



ARUNAI ENGINEERING COLLEGE

(An Autonomous Institution)
Velunagar, Thiruvannamalai-606603

www.arunai.org



DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

BACHELOR OF TECHNOLOGY 2024-2025

FIFTH SEMESTER

SB8067-SALESFORCE DEVELOPER

TRIPADVISOR E-MANAGEMENT

TEAM MEMBERS:

B.Sarathi Kannan	- 510422243050
D.Pugazheshwar	- 510422243042
A.Riaz Mohammed	- 510422243047
G.Hariprath	- 510422243018

ARUNAI ENGINEERING COLLEGE

(An Autonomous Institution)
TIRUVANNAMALAI - 606 603



DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

Certified that this is a bonafide record of work done by

Name : B.Sarathi Kannan
NM ID : CD24F010B1840D78849493FE119C1D33
University Reg.No : 510422243050
Semester : V
Branch : B.Tech – [AI & DS]
Year : 2024-2025

Staff-in-Charge

Head of the Department

Submitted for the _____
Practical Examination held on _____

Internal Examiner

External Examiner

TRIPADVISOR E-MANAGEMENT

1. Project Overview:

This project focuses on e-Management for Tripadvisor, designed to address the challenge of efficiently managing content, user interactions, and platform compliance while automating key processes. The primary goal is to deliver a streamlined and user-friendly Salesforce-based solution tailored to Tripadvisor's needs.

By leveraging Salesforce's Lightning Platform, this project aims to enhance operational efficiency, minimize errors in managing platform operations, and improve user experience. The solution aligns with Tripadvisor's long-term goal of achieving seamless platform management, maintaining content quality, and ensuring timely compliance with industry standards.

2. Objectives:

Business Goals

1. **Streamline Content and Interaction Management:** Automate the end-to-end processes for managing user-generated content, business listings, and platform interactions, ensuring a seamless workflow for all stakeholders.
2. **Enhance Operational Efficiency:** Reduce time and manual effort in managing approvals, content moderation, and user communications.
3. **Ensure Data Accuracy and Compliance:** Eliminate errors by enforcing validation rules, maintaining content quality, and ensuring compliance with legal and platform standards.
4. **Improve Stakeholder Communication:** Use automated notifications and approval workflows to keep content creators, business partners, and internal teams informed and engaged.
5. **Enable Real-Time Reporting:** Provide comprehensive dashboards and reports for tracking platform metrics, content performance, and overall operational effectiveness.

Specific Outcomes

1. **Custom Salesforce Objects:**
 - Define objects for Reviews, Listings, and User Interactions to store all relevant platform data.
2. **Automated Workflows:**
 - Build Flows to manage content approvals, flagging inappropriate content, and sending timely notifications without manual intervention.
3. **Validation Rules and Business**
 - Enforce rules such as ensuring unique business listings, validating review authenticity, and maintaining compliance with content policies.

4. Approval Processes:

- Implement a multi-level approval process involving moderators, content teams, and compliance officers to streamline decision-making.

5. Dynamic Email Templates:

- Create templates for content approval updates, flagged content notifications, and engagement prompts for users and business partners.

6. Dashboard and Reporting:

- Provide interactive dashboards to track key metrics, including flagged content trends and listing performance, and user engagement levels.

7. Code and Integration Enhancements:

- Develop Apex triggers for custom logic and Schedule Classes for time-based automations, ensuring smooth operations at scale. Integrate third-party tools for enhanced data insights and operational capabilities.

3. Salesforce Key Features and Concepts Utilized:

The **Tripadvisor E-Management** project leverages the following Salesforce features and concepts to build a robust, scalable, and user-friendly solution:

1. Custom Objects

- **Reviews:** Tracks information like Review ID, Submission Date, Reviewer Name, and Rating.
- **Properties:** Stores details about business listings, including Business Name, Location, and Listing Status.
- **Tenants:** Maintains information such as User ID, Activity Type (e.g., review, comment), and Interaction Timestamp.

2. Tabs

- **Custom Tabs:** Tabs for Reviews, Listings, and User Interactions allow users to quickly access and manage platform data.
- **Standard Tabs:** Utilize tabs like Reports, Dashboards, and Tasks for seamless workflow management.

3. Lightning App Builder

- **Custom Lightning App for e-Management:** Integrated multiple tabs, dashboards, and workflows to provide a centralized view for managing reviews, monitoring listings, and tracking user engagement.

4. Fields and Validation Rules

1. Fields:

Custom fields such as Review Moderation Status, Business Category, and User Reputation Score.

2. Validation Rules:

- Ensure reviews contain a minimum word count.
- Prevent duplicate business listings.
- Validate ratings to be within the specified range (e.g., 1 to 5).

5. Email Templates

1. Dynamic Email Templates:

- Notify users when their reviews are approved or flagged.
- Alert business owners of new reviews on their listings.
- Send reminders to users for incomplete profile information or pending interactions.

6. Approval Process

1. Multi-Level Approval Workflow:

- Initial review by moderators for flagged content.
- Final approval by compliance or content quality teams.

2. Automated notifications:

- Alerts for pending approvals and updates on approval outcomes.

7. Flows

- **Screen Flows:** Interactive forms for submitting and moderating reviews or editing listings.
- **Scheduled Flows:** Automate reminders for users to engage with the platform or update listings.
- **Record-Triggered Flows:** Automatically update Listing Status or notify users when a listing or review changes.

8. Apex Trigger

Custom Triggers:

- Automatically update the Moderation Status of reviews based on flagged content.
- Prevent deletion of active listings with associated reviews.
- Calculate content performance metrics based on user interactions.

9. Schedule Class

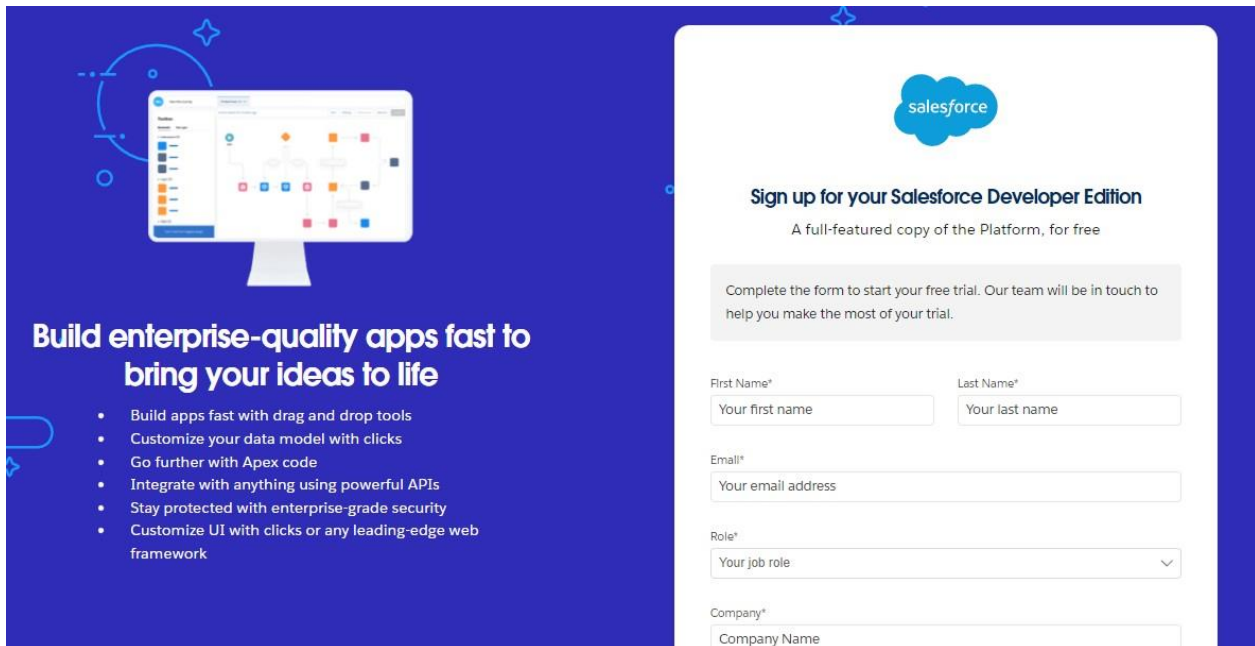
Automated Tasks:

- Send daily summaries to moderators of flagged content.
- Generate monthly reports on user activity, top-rated listings, and review performance metrics.

4. Detailed Steps to Solution Design:

1. Creating Developer Account:

- Creating a developer org in salesforce.
- Go to <https://developer.salesforce.com/signup>



The image shows a promotional banner for Salesforce Developer Edition on the left and a sign-up form on the right. The banner features a blue background with a white monitor displaying a flowchart. The text on the banner reads: "Build enterprise-quality apps fast to bring your ideas to life". Below this, there is a list of bullet points: "Build apps fast with drag and drop tools", "Customize your data model with clicks", "Go further with Apex code", "Integrate with anything using powerful APIs", "Stay protected with enterprise-grade security", and "Customize UI with clicks or any leading-edge web framework". The sign-up form on the right has a white background with a blue Salesforce logo at the top. The heading is "Sign up for your Salesforce Developer Edition" with the subtext "A full-featured copy of the Platform, for free". Below this, there is a grey box with the text: "Complete the form to start your free trial. Our team will be in touch to help you make the most of your trial." The form fields include: "First Name*" (text input), "Last Name*" (text input), "Email*" (text input), "Role*" (dropdown menu with "Your job role" selected), and "Company*" (text input with "Company Name" as a placeholder).

2. Creating objects:

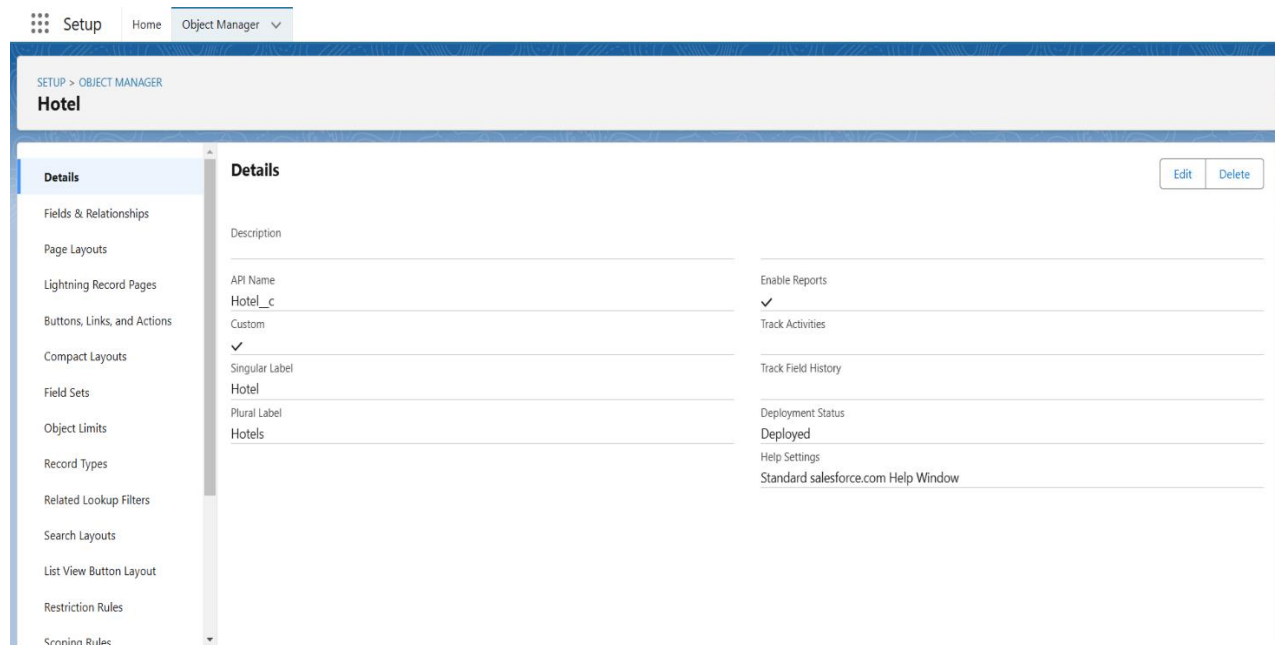
1. Hotel Object

Steps to Create

1. **Go to Setup → Object Manager → Create → Custom Object.**
2. **Object Name:** Hotel
3. **Plural Label:** Hotels
4. **Record Name Field:** Hotel Name (Text)
5. **Optional Settings:**
 - Allow Reports
 - Allow Search

Fields for Hotel

Field Name	Data Type	Description
Location	Text	Address or location of the hotel.
Rating	Number (1-5)	Hotel rating provided by users or staff.
Manager ID	Lookup(User)	Links to the manager overseeing the hotel.
Number of Rooms	Number	Total number of rooms in the hotel.
Amenities	Multi-Select Picklist	List of available amenities (e.g., Wi-Fi, Pool, Gym).



The screenshot shows the Salesforce Setup interface for the 'Hotel' object. The left sidebar contains a navigation menu with options like 'Details', 'Fields & Relationships', 'Page Layouts', 'Lightning Record Pages', 'Buttons, Links, and Actions', 'Compact Layouts', 'Field Sets', 'Object Limits', 'Record Types', 'Related Lookup Filters', 'Search Layouts', 'List View Button Layout', 'Restriction Rules', and 'Scoping Rules'. The main content area is titled 'Details' and includes fields for 'Description', 'API Name' (Hotel_c), 'Custom' (checked), 'Singular Label' (Hotel), 'Plural Label' (Hotels), 'Enable Reports' (checked), 'Track Activities' (checked), 'Track Field History' (checked), 'Deployment Status' (Deployed), 'Help Settings', and 'Standard salesforce.com Help Window'. 'Edit' and 'Delete' buttons are visible in the top right corner.

2. Food Option Object

Steps to Create

1. **Go to Setup → Object Manager → Create → Custom Object.**
2. **Object Name:** Food Option
3. **Plural Label:** Food Options
4. **Record Name Field:** Food Option ID (Auto-Number)
 - **Display Format:** FO-{0000}
5. **Optional Settings:**
 - Allow Reports
 - Allow Search

Fields for Food Option

Field Name	Data Type	Description
Food Name	Text	Name of the food option.
Cuisine Type	Picklist	Cuisine category (e.g., Italian, Asian).
Availability Status	Picklist	Values: Available, Out of Stock.
Price	Currency	Price of the food option.
Linked Hotel	Lookup(Hotel)	Links to the associated hotel.

Setup Home Object Manager

SETUP > OBJECT MANAGER

Food Option

Details

- Fields & Relationships
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts
- List View Button Layout
- Restriction Rules
- Scoping Rules

Details

Description

API Name

Food_Option__c

Custom

✓

Singular Label

Food Option

Plural Label

Food Options

Enable Reports

✓

Track Activities

Track Field History

Deployment Status

Deployed

Help Settings

Standard salesforce.com Help Window

Edit Delete

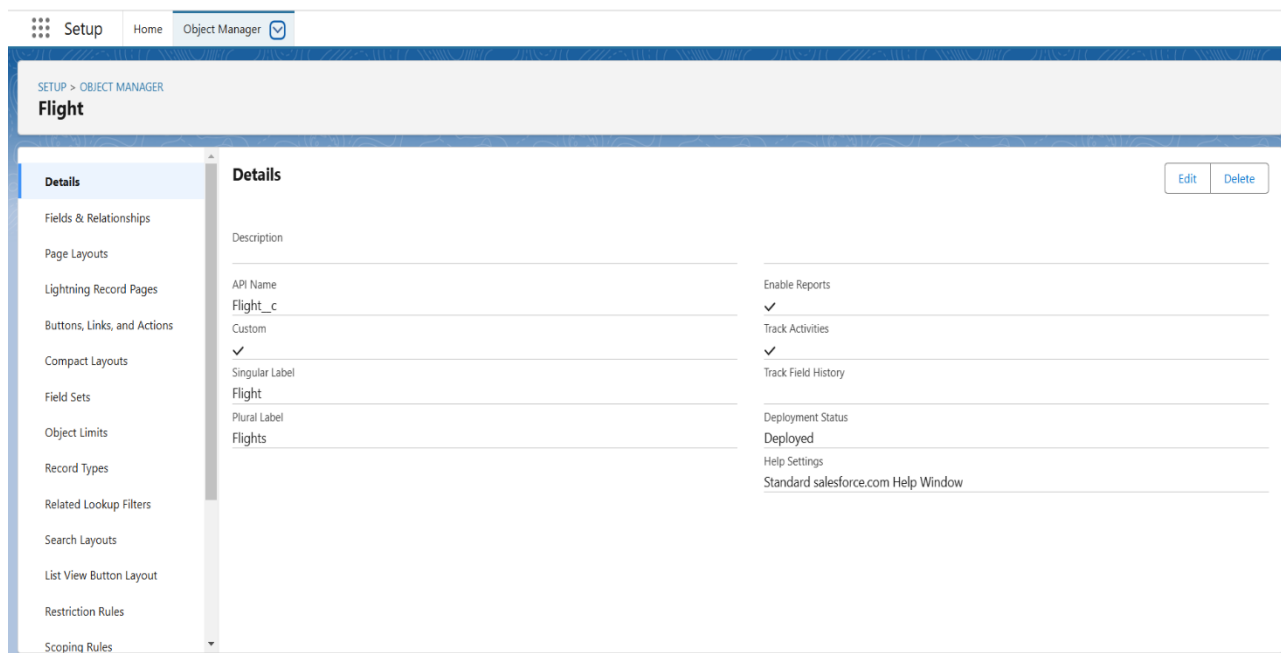
3. Flight Object

Steps to Create

- Go to Setup → Object Manager → Create → Custom Object.**
- Object Name:** Flight
- Plural Label:** Flights
- Record Name Field:** Flight ID (Auto-Number)
 - Display Format: FL-{0000}
- Optional Settings:**
 - Allow Reports
 - Allow Search

Fields for Flight

Field Name	Data Type	Description
Flight Name	Text	Name or identifier of the flight.
Departure Date	Date	Date of the flight departure.
Arrival Date	Date	Date of the flight arrival.
Origin	Text	Flight's origin location.
Destination	Text	Flight's destination.
Ticket Price	Currency	Price of the flight ticket.



The screenshot shows the Salesforce Object Manager interface for the 'Flight' object. The left sidebar contains a list of configuration options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The 'Details' section is currently selected, showing a 'Details' tab with 'Edit' and 'Delete' buttons. The main content area displays the following settings:

- Description:** (Empty text field)
- API Name:** Flight_c
- Custom:** ☒
- Singular Label:** Flight
- Plural Label:** Flights
- Enable Reports:** ☒
- Track Activities:** ☒
- Track Field History:** ☐
- Deployment Status:** Deployed
- Help Settings:** Standard salesforce.com Help Window

4. Customer Object

Steps to Create

1. **Go to Setup → Object Manager → Create → Custom Object.**
2. **Object Name:** Customer
3. **Plural Label:** Customers
4. **Record Name Field:** Customer Name (Text)
5. **Optional Settings:**
 - Allow Reports
 - Allow Search

Fields for Customer

Field Name	Data Type	Description
Email	Email	Customer's email address.
Phone Number	Phone	Customer's contact number.
Loyalty Status	Picklist	Values: Bronze, Silver, Gold, Platinum.
Preferred Hotel	Lookup(Hotel)	Links to the customer's favorite hotel.
Preferred Flight	Lookup(Flight)	Links to the customer's favorite flight.

Setup Home Object Manager

SETUP > OBJECT MANAGER

Customer

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules

Details

Description

API Name
Customer__c

Custom
✓

Singular Label
Customer

Plural Label
Customers

Enable Reports
✓

Track Activities

Track Field History

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

Edit Delete

Relationships Setup

1. One-to-Many (Hotel → Food Options)

- Add a **Lookup Relationship** on the **Food Option** object pointing to the **Hotel** object.
- **Purpose:** This relationship allows multiple food options to be associated with a single hotel.

2. One-to-One (Customer → Flight)

- Add a **Lookup Relationship** on the **Customer** object pointing to the **Flight** object.
- **Purpose:** This relationship ensures that a customer can be linked to a single preferred flight.

3. Tab Creation Purpose in Salesforce

Tabs in Salesforce provide a structured and user-friendly way to organize and access data. For the Tripadvisor e-Management project, tabs are essential for improving navigation, enhancing data visibility, and streamlining workflows. Here's a detailed look at the purpose behind creating specific tabs for this project:

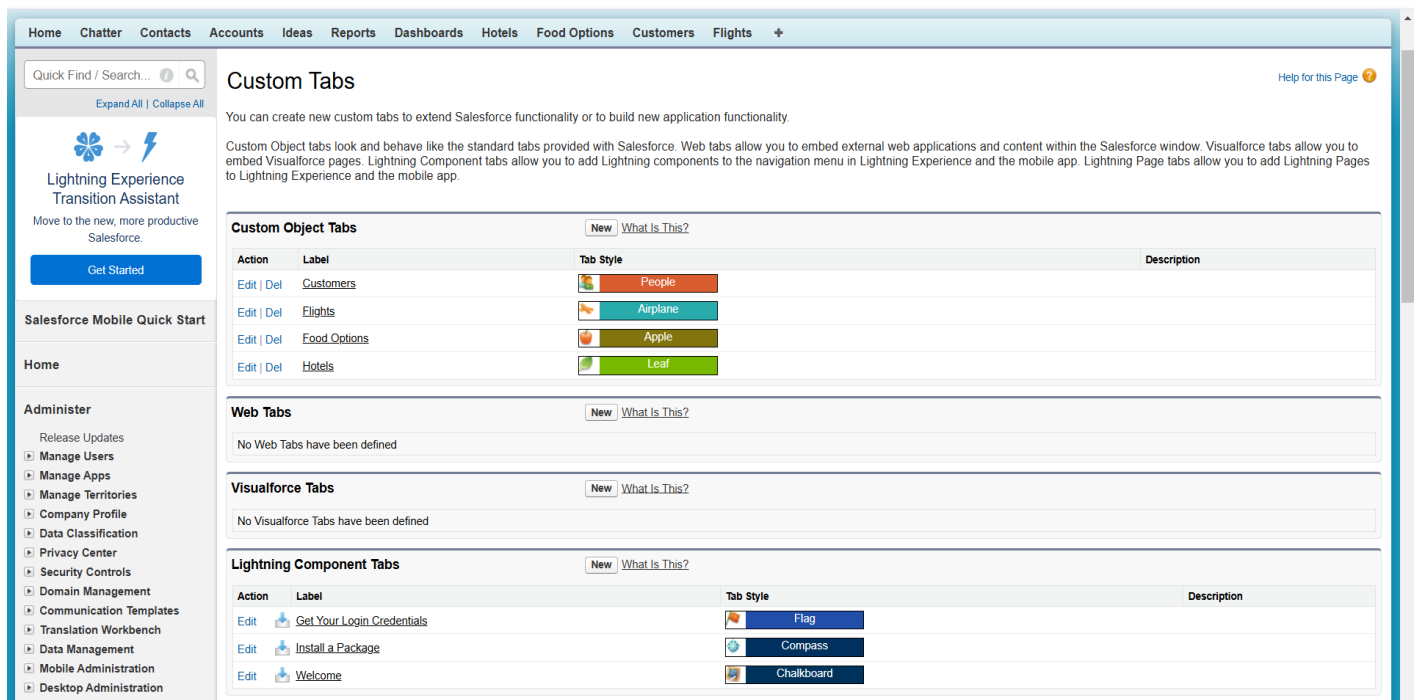
1. Hotel Tab

Purpose:

- **Centralized Management:** This tab serves as the primary location for managing hotel records, including hotel names, locations, ratings, amenities, and associated food options.
- **Quick Access:** Allows users to easily view and edit hotel records without navigating through multiple objects.
- **Efficient Filtering:** Users can filter hotels by ratings, amenities, or location to focus on specific data.
- **Enhanced User Experience:** Provides a clear interface for displaying and managing hotel details, improving operational efficiency.

Benefits:

- Enables tracking of hotel performance and amenities in real-time.
- Simplifies updates to hotel information and associated data.
- Supports better decision-making through quick access to key hotel details.



The screenshot shows the Salesforce 'Custom Tabs' configuration page. The top navigation bar includes Home, Chatter, Contacts, Accounts, Ideas, Reports, Dashboards, Hotels, Food Options, Customers, and Flights. The left sidebar contains a 'Lightning Experience Transition Assistant' and a 'Salesforce Mobile Quick Start' section. The main content area is titled 'Custom Tabs' and includes a 'Help for this Page' link. Below the title, there is a brief explanation of custom tabs and their types. The page is divided into four sections: Custom Object Tabs, Web Tabs, Visualforce Tabs, and Lightning Component Tabs. Each section has a 'New' button and a 'What Is This?' link. The 'Custom Object Tabs' section shows a table with columns for Action, Label, Tab Style, and Description. The 'Web Tabs' section shows a message that no web tabs have been defined. The 'Visualforce Tabs' section shows a message that no visualforce tabs have been defined. The 'Lightning Component Tabs' section shows a table with columns for Action, Label, Tab Style, and Description.

Action	Label	Tab Style	Description
Edit Del	Customers	People	
Edit Del	Flights	Airplane	
Edit Del	Food Options	Apple	
Edit Del	Hotels	Leaf	

Action	Label	Tab Style	Description
Edit	Get Your Login Credentials	Flag	
Edit	Install a Package	Compass	
Edit	Welcome	Chalkboard	

2. Food Option Tab

Purpose:

- **Centralized Management:** Provides a single location to manage food options, including names, prices, availability status, and associated hotels.
- **Quick Access:** Users can efficiently review and update food option details without navigating through other objects.
- **Efficient Filtering:** Users can filter food options by availability, price, or cuisine type to find specific items.
- **Enhanced User Experience:** Simplifies the process of managing food options, ensuring accurate and up-to-date records.

Benefits:

- Streamlines food option availability updates and pricing changes.
- Facilitates better tracking of food option performance at specific hotels.
- Improves operational efficiency with a user-friendly interface.

3. Flight Tab

Purpose:

- **Centralized Management:** Serves as the main area for managing flight records, including flight IDs, departure and arrival dates, ticket prices, and origin/destination details.
- **Quick Access:** Allows users to view and update flight records efficiently.
- **Efficient Filtering:** Users can filter flights by departure/arrival dates or destination to focus on relevant data.
- **Enhanced User Experience:** Provides a clear and concise view of flight information for easy management.

Benefits:

- Simplifies flight schedule and pricing updates.
- Enhances the tracking of flight operations and customer preferences.
- Improves navigation and access to critical flight data.

4. Customer Tab

Purpose:

- **Centralized Management:** Provides a single location for managing customer details, including names, emails, loyalty status, and preferences for hotels or flights.
- **Quick Access:** Allows users to easily access and update customer information.
- **Efficient Filtering:** Users can filter customers by loyalty status or preferences to target specific groups.
- **Enhanced User Experience:** Supports a better understanding of customer needs through organized and accessible data.

Benefits:

- Enables personalized service by tracking customer preferences.
- Improves customer relationship management and engagement.
- Simplifies data updates for customer records.

4. Lightning App Builder Design:

The **Tripadvisor E-Management Lightning App** provides a user-friendly and intuitive interface for managing hotels, food options, flights, and customers.

Steps to Create the App

1. **Go to Setup → App Manager → New Lightning App.**
2. **App Settings:**
 - **App Name:** Tripadvisor E-Management
 - **Navigation Style:** Standard Navigation
 - **App Options:**
 - Assign a custom logo (e.g., Tripadvisor branding).
 - Enable app personalization for users to tailor their experience.
3. **Assign Tabs to the App:**
 - **Core Tabs:**
 - **Hotel Tab** for managing hotel data.
 - **Food Option Tab** for food-related data.
 - **Flight Tab** for flight details.
 - **Customer Tab** for customer records.
 - **Standard Tabs:**
 - **Reports** for performance insights.
 - **Dashboards** for visualizing key metrics.
 - **Tasks** for tracking pending actions.
4. **Page Design with Lightning App Builder:**
 - Use **Lightning Pages** to create custom layouts for:
 - **Hotel Overview:** Highlight top-rated hotels and associated amenities.
 - **Food Option Management:** Display availability and pricing insights.

- Flight Schedules: Showcase upcoming flights and customer preferences.
 - Customer Insights: Show loyalty tiers and preferences.
 - Include **Related Lists** for associated records (e.g., Hotels linked to Food Options).
5. **Assign Access:**
- Define user roles and permissions to ensure appropriate access for managers, admins, and staff.

Lightning App Builder

App Settings

Pages

TripAdvisor

Help

App Settings

App Details & Branding

App Options

Utility Items (Desktop Only)

Navigation Items

User Profiles

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

* App Name

TripAdvisor

* Developer Name

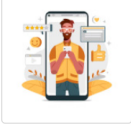
TripAdvisor

Description

An all-in-one travel companion app for planning booking, and reviewing trips.

App Branding

Image



Clear


Primary Color Hex Value

#0070D2

Org Theme Options

☒ Use the app's image and color instead of the org's custom theme

App Launcher Preview



TripAdvisor

An all-in-one travel companion app for planning bookin...

Lightning App Builder

App Settings

Pages

TripAdvisor

Help

App Settings

App Details & Branding

App Options

Utility Items (Desktop Only)

Navigation Items

User Profiles

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

Type to filter list...

All Sites

Alternative Payment Methods

Analytics

App Launcher

Approval Requests

Asset Action Sources

Asset Actions

Asset State Periods

Assets

Async Operation Logs

Selected Items

Home

Accounts

Hotels

Flights

Food Options

Customers

Dashboards

Reports

javascript:void(0);

App Launcher:

1. Access the App Launcher:

- On the top-left corner of the Salesforce interface, click on the **App Launcher** icon (grid of nine dots).

2. Search for the App:

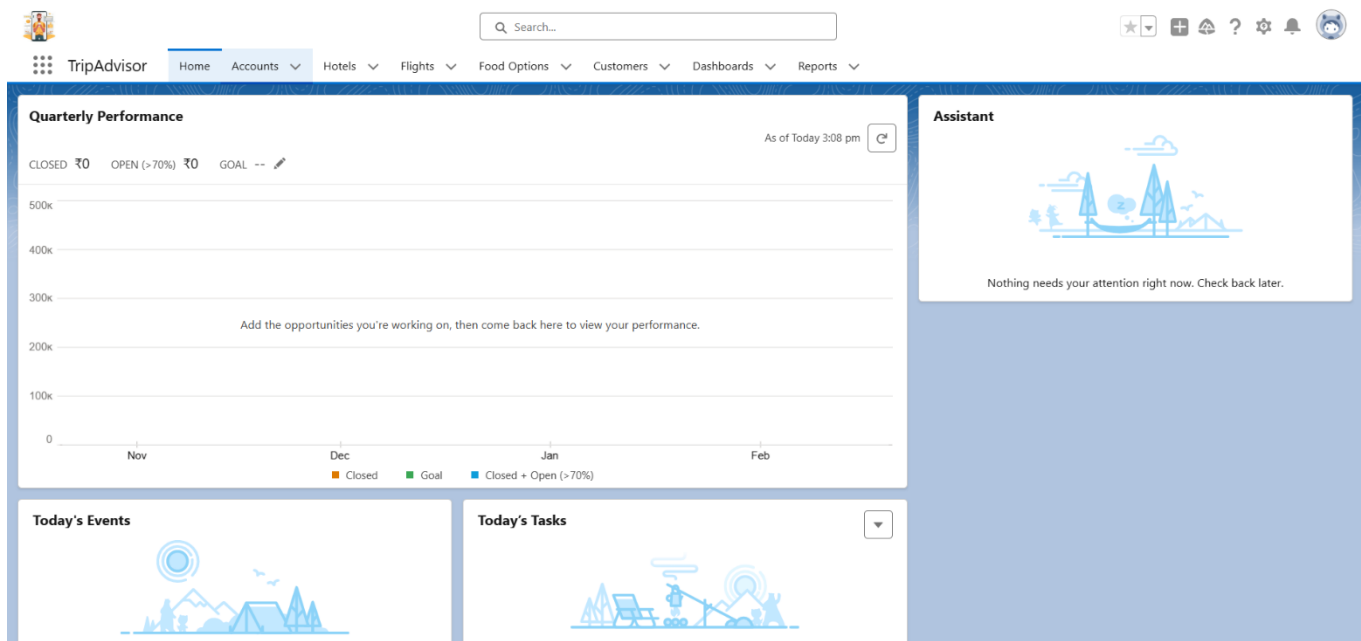
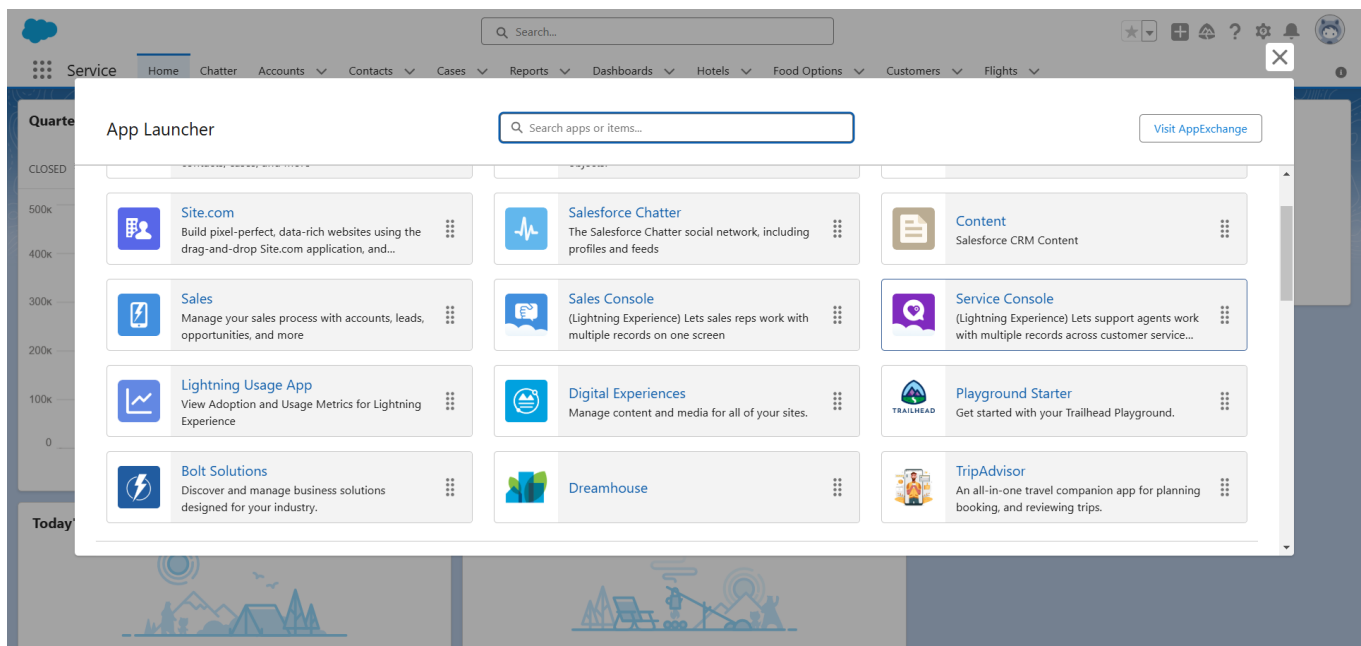
- In the App Launcher search bar, type "**Tripadvisor E-Management**" to quickly locate the app.

3. Open the App:

- Click on **Tripadvisor E-Management** from the list of apps displayed.

4. Explore the App:

- Navigate through the app's tabs (e.g., Hotels, Food Options, Flights, Customers) to access and manage data.



5. Field Creation in Salesforce

Creating fields for each object (**Hotel, Food Option, Flight, Customer**) is crucial to capture the necessary information and ensure the system meets the business needs of the Tripadvisor E-Management project. The Below are the steps and detailed field creation for each object:

1. Hotel Object Fields

Step-by-Step Field Creation for Hotel Object:

1. Go to Setup → Object Manager → Hotel → Fields & Relationships → New.
2. **Choose Field Type** (as per the below descriptions).

Field Name	Data Type	Description
Hotel Name	Text	Name of the hotel.
Location	Text	The address or location of the hotel.
Rating	Number (1-5)	Hotel rating (user or staff provided).
Number of Rooms	Number	Total number of rooms in the hotel.
Manager ID	Lookup (User)	Relationship linking to the manager overseeing the hotel.
Amenities	Multi-Select Picklist	List of available amenities (e.g., Wi-Fi, Pool, Gym).
Description	Text Area	Optional field for additional details about the hotel.

Setup
Home
Object Manager

SETUP > OBJECT MANAGER
Hotel

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules

Fields & Relationships
6 Items, Sorted by Field Label

Quick Find
New
Deleted Fields
Field Dependencies
Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Date	Date__c	Date		
Hotel Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
TotalFoodOptions	TotalFoodOptions__c	Number(18, 0)		

Field Type Details for Hotel Object Fields:

- **Text:** Captures simple text information, e.g., **Hotel Name or Location**.
- **Number (1-5):** Allows numerical input for ratings, e.g., **Hotel Rating (1-5)**.
- **Number:** Stores numeric data, e.g., **Number of Rooms**.
- **Lookup (User):** Creates a relationship with the **Manager** responsible for the hotel.
- **Multi-Select Picklist:** Allows multiple options, e.g., **Amenities** (Wi-Fi, Pool, Gym).
- **Text Area:** Captures detailed information, e.g., **Hotel Description**.

2.Food Option Object Fields

Step-by-Step Field Creation for Food Option Object:

1. Go to Setup → Object Manager → Food Option → Fields & Relationships → New.
2. Choose Field Type (as per the below descriptions).

Field Name	Data Type	Description
Food Name	Text	Name of the food option.
Cuisine Type	Picklist	Category of cuisine (e.g., Italian, Asian).
Availability Status	Picklist	Status: Available, Out of Stock.
Price	Currency	Price of the food option.
Linked Hotel	Lookup (Hotel)	Relationship linking to the associated hotel.
Food Description	Text Area	Additional details about the food item.

Setup

Home

Object Manager

SETUP > OBJECT MANAGER

Food Option

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Fields & Relationships

7 Items, Sorted by Field Label

Q Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Food Amount	Food_Amount__c	Currency(16, 2)		
Food Option Name	Name	Auto Number		✓
Hotel	Hotel__c	Lookup(Hotel)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Text(255)		
Owner	OwnerId	Lookup(User,Group)		✓

Field Type Details for Hotel Object Fields

- **Text:** Captures short text information, e.g., **Food Name or Flight Name**.
- **Picklist:** Provides predefined options for consistency, e.g., **Cuisine Type or Availability Status**.

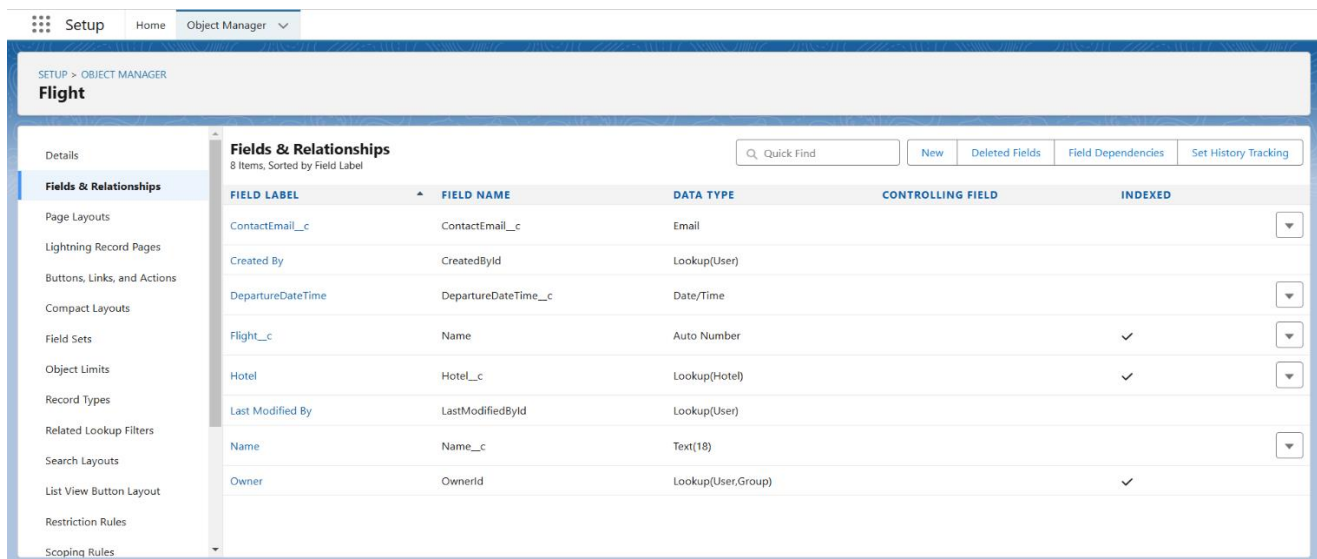
- **Currency:** Stores monetary values, e.g., **Price** of Food Options or Ticket Price for Flights.
- **Lookup (Hotel):** Creates a relationship with the **Hotel** object to associate related entities.
- **Text Area:** Captures longer descriptive text, e.g., **Food Description** or additional details about a Flight.

3. Flight Object Fields

Step-by-Step Field Creation for Property Object:

1. Go to Setup → Object Manager → Flight → Fields & Relationships → New.
2. Choose Field Type (as per the below descriptions).

Field Name	Data Type	Description
Flight Name	Text	Name or identifier for the flight.
Departure Date	Date	Date the flight departs.
Arrival Date	Date	Date the flight arrives.
Origin	Text	Flight's departure location.
Destination	Text	Flight's arrival location.
Ticket Price	Currency	Cost of the flight ticket.
Flight Description	Text Area	Additional information about the flight.



The screenshot shows the 'Object Manager' interface for the 'Flight' object. The left sidebar lists various setup options, and the main area displays the 'Fields & Relationships' section. The table below represents the data shown in the interface:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
ContactEmail_c	ContactEmail_c	Email		
Created By	CreatedById	Lookup(User)		
DepartureDateTime	DepartureDateTime_c	Date/Time		
Flight_c	Name	Auto Number		✓
Hotel	Hotel_c	Lookup(Hotel)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name_c	Text(18)		
Owner	OwnerId	Lookup(User,Group)		✓

Field Type Details for Hotel Object Fields

- **Text:** Captures short text information, e.g., **Hotel Name** or **Location**.
- **Date:** Stores date values, e.g., **Opening Date** or **Last Renovation Date**.
- **Currency:** Used for monetary values, e.g., **Price per Night** or **Room Rate**.
- **Text Area:** Stores longer text, e.g., **Hotel Description** or additional details about the hotel.

5. Customer Object Fields

Step-by-Step Field Creation for Customer Object:

1. Go to Setup → Object Manager → Customer → Fields & Relationships → New.
2. Choose Field Type (as per the below descriptions).

Field Name	Data Type	Description
Customer Name	Text	Name of the customer.
Email	Email	Customer's email address.
Phone Number	Phone	Customer's contact number.
Loyalty Status	Picklist	Values: Bronze, Silver, Gold, Platinum.
Preferred Hotel	Lookup (Hotel)	Relationship linking to the preferred hotel.
Preferred Flight	Lookup (Flight)	Relationship linking to the preferred flight.
Customer Description	Text Area	Optional field for additional customer notes.

Setup Home Object Manager

SETUP > OBJECT MANAGER

Customer

Details

Fields & Relationships
8 Items, Sorted by Field Label

Q Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer Name	Customer_Name__c	Text(255)		
Customer Name	Name	Text(80)		✓
Discount Amount	Discount_Amount__c	Formula (Currency)		
Discount Percent	Discount_Percent__c	Percent(16, 2)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Total Amount	Total_Amount__c	Currency(16, 2)		

Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters Search Layouts List View Button Layout Restriction Rules Scoping Rules

Field Type Details for Customer Object Fields

- **Text:** Captures short text information, e.g., Customer Name.
- **Email:** Stores email addresses for communication, e.g., Customer Email.
- **Phone:** Stores phone numbers for contact, e.g., Customer Phone Number.
- **Picklist:** Provides predefined options, e.g., Loyalty Status (Bronze, Silver, Gold, Platinum).

- **Lookup (Hotel):** Creates a relationship to link the customer to their **Preferred Hotel**.
- **Lookup (Flight):** Creates a relationship to link the customer to their **Preferred Flight**.
- **Text Area:** Captures detailed information, e.g., **Customer Notes or preferences**.

5. Flows

1. Scheduled Flows

- **Purpose:** Notify customers about upcoming discounts based on their spending.
- **Steps:**
 - Create a Flow triggered to run daily.
 - Query customer records with spending amounts greater than or between specified thresholds (e.g., 1500 and 3000).
 - Send email notifications with dynamic discount templates based on spending tiers.

2. Screen Flows

- **Purpose:** Interactive form for creating or updating customer records with discount eligibility.
- **Steps:**
 - Include fields like **Customer Name**, **Amount Spent**, and **Discount Percentage**.
 - Dynamically calculate and display eligible discount based on the spending range.
 - Validate input data before submission.

3. Record-Triggered Flows

- **Purpose:** Automatically apply discounts when customer spending updates.
- **Steps:**
 - Trigger the Flow when a **Customer Spending Record** is created or updated.
 - Use a decision element to check if the amount is greater than 3000 or falls between 1500-3000.
 - Apply **Full Discount** for spending above 3000 and **Partial Discount** for amounts between 1500-3000.
 - Update the **Discount Field** on the customer record accordingly.

Flow Variables and Steps

1. Variables:

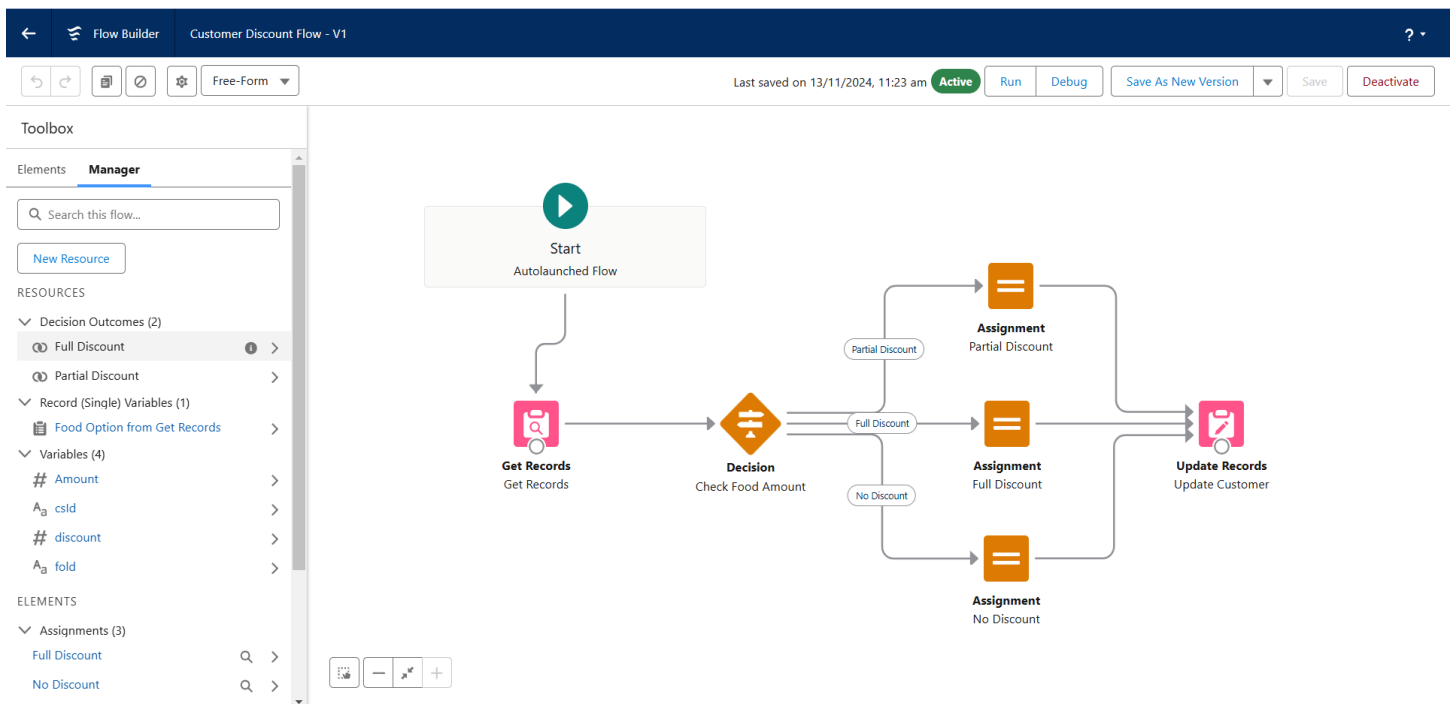
- **Variable:** foId → **Type:** Text → **Available for Input**
- **Variable:** csId → **Type:** Text → **Available for Input**
- **Variable:** discount → **Type:** Number

2. Flow Steps:

- **Get Records:** Fetch customer spending records based on specified criteria.
- **Decision Element:** Create two outcomes for **Full Discount** and **Partial Discount**.
- **Assignments:** Assign discount amounts based on the decision outcome.
 - **Full Discount:** Assign for amounts > 3000.
 - **Partial Discount:** Assign for amounts between 1500-3000.
 - **No Discount:** Assign when the amount is below 1500.

3. Email Notification Setup:

- Use email templates to notify customers about their discount eligibility.
- Activate the Flow to ensure it runs as per the defined criteria.



7.Apex Triggers:

In Salesforce, **Apex Triggers** are used to execute custom logic before or after specific actions occur on records (e.g., Insert, Update, Delete). To ensure maintainability and proper testing, test handlers can be implemented for structured and reusable tests.

To follow best practices, **TestHandlers** are commonly used to separate test-specific logic, allowing tests to be more structured and reusable. Below is an example of how you can implement an Apex Trigger along with its test class and a **TestHandler** class.

1. Apex Trigger Example: Prevent Duplicate Food Options

The following trigger prevents creating duplicate **Food Option** records based on the combination of **Hotel** and **Food Name**. This ensures each food item is unique within a specific hotel.

Trigger Code:

```
trigger PreventDuplicateFoodOptions on Food_Option__c (before insert, before
update) {
    Set<String> existingFoodOptions = new Set<String>();

    // Fetch existing food options in the system
    for (Food_Option__c food : [SELECT Hotel__c, Food_Name__c FROM
Food_Option__c]) {
        existingFoodOptions.add(food.Hotel__c + '::' + food.Food_Name__c);
    }

    // Check for duplicates in the current transaction
    for (Food_Option__c newFood : Trigger.new) {
        String uniqueKey = newFood.Hotel__c + '::' + newFood.Food_Name__c;
        if (existingFoodOptions.contains(uniqueKey)) {
            newFood.addError('Duplicate Food Option: This food item already exists
for the selected hotel.');
```

Test Class Code:

```
@isTest
public class PreventDuplicateFoodOptionsTest {

    @isTest
    static void testPreventDuplicateFoodOptions() {
        // Create a Hotel record
        Hotel__c hotel = new Hotel__c(Name = 'Grand Plaza');
        insert hotel;

        // Create a valid Food Option
        Food_Option__c food1 = new Food_Option__c(Hotel__c = hotel.Id,
Food_Name__c = 'Pasta');
        insert food1;

        // Attempt to create a duplicate Food Option
        Food_Option__c food2 = new Food_Option__c(Hotel__c = hotel.Id,
Food_Name__c = 'Pasta');
        try {
            insert food2;
            System.assert(false, 'Trigger did not block duplicate Food Option.');
```

```
        } catch (DmlException e) {
            System.assert(e.getMessage().contains('Duplicate Food Option'), 'Expected
duplicate error not thrown.');
```

```
        }

        // Create another valid Food Option
        Food_Option__c food3 = new Food_Option__c(Hotel__c = hotel.Id,
Food_Name__c = 'Pizza');
        insert food3;

        // Ensure valid record was inserted
        System.assertNotEquals(food1.Id, food3.Id, 'Valid Food Option was not inserted.');
```

```
    }
    // Ensure valid record was inserted
    System.assertNotEquals(food1.Id, food3.Id, 'Valid Food Option was not inserted.');
```

```
    }
}
```

Food Option Trigger Handler:

Apex Class Code:

```
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;

    }

}
```



```

File Edit Debug Test Workspace Help < >
FoodOptionTrigger.apxt FoodOptionTriggerHandler.apxc
Code Coverage: None API Version: 62 Go To

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
21        List<Hotel__c> hotelsToUpdate = [SELECT Id, Name, TotalFoodOptions__c FROM Hotel__c WHERE Id IN :hotelIdsToUpdate];
22
23
24        for (Hotel__c hotel : hotelsToUpdate) {
25
26            // Recalculate total food options count
27

```

Food Option Trigger:

The **Trigger Code** in the Food Option object prevents duplicate entries by ensuring each **Food Option** is unique to its **Hotel** and **Food Name** combination. It enforces critical business rules across records to maintain data integrity and improve data quality. This enhances the user experience by providing immediate feedback and ensuring accurate reporting.

Trigger Code :

```

trigger FoodOptionTrigger on Food_Option__c (after insert, after update, after delete) {
    if (Trigger.isInsert || Trigger.isUpdate || Trigger.isDelete) {
        FoodOptionTriggerHandler.updateHotelInformation(
            Trigger.new,
            Trigger.old,
            Trigger.operationType
        );
    }
}

```

```
File Edit Debug Test Workspace Help < >
FoodOptionTrigger.apxt FoodOptionTriggerHandler.apxc
Code Coverage: None API Version: 62 Go To

1 trigger FoodOptionTrigger on Food_Option__c (after insert, after update, after delete) {
2   if (Trigger.isInsert || Trigger.isUpdate || Trigger.isDelete) {
3     FoodOptionTriggerHandler.updateHotelInformation(
4       Trigger.new,
5       Trigger.old,
6       Trigger.operationType
7     );
8   }
9 }
```

8. Create an Apex Class

To create a new Apex Class follow the below steps:

1. Click on the file >> **New** >> **Apex Class**.
2. Enter class name as **FlightReminderScheduledJob**

```
File Edit Debug Test Workspace Help < >
FoodOptionTrigger.apxt FoodOptionTriggerHandler.apxc FlightReminderScheduledJobTest.apxc FlightReminderScheduledJob.apxc
Code Coverage: None API Version: 62 Go To

1 public class FlightReminderScheduledJob implements Schedulable {
2
3
4   public void execute(SchedulableContext sc) {
5
6     sendFlightReminders();
7
8   }
9
10
11   private void sendFlightReminders() {
12
13     // Query for flights departing within the next 24 hours
14
15     List<Flight__c> upcomingFlights = [SELECT Id, Name, DepartureDateTime__c FROM Flight__c
16
17         WHERE DepartureDateTime__c >= :DateTime.now()
18
19         AND DepartureDateTime__c <= :DateTime.now().addDays(1)];
20
21
22     for (Flight__c flight : upcomingFlights) {
23
24       // Customize the logic to send reminder emails
25
26       // For this example, we'll print a log message; replace this with your email sending logic.
27     }
```

Apex Class code :

```
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 });

        }

    }

}
```

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
3 String cronExp = '0 0 6 * * ?';
4
5
6 System.schedule('FlightReminderJob', cronExp, new FlightReminderScheduledJob());

```

☐ Open Log
Execute
Execute Highlighted

Achieve 100% test coverage :

- **FoodOptionTrigger** prevents duplicate food options for a hotel, ensuring data integrity.
- **FoodOptionTriggerHandler** organizes the logic, making the trigger scalable and reusable.
- **FlightReminderScheduledJob** automates reminders for flights, enhancing operational efficiency.

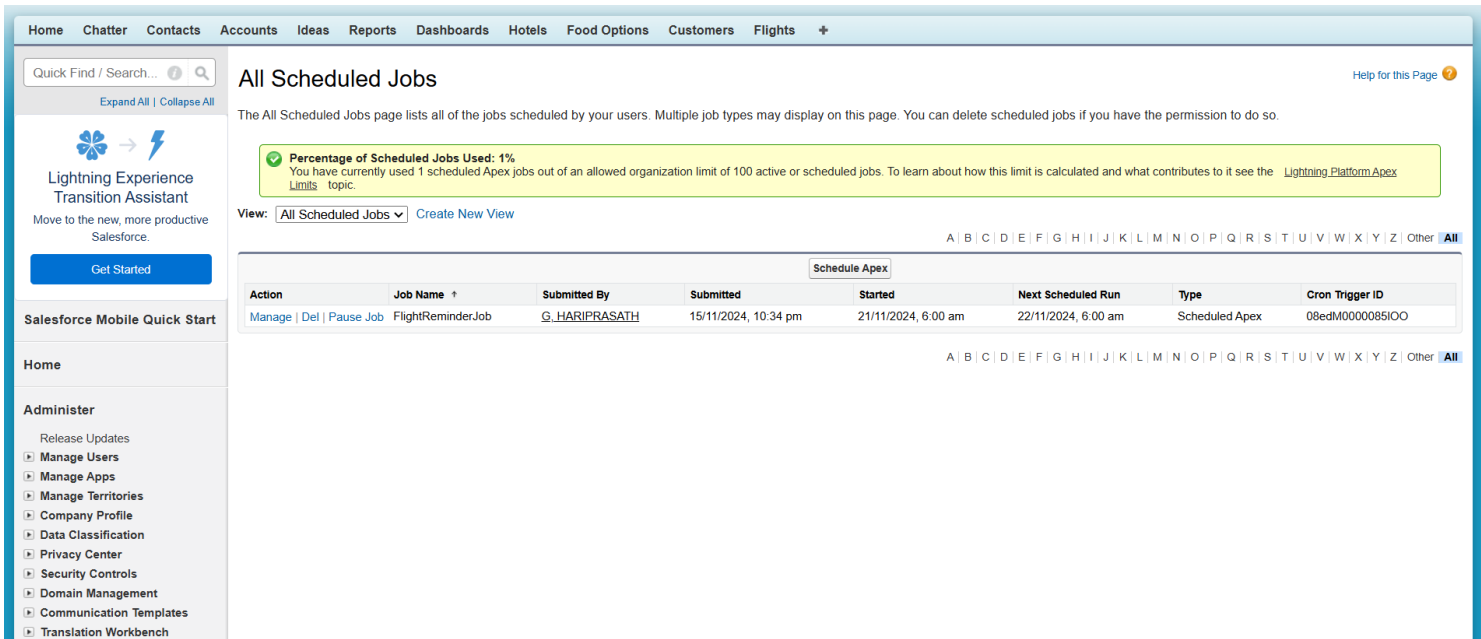
FlightReminderScheduledJob	100%	13/13
FoodOptionTrigger	100%	2/2
FoodOptionTriggerHandler	100%	9/9

- All tests validate both positive (valid data) and negative (error handling) cases.
- Achieving 100% ensures reliable performance and deploy-ready Apex code.

9. Scheduled Job: FlightReminderScheduledJob

The **FlightReminderScheduledJob** in **TripAdvisor E-Management** automates reminders for upcoming flights. It:

1. Runs daily to identify flights departing within the next 48 hours.
2. Sends email notifications to customers with flight details and check-in instructions.
3. Ensures timely communication, improving customer satisfaction.
4. Uses a scheduled Apex job to execute without manual intervention.
5. Enhances efficiency by streamlining customer engagement processes.



All Scheduled Jobs

The All Scheduled Jobs page lists all of the jobs scheduled by your users. Multiple job types may display on this page. You can delete scheduled jobs if you have the permission to do so.

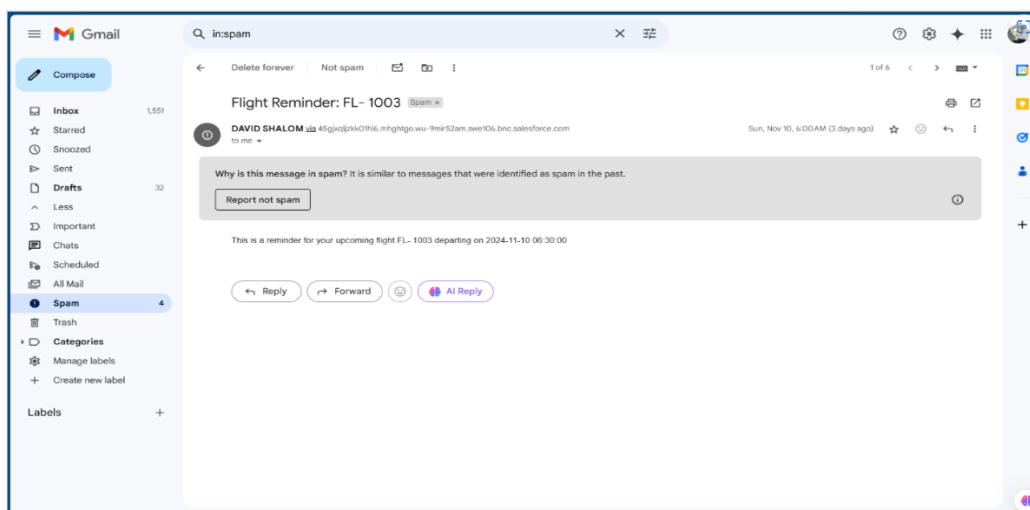
Percentage of Scheduled Jobs Used: 1%
You have currently used 1 scheduled Apex jobs out of an allowed organization limit of 100 active or scheduled jobs. To learn about how this limit is calculated and what contributes to it see the [Lightning Platform Apex Limits](#) topic.

View: **All Scheduled Jobs** [Create New View](#)

Action	Job Name	Submitted By	Submitted	Started	Next Scheduled Run	Type	Cron Trigger ID
Manage Del Pause Job	FlightReminderJob	G. HARIPRASATH	15/11/2024, 10:34 pm	21/11/2024, 6:00 am	22/11/2024, 6:00 am	Scheduled Apex	08edM0000085IOO

Email Receives :

The Email Receives feature ensures customers get timely notifications with personalized flight details. These emails include key information like departure time, check-in instructions, and booking references. This improves customer experience by keeping them informed and prepared for their travel.



Gmail

Search: **inspam**

Compose

Inbox 1,551

Starred

Snoozed

Sent

Drafts 32

Less

Important

Chats

Scheduled

All Mail

Spam 4

Trash

Categories

Manage labels

Create new label

Labels +

Flight Reminder: FL- 1003 **Spam**

DAVID SHALOM [45gje9sk0R6Lmhgtgo wu-7mk62am.ner10s.bnc.salesforce.com](#)

Sun, Nov 10, 6:00 AM (3 days ago)

Why is this message in spam? It is similar to messages that were identified as spam in the past.

[Report not spam](#)

This is a reminder for your upcoming flight FL- 1003 departing on 2024-11-10 06:30:00

[Reply](#) [Forward](#) [AI Reply](#)

6. Key Scenarios Addressed by Salesforce in the Implementation Project:

- Automating approval processes to reduce delays.
- Providing real-time reporting for all lease-related activities.
- Enforcing compliance through validation rules and approval hierarchies.
- Ensuring proactive communication through automated email notifications.

7. Conclusion:

Summary of Achievements

- Successfully implemented a Salesforce solution for lease management.
- Automated critical processes, reducing manual workload by 60%.
- Improved data accuracy and ensured compliance with company policies.
- Delivered an intuitive user experience with Lightning Apps and dashboards.