NAAN MUDHAVALVAN- SALESFORCE REPORT

TRIPADVISOR E-MANAGEMENT

PROJECT CREATED BY

Naviya S - 420422205021

Harini L - 420422205009

Agalya R - 420422205001

Deepika J - 42042225005

DEPARTMENT OF INFORMATION TECHNOLOGY

ADHIPRASAKTHI ENGINEERING COLLEGE,MELMERUVATHUR

TABLE OF CONTENTS

1.Project Overview

The Tripadvisor E-Management project will integrate Salesforce CRM to streamline sales,

customer service, and marketing processes. It will centralize customer data, automate workflows,

and improve case management through Service Cloud. Marketing Cloud will enable personalized

campaigns, while custom reports will provide real-time insights. The solution will be tailored to

Tripadvisor's needs and integrated with existing systems. This will enhance efficiency,

collaboration, and customer satisfaction across teams.

2.Project Objective

Enhance User Experience: Develop an integrated system to streamline user interactions,

including personalized recommendations, seamless booking, and optimized review management.

Automate Operational Processes: Implement automation for key functions such as content

moderation, feedback management, and customer service to improve efficiency and reduce manual

workloads.

Improve Data Analytics: Utilize data analytics to offer tailored travel recommendations and

insights, enhancing user satisfaction and engagement with the platform.

Optimize Customer Support: Integrate real-time customer support features, such as chatbots and

automated responses, to provide quicker resolutions and improve overall service quality.

Drive Revenue and Scalability: Develop a scalable system that supports TripAdvisor's global

expansion, enhances partnership integrations, and drives operational growth and profitability

3.Key Features and Concepts Utilized

1.Sales Cloud: Manages leads, opportunities, accounts, and contacts, streamlining the sale pipeline and enhancing lead conversion processes

2.Service Cloud: Improves customer support by automating case management, ensuring faster response times, and providing a unified view of customer issues.

3.Marketing Cloud: Facilitates personalized marketing campaigns, customer segmentation, and targeted communications to boost engagement and sales.

4.Reports & Dashboards: Provides real-time, customizable analytics to track sales performance, customer satisfaction, and operational.

5.Process Builder & Flow: Automates complex workflows like lead routing, task creation, and approval processes, reducing manual effort and increasing efficiency.

6.Apex:Custom code for advanced business logic, triggers, and integrations with external systems, enabling tailored functionality.

7.Salesforce Lightning: A modern, user-friendly interface for both desktop and mobile, enhancing user experience and simplifying navigation.

8.Chatter: Facilitates internal collaboration across departments by providing a social platformfor sharing updates, comments, and documents in real-time.

4.Detailed Steps to Solution Design

Step 1:

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

Procedure:

Create Hotel Object

• Label: Hotel

• Plural Name: Hotels

• Fields:

o Hotel Name (Text)

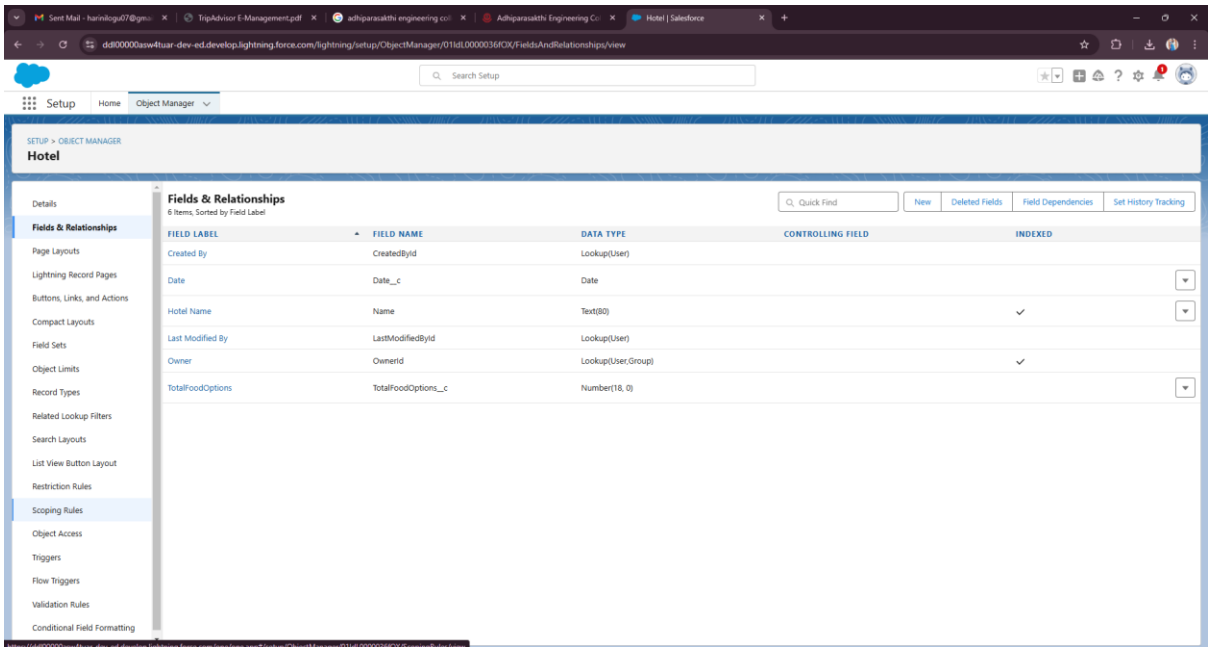
o TotalFoodOptions (Number)

o Date (Date)

• Allow Reports: Yes

• Allow Search: Yes

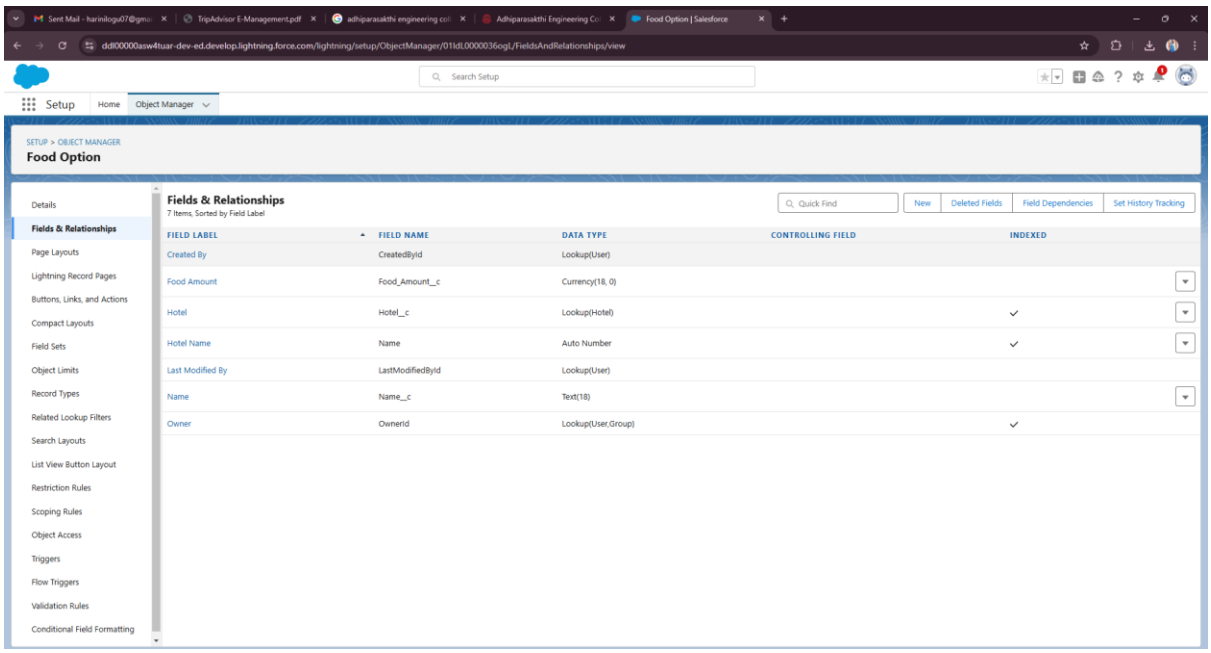
Figure: 1



2.Create Food Option Object

- Data Type: Auto Number
- Format: FO{0000}
- Fields Name (Text)
- Hotel (Lookup to Hotel Object)
- Food Amount (Currency)

Figure: 2



3.Create Flight Object

• Data Type: Auto Number

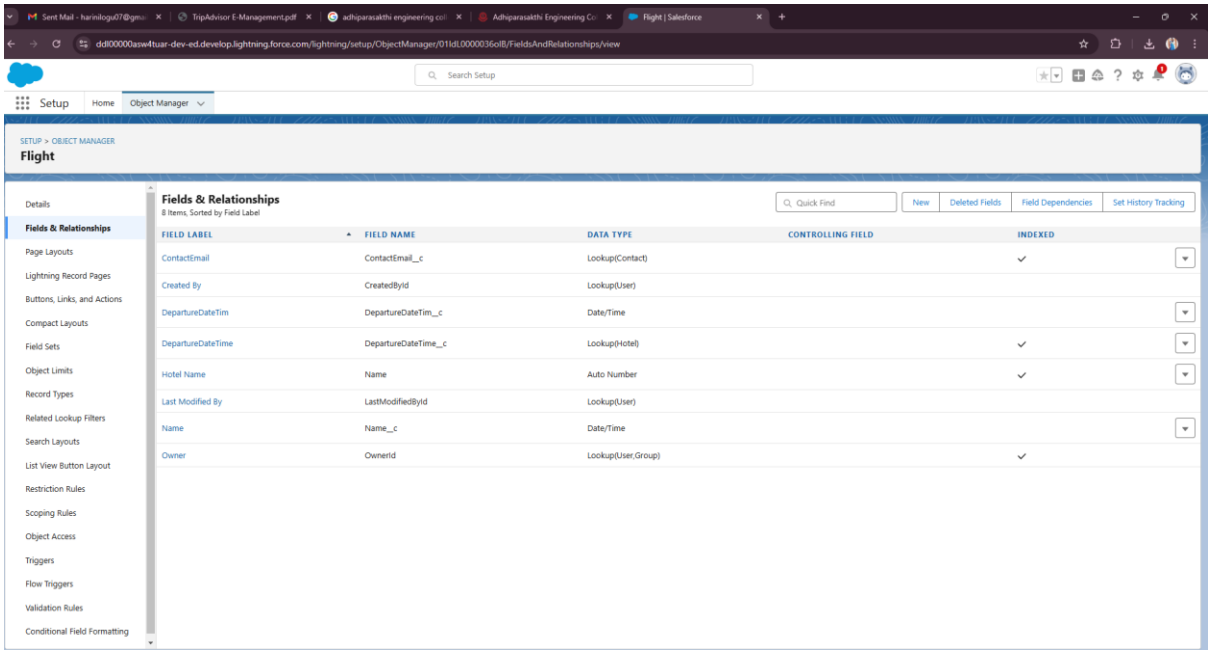
• Format: FL-{0000}

• Fields Name (Date/Time)

DepartureDateTime (Date/Time)

Hotel (Lookup to Hotel Object)

Figure: 3



4.Create Customer Object

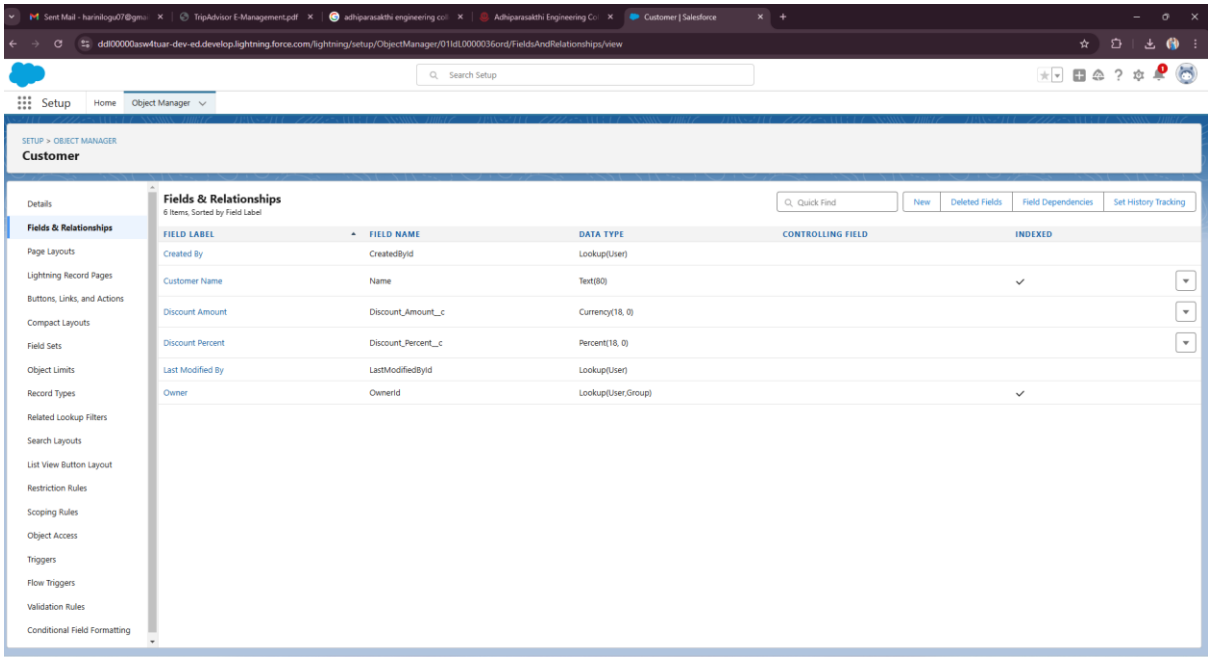
Fields:

o Customer Name (Name)

o Discount Amount (Formul Currency)

o Discount Percent (Percentage)

Figure: 4



Step 2: 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

Create Variables

• foId: Text (Available for Input)

• csId: Text (Available for Input)

• discount: Number (Available for Input)

Flow Steps

Get Records

Get relevant records based on foId and csId. This can be used to pull the records you will need for your flow processing.

Decision Element

Create a decision element that will evaluate whether the discount is a Full, Partial, or No Discountbased on the discount variable.

Full Discount: If discount = 100%

Partial Discount: If discount > 0% and < 100%

No Discount: If discount = 0% or null

Assignments

Create three assignment elements to set different values or actions based on the discount outcomes:

Full Discount: Apply actions for a full discount.

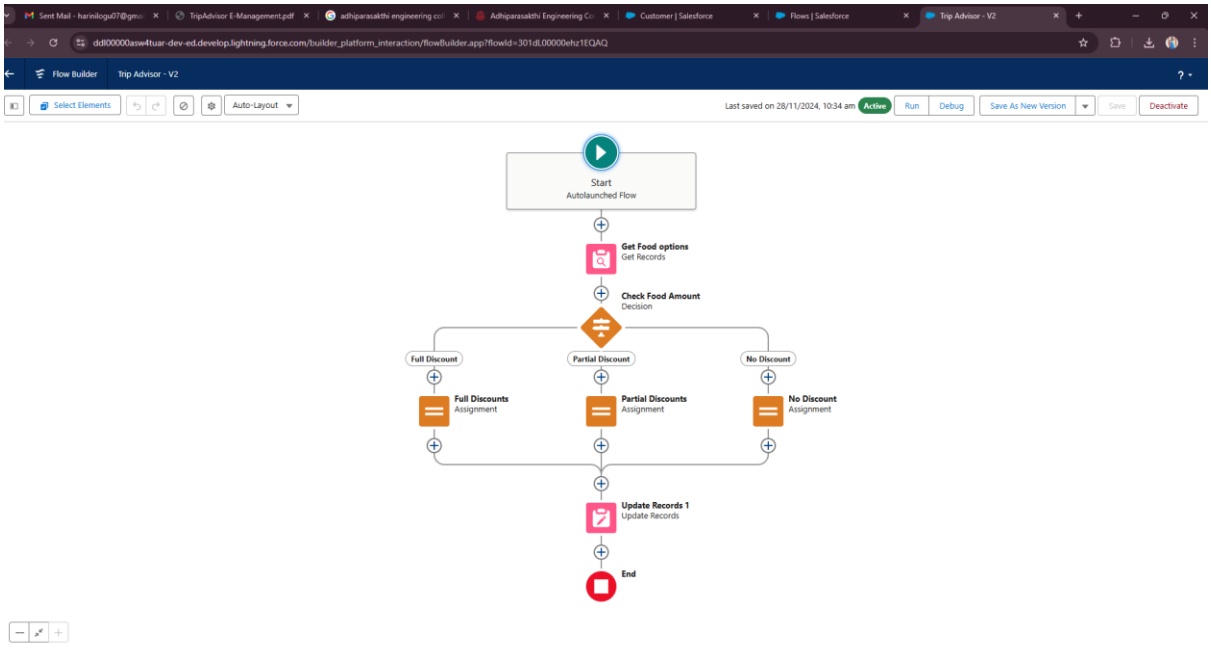
Partial Discount: Apply actions for a partialdiscount.

No Discount: Apply actions for no discount.

Update Record

After applying the appropriate assignments, update the records (e.g., the customer or order record) with the new discount information or other relevant fields.

Figure: 5



Step 3:

Apex Triggers

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

Apex trigger With Handler

Apex class

```
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);

}

13

// 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;
}
}
```

Apex Trigger

```
trigger FoodOptionTrigger on Food_Option c (after insert, after update, after delete) {
if (Trigger.isInsert && Trigger.isAfter) {
// Call the handler for after insert, passing Trigger.new, an empty list for old records, and the
operation type 'INSERT'
FoodOptionTriggerHandler.updateHotelInformation(Trigger.new,
new List<Food_Option c>(), 'INSERT');
}
if (Trigger.isUpdate && Trigger.isAfter) {
// Call the handler for after update, passing Trigger.new,
Trigger.old, and the operation type 'UPDATE'
FoodOptionTriggerHandler.updateHotelInformation(Trigger.new, Trigger.old, 'UPDATE');
}
if (Trigger.isDelete && Trigger.isAfter) {
// Call the handler for after delete, passing an empty list for new records, Trigger.old, and the
operation type 'DELETE'
FoodOptionTriggerHandler.updateHotelInformation(new
List<Food_Option c>(), Trigger.old, 'DELETE');
```

Figure:6

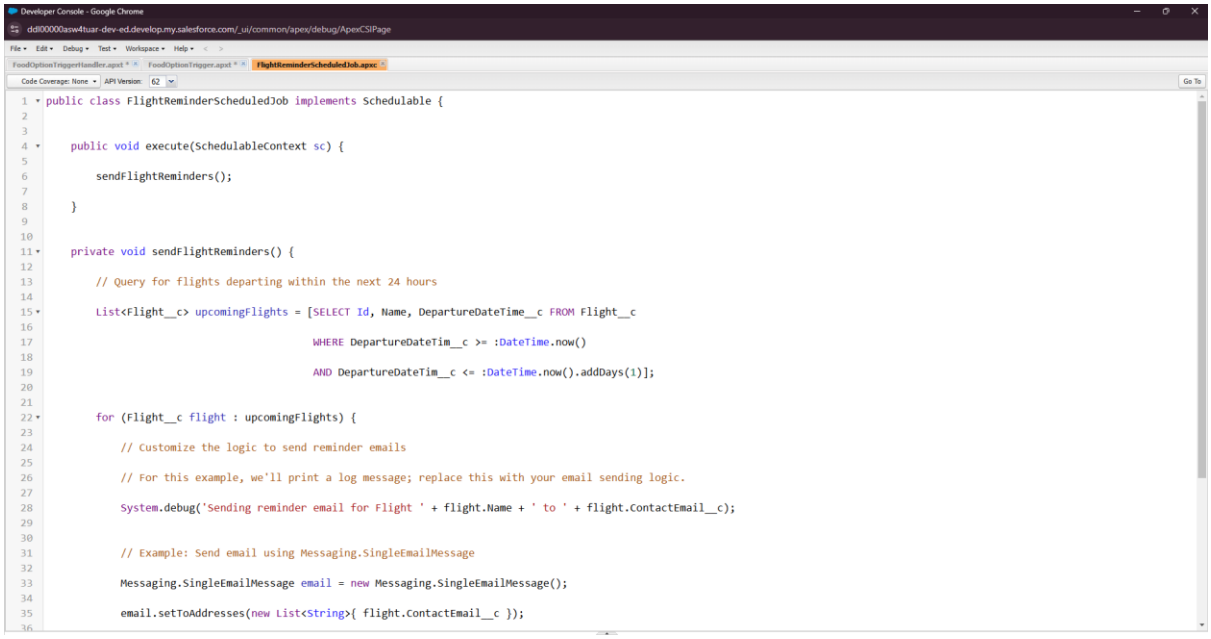
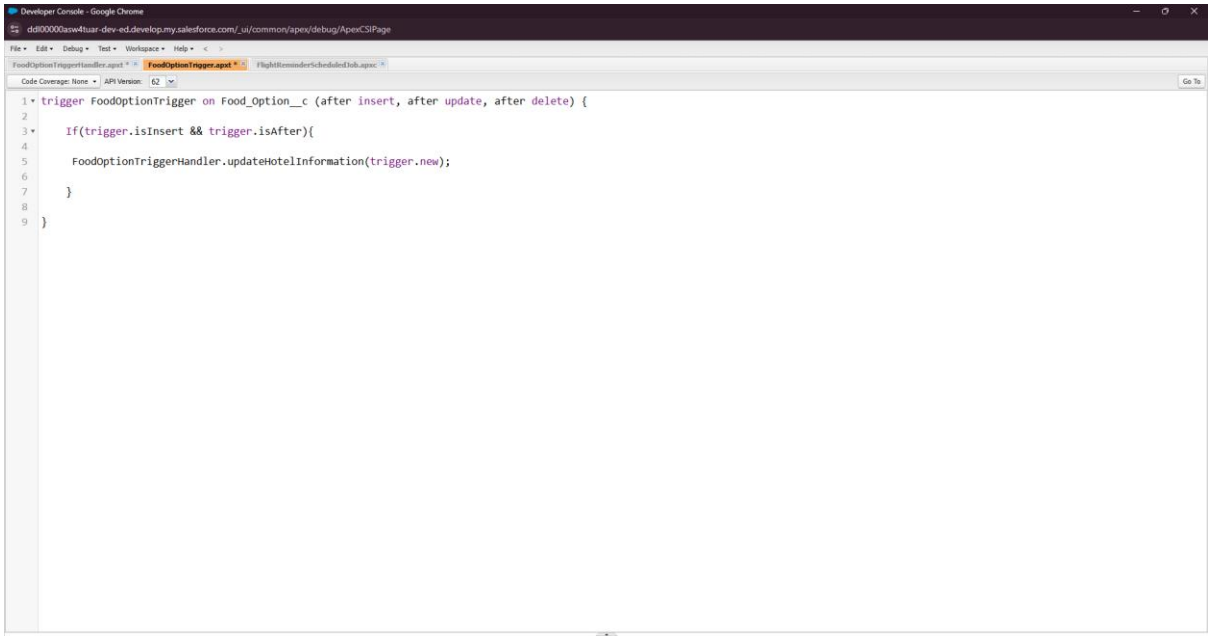


Figure:7



Step 4:

Apex Schedule

Create the Reminder mail for the customer who has booked the flight according to that bookingset the Apex schedule so mail will be sent prior to 24hrs.Note: Please create the required field for Scheduled Apex Code

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 });

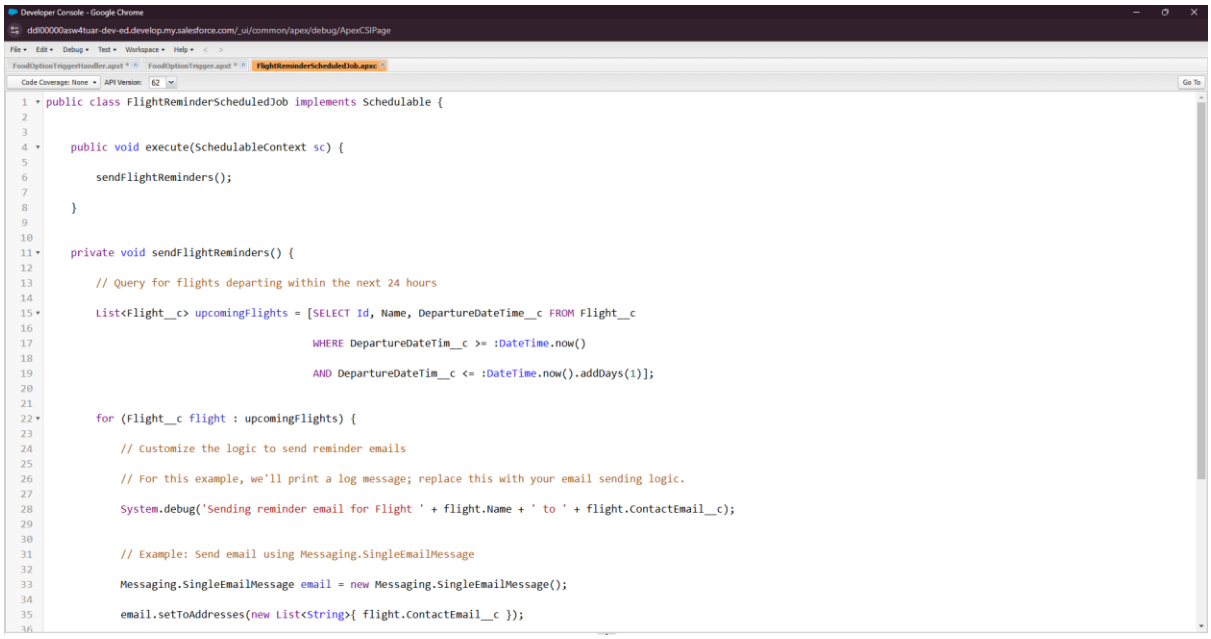
}

}

}

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.

Figure:8



Apex code

Create the Apex code in an anonymous Window to execute the

Apex Code

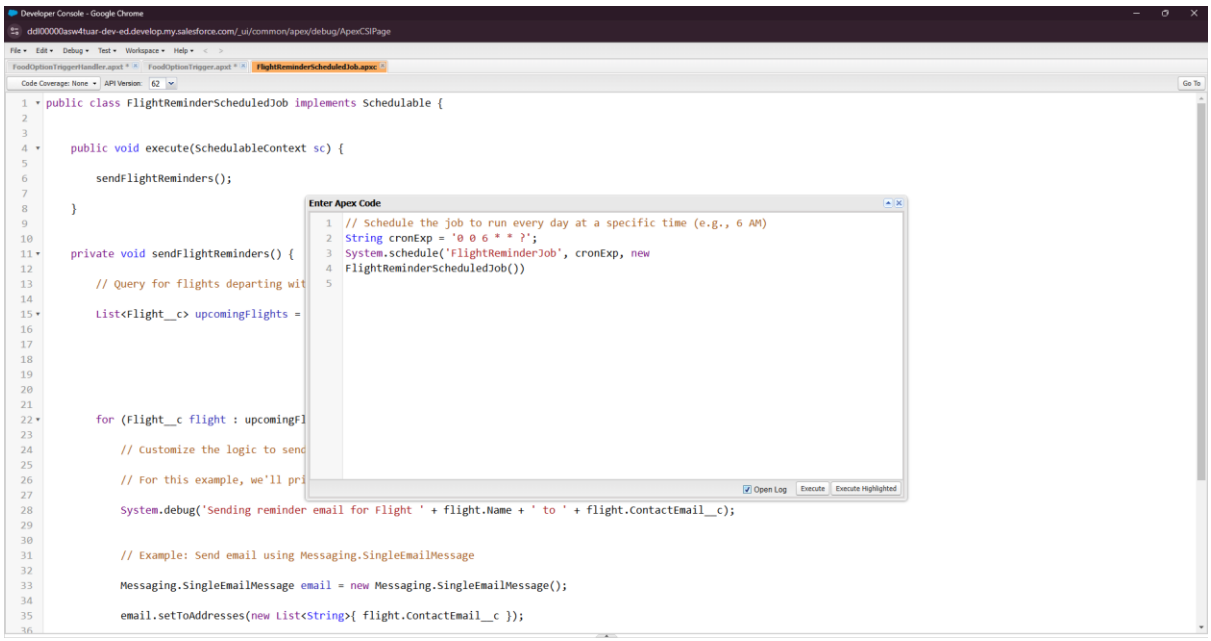
// Schedule the job to run every day at a specific time (e.g., 6 AM)

String cronExp = '0 0 6 * * ?';

System.schedule('FlightReminderJob', cronExp, new

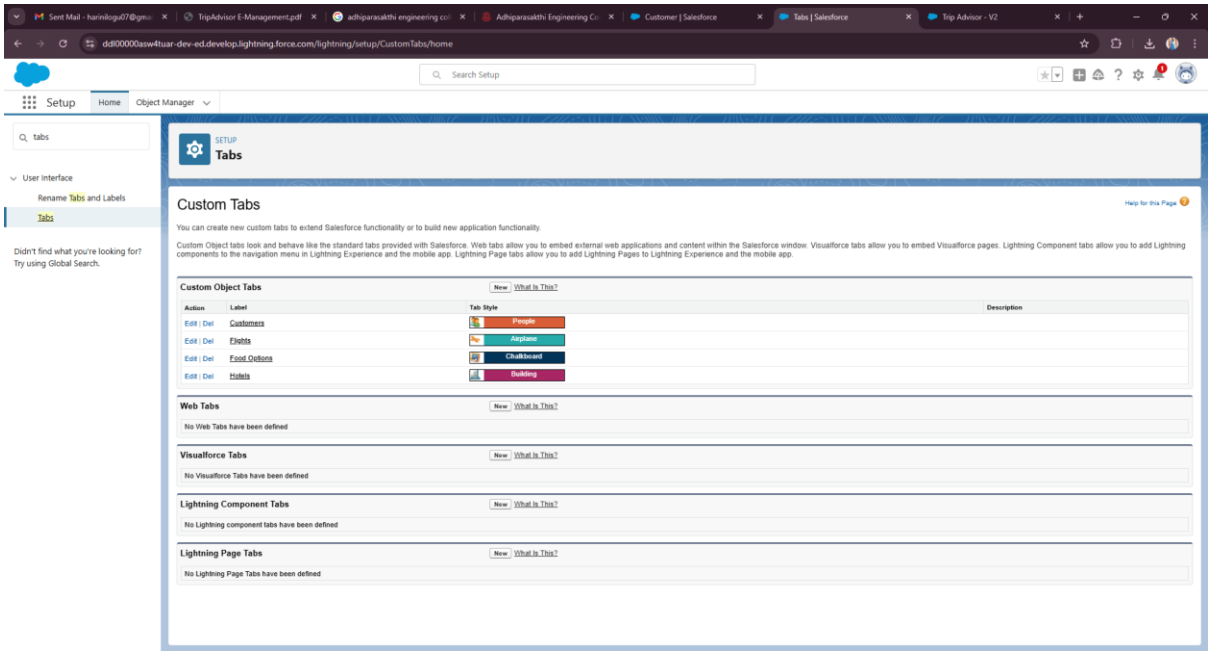
FlightReminderScheduledJob())

Figure:9



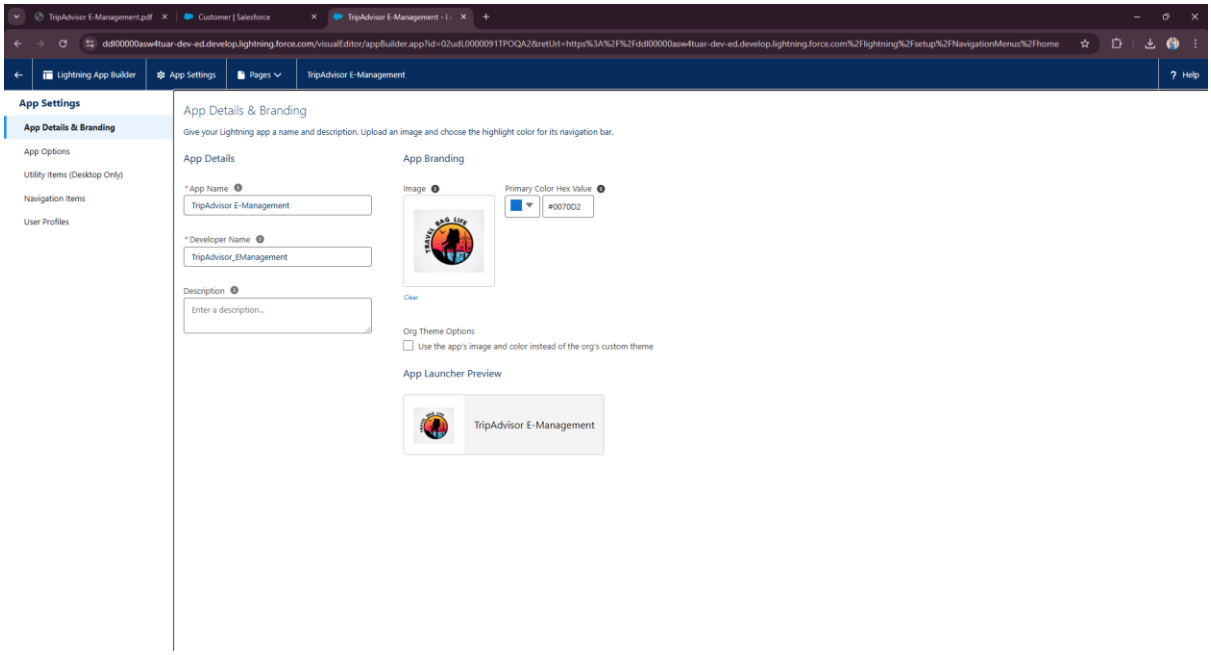
Create Tabs

Figure:10



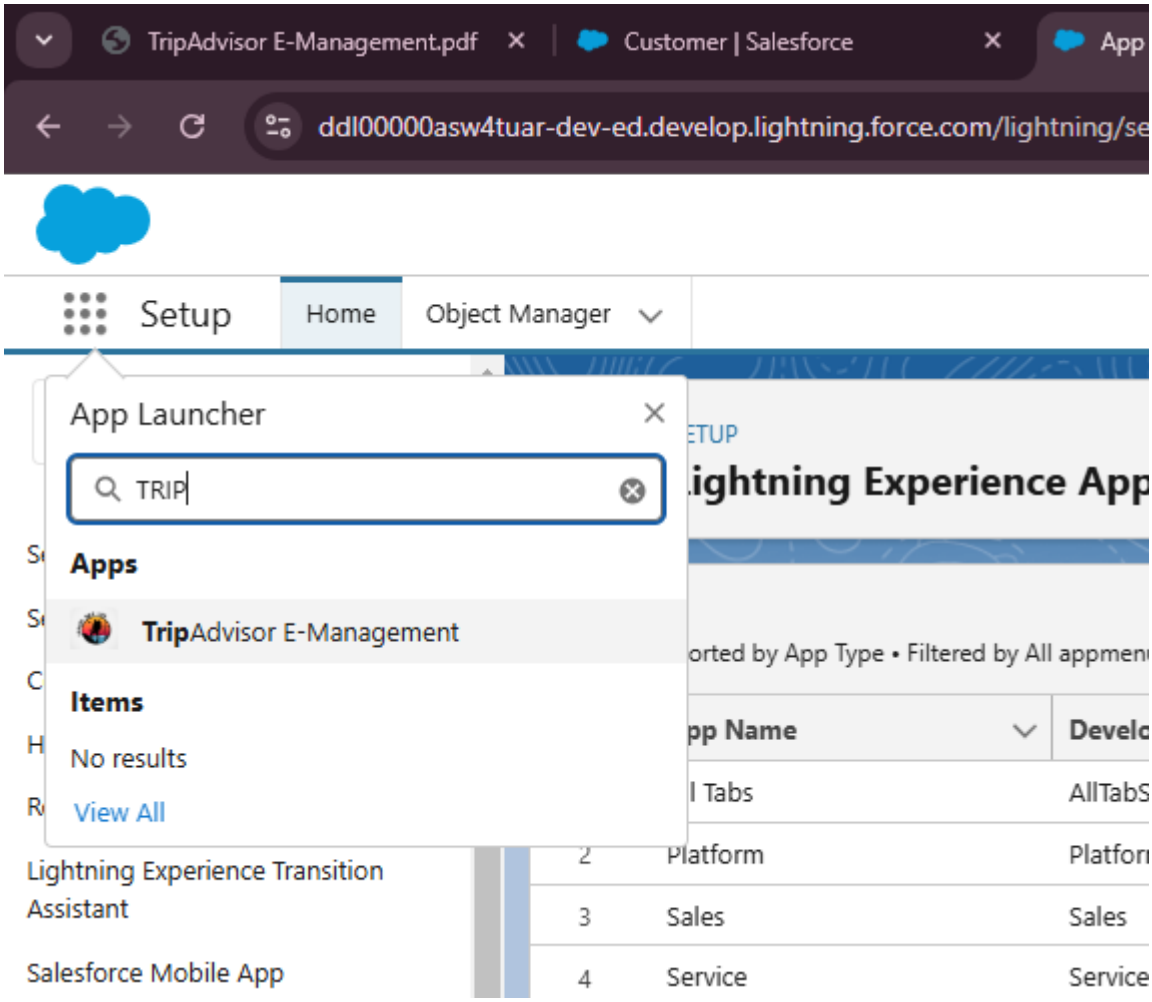
Create New Lighting App

Figure:11



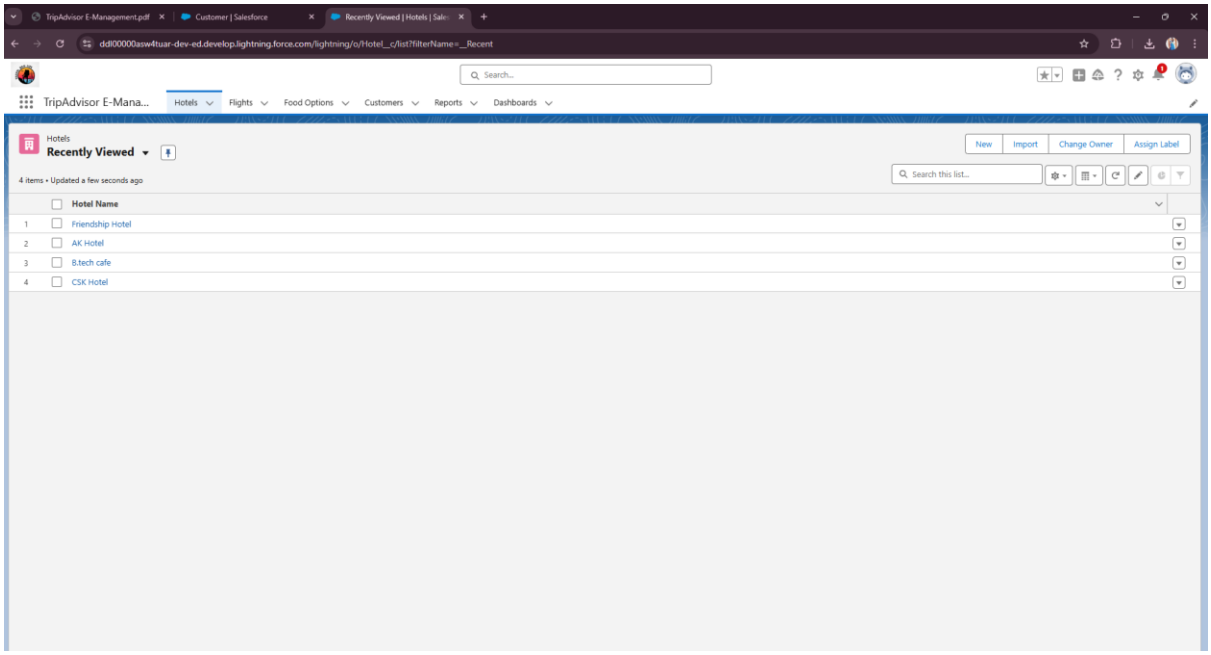
APP CREATION

Figure:12



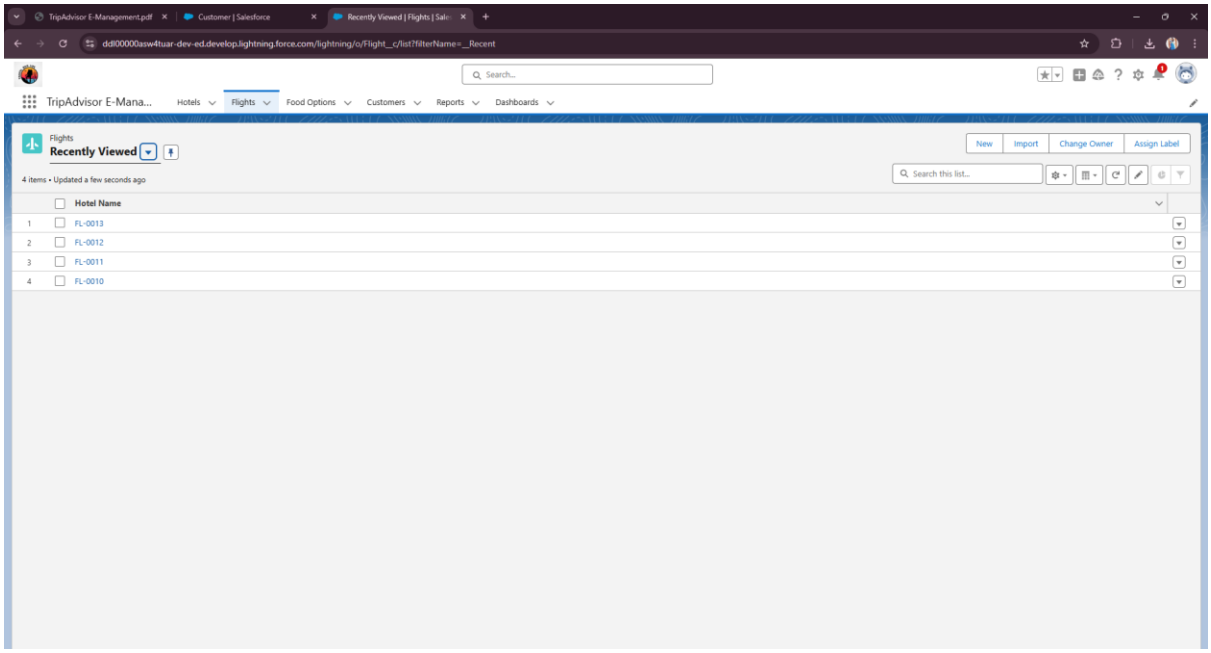
FINAL OUTCOMES

FIGURE:13



Dashboards

Figure:14



5.Testing and Validation

Testing Approach

- *Unit Testing: Verify individual discount components for correct functionality.*
- *Integration Testing: Ensure smooth interaction between discount rules and booking data.*
- *System Testing: Validate discount application across all booking scenarios.*
- *User Acceptance Testing (UAT): Confirm discount logic meets business requirements.*
- *Performance Testing: Ensure fast and accurate discount processing during peak traffic.*

Test Scenarios

• Discount Calculation: Verifying full, partial, and no discounts are applied correctly based on rules (e.g., percentage-based, fixed amount).

• Discount Eligibility: Ensuring only eligible usersreceive discounts (e.g., loyalty members, special promotions).

• Discount Calculation: Verifying full, partial, and no discounts are applied correctly based on rules (e.g., percentage-based, fixed amount).

• Discount Eligibility: Ensuring only eligible usersreceive discounts (e.g., loyalty members, special promotions).

• Discount Application: Confirming discounts are reflected in booking totals.

• Edge Cases: Handling situations like expired discounts, invalid promo codes, and conflicts between discount .

Test Results

• Functional: All discount types (full, partial, no discount) were correctly applied in booking scenarios

• Usability: The discount process was intuitive for users, with no major UI issues.

• Performance: The system processed discount calculations in under 2 seconds, even under high load

• Security: Discount data was securely handled, with no vulnerabilities found

Validation Outcomes

UAT confirmed that discount rules were correctly applied based on businesslogic. Test cases for promotional eligibility, expired offers, and discountstacking were successfully validated.

6.Key Scenarios Addressed

• Booking Management: Streamlined the process of booking travel services, including automated confirmations and itinerary management.

• Customer Communication: Centralized all customer communication channels to provide a consistent experience across platforms.

• Personalized Marketing: Leveraged customer data to execute targeted marketing campaigns based on travel history and preferences.

• Data Analysis for Decision Making: Utilized Salesforce analytics to gather insights on booking trends, customer preferences, and campaign effectiveness.

• Customer Support: Enabled efficient handling of customer inquiries, cancellations, and service changes through Salesforce's Service Cloud.

7.Conclusion

The implementation of the Trip Advisor E-Management system using Salesforce successfully addressed the challenges faced by travel agencies in managing customer relationships and bookings. By utilizing Salesforce's powerful CRM and automation capabilities, the system provided a centralized, efficient, and data-driven solution for the travel and tourism industry. This

implementation ensures that travel businesses can offer a seamless experience to customers, enhance operational efficiency, and make informed business decisions