ANGELES UNIVERSITY FOUNDATION

Angeles City

Tours & Travels Platform Development

Using CRM-Salesforce

Documentation

A Capstone Project

Submitted as Part of the Internship Program

of the College of Computer Studies

Angeles University Foundation

By

Naguit, Kate Louise C.

July 18, 2025

Project Overview

The Tours and Travels CRM is a Salesforce-based system made to help travel agencies work faster and serve customers better. It solves problems like slow booking processes, manual data tracking, and delayed communication by putting all customer and travel information in one place. The system has key features such as booking management, travel package tracking, payment monitoring, and customer records. It also has automated tools for booking confirmations, payment updates, and approvals, which save time and reduce mistakes. With real-time updates on available travel packages, easy-to-use pages for different types of users (like agents and managers), and clear reports and dashboards, the CRM helps the team make better decisions and respond to customers quickly. Thanks to this system, travel agencies can work more smoothly, improve data accuracy, and give customers a better experience.

Objectives

The main goal of building the Tours and Travels CRM is to help travel agencies manage their customers and bookings more easily and efficiently. By having all customer details, bookings, payments, and travel packages in one system, the CRM makes work faster and more organized. It aims to reduce manual tasks, avoid errors, and speed up processes like confirming bookings and updating payments. With automated tools and real-time information, travel agents can provide better and quicker service, leading to happier customers. Overall, the CRM is designed to improve daily operations, support data-driven decisions, and help the business grow smoothly.

Phase 1: Requirement Analysis & Planning

In this phase, foundational activities were conducted to ensure that the CRM system aligned with the actual needs of the Tours & Travels business. The objective was to identify

business challenges, define project goals, establish the data and security models, and outline a detailed development plan with clearly defined milestones.

Understanding Business Requirements

The initial step involved gathering user needs and identifying existing problems in the current operations. The business required a centralized CRM system capable of handling travel bookings, managing customer information, tracking employee tasks, monitoring payments, and processing guest feedback. Key issues that needed to be addressed included:

- Manual handling of bookings and payments leading to delays and errors
- Limited visibility into revenue trends and staff performance
- Absence of an automated reminder or notification system
- Lack of structured data collection for feedback and guest preferences
- Inefficient approval processes for booking cancellations

These pain points were translated into actionable system features such as automation workflows, real-time dashboards, and user-specific access controls.

Defining Project Scope and Objectives

The scope of the project included the full development of a custom Salesforce CRM application tailored to a Tours & Travels business. The objectives were as follows:

- Develop a modular CRM system with custom objects for Customers, Bookings, Travel Packages, Guests, Employees, Payments, and Feedback.

- Automate key processes such as booking confirmation, payment tracking, task creation, and approval workflows.
- Enhance data visibility through custom reports and dashboards.
- Design a user-friendly interface using Lightning App Builder and Dynamic Forms.
- Integrate custom logic through Apex triggers, batch processes, and Lightning Web Components for extended functionality.
- Ensure data validation and consistency through business rules and conditional logic.

Designing the Data Model and Security Model

A comprehensive data model was designed to define how the various CRM entities interact. This included:

- Custom objects with relationships such as master-detail (e.g., Booking to Booking Guest) and lookup (e.g., Booking to Travel Package).
- Fields configured with appropriate data types, validation rules, and picklists to maintain accuracy and integrity.

The Security Model was structured based on role hierarchy and user access needs. Key elements included:

- User roles such as Travel Agent, Travel Manager, and Finance Admin
- Custom profiles to restrict or permit access to specific objects and fields
- Permission sets for additional flexibility
- Record-level access configured using Organization-Wide Defaults (OWDs), Sharing Rules, and Manual Sharing where necessary

This model ensured that users only had access to relevant records and operations based on their roles.

Creating Project Roadmap & Milestones

A development roadmap was created, dividing the project into distinct, trackable milestones. Each phase built upon the previous, ensuring a smooth and logical development process. The roadmap was structured as follows:

- Phase 1: Requirement Analysis & Planning
  - — Understand business needs, define scope, plan project roadmap
- Phase 2: Salesforce Development – Backend & Configurations
  - — Create custom objects, configure validation, set automation (Flows, Workflow Rules, Apex)
- Phase 3: UI/UX Development & Customization
  - — Configure navigation, dynamic forms, app pages, reports, dashboards, user management
- Phase 4: Testing, Deployment, and Final Documentation
  - — Conduct test case design and execution, user training, data import, deployment, and documentation

Each milestone included subtasks with defined deliverables and timelines to maintain development efficiency and ensure project alignment with business objectives.

Phase 2: Salesforce Development - Backend & Configurations

In this phase, the backend structure and automation logic of the Tours & Travels CRM were developed and configured. It began with setting up the Salesforce environment by creating a developer account tailored to the project needs. This enabled access to key development tools such as Object Manager, Developer Console, and Setup functionalities.

Custom objects such as Customer Info, Booking, Travel Package, Booking Payment, Booking Guest, Employee, and Feedback were created to represent the core entities of the system. Each object was configured with appropriate fields and relationships, including lookup and master-detail links, to maintain data integrity and support functional interdependencies.

To enforce business logic and ensure data accuracy, a range of Validation Rules were implemented: for example, ensuring correct phone number formats, enforcing age limits, and mandating values based on conditional input. Field dependencies were also configured to guide users through relevant data input based on selections, such as filtering city options based on country selection.

Automation played a key role in streamlining workflows. A combination of Flows, Workflow Rules, and Process Builder logic was implemented:

- A record-triggered flow restricted adding more guests than allowed by a booking.
- A workflow rule automatically created follow-up tasks when a trip was marked as completed.
- Process Builder logic updated the booking status to "Confirmed" once the associated payment was marked "Completed."

An Approval Process was configured to manage booking cancellations, including automated email notifications using custom templates for submission, approval, and rejection outcomes. This ensured accountability and transparency throughout the review process.

On the development side, Apex Triggers and Handler Classes were created to automate backend operations. These included:

- Automatically generating Booking Payment and Booking Guest records when a new booking is created.

- Setting default values and enforcing business flows programmatically.

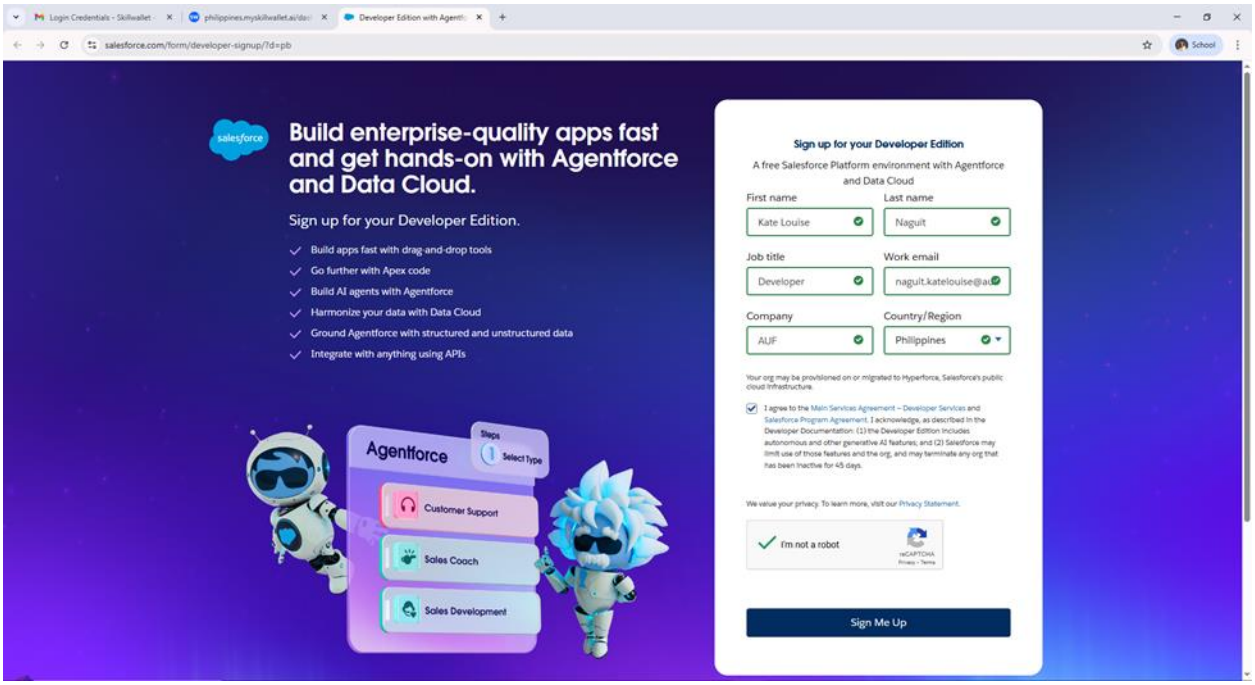
Additionally, Asynchronous Apex was implemented to handle performance-critical tasks. This included:

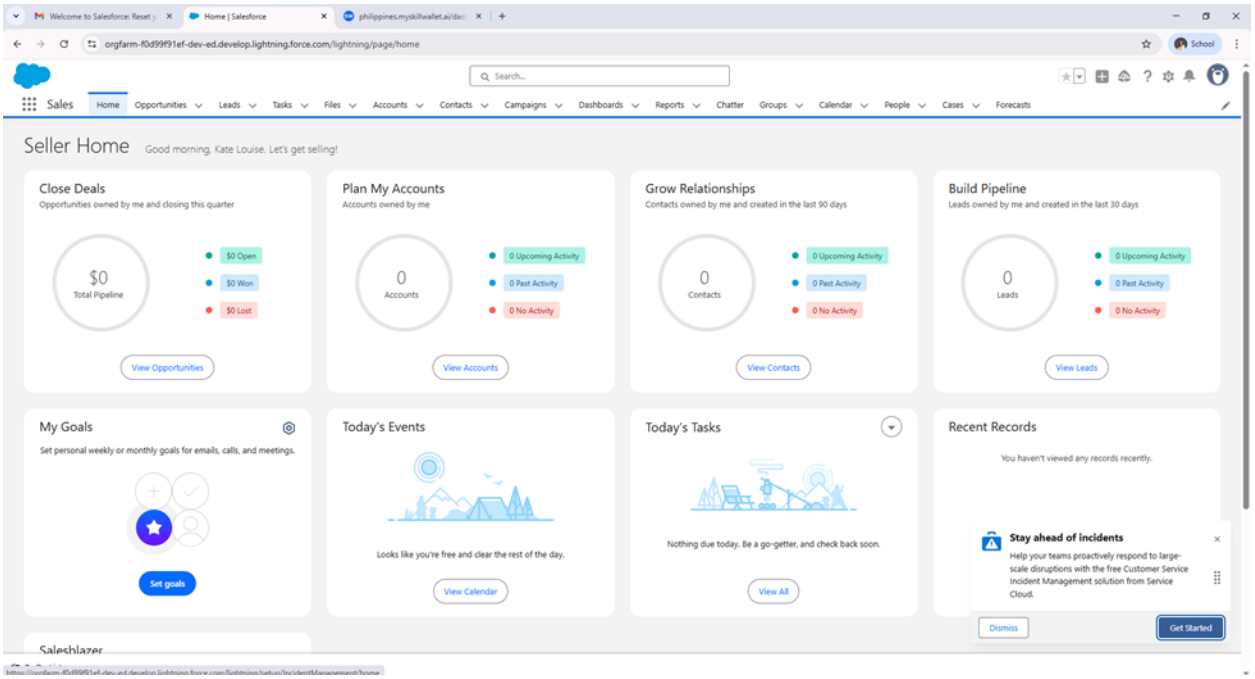
- A @future method for sending booking confirmation emails.
- A Queueable Apex class for sending travel reminders.
- A Batch Apex class for sending pending payment notifications, all scheduled via Schedulable Apex classes using CRON expressions.

This phase established a robust and scalable backend for the CRM system by combining declarative configurations with custom Apex development, resulting in a system that is efficient, accurate, and aligned with real-world business processes.

Milestone 1: Salesforce Account Setup

I started by creating my Salesforce account and choosing the edition that fit my project needs. I signed up through Salesforce's official developer website, providing my personal and academic details, including my name, email, and college name as the organization. This initial setup was important to build a strong foundation before moving on to customizing the system. This gave me access to the Salesforce Setup area, where I could start configuring objects, setting up automations, and developing custom logic.

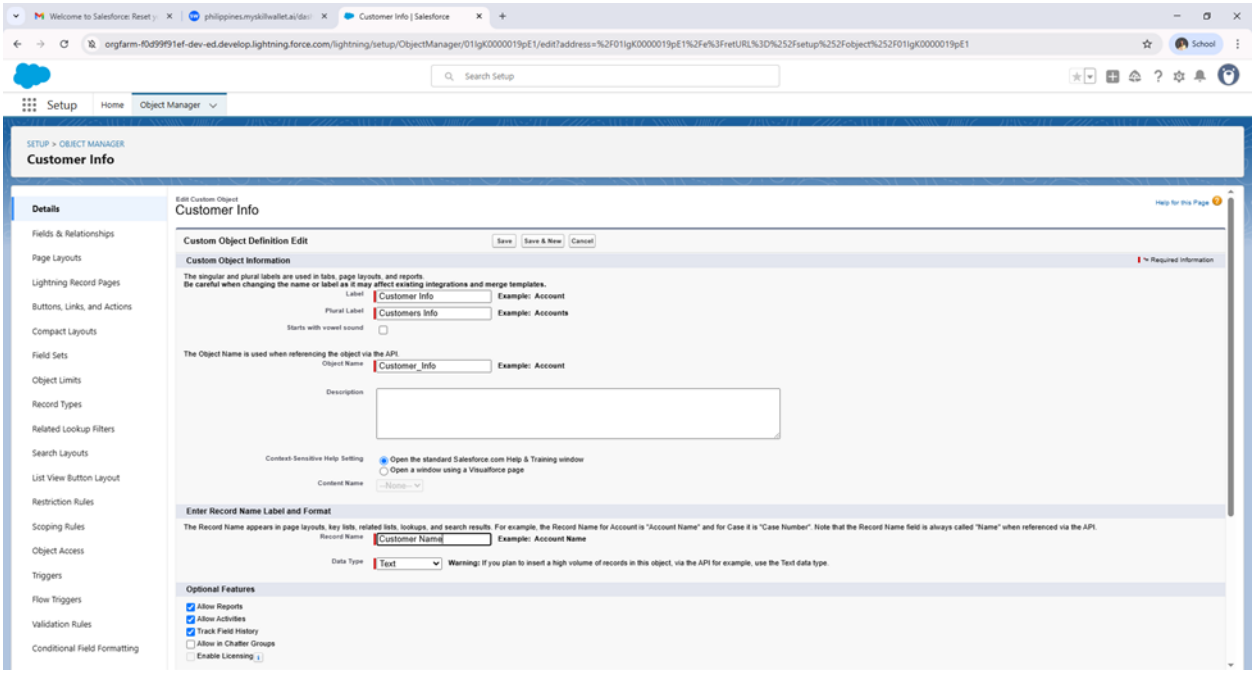


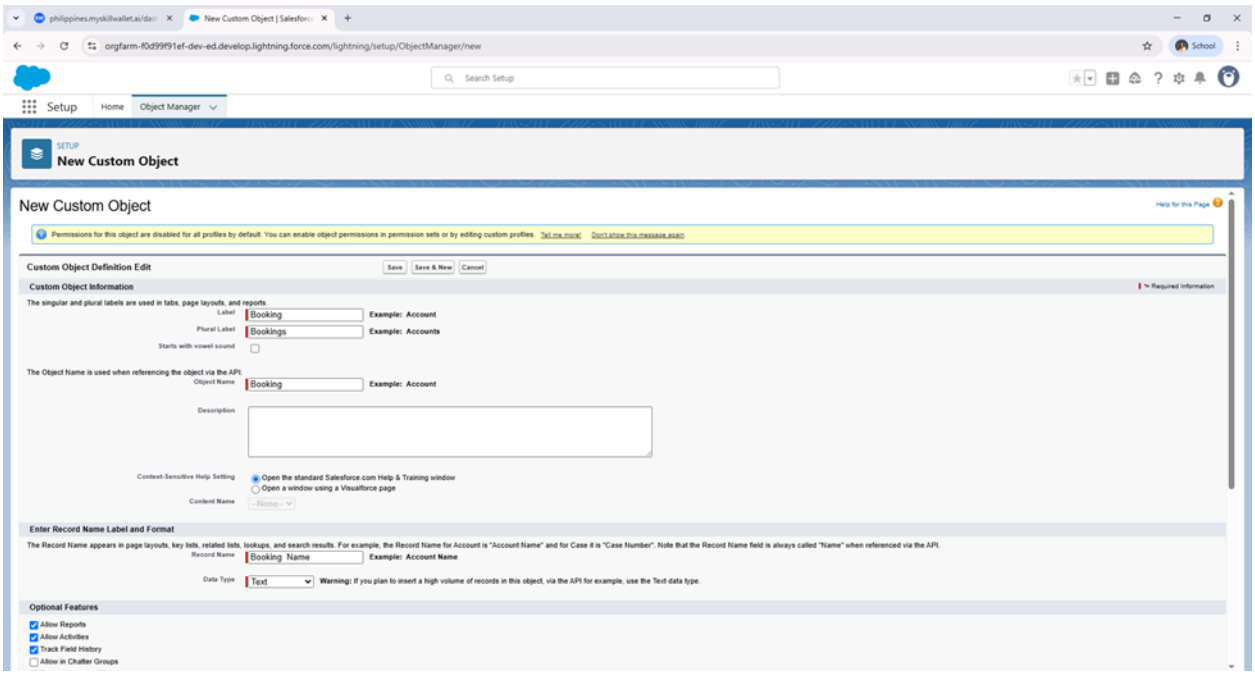


Milestone 2: Custom Objects Creation

I set up the main custom objects required to handle the key features of the Tours and Travels CRM. I began with the Customer Info object, where I defined the object label, record name, and turned on reporting, search capabilities, and field history tracking to

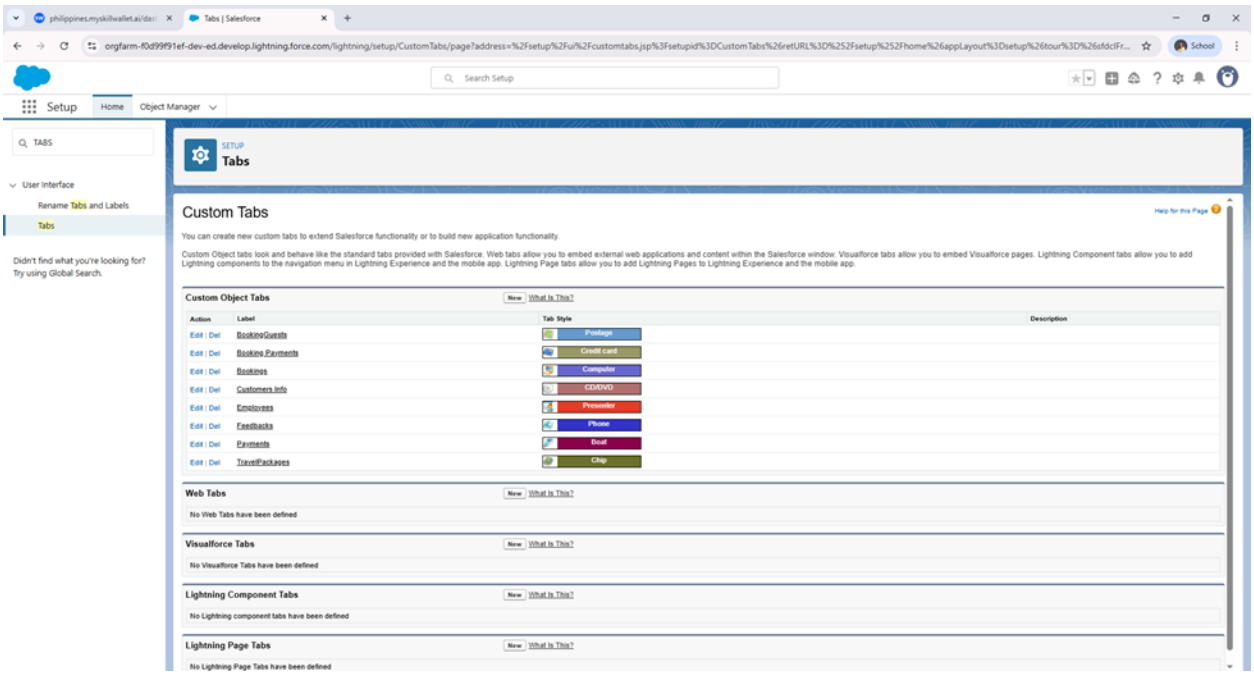
ensure data is easy to monitor and audit. After that, I created other important objects like Booking, Booking Guest, Travel Package, Booking Payment, Employee, and Feedback. Each object was designed to capture and manage specific details essential for travel business operations. These objects were built with appropriate field types and relationships to enable automation, simplify reporting, and support smooth user interactions.





Milestone 3: Setup of Tabs

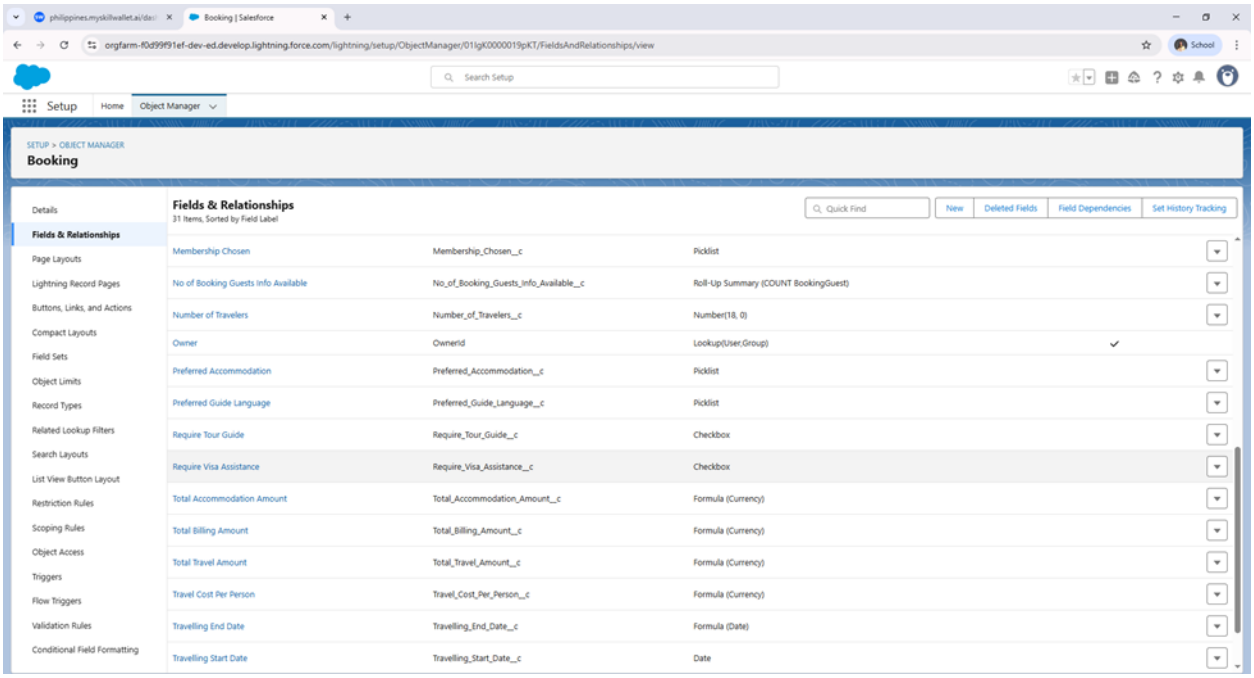
To make the custom objects visible in the user interface, I created custom tabs for each one, including Customer Info, Booking, Booking Guest, Travel Package, Payment, Employee, and Feedback. This setup allowed users to easily access all important records, while administrators could control visibility and permissions through profiles and app settings. Adding these tabs was essential to ensure smooth navigation and a user-friendly experience within the CRM.

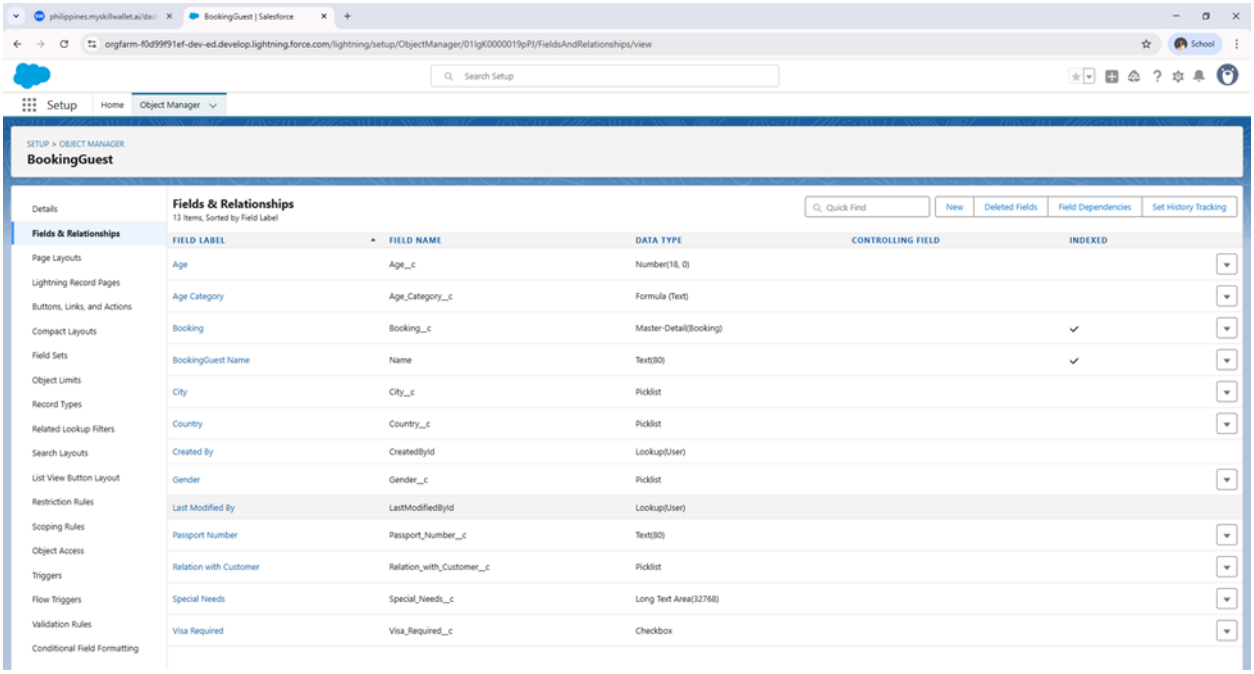


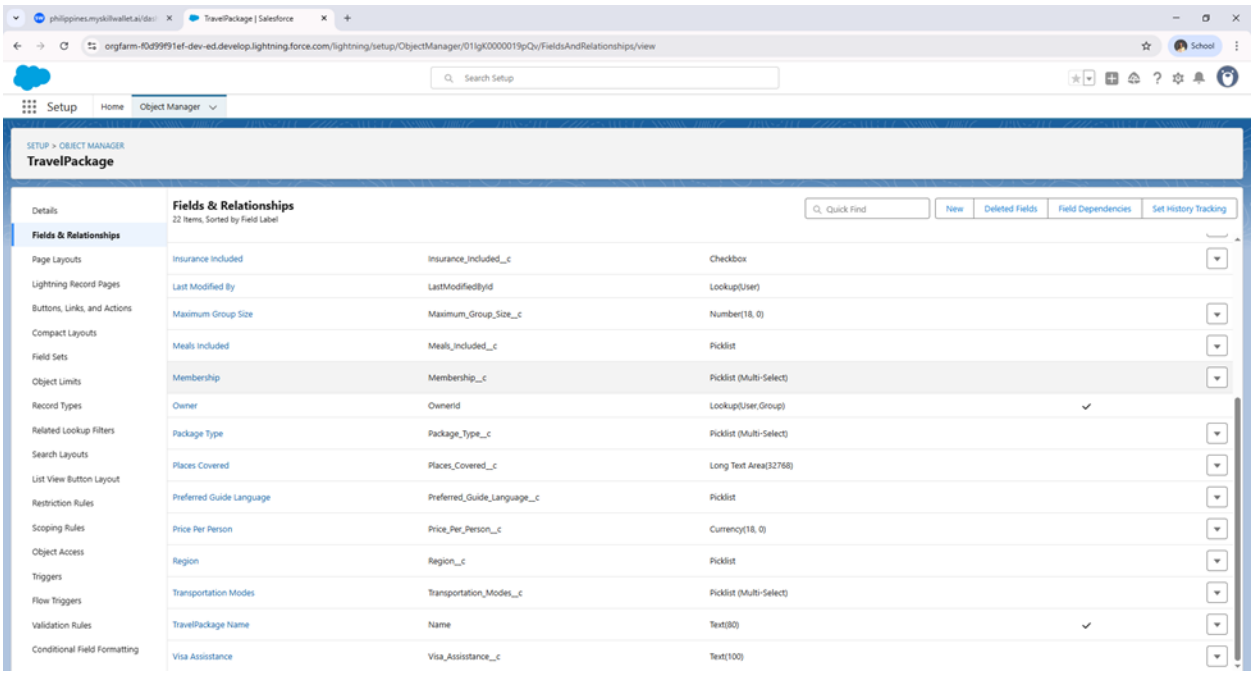
Milestone 4: Fields & Relationships

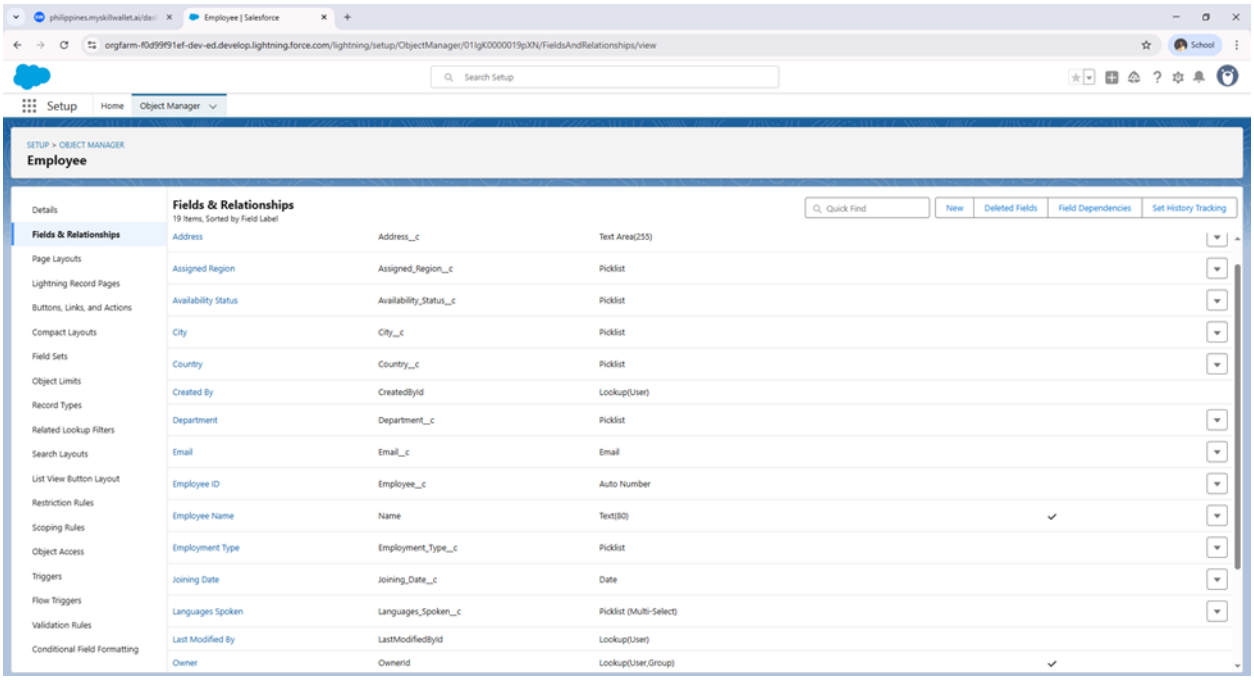
I created all essential fields and set up relationships among the custom objects. Key fields, such as Email for Customer Info, were designed with the right data types and enabled for reporting and search. A Global Picklist Value Set was used for country data to keep it consistent across objects.

Relationships like the Lookup from Booking to Customer Info ensured data stayed properly connected. Important picklists (e.g., Relation with Customer, Availability Status) and multi-select picklists (e.g., Membership, Package Type, Transportation Modes) were added to reflect real travel business needs. Additional fields for employee roles, regions, payment methods, and feedback were also configured, all chosen carefully to support accurate data collection and smooth business processes within the CRM.


Booking | Salesforce
orgfarm-f0d99f91ef-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgK0000019pKT/FieldsAndRelationships/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Booking
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Fields & Relationships
31 Items, Sorted by Field Label
Quick Find
New
Deleted Fields
Field Dependencies
Set History Tracking
Membership Chosen | Membership_Chosen__c | Picklist
No of Booking Guests Info Available | No_of_Booking_Guests_Info_Available__c | Roll-Up Summary (COUNT BookingGuest)
Number of Travelers | Number_of_Travelers__c | Number(18, 0)
Owner | OwnerId | Lookup(User,Group)
Preferred Accommodation | Preferred_Accommodation__c | Picklist
Preferred Guide Language | Preferred_Guide_Language__c | Picklist
Require Tour Guide | Require_Tour_Guide__c | Checkbox
Require Visa Assistance | Require_Visa_Assistance__c | Checkbox
Total Accommodation Amount | Total_Accommodation_Amount__c | Formula (Currency)
Total Billing Amount | Total_Billing_Amount__c | Formula (Currency)
Total Travel Amount | Total_Travel_Amount__c | Formula (Currency)
Travel Cost Per Person | Travel_Cost_Per_Person__c | Formula (Currency)
Travelling End Date | Travelling_End_Date__c | Formula (Date)
Travelling Start Date | Travelling_Start_Date__c | Date


BookingGuest | Salesforce
orgfarm-f0d99f91ef-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgK0000019pPJ/FieldsAndRelationships/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
BookingGuest
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Fields & Relationships
13 Items, Sorted by Field Label
Quick Find
New
Deleted Fields
Field Dependencies
Set History Tracking
FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED
Age | Age__c | Number(18, 0)
Age Category | Age_Category__c | Formula (Text)
Booking | Booking__c | Master-Detail(Booking)
BookingGuest Name | Name | Text(80)
City | City__c | Picklist
Country | Country__c | Picklist
Created By | CreatedById | Lookup(User)
Gender | Gender__c | Picklist
Last Modified By | LastModifiedById | Lookup(User)
Passport Number | Passport_Number__c | Text(80)
Relation with Customer | Relation_with_Customer__c | Picklist
Special Needs | Special_Needs__c | Long Text Area(32768)
Visa Required | Visa_Required__c | Checkbox

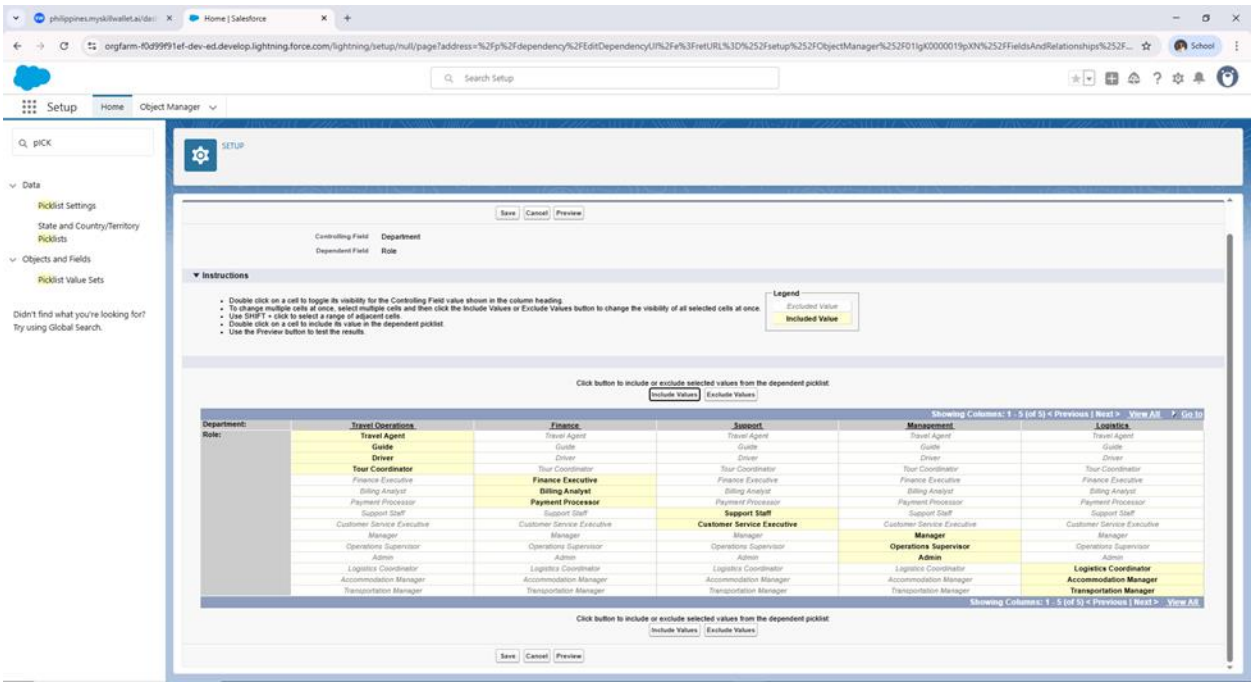


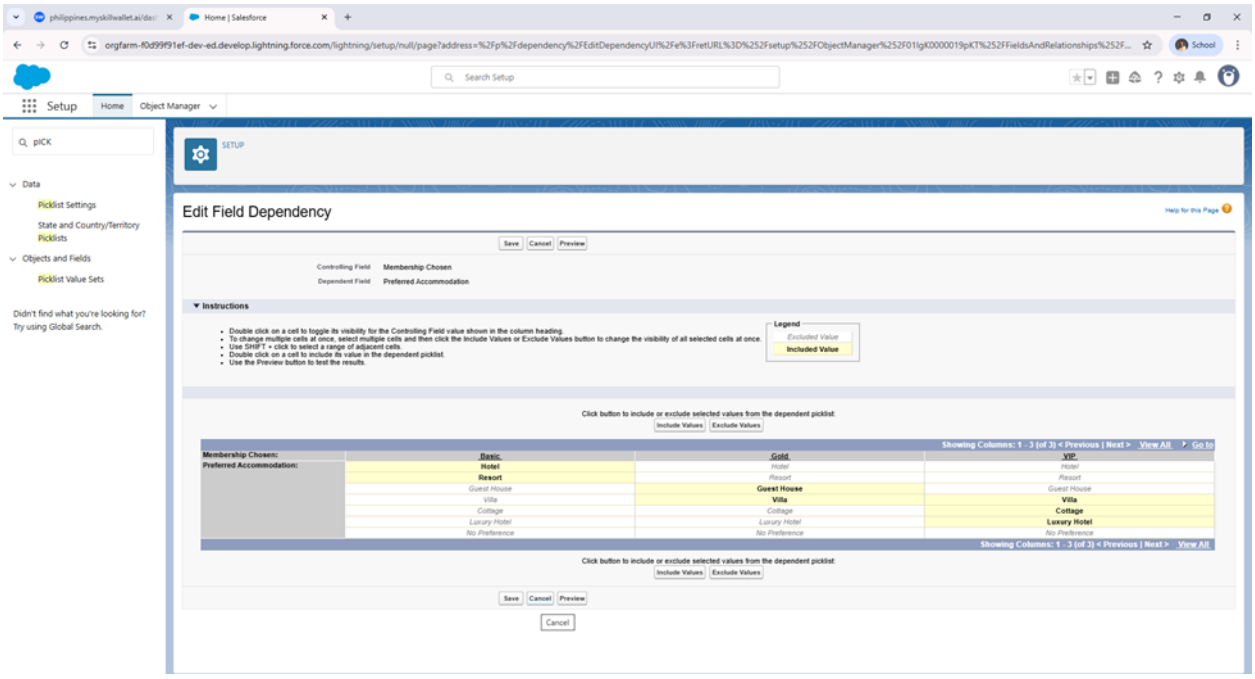


Milestone 5: Field Dependencies Configuration

I improved data accuracy and made forms more intuitive by setting up field dependencies across several objects. For Booking Guest records, city options are shown only after a country is picked, preventing users from selecting unrelated cities.

Within the Booking object, available accommodations are filtered based on the customer's membership tier, so only options that match their level (like Basic, Gold, or VIP) are displayed, supporting tailored booking choices. For Employee records, job roles are shown depending on the department selected, ensuring each role is matched to its correct department. Overall, these dependencies guide users step-by-step, reduce mistakes, and maintain organized, consistent information throughout the CRM.





Milestone 6: Validation Rules Implementation

To ensure the reliability and consistency of data entered into the Tours and Travels CRM, a series of validation rules were applied across key custom objects. These rules act as safeguards against incorrect or incomplete data and help enforce business requirements at the point of entry.

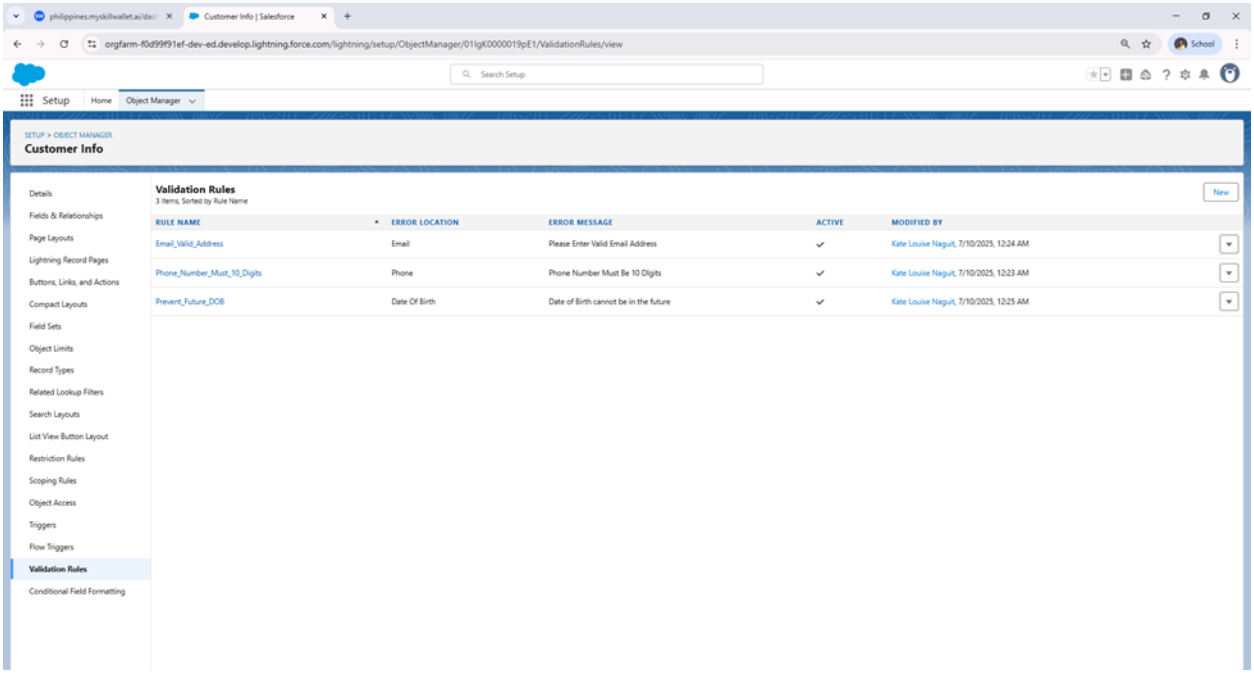
In the Customer Info object, validations were put in place to make sure phone numbers contain exactly 10 digits, email addresses follow a valid format, and birth dates cannot be set in the future. This ensures only realistic and standardized customer records are saved.

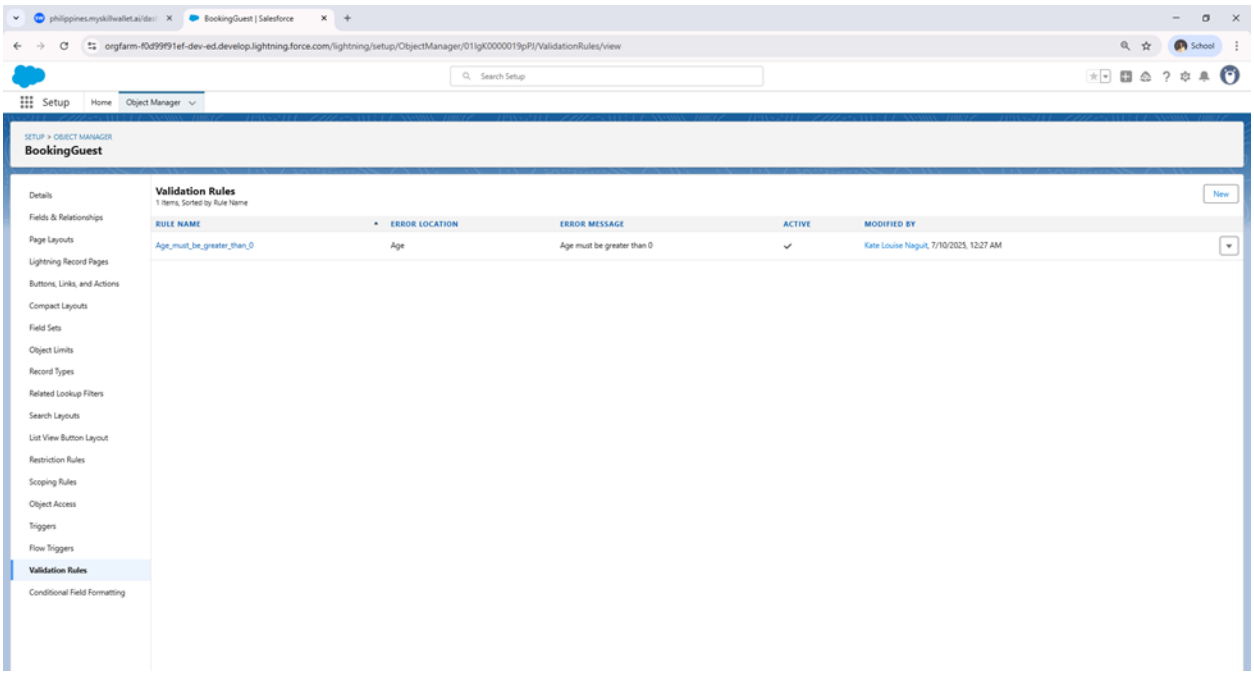
For the Booking Guest object, the system enforces that the age entered must be greater than zero. It also includes conditional logic where the email field becomes mandatory based on the role assigned to the guest, supporting role-based completeness in data entry.

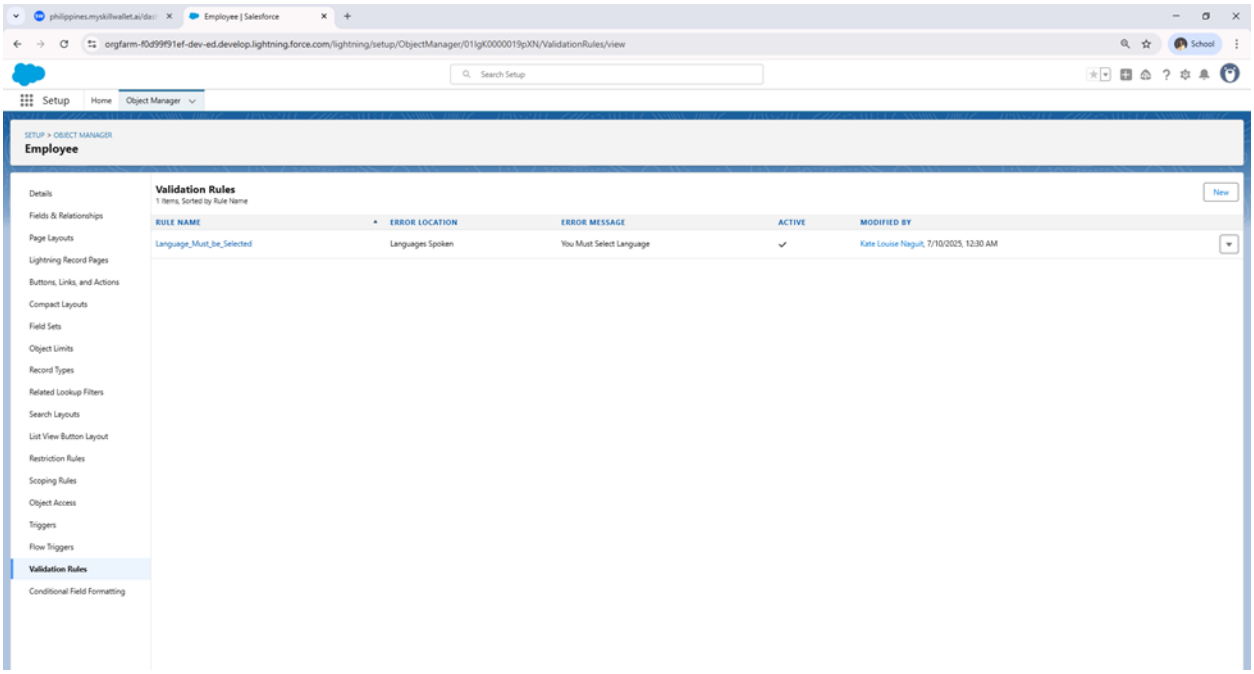
Within the Employee object, validation was added to make sure tour guides are not saved without selecting at least one language spoken, aligning with the operational requirement that guides must be multilingual to serve clients effectively.

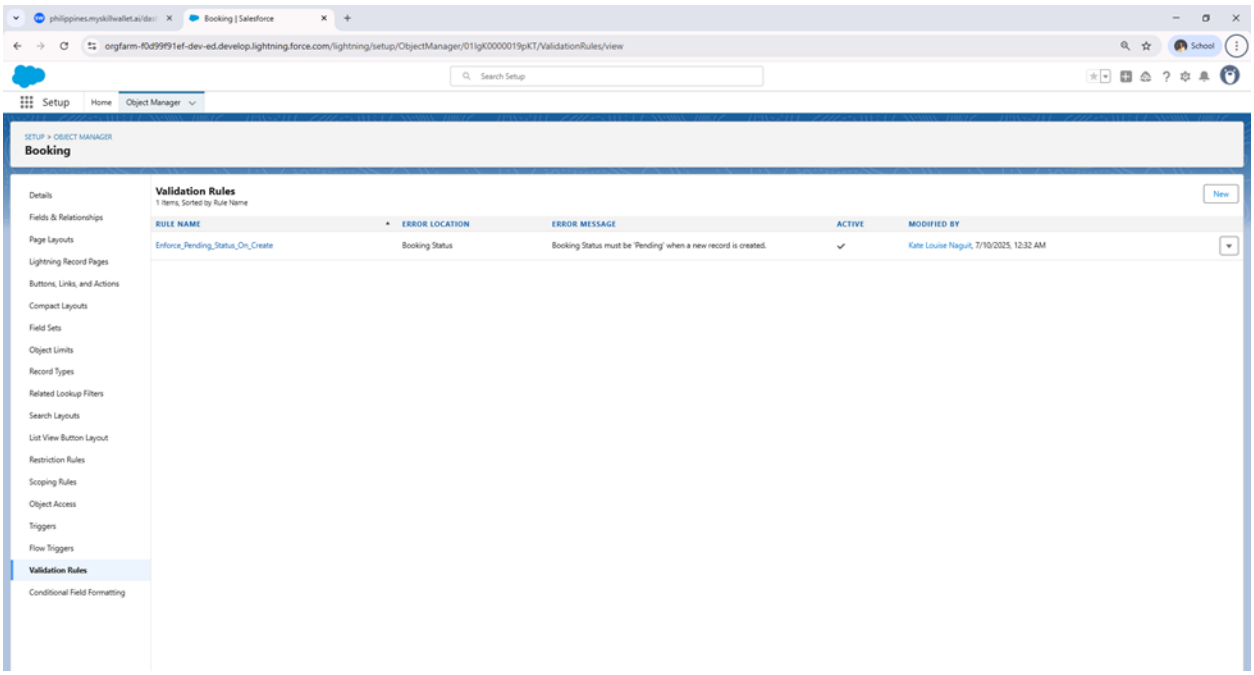
Moreover, the Booking object automatically assigns a default value of "Pending" to the status field when a new booking is created. This promotes standardized booking lifecycles and prevents untracked entries.

In addition, these rules not only prevent errors but also reduce the need for manual corrections later on, saving time and improving the overall efficiency of system use.




BookingGuest | Salesforce
orgfarm-f0d99f91ef-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgK0000019pPJ/ValidationRules/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
BookingGuest
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Validation Rules
1 Items, Sorted by Rule Name
New
RULE NAME
ERROR LOCATION
ERROR MESSAGE
ACTIVE
MODIFIED BY
Age_must_be_greater_than_0
Age
Age must be greater than 0
Kate Louise Naguit, 7/10/2025, 12:27 AM


Employee | Salesforce
orgfarm-f0d99f91ef-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgK0000019pXN/ValidationRules/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Employee
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Validation Rules
1 Items, Sorted by Rule Name
New
RULE NAME
ERROR LOCATION
ERROR MESSAGE
ACTIVE
MODIFIED BY
Language_Must_be_Selected
Languages Spoken
You Must Select Language
Kate Louise Naguit, 7/10/2025, 12:30 AM



Milestone 7: Approval Process

The approval process setup began with the creation of necessary components such as user roles, profiles, and individual user accounts to establish the right hierarchy and permissions. After preparing these foundational elements, I designed a classic email template to notify managers whenever a booking cancellation request is submitted. This notification includes key information about the booking and directs the recipient to log in and review the request.

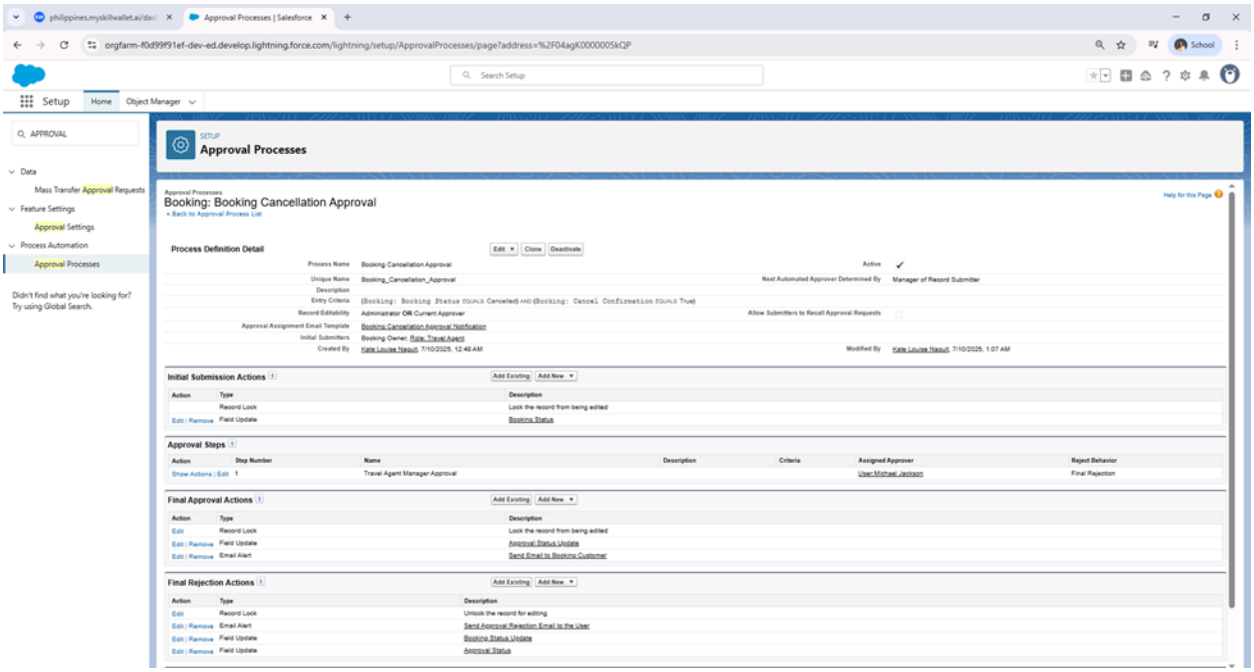
To support transparent communication, I also developed two more email templates-one for approval notifications and another for rejection messages. These are sent to customers with personalized booking details, ensuring they are well-informed about the status and outcome of their cancellation requests.

After setting up the communication tools, I configured the actual approval process for the Booking object using Salesforce's built-in wizard. The process was triggered only

when two specific conditions were met. This ensured that only verified cancellation requests entered the approval workflow.

The approval request was assigned to a designated user, and the first email template was linked to notify them accordingly. I customized the approval page layout to show relevant fields such as the booking number, travel package, and cancellation reason for clear and efficient decision-making.

Submission privileges were given to travel agents and booking record owners. Lastly, I defined a single approval step assigned to the Travel Agent Manager and activated the workflow. This ensures that every cancellation is properly reviewed, approved or rejected with traceability, and that customers receive timely updates on their requests.

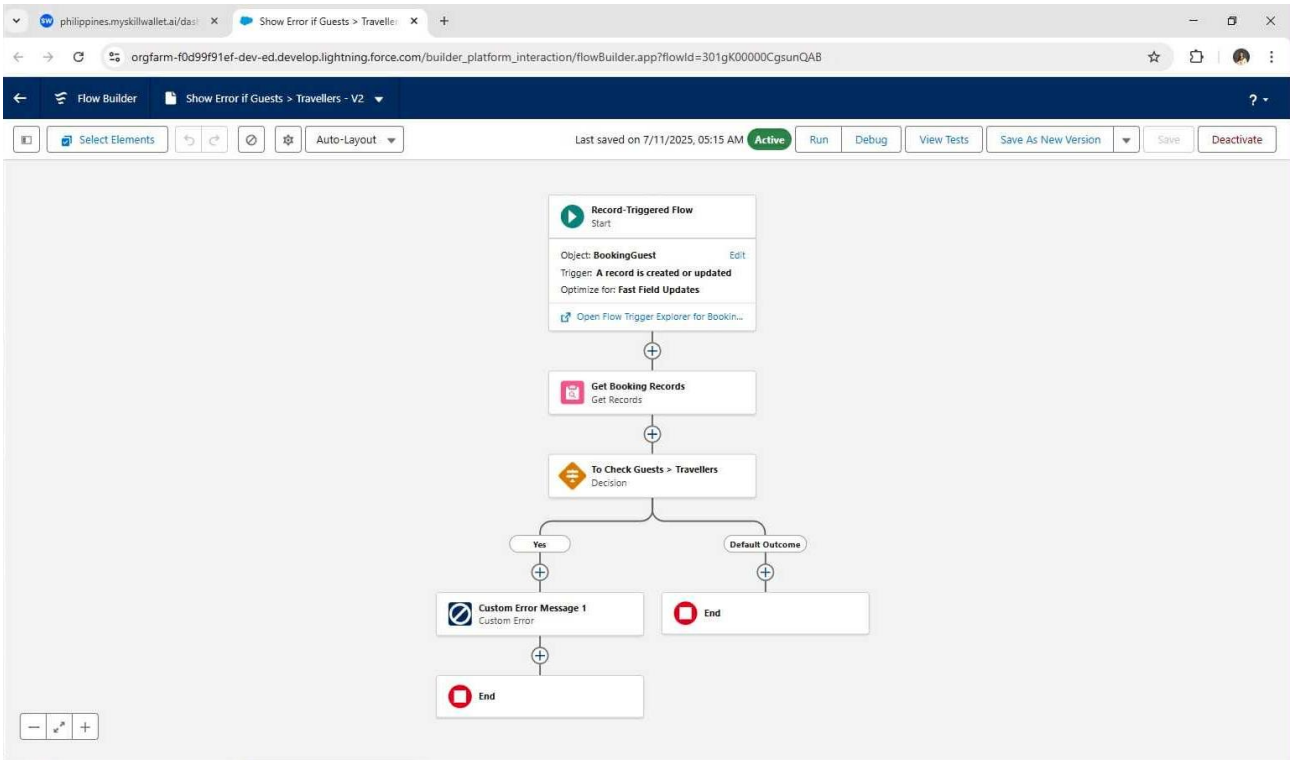


Milestone 8: Flow Automation Development

To uphold booking accuracy, I designed a record-triggered flow that restricts users from adding more guests than the total number of travelers specified in a booking. This automation was set up on the Booking Guest object, with the trigger configured to activate when a record is either created or updated.

The flow began by using a Get Records element to pull the associated Booking record and all its fields. A Decision element followed, which checked whether the count of existing Booking Guest entries exceeded the number of travelers defined in the booking. If this condition was true, the flow proceeded down the "Yes" path. To inform users of the issue, I created a Text Template resource called ErrorMessageTemplate. I then added a Custom Error element under the "Yes" branch of the flow to display the message in a pop-up window directly on the record page. Once the flow was saved and activated, it began working as intended-blocking any attempt to add excess guests beyond the allowed limit.

This automation reinforces business logic and helps ensure data accuracy within the Tours and Travels CRM system by proactively preventing overbooking at the point of entry.

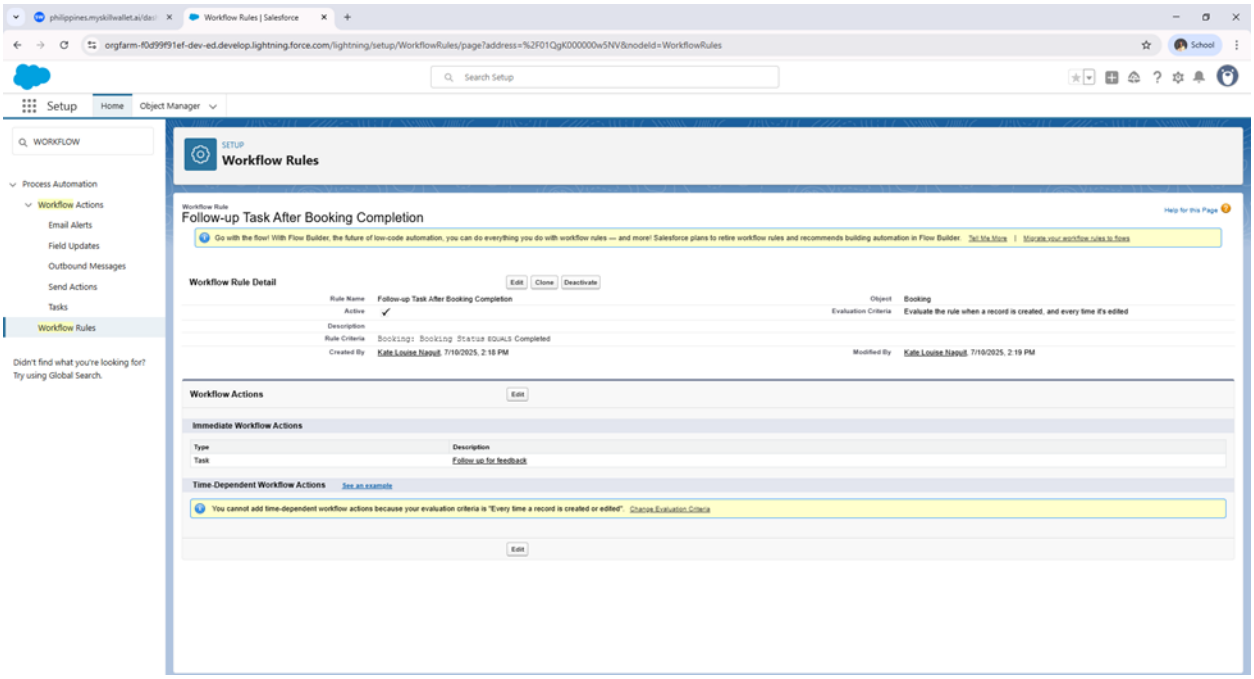


Milestone 9: Workflow Rules Setup

To improve post-trip engagement and ensure timely customer feedback, I implemented an automated workflow that assigns a follow-up task to travel agents when a booking is marked as Completed. Using the Workflow Rules feature in Setup, I created a new rule under the Booking object. This rule, titled "Follow-up Task After Booking Completion," was set to trigger on both the creation and update of booking records. The workflow criteria were clearly defined: it would only activate if the Booking Status equals Completed, indicating the trip had concluded.

For the workflow action, I added a new task that automatically gets assigned to the Booking Owner. The task included a subject line, "Follow up for feedback," and was scheduled with a due date three days after the Traveling End Date. Its priority was set to Normal, with the status initialized as Not Started. I also included a description instructing the agent to contact the customer and request feedback about their travel experience.

To ensure proper task visibility, I confirmed that the task appears in the agent's activity timeline. After completing the configuration, I saved and activated the workflow. This automation helps maintain consistent communication with customers, encourages feedback collection, and supports continuous improvement in travel service delivery.



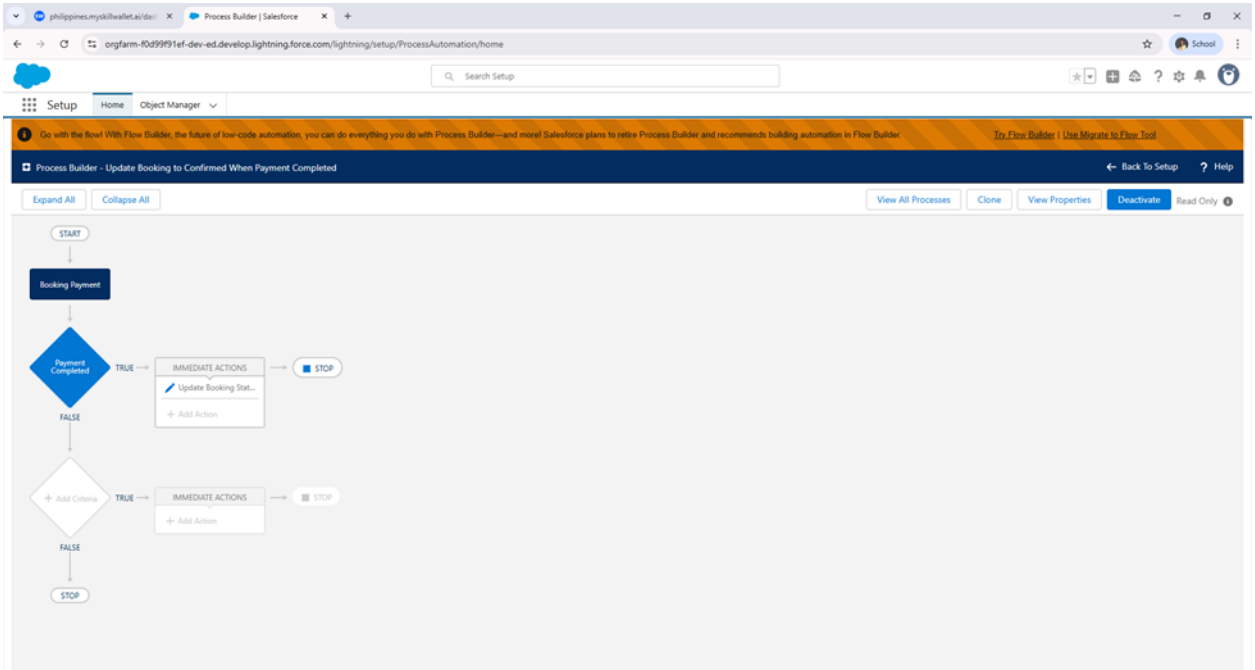
Milestone 10: Process Builder Implementation

To automate status updates and reduce manual workload in the Tours and Travels CRM, I developed a Process Builder flow that changes the Booking Status to "Confirmed" once a related Payment record reaches the status "Completed." This ensures smooth and timely updates across the system.

I launched Process Builder from Setup and created a new process named "Update Booking to Confirmed When Payment Completed." The process is set to trigger whenever a Booking Payment record is created or modified.

For the criteria, I added a condition specifying that the Payment_Status__c field must equal "Completed." This ensures the automation only runs when a payment is fully processed. As the next step, I defined an immediate action to update the linked Booking record. The action modifies the Booking Status field, setting its value to "Confirmed" automatically.

After configuring the necessary logic and actions, I saved and activated the process. This automation now ensures that once a payment is completed, the booking status is instantly updated, supporting real-time tracking and operational accuracy without manual intervention.

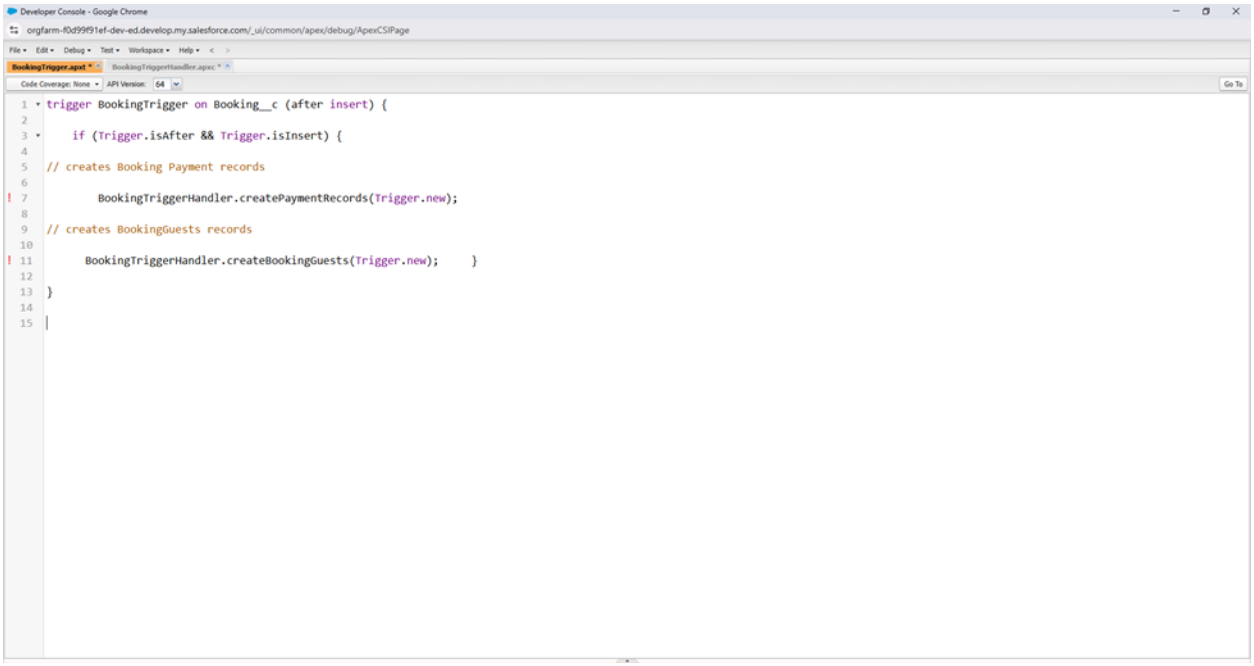


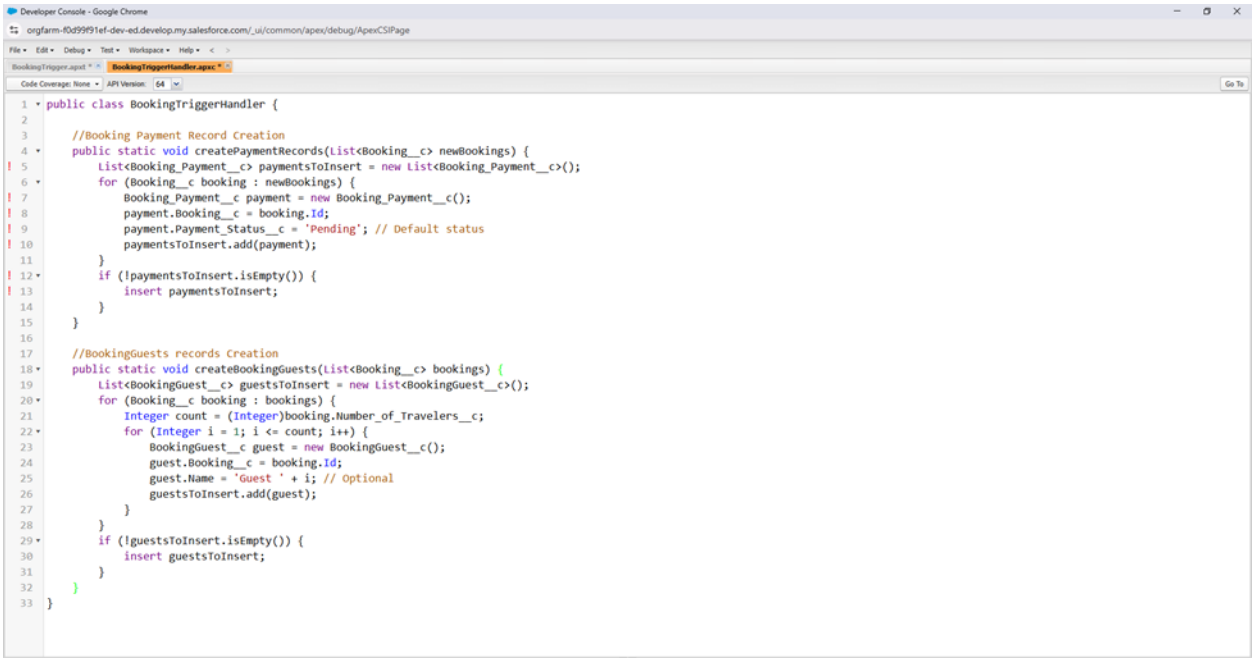
Milestone 11: Apex Triggers Development

I developed an Apex Trigger to automate two key actions every time a new Booking record is added to the system. After logging into Salesforce with administrative privileges, I launched the Developer Console and created a trigger on the Booking__c object, named BookingTrigger. This trigger is configured to run after insert and delegates its logic to a handler class named BookingTriggerHandler.

Within the handler class, I implemented a method that automatically generates a corresponding Booking Payment record for each newly added booking. Each payment record is initialized with a default status of "Pending" to reflect that the transaction hasn't been completed yet. I then expanded the trigger's functionality by adding another method which dynamically creates Booking Guest records based on the value of the Number_of_Travelers__c field. The system generates a matching number of guest records for each booking, assigning labels and links them to the relevant booking record.

Once both the trigger and the handler class were finalized and saved, the automation now guarantees that every new booking entry will be paired with a pending payment and the appropriate number of guest records-streamlining the data entry process and reinforcing system consistency.





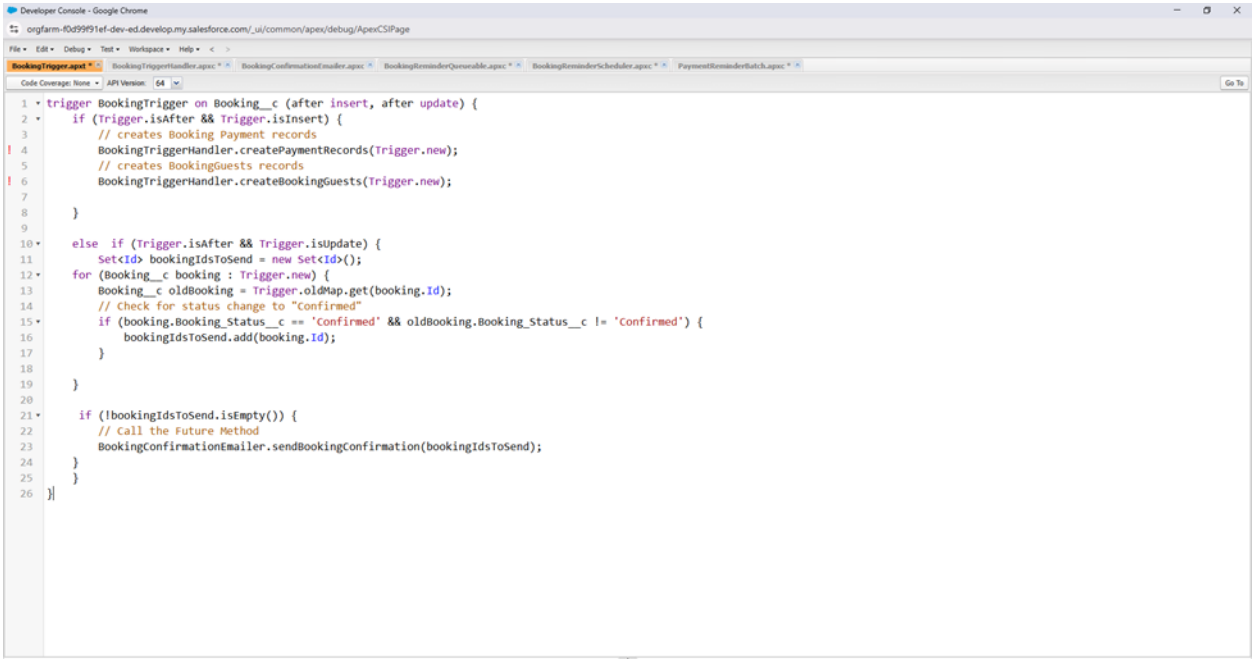
Milestone 12: Asynchronous Processing Setup

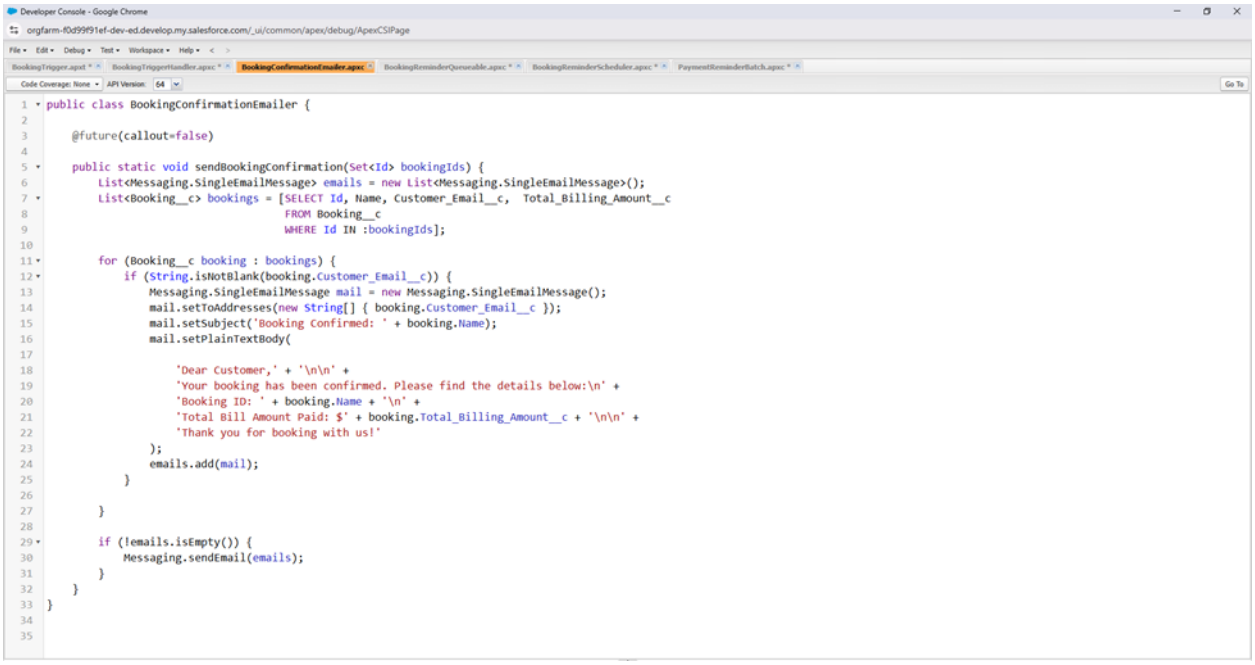
To improve system efficiency and ensure timely communication, I automated important email processes using asynchronous Apex. I created a @future method that sends

a confirmation email once a booking is marked as “Confirmed.” This was triggered through an update in the existing booking trigger.

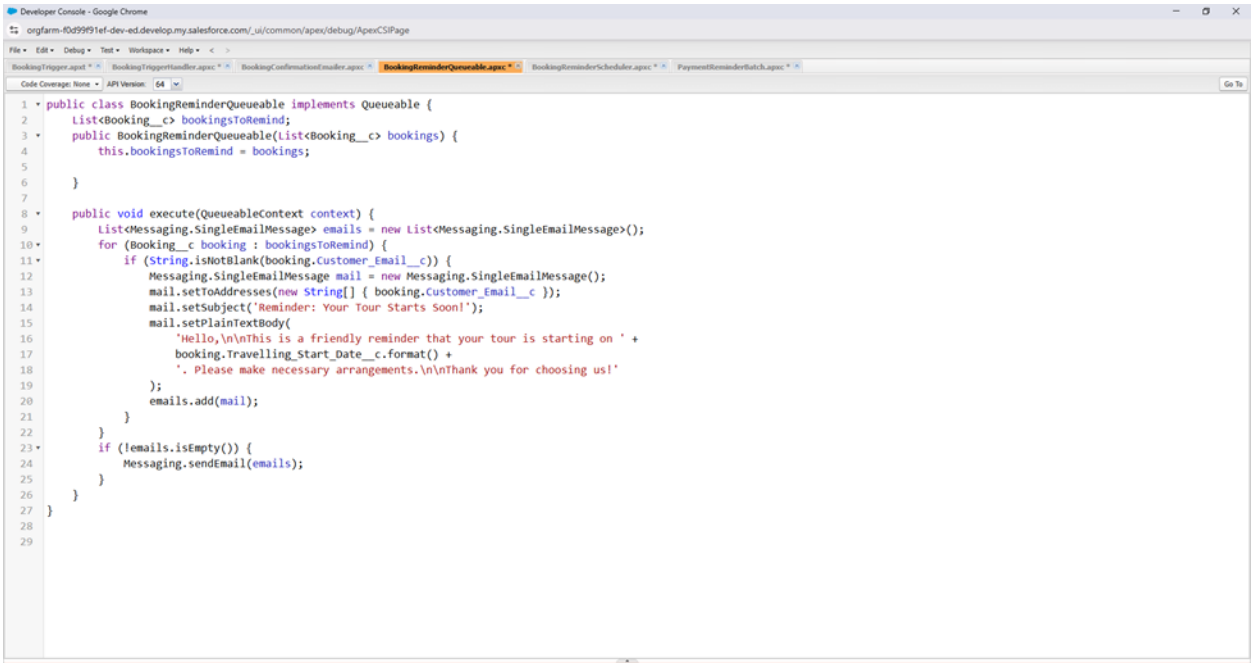
I developed a Queueable Apex class for sending tour reminders three days before the travel date. I scheduled this to run daily at 6 AM using a Schedulable class and CRON expression. I built a Batch Apex class to send payment reminders for bookings still marked as “Pending” from the day before. This was also scheduled to run every day at 5 AM and included a notification to the admin once completed.

These asynchronous processes ensure that important emails-confirmation, tour reminders, and payment follow-ups are sent automatically without slowing down system performance.

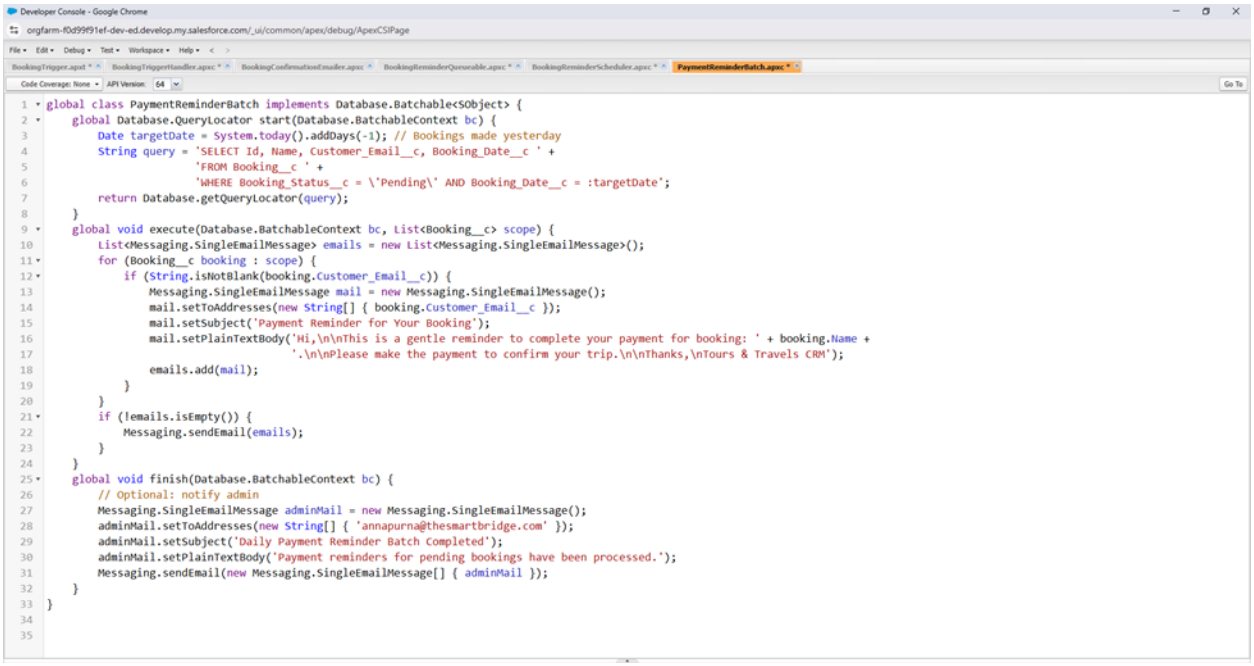





```apex
public class BookingConfirmationEmailer {

    @future(callout=false)

    public static void sendBookingConfirmation(Set<Id> bookingIds) {
        List<Messaging.SingleEmailMessage> emails = new List<Messaging.SingleEmailMessage>();
        List<Booking__c> bookings = [SELECT Id, Name, Customer_Email__c,  Total_Billing_Amount__c
                                     FROM Booking__c
                                     WHERE Id IN :bookingIds];

        for (Booking__c booking : bookings) {
            if (String.isNotBlank(booking.Customer_Email__c)) {
                Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
                mail.setToAddresses(new String[] { booking.Customer_Email__c });
                mail.setSubject('Booking Confirmed: ' + booking.Name);
                mail.setPlainTextBody(

                    'Dear Customer,' + '\n\n' +
                    'Your booking has been confirmed. Please find the details below:\n' +
                    'Booking ID: ' + booking.Name + '\n' +
                    'Total Bill Amount Paid: $' + booking.Total_Billing_Amount__c + '\n\n' +
                    'Thank you for booking with us!'
                );
                emails.add(mail);
            }

        }

        if (!emails.isEmpty()) {
            Messaging.sendEmail(emails);
        }
    }
}
```




```apex
public class BookingReminderQueueable implements Queueable {
    List<Booking__c> bookingsToRemind;
    public BookingReminderQueueable(List<Booking__c> bookings) {
        this.bookingsToRemind = bookings;

    }

    public void execute(QueueableContext context) {
        List<Messaging.SingleEmailMessage> emails = new List<Messaging.SingleEmailMessage>();
        for (Booking__c booking : bookingsToRemind) {
            if (String.isNotBlank(booking.Customer_Email__c)) {
                Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
                mail.setToAddresses(new String[] { booking.Customer_Email__c });
                mail.setSubject('Reminder: Your Tour Starts Soon!');
                mail.setPlainTextBody(
                    'Hello,\n\nThis is a friendly reminder that your tour is starting on ' +
                    booking.Travelling_Start_Date__c.format() +
                    '. Please make necessary arrangements.\n\nThank you for choosing us!'
                );
                emails.add(mail);
            }
        }
        if (!emails.isEmpty()) {
            Messaging.sendEmail(emails);
        }
    }
}
```



Phase 3: UI/UX Development & Customization

In this phase, the focus was on enhancing the overall user interface and user experience of the Tours & Travels CRM by configuring navigation, optimizing form layouts, managing users, building visual reports and dashboards, and developing custom components to support interactive functionality.

A custom Lightning App named "Tours & Travels CRM" was created using the App Manager to centralize all essential CRM functionalities in one accessible location. Branding elements were applied, and key objects such as Customer Info, Travel Packages, Bookings, Payments, Guests, Employees, Feedback, Tasks, Reports, and Dashboards were added to the navigation bar. The app was activated with the System Administrator profile to ensure proper access and availability.

To streamline data entry and improve usability:

Page Layouts for major standard and custom objects (e.g., Customer Info, Booking, Booking Guest, Travel Package, Employee, Booking Payments, and Feedback) were customized using the Object Manager. Fields were strategically organized, with the most critical fields placed at the top for improved visibility and efficiency.

Dynamic Forms were implemented on the Booking object to enhance responsiveness. Conditional visibility rules were applied—for example, fields like Cancellation Date and Approval Status only appear when the Booking Status is set to "Cancelled". This reduced interface clutter and ensured users only saw relevant information based on context.

User accounts were created and assigned with predefined Roles and Profiles to ensure appropriate access levels across the CRM system. Each user was given a relevant role (e.g., Travel Agent Manager, Travel Agent) and associated with either the Salesforce Platform license or the appropriate custom profile. Multiple users were added under each role to support testing and role-based functionality validation.

To support data analysis and business insights:

- Several Reports were developed, including a Monthly Revenue Report grouped by Travel Package and Revenue Category. Key fields such as Booking Name, Booking Date, and Total Billing Amount were included, and visual charts (e.g., bar charts) were auto-generated to illustrate trends.
- A comprehensive Dashboard named "Tours & Travels Dashboard" was created, consisting of multiple components such as:
- A vertical bar chart for Employee Role Breakdown
- A gauge chart for Top Travel Packages
- A table view for Monthly Revenue Report

- A placeholder for Booking Payments to show payment activity (not yet populated)

This dashboard provided real-time, visual insights into CRM performance and operations.

As part of a bonus enhancement, a custom Lightning Web Component (LWC) was developed and integrated:

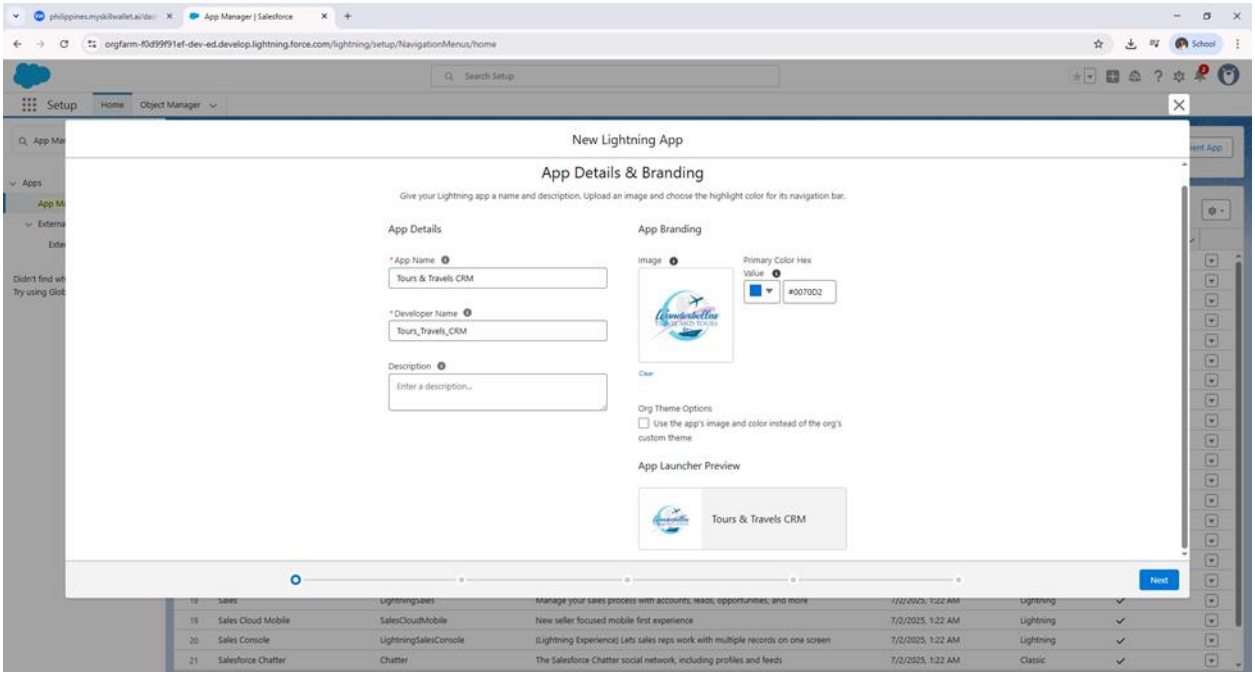
- An Apex class, TravelPackageController, was created to retrieve Travel Package records filtered by country using a SOQL query.
- The method was annotated with @AuraEnabled(cacheable=true) for efficient, secure data fetching.
- This backend logic was connected to the LWC using @wire or import in JavaScript.
- The component allowed users to dynamically select a country and view available travel packages.

To present the custom LWC within a more interactive interface, a dedicated Lightning App Page titled "Travel Package Selector" was built using the Lightning App Builder. A single-region layout was used for simplicity, and the custom component was placed at the center. The page was activated and added to the Tours & Travels CRM app, making it easily accessible from the App Launcher. This enhanced the system's interactivity and improved the user experience.

Overall, this phase established a modern, responsive, and user-friendly CRM interface by combining standard Salesforce UI customization tools with custom LWC development, ensuring that users can navigate and interact with the system effectively and efficiently.

Milestone 13: Lightning App Setup

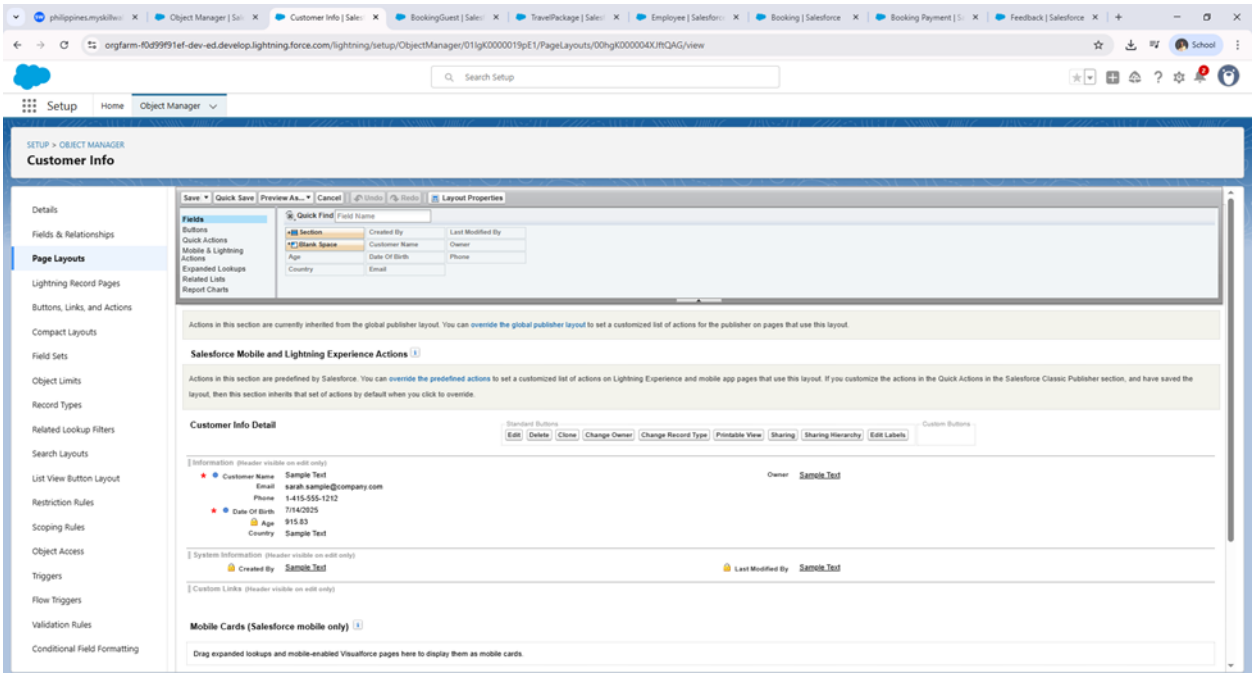
I developed a custom Lightning App called "Tours & Travels CRM" to simplify user navigation and bring all important modules together. Using the App Manager, I customized the branding with an image, kept default settings, and added core objects such as Customer Info, Travel Packages, Bookings, Payments, Guests, Employees, Feedback, Tasks, Reports, and Dashboards. To ensure the right access, I assigned the System Administrator profile and finalized the setup by saving and activating the app.

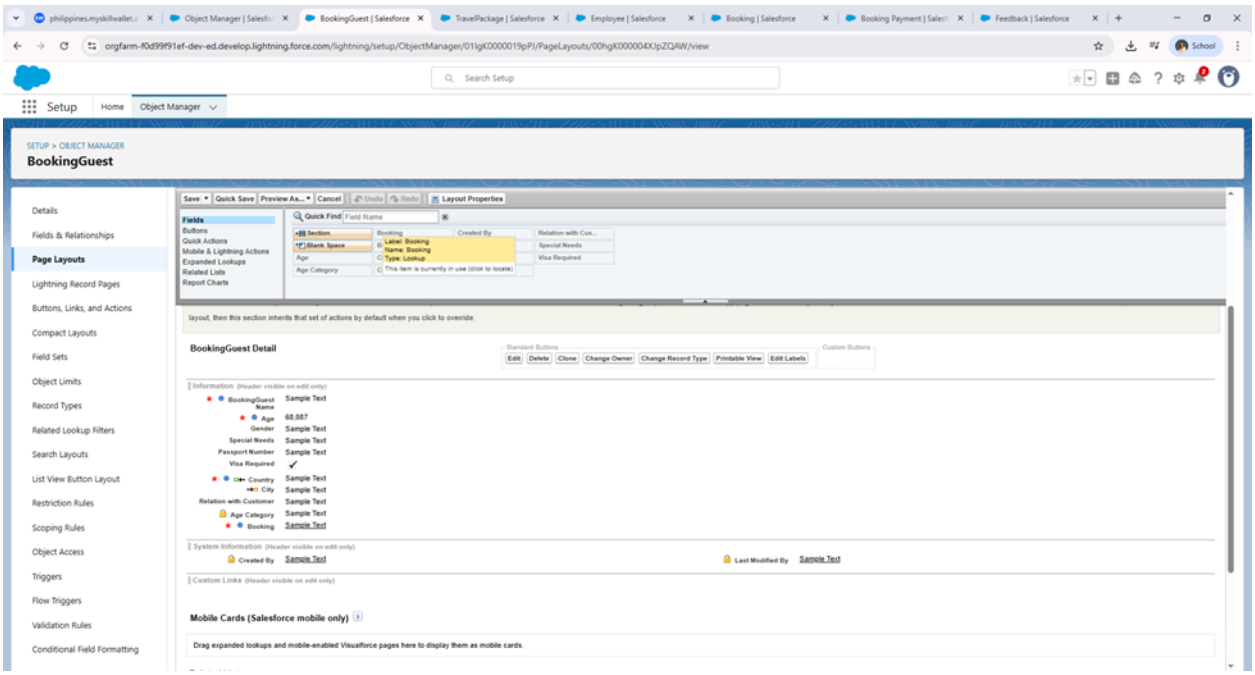


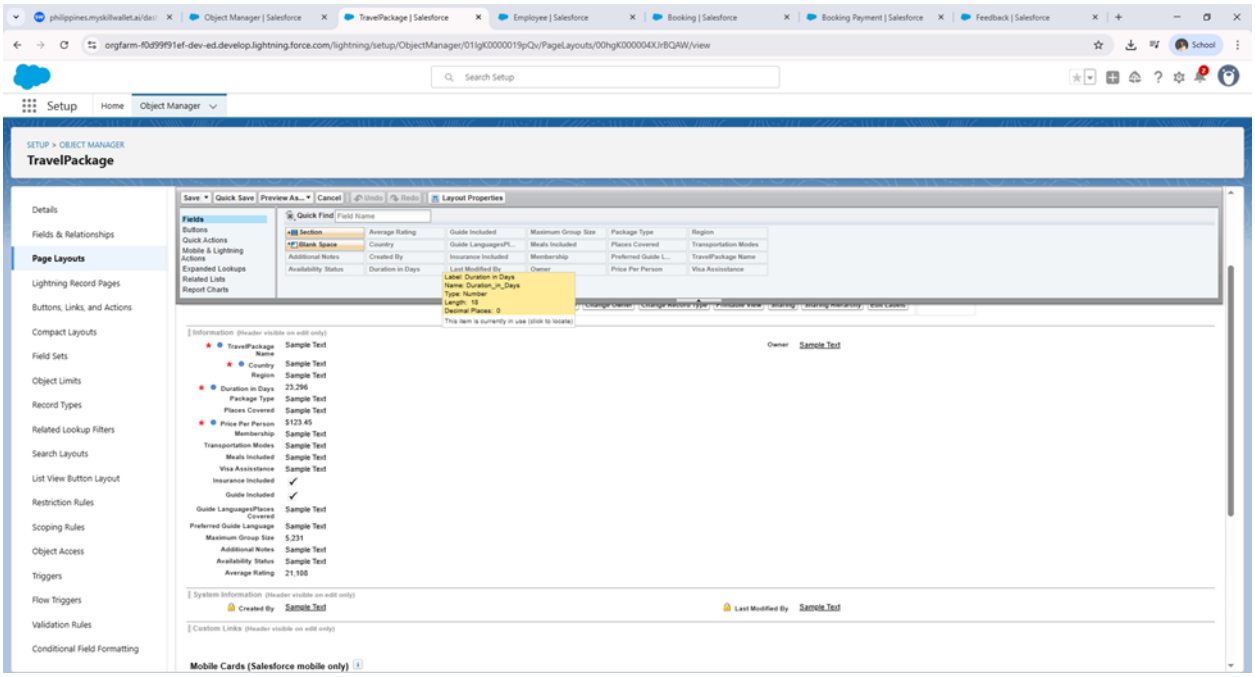
Milestone 14: Editing of Page Layouts

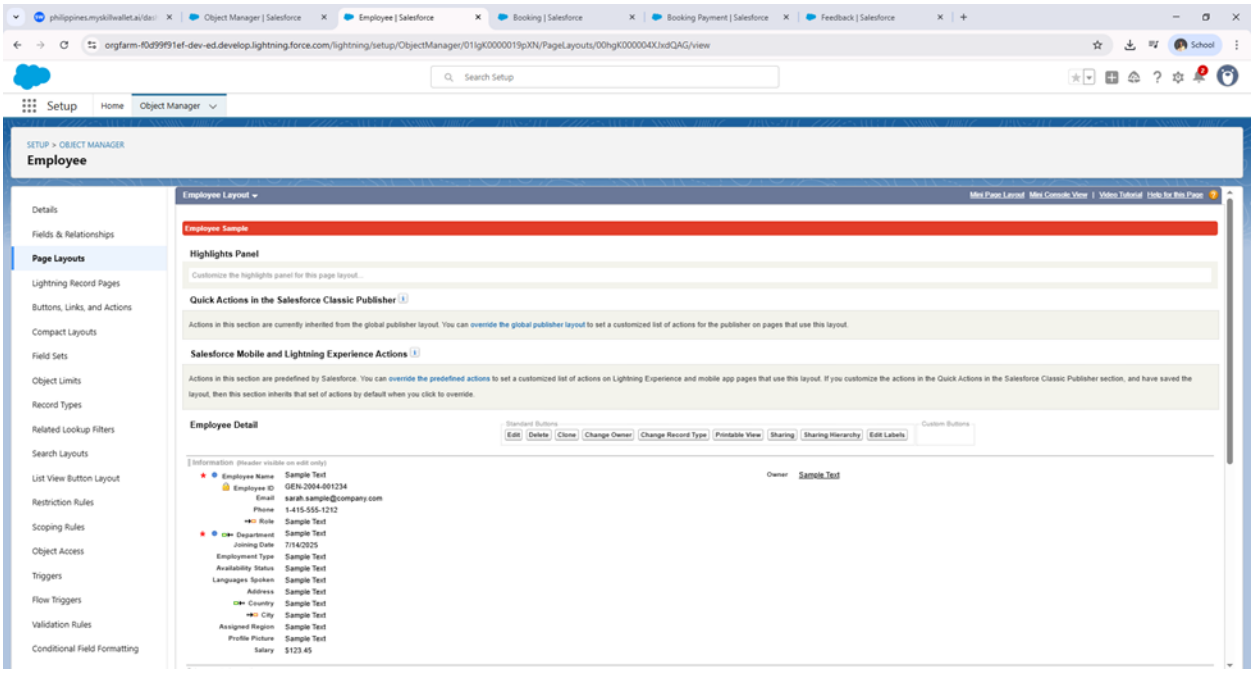
I improved the layout design of several standard and custom objects within the Tours and Travels CRM to create a more user-friendly and efficient interface. Through the Object Manager in Salesforce Setup, I customized the page layouts for key objects like Customer Info, Booking Guest, Travel Package, Employee, Booking, Booking Payments, and Feedback.

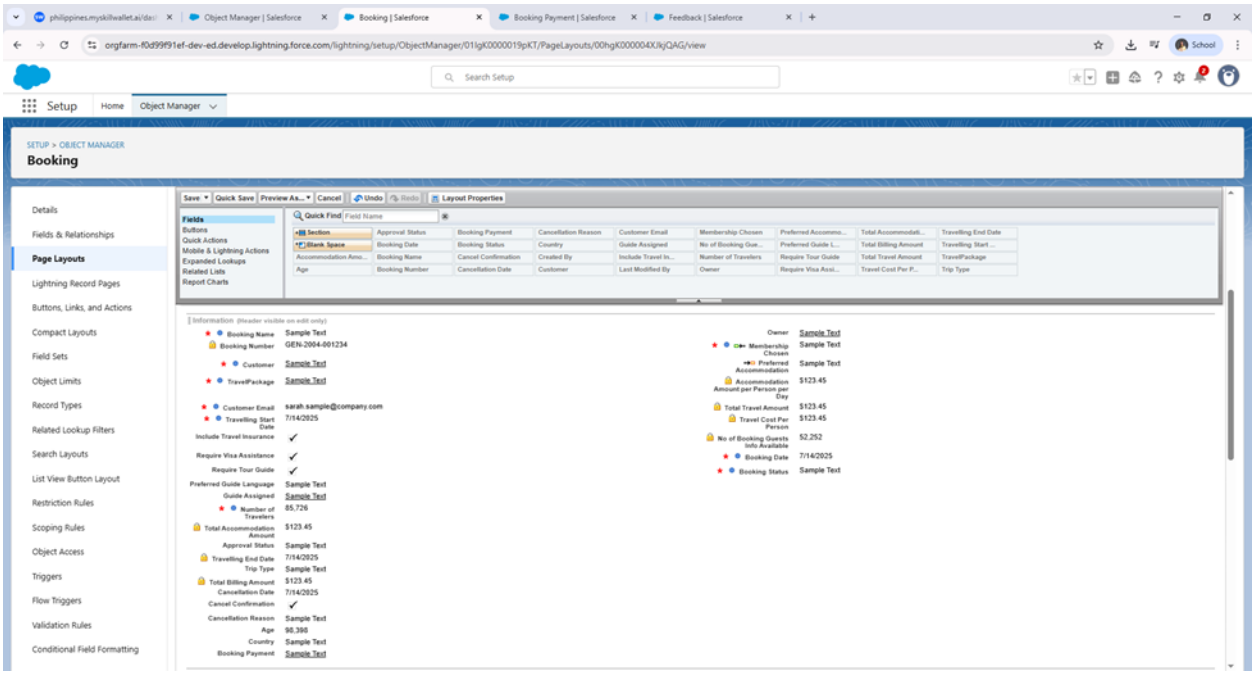
For each one, I strategically arranged fields, placing the most important ones at the top for easier access and organizing the rest for better clarity. These changes were made to enhance readability, minimize data entry errors, and make navigation smoother. After finalizing the layout updates, I saved the configurations, making the enhancements live for users. Overall, these improvements lead to a cleaner interface and a more streamlined workflow for CRM users.

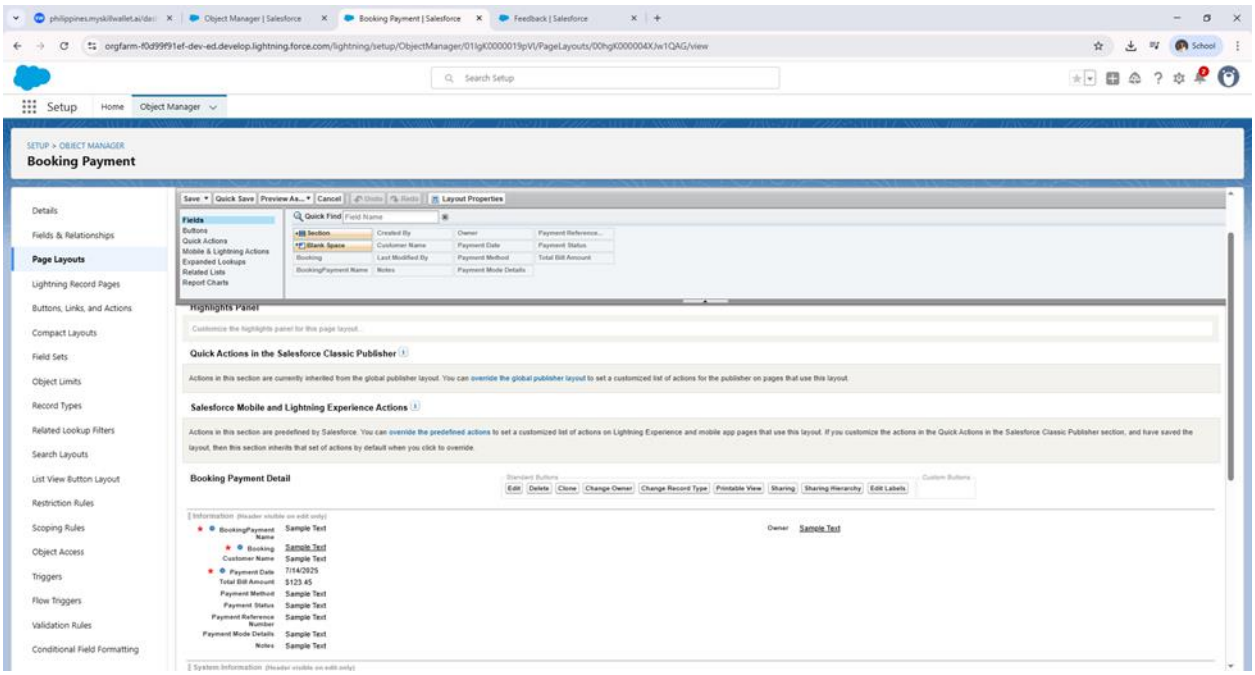


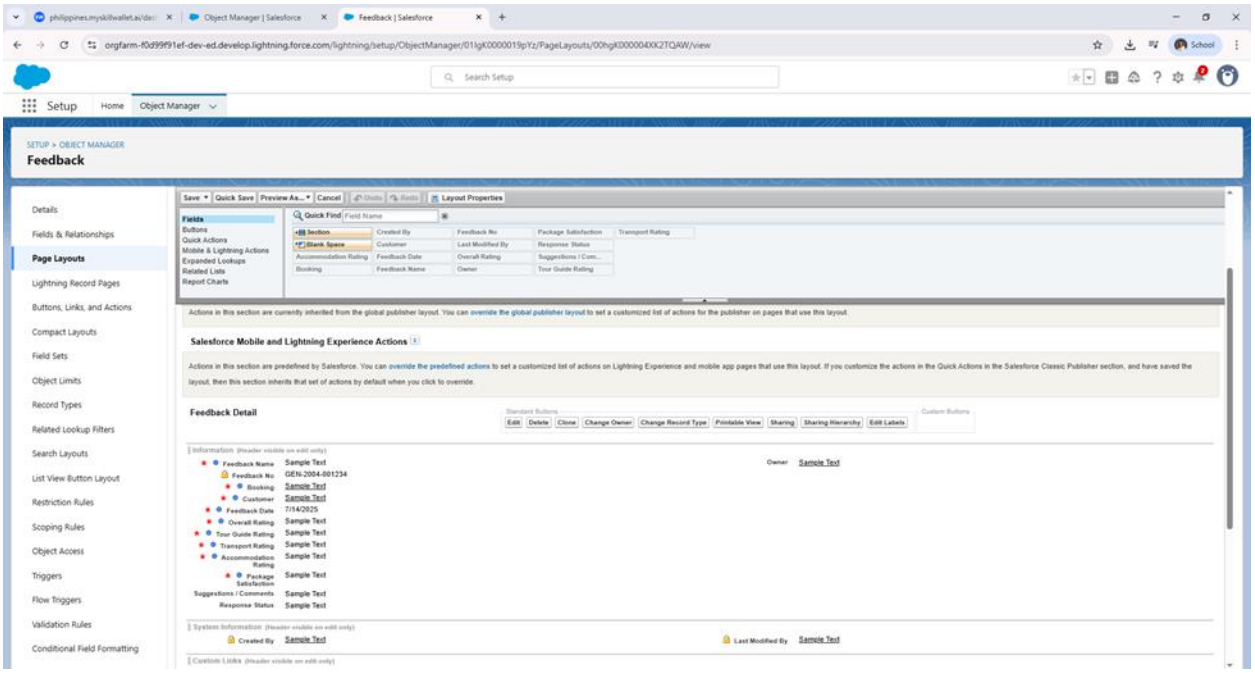

BookingGuest | Salesforce
orgfarm-f0d99f91ef-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgK0000019pPJ/PageLayouts/00hgK000004XJpZQAW/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
BookingGuest
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Save
Quick Save
Preview As...
Cancel
Undo
Redo
Layout Properties
Fields
Buttons
Quick Actions
Mobile & Lightning Actions
Expanded Lookups
Related Lists
Report Charts
Quick Find
Field Name
Section
Blank Space
Booking
Created By
Relation with Cus...
Age
Age Category
Special Needs
Visa Required
Label: Booking
Name: Booking
Type: Lookup
This item is currently in use (click to locate)
layout, then this section inherits that set of actions by default when you click to override.
BookingGuest Detail
Standard Buttons
Edit
Delete
Clone
Change Owner
Change Record Type
Printable View
Edit Labels
Custom Buttons
Information (Header visible on edit only)
BookingGuest Name
Sample Text
Age
68.087
Gender
Special Needs
Passport Number
Visa Required
Country
City
Relation with Customer
Age Category
Booking
System Information (Header visible on edit only)
Created By
Last Modified By
Custom Links (Header visible on edit only)
Mobile Cards (Salesforce mobile only)
Drag expanded lookups and mobile-enabled Visualforce pages here to display them as mobile cards.


TravelPackage | Salesforce
orgfarm-f0d99f91ef-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgK0000019pQv/PageLayouts/00hgK000004XJrBQAW/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
TravelPackage
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Save
Quick Save
Preview As...
Cancel
Undo
Redo
Layout Properties
Fields
Buttons
Quick Actions
Mobile & Lightning Actions
Expanded Lookups
Related Lists
Report Charts
Quick Find
Field Name
Section
Blank Space
Average Rating
Guide Included
Maximum Group Size
Package Type
Region
Country
Guide LanguagesPl...
Meals Included
Places Covered
Transportation Modes
Additional Notes
Created By
Insurance Included
Membership
Preferred Guide L...
TravelPackage Name
Availability Status
Duration in Days
Last Modified By
Owner
Price Per Person
Visa Assistance
Label: Duration in Days
Name: Duration_in_Days
Type: Number
Length: 10
Decimal Places: 0
This item is currently in use (click to locate)
Information (Header visible on edit only)
TravelPackage Name
Sample Text
Country
Region
Duration in Days
23.296
Package Type
Places Covered
Price Per Person
$123.45
Membership
Transportation Modes
Meals Included
Visa Assistance
Insurance Included
Guide Included
Guide LanguagesPlaces Covered
Preferred Guide Language
Maximum Group Size
5.231
Additional Notes
Availability Status
Average Rating
21.100
Owner
System Information (Header visible on edit only)
Created By
Last Modified By
Custom Links (Header visible on edit only)
Mobile Cards (Salesforce mobile only)


Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Employee
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Employee Layout
Employee Sample
Highlights Panel
Quick Actions in the Salesforce Classic Publisher
Salesforce Mobile and Lightning Experience Actions
Employee Detail
Edit
Delete
Clone
Change Owner
Change Record Type
Printable View
Sharing
Sharing Hierarchy
Edit Labels
Employee Name
Employee ID
Email
Phone
Role
Department
Joining Date
Employment Type
Availability Status
Languages Spoken
Address
Country
City
Assigned Region
Profile Picture
Salary
Owner


Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Booking
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Save
Quick Save
Preview As...
Cancel
Undo
Redo
Layout Properties
Booking Name
Booking Number
Customer
TravelPackage
Customer Email
Travelling Start Date
Include Travel Insurance
Require Visa Assistance
Require Tour Guide
Preferred Guide Language
Guide Assigned
Number of Travelers
Total Accommodation Amount
Approval Status
Travelling End Date
Trip Type
Total Billing Amount
Cancellation Date
Cancel Confirmation
Cancellation Reason
Age
Country
Booking Payment
Owner
Membership Chosen
Preferred Accommodation
Accommodation Amount per Person per Day
Total Travel Amount
Travel Cost Per Person
No of Booking Guests Info Available
Booking Date
Booking Status



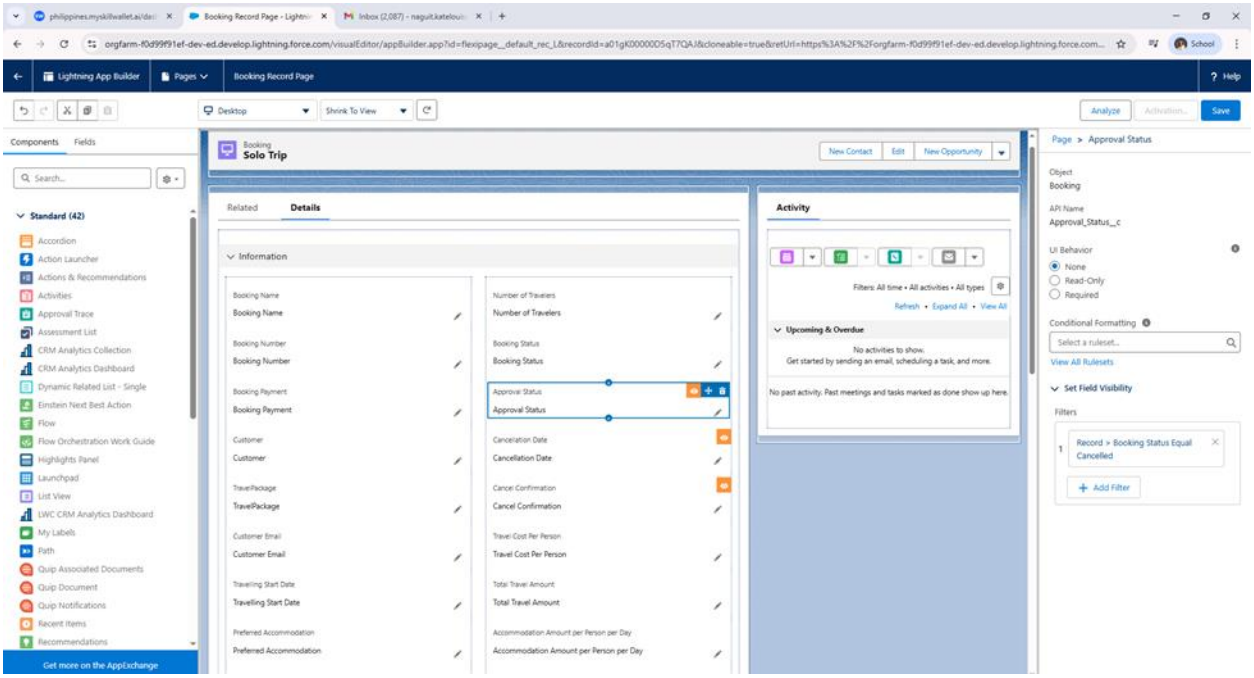


Milestone 15: Dynamic Forms

To create a more responsive user interface in the Tours & Travels CRM, I implemented Dynamic Forms for the Booking object. First, I accessed the app via the App

Launcher and went to the Booking tab to create a sample record. After saving the record, I opened the Lightning App Builder by clicking the gear icon and selecting Edit Page.

Inside the builder, I enabled Dynamic Forms by upgrading the Details section and selecting the existing Booking Page Layout. This allowed me to apply visibility filters to specific fields so they appear only under certain conditions. For example, I set the Cancellation Date, Cancel Confirmation, and Approval Status fields to display only when the Booking Status is set to "Cancelled." After configuring these filters, I saved and activated the changes, applying them to both Desktop and Mobile views. This setup ensures that users only see relevant fields based on the booking's current status, improving usability and reducing clutter in the interface.



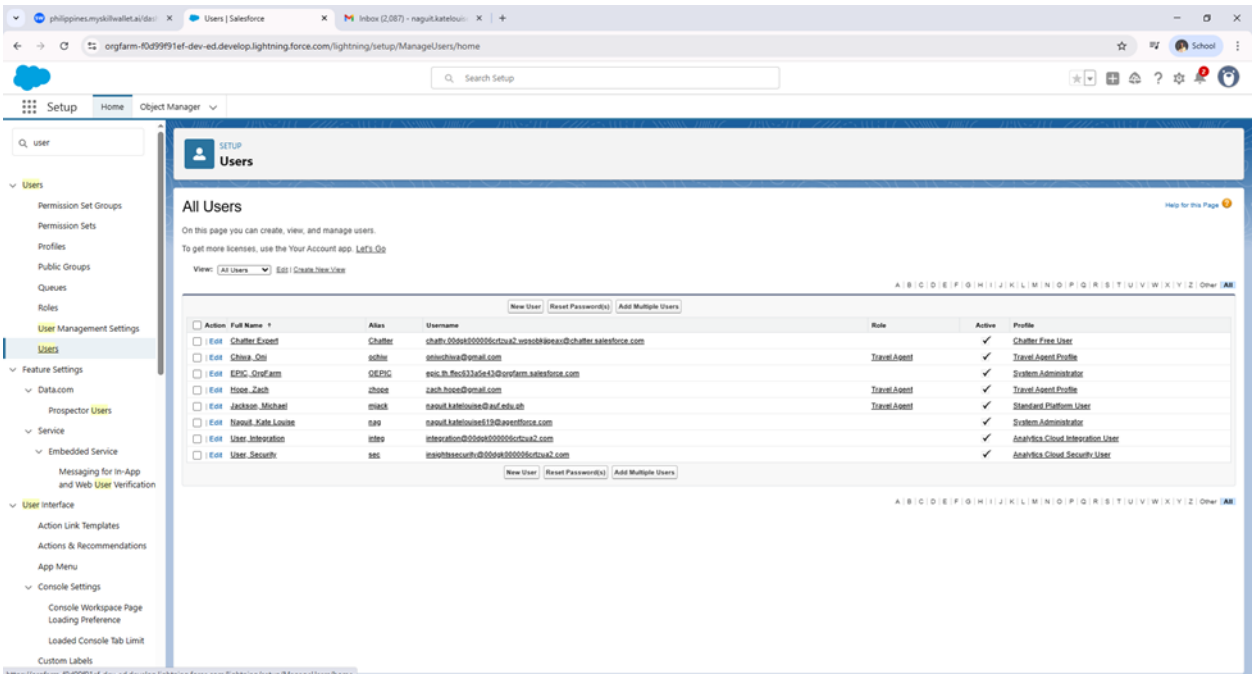
Milestone 16: User Management

Before I started creating users in the Tours & Travels CRM, I made sure that the Profiles and Roles were already set up, since these are required to assign the right access

levels to each user. Once everything was ready, I went to Setup, searched for Users in the Quick Find box, and clicked New User to begin the process.

I started by creating a user named Michael Jackson. I filled out the necessary details such as the first name, last name, alias, a valid email address, and a unique username. I also assigned a nickname for easier identification within the system. For access, I set the Role to Travel Agent Manager Role, selected Salesforce Platform as the User License, and chose Travel Agent Profile as the profile.

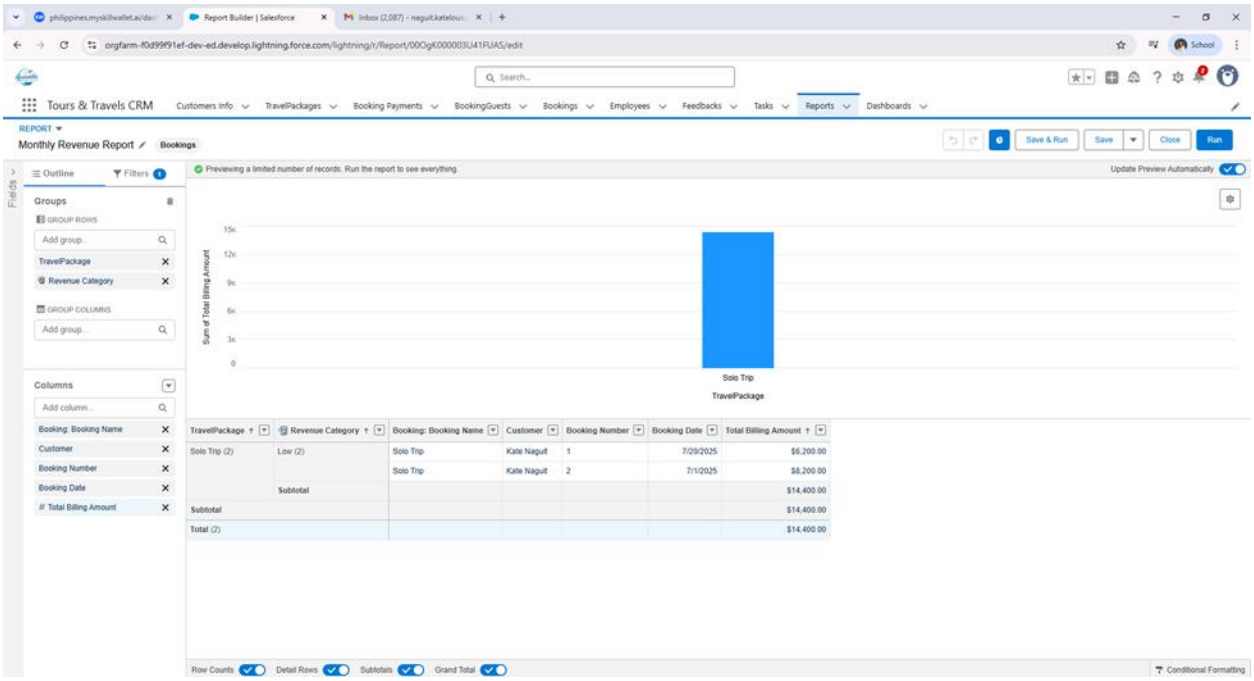
After saving the first user, I continued by adding more users. For the rest, I assigned them the Travel Agent Role but kept the same license and profile settings. I made sure to create at least two users per role so that each function in the system would be properly represented and could be tested accordingly. This step helped me ensure that every user had the appropriate level of access and could interact with the system based on their responsibilities.

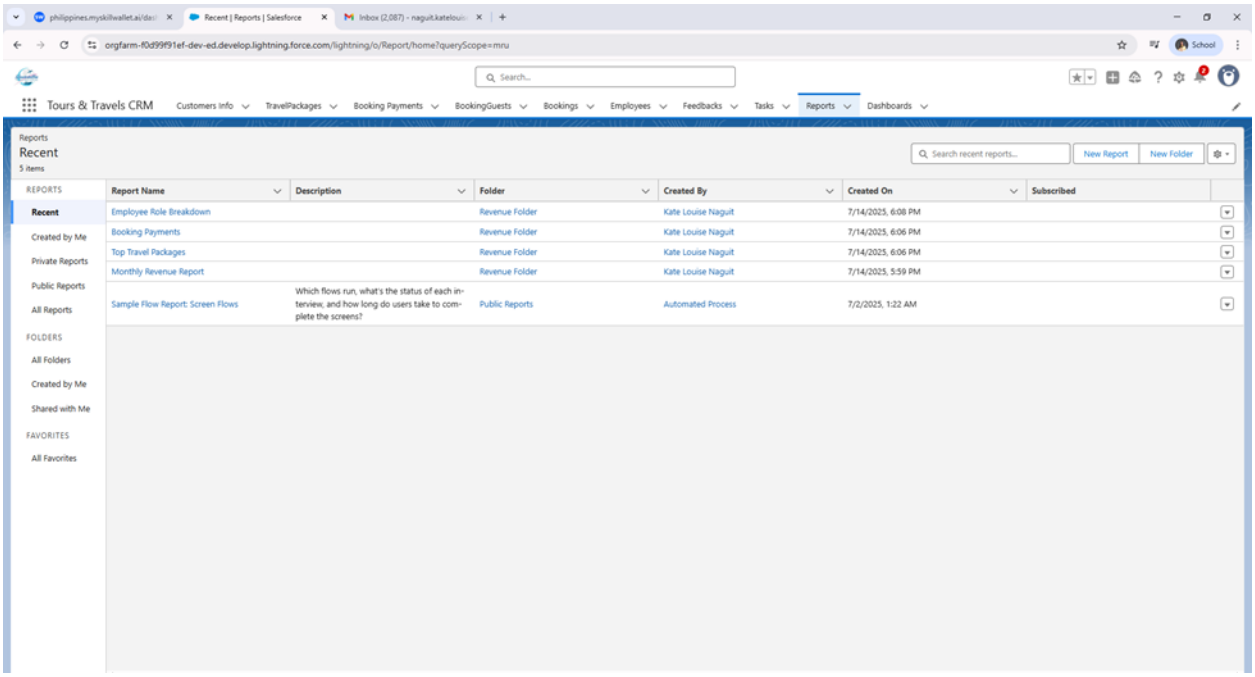


Milestone 17: Reports

To begin setting up the reports for the Tours & Travels CRM, I opened the App Launcher, searched for Reports, and clicked on the Reports tab. To create the Monthly Revenue Report, I started by clicking on “New Report.” I selected “Bookings” as the report type because I wanted to analyze customer travel bookings. I named the report “Monthly Revenue Report” and moved on to structuring it. In the Outline panel, I added “TravelPackage” to the row groupings to categorize the data by the type of travel package, such as “Solo Trip.” I also added “Revenue Category” as a second row grouping to break down the bookings by revenue classification like Low, Medium, or High.

Next, I included the necessary columns in the report: Booking Name, Customer, Booking Number, Booking Date, and Total Billing Amount. These fields gave me a detailed view of each booking's specifics. Once the data was loaded, a customer had made two bookings for the Solo Trip package: one on July 1, 2025 for $8,200, and another on July 20, 2025 for $6,200, with a total revenue of $14,400.00. A bar chart was automatically generated, visually representing the total billing amount for the Solo Trip package. After reviewing everything, I clicked “Save & Run” to finalize and view the report.





Milestone 18: Dashboards

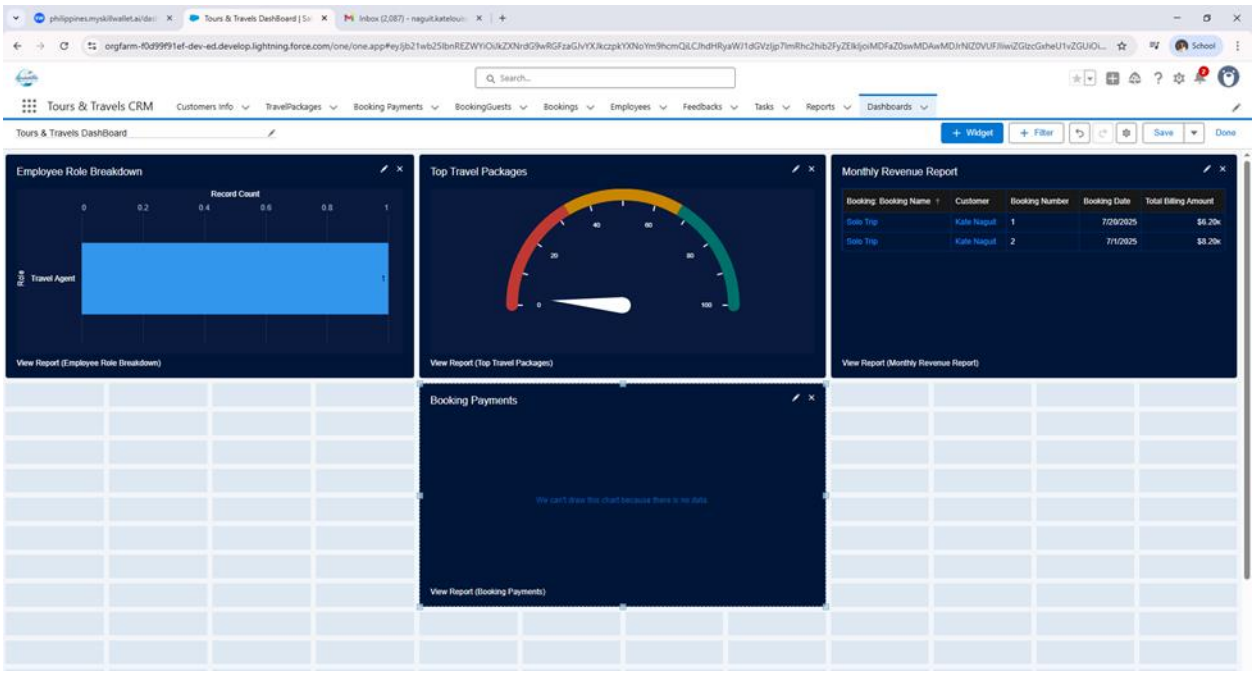
After creating the reports I needed, I proceeded to build the Tours & Travels Dashboard to visually present key metrics for easier monitoring. I navigated to the

Dashboards tab and clicked “+ New Dashboard”, giving it the name “Tours & Travels Dashboard.” Inside the dashboard builder, I added multiple widgets by clicking the “+ Widget” button and selected the appropriate reports to display.

First, I added the “Employee Role Breakdown” report using a vertical bar chart widget to show how many employees are assigned to each role-in this case, it displayed that there’s currently 1 Travel Agent in the system. Next, I added a gauge chart for the “Top Travel Packages” report. Although the data is minimal, the gauge is configured to reflect booking activity, currently showing a value near the low end, indicating limited package diversity or engagement.

Then, I added the “Monthly Revenue Report” as a table widget to display detailed booking information. This table shows two bookings for the Solo Trip package, with billing amounts of $6,200 and $8,200, dated July 20, 2025, and July 1, 2025, respectively. I also included a widget for the “Booking Payments” report; however, it currently shows a message stating “We can’t draw this chart because there is no data,” indicating that no booking payment records are available yet.

Once all the widgets were placed and configured, I clicked “Save” and then “Done” to complete the dashboard. This setup now provides a quick, interactive overview of key operational metrics such as employee roles, top travel packages, booking revenue, and payment activity.



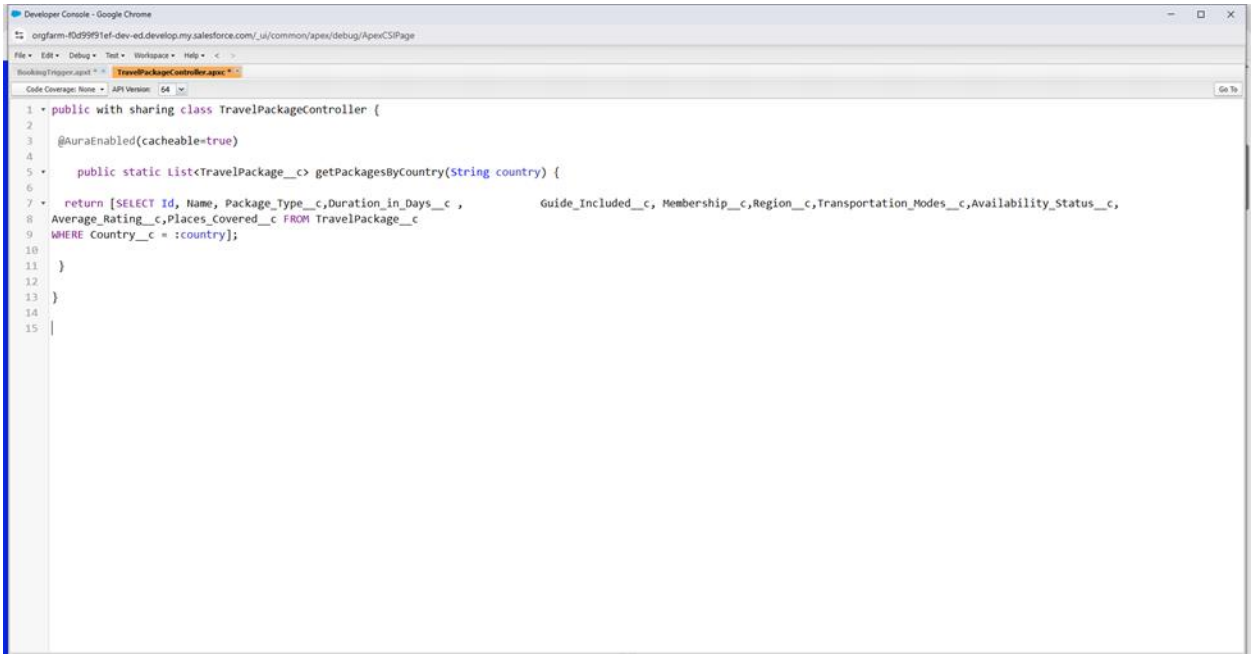
Milestone 19: Lightning Web Components (LWC) Development

I created an Apex controller class named TravelPackageController to allow my LWC to retrieve travel package records based on a selected country. I opened the Developer Console in Salesforce and wrote the Apex class with the *with sharing* keyword to ensure that the controller respects the user's sharing rules. Inside the class, I created a public static method called getPackagesByCountry, which accepts a String parameter named country.

I annotated this method with @AuraEnabled(cacheable=true) to make it accessible from LWC and to enable client-side caching for performance optimization. The method returns a list of TravelPackage__c records by executing a SOQL query that selects various fields such as Id, Name, Package_Type__c, Duration_in_Days__c, Guide_Included__c, Membership__c, Region__c, Transportation_Modes__c, Availability_Status__c, Average_Rating__c, and Places_Covered__c from the TravelPackage__c object. The

query uses a WHERE clause to filter records where Country__c matches the input parameter.

This Apex controller acts as the backend logic for my LWC component, allowing users to dynamically view travel packages specific to a selected country. Once I saved the class, I proceeded to connect it with the LWC JavaScript file using @wire or import syntax so that the data can be fetched and rendered on the frontend.



Milestone 20: Lightning App Page Creation

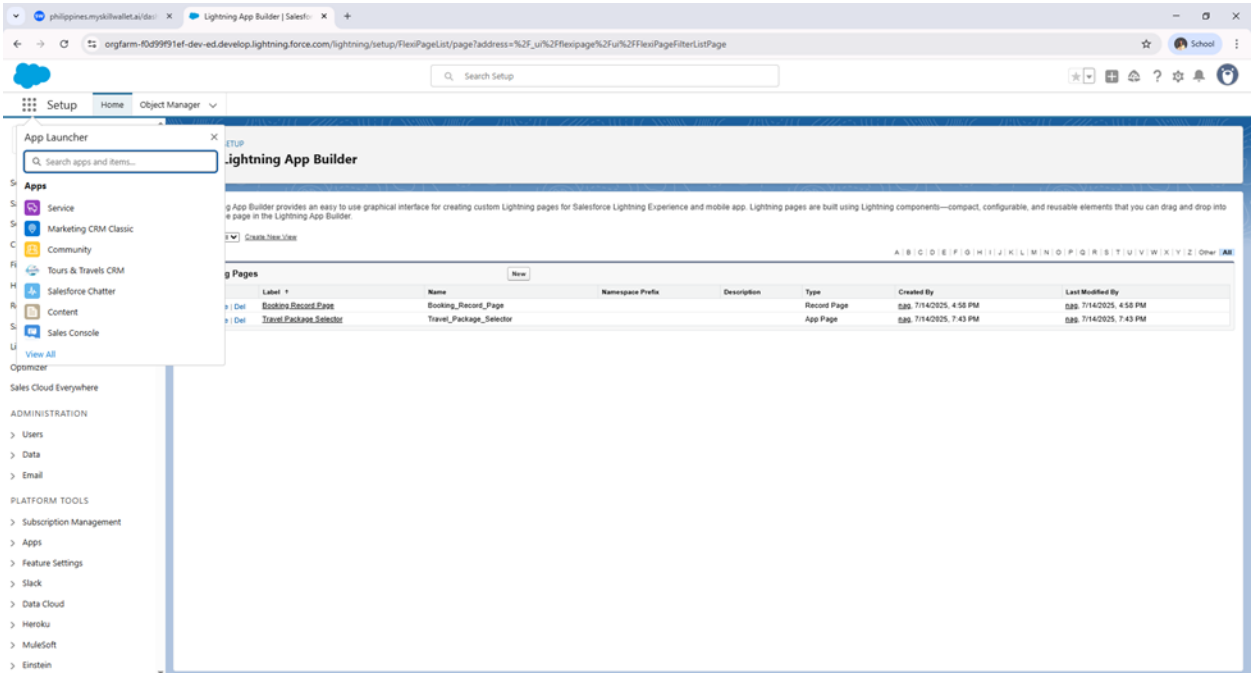
To improve the user interface and make the system more interactive, I created a dedicated Lightning App Page that features my custom Lightning Web Component.

I started by going to Setup and opening the Lightning App Builder. From there, I selected New App Page and named it Travel Package Selector. I chose a single-region layout to maintain a clean and focused design that puts the spotlight on the main functionality.

Next, I dragged my travelpackageselector component into the layout area and saved the changes. To ensure users could access it easily, I clicked Activate, then added the page to the Tours & Travels CRM app.

Finally, I went to the App Launcher to verify that both the CRM app and the new standalone app page were available. When users open this page, they can select a country and view the corresponding travel packages dynamically making the process more seamless and engaging.

This milestone helped improve overall data visibility and enhanced the user experience by providing a more interactive and responsive interface within the CRM system.



Phase 4: Data Migration, Testing & Security

In this phase, the focus was on migrating accurate data into the system, enforcing robust security measures, and conducting comprehensive testing to ensure the stability and reliability of the Tours & Travels CRM platform.

The Data Import Wizard was utilized to load initial data for the Customer Info, TravelPackage, and Employee objects. Each CSV file included a minimum of 20 records with complete and properly formatted information. The process involved uploading the files, verifying field mappings, and monitoring the import status through the Bulk Data Load Jobs section. After import completion, the data was reviewed to ensure accuracy and proper integration within the system.

To support data transparency and integrity:

- Field History Tracking was enabled on key fields in the Booking and TravelPackage objects, such as Booking Status, Number of Travelers, Price Per Person, and Availability Status. This allows the system to log historical changes for audit and reference purposes.
- A Matching Rule was created to identify duplicates in the Customer Info object by checking for exact matches in the Email and Phone Number fields, including cases with blank entries.
- A Duplicate Rule was implemented to work with the matching rule. Instead of blocking duplicate entries, it displays a warning message ("Email and Phone must be Unique") to inform users while still allowing flexibility.

Access control was structured using Profiles, Roles, Permission Sets, and Sharing Rules:

- Custom Profiles were created for each user group, including Travel Agents, Tour Guides, Finance Officers, Marketing Executives, and Customer Service

Representatives. Permissions were tailored based on job functions, with varying access levels across objects.

- A clear Role Hierarchy was established, placing department-specific roles under the CEO. This setup ensures data visibility flows appropriately, with managers having access to their subordinates' records.
- Permission Sets were used to extend access without altering base profiles. For example, a permission set named Extra Permission for Travel Agent Manager was created to grant full access to the TravelPackage object.
- Sharing Rules were configured to balance security with collaboration. The Customer Info object was set to Private by default, with specific read-only sharing rules applied such as allowing Tour Guides to view records owned by Travel Agents.

An Apex test class named BookingTriggerTest was created to validate the Booking trigger functionality. The test simulated realistic booking scenarios by inserting sample customer and travel package records. Assertions were used to verify the automatic creation of Booking_Payment__c and BookingGuest__c records. The test execution was wrapped in Test.startTest() and Test.stopTest() to ensure isolation and accurate performance monitoring, helping meet Salesforce’s deployment requirements.

Comprehensive test cases were prepared to evaluate the functionality of all major CRM components. Each test case included:

- A unique identifier and test case name
- The involved object(s)
- Detailed step-by-step instructions
- Expected outcomes
- Actual results and status (e.g., Passed/Failed)

The test scenarios covered:

- Adding new employee and customer records
- Booking creation with automated payment and guest records
- Validation rules and duplicate detection
- Flows, triggers, and approval processes
- Automatic task creation
- System responses for each action

All test cases were documented with corresponding input and output screenshots as part of the testing deliverables.

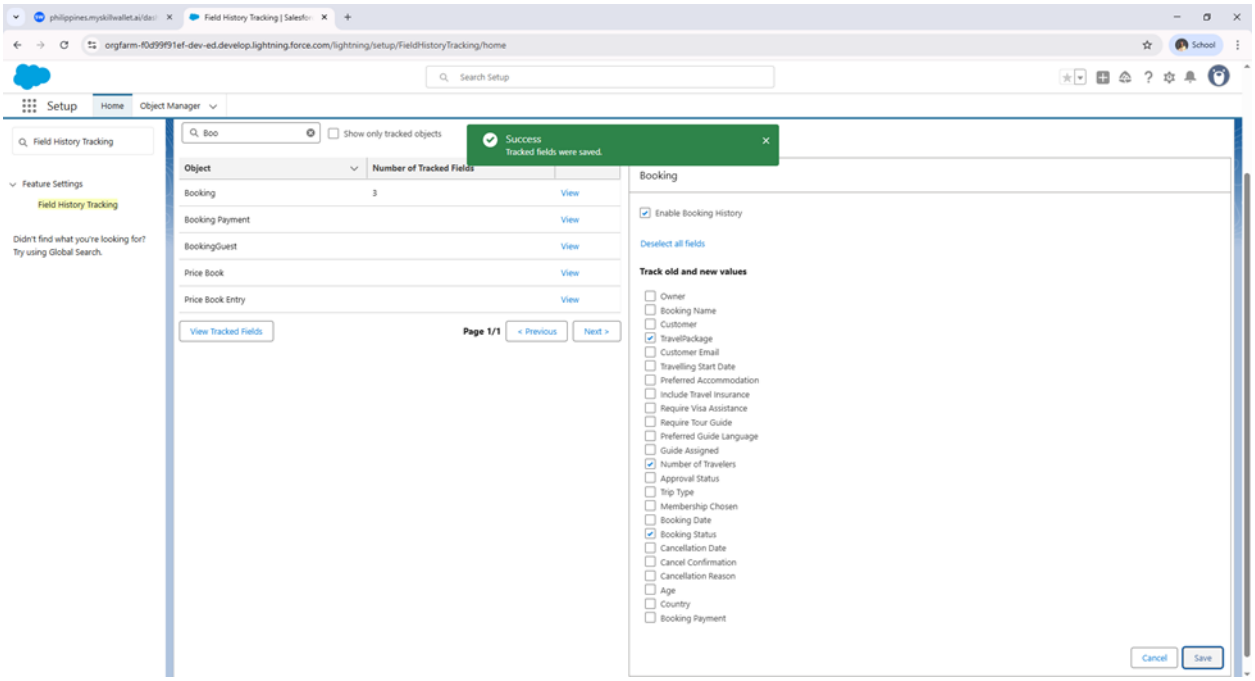
By the end of this phase, the CRM system was equipped with clean data, safeguarded through well-defined security layers, and rigorously tested to confirm functionality and performance across all critical features.

Milestone 21: Field History Tracking

To make sure key data changes in the system are traceable, I enabled Field History Tracking for important fields across selected objects in the Tours & Travels CRM.

I started with the Booking object, where I enabled tracking for the Number of Travelers, Booking Status, and TravelPackage fields. This means that any changes made to these fields are now automatically recorded in the History related list, helping maintain a clear record of updates. Also, I set up tracking for the TravelPackage object, specifically for the Price Per Person and Availability Status fields. These are critical for monitoring pricing adjustments and availability changes over time.

By implementing field history tracking, I created a reliable audit trail that strengthens data transparency and accountability within the system.



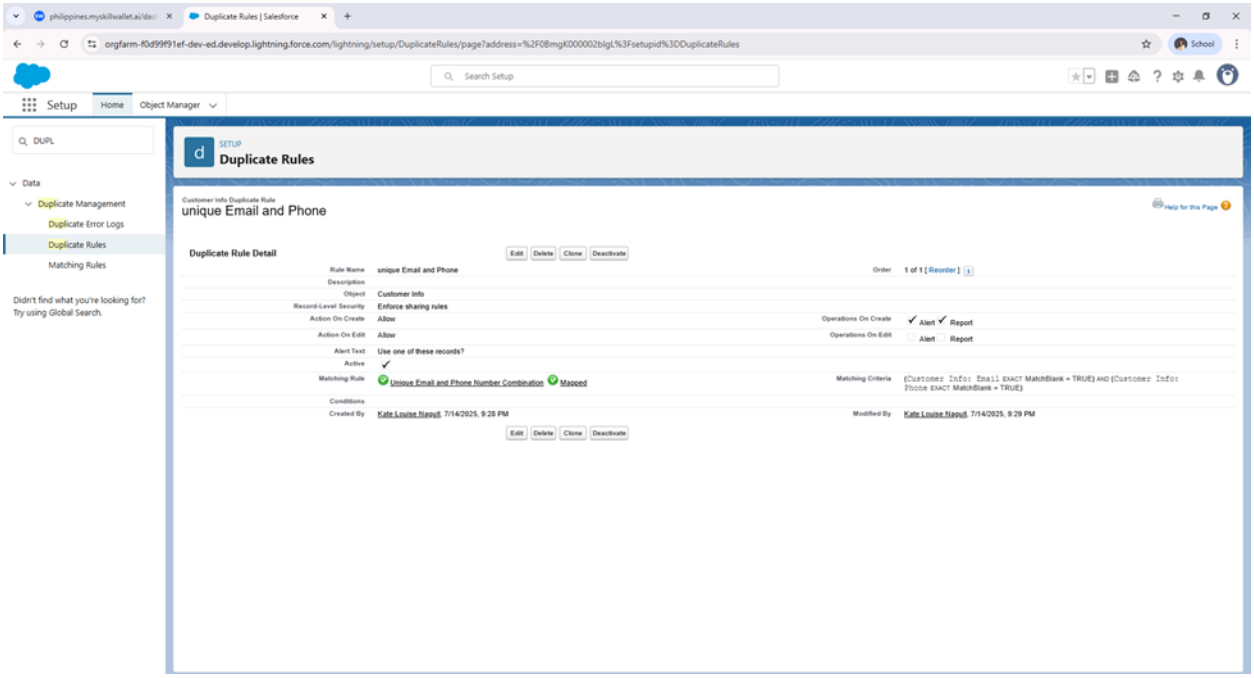
Milestone 22: Duplicate and Matching Rules

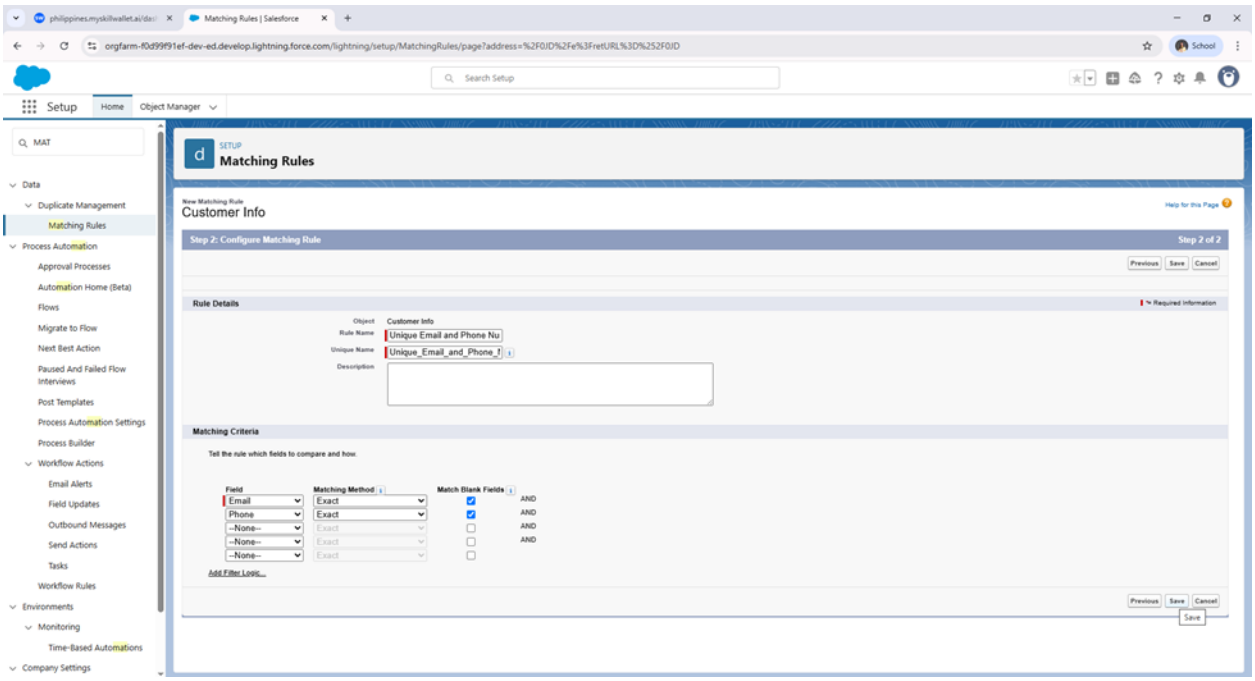
To help maintain clean and accurate data in the Customer Info object, I implemented a matching rule that detects duplicate records based on an exact match of both email and phone number. I named this rule "Unique Email and Phone Number Combination" and enabled the "Match Blank Fields" option so the system could still evaluate potential duplicates even if one of the fields is empty. Once everything was set, I saved and activated the rule.

After that, I created a duplicate rule called "Unique Email and Phone", which determines what happens when a duplicate is found. Instead of blocking the creation or editing of records, this rule simply gives users a warning message: "Email and Phone must

be Unique." I connected the earlier matching rule to this setup for consistency, then saved and activated the rule.

With both rules in place, the CRM now provides a smoother way to detect duplicates without disrupting the user's workflow, ensuring more reliable customer data.



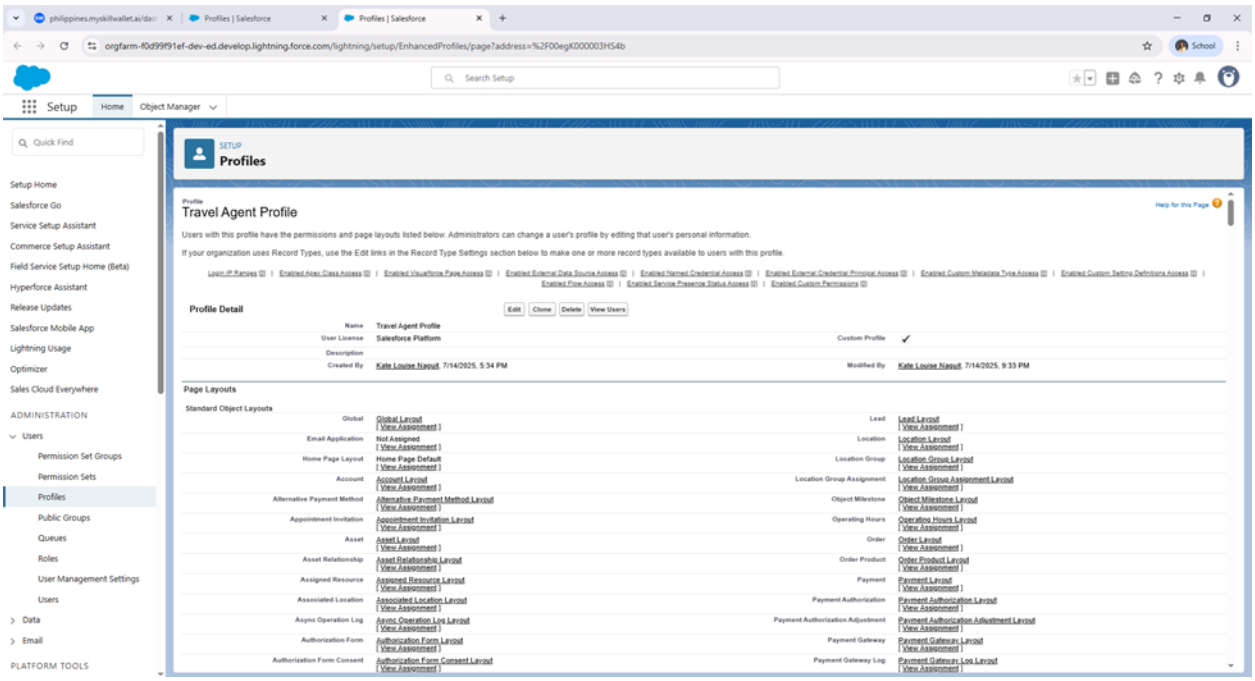


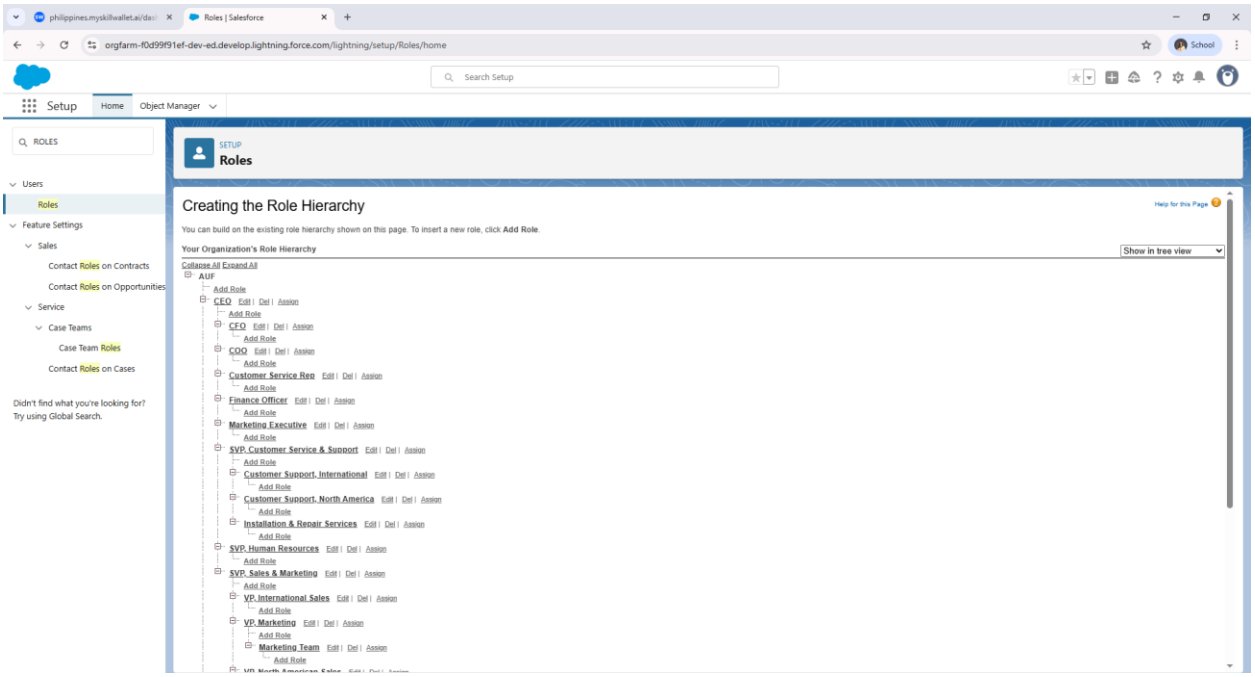
Milestone 23 & 24: Profiles and Roles

To manage user access based on specific job roles in the Tours and Travels CRM, I created several custom profiles by cloning existing ones like "Standard Platform User" and "Salesforce Platform User." I built the Travel Agent Profile by cloning "Standard Platform User." I set the default app to the Tours and Travels CRM and configured object permissions granting full Create, Read, and Edit rights for key objects like Bookings, Booking Guests, Booking Payments, Customer Info, and Travel Packages, while providing read-only access to Employee and Feedback objects. I also enforced a 2-hour session timeout and set password rules with a minimum of 8 characters and no expiration. Then, I created the Tour Guide Profile from the "Salesforce Platform User." Since this role needed limited access, I only enabled Read permissions for Bookings, Booking Guests, and Travel Packages. For the Finance Officer Profile, also based on "Salesforce Platform User," I allowed full access to Bookings and Booking Payments, but only view access to Travel Packages and Employee records giving them control over transactions without modifying HR or sales data. The Marketing Executive Profile was tailored for handling promotions. I

granted full access to the Travel Package object and read-only access to related data such as Bookings, Guests, Customer Info, Employees, and Feedbacks allowing visibility while preventing unintended edits. Finally, I created a Customer Service Rep Profile that focused on managing customer feedback. This profile has full rights over the Feedback object and read-only access to Bookings, Customer Info, and Travel Packages ensuring they can help clients without altering sensitive records.

To establish proper data access control and visibility in the Tours and Travels CRM system, I configured a structured role hierarchy. I started by creating the Travel Agent Manager role under the CEO using the Roles setup in Salesforce. This role serves as the supervisor for team members managing travel bookings. Under this manager, I added two sub-roles: Travel Agent, responsible for processing booking requests, and Travel Tour Guide, intended for those leading travel groups with limited system access. I then expanded the hierarchy by adding three more roles directly under the CEO to represent key departments. The Finance Officer Role handles payment records and financial transactions; the Marketing Executive Role manages promotions, package listings, and analytics; and the Customer Service Rep Role supports customer inquiries and feedback. This structured hierarchy ensures that access flows logically from top to bottom, managers can view and control their team's data while each role remains limited to what they specifically need, maintaining both efficiency and data security across the organization.



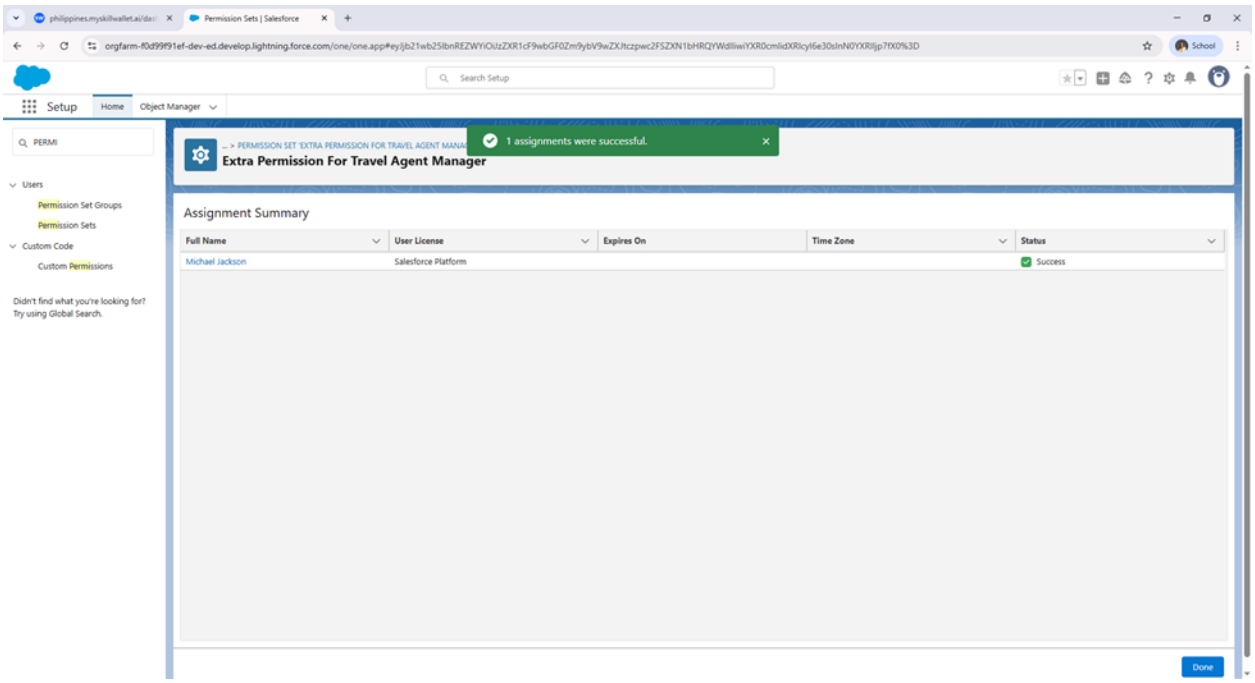


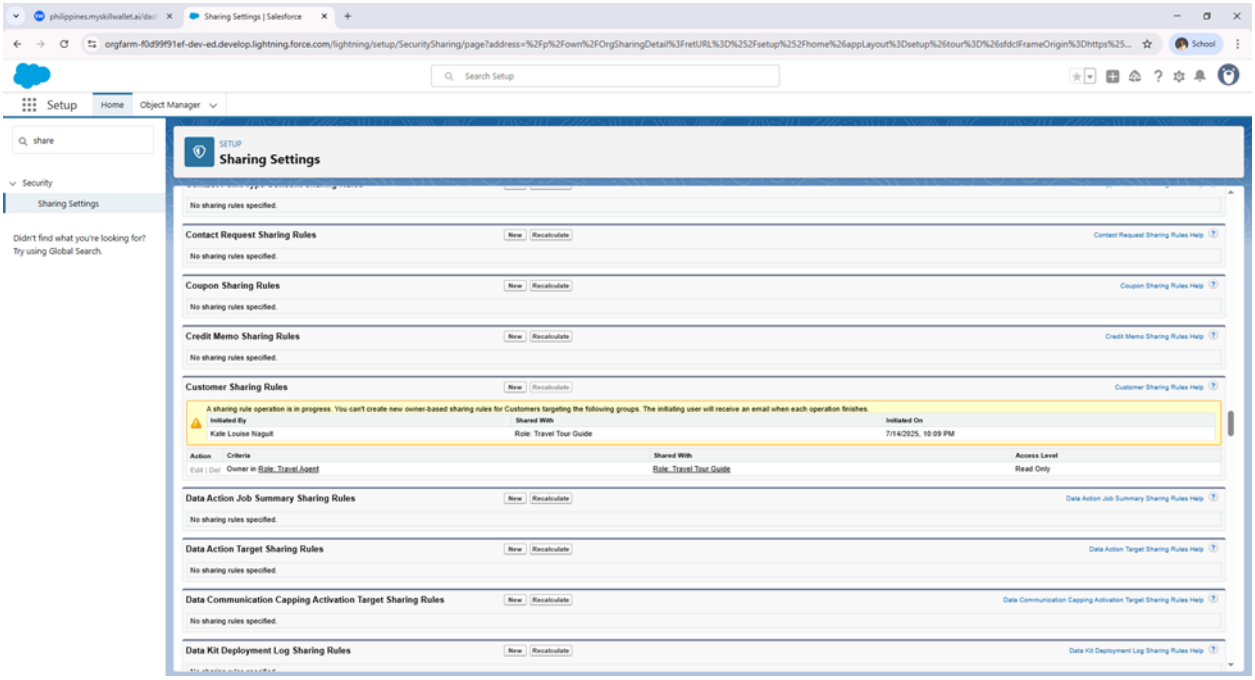
Milestone 25 & 26: Permission Sets and Sharing Rules

I created a custom Permission Set to provide specific users with additional access without modifying their base profile. I navigated to Setup > Permission Sets and created a

new set named "Extra Permission for Travel Agent Manager." This approach allowed me to assign extended privileges in a more flexible and secure way. After creating the permission set, I accessed its Object Settings and configured the TravelPackage object to grant full permissions-Read, Create, Edit, and Delete (R/E/C/D). This ensured that users assigned to this permission set could manage travel packages completely. Once the changes were saved, I assigned the permission set to the appropriate user by going to Manage Assignments, clicking Add Assignment, and selecting the user with the Travel Agent Manager role. This method helps maintain clean and manageable access control by enhancing user privileges without altering their core profile.

I focused on strengthening data security and ensuring appropriate access control within the CRM by configuring sharing settings for customer records. I began by setting the Customer Info object’s organization-wide default (OWD) to Private, which limits access to record owners and their superiors in the role hierarchy. This step was important to protect sensitive customer data. To support collaboration while maintaining security, I created a sharing rule that allows users in the Tour Guide Role to view records owned by those in the Travel Agent Role. I granted read-only access through this rule, enabling tour guides to see necessary customer details without making changes. This setup ensures that data is shared responsibly, promoting team coordination without compromising confidentiality.

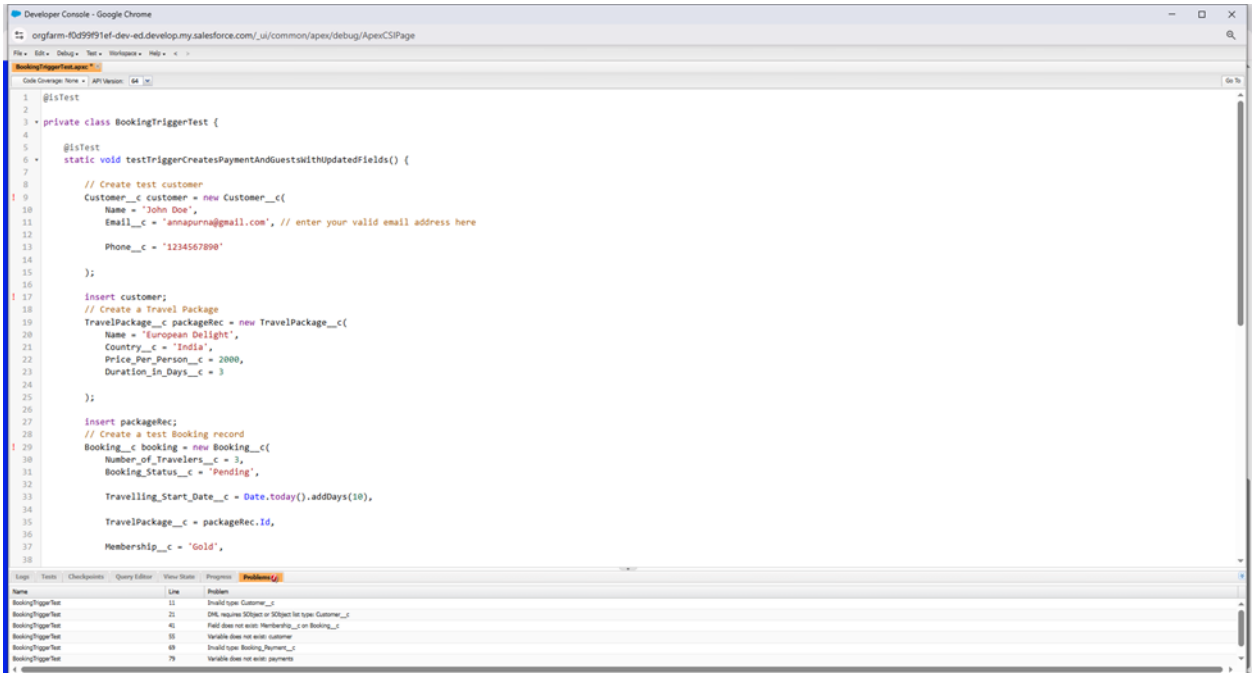




Milestone 27: Creation of Test Classes

I created an Apex test class named BookingTriggerTest to validate the functionality of the Booking Trigger and its associated TriggerHandler. To simulate a realistic scenario,

I first generated a sample customer and a travel package record. I then inserted a new booking that referenced both records. The test was designed to ensure that the trigger automatically creates the correct related records—specifically, a Booking_Payment__c record with a default status of "Pending", and three BookingGuest__c records, one for each traveler. I used assertions to confirm the creation of these records and wrapped the logic in Test.startTest() and Test.stopTest() to isolate the trigger execution and accurately measure its performance. This test not only verified that the trigger logic worked as intended but also helped meet Salesforce's code coverage requirements for deployment.



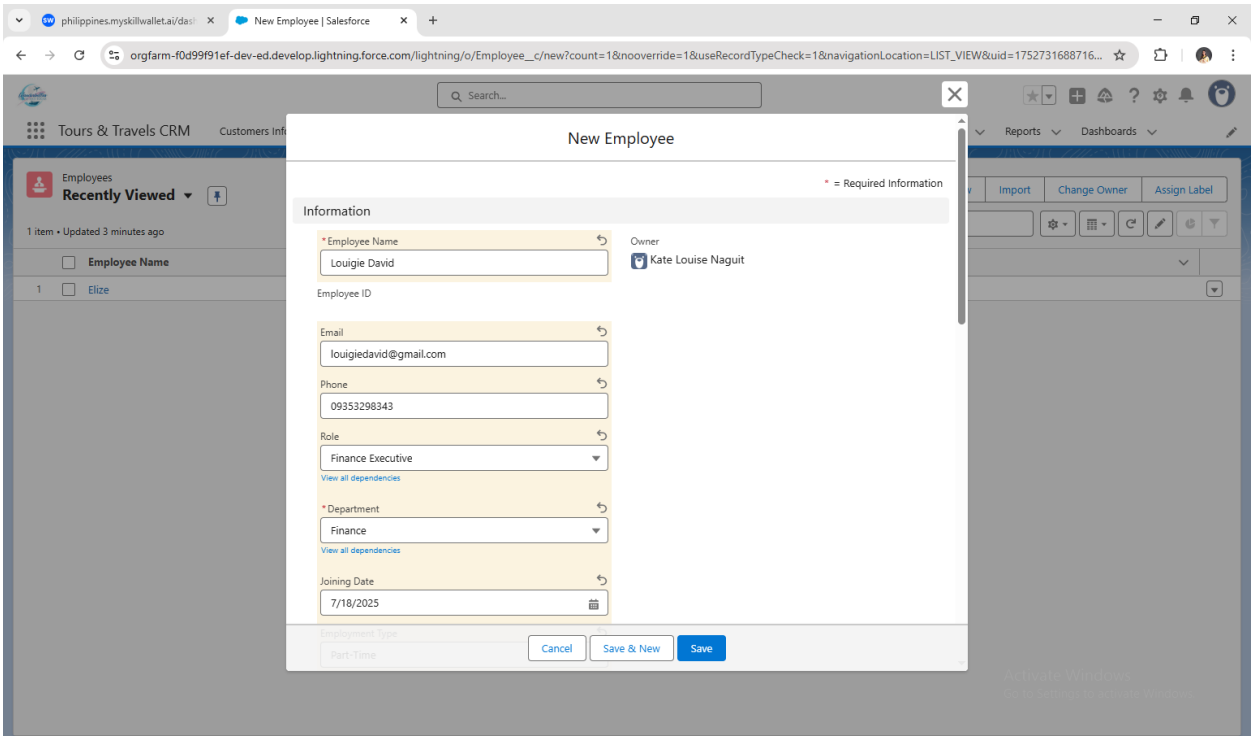
Milestone 28: Preparation of Test Cases

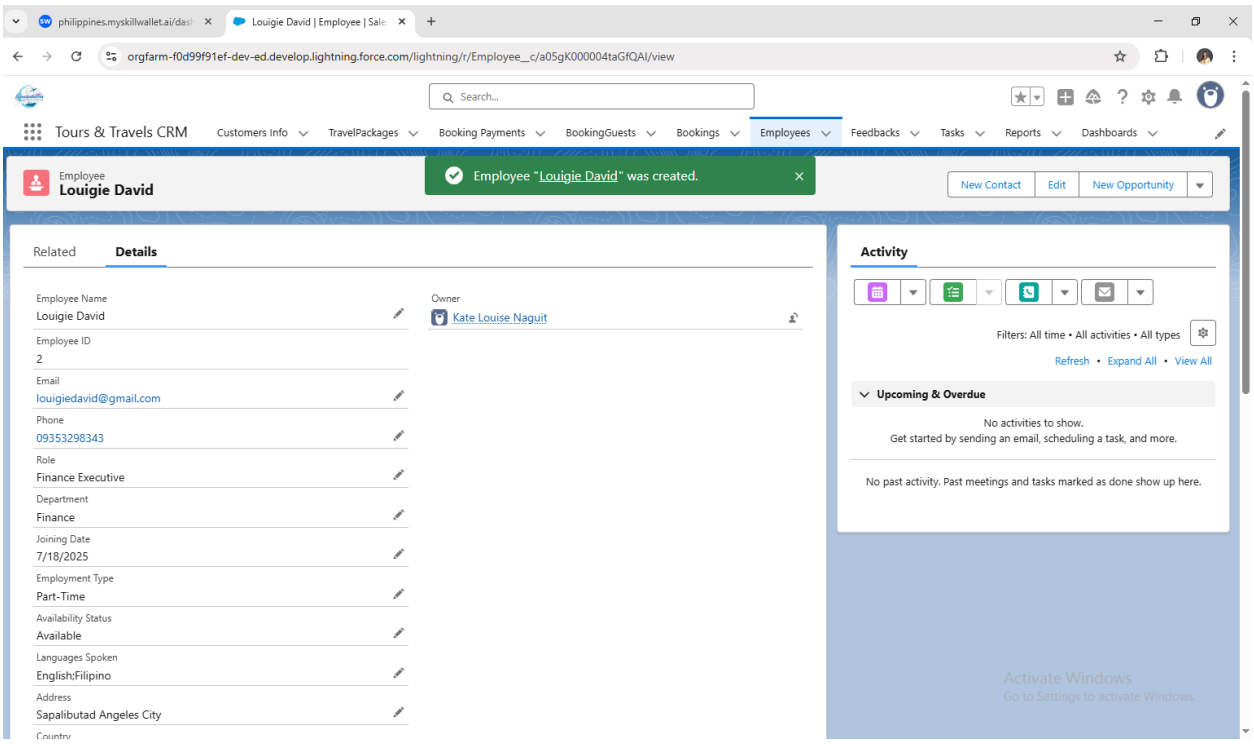
I created comprehensive test cases for each core feature of the CRM to ensure proper functionality and reliability. Each test case was documented with a unique number, a clear test case name, the objects involved, specific step-by-step instructions, the expected result, and the final status.

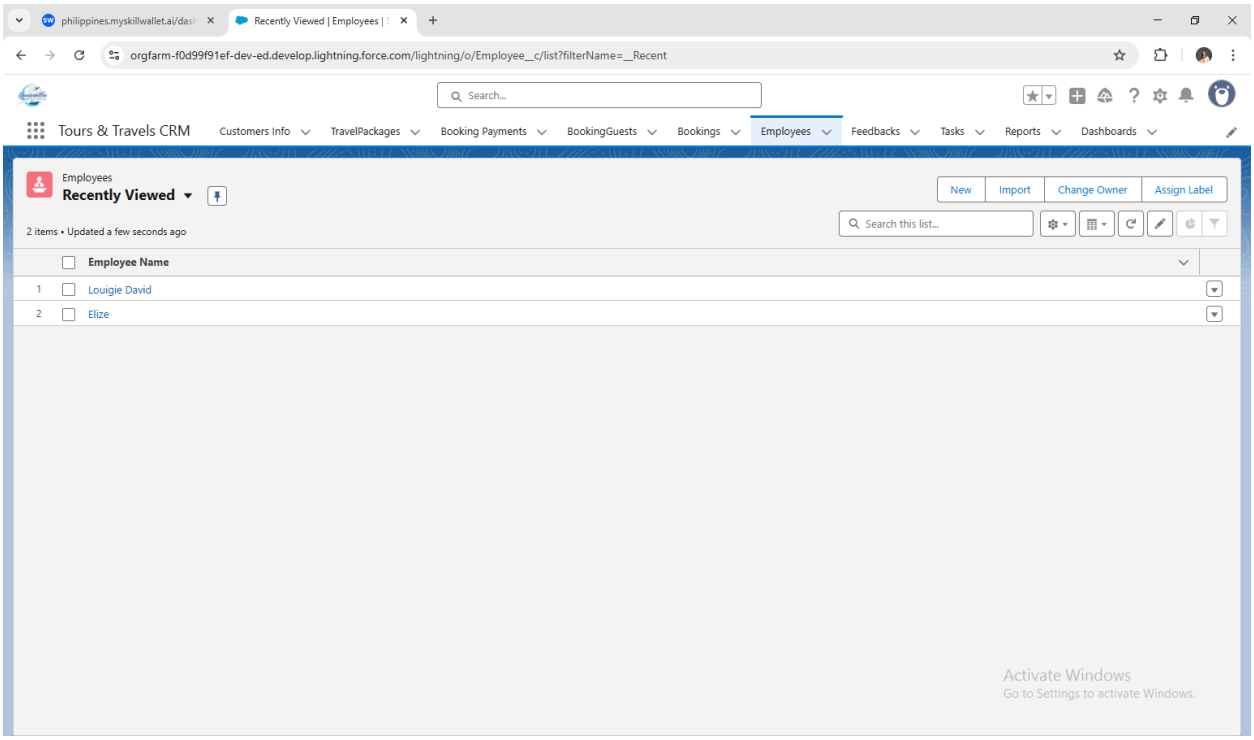
Examples include adding a new employee record, registering a new customer, and creating a booking that automatically generates related payment records. Other test cases covered updating customer information, enforcing validation rules, preventing duplicate entries based on email and phone, and verifying system responses through each action.

No.	Test Case Name	Object(s) Involved	Steps	Expected Result	Status
1	Add Employee Record	Employee	Add a new employee to the system	New employee record is saved and listed	Passed
2	New Customer Entry	Customer Info	Create and save a new customer record	Customer is saved and appears in list view	Passed
3	Create Booking Record	Booking, Payment	Create a booking for a customer	Booking and related payment records are created automatically	Passed
4	Update Customer Info	Customer Info	Edit customer phone number	Changes are saved; validation rules work as expected	Passed
5	Prevent Duplicate Entry	Customer Info	Attempt to create customer with same email and phone	Warning message appears; duplicate prevented	Passed

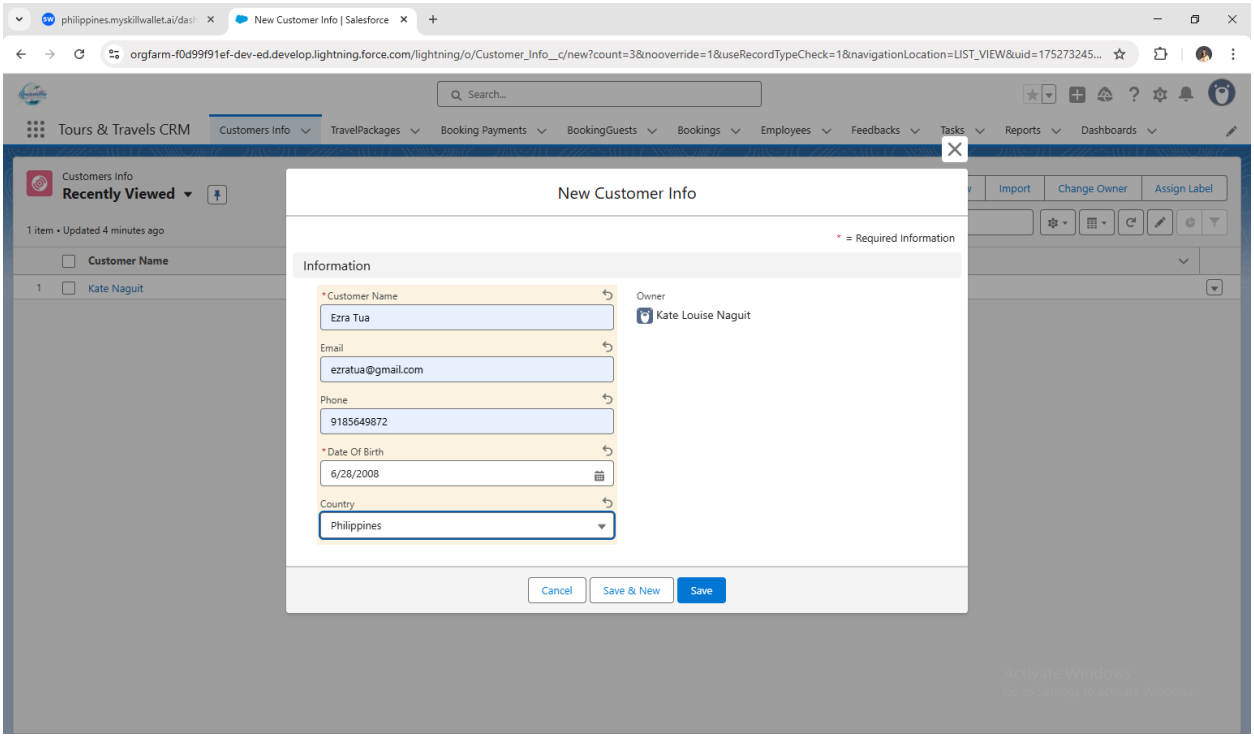
Test Case 1: Add Employee Record

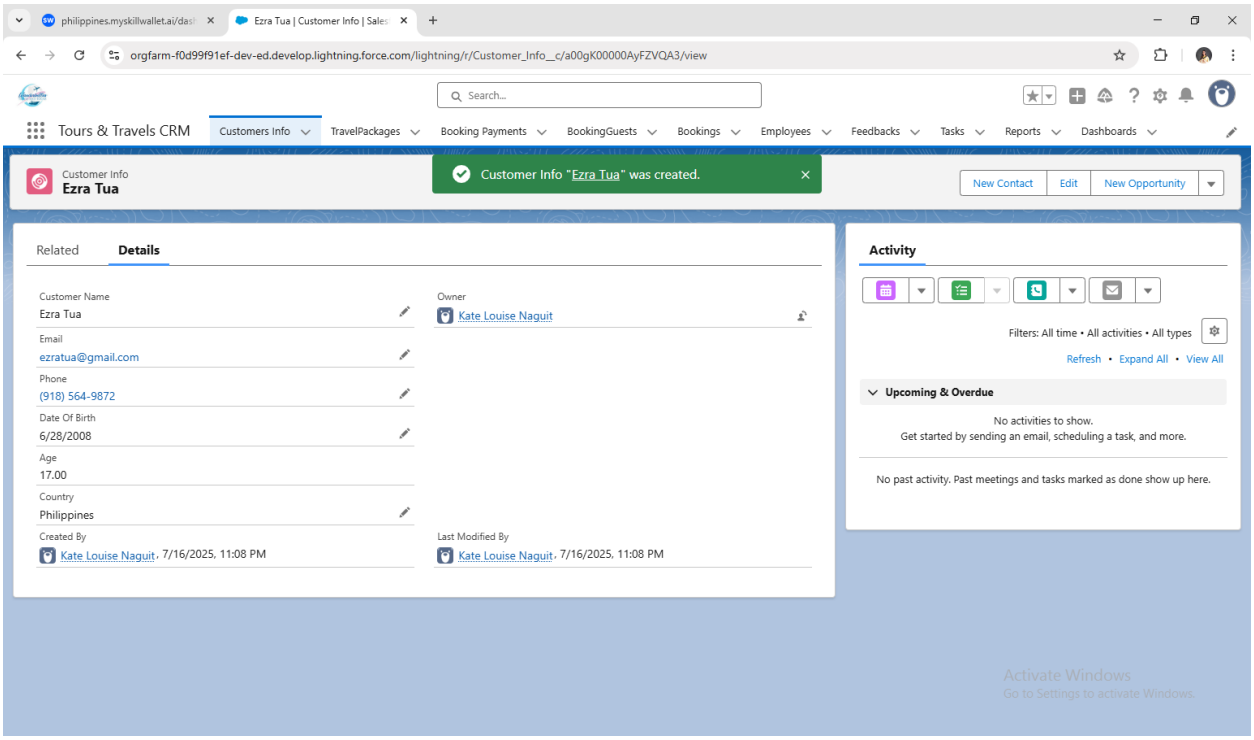

New Employee | Salesforce
orgfarm-f0d99f91ef-dev-ed.develop.lightning.force.com/lightning/o/Employee__c/new?count=1&nooverride=1&useRecordTypeCheck=1&navigationLocation=LIST_VIEW&uid=1752731688716...
Tours & Travels CRM
Customers Info
Reports
Dashboards
Employees
Recently Viewed
1 item • Updated 3 minutes ago
Employee Name
Elize
Import
Change Owner
Assign Label
New Employee
* = Required Information
Information
* Employee Name
Louigie David
Owner
Kate Louise Naguit
Employee ID
Email
louigiedavid@gmail.com
Phone
09353298343
Role
Finance Executive
View all dependencies
* Department
Finance
View all dependencies
Joining Date
7/18/2025
Employment Type
Part-Time
Cancel
Save & New
Save

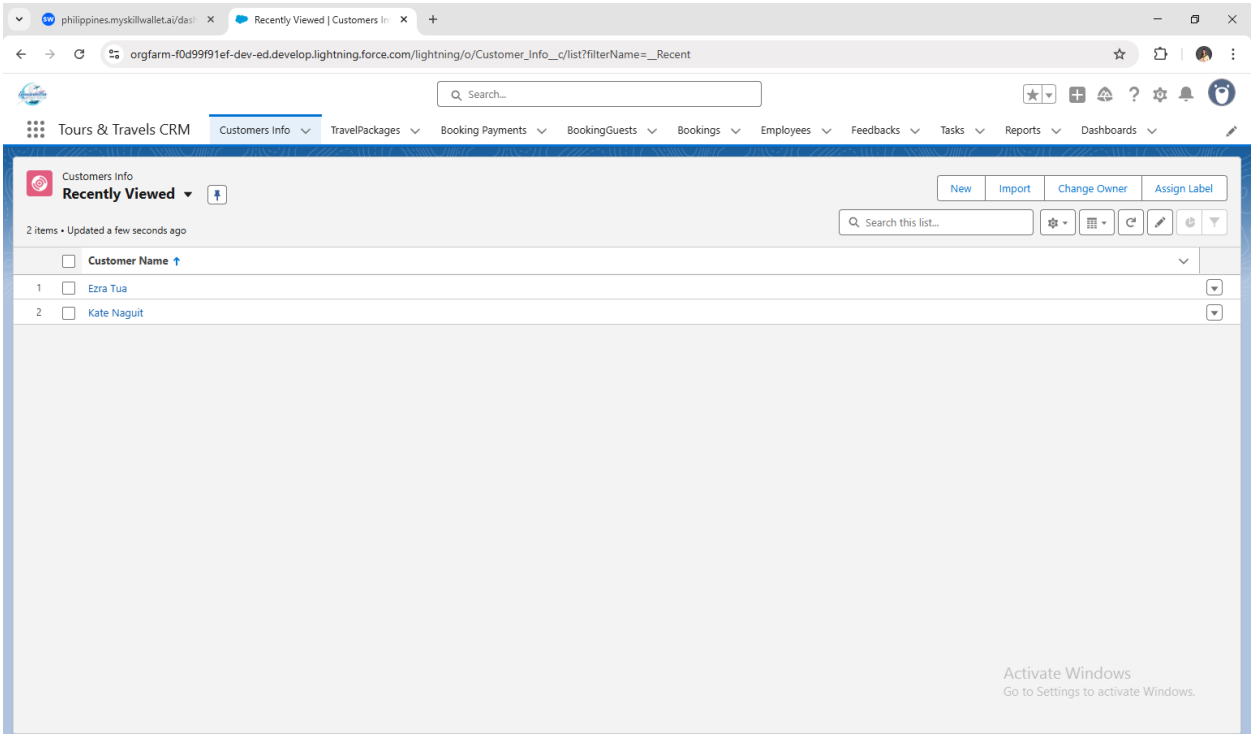

Louigie David | Employee | Sale
orgfarm-f0d99f91ef-dev-ed.develop.lightning.force.com/lightning/r/Employee__c/a05gK000004taGfQAI/view
Tours & Travels CRM
Customers Info
TravelPackages
Booking Payments
BookingGuests
Bookings
Employees
Feedbacks
Tasks
Reports
Dashboards
Employee
Louigie David
Employee "Louigie David" was created.
New Contact
Edit
New Opportunity
Related
Details
Employee Name
Louigie David
Owner
Kate Louise Naguit
Employee ID
2
Email
louigiedavid@gmail.com
Phone
09353298343
Role
Finance Executive
Department
Finance
Joining Date
7/18/2025
Employment Type
Part-Time
Availability Status
Available
Languages Spoken
English;Filipino
Address
Sapalibutad Angeles City
Country
Activity
Filters: All time • All activities • All types
Refresh • Expand All • View All
Upcoming & Overdue
No activities to show.
Get started by sending an email, scheduling a task, and more.
No past activity. Past meetings and tasks marked as done show up here.



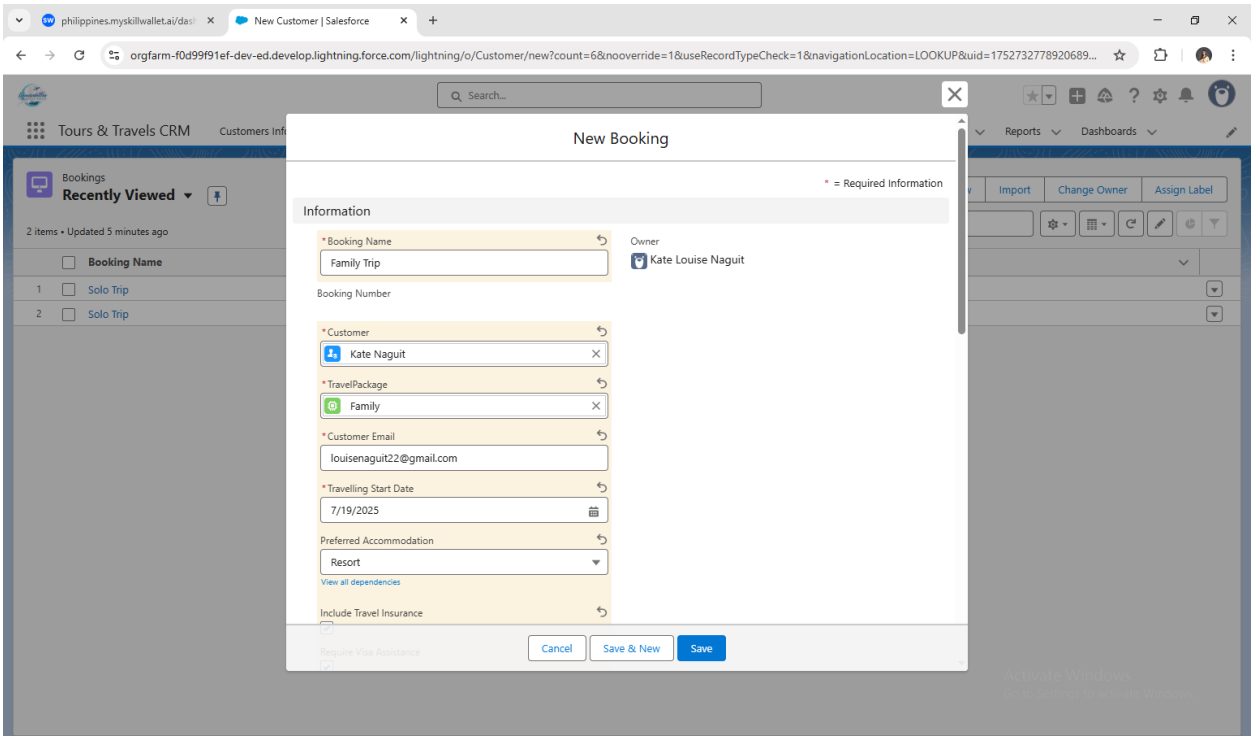
Test Case 2: New Customer Entry

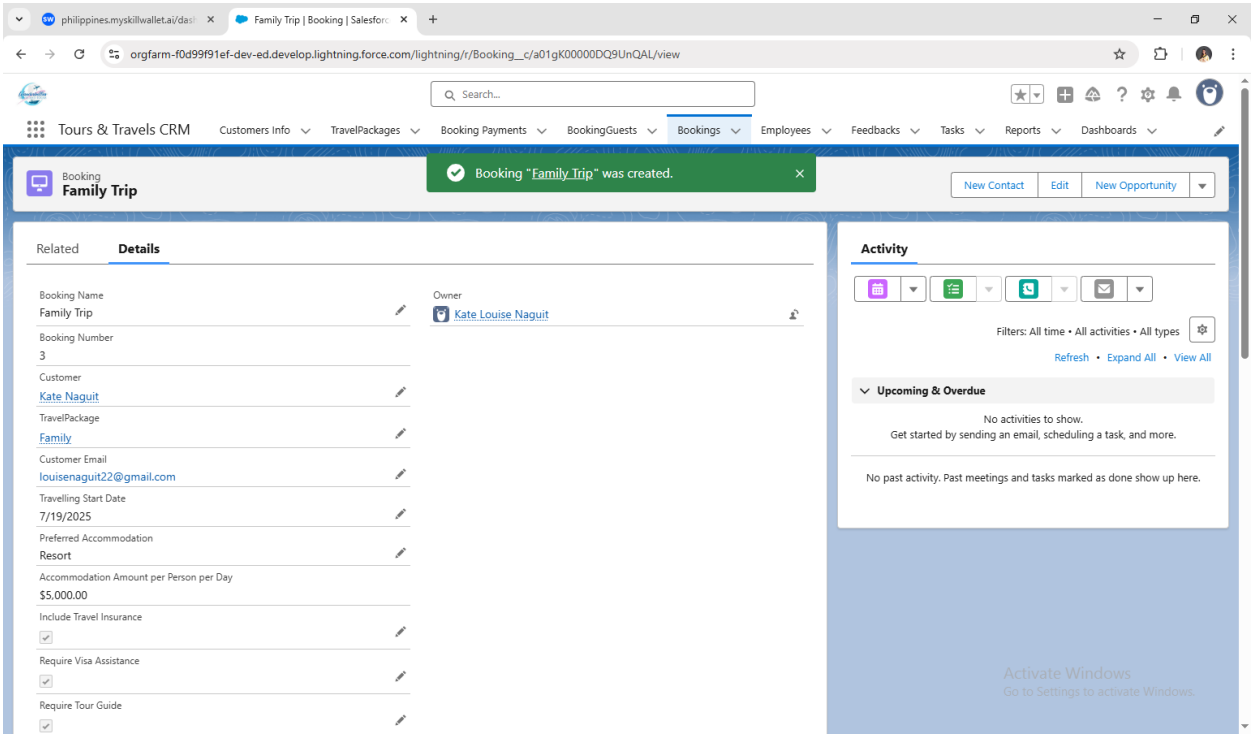


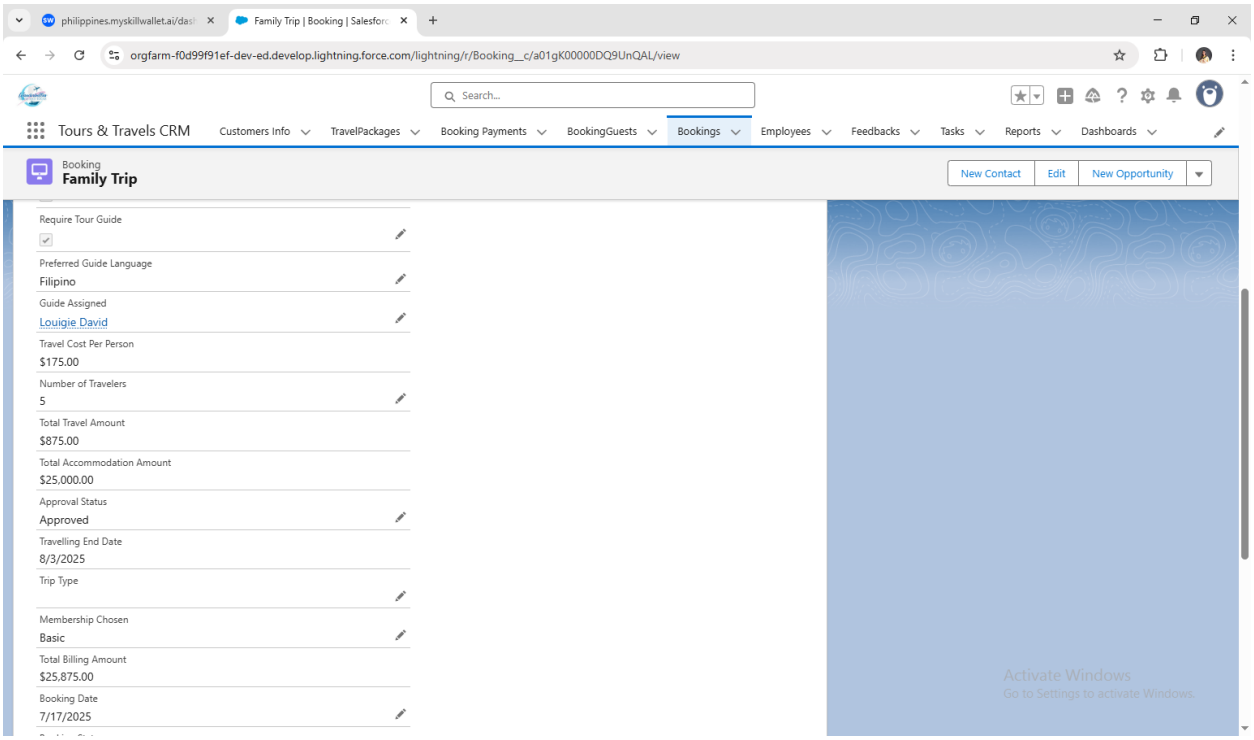


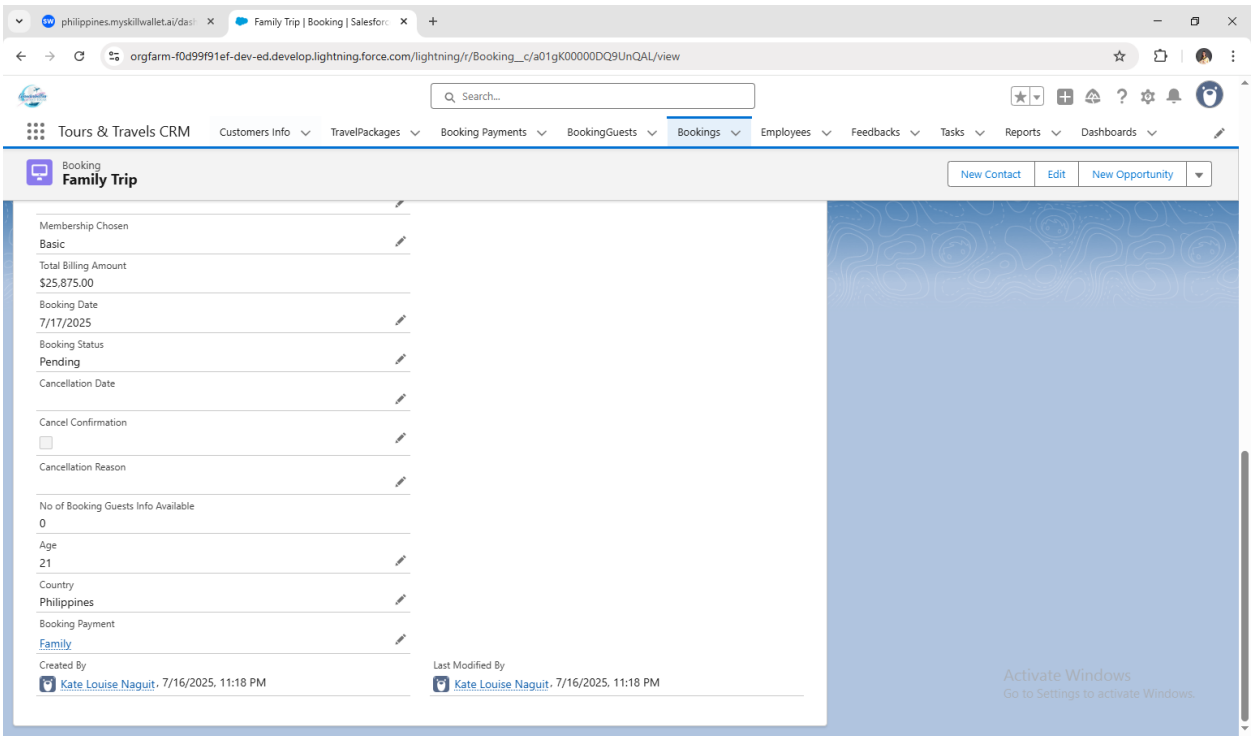


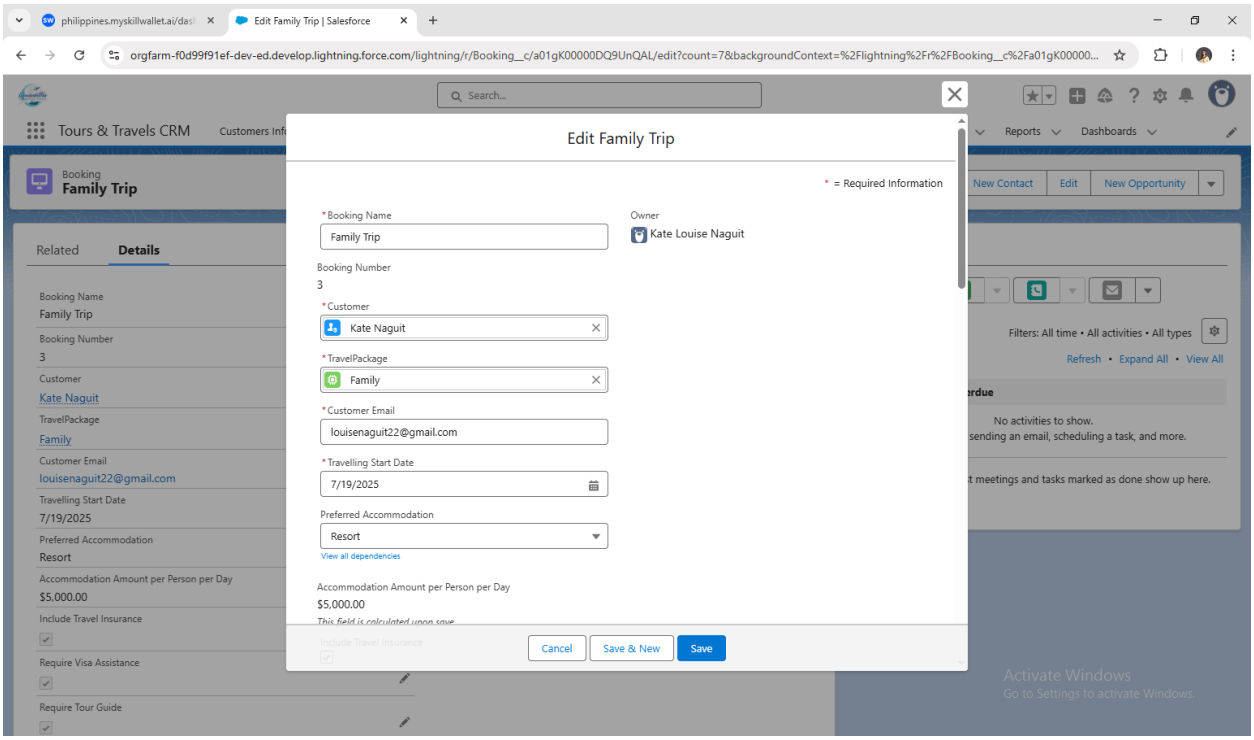
Test Case 3: Create Booking Record

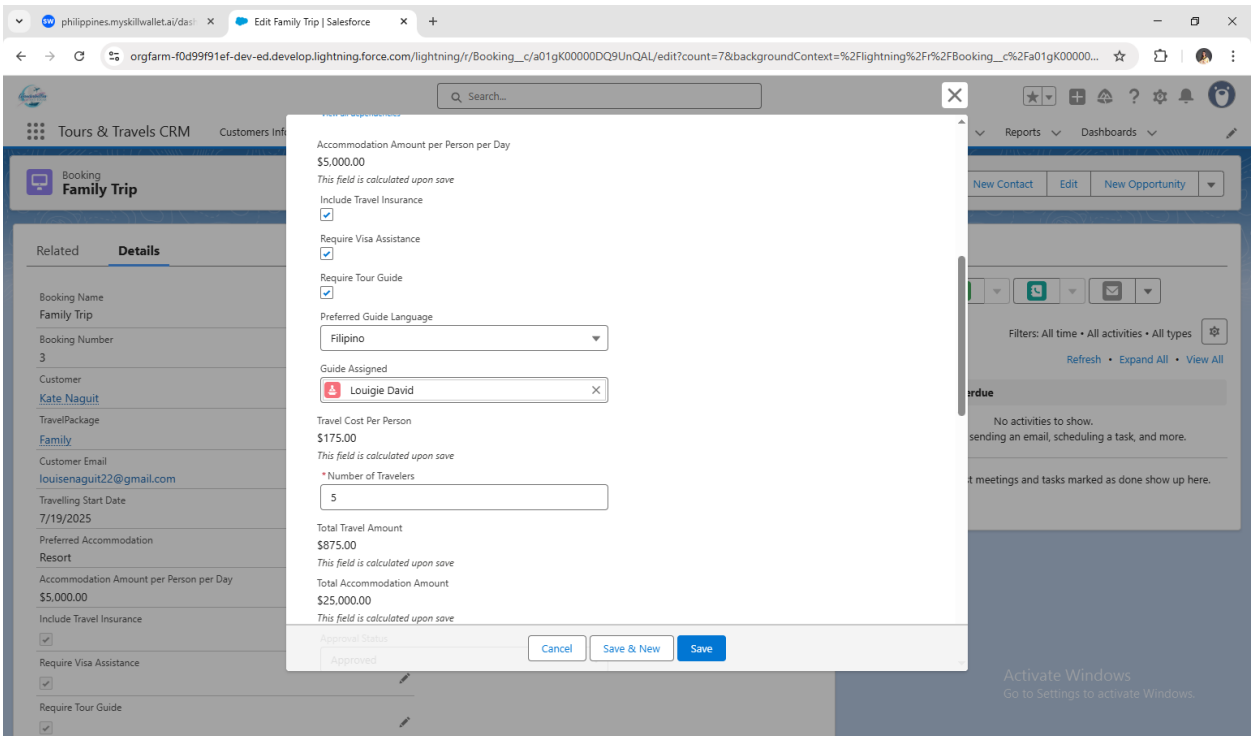

New Customer | Salesforce
Tours & Travels CRM
New Booking
* = Required Information
Information
*Booking Name
Family Trip
Booking Number
Owner
Kate Louise Naguit
*Customer
Kate Naguit
*TravelPackage
Family
*Customer Email
louisenaguit22@gmail.com
*Travelling Start Date
7/19/2025
Preferred Accommodation
Resort
Include Travel Insurance
Cancel
Save & New
Save

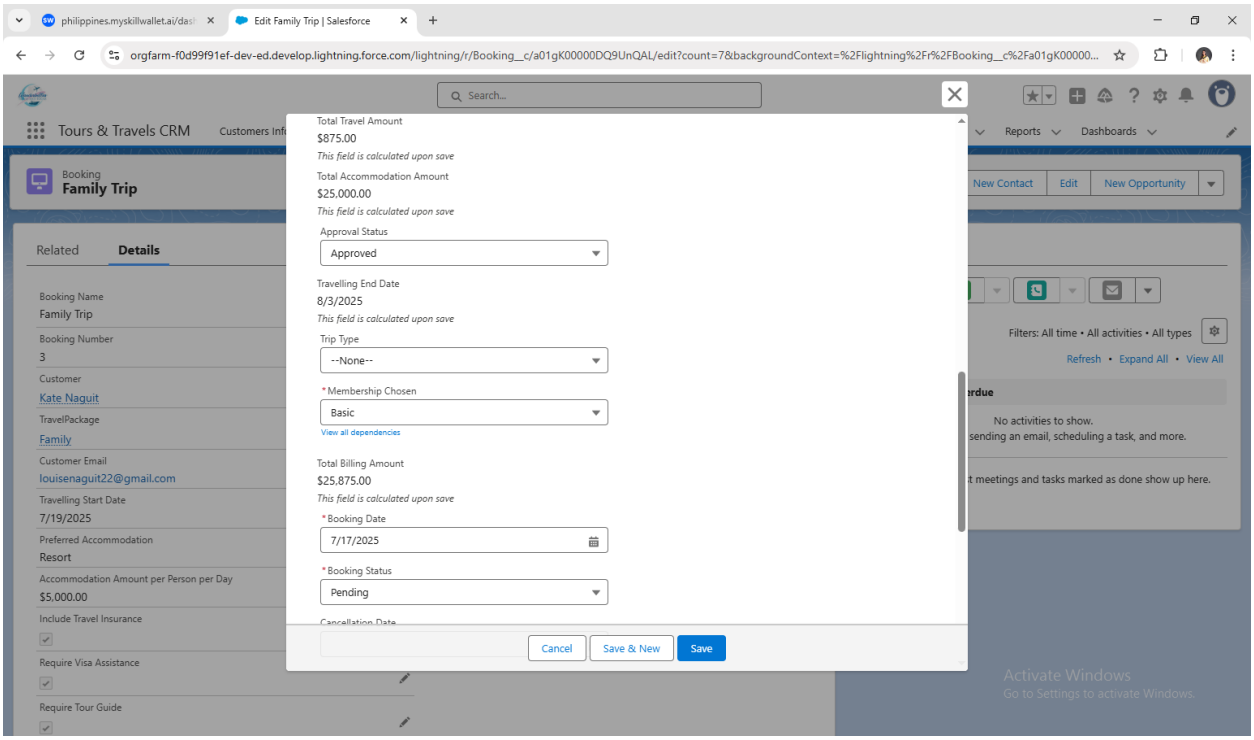

Family Trip | Booking | Salesforce
Tours & Travels CRM
Booking "Family Trip" was created.
Booking
Family Trip
New Contact
Edit
New Opportunity
Related
Details
Booking Name
Family Trip
Booking Number
3
Customer
Kate Naguit
TravelPackage
Family
Customer Email
louisenaguit22@gmail.com
Travelling Start Date
7/19/2025
Preferred Accommodation
Resort
Accommodation Amount per Person per Day
$5,000.00
Include Travel Insurance
Require Visa Assistance
Require Tour Guide
Owner
Kate Louise Naguit
Activity
Filters: All time • All activities • All types
Refresh • Expand All • View All
Upcoming & Overdue
No activities to show.
Get started by sending an email, scheduling a task, and more.
No past activity. Past meetings and tasks marked as done show up here.


Family Trip | Booking | Salesforc
orgfarm-f0d99f91ef-dev-ed.develop.lightning.force.com/lightning/r/Booking__c/a01gK00000DQ9UnQAL/view
Search...
Tours & Travels CRM
Customers Info
TravelPackages
Booking Payments
BookingGuests
Bookings
Employees
Feedbacks
Tasks
Reports
Dashboards
Booking
Family Trip
New Contact
Edit
New Opportunity
Require Tour Guide
Preferred Guide Language
Filipino
Guide Assigned
Louigie David
Travel Cost Per Person
$175.00
Number of Travelers
5
Total Travel Amount
$875.00
Total Accommodation Amount
$25,000.00
Approval Status
Approved
Travelling End Date
8/3/2025
Trip Type
Membership Chosen
Basic
Total Billing Amount
$25,875.00
Booking Date
7/17/2025
Activate Windows
Go to Settings to activate Windows.

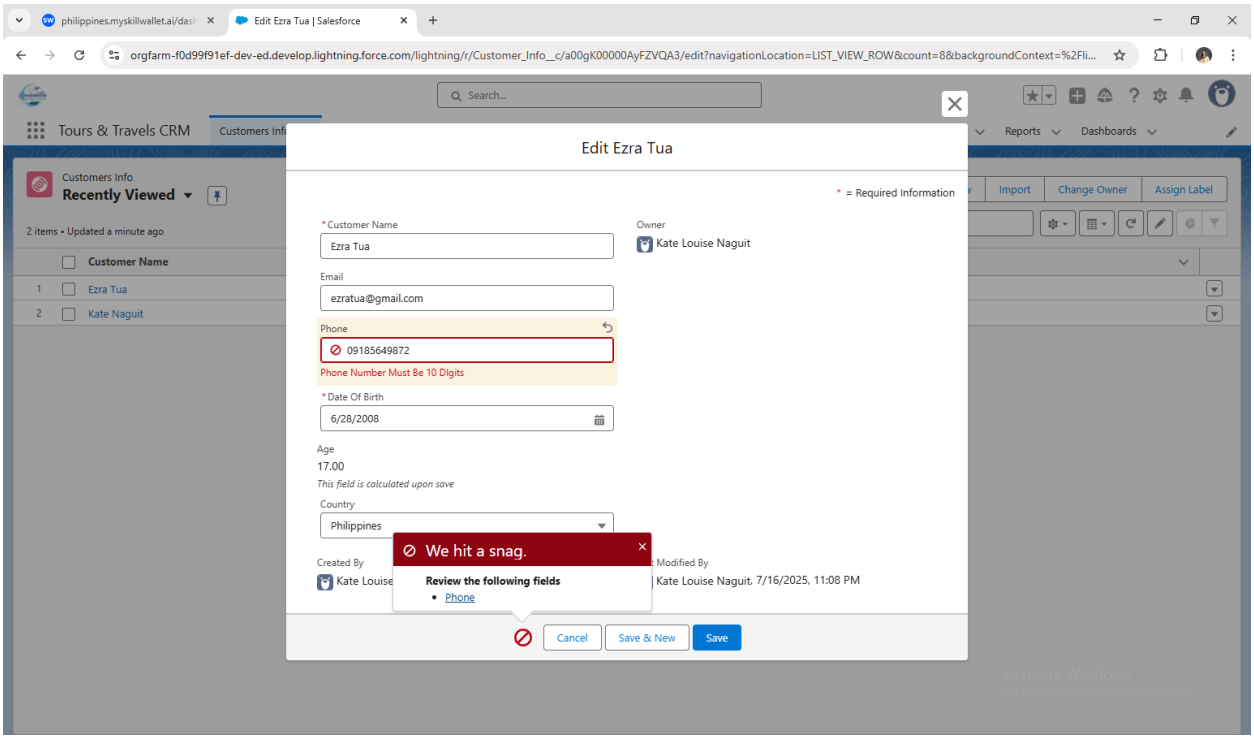

Family Trip | Booking | Salesforc
orgfarm-f0d99f91ef-dev-ed.develop.lightning.force.com/lightning/r/Booking__c/a01gK00000DQ9UnQAL/view
Search...
Tours & Travels CRM
Customers Info
TravelPackages
Booking Payments
BookingGuests
Bookings
Employees
Feedbacks
Tasks
Reports
Dashboards
Booking
Family Trip
New Contact
Edit
New Opportunity
Membership Chosen
Basic
Total Billing Amount
$25,875.00
Booking Date
7/17/2025
Booking Status
Pending
Cancellation Date
Cancel Confirmation
Cancellation Reason
No of Booking Guests Info Available
0
Age
21
Country
Philippines
Booking Payment
Family
Created By
Kate Louise Naguit, 7/16/2025, 11:18 PM
Last Modified By
Kate Louise Naguit, 7/16/2025, 11:18 PM
Activate Windows
Go to Settings to activate Windows.


Edit Family Trip | Salesforce
Tours & Travels CRM
Booking
Family Trip
Edit Family Trip
* = Required Information
*Booking Name
Family Trip
Owner
Kate Louise Naguit
Booking Number
3
*Customer
Kate Naguit
*TravelPackage
Family
*Customer Email
louisenaguit22@gmail.com
*Travelling Start Date
7/19/2025
Preferred Accommodation
Resort
View all dependencies
Accommodation Amount per Person per Day
$5,000.00
Cancel
Save & New
Save

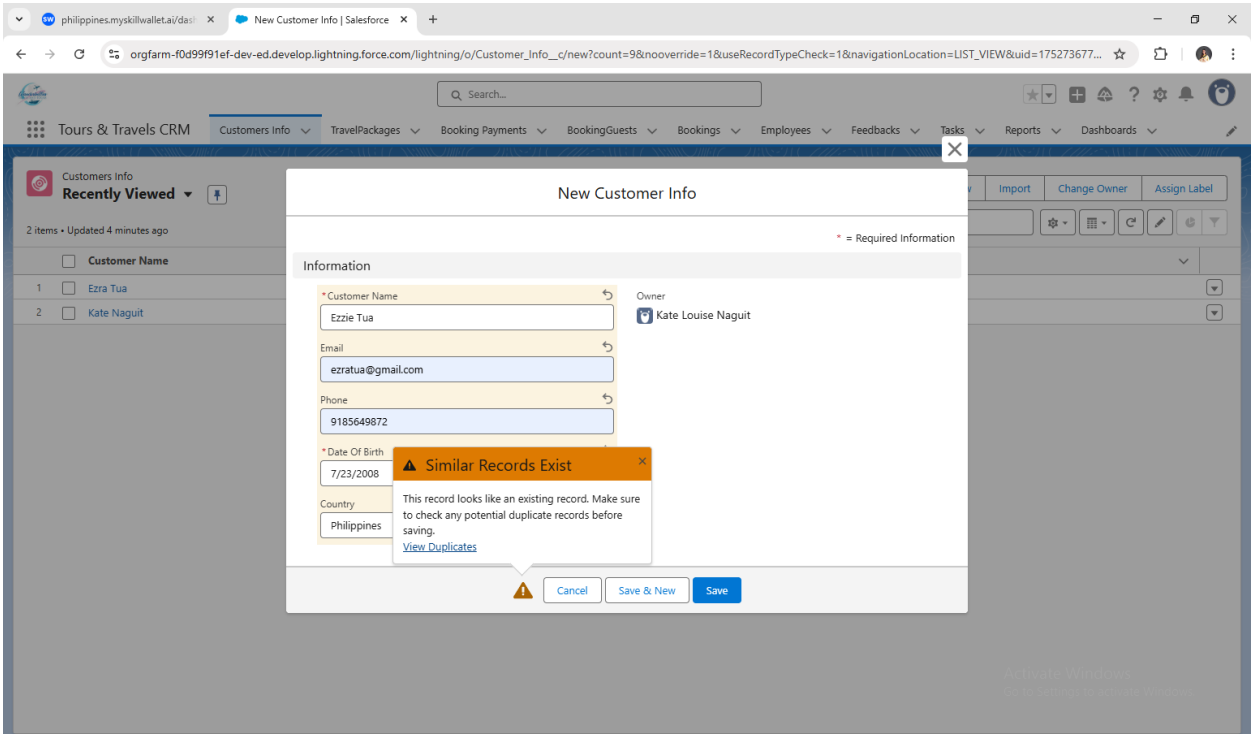

Edit Family Trip | Salesforce
Tours & Travels CRM
Booking
Family Trip
Accommodation Amount per Person per Day
$5,000.00
This field is calculated upon save
Include Travel Insurance
Require Visa Assistance
Require Tour Guide
Preferred Guide Language
Filipino
Guide Assigned
Louigie David
Travel Cost Per Person
$175.00
This field is calculated upon save
*Number of Travelers
5
Total Travel Amount
$875.00
This field is calculated upon save
Total Accommodation Amount
$25,000.00
This field is calculated upon save
Cancel
Save & New
Save



Test Case 4: Update Customer Info



Test Case 5: Prevent Duplicate Entry



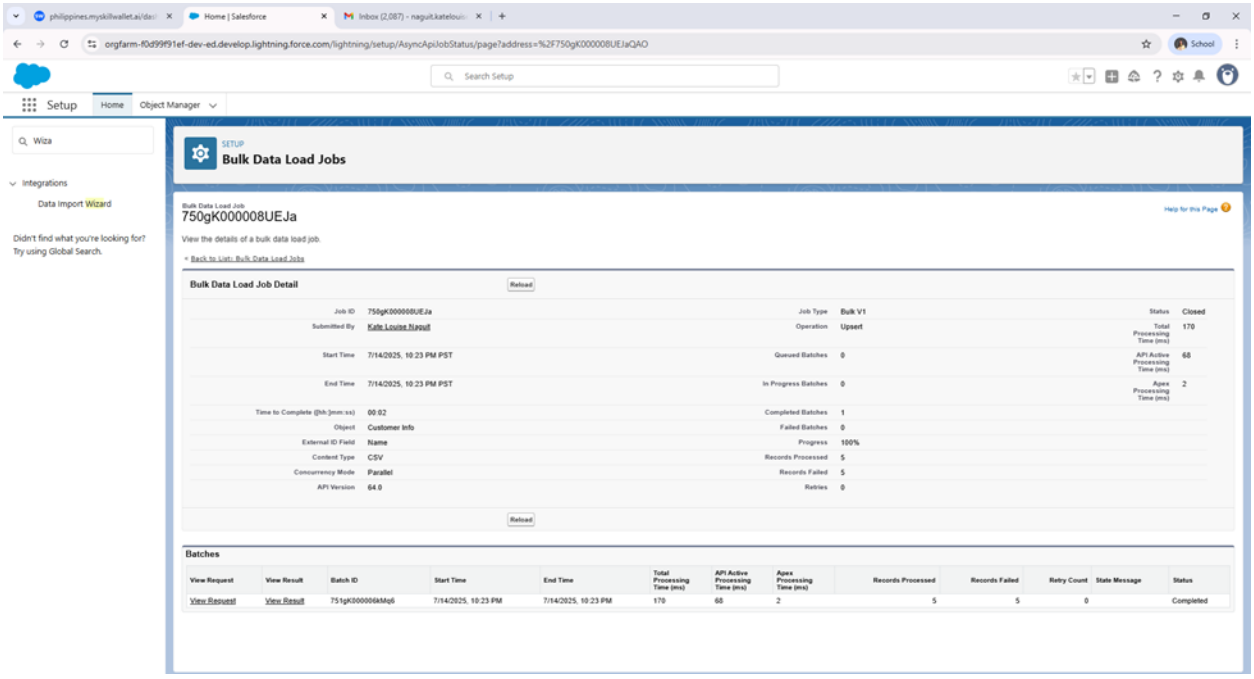
Milestone 29: Data Import Wizard

To set up initial data within the Salesforce environment, I utilized the Data Import Wizard to bring in sample records for the Customer Info, TravelPackage, and Employee objects. Each CSV file included at least 20 entries, with all essential and relevant fields completed to ensure a smooth and successful upload.

I began by signing in as an administrator and searching for Data Import Wizard through the Quick Find box in Setup. For each object starting with Customer Info, I selected the “Add new records” option and uploaded the corresponding CSV file. During the import setup, I carefully reviewed the field mappings. While Salesforce auto-mapped most fields, I double-checked and manually corrected any mismatches to maintain data accuracy.

After verifying the field mappings, I proceeded with the import and monitored its status via the Bulk Data Load Jobs section in Setup. I followed the same steps for importing data into the TravelPackage and Employee objects to keep the process consistent.

Once all the data was imported, I checked the records within each object to confirm that everything had been loaded correctly and displayed as intended. This initial data population provided a dependable dataset for further testing, reporting, and hands-on use within the Tours & Travels CRM platform.



Phase 5: Deployment, Documentation & Maintenance

For this project, the Tours and Travels Salesforce CRM was fully built and configured within a Salesforce Developer Org. The main goal was to learn and demonstrate how to design, customize, and test a complete Salesforce CRM system similar to a real business setup.

Actual deployment to a live production environment was not required or performed, as Developer Edition orgs operate independently and are not connected to production systems. In enterprise scenarios, deployment usually involves sandboxes, production orgs, and tools such as Change Sets or CI/CD pipelines; however, these were beyond the scope of this exercise.

Even though the system remains in a Developer Org, a maintenance plan was outlined to ensure smooth operation. This included regular checks for data accuracy, monitoring of automations, reviewing error logs, and gathering user feedback for continuous improvements.

Comprehensive documentation was also prepared, covering setup details and user guides to ensure everything is clearly recorded. For troubleshooting, general practices were outlined, such as identifying and confirming issues, checking logs and error messages, reviewing recent changes, testing fixes within the Developer Org, and making updates as needed.

This approach ensures the system remains reliable, understandable, and easy to enhance in the future, even without a live production deployment.

Conclusion

The development of the Tours and Travels CRM project in Salesforce followed a complete lifecycle approach, with each of the five project phases contributing to the creation of a functional, scalable, and user-focused CRM system. This experience not only tackled real-world business challenges faced by the travel and tourism industry but also offered a comprehensive opportunity to apply Salesforce development and customization skills in a structured and professional context.

In Phase 1: Requirement Analysis & Planning, business needs were assessed to address issues such as slow booking processes, manual payment tracking, and scattered customer data. The project scope and objectives were clearly defined, and a detailed roadmap was created. A custom data model was designed, along with a role-based security model, to guide the implementation process and ensure alignment with business workflows.

Phase 2: Backend Development and Automation focused on structuring the system through custom objects such as Booking, Booking Payment, Customer Info, Employee, Travel Package, and Feedback. Relationships among these objects were established to mirror actual business operations. Automation tools like validation rules, flows, workflow rules, approval processes, and Apex triggers were implemented to ensure data accuracy and process efficiency. Asynchronous processing using batch, queueable, and schedulable Apex classes further enhanced system performance.

In Phase 3: User Interface and Experience, the system was designed to offer intuitive navigation and accessibility. Dynamic Lightning App Pages were developed for different user roles, making data management streamlined and efficient. Reports and dashboards were configured to support decision-making based on booking trends, payment statuses, and customer feedback. User access was controlled using profiles, permission sets, role hierarchies, and org-wide defaults, while branding elements and email templates created a more polished experience.

Phase 4: Testing and Deployment Preparation ensured the system's readiness and reliability. The Data Import Wizard was used to upload sample records for essential objects. Test classes were written for backend logic, and detailed manual test cases were created for each key feature. Additional configurations such as duplicate rules, field history tracking, and sharing rules were added to maintain data quality and visibility. These efforts helped verify that the system would perform reliably under different conditions.

Finally, in Phase 5: Deployment, Documentation & Maintenance, the CRM system was finalized within a Salesforce Developer Org. Although no production deployment was required, deployment practices were studied, and a maintenance plan was established. System documentation and a demo video were prepared to support long-term usability and understanding of the CRM. These resources serve as clear references for system setup, navigation, and key processes, ensuring that future users or developers can maintain and enhance the system effectively.

Through the successful completion of all five phases, this project provided a valuable, hands-on experience in building and managing a Salesforce CRM application from start to finish. Key lessons included the importance of understanding business requirements, the power of automation, the need for user-friendly design, and the role of testing and documentation in maintaining quality. Most importantly, it highlighted how CRM solutions when well-designed and properly maintained can streamline operations, enhance customer service, and provide actionable insights for growth.

This experience not only strengthened technical expertise in Salesforce but also developed critical thinking, project management, and system design skills serving as strong preparation for future roles in CRM development and enterprise system customization.