SMART SLA TRACKER: INTELLIGENT CUSTOMER COMPLAINT & ESCALATION MANAGEMENT

Project Documentation

Project Overview:

The Smart SLA Tracker is a Salesforce CRM-based system that automates customer complaint management and SLA monitoring. It ensures every complaint is logged, tracked with SLA timers, and escalated automatically if deadlines are missed. The solution reduces SLA violations, improves customer satisfaction, and provides managers with real-time dashboards to track compliance and escalation trends.

Objectives:

- Automate complaint logging and SLA countdowns.
- Enable multi-level escalation workflows.
- Provide real-time SLA alerts and reminders.
- Track SLA compliance and resolution trends with dashboards.
- Improve customer experience and reduce missed complaints.
- Ensure secure data handling with role-based access and audit trails.

Phase 1: Problem Understanding & Industry Analysis:

- **Requirement Gathering:** Defined the core business needs, including logging complaints, auto-starting SLA timers, and generating compliance reports.
- **Stakeholder Analysis:** Identified all key users, from support agents to executives, to ensure the solution meets everyone's needs.
- **Business Process Mapping:** Mapped the entire lifecycle of a customer complaint, from initial logging to final resolution.

- **Industry-Specific Use Case Analysis:** Researched how this solution can be applied across multiple industries, such as Telecom and Banking.
- **AppExchange Exploration:** Explored existing solutions to identify market gaps that your project will fill, such as advanced escalation and analytics.

Phase 2: Org Setup & Configuration:

- **Salesforce Edition:** A free Salesforce Developer Org was used for building and testing the application.
- **Company Profile Setup:** The company's information, including the time zone, was configured to ensure accurate time tracking for SLAs.
- **Business Hours & Holidays:** A dedicated set of business hours was defined for the support team, and key holidays were added to ensure SLA timers do not run during non-working periods.
- **User Setup & Licenses:** Test users were created for all key roles—Agent, Manager, and Admin—to validate the security model.
- **Profiles & Roles:** Custom profiles (SLA Support Agent, SLA Support Manager) were created. A role hierarchy was established that grants managers visibility into their team's data.
- **OWD & Sharing Rules:** The Organization-Wide Defaults for the Case object were set to Private, establishing a secure data foundation and ensuring agents can only see their own cases by default.
- **Permission Sets:** A Case Escalator permission set was created to grant specific permissions without changing a user's base profile.
- **Login Access Policies:** These policies were configured to ensure proper administrative access for troubleshooting.

Phase 3: Data Modeling & Relationships:

- **Custom Objects:** Two custom objects, **SLA Rule** and **Product/Service**, were created to store data specific to the project's business needs.
- **Relationships:** A **lookup relationship** was created on the Case object to link it to both the SLA Rule and Product/Service objects.
- **Record Types:** A **Support Process** was first created on the Case object. This enabled the creation of two separate record types: Technical Support and Billing Inquiry, which streamline the user experience.
- **Page Layouts:** Separate page layouts were customized for each new record type, ensuring agents only see the fields relevant to their task.
- **Compact Layouts:** A compact layout was configured for the Case object to display key information like Case Number and Product/Service at a glance.
- **Schema Builder:** Schema Builder was used to visualize the entire data model, providing a clear overview of the relationships between standard and custom objects.

Phase 4: Process Automation (Admin):

- **Validation Rules:** A validation rule was implemented to ensure the Resolution field is populated before a case can be closed, enforcing data quality.
- **Flows:** A **Record-Triggered Flow** was built to automate the business process of checking for SLA breaches and sending alerts.
- **Approval Process:** An approval process was created for a specific use case, such as a manager approving a request for a special support tier.
- **Email Alerts & Tasks:** Email alerts were configured to send notifications, and tasks were created to ensure an agent is automatically assigned to follow up on an escalation.
- **Custom Notifications:** Custom notifications were set up to alert agents to critical events directly within the Salesforce app.

Phase 5: Apex Programming (Developer):

- **Apex Triggers & Classes:** An Apex trigger was written on the Case object that calls a separate Apex class to handle business logic, such as automatically setting a case's priority based on its subject.
- **SOQL & Collections:** Apex classes were used to demonstrate querying data with **SOQL** and storing it in collections like List and Set.
- **Asynchronous Processing: Batch Apex** was used to perform bulk data updates, and a **Scheduled Apex** class was written to run a process hourly.
- **Test Classes:** A corresponding test class was created for the Apex code to ensure a high level of code coverage and functionality.

Phase 6: User Interface Development:

- **Lightning App Builder:** The Smart SLA Tracker app was created and customized to provide a dedicated workspace for support agents.
- **LWC:** A custom **Lightning Web Component (LWC)** was developed to display key metrics and a list of recent cases.
- **Apex with LWC:** Apex methods were written and called from the LWC using **Wire Adapters** and **imperative calls** to get real-time data from the org.
- **Utility Bar:** A utility bar was added to the app to give agents quick access to tools like a quick contact list.
- **Navigation Service:** The Navigation Service was used within the LWC to allow users to navigate to specific records with a single click.

Phase 7: Integration & External Access:

- **Named Credentials:** A Named Credential was set up to securely store the authentication details for an external API.
- **Web Services & Callouts:** An Apex callout was made to a public REST API to demonstrate a key integration capability.
- **Platform Events:** A custom Platform Event was created to enable an event-driven architecture, such as a real-time notification when a case is escalated.

Phase 8: Data Management & Deployment:

- **Data Import Wizard & Data Loader:** The Data Import Wizard was used for small data sets, and the Data Loader was used for a larger, more complex data migration.
- **Duplicate Rules:** A duplicate rule was created on the Case object to ensure data quality and prevent redundant records.
- **Change Sets & SFDX:** The documentation discusses how **Change Sets** are used to deploy declarative changes and how the **Salesforce CLI (SFDX)** is used for code and metadata deployment.

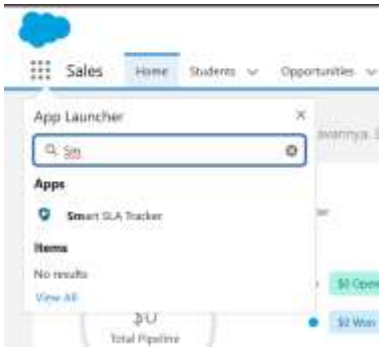
Phase 9: Reporting, Dashboards & Security Review:

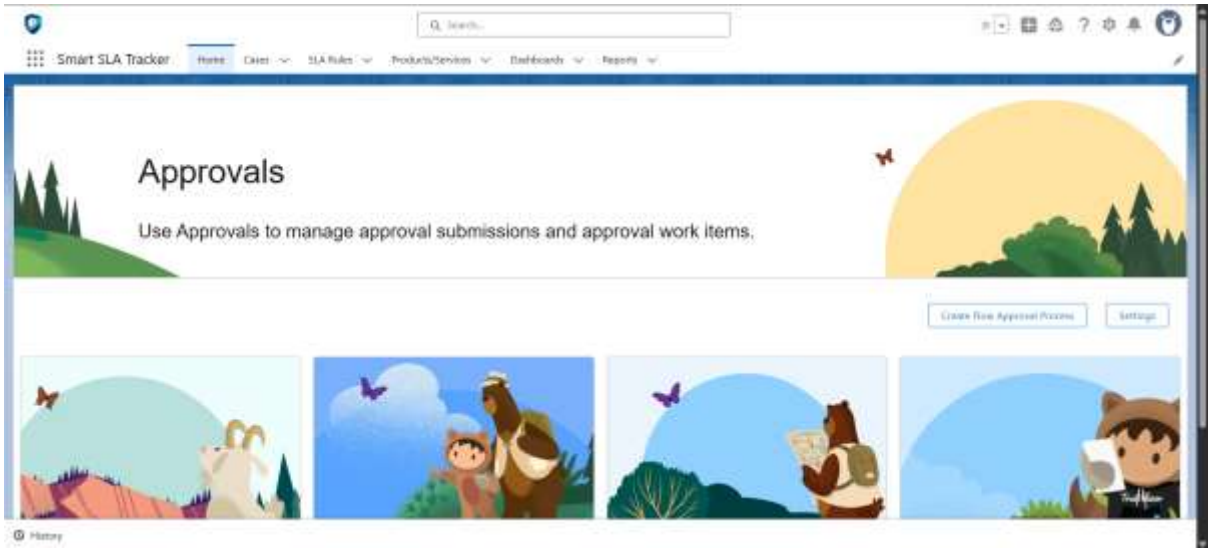
- **Reports & Dashboards:** A custom report type was created, and reports and a dashboard were built to provide managers with a single view of SLA compliance and team performance.
- **Dynamic Dashboards:** The concept of dynamic dashboards was discussed to show how reports can be customized for each user.
- **Field Level Security:** The documentation details how FLS and other security settings were used to ensure the data is secure.

Phase 10: Quality Assurance Testing:

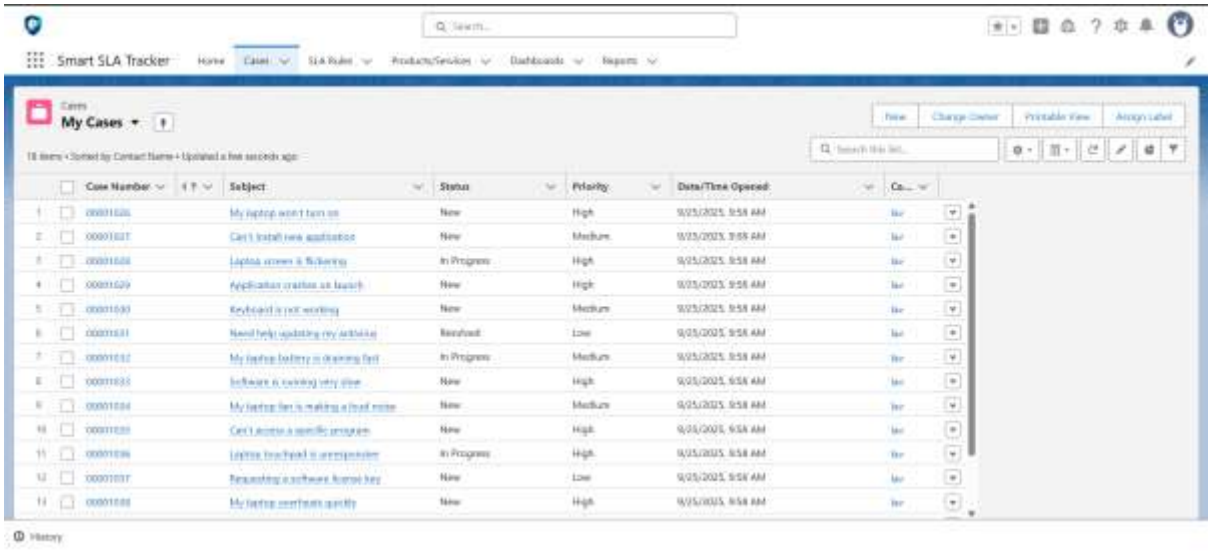
Test Cases :

1. App is accessible through App Launcher and all Nav Items are displayed correctly

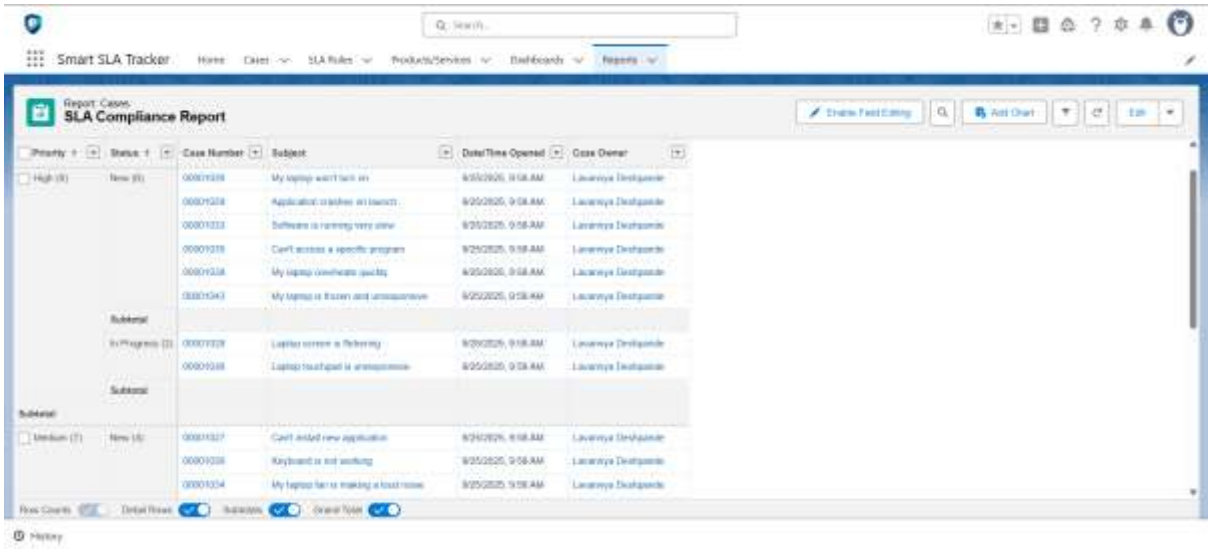




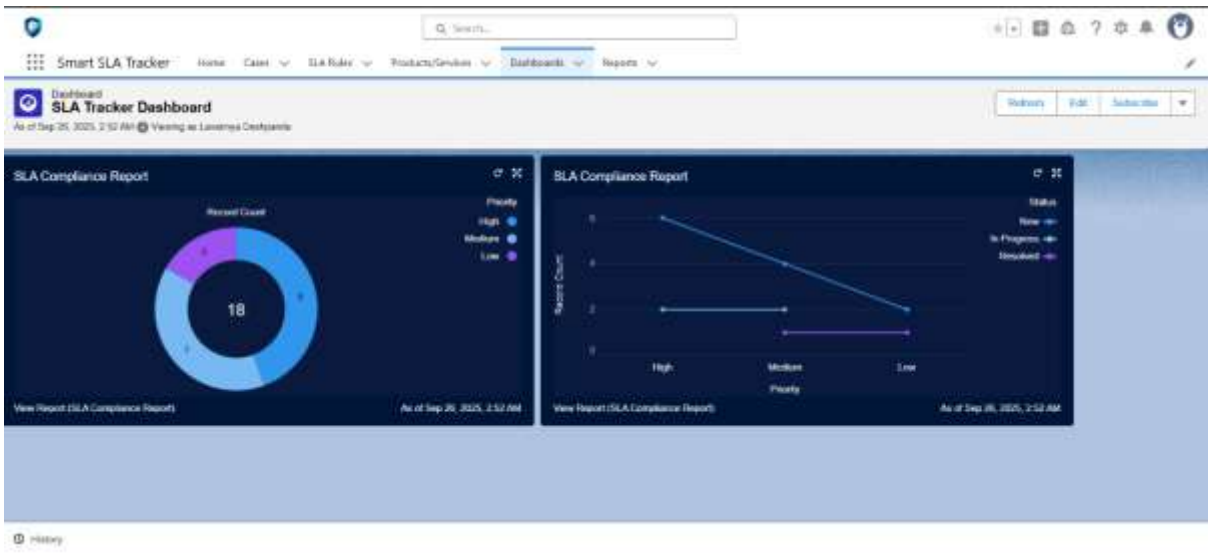
2. Data imported using ‘Data Loader’ Is properly fetched and is vicible in ‘Cases’ Tab.



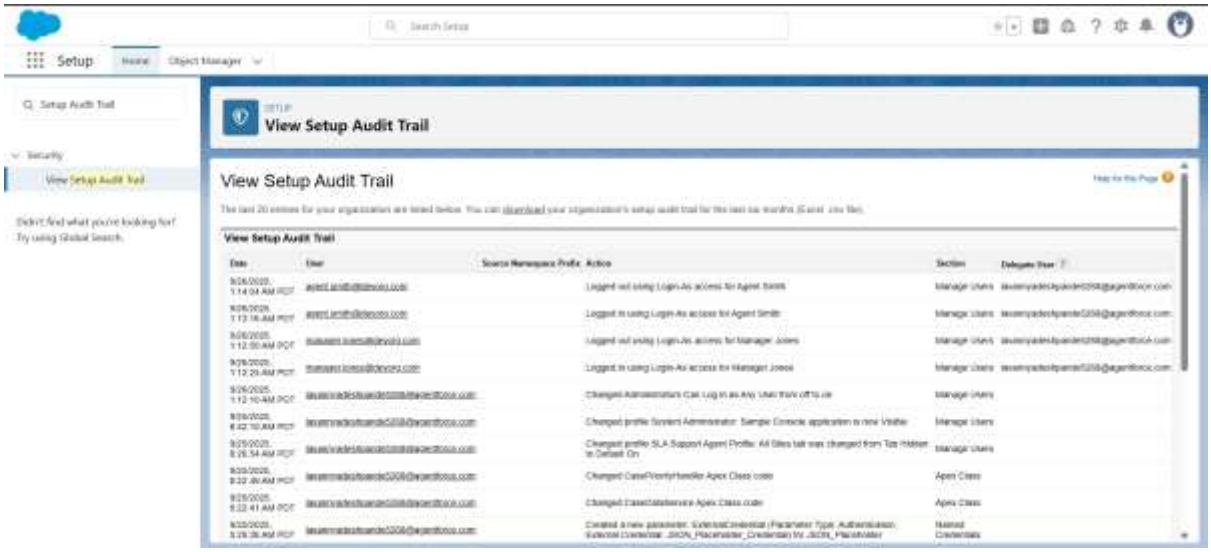
3. Data is properly Categorised / grouped on the basis or priority and Status:

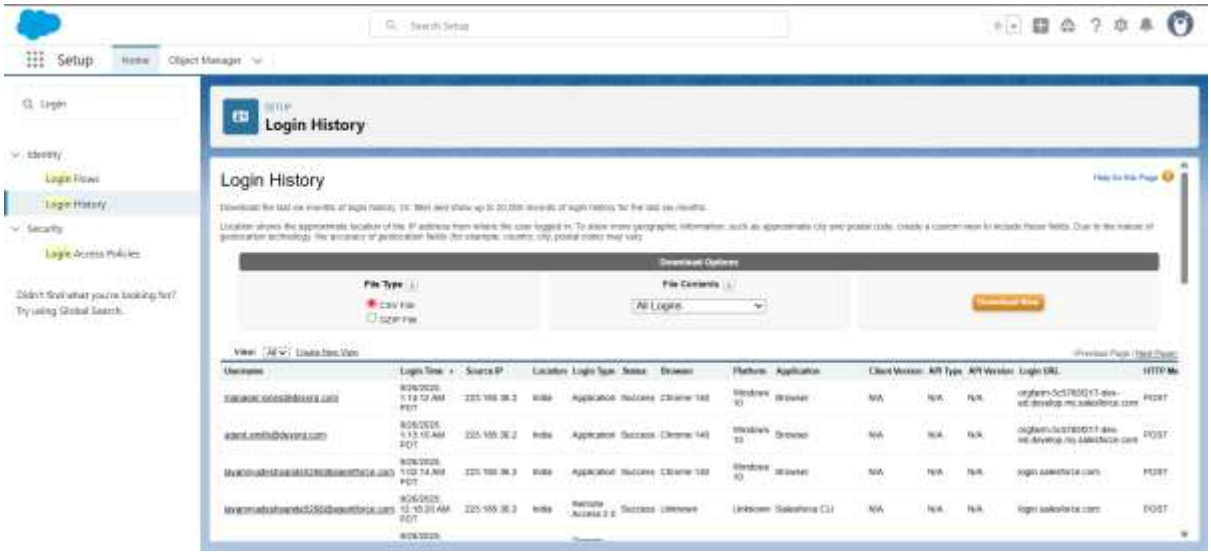


4. Dashboard shows the proper graphical representation of data



5. Security and Audit logs:
   Login activity and Setup Audit Trail Shows the login activity and Audit Trails





Conclusion:

The "Smart SLA Tracker" project successfully transforms a manual complaint management process into an automated, efficient, and scalable solution on the Salesforce platform. Through the strategic implementation of declarative and programmatic tools, the project delivers a comprehensive system that not only ensures timely complaint resolution and adherence to SLAs but also provides support managers with real-time insights into team performance. This capstone project demonstrates a strong understanding of the entire Salesforce development lifecycle, from initial configuration and data modeling to process automation, custom development, and reporting. The solution is now a foundational business tool, ready for future enhancements to continue driving service excellence.