TripAdvisor E-Management

College: 7155 - PSG Institute of Technology and Applied Research

Team ID: NM2024TMID00621

Team Lead: Suriya Sampathkumar (D1D35391C5E72E6995F4A4F9E021C474)

Team Members:

Devadarshini M	20F3289BE510861861D41E9D291D61C9
Ramya A	CED35E837A249C183384073A9FC98513
Renil Shaji	5393AB9DFFECB9782D14BC338E46948E
Suriya Sampathkumar	D1D35391C5E72E6995F4A4F9E021C474
Varshha D	9CDCDC27EFB027899CC4E995C0D2EBD2

1. Project Overview

The **TripAdvisor E-Management app** integrates with the **Salesforce platform** to serve as an all-in-one travel companion. This app provides users with the ability to plan, book, and optimize their travel experience, while helping them discover millions of reviews, recommendations, and insights shared by fellow travellers. The application focuses on managing hotels, flights, and food options, making it easier for customers to make informed decisions and streamline their travel planning process. This system also helps businesses automate the data management process related to hotels, food options, and flight schedules while offering customers benefits such as personalized discounts and timely flight reminders.

2. Objectives

The primary goals of this project are:

1. **Data Management Efficiency**: Automate the data management processes for hotels, flights, and food options, reducing manual effort and errors.
2. **Personalized Customer Discounts**: Implement a discount system based on the customer’s spending, with tiered discount rates for specific spending thresholds.
3. **Timely Flight Reminders**: Automate the process of sending reminders to customers about their flight schedules 24 hours prior to departure.
4. **Real-Time Updates**: Ensure real-time updates to hotel data when food options are added or modified, maintaining accurate counts and availability.

Business Goals:

1. **Simplify Operations**: Streamline the management of hotel data, flight schedules, and food options, saving time and minimizing errors.
2. **Enhance Customer Experience**: Provide a better customer experience through automatic discounts based on purchase amounts and timely notifications about upcoming flights.
3. **Automate Repetitive Tasks**: Replace manual processes with automated solutions that reduce the administrative burden and improve accuracy.
4. **Improve Data Integrity**: Maintain accurate data relationships across hotel, food, and flight records through Salesforce automation tools like Apex Triggers and Flows.

Specific Outcomes:

1. **Automated Hotel and Food Option Management**: When food options are added or updated, the system will automatically update the corresponding hotel records, including recalculating the total count of available food options for each hotel.
2. **Customer Discount System**: A flow-based automation will calculate discounts based on the customer’s total purchase amount. Discounts are applied in tiers—full, partial, or none—depending on spending.
3. **Flight Schedule Reminders**: A scheduled job will send reminder emails to customers who have booked flights, ensuring they are aware of their flight timing 24 hours before departure.

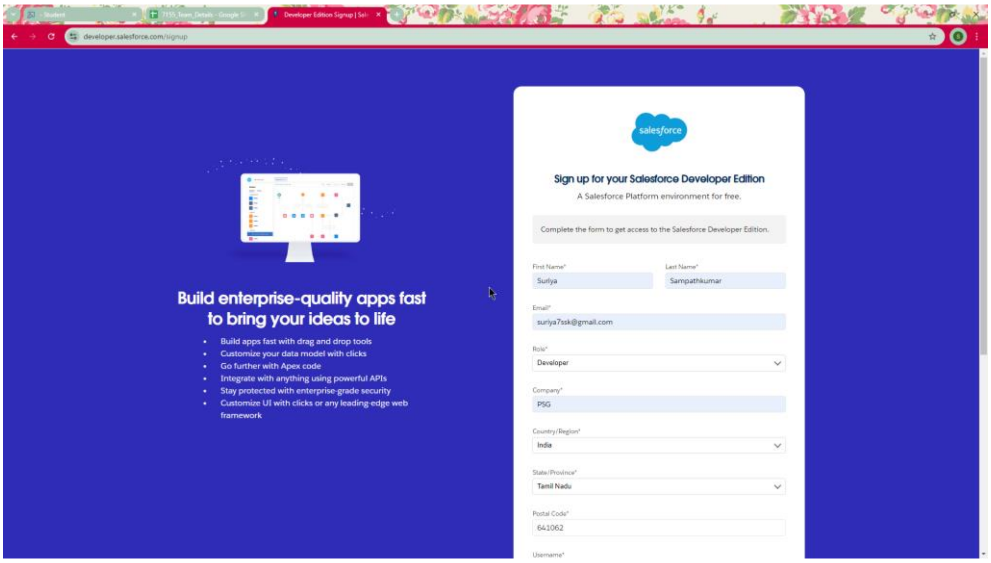
3. Salesforce Key Features and Concepts Utilized

1. **Custom Objects**: Custom objects have been created for Hotels, Food Options, Customers, and Flights, each with specific fields that support business processes.
2. **Custom Fields**: Fields such as total food options, food amount, and departure times have been added to store and track relevant data for each object.
3. **Flows**: Flows are used to automate the discount process, triggering different actions based on the customer's total purchase amount.
4. **Apex Triggers**: Triggers are used to update hotel information when food options are modified, ensuring that the hotel's data stays consistent and up-to-date.
5. **Apex Schedulable Classes**: A scheduled class sends timely flight reminders to customers 24 hours before departure, improving customer engagement and reducing the likelihood of missed flights.

4. Detailed Steps to Solution Design

Step 1: Setting Up the Salesforce Developer Account

Created a Salesforce Developer Account to access the development tools necessary for building and testing the application. This provided a controlled environment for implementing and refining features.



developer.salesforce.com/signup

Build enterprise-quality apps fast to bring your ideas to life

- Build apps fast with drag and drop tools
- Customize your data model with clicks
- Go further with Apex code
- Integrate with anything using powerful APIs
- Stay protected with enterprise-grade security
- Customize UI with clicks or any leading-edge web framework

Sign up for your Salesforce Developer Edition

A Salesforce Platform environment for free.

Complete the form to get access to the Salesforce Developer Edition.

First Name*: Suriya

Last Name*: Sampathkumar

Email*: suriya7ssk@gmail.com

Role*: Developer

Company*: PSG

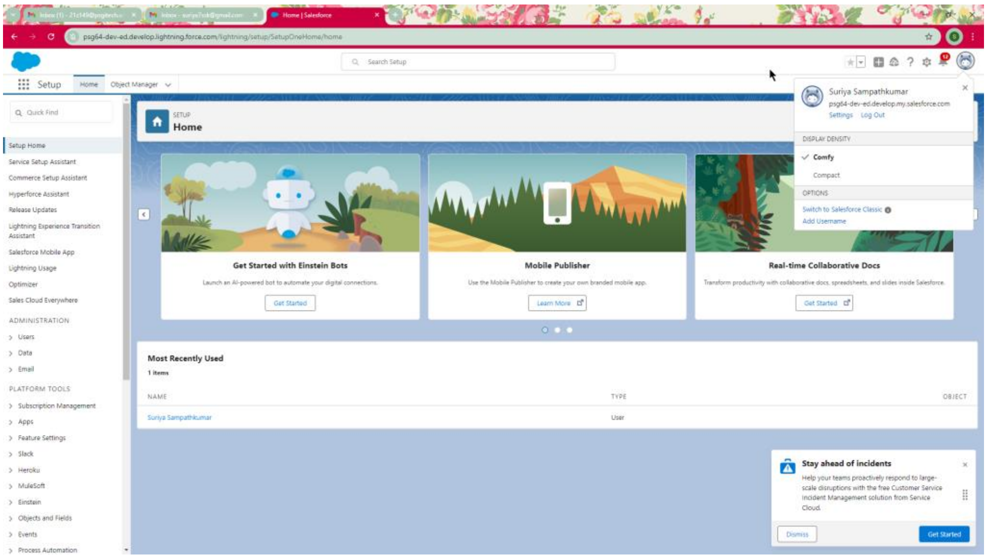
Country/Region*: India

State/Province*: Tamil Nadu

Postal Code*: 641062

Username*

Creating developer account



Get Started with Einstein Bots

Launch an AI-powered bot to automate your digital connections.

Mobile Publisher

Use the Mobile Publisher to create your own branded mobile app.

Real-time Collaborative Docs

Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce.

Most Recently Used

1 items

NAME	TYPE	OBJECT
Suriya Sampathkumar	User	

Stay ahead of incidents

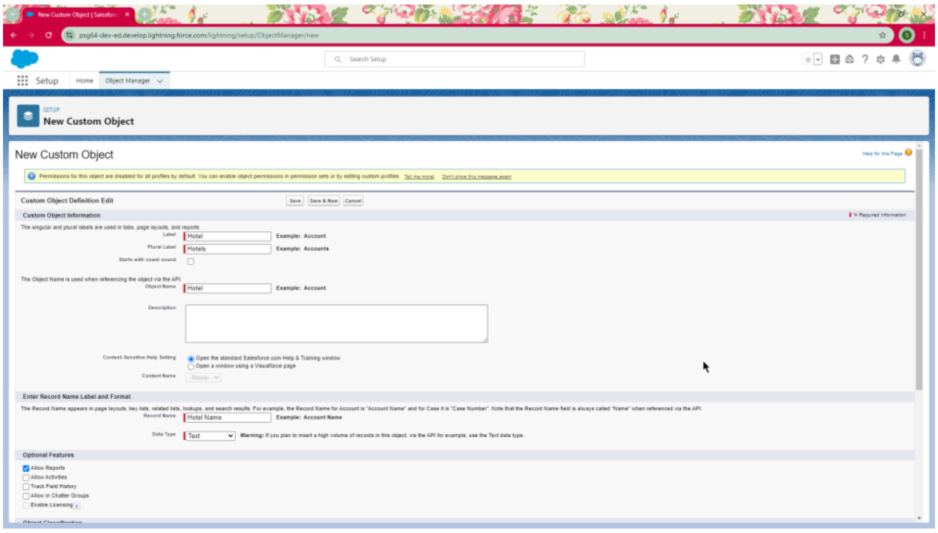
Help your teams proactively respond to large-scale disruptions with the free Customer Service Incident Management solution from Service Cloud.

Account activation

Step 2: Defining Custom Objects

Hotel Object:

- **Fields**: Hotel Name (Text), Total Food Options (Number).
- **Purpose**: To store information about hotels and track the availability of meal options.



New Custom Object | Salesforce
psg64-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/new
Search Setup
Setup
Home
Object Manager
SETUP
New Custom Object
Help for this Page
Save
Save & New
Cancel

New Custom Object

Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. Tell me more! Don't show this message again

Custom Object Definition Edit

Custom Object Information

= Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label: Hotel Example: Account

Plural Label: Hotels Example: Accounts

Starts with vowel sound: □

The Object Name is used when referencing the object via the API.

Object Name: Hotel Example: Account

Description:

Content-Sensitive Help Setting: ● Open the standard Salesforce.com Help & Training window
○ Open a window using a Visualforce page

Content Name: --None--

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name: Hotel Name Example: Account Name

Data Type: Text Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

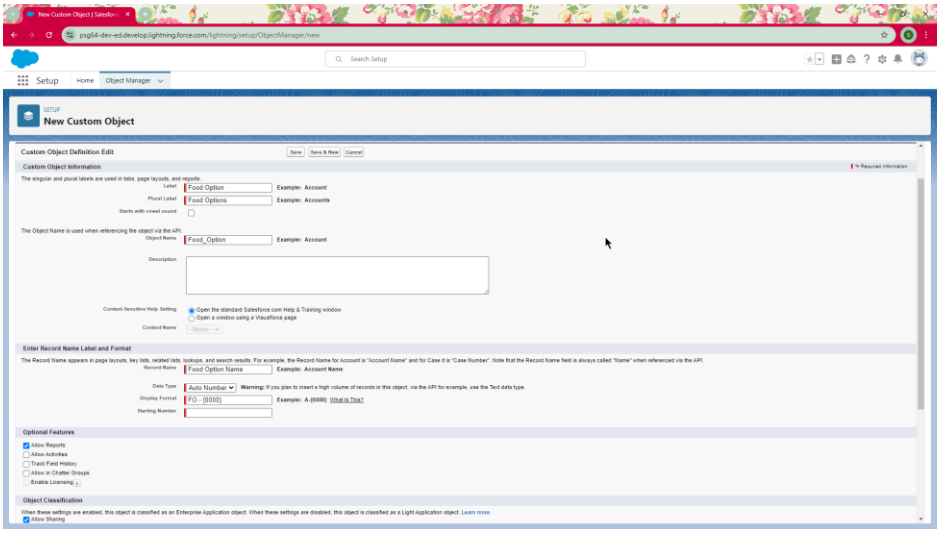
Optional Features

☑ Allow Reports
□ Allow Activities
□ Track Field History
□ Allow in Chatter Groups
□ Enable Licensing

Creating object Hotel

Food Option Object:

- **Fields**: Name (Text), Hotel (Lookup), Food Amount (Currency).
- **Purpose**: To catalog food offerings associated with hotels.


New Custom Object | Salesfor
psg64-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/new
Search Setup
Setup
Home
Object Manager
SETUP
Save
Save & New
Cancel

New Custom Object

Custom Object Definition Edit

Custom Object Information — | = Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label: Food Option — Example: Account

Plural Label: Food Options — Example: Accounts

Starts with vowel sound: □

The Object Name is used when referencing the object via the API.

Object Name: Food_Option — Example: Account

Description:

Context-Sensitive Help Setting: ● Open the standard Salesforce.com Help & Training window ○ Open a window using a Visualforce page

Content Name: --None--

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name: Food Option Name — Example: Account Name

Data Type: Auto Number — **Warning:** If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Display Format: FO - {0000} — Example: A-{0000} What Is This?

Starting Number:

Optional Features

☑ Allow Reports
□ Allow Activities
□ Track Field History
□ Allow in Chatter Groups
□ Enable Licensing

Object Classification

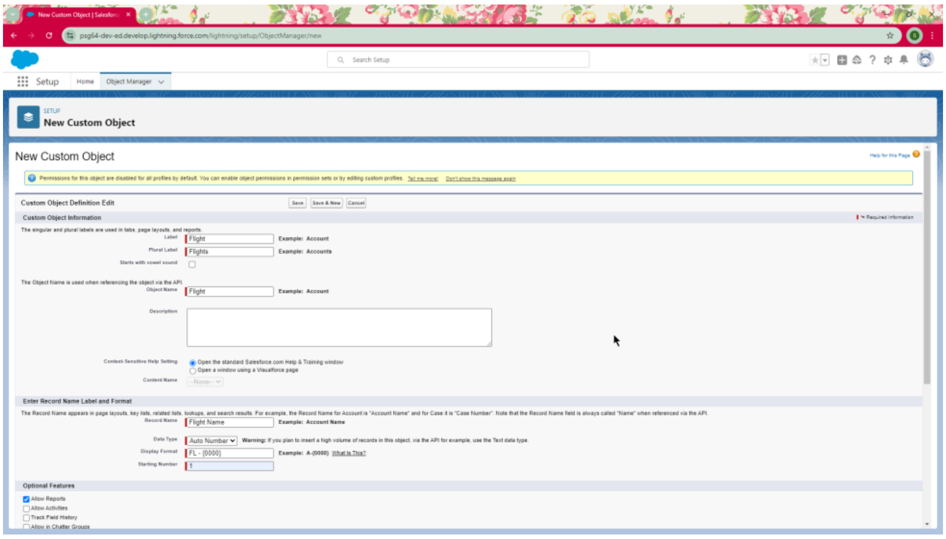
When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. Learn more.

☑ Allow Sharing

Creating object Food Option

Flight Object:

- **Fields**: DepartureDateTime (Date/Time), ContactEmail (Text).
- **Purpose**: To manage flight schedules and send reminders.


New Custom Object | Salesfor
psg64-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/new
Search Setup
Setup
Home
Object Manager
SETUP
Help for this Page
Save
Save & New
Cancel

New Custom Object

Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. Tell me more! Don't show this message again

Custom Object Definition Edit

Custom Object Information — | = Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label: Flight — Example: Account

Plural Label: Flights — Example: Accounts

Starts with vowel sound: □

The Object Name is used when referencing the object via the API.

Object Name: Flight — Example: Account

Description:

Context-Sensitive Help Setting: ● Open the standard Salesforce.com Help & Training window ○ Open a window using a Visualforce page

Content Name: --None--

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name: Flight Name — Example: Account Name

Data Type: Auto Number — **Warning:** If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Display Format: FL - {0000} — Example: A-{0000} What Is This?

Starting Number: 1

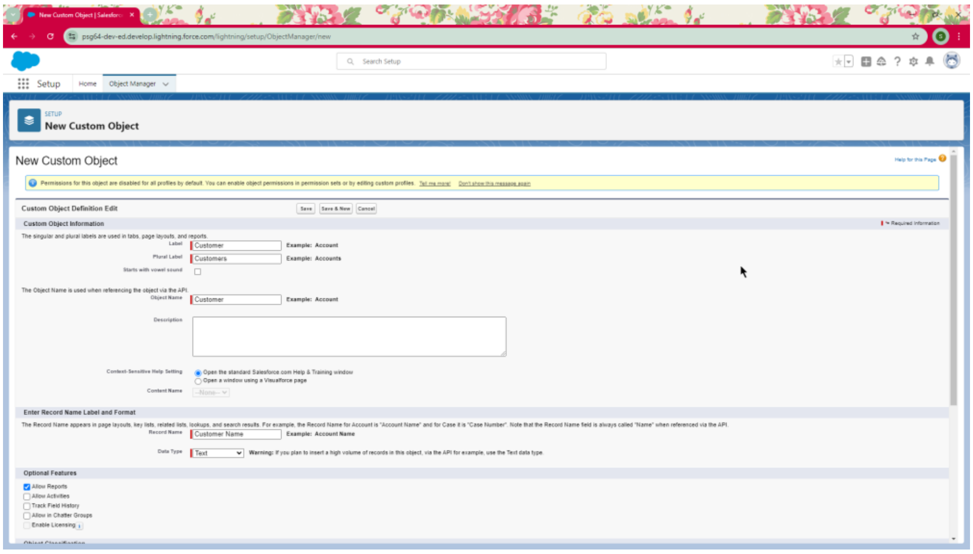
Optional Features

☑ Allow Reports
□ Allow Activities
□ Track Field History
□ Allow in Chatter Groups

Creating object Flight

Customer Object:

- **Fields**: Name (Text), Total Spending (Currency), Discount (Percentage).
- **Purpose**: To track customer interactions and calculate eligible discounts.


New Custom Object | Salesforce
psg64-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/new
Search Setup
Setup
Home
Object Manager
SETUP
Save
Save & New
Cancel
Help for this Page
Tell me more!
Don't show this message again

New Custom Object

New Custom Object

Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles.

Custom Object Definition Edit

Custom Object Information

= Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label: Customer — Example: Account

Plural Label: Customers — Example: Accounts

Starts with vowel sound: □

The Object Name is used when referencing the object via the API.

Object Name: Customer — Example: Account

Description:

Context-Sensitive Help Setting: ● Open the standard Salesforce.com Help & Training window ○ Open a window using a Visualforce page

Content Name: --None--

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name: Customer Name — Example: Account Name

Data Type: Text — **Warning:** If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Optional Features

☑ Allow Reports
□ Allow Activities
□ Track Field History
□ Allow in Chatter Groups
□ Enable Licensing

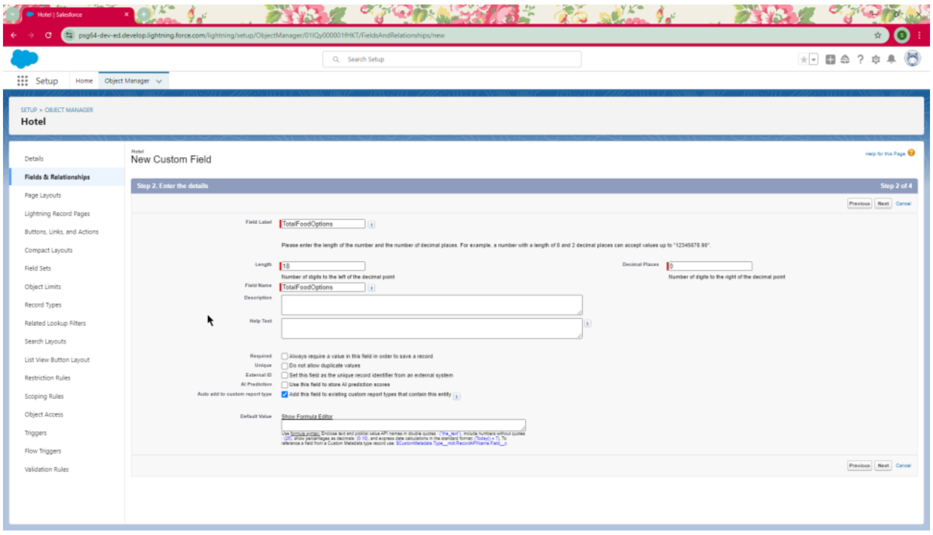
Object Classification

Creating object Customer

Step 3: Field Definitions

Hotel Object Fields:

1. **TotalFoodOptions** (Number): Tracks the total number of food options available at each hotel.
2. **Date** (Date): The date of hotel information entry or update.



Hotel

Hotel

New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: TotalFoodOptions

Please enter the length of the number and the number of decimal places. For example, a number with a length of 8 and 2 decimal places can accept values up to "12345678.90".

Length: 18 — Number of digits to the left of the decimal point

Decimal Places: 0 — Number of digits to the right of the decimal point

Field Name: TotalFoodOptions

Description:

Help Text:

Required: □ Always require a value in this field in order to save a record

Unique: □ Do not allow duplicate values

External ID: □ Set this field as the unique record identifier from an external system

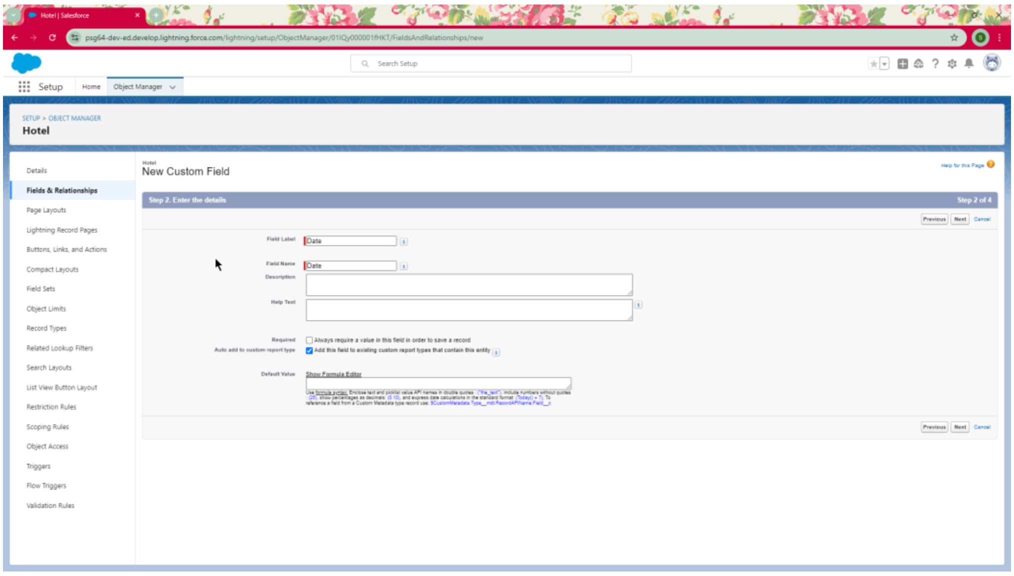
AI Prediction: □ Use this field to store AI prediction scores

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Default Value: Show Formula Editor

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (25), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today() + 7). To reference a field from a Custom Metadata type record use: $CustomMetadata.Type__mdt.RecordAPIName.Field__c

Creating field TotalFoodOptions



Hotel
New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: Date

Field Name: Date

Description:

Help Text:

Required: □ Always require a value in this field in order to save a record

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

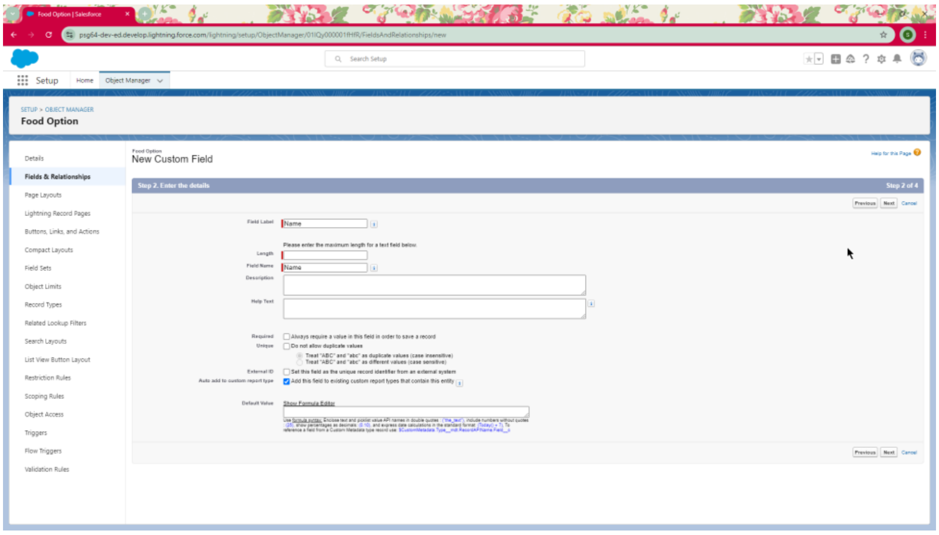
Default Value:

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (25), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today() + 7). To reference a field from a Custom Metadata type record use: $CustomMetadata.Type__mdt.RecordAPIName.Field__c

Creating field Date

Food Option Fields:

1. **Name (Text):** The name of the food option.
2. **Hotel (Lookup):** A lookup field linking each food option to the respective hotel.
3. **Food Amount (Currency):** The price of the food option.



Food Option
New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: Name

Please enter the maximum length for a text field below.

Length:

Field Name: Name

Description:

Help Text:

Required: □ Always require a value in this field in order to save a record

Unique: □ Do not allow duplicate values

○ Treat "ABC" and "abc" as duplicate values (case insensitive)

○ Treat "ABC" and "abc" as different values (case sensitive)

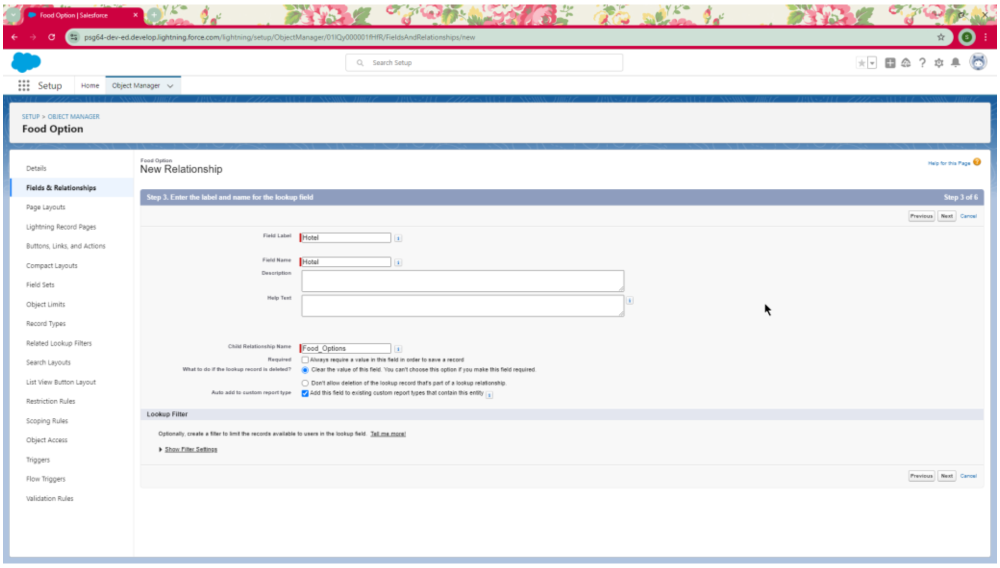
External ID: □ Set this field as the unique record identifier from an external system

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Default Value:

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (25), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today() + 7). To reference a field from a Custom Metadata type record use: $CustomMetadata.Type__mdt.RecordAPIName.Field__c

Creating field Name



Food Option

New Relationship

Step 3. Enter the label and name for the lookup field — Step 3 of 6

Field Label: Hotel

Field Name: Hotel

Description:

Help Text:

Child Relationship Name: Food_Options

Required: □ Always require a value in this field in order to save a record

What to do if the lookup record is deleted? ● Clear the value of this field. You can't choose this option if you make this field required.

○ Don't allow deletion of the lookup record that's part of a lookup relationship.

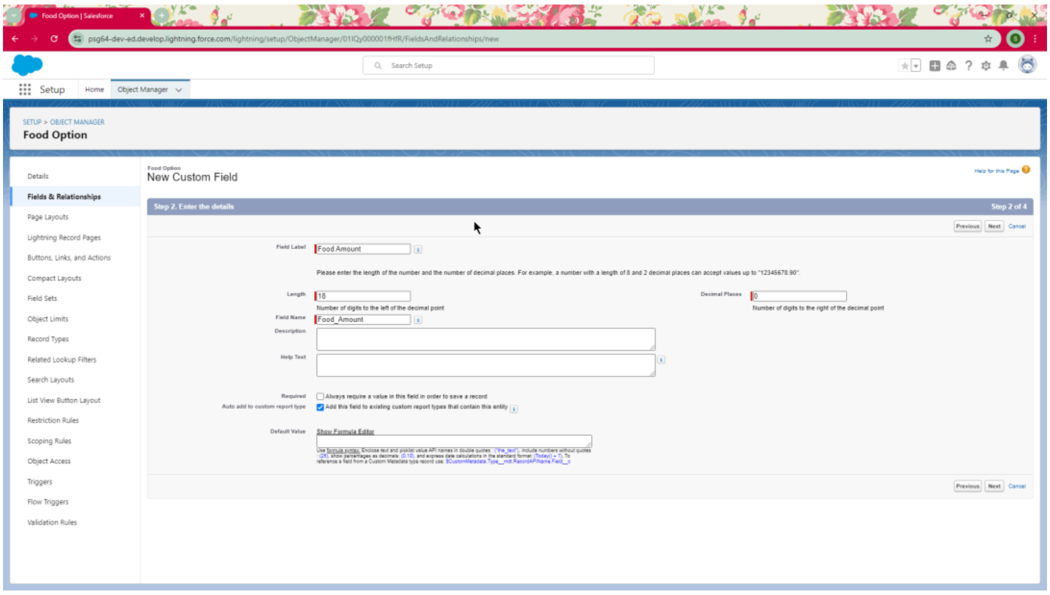
Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Lookup Filter

Optionally, create a filter to limit the records available to users in the lookup field. Tell me more!

Show Filter Settings

Creating field Hotel



Food Option

New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: Food Amount

Please enter the length of the number and the number of decimal places. For example, a number with a length of 8 and 2 decimal places can accept values up to "12345678.90".

Length: 18 — Number of digits to the left of the decimal point

Decimal Places: 0 — Number of digits to the right of the decimal point

Field Name: Food_Amount

Description:

Help Text:

Required: □ Always require a value in this field in order to save a record

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

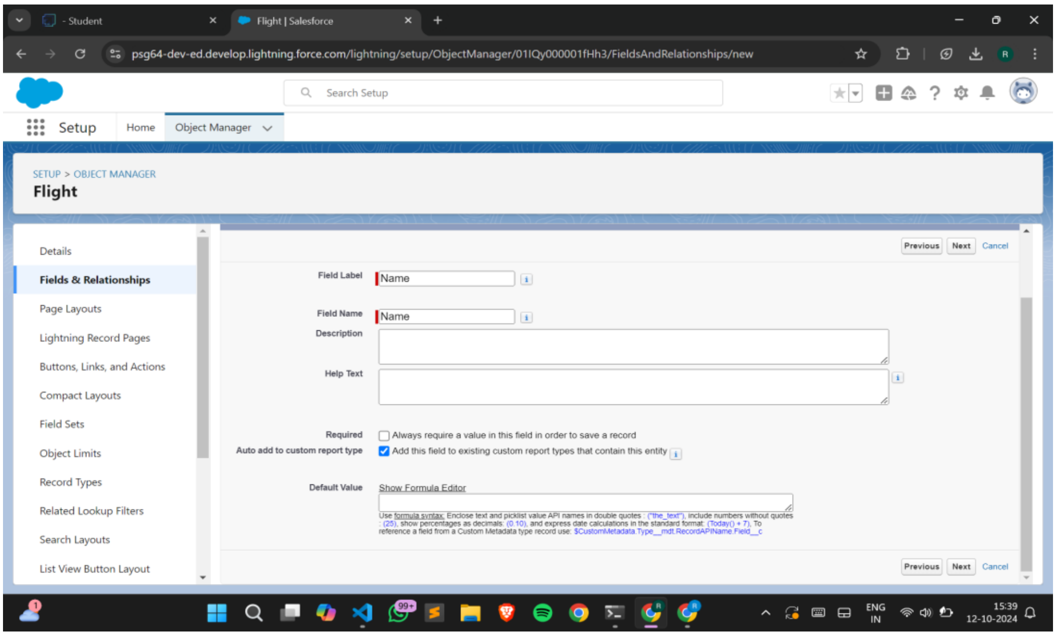
Default Value: Show Formula Editor

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (25), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today() + 7). To reference a field from a Custom Metadata type record use: $CustomMetadata.Type__mdt.RecordAPIName.Field__c

Creating field Food Amount

Flight Object Fields:

1. **Name (Text):** A name or code identifying the flight.
2. **DepartureDateTime (Date/Time):** The date and time of flight departure.



Field Label: Name

Field Name: Name

Description:

Help Text:

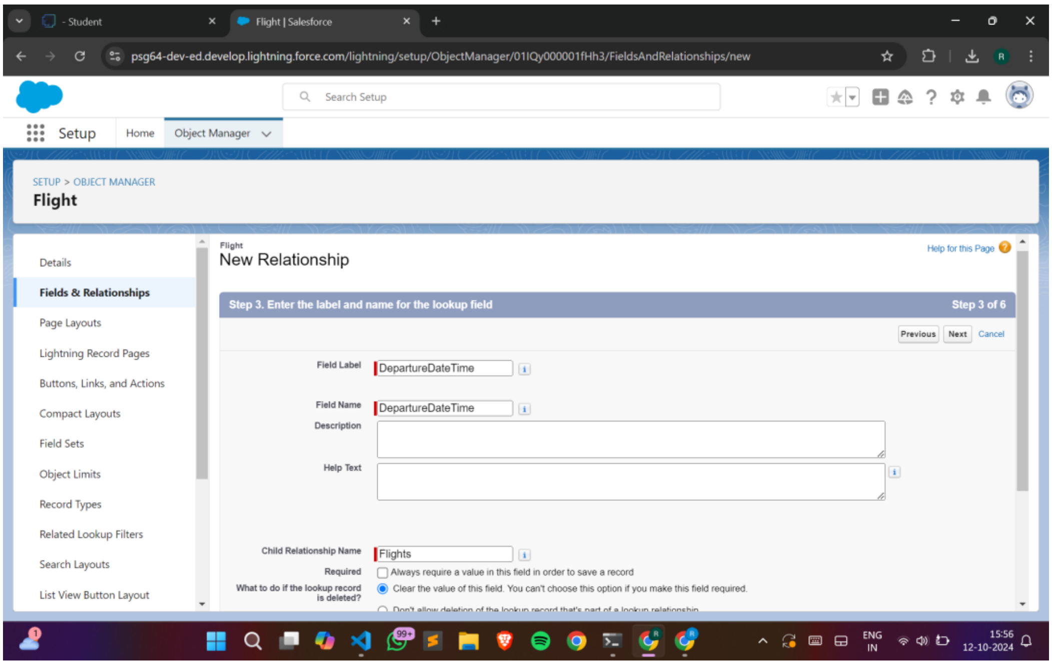
Required: □ Always require a value in this field in order to save a record

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Default Value:

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (25), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today() + 7). To reference a field from a Custom Metadata type record use: $CustomMetadata.Type__mdt.RecordAPIName.Field__c

Creating field Name



Flight

New Relationship

Step 3. Enter the label and name for the lookup field — Step 3 of 6

Field Label: DepartureDateTime

Field Name: DepartureDateTime

Description:

Help Text:

Child Relationship Name: Flights

Required: □ Always require a value in this field in order to save a record

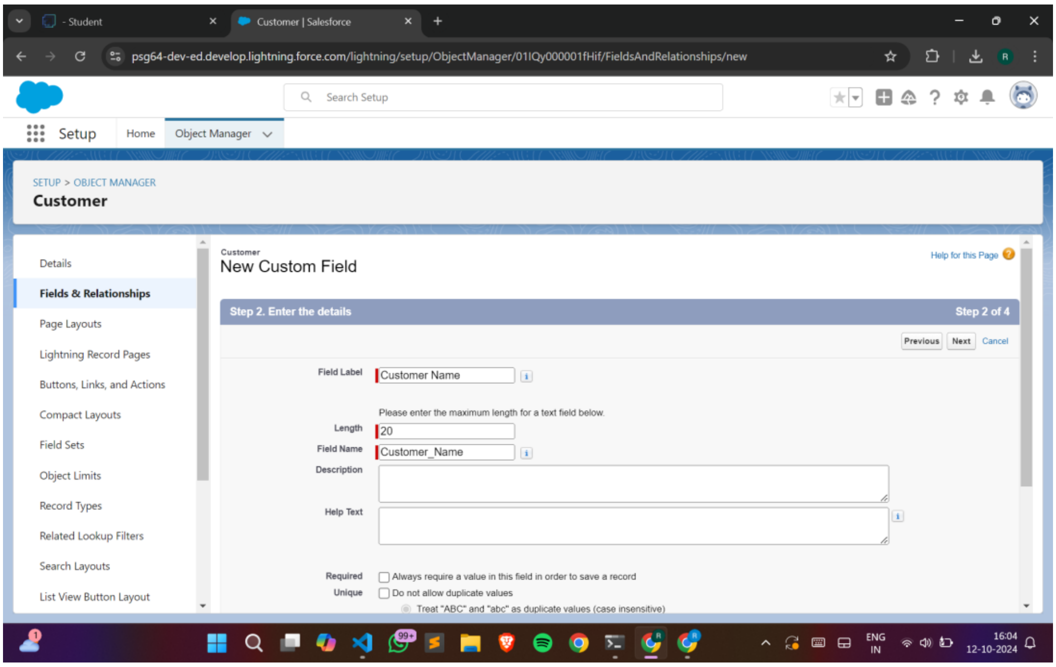
What to do if the lookup record is deleted?: ● Clear the value of this field. You can't choose this option if you make this field required.

○ Don't allow deletion of the lookup record that's part of a lookup relationship.

Creating field DepartureDateTime

Customer Object Fields:

1. **Customer Name (Text):** The name of the customer.
2. **Discount Amount (Formula - Currency):** A formula that calculates the discount based on customer criteria.
3. **Discount Percent (Percentage):** The percentage discount applied to the customer.



Customer

New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: Customer Name

Please enter the maximum length for a text field below.

Length: 20

Field Name: Customer_Name

Description:

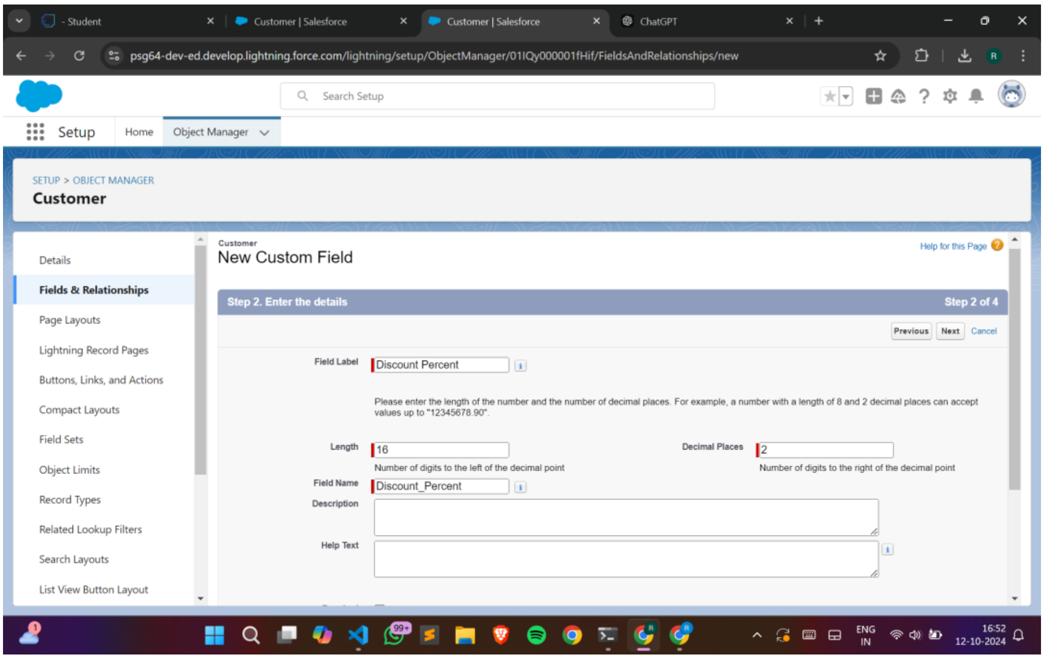
Help Text:

Required: □ Always require a value in this field in order to save a record

Unique: □ Do not allow duplicate values

○ Treat "ABC" and "abc" as duplicate values (case insensitive)

Creating field Customer Name



Customer

New Custom Field

Step 2. Enter the details — Step 2 of 4

Field Label: Discount Percent

Please enter the length of the number and the number of decimal places. For example, a number with a length of 8 and 2 decimal places can accept values up to "12345678.90".

Length: 16 — Number of digits to the left of the decimal point

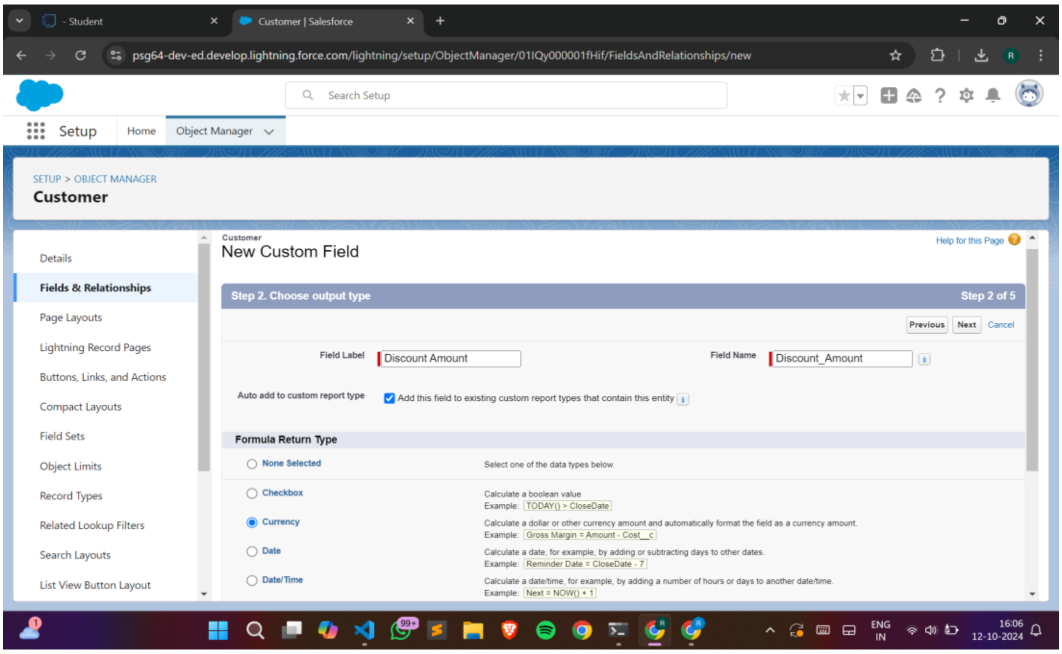
Decimal Places: 2 — Number of digits to the right of the decimal point

Field Name: Discount_Percent

Description:

Help Text:

Creating field Discount Percent



Customer

New Custom Field

Step 2. Choose output type — Step 2 of 5

Field Label: Discount Amount

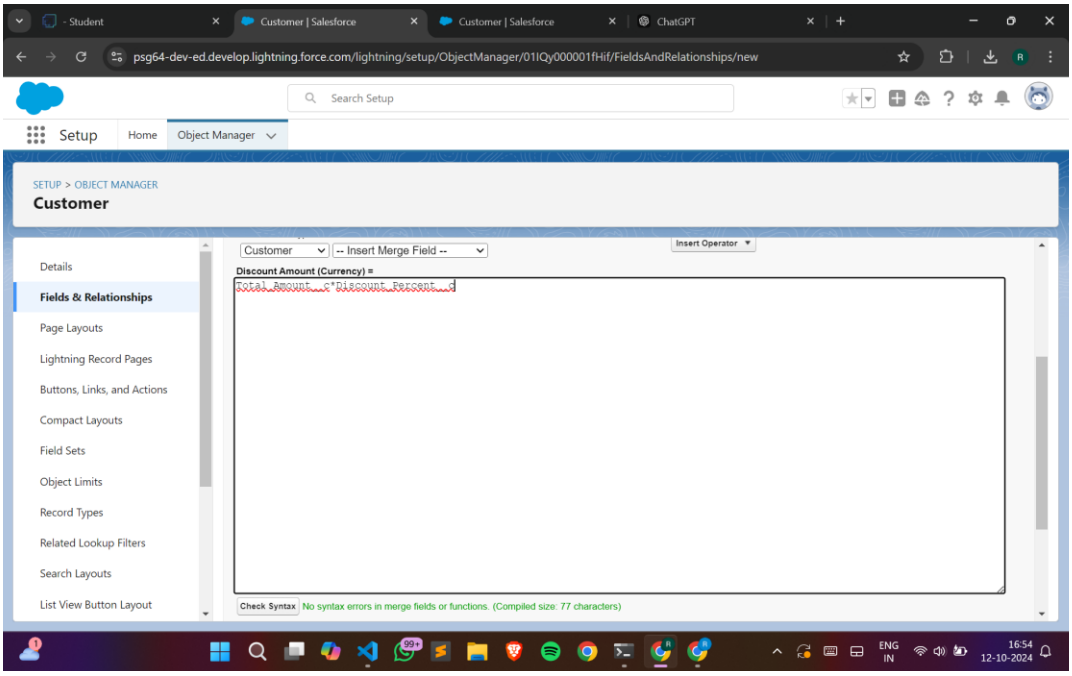
Field Name: Discount_Amount

Auto add to custom report type: ☑ Add this field to existing custom report types that contain this entity

Formula Return Type

○ None Selected	Select one of the data types below.
○ Checkbox	Calculate a boolean value Example: TODAY() > CloseDate
● Currency	Calculate a dollar or other currency amount and automatically format the field as a currency amount. Example: Gross Margin = Amount - Cost__c
○ Date	Calculate a date, for example, by adding or subtracting days to other dates. Example: Reminder Date = CloseDate - 7
○ Date/Time	Calculate a date/time, for example, by adding a number of hours or days to another date/time. Example: Next = NOW() + 1

Creating field Discount Amount



Discount Amount (Currency) =

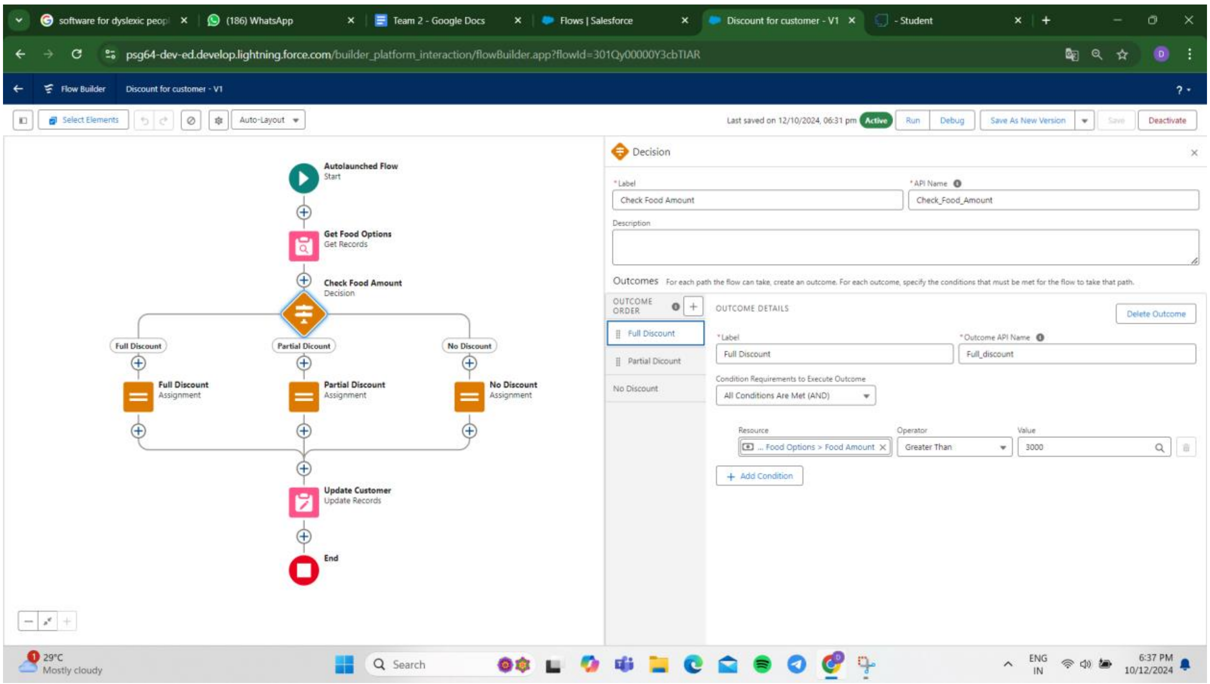
```
Total_Amount__c*Discount_Percent__c
```

No syntax errors in merge fields or functions. (Compiled size: 77 characters)

Creating field Discount Amount

Step 4: Create flow for Customer Discounts

- **Variables:**
  - **foId (Text) -** Available for Input
  - **csId (Text) -** Available for Input
  - **discount (Number)**
- **Flow Steps:**
  - **Get Records:** Retrieve customer data based on purchase histo**ry.**
  - **Decision Element: Add decision outcomes for different discoun**t thresholds:
    - **Full Discount:** For customers with amounts greater than 3000.
    - **Partial Discount:** For customers with amounts between 1500 and 3000.
    - **No Discount:** For customers below 1500.
  - **Assignments:** Assign appropriate discounts based on decisions made above.
- **Flow Outcome:**
  - Apply the correct discount to the customer's account based on the flow's logic.



Decision

*Label: Check Food Amount

*API Name: Check_Food_Amount

Description:

Outcomes For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.

OUTCOME ORDER: Full Discount, Partial Dicount, No Discount

OUTCOME DETAILS

Delete Outcome

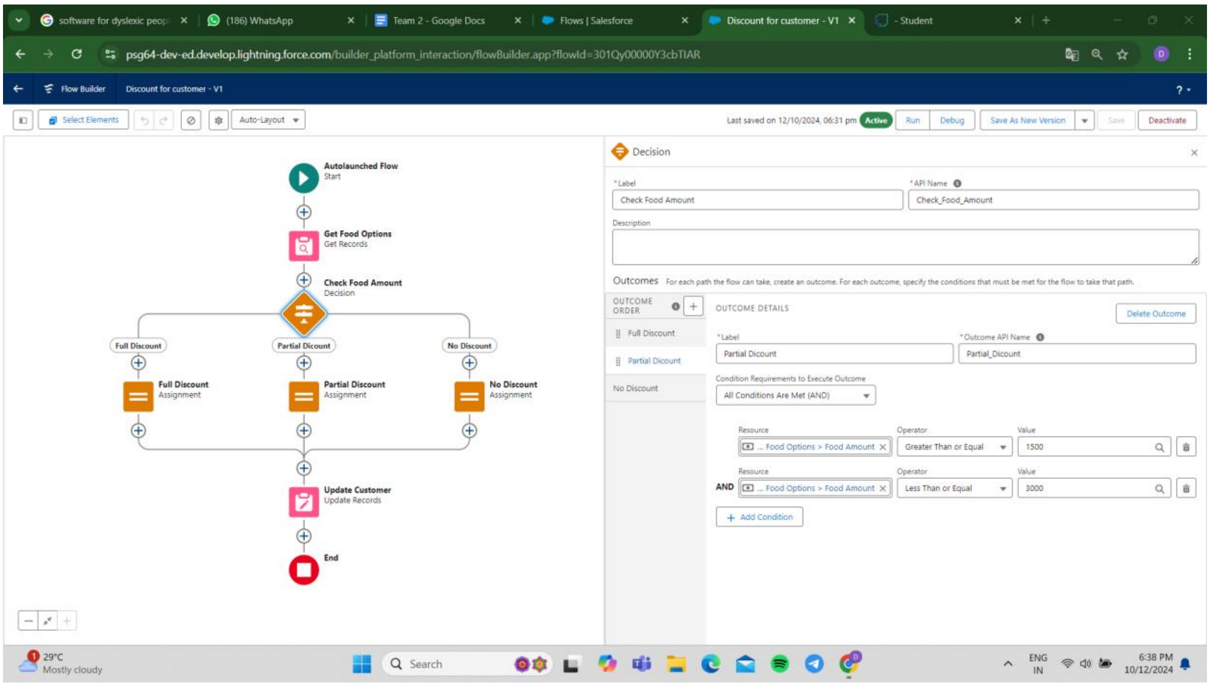
*Label: Full Discount

*Outcome API Name: Full_discount

Condition Requirements to Execute Outcome: All Conditions Are Met (AND)

Resource	Operator	Value
... Food Options > Food Amount	Greater Than	3000

+ Add Condition



Decision

*Label: Check Food Amount

*API Name: Check_Food_Amount

Description:

Outcomes For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.

OUTCOME ORDER

- Full Discount
- Partial Dicount
- No Discount

OUTCOME DETAILS

*Label: Partial Dicount

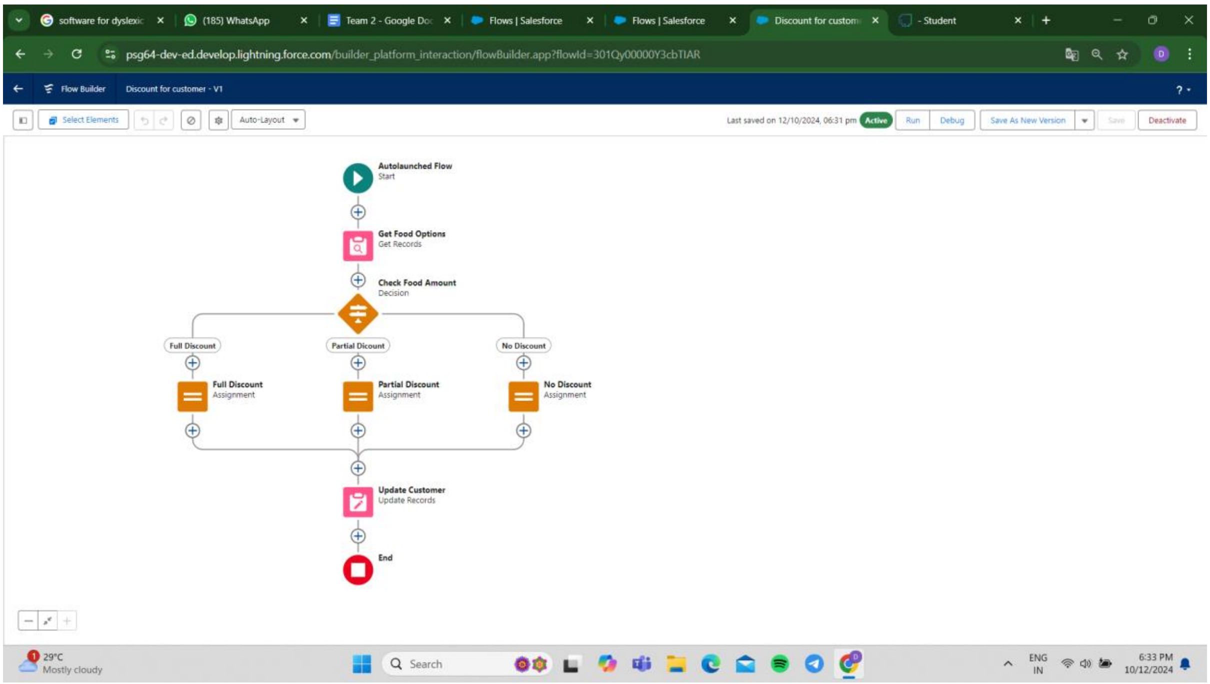
*Outcome API Name: Partial_Dicount

Condition Requirements to Execute Outcome: All Conditions Are Met (AND)

	Resource	Operator	Value
	... Food Options > Food Amount	Greater Than or Equal	1500
AND	... Food Options > Food Amount	Less Than or Equal	3000

Last saved on 12/10/2024, 06:31 pm Active

Condition for full discount and partial discount



Flow created for discount approval

Step 5: Apex Trigger and Handler for Updating Hotel Food Option Counts

Variables:

- newFoodOptions (List of Food_Option__c) - List of new or updated food options passed to the handler.
- oldFoodOptions (List of Food_Option__c) - List of old food options (used only if required for comparisons).
- operation (TriggerOperation) - Type of trigger operation (e.g., insert, update, delete).
- hotelIdsToUpdate (Set of Id) - Set of unique Hotel IDs affected by the changes to the food options.
- hotelsToUpdate (List of Hotel__c) - List of hotel records that need to be updated.

Apex Trigger Steps:

1. Trigger Setup:
   - Define a trigger on the Food_Option__c object.
   - The trigger is set to run after insert, after update, and after delete to handle changes in food options.

2. Trigger Conditions:
   - Check if the trigger is running after insert to ensure updates happen after new food options are created.

3. Call Handler Method:
   - If the trigger is after insert, invoke updateHotelInformation from FoodOptionTriggerHandler to update hotel details based on the new food options.
   - Pass trigger.new to the handler to retrieve the list of newly inserted food options.

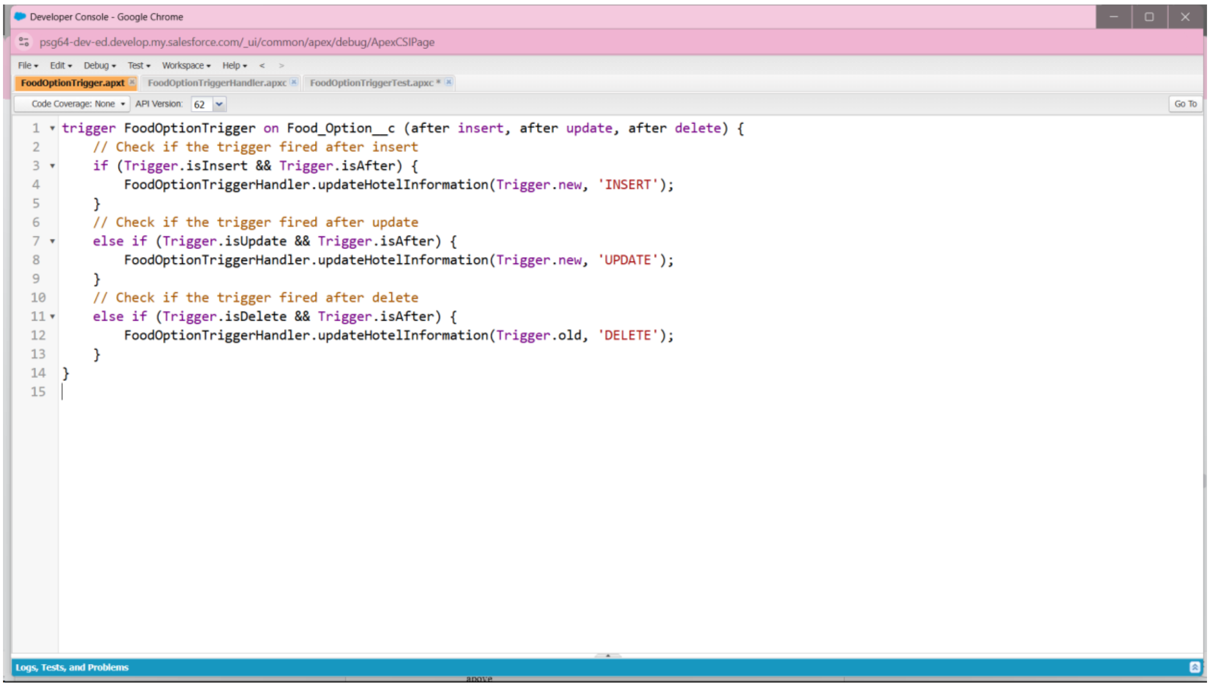
Apex Handler Steps (updateHotelInformation method):

1. Collect Hotel IDs:
   - Loop through each record in newFoodOptions.
   - Add the Hotel__c (Hotel ID) from each food option to hotelIdsToUpdate to collect unique hotel IDs affected by the changes.

2. Retrieve Hotels to Update:
   - Query the Hotel__c object for records with IDs in hotelIdsToUpdate.
   - Retrieve fields Id, Name, and TotalFoodOptions__c to calculate and update the total food options for each hotel.

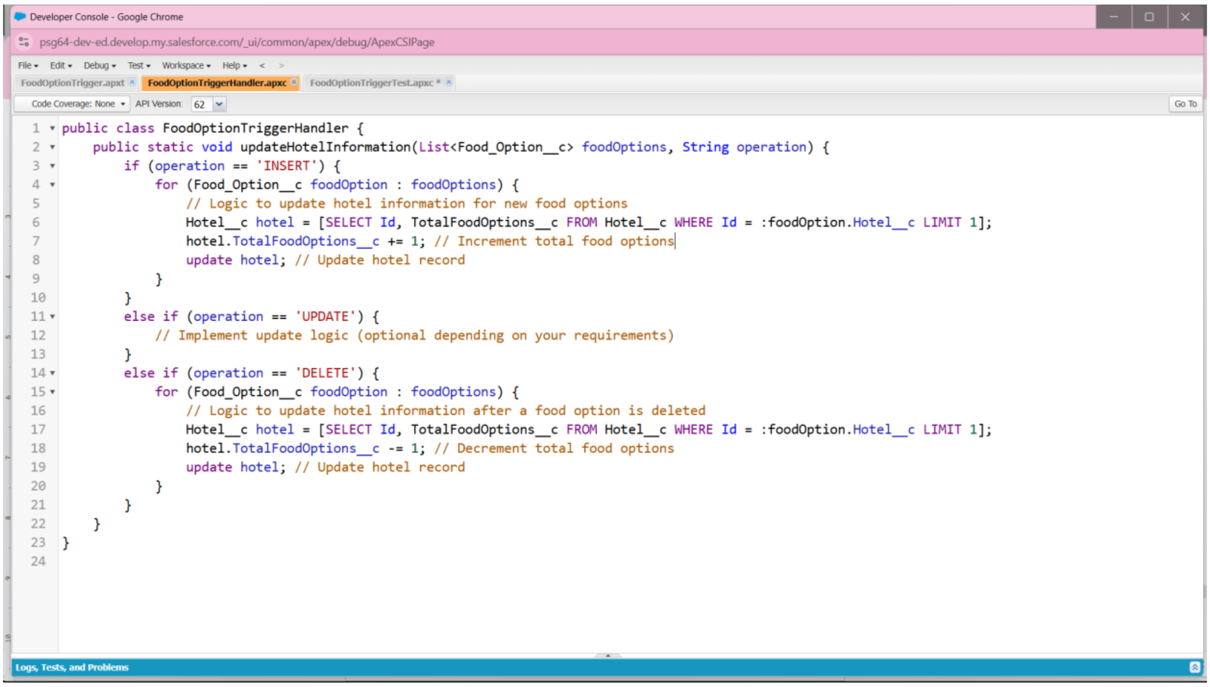
3. Recalculate Total Food Options:
   - Loop through each hotel record in hotelsToUpdate.
   - Run a query to count the total food options linked to each hotel using COUNT().
   - Assign the result to the TotalFoodOptions__c field on the Hotel__c record.

4. Update Hotel Records:
   - Perform an update on hotelsToUpdate to save the recalculated TotalFoodOptions__c values.


Developer Console - Google Chrome
psg64-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File
Edit
Debug
Test
Workspace
Help
FoodOptionTrigger.apxt
FoodOptionTriggerHandler.apxc
FoodOptionTriggerTest.apxc *
Code Coverage: None
API Version: 62
Go To
Logs, Tests, and Problems

```
trigger FoodOptionTrigger on Food_Option__c (after insert, after update, after delete) {
    // Check if the trigger fired after insert
    if (Trigger.isInsert && Trigger.isAfter) {
        FoodOptionTriggerHandler.updateHotelInformation(Trigger.new, 'INSERT');
    }
    // Check if the trigger fired after update
    else if (Trigger.isUpdate && Trigger.isAfter) {
        FoodOptionTriggerHandler.updateHotelInformation(Trigger.new, 'UPDATE');
    }
    // Check if the trigger fired after delete
    else if (Trigger.isDelete && Trigger.isAfter) {
        FoodOptionTriggerHandler.updateHotelInformation(Trigger.old, 'DELETE');
    }
}
```

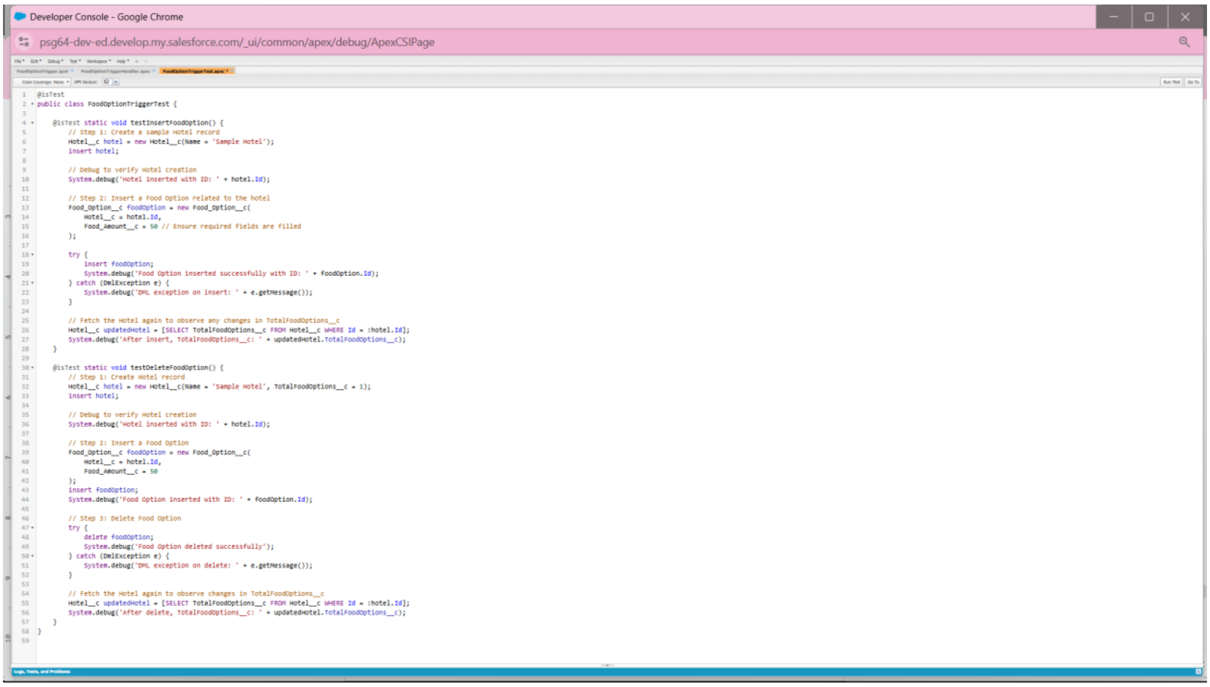
Trigger Code (FoodOptionTrigger)



Developer Console - Google Chrome
psg64-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
FoodOptionTrigger.apxt
FoodOptionTriggerHandler.apxc
FoodOptionTriggerTest.apxc *
Code Coverage: None
API Version: 62
Go To
Logs, Tests, and Problems

```
public class FoodOptionTriggerHandler {
    public static void updateHotelInformation(List<Food_Option__c> foodOptions, String operation) {
        if (operation == 'INSERT') {
            for (Food_Option__c foodOption : foodOptions) {
                // Logic to update hotel information for new food options
                Hotel__c hotel = [SELECT Id, TotalFoodOptions__c FROM Hotel__c WHERE Id = :foodOption.Hotel__c LIMIT 1];
                hotel.TotalFoodOptions__c += 1; // Increment total food options
                update hotel; // Update hotel record
            }
        }
        else if (operation == 'UPDATE') {
            // Implement update logic (optional depending on your requirements)
        }
        else if (operation == 'DELETE') {
            for (Food_Option__c foodOption : foodOptions) {
                // Logic to update hotel information after a food option is deleted
                Hotel__c hotel = [SELECT Id, TotalFoodOptions__c FROM Hotel__c WHERE Id = :foodOption.Hotel__c LIMIT 1];
                hotel.TotalFoodOptions__c -= 1; // Decrement total food options
                update hotel; // Update hotel record
            }
        }
    }
}
```

Handler Class Code (FoodOptionTriggerHandler)



Developer Console - Google Chrome
psg64-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
FoodOptionTrigger.apxt
FoodOptionTriggerHandler.apxc
FoodOptionTriggerTest.apxc *
Code Coverage: None
API Version: 62
Logs, Tests, and Problems

```
@isTest
public class FoodOptionTriggerTest {

    @isTest static void testInsertFoodOption() {
        // Step 1: Create a sample Hotel record
        Hotel__c hotel = new Hotel__c(Name = 'Sample Hotel');
        insert hotel;

        // Debug to verify Hotel creation
        System.debug('Hotel inserted with ID: ' + hotel.Id);

        // Step 2: Insert a Food Option related to the hotel
        Food_Option__c foodOption = new Food_Option__c(
            Hotel__c = hotel.Id,
            Food_Amount__c = 50 // Ensure required fields are filled
        );

        try {
            insert foodOption;
            System.debug('Food Option inserted successfully with ID: ' + foodOption.Id);
        } catch (DmlException e) {
            System.debug('DML exception on insert: ' + e.getMessage());
        }

        // Fetch the Hotel again to observe any changes in TotalFoodOptions__c
        Hotel__c updatedHotel = [SELECT TotalFoodOptions__c FROM Hotel__c WHERE Id = :hotel.Id];
        System.debug('After insert, TotalFoodOptions__c: ' + updatedHotel.TotalFoodOptions__c);
    }

    @isTest static void testDeleteFoodOption() {
        // Step 1: Create Hotel record
        Hotel__c hotel = new Hotel__c(Name = 'Sample Hotel', TotalFoodOptions__c = 1);
        insert hotel;

        // Debug to verify Hotel creation
        System.debug('Hotel inserted with ID: ' + hotel.Id);

        // Step 2: Insert a Food Option
        Food_Option__c foodOption = new Food_Option__c(
            Hotel__c = hotel.Id,
            Food_Amount__c = 50
        );
        insert foodOption;
        System.debug('Food Option inserted with ID: ' + foodOption.Id);

        // Step 3: Delete Food Option
        try {
            delete foodOption;
            System.debug('Food Option deleted successfully');
        } catch (DmlException e) {
            System.debug('DML exception on delete: ' + e.getMessage());
        }

        // Fetch the Hotel again to observe changes in TotalFoodOptions__c
        Hotel__c updatedHotel = [SELECT TotalFoodOptions__c FROM Hotel__c WHERE Id = :hotel.Id];
        System.debug('After delete, TotalFoodOptions__c: ' + updatedHotel.TotalFoodOptions__c);
    }
}
```

Test Code (FoodOptionTriggerTest)

Step 6: Scheduled Apex Class for Automated Flight Reminder Emails

Variables:

- DepartureDateTime__c (DateTime) - Custom field on the Flight__c object representing the flight's departure date and time.
- ContactEmail__c (Email) - Custom field on the Flight__c object holding the customer's contact email address for sending reminders.

Apex Schedule Steps:

1. Define Apex Schedule Class:
    - Create a class named FlightReminderScheduledJob that implements the Schedulable interface, allowing it to be scheduled to run at specified times.

2. Schedulable Interface Implementation:
    - Implement the execute method from the Schedulable interface.
    - Inside execute, call the sendFlightReminders method to handle the logic for querying flights and sending reminders.

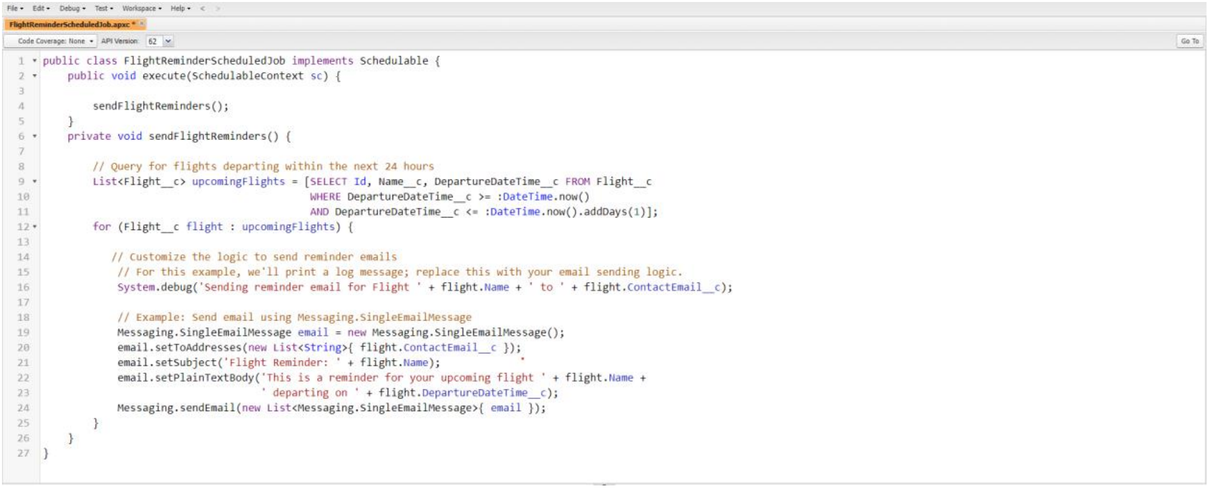
3. Query Upcoming Flights:
    - In the sendFlightReminders method, query the Flight__c records where DepartureDateTime__c is within the next 24 hours (>= DateTime.now() and <= DateTime.now().addDays(1)).
    - Retrieve fields Id, Name, DepartureDateTime__c, and ContactEmail__c.

4. Send Reminder Email:
    - For each flight in upcomingFlights, create an email message using Messaging.SingleEmailMessage.
    - Set the ToAddresses, Subject, and PlainTextBody of the email to include flight details and departure time.
    - Send the email using Messaging.sendEmail().

Anonymous Apex Code for Scheduling:

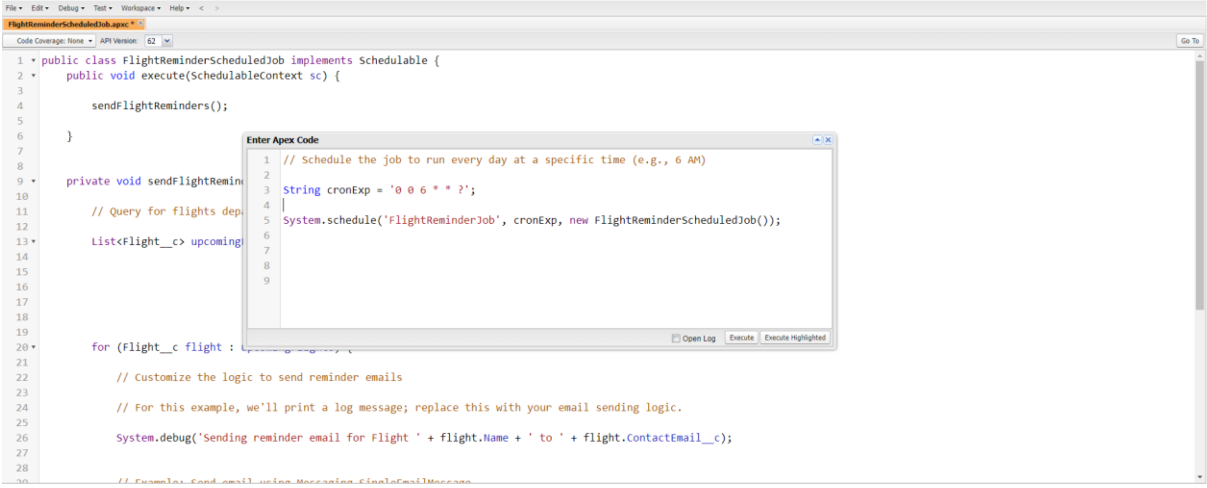
1. Define Cron Expression:
    - Set a cron expression to schedule the job daily at 6 AM.

2. Schedule the Job:
    - Use System.schedule to run FlightReminderScheduledJob with the cron expression.



```
public class FlightReminderScheduledJob implements Schedulable {
    public void execute(SchedulableContext sc) {

        sendFlightReminders();
    }
    private void sendFlightReminders() {

        // Query for flights departing within the next 24 hours
        List<Flight__c> upcomingFlights = [SELECT Id, Name__c, DepartureDateTime__c FROM Flight__c
                                           WHERE DepartureDateTime__c >= :DateTime.now()
                                           AND DepartureDateTime__c <= :DateTime.now().addDays(1)];
        for (Flight__c flight : upcomingFlights) {

           // Customize the logic to send reminder emails
            // For this example, we'll print a log message; replace this with your email sending logic.
            System.debug('Sending reminder email for Flight ' + flight.Name + ' to ' + flight.ContactEmail__c);

            // Example: Send email using Messaging.SingleEmailMessage
            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
            email.setToAddresses(new List<String>{ flight.ContactEmail__c });
            email.setSubject('Flight Reminder: ' + flight.Name);
            email.setPlainTextBody('This is a reminder for your upcoming flight ' + flight.Name +
                                  ' departing on ' + flight.DepartureDateTime__c);
            Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{ email });
        }
    }
}
```

Flight Reminder Schedule Class Code



```
public class FlightReminderScheduledJob implements Schedulable {
    public void execute(SchedulableContext sc) {

        sendFlightReminders();

    }


    private void sendFlightRemin

        // Query for flights dep

        List<Flight__c> upcoming






        for (Flight__c flight :

            // Customize the logic to send reminder emails

            // For this example, we'll print a log message; replace this with your email sending logic.

            System.debug('Sending reminder email for Flight ' + flight.Name + ' to ' + flight.ContactEmail__c);
```

```
// Schedule the job to run every day at a specific time (e.g., 6 AM)

String cronExp = '0 0 6 * * ?';

System.schedule('FlightReminderJob', cronExp, new FlightReminderScheduledJob());
```

Scheduling Job to run at Specific Time

5. Conclusion

Summary of Achievements:

The TripAdvisor E-Management Project successfully streamlined and automated various processes related to hotel management, flight bookings, food options, and customer notifications, significantly enhancing both operational workflows and the overall customer experience. Key accomplishments include:

- Optimized Operational Workflow: Implemented an integrated system for managing hotel requirements, food options, and flight bookings, ensuring seamless coordination and reducing the need for manual intervention in the TripAdvisor platform.
- Automated Customer Notifications: Developed a schedulable Apex class to send timely flight reminders to customers 24 hours before departure, improving customer service by ensuring customers are well-informed and prepared for their travels.
- Real-Time Data Synchronization: Designed and deployed Apex triggers and flows to automatically update and synchronize hotel and food option data in real-time, ensuring that customers have access to accurate and up-to-date information.
- Dynamic Discount Allocation: Created automated processes to apply personalized discounts for customers based on their purchasing behavior, offering a tailored experience that enhances customer satisfaction and loyalty.
- Improved Efficiency and Customer Experience: Leveraged Salesforce's automation tools to reduce manual processes, allowing for quicker updates, more accurate data, and a smoother experience for customers booking hotels, flights, and food options.

This project demonstrates how Salesforce automation can effectively improve the efficiency of business operations and enhance the customer experience in the travel and hospitality industry, particularly for platforms like TripAdvisor. By integrating key processes into a single, automated system, the project helps streamline workflows, reduce manual effort, and provide a superior service to customers.