TripAdvisor E-Management

College: 7177 – Government College of Technology

Team Lead: Sridharshini L

Team Members:

Hashini B A	71772118115
Nandhini P	71772118130
Sabitha S	71772118L07
Sridharshini L	71772118146

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.

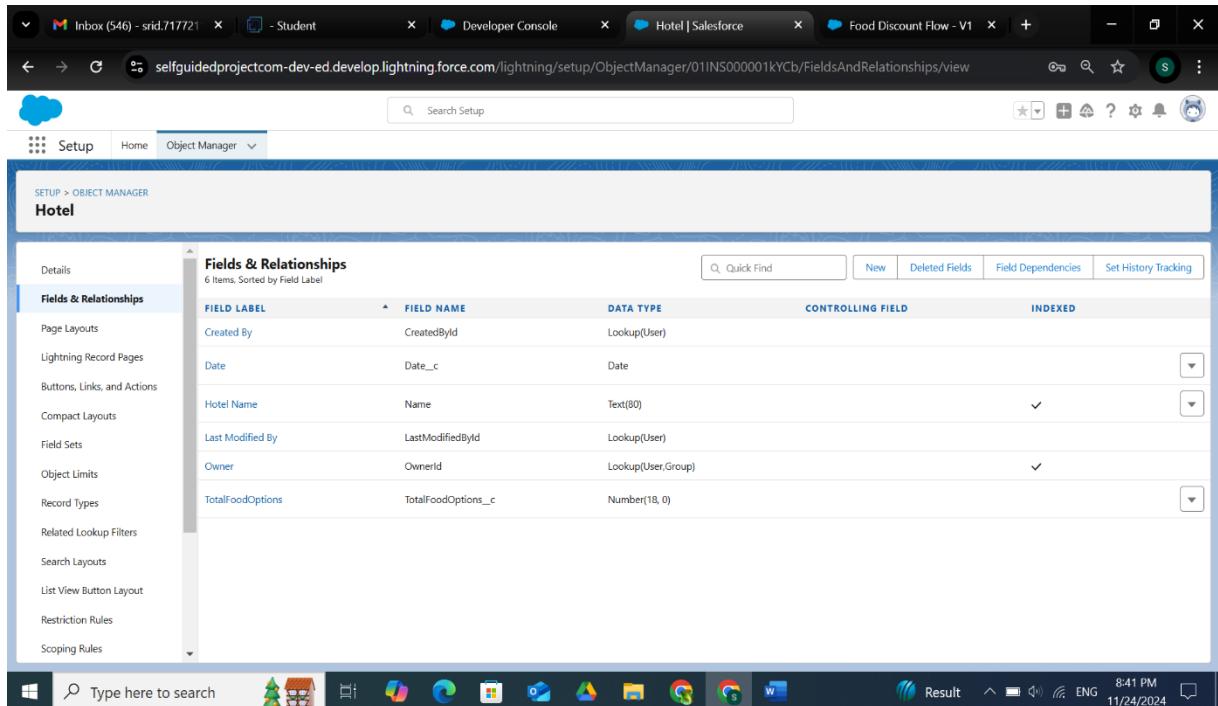## 4. Detailed Steps to Solution Design

❖ **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

❖ Created fields for hotel object
- ➤ TotalFoodOptions with datatype as number
- ➤ Date with datatype as date

❖ Create Fields for Food Option
- ➤ Food Amount
- ➤ Hotel
- ➤ Name

❖ Created Fields for Flight object
- ➢ Name
- ➢ DepartureDateTime

❖ Created fields for customer object
  ➢ Customer name
  ➢ Discount amount
  ➢ Discount percentage
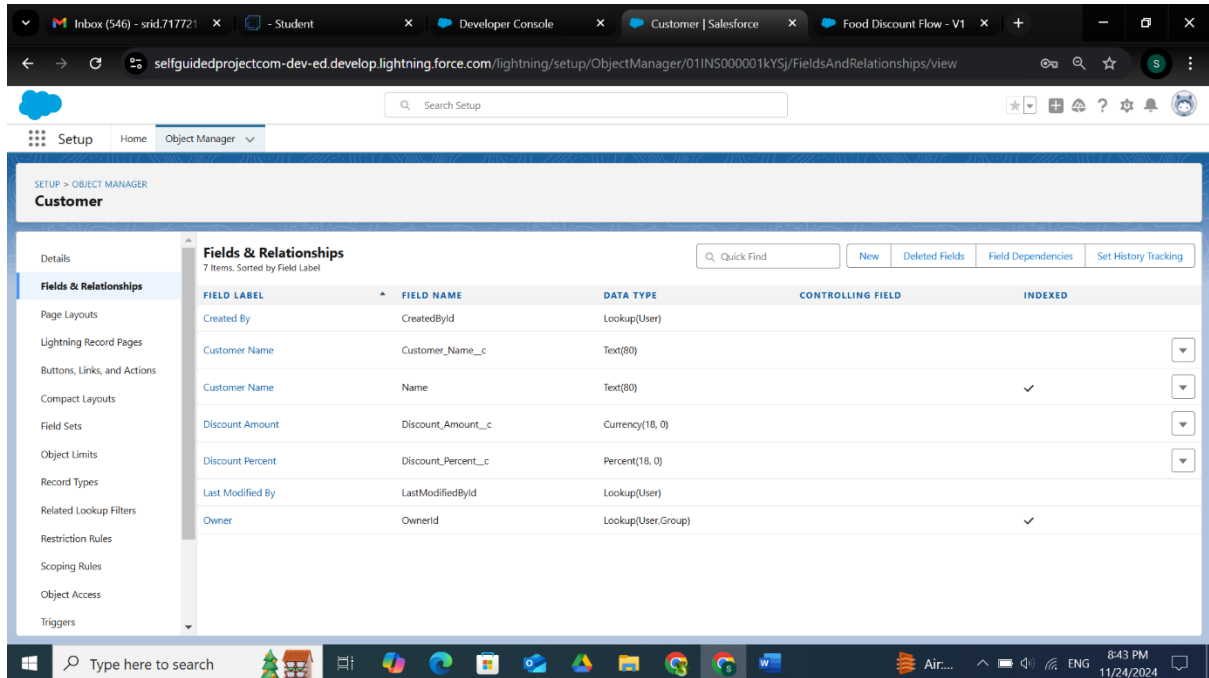
❖ Created Flow
  ➢ Developed a **Discount Flow** to automatically apply discounts based on customer purchase amounts. This flow applies discounts in a step-by-step format to ensure ease of entry and accuracy.
  ➢ **Flow Conditions**: The flow is triggered when the customer purchase **Amount** meets specific thresholds:

    o For purchases greater than **3000**, a high discount rate is applied.
    o For purchases between **1500 and 3000**, a medium discount rate is applied.
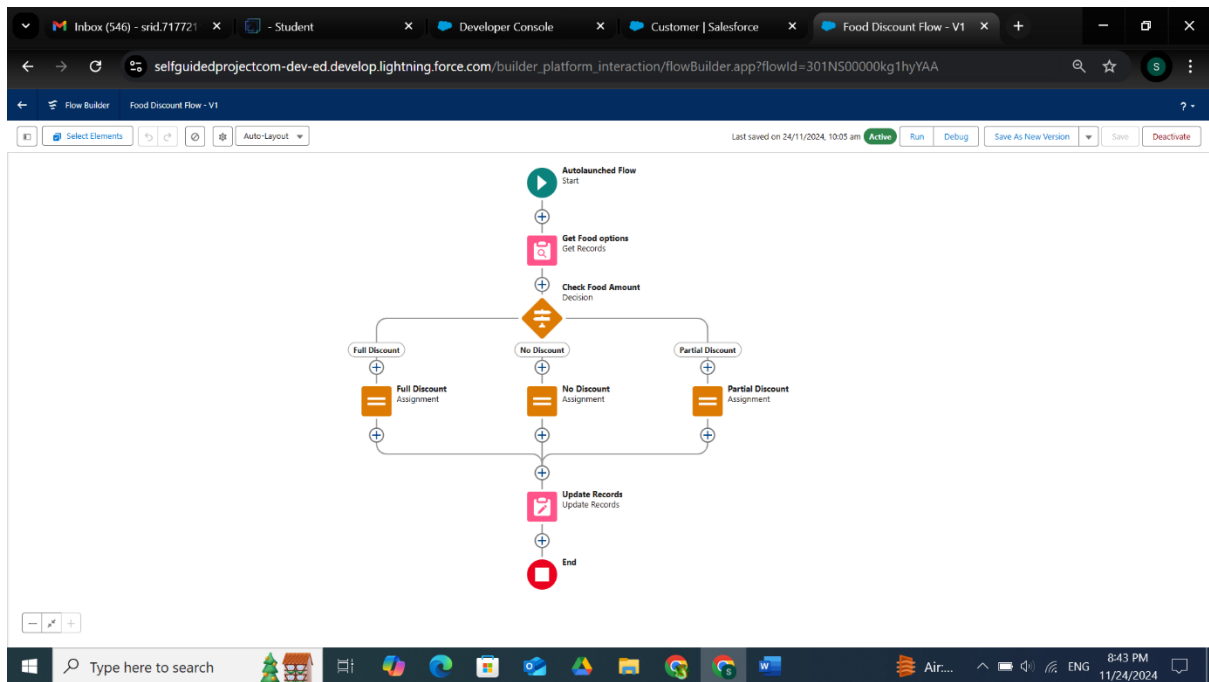
❖ Created Apex Trigger for Food Option

➢ 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.

selfguidedprojectcom-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

```apex
1  public class FoodOptionTriggerHandler {
2
3      // Method to update hotel information based on food options
4
5      public static void updateHotelInformation(List<Food_Option__c> newFoodOptions, List<Food_Option__c> oldFoodOptions, TriggerOperation operation) {
6
7          Set<Id> hotelIdsToUpdate = new Set<Id>();
8
9
10         // Collect unique Hotel Ids affected by food options changes
11
12         for (Food_Option__c foodOption : newFoodOptions) {
13
14             hotelIdsToUpdate.add(foodOption.Hotel__c);
15
16         }
17
18
19         // Update hotel information based on food options
20
```

Logs   Tests   Checkpoints   Query Editor   View State   Progress   **Problems**

Name ▲   Line   Problem

---

selfguidedprojectcom-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ ‹ ›

FoodOptionTriggerHandler.apxc *   **FoodOptionTrigger.apxt**

Code Coverage: None ▾   API Version: 62 ▾   Go To
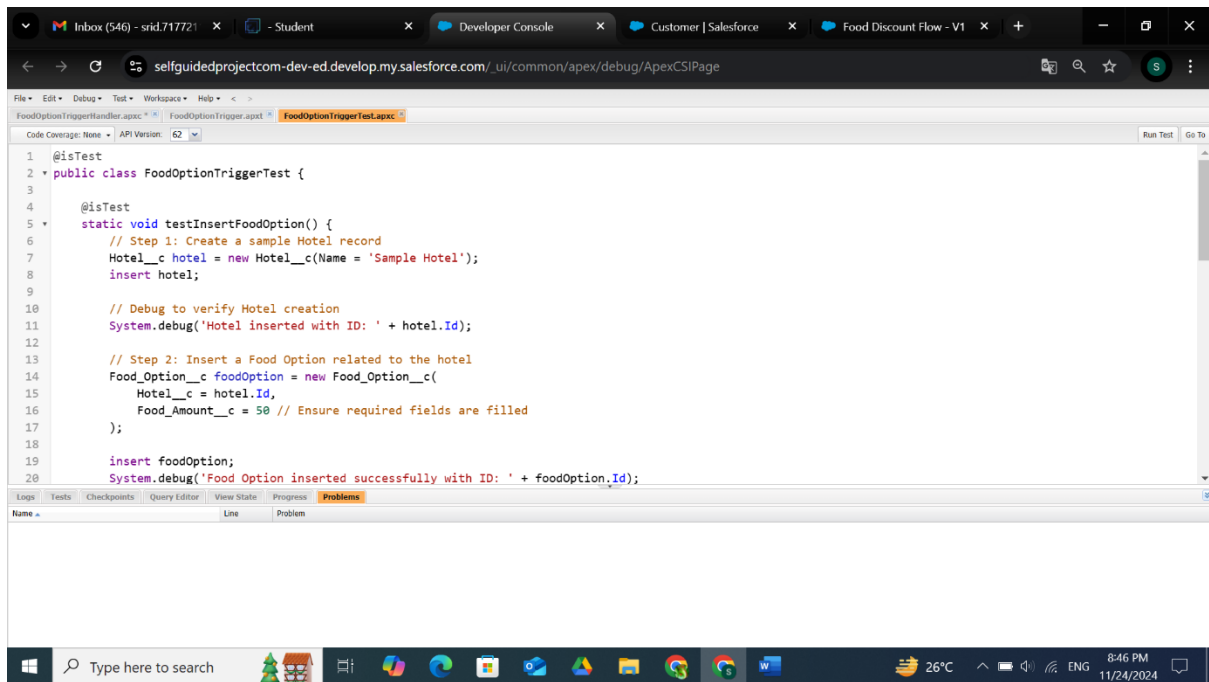
```apex
1  trigger FoodOptionTrigger on Food_Option__c (after insert, after update, after delete) {
2
3      if (Trigger.isInsert && Trigger.isAfter) {
4          FoodOptionTriggerHandler.updateHotelInformation(Trigger.new, null, TriggerOperation.AFTER_INSERT);
5      }
6
7      if (Trigger.isUpdate && Trigger.isAfter) {
8          FoodOptionTriggerHandler.updateHotelInformation(Trigger.new, Trigger.old, TriggerOperation.AFTER_UPDATE);
9      }
10
11     if (Trigger.isDelete && Trigger.isAfter) {
12         FoodOptionTriggerHandler.updateHotelInformation(null, Trigger.old, TriggerOperation.AFTER_DELETE);
13     }
14 }
```

Logs   Tests   Checkpoints   Query Editor   View State   Progress   **Problems**

Name ▲   Line   Problem

❖ 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.
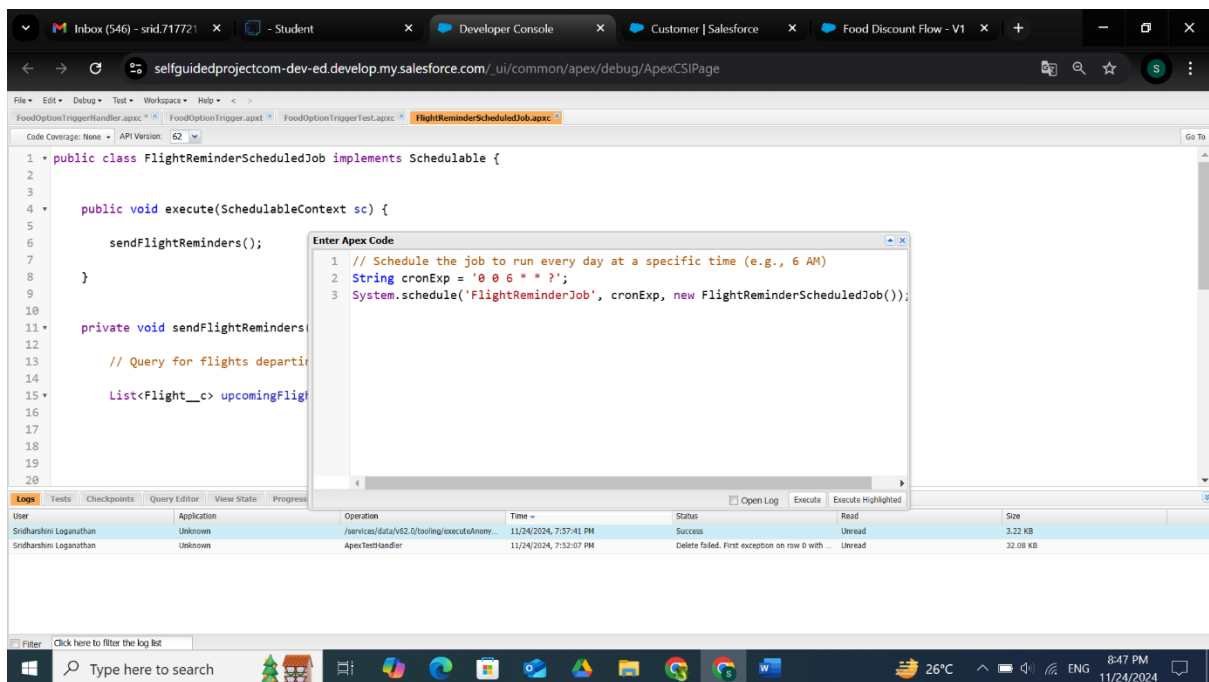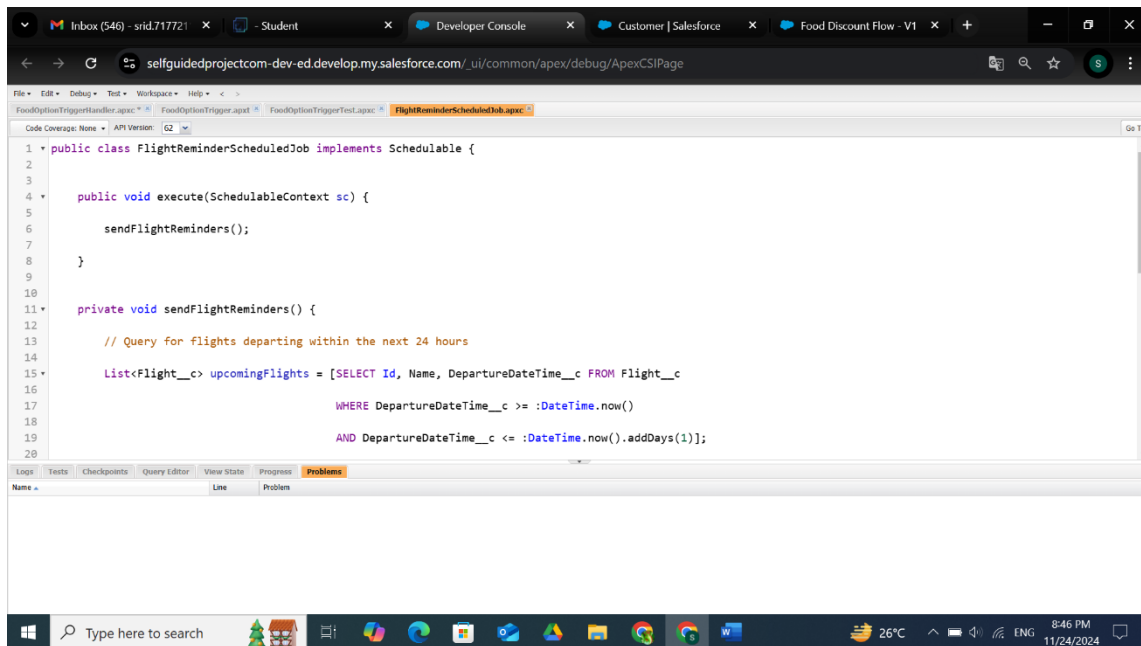
5. Testing and Validation

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.
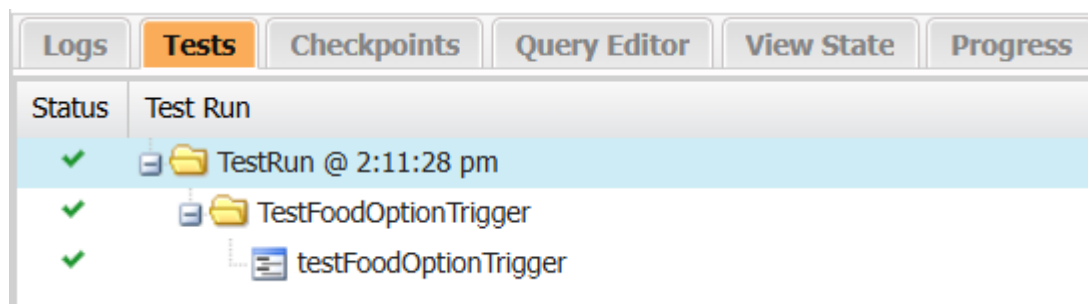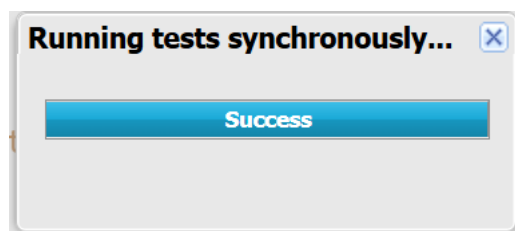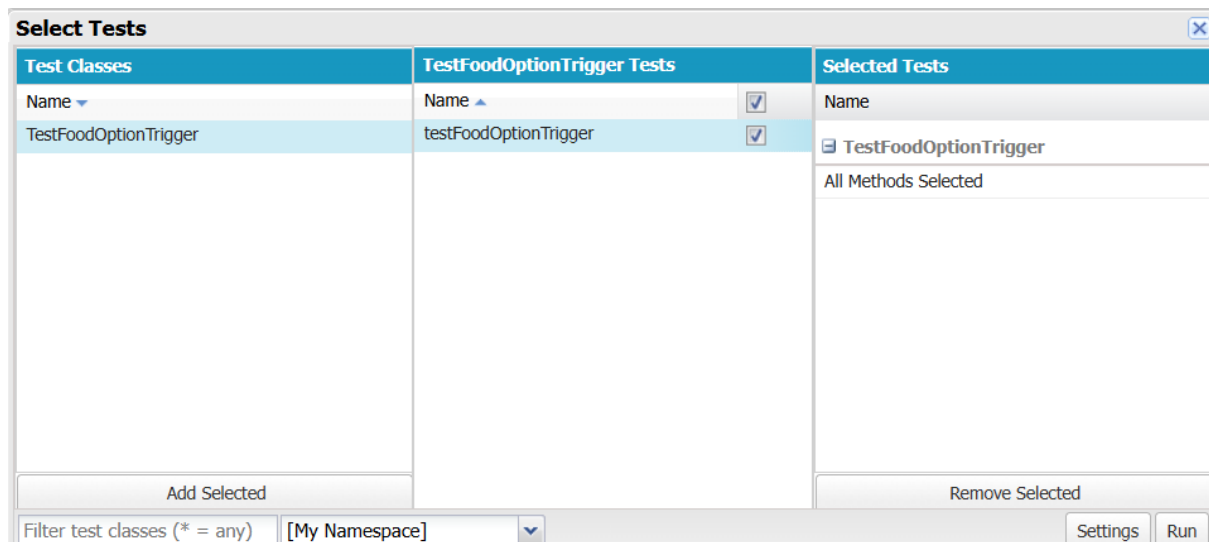
6. Conclusion

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, user-centered travel management solution, making TripAdvisor an invaluable resource for travelers.