



Capstone Project Report

On “Travel App”

Submitted by:

Name: Meghana

Batch: Wipro Salesforce B13 (WIP-SF-13)

LMS Id: MGSA_500

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 as

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.

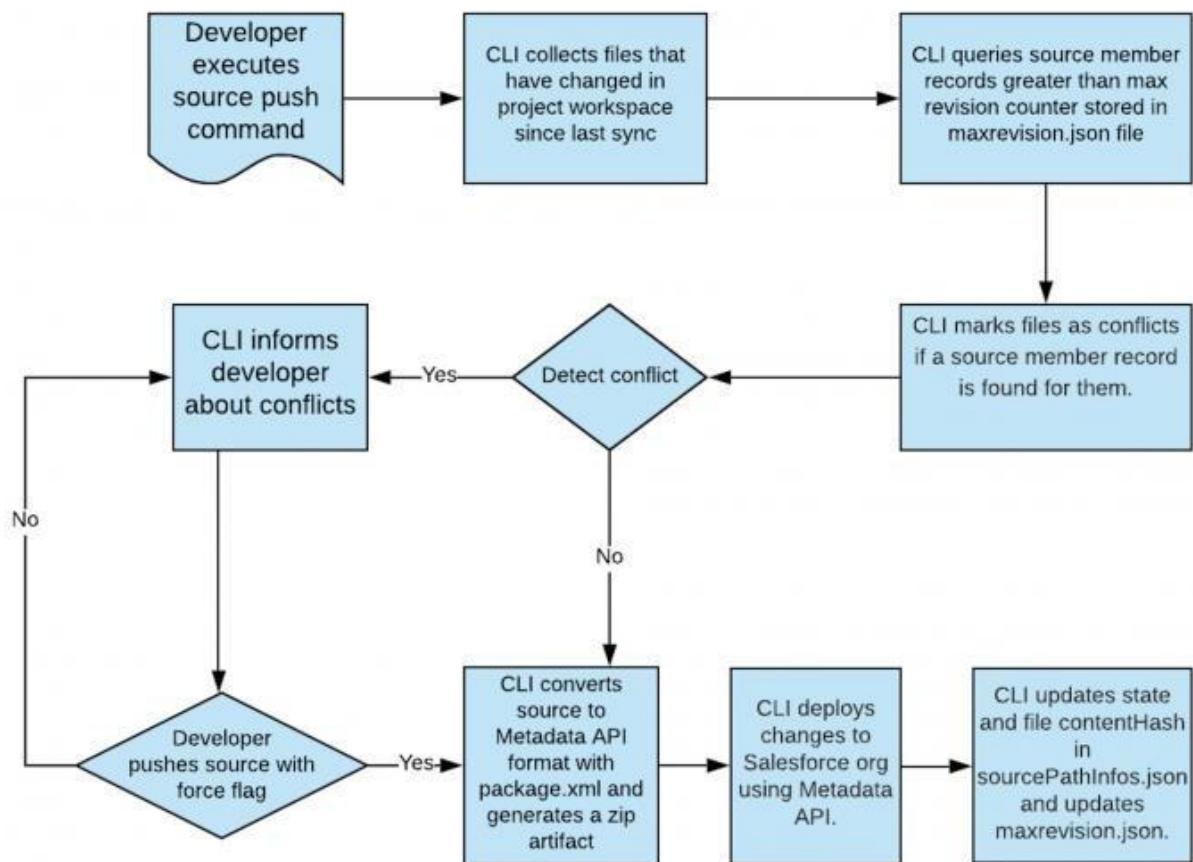
Salesforce is currently providing various software solutions and platforms for developers to create and distribute custom software/applications. Tech giants like Google, Twitter, Amazon, and Facebook are using Salesforce either in the form of SaaS or PaaS.



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

Screen shots

Module – 1

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>i</small> <input type="text" value="Travel App"/>	Image <small>i</small>  Primary Color Hex Value <input type="color" value="#0070D2"/> #0070D2
* Developer Name <small>i</small> <input type="text" value="Travel App"/>	
Description <small>i</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 <small>i</small>	Setup and Personalization <small>i</small>
* 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 Selected Items

Chatter
Reports
Dashboards

Back Next

New Lightning App

Available Profiles Selected Profiles

System Administrator

Back Save & Finish

Final output of step 1 -

Travel App Chatter Reports Dashboards Departments Travel Approvals

What I Follow
To Me
Bookmarked
Company Highlights
My Drafts
STREAMS +
RECENT GROUPS +

Share an update... Share

Sort by: Top Posts Search this feed...

TA-00001 — Meghana M
22 February 2023 at 1:09 pm
@Eric Executive
Which department should I associate this travel request with?
Like Comment 1 comment · 1 view

Meghana M
5 days ago
Technology is the correct department.
Like

Write a comment...

Einstein Recommendations

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

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. A search bar at the top right contains the text 'Search Setup'. Below the navigation bar, a blue header bar displays 'SETUP' and 'New Custom Object'. The main content area is titled 'Custom Object Information'. It includes fields for 'Label' (set to 'Department') and 'Plural Label' (set to 'Departments'). There is also a checkbox for 'Starts with vowel sound'. A note below these fields states: 'The singular and plural labels are used in tabs, page layouts, and reports.' The 'Object Name' field is set to 'Department', with an example 'Account' provided. A large text area labeled 'Description' is empty. 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, a section titled 'Enter Record Name Label and Format' contains a note about record names appearing in various layouts. The 'Record Name' field is set to 'Department Name', with an example 'Account Name'. The 'Data Type' dropdown is set to '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

- Treat 'ABC' and 'abc' as duplicate values (case insensitive)
- Treat 'ABC' and 'abc' as different values (case sensitive)

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 ("theText"), include numbers without quotes : (2), show percentages as decimals (.01), and express date calculations in the standard format ('Today') + 7). To reference a field from a Custom Metadata type record use: [CustomMetadata.Type__inst.RecordName.Field__c]

Previous Next Cancel

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

SETUP > OBJECT MANAGER
Department

Details

Fields & Relationships

Step 2. Enter the details

Field Label: Location

Values:

- 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:

Step 2 of 4

Previous Next Cancel

(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

Location:	Department Type:	Kolkata	Delhi
		Banking	Banking
		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 'Custom Object Information' and 'Required Information'. The 'Custom Object Information' tab is active. It contains fields for 'Label' (Travel Approval), 'Plural Label' (Travel Approvals), and 'Starts with vowel sound' (checkbox). Below these, the 'Object Name' field is set to 'Travel_Approval' with 'Example: Account' shown. A 'Description' text area is present. Under 'Context-Sensitive Help Setting', the 'Open the standard Salesforce.com Help & Training window' radio button is selected. The 'Content Name' dropdown is set to 'None'. In the 'Enter Record Name Label and Format' section, the 'Record Name' field is set to 'Travel Approval #' with 'Example: Account Name' shown. The 'Data Type' dropdown is set to '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

Fields & Relationships					
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Page Layouts	Created By	CreatedById	Lookup(User)		
Lightning Record Pages	Department	Department__c	Lookup(Department)		✓
Buttons, Links, and Actions	Destination State	Destination_State__c	Text(2)		▼
Compact Layouts	Last Modified By	LastModifiedById	Lookup(User)		▼
Field Sets	Out-of-State	Out_of_State__c	Checkbox		▼
Object Limits	Owner	OwnerId	Lookup(User,Group)		✓
Record Types	Purpose of Trip	Purpose_of_Trip__c	Text Area(255)		▼
Related Lookup Filters	Status	Status__c	Picklist		▼
Search Layouts					▼
List View Button Layout					▼

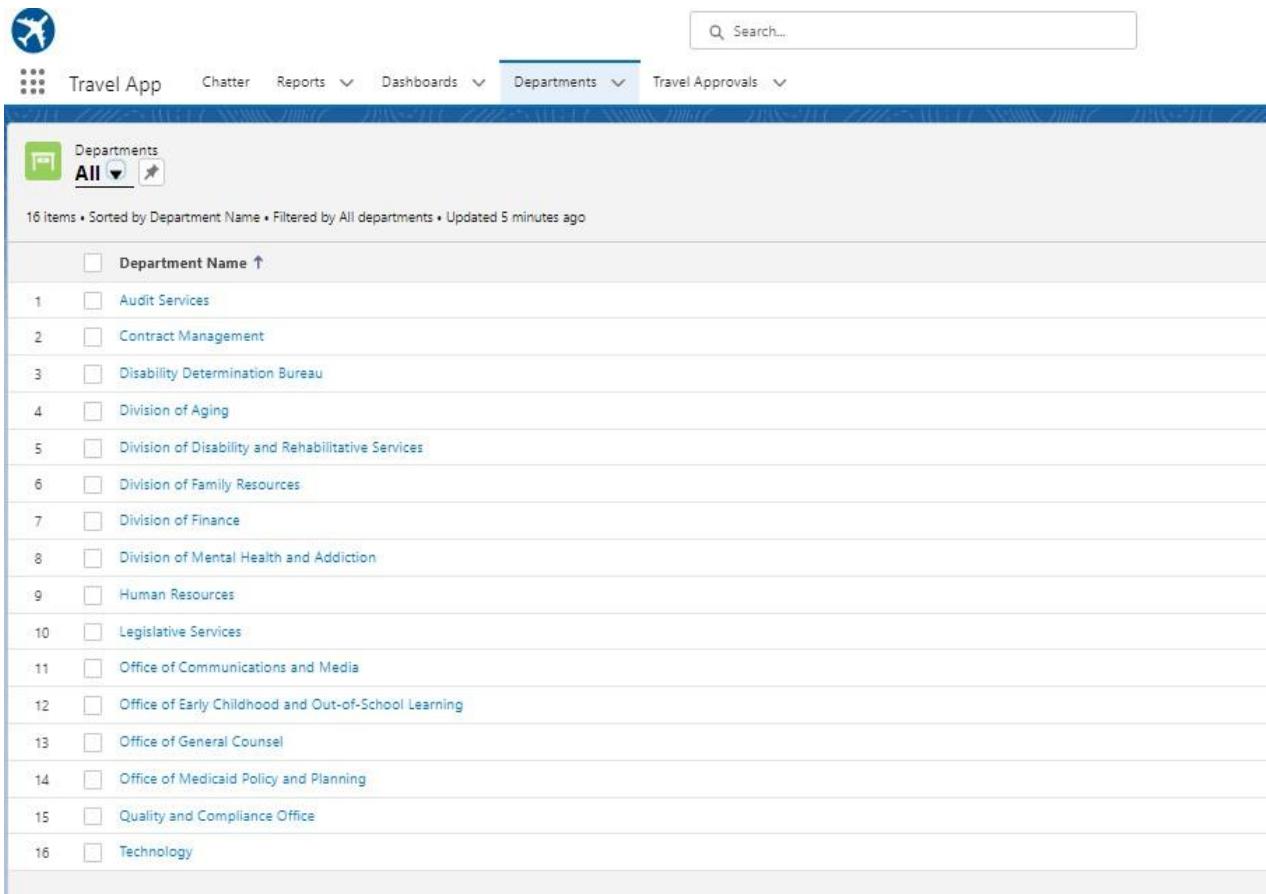
Step 5 : - Import the data store in Department.csv by using “Data Import wizard” tool.

5(a) -

5(b) -

Edit	Mapped Salesforce Object	CSV Header	Example	Example	Example
Change	Department Name	Department Name	Audit Services	Contract Managem	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 (selected), and Travel Approvals. A search bar is located at the top right. 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. Each item is numbered from 1 to 16 and includes a checkbox and the department name. 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.

The screenshot shows a Salesforce page for a 'Travel Approval' record with ID TA-00001. The page has a 'Details' tab selected. Under 'Trip Info', it shows the purpose of the trip as 'Attend Dreamforce', start date as 21/02/2023, end date as 23/02/2023, total expenses of ₹1,320.00, and an 'Out-of-State' checkbox which is checked. The destination state is listed as CA. On the right side, the record is owned by Meghana M, belongs to the Technology department, and was last modified by Meghana M on 27/02/2023 at 3:51 pm. To the right of the main record view is a sidebar titled 'Activity' which displays a section for 'Upcoming & Overdue' activities.

Step 2 : - Create a Expense Items custom object.

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

The screenshot shows the 'Object Manager' screen in Salesforce. A new object named 'Expense Item' is being created. The 'Details' tab is selected. In the 'Fields & Relationships' sidebar, there are many options listed: Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Restriction Rules, Scoping Rules, Triggers, Flow Triggers, and Validation Rules. The main details area shows the API Name as 'Expense_Item__c', the singular label as 'Expense Item', and the plural label as 'Expense Items'. Other settings include enabling reports (checked), tracking activities, tracking field history, and deployment status set to 'Deployed'. There are also help settings pointing to the standard Salesforce help window.

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 configuration 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, Restriction Rules, Scoping Rules, Triggers, Flow Triggers, and Validation Rules. The main content 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 .

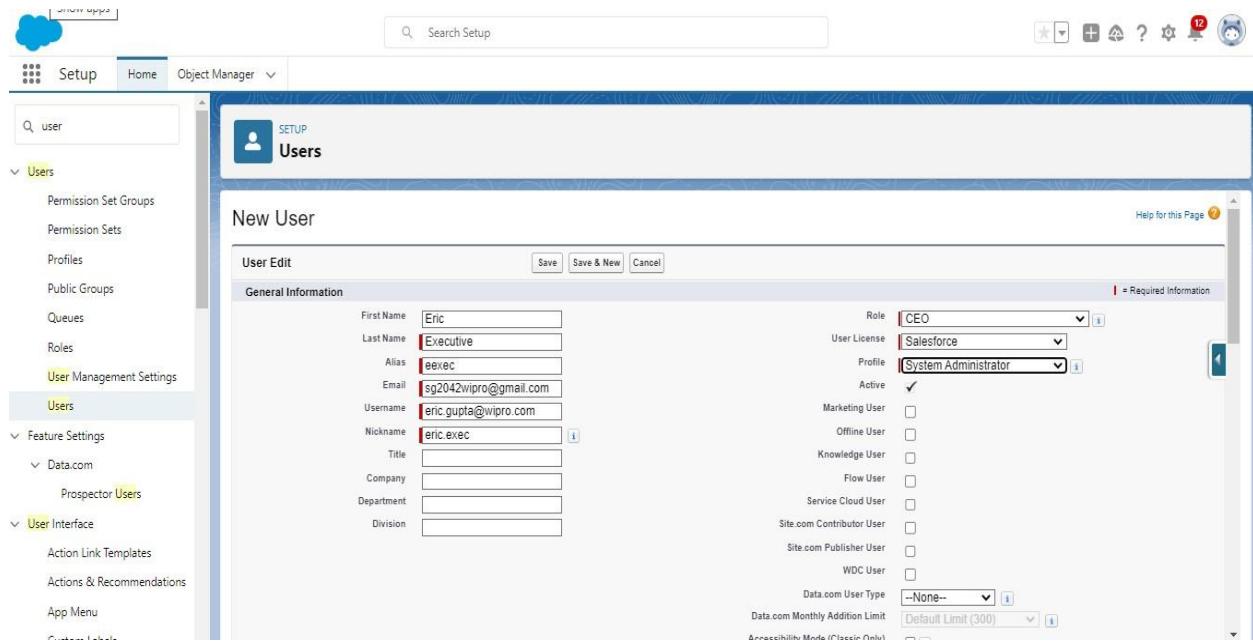
- Amount = 870, Expense Type = “Hotel”. Save.
- Amount = “450”, Expense Type = “Airfare” Save.

The screenshot shows the Travel App interface with the 'Travel Approvals' tab selected. The top navigation bar includes links for Travel App, Charter, Reports, Dashboards, Departments, and Travel Approvals. The main content area displays the 'Travel Approval' record for TA00001. The 'Related' tab is selected, showing the 'Notes & Attachments (0)' and 'Expense Items (2)' sections. The 'Expense Items' section lists two items: E-00001 and E-00002. The 'Details' tab is also visible. On the right side, there is an 'Activity' section with a toolbar for creating new activities (Email, Call, Task, etc.) and a status message indicating 'No past activity. Past meetings and tasks marked as done show up here.'

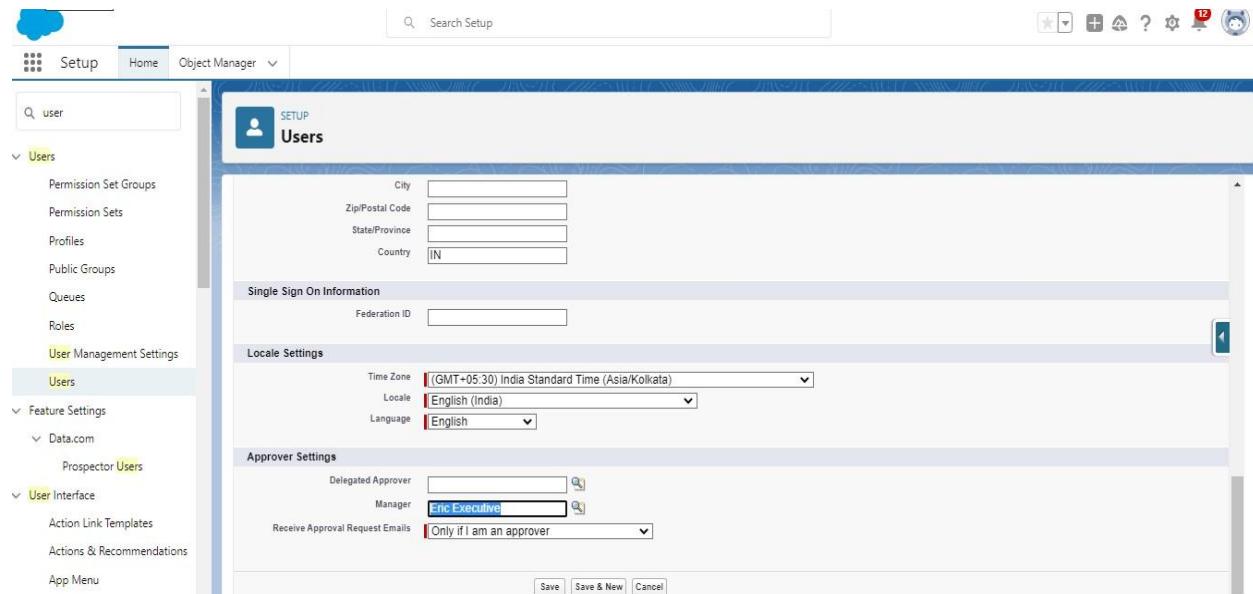
Step 5 :- Create a user.

- 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. On the left, a sidebar lists various customization options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts. 'Search Layouts' is currently selected. The main area is titled 'Edit Search Layout' and 'Travel Approval Search Results'. It displays a list of available fields on the left and selected fields on the right. Available fields include: 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, Purpose Of Trip, Department, Status, Destination State, Trip Start Date, and Trip End Date. The 'Trip End Date' field is highlighted in blue and has been moved from the available list to the selected list. Below the lists are buttons for 'Add', 'Remove', 'Up', 'Down', and a checkbox for 'Override the search result column customizations for all users'. Standard and Custom buttons sections are also present but are empty.

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 the 'Travel Approvals' object. The top navigation bar includes links for Chatter, Reports, Dashboards, Departments, and Travel Approvals. The main list shows one item: '1 TA00001'. A modal dialog box titled 'Select Fields to Display' is open over the list view. This dialog contains two columns: 'Available Fields' and 'Visible Fields'. The 'Available Fields' list includes: Out Of State, Owner Alias, Owner First Name, Owner Last Name, Purpose Of Trip, Record ID, Travel Approval #, Department, Created By, Status, Trip Start Date, and Trip End Date. The 'Visible Fields' list contains the same fields, with 'Travel Approval #' at the top. At the bottom of the dialog are 'Cancel' and 'Save' buttons. The background list view shows standard buttons like New, Import, Change Owner, and Printable View.

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 custom list view for 'Travel Approvals'. The title of the list is 'Open Out of State Travel Requests'. The list contains one item, TA00001, sorted by Travel Approval #. The filter sidebar on the right shows two filters: 'Out Of State*' equals True and 'Status*' not equal to Approved, Rejected.

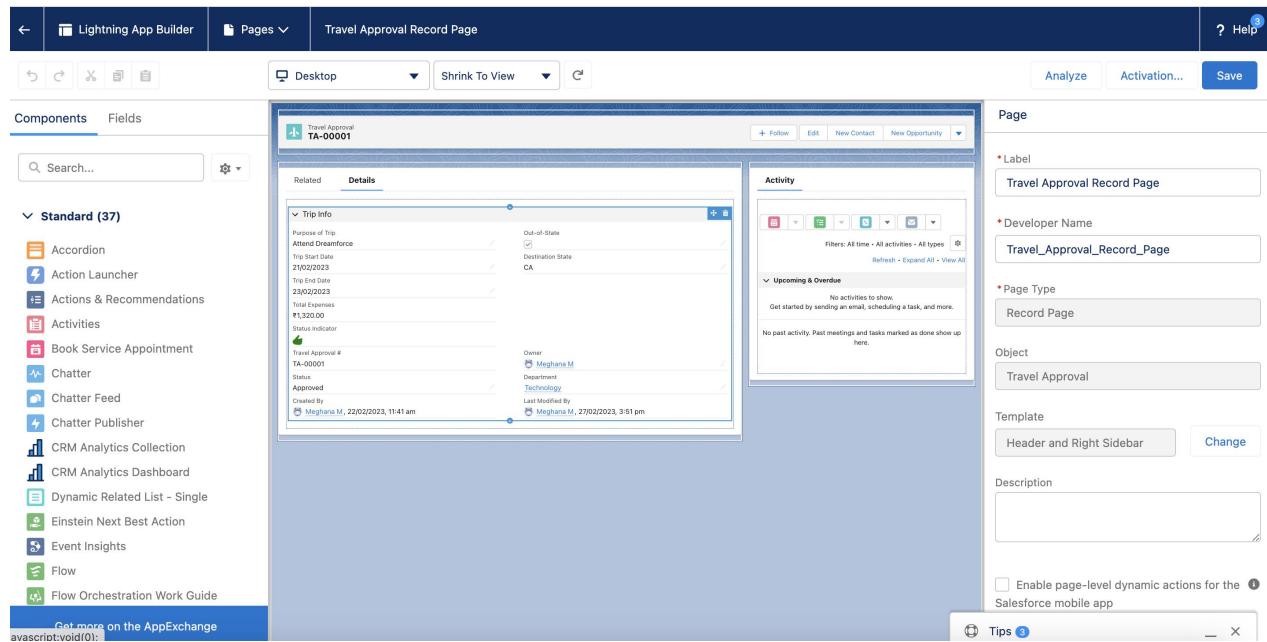
Step 10 :- Select fields to display in the Travel Approval “Open Out of State Travel Requests” List view, as shown in the screen shot & Save.

The screenshot shows 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 "Travel Approval # ↑" and "TA00001". The "Visible Fields" section includes "Department", "Created By", "Status", "Destination State", "Trip Start Date", and "Trip End Date". At the bottom of the modal are "Cancel" and "Save" buttons. In the background, a list titled "Open Out of State Travel Requests" is displayed, showing one item: "1 TA00001". The top navigation bar includes links for Travel App, Chatter, Reports, Dashboards, Departments, and Travel Approvals. The status bar at the bottom right shows a user icon with 12 notifications.

Step 11 :-

(a) 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.



(b) :- 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 :-

The screenshot shows the Salesforce Setup interface for the 'Travel Approval' object. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, and Lightning Record Pages. The 'Page Layouts' section is selected. The main area displays the 'Travel Approval Detail' page layout. A 'Fields' panel on the left contains sections for Buttons, Quick Actions, Mobile & Lightning Actions, Expanded Lookups, Related Lists, and Report Charts. The 'Related Lists' section is expanded, showing fields for Section, Destination State, Purpose Of Trip, Trip Start Date, Last Modified By, Status, Created By, Out Of State, Travel Approval #, Department, Owner, and Trip End Date. Below this is the 'Travel Approval Detail' form with sections for Information, Trip Info, and System Information. Buttons at the top of the form include Edit, Delete, Clone, Change Owner, Change Record Type, Printable View, Sharing, and Sharing Hierarchy.

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

The screenshot shows the Salesforce Setup interface for the 'Travel Approval' object, specifically customizing the 'Expense Items' related list. The left sidebar shows various setup options. The 'Page Layouts' section is selected. The main area shows the 'Related List Properties - Expense Items' dialog. Under 'Available Fields', fields like Created By, Created By Alias, Created Date, Last Modified By, Last Modified Date, Last Modified Alias, and Record ID are listed. Under 'Selected Fields', fields like Expense Item Number, Expense Type, Amount, and Total are selected. Buttons for 'Add', 'Up', 'Down', and 'Remove' are available to manage the list. Other sections in the dialog include 'Columns' (with a note about re-ordering), 'Buttons' (with OK, Cancel, and Revert to Defaults), and a preview pane showing sample expense item data.

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

Step 13 :-

(a)

- Enable Chatter on the Travel Approval Object.
- Enable “Feed Tracking” for Travel ApprovalObject. Select these 2 fields:
- Destination State Status Save.

(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 for the 'Travel App' tab. On the left sidebar, there are sections for 'What I Follow', 'To Me', 'Bookmarked', 'Company Highlights' (which is selected), 'My Drafts', 'STREAMS' (with a note about no streams yet), 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 right says 'Search...'. Below it, a text input field says 'Share an update...' with a 'Share' button. A 'Sort by:' dropdown is set to 'Top Posts'. A search bar says 'Search this feed...' with a magnifying glass icon. A post from 'TA-00001 — Meghana M' dated '22 February 2023 at 1:09 pm' asks 'Which department should I associate this travel request with?'. Meghana M commented 'Technology is the correct department.' Below the comment is a 'Write a comment...' input field. To the right, a blue sidebar titled 'Einstein Recommendations' lists three users: 'Eric Executive' (Your manager), 'Integration User' (Joined in the last week), and 'Security User' (Joined in the last week), each with a '+ Follow' and 'Skip' button.

Test the App , Let see how its look like

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

The screenshot shows the Salesforce interface for the Travel Approval app. At the top, there's a navigation bar with icons for Home, Chatter, Reports, Dashboards, Departments, and Travel Approvals. A search bar is at the top right. Below the navigation is a header bar with a travel icon, the text "Travel Approval TA-00001", and buttons for "+ Follow", "Edit", "New Contact", and "New Opportunity".

The main area is divided into two sections: "Related" and "Details". The "Related" section contains a "Trip Info" section with fields for Purpose of Trip (Attend Dreamforce), Out-of-State (checkbox checked), Trip Start Date (21/02/2023), Destination State (CA), Trip End Date (23/02/2023), Total Expenses (₹1,320.00), Status Indicator (green checkmark), Travel Approval # (TA-00001), Status (Approved), and Created By (Meghana M, 22/02/2023, 11:41 am). The "Details" section includes fields for Owner (Meghana M), Department (Technology), and Last Modified By (Meghana M, 27/02/2023, 3:51 pm).

To the right, there's a "Activity" section with a Chatter feed, showing a message from Meghana M: "No activities to show. Get started by sending an email, scheduling a task, and more." It also indicates "No past activity. Past meetings and tasks marked as done show up here."

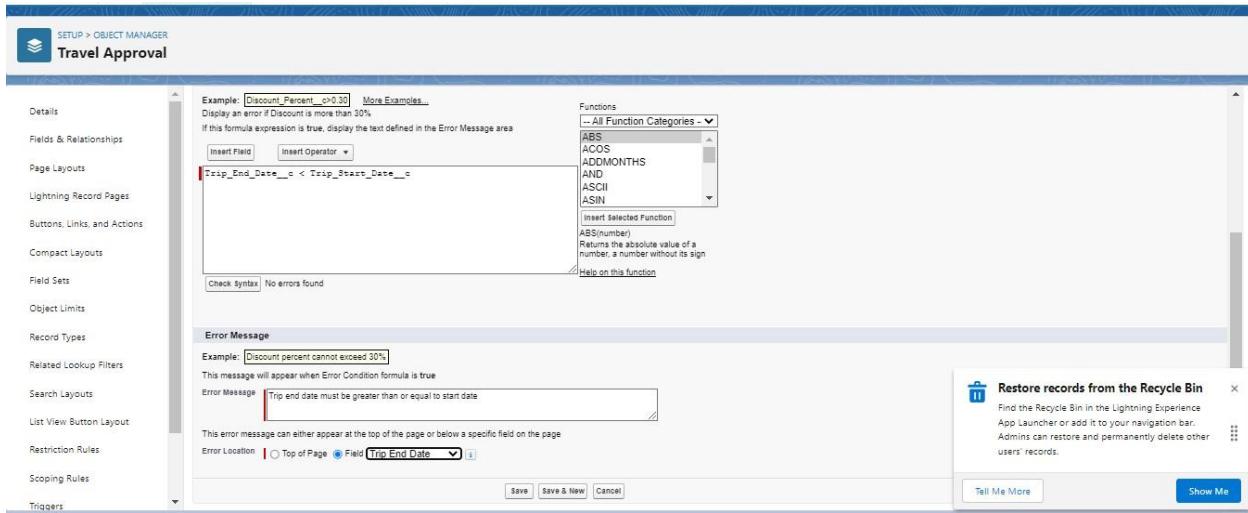
After completing the exercise 2 you will see the user can post their query by using chatter and other can reply to that query. Now, Here Our module 1 is completed lets move to our next module 2.

Module – 2

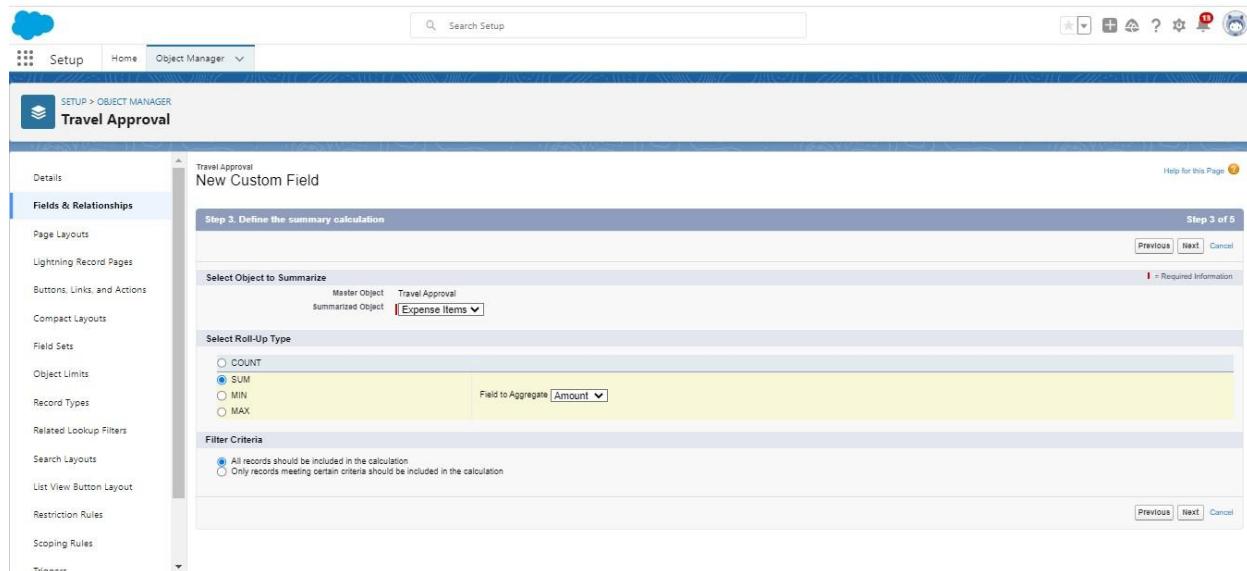
Exercise 1: -

Step 1:- Create Validation Rule.

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

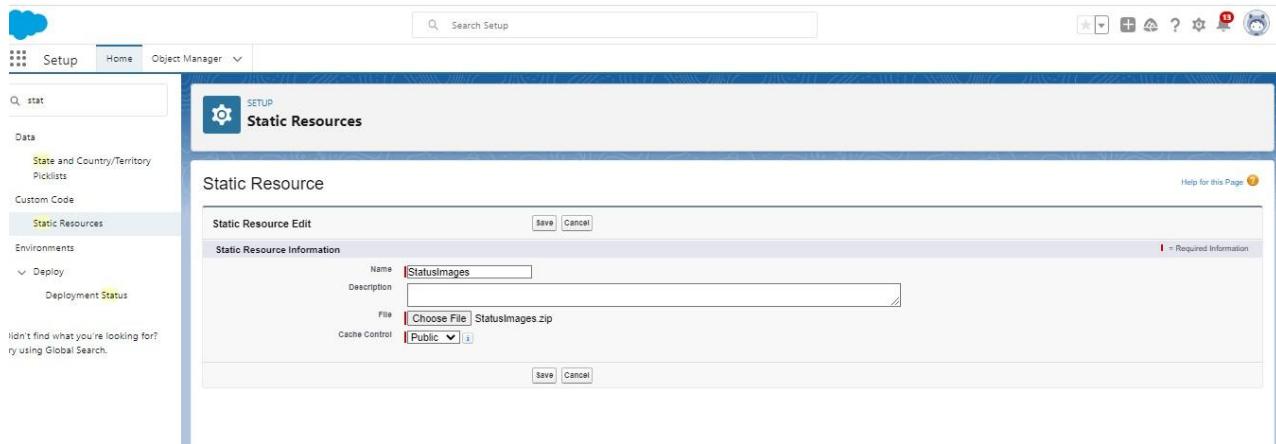


Step 2 :- Create a Roll-Up Summary Field on Travel Approval object.



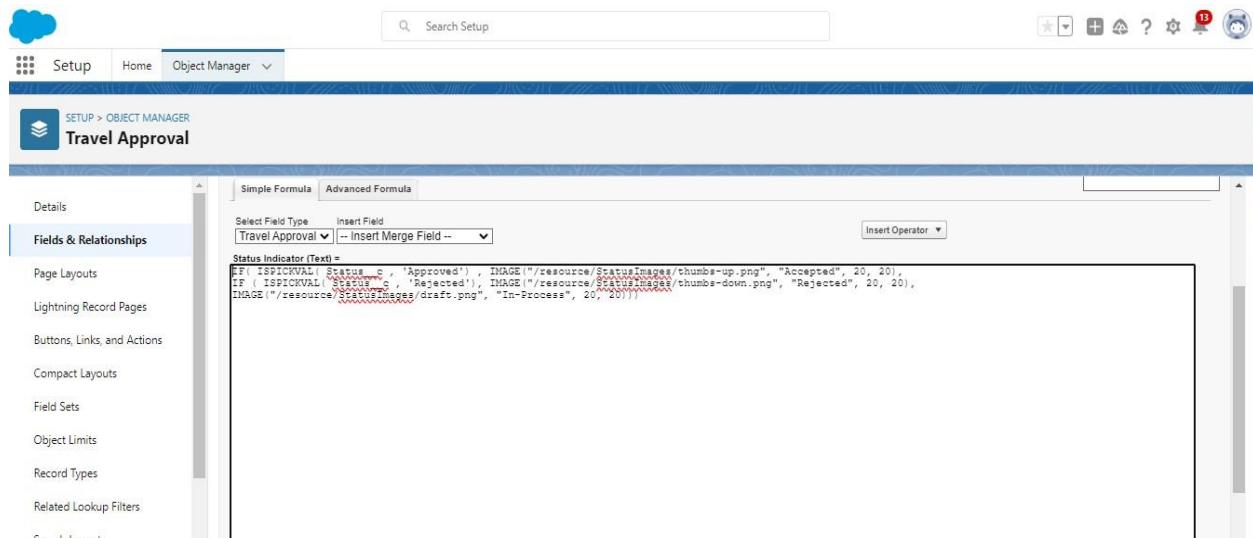
Step 3 :- Business Logic: Create a field that shows a visual indicator based on the value of the Status field

- 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.

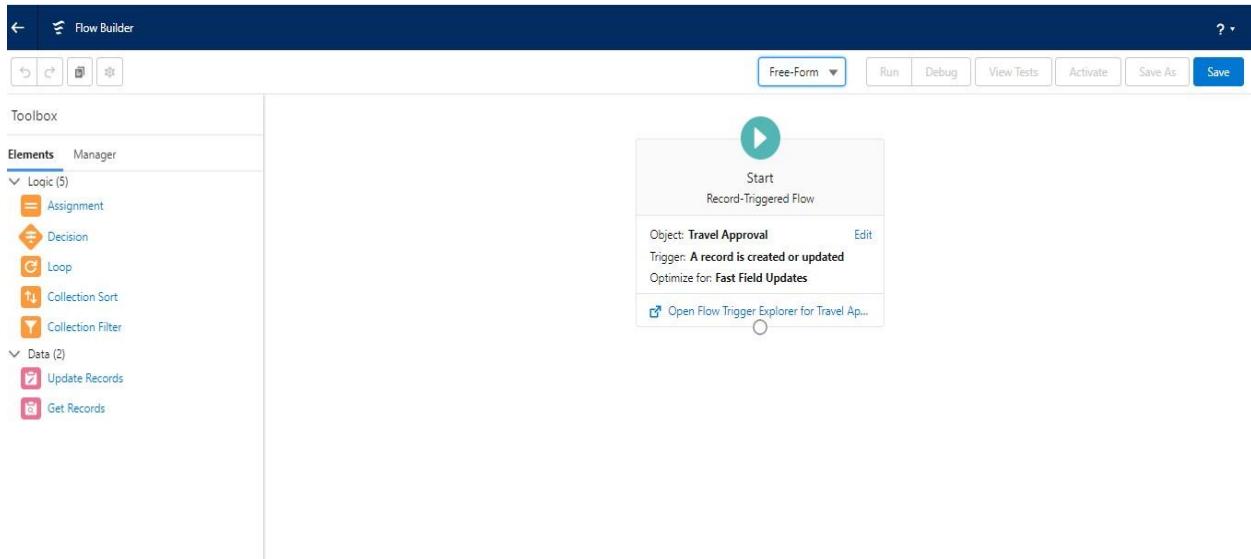
- Field Label: Status Indicator
- Formula Return Type = Text
- Formula: IF(ISPICKVAL(Status c , 'Approved') ,
IMAGE("/resource/StatusImages/thumbs-up.png", "Accepted", 20, 20),
IF (ISPICKVAL(Status c , 'Rejected'),
IMAGE("/resource/StatusImages/thumbs-down.png", "Rejected", 20, 20),
IMAGE("/resource/StatusImages/draft.png",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



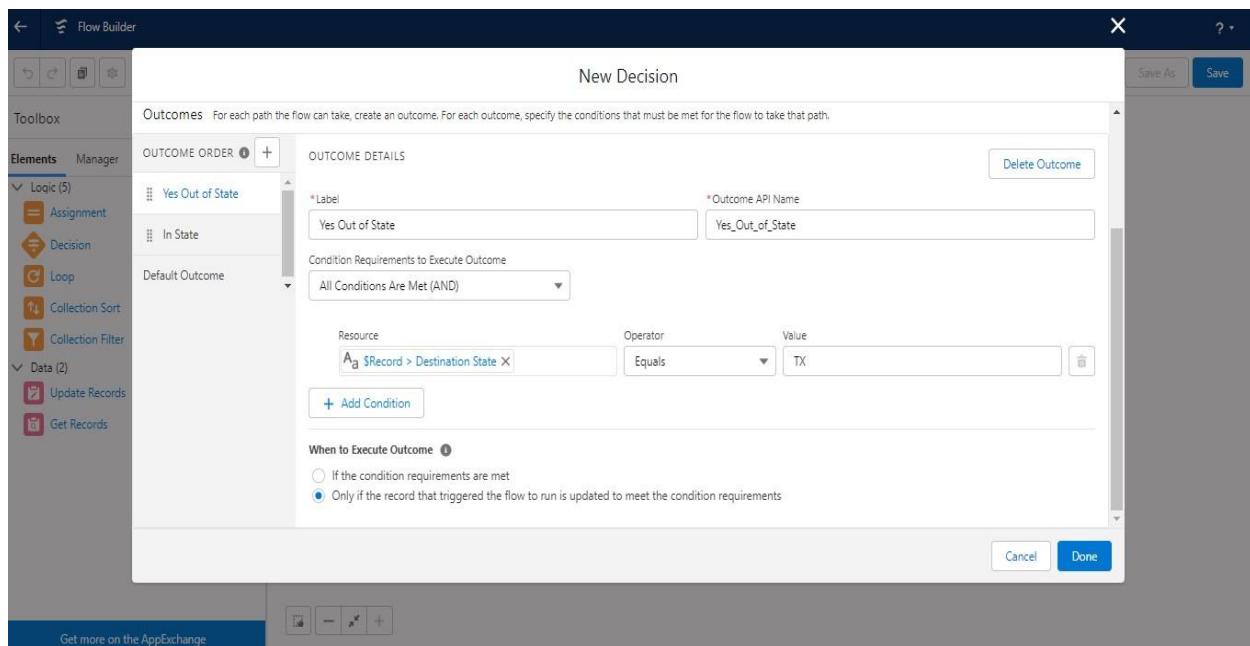
Step 5 :-

(a).Create a Record – Triggered Flow.

- Flow should look like this:

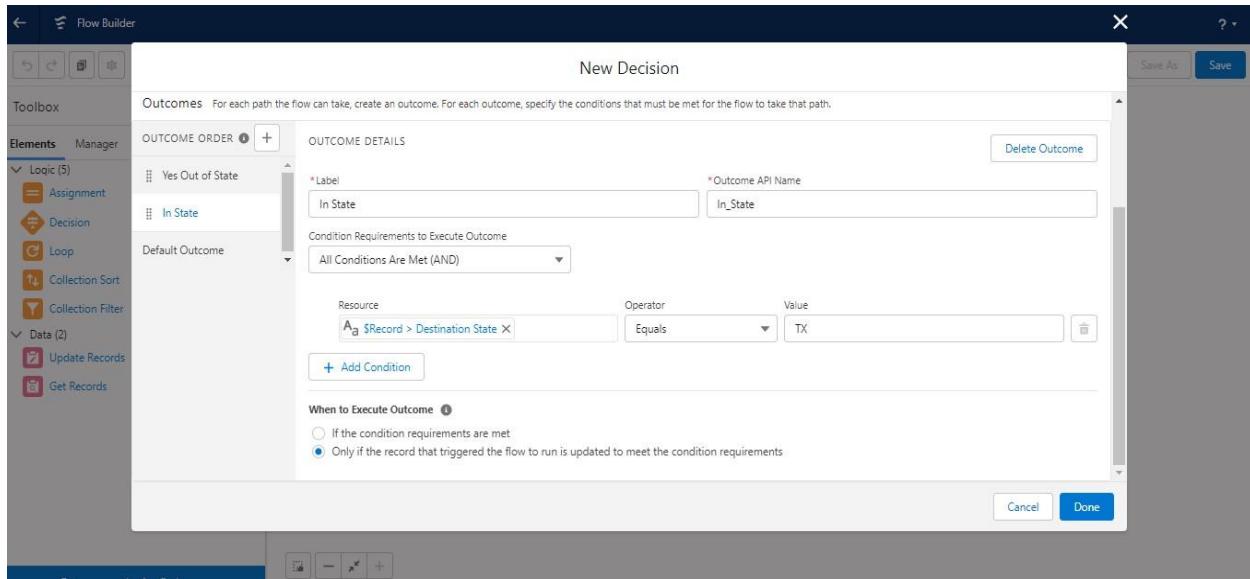


(b). Add a Decision Element to the Flow.

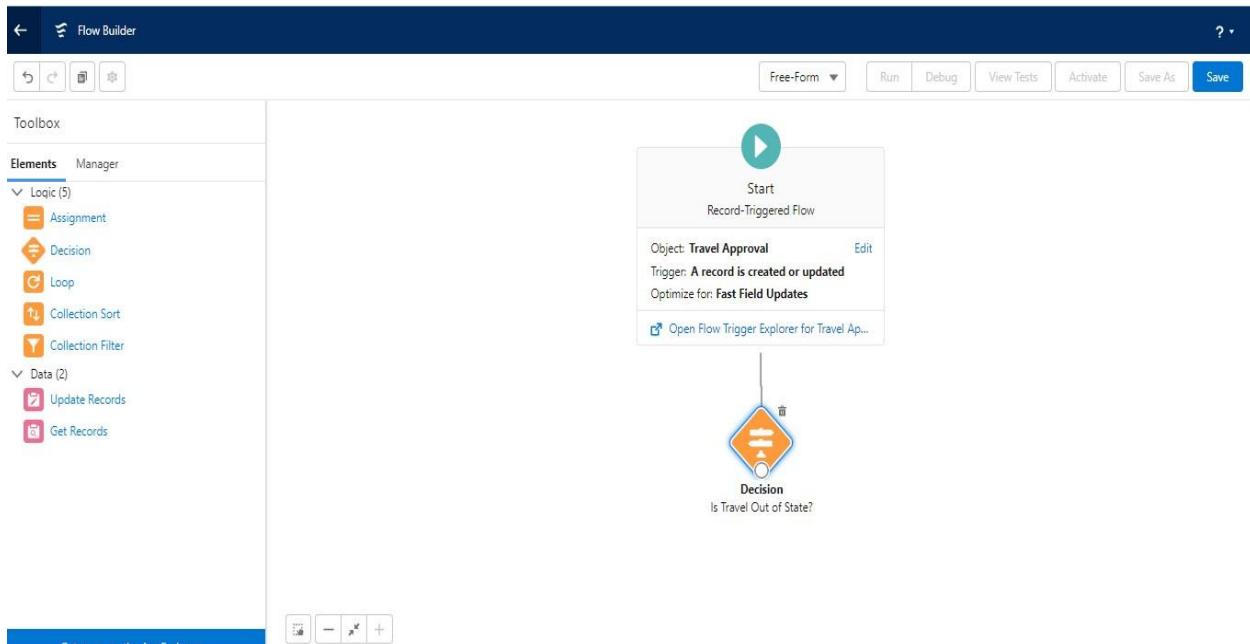


(c). Next to Outcome Order click the + button to add another outcome.

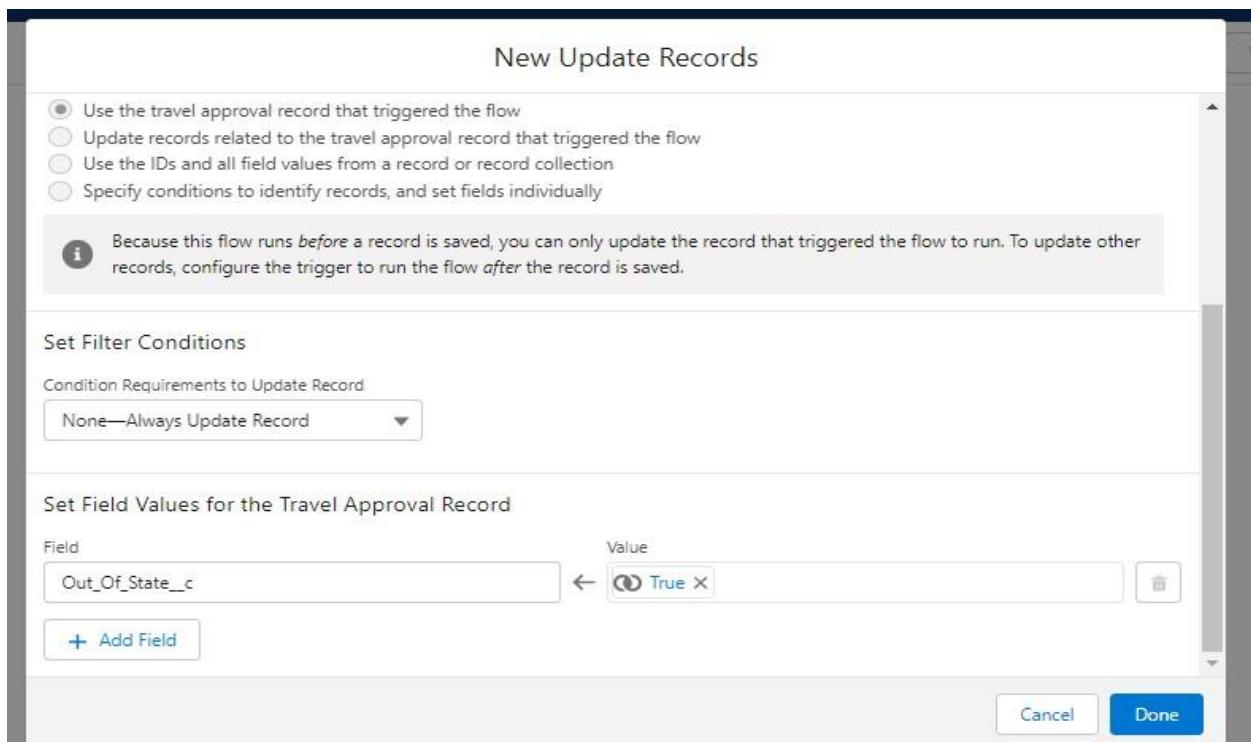
- Flow should look like this:



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

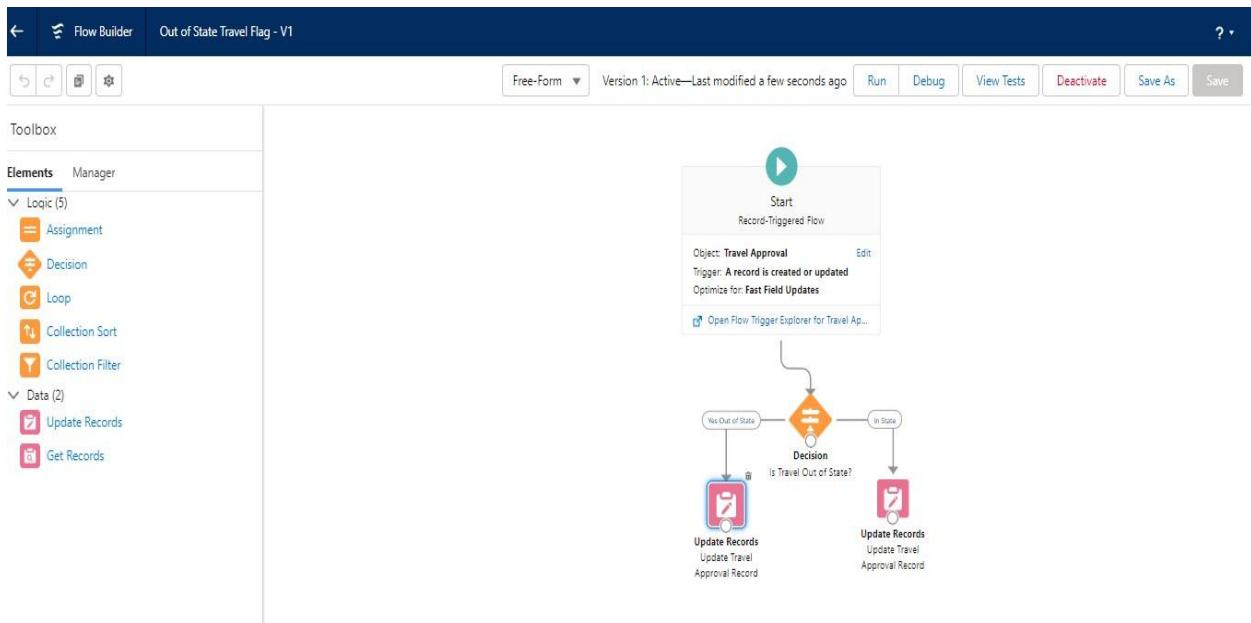


(e). Create an Action for the Flow Using Update Records Elements.



(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 under the 'Approval Processes' section. A specific approval process is selected: 'Travel Approval: Travel Approval Request'. The process definition detail includes:

- Process Name:** Travel Approval Request
- Unique Name:** Travel_Approval_Request
- Description:** Travel Approval: Total Expenses GREATER THAN 0
- Entry Criteria:** Travel Approval: Total Expenses GREATER THAN 0
- Record Editability:** Administrator ONLY
- Approval Assignment Email Template:** Initial Submitters
- Initial Submitters:** Travel Approval Owner
- Created By:** Meghana M, 22/02/2023, 7:56 pm
- Active:** ✓
- Next Automated Approver Determined By:** Manager of Record Submitter
- Allow Submitters to Recall Approval Requests:**
- Modified By:** Meghana M, 22/02/2023, 8:52 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
--------	-------------	------	-------------	----------	-------------------	-----------------

(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:

The screenshot shows the Travel App interface with a travel approval record. The record details are:

- Submitter:** Meghana M
- Date Submitted:** 22-Feb-2023
- Actual Approver:** Meghana M
- Assigned To:** Eric Executive

Details:

Approval Details	
Travel Approval #	Owner
TA-00001	Meghana M

Approver Comments:

Meghana M
approved status
22-Feb-2023, 9:05:09 pm

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:

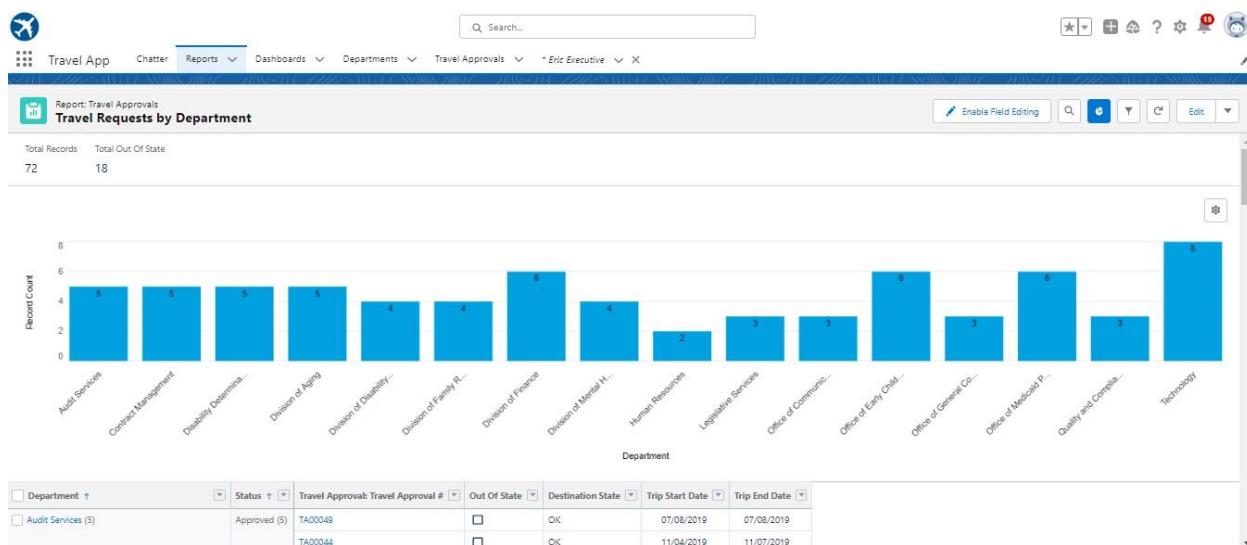
The screenshot shows a Salesforce page titled "Travel Approvals". At the top, there's a search bar and a toolbar with various icons. Below the header, a message says "50+ items • Sorted by Travel Approval # • Filtered by All travel approvals • Updated a few seconds ago". The main area is a table with columns: "Travel Approval #", "Department", "Created By", "Status", "Trip Start Date", and "Trip End Date". The table lists 15 travel approvals from different departments, all created by Sagar Gupta and mostly in the "Approved" status. The last entry is for "Division of Disability and Rehabilitative Services" with a status of "Announced".

	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 Disability and Rehabilitation Services	Sagar Gupta	Announced	10/11/2019	10/12/2019

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

The Report

- Report might look as per the screen shot:



Step 3 :- Create a Travel Requests by Month Report

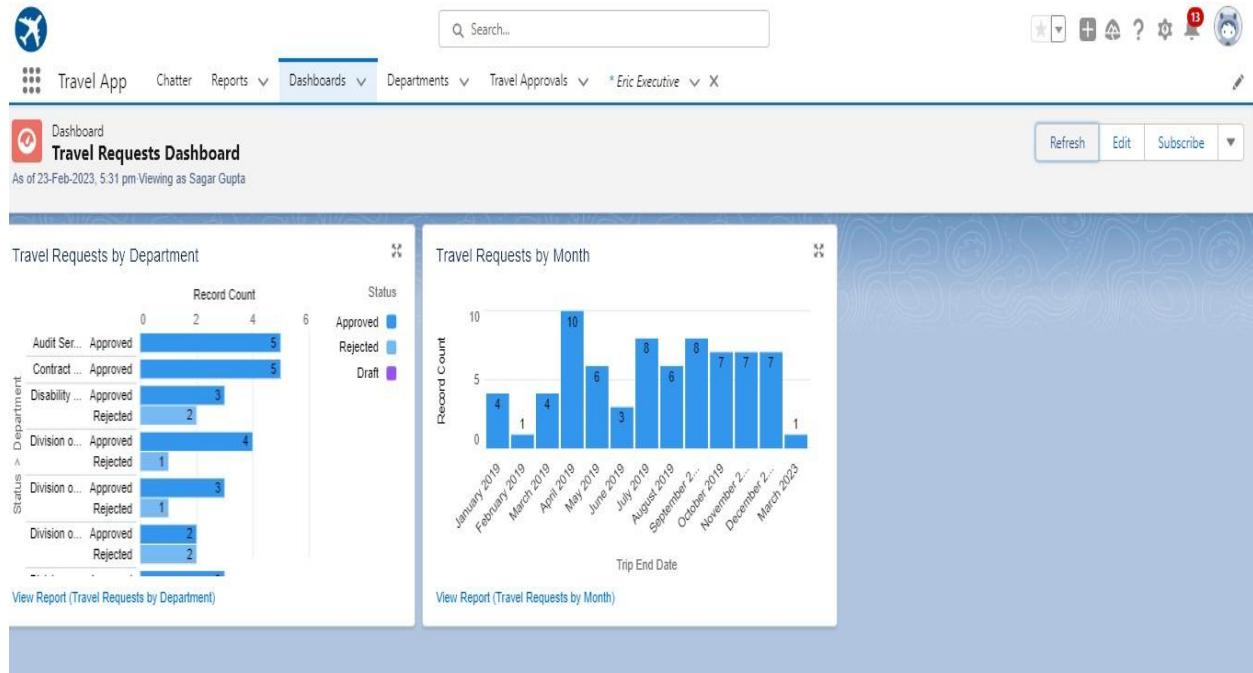
- Test The Report
- Report might look as per the screen shot:

The screenshot shows a Salesforce report titled "Report: Travel Approvals" with the specific view "Travel Requests by Month". The report displays a table of travel approvals across three months: January 2019, February 2019, and March 2019. The columns include Trip End Date, Out Of State, Travel Approval, Travel Approval #, Department, Status, Destination State, and Trip Start Date. The data shows several entries for each month, with subtotals for each month and a grand total for all entries.

Trip End Date ↑	Out Of State ↑	Travel Approval: Travel Approval #	Department	Status	Destination State	Trip Start Date
January 2019 (4)	<input type="checkbox"/> (1)	TA00002	Disability Determination Bureau	Approved	OK	10/01/2019
	Subtotal					
	<input checked="" type="checkbox"/> (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)	<input type="checkbox"/> (1)	TA00062	Division of Family Resources	Approved	FL	12/02/2019
	Subtotal					
March 2019 (4)	<input type="checkbox"/> (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 : -

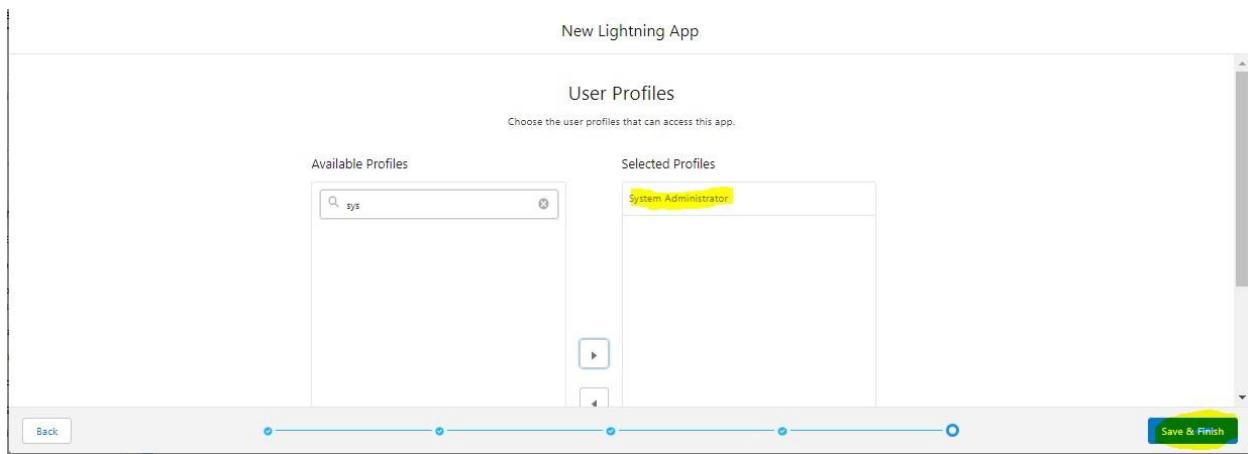
Step 1:- Create a new custom lightning App, name: Code Playground.

App Settings

App Details & Branding

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>i</small> <input type="text" value="Code Playground"/>	Image <small>i</small>  <small>Clear</small>
* Developer Name <small>i</small> <input type="text" value="Code_Playground"/>	Primary Color Hex Value <small>i</small> <input type="color" value="#0070D2"/>
Description <small>i</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
App Launcher Preview	
	



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

The image contains two screenshots from the Salesforce Setup interface.

Screenshot 1: New Custom Object

This screenshot shows the 'Custom Object Definition Edit' page for creating a new custom object named 'Customer'. The 'Label' field is set to 'Customer' and the 'Plural Label' field is set to 'Customers'. The 'Description' field is empty. A note at the top states: 'Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles.' Buttons at the top right include 'Save', 'Save & New', and 'Cancel'.

Screenshot 2: Tabs

This screenshot shows the 'New Custom Object Tab' configuration. It is Step 1 of 3. The user has selected the 'Customer' object and chosen the 'Credit card' tab style. An optional field for a 'Splash Page Custom Link' is set to '--None--'. A note at the top says: 'Choose the custom object for this new custom tab. Fill in other details.' A sidebar on the left shows tabs for 'User Interface' and 'Tabs'.

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

The screenshot shows the Salesforce Setup interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'Customer'. On the left sidebar, under 'Fields & Relationships', several options are listed: Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts. The central panel displays 'Step 2. Enter the details' for creating a new field. The 'Field Label' is set to 'Active'. The 'Default Value' is 'Unchecked' (radio button selected). The 'Field Name' is also 'Active'. There are fields for 'Description' and 'Help Text', both of which are currently empty. A checkbox at the bottom indicates 'Add this field to existing custom report types that contain this entity'. Navigation buttons 'Previous', 'Next', and 'Cancel' are visible at the bottom right of the form.

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

The screenshot shows the Salesforce Setup interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'Customer'. On the left sidebar, under 'Fields & Relationships', several options are listed: 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, and Restriction Rules. The central panel displays 'Step 2. Enter the details' for creating a new field. The 'Field Label' is 'Customer Type'. The 'Values' section has 'Enter values, with each value separated by a new line' selected. In the text area below, the values 'Premium' and 'Standard' are listed. There are checkboxes for 'Display values alphabetically, not in the order entered' and 'Use first value as default value'. The 'Restrict picklist to the values defined in the value set' checkbox is checked. The 'Field Name' is 'Customer_Type' and the 'Description' is empty. Navigation buttons 'Previous', 'Next', and 'Cancel' are visible at the bottom right of the form.

c. Label = Description, Text Area, Save.

Customer
New Custom Field

Step 2. Enter the details

Field Label: **Description**

Field Name: **Description**

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

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.RecordAPIName.FieldName__c}

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: **Customer**

Step 4:- Create a Custom Object for Billing.

SETUP Home Object Manager

New Custom Object

Help for this Page

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

Custom Object Information

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

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

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

Object Name	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-

Save Save & New Cancel

SETUP Tabs

New Custom Object Tab

Help for this Page

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	Billing	<input type="button" value=""/>
--------	---------	---------------------------------

Tab Style

Bank	<input type="button" value=""/>
------	---------------------------------

(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

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

Edit Billing Custom Field
Amount Paid

Custom Field Definition Edit

Field Information

Field Label	Amount Paid	Data Type	Currency
Field Name	Amount_Paid	= Required Information	
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

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

Setup > OBJECT MANAGER
Billing

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

Edit Billing Custom Field
Customer Type

Custom Field Definition Edit

Field Information

Field Label	Customer Type	Data Type	Picklist
Field Name	Customer_Type	= Required Information	
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

Use formula syntax. Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes.
NOTE: always enclose values in double quotes (e.g.) and enclose date references in the standard format (MM/DD/YYYY).

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

SETUP > OBJECT MANAGER

Billing

Field Information

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

Field Label: Status

Field Name: Status

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:

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 user: \$CustomMetadataType__mdt.RecordAPIName.Field__c

Test The App

- Code Playground App should look like this:

Accounts

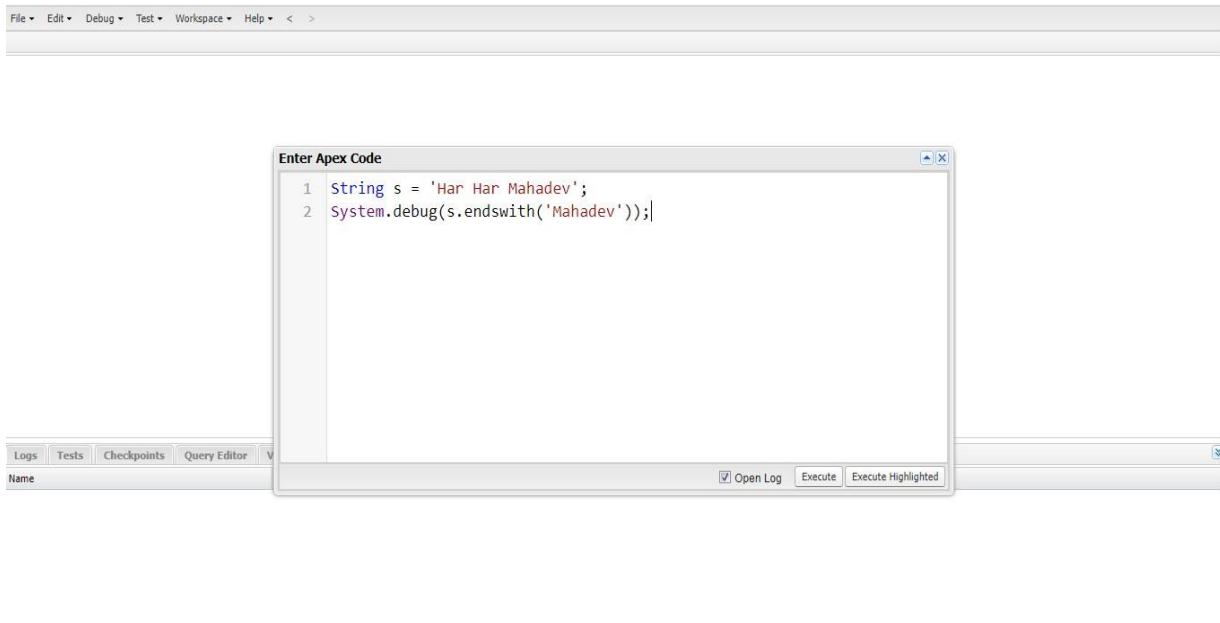
All Accounts

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

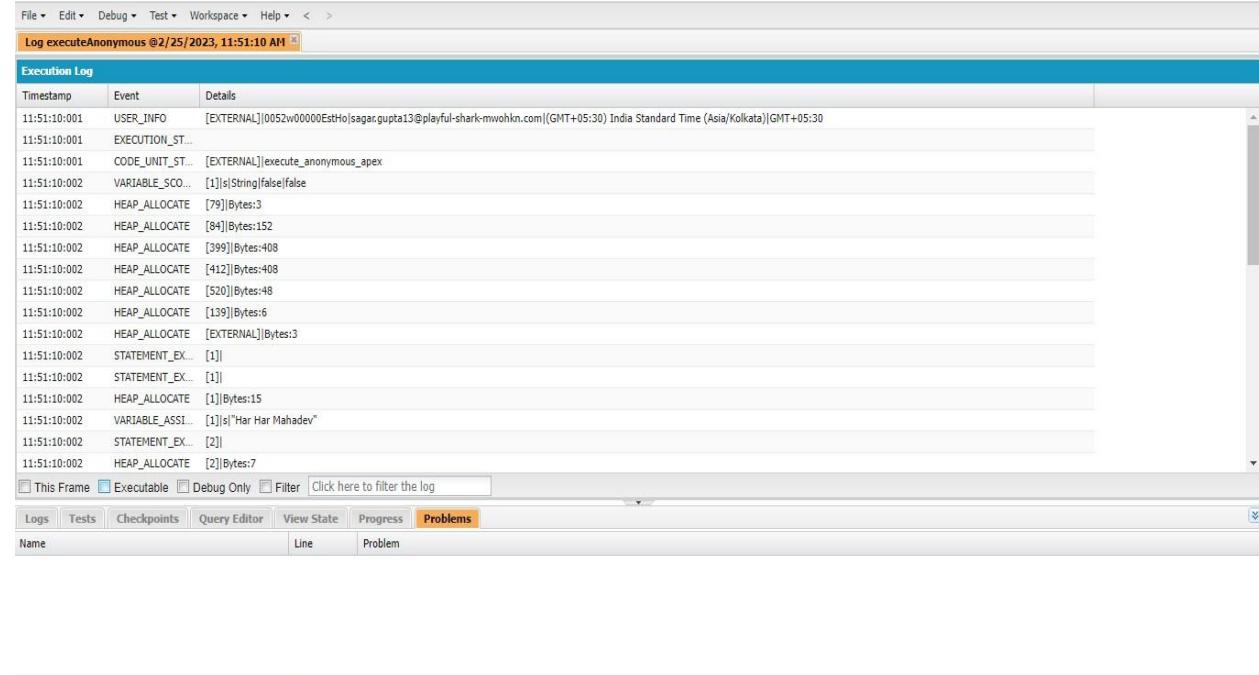
	Account Name	Account Site	Billing State/Province	Phone	Type	Account Owner Alias
1	Burlington Textiles Corp of America		NC	(336) 222-7000	Customer - Direct	SGupt
2	Dickenson plc		KS	(785) 241-6200	Customer - Channel	SGupt
3	Edge Communications		TX	(512) 757-6000	Customer - Direct	SGupt
4	Express Logistics 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					autproc
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 Problems tab which is currently selected. The Problems tab has sub-tabs for Name, Line, and Problem.

Output :

The screenshot shows the Salesforce Developer Console interface after the code has been executed. The top part of the screen displays the execution log:

Log executeAnonymous @2/25/2023, 12:00:59 PM

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 several filter options: This Frame, Executable, Debug Only, Filter, and Click here to filter the log. Below these filters is another set of tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems, with the Problems tab being the active one. The Problems tab has sub-tabs for Name, Line, and 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

Name Line Problem

Open Log Execute Execute Highlighted

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 Developer Console interface. 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 contains an 'Enter Apex Code' dialog box with the following code:

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

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

Output :

The screenshot shows the Salesforce Developer Console interface. 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, 12:15:57 PM'. Below the tabs is a header bar with 'File', 'Edit', 'Debug', 'Test', 'Workspace', 'Help', and other buttons. The main area contains an 'Execution Log' table with one entry:

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

Below the log is a 'Logs' tab section with 'This Frame', 'Executable', 'Debug Only' (checked), 'Filter', and 'Click here to filter the log' buttons. The 'Problems' tab is selected. The status bar at the bottom shows 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'.

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 Execute Execute Highlighted

Output :

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

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

Execution Log

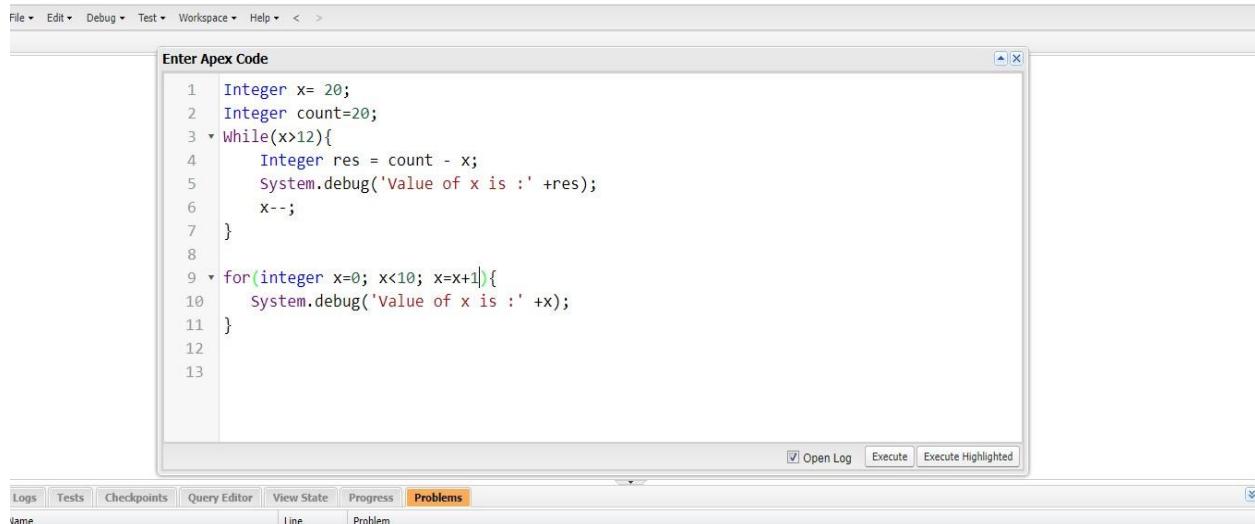
Timestamp	Event	Details
12:24:55:003	USER_DEBUG	[8]DEBUG (1, 2, 3, 4)
12:24:55:003	USER_DEBUG	[10]DEBUG 2
12:24:55:003	USER_DEBUG	[13]DEBUG (1, 2, 3, 40)
12:24:55:003	USER_DEBUG	[16]DEBUG ()

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

Logs Tests Checkpoints Query Editor View State Progress Problems

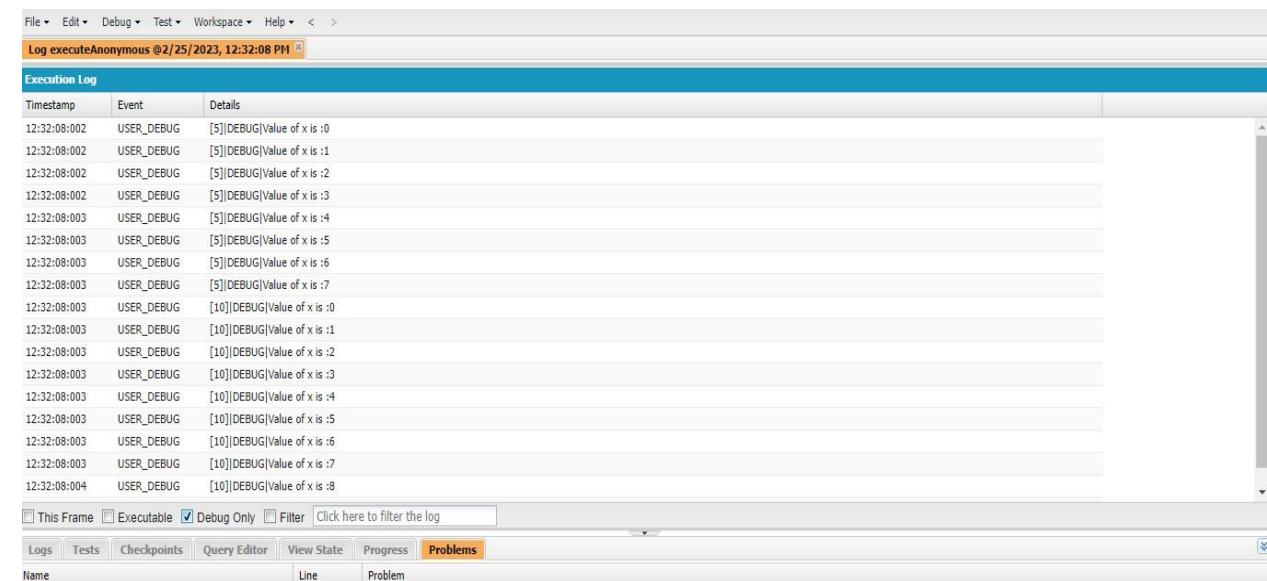
Name Line Problem

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 is a menu bar with File, Edit, Debug, Test, Workspace, Help, and navigation buttons. Below the menu is a toolbar with icons for New, Open, Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and Delete. The main area contains an "Enter Apex Code" window with the following code:

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

Below the code editor is a "Problems" tab in the toolbar. The Problems tab has three tabs: Name, Line, and Problem. The Line tab is selected. At the bottom of the interface are buttons for Open Log, Execute, and Execute Highlighted.

Output :

The screenshot shows the Salesforce IDE interface. At the top is a menu bar with File, Edit, Debug, Test, Workspace, Help, and navigation buttons. Below the menu is a toolbar with icons for New, Open, Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and Delete. The main area contains an "Execution Log" window with the following log entry:

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

Below the log is a "Problems" tab in the toolbar. The Problems tab has three tabs: Name, Line, and Problem. The Line tab is selected. At the bottom of the interface are buttons for This Frame, Executable, Debug Only, Filter, Click here to filter the log, Open Log, Execute, and Execute Highlighted.

Answer the following in True Or False:

Exercise 4 :-

The screenshot shows the Salesforce IDE interface. In the top navigation bar, the 'Debug' dropdown is selected. A modal window titled 'Enter Apex Code' contains the following code:

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

Below the modal, the 'Problems' tab is selected in the bottom navigation bar. The status bar at the bottom right includes buttons for 'Open Log', 'Execute', and 'Execute Highlighted'.

Output :

The screenshot shows the Salesforce IDE interface. The 'Logs' tab is selected in the bottom navigation bar. The execution log table displays the following entry:

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

At the bottom of the screen, there is a toolbar with checkboxes for 'This Frame', 'Executable', 'Debug Only', and 'Filter', followed by a link 'Click here to filter the log'. The 'Problems' tab is also visible in the bottom navigation bar.

Answer the following in True Or False:

Exercise 5 :-

The screenshot shows the Salesforce IDE interface. In the top navigation bar, the 'Debug' item is selected. Below it, the 'Enter Apex Code' window contains the following Apex code:

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

At the bottom of the 'Enter Apex Code' window, there are three buttons: 'Open Log' (with a checked checkbox), 'Execute', and 'Execute Highlighted'. Below this window, the 'Problems' tab is selected in the navigation bar. The 'Logs' tab is also visible. The 'Problems' section is currently empty.

Output :

The screenshot shows the 'Execution Log' window. The title bar indicates the log was generated at '2/25/2023, 12:40:12 PM'. The window has a header 'Execution Log' and a table with columns: 'Timestamp', 'Event', and 'Details'. One entry is present:

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

At the bottom of the 'Execution Log' window, there are several filter options: 'This Frame', 'Executable', 'Debug Only' (which is checked), 'Filter', and 'Click here to filter the log'. Below the log window, the 'Problems' tab is selected in the navigation bar, and the 'Logs' tab is also visible.

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

Exercise 6 :-

The screenshot shows the Salesforce IDE interface. At the top, there's a toolbar with 'File', 'Help', and navigation icons. Below it is a header bar with 'Click to go back, hold to see history' and other navigation links. The main area is titled 'Enter Apex Code' and contains the following Apex code:

```
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 }
```

At the bottom of the code editor are buttons for '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. Underneath is a search bar with 'Name' and filters for 'Line' and 'Problem'.

Output :

The screenshot shows the Salesforce IDE interface with the 'Execution Log' tab selected. The log table has columns: Timestamp, Event, and Details. One entry is visible:

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

At the top, there's a toolbar with 'File', 'Edit', 'Debug', 'Test', 'Workspace', and 'Help'. Below the toolbar is a message bar: 'Log executeAnonymous @2/25/2023, 12:45:28 PM'. At the bottom is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The 'Problems' tab is selected. There's also a filter bar at the very bottom with options: 'This Frame', 'Executable', 'Debug Only', 'Filter', and a 'Click here to filter the log' link.

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 titled "Billings All" with 4 items. The columns are Bill Number, Customer Type, and Status. The data is as follows:

Bill Number	Customer Type	Status
1 B - 0001	Premium	Paid
2 B - 0002	Standard	Paid
3 B - 0003	Premium	Unpaid
4 B - 0004	Standard	Unpaid

The main area displays the Apex code for "Billing.apxc":

```
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 }
14 }
```

Below the code editor are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected. At the bottom, there are buttons for Name, Line, and Problem.

Output :

The screenshot shows the Salesforce Execution Log and the Enter Apex Code interface. The Execution Log table has columns for Timestamp, Event, and Details. The log entries are:

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)

The Enter Apex Code window contains the code: "1 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. At the top, there's a menu bar with File, Edit, Debug, Test, Workspace, Help, and a Go To button. Below the menu is a toolbar with Code Coverage: None, API Version: 57, and a dropdown arrow. The main area contains the code for a class named DiscountClass:

```
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 }
```

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 a table with columns Name, Line, and Problem.

Output :

The screenshot shows the Salesforce IDE interface with the Execution Log and Enter Apex Code dialog open.

The Execution Log window at the top shows a single entry:

Timestamp	Event	Details
13:21:51:011	USER_DEBUG	[2]!DEBUG finalPrice90.0

The Enter Apex Code dialog box in the center contains the following 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.

Below the dialogs, there's a log viewer with a checkbox for This Frame, a checkbox for Executable, a checked checkbox for Debug Only, a Filter button, and a Click here to filter the log link. There are also tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems, with 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 Trailhead developer console interface. At the top, there are tabs for InterfaceExample.apxc, PremiumCustomer.apxc, and normalCustomer.apxc. A message bar indicates "Log executeAnonymous @2/25/2023, 4:37:56 PM". Below this 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	[5]DEBUG Discount in Percentage From Normal10.0%

Below the log is the "Enter Apex Code" editor 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 editor 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 selected.

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

The screenshot shows the Salesforce Trailhead developer console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a search bar. The current tab is "DML.apxc". The code coverage is set to "None" and the API version is "57". The code in the editor is as follows:

```
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 the Problems tab selected.

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

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

Execution Log

Timestamp	Event	Details
23:37:30:001	USER_INFO	[EXTERNAL][0052v00000EstHo sagar.gupta13@playful-shark-mwohkn.com (GMT+05:30) India Standard Time (Asia/Kolkata) (GMT+05:30)
23:37:30:002	EXECUTION_STARTED	
23:37:30:002	CODE_UNIT_STARTED	[EXTERNAL]execute_anonymous_apex
23:37:30:002	VARIABLE_SCOPE	[1]cust Customer__c true false
23:37:30:002	HEAP_ALLOCATE	[79]Bytes:3
23:37:30:002	HEAP_ALLOCATE	[84]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	[320]Bytes:48
23:37:30:002	HEAP_ALLOCATE	[139]Bytes:6
23:37:30:002	HEAP_ALLOCATE	[EXTERNAL]Bytes:7
23:37:30:003	STATEMENT_EXECUTE	[1]
23:37:30:003	STATEMENT_EXECUTE	[1]
23:37:30:003	HEAP_ALLOCATE	[1]Bytes:4
23:37:30:003	VARIABLE_ASSIGNMENT	[1]cust:{} 0x7d6306ae
23:37:30:003	STATEMENT_EXECUTE	[2]
23:37:30:003	HEAP_ALLOCATE	[2]Bytes:5
23:37:30:003	VARIABLE_ASSIGNMENT	[2] this.Name "Wipro" 0x7d6306ae
23:37:30:003	STATEMENT_EXECUTE	[3]
23:37:30:003	HEAP_ALLOCATE	[3]Bytes:7
23:37:30:004	VARIABLE_ASSIGNMENT	[3] this.Customer_Type__c "Premium" 0x7d6306ae
23:37:30:004	STATEMENT_EXECUTE	[4]
23:37:30:004	HEAP_ALLOCATE	[52]Bytes:5

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

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

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

Output :

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 bar is a search bar and a toolbar with various icons. The main area displays a list titled 'Customers' with a dropdown menu showing 'All'. The list is sorted by 'Customer Name' and filtered by 'All customers'. A single record is visible: 'Wipro' with 'Premium' as the 'Customer Type'. There are buttons for 'New', 'Import', 'Change Owner', and 'Printable View' at the top right of the list area.

The screenshot shows the Salesforce Developer Console interface. At the top, there's a navigation bar with File, Edit, Debug, Test, Workspace, and Help. Below it is a toolbar with various icons. The main area has tabs for Execution Log, Details, and a large code editor window. The code editor contains 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

```

Below the code editor is a log viewer with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected. The log shows a single entry:

[5] [DEBUG] Successfully inserted Billing@012w000017zbEAA0

Output :

The screenshot shows the Salesforce Lightning Experience. At the top, there's a navigation bar with a star icon, a search bar, and various user icons. Below it is a header for "Travel App" with links for Chatter, Reports, Dashboards, Departments, Travel Approvals, and a specific record ID "B - 0004". The main content area displays a "Billing" record detail page for "B - 0004". The "Details" tab is selected. The record contains the following fields:

- Bill Number: B - 0004
- Amount Paid: ₹50,00,000
- Customer Type: (empty)
- Status: Paid
- Created By: Meghana M, 25/02/2023, 11:52 am
- Owner: Meghana M
- Last Modified By: Meghana M, 25/02/2023, 11:52 am

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

Output :

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

DML.apocx [] 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
0062v00000KCU1tAAH	125000	Negotiation/Review	United Oil & Gas Corp.	Energy	http://www.us.com
0062v00000KCU1tAAH	270000	Proposal/Price Quote	United Oil & Gas Corp.	Energy	http://www.us.com
0062v00000KCU1yAAH	120000	Closed Won	United Oil & Gas Corp.	Energy	http://www.us.com
0062v00000KCU26AAH	270000	Negotiation/Review	United Oil & Gas Corp.	Energy	http://www.us.com
0062v00000KCU29AAH	270000	Closed Won	United Oil & Gas Corp.	Energy	http://www.us.com
0062v00000KCU2BAAX	915000	Closed Won	United Oil & Gas Corp.	Energy	http://www.us.com
0062v00000KCU2GAAK	235000	Closed Won	United Oil & Gas Corp.	Energy	http://www.us.com
0062v00000KCU2HAAX	440000	Closed Won	United Oil & Gas Corp.	Energy	http://www.us.com
0062v00000KCU2JAAK	120000	Closed Won	United Oil & Gas Corp.	Energy	http://www.us.com
0062v00000KCU2LAAK	675000	Needs Analysis	United Oil & Gas Corp.	Energy	http://www.us.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 code editor open. The file is named CustomerTrigger.apex. The code implements a trigger on the Customer__c object that inserts a Billing__c record for each new customer if their Active__c field is false. The Billing__c record has a Status__c of 'Paid' and an Amount_Paid__c of 1000000. The trigger also 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 }
```

Below the code editor, there is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, and the status bar at the bottom shows 'Name' and 'Line'.

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

Output :

The screenshot shows the Salesforce IDE interface with the code editor open. The file is named CustomerTriggerTest.apex. It contains a single test method named testName that creates a new customer record with Active__c set to false, updates it to true, and then runs a test. The test fails because the trigger did not run.

```
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 = True;
10        update cust;
11        Test.stopTest();
12    }
13 }
```

Below the code editor, there is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Tests tab is selected. The status bar at the bottom shows 'Status' and 'Test Run'. The results table shows four test runs:

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage
✗	7U2W000008M8950	Sat Feb 25 2023 22:59:53 GMT...	0:00	1	3	Class Overall 33%
✓	7072w00008M895v	Sat Feb 25 2023 22:04:49 GMT...	0:00	0	2	Class Billing 0% 0/6
✗	7072w00008M896Qq	Sat Feb 25 2023 22:57:21 GMT...	0:00	1	3	Class CustomerTrigger 100% 9/9
✓	TestRun @ 10:58:00 pm	Sat Feb 25 2023 22:59:25 GMT...	0:00	0	1	Class demo 0% 0/4
✓	DisqualifyTestLeads			0	1	Class DiscountClass 0% 0/4
✗	a			1	4	

Exercise 14

Write an Apex Trigger, Name = DisqualifyTestLeads.

The screenshot shows the Salesforce code editor with the following trigger code:

```
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 trigger. The test class contains the following code:

```
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 }
```

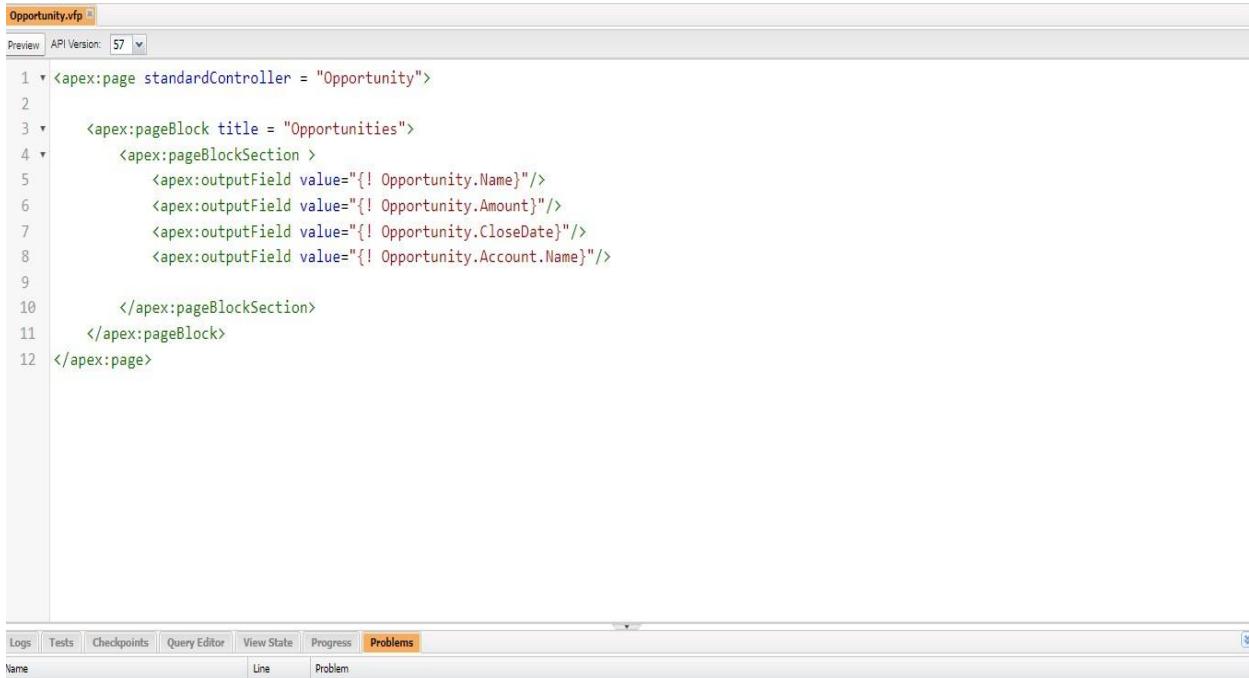
The test results table shows the following runs:

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	7072w00008M8Sv	Sat Feb 25 2023 22:04:49 GMT...	0:00	0	2
✗	7072w00008M8Gq	Sat Feb 25 2023 22:57:21 GMT...	0:00	1	3
✓	7072w00008M8Gq	Sat Feb 25 2023 22:57:21 GMT...	0:00	0	1
✓	DisqualifyTestLeads			0	1
✓	a			0	1
✗	7072w00008M8MK	Sat Feb 25 2023 22:59:25 GMT...	0:00	1	4
✓	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 output fields.



```
Opportunity.vfp
Preview API Version 57
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>
```

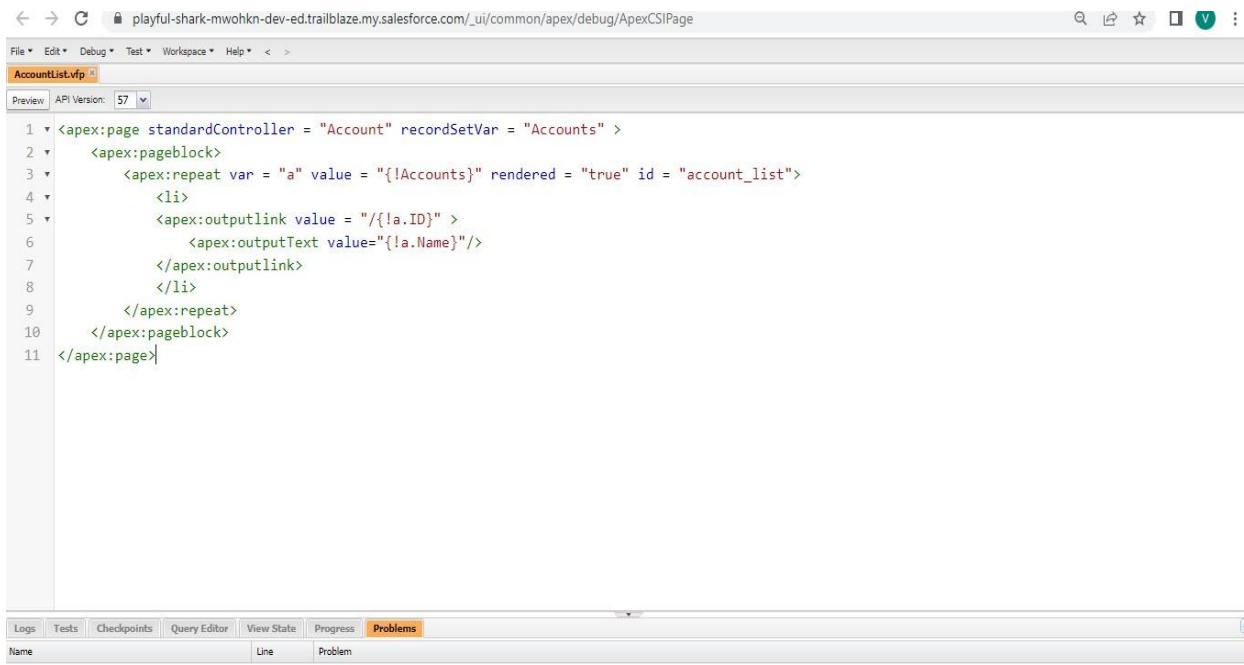
The screenshot shows the Salesforce Visualforce Editor interface. The top bar includes tabs for 'Preview' and 'API Version 57'. The main area contains the Visualforce page code for 'Opportunity.vfp'. The code defines a page with a standard controller of 'Opportunity', containing a single 'pageBlock' with a title of 'Opportunities'. Inside this block is a 'pageBlockSection' that outputs four fields: Name, Amount, CloseDate, and Account Name. Below the code editor is a navigation bar with tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Problems' tab is currently selected. At the bottom of the editor is a search bar with fields for 'Name', 'Line', and 'Problem'.

Output :



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

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: playul-shark-mwohkn-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The main area shows the Visualforce page code named AccountList.vfp. The code uses standardController="Account" and recordSetVar="Accounts". It contains an apex:repeat loop that iterates over accounts, rendering them as list items. Each item contains an outputLink pointing to the account's detail page and an outputText displaying the account name. The code ends with a closing apex:page tag. Below the code editor is a toolbar with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently selected. The bottom status bar shows the word "Name" and buttons for Line and Problem.

```
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>
```

Output :



A screenshot of a web browser window. The address bar shows the URL: playful-shark-mwohkn-dev-ed--c.trailblaze.vf.force.com/apex/Account?core.apexpages.request.devconsole=1. Below the address bar is a horizontal toolbar with icons for back, forward, search, and other browser functions. The main content area displays a list of account names, each preceded by a small blue circular icon with a white checkmark. 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 IDE. The top menu bar includes File, Edit, Debug, Test, Workspace, Help, and a search bar. The current file is "NewCaseList.vfp", which is associated with "NewCaseListController.apxc". The API Version is set to 57. The code editor contains the following Visualforce markup:

```
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
5   </apex:repeat>
6 </apex:page>
```

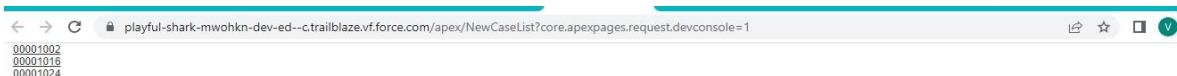
The bottom navigation bar includes Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently selected, showing no errors or warnings.

The screenshot shows the Salesforce 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 area displays the Apex code for NewCaseListController:

```
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 }
```

At the bottom of the interface, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. 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](#)