



Capstone Project Report On “Travel App”

Submitted by:

Name: Sai Rishitha Mangipudi

Batch: WIP-SF-13

LMS Id: MGSA_513

Date: 28/02/2023

GitHub Link: <https://github.com/Sai-Rishitha-Mangipudi/Projects.git>

Table of contents

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

Abstract

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

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

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

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

Introduction

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

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

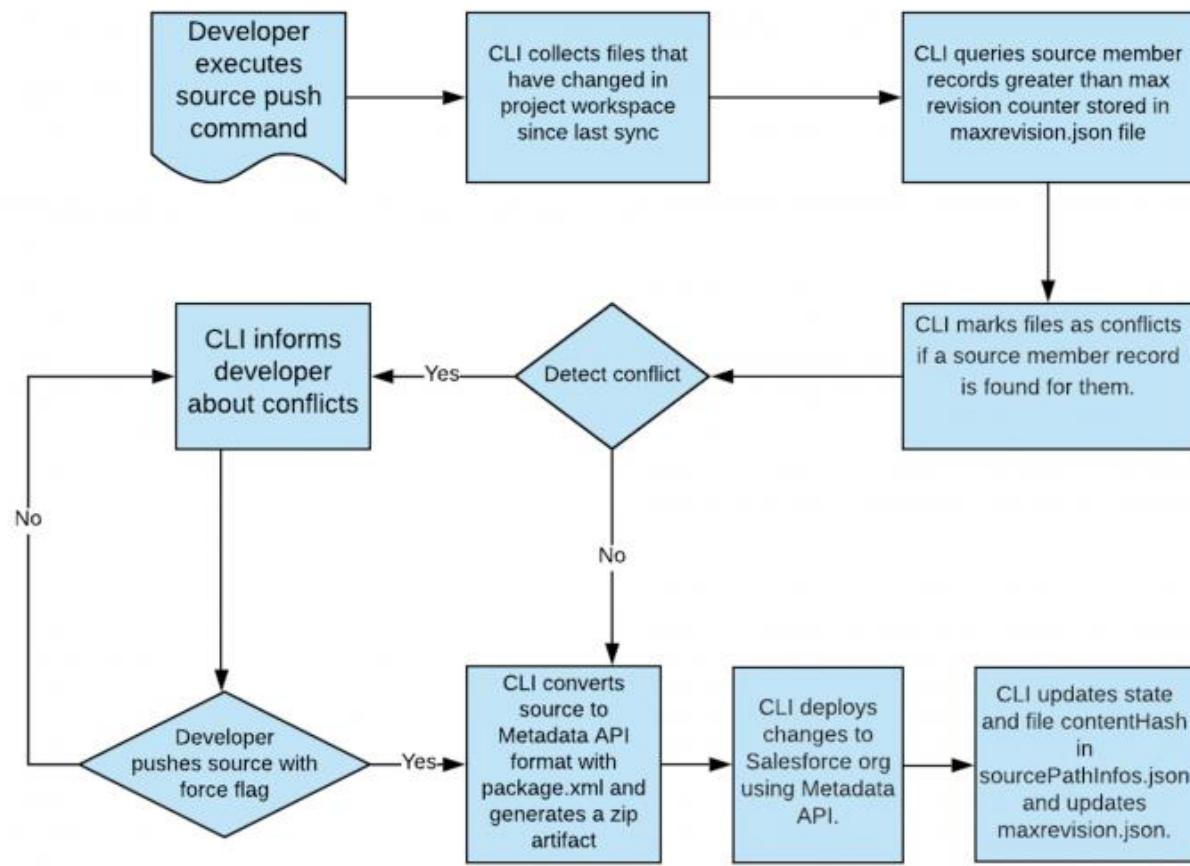


Salesforce developers can make an application on the cloud and share it with multiple companies across multiple domains by using Salesforce.

Talking about HR systems, every company across the globe has an HR team. Each HR team would require an HR application to store employee records. Almost all specifications for such an application would be common for all companies. So, as a developer, it would be very easy to create a Salesforce application for such specifications, post it onto the cloud, and provide it as a service to multiple clients.

at the same time. Maintenance of the same can be done altogether too. So basically, the problem of scalability gets eliminated.

Flow of the Project



Software Requirements

For the fastest and most stable experience, we recommend:

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

Minimum requirements are:

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

OR

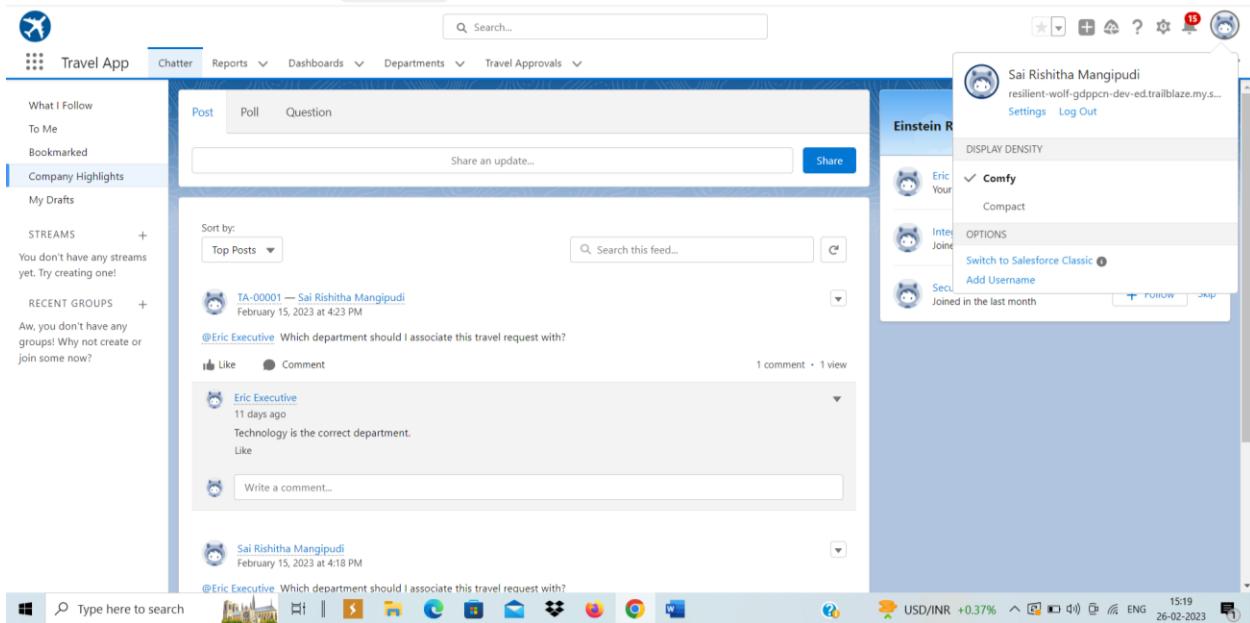
Requirements	
Windows	
Operating system	Windows 8.1 64-bit, Windows 8 64-bit, Windows 7 Service Pack 1 64-bit, Windows Vista Service Pack 2 64-bit
CPU	Core 2 Quad Q6600 at 2.4 GHz or AMD Phenom 9850 at 2.5 GHz
Memory	4 GB RAM
Free space	65 GB of free space
Graphics hardware	DirectX 10-compatible GPU: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

Module-1

Step-1:

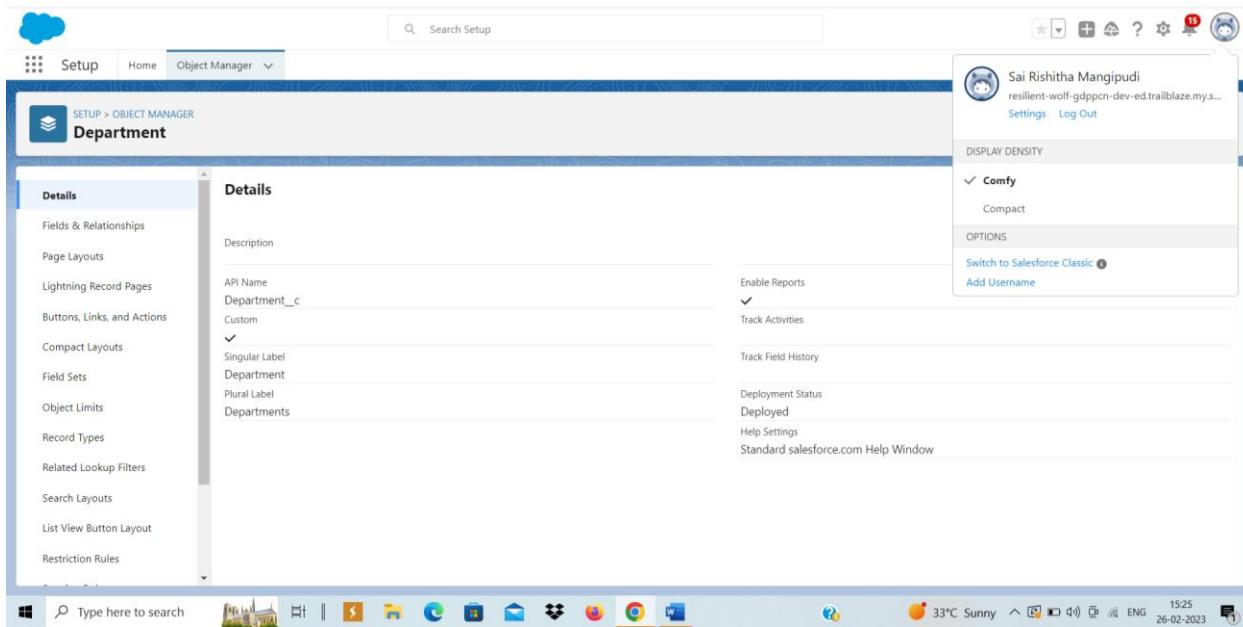
Created a new custom lightning App named, Travel App. The User Profile was set to “System Administrator”. Chatter, Reports & Dashboards were selected and added to the Navigation items. Navigation Style selected was “Standard Navigation’. Travel.png image from the downloaded zip file was uploaded as App image, keeping “Org Theme options” checked.

Test The App:

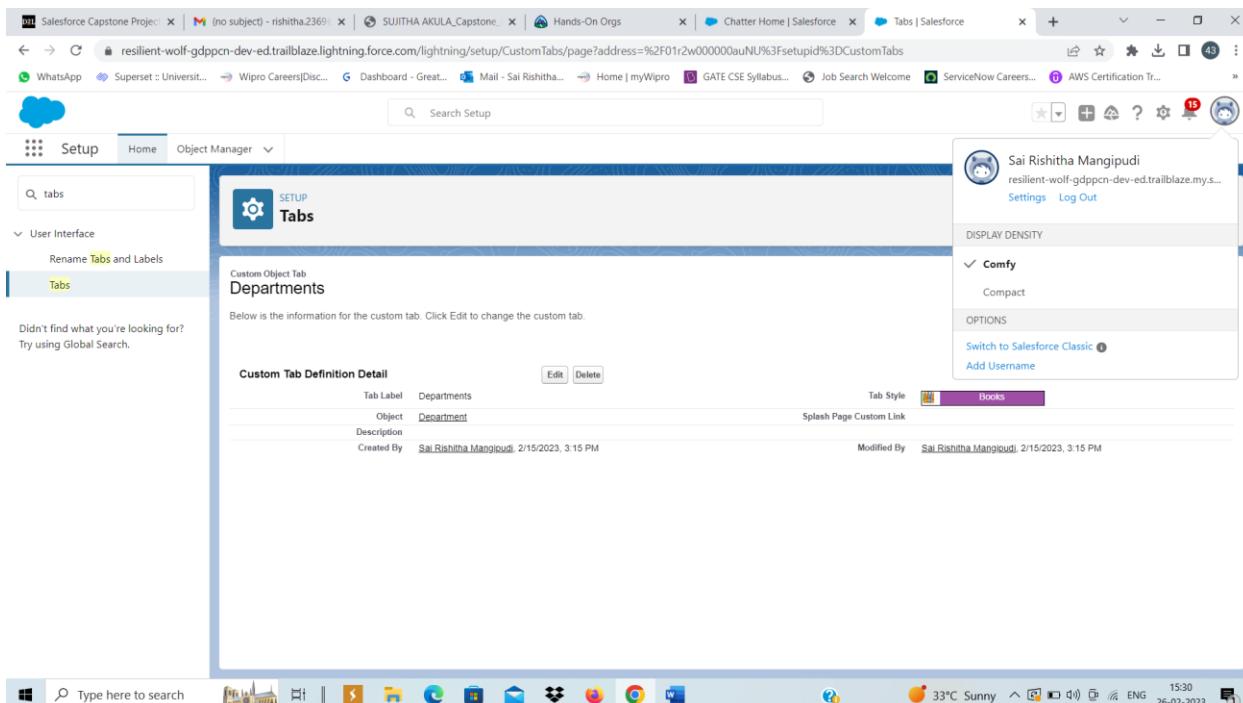


Step-2:

A custom object named “Department” was created with the credentials Label = “Department”, Plural = “Departments” ,Data Type = “Text”. “Allow Reports”, “Allow Search” checkboxes were selected.



A custom tab was created for this object. Custom tab Style “Books” was chosen and the tab was included only for “Travel App”.



Step-3:

Required Custom fields were created on the “Department” Custom Object.

Setup > OBJECT MANAGER Department

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING
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)	

The required field dependencies were created.

Setup > OBJECT MANAGER Department

Edit Field Dependency

Controlling Field	Location
Dependent Field	Department Type
Location:	Kolkata
Department Type:	Banking
	Finance
	Education
	Energy
	IT
	Delhi
	Banking
	Finance
	Education
	Energy
	IT

Legend

- Excluded Value
- Included Value

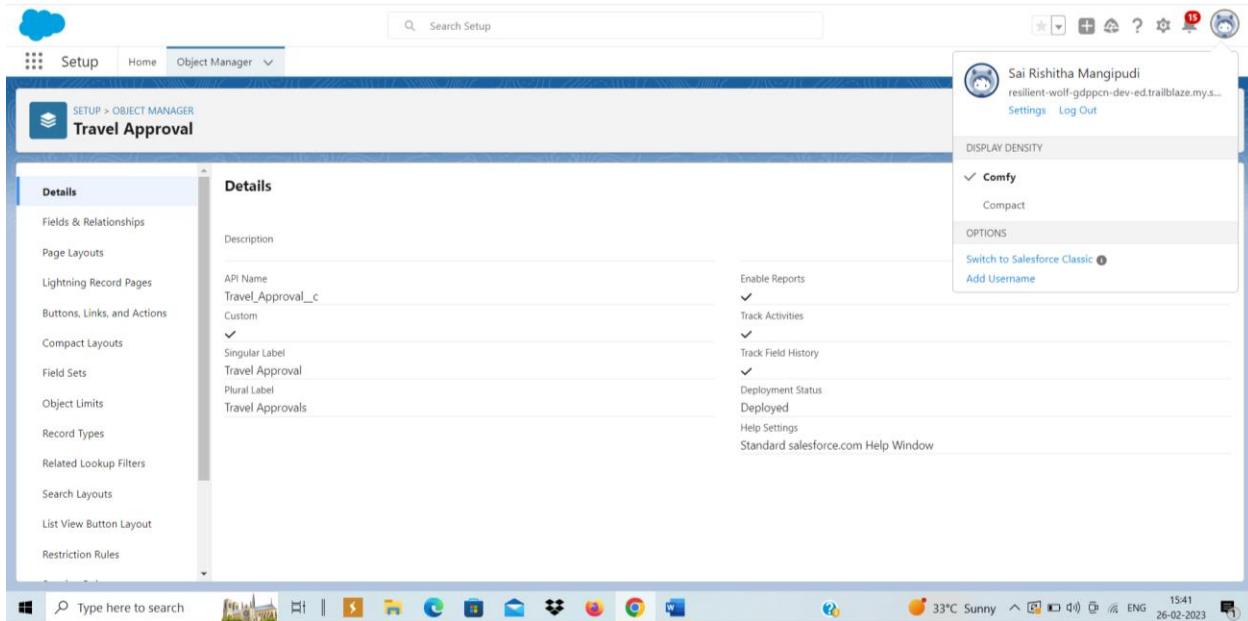
Instructions

- 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.

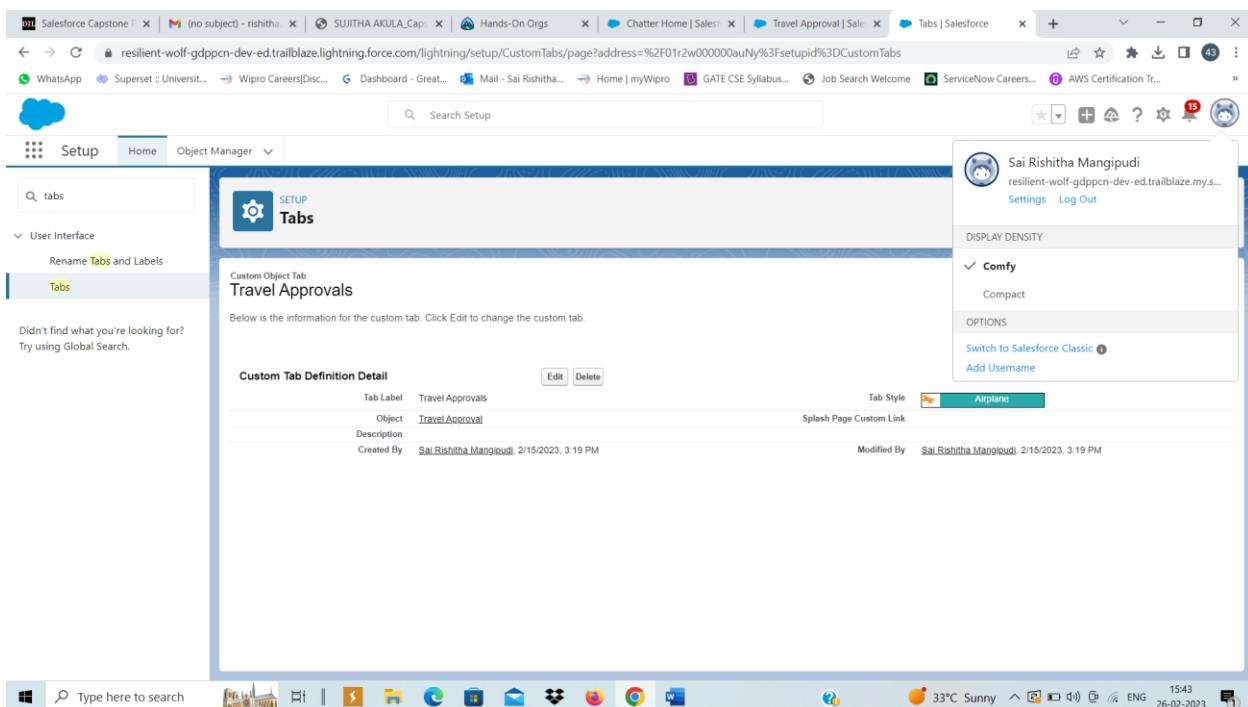
Step-4:

A “Travel Approval” Custom Object was created with 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.

Additionally, “Allow Reports”, “Allow Activities”, “Track Field History”, “Allow Search”, “Add Notes & Attachments” checkboxes were selected.



A Custom Tab for the Travel Approval object was created with style “Airplane” and the tab was added only to Travel App.



Custom fields were created on the Travel Approval Object as below, according to the requirement.

Field Name	Type	Description
Created By	Lookup(User)	CreatedById
Department	Lookup(Department)	Department_c
Destination State	Text(2)	Destination_State_c
Last Modified By	Lookup(User)	LastModifiedById
Out-of-State	Checkbox	Out_of_State_c
Owner	Lookup(User,Group)	OwnerId
Purpose of Trip	Text Area(255)	Purpose_of_Trip_c
Status	Picklist	Status_c
Status Indicator	Formula (Text)	Status_Indicator_c
Total Expenses	Roll-Up Summary (SUM Expense Item)	Total_Expenses_c

Test The App:

Travel App now looks like this.

Step-5:

The Department.csv file was downloaded. Through “Data Import Wizard”, in

“Launch Wizard”, “Departments” Custom Object was selected and “Add New Records” was chosen . The file was uploaded by clicking the CSV icon. The Fields were mapped, and Imported. All the records were inserted.

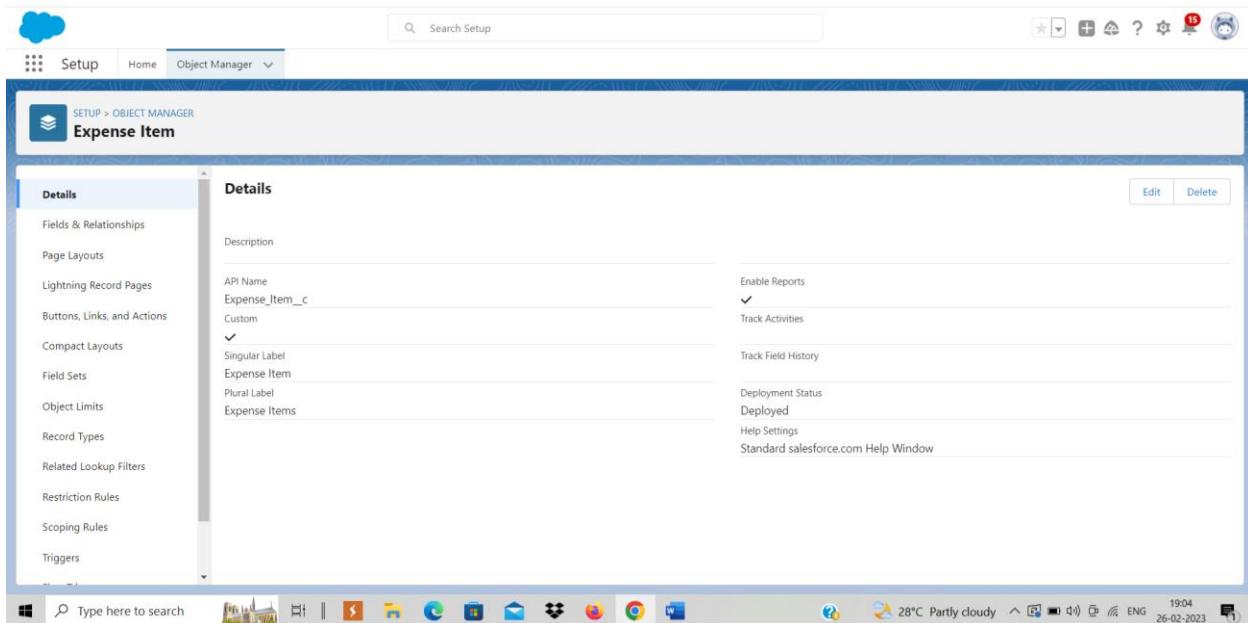
EXERCISE-2:

Step-1

A Travel Approval record was created with Purpose of Trip = “Attend Dreamforce”, Status = “Draft”, Trip Start Date/End Date = “22-2-2022”/”28-02-2022” , Out of state = True, Destination State = CA, Department = Technology.

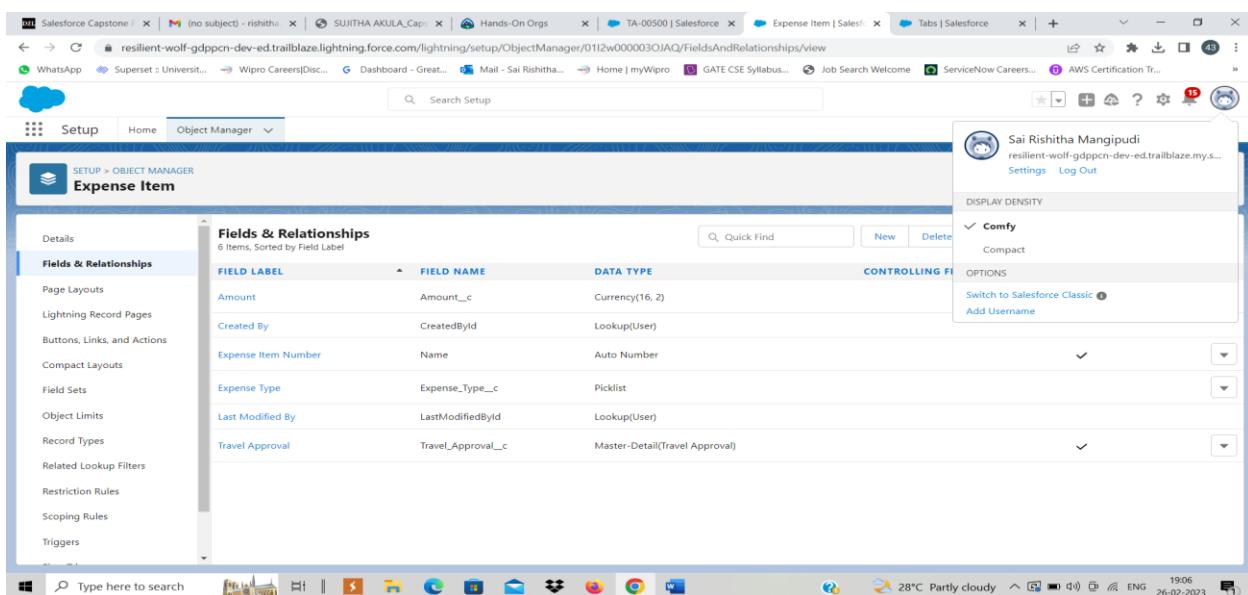
Step-2:

An “Expense Item” Custom Object was created with Label = “Expense Item”, Plural = “Expense Items”, ”Starts with vowel sound” was checked and Record Name = Expense Item Number, Data Type = Auto Number, Display Format = E – {00000},Starting Number = 1. Also, “Allow Reports” checkbox was selected.



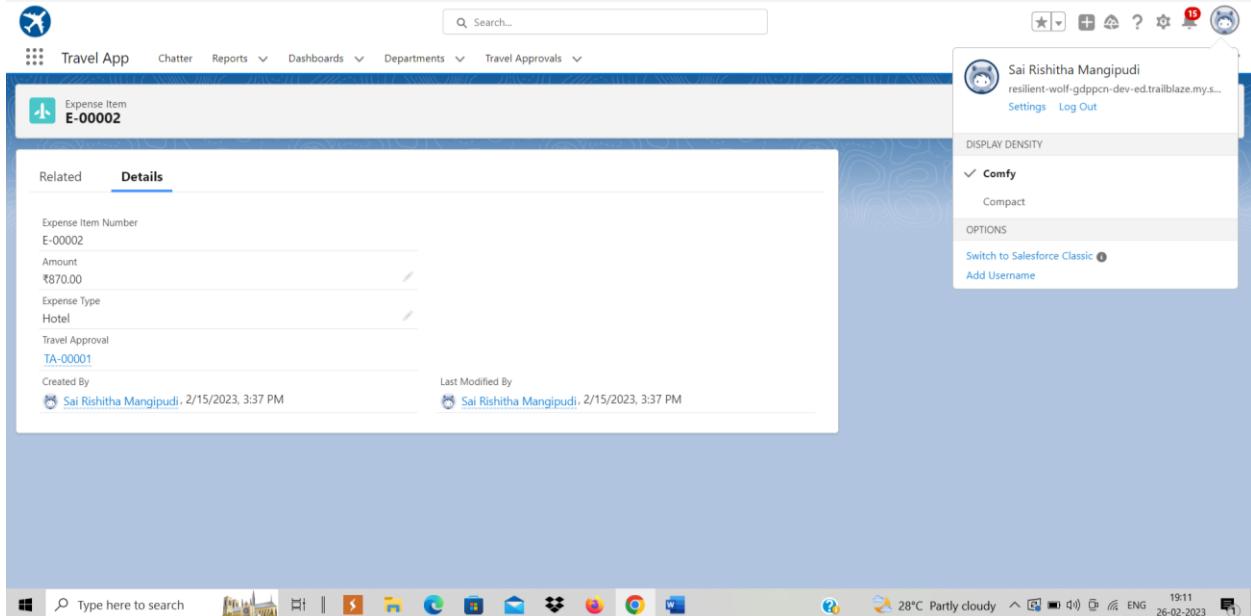
Step-3:

The required custom fields were created.



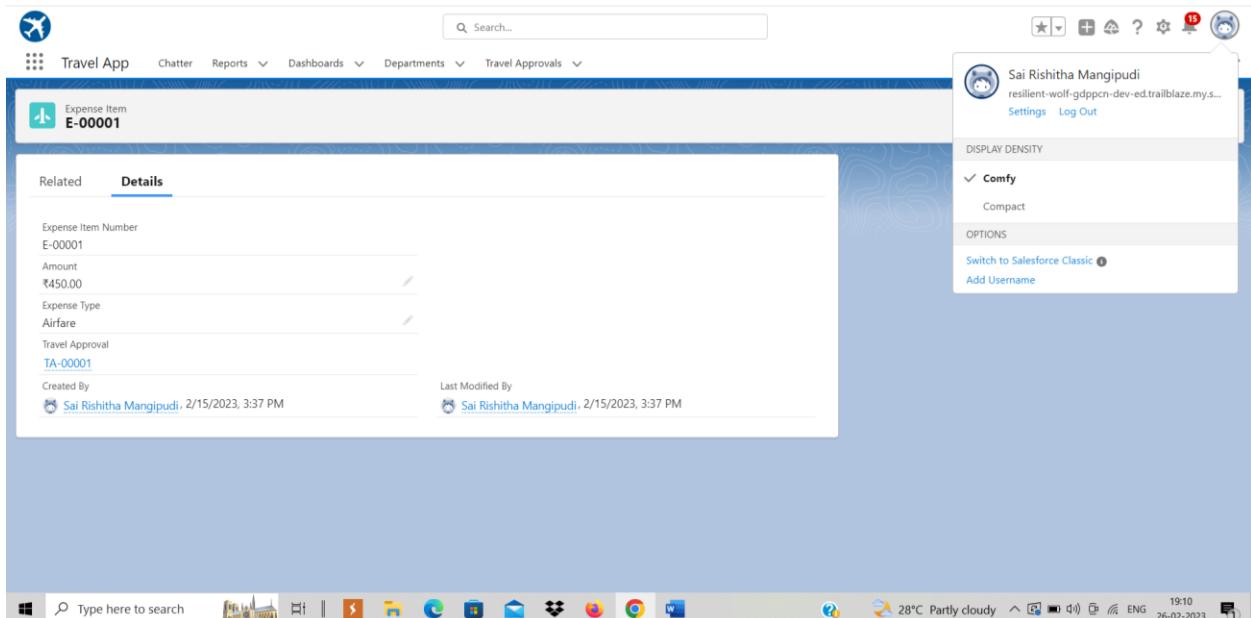
Step-4:

Two Expense Items were created with the following, Amount = 870, Expense Type = “Hotel” and Amount = “450”, Expense Type = “Airfare”.



Screenshot of the Travel App showing the creation of an expense item E-00002. The details section shows:

- Expense Item Number: E-00002
- Amount: ₹870.00
- Expense Type: Hotel
- Travel Approval: TA-00001
- Created By: Sai Rishitha Mangipudi, 2/15/2023, 3:37 PM
- Last Modified By: Sai Rishitha Mangipudi, 2/15/2023, 3:37 PM



Screenshot of the Travel App showing the creation of an expense item E-00001. The details section shows:

- Expense Item Number: E-00001
- Amount: ₹450.00
- Expense Type: Airfare
- Travel Approval: TA-00001
- Created By: Sai Rishitha Mangipudi, 2/15/2023, 3:37 PM
- Last Modified By: Sai Rishitha Mangipudi, 2/15/2023, 3:37 PM

Step-5:

A User with First Name = “Eric”, Last Name = “Executive”, Email = “sai.mangipudi2@wipro.com”, Username Name, Role = “CEO”, License = Salesforce, Profile = “System Administrator” was created.

The screenshot shows the Salesforce Setup interface under the 'Users' section. A new user record is being created for 'Eric Executive'. The 'User Detail' section displays the following information:

Name	Eric Executive	Role	CEO
Alias	exec	User License	Salesforce
Email	sai.mangipudi2@wipro.com	Profile	System Administrator
Username	eric.mangipudi@wipro.com	Active	<input checked="" type="checkbox"/>
Nickname	eric.exec	Marketing User	<input type="checkbox"/>
Title		Offline User	<input type="checkbox"/>
Company		Knowledge User	<input type="checkbox"/>
Department		Flow User	<input type="checkbox"/>
Division		Service Cloud User	<input type="checkbox"/>
Address		Site.com Contributor User	<input type="checkbox"/>
Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)	Site.com Publisher User	<input type="checkbox"/>
Locale	English (India)	WDC User	<input type="checkbox"/>
Language	English	Mobile Push Registrations	<input type="checkbox"/> View
Delegated Approver		Data.com User Type	<input type="checkbox"/> View
Manager	Eric Executive	Accessibility Mode (Classic Only)	<input type="checkbox"/> View
Receive Approval Request Emails	Only if I am an approver	Debug Mode	<input type="checkbox"/> View
Federation ID		High-Contrast Palette on Charts	<input type="checkbox"/> View

The right sidebar shows the user's profile picture and details: 'Sai Rishitha Mangipudi' from 'resilient-wolf-gdppcn-dev-ed.trailblaze.my.s...'. It also includes options like 'DISPLAY DENSITY' (set to 'Comfy'), 'OPTIONS' (with 'Switch to Salesforce Classic' and 'Add Username' buttons), and system status indicators.

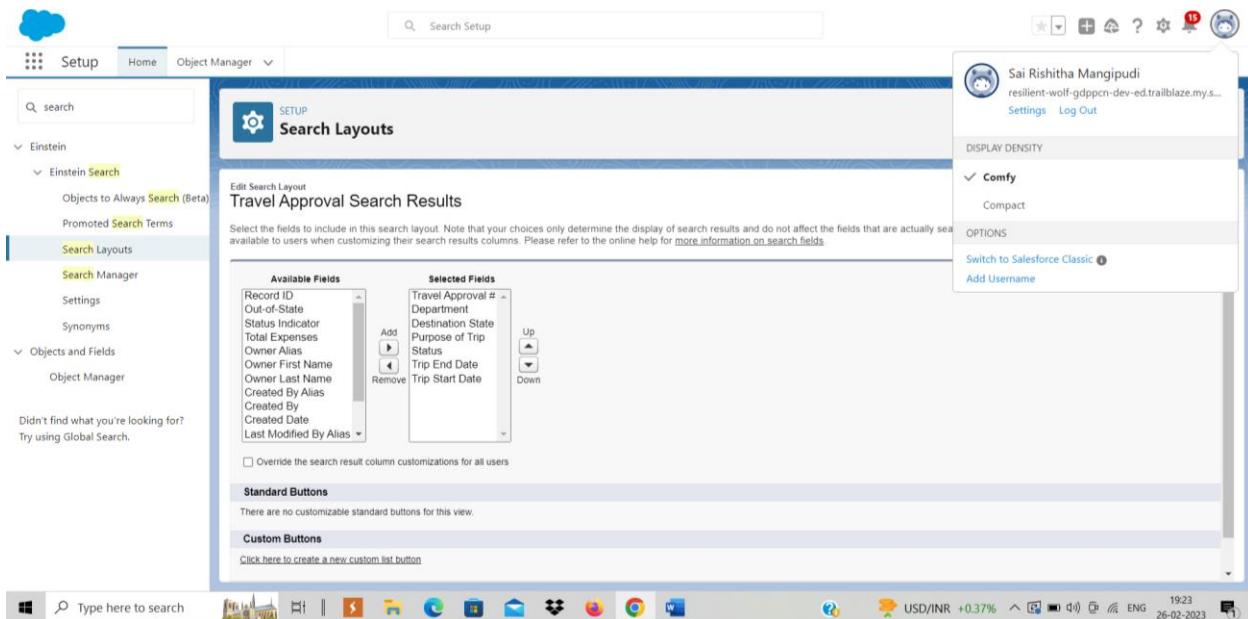
Step-6:

Eric Executive was added as Manager to System Administrator profile.

The screenshot shows the Salesforce Setup interface under the 'Users' section. The user 'Sai Rishitha Mangipudi' is being edited. In the 'User Detail' section, under the 'Manager' field, 'Eric Executive' is listed as the manager. The rest of the user details for 'Sai Rishitha Mangipudi' are identical to the previous user record.

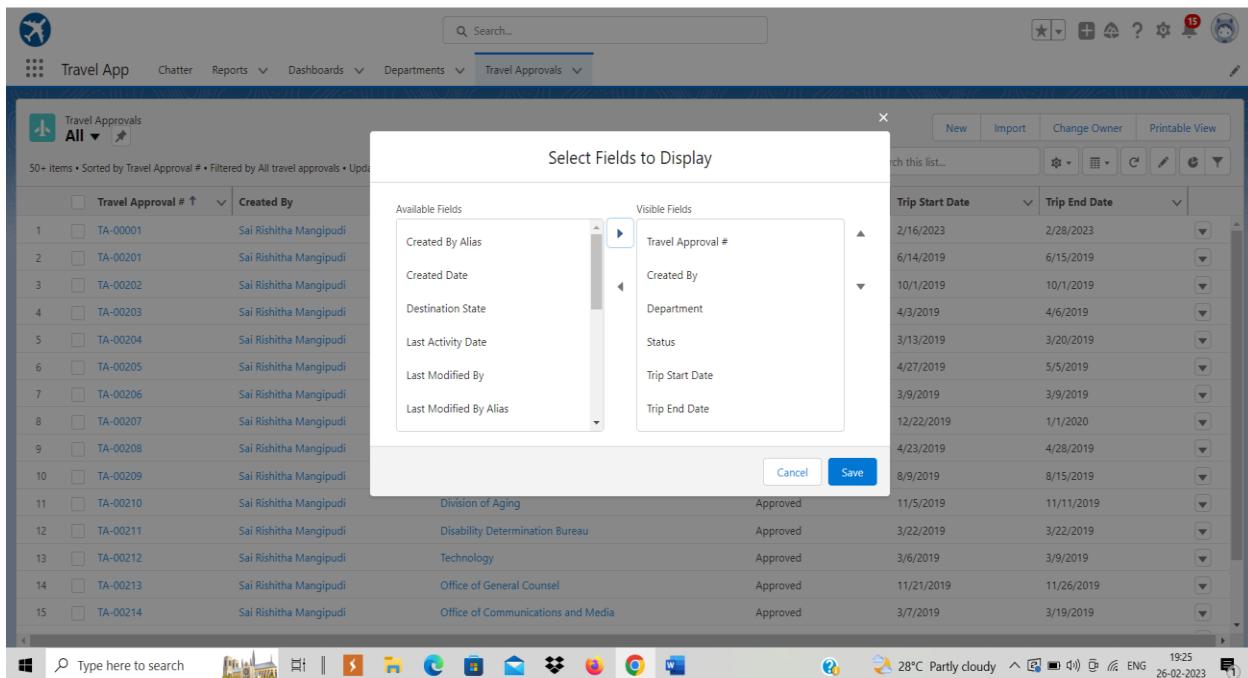
Step-7:

Travel Approval Default search layout was modified based on requirement as shown in the screenshot.



Step-8:

Selected fields to display in the Travel Approval “All” List view, as shown in the screen shot.



Step-9:

Created Travel approval custom List View “Open Out of State Travel Requests” as shown in the screen shot, so that All users were be able to see this list view.

The screenshot shows a Salesforce list view titled "Travel Approvals" with a specific filter applied: "Open Out of State Travel Requests". The list includes columns for Travel Approval #, Department, Created By, Status, Destination State, Trip End Date, and Trip Start Date. A single item is listed: TA-00500 from the Technology department, created by Sai Rishitha Mangipudi, with a Draft status, destination CA, end date 2/28/2023, and start date 2/22/2023. To the right of the list is a "Filters" sidebar showing the applied filters: "Out-of-State equals True" and "Status not equal to Approved, Rejected".

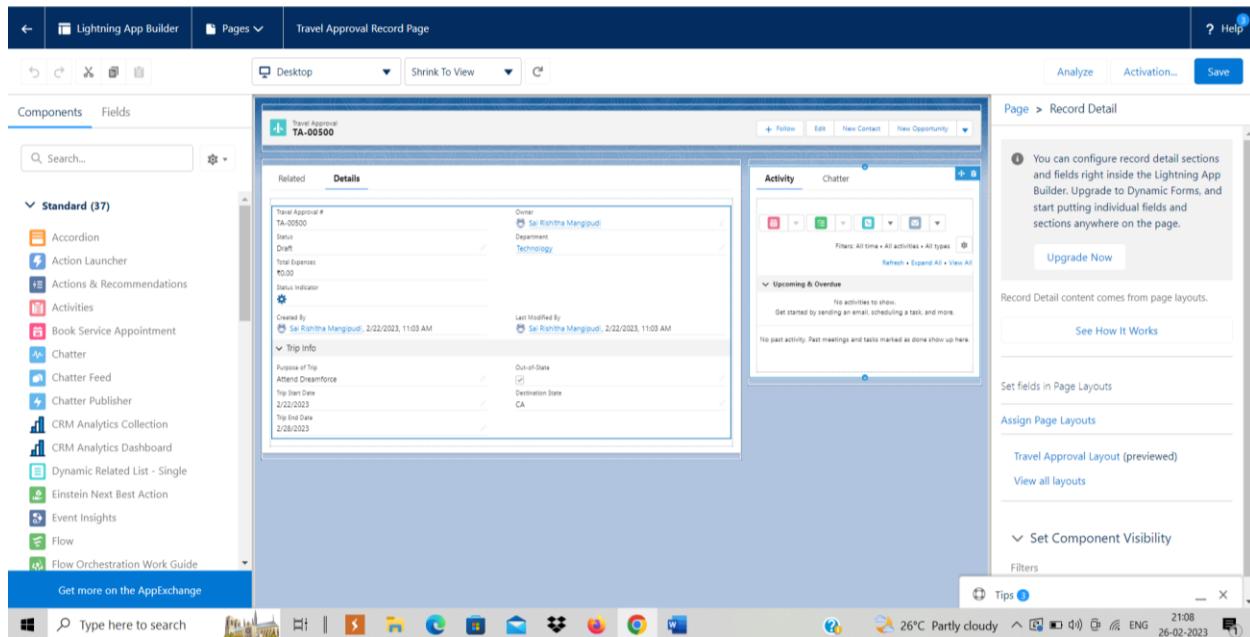
Step-10:

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

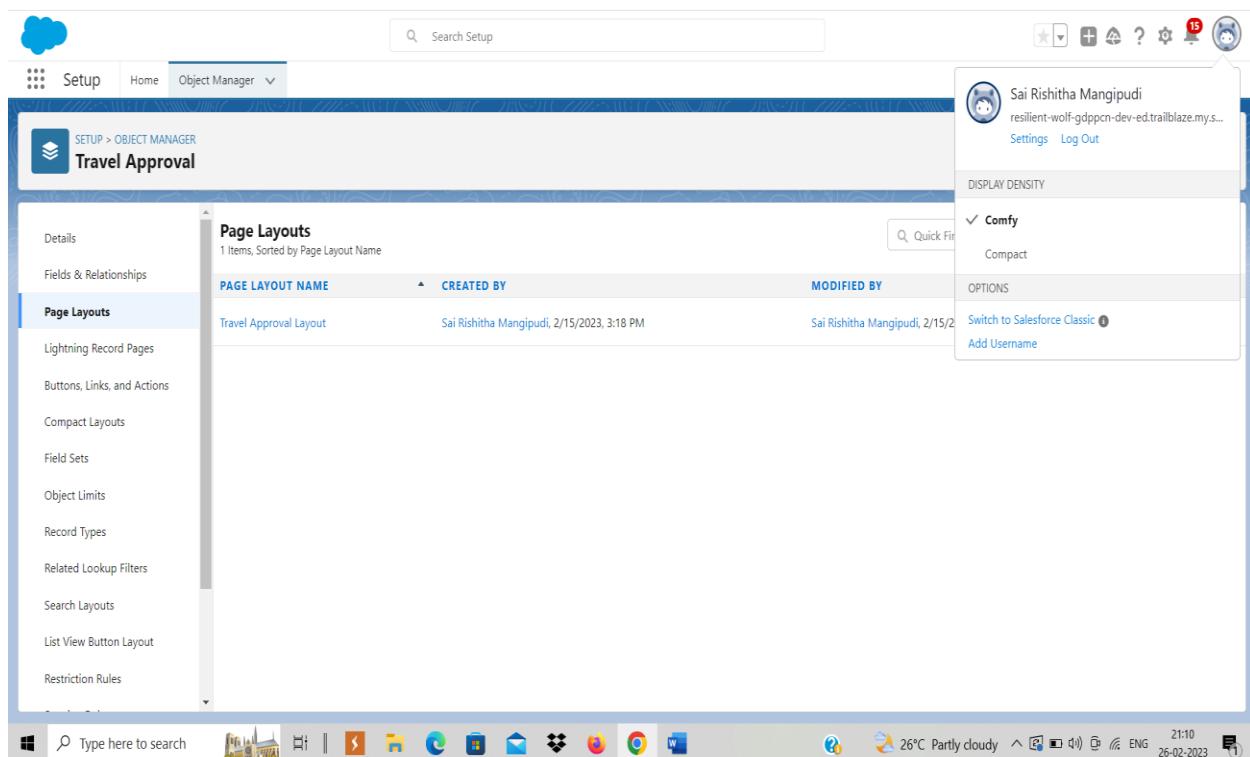
The screenshot shows the "Select Fields to Display" dialog box overlaid on the list view. The "Available Fields" section contains "Created By Alias", "Created Date", "Last Activity Date", "Last Modified By", "Last Modified By Alias", and "Last Modified Date". The "Visible Fields" section contains "Department", "Created By", "Status", "Destination State", "Trip End Date", and "Trip Start Date". The "Filters" sidebar on the right is identical to the one in Step 9, showing the same filters applied to the list view.

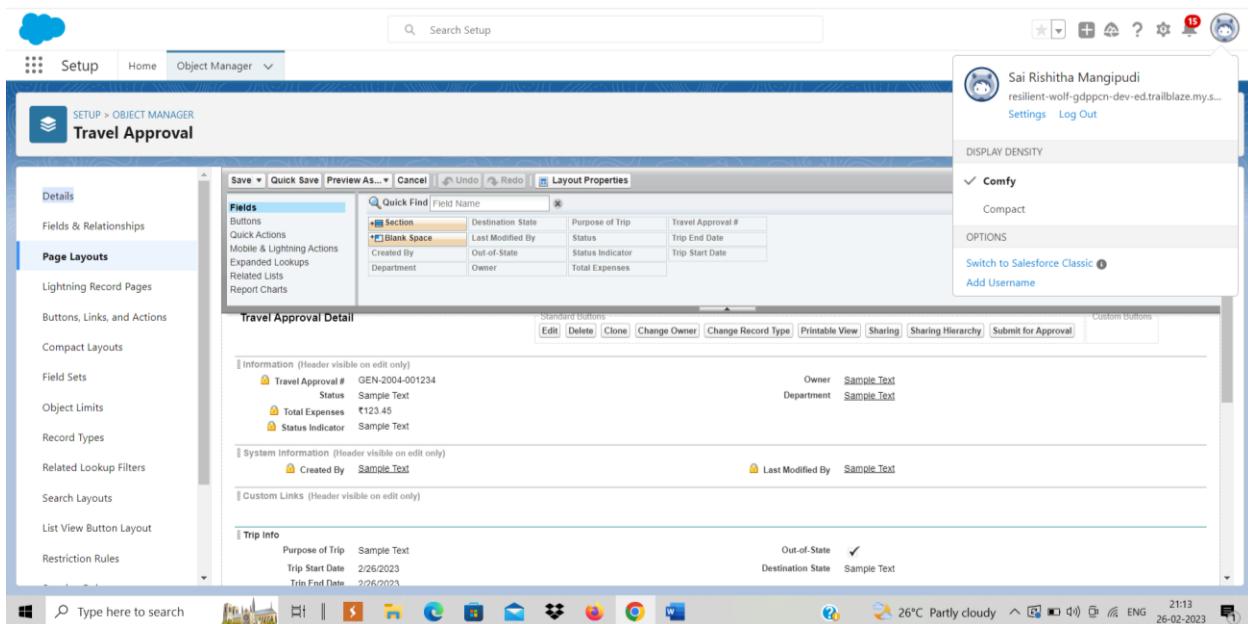
Step-11:

Travel Approval Page Layout has been customized.



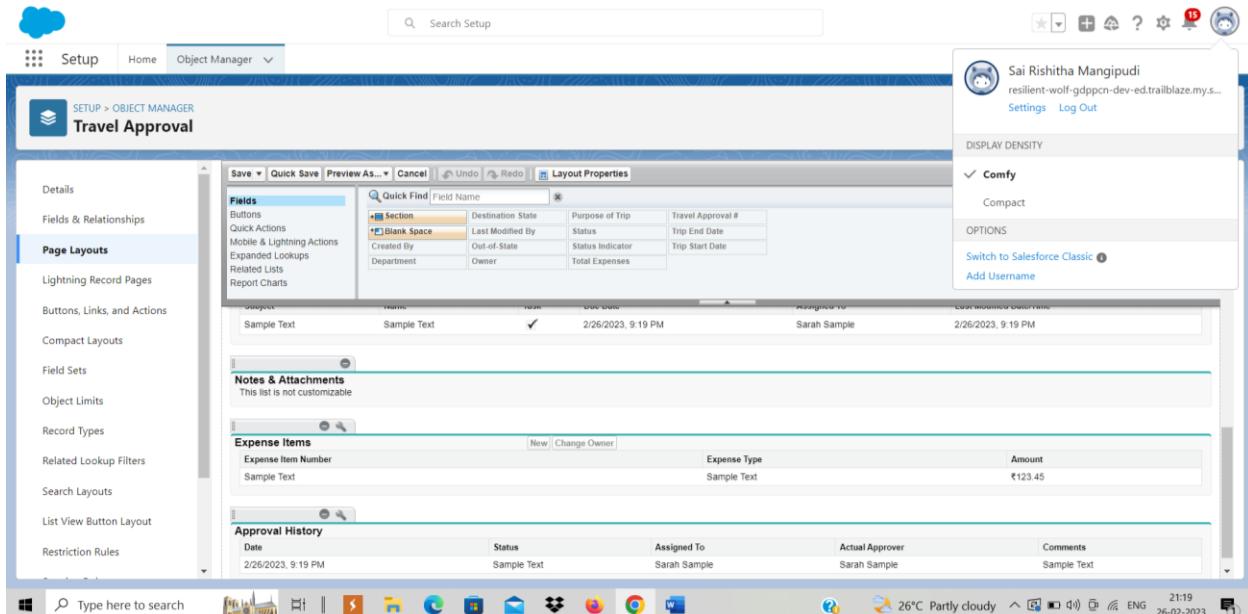
Travel Approval Page Layout has been customized by adding a section to the page layout called “Trip Info” and add the fields as shown.



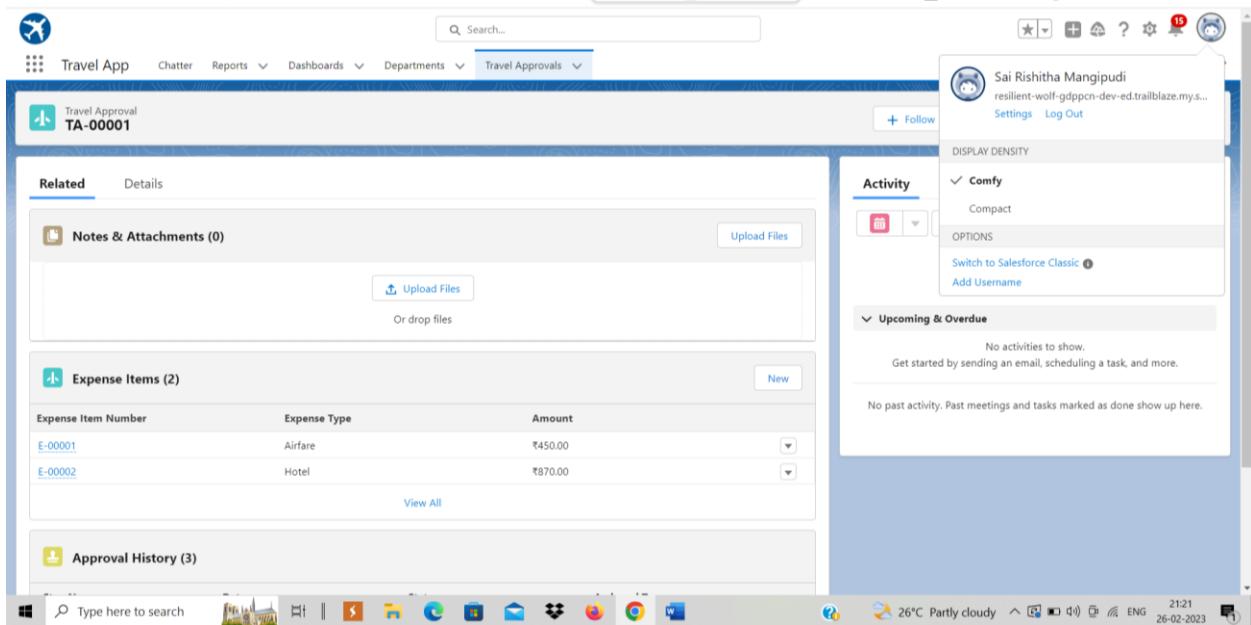


Step-12:

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

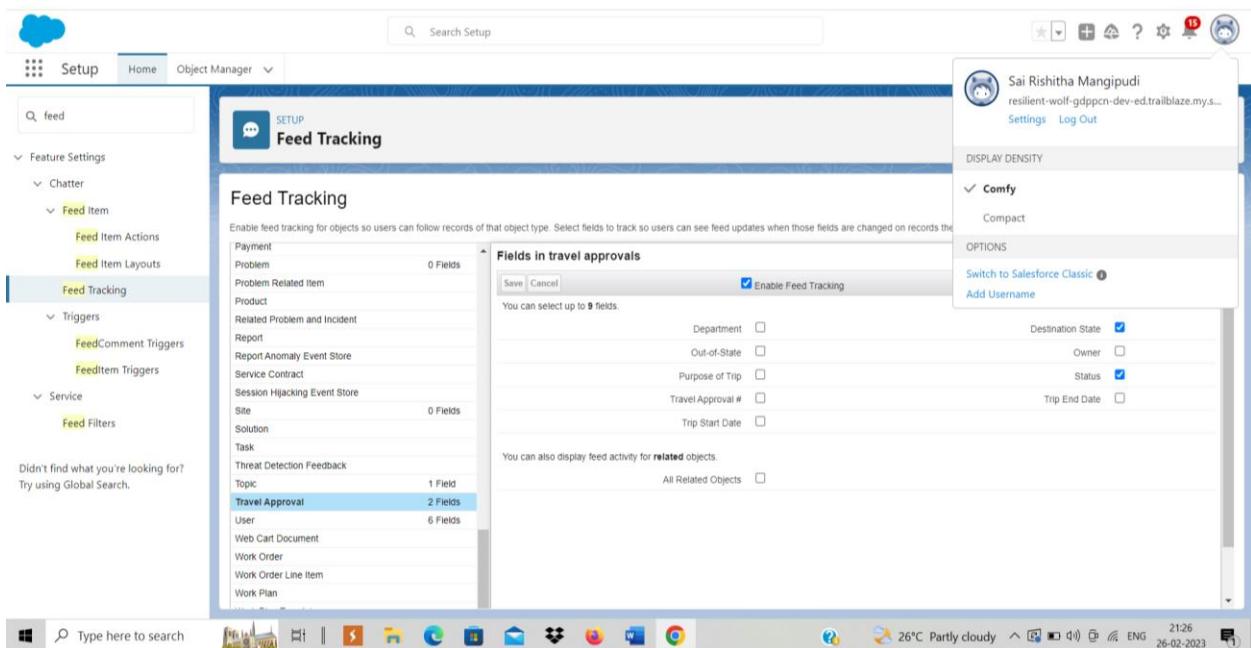


Test The App:



Step-13:

Chatter on the Travel Approval Object has been enabled. “Feed Tracking” for Travel Approval Object has been enabled and selected these 2 fields: Destination State, Status.



Test Collaboration:

The screenshot shows the Chatter feed of the Travel App. A user named @Eric Executive posted a question: "Which department should I associate this travel request with?". Another user, Eric-Executive, responded: "Technology is the correct department." The feed also shows a comment from @Sai Rishitha Mangipudi. The interface includes a sidebar with navigation links like 'What I Follow', 'My Drafts', and 'RECENT GROUPS'. The top navigation bar has tabs for 'Chatter', 'Reports', 'Dashboards', 'Departments', and 'Travel Approvals'.

Test The App:

The screenshot shows the details page for a Travel Approval record (TA-00001). The record is owned by Sai Rishitha Mangipudi and is associated with the Technology department. The 'Trip Info' section indicates the purpose of the trip is to attend Dreamforce, and the destination state is CA. The status of the approval is currently 'Approved'. The right side of the screen shows the Chatter feed for this record, which includes a post from the owner updating the status from 'Draft to Approved'.

Module-2

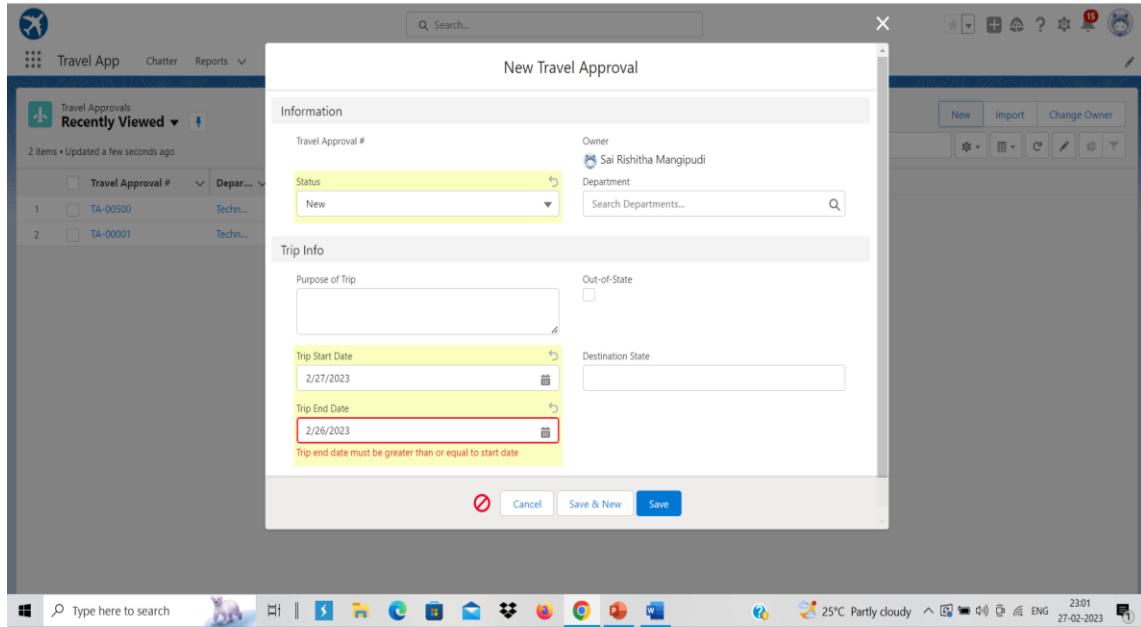
Step-1:

A Validation Rule with name “Trip end date after start date.” has been created in Travel Approval Object and the Error Message “Trip end date must be greater than or equal to start date” is displayed and “Active” checkbox is selected/checked checking for the business logic, Trip end date must always be greater than the trip start date.

The screenshot shows the Salesforce Object Manager interface for the 'Travel Approval' object. On the left, a sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, etc. The main content area displays the 'Travel Approval Validation Rule'. The rule details are as follows:

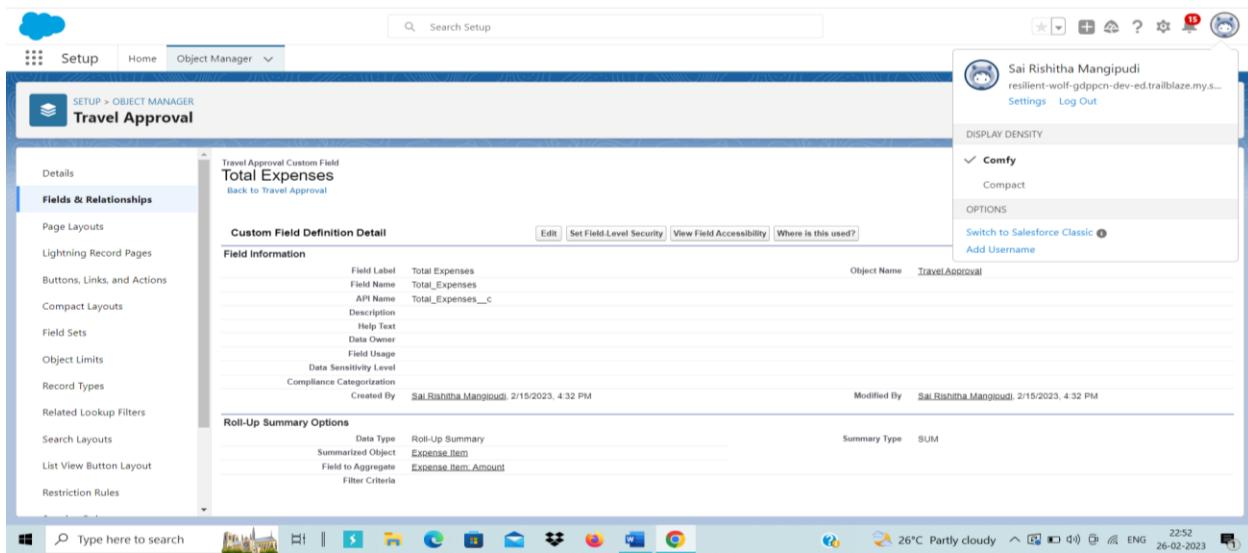
- Validation Rule Detail**
- Rule Name:** Trip_end_date_after_start_date
- Error Condition Formula:** Trip_End_Date__c < Trip_Start_Date__c
- Error Message:** Trip end date must be greater than or equal to start date
- Description:** (empty)
- Created By:** Sai Rishitha Mangipudi, 2/15/2023, 4:26 PM
- Active:** checked
- Error Location:** Trip End Date
- Modified By:** Sai Rishitha Mangipudi, 2/15/2023, 4:26 PM

The screenshot shows the 'Validation Rule Edit' screen for the 'Travel Approval' object. The validation rule is named 'Trip_end_date_after_start_date' and is set to Active. The error condition formula is defined as 'Trip_End_Date__c < Trip_Start_Date__c'. A tooltip for the formula explains: 'Display an error if Discount is more than 30%'. The formula builder dropdown shows various functions like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc. The right side of the screen includes a user profile for 'Sai Rishitha Mangipudi' and a sidebar with options like 'DISPLAY DENSITY' (Compact selected), 'OPTIONS' (Switch to Salesforce Classic, Add Username), and a 'Operators & Functions' link.



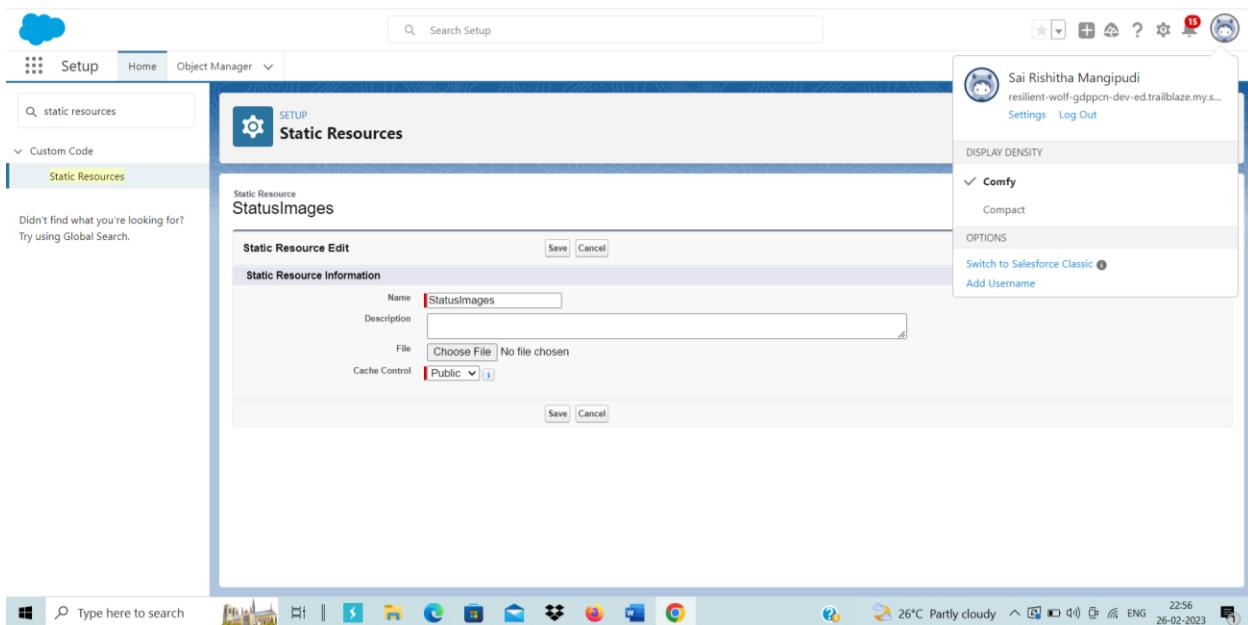
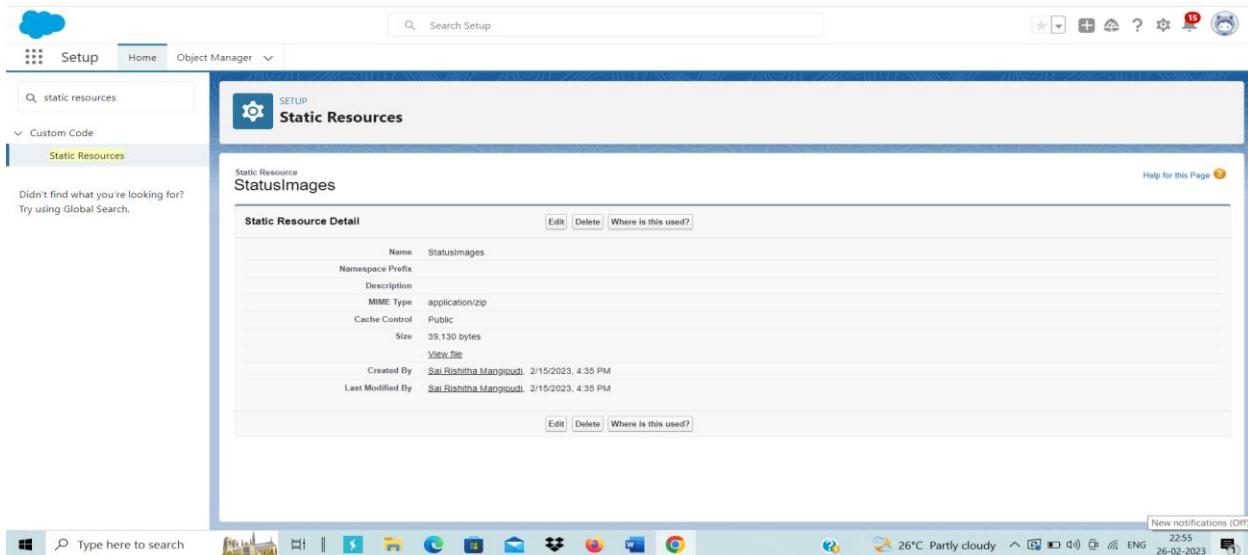
Step-2:

Created a Roll-Up Summary Field on Travel Approval object with Filter Criteria:
 All records should be included in the calculation, Field to Aggregate: Amount,
 Roll-Up Type: SUM, Field Name = Total_Expenses, Field Label = Total Expenses
 applying Business Logic: Automatically sum up the total amount of expenses from
 the related Expense Item.



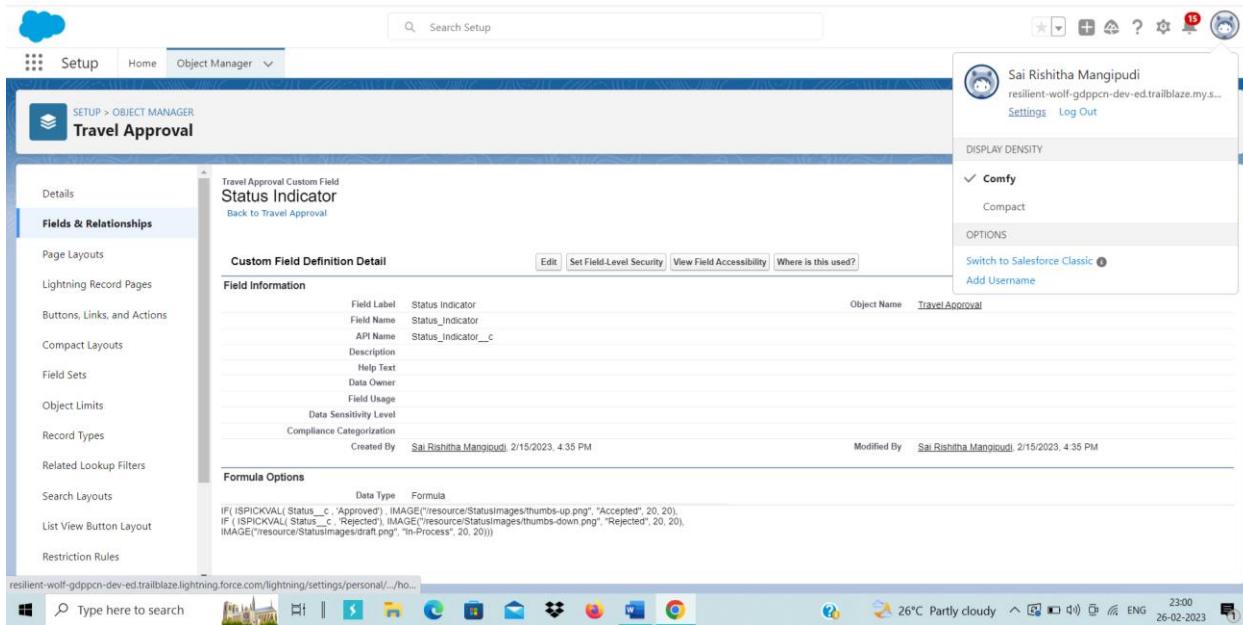
Step-3:

Created Formula Fields with Cache-Control = Public, File = StatusImages.png, Name = StatusImages.



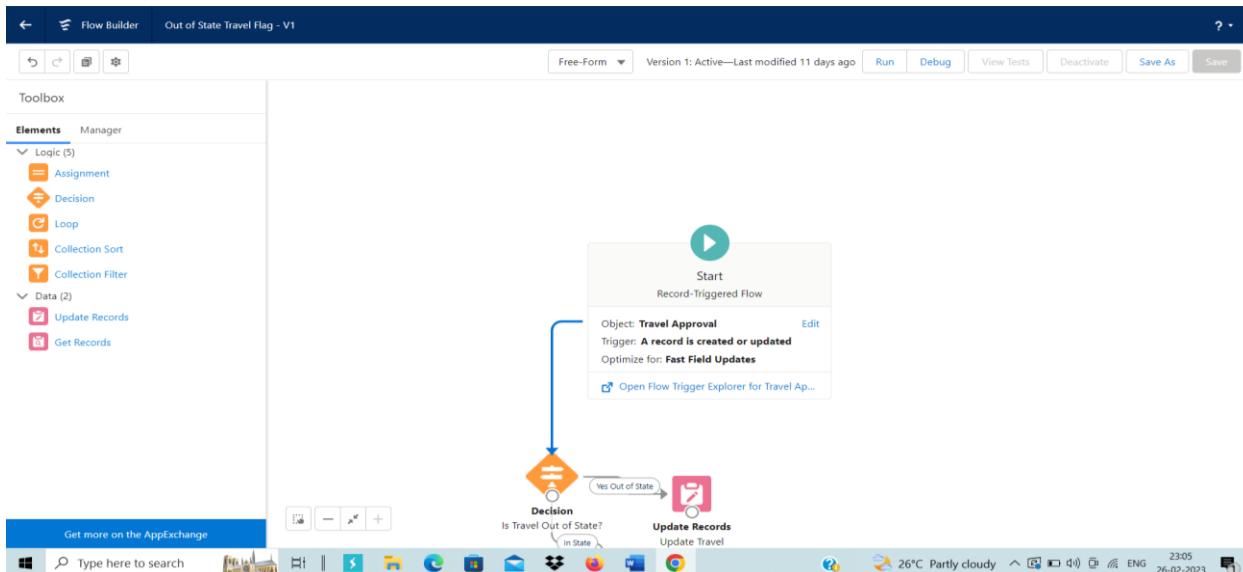
Step-4:

Created a Formula field on the Travel Approval object to show an image based on the Status field, Field Label: Status Indicator ,Formula Return Type = Text



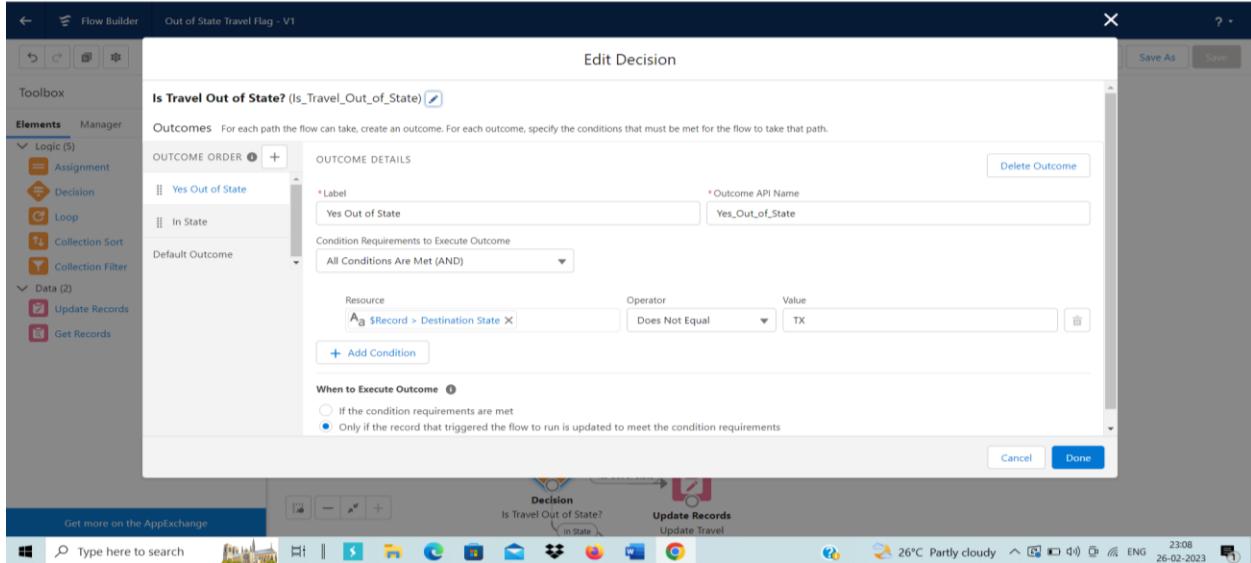
Step-5:

Created a Record – Triggered Flow, Set the Out-of-State checkbox field on the Travel Approval object if out-of-state travel has been chosen. Selected “Freeform”, Optimize the Flow For: Fast Field Updates, Configure Trigger: Trigger the flow when: A record is created or updated on the object “Travel Approval.”

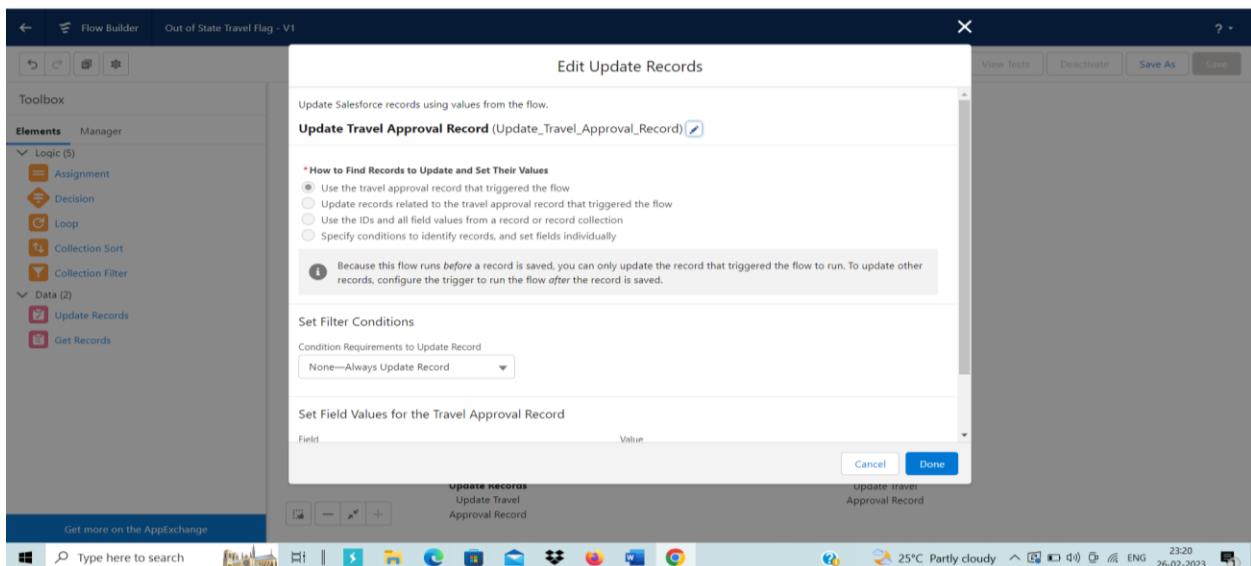


Added a Decision Element to the Flow, Label = Is Travel Out of State?, set the Outcomes, For the first outcome, set these parameters: Label = Yes Out of State, Condition Requirements = All Conditions Are Met (AND), Resource = \$Record >

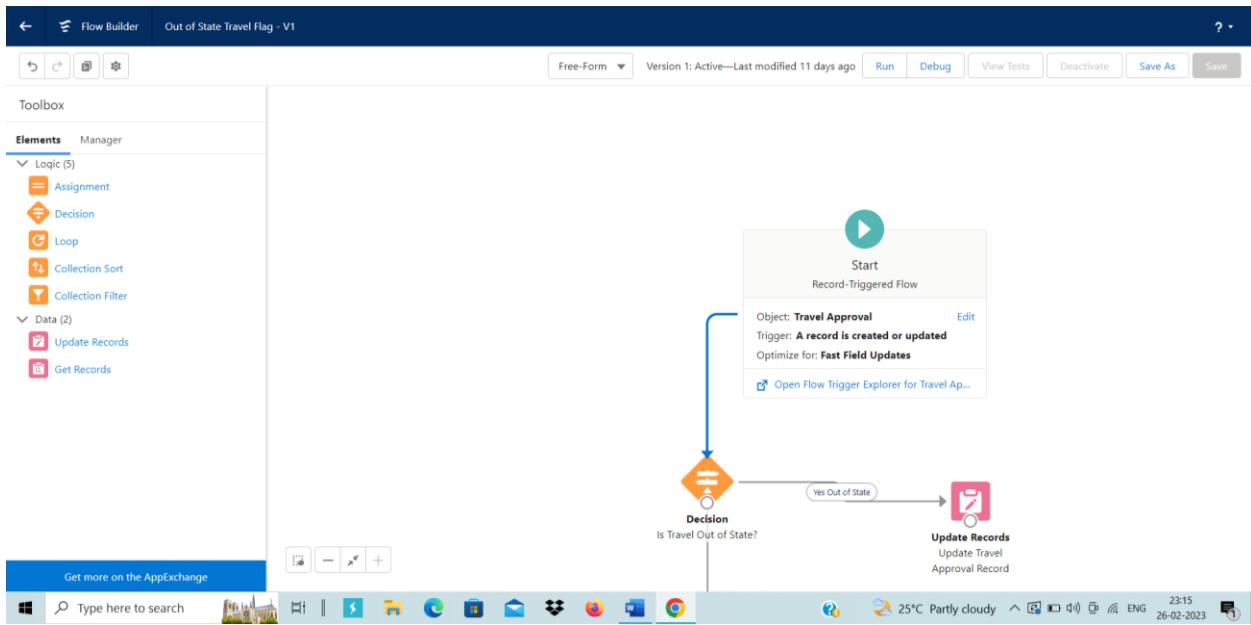
Destination State, Operator: Does Not Equal, Value: Tx, When to Execute the Outcome: Only if the record that triggered the flow to run is updated to meet the condition requirements.



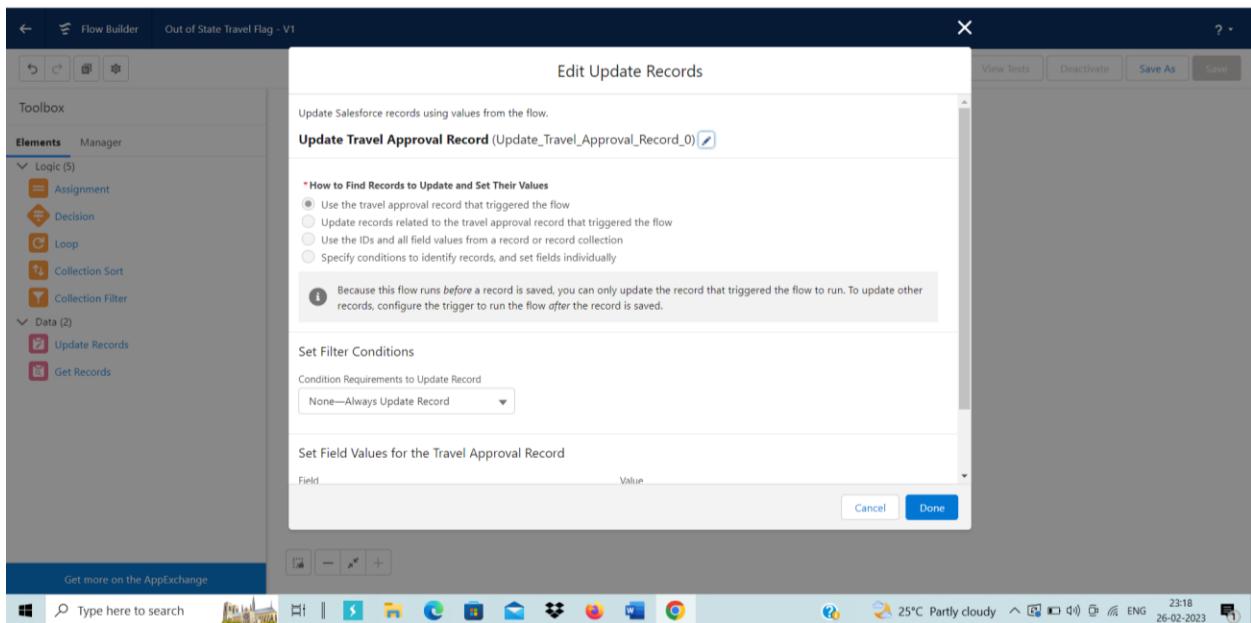
Created an Action for the Flow Using Update Records Elements. From the flow toolbox, dragged an Update Records element onto the flow screen, Label = Update Travel Approval Record , How to find Records = Use the travel approval record that triggered the flow, Condition Requirements to Update the Record = None—Always Update the Record, Field: Out_of_State__c, Value = \$GlobalConstant.True



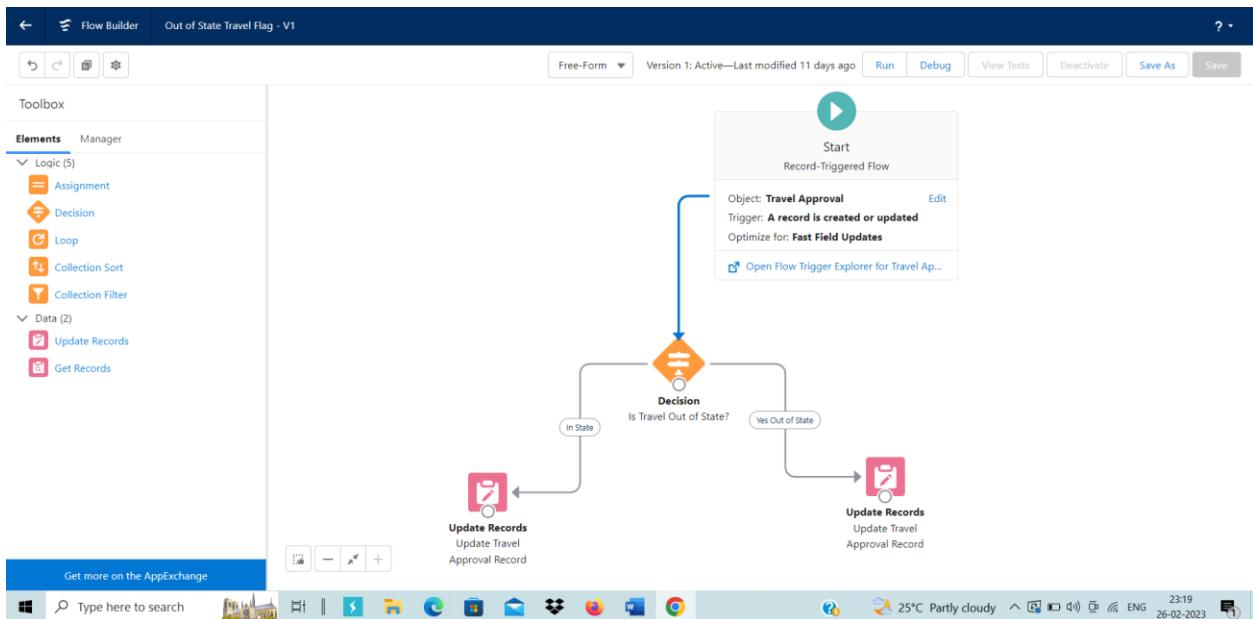
Test Your Screen:



Created an Action for the Flow Using Update Records Elements. From the flow toolbox, dragged an Update Records element onto the flow screen. Label = Update Travel Approval Record, How to find Records = Use the travel approval record that triggered the flow, Condition Requirements to Update the Record = None—Always Update the Record, Field: Out_of_State__c, Value = \$GlobalConstant.False

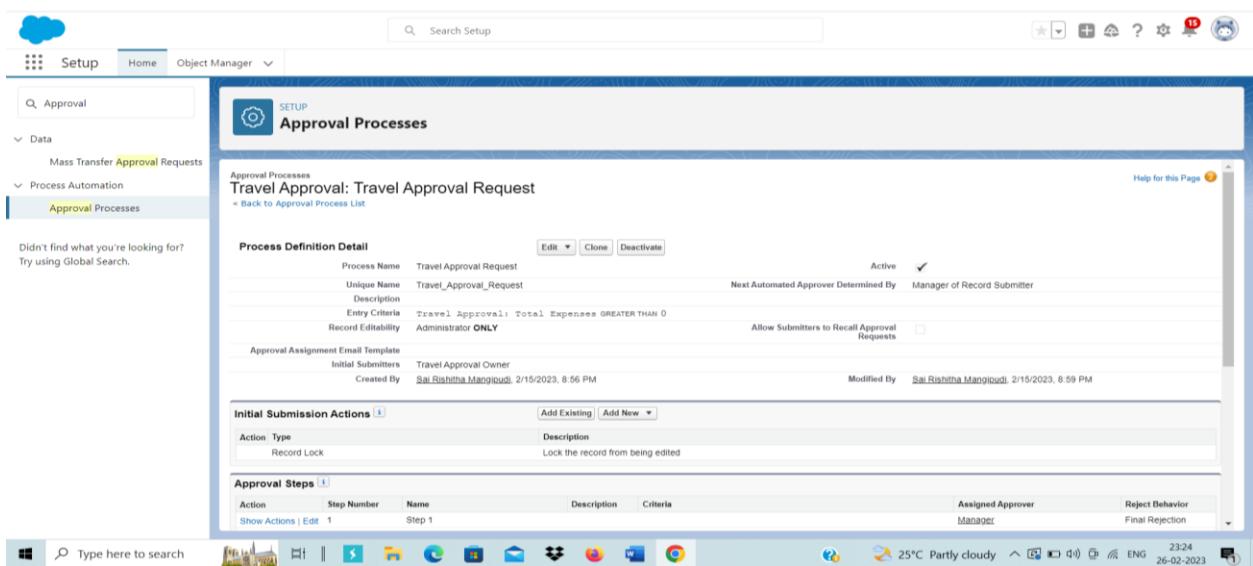


Test Your Screen:



Step-6:

Created an Approval Process to send Travel approvals to Manager or Travel coordinator with Approval Process Name: Travel Approval Request, Specify Entry Criteria: Use this approval process if the following: Formula Evaluates to True, Select Approver: Automatically assign an approver using a standard or custom hierarchy field: Manager



Created an Approval Step for Out-of-State Travel named Travel_Coordinator_Approval, Automatically assign to the approver(s): User selected Eric Executive as the travel coordinator with formula criteria: Field: Travel Approval: Out-of-State = True.Enter this step if the following: criteria are met ,Step Number: 2

Action	Type	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit	1	Step 1		Manager	Final Rejection
Show Actions Edit	2	Travel Coordinator Approval	Travel Approval: Out-of-State EQUALS True	User:Sai Rishitha Mangipudi	Final Rejection

Created Final Approval Action and Final Rejection Action based on the requirements

Action	Type	Description
Edit	Record Lock	Lock the record from being edited
Edit Remove	Field Update	Set Status to Approved

Test the Approval Process:

This screenshot shows a travel approval record in a CRM application. The record is for Travel Approval # TA-00500, which is currently in Draft status. The total expenses are listed as ₹0.00. The trip info indicates a purpose of trip to attend Dreamforce, starting on 2/22/2023 and ending on 2/28/2023. The record was created by Sai Rishitha Mangipudi on 2/22/2023 at 11:03 AM. The last modification was also made by her on the same date and time. The owner of the record is also Sai Rishitha Mangipudi, and the department is Technology. The Chatter panel on the right shows no activities or posts.

This screenshot shows the same travel approval record after it has been approved. The status is now Approved, and the total expenses are ₹1,320.00. The trip info remains the same. The record was updated by Sai Rishitha Mangipudi on 2/15/2023 at 3:37 PM. The last modification was made by her on 2/15/2023 at 10:55 PM. The owner and department are still the same. The Chatter panel on the right shows a post from Sai Rishitha Mangipudi updating the record on February 15, 2023, at 10:55 PM. The status is now Draft to Approved.

EXERCISE-2:

Step-1:

Used Data import wizard to import travel approval records.

Travel Approvals

All

50+ items • Sorted by Travel Approval # • Filtered by All travel approvals • Updated a few seconds ago

	Travel Approval #	Created By	Department	Status	Trip Start Date	Trip End Date
1	TA-00001	Sai Rishitha Mangipudi	Technology	Approved	2/16/2023	2/28/2023
2	TA-00201	Sai Rishitha Mangipudi	Office of Communications and Media	Approved	6/14/2019	6/15/2019
3	TA-00202	Sai Rishitha Mangipudi	Disability Determination Bureau	Approved	10/1/2019	10/1/2019
4	TA-00203	Sai Rishitha Mangipudi	Division of Disability and Rehabilitative Services	Rejected	4/3/2019	4/6/2019
5	TA-00204	Sai Rishitha Mangipudi	Technology	Approved	3/13/2019	3/20/2019
6	TA-00205	Sai Rishitha Mangipudi	Human Resources	Approved	4/27/2019	5/5/2019
7	TA-00206	Sai Rishitha Mangipudi	Division of Finance	Rejected	3/9/2019	3/9/2019
8	TA-00207	Sai Rishitha Mangipudi	Contract Management	Approved	12/22/2019	1/1/2020
9	TA-00208	Sai Rishitha Mangipudi	Division of Aging	Approved	4/23/2019	4/28/2019
10	TA-00209	Sai Rishitha Mangipudi	Audit Services	Rejected	8/9/2019	8/15/2019
11	TA-00210	Sai Rishitha Mangipudi	Division of Aging	Approved	11/5/2019	11/11/2019
12	TA-00211	Sai Rishitha Mangipudi	Disability Determination Bureau	Approved	3/22/2019	3/22/2019
13	TA-00212	Sai Rishitha Mangipudi	Technology	Approved	3/6/2019	3/9/2019
14	TA-00213	Sai Rishitha Mangipudi	Office of General Counsel	Approved	11/21/2019	11/26/2019
15	TA-00214	Sai Rishitha Mangipudi	Office of Communications and Media	Approved	3/7/2019	3/19/2019

Step-2:

Created a Report on Travel Requests by Department.

Report: Travel Approvals

Travel Requests by Department

Total Records: 301 Total Out-of-State: 232

Department	Travel Approval: Travel Approval #	Status	Out-of-State	Destination State	Trip Start Date	Trip End Date
Technology (28)	TA-00001	Approved	<input checked="" type="checkbox"/>	CA	2/16/2023	2/28/2023
	TA-00238	Approved	<input checked="" type="checkbox"/>	CA	7/21/2019	7/31/2019
	TA-00241	Approved	<input checked="" type="checkbox"/>	FL	4/25/2019	4/27/2019
	TA-00242	Approved	<input checked="" type="checkbox"/>	FL	7/31/2019	8/21/2019
	TA-00245	Approved	<input type="checkbox"/>	TX	11/19/2019	11/29/2019
	TA-00248	Approved	<input checked="" type="checkbox"/>	GA	3/15/2019	3/19/2019
	TA-00270	Approved	<input checked="" type="checkbox"/>	CA	9/9/2019	9/21/2019
	TA-00290	Rejected	<input type="checkbox"/>	TX	6/12/2019	6/18/2019
	TA-00297	Approved	<input checked="" type="checkbox"/>	CA	11/27/2019	12/4/2019
	TA-00307	Approved	<input checked="" type="checkbox"/>	FL	9/8/2019	9/18/2019
	TA-00332	Approved	<input checked="" type="checkbox"/>	CA	4/4/2019	4/10/2019
	TA-00341	Approved	<input checked="" type="checkbox"/>	OK	6/19/2019	6/25/2019
	TA-00347	Approved	<input type="checkbox"/>	TX	11/14/2019	11/23/2019
	TA-00348	Approved	<input checked="" type="checkbox"/>	GA	2/4/2019	2/10/2019

Travel Requests by Department

Travel Approvals

Department	Travel Approval: Travel Approval #	Status	Out-of-State	Destination State	Trip Start Date	Trip End Date
Technology (6)	TA-00001	Approved		CA	2/16/2023	2/28/2023
	TA-00238	Approved		CA	7/21/2019	7/31/2019
	TA-00241	Approved		FL	4/25/2019	4/27/2019
	TA-00242	Approved		FL	7/31/2019	8/21/2019
	TA-00245	Approved		TX	11/19/2019	11/29/2019
	TA-00248	Approved		GA	3/15/2019	3/19/2019
Subtotal			5			
Office of Medicaid Policy and Planning (1)	TA-00246	Approved		CA	11/5/2019	11/11/2019
Subtotal			1			
Office of Early Childhood and Out-of-School Learning (1)	TA-00234	Approved		TX	7/13/2019	7/19/2019
Subtotal			0			
Office of Communications and Media (1)	TA-00240	Approved		OK	6/2/2019	6/12/2019
Subtotal			1			
Human Resources (1)	TA-00251	Approved		FL	7/9/2019	7/11/2019
Subtotal			1			
Division of Mental Health and Addiction (1)	TA-00235	Approved		TX	12/19/2019	12/19/2019

Test The Report:

Report: Travel Approvals

Travel Requests by Department

Total Records Total Out-of-State
301 232

Sum of Out-of-State

Department

Department	Travel Approval: Travel Approval #	Status	Out-of-State	Destination State	Trip Start Date	Trip End Date
Technology	TA-00001	Approved		CA	2/16/2023	2/28/2023
Quality and Complia...	TA-00238	Approved		CA	7/21/2019	7/31/2019
Office of Medicaid P...	TA-00241	Approved		FL	4/25/2019	4/27/2019
Office of General Co...	TA-00242	Approved		FL	7/31/2019	8/21/2019
Office of Early Chil...	TA-00245	Approved		TX	11/19/2019	11/29/2019
Office of Communica...	TA-00248	Approved		GA	3/15/2019	3/19/2019
Legislative Services	TA-00240	Approved		OK	6/2/2019	6/12/2019
Human Resources	TA-00251	Approved		FL	7/9/2019	7/11/2019
Division of Mental H...	TA-00235	Approved		TX	12/19/2019	12/19/2019
Division of Finance						
Division of Family R...						
Division of Disability						
Division of Aging						
Division of Aging						
Disability Determina...						
Contract Management						
Audit Services						

Step-3:

Created a Travel Requests by Monthly Report

Travel Requests by Month

Travel Approvals

Trip End Date	Out-of-State	Travel Approval: Travel Approval #	Department	Status	Destination State	Trip Start Date
1/24/2019 (1)	(1)	TA-00244	Division of Aging	Approved	OK	1/24/2019
Subtotal						
2/20/2019 (1)	(1)	TA-00237	Division of Aging	Approved	FL	2/17/2019
Subtotal						
3/14/2019 (1)	(1)	TA-00243	Division of Aging	Approved	GA	3/5/2019
Subtotal						
3/19/2019 (1)	(1)	TA-00248	Technology	Approved	GA	3/15/2019
Subtotal						
4/1/2019 (1)	(1)	TA-00249	Division of Family Resources	Approved	TX	4/1/2019
Subtotal						
4/27/2019 (1)	(1)	TA-00241	Technology	Approved	FL	4/25/2019
Subtotal						

Test The Report:

The screenshot shows a Salesforce report titled "Report: Travel Approvals" with the specific view "Travel Requests by Month". The report displays 301 total records. The columns include Trip End Date, Out-of-State, Travel Approval: Travel Approval #, Department, Status, Destination State, and Trip Start Date. The data shows several entries for January 2019, categorized by department such as "Office of Early Childhood and Out-of-School Learning", "Division of Aging", "Quality and Compliance Office", "Division of Mental Health and Addiction", and "Contract Management". Some entries are marked as Approved (e.g., TA-00413, TA-00304, TA-00448, TA-00314) while others are Rejected (e.g., TA-00457). The status column also includes OK and TX.

Step-4:

Created a Dashboard for Travel Approvals

The screenshot shows a Salesforce dashboard titled "Travel Requests Dashboard". The dashboard is located in the "Private Dashboards" folder and was created by Sai Rishitha Mangipudi on 2/16/2023 at 10:33 AM. The dashboard has one item, which is the "Travel Requests Dashboard" itself. The sidebar on the left shows recent dashboards, private dashboards, all dashboards, folders, and favorites. A user profile menu is open on the right, showing the user's name (Sai Rishitha Mangipudi), email (resilient-wolf-gdpcn-dev-edtrailblaze.my...), settings, and log out options. The system status bar at the bottom indicates it's 24°C, partly cloudy, and the date is 26-02-2023.

Test Your Dashboard:

Sai Rishitha Mangipudi
resilient-wolf-gdppcn-dev-ed.traillblaze.my.salesforce.com
[Settings](#) [Log Out](#)

DISPLAY DENSITY

✓ Comfy

Compact

OPTIONS

[Switch to Salesforce Classic](#)

[Add Username](#)

Travel Requests Dashboard

⚠ Last refreshed 10 days ago. Refresh this dashboard to see the latest data.

As of Feb 16, 2023, 10:35 AM Viewing as Sai Rishitha Mangipudi

Travel Requests by Department

A horizontal bar chart titled "Travel Requests by Department". The y-axis is labeled "Department" and lists various departments. The x-axis is labeled "Record Count" with values 0, 9, 18, and 27. Each department has a blue bar representing its record count. The data is as follows:

Department	Record Count
Audit Services	18
Contract Management	20
Disability Determination Bur	19
Division of Aging	20
Division of Disability and Re.	21
Division of Family Resources	15
Division of Finance	19
Division of Mental Health an	17
Human Resources	16
Legislative Services	19
Office of Communications a	21
Office of Early Childhood an...	19

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

Travel Requests by Month

A stacked bar chart titled "Travel Requests by Month". The y-axis is labeled "Record Count" with values 0, 2, and 4. The x-axis shows months from 14/1 to 24/1, followed by "Out-of-State". Each bar is divided into two segments: "true" (blue) and "false" (light blue). The data is as follows:

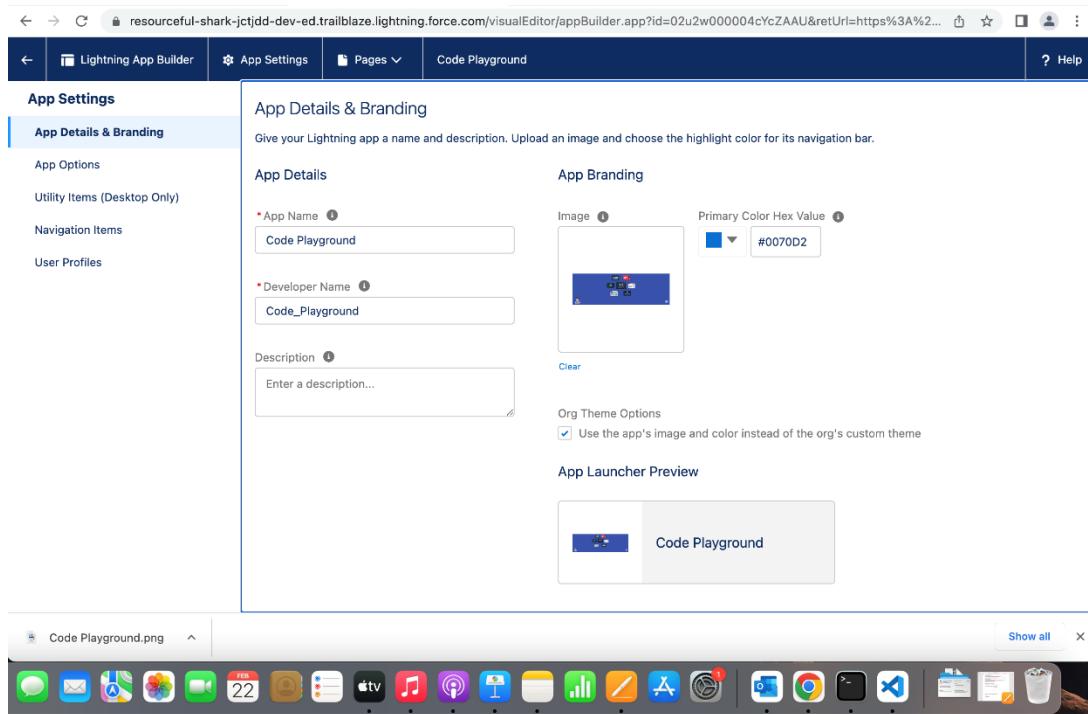
Trip End Date	true	false
14/1	1	1
17/1	1	1
19/1	1	1
1/1	1	1
11/1	1	1
12/1	1	1
12/1	1	1
12/1	2	1
12/1	2	2
13/1	2	2
24/1	3	1
Out-of-State	1	1

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

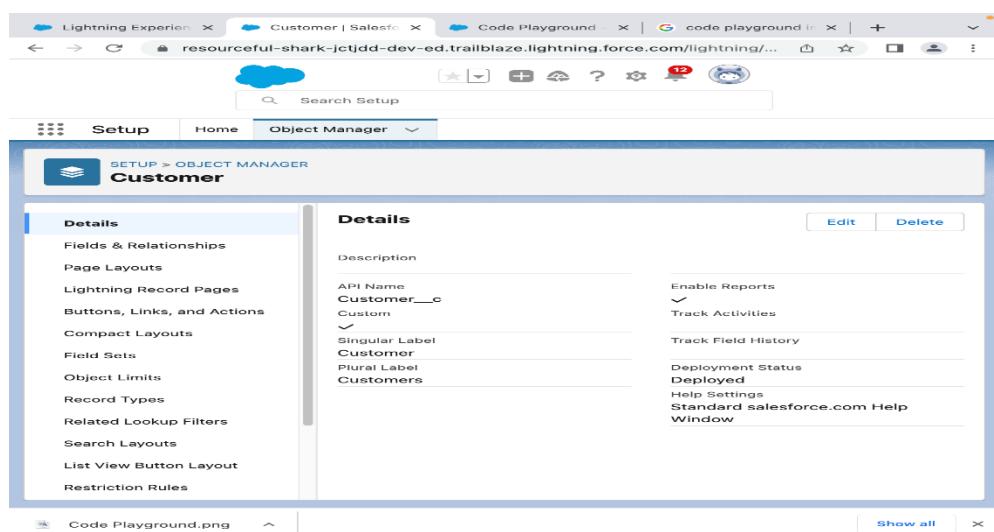
Module-3

EXERCISE-1

Created a new Custom lightning App named “Code Playground” and added the requirements accordingly.



Created custom objects "Customer" and "Billing" with required fields and Tabs



The screenshot shows the Salesforce Setup interface with the 'Object Manager' tab selected. Under the 'Customer' object, the 'Fields & Relationships' section is displayed. The table lists the following fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD
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)	

On the right side of the screen, there is a sidebar with user information and display density settings.

Created Tab for “Customer” custom object.

The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected under the 'Customer' object. The 'Custom Object Tab' section displays the following details:

Tab Label	Customers	Tab Style	
Object	Customer	Splash Page Custom Link Hammer	
Description	Sai Rishitha Mangipudi, 22/02/2023, 11:17 am	Modified By	Sai Rishitha Mangipudi, 22/02/2023, 11:17 am

Below this, there is a note: "Didn't find what you're looking for? Try using Global Search."

The screenshot shows the Salesforce Setup interface with the 'Object Manager' tab selected. Under the 'Billing' object, the 'Details' section is displayed. The table lists the following details:

Details	Details
Fields & Relationships	
Page Layouts	
Lightning Record Pages	
Buttons, Links, and Actions	
Compact Layouts	
Field Sets	
Object Limits	
Record Types	
Related Lookup Filters	
Search Layouts	
List View Button Layout	
Restriction Rules	

Setup > Object Manager
Billing

Fields & Relationships
7 items, Sorted by Field Label

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		

Created Custom Tab for “Billing” object

Setup > Object Manager
Tabs

Custom Object Tab
Billings

Below is the information for the custom tab. Click Edit to change the custom tab.

Custom Tab Definition Detail	Edit Delete	Tab Style
Tab Label: Billings Object: Billing Description: Created By: Sai Rishitha Mangipudi, 22/02/2023, 11:31 am		Splash Page Custom Link
		Computer
		Mobile
		Tablet

Test Your App:

Code Playground
Leads | Accounts | Contacts | Opportunities | Cases | Customers | Billing

B-0019

Related **Details**

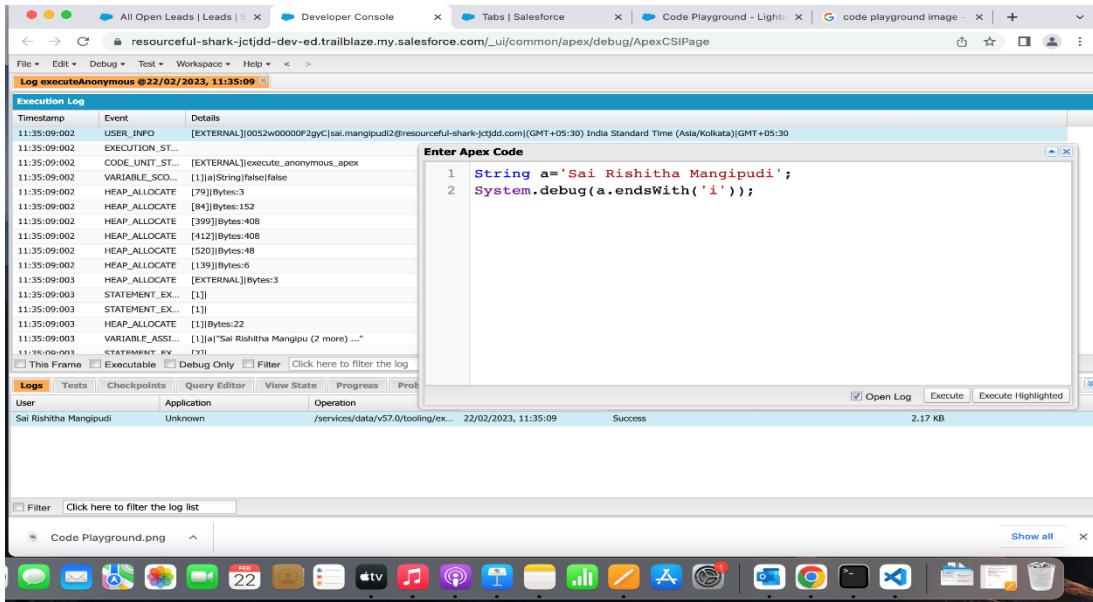
Billing Number: B-0019
Amount Paid: ₹10,00,000
Customer Type:
Status: Paid
Customer: Wipro
Created By: Sai Rishitha Mangipudi, 28/02/2023, 12:37 pm
Last Modified By: Sai Rishitha Mangipudi, 28/02/2023, 12:37 pm

New Contact | Edit | New Opportunity

EXERCISE-2:

Used Execute Anonymous to define and execute the following codes:

Defined a String Variable & use string method ‘endsWith()’ to display the output.



The screenshot shows the Salesforce Developer Console interface. In the center, there's a modal window titled "Enter Apex Code" containing the following code:

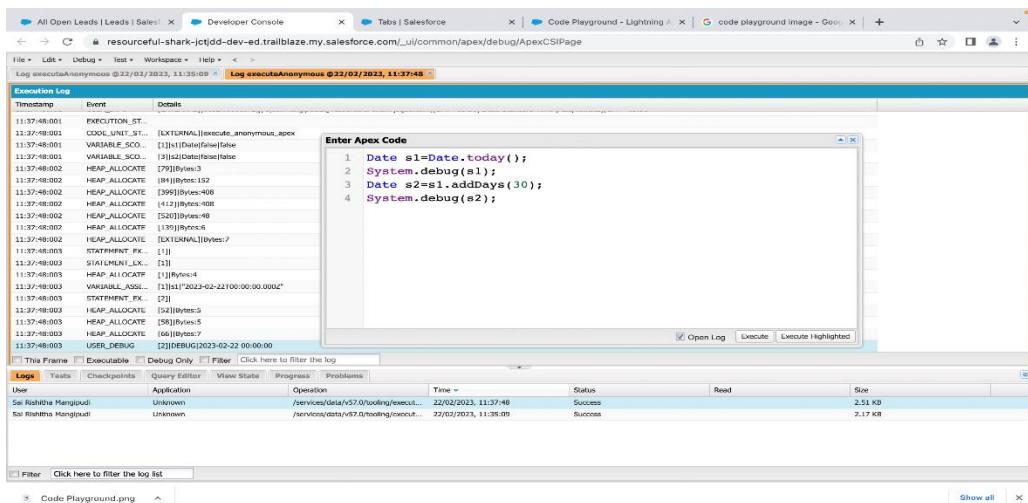
```
1 String a='Sai Rishitha Mangipudi';
2 System.debug(a.endsWith('i'));
```

Below the code, the "Logs" tab is selected in the navigation bar. The log table shows the execution details:

User	Application	Operation	Time	Status	Size
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex... 22/02/2023, 11:35:09	Success		2.17 KB

The log table has a "Filter" button at the bottom. At the very bottom of the interface, there's a Mac OS X-style dock with various application icons.

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



The screenshot shows the Salesforce Developer Console interface. In the center, there's a modal window titled "Enter Apex Code" containing the following code:

```
1 Date s1=Date.today();
2 System.debug(s1);
3 Date s2=s1.adddays(30);
4 System.debug(s2);
```

Below the code, the "Logs" tab is selected in the navigation bar. The log table shows the execution details:

User	Application	Operation	Time	Status	Size
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/execut... 22/02/2023, 11:37:46	Success		2.51 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/execut... 22/02/2023, 11:38:09	Success		2.17 KB

The log table has a "Filter" button at the bottom. At the very bottom of the interface, there's a Mac OS X-style dock with various application icons.

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

The screenshot shows the Salesforce Developer Console interface. In the top right, there are tabs for 'Developer Console', 'Tabs | Salesforce', 'Code Playground - Light...', and 'code playground image...'. The main area has tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Logs' tab is selected. The log table shows four entries:

User	Application	Operation	Time	Status	Read	Size
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:42:47	Success	2.45 KB	
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:37:48	Success	2.51 KB	
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:35:09	Success	2.17 KB	

In the center, there is a 'Enter Apex Code' window with the following code:

```
1 String str='10';
2 Integer a1=Integer.Valueof(str1);
3 System.debug(a1);
4 System.debug(a1+20);
```

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

The screenshot shows the Salesforce Developer Console interface. The 'Logs' tab is selected in the bottom navigation bar. The log table shows five entries:

User	Application	Operation	Time	Status	Read	Size
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:40:33	Success	2.45 KB	
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:42:47	Success	2.45 KB	
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:37:48	Success	2.51 KB	
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:35:09	Success	2.17 KB	

In the center, there is a 'Enter Apex Code' window with the following code:

```
1 String str='Sai Rishitha Mangipudi';
2 System.debug('Size of the string'+str.length());
```

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

The screenshot shows the Salesforce Developer Console interface. The 'Logs' tab is selected in the bottom navigation bar. The log table shows five entries:

User	Application	Operation	Time	Status	Read	Size
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:40:33	Success	2.45 KB	
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:42:47	Success	2.45 KB	
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:37:48	Success	2.51 KB	
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:35:09	Success	2.17 KB	

In the center, there is a 'Enter Apex Code' window with the following code:

```
1 List<String> lst=new List<String>();
2 lst.add('First Element');
3 lst.add('Second Element');
4 lst.add('Third Element');
5 System.debug('After using add() method:'+lst);
6 System.debug('After using get() method:' +lst.get(0));
7 lst.set(2,'Replaced Third Element');
8 System.debug('After using set() method:' +lst);
9 lst.clear();
10 System.debug('After clearing all elements using clear() method'+lst);
```

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

```

1 Integer x=20;
2 while(x>12){
3   for(x=0;x<10;x++){
4     System.debug('Value of x is:' +x);
5   }
6 }

```

The screenshot shows the Salesforce Developer Console interface. On the left, the Execution Log pane displays the following entries:

Timestamp	Event	Details
11:58:38:002	USER_DEBUG	[4]DEBUG Value of x is:0
11:58:38:002	USER_DEBUG	[4]DEBUG Value of x is:1
11:58:38:003	USER_DEBUG	[4]DEBUG Value of x is:2
11:58:38:003	USER_DEBUG	[4]DEBUG Value of x is:3
11:58:38:003	USER_DEBUG	[4]DEBUG Value of x is:4
11:58:38:003	USER_DEBUG	[4]DEBUG Value of x is:5
11:58:38:003	USER_DEBUG	[4]DEBUG Value of x is:6
11:58:38:003	USER_DEBUG	[4]DEBUG Value of x is:7
11:58:38:003	USER_DEBUG	[4]DEBUG Value of x is:8
11:58:38:003	USER_DEBUG	[4]DEBUG Value of x is:9

On the right, the Enter Apex Code pane contains the provided anonymous apex code. Below the code editor is a toolbar with buttons for Open Log, Execute, and Execute Highlighted.

EXERCISE-3:

Answer the following in True Or False:

Integer myluckyNumber = 15;

Integer myunluckyNumber = 7;

myluckyNumber != myunluckyNumber + 8.

```

1 Integer myluckyNumber=15;
2 Integer myunluckyNumber=7;
3 if(myluckyNumber != myunluckyNumber+8)
4 {
5   System.debug('False');
6 }
7 else{
8   System.debug('True');
9 }
10

```

The screenshot shows the Salesforce Developer Console interface. On the left, the Execution Log pane displays the following entries:

User	Application	Operation	Time	Status	Read	Size
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:58:38	Success		5.61 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:57:26	Success		5.61 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:53:27	Success		4.23 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:46:13	Success		2.3 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:42:47	Success		2.45 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:37:48	Success		2.51 KB

On the right, the Enter Apex Code pane contains the provided anonymous apex code. Below the code editor is a toolbar with buttons for Open Log, Execute, and Execute Highlighted.

EXERCISE-4:

Answer the following in True Or False:

Boolean.isTrue = True;

Boolean.isFalse = false;

isTrue || isFalse

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for 'Developer Console', 'Tabs | Salesforce', 'Code Playground - Light', and 'code playground Image'. Below the navigation bar, the 'Execution Log' section displays the following Apex code:

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

The log table shows the execution details:

User	Application	Operation	Time	Status	Read	Size
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 12:01:56	Success	2.46 KB	2.46 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 12:01:56	Success	5.61 KB	5.61 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:57:26	Success	4.23 KB	4.23 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:53:27	Success	2.3 KB	2.3 KB

The bottom of the screen shows the Mac OS X dock with various application icons.

EXERCISE-5:

Answer the following in True Or False:

Date.today() = Date.today();

Date.tomorrow() = Date.today().addDays(1);

today != tomorrow

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for 'Developer Console', 'Tabs | Salesforce', 'Code Playground - Light', and 'code playground Image'. Below the navigation bar, the 'Execution Log' section displays the following Apex code:

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

The log table shows the execution details:

User	Application	Operation	Time	Status	Read	Size
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 12:29:42	Success	2.46 KB	2.46 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 12:28:12	Success	2.65 KB	2.65 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 12:01:56	Success	5.61 KB	5.61 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:58:38	Success	4.23 KB	4.23 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:57:26	Success	5.61 KB	5.61 KB
Sai Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/ex...	22/02/2023, 11:53:27	Success	2.3 KB	2.3 KB

The bottom of the screen shows the Mac OS X dock with various application icons.

EXERCISE-6:

Write a program and execute to demo the use of “If..else if...else”. Define an Integer variable called Score.

System.debug ('Grade: A+'), if Score is equal to 100.

System.debug ('Grade: A'), if Score is greater than or equals to 90.

System.debug ('Grade: B'), if Score is greater than or equals to 80.

System.debug ('Grade: Failed'), define as final else.

The screenshot shows the Salesforce Developer Console interface. The top window displays the Apex code for the `StudentGrade` class, which contains a method `gradeAssigning`. The code uses nested if statements to determine a grade based on a score. The bottom window shows the `Execution Log` with two entries: a `USER_DEBUG` log at timestamp 12:37:34:013 indicating a score of 100, and another `USER_DEBUG` log at the same timestamp indicating the grade as A+.

```
1 public class StudentGrade{
2     public static void gradeAssigning(){
3         Integer Score=100;
4         System.debug('Score is:'+Score);
5         if(Score==100){
6             System.debug('Grade: A+');
7         }
8         else if(Score>=90){
9             System.debug('Grade: A');
10        }
11        else if(Score>=80){
12            System.debug('Grade: B');
13        }
14        else{
15            System.debug('Grade: Failed');
16        }
17    }
18 }
```

Timestamp	Event	Details
12:37:34:013	USER_DEBUG	[4]DEBUG Score is:100
12:37:34:013	USER_DEBUG	[6]DEBUG Grade: A+

EXERCISE-7:

Wrote a program to execute and demo the use of “Apex – for Loop”. Initialized the Billing object list to store the Billing Records created today, using SQOL. Used for Loop to iterate over the list and check if the Billings status = “Paid”.

System.debug ('Value of Current Record in the Loop'+ Billing List);.

System.debug('Value of BillingList '+BillingList);

If Status is “Paid” then added the Billings into a List of String.

```

1 v public class BillingObj {
2 v     public static void createrecords(){
3 v         Billing__c b1=new Billing__c();
4 v         b1.Amount_Paid__c=15000;
5 v         b1.Status__c='Paid';
6 v         Billing__c b2=new Billing__c();
7 v         b2.Amount_Paid__c=30000;
8 v         b2.Status__c='Paid';
9 v         Billing__c b3=new Billing__c();
10 v        b3.Amount_Paid__c=45000;
11 v        b3.Status__c='Paid';
12 v        List<Billing__c> bills=new List<Billing__c>();
13 v        bills.add(b1);
14 v        bills.add(b2);
15 v        bills.add(b3);
16 v        try{
17 v            insert bills;
18 v        }
19 v        catch(Exception e){
20 v            System.debug('Inavalid Data');
21 v        }
22 v        bills=[Select Id,Name,Status__c FROM Billing__c WHERE CreatedDate=today];
23 v        System.debug('Value of Billing List:'+bills);
24 v        for(Billing__c bill:bills)
25 v        {
26 v            if(bill.Status__c=='Paid')
27 v            {
28 v                System.debug('Value of current record over which current record is iterating is:'+bill);
29 v                System.debug('Billing Records are:'+bill.Name);
30 v            }
31 v        }
32 v    }
33 v}
34 v
35 v

```

Timestamp	Event	Details
12:50:126	USER_DEBUG	[23]DEBUG Value of Billing List:(Billing__c:{Id=a012w000017mngfAAA, Name=B-0001, Status__c='Paid'}, Billing__c:{Id=a012w000017mngfAAA, Name=B-0002, Status__c='Paid'}, Billing__c:{Id=a012w000017mngfAAA, Name=B-0003, Status__c='Paid'})
12:50:126	USER_DEBUG	[28]DEBUG Value of current record over which current record is iterating is:Billing__c:{Id=a012w000017mngfAAA, Name=B-0001, Status__c='Paid'}
12:50:126	USER_DEBUG	[29]DEBUG Billing Records are:B-0001
12:50:126	USER_DEBUG	[28]DEBUG Value of current record over which current record is iterating is:Billing__c:{Id=a012w000017mngfAAA, Name=B-0002, Status__c='Paid'}
12:50:126	USER_DEBUG	[29]DEBUG Billing Records are:B-0002
12:50:127	USER_DEBUG	[28]DEBUG Value of current record over which current record is iterating is:Billing__c:{Id=a012w000017mngfAAA, Name=B-0003, Status__c='Paid'}
12:50:127	USER_DEBUG	[29]DEBUG Billing Records are:B-0003

EXERCISE-8:

Wrote a Class to demo the use of Constants in Apex. Wrote a class named “DiscountClass”. Defined a Final Decimal variable called “regularDiscount”. Defined a Decimal variable called “finalPrice”.

```
finalPrice = price - price*regularDiscount;
```

Defined a Method called “calculateDiscount(Integer price)”, takes a Decimal data type.Returned a Decimal data type.

```
1 public class DiscountClass {  
2     Decimal regularDiscount=0.20;  
3     Decimal finalPrice;  
4     public Decimal calculateDiscount(Integer price){  
5         finalPrice=price-price*regularDiscount;  
6         System.debug('finalPrice : '+finalPrice);  
7         return finalPrice;  
8     }  
9 }  
10  
11 Enter Apex Code  
12 1 DiscountClass d=new DiscountClass();  
13 2 d.calculateDiscount(400);
```

Timestamp	Event	Details
10:46:37:018	USER_DEBUG	[6]DEBUG finalPrice : 320.00

EXERCISE-9

Write a Class to demo the use of Interface in Apex.Created Interface Class Named “InterfaceExample” with Method Signature “Double percentageDiscountTobeApplied();”.Created an Apex Class to implement the Interface Class named “PremiumCustomer”.Called the interface method to return 30% Discount.Created another Apex Class to implement the Interface Class named “NormalCustomer”. Call the interface method to return 10% Discount.

Developer Console

resourceful-shark-jctjdd-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

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

InterfaceExample.apxc PremiumCustomer.apxc NormalCustomer.apxc

Code Coverage: None API Version: 57 Go To

```
1 public interface InterfaceExample {  
2     Double percentageDiscountToBeApplied();  
3 }
```

Developer Console

resourceful-shark-jctjdd-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

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

InterfaceExample.apxc PremiumCustomer.apxc NormalCustomer.apxc Log executeAnonymous @27/02/2023, 15:37:11

Code Coverage: None API Version: 57 Go To

```
1 public class PremiumCustomer implements InterfaceExample {  
2     public Double percentageDiscountToBeApplied(){  
3         return 0.30;  
4     }  
5 }
```

Enter Apex Code

```
1 PremiumCustomer p=new PremiumCustomer();  
2 double y=p.percentageDiscountToBeApplied;  
3 System.debug(y);
```

Open Log Execute Execute Highlighted

Developer Console

resourceful-shark-jctjdd-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

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

PremiumCustomer.apxc NormalCustomer.apxc Log executeAnonymous @23/02/2023, 10:54:17 Log executeAnonymous @23/02/2023, 10:55:05

Execution Log

Timestamp	Event	Details
10:55:05.017	USER_DEBUG	[3]DEBUG 0.3

Developer Console - Google Chrome

resourceful-shark-jctjdd-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

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

NormalCustomer.apxc PremiumCustomer.apxc CustomerTrigger.apat CustomerTriggerTestApac DisqualifyTestLeads.apat DisqualifyTestLeads.apxc OppView.vfp AccountList.vfp NewCaseListController.vfp BillingRecord

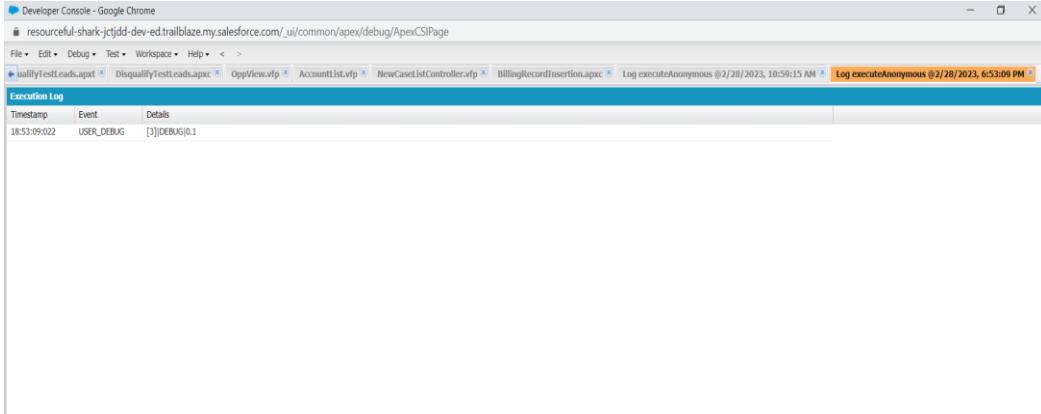
Code Coverage: None API Version: 57 Go To

```
1 public class NormalCustomer implements InterfaceExample{  
2     public Double percentageDiscountToBeApplied(){  
3         return 0.10;  
4     }  
5 }
```

Enter Apex Code

```
1 NormalCustomer n=new NormalCustomer();  
2 Double res=n.percentageDiscountToBeApplied();  
3 System.debug(res);  
4
```

Open Log Execute Execute Highlighted



EXERCISE-10:

Demo on DML Insert Operation Using Database methods. Inserted a Customer Record First using simple DML Statement. Customer Name = “Wipro”, Type = “Premium”. Inserted Billing record using Database methods. Iterated through the Success/Error result, showed inserted record ID, or the error message with fields. Used a List<Billing c> to perform the Database insert method & store the result in srList. Billing Status = Paid, Amount Paid = 5000000, linked Billing record to Customer using the customer ID.

Screenshot of the Salesforce Developer Console showing the BillingRecordInsertion.apxc code and the Execution Log.

```

1 * public class BillingRecordInsertion {
2     public static void recInsertion(){
3         Customer__c cust=new Customer__c();
4         cust.Name='Wipro';
5         cust.Customer_Type__c='Premium';
6         insert cust;
7         ID cust_id=cust.Id;
8         Billing__c[] bill=new List<Billing__c>{
9             new Billing__c(Status__c='Paid',Amount_Paid__c=5000000,Customer__c=cust_id);
10            //new Billing__c());
11        Database.SaveResult[] l=Database.insert(bill,false);
12        for(Database.SaveResult sr:l){
13            if(sr.isSuccess()){
14                System.debug('Record Inserted. Billing ID:' +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 }
```

Execution Log:

User	Application	Operation	Time	Status	Read	Size
Sal Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/execut...	23/02/2023, 11:50:15	Success		5.54 KB
Sal Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/execut...	23/02/2023, 11:47:46	Success		7.74 KB
Sal Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/execut...	23/02/2023, 11:45:57	Insert failed. First exception on row ...	Unread	3.95 KB
Sal Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/execut...	23/02/2023, 11:43:38	Insert failed. First exception on row ...	Unread	4.97 KB
Sal Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/execut...	23/02/2023, 11:41:26	Insert failed. First exception on row ...	Unread	4.92 KB
Sal Rishitha Mangipudi	Unknown	/services/data/v57.0/tooling/execut...	23/02/2023, 11:40:38	Insert failed. First exception on row ...	Unread	4.97 KB

The screenshot shows the Salesforce Developer Console interface. At the top, the URL is `resourceful-shark-jctjdd-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. Below the URL, there are tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Logs' tab is selected. In the main area, there is a table titled 'Execution Log' with columns 'Timestamp', 'Event', and 'Details'. A single log entry is present:

Timestamp	Event	Details
11:50:15:072	USER_DEBUG	[14]I DEBUG Record Inserted. Billing ID:a012w000017n5DaAAI

At the bottom of the developer console, there are filters: 'This Frame', 'Executable', 'Debug Only', 'Filter', and a search bar 'Click here to filter the log'. The 'Problems' tab is highlighted.

EXERCISE-11:

Wrote and executed SOQL queries from Developer Console to fetch Display ID, Amount, Stage, Account Name, Account Industry, Account Website From Opportunity. Added a Where Clause “Account Industry = Energy” AND “Account Annual Revenue > 5000”

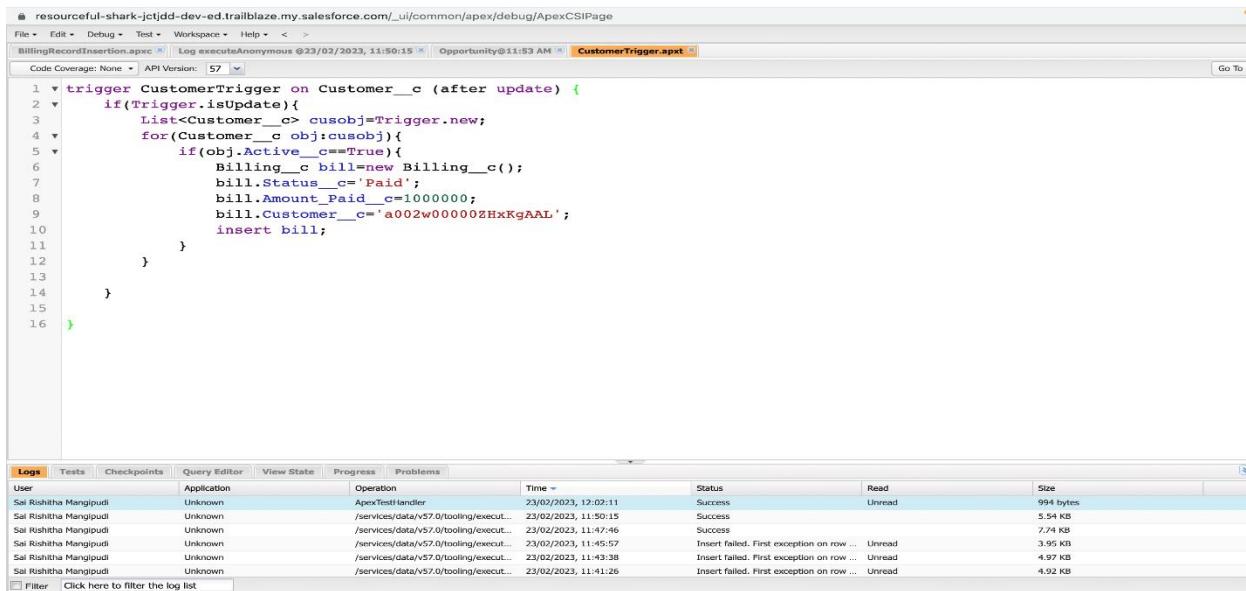
The screenshot shows the Salesforce Developer Console interface. At the top, the URL is `resourceful-shark-jctjdd-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. Below the URL, there are tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Query Editor' tab is selected. In the main area, there is a table titled 'Query Results - Total Rows: 10' with columns 'AccountId', 'Amount', 'Name', 'StageName', 'Account.Industry', and 'Account.Website'. The data is as follows:

AccountId	Amount	Name	StageName	Account.Industry	Account.Website
0012w00001LsJ8ZAAV	125000	United Oil Office Portable Generators	Negotiation/Review	Energy	http://www.uos.com
0012w00001LsJ8ZAAV	270000	United Oil Refinery Generators	Proposal/Price Quote	Energy	http://www.uos.com
0012w00001LsJ8ZAAV	120000	United Oil SLA	Closed Won	Energy	http://www.uos.com
0012w00001LsJ8ZAAV	270000	United Oil Installations	Negotiation/Review	Energy	http://www.uos.com
0012w00001LsJ8ZAAV	270000	United Oil Installations	Closed Won	Energy	http://www.uos.com
0012w00001LsJ8ZAAV	915000	United Oil Refinery Generators	Closed Won	Energy	http://www.uos.com
0012w00001LsJ8ZAAV	235000	United Oil Installations	Closed Won	Energy	http://www.uos.com
0012w00001LsJ8ZAAV	440000	United Oil Emergency Generators	Closed Won	Energy	http://www.uos.com
0012w00001LsJ8ZAAV	120000	United Oil Standby Generators	Closed Won	Energy	http://www.uos.com
0012w00001LsJ8ZAAV	675000	United Oil Plant Standby Generators	Needs Analysis	Energy	http://www.uos.com

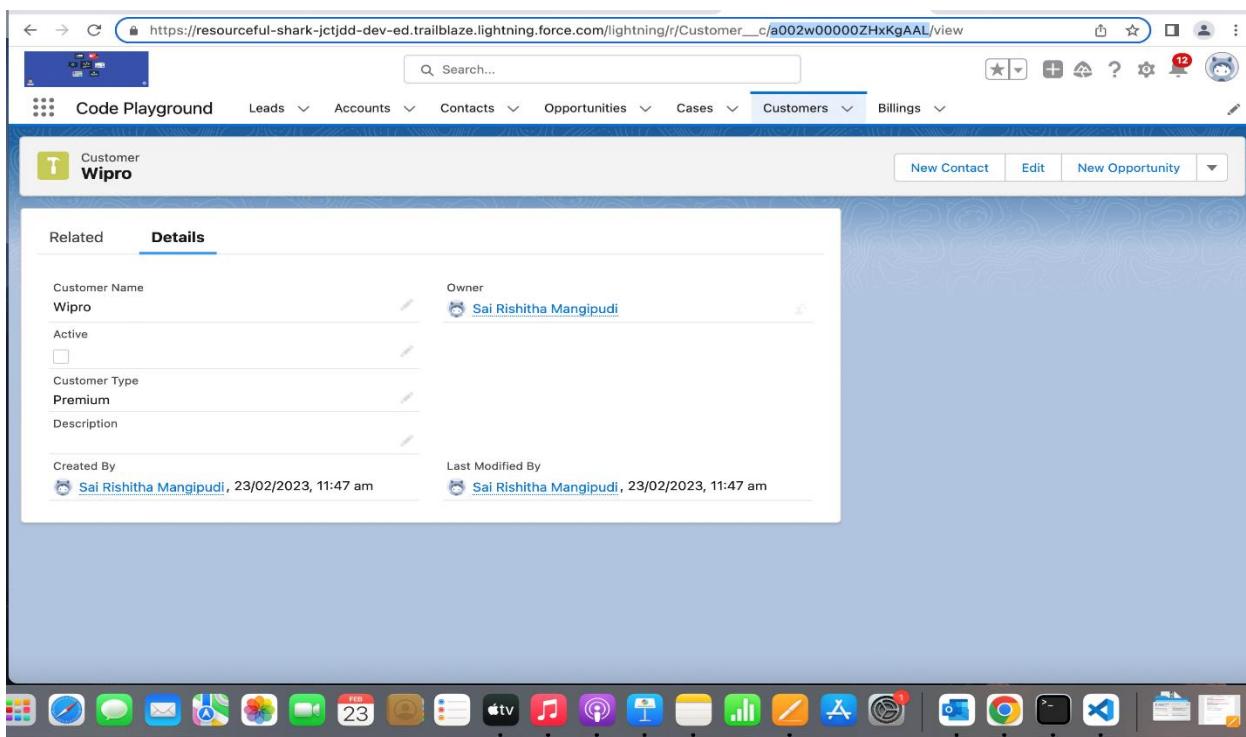
Below the results, there is a 'Query Grid' section with buttons for 'Save Rows', 'Insert Row', 'Delete Row', and 'Refresh Grid'. There is also a 'History' section showing the query was 'Executed'.

EXERCISE-12:

Wrote an Apex Trigger named “CustomerTrigger” to create a Billing Record when on Customer Object’s ‘Status’ field changes to Active from Inactive. Billing “Status” must be: Paid, Amount Paid = 1000000. Perform the DML on a List.

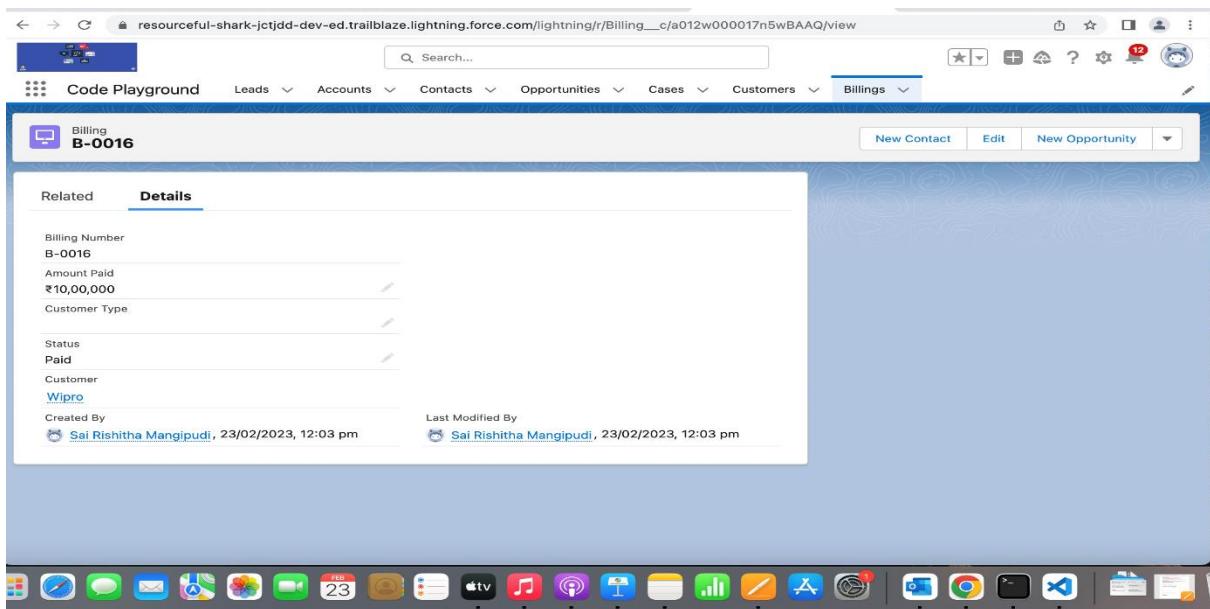


```
trigger CustomerTrigger on Customer__c (after update) {
    if(Trigger.isUpdate){
        List<Customer__c> cusobj=Trigger.new;
        for(Customer__c obj:cusobj){
            if(obj.Active__c==true){
                Billing__c bill=new Billing__c();
                bill.Status__c='Paid';
                bill.Amount_Paid__c=1000000;
                bill.Customer__c='a002w00000ZHxKgAAL';
                insert bill;
            }
        }
    }
}
```



The screenshot shows the Salesforce Lightning Experience interface. At the top, there's a navigation bar with links for Leads, Accounts, Contacts, Opportunities, Cases, Customers, and Billings. Below the navigation is a search bar and a toolbar with various icons. The main area displays a customer record for "Customer Wipro". The record has the following details:

Customer Name	Owner
Wipro	Sai Rishitha Mangipudi
Active	(checkbox checked)
Customer Type	Premium
Description	(empty)
Created By	Sai Rishitha Mangipudi, 23/02/2023, 11:47 am
Last Modified By	Sai Rishitha Mangipudi, 23/02/2023, 11:47 am



EXERCISE-13:

Wrote a Test Class for CustomerTrigger. Created the Data for the Customer custom Object. Used Test.startTest() & Test.stopTest(). Used assert statement to validate the output, Test Class must have full code coverage.

Developer Console - Google Chrome

```

1  @isTest
2  public class CustomerTriggerTest {
3      @isTest
4      static void testCustomer(){
5          //Creating Customer
6          Customer__c cust1 = new Customer__c();
7          cust1.Name='test1';
8          cust1.Customer_Type__c = 'Premium';
9          insert cust1;
10
11         //Updating Customer
12         cust1.Active__c = True;
13         update cust1;
14
15         //After Updating, changing the billing information
16         if(cust1.Active__c == True){
17             Test.startTest();
18             Billing__c billing = new Billing__c();
19             billing.Status__c = 'Paid';
20             billing.Amount_Paid__c = 1000000;
21             Test.stopTest();
22             system.assertEquals('Paid', billing.Status__c);
23             system.assertEquals(1000000, billing.Amount_Paid__c);
24         }
25     }
26 }
```

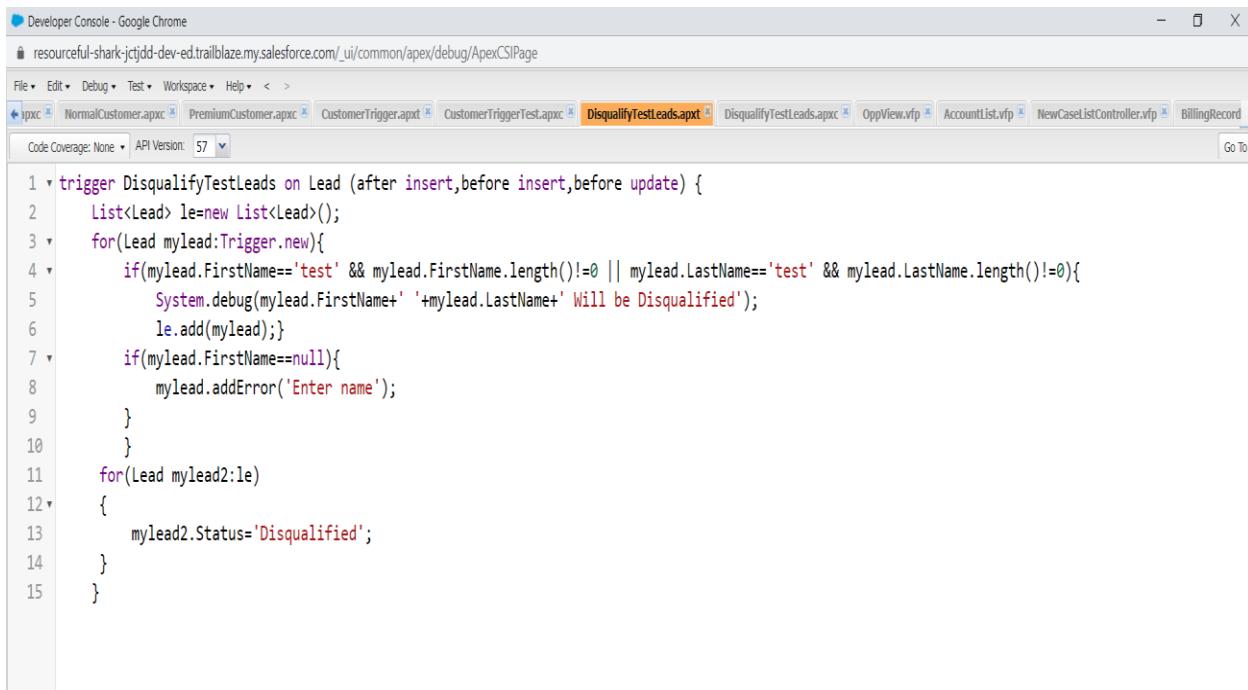
Logs Tests Checkpoints Query Editor View State Progress Problems

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage
						Class Percent Lines
						Overall 26%
						BillingObj 0% 0/21
						BillingRecordInsertion 0% 0/12
						CustomerTrigger 100% 9/9
						DiscountClass 0% 0/4
						DisqualifyTestLeads 100% 9/9

Type here to search 27°C Partly cloudy 19:15 28-02-2023 ENG

EXERCISE-14:

Wrote an Apex Trigger named “DisqualifyTestLeads”. Validated that the Lead First/Last Name must not be Blank. Defined logic to find Leads with ‘Test’ in the Name. IF ‘Test Lead’ found, system.debug(Lead First Name + ‘ ’ + Lead Last Name + ‘Will be disqualified!'); Added the disqualified Leads to a List<Lead>, used for loop to iterate through the list and updated Lead status field to ‘Disqualified’.



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is resourceful-shark-jctjdd-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab bar shows several files: NormalCustomer.apxc, PremiumCustomer.apxc, CustomerTrigger.apxc, CustomerTriggerTest.apxc, DisqualifyTestLeads.apxc (which is the active tab), DisqualifyTestLeads.apxc, OppView.vfp, AccountList.vfp, NewCaseListController.vfp, and BillingRecord. Below the tabs, there are dropdowns for 'Code Coverage: None' and 'API Version: 57'. The main area contains the Apex trigger code:

```
trigger DisqualifyTestLeads on Lead (after insert,before insert,before update) {
    List<Lead> le=new List<Lead>();
    for(Lead mylead:Trigger.new){
        if(mylead.FirstName=='test' && mylead.FirstName.length()!=0 || mylead.LastName=='test' && mylead.LastName.length()!=0){
            System.debug(mylead.FirstName+' '+mylead.LastName+' Will be Disqualified');
            le.add(mylead);
        }
        if(mylead.FirstName==null){
            myleadaddError('Enter name');
        }
    }
    for(Lead mylead2:le)
    {
        mylead2.Status='Disqualified';
    }
}
```

EXERCISE-15:

Wrote a Test Class for DisqualifyTestLeads. Created the Data for the Lead Object. Used Test.startTest() & Test.stopTest(). Used assert statement to validate the output, Test Class have full code coverage.

Developer Console - Google Chrome

resourceful-shark-jctjdd-dev-ed.traillblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

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

↳ DisqualifyTestLeads.apxc [] PremiumCustomer.apxc [] CustomerTrigger.apxc [] CustomerTriggerTest.apxc [] DisqualifyTestLeads.apxt [] OppView.vfp [] AccountList.vfp [] NewCaseListController.vfp [] BillingRecord [] Run Test Go To

Code Coverage: None ▾ API Version: 57

```
1 @isTest
2 public class DisqualifyTestLeads {
3
4     @isTest
5     private static void insertData1(){
6         Lead l1 = new Lead();
7         l1.FirstName = 'Test';
8         l1.LastName = 'Test';
9         l1.Company = 'Test';
10        Test.startTest();
11        insert l1;
12        Test.stopTest();
13        System.assertEquals('Test', l1.FirstName);
14        System.assertEquals('Test', l1.LastName);
15        System.assertEquals('Test', l1.Company);
16    }
17    @isTest
18    private static void insertData2(){
19        Lead l2 = new Lead();
20        l2.LastName = 'Test';
21        l2.Company = 'Wipro';
22        Test.startTest();
23        insert l2;
24        Test.stopTest();
25        System.assertEquals(null, l2.FirstName);
26        System.assertEquals('Test', l2.LastName);
27        System.assertEquals('Test', l2.Company);
28    }
29 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage																								
						<table border="1"><thead><tr><th>Class</th><th>Percent</th><th>Lines</th></tr></thead><tbody><tr><td>NormalCustomer</td><td>0%</td><td>0/4</td></tr><tr><td>BillingRecordInsertion</td><td>0%</td><td>0/12</td></tr><tr><td>CustomerTrigger</td><td>100%</td><td>9/9</td></tr><tr><td>DiscountClass</td><td>0%</td><td>0/4</td></tr><tr><td>DisqualifyTestLeads</td><td>100%</td><td>9/9</td></tr><tr><td>NewCaseListController</td><td>0%</td><td>0/4</td></tr><tr><td>NormalCustomer</td><td>0%</td><td>0/2</td></tr></tbody></table>	Class	Percent	Lines	NormalCustomer	0%	0/4	BillingRecordInsertion	0%	0/12	CustomerTrigger	100%	9/9	DiscountClass	0%	0/4	DisqualifyTestLeads	100%	9/9	NewCaseListController	0%	0/4	NormalCustomer	0%	0/2
Class	Percent	Lines																												
NormalCustomer	0%	0/4																												
BillingRecordInsertion	0%	0/12																												
CustomerTrigger	100%	9/9																												
DiscountClass	0%	0/4																												
DisqualifyTestLeads	100%	9/9																												
NewCaseListController	0%	0/4																												
NormalCustomer	0%	0/2																												

Type here to search              27°C Partly cloudy   ENG 19:24 28-02-2023

EXERCISE-16:

Created a Visualforce page which displays Opportunity fields as output fields. The page is named “OppView”, It references the Opportunity standard controller. Defined 4 apex outputField components: Opportunity Name, Amount, Close Date, Opportunity Account Name.

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

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
ipxc NormalCustomer.apxc PremiumCustomer.apxc CustomerTrigger.apxt CustomerTriggerTest.apxc DisqualifyTestLeads.apxc DisqualifyTestLeads.apxc OppView.vfp AccountList.vfp NewCaseListController.vfp BillingRecord

Preview API Version: 57

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

Logs Tests Checkpoints Query Editor View State Progress Problems

Status	Test Run	Enqueued Time	Duration	Failures	Total

Overall Code Coverage

Class	Percent	Lines
NormalCustomer	0%	0/12
BillingRecordInsertion	0%	0/12
Customer trigger	100%	9/9
DiscountClass	0%	0/4
DisqualifyTestLeads	100%	9/9
NewCaseListController	0%	0/4
NormalCustomer	0%	0/7

Type here to search 27°C Partly cloudy 19:28 28-02-2023

Calendar - Sai Rishi | Microsoft (2) | Hands-On Orgs | Recently Viewed | Customer | Salesforce | https://resourceful-shark-jctjdd-dev-ed--c.trailblaze.vf.force.com/apex/OppView?core.apexpages.request.devconsole=1

Opportunities

Opportunity Name	Amount	Account Name

EXERCISE-17:

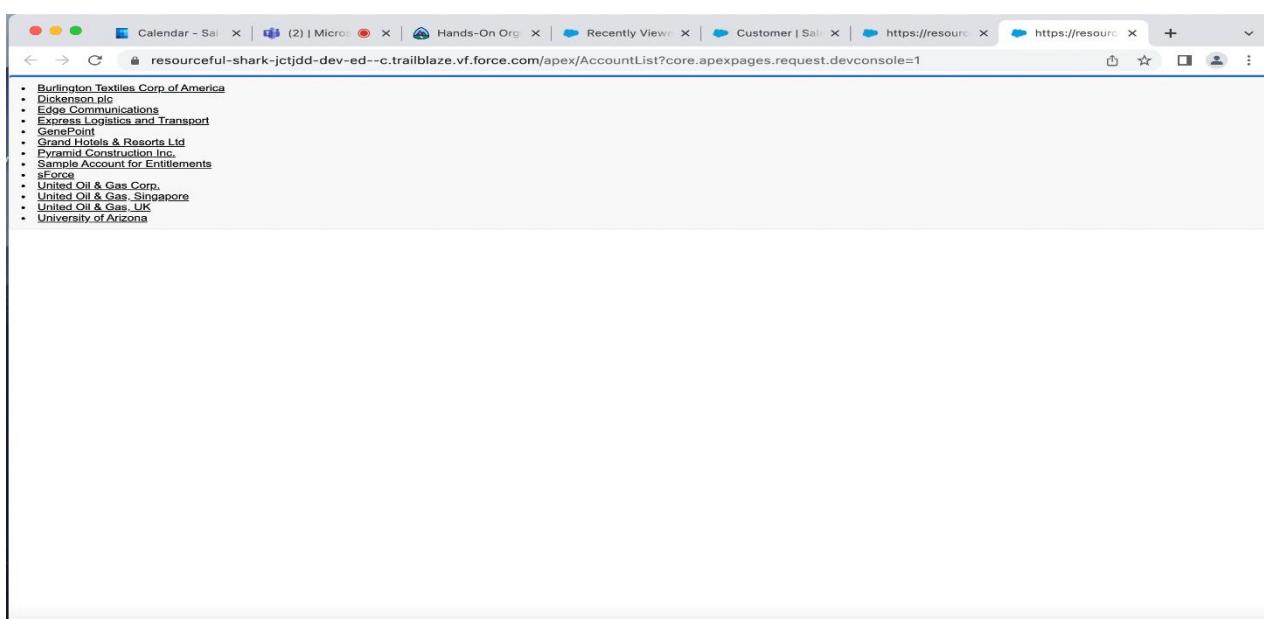
Created a Visualforce page which shows a list of Accounts linked to their record pages. The page named is AccountList. It referenced the Account standard controller. It has a recordSetVar attribute equal to accounts. It has a Visualforce apex:repeat component, with the following: Used the var attribute set to “a”, Used the HTML list tag, Used the apex:outputLink component to link to the respective record detail page.

The screenshot shows the Salesforce Developer Console in Google Chrome. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a preview section for AccountList.vfp. Below the navigation is a list of open pages: NormalCustomer.apxc, PremiumCustomer.apxc, CustomerTrigger.apxc, DisqualifyTestLeads.apxc, OppView.vfp, AccountList.vfp (selected), NewCaseListController.vfp, and BillingRecord. The main area displays the Visualforce page code:

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

Below the code, the Test tab is selected in the navigation bar. The status bar shows "Test Run". The overall code coverage table is as follows:

Class	Percent	Lines
NormalCustomer	0%	0/12
BillingRecordInsertion	0%	0/12
CustomerTrigger	100%	0/4
DiscountClass	0%	0/4
DisqualifyTestLeads	100%	9/9
NewCaseListController	0%	0/4
NormalCustomer	0%	0/2



EXERCISE-18:

Created a Visualforce page that uses a custom controller to display a list of cases with the status of 'New'. The is named "NewCaseList". The custom controller Apex class is named "NewCaseListController" and includes the following: A publicly scoped method named "getNewCases", with the return type of List<Case>. Returns a list of case records that includes the ID and CaseNumber fields. Filtered the results returned to only have a status of "New". The NewCaseList Visualforce page used 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.

The screenshot shows the Salesforce Developer Console interface. At the top, there's a tabs bar with various Apex files and a 'NewCaseListController.vfp' tab which is currently selected. Below the tabs, there's a preview area showing the Visualforce markup. The code is as follows:

```
1 <apex:page controller="NewCaselistController" >
2   <apex:repeat var="case" value="{!Newcases}">
3     <li>
4       <apex:outputLink value="/{!case.id}">{!case.id}</apex:outputLink>
5       {!case.CaseNumber}
6     </li>
7   </apex:repeat>
8 </apex:page>
```

Below the code, there's a 'Logs' section and a 'Tests' section. The 'Tests' section has a table titled 'Overall Code Coverage' showing coverage for several classes. The table looks like this:

Class	Percent	Lines
DiscountClass	0%	0/4
BillingRecordInsertion	0%	0/12
CustomerTrigger	100%	9/9
NewCaselistController	0%	0/4
NormalCustomer	0%	0/2



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](#)