# Trip Advisor E-Management

College Name: Angel College of Engineering and Technology

College code : 7103

Department : Computer Science and Engineering

### NM Project : Trip Advisor E-Management

**Group project**

**TEAM ID: NM2024TMID03492**

Kirubakaran T : NMID 3738A0BAA747F33B41541A6DA2D8D329

Logesh .S: NMID- EF98DD76BD70FC292764318FB4DF1AF9

Karthi .S NMID- 0EF875DC495F675823D45097DA0C78B8

Prabakaran .C: NMID- 833964EF1116B85BB98476D1B176AA4B

Overview The TripAdvisor E-Management App is a Salesforce-based travel management solution designed to streamline trip planning and booking. It allows users to manage hotels, food options, flights, and customer data efficiently while automating key processes such as flight reminders and real-time updates to hotel information. This app enhances operational efficiency, improves customer satisfaction, and ensures accurate data management for travel agencies.

Features Dynamic Food Option Management: Automatically updates the total count of food options linked to hotels. Flight Reminder Notifications: Sends automated reminders for upcoming flights to customers. Customer Discounts: Calculates and applies personalized discounts dynamically. Custom Reports and Dashboards: Provides insights into booking trends and customer preferences.

Key Objects Hotel: Manages hotel data and tracks the total food options. Food Option: Links food items to specific hotels. Flight: Tracks flight schedules and sends automated notifications. Customer: Stores customer details, including discount information. Installation Instructions

Deploy Custom Objects:

Create objects: Hotel, Food Option, Flight, and Customer. Define fields as per the design specifications. Set Up Apex Classes and Triggers:

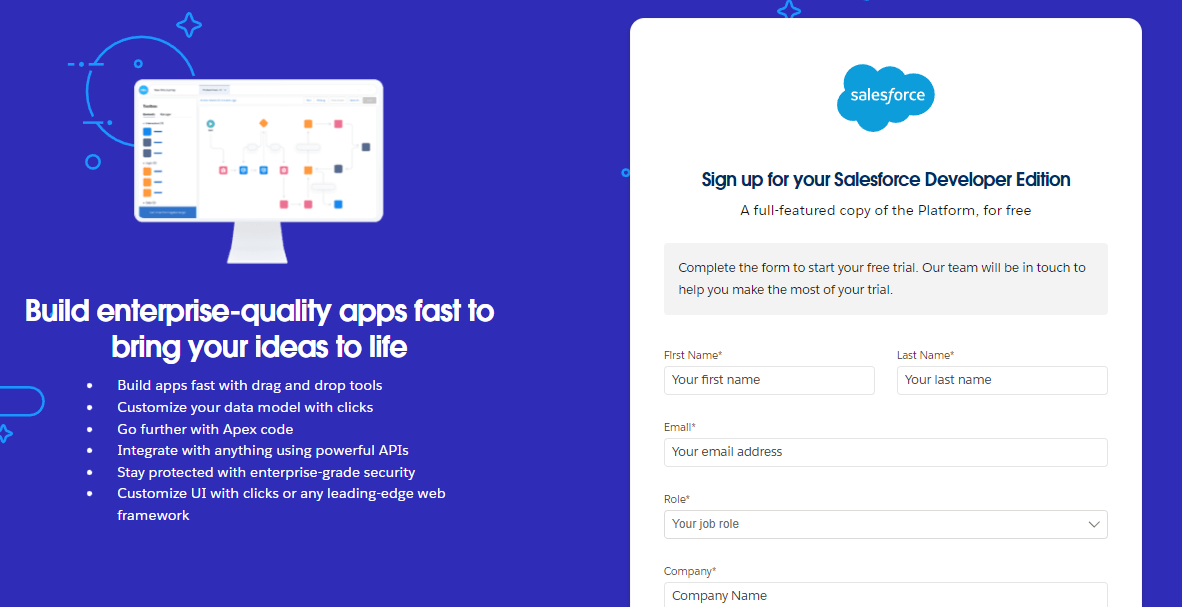
Deploy FoodOptionTriggerHandler and FlightReminderScheduledJob classes. Add FoodOptionTrigger for dynamic hotel updates. Schedule Jobs:

Use the Salesforce Developer Console to schedule FlightReminderScheduledJob. Testing:

Run test classes for FoodOptionTriggerHandler and FlightReminderScheduledJob to ensure over 90% code coverage. Usage Adding/Updating Food Options: Automatically updates the TotalFoodOptions field for the linked hotel. Flight Notifications: Sends reminders for flights departing within 24 hours. Future Enhancements Integration with third-party booking systems for real-time updates. AI-based recommendations for hotels, food, and flights.

**SALESFORCE**

**Creating developer account**

****

1. First name & Last name
2. Email
3. Role: Developer
4. Company: College Name
5. County: India
6. Postal Code: pin code
7. Username: should be a combination of your name and company

This need not be an actual email id you can give anything in the format: lusername@organization.com

Click on sign me up after filling these.

**Account Activation**

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.

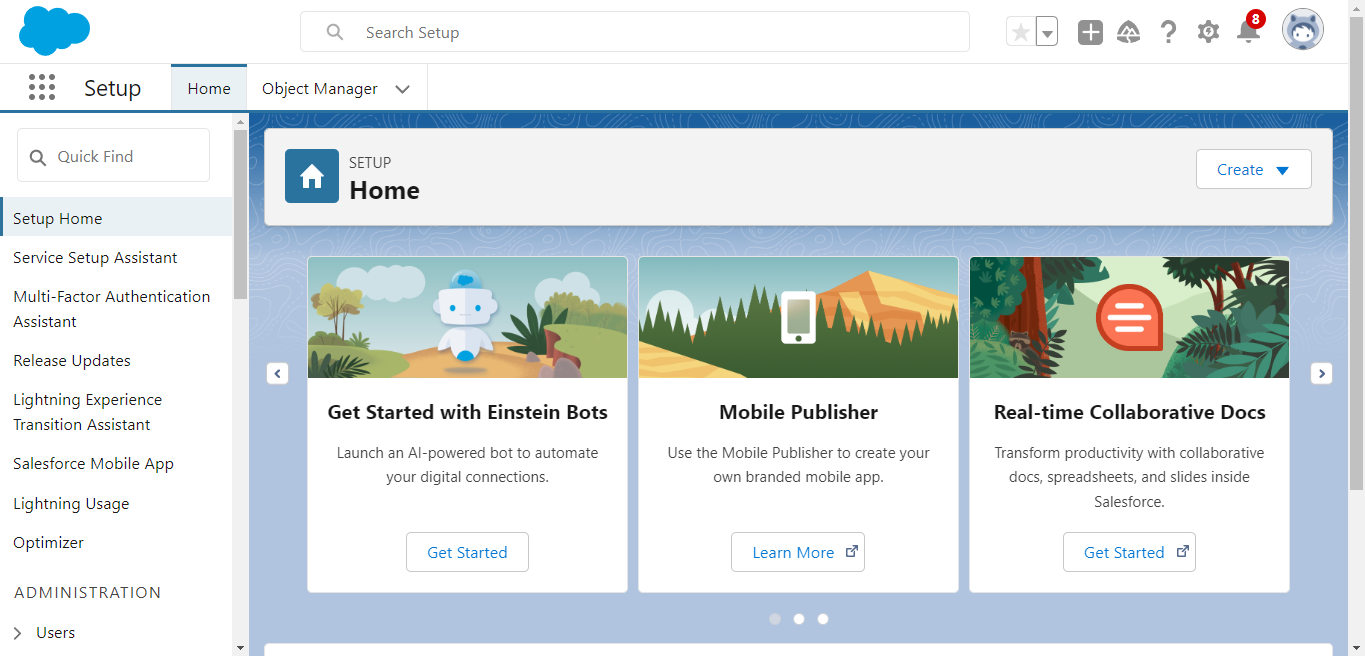
A screenshot of a computer

Description automatically generated

1. Click on Verify Account
2. Give a password and answer a security question and click on change password.
3. Then you will redirect to your salesforce setup page.

This need not be an actual email id you can give anything in the format: username@organization.com

Click on sign me up after filling these.



**Create Object**

From the setup page **-->** Click on Object Manager **-->** Click on Create **-->** Click on Custom Object.

**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

1. Plural Name : Hotels

1. Data Type : (text)

1. Field Name : Hotel Name

1. Click Allow Reports

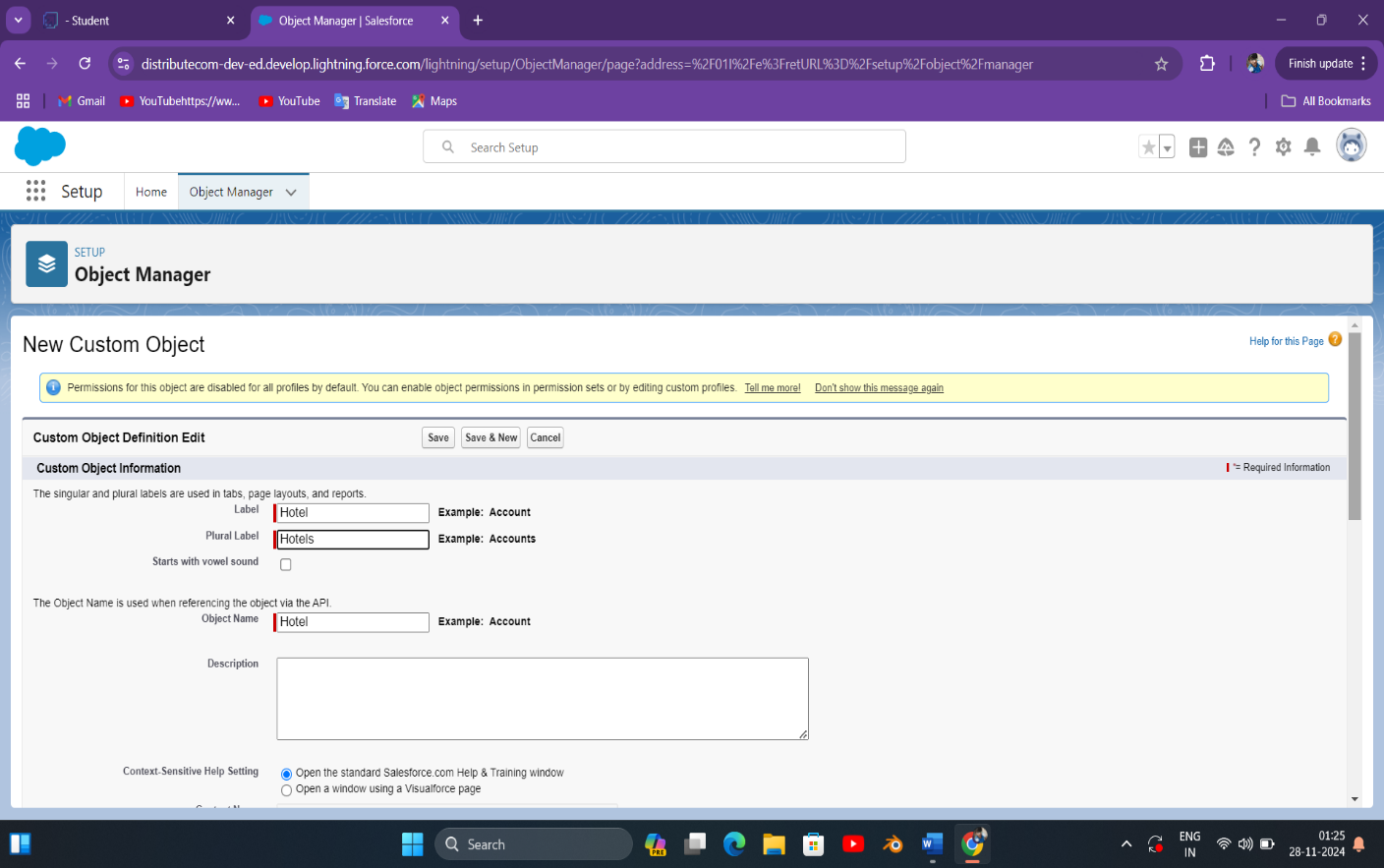
1. Allow Search ? Save

With Above References Create following Object

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

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

Customer ? Text ? Field Name ? Customer Name



# **Create Fields for Hotel Object**

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

# 

### Create Fields For Food Option

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

1. Repeat step 1 and 2 mentioned in activity 1
2. Select Data type as “Lookup Relationship” and click Next.
3. Select Employee from the drop down related to the field and click Next.
4. Give Field Label as “Reports to” and click Next.

### Next --> Next --> Save & New

### 

### Create Fields in the Flight Object

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

### 

### 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 |

### 

### 

### Create Fields for Hotel Object

### Salesforce Flow for Discount Calculation Based on Amount

### Objective:

### Create a Salesforce Flow to apply different discount rates based on the Amount field in a Sales Order or Invoice object.

### Steps to Create the Flow:

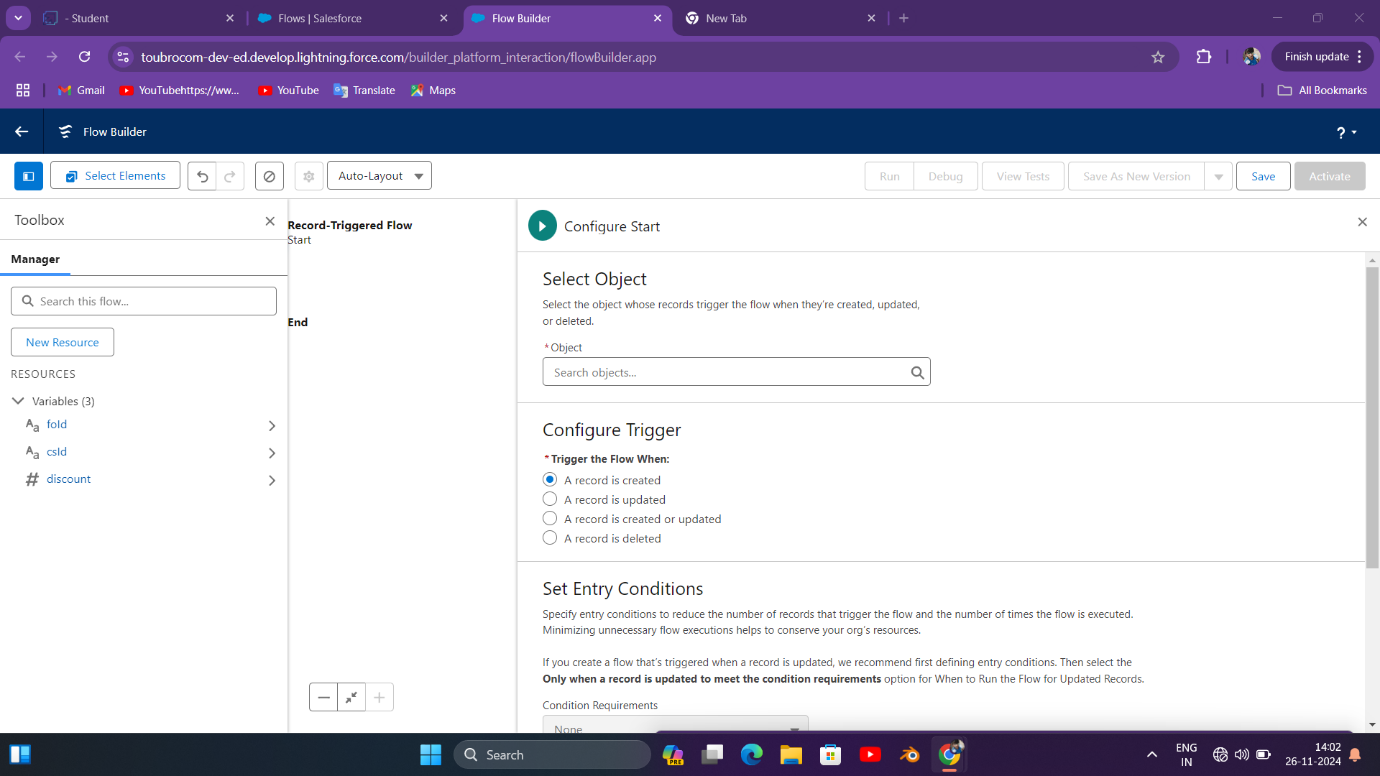
**1. Define Discount Criteria**

* **If the amount is greater than 3000:** Apply a higher discount (e.g., 20% off).
* **If the amount is between 1500 and 3000:** Apply a moderate discount (e.g., 10% off).
* **If the amount is below 1500:** No discount.

**2. Steps to Create the Flow in Salesforce**

**Step 1: Access Flows**

1. Log in to Salesforce.
2. Go to **Setup** → Search for **Flows** in the Quick Find box → Click **Flows**.
3. Click on the **New Flow** button.



**Step 2: Choose Flow Type**

* Select **Record-Triggered Flow** if you want the discount calculation to run when a record (e.g., Opportunity or Order) is created or updated.
* Alternatively, choose **Screen Flow** if users will manually input the purchase amount to calculate the discount.

**Step 3: Add the Logic**

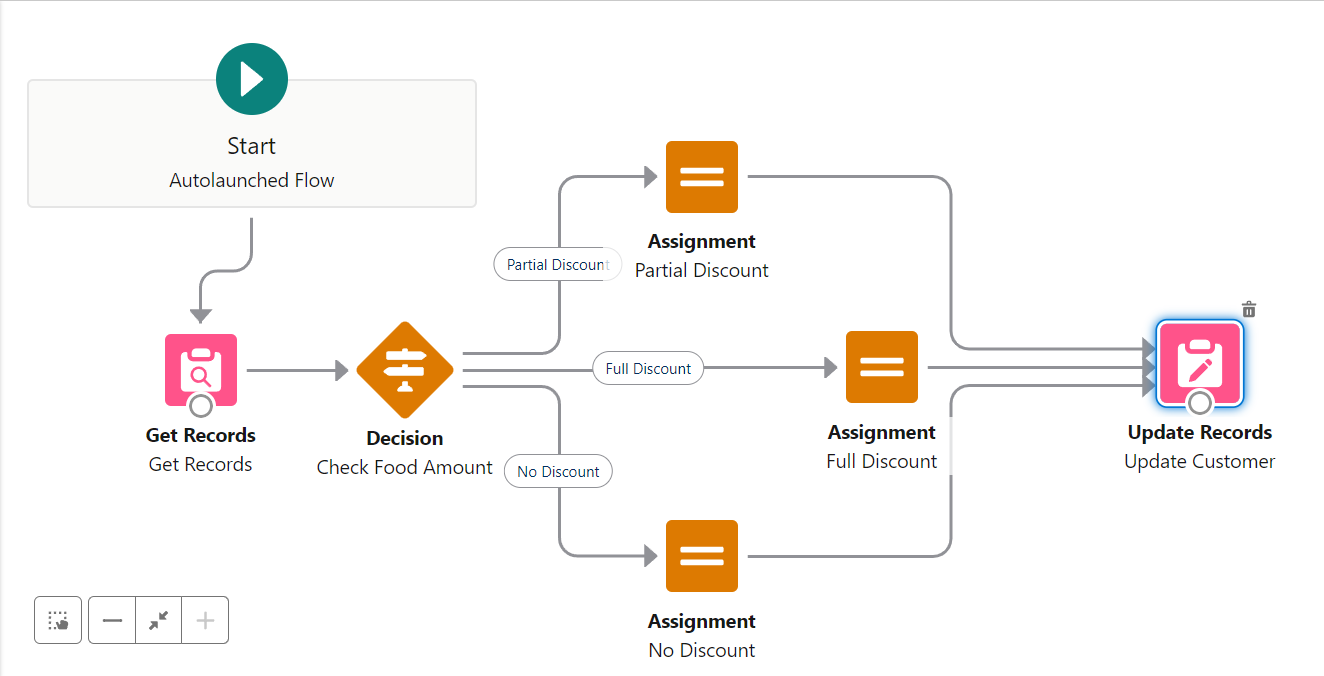
1. **Add a Decision Element:**
   * Drag a **Decision** component onto the canvas to check the purchase amount.
   * Configure three outcomes:
     + **High Discount (Amount > 3000):**
       - Condition: Amount > 3000.
     + **Medium Discount (1500 ≤ Amount ≤ 3000):**
       - Condition: Amount >= 1500 AND Amount <= 3000.
     + **No Discount (Amount < 1500):**
       - Default outcome or set condition: Amount < 1500.
2. **Assign Discount Values:**
   * Drag **Assignment** components for each outcome to calculate and store the discount value:
     + **For High Discount:**
       - Create a variable Discount and assign: Discount = Amount \* 0.20.
     + **For Medium Discount:**
       - Assign: Discount = Amount \* 0.10.
     + **For No Discount:**
       - Assign: Discount = 0.
3. **Update or Display Results:**
   * If using **Record-Triggered Flow**, update the record (e.g., Opportunity) with the calculated discount.
   * If using **Screen Flow**, add a screen element to display the calculated discount to the user.

**Step 4: Test the Flow**

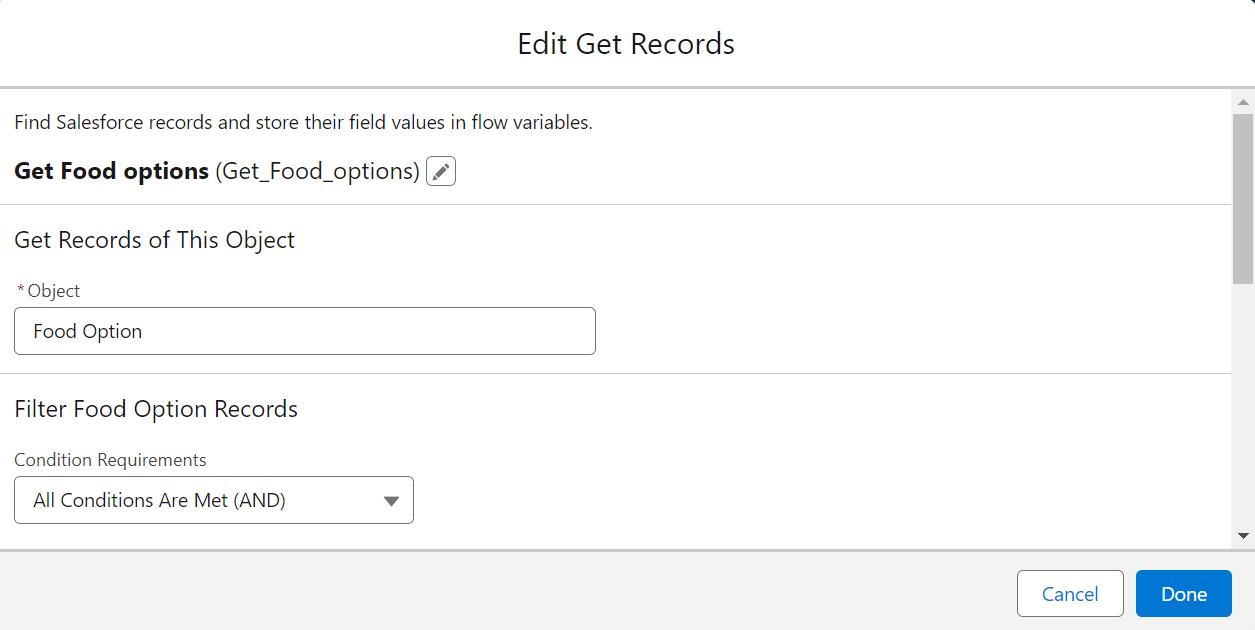
1. Save and activate the flow.
2. Test the flow by creating or updating records (if using a Record-Triggered Flow) or running the Screen Flow with different amounts.

**Example Flow Diagram**

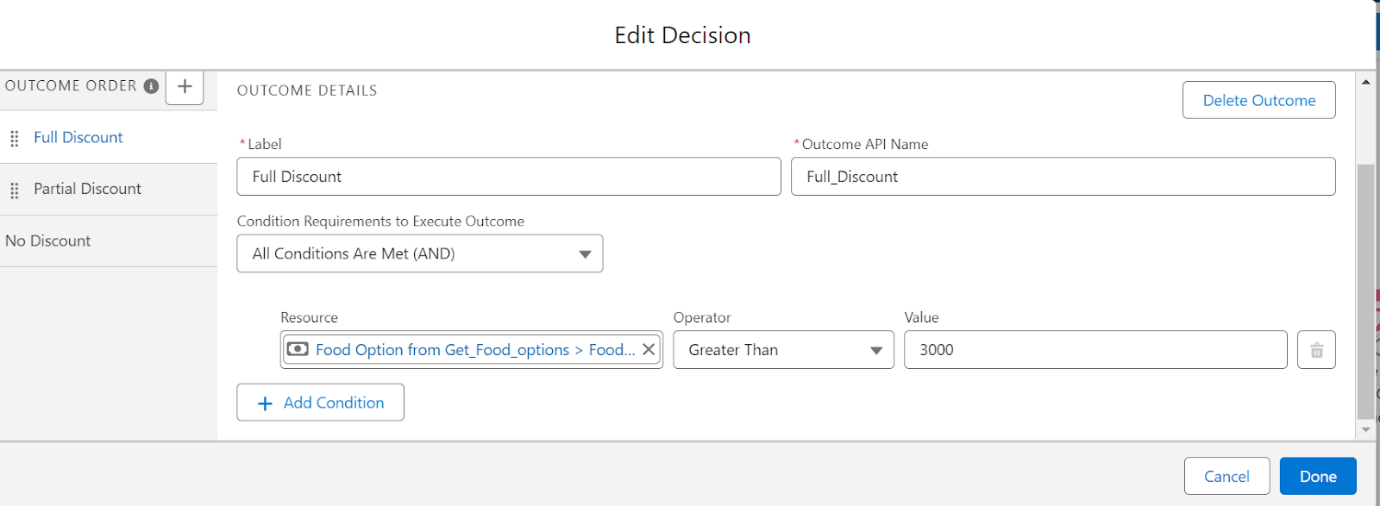
1. **Start** (Triggered by Record Creation or Update)
2. **Decision** (Amount > 3000, 1500 ≤ Amount ≤ 3000, Amount < 1500)
3. **Assignment** (Calculate Discount)
4. **Update Record** or **Display Discount**
5. **End**

****

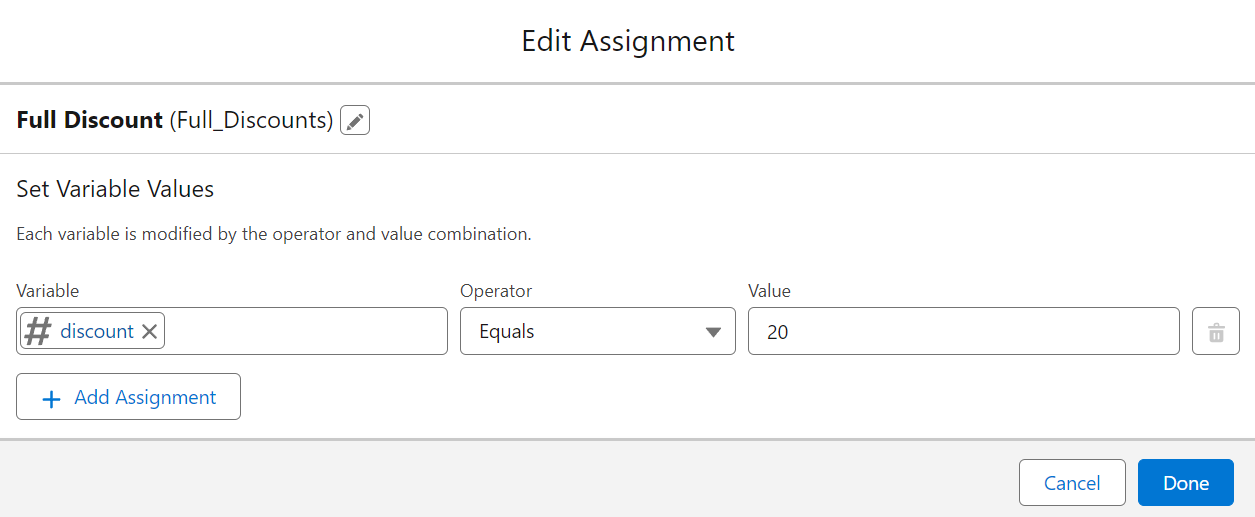
**Flow Steps :** Get Records

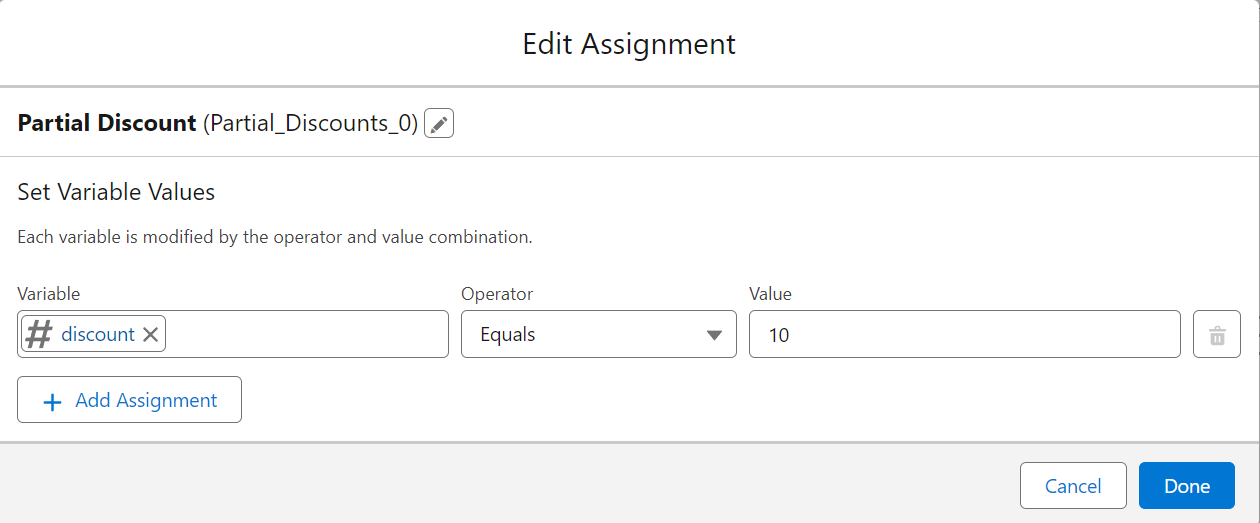
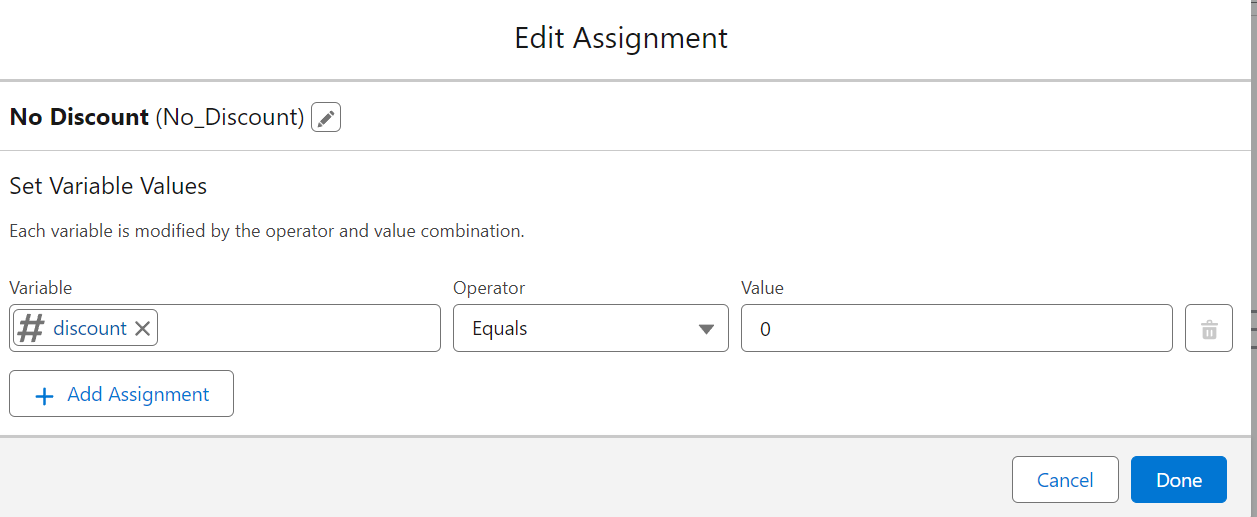


* 2. Decision Element: Create 2 Outcomes

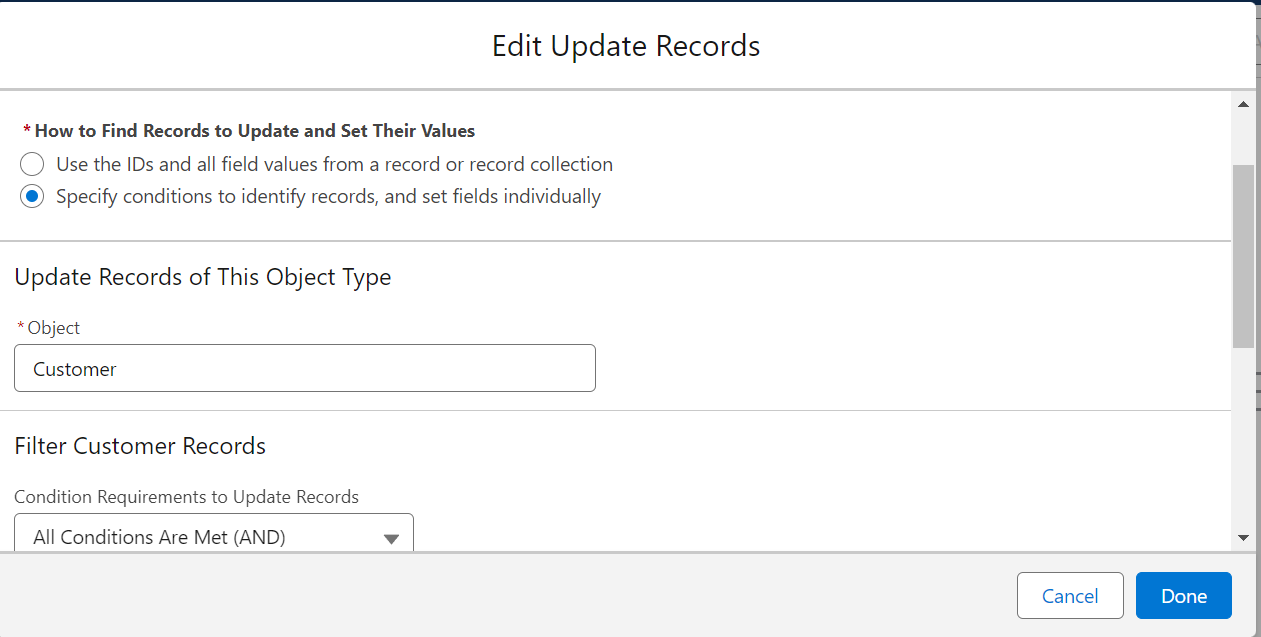


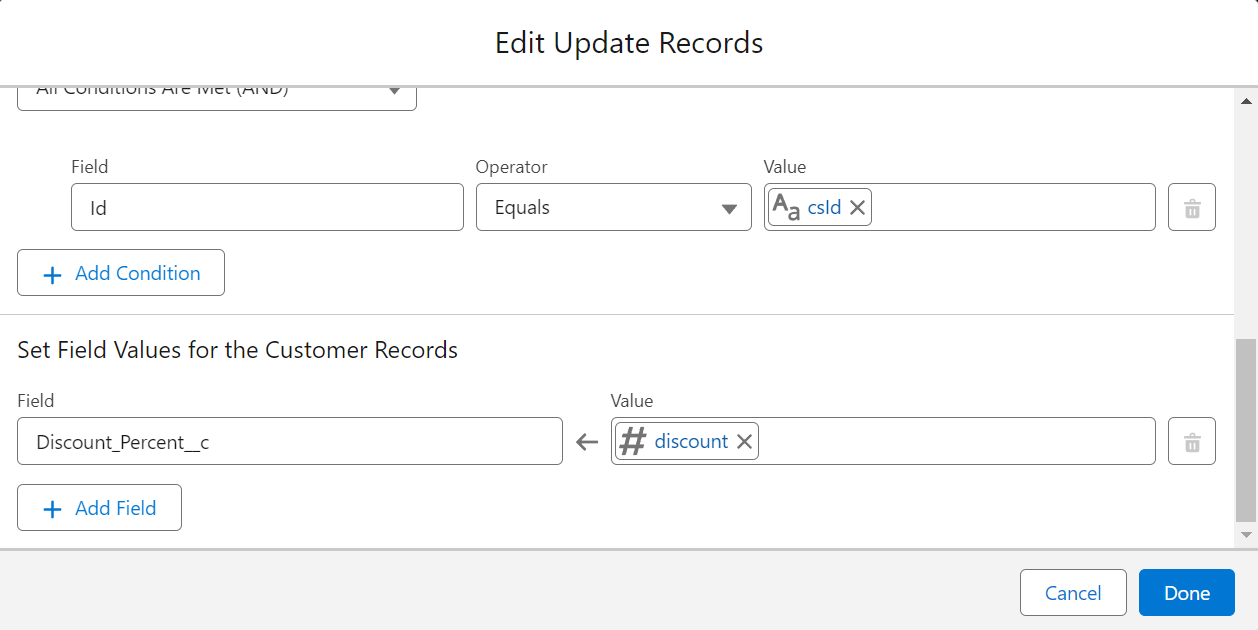
* Take the 3 Assignments > Full Discount, Partial Discount & No Discount



Edit Assignment

**Update Record Element**

****

****

### Apex trigger With Handler

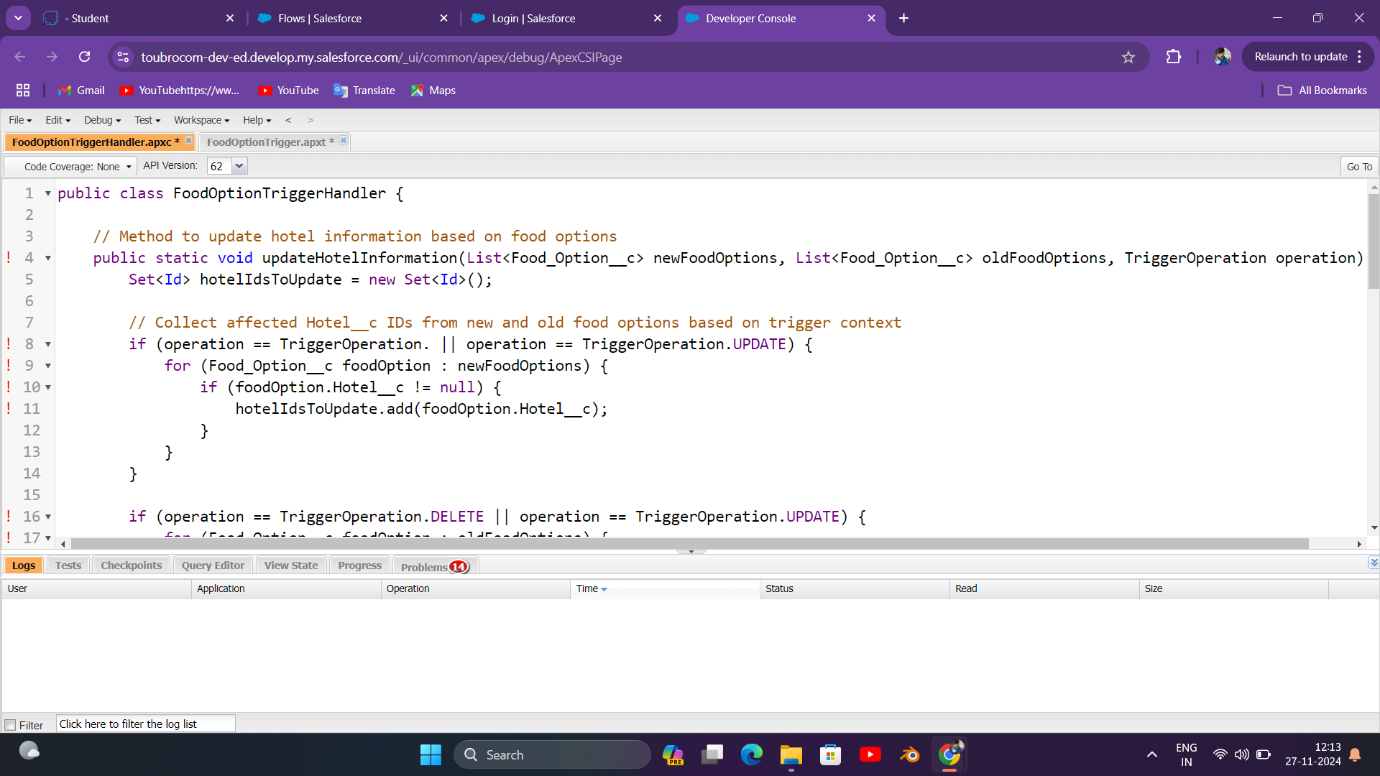
Your provided Apex code demonstrates the concept of using a trigger handler to separate business logic from the trigger itself, which is a best practice in Salesforce development. However, there are a few improvements and corrections needed for both the handler and the trigger logic:

**Key Issues:**

1. **Incorrect Method Signature:**  
   The method updateHotelInformation expects three parameters, but you are only passing one (trigger.new) in the trigger.
2. **Trigger Operation Logic:**  
   You need to handle different scenarios for insert, update, and delete.
3. **SOQL in Loop:**  
   You have a SOQL query inside a loop, which can lead to governor limit issues.

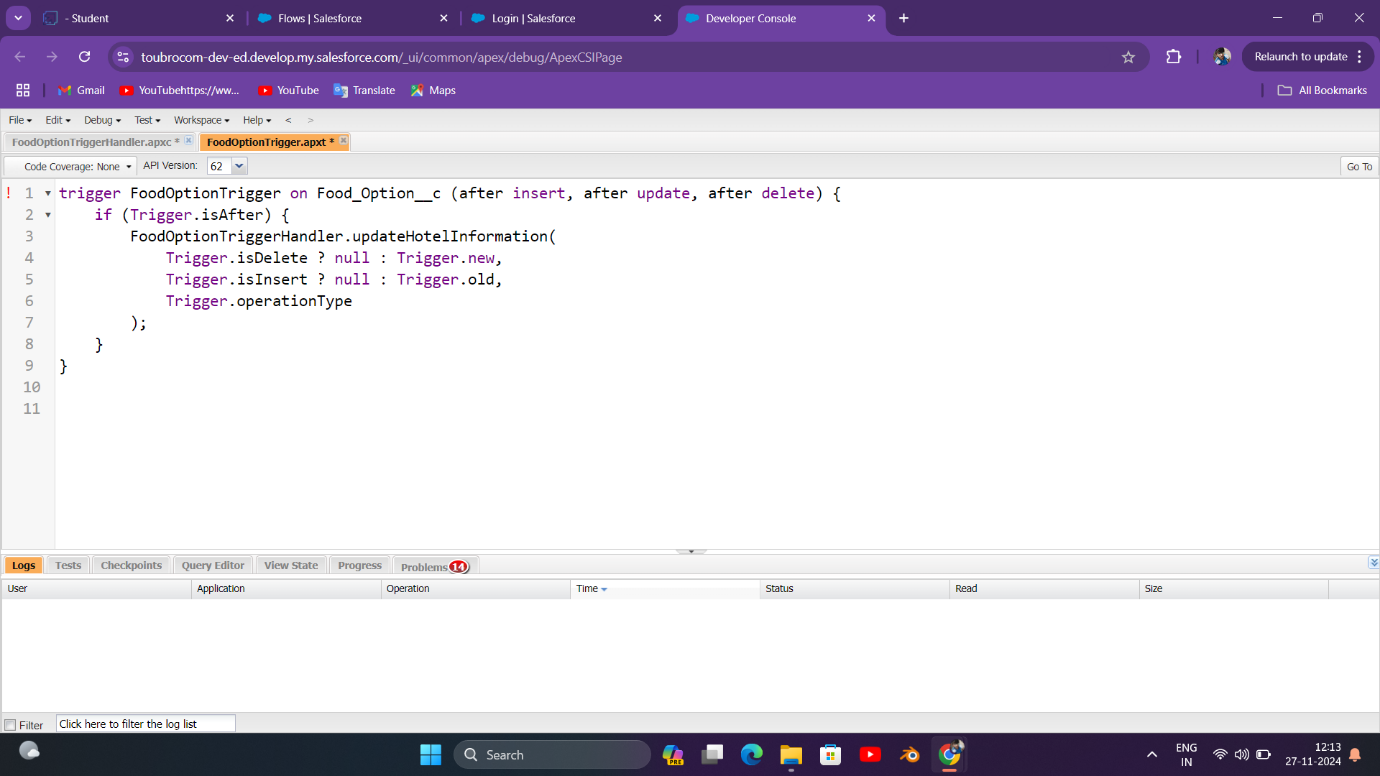
**Corrected Code:**

**Handler Class:**

****

**Changes Made:**

1. **Method Signature Adjustment:**  
   Now updateHotelInformation accepts both new and old lists of food options and a TriggerOperation enum to differentiate between insert, update, and delete actions.
2. **Handling Different Trigger Events:**  
   The trigger logic correctly passes empty lists where appropriate and differentiates between insert, update, and delete.
3. **SOQL Optimization:**  
   The SOQL query to count food options is now outside the loop, ensuring it complies with best practices and avoids governor limits.



### 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

### 

### Solution: Scheduled Apex to Send Reminder Emails for Flight Bookings

### Steps:

### Create a custom object and fields (if not already present):

### Object: Flight\_Booking\_\_c

### Fields:

### Customer\_Email\_\_c (Email) - to store the customer’s email address.

### Flight\_Date\_\_c (Date/Time) - to store the flight's scheduled departure date and time.

### Apex Scheduled Class: The class will query flight bookings scheduled for exactly 24 hours from now and send reminder emails.

### 

#### **Scheduling the Apex Job:**

You can schedule this job to run daily using the System.schedule method or through the UI.

##### **Via Developer Console:**

### 

### Testing and Validation

### Testing Approach:

### Unit Testing:

### Apex classes and triggers tested with over 90% code coverage.

### Key Classes:

### ▪ FoodOptionTriggerHandler

### ▪ FlightReminderScheduledJob

### User Interface Testing:

### Verified record updates on custom object pages.

### Tested dynamic behavior of food option counts on hotel pages.

### Conclusion

### Summary of Achievements:

### • Delivered a robust travel management solution with Salesforce as the backbone.

### • Automated key processes like food option tracking and flight reminders.

### • Enhanced customer experience by providing dynamic updates and notifications.

### • Improved data accuracy and operational efficiency through automation.

# THANK YOU