

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.

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.

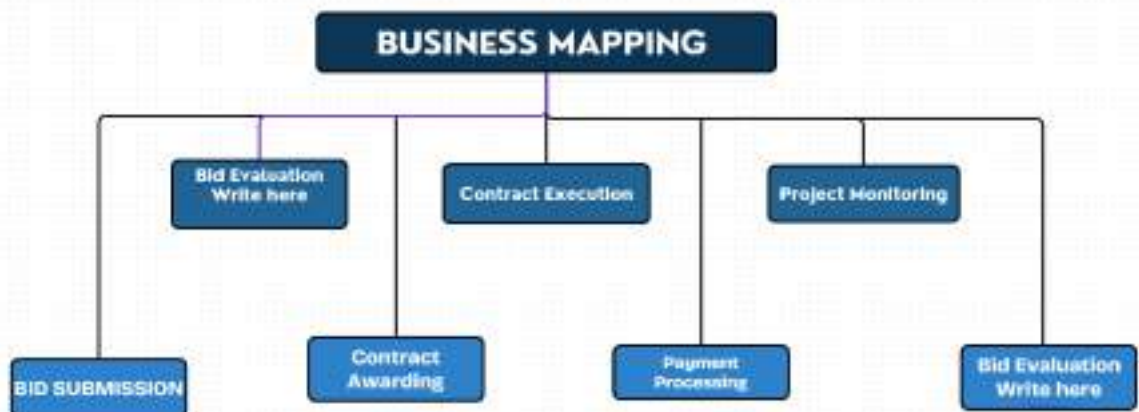
2. Requirement Gathering : Talk to stakeholders (contractor , engineers , government companies etc).

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 :

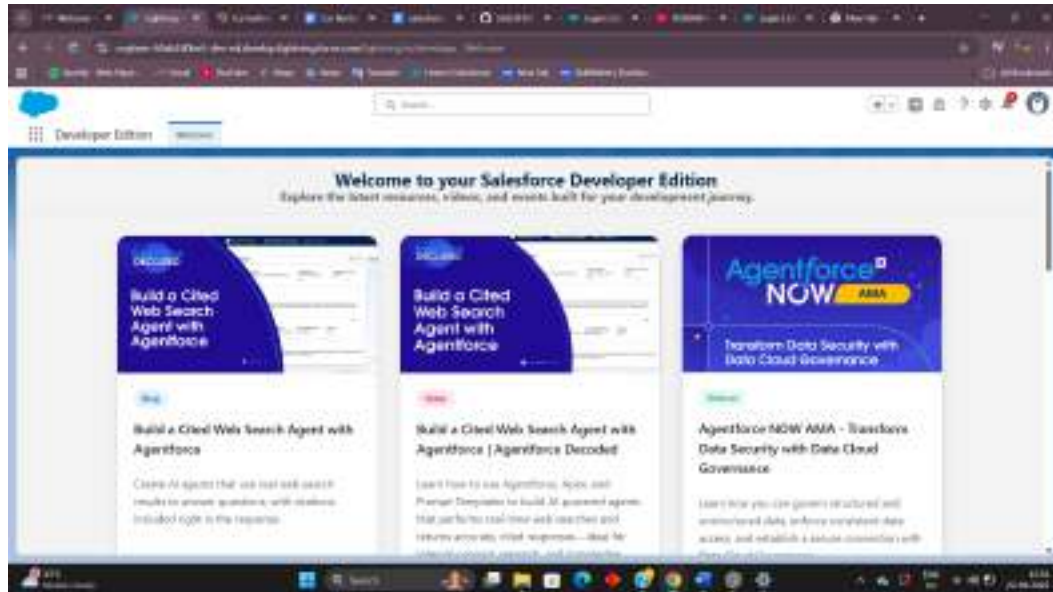
- DocuSign for Salesforce: For digital signing of contracts
- Conga Composer: To generate tender documents and reports
- Salesforce Maps: For visualizing project locations
- Accounting Seed / FinancialForce: For financial tracking and integration
- Formstack / Nintex: To create digital forms for bid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

Phase 2 : Org Setup & Configuration

👉 Goal: Prepare Salesforce environment.

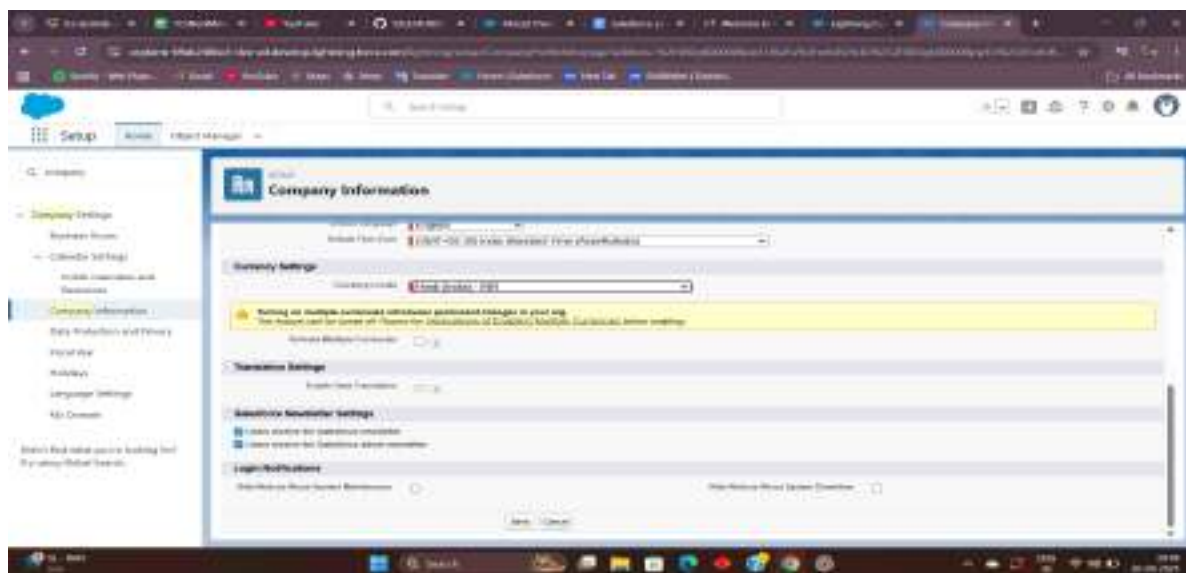
1. Salesforce Editions :

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



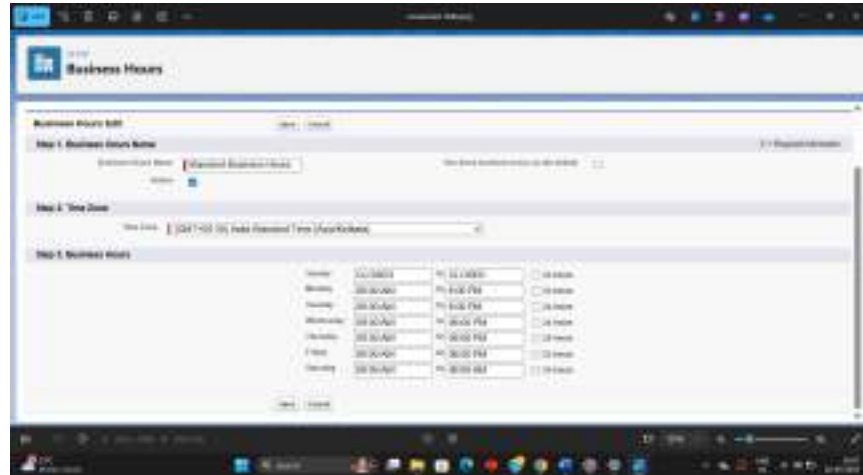
2 . Company Profile Setup :

- 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)



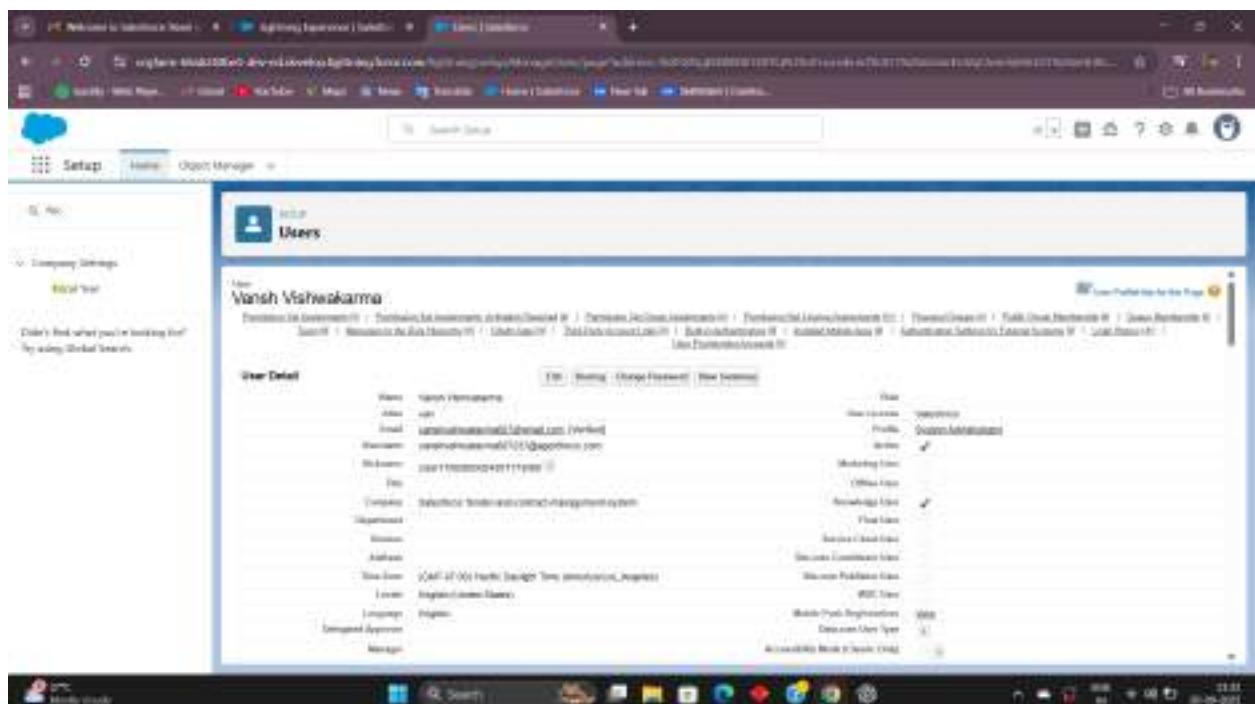
3. Business Hours & Holidays :

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



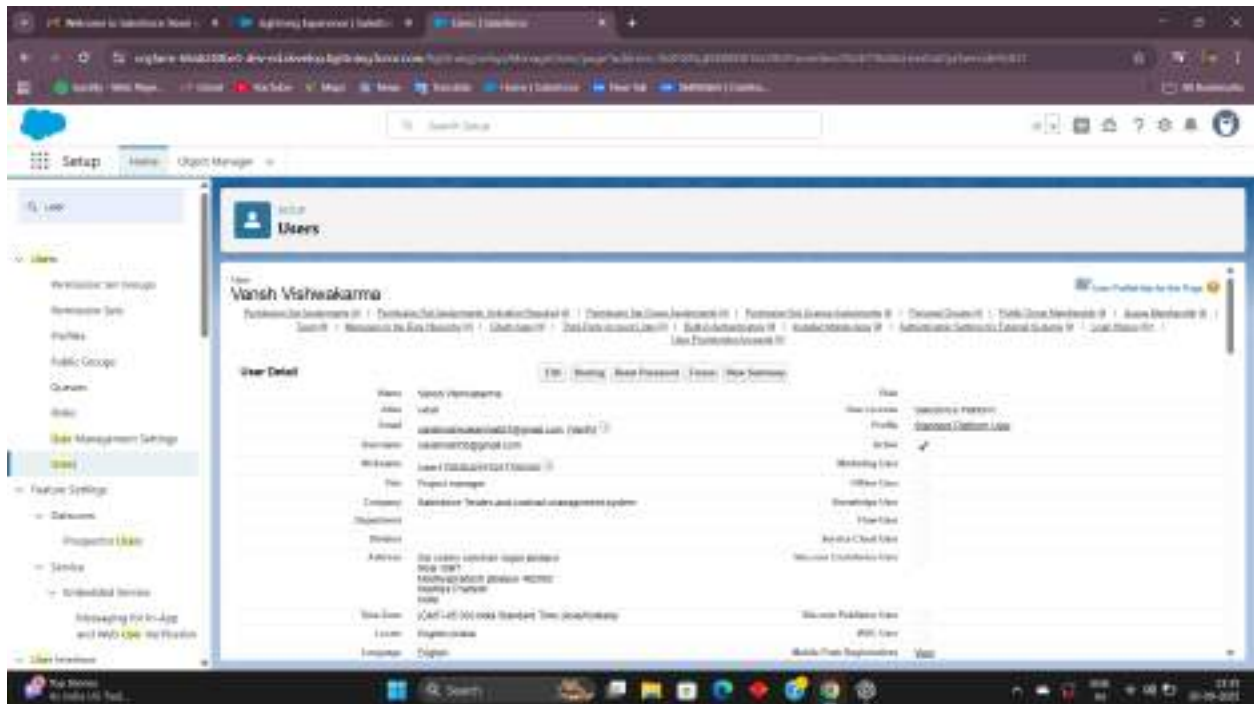
4. Fiscal Year Settings :

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



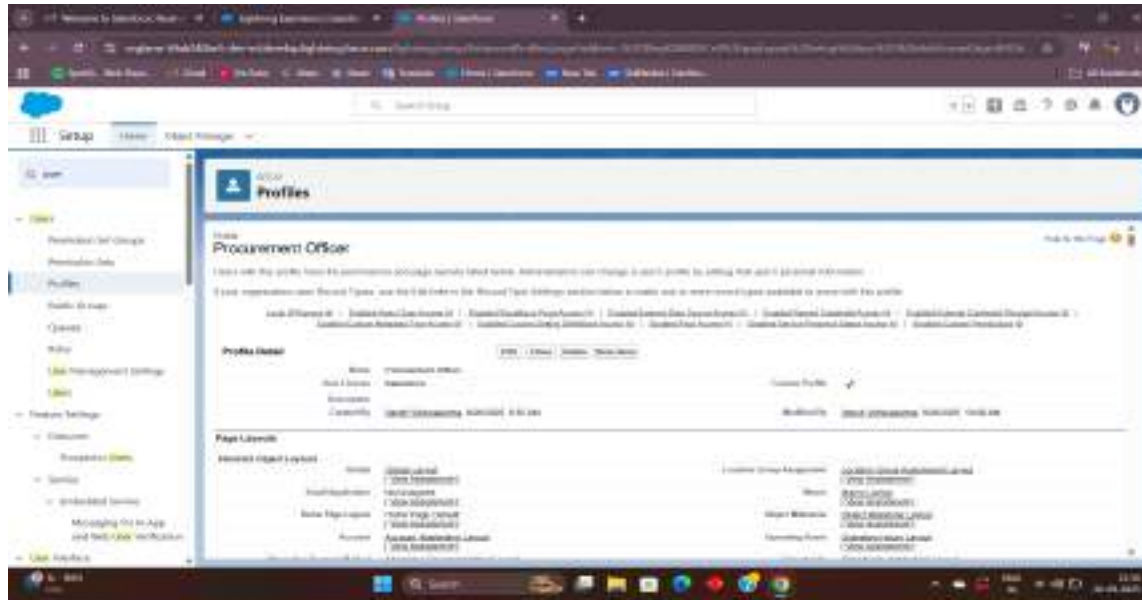
5. User Setup & Licenses :

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



6 . Profiles :

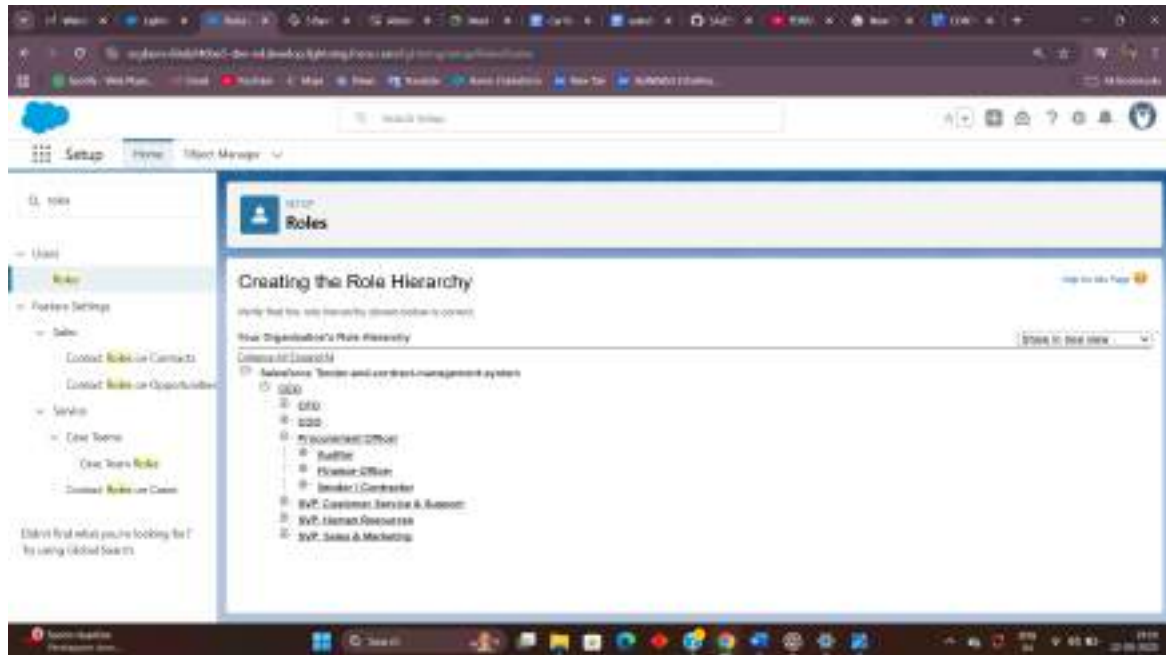
- Standard Profiles: Admin, Standard User
- Custom Profile: Procurement User (restricted to Tender objects)



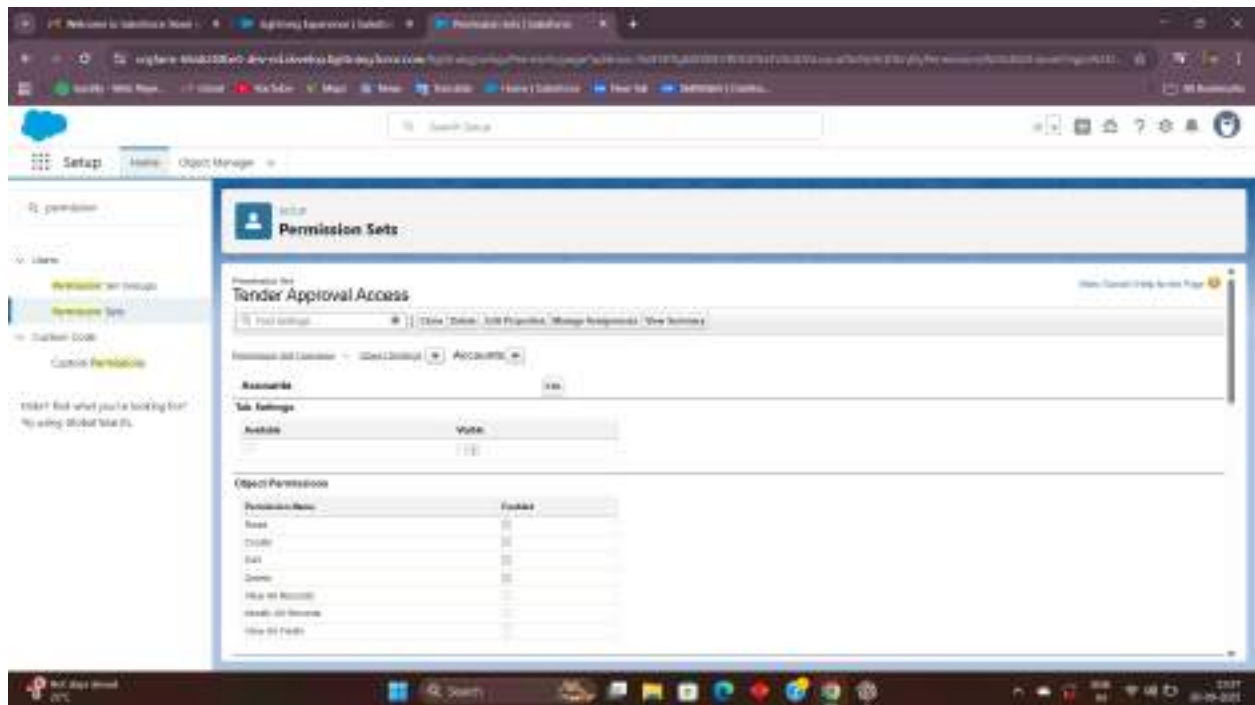
7. Roles :

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



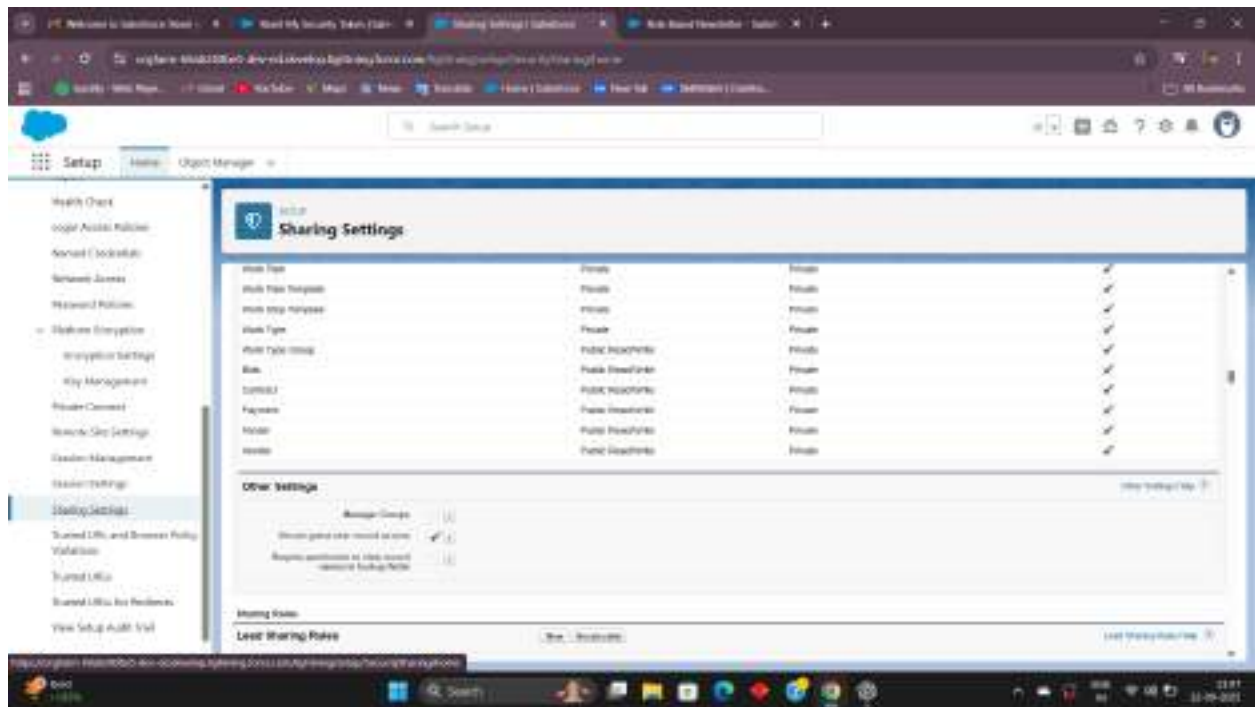


Permission Sets :



9. OWD (Org-Wide Defaults) :

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



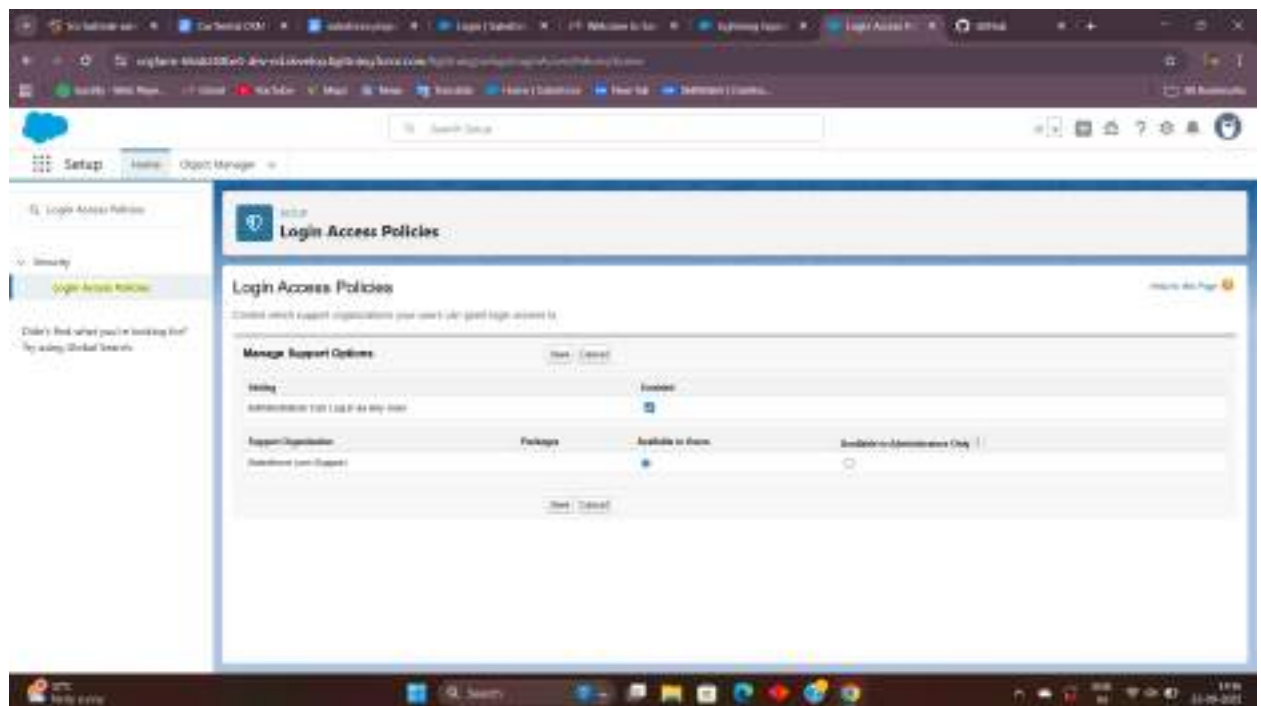
10. Sharing Rules :

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



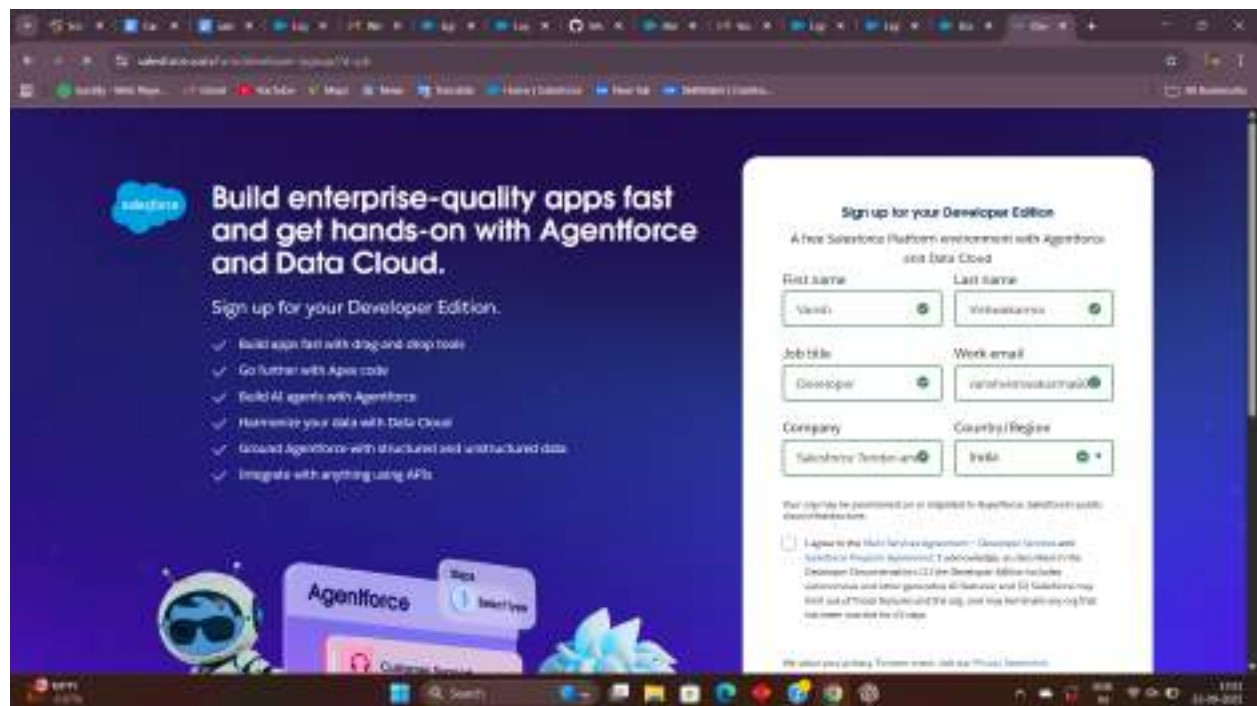
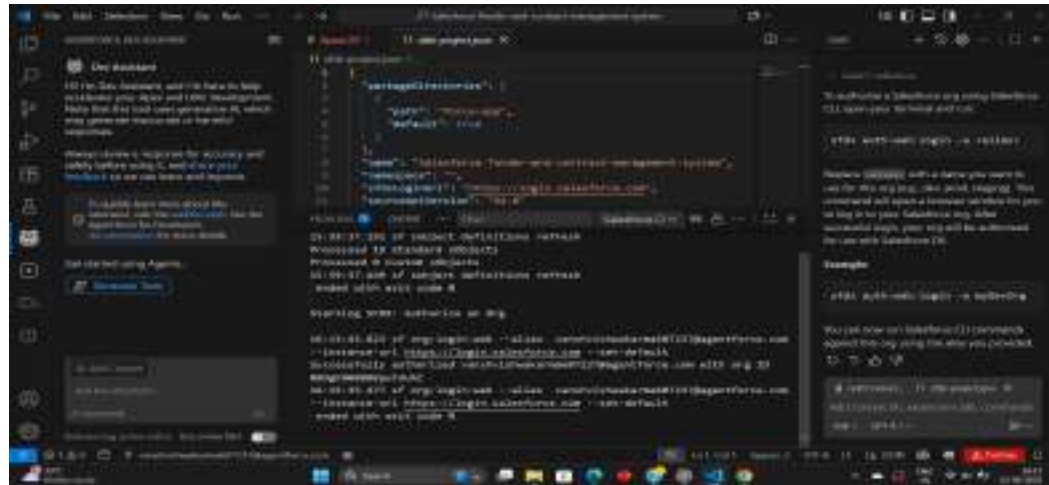
11. Login Access Policies :

- 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



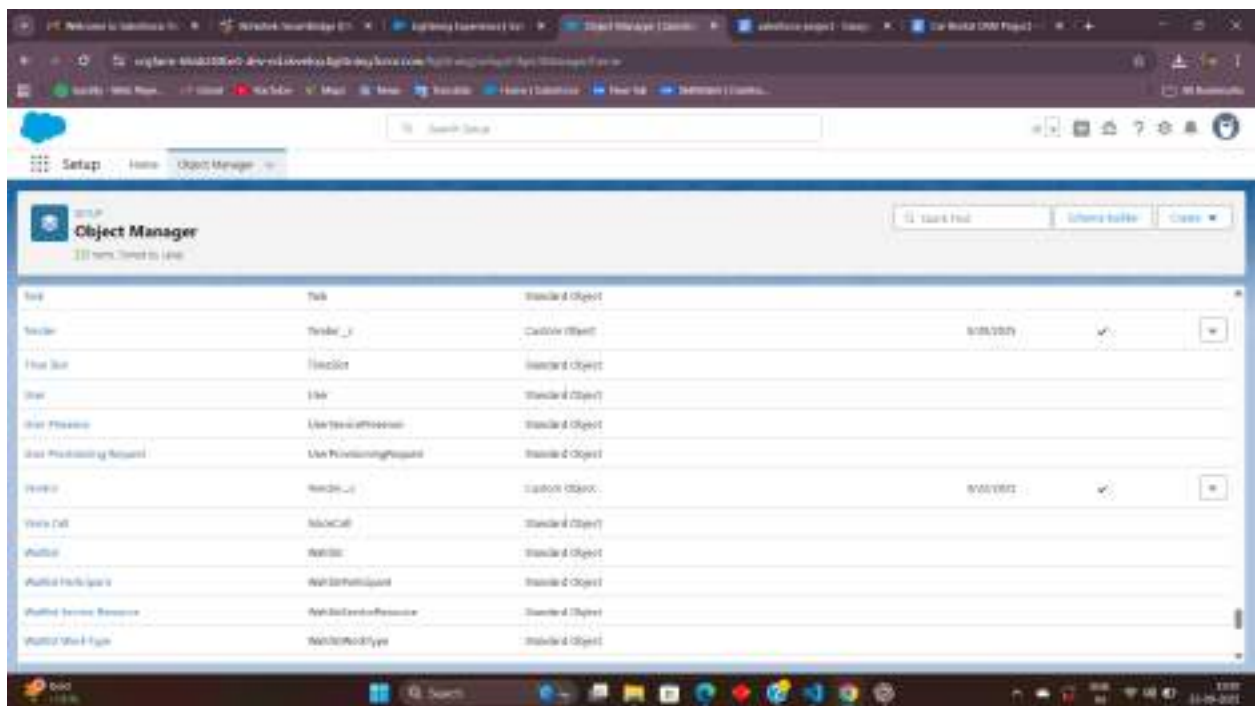
Phase 3: Data Modeling & Relationships

👉 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.

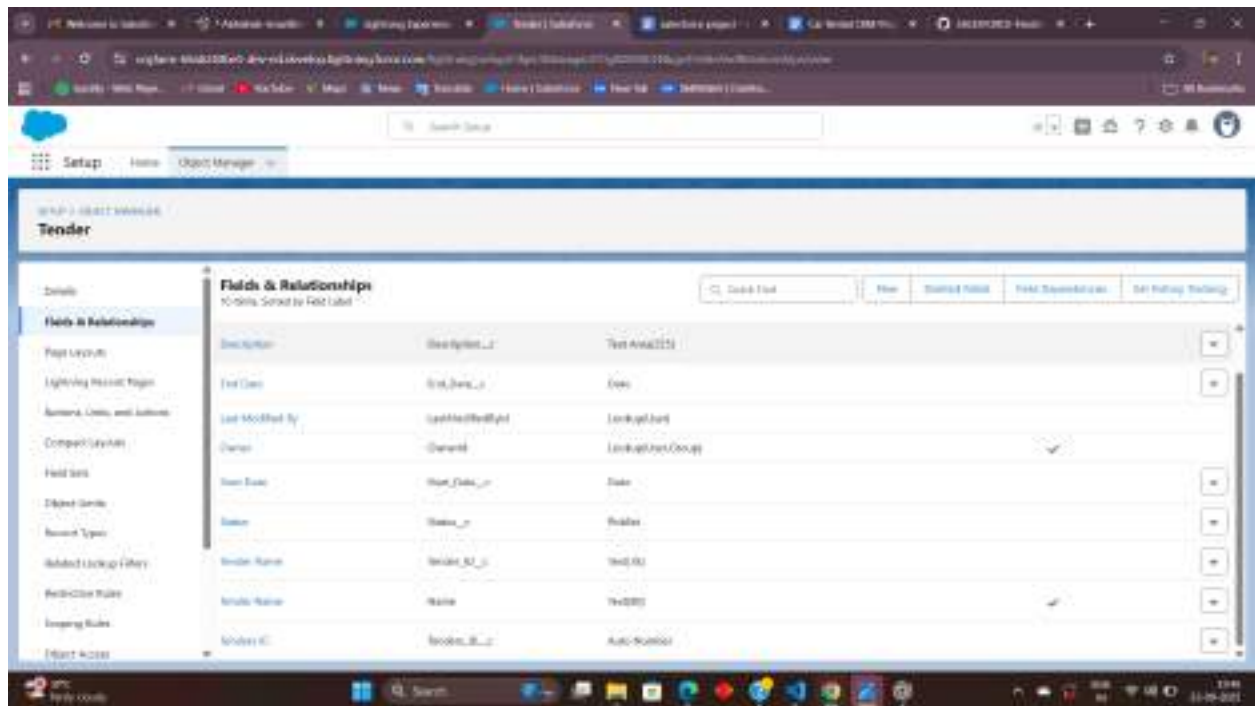


2. Fields :

- 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**



- **Bid__c**

Fields & Relationships
12 fields. Sorted by Field Label

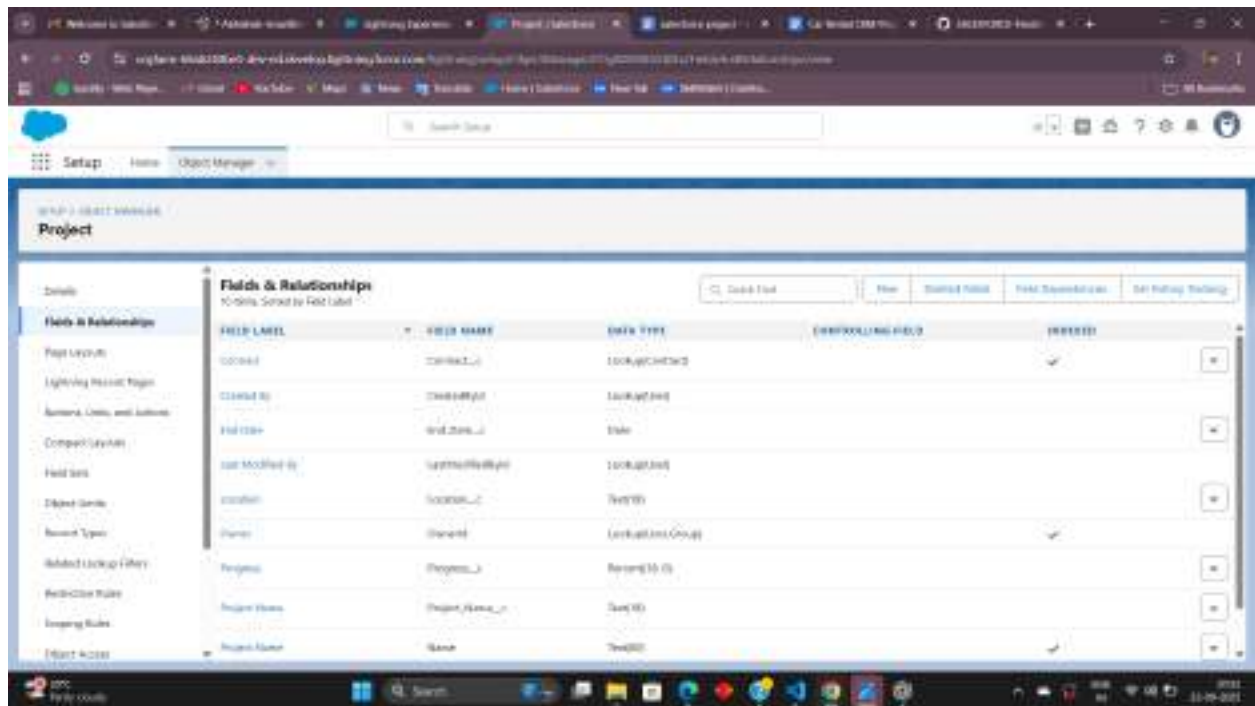
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Bid Amount	Bid_Amount__c	Currency (18, 0)		
Bid ID	Bid_ID__c	Auto Number		
Bid Name	Name	Text (50)		
Created By	CreatedBy	Lookup (User)		
Last Modified By	LastModifiedBy	Lookup (User)		
Owner	OwnerId	Lookup (User/Group)		
Remarks	Remarks__c	Text Area (255)		
Status	Status__c	Picklist		
Submitted Date	SubmittedDate__c	Date/Time		

○ **Contract__c :**

Fields & Relationships
11 fields. Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Assigned By	AssignedTo__c	Lookup (User)		
Contract ID	Contract_ID__c	Auto Number		
Contract Name	Name	Text (50)		
Contract Value	ContractValue__c	Currency (18, 0)		
Created By	CreatedBy	Lookup (User)		
End Date	EndDate__c	Date		
Last Modified By	LastModifiedBy	Lookup (User)		
Owner	OwnerId	Lookup (User/Group)		
Start Date	StartDate__c	Date		

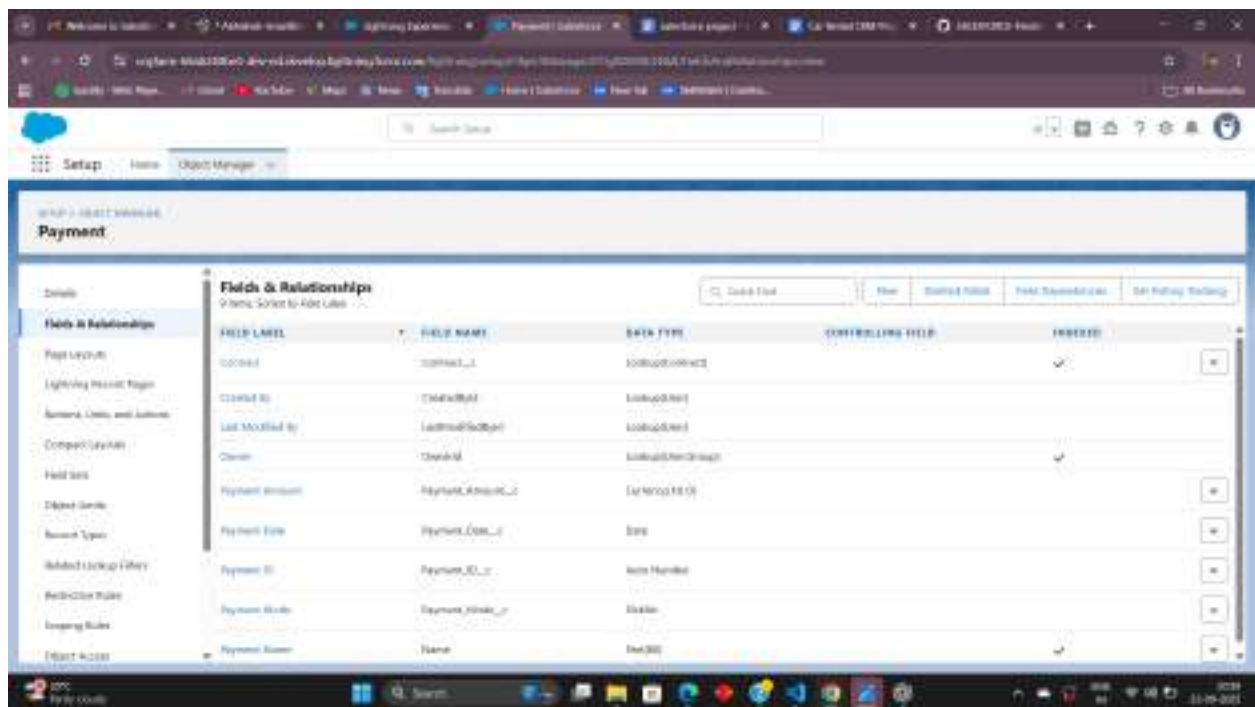
- Project_c :



The screenshot shows the Salesforce 'Project' object configuration page. The 'Fields & Relationships' tab is active, displaying a table of fields and their relationships.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Contract	Contract__c	Lookup (Contract)		✓
Contract ID	ContractId__c	Lookup (ID)		
End Date	EndDate__c	Date		
Last Modified By	LastModifiedDate__c	Lookup (User)		
Location	Location__c	Text (50)		
Name	Name__c	Lookup (Name Group)		✓
Progress	Progress__c	Percent (0-100)		
Project Owner	ProjectOwner__c	User (ID)		
Project Status	ProjectStatus__c	Text (50)		
Project Number	Number	Text (20)		✓

- Payment_c :

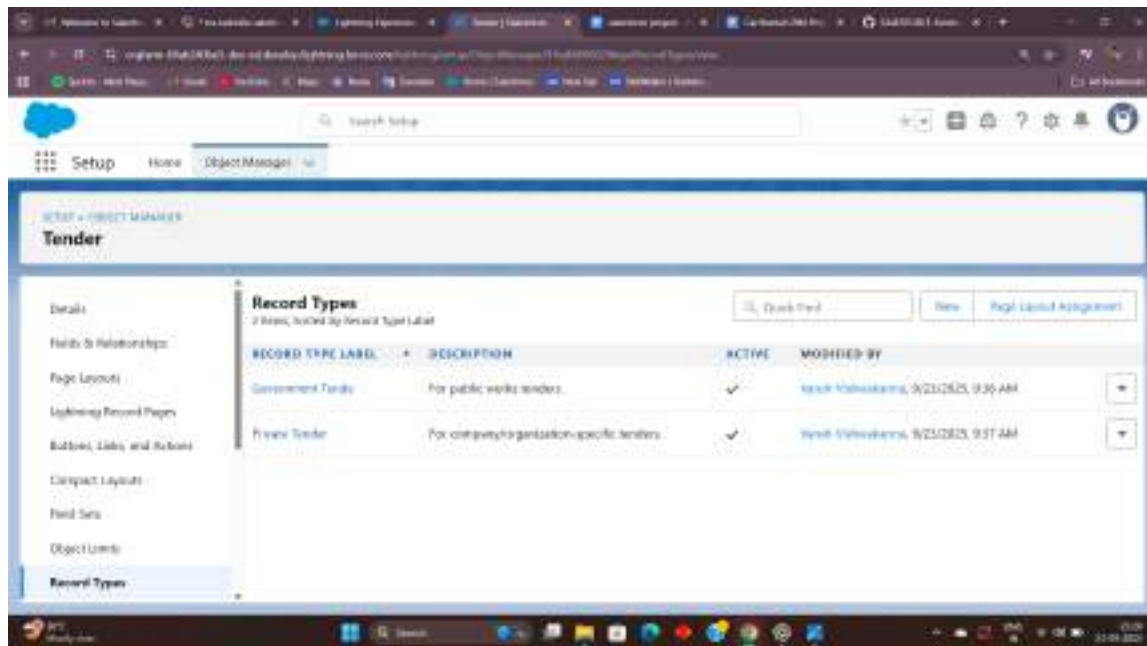


The screenshot shows the Salesforce 'Payment' object configuration page. The 'Fields & Relationships' tab is active, displaying a table of fields and their relationships.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Contract	Contract__c	Lookup (Contract)		✓
Contract ID	ContractId__c	Lookup (ID)		
Last Modified By	LastModifiedDate__c	Lookup (User)		
Name	Name__c	Lookup (Name Group)		✓
Payment Amount	PaymentAmount__c	Currency (18-10)		
Payment Date	PaymentDate__c	Date		
Payment ID	PaymentID__c	Auto Number		
Payment Owner	PaymentOwner__c	User (ID)		
Payment Status	Status	Text (20)		✓

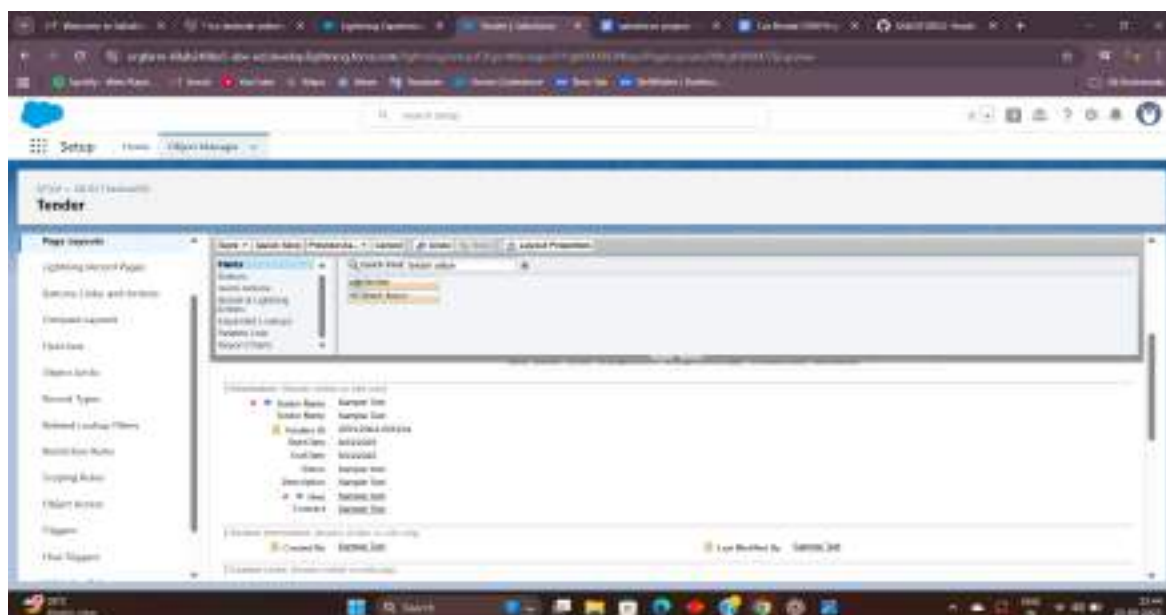
3. Record Types :

- Tender__c → Govt Tender, Private Tender
- Contract__c → Short-Term, Long-Term



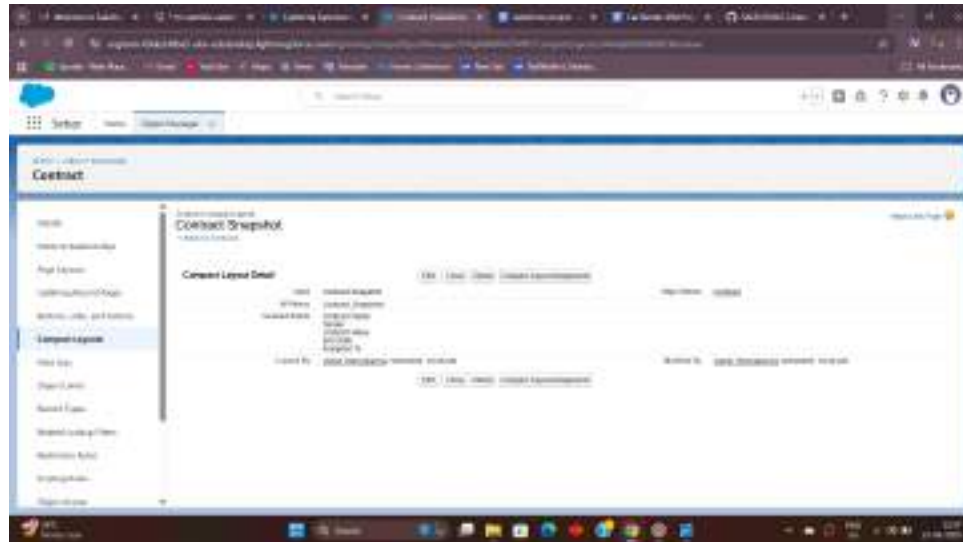
4. Page Layouts :

- 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)



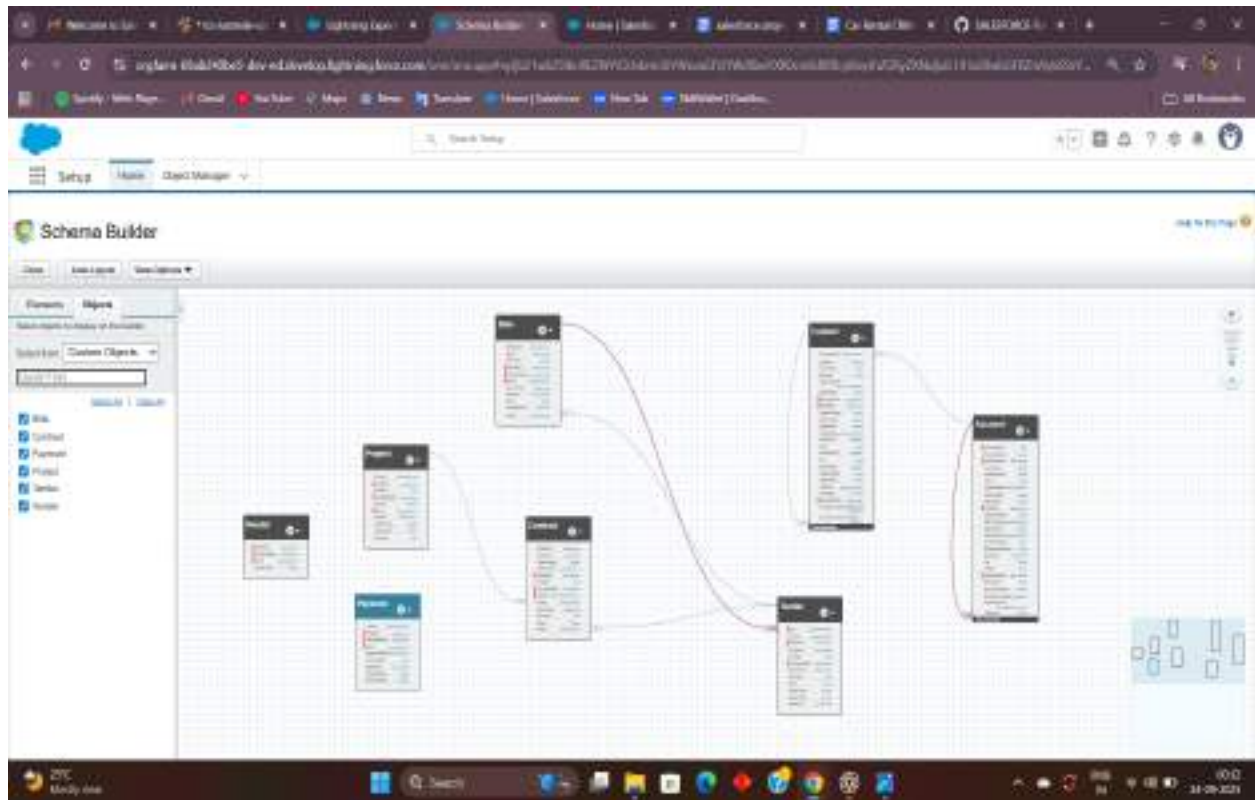
5. Compact Layouts :

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



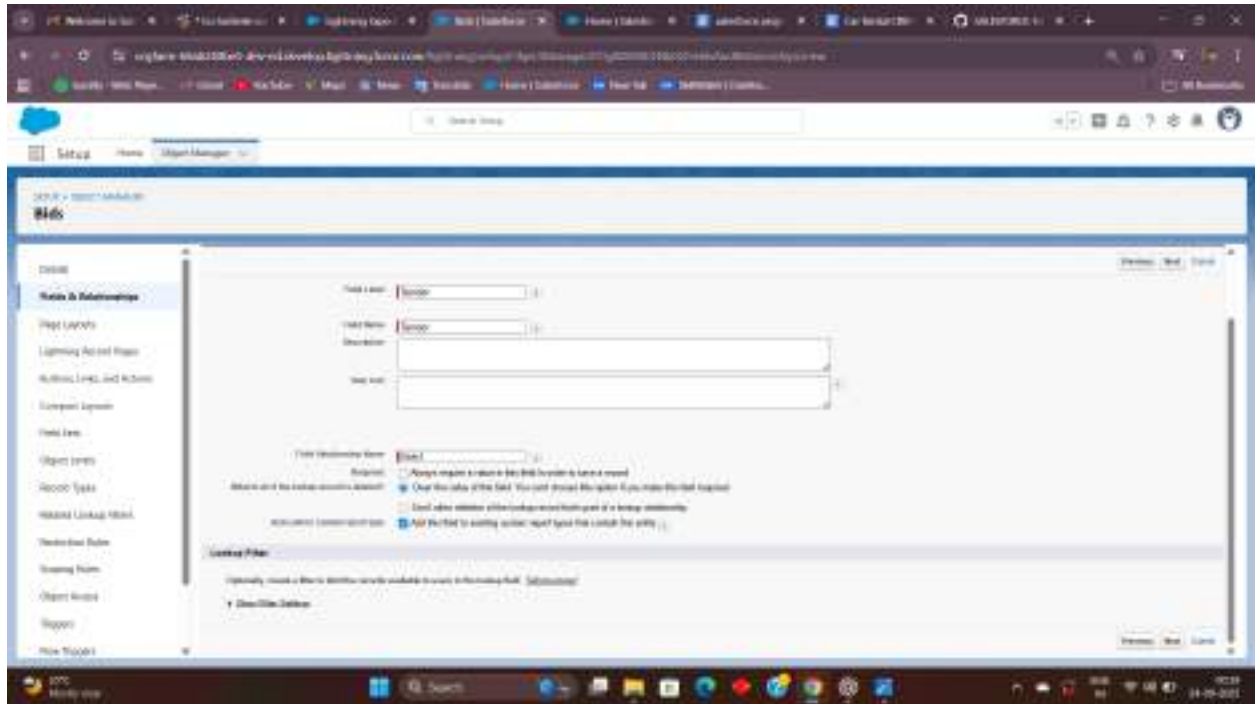
6. Schema Builder :

- 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.

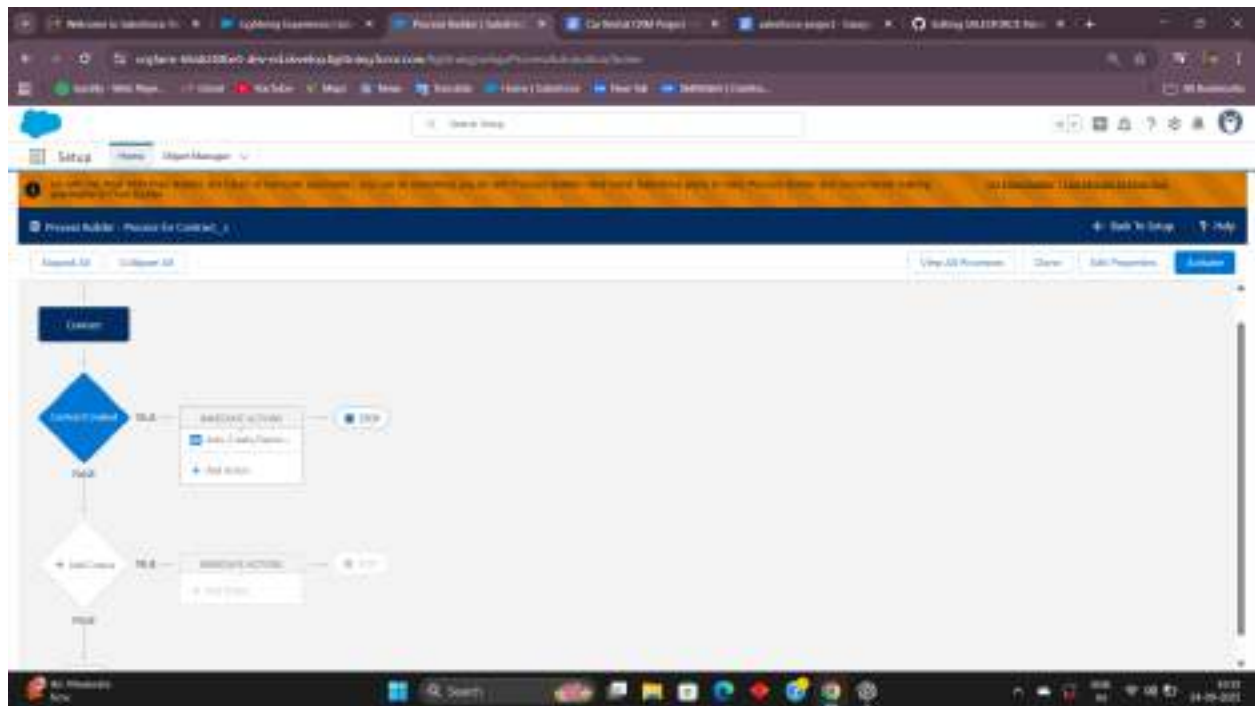
1. Validation Rules :

- 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."*

-

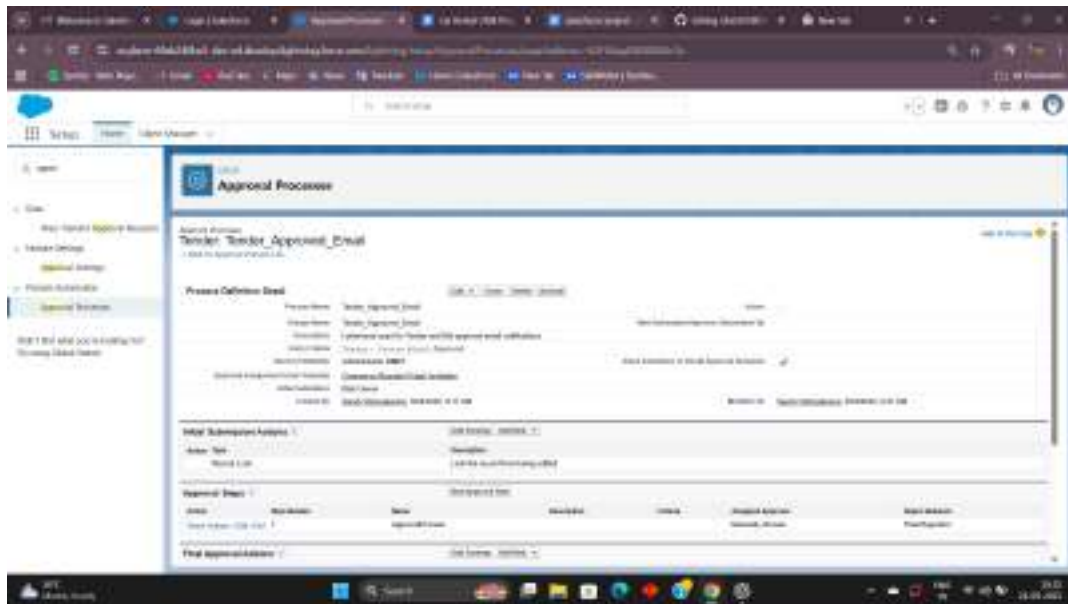
- Tender Object (Tender __c):
When the Status of a Tender becomes Approved, the system will automatically send an Email Alert to the Procurement Officer.
- Bid Object (Bid __c):
When the Status of a Bid is set to Submitted, a Task will be generated to notify the Project Manager about the new bid.
- Contract 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.

- 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.



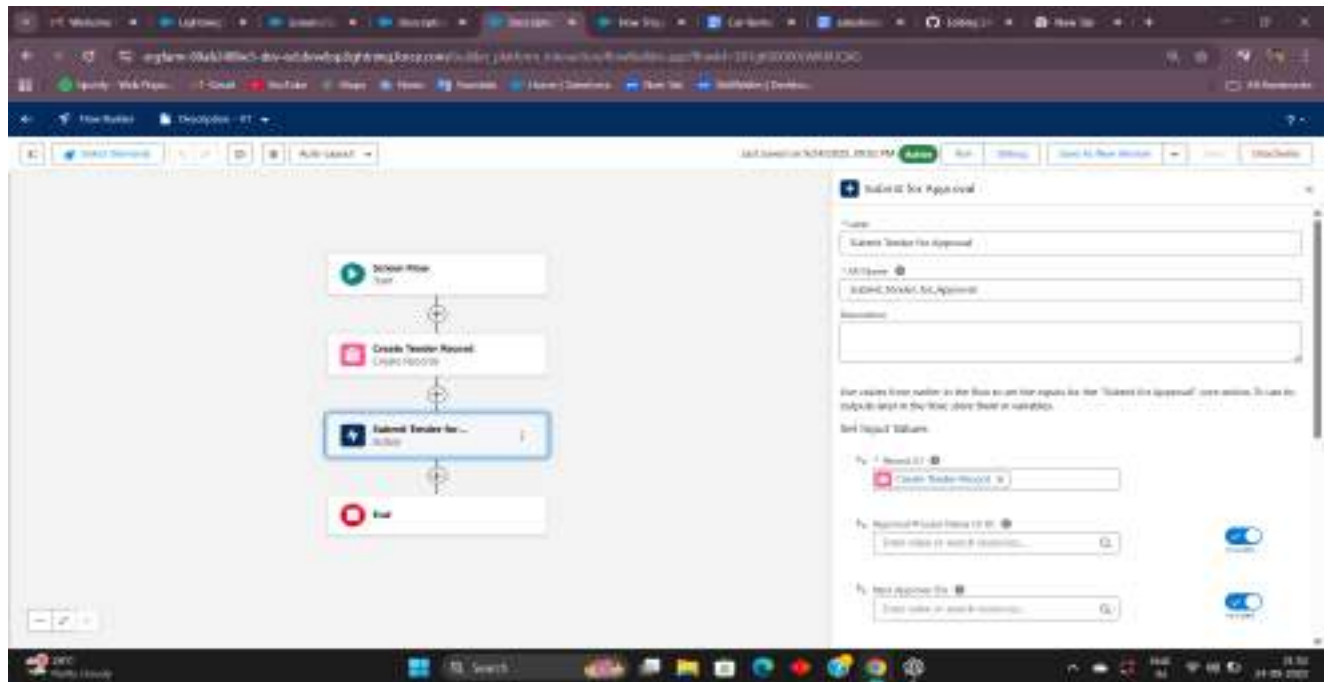
4. Approval Process :

- 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



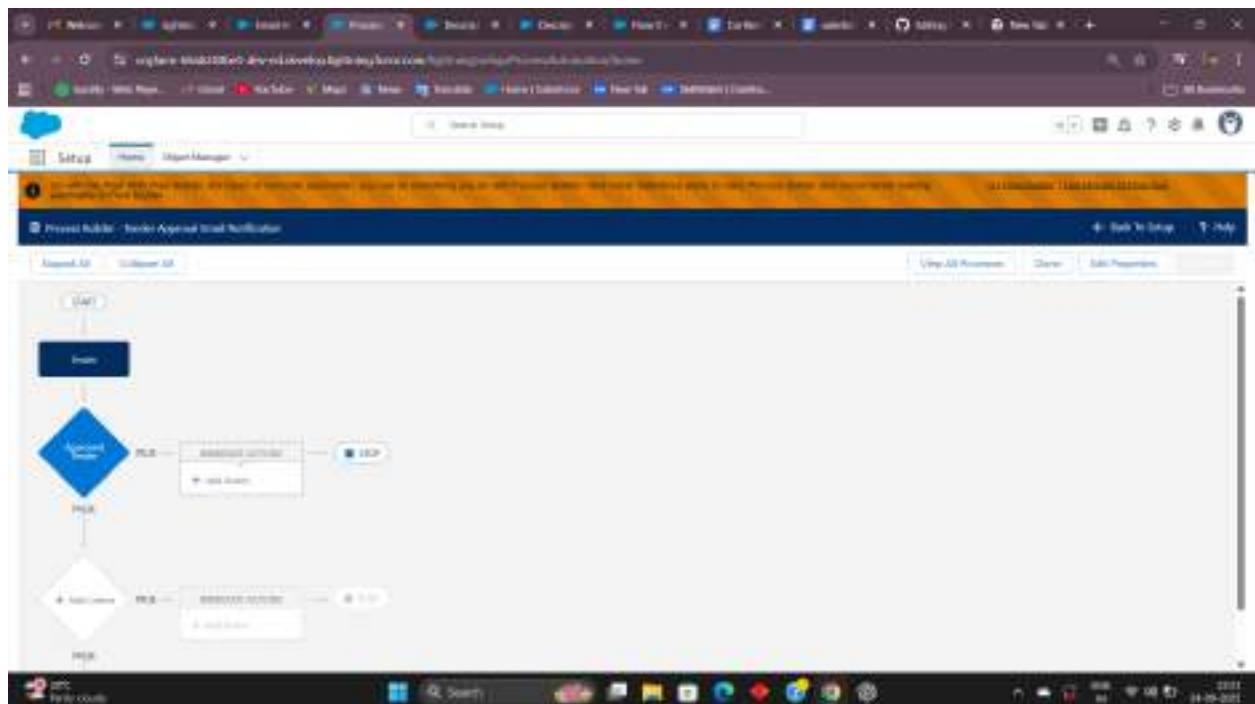
5. Flow Builder :

- 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 :

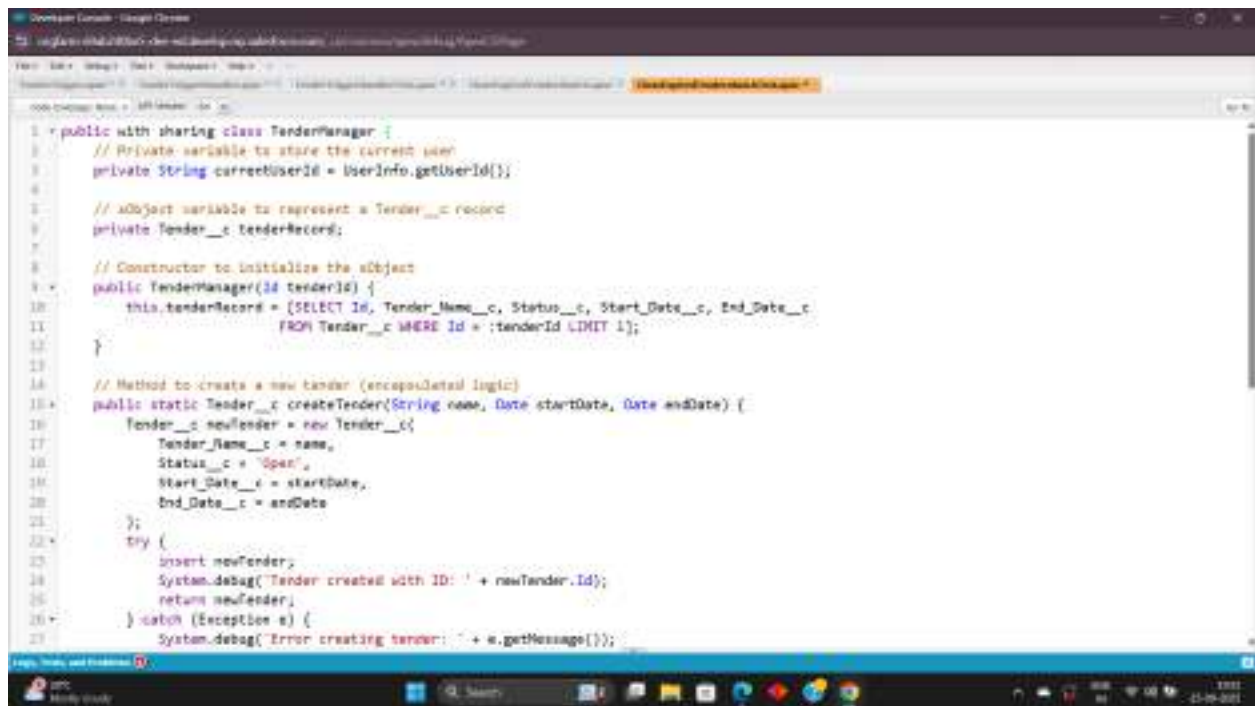
Customer email after approval.



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.
- records.



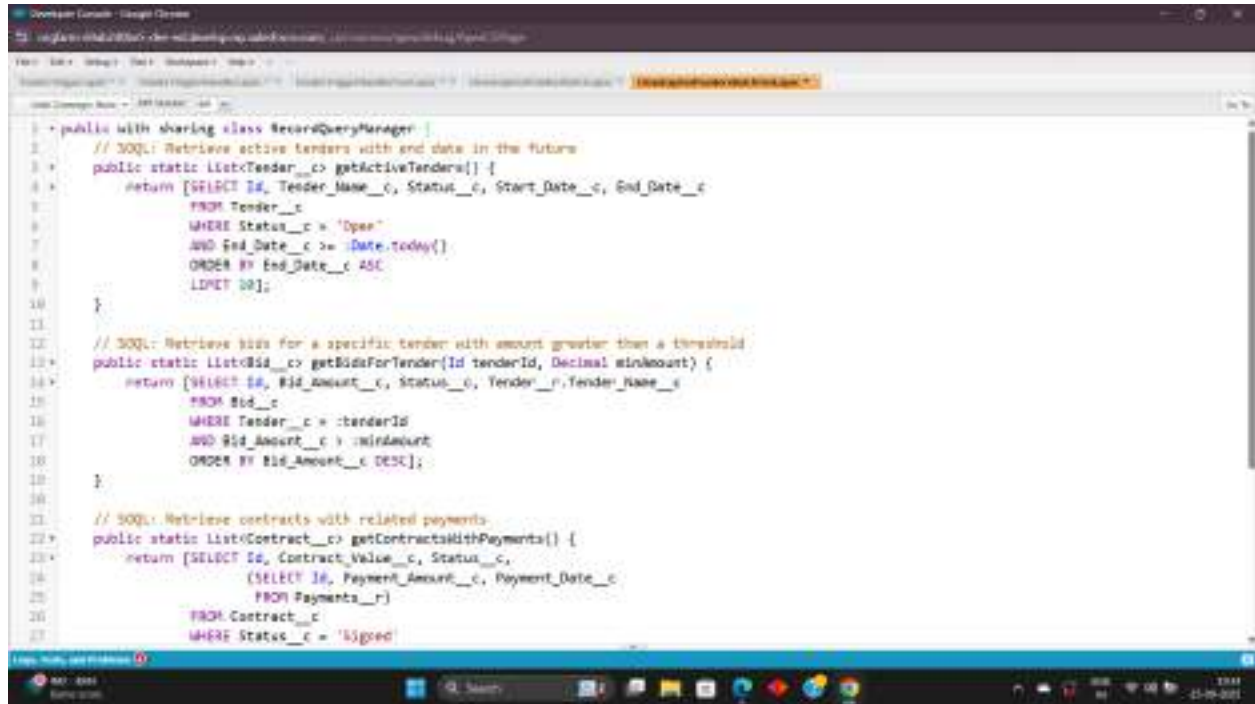
```
1 public with sharing class TenderManager {
2     // Private variable to store the current user
3     private String currentUserId = UserInfo.getUserId();
4
5     // sObject variable to represent a Tender__c record
6     private Tender__c tenderRecord;
7
8     // Constructor to initialize the object
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 }
```

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

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

4. SOQL & SOSL :

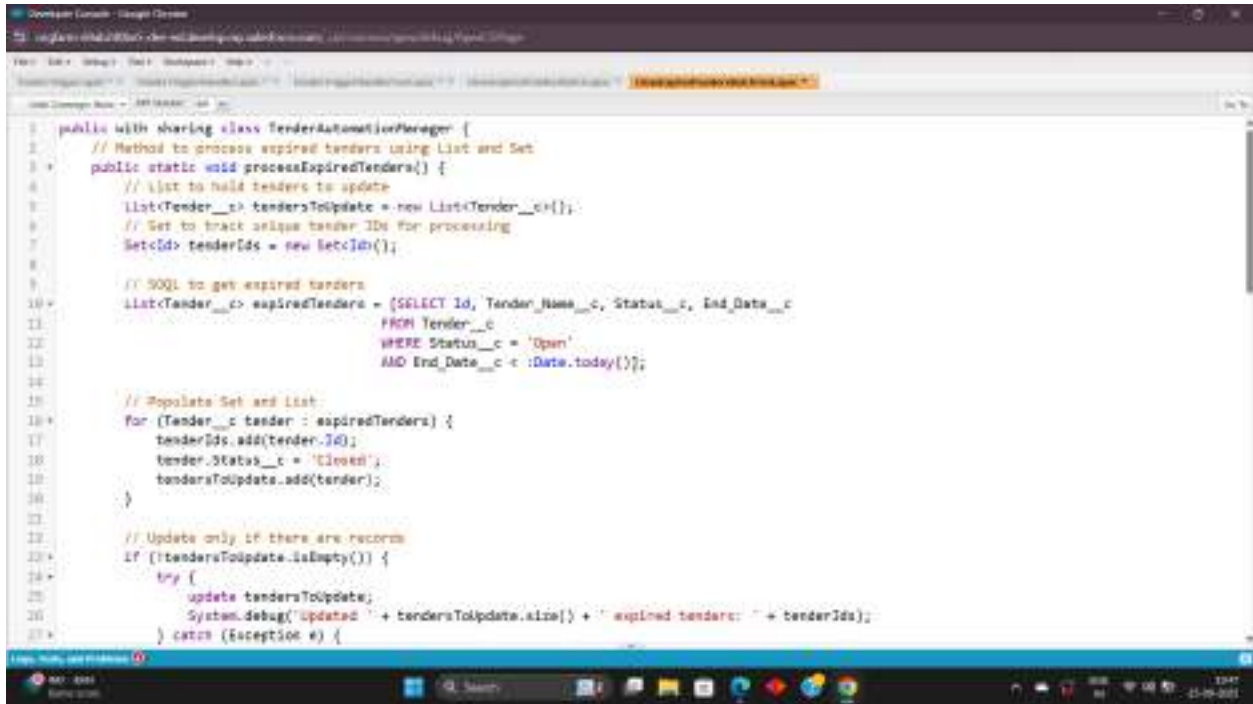
- SOQL (Salesforce Object Query Language) → Query Salesforce records.
- SOSL (Salesforce Object Search Language) → Search text across multiple objects.

A screenshot of a code editor window showing Apex code with SOQL queries. The code is for a class named RecordQueryManager. It contains three methods: getActiveTenders, getBidsForTender, and getContractsWithPayments. Each method uses SOQL to query Salesforce data. The first method queries for active tenders, the second for bids on a specific tender, and the third for contracts with related payments. The code is color-coded and includes comments explaining the purpose of each query.

```
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 20];
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'];
28     }
29 }
```

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.



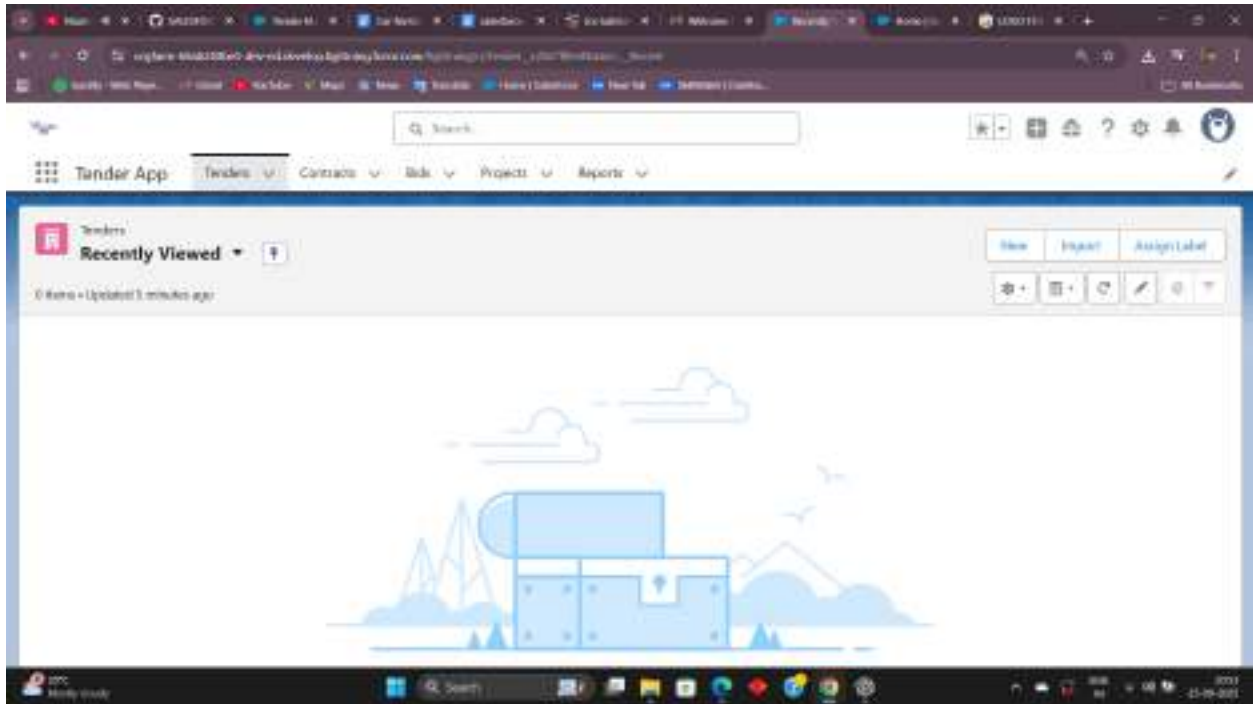
```
1 public with sharing class TenderAutomationHandler {
2     // Method to process expired tenders using List and Set
3     public static void processExpiredTenders() {
4         // List to hold tenders to update
5         List<Tender__c> tendersToUpdate = new List<Tender__c>();
6         // Set to track unique tender IDs for processing
7         Set<Id> tenderIds = new Set<Id>();
8
9         // SQL 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) {
28            }
29        }
30    }
31 }
```

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.

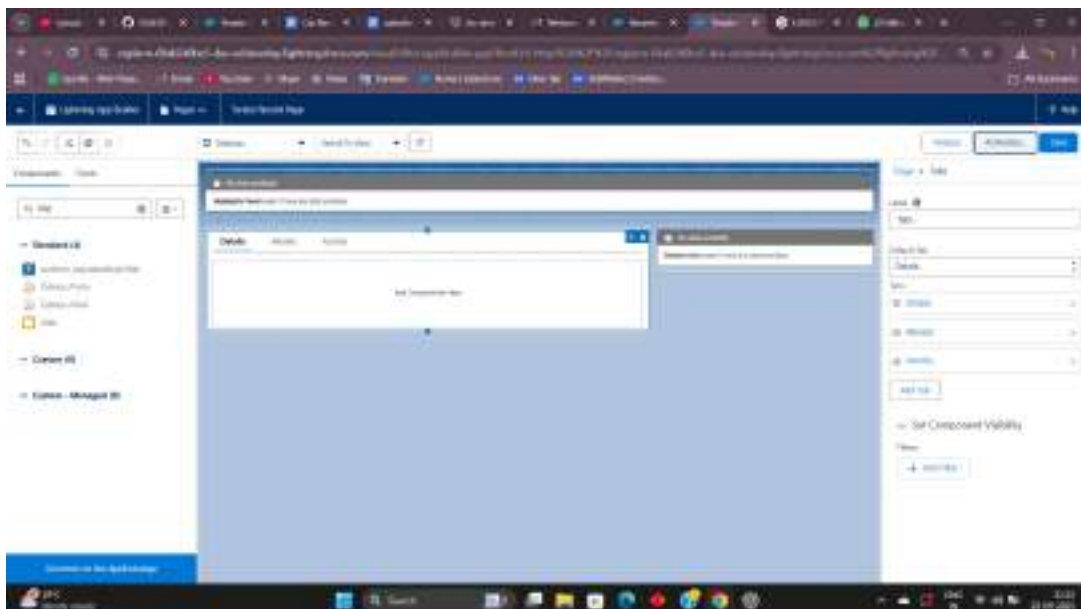


2. Record Pages :

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

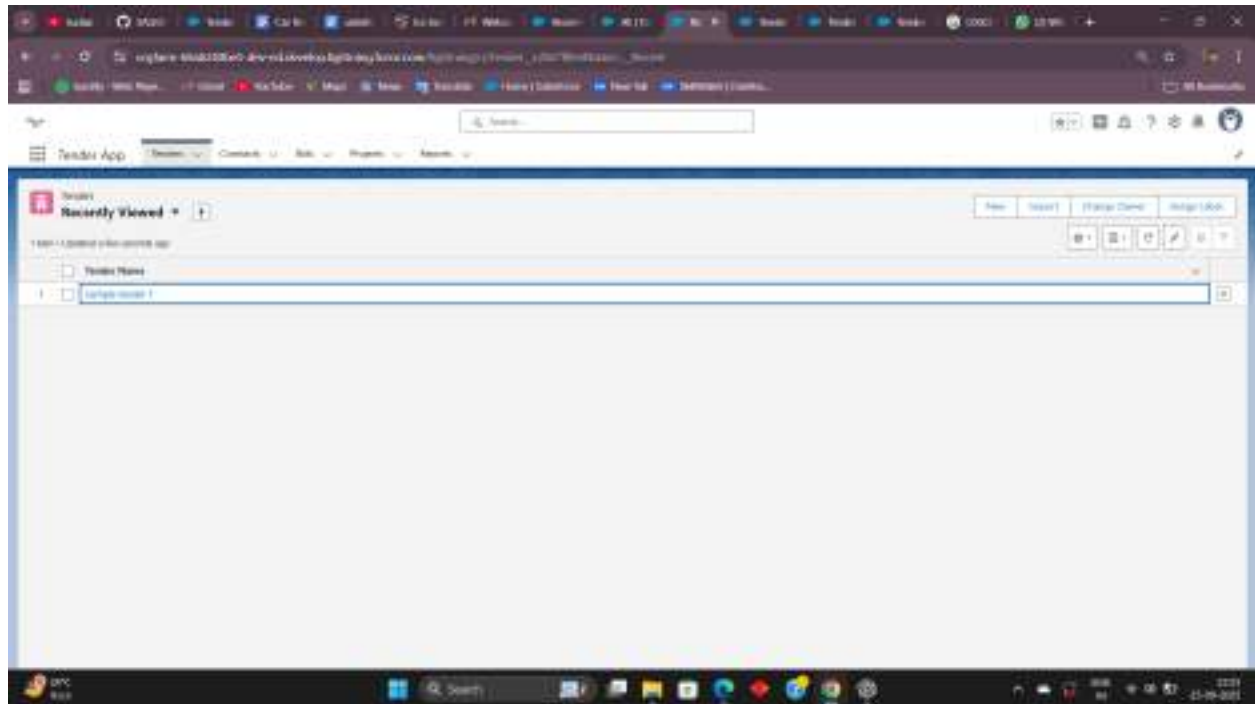
Steps:

- 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.

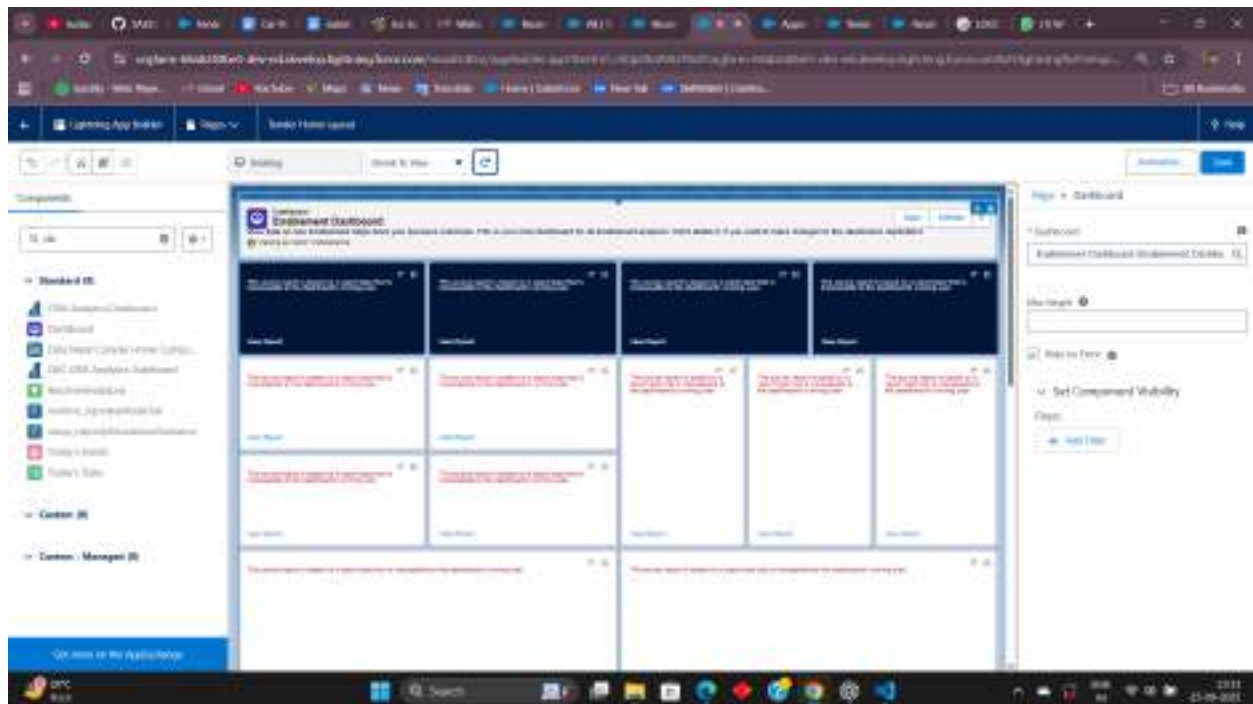


4. Home Page Layouts :

- Purpose: Customize the Salesforce Home page.

Steps:

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

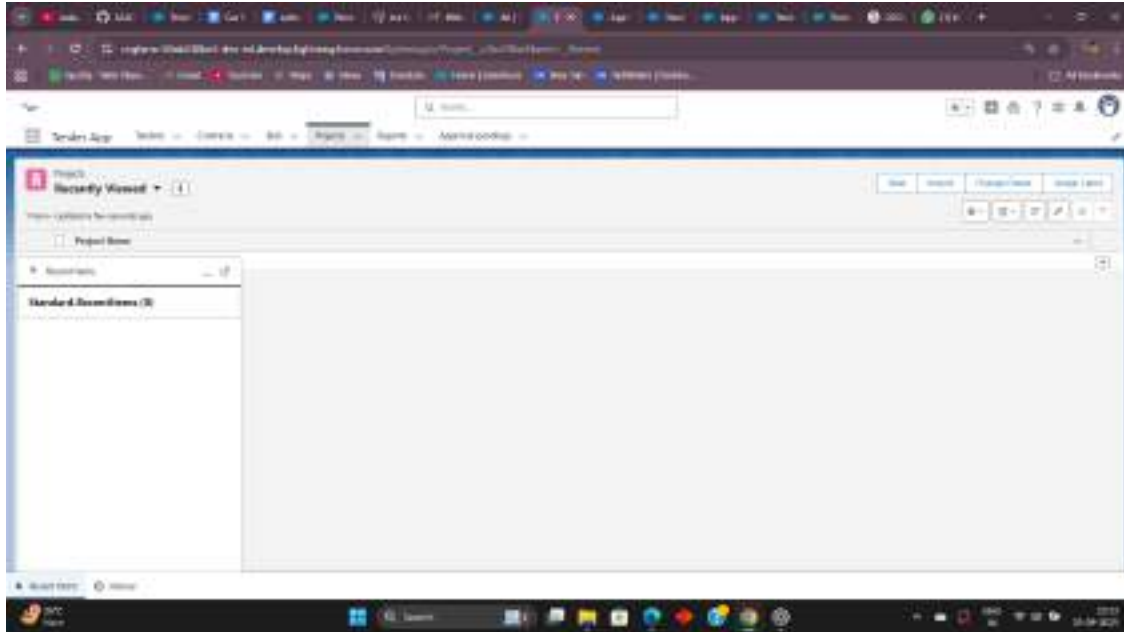


5. Utility Bar :

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

Examples:

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

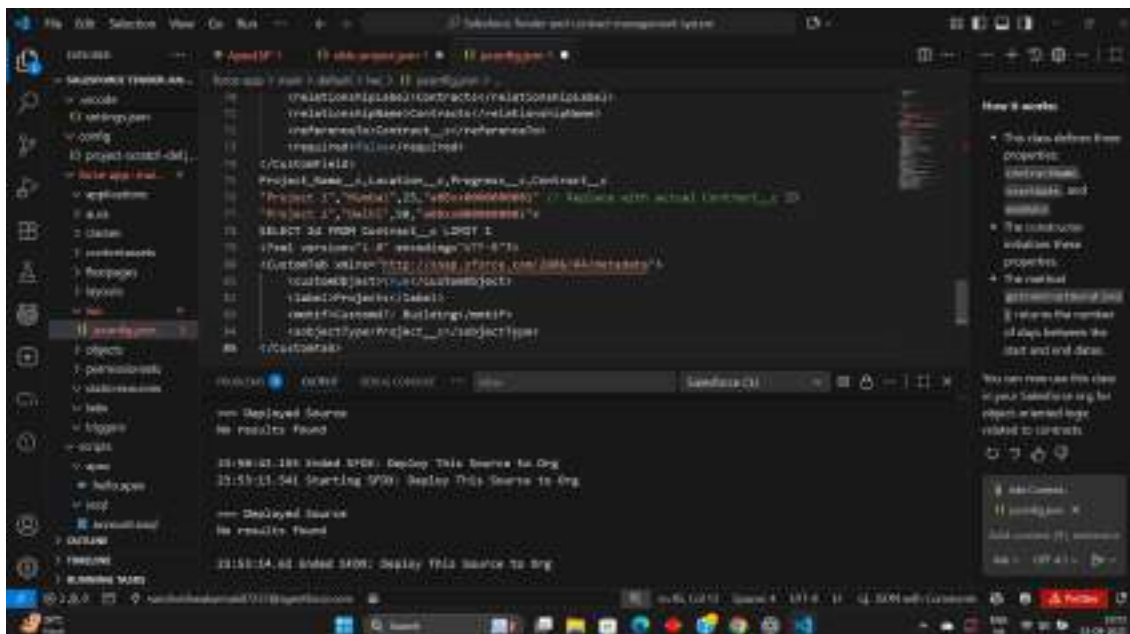


6. LWC :

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

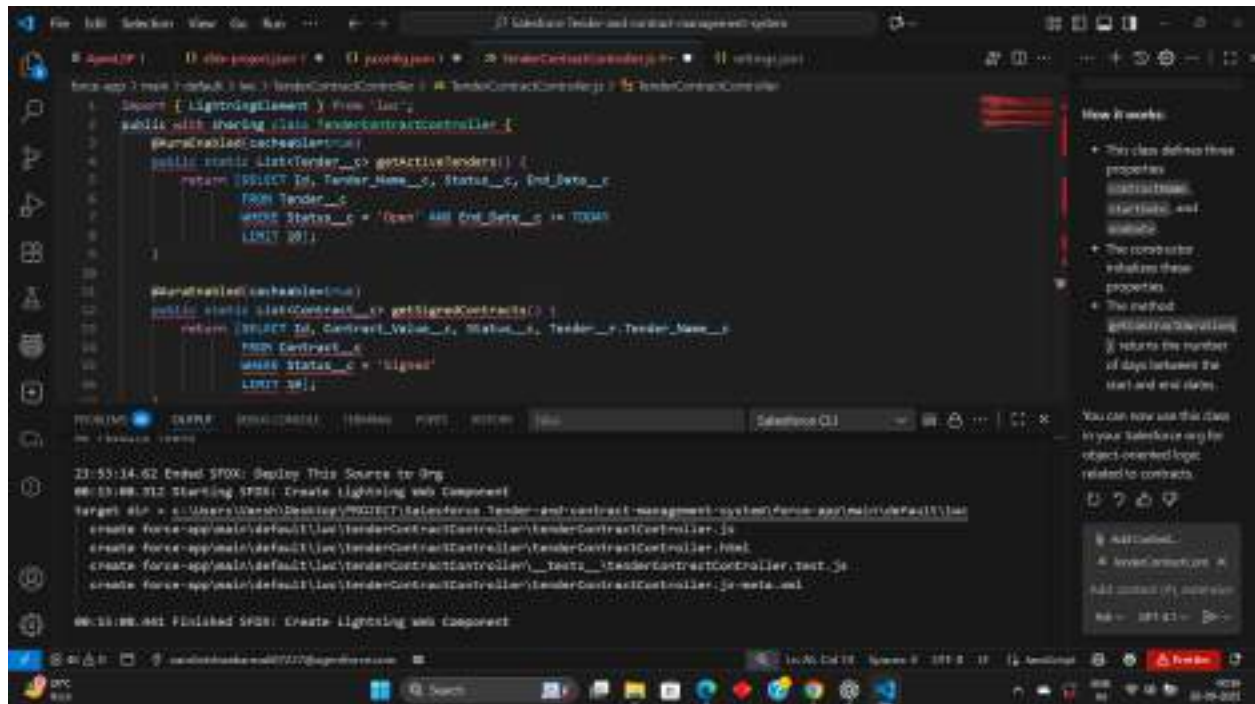
Examples for Tender App:

- 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 :

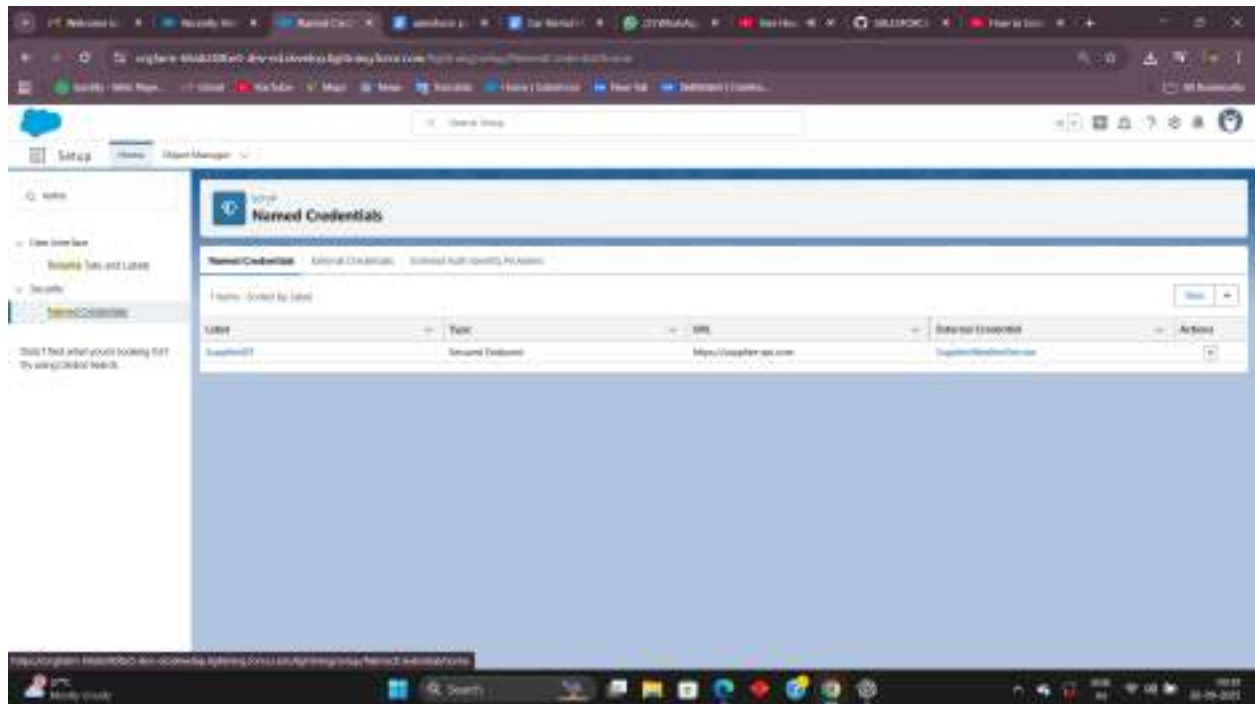
Purpose: Fetch Salesforce data via Apex into LWCs.



Phase 7: Integration & External Access

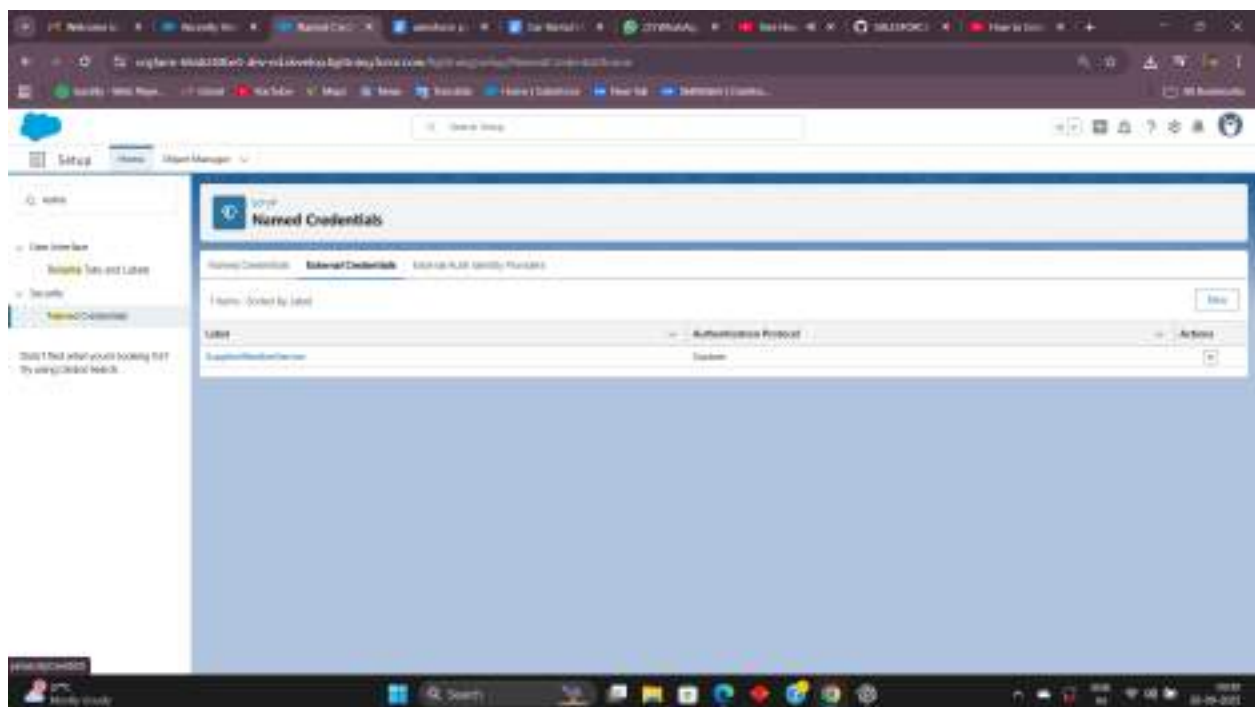
1 Named Credentials :

- Store authentication settings for external systems securely.



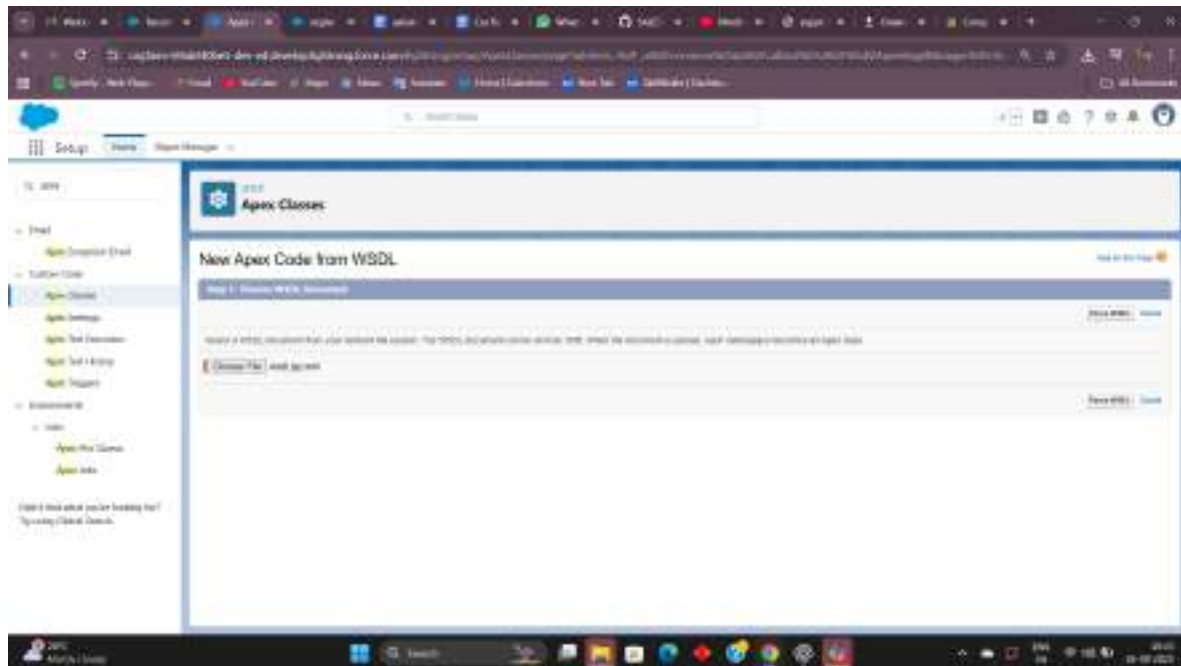
2. External Services :

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



3. Web Services (REST/SOAP)

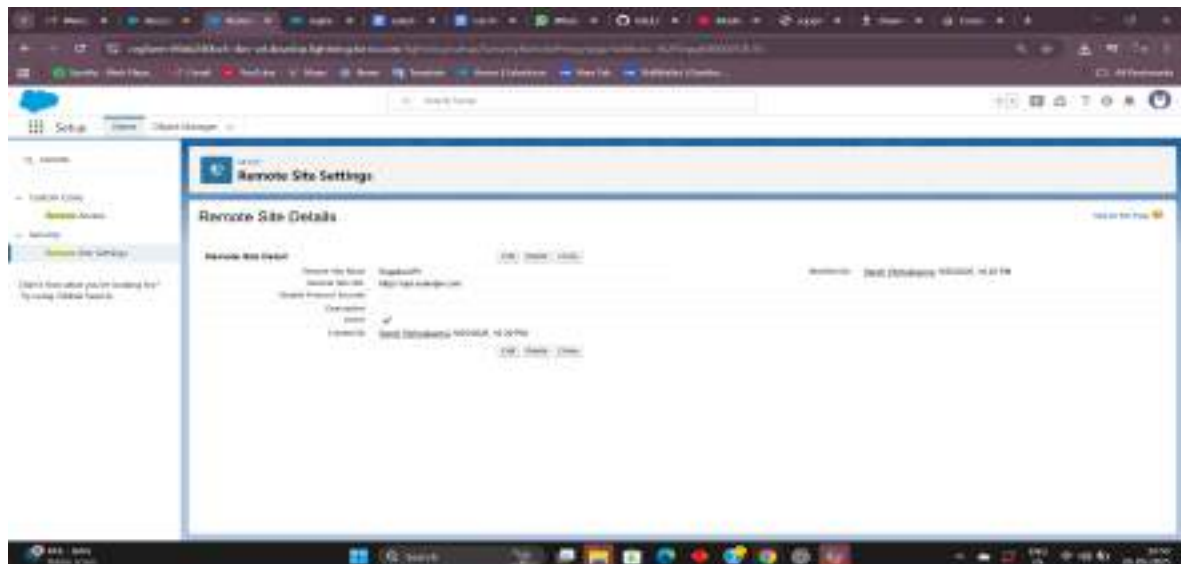
REST callout: Get insurance status.



4. Callouts :

Use HTTP Callouts to integrate Salesforce with external APIs.

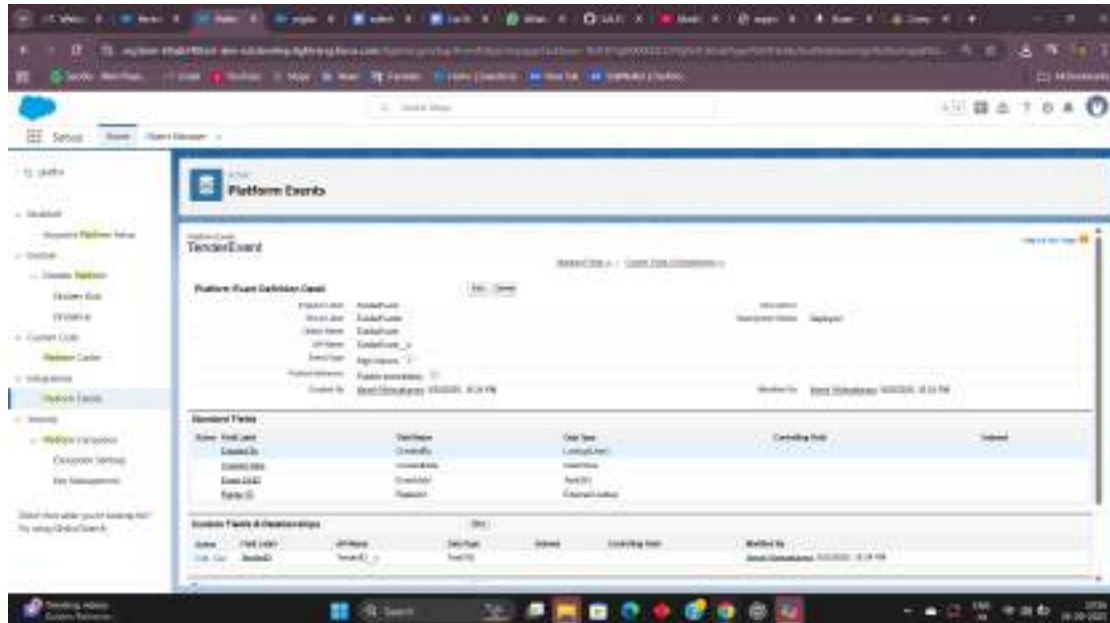
Example: Push contract approvals to an external project management tool.



5. Platform Events :

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

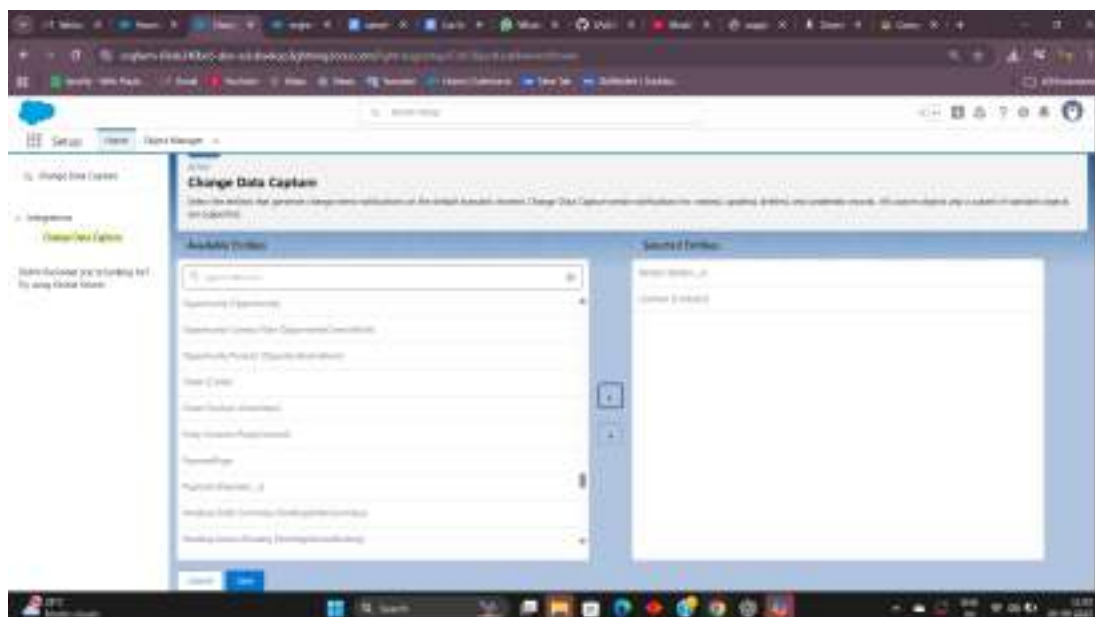
Example: Notify external systems when a tender is approved.



6. Change Data Capture :

Monitor Salesforce record changes in real-time.

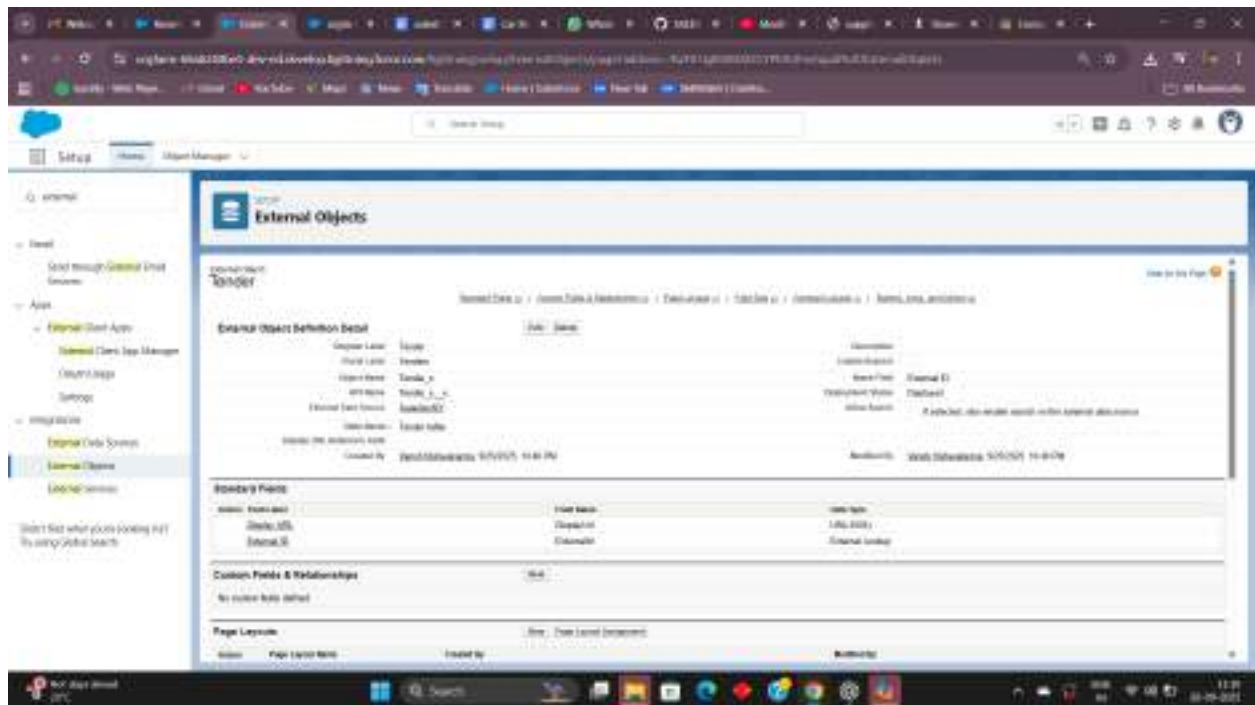
Example: Automatically sync updates on Contracts to ERP systems.



7. Salesforce Connect :

Access external objects and data without storing it in Salesforce.

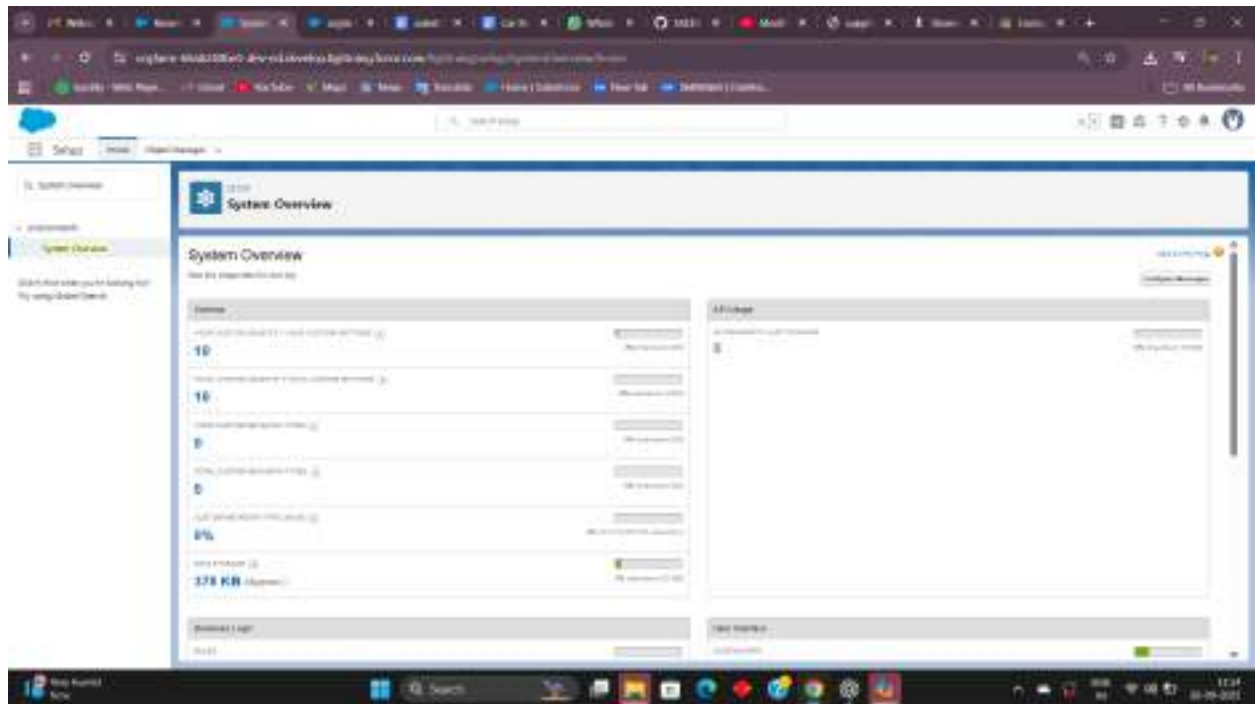
Example: View supplier bids from external database within Salesforce UI.



8. API Limits :

Monitor and manage API usage to avoid hitting limits.

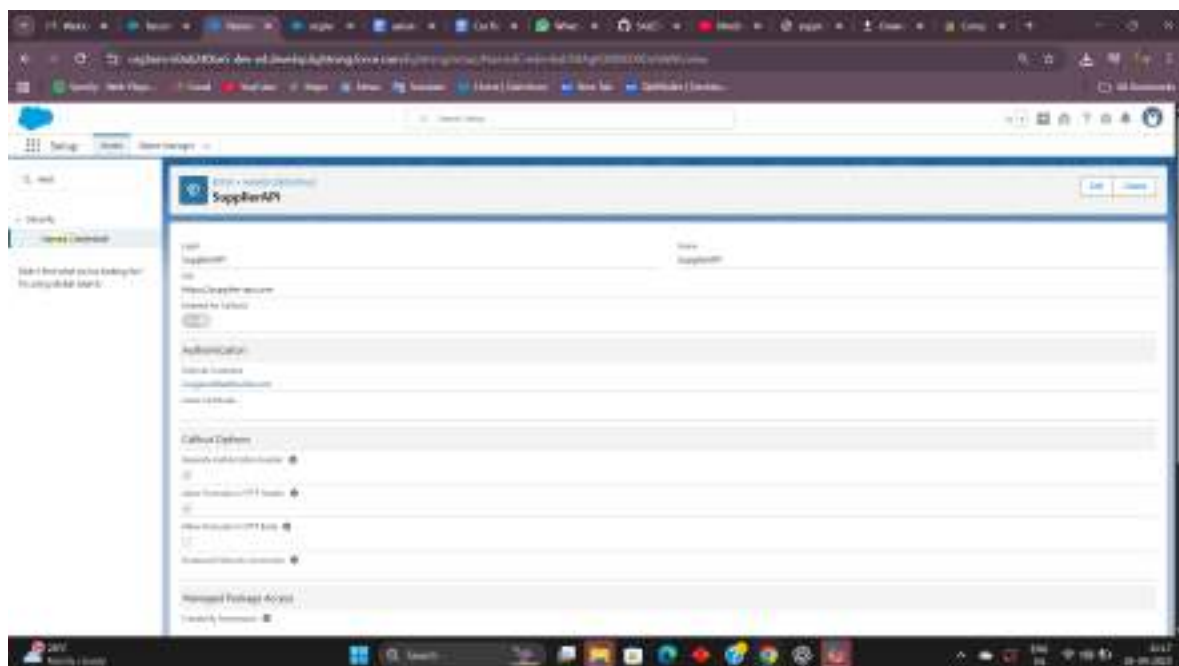
Example: Limit external integrations to prevent exceeding daily API calls.



9. OAuth & Authentication :

Securely authenticate Salesforce with external apps.

Example: OAuth 2.0 flow for integrating with partner systems.



10. Remote Site Settings :

Register external endpoints to allow Salesforce callouts.

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

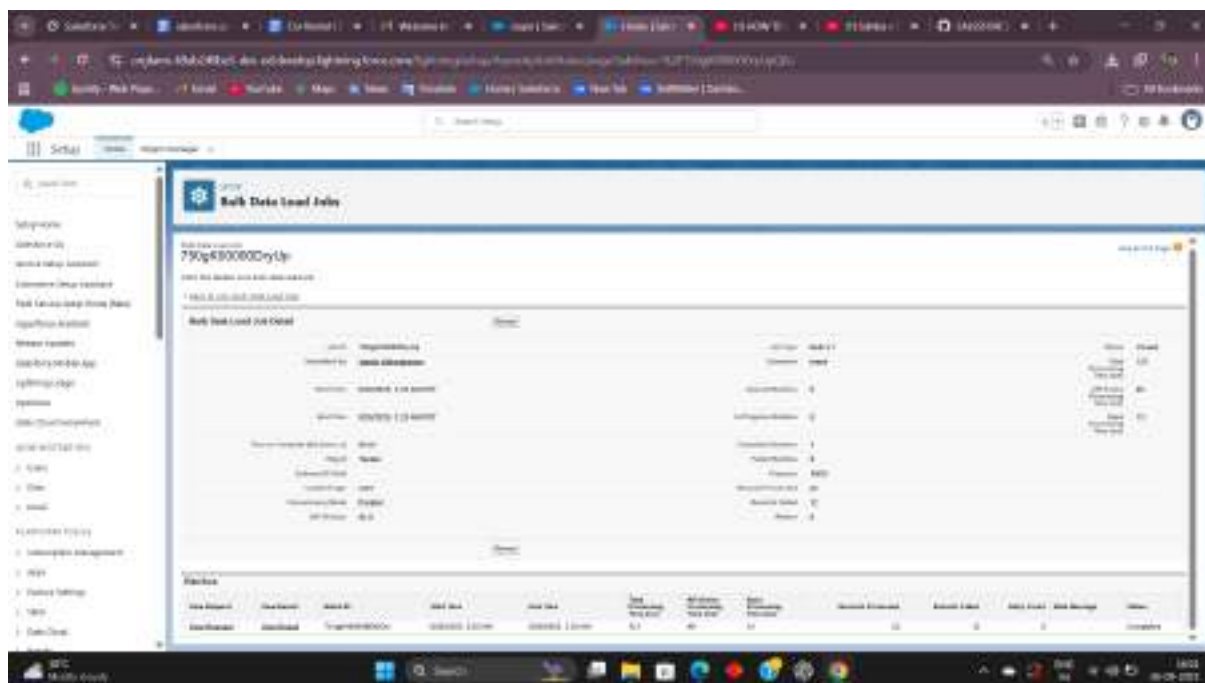
NOTE : we have already done in phase 4 .

Phase 8: Data Management & Deployment

1. Data Import Wizard :

Tool for importing small to medium datasets

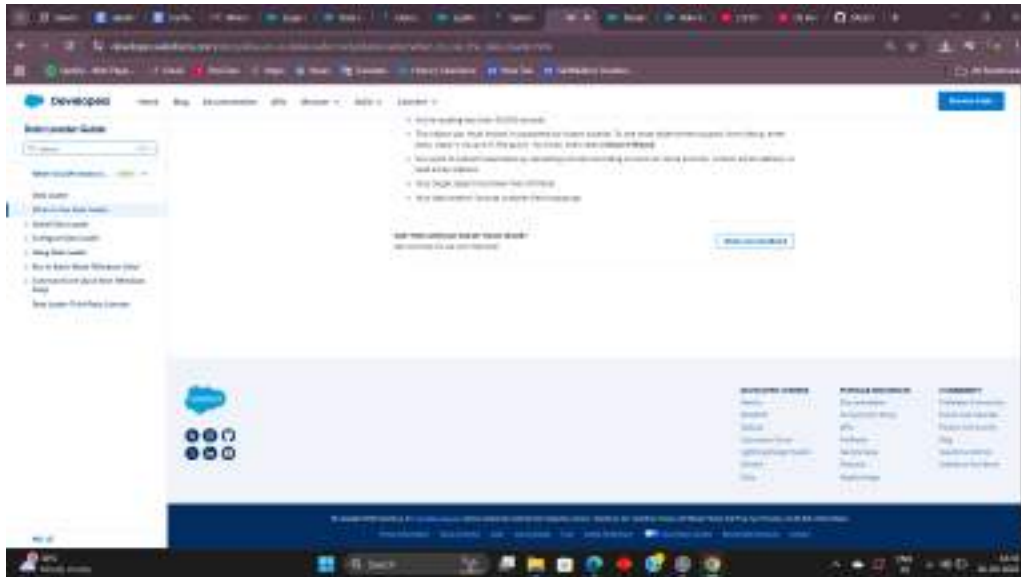
Enables adding records for Contractors, Tenders, Bids, Projects, and Payments.



2. Data Loader :

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

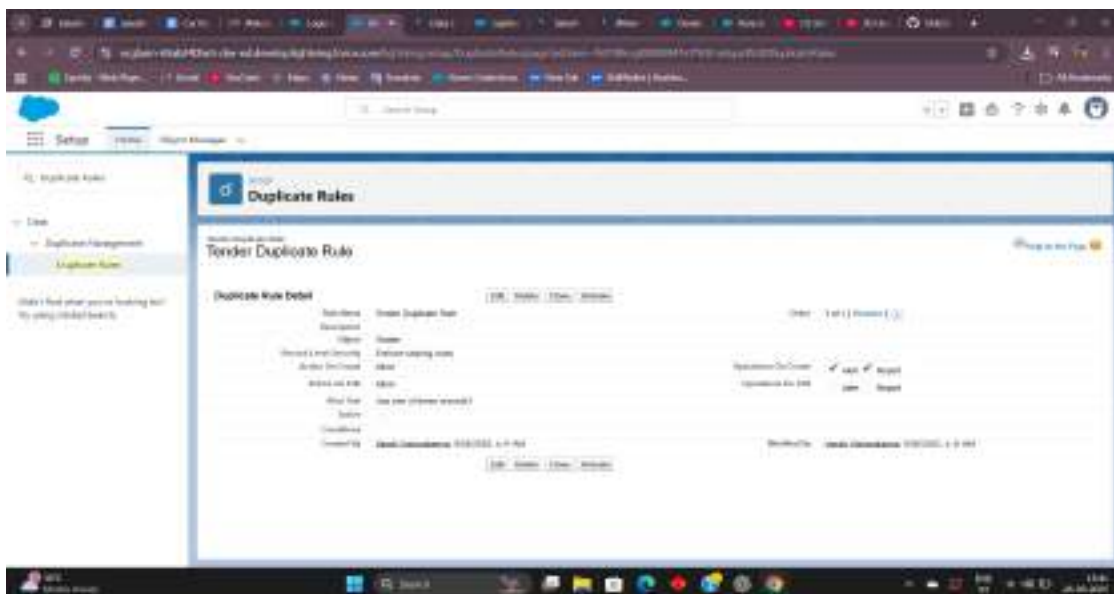
Ideal for large-scale data operations beyond the limits of the Data Import Wizard.



3. Duplicate Rules :

Prevents duplicate records and maintains data accuracy.

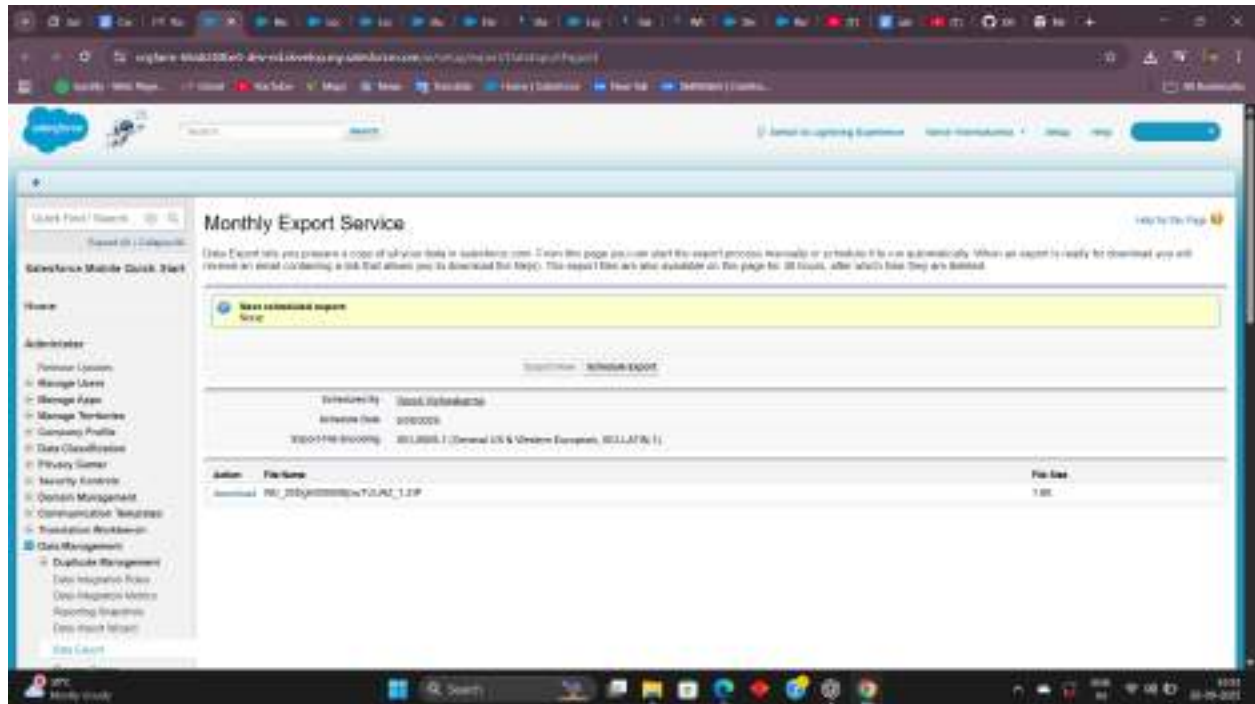
Ensures unique entries for Accounts, Contacts, Tenders, and Bids.



4. Data Export & Backup :

Provides regular backup of Salesforce data.

Helps recover data in case of accidental deletion or corruption.



5. ANT Migration Tool :

Enables programmatic metadata deployment using XML descriptors.

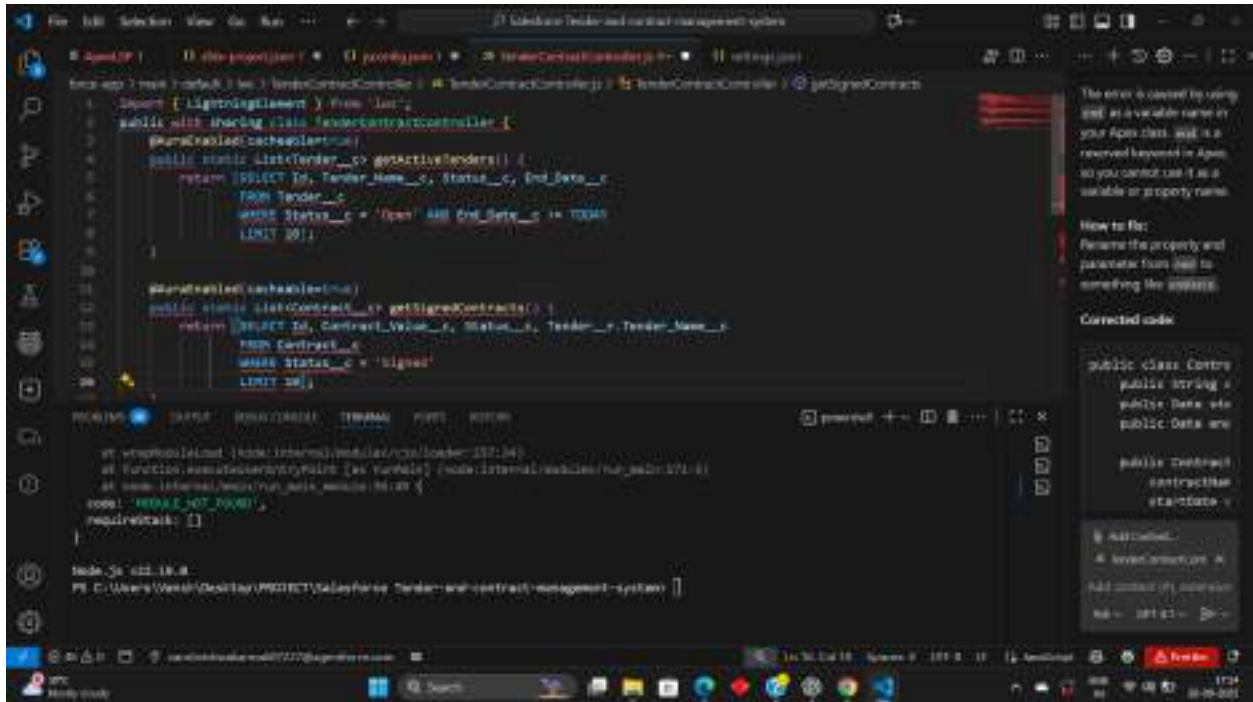
Supports version control and automated deployment pipelines.

Already done in step 4

6. VS Code & Salesforce CLI (SFDX) ;

Modern development environment for Salesforce.

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



Phase 9: Reporting, Dashboards & Security Review

👉 Goal: Monitor business & secure data.

1. Reports:

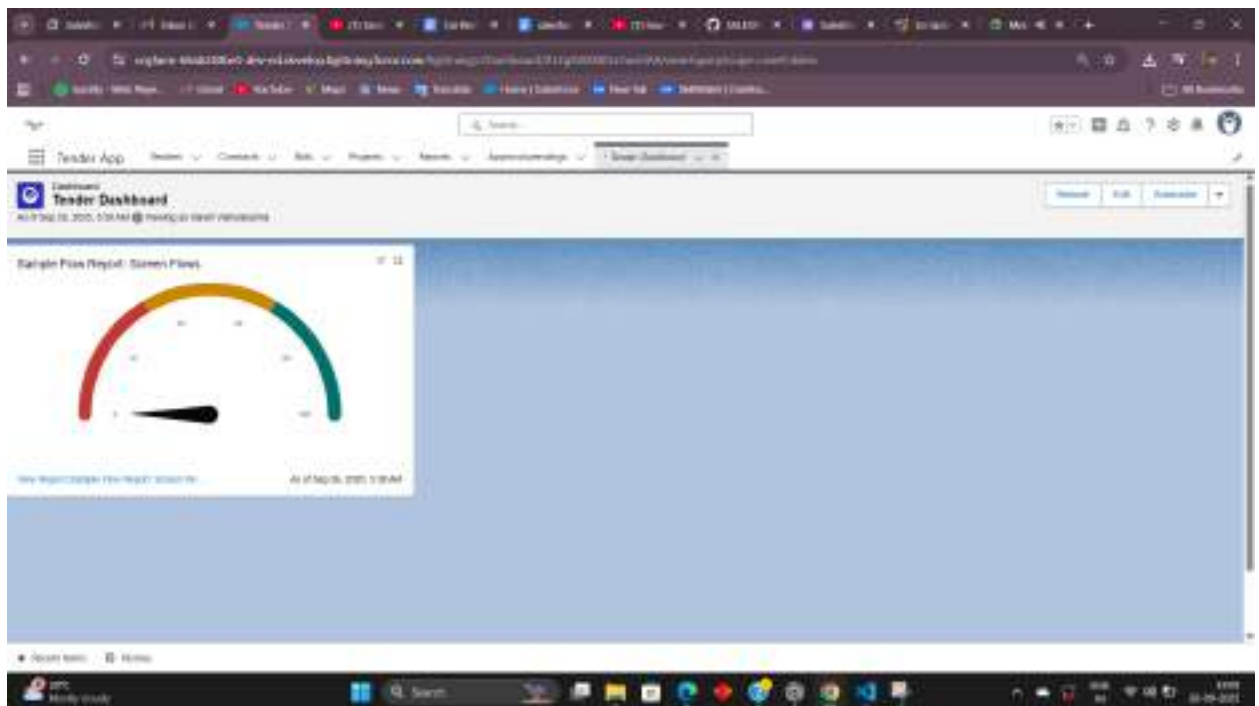
- Tabular, Summary, Matrix, Joined reports provide different ways to view and analyze Salesforce data.
- Reports can track Tenders, Bids, Contracts, Payments, and their status or value.

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.

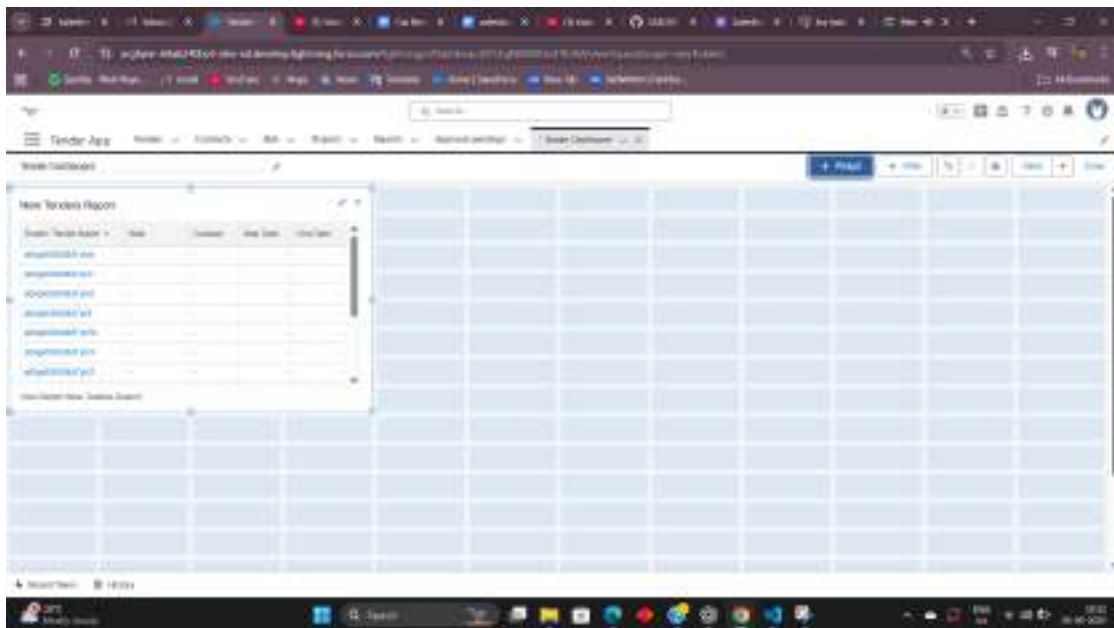
3. Dashboards:

- 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.

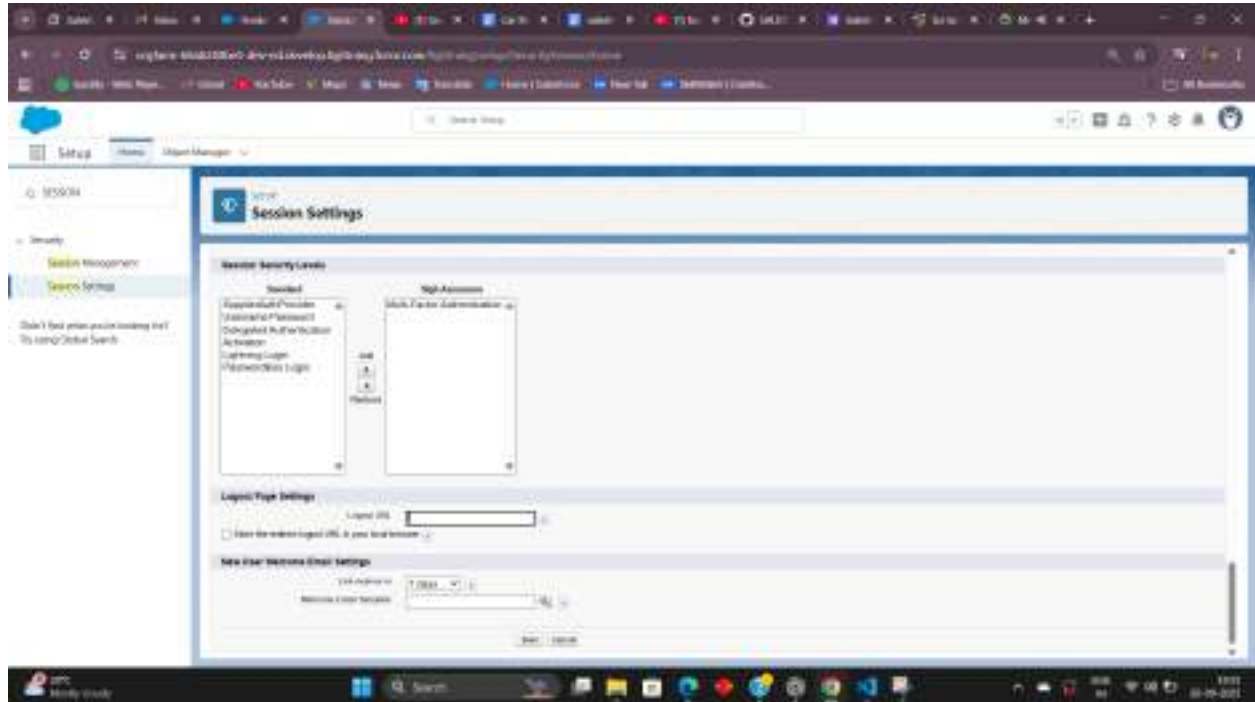


6. Sharing Settings :

- 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.

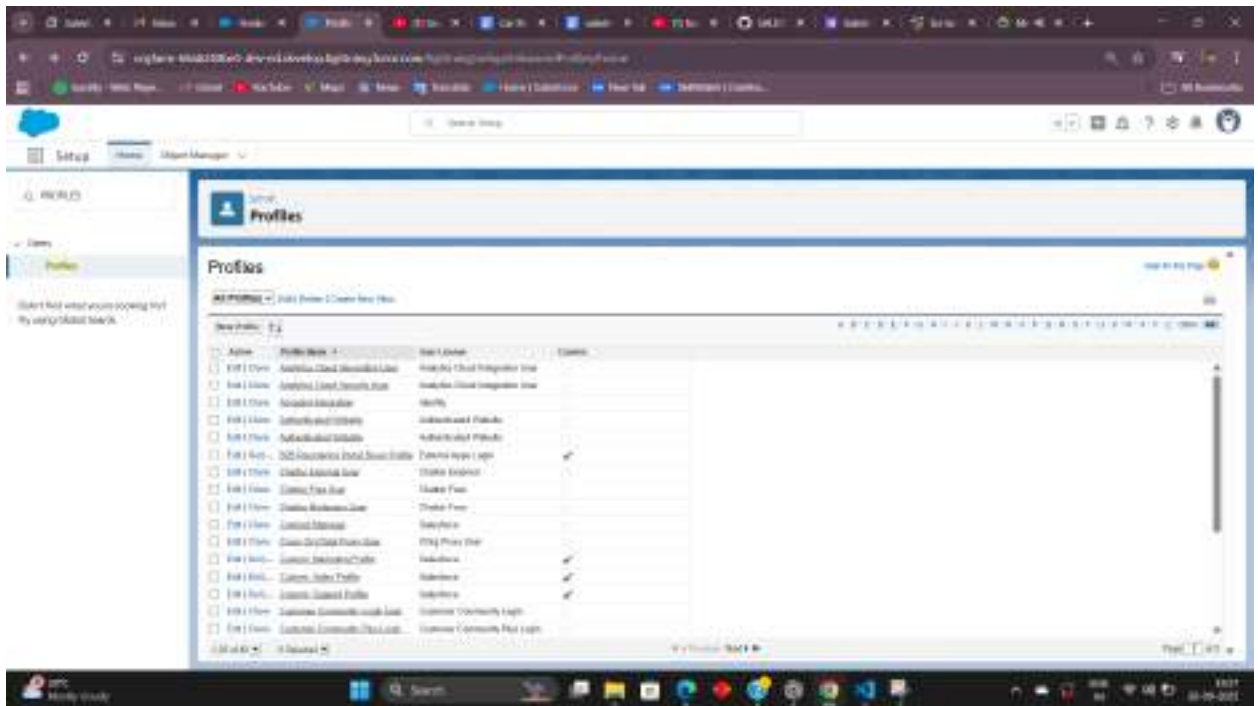
8. Session Settings :

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



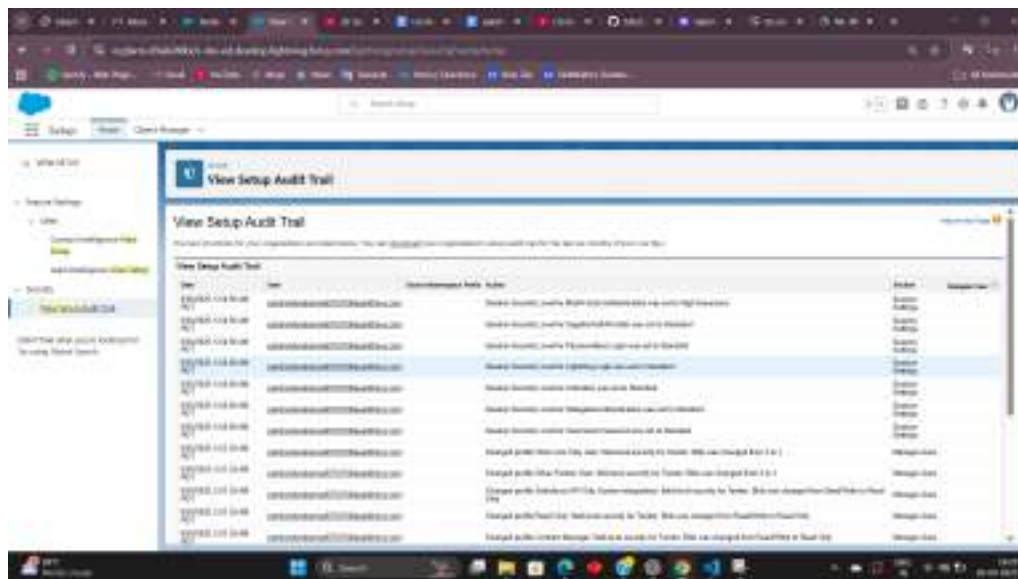
9. Login IP Ranges :

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



10 . Audit Trail :

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



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.