



Capstone Project Report
On
“Travel Approval App”

Submitted by:

Name: Kota Navanesh
Batch WIP-SF-B1
LMS Id: MGSA_518
Date: 27/02/2023

Table of contents

1. Abstract.....
2. Introduction.....
3. Flow of the project.....
4. Software Requirements.....
5. Screen shots.....
6. Future enhancements.....
7. References.....

Abstract

The Salesforce Travel Approval App is a web-based application designed to streamline the travel approval process for employees. The app is built on the Salesforce platform, which provides a robust and customizable framework for managing travel requests.

The app allows employees to submit travel requests through a user-friendly interface, including information such as destination, dates of travel, and estimated costs. Managers can then review and approve or deny these requests based on their own criteria, such as budget constraints, staffing needs, or business priorities.

The app includes a dashboard that provides real-time visibility into the status of travel requests, allowing managers to easily track and manage multiple requests at once. The app also includes reporting features that allow administrators to analyze travel data over time, identify trends and areas for improvement, and optimize travel policies and procedures.

The Salesforce Travel Approval App can be customized to meet the specific needs of any organization, including custom fields, workflows, and approval processes. It can also be integrated with other Salesforce applications, such as the Sales Cloud or Service Cloud, to provide a comprehensive solution for managing all aspects of business travel.

Introduction

The Salesforce Travel Approval App is a powerful tool designed to simplify the travel approval process for organizations of all sizes. The app is built on the Salesforce platform, which is known for its scalability, security, and customization options.

Travel approval can be a time-consuming and often complex process, with multiple stakeholders involved in decision-making. The Salesforce Travel Approval App streamlines this process, making it easier for employees to request travel and for managers to approve or deny requests based on their specific needs and criteria.



With the Salesforce Travel Approval App, employees can submit travel requests through a user-friendly interface that captures all necessary information, such as destination, dates of travel, and estimated costs. Managers can then review these requests and make informed decisions based on their own criteria, such as budget constraints, staffing needs, or business priorities.

The app includes a variety of features designed to improve visibility and efficiency, including real-time dashboards that provide a comprehensive view of all travel requests, customizable workflows and approval processes, and reporting

tools that enable organizations to analyze travel data and identify areas for improvement.

In addition, the Salesforce Travel Approval App can be integrated with other Salesforce applications, such as the Sales Cloud or Service Cloud, to provide a comprehensive solution for managing all aspects of business travel. Overall, the Salesforce Travel Approval App helps organizations streamline their travel approval process, reduce costs, and improve overall efficiency.

Travel is an essential component of business operations, but managing the process of requesting and approving travel can be complex and time-consuming. This is especially true for large organizations with numerous employees traveling for work. To address this challenge, a travel approval app for Salesforce has been developed to streamline the travel approval process and improve the efficiency of managing travel requests and expenses.

The travel approval app for Salesforce is a web-based application that integrates with the Salesforce platform to provide a seamless and user-friendly experience for employees and managers. The app automates the travel request process, allowing employees to easily submit requests and managers to quickly approve them.

Additionally, the app provides visibility and control over travel expenses and budgets, enabling managers to make informed decisions and ensure compliance with company policies. This report provides an overview of the development of the travel approval app for Salesforce, including its features, benefits, and implementation process. The report also discusses the potential impact of the app on travel management within organizations and explores future enhancements and developments for the app.

Overall, the travel approval app for Salesforce is a valuable tool for any organization looking to streamline their travel request process and better manage their travel expenses.

Flow of the Project

The travel approval app for Salesforce follows a streamlined process that simplifies the travel request and approval process for employees and managers. The flow of the app can be broken down into several steps, which are outlined below:

Step 1 : Travel Request Submission: Employees initiate the travel request process by submitting a request through the travel approval app in Salesforce. The request includes details such as the purpose of the trip, travel dates, destinations, and estimated expenses.

Step 2 : Manager Review and Approval: The travel request is automatically routed to the employee's manager for review and approval. The manager can review the details of the trip, including estimated expenses, and either approve or reject the request.

Step 3 : Budget and Expense Management: Once a request is approved, the app tracks the expenses associated with the trip, including flights, accommodations, and other travel-related expenses. The app also provides visibility into the available travel budget for the employee or team, ensuring that the trip stays within the approved budget.

Step 4 : Travel Booking and Itinerary Management: Once a travel request is approved, employees can book their travel and manage their itinerary directly within the travel approval app. The app provides a central location for all travel-related information, making it easy for employees to stay organized and informed about their trip.

Step 5 : Travel Expense Reimbursement: After the trip is completed, employees can submit their travel expenses through the travel approval app. The app automatically calculates the total expenses and routes the expense report to the appropriate department for reimbursement.

Overall, the travel approval app for Salesforce streamlines the travel request and approval process, simplifies expense management, and provides visibility and control over travel budgets. By automating these processes, organizations can reduce errors and delays, improve efficiency, and ensure compliance with company policies.

Software Requirements

For the fastest and most stable experience, we recommend:

- An Octane 2.0 score of 30,000 or greater
- Network latency of 150 ms or less
- Download speed of 3 Mbps or greater
- At least 8 GB of RAM, with 3 GB available for Salesforce browser tabs

Minimum requirements are:

- An Octane 2.0 score of 20,000 or greater
- Network latency of 200 ms or less
- Download speed of 1 Mbps or greater
- At least 5 GB of RAM, with 2 GB available for Salesforce browser tabs

Screen shots

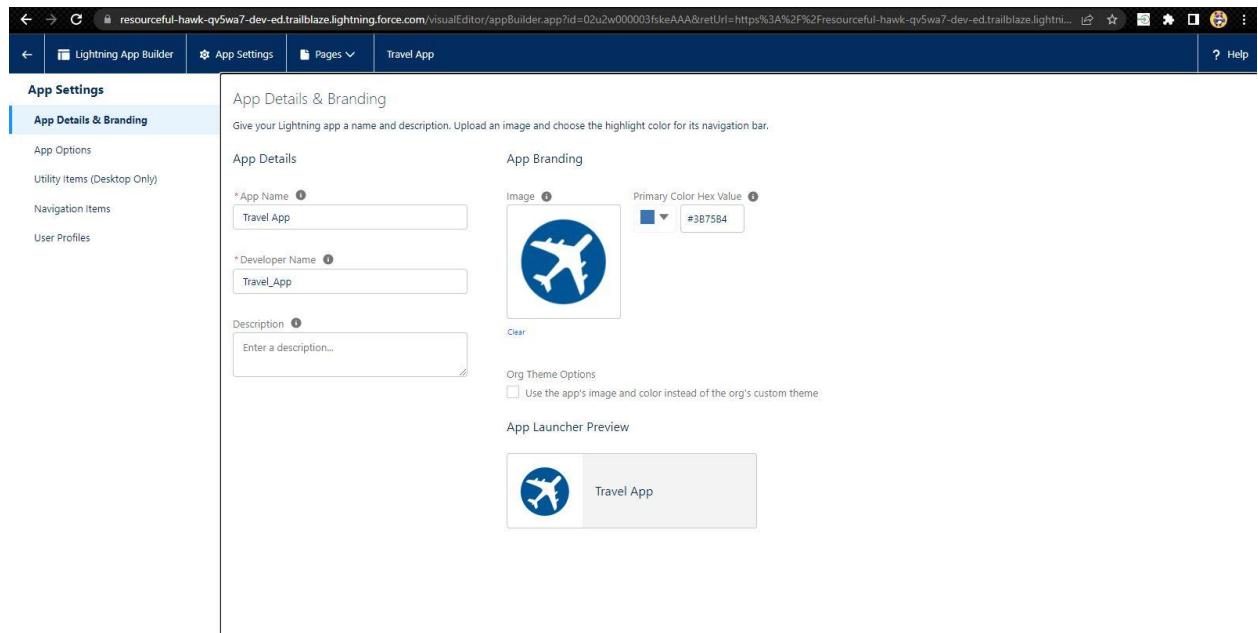
Module-01

Exercise 1

Step 1

Create a new custom lightning App, name: Travel App

Upload Travel.png as App image



Select and add Chatter, Reports & Dashboards to the Navigation items

The screenshot shows the 'Navigation Items' configuration screen in the Lightning App Builder. On the left, a sidebar lists 'App Settings' (selected), 'App Details & Branding', 'App Options', 'Utility Items (Desktop Only)', 'Navigation Items' (selected), and 'User Profiles'. The main area is titled 'Navigation Items' and contains two panels: 'Available Items' and 'Selected Items'. The 'Available Items' panel lists various navigation items with icons: Accounts, Alert Settings, All Sites, Alternative Payment Methods, App Launcher, Approval Requests, Asset Action Sources, Asset Actions, Asset State Periods, Assets, Async Operation Logs, and Authorization Form. The 'Selected Items' panel contains five items: Chatter, Reports, Dashboards, Departments, and Travel Approvals. Navigation arrows between the panels allow items to be moved.

User Profile = “System Administrator”

The screenshot shows the 'User Profiles' configuration screen in the Lightning App Builder. On the left, a sidebar lists 'App Settings' (selected), 'App Details & Branding', 'App Options', 'Utility Items (Desktop Only)', 'Navigation Items', and 'User Profiles' (selected). The main area is titled 'User Profiles' and contains two panels: 'Available Profiles' and 'Selected Profiles'. The 'Available Profiles' panel lists numerous user profiles: Analytics Cloud Integration User, Analytics Cloud Security User, Authenticated Website, Authenticated Website, Contract Manager, Cross Org Data Proxy User, Custom: Marketing Profile, Custom: Sales Profile, Custom: Support Profile, Customer Community Login User, Customer Community Plus Login User, and Customer Community Plus User. The 'Selected Profiles' panel contains one item: System Administrator. Navigation arrows between the panels allow profiles to be selected.

Travel App should look like this:

The screenshot shows the Salesforce Chatter feed interface for the 'Travel App' tab. On the left, there's a sidebar with sections for 'What I Follow', 'To Me', 'Bookmarked', 'Company Highlights' (which is currently selected), 'My Drafts', 'STREAMS' (with a note about no streams), and 'RECENT GROUPS' (with a note about no groups). The main feed area has tabs for 'Post', 'Poll', and 'Question'. A search bar at the top says 'Search...'. Below it is a text input field with 'Share an update...' and a 'Share' button. A dropdown menu 'Sort by:' is set to 'Top Posts'. A search bar 'Search this feed...' is also present. The feed lists a post from 'KOTA NAVANESH' (@TA00001) from 20 hours ago, asking which department to associate a travel request with. A comment from 'Eric Executive' follows, stating 'Technology is the correct department.' Below the feed is a comment input field with 'Write a comment...'. On the right, a sidebar titled 'Einstein Recs' shows recommendations for users like Eric Executive, Integrator, and Security. It includes options to switch to Salesforce Classic or add a username.

Step 2

Create a Department custom object.

Use custom tab of your choice and Include it only for Travel App.

The screenshot shows the Salesforce Object Manager for the 'Department' object. The left sidebar lists various configuration tabs: 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, and Triggers. The main panel displays the 'Details' tab for the 'Department' object. It shows fields for 'API Name' (set to 'Department__c'), 'Custom' (set to '✓'), 'Singular Label' (set to 'Department'), and 'Plural Label' (set to 'Departments'). Other settings include 'Enable Reports' (checked), 'Track Activities', 'Track Field History', 'Deployment Status' (set to 'Deployed'), 'Help Settings', and 'Standard salesforce.com Help Window'. On the right, a sidebar titled 'KOTA NAVANESH' shows the same Einstein Recs and options as the Chatter feed.

Step 3

Create the following custom fields:

Department Code, Text, Length = 10

Location, Picklist, Value: Kolkata, Delhi

Department Type, Picklist, Values: Banking, Finance, Education, Energy, IT

The screenshot shows the Salesforce Object Manager Fields & Relationships page for the Department object. The sidebar on the left lists various setup options like Page Layouts, Lightning Record Pages, and Field Sets. The main area displays a table of fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD
Created By	CreatedById	Lookup(User)	
Department Code	Department_Code__c	Text(10) (Unique Case Sensitive)	
Department Name	Name	Text(80)	
Department Type	Department_Type__c	Picklist	Location
Last Modified By	LastModifiedById	Lookup(User)	
Location	Location__c	Picklist	
Owner	OwnerId	Lookup(User,Group)	

A context menu is open on the right side of the page, showing options like 'Comfy' (selected), 'Compact', 'Display Density', and 'Options'.

Create Field Dependency

The screenshot shows the Edit Field Dependency page for the Department object. The sidebar on the left is identical to the previous screenshot. The main area shows the 'Edit Field Dependency' interface with two picklist columns:

Controlling Field	Location
Dependent Field	Department Type

Below the table, there's an 'Instructions' section with a legend:

- Double click on a cell to toggle its visibility for the Controlling Field value shown in the column heading.
- To change multiple cells at once, select multiple cells and then click the Include Values or Exclude Values button to change the visibility of all selected cells at once.
- Use SHIFT + click to select a range of adjacent cells. Use CTRL + click to select multiple cells that are not adjacent.
- Use the Preview button to test the results.

The 'Included Value' button is highlighted in yellow. Below the instructions, there are two tables for 'Location' and 'Department Type' with rows for 'Kolkata', 'Delhi', 'Banking', 'Finance', 'Education', 'Energy', and 'IT'. Buttons for 'Include Values' and 'Exclude Values' are located at the bottom of each table.

Step 4

Create a Travel Approval Object.

Label = "Travel Approval"

Object Name = "Travel_Approval"

Record Name = "Travel Approval #"

Data Type = "Auto Number"

Display format "TA{00000}".

Allow Reports.

Allow Activities.

Track Field History.

Allow Search

Add Notes & Attachments.

Create custom Tab of your choice.

Add the Tab only to Travel App.

The screenshot shows the Salesforce Setup interface for managing objects. The main area displays the 'Travel Approval' object details, including its API name (Travel_Approval__c) and various configuration tabs like Fields & Relationships, Page Layouts, and Record Types. The right sidebar shows the user profile of KOTA NAVANESH and the current display density setting, which is set to 'Comfy'. Other options include 'Compact' and 'Switch to Salesforce Classic'.

Create the Custom fields...

Purpose Of Trip, Text Area.

Status, Picklist, Values = New, Submitted, Pending Approval, Approved, Rejected, Draft.

Trip Start Date, Date.

Trip End Date, Date.

Out Of State, Checkbox.

Destination State, Text, Length = 2.

Department, Lookup, Related To = Department custom object.

The screenshot shows the Salesforce Setup interface with the URL resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/setup/ObjectManager/01l2w000001frqN/FieldsAndRelationships/view. The page title is "Travel Approval". On the left, there's a sidebar with various setup options like 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, and Triggers. The main content area is titled "Fields & Relationships" and lists 13 items, sorted by Field Label. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, and CONTROLLING FILE. The listed fields are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FILE
Created By	CreatedById	Lookup(User)	
Department	Department_c	Lookup(Department)	
Destination State	Destination_State__c	Text(2)	
Last Modified By	LastModifiedBy/d	Lookup(User)	
Out Of State	Out_Of_State__c	Checkbox	
Owner	OwnerId	Lookup(User,Group)	✓
Purpose of Trip	Purpose_of_Trip__c	Text Area(255)	
Status	Status__c	Picklist	
Status Indicator	Status_Indicator__c	Formula (Text)	
Total Expenses	Total_Expenses__c	Roll-Up Summary (SUM Expense Item)	
Travel Approval #	Name	Auto Number	✓

A context menu is open on the right side of the page, showing options for "DISPLAY DENSITY" (Compact or Comfy), "OPTIONS" (Switch to Salesforce Classic or Add Username), and user information for "KOTA NAVANESH".

Travel App should look like this:
Test The App

The screenshot shows a Salesforce Lightning interface for a Travel Approval record. The URL in the browser is resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/r/Travel_Approval_c/a01zw000017fpGHAA0/view.

Travel Approval #: TA00001

Status: Approved

Out Of State:

Destination State: CA

Trip Info:

- Purpose of Trip: Attend Dreamforce
- Trip Start Date: 2/20/2023
- Trip End Date: 2/28/2023

Created By: KOTA NAVANESH (2/20/2023, 6:14 AM)

Last Modified By: Eric Executive (2/20/2023, 7:31 PM)

Owner: KOTA NAVANESH

Department: Technology

Activity: Comfy (selected)

Upcoming & Overdue: No activities to show.

DISPLAY DENSITY: Comfy (selected), Compact

OPTIONS: Switch to Salesforce Classic, Add Username

Step 5

Import Departments

Download The Department.CSV File

Open “Data Import Wizard”, Click “Launch Wizard”

Select “Departments” Custom Object

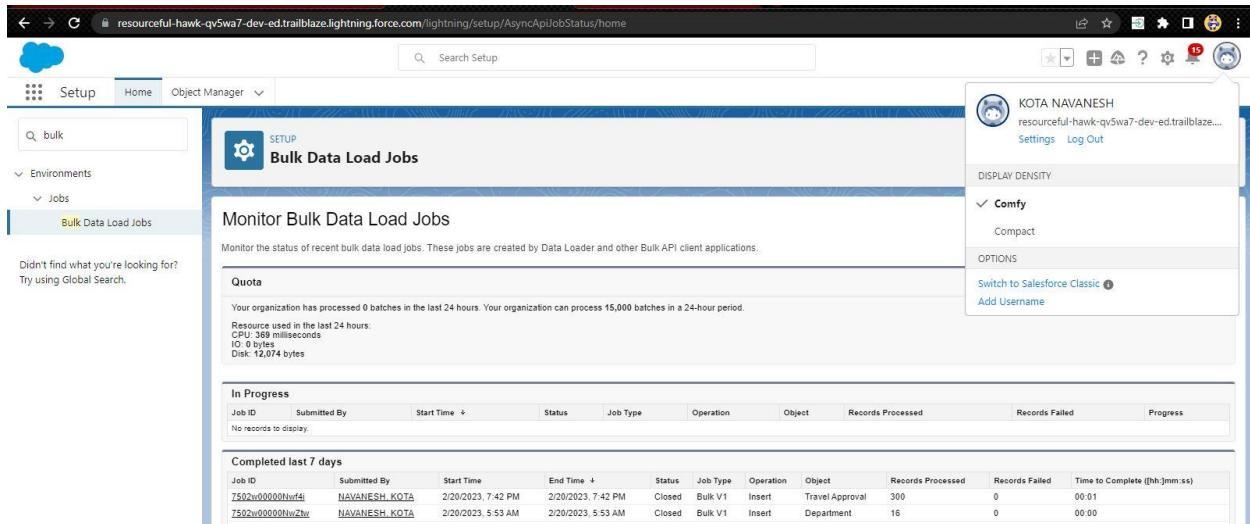
Add New Records Upload the file by clicking the CSV icon

Map Map the Fields, click Next

Start Start Import

Make sure all the records were inserted.

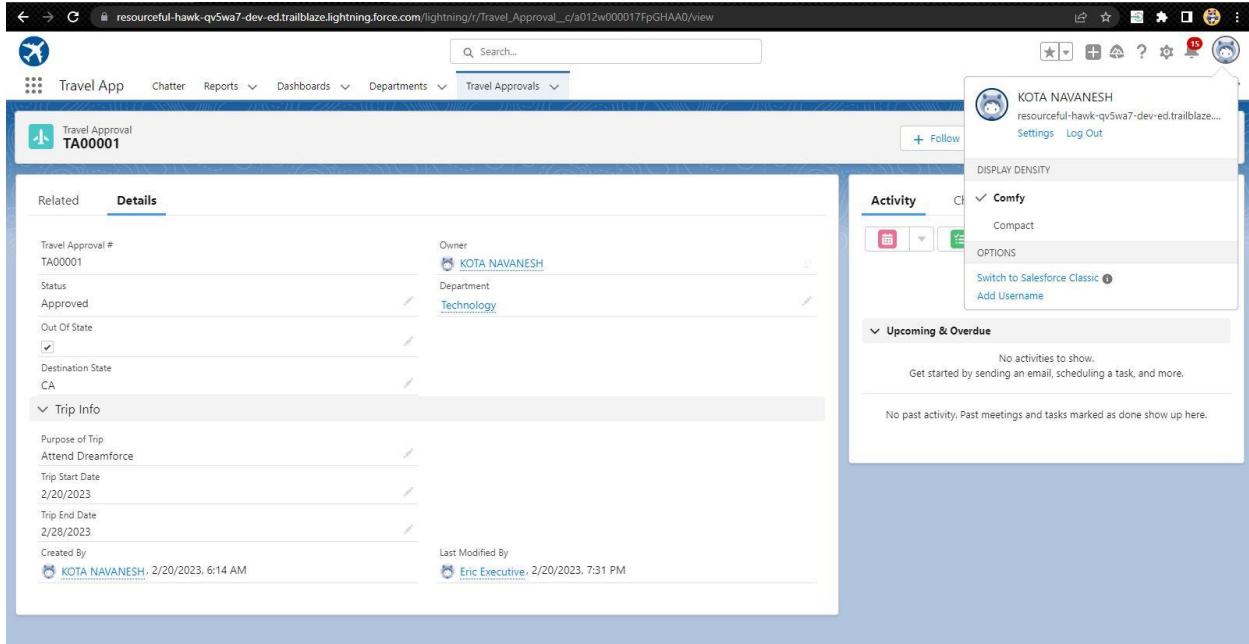
After import then check the status in Bulk Data Load Jobs



Exercise 2

Step 1

Create Travel Approval record



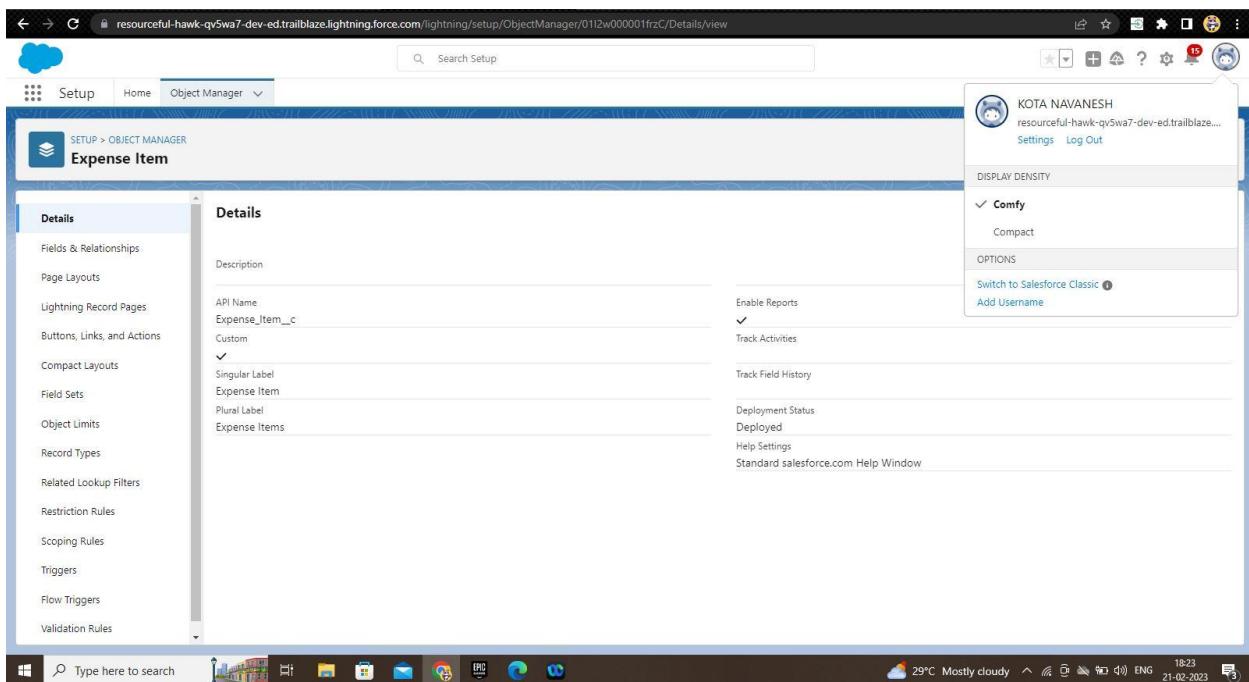
The screenshot shows a Salesforce Lightning page for a Travel Approval record. The URL is https://resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/r/travel_Approval_c/a012w000017FpGHAA0/view. The page title is "Travel Approval TA00001". The "Details" tab is selected. The record contains the following fields:

- Travel Approval #: TA00001
- Status: Approved
- Out Of State:
- Destination State: CA
- Trip Info:
 - Purpose of Trip: Attend Dreamforce
 - Trip Start Date: 2/20/2023
 - Trip End Date: 2/28/2023
- Created By: KOTA NAVANESH (2/20/2023, 6:14 AM)
- Last Modified By: Eric Executive (2/20/2023, 7:31 PM)

The right sidebar shows the user profile of KOTA NAVANESH and a "DISPLAY DENSITY" dropdown set to "Comfy".

Step 2

Create Expense Item Object



The screenshot shows the Salesforce Setup interface for creating a new object named "Expense Item". The URL is <https://resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/setup/ObjectManager/01l2w000001frzC/Details/view>. The "Details" tab is selected. The object configuration includes:

- Description: Expense Item
- API Name: Expense_Item__c
- Custom:
- Singular Label: Expense Item
- Plural Label: Expense Items
- Enable Reports:
- Track Activities:
- Track Field History:
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

The left sidebar lists various setup categories like Fields & Relationships, Page Layouts, and Record Types. The right sidebar shows the user profile of KOTA NAVANESH and a "DISPLAY DENSITY" dropdown set to "Comfy".

Step 3

Create the following custom fields

The screenshot shows the Salesforce Object Manager for the 'Expense Item' object. The 'Fields & Relationships' section is displayed, listing the following fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD
Amount	Amount__c	Currency(16, 2)	
Created By	CreatedById	Lookup(User)	
Expense Item Number	Name	Auto Number	
Expense Type	Expense_Type__c	Picklist	
Last Modified By	LastModifiedById	Lookup(User)	
Travel Approval	Travel_Approval__c	Master-Detail(Travel Approval)	

Step 4

Create Expense Items

The screenshot shows a Travel Approval record with ID TA00001. The 'Expense Items' related list contains two items:

Expense Item Number	Expense Type	Amount
E - 00001	Airfare	₹450.00
E - 00002	Hotel	₹870.00

Step 5 & 6

Create a User

Add user Eric Executive as your manager

The screenshot shows the Salesforce Setup interface under the 'Users' section. A new user record is being created for 'Eric Executive'. The 'User Detail' section includes fields like Name, Alias, Email, Username, Nickname, Title, Company, Department, Division, Address, Time Zone, Locale, Language, Delegated Approver, Manager, Receive Approval Request Emails, Federation ID, App Registration, One-Time Password, and Authenticator. The 'Role' field is set to 'CEO'. The 'User License' field is set to 'Salesforce'. The 'Profile' field is set to 'System Administrator'. The 'Active' checkbox is checked. The 'DISPLAY DENSITY' sidebar is open, showing 'Comfy' selected. The top right corner shows the user's name 'KOTA NAVANESH' and the URL 'resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/setup/ManageUsers/page?address=%2F0052w00000DX255%3Fnoredirect%3D1%26isUserEntityOverride%3D1'.

Step 7

Customize the Travel Approval Default search layout as shown in the screen shot & Save

The screenshot shows the Salesforce Setup interface under the 'Object Manager' section for the 'Travel Approval' object. The 'Search Layouts' tab is selected. A modal window titled 'Edit Search Layout' is open, showing the 'Travel Approval Search Results' configuration. It lists 'Available Fields' such as Record ID, Out Of State, Status Indicator, Total Expenses, Owner Alias, Owner First Name, Owner Last Name, Created By Alias, Created By, Created Date, and Last Modified By Alias. The 'Selected Fields' section contains 'Travel Approval #' (with a dropdown arrow), Purpose of Trip, Department, Status, Destination State, Trip Start Date, and Trip End Date. The 'Up' and 'Down' arrows are used to re-order the selected fields. Below the fields, there is an unchecked checkbox for 'Override the search result column customizations for all users'. The bottom of the modal has 'Save' and 'Cancel' buttons. The top right corner shows the user's name 'KOTA NAVANESH' and the URL 'resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/setup/ObjectManager/01l2w000001frqN/MySearchLayouts/page?address=%2Fp%2Fsetup%2Flayout%2FlistLayoutEdit%...'. The left sidebar lists various customization 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, and Triggers.

Step 8 & 9

Select fields to display in the Travel Approval “All” List view, as shown in the screen shot & Save

The screenshot shows the Salesforce Lightning interface for the 'Travel Approvals' application. A modal dialog titled 'Select Fields to Display' is open over a list of travel approvals. The list is filtered by 'Department' and shows 50+ items. The modal has two panels: 'Available Fields' on the left and 'Visible Fields' on the right. In the 'Available Fields' panel, 'Created By Alias' is selected. In the 'Visible Fields' panel, 'Department', 'Created By', 'Status', 'Trip Start Date', and 'Trip End Date' are listed. At the bottom of the modal are 'Cancel' and 'Save' buttons.

Step 10

Select fields to display in the Travel Approval “Open Out of State Travel Requests” List view, as shown in the screen shot & Save.

The screenshot shows the Salesforce Lightning interface for the 'Travel Approvals' application. A modal dialog titled 'Select Fields to Display' is open over a list of travel approvals. The list is filtered by 'Department' and shows 0 items. The modal has two panels: 'Available Fields' on the left and 'Visible Fields' on the right. In the 'Available Fields' panel, 'Created By Alias' is selected. In the 'Visible Fields' panel, 'Department', 'Created By', 'Status', 'Destination State', 'Trip Start Date', and 'Trip End Date' are listed. At the bottom of the modal are 'Cancel' and 'Save' buttons.

Step 11

Customize the Travel Approval Page Layout as shown in the screen shot.

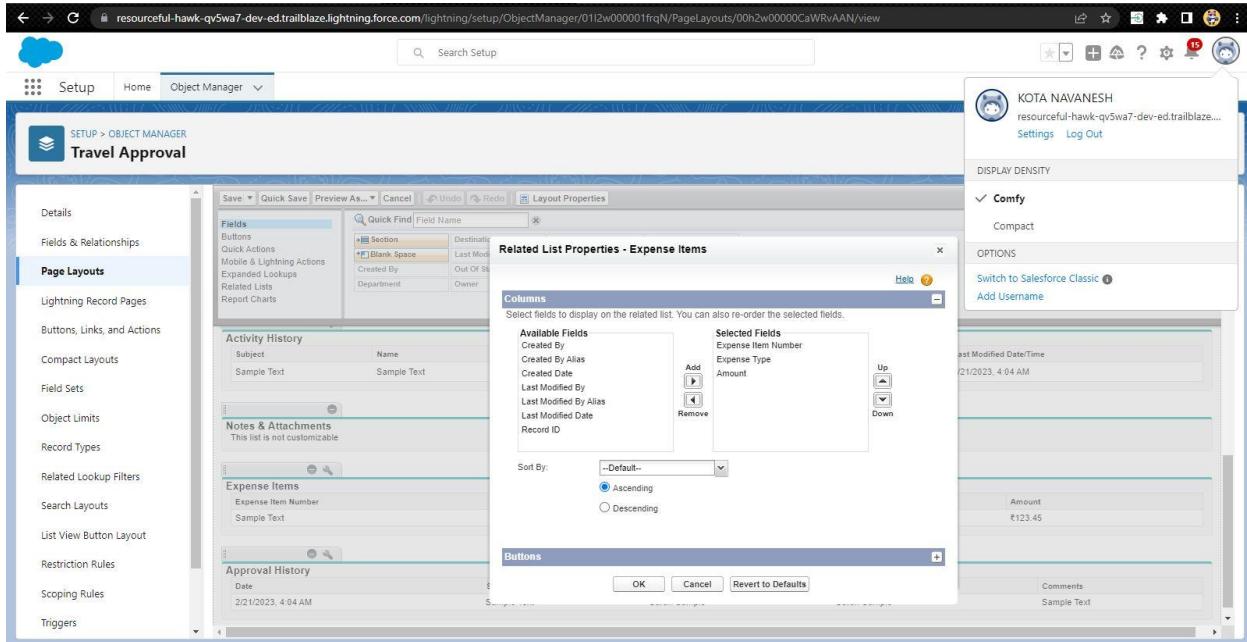
The screenshot shows the Salesforce Visualforce Editor interface. On the left, there's a sidebar with 'Components' and 'Fields' tabs, and a search bar. Below these are various standard components listed under 'Standard (37)'. The main area displays a 'Travel Approval' record page with sections for 'Related' and 'Details'. The 'Details' section contains fields like 'Travel Approval #', 'Status', 'Approved', 'Out Of State', 'Destination State', and 'Trip Info'. The 'Trip Info' section includes fields for 'Purpose of Trip', 'Last Modified By', 'Trip Start Date', 'Trip End Date', and 'Created By'. To the right of the record page is a 'Page' configuration panel with sections for 'Label' (set to 'Travel Approval Record Page'), 'Object' (set to 'Travel Approval'), 'Template' (set to 'Header and Right Sidebar'), and 'Description'. There's also a checkbox for 'Enable page-level dynamic actions for the Salesforce mobile app'. At the top right, there are buttons for 'Analyze', 'Activation...', 'Save', and 'Help'.

Customize the Travel Approval Page Layout as shown in the screen shot

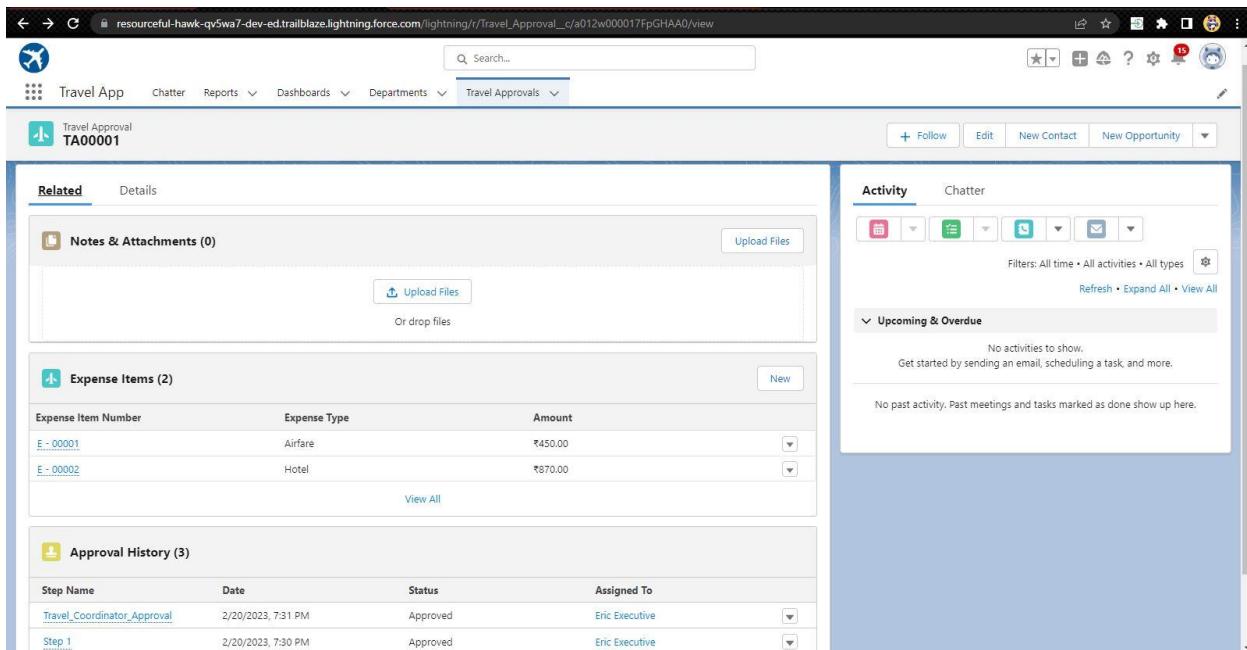
The screenshot shows the Salesforce Setup Object Manager for the 'Travel Approval' object. The left sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, etc. The 'Page Layouts' category is selected. In the main area, the 'Travel Approval Detail' page layout is displayed with sections for 'Information', 'Trip Info', and 'System Information'. A 'Layout Properties' dialog is open over the page layout, showing a 'Fields' section with a 'Quick Find' field name search bar and a list of fields including 'Section', 'Destination State', 'Purpose of Trip', 'Travel Approval #', 'Bank Space', 'Last Modified By', 'Status', 'Trip End Date', 'Created By', 'Out Of State', 'Status Indicator', 'Trip Start Date', 'Department', 'Owner', and 'Total Expenses'. To the right of the page layout, a user profile sidebar shows the user 'KOTA NAVANESH' and options for 'DISPLAY DENSITY' (set to 'Comfy') and 'OPTIONS' (with 'Switch to Salesforce Classic' and 'Add Username' buttons). The top navigation bar includes 'Setup', 'Home', 'Object Manager', and a search bar.

Step 12

Customize the Expense Item Related List under the Travel Approval page layout as shown in the screen shots.



The Travel Approval App should look as shown in the screen shot



Step 13

Enable “Feed Tracking” for Travel Approval Object. Select these 2 fields: Destination State Status Save.

The screenshot shows the Salesforce Setup interface under the Feed Tracking section. A modal window titled "Fields in travel approvals" is open, listing various fields from the Travel Approval object. The "Destination State" field has a checked checkbox labeled "Enable Feed Tracking". Other fields listed include Department, Out Of State, Purpose of Trip, Travel Approval #, Trip Start Date, Owner, Status, and Trip End Date. Below the list, there's a note: "You can also display feed activity for related objects." and a checkbox for "All Related Objects". At the bottom of the modal are "Save" and "Cancel" buttons, along with another "Enable Feed Tracking" checkbox. The background sidebar shows other setup categories like Feature Settings, Chatter, and Service.

Open a Travel Approval record. Click on Chatter Tab. Share a post: Which Department should I associate this travel request with? Mention user Eric Executive on the Post using @.

Note: Login in as Eric and reply to the email, saying: “Technology is the correct department”.

Note: Enable “Administrator can Log in as any user.

The screenshot shows a Travel Approval record with ID TA00001. The Chatter tab is active, displaying a post from user KOTA NAVANESH (@Eric Executive) asking which department to associate the travel request with. A reply from user Eric Executive (@Eric Executive) stating "Technology is the correct department." is shown below. The main record view shows travel details like Airfare and Hotel costs, and an approval history table with three rows: Travel_Coordinator_Approval, Step 1, and Approval Request Submitted.

Test the App

The Travel Approval App should look as shown in the screen shot

The screenshot shows the Salesforce Lightning Experience interface for a Travel Approval record. The URL in the browser is https://resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/r/Travel_Approval/c/a012w000017FpGHAA0/view.

The page title is "Travel Approval TA00001". The main content area displays the following details:

- Travel Approval #: TA00001
- Status: Approved
- Out Of State:
- Destination State: CA
- Trip Info:
 - Purpose of Trip: Attend Dreamforce
 - Trip Start Date: 2/20/2023
 - Trip End Date: 2/28/2023
- Created By: KOTA NAVANESH (2/20/2023, 6:14 AM)
- Last Modified By: Eric Executive (2/20/2023, 7:31 PM)

The top navigation bar includes links for Travel App, Chatter, Reports, Dashboards, Departments, and Travel Approvals. A search bar is also present.

A sidebar on the right shows the user profile of KOTA NAVANESH and various settings options, including "DISPLAY DENSITY" set to "Comfy", "Activity" section showing no upcoming or overdue items, and a "Switch to Salesforce Classic" link.

Module-02

Exercise 1

Step 1

Create Validation Rule

Error Location: Select Field and pick Trip End Date as the location for the error.

Error Message: Trip end date must be greater than or equal to start date.

Make sure to keep “Active” selected/checked.

Name: Trip end date after start date.

Business Logic: Trip end date must always be greater than (\geq) the trip start date.

The screenshot shows the Salesforce Setup interface for creating a Validation Rule. The URL in the browser is <https://resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/setup/ObjectManager/01l2w000001frqN/ValidationRules/03d2w000000Z5fQAA/view>. The page title is "Travel Approval Validation Rule". The validation rule details are as follows:

Rule Name	Trip_end_date_after_start_date	Active
Error Condition Formula	Trip_End_Date__c < Trip_Start_Date__c	
Error Message	Trip end date must be greater than or equal to start date	Error Location
Description		Created By
Created By	KOTA NAVANESH, 2/14/2023, 4:37 AM	Modified By
		KOTA NAVANESH, 2/20/2023, 7:47 PM

The right side of the screen shows a user profile for KOTA NAVANESH and options for changing display density (Compact or Comfy) and switching to Salesforce Classic.

Step 2

Create a Roll-Up Summary Field on Travel Approval object.

Filter Criteria: All records should be included in the calculation

Field to Aggregate: Amount

Roll-Up Type: SUM

Field Name = Total_Expenses

Field Label = Total Expenses

Business Logic: Automatically sum up the total amount of expenses from the related Expense Item.

Save.

The screenshot shows the Salesforce Object Manager interface for creating a custom field. The object being modified is 'Travel Approval'. The 'Fields & Relationships' tab is active. On the left, a sidebar lists various customization options: Details, Fields & Relationships (selected), 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, and Triggers. The main panel displays the 'Custom Field Definition Detail' for the field 'Total Expenses'. Key details shown include:

- Field Information:** Field Label: Total Expenses, Field Name: Total_Expenses, API Name: Total_Expenses__c, Description: (empty), Help Text: (empty), Data Owner: (empty), Field Usage: (empty), Data Sensitivity Level: (empty), Compliance Categorization: (empty).
- Object Name:** Travel Approval
- Created By:** KOTA NAVANESH (2/14/2023, 4:41 AM)
- Modified By:** KOTA NAVANESH (2/14/2023, 4:41 AM)
- Roll-Up Summary Options:** Data Type: Roll-Up Summary, Summarized Object: Expense Item, Field to Aggregate: Expense Item: Amount, Filter Criteria: (empty). Summary Type: SUM.

A context menu is open on the right side of the screen, listing options: Comfy (selected), Compact, OPTIONS, and Add Username. The user profile 'KOTA NAVANESH' is displayed at the top right.

Step 3

Create Formula Fields.

Cache-Control = Public.

File = StatusImages.zip [upload it from the Project Folder]

Name = StatusImages

Setup | Custom Code | Static Resource | New

Business Logic: Create a field that shows a visual indicator based on the value of the Status field.

Save

The screenshot shows the Salesforce Setup interface with the following details:

- Page Header:** resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/setup/StaticResources/page?address=%2F0812w0000008AeL
- Left Navigation Bar:** Includes links for Setup, Home, Object Manager, and a search bar labeled "Search Setup".
- Main Content Area:** Title "Static Resources" with a gear icon. Subtitle "StatusImages".
 - Static Resource Detail:** Name: Statusimages, Namespace Prefix: (empty), Description: (empty), MIME Type: application/zip, Cache Control: Public, Size: 39,130 bytes, Created By: KOTA NAVANESH 2/14/2023, 4:44 AM, Last Modified By: KOTA NAVANESH 2/14/2023, 4:44 AM.
 - Buttons: Edit, Delete, Where is this used?
- Right Sidebar:** User profile for KOTA NAVANESH, resourceful-hawk-qv5wa7-dev-ed.trailblaze...., with options to Settings and Log Out. Includes "DISPLAY DENSITY" dropdown set to "Comfy" (Compact is also listed), "OPTIONS" dropdown with "Switch to Salesforce Classic" and "Add Username" options, and a notification bell icon with 15 notifications.

Step 4

Create a Formula field on the Travel Approval object to show an image based on the Status field.

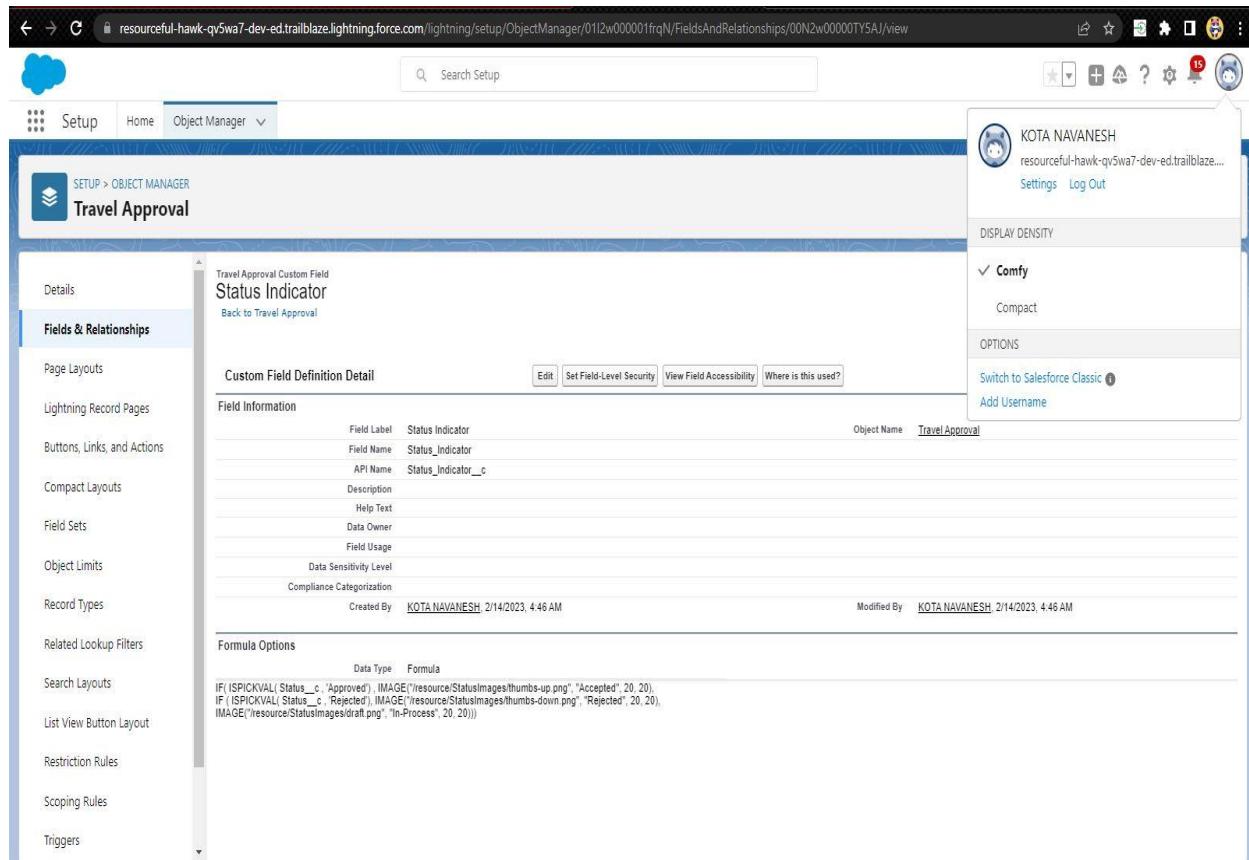
Field Label: Status Indicator

Formula Return Type = Text

Copy & Paste the formula:

```
IF( ISPICKVAL( Status_c , 'Approved') ,  
IMAGE("/resource/StatusImages/thumbs-up.png", "Accepted", 20, 20),  
IF ( ISPICKVAL( Status_c , 'Rejected'), IMAGE("/resource/StatusImages/thumbs-  
down.png", "Rejected", 20, 20),  
IMAGE("/resource/StatusImages/draft.png", "In-Process", 20, 20)))
```

Save



Step 5

Create a Record – Triggered Flow.

Set the Out-of-State checkbox field on the Travel Approval object if out-of-state travel has been chosen.

Select “Freeform”.

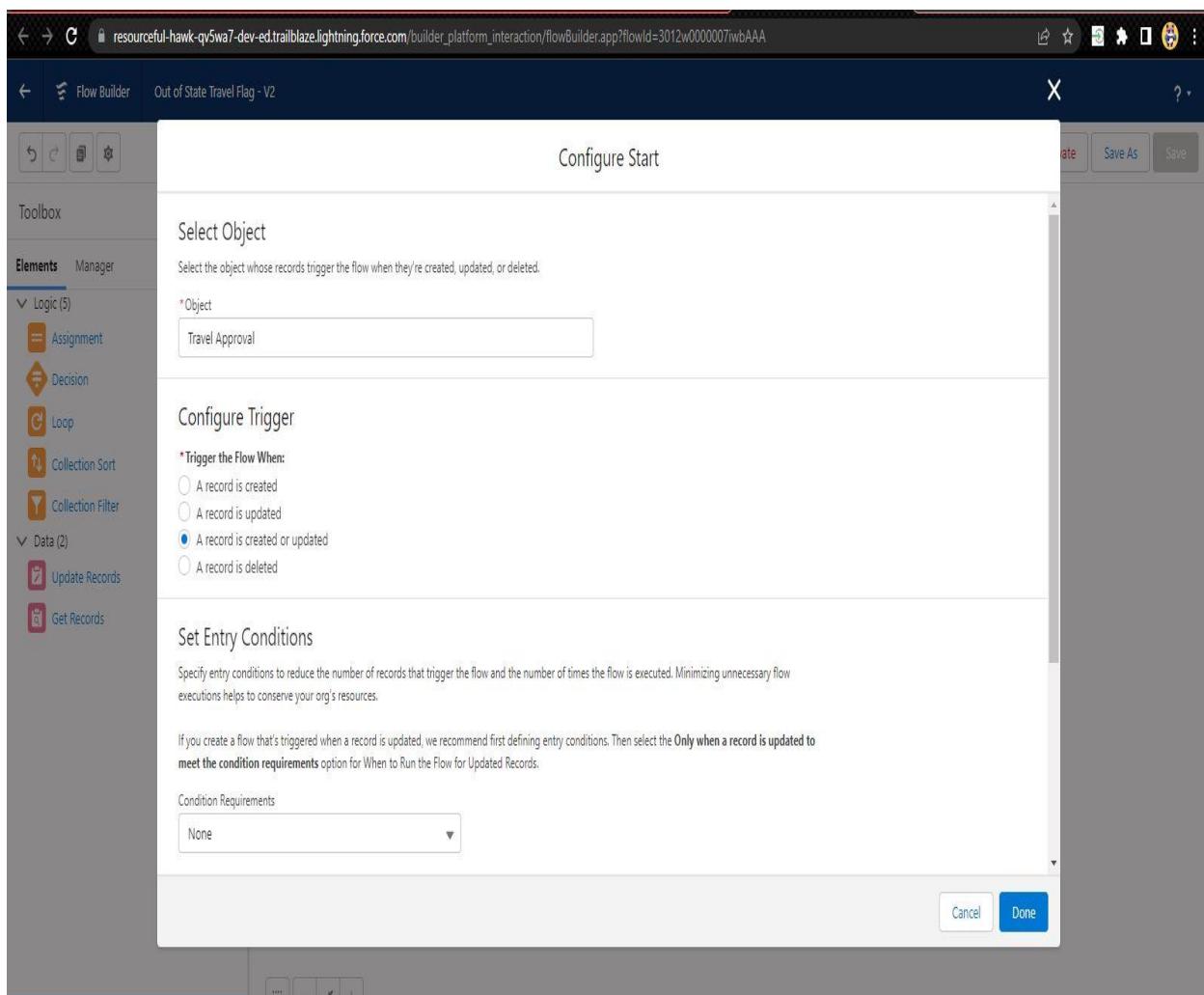
Optimize the Flow For: Fast Field Updates.

Configure Trigger: Trigger the flow when: A record is created or updated.

Condition Requirements = None.

Object: Travel Approval.

Done.



Add a Decision Element to the Flow.

Label = Is Travel Out of State?

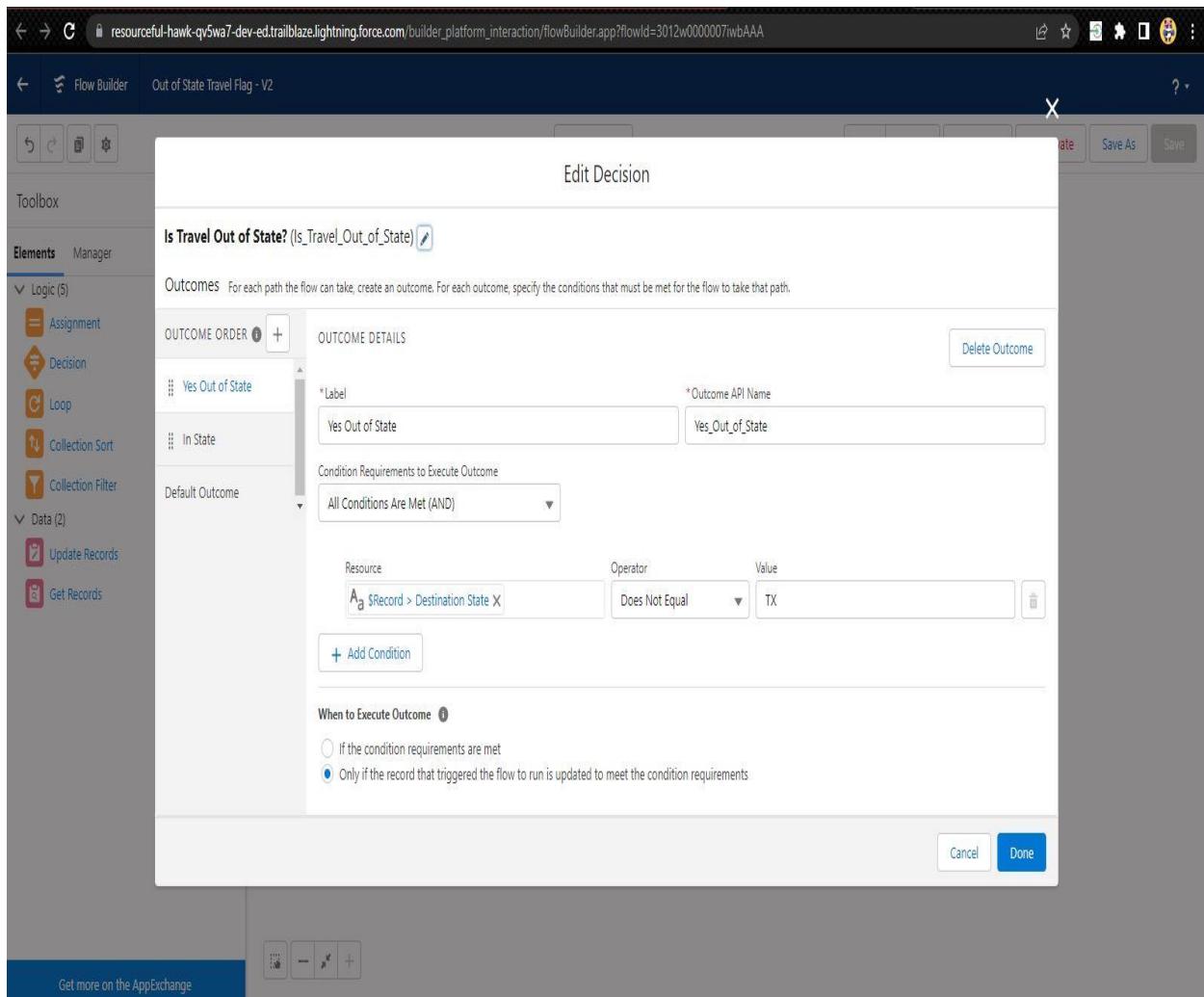
Set the Outcomes, For the first outcome, set these parameters:

Label = Yes Out of State, Condition Requirements = All Conditions Are Met (AND).

Resource = \$Record > Destination State

Operator: Does Not Equal, Value: Tx

When to Execute the Outcome: Only if the record that triggered the flow to run is updated to meet the condition requirements.



Next to Outcome Order click the + button to add another outcome.

Label = In State

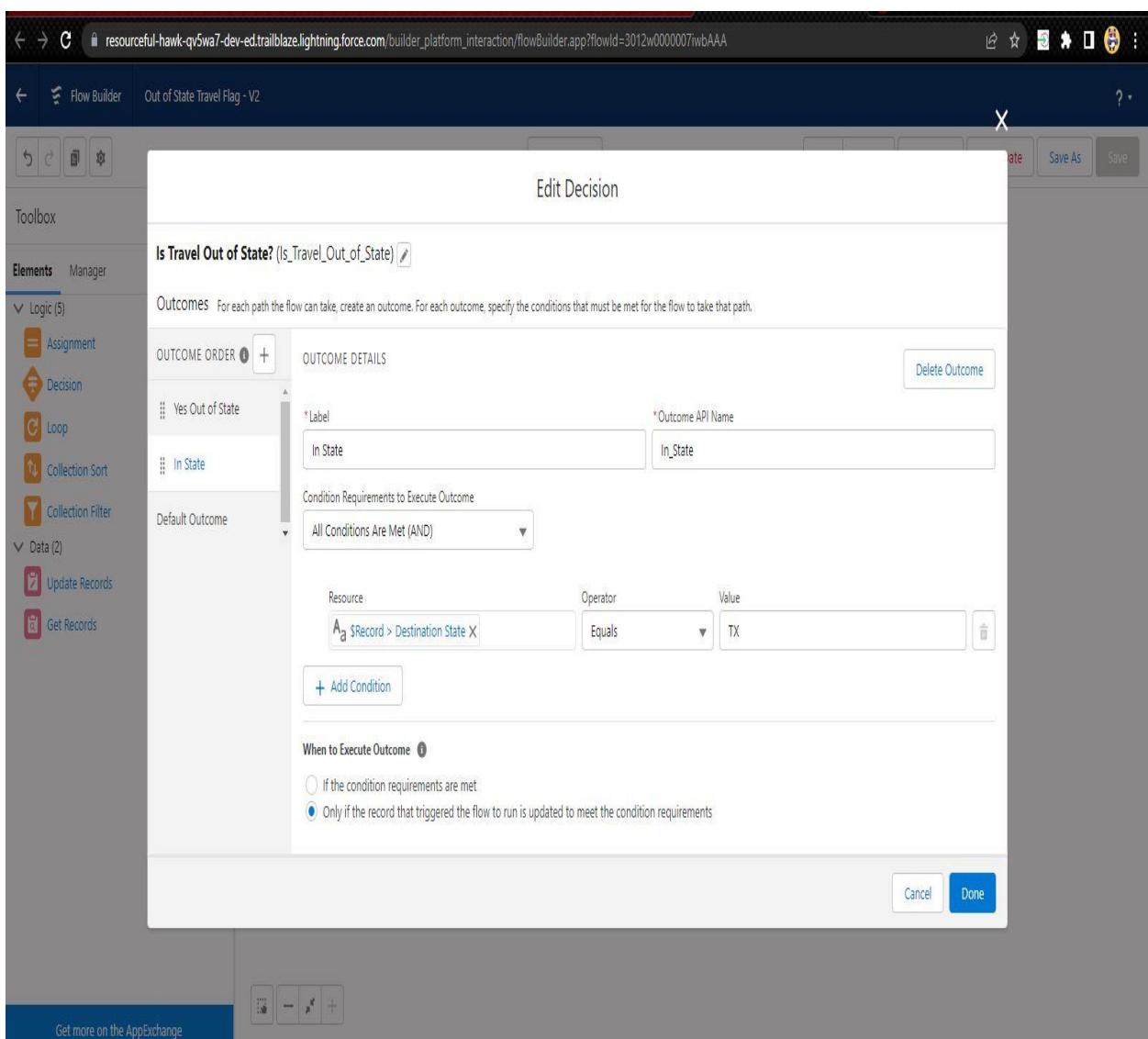
Condition Requirements = All Conditions Are Met (AND).

Resource = \$Record > Destination State

Operator: Equal, Value: Tx

When to Execute the Outcome: Only if the record that triggered the flow to run is updated to meet the condition requirement.

Done



Create an Action for the Flow Using Update Records Elements.

From the flow toolbox, drag an Update Records element onto the flow screen.

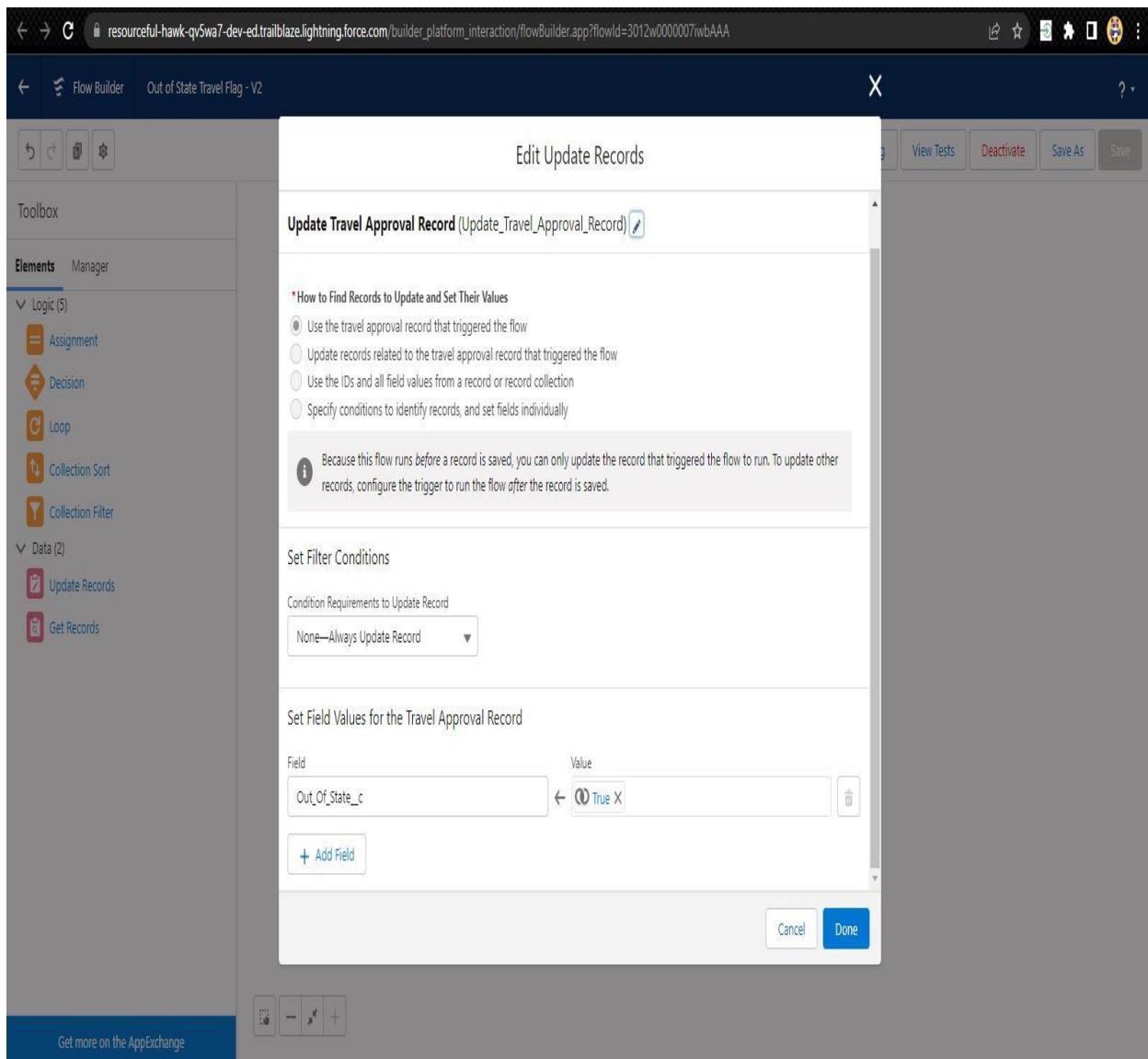
Label = Update Travel Approval Record

How to find Records = Use the travel approval record that triggered the flow

Condition Requirements to Update the Record = None—Always Update the Record

Field: Out_of_State_c

Value = \$GlobalConstant.True (start typing True and this value will come up), Done.



Create an Action for the Flow Using Update Records Elements.

From the flow toolbox, drag an Update Records element onto the flow screen.

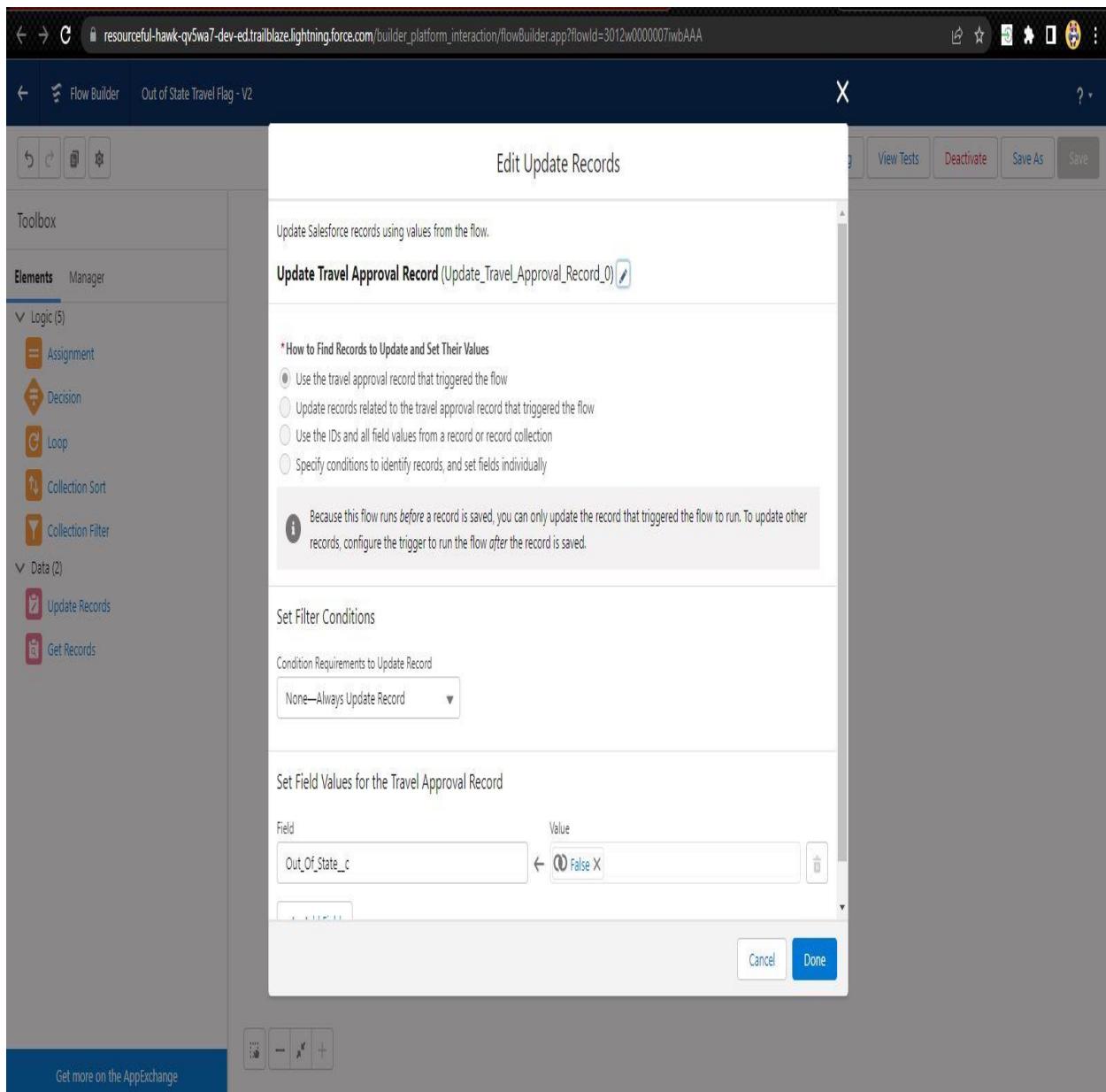
Label = Update Travel Approval Record

How to find Records = Use the travel approval record that triggered the flow

Condition Requirements to Update the Record = None—Always Update the Record

Field: Out_of_State_c

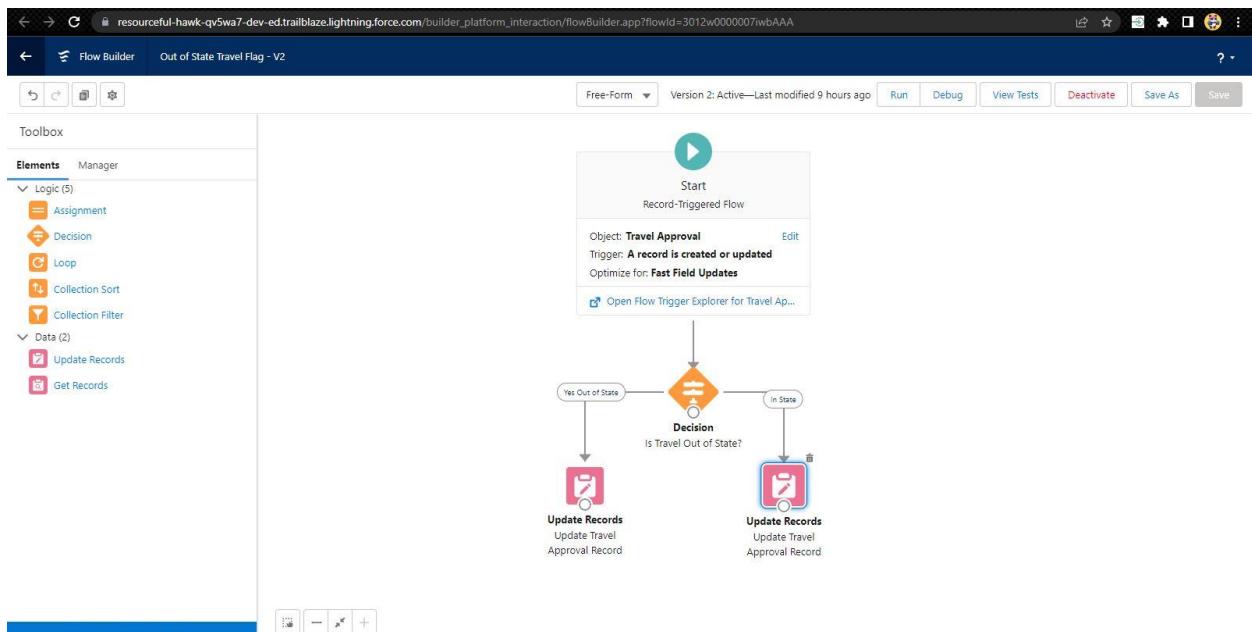
Value = \$GlobalConstant.False (start typing False and this value will come up), Done.



Test The Flow Screen

Drag the white circle from the Decision Node to the Update Records you just created and select the decision In State Done.

Flow screen should look this:



Make sure to save and activate the flow.

Click Activate

Click Save.

How to find Records = Use the travel approval record that triggered the flow

Flow Label = Out of State Travel Flag

Click Save.

Step 6

Create an Approval Process to send Travel approvals to Manager or Travel coordinator. Approval Process Name: Travel Approval Request (Leave rest of the options as it is)

Specify Entry Criteria: Use this approval process if the following: Formula Evaluates to True.

In Formula space, Add “True”. There will be no Error.

Select Approver: Automatically assign an approver using a standard or custom hierarchy field: Manager

Click Save.

The screenshot shows the Salesforce Setup interface with the following details:

Approval Processes - Travel Approval: Travel Approval Request

Process Definition Detail

Process Name	Travel Approval Request	Active
Unique Name	Travel_Approval_Request	Next Automated Approver Determined By
Description		Manager of Record Submitter
Entry Criteria	True	
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests
Approval Assignment Email Template		
Initial Submitters	Travel Approval Owner	
Created By	KOTA NAVANESH 2/14/2023, 6:32 PM	Modified By KOTA NAVANESH 2/20/2023, 7:15 PM

Initial Submission Actions

Action Type	Description
Record Lock	Lock the record from being edited

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit	1	Step 1			Manager	Final Rejection
Show Actions Edit	2	Travel_Coordinator_Approval		Travel Approval: Out Of State EQUALS True	User Eric Executive	Final Rejection

Create Final Approval action.

Click Add New in the Final Approval Actions.

Select Field Update.

Field to Update: Status

Name: Set Status to Approved

Picklist Options: Select A specific value and select Approved from the dropdown list.

Save

The screenshot shows the Salesforce Setup interface with the following details:

Search Bar: Search Setup

Header: resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/setup/WorkflowFieldUpdates/page?address=%2F04Y2w000000UKI4

Left Navigation: Setup, Home, Object Manager, Data (Mass Transfer Approval Requests, Process Automation, Approval Processes), Global Search.

Current Page: SETUP - Field Updates

Field Update Detail:

- Name: Set Status to Approved
- Unique Name: Set_Status_to_Approved
- Description: (empty)
- Object: Travel Approval
- Field to Update: Travel Approval.Status
- Field Data Type: Picklist
- Re-evaluate Workflow Rules after Field Change:
- New Field Value: Approved

Buttons: Edit, Delete

Related Sections:

- Rules Using This Field Update:** This field update is currently not used by any rules.
- Approval Processes Using This Field Update:** Action: Approval Process Name, Description, Type, State.

Action	Approval Process Name	Description	Type	State
Edit Del	Travel Approval Request	Travel Approval	Active	
- Entitlement Processes Using This Field Update:** This field update is currently not used by any entitlement processes.

User Profile Overlay: KOTA NAVANESH, resourceful-hawk-qv5wa7-dev-ed.trailblaze..., Settings, Log Out, DISPLAY DENSITY (Compact selected), OPTIONS, Switch to Salesforce Classic, Add Username.

Create Final Rejection action

The screenshot shows the Salesforce Setup interface with the following details:

- Page Header:** resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/setup/WorkflowFieldUpdates/page?address=%2F04Y2w000000UKi9
- Left Sidebar:** Shows sections like Data, Mass Transfer, Approval Requests, Process Automation, and Approval Processes.
- Current Page:** Field Updates - Set Status to Rejected
- Field Update Detail:**
 - Name: Set Status to Rejected
 - Unique Name: Set_Status_to_Rejected
 - Description: Set Status to Rejected
 - Object: Travel Approval
 - Field to Update: Travel Approval: Status
 - Field Data Type: Picklist
 - Re-evaluate Workflow Rules after Field Change:
 - New Field Value: Rejected
- Rules Using This Field Update:** This field update is currently not used by any rules.
- Approval Processes Using This Field Update:**

Action	Approval Process Name	Description	Type	State
Edit Del	Travel Approval Request	Travel Approval	Travel Approval	Active
- Entitlement Processes Using This Field Update:** This field update is currently not used by any entitlement processes.
- User Profile Overlay:** Shows KOTA NAVANESH, resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com, Settings, Log Out, and options for Display Density (Compact) and Options (Switch to Salesforce Classic, Add Username).

Test the Approval Process

The screenshot shows the Travel App in the Salesforce Lightning interface with the following details:

- Page Header:** resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/r/travel_Approval_c/a012w000017FpGHAA0/view
- Left Sidebar:** Travel App, Chatter, Reports, Dashboards, Departments, Travel Approvals.
- Current Record:** Travel Approval (c/a012w000017FpGHAA0)
- Notes & Attachments (0):** No attachments.
- Expense Items (2):**

Expense Item Number	Expense Type	Amount
E - 00001	Airfare	₹450.00
E - 00002	Hotel	₹870.00
- Approval History (3):**

Step Name	Date	Status	Assigned To
Travel_Coordinator_Approval	2/20/2023, 7:31 PM	Approved	Eric Executive
Step 1	2/20/2023, 7:30 PM	Approved	Eric Executive
Approval Request Submitted	2/20/2023, 7:19 PM	Submitted	KOTA NAVANESH
- User Profile Overlay:** Shows KOTA NAVANESH, resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com, Settings, Log Out, and options for Display Density (Compact) and Options (Switch to Salesforce Classic, Add Username). It also indicates "No past activity. Past meetings and tasks marked as done show up here."

Exercise 2

Step 1

Use Data Import Wizard to import Travel Approval records.
After import check the status in Bulk Data Load Jobs

The screenshot shows the Salesforce Lightning interface with the URL <https://resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/setup/AsyncApiJobStatus/home>. The page title is "Bulk Data Load Jobs". The left sidebar is open, showing "Setup" selected under "Jobs" and "Bulk Data Load Jobs" highlighted. The main content area displays the "Monitor Bulk Data Load Jobs" section, which includes a quota summary and a table of completed jobs over the last 7 days. A user profile menu is visible on the right, showing "KOTA NAVANESH" and "resourceful-hawk-qv5wa7-dev-ed.trailblaze...".

Monitor Bulk Data Load Jobs

Monitor the status of recent bulk data load jobs. These jobs are created by Data Loader and other Bulk API client applications.

Quota

Your organization has processed 0 batches in the last 24 hours. Your organization can process 15,000 batches in a 24-hour period.

Resource used in the last 24 hours:

- CPU: 369 milliseconds
- IO: 0 bytes
- Disk: 12,074 bytes

In Progress

Job ID	Submitted By	Start Time	Status	Job Type	Operation	Object	Records Processed	Records Failed	Progress
No records to display.									

Completed last 7 days

Job ID	Submitted By	Start Time	End Time	Status	Job Type	Operation	Object	Records Processed	Records Failed	Time to Complete (hh:mm:ss)
7502w00000Nwrf4i	NAVANEESH_KOTA	2/20/2023, 7:42 PM	2/20/2023, 7:42 PM	Closed	Bulk V1	Insert	Travel Approval	300	0	00:01
7502w00000NwZhw	NAVANEESH_KOTA	2/20/2023, 5:53 AM	2/20/2023, 5:53 AM	Closed	Bulk V1	Insert	Department	16	0	00:00

Make sure all the 301 Travel Approval records are successfully imported, check the screen shot:

Step 2

Create a Travel Requests by Department Report

X Axis = Department

Y Axis = Record Count

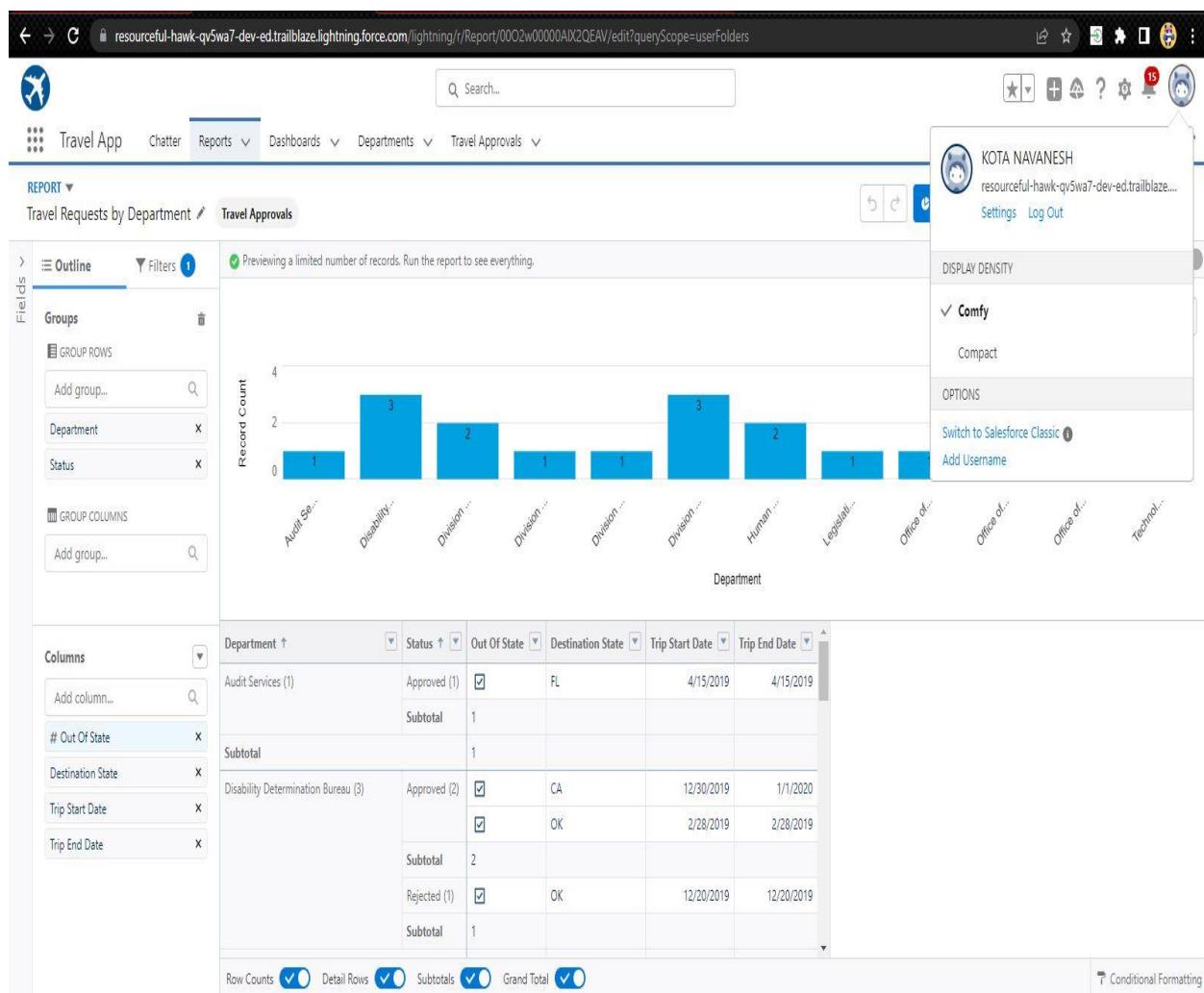
Show Values = Checked.

Group Rows by This Field: Department & Status

Add column: Department, Status, Out-of-State, Destination State, Trip Start/End Date

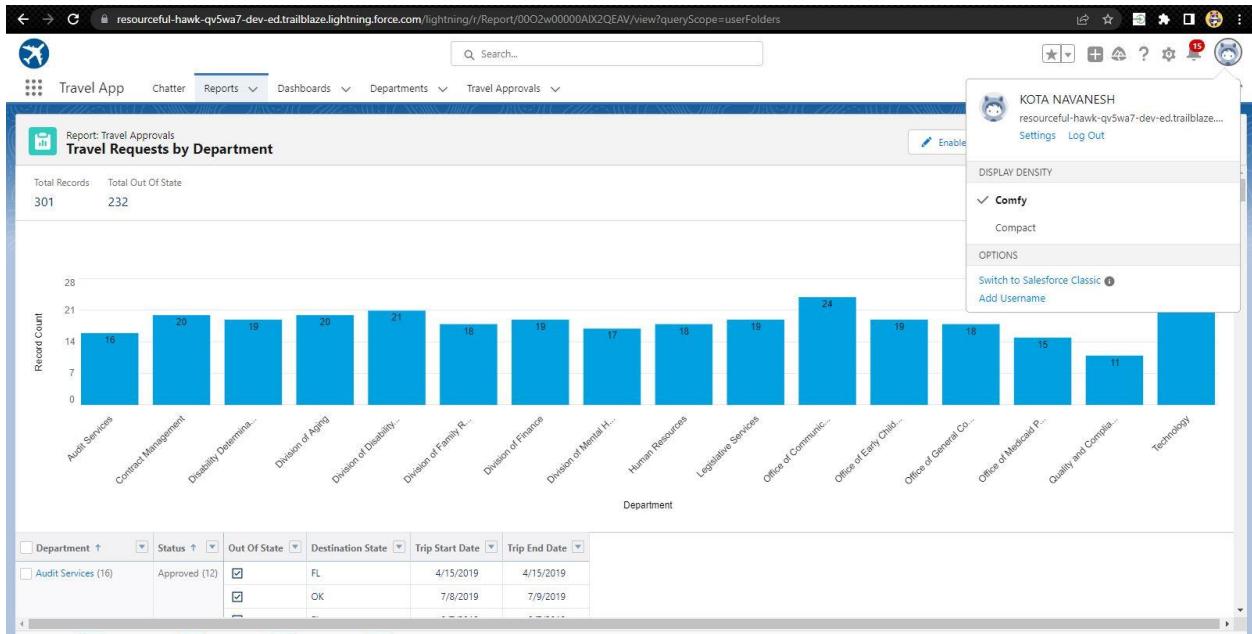
Date

Report types = Travel Approvals.



Test The Report

Report might look as per the screen shot:



Step 3

Create a Travel Requests by Month Report

Group Rows by This Field: Out-Of-State.

Group Rows by This Field: Trip End Date, select Group Dates By | Calendar Month.

Add column: Department, Status, Out-of-State, Destination State, Trip Start/End Date

Report types = Travel Approvals.

The screenshot shows the Salesforce Lightning Report builder interface. The report is titled "Travel Requests by Month" and is set to "Travel Approvals". The report structure is defined by the following filters and groupings:

- Groups:** Out Of State (checkbox selected), Trip End Date (checkbox selected).
- Columns:** Department, Status, Out Of State, Destination State, Trip Start Date.

The report displays data for five months (January 2019 to May 2019) with subtotals for each month. The data is as follows:

Trip End Date	Out Of State	Department	Status	Destination State	Trip Start Date
January 2019 (1)	(1)	Division of Aging	Approved	TX	1/22/2019
		Subtotal			
February 2019 (4)	(4)	Division of Family Resources	Rejected	GA	2/23/2019
		Division of Mental Health and Addiction	Approved	CA	2/7/2019
		Disability Determination Bureau	Approved	OK	2/28/2019
		Division of Aging	Approved	FL	2/18/2019
		Subtotal			
March 2019 (2)	(2)	Office of Early Childhood and Out-of-School Learning	Approved	GA	3/3/2019
		Division of Mental Health and Addiction	Approved	CA	3/18/2019
		Subtotal			
April 2019 (2)	(2)	Audit Services	Approved	FL	4/15/2019
		Office of General Counsel	Approved	FL	4/9/2019
		Subtotal			
May 2019 (1)	(1)	Division of Mental Health and Addiction	Approved	TX	5/2/2019

At the bottom of the report, there are checkboxes for Row Counts, Detail Rows, Subtotals, and Grand Total.

Test The Report

Report might look as per the screen shot:

The screenshot shows a Salesforce Lightning interface for a report titled "Travel Requests by Month". The report displays travel approvals for January 2019, categorized by department. The columns include Trip End Date, Out Of State, Department, Status, Destination State, and Trip Start Date. The data shows 17 records in total, with 6 from the Division of Aging, 11 from various other departments, and 1 from Contract Management.

Trip End Date	Out Of State	Department	Status	Destination State	Trip Start Date
January 2019 (17)	(6)	Division of Aging	Approved	TX	1/22/2019
		Office of General Counsel	Rejected	TX	1/27/2019
		Disability Determination Bureau	Approved	TX	1/12/2019
		Quality and Compliance Office	Approved	TX	1/14/2019
		Legislative Services	Approved	TX	1/20/2019
		Contract Management	Approved	TX	1/18/2019
	Subtotal				
	(11)	Division of Mental Health and Addiction	Approved	OK	1/3/2019
		Contract Management	Approved	GA	1/30/2019
		Division of Disability and Rehabilitative Services	Approved	FL	1/29/2019
		Office of Early Childhood and Out-of-School Learning	Approved	GA	1/18/2019
		Division of Family Resources	Approved	FL	1/21/2019
		Contract Management	Approved	FL	1/29/2019
		Office of Early Childhood and Out-of-School Learning	Rejected	OK	1/4/2019
		Division of Aging	Approved	FL	1/3/2019
		Office of Communications and Media	Rejected	FL	1/12/2019

Report Options at the bottom:

- Row Counts
- Detail Rows
- Subtotals
- Grand Total

User Profile on the right:

- KOTA NAVANESH
- resourceful-hawk-qv5wa7-dev-ed.trailblaze.lightning.force.com/lightning/r/Report/00O2w00000AIXAqEAN/view?queryScope=userFolders
- Settings Log Out

Display Density and Options menu:

- ✓ Comfy
- Compact
- OPTIONS
- Switch to Salesforce Classic ?
- Add Username

Step 4

Create a Travel Approvals Dashboard

Display As | Stacked Vertical Bar Chart, Sort By | Trip End Date, Click Save, click Done.

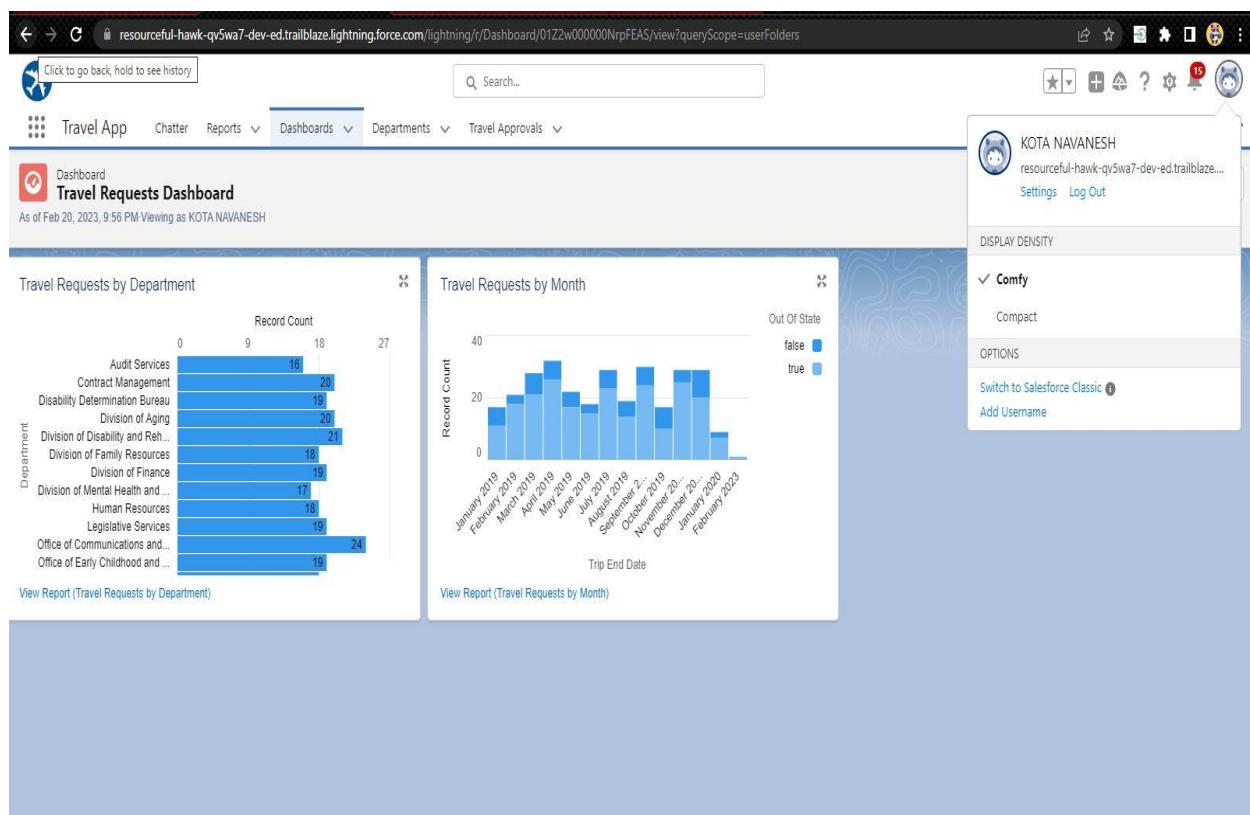
Click + Component, Click the Travel Request by Month report and click Select
Make sure the Y-Axis is set to Department and the X-Axis is set to Record Count.
Click Add

Click + Component, Click the Travel Request by Department report and click Select

Name: Travel Requests Dashboard, Folder: Private Dashboard. Click Create.

Test The Dashboard

Dashboard will look as per the screen shot:

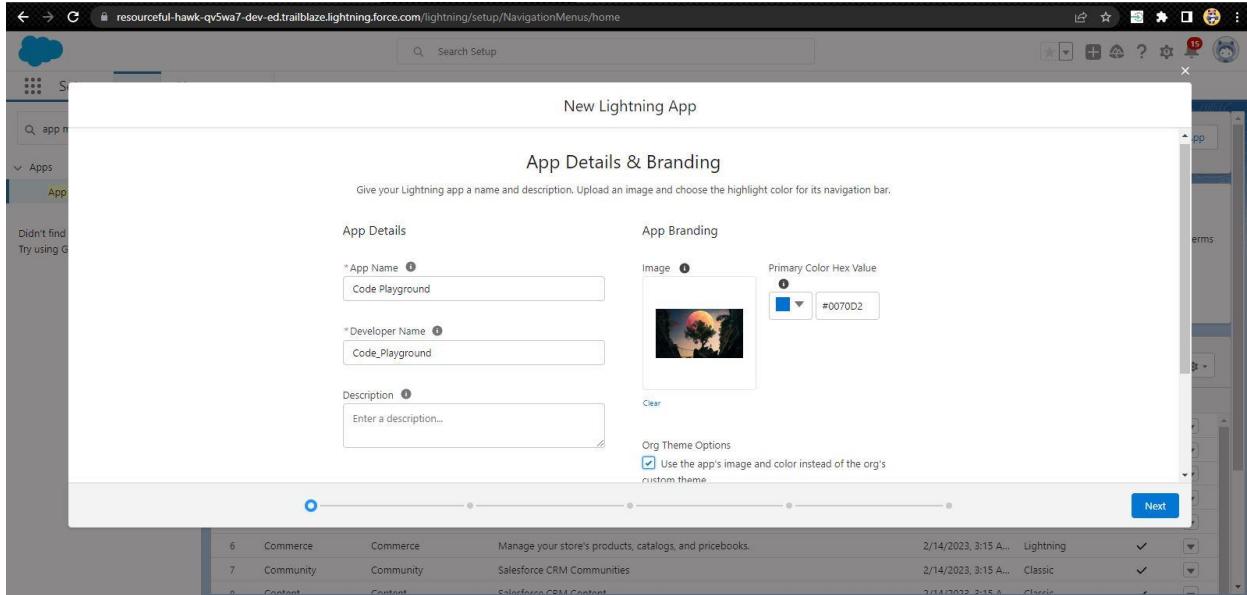


Module-03

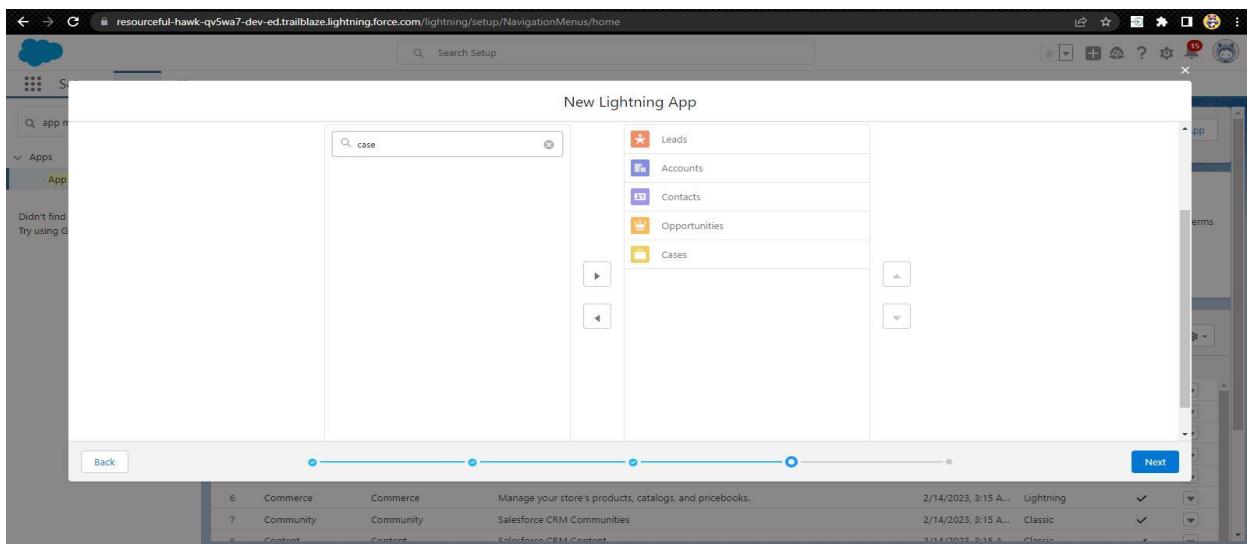
Exercise 1

Step 1

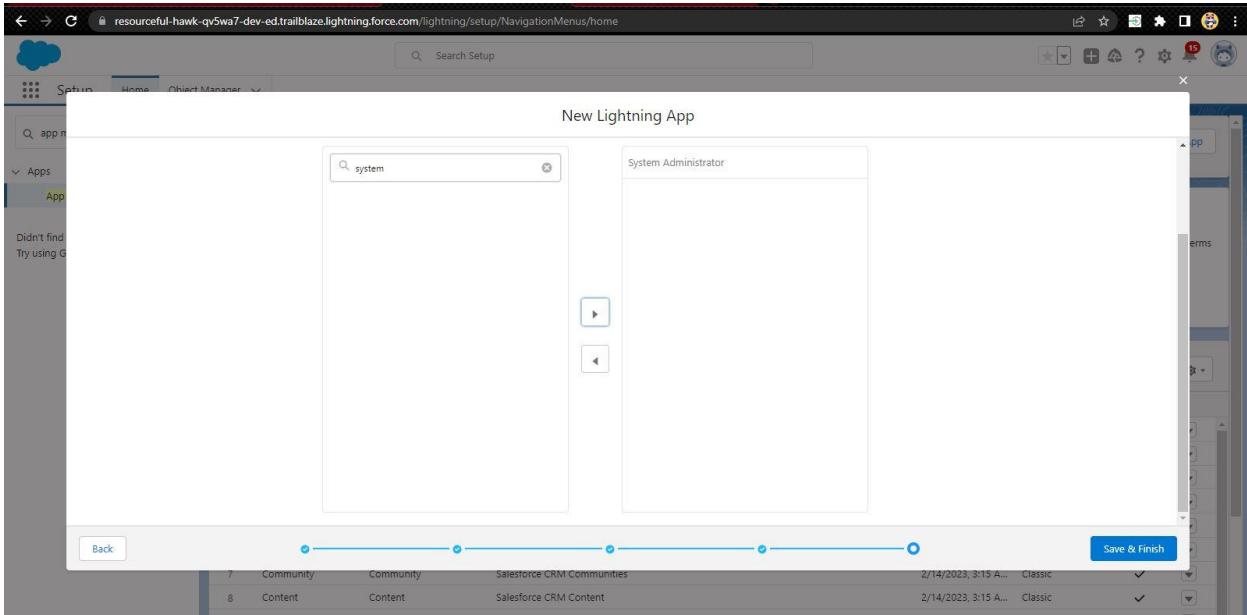
Create a new custom lightning App, name: Code Playground.



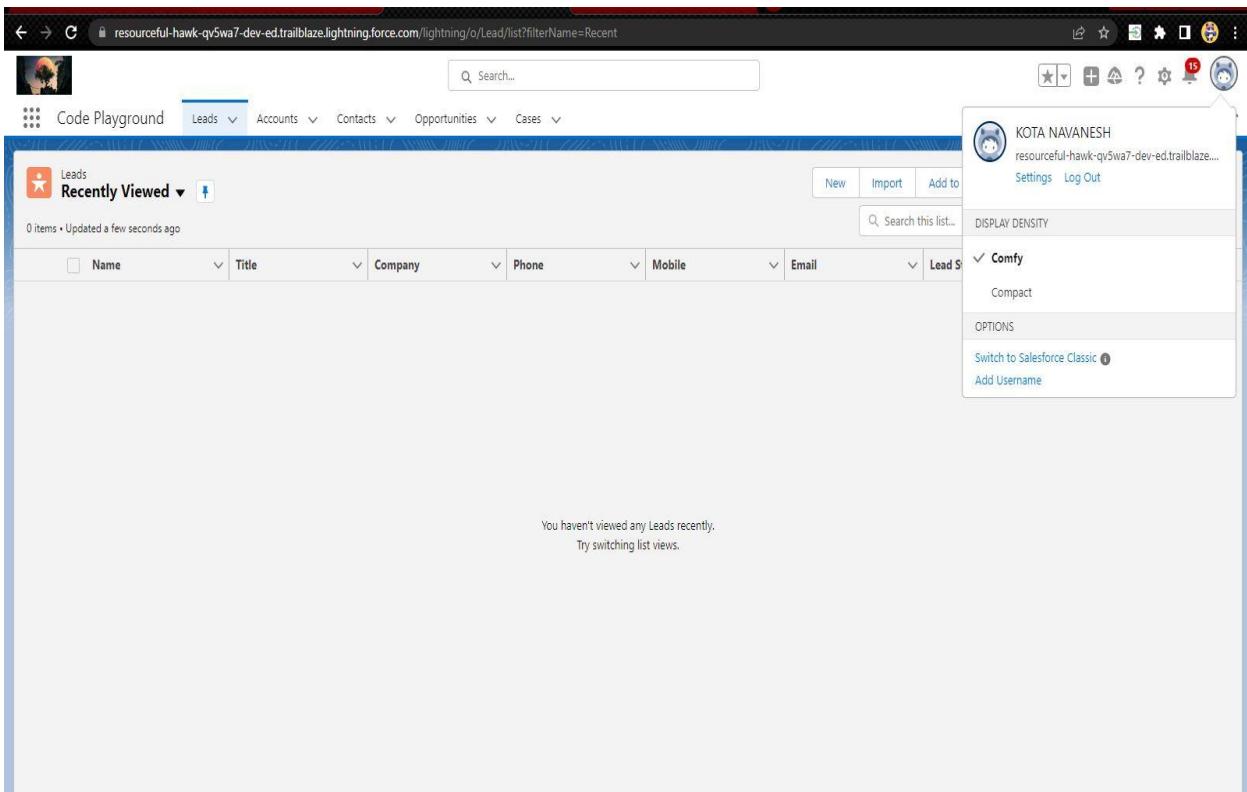
Add Lead, Account, Contact, Opportunity, Case to the Navigation item



User Profile = “System Administrator”



App should be like this :



Create a Custom Object.

The screenshot shows the Salesforce Setup interface for creating a new custom object. The left sidebar lists various configuration options like Fields & Relationships, Page Layouts, and Buttons, Links, and Actions. The main pane displays the 'Customer' object details, including its API name 'Customer__c'. The right sidebar shows user information and settings for display density (set to 'Comfy') and options for switching to Salesforce Classic or adding a username.

Create a Custom fields for Customer Object

The screenshot shows the Salesforce Setup interface for creating custom fields for the 'Customer' object. The left sidebar lists various configuration options. The main pane displays the 'Fields & Relationships' section, which lists several fields: Active (checkbox), Created By (lookup to User), Customer (Master-Detail relationship to Customer), Customer Name (text), Customer Type (picklist), Last Modified By (lookup to User), and Text Area (text area). The right sidebar shows user information and settings for display density (set to 'Comfy') and options for switching to Salesforce Classic or adding a username.

Create a Custom Object.
 Add it to “Code Playground” App only.
 Record Name = Bill Number, Type = Auto Number, Format = B - {0000}, Starting number 1.
 Label = Billing, Plural Label = Billings.

The screenshot shows the Salesforce Object Manager interface. On the left, a sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Billing' under 'Details'. It shows the API Name as 'Billing__c', the Singular Label as 'Billing', and the Plural Label as 'Billings'. A right-hand sidebar displays user information (KOTA NAVANESH) and settings for 'DISPLAY DENSITY' (set to 'Comfy') and 'OPTIONS' (with links to 'Switch to Salesforce Classic' and 'Add Username').

Create a Custom fields for Billing Object

The screenshot shows the 'Fields & Relationships' section of the Object Manager for the 'Billing' object. It lists several fields: Amount Paid (Currency(18, 0)), Bill Number (Auto Number), Created By (Lookup(User)), Customer Type (Picklist), Last Modified By (Lookup(User)), Owner (Lookup(User/Group)), and Status (Picklist). The sidebar on the left shows the 'Fields & Relationships' tab is selected. The right-hand sidebar is identical to the previous screenshot, showing user information and 'DISPLAY DENSITY' settings.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD
Amount Paid	Amount_Paid__c	Currency(18, 0)	
Bill Number	Name	Auto Number	
Created By	CreatedById	Lookup(User)	
Customer Type	Customer_Type__c	Picklist	
Last Modified By	LastModifiedById	Lookup(User)	
Owner	OwnerId	Lookup(User/Group)	
Status	Status__c	Picklist	

Test The App

Code Playground App should look like this:

The screenshot shows the Salesforce Lightning Experience interface. The top navigation bar includes links for Accounts, Leads, Contacts, Opportunities, Cases, Customers, and Billings. Below this is a search bar and a user profile icon for 'KOTA NAVANESH'. A context menu is open on the right side, titled 'DISPLAY DENSITY' with the 'Comfy' option selected. Other options in the menu include 'Compact', 'OPTIONS', 'Switch to Salesforce Classic', and 'Add Username'. The main content area displays a table of 13 account records, each with a checkbox, account name, account site, billing state/province, phone number, and type. The data is as follows:

	Account Name	Account Site	Billing State/Province	Phone	Type	Display Density
1	Burlington Textiles Corp of America		NC	(336) 222-7000	Customer - Direct	KNAVA
2	Dickenson plc		KS	(785) 241-6200	Customer - Channel	KNAVA
3	Edge Communications		TX	(512) 757-6000	Customer - Direct	KNAVA
4	Express Logistics and Transport		OR	(503) 421-7800	Customer - Channel	autproc
5	GenePoint		CA	(650) 867-3450	Customer - Channel	KNAVA
6	Grand Hotels & Resorts Ltd		IL	(312) 596-1000	Customer - Direct	KNAVA
7	Pyramid Construction Inc.			(014) 427-4427	Customer - Channel	KNAVA
8	Sample Account for Entitlements					
9	sForce		CA	(415) 901-7000		KNAVA
10	United Oil & Gas Corp.		NY	(212) 842-5500	Customer - Direct	KNAVA
11	United Oil & Gas, Singapore		Singapore	(650) 450-8810	Customer - Direct	KNAVA
12	United Oil & Gas, UK		UK	+44 191 4956203	Customer - Direct	KNAVA
13	University of Arizona		AZ	(520) 773-9050	Customer - Direct	KNAVA

Exercise 2

Step 1

Define a String Variable & use string method ‘endsWith’ to display the output.

The screenshot shows the Salesforce Developer Console interface. At the top, there's a navigation bar with links like File, Edit, Debug, Test, Workspace, Help, and a search bar. Below the navigation bar is a tab bar with 'Ex_2_L.apxc' and 'Log executeAnonymous @2/22/2023, 6:38:21 PM'. The main area has a 'Logs' tab selected, showing an 'Execution Log' table with two rows of data. The log table has columns for Timestamp, Event, and Details. The first row shows '18:38:21:003 USER_DEBUG [2]DEBUG|true' and the second row shows '18:38:21:003 USER_DEBUG [3]DEBUG|false'. Below the log table is a large 'Enter Apex Code' window containing the following Apex code:

```
1 String data = 'Wipro Limited';
2 system.debug(data.endsWith('Limited'));
3 system.debug(data.endsWith('Wipro'));
```

At the bottom of the developer console, there are buttons for 'Open Log', 'Execute', and 'Execute Highlighted'. There are also checkboxes for 'This Frame', 'Executable', and 'Debug Only', with 'Debug Only' being checked. A 'Filter' button and a 'Click here to filter the log' link are also present. The bottom navigation bar includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems.

Output

This screenshot is identical to the one above it, showing the Salesforce Developer Console interface. It displays the same 'Logs' tab, 'Execution Log' table with the same two log entries, and the same Apex code in the 'Enter Apex Code' window. The interface elements at the bottom, including the log filtering options and the bottom navigation bar, are also identical.

Define 2 Date type variables, use Date method today() & addDays(30) to display the output

The screenshot shows the Salesforce Developer Console interface. In the top-left window, titled 'Enter Apex Code', the following Apex code is written:

```
1 Date thisDay = Date.today();
2 Date newDate = thisDay.addDays(30);
3 system.debug(newDate);
```

Below this window are three buttons: 'Open Log', 'Execute', and 'Execute Highlighted'. The 'Execute' button is highlighted in blue. In the bottom-right window, titled 'Logs', there are two log entries:

User	Operation	Time	Status	Size
KOTA NAVANESH	/services/data/v57.0/tooling/executeAnonymous	2/22/2023, 6:47:03 PM	Success	2.42 KB
KOTA NAVANESH	/services/data/v57.0/tooling/executeAnonymous	2/22/2023, 6:38:21 PM	Success	2.35 KB

Output

The screenshot shows the Salesforce Developer Console interface. In the top-left window, titled 'Execution Log', there is one entry:

Timestamp	Event	Details
18:49:30:003	USER_DEBUG	[3]DEBUG 2023-03-24 00:00:00

Below this window are three checkboxes: 'This Frame', 'Executable', and 'Debug Only', with 'Debug Only' checked. There is also a 'Filter' link and a 'Click here to filter the log' link. In the bottom-right window, titled 'Logs', there are three log entries:

User	Operation	Time	Status	Size
KOTA NAVANESH	/services/data/v57.0/tooling/executeAnonymous	2/22/2023, 6:49:30 PM	Success	2.42 KB
KOTA NAVANESH	/services/data/v57.0/tooling/executeAnonymous	2/22/2023, 6:47:03 PM	Success	2.42 KB
KOTA NAVANESH	/services/data/v57.0/tooling/executeAnonymous	2/22/2023, 6:38:21 PM	Success	2.35 KB

Display the output of an Integer variable from string ‘10’ and then add 20 to it.

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a Log executeAnonymous entry. The main area is titled "Enter Apex Code" and contains the following Apex code:

```
1 String data1 = '10';
2 Integer data2 = Integer.valueOf(data1);
3 system.debug('Sum = '+ (data2+20));
4
5
6
7
8
9
10
11
12
13
14
15
16
17 system.debug('Sum = '+ (data2+20));
18 */
19 }
```

Below the code editor are three buttons: Open Log, Execute, and Execute Highlighted. The "Execute" button is highlighted. The bottom section of the console displays the "Logs" tab, which shows the following log entries:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:52:53 PM	Success		2.5 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:49:30 PM	Success		2.42 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:47:03 PM	Success		2.42 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:38:21 PM	Success		2.35 KB

Output

The screenshot shows the Salesforce Developer Console interface, specifically the "Execution Log" tab. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a Log executeAnonymous entry. The main area is titled "Execution Log" and shows the following log entry:

Timestamp	Event	Details
18:52:53:003	USER_DEBUG	[3][DEBUG]Sum = 30

At the bottom of the screen, there is a "Logs" tab, which displays the same log entries as the Execution Log tab:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:52:53 PM	Success		2.5 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:49:30 PM	Success		2.42 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:47:03 PM	Success		2.42 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:38:21 PM	Success		2.35 KB

Define a String Variable & use string method length() to display the output

The screenshot shows the Salesforce Developer Console interface. In the top navigation bar, it says "Developer Console - Google Chrome" and the URL "resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/ui/common/apex/debug/ApexCSIPage". Below the navigation is a toolbar with "File", "Edit", "Debug", "Test", "Workspace", "Help", and a dropdown for "Code Coverage: None" and "API Version: 57". A sub-menu for "Ex_2_1.apxc" is open, showing the log entry "Log executeAnonymous @2/22/2023, 6:52:53 PM Log executeAnonymous @2/22/2023, 6:55:24 PM". The main area contains an "Enter Apex Code" window with the following code:

```
1 String data1 = 'Manipal Global';
2 system.debug('Length = '+data1.length());
```

Below the code editor are buttons for "Open Log", "Execute", and "Execute Highlighted". To the right of the code editor is a large, empty white space. At the bottom of the developer console are tabs for "Logs", "Tests", "Checkpoints", "Query Editor", "View State", "Progress", and "Problems". The "Logs" tab is selected, showing a table of log entries:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 6:55:24 PM	Success		2.37 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 6:52:53 PM	Success		2.5 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 6:49:30 PM	Success		2.42 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 6:47:03 PM	Success		2.42 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 6:38:21 PM	Success		2.35 KB

Output

The screenshot shows the Salesforce Developer Console interface. The top navigation bar and toolbar are identical to the previous screenshot. The "Logs" tab is selected in the developer console. The "Execution Log" section at the top shows a single entry:

Timestamp	Event	Details
18:55:24:002	USER_DEBUG	[2]@DEBUG Length = 14

Below the execution log is a "Logs" table, which is identical to the one shown in the previous screenshot, listing five log entries from the user KOTA NAVANESH.

Define a List of integer and display the output using add(), get(), set(), clear(), methods

The screenshot shows the Salesforce Developer Console interface. At the top, the URL is `resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCsPage`. Below the header, there's a toolbar with File, Edit, Debug, Test, Workspace, Help, and a Go To button. The main area is titled "Ex_2_1.apxc" and contains the following Apex code:

```

15
16 List<Integer> list1 = new List<Integer>();
17 list1.add(10);
18 list1.add(20);
19 system.debug('List after add = ' + list1);
20 system.debug('Second Element = ' + list1.get(1));
21 list1.set(1,30);
22 system.debug('List after set = ' + list1);
23 list1.clear();
24 system.debug('List after clear = ' + list1);

```

Below the code editor is a toolbar with Open Log, Execute, and Execute Highlighted buttons. The "Logs" tab is selected in the navigation bar. The log table shows the following entries:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:02:09 PM	Success		3.87 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:55:24 PM	Success		2.27 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:52:53 PM	Success		2.5 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:49:30 PM	Success		2.42 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:47:03 PM	Success		2.42 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:38:21 PM	Success		2.35 KB

Output

The screenshot shows the Salesforce Developer Console interface. At the top, the URL is `resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCsPage`. Below the header, there's a toolbar with File, Edit, Debug, Test, Workspace, Help, and a Go To button. The main area is titled "Ex_2_1.apxc" and shows the execution log table:

Execution Log

Timestamp	Event	Details
19:02:09:003	USER_DEBUG	[4]DEBUG>List after add = (10, 20)
19:02:09:003	USER_DEBUG	[5]DEBUG>Second Element = 20
19:02:09:003	USER_DEBUG	[7]DEBUG>List after set = (10, 30)
19:02:09:003	USER_DEBUG	[9]DEBUG>List after clear = ()

Below the log table is a toolbar with This Frame, Executable, Debug Only, Filter, and Click here to filter the log buttons. The "Logs" tab is selected in the navigation bar. The log table shows the same entries as the one in the previous screenshot.

Use Execute Anonymous to define and execute the following code to display the value of x = 0 to 9.

The screenshot shows the Salesforce Developer Console interface. The main area displays the following Apex code:

```

1 Integer x = 20;
2 for(Integer x = 0; x < 10; x++){
3     system.debug('value of x is: '+x);
4 }

```

Below the code editor, there are three buttons: "Open Log", "Execute", and "Execute Highlighted". The "Logs" tab is selected in the navigation bar at the bottom. The log table shows the following entries:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:08:20 PM	Success		5.6 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:02:09 PM	Success		3.87 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:55:24 PM	Success		2.27 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:52:53 PM	Success		2.5 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:49:30 PM	Success		2.42 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:47:03 PM	Success		2.42 KB

Output :

The screenshot shows the Salesforce Developer Console interface. The main area displays the execution log for the anonymous apex script. The log table shows the following entries:

Timestamp	Event	Details
19:08:20:002	USER_DEBUG	[3]DEBUG Value of x is: 0
19:08:20:002	USER_DEBUG	[3]DEBUG Value of x is: 1
19:08:20:002	USER_DEBUG	[3]DEBUG Value of x is: 2
19:08:20:002	USER_DEBUG	[3]DEBUG Value of x is: 3
19:08:20:002	USER_DEBUG	[3]DEBUG Value of x is: 4
19:08:20:003	USER_DEBUG	[3]DEBUG Value of x is: 5
19:08:20:003	USER_DEBUG	[3]DEBUG Value of x is: 6
19:08:20:003	USER_DEBUG	[3]DEBUG Value of x is: 7
19:08:20:003	USER_DEBUG	[3]DEBUG Value of x is: 8
19:08:20:003	USER_DEBUG	[3]DEBUG Value of x is: 9

At the bottom, the "Logs" tab is selected in the navigation bar. The log table shows the same six successful operations from the previous screenshot.

Exercise 3

Answer the following in True Or False:

Integer myunluckyNumber = 7;

Integer myluckyNumber = 15;

myluckyNumber != myunluckyNumber + 8.

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/ui/common/apex/debug/ApexCSIPage`. The tab bar has two tabs: `Ex_2.apxc` and `Ex_3.apxc`, with `Ex_3.apxc` being the active one. The main area is titled "Enter Apex Code" and contains the following Apex code:

```
1 Integer myLuckyNumber = 15;
2 Integer myUnluckyNumber = 7;
3 if(myLuckyNumber != (myUnluckyNumber+8)){
4     system.debug('True');
5 }
6 else{
7     system.debug('False');
8 }
```

Below the code editor are three buttons: "Open Log", "Execute", and "Execute Highlighted". The "Logs" tab is selected in the navigation bar at the bottom. The log table shows the following entries:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 7:13:23 PM	Success		2.64 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 7:08:20 PM	Success		5.6 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 7:02:09 PM	Success		3.87 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 6:55:24 PM	Success		2.27 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 6:52:53 PM	Success		2.5 KB

Answer : False

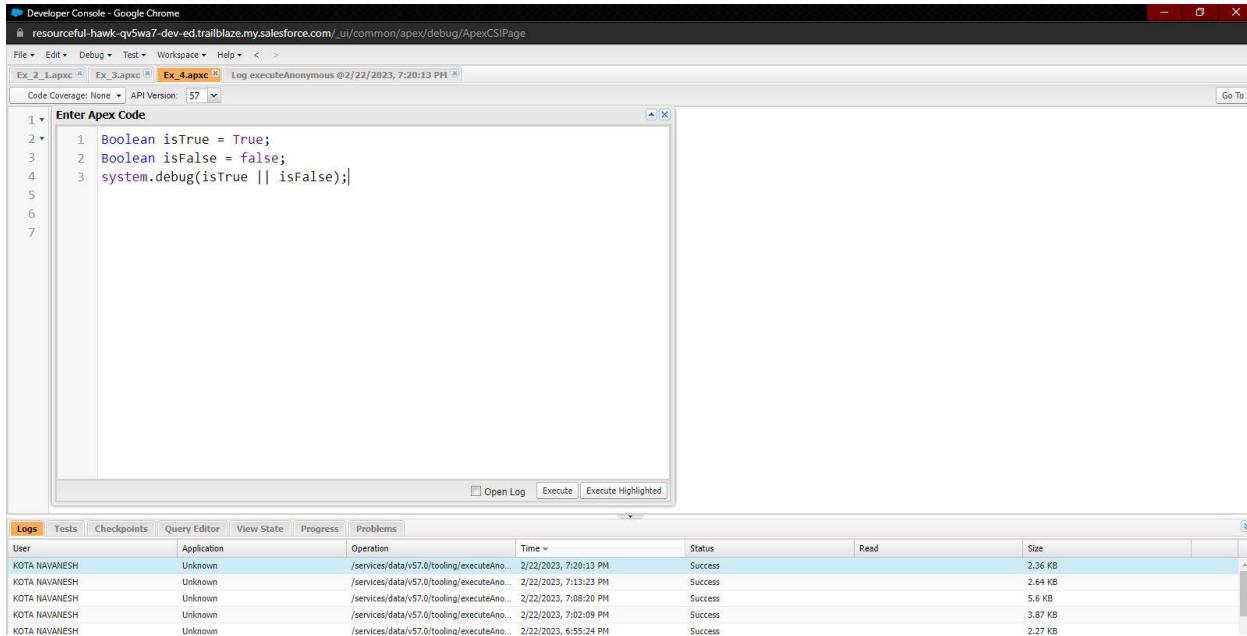
The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/ui/common/apex/debug/ApexCSIPage`. The tab bar has two tabs: `Ex_2.apxc` and `Log executeAnonymous @2/22/2023, 7:13:23 PM`, with the log tab being the active one. The main area displays the "Execution Log" table:

Timestamp	Event	Details
19:13:23:002	USER_DEBUG	[?][DEBUG]False

Below the log table is a "Logs" table, which is identical to the one in the previous screenshot, showing the same five log entries from the developer console.

Exercise 4

Answer the following in True Or False:



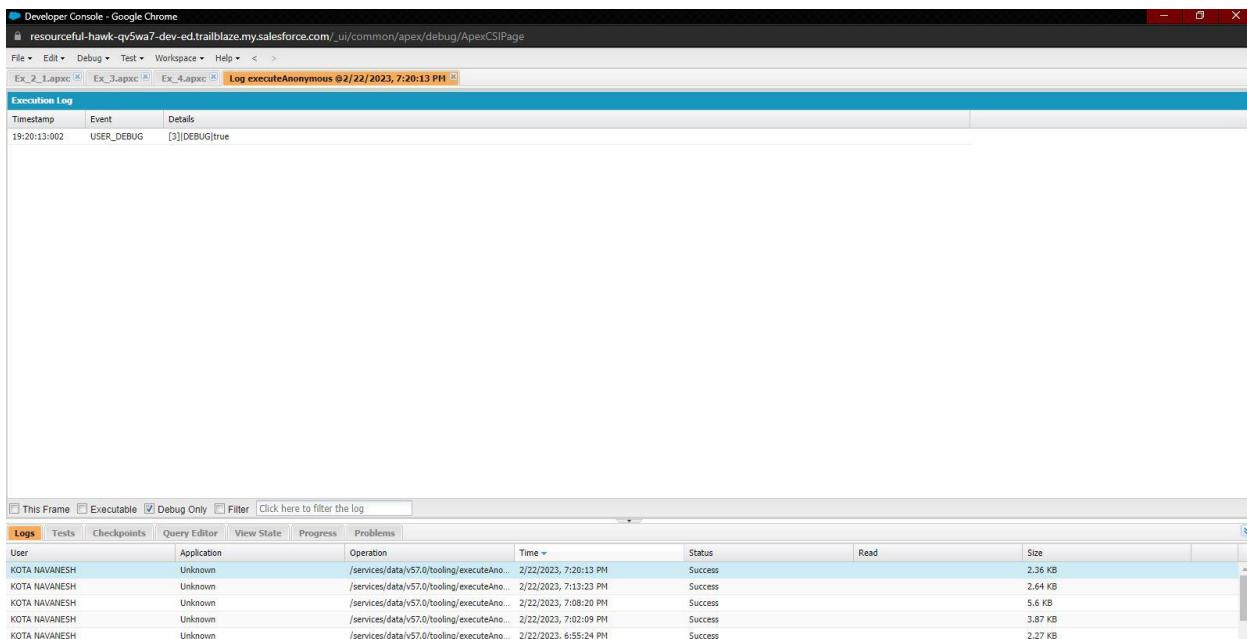
The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tabs at the top are Ex_2.apxc, Ex_3.apxc, and Ex_4.apxc. The current tab is Ex_4.apxc, which contains the following Apex code:

```
1 Boolean.isTrue = True;
2 Boolean.isFalse = false;
3 system.debug(isTrue || isFalse);
```

Below the code editor is a log table:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:20:13 PM	Success		2.36 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:13:23 PM	Success		2.64 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:08:20 PM	Success		5.6 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:02:09 PM	Success		3.87 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:55:24 PM	Success		2.27 KB

Answer : True



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tabs at the top are Ex_2.apxc, Ex_3.apxc, and Ex_4.apxc. The current tab is Ex_4.apxc, which shows the execution log:

Timestamp	Event	Details
19:20:13:002	USER_DEBUG	[3]DEBUG true

Below the log is a log table:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:20:13 PM	Success		2.36 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:13:23 PM	Success		2.64 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:08:20 PM	Success		5.6 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:02:09 PM	Success		3.87 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:55:24 PM	Success		2.27 KB

Exercise 5

Answer the following in True Or False:

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/ui/common/apex/debug/ApexCSIPage`. The tabs at the top include `Ex_2_1.apxc`, `Ex_3.apxc`, `Ex_4.apxc`, `Log executeAnonymous @2/22/2023, 7:20:13 PM`, and `Ex_5.apxc`. The code editor contains the following Apex code:

```
1 Date today = Date.today();
2 Date tomorrow = Date.today().addDays(1);
3 system.debug(today != tomorrow);
```

Below the code editor are buttons for `Open Log`, `Execute`, and `Execute Highlighted`. The log tab is selected, showing the following log entries:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:24:08 PM	Success	Unread	2.46 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:20:13 PM	Success		2.36 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:13:23 PM	Success		2.64 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:08:20 PM	Success		5.6 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:02:09 PM	Success		3.87 KB

Answer : True

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/ui/common/apex/debug/ApexCSIPage`. The tabs at the top include `Ex_2_1.apxc`, `Ex_3.apxc`, `Ex_4.apxc`, `Log executeAnonymous @2/22/2023, 7:20:13 PM`, and `Ex_5.apxc`. The execution log tab is selected, showing the following log entry:

Timestamp	Event	Details
19:24:08:002	USER_DEBUG	[3]DEBUG true

Below the log tab are buttons for `This Frame`, `Executable`, `Debug Only`, `Filter`, and `Click here to filter the log`. The log tab is selected, showing the following log entries:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:24:08 PM	Success		2.46 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:20:13 PM	Success		2.36 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:13:23 PM	Success		2.64 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:08:20 PM	Success		5.6 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:02:09 PM	Success		3.87 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 6:55:24 PM	Success		2.27 KB

Exercise 6

Write a program and execute to demo the use of “If..else if...else”

The screenshot shows the Salesforce Apex Dev Console. The code editor contains the following Apex code:

```
1 Integer score = 90;
2 if(score == 100){
3     system.debug('Grade: A+');
4 }
5 else if(score >= 90){
6     system.debug('Grade: A');
7 }
8 else if(score >= 80){
9     system.debug('Grade: B');
10 }
11 else{
12     system.debug('Grade: Failed');
13 }
```

The logs section shows the following execution results:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 7:31:07 PM	Success		2.55 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 7:24:08 PM	Success		2.46 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 7:20:13 PM	Success		2.36 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 7:13:23 PM	Success		2.64 KB

Output :

The screenshot shows the Salesforce Apex Dev Console. The execution log displays the following output:

Timestamp	Event	Details
19:31:07:002	USER_DEBUG	[6]DEBUG:Grade: A

The logs section shows the following execution results:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 7:31:07 PM	Success		2.55 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 7:24:08 PM	Success		2.46 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnon...	2/22/2023, 7:20:13 PM	Success		2.36 KB

Exercise 7

Write a program to execute and demo the use of “Apex – for Loop”

The screenshot shows the Salesforce Developer Console in Google Chrome. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for various apex files (Ex_2.apxc, Ex_3.apxc, Ex_4.apxc, Ex_5.apxc, Ex_6.apxc, Ex_7.apxc). The active tab is Ex_7.apxc. Below the tabs, there's a code editor with the following Apex code:

```
1 Enter Apex Code
2 1 List<Billing__c> bills = [select Name, Status__c from Billing__c];
3 2 List<String> names = new List<String>();
4 3 for(Billing__c bill:bills){
5 4     system.debug('Value of the Current Recordin the Loop'+bill);
6 5     if(bill.Status__c=='Paid'){
7 6         names.add(bill.Name);
8 7     }
9 8 }
10 9 system.debug('Value of BilliingList '+names);
```

Below the code editor is a toolbar with Open Log, Execute, and Execute Highlighted buttons. The main area shows the execution log table:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:02:58 PM	Success		6.7 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:00:30 PM	Success		6.02 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:31:07 PM	Success		2.55 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:24:08 PM	Success		2.46 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:20:13 PM	Success		2.36 KB

Output:

The screenshot shows the Salesforce Developer Console in Google Chrome. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for various apex files (Ex_2.apxc, Ex_3.apxc, Ex_4.apxc, Ex_5.apxc, Ex_6.apxc, Ex_7.apxc). The active tab is Ex_7.apxc. Below the tabs, there's a table titled "Execution Log" showing the following logs:

Timestamp	Event	Details
20:02:58:049	USER_DEBUG	[4]DEBUG:Value of the Current Recordin the LoopBilling__c{Name=B - 0002, Status__c=Paid, Id=a052v00000EwuFFAAZ}
20:02:58:050	USER_DEBUG	[4]DEBUG:Value of the Current Recordin the LoopBilling__c{Name=B - 0001, Status__c=Paid, Id=a052v00000EwuFFAAZ}
20:02:58:050	USER_DEBUG	[4]DEBUG:Value of the Current Recordin the LoopBilling__c{Name=B - 0003, Status__c=Unpaid, Id=a052v00000EwuFFAAZ}
20:02:58:051	USER_DEBUG	[4]DEBUG:Value of the Current Recordin the LoopBilling__c{Name=B - 0004, Status__c=Paid, Id=a052v00000EwuFBAAZ}
20:02:58:051	USER_DEBUG	[9]DEBUG:Value of BillingList {B - 0002, B - 0001, B - 0004}

Below the execution log is a toolbar with checkboxes for This Frame, Executable, Debug Only, Filter, and a link to Click here to filter the log. The main area shows the execution log table:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:02:58 PM	Success		6.7 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:00:30 PM	Success		6.02 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:31:07 PM	Success		2.55 KB

Exercise 8

Write a Class to demo the use of Constants in Apex

The screenshot shows the Salesforce Developer Console in Google Chrome. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for Ex_2_1.apxc, Ex_3.apxc, Ex_4.apxc, Ex_5.apxc, Ex_6.apxc, Ex_7.apxc, and DiscountClass.apxc. The status bar indicates Log executeAnonymous @2/22/2023, 8:17:19 PM.

```
//Exercise-8
public class DiscountClass {
    static final Decimal regularDiscount = 0.1;
    static Decimal finalPrice = 0;

    public static Decimal calculateDiscount(Integer price){
        finalPrice = price - price * regularDiscount;
        return finalPrice;
    }

    /* Anonymous window
    Decimal price = DiscountClass.calculateDiscount(100);
    system.debug('Final Price '+ price);
    */
}
```

The bottom section displays the Log table:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:17:19 PM	Success		3.96 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:02:58 PM	Success		6.7 kB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:00:30 PM	Success		6.02 kB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:31:07 PM	Success		2.55 kB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:24:08 PM	Success		2.46 kB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:20:13 PM	Success		2.36 kB

Anonymous Window :

The screenshot shows the Salesforce Developer Console in Google Chrome. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for Ex_2_1.apxc, Ex_3.apxc, Ex_4.apxc, Ex_5.apxc, Ex_6.apxc, Ex_7.apxc, and DiscountClass.apxc. The status bar indicates Log executeAnonymous @2/22/2023, 8:17:19 PM.

```
Decimal price = DiscountClass.calculateDiscount(100);
system.debug('Final Price '+ price);
```

The bottom section displays the Log table:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:17:19 PM	Success		3.96 kB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:02:58 PM	Success		6.7 kB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:00:30 PM	Success		6.02 kB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:31:07 PM	Success		2.55 kB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:24:08 PM	Success		2.46 kB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:20:13 PM	Success		2.36 kB

Output :

Developer Console - Google Chrome
resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File • Edit • Debug • Test • Workspace • Help • < >
Ex_2_1.apxc | Ex_3.apxc | Ex_4.apxc | Ex_5.apxc | Ex_6.apxc | Ex_7.apxc | DiscountClass.apxc | Log executeAnonymous @2/22/2023, 8:17:19 PM

Execution Log

Timestamp	Event	Details
20:17:19:015	USER_DEBUG	[2][DEBUG]Final Price 90.0

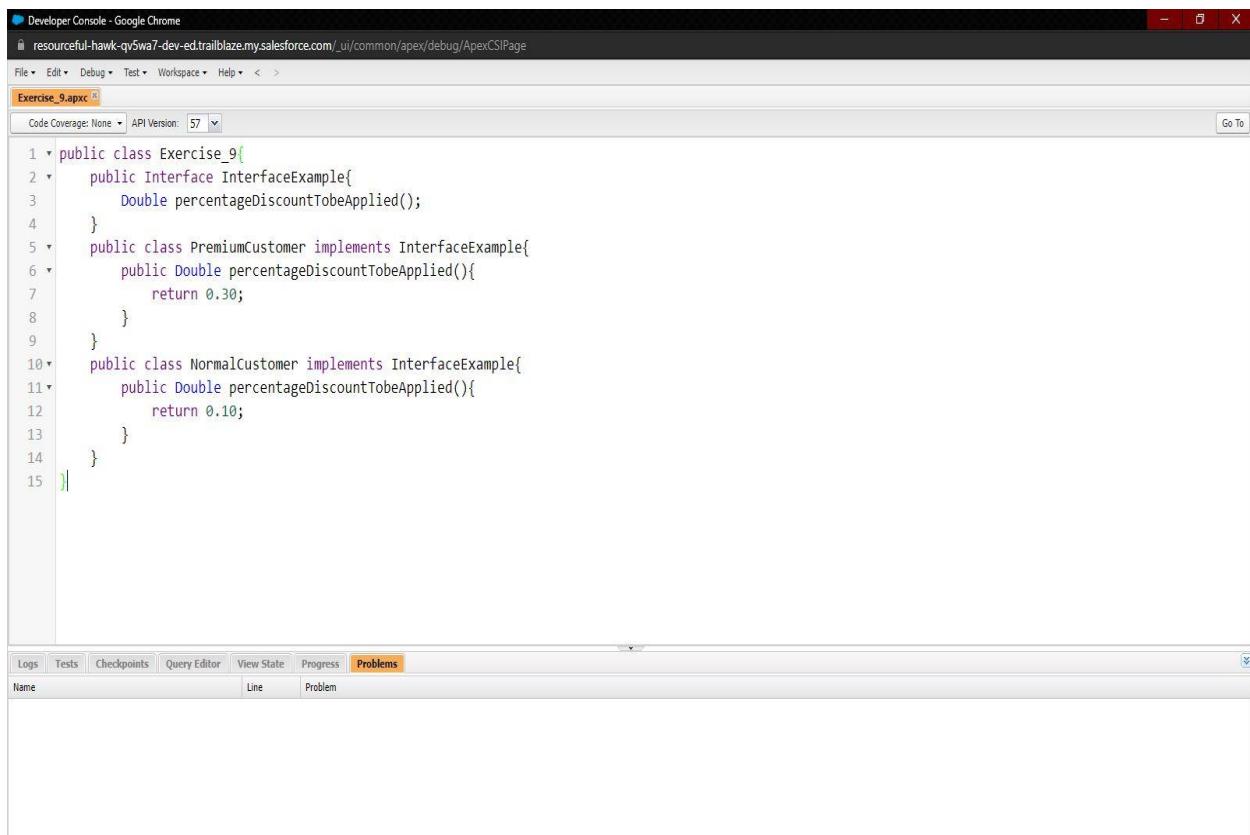
This Frame Executable Debug Only Filter Click here to filter the log

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:17:19 PM	Success		3.96 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:02:58 PM	Success		6.7 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 8:00:30 PM	Success		6.02 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:31:07 PM	Success		2.55 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:24:08 PM	Success		2.46 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAno...	2/22/2023, 7:19:13 PM	Success		2.16 KB

Exercise 9

Write a Class to demo the use of Interface in Apex



The screenshot shows the Salesforce Developer Console interface. The title bar indicates it's running in Google Chrome on a Salesforce developer instance. The main area displays the code for 'Exercise_9.apxc'. The code defines a class 'Exercise_9' containing an interface 'InterfaceExample' with a method 'percentageDiscountTobeApplied()'. It also contains two implementations: 'PremiumCustomer' and 'NormalCustomer', both returning different discount percentages. The code editor has syntax highlighting and line numbers. Below the code editor is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The 'Problems' tab is currently selected, showing an empty list.

```
1 * public class Exercise_9{
2 *     public Interface InterfaceExample{
3 *         Double percentageDiscountTobeApplied();
4 *     }
5 *     public class PremiumCustomer implements InterfaceExample{
6 *         public Double percentageDiscountTobeApplied(){
7 *             return 0.30;
8 *         }
9 *     }
10 *    public class NormalCustomer implements InterfaceExample{
11 *        public Double percentageDiscountTobeApplied(){
12 *            return 0.10;
13 *        }
14 *    }
15 }
```

Exercise 10

Demo on DML Insert Operation Using Database methods

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes 'File', 'Edit', 'Debug', 'Test', 'Workspace', 'Help', and tabs for 'Ex_10.apex' and 'Logs'. Below the tabs, there's a dropdown for 'Code Coverage' and an 'API Version' selector set to 57. A large central window titled 'Enter Apex Code' contains the following Apex code:

```
1 Customer__c customer = new Customer__c(Name='Wipro', Customer_Type__c='Premium');
2 insert customer;
```

At the bottom of this window are three buttons: 'Open Log', 'Execute', and 'Execute Highlighted'. Below this window is a 'Logs' tab. The log table has columns for 'User', 'Application', 'Operation', 'Time', 'Status', 'Read', and 'Size'. One entry is visible:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnonymous	2/23/2023, 4:21:23 PM	Insert failed. First exception on row 0; fi...	Unread	3.15 KB

Output :

The screenshot shows the Salesforce Developer Console interface, similar to the previous one but with different tabs selected. The top navigation bar includes 'File', 'Edit', 'Debug', 'Test', 'Workspace', 'Help', and tabs for 'Ex_10.apex' and 'Logs'. Below the tabs, there's a dropdown for 'Code Coverage' and an 'API Version' selector set to 57. A large central window titled 'Execution Log' displays the execution details of the anonymous apex code. At the bottom of this window are buttons for 'This Frame', 'Executable', 'Debug Only', 'Filter', and a link to 'Click here to filter the log'. Below this window is a 'Logs' tab. The log table has columns for 'User', 'Application', 'Operation', 'Time', 'Status', 'Read', and 'Size'. One entry is visible:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeAnonymous	2/23/2023, 4:21:23 PM	Insert failed. First exception on row 0; fi...	Unread	3.15 KB

Insert Billing record using Database methods

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a Go To button. The main area is titled "Ex_10.apxc" and contains the following Apex code:

```
2 Enter Apex Code
3
4 List<Billing__c> billList = new List<Billing__c>(new Billing__c(Status__c='Paid',Amount_Paid__c=5000000));
5
6 Database.SaveResult[] srList = Database.insert(billList,false);
7
8 for(Database.SaveResult sr:srList){
9     if(sr.isSuccess()){
10         system.debug('Successfully inserted bill. Billing ID = '+sr.getId());
11     }
12     else{
13         for(Database.Error err:sr.getErrors()){
14             system.debug('Error is '+err.getMessage());
15             system.debug('Fields effected by the error = '+err.getFields());
16         }
17     }
18 }
19
20
21
22 }
```

Below the code editor are three buttons: Open Log, Execute, and Execute Highlighted. The bottom section of the console displays the "Logs" tab, which shows two log entries:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeA... 2/23/2023, 4:37:45 PM		Success		4.39 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeA... 2/23/2023, 4:21:23 PM		Insert failed. First exception on row 0; fi...		3.15 KB

Output :

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a Go To button. The main area is titled "Ex_10.apxc" and contains the following Apex code:

```
2 Enter Apex Code
3
4 List<Billing__c> billList = new List<Billing__c>(new Billing__c(Status__c='Paid',Amount_Paid__c=5000000));
5
6 Database.SaveResult[] srList = Database.insert(billList,false);
7
8 for(Database.SaveResult sr:srList){
9     if(sr.isSuccess()){
10         system.debug('Successfully inserted bill. Billing ID = '+sr.getId());
11     }
12     else{
13         for(Database.Error err:sr.getErrors()){
14             system.debug('Error is '+err.getMessage());
15             system.debug('Fields effected by the error = '+err.getFields());
16         }
17     }
18 }
19
20
21
22 }
```

Below the code editor are three buttons: Open Log, Execute, and Execute Highlighted. The bottom section of the console displays the "Logs" tab, which shows two log entries:

User	Application	Operation	Time	Status	Read	Size
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeA... 2/23/2023, 4:37:45 PM		Success		4.39 KB
KOTA NAVANESH	Unknown	/services/data/v57.0/tooling/executeA... 2/23/2023, 4:21:23 PM		Insert failed. First exception on row 0; fi...		3.15 KB

Exercise 11

Write and execute SOQL queries from Developer Console

The screenshot shows the Salesforce Developer Console interface. At the top, there's a navigation bar with links for File, Edit, Debug, Test, Workspace, Help, and a search bar. Below the navigation bar, two tabs are open: 'Ex_10.apxc' and 'ex_11.apxc'. The 'ex_11.apxc' tab is active and displays the following SOQL query:

```
Select Id, Amount, StageName, Account.Name, Account.Industry, Account.Website from Opportunity where Account.Industry = 'Energy' and Account.AnnualRevenue > 5000
```

The results of this query are displayed in a table titled 'Query Results Total Rows: 10'. The columns are: Id, Amount, StageName, Account.Name, Account.Industry, and Account.Website. The data shows 10 rows of opportunities from the 'United Oil & Gas Corp.' account, all categorized under the 'Energy' industry.

Id	Amount	StageName	Account.Name	Account.Industry	Account.Website
0062w00000Jp1ybAAJ	125000	Negotiation/Review	United Oil & Gas Corp.	Energy	http://www.uos.com
0062w00000Jp1yAAJ	270000	Proposal/Price Quote	United Oil & Gas Corp.	Energy	http://www.uos.com
0062w00000Jp1yAAJ	120000	Closed Won	United Oil & Gas Corp.	Energy	http://www.uos.com
0062w00000Jp1yAAJ	270000	Negotiation/Review	United Oil & Gas Corp.	Energy	http://www.uos.com
0062w00000Jp1yAAJ	270000	Closed Won	United Oil & Gas Corp.	Energy	http://www.uos.com
0062w00000Jp1yAAJ	915000	Closed Won	United Oil & Gas Corp.	Energy	http://www.uos.com
0062w00000Jp1yAAJ	235000	Closed Won	United Oil & Gas Corp.	Energy	http://www.uos.com
0062w00000Jp1yAAJ	440000	Closed Won	United Oil & Gas Corp.	Energy	http://www.uos.com
0062w00000Jp1z1AAJ	120000	Closed Won	United Oil & Gas Corp.	Energy	http://www.uos.com
0062w00000Jp1z3AAJ	675000	Needs Analysis	United Oil & Gas Corp.	Energy	http://www.uos.com

Below the table, there are buttons for 'Query Grid', 'Save Rows', 'Insert Row', 'Delete Row', and 'Refresh Grid'. To the right, there are links for 'Access in Salesforce', 'Create New', 'Open Detail Page', and 'Edit Page'. The bottom left of the interface shows a message: 'Any query errors will appear here...'. On the right side, there's a 'History' panel with a single entry: 'Executed'.

Exercise 12

Write an Apex Trigger, Name = CustomerTrigger

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab title is "CustomerTrigger.apxc". The code editor contains the following Apex trigger:

```
trigger CustomerTrigger on Customer__c (after insert, after update) {
    List<Billing__c> billingList = new List<Billing__c>();
    for(Customer__c customer:Trigger.New){
        if(customer.Active__c == True){
            Billing__c billing = new Billing__c();
            billing.Status__c = 'Paid';
            billing.Amount_Paid__c = 1000000;
            billingList.add(billing);
        }
    }
    insert billingList;
}
```

The console interface includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Tests tab is selected. Below the tabs is a table with columns: User, Application, Operation, Time, Status, Read, and Size. There are no entries in the table.

Exercise 13

Write a Test Class for CustomerTrigger

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab title is "TestCustomerTrigger.apxc". The code editor contains the following Apex test class:

```
//Exercise 13
@isTest
public class TestCustomerTrigger {
    @isTest static void testName(){
        Customer__c cus = new Customer__c();
        cus.Active__c = False;
        insert cus;

        Test.startTest();
        cus.Active__c = True;
        update cus;
        Test.stopTest();
    }
}
```

The console interface includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Tests tab is selected. Below the tabs is a table with columns: Status, Test Run, Enqueued Time, Duration, Failures, Total, and Overall Code Coverage. The table shows multiple successful test runs. To the right of the table is a "Overall Code Coverage" summary table:

Class	Percent	Lines
Overall	52%	
CustomerTrigger	100%	10/10
DiscountClass	0%	0/5
Exercise_9	0%	0/4

Exercise 14

Write an Apex Trigger, Name = DisqualifyTestLeads

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab title is "DisqualifyTestLeads.aptr". The code editor contains the following Apex trigger:

```
1 //exercise-14
2 trigger DisqualifyTestLeads on Lead (before insert,before update,after insert) {
3
4     List<Lead> li = new List<Lead>();
5     if(Trigger.isBefore && Trigger.isInsert){
6         for(Lead myLead : Trigger.New){
7             if(( String.isNotBlank(myLead.FirstName) && String.isNotBlank(myLead.LastName)) && (myLead.FirstName=='Test' || myLead.LastName=='Test')){
8                 System.debug(myLead.FirstName + ' ' + myLead.LastName + ' Will be disqualified!');
9                 li.add(myLead);
10            }
11        }
12
13        for(Lead myLead : li){
14            myLead.Status = 'Disqualified';
15        }
16    }
17
18 }
```

The code editor has tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Tests tab is selected. Below the code editor is a table with columns: User, Application, Operation, Time, Status, Read, and Size.

Exercise 15

Write a Test Class for DisqualifyTestLeads

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab title is "DisqualifyTestLeads.aptr" and the sub-tab is "DisqualifyTestLeads.apxc". The code editor contains the following Apex test class:

```
1 @isTest
2 public class DisqualifyTestLeads {
3     @isTest
4     private static void insertData1(){
5         Lead l1 = new Lead();
6         l1.FirstName = 'Test';
7         l1.LastName = 'Test';
8         l1.Company = 'Test';
9         Test.startTest();
10        insert l1;
11        Test.stopTest();
12        System.assertEquals('Test', l1.FirstName);
13        System.assertEquals('Test', l1.LastName);
14        System.assertEquals('Test', l1.Company);
15    }
16 }
```

The code editor has tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Tests tab is selected. Below the code editor is a table with columns: Status, Test Run, Enqueued Time, Duration, Failures, Total, and Overall Code Coverage. The "Test Run" row shows a successful run at 6:26:30 pm. The "Overall Code Coverage" table shows the following data:

Class	Percent	Lines
Overall	26%	
CustomerTrigger	0%	0/10
DiscountClass	0%	0/5
DisqualifyTestLeads	100%	7/7
Exercise_9	0%	0/4

Exercise 16

Create a Visualforce page which displays Opportunity fields as output fields

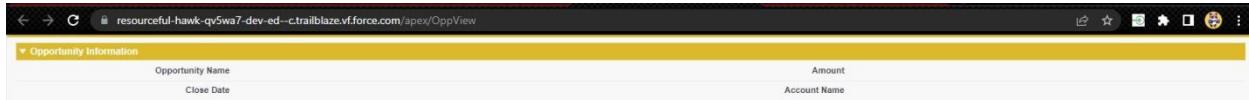
The screenshot shows the Salesforce Developer Console interface. At the top, the URL is `resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. Below the URL, there are tabs for `DisqualifyTestLeads.apxc`, `OppView.vfp`, and `OppView.vfp` (highlighted in orange). The `OppView.vfp` tab contains the following Apex code:

```
1 <apex:page standardController="Opportunity" tabStyle="Opportunity">
2   <apex:pageBlock>
3     <apex:pageBlockSection title="Opportunity Information">
4       <apex:outputField value="{!Opportunity.name}" />
5       <apex:outputField value="{!Opportunity.amount}" />
6       <apex:outputField value="{!Opportunity.closeDate}" />
7       <apex:outputField value="{!Opportunity.account.name}" />
8     </apex:pageBlockSection>
9   </apex:pageBlock>
10 </apex:page>
```

Below the code, there is a table titled "Tests" showing a single test run named "Test Run". The table includes columns for Status, Test Run, Enqueued Time, Duration, Failures, Total, and Overall Code Coverage. The coverage details are as follows:

		Overall Code Coverage						
Status	Test Run	Enqueued Time	Duration	Failures	Total	Class	Percent	Lines
✓	TestRun @ 6:26:30 pm			0	1	CustomerTrigger	26%	0/10
						DiscountTrigger	n/a	n/a

Preview :



Exercise 17

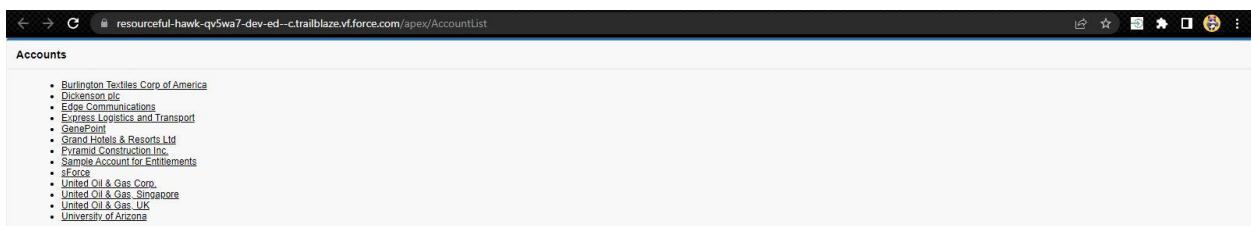
Create a Visualforce page which shows a list of Accounts linked to their record pages

The screenshot shows the Salesforce Developer Console interface. At the top, there are tabs for 'DisqualifyTestleads.apex', 'OppView.vfp', and 'AccountList.vfp'. The 'AccountList.vfp' tab is active. Below the tabs, the code for the Visualforce page is displayed:

```
1 <apex:page standardController="Account" recordSetVar="accounts">
2
3     <apex:pageBlock title="Accounts">
4         <ul>
5             <apex:repeat value="{!Accounts}" var="a">
6                 <li>
7                     <apex:outputLink value="/{!a.Id}">{!a.Name}</apex:outputLink>
8                 </li>
9             </apex:repeat>
10        </ul>
11    </apex:pageBlock>
12 </apex:page>
```

Below the code, there is a 'Logs' section with a single entry: 'Test Run' at 6:26:30 pm. Underneath the logs is a 'Tests' section showing a single test run with 0 failures and 1 total. The 'Overall Code Coverage' table shows 26% coverage across all classes.

Preview :



Exercise 18

Create a Visualforce page that uses a custom controller to display a list of cases with the status of 'New'. The page must be named NewCaseList.

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/ui/common/apex/debug/ApexCSIPage`. The tabs at the top include DisqualifyTestLeads.apxt, DisqualifyTestLeads.apxc, OppView.vfp, AccountList.vfp, and NewListview.apxc. The code editor contains the following Apex class:

```
1 public class NewListview {
2     public List<Case> getNewCases(){
3         List<Case> new_cases = [Select Id, CaseNumber FROM Case WHERE Status ='New'];
4         return new_cases;
5     }
6 }
7 }
```

Below the code editor is a test result table:

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage
✓	TestRun @ 6:26:30 pm			0	1	Class: Overall Percent: 24% Lines: 24%

Visualforce Page :

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `resourceful-hawk-qv5wa7-dev-ed.trailblaze.my.salesforce.com/ui/common/apex/debug/ApexCSIPage`. The tabs at the top include DisqualifyTestLeads.apxt, DisqualifyTestLeads.apxc, OppView.vfp, AccountList.vfp, and NewListview.apxc. The code editor contains the following Visualforce page:

```
1 <apex:page controller="NewListview">
2     <apex:repeat value="{! newCases}" var="case">
3         <apex:outputLink value="/{case.Id}">
4             {!case.CaseNumber}
5         </apex:outputLink>
6     </apex:repeat>
7 </apex:page>
```

Below the code editor is a test result table:

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage
✓	Test Run					Overall 24%

Output :



References

1. Trailhead : <https://trailhead.salesforce.com/>
2. Developer Guide : https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_dev_guide.htm