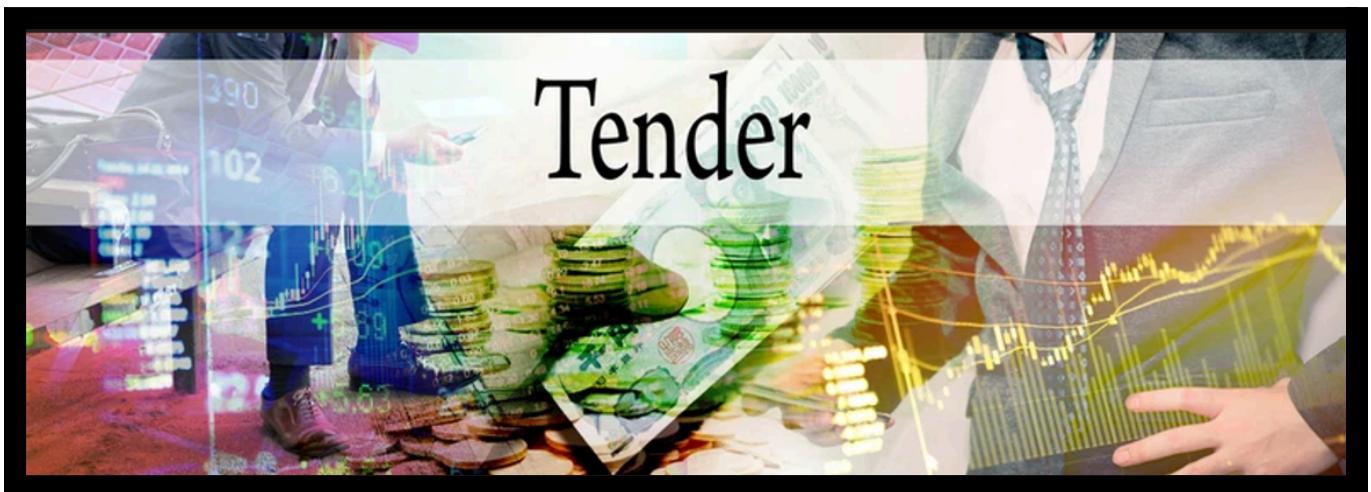


Salesforce

Tender-and-contract-management-system



INTRODUCTION

SALESFORCE-Tender-and-contract-management-system

The traditional process of tendering and contract management in the civil engineering and construction industry is often manual, paper-based, and fragmented, leading to delays, miscommunication, and lack of transparency ,tracking multiple tenders and bid submissions real-time updates on contract approval status, Manual evaluation of bids.

INDUSTRY : Construction / Infrastructure / Civil Engineering Industry.

PROJECT TYPE : Salesforce Custom App Development Project (Admin + Developer).

TARGET USER : “The target users of this system include procurement managers, contractors, project managers, finance teams, and executives who require streamlined tendering, bidding, and contract lifecycle management.

Problem Statement

The traditional process of tendering and contract management in the civil engineering and construction industry is often manual, paper-based, and fragmented. This leads to:

- Delays in processing tenders and contracts
- Miscommunication among stakeholders
- Lack of transparency in bid evaluations
- Challenges in tracking multiple tenders and bid submissions
- Absence of real-time updates on contract approval status

Objectives

- Automate the tender creation and approval process
- Enable contractors to submit and track bids online
- Streamline contract awarding and payment tracking
- Provide dashboards for monitoring tender lifecycle and performance metrics

Phase 1: Problem Understanding & Industry Analysis

👉 Need : To understand what we have to building and why so it will help in project.

1. Problem statement : The traditional process of tendering and contract management in the civil engineering and construction industry is often manual, paper-based, and fragmented. This leads to:

- Delays in processing tenders and contracts
- Miscommunication among stakeholders
- Lack of transparency in bid evaluations
- Challenges in tracking multiple tenders and bid submissions
- Absence of real-time updates on contract approval status

🎯 Objectives :

- Automate the tender creation and approval process
- Enable contractors to submit and track bids online
- Streamline contract awarding and payment tracking
- Provide dashboards for monitoring tender lifecycle and performance metrics.

2. Requirement Gathering : Talk to stakeholders (contractor , engineers , government companies etc) .

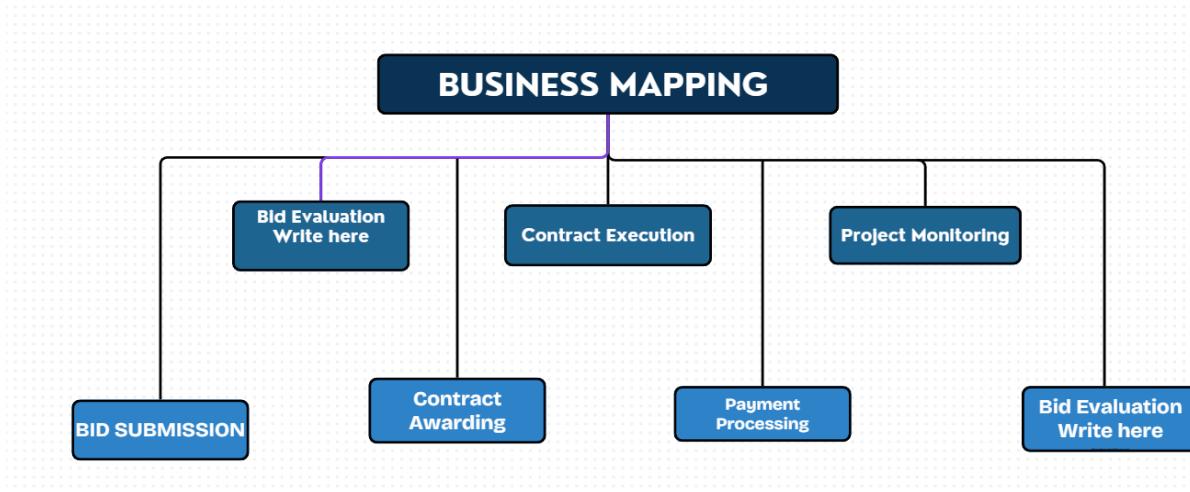
Example

- Stakeholder Interviews: Conducted with procurement, finance, and project management teams
- Document Analysis: Reviewed existing tender and contract documents
- System Analysis: Assessed current manual processes and identified automation opportunities.

3. Business Process Mapping: Tender Creation: Procurement team creates and publishes tenders

- ❖ **Bid Submission:** Contractors submit bids through the system
- ❖ **Bid Evaluation:** Procurement team evaluates bids based on predefined criteria
- ❖ **Contract Awarding:** Selected contractor is awarded the contract
- ❖ **Contract Execution:** Contractor performs work as per contract terms
- ❖ **Payment Processing:** Finance department processes payments based on milestones

- ❖ **Project Monitoring:** Project managers track progress and ensure compliance
- ❖ **Contract Closure:** Upon completion, the contract is closed, and final reports are generated



4. Industry-specific Use Case Analysis :

- Government Infrastructure Projects: Managing tenders for road, bridge, and building construction
- Private Construction Companies: Handling contracts with multiple subcontractors
- Real Estate Developers: Automating tendering process for material supply
- Engineering Procurement & Construction (EPC) Firms: Managing large-scale.

5 AppExchange Tools Explored :

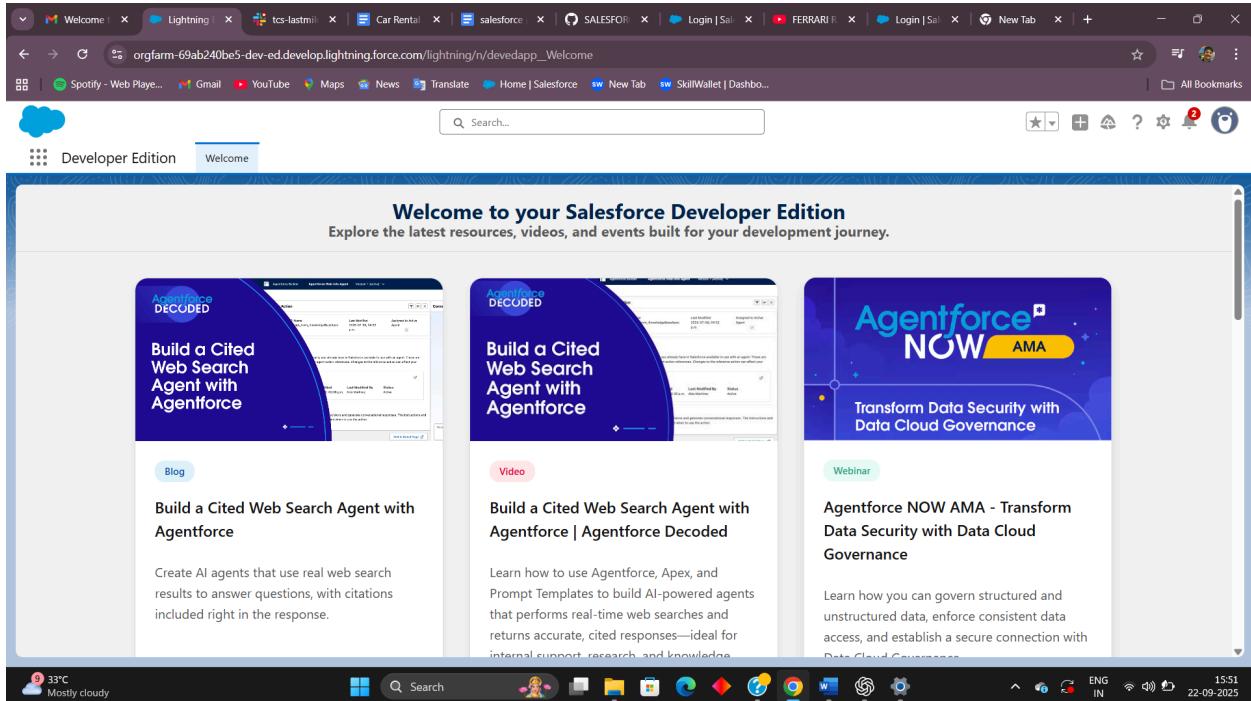
- DocuSign for Salesforce: For digital signing of contracts
- Conga Composer: To generate tender documents and reports
- Salesforce Maps: For visualizing project locations
- Accounting Seed / FinancialForce: For financial tracking and integration
- Formstack / Nintex: To create digital forms for bid submissions
- Tableau CRM (Einstein Analytics): For advanced analytics and dashboards

Phase 2 : Org Setup & Configuration

👉 Goal: Prepare Salesforce environment.

1. Salesforce Editions :

- Select Developer Edition (free, best for practice) .
- Justify why (for learning + testing Tender & Contract Management system).



2 . Company Profile Setup :

- Company Name: Tender & Contract Management System Pvt. Ltd.
- Primary Contact: System Administrator (Admin User)
- Default Currency: INR (₹) – Indian Rupee
- Default Locale: English (India)
- Default Time Zone: (GMT+5:30) Asia/Kolkata
- Default Language: English
- Corporate Address: (You can add a dummy address like New Delhi, India for practice)

Company Information

Default Language: ENGLISH
Default Time Zone: (GMT+05:30) India Standard Time (Asia/Kolkata)

Currency Settings: Currency Locale: Hindi (India) - INR

Turning on multiple currencies introduces permanent changes in your org. This feature can't be turned off. Review the [Implications of Enabling Multiple Currencies](#) before enabling.

Activate Multiple Currencies:

Translation Settings: Enable Data Translation:

Salesforce Newsletter Settings: Users receive the Salesforce newsletter
 Users receive the Salesforce admin newsletter

Login Notifications: Hide Notices About System Maintenance:
Hide Notices About System Downtime:

Save | Cancel

3. Business Hours & Holidays :

- Define Business Hours (Mon–Fri, 9:00 AM – 6:00 PM IST)
- Add Holidays (e.g., Independence Day, Diwali, New Year)

Business Hours Edit

Step 1. Business Hours Name: Standard Business Hours
Active:

Step 2. Time Zone: (GMT+05:30) India Standard Time (Asia/Kolkata)

Step 3. Business Hours

	Sunday	CLOSED	<input type="checkbox"/> CLOSED	<input type="checkbox"/> 24 hours
Monday	09:00 AM	<input type="checkbox"/> 6:00 PM	<input type="checkbox"/> 24 hours	
Tuesday	09:00 AM	<input type="checkbox"/> 6:00 PM	<input type="checkbox"/> 24 hours	
Wednesday	09:00 AM	<input type="checkbox"/> 6:00 PM	<input type="checkbox"/> 24 hours	
Thursday	09:00 AM	<input type="checkbox"/> 6:00 PM	<input type="checkbox"/> 24 hours	
Friday	09:00 AM	<input type="checkbox"/> 6:00 PM	<input type="checkbox"/> 24 hours	
Saturday	09:00 AM	<input type="checkbox"/> 6:00 AM	<input type="checkbox"/> 24 hours	

Save | Cancel

4. Fiscal Year Settings :

- Choose Standard Fiscal Year (Jan–Dec OR Apr–Mar, based on industry).

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

User Detail

Field	Value	Role
Name	Vanish Vishwakarma	Salesforce
Alias	van	Profile
Email	vanshvishwakarma607@gmail.com [Verified]	System Administrator
Username	vanshvishwakarma607237@agentforce.com	Active
Nickname	User17583803024207173080	Marketing User
Title		Flow User
Company	Salesforce Tender-and-contract-management-system	Service Cloud User
Department		Site.com Contributor User
Division		Site.com Publisher User
Address		WDC User
Time Zone	(GMT-07:00) Pacific Daylight Time (America/Los_Angeles)	Mobile Push Registrations
Locale	English (United States)	Data.com User Type
Language	English	Accessibility Mode (Classic Only)
Delegated Approver		
Manager		

5. User Setup & Licenses :

- Create sample users:
- Procurement Manager
- -Contractor
- -Finance Officer
- -System Admin

The screenshot shows the Salesforce Lightning Experience interface. The left sidebar is titled 'Setup' and includes sections for User Management Settings, such as 'User', 'Feature Settings', 'Data.com', 'Service', 'Embedded Service', and 'User Interface'. The main content area is titled 'Users' and shows a user record for 'Vansh Vishwakarma'. The 'User Detail' section contains fields like Name, Alias, Email, Username, Nickname, Title, Company, Department, Division, Address, Time Zone, Locale, and Language. To the right of the detail section, there is a grid of checkboxes for various user profiles, including Active, Marketing User, Offline User, Knowledge User, Flow User, Service Cloud User, Site.com Contributor User, Site.com Publisher User, WDC User, and Mobile Push Registrations. A 'View' button is also present in this grid.

6 . Profiles :

- Standard Profiles: Admin, Standard User
- Custom Profile: Procurement User (restricted to Tender objects)

The screenshot shows the Salesforce Lightning Experience interface. The left sidebar is titled 'Setup' and includes sections for User Management Settings, such as 'User', 'Feature Settings', 'Data.com', 'Service', 'Embedded Service', and 'User Interface'. The main content area is titled 'Profiles' and shows a profile record for 'Procurement Officer'. The 'Profile Detail' section contains fields like Name, User License, Description, and Created By. The 'Page Layouts' section lists standard object layouts for Global, Email Application, Home Page Layout, and Account, along with their respective location group assignments. A 'Custom Profile' checkbox is checked in the profile detail section. A 'View Users' button is also present.

7. Roles :

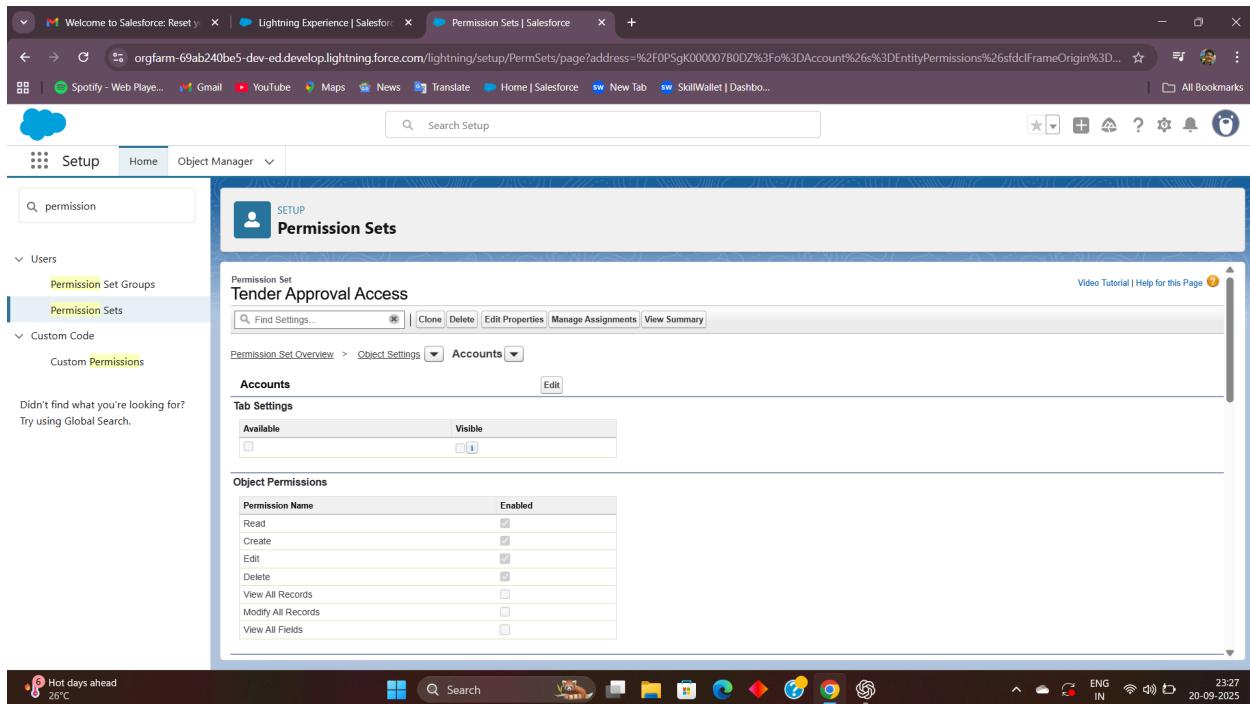
- Role Hierarchy:
- CEO → Procurement Head → Procurement Officer → Contractor



The screenshot shows the Salesforce Setup interface for managing roles. The left sidebar navigation includes 'Setup', 'Home', and 'Object Manager'. Under 'Users', the 'Roles' section is selected. The main content area displays the 'Creating the Role Hierarchy' page. It features a tree view of the role structure under 'Your Organization's Role Hierarchy'. The tree starts with 'Salesforce.Tender-and-contract-management-system' and branches into 'CEO', 'CFO', 'COO', 'Procurement Officer', 'Auditor', 'Finance Officer', 'Vendor / Contractor', 'SVP.Customer Service & Support', 'SVP.Human Resources', and 'SVP.Sales & Marketing'. A search bar at the top right says 'Show in tree view'.

8. Permission Sets :

- Create extra permissions (e.g., Finance Approval, Contract Editing)



9. OWD (Org-Wide Defaults) :

- Tenders – Private
- Contracts – Private
- Bids – Private
- Payments – Private
- Vendors – Controlled by Parent
- Accounts – Private
- Contacts – Controlled by Parent

The screenshot shows the Salesforce Sharing Settings page. The left sidebar lists categories such as Health Check, Login Access Policies, Named Credentials, Network Access, Password Policies, Platform Encryption, Encryption Settings, Key Management, Private Connect, Remote Site Settings, Session Management, Session Settings, and Sharing Settings. Under Sharing Settings, there are sections for Trusted URL and Browser Policy Violations, Trusted URLs, Trusted URLs for Redirects, and View Setup Audit Trail. The main content area displays a table of sharing rules for various objects, with columns for object name, shared with, access level, and a checkmark column. Below the table are sections for Other Settings (Manager Groups, Secure guest user record access, Require permission to view record names in lookup fields) and Sharing Rules (Lead Sharing Rules). The browser status bar at the bottom shows the URL as https://orgfarm-69ab240be5-dev-ed.lightning.force.com/lightning/setup/SecuritySharing/home.

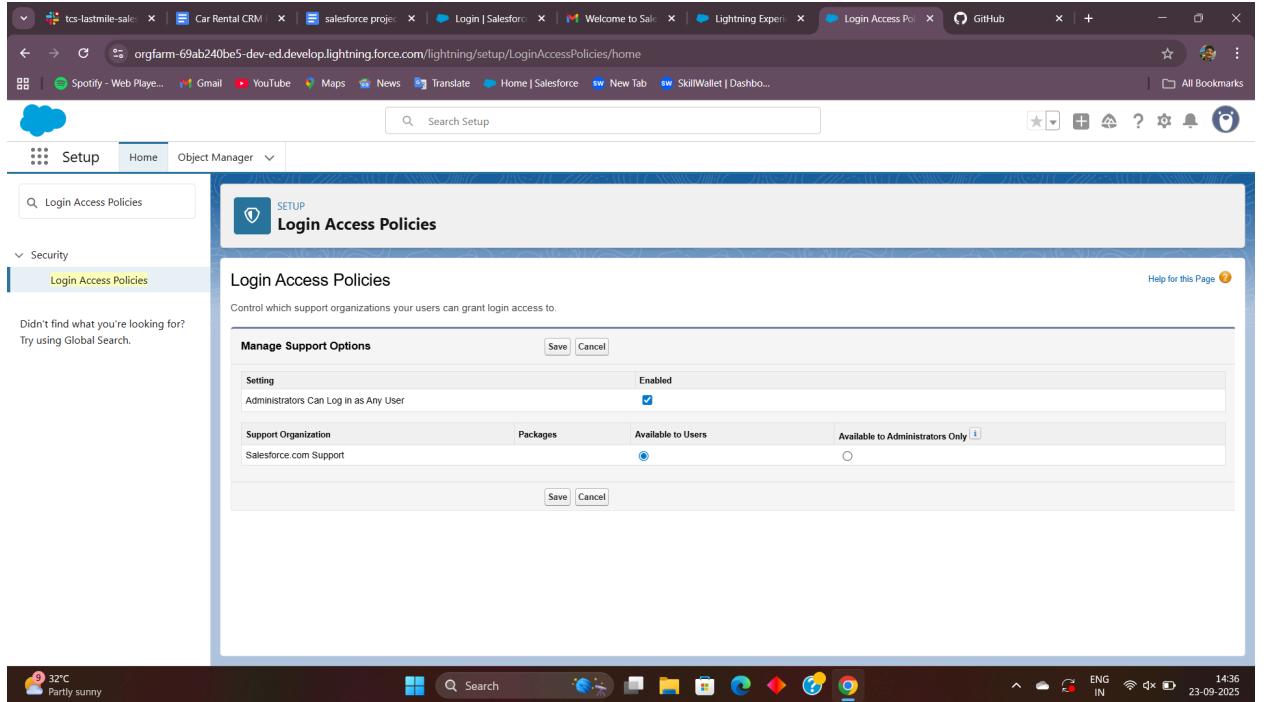
10. Sharing Rules :

- Allow Procurement Head to see all Contracts
- Contractors can only see their own tenders

The screenshot shows the Salesforce Sharing Rules page. It contains five sections: Bids Sharing Rules, Contract Sharing Rules, Payment Sharing Rules, Tender Sharing Rules, and Vendor Sharing Rules. Each section has a table with columns for Action, Criteria, Shared With, and Access Level. The Bids Sharing Rules section shows a rule for 'Owner in Role: Procurement Officer' shared with 'All Internal Users' with 'Read Only' access. The Contract Sharing Rules section shows a rule for 'Owner in Role: Finance Officer' shared with 'All Internal Users' with 'Read Only' access. The Payment Sharing Rules section shows a rule for 'Owner in Role: Finance Officer' shared with 'All Internal Users' with 'Read Only' access. The Tender Sharing Rules section shows a rule for 'Owner in Role: Procurement Officer' shared with 'All Internal Users' with 'Read Only' access. The Vendor Sharing Rules section shows a rule for 'Owner in Role: Procurement Officer' shared with 'All Internal Users' with 'Read Only' access. The browser status bar at the bottom shows the URL as https://orgfarm-69ab240be5-dev-ed.lightning.force.com/lightning/setup/ObjectManager/home.

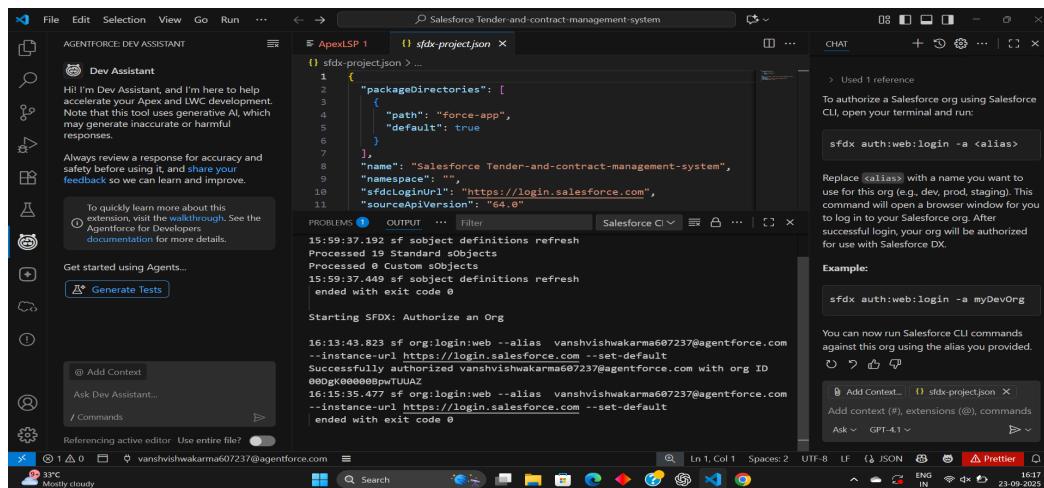
11. Login Access Policies :

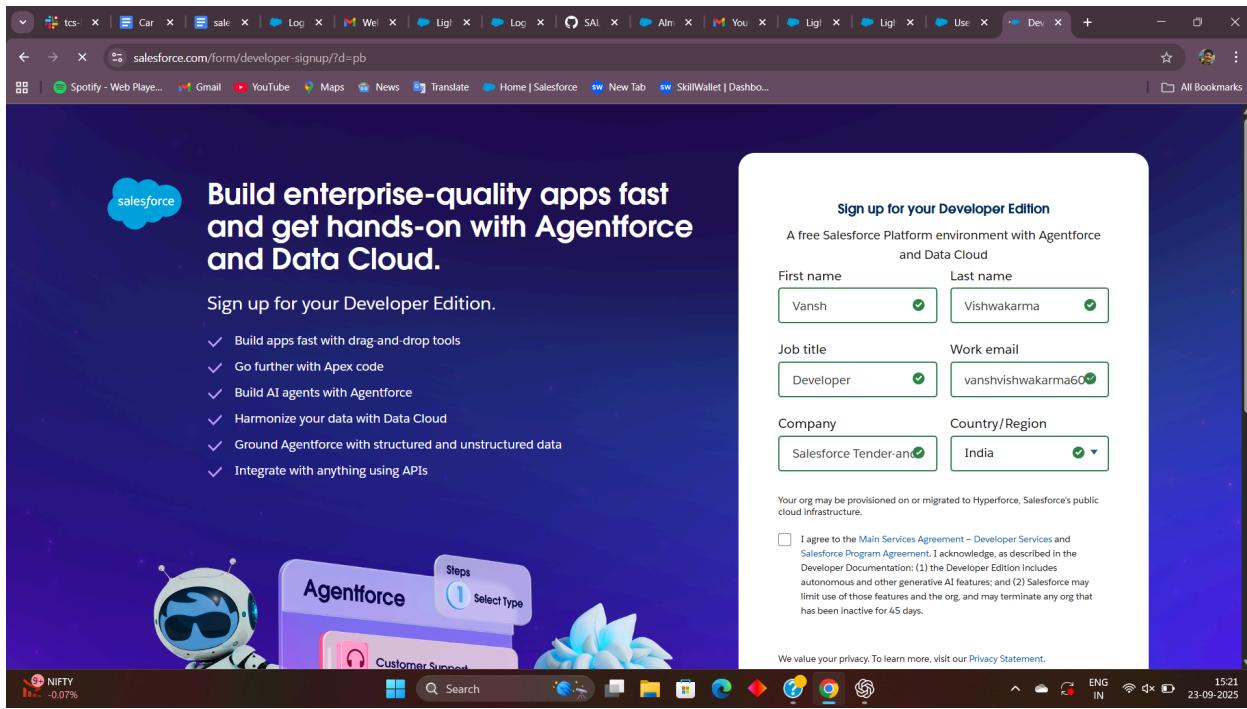
- Define IP Ranges (office-based)
- Set Login Hours (9:00 AM – 7:00 PM)



12. Dev Org Setup :

- Create Developer Org
- Install Salesforce Extensions in VS Code
- Connect Org with Salesforce CLI





Phase 3: Data Modeling & Relationships

👉 Goal: Build data structure.

1. Standard & Custom Objects :

- Standard Objects
 - i. Account → Represents Contractor/Company
 - ii. Contact → Represents People (Contract Manager, Procurement Officer)
- User → Internal Salesforce Users (roles like Manager, Bidder)
 - i. Tender_c → Stores tender details
 - ii. Bid_c → Stores bids submitted by contractors
 - iii. Contract_c → Stores awarded contract details
 - iv. Project_c → Tracks the project execution linked to contracts
 - v. Payment_c → Stores payments made against contracts

NOTE: Here we are creating the object which will help in showing relationship in project.

The screenshot shows the Salesforce Object Manager page. At the top, there's a navigation bar with tabs for Setup, Home, and Object Manager. Below the navigation is a search bar labeled "Search Setup". The main area is titled "Object Manager" with a sub-header "220 Items. Sorted by Label". It displays a table with columns: Task, Task, Standard Object, and various status and date fields. The table includes rows for Tender, Time Slot, User, User Presence, User Provisioning Request, Vendor, Voice Call, Waitlist, Waitlist Participant, Waitlist Service Resource, and Waitlist Work Type. The "Task" column contains names like "Tender__c", "TimeSlot", "User", etc. The "Standard Object" column indicates whether they are Custom or Standard objects. The "Task" column also contains dates like "9/20/2025" and dropdown menus. The bottom of the screen shows a Windows taskbar with icons for various applications and system status indicators.

2. Fields :

- Tender__c → Tender ID (Auto Number), Tender Name (Text), Status (Picklist: Open/Closed/Awarded)
- Bid__c → Bid ID (Auto Number), Amount (Currency), Status (Picklist: Pending/Approved/Rejected)
- Contract__c → Contract Value (Currency), Start Date (Date), End Date (Date)
- Project__c → Project Name (Text), Location (Text), Progress (%)
- Payment__c → Payment Date (Date), Payment Amount (Currency), Mode (Picklist: Bank Transfer, Cheque, Online)

NOTE : Fields are essentially data containers in Salesforce objects. They define what information you want to store about an object

- **Tender__c**

The screenshot shows the Salesforce Object Manager for the 'Tender' object. The left sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main area displays the 'Fields & Relationships' section with 10 items. The fields listed are:

Description	Field Label	Type
Description	Description__c	Text Area(255)
End Date	End_Date__c	Date
Last Modified By	LastModifiedBy	Lookup(User)
Owner	OwnerId	Lookup(User,Group)
Start Date	Start_Date__c	Date
Status	Status__c	Picklist
Tender Name	Tender_ID__c	Text(18)
Tender Name	Name	Text(80)
Tenders ID	Tenders_ID__c	Auto Number

At the bottom, there are standard browser controls and a taskbar showing the date and time.

○ Bid __ c

The screenshot shows the Salesforce Object Manager for the 'Bids' object. The left sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main area displays the 'Fields & Relationships' section with 10 items. The fields listed are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Bid Amount	Bid_Amount__c	Currency(18, 0)		
Bid ID	Bid_ID__c	Auto Number		
Bids Name	Name	Text(80)		
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Remarks	Remarks__c	Text Area(255)		
Status	Status__c	Picklist		
Submitted Date	Submitted_Date__c	Date/Time		

At the bottom, there are standard browser controls and a taskbar showing the date and time.

○ Contract_c :

The screenshot shows the Salesforce Setup interface for the Contract object. The left sidebar lists various setup categories under 'Fields & Relationships'. The main area displays a table titled 'Fields & Relationships' with 11 items, sorted by Field Label. The columns include FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Assigned To	Assigned_To__c	Lookup(User)		✓
Contract ID	Contract_ID__c	Auto Number		▼
Contract Name	Name	Text(80)		✓
Contract Value	Contract_Value__c	Currency(18, 0)		▼
Created By	CreatedById	Lookup(User)		▼
End Date	End_Date__c	Date		▼
Last Modified By	LastModifiedById	Lookup(User)		▼
Owner	OwnerId	Lookup(User,Group)		✓
Start Date	Start_Date__c	Date		▼

● Project_c :

The screenshot shows the Salesforce Setup interface for the Project object. The left sidebar lists various setup categories under 'Fields & Relationships'. The main area displays a table titled 'Fields & Relationships' with 10 items, sorted by Field Label. The columns include FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Contract	Contract__c	Lookup(Contract)		✓
Created By	CreatedById	Lookup(User)		▼
End Date	End_Date__c	Date		▼
Last Modified By	LastModifiedById	Lookup(User)		▼
Location	Location__c	Text(18)		▼
Owner	OwnerId	Lookup(User,Group)		✓
Progress	Progress__c	Percent(18, 0)		▼
Project Name	Project_Name__c	Text(18)		▼
Project Name	Name	Text(80)		✓

○ Payment_c :

Fields & Relationships
9 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Contract	Contract_c	Lookup(Contract)		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment Amount	Payment_Amount_c	Currency(18, 0)		
Payment Date	Payment_Date_c	Date		
Payment ID	Payment_ID_c	Auto Number		
Payment Mode	Payment_Mode_c	Picklist		
Payment Name	Name	Text(80)		

3. Record Types :

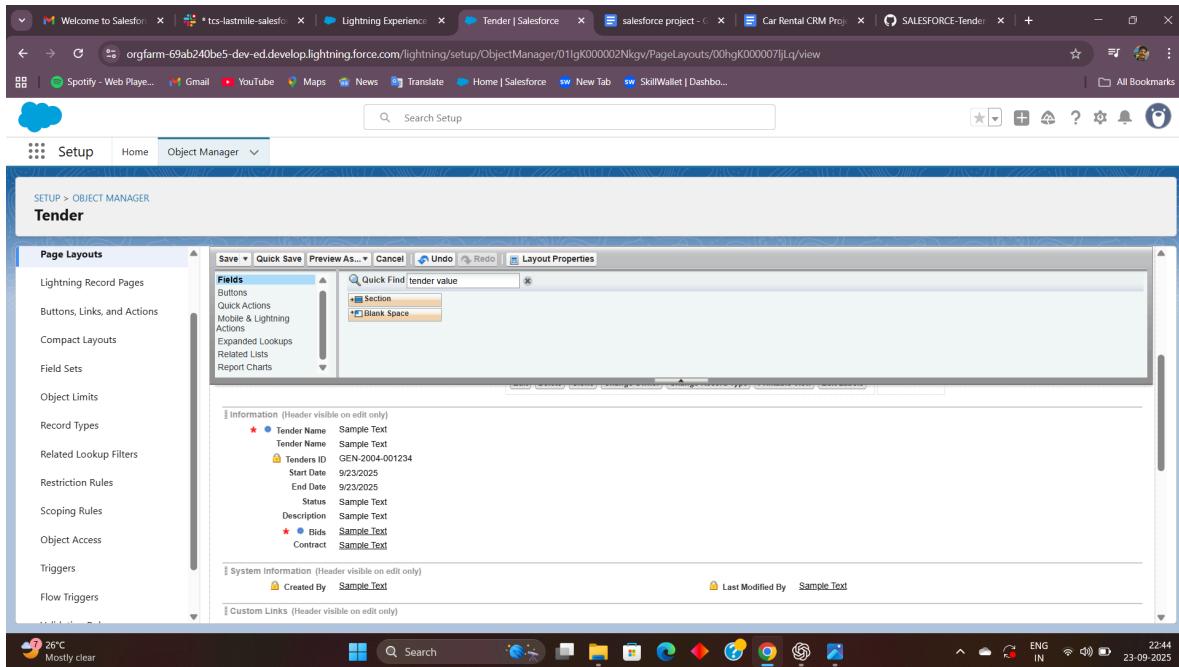
- Tender_c → Govt Tender, Private Tender
- Contract_c → Short-Term, Long-Term

Record Types
2 Items, Sorted by Record Type Label

RECORD TYPE LABEL	DESCRIPTION	ACTIVE	MODIFIED BY
Government Tende	For public works tenders.	✓	Vansh Vishwakarma, 9/23/2025, 9:36 AM
Private Tender	For company/organization-specific tenders.	✓	Vansh Vishwakarma, 9/23/2025, 9:37 AM

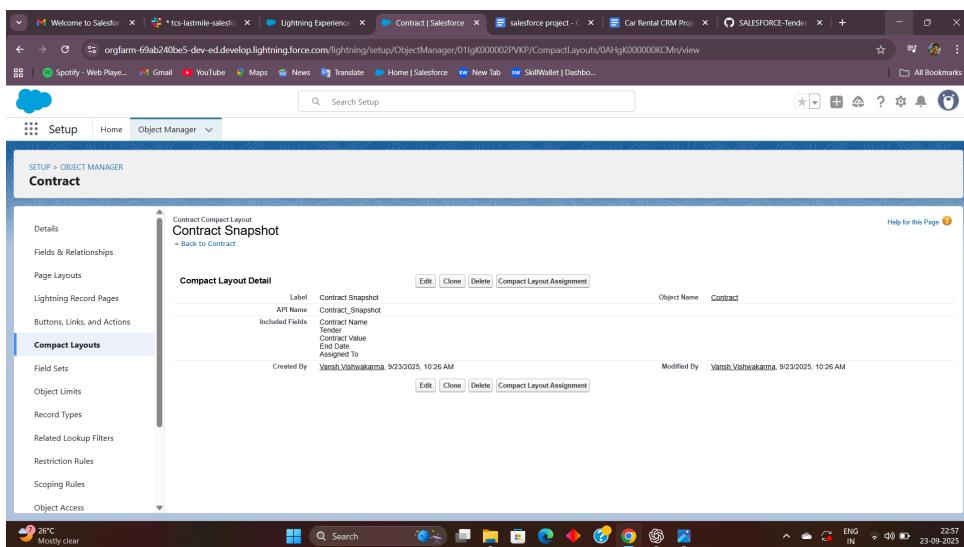
4. Page Layouts :

- `Tender__c` → Fields arranged for Procurement Officers (Name, Status, Start/End Dates, Related Bids)
- `Bid__c` → Layout for Bidders (Amount, Submitted Date, Tender Reference)
- `Contract__c` → Layout for Managers (Contract Value, Status, Related Payments/Projects)



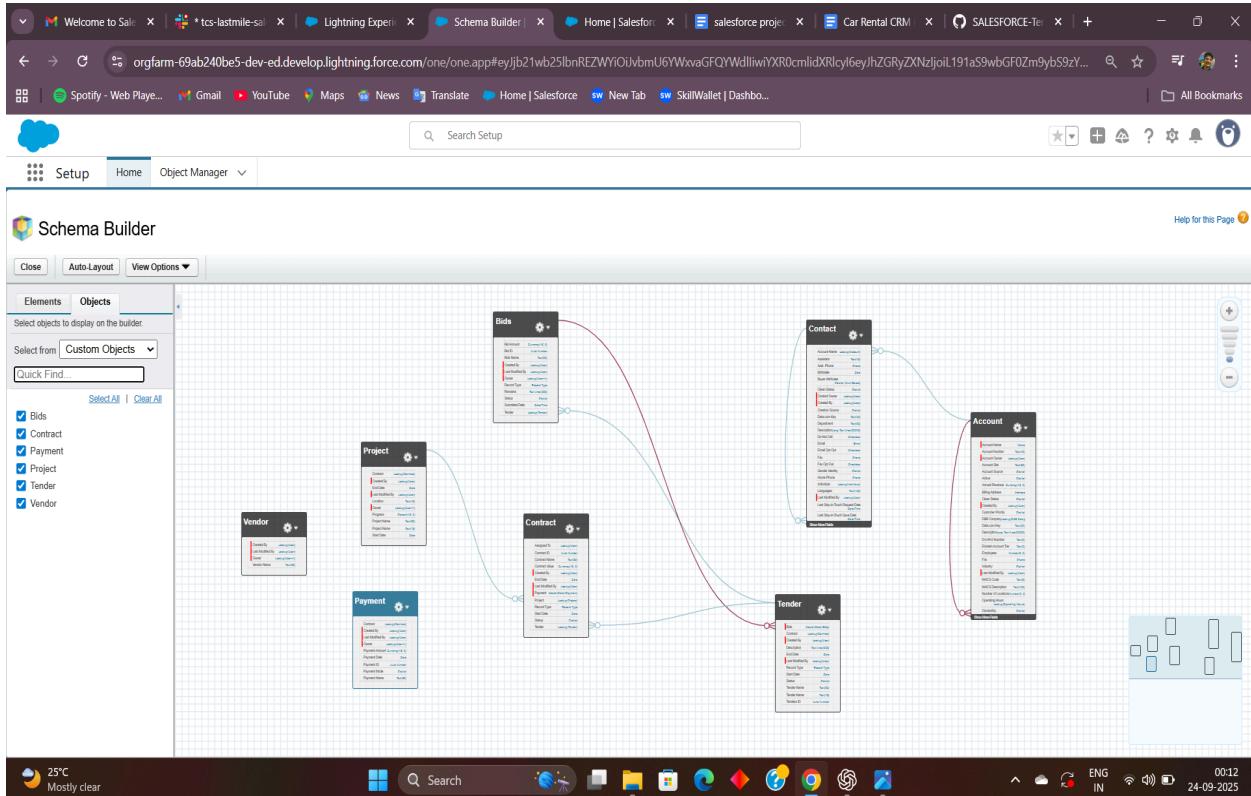
5. Compact Layouts :

- `Tender__c` → Show Tender Name, Status, Start Date, End Date
- `Bid__c` → Show Bid ID, Amount, Status
- `Contract__c` → Show Contract ID, Value, Status



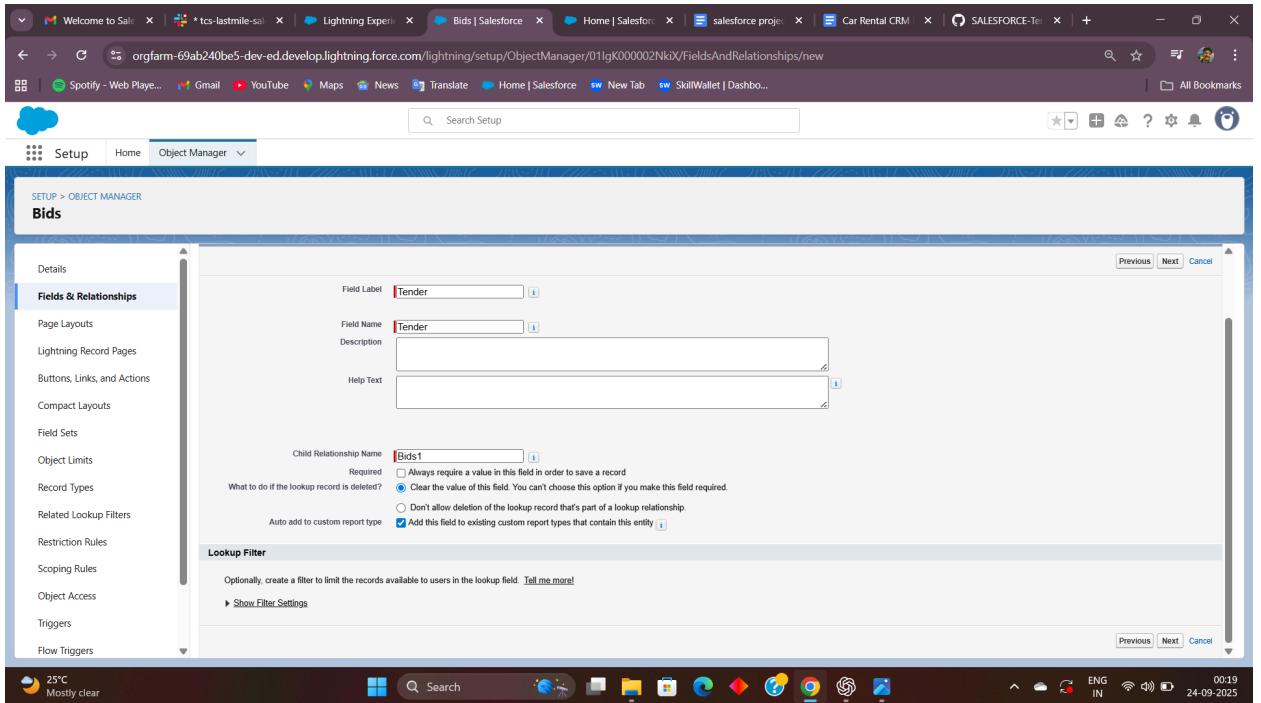
6. Schema Builder :

- Use Setup → Schema Builder
- Add Tender__c, Bid__c, Contract__c, Project__c, Payment__c
- Draw relationships between them → this acts as your ERD (Entity Relationship Diagram).



7. Lookup vs Master-Detail vs Hierarchical :

- Tender__c → Bid__c = Master-Detail (One Tender, many Bids)
- Tender__c → Contract__c = Lookup (One Tender leads to one Contract)
- Contract__c → Payment__c = Master-Detail (One Contract has many Payments)
- Contract__c → Project__c = Lookup (Link Project execution to Contract)
- User Hierarchy (Role Reporting) = Hierarchical



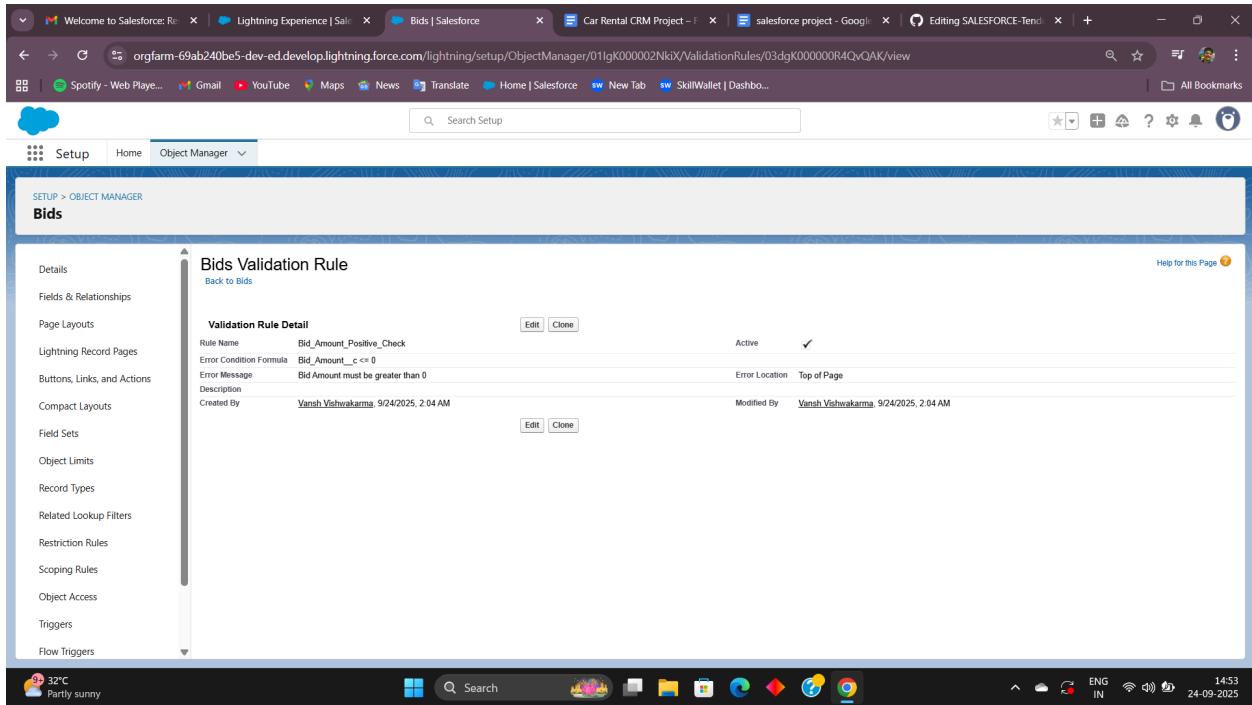
Phase 4: Process Automation (Admin)

👉 Goal: Automate tasks.

1. Validation Rules :

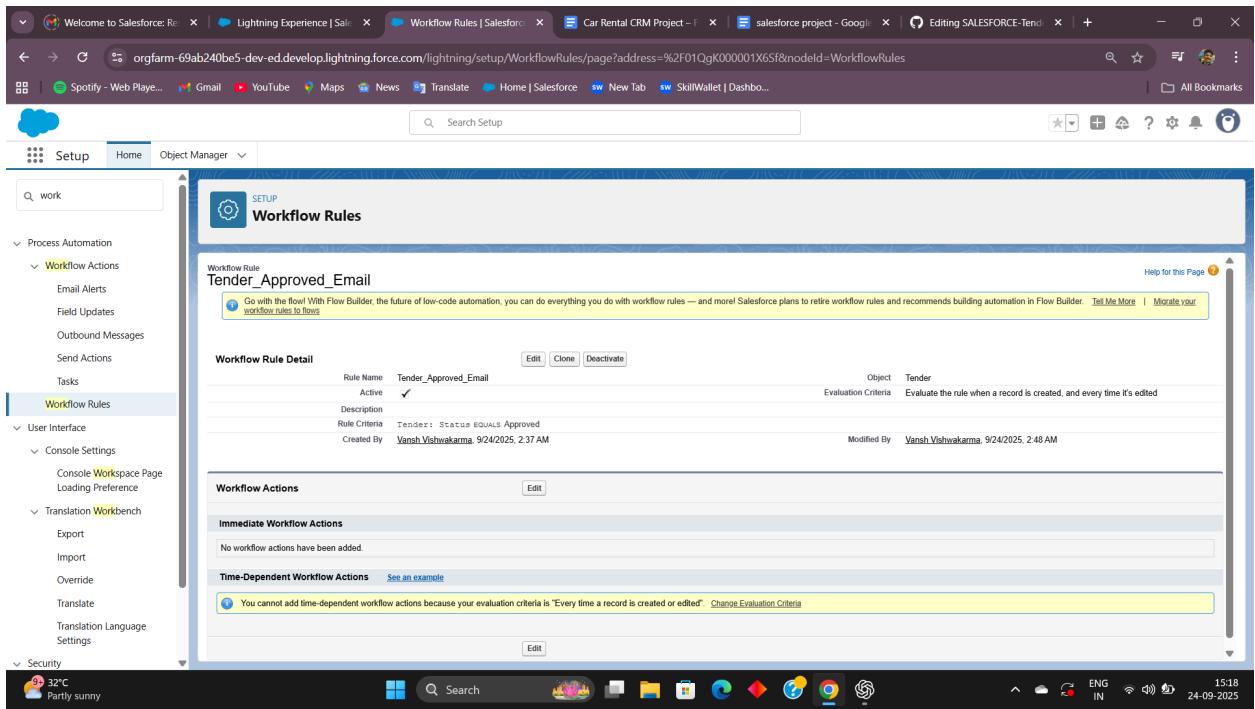
- Validation rules are used to ensure data integrity by preventing users from saving invalid data. Below are the rules created for each object:
- Tender__c (End Date Check)
 - Logic: The End Date cannot be earlier than the Start Date.
 - Error Message: "*End Date cannot be before Start Date.*"
- Bid__c (Bid Amount Positive Check)
 - Logic: The Bid Amount should always be greater than zero.
 - Error Message: "*Bid Amount must be greater than 0.*"

- Contract_c (Contract Value Positive Check)
 - Logic: The Contract Value must be greater than zero.
 - Error Message: “*Contract Value must be greater than 0.*”
- Payment_c (Payment Date Check)
 - Logic: The Payment Date should not be later than the Contract End Date.
 - Error Message: “*Payment Date cannot exceed Contract End Date.*”



2. Workflow Rules (legacy) :

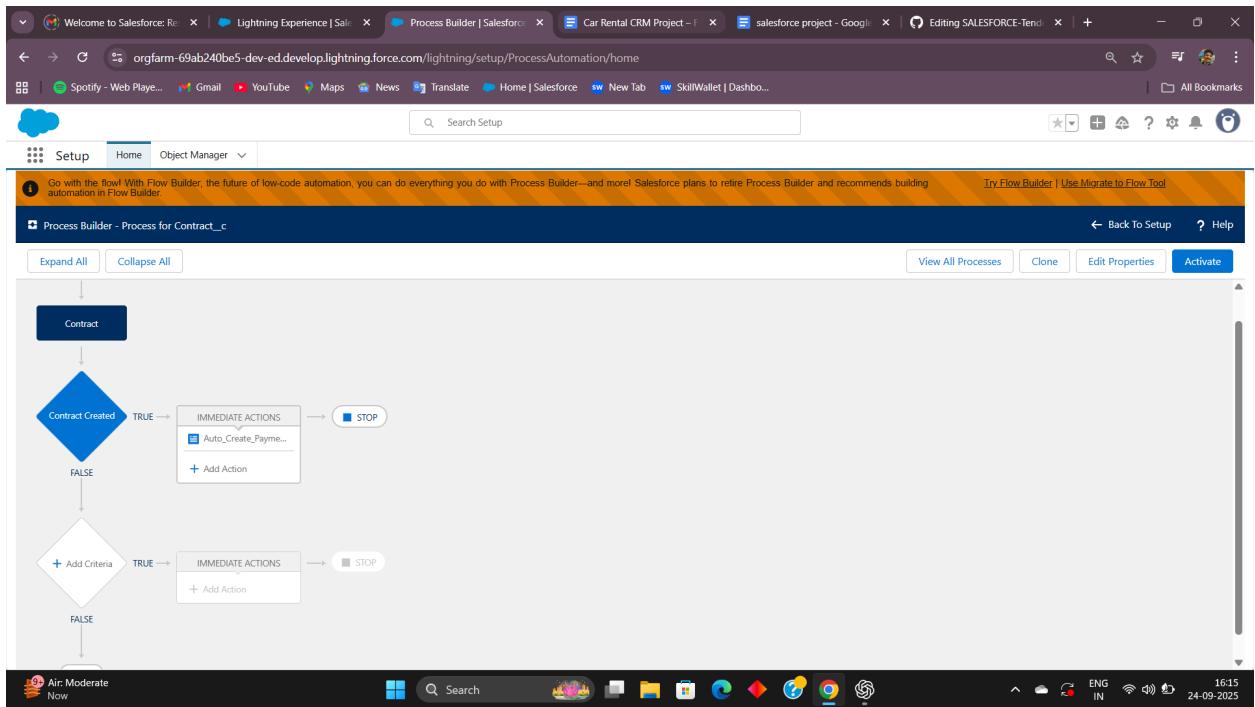
- Tender Object (Tender_c):
When the Status of a Tender becomes Approved, the system will automatically send an Email Alert to the Procurement Officer.
- Bid Object (Bid_c):
When the Status of a Bid is set to Submitted, a Task will be generated to notify the Project Manager about the new bid.
- Contract Object (Contract_c):
When the Status of a Contract is marked as Signed, a Field Update will occur automatically, changing the related Tender's status to Closed.



3. Process Builder (legacy) :

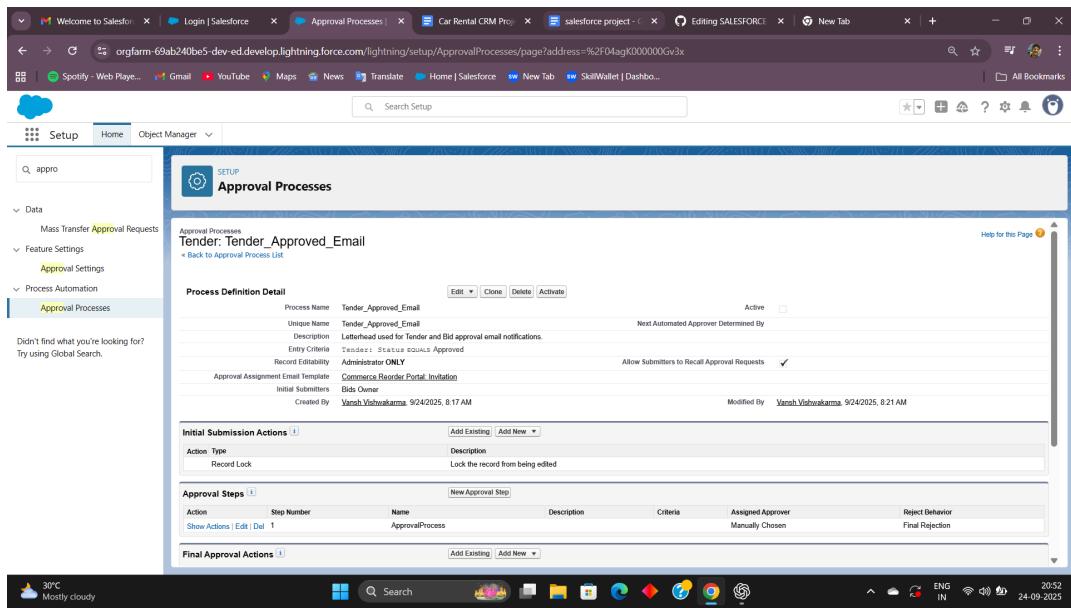
Automate multi-step business logic.

- Object Criteria Action
- Bid__c Status = Submitted Update Tender Status → “Bids Received”, Send Email Notification
- Contract__c Contract Created Auto-create Payment records for milestones
- Tender__c Status = Cancelled Update all related Bids → Status = Cancelled



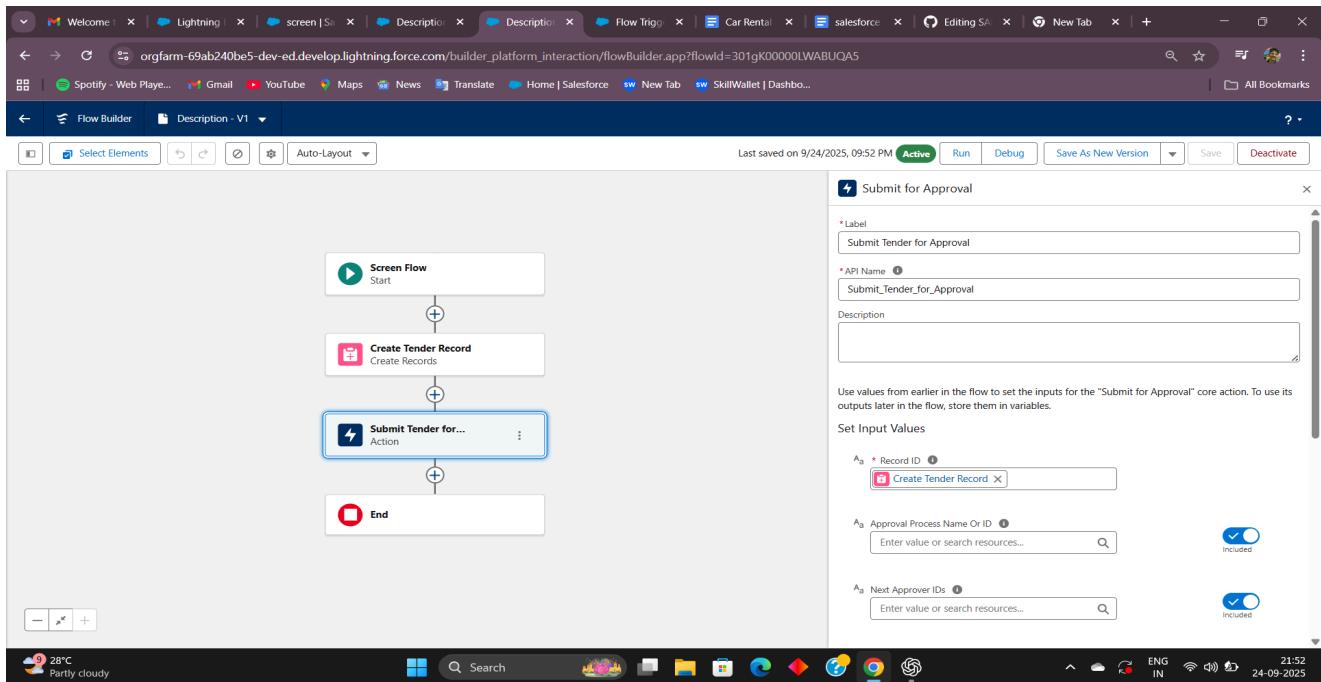
4. Approval Process :

- Automate approvals for Tenders and Bids.
- Tender Approval Flow:
- Procurement Officer submits Tender → Project Manager approval
- Project Manager approves → Director approval
- Final Approval → Status = Approved + Email notification
- Rejection → Status = Rejected + Email notification
- Bid Approval Flow:
- Bid submitted → Procurement Officer review
- Approval → Status = Approved, Tender updated
- Rejection → Status = Rejected, Bidder notified



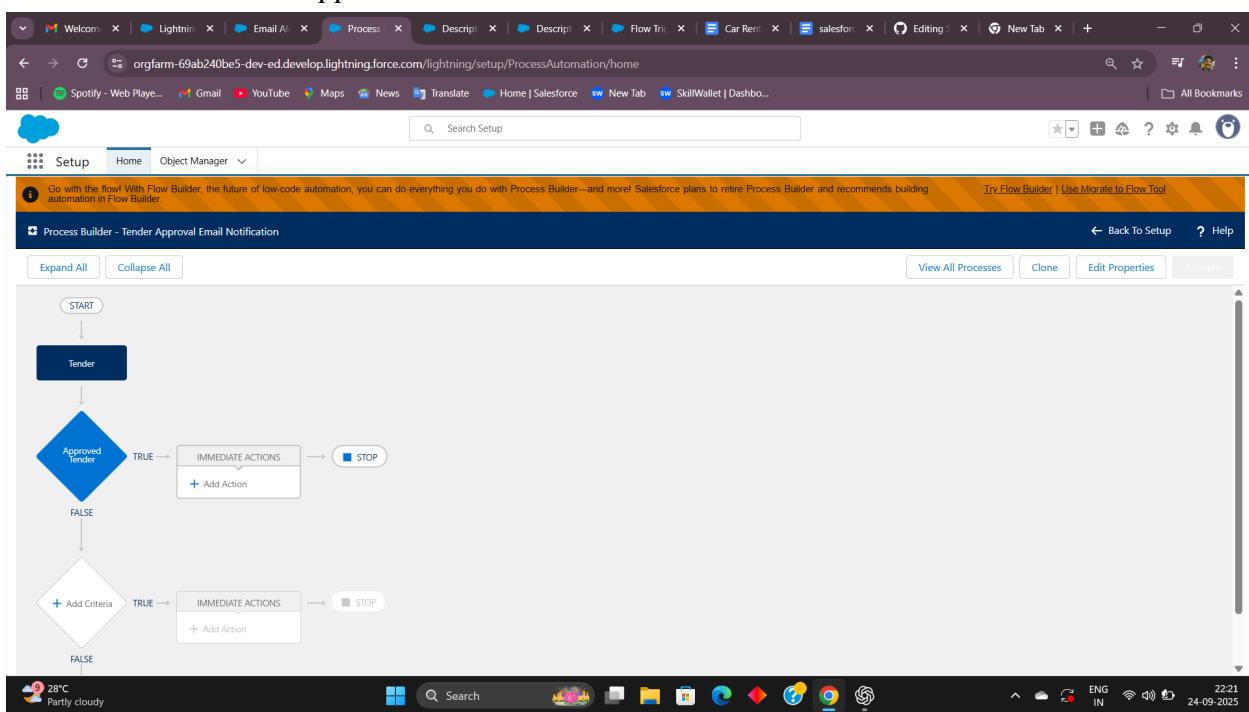
5. Flow Builder :

- Automate complex processes using flows.
- Flow Type Object Purpose Key Steps
- Screen Flow Tender__c Submission form for Procurement Officer Input Tender details → Upload documents → Submit for Approval.



6. Email Alerts :

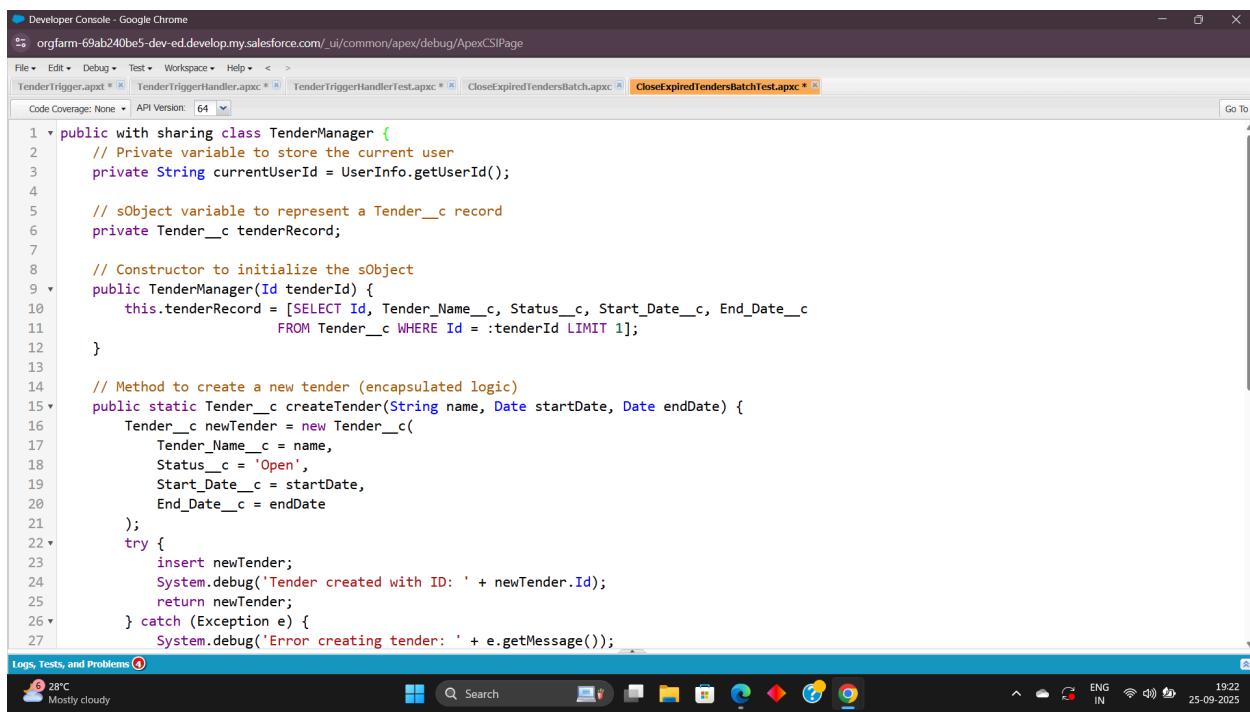
Customer email after approval.



Phase 5: Apex Programming (Developer)

1. Classes & Objects

- Encapsulate logic in Apex Classes for reusability and modularity.
- Create objects (variables, sObjects) to represent and manipulate Salesforce records.



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is orgfarm-69ab240be5-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The console window has multiple tabs open: TenderTrigger.apx, TenderTriggerHandler.apx, TenderTriggerHandlerTest.apx, CloseExpiredTendersBatch.apx, and CloseExpiredTendersBatchTest.apx. The CloseExpiredTendersBatchTest.apx tab is active. The code editor displays the following Apex code:

```
1 public with sharing class TenderManager {  
2     // Private variable to store the current user  
3     private String currentUserId = UserInfo.getUserId();  
4  
5     // sObject variable to represent a Tender__c record  
6     private Tender__c tenderRecord;  
7  
8     // Constructor to initialize the sObject  
9     public TenderManager(Id tenderId) {  
10         this.tenderRecord = [SELECT Id, Tender_Name__c, Status__c, Start_Date__c, End_Date__c  
11                         FROM Tender__c WHERE Id = :tenderId LIMIT 1];  
12     }  
13  
14     // Method to create a new tender (encapsulated logic)  
15     public static Tender__c createTender(String name, Date startDate, Date endDate) {  
16         Tender__c newTender = new Tender__c(  
17             Tender_Name__c = name,  
18             Status__c = 'Open',  
19             Start_Date__c = startDate,  
20             End_Date__c = endDate  
21         );  
22         try {  
23             insert newTender;  
24             System.debug('Tender created with ID: ' + newTender.Id);  
25             return newTender;  
26         } catch (Exception e) {  
27             System.debug('Error creating tender: ' + e.getMessage());  
28         }  
29     }  
30 }
```

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

- Automate actions on Salesforce records when they are created, updated, deleted, or undeleted

```

trigger TenderTrigger on Tender__c (before insert, before update, before delete, after insert, after update, after delete, after undelete) {
    // Before triggers
    if (Trigger.isBefore) {
        if (Trigger.isInsert) {
            TenderTriggerHandler.handleBeforeInsert(Trigger.new);
        } else if (Trigger.isUpdate) {
            TenderTriggerHandler.handleBeforeUpdate(Trigger.new, Trigger.oldMap);
        } else if (Trigger.isDelete) {
            TenderTriggerHandler.handleBeforeDelete(Trigger.old);
        }
    }
    // After triggers
    if (Trigger.isAfter) {
        if (Trigger.isInsert) {
            TenderTriggerHandler.handleAfterInsert(Trigger.new);
        } else if (Trigger.isUpdate) {
            TenderTriggerHandler.handleAfterUpdate(Trigger.new, Trigger.oldMap);
        } else if (Trigger.isDelete) {
            TenderTriggerHandler.handleAfterDelete(Trigger.old);
        } else if (Trigger.isUndelete) {
            TenderTriggerHandler.handleAfterUndelete(Trigger.new);
        }
    }
}

```

3. Trigger Design Pattern :

- Use one trigger per object.
- Delegate logic to a handler class to keep triggers clean and maintainable.

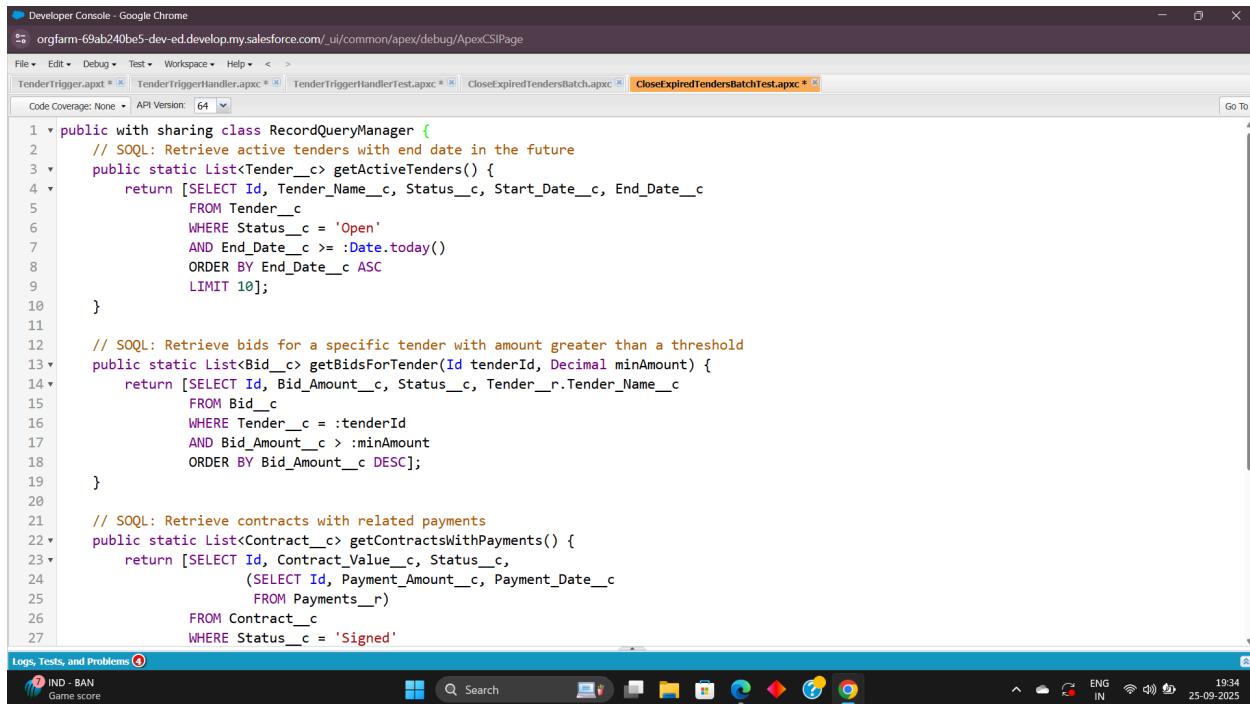
```

trigger TenderTrigger on Tender__c (before insert, before update, before delete, after insert, after update, after delete, after undelete) {
    // Delegate all logic to the handler class
    if (Trigger.isBefore) {
        if (Trigger.isInsert) {
            TenderTriggerHandler.handleBeforeInsert(Trigger.new);
        } else if (Trigger.isUpdate) {
            TenderTriggerHandler.handleBeforeUpdate(Trigger.new, Trigger.oldMap);
        } else if (Trigger.isDelete) {
            TenderTriggerHandler.handleBeforeDelete(Trigger.old);
        }
    }
    if (Trigger.isAfter) {
        if (Trigger.isInsert) {
            TenderTriggerHandler.handleAfterInsert(Trigger.new);
        } else if (Trigger.isUpdate) {
            TenderTriggerHandler.handleAfterUpdate(Trigger.new, Trigger.oldMap);
        } else if (Trigger.isDelete) {
            TenderTriggerHandler.handleAfterDelete(Trigger.old);
        } else if (Trigger.isUndelete) {
            TenderTriggerHandler.handleAfterUndelete(Trigger.new);
        }
    }
}

```

4. SOQL & SOSL :

- SOQL (Salesforce Object Query Language) → Query Salesforce records.
- SOSL (Salesforce Object Search Language) → Search text across multiple objects.



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is orgfarm-69ab240be5-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tabs at the top include TenderTrigger.apxt, TenderTriggerHandler.apxc, CloseExpiredTendersBatch.apxc, and CloseExpiredTendersBatchTest.apxc. The code coverage is set to 64%. The code itself is written in Apex and uses SOQL queries to retrieve tender and bid data based on specific criteria like status and date.

```
1 *public with sharing class RecordQueryManager {
2     // SOQL: Retrieve active tenders with end date in the future
3     public static List<Tender__c> getActiveTenders() {
4         return [SELECT Id, Tender__Name__c, Status__c, Start__Date__c, End__Date__c
5                 FROM Tender__c
6                 WHERE Status__c = 'Open'
7                 AND End__Date__c >= :Date.today()
8                 ORDER BY End__Date__c ASC
9                 LIMIT 10];
10    }
11
12    // SOQL: Retrieve bids for a specific tender with amount greater than a threshold
13    public static List<Bid__c> getBidsForTender(Id tenderId, Decimal minAmount) {
14        return [SELECT Id, Bid__Amount__c, Status__c, Tender__r.Tender__Name__c
15                 FROM Bid__c
16                 WHERE Tender__c = :tenderId
17                 AND Bid__Amount__c > :minAmount
18                 ORDER BY Bid__Amount__c DESC];
19    }
20
21    // SOQL: Retrieve contracts with related payments
22    public static List<Contract__c> getContractsWithPayments() {
23        return [SELECT Id, Contract__Value__c, Status__c,
24                 (SELECT Id, Payment__Amount__c, Payment__Date__c
25                  FROM Payments__r)
26                 FROM Contract__c
27                 WHERE Status__c = 'Signed']
```

5. Collections: List, Set, Map :

- List: Ordered collection of records.
- Set: Unique collection of values.
- Map: Key-value pair collection, useful for fast lookups.
- Automate periodic tasks, like daily updates, reminders, or calculations.

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is org://a0gab240be5-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tabs at the top include 'TenderTrigger.apxt', 'TenderTriggerHandler.apxc', 'TenderTriggerHandlerTest.apxc', 'CloseExpiredTendersBatch.apxc', and 'CloseExpiredTendersBatchTest.apxc'. The code editor displays the following Apex class:

```

1 public with sharing class TenderAutomationManager {
2     // Method to process expired tenders using List and Set
3     public static void processExpiredTenders() {
4         // List to hold tenders to update
5         List<Tender__c> tendersToUpdate = new List<Tender__c>();
6         // Set to track unique tender IDs for processing
7         Set<Id> tenderIds = new Set<Id>();
8
9         // SOQL to get expired tenders
10        List<Tender__c> expiredTenders = [SELECT Id, Tender_Name__c, Status__c, End_Date__c
11                                         FROM Tender__c
12                                         WHERE Status__c = 'Open'
13                                         AND End_Date__c < :Date.today()];
14
15        // Populate Set and List
16        for (Tender__c tender : expiredTenders) {
17            tenderIds.add(tender.Id);
18            tender.Status__c = 'Closed';
19            tendersToUpdate.add(tender);
20        }
21
22        // Update only if there are records
23        if (!tendersToUpdate.isEmpty()) {
24            try {
25                update tendersToUpdate;
26                System.debug('Updated ' + tendersToUpdate.size() + ' expired tenders: ' + tenderIds);
27            } catch (Exception e) {

```

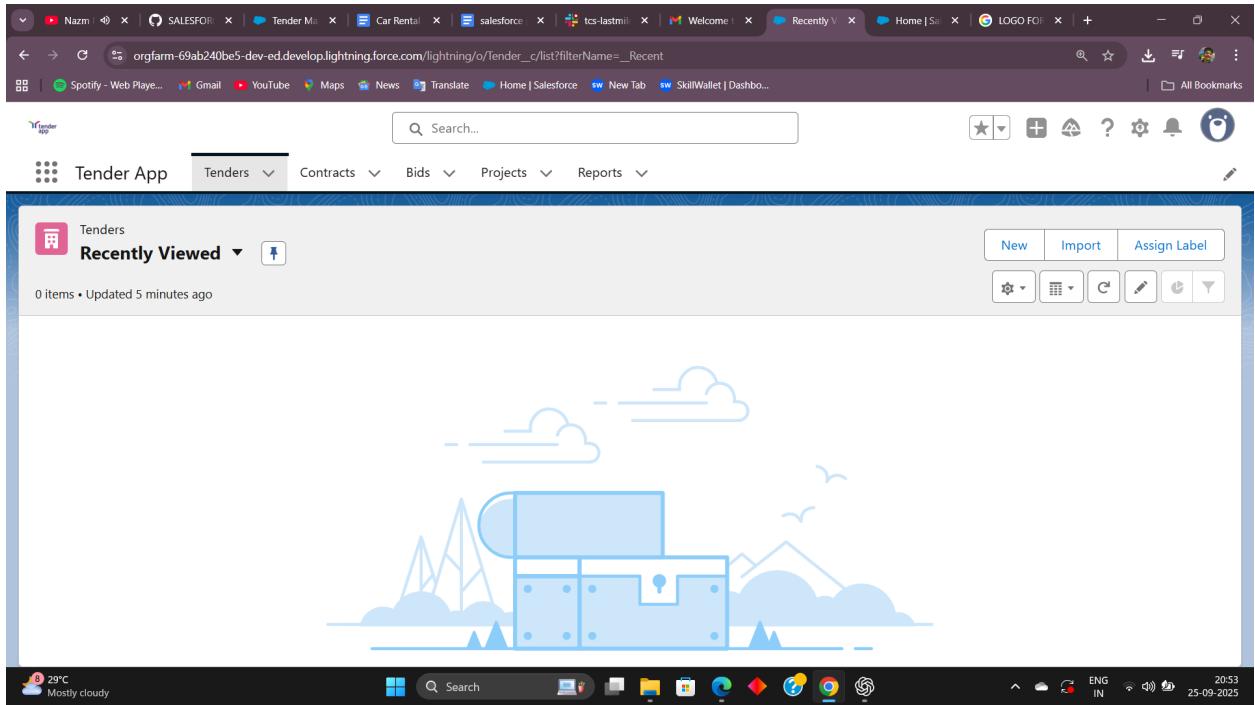
The status bar at the bottom shows 'Logs, Tests, and Problems' with one error icon, system status icons like battery and signal, and the date '25-09-2025'.

Phase 6: User Interface Development

👉 Goal: Make it user-friendly.

1. Lightning App Builder :

- Purpose: Build custom user interfaces without coding.
- What to Do:
- Navigate to Setup → Lightning App Builder.
- Create Custom Pages for different user groups (Procurement Officer, Project Manager, Director).
- Add components like Related Lists, Tabs, Reports, and LWCs.

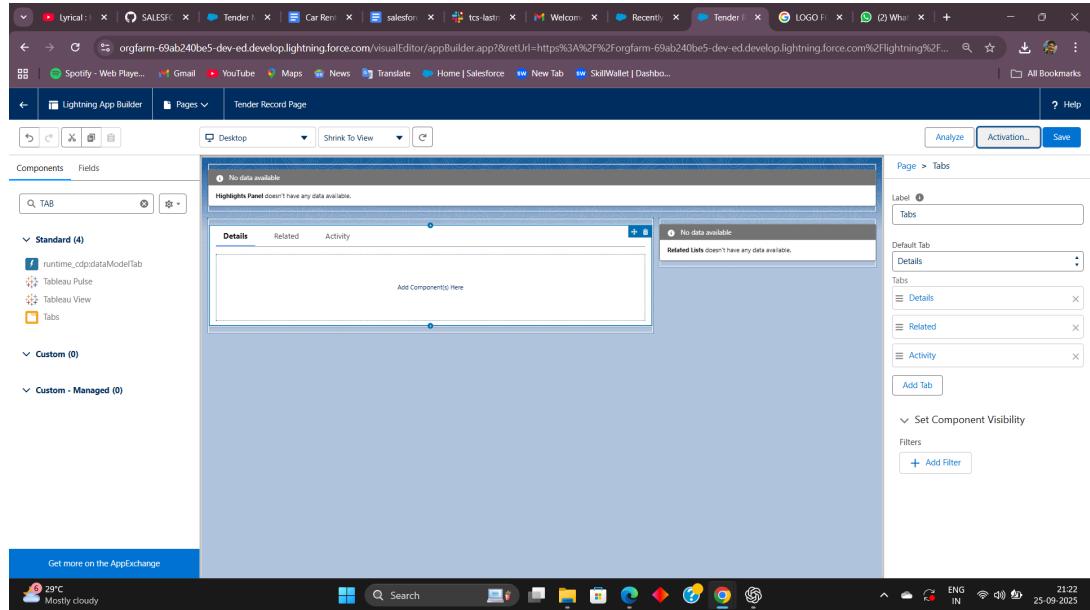


2. Record Pages :

- Purpose: Customize how records (Tender, Bid, Contract, Payment) appear

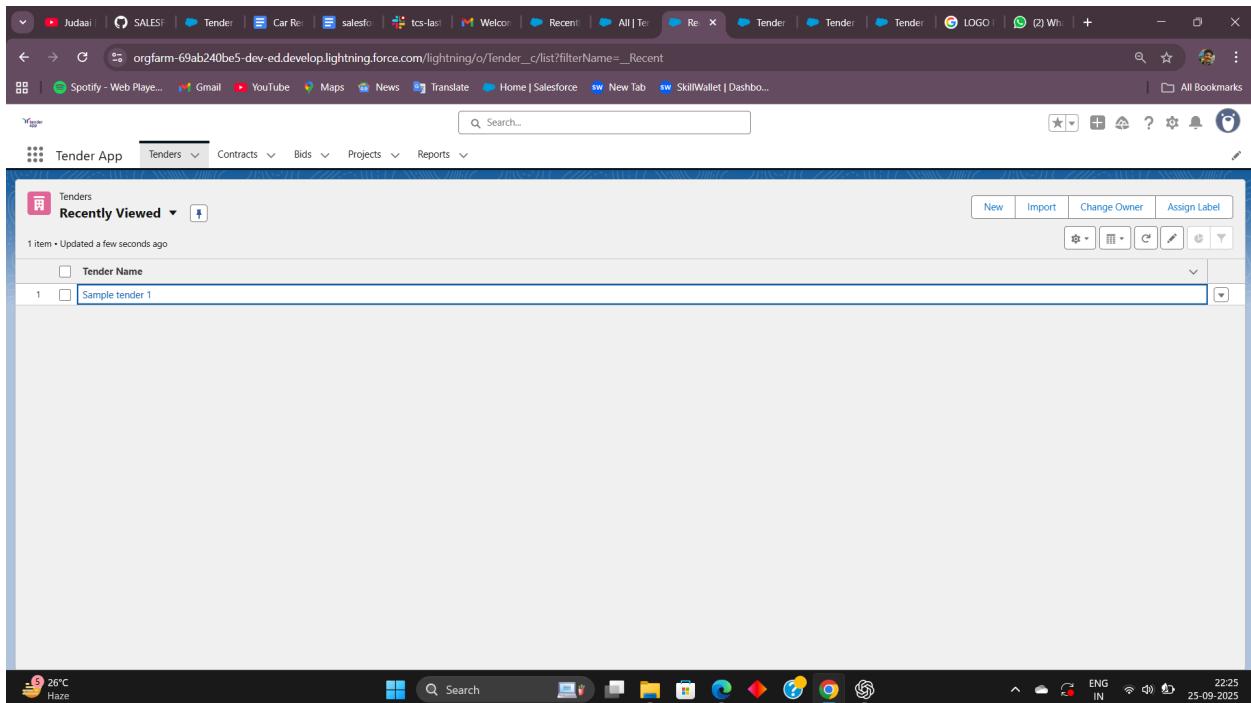
Steps:

- Go to Object Manager → Tender → Lightning Record Pages
- Add Highlights Panel, Tabs (Details, Related, Notes).
- Insert Custom LWCs (like Tender Summary).



3. Tabs :

- Purpose: Provide quick navigation for custom objects.
- Steps:
- Setup → Tabs → New Custom Object Tab.
- Add Tabs for Tender, Bid, Contract, Payment.
- Assign to App Navigation.

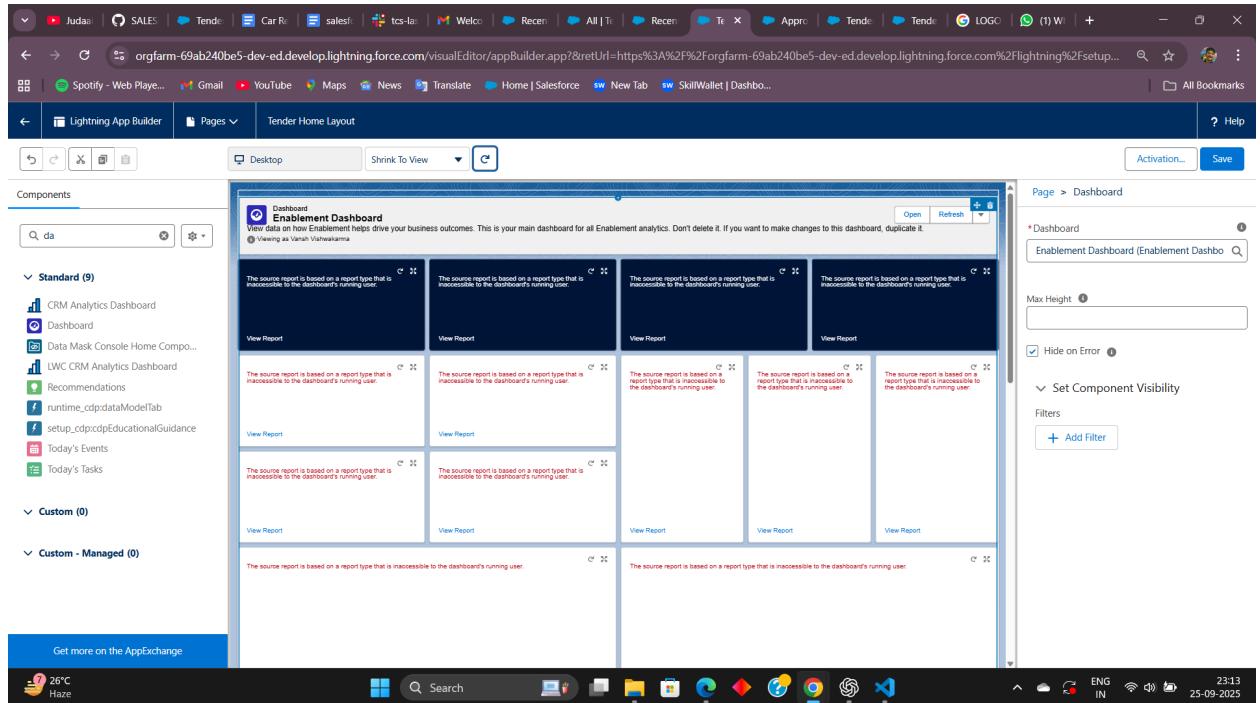


4. Home Page Layouts :

- Purpose: Customize the Salesforce Home page.

Steps:

- Setup → Lightning App Builder → Home Page.
- Add
- Reports/Charts
- Tasks List.
- Approvals Pending.
- Custom Notifications panel.

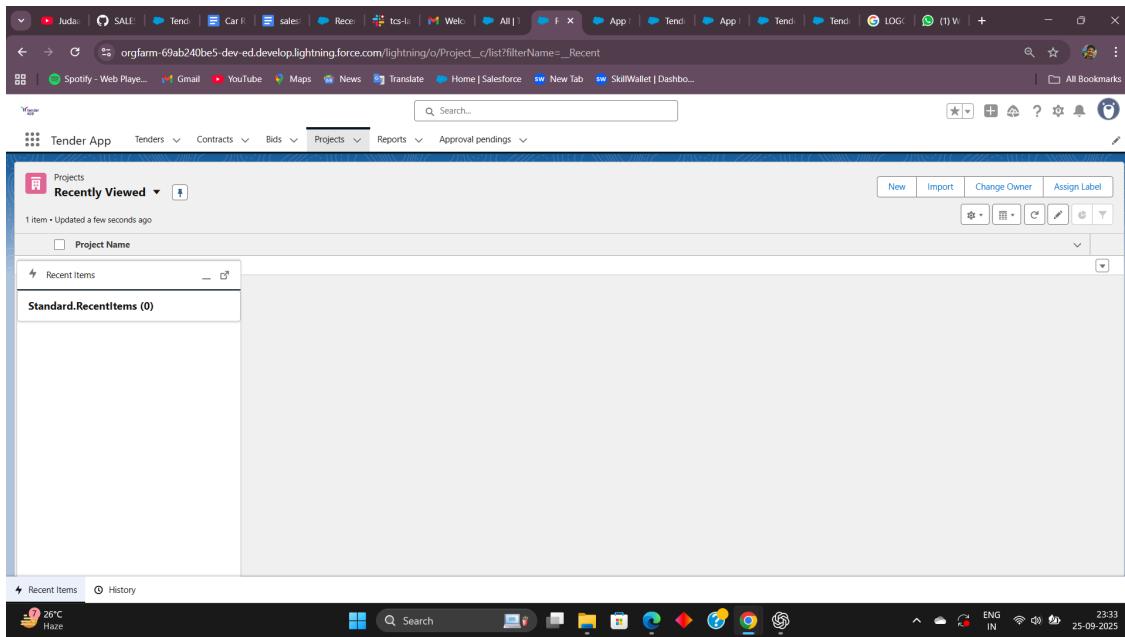


5. Utility Bar :

- Purpose: Provide quick access to tools at the bottom of the screen.

Examples:

- Add Notes.
- Add Recent Items.
- Add History .

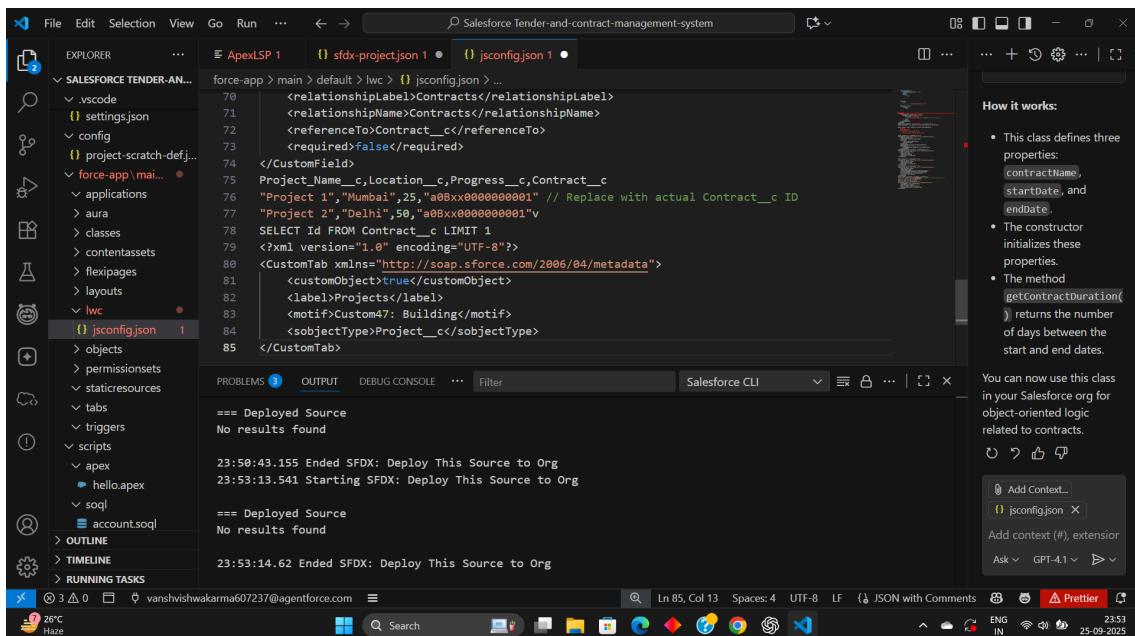


6. LWC :

- Purpose: Build custom UI with JavaScript, HTML, and Apex integration.

Examples for Tender App:

- Tender Summary Component → Displays all bids and total amount.
- Bid Submission Form → Allows users to enter and submit a bid.
- Contract Dashboard → Shows contract progress and payments.



7. Apex with LWC :

Purpose: Fetch Salesforce data via Apex into LWCs.

```
force-app > main > default > lwc > TenderContractController > JS TenderContractController.js 9+ > TenderContractController

1 import { LightningElement } from 'lwc';
2 public with sharing class TenderContractController {
3     @AuraEnabled(cacheable=true)
4     public static List<Tender__c> getActiveTenders() {
5         return [SELECT Id, Tender__Name__c, Status__c, End_Date__c
6                 FROM Tender__c
7                 WHERE Status__c = 'Open' AND End_Date__c >= TODAY
8                 LIMIT 10];
9     }
10
11     @AuraEnabled(cacheable=true)
12     public static List<Contract__c> getSignedContracts() {
13         return [SELECT Id, Contract__Value__c, Status__c, Tender__r.Tender__Name__c
14                 FROM Contract__c
15                 WHERE Status__c = 'Signed'
16                 LIMIT 10];
17 }

PROBLEMS 46 OUTPUT DEBUG CONSOLE TERMINAL PORTS HISTORY Filter Salesforce CLI ... | [] x
NO RESULTS FOUND

23:53:14.62 Ended SFDX: Deploy This Source to Org
00:15:08.312 Starting SFDX: Create Lightning Web Component
target dir = c:\Users\Vansh\Desktop\PROJECT\Salesforce Tender-and-contract-management-system\force-app\main\default\lwc
create force-app\main\default\lwc\tenderContractController\tenderContractController.js
create force-app\main\default\lwc\tenderContractController\tenderContractController.html
create force-app\main\default\lwc\tenderContractController\_tests_\tenderContractController.test.js
create force-app\main\default\lwc\tenderContractController\tenderContractController.js-meta.xml
00:15:08.441 Finished SFDX: Create Lightning Web Component
```

Phase 7: Integration & External Access

1 Named Credentials :

- Store authentication settings for external systems securely.

The screenshot shows the Salesforce Setup interface for Named Credentials. The URL in the browser is <https://orgfarm-69ab240be5-dev-ed.lightning.force.com/lightning/setup/NamedCredential/home>. The page title is "Named Credentials". The "Named Credentials" tab is selected. A table lists one item:

Label	Type	URL	External Credential	Actions
SupplierAPI	Secured Endpoint	https://supplier-api.com	SupplierWeatherService	[Edit]

The sidebar on the left includes sections for User Interface (Rename Tabs and Labels), Security, and Named Credentials (which is currently selected). A message at the bottom says "Didn't find what you're looking for? Try using Global Search." The system status bar at the bottom shows "27°C Mostly cloudy" and the date "26-09-2025".

2. External Services :

- Register and invoke APIs from external systems directly in Salesforce.

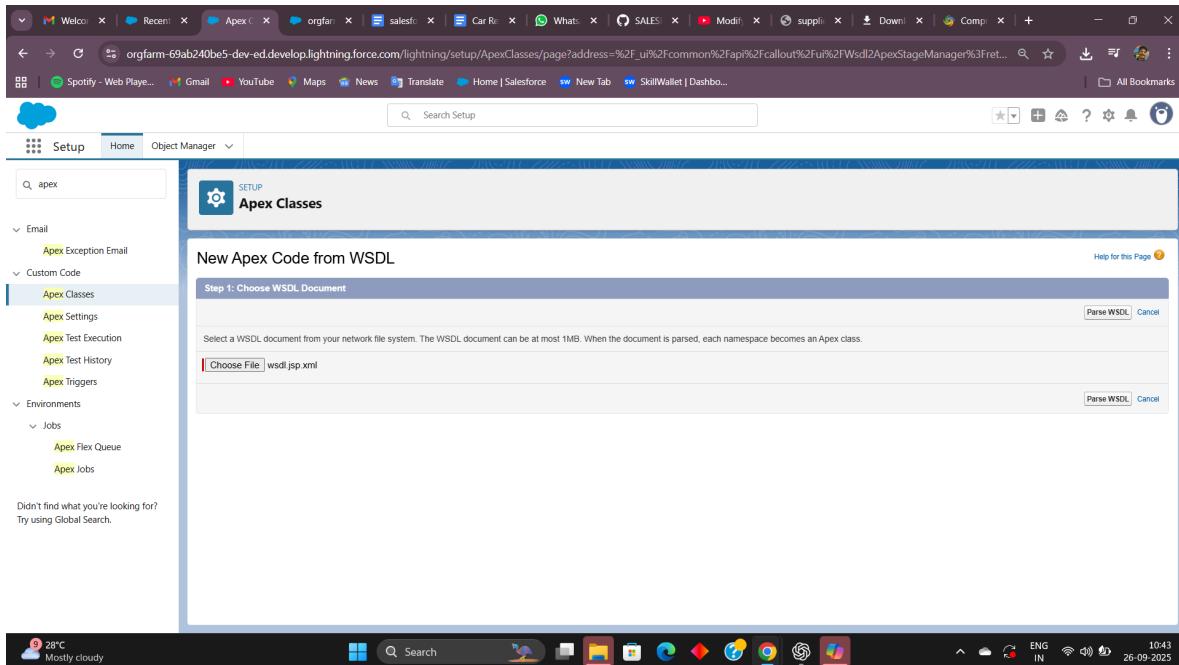
The screenshot shows the Salesforce Setup interface for External Credentials. The URL in the browser is <https://orgfarm-69ab240be5-dev-ed.lightning.force.com/lightning/setup/NamedCredential/home>. The page title is "Named Credentials". The "External Credentials" tab is selected. A table lists one item:

Label	Authentication Protocol	Actions
SupplierWeatherService	Custom	[Edit]

The sidebar on the left includes sections for User Interface (Rename Tabs and Labels), Security, and Named Credentials (which is currently selected). A message at the bottom says "Didn't find what you're looking for? Try using Global Search." The system status bar at the bottom shows "27°C Mostly cloudy" and the date "26-09-2025".

3. Web Services (REST/SOAP)

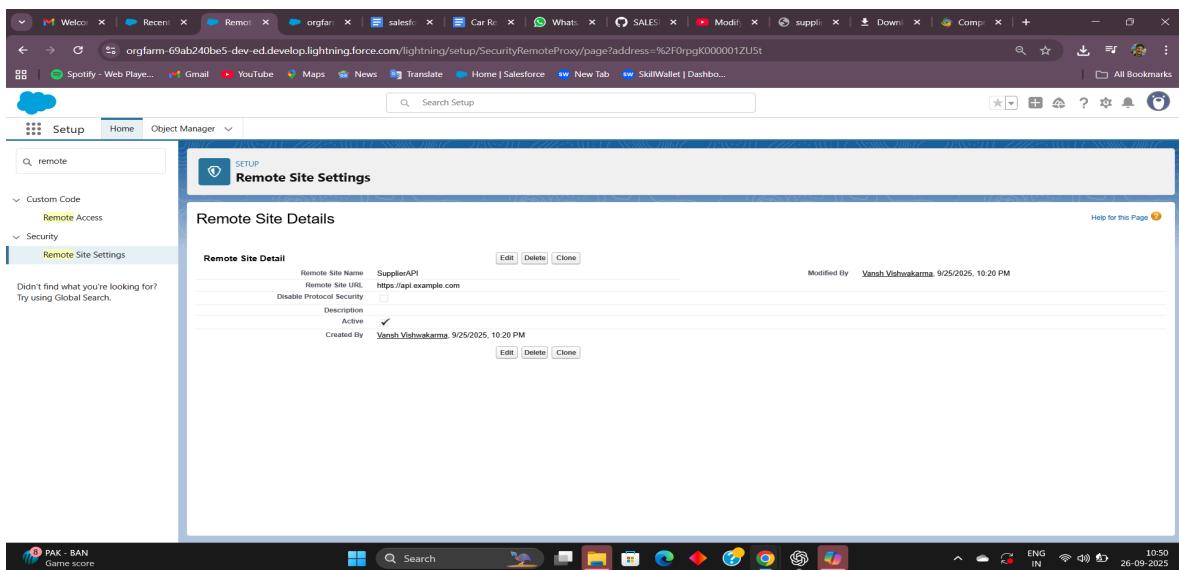
REST callout: Get insurance status.



4. Callouts :

Use HTTP Callouts to integrate Salesforce with external APIs.

Example: Push contract approvals to an external project management tool.



5. Platform Events :

Use event-driven architecture to trigger actions in real-time.

Example: Notify external systems when a tender is approved.

The screenshot shows the Salesforce Setup interface for creating a Platform Event named 'TenderEvent'. The 'Platform Event Definition Detail' section includes fields like Singular Label (TenderEvent), Plural Label (TenderEvents), Object Name (TenderEvent), API Name (TenderEvent__e), Event Type (High Volume), Publish Behavior (Publish Immediately), and Created By (Vansh Vishwakarma). The 'Standard Fields' section lists fields such as Created By, Created Date, Event UUID, and Replay Id. The 'Custom Fields & Relationships' section shows a relationship named 'TenderID' with API Name 'TenderID__c' and Data Type 'Text(18)'. The status is 'Deployed'.

6. Change Data Capture :

Monitor Salesforce record changes in real-time.

Example: Automatically sync updates on Contracts to ERP systems.

The screenshot shows the Salesforce Setup interface for configuring Change Data Capture. It lists available entities like Opportunity, Order, and Payment, and selected entities like 'Tender (Tender__c)' and 'Contract (Contract)'. The status is 'Not Started'.

7. Salesforce Connect :

Access external objects and data without storing it in Salesforce.

Example: View supplier bids from external database within Salesforce UI.

The screenshot shows the Salesforce Setup interface with the 'External Objects' page open. The URL in the browser is <https://orgfarm-69ab240be5-dev-ed.lightning.force.com/lightning/setup/ExternalObjects/page?address=%2F01lgK000002UYPh%3Fsetupid%3DExternalObjects>. The left sidebar shows navigation categories like Email, Apps, Integrations, and External Data Sources. The 'External Objects' section is highlighted. The main content area displays the 'External Object Definition Detail' for the 'Tender' object. It includes fields for Singular Label (Tender), Plural Label (Tenders), Object Name (Tender_x), API Name (Tender_x_x), External Data Source (SupplierAPI), Table Name (Tender table), and various status and creation/modification details. Below this, sections for Standard Fields, Custom Fields & Relationships, and Page Layouts are visible. The bottom of the screen shows the Windows taskbar with various application icons and system status indicators.

8. API Limits :

Monitor and manage API usage to avoid hitting limits.

Example: Limit external integrations to prevent exceeding daily API calls.

The screenshot shows the Salesforce System Overview page. At the top, there's a header bar with various tabs and links. Below it is a navigation bar with 'Setup', 'Home', and 'Object Manager'. The main content area has a title 'System Overview' with a subtitle 'View key usage data for your org.' A message says 'Didn't find what you're looking for? Try using Global Search.' The page is divided into several sections: 'Schema' (with counts for custom objects, total custom objects, custom metadata types, total custom metadata types, custom metadata type usage, and data storage), 'API Usage' (showing 0 API requests last 24 hours), 'Business Logic' (showing rules), and 'User Interface' (showing custom apps). On the right side, there's a vertical scroll bar and a 'Help for this Page' link. The bottom of the screen shows a taskbar with icons for various applications and a system tray with weather information ('Very humid Now'), system status, and date/time ('26-09-2025 11:14').

9. OAuth & Authentication :

Securely authenticate Salesforce with external apps.

Example: OAuth 2.0 flow for integrating with partner systems.

The screenshot shows the Salesforce Named Credentials setup page. The URL is 'orgfarm-69ab240be5-dev-ed.lightning.force.com/lightning/setup/NamedCredential/0XAgK0000009DvhWAE/view'. The page has a header with 'SETUP > NAMED CREDENTIALS' and a title 'SupplierAPI'. It includes fields for 'Label' (SupplierAPI), 'Name' (SupplierAPI), 'URL' (https://supplier-api.com), and 'Enabled for Callouts' (checked). Under 'Authentication', there's an 'External Credential' section with 'SupplierWeatherService' selected. The 'Callout Options' section contains 'Generate Authorization Header' (checked) and 'Allow Formulas in HTTP Header' (checked). The 'Managed Package Access' section shows 'Created By Namespace'. The bottom of the screen shows a taskbar with icons for various applications and a system tray with weather information ('28°C Mostly cloudy'), system status, and date/time ('26-09-2025 11:17').

10. Remote Site Settings :

Register external endpoints to allow Salesforce callouts.

Example: Add supplier API URL in Remote Site Settings to enable communication.

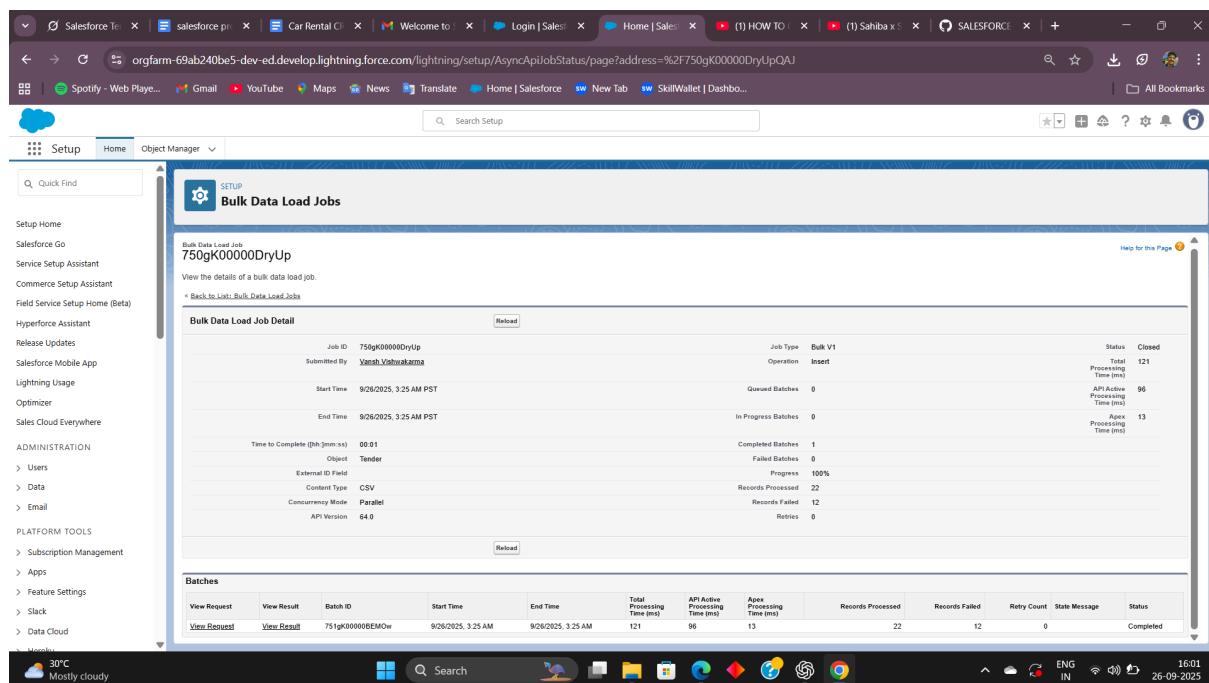
NOTE : we have already done in phase 4 .

Phase 8: Data Management & Deployment

1. Data Import Wizard :

Tool for importing small to medium datasets

Enables adding records for Contractors, Tenders, Bids, Projects, and Payments.



The screenshot shows the Salesforce Setup interface with the 'Bulk Data Load Jobs' page open. The job details are as follows:

Job ID	Submitted By	Operation	Status
750gK00000DryUp	Vashu Vaishwakarma	Insert	Closed
Start Time	9/26/2025, 3:25 AM PST	Queued Batches	0
End Time	9/26/2025, 3:25 AM PST	In Progress Batches	0
Time to Complete (hh:mm:ss)	00:01	Completed Batches	1
Object	Tender	Failed Batches	0
External ID Field		Progress	100%
Content type	CSV	Records Processed	22
Concurrency Mode	Parallel	Records Failed	12
API Version	64.0	Reruns	0

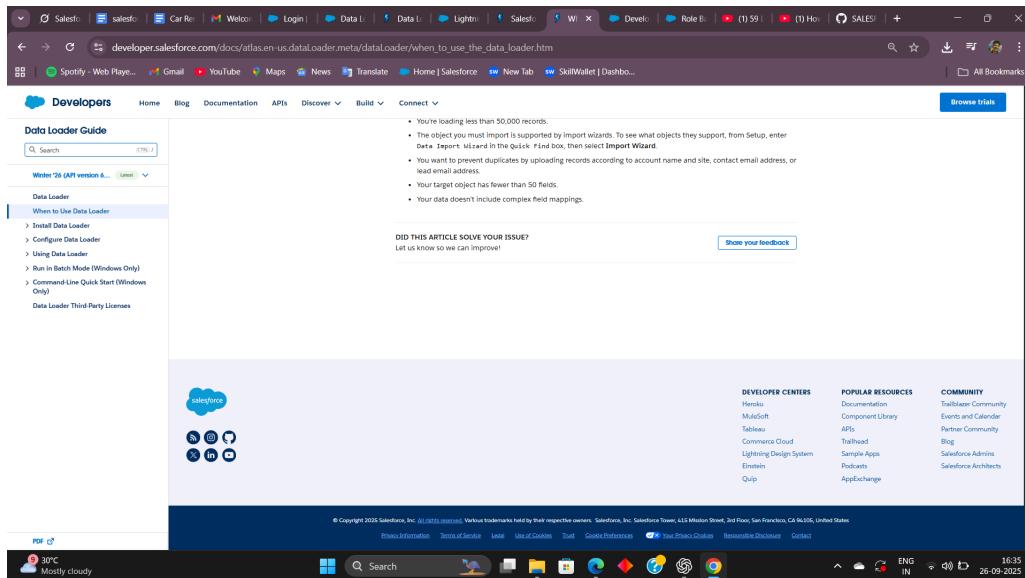
The 'Batches' section shows one batch with the following details:

View Request	View Result	Batch ID	Start Time	End Time	Total Processing Time (ms)	API Active Processing Time (ms)	Avg Processing Time (ms)	Records Processed	Records Failed	Retry Count	State Message	Status
View Request	View Result	751gk000008EM0w	9/26/2025, 3:25 AM	9/26/2025, 3:25 AM	121	96	13	22	12	0	Completed	Completed

2. Data Loader :

Supports bulk import, update, upsert, export, and deletion of large datasets.

Ideal for large-scale data operations beyond the limits of the Data Import Wizard.

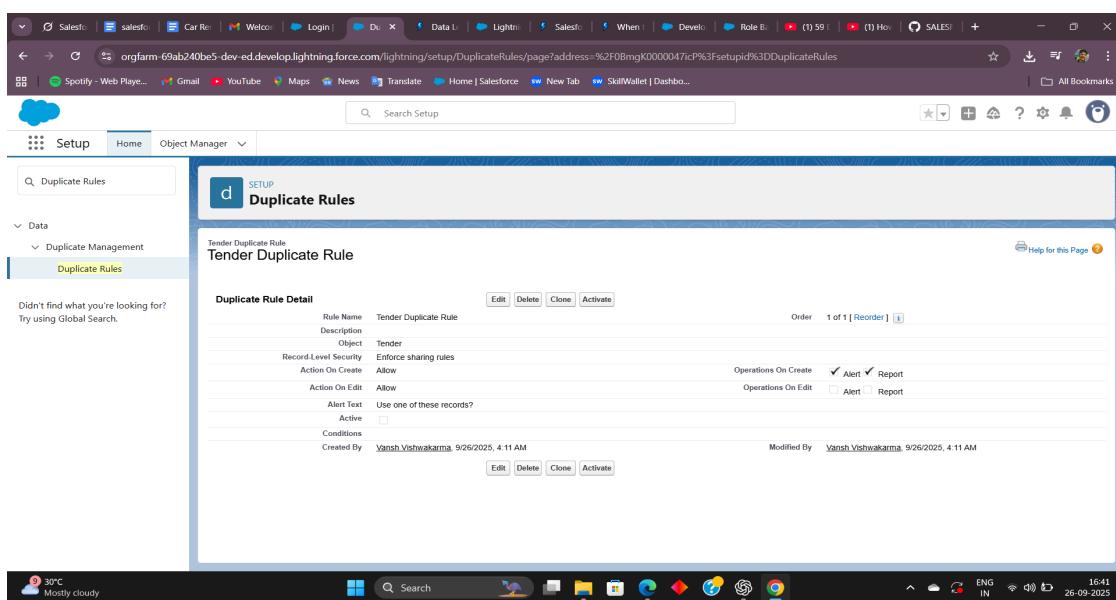


The screenshot shows a browser window with the URL developer.salesforce.com/docs/atlas.en-us.data_loader.meta/dataLoader/when_to_use_the_data_loader.htm. The page title is "Data Loader Guide". The main content is titled "Winter '20 (API version 6.0)" and contains a section "When to Use Data Loader" with several bullet points. To the right, there's a sidebar with "DEVELOPER CENTERS" (Heroku, MuchoSoft, Tableau, Commerce Cloud, Lightning Design System, Einstein, Quip), "POPULAR RESOURCES" (Documentation, Component Library, APIs, Trailhead, Single-App, Podcasts, AppExchange), and a "COMMUNITY" section (Trailblazer Community, Events and Calendar, Partner Community, Blog, Salesforce Admins, Salesforce Architects). At the bottom, there's a footer with copyright information and links to various Salesforce services like Pardot, Marketing Cloud, and AppExchange.

3. Duplicate Rules :

Prevents duplicate records and maintains data accuracy.

Ensures unique entries for Accounts, Contacts, Tenders, and Bids.

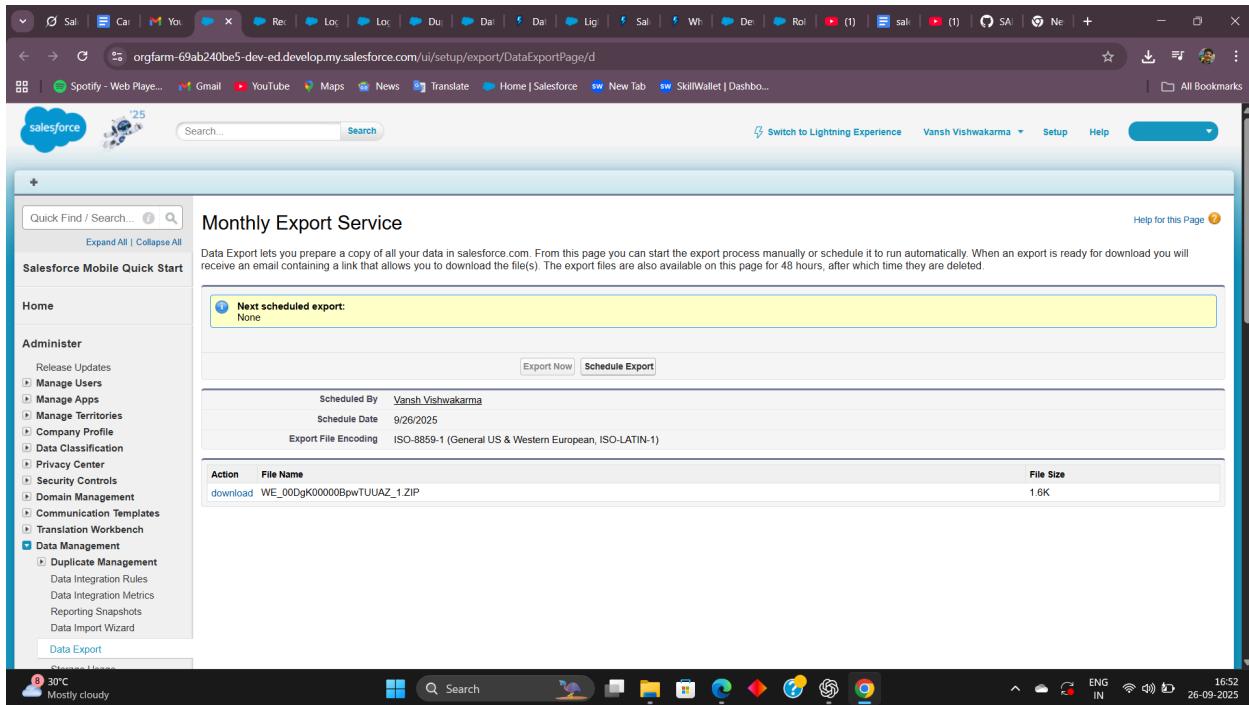


The screenshot shows a browser window with the URL orgfarm-69ab240be5-dev-ed.lightning.force.com/lightning/setup/DuplicateRules/page?address=%2F0Bmgk0000047icP%3Fsetupid%3DDuplicateRules. The page title is "Duplicate Rules". The main content shows a "Tender Duplicate Rule" with details like "Rule Name: Tender Duplicate Rule", "Object: Tender", "Record Level Security: Enforce sharing rules", and "Action On Create: Allow". It also shows sections for "Operations On Create" and "Operations On Edit", both with "Alert" and "Report" checkboxes checked. At the bottom, it shows "Created By: Vansh Vishwakarma, 9/26/2025, 4:11 AM" and "Modified By: Vansh Vishwakarma, 9/26/2025, 4:11 AM". The interface includes a left sidebar with "Data" and "Duplicate Management" sections, and a bottom navigation bar with various icons.

4. Data Export & Backup :

Provides regular backup of Salesforce data.

Helps recover data in case of accidental deletion or corruption.



The screenshot shows the 'Data Export' page in the Salesforce setup. The main content area displays a table of scheduled exports. One entry is visible:

Action	File Name	File Size
download	WE_00DgK00000BpwTUUAZ_1.ZIP	1.6K

The sidebar on the left shows the 'Data Management' section under 'Administer'. The 'Data Export' option is highlighted. The top navigation bar includes links for various Salesforce features like Sales, Case, YouTube, etc.

5. ANT Migration Tool :

Enables programmatic metadata deployment using XML descriptors.

Supports version control and automated deployment pipelines.

Already done in step 4

6. VS Code & Salesforce CLI (SFDC) :

Modern development environment for Salesforce.

Facilitates metadata management, scratch orgs, code deployment, and continuous integration.

```

1 import { LightningElement } from 'lwc';
2 public with sharing class TenderContractController {
3     @AuraEnabled(cacheable=true)
4     public static List<Tender__c> getActiveTenders() {
5         return [SELECT Id, Tender__Name__c, Status__c, End__Date__c
6                 FROM Tender__c
7                 WHERE Status__c = 'Open' AND End__Date__c >= TODAY
8                 LIMIT 10];
9     }
10
11     @AuraEnabled(cacheable=true)
12     public static List<Contract__c> getSignedContracts() {
13         return [SELECT Id, Contract_Value__c, Status__c, Tender__r.Tender__Name__c
14                 FROM Contract__c
15                 WHERE Status__c = 'Signed'
16                 LIMIT 10];
17 }

```

The error is caused by using `end` as a variable name in your Apex class. `end` is a reserved keyword in Apex, so you cannot use it as a variable or property name.

How to fix:
Rename the property and parameter from `end` to something like `enddate`.

Corrected code:

```

public class Contract {
    public String c
    public Date sta
    public Date enc

    public Contract {
        contractNam
        startDate =
    }
}

```

Phase 9: Reporting, Dashboards & Security Review

👉 Goal: Monitor business & secure data.

1. Reports :

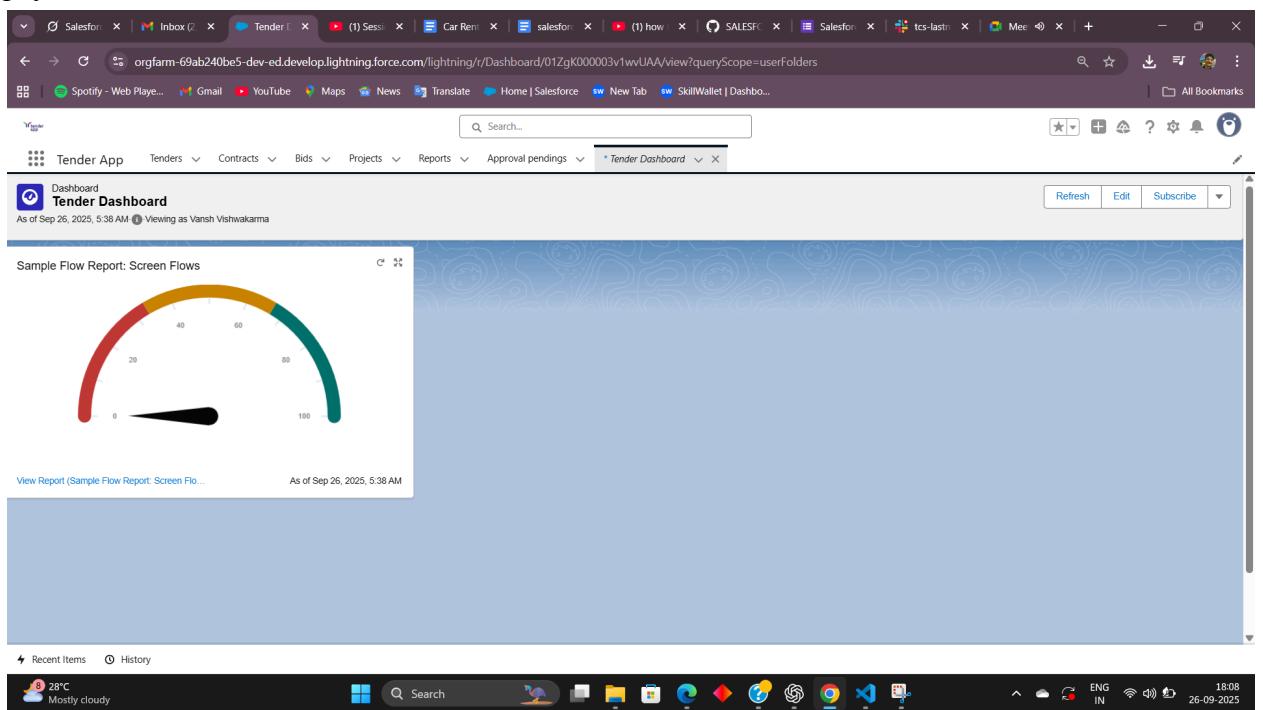
- Tabular, Summary, Matrix, Joined reports provide different ways to view and analyze Salesforce data.
- Reports can track Tenders, Bids, Contracts, Payments, and their status or value.

2. Report Types:

- Define which objects and related records can be included in reports.
- Custom report types (e.g., Tender with Bids) allow detailed reporting on multiple related objects.

3. Dashboards:

- Visual representation of reports with charts, graphs, and tables.
- Provides at-a-glance monitoring of tender lifecycle, bid status, contract progress, and payments.



5. Dynamic Dashboards :

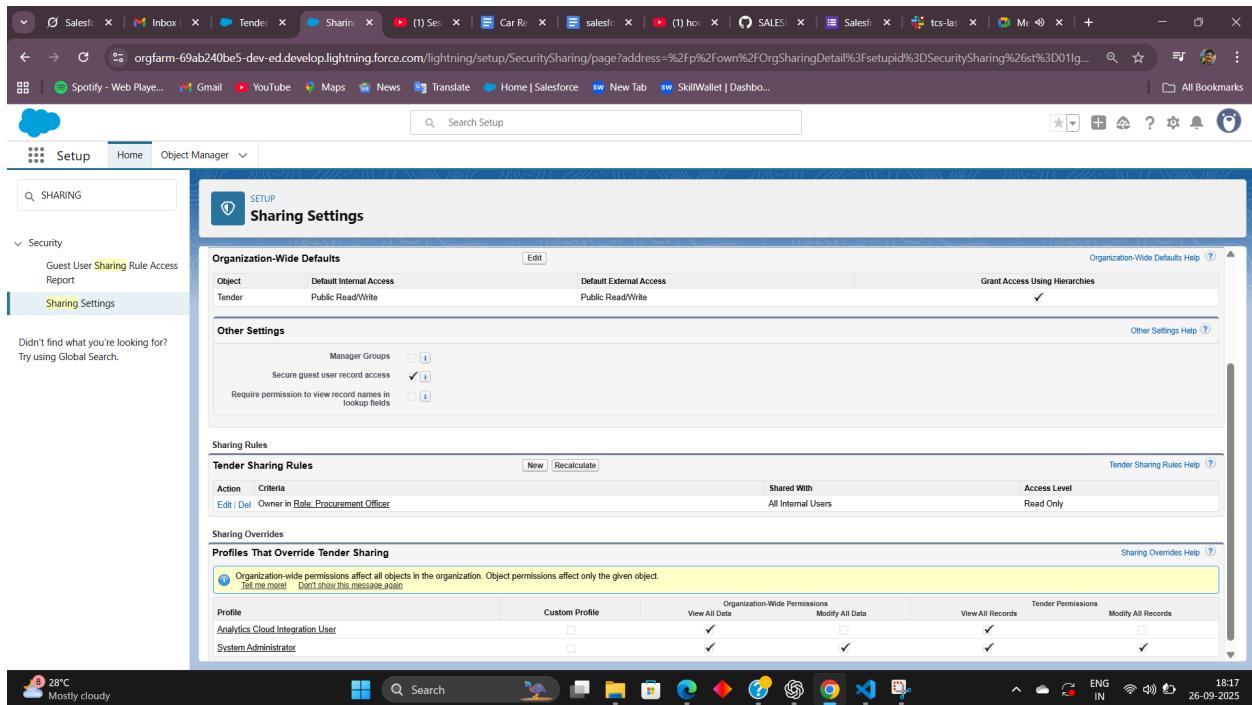
- Display data based on the viewing user's role or access level.
- Ensures managers, finance officers, and contractors see only relevant information.

The screenshot shows a Salesforce Lightning interface with a top navigation bar containing various tabs like 'Tender App', 'Tenders', 'Contracts', 'Bids', 'Projects', 'Reports', 'Approval pending', and 'Tender Dashboard'. Below the navigation is a search bar and a toolbar with icons for 'Widget', 'Filter', 'Save', and 'Done'. The main content area displays a table titled 'New Tenders Report' with columns: Tender, Tender Name, Bids, Contract, Start Date, and End Date. The table lists several entries, each starting with 'a00gk00000Mtpv'. At the bottom of the table is a link 'View Report (New Tenders Report)'. The status bar at the bottom shows weather (28°C, Mostly cloudy), system icons, and the date/time (26-09-2025, 18:12).

Tender	Tender Name	Bids	Contract	Start Date	End Date
a00gk00000MtpvA	-	-	-	-	-
a00gk00000MtpvC	-	-	-	-	-
a00gk00000MtpvD	-	-	-	-	-
a00gk00000MtpvF	-	-	-	-	-
a00gk00000MtpvW	-	-	-	-	-
a00gk00000MtpvX	-	-	-	-	-
a00gk00000MtpvZ	-	-	-	-	-

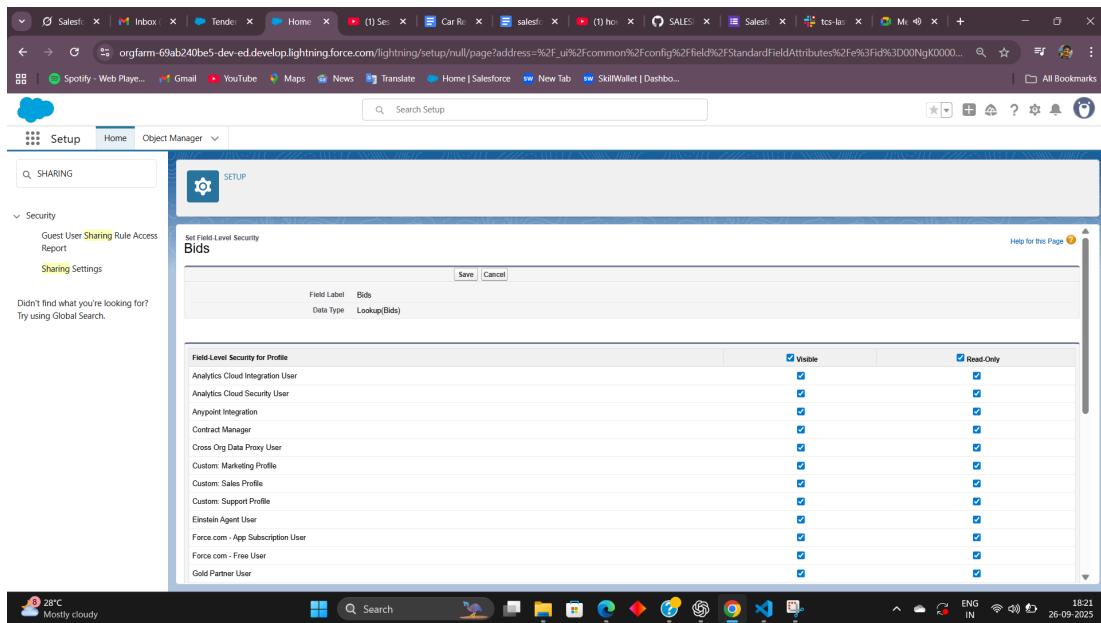
6. Sharing Settings :

- Control record-level access using **Org-Wide Defaults (OWD)**, role hierarchy, and sharing rules.
- Ensures sensitive data like contract values are visible only to authorized users.



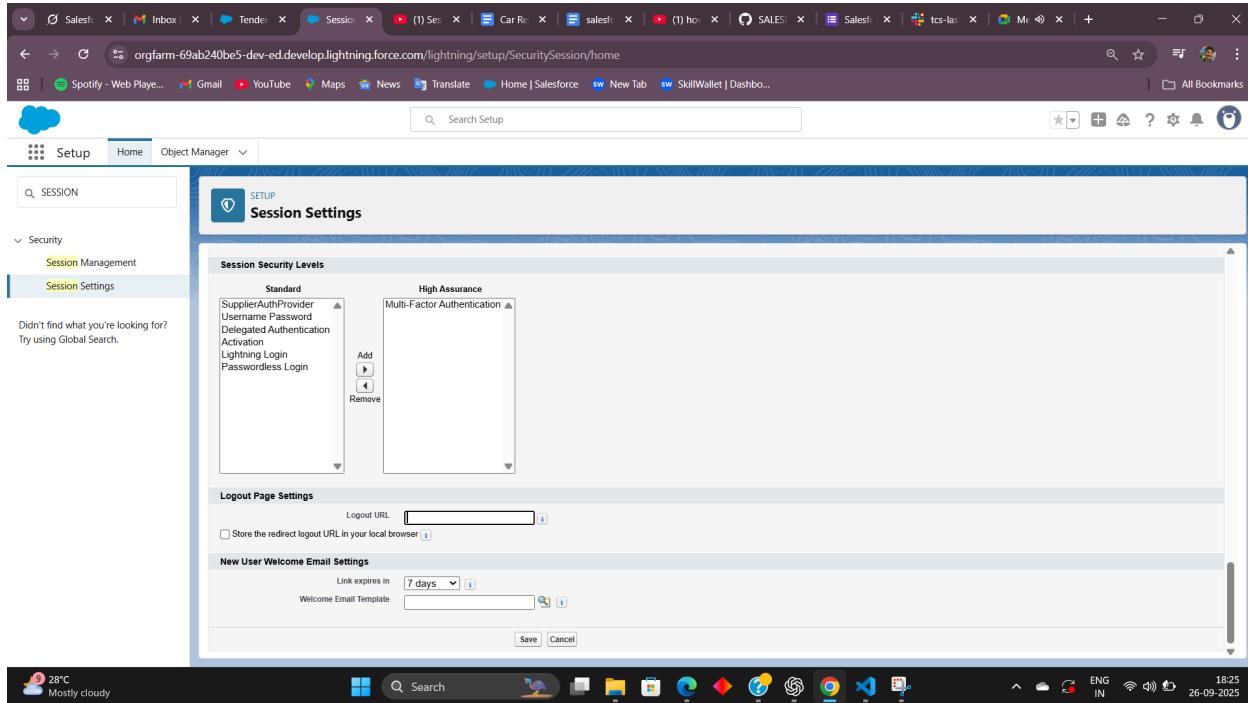
7. Field-Level Security (FLS) :

- Controls visibility and editability of individual fields for different profiles.
- Protects confidential data while providing necessary access to relevant users.



8. Session Settings :

- Manage session duration, security policies, and login behavior to maintain secure access.



9. Login IP Ranges :

- Restrict access to Salesforce org based on trusted IP addresses.
- Enhances security by limiting logins to office or approved locations.

Profiles

All Profiles | Edit | Delete | Create New View

Action	Profile Name	User License	Custom
Edit Clone	Analytics Cloud Integration User	Analytics Cloud Integration User	<input type="checkbox"/>
Edit Clone	Analytics Cloud Security User	Analytics Cloud Integration User	<input type="checkbox"/>
Edit Clone	Apoint Integration	Identity	<input type="checkbox"/>
Edit Clone	Authenticated Website	Authenticated Website	<input type="checkbox"/>
Edit Clone	Authenticated Website	Authenticated Website	<input type="checkbox"/>
Edit Del ...	B2B Reordering Portal Buyer Profile	External Apps Login	<input checked="" type="checkbox"/>
Edit Clone	Chatter External User	Chatter External	<input type="checkbox"/>
Edit Clone	Chatter Free User	Chatter Free	<input type="checkbox"/>
Edit Clone	Chatter Moderator User	Chatter Free	<input type="checkbox"/>
Edit Clone	Contract Manager	Salesforce	<input type="checkbox"/>
Edit Clone	Cross Org Data Proxy User	XOrg Proxy User	<input type="checkbox"/>
Edit Del ...	Custom Marketing Profile	Salesforce	<input checked="" type="checkbox"/>
Edit Del ...	Custom Sales Profile	Salesforce	<input checked="" type="checkbox"/>
Edit Del ...	Custom Support Profile	Salesforce	<input checked="" type="checkbox"/>
Edit Clone	Customer Community Login User	Customer Community Login	<input type="checkbox"/>
Edit Clone	Customer Community Plus Login	Customer Community Plus Login	<input type="checkbox"/>

1-25 of 45 | 0 Selected | Page 1 of 2

10 . Audit Trail :

- Tracks configuration and metadata changes in Salesforce setup.
- Provides accountability and supports compliance audits.

View Setup Audit Trail

The last 20 entries for your organization are listed below. You can download your organization's setup audit trail for the last six months (Excel .csv file).

Date	User	Source Namespace Prefix	Action	Section	Delegate User
9/26/2025, 5:54:56 AM	varshishwakarma657237@agentforce.com		Session Security Level for Multi-Factor Authentication was set to High Assurance	Session Settings	
9/26/2025, 5:54:56 AM	varshishwakarma657237@agentforce.com		Session Security Level for SupplierAuthProvider was set to Standard	Session Settings	
9/26/2025, 5:54:56 AM	varshishwakarma657237@agentforce.com		Session Security Level for Passwordless Login was set to Standard	Session Settings	
9/26/2025, 5:54:56 AM	varshishwakarma657237@agentforce.com		Session Security Level for Lightning Login was set to Standard	Session Settings	
9/26/2025, 5:54:56 AM	varshishwakarma657237@agentforce.com		Session Security Level for Activation was set to Standard	Session Settings	
9/26/2025, 5:54:56 AM	varshishwakarma657237@agentforce.com		Session Security Level for Delegated Authentication was set to Standard	Session Settings	
9/26/2025, 5:54:56 AM	varshishwakarma657237@agentforce.com		Session Security Level for Username Password was set to Standard	Session Settings	
9/26/2025, 5:51:25 AM	varshishwakarma657237@agentforce.com		Changed profile Work.com Only User: field-level security for Tender: Bids was changed from 2 to 1	Manage Users	
9/26/2025, 5:51:25 AM	varshishwakarma657237@agentforce.com		Changed profile Silver Partner User: field-level security for Tender: Bids was changed from 2 to 1	Manage Users	
9/26/2025, 5:51:25 AM	varshishwakarma657237@agentforce.com		Changed profile Salesforce API Only System Integrations: field-level security for Tender: Bids was changed from Read/Write to Read Only	Manage Users	
9/26/2025, 5:51:25 AM	varshishwakarma657237@agentforce.com		Changed profile Read Only: field-level security for Tender: Bids was changed from Read/Write to Read Only	Manage Users	
9/26/2025, 5:51:25 AM	varshishwakarma657237@agentforce.com		Changed profile Contract Manager: field-level security for Tender: Bids was changed from Read/Write to Read Only	Manage Users	

18:29 26-09-2025

Phase 10: Final Presentation & Demo Day

👉 Goal: Wrap it up like a real project delivery.

1. Pitch Presentation :

Present the project idea, objectives, and outcomes to stakeholders.

2. **Demo Walkthrough :**

Showcase the working solution, highlighting key features and functionalities.

3. **Feedback Collection :**

Gather input and suggestions from audience or evaluators for improvement.

4 . Handoff Documentation:

Provide comprehensive project documents, including configurations, reports, and manuals.

5. LinkedIn/Portfolio Project Showcase :

Share the project publicly on professional platforms to demonstrate skills and experience.