

Project title: Salesforce Developer

Name: Shalini.N

Sureha.K

Priyadharshini.S

Sandhiya.S

Registered Email id: s6777019@gmail.com

TripAdvisor E-Management

1. PROJECT OVERVIEW :

A Tripadvisor e-management project aims to streamline online operations for hospitality and tourism businesses by enhancing their visibility, reviews, and engagement on Tripadvisor. The project typically focuses on optimizing listings, responding to customer reviews, and utilizing data analytics to monitor customer feedback and competitor performance. E-management solutions can include training staff to respond effectively, leveraging Tripadvisor's promotional tools, and ensuring content accuracy. By actively managing Tripadvisor profiles, businesses can improve their reputation, increase customer satisfaction, and attract more visitors, ultimately driving revenue growth. The project aligns business strategies with Tripadvisor's platform capabilities to create a stronger digital presence.

2. OBJECTIVES

- ✓ **Goal:** Enhance online reputation
- ✓ **Outcome:** Higher review ratings and improved brand perception

- ✓ **Goal:** Optimize Tripadvisor listings
- ✓ **Outcome:** Accurate, engaging content that attracts more views and bookings

- ✓ **Goal:** Actively respond to reviews
- ✓ **Outcome:** Improved customer satisfaction and trust in brand responsiveness

- ✓ **Goal:** Boost visibility and ranking
- ✓ **Outcome:** Increased profile visits and bookings through better placement

- ✓ **Goal:** Use data analytics for insights
- ✓ **Outcome:** Identified trends and actionable insights for service improvements

3. SALESFORCE KEY FEATURES AND CONCEPTS UTILIZED

The main functionalities and concepts applied within the Salesforce project:

- **Service Cloud:** Manages and responds to Tripadvisor reviews efficiently.
- **Marketing Cloud:** Personalizes promotional campaigns to boost visibility.
- **Customer 360:** Provides a unified view of each customer for tailored interactions.
- **Analytics & AI Insights:** Monitors sentiment and identifies trends in reviews and feedback.
- **Workflow Automation:** Streamlines listing updates and repetitive tasks.
- **Collaboration Tools:** Enhances internal communication and team efficiency.

These Salesforce tools help businesses optimize their Tripadvisor presence, engage customers more effectively, and drive improved satisfaction and revenue.

4. DETAILED STEPS TO SOLUTION DESIGN

Acceptance Criteria & Solution

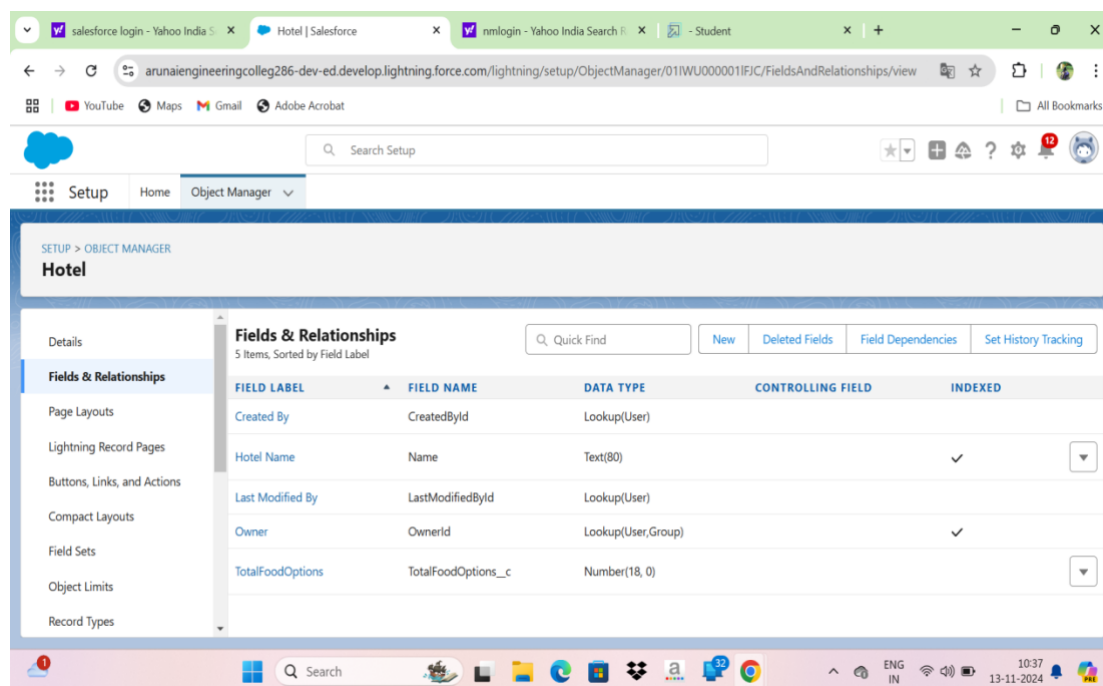
- As the Salesforce User we have to manage the data for the Hotels, Flights, and Food Options for this we have to create some automation for simplification.
- To ensure that when a new Food Option is added or updated, the corresponding Hotel's information is updated accordingly. For example, you might want to maintain a total count of food options for each hotel.
- Also there is automation for the customer benefits if the there buying amount is with respect to some amount then they will get some discounts on their bill
- For the flights there schedule process being involved where the customer who has booked the flight will get the reminder mail alert for knowing proper timing of the flight before 24 hrs it's important to manage the in a good way.
- The system should provide confirmation or notification to the user upon successful sending of the email.

Solution: For the Above requirements of TripAdvisor we have created the solutions by creating the custom objects and Fields the Custom Objects that are created are Hotels, Food Options, Customer & Flights. For the Automation we have used here a flow and triggers and for scheduling the email alerts we have created the Apex Schedulable class so email alerts will be created.

Activity 1 : Create Object

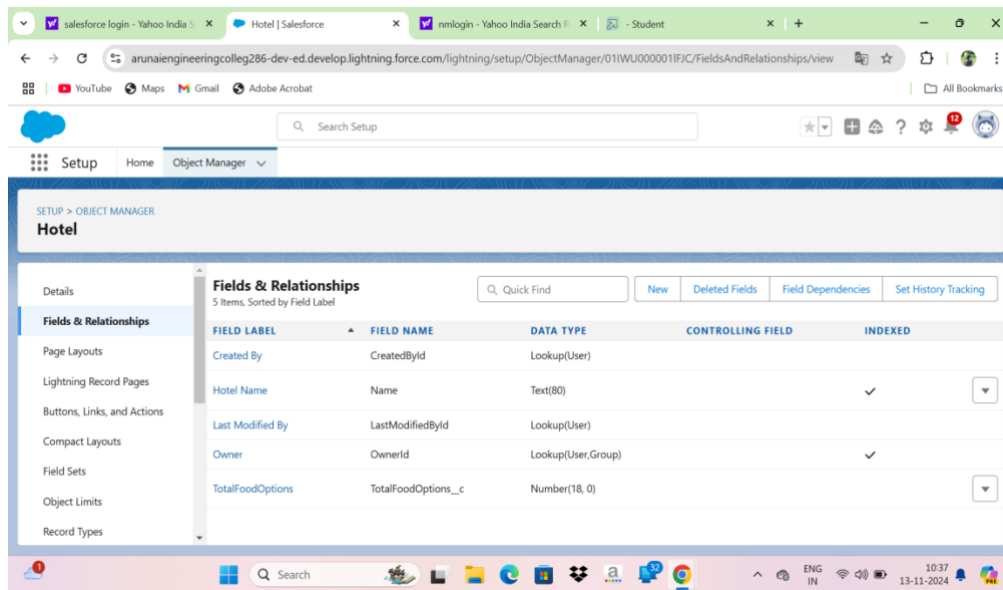
→ Hotel Object is created to ensure that when a new Food Option is added or updated with the necessary information

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



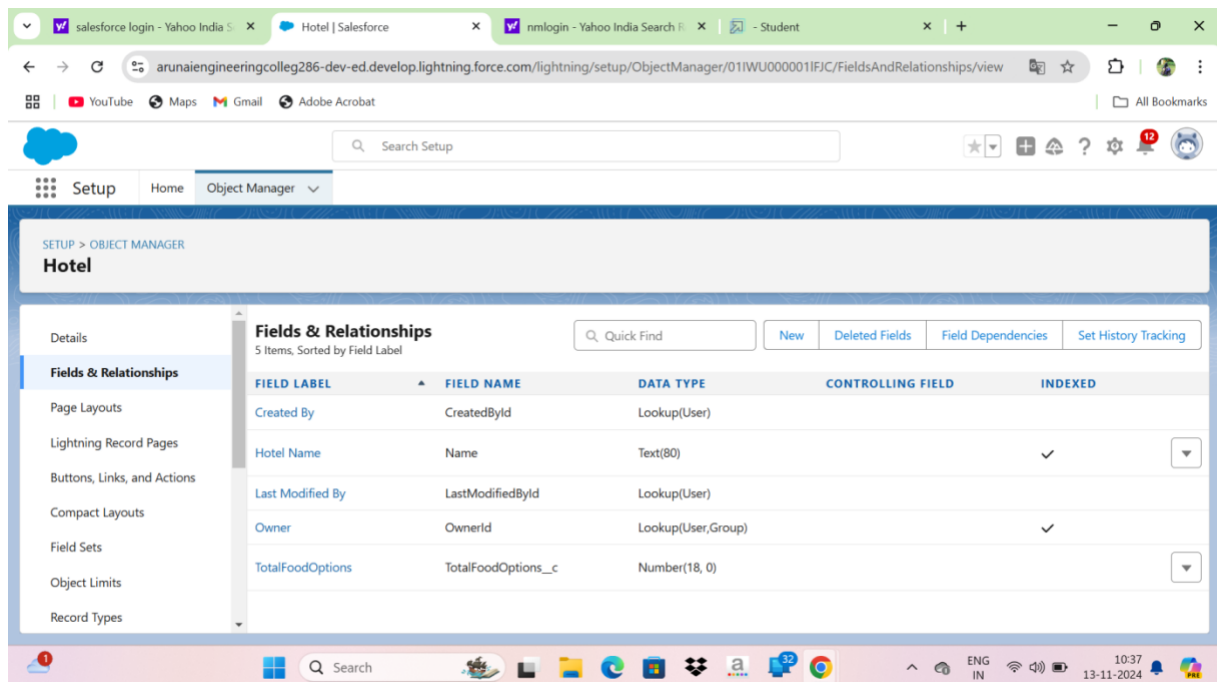
Activity-2 : Create Fields for Hotel Object

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



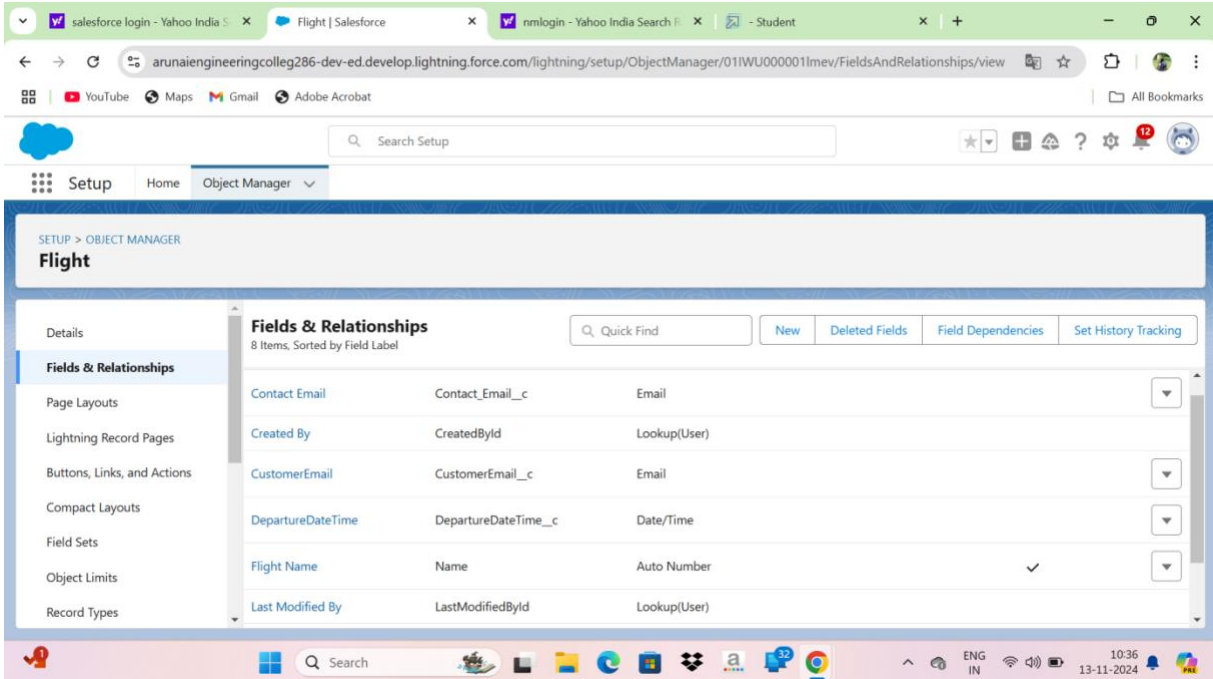
Activity-3: Create Fields For Food Option

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



Activity-3: Create Fields in the Flight Object

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

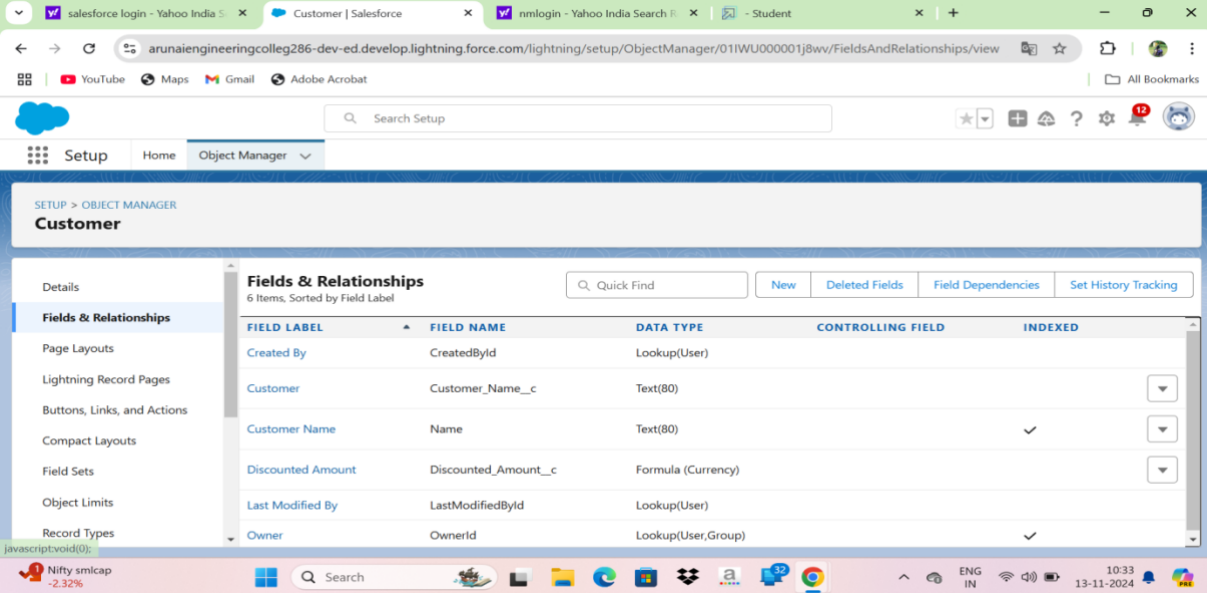


The screenshot shows the Salesforce Setup interface for the 'Flight' object. The 'Fields & Relationships' section is active, displaying a list of 8 fields sorted by label. The fields are:

Field Label	Field Name	Data Type	Field Type
Contact Email	Contact_Email_c	Email	Text
Created By	CreatedById	Lookup(User)	Text
CustomerEmail	CustomerEmail_c	Email	Text
DepartureDateTime	DepartureDateTime_c	Date/Time	Text
Flight Name	Name	Auto Number	Text
Last Modified By	LastModifiedById	Lookup(User)	Text

Activity-4: Create Fields in the Customer Object

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 'Fields & Relationships' section is active, displaying a list of 6 fields sorted by Field Label. The fields are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer	Customer_Name__c	Text(80)		
Customer Name	Name	Text(80)		✓
Discounted Amount	Discounted_Amount__c	Formula (Currency)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓

Create Flow

- Create the Flow for the discount for customer when the Amount is greater than 3000 some some Amount of Discounts will be there if the Amount is between 1500 to 3000 so Some Amount of Discount will be there for them

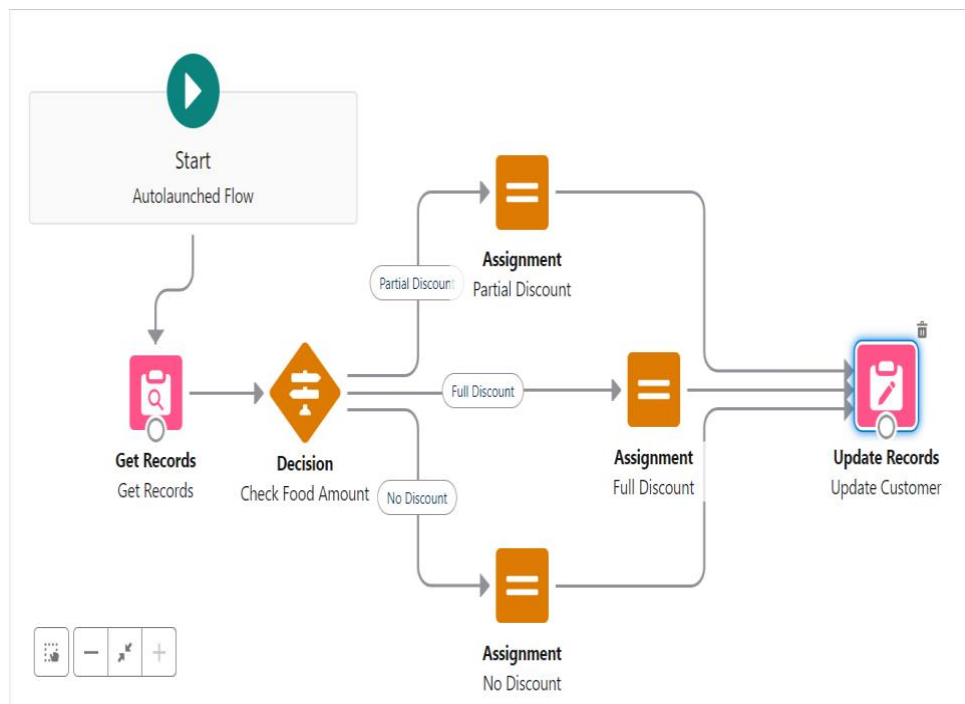
Activity-1: Flow Procedure

Create 3 variable :

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

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

Variable > Api name > discount > Number



Apex Triggers

- ➔ Scenario: In the Hotel you have to ensure that when a new Food Option is added or updated, the corresponding Hotel's information is updated accordingly. For example, you might want to maintain a total count of food options for each hotel. To manage the things properly with perspective to the Hotel things should be clearly manageable for making the food options available with respect to hotels

Activity-1: Apex trigger With Handler

Apex Trigger With Handler

```

public class FoodOptionTriggerHandler {
// Method to update hotel information based on food options
public static void updateHotelInformation(List<Food_Option__c>
newFoodOptions,    List<Food_Option__c> oldFoodOptions,
TriggerOperation operation) {
Set<Id> hotelIdsToUpdate = new Set<Id>();

// Collect unique Hotel Ids affected by food options changes
for (Food_Option__c foodOption : newFoodOptions) {
hotelIdsToUpdate.add(foodOption.Hotel__c);
}
}
}
  
```



```
}
```

```
// Update hotel information based on food options
```

```
List<Hotel__c> hotelsToUpdate = [SELECT Id, Name, TotalFoodOptions__c  
FROM Hotel__c WHERE Id IN :hotelIdsToUpdate];
```

```
for (Hotel__c hotel : hotelsToUpdate) {
```

```
// Recalculate total food options count
```

```
Integer totalFoodOptions = [SELECT COUNT() FROM Food_Option__c  
WHERE Hotel__c = :hotel.Id];
```

```
hotel.TotalFoodOptions__c = totalFoodOptions;
```

```
}
```

```
// Update hotels with new total food options count
```

```
update hotelsToUpdate;
```

```
}
```

```
}
```


```
1 public class FoodOptionTriggerHandler {  
2     // Method to update hotel information based on food options  
3     public static void updateHotelInformation(List<Food_Option__c> newFoodOptions, List<Food_Option__c> oldFoodOptions, TriggerOperation operation) {  
4         Set<Id> hotelIdsToUpdate = new Set<Id>();  
5  
6         // Collect unique Hotel Ids affected by food options changes  
7         for (Food_Option__c foodOption : newFoodOptions) {  
8             hotelIdsToUpdate.add(foodOption.Hotel__c);  
9         }  
10  
11         // Update hotel information based on food options  
12         List<Hotel__c> hotelsToUpdate = [SELECT Id, Name, TotalFoodOptions__c FROM Hotel__c WHERE Id IN :hotelIdsToUpdate];  
13  
14         for (Hotel__c hotel : hotelsToUpdate) {  
15             // Recalculate total food options count  
16             Integer totalFoodOptions = [SELECT COUNT() FROM Food_Option__c WHERE Hotel__c = :hotel.Id];  
17             hotel.TotalFoodOptions__c = totalFoodOptions;  
18         }  
19  
20         // Update hotels with new total food options count  
21         update hotelsToUpdate;  
22     }  
23  
24  
25 }
```

Trigger

```
trigger FoodOptionTrigger on Food_Option__c (after insert, after update, after delete) {
    If(trigger.isInsert && trigger.isAfter){
        FoodOptionTriggerHandler.updateHotelInformation(trigger.new);
    }
}
```

Edit Get Records

Find Salesforce records and store their field values in flow variables.

Get Food options (Get_Food_options) 

Get Records of This Object

* Object

Food Option

Filter Food Option Records


Condition Requirements

All Conditions Are Met (AND) ▼

Cancel
Done

2. Decision Element: Create 2 Outcomes

Edit Decision

OUTCOME ORDER 

- Full Discount
- Partial Discount
- 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


 Food Option from Get_Food_options > Food... X

Operator

Greater Than ▼

Value

3000




+ Add Condition

Cancel
Done


Take the 3 Assignments > Full Discount, Partial Discount & No Discount

Edit Assignment

Full Discount (Full_Discounts) 

Set Variable Values

Each variable is modified by the operator and value combination.


Variable	Operator	Value	
<input type="text" value="# discount x"/>	<input type="text" value="Equals"/>	<input type="text" value="20"/>	

+ Add Assignment

Cancel


Done

Edit Assignment

Partial Discount (Partial_Discounts_0) 

Set Variable Values

Each variable is modified by the operator and value combination.

Variable	Operator	Value	
<input type="text" value="# discount x"/>	<input type="text" value="Equals"/>	<input type="text" value="10"/>	


+ Add Assignment

Cancel

Done


Edit Assignment

Edit Assignment

No Discount (No_Discount) 

Set Variable Values

Each variable is modified by the operator and value combination.

Variable	Operator	Value	
<input type="text" value="# discount x"/>	<input type="text" value="Equals"/>	<input type="text" value="0"/>	

+ Add Assignment

Cancel

Done

Update Record Element

Edit Update Records

*** How to Find Records to Update and Set Their Values**

☐ Use the IDs and all field values from a record or record collection
☒ Specify conditions to identify records, and set fields individually

Update Records of This Object Type

* Object

Customer

Filter Customer Records

Condition Requirements to Update Records

All Conditions Are Met (AND)

Cancel Done

Edit Update Records

All Conditions Are Met (AND)

Field	Operator	Value
Id	Equals	Aa csld

+ Add Condition

Set Field Values for the Customer Records

Field	Value
Discount_Percent_c	# discount

+ Add Field

Cancel Done

Apex Triggers :

Scenario: In the Hotel you have to ensure that when a new Food Option is added or updated, the corresponding Hotel's information is updated accordingly. For example, you might want to maintain a total count of food options for each hotel. To

manage the things properly with perspective to the Hotel things should be clearly manageable for making the food options available with respect to hotels

Activity-1: Apex trigger With Handler

Apex Trigger With Handler

```
public class FoodOptionTriggerHandler {  
    // Method to update hotel information based on food options  
    public static void updateHotelInformation(List<Food_Option__c> newFoodOptions,  
    List<Food_Option__c> oldFoodOptions, TriggerOperation operation) {  
        Set<Id> hotelIdsToUpdate = new Set<Id>();  
  
        // Collect unique Hotel Ids affected by food options changes  
        for (Food_Option__c foodOption : newFoodOptions) {  
            hotelIdsToUpdate.add(foodOption.Hotel__c);  
        }  
  
        // Update hotel information based on food options  
        List<Hotel__c> hotelsToUpdate = [SELECT Id, Name, TotalFoodOptions__c FROM Hotel__c  
        WHERE Id IN :hotelIdsToUpdate];  
  
        for (Hotel__c hotel : hotelsToUpdate) {  
            // Recalculate total food options count  
            Integer totalFoodOptions = [SELECT COUNT() FROM Food_Option__c WHERE Hotel__c =  
            :hotel.Id];  
            hotel.TotalFoodOptions__c = totalFoodOptions;  
        }  
  
        // Update hotels with new total food options count  
        update hotelsToUpdate;  
    }  
}
```

```

1 public class FoodOptionTriggerHandler {
2     // Method to update hotel information based on food options
3     public static void updateHotelInformation(List<Food_Option__c> newFoodOptions, List<Food_Option__c> oldFoodOptions, TriggerOperation operation) {
4         Set<Id> hotelIdsToUpdate = new Set<Id>();
5
6         // Collect unique Hotel Ids affected by food options changes
7         for (Food_Option__c foodOption : newFoodOptions) {
8             hotelIdsToUpdate.add(foodOption.Hotel__c);
9         }
10
11         // Update hotel information based on food options
12         List<Hotel__c> hotelsToUpdate = [SELECT Id, Name, TotalFoodOptions__c FROM Hotel__c WHERE Id IN :hotelIdsToUpdate];
13
14         for (Hotel__c hotel : hotelsToUpdate) {
15             // Recalculate total food options count
16             Integer totalFoodOptions = [SELECT COUNT() FROM Food_Option__c WHERE Hotel__c = :hotel.Id];
17             hotel.TotalFoodOptions__c = totalFoodOptions;
18         }
19
20         // Update hotels with new total food options count
21         update hotelsToUpdate;
22     }
23 }
24
25 }

```

Trigger

```

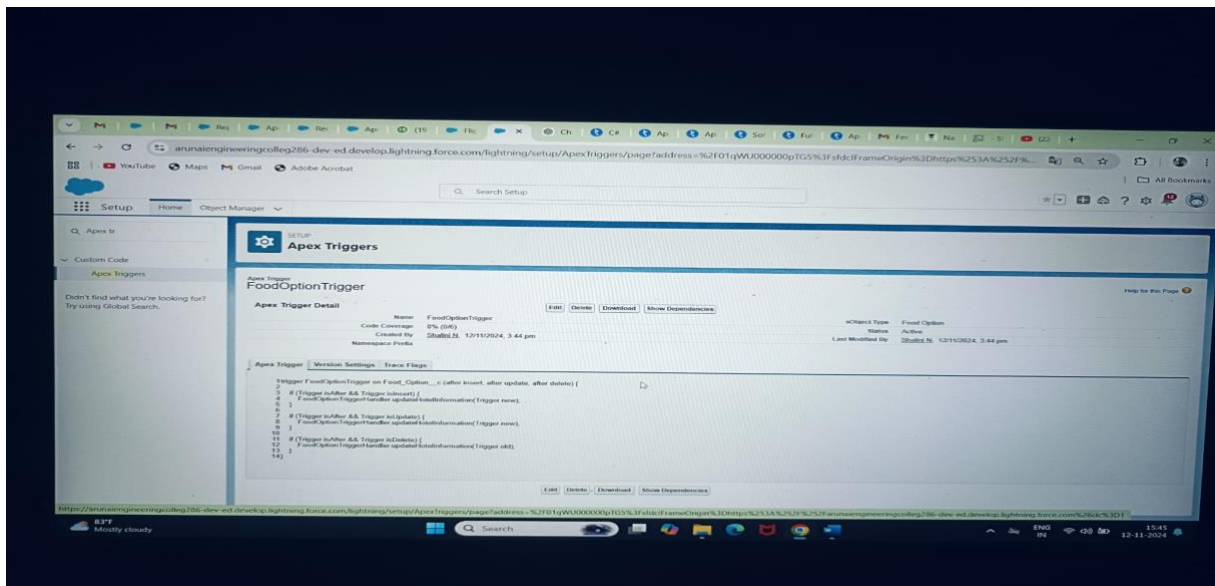
trigger FoodOptionTrigger on Food_Option__c (after insert, after update, after delete) {
    If(trigger.isInsert && trigger.isAfter){
        FoodOptionTriggerHandler.updateHotelInformation(trigger.new);
    }
}

```

```

1 trigger FoodOptionTrigger on Food_Option__c (after insert, after update, after delete) {
2     If(trigger.isInsert && trigger.isAfter){
3         FoodOptionTriggerHandler.updateHotelInformation(trigger.new);
4     }
5 }
6

```



Apex Schedule:

- ➔ Create the Reminder mail for the customer who has booked the flight according to that booking set the Apex schedule so mail will be sent prior to 24hrs.

Note: Please create the required field for Scheduled Apex Code

Activity-1: Apex Schedule Class Solution

```
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,
        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 });
    }
}
}
```



```

FlightReminderScheduledJob.apex
Code Coverage: None | API Version: 59
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 FlightReminderScheduledJob class implements the Schedulable interface, and the execute method is where you put the logic to send reminder emails.

The sendFlightReminders method queries for flights departing within the next 24 hours. You can customize the query based on your specific requirements.

Create the Apex code in an anonymous Window to execute the Apex Code

// 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());

```

Enter Apex Code
1 // Schedule the job to run every day at a specific time (e.g., 6 AM)
2 String cronExp = '0 0 6 * * ?';
3 System.schedule('FlightReminderJob', cronExp, new FlightReminderScheduledJob());
4
[Open Log] [Execute] [Execute Highlighted]

```

Conclusion :

In conclusion, TripAdvisor's e-management practices demonstrate a strong alignment with user-centric digital strategies that effectively build engagement and trust. The platform's emphasis on transparency through detailed user reviews and ratings has fostered a community-driven approach, establishing TripAdvisor as a reliable source for travel planning. Its use of advanced technologies, like AI and real-time analytics, allows for personalized recommendations and enhanced user experiences, setting it apart in the travel industry.

However, challenges such as addressing fake reviews and further improving mobile usability highlight areas for refinement to boost reliability and ease of use. Moving forward, TripAdvisor's focus on evolving technologies and sustainable travel can help it stay competitive and meet shifting traveler needs. Overall, its e-management success exemplifies how digital strategies, continuous innovation, and customer trust can create significant value, securing TripAdvisor's position as a market leader in travel services.