



Capstone Project Report

On “Travel App and code playground”

Submitted by:

Name: Ashutosh Pattnayak

Batch Wipro Salesforce B13 (WIP-SF-13)

LMS Id: MGSA_504

Date: 28/02/2023

Table of contents

S.no	Content	Page No.
1.	Abstract	
2.	Introduction	
3.	Flow of the project	
4.	Software Requirements	
5.	Screen shots 5(a). Module - 1 5(b). Module - 2 5(c). Module - 3	
6.	Future Enhancements	
7.	References	

Abstract

Salesforce, Inc. is an American cloud-based software company headquartered in San Francisco, California. It provides customer relationship management (CRM) software and applications focused on sales, customer service, marketing automation, analytics, and application development.

Salesforce's main technologies are tools for customer management. Other products enable customers to create apps, integrate data from other systems, visualize data, and offer training courses.

Force.com applications are built using declarative tools, backed by Lightning and Apex, a proprietary Java-like programming language for Force.com, as well a Visualforce, a framework including an XML syntax typically used to generate HTML. The Force.com platform typically receives three complete releases a year. As the platform is provided as a service to its developers, every single development instance also receives all these updates.

In here we work on how to Set up the Company Profile, Configuring the user Interface, setting up Activities and Calendars, Configuring Search Settings, Setting up Chatter Groups, Mobile Access with salesforce.

Introduction

About Salesforce:-

Salesforce is a cloud-based software company that provides its customers with a platform to develop their own applications without following the tough steps that they used to follow in the legacy system. The software or application once created can be uploaded onto the cloud allowing the end-users to view them.

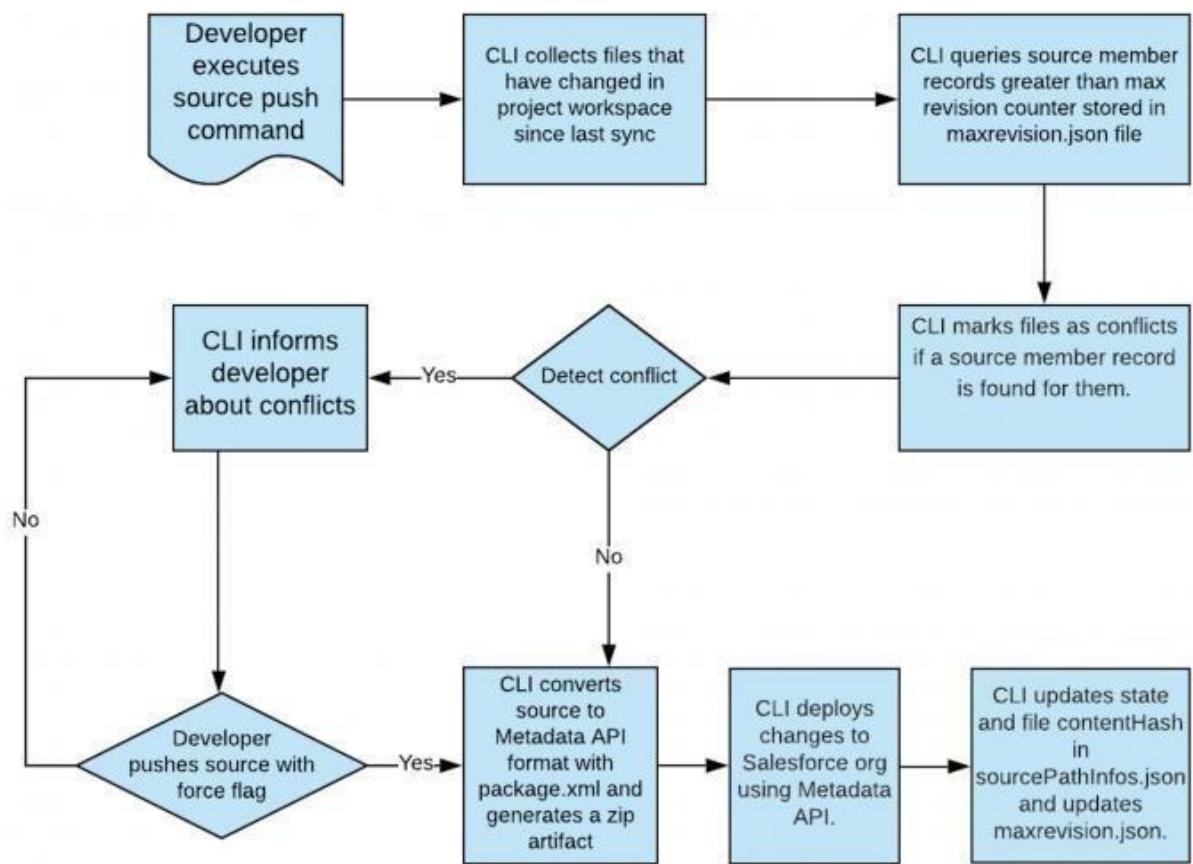
Salesforce is currently providing various software solutions and platforms for developers to create and distribute custom software/applications. Tech giants like Google, Twitter, Amazon, and Facebook are using Salesforce either in the form of SaaS or PaaS.



Salesforce developers can make an application on the cloud and share it with multiple companies across multiple domains by using Salesforce.

Talking about HR systems, every company across the globe has an HR team. Each HR team would require an HR application to store employee records. Almost all specifications for such an application would be common for all companies. So, as a developer, it would be very easy to create a Salesforce application for such specifications, post it onto the cloud, and provide it as a service to multiple clients at the same time. Maintenance of the same can be done altogether too. So basically, the problem of scalability gets eliminated.

Flow of the Project



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

OR

Requirements	
Windows	
Operating system	Windows 8.1 64-bit, Windows 8 64-bit, Windows 7 Service Pack 1 64-bit, Windows Vista Service Pack 2 64-bit
CPU	Core 2 Quad Q6600 at 2.4 GHz or AMD Phenom 9850 at 2.5 GHz
Memory	4 GB RAM
Free space	65 GB of free space
Graphics hardware	DirectX 10-compatible GPU : GeForce 9800GT 1GB or ATI Radeon HD 4870 1GB
Sound hardware	DirectX 10 compatible sound card

Screeen shots :-

Exercise 1:-

Step 1: - Create a new custom lightning App, name: **Travel App**

New Lightning App

App Details & Branding

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

App Details	App Branding
* App Name <small>1</small> <input type="text" value="Travel App"/>	Image <small>1</small>  Primary Color Hex Value <input type="color" value="#0070D2"/> #0070D2
* Developer Name <small>1</small> <input type="text" value="Travel_App"/>	
Description <small>1</small> <input type="text" value="Enter a description..."/>	Org Theme Options <input checked="" type="checkbox"/> Use the app's image and color instead of the org's custom theme

Next

New Lightning App

App Options

Navigation and Form Factor	Setup and Personalization
* Navigation Style <input checked="" type="radio"/> Standard navigation <input type="radio"/> Console navigation	Setup Experience <input checked="" type="radio"/> Setup (full set of Setup options) <input type="radio"/> Service Setup
* Supported Form Factors <input checked="" type="radio"/> Desktop and phone <input type="radio"/> Desktop <input type="radio"/> Phone	App Personalization Settings <input type="checkbox"/> Disable end user personalization of nav items in this app <input type="checkbox"/> Disable temporary tabs for items outside of this app

Back Next

New Lightning App

Available Items

[Create](#)

Selected Items

- Chatter
- Reports
- Dashboards

Back
Next

New Lightning App

Available Profiles

Selected Profiles

- System Administrator

Back
Save & Finish

Final output of step 1 -

The screenshot shows the Chatter feed within the "Travel App". The feed includes:

- A sidebar on the left with sections like "What I Follow", "To Me", "Bookmarked", "Company Highlights", "My Drafts", "STREAMS" (empty), and "RECENT GROUPS" (empty).
- The main Chatter feed area with a "Post" tab selected. A user named "TA00002" has posted "Ashutosh Pattnayak" on July 1, 2022 at 12:56 PM.
- A comment from "Eric Executive" asking, "Which Department should I associate this travel request with?"
- Below the comment, another post from "Eric Executive" stating, "Technology is the correct department".
- An "Einstein Recommendations" sidebar on the right featuring a cartoon character and the text "Eric Executive Your manager".
- A top navigation bar with tabs for "Chatter", "Reports", "Dashboards", "Departments", "Travel Approvals", "Expense Items", "Customers", and "Billings".

Step 2 : - Create a Department custom object.

Setup->Object Manager->Create Custom object (From top left dropdown button)

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected in the top navigation bar. The main title is 'New Custom Object'. The 'Custom Object Information' section contains fields for 'Label' (Department) and 'Plural Label' (Departments). Below this, there's a checkbox for 'Starts with vowel sound'. The 'Object Name' field is also labeled 'Department'. A 'Description' text area is present. Under 'Context-Sensitive Help Setting', the radio button for 'Open the standard Salesforce.com Help & Training window' is selected. The 'Content Name' dropdown is set to 'None'. At the bottom, there's a section titled 'Enter Record Name Label and Format' with a note about record names appearing in various layouts. The 'Record Name' field is set to 'Department Name' and the 'Data Type' is 'Text'.

Step 3 :-Create the following Custom Field in Department Object.

- Department Code, Text, Length = 10, Required, Select Unique & Case sensitive

SETUP > OBJECT MANAGER
Department

Details

Fields & Relationships

Field Label: Department Code

Length: 10

Field Name: Department_Code

Description:

Help Text:

Required: Always require a value in this field in order to save a record

Unique: Do not allow duplicate values

External ID: Set this field as the unique record identifier from an external system

Auto add to custom report type: Add this field to existing custom report types that contain this entity

Default Value: Show Formula Editor

Use formula syntax: Enclose text and picklist value API names in double quotes ("the_text"), include numbers without quotes : (12), show percentages as decimals (.10), and express date calculations in the standard format (Today) + 7). To reference a field from a Custom Metadata type record use: \${CustomMetadata.Type__md.RecordName.FieldName__c}

(b). Location, Picklist, Value: Kolkata, Delhi.

SETUP > OBJECT MANAGER
Department

Details

Fields & Relationships

Step 2. Enter the details

Field Label: Location

Value: Use global picklist value set
 Enter values, with each value separated by a new line

Kolkata
Delhi

Display values alphabetically, not in the order entered
Use first value as default value
 Restrict picklist to the values defined in the value set

Field Name: Location

Description:

Help Text:

(c). Department Type, Picklist, Values: Banking, Finance, Education, Energy, IT.

Department
New Custom Field

Step 2. Enter the details

Field Label: Department Type

Values:

- Use global picklist value set
- Enter values, with each value separated by a new line

Banking
Finance
Education
Energy
IT

Display values alphabetically, not in the order entered

Use first value as default value

Restrict picklist to the values defined in the value set

Field Name: Department_Type

Description:

Help Text:

(d). Create Field Dependency, Controlling field =Location, Dependent field = Department Type

Controlling Field: Location
Dependent Field: Department Type

Instructions:

- 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.

Legend:

- Excluded Value
- Included Value

		Kolkata	Delhi
Location:	Banking	Banking	
Department Type:	Finance	Finance	
	Education	Education	
	Energy	Energy	
	IT	IT	

Step 4 : - Create a Travel Approval custom object.

Setup->Object Manager->Create Custom object (From top left dropdown button)

The screenshot shows the 'Custom Object Definition Edit' screen in the Salesforce Setup interface. At the top, there's a message bar stating: 'Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles.' Below this are tabs for 'Save', 'Save & New', and 'Cancel'. The main section is titled 'Custom Object Information' and contains the following fields:

- Label:** Travel Approval (with Example: Account)
- Plural Label:** Travel Approvals (with Example: Accounts)
- Starts with vowel sound:**
- Object Name:** Travel_Approval (with Example: Account)
- Description:** (Empty text area)
- Context-Sensitive Help Setting:** Open the standard Salesforce.com Help & Training window
 Open a window using a Visualforce page
- Content Name:** -None-

At the bottom, there's a section for 'Enter Record Name Label and Format' with the following fields:

- Record Name:** Travel Approval # (with Example: Account Name)
- Data Type:** Auto Number

Now, Create the following Custom Field in Travel Approval Object.

- 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.
- Save

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Department	Department__c	Lookup(Department)		✓
Destination State	Destination_State__c	Text(2)		
Last Modified By	LastModifiedById	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		

Step 5 : - Import the data store in Department.csv by using “Data Import wizard” tool.

5(a) -

Import your Data into Salesforce
You can import up to 50,000 records at a time.

What kind of data are you importing?

Standard objects Custom objects

Departments

Travel Approvals

What do you want to do?

Add new records

Match by:
-None-

Which User field in your file designates record owners?
-None-

Trigger workflow rules and processes?
 Trigger workflow rules and processes for new and updated records.

Update existing records

Where is your data located?

Drag CSV file here to upload

CSV

File Departments.csv

Character Code
ISO-8859-1 (General US & Western European, ISO-LATIN-1)

Values Separated By
Comma

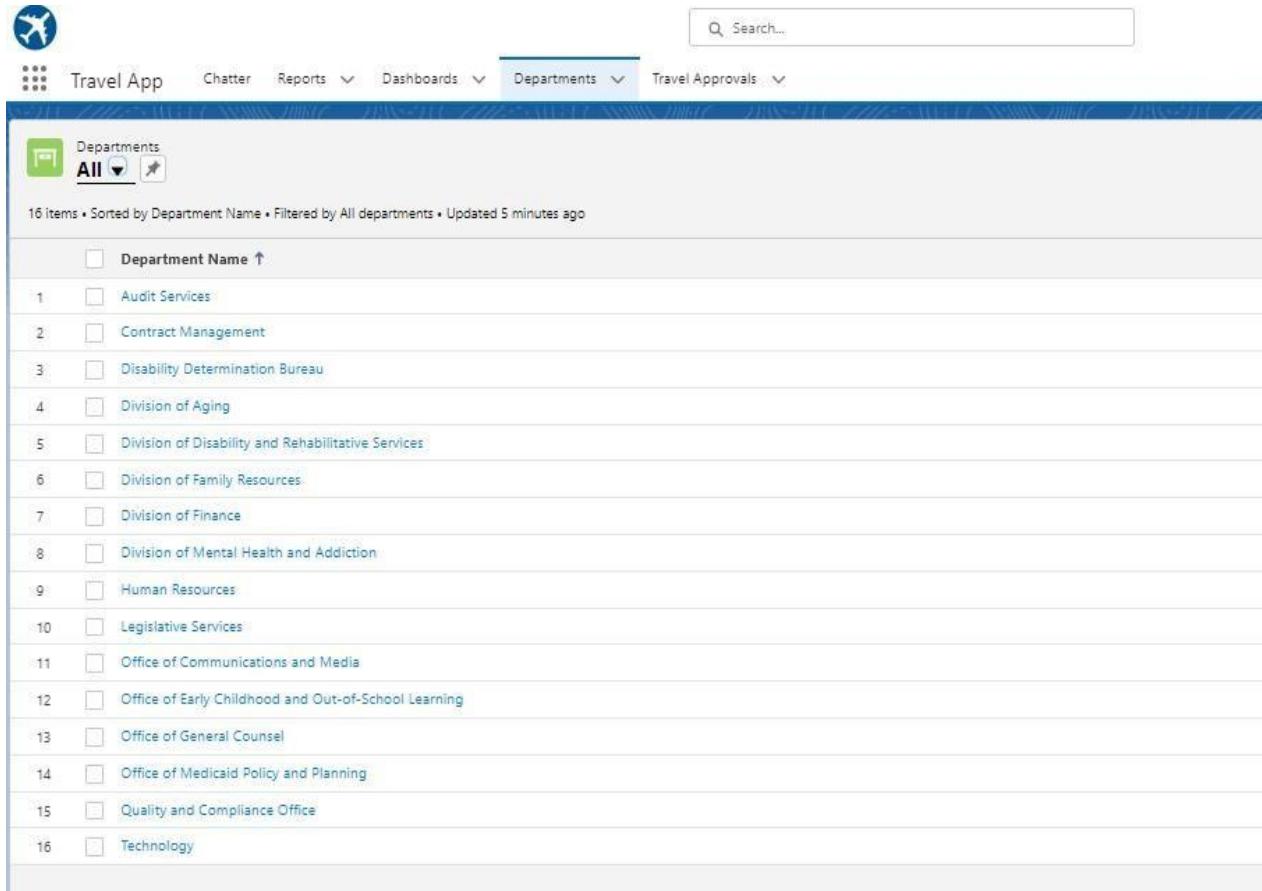
Cancel Previous Next

5(b) -

Edit Field Mapping: Departments
Your file has been auto-mapped to existing Salesforce fields, but you can edit the mappings if you wish. Unmapped fields will not be imported.

Edit	Mapped Salesforce Object	CSV Header	Example	Example	Example
Change	Department Name	Department Name	Audit Services	Contract Manager	Division of Finance
Change	Department Code	Department Code	405-01	405-02	405-03

-Test the App, till yet what we have done.



The screenshot shows a Salesforce application interface titled "Travel App". The top navigation bar includes links for "Chatter", "Reports", "Dashboards", "Departments", and "Travel Approvals". A search bar is located in the top right corner. The main content area is titled "Departments" and shows a list of 16 items, all filtered by "All departments". The list is sorted by "Department Name" in ascending order. Each item is numbered from 1 to 16 and includes a checkbox and a link to the department's detail page. The departments listed are: Audit Services, Contract Management, Disability Determination Bureau, Division of Aging, Division of Disability and Rehabilitative Services, Division of Family Resources, Division of Finance, Division of Mental Health and Addiction, Human Resources, Legislative Services, Office of Communications and Media, Office of Early Childhood and Out-of-School Learning, Office of General Counsel, Office of Medicaid Policy and Planning, Quality and Compliance Office, and Technology.

Rank	Department Name
1	Audit Services
2	Contract Management
3	Disability Determination Bureau
4	Division of Aging
5	Division of Disability and Rehabilitative Services
6	Division of Family Resources
7	Division of Finance
8	Division of Mental Health and Addiction
9	Human Resources
10	Legislative Services
11	Office of Communications and Media
12	Office of Early Childhood and Out-of-School Learning
13	Office of General Counsel
14	Office of Medicaid Policy and Planning
15	Quality and Compliance Office
16	Technology

See the data is imported Successfully.

Note :- Before Importing the data make sure you clean the data otherwise some data will not be imported successfully.

Exercise 2: -

Step 1: - Create Travel Approval Record.

This screenshot shows a Salesforce record page for a Travel Approval. The record ID is TA00001. The page has a 'Details' tab selected, showing fields like Travel Approval #, Status (Approved), Out Of State (checked), Destination State (US), Total Expenses (\$870.00), and a green status indicator. A 'Trip Info' section is expanded, showing Purpose Of Trip (Businesstrip Trip to USA). The right sidebar displays the Chatter feed and activity history.

This screenshot shows the same Travel Approval record (TA00001) from a slightly different perspective. The 'Details' tab is still selected, but the 'Trip Info' section is collapsed. The right sidebar shows the Chatter feed and activity history.

Step 2 : - Create a Expense Items custom object.

Setup->Object Manager->Create Custom object (From top left dropdown button)



Step 3 :-Create the following custom fields on Expense Item Object.

- Amount, Length = 16, Type = Currency, Decimal = 2, Required = True.
- Expense Type, Type = Picklist, Values = Airfare, Hotel, Rental Car, Meals, Other, Required = True.
- Travel Approval, Type = Master-Detail, Related To – Travel Approval.

The screenshot shows the Salesforce Object Manager interface for the 'Expense Item' object. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, etc. The main area displays the 'Fields & Relationships' section with the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
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 the Expense Items .

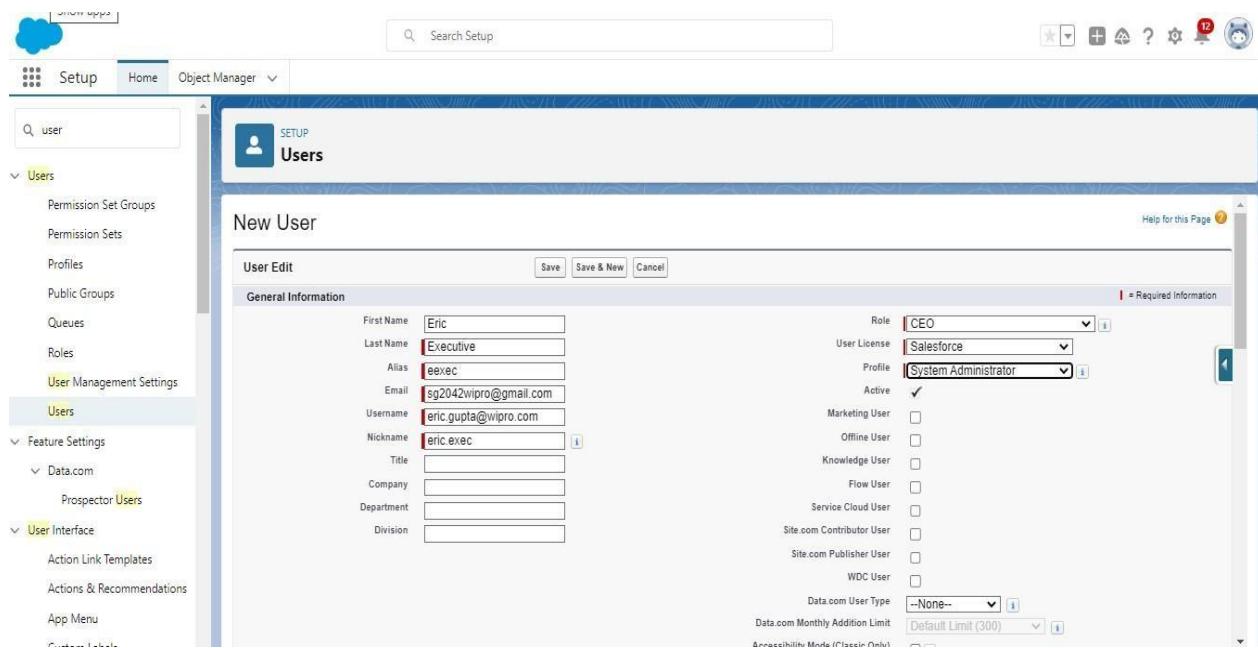
- Amount = 870, Expense Type = “Hotel”. Save.
- Amount = “450”, Expense Type = “Airfare” Save.

The screenshot shows the Salesforce Travel App interface. A new 'Travel Approval' record is being created with the ID TA00001. The 'Related' tab is selected, showing a list of 'Expense Items (2)'. Two expense items are listed: E-00001 and E-00002. The 'Activity' tab is also visible on the right side of the screen.

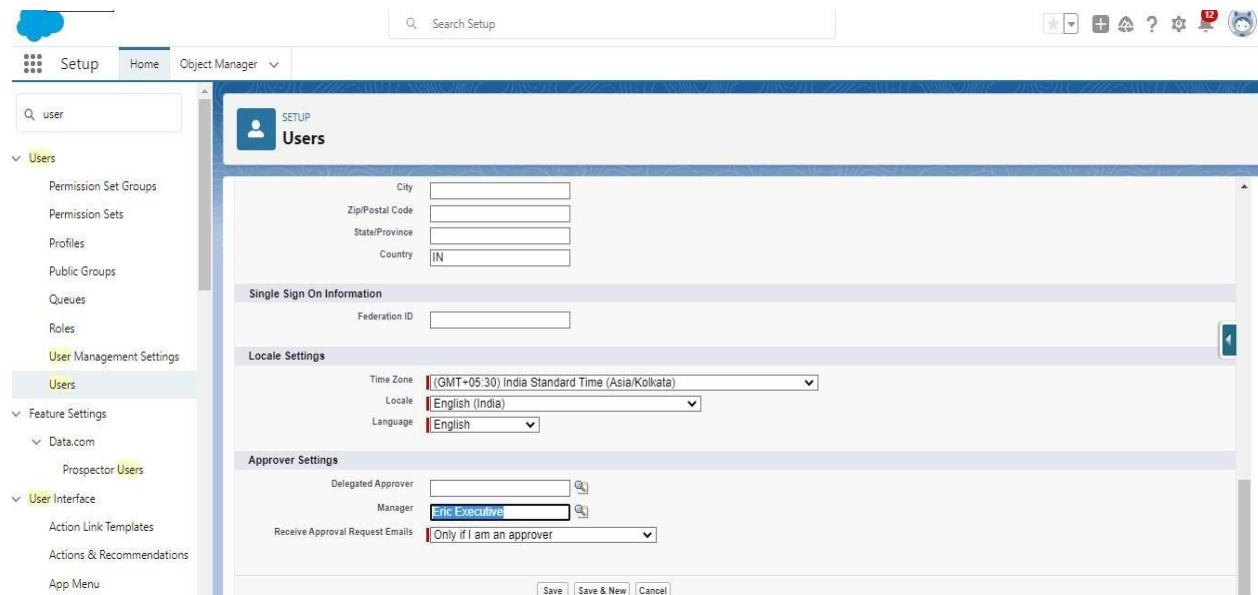
Step 5 :- Create a user.

- First Name = “Eric”, Last Name = “Executive”, Email = “Use your own email”,

- Username Name = “Choose a Unique username”
- Role = “CEO”
- License = Salesforce.
- Profile = System Administrator.
- Save.



Step 6:- Add user Eric Executive as your manager as shown in the screen shot.



Step 7:- Customize the Travel Approval Default search layout as shown in the screen shot & Save.

The screenshot shows the Salesforce Object Manager interface for the 'Travel Approval' object. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, and Search Layouts, with 'Search Layouts' currently selected. The main area is titled 'Edit Search Layout' and displays the 'Travel Approval Search Results'. It shows two columns: 'Available Fields' (Record ID, Out Of State, Owner Alias, Owner First Name, Owner Last Name, Created By Alias, Created By, Created Date, Last Modified By Alias, Last Modified By, Last Modified Date) and 'Selected Fields' (Travel Approval #, Purpose Of Trip, Department, Status, Destination State, Trip Start Date, Trip End Date). A checkbox for 'Override the search result column customizations for all users' is present. Below are sections for 'Standard Buttons' (none listed) and 'Custom Buttons' (none listed).

Step 8:- Select fields to display in the Travel Approval “All” List view, as shown in the screen shot & Save.

The screenshot shows the Salesforce list view for 'Travel Approvals' with the filter set to 'All'. The top navigation bar includes 'Travel App', 'Chatter', 'Reports', 'Dashboards', 'Departments', and 'Travel Approvals'. The list view shows one item: '1 TA0001'. A modal window titled 'Select Fields to Display' is open over the list view. It contains two columns: 'Available Fields' (Out Of State, Owner Alias, Owner First Name, Owner Last Name, Purpose Of Trip, Record ID) and 'Visible Fields' (Travel Approval #, Department, Created By, Status, Trip Start Date, Trip End Date). The 'Travel Approval #' field is highlighted with a blue border. At the bottom of the modal are 'Cancel' and 'Save' buttons.

Step 9 :- Create Travel approval custom List View “Open Out of State Travel Requests” as shown in the screen shot & Save.

- All users should be able to see this list view.

The screenshot shows a Salesforce application interface titled 'Travel Approvals'. The top navigation bar includes links for 'Travel App', 'Chatter', 'Reports', 'Dashboards', 'Departments', and 'Travel Approvals'. A search bar is at the top right. The main area displays a list titled 'Open Out of State Travel Requests'. It shows one item, TA00001, with a checkbox next to it. To the right of the list are buttons for 'New', 'Import', 'Change Owner', and 'Printable View'. Below the list is a 'Search this list...' field and a set of filter and sorting tools. On the far right, there are 'Cancel' and 'Save' buttons. A sidebar on the right contains a 'Filter by Owner' section with 'My travel approvals' selected, and a 'Matching all of these filters' section containing two conditions: 'Out Of State*' equals True and 'Status*' not equal to Approved, Rejected. Buttons for 'Add Filter' and 'Remove All' are also present.

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.

Travel App Chatter Reports Dashboards Departments Travel Approvals

Travel Approvals Open Out of State Travel Requests

1 item • Sorted by Travel Approval # • Filtered by My travel approvals - Status

Travel Approval # ↑
1 TA00001

Select Fields to Display

Available Fields: Owner Alias, Owner First Name, Owner Last Name, Purpose Of Trip, Record ID, Travel Approval #

Visible Fields: Department, Created By, Status, Destination State, Trip Start Date, Trip End Date

Filters:

- Filter by Owner: My travel approvals
- Matching all of these filters:
 - Out Of State equals True
 - Status not equal to Approved, Rejected

Add Filter Remove All Add Filter Logic

New Import Change Owner Printable View

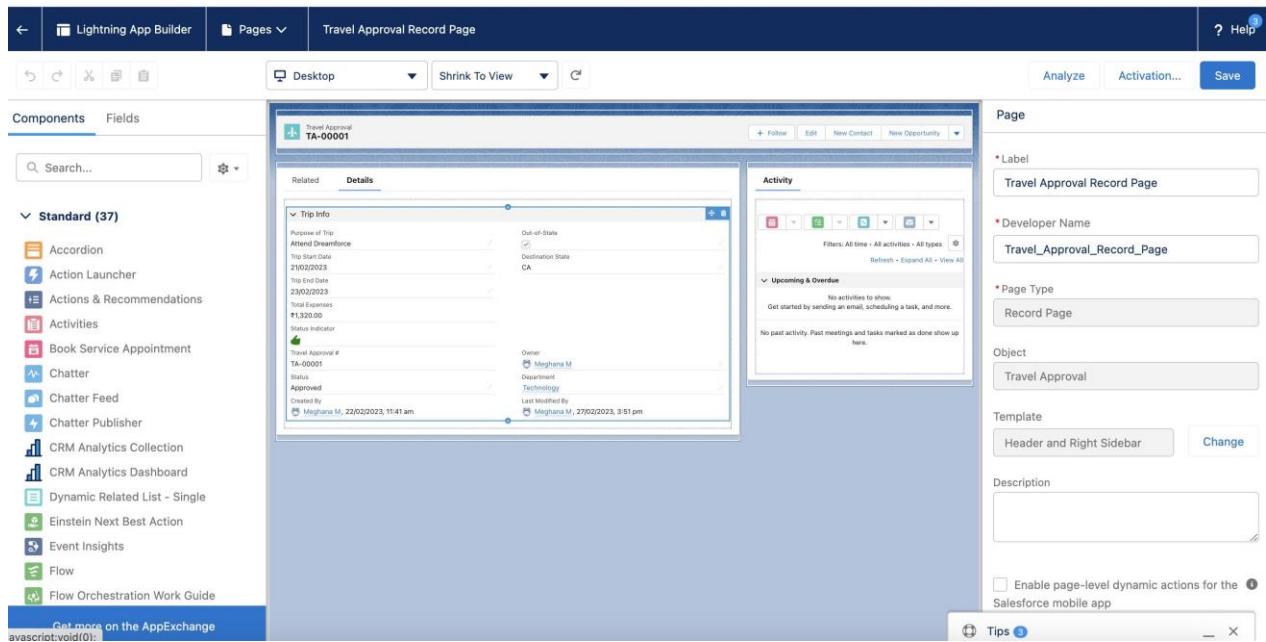
Cancel Save

This screenshot shows a Salesforce interface for managing travel approvals. A modal window titled 'Select Fields to Display' is open, allowing the user to choose which fields are visible in the list. The 'Available Fields' section contains: Owner Alias, Owner First Name, Owner Last Name, Purpose Of Trip, Record ID, and Travel Approval #. The 'Visible Fields' section contains: Department, Created By, Status, Destination State, Trip Start Date, and Trip End Date. Below the modal, the main list shows one item: 'TA00001'. The list is sorted by Travel Approval # and filtered by 'My travel approvals - Status'. The 'Filters' sidebar on the right shows two filters applied: 'Out Of State equals True' and 'Status not equal to Approved, Rejected'. Navigation and action buttons like New, Import, Change Owner, and Printable View are visible at the top.

Step 11 :-

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

- In the travel app, click travel approval tab and open TA-00001
- Then click Gear button and select “Edit Button”. It will open as shown as screenshots.



(a) :- Travel Approval Page Layout should look as shown in the screen shot.

Note: Add a section to the page layout called “Trip Info” and add the fields as shown. Click Save.

Step 12 :-

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

(b)- The Travel Approval App should look as shown in the screenshot.

Step 13 :-

The screenshot shows the Salesforce Travel App interface. At the top, there's a navigation bar with links for Travel App, Chatter, Reports, Dashboards, Departments, and Travel Approvals. Below the navigation is a search bar. The main content area is titled "Travel Approval TA-00001 (a)". It contains sections for "Notes & Attachments (0)", "Expense Items (2)", and "Approval History (3)". The "Expense Items" section lists two items: E-00001 (Airfare) and E-00002 (Hotel). The "Approval History" section shows three entries. To the right, a sidebar titled "Activity" lists recent posts from users like Meghana M and others. It also includes options for "DISPLAY DENSITY" (Compact selected), "OPTIONS" (Switch to Salesforce Classic, Add Username), and a "Get started" link.

Enable Chatter on the Travel Approval Object.

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

The screenshot shows the Salesforce Setup interface under the "Chatter" section. The left sidebar has a search bar and a list of feature settings, with "Feed Tracking" selected. The main content area is titled "Feed Tracking" and shows the configuration for the "Travel Approval" object. It lists fields like Department, Out Of State, Purpose Of Trip, Travel Approval #, Trip Start Date, Destination State, Owner, Status, and Trip End Date. There are checkboxes for "Enable Feed Tracking" and "Restore Defaults". A note says "You can select up to 9 fields." Below this, there's a section for "Fields in travel approvals" with a "Save" button. A note at the bottom says "You can also display feed activity for related objects." and "All Related Objects" is checked.

(b) Test Collaboration :-

- 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.”

Step 14 :-

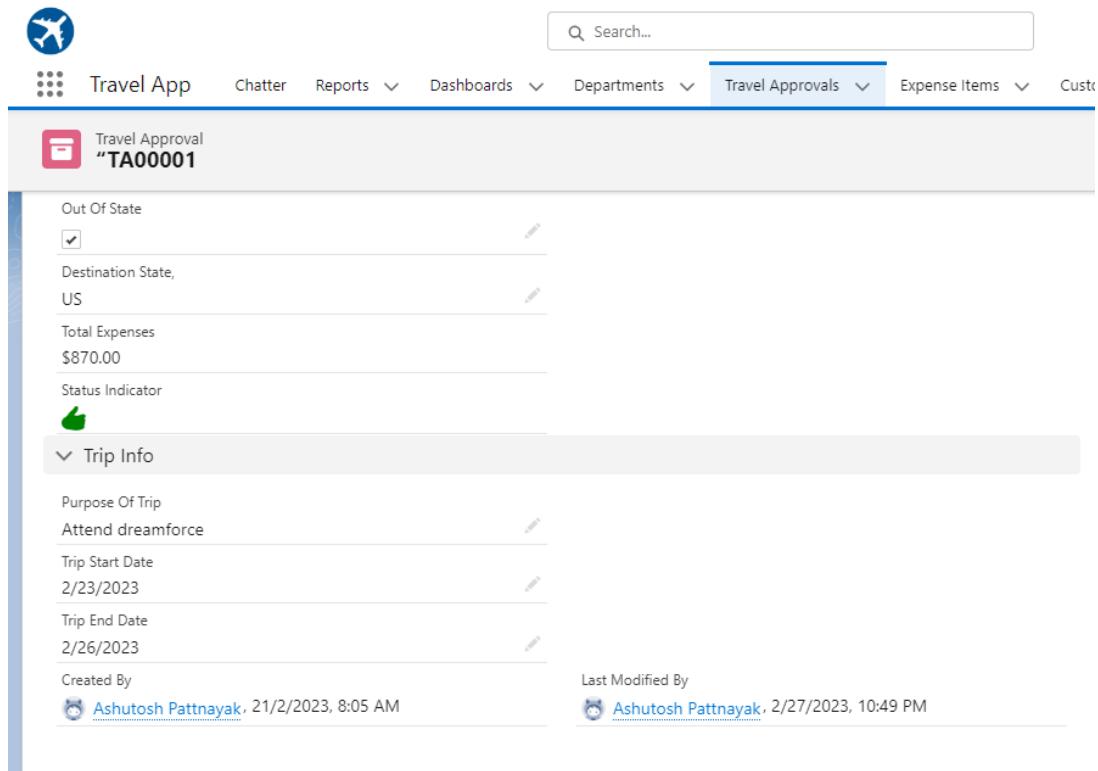
The screenshot shows the Salesforce Chatter interface within the 'Travel App' tab. On the left sidebar, under 'Company Highlights', the 'Travel App' icon is highlighted. The main Chatter feed displays a post from user TAA00002 - Ashutosh Pattnayak, dated July 1, 2022 at 12:56 PM. The post asks, '@Eric Executive Which Department should I associate this travel request with?'. Below the post, Eric Executive comments, 'Technology is the correct department'. The Einstein Recommendations sidebar on the right lists three suggestions:

- Eric Executive (Your manager) - + Follow Skip
- Integration User (Joined in the last week) - + Follow Skip
- Security User (Joined in the last week) - + Follow Skip

At the top of the page, there is a navigation bar with links to Chatter, Reports, Dashboards, Departments, Travel Approvals, Expense Items, Customers, and Billings. A search bar and various icons are also present.

Test the App , Let see how its look like

The Travel Approval App should look as shown in the screenshot.



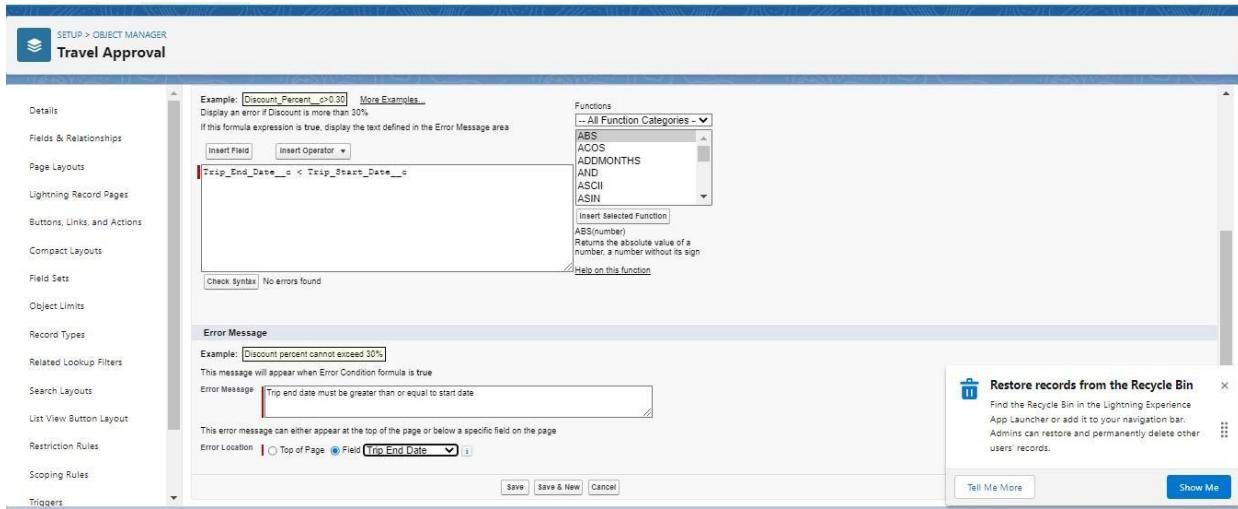
After completing the exercise 2 you will see the user can post their query by using chatter and other can reply to that query. Now, Here Our module 1 is completed lets move to our next module 2.

Module – 2

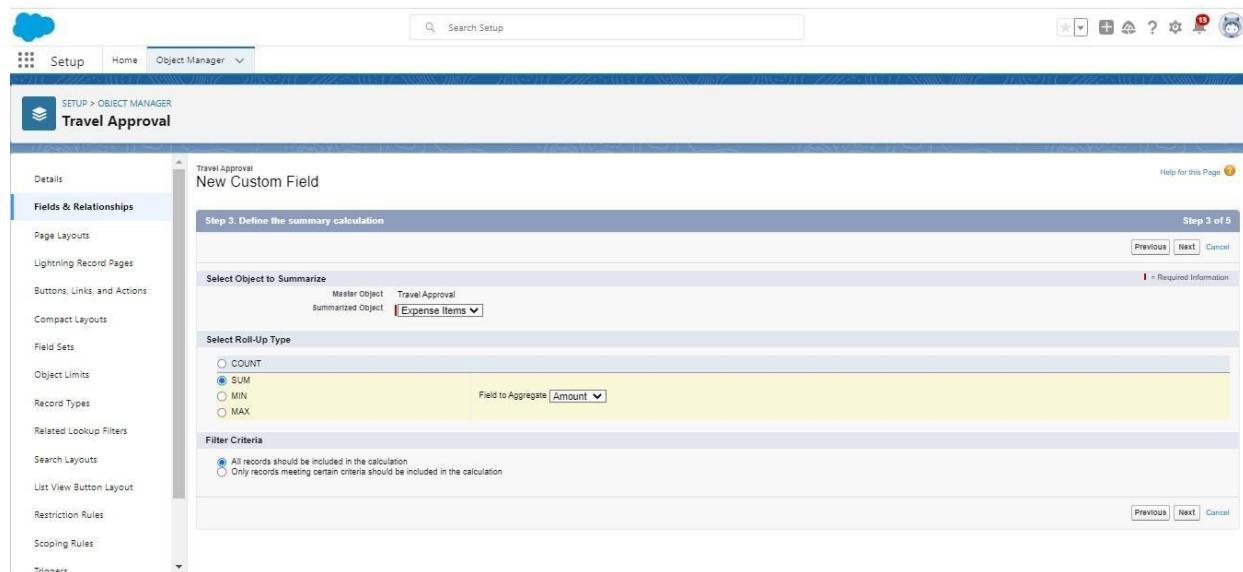
Exercise 1: -

Step 1:- Create Validation Rule.

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

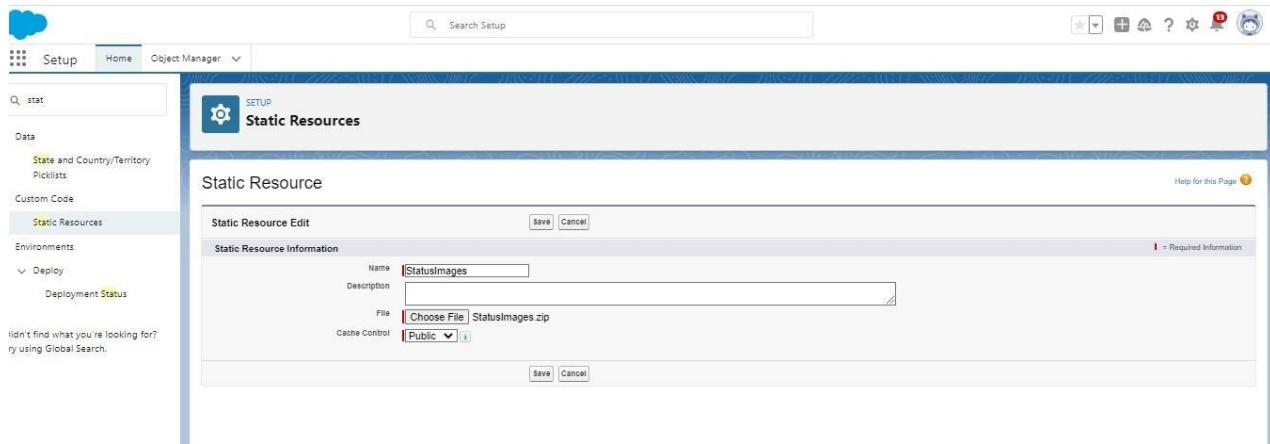


Step 2 :- Create a Roll-Up Summary Field on Travel Approval object.



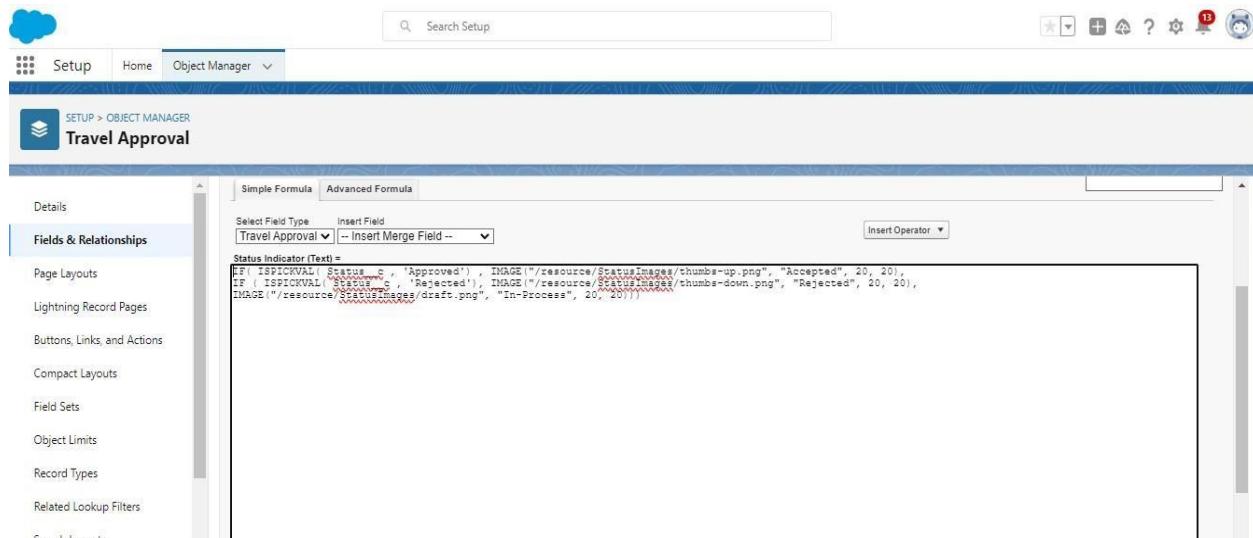
Step 3 :- Business Logic: Create a field that shows a visual indicator based on the value of the Status field

- Setup | Custom Code | Static Resource | New



Step 4 :- Create a Formula field on the Travel Approval object to show an image based on the Status field.

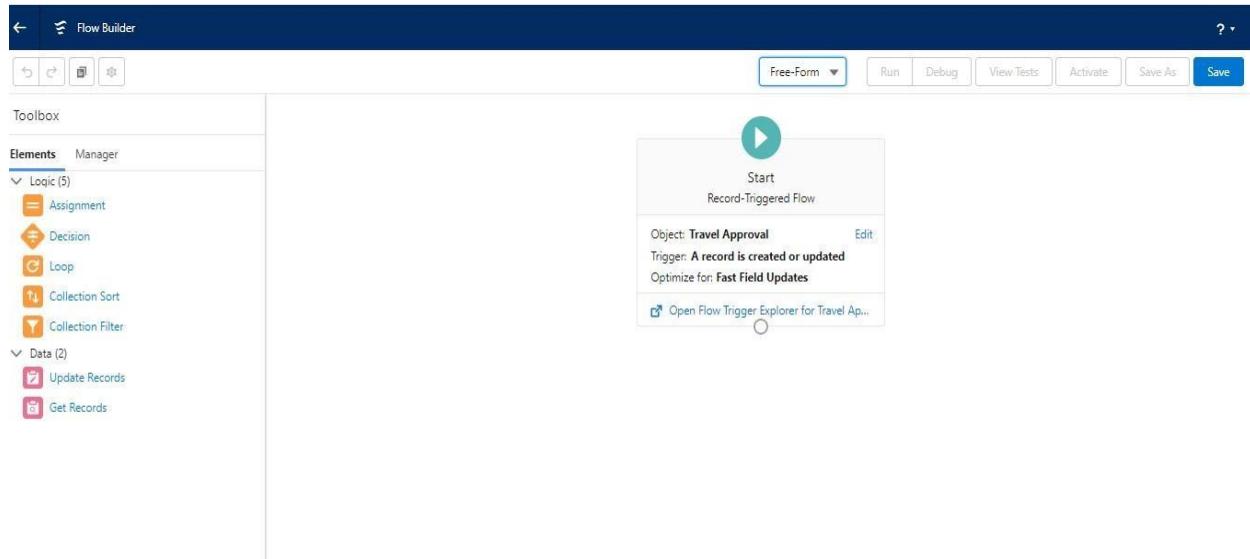
- Field Label: Status Indicator
- Formula Return Type = Text
- Formula: IF(ISPICKVAL(Status c , 'Approved') ,
IMAGE("/resource/StatusImages/thumbs-up.png", "Accepted", 20, 20),
IF (ISPICKVAL(Status c , 'Rejected'),
IMAGE("/resource/StatusImages/thumbs-down.png", "Rejected", 20, 20),
IMAGE("/resource/StatusImages/draft.png", "In-Process", 20, 20)))
- Save



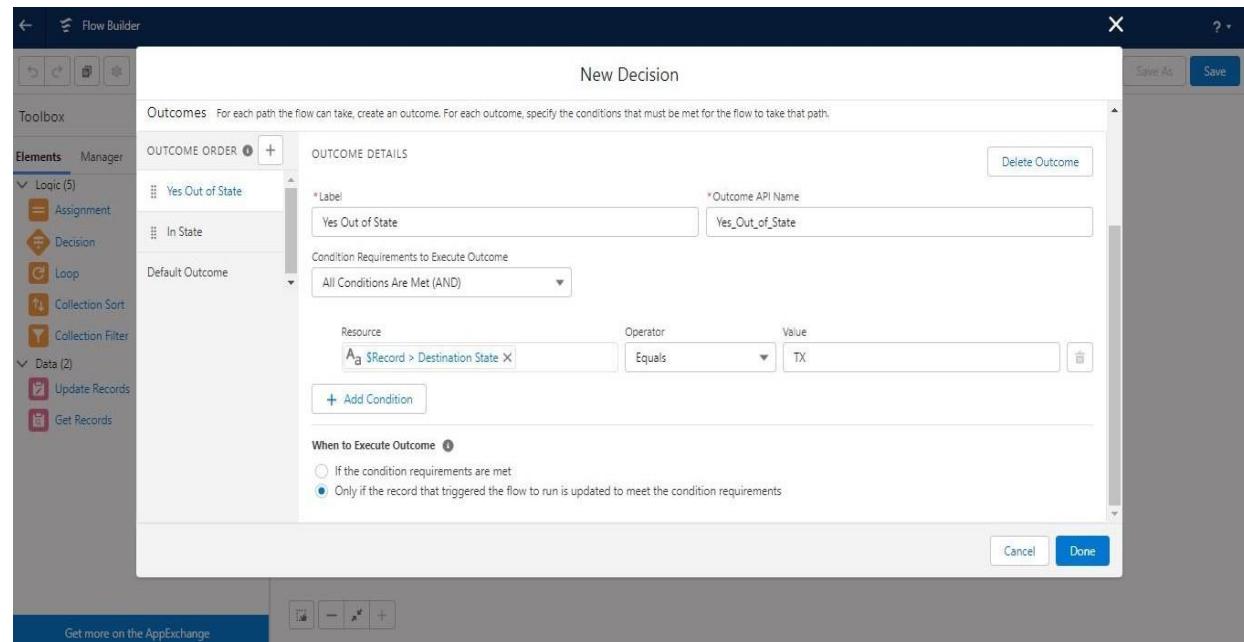
Step 5 :-

(a).Create a Record – Triggered Flow.

- Flow should look like this:

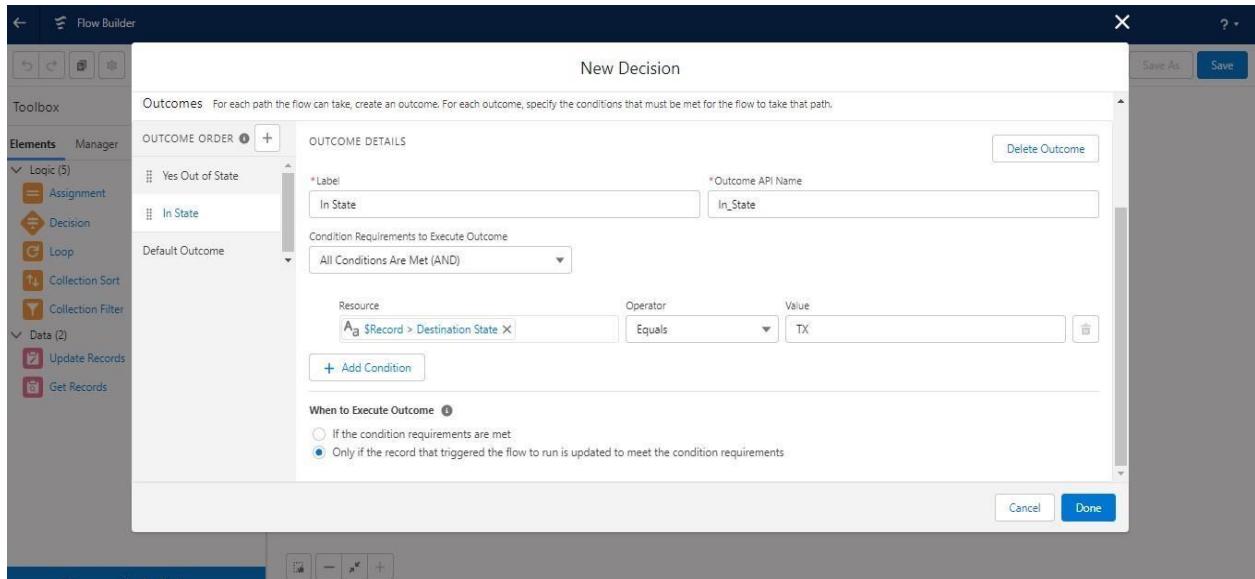


(b). Add a Decision Element to the Flow.

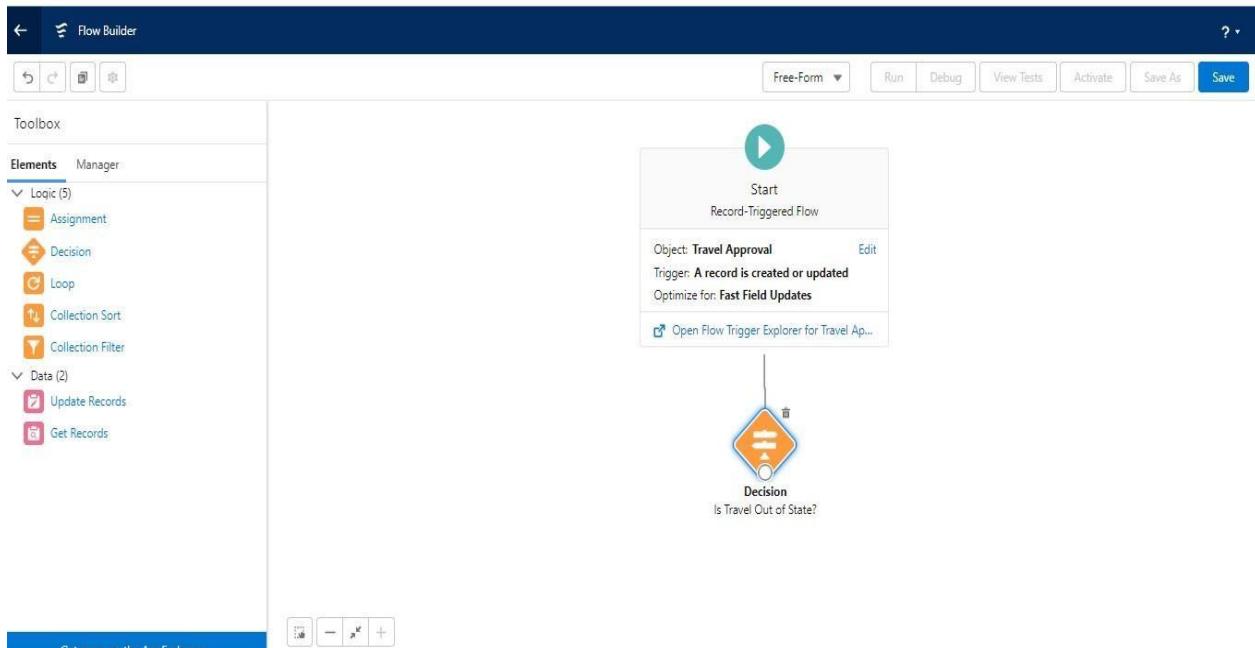


(c). Next to Outcome Order click the + button to add another outcome.

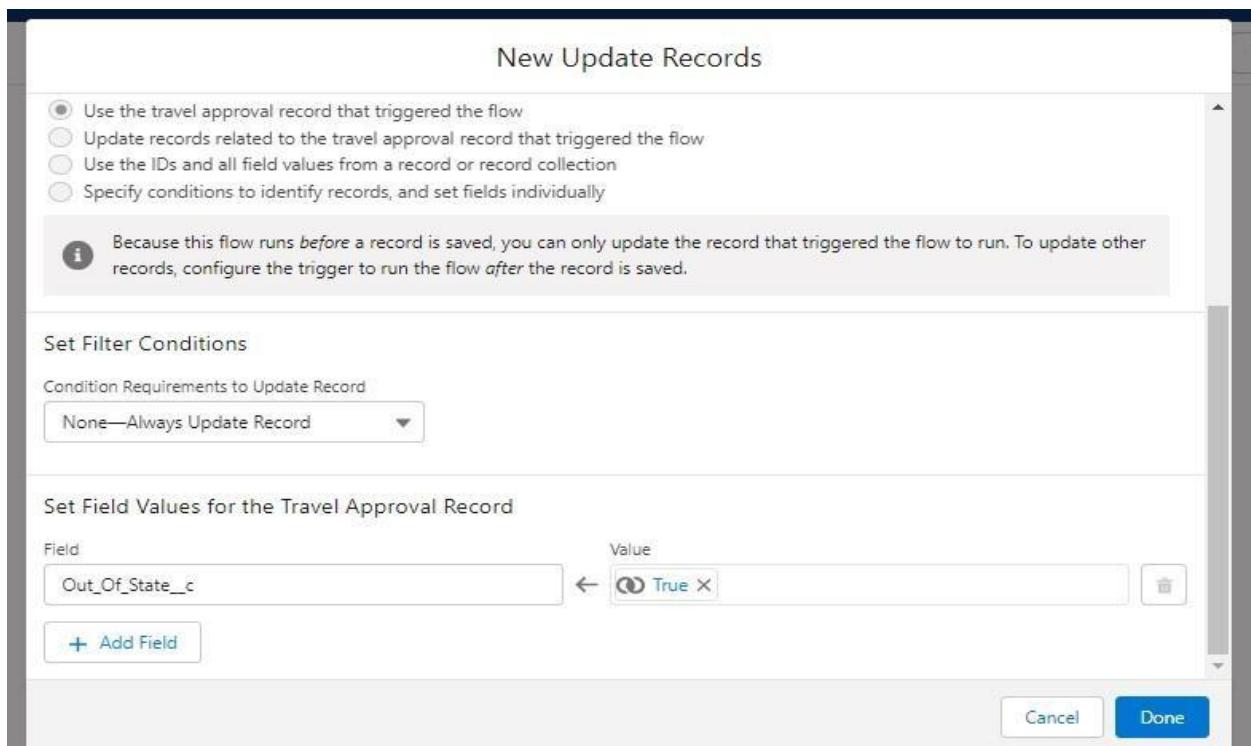
- Flow should look like this:



(d). Connect the Start Flow element to the Decision element.

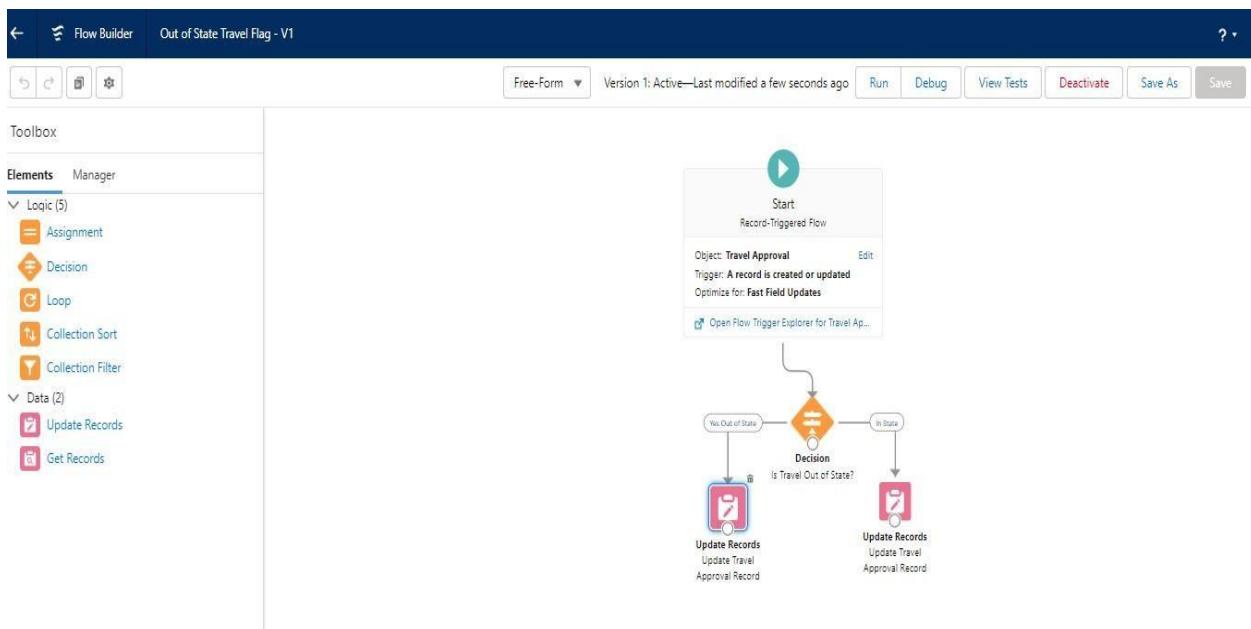


(e). Create an Action for the Flow Using Update Records Elements.



(f). 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:



Step 6 :-

(a). Create an Approval Process to send Travel approvals to Manager or Travel coordinator.

The screenshot shows the Salesforce Setup interface. On the left, there's a sidebar with various links like Setup Home, Service Setup Assistant, Multi-Factor Authentication Assistant, Release Updates, Lightning Experience Transition Assistant, Salesforce Mobile App, Lightning Usage, Optimizer, Administration (with sub-links for Users, Data, Email), and Platform Tools (with sub-links for Apps, Feature Settings, Slack, and Workflow Services). The main area is titled "Approval Processes" and shows a specific process named "Travel Approval: Travel Approval Request". The process definition details include:

Setting	Value
Process Name	Travel Approval Request
Unique Name	Travel_Approval_Request
Description	Travel Approval: Total Expenses GREATER THAN 0
Record Editability	Administrator ONLY
Approval Assignment Email Template	Initial Submitters
Created By	Meghana M, 22/02/2023, 7:56 pm
Active	<input checked="" type="checkbox"/>
Next Automated Approver Determined By	Manager of Record Submitter
Allow Submitters to Recall Approval Requests	<input type="checkbox"/>
Modified By	Meghana M, 22/02/2023, 8:52 pm

Below this, there are sections for "Initial Submission Actions" (with an "Add Existing" button) and "Approval Steps" (with columns for Action, Step Number, Name, Description, Criteria, Assigned Approver, and Reject Behavior).

(b). Test The Approval Process

- Create few Travel Approval records and Submit for Approval.
- Login as Eric, approve and reject the records randomly as shown in the screen shot:

Exercise 2: -

Step 1 :- Use Data Import Wizard to import Travel Approval records.

- Test The App
- Make sure all the Travel Approval records are successfully imported, check the screenshot:

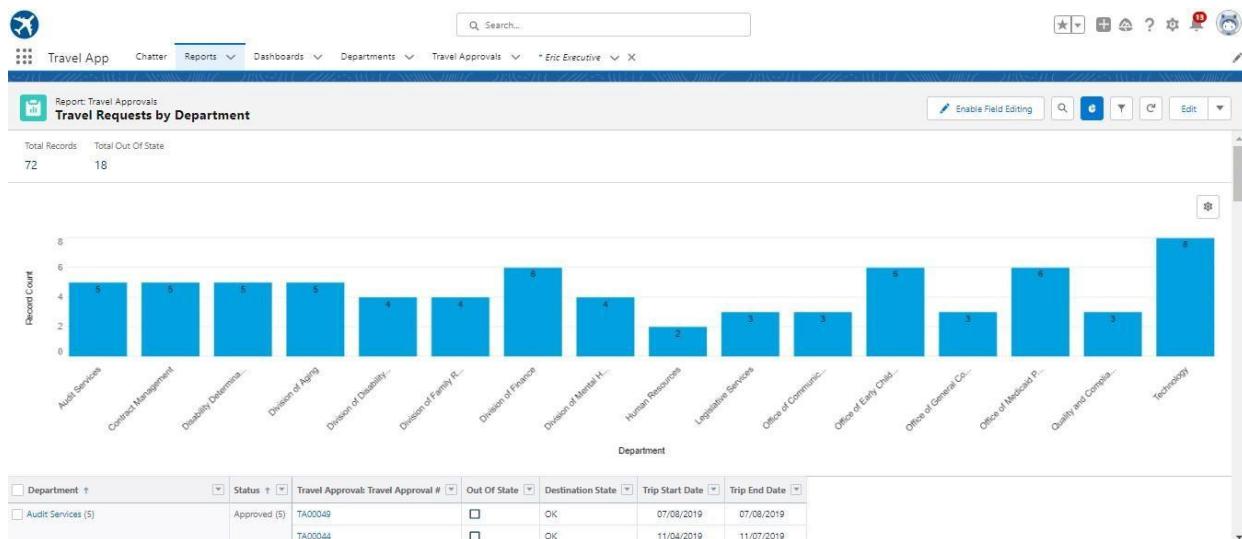
The screenshot shows a Salesforce page titled "Travel Approvals". At the top, there's a search bar and a toolbar with various icons. Below the header, a message says "50+ items • Sorted by Travel Approval # • Filtered by All travel approvals • Updated a few seconds ago". A table lists 15 travel approval records, each with a checkbox, ID, department, created by, status, and trip dates. The columns are: Travel Approval #, Department, Created By, Status, Trip Start Date, and Trip End Date. The data includes entries like TA00001 (Technology, Draft), TA00002 (Disability Determination Bureau, Approved), and TA00015 (Division of Disabilities and Rehabilitation Services, Approved).

	Travel Approval #	Department	Created By	Status	Trip Start Date	Trip End Date
1	TA00001	Technology	Sagar Gupta	Draft	28/02/2023	05/03/2023
2	TA00002	Disability Determination Bureau	Sagar Gupta	Approved	10/01/2019	10/01/2019
3	TA00003	Division of Disability and Rehabilitative Services	Sagar Gupta	Rejected	04/03/2019	04/06/2019
4	TA00004	Division of Finance	Sagar Gupta	Rejected	03/09/2019	03/09/2019
5	TA00005	Division of Aging	Sagar Gupta	Approved	11/05/2019	11/11/2019
6	TA00006	Technology	Sagar Gupta	Approved	03/06/2019	03/09/2019
7	TA00007	Division of Disability and Rehabilitative Services	Sagar Gupta	Approved	11/06/2019	11/12/2019
8	TA00008	Disability Determination Bureau	Sagar Gupta	Rejected	03/06/2019	03/07/2019
9	TA00009	Contract Management	Sagar Gupta	Approved	05/11/2019	05/11/2019
10	TA00010	Division of Disability and Rehabilitative Services	Sagar Gupta	Approved	07/07/2019	07/07/2019
11	TA00011	Office of Communications and Media	Sagar Gupta	Approved	06/02/2019	06/12/2019
12	TA00012	Office of Medicaid Policy and Planning	Sagar Gupta	Approved	11/05/2019	11/11/2019
13	TA00013	Division of Family Resources	Sagar Gupta	Approved	04/01/2019	04/01/2019
14	TA00014	Human Resources	Sagar Gupta	Approved	07/09/2019	07/11/2019
15	TA00015	Division of Disabilities and Rehabilitation Services	Sagar Gupta	Approved	10/11/2019	10/12/2019

Step 2 :- Create a Travel Requests by Department Report. Test

The Report

- Report might look as per the screen shot:



Step 3 :- Create a Travel Requests by Month Report

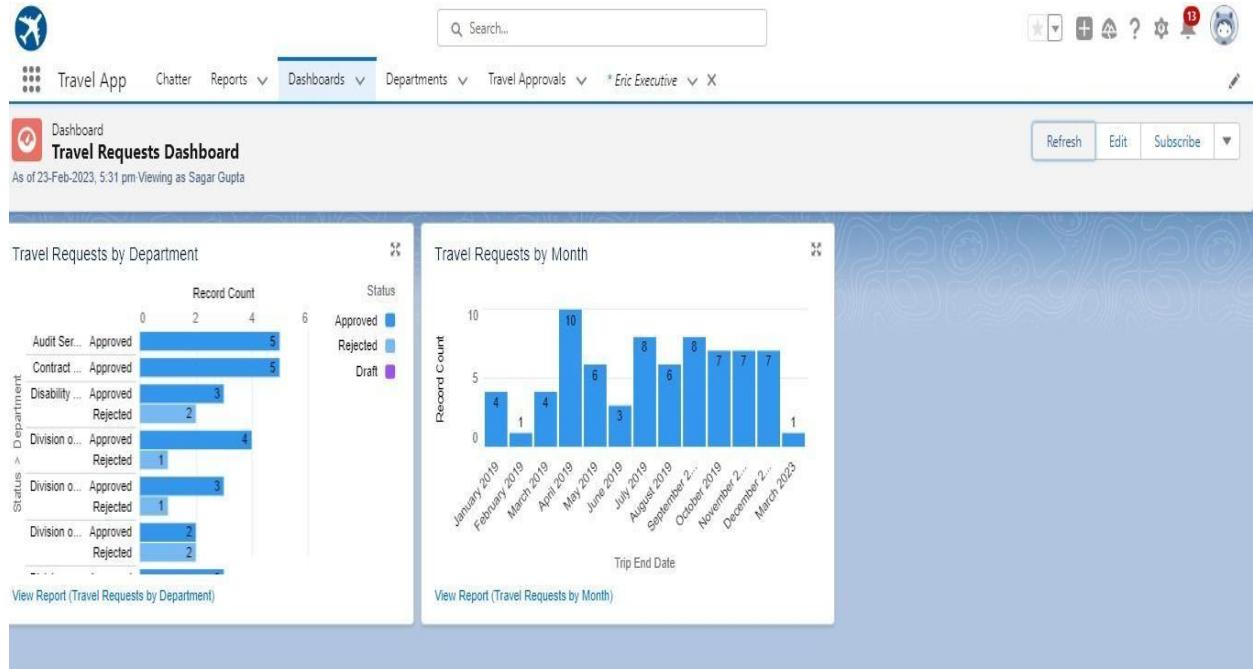
- Test The Report
- Report might look as per the screen shot:

The screenshot shows a Salesforce report titled "Report: Travel Approvals" with the specific view "Travel Requests by Month". The report displays 72 total records. The data is organized by month (January 2019, February 2019, March 2019) and then by individual travel requests. Each request includes fields such as Trip End Date, Out Of State, Travel Approval: Travel Approval #, Department, Status, Destination State, and Trip Start Date. Subtotals are provided for each month and for each travel approval number.

Trip End Date ↑	Out Of State ↑	Travel Approval: Travel Approval #	Department	Status	Destination State	Trip Start Date
January 2019 (4)	□ (1)	TA00002	Disability Determination Bureau	Approved	OK	10/01/2019
	Subtotal					
	□ (3)	TA00039	Division of Finance	Approved	TX	10/01/2019
		TA00072	Office of General Counsel	Approved	TX	04/01/2019
		TA00013	Division of Family Resources	Approved	TX	04/01/2019
	Subtotal					
February 2019 (1)	□ (1)	TA00062	Division of Family Resources	Approved	FL	12/02/2019
	Subtotal					
March 2019 (4)	□ (2)	TA00024	Audit Services	Approved	OK	09/03/2019
		TA00023	Division of Finance	Approved	GA	09/03/2019
	Subtotal					

Step 4 :- Create a Travel Approvals Dashboard.

- Test The Dashboard
- Dashboard will look as per the screen shot:



Module – 3

Exercise 1 :-

Step 1:- Create a new custom lightning App, name: Code Playground.

App Settings

App Details & Branding

App Options

- Utility Items (Desktop Only)
- Navigation Items
- User Profiles

App Details & Branding

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

App Details

* App Name: Code Playground

* Developer Name: Code_Playground

Description: Enter a description...

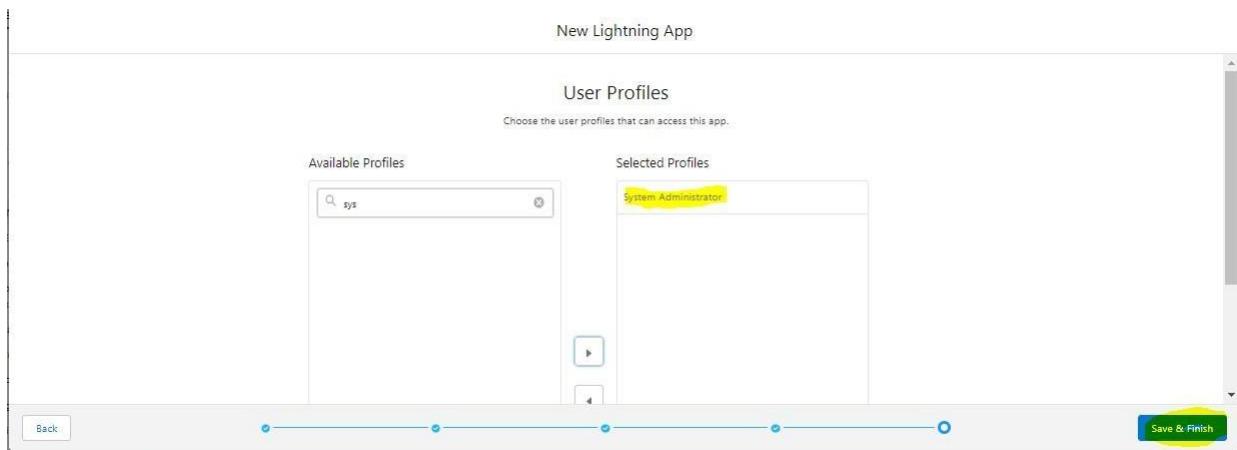
App Branding

Image: Primary Color Hex Value: #0070D2

Org Theme Options: Use the app's image and color instead of the org's custom theme

App Launcher Preview

Code Playground



Step 2:- Create a Custom Object and tab for Customers.

The top screenshot shows the 'New Custom Object' setup page. It includes fields for 'Label' (Customer), 'Plural Label' (Customers), and 'Object Name' (Customer). The bottom screenshot shows the 'New Custom Object Tab' setup page, where 'Customer' is selected as the object and 'Credit card' is chosen as the tab style.

New Custom Object

Custom Object Definition Edit

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label: Customer Example: Account

Plural Label: Customers Example: Accounts

Starts with vowel sound

The Object Name is used when referencing the object via the API.

Object Name: Customer Example: Account

Description:

Setup **Home** **Object Manager**

Tabs

New Custom Object Tab

Step 1. Enter the Details

Select an existing custom object or [create a new custom object now](#).

Object: Customer

Tab Style: Credit card

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

Splash Page Custom Link: --None--

Description:

Step 3:- Create a Custom fields for Customer Object a. Label = Active, Checkbox, Save.

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

- Setup** tab selected in the top navigation bar.
- Object Manager** selected in the dropdown menu.
- Customer** object selected in the list.
- Fields & Relationships** section selected in the left sidebar.
- Step 2. Enter the details** screen displayed.
- Field Label**: Active
- Default Value**: Unchecked (radio button selected)
- Field Name**: Active
- Description**: (empty)
- Help Text**: (empty)
- Auto add to custom report type**: Add this field to existing custom report types that contain this entity (checkbox checked)
- Step 2 of 4** in the top right corner.
- Previous**, **Next**, and **Cancel** buttons at the bottom right.

b. Label = Customer Type, Picklist, Values: Premium, Standard

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

- Setup** tab selected in the top navigation bar.
- Object Manager** selected in the dropdown menu.
- Customer** object selected in the list.
- Fields & Relationships** section selected in the left sidebar.
- New Custom Field** screen displayed.
- Step 2. Enter the details** screen displayed.
- Field Label**: Customer Type
- Values**: Enter values, with each value separated by a new line (radio button selected)
- Premium**
Standard (values entered in the text area)
- Display values alphabetically, not in the order entered** (checkbox)
- Use first value as default value** (checkbox)
- Restrict picklist to the values defined in the value set** (checkbox checked)
- Field Name**: Customer_Type
- Description**: (empty)
- Step 2 of 4** in the top right corner.
- Previous**, **Next**, and **Cancel** buttons at the bottom right.

c. Label = Description, Text Area, Save.

Customer
New Custom Field

Step 2. Enter the details

Field Label: [i]

Field Name: [i]

Description:

Help Text:

Required: Always require a value in this field in order to save a record

Auto add to custom report type: Add this field to existing custom report types that contain this entity [i]

Default Value: [i]

Use formula syntax. Enclose text and picklist value API names in double quotes ; ("the_text"), include numbers without quotes ; (25), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today) + 7). To reference a field from a Custom Metadata type record use: \$CustomMetadata.Type__mdt.RecordAPIName.Field__c

Step 2 of 4

Previous Next Cancel

d. Label = Customer, Master-Detail, Related To – Customer custom object

Customer
New Relationship

Step 2. Choose the related object

Select the other object to which this object is related.

Related To: [i]

Step 2 of 6

Previous Next Cancel

Step 4:- Create a Custom Object for Billing.

SETUP Home Object Manager ▾

New Custom Object

Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. [Tell me more!](#) [Don't show this message again](#)

Custom Object Definition Edit [Save](#) [Save & New](#) [Cancel](#)

Custom Object Information | * Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label	<input type="text" value="Billing"/>	Example: Account
Plural Label	<input type="text" value="Billings"/>	Example: Accounts
Starts with vowel sound	<input type="checkbox"/>	

The Object Name is used when referencing the object via the API.

Object Name	<input type="text" value="Billing"/>	Example: Account
-------------	--------------------------------------	------------------

Description

Context-Sensitive Help Setting Open the standard Salesforce.com Help & Training window Open a window using a Visualforce page

Content Name [-None-](#) ▾

SETUP Tabs

New Custom Object Tab

Step 1. Enter the Details Step 1 of 3

Choose the custom object for this new custom tab. Fill in other details.

Select an existing custom object or [create a new custom object now](#).

Object	<input style="width: 150px;" type="text" value="Billing"/>	🔍
Tab Style	<input style="width: 150px; background-color: #ffffcc; border: 1px solid #ccc;" type="text" value="Bank"/>	🔍

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

Splash Page Custom Link [-None-](#) ▾

Enter a short description.

Description

[Next](#) [Cancel](#)

Step 5:- Create a Custom fields for Billing Object. a. Label = Amount Paid, Currency, Save.

Setup > OBJECT MANAGER
Billing

Fields & Relationships

Custom Field Definition Edit

Field Information

- Field Label: Amount Paid
- Field Name: Amount_Paid
- Description:
- Help Text:
- Data Owner: User
- Field Usage: --None--
- Data Sensitivity Level: --None--
- Compliance Categorization:
 - Available: PII, HIPAA, GDPR, PCI
 - Chosen:

General Options

Required: Always require a value in this field in order to save a record

b. Label = Customer Type, Picklist, Values: Premium, Standard.

Setup > OBJECT MANAGER
Billing

Fields & Relationships

Custom Field Definition Edit

Field Information

- Field Label: Customer Type
- Field Name: Customer_Type
- Description:
- Help Text:
- Data Owner: User
- Field Usage: --None--
- Data Sensitivity Level: --None--
- Compliance Categorization:
 - Available: PII, HIPAA, GDPR, PCI
 - Chosen:

General Options

Required: Always require a value in this field in order to save a record

Default Value: Show Formula Editor

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (100) other considerations on documents (in .pdf and .rtf) and external data with commas for their individual elements (Product 1, 99, 45)

c. Label = Status, Picklist, Values: Paid, Unpaid.

SETUP > OBJECT MANAGER
Billing

Fields & Relationships

Field Information

Field Label: Status
Field Name: Status
Description:
Help Text:
Data Owner: User
Field Usage: None
Data Sensitivity Level: None
Compliance Categorization:

Available	Chosen
PII	
HIPAA	
GDPR	
PCI	

General Options

Required: Always require a value in this field in order to save a record
Default Value: Show Formula Editor

Use formula syntax. Enclose text and picklist value API names in double quotes ; ("the_text"), include numbers without quotes ; (25), show percentages as decimals (0.10), and express date calculations in the standard format : (Today) + 7). To reference a field from a Custom Metadata type record use : \$CustomMetadata.Type__mdt.RecordId\$Name.FieldName__c

Test The App

- Code Playground App should look like this:

Code Playground

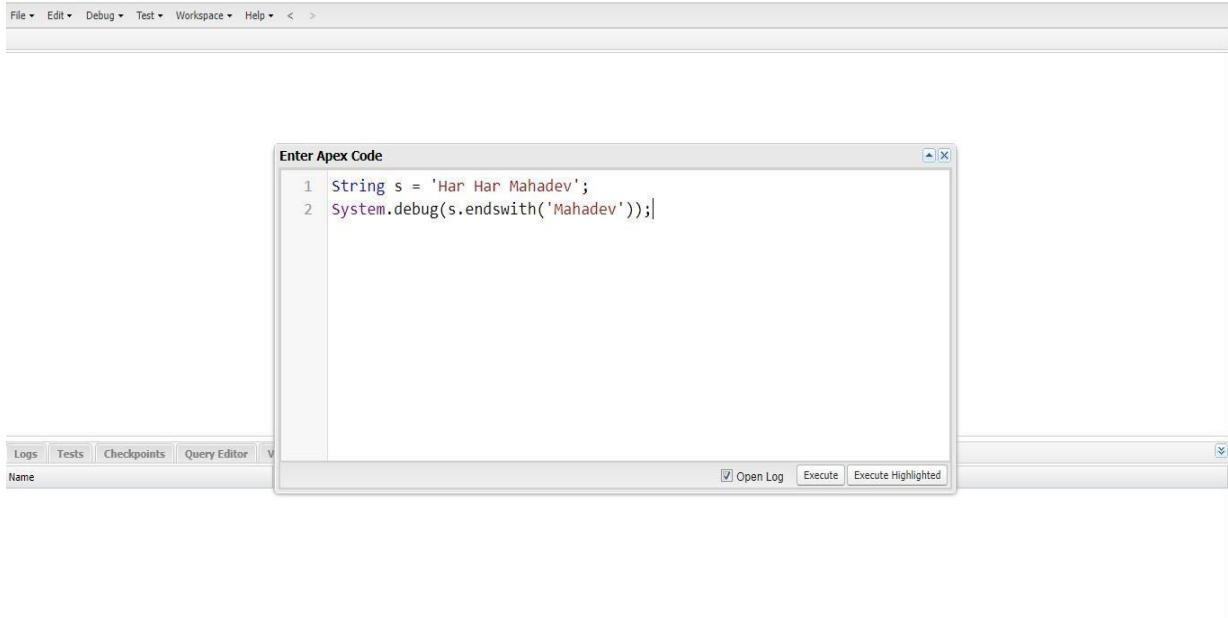
All Accounts

13 items • Sorted by Account Name • Filtered by All accounts • Updated 4 minutes ago

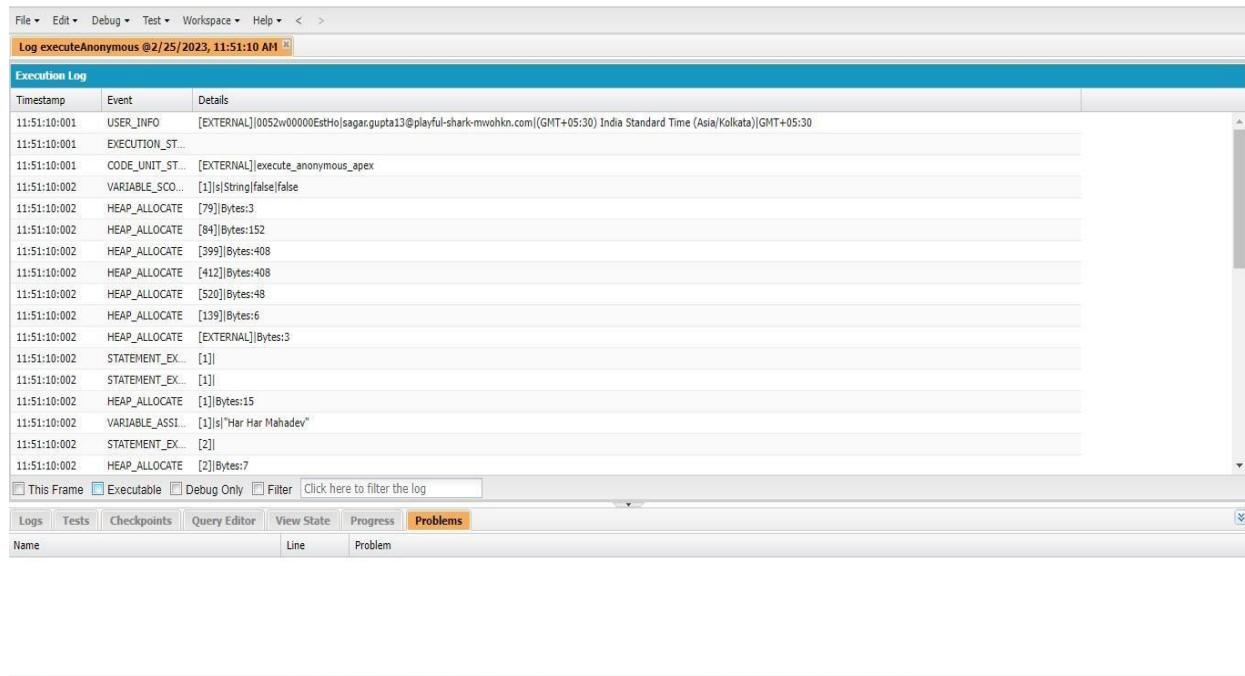
	Account Name	Account Site	Billing State/Province	Phone	Type	Account Owner Alias
1	Burlington Textiles Corp of America		NC	(336) 222-7000	Customer - Direct	SGupt
2	Dickenson plc		KS	(785) 241-6200	Customer - Channel	SGupt
3	Edge Communications		TX	(512) 757-6000	Customer - Direct	SGupt
4	Express Logistics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

Exercise 2 : - Use Execute Anonymous to define and execute the following code:

Code 1:- Define a String Variable & use string method ‘endsWith’ to display the output.



Output :



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

The screenshot shows the Salesforce Developer Console interface. At the top, there's a menu bar with options like File, Edit, Debug, Test, Workspace, and Help. Below the menu is a toolbar with icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and Delete. A central window titled "Enter Apex Code" contains the following Apex code:

```
1 Date date1 = Date.today();
2 Date date2 = date1.adddays(30);
3 System.debug(date2);
```

Below the code editor is a toolbar with buttons for Open Log, Execute, and Execute Highlighted. At the bottom of the interface, there are tabs for Logs, Tests, Checkpoints, Query Editor, and a search bar. The search bar has dropdown menus for Name, Line, and Problem.

Output :

The screenshot shows the Salesforce Developer Console interface, specifically the Execution Log section. The title bar indicates the log was created at 2/25/2023, 12:00:59 PM. The log table has columns for Timestamp, Event, and Details. One entry is visible:

Timestamp	Event	Details
12:00:59:003	USER_DEBUG	[3]DEBUG 2023-03-27 00:00:00

At the bottom of the interface, there are checkboxes for This Frame, Executable, and Debug Only, along with a Filter button and a link to "Click here to filter the log". There are also tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently selected. A search bar at the bottom is set to search by Name, Line, or Problem.

Code 4 : Display the output of an Integer variable from string ‘10’ and then add 20 to it.

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

Enter Apex Code

```
1 String x = '10';
2 integer y = 20;
3 integer z = integer.valueOf(x) +y;
4 System.debug(z);
```

Logs Tests Checkpoints Query Editor

Open Log Execute Execute Highlighted

Name Line Problem

Output :

Log executeAnonymous @2/25/2023, 12:11:08 PM

Execution Log

Timestamp	Event	Details
12:11:08:002	USER_DEBUG	[4]DEBUG 30

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

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

Code 5 : Define a String Variable & use string method length() to display the output.

The screenshot shows the Salesforce Execute Anonymous window. At the top, there are three tabs: 'Log executeAnonymous @2/25/2023, 12:11:08 PM' (selected), 'Log executeAnonymous @2/25/2023, 12:14:54 PM', and 'Log executeAnonymous @2/25/2023, 12:14:57 PM'. Below the tabs is a header bar with 'Execution Log' and tabs for 'Timestamp', 'Event', and 'Details'. A timestamp '12:14:54:002' and event 'USER_DEBUG [2]DEBUG|15' are listed. The main area is titled 'Enter Apex Code' and contains the following Apex code:

```
1 String x = 'Har har Mahadev';
2 System.debug(x.length());
```

Below the code editor are several buttons: 'This Frame', 'Executable', 'Debug Only' (which is checked), 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'Open Log' (which is checked), 'Execute', and 'Execute Highlighted'. The status bar at the bottom shows 'Name' and tabs for 'Line' and 'Problem'.

Output :

The screenshot shows the Salesforce Execution Log window. At the top, there are three tabs: 'File', 'Edit', 'Debug', 'Test', 'Workspace', 'Help', and 'Logs' (selected). Below the tabs are three log entries:

Timestamp	Event	Details
12:15:57:003	USER_DEBUG	[2]DEBUG 15
12:15:57:003	USER_DEBUG	[2]DEBUG 15
12:15:57:003	USER_DEBUG	[2]DEBUG 15

Below the logs is a toolbar with 'This Frame', 'Executable', 'Debug Only' (checked), 'Filter' (disabled), and 'Click here to filter the log'. The toolbar also includes 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems' (selected). The status bar at the bottom shows 'Name' and tabs for 'Line' and 'Problem'.

Code 6 : Define a List of integer and display the output using add(), get(), set(), clear(), methods

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

Enter Apex Code

```
1 List<integer> l= new List<integer>();  
2 //Add  
3 l.add(1);  
4 l.add(2);  
5 l.add(3);  
6 l.add(4);  
7 //get  
8 System.debug(l);  
9 integer num = l.get(1);  
10 System.debug(num);  
11 //set  
12 l.set(3,40);  
13 System.debug(l);  
14 //clear  
15 l.clear();  
16 System.debug(l);
```

Logs Tests Checkpoint

Name

Open Log

Output :

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

Log executeAnonymous @2/25/2023, 12:24:55 PM

Execution Log

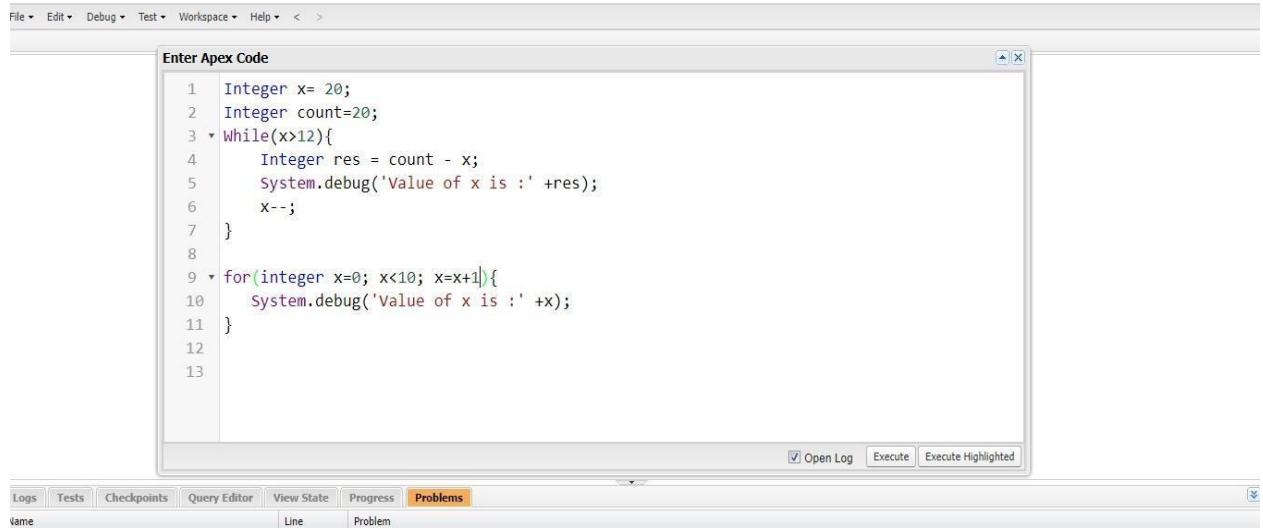
Timestamp	Event	Details
12:24:55:003	USER_DEBUG	[8]DEBUG (1, 2, 3, 4)
12:24:55:003	USER_DEBUG	[10]DEBUG 2
12:24:55:003	USER_DEBUG	[13]DEBUG (1, 2, 3, 40)
12:24:55:003	USER_DEBUG	[16]DEBUG ()

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

Logs Tests Checkpoints Query Editor View State Progress Problems

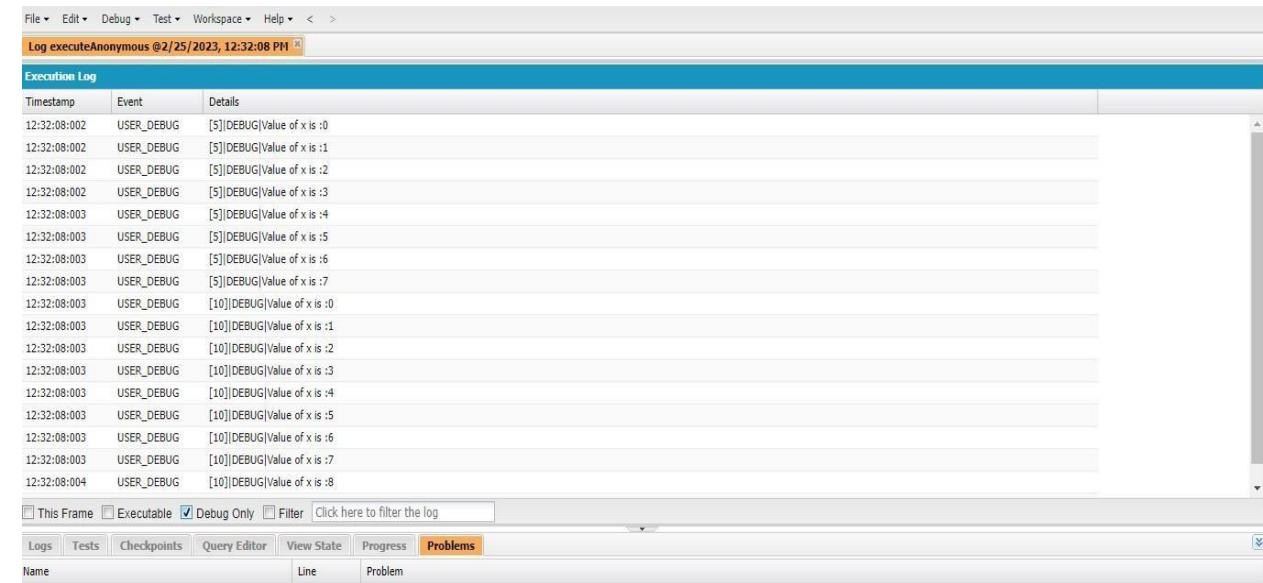
Name Line Problem

Code 7 : Use Execute Anonymous to define and execute the following code to display the value of x = 0 to 9.



```
1 Integer x= 20;
2 Integer count=20;
3 While(x>12){
4     Integer res = count - x;
5     System.debug('Value of x is :' +res);
6     x--;
7 }
8
9 for(integer x=0; x<10; x=x+1){
10     System.debug('Value of x is :' +x);
11 }
12
13
```

Output :



Timestamp	Event	Details
12:32:08:002	USER_DEBUG	[5]DEBUG value of x is :0
12:32:08:002	USER_DEBUG	[5]DEBUG value of x is :1
12:32:08:002	USER_DEBUG	[5]DEBUG value of x is :2
12:32:08:002	USER_DEBUG	[5]DEBUG value of x is :3
12:32:08:003	USER_DEBUG	[5]DEBUG value of x is :4
12:32:08:003	USER_DEBUG	[5]DEBUG value of x is :5
12:32:08:003	USER_DEBUG	[5]DEBUG value of x is :6
12:32:08:003	USER_DEBUG	[5]DEBUG value of x is :7
12:32:08:003	USER_DEBUG	[10]DEBUG Value of x is :0
12:32:08:003	USER_DEBUG	[10]DEBUG Value of x is :1
12:32:08:003	USER_DEBUG	[10]DEBUG Value of x is :2
12:32:08:003	USER_DEBUG	[10]DEBUG Value of x is :3
12:32:08:003	USER_DEBUG	[10]DEBUG Value of x is :4
12:32:08:003	USER_DEBUG	[10]DEBUG Value of x is :5
12:32:08:003	USER_DEBUG	[10]DEBUG Value of x is :6
12:32:08:003	USER_DEBUG	[10]DEBUG Value of x is :7
12:32:08:004	USER_DEBUG	[10]DEBUG Value of x is :8

Exercise 3 : Answer the following in True Or False:

The screenshot shows the Salesforce IDE interface. At the top, there's a menu bar with File, Edit, Debug, Test, Workspace, Help, and navigation buttons. Below the menu is an "Enter Apex Code" window containing the following Apex code:

```
1 Integer myunluckyNumber = 7;
2 Integer myluckyNumber = 15;
3 System.debug(myluckyNumber != myunluckyNumber + 8);
```

Below the code editor is a toolbar with Open Log, Execute, and Execute Highlighted buttons. Underneath the code editor is a "Problems" tab in the navigation bar. The "Logs" tab is also visible. The main workspace area below the tabs is currently empty.

Output :

The screenshot shows the Salesforce IDE interface. At the top, there's a menu bar with File, Edit, Debug, Test, Workspace, Help, and navigation buttons. A specific log entry is highlighted: "Log executeAnonymous @2/25/2023, 12:35:10 PM". Below the menu is an "Execution Log" window with a header row for Timestamp, Event, and Details. The log contains one entry:

Timestamp	Event	Details
12:35:10:002	USER_DEBUG	[3]DEBUG:false

At the bottom of the log window, there are checkboxes for "This Frame", "Executable", "Debug Only", and "Filter", along with a "Click here to filter the log" link. Below the log is a "Problems" tab in the navigation bar. The "Logs" tab is also visible. The main workspace area below the tabs is currently empty.

Answer the following in True Or False

Exercise 4 : -

The screenshot shows the Salesforce IDE interface. At the top is a menu bar with File, Edit, Debug, Test, Workspace, Help, and navigation icons. Below the menu is a toolbar with standard file operations. The main area is titled "Enter Apex Code" and contains the following Apex code:

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

At the bottom of the code editor are three buttons: Open Log, Execute, and Execute Highlighted. Below the code editor is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing a table with columns: Name, Line, and Problem. There are no entries in the table.

Output :

The screenshot shows the Salesforce IDE interface with the Execution Log panel open. The title bar says "Log executeAnonymous @2/25/2023, 12:38:19 PM". The Execution Log table has columns: Timestamp, Event, and Details. One entry is shown:

Timestamp	Event	Details
12:38:19:002	USER_DEBUG	[3] DEBUG true

Below the log table is a filter bar with checkboxes for "This Frame", "Executable", "Debug Only", and "Filter", and a link "Click here to filter the log". At the bottom is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing a table with columns: Name, Line, and Problem. There are no entries in the table.

Answer the following in True Or False

Exercise 5 : -

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

Output :

Log executeAnonymous @2/25/2023, 12:40:12 PM

Timestamp	Event	Details
12:40:12:002	USER_DEBUG	[3]DEBUG true

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

Exercise 6 :-

The screenshot shows the Salesforce Apex code editor. The code in the editor is:

```
1 Integer Score =80;
2 If (Score == 100){
3     System.debug('Grade: A+');
4 }else If (Score >= 90){
5     System.debug('Grade: A');
6 }else If (Score >= 80){
7     System.debug('Grade: B');
8 }else{
9     System.debug('Grade:Failed');
10 }
```

Below the code editor, there are buttons for "Open Log", "Execute", and "Execute Highlighted". The status bar at the bottom shows tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems, with the Problems tab currently selected.

Output :

The screenshot shows the Salesforce log viewer. The log message is:

Timestamp	Event	Details
12:45:28:002	USER_DEBUG	[7]DEBUG Grade: B

At the top, it says "Log executeAnonymous @2/25/2023, 12:45:28 PM". Below the log table, there are checkboxes for "This Frame", "Executable", "Debug Only", and "Filter", followed by a link "Click here to filter the log". The status bar at the bottom shows tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems, with the Problems tab currently selected.

Exercise 7 : - Write a program to execute and demo the use of “Apex – for Loop”

Note: Create at least 2 Billing records with Status = Paid for this exercise.

The screenshot shows the Salesforce Code Playground interface. At the top, there's a navigation bar with links for Accounts, Leads, Contacts, Opportunities, Cases, Customers, and Billings. Below the navigation is a list of 'Billings' with columns for Bill Number, Customer Type, and Status. The list contains four items: B - 0001 (Premium, Paid), B - 0002 (Standard, Paid), B - 0003 (Premium, Unpaid), and B - 0004 (Standard, Unpaid). Below this is a code editor window titled 'Billing.apxc'. The code is as follows:

```
1 * public class Billing {
2 *     public static void viewbills(){
3 *         List<Billing__c> Billinglist=[SELECT Id, Name, Status__c FROM Billing__c];
4 *         List<String> Billsrecord = new List<String>();
5 *         for(Billing__c bill : Billinglist){
6 *             System.debug ('Value of Current Record in the Loop' + BillingList);
7 *             if(bill.Status__c == 'paid'){
8 *                 Billsrecord.add(bill.name);
9 *             }
10 *         }
11 *         System.debug('Value of BillingList '+Billsrecord);
12 *     }
13 * }
```

At the bottom of the code editor, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing no errors.

Output :

The screenshot shows the Salesforce Log executeAnonymous interface. It displays the execution log with three entries:

Timestamp	Event	Details
13:10:44:067	USER_DEBUG	[6]DEBUG Value of Current Record in the Loop(Billing__c:{Id=a012w000017Z6EXAA0, Name=B - 0002, Status__c=Paid}, Billing__c:{Id=a012w000017Z6ESAA0, Name=B - 0001, Status__c=Paid})
13:10:44:068	USER_DEBUG	[6]DEBUG value of Current Record in the Loop(Billing__c:{Id=a012w000017Z6EXAA0, Name=B - 0002, Status__c=Paid}, Billing__c:{Id=a012w000017Z6ESAA0, Name=B - 0001, Status__c=Paid})
13:10:44:068	USER_DEBUG	[11]DEBUG Value of BillingList (B - 0002, B - 0001)

Below the log is an 'Enter Apex Code' dialog box containing the code: 'billing.viewbills();'. At the bottom of the interface, there are buttons for Open Log, Execute, and Execute Highlighted.

Exercise 8 : - Write a Class to demo the use of Constants in Apex

The screenshot shows the Salesforce IDE interface. The top menu bar includes File, Edit, Debug, Test, Workspace, Help, and several dropdowns. The main area displays the code for `DiscountClass.apxc`. The code defines a class `DiscountClass` with a static method `calculateDiscount` that takes an integer price and returns a decimal final price after applying a 10% discount. The API version is set to 57. Below the code editor is a toolbar with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems, with Problems selected. A status bar at the bottom shows the name, line, and problem count.

```
1 public class DiscountClass {  
2     public Static Decimal calculateDiscount(Integer price){  
3         Decimal regularDiscount =0.1;  
4         Decimal finalPrice = price - price*regularDiscount;  
5         return finalPrice;  
6     }  
7 }  
8 |
```

Output :

The screenshot shows the Salesforce IDE interface with the Execution Log and Enter Apex Code dialog open. The Execution Log table shows a single entry for a USER_DEBUG log at 13:21:51:011 with the message "[2] DEBUG|finalPrice90.0". The Enter Apex Code dialog contains the following Apex code:

```
1 Decimal finalPrice= DiscountClass.calculateDiscount(100);  
2 System.debug('finalPrice' +finalPrice);
```

At the bottom, there are buttons for Open Log, Execute, and Execute Highlighted. The status bar at the bottom shows the log filter settings and the Problems tab selected.

Exercise 9 : - Write a Class to demo the use of Interface in Apex

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
InterfaceExample.apxc [PremiumCustomer.apxc [normalCustomer.apxc [Log executeAnonymous @2/25/2023, 4:37:56 PM []
Code Coverage: None ▾ API Version: 57 ▾ Go To

```
1 public interface InterfaceExample {  
2     Double percentageDiscountTobeApplied(); //method signature only  
3 }  
4  
5 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
InterfaceExample.apxc [PremiumCustomer.apxc [normalCustomer.apxc [Log executeAnonymous @2/25/2023, 4:37:56 PM []
Code Coverage: None ▾ API Version: 57 ▾ Go To

```
1 //Premium Customer Class  
2 public class PremiumCustomer implements InterfaceExample{  
3     //method Call  
4     public Double percentageDiscountTobeApplied(){  
5         return 0.30; //For premium Customer , Discount should be 30%  
6     }  
7 }  
8 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

File ▾ Click to go back, hold to see history [Help ▾ < >
InterfaceExample.apxc [PremiumCustomer.apxc [normalCustomer.apxc [Log executeAnonymous @2/25/2023, 4:37:56 PM []
Code Coverage: None ▾ API Version: 57 ▾ Go To

```
1 //normal Customer Class  
2 public class normalCustomer implements InterfaceExample{  
3     //method Call  
4     public Double percentageDiscountTobeApplied(){  
5         return 0.10; //For Normal Customer , Discount should be 10%  
6     }  
7 }  
8 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

Output :

The screenshot shows the Salesforce Apex Dev Console interface. At the top, there are tabs for 'InterfaceExample.apxc', 'PremiumCustomer.apxc', 'normalCustomer.apxc', and 'Log executeAnonymous @2/25/2023, 4:37:56 PM'. Below the tabs is the 'Execution Log' section with columns for 'Timestamp', 'Event', and 'Details'. Two entries are listed:

Timestamp	Event	Details
16:37:56:026	USER_DEBUG	[3]DEBUG Discount in Percentage From premium30.0%
16:37:56:027	USER_DEBUG	[6]DEBUG Discount in Percentage From Normal10.0%

Below the log is the 'Enter Apex Code' panel containing the following Apex code:

```
1 PremiumCustomer p1=new PremiumCustomer();
2 Double discount = p1.percentageDiscountTobeApplied();
3 System.debug('Discount in Percentage From premium'+(discount*100)+'%');
4 normalCustomer n1=new normalCustomer();
5 discount = n1.percentageDiscountTobeApplied();
6 System.debug('Discount in Percentage From Normal'+(discount*100)+'%');
```

At the bottom of the interface, there are buttons for 'Open Log', 'Execute', and 'Execute Highlighted'. The status bar at the bottom shows tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems', with 'Problems' being the active tab.

Exercise 10 : - Demo on DML Insert Operation Using Database methods

The screenshot shows the Salesforce Apex Dev Console interface. The top bar includes tabs for 'File', 'Edit', 'Debug', 'Test', 'Workspace', 'Help', and 'Log executeAnonymous @2/26/2023, 12:08:53 AM'. The status bar indicates 'Code Coverage: None' and 'API Version: 57'. The main area displays the following Apex code:

```
1 * public class DML {
2     Public void test()
3     {
4         Customer__c cust = new Customer__c();
5         cust.name = 'Wipro';
6         cust.Customer_Type__c = 'Premium';
7
8         insert cust;
9
10        List<Billing__c> listinsert = new List<Billing__c>{new Billing__c(Status__c = 'paid',Amount_Paid__c = 5000000)};
11        Database.SaveResult[] srList = Database.insert(listinsert,false);
12        For(Database.SaveResult sr : srList{
13            if(sr.isSuccess()){
14                System.debug('Successfully inserted Billing'+sr.getId());
15            }else{
16                for(Database.Error err : sr.getErrors()){
17                    System.debug('the Following error has Occurred.');
18                    System.debug(err.getStatusCode()+' : '+err.getMessage());
19                    System.debug('Billing object Field which are Affect by the error :'+err.getFields());
20
21                }
22            }
23        }
24    }
25 }
```

The status bar at the bottom shows tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems', with 'Problems' being the active tab.

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

DHLLap005 Log executeAnonymous @ 2/25/2023, 11:37:30 PM

Execution Log

Timestamp	Event	Details
23:37:30:001	USER_INFO	[EXTERNAL][0052w00000EstHo]sagar.gupta13@playful-shark-mwohn.com[(GMT+05:30) India Standard Time (Asia/Kolkata)](GMT+05:30)
23:37:30:002	EXECUTION_ST...	
23:37:30:002	CODE_UNIT_ST...	[EXTERNAL]execute_anonymous_apex
23:37:30:002	VARIABLE_SCOP...	[1]cust:Customer__c true false
23:37:30:002	HEAP_ALLOCATE	[79]Bytes:3
23:37:30:002	HEAP_ALLOCATE	[64]Bytes:152
23:37:30:002	HEAP_ALLOCATE	[399]Bytes:408
23:37:30:002	HEAP_ALLOCATE	[412]Bytes:408
23:37:30:002	HEAP_ALLOCATE	[520]Bytes:48
23:37:30:002	HEAP_ALLOCATE	[139]Bytes:6
23:37:30:002	HEAP_ALLOCATE	[64]Bytes:7
23:37:30:003	STATEMENT_EX...	[1]
23:37:30:003	STATEMENT_EX...	[1]
23:37:30:003	HEAP_ALLOCATE	[1]Bytes:4
23:37:30:003	VARIABLE_ASSI...	[1]cust () 0x7d6306ae
23:37:30:003	STATEMENT_EX...	[2]
23:37:30:003	HEAP_ALLOCATE	[2]Bytes:5
23:37:30:003	VARIABLE_ASSI...	[2]this.Name Wipro 0x7d6306ae
23:37:30:003	STATEMENT_EX...	[3]
23:37:30:003	HEAP_ALLOCATE	[3]Bytes:7
23:37:30:004	VARIABLE_ASSI...	[3]this.Customer_Type__c Premium 0x7d6306ae
23:37:30:004	STATEMENT_EX...	[4]
23:37:30:004	HEAP_ALLOCATE	[52]Bytes:5

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

Enter Apex Code

```

10 Customer__c cust = new Customer__c();
11 cust.name = 'Wipro';
12 cust.Customer_Type__c = 'Premium';
13 insert cust;
14
15
16
17
18
19
20
21
22

```

Open Log Execute Execute Highlighted

Logs Tests Checkpoints Query Editor View State Progress Problems

Output :

Code Playground Accounts Leads Contacts Opportunities Cases Customers Billings

Customers All

1 item • Sorted by Customer Name • Filtered by All customers • Updated a few seconds ago

Customer Name ↑	Customer Type
1 Wipro	Premium

The screenshot shows the Salesforce Developer Console. At the top, there's a navigation bar with File, Edit, Debug, Test, Workspace, Help, and a log entry for 'Log executeAnonymous @2/26/2023, 12:08:53 AM'. Below the navigation is an 'Execution Log' table with columns for Timestamp, Event, and Details. A single log entry is shown: '\$DEBUG|Successfully inserted Billing@012w00000172bEAA0'. Below the log is a code editor window titled 'Enter Apex Code' containing the following Apex code:

```

1 List<Billing__c> listinsert = new List<Billing__c>{new Billing__c(Status__c = 'paid',Amount_Paid__c = 5000000)};
2 Database.SaveResult[] srList = Database.insert(listinsert,false);
3 For(Database.SaveResult sr : srList){
4     if(sr.isSuccess()){
5         System.debug('Successfully inserted Billing'+sr.getId());
6     }else{
7         for(Database.Error err : sr.getErrors()){
8             System.debug('the Following error has Occurred.');
9             System.debug(err.getStatusCode()+' :'+err.getMessage());
10            System.debug('Billing object Field which are Affect by the error :'+err.getFields());
11        }
12    }
13}
14
15
16

```

At the bottom of the code editor are buttons for Open Log, Execute, and Execute Highlighted. Below the code editor is a toolbar with filters for This Frame, Executable, Debug Only, and Filter, along with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems.

Output :

The screenshot shows the Salesforce Lightning Experience. At the top, there's a navigation bar with the Travel App logo, Chatter, Reports, Dashboards, Departments, Travel Approvals, Expense Items, Customers, and Billings. Below the navigation is a search bar and a toolbar with icons for New Contact, Edit, and New Opportunity. The main area displays a Billing record detail page for 'B - 0018'. The 'Details' tab is selected. The record contains the following fields:

- Billing Number: B - 0018
- Amount Paid: \$7.867
- Customer Type: Premium
- Status: Unpaid
- Created By: Ashutosh Pattnayak (21/2/2023, 2:07 AM)
- Owner: Ashutosh Pattnayak
- Last Modified By: Ashutosh Pattnayak (7/4/2022, 2:07 AM)

Exercise 11 :- Write and execute SOQL queries from Developer Console.

Output :

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

DMLapocx 8.0 Opportunity@10:21 PM

Select ID, Amount, StageName, Account.Name, Account.Industry, Account.Website From Opportunity Where Account.Industry = 'Energy' AND Account.AnnualRevenue > 5000

Query Results - Total Rows: 10					
Id	Amount	StageName	Account.Name	Account.Industry	Account.Website
0062v00000KCU1AAH	125000	Negotiation/Review	United Oil & Gas Corp.	Energy	http://www.uos.com
0062v00000KCU1sAAH	270000	Proposal/Price Quote	United Oil & Gas Corp.	Energy	http://www.uos.com
0062v00000KCU1yAAH	120000	Closed Won	United Oil & Gas Corp.	Energy	http://www.uos.com
0062v00000KCU26AAH	270000	Negotiation/Review	United Oil & Gas Corp.	Energy	http://www.uos.com
0062v00000KCU29AAH	270000	Closed Won	United Oil & Gas Corp.	Energy	http://www.uos.com
0062v00000KCU2BAAX	915000	Closed Won	United Oil & Gas Corp.	Energy	http://www.uos.com
0062v00000KCU2GAAK	235000	Closed Won	United Oil & Gas Corp.	Energy	http://www.uos.com
0062v00000KCU2HAAK	440000	Closed Won	United Oil & Gas Corp.	Energy	http://www.uos.com
0062v00000KCU2JAAK	120000	Closed Won	United Oil & Gas Corp.	Energy	http://www.uos.com
0062v00000KCU2LAAK	675000	Needs Analysis	United Oil & Gas Corp.	Energy	http://www.uos.com

Query Grid | Save Rows | Insert Row | Delete Row | Refresh Grid | Access in Salesforce | Create New | Open Detail Page | Edit Page

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

Select ID, Amount, StageName, Account.Name, Account.Industry, Account.Website
From Opportunity Where Account.Industry = 'Energy' AND Account.AnnualRevenue > 5000

Any query errors will appear here...

History
Executed
SELECT Id, Name, Status__c FROM Billing__c
Select ID, Amount, StageName, Account.Name, Account.Industry, Accou...
Select ID, Amount, StageName, Account.Name, Account.Industry, Accou...

Exercise 12:- Write an Apex Trigger, Name = CustomerTrigger.

The screenshot shows the Salesforce IDE interface with the CustomerTrigger.apex file open. The code defines a trigger named CustomerTrigger on the Customer__c object. It iterates through new records, checks if Active__c is false, creates a Billing__c record with Status__c set to 'Paid' and Amount_Paid__c set to 1000000, and adds it to a BillingList. Finally, it inserts the Billinglist.

```
1 + trigger CustomerTrigger on Customer__c (after insert, after update) {
2     List<Billing__c> BillingList = new List<Billing__c>();
3
4     for (Customer__c objCustomer: Trigger.new)
5     {
6
7         if (objCustomer.Active__c == False)
8         {
9             Billing__c objbill = new Billing__c();
10            objbill.Status__c = 'Paid';
11            objbill.Amount_Paid__c=1000000;
12            BillingList.add(objbill);
13        }
14    }
15
16    insert BillingList;
17
18 }
```

Exercise 13 :- Write a Test Class for Customer Trigger.

Output :

The screenshot shows the Salesforce IDE interface with the CustomerTriggerTest.apex file open. The test class contains a single method named testName that creates a customer record with Active__c set to false, updates it to Active__c = false, and then runs the test. The test results table below shows four test runs: one failed and three passed.

```
1 + @isTest
2 + public class CustomerTriggerTestClass {
3 +     @isTest static void testName() {
4         Customer__c cust = new Customer__c();
5         cust.Active__c = False;
6         insert cust;
7
8         Test.startTest();
9         cust.Active__c = False;
10        update cust;
11        Test.stopTest();
12    }
13 }
```

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage
+	7072v00008M8P50	Sat Feb 25 2023 22:56:53 GMT...		1	3	Class Overall 33%
+	7072v00008M8P5v	Sat Feb 25 2023 22:04:49 GMT...		0	2	Billing 0% 0/6
+	7072v00008M8Gq	Sat Feb 25 2023 22:57:21 GMT...		1	3	CustomerTrigger 100% 9/9
+	TestRun @ 10:58:00 pm			0	1	demo 0% 0/4
+	DisqualifyTestLeads			0	1	
+	a		0:00	1	4	DiscountClass 0% 0/4

Exercise 14:-Write an Apex Trigger, Name = DisqualifyTestLeads.

The screenshot shows the Salesforce code editor with the tab 'DisqualifyTestLeads.apxt' selected. The code is as follows:

```
1 trigger DisqualifyTestLeads on Lead (before insert) {
2     List<Lead> llist = new List<Lead>();
3     for(Lead le:Trigger.new)
4     {
5         if(le.FirstName.containsIgnoreCase('test')|| string.isBlank(le.FirstName)
6             || le.LastName.containsIgnoreCase('test')||string.isBlank(le.LastName))
7         {
8             system.debug(le.FirstName + ' ' + le.LastName + 'Will be disqualified!');
9             llist.add(le);
10        }
11    }
12    for(Lead l :llist){
13        l.status='Disqualified';
14    }
15 }
```

Exercise 15 :- Write a Test Class for DisqualifyTestLeads.

Output :

The screenshot shows the Salesforce test results for the DisqualifyTestLeads class. The code is as follows:

```
1 @isTest
2 public class DisqualifyTestLeads {
3     @isTest static void a()
4     {
5         Lead le=new Lead();
6         le.FirstName = 'Role';
7         le.LastName = 'Test';
8         le.Company = 'Wipro';
9
10        Test.startTest();
11        insert le;
12        Test.stopTest();
13    }
14 }
```

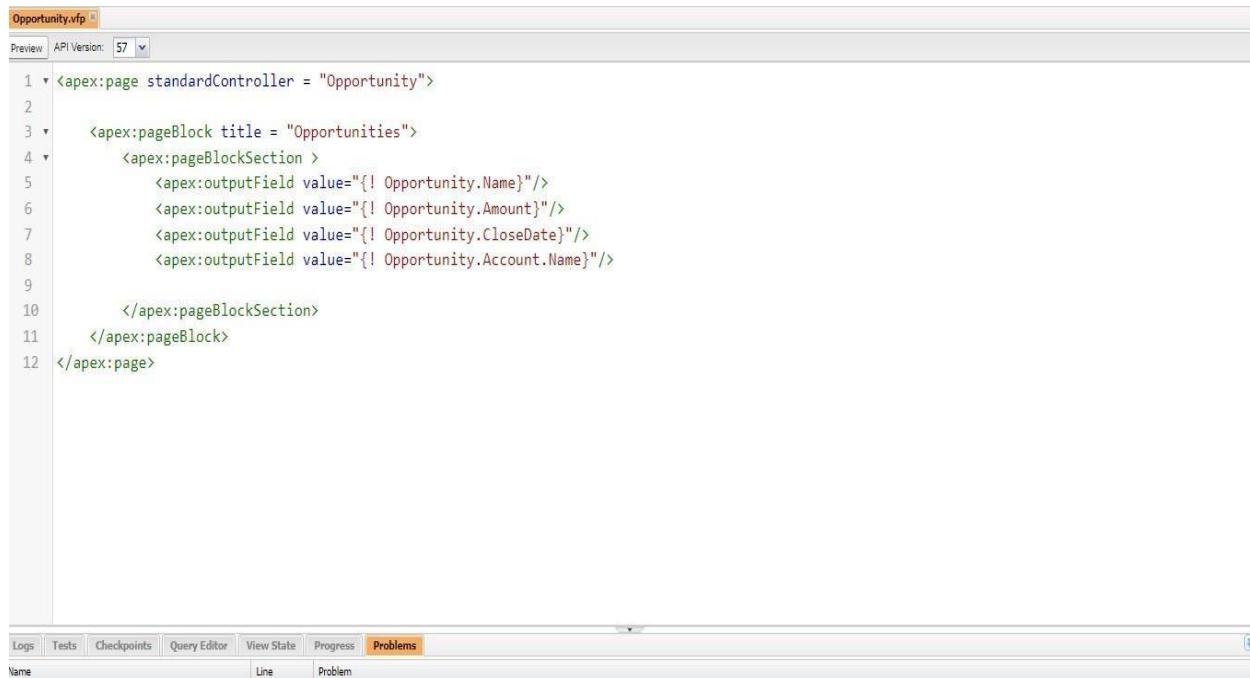
The test results table shows the following runs:

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	7072w00008M85v	Sat Feb 25 2023 22:04:49 GMT...	0:00	0	2
✗	7072w00008M85gq	Sat Feb 25 2023 22:57:21 GMT...	0:00	1	3
✓	7072w00008M85gq TestRun @ 10:58:00 pm	Sat Feb 25 2023 22:57:21 GMT...	0:00	0	1
✓	7072w00008M85v DisqualifyTestLeads	Sat Feb 25 2023 22:59:25 GMT...	0:00	0	1
✓	7072w00008M85v a	Sat Feb 25 2023 22:59:25 GMT...	0:00	1	4
✗	7072w00008M85MK	Sat Feb 25 2023 22:59:25 GMT...	0:00	1	2
✓	7072w00008M85K TestRun @ 11:04:22 pm	Sat Feb 25 2023 22:59:25 GMT...	0:00	0	1

The overall code coverage table is as follows:

Class	Percent	Lines
Overall	20%	0/6
Billing	0%	0/4
demo	0%	0/4
DiscountClass	0%	0/4
DisqualifyTestLeads	100%	8/8
DML	0%	0/13

Exercise 16 Create a Visualforce page which displays Opportunity fields as outputfields.



The screenshot shows the Salesforce Visualforce Editor interface. The top bar includes tabs for 'Preview' and 'API Version 57'. The main area contains the Visualforce page code:

```
1 <apex:page standardController = "Opportunity">
2
3     <apex:pageBlock title = "Opportunities">
4         <apex:pageBlockSection >
5             <apex:outputField value=" {! Opportunity.Name}"/>
6             <apex:outputField value=" {! Opportunity.Amount}"/>
7             <apex:outputField value=" {! Opportunity.CloseDate}"/>
8             <apex:outputField value=" {! Opportunity.Account.Name}"/>
9
10        </apex:pageBlockSection>
11    </apex:pageBlock>
12 </apex:page>
```

Below the code editor is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The 'Problems' tab is selected. A table below the navigation bar shows the following data:

Name	Line	Problem

Output :



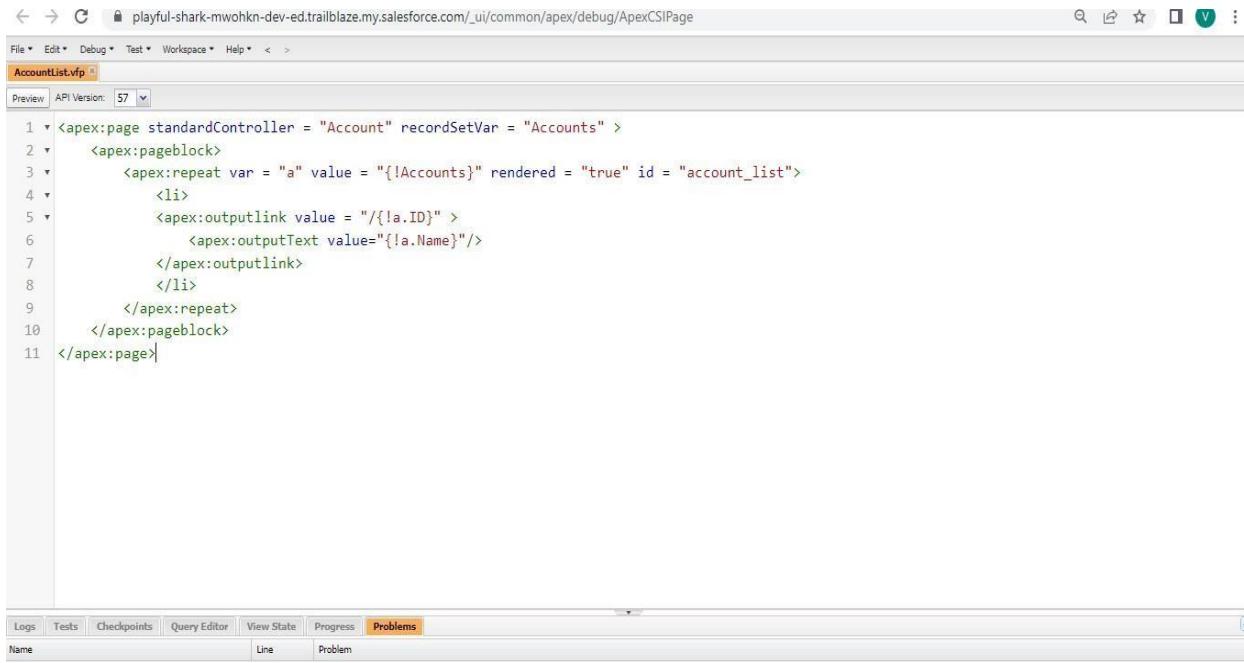
The screenshot shows a browser window displaying the generated Visualforce page. The URL is playul-shark-mwohkn-dev-ed--c.trailblaze.vf.force.com/apex/Opportunity?core.apexpages.request.devconsole=1. The page title is 'Opportunities'. The page content is a table with four columns: Opportunity Name, Amount, Close Date, and Account Name. There is no data present in the table rows.

Opportunity Name	Amount

Close Date	Account Name

:-

Exercise 17 Create a Visualforce page which shows a list of Accounts linked to their record page.



The screenshot shows the Salesforce IDE interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a search bar. The title bar displays the URL: playful-shark-mwohkn-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. Below the title bar, the tab bar shows "AccountList.vfp" and "Preview". The API Version is set to 57. The main content area contains the Visualforce page code:

```
1 <apex:page standardController = "Account" recordSetVar = "Accounts" >
2   <apex:pageblock>
3     <apex:repeat var = "a" value = "{!Accounts}" rendered = "true" id = "account_list">
4       <li>
5         <apex:outputlink value = "/{!a.ID}" >
6           <apex:outputText value="{!a.Name}" />
7         </apex:outputlink>
8       </li>
9     </apex:repeat>
10   </apex:pageblock>
11 </apex:page>
```

At the bottom of the interface, there is a toolbar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The "Problems" tab is currently selected. A status bar at the bottom indicates "Name" and "Line Problem".

Output :

A screenshot of a web browser window. The address bar shows the URL: playful-shark-ohkn-dev-ed-c.trailblaze.vf.force.com/apex/Account?core.apexpages.request.devconsole=1. The page content is a list of account names, each preceded by a small blue circular icon with a white question mark. The accounts listed are:

- Burlington Textiles Corp of America
- Dickenson plc
- Edge Communications
- Express Logistics and Transport
- GenePoint
- Grand Hotels & Resorts Ltd
- Pyramid Construction Inc.
- Sample Account for Entitlements
- sForce
- United Oil & Gas Corp.
- United Oil & Gas, Singapore
- United Oil & Gas, UK
- University of Arizona

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.

A screenshot of the Salesforce Dev Console. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for Preview and API Version (57). The main area displays the Visualforce page code:

```
1 <apex:page controller="NewCaseListController">
2   <apex:repeat value="{!NewCases}" var="case">
3     <li><apex:outputLink value="/{!case.id}" target="_new"> {!case.CaseNumber}</apex:outputLink></li>
4   </apex:repeat>
5 </apex:page>
```

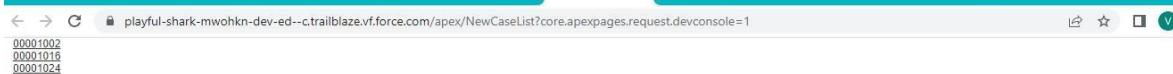
The bottom navigation bar includes Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently selected, showing no errors or warnings.

The screenshot shows the Force.com IDE interface. The top menu bar includes File, Edit, Debug, Test, Workspace, Help, and a search bar. Below the menu is a toolbar with Code Coverage (None), API Version (57), and a Go To button. The main editor area contains the following Apex code:

```
1 public class NewCaseListController {  
2     public List<case> getNewCases(){  
3         List<case>results =Database.query('SELECT ID, CaseNumber from Case where Status = \'New\'');  
4         return results;  
5     }  
6 }
```

Below the editor is a navigation bar with Log, Tests, Checkpoints, Query Editor, View State, Progress, and Problems tabs. The Problems tab is selected, showing two entries: Name 00001002 and Name 00001016.

Output :



References

1. [Manage sales - Salesforce IN](#)
2. [Salesforce - ADX201 Administrative Essentials for New Admins in Lightning Experience \(SFADX201\) \(qa.com\)](#)
3. [Understand the Salesforce Architecture Unit | Salesforce Trailhead](#)