



Capstone Project Report
On
“Travel App & Code Playground App”

Submitted by:-

Name: VEMPALLI MAHAMMAD AFROZ

Batch: WIP-SF-13

LMS Id: MGSA_517

Date: 28/02/2023

GitHub link: <https://github.com/MahammadAfroz/Travel-Application-Code-Playground-Application>

Table of Contents

Abstract.....	5
Introduction	6
Software Requirements	7
Module-1	8-33
Prerequisites:-.....	8
Create a New Trailhead Playground:-	8
Downloaded the Import File:-.....	8
EXERCISE-1:-	9-16
Step 1:- Create a new custom lightning App.	9-10
Step 2:- Create a Department custom object.....	10-11
Step 3:- Create the custom fields.	11-12
Step 4:- Create a Travel Approval Object.....	12-14
Step 5:- Import Department Data using “Data Import Wizard”.	15-16
EXERCISE-2:-	17-33
Step 1:- Create Travel Approval record.....	17
Step 2:- Create Expense Item Object.	17-18
Step 3:- Create the following custom fields.	18-19
Step 4:- Create Expense Items.	19-20
Step 5:- Create a User.....	20-21
Step 6: Add user Eric Executive as your Manager.	21
Step 7:- Customize the Travel Approval Default search layout.....	22
Step 8:- Select fields to display in the Travel Approval “All” List view.	23
Step 9:- Create Travel approval custom a List View “Open Out of State Travel Requests”.....	23-24
Step 10:- Select fields to display in the Travel Approval “Open Out of State Travel Requests”.....	25
Step 11:- Customize the Travel Approval Page Layout.	25-28
Step 12:- Customize the Expense Item Related List under the Travel Approval page layout.	28-29
Step 13:- Enable Chatter on the Travel Approval Object.....	29-33
Module-2	34-68
EXERCISE-1:-	34-55
Step 1:- Create Validation Rule.	34-36
Step 2:- Create a Roll-Up Summary Field on Travel Approval object.	36-37
Step 3:- Create Formula Fields.....	37-38

Step 4:- Create a Formula field on the Travel Approval object to show an image based on the Status field.....	38-39
Step 5:- Create a Record – Triggered Flow.....	40-48
Step 6: Create the Approval Processes:-	48-55
EXERCISE-2:-	55-68
Step 1:- Use Data Import Wizard to import Travel Approval records.....	55-57
Step 2:- Create a Travel Requests by Department Report.....	58-61
Step 3:- Create a Travel Requests by Month Report.....	61-63
Step 4:- Create a Travel Approvals Dashboard.....	64-68
Module:-3	69-103
Prerequisites:-.....	69
Create a New Trailhead Playground:-	69
EXERCISE-1:-	70-75
Create a new custom lightning App.....	70-72
Create a Custom ‘Customer’ Object:-	72
Create a Custom Fields for ‘Customer’ Object.....	73
Create a Custom ‘Billing’ Object:-.....	73-74
Create a Custom Fields for ‘Billing’ Object.....	74-75
EXERCISE-2:- Use Execute Anonymous to define and execute the following code: ..	76-81
1. Define a String Variable & use string method ‘endsWith’ to display the output.....	76
2. Define 2 Date type variables, use Date method today() & addDays(30) to display the output.....	77
3. Display the output of an Integer variable from string ‘10’ and then add 20 to it.	78
4. Define a String Variable & use string method length() to display the output.	79
5. Define a List of integer and display the output using add(), get(), set(), clear(), methods.	80
6. Use Execute Anonymous to define and execute the following code to display the value of x = 0 to 9....	81
EXERCISE-3:- Answer the following in True or False:	82
EXERCISE-4:- Answer the following in True or False:	83
EXERCISE-5:- Answer the following in True or False:	84
EXERCISE-6:- Write a program and execute to demo the use of “If...else if...else”.....	85
EXERCISE-7:- Write a program to execute and demo the use of “Apex – for Loop”. ...	86
EXERCISE-8:- Write a Class to demo the use of Constants in Apex.....	87-88
EXERCISE-9:- Write a Class to demo the use of Interface in Apex.....	88-90
EXERCISE-10:- Demo on DML Insert Operation Using Database methods.	90-92

EXERCISE-11:- Write and execute SOQL queries from Developer Console.	92-93
EXERCISE-12:- Write an Apex Trigger.....	93-94
EXERCISE-13:- Write a Test Class for CustomerTrigger.....	94-96
EXERCISE-14:- Write an Apex Trigger.....	97
EXERCISE-15:- Write a Test Class for DisqualifyTestLeads.....	98-99
EXERCISE-16:- Create a Visualforce page which displays Opportunity fields as output fields.....	99-100
EXERCISE-17:- Create a Visualforce page which shows a list of Accounts linked to their record pages.....	100-101
EXERCISE-18:- Create a Visualforce page that uses a custom controller to display a list of cases with the status of 'New'.....	102-103
Future Enhancements.....	104
References	105

Abstract

The proposed Travel Approval App is a web-based solution that simplifies the process of requesting and approving travel requests within an organization. The app is designed to automate the approval process, eliminate manual tasks, and ensure compliance with company travel policies.

The app includes a user-friendly interface that allows employees to easily submit travel requests, select travel dates, destinations, and expenses. The app also provides the ability to attach supporting documents, such as flight itineraries and hotel bookings, to the travel request for review.

The app features an automated approval workflow that routes the travel request to the appropriate approver based on predefined approval rules. The app also provides visibility into the approval process, allowing approvers to easily review and approve or reject travel requests.

Furthermore, the app includes a comprehensive reporting system that provides insights into travel spend and usage, enabling companies to optimize travel budgets and identify potential cost-saving opportunities. The app also integrates with various travel booking systems, enabling companies to book travel directly within the app.

Overall, the Travel Approval App is a powerful tool that streamlines the travel approval process, reduces manual tasks, and ensures compliance with company travel policies. The app also provides valuable data insights to help companies make informed decisions about their travel programs.

Introduction

Salesforce is a leading cloud-based customer relationship management (CRM) platform that helps organizations manage their sales, marketing, customer service, and other business operations. The platform provides a suite of tools and applications that help businesses improve their customer engagement, automate business processes, and gain valuable insights into their operations.

Salesforce is a highly customizable platform that allows businesses to tailor their CRM solutions to meet their unique needs. The platform includes a range of tools, such as Sales Cloud, Marketing Cloud, Service Cloud, and Commerce Cloud that help businesses manage customer relationships at every stage of the customer journey.

One of the key benefits of Salesforce is its cloud-based architecture. This allows users to access their data and applications from anywhere, at any time, using any device with an internet connection. It also enables businesses to easily scale their operations and add new users and functionality as their needs change.

In addition, Salesforce provides a range of tools and resources to help businesses succeed, including a large community of users, developers, and partners who share best practices and provide support. The platform is constantly evolving, with new features and capabilities added regularly to help businesses stay ahead of the competition.

Overall, Salesforce is a powerful and flexible platform that helps businesses improve their customer engagement, automate their operations, and gain valuable insights into their business. With its user-friendly interface, advanced features, and cloud-based architecture, Salesforce is a must-have tool for any organization looking to streamline their operations and grow their business.

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

Module-1

Prerequisites:-

Create a New Trailhead Playground:-

1. For this project, you need to create a new Trailhead Playground. Named it as “Travel App”.



2. Click “Create”. It typically takes 3-4 minutes to create a new Trailhead Playground.

A screenshot of a Trailhead playground list. It shows a card for 'Travel App' with the following details:

Username	resilient-shark-8mwhwe.com
Type	Trailhead Playground
Created	2/21/2023
Last Activity	Created on 2/21/2023

On the right side of the card are three buttons: 'Rename', 'Disconnect', and 'Launch'. The 'Launch' button is highlighted.

3. Once Created Open it by clicking “Launch”.

Downloaded the Import File:-

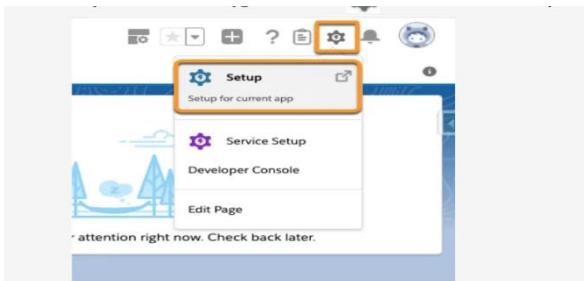
Before you begin, download Workshop Lab Files, which you use later in the project via this

link: <https://developer.salesforce.com/files/TravelAppWorkshopFiles.zip>. Unzip to a directory of your choice on your computer.

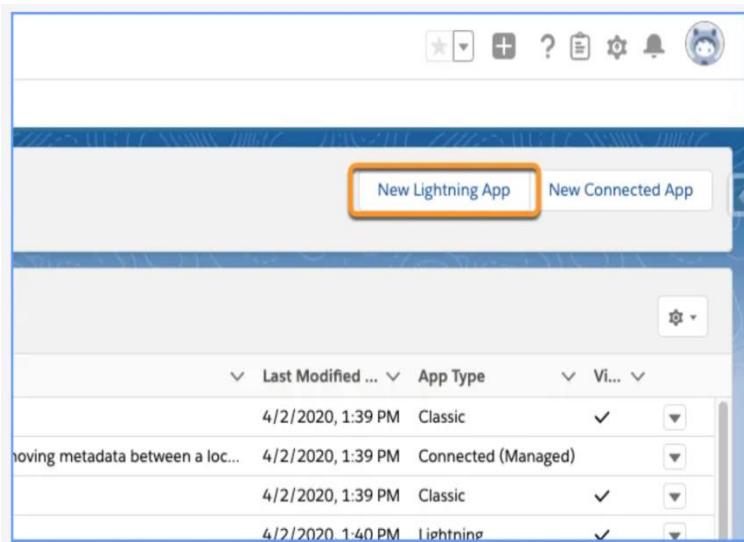
EXERCISE-1:-

Step 1:- Create a new custom lightning App.

1. In your Trailhead Playground, click  and then select **Setup**.



2. In the Quick Find box, enter App Manager and select **App Manager**.
3. Click **New Lightning App**.



4. In the App Details & Branding window, enter these details.
5. For App Name, enter “**Travel App**”.
6. For the Image, click **Upload** and select **travel.png** from the files you downloaded above.
7. Click **Next**.
8. On the App Options screen, select **Standard navigation** and click **Next**.
9. On the Utility Items screen, click **Next**.

Navigation Items

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

Available Items

- Accounts
- App Launcher
- Approval Requests
- Assets

Selected Items

- Chatter
- Reports
- Dashboards

10. On the Navigation items screen, select **Chatter, Reports, and Dashboards** from the Available items list, and move them to Selected items list using the arrow. Then click **Next**.
11. On the User Profiles screen, select **System Administrator** and add it to Selected Profiles, then click **Save & Finish**.
12. Click , then search for Travel App and select a “Travel App”. And open it.

Step 2:- Create a Department custom object.

1. From Setup, click **Object Manager**.
2. Click Create, and then select **Custom Object**.
3. Enter These Details:-
 - Label = “Department”
 - Plural = “Departments”
 - Data Type = “Text”
 - Select “Allow Reports”, “Allow Search”.
 - Keep other options as it is.

- Use custom tab of your choice and include it only for Travel App.
- Tab visibility = As it is, Save.

The screenshot shows the Salesforce Setup interface under the Object Manager. A sidebar on the left lists various configuration options for the 'Department' object. The main pane displays the 'Details' section for the 'Department' object. Key fields shown include:

- API Name:** Department__c
- Singular Label:** Department
- Plural Label:** Departments
- Enable Reports:** ✓
- Track Activities:** ✓
- Track Field History:** (checkbox)
- Deployment Status:** Deployed
- Help Settings:** Standard salesforce.com Help Window

At the bottom right of the main pane, there are 'Edit' and 'Delete' buttons, and a message: "Activate Windows Go to Settings to activate Windows."

Step 3:- Create the custom fields.

1. On the next screen, click **Fields & Relationships**, and then click **New**.
2. Select **Text** as the data type, and then click **Next**.
3. Enter these details.
 - For Field Label, enter **Department Code**
 - For Length, enter **10**
 - Select **Required**
 - Select **Unique**, then select **case sensitive**
4. Click Next, Next and Save.
5. Repate 1 to 4 steps for bellow fields.
 - Location, Picklist, Value: Kolkata, Delhi.
 - Department Type, Picklist, Values: Banking,
 - Finance, Education, Energy, IT.

SETUP > OBJECT MANAGER
Department

Fields & Relationships
7 items. Sorted by Field Label

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

Activate Windows
Go to Settings to activate Windows.

6. Create Field Dependency.
7. Controlling Field = “Location”.
8. Dependent Field = “Department Type”.
9. Label Name = “Location”.

SETUP > OBJECT MANAGER
Department

Edit Field Dependency

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

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

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

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

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

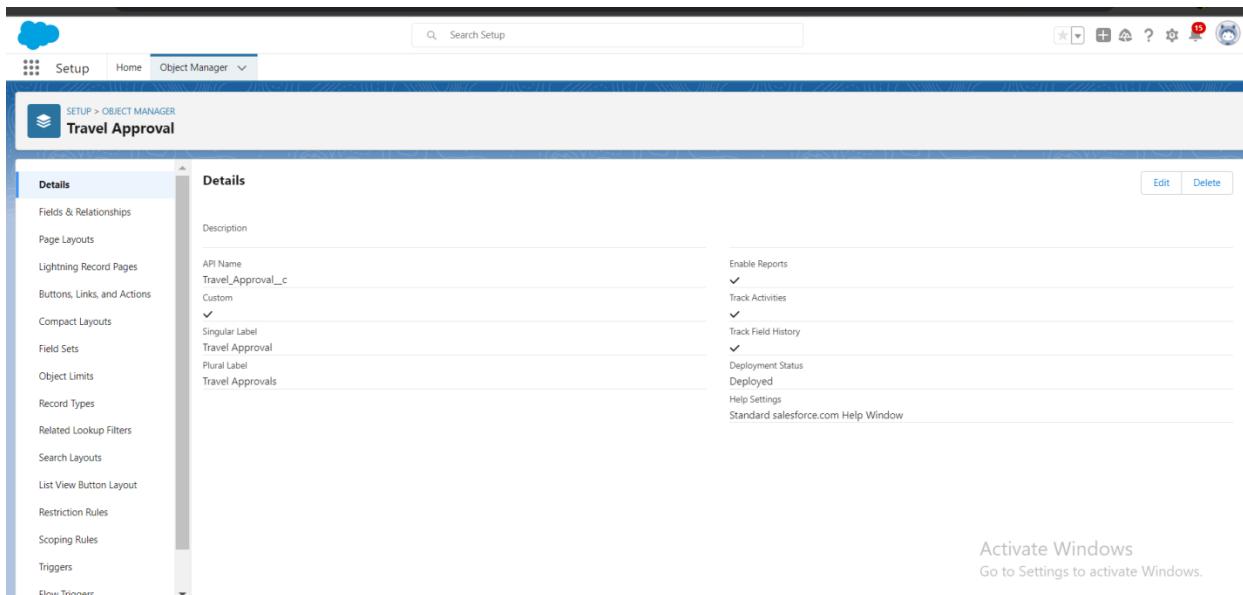
Showing Columns: 1 - 2 (of 2) < Previous | Next > | View All | Go to

Activate Windows
Go to Settings to activate Windows.

10. Save it.

Step 4:- Create a Travel Approval Object.

1. From Setup, click **Object Manager**.
2. Click Create, and then select **Custom Object**.
3. Enter These Details:-
 - **Label = “Travel Approval””**
 - **Plural = “Travel Approvals””**
 - **Object Name = “Travel_Approval”**
 - **Record Name = “Travel Approval #”**
 - **Data Type = “Auto Number”**
 - **Display format “TA{00000}”.**
 - **Starting Number = 1.**
4. Select these additional options...
 - **Allow Reports.**
 - **Allow Activities.**
 - **Track Field History.**
 - **Allow Search**
 - **Add Notes & Attachments.**
 - Create custom Tab of your choice.
 - Add the Tab only to Travel App.
 - **Click Save.**



5. Create the Custom fields...
6. On the next screen, click **Fields & Relationships**, and then click **New**.
7. Select **Text Area** as the data type, and then click **Next**.
8. For Field Label, enter Purpose of Trip.
9. Click **Next**.
10. Leave the Establish field-level security screen as is, and click **Next**.
11. Keep the Add to page layouts screen as is, and click **Save**.

To create the remaining fields for the Travel Approval object, repeat step 9 to begin, then follow the parameters below.

12. Create the **Status** field with a **Picklist** data type. Click **Enter values, with each value separated by a new line**. Use the following values:

- **New**
- **Submitted**
- **Pending Approval**
- **Approved**
- **Rejected**
- **Draft**

13. Create the **Trip Start Date** field with a **Date** data type.
14. Create the **Trip End Date** field with a **Date** data type.
15. Create the **Out-of-State** field with a **Checkbox** data type.
16. Create the **Destination State** field with a **Text** data type. Set the text **Length** to 2.
17. Create the **Department** field with a **Lookup Relationship** data type, and then select **Department** for the Related To field. Click **Next** 4 times, and then **Save**.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		✓
Department	Department__c	Lookup(Department)		✓
Destination State	Destination_State__c	Text(2)		✓
Last Modified By	LastModifiedById	Lookup(User)		✓
Out-of-State	Out_of_State__c	Checkbox		✓
Owner	OwnerId	Lookup(User,Group)		✓
Purpose of Trip	Purpose_of_Trip__c	Text Area(255)		✓
Status	Status__c	Picklist		✓
Status Indicator	Status_Indicator__c	Formula (Text)		✓
Total Expenses	Total_Expenses__c	Roll-Up Summary (SUM Expense Item)		✓
Travel Approval #	Name	Auto Number		✓
Trip End Date	Trip_End_Date__c	Date		✓
Trip Start Date	Trip_Start_Date__c	Date		✓

Step 5:- Import Department Data using “Data Import Wizard”.

1. From Setup, click the **Home** tab.
2. In the Quick Find box, enter Data Import and select **Data Import Wizard**.
3. Click **Launch Wizard!**

The screenshot shows the Salesforce Data Import Wizard interface. At the top, there's a navigation bar with icons for Setup, Home, and Object Manager. A search bar is on the right. Below the navigation is a title bar with "Data Import Wizard" and a "Help for this page" link. Underneath is a section titled "Recent Import Jobs" with a table header row containing columns for Status, Object, Records Created, Records Updated, Records Failed, Start Date, and Processing Time (ms). A prominent dark button labeled "Bulk Api Monitoring" is centered below the table. To the left, there's a sidebar with a binoculars icon and the text "Before you import your data...". On the right, three collapsed tips are visible: "Clean up your data import file", "Make sure your field names match Salesforce field names", and "Don't import too many records at once". Below this is a section titled "Import your data in 3 easy steps!" with four sequential steps: "Pre-step: Prepare your data for import" (brush icon), "Choose data to import" (cloud icon), "Edit field mapping" (box icon), and "Review and start import" (checkmark icon). A large green button at the bottom center says "Launch Wizard!".

4. Click the **Custom Objects** tab and select the **Departments** object.
5. Next, select **Add new records**.
6. Drag and drop the **Departments.csv** file you downloaded in Step 1 - Create a Travel Approval Lightning App to the Drag CSV file here to upload section, or click the CSV icon and browse to select your file. Select **Next**.
7. Since the field names in the CSV file (CSV Header) are the same as the field names in your object (Mapped Salesforce Object), the fields are automatically mapped. Click **Next**.

Edit	Mapped Salesforce Object	CSV Header	Example
Change	Department Name	Department Name	Audit Services
Change	Department Code	Department Code	405-01

8. The next screen gives you a summary of your data import. Click **Start Import**.

9. Click **OK** on the popup.

Now that you've imported department data, you can test out the app.

10. Open **Travel App**.

11. Click on **Departments Tab**.

12. Click **Recently View And select All**.

	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	

EXERCISE-2:-

Step 1:- Create Travel Approval record.

1. Click on Travel Approval tab and Click New.
2. Enter these details:-
 - Purpose of Trip = “Attend Dreamforce”
 - Status = “Draft”.
 - Trip Start Date/End Date = “Pick any Date”
 - Out of state = True, Destination State = CA.
 - Department = Technology. Save.

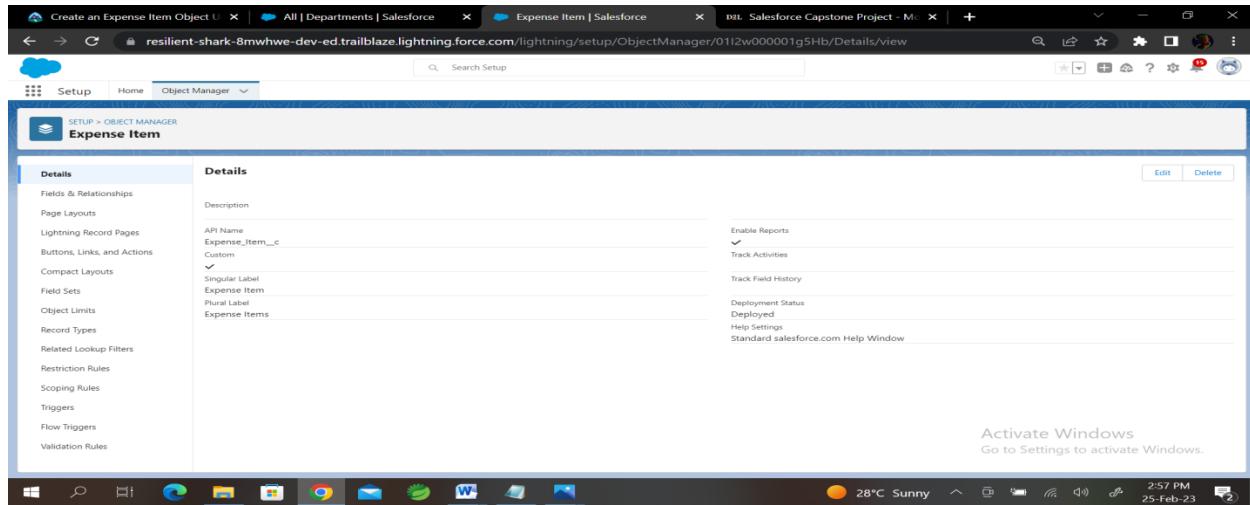
The screenshot shows the Salesforce interface for the Travel App. The top navigation bar includes links for Travel App, Chatter, Reports, Dashboards, Departments, and Travel Approvals. The main content area displays a Travel Approval record with the ID TA-00001. The record has the following details:

Field	Value
Travel Approval #	TA-00001
Status	Draft
Out-of-State	<input checked="" type="checkbox"/>
Destination State	CA
Total Expenses	₹1,320.00
Status Indicator	(gear icon)
Trip Info	
Purpose of Trip	Attend Dreamforce
Trip Start Date	21/02/2023
Trip End Date	28/02/2023
Created By	VEMPALLI MAHAMMAD AFROZ, 21/02/2023, 3:04 pm
Owner	VEMPALLI MAHAMMAD AFROZ
Department	Technology
Last Modified By	VEMPALLI MAHAMMAD AFROZ, 21/02/2023, 3:04 pm

Step 2:- Create Expense Item Object.

1. Navigate back to **Object Manager**.
2. Click **Create** then select **Custom Object**.
3. Enter these details:-
 - Label = “Expense Item”.
 - Plural = “Expense Items”.
 - Starts with vowel sound = Checked.
 - Record Name = Expense Item Number.
 - Data Type = Auto Number.
 - Display Format = E – {00000}
 - Starting Number = 1

- Allow Reports.
- Do not create Custom Tab.
- Save.



Step 3:- Create the following custom fields.

1. Click Fields & Relationships, and click **New**.
2. For data type, select **Currency**.
3. Enter these details.
 - For Field Label, enter **Amount**.
 - For Length, enter **16**.
 - For Decimal places, enter **2**.
4. Select Required
5. Click **Next, Next**, then **Save & New**.
6. Next, create the **Expense Type field**.
7. Select **Picklist** as the data type.
8. Select Enter values, with each value separated by a new line.
9. Add these values:
 - **Airfare**
 - **Hotel**
 - **Rental car**
 - **Meals**
 - **Other**

10. Select **Required**.

11. Click **Next**, **Next**, then **Save & New**.

Create the **Travel Approval** field.

1. Select **Master-Detail Relationship** data type, click **Next**.
2. Select **Travel Approval** from the Related To menu.
3. Click **Next** four times, then click **Save**.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes the Salesforce logo, a search bar labeled "Search Setup", and various global buttons. Below the header, the breadcrumb trail reads "SETUP > OBJECT MANAGER" and the specific object is "Expense Item". On the left, a sidebar lists various setup options under "Fields & Relationships". The main content area is titled "Fields & Relationships" and displays a table of fields. The table has columns for "FIELD LABEL", "FIELD NAME", "DATA TYPE", "CONTROLLING FIELD", and "INDEXED". The table contains the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount_c	Currency(16, 2)		
Created By	CreatedBy	Lookup(User)		
Expense Item Number	Name	Auto Number		✓
Expense Type	Expense_Type_c	Picklist		
Last Modified By	LastModifiedBy	Lookup(User)		
Travel Approval	Travel_Approval_c	Master-Detail(Travel Approval)		✓

A watermark for "Activate Windows" is visible in the bottom right corner of the screenshot.

Step 4:- Create Expense Items.

Finally, add expense estimates for Airfare and Hotel to the Travel Approval. Click the Related tab.

1. Click **New**, then enter these details:
 - Amount: 450
2. Expense Type: **Airfare**
3. Click **Save**.
4. Click **New** then enter these details.

- Amount: 870
- Expense Type: **Hotel**
 - Click **Save**.

The screenshot shows a Salesforce Trailhead Playground interface. At the top, there's a navigation bar with links like 'Travel App', 'Chatter', 'Reports', 'Dashboards', 'Departments', and 'Travel Approvals'. Below the navigation is a search bar and a toolbar with various icons.

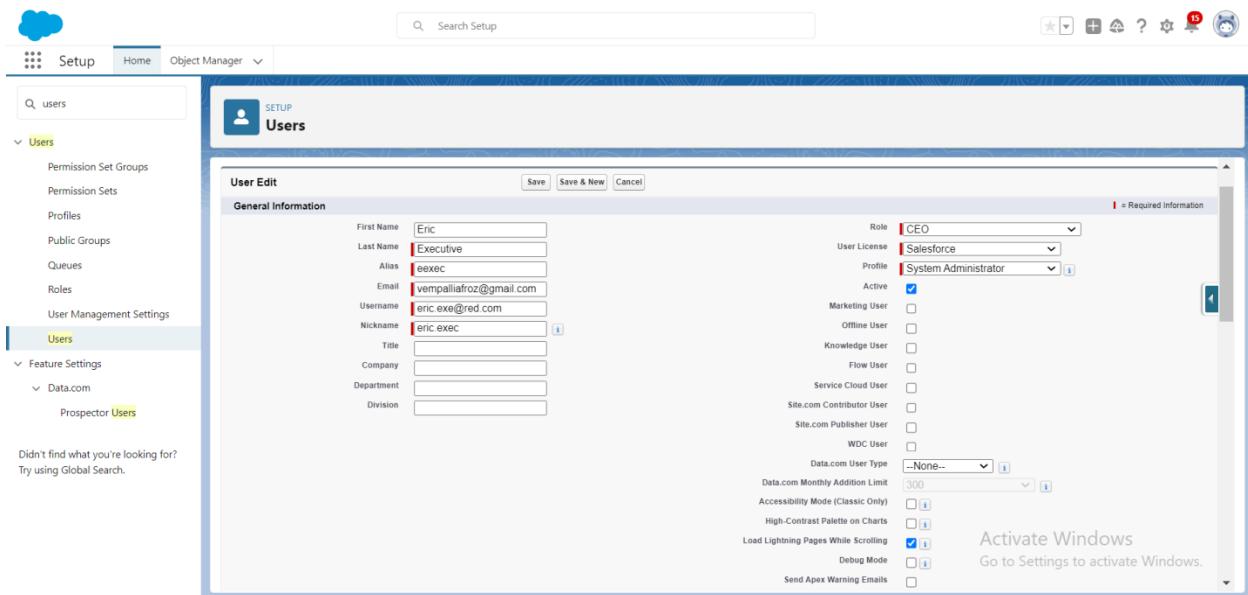
The main content area displays a 'Travel Approval' record with ID 'TA-00001'. The record has tabs for 'Related' and 'Details'. Under 'Related', there are sections for 'Notes & Attachments (0)', 'Expense Items (2)', and 'Approval History (0)'. The 'Expense Items' section contains two entries:

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

At the bottom of the expense items section is a 'View All' link. To the right of the record details is an 'Activity' sidebar. It shows a header for 'Activity' and 'Chatter', followed by a list of activity types (Calendar, Tasks, Calls, etc.) with dropdown arrows. Below this is a 'Filters' section with options 'All time', 'All activities', 'All types', and a refresh button. The sidebar also includes 'Upcoming & Overdue' and 'Past activity' sections, both of which are currently empty.

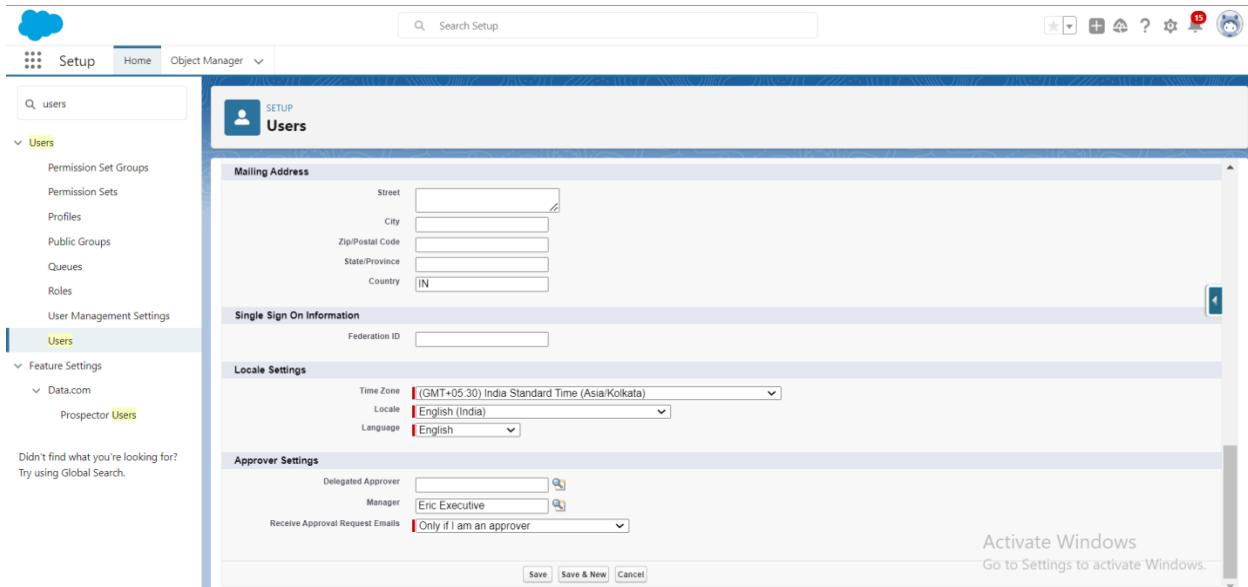
Step 5:- Create a User.

- In your Trailhead Playground, click  and then select **Setup**.
- Enter Users in the Quick Find box, and select **Users**.
- Click **New User** and enter these details:-
 - First Name = “Eric”.
 - Last Name = “Executive”
 - Email = vempalliafroz@gmail.com”
 - Username Name = “eric.exe@red.com”
 - Role = “CEO”
 - License = Salesforce.
 - Profile = System Administrator.
 - Save.



Step 6: Add user Eric Executive as your Manager.

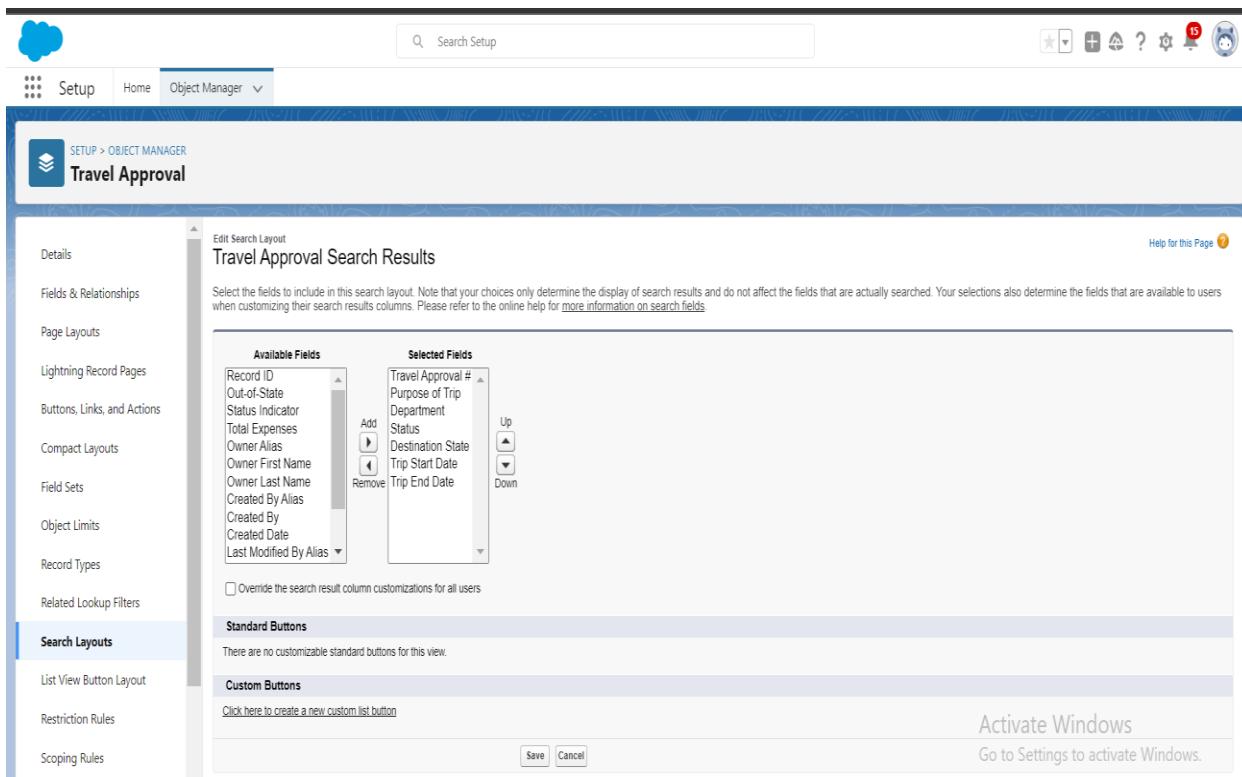
1. From Setup, enter Users in the Quick Find box and select **Users**.
2. Select your user account in the list provided.
3. Click **Edit**.
4. Scroll down to Approver Settings. Set your manager as Eric Executive by clicking and selecting **Eric Executive** from the list.



5. Click **Save**.

Step 7:- Customize the Travel Approval Default search layout.

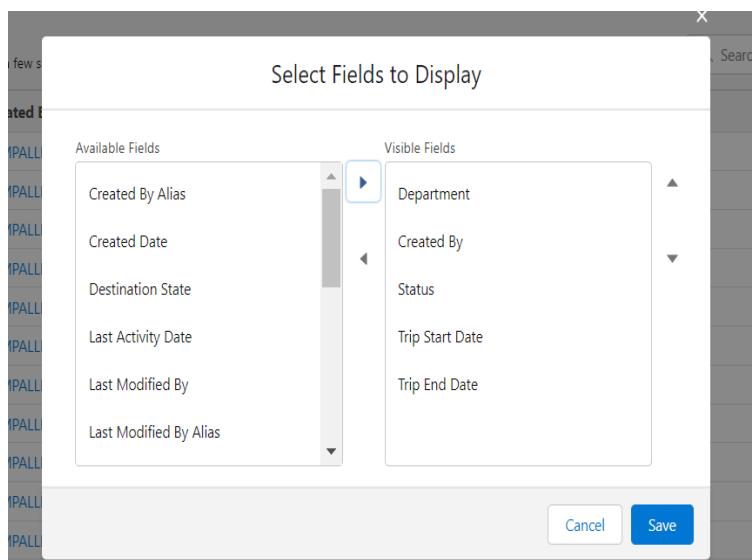
1. From the App Launcher, find and select the **Travel App** and select the **Travel Approvals** tab.
2. Select record **TA-00001** under **All LIST VIEWS**.
3. Click the gear icon, and then select **Edit Object**. This loads the object configuration page for the Travel Approval object.
4. Click **Search Layouts**. Click the down arrow for the Default Layout and select **Edit** from the dropdown.
5. Use the Add arrow to move these fields into the Selected Fields column, in order.
 - Purpose of Trip
 - Department
 - Status
 - Destination State
 - Trip Start Date
 - Trip End Date



6. Click **Save**.

Step 8:- Select fields to display in the Travel Approval “All” List view.

1. Using the App Launcher, navigate back to the **Travel App** and click the **Travel Approvals** tab.
2. Click **Recently Viewed** and select the **All** list view.
3. Click the gear icon then select **Select Fields to Display** from the dropdown.
4. Use the Add arrow to move these fields to the Selected Fields column, in order.
 - Department
 - Created By
 - Status
 - Trip Start Date
 - Trip End Date



5. Click **Save**.

Step 9:- Create Travel approval custom List View “Open Out of State Travel Requests”.

1. Click the List View gear icon and select **New** option.
2. Name the new list view **Open Out of State Travel Requests** and select **All users can see this list view**.
3. Click **Save**.

The screenshot shows the Salesforce interface for the Travel Approvals app. The top navigation bar includes links for Travel App, Chatter, Reports, Dashboards, Departments, and Travel Approvals. The main content area is titled "Travel Approvals" and "Open Out of State Travel Requests". A search bar at the top right contains the placeholder "Search...". Below the search bar are buttons for "New", "Import", "Change Owner", and "Printable View". A toolbar with various icons is located at the top right. The main table displays one item: "TA-00001" from the "Technology" department, created by "VEMPALLI MAHAMMAD AFROZ" in Draft status, traveling to CA from 21/02/2023 to 28/02/2023.

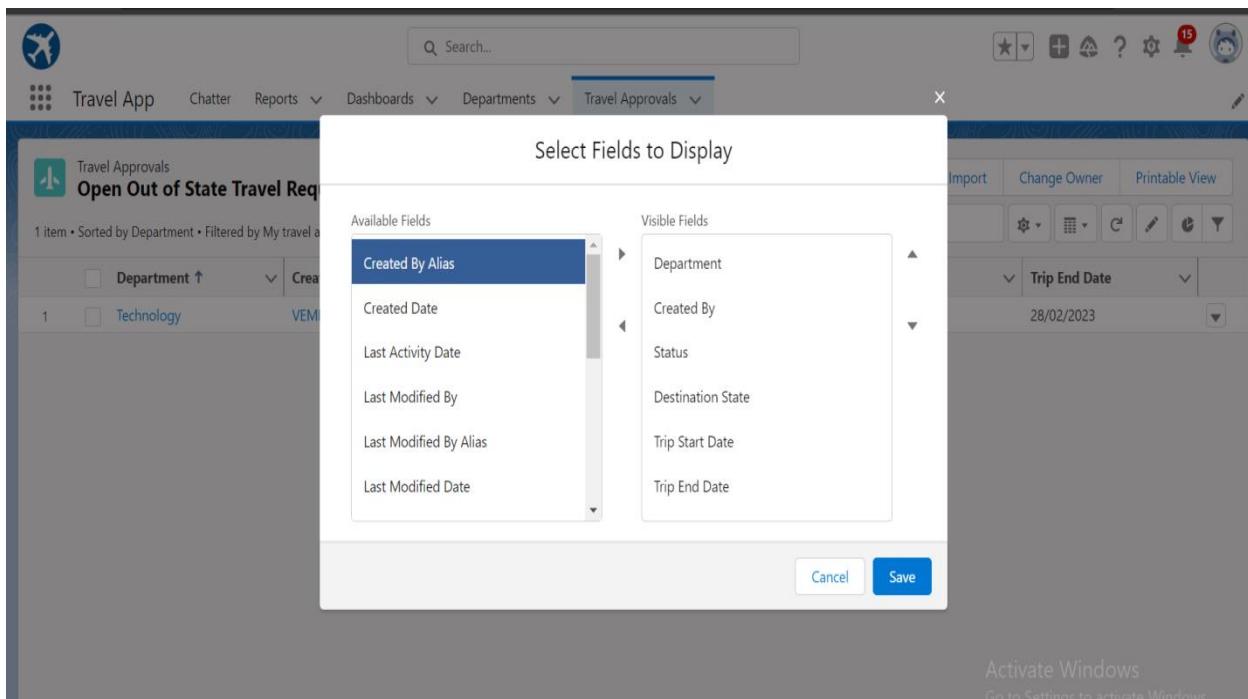
4. Click **Add Filter** and enter these details:
 - Field: **Out-of-State**
 - Operator: **equals**
 - Value: **True**
5. Click **Done**.
6. Click **Add Filter** again and enter these details:
 - Field: **Status**
 - Operator: **not equal to**
 - Value: **Approved, Rejected**
7. Click **Done**.

The screenshot shows the same Salesforce interface as the previous one, but with the "Filters" sidebar open on the right. The sidebar lists two filters: "Out-of-State equals True" and "Status not equal to Approved, Rejected". At the bottom of the sidebar, there are buttons for "Add Filter", "Remove All", and "Activate Windows".

8. Click **Save**.

Step 10:- Select fields to display in the Travel Approval “Open Out of State Travel Requests”.

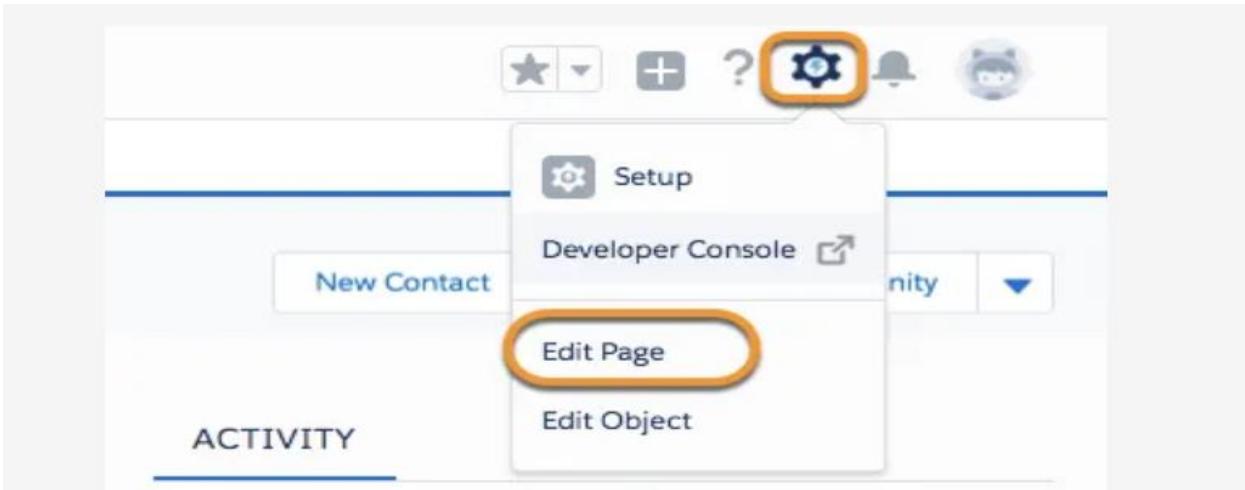
1. Click the gear icon and select **Select Fields to Display**.
2. Use the Add arrow to move the following fields to the Visible Fields column, in order.
 - Department
 - Created By
 - Status
 - Destination State
 - Trip Start Date
 - Trip End Date



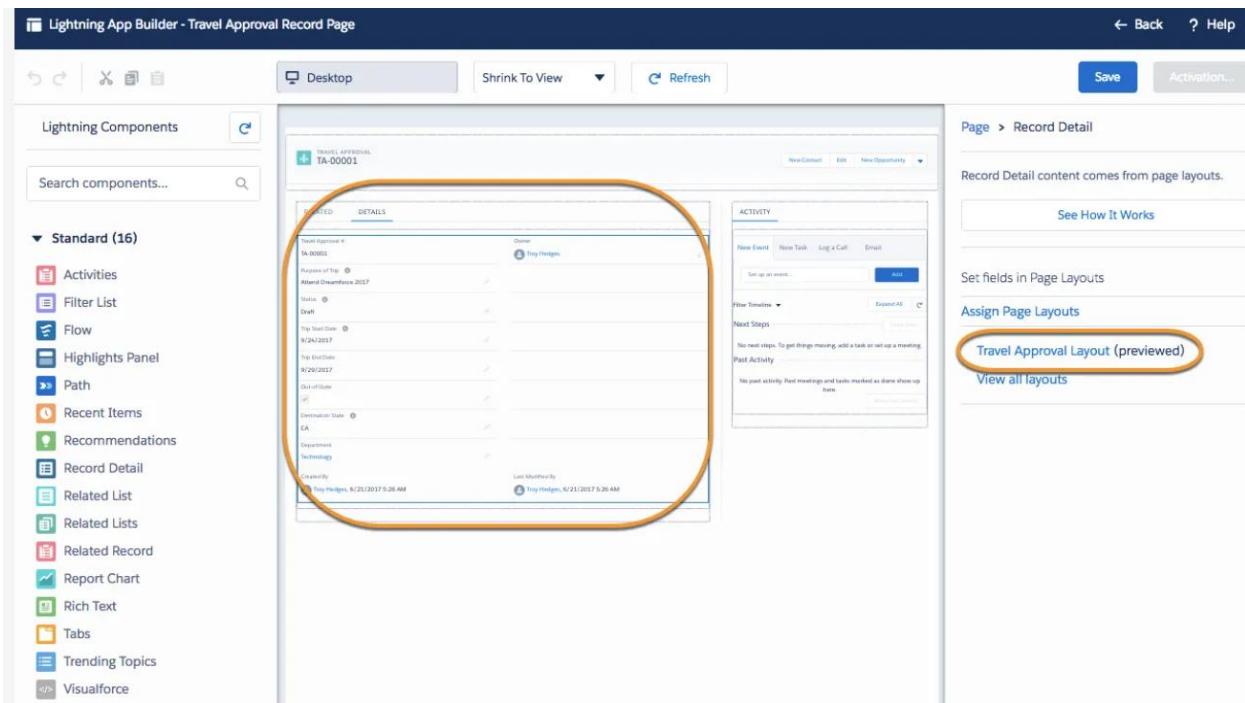
3. Click **Save**.

Step 11:- Customize the Travel Approval Page Layout.

1. In the Travel App app, click the **Travel Approvals** tab and open TA-00001.
2. Click and then select **Edit Page**.



3. Left-click once in the middle of the form to select it. You should see a light blue border around the form. On the right-hand side, click **Travel Approval Layout (previewed)**.



4. Scroll down to the **Travel Approval Detail** section. Here's where you can change the order and position of fields and create a new section to group the trip detail fields.

The screenshot shows the Salesforce Layout Editor interface. At the top, there's a toolbar with Save, Quick Save, Preview As..., Cancel, Undo, Redo, and Layout Properties buttons. On the left, a sidebar titled 'Fields' lists Buttons, Quick Actions, Salesforce1 & Lightning Actions, Expanded Lookups, and Related Lists. In the center, a 'Quick Find' search bar is followed by a table with four columns: 'Section', 'Destination State', 'Purpose of Trip', and 'Trip Start Date'. Below this table, another row contains 'Blank Space', 'Last Modified By', 'Status', and 'Created By', 'Out-of-State', 'Travel Approval #'. A third row contains 'Department', 'Owner', and 'Trip End Date'. At the bottom of the editor, the page title 'Travel Approval Detail' is visible, along with standard buttons (Edit, Delete, Clone, Change Owner, Change Record Type, Sharing) and custom buttons.

Information (Header visible on edit only)

Travel Approval #	GEN-2004-001234	Owner	Sample User
Purpose of Trip	Sample Purpose of Trip		
Status	Sample Status		
Trip Start Date	6/21/2017		
Trip End Date	6/21/2017		
Out-of-State	✓		
Destination State	Sample Destination State		
Department	Sample Department		

5. Drag **Section** from the top pane to the lower pane directly below the Information section. When dragging over the page, you get a visual indicator of where you can drop the new section.
6. Name the section **Trip Info** and leave the rest of the settings at their default values, then click **OK**.
7. Drag **Purpose of Trip**, **Trip Start Date**, and **Trip End Date** from the Information section to the left-hand column of the Trip Info section.
8. Drag **Out-of-State** and **Destination State** from the Information section to the right-hand column of the Trip Info section.
9. Drag the **Department** field from the left-hand column of the Information section to the right-hand column.

10. Click Save.

Step 12:- Customize the Expense Item Related List under the Travel Approval page layout.

- From Page Layouts in the Object Manager, select Travel Approval Layout. Scroll down to the Expense Item related list.
- Click the Wrench icon in the header tab.

3. Move the **Expense Type** and **Amount** fields from Available Fields to Selected Fields.
4. Click **OK**. Click **Yes** if you're asked if you want to Overwrite Users' Related Lists Customizations.
5. Click **Save**.
6. Navigate back to a Travel Approval record and look at the related lists. You should see the extra columns you added to the expense item related list.

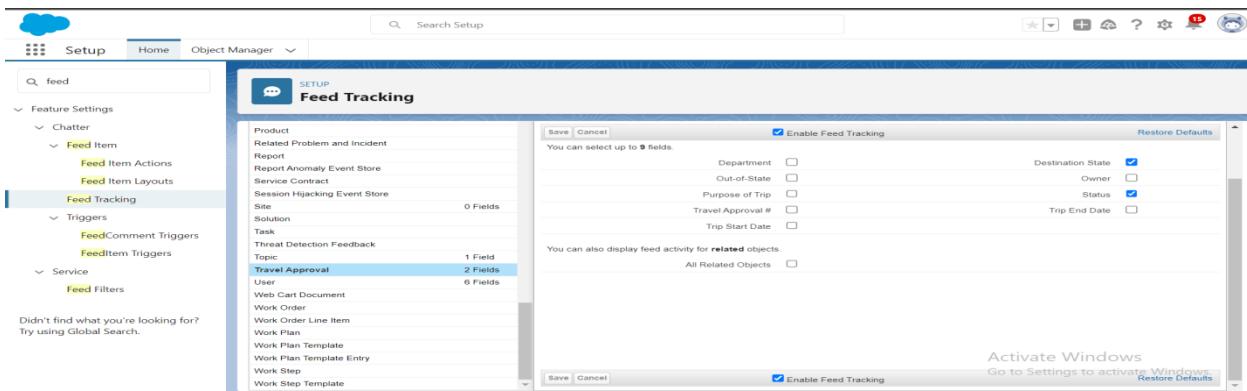
The screenshot shows the Salesforce interface for a 'Travel Approval' record. At the top, there's a navigation bar with links for 'Travel App', 'Chatter', 'Reports', 'Dashboards', 'Departments', and 'Travel Approvals'. Below the navigation is a search bar and a toolbar with various icons. The main content area displays a travel approval record with ID 'TA-00001'. On the left, there are tabs for 'Related' and 'Details'. Under 'Related', there are three sections: 'Notes & Attachments (0)', 'Expense Items (2)', and 'Approval History (0)'. The 'Expense Items' section contains a table with two rows:

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

On the right side of the page, there's a 'Activity' sidebar with a 'Chatter' tab selected. It shows a feed with icons for different activity types like calls, messages, and tasks. A message at the bottom of the sidebar says 'No activities to show. Get started by sending an email, scheduling a task, and more.' and 'No past activity. Past meetings and tasks marked as done show up here.'

Step 13:- Enable Chatter on the Travel Approval Object.

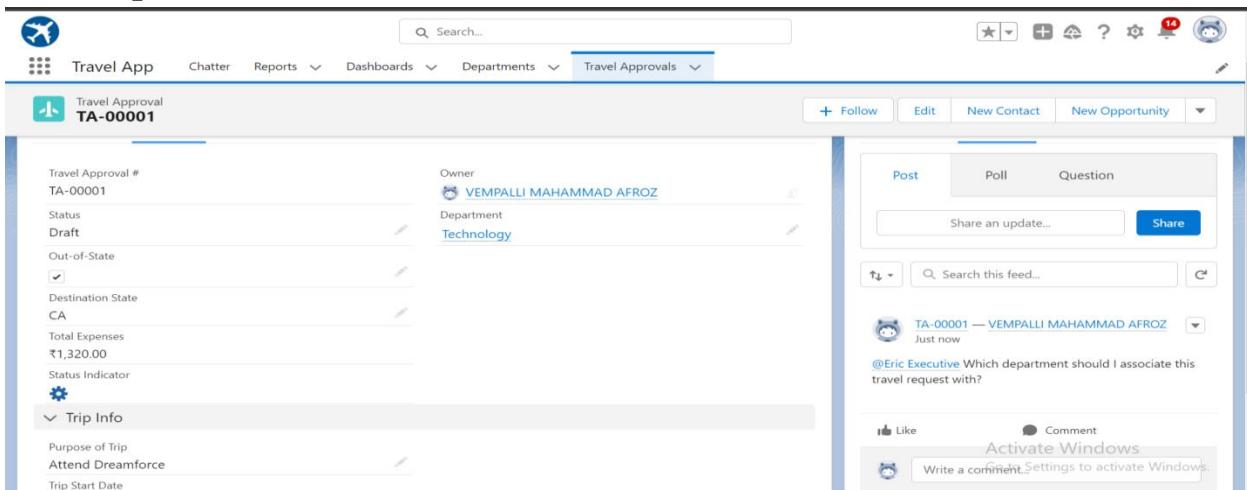
1. From Setup, enter Feed Tracking in the Quick Find box, then select **Feed Tracking**.
2. In the list of available objects, click **Travel Approval**.
3. Select **Enable Feed Tracking**.
4. Select these fields:
 - Destination State
 - Status



- Click Save.

Test Collaboration:-

- Via the App Launcher, navigate back to the **Travel App** and click the **Travel Approvals** tab. Click the **TA-00001** record. For record collaboration, notice you have a tab called Chatter.
- Click the **Chatter** tab. Note: By default you have the option to make a Post, submit a Poll for feedback, or ask a Question.
- Click the **Chatter** tab. Note: By default you have the option to make a Post, submit a Poll for feedback, or ask a Question.
- Click the **Post** sub-tab. Click in the area labeled **Share an update...** and enter @E and select **Eric Executive** from the popup window. Add the remainder of the message: **Which department should I associate this travel request with?**



5. Click **Share** to post. Note: Eric and you will both receive an email and notification about the post.

VEMPALLI MAHAMMAD AFROZ mentioned you in a post [Inbox](#)

 VEMPALLI MAHAMMAD AFROZ in Chatter <reply@chatter.salesforce.com>

to me ▾ 10:16 (2 minutes ago) [Star](#) [Reply](#) [More](#)

 TA-00001 — VEMPALLI MAHAMMAD AFROZ

@Eric Executive Which department should I associate this travel request with?

[View/Comment](#) or reply to this email

- 6.** In your email application, reply to the email, saying: **Technology is the correct department.**

The screenshot shows the Salesforce interface for the 'Travel App'. At the top, there's a navigation bar with a user icon, the text 'Logged in as Eric Executive (eric.exe@red.com)', and a 'Log out as Eric Executive' link. To the right are various global navigation icons. Below the navigation bar is a search bar with the placeholder 'Search...'. The main content area has a blue header bar with the 'Travel App' logo and several dropdown menus: 'Chatter', 'Reports', 'Dashboards', 'Departments', and 'Travel Approvals'. On the left, a sidebar titled 'What I Follow' lists items like 'To Me', 'Bookmarked', 'Company Highlights', 'My Drafts', and 'STREAMS'. Under 'STREAMS', it says 'You don't have any streams yet. Try creating one!' with a '+' sign. Another section titled 'RECENT GROUPS' says 'Aw, you don't have any groups! Why not create or join some now?' with a '+' sign. The main content area displays a feed of posts. The first post is from 'TA-00001 — VEMPALLI MAHMAD AFROZ' posted 3m ago, asking '@Eric Executive Which department should I associate this travel request with?'. Below it, 'Eric Executive' responded a few seconds ago, saying 'Technology is the correct department.' A comment input field with a placeholder 'Write a comment...' and a toolbar below it are also visible.

7. We get a mail after Eric gave a comment.

The screenshot shows an email from 'Eric Executive in Chatter <reply@chatter.salesforce.com>' to 'VEMPALLI MAHAMMAD AFROZ' on Thursday, 23/02/2023 at 10:20. A caution message at the top reads: 'CAUTION: This email is received from an external domain. Open the hyperlink(s) & attachment(s) with caution.' Below the message, there is a reply button, the recipient's name 'Eric Executive', and the text 'Technology is the correct department.' A blue button at the bottom says 'View/Comment or reply to this email'. Under the heading 'Original post', it shows a message from 'TA-00001 — VEMPALLI MAHAMMAD AFROZ' posted on Thursday, 23 February 2023 at 10:16 am. The message asks '@Eric Executive Which department should I associate this travel request with?'. To the right of the message is an 'Activate Windows' advertisement.

8. Refresh your travel app page to see the response in Chatter.

The screenshot shows the 'Travel App' interface. The left sidebar includes links for 'What I Follow', 'To Me', 'Bookmarked', 'Company Highlights' (which is selected), 'My Drafts', 'STREAMS' (with a note about creating one), and 'RECENT GROUPS' (with a note about joining). The main area displays the Chatter feed with a post from 'TA-00001 — VEMPALLI MAHAMMAD AFROZ' 5m ago, asking '@Eric Executive Which department should I associate this travel request with?'. Eric Executive responded 'Technology is the correct department.' Below the feed is a comment input field. On the right side, there is an 'Einstein Recommendations' sidebar featuring suggestions for '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. An 'Activate Windows' advertisement is also visible.

9. The Travel Approval App should look like this.

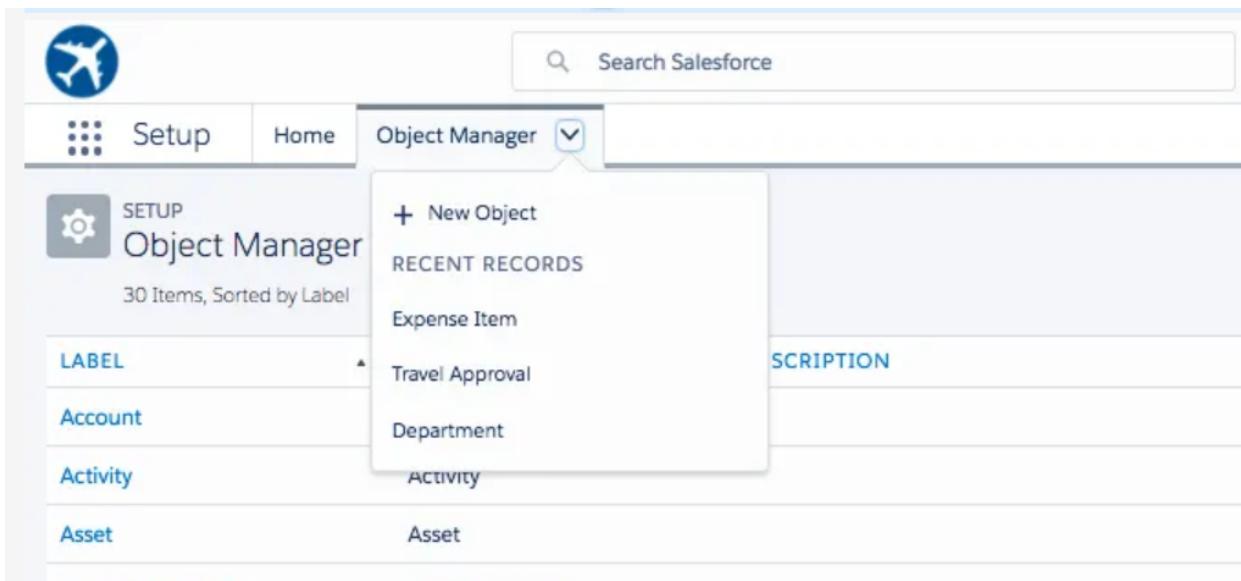
The screenshot shows the Salesforce interface for a Travel Approval application. At the top, there's a navigation bar with links for Travel App, Chatter, Reports, Dashboards, Departments, and Travel Approvals. The main content area displays a travel approval record for "Travel Approval # TA-00001". The "Details" tab is selected, showing fields such as Travel Approval # (TA-00001), Status (Draft), Out-of-State (checked), Destination State (CA), Total Expenses (\$1,320.00), and Status Indicator (blue gear icon). A "Trip Info" section is expanded, showing Purpose of Trip (Attend Dreamforce), Trip Start Date (21/02/2023), and Trip End Date (28/02/2023). The record was Created By VEMPALLI MAHMAD AFROZ on 21/02/2023 at 3:04 pm, and Last Modified By the same user on the same date and time. On the right side, there's a Chatter feed with a post from VEMPALLI MAHMAD AFROZ asking which department to associate the travel request with. Another comment from Eric Executive suggests Technology is the correct department. A "Activate Windows" message is visible at the bottom of the Chatter feed.

Module-2

EXERCISE-1:-

Step 1:- Create Validation Rule.

1. Click  and select **Setup**.
2. Click **Object Manager** and then click the  icon. This provides a list of recent objects you've edited.



3. Select **Travel Approval** to open the configuration page for the Travel Approval object.
4. Click **Validation Rules**.
5. Click **New**.
6. Enter the following values:-
 - Business Logic: Trip end date must always be greater than ($>=$) the trip start date.
 - Name: **Trip end date after start date**.
 - Make sure to keep “Active” selected/checked.
 - Condition Formula: “**Trip_End_Date__c <= Trip_Start_Date__c**”
 - Error Message: **Trip end date must be greater than or equal to start date**.

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

Here's what the rule should look like.

SETUP > OBJECT MANAGER
Travel Approval

Validation Rules

Rule Name: **Trip_end_date_after_start_date**

Active:

Description:

Error Condition Formula

Example: `Discount_Percent_c > 0.30` More Examples...
Display an error if Discount is more than 30%

If this formula expression is true, display the text defined in the Error Message area

Functions

Insert Field Insert Operator ▾

Trip_End_Date__c <= Trip_Start_Date__c

Check Syntax: No errors found

Error Message

Example: `Discount percent cannot exceed 30%`
This message will appear when Error Condition formula is true

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

This error message can either appear at the top of the page or below a specific field on the page

Error Location: Top of Page Field **Trip End Date**

Save Save & New Cancel

7. Check Syntax. If No Error Found.

8. Click Save.

SETUP > OBJECT MANAGER
Travel Approval

Travel Approval Validation Rule

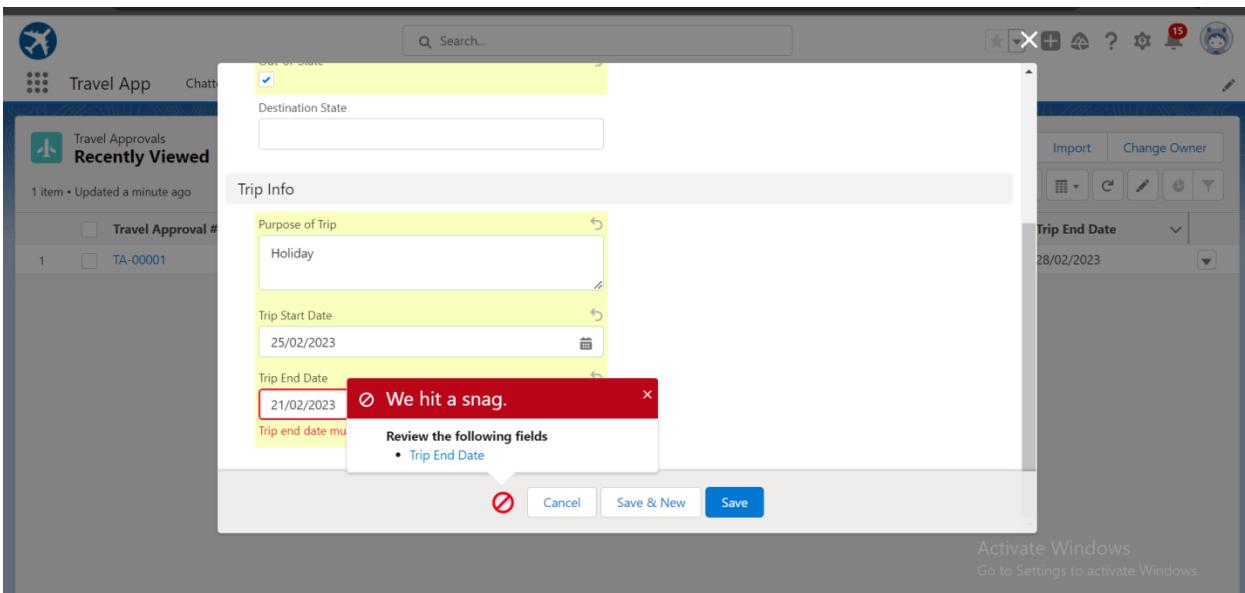
Validation Rule Detail

Rule Name	Trip_end_date_after_start_date	Edit	Clone
Error Condition Formula	<code>Trip_End_Date__c <= Trip_Start_Date__c</code>	Active	<input checked="" type="checkbox"/>
Error Message	Trip end date must be greater than or equal to start date	Error Location	Trip End Date
Description		Created By	VEMPALLI MAHAMMAD AFROZ, 21/02/2023, 7:33 pm
Created By		Modified By	VEMPALLI MAHAMMAD AFROZ, 25/02/2023, 10:32 pm

Help for this Page

Activate Windows
Go to Settings to activate Windows.

Look what we have done.



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

1. From the Travel Approval object, select **Fields & Relationships**.
2. Click **New**.
3. Select the **Roll-Up Summary** data type.
4. Click **Next**.
5. Enter the following values for the field details:
 - Field Label: Total Expenses
 - Field Name: Total_Expenses (this automatically gets set when you tab out of the Field Label field)

A screenshot of the 'New Custom Field' wizard, Step 2. The title bar says 'Travel Approval' and 'New Custom Field'. The top navigation bar says 'Step 2 of 5' and has 'Previous', 'Next', and 'Cancel' buttons. The form fields are:

- Field Label: Total Expenses
- Field Name: Total_Expenses
- Description: (empty)
- Help Text: (empty)

Below the form is another set of 'Previous', 'Next', and 'Cancel' buttons.

6. Click **Next**.
7. Configure the roll-up calculation.
 - Summarized Object: **Expense Items**
 - Roll-Up Type: **SUM**
 - Field to Aggregate: **Amount**
 - Filter Criteria: **All records should be included in the calculation**

The screenshot shows the 'New Custom Field' setup page in Salesforce. The top navigation bar includes 'Travel Approval' and 'Help for this Page'. The main title is 'New Custom Field'. The current step is 'Step 3 of 5'. The configuration section is titled 'Step 3. Define the summary calculation'. It shows the following settings:

- Select Object to Summarize:** Master Object is 'Travel Approval' and Summarized Object is 'Expense Items'.
- Select Roll-Up Type:** The 'SUM' option is selected, and the 'Field to Aggregate' dropdown is set to 'Amount'.
- Filter Criteria:** The 'All records should be included in the calculation' option is selected.

Buttons at the bottom include 'Previous', 'Next', and 'Cancel'.

8. Click **Next, Next, Save**.

Step 3:- Create Formula Fields.

1. Click the **Home** tab to navigate back to the main setup page.
2. Click **Custom Code | Static Resources** (or enter **Static** in the Quick Find to filter down the options).
3. Click **New**.
4. Enter the following values for your static resource:
 - Name = StatusImages
 - File = StatusImages.zip [upload it from the Project Folder]
 - Cache-Control = Public
 - Save

The screenshot shows the Salesforce Setup interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. A search bar says 'Search Setup'. The main content area is titled 'Static Resources' under 'SETUP'. A sub-section titled 'StatusImages' is selected. The 'Static Resource Detail' section shows the following information:

Name	StatusImages
Namespace Prefix	
Description	
MIME Type	application/zip
Cache Control	Public
Size	39,130 bytes
View file	
Created By	VEMPALLI MAHMAD AFROZ, 21/02/2023, 4:03 pm
Last Modified By	VEMPALLI MAHMAD AFROZ, 21/02/2023, 4:03 pm

At the bottom of the detail view are 'Edit', 'Delete', and 'Where is this used?' buttons. To the right, there's a message 'Activate Windows' with a link 'Go to Settings to activate Windows.'

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

1. Click the icon next to the **Object Manager** tab. This provides a shortcut to the Object Manager for the recent objects you have edited.
2. Select **Travel Approval**.

The screenshot shows the Salesforce Home page. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. A search bar says 'Search Salesforce'. The main content area shows a sidebar with 'Static' and 'Custom Code' sections, and a 'Static Resources' section which is currently selected. A dropdown menu from the 'Object Manager' tab lists several objects:

- + New Object
- RECENT RECORDS
- Expense Item
- Travel Approval
- Department

3. Select Fields & Relationships.
4. Click New
5. Select Formula data type.
6. Click Next.

7. Enter the following values:

- Field Label: Status Indicator
- Field Name: Status_Indicator (This automatically gets sent when you tab out of the Field Label field)

8. Formula Return Type: Text

9. Click Next.

10. Copy and paste the following formula into the formula editor.

```
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)))
```

11. Click Next, Next, Save.

The screenshot shows the Salesforce Setup interface for creating a custom field. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, Buttons, etc. The main area shows the 'Travel Approval Custom Field' named 'Status Indicator'. The formula is defined as:

```
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)))
```

The formula type is set to 'Formula'.

Step 5:- Create a Record – Triggered Flow.

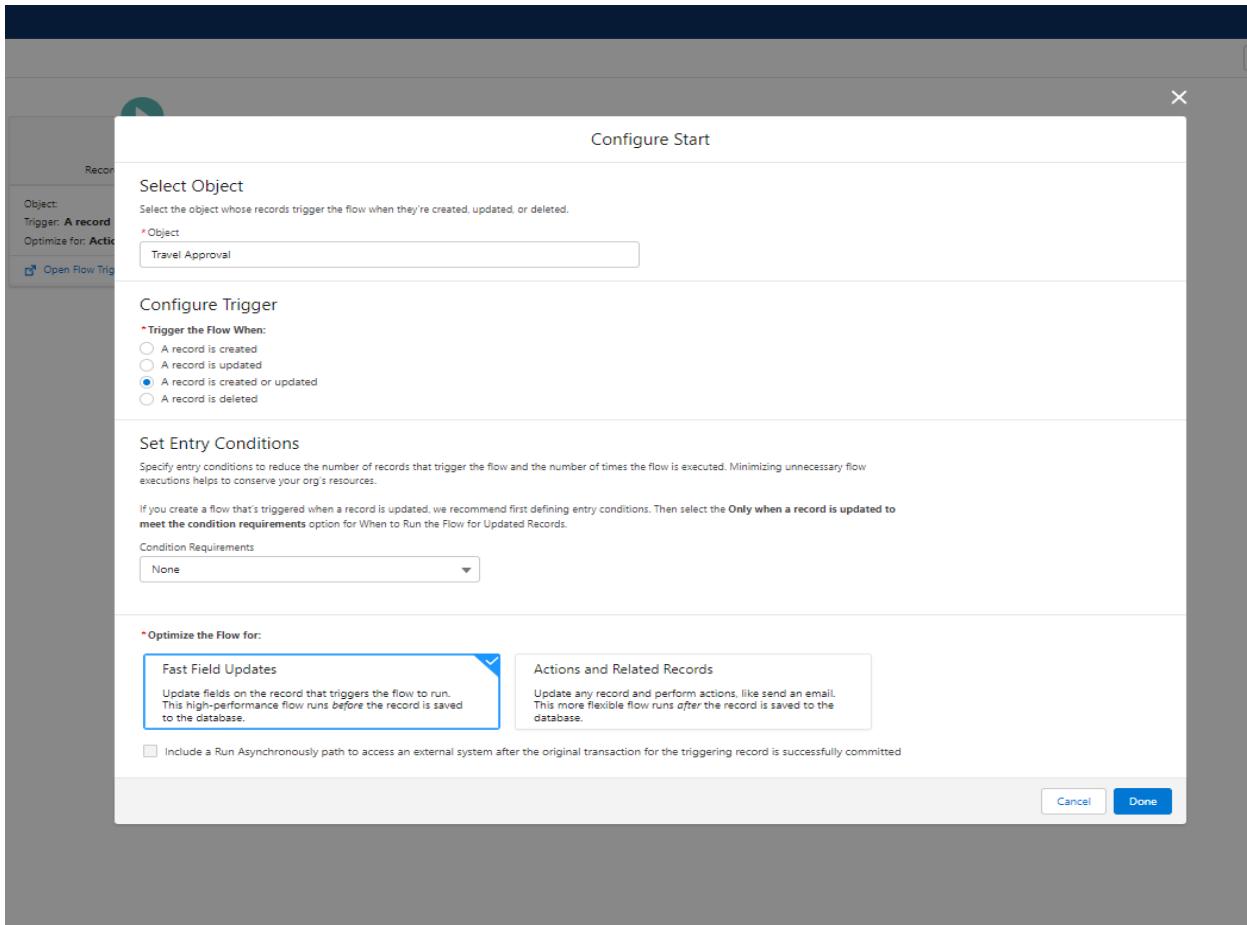
1. From Setup, click the **Home** tab.
2. In Quick Find and search for flows.
3. And select **Flows**.

The screenshot shows the Salesforce Setup Home page. At the top, there's a search bar labeled "Search Setup" and various navigation icons. Below the header, the "Setup" tab is selected. On the left, a sidebar menu includes "Process Automation" (with "Flows" selected), "Identity", and "Login Flows". A message says " Didn't find what you're looking for? Try using Global Search." The main content area is titled "SETUP Flows" and shows a table of "Flow Definitions". The table has columns for "Flow Label", "Process Type", "Active", "Tem...", "Package State", "Pack...", "Last Modifie...", and "Last Modified D...". There are 8 items listed, all of which are Screen Flows and Managed-Installed. The flows include "Cancel Item Flow", "Create a Case", "Create Order Summary Flow", "Create Process Exception Flow", "Reset Password", "Return Item Flow", "Travel Approval Expense Items Flow", and "Verify Identity". The last flow, "Travel Approval Expense Items Flow", was created by "Rachel Dishman" on "12/1/2021, 2:51 PM".

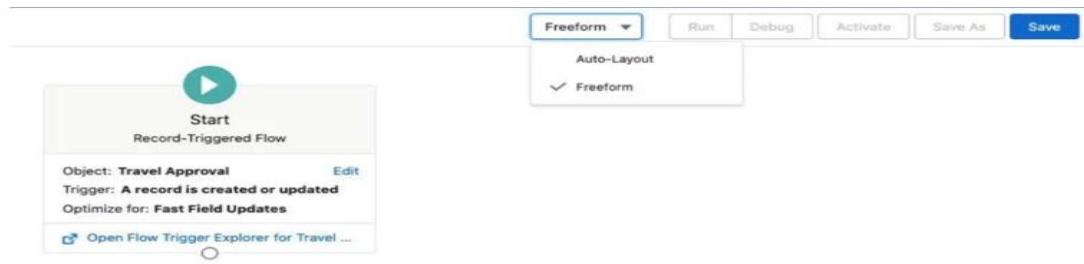
Flow Label ↑	Process Type	Active	Tem...	Package State	Pack...	Last Modifie...	Last Modified D...
Cancel Item Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Create a Case	Screen Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Create Order Summary Flow	Autolaunched Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Create Process Exception Flow	Autolaunched Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Reset Password	Screen Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Return Item Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Travel Approval Expense Items Flow	Screen Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Unmanaged	Rachel Dishman	12/1/2021, 2:51 PM	
Verify Identity	Screen Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			

4. Click **New Flow**.
5. Select **Record-Triggered Flow** then click **Create**.
6. Enter these values:-
 - Object: **Travel Approval**
 - Configure Trigger: Trigger the flow when: **A record is created or updated.**
 - Condition Requirements = **None**.
 - Optimize the Flow For: **Fast Field Updates**.

The Configure Start Screen should look like this:-



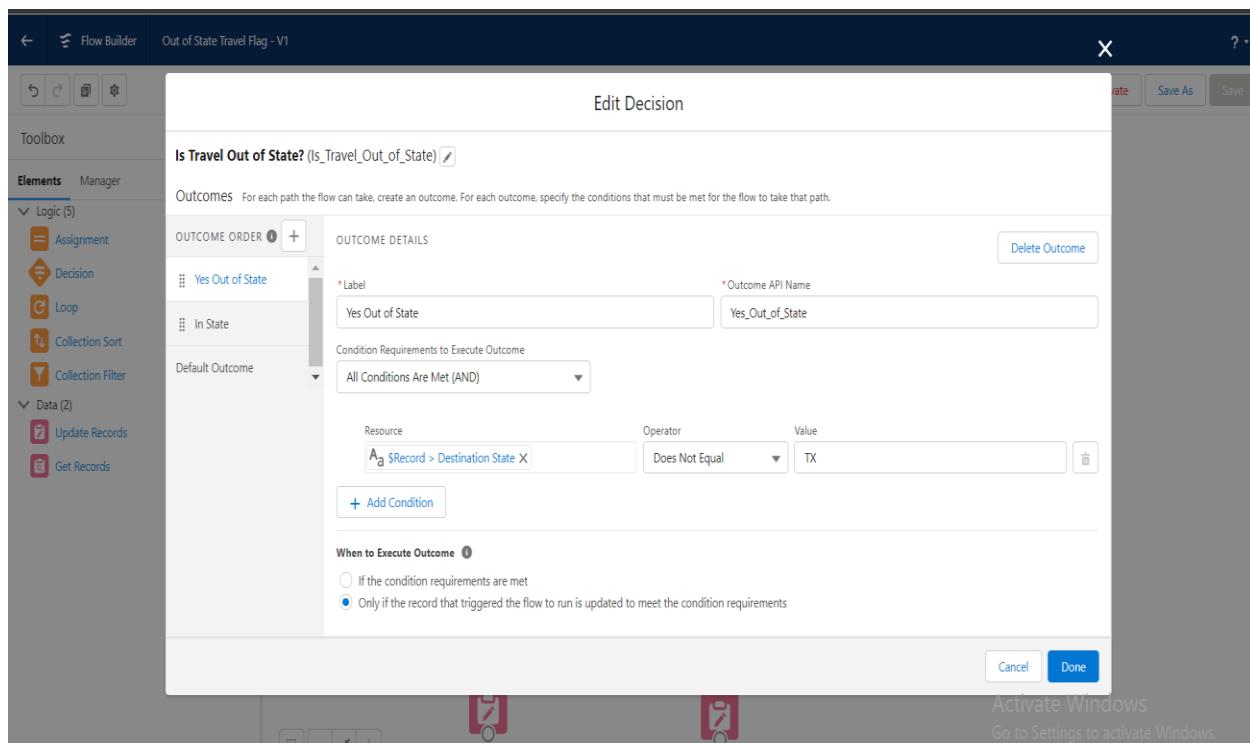
7. Click Done.



8. In the Layout dropdown, select Freeform.

Add a Decision Element to the Flow:-

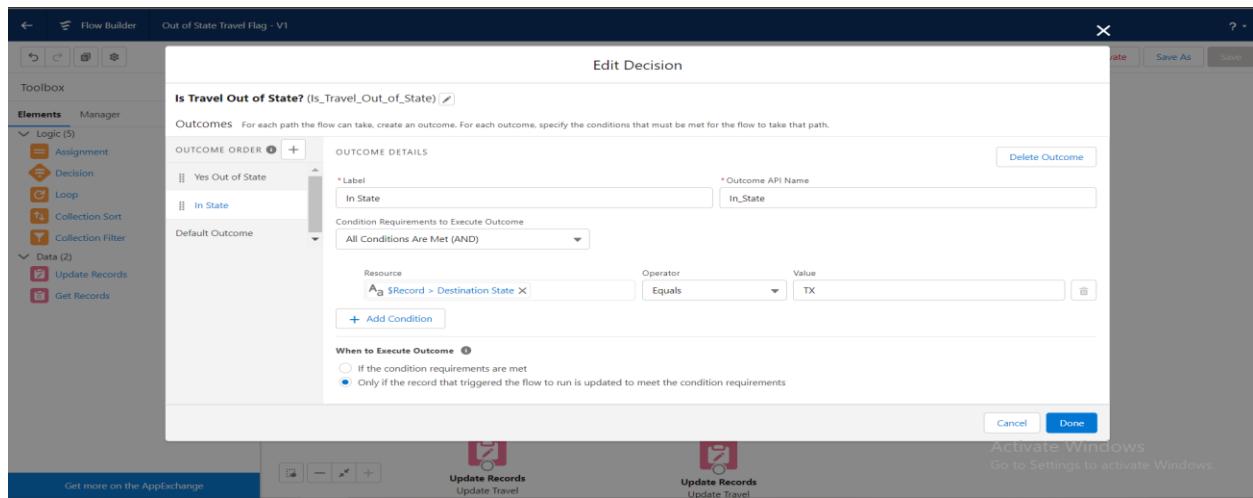
1. From the left-hand toolbox column, drag a **Decision** element onto the flow builder screen.
2. Set the following parameters:-
 - Label = Is Travel Out of State?
 - API Name = Is_Travel_Out_of_State (This automatically gets set when you tab out of the Label field).
 - Description = Leave blank.
3. Set the **Outcomes**.
4. For the first outcome, set these parameters:-
 - Label = **Yes Out of State**.
 - Condition Requirements = **All Conditions Are Met (AND)**.
 - Resource = \$Record > Destination State.
 - Operator: **Does Not Equal**, Value: **TX**.
 - When to Execute the Outcome: - **Only if the record that triggered the flow to run is updated to meet the condition requirements**.



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

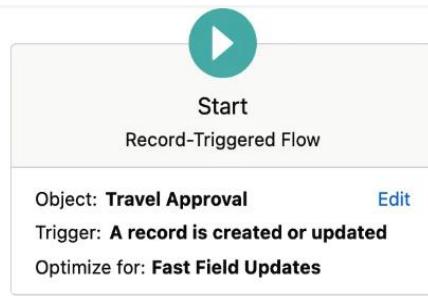
6. Set these parameters:-

- Label = **In State**
- Condition Requirements = **All Conditions Are Met (AND)**.
- Resource = **\$Record > Destination State**
- Operator: **Equal**, Value: **Tx**
- When to Execute the Outcome: **Only if the record that triggered the flow to run is updated to meet the condition requirement.**



7. Click Done

8. Drag the white circle from the Start Flow element to the Decision element you just created to link the direction of the arrow.



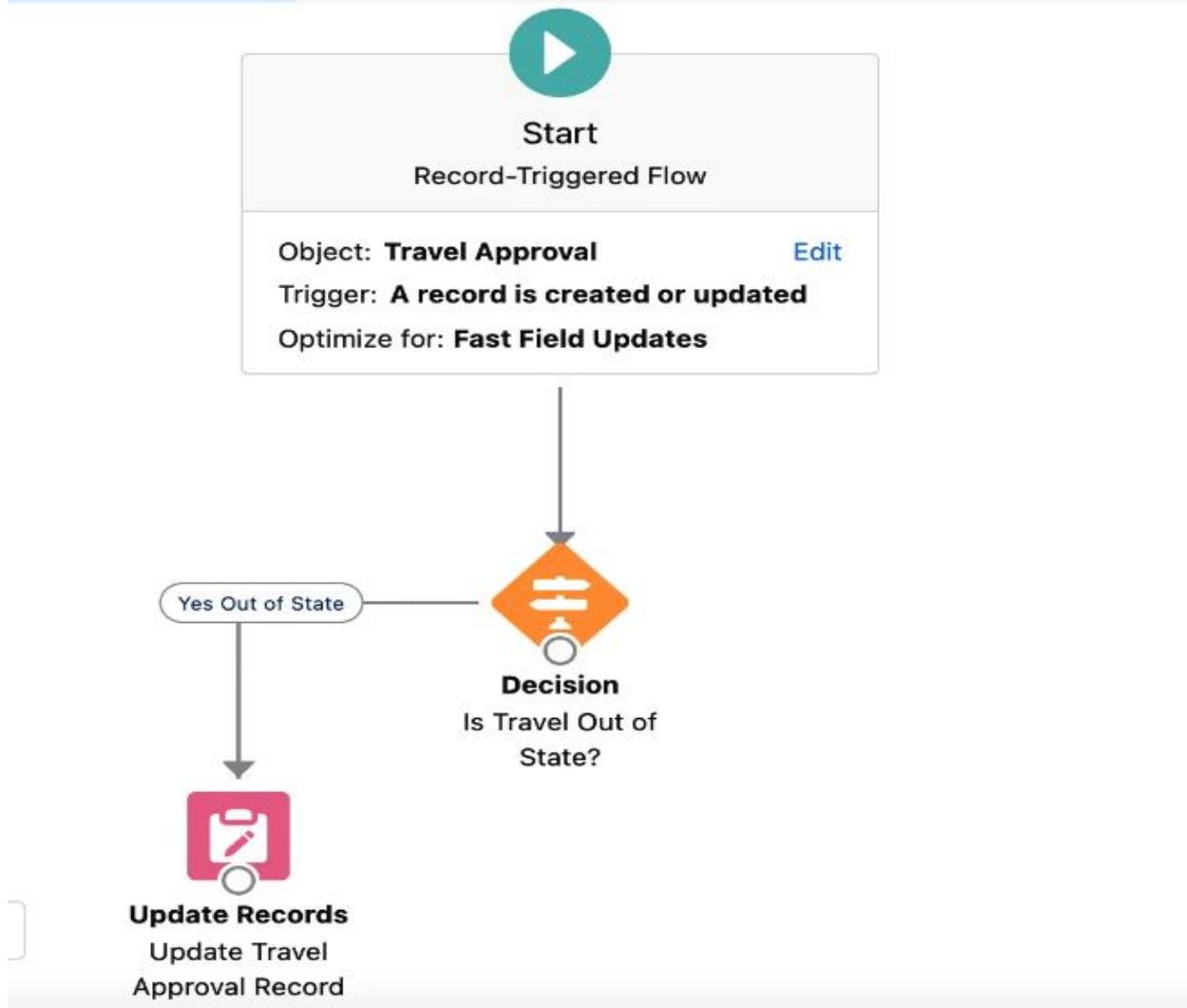
Create an Action for the Flow Using Update Records Elements:-

1. From the left-hand column, the flow toolbox, drag an **Update Records** element onto the flow screen.
2. Set the parameters for the element:-
 - Label = **Update Travel Approval Record**.
 - How to find Records = **Use the travel approval record that triggered the flow**.
 - Condition Requirements to Update the Record = **None—Always Update the Record**.
 - Field: **Out_of_State__c**
 - Value = **\$GlobalConstant.True** (start typing True and this value will come up).

The screenshot shows the 'Edit Update Records' configuration page. At the top, it says 'Edit Update Records'. Below that, the title is 'Update Travel Approval Record (Update_Travel_Approval_Record)'. There is a note: '* How to Find Records to Update and Set Their Values'. Three options are listed: 'Use the travel approval record that triggered the flow' (selected), 'Use the IDs and all field values from a record or record collection', and 'Specify conditions to identify records, and set fields individually'. A note below says: 'Because this flow runs before a record is saved, you can only update the record that triggered the flow to run. To update other records, configure the trigger to run the flow after the record is saved.' Under 'Set Filter Conditions', it says 'Condition Requirements to Update Record' with a dropdown menu showing 'None—Always Update Record'. Under 'Set Field Values for the Travel Approval Record', there is a table with one row: 'Field' 'Out_of_State__c' and 'Value' '\$True'. At the bottom right are 'Cancel' and 'Done' buttons.

3. Click **Done**.

4. Drag the white circle from the Decision Node to the Update Records you just created, and select the decision **Yes Out of State**.
5. Click **Done**.



Create an Action for the Flow Using Update Records Elements:-

1. From the left-hand column, the flow toolbox, drag an **Update Records** element onto the flow screen.
2. Set the parameters for the element:-
 - Label = **Update Travel Approval Record**.
 - How to find Records = **Use the travel approval record that triggered the flow**.

- Condition Requirements to Update the Record = **None—Always Update the Record.**
- Field: **Out_of_State__c**
- Value = **\$GlobalConstant.False** (start typing False and this value will come up).

Edit Update Records

Update Travel Approval Record (Update_Travel_Approval_Record_0) 

* How to Find Records to Update and Set Their Values

Use the travel approval record that triggered the flow
 Use the IDs and all field values from a record or record collection
 Specify conditions to identify records, and set fields individually

 Because this flow runs *before* a record is saved, you can only update the record that triggered the flow to run. To update other records, configure the trigger to run the flow *after* the record is saved.

Set Filter Conditions

Condition Requirements to Update Record

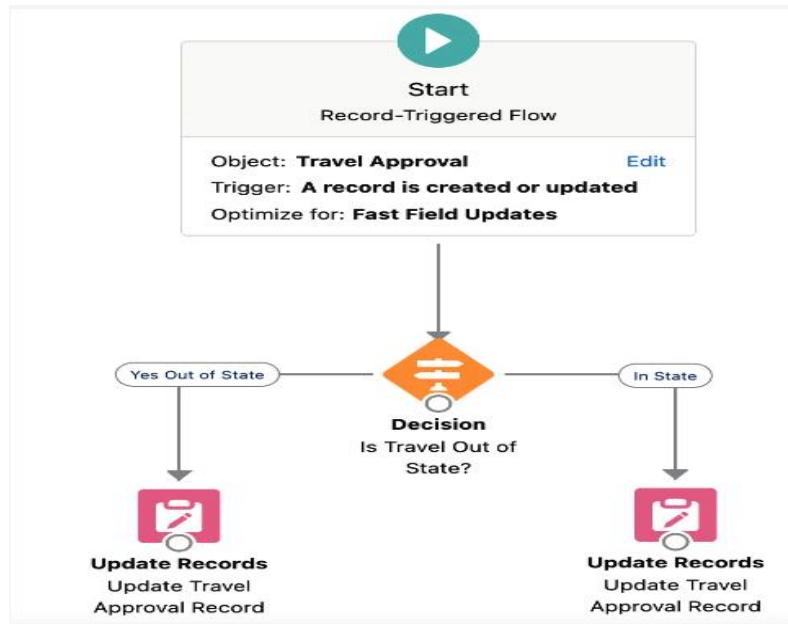
None—Always Update Record ▾

Set Field Values for the Travel Approval Record

Field	Value
Out_of_State__c	 False X 

Cancel Done

3. Click **Done**.
4. Drag the white circle from the Decision Node to the second Update Records you just created and select the decision **In State**.
5. Click **Done**.



Make sure to save and activate the flow:-

1. Click **Save**.
2. Flow Label: **Out of State Travel Flag**. Flow API Name will auto populate to **Out_of_State_Travel_Flag**.
3. Leave Description blank and advanced settings as is.
4. Click **Save**.
5. Click **Activate**.

The screenshot shows the Flow Builder interface with the following details:

- Flow Details:** Out of State Travel Flag - V1, Free-Form, Version 1: Active—Last modified 4 days ago, Run, Debug, View Tests, Deactivate, Save As, Save.
- Toolbox:** Elements Manager, Logic (5) includes Assignment, Decision, Loop, Collection Sort, Collection Filter; Data (2) includes Update Records, Get Records.
- Flow Structure:**
 - Start (Record-Triggered Flow)
 - Decision Diamond: Is Travel Out of State?
 - If Yes Out of State: Update Records (Update Travel Approval Record)
 - If In State: Update Records (Update Travel Approval Record)
- Note:** Activate Windows Go to Settings to activate Windows.

Travel Approval # TA-00001

Status: Draft

Owner: VEMPALLI MAHMAMD AFROZ

Department: Technology

Trip Info:

- Purpose of Trip: Attend Dreamforce
- Trip Start Date: 21/02/2023
- Trip End Date: 28/02/2023

Last Modified By: VEMPALLI MAHMAMD AFROZ, 21/02/2023, 3:04 pm

No activities to show.

No past activity. Past meetings and tasks marked as done show up here.

Activate Windows
Go to Settings to activate Windows.

Step 6: Create the Approval Processes:-

1. Click and select **Setup**.
2. Quick Find and search for Approval Processes.
3. Select **Approval Processes**.

PLATFORM TOOLS

- Apps
- Feature Settings
- Objects and Fields
- Process Automation
- Approval Processes**
- Flows
- Post Templates

Most Recently Used

10 items

NAME
Troy Hedges
Eric Executive

4. In the Manage Approval Processes For list, select **Travel Approval**.
5. Click **Create New Approval Process** and select **Use Jump Start Wizard**.

Manage Approval Processes

A listing of both active and inactive approval processes for **Travel Approvals** is displayed below. Set up your approval process in a few short steps. Or, select Use Standard Wizard to config...

Create New Approval Process ▾

[Use Jump Start Wizard](#)

[Use Standard Setup Wizard](#)

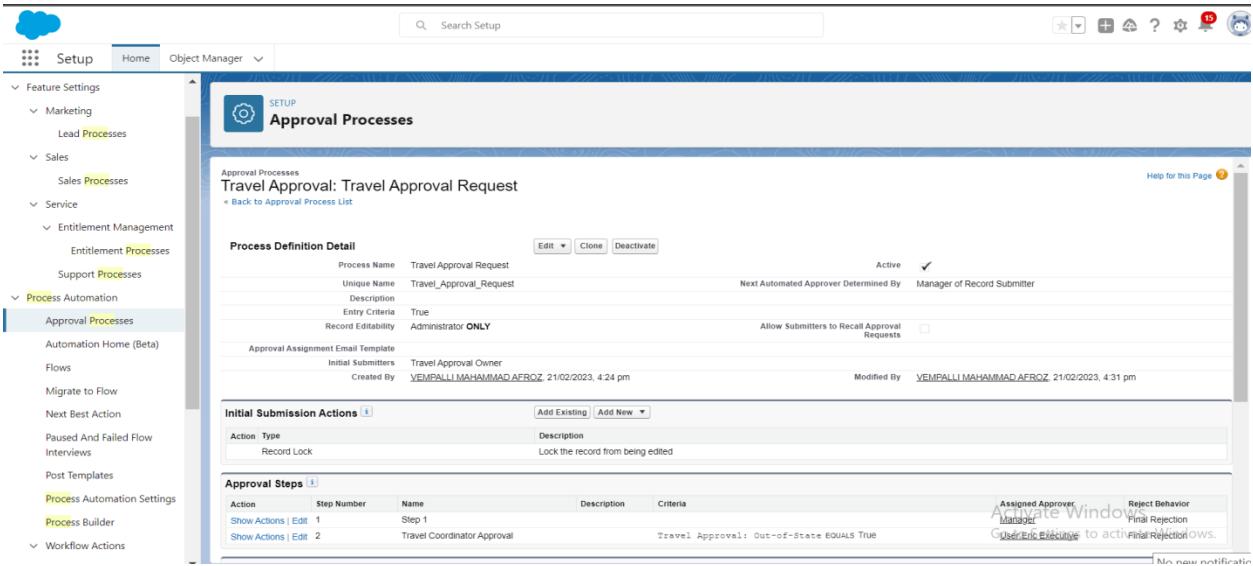
ACTIVE APPROVAL PROCESSES

No approval processes available

6. Approval Process Name: **Travel Approval Request** (Leave rest of the options as it is)
7. Specify Entry Criteria: Use this approval process if the following: **Formula Evaluates to True**.
8. In Formula space, Add “**True**”. There will be no Error.
9. Select Approver: Automatically assign an approver using a standard or custom hierarchy field: **Manager**
10. Click **Save**.
11. Click **View Approval Process Detail Page**.

Create an Approval Step for Out-of-State Travel.

1. Click New Approval Step.
2. Enter the following details:
 - Name: **Travel_Coordinator_Approval**
 - Step Number: **2**
 - Next
 - Enter this step if the following: **Criteria are met**
 - Formula criteria: **Field: Travel Approval: Out-of-State = True.**
 - Automatically assign to the approver(s): **User: select Eric Executive as the travel coordinator.**
 - **Save.**
 - **No, I will do this later.**



Create Final Approval action.

1. Click **Add New** in the Final Approval Actions area of the approval process form.
2. Select **Field Update** from the dropdown list.
3. Enter the following Values:-
 - Name: **Set Status to Approved**
 - Field to Update: **Status**
 - Picklist Options: **Select A specific value and select Approved from the dropdown list.**

Action	Type	Description
Edit	Record Lock	Unlock

Add New ▾

- Task
- Email Alert
- Field Update**
- Outbound Message

- **Save.**

Create Final Rejection action.

1. Click **Add New** in the Final Rejection Actions area of the approval process form.
2. Select **Field Update** from the dropdown list.
3. Enter the following Values:
 - Name: **Set Status to Rejected**
 - Field to Update: **Status**
 - Picklist Options: Select **A specific value** and select **Rejected** from the dropdown list.
4. **Save.**
5. **Activate.**

Test The Approval Process.

1. Create few Travel Approval records and Submit for Approval.
2. Login as Eric
3. Approve and reject the records randomly as shown in the below.

Record-2 is Approved by Eric.

Step Name	Date	Status	Assigned To
Travel Coordinator Approval	21/02/2023, 4:36 pm	Approved	Eric Executive
Step 1	21/02/2023, 4:36 pm	Approved	Eric Executive
Approval Request Submitted	21/02/2023, 4:34 pm	Submitted	VEMPALLI MAHMAD AFROZ

Record-3 is Rejected by Eric.

This screenshot shows a travel approval record for TA-00003. The status is listed as 'Rejected' under the 'Step Name' column. The record includes sections for Notes & Attachments, Expense Items, and Approval History. The Approval History table shows two steps: 'Step 1' was rejected, and 'Approval Request Submitted' was submitted. The right sidebar displays activity and Chatter feeds.

Step Name	Date	Status	Assigned To
Step 1	21/02/2023, 4:40 pm	Rejected	Eric Executive
Approval Request Submitted	21/02/2023, 4:40 pm	Submitted	VEMPALLI MAHMAD AFROZ

Approved Record Status Shows “Thumbs Up” in “Green Color”.

This screenshot shows a travel approval record for TA-00002. The status is listed as 'Approved' under the 'Status' field. The record includes sections for Related, Details, Activity, and Chatter. The Details section shows the owner as VEMPALLI MAHMAD AFROZ and the department as Technology. The Activity sidebar is empty.

Travel Approval #	Owner
TA-00002	VEMPALLI MAHMAD AFROZ

Rejected Record Status will Show the “Thumb Down” in “Red color”.

The screenshot shows a Salesforce page for a Travel Approval record (TA-00003). The 'Details' tab is selected. Key fields visible include:

- Travel Approval #: TA-00003
- Status: Rejected (indicated by a red thumbs-down icon)
- Out-of-State: checked
- Destination State: KA
- Total Expenses: ₹0.00
- Status Indicator: a red thumbs-down icon

The 'Activity' sidebar shows no upcoming or overdue activities. A banner at the bottom right says "Activate Windows Go to Settings to activate Windows."

We can see the Approval History by open the Related tab of that Record.

Approved Record:-

The screenshot shows a Salesforce page for a Travel Approval record (TA-00002). The 'Notes & Attachments' and 'Expense Items' tabs are empty. The 'Approval History' tab shows three steps:

Step Name	Date	Status	Assigned To
Travel Coordinator Approval	21/02/2023, 4:36 pm	Approved	Eric Executive
Step 1	21/02/2023, 4:36 pm	Approved	Eric Executive
Approval Request Submitted	21/02/2023, 4:34 pm	Submitted	VEMPALLI MAHMAD AFROZ

The 'Activity' sidebar shows no upcoming or overdue activities. A banner at the bottom right says "Activate Windows Go to Settings to activate Windows."

Rejected Record:-

The screenshot shows a Salesforce page for a 'Travel Approval' record with ID TA-00003. The record is in a 'Rejected' state. The page includes sections for 'Notes & Attachments', 'Expense Items', and 'Approval History'. The approval history table shows two steps: 'Step 1' (Rejected) and 'Approval Request Submitted' (Submitted). On the right side, there is an 'Activity' sidebar with no upcoming activities.

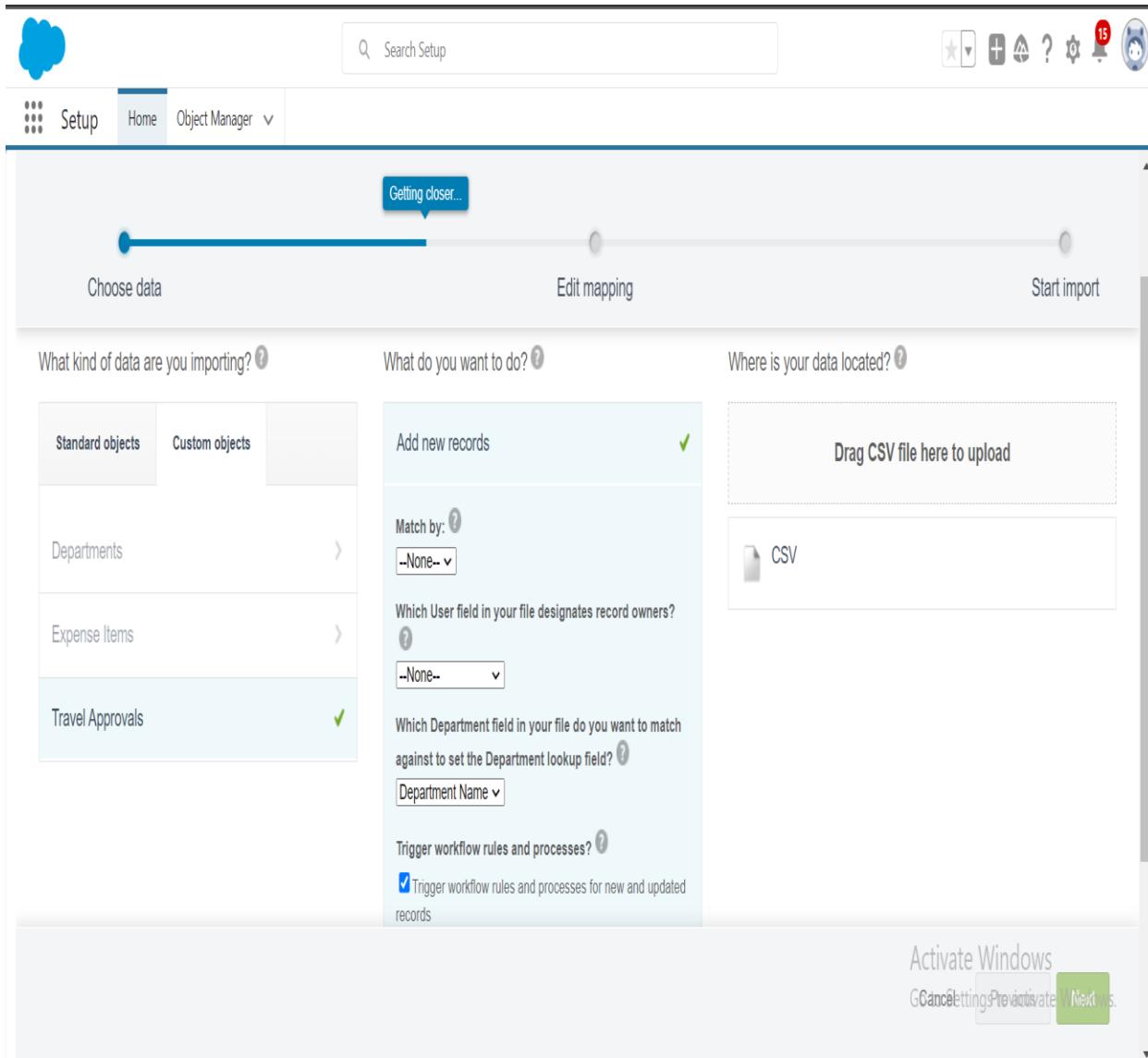
EXERCISE-2:-

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

1. From Setup, click the **Home** tab.
2. In the Quick Find box, enter Data Import and select **Data Import Wizard**.

The screenshot shows the 'Data Import Wizard' setup page. It features a 'Recent Import Jobs' section with a 'Bulk Api Monitoring' button. Below this is a 'Before you import your data...' section with tips for preparing data imports. A large 'Import your data in 3 easy steps!' section details the process: 'Pre-step: Prepare your data for import', 'Choose data to import', 'Edit field mapping', and 'Review and start import'. A prominent 'Launch Wizard!' button is at the bottom.

3. Select **Launch Wizard!**
4. Select the **Custom Objects** tab and select the **Travel Approval** object.
5. Select **Add New Records**.



6. Drag the **TravelApprovals.csv** file (from the folder you downloaded in the Build a Data Model for a Travel Approval App project) to the **Drag CSV file here to upload** section.
7. Select **Next**.
8. Ensure the names in the CSV file are the same as your object (these are automatically mapped).

9. Click **Next**.

10. Click **Start Import**.

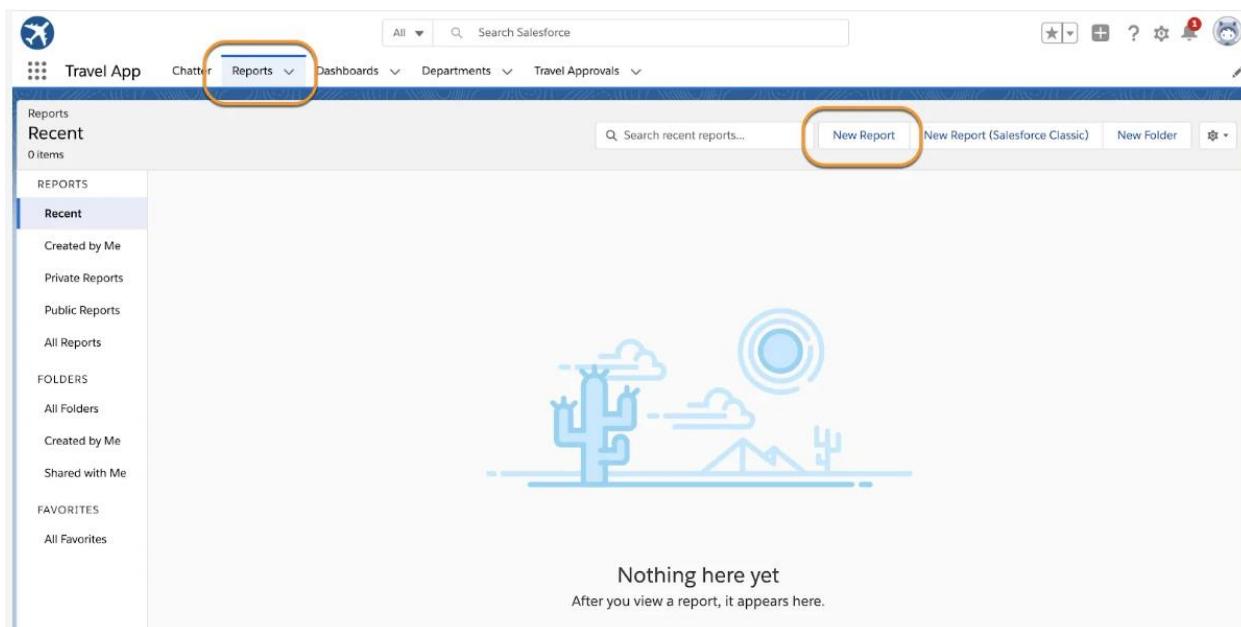
11. Click **OK**.

Make sure all the 301 Travel Approval records are successfully imported.

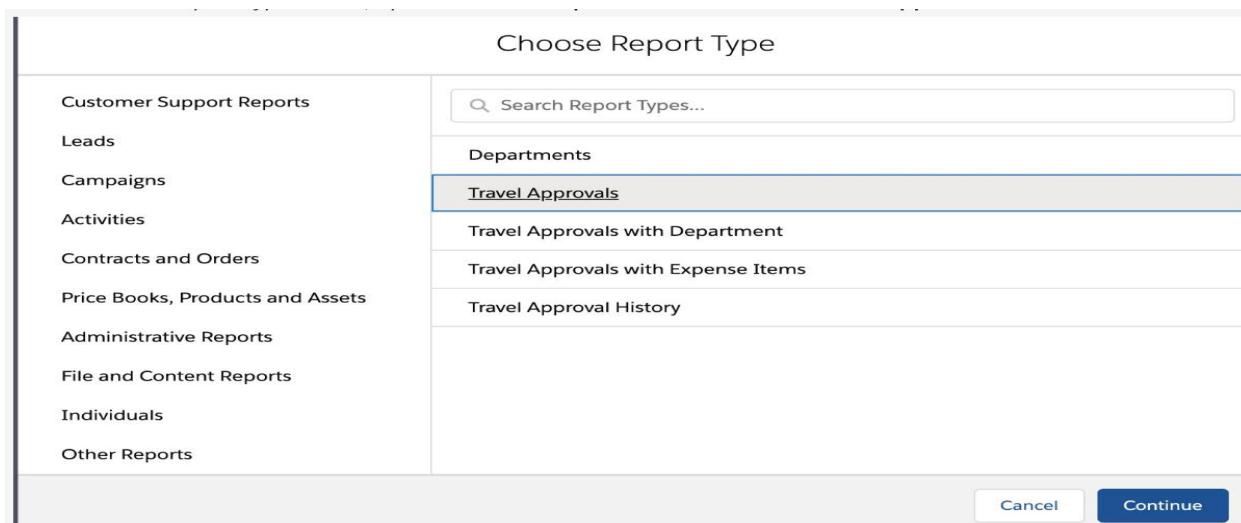
	Travel Approval #	Department	Created By	Status	Trip Start Date	Trip End Date	
289	IA-00360	Quality and Compliance Office	VEMPALLI MAHAMMAD AFROZ	Approved	13/03/2019	18/03/2019	
290	TA-00361	Division of Disability and Rehabilitative Services	VEMPALLI MAHAMMAD AFROZ	Approved	03/12/2019	15/12/2019	
291	TA-00362	Contract Management	VEMPALLI MAHAMMAD AFROZ	Approved	14/04/2019	05/05/2019	
292	TA-00363	Office of Communications and Media	VEMPALLI MAHAMMAD AFROZ	Approved	29/11/2019	29/11/2019	
293	TA-00364	Division of Disability and Rehabilitative Services	VEMPALLI MAHAMMAD AFROZ	Approved	15/03/2019	23/03/2019	
294	TA-00365	Legislative Services	VEMPALLI MAHAMMAD AFROZ	Approved	15/04/2019	24/04/2019	
295	TA-00366	Audit Services	VEMPALLI MAHAMMAD AFROZ	Rejected	07/04/2019	19/04/2019	
296	TA-00367	Division of Family Resources	VEMPALLI MAHAMMAD AFROZ	Approved	21/12/2019	27/12/2019	
297	TA-00368	Office of General Counsel	VEMPALLI MAHAMMAD AFROZ	Approved	01/04/2019	01/04/2019	
298	TA-00369	Office of General Counsel	VEMPALLI MAHAMMAD AFROZ	Approved	10/08/2019	19/08/2019	
299	TA-00370	Division of Finance	VEMPALLI MAHAMMAD AFROZ	Approved	30/03/2019	05/04/2019	
300	TA-00371	Contract Management	VEMPALLI MAHAMMAD AFROZ	Approved	17/06/2019	20/06/2019	
301	TA-00372	Office of General Counsel	VEMPALLI MAHAMMAD AFROZ	Approved	22/12/2019	28/12/2019	
302	TA-00373	Division of Mental Health and Addiction	VEMPALLI MAHAMMAD AFROZ	Approved	24/03/2019	25/03/2019	
303	TA-00374	Contract Management	VEMPALLI MAHAMMAD AFROZ	Approved	30/10/2019	02/11/2019	

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

1. Via the App Launcher, navigate to the **Travel App** and click the **Report tab**.



2. Click **New Report**.
3. In the Choose Report Type Screen, open the **Other** Reports folder and select **Travel Approvals**.

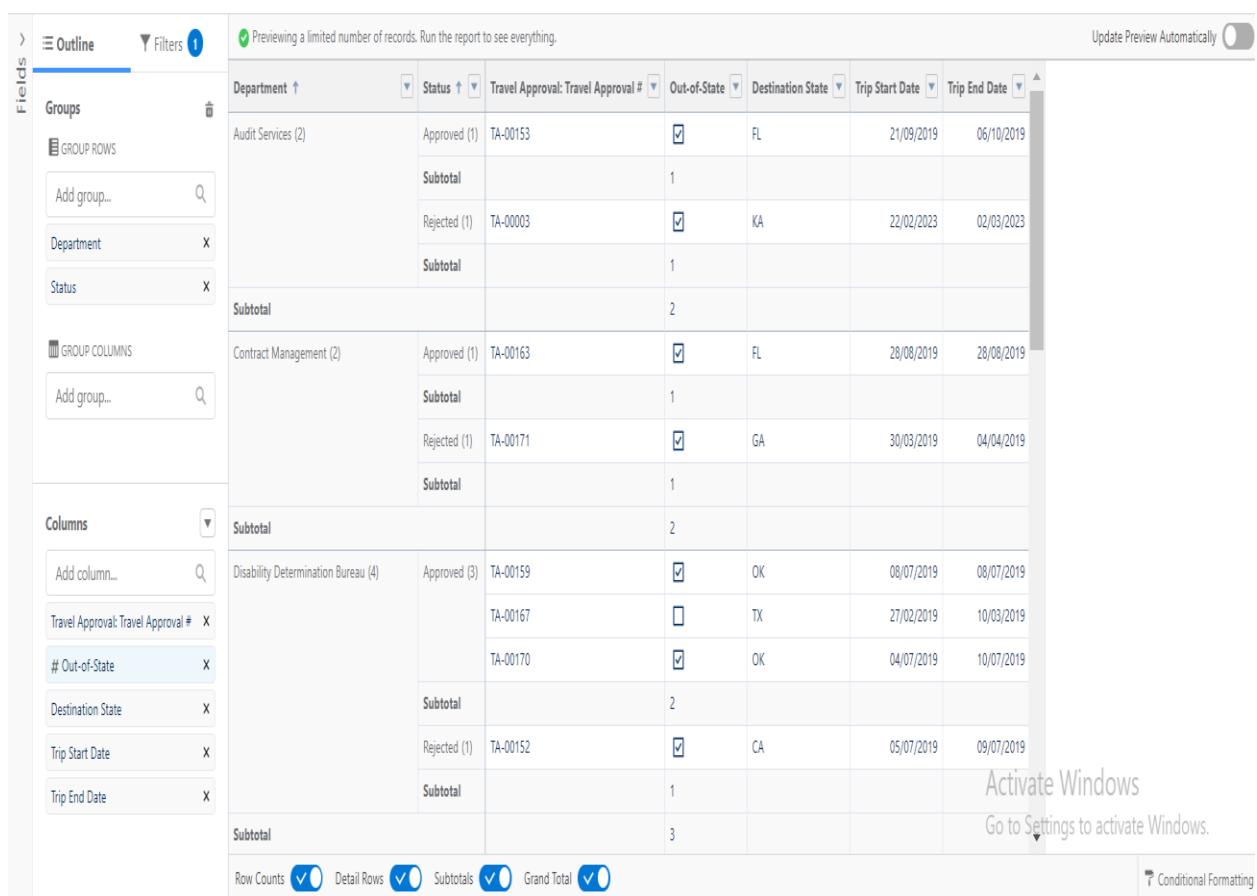


4. Click **Continue**.
5. Add column:-
 - Department

- Status
- Out-of-State
- Destination State
- Trip Start Date
- Trip End Date

6. Group Rows by This Field:-

- Department
- Status

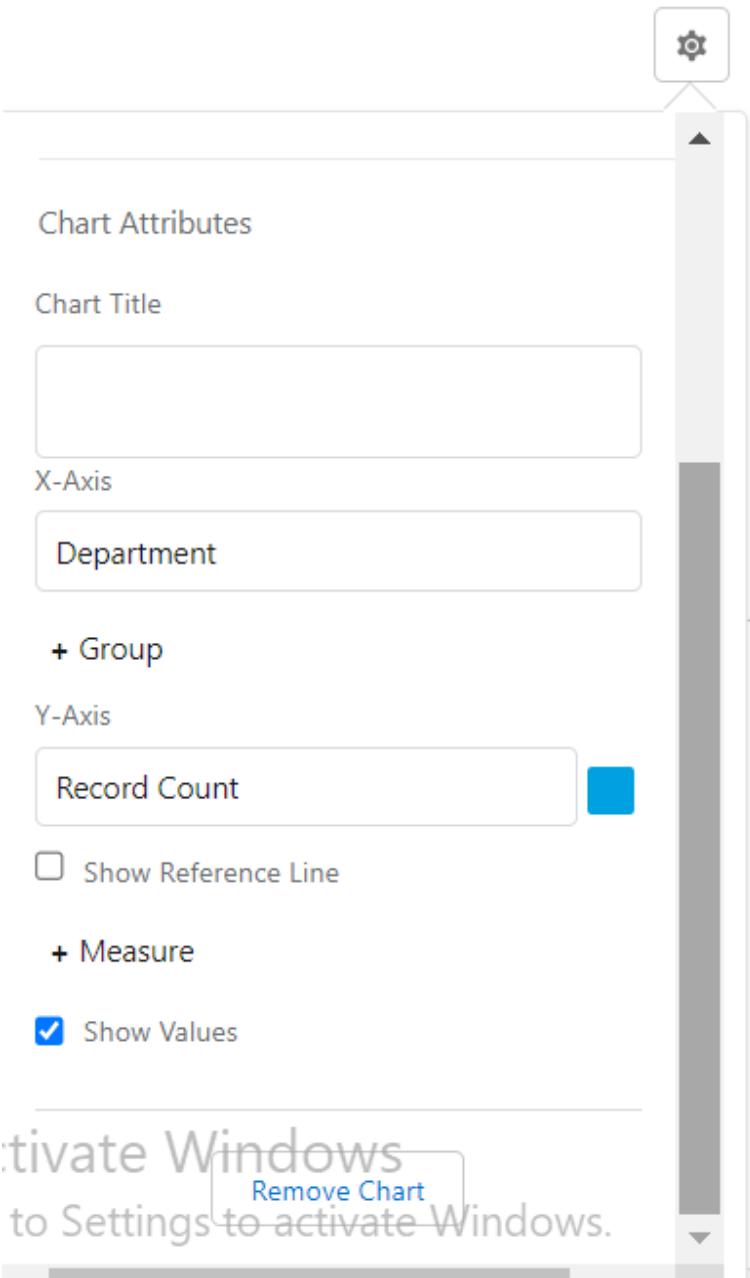


The screenshot shows a Power BI report interface with the following details:

- Fields pane (left):**
 - Groups:** Audit Services (2), Contract Management (2), Disability Determination Bureau (4).
 - Columns:** Status, Travel Approval: Travel Approval #, Out-of-State, Destination State, Trip Start Date, Trip End Date.
- Report Preview:** Shows data grouped by Department and Status. For Audit Services, there are two rows: Approved (1) TA-00153 and Rejected (1) TA-00003. For Contract Management, there are three rows: Approved (1) TA-00163, Rejected (1) TA-00171, and another Approved row. For Disability Determination Bureau, there are four rows: Approved (3) TA-00159, TA-00167, TA-00170, and Rejected (1) TA-00152.
- Report Footer:**
 - Activate Windows
 - Go to Settings to activate Windows.
 - Conditional Formatting

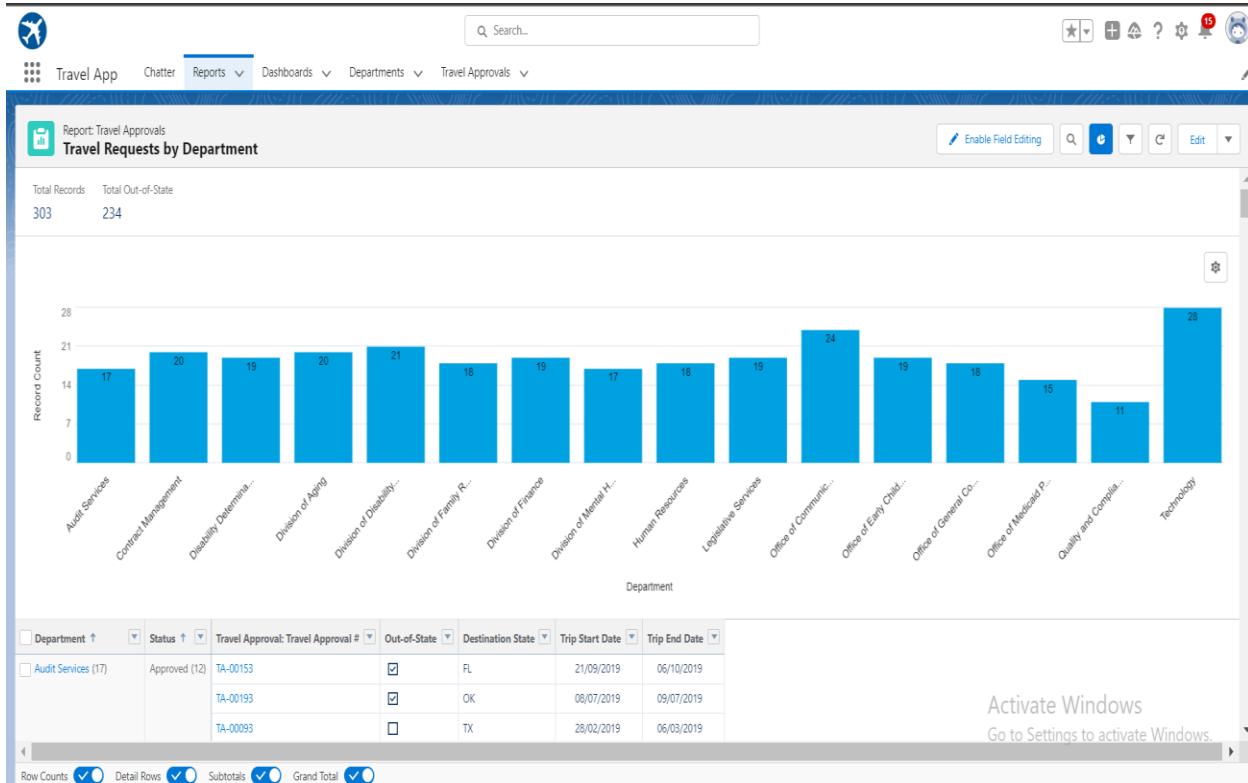
7. Add a Chart(Column):-

- X Axis = Department
- Y Axis = Record Count
- Show Values = Checked.



8. Click **Save** and set the following parameters:-
 - Report Name:- **Travel Requests by Department**
 - Report Description:- Leave as Blank
 - Report Folder:- **Public Reports**
9. Click **Save**.
10. Click **Run**.

Report should look like this:-



Step 3:- Create a Travel Requests by Month Report.

1. Via the App Launcher, navigate to the **Travel App** and click the **Report** tab.
2. Click **New Report**.
3. In the Choose Report Type Screen, open the **Other Reports** folder and select **Travel Approvals**.

The screenshot shows the "Choose Report Type" screen. On the left, there is a sidebar with various report categories:

- Customer Support Reports
- Leads
- Campaigns
- Activities
- Contracts and Orders
- Price Books, Products and Assets
- Administrative Reports
- File and Content Reports
- Individuals
- Other Reports

On the right, there is a search bar labeled "Search Report Types..." and a list of report types:

- Departments
- Travel Approvals** (highlighted)
- Travel Approvals with Department
- Travel Approvals with Expense Items
- Travel Approval History

At the bottom right are "Cancel" and "Continue" buttons.

4. Click **Continue**.

5. Add column:-

- Department
- Status
- Out-of-State
- Destination State
- Trip Start Date
- Trip End Date

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

The screenshot shows a Microsoft Power BI report interface. On the left, there's a sidebar with 'Fields' and sections for 'Groups' (with 'GROUP ROWS' and 'Add group...'), 'Trip End Date', 'Out-of-State', 'GROUP COLUMNS' (with 'Add group...'), 'Columns' (with 'Add column...'), 'Travel Approval: Travel Approval #', 'Department', 'Status', 'Destination State', 'Trip Start Date', and 'Subtotal'. The main area shows a table with columns: Trip End Date, Out-of-State, Travel Approval: Travel Approval #, Department, Status, Destination State, and Trip Start Date. Data rows are grouped by month: January 2019 (1), February 2019 (1), March 2019 (1), April 2019 (3), and May 2019 (2). A context menu is open over the March 2019 group, with 'Group Date By...' selected, showing options like Day, Calendar Week, Calendar Month (which is checked), Calendar Quarter, Calendar Year, and Fiscal Quarter. At the bottom right, there's a watermark: 'Activate Windows Go to Settings to activate Windows.'

7. Click **Save** and set the following parameters for the Report.

- Report Name:- **Travel Requests by Month**
- Report Description: - Leave a blank.
- Report Folder: - Public Reports.

8. Click **Save**.

9. Click **Run**.

Report should look like this:-

Report: Travel Approvals
Travel Requests by Month

Total Records
303

Trip End Date	Out-of-State	Travel Approval	Department	Status	Destination State	Trip Start Date
January 2019 (17)	(6)	TA-00164	Office of General Counsel	Rejected	TX	27/01/2019
		TA-00177	Disability Determination Bureau	Approved	TX	12/01/2019
		TA-00141	Legislative Services	Approved	TX	20/01/2019
		TA-00323	Quality and Compliance Office	Approved	TX	14/01/2019
		TA-00332	Contract Management	Approved	TX	18/01/2019
		TA-00311	Division of Aging	Approved	TX	22/01/2019
	Subtotal					
	(11)	TA-00179	Division of Aging	Approved	FL	03/01/2019
		TA-00189	Division of Mental Health and Addiction	Approved	OK	03/01/2019
		TA-00190	Contract Management	Approved	GA	30/01/2019
		TA-00097	Office of Communications and Media	Rejected	FL	12/01/2019
		TA-00104	Division of Disability and Rehabilitative Services	Approved	OK	23/01/2019
		TA-00119	Division of Aging	Approved	OK	24/01/2019
		TA-00234	Division of Disability and Rehabilitative Services	Approved	FL	29/01/2019
		TA-00270	Contract Management	Approved	FL	29/01/2019
		TA-00283	Office of Early Childhood and Out-of-School Learning	Approved	GA	18/01/2019
		TA-00285	Division of Family Resources	Approved	FL	21/01/2019

Row Counts Detail Rows Subtotals Grand Total

Activate Windows
Go to Settings to activate Windows.

Reports that what we have made:-

Reports
Public Reports
3 items

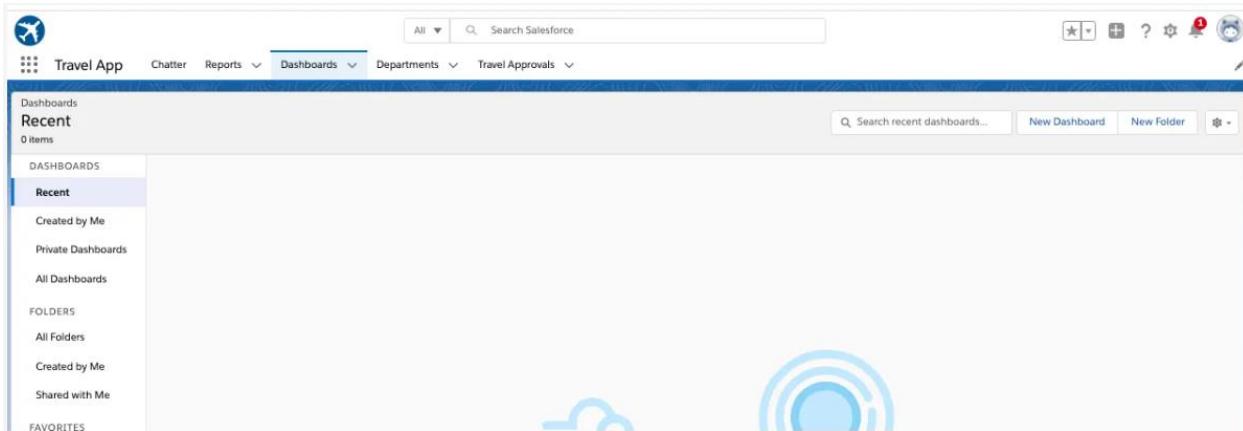
REPORTS	Report Name	Description	Folder	Created By	Created On	Subscribed
Recent	Sample Flow Report: Screen Flows	Which flows run, what's the status of each interview, and how long do users take to complete the screens?	Public Reports	Automated Process	21/2/2023, 2:24 pm	
Created by Me	Travel Requests by Department		Public Reports	VEMPALLI MAHMAMD AFROZ	21/2/2023, 10:08 pm	
Private Reports	Travel Requests by Month		Public Reports	VEMPALLI MAHMAMD AFROZ	21/2/2023, 10:16 pm	

REPORTS
Recent
Created by Me
Private Reports
Public Reports
All Reports
FOLDERS
All Folders
Created by Me
Shared with Me
FAVORITES
All Favorites

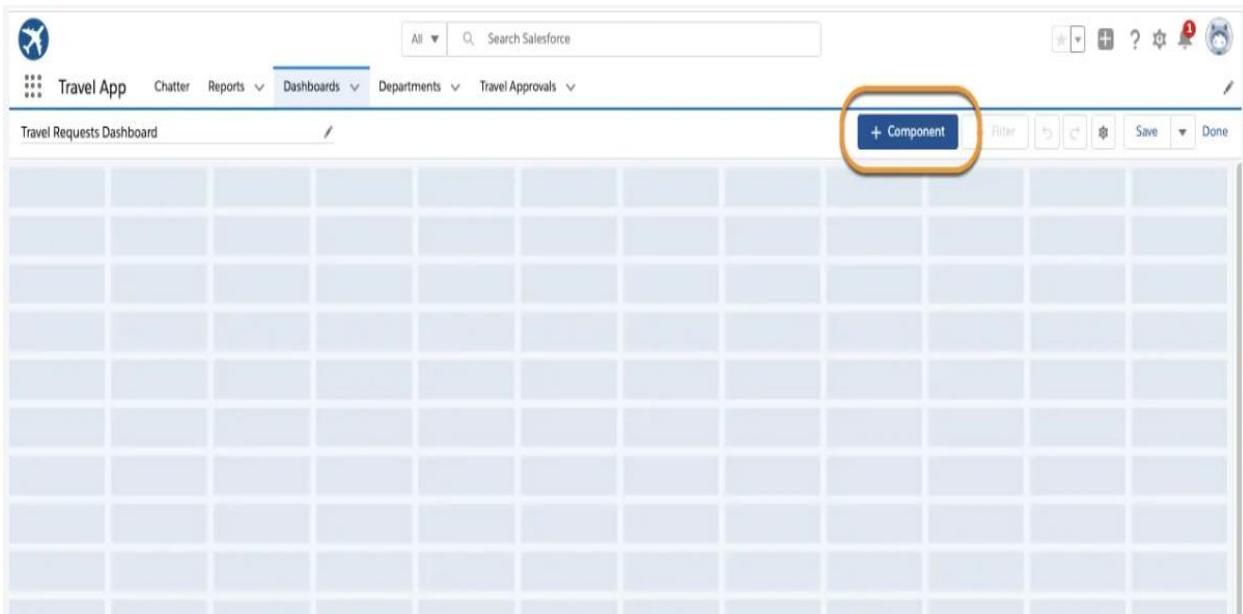
Activate Windows
Go to Settings to activate Windows.

Step 4:- Create a Travel Approvals Dashboard.

1. Click the **Dashboards** tab.
2. Click **New Dashboard**.

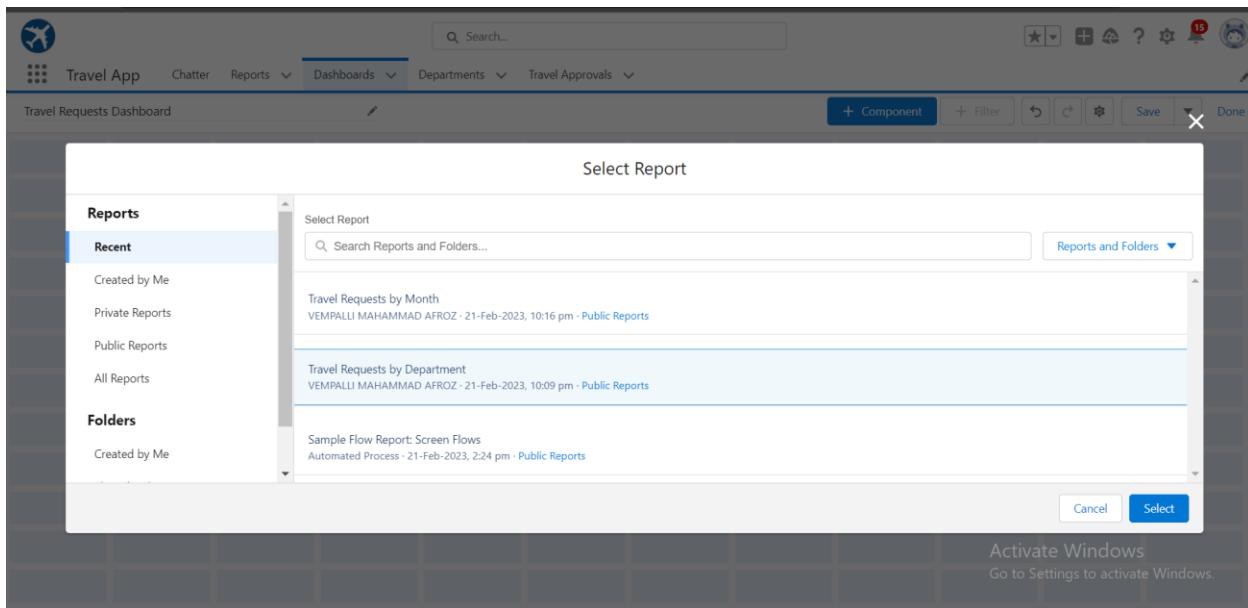


3. Enter the following values:-
 - Name:- **Travel Requests Dashboard**
 - Description:- **Leave as blank**
 - Folder:- **Private Folder**
4. Click **Create**.

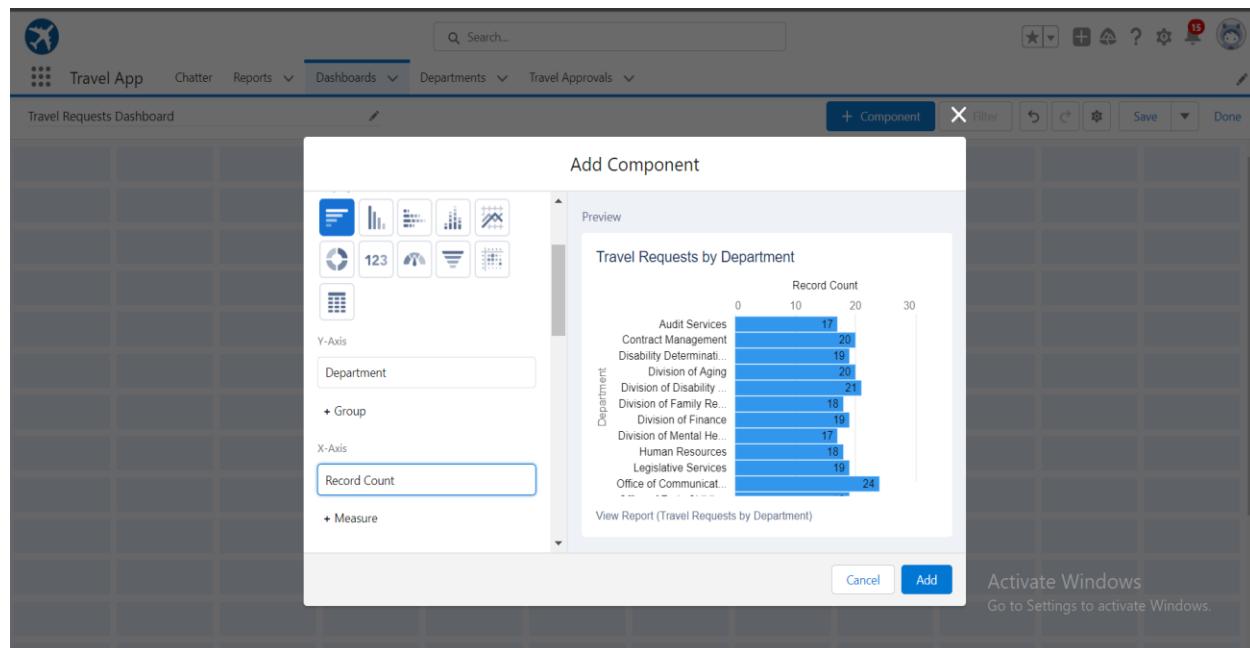


5. Click on **+ Component**.

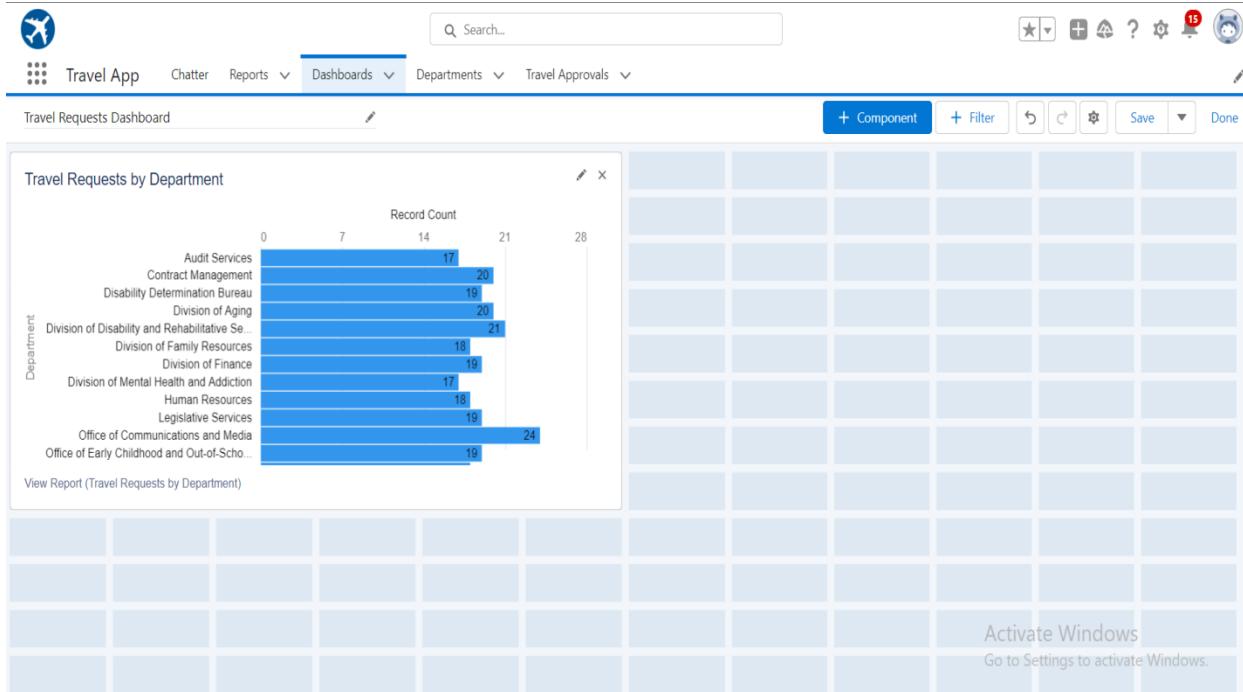
6. Click the **Travel Request by Department** report and click **Select**. You are presented with options to configure this component for the dashboard.



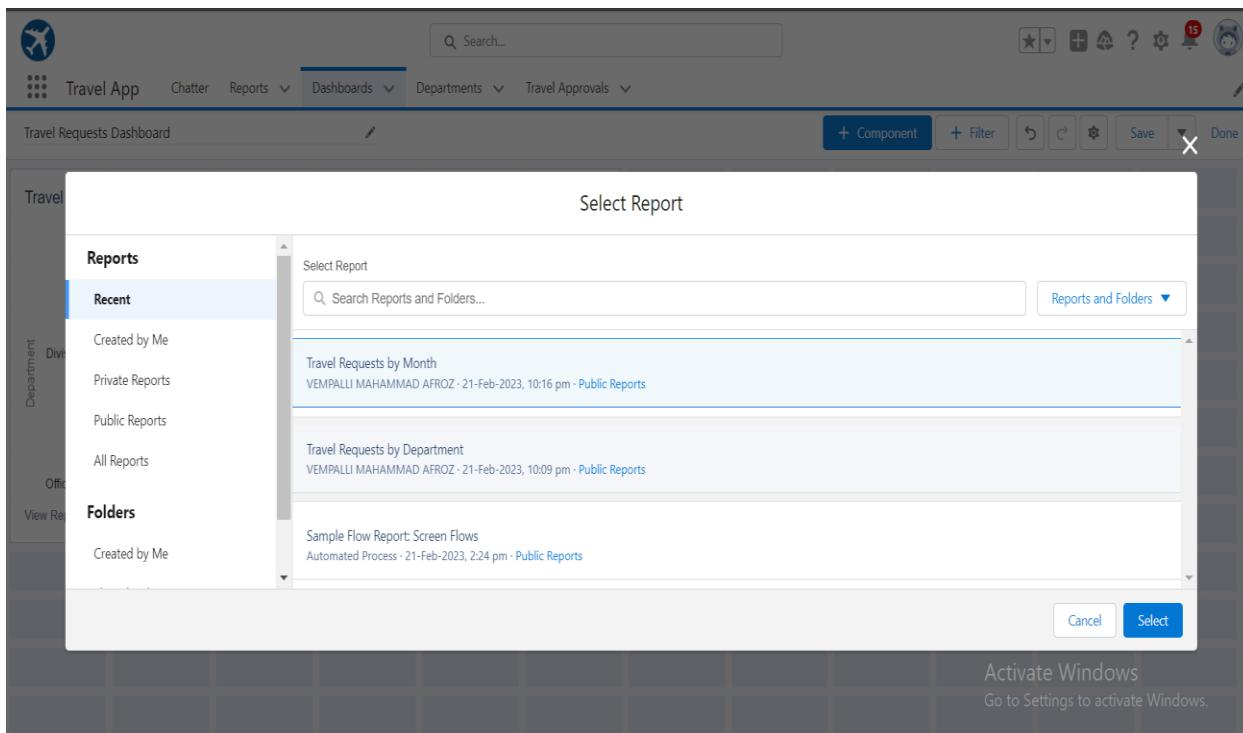
7. Make sure the Y-Axis is set to **Department** and the X-Axis is set to **Record Count**.



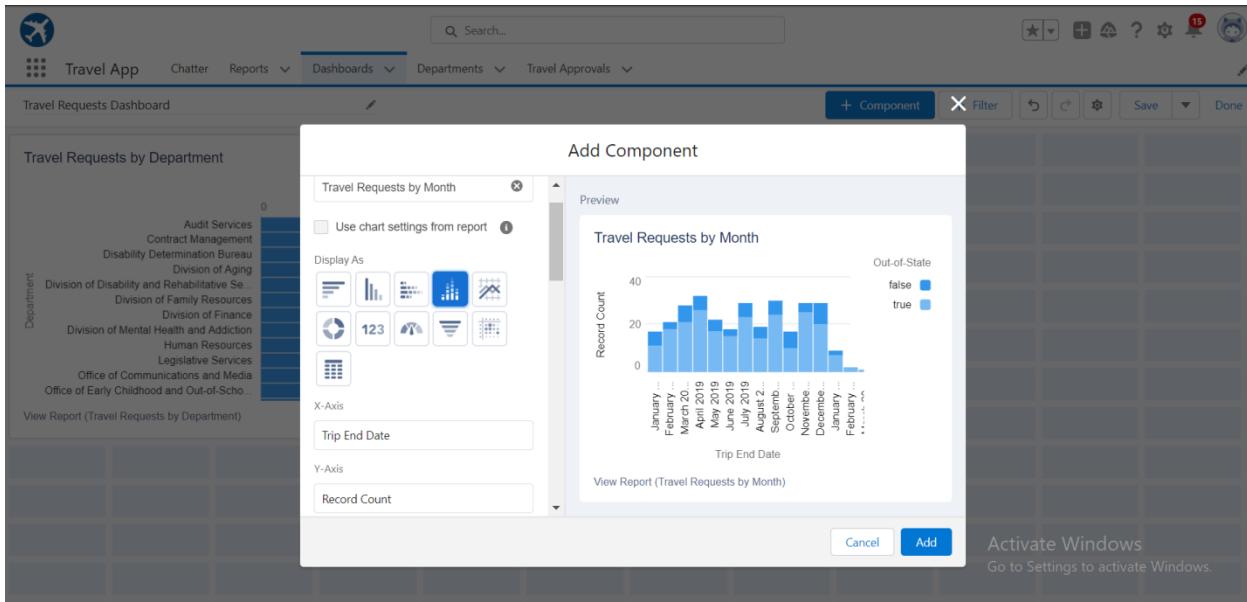
8. Click **Add**.



9. Click + Component to add our other report.
10. Select the **Travel Requests by Month** report.

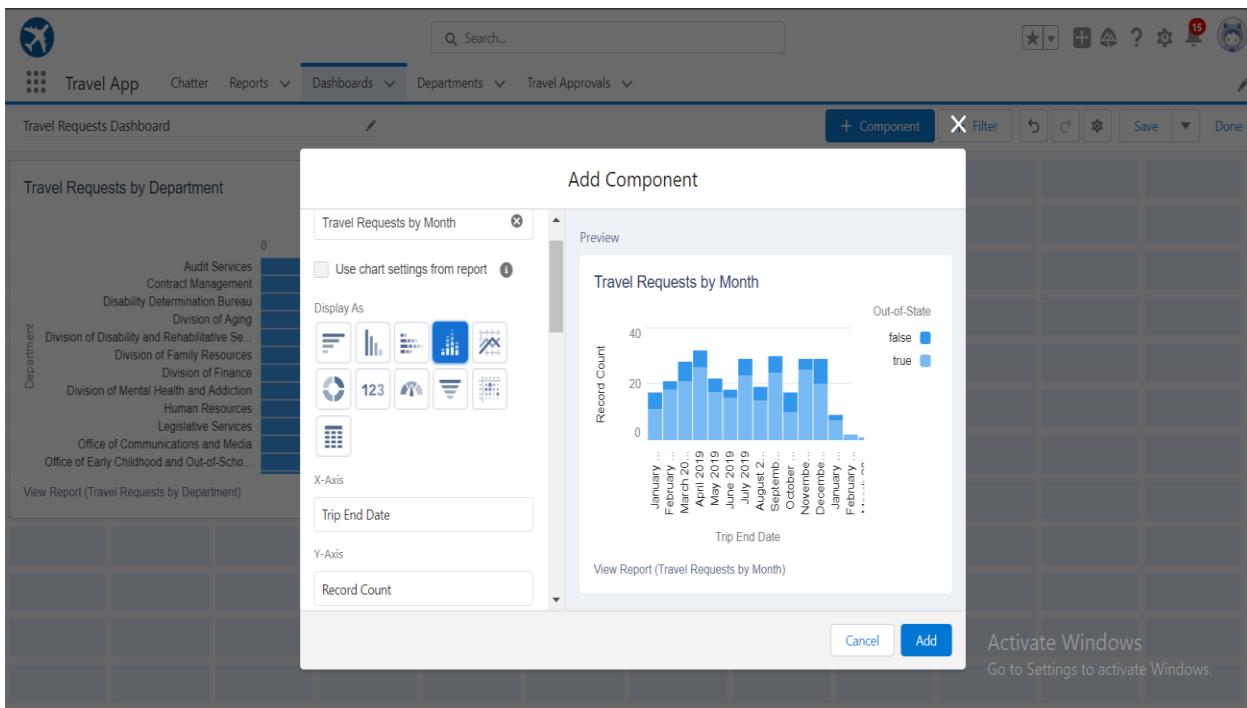


11.Click Select.



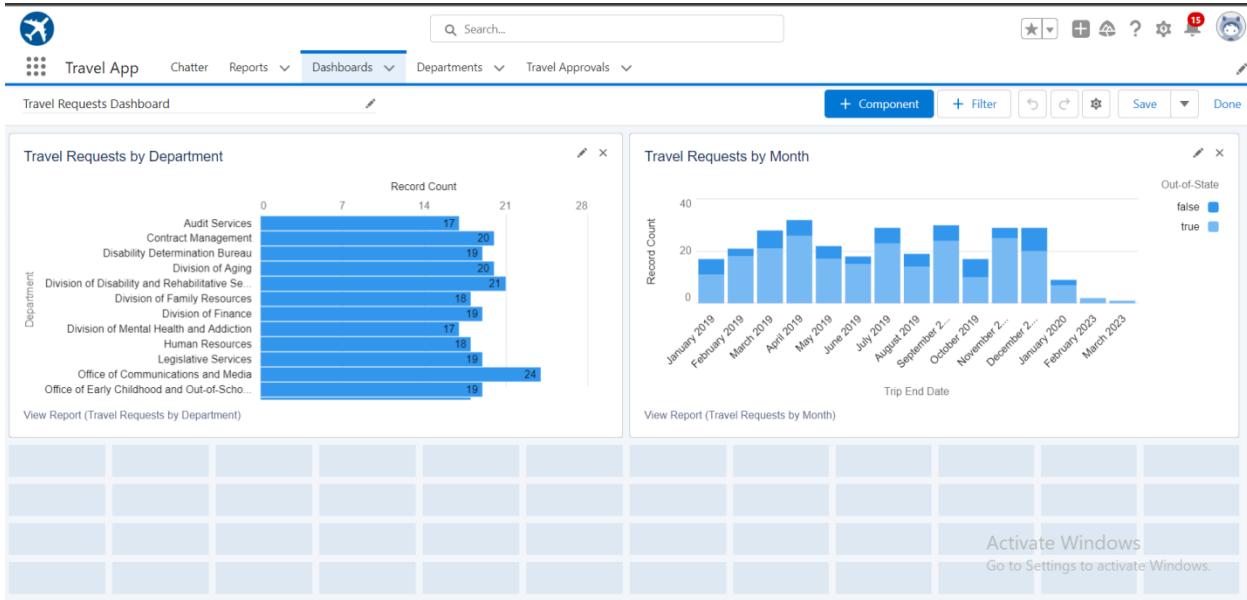
12.Display As | Stacked Vertical Bar Chart.

13.Sort By | Trip End Date.

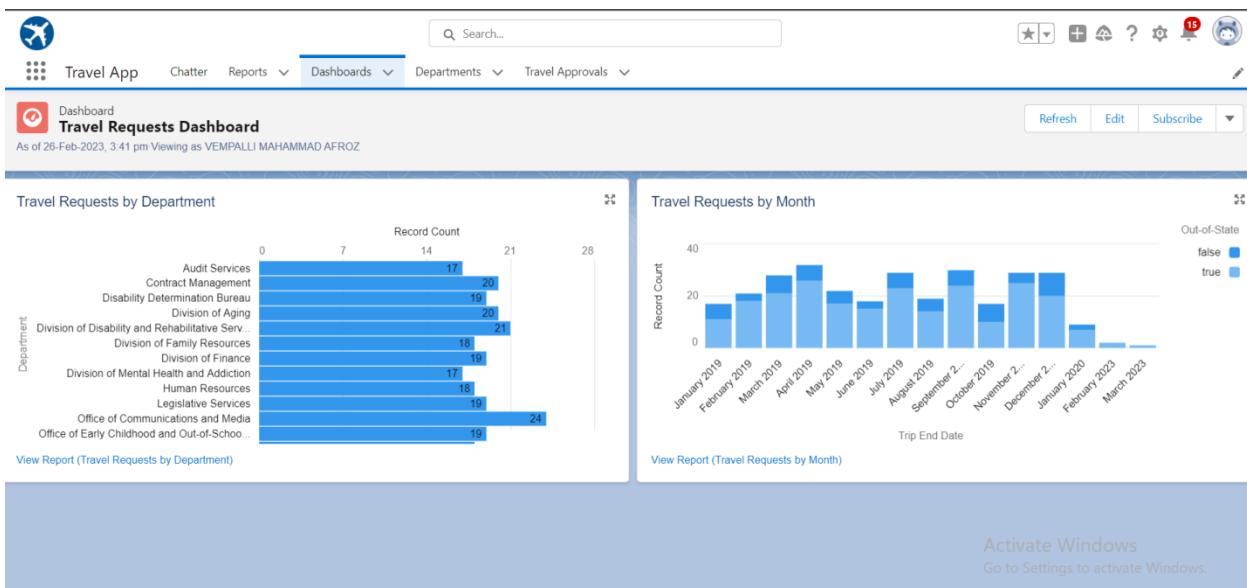


14.Click Add.

15.Drag the chart on the bottom, and position it to the right of the first chart we added.



16.Click Save and then the Done. Your dashboard will look like the following.

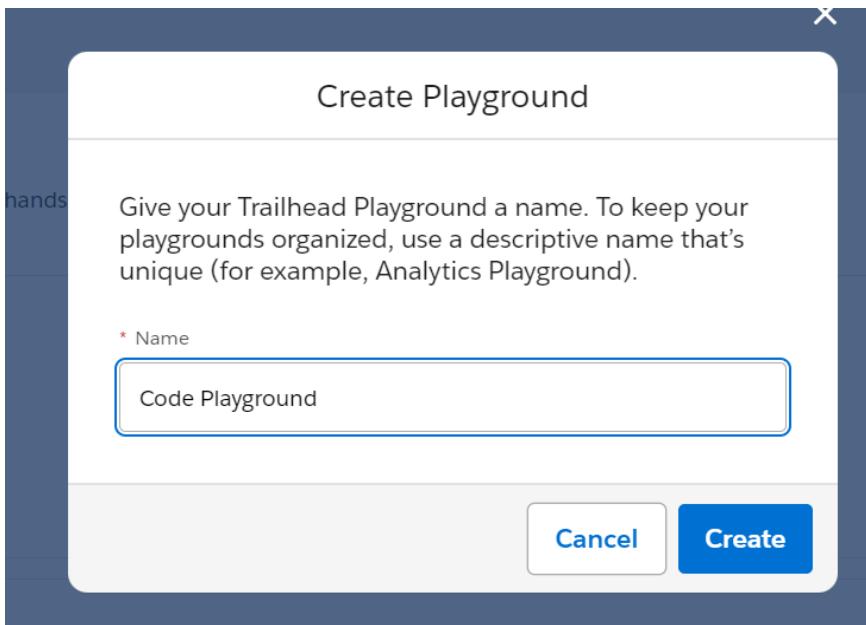


Module:-3

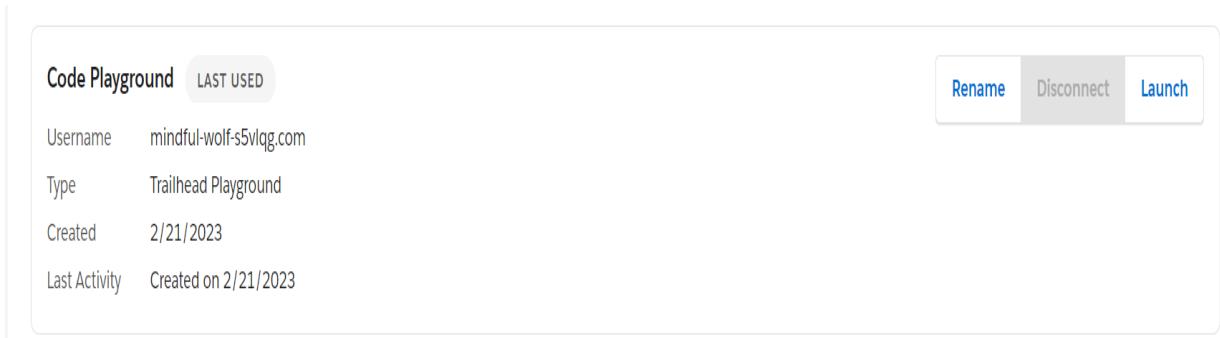
Prerequisites:-

Create a New Trailhead Playground:-

1. For this project, you need to create a new Trailhead Playground. Named it as “Code Playground”.



2. Click “Create”. It typically takes 3-4 minutes to create a new Trailhead Playground.

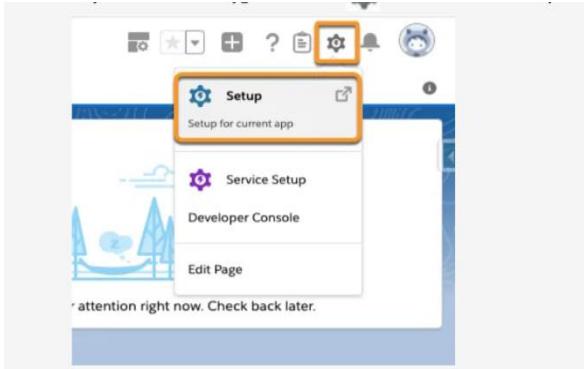


3. Once Created Open it by clicking “Launch”.

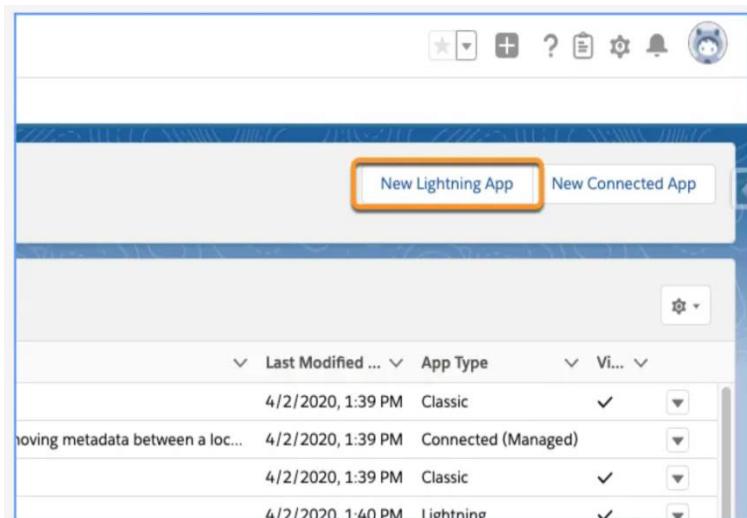
EXERCISE-1:-

Create a new custom lightning App.

1. In your Trailhead Playground, click  and then select **Setup**.



2. In the Quick Find box, enter App Manager and select **App Manager**.
3. Click **New Lightning App**.



4. In the App Details & Branding window, enter these details.
5. For App Name, enter “**Code Playground**”.
6. For the Image, click **Upload** and select **image in .png format**.
7. Click **Next**.
8. On the App Options screen, select **Standard navigation** and click **Next**.
9. On the Utility Items screen, click **Next**.

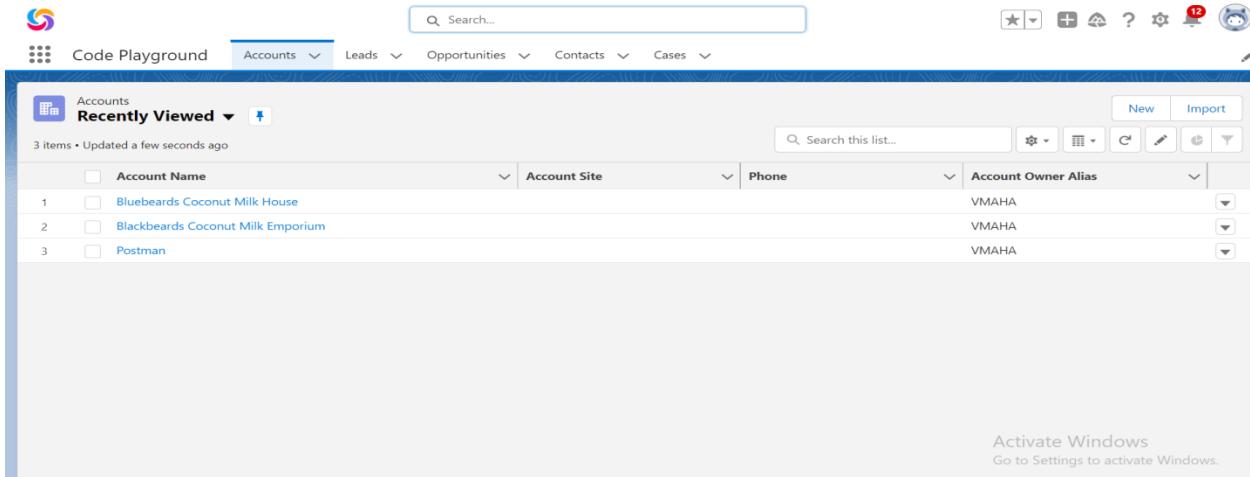
10. Add Lead, Account, Contact, Opportunity, Case to the Navigation items.

The screenshot shows the 'Navigation Items' configuration screen. On the left, a sidebar lists 'App Settings' sections: App Details & Branding, App Options, Utility Items (Desktop Only), and Navigation Items (which is selected). Under 'Navigation Items', there's a 'User Profiles' section. The main area is titled 'Navigation Items' and contains two panels: 'Available Items' and 'Selected Items'. The 'Available Items' panel lists various navigation items with icons: Alert Settings, All Sites, Alternative Payment Methods, App Launcher, Approval Requests, and Asset Action Sources. The 'Selected Items' panel lists five items: Leads, Accounts, Contacts, Opportunities, and Cases. Below these panels are 'Create' and 'Cancel' buttons, and a message: 'Activate Windows Go to Settings to activate Windows.' A 'Save' button is located at the bottom right of the configuration area.

11. On the User Profiles screen, select System Administrator and add it to Selected Profiles, then click Save & Finish.

The screenshot shows the 'User Profiles' configuration screen. On the left, a sidebar lists 'App Settings' sections: App Details & Branding, App Options, Utility Items (Desktop Only), Navigation Items, and User Profiles (which is selected). The main area is titled 'User Profiles' and contains two panels: 'Available Profiles' and 'Selected Profiles'. The 'Available Profiles' panel lists several user profiles: Analytics Cloud Integration User, Analytics Cloud Security User, Authenticated Website, Contract Manager, Cross Org Data Proxy User, Custom: Marketing Profile, and Custom: Sales Profile. The 'Selected Profiles' panel contains one profile: System Administrator. Below these panels are 'Activate Windows' and 'Save' buttons, with the message: 'Go to Settings to activate Windows.'

12. Click , then search for Code and select a “Code Playground”. And open it.



The screenshot shows the Salesforce Code Playground interface. At the top, there's a navigation bar with links for Accounts, Leads, Opportunities, Contacts, and Cases. Below the navigation is a search bar labeled "Search...". The main area displays a list titled "Recently Viewed" under the "Accounts" category. The list contains three items, each with a checkbox, account name, account site, phone number, and account owner alias. The items are:

	Account Name	Account Site	Phone	Account Owner Alias
1	Bluebeards Coconut Milk House			VMAHA
2	Blackbeards Coconut Milk Emporium			VMAHA
3	Postman			VMAHA

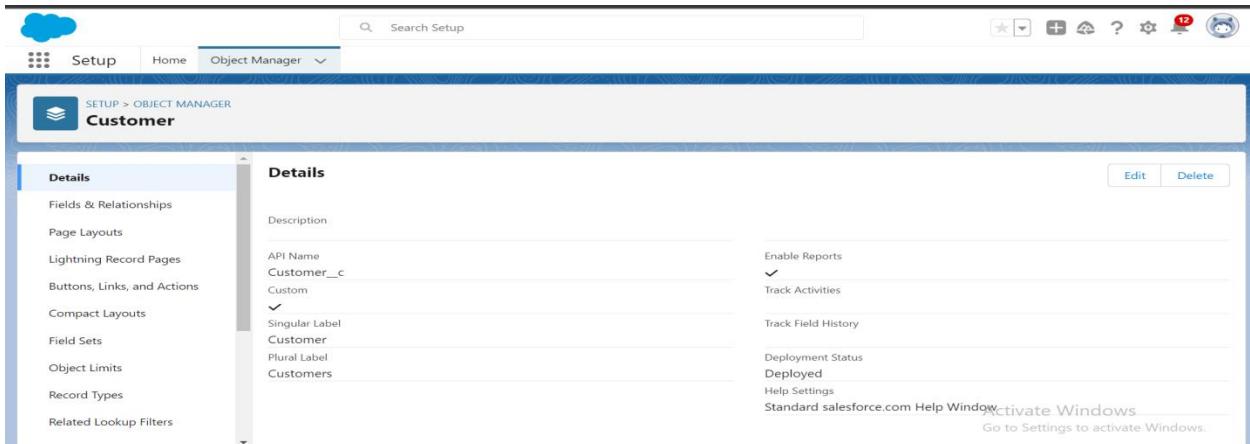
At the bottom right of the interface, there's a message: "Activate Windows Go to Settings to activate Windows."

Create a Custom ‘Customer’ Object:-

1. From Setup, click **Object Manager**.
2. Click Create, and then select **Custom Object**.
3. Enter These Details:-

 - Label = “Customer”
 - Plural = “Customers”
 - Record Name = **Customer Name**
 - Data Type = “Text”
 - Select “Allow Reports”, “Allow Search”.

4. Create a New Custom Tab of your choice.



The screenshot shows the Salesforce Setup - Object Manager interface. At the top, there's a navigation bar with links for Setup, Home, and Object Manager. Below the navigation is a search bar labeled "Search Setup". The main area displays a list titled "Customer" under the "SETUP > OBJECT MANAGER" section. On the left, there's a sidebar with various options like Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main panel shows the "Details" tab for the "Customer" object. It includes fields for API Name (Customer__c), Singular Label (Customer), and Plural Label (Customers). On the right, there are checkboxes for "Enable Reports" (checked) and "Track Activities". Below these are "Track Field History", "Deployment Status (Deployed)", "Help Settings", and a link to "Standard salesforce.com Help Window". At the bottom right, there's a message: "Activate Windows Go to Settings to activate Windows."

Create a Custom Fields for ‘Customer’ Object.

1. Label = **Description**, **Text Area**, Save.
2. Label = **Customer Type**, **Picklist**, Values: **Premium, Standard**. Save.
3. Label = **Active**, **Checkbox**, Save.

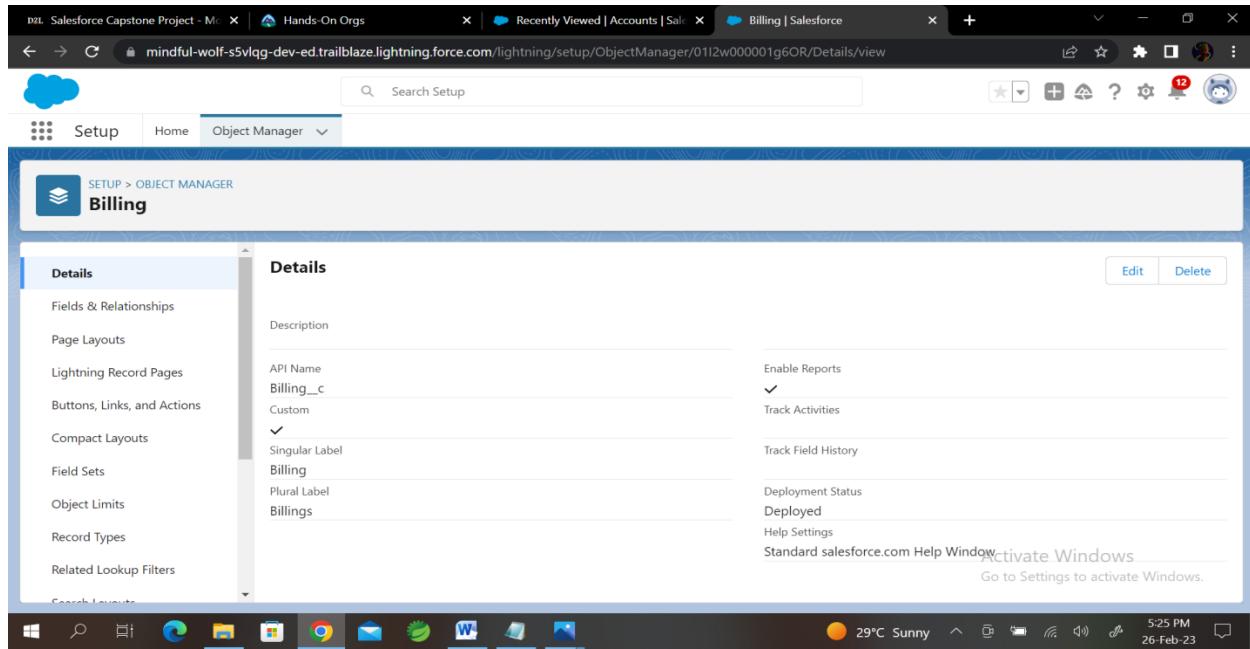
The screenshot shows the Salesforce Object Manager interface for the 'Customer' object. The left sidebar lists various layout-related options like Page Layouts, Lightning Record Pages, etc. The main area is titled 'Fields & Relationships' and displays seven items. The table has columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data is as follows:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Active	Active_c	Checkbox		
Created By	CreatedById	Lookup(User)		
Customer Name	Name	Text(80)		✓
Customer Type	Customer_Type_c	Picklist		
Description	Description_c	Text Area(255)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		

At the bottom right of the table, there are links: 'Activate Windows' and 'Go to Settings to activate Windows.'

Create a Custom ‘Billing’ Object:-

1. From Setup, click **Object Manager**.
2. Click Create, and then select **Custom Object**.
3. Enter These Details:-
4. Label = “**Billing**”
5. Plural = “**Billings**”
6. Record Name = **Bill Number**
7. Data Type = “**Auto Number**”
8. Format = B - {0000}
9. Starting number = 1
10. Select “**Allow Reports**”, “**Allow Search**”.
11. Create a New Custom Tab of your choice.
12. Add it to “**Code Playground**” App only, Save.



Create a Custom Fields for ‘Billing’ Object.

1. Label = **Amount Paid, Currency**, Save.
2. Label = **Status, Picklist**, Values: **Paid, Unpaid**. Save.
3. Label = **Customer Type, Picklist**, Values: **Premium, Standard**. Save.
4. Label = **Customer, Master-Detail**, Related To – **Customer** custom object. Save.

Fields & Relationships				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount Paid	Amount_Paid__c	Currency(18, 0)		
Billing Number	Name	Auto Number		
Created By	CreatedBy	Lookup(User)		
Customer	Customer__c	Master-Detail(Customer)		
Customer Type	Customer_Type__c	Picklist		
Last Modified By	LastModifiedBy	Lookup(User)		
Status	Status__c	Picklist		

Code Playground App should look like this:

The screenshot shows a CRM application interface with the following elements:

- Header:** A logo icon, a search bar with placeholder text "Search...", and a toolbar with various icons (star, plus, question mark, gear, bell, user profile).
- Breadcrumb:** "Code Playground" followed by a dropdown menu for "Accounts".
- Navigation:** A horizontal menu with dropdowns for "Leads", "Contacts", "Opportunities", "Cases", "Customers", and "Billings".
- Section Header:** "Accounts" and "Recently Viewed" with a refresh icon.
- Search Bar:** "Search this list..." with filter and sort icons.
- Table Headers:** "Account Name", "Account Site", "Phone", and "Account Owner Alias".
- Data Row:** One item listed: "Burlington Textiles Corp of America" with phone "(336) 222-7000" and owner alias "VMAHA".
- Activation Message:** "Activate Windows" and "Go to Settings to activate Windows." at the bottom right.

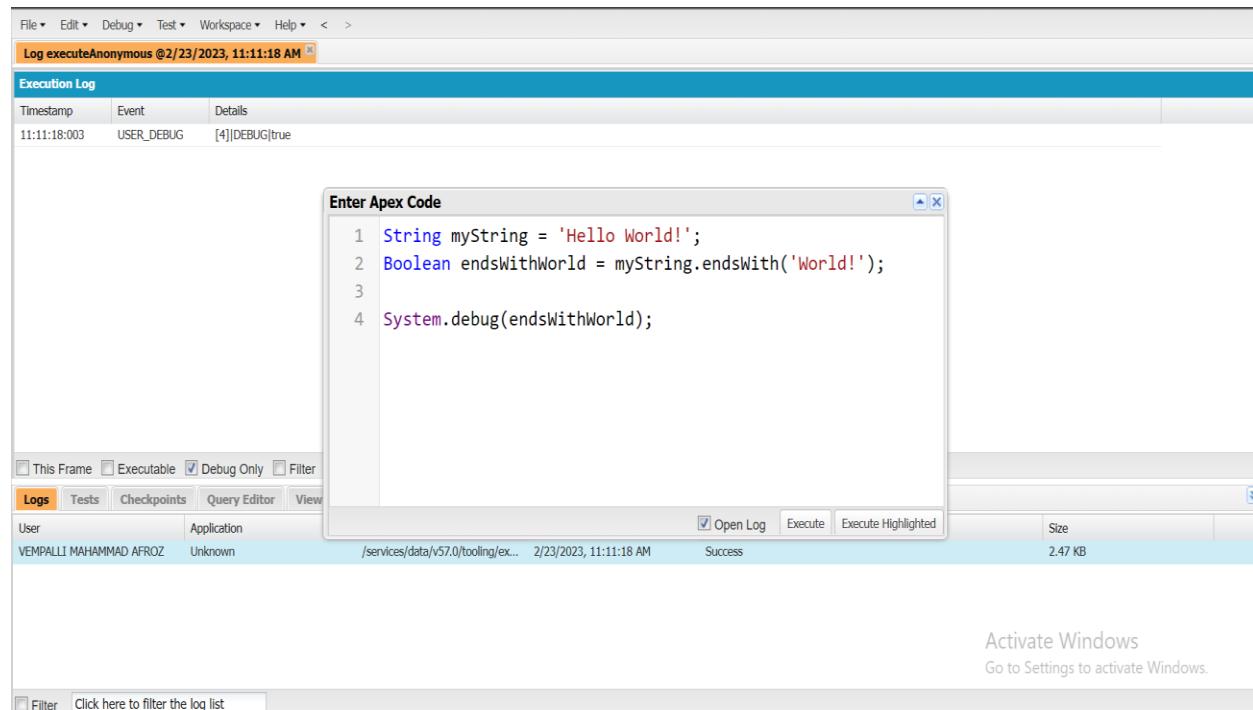
EXERCISE-2:- Use Execute Anonymous to define and execute the following code:

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

String Variable:-

A string variable is **a variable that holds a character string**. It is a section of memory that has been given a name by the programmer. The name looks like those variable names you have seen so far, except that the name of a string variable ends with a dollar sign, \$.

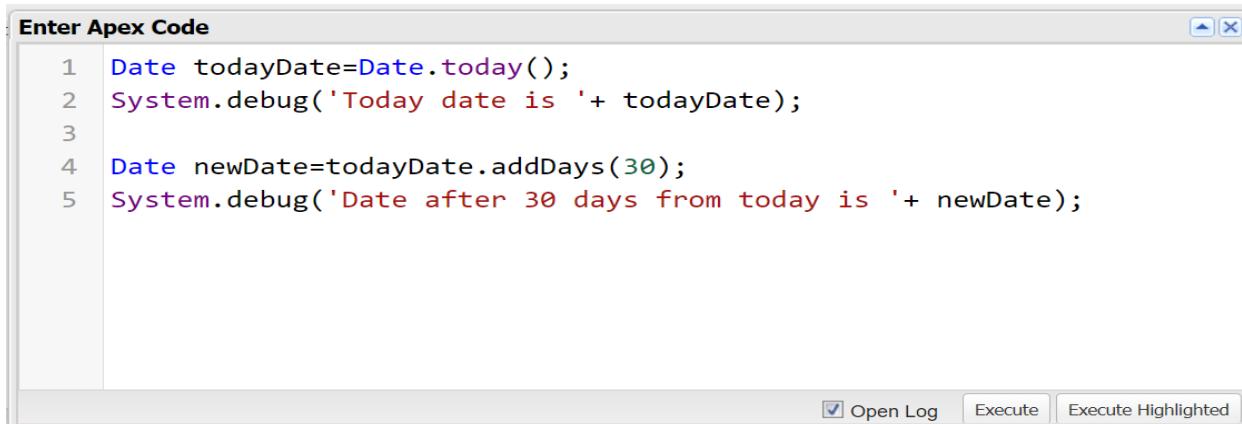
1. In your Trailhead Playground, click  and then select **Developer Console**.
2. Open the **Debug Tab** and select the **Open Execute anonymous Window**.
3. In new window **Enter Apex Code**.
4. Check the **Open Log**.
5. And Click the **Execute button**.
6. Open the File under **Logs**, And check to the **Debug Only**.
7. See the Output.



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

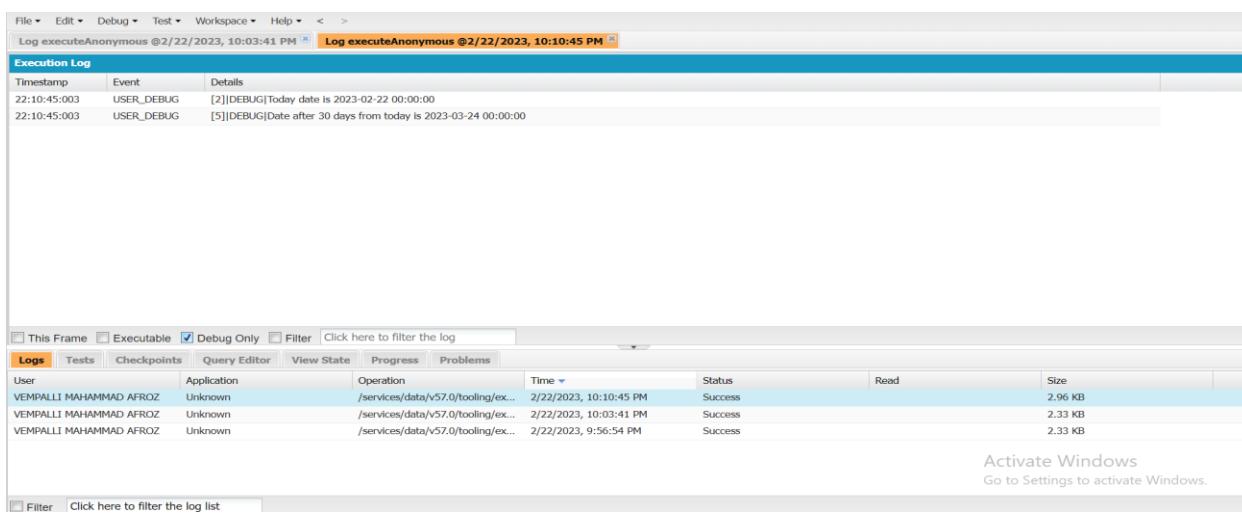
1. **Float:** - stores floating point numbers, with decimals, such as 19.99 or -19.99.
2. **Boolean:** - stores values with two states: true or false.

1. In your Trailhead Playground, click  and then select **Developer Console**.
2. Open the **Debug Tab** and select the **Open Execute anonymous Window**.
3. In new window **Enter Apex Code**.



```
1 Date todayDate=Date.today();
2 System.debug('Today date is '+ todayDate);
3 
4 Date newDate=todayDate.addDays(30);
5 System.debug('Date after 30 days from today is '+ newDate);
```

4. Check the **Open Log**.
5. And Click the **Execute** button.
6. Open the File under **Logs**, And check to the **Debug Only**.
7. See the Output.



Execution Log

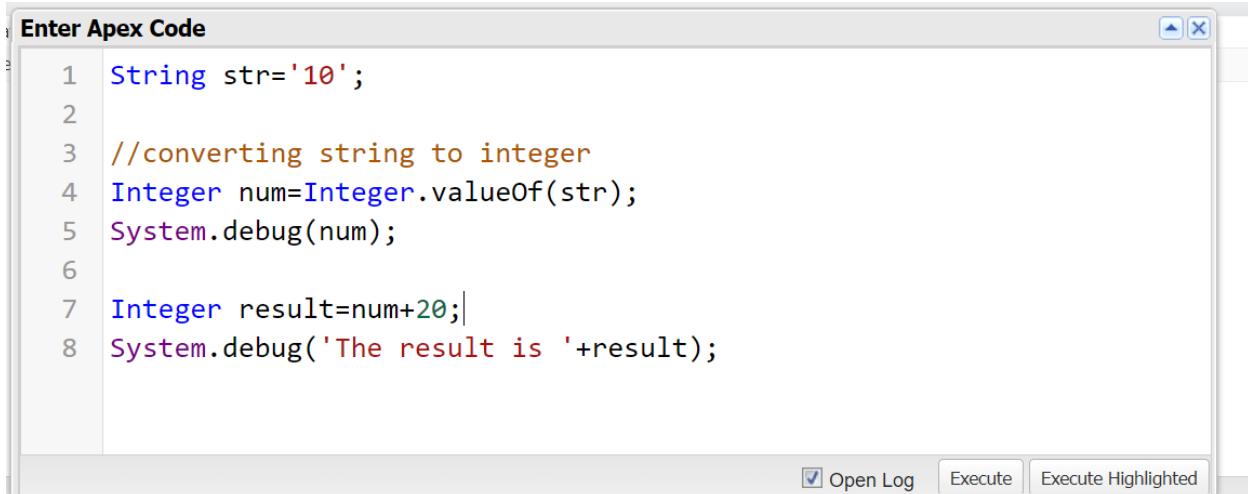
Timestamp	Event	Details
22:10:45:003	USER_DEBUG	[2] DEBUG Today date is 2023-02-22 00:00:00
22:10:45:003	USER_DEBUG	[5] DEBUG Date after 30 days from today is 2023-03-24 00:00:00

Logs

User	Application	Operation	Time	Status	Read	Size
VEMPALLI MAHAMMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 10:10:45 PM	Success		2.96 KB
VEMPALLI MAHAMMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 10:03:41 PM	Success		2.33 KB
VEMPALLI MAHAMMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 9:56:54 PM	Success		2.33 KB

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

1. In your Trailhead Playground, click  and then select **Developer Conole**.
2. Open the **Debug Tab** and select the **Open Execute anonymous Window**.
3. In new window **Enter Apex Code**.



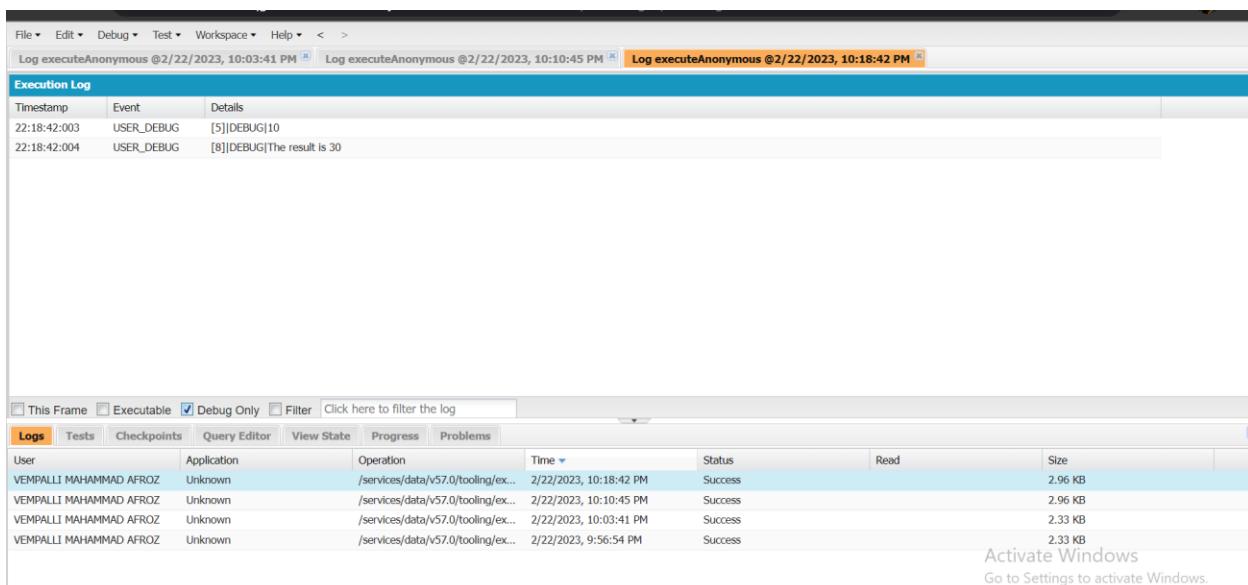
```
1 String str='10';
2
3 //converting string to integer
4 Integer num=Integer.valueOf(str);
5 System.debug(num);
6
7 Integer result=num+20;
8 System.debug('The result is '+result);
```

The screenshot shows the 'Enter Apex Code' window with the following code:

```
1 String str='10';
2
3 //converting string to integer
4 Integer num=Integer.valueOf(str);
5 System.debug(num);
6
7 Integer result=num+20;
8 System.debug('The result is '+result);
```

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

4. Check the **Open Log**.
5. And Click the **Execute button**.
6. Open the File under **Logs**, And check to the **Debug Only**.
7. See the Output.



The screenshot shows the 'Execution Log' window with the following log entries:

Timestamp	Event	Details
22:18:42:003	USER_DEBUG	[5] DEBUG 10
22:18:42:004	USER_DEBUG	[8] DEBUG The result is 30

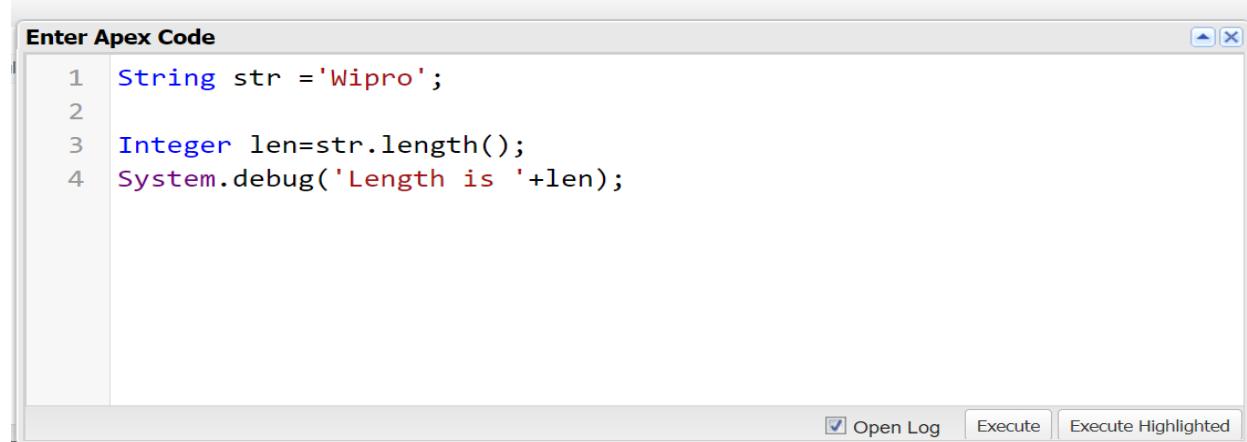
At the bottom of the window, there are several filter options: 'This Frame', 'Executable', 'Debug Only' (which is checked), 'Filter', and 'Click here to filter the log'. There is also a progress bar and a message: 'Activate Windows Go to Settings to activate Windows.'

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

String Variable:-

A string variable is a **variable that holds a character string**. It is a section of memory that has been given a name by the programmer. The name looks like those variable names you have seen so far, except that the name of a string variable ends with a dollar sign, \$.

1. In your Trailhead Playground, click  and then select **Developer Console**.
2. Open the **Debug Tab** and select the **Open Execute anonymous Window**.
3. In new window **Enter Apex Code**.



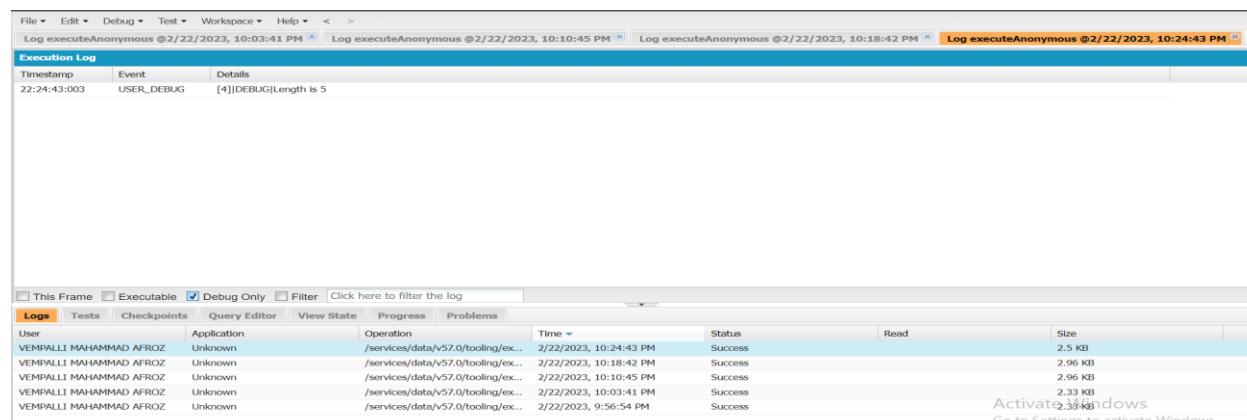
```
1 String str ='Wipro';
2
3 Integer len=str.length();
4 System.debug('Length is '+len);
```

The screenshot shows the "Enter Apex Code" window. It contains the following Apex code:

```
1 String str ='Wipro';
2
3 Integer len=str.length();
4 System.debug('Length is '+len);
```

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

4. Check the **Open Log**.
5. And Click the **Execute button**.
6. Open the File under **Logs**, and check to the **Debug Only**.
7. See the Output.



The screenshot shows the "Execution Log" window. It displays the following log entry:

Timestamp	Event	Details
22:24:43:003	USER_DEBUG	[4] DEBUG Length is 5

Below the log, there is a table titled "Logs" showing a list of log entries. The table has columns: User, Application, Operation, Time, Status, Read, and Size. The data in the table is as follows:

User	Application	Operation	Time	Status	Read	Size
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 10:24:43 PM	Success		2.5 KB
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 10:18:42 PM	Success		2.96 KB
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 10:10:45 PM	Success		2.96 KB
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 10:03:41 PM	Success		2.33 KB
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 9:56:54 PM	Success		2.33 KB

On the right side of the log table, there is a message: "Activate Windows Go to Settings to activate Windows."

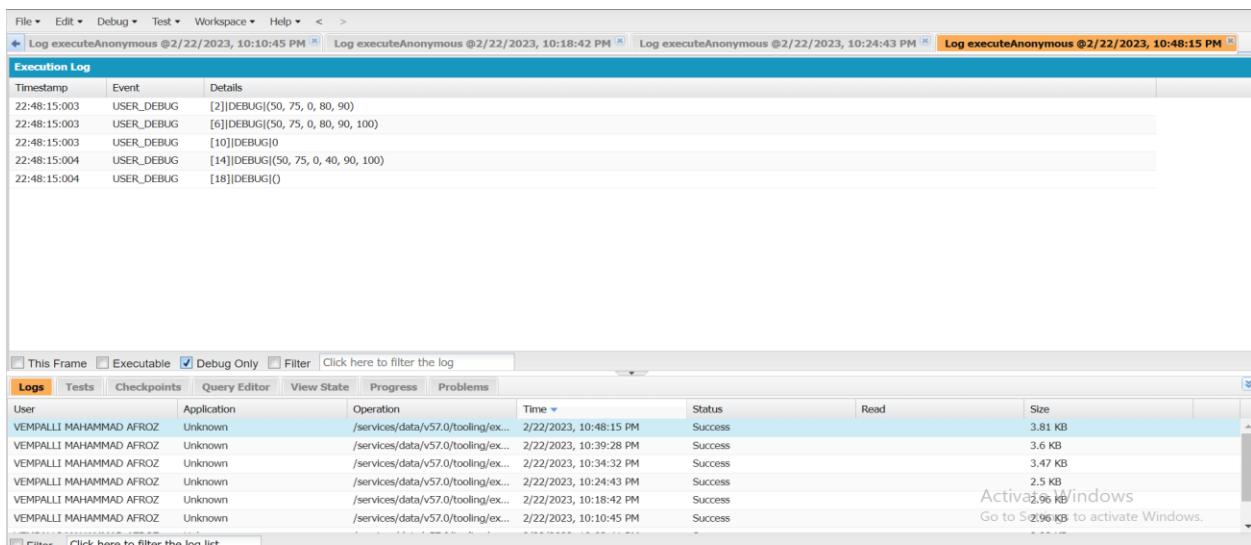
5. Define a List of integer and display the output using add(), get(), set(), clear(), methods.

1. In your Trailhead Playground, click  and then select **Developer Conole**.
2. Open the **Debug Tab** and select the **Open Execute anonymous Window**.
3. In new window **Enter Apex Code**.



```
1 List<Integer> score=new List<Integer>{50,75,0,80,90};
2 System.debug(score);
3
4 //add() method
5 score.add(100);
6 system.debug(score);
7
8 //get() method
9 Integer num=score.get(2);
10 System.debug(num);
11
12 //set() method
13 score.set(3,40);
14 System.debug(score);
15
16 //clear() method
17 score.clear();
18 System.debug(score);
```

4. Check the **Open Log**.
5. And Click the **Execute button**.
6. Open the File under **Logs**, and check to the **Debug Only**.
7. See the Output.



The screenshot shows the Developer Console interface. At the top, there are four tabs labeled "Log executeAnonymous @2/22/2023, 10:10:45 PM", "Log executeAnonymous @2/22/2023, 10:18:42 PM", "Log executeAnonymous @2/22/2023, 10:24:43 PM", and "Log executeAnonymous @2/22/2023, 10:48:15 PM". The bottom tab is highlighted in orange. Below the tabs, the "Execution Log" section displays a table of log entries:

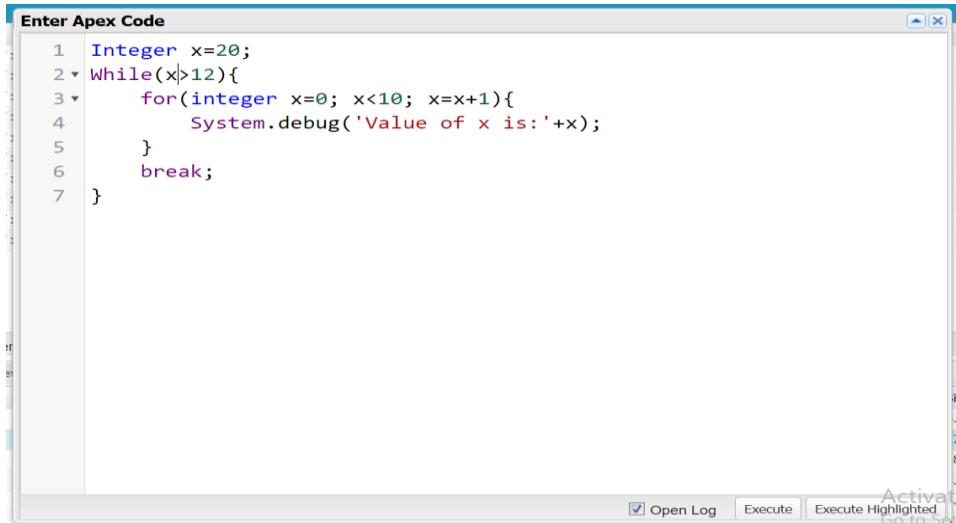
Timestamp	Event	Details
22:48:15:003	USER_DEBUG	[2]DEBUG (50, 75, 0, 80, 90)
22:48:15:003	USER_DEBUG	[6]DEBUG (50, 75, 0, 80, 90, 100)
22:48:15:003	USER_DEBUG	[10]DEBUG 0
22:48:15:004	USER_DEBUG	[14]DEBUG (50, 75, 0, 40, 90, 100)
22:48:15:004	USER_DEBUG	[18]DEBUG 0

At the bottom of the screen, the "Logs" tab is selected, showing a table of log files:

User	Application	Operation	Time	Status	Read	Size
VEMPALLI MAHAMMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 10:48:15 PM	Success		3.81 KB
VEMPALLI MAHAMMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 10:39:28 PM	Success		3.6 KB
VEMPALLI MAHAMMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 10:34:32 PM	Success		3.47 KB
VEMPALLI MAHAMMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 10:24:43 PM	Success		2.5 KB
VEMPALLI MAHAMMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 10:18:42 PM	Success		2.96 KB
VEMPALLI MAHAMMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 10:10:45 PM	Success		2.96 KB

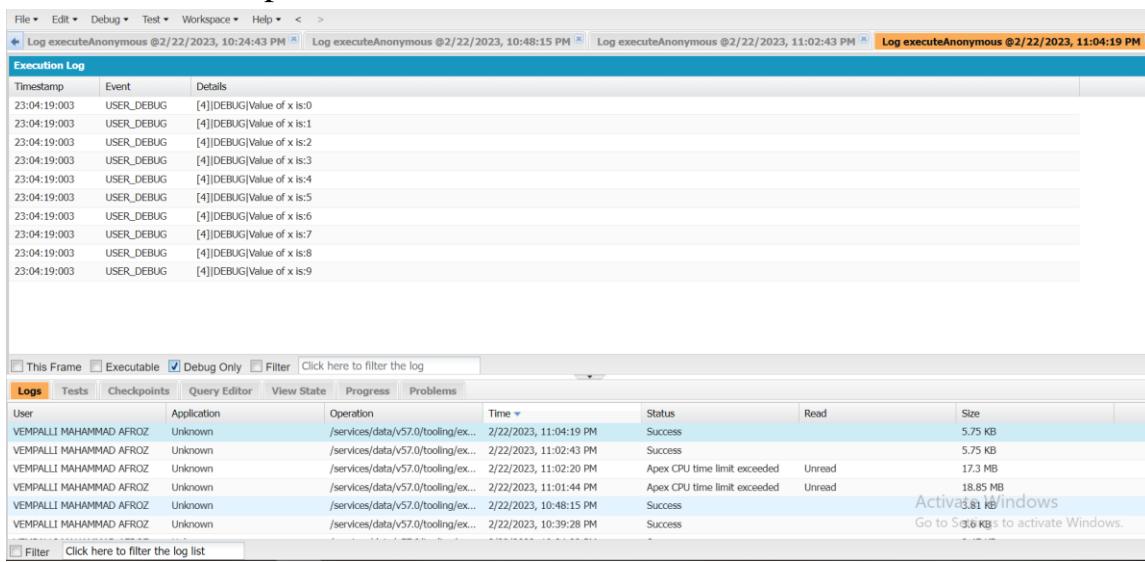
6. Use Execute Anonymous to define and execute the following code to display the value of x = 0 to 9.

1. In your Trailhead Playground, click  and then select **Developer Conole**.
2. Open the **Debug Tab** and select the **Open Execute anonymous Window**.
3. In new window **Enter Apex Code**.



```
Integer x=0;
While(x>12){
    for(integer x=0; x<10; x=x+1){
        System.debug('Value of x is:' +x);
    }
    break;
}
```

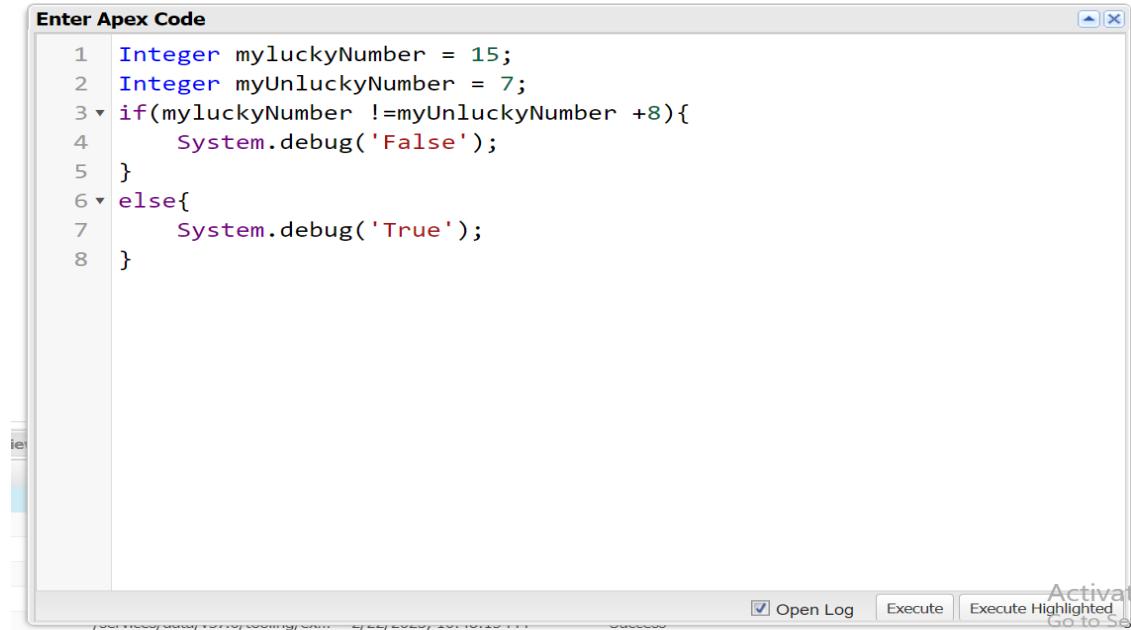
4. Check the **Open Log**.
5. And Click the **Execute** button.
6. Open the File under Logs, And check to the **Debug Only**.
7. See the Output.



Timestamp	Event	Details
23:04:19:003	USER_DEBUG	[4]DEBUG Value of x is:0
23:04:19:003	USER_DEBUG	[4]DEBUG Value of x is:1
23:04:19:003	USER_DEBUG	[4]DEBUG Value of x is:2
23:04:19:003	USER_DEBUG	[4]DEBUG Value of x is:3
23:04:19:003	USER_DEBUG	[4]DEBUG Value of x is:4
23:04:19:003	USER_DEBUG	[4]DEBUG Value of x is:5
23:04:19:003	USER_DEBUG	[4]DEBUG Value of x is:6
23:04:19:003	USER_DEBUG	[4]DEBUG Value of x is:7
23:04:19:003	USER_DEBUG	[4]DEBUG Value of x is:8
23:04:19:003	USER_DEBUG	[4]DEBUG Value of x is:9

EXERCISE-3:- Answer the following in True or False:

1. In your Trailhead Playground, click  and then select **Developer Conole**.
2. Open the **Debug Tab** and select the **Open Execute anonymous Window**.
3. In new window **Enter Apex Code**.

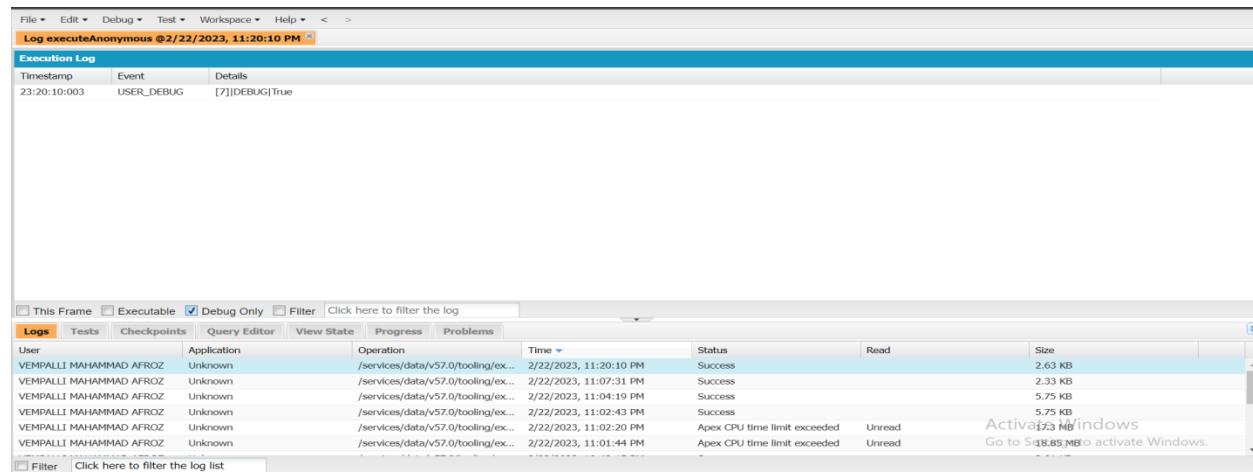


The screenshot shows the 'Enter Apex Code' window. The code entered is:

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

At the bottom of the window, there are three buttons: 'Open Log' (unchecked), 'Execute' (disabled), and 'Execute Highlighted'.

4. Check the **Open Log**.
5. And Click the **Execute button**.
6. Open the File under **Logs**, and check to the **Debug Only**.
7. See the Output.



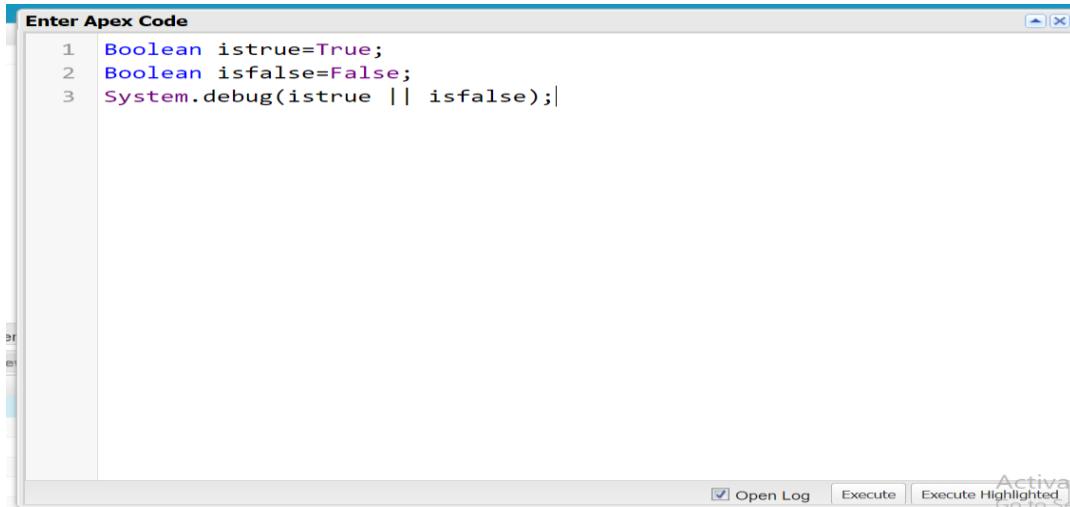
The screenshot shows the 'Execution Log' window. The log entries are:

Timestamp	Event	Details
23:20:10:003	USER_DEBUG	[7][DEBUG]True

At the bottom of the window, there is a 'Logs' tab selected, and a status message: 'Windows Go to Settings to activate Windows.'

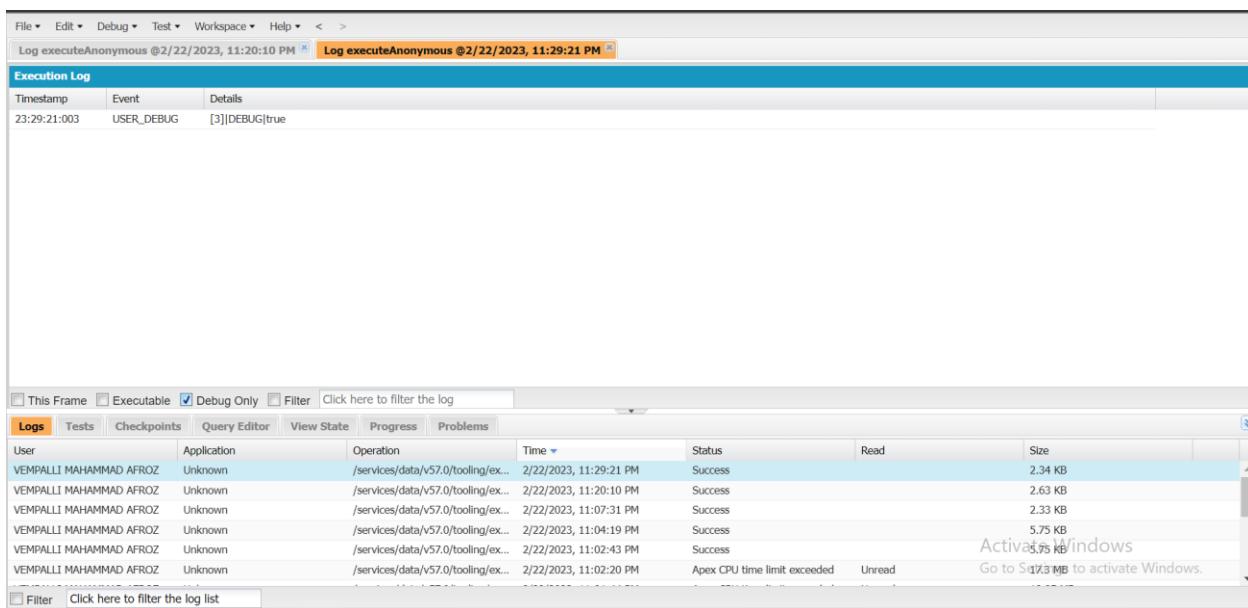
EXERCISE-4:- Answer the following in True or False:

1. In your Trailhead Playground, click  and then select **Developer Conole**.
2. Open the **Debug Tab** and select the **Open Execute anonymous Window**.
3. In new window **Enter Apex Code**.



```
Boolean istrue=True;
Boolean isfalse=False;
System.debug(istrue || isfalse);|
```

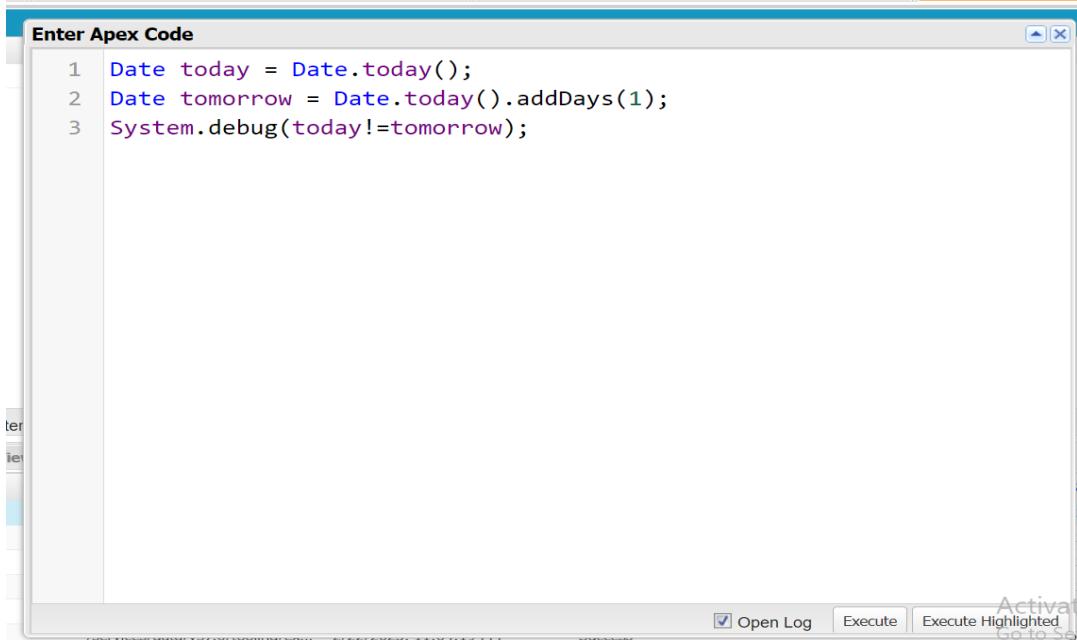
4. Check the **Open Log**.
5. And Click the **Execute button**.
6. Open the File under **Logs**, and check to the **Debug Only**.
7. See the Output.



Timestamp	Event	Details
23:29:21:003	USER_DEBUG	[3] DEBUG true

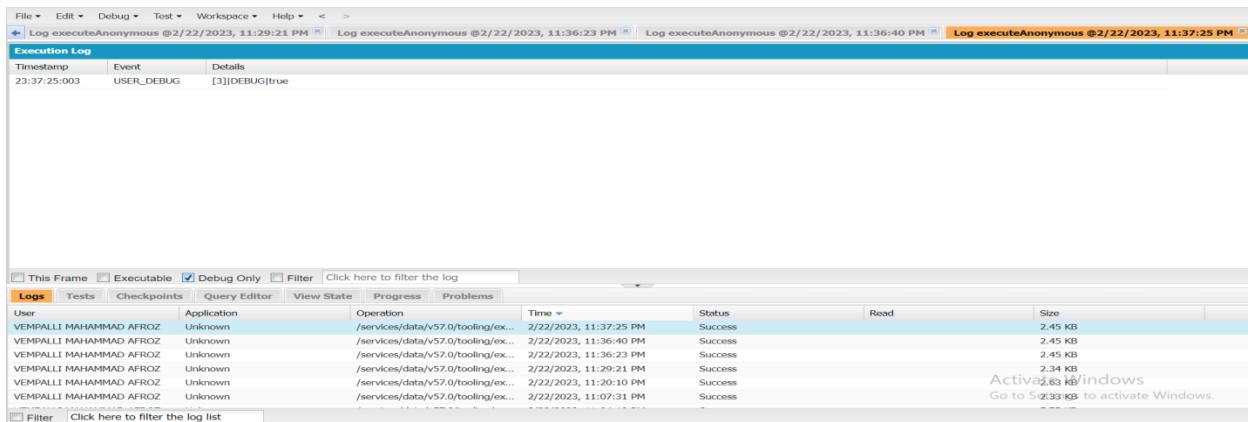
EXERCISE-5:- Answer the following in True or False:

1. In your Trailhead Playground, click  and then select **Developer Conole**.
2. Open the **Debug Tab** and select the **Open Execute anonymous Window**.
3. In new window **Enter Apex Code**.



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

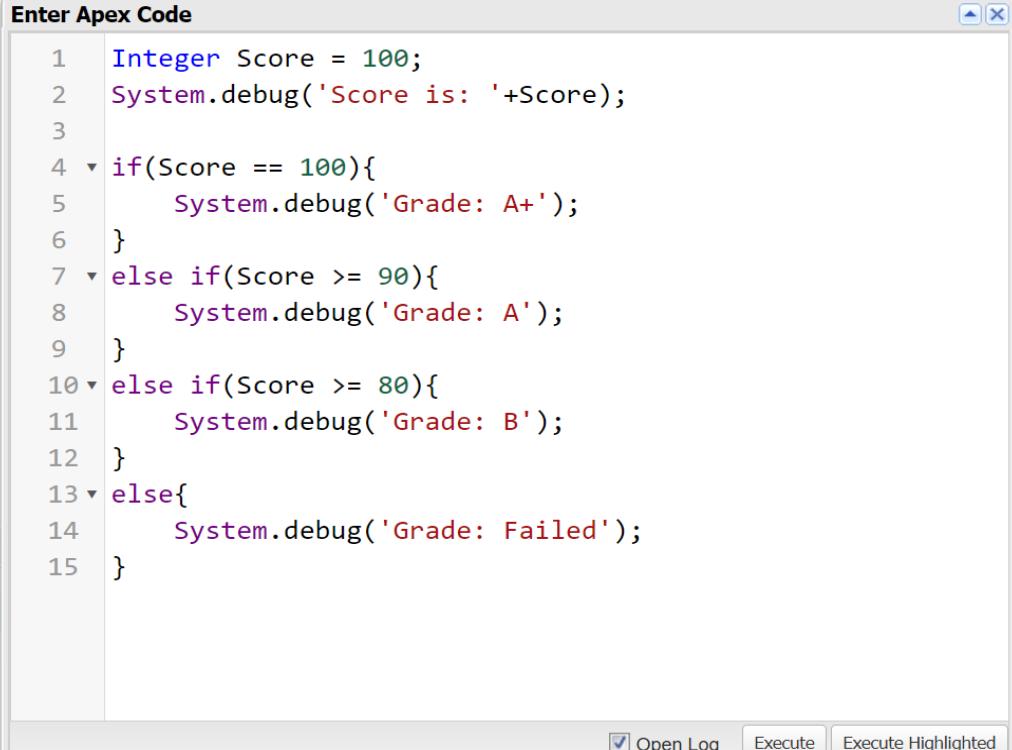
4. Check the **Open Log**.
5. And Click the **Execute button**.
6. Open the File under **Logs**, and check to the **Debug Only**.
7. See the Output.



User	Application	Operation	Time	Status	Read	Size
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 11:37:25 PM	Success		2.45 KB
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 11:36:40 PM	Success		2.45 KB
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 11:36:23 PM	Success		2.34 KB
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 11:29:21 PM	Success		2.63 KB
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 11:20:10 PM	Success		2.33 KB
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 11:07:31 PM	Success		2.33 KB

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

1. In your Trailhead Playground, click  and then select **Developer Conole**.
2. Open the **Debug Tab** and select the **Open Execute anonymous Window**.
3. In new window **Enter Apex Code**.

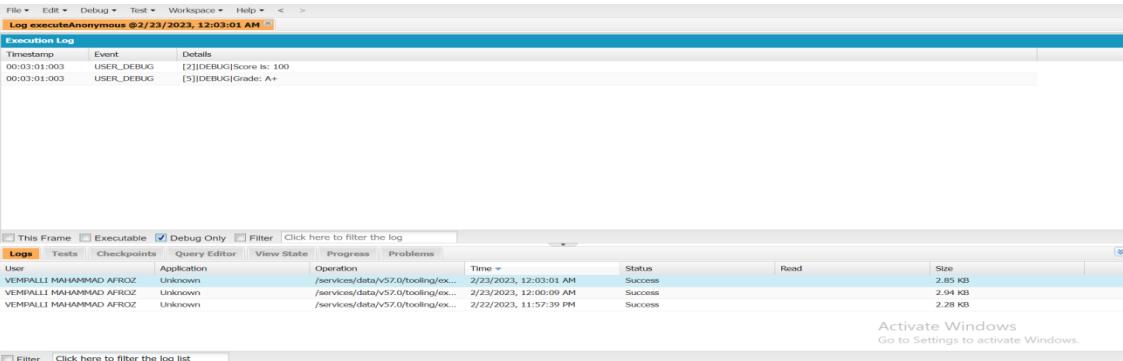


The screenshot shows the "Enter Apex Code" window with the following Apex code:

```
1 Integer Score = 100;
2 System.debug('Score is: '+Score);
3
4 if(Score == 100){
5     System.debug('Grade: A+');
6 }
7 else if(Score >= 90){
8     System.debug('Grade: A');
9 }
10 else if(Score >= 80){
11     System.debug('Grade: B');
12 }
13 else{
14     System.debug('Grade: Failed');
15 }
```

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

4. Check the **Open Log**.
5. And Click the **Execute button**.
6. Open the File under **Logs**, and check to the **Debug Only**.
7. See the Output.



The screenshot shows the "Execution Log" window with the following log entries:

Timestamp	Event	Details
00:03:01:003	USER_DEBUG	[2]DEBUG Score is: 100
00:03:01:003	USER_DEBUG	[5]DEBUG Grade: A+

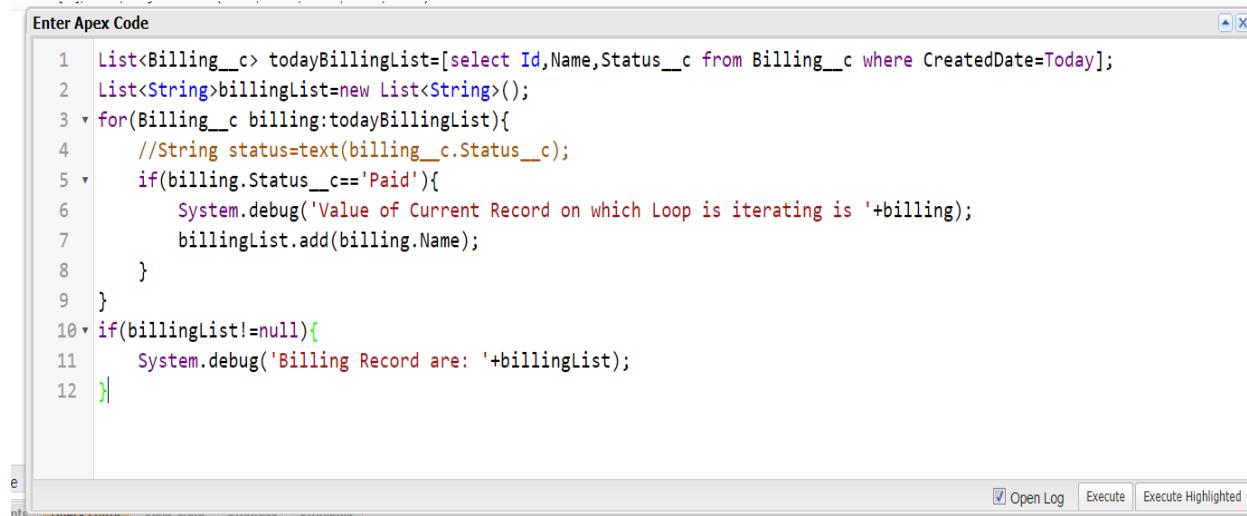
Below the log, there is a table of files:

User	Application	Operation	Time	Status	Read	Size
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/23/2023, 12:03:01 AM	Success		2.85 KB
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/23/2023, 12:00:09 AM	Success		2.94 KB
VEMPALLI MAHMAD AFROZ	Unknown	/services/data/v57.0/tooling/ex...	2/22/2023, 11:57:39 PM	Success		2.28 KB

At the bottom right, it says "Activate Windows" and "Go to Settings to activate Windows".

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

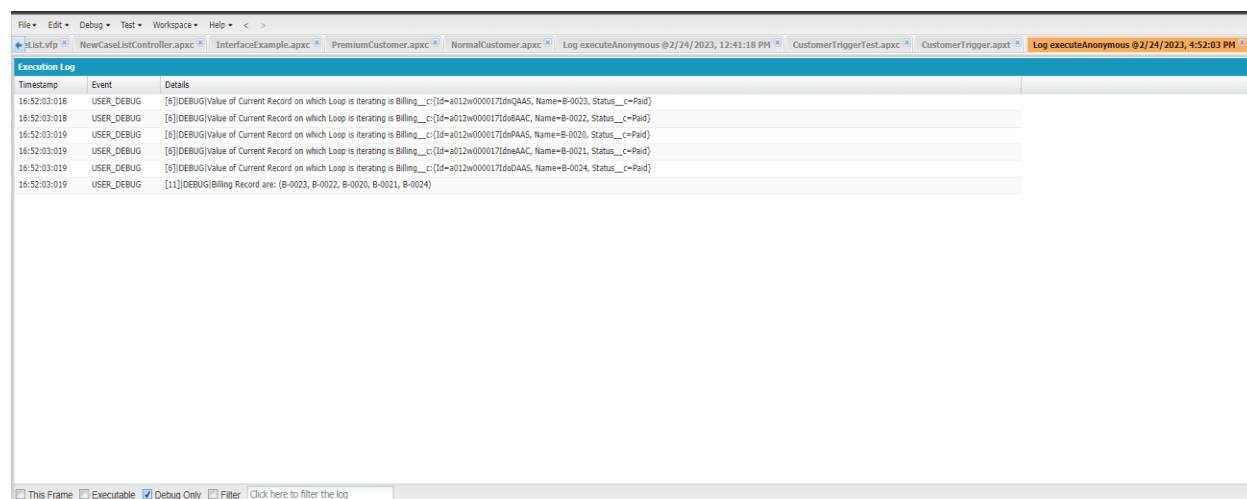
1. In your Trailhead Playground, click  and then select **Developer Conole**.
2. Open the **Debug Tab** and select the **Open Execute anonymous Window**.
3. In new window **Enter Apex Code**.



```
1 List<Billing__c> todayBillingList=[select Id,Name,Status__c from Billing__c where CreatedDate=Today];
2 List<String>billingList=new List<String>();
3 for(Billing__c billing:todayBillingList){
4     //String status=text(billing__c.Status__c);
5     if(billing.Status__c=='Paid'){
6         System.debug('Value of Current Record on which Loop is iterating is '+billing);
7         billingList.add(billing.Name);
8     }
9 }
10 if(billingList!=null){
11     System.debug('Billing Record are: '+billingList);
12 }
```

The screenshot shows the 'Enter Apex Code' window with the above Apex code. Below the code editor are three buttons: 'Open Log' (unchecked), 'Execute' (unchecked), and 'Execute Highlighted' (unchecked).

4. Check the **Open Log**.
5. And Click the **Execute button**.
6. Open the File under **Logs**, and check to the **Debug Only**.
7. See the Output.

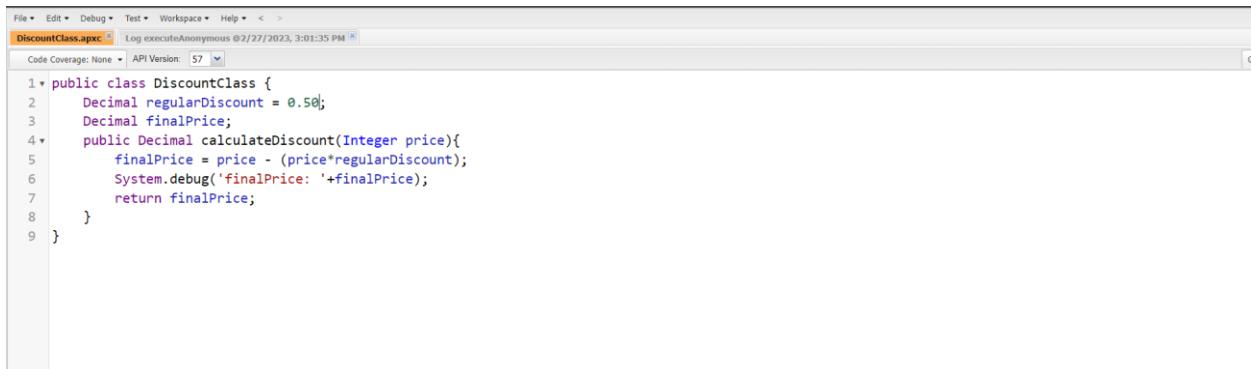


Timestamp	Event	Details
16:52:03:018	USER_DEBUG	[6]DEBUG Value of Current Record on which Loop is iterating is Billing__c:[Id=a012w000017ldnQAS, Name=B-0023, Status__c=<Paid>]
16:52:03:018	USER_DEBUG	[6]DEBUG Value of Current Record on which Loop is iterating is Billing__c:[Id=a012w000017ldn8AC, Name=B-0022, Status__c=<Paid>]
16:52:03:019	USER_DEBUG	[6]DEBUG Value of Current Record on which Loop is iterating is Billing__c:[Id=a012w000017ldnPAAS, Name=B-0020, Status__c=<Paid>]
16:52:03:019	USER_DEBUG	[6]DEBUG Value of Current Record on which Loop is iterating is Billing__c:[Id=a012w000017ldn8AAC, Name=B-0021, Status__c=<Paid>]
16:52:03:019	USER_DEBUG	[6]DEBUG Value of Current Record on which Loop is iterating is Billing__c:[Id=a012w000017ldoDAAS, Name=B-0024, Status__c=<Paid>]
16:52:03:019	USER_DEBUG	[11]DEBUG Billing Record are: (B-0023, B-0022, B-0020, B-0021, B-0024)

The screenshot shows the 'Execution Log' window with the log entries listed above. At the bottom, there are checkboxes for 'This Frame', 'Executable', 'Debug Only' (which is checked), 'Filter', and 'Click here to filter the log'.

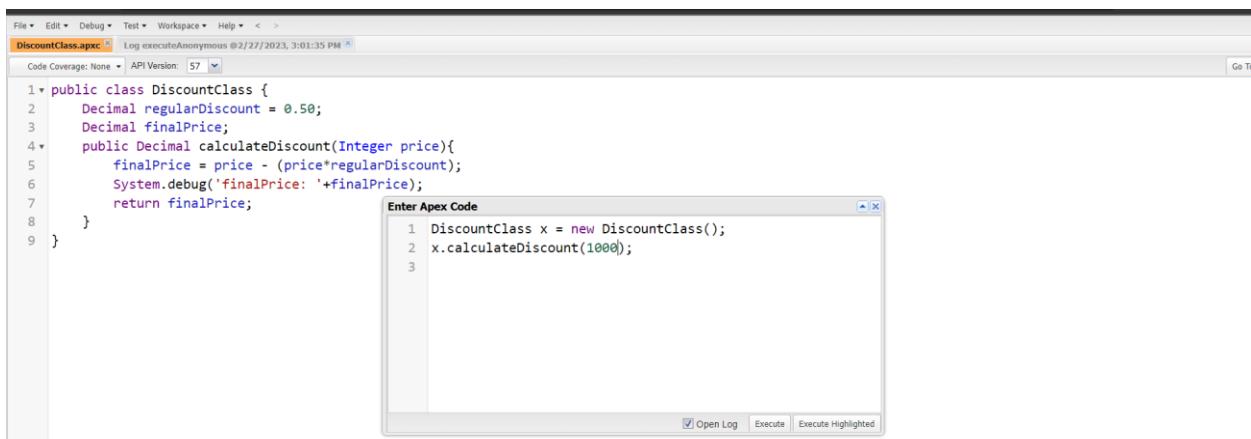
EXERCISE-8:- Write a Class to demo the use of Constants in Apex.

1. In your Trailhead Playground, click  and then select **Developer Conole**.
2. Open New tab Select **Apex class**
3. Class Name = **DiscountClass**.
4. Write a Code:-
 - Define a Final Decimal variable called “regularDiscount”.
 - Define a Decimal variable called “finalPrice”.
 - Define a Method called “calculateDiscount(Integer price)”, takes a Decimal data type.
 - finalPrice = price – (price*regularDiscount)
 - Don’t forget to return a Decimal data type.



```
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
DiscountClass.apxc Log executeAnonymous @2/27/2023, 3:01:35 PM
Code Coverage: None ▾ API Version: 57 ▾ G
1 * public class DiscountClass {
2     Decimal regularDiscount = 0.50;
3     Decimal finalPrice;
4     public Decimal calculateDiscount(Integer price){
5         finalPrice = price - (price*regularDiscount);
6         System.debug('finalPrice: '+finalPrice);
7         return finalPrice;
8     }
9 }
```

5. Use **Execute Anonymous Window** to execute this code.



```
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
DiscountClass.apxc Log executeAnonymous @2/27/2023, 3:01:35 PM
Code Coverage: None ▾ API Version: 57 ▾ Go To
1 * public class DiscountClass {
2     Decimal regularDiscount = 0.50;
3     Decimal finalPrice;
4     public Decimal calculateDiscount(Integer price){
5         finalPrice = price - (price*regularDiscount);
6         System.debug('finalPrice: '+finalPrice);
7         return finalPrice;
8     }
9 }
```

Enter Apex Code

```
1 DiscountClass x = new DiscountClass();
2 x.calculateDiscount(1000);
3
```

Open Log

6. Check the **Open Log**.

7. And Click the **Execute** button.
8. Open the File under **Logs**, and check to the **Debug Only**.
9. See the Output.

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for DiscountClass.apxc and Log executeAnonymous @2/27/2023, 3:01:35 PM. The main area is titled "Execution Log" with columns for Timestamp, Event, and Details. A single log entry is displayed: 15:01:35:013, USER_DEBUG, [6]|DEBUG|finalPrice: 500.00.

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

1. In your Trailhead Playground, click and then select Developer Conole.
2. Open New tab Select **Apex class**
3. Class Name = **InterfaceExample**.
 - Method Signature = Double percentageDiscountTobeApplied();

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for various files like OppView.vfp, AccountList.vfp, etc., and InterfaceExample.apxc. The main area shows the code for the InterfaceExample class:

```

1 public Interface InterfaceExample {
2     Double percentageDiscountTobeApplied();
3 }
4

```

4. Create an Apex Class to implement the Interface.
5. Class Name = **PremiumCustomer**.
 - Call the interface method to return 30% Discount.

```
File Edit Debug Test Workspace Help < >
OppView.vfp AccountList.vfp NewCaseList.vfp NewCaseListController.apxc CustomerTrigger.apxt CustomerTriggerTest.apxc InterfaceExample.apxc PremiumCustomer.apxc NormalCustomer.apxc Log execi
Code Coverage: None API Version: 57 Go To
1 public class PremiumCustomer implements InterfaceExample {
2     public Double percentageDiscountTobeApplied() {
3         return 0.3; // 30% discount
4     }
5 }
6 }
```

6. Create another Apex Class to implement the Interface.

7. Class Name = **normalCustomer**.

- Call the interface method to return 10% Discount.

```
File Edit Debug Test Workspace Help < >
OppView.vfp AccountList.vfp NewCaseList.vfp NewCaseListController.apxc CustomerTrigger.apxt CustomerTriggerTest.apxc InterfaceExample.apxc PremiumCustomer.apxc NormalCustomer.apxc Log execi
Code Coverage: None API Version: 57 Go To
1 public class NormalCustomer implements InterfaceExample {
2     public Double percentageDiscountTobeApplied() {
3         return 0.1; // 10% discount
4     }
5 }
6 }
7 }
```

8. Use **Execute Anonymous Window** to execute this code.

```
File Edit Debug Test Workspace Help < >
OppView.vfp AccountList.vfp NewCaseList.vfp NewCaseListController.apxc CustomerTrigger.apxt CustomerTriggerTest.apxc InterfaceExample.apxc PremiumCustomer.apxc NormalCustomer.apxc Log execi
Code Coverage: None API Version: 57 Go To
1 public Interface InterfaceExample {
2     Double percentageDiscountTobeApplied();
3 }
4 
```

Enter Apex Code

```
1 InterfaceExample customer = new NormalCustomer();
2 Double discount = customer.percentageDiscountTobeApplied();
3 System.debug('Discount: ' + discount);
```

Logs Tests Checkpoints Query Editor View State Progress Problem

Name Line Problem

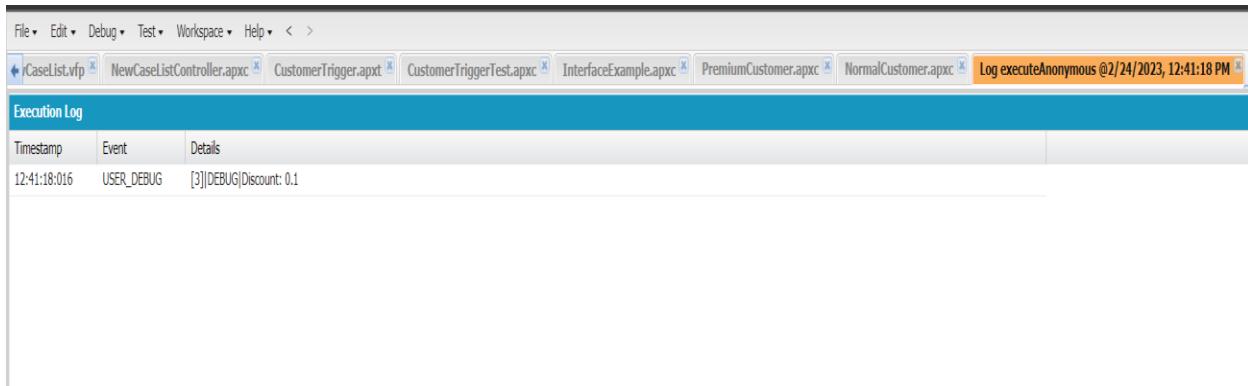
Open Log Execute Execute Highlighted

9. Check the **Open Log**.

10. And Click the **Execute button**.

11. Open the File under **Logs**, and check to the **Debug Only**.

12. See the Output.

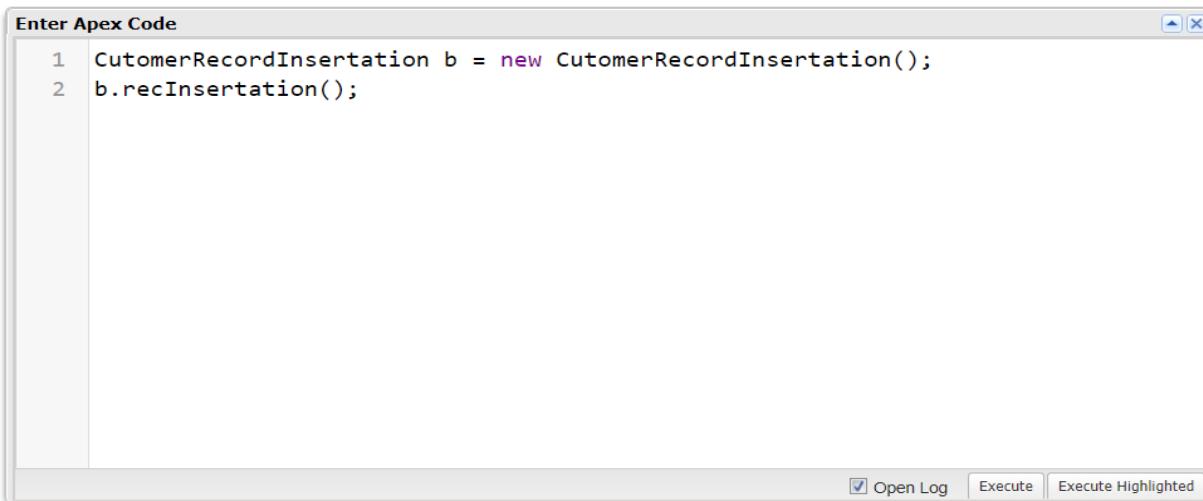


EXERCISE-10:- Demo on DML Insert Operation Using Database methods.

1. Insert a Customer Record First using simple DML Statement.
2. Customer Name = “Wipro”, Type = “Premium”.
3. Insert Billing record using Database methods.
4. Billing Status = Paid, Amount Paid = 5000000, link Billing record to Customer using the customer ID.
5. Use a List<Billing c> to perform the Database insert method & store the result in srList.
6. Iterate through the Success/Error result, show inserted record ID, or the error message with fields.

```
File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
DisqualificationTestLeads.apxc DisqualificationTestLeads.apxt CustomerRecordInsertion.apxc CustomerTriggerTest.apxc CustomerTrigger.apxt
Code Coverage: None ▾ API Version: 57 ▾ Go To
1 public class CustomerRecordInsertion {
2     public void recInsertation() {
3         Customer__c customer = new Customer__c();
4         Customer.Name ='Wipro';
5         Customer.Customer_Type__c = 'Premium';
6         Insert Customer;
7         Id Customer_id = Customer.Id;
8         Billing__c[] billing = new List<Billing__c>{
9             new Billing__c(Status__c = 'Paid', Amount_Paid__c = 5000000, Customer__c = Customer_id),new Billing__c()};
10        Database.SaveResult[] 1 = Database.insert(billing,false);
11
12        for (database.SaveResult sr : 1){
13            if(sr.isSuccess()){
14                System.debug('Record was Inserted.Billing ID is: '+sr.getId());
15            }
16            else{
17                for(Database.Error e : sr.getErrors()){
18                    System.debug(e.getStatusCode() + ':' +e.getMessage());
19                    System.debug('Billing Fields Which Have Error: '+e.getFields());
20                }
21            }
22        }
23    }
24 }
```

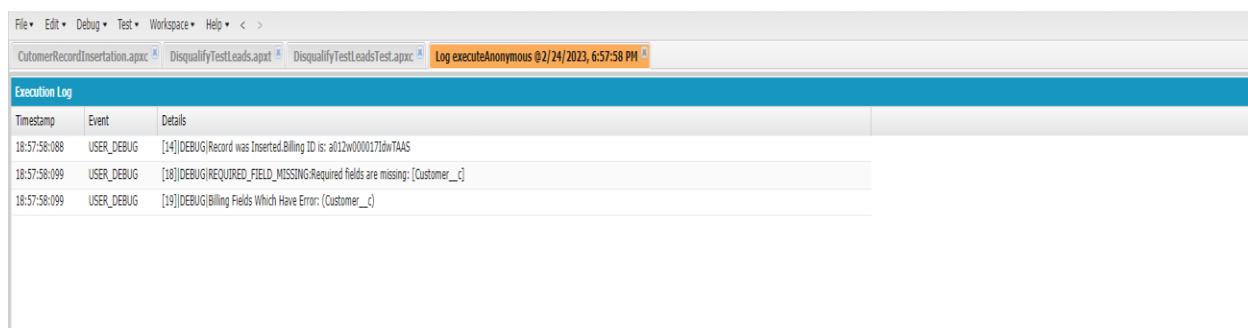
7. Use **Execute Anonymous Window** to execute this code.



```
1 CustomerRecordInsertation b = new CustomerRecordInsertation();
2 b.recInsertation();
```

Open Log Execute Execute Highlighted

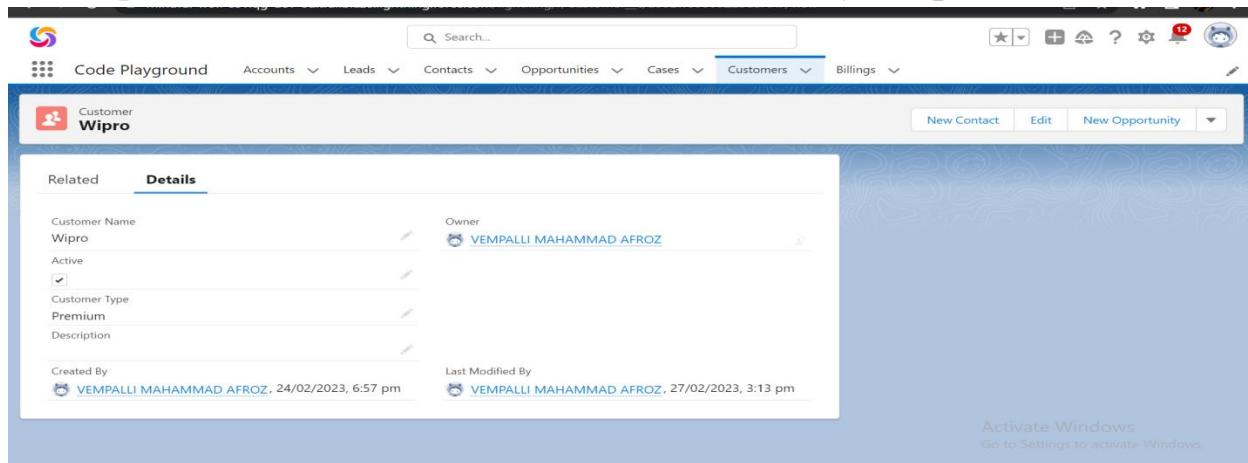
8. See the Output.



Timestamp	Event	Details
18:57:58:088	USER_DEBUG	[14]DEBUG Record was Inserted.Billing ID is: a012w000017dwTAAS
18:57:58:099	USER_DEBUG	[18]DEBUG REQUIRED_FIELD_MISSING Required fields are missing: [Customer__c]
18:57:58:099	USER_DEBUG	[19]DEBUG Billing Fields Which Have Error: (Customer__c)

9. Open the Code Playground App.

10. Open the **Customer** tab select the record named by “Wipro”. Active it



Customer Wipro

Related Details

Customer Name: Wipro

Active:

Customer Type: Premium

Description:

Created By: VEMPALLI MAHAMMAD AFROZ, 24/02/2023, 6:57 pm

Last Modified By: VEMPALLI MAHAMMAD AFROZ, 27/02/2023, 3:13 pm

Activate Windows
Go to Settings to activate Windows.

11. Click the **Billing** tab. Open the All billings and you will see the latest record.

The screenshot shows two views of a CRM application. The top view is a list of all billings, showing 8 items. The columns are: Billing Number (sorted by number), Customer Type, Amount Paid, and Status. The bottom view is a detailed view of a specific billing record for B-0028, which belongs to Wipro and has an amount paid of ₹50,00,000.

	Billing Number ↑	Customer Type	Amount Paid	Status
1	B-0020	Premium	₹10,000	Paid
2	B-0021	Premium	₹10,00,000	Paid
3	B-0022	Standard	₹1,000	Paid
4	B-0023	Standard	₹2,000	Paid
5	B-0024	Premium	₹5,000	Paid
6	B-0025	Standard	₹3,000	Unpaid
7	B-0026	Premium	₹8,000	Unpaid
8	B-0028		₹50,00,000	Paid

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

1. Display ID, Amount, Stage, Account Name, Account Industry, Account Website from Opportunity.
2. Add a Where Clause “Account Industry = Energy”.
3. AND “Account Annual Revenue > 5000”.

```

Log executeAnonymous @2/23/2023, 11:11:18 AM [ Opportunity@11:37 AM ]
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
0062w00000K4LcAAN 125000 Negotiation/Review United Oil & Gas Corp. Energy http://www.uos.com
0062w00000K4LcVAN 270000 Proposal/Price Quote United Oil & Gas Corp. Energy http://www.uos.com
0062w00000K4LcWAN 120000 Closed Won United Oil & Gas Corp. Energy http://www.uos.com
0062w00000K4LD4AAN 270000 Negotiation/Review United Oil & Gas Corp. Energy http://www.uos.com
0062w00000K4LD7AAN 270000 Closed Won United Oil & Gas Corp. Energy http://www.uos.com
0062w00000K4LD9AAN 915000 Closed Won United Oil & Gas Corp. Energy http://www.uos.com
0062w00000K4LDEA3 235000 Closed Won United Oil & Gas Corp. Energy http://www.uos.com
0062w00000K4LDFAA3 440000 Closed Won United Oil & Gas Corp. Energy http://www.uos.com
0062w00000K4LDHA3 120000 Closed Won United Oil & Gas Corp. Energy http://www.uos.com
0062w00000K4LDJAA3 675000 Needs Analysis United Oil & Gas Corp. Energy http://www.uos.com

Query Grid: Save Rows | Insert Row | Delete Row | Refresh Grid | Access in Salesforce: Create New | Open Detail Page | Edit Page
Logs Tests Checkpoints Query Editor View State Progress Problems History
SELECT Id, Amount, StageName, Account.Name, Account.Industry, Account.Website
FROM Opportunity
WHERE Account.Industry = 'Energy' AND Account.AnnualRevenue > 5000
Executed ▲
SELECT Id, Amount, StageName, Account.Name, Account.Industry, Account.Website

```

EXERCISE-12:- Write an Apex Trigger.

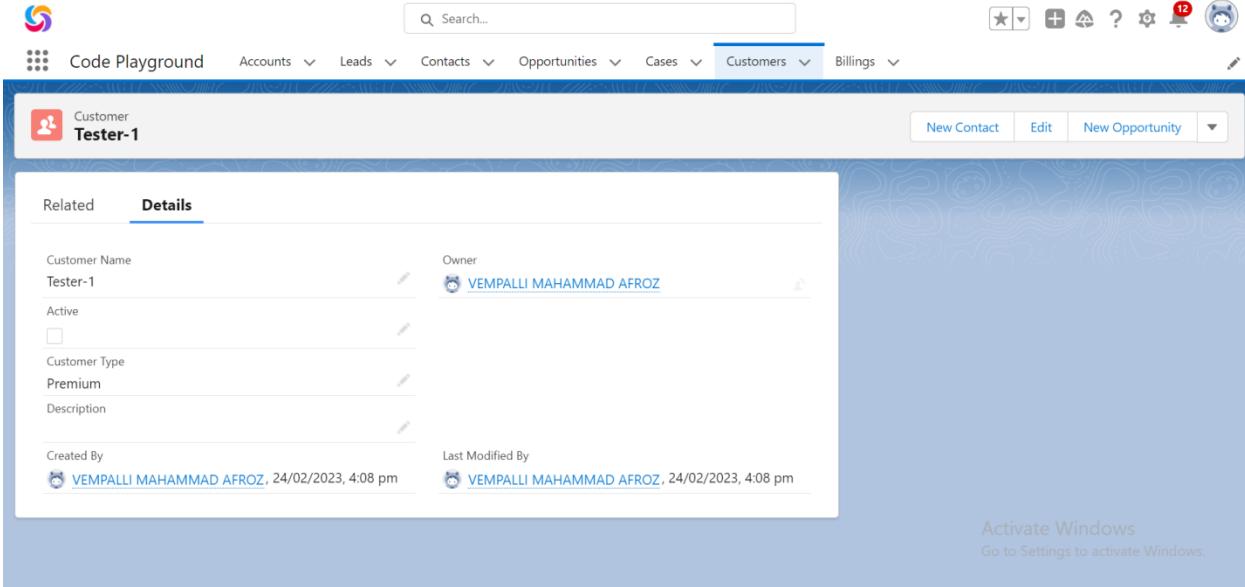
1. Open Developer Console.
2. Open the File tab. Click **New**.
3. Select **Apex Trigger**.
4. Trigger Name = **CustomerTrigger**.
5. Object = **Customer__c**

```

trigger CustomerTrigger on Customer__c (after update) {
    List<Billing__c> billingRecords=new List<Billing__c>();
    for(Customer__c customer:Trigger.new){
        if(customer.Active__c==true && Trigger.oldmap.get(customer.id).Active__c==false){
            Billing__c billing=new Billing__c();
            billing.Status__c='Paid';
            billing.Amount_Paid__c=1000000;
            billing.Customer__c=customer.Id;
            billingRecords.add(billing);
        }
    }
    if(billingRecords!=null){
        insert billingRecords;
    }
}

```

6. Create a Billing Record when on Customer Object the 'Status' field changes to **Active** from **Inactive**.



7. Billing “**Status**” must be: Paid, Amount Paid = **1000000**.
8. Perform the DML on a List.

EXERCISE-13:- Write a Test Class for CustomerTrigger.

1. Open Developer Console.
2. Open the **File** tab click the **New**.
3. Select **Apex Class**.
4. Class Name = **CustomerTriggerTest**.

```

trigger CustomerTrigger on Customer__c (after update) {
    List<Billing__c> billingRecords=new List<Billing__c>();
    for(Customer__c customer:Trigger.new){
        if(customer.Active__c==true && Trigger.oldmap.get(customer.id).Active__c==false){
            Billing__c billing=new Billing__c();
            billing.Status__c='Paid';
            billing.Amount_Paid__c=1000000;
            billing.Customer__c=customer.Id;
            billingRecords.add(billing);
        }
    }
    if(billingRecords!=null){
        insert billingRecords;
    }
}

```

5. Click the **Run Test** Button.
6. Back to the **Code Playground App**.

7. Open **Customer** tab and open the record.
8. Check the **Active**.

The screenshot shows the 'Customer' record for 'Tester-1'. The 'Details' tab is selected. Key fields shown include:

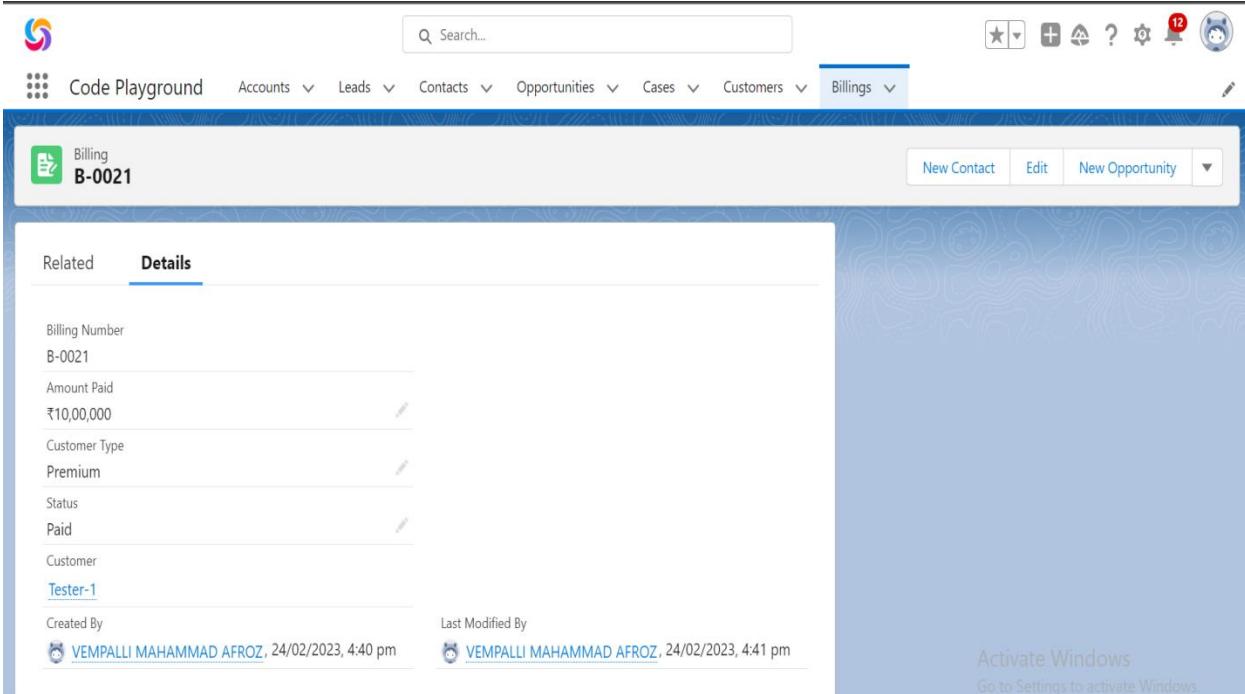
- Customer Name: Tester-1
- Owner: VEMPALLI MAHMAD AFROZ
- Active: checked (indicated by a checked checkbox)
- Customer Type: Premium
- Description: (empty)
- Created By: VEMPALLI MAHMAD AFROZ, 24/02/2023, 4:08 pm
- Last Modified By: VEMPALLI MAHMAD AFROZ, 24/02/2023, 4:40 pm

9. Now open the **Billing** tab. Refersh the page.
10. And we get new Billing Record.

The screenshot shows the 'Billings' list page. The 'Recently Viewed' section displays two items:

- 1 B-0021
- 2 B-0020

11. Open it.



12. Test Class must have full code coverage.

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

CustomerTrigger.apxt CustomerTriggerTest.apxc

Code Coverage: CustomerTriggerTest.testCustomer 100% ▾ API Version: 57 ▾ Go To

```

1 trigger CustomerTrigger on Customer_c (after update) {
2     List<Billing_c> billingRecords=new List<Billing_c>();
3     for(Customer_c customer:Trigger.new){
4         if(customer.Active_c==true && Trigger.oldmap.get(customer.id).Active_c==false){
5             Billing_c billing=new Billing_c();
6             billing.Status_c='Paid';
7             billing.Amount_Paid_c=1000000;
8             billing.Customer_c=customer.Id;
9             billingRecords.add(billing);
10        }
11    }
12    if(billingRecords!=null){
13        insert billingRecords;
14    }
15 }

```

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
VEMPALLI MAHMAMD AFROZ	Unknown	/services/data/v57.0/tooling/r... u...	2/26/2023, 9:26:36 PM	Success		3.89 KB

EXERCISE-14:- Write an Apex Trigger.

1. Open Developer Console.
2. Open the **File** tab. Click **New**.
3. Select **Apex Trigger**.
4. Trigger Name = DisqualifyTestLeads.
5. Object = **Leads**.

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for various files like DiscountClass.aprx, Log executeAnonymous @2/27/2023, 3:01:35 PM, CustomerTriggerTest.apxc, CustomerTrigger.apxt, DisqualifyTestLeads.apxt, and DisqualifyTestLeadsTest.apxc. A dropdown for Code Coverage is set to None, and the API Version is 57. The main editor area contains the following Apex code:

```
trigger DisqualifyTestLeads on Lead (before insert,before update) {
    List<Lead> leadList=new List<Lead>();
    for(Lead ld:Trigger.new){
        if(ld.FirstName==null){
            ldaddError('Enter name');
        }
        else{
            if((ld.FirstName=='Test' && ld.FirstName.length()!=0) || (ld.LastName=='Test' && ld.LastName.length()!=0)){
                System.debug(ld.FirstName+' '+ld.LastName+' '+'Will be Disqualified');
                leadList.add(ld);
            }
        }
        for(Lead lead:leadList){
            lead.Status='Disqualified';
        }
    }
}
```

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

6. Define logic to find Leads with ‘Test’ in the Name, Ignore Case.
7. Validate that the Lead First/Last Name must not be Blank.
8. IF ‘Test Lead’ found, system.debug(Lead First Name + ‘ ’ + Lead Last Name + ‘Will be disqualified!’).
9. Add the disqualified Leads to a List<Lead>, use for loop to iterate through the list and update Lead status field to ‘Disqualified’.

EXERCISE-15:- Write a Test Class for DisqualifyTestLeads.

1. Open Developer Console.
2. Open the **File** tab click the **New**.
3. Select **Apex Class**.
4. Class Name = **DisqualifyTestLeadsTest**.
5. Create the Data for the Lead Object
6. Use `Test.startTest()` & `Test.stopTest()`
7. Use assert statement to validate the output,

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a search bar. The main area displays the code for `DisqualifyTestLeadsTest.apxc`. The code defines two test methods: `DisqualifyTestMethod` and `DisqualifyTestMethod1`, both annotated with `@isTest`. Each method creates a new `Lead` object with specific values for `LastName` and `Company`, then performs a `Test.startTest`, inserts the lead, and performs a `Test.stopTest`. The API version is set to 57. A watermark for "Activate Windows" is visible in the bottom right corner.

```
1  @IsTest
2  *public class DisqualifyTestLeadsTest {
3      @isTest public static void DisqualifyTestMethod(){
4          Lead ld = new Lead();
5          ld.LastName = 'XYZ';
6          ld.Company = 'ABC';
7
8          Test.startTest();
9          insert ld;
10         Test.stopTest();
11     }
12
13     @isTest public static void DisqualifyTestMethod1(){
14         Lead ld = new Lead();
15         ld.FirstName = 'Test';
16         ld.LastName = 'XYZ';
17         ld.Company = 'ABC';
18
19         Test.startTest();
20         insert ld;
21         Test.stopTest();
22     }
23
24 }

```

8. Run the test Calss.
9. Check the Debug only.
10. Output:-

The screenshot shows the Salesforce Developer Console interface with the execution log open. The log header indicates "Log executeAnonymous @2/26/2023, 10:00:56 PM". The log table has columns for Timestamp, Event, and Details. A single entry is shown: a `USER_DEBUG` event at 22:00:56:064 with the message "[11]DEBUG:Test XYZ Will be Disqualified".

Timestamp	Event	Details
22:00:56:064	USER_DEBUG	[11]DEBUG:Test XYZ Will be Disqualified

11. Navigate back to Code Playground App.
12. Open the Leads Tab.
13. Create New.
14. And Save it.
15. Lead Status will Automatically Changed.

The screenshot shows the Salesforce Leads page for a lead named "Mr. Test XYZ". The lead's details include:

- Lead Owner:** VEMPALLI MAHMAD AFROZ
- Name:** Mr. Test XYZ
- Company:** ABC
- Title:** (empty)
- Lead Source:** (empty)
- Industry:** (empty)
- Annual Revenue:** (empty)
- Phone:** (empty)
- Mobile:** (empty)
- Fax:** (empty)
- Email:** (empty)
- Website:** (empty)
- Lead Status:** Disqualified
- Rating:** (empty)
- No. of Employees:** (empty)

The "Details" tab is selected. On the right, there is a "Related" section with a message: "We found no potential duplicates of this Lead." It also shows a "Campaign History (0)" section. At the bottom right, there is a message: "Activate Windows Go to Settings to activate Windows."

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

1. Open Developer Console.
2. Open the **File** tab click the **New**.
3. Select **Visualforce Page**.
4. Page Name = OppView.
5. It must reference the Opportunity standard controller.
6. Define 4 apex:outputField components:
7. **Opportunity Name, Amount, Close Date, Opportunity Account Name.**

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a preview button. Below the navigation is a tab bar with 'OppView.vfp' selected, followed by 'AccountList.vfp' and 'NewCaseList.vfp'. The main area displays the Visualforce page code:

```

1 <apex:page standardController="Opportunity" lightningStylesheets="true">
2   <apex:sectionHeader title="Opportunity Detail" subtitle="{!Opportunity.Name}" />
3   <apex:pageBlock title="Opportunity Information">
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     </apex:pageBlockSection>
10    </apex:pageBlock>
11 </apex:page>

```

8. Click on the **Preview Button**.
9. And output should like this.

The screenshot shows the preview of the 'OppView.vfp' page. The title is 'Opportunity Detail' with a crown icon. Below it is a section titled 'Opportunity Information'.

Opportunity Name	Amount
Close Date	Account Name

EXERCISE-17:- Create a Visualforce page which shows a list of Accounts linked to their record pages.

1. Open Developer Console.
2. Open the **File** tab click the **New**.
3. Select **Visualforce Page**.
4. Page Name = AccountList
5. It must reference the Account standard controller.
6. It must have a recordSetVar attribute equal to accounts.
7. It must have a Visualforce apex:repeat component, with the following:
 - Use the var attribute set to “a”, Use the HTML list tag
 - Use the apex:outputLink component to link to the respective record detail page.

```

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

```

8. Click on Preview button.
9. It shows the Accounts list.

- 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

10. Click on any one.
11. It redirects to your Account list in Org.

The screenshot shows the Salesforce Account detail page for 'Burlington Textiles Corp of America'. The top navigation bar includes 'Code Playground' and various tabs like Accounts, Leads, Contacts, Opportunities, Cases, Customers, and Billings. The main account card displays basic information: Type (Customer - Direct), Phone ((336) 222-7000), Website (www.burlington.com), Account Owner (VEMPALLI MAHAMMAD AFROZ), Account Site, and Industry (Apparel). Below the card, the 'Related' section shows 'Contacts (1)' with a list for 'Jack Rogers' (Title: VP, Facilities, Email: jrogers@burlington..., Phone: (336) 222-7000). The right sidebar features sections for 'Activity' (with no items) and 'Chatter' (with a message from Salesforce). A note at the bottom of the sidebar says 'Get started by sending an email, scheduling a task, and more.' and 'Go to Settings to activate Windows.'

EXERCISE-18:- Create a Visualforce page that uses a custom controller to display a list of cases with the status of 'New'.

1. Open Developer Console.
2. Open the **File** tab click the **New**.
3. Select **Apex Class**.
4. Page Name = **NewCaseListController**.
5. The custom controller Apex class must be named NewCaseListController and include the following:
 - A publicly scoped method named getNewCases | Use the return type of List<Case>.
 - Return a list of case records that includes the ID and CaseNumber fields.
 - Filter the results returned to only have a status of New.



```
File Edit Debug Test Workspace Help < >
OppView.vfp AccountList.vfp NewCaseList.vfp NewCaseListController.apxc
Code Coverage: None API Version 57 Go To
1+ public with sharing class NewCaseListController {
2
3+   public List<Case> getNewCases() {
4     List<Case> cases = [SELECT Id, CaseNumber FROM Case WHERE Status = 'New'];
5     return cases;
6   }
7
8 }
```

6. The NewCaseList Visualforce page must use an apex:repeat component, which is: Bound to newCases, Refers to the var attribute as case, bind an apex:outputLink component to the ID of the case.



```
File Edit Debug Test Workspace Help < >
OppView.vfp AccountList.vfp NewCaseList.vfp
Preview API Version 57
1+ <apex:page controller="NewCaseListController">
2+   <apex:pageBlock title="New Cases">
3+     <apex:pageBlockTable value="{!newCases}" var="case">
4+       <apex:column headerValue="Case Number">
5         <apex:outputLink value="/{!case.Id}">{!case.CaseNumber}</apex:outputLink>
6       </apex:column>
7     </apex:pageBlockTable>
8   </apex:pageBlock>
9 </apex:page>
```

7. Click on **Preview** button.
8. It shows the **New Cases** list.

Case Number	Status	Type
00001002	Open	
00001016	Open	
00001024	Open	

Activate Windows
Go to Settings to activate Windows.

9. Click it any of case.
10. It redirect to New Case list in Org.

Case Owner	Status
VEMPALLI MAHA MMAD AFROZ	New

Case Number	Priority
00001002	Low

Contact Name	Contact Phone
Stella Pavlova	(212) 842-5500

Account Name	Contact Email
United Oil & Gas Corp.	spavlova@uog.com

Type	Case Origin

Future Enhancements

There are several future enhancements that can be considered for the Travel Approval App project, including:

1. **Mobile Optimization:** As more employees are working remotely, it is important to optimize the Travel Approval App for mobile devices. This will allow users to easily submit travel requests and receive approvals from anywhere, at any time.
2. **Expense Management Integration:** Integrating the Travel Approval App with an expense management system can automate the expense reporting process and ensure that expenses are accurately tracked and reimbursed.
3. **Automated Notifications:** The Travel Approval App can be enhanced to provide automated notifications to approvers and travelers, keeping them informed of the status of their requests and any changes to their itinerary.
4. **Machine Learning and Artificial Intelligence:** The Travel Approval App can leverage machine learning and artificial intelligence to provide personalized travel recommendations based on traveler preferences and booking history. This can improve the user experience and increase adoption of the app.
5. **Real-Time Travel Alerts:** The Travel Approval App can be enhanced to provide real-time travel alerts, such as flight delays or cancellations, allowing travelers to make alternative arrangements quickly and easily.
6. **Integration with Additional Travel Providers:** The app can be integrated with additional travel providers to expand the options available to travelers and ensure they are getting the best prices.

Overall, these future enhancements can help the Travel Approval App project to continue providing value to the organization, improving the user experience, and simplifying the travel approval process.

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](#)
4. "Building a Travel Approval Process in Salesforce" by Andrew Smith, Salesforce Ben, <https://www.salesforceben.com/building-a-travel-approval-process-in-salesforce/>
5. "Travel Approval App for Salesforce" by Lightening App Builder, AppExchange, <https://appexchange.salesforce.com/appxListingDetail?listingId=a0N3A00000EJGzFUAX>
6. "How to Build a Salesforce App in 5 Steps" by The App Shack, The App Shack Blog, <https://www.theappshack.co.uk/how-to-build-a-salesforce-app-in-5-steps/>
7. "Creating a Travel Approval App on Salesforce" by Priya Ravindranath, Algoworks, <https://www.algoworks.com/blog/creating-a-travel-approval-app-on-salesforce/>
8. "Top Salesforce Apps for Travel and Hospitality" by Cloufi, Cloufi Blog, <https://www.cloufi.com/blog/top-salesforce-apps-for-travel-and-hospitality/>