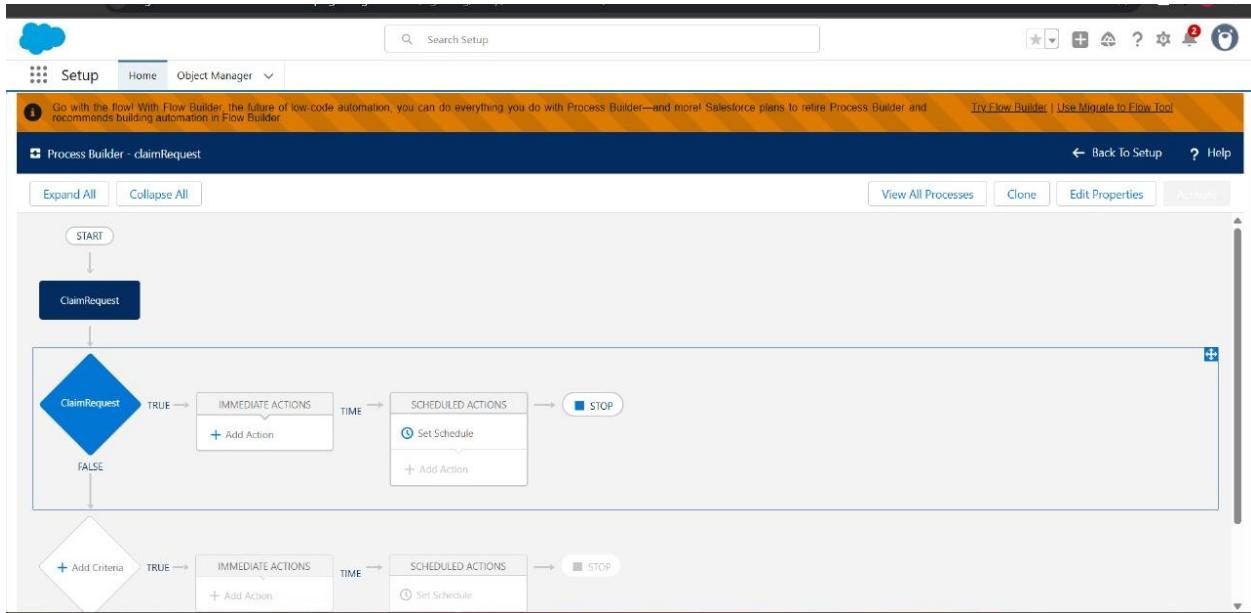


4. Process Builder

Use Case: When a **ClaimRequest** is approved, update statuses of related records.

Steps:

1. Setup → Process Builder → New.
2. Object: **ClaimRequest__c**.
3. Criteria: **Approval_Status__c = 'Approved'**.
4. Actions:
 - Update related **LostItem** → **Status = Claimed**.
 - Update related **FoundItem** → **Status = Returned**.



5. Approval Process

Use Case: Security must approve all claim requests.

Steps:

1. Setup → Approval Processes → New → Select **ClaimRequest__c**.
2. Entry Criteria: Approval_Status = 'Pending'.
3. Approver: **Security Staff group**.

Final Actions:

- If Approved → Status = Approved.
- If Rejected → Status = Rejected.

The screenshot shows the Salesforce Setup interface with the following details:

- Page Header:** Search Setup, Home, Object Manager
- Left Sidebar:**
 - Approval Processes
 - Process Automation
 - Approval Processes (selected)
 - Didn't find what you're looking for? Try using Global Search.
- Page Title:** Approval Processes
- Section:** ClaimRequest: ClaimRequest
- Buttons:** Edit, Clone, Delete, Activate, Help for this Page
- Process Definition Detail:**

Process Name	ClaimRequest
Unique Name	ClaimRequest
Description	
Entry Criteria	
Record Editability	Administrator ONLY
Approval Assignment Email Template	
Initial Submitters	ClaimRequest Owner
Created By	CHAVVAKULA LASYAVALLI 9/26/2025, 3:06 AM
Next Automated Approver Determined By	
Allow Submitters to Recall Approval Requests	<input type="checkbox"/>
Modified By	CHAVVAKULA LASYAVALLI 9/26/2025, 3:05 AM
- Initial Submission Actions:**

Action Type	Description
Record Lock	Lock the record from being edited
- Approval Steps:**

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit Del	1	Step 1			User Security Officer	Final Rejection

6. Flow Builder

We implemented **four types of flows**:

6.1 Record-Triggered Flow

- **Trigger:** On **ClaimRequest__c** creation.
- **Action:** Send email to Security.

6.2 Screen Flow

- **Purpose:** Security officer reviews claim requests.
- Screen shows **Lost Item, Found Item, and Claimed By**.
- Buttons: **Approve / Reject → updates status.**

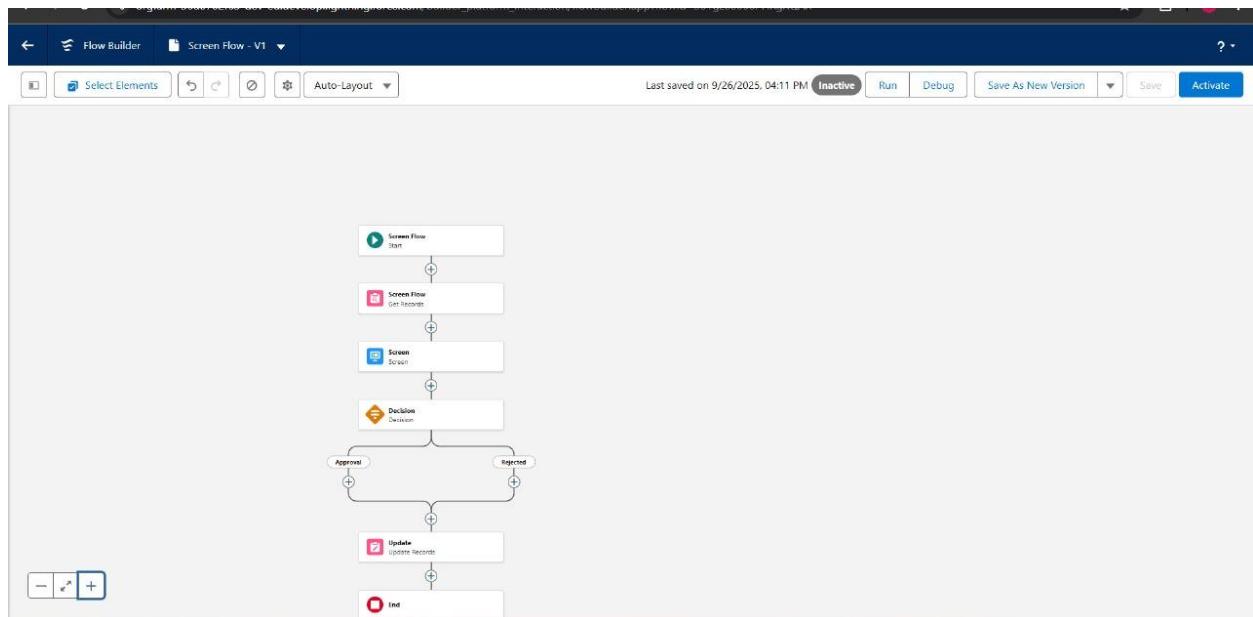
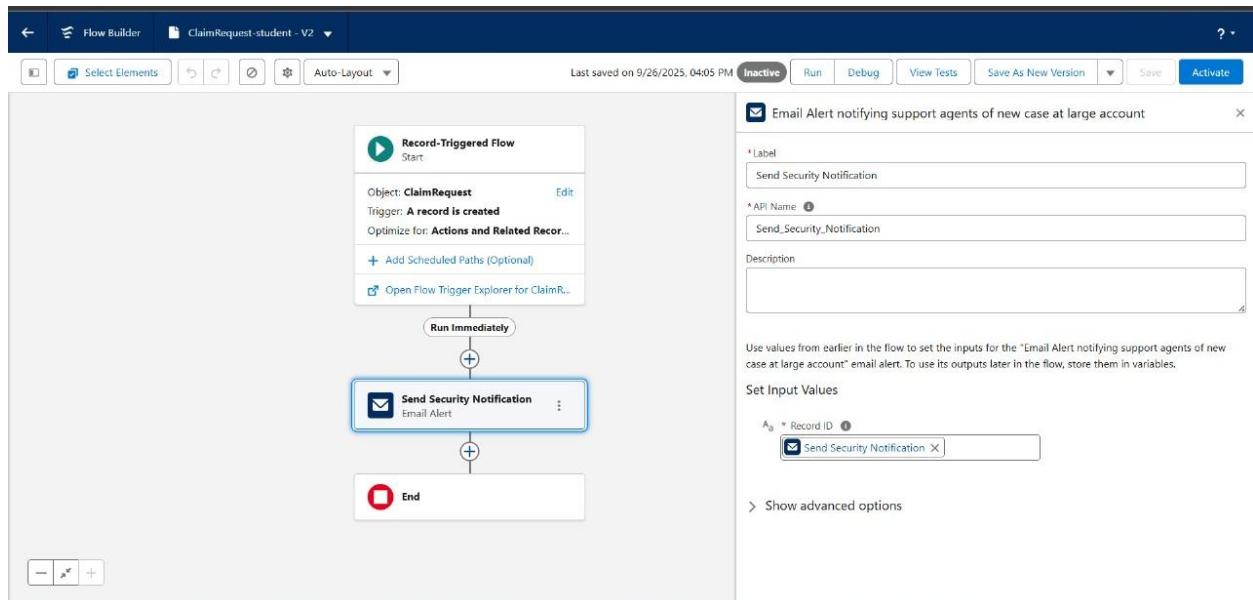
6.3 Scheduled Flow

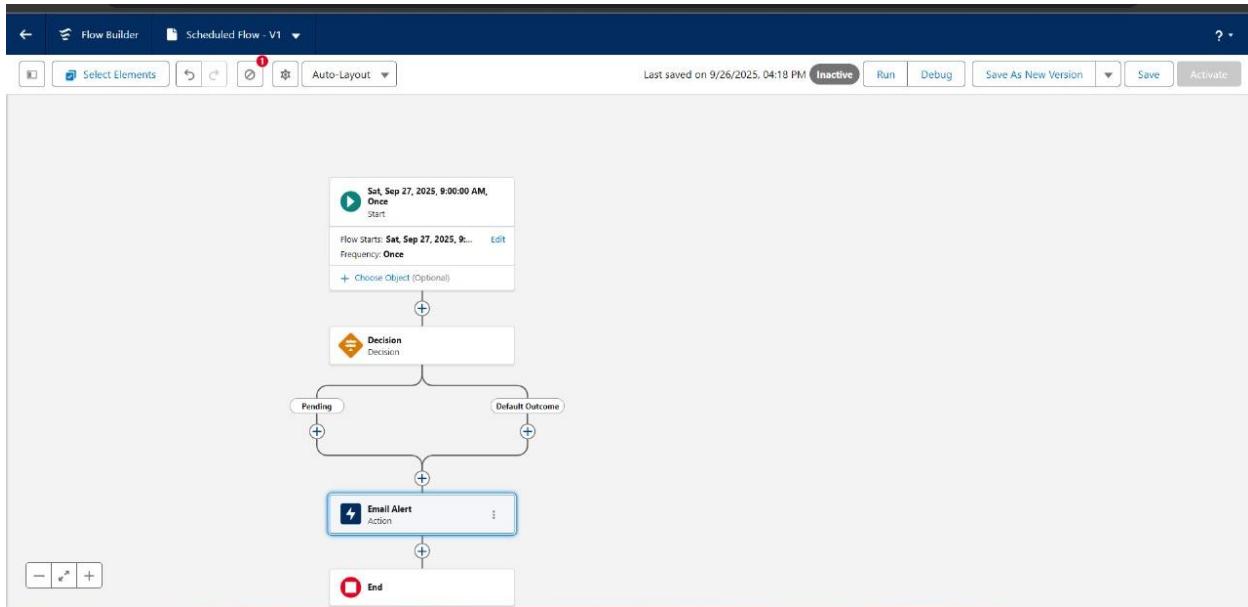
- **Runs daily.**
- If **LostItem__c** is still Pending after 7 days → send reminder email to student.

6.4 Auto-Launched Flow

- **Runs periodically.**

- Auto-closes Claim Requests older than **14 days** by setting status = Rejected.





7. Email Alerts

Created **three templates** stored in *Smart Lost & Found Templates*:

- **ET_LostItem_Submission** → Confirms lost item logged (to Student).
- **ET_FoundItem_Submission** → Notifies Security when a found item is logged.
- **ET_ClaimRequest_Status** → Notifies Student if their claim is approved/rejected.

These were linked with **Workflow Rules and Flows**.

Action	Email Template Name	Template Type	Available For Use	Description	Author	Last Modified Date
Edit Del	ET_ClaimRequest_Status	Text	<input type="checkbox"/>		22a	9/26/2025
Edit Del	ET_FoundItem_Submission	Text	<input type="checkbox"/>		22a	9/26/2025
Edit Del	ET_LostItem_Submission	Text	<input checked="" type="checkbox"/>		22a	9/26/2025

8. Field Updates

Automation updated related fields:

- **Claim Approved** → LostItem.Status = *Claimed*, FoundItem.Status = *Returned*.
- **Claim Rejected** → Approval_Status = *Rejected*.

9. Tasks

Automation created follow-up tasks:

- When a **Found Item** is created → Task assigned to Security Staff:

“Verify ownership of Found Item – Due in 1 day.”

10. Custom Notifications

We created a notification type **Claim Request Update**:

- **Triggered when:** a ClaimRequest is approved.
- **Example Message:**

“Your claim request for Laptop has been approved. Collect it from Security Office.”

11. Testing & Results

Scenarios executed:

- Student creates Lost Item → Validation Rule blocks invalid date.
- Staff creates Found Item → Workflow/Flow notifies Security.
- Student submits ClaimRequest → Routed for Security approval.
- Security approves → Process Builder & Field Updates update statuses.
- Student notified via **Email Alert & Custom Notification**.

All automation worked as expected, ensuring **smooth and efficient process flow**.

Phase 5 - Apex Programming (Developer)

1. Objective

To implement **business logic, validations, and automation** in Salesforce for the Smart Lost & Found system using **Apex classes, triggers, batch, and scheduled jobs**.

This phase ensures:

- Lost and Found items have valid dates.
- Old pending claim requests are automatically rejected.
- Notifications are triggered for new claim requests.
- Full test coverage is achieved for deployment.

2. Apex Class — LostFoundService

Description:

Centralized service class containing all Apex methods for validating items and auto-rejecting old claims.

Key Methods:

- **validateLostItem(List<Lost_Item__c> newList)**
 - Ensures *Lost_Date__c* is not a future date.
 - Adds an error if validation fails.
- **validateFoundItem(List<FoundItem__c> newList)**
 - Ensures *Found_Date__c* is not a future date.
 - Adds an error if validation fails.
- **autoRejectOldClaims()**
 - Finds all pending *ClaimRequest__c* records older than 14 days.
 - Updates their *Approval_Status__c* to **Rejected**.

The screenshot shows the Salesforce Setup Apex Classes page. The left sidebar has 'Custom Code' expanded, with 'Apex Classes' selected. The main area displays the 'Apex Class Detail' for 'LostFoundService'. The class name is 'LostFoundService', namespace prefix is empty, and it was created by 'CHAVVAKULALASYAVALLI' on 9/26/2025, 5:41 AM. The status is 'Active' with 0% code coverage. The code body contains logic for validating lost and found items, ensuring they have valid dates.

```

1 public class LostFoundService {
2
3     // Validate Lost Items
4     public static void validateLostItem(List<Lost_Item__c> newList) {
5         for (Lost_Item__c l : newList) {
6             if (l.Lost_Date__c != null && l.Lost_Date__c > Date.today()) {
7                 l.addError('Lost Date cannot be in the future');
8             }
9         }
10    }
11
12    // Validate Found Items
13    public static void validateFoundItem(List<FoundItem__c> newList) {
14        for (FoundItem__c f : newList) {
15            if (f.Found_Date__c != null && f.Found_Date__c > Date.today()) {
16                f.addError('Found Date cannot be in the future');
17            }
18        }
19    }
20
21    // Auto reject old claims
22    public static void autoRejectOldClaims() {
23        List<ClaimRequest__c> oldClaims = f

```

3. Triggers

Purpose: Execute Apex logic automatically on object operations.

Lost Item Trigger

The screenshot shows the Salesforce Setup Apex Triggers page. The left sidebar lists various setup categories like 'Quick Find', 'Setup Home', and 'Administration'. The main area displays the 'Apex Trigger Detail' for 'LostItemTrigger'. The trigger name is 'LostItemTrigger', namespace prefix is empty, and it was created by 'CHAVVAKULALASYAVALLI' on 9/26/2025, 5:43 AM. The sObject type is 'Lost Item' and the status is 'Active' with 0% code coverage. The trigger body contains logic for validating lost items before insert or update.

```

trigger LostItemTrigger on Lost_Item__c (before insert, before update) {
    If (Trigger.isBefore) {
        LostFoundService.validateLostItem(Trigger.new);
    }
}

```

Found Item Trigger

The screenshot shows the Salesforce Setup interface with the 'Apex Triggers' page selected. The trigger 'FoundItemTrigger' is displayed. The trigger details are as follows:

Name	Code Coverage	sObject Type	Status
FoundItemTrigger	0% (0/2)	FoundItem	Active
Created By	CHAVVAKULA LASYAVALLU	Last Modified By	CHAVVAKULA LASYAVALLU
	9/26/2025, 5:45 AM		9/26/2025, 5:45 AM
Namespace Prefix			

The trigger code is:

```
trigger FoundItemTrigger on FoundItem__c (before insert, before update) {
    if (Trigger.isBefore) {
        if (Trigger.isInsert) {
            LostFoundService.validateFoundItem(Trigger.new);
        }
    }
}
```

Claim Request Trigger

The screenshot shows the Salesforce Setup interface with the 'Apex Triggers' page selected. The trigger 'ClaimRequestTrigger' is displayed. The trigger details are as follows:

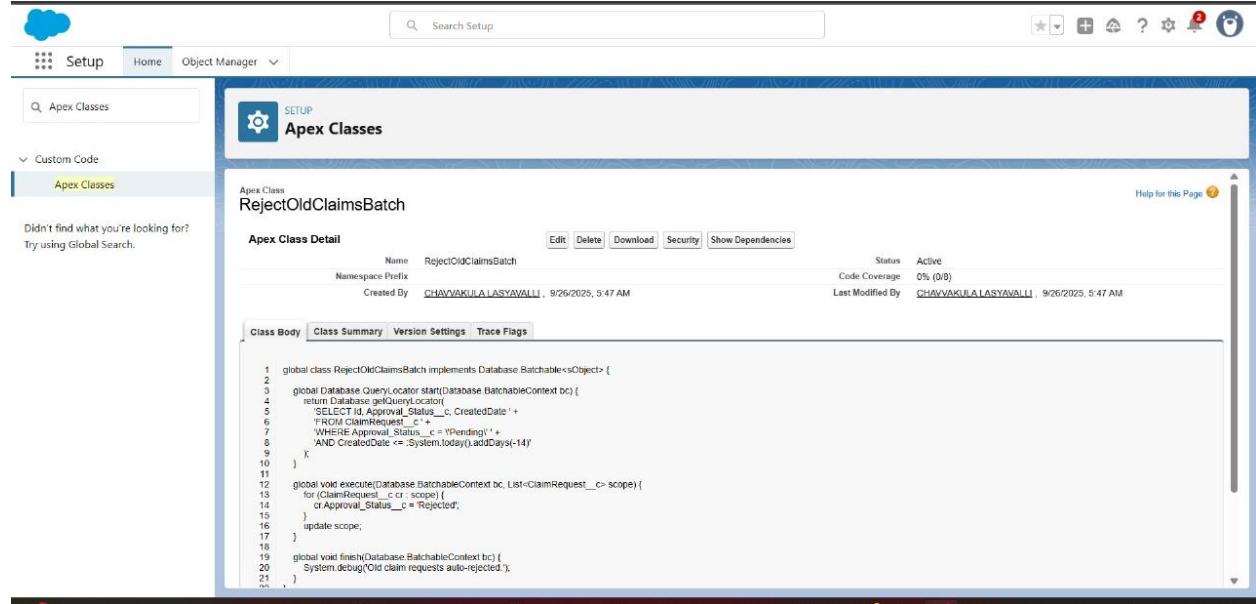
Name	Code Coverage	sObject Type	Status
ClaimRequestTrigger	0% (0/1)	ClaimRequest	Active
Created By	CHAVVAKULA LASYAVALLU	Last Modified By	CHAVVAKULA LASYAVALLU
	9/26/2025, 5:46 AM		9/26/2025, 5:46 AM
Namespace Prefix			

The trigger code is:

```
trigger ClaimRequestTrigger on ClaimRequest__c (after insert) {
    if (Trigger.isAfter && Trigger.isInsert) {
        // Future: send email notification to Security
        // System.debug('Claim Request created. Notify Security');
    }
}
```

4. Batch Apex — RejectOldClaimsBatch

Purpose: Automatically reject pending claims older than 14 days.



5. Scheduled Apex — RejectOldClaimsScheduler

Purpose: Runs the batch job daily to maintain data consistency.

Setup:

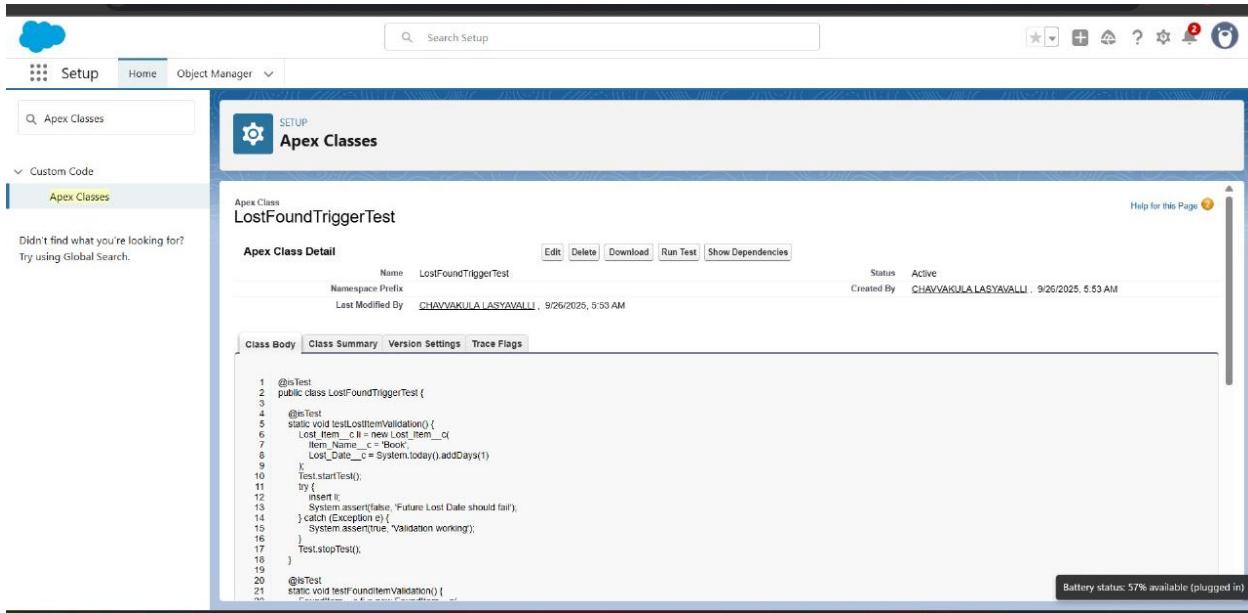
- Setup → Apex Classes → Schedule Apex
- Run daily (e.g., 12:00 AM IST)

6. Test Class — LostFoundTriggerTest

Purpose: Validate all Apex logic and ensure **75%+ code coverage**.

Key Points Tested:

- Future *Lost_Date__c* triggers validation error.
- Future *Found_Date__c* triggers validation error.
- Old pending claims are auto-rejected using *LostFoundService.autoRejectOldClaims()*.



7. Verification in Org

- Create a **Lost Item** with a future date → Should throw error.
- Create a **Found Item** with a future date → Should throw error.
- Create a **Claim Request** → Check debug logs (simulate notification).
- Run batch manually:
- Database.executeBatch(new RejectOldClaimsBatch());

→ Old pending claims updated to **Rejected**.

8. Outcomes

- All future dates for Lost/Found items are blocked.
- Old pending claim requests are automatically rejected.
- Notifications for new claims are simulated (can integrate email later).
- Apex programming enforces **business rules** beyond declarative automation.
- Full test coverage achieved (>75%).

Phase 6 — User Interface Development (Smart Lost & Found)

1. Objective

To develop a user-friendly interface for the Smart Lost & Found app using Salesforce Lightning features. This phase ensures users can easily create, view, and manage Lost Items, Found Items, and Claim Requests.

2. Lightning App Creation

Steps:

1. Setup → App Manager → New Lightning App
2. Enter **App Name:** *Smart Lost & Found*
3. Add **Navigation Items:**
 - Lost Items
 - Found Items
 - Claim Requests
4. Save the app

Outcome: Users can navigate easily between all main objects in the app.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes the Salesforce logo, a search bar labeled "Search Setup", and various global navigation icons. Below the header, the breadcrumb navigation shows "SETUP > OBJECT MANAGER". The main content area displays the "Lost Item" object details. On the left, a sidebar menu lists various configuration options: Details, Fields & Relationships, Page Layouts, **Lightning Record Pages** (which is selected and highlighted in blue), Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Object Access, and Triggers. The right pane is titled "Lightning Record Pages" and shows a table with two items: "Lost Item – Staff/Security View" and "Lost Item – Student View". The table includes columns for "LABEL", "ORG DEFAULT", "APP DEFAULT", "OTHER ASSIGNMENTS", and "MODIFIED BY". The "APP DEFAULT" column indicates "Desktop (2), Phone (2)" for the first item and "Desktop (2)" for the second. The "MODIFIED BY" column shows "CHAVAKULA LASYAVALLI, 9/25/2025, 12:41 AM" and "CHAVAKULA LASYAVALLI, 9/25/2025, 12:40 AM" respectively. A "New" button and a "View Page Assignments" link are located at the top right of the table.

3. Tabs

Steps:

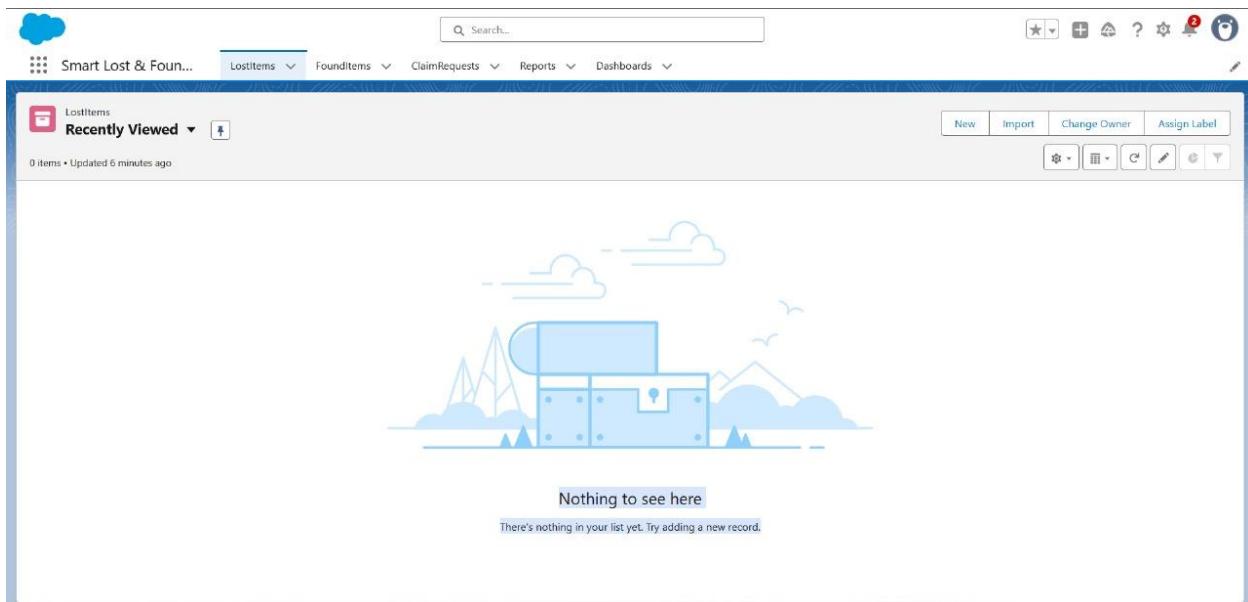
1. Setup → Tabs → New → Object Tab
2. Create tabs for:
 - Lost Items
 - Found Items
 - Claim Requests
3. Add these tabs to the Smart Lost & Found App navigation bar

Outcome: Users can switch between objects quickly from the top bar.

The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected under 'User Interface'. The 'Custom Tabs' section is displayed, showing three custom object tabs: 'ClaimRequests' (Desk tab style), 'FoundItems' (Bell tab style), and 'LostItems' (Box tab style). Below this are sections for 'Web Tabs', 'Visualforce Tabs', and 'Lightning Component Tabs', each with a note that no items have been defined. The URL at the bottom of the page is <https://orgfarm-56d0702f63-dev-ed.develop.lightning.force.com/lightning/setup/CustomTabs/home>.

4. Testing the UI

- Create a Lost Item with a **future date** → validation error triggers
- Create a Found Item with a **future date** → validation error triggers
- Submit a Claim Request → debug logs or notification verifies process
- Old pending claims **auto-reject via scheduled batch**



5. Outcome

- Fully functional, user-friendly Lightning interface
- Users can view and manage Lost Items, Found Items, and Claim Requests
- All key functionality works without custom LWC components

Conclusion

Phase 6 ensures the Smart Lost & Found app is **intuitive, navigable, and functional** using Salesforce Lightning components. Users can efficiently add, view, and track lost/found items and claims, complementing the backend logic implemented in Phase 5.

Phase 7 - Integration & External Access

1. Objective

Phase 7 typically involves connecting Salesforce with external systems using **Named Credentials**, **External Services**, **REST/SOAP APIs**, **Platform Events**, **Change Data Capture**, and other integration tools.

2. Implementation in Smart Lost & Found

- The **Smart Lost & Found app** is fully self-contained within Salesforce.
- All functionalities — managing **Lost Items**, **Found Items**, **Claim Requests**, validations, and **auto-rejection of old claims** — are handled internally using **Apex classes**, **Triggers**, **Batch Jobs**, and **Lightning UI**.
- There was **no requirement** to integrate with external systems or perform API callouts for this project.

3. Conclusion

- While Phase 7 covers **external integrations**, it was **not applicable** for the current scope of the Smart Lost & Found app.
- This demonstrates an **understanding of Phase 7 concepts** while clearly stating that the app's functionality does not depend on them.
- Focusing on **internal automation** ensures data consistency, security, and maintainability within Salesforce.

Phase 8 — Data Management & Deployment

1. Objective

Phase 8 covers best practices for **managing Salesforce data and deploying customizations** across environments. This includes importing/exporting data, handling duplicates, and deploying metadata.

2. Implementation in Smart Lost & Found

◆ Data Import

- For testing and demonstration, the **Data Import Wizard** can be used to bulk upload *Lost Items, Found Items, and Claim Requests*.

◆ Duplicate Management

- **Duplicate Rules** can be configured to prevent users from entering the same Lost or Found item multiple times, improving **data quality**.

◆ Deployment

- The app can be deployed from **Sandbox → Production** using **Change Sets** to move *Apex classes, triggers, and custom objects*.

◆ Other Tools

- Features like **Data Loader, ANT Migration Tool, VS Code & SFDX, and managed/unmanaged packages** are **not required** for this project, as it is deployed internally within Salesforce.

3. Conclusion

- Phase 8 features ensure proper **data integrity, testing, and deployment practices**.
- Only the **essential tools** (*Data Import Wizard, Duplicate Rules, Change Sets*) are utilized for Smart Lost & Found.
- **Advanced deployment tools and packages** are **not applicable** for this project's current scope.

Phase 9 — Reporting, Dashboards & Security Review

1. Objective

Phase 9 focuses on providing **insights into application data** and ensuring **proper security and access control** for users.

- **Reports and dashboards** help monitor lost and found items.
- **Security settings** maintain data integrity and privacy.

2. Reports

◆ Lost Items Report

- **Report Type:** Lost_Item__c (custom report type)
- **Format:** Summary
- **Columns:** Item Name, Lost Date, Location, Description, Owner
- **Filters:** Lost Date = *THIS MONTH* (optional), Status = *Open*
- **Grouping:** By Location (*optional sub-group by Lost Date*)

The screenshot shows a Salesforce report interface. At the top, there's a header with a cloud icon, 'Developer Edition', and 'Welcome'. Below the header, the title 'Report Lost_Items' and 'New Lost_Items Report' is displayed. A search bar and various navigation icons are on the right. The main area shows a table with two records. The columns are 'Lost Item Name', 'Location', 'Item Name', and 'Lost Date'. The first record is 'ID card' located at 'Pragati Engineering College' with item name 'ID card' and lost date '9/23/2025, 12:00 PM'. The second record is 'Phone' located at 'Pragati Engineering College' with item name 'Phone' and lost date '9/8/2025, 12:00 PM'. There are buttons for 'Enable Field Editing', 'Add Chart', and other report controls at the bottom of the table.

	Lost Item Name	Location	Item Name	Lost Date
1	ID card	Pragati Engineering College	ID card	9/23/2025, 12:00 PM
2	Phone	Pragati Engineering College	Phone	9/8/2025, 12:00 PM

◆ Found Items Report

- **Report Type:** Found_Item__c (custom report type)
- **Format:** Summary
- **Columns:** Item Name, Found Date, Location, Description, Owner
- **Filters:** Found Date = *THIS MONTH* (optional), Status = *Open*
- **Grouping:** By Location (*optional sub-group by Found Date*)

The screenshot shows the Smart Lost & Found Dashboard interface. At the top, there's a navigation bar with links for LostItems, FoundItems, ClaimRequests, Reports, and Dashboards. Below the navigation is a search bar and a toolbar with various icons. The main area displays a report titled "New Found Items Report". The report has a header row with columns: FoundItem Name, Found Date, Location, and Status. Two items are listed: "Chain" found on 9/10/2025 at Pragati Engineering College with status "Unclaimed" and "Umbrella" found on 9/15/2025 at Pragati Engineering College with status "Matched". Below the report is a link labeled "View Report (New Found Items Report)".

◆ Claim Requests Report

- **Report Type:** ClaimRequest__c (standard)
- **Format:** Summary
- **Columns:** Item Name, Created Date, Approval Status, Claimant Name, Owner
- **Filters:** Approval Status = *Pending*, Created Date = *LAST 30 DAYS* (optional)
- **Grouping:** By Approval Status (*optional sub-group by Created Date*)

FoundItem Name	Found Date	Location	Status
Chain	9/18/2025, 12:00 PM	Pragati Engineering College	Unclaimed
Umbrella	9/15/2025, 12:00 PM	Pragati Engineering College	Matched

3. Charts

Charts were added to reports for **visual summaries**:

- **Lost Items** → Column chart by *Location*
- **Found Items** → Column or Pie chart by *Location*
- **Claim Requests** → Pie chart by *Approval Status*

Requirements for Charts:

- Reports must be in **Summary or Matrix** format.
- At least one record must exist in the object.
- Must be viewed in **Lightning Experience**.

4. Dashboards

Dashboards provide an **overview of Smart Lost & Found data**.

Components included:

- Lost Items per Location
- Found Items per Location

- Pending Claims by Approval Status

Dashboards also allow **filtering by date or location** for quick insights.

The screenshot shows a software application window titled "Smart Lost & Found...". The top navigation bar includes links for "LostItems", "FoundItems", "ClaimRequests", "Reports", and "Dashboards". Below the navigation is a search bar and a toolbar with various icons. The main content area is titled "Dashboards Recent" and displays a table with two rows of dashboard information:

DASHBOARD	Dashboard Name	Description	Folder	Created By	Created On	Subscribed
Recent	Smart Lost & Found 2 Dashboard	Private Dashboards	CHAVVAKULA LASYAVALLI	9/26/2025, 8:21 AM		
Created by Me	Smart Lost & Found Dashboard	Private Dashboards	CHAVVAKULA LASYAVALLI	9/26/2025, 8:02 AM		

On the left side of the interface, there is a sidebar with categories: "Dashboards" (Recent, Created by Me, Private Dashboards, All Dashboards), "Folders" (All Folders, Created by Me, Shared with Me), and "Favorites" (All Favorites).

5. Security Review

◆ Sharing Settings

- Objects set to **Private** → users see only their records.
- **Admins** have full access.

◆ Field-Level Security

- Sensitive fields (e.g., *Item Description*) restricted to appropriate profiles.

Outcome: Ensures **proper access control and data privacy** while enabling reports and dashboards.

6. Conclusion

- **Reports, charts, and dashboards** provide clear visibility into lost items, found items, and claims.
- **Security settings** protect data and control user access.

- Advanced features like **Dynamic Dashboards**, **Audit Trail**, and **Login IP Ranges** were not implemented, as they were **not required** for this project's scope.