TripAdvisor E-Management

College: 7125 - PPG Institute of Technology

Team ID: NM2024TMID17249

Team Lead: Dhanalakshmi K (1812F1055D0A6BAA9FF4BBAACDF75820)
Team Members:

Dhanalakshmi K	1812F1055D0A6BAA9FF4BBAACDF75820
Saranprasath T	9698D779A43529FAB574880EDE77B6D4
Arunpandi A	5B1888D5510255486794C8402C35EFA1
Boopathi N	F215F5639F2EEFE89AD4237E60F1E582

1. Project Overview

The TripAdvisor E-Management app integrates with Salesforce to create a comprehensive travel companion. The app enables users to plan, book, and optimize their travel experiences by managing information related to hotels, flights, and food options. The platform facilitates informed travel decisions, leveraging millions of reviews and insights to support the best choices in accommodations, dining, attractions, and deals.

2. Objectives

Business Goals:

- Enhance the efficiency and user experience of the travel management process.
- Automate data management tasks, discount application, and email notifications for timely customer engagement.
- Provide seamless tracking and updates to ensure accurate information on hotels, food options, and flight details.

Specific Outcomes:

- Automated synchronization between food options and hotels to ensure accurate hotel data.
- A dynamic discount system for customers based on their spending, enhancing customer satisfaction.
- Scheduled email alerts for flight bookings to ensure timely notifications, improving customer travel experience.

3. Salesforce Key Features and Concepts Utilized

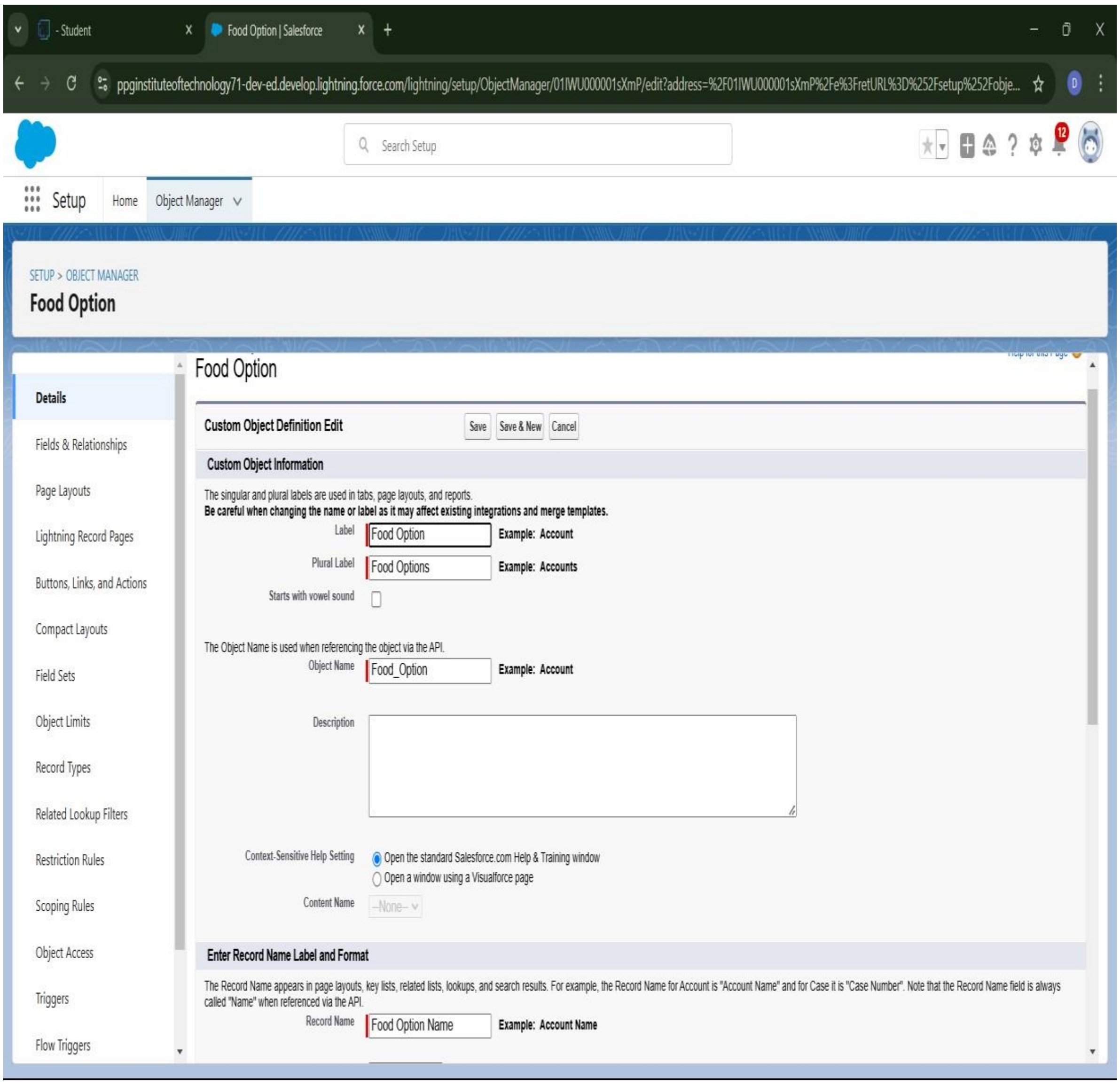
This project leverages key Salesforce functionalities and concepts to create a seamless and effective food distribution system:

- **Custom Objects and Fields** ○ **Hotel**: Stores hotel details, including associated food options.
    - **Food Option**: Tracks available food choices per hotel (Auto Number: FO-{0000}).
    - **Flight**: Records customer flight details (Auto Number: FL-{0000}).
    - **Customer**: Manages customer details for discount eligibility.
- **Flow for Discounts**
  Created a Flow to apply automatic discounts based on customer spending:
    - **Spending > 3000**: Apply higher discount rate.
    - **Spending between 1500 and 3000**: Apply lower discount rate.
- **Apex Triggers for Data Synchronization** ○ **Hotel-Food Option Synchronization:** Developed an Apex trigger that updates hotel information whenever a new food option is added or modified, ensuring accurate food count per hotel.
- **Apex Schedulable Class for Flight Reminders**
  Created an **Apex Schedulable class** to automate email reminders for customers with booked flights.
    - **Reminder Schedule:** Sends an email notification 24 hours before departure.
    - **Confirmation:** System provides confirmation that the email was sent successfully.

These Salesforce features collectively ensure that the project operates with high efficiency, transparency, and data-driven decision-making to maximize food distribution effectiveness.

4. Detailed Steps to Solution Design ✞ Created objects

- In the salesforce developer platform, we created custom objects that were required for the project.
- There were 4 main objects
- They were Hotel, Flight, Food Option, Customer.
- This was done by using the object manager



Food Option

Custom Object Definition Edit Save | Save & New | Cancel

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.
Be careful when changing the name or label as it may affect existing integrations and merge templates.

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--	

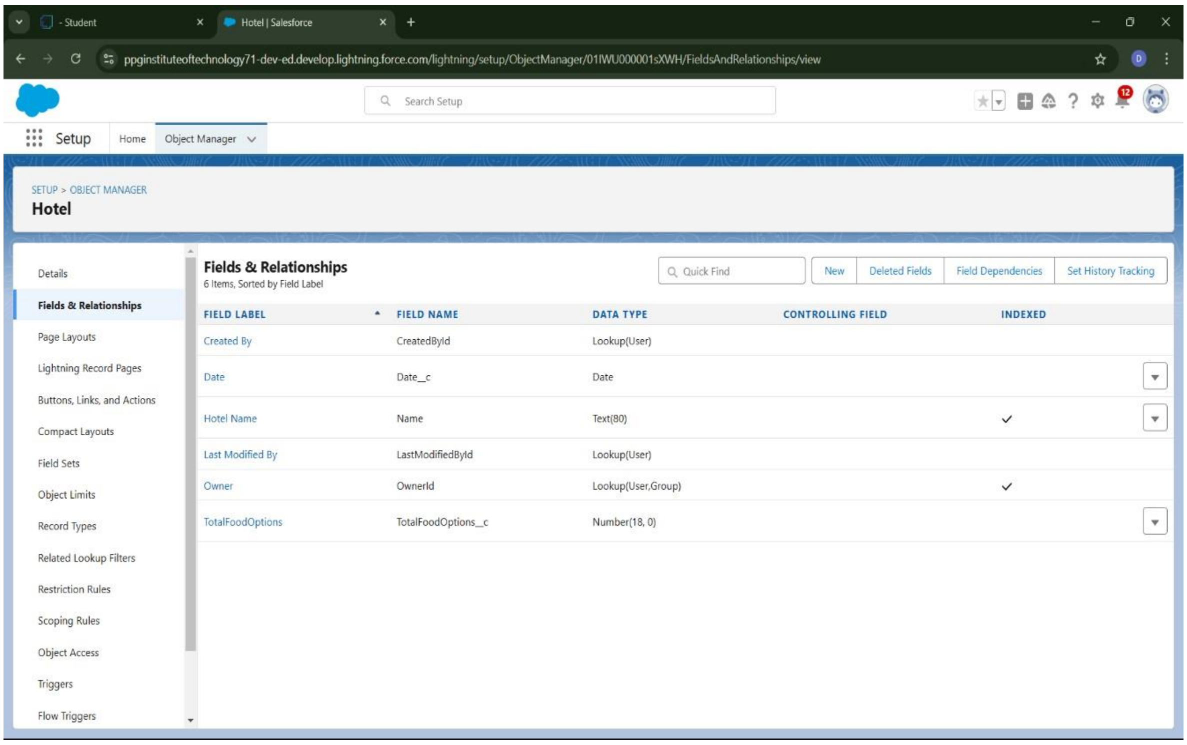
Enter Record Name Label and Format

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**

Created fields for hotel object

- TotalFoodOptions with datatype as number
- Date with datatype as date



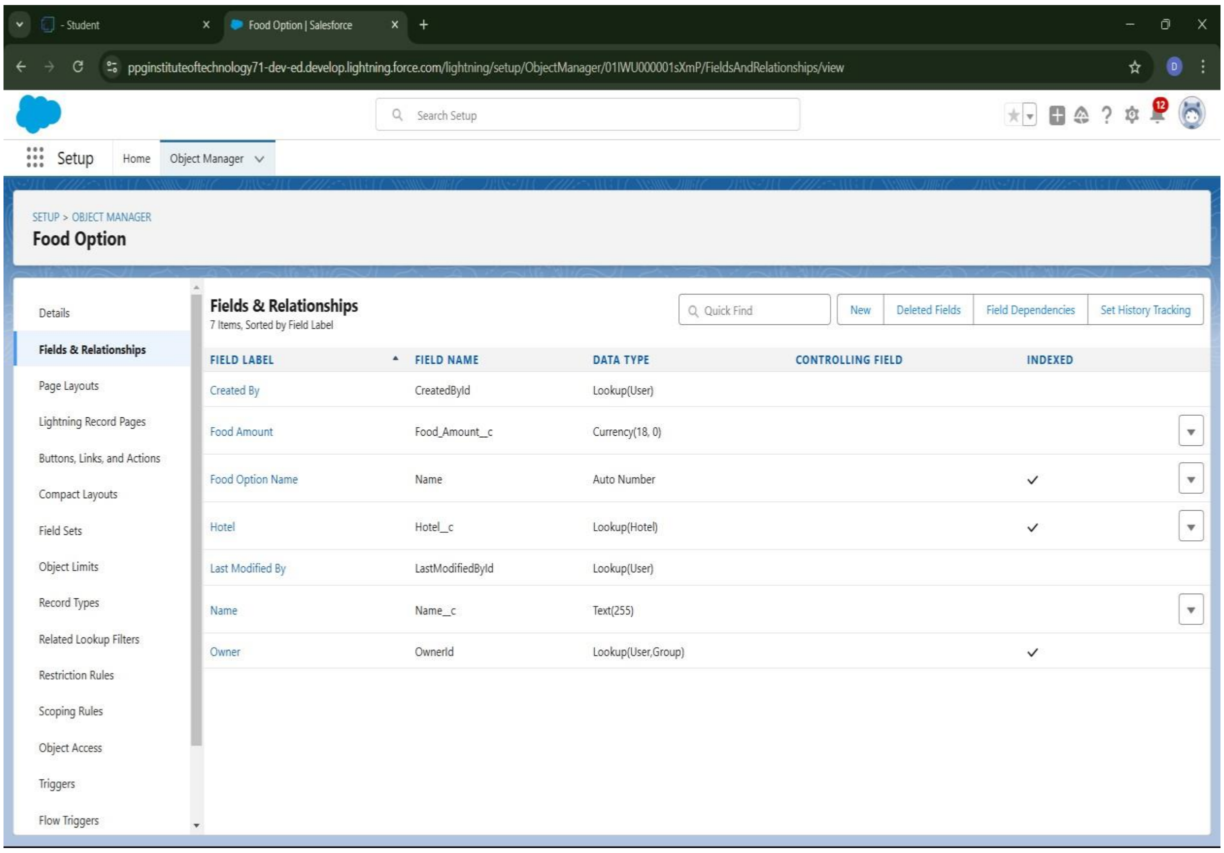
Fields & Relationships
6 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Date	Date__c	Date		
Hotel Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
TotalFoodOptions	TotalFoodOptions__c	Number(18, 0)		

Create Fields for Food Option

- Food Amount
- Hotel
- Name

✞



ppginstituteoftechnology71-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IWU000001sXmP/FieldsAndRelationships/view

SETUP > OBJECT MANAGER

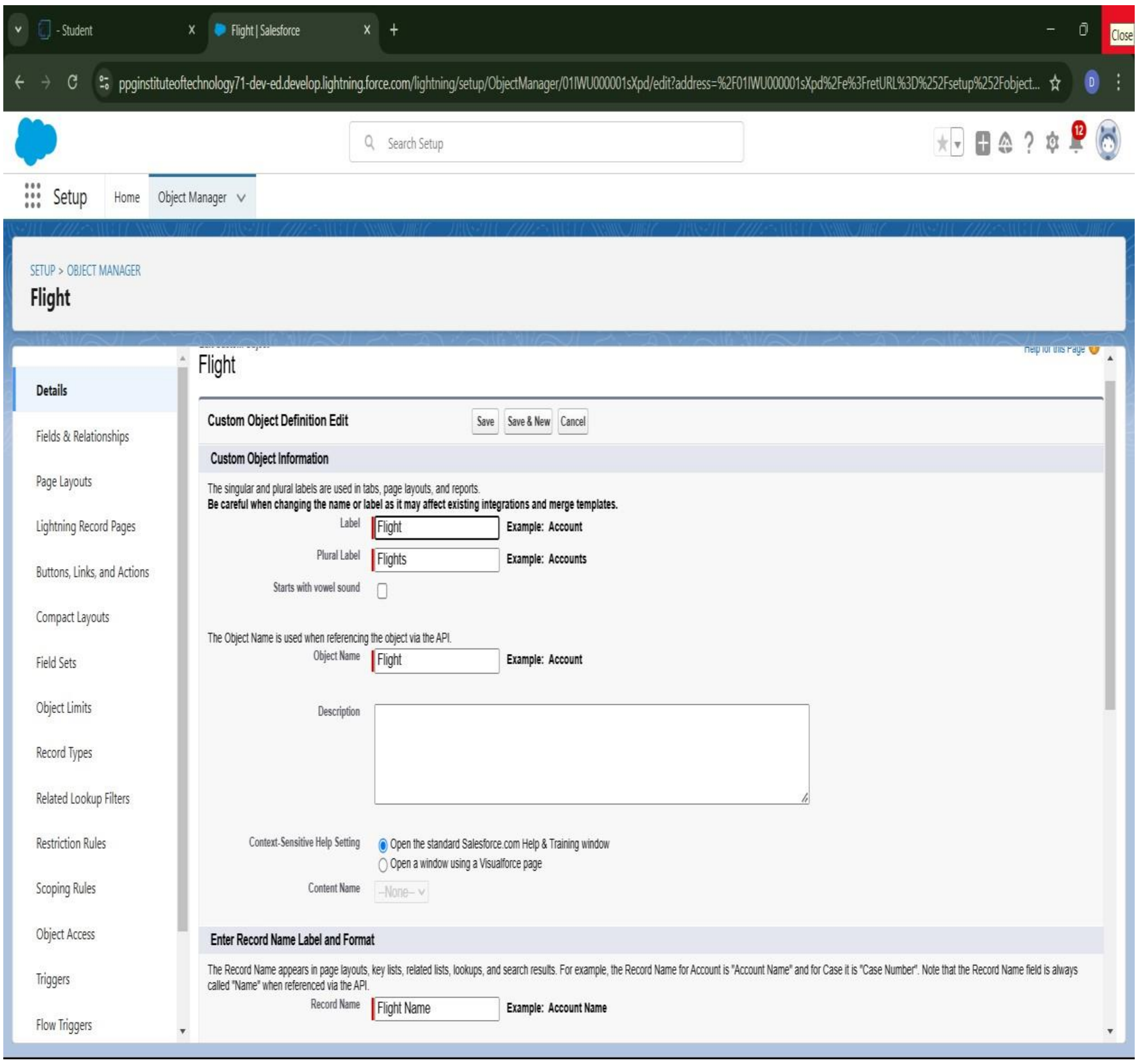
Food Option

Fields & Relationships
7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Food Amount	Food_Amount__c	Currency(18, 0)		
Food Option Name	Name	Auto Number		✓
Hotel	Hotel__c	Lookup(Hotel)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Text(255)		
Owner	OwnerId	Lookup(User,Group)		✓

Created Fields for Flight object

- Name
- DepartureDateTime


SETUP > OBJECT MANAGER

Flight

Flight

Custom Object Definition Edit

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.
Be careful when changing the name or label as it may affect existing integrations and merge templates.

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--	

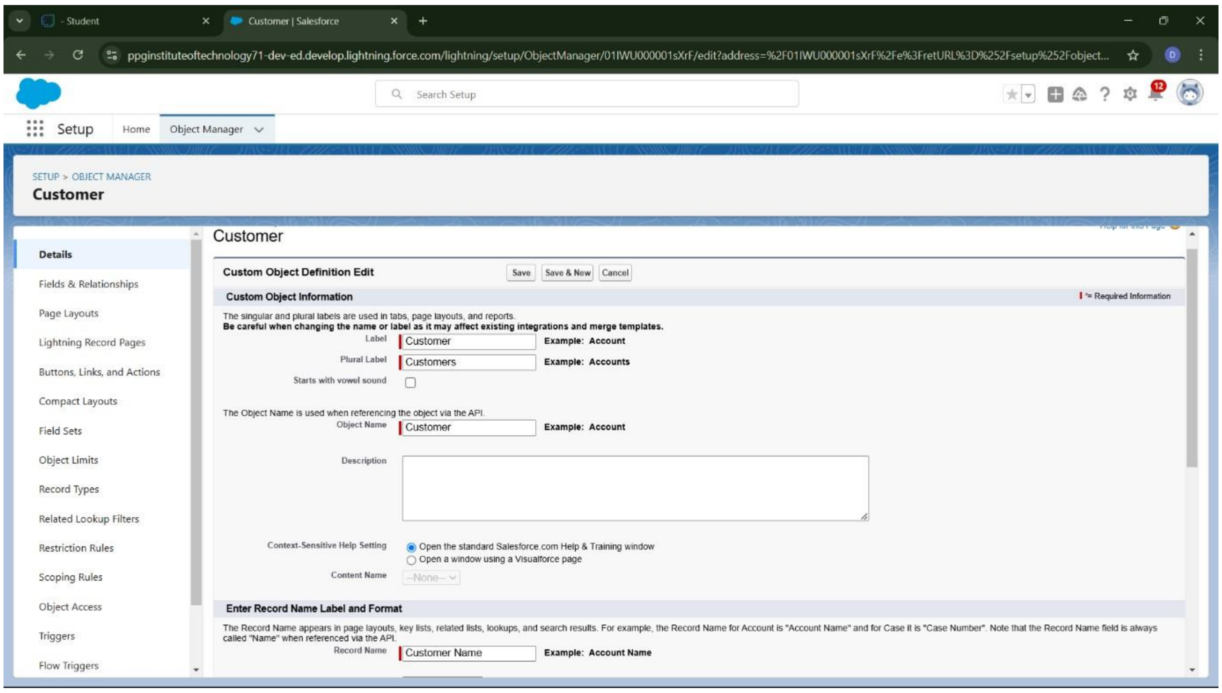
Enter Record Name Label and Format

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**

Created fields for customer object

- Customer name
- Discount amount
- Discount percentage



Customer

Custom Object Definition Edit Save | Save & New | Cancel

Custom Object Information — = Required Information

The singular and plural labels are used in tabs, page layouts, and reports.
Be careful when changing the name or label as it may affect existing integrations and merge templates.

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

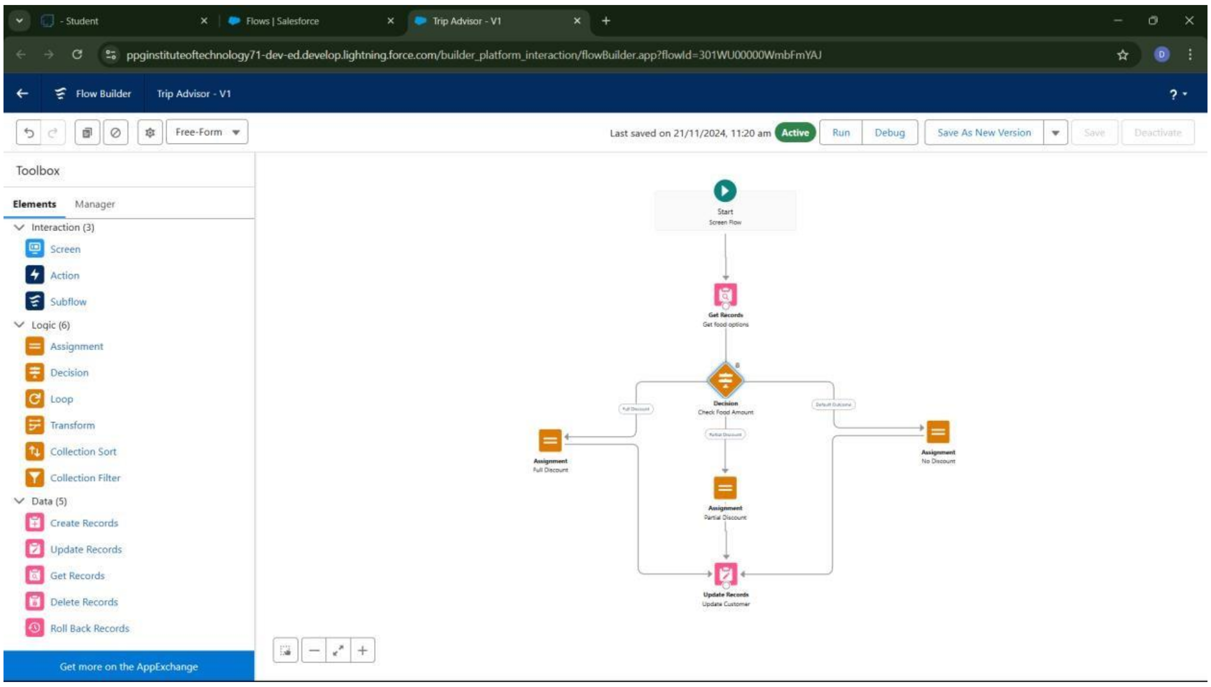
The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name	Customer Name	**Example: Account Name**

Created Flow

- Developed a **Discount Flow** to automatically apply discounts based on customer purchase amounts. This flow applies discounts in a step-by-step format to ensure ease of entry and accuracy.
- **Flow Conditions**: The flow is triggered when the customer purchase **Amount** meets specific thresholds:
  - For purchases greater than **3000**, a high discount rate is applied.
  - For purchases between **1500 and 3000**, a medium discount rate is applied.

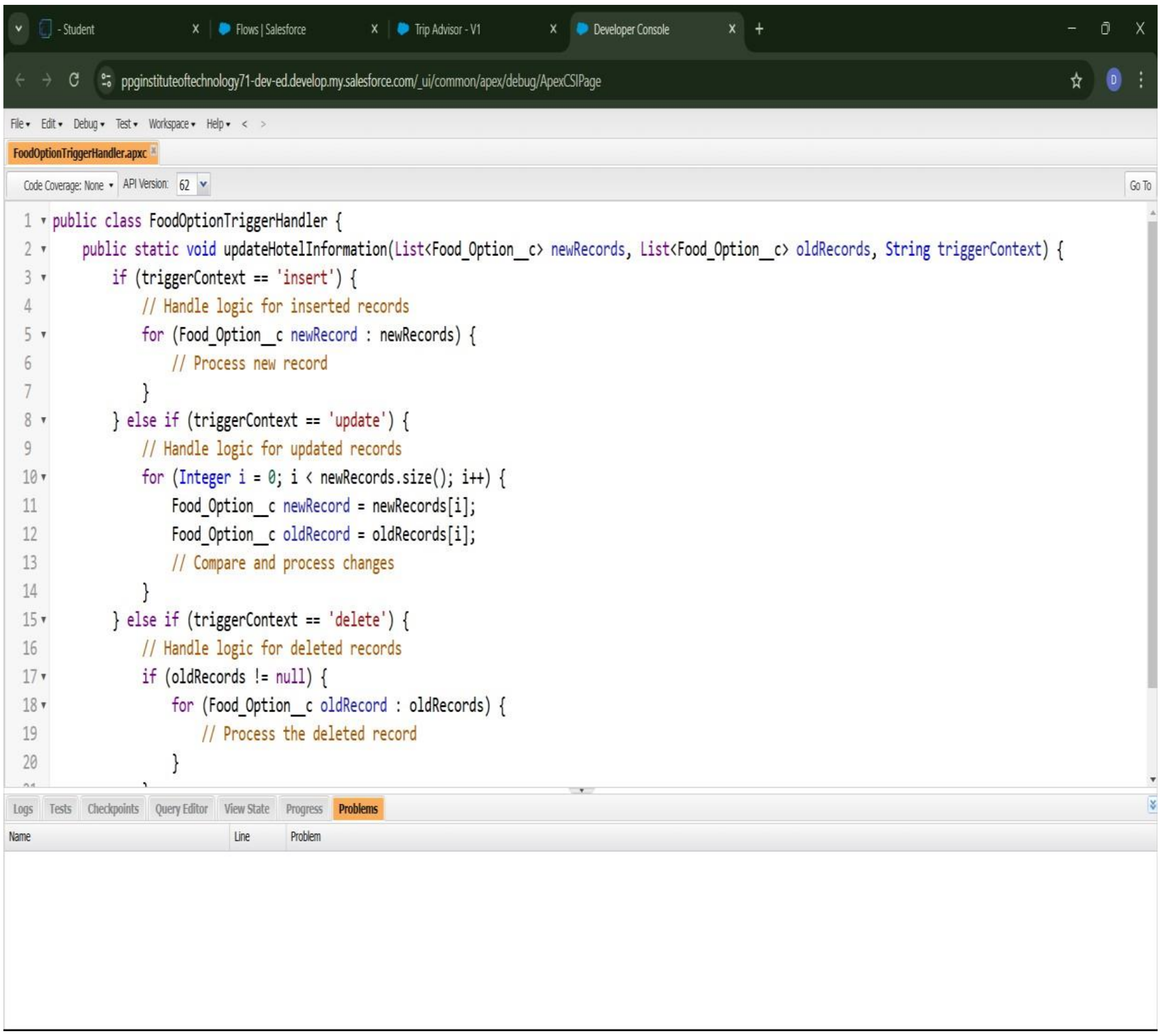
✞



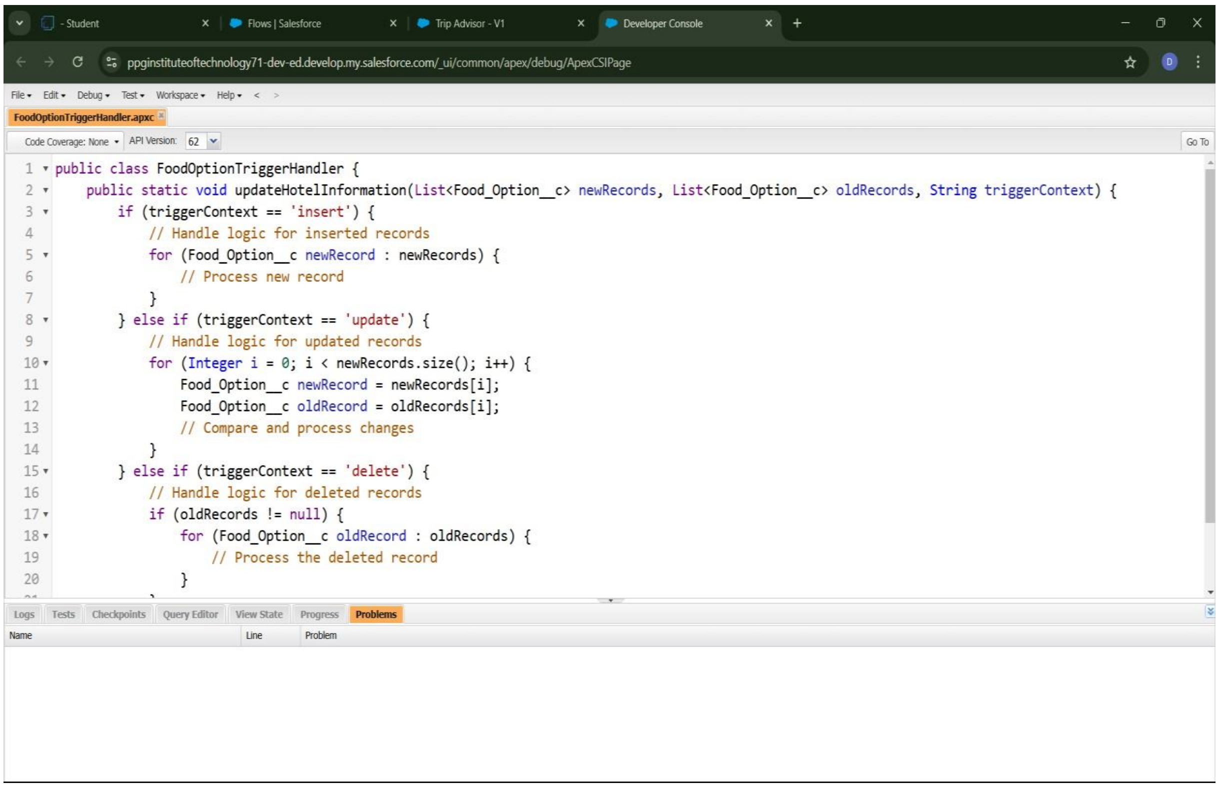
Created Apex Trigger for Food Option

- Developed an **Apex Trigger** to ensure synchronization between **Hotel** and **Food Option** records, maintaining clear and manageable records of food options available at each hotel.
- Trigger Conditions: The trigger is activated whenever a **Food Option** record is added or updated to reflect changes in the associated **Hotel** record.

✞



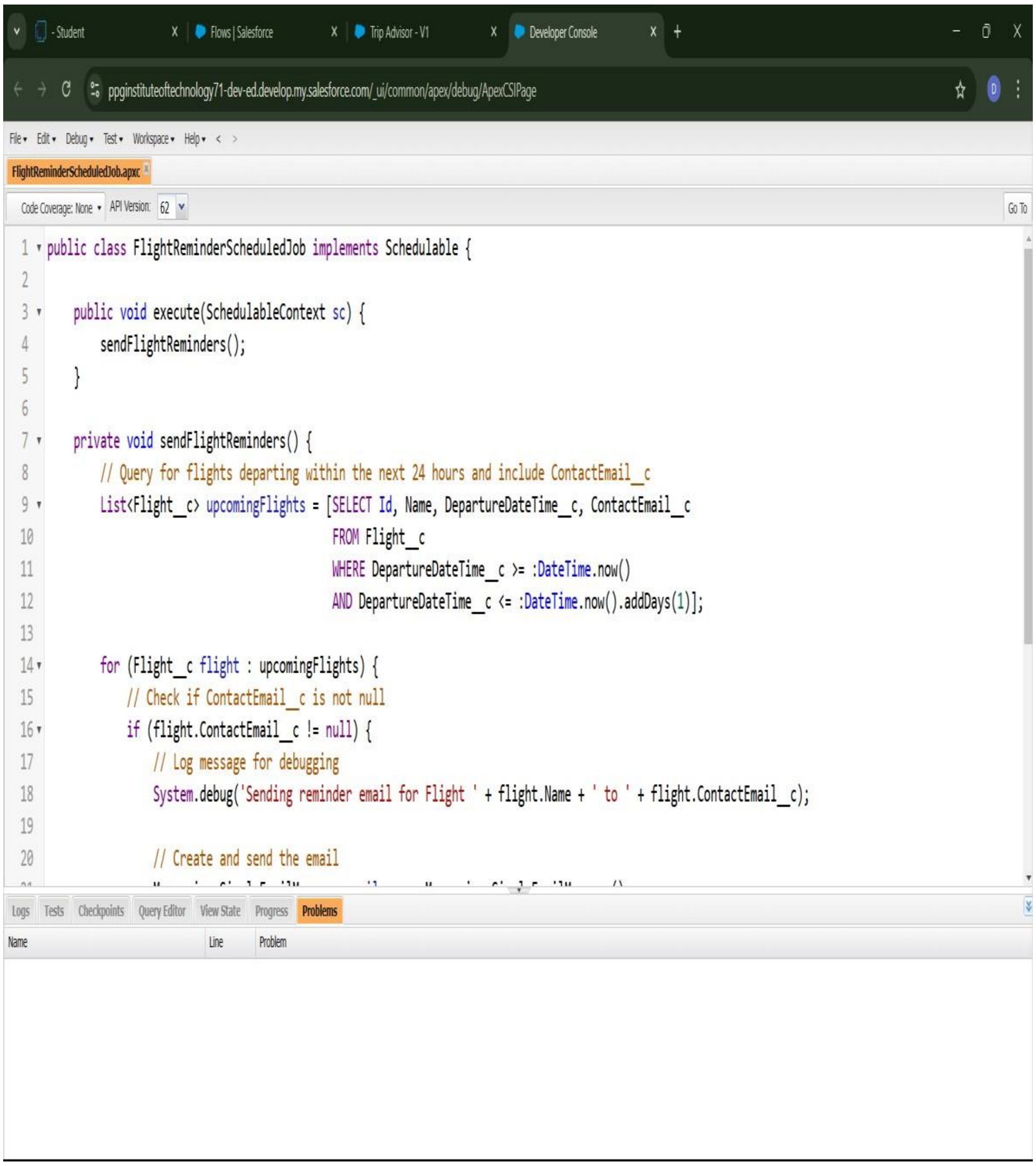
```
public class FoodOptionTriggerHandler {
    public static void updateHotelInformation(List<Food_Option__c> newRecords, List<Food_Option__c> oldRecords, String triggerContext) {
        if (triggerContext == 'insert') {
            // Handle logic for inserted records
            for (Food_Option__c newRecord : newRecords) {
                // Process new record
            }
        } else if (triggerContext == 'update') {
            // Handle logic for updated records
            for (Integer i = 0; i < newRecords.size(); i++) {
                Food_Option__c newRecord = newRecords[i];
                Food_Option__c oldRecord = oldRecords[i];
                // Compare and process changes
            }
        } else if (triggerContext == 'delete') {
            // Handle logic for deleted records
            if (oldRecords != null) {
                for (Food_Option__c oldRecord : oldRecords) {
                    // Process the deleted record
                }
```


- Student
Flows | Salesforce
Trip Advisor - V1
Developer Console
ppginstituteoftechnology71-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
FoodOptionTriggerHandler.apxc
Code Coverage: None
API Version: 62
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem

```
public class FoodOptionTriggerHandler {
    public static void updateHotelInformation(List<Food_Option__c> newRecords, List<Food_Option__c> oldRecords, String triggerContext) {
        if (triggerContext == 'insert') {
            // Handle logic for inserted records
            for (Food_Option__c newRecord : newRecords) {
                // Process new record
            }
        } else if (triggerContext == 'update') {
            // Handle logic for updated records
            for (Integer i = 0; i < newRecords.size(); i++) {
                Food_Option__c newRecord = newRecords[i];
                Food_Option__c oldRecord = oldRecords[i];
                // Compare and process changes
            }
        } else if (triggerContext == 'delete') {
            // Handle logic for deleted records
            if (oldRecords != null) {
                for (Food_Option__c oldRecord : oldRecords) {
                    // Process the deleted record
                }
```

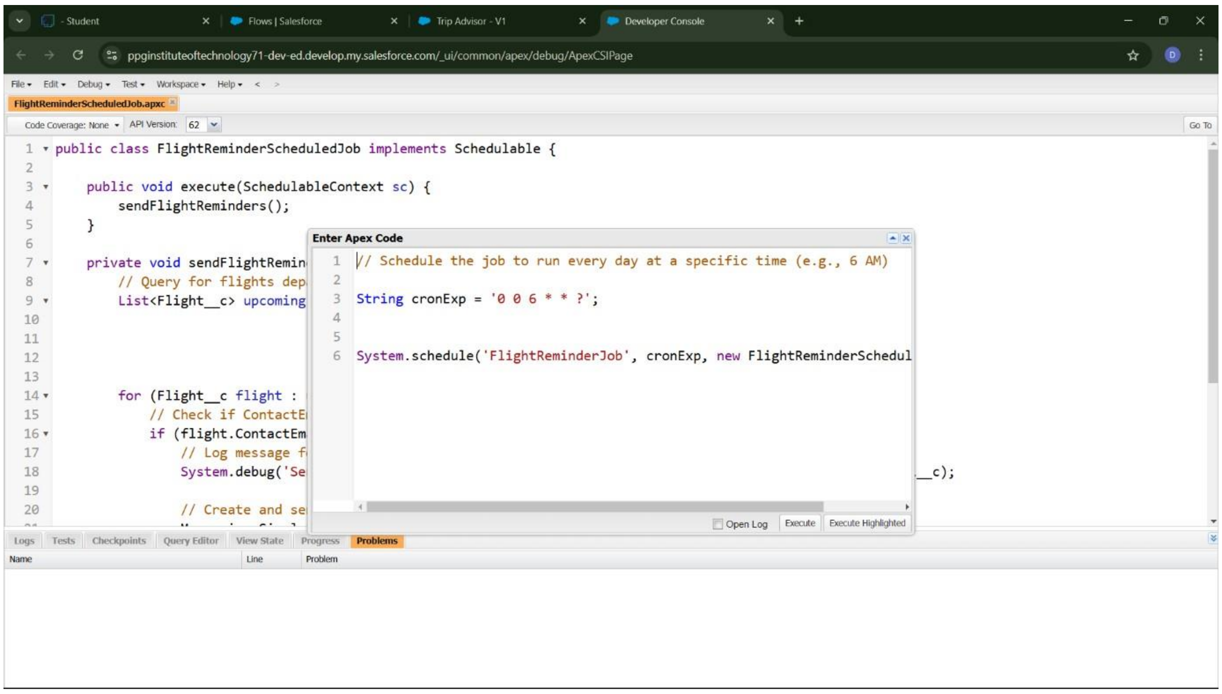
✞ Created Apex Schedule

- Developed an **Apex Schedule** to send reminder emails to customers who have booked flights, ensuring they receive a notification 24 hours before their scheduled flight.
- The **Apex Schedule** is set to run daily and check for flight bookings scheduled within the next 24 hours.
- If the booking is within 24 hours, an **email alert** is triggered to remind the customer of their upcoming flight.



- Student
Flows | Salesforce
Trip Advisor - V1
Developer Console
ppginstituteoftechnology71-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
FlightReminderScheduledJob.apxc
Code Coverage: None API Version: 62
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem

```
public class FlightReminderScheduledJob implements Schedulable {

    public void execute(SchedulableContext sc) {
        sendFlightReminders();
    }

    private void sendFlightReminders() {
        // Query for flights departing within the next 24 hours and include ContactEmail__c
        List<Flight__c> upcomingFlights = [SELECT Id, Name, DepartureDateTime__c, ContactEmail__c
                                           FROM Flight__c
                                           WHERE DepartureDateTime__c >= :DateTime.now()
                                           AND DepartureDateTime__c <= :DateTime.now().addDays(1)];

        for (Flight__c flight : upcomingFlights) {
            // Check if ContactEmail__c is not null
            if (flight.ContactEmail__c != null) {
                // Log message for debugging
                System.debug('Sending reminder email for Flight ' + flight.Name + ' to ' + flight.ContactEmail__c);

                // Create and send the email
```


Student
Flows | Salesforce
Trip Advisor - V1
Developer Console
ppginstituteoftechnology71-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
FlightReminderScheduledJob.apxc
Code Coverage: None
API Version: 62
Go To
Enter Apex Code
Open Log
Execute
Execute Highlighted
Logs Tests Checkpoints Query Editor View State Progress Problems
Name
Line
Problem

```
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 :
            // Check if ContactE
            if (flight.ContactEm
                // Log message f
                System.debug('Se                                                        __c);

                // Create and se
```

```
// 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 FlightReminderSchedul
```

5. Testing and Validation Apex

Trigger:

```
trigger FoodOptionTrigger on Food_Option_c (after insert, after update, after
delete) {

   if (trigger.isInsert && trigger.isAfter) {

      FoodOptionTriggerHandler.updateHotelInformation(trigger.new);

   }

}
```

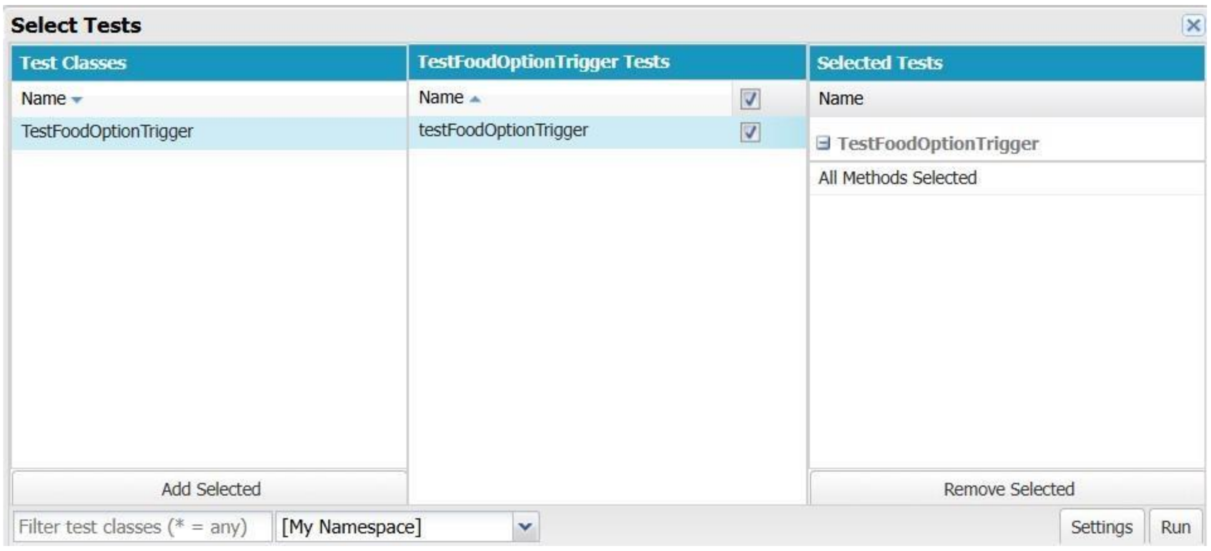
Test Class:

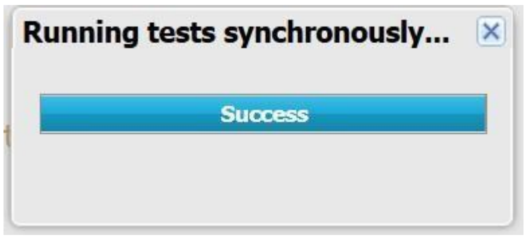
```
@isTest
private class TestFoodOptionTrigger {

   @isTest static void testFoodOptionTrigger() {
      // Create a Hotel record for reference
      Hotel_c hotel = new Hotel_c(Name = 'Test Hotel');
insert hotel;
```

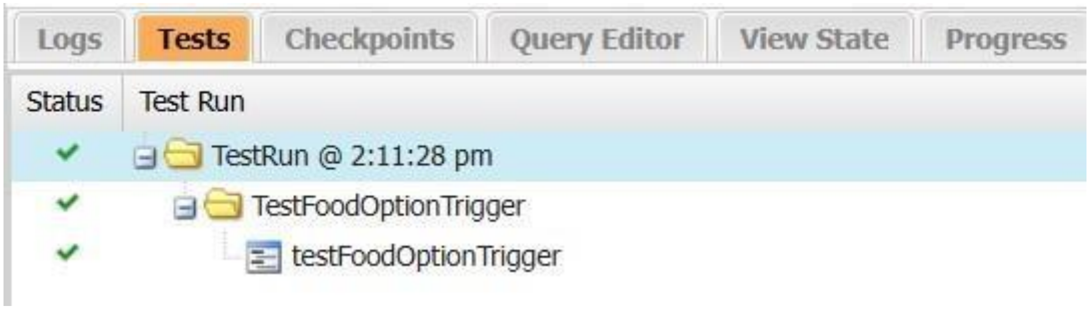
```
        // Create a Food Option record linked to the Hotel
        Food_Option_c foodOption1 = new Food_Option_c(Hotel_c = hotel.Id);
insert foodOption1;

        // Verify if Hotel's TotalFoodOptions__c is updated correctly
        Hotel__c updatedHotel = [SELECT TotalFoodOptions__c FROM Hotel__c WHERE
Id = :hotel.Id];
        System.assertEquals(1, updatedHotel.TotalFoodOptions_c,
'TotalFoodOptions_c should be updated to 1');

        // Create another Food Option and check the count again
        Food_Option_c foodOption2 = new Food_Option_c(Hotel_c = hotel.Id);
insert foodOption2;

        updatedHotel = [SELECT TotalFoodOptions__c FROM Hotel__c WHERE Id =
:hotel.Id];
        System.assertEquals(2, updatedHotel.TotalFoodOptions_c,
'TotalFoodOptions_c should be updated to 2');
    }
}
```

STEPS:

Step 1: Creates a Hotel_c record with TotalFoodOptions_c initialized to 0.
Step 2: Inserts a Food_Option_c record associated with the hotel.
Step 3: Verifies that TotalFoodOptions_c on the hotel is updated to 1 after adding the first food option.
Step 4: Adds another Food_Option__c to check if TotalFoodOptions__c increments to 2.
Step 5: Updates a food option record to confirm that updates do not affect the count.
Step 6: Deletes one Food_Option_c and verifies that TotalFoodOptions_c decrements accordingly.







6. Conclusion

Summary of Achievements:

The **TripAdvisor E-Management** project successfully established an all-in-one travel management platform on Salesforce. Key achievements include:

- **Comprehensive Data Management**: Created custom objects and fields to manage essential travel data, supporting organized and accessible records.
- **Automated Processes**: Used flows and Apex triggers to enhance operational efficiency, reducing manual input and improving data accuracy.
- **Enhanced Collaboration**: Configured profiles and public groups to allow secure collaboration, protecting data privacy.

- **Real-Time Monitoring**: Developed custom reports and a centralized dashboard for real-time insights into booking trends and customer preferences.
- **Improved Decision-Making**: Created a streamlined, user-friendly system that supports seamless travel planning and booking, enabling quick, informed decision-making for users.

This project demonstrates the effective use of Salesforce to support a dynamic, usercentered travel management solution, making TripAdvisor an invaluable resource for travelers.