



Capstone Project Report On “Travel App”

Submitted by:

Name: Anadi Gautam

Batch WIP-SF-13

LMS Id: MGSA_491

Date: 28/02/2023

Github:https://github.com/anadigautam5/Capstone_Project.git

Table Of Contents

S.no	Content	Page No.
1.	Abstract	
2.	Introduction	
3.	Flow of the project	
4.	Software Requirements	
5.	Screen shots	
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

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.

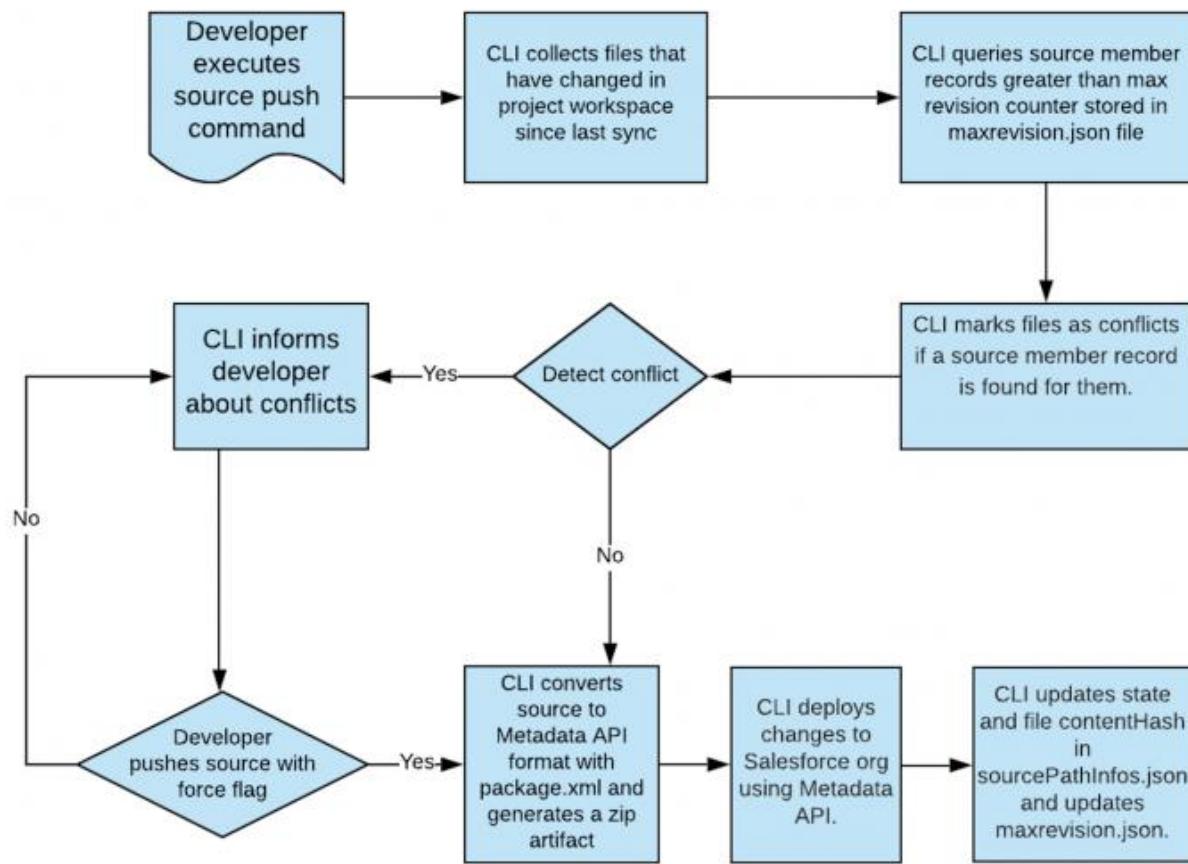
Salesforce is currently providing various software solutions and platforms for developers to create and distribute custom software/applications. Tech giants like Google, Twitter, Amazon, and Facebook are using Salesforce either in the form of SaaS or PaaS.



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>ⓘ</small> <input type="text" value="Travel App"/>	Image <small>ⓘ</small>  Primary Color Hex Value <input type="color" value="#0070D2"/> #0070D2
* Developer Name <small>ⓘ</small> <input type="text" value="Travel_App"/>	
Description <small>ⓘ</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 ?

- *Navigation Style
 Standard navigation
 Console navigation

- *Supported Form Factors
 Desktop and phone
 Desktop
 Phone

Setup and Personalization ?

- Setup Experience
 Setup (full set of Setup options)
 Service Setup

App Personalization Settings

- Disable end user personalization of nav items in this app
 Disable temporary tabs for items outside of this app

Back

Next

New Lightning App

Available Items

Create

▼

das

Selected Items

 Chatter
 Reports
 Dashboards

▲

▼

Back

Next

New Lightning App

Available Profiles

Selected Profiles

sys

System Administrator

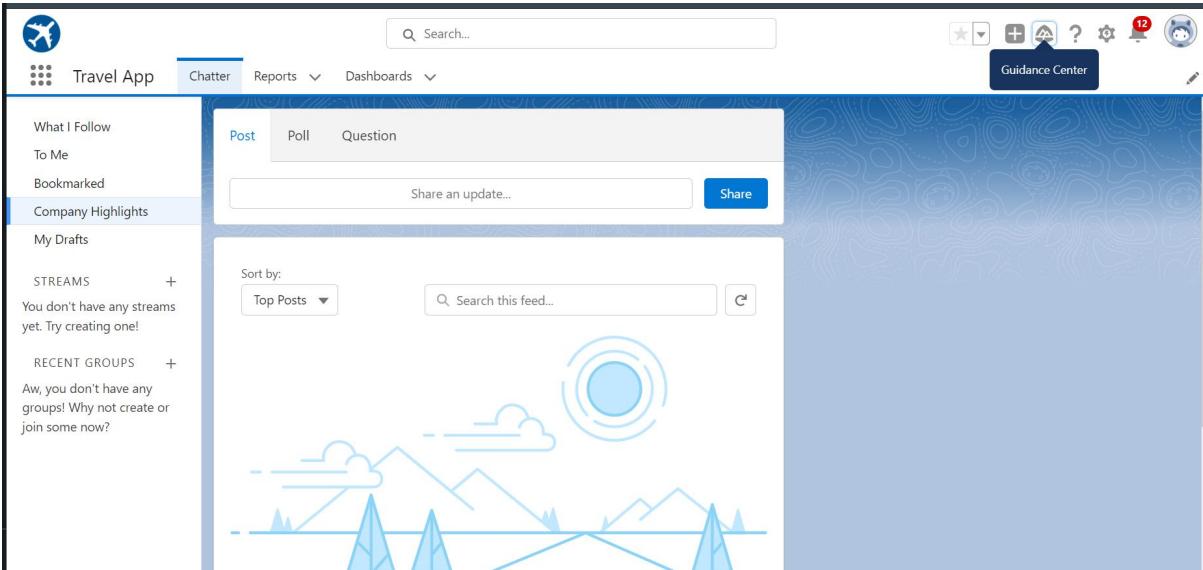
▶

◀

Back

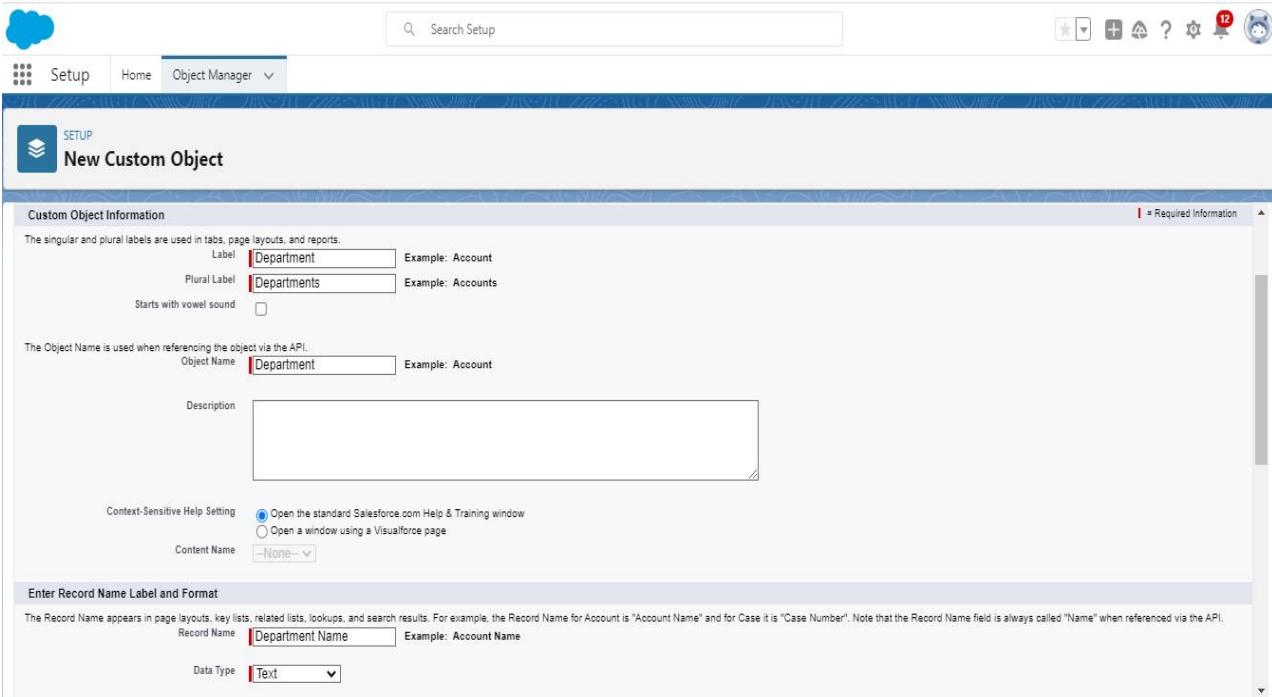
Save & Finish

Final output of step 1 -



Step 2 : - Create a Department custom object.

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



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

(a). Department Code, Text, Length = 10, Required, Select Unique & Case sensitive

Setup > Object Manager > Department

Fields & Relationships

Field Label: Department Code

Length: 10

Field Name: Department_Code

Description: Please enter the maximum length for a text field below.

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

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

Setup > Object Manager > Department

Fields & Relationships

Field Label: Location

Values: Enter values, with each value separated by a new line

Kolkata
Delhi

Field Name: Location

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

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

Search Setup

Setup > OBJECT MANAGER
Department

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

Field Name: Department_Type

Description:

Help Text:

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

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

Save | Cancel | Preview

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

Click button to include or exclude selected values from the dependent picklist:
 Include Values Exclude Values

Showing Columns: 1 - 2 (of 2) < Previous | Next > [View All](#) [Go to](#)

Location:	Department Type:	Showing Columns: 1 - 2 (of 2)
Kolkata	Banking	Delhi
	Finance	Banking
	Education	Finance
	Energy	Education
	IT	Energy
		IT

Click button to include or exclude selected values from the dependent picklist:
 Include Values Exclude Values

Showing Columns: 1 - 2 (of 2) < Previous | Next > [View All](#)

Step 4 : - Create a Travel Approval custom object.

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

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	<input type="text" value="Travel Approval"/>	Example: Account
Plural Label	<input type="text" value="Travel Approvals"/>	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="Travel_Approval"/>	Example: Account
-------------	--	------------------

Description

Context-Sensitive Help Setting

<input checked="" type="radio"/> Open the standard Salesforce.com Help & Training window
<input type="radio"/> Open a window using a Visualforce page

Content Name

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name	<input type="text" value="Travel Approval #"/>	Example: Account Name
Data Type	<input type="text" value="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

Setup Home Object Manager

SETUP > OBJECT MANAGER Travel Approval

Fields & Relationships

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Triggers

Flow Triggers

Validation Rules

Fields & Relationships

11 items, Sorted by Field Level

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		
Travel Approval #	Name	Auto Number	✓	
Trip End Date	Trip_End_Date__c	Date		
Trip Start Date	Trip_Start_Date__c	Date		

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? Getting closer...

Standard objects Custom objects

Departments

Travel Approvals

What do you want to do? Getting closer...

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? Getting closer...

Drag CSV file here to upload

CSV File Choose File Departments.csv

Character Code ISO-8659-1 (General US & Western European, ISO-LATIN-1)

Values Separated By Comma

Cancel Previous Next

5(b) -

Setup Home Object Manager

Almost done

Choose data Edit mapping Start import

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.

Help for this page

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 travel application interface. At the top, there is a navigation bar with links for Travel App, Chatter, Reports, Dashboards, Departments, and Travel Approvals. Below the navigation bar is a search bar and a toolbar with various icons. The main content area displays a list titled "Departments All". The list contains 17 items, each with a checkbox and a department name. The items are numbered from 1 to 17 and include: 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, Technology, and Test 1.

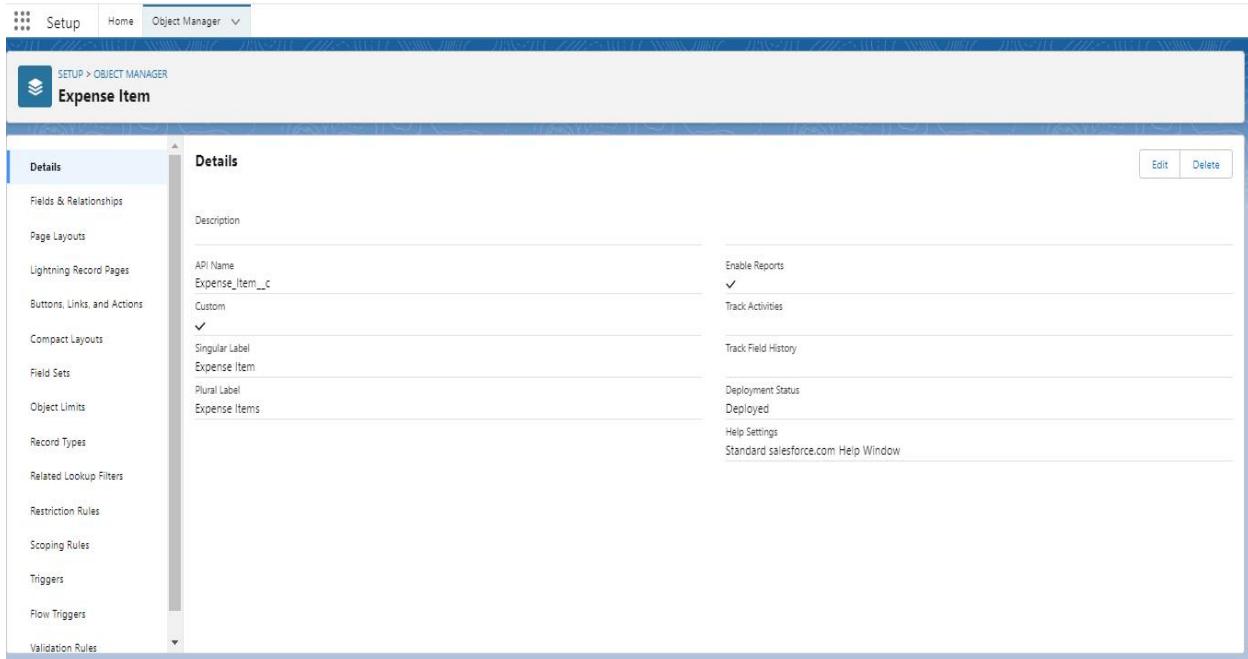
Exercise 2

Step 1: - Create Travel Approval Record

The screenshot shows a travel approval record creation page. The top navigation bar includes links for Travel App, Chatter, Reports, Dashboards, Departments, and Travel Approvals. The main area is divided into two sections: "Details" and "Activity". The "Details" section contains fields for Travel Approval # (TA00001), Purpose Of Trip (Business Trip to USA), Status (New), Trip Start Date (30/04/2022), Trip End Date (07/05/2022), Out Of State (checked), Destination State (US), Department (Test 1), and Created By (Anadi Gautam). The "Activity" section shows a list of activities with a header for "Upcoming & Overdue". It includes a toolbar with various icons and a message stating "No activities to show. Get started by sending an email, scheduling a task, and more." The last modified information is also present at the bottom of the details section.

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.

Fields & Relationships
6 items, Sorted by Field Label

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.

Travel Approval TA00002

Related Details

Notes & Attachments (0)

Expense Items (2)

Expense Item Number
E-00001
E-00002

Activity

No activities to show.
Get started by sending an email, scheduling a task, and more.

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.

User Detail

Name: Eric Executive
Alias: eexec
Email: andri.gautam@wuncom.com
Username: andri.gautameexec@wunpro.com
Nickname: User16774165923520590139

Title:
Company:
Department:
Division:
Address:
Time Zone: (GMT+05:30) India Standard Time (Asia/Kolkata)
Locale: English (India)
Language: English
Delegated Approver: Manager Only if I am an approver

Role: CEO
User Profile: Salesforce
Profile: System Administrator
Active:

Marketing User:
Offline User:
Knowledge User:
Flow User:
Service Cloud User:
Site.com Contributor User:
Site.com Publisher User:
WDC User:
Mobile Push Registrations: View
Data.com User Type:

Accessibility Mode (Classic Only):
Debug Mode:
High-Contrast Palette on Charts:
Load Lightning Pages While Scrolling:
Send Apex Warning Emails:
Salesforce CRM Content User:
Receive Salesforce CRM Content Email Alerts:
Receive Salesforce CRM Content Alerts as Daily Digest:
Make Setup My Default Landing Page:

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

City:
Zip/Postal Code:
State/Province:
Country: IN

Single Sign On Information:

Locale Settings:
Time Zone: (GMT+05:30) India Standard Time (Asia/Kolkata)
Locale: English (India)
Language: English

Approver Settings:
Delegated Approver:
Manager: Eric Executive
Receive Approval Request Emails: Only if I am an approver

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 setup options like Details, Fields & Relationships, Page Layouts, and Search Layouts. The 'Search Layouts' section is currently selected, showing the 'Edit Search Layout' screen for 'Travel Approval Search Results'. The interface includes 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, Destination State, Trip End Date, Trip Start Date). Buttons for 'Add', 'Remove', 'Up', 'Down', and 'Override the search result column customizations for all users' are present. Below these are sections for 'Standard Buttons' (none listed) and 'Custom Buttons' (link to create a new custom list button). At the bottom are 'Save' and 'Cancel' buttons.

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

The screenshot shows the Travel App interface with the 'Travel Approvals' list view. The 'All' filter is selected. A modal dialog titled 'Select Fields to Display' is open, allowing the user to map fields from the 'Available Fields' list to the 'Visible Fields' list. The 'Available Fields' list includes: Out Of State, Owner Alias, Owner First Name, Owner Last Name, Purpose Of Trip, and Record ID. The 'Visible Fields' list includes: Travel Approval #, Department, Created By, Status, Trip Start Date, and Trip End Date. The 'Save' button at the bottom right of the dialog is highlighted.

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.

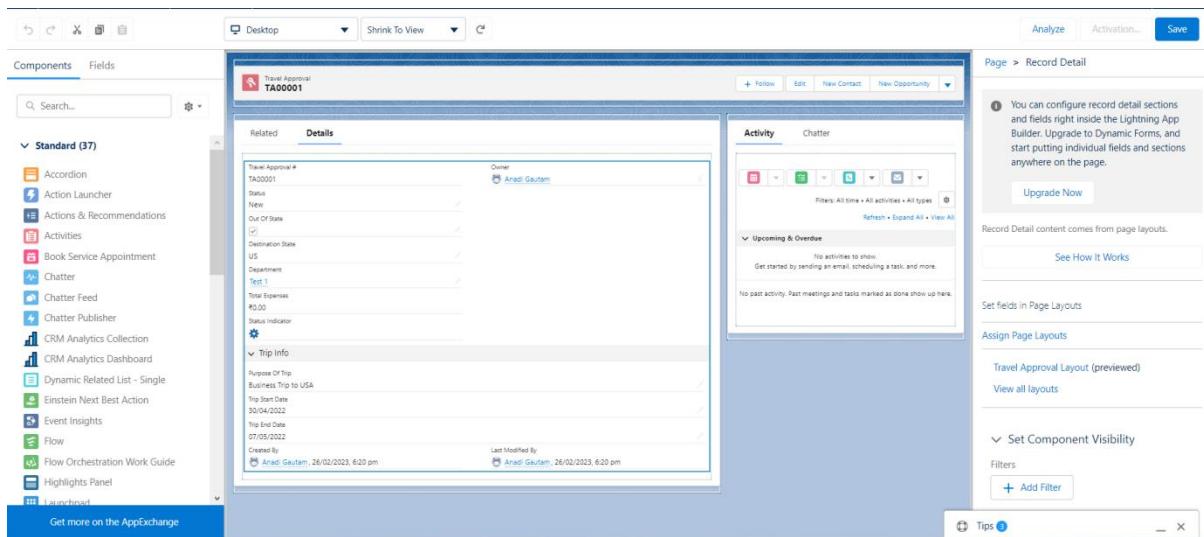
The screenshot shows a Salesforce list view titled "Travel Approvals Open Out of State Travel Requests". The view displays two items, both labeled "TA00001" and "TA00002". On the right side of the screen, there is a sidebar titled "Filters" which contains three filter conditions: "Filter by Owner My travel approvals", "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.

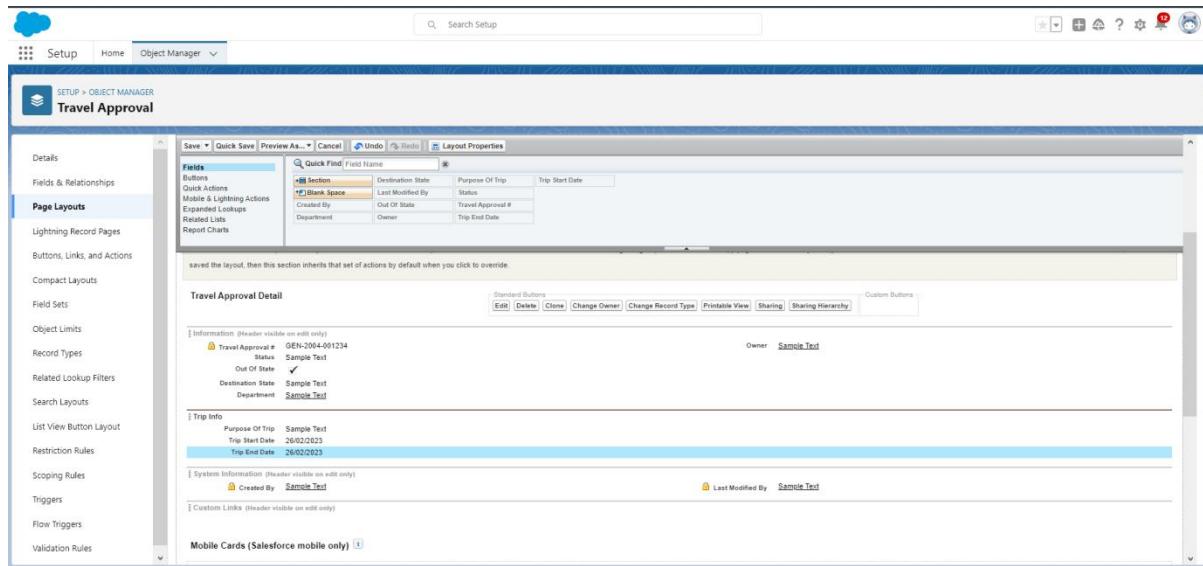
The screenshot shows the "Select Fields to Display" dialog box overlaid on the list view. The dialog has two main sections: "Available Fields" on the left and "Visible Fields" on the right. Under "Available Fields", there are fields: Owner Alias, Owner First Name, Owner Last Name, Purpose Of Trip, Record ID, and Travel Approval #. Under "Visible Fields", there are fields: Department, Created By, Status, Destination State, Trip Start Date, and Trip End Date. At the bottom of the dialog are "Cancel" and "Save" buttons.

Step 11 :- (a) Customize the Travel Approval Page Layout as shown in the screen shot.

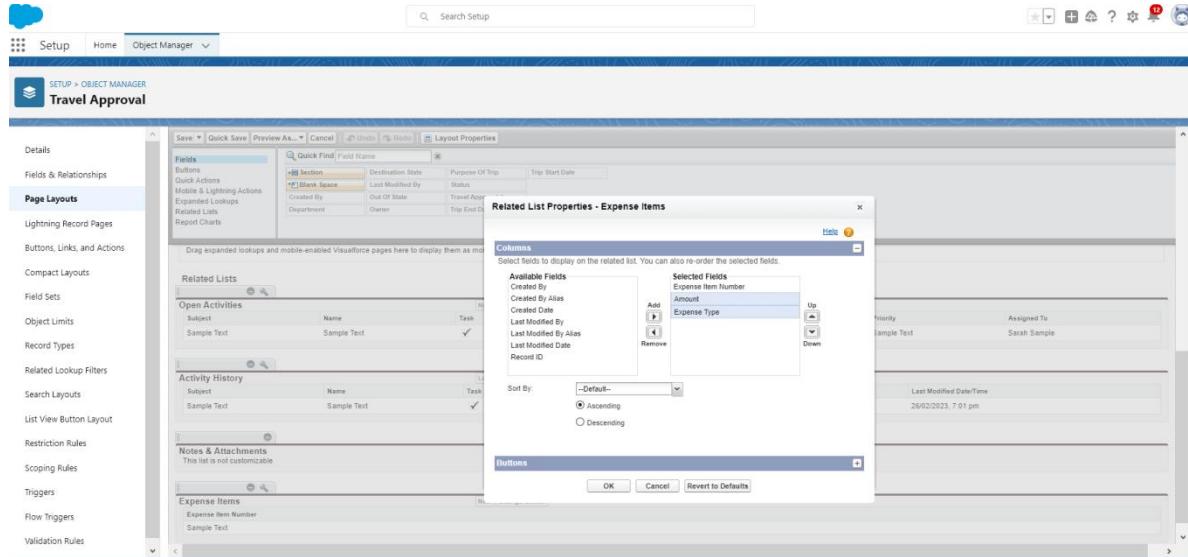
- In the travel app, click travel approval tab and open TA-00001
- Then click Gear button and select “Edit Button” ,It will open as shown as screenshots



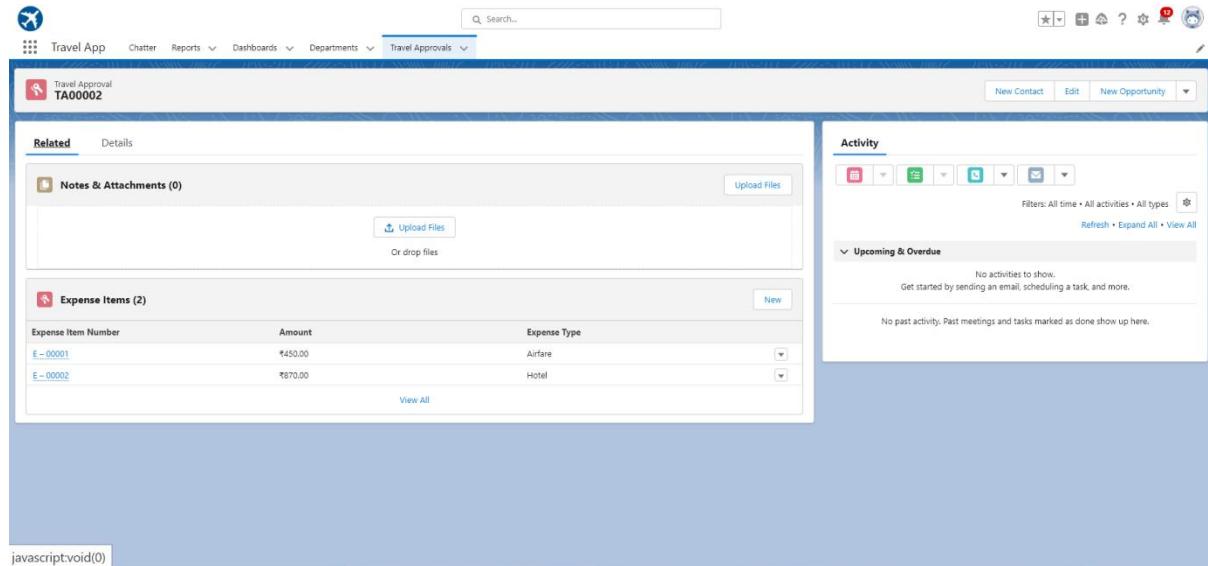
(b) :- Travel Approval Page Layout



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 :- (a)

- 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 page under the Feed Tracking section. On the left, there's a sidebar with a search bar and a list of objects: Feature Settings, Chatter, and Feed Tracking (which is selected). The main area displays the 'Feed Tracking' configuration for the 'Travel Approval' object. It lists various fields under 'Fields in travel approvals' and includes checkboxes for 'Enable Feed Tracking' and 'Destination State'. Other fields listed include Department, Out Of State, Purpose Of Trip, Travel Approval #, Trip Start Date, Owner, Status, and Trip End Date. A note at the bottom says 'You can also display feed activity for related objects.' and 'All Related Objects'.

(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.”

The screenshot shows the Chatter feed within the Travel App. The feed has a 'Post' tab selected. A post from 'TA00002 — Anadi Gautam' is visible, asking 'Which Department should I associate this travel request with?'. Below it, a comment from 'Eric Executive' replies with 'Technology is the correct department.' The interface includes a sidebar with navigation links like 'What I Follow', 'To Me', 'Bookmarked', 'Company Highlights', 'My Drafts', 'STREAMS', and 'RECENT GROUPS'.

Test the App , Let see how its look like

The Travel Approval App should look as shown in the screenshot

This screenshot shows a travel approval record in a Salesforce-like application. The record ID is TA00002. The details tab is selected, showing the following information:

- Travel Approval #: TA00002
- Status: Draft
- Out Of State: CA
- Destination State: CA
- Department: Technology
- Trip Info:
 - Purpose Of Trip: Attend Dreamforce
 - Trip Start Date: 23/02/2023
 - Trip End Date: 16/03/2023
- Created By: Anadi Gautam on 26/02/2023, 6:25 pm
- Last Modified By: Anadi Gautam on 26/02/2023, 6:25 pm

The right side of the screen shows the Chatter feed for this record. A post from Anadi Gautam asks, "Which Department should I associate this travel request with?". Eric Executive replies, "Technology is the correct department." There are like and comment buttons below the post.

This screenshot shows the Chatter feed for the travel approval record TA00002. The feed includes the initial post from Anadi Gautam and the reply from Eric Executive. The feed also shows a search bar and a comment input field.

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.

The screenshot shows the Salesforce Setup interface under the Object Manager for the 'Travel Approval' object. A validation rule named 'Trip_end_date_after_start_date' is being created. The formula is set to '`Trip_End_Date__c > Trip_Start_Date__c`'. A tooltip for the 'ABS' function is displayed, stating it returns the absolute value of a number. The error message field contains 'Trip end date must be greater than or equal to start date'.

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

The screenshot shows the Salesforce Setup interface under the Object Manager for the 'Travel Approval' object. A new custom field is being created, defined as a roll-up summary for the 'Expense Items' object, summing the 'Amount' field. The calculation type is set to 'SUM'.

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

The screenshot shows the Salesforce Setup page. In the left sidebar, under 'Data', 'Static Resources' is selected. The main area displays the 'Static Resource Edit' screen for a resource named 'StatusImages'. The 'File' field contains 'StatusImages.zip' and the 'Cache Control' dropdown is set to 'Public'. The status bar at the bottom right indicates 'Help for this Page'.

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

The screenshot shows the Salesforce Object Manager. A new formula field 'Status Indicator' is being created for the 'Travel Approval' object. The formula is defined as follows:

```

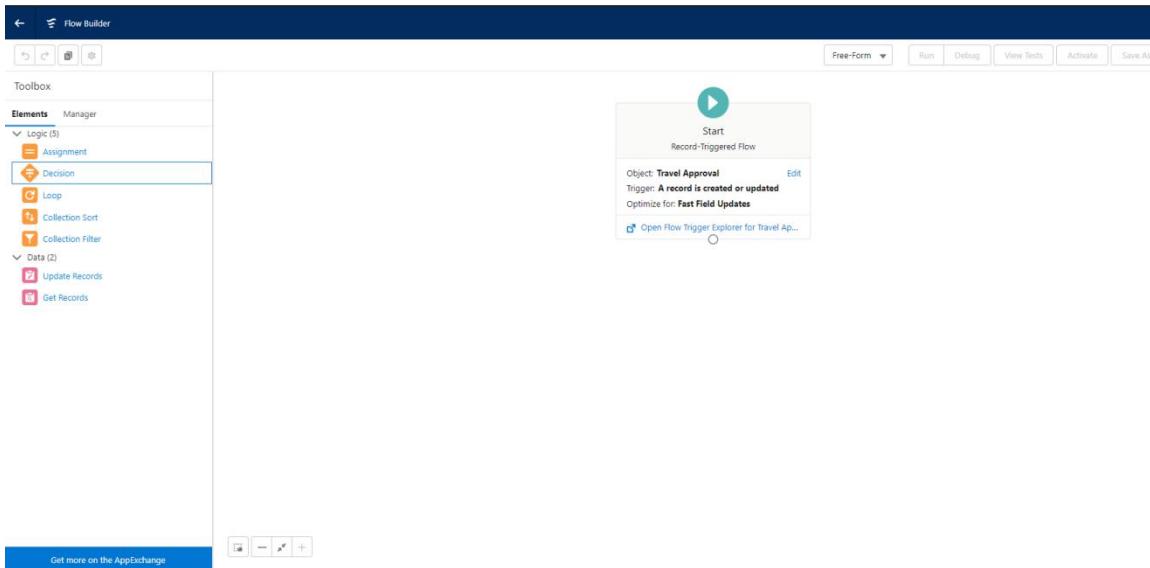
IF( ISPICKVAL( Status_c , 'Approved') , IMAGE("/resource/StatusImages/thumbs-up.png", "Accepted", 20, 20),
IF ( ISPICKVAL( Status_c , 'Rejected'), IMAGE("/resource/StatusImages/thumbs-down.png", "Rejected", 20, 20),
IMAGE("/resource/StatusImages/draft.png", "In-Process", 20, 20)))

```

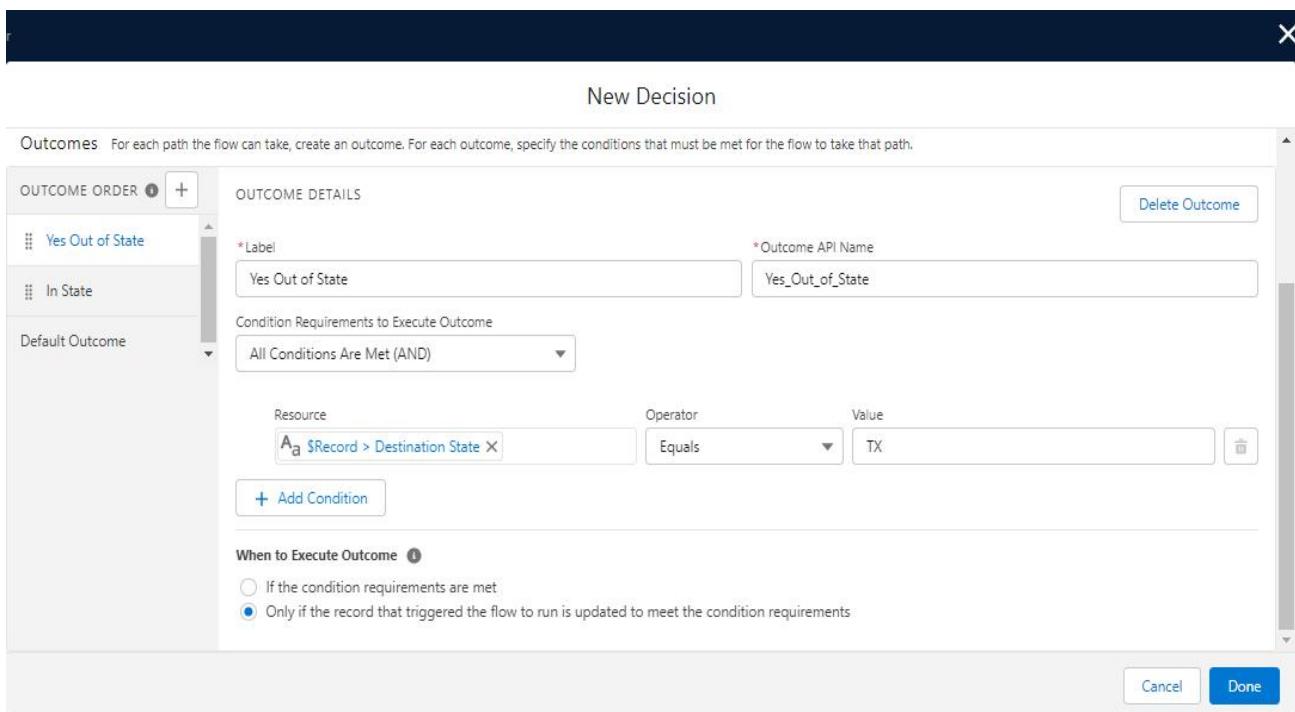
Step 5 :-

(a). Create a Record – Triggered Flow.

- Flow should look like this:

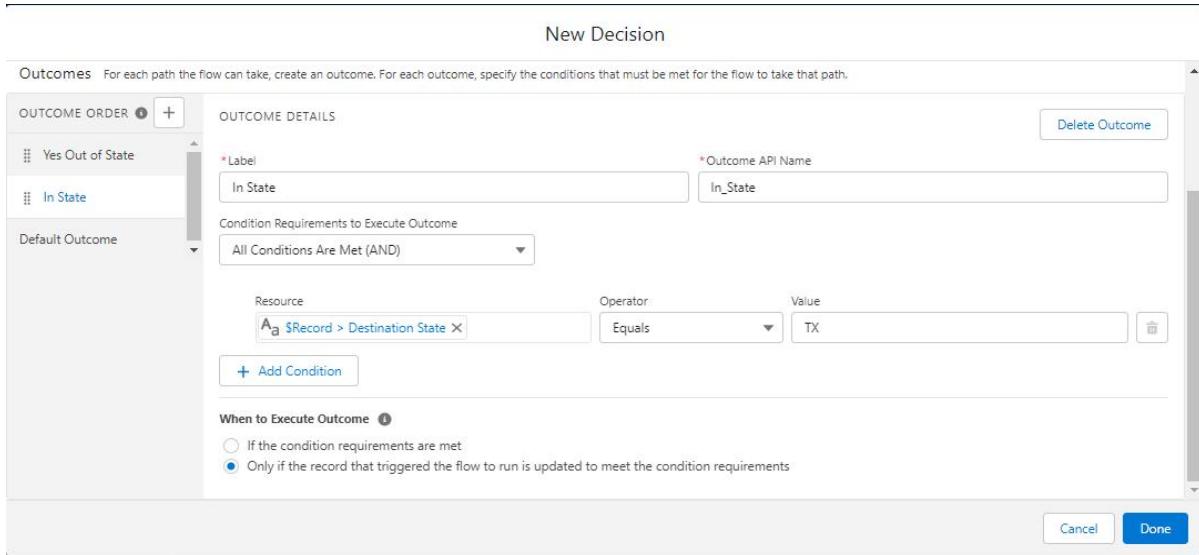


(b). Add a Decision Element to the Flow.

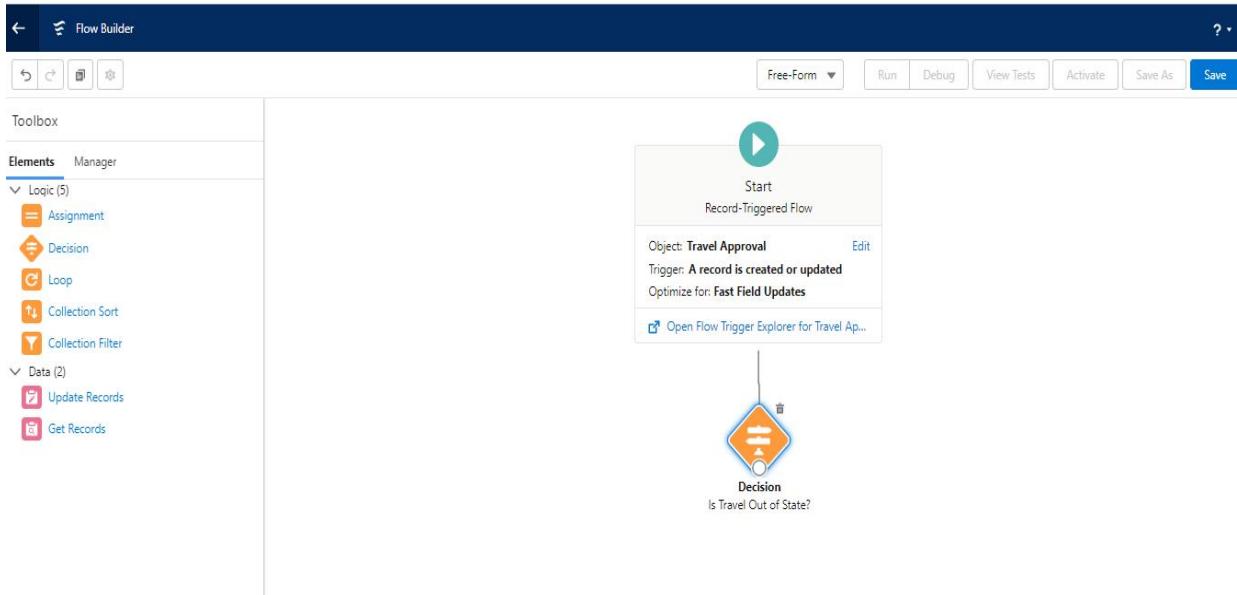


(c). Next to Outcome Order click the + button to add another outcome.

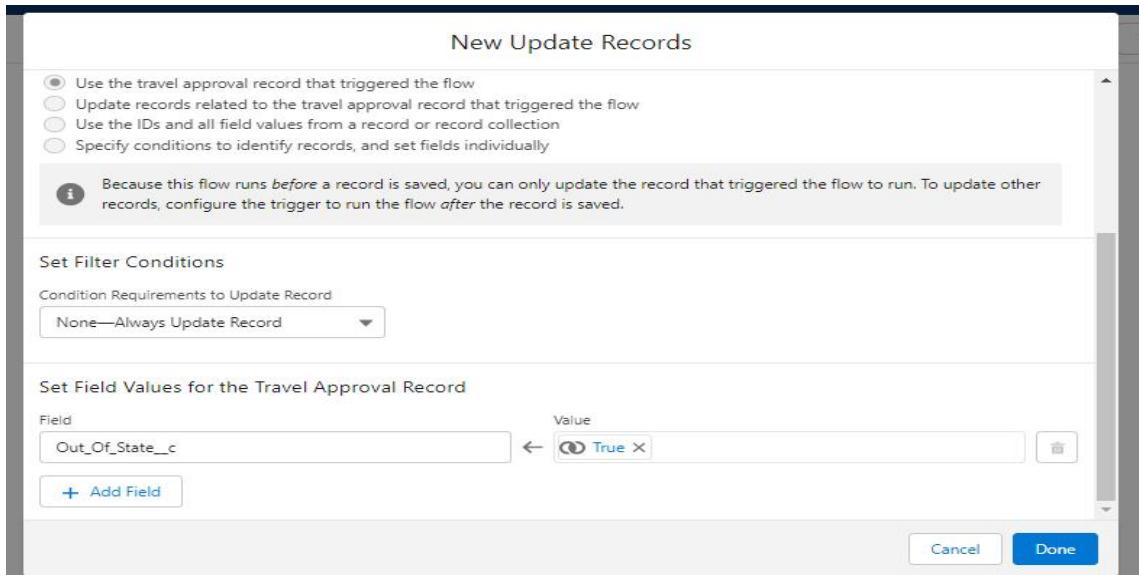
- Flow should look like this:



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

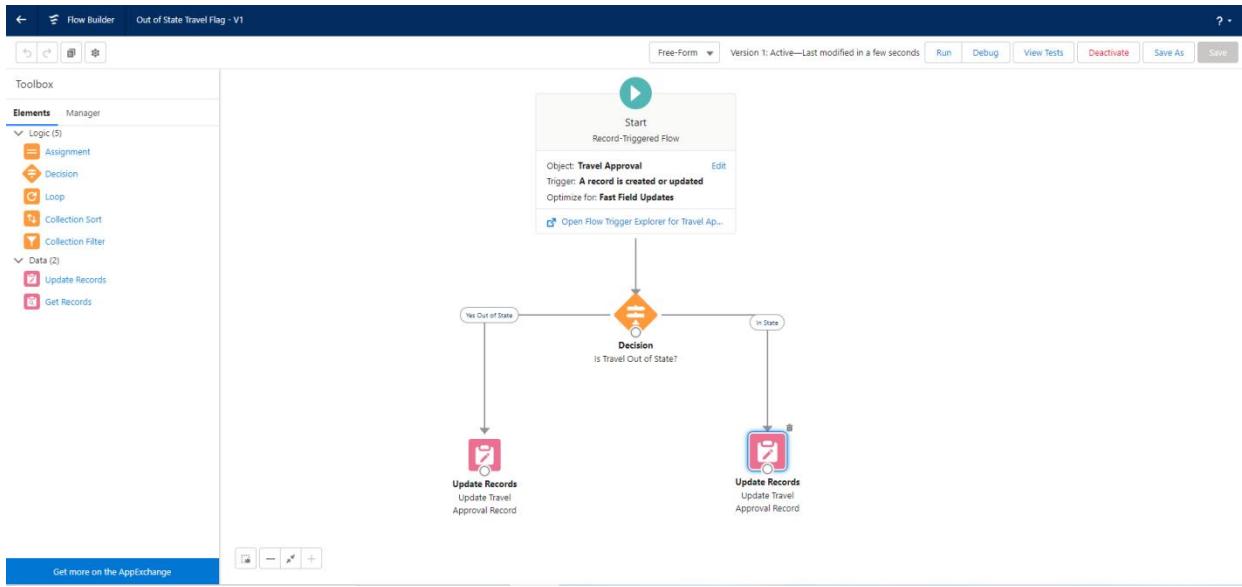


(e). Create an Action for the Flow Using Update Records Elements.



(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 page for 'Approval Processes'. The process is titled 'Travel Approval Request'. Key configuration details include:

- Process Name:** Travel Approval Request
- Unique Name:** Travel_Approval_Request
- Description:** (empty)
- Entry Criteria:** True
- Record Editability:** Administrator ONLY
- Next Automated Approver Determined By:** Manager of Record Submitter
- Approval Assignment Email Template:** (empty)
- Initial Submitters:** Travel Approval Owner
- Created By:** Anadi.Gautam 26/02/2023, 8:14 pm
- Modified By:** Anadi.Gautam 26/02/2023, 8:29 pm
- Active:** checked
- Allow Submitters to Recall Approval Requests:** unchecked

Initial Submission Actions:

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

Approval Steps:

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

(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 a 'Travel Approval' instance in the 'Pending' state. The details are:

- Submitter:** Anadi Gautam
- Date Submitted:** 26-Feb-2023
- Actual Approver:** Eric Executive
- Assigned To:** Eric Executive

Details:

Travel Approval #	Owner
TA00001	Anadi Gautam

Approver Comments:

- Eric Executive:** Approved
26-feb-2023, 8:31:27 pm

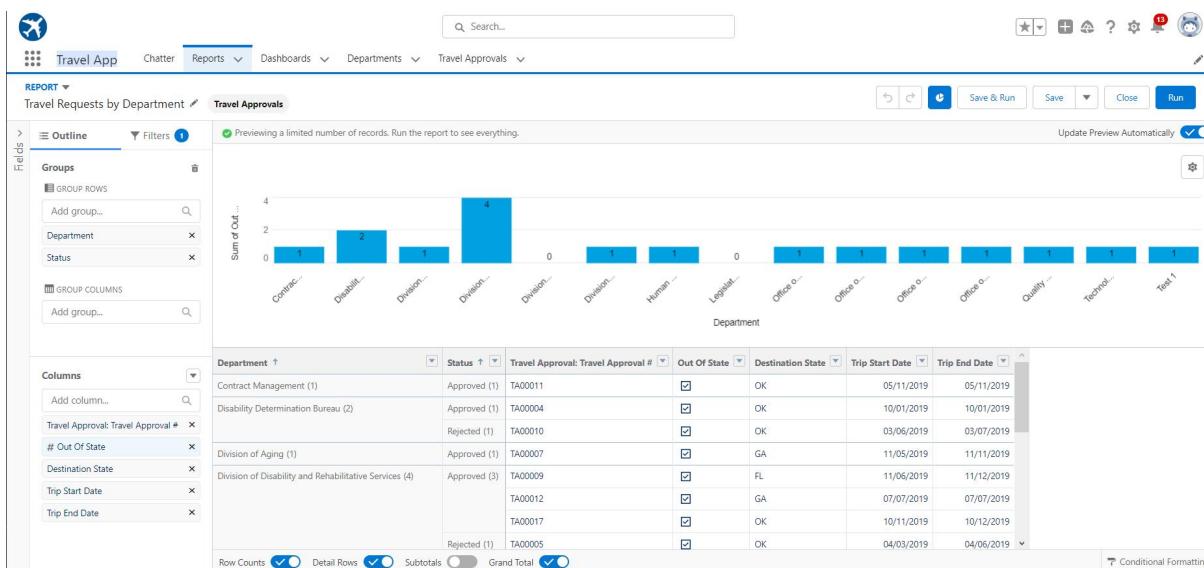
Step 1 :- Use Data Import Wizard to import Travel Approval records.

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

Department	Created By	Status	Trip Start Date	Trip End Date
1 Audit Services	Anadi Gautam	Pending	27/02/2023	30/03/2023
2 Audit Services	Anadi Gautam	Approved	09/03/2019	09/03/2019
3 Audit Services	Anadi Gautam	Approved	07/08/2019	07/09/2019
4 Audit Services	Anadi Gautam	Approved	09/07/2019	08/07/2019
5 Audit Services	Anadi Gautam	Approved	11/04/2019	11/07/2019
6 Audit Services	Anadi Gautam	Approved	07/08/2019	07/08/2019
7 Contract Management	Anadi Gautam	Approved	05/11/2019	05/11/2019
8 Contract Management	Anadi Gautam	Approved	02/09/2019	02/09/2019
9 Contract Management	Anadi Gautam	Approved	07/05/2019	07/08/2019
10 Disability Determination Bureau	Anadi Gautam	Approved	10/01/2019	10/01/2019
11 Disability Determination Bureau	Anadi Gautam	Approved	07/08/2019	07/08/2019
12 Disability Determination Bureau	Anadi Gautam	Approved	07/04/2019	07/10/2019
13 Division of Aging	Anadi Gautam	Approved	11/05/2019	11/11/2019
14 Division of Aging	Anadi Gautam	Approved	01/03/2019	01/07/2019
15 Division of Aging	Anadi Gautam	Approved	09/07/2019	09/07/2019
16 Division of Aging	Anadi Gautam	Approved	09/02/2019	09/10/2019
17 Division of Disability and Rehabilitative Services	Anadi Gautam	Approved	11/06/2019	11/12/2019
18 Division of Disability and Rehabilitative Services	Anadi Gautam	Approved	07/07/2019	07/07/2019
19 Division of Disability and Rehabilitative Services	Anadi Gautam	Approved	10/11/2019	10/12/2019
20 Division of Family Resources	Anadi Gautam	Approved	12/02/2019	12/02/2019
21 Division of Finance	Anadi Gautam	Approved	09/03/2019	09/03/2019

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 travel approvals from January to March 2019, categorized by month and department. The columns include Trip End Date, Out Of State, Travel Approval: Travel Approval #, Department, Destination State, and Trip Start Date.

Trip End Date	Out Of State	Travel Approval: Travel Approval #	Department	Destination State	Trip Start Date		
January 2019 (4)	(3)	TA00041	Division of Finance	TX	10/01/2019		
		TA00074	Office of General Counsel	TX	04/01/2019		
		TA00015	Division of Family Resources	TX	04/01/2019		
	Subtotal	(1)	Disability Determination Bureau	OK	10/01/2019		
	Subtotal						
February 2019 (1)	(1)	TA00064	Division of Family Resources	FL	12/02/2019		
			Subtotal				
	Subtotal						
March 2019 (4)	(2)	TA00019	Legislative Services	TX	06/03/2019		
		TA00065	Office of Medicaid Policy and Planning	TX	10/03/2019		
			Subtotal	(2)	Audit Services	OK	09/03/2019
			Subtotal		Division of Finance	GA	09/03/2019

Step 4 :- Create a Travel Approvals Dashboard.

- Test The Dashboard

Dashboard will look as per the screen shot:

The screenshot shows a Salesforce dashboard titled "Dashboard" with the specific view "Travel Requests Dashboard". The dashboard contains two charts: "Travel Requests by Department" and "Travel Requests by Month". A green banner at the top right indicates "Dashboard saved".

Travel Requests by Department

Department	Record Count
Audit Services	6
Contract Management	5
Disability Determination Bureau	5
Division of Aging	4
Division of Disability and Rehabilitation Services	4
Division of Family Resources	4
Division of Finance	6
Division of Mental Health and Addiction	4
Human Resources	2
Legislative Services	3
Office of Communications and Media	1
Office of Early Childhood and Out-of-School	6
Office of General Counsel	1
Office of Medicaid Policy and Planning	6
Quality and Compliance Office	5
Technology	8
Test 1	1

[View Report \(Travel Requests by Department\)](#)

Travel Requests by Month

Trip End Date	Record Count
January 2019	4
February 2019	1
March 2019	4
April 2019	10
May 2019	6
June 2019	3
July 2019	8
August 2019	6
September 2019	8
October 2019	7
November 2019	7
December 2019	7
May 2020	1
March 2021	2

[View Report \(Travel Requests by Month\)](#)

Module – 3

Exercise 1 : -

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

The screenshot shows the 'App Details & Branding' section of the Lightning App Builder. On the left, the sidebar lists 'App Options', 'Utility Items (Desktop Only)', 'Navigation Items', and 'User Profiles'. The main area has two tabs: 'App Details' and 'App Branding'. In 'App Details', the 'App Name' is set to 'Code Playground' and the 'Developer Name' is set to 'Code_Playground'. In 'App Branding', there is a placeholder image for the app icon, a color swatch for the primary color (hex #0070D2), and a checked checkbox for 'Use the app's image and color instead of the org's custom theme'. An 'Org Theme Options' section is also present. At the bottom, there is an 'App Launcher Preview' section.

The screenshot shows the 'Navigation Items' section of the Lightning App Builder. The sidebar lists 'App Details & Branding', 'App Options', 'Utility Items (Desktop Only)', and 'User Profiles'. The main area shows a list of 'Available Items' on the left and 'Selected Items' on the right. The 'Available Items' list includes 'Accounts', 'Alert Settings', 'All Sites', 'Alternative Payment Methods', 'App Launcher', 'Approval Requests', and a few others. The 'Selected Items' list contains 'Leads', 'Contacts', 'Opportunities', and 'Cases'. Navigation arrows between the two lists allow items to be moved.

The screenshot shows the 'App Settings' section of the Lightning App Builder. On the left, a sidebar lists 'App Details & Branding', 'App Options', 'Utility Items (Desktop Only)', and 'Navigation Items'. The 'User Profiles' option is selected, highlighted with a blue background. The main content area is titled 'User Profiles' and contains the sub-instruction: 'Choose the user profiles that can access this app.' Below this is a 'Available Profiles' list containing ten items: 'Analytics Cloud Integration User', 'Analytics Cloud Security User', 'Authenticated Website', 'Authenticated Website', 'Contract Manager', 'Cross Org Data Proxy User', and 'Custom: Marketing Profile'. To the right is a 'Selected Profiles' list containing one item: 'System Administrator'. Navigation arrows between the two lists allow for selection.

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

The screenshot shows the 'Object Manager' setup page for the 'Customer' object. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main area displays the 'Customer' object details. On the left is a sidebar with links like 'Fields & Relationships', 'Page Layouts', 'Lightning Record Pages', etc. The main 'Details' section shows the API name 'Customer__c' and various labels: 'Custom' (selected), 'Singular Label: Customer', and 'Plural Label: Customers'. On the right, settings include 'Enable Reports' (checked), 'Track Activities', 'Track Field History', 'Deployment Status: Deployed', and 'Help Settings: Standard salesforce.com Help Window'. At the bottom, there are 'Edit' and 'Delete' buttons. The URL in the browser's address bar is 'empathetic-panda-3h1kd0-dev-ed.trailblaze.lightning.force.com/lightning/setup/ObjectManager.../view'.

Step 3:- Create a Custom fields for Customer Object

a. Label = Active, Checkbox, Save.

SETUP > OBJECT MANAGER
Customer

Step 2. Enter the details Step 2 of 4

Field Label: Active

Default Value: Checked Unchecked

Field Name: Active

Description:

Help Text:

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

Previous Next Cancel

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

SETUP > OBJECT MANAGER
Customer

Customer New Custom Field Step 2 of 4

Field Label: Customer Type

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

Premium
Standard

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: Customer_Type

Description:

Help for this Page

Previous Next Cancel

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 entry

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: Customer

Step 2 of 6

Previous Next Cancel

Step 4:- Create a Custom Object for Billing.

Setup > OBJECT MANAGER
Billing

Details

Description:

API Name: Billing__c

Custom: ✓

Singular Label: Billing

Plural Label: Billings

Enable Reports: ✓

Track Activities

Track Field History

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Edit Delete

Step 5:- Create a Custom fields for Billing Object.

- 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

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

- 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

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

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

The screenshot shows the 'Field Information' page for a 'Status' field on the 'Billing' object. The 'Field Label' is set to 'Status' and the 'Field Name' is also 'Status'. The 'Data Type' is 'Picklist'. The 'Available' section contains the values 'Pill', 'HIPAA', 'GDPR', and 'PCI'. The 'Chosen' section contains the value 'Chosen'. Under 'General Options', there is a checkbox for 'Always require a value in this field in order to save a record' which is unchecked. A note below it says: 'Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (2), show percentages as decimal : (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__md__RecordAPIName.FieldName__c'.

Test The App

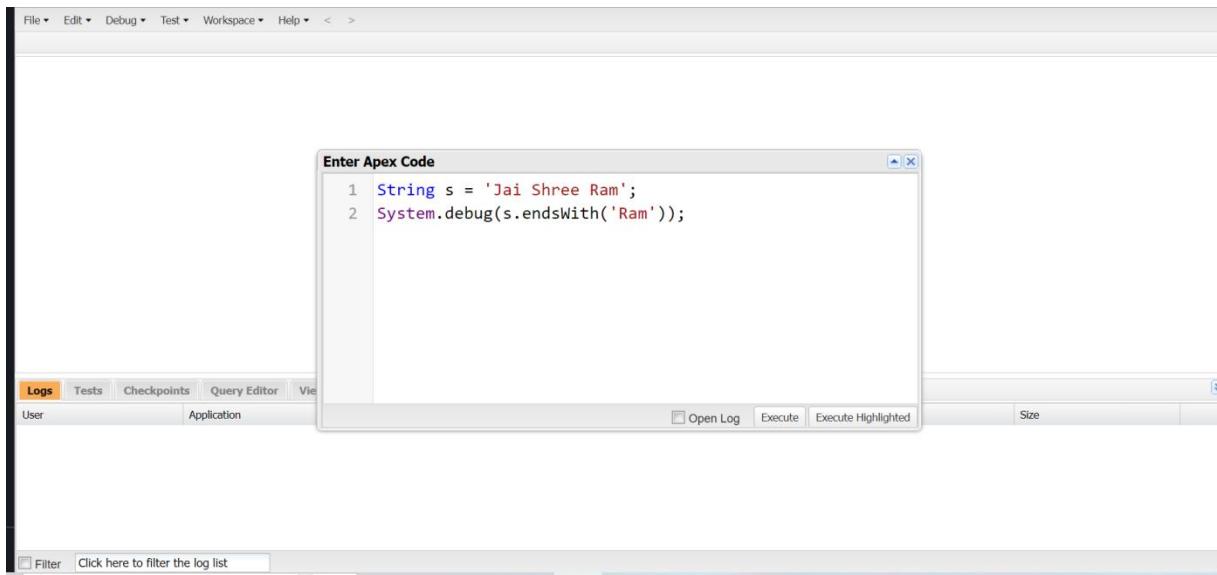
- Code Playground App should look like this:

The screenshot shows a list of accounts in the 'Code Playground' app. The table has columns: Account Name, Account Site, Billing State/Province, Phone, Type, and Account Owner Alias. The data is as follows:

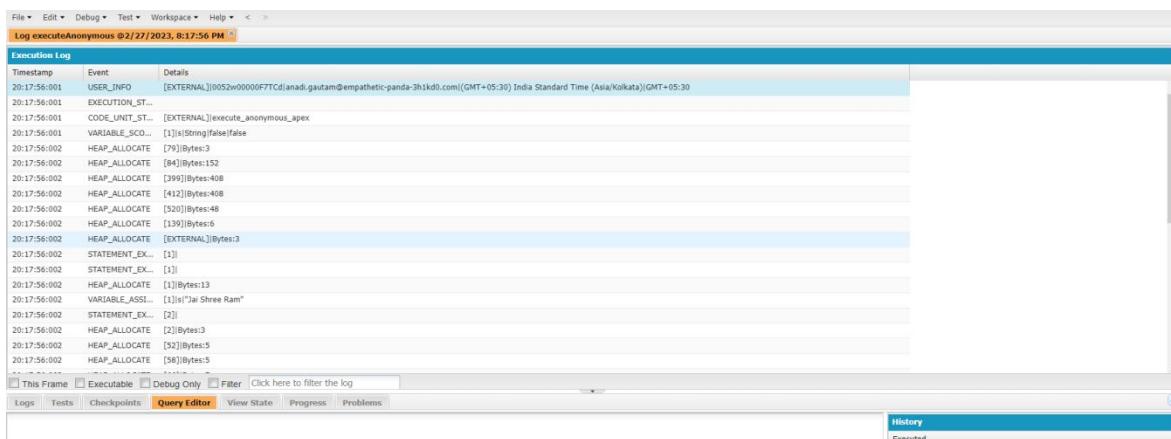
	Account Name	Account Site	Billing State/Province	Phone	Type	Account Owner Alias
1	Burlington Textiles Corp of America	NC	(336) 222-7000		Customer - Direct	AGaut
2	Dickenson plc	KS	(785) 241-6200		Customer - Channel	AGaut
3	Edge Communications	TX	(512) 757-6000		Customer - Direct	AGaut
4	Express Logistics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

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

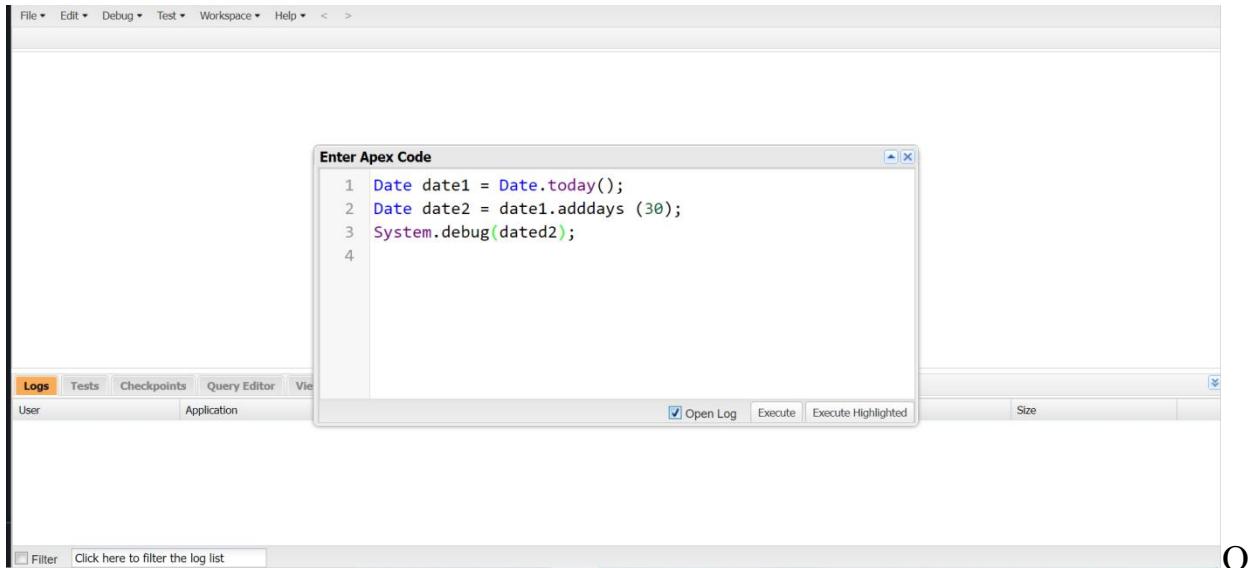
Code 1:- Define a String Variable & use string method ‘endsWith’ to display the output.



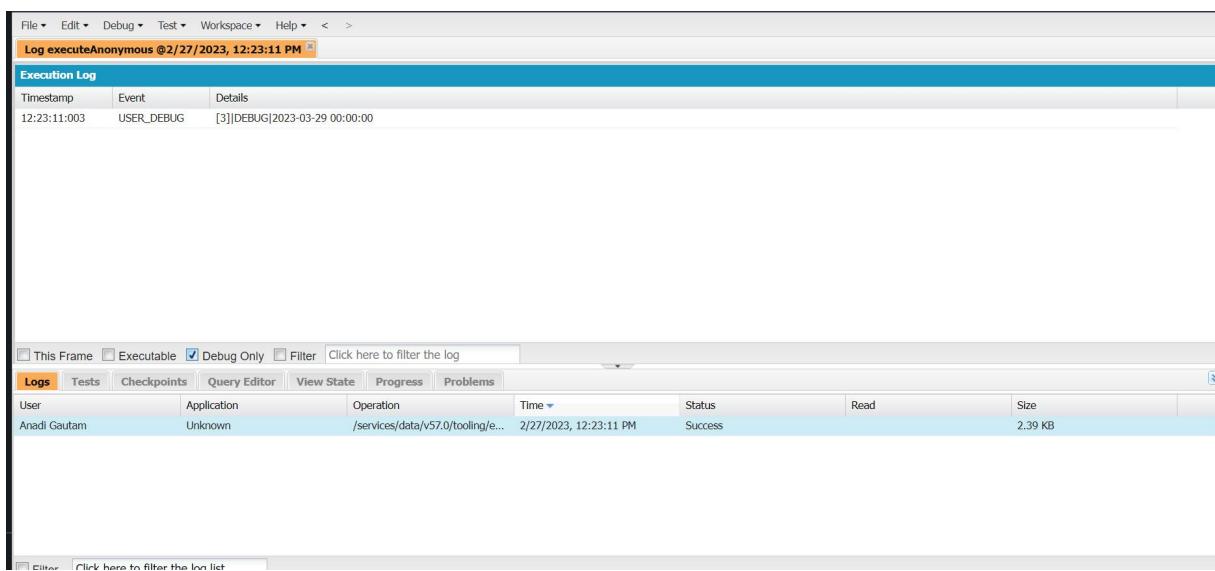
Output :



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



Output :



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

```
String x = '10';
integer y = 20;
integer z = integer.valueof(x) +y;
System.debug(z);
```

Output :

Timestamp	Event	Details
12:26:36:003	USER_DEBUG	[4]DEBUG 30

Execution Log

User	Application	Operation	Time	Status	Read	Size
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:26:36 PM	Success		2.55 KB
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:23:11 PM	Success		2.39 KB

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

The screenshot shows the Salesforce Dev Console interface. At the top, there's a menu bar with File, Edit, Debug, Test, Workspace, Help, and navigation buttons. Below the menu is a toolbar with tabs for Logs, Tests, Checkpoints, Query Editor, and View. The main area is titled "Enter Apex Code" and contains the following code:

```
1 String x = 'Jai Shree Ram';
2 System.debug(x.length());
```

Below the code editor, there's a log table with the following data:

User	Application	Operation	Time	Status	Size
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:23:11 PM	Success	2.55 KB
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:23:11 PM	Success	2.39 KB

At the bottom, there are filter buttons for "Filter" and "Click here to filter the log list".

Output :

The screenshot shows the "Execution Log" interface. At the top, it displays the log entry: "Log executeAnonymous @2/27/2023, 12:28:14 PM". Below this is a table titled "Execution Log" with columns: Timestamp, Event, and Details. The data is as follows:

Timestamp	Event	Details
12:28:14:002	USER_DEBUG	[2] DEBUG 13

At the bottom, there are filter buttons for "This Frame", "Executable", "Debug Only", "Filter", and "Click here to filter the log".

Below the log table is another log table titled "Logs" with the following data:

User	Application	Operation	Time	Status	Read	Size
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:28:14 PM	Success		2.09 KB
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:26:36 PM	Success		2.55 KB
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:23:11 PM	Success		2.39 KB

At the bottom, there are filter buttons for "Filter" and "Click here to filter the log list".

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

The screenshot shows the Salesforce IDE interface. In the top navigation bar, the 'Logs' tab is selected. Below it, the 'Enter Apex Code' window contains the following 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(1);  
14 //clear  
15 l.clear();  
16 System.debug(1);
```

Below the code editor, the 'Logs' tab is highlighted. The log table shows the following entries:

User	Application	Operation	Time	Status	Read	Size
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:31:18 PM	Success		2.09 KB
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:28:14 PM	Success		2.55 KB
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:26:36 PM	Success		2.39 KB

At the bottom of the logs section, there are buttons for 'Open Log', 'Execute', and 'Execute Highlighted'.

Output :

The screenshot shows the Salesforce IDE interface. In the top navigation bar, the 'Logs' tab is selected. Below it, the 'Execution Log' window displays the following log entries:

Timestamp	Event	Details
12:31:18:003	USER_DEBUG	[8] DEBUG l, 2, 3, 4)
12:31:18:003	USER_DEBUG	[10] DEBUG 2
12:31:18:004	USER_DEBUG	[13] DEBUG 1
12:31:18:004	USER_DEBUG	[16] DEBUG 1

Below the execution log, the 'Logs' tab is highlighted. The log table shows the same entries as the previous screenshot:

User	Application	Operation	Time	Status	Read	Size
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:31:18 PM	Success		3.74 KB
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:28:14 PM	Success		2.09 KB
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:26:36 PM	Success		2.55 KB
Anadi Gautam	Unknown	/services/data/v57.0/tooling/e...	2/27/2023, 12:23:11 PM	Success		2.39 KB

At the bottom of the logs section, there are buttons for 'This Frame', 'Executable', 'Debug Only', 'Filter', and 'Click here to filter the log'.

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

The screenshot shows the Salesforce Execute Anonymous window. The code entered is:

```
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 for(integer x=0; x<10; x=x+1){
9     System.debug('value of x is :'+x);
10}
11
```

The Logs tab is selected, showing the following log entries:

User	Applid
Anadi Gautam	Unknow

At the bottom right, there are buttons for Open Log, Execute, and Execute Highlighted. A file size summary is also present:

Size
3.74 KB
2.09 KB
2.55 KB
2.39 KB

Output :

The screenshot shows the Salesforce Execution Log window. The log title is "Log executeAnonymous @2/27/2023, 12:34:31 PM". The log table has columns: Timestamp, Event, and Details. The log entries show the values of x from 0 to 9 being debugged:

Timestamp	Event	Details
12:34:31:002	USER_DEBUG	[5] DEBUG Value of x is :0
12:34:31:002	USER_DEBUG	[5] DEBUG Value of x is :1
12:34:31:002	USER_DEBUG	[5] DEBUG Value of x is :2
12:34:31:003	USER_DEBUG	[5] DEBUG Value of x is :3
12:34:31:003	USER_DEBUG	[5] DEBUG Value of x is :4
12:34:31:003	USER_DEBUG	[5] DEBUG Value of x is :5
12:34:31:003	USER_DEBUG	[5] DEBUG Value of x is :6
12:34:31:003	USER_DEBUG	[5] DEBUG Value of x is :7
12:34:31:003	USER_DEBUG	[9] DEBUG Value of x is :0
12:34:31:003	USER_DEBUG	[9] DEBUG Value of x is :1
12:34:31:003	USER_DEBUG	[9] DEBUG Value of x is :2
12:34:31:003	USER_DEBUG	[9] DEBUG Value of x is :3
12:34:31:003	USER_DEBUG	[9] DEBUG Value of x is :4
12:34:31:003	USER_DEBUG	[9] DEBUG Value of x is :5
12:34:31:003	USER_DEBUG	[9] DEBUG Value of x is :6
12:34:31:003	USER_DEBUG	[9] DEBUG Value of x is :7
12:34:31:003	USER_DEBUG	[9] DEBUG Value of x is :8
12:34:31:004	USER_DEBUG	[9] DEBUG Value of x is :9

At the bottom left, there are checkboxes for This Frame, Executable, and Debug Only. A filter input field is also present.

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 the 'Enter Apex Code' window. Inside, the following Apex code is written:

```
1 Integer myunluckyNumber = 7;
2 Integer myluckyNumber = 15;
3 System.debug(myluckyNumber != myunluckyNumber + 8);
4 |
```

At the bottom of the code editor, there are three buttons: Open Log (with a checked checkbox), Execute, and Execute Highlighted.

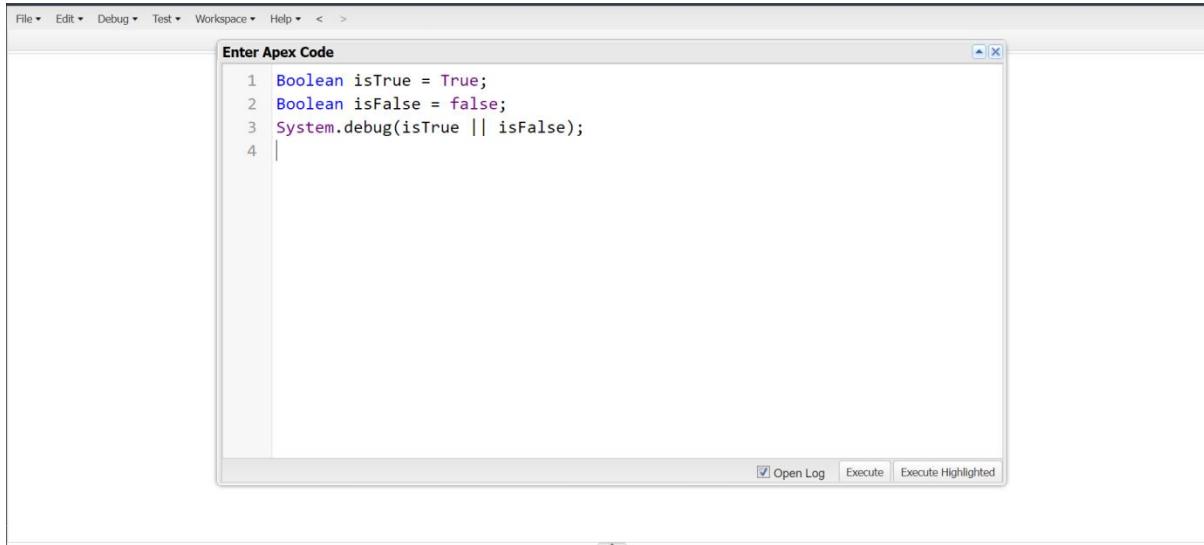
Output :

The screenshot shows the Salesforce IDE interface with the 'Logs, Tests, and Problems' tab selected at the bottom. Above it is the 'Execution Log' window. The log header shows 'Log executeAnonymous @2/27/2023, 12:36:28 PM'. The log table has columns for Timestamp, Event, and Details. One entry is visible:

Timestamp	Event	Details
12:36:28:003	USER_DEBUG	[3]DEBUG false

At the bottom of the log window, there are several filter options: This Frame, Executable, Debug Only (which is checked), Filter, and Click here to filter the log.

Exercise 4 : - Answer the following in True Or False:

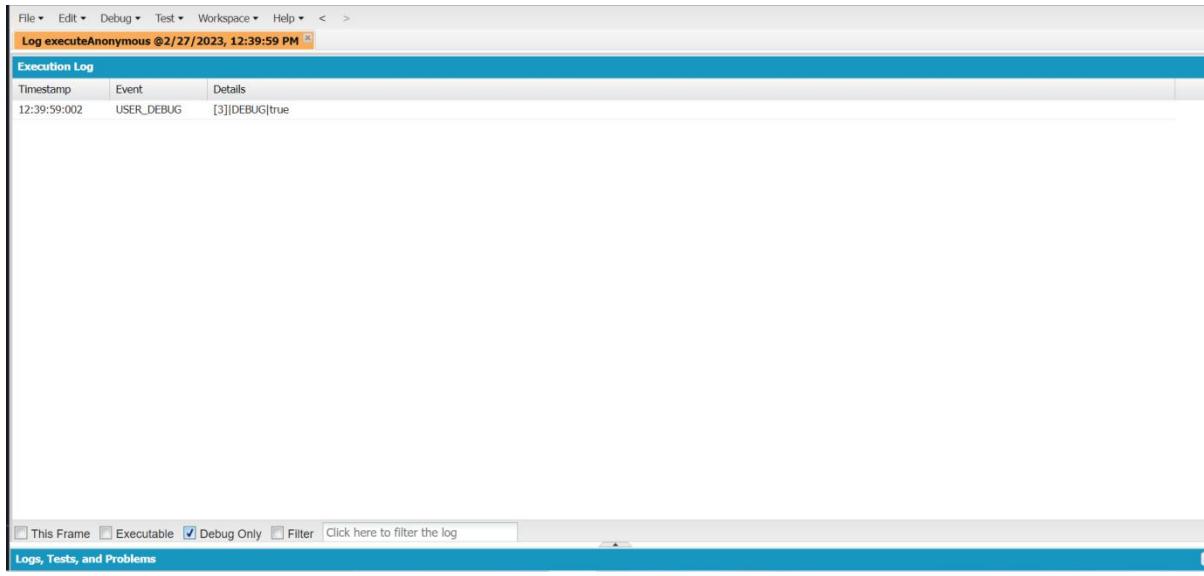


The screenshot shows the Salesforce IDE interface with the 'Enter Apex Code' window open. The code editor contains the following Apex code:

```
1 Boolean.isTrue = True;
2 Boolean.isFalse = false;
3 System.debug(isTrue || isFalse);
4
```

Below the code editor is a toolbar with three buttons: 'Open Log' (unchecked), 'Execute' (disabled), and 'Execute Highlighted'.

Output :



The screenshot shows the Salesforce IDE interface with the 'Execution Log' window open. The log header indicates the log was executed at 2/27/2023, 12:39:59 PM. The log table has columns: Timestamp, Event, and Details. One entry is visible:

Timestamp	Event	Details
12:39:59:002	USER_DEBUG	[3]DEBUG true

At the bottom of the log window, there are filter options: 'This Frame', 'Executable', 'Debug Only' (which is checked), 'Filter', and a link 'Click here to filter the log'. Below the log window is a navigation bar with tabs: 'Logs, Tests, and Problems'.

Exercise 5 : - Answer the following in True Or False:

The screenshot shows the Salesforce IDE interface. At the top, there is a menu bar with options: File, Edit, Debug, Test, Workspace, Help, < >. Below the menu is a toolbar with icons for New, Open, Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and Delete. The main area is titled "Enter Apex Code" and contains the following code:

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

At the bottom of the code editor are three buttons: "Open Log" (with a checkmark), "Execute", and "Execute Highlighted".

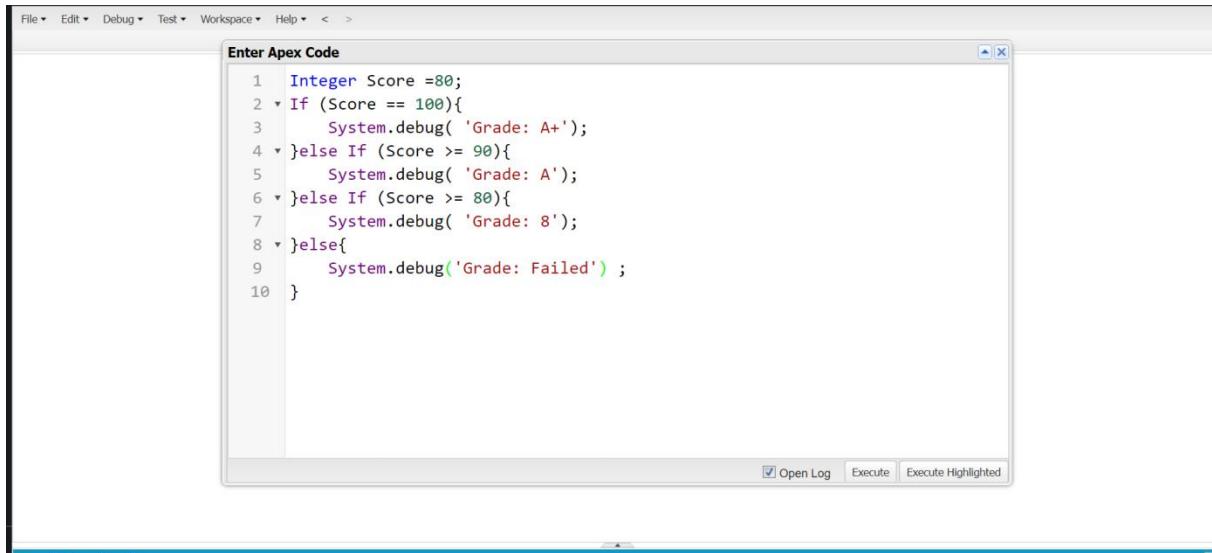
Output :

The screenshot shows the "Execution Log" window in the Salesforce IDE. The title bar indicates "Log executeAnonymous @2/27/2023, 12:42:20 PM". The log table has columns: Timestamp, Event, and Details. One entry is visible:

Timestamp	Event	Details
12:42:20:003	USER_DEBUG	[3] DEBUG true

At the bottom of the log window, there are several filter options: "This Frame", "Executable", "Debug Only" (which is checked), "Filter", and "Click here to filter the log".

Exercise 6 : - Write a program and execute to demo the use of “If..else if...else”.

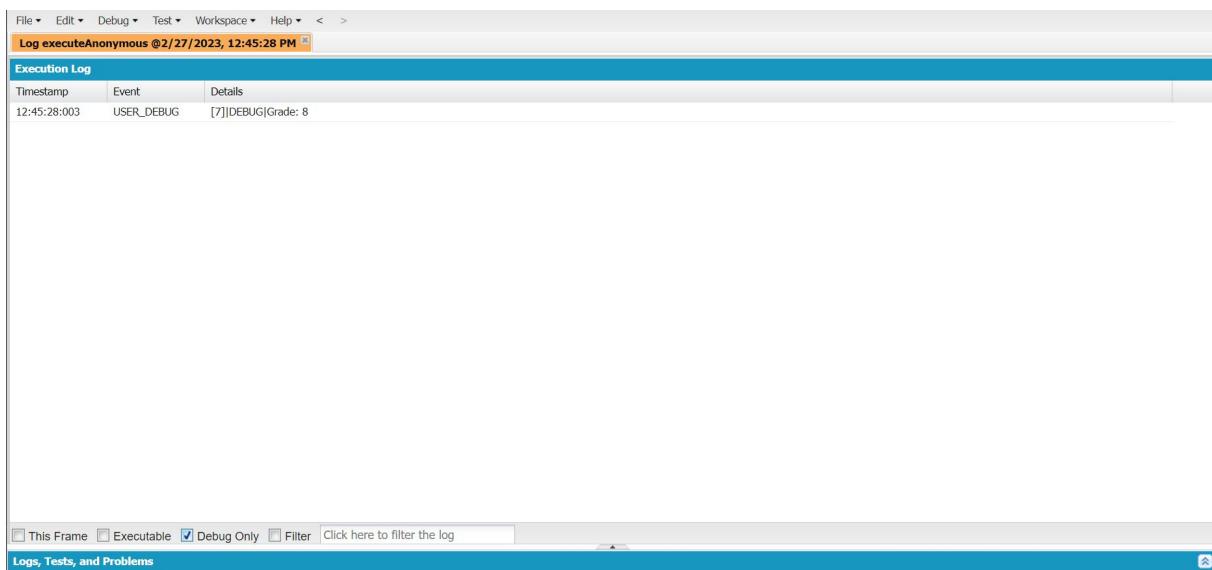


The screenshot shows the Salesforce IDE's "Enter Apex Code" window. The code editor contains the following Apex code:

```
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: 8' );
8 }else{
9     System.debug('Grade: Failed') ;
10 }
```

Below the code editor are three buttons: "Open Log" (unchecked), "Execute" (disabled), and "Execute Highlighted".

Output :



The screenshot shows the Salesforce IDE's "Execution Log" window. The log header indicates the log was created at "Log executeAnonymous @2/27/2023, 12:45:28 PM". The log table has columns for "Timestamp", "Event", and "Details". One entry is present:

Timestamp	Event	Details
12:45:28:003	USER_DEBUG	[7]DEBUG Grade: 8

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. The status bar at the bottom of the IDE window displays "Logs, Tests, and Problems".

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

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

```
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < ▾ >
Billing.apxc [1]
Code Coverage: None ▾ API Version: 57 ▾ Go To
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 }
```

Enter Apex Code
1 billing.viewbills();

Open Log | Execute | Execute Highlighted

Output :

The screenshot shows the Salesforce IDE interface. At the top, there's a menu bar with File, Edit, Debug, Test, Workspace, Help, and a search bar. Below the menu is the title "Billing.apxc" and "Log executeAnonymous @2/25/2023, 1:10:44 PM".

Execution Log:

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)

Enter Apex Code:

```
1 billing.viewbills();
```

Below the code editor, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is 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 search bar. Below the menu is the title "DiscountClass.apxc" and "API Version: 57".

DiscountClass.apxc:

```
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 }
```

Below the code editor, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is highlighted.

Output :

The screenshot shows the Salesforce IDE interface. At the top, there's a menu bar with File, Edit, Debug, Test, Workspace, Help, and a search bar. Below the menu is the title "DiscountClass.apxc" and "Log executeAnonymous @2/25/2023, 1:21:51 PM".

Execution Log:

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

Enter Apex Code:

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

Below the code editor, there are tabs for This Frame, Executable, Debug Only, Filter, Click here to filter the log, Open Log, Execute, and Execute Highlighted.

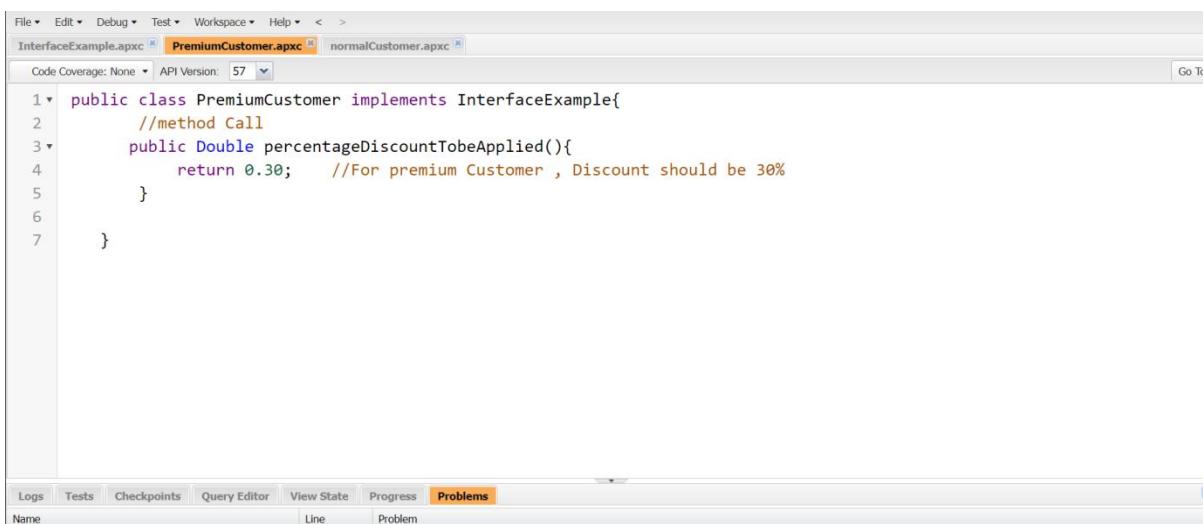
Exercise 9 : - Write a Class to demo the use of Interface in Apex



```
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
InterfaceExample.apxc [ ] PremiumCustomer.apxc [ ] normalCustomer.apxc [ ]
Code Coverage: None ▾ API Version: 57 ▾ Go To
1 * public interface InterfaceExample {
2     Double percentageDiscountTobeApplied(); //method signature only
3 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem



```
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
InterfaceExample.apxc [ ] PremiumCustomer.apxc [ ] normalCustomer.apxc [ ]
Code Coverage: None ▾ API Version: 57 ▾ Go To
1 * public class PremiumCustomer implements InterfaceExample{
2     //method Call
3     public Double percentageDiscountTobeApplied(){
4         return 0.30; //For premium Customer , Discount should be 30%
5     }
6
7 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

```
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
InterfaceExample.apxc [ ] PremiumCustomer.apxc [ ] normalCustomer.apxc [ ]
Code Coverage: None ▾ API Version: 57 ▾ Go To
1 public class normalCustomer implements InterfaceExample{
2     //method Call
3     public Double percentageDiscountTobeApplied(){
4         return 0.10;    //For Normal Customer , Discount should be 10%
5     }
6
7 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Output :

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
InterfaceExample.apxc [] PremiumCustomer.apxc [] normalCustomer.apxc [] Log executeAnonymous @2/27/2023, 1:51:26 PM []

Execution Log

Timestamp	Event	Details
13:51:26:037	USER_DEBUG	[3]DEBUG Discount in Percentage From premium30.0%
13:51:26:037	USER_DEBUG	[6]DEBUG Discount in Percentage From Normal10.0%

Enter 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)+'%' );
```

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

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

The screenshot shows the Salesforce IDE interface with the code editor open. The file is named DML.apxc. The code implements a test method that creates a Customer__c record with name 'Wipro' and Customer_Type__c 'Premium'. It then inserts this record into the Billing__c object with Status__c 'paid' and Amount_Paid__c 5000000. The code includes error handling logic to debug errors if any occur during the insert operation.

```
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 screenshot shows the Salesforce IDE interface with the execution log and an enter apex code dialog. The log displays the execution of the test method from the previous screenshot. The log entries show the creation of a Customer__c record and its insertion into the Billing__c object. The enter apex code dialog contains the same code as the DML.apxc file, demonstrating the execution of the Apex code.

Timestamp	Event	Details
14:04:49:002	USER_INFO	[EXTERNAL]0052w00000F7TCd anadi.gautam@empathetic-panda-3h1kd0.com (GMT+05:30) India Standard Time (Asia/Kolkata) GMT+05:30
14:04:49:002	EXECUTION_ST...	
14:04:49:002	CODE_UNIT_ST...	[EXTERNAL]\execute_anonymous_apex
14:04:49:002	VARIABLE_SCO...	[1];cust:Customer__c true false
14:04:49:002	HEAP_ALLOCATE	[79]Bytes:3
14:04:49:002	HEAP_ALLOCATE	[84]Bytes:152
14:04:49:002	HEAP_ALLOCATE	[399]Bytes:408
14:04:49:002	HEAP_ALLOCATE	[412]Bytes:408
14:04:49:002	HEAP_ALLOCATE	[520]Bytes:48
14:04:49:002	HEAP_ALLOCATE	[139]Bytes:6
14:04:49:002	HEAP_ALLOCATE	[EXTERNAL]Bytes:7
14:04:49:002	STATEMENT_EX...	[1]
14:04:49:002	STATEMENT_EX...	[1]
14:04:49:002	HEAP_ALLOCATE	[1]Bytes:4
14:04:49:003	VARIABLE_ASSI...	[1];cust:{} 0x449a6ab1
14:04:49:003	STATEMENT_EX...	[2]
14:04:49:003	HEAP_ALLOCATE	[2]Bytes:5
14:04:49:003	VARIABLE_ASSI...	[2];this.Name "Wipro" 0x449a6ab1
14:04:49:003	STATEMENT_EX...	[3]
14:04:49:003	HEAP_ALLOCATE	[3]Bytes:7
14:04:49:003	VARIABLE_ASSI...	[3];this.Customer_Type__c "Premium" 0x449a6ab1
14:04:49:003	STATEMENT_EX...	[5]
14:04:49:003	HEAP_ALLOCATE	[52]Bytes:5
14:04:49:003	HEAP_ALLOCATE	[58]Bytes:5
14:04:49:003	HEAP_ALLOCATE	[66]Bytes:7
14:04:49:003	HEAP_ALLOCATE	[5]Bytes:8
14:04:49:003	DML_BEGIN	[5];Op:Insert Type:Customer__c Rows:1

Output :

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

DML.apxc Log executeAnonymous @2/27/2023, 2:20:40 PM [5]

Execution Log

Timestamp	Event	Details
14:20:40:046	USER_DEBUG	[5] DEBUG Successfully inserted Billing@012w000018Fkj4AAC

Enter Apex Code

```

1 List<Billing__c> listinsert = new List<Billing__c>{new Billing__c(Status__c = 'paid',Amount_Paid__c = 500000
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}

```

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

User	Application	Operation	Time	Status	Read	Size
Anadi Gautam	Unknown	/services/data/v57.0/tooling/...	2/27/2023, 2:20:40 PM	Success		4.57 KB
Anadi Gautam	Unknown	/services/data/v57.0/tooling/...	2/27/2023, 2:18:21 PM	Success		2.73 KB
Anadi Gautam	Unknown	/services/data/v57.0/tooling/...	2/27/2023, 2:04:49 PM	Success		2.73 KB

Click here to filter the Inv list

Code Playground Accounts Leads Contacts Opportunities Cases Billings Customers

Customers Recently Viewed

4 items • Updated a minute ago

Customer Name
1 Wipro
2 Wipro
3 Wipro
4 Wipro

Code Playground Accounts Leads Contacts Opportunities Cases Billings Customers

Billing B - 0005

Related Details

Bill Number B - 0005	Owner Anadi Gautam
Amount Paid ₹50,00,000	
Customer Type	
Status Paid	
Created By Anadi Gautam, 27/02/2023, 2:20 pm	Last Modified By Anadi Gautam, 27/02/2023, 2:20 pm

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

Output :

The screenshot shows the Salesforce Developer Console interface. At the top, there is a menu bar with options like File, Edit, Debug, Test, Workspace, Help, and a timestamp "Opportunity@10:21 PM". Below the menu is a search bar with the query: "Select ID, Amount, StageName, Account.Name, Account.Industry, Account.Website From Opportunity Where Account.Industry = 'Energy' AND Account.AnnualRevenue > 5000". The main area displays the "Query Results - Total Rows: 10" table with the following data:

ID	Amount	StageName	Account.Name	Account.Industry	Account.Website
0062w00000KCU11AAH	125000	Negotiation/Review	United Oil & Gas Corp.	Energy	http://www.us.com
0062w00000KCU1xAAH	270000	Proposal/Price Quote	United Oil & Gas Corp.	Energy	http://www.us.com
0062w00000KCU1yAAH	120000	Closed Won	United Oil & Gas Corp.	Energy	http://www.us.com
0062w00000KCU26AAH	270000	Negotiation/Review	United Oil & Gas Corp.	Energy	http://www.us.com
0062w00000KCU29AAH	270000	Closed Won	United Oil & Gas Corp.	Energy	http://www.us.com
0062w00000KCU2ZBAAX	915000	Closed Won	United Oil & Gas Corp.	Energy	http://www.us.com
0062w00000KCU2GAAK	235000	Closed Won	United Oil & Gas Corp.	Energy	http://www.us.com
0062w00000KCU2JHAAX	440000	Closed Won	United Oil & Gas Corp.	Energy	http://www.us.com
0062w00000KCU2JAAK	120000	Closed Won	United Oil & Gas Corp.	Energy	http://www.us.com
0062w00000KCU2LAAK	675000	Needs Analysis	United Oil & Gas Corp.	Energy	http://www.us.com

Below the results, there is a "Logs" section with the query: "Select ID, Amount, StageName, Account.Name, Account.Industry, Account.Website From Opportunity Where Account.Industry = 'Energy' AND Account.AnnualRevenue > 5000". The "History" panel shows the executed query and its results.

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

The screenshot shows the Salesforce Developer Console code editor for an Apex trigger named "CustomerTrigger.apxt". The code is as follows:

```

trigger CustomerTrigger on Customer__c (after insert, after update) {
    List<Billing__c> BillingList = new List<Billing__c>();

    for (Customer__c objCustomer: Trigger.new)
    {
        if (objCustomer.Active__c == False)
        {
            Billing__c objbill = new Billing__c();
            objbill.Status__c = 'Paid';
            objbill.Amount_Paid__c=1000000;
            BillingList.add(objbill);
        }
    }
    insert Billinglist;
}

```

The code editor includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The "Problems" tab is currently selected, showing no errors.

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

Output :

The screenshot shows the Salesforce IDE interface. At the top, there's a menu bar with File, Edit, Debug, Test, Workspace, Help, and a dropdown for API Version set to 57. Below the menu is a toolbar with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Tests tab is currently selected. In the center, there's a code editor window titled "CustomerTriggerTest.apc" which contains the following Apex test class:

```
1  @isTest
2  public class CustomerTriggerTest{
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 is a test run history table. It has columns for Status, Test Run, Enqueued Time, Duration, Failures, Total, and Overall Code Coverage. The table shows several test runs, with one failing test named "CustomerTriggerTestClass". The overall code coverage is listed as 27%.

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage
✓	7072w00008MUKxh	Mon Feb 27 2023 15:43:43 GM...	0:00	0	2	Class Overall: 27%
✓	trihdhtps.tt_UtUiControllerTest		0:00	0	2	Billing: 0% / 0/6
✓	testGetCurrentUser		0:00	0	1	CustomerTrigger: 100% / 9/9
✓	testResetMyPassword		0:00	0	1	DiscountClass: 0% / 0/4
✗	7072w00008MUKxE	Mon Feb 27 2023 16:15:02 GM...	0:00	1	3	DML: 0% / 0/10
✗	CustomerTriggerTestClass		0:00	1	1	normalCustomer: 0% / 0/2
✗	testName		0:00	0	2	PremiumCustomer: 0% / 0/2
✓	trihdhtps.tt_UtUiControllerTest		0:00	0	1	

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

```

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 }

```

The screenshot shows the Salesforce IDE interface with the trigger code for 'DisqualifyTestLeads'. The code checks if the FirstName or LastName contains the word 'test' or is blank, and if so, sets the status to 'Disqualified'. The code coverage is listed as 'None'.

Exercise 15 :- Write a Test Class for DisqualifyTestLeads.

Output :

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage
✓	7072w00008MULdq	Mon Feb 27 2023 16:24:50 GM...		0	4	Class: Billing Percent: 41% Lines: 0/6
✓	7072w00008MUJd9	Mon Feb 27 2023 16:19:35 GM...		0	2	CustomerTrigger 100% 9/9
✓	7072w00008MULq8	Mon Feb 27 2023 16:20:05 GM...		0	3	DiscountClass 0% 0/4
✗	7072w00008MULjZ	Mon Feb 27 2023 16:14:05 GM...		0	2	DisqualifyTestLeads 100% 8/8
✗	7072w00008MULj66	Mon Feb 27 2023 16:16:00 GM...		0	2	DML 0% 0/10
						... normalCustomer 0% 0/7

The screenshot shows the Salesforce IDE interface with the trigger code for 'DisqualifyTestLeads'. The code coverage is now at 100%. Below the code, a 'Tests' tab is selected, showing a table of test results. The table lists various test runs, their enqueue time, duration, failures, and total tests. It also includes an 'Overall Code Coverage' section showing the class name, percent coverage, and lines covered.

Exercise 16 :- Create a Visualforce page which displays Opportunity fields as output fields.

The screenshot shows the Salesforce IDE interface. At the top, there's a menu bar with File, Edit, Debug, Test, Workspace, Help, and a preview section. Below the menu is a toolbar with Preview, API Version (set to 57), and other icons. 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>
13
```

Below the code, there's a tabs bar with Logs, Tests (which is selected), Checkpoints, Query Editor, View State, Progress, and Problems. The Tests tab shows a list of test runs:

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	7072w00008MUjdq	Mon Feb 27 2023 16:24:50 GM...		0	4
✓	7072w00008MUIControllerTest			0	2
✓	7072w00008MUjLeads			0	1
✓	7072w00008MUjTrigger			0	1
✓	7072w00008MUj9	Mon Feb 27 2023 16:19:35 GM...		0	2
✓	7072w00008MUjA8	Mon Feb 27 2023 16:20:05 GM...		0	3
✗	7072w00008MUj6Z	Mon Feb 27 2023 16:14:05 GM...		0	2
✗	7072w00008MUj66	Mon Feb 27 2023 16:16:00 GM...		0	2

To the right of the test results, there's a "Overall Code Coverage" table:

Class	Percent	Lines
Overall	41%	0/6
Billing	0%	9/9
CustomerTrigger	100%	0/4
DiscountClass	0%	8/8
DisqualifyTestLeads	100%	0/10
DML	0%	0/2
normalCustomer	0%	0/2

Output :

The screenshot shows a Visualforce page titled "Opportunities". The page has a table with four columns: Opportunity Name, Amount, Close Date, and Account Name. The rows are empty, indicating no data is currently displayed.

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 Visualforce editor interface. The title bar says "File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >". Below it is a toolbar with "AccountList.vfp" highlighted in orange, "Preview", "API Version: 57", and a dropdown menu. The main 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>
```

Output :

- [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.

The screenshot shows the Salesforce IDE interface with the 'NewCaseList.vfp' page open. The code editor contains the following Apex page 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   </apex:repeat>
5 </apex:page>
```

The tabs at the bottom of the editor are: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The 'Tests' tab is currently selected.

The screenshot shows the Salesforce IDE interface with the 'NewCaseListController.apxc' controller class open. The code editor contains the following Apex controller 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 }
```

The tabs at the bottom of the editor are: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The 'Problems' tab is currently selected.

Output :

The screenshot shows the browser output window displaying the following list of case numbers:

- 00001002
- 00001016
- 00001024

The browser interface includes zoom controls (- 200% +) and a 'Reset' button.

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](#)