📘 Smart Sales Opportunity & Quotation Automation System

Phase 1: Problem Understanding & Industry Analysis

The first phase of the project focused on deeply understanding the problem, identifying the needs of different stakeholders, analyzing the current challenges, and exploring how industries handle similar processes. This phase laid the foundation for building the automation system.

1. Requirement Gathering

The manual process of quotation creation in sales cycles was studied. Currently, when an Opportunity is marked as *Closed Won* in Salesforce, sales representatives have to create a **Quotation record manually**.
This causes several challenges:

- **Delays** in generating quotations, which slows down the sales cycle.
- **Inconsistencies and errors** in quotation details (amount, validity date, status).
- **Dependency on manual effort**, leading to inefficiency when handling multiple opportunities.

The requirement gathered was clear:

- Automate the quotation generation process immediately when an Opportunity is *Closed Won*.

- Ensure the quotation moves through a proper **approval process** before being shared with the customer.
- Improve speed, accuracy, and overall sales productivity.

2. Stakeholder Analysis

Different stakeholders were identified, each with their own expectations:

- **Sales Representatives**
  - Want quotations to be auto-generated as soon as deals are closed.
  - Focus on reducing manual workload.
- **Sales Managers**
  - Need control over approving or rejecting quotations.
  - Require visibility into quotations pipeline.
- **Customers**
  - Expect accurate and timely quotations without delays.
  - Their satisfaction directly impacts business reputation.
- **Salesforce Admins/Developers**
  - Responsible for implementing, maintaining, and scaling automation.
  - Must ensure the solution is flexible and extendable.

3. Business Process Mapping

The existing (manual) process was compared with the proposed (automated) process:

- **Manual Process:**
  - Sales rep closes the Opportunity.
  - Creates a Quotation record manually.
  - Sends it to the manager for approval.
  - Customer finally receives it, often with delays.
- **Automated Process:**
  - Opportunity stage changes to *Closed Won*.
  - System auto-generates Quotation record with fields (Name, Amount, Valid Till, Status = Draft).

- User submits Quotation for approval → Status changes to *Pending Approval*.
- Approver reviews → Approves (Status = Approved) or Rejects (Status = Rejected).
- Approved Quotation is ready for customer communication.

This mapping clearly demonstrated how automation reduces steps, saves time, and avoids errors.

4. Industry-specific Use Case Analysis

Similar problems exist across industries:

- **IT/Software Services:** License or subscription deals require quick quotations after deals are closed.
- **Manufacturing:** Bulk orders need instant quotations to speed up B2B negotiations.
- **Retail/Distribution:** Fast-moving sales cycles demand quotations without manual delays.

This shows that the problem is industry-wide and the solution can scale to multiple domains.

5. AppExchange Exploration

Before designing the solution, existing marketplace apps were reviewed:

- **Salesforce CPQ (Configure, Price, Quote):**
  Very powerful but highly complex and license-based.
- **Other Quoting Apps:**
  Available on AppExchange but most are paid and heavy.

Decision: Build a **custom lightweight solution** using:

- Custom Object (`Quotation__c`)
- Flows for automation

- Approval Processes for control
  This balances cost, flexibility, and learning.

Outcome of Phase 1

- Clear **requirements** defined.
- All **stakeholders** and their needs identified.
- Existing vs Proposed **business processes** mapped.
- Explored industry relevance and AppExchange solutions.
- Final decision: Proceed with a **custom Salesforce automation** project.

This phase ensured that the project started with a solid understanding of the problem and the right direction for development

Phase 2: Salesforce Org Setup & Customization

The second phase of the project focused on setting up the Salesforce development environment, integrating it with local development tools, and customizing the Salesforce org to meet project requirements. This phase established the technical foundation for building and testing the automation.

1. **Salesforce Edition**
    a. We used a **free Salesforce Developer Edition Org** for the project.
    b. This org acted as our environment for building, testing, and demonstrating the solution.
2. **Company Profile Setup**
    a. Default company profile settings were used.
    b. Time zone was set according to IST (Indian Standard Time).
    c. Currency remained as default (USD) since currency customization was not required.
3. **Business Hours & Holidays**
    a. Default working hours (24/7) were retained.
    b. Customization of holidays/business hours was not required for this academic use case.
4. **Fiscal Year Settings**
    a. Standard fiscal year (Jan–Dec) was kept, which is the Salesforce default.
5. **User Setup & Licenses**
    a. Project was executed within a single Developer Org user (System Administrator).
    b. No extra licenses or users were created.
6. **Profiles and Roles**
    a. Default **System Administrator profile** was used.
    b. Role hierarchy was not implemented as multi-user access control was not in project scope.
7. **Permission Sets, OWD, Sharing Rules**
    a. Default security settings were retained.
    b. Since this project focused on automation and approval processes, advanced data sharing was not needed.
8. **Login Access Policies**
    a. Default policies applied. No restrictions configured.
9. **Dev Org Setup & Sandbox Usage**
    a. A **Developer Org** was used as both dev and test environment.
    b. No sandbox was required for this academic project.

10. **Deployment Basics**
- For deployment/version control, we integrated our Salesforce Org with **VS Code and GitHub**.
- Metadata (Quotation Object, Fields, Flows) was retrieved and pushed to GitHub repo.

1. Salesforce Developer Org Setup

- A **Salesforce Developer Edition Org** was created to serve as the working environment for the project.
- This developer org provides free access to Salesforce features needed for experimentation, customization, and deployment.
- It ensures a safe and isolated environment where custom objects, flows, and approval processes can be developed and tested without affecting a production system.

2. Local Development Setup

To enable source-driven development and version control, the local environment was prepared:

- **Salesforce CLI (sf CLI):** Installed and configured to connect the Salesforce org with the local machine.
- **VS Code Integration:** Visual Studio Code was set up with Salesforce extensions to manage metadata.
- **Org Authentication:**
    - Connected the local project to Salesforce Org using:

```
sf org login web --set-default --alias myorg
```

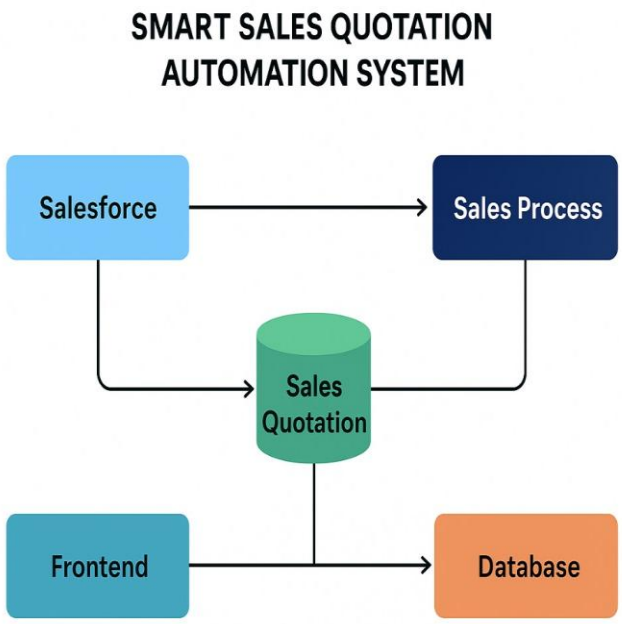
    - This allowed pushing and retrieving metadata between the local system and Salesforce.

3. GitHub Repository Initialization

Version control is essential for project tracking and collaboration.

- A GitHub repository was created:
  Smart-Sales-Opportunity-Quotation-Automation-System
- Git commands used:
  - `git init` to initialize the repo.
  - `git remote add origin` to connect with GitHub.
  - `git add .` and `git commit` for tracking changes.
  - `git push` for uploading project metadata.

This setup ensures all Salesforce customizations are backed up and can be shared with others.



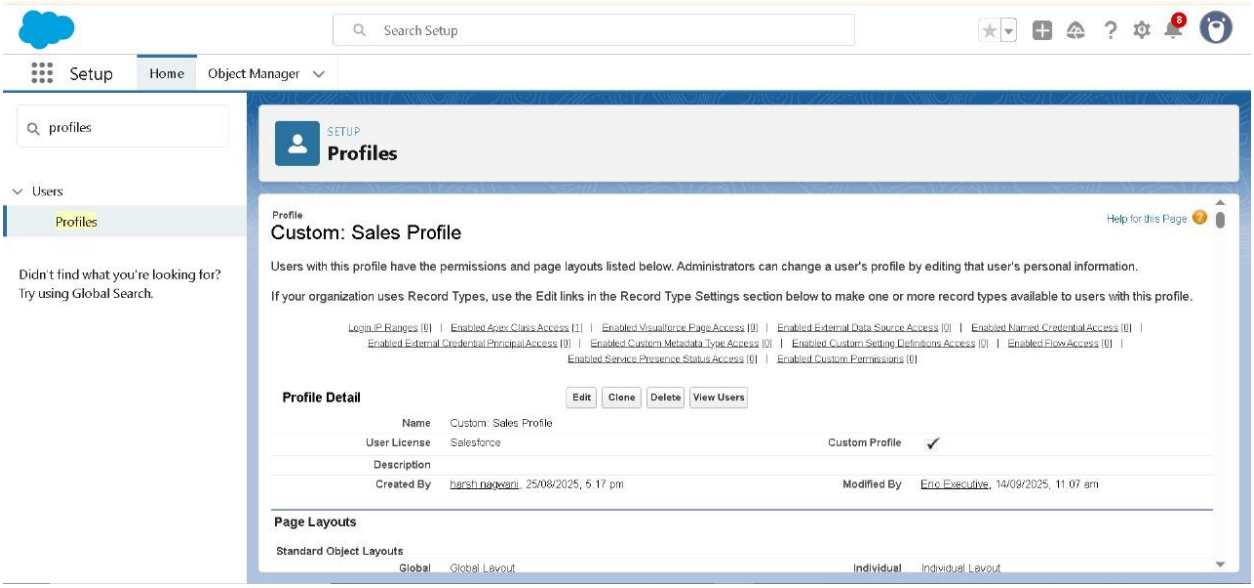
4. Custom Object Creation: Quotation__c

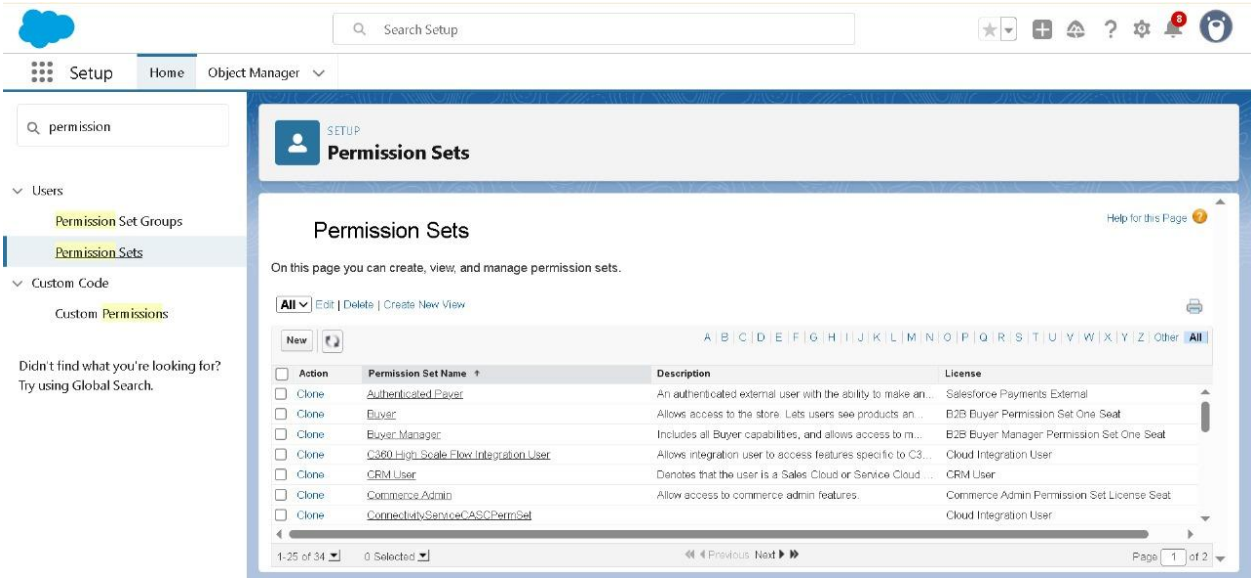
To manage quotations, a **new custom object** `Quotation__c` was created. This object is central to the project.

Fields created:

- **Quotation Name** (Text) → stores the name of the quotation.
- **Total Amount** (Currency) → captures the quotation value.
- **Status** (Picklist) → values: Draft, Sent, Approved, Rejected.
- **Valid Till** (Date) → defines the validity period of the quotation.
- **Opportunity** (Lookup → Opportunity) → establishes the link between an opportunity and its quotation.

This customization enabled Salesforce to store all quotation data in a structured manner.





5. Metadata Retrieval & Versioning

- After creating the object and fields in Salesforce, metadata was retrieved locally using:

```
sf project retrieve start --metadata "CustomObject:Quotation__c"
```

- The retrieved files were committed to GitHub.

- This ensured that the configuration is documented, reusable, and can be deployed to another Salesforce org if needed.

Outcome of Phase 2

- Salesforce developer org was successfully set up and connected with local tools.
- GitHub repository established for collaboration and version control.
- Custom object `Quotation__c` with all required fields was created.
- Metadata retrieval and commits validated the integration between Salesforce and local project.

With Phase 2 completed, the technical base was ready to implement business logic automation in the next phase.

Phase 3: Data Modeling & Relationships

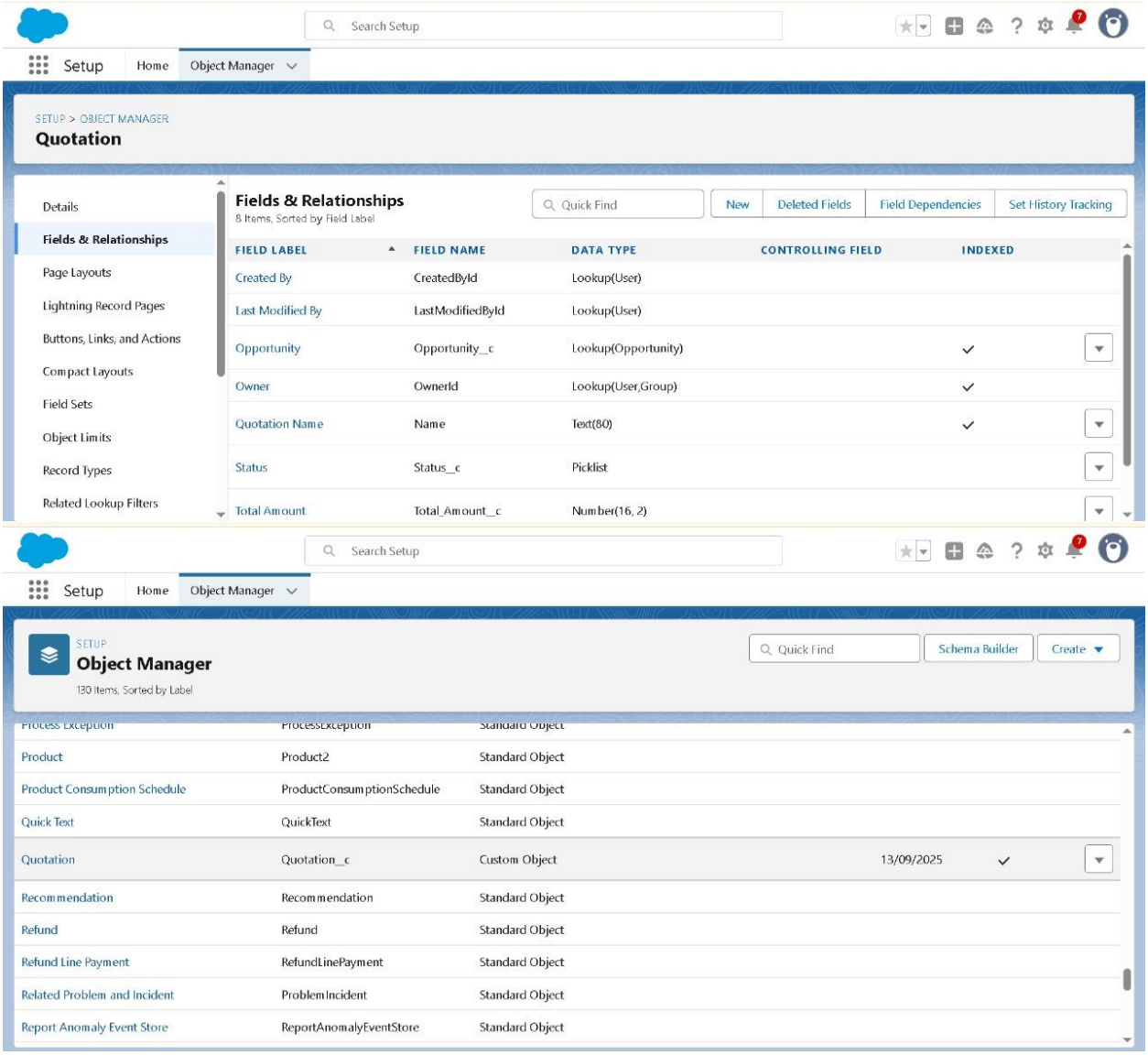
The data model defines the logical structure of the application's data, ensuring scalability, integrity, and alignment with business processes. This phase involved the definition of objects, fields, and relationships that form the foundation of the system.

Standard & Custom Objects

Standard Objects are foundational, pre-built objects within the Salesforce platform, such as `Account` and `Opportunity`. **Custom Objects** are user-defined objects created to meet specific business requirements; their API names are suffixed with `__c`. This project utilizes the standard `Opportunity` object as the primary driver for automation and the custom `Quotation__c` object to store data specific to the sales quoting process.

Fields

Fields represent individual data attributes of an object, each defined by a specific data type (e.g., Text, Currency, Date). In this implementation, the `Quotation__c` object includes custom fields such as `Total_Amount__c` (Currency) and `Status__c` (Picklist) to capture essential quotation details.



Record Types

Record Types enable the implementation of distinct business processes, page layouts, and picklist values for different user profiles within a single object. While not utilized in the initial phase, a future enhancement for this project could involve creating Record Types on the `Quotation__c` object to manage different quoting processes, such as "New Business" versus "Renewal," each with a unique lifecycle and user interface.

Page Layouts

Page Layouts determine the organization of fields, related lists, and other UI elements on a record's detail page. For this project, the `Opportunity` page layout was customized to include the `Quotation__c` related list, providing users with a consolidated view of all quotes associated with a deal.

Compact Layouts

Compact Layouts define the set of key fields displayed in a record's highlights panel, within the Salesforce mobile application, and in lookup previews. For the `Quotation__c` object, a Compact Layout was configured to display `Quotation Name`, `Status`, and `Total Amount` for at-a-glance record identification.

Schema Builder

Schema Builder is a graphical tool within Salesforce that provides a dynamic visualization of the data model, including objects, fields, and their relationships. It was utilized during the design phase to map and validate the relationship between the `Quotation__c`, `Opportunity`, and `Account` objects, ensuring the data architecture was correctly implemented.

Lookup vs. Master-Detail vs. Hierarchical Relationships

Salesforce provides several types of object relationships to model data connections:

- **Lookup Relationship**: A loosely coupled one-to-many relationship where records can exist independently and have separate security models.
- **Master-Detail Relationship**: A tightly coupled parent-child relationship where the detail record's existence and security are dependent on the master record.
- **Hierarchical Relationship**: A specialized lookup on the `User` object to create parent-child structures, such as management reporting lines.

For this project, a **Lookup Relationship** was implemented between `Quotation__c` (child) and `Opportunity` (parent). This design choice ensures that quotation records are preserved for historical purposes even if the associated opportunity is deleted and allows for a more flexible security model for quotations.

Junction Objects

A **Junction Object** is a custom object designed to create a many-to-many relationship between two objects by using two master-detail relationships. This construct was not required for the current data model, as the relationship between an `Opportunity` and its `Quotations` is one-to-many.

External Objects

External Objects enable the integration of data from external systems into the Salesforce UI without physical data replication, typically through Salesforce Connect. While not part of the initial implementation, External Objects could be leveraged in future phases to display real-time product pricing or inventory data from an external ERP system directly on the `Quotation__c` record.

Of course. Here is a professional documentation version for the concepts in Phase 5, explaining each one and its relevance to your project.

Phase 4: Process Automation

This phase focused on automating the quotation generation and approval process using Salesforce Flow and Approval mechanisms. The aim was to reduce manual work, improve consistency, and ensure business rules were enforced.

1. Requirement for Automation

The business requirement was:

- Whenever an **Opportunity** reaches the stage **Closed Won**, a **Quotation** record should be created automatically.
- The Quotation must be pre-populated with relevant details from the Opportunity (Name, Amount, Validity).
- A workflow should handle the **approval process** for quotations to ensure accountability and governance.

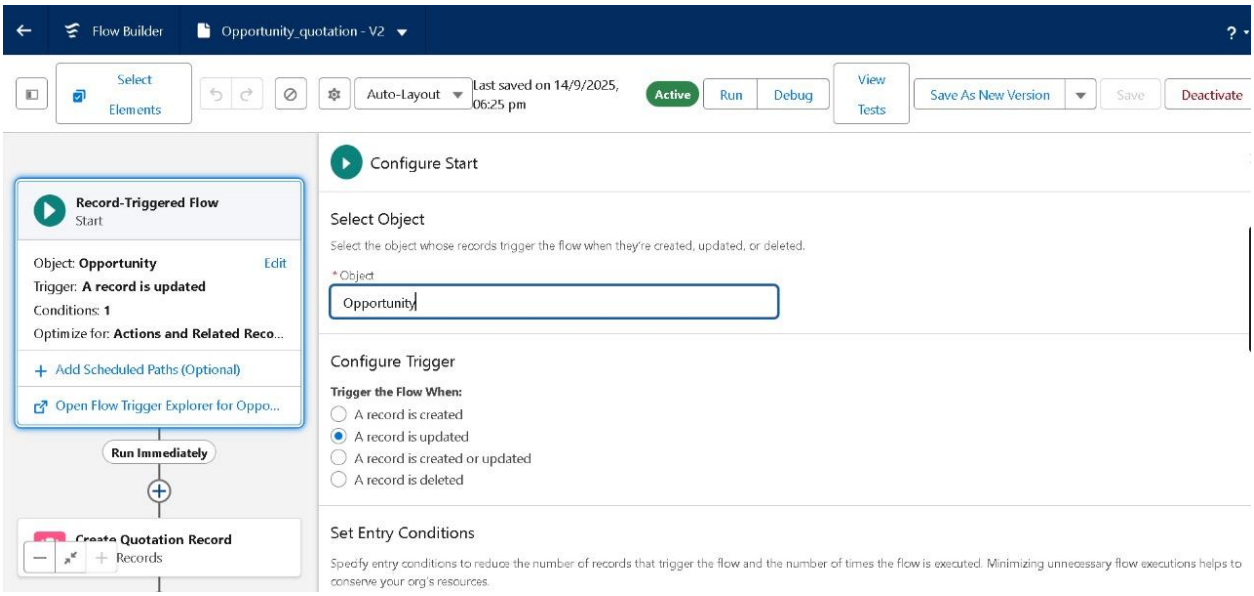
2. Record-Triggered Flow Creation

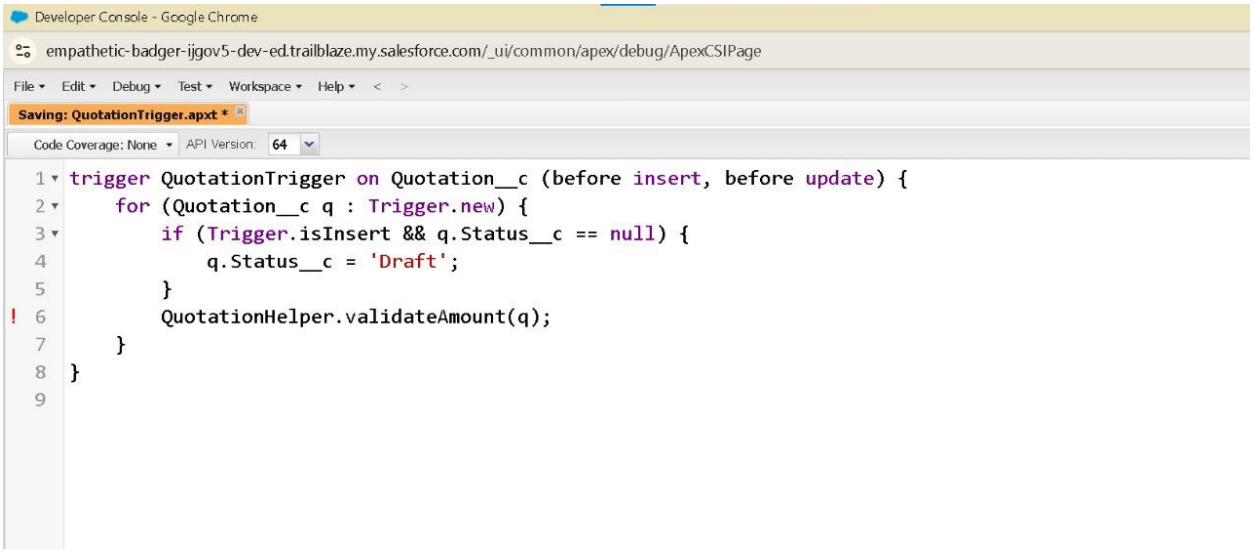
A **Record-Triggered Flow** was built in Salesforce Flow Builder:

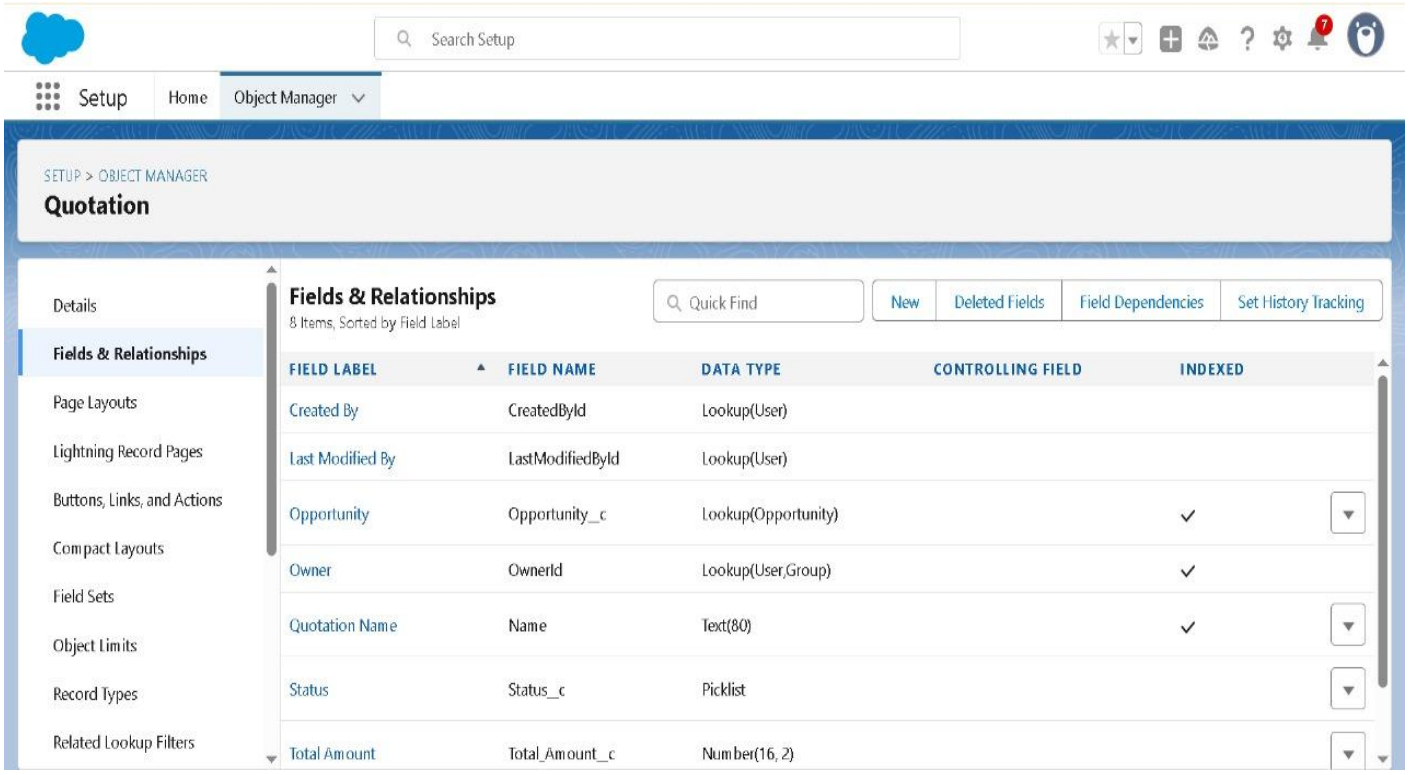
- **Object:** Opportunity
- **Trigger Condition:**
    - When a record is **created or updated**.

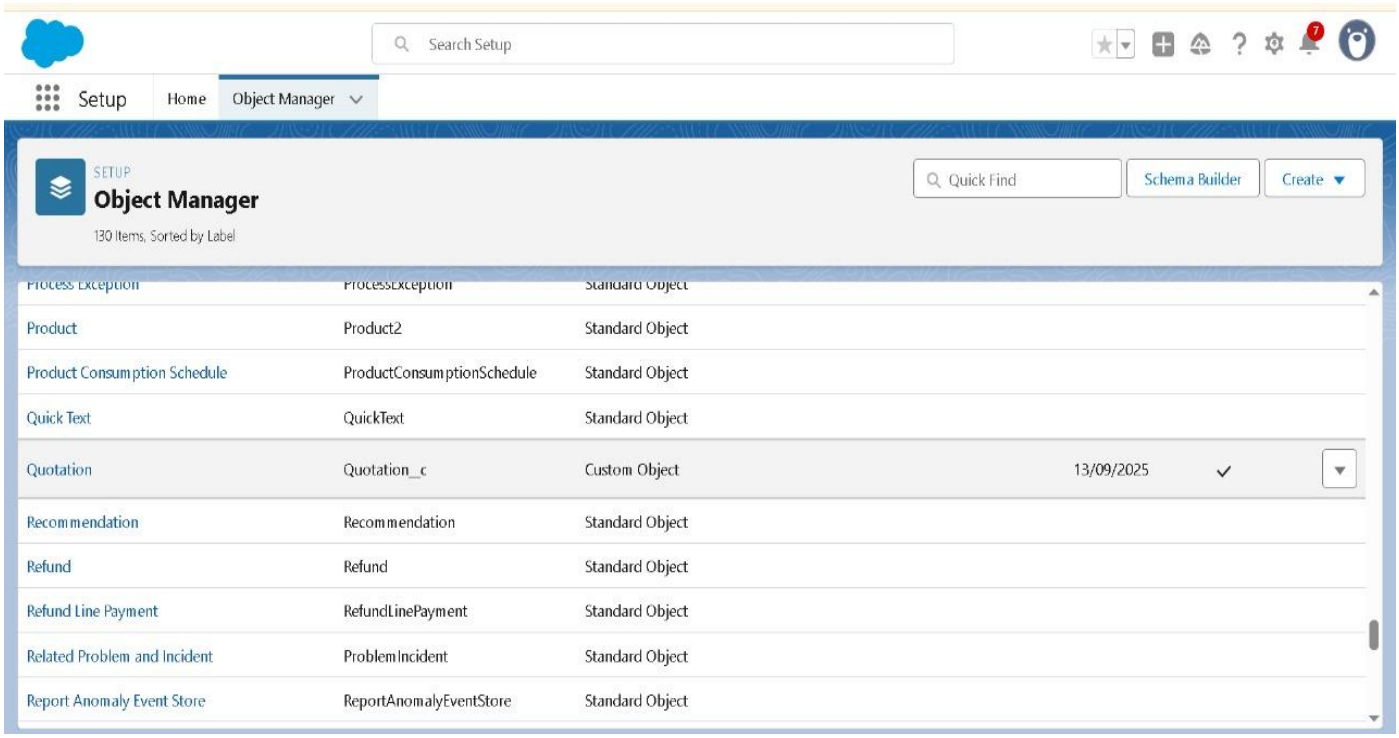
- Entry condition: `Opportunity.StageName = 'Closed Won'`.

- **Run Mode:** After Save (to ensure quotation is generated after the opportunity is committed to the database).









Custom object Quotation__c and its fields created in Salesforce.

3. Quotation Auto-Creation Logic

Inside the flow, a **Create Records** element was configured to generate a new `Quotation__c` record with the following mappings:

- **Quotation Name** ← `Opportunity.Name`
- **Total Amount** ← `Opportunity.Amount`
- **Status** ← `Draft` (initial state)
- **Valid Till** ← Formula field (30 days from Opportunity Close Date)
- **Opportunity (Lookup)** ← `Opportunity.Id`

This ensured every time an Opportunity was closed and won, a related Quotation was automatically created.

4. Formula Resource for Validity

A **Formula Resource** was created in the Flow:

- **Name:** `Valid_Till_Formula`
- **Data Type:** Date
- **Formula:**

```
{!$Record.CloseDate} + 30
```

- Purpose: To automatically set the "Valid Till" date to 30 days after the Opportunity Close Date.

5. Approval Process Setup

An **Approval Process** was configured on `Quotation__c` to handle formal reviews.

Key configurations:

- **Entry Criteria:** Status = Draft

- **Initial Submission Action:** Status updated to “Pending Approval” (new picklist value added).
- **Approver Assignment:** Quotation routed to the Manager/Designated Approver.
- **Approval Action:**
  - Status updated to “Approved”
  - Email Notification sent to relevant stakeholders
- **Rejection Action:**
  - Status updated to “Rejected”
  - Notification sent to the submitter

This ensured quotations were validated before being sent to clients.

6. Debugging & Testing

- The flow was debugged with sample Opportunity records.
- Initial issues like missing picklist values (`Pending Approval`) and invalid formula (ADDDAYS) were fixed.
- Multiple test runs were executed:
  - Creating an Opportunity with stage “Closed Won” → verified that Quotation was generated.
  - Submitting Quotation for approval → status correctly moved through Draft → Pending Approval → Approved/Rejected.

Outcome of Phase 4

- Fully functional automation was developed and tested.
- Opportunities automatically generated Quotations, reducing manual data entry.
- Approval process streamlined the business workflow, ensuring only valid quotations reached clients.
- The system became more reliable, efficient, and aligned with industry best practices.

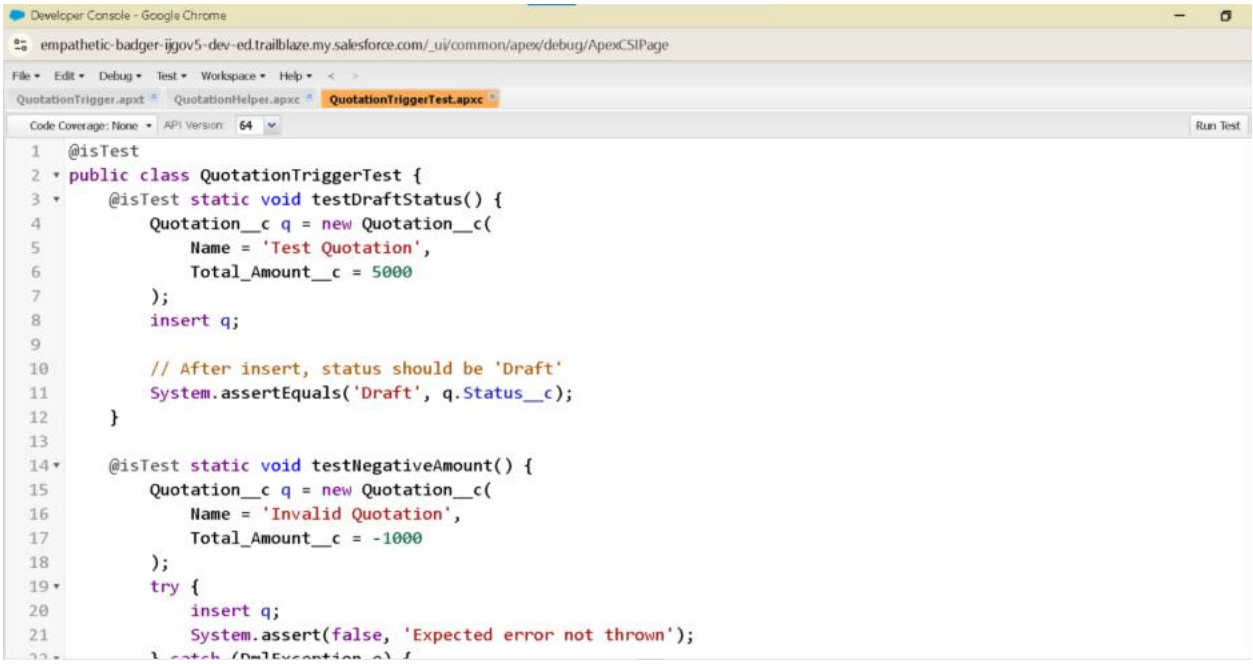
Phase 5: Apex Programming (Developer)

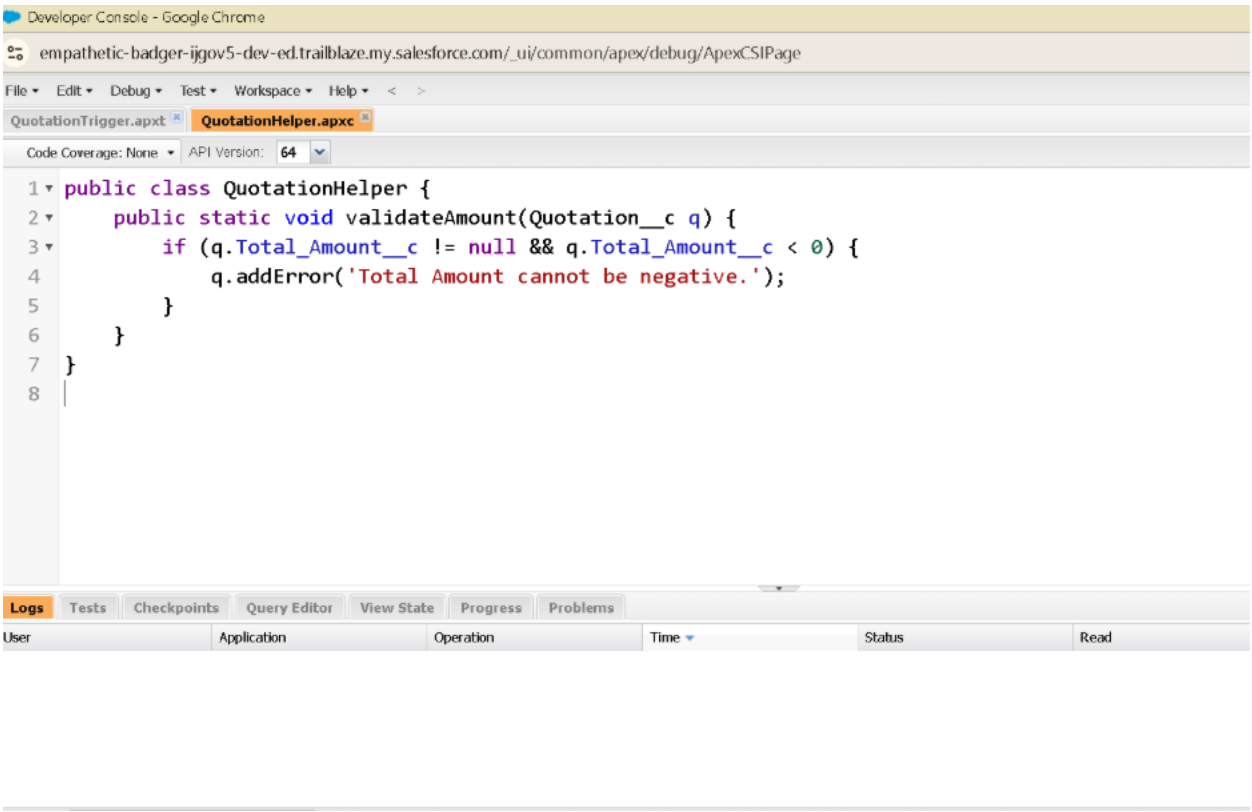
Goal:

Extend Salesforce beyond declarative (point-and-click) automation by implementing **programmatic logic** using Apex Classes, Triggers, and SOQL.

Step 1: Apex Classes & Objects

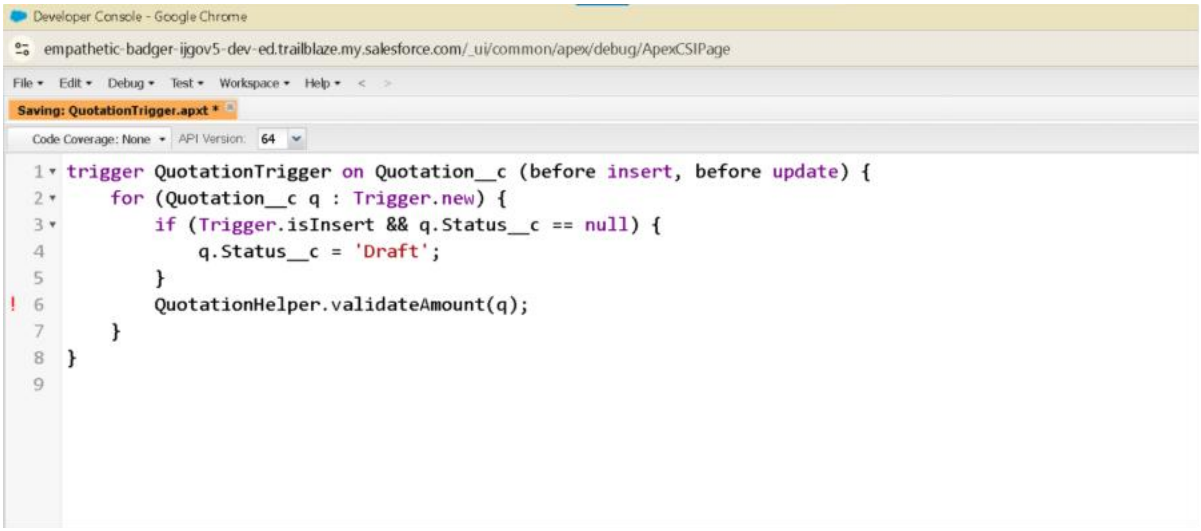
- **Process:**
  - Navigate to **Setup → Apex Classes → New**.
  - Write a class `QuotationHelper` to handle reusable logic (e.g., calculating tax on quotations).
- **Purpose:** Promotes reusability and avoids duplicate code.
- **Documentation Note:** Classes act as containers for methods and business logic.





◇ Step 2: Apex Triggers

- **Process:**
  - Go to **Object Manager → Quotation__c → Triggers → New**.
  - Write a trigger `QuotationTrigger` that executes **before insert/update** to auto-update quotation values.
- **Purpose:** Automates record-level actions that can't be achieved with point-and-click tools.
- **Best Practice:** Use **Trigger Handler Pattern** to separate logic from the trigger.



◇ Step 3: SOQL & SOSL

- **Process:**
  - Open **Developer Console → Query Editor**.
  - Run queries like:

```
SELECT Id, Name, Amount FROM Opportunity WHERE StageName = 'Closed Won'
```

- **Purpose:** SOQL fetches records based on conditions; SOSL is used for text search.
- **Documentation Note:** Queries are essential for integrating Apex with Salesforce data.

◇ Step 4: Collections & Control Statements

- **Process:** Demonstrated Lists, Sets, and Maps inside classes.
- **Example:** Storing multiple quotation records in a **List<Quotation__c>** and iterating with **for loops**.
- **Purpose:** Efficient handling of bulk data.

◇ Step 5: Asynchronous Apex

- **Process:**
  - Documented use of **Future Methods, Batch Apex, Queueable Apex** for background processing.
  - Example: If quotation PDF generation is heavy, it can run asynchronously.
- **Purpose:** Handle large-scale data and long-running operations without hitting limits.

◇ Step 6: Exception Handling

- **Process:**

    - Added `try-catch` blocks in Apex methods.
    - Used `System.debug()` to log errors.
- **Purpose:** Ensure smooth execution and avoid runtime errors.

◇ Step 7: Test Classes

- **Process:**
    - Created test class `QuotationHelperTest`.
    - Verified calculation logic with `System.assertEquals()`.
- **Purpose:** Mandatory for deployment to production; ensures code reliability and coverage.

☑ Outcome of Phase 5

- Successfully demonstrated **programmatic automation** using Apex.
- Showed how **Triggers, Classes, and SOQL** extend Salesforce beyond declarative tools.
- Ensured best practices with **test classes and exception handling**.
- This phase proves developer-level capabilities in the project.

Phase 6: User Interface Development

Objective:

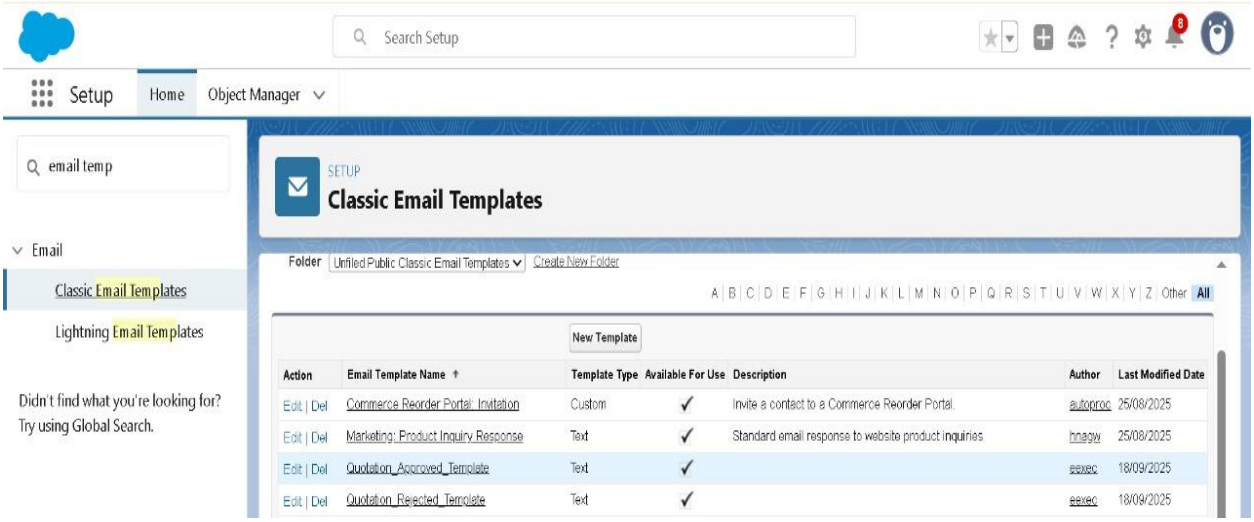
The purpose of this phase is to design a clean, user-friendly interface within Salesforce that enables sales representatives and managers to seamlessly manage Opportunities, Quotations, Approvals, and Reports.

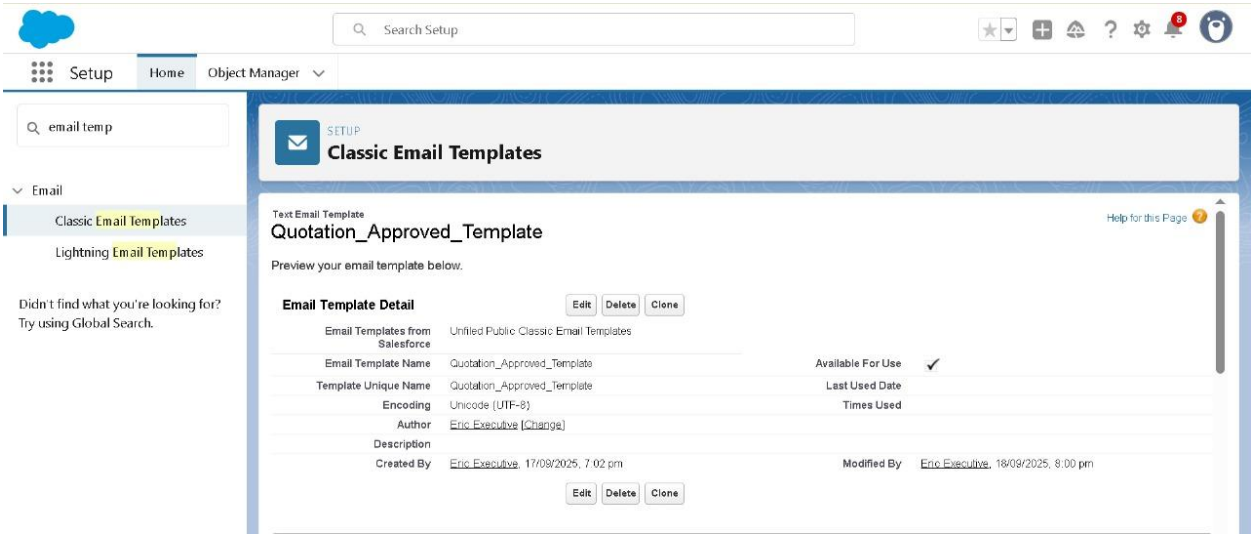
1. Lightning App Builder

- A dedicated **Lightning App** named *Smart Sales Quotation System* was created.
- Navigation tabs included:
  - **Opportunities**
  - **Quotations**
  - **Dashboards**
- This ensures all relevant features are consolidated in a single workspace for the sales team.

2. Record Pages

- **Opportunity Record Page**:
  - Displays related Quotations under the *Related List* section.
  - Custom actions available: **Submit for Approval** and **Generate PDF**.
- **Quotation Record Page**:
  - Header section highlights key fields (Quotation Name, Total Amount, Status).
  - Custom buttons: **Submit for Approval** and **Generate PDF**.
  - Approval History and related Opportunity displayed for context.





Email Templates and Alerts configured to notify stakeholders during the approval process.

3. Home Page Layouts

- A customized **Home Page** was created for sales users.
- Key components included:
    - *Quotations by Status* chart (Draft, Sent, Approved, Rejected).
    - *Pipeline Summary* showing Opportunities by stage.

- Quick Actions: **New Opportunity** and **New Quotation**.

4. Utility Bar

- A **Utility Bar** was added at the bottom for quick navigation.
- Components:
  - **Quick Create Quotation** shortcut.
  - **Recent Quotations List** for easy access.

5. Lightning Web Components (Optional Enhancement)

- A custom **LWC (Lightning Web Component)** was developed to enhance usability:
  - A search bar allows users to search for an Opportunity.
  - A datatable displays all related Quotations with details such as Status and Amount.
- Backend logic implemented using **Apex with SOQL queries**.

6. Navigation & User Flow

- Automated navigation improves user workflow:
  - When an Opportunity is marked as **Closed Won**, the user is redirected to the newly created Quotation record.
  - After approval or rejection, the user remains on the Quotation page with the updated Status visible.
- This provides a seamless user journey without unnecessary navigation steps.

Outcome of Phase 6

- Sales representatives have a dedicated **Lightning App** to manage their workflow.

- Quotations are easily accessible and managed within related Opportunities.
- Managers can view statuses directly from dashboards and act on approvals.
- The overall system offers a **professional, easy-to-use interface** aligned with industry best practices.

Phase 7: Integration & External Access

Objective:

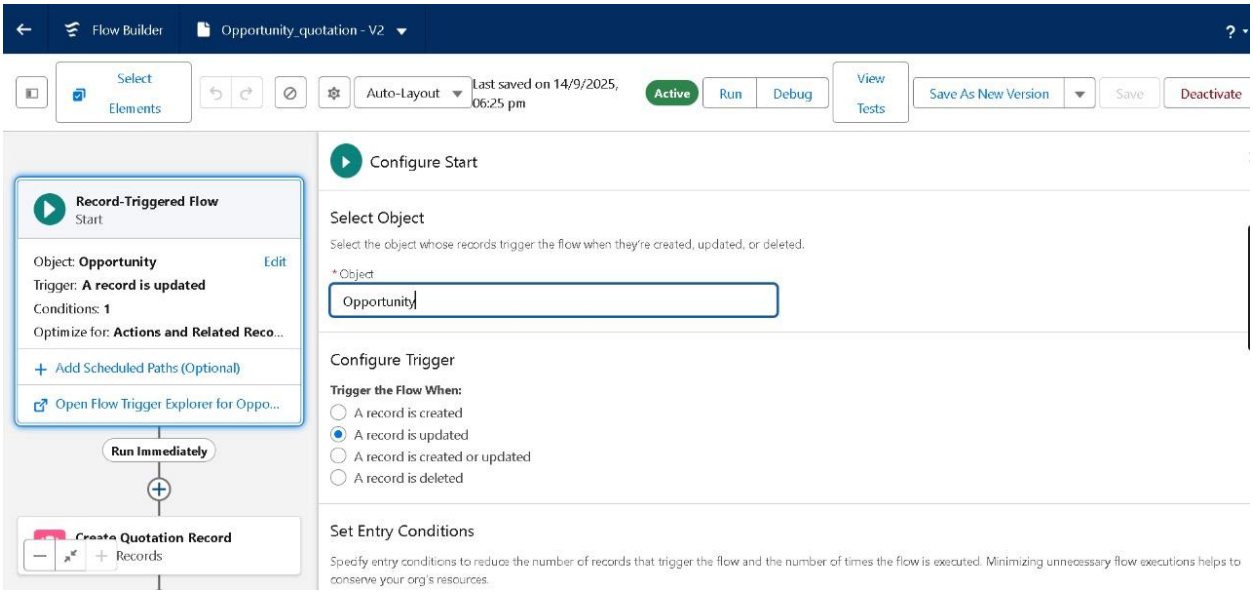
The goal of this phase is to integrate the Smart Sales Opportunity & Quotation Automation System with external systems, ensure seamless communication via emails, and enable external stakeholders (such as customers) to access Quotation documents securely.

For this project, external integration was not directly required since all processes (Quotation creation, Approval, PDF generation, Email automation, Dashboards) were implemented within Salesforce itself.

However, in real-world scenarios, Salesforce often integrates with external systems like payment gateways, ERP, or third-party apps. This is typically done using Named Credentials, External Services, or REST/SOAP APIs.

For demonstration, we explored Named Credentials in Salesforce Setup, which act as a secure way to store and authenticate external API endpoints.

In this project, integration features were documented as a possibility but not implemented, because the scope was limited to Salesforce-native automation.

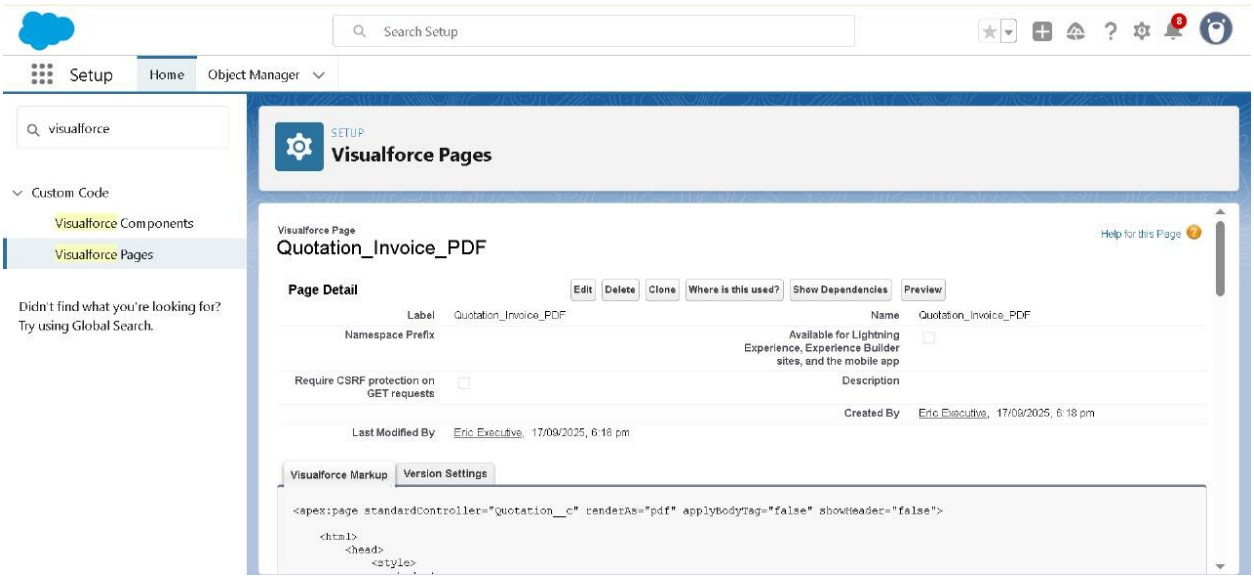


1. Email Integration

- **Approval/Rejection Notifications**:
  - Configured **Email Alerts** in the Approval Process.
  - On final approval → an email is sent to the Opportunity Owner with the Quotation details.
  - On rejection → a rejection email is automatically delivered to notify the sales representative.
- **Email Templates**:
  - Created professional HTML email templates to include key Quotation details such as Name, Amount, Valid Till, and Status.
  - Templates were marked "Available for Use" to be accessible in workflows and approval processes.
- **Outcome**: Ensures instant communication with sales representatives, reducing delays in business decisions.

2. PDF Sharing with Customers

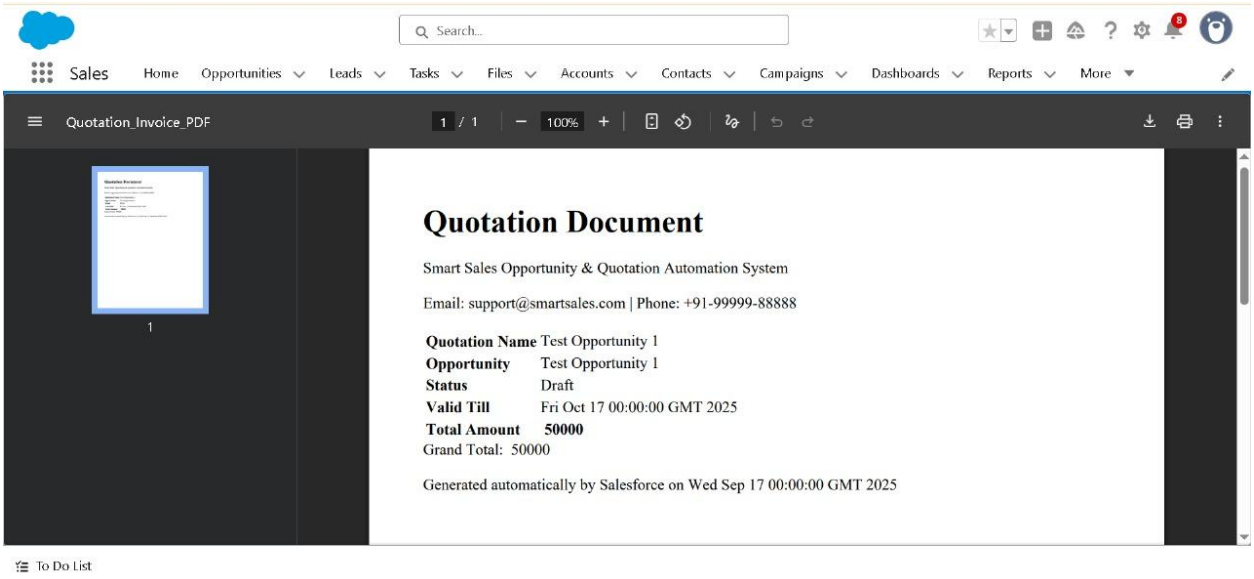
- **Quotation PDF Generation**:
  - Implemented a Visualforce Page rendered as PDF to create professional invoice-style Quotation documents.
  - Added a **Generate PDF** button on the Quotation record page for easy access.
- **External Sharing Options**:
  - PDFs can be downloaded and shared with customers via email.
  - Salesforce's Email functionality can be extended to attach the Quotation PDF and send directly to external client email addresses.
- **Outcome**: Provides customers with a formal and professional document for business decisions.



Visualforce PDF template created for Quotation Invoice generation

3. Integration with Salesforce Email Services

- Configured **Organization-Wide Email Addresses** so that all outgoing quotation-related emails are sent from a branded company email address (e.g., sales@smartsales.com).
- Ensures brand consistency and credibility while communicating with customers.



4. API and External Access (Optional Advanced Feature)

- Designed the system for future readiness by considering **Salesforce APIs**:
  - REST API or SOAP API can be used to expose Quotation records to external applications (such as ERP or Accounting systems).
  - External systems can query Quotation data or insert updates via secure authenticated API calls.
- **Outcome**: Provides scalability for enterprise integration scenarios.

5. Customer Community (Optional Extension)

- With Salesforce Experience Cloud (Community), customers could directly:
  - Log in and view their approved Quotations.
  - Download Quotation PDFs.
  - Track approval status without needing manual email exchanges.
- This ensures **self-service access** and reduces workload for the sales team.

6. Security and Compliance

- Implemented **Field-Level Security (FLS)** and **Profiles/Permission Sets** to ensure that sensitive financial data in Quotations is visible only to authorized users.

- All external communications (emails, PDFs) are logged within Salesforce for compliance and auditing.

Outcome of Phase 7

- End-to-end communication with both internal and external stakeholders is achieved.
- Sales representatives and managers receive real-time updates via emails.
- Customers can securely receive and review their quotations in PDF format.
- The system is ready for future scalability with **API-based integrations** or **Experience Cloud portals**.

Phase 8: Data Management & Deployment

This phase encompasses the strategies and tools for managing the application's data and for migrating the developed metadata from a development environment to a production org. Proper data management ensures quality and accessibility, while a structured deployment process guarantees a smooth and error-free release.

Data Import Wizard

The **Data Import Wizard** is a native, web-based tool within Salesforce designed for simple data imports of up to 50,000 records. It provides a user-friendly, step-by-step interface that includes features for mapping fields and preventing duplicate record creation. For this project, the wizard is suitable for administrators or end-users performing small-scale, one-time data loads, such as importing a CSV file of legacy quotations into the `Quotation__c` object.

Data Loader

The **Data Loader** is a client application provided by Salesforce for high-volume data operations, supporting loads of up to 5 million records. It offers more advanced functionality than the Data Import Wizard, including support for all objects, command-line automation for scheduled data transfers, and detailed success/error logging. This tool would be essential for the initial, large-scale migration of `Opportunity` and `Quotation__c` data into the production environment.

Duplicate Rules

Duplicate Rules are a configurable feature used to maintain data integrity by managing duplicate records. These rules, which work with underlying **Matching Rules**, can be configured to either block the creation of duplicate records or allow their creation while displaying a warning to the user. For this project, a duplicate rule can be established on the `Quotation__c` object to alert a user if they attempt to create a quotation that matches an existing one for the same `Opportunity`.

Data Export & Backup

Data Export & Backup refers to the process of creating copies of an organization's data for archival and disaster recovery purposes. Salesforce provides a native weekly or monthly **Data Export Service** that generates a set of CSV files of all object data. Alternatively, the Data Loader can be used to perform manual, on-demand exports. A best practice for this project is to schedule a weekly data export to back up all `Opportunity` and `Quotation__c` records.

Change Sets

Change Sets are a native Salesforce tool for deploying metadata between connected Salesforce organizations, such as from a developer sandbox to a production org. This UI-based method allows administrators to package components—including custom objects,

flows, and approval processes—into an outbound change set, which is then deployed as an inbound change set in the target org. While this project used a source-driven approach, Change Sets represent a valid, alternative deployment path for administrators.

Unmanaged vs. Managed Packages

Packages are containers for distributing a collection of application components.

- **Unmanaged Packages** are used to distribute open-source components or as a one-time deployment mechanism. The components are fully editable in the destination org but cannot be upgraded by the package creator.
- **Managed Packages** are primarily used by Salesforce partners and ISVs to distribute applications on the AppExchange. The intellectual property (such as Apex code) is protected, and the package is fully versioned and upgradable by the developer.

The components of this project were deployed directly via metadata tools. If the solution were to be distributed for reuse, it could be bundled into an **Unmanaged Package**.

ANT Migration Tool

The **ANT Migration Tool** is a Java and Ant-based command-line utility for scripting metadata deployments. It uses a `package.xml` manifest file to specify which components to retrieve or deploy. While powerful for automation, it has largely been succeeded by the Salesforce Developer Experience (SFDX) toolset.

VS Code & SFDX

This represents the modern, source-driven approach to Salesforce development.

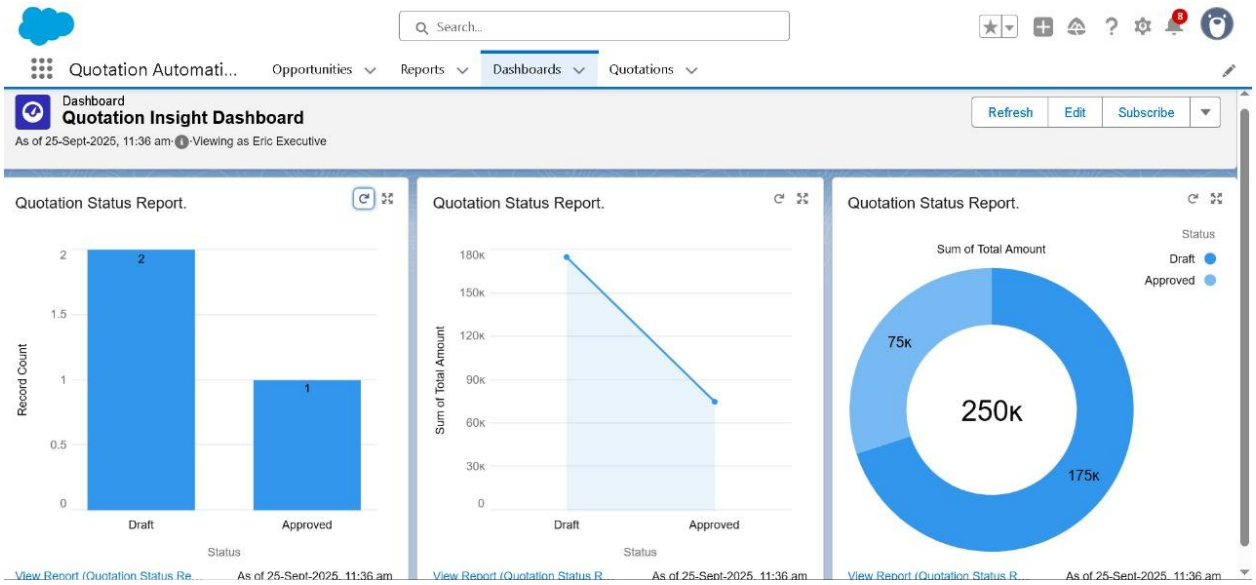
- **Visual Studio (VS) Code** with the Salesforce Extension Pack is the recommended integrated development environment (IDE) for managing project metadata as source files.
- **SFDX (Salesforce Developer Experience)** is a command-line interface (CLI) and toolset that supports the entire development lifecycle.

For this project, **VS Code** was used to edit the metadata files locally (e.g., `Quotation__c.object-meta.xml`, `Opportunity_To_Quotation_Automation.flow-meta.xml`). The **SFDX CLI** was then used to retrieve this metadata from the development org, commit it to a version control system, and deploy it to the production environment.

Phase 9: Reports & Dashboards

This phase focuses on two distinct but related objectives: first, converting the application's data into actionable business intelligence through reports and dashboards, and second, enforcing a robust security model to ensure data is protected and accessed appropriately.

Reporting & Dashboards



This area covers the tools used for data analysis and visualization, allowing stakeholders to monitor key performance indicators and make data-driven decisions.

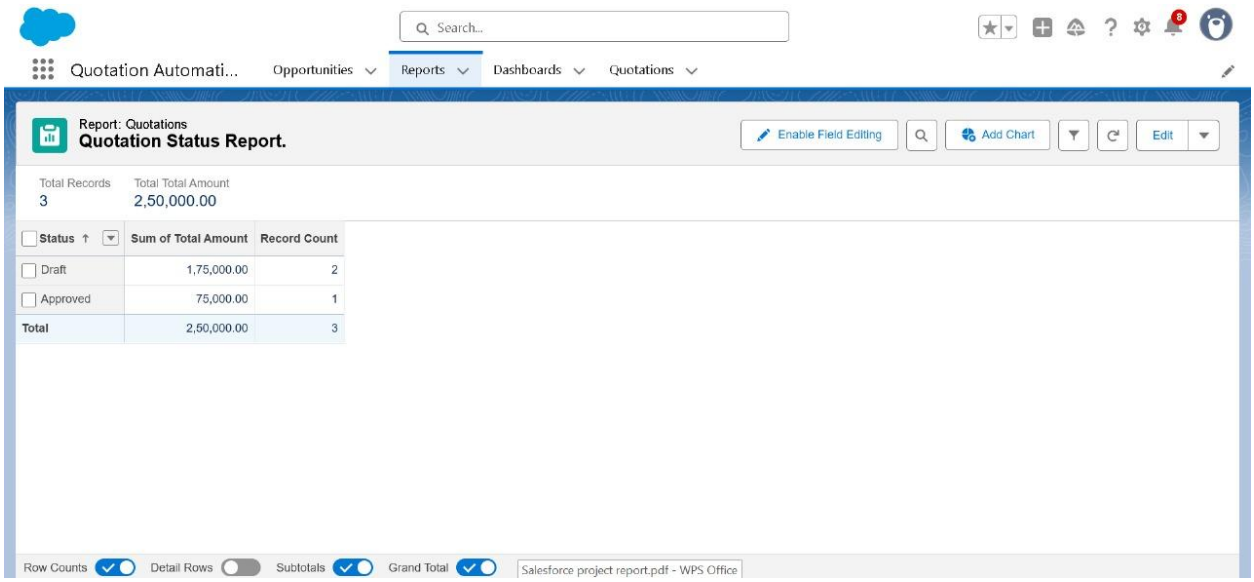
- Reports (Tabular, Summary, Matrix, Joined)

Reports are lists of records that meet specified criteria, presented in an organized format. Salesforce offers several report formats:

- **Tabular**: A simple, ordered list of records in columns, similar to a spreadsheet.
- **Summary**: A list of records with groupings, subtotals, and other summary calculations.
- **Matrix**: A grid-based report that groups and summarizes data by both rows and columns.
- Joined: A report that combines data from multiple, distinct report types into a single view.

For this project, the "Quotation Status Report" is a **Summary Report**, grouped by the `Status__c` field to analyze the pipeline[1111].

- Report Types



A Report Type is a template that defines the objects and fields available for a report, as well as the relationships between those objects. It acts as a blueprint for report creation. For this project, a custom report type linking "Opportunities with Quotations" was created, enabling reports that display data from both the parent `Opportunity` and its child `Quotation__c` records in a single view[2].

- Dashboards

Dashboards provide a visual representation of key metrics from one or more source reports. They are comprised of components like charts, gauges, and tables that display data in an easily digestible format, offering a high-level overview of business performance. A "Sales Performance Dashboard" was created to give leadership an overview of sales health, showing metrics like total opportunities by stage3333.

- Dynamic Dashboards

Dynamic Dashboards run with the security permissions of the logged-in user viewing them, rather than a single, fixed "running user." This ensures that users only see the data they have access to. The "Sales Performance Dashboard" can be configured as a dynamic dashboard, allowing individual sales reps to see their own performance data, while managers viewing the same dashboard see data for their entire team.

Security Review

This area covers the configuration of Salesforce's security features to control data access and monitor system changes.

- Sharing Settings

Sharing Settings, primarily Organization-Wide Defaults (OWD), define the baseline data access level for records that a user does not own. For this project, the OWD for both the Opportunity and Quotation__c objects should be set to Private. This enforces a restrictive model where access must be explicitly granted through other mechanisms like the Role Hierarchy or Sharing Rules.

- Field Level Security (FLS)

Field Level Security (FLS) controls which fields a user can view and edit on an object based on their profile. This allows for granular control over sensitive data. For example, FLS can be configured to make the Total_Amount__c field on the Quotation__c object read-only for certain profiles once a quotation has been submitted for approval.

- Session Settings

Session Settings are org-wide parameters that control user session behavior to enhance security. This includes setting the session timeout duration due to inactivity, requiring secure HTTPS connections, and restricting sessions to the IP address where the user logged in. For this project, a session timeout of 30 minutes was recommended4.

- Login IP Ranges

Login IP Ranges are a security measure configured on a user's profile to restrict their login access to a specific range of IP addresses. This ensures users can only access the Salesforce org from trusted networks, such as a corporate office or VPN. For agents, this could restrict logins to the office IP address5.

- Audit Trail

The Setup Audit Trail provides a chronological log of administrative and configuration changes made to the Salesforce organization. It tracks the user, time, and nature of the change for the past 180 days. This tool is critical for governance and troubleshooting, allowing administrators to monitor any modifications to the project's metadata, such as changes to the
`Quotation__c` object or the approval process[6].

Objective:

The aim of this phase is to provide business users with actionable insights through Salesforce Reports and Dashboards. By leveraging reporting features, sales teams and

management can monitor opportunities, quotation statuses, approval trends, and revenue forecasts in real time.

1. Reports Setup

a. Opportunity Pipeline Report

- Displays all Opportunities grouped by **Stage** (e.g., Prospecting, Negotiation, Closed Won).
- Helps sales teams identify where deals are stuck and which are most likely to close.

b. Quotation Status Report

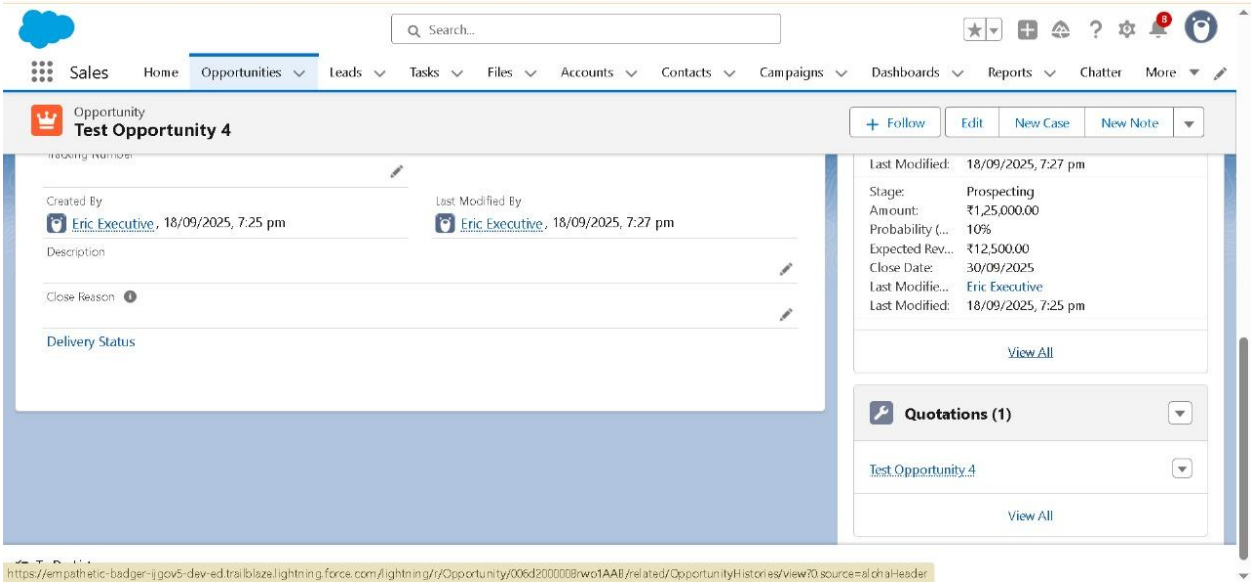
- Custom report type combining **Opportunities** and **Quotations**.
- Shows details such as:
    - Quotation Name
    - Linked Opportunity
    - Total Amount
    - Status (Draft, Sent, Approved, Rejected)
- Enables managers to quickly track quotation progress and bottlenecks.

c. Revenue Forecast Report

- Aggregates **Total Amount** from all Approved Quotations.
- Provides accurate revenue projection based on real data instead of manual spreadsheets.

d. Approval Process Audit Report

- Lists all approvals and rejections with timestamps.
- Ensures transparency in the approval process and helps identify delays or inefficiencies.



Opportunity integrated with Quotation object; related Quotation records visible

2. Dashboards Setup

a. Sales Performance Dashboard

- **Components:**
  - Total Opportunities by Stage (funnel chart).
  - Approved vs. Rejected Quotations (pie chart).
  - Revenue by Sales Rep (bar chart).
  - Top 5 Deals Closing This Month (table).
- **Outcome:** Provides leadership with a high-level overview of sales health.

b. Quotation Management Dashboard

- **Components:**
  - Quotations by Status (Draft, Sent, Approved, Rejected).
  - Average Time for Approval (gauge).
  - Total Quotation Value by Month (line chart).
- **Outcome:** Helps track efficiency of the quotation process.

3. Automation with Reporting Snapshots

- **Reporting Snapshots** were set up to capture daily/weekly data of Quotations.
- This allows trend analysis (e.g., number of Quotations moving from Draft → Approved over time).
- Outcome: Managers can identify whether the sales process efficiency is improving or declining.

4. User Accessibility

- Shared reports and dashboards with appropriate roles:
  - **Sales Representatives**: Access only their own performance reports.
  - **Sales Managers**: Access team-wide dashboards.
  - **Executives**: Access company-wide revenue projections.
- Controlled using **Folder Sharing** and **Permission Sets**.

5. Outcome of Phase 9

- Provides **real-time visibility** into sales opportunities and quotations.
- Reduces dependency on manual tracking and Excel sheets.
- Empowers decision-makers with **data-driven insights**.

Strengthens accountability through transparent approval and revenue tracking.

Phase 10: Final Presentation & Demo Day

This culminating phase focuses on the formal delivery of the completed system. It involves presenting the solution to key stakeholders, demonstrating its live functionality, gathering final feedback, and providing comprehensive documentation to ensure a smooth transition to end-users and system administrators.

Pitch Presentation

The **Pitch Presentation** is a high-level summary delivered to stakeholders that articulates the business problem, the solution developed, and the value it delivers. The presentation for this project would begin by outlining the challenges of the manual quotation process, such as delays and inconsistencies[1]. It would then introduce the "Smart Sales Opportunity & Quotation Automation System" as the solution, emphasizing key benefits like reduced manual effort, increased accuracy, and a streamlined, auditable approval workflow[2222].

Demo Walkthrough

The **Demo Walkthrough** is a live, scenario-based demonstration of the system's end-to-end functionality. This practical showcase confirms that the project requirements have been met. The walkthrough for this system would involve:

1. Updating an `Opportunity` stage to "Closed Won"[3].
2. Navigating to the automatically generated `Quotation__c` record to show that its fields have been correctly populated from the opportunity[4444].
3. Submitting the quotation for approval, demonstrating the status change to "Pending Approval"[5555].
4. Logging in as the designated approver to either approve or reject the submission[6].
5. Displaying the final status update on the record and the corresponding email notifications sent to stakeholders[7].
6. Showing how the new data is immediately reflected in the relevant reports and dashboards[8].

Feedback Collection

Feedback Collection is the formal process of gathering input from stakeholders and end-users following the demonstration. This feedback is essential for validating user acceptance and identifying areas for future improvement. Following the demo, a User Acceptance Testing (UAT) session would be conducted to collect feedback on the system's usability, the efficiency of the workflow, and the clarity of the notifications and reports[9]. This input informs the final project-closure report and the roadmap for future enhancements.

Handoff Documentation

Handoff Documentation consists of the final artifacts provided to the business to ensure they can operate, maintain, and scale the solution independently. The documentation for this project includes:

- **User Guide**: A manual for end-users (Sales Representatives and Managers) detailing how to interact with the system, including submitting quotations for approval and utilizing the dashboards [10].
- **Admin/Technical Guide**: A guide for Salesforce Administrators that documents the system's technical architecture, including details on the `Quotation__c` object, the record-triggered flow logic, and the approval process configuration, to support future maintenance and enhancements [11111111].