

Report: TripAdvisor E-Management

Project Overview

Project Title: TripAdvisor E-Management

This project is focused on developing the *TripAdvisor E-Management System* using Salesforce to streamline travel-related services, such as hotel, food option, and flight management, along with customer discount automation and notification services. The project aims to address inefficiencies in data handling and communication within TripAdvisor's ecosystem by implementing a comprehensive and automated solution. The primary challenge was to ensure seamless integration of various services to enhance operational efficiency, reduce manual errors, and improve the user experience.

By leveraging Salesforce's robust platform features, such as automation tools, Apex triggers, and schedulable classes, this project provides a scalable, reliable, and efficient solution to meet the business needs of TripAdvisor.

Project Description:

The TripAdvisor E-Management system, integrated with Salesforce, aims to provide an all-in-one travel companion app that empowers users to plan, book, and make the most of their trips. This system brings TripAdvisor's massive repository of user-generated reviews and insights into the Salesforce ecosystem, allowing users to access information on hotels, flights, food options, and customer deals seamlessly. This report outlines the system requirements, acceptance criteria, and solutions developed for this integration, with a focus on automation and streamlined customer experience.

Short Description:

TripAdvisor E-Management Solution streamlines travel itinerary management, centralizing booking data, trip tracking, and user feedback, making it easier for organizations to oversee and enhance travel experiences.

Objectives

Business Goals:

1. Automate the management of hotels, flights, and food options to reduce manual administrative tasks.
2. Provide a personalized customer experience through automated discount mechanisms.
3. Ensure timely communication with customers via email notifications for flight reminders.

Specific Outcomes:

1. Automated tracking and updating of hotel information based on food options.
2. Automated discount calculation and application based on customer purchase thresholds.
3. Flight reminder emails sent automatically 24 hours before flight departure, improving customer satisfaction.
4. Accurate, real-time reporting and data analytics for better business insights.

Salesforce Key Features and Concepts Utilized

The project leverages the following Salesforce features and functionalities:

1. Custom Objects:

- **Hotel Object:** Stores hotel-related data and calculates the total number of food options associated with each hotel.
- **Food Option Object:** Tracks food options linked to hotels.
- **Flight Object:** Manages flight bookings and departure schedules.
- **Customer Object:** Stores customer information, such as name and booking details.

2. Apex Triggers:

- Ensures automatic updating of hotel data whenever food options are added or modified.
- Example: Updating the total count of food options for each hotel.

3. Flows:

- Automates customer discount application based on purchase amounts.
- Example: Full discounts for purchases exceeding 3000 and partial discounts for purchases between 1500 and 3000.

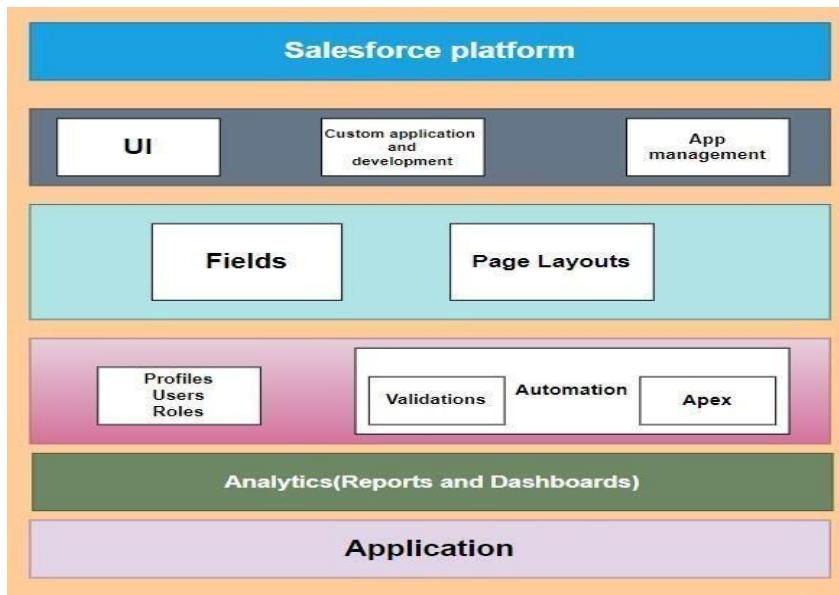
4. Schedulable Apex Classes:

- Handles scheduled email notifications for customers regarding upcoming flights.
- Example: Sends email reminders 24 hours before the scheduled departure.

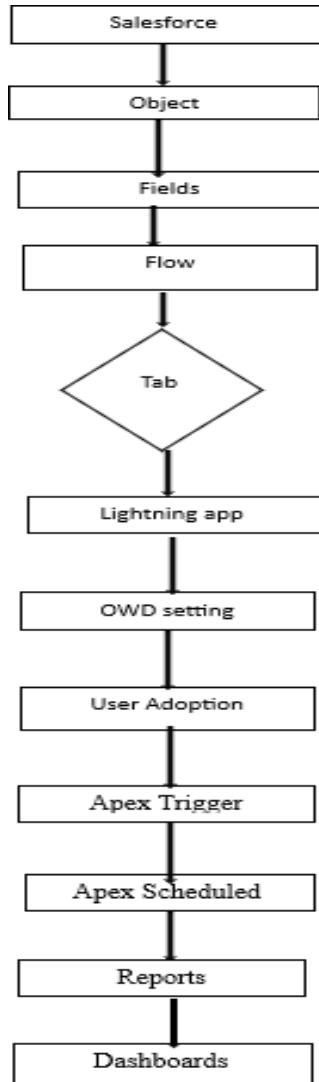
5. Reports and Dashboards:

- Tracks customer discounts, flight bookings, and hotel occupancy rates for operational insights.

Technical Architecture:



Project Flow:



Detailed Steps to Solution Design

Milestone 1- Salesforce

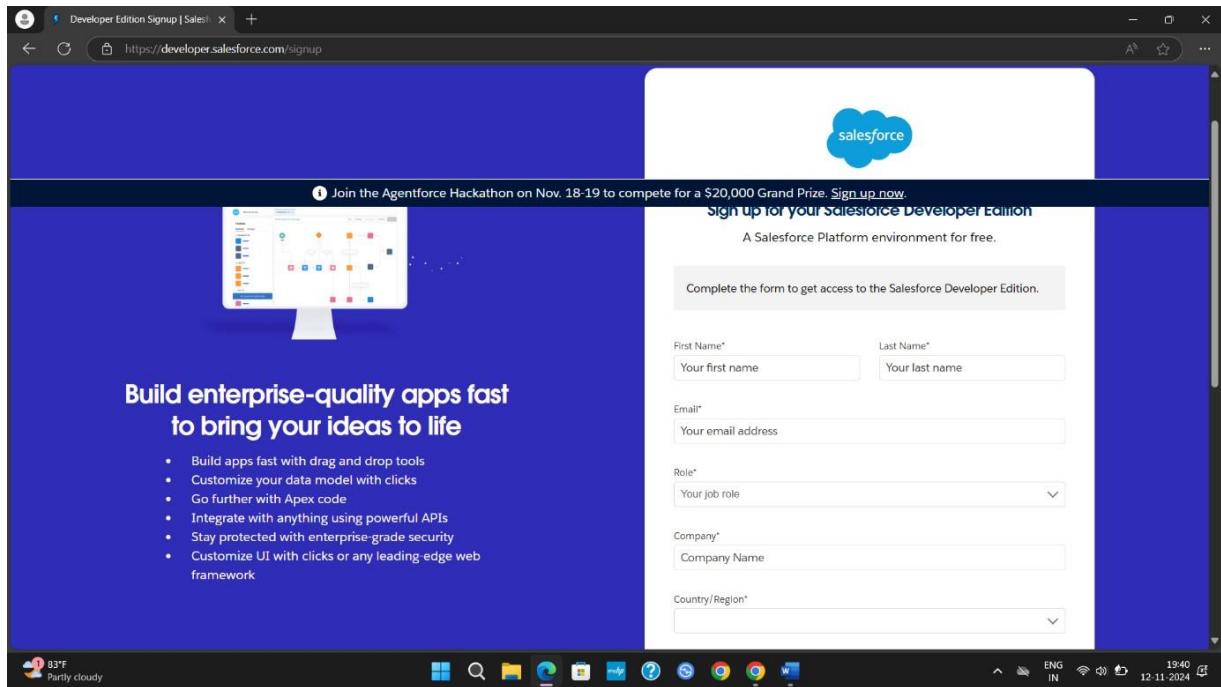
Developer Account Creation

To start working with Salesforce CRM, a developer account is essential. Follow these steps to create an account:

1.Sign-Up Process

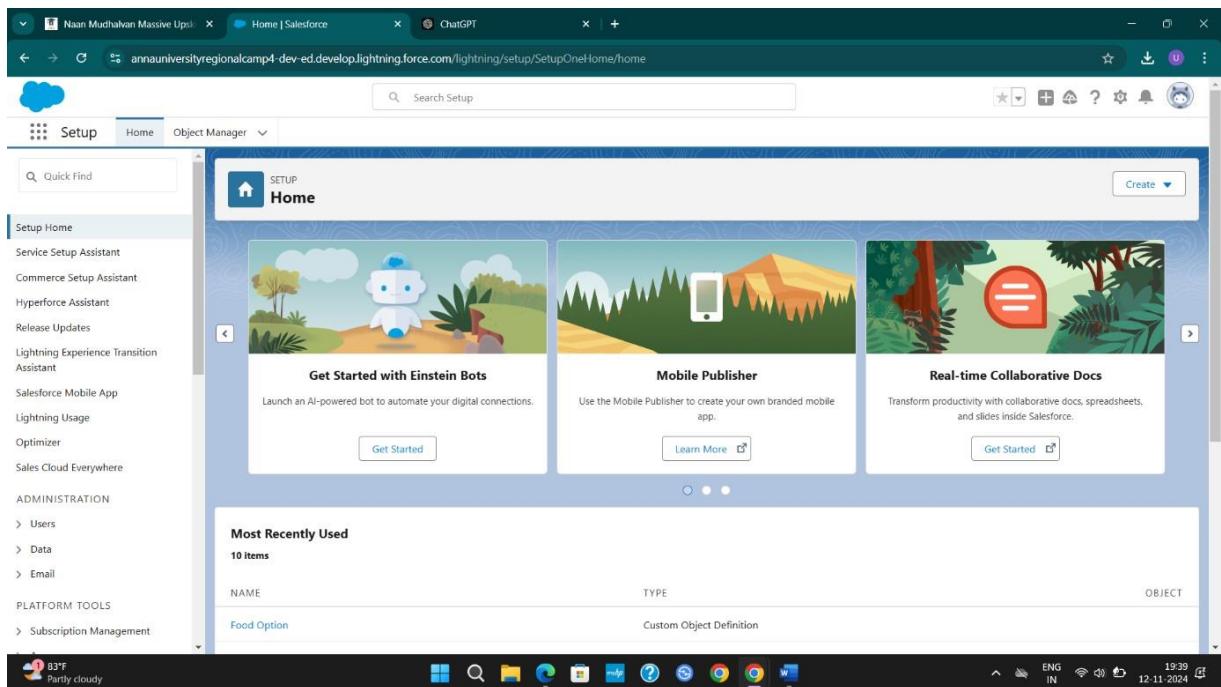
- Go to [Salesforce Developer Sign-Up](#).
- Enter your **First and Last Name**, **Email**, and set **Role** as “Developer.”
- Input your **Company** (College Name), **Country** (India), **Postal Code**, and **Username** (formatted as username@organization.com).

Click **Sign Me Up** after filling out the form



Account Activation

- Open the inbox of the email used for registration, locate the Salesforce verification email, and click **Verify Account**.
- Set a password, choose a security question, and log into your Salesforce account to access the setup page.



Milestone 2 - Objects in Salesforce

Salesforce objects function as database tables for storing and organizing data relevant to the organization.

- **Standard Objects:** Provided by Salesforce by default (e.g., Accounts, Contacts).
- **Custom Objects:** User-defined objects to store unique organizational data.

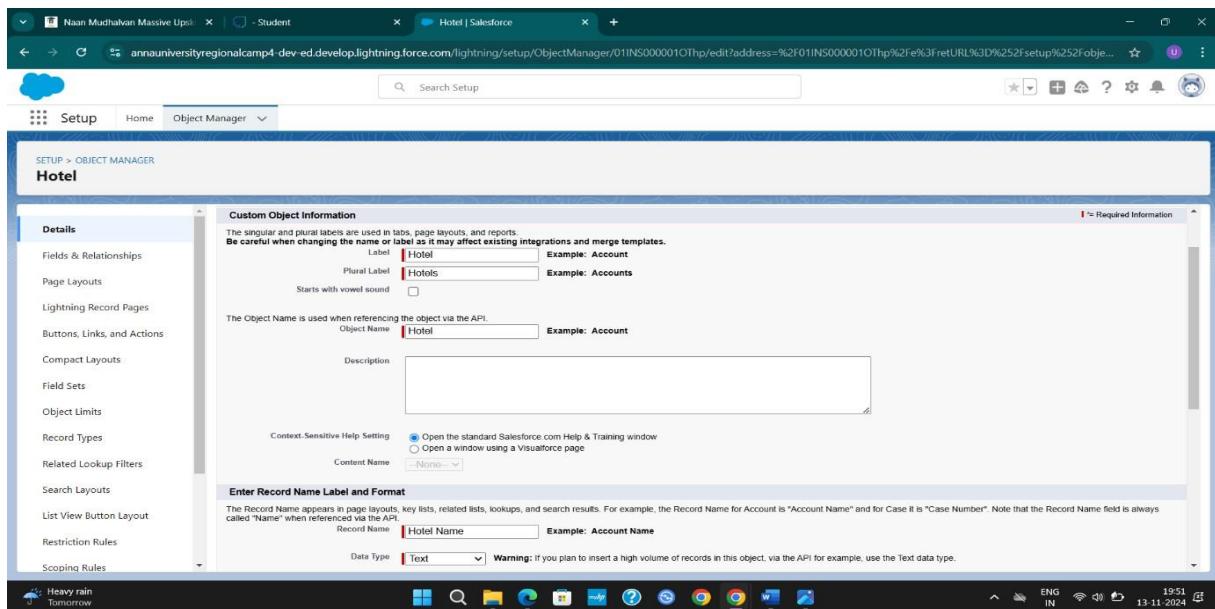
System Development: Custom Objects

Hotel Object:

Hotel Object is created to ensure that when a new Food Option is added or updated with the necessary information

1. Enter label : Hotel
2. Plural Name : Hotels
3. Data Type : (text)
4. Field Name : Hotel Name
5. Click Allow Reports
6. Allow Search ? Save

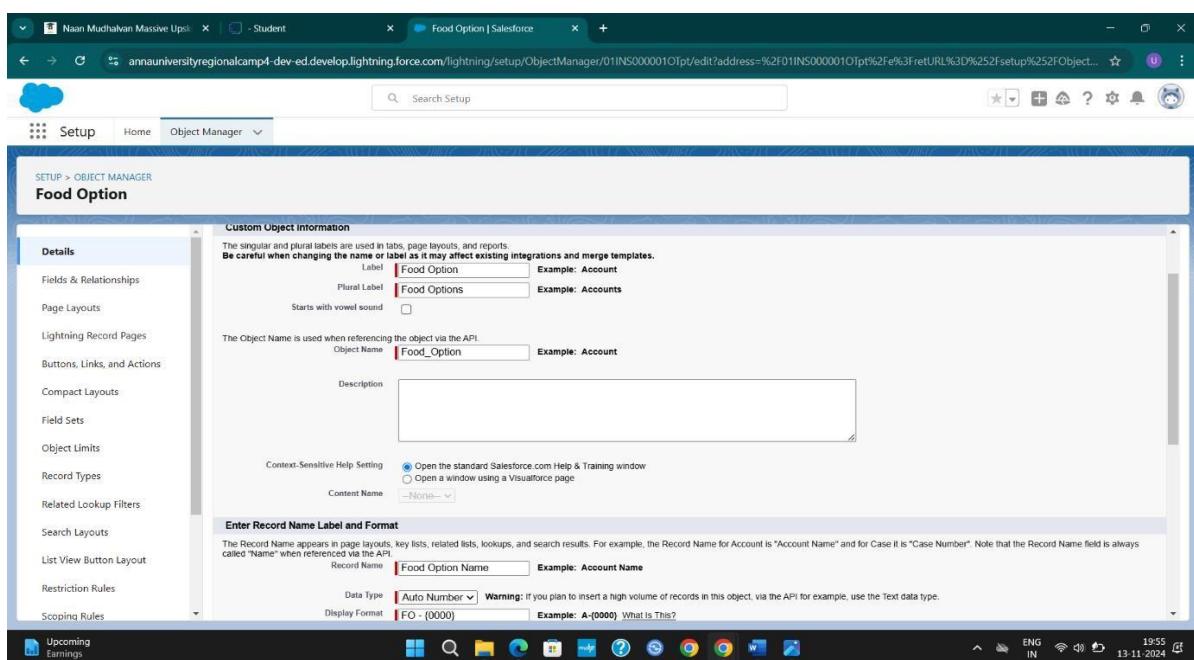
Purpose: Store data about hotels and update hotel information when new food options are added.



Food Option Object:

Food Option > Data Type > Auto Number > Format > FO - {0000}

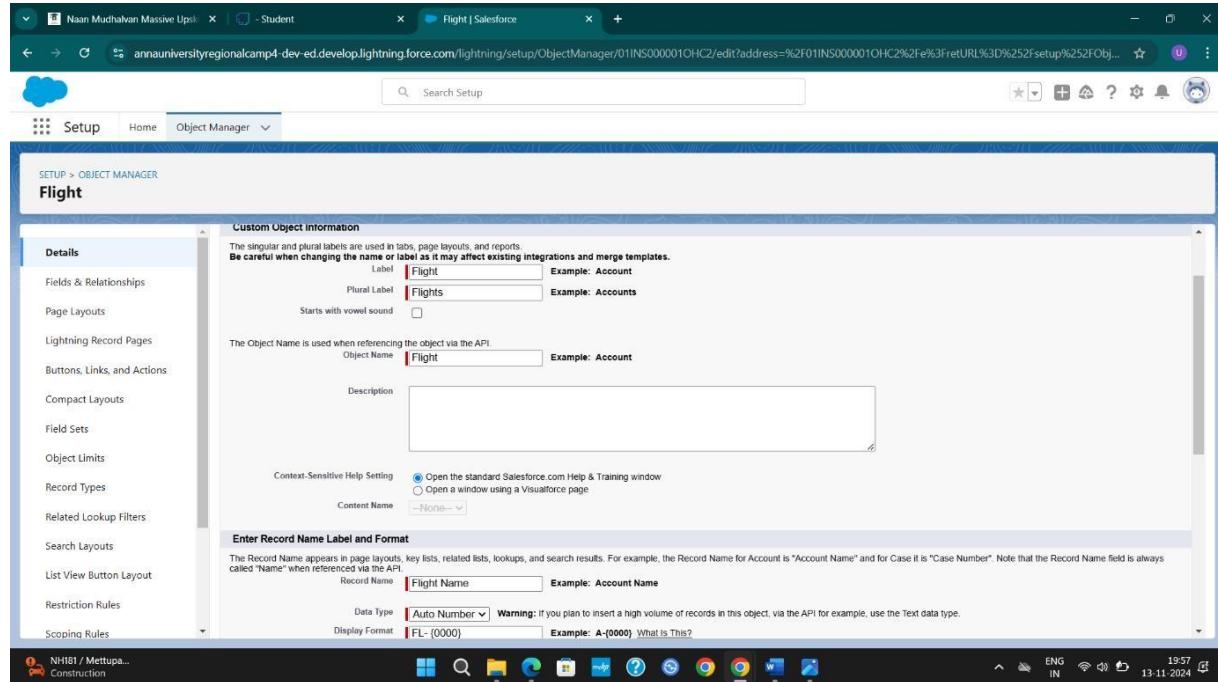
Purpose: Track food options associated with hotels.



Flight Object:

Flight > Data Type > Auto Number > Format > FL- {0000}

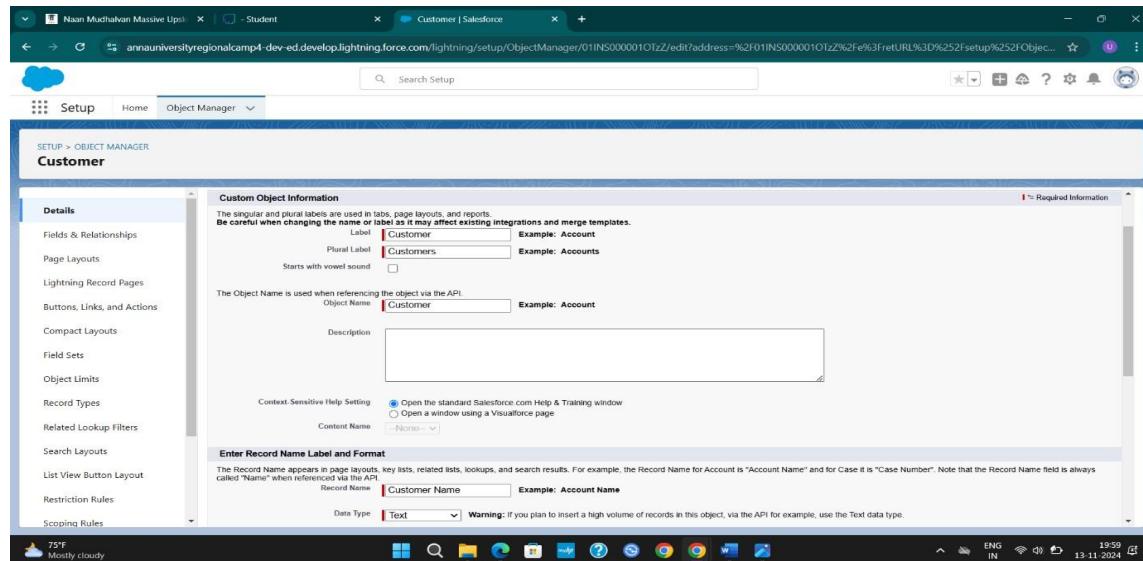
Purpose: Track flight bookings and manage customer notifications.



Customer Object:

Customer > Text > Field Name > Customer Name

Purpose: Manage customer information.

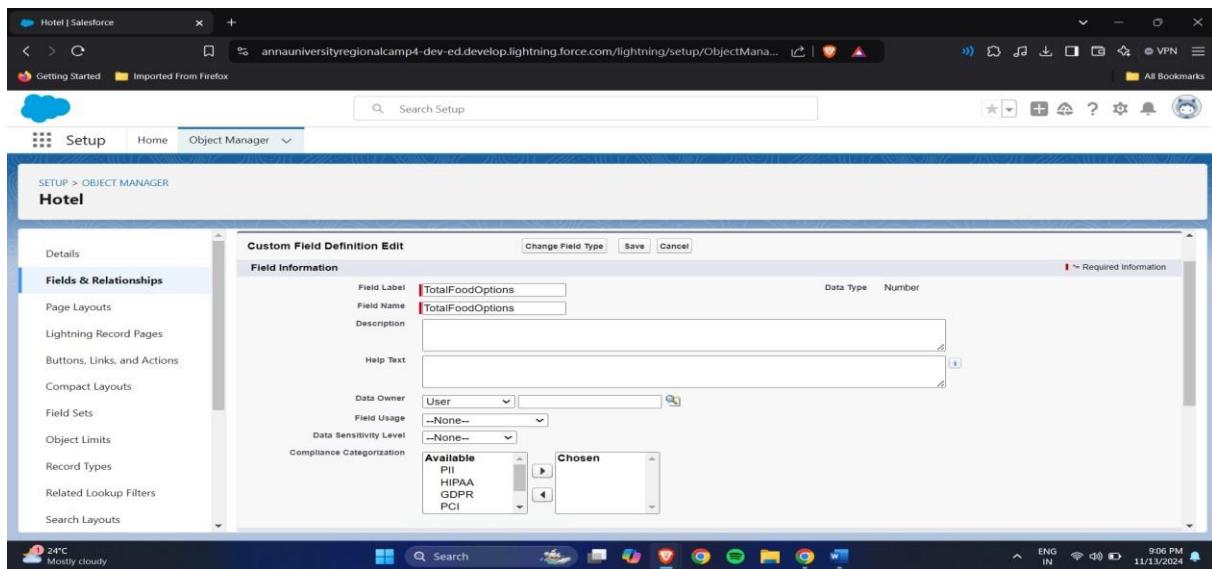


Milestone 3 - Fields

Create Fields for Hotel Object:

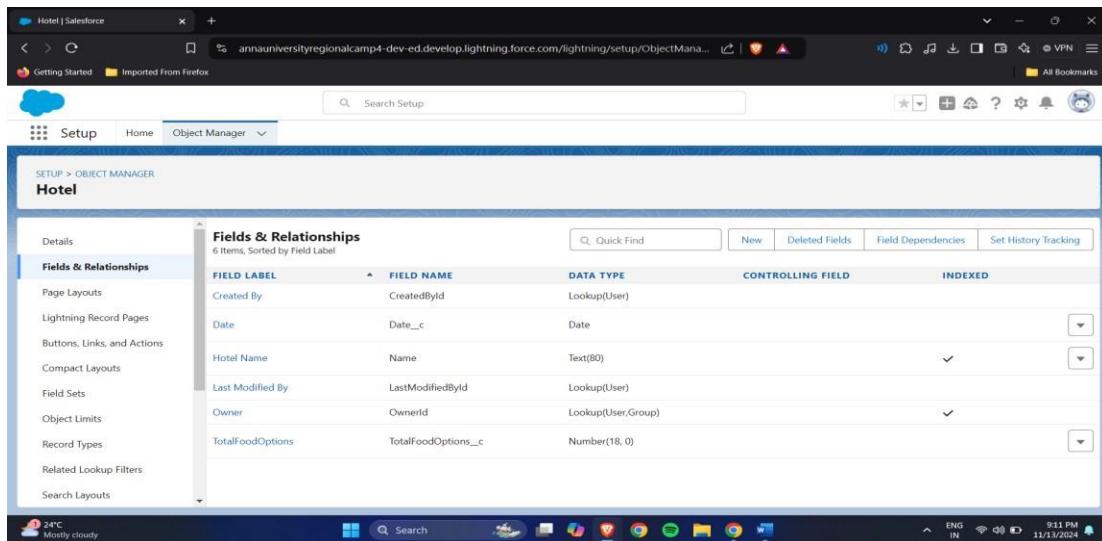
Creating fields for a "Hotel" object involves defining the data attributes that represent essential information about a hotel. These fields should capture the details needed to describe and manage the hotel within an application, database, or any system that tracks hotel information.

Sr. No.	Field Name	Data Type
1	TotalFoodOptions	Number
2	Date	Date



Fields & Relationships of all Hotel Fields:

In a system where you manage hotel data, creating fields and defining relationships for the "Hotel" object is crucial for organizing and retrieving information efficiently. Here's a detailed overview of the fields and relationships typically associated with a "Hotel" object.



Create Fields For Food Option:

Creating fields for a "Food Option" object is essential when building a system to manage food items, such as a restaurant menu or a hotel's food service options. These fields should cover all the necessary details that define each food item and make it easy for users to search, categorize, and manage food options.

Sr. No.	Field Name	Data Type
1	Name	Text
2	Hotel	Hotel(Lookup)
3	Food Amount	Currency

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Food Amount	Food_Amount__c	Currency(18, 0)		
Food Option Name	Name	Auto Number		
Hotel	Hotel_c	Lookup(Hotel)		
Last Modified By	LastModifiedBy	Lookup(User)		
Name	Name_c	Text(255)		
Owner	OwnerId	Lookup(User,Group)		

Create Fields in the Flight Object:

Creating fields in a "Flight" object involves defining essential attributes that represent information about a flight. These fields help to manage and organize flight details within a system for booking, tracking, or scheduling flights. Here's an example of typical fields for a "Flight" object

Sr. No.	Field Name	Data Type
1	Name	Date/Time
2	DepartureDateTime	Hotel(Lookup)

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
ContactEmail	ContactEmail__c	Email		
Created By	CreatedBy	Lookup(User)		
DepartureDateTime	DepartureDateTime__c	Date/Time		
Flight Name	Name	Auto Number		
Last Modified By	LastModifiedBy	Lookup(User)		
Name	Name_c	Lookup(Hotel)		
Owner	OwnerId	Lookup(User,Group)		

Create Fields in the Customer Object:

Creating fields for a "Customer" object involves defining essential details to identify and understand each customer in the system. Here are common fields typically included

Sr. No.	Field Name	Data Type
1	Customer Name	Name
2	Discount Amount	Formula (Currency)
3	Discount Percent	Percentage

The screenshot shows the Salesforce Object Manager interface for the 'Customer' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, Buttons, etc. The main content area is titled 'Fields & Relationships' and displays a table of fields. The table has columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are: Created By (CreatedBy), Customer Name (Customer_Name__c), Customer Name (Name), Discount Amount (Discount_Amount__c), Discount Percent (Discount_Percent__c), Last Modified By (LastModifiedBy), and Owner (OwnerId). The 'Customer Name' field (Name) is indexed.

Milestone 4 -Flow

Flow for Customer Discount Automation:

A Salesforce Flow was created to apply discounts based on the customer's purchase amount. Discounts are granted if the amount exceeds certain thresholds:

Create a new flow variable "TripAdviser".

Flow Variables

Create 3 variable :

Variable > Api name > foId > text > Available for Input

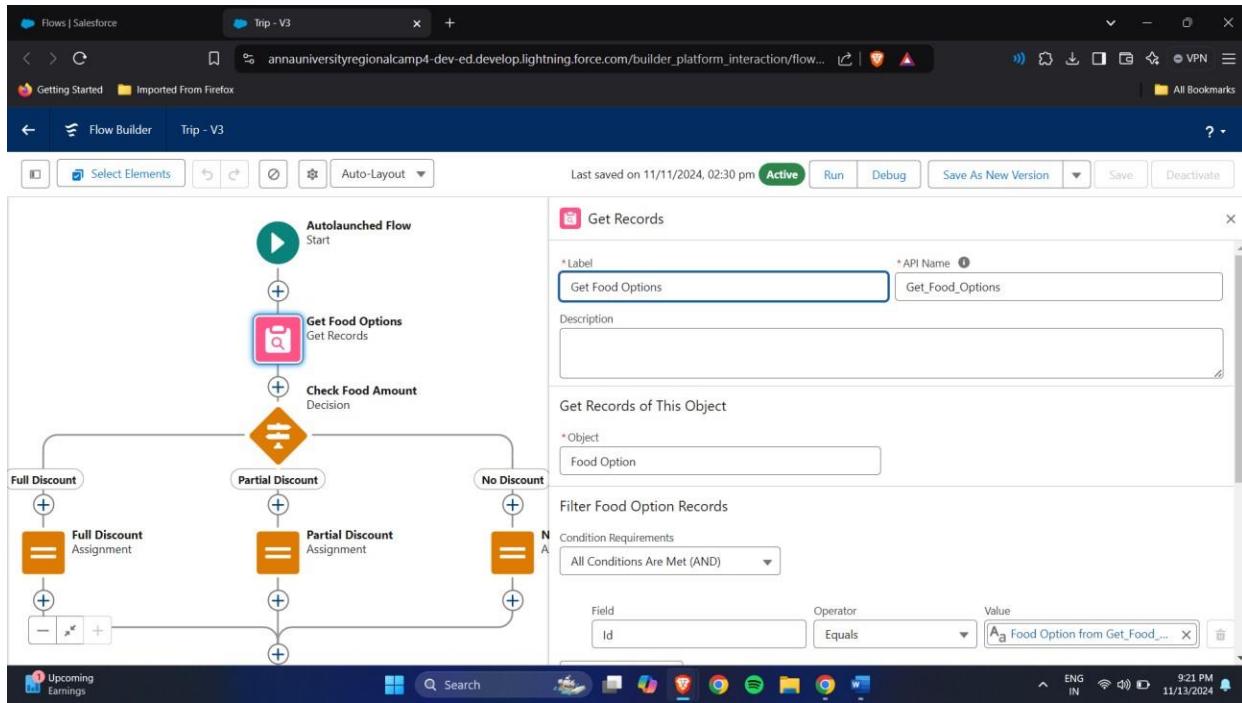
Variable > Api name > csId > text > Available for Input

Variable > Api name > discount > Number

Flow Logic:

Flow Variables are temporary placeholders used within a process or workflow to store and manipulate data as it moves through different stages of execution. These variables enable dynamic data handling, allowing information to be passed from one step to another within a flow.

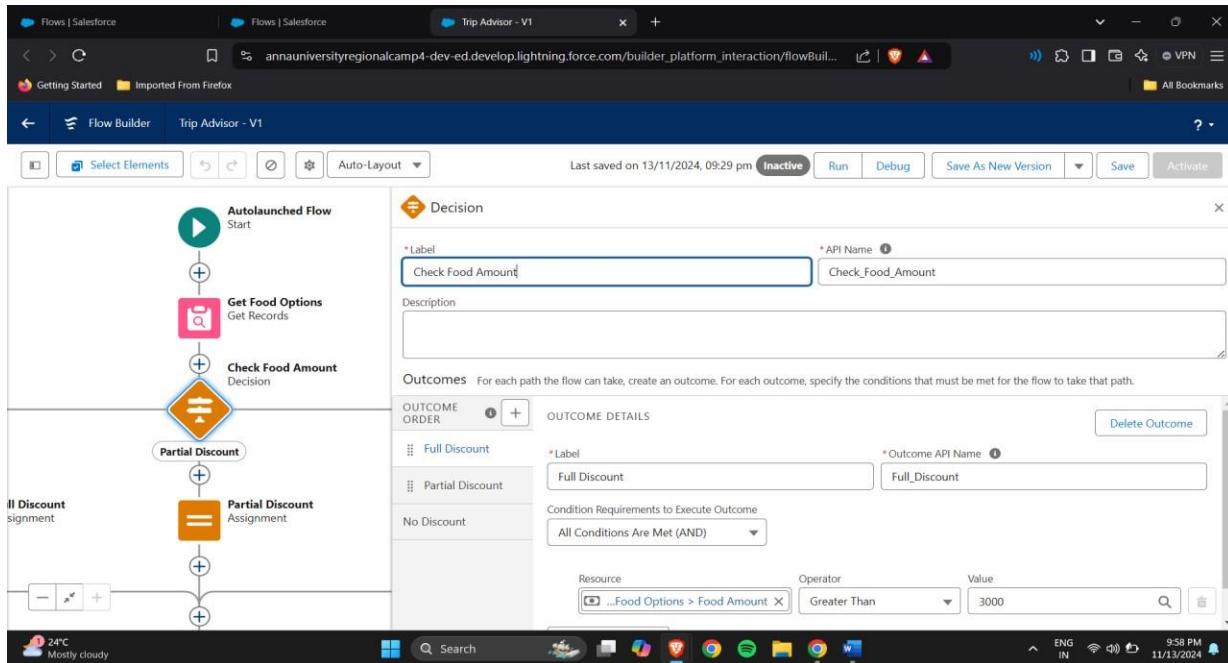
Get Records: Retrieve the necessary customer records.



Purpose:

The "Get Records" element in a flow (such as in Salesforce Flow or similar automation platforms) is to retrieve specific records from a database based on defined criteria. This action allows you to fetch data that can be used later in the flow for various purposes, such as updating records, making decisions, or displaying information.

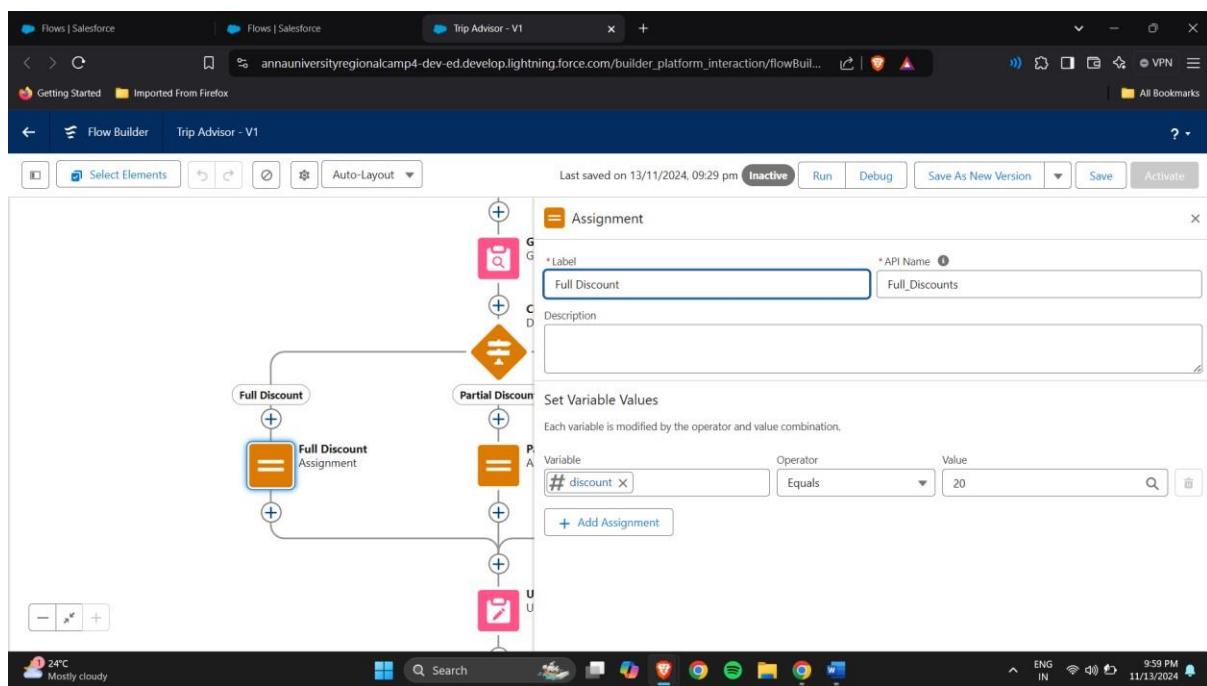
Decision Element: Determine the discount rate based on the purchase amount:



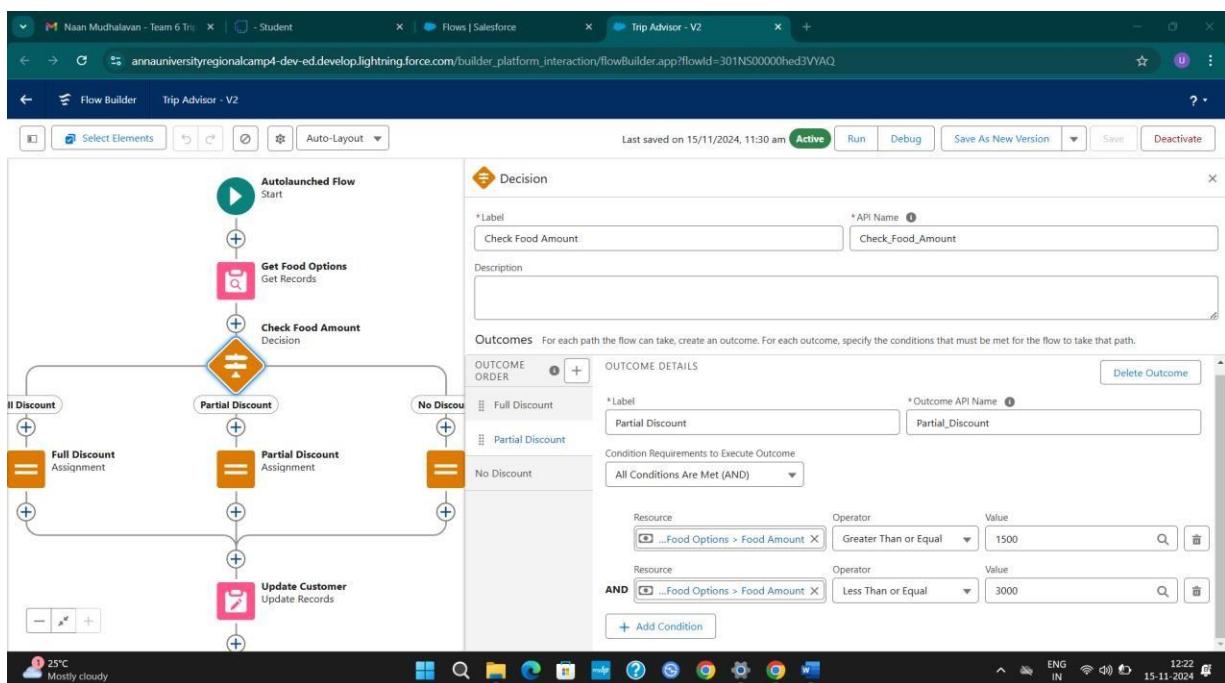
Purpose:

The Decision Element in a flow is used to control the flow's path based on specified conditions. This is particularly useful for creating dynamic, condition-based workflows.

Full Discount: Amount exceeds 3000.



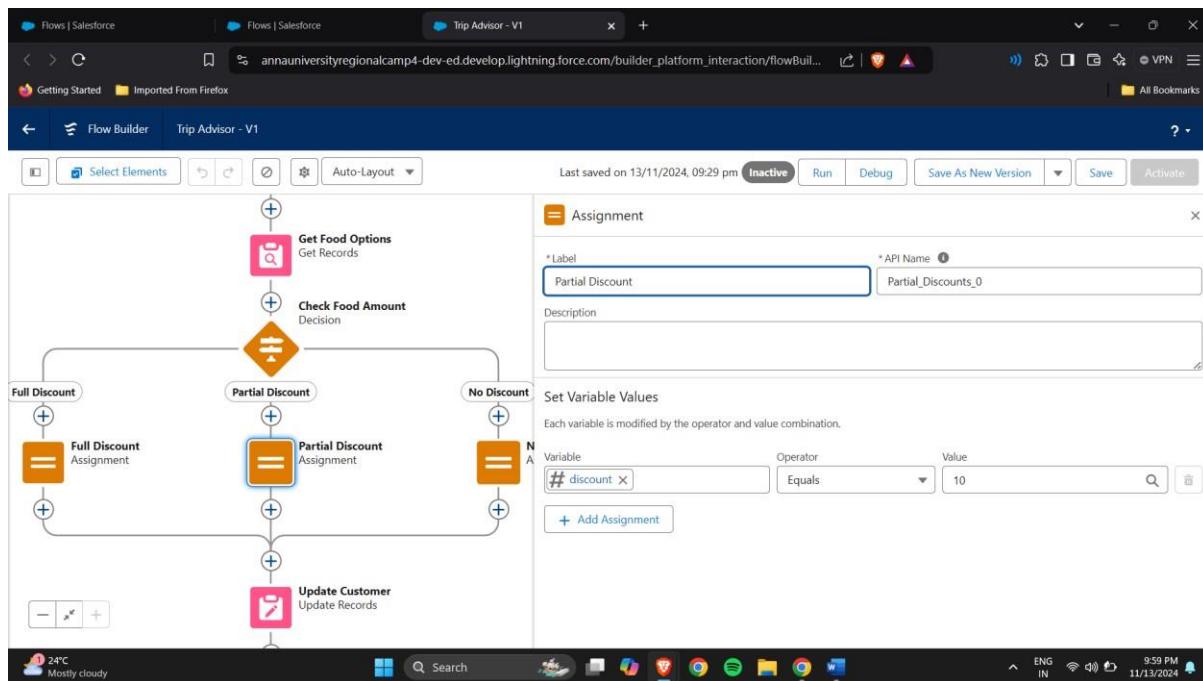
Partial Discount: Amount In-between 1500 to 3000.



Purpose:

The Full Discount in a flow (such as a sales or customer service process) is to Provide Complete Financial Relief, Streamline Issue Resolution, Enhance Customer Loyalty and Support Promotional Strategies.

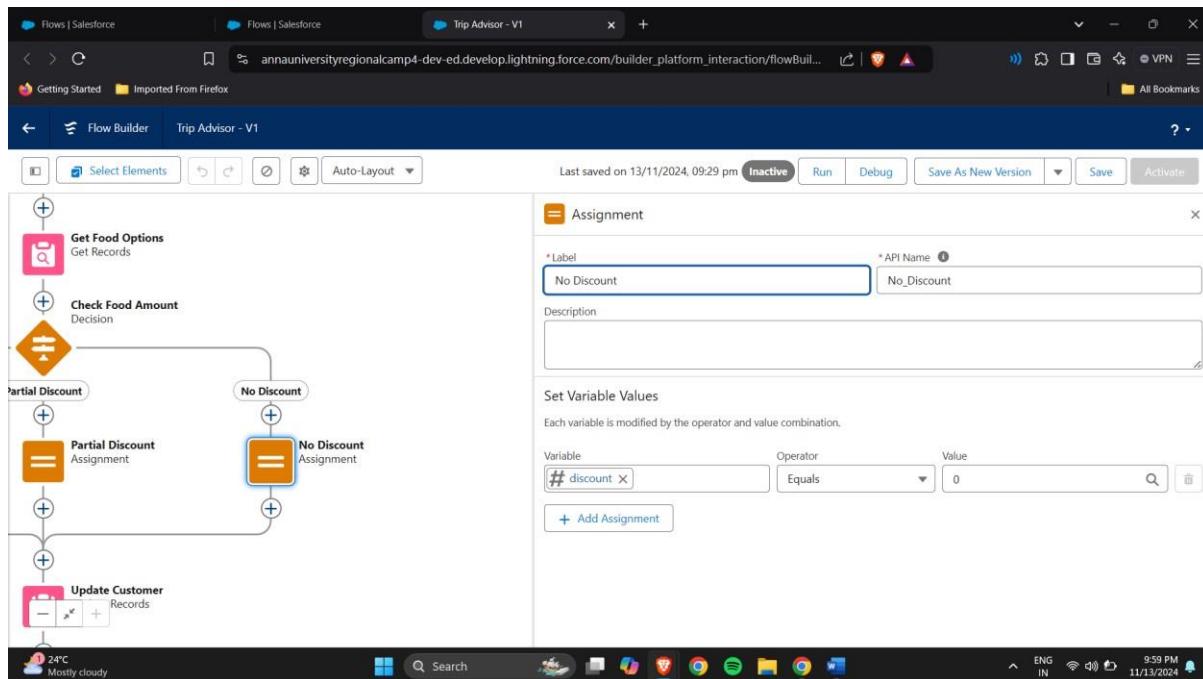
Partial Discount: Amount between 1500 and 3000.



Purpose:

A Partial Discount in a flow (such as an e-commerce or booking process) is to allow a reduction in the total price of a product or service by a specific percentage or amount, rather than a full discount. Partial discounts are often used to incentivize purchases while still maintaining profitability.

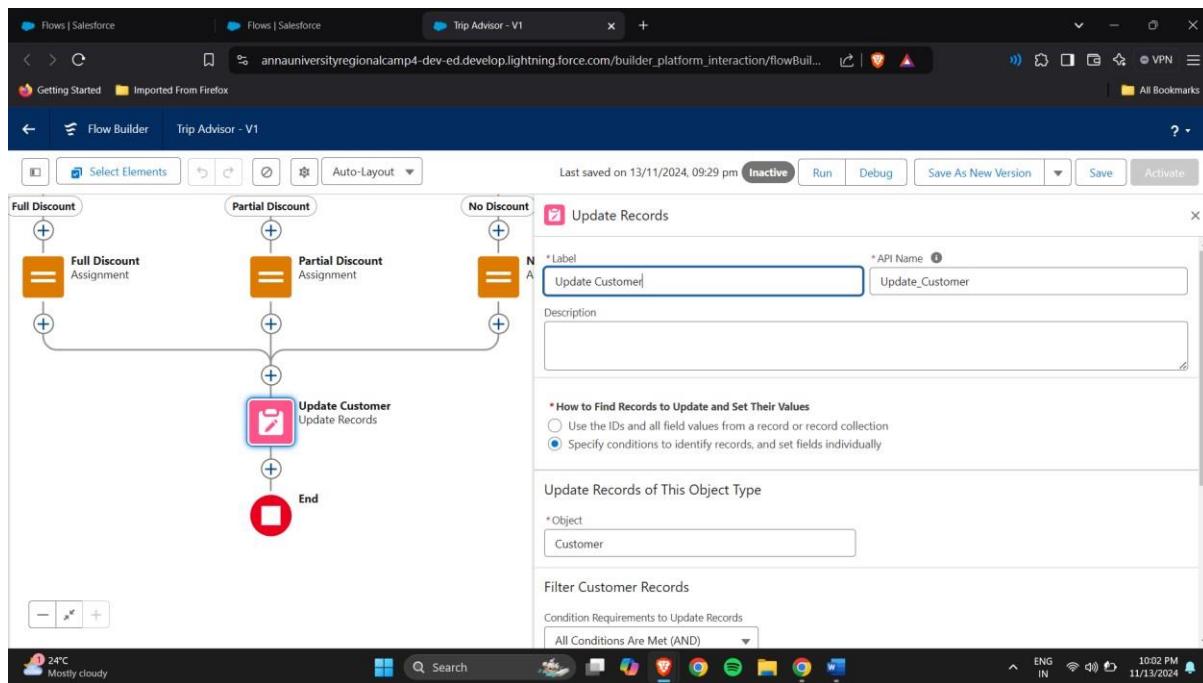
No Discount: Amount below 1500.



Purpose:

A "No Discount" option in a flow is to provide flexibility in scenarios where certain products, services, or customers may not be eligible for any discount.

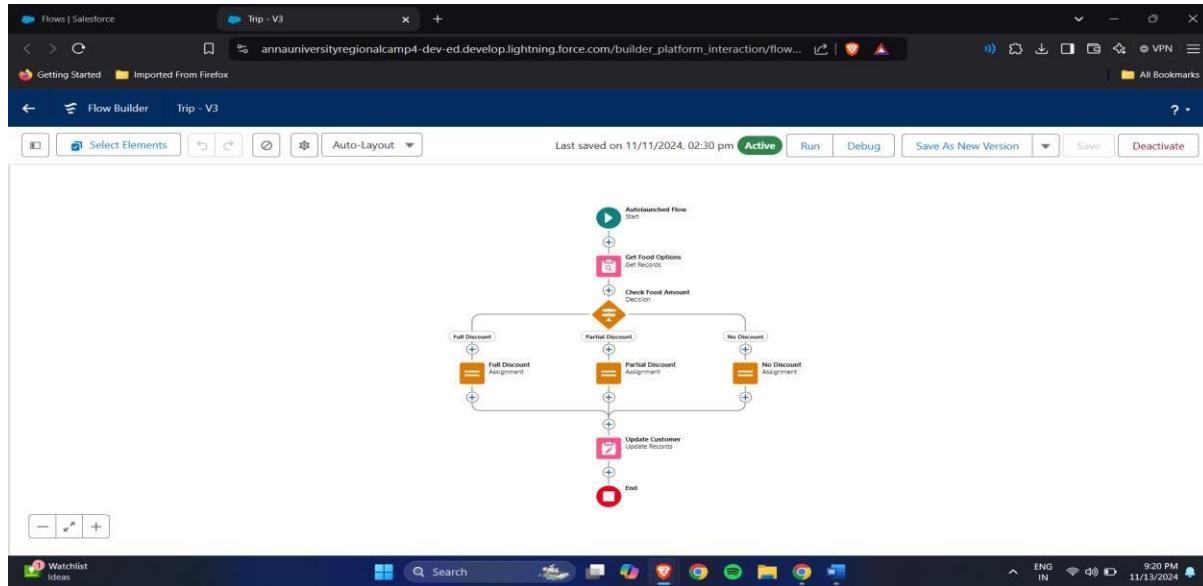
Update Record: Assign discount amounts and update records accordingly.



Purpose:

A "Update Record" element in a flow (commonly in CRM platforms like Salesforce) is to modify existing records in the database based on specified criteria and conditions. This element allows automation of record updates within workflows, saving time and ensuring data consistency.

Final Output of the Flow activate:



Milestone 5 - Tabs:

Tabs in Salesforce provide a user interface for managing and viewing records.

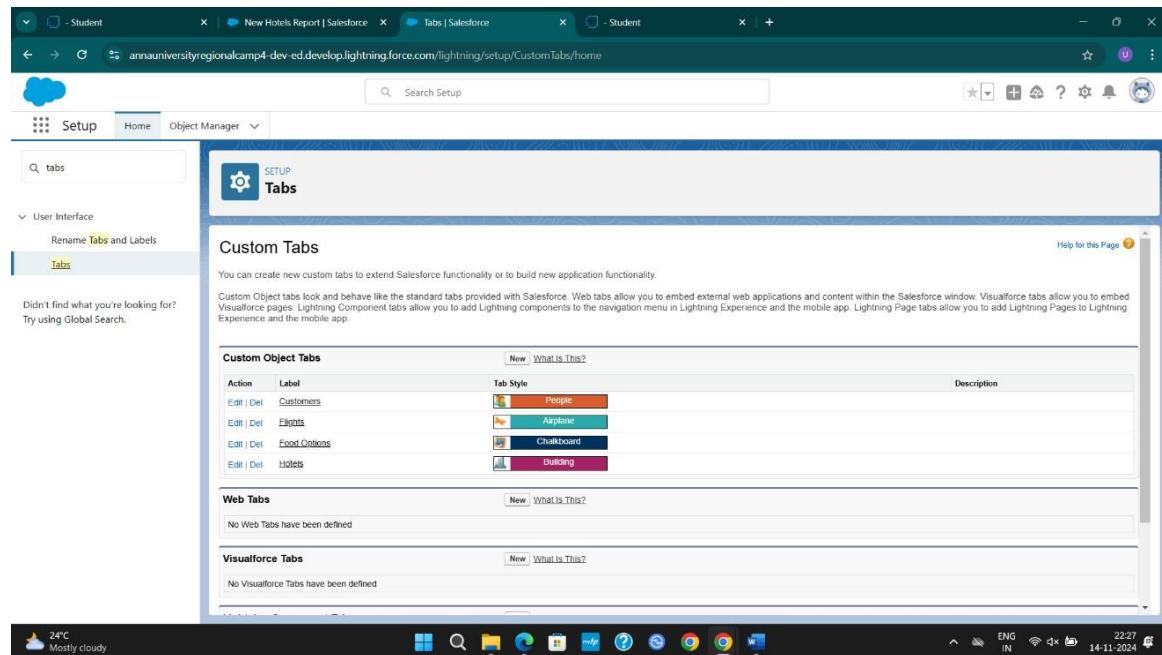
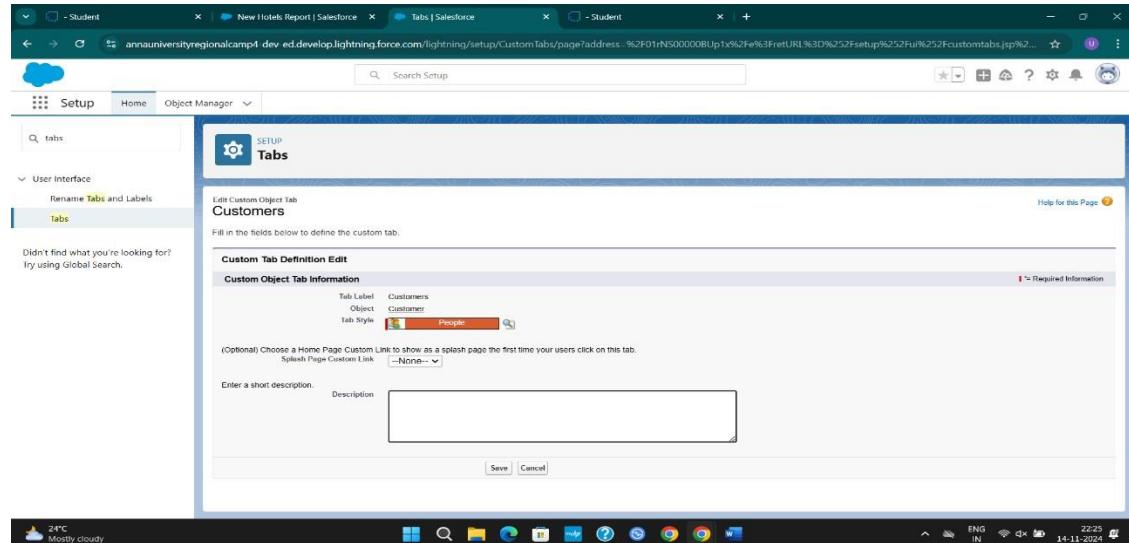
1.Types of Tabs:

- Custom Tabs: Specific to custom objects.
- Web Tabs: Display web content.
- Visualforce Tabs: Display Visualforce pages.
- Lightning Component Tabs: Add Lightning components to the navigation.
- Lightning Page Tabs: Add Lightning Pages to mobile app navigation.

Use Case:

Creating objects and storing TripAdvisor E-management data is the first step to meet their requirements. To enable employees to access stored data efficiently, the admin needs to create dedicated tabs. By designing specific tabs, the organization can enhance the user experience, streamline navigation features. This approach helps employees find and manage data efficiently, supporting better service and operational effectiveness within TripAdvisor E-management

Creating a Custom Tab



- From Setup, search Tabs and select New (Custom Object Tab).
- Choose Opportunity Automobile and complete the setup.

Milestone 6 - Lightning App:

An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps gives users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

Lightning apps let you brand your apps with a custom color and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps.

Use case:

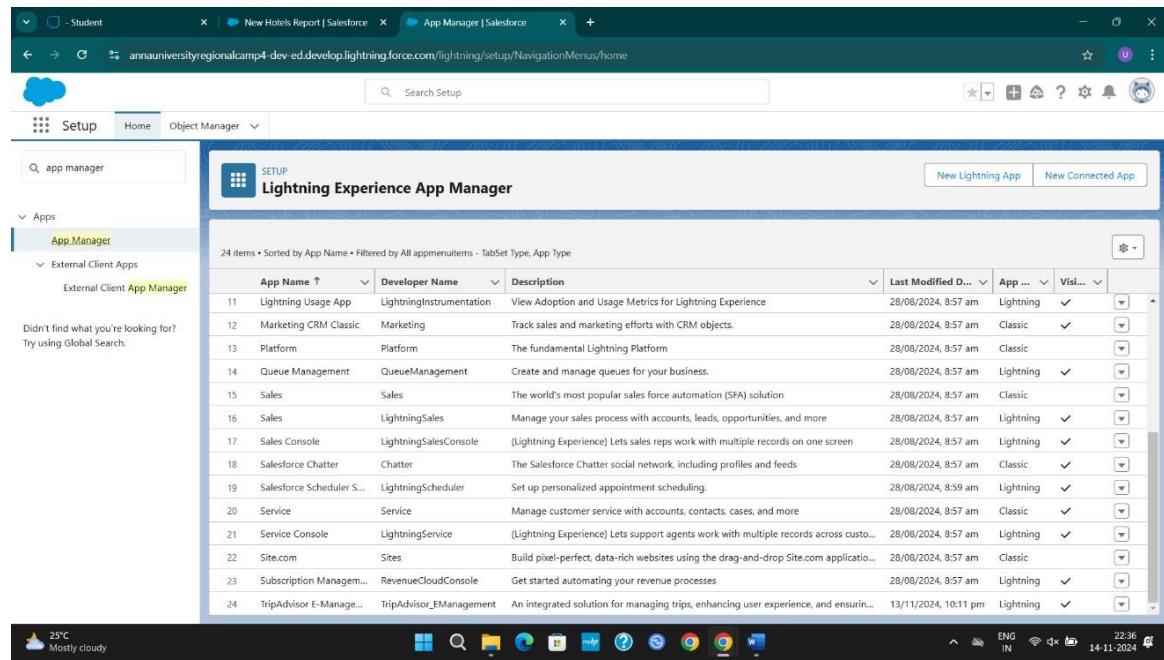
Well done! You're close to meeting the requirements of TripAdvisor E-Management by creating objects to store the organization's data effectively. However, building a database alone is not enough to fully meet organizational needs. The real challenge lies in ensuring that users within TripAdvisor E-Management can easily access and interact with the objects you've created for them.

As the Admin for TripAdvisor E-Management, it's your responsibility to ensure that every user in the organization has appropriate access to the data modeling structure, enabling them to retrieve, update, and manage the data they need seamlessly. This will help TripAdvisor E-Management operate efficiently and provide users with a smooth experience as they engage with the system.

Activity 1:

Create a Lightning App To create a lightning app page:

1. Go to setup page → search “app manager” in quick find → select “app manager” → click on New lightning App.



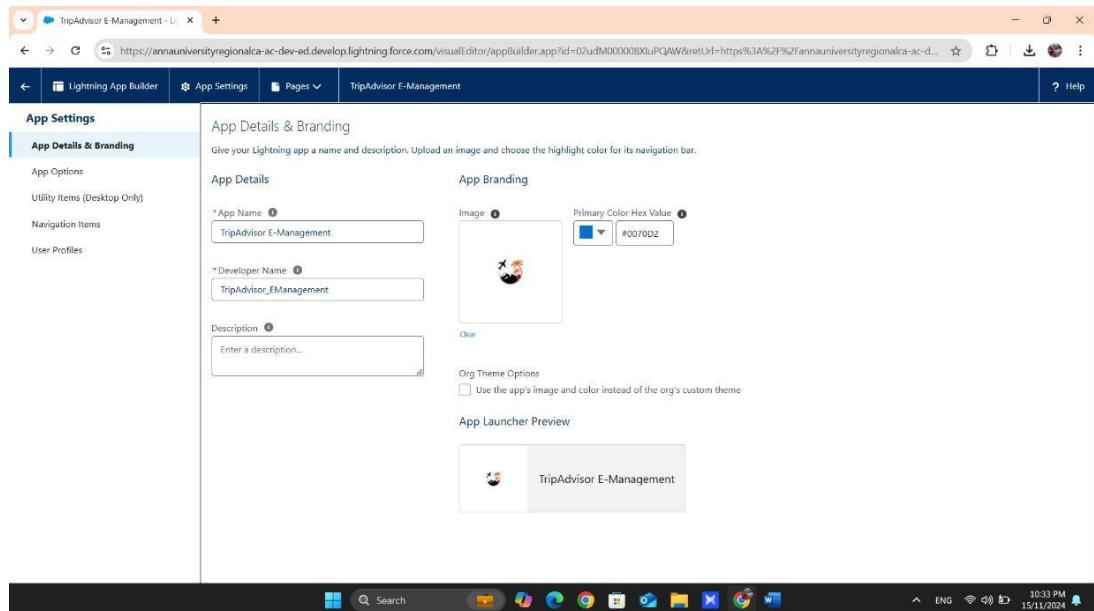
2. Fill the app name in app details and branding as follow

App Name : TripAdvisor E-Management.

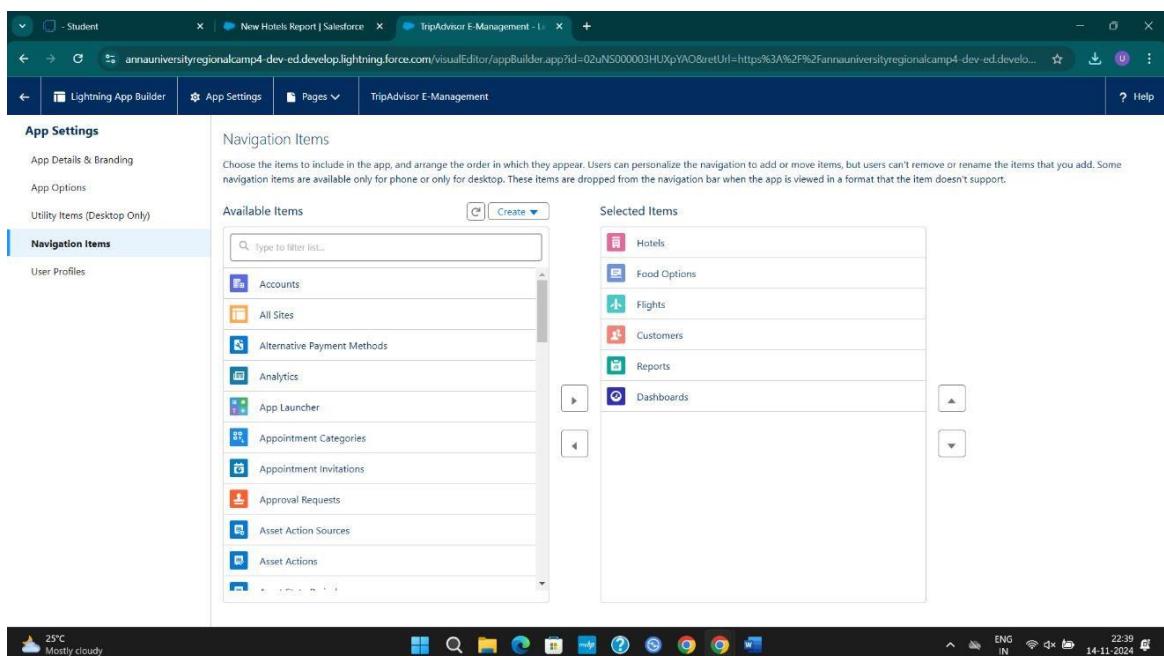
Developer Name : this will auto populated

Description : Give a meaningful description

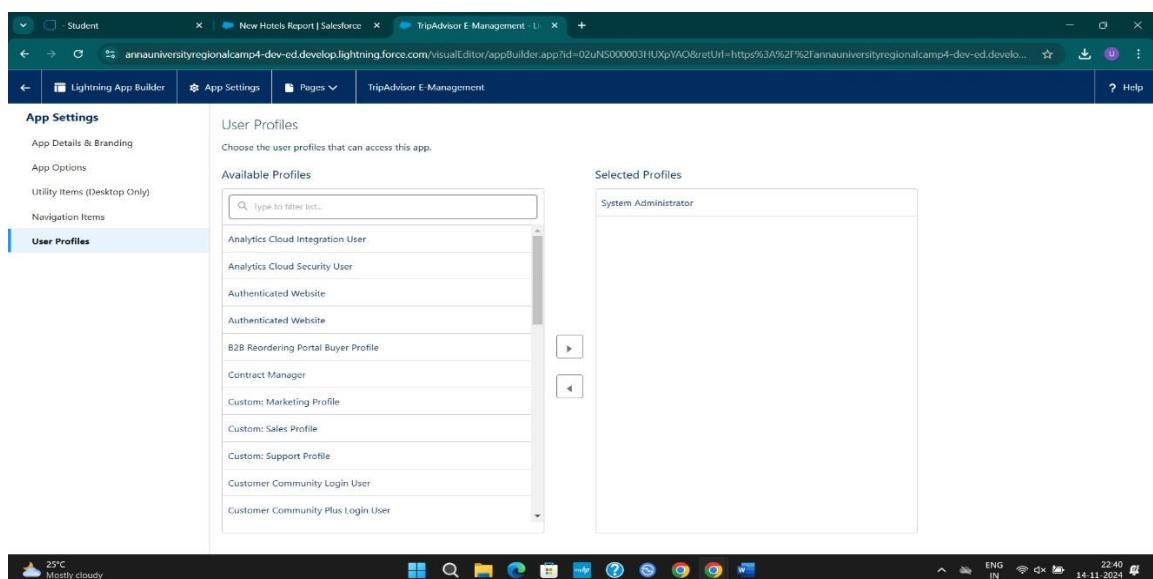
Image : optional (if you want to give any image you can otherwise not mandatory) Primary color hex value : keep this default



3. Then click Next → (App option page) keep it as default → Next → (Utility Items) keep it as default → Next.



4. To Add Navigation Items:



5. Search profiles (System administrator) in the search bar → click on the arrow button → save & finish.

Milestone 7 - OWD Setting:

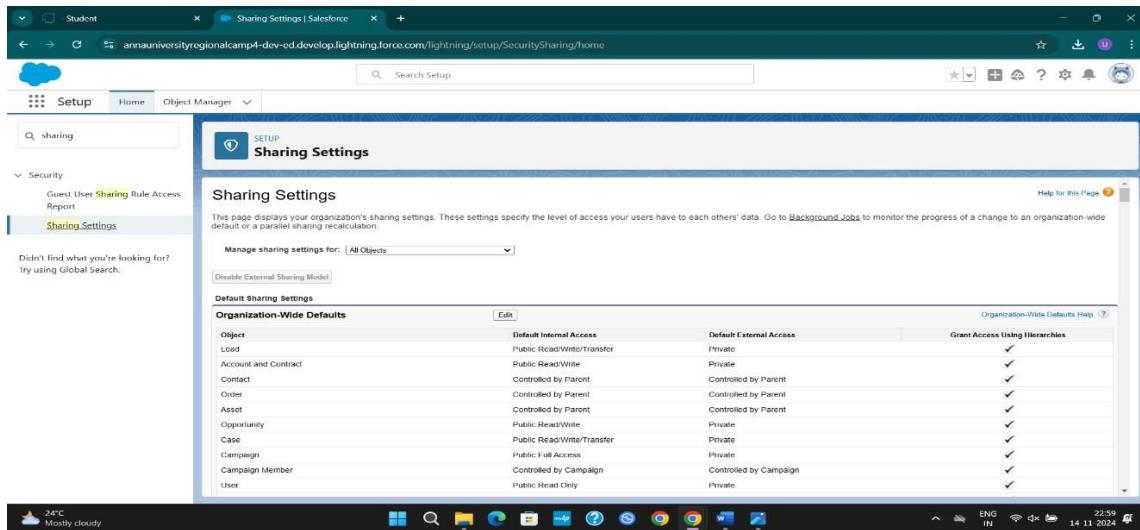
Use case:

TripAdviser E-Management, **Organization-Wide Defaults (OWDs)** are the foundational security settings that determine access to data across the system. OWDs are used to control who can access specific information within the platform. You can extend or restrict access through additional methods such as sharing rules, role hierarchies, team structures, and account groups, as well as manual sharing options.

Activity 1:

Create OWD Setting

1. Go to Set Up → in the Quick Find box type Sharing Settings → click on it.
2. Click Edit in the Organization-Wide Defaults area.



3. Search for the Employee object.

4. Under default internal access and default external access change the options to “Private” and under grant access using hierarchies select the check box.

5. Click on save.

A screenshot of the 'Other Settings' section of the Sharing Settings page. It shows a table with four rows corresponding to objects: Customer, Flight, Food Option, and Hotel. Each row has three columns: 'Default Internal Access' (dropdown menus set to 'Private'), 'Default External Access' (dropdown menus set to 'Private'), and 'Grant Access Using Hierarchies' (checkboxes all checked). Below this table are two checkboxes: 'Standard Report Visibility' (checked) and 'Manual User Record Sharing' (unchecked). At the bottom are 'Save' and 'Cancel' buttons.

6. This Setting is for all the Users Which have been Created.

Milestone 8 - User Adoption:

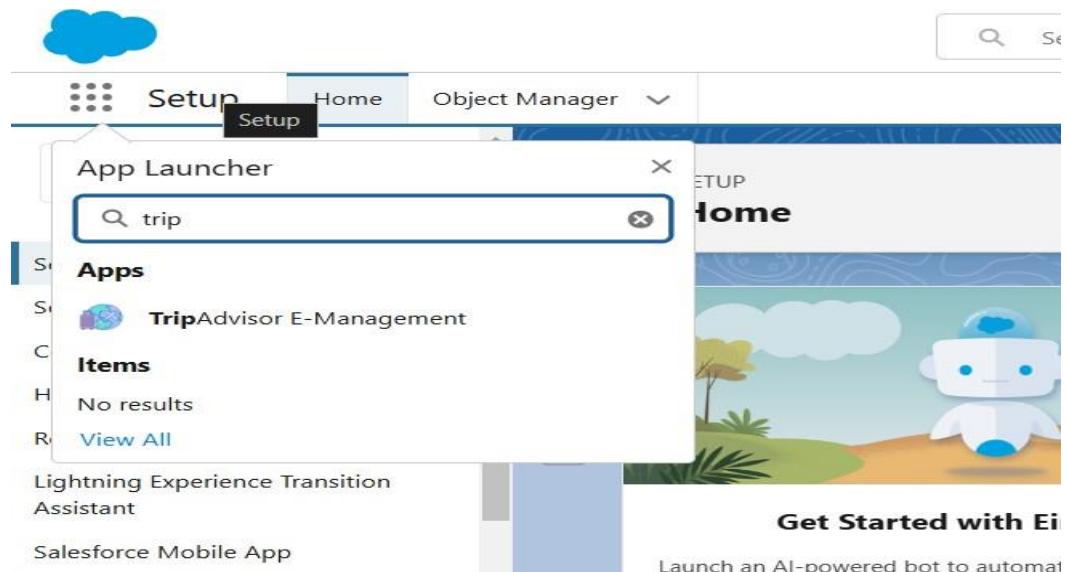
Use Case:

As a new Administrator in the TripAdvisor E-management system, you handle user management tasks such as creating and editing user accounts, resetting passwords, assigning permissions, configuring access to travel data, and more. In this unit, you will learn about managing users and adding them to your TripAdvisor E-management platform.

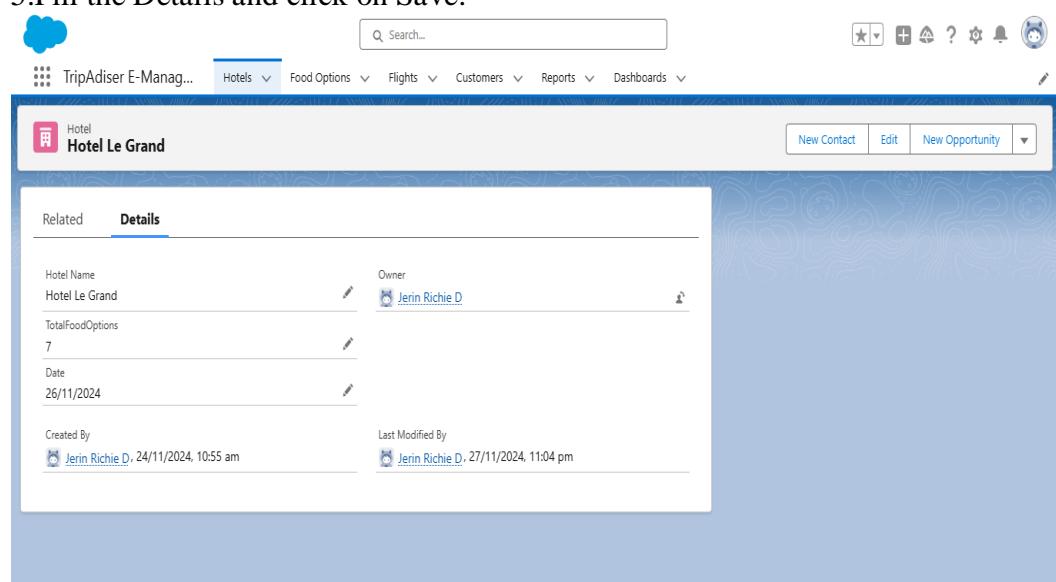
Activity 1:

Create a Record (Employee)

1. Click on App Launcher on the left side of the screen.
2. Search Employee Management System & click on it.



3. Click on the Employee tab.
4. Click New.
5. Fill the Details and click on Save.



Activity 4:

Create at least 10 records for each of the objects: Hotel, Flight, Customers, Food Options.

Hotel Name:

The screenshot shows a software interface for managing travel-related data. At the top, there's a navigation bar with icons for cloud, user, and various menu items like 'Hotels', 'Food Options', 'Flights', 'Customers', 'Reports', and 'Dashboards'. Below the navigation is a search bar with placeholder 'Search...' and a 'Recently Viewed' section titled 'Hotels'. This section displays a list of 7 items, all updated a few seconds ago. The list includes:

Index	Hotel Name
1	Hotel Le Grand
2	Starry Sky Cabins
3	Eco Elite
4	The Acacia Hotel
5	Holiday Dream Resorts
6	High-Sky Abode
7	Ocean Heaven Hotel

At the bottom right of the interface, there are buttons for 'New', 'Import', 'Change Owner', and 'Assign Label'.

Flight Name:

This screenshot shows the same software interface as the previous one, but the 'Recently Viewed' section is now titled 'Flights'. It displays a list of 4 items, all updated a few seconds ago. The list includes:

Index	Flight Name
1	FL- 0007
2	FL- 0006
3	FL- 0005
4	FL- 0004

Customer Name:

This screenshot shows the 'Customers' section of the software. The 'Recently Viewed' section is titled 'Customers' and displays a list of 5 items, all updated a few seconds ago. The list includes:

Index	Customer Name
1	Divya
2	Leela
3	Jerin
4	Abinaya
5	Hema

Food Options Name:

The screenshot shows a software interface titled "Food Options" under the "Recently Viewed" section. The list contains 7 items, all labeled "FO - 00XX".

Food Option Name
FO - 0003
FO - 0008
FO - 0005
FO - 0001
FO - 0002
FO - 0004
FO - 0006

Milestone 8 - Apex Trigger

Apex Trigger for Food Options Management

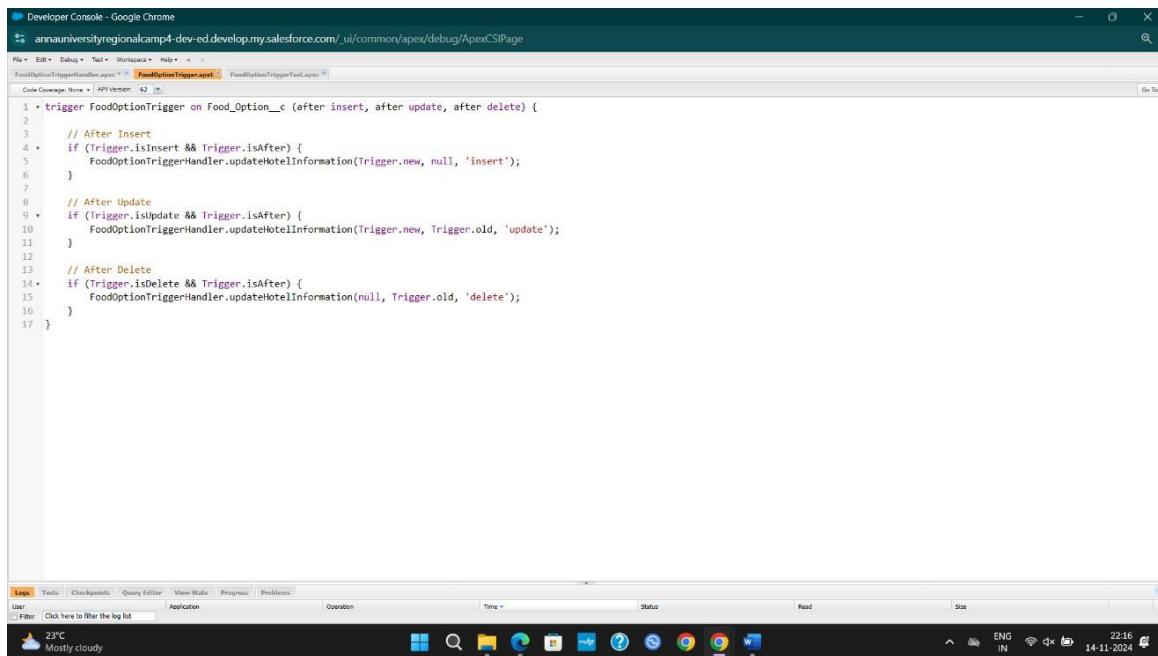
An Apex Trigger was implemented to update hotel information whenever a new food option is added or updated. This ensures the hotel's total food options count reflects all associated food options.

Apex Trigger Handler:

```
1 * public class FoodOptionTriggerHandler {
2     *     public static void updateHotelInformation(List<Food_Option__c> newFoodOptions, List<Food_Option__c> oldFoodOptions, String operation) {
3         Set<Id> hotelIdsToUpdate = new Set<Id>();
4         // Collect unique Hotel IDs from new Food Options (insert or update)
5         if (newFoodOptions != null) {
6             for (Food_Option__c foodOption : newFoodOptions) {
7                 if (foodOption.Hotel__c != null) {
8                     hotelIdsToUpdate.add(foodOption.Hotel__c);
9                 }
10            }
11        }
12        // Collect unique Hotel IDs from old Food Options (update or delete)
13        if (oldFoodOptions != null) {
14            for (Food_Option__c foodOption : oldFoodOptions) {
15                if (foodOption.Hotel__c != null) {
16                    hotelIdsToUpdate.add(foodOption.Hotel__c);
17                }
18            }
19        }
20        if (hotelIdsToUpdate.isEmpty()) {
21            return;
22        }
23        // Query the affected Hotel records
24        List<Hotel__c> hotelsToUpdate = [SELECT Id, TotalFoodOptions__c FROM Hotel__c WHERE Id IN :hotelIdsToUpdate];
25        // Recalculate the total food options count for each hotel
26        for (Hotel__c hotel : hotelsToUpdate) {
27            Integer totalFoodOptions = [SELECT COUNT() FROM Food_Option__c WHERE Hotel__c = :hotel.Id];
28            hotel.TotalFoodOptions__c = totalFoodOptions;
29        }
30        // Update the Hotel records with the new total count
31        if (!hotelsToUpdate.isEmpty()) {
32            update hotelsToUpdate;
33        }
34    }
35 }
```

An **Apex Trigger Handler** is a design pattern used to organize and manage the logic of an Apex trigger. It helps in maintaining clean, reusable, and easily maintainable code. Instead of placing the logic directly within the trigger, it delegates it to a handler class.

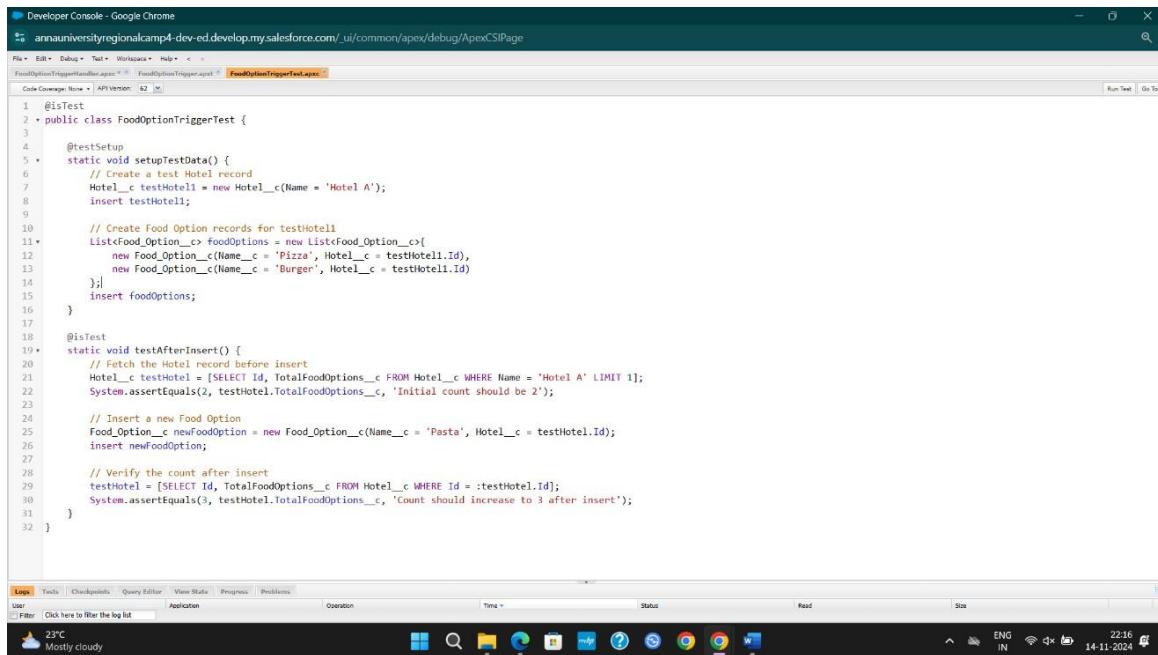
Trigger:



```
1 trigger FoodOptionTrigger on Food_Option__c (after insert, after update, after delete) {
2
3     // After Insert
4     if (Trigger.isInsert && Trigger.isAfter) {
5         FoodOptionTriggerHandler.updateHotelInformation(Trigger.new, null, 'insert');
6     }
7
8     // After Update
9     if (Trigger.isUpdate && Trigger.isAfter) {
10        FoodOptionTriggerHandler.updateHotelInformation(Trigger.new, Trigger.old, 'update');
11    }
12
13     // After Delete
14    if (Trigger.isDelete && Trigger.isAfter) {
15        FoodOptionTriggerHandler.updateHotelInformation(null, Trigger.old, 'delete');
16    }
17 }
```

An **Trigger** in Salesforce is a piece of code that automatically executes (or "fires") when a specific event occurs on a record in Salesforce, such as creating, updating, or deleting a record. It allows developers to add custom logic to standard operations, providing more control over data and business processes.

Test Trigger:



```
1 @isTest
2 public class FoodOptionTriggerTest {
3
4     @testSetup
5     static void setupTestData() {
6         // Create a test Hotel record
7         Hotel__c testHotel1 = new Hotel__c(Name = 'Hotel A');
8         insert testHotel1;
9
10        // Create Food Option records for testHotel1
11        List<Food_Option__c> foodOptions = new List<Food_Option__c>{
12            new Food_Option__c(Name__c = 'Pizza', Hotel__c = testHotel1.Id),
13            new Food_Option__c(Name__c = 'Burger', Hotel__c = testHotel1.Id)
14        };
15        insert foodOptions;
16    }
17
18    @isTest
19    static void testAfterInsert() {
20        // Fetch the Hotel record before insert
21        Hotel__c testHotel = [SELECT Id, TotalFoodOptions__c FROM Hotel__c WHERE Name = 'Hotel A' LIMIT 1];
22        System.assertEquals(2, testHotel.TotalFoodOptions__c, 'Initial count should be 2');
23
24        // Insert a new Food Option
25        Food_Option__c newFoodOption = new Food_Option__c(Name__c = 'Pasta', Hotel__c = testHotel.Id);
26        insert newFoodOption;
27
28        // Verify the count after insert
29        testHotel = [SELECT Id, TotalFoodOptions__c FROM Hotel__c WHERE Id = :testHotel.Id];
30        System.assertEquals(3, testHotel.TotalFoodOptions__c, 'Count should increase to 3 after insert');
31    }
32 }
```

A **Test Trigger** in Salesforce is used to validate that the trigger behaves as expected under different conditions. It is written using Apex test methods to simulate various scenarios, ensuring that triggers perform the correct operations, like inserting, updating, or deleting records.

Test case Result:

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is https://annauiversityregionalcamp4-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tabs at the top are "FoodOptionTriggerHandler.apxc", "FoodOptionTrigger.apxt", and "FoodOptionTriggerTest.apxc". The code editor contains an Apex test class:

```
1 @isTest
2 public class FoodOptionTriggerTest {
3
4     @testSetup
5     static void setupTestData() {
6         // Create a test Hotel record
7         Hotel__c testHotel1 = new Hotel__c(Name = 'Hotel A');
8         insert testHotel1;
9
10        // Create Food Option records for testHotel1
11        List<Food_Option__c> foodOptions = new List<Food_Option__c>{
12            new Food_Option__c(Name__c = 'Pizza', Hotel__c = testHotel1.Id),
13            new Food_Option__c(Name__c = 'Burger', Hotel__c = testHotel1.Id)
14        };
15        insert foodOptions;
16    }
17
18    @isTest
19    static void testAfterInsert() {
20        // Fetch the Hotel record before insert
21    }
}
```

The "Logs" tab is selected, showing a single test run:

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	707N50000003yO4	Thu Nov 14 2024 22:14:15 GM...	0:00	0	1
✓	707N50000003yLo	Thu Nov 14 2024 22:16:44 GM...	0:00	0	1

The status bar at the bottom shows "22°C Mostly cloudy" and the date "14-11-2024".

A "Test Trigger case run successfully" means that a trigger (an automated process or function) was executed, and it completed without errors or failures.

- **Trigger:** A piece of code that runs automatically in response to specific events (like creating, updating, or deleting a record).
- **Test Case:** A scenario designed to verify that the trigger works as expected under certain conditions.
- **Successful Run:** The trigger was executed correctly, and the desired results were achieved, with no issues encountered during testing (e.g., data was updated correctly, no errors occurred).

Milestone 10 - Apex Scheduled

Apex Scheduled Class for Flight Reminders

The Apex Scheduled class, `FlightReminderScheduledJob`, was created to send reminder emails to customers who have booked flights, 24 hours prior to departure.

Scheduled Class Code

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is https://annauiversityregionalcamp4-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tabs at the top are "FlightReminderScheduledJob.apxc" and "FlightReminderScheduledJob.apxt". The code editor contains an Apex scheduled class:

```
1 public class FlightReminderScheduledJob implements Schedulable {
2
3     public void execute(SchedulableContext sc) {
4         sendFlightReminders();
5     }
6
7     private void sendFlightReminders() {
8         // Query for flights departing within the next 24 hours
9         List<Flight__c> upcomingFlights = [SELECT Id, Name, DepartureDateTime__c FROM Flight__c
10                                         WHERE DepartureDateTime__c >= :DateTime.now()
11                                         AND DepartureDateTime__c <= :DateTime.now().addDays(1)];
12
13     for (Flight__c flight : upcomingFlights) {
14         // Customize the logic to send reminder emails
15         // For this example, we'll print a log message; replace this with your email sending logic.
16         System.debug('Sending reminder email for Flight ' + flight.Name + ' to ' + flight.ContactEmail__c);
17
18         // Example: Send email using Messaging.SingleEmailMessage
19         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
20         email.setToAddresses(new List<String>{ flight.ContactEmail__c });
21         email.setSubject('Flight Reminder: ' + flight.Name);
22         email.setPlainTextBody('This is a reminder for your upcoming flight ' + flight.Name +
23                               ' departing on ' + flight.DepartureDateTime__c);
24    }
}
```

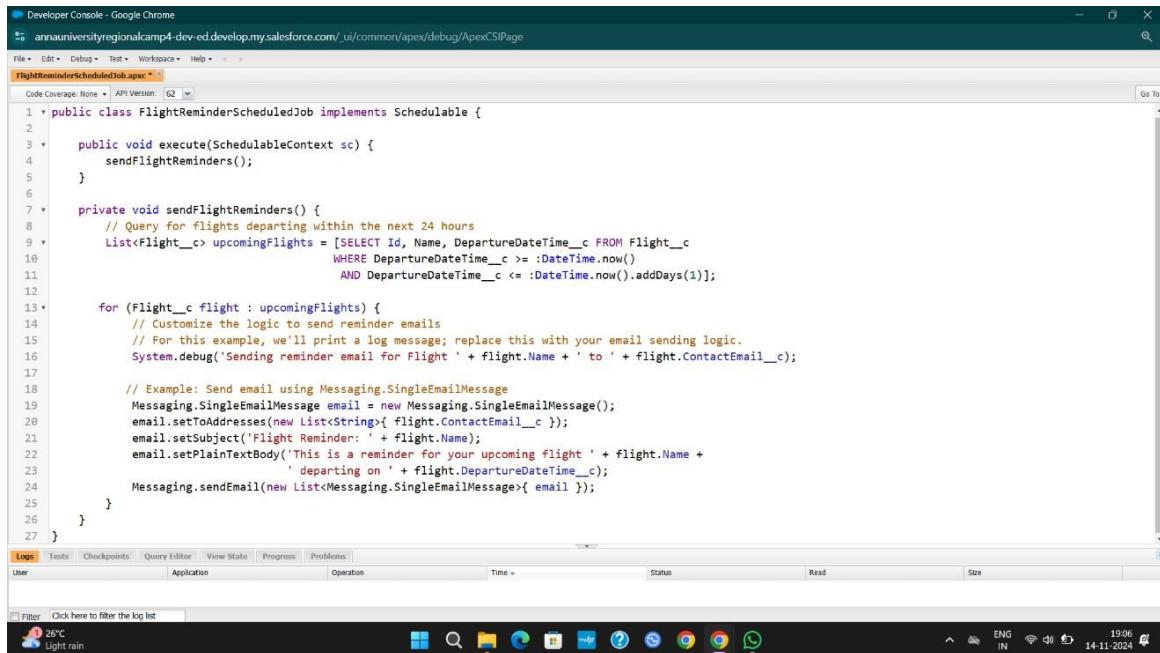
The "Logs" tab is selected, showing the execution log:

User	Application	Operation	Time	Status	Read	Size
Umar Ahmed Khan A	Unknown	/services/data/v52.0/tooling/executeAnonymous	11/14/2024, 7:12:54 PM	Success		3.24 KB

The status bar at the bottom shows "26°C Light rain" and the date "14-11-2024".

A **Scheduled Class** in platforms like Salesforce allows you to automate and schedule the execution of Apex classes at specified times or intervals. It is particularly useful for recurring tasks, such as sending emails, updating records, or integrating external systems, without manual intervention.

Scheduling the Job



```

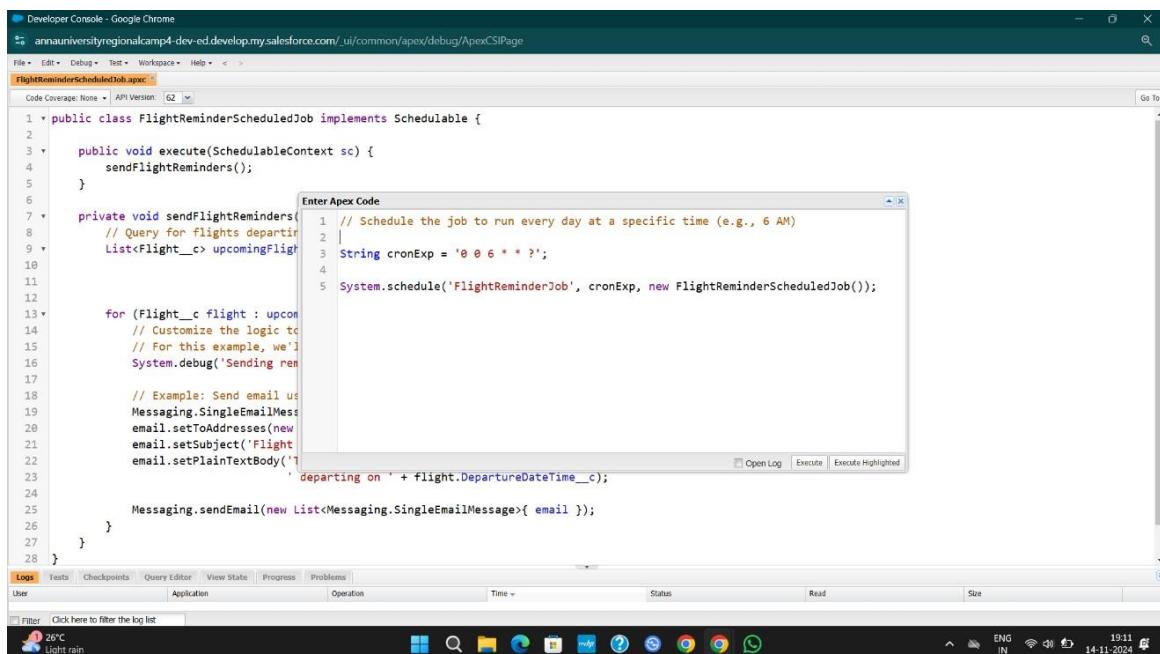
1 * public class FlightReminderScheduledJob implements Schedulable {
2
3     public void execute(SchedulableContext sc) {
4         sendFlightReminders();
5     }
6
7     private void sendFlightReminders() {
8         // Query for flights departing within the next 24 hours
9         List<Flight__c> upcomingFlights = [SELECT Id, Name, DepartureDateTime__c FROM Flight__c
10                                         WHERE DepartureDateTime__c >= :DateTime.now()
11                                         AND DepartureDateTime__c <= :DateTime.now().addDays(1)];
12
13     for (Flight__c flight : upcomingFlights) {
14         // Customize the logic to send reminder emails
15         // For this example, we'll print a log message; replace this with your email sending logic.
16         System.debug('Sending reminder email for Flight ' + flight.Name + ' to ' + flight.ContactEmail__c);
17
18         // Example: Send email using Messaging.SingleEmailMessage
19         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
20         email.setToAddresses(new List<String>{ flight.ContactEmail__c });
21         email.setSubject('Flight Reminder: ' + flight.Name);
22         email.setPlainTextBody('This is a reminder for your upcoming flight ' + flight.Name +
23                               ' departing on ' + flight.DepartureDateTime__c);
24         Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{ email });
25     }
26 }
27

```

The screenshot shows the Salesforce Developer Console interface. The code editor displays the `FlightReminderScheduledJob.apxc` file. The code itself is a scheduled Apex class that queries for upcoming flights and sends a reminder email for each. The developer has added a `System.debug` statement to log the recipient's contact email. The browser title bar indicates the session is in Google Chrome on the Anna University regional camp4 dev-ed instance.

Scheduling a Job Code refers to setting up an automated process to run at a specific time or interval without manual intervention. This is commonly used in systems like CRM platforms, databases, or job scheduling tools. The purpose is to perform repetitive tasks (e.g., data updates, reports generation) at predefined times.

Output of the job Scheduling:



The screenshot shows the Salesforce Developer Console with the same `FlightReminderScheduledJob.apxc` file open. A modal dialog box titled "Enter Apex Code" is displayed over the code editor. The code within the dialog is as follows:

```

1 // Schedule the job to run every day at a specific time (e.g., 6 AM)
2 |
3 String cronExp = '0 0 6 * * ?';
4
5 System.schedule('FlightReminderJob', cronExp, new FlightReminderScheduledJob());

```

This code defines a cron expression to run the scheduled job daily at 6 AM. The developer has also added a `System.schedule` call to the end of the `sendFlightReminders` method. The browser title bar and system tray are visible at the bottom of the screen.

A Scheduled Job Code is a task or process that runs automatically at predefined times or intervals. When it runs successfully, it means the scheduled job has executed without errors, completing the intended task as per the schedule.

All job scheduled has been updated:

The screenshot shows the Salesforce Setup interface with the 'Scheduled Jobs' page open. The left sidebar includes sections like Feature Settings, Sales, Products, Salesforce Scheduler, Environments, and Jobs, with 'Scheduled Jobs' selected. The main content area displays a table of scheduled jobs with columns for Action, Job Name, Submitted By, Submitted, Started, Next Scheduled Run, Type, and Cron Trigger ID. A message at the top indicates that 1% of scheduled jobs have been used.

Action	Job Name	Submitted By	Submitted	Started	Next Scheduled Run	Type	Cron Trigger ID
Manage Del Pause Job	FlightReminderJob	A. Umar Ahmed Khan	14/11/2024, 7:12 pm		15/11/2024, 6:00 am	Scheduled Apex	08eNS000003o9fP
Del	Metalytics Data Loader Job for Org : 00DN5000002eiHJ	User_Integration	28/08/2024, 8:59 am	14/11/2024, 3:47 am	15/11/2024, 3:47 am	Autonomous Data Loader Job	08eNS000000yxbz
	Program Milestone Computation Cron Job	Process, Automated	28/08/2024, 8:59 am	14/11/2024, 1:29 pm	14/11/2024, 8:29 pm	Program Milestone Computation Cron Job	08eNS000000yxtb
	Program Status Update Cron Job	Process, Automated	28/08/2024, 8:59 am	14/11/2024, 6:30 pm	15/11/2024, 9:30 am	Program Status Update Cron Job	08eNS000000yxtv

"All jobs scheduled have been updated" means that the tasks or processes that were planned or set to run at specific times have been modified or refreshed. This could involve changing the timing, parameters, or details of the scheduled jobs to ensure they align with new requirements, improve efficiency, or reflect updated information.

Milestone 11 - Reports:

Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

Types of Reports in Salesforce

1. Tabular
2. Summary
3. Matrix
4. Joined Reports

Use Case:

The CEO of TripAdvisor wants a concise overview of employee activities, current projects, project progress, and assigned assets, along with the condition of these assets. This data will provide a clear snapshot of the organization's status, enabling informed decision-making. The CEO has tasked you with presenting this information in an organized, easily interpretable format.

Let's create a Report.

Activity 1:

Create Report

1. Go to the app → click on the reports tab 2. Click New Report
2. Select report type from category or from report type panel or from search panel → click on start

report.

Create Report

Category

Recently Used

All

Accounts & Contacts

Opportunities

Customer Support Reports

Leads

Campaigns

Activities

Contracts and Orders

Price Books, Products and Assets

Administrative Reports

File and Content Reports

Individuals

Select a Report Type

Showing results for hotel

Report Type Name Category

Hotels Standard

Food Options with Hotel Standard

Details

Hotels Standard Report Type

Start Report

Created By You

New Hotels Report

Created By Others

No Reports Yet

Objects Used in Report Type

Owner

4. Customize your report and add fields from left pane as shown below

5. Save or run it.

Reports for Hotel, Flight, Customer, and Food Option have been created using a standardized method, streamlining data management and analysis for improved decision-making and business operations.

New Hotel Report:

Search...

TripAdiser E-Manag...

Hotels Food Options Flights Customers Reports Dashboards

Report: Hotels

New Hotels Report

Total Records 7

	Hotel: Hotel Name	Date
1	Ocean Heaven Hotel	-
2	Starry Sky Cabins	-
3	High-Sky Abode	-
4	Holiday Dream Resorts	-
5	The Acacia Hotel	-
6	Hotel Le Grand	26/11/2024
7	Eco Elite	-

A new hotel report has been created, streamlining data management and providing valuable insights for decision-making.

New Flight Report:

The screenshot shows a flight report titled "Report: Flights New Flights Report". It displays 4 total records. The data is organized into columns: Name (Flight: Flight Name), Date, and Flight Number. The flights listed are FL- 0005 (26/11/2024), FL- 0004 (26/11/2024), FL- 0006 (28/11/2024), and FL- 0007 (29/11/2024). Subtotal rows are present for each date group.

Name	Flight: Flight Name	Date	Flight Number
26/11/2024 (2)	FL- 0005		
	FL- 0004		
Subtotal			
28/11/2024 (1)	FL- 0006		
Subtotal			
29/11/2024 (1)	FL- 0007		
Subtotal			
Total (4)			

Row Counts: Detail Rows: Subtotals: Grand Total:

A new flight report has been created, providing updated and relevant data for better analysis and decision-making.

New Customers Report:

The screenshot shows a customer report titled "Report: Customers New Customers Report". It displays 5 total records. The data is organized into columns: Customer Name, Discount Amount, Total Food Options, and Discount Percent. The customers listed are Jerin, Divya, Hema, Abinaya, and Leela. All customers received a 30% discount on 5 food options.

Customer: Customer Name	Discount Amount	TotalFoodOptions	Discount Percent	Date
1 Jerin	₹0.00	-	-	-
2 Divya	₹0.30	5	30%	26/11/2024
3 Hema	₹0.00	-	-	-
4 Abinaya	₹0.00	-	-	-
5 Leela	₹0.00	-	-	-
6	₹0.30	5	30%	

A new customer report has been created to manage and analyze customer data efficiently.

New Food Options Report:

The screenshot shows a report titled "Report: Food Options New Food Options Report". The report displays a table of food options categorized by hotel. The table includes columns for Hotel, Food Option Name, Name, and Food Amount. The total number of records is 7, and the total food amount is ₹1,070.

Hotel	Food Option Name	Name	Food Amount
Holiday Dream Resorts (1)	FO - 0003	Beverage	₹70
Subtotal			₹70
Hotel Le Grand (4)	FO - 0006	Dessert	₹50
	FO - 0004	Non-veg meal	₹350
	FO - 0002	Snack and fries	₹50
	FO - 0001	Dinner	₹250
Subtotal			₹700
Starry Sky Cabins (2)	FO - 0008	Simple Breakfast	₹100
	FO - 0005	Veg meal	₹200
Subtotal			₹300
Total (7)			₹1,070

Row Counts: Detail Rows: Subtotals: Grand Total:

A new FoodOption report has been created to streamline the management and analysis of food-related data.

Activity 2:

1. Create a report with report type: “TripAdiser E-Management and Projects”.

The screenshot shows the "Reports" section of the application. It displays a list of recent reports, categorized into "Recent", "Created by Me", "Private Reports", "Public Reports", "All Reports", "Folders", and "Favorites". The "Recent" section lists four reports: "New Food Options Report", "New Customers Report", "New Flights Report", and "New Hotels Report". The "Sample Flow Report: Screen Flows" report is also listed under "Recent".

Report Name	Description	Folder	Created By	Created On	Subscribed
New Food Options Report		Trip	Jerin Richie D	27/11/2024, 11:15 pm	
New Customers Report		Trip	Jerin Richie D	27/11/2024, 11:19 pm	
New Flights Report		Trip	Jerin Richie D	27/11/2024, 11:17 pm	
New Hotels Report		Trip	Jerin Richie D	27/11/2024, 11:18 pm	
Sample Flow Report: Screen Flows	Which flows run, what's the status of each interview, and how long do users take to complete the screens?	Public Reports	Automated Process	18/10/2024, 4:23 pm	

Milestone 12 - Dashboards:

Dashboards provide a visual summary of real-time data, enabling users to quickly understand business trends, monitor performance, and make informed decisions. They allow easy access to report data through visual components.

UseCase:

As an Admin for TripAdvisor E-Management, you continually strive to meet business requirements, driving the organization toward peak performance. Your dedication and effective data visualization in reports have greatly impressed your superiors, making it effortless for the CEO to access and view essential data during meetings without having to search.

Activity 1:

Create Dashboard

1. Go to the app → click on the Dashboards tabs.

New Dashboard

* Name

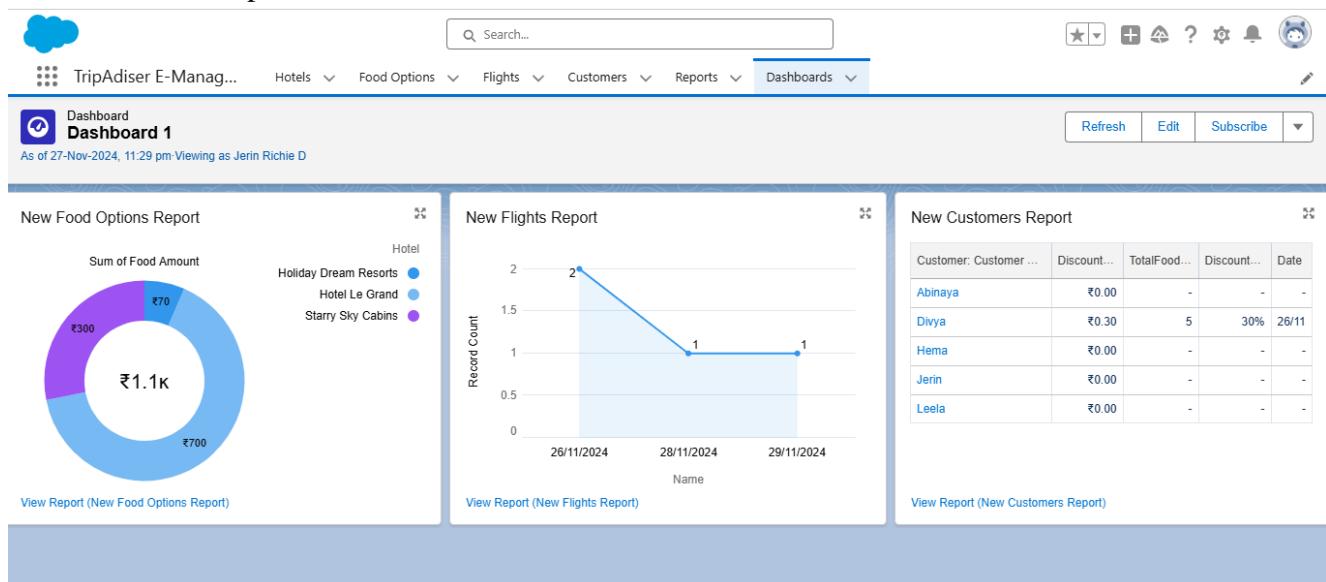
Description

Folder

2. Give a Name and click on Create.

3. Select add component.

4. Select a Report and click on select.



Key Scenarios Addressed by Salesforce in the Implementation Project

1. Automating Hotel Data Updates:

Triggers handle changes to food options without manual intervention.

2. Customer Discount Management:

Flows automate discount calculation and application, enhancing customer satisfaction.

3. Flight Reminder Notifications:

Schedulable Apex ensures timely communication, reducing customer queries.

Conclusion

Summary of Achievements

The *TripAdvisor E-Management* Salesforce project successfully streamlined the management of travel-related services by implementing a comprehensive solution that integrates automation, custom user interfaces, and real-time notifications. Key achievements include:

- Developed a user-friendly CRM tailored to manage hotel, flight, and food option data efficiently.
- Automated discount calculations and flight reminder notifications, significantly reducing manual processes and improving customer satisfaction.
- Provided actionable insights into hotel occupancy, food option availability, and flight booking trends, supporting better business decisions.

This solution is scalable and adaptable, providing a robust foundation for future enhancements, such as advanced customer personalization or integration with third-party travel platforms.