

# Tours and Travel Salesforce Documentation

The Tours & Travels CRM project was initiated to address critical operational challenges in the travel industry, including inconsistent customer experiences, inefficient booking processes, manual feedback tracking, and a lack of centralized employee coordination. Users, such as customers, agents, guides, and finance officers, required a system that streamlined communication, automated workflows, and provided role-specific access and reporting. Specific needs addressed included real-time booking updates, automated follow-ups, dynamic pricing, and secure data management.

## Defining Project Scope and Objectives

### Scope:

- Develop a multi-role CRM for global travel agencies.
- Support full booking lifecycle: inquiry to feedback.
- Include object-level data design, workflows, validations, and UI elements.
- Implement security with Profiles, Roles, Sharing Rules, and FLS.

### Objectives:

- Enhance operational efficiency by automating workflows.
- Improve customer satisfaction through faster communication.
- Enable real-time monitoring with analytics and dashboards.
- Ensure data accuracy and security through validation and controlled access.

## Designing the Data Model and Security Model

### Data Model Highlights:

- Custom objects: Customer\_info\_c, Booking\_c, TravelPackage\_c, Employee\_c, Feedback\_c, BookingPayment\_c, BookingGuest\_c
- Relationships: Lookup, Master-Detail, and formula fields to link entities logically.
- Roll-up summaries to calculate guest count.

## **Security Model:**

**Profiles:** Admin, Travel Agent, Finance Officer, Tour Guide, Customer Service.

- Admin Profile has full access to all features.
- Travel Agent Profile can create and manage bookings.
- Finance Officer Profile manages payment-related data.
- Tour Guide Profile has read-only access to travel assignments.
- Customer Service Profile can view and manage customer feedback.

**Role Hierarchy:** CEO > Manager > Agent > Guide.

- The CEO has access to everything.
- Managers can see data from the agents and guides they supervise.
- Agents handle customer bookings directly.
- Guides only see what's related to their assignments.

**Permission Sets:** These provide additional access when needed without modifying the user's main profile. For instance, a Travel Agent Manager might get extra reporting permissions via a permission set.

**Sharing Rules:** These automatically share specific records. For example, customer details are auto-shared with guides assigned to those customers, but only for viewing, not editing.

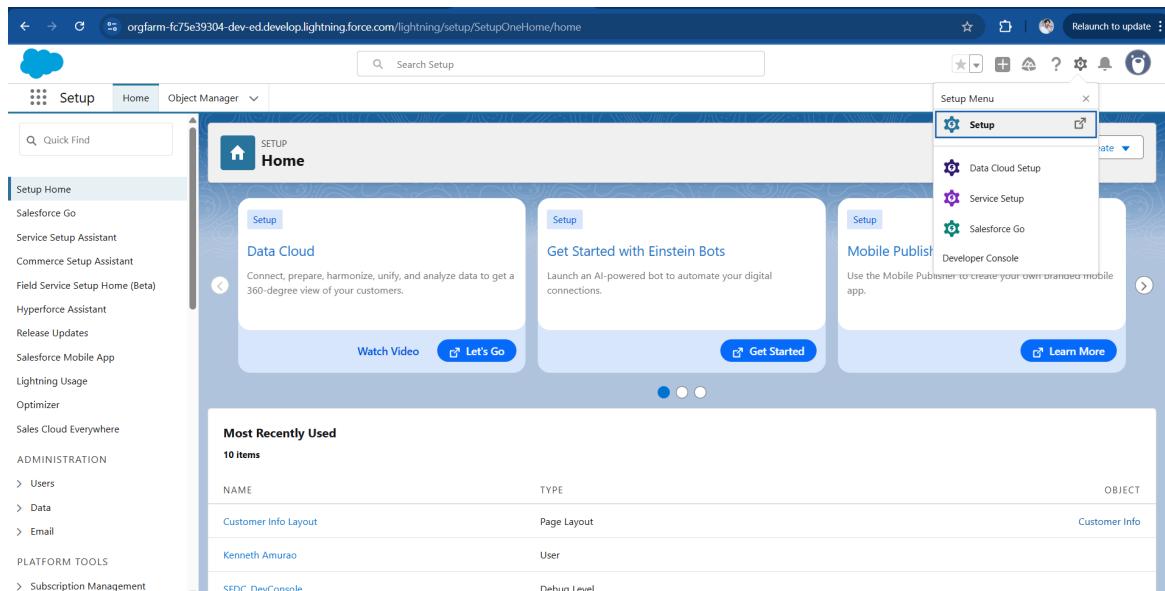
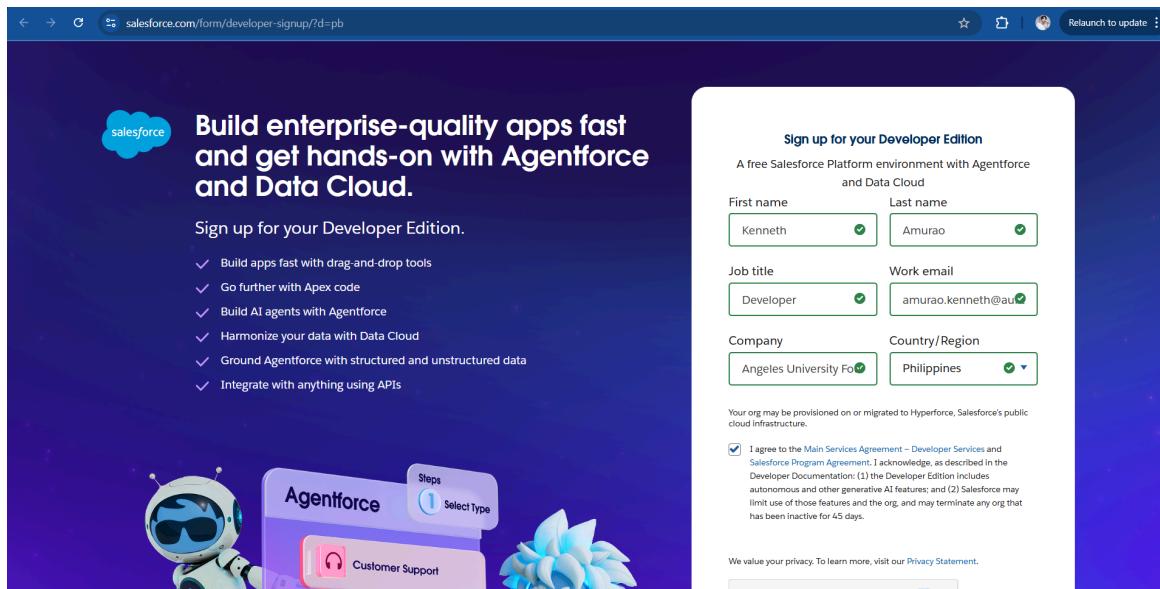
## Project Roadmap and Milestones

- **Milestone 1–6:** Object, Field, and UI Setup
- **Milestone 7–10:** Automation (Approval Process, Flows, Workflow, Process Builder)
  - **Note:** Before starting Milestone 7, start the following pre-requisite Milestone activity:
    - Start with **Milestone 13 (Creating Lightning App "Tours & Travels CRM")**
    - Then proceed to **Milestone 24 (Roles & Role Hierarchy)** before Profile to prevent conflict for further errors,
    - After that, I now proceed to **Milestone 23 (Profile)**
    - Next to profile **Milestone 16 (User)**
    - Once all is complete, that's the time I start **Milestone 7**
  -
- **Milestone 11–12:** Apex Development (Trigger, Future, Queueable, Batch Apex)
- **Milestone 13–20:** App Manager, UI Enhancement, LWC, Dynamic Forms
- **Milestone 21–26:** Field History, Matching Rules, Profiles, Roles, Permissions
- **Milestone 27–29:** Test Classes, Test Cases, Data Import

# Phase 2: Salesforce Development - Backend & Configurations

## Setup Environment & DevOps Workflow

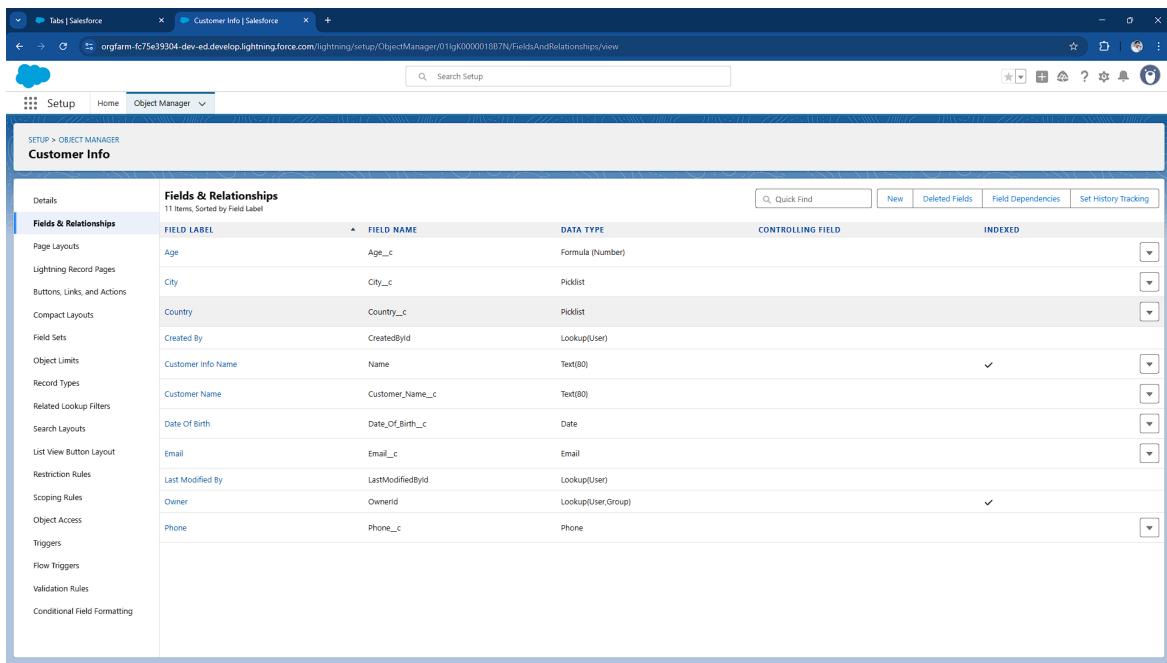
- Developer Org was created via Salesforce Developer Signup.
- DevOps followed iterative testing and validations using Developer Console and Change Sets for deployment



## Customizations Implemented

- **Seven Custom Objects & Fields:** Customer\_\_c, Booking\_\_c, BookingGuest\_\_c, TravelPackage\_\_c, BookingPayment\_\_c, Employee\_\_c, and Feedback\_\_c. Over 60 fields were configured across these objects, including lookups, picklists, formula fields, and roll-up summaries.

### Customer info (Object & Fields)



The screenshot shows the Salesforce Object Manager interface for the 'Customer Info' object. The left sidebar lists various configuration tabs: Details, Fields & Relationships (selected), 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, and Conditional Field Formatting. The main content area displays the 'Fields & Relationships' section, which lists 11 items sorted by Field Label. The table columns include FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The listed fields are: Age (Age\_\_c, Formula (Number)), City (City\_\_c, Picklist), Country (Country\_\_c, Picklist), Created By (CreatedBy, Lookup(User)), Customer Info Name (Name, Text(80)), Customer Name (Customer\_Name\_\_c, Text(80)), Date Of Birth (Date\_of\_Birth\_\_c, Date), Email (Email\_\_c, Email), Last Modified By (LastModifiedBy, Lookup(User)), Owner (OwnerId, Lookup(User,Group)), and Phone (Phone\_\_c, Phone). The 'Customer Info' tab is selected in the top navigation bar.

## BookingGuest (Object & Fields)

The screenshot shows the Salesforce Object Manager interface for the 'BookingGuest' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, and Field Sets. The main area is titled 'Fields & Relationships' and displays 14 items, sorted by Field Label. The table includes columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed status.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Age	Age__c	Number(18, 0)		
Age Category	Age_Category__c	Formula (Text)		
BookingGuest Name	BookingGuest_Name__c	Text(80)		
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)		

This screenshot shows the same Salesforce Object Manager interface for the 'BookingGuest' object, but with a different set of fields listed. The table below shows the updated field definitions.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
City	City__c	Picklist		
Country	Country__c	Picklist		
Created By	CreatedById	Lookup(User)		
Gender	Gender__c	Picklist		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
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		

# Booking (Object & Fields)

Booking | Salesforce

orgfarm-fc75e39304-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK0000018Byb/FieldsAndRelationships/view

Setup Home Object Manager

Search Setup

SETUP > OBJECT MANAGER Booking

Fields & Relationships		Fields & Relationships			
30 items, Sorted by Field Label		30 items, Sorted by Field Label			
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	
Accommodation Amount per Person per Day	Accommodation_Amount_per_Person_per_Day__c	Formula (Currency)			
Approval Status	Approval_Status__c	Picklist			
Booking Date	Booking_Date__c	Date			
Booking Name	Name	Auto Number			
Booking Number	Booking_Number__c	Auto Number			
Booking Status	Booking_Status__c	Picklist			
Cancel Confirmation	Cancel_Confirmation__c	Checkbox			
Cancellation Date	Cancellation_Date__c	Date			
Cancellation Reason	Cancellation_Reason__c	Text Area(255)			
Created By	CreatedById	Lookup(User)			
Customer	Customer__c	Lookup(Customer)	✓		
Customer Email	Customer_Email__c	Email			
Include Travel Insurance	Include_Travel_Insurance__c	Checkbox			
Last Modified By	LastModifiedById	Lookup(User)			
Membership Chosen (Req)	Membership_Chosen_Req__c	Picklist			

Booking | Salesforce

orgfarm-fc75e39304-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK0000018Byb/FieldsAndRelationships/view

Setup Home Object Manager

Search Setup

SETUP > OBJECT MANAGER Booking

Fields & Relationships		Fields & Relationships			
30 items, Sorted by Field Label		30 items, Sorted by Field Label			
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	
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			
TravelPackage	TravelPackage__c	Lookup(TravelPackage)	✓		
Trip Type	Trip_Type__c	Picklist			

# Travel Package Object (Object & Fields)

Screenshot of the Salesforce Object Manager showing the fields and relationships for the TravelPackage object.

**Fields & Relationships** (23 items, Sorted by Field Label)

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Additional Notes	Additional_Notes_c	Text Area(255)		
Availability Status	Availability_Status_c	Picklist		
Average Rating	Average_Rating_c	Number(18, 0)		
Country	Country_c	Picklist		
Created By	CreatedById	Lookup(User)		
Duration in Days	Duration_in_Days_c	Number(18, 0)		
Guide Included	Guide_Included_c	Checkbox		
Guide LanguagesPlaces Covered	Guide_LanguagesPlaces_Covered_c	Picklist (Multi-Select)		
Insurance Included	Insurance_Included_c	Checkbox		
Last Modified By	LastModifiedById	Lookup(User)		
Maximum Group Size	Maximum_Group_Size_c	Number(18, 0)		
Meals Included	Meals_Included_c	Picklist		
Membership	Membership_c	Picklist (Multi-Select)		
Owner	OwnerId	Lookup(User/Group)		
Package Type	Package_Type_c	Picklist (Multi-Select)		

Screenshot of the Salesforce Object Manager showing the fields and relationships for the TravelPackage object.

**Fields & Relationships** (23 items, Sorted by Field Label)

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Insurance Included	Insurance_Included_c	Checkbox		
Last Modified By	LastModifiedById	Lookup(User)		
Maximum Group Size	Maximum_Group_Size_c	Number(18, 0)		
Meals Included	Meals_Included_c	Picklist		
Membership	Membership_c	Picklist (Multi-Select)		
Owner	OwnerId	Lookup(User/Group)		
Package Type	Package_Type_c	Picklist (Multi-Select)		
Places Covered	Places_Covered_c	Text Area(255)		
Preferred Guide Language	Preferred_Guide_Language_c	Picklist		
Price Per Person	Price_Per_Person_c	Currency(18, 0)		
Region	Region_c	Picklist		
Transportation Modes	Transportation_Modes_c	Picklist (Multi-Select)		
Travel Package Name	Travel_Package_Name_c	Text(80)		
TravelPackage Name	Name	Text(80)		
Visa Assistance	Visa_Assistance_c	Checkbox		

## Booking Payment (Object & Fields)

The screenshot shows the Salesforce Object Manager interface for the 'Booking Payment' object. The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, etc. The main content area displays a table titled 'Fields & Relationships' with 15 items. The columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data includes:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Booking	Booking__c	Lookup(Booking)		✓
Booking Payment Name		Name	Auto Number	✓
Created By	CreatedById	Lookup(User)		✓
Customer Name	Customer_Name__c	Formula (Text)		✓
Last Modified By	LastModifiedById	Lookup(User)		✓
Notes	Notes__c	Text Area(255)		✓
Owner	OwnerId	Lookup(User,Group)		✓
Payment Date	Payment_Date__c	Date		✓
Payment Method	Payment_Method__c	Picklist		✓
Payment Mode Details	Payment_Mode_Details__c	Text Area(255)		✓
Payment Number	Payment_Number__c	Auto Number		✓
Payment Receipt Sent	Payment_Receipt_Sent__c	Checkbox		✓
Payment Reference Number	Payment_Reference_Number__c	Text(50)		✓
Payment Status	Payment_Status__c	Picklist		✓
Total Bill Amount	Total_Bill_Amount__c	Formula (Currency)		✓

## Feedback Object

The screenshot shows the Salesforce Object Manager interface for the 'Feedback' object. The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, etc. The main content area displays a table titled 'Fields & Relationships' with 15 items. The columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data includes:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Accommodation Rating	Accommodation_Rating__c	Picklist		✓
Booking	Booking__c	Lookup(Booking)		✓
Created By	CreatedById	Lookup(User)		✓
Customer	Customer__c	Lookup(Customer)		✓
Feedback Date	Feedback_Date__c	Date		✓
Feedback Name		Name	Auto Number	✓
Feedback No	Feedback_No__c	Auto Number		✓
Last Modified By	LastModifiedById	Lookup(User)		✓
Overall Rating	Overall_Rating__c	Picklist		✓
Owner	OwnerId	Lookup(User,Group)		✓
Package Satisfaction	Package_Satisfaction__c	Picklist		✓
Response Status	Response_Status__c	Picklist		✓
Suggestions / Comments	Suggestions_Comments__c	Long Text Area(32768)		✓
Tour Guide Rating	Tour_Guide_Rating__c	Picklist		✓
Transport Rating	Transport_Rating__c	Picklist		✓

- **Validation Rules:** Ensured data accuracy (e.g., 10-digit phone, email regex).
  - Phone field must be 10 digits.(Customer Info, BookingGuest)

The screenshot shows the Salesforce Object Manager for the 'Customer Info' object. On the left, there's a sidebar with various tabs like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Customer Info Validation Rule' and shows a single validation rule named 'Phone\_Number\_Must\_10\_Digits'. The rule details are as follows:

Field	Value
Rule Name	Phone_Number_Must_10_Digits
Error Condition Formula	LEN(Phone__c) <> 10
Error Message	Phone Number Must Be 10 Digits
Description	
Created By	Kenneth Amurao 7/9/2025, 7:08 AM
Active	<input checked="" type="checkbox"/>
Error Location	Phone
Modified By	Kenneth Amurao 7/9/2025, 7:08 AM

- The email must match the standard email format.

This screenshot is similar to the previous one, showing the 'Customer Info Validation Rule' page. The validation rule is named 'Email\_Valid\_Address' and its details are:

Field	Value
Rule Name	Email_Valid_Address
Error Condition Formula	NOT(REGEX(Email__c, "[a-zA-Z0-9_.]+@[a-zA-Z][a-zA-Z].[a-zA-Z]{2,5}"))
Error Message	Please Enter Valid Email Address
Description	
Created By	Kenneth Amurao 7/9/2025, 7:10 AM
Active	<input checked="" type="checkbox"/>
Error Location	Email
Modified By	Kenneth Amurao 7/9/2025, 7:10 AM

- Date of Birth must not be in the future. (Customer Info)

This screenshot shows the 'Customer Info Validation Rule' page again. The validation rule is named 'Prevent\_Future\_DOB' and its details are:

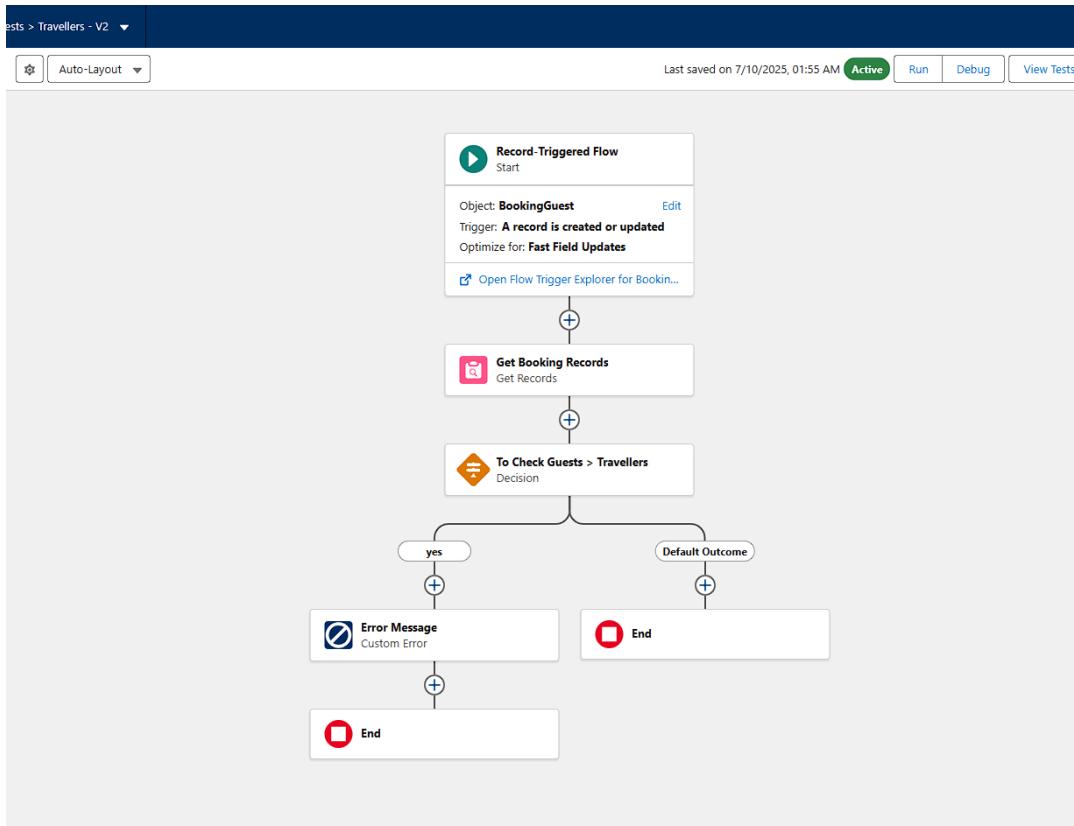
Field	Value
Rule Name	Prevent_Future_DOB
Error Condition Formula	DateOfBirth__c > TODAY()
Error Message	Date of Birth cannot be in the future
Description	
Created By	Kenneth Amurao 7/9/2025, 7:11 AM
Active	<input checked="" type="checkbox"/>
Error Location	Date Of Birth
Modified By	Kenneth Amurao 7/9/2025, 7:11 AM

- Mandatory Email and Language Spoken( Employee)

Validation Rules 2 Items, Sorted by Rule Name				
RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Email_is_mandatory	Email	You Must Enter the Email	✓	Kenneth Amurao, 7/9/2025, 7:15 AM
Language_Must_be_Selected	Languages Spoken	You Must Select Language	✓	Kenneth Amurao, 7/9/2025, 7:16 AM

## ● Automation Tools:

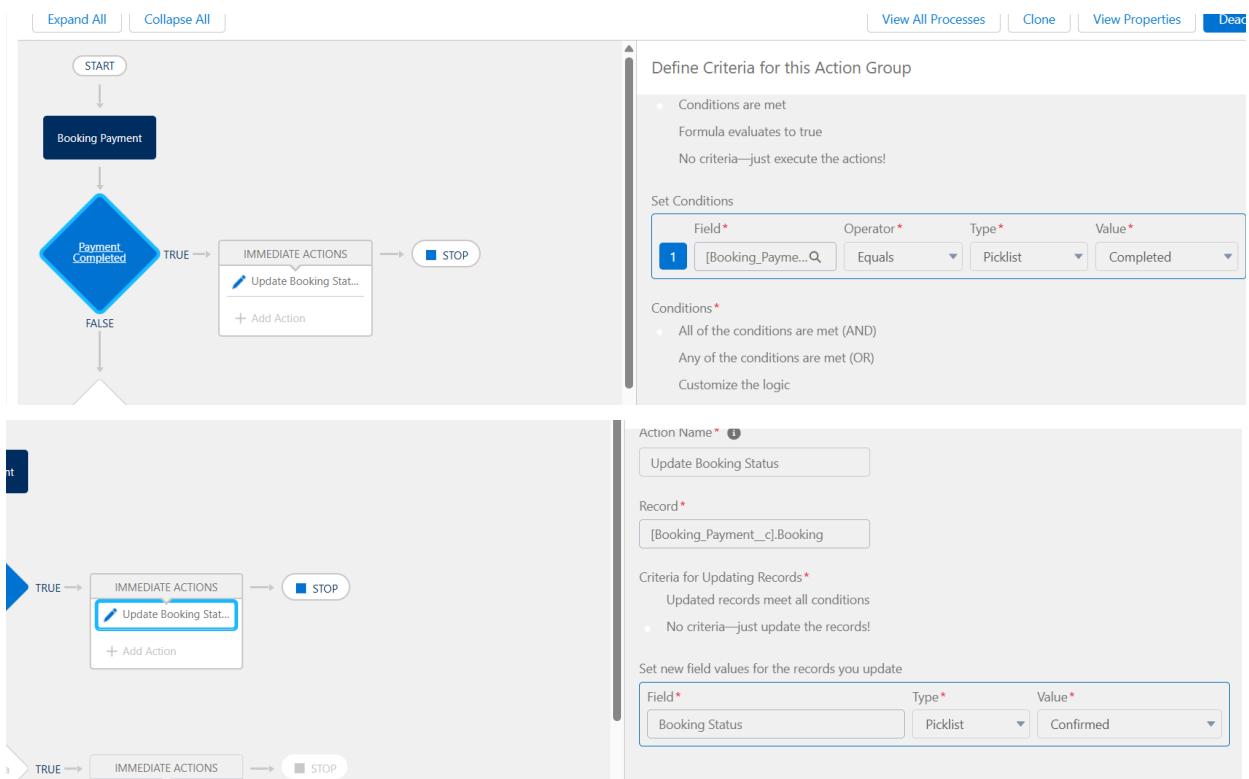
- **Flows:** A flow was built to prevent adding more guests than travelers in a booking. Another flow handled error messages for violations.



- **Workflow Rule:** Automatically assigned a follow-up Task to the Travel Agent three days after the trip's end.

The screenshot shows the 'Workflow Rule Detail' page for a rule named 'Follow-up Task After Booking Completion'. The rule is active and triggered by the 'Booking' object when a record is created or updated. It was created by Kenneth Amuro on 7/9/2025 at 10:59 AM and modified by him on 7/9/2025 at 11:02 AM. The rule has no immediate workflow actions defined.

- **Process Builder:** Automatically updated Booking Status to "Confirmed" when payment status became "Completed".



- **Approval Process:** Canceled bookings required manager approval, triggering approval emails and status updates based on the outcome.

Approval Processes  
Booking: Booking Cancellation Approval  
[Back to Approval Process List](#)

Process Definition Detail	
Process Name	Booking Cancellation Approval
Unique Name	<b>Booking_Cancellation_Approval</b>
Description	Process to approve booking cancellations
Entry Criteria	(Booking: Booking Status EQUALS Cancelled) AND (Booking: Cancel Confirmation EQUALS True)
Record Editability	Administrator OR Current Approver
Approval Assignment Email Template	<b>Booking Cancellation Approval Notification</b>
Initial Submitters	Booking Owner, <a href="#">Travel Agent Manager</a>
Created By	Kenneth Amuro, 7/9/2025, 9:59 AM
Modified By	Kenneth Amuro, 7/9/2025,
<input checked="" type="checkbox"/> Active <input checked="" type="checkbox"/> Next Automated Approver Determined By Manager of Record Submit <input type="checkbox"/> Allow Submitters to Recall Approval Requests	

Initial Submission Actions <small>[1]</small>	
<a href="#">Add Existing</a> <a href="#">Add New ▾</a>	
Action	Type
<a href="#">Edit</a>   <a href="#">Remove</a>	Record Lock
<a href="#">Edit</a>   <a href="#">Remove</a>	Field Update
	<a href="#">Booking status</a>

Approval Steps <small>[1]</small>			
Action	Step Number	Name	Description
<a href="#">Show Actions</a>   <a href="#">Edit</a>	1	Travel Agent Manager Approval	

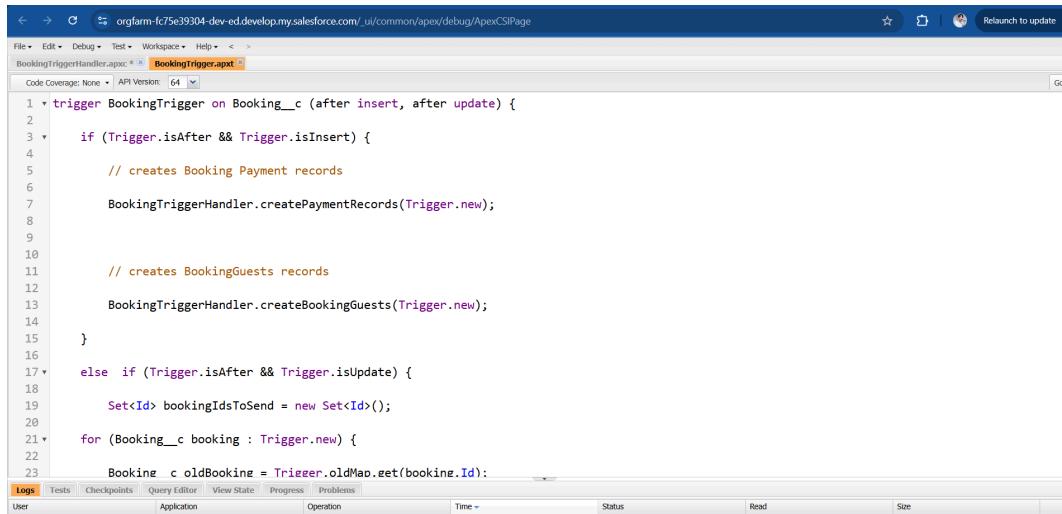
Final Approval Actions <small>[1]</small>	
<a href="#">Add Existing</a> <a href="#">Add New ▾</a>	
Action	Type
<a href="#">Edit</a>	Record Lock
<a href="#">Edit</a>   <a href="#">Remove</a>	Field Update
<a href="#">Edit</a>   <a href="#">Remove</a>	Email Alert
	<a href="#">Approval Status Update</a>
	<a href="#">Send Email to Booking Customer</a>

Final Rejection Actions <small>[1]</small>	
<a href="#">Add Existing</a> <a href="#">Add New ▾</a>	
Action	Type
<a href="#">Edit</a>	Record Lock
<a href="#">Edit</a>   <a href="#">Remove</a>	Email Alert
<a href="#">Edit</a>   <a href="#">Remove</a>	Field Update
<a href="#">Edit</a>   <a href="#">Remove</a>	Field Update
	<a href="#">Unlock the record for editing</a>
	<a href="#">Send Approval Rejection Email to the User</a>
	<a href="#">Booking Status to Confirmed</a>
	<a href="#">Approval Status to Rejected</a>

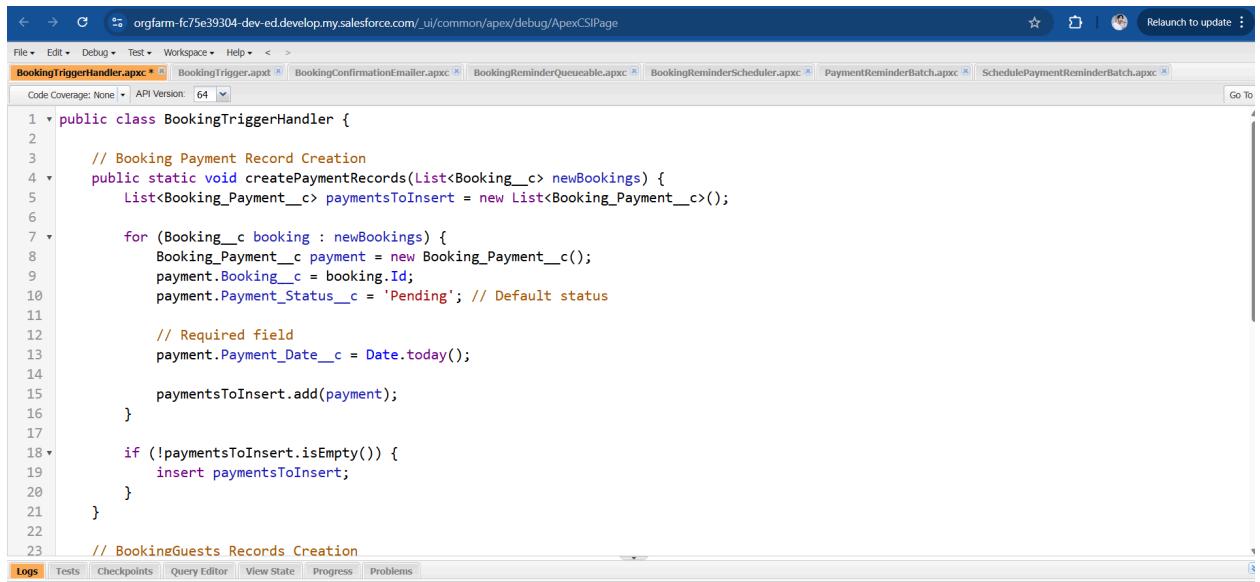
## Apex Development

- **Trigger:** A trigger on the Booking\_\_c object created BookingPayment\_\_c and BookingGuest\_\_c records automatically when a booking was inserted.



```
trigger BookingTrigger on Booking__c (after insert, after update) {
    if (Trigger.isAfter && Trigger.isInsert) {
        // creates Booking Payment records
        BookingTriggerHandler.createPaymentRecords(Trigger.new);
    }
    else if (Trigger.isAfter && Trigger.isUpdate) {
        Set<Id> bookingIdsToSend = new Set<Id>();
        for (Booking__c booking : Trigger.new) {
            Booking_c oldBooking = Trigger.oldMap.get(booking.Id);
            Booking_c newBooking = Trigger.newMap.get(booking.Id);
            if (oldBooking != null && newBooking != null && oldBooking.Status__c != newBooking.Status__c) {
                bookingIdsToSend.add(booking.Id);
            }
        }
        if (!bookingIdsToSend.isEmpty()) {
            BookingTriggerHandler.createBookingGuests(bookingIdsToSend);
        }
    }
}
```

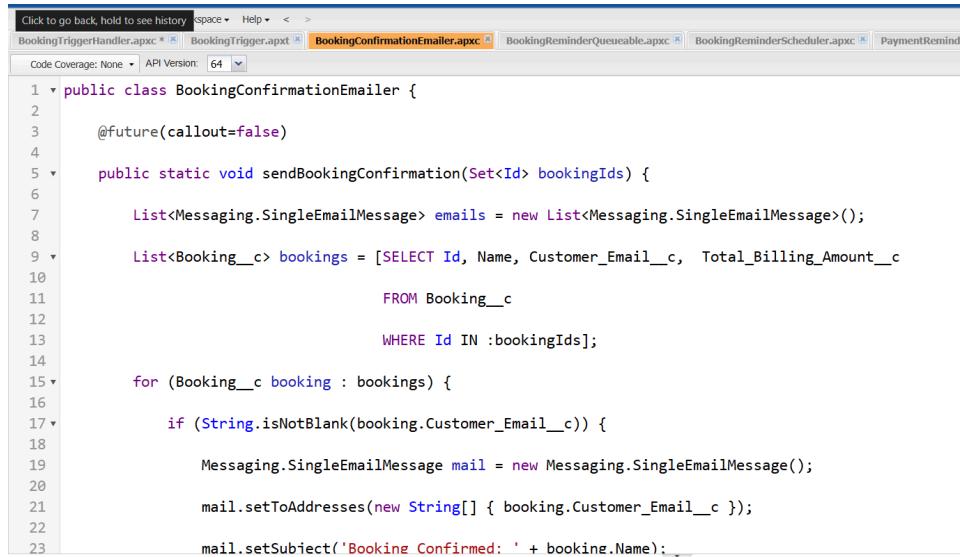
- **Apex Class:** The logic was abstracted into BookingTriggerHandler class for modularity and reusability.



```
public class BookingTriggerHandler {
    // Booking Payment Record Creation
    public static void createPaymentRecords(List<Booking__c> newBookings) {
        List<Booking_Payment__c> paymentsToInsert = new List<Booking_Payment__c>();
        for (Booking__c booking : newBookings) {
            Booking_Payment__c payment = new Booking_Payment__c();
            payment.Booking__c = booking.Id;
            payment.Payment_Status__c = 'Pending'; // Default status
            payment.Payment_Date__c = Date.today();
            paymentsToInsert.add(payment);
        }
        if (!paymentsToInsert.isEmpty()) {
            insert paymentsToInsert;
        }
    }
    // BookingGuests Records Creation
}
```

- **Asynchronous Apex:**

- **Future Method:** This method was used to send booking confirmation emails after the booking status was updated to "Confirmed."

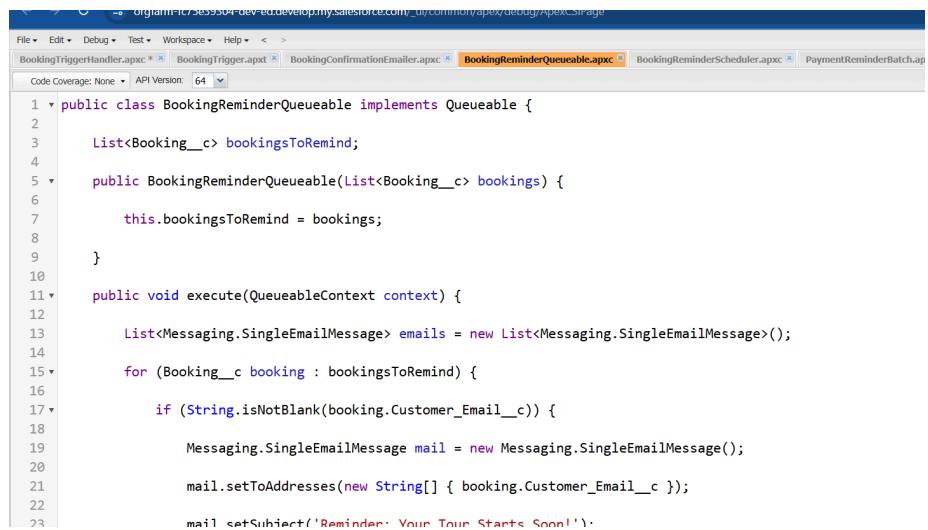


```

1 public class BookingConfirmationEmailer {
2
3     @future(callout=false)
4
5     public static void sendBookingConfirmation(Set<Id> bookingIds) {
6
7         List<Messaging.SingleEmailMessage> emails = new List<Messaging.SingleEmailMessage>();
8
9         List<Booking__c> bookings = [SELECT Id, Name, Customer_Email__c, Total_Billing_Amount__c
10
11             FROM Booking__c
12
13             WHERE Id IN :bookingIds];
14
15         for (Booking__c booking : bookings) {
16
17             if (String.isNotBlank(booking.Customer_Email__c)) {
18
19                 Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
20
21                 mail.setToAddresses(new String[] { booking.Customer_Email__c });
22
23                 mail.setSubject('Booking Confirmed: ' + booking.Name);
24
25             }
26
27         }
28
29     }
30
31 }

```

- **Queueable Class:** A queueable class was created to handle the logic of sending personalized reminder emails to customers exactly 3 days before their tour starts.

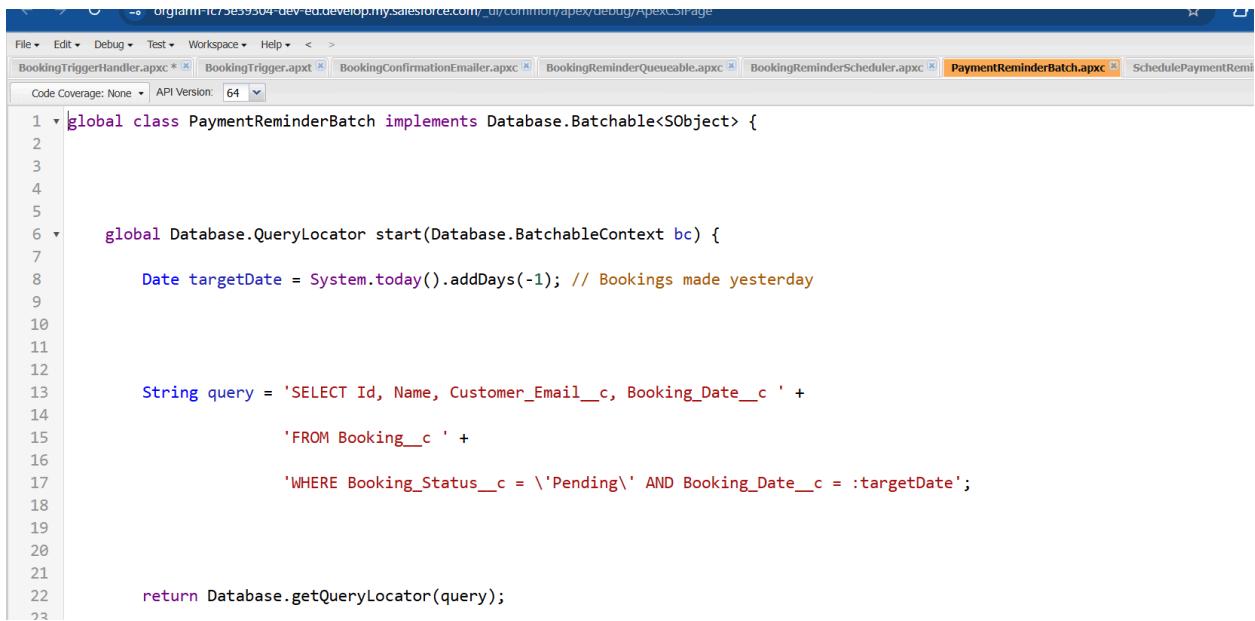


```

1 public class BookingReminderQueueable implements Queueable {
2
3     List<Booking__c> bookingsToRemind;
4
5     public BookingReminderQueueable(List<Booking__c> bookings) {
6
7         this.bookingsToRemind = bookings;
8
9     }
10
11     public void execute(QueueableContext context) {
12
13         List<Messaging.SingleEmailMessage> emails = new List<Messaging.SingleEmailMessage>();
14
15         for (Booking__c booking : bookingsToRemind) {
16
17             if (String.isNotBlank(booking.Customer_Email__c)) {
18
19                 Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
20
21                 mail.setToAddresses(new String[] { booking.Customer_Email__c });
22
23                 mail.setSubject('Reminder: Your Tour Starts Soon!');
24
25             }
26
27         }
28
29     }
30
31 }

```

- **Batch Class:** Daily payment reminder emails. A batch job was implemented to send payment reminders on a daily basis. It queried bookings with a 'Pending' payment status and emailed the respective customers.



```

1 global class PaymentReminderBatch implements Database.Batchable<SObject> {
2
3
4
5
6     global Database.QueryLocator start(Database.BatchableContext bc) {
7
8         Date targetDate = System.today().addDays(-1); // Bookings made yesterday
9
10
11
12         String query = 'SELECT Id, Name, Customer_Email__c, Booking_Date__c '
13             'FROM Booking__c '
14             'WHERE Booking_Status__c = \'Pending\' AND Booking_Date__c = :targetDate';
15
16
17         return Database.getQueryLocator(query);
18
19
20
21
22
23

```

- **Schedulable Apex:** Scheduled jobs for automation.



```

1 public class SchedulePaymentReminderBatch implements Schedulable {
2
3     public void execute(SchedulableContext sc) {
4
5         PaymentReminderBatch batch = new PaymentReminderBatch();
6
7         Database.executeBatch(batch, 200);
8
9     }
10
11 }

```

# Phase 3: UI/UX Development & Customization

## Lightning App Setup

A custom app named "**Tours & Travels CRM**" was built using the **App Manager** in Salesforce Lightning. This app centralized access to all CRM components such as Customer Info, Bookings, Payments, Employees, and more.

- Tabs were organized for ease of navigation: Customer, Booking, TravelPackage, Employee, Feedback, Payment, Task, Reports, and Dashboard.

The screenshot shows the Salesforce Lightning interface for the 'Tours & Travels CRM' app. The top navigation bar is highlighted with a red box and contains tabs for various CRM components. On the left, there's a sidebar with a 'Recently Viewed' section showing a list of customer names. An arrow points from this sidebar to the 'Customers Info' tab in the navigation bar.

## UI Elements

### Page Layouts

Customized layouts were created for each object to improve data visibility and user productivity. Key fields were rearranged logically to streamline user data entry and reduce cognitive load. (Customer Info, BookingGuest, TravelGuest, Employee, Booking, Booking Payment, Feedback)

### Customer Info

The screenshot shows the 'Customer Info' object in the 'Object Manager' under 'Setup'. The 'Page Layouts' tab is selected in the sidebar. The main area displays the layout configuration for the 'Customer Info' object, showing fields like Name, Email, Phone, and Date of Birth. The layout is designed to be user-friendly and efficient for data entry.

## BookingGuest

SETUP > OBJECT MANAGER  
**BookingGuest**

**Fields**

Section	Booking	Country	Passport Number
Blank Space	BookingGuest Name	Created By	Relation with Cus...
Age	BookingGuest Name	Gender	Special Needs
Age Category	City	Last Modified By	Visa Required

**BookingGuest Detail**

Information (Header visible on edit only)

Booking	Sample Text
Booking Guest Name	Sample Text
Age	29.646
Gender	Sample Text
Relation with Customer	Sample Text
Age Category	Sample Text
Special Needs	Sample Text

Standard Buttons: Edit, Delete, Clone, Change Owner, Change Record Type, Printable View, Edit Labels

Custom Buttons:

- Country: Sample Text
- City: Sample Text
- Passport Number: Sample Text
- Visa Required: ✓

## TravelGuest

SETUP > OBJECT MANAGER  
**TravelPackage**

**Fields**

Section	Average Rating	Guide Included	Maximum Group Size	Package Type	Region
Blank Space	Country	Guide Languages	Meals Included	Places Covered	Transportation Modes
Additional Notes	Created By	Insurance Included	Membership	Preferred Guide	Travel Package Name
Availability Status	Duration in Days	Last Modified By	Owner	Price Per Person	Travel Package Name

**TravelPackage Detail**

Information (Header visible on edit only)

TravelPackage Name	Sample Text
Region	Sample Text
Duration in Days	49.223
Package Type	Sample Text
Transportation Modes	Sample Text
Availability Status	Sample Text
Average Rating	99.957
Meals Included	Sample Text

Standard Buttons: Edit, Delete, Clone, Change Owner, Change Record Type, Printable View, Sharing, Sharing Hierarchies

Custom Buttons:

- Country: Sample Text
- Places Covered: Sample Text
- Price Per Person: \$123.45
- Membership: Sample Text
- Owner: Sample Text

## Employee

SETUP > OBJECT MANAGER  
**Employee**

**Fields**

Section	Availability Status	Department	Employee Name	Last Modified By	Role
Blank Space	City	Email	Employment Type	Owner	Salary
Address	Country	Employee ID	Employee Name	Joining Date	Phone
Assigned Region	Created By	Employee Name	Languages Spoken	Profile Picture	Profile Picture

**Employee Detail**

Information (Header visible on edit only)

Employee ID	GFM-2004-001234
Employee Name	GEN-2004-001234
Employee Name	Sample Text
Email	sarah.sample@company.com
Phone	1.415.555.1212
Role	Sample Text
Department	Sample Text
Employment Type	Sample Text
Salary	\$123.45
Joining Date	7/10/2025
Availability Status	Sample Text
Languages Spoken	Sample Text
Assigned Region	Sample Text

Standard Buttons: Edit, Delete, Clone, Change Owner, Change Record Type, Printable View, Sharing, Sharing Hierarchies

Custom Buttons:

- Profile Picture: Sample Text
- Owner: Sample Text

## Booking

SETUP > OBJECT MANAGER  
**Booking**

## Booking Payment

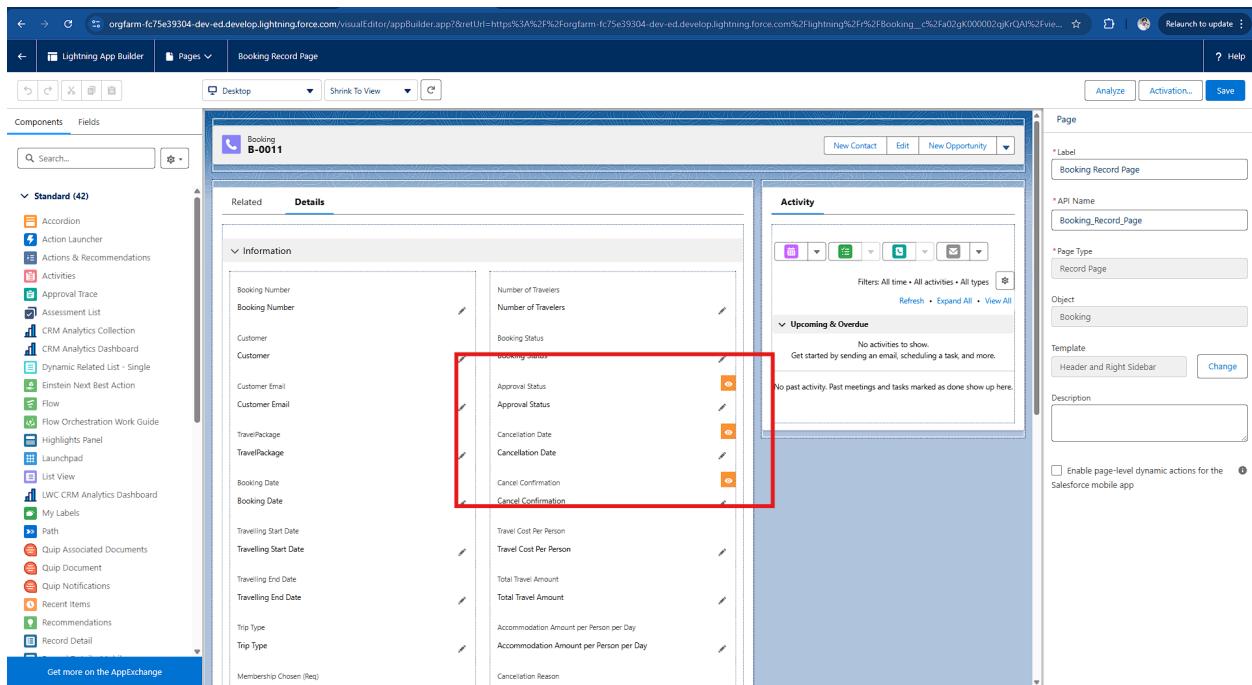
SETUP > OBJECT MANAGER  
**Booking Payment**

## Feedback

SETUP > OBJECT MANAGER  
**Feedback**

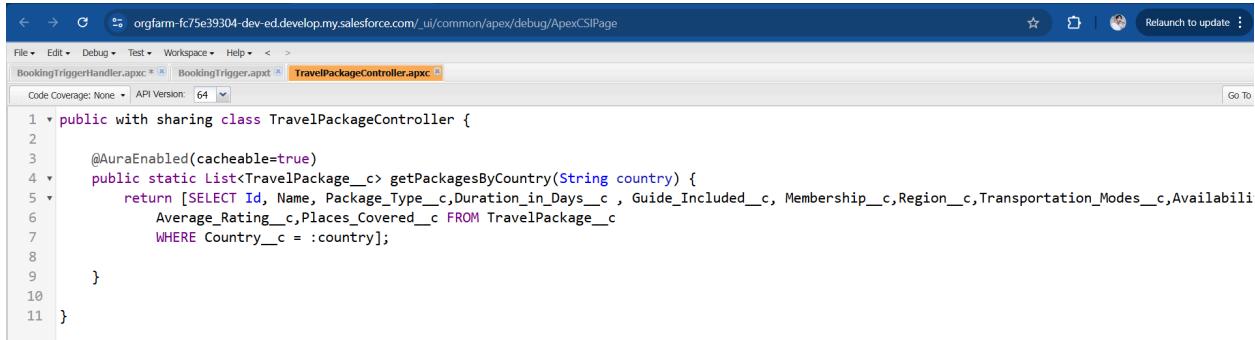
## Dynamic Forms

Dynamic Forms were enabled for the Booking object to show or hide fields based on conditions. For instance, cancellation-related fields only appear when the booking status is set to “Cancelled”.

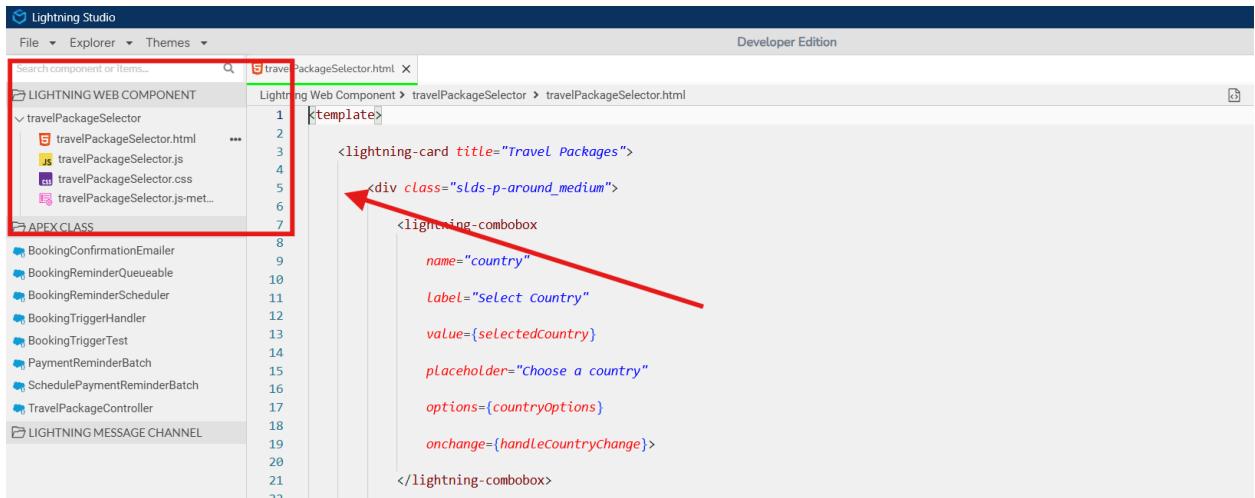


## Lightning Pages

Lightning Record Pages were customized to display key information, related lists, charts, and LWCs. This provided users with a single pane view of all essential CRM operations.



```
orgfarm-fc75e39304-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help < >
BookingTriggerHandler.apxc BookingTrigger.apxc TravelPackageController.apxc
Code Coverage: None API Version: 64 Go To
1 public with sharing class TravelPackageController {
2
3     @AuraEnabled(cacheable=true)
4     public static List<TravelPackage__c> getPackagesByCountry(String country) {
5         return [SELECT Id, Name, Package_Type__c, Duration_in_Days__c , Guide_Included__c, Membership__c, Region__c, Transportation_Modes__c, Availability_Average_Rating__c, Places_Covered__c FROM TravelPackage__c
6             WHERE Country__c = :country];
7
8     }
9
10 }
11 }
```



Lightning Studio

File Explorer Themes

Developer Edition

Search component or items... travelPackageSelector.html

LIGHTNING WEB COMPONENT

- travelPackageSelector
  - travelPackageSelector.html
  - travelPackageSelector.js
  - travelPackageSelector.css
  - travelPackageSelector.js-met...

APEX CLASS

- BookingConfirmationEmailer
- BookingReminderQueueable
- BookingReminderScheduler
- BookingTriggerHandler
- BookingTriggerTest
- PaymentReminderBatch
- SchedulePaymentReminderBatch
- TravelPackageController

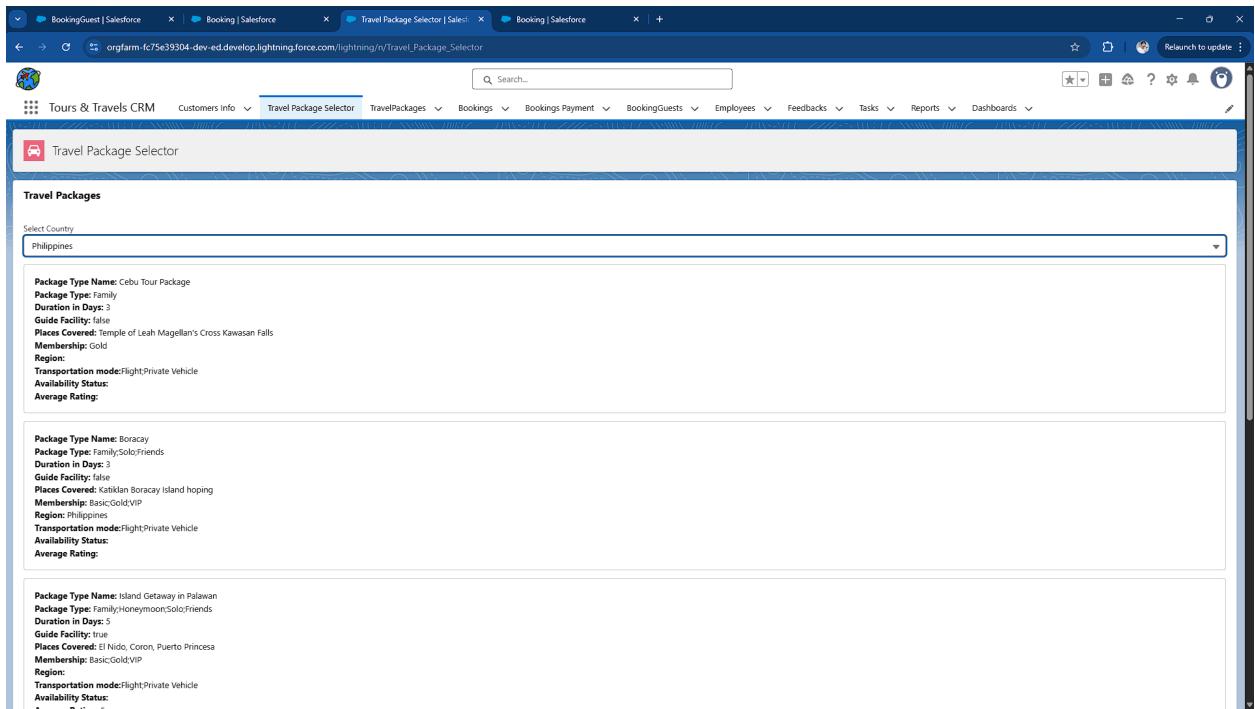
LIGHTNING MESSAGE CHANNEL

travelPackageSelector.html

```
<template>
    <lightning-card title="Travel Packages">
        <div class="slds-p-around_medium">
            <lightning-combobox
                name="country"
                label="Select country"
                value={selectedCountry}
                placeholder="Choose a country"
                options={countryOptions}
                onchange={handleCountryChange}>
            </lightning-combobox>
        </div>
    </lightning-card>
</template>
```

## Lightning Web Components (LWC)

Developed a custom LWC named TravelPackageSelector, which filters packages by country dynamically. This component enhances user interactivity by allowing users to select packages without refreshing the page.



## User Management

Created users for each role (e.g., Travel Agent, Tour Guide, Finance Officer) and assigned them appropriate Profiles and Roles. This ensured secure, role-specific access throughout the CRM.

- Users created for each role (Travel Agent, Guide, Finance, Marketing Executive Profile & Customer Service Rep) cloning Standard Platform User. I set Custom Object Permission for each Profile.

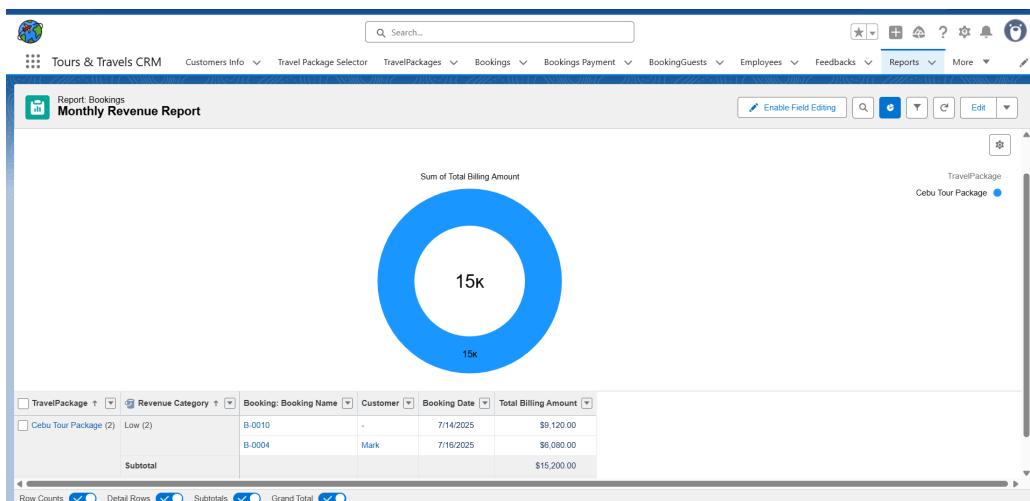
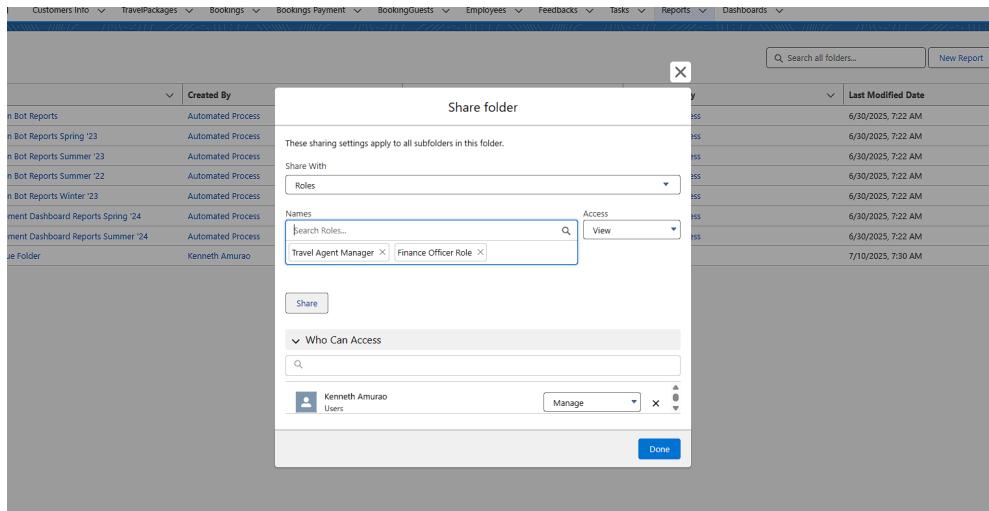
Action	Full Name	Alias	Username	Role	Active	Profile
Edit	Amurao, Kenneth	amu	amurao kenneth372@agentforce.com		✓	System Administrator
Edit	Chatter Expert	Chatter	chatty-00dg000000swpwua! ik0m0uxayh@chatter.salesforce.com		✓	Chatter Free User
Edit	EPIC_OrgFarm	OEPIC	epic.th.8ce6bc2820b5@orgfarm.salesforce.com		✓	System Administrator
Edit	Jackson, Michael	mjack	nethamurao005@salesforce.com	TravelAgent Manager	✓	Travel Agent Profile
Edit	Kent, Clark	ckent	sia.amurao kenneth@salesforce.com	TravelAgent	✓	Travel Agent Profile
Edit	User_Integration	Integ	integration@00dgk00006swpwua.com		✓	Analytics Cloud Integration User
Edit	User_Security	sec	insightssecurity@00dgk00006swpwua.com		✓	Analytics Cloud Security User
Edit	Wayne, Bruce	bwayn	nethamurao@salesforce.com	TravelAgent	✓	Travel Agent Profile

- Assigned appropriate profiles and roles to enforce hierarchy.

- Angular University Foundation
  - Add Role
  - CEO
    - Edit | Del | Assign
    - Add Role
  - CFD
    - Edit | Del | Assign
    - Add Role
  - COO
    - Edit | Del | Assign
    - Add Role
  - Customer Service Rep Role
    - Edit | Del | Assign
    - Add Role
  - Finance Officer Role
    - Edit | Del | Assign
    - Add Role
  - Marketing Executive Role
    - Edit | Del | Assign
    - Add Role
  - SVP\_Customer Service & Support
    - Edit | Del | Assign
    - Add Role
  - SVP\_Human Resources
    - Edit | Del | Assign
    - Add Role
  - SVP\_Sales & Marketing
    - Edit | Del | Assign
    - Add Role
  - Travel Agent Manager
    - Edit | Del | Assign
    - Add Role
  - Travel Agent
    - Edit | Del | Assign
    - Add Role
  - Travel Tour Guide
    - Edit | Del | Assign
    - Add Role

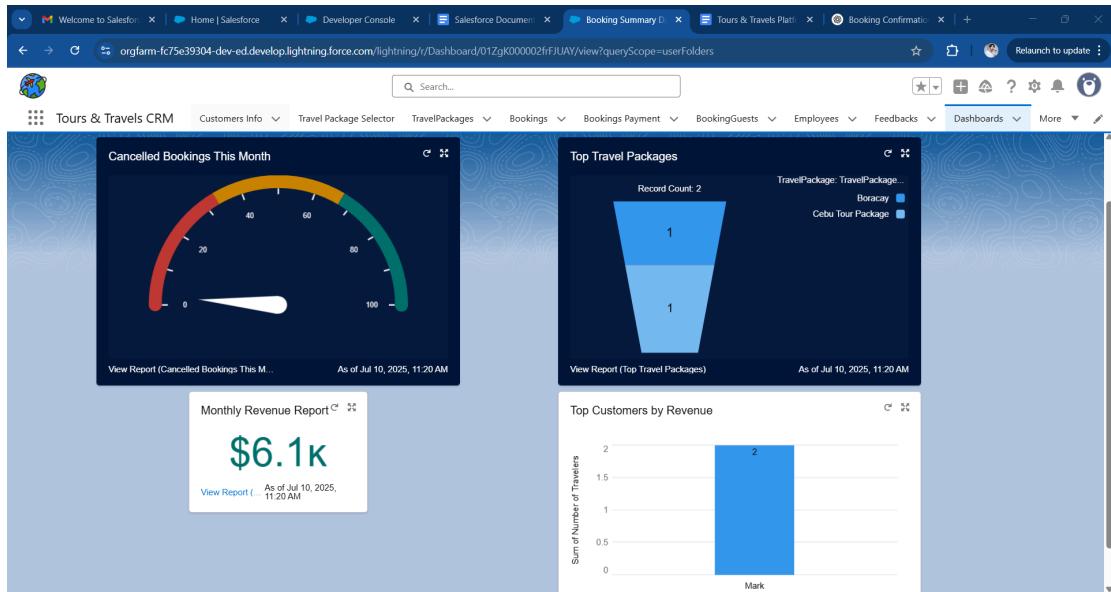
## Reports and Dashboards

- Reports: Revenue, Top Packages, Pending Payments, Employee Role Breakdown.



Note: Create a “share folder” first, where you can assign a specific role and store the created report.

- Dashboards: Monthly Revenue, Booking Trends, Package Demand.



Note: If you Edit or Create a Dashboard you can customize your charts through adding “widgets” KPI’s based on the created from “Reports”

# Phase 4: Data Migration, Testing & Security

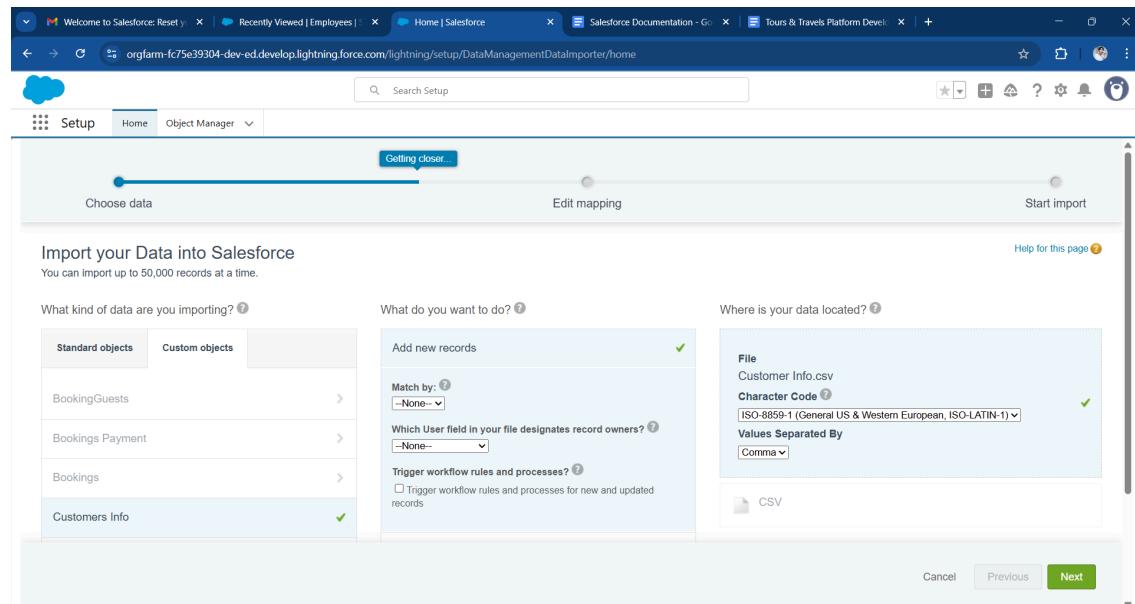
## Data Loading Process

To bring existing data into the CRM, the **Salesforce Data Import Wizard** was used due to its simplicity and support for standard and custom objects. This tool allowed a user-friendly, guided process for importing:

- **Objects Migrated:** Customer\_\_c, TravelPackage\_\_c, and Employee\_\_c.
- Each CSV file contained a minimum of 20 records with clean data.
- Fields were carefully mapped to ensure accurate data alignment with the corresponding Salesforce fields.

Here is the sample Data Loading Process:

### Customer Info CSV import wizard



## Import Map Field

The screenshot shows two consecutive steps in the Salesforce Data Import Wizard:

- Step 1: Review & Start Import** (Top Window): A modal dialog box displays the message: "Congratulations, your import has started! Click OK to view your import status on the Bulk Data Load Job page." An "OK" button is visible at the bottom right of the modal.
- Step 2: Bulk Data Load Jobs** (Bottom Window): The Bulk Data Load Jobs page is displayed. It shows a single job entry with the Job ID: 750gK000008MQ8v. The page includes details such as Start Time (7/13/2025, 7:46 AM PST), End Time (7/13/2025, 7:46 AM PST), and Progress (100%). The "Object" field is set to "Customer Info". The "Status" is "Closed" with a processing time of 135 ms.

Milestone proof activity completed minimum 20 records for (Customer\_Info\_\_c, TravelPackage\_\_c, and Employee\_\_c.)

The screenshot shows the Bulk Data Load Jobs page with the following details:

- Job Summary:** CPU: 722 milliseconds, IO: 5,193 bytes, Disk: 28,992 bytes.
- In Progress:** No records to display.
- Completed last 7 days:** A table lists six completed jobs. The table columns include Job ID, Submitted By, Start Time, End Time, Status, Job Type, Operation, Object, Records Processed, Records Failed, and Progress. The last three rows of the table are highlighted with red boxes.

Job ID	Submitted By	Start Time	End Time	Status	Job Type	Operation	Object	Records Processed	Records Failed	Progress
750gK000008MWXX	Amurao, Kenneth	7/13/2025, 9:14 AM	7/13/2025, 9:14 AM	Closed	Bulk V1	Insert	Employee	20	0	00:01
750gK000008MYIG	Amurao, Kenneth	7/13/2025, 9:06 AM	7/13/2025, 9:07 AM	Closed	Bulk V1	Insert	Employee	20	1	00:00
750gK000008MSUz	Amurao, Kenneth	7/13/2025, 8:41 AM	7/13/2025, 8:41 AM	Closed	Bulk V1	Insert	Employee	20	19	00:01
750gK000008MWCL	Amurao, Kenneth	7/13/2025, 8:28 AM	7/13/2025, 8:28 AM	Closed	Bulk V1	Insert	TravelPackage	20	0	00:00
750gK000008MT6L	Amurao, Kenneth	7/13/2025, 8:06 AM	7/13/2025, 8:06 AM	Closed	Bulk V1	Insert	Customer Info	20	0	00:01
750gK000008MQ8v	Amurao, Kenneth	7/13/2025, 7:46 AM	7/13/2025, 7:46 AM	Closed	Bulk V1	Insert	Customer Info	20	0	00:00

After import, the integrity of records was verified by reviewing the record list views and sample record detail pages.

Note: After clicking “View Record” it will automatically download imported record

End Time	/13/2025, 7:46 AM PST	In Progress Batches	0	Apex Processing Time (ms)	29
Time to Complete ([hh:]mm:ss)	00:01	Completed Batches	1		
Object	Customer Info	Failed Batches	0		
External ID Field		Progress	100%		
Content Type	CSV	Records Processed	20		
Concurrency Mode	Parallel	Records Failed	0		
API Version	64.0	Retries	0		

Batches									
View Request	View Result	Batch ID	Start Time	End Time	Total Processing Time (ms)	API Active Processing Time (ms)	Apex Processing Time (ms)	Records Processed	Records Failed
<a href="#">View Request</a>	<a href="#">View Result</a>	751gK000006dKLC	7/13/2025, 7:46 AM	7/13/2025, 7:46 AM	135	79	29	20	0
								0	0
								Completed	

A	B	C	D	E	F	G	H	I
1 City_c	Country_c	Customer_Name_c	DateOfBirth_c	Email_c	Phone_c			
2 Palayan	Philippines	Paul Ford	29/07/1992	paul.ford@yahoo.com	1953867218			
3 New York	USA	Amy Allen	07/12/1960	amy.allen@gmail.com	1975243167			
4 Palayan	Switzerland	Barbara Frye	31/01/1969	barbara.frye@yahoo.com	1986273561			
5 Bangalore	Thailand	Sarah Lucas	13/12/1978	sarah.lucas@yahoo.com	1973623187			
6 Jeddah	Thailand	Sabrina Cortez	01/06/1966	sabrina.cortez@yahoo.com	1987435612			
7 Quezon City	Thailand	Jason Lee Jr.	25/07/1987	jason.jr@yahoo.com	1983716481			
8 Bangalore	USA	Andrea Downs	30/01/1987	andrea.downs@gmail.com	1915736481			
9 Mumbai	Philippines	Ashley Lynn	15/03/2003	ashley.lynn@yahoo.com	1966381558			
10 Chennai	Thailand	Daniel Lindsey	08/02/2007	daniel.lindsey@yahoo.com	1967361852			
11 Mumbai	Thailand	Beth Collins	22/07/1982	beth.collins@yahoo.com	1975834658			
12 Hyderabad	Switzerland	Edwin Garcia	01/11/1996	edwin.garcia@gmail.com	1967843158			
13 Columbus	Thailand	Alec Medina	03/08/1970	alec.medina@yahoo.com	1987543185			
14 Bangalore	USA	Christopher Simon	29/03/2000	christopher.simon@gmail.com	1967391873			
15 Los Angeles	India	Jerry Lopez	12/10/2004	jerry.lopez@gmail.com	1948613287			
16 Alaminos	Thailand	Lindsay Lewis	16/09/1965	lindsay.lewis@gmail.com	1987328457			
17 Chicago	India	Haley Clayton	19/11/1967	haley.clayton@yahoo.com	1915746132			
18 Delhi	Philippines	Katherine Shaw	08/06/1976	katherine.shaw@gmail.com	1936857286			
19 Medina	Thailand	Chelsea Ashley	13/08/1977	chelsea.ashley@gmail.com	1978653264			
20 Geneva	Switzerland	Gerald Bryant	16/09/1988	gerald.bryant@gmail.com	1986432672			
21 Bern	India	Joshua Hill	01/02/1996	joshua.hill@gmail.com	1978436481			

## Security Implementation

To protect business-sensitive information and ensure proper user access:

- **Field History Tracking:** Enabled for key fields on Booking\_c and TravelPackage\_c to monitor changes over time. This allows audit trails for tracking who changed what and when.

- **Duplicate Rules & Matching Rules:** Created on the Customer\_\_c object to prevent duplicate entries based on unique combinations of Email and Phone.

## Custom Matching Rule

Matching Rule Detail	
Object	Customer Info
Rule Name	Unique_Email_and_Phone_Number_Combination
Unique Name	Unique_Email_and_Phone_Number_Combination
Description	
Matching Criteria	(Customer Info: Email EXACT MatchBlank = TRUE) AND (Customer Info: Phone EXACT MatchBlank = TRUE)
Status	Inactive
Created By	Kenneth Amurao, 7/10/2025, 2:00 PM
Modified By	Kenneth Amurao, 7/10/2025, 2:00 PM

## Duplication Rule

Duplicate Rule Detail	
Rule Name	unique Email and Phone
Description	
Object	Customer Info
Record-Level Security	Enforce sharing rules
Action On Create	Allow
Action On Edit	Allow
Alert Text	Email and Phone must be Unique
Active	✓
Matching Rule	Unique_Email_and_Phone_Number_Combination Mapped
Conditions	
Created By	Kenneth Amurao, 7/10/2025, 2:07 PM
Modified By	Kenneth Amurao, 7/10/2025, 2:10 PM

- **Profiles & Roles:** Users were assigned to predefined profiles such as Travel Agent, Finance Officer, and Tour Guide. Roles were used to determine visibility and record-level access.

The screenshot shows the Salesforce Setup interface under the 'Profiles' section. A search bar at the top left contains 'profile'. The main content area is titled 'Profile' and shows the 'Travel Agent Profile'. Below the title, it says 'Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.' A note indicates that if Record Types are used, edit links are available in the Record Type Settings section. The 'Profile Detail' section shows the profile name 'Travel Agent Profile', user license 'Salesforce Platform', and a checked 'Custom Profile' checkbox. It also lists the 'Created By' and 'Modified By' users and dates. The 'Page Layouts' section shows 'Standard Object Layouts' with tabs for 'Global' and 'Global Layout', and 'Lead' with 'Lead Layout' selected. A note at the bottom right says 'Did you find what you're looking for? Try using Global Search.'

- **Permission Sets:** Used to extend object access for specific users (e.g., Travel Agent Manager needing report access) without changing their base profile.

The screenshot shows the Salesforce Setup interface under the 'Permission Sets' section. A search bar at the top left contains 'Permission Set'. The main content area is titled 'Permission Set' and shows the 'Extra Permission For Travel Agent Manager'. Below the title, it has a 'Find Settings...' search bar and buttons for 'Clone', 'Delete', 'Edit Properties', 'Manage Assignments', and 'View Summary'. The 'Permission Set Overview' section shows the API name 'Extra\_Permission\_For\_Travel\_Agent\_Manager', namespace prefix 'Extra\_Permission', and creation and modification details for 'Kenneth Amurao'. It also shows session activation required and permission set groups added. A note at the bottom left says 'Did you find what you're looking for? Try using Global Search.'

- **Sharing Rules:** Configured to share customer records automatically with Tour Guide roles based on their association with bookings.

The screenshot shows the Salesforce Setup interface under the 'Sharing Rules' section. It displays four sections: 'Booking Sharing Rules' (no sharing rules specified), 'Booking Payment Sharing Rules' (no sharing rules specified), 'Customer Info Sharing Rules' (action 'Edit | Del' and criteria 'Owner in Role: Travel Agent', shared with 'Role: Travel Tour Guide', access level 'Read Only'), and 'Employee Sharing Rules' (no sharing rules specified). Each section has 'New' and 'Recalculate' buttons. Help links are provided for each section: 'Booking Sharing Rules Help', 'Booking Payment Sharing Rules Help', 'Customer Info Sharing Rules Help', and 'Employee Sharing Rules Help'.

## Test Class Creation

- **Apex Test Class:** A test class named BookingTriggerTest was developed to verify the logic inside Booking triggers and handlers. The class inserted mock records and confirmed that associated BookingGuest\_\_c and BookingPayment\_\_c records were automatically created.
- **Challenges and Error:** The biggest challenge is that some of the fields given in Milestone 4 are not aligned with the code provided in BookingTriggerHandler.apxc and BookingTriggerTest.apxc. Some of the errors are in the BookingTriggerTest.apxc file, where certain fields are not properly defined in Customer\_Info\_\_c and Membership\_Chosen\_\_c. For example, I encountered a "Missing Date\_Of\_Birth\_\_c" error on BookingTriggerTest.apxc. To fix this, I added the code: Date\_of\_Birth\_\_c = Date.today().addYears(-30);
- **Coverage:** More than 90% code coverage was achieved across Apex triggers and classes.

Status	Test Run	Enqueued Time	Duratio	Failures	Total
x	BookingTriggerTest			1	1
x	TestRun @ 2:30:14 am			1	1
x	BookingTriggerTest			1	1
x	TestRun @ 2:35:42 am			1	1
x	BookingTriggerTest			1	1
x	testTriggerCreatesPaymentAndGuestsWithUpdatedFields			1	1
x	TestRun @ 2:59:01 am			1	1
x	BookingTriggerTest			1	1
x	testTriggerCreatesPaymentAndGuestsWithUpdatedFields			1	1
✓	TestRun @ 2:59:49 am			0	1
✓	TestRun @ 3:00:01 am			0	1
✓	BookingTriggerTest			0	1
✓	testTriggerCreatesPaymentAndGuestsWithUpdatedFields			0	1

BookingTriggerTest.apxc

@isTest

```
private class BookingTriggerTest {
```

@isTest

```
static void testTriggerCreatesPaymentAndGuestsWithUpdatedFields() {
```

```
// Create test customer
```

```
Customer_Info__c customer = new Customer_Info__c(
```

```
Name = 'John Doe',
```

```
Email__c = 'annapurna@gmail.com', // enter your valid email address here  
Phone__c = '1234567890',  
DateOfBirth__c = Date.today().addYears(-30)  
);  
  
insert customer;  
  
// Create a Travel Package  
  
TravelPackage__c packageRec = new TravelPackage__c(  
    Name = 'European Delight',  
    Country__c = 'India',  
    Price_Per_Person__c = 2000,  
    Duration_in_Days__c = 3,  
    Places_Covered__c = 'Paris, Rome, Barcelona'  
);  
  
insert packageRec;  
  
// Create a test Booking record  
  
Booking__c booking = new Booking__c(  
    Number_of_Travelers__c = 3,  
    Booking_Status__c = 'Pending',  
    Travelling_Start_Date__c = Date.today().addDays(10),  
    TravelPackage__c = packageRec.Id,  
    Membership_Chosen_Req__c = 'Gold',  
    Preferred_Accommodation__c = 'Guest House',
```

```

Customer_Info__c = customer.Id,
//Customer__c = customer.Id, // fixed field name

Customer_Email__c ='annapurna@gmail.com', // enter a valid email address here

Booking_Date__c = Date.today() // added required field

);

Test.startTest();

insert booking; // Trigger will fire here

Test.stopTest();

// Validate Booking_Payment__c creation

List<Booking_Payment__c> payments = [
    SELECT Id, Booking__c, Payment_Status__c
    FROM Booking_Payment__c
    WHERE Booking__c = :booking.Id
];

System.assertEquals(1, payments.size(), 'One payment record should be created.');

System.assertEquals('Pending', payments[0].Payment_Status__c, 'Default Payment Status should be Pending.');

// Validate BookingGuest__c creation

List<BookingGuest__c> guests = [
    SELECT Id, Booking__c, Name
    FROM BookingGuest__c
    WHERE Booking__c = :booking.Id
];

```

```
System.assertEquals(3, guests.size(), 'Three BookingGuest records should be created.');
```

```
System.assertEquals('Guest 1', guests[0].Name, 'Guest naming should follow convention.');
```

```
}
```

## Testing Execution

**Test Case 1:** Ensure that a new Customer record can be added successfully when all required fields are filled.

### Test Steps:

- Open the Customer module or object page.
- Enter values for all mandatory fields.
- Click on the **Save** button

### Expected Result:

- The Customer record should be created and stored without any errors.
- The newly added record should be visible in the list of existing Customers.

The screenshot shows the Salesforce Lightning interface for the Tours & Travels CRM. A new Customer record for "Kenneth Amurao" has been created, as indicated by the green success message at the top right: "Customer Info 'Kenneth Amurao' was created." The "Details" tab is selected, displaying the record's ID (C-0003), name (Kenneth Amurao), email (nethamurao05@gmail.com), date of birth (7/5/2025), age (0), and owner (Kenneth Amurao). In the "Activity" sidebar, a list of 3 items is shown, sorted by Customer Info Name, with Kenneth Amurao listed at position 3. A red box highlights the entry for Kenneth Amurao, and a red arrow points from this highlighted area towards the bottom of the list, indicating the newly created record.

**Test Case 2:** Confirm that a Booking can be successfully recorded when all required information is provided.

**Test Steps:**

- Go to the Booking object interface.
- Complete all necessary fields with appropriate values.
- Click **Save** to submit the form.

**Expected Result:**

- A new Booking entry should be created and saved properly.
- The record should show up on the Booking List view.
- A corresponding entry should automatically be created in the Booking Payments object with the Payment Status set to "Pending".
- A number of BookingGuest entries equal to the "Number of Travellers" specified in the Booking should also be created and linked to the Booking record.

The screenshot displays two views of the Tours & Travels CRM system. The top view shows the 'Booking' details for record B-0010. The 'Details' tab is selected, showing fields like Booking Number (B-0010), Number of Travelers (3), Customer (Kenneth Amurao), Booking Status (Pending), and Travel Cost Per Person (\$40.00). The bottom view shows the 'Recently Viewed' list, which includes three items: B-0004, B-0007, and B-0010. The item B-0010 is highlighted with a red box and has a red arrow pointing from the 'Booking Name' column towards it, indicating that the recently viewed list includes the newly created booking.

**Test Case 3:** Verify that when the Payment Status of a Booking Payment record is changed to "Completed", the related Booking record updates to "Confirmed", and the customer is notified via email.

#### Test Steps:

- Navigate to the Booking Payment object page.
- Change the **Payment Status** field value to "Completed".
- Save the changes.

#### Expected Result:

- The Booking Payment record should be updated successfully.
- The corresponding Booking record should automatically reflect a "Confirmed" status.
- An email should be sent to the customer confirming the Booking and successful payment.

## Test Case 4: Booking Creation With Employee TourGuide

### Objective:

- Prevent booking with a past travel start date.

### Test Steps:

- Navigate to the Booking object.
- Create a new booking.
- Fill other required fields
- Checked the “Required” Tou Guide”.
- Create or add Existing Employee and click Save.

### Expected Result:

- Successful Banner Appear then te Guide employee assigned will booking details

### Validation Area:

- Lookup Filter on Employee\_\_c field in booking, Role and Availability

The screenshot shows a Salesforce booking creation interface. At the top left is a phone icon and the text "Booking B-0011". The page contains several input fields and sections:

- Travelling End Date:** 8/2/2025
- Trip Type:** Family
- Membership Chosen (Req):** Gold
- Preferred Accommodation:** Guest House
- Include Travel Insurance:** (checkbox)
- Require Visa Assistance:** (checkbox)
- Require Tour Guide:** (checkbox) - This field is highlighted with a red border and has a checked mark.
- Guide Assigned:** E-0005
- Preferred Guide Language:** Filipino
- No. of Booking Guests Info Available:** 3

On the right side, there are summary sections:

- Total Accommodation Amount:** \$12,000.00
- Total Billing Amount:** \$12,120.00
- Owner:** Kenneth Amurao (with a user icon)

## Test Case 5: Duplicate Customer Creation

### Objective:

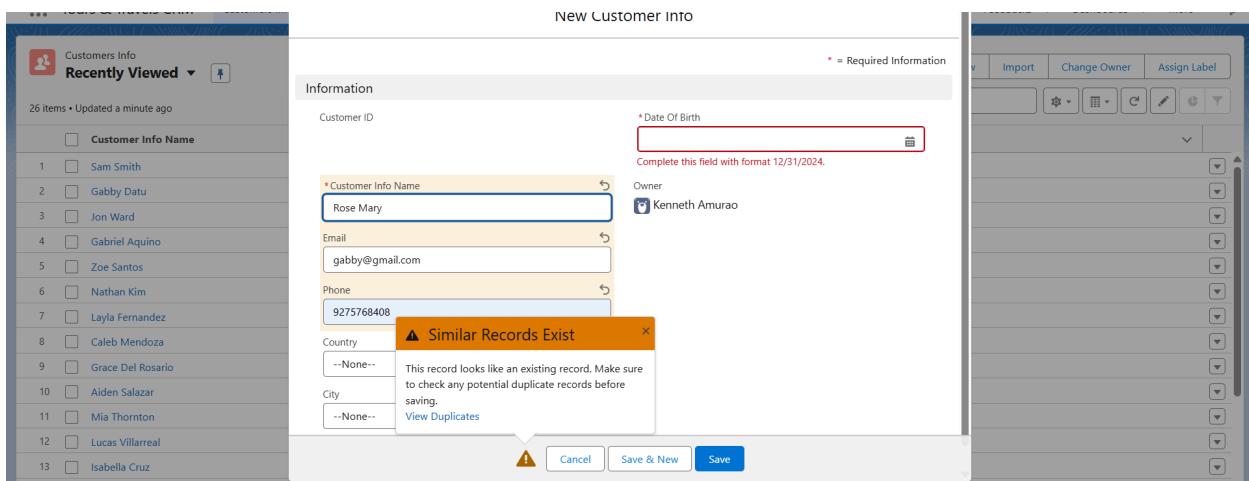
- Prevent creation of customers with same phone and email.

### Test Steps:

- Create a Customer Info record with Email = Gabby@gmail.com and Phone = 1234567890.
- Try creating another customer with the same email and phone.

### Expected Result:

- Error: "Email and Phone must be Unique."
- Validation Area:** Duplicate rule, Matching rule.



## Test Case 6: Dynamic Field Visibility

### Objective:

- Verify dynamic forms hide/show fields.

### Test Steps:

- Open a Booking record with Booking Status = Cancelled.
- Check if *Cancellation Date*, *Cancel Confirmation*, *Approval Status* are visible.

### Expected Result:

- Fields are visible only when status = Cancelled.

### Validation Area:

- Dynamic Forms.

The screenshot shows a 'New Booking' form in the Tours & Travels CRM. The 'Booking Status' field is set to 'Confirmed'. A red error message box appears at the bottom left, stating '∅ We hit a snag.' and listing 'Review the following fields' with a bullet point for 'Booking\_Status'. The right side of the screen shows a sidebar with various tabs like Import, Change Owner, and Assign Label.

## Test Case 7: Trip Cancellation Approval Workflow

### Objective:

- Validate approval process when booking is cancelled.

### Test Steps:

- Update an existing Booking record:
- Booking Status = Cancelled
- Cancel Confirmation = True
- Submit the record for approval.

### Expected Result:

- Approval request sent to manager.
- Email sent upon approval or rejection.

**Validation Area:** Approval process, email templates.

The screenshot displays two main sections of a software interface. On the left, the 'Details' tab of a booking record is shown. The booking number is B-0013, and the customer is Gabriel Aquino. The travel package is Cebu Cultural Tour, and the booking date is 7/17/2025. The travelling start date is 8/1/2025, and the end date is 8/5/2025. The trip type is Group. The membership chosen is required. The booking status is set to 'Cancelled', and the cancel confirmation is checked. The number of travelers is 2. The travel cost per person is \$300.00, and the total travel amount is \$600.00. The accommodation amount per person per day is \$3,000.00. The cancellation reason is listed as 'Cancelling Reason'. On the right, the 'Activity' section shows no upcoming or overdue activities, with a note to get started by sending an email, scheduling a task, and more. It also indicates no past activity.

## Phase 5: Deployment, Documentation & Maintenance

### Deployment Strategy

The deployment methodology for the Tours & Travels CRM system was executed through the use of **Change Sets**, a widely accepted and secure method in the Salesforce ecosystem. This process entailed:

- **Preparation of Outbound Change Sets:** All essential metadata components—including custom objects, fields, automation flows, Apex classes, analytical reports, dashboards, and user access configurations (profiles and permission sets)—were aggregated and packaged within the source environment.
- **Transmission to Target Environment:** The prepared change set was transmitted to the designated target instance, whether production or staging.
- **Validation and Execution:** Prior to final deployment, a comprehensive validation step was conducted to confirm that all required dependencies were included. Once validated, the components were successfully deployed.
- 

### Maintenance and Monitoring Plan

To ensure continuous operational integrity and system resilience, the following monitoring and maintenance protocols are established:

- **Audit Log Supervision:** Login histories and setup audit trails are reviewed on a regular basis to detect unauthorized access or administrative changes.
- **Scheduled Job Oversight:** The statuses of automated jobs, including both Schedulable and Batch Apex executions, are monitored through the Apex Jobs interface.
- **Data Quality Assurance:** Duplicate management rules and validation rules are utilized to ensure the integrity and reliability of entered data.
- **Field History Analysis:** Key object fields (such as those in Bookings and Travel Packages) are tracked to audit data changes and facilitate debugging.
- **Access Control Reviews:** Profiles, Permission Sets, and Role assignments are periodically audited to maintain appropriate data visibility and compliance.

## Troubleshooting Approach

In the event of unexpected application behavior or systemic errors, a formalized troubleshooting methodology is employed:

1. **User Issue Documentation:** Collect and analyze user-submitted reports or support tickets detailing the observed issues.
2. **Debug Log Activation:** Enable debug logs for relevant users or classes to trace runtime exceptions and performance bottlenecks.
3. **Automation Failure Inspection:** Examine error logs in Flows and Process Builders to identify failed or misconfigured automation.
4. **Embedded Error Messaging:** Utilize predefined error messages within Flows and Apex logic to guide users and expedite issue identification.
5. **Deployment Diagnostics:** Review validation logs in Change Sets to troubleshoot deployment failures and missing dependencies.

## Conclusion

The development and implementation of the Tours & Travels CRM platform on Salesforce comprehensively fulfill the strategic and operational needs of modern travel service providers. From efficient customer onboarding and itinerary bookings to automated approval workflows and payment tracking, the system has proven effective in enhancing organizational efficiency.

By integrating declarative tools such as Flows and Reports alongside programmatic assets like Apex Triggers, Classes, and Lightning Web Components, the solution is both robust and extensible. The modular design ensures adaptability to future business requirements and facilitates potential upgrades such as AI-driven recommendations, chatbot support, and mobile accessibility.

The successful culmination of this project reflects a thorough understanding of Salesforce architecture and development best practices, reinforcing the project's viability for real-world deployment.