



Capstone Project Report  
On  
**Travel Application**  
**And**  
**Code Playground Application**

Submitted by: Ravi Teja Kuchipudi

**Name:** Ravi Teja Kuchipudi

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

**LMS Id:** MGSA\_519

**Date:** 28/02/2023

## Table of contents

|   |    |
|---|----|
| 1. Abstract.....                            | 2  |
| 2. Introduction.....                        | 3  |
| 3. Flow of the project.....                 | 4  |
| 4. Software and Hardware Requirements.....  | 5  |
| 5. Screen shots.....                        | 6  |
| 5.1. Module 1.....                          | 6  |
| 5.2. Module 2.....                          | 20 |
| 5.3. Module 3.....                          | 34 |
| 6. References.....                          | 62 |
| 7. <a href="#"><u>Github Link</u></a> ..... | 63 |

## **Abstract**

Salesforce is a comprehensive and flexible platform that enables businesses to manage their customer relationships, streamline their sales and marketing processes, and promote business growth.

There are several ways to adjust the user interface in your Salesforce org to meet your needs. Many customizations can be made through configuration menus and drag-and-drop page layout editors, all with no coding required. This project contains three modules. First module focuses on the non-coding approach to modifying the user interface, as you customize an app that tracks company travel.

Within the Salesforce Platform there are several features and tools that let you streamline and automate your business processes. As you discover in this project, the Salesforce Lightning Platform makes business process automation a snap and doesn't require any code. This Second module focuses on the Automation and Analytics Tools to enhance your custom App development.

Apex enables developers to build complex business processes, customize user interfaces, and integrations with third-party systems, when declarative tools aren't up to the task. This last module focuses on the coding approach to extend the functional capabilities of your custom App.

## Introduction

Salesforce is a cloud-based Customer Relationship Management (CRM) platform that helps businesses manage and streamline their customer interactions. The platform offers a suite of tools that enable organizations to effectively manage their sales, marketing, customer service, and analytics functions all in one place.

With Salesforce, businesses can manage their entire customer lifecycle, from lead generation to customer retention, all while providing their employees with real-time access to customer data and insights. The platform offers a wide range of integrations with other software solutions, allowing businesses to easily connect Salesforce with their existing systems and workflows.



Figure 1: Salesforce Cloud Services ([Source](#))

Talking about Travel Application and Code playground Application, Travel Application focuses on the non-coding approach to modifying the user interface, as you customize an app that tracks company travel and also focuses on the Automation and Analytics Tools to enhance your custom App development. By the end of the Travel Application , we will have a working prototype of the new travel approval application and also a working custom Travel Approval App with added Business logics and analytics to improve the user experience. Code Playground Application focuses on the coding approach to extend the functional capabilities of your custom App.

## Flow of the Project

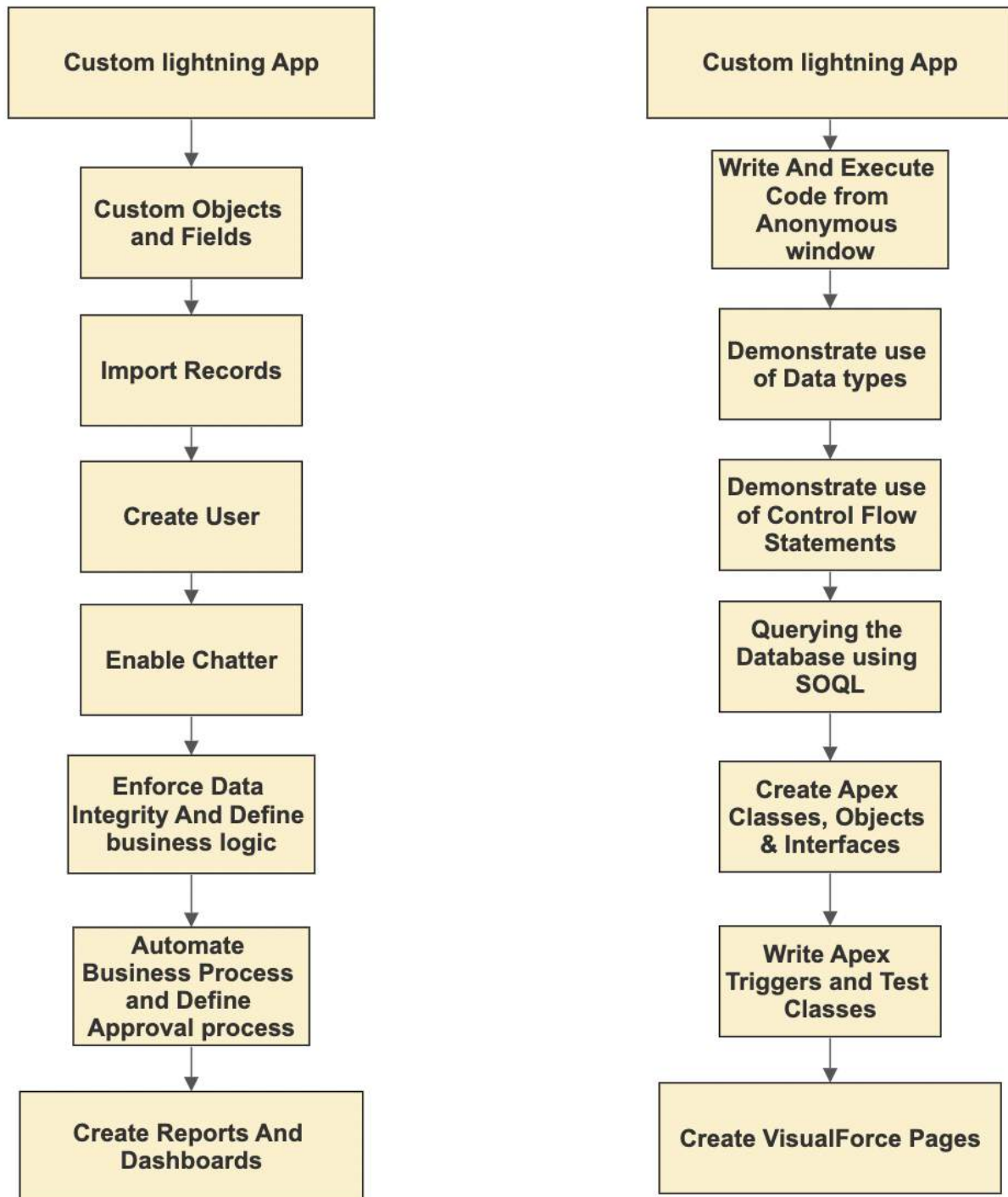


Figure 1: Flowcharts.

## Software and Hardware Requirements

### Software Requirements:

1. **Web browser:** Latest version of Google Chrome, Mozilla Firefox, Apple Safari or Microsoft Edge
2. **Operating System:** Windows 10, Mac OS X, or any Linux distribution that supports the latest versions of Firefox or Chrome
3. **Internet Connection:** 1 Mbps or faster

### Hardware Requirements:

1. **Processor:** Minimum 2 GHz
2. **RAM:** Minimum 4 GB
3. **Storage Space:** Minimum 20 GB

# Screen Shots

## Module 1

### Business Requirements:

- Set up a Lightning app to streamline the travel approval process.
- You need to create a custom application that meets these requirements: Build a Lightning app, add tabs, and customize page layouts.
- Create custom objects and fields for the app.
- Define relationships between objects.
- Import data.
- Create Travel Approvals, Travel Expenses & Users.
- Customize List views, Page Layouts, Related Lists.
- Setup Approval settings for User.
- Enable Chatter on the Travel Approval Object.

### Exercise 1:

#### Step 1:

1. Create a new custom lightning App, name: Travel App.
2. Upload Travel.png as App image, Keep “Org Theme options” checked.
3. Navigation Style = ‘Standard Navigation’, leave rest as it is.
4. Select and add Chatter, Reports & Dashboards to the Navigation items.
5. User Profile = “System Administrator”.
6. Save & Finish.

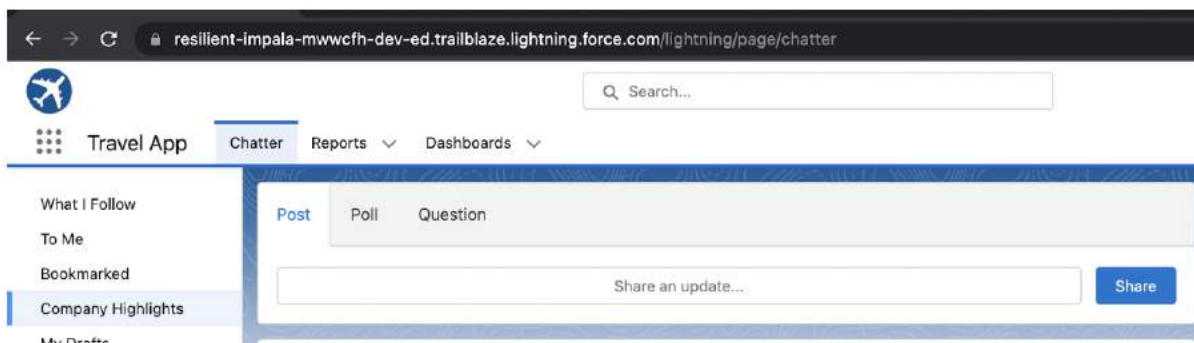


Figure 3

## Step 2:

1. Create a Department custom object.
2. Label = “Department” ,Plural = “Departments” ,Data Type = “Text”
3. Select “Allow Reports”, “Allow Search”. Keep other options as it is.
4. Use the custom tab of your choice and Include it only for the Travel App.
5. Tab visibility = As it is, Save.

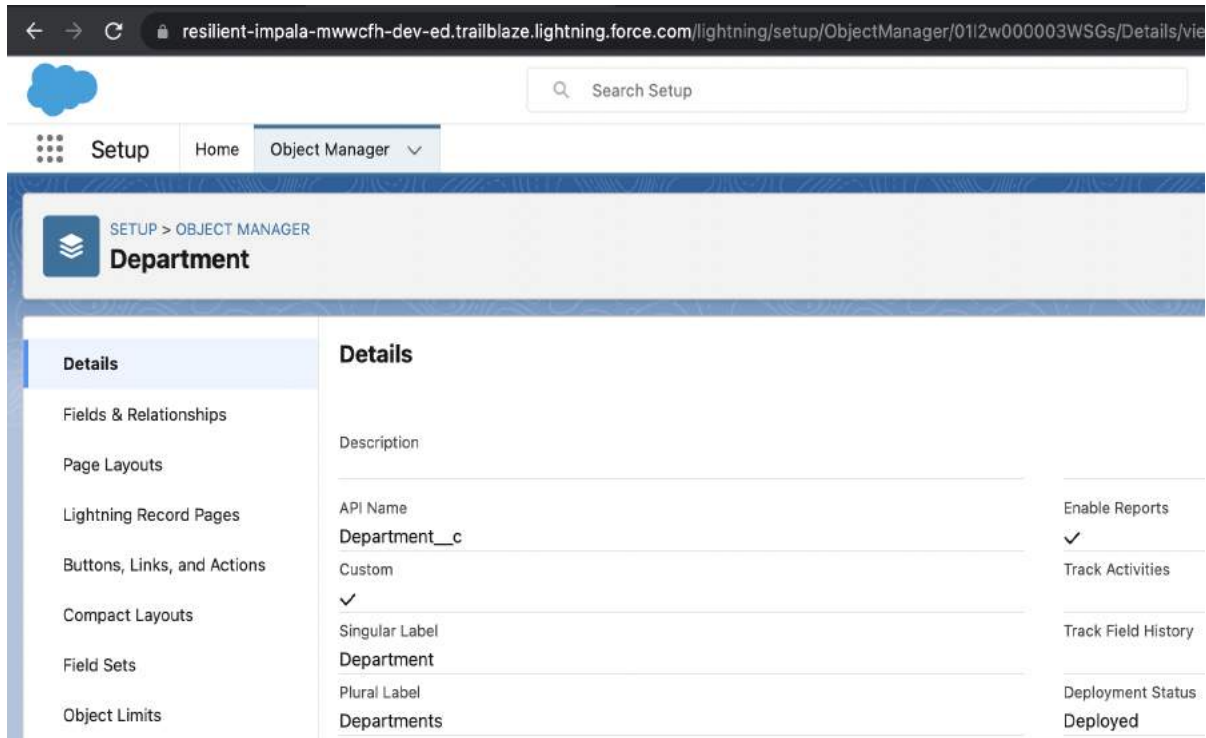


Figure 4

## Step 3:

1. Department Code, Text, Length = 10, Required, Select Unique & Case sensitive.
2. Location, Picklist, Value: Kolkata, Delhi.
3. Department Type, Picklist, Values: Banking, Finance, Education, Energy, IT.
4. Create Field Dependency, Controlling field = Location, Dependent field = Department Type.



| 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)               |             |

Figure 5

#### Step 4:

1. Create a Travel Approval Object.
2. Label = “Travel Approval”, Starting Number = 1., Plural = “Travel Approvals”, Object Name = “Travel\_Approval”
3. Record Name = “Travel Approval #”, Data Type = “Auto Number”, Display format “TA{00000}”.
4. Select additional options(Allow Reports, Allow Activities, Track Field History, Allow Search, Add Notes & Attachments.)
5. Create a custom Tab of your choice, Add the Tab only to the Travel App.
6. Create Custom fields:
  - a. Purpose Of Trip, Text Area.
  - b. Status, Picklist, Values = New, Submitted, Pending Approval, Approved, Rejected, Draft.
  - c. Trip Start Date, Date.
  - d. Trip End Date, Date.
  - e. Out Of State, Checkbox.
  - f. Destination State, Text, Length = 2.
  - g. Department, Lookup, Related To = Department custom object.

The screenshot shows the Salesforce Setup interface for the 'Travel Approval' object. The left sidebar contains a navigation menu with options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, Scoping Rules, Triggers, and Flow Triggers. The main content area is titled 'Fields & Relationships' and shows a list of 11 fields. The fields are sorted by Field Label. The table has columns for FIELD LABEL, FIELD NAME, DATA TYPE, and CONTROLLING FIELD.

| FIELD LABEL       | FIELD NAME           | DATA TYPE          | CONTROLLING FIELD |
|-------------------|----------------------|--------------------|-------------------|
| Created By        | CreatedById          | Lookup(User)       |                   |
| Department        | Department__c        | Lookup(Department) |                   |
| Destination State | Destination_State__c | Text(2)            |                   |
| Last Modified By  | LastModifiedById     | Lookup(User)       |                   |
| Out Of State      | Out_Of_State__c      | Checkbox           |                   |
| Owner             | OwnerId              | Lookup(User,Group) |                   |
| Purpose Of Trip   | Purpose_Of_Trip__c   | Text Area(255)     |                   |
| Status            | Status__c            | Picklist           |                   |
| Travel Approval # | Name                 | Auto Number        |                   |
| Trip End Date     | Trip_End_Date__c     | Date               |                   |
| Trip Start Date   | Trip_Start_Date__c   | Date               |                   |

Figure 6

The screenshot shows the 'Travel App' interface for a 'Travel Approval' record with ID 'TA00001'. The top navigation bar includes 'Travel App', 'Chatter', 'Reports', 'Dashboards', 'Departments', and 'Travel Approvals'. The main content area is titled 'Travel Approval TA00001' and has tabs for 'Related' and 'Details'. The 'Details' tab is active, showing a list of fields and their values. The fields are: Travel Approval # (TA00001), Purpose Of Trip (Business Trip to USA), Status (New), Trip Start Date (21/02/2023), Trip End Date (25/02/2023), Out Of State (checked), Destination State (US), Department (Test 1), Created By (Ravi Teja Kuchipudi, 20/02/2023, 8:43 pm), Owner (Ravi Teja Kuchipudi), and Last Modified By (Ravi Teja Kuchipudi, 20/02/2023, 8:43 pm).

| Field             | Value                                    |
|-------------------|--|
| Travel Approval # | TA00001                                  |
| Purpose Of Trip   | Business Trip to USA                     |
| Status            | New                                      |
| Trip Start Date   | 21/02/2023                               |
| Trip End Date     | 25/02/2023                               |
| Out Of State      | <input checked="" type="checkbox"/>      |
| Destination State | US                                       |
| Department        | Test 1                                   |
| Created By        | Ravi Teja Kuchipudi, 20/02/2023, 8:43 pm |
| Owner             | Ravi Teja Kuchipudi                      |
| Last Modified By  | Ravi Teja Kuchipudi, 20/02/2023, 8:43 pm |

Figure 7

## Step 5:

1. Import Departments
2. Download The Department.CSV File
3. Open “Data Import Wizard”, Click “Launch Wizard”
4. Select “Departments” Custom Object
5. Add New Records Upload the file by clicking the CSV icon
6. Map the Fields, click Next
7. Start Start Import
8. Make sure all the records were inserted.

resilient-impala-mwwcfh-dev-ed.trailblaze.lightning.force.com/lightning/o/Department\_\_c

Travel App Chatter Reports Dashboards Departments Travel Approvals

Departments All

16 items • Sorted by Department Name • Filtered by All departments • Updated a few seconds ago

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

Figure 8

## Exercise 2:

### Step 1:

1. Create a Travel Approval record.
2. 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 a Salesforce record for a Travel Approval with ID TA00002. The interface includes a top navigation bar with a search bar and tabs for Travel App, Chatter, Reports, Dashboards, Departments, and Travel Approvals. The record details are displayed in a table-like format under the 'Details' tab.

| Field             | Value                                    | Owner                                    |
|-------------------|--|--|
| Travel Approval # | TA00002                                  | Ravi Teja Kuchipudi                      |
| Purpose Of Trip   | Attend Dreamforce                        |  |
| Status            | Draft                                    |  |
| Trip Start Date   | 20/02/2023                               |  |
| Trip End Date     | 24/02/2023                               |  |
| Out Of State      | <input checked="" type="checkbox"/>      |  |
| Destination State | CA                                       |  |
| Department        | Technology                               |  |
| Created By        | Ravi Teja Kuchipudi, 20/02/2023, 8:53 pm | Last Modified By                         |
|                   |  | Ravi Teja Kuchipudi, 21/02/2023, 9:44 am |

Figure 9

### Step 2:

1. Create an Expense Item Object, Label = “Expense Item”, Plural = “Expense Items”, Starts with vowel sound = Checked,
2. Record Name = Expense Item Number, Data Type = Auto Number, Display Format = E – {00000}, Starting Number = 1
3. Allow Reports, Do not create Custom Tab, Save.

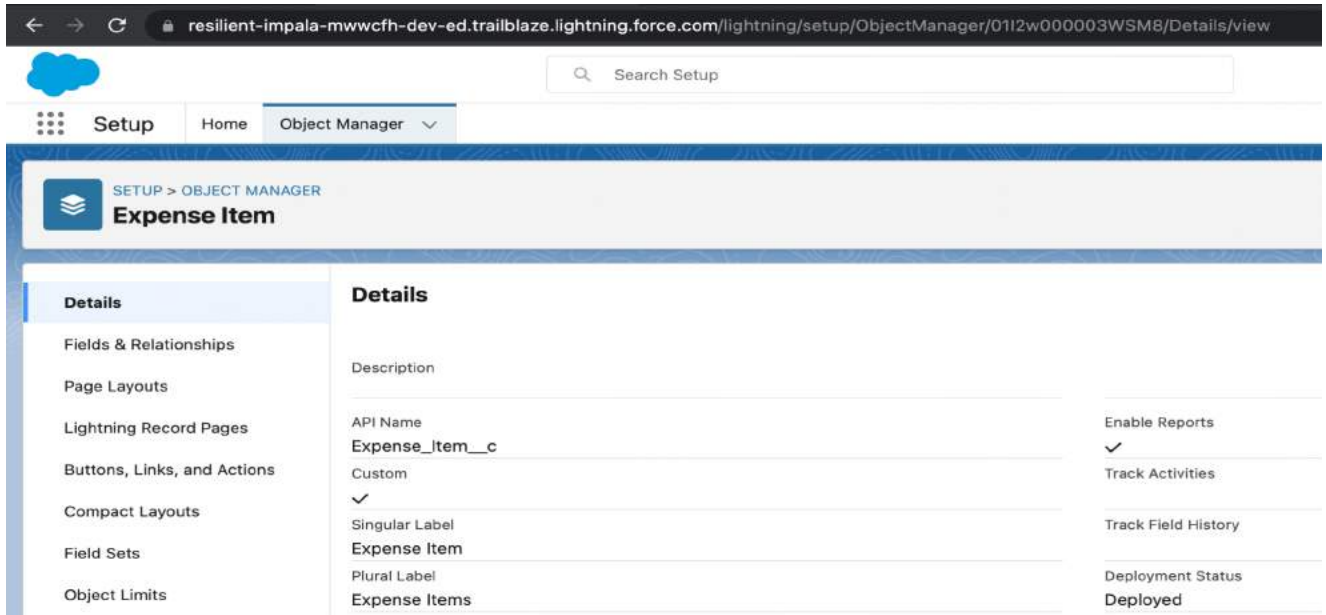


Figure 10

### Step 3:

1. Create the following custom fields:
  - a. Amount, Length = 16, Type = Currency, Decimal = 2, Required = True.
  - b. Expense Type, Type = Picklist, Values = Airfare, Hotel, Rental Car, Meals, Other, Required = True.
  - c. Travel Approval, Type = Master-Detail, Related To – Travel Approval.

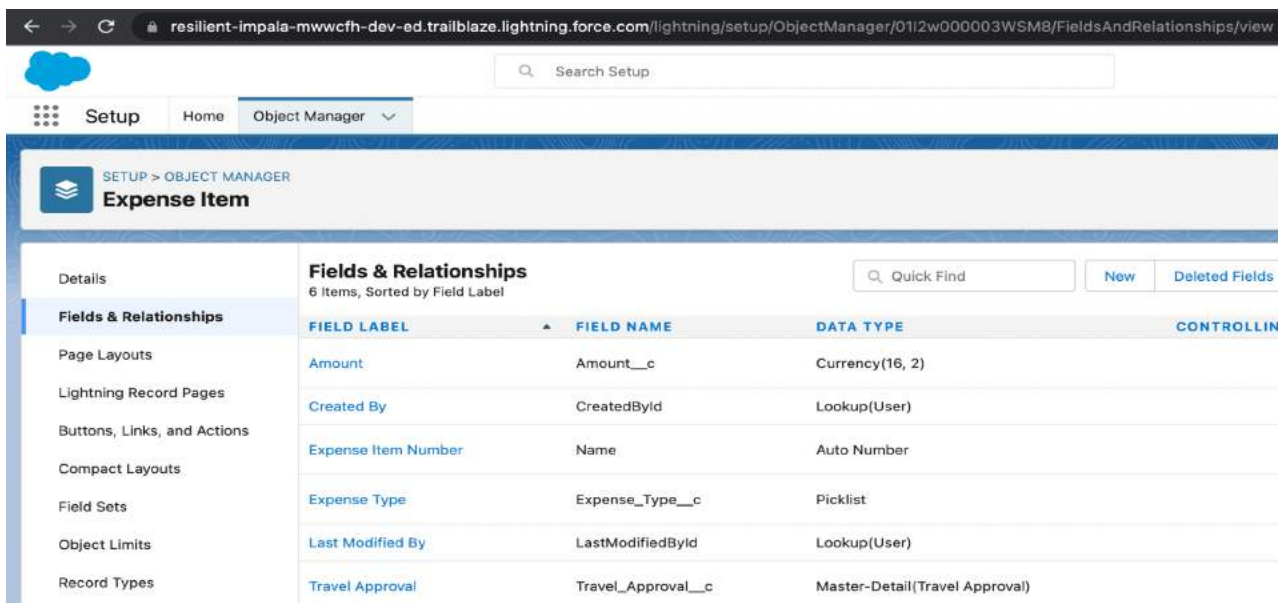


Figure 11

#### Step 4:

1. Create Expense Items.
  - a. Amount = 870, Expense Type = “Hotel”, Save.
  - b. Amount = “450”, Expense Type = “Airfare”, Save.

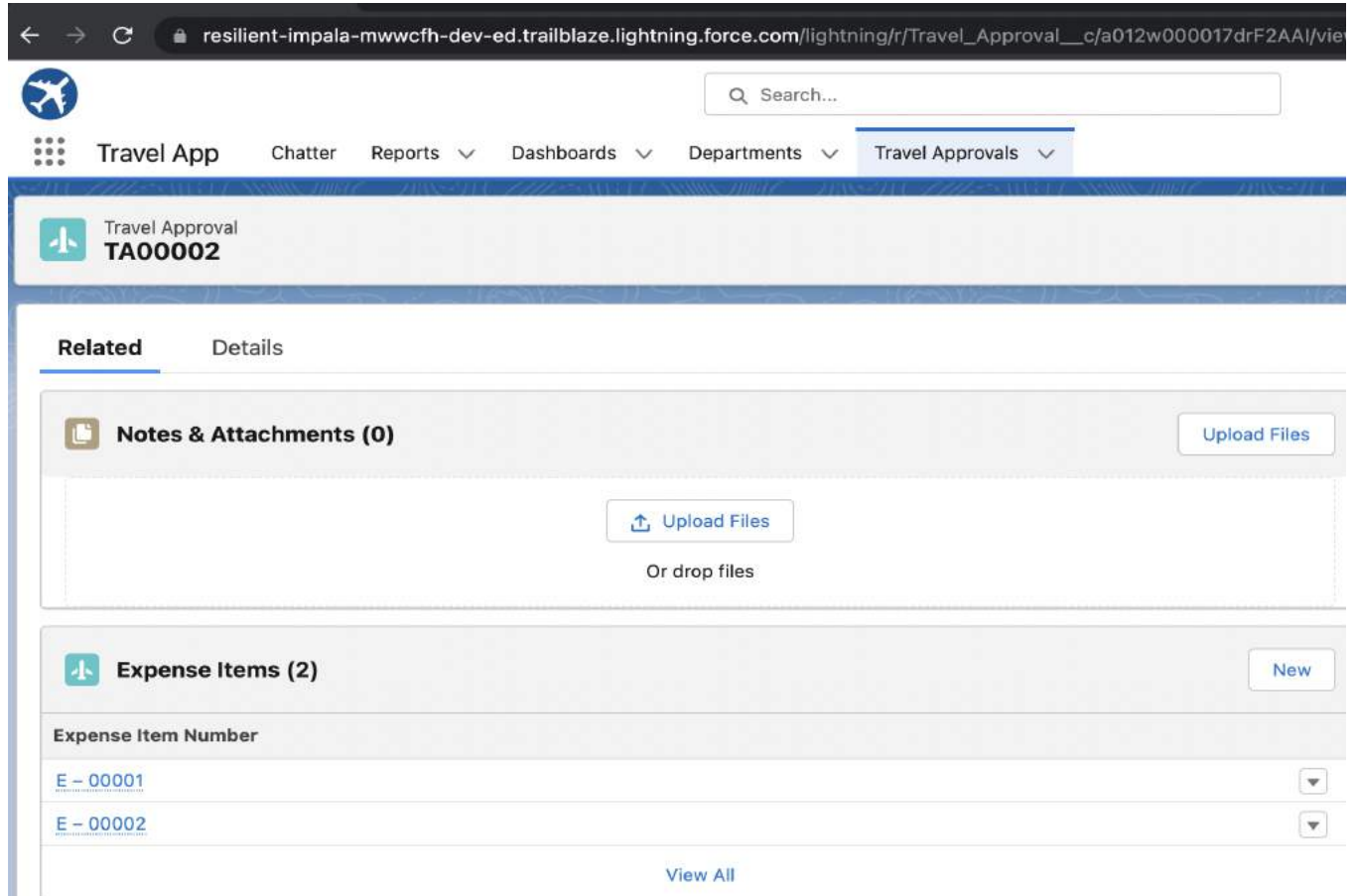


Figure 12

#### Step 5:

1. Create a User.
2. First Name = “Eric”, Last Name = “Executive”, Email = “Use your own email”, Username Name = “Choose a Unique username”
3. Role = “CEO”
4. License = Salesforce.
5. Profile = System Administrator.
6. Save.



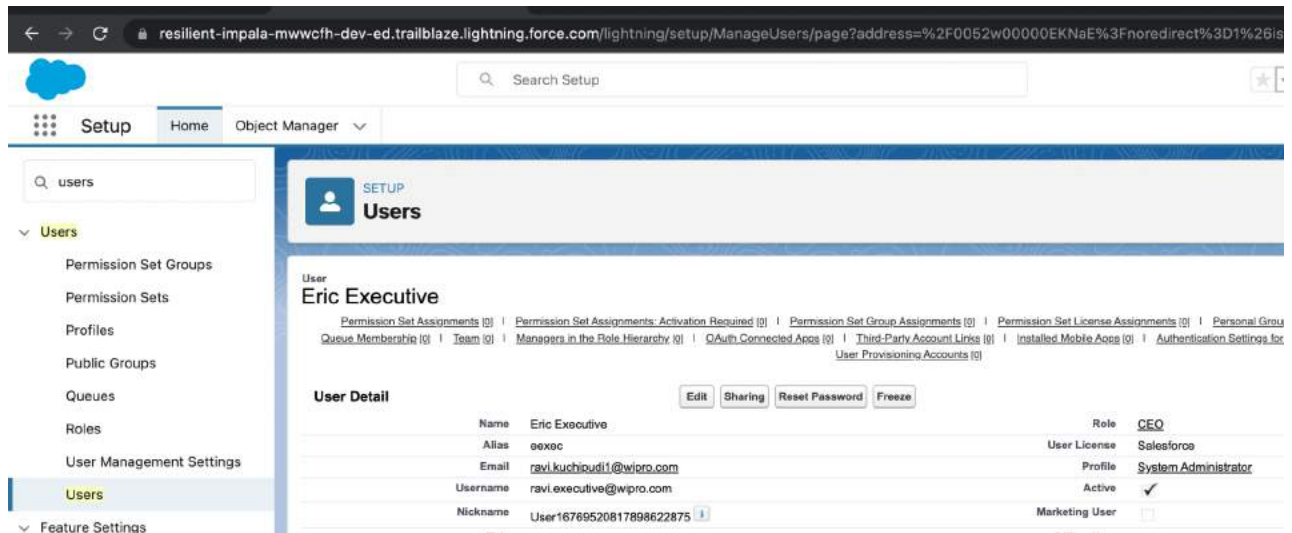


Figure 13

## Step 6: Add user Eric Executive as your manager

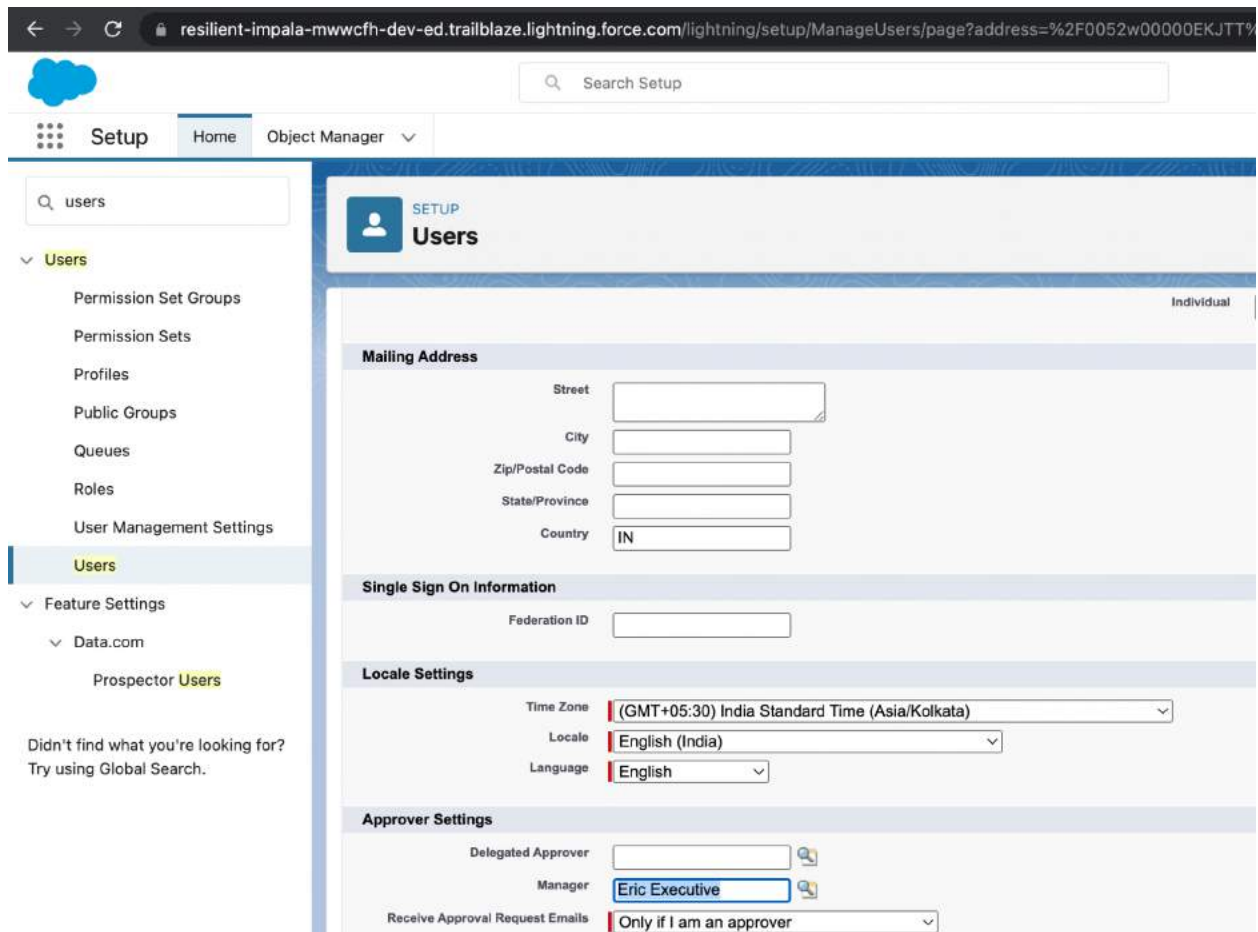


Figure 14

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

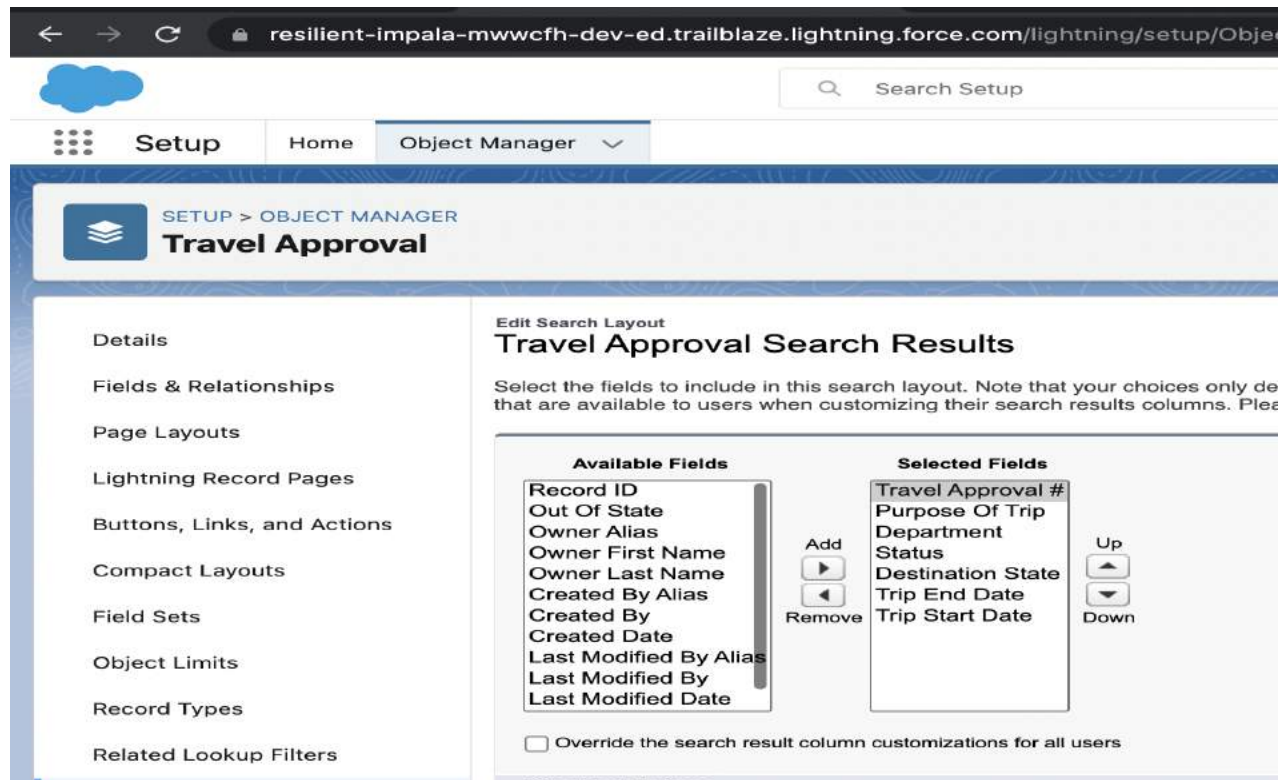


Figure 15

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

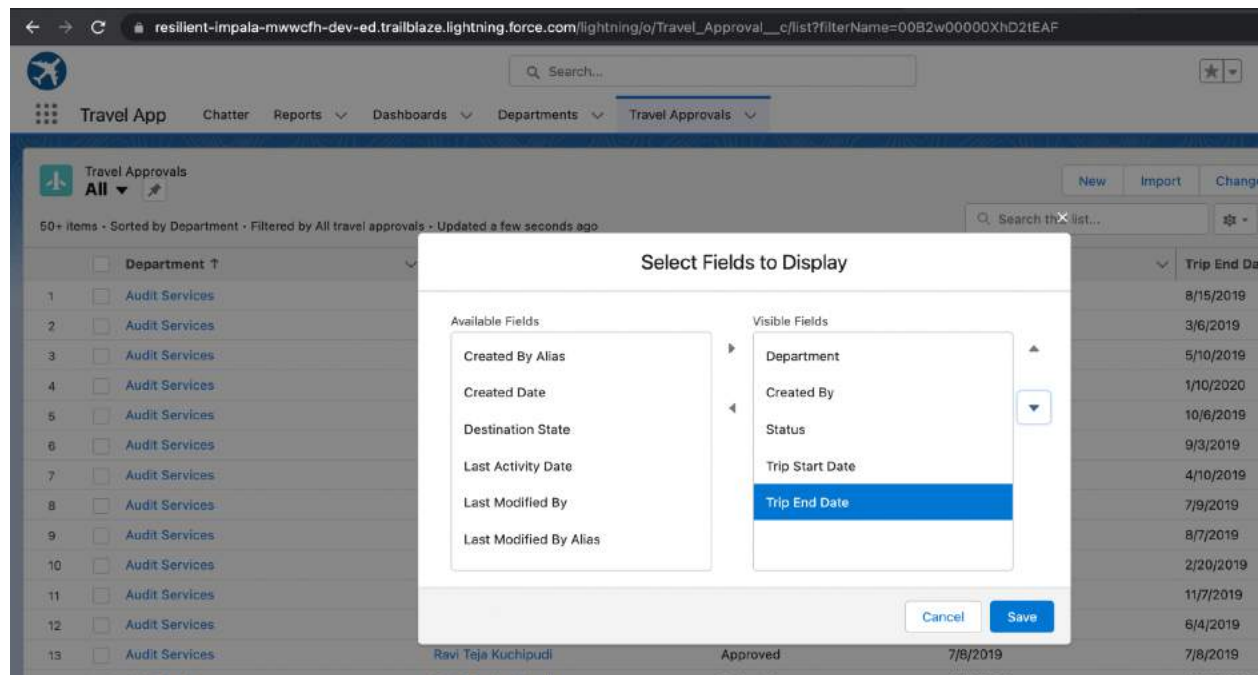


Figure 16



## Step 9:

1. Create Travel approval custom List View “Open Out of State Travel Requests”.
2. All users should be able to see this list view.

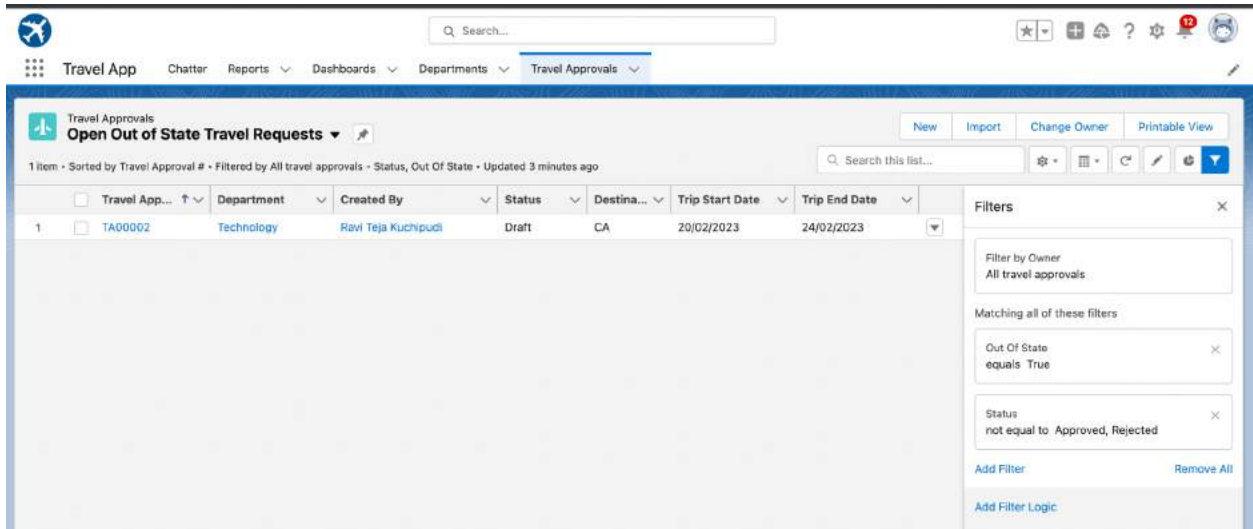


Figure 17

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

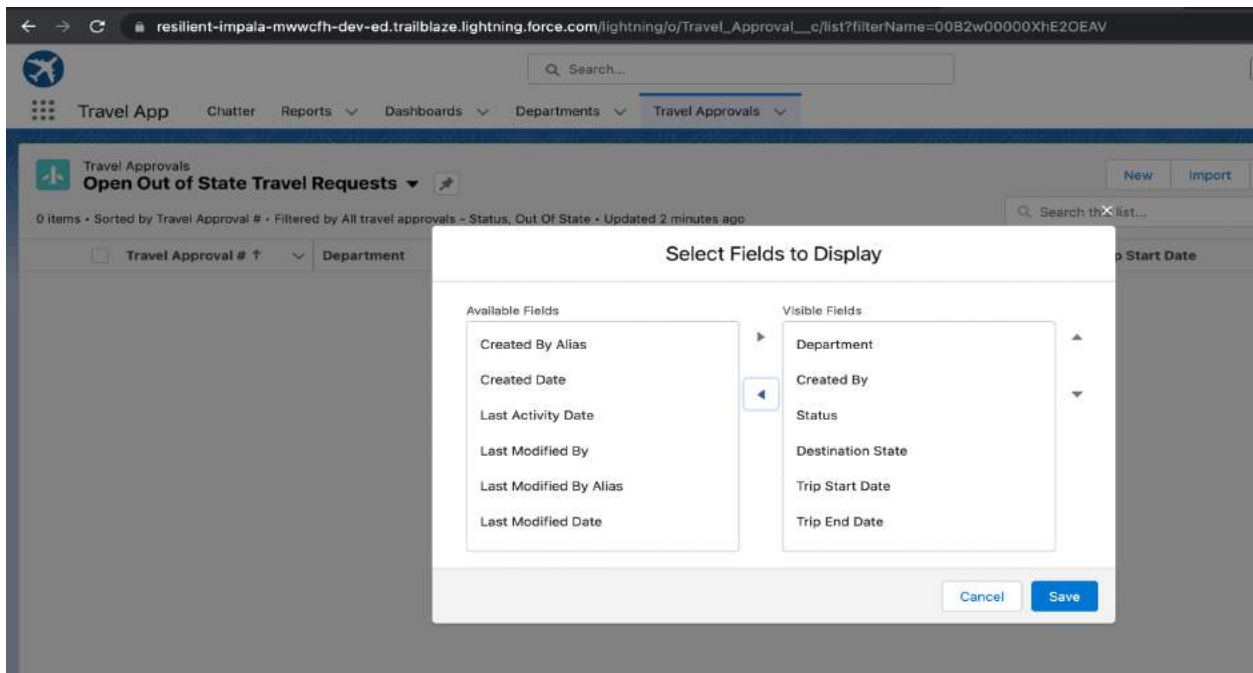


Figure 18

## Step 11:

1. Add a section to the page layout called “Trip Info” and add the fields.

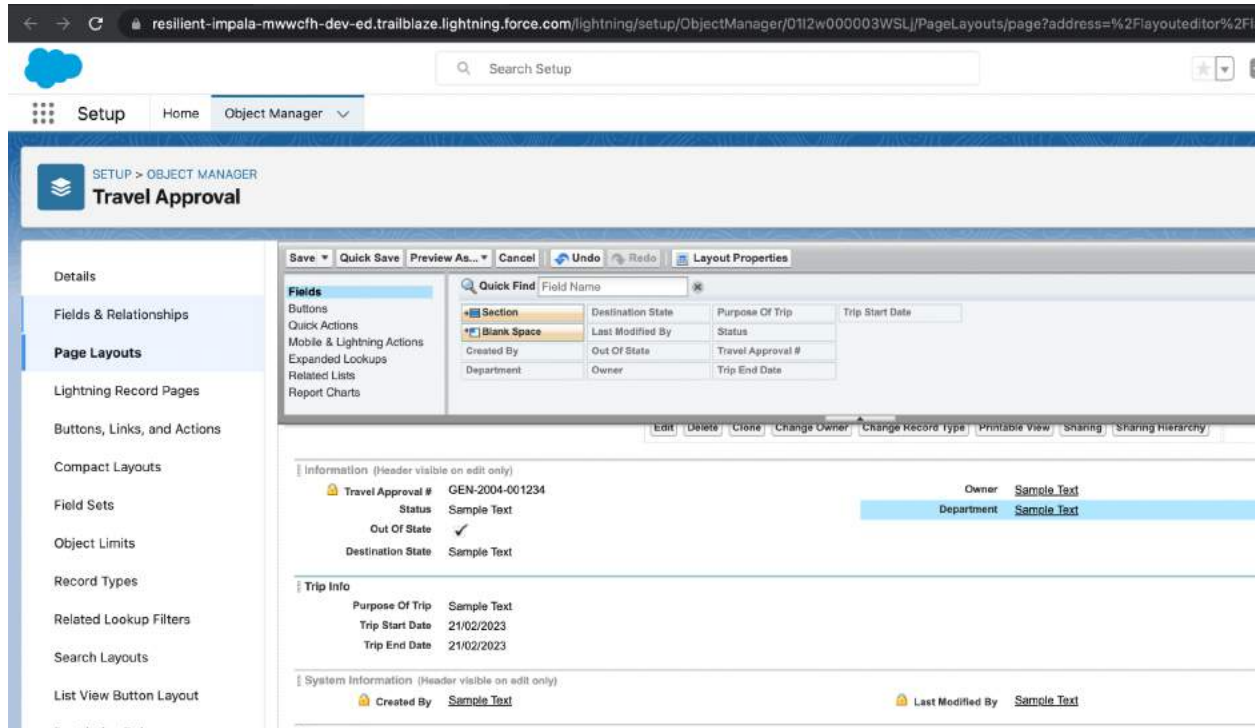


Figure 19

## Step 12: Customize the Expense Item Related List under the Travel Approval page

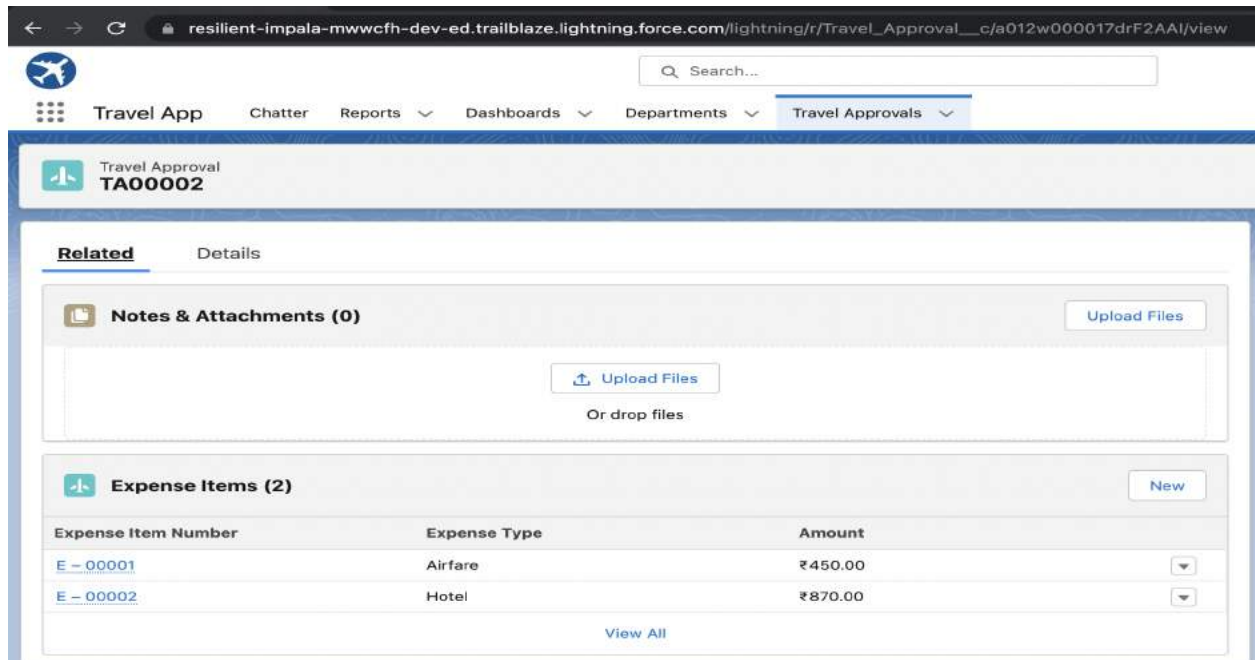


Figure 20

### Step 13:

1. Enable Chatter and “Feed Tracking” for Travel Approval Object.
2. Select these 2 fields: Destination State Status Save.
3. Open a Travel Approval record. Click on Chatter Tab. Share a post
4. Login in as Eric and reply to the email.

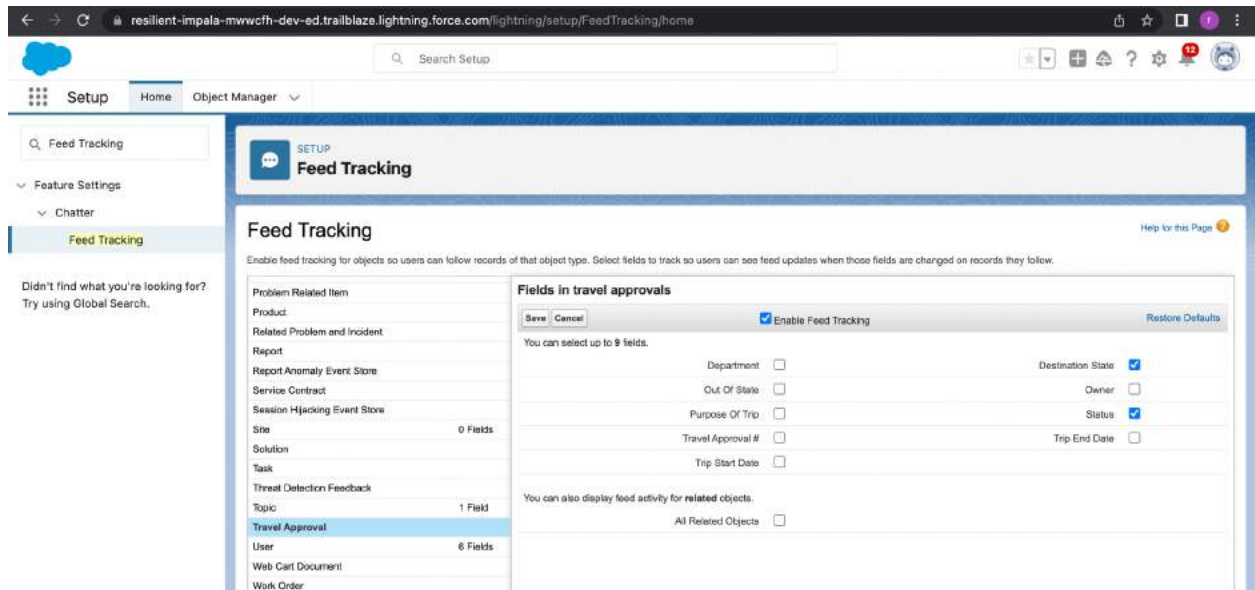


Figure 21

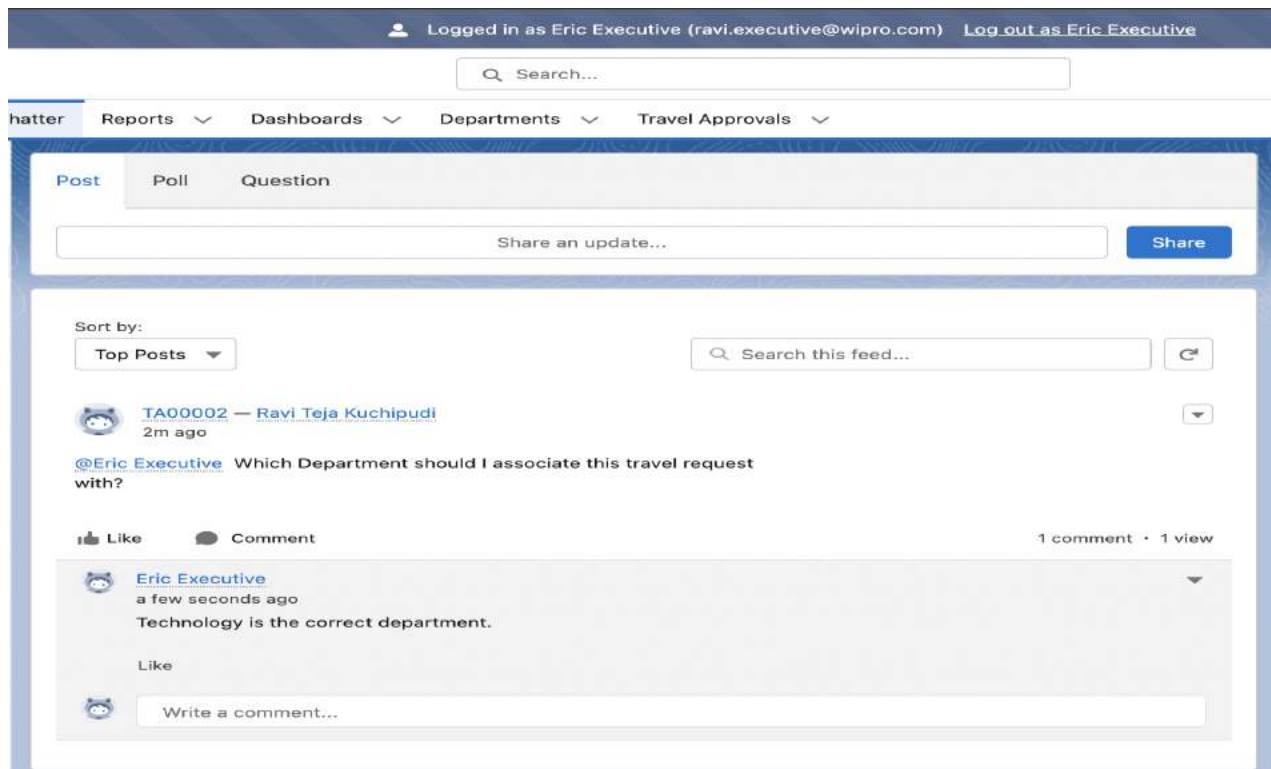


Figure 22

## Test The App:

The screenshot displays a Salesforce Lightning interface for a Travel Approval record (TA00002). The top navigation bar includes a search bar and a menu with options like Chatter, Reports, Dashboards, Departments, and Travel Approvals. The main content area is divided into two panels: 'Details' and 'Activity'.

**Details Panel:**

| Field             | Value                                   |
|-------------------|---|
| Travel Approval # | TA00002                                 |
| Status            | Approved                                |
| Owner             | Ravi Teja Kuchipudi                     |
| Department        | Technology                              |
| <b>Trip Info</b>  |   |
| Purpose Of Trip   | Attend Dreamforce                       |
| Out Of State      | <input checked="" type="checkbox"/>     |
| Trip Start Date   | 2/20/2023                               |
| Destination State | CA                                      |
| Trip End Date     | 2/22/2023                               |
| Created By        | Ravi Teja Kuchipudi, 2/20/2023, 8:53 PM |
| Last Modified By  | Eric Executive, 2/21/2023, 3:16 PM      |

**Activity Panel:**

The Activity panel shows a list of activities. The current view is 'Upcoming & Overdue', which displays no activities. The panel includes filters for 'All time', 'All activities', and 'All types', along with buttons for 'Refresh', 'Expand All', and 'View All'.

Figure 23

## Module 2

### Business Requirements:

- Add Business logic & Analytics to the Travel Approval App.
- You will be adding Business logics, Reports and Dashboards to:
- Enforce data integrity with validation rules.
- Define business logic using roll-up summary and formula fields
- Automate business processes with Flows.
- Define Approval Process for Travel Approval App.
- Create Reports to analyze Travel Approvals.
- Build Dashboards for Travel Approval Management.

### Exercise 1:

#### Step 1:

1. Create Validation Rule.
2. Trip end date must always be greater than ( $\geq$ ) the trip start date.
3. Name: Trip end date after start date.
4. Make sure to keep “Active” selected/checked.
5. Error Message: Trip end date must be greater than or equal to start date.
6. Error Location: Select Field and pick Trip End Date as the location.

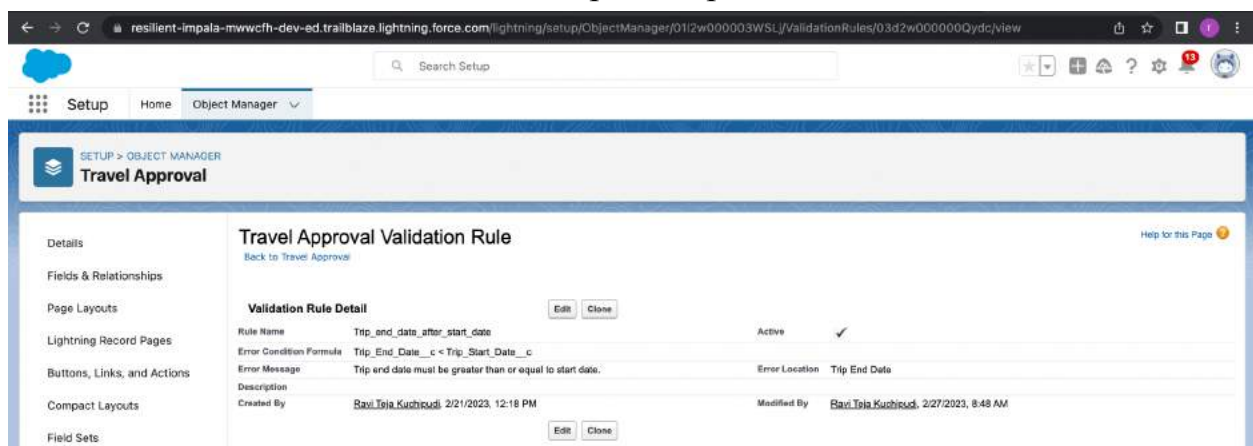


Figure 24

**Trip Info**

Purpose Of Trip  
Attend Dreamforce

Trip Start Date  
20/02/2023

Trip End Date  
10/02/2023

Trip end date must be greater than

**We hit a snag.**  
Review the following fields  
• [Trip End Date](#)

Created By  
Ravi Teja Kuchipudi, 20/02/2023, 8:53 pm

Modified By  
Ravi Teja Kuchipudi, 21/02/2023, 12:38 pm

Cancel Save

Figure 25

## Step 2:

1. Create a Roll-Up Summary Field on Travel Approval object.
2. Business Logic: Automatically sum up the total amount of expenses from the related Expense Item.
3. Field Label = Total Expenses
4. Field Name = Total\_Expenses
5. Roll-Up Type: SUM
6. Field to Aggregate: Amount
7. Filter Criteria: All records should be included in the calculation and Save.

resilient-impala-mwcfh-dev-ed.trailblaze.lightning.force.com/lightning/setup/ObjectManager/0112w000003WSLJ/FieldsAndRelationships/00N2w00000U0t26N/view

Setup Home Object Manager

**Travel Approval**

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

**Travel Approval Custom Field**  
**Total Expenses**  
[Back to Travel Approval](#)

[Edit](#) [Set Field-Level Security](#) [View Field Accessibility](#) [Where is this used?](#)

**Custom Field Definition Detail**

| Field Information         |   |
|---------------------------|---|
| Field Label               | Total Expenses                            |
| Field Name                | Total_Expenses                            |
| API Name                  | Total_Expenses__c                         |
| Description               |   |
| Help Text                 |   |
| Data Owner                |   |
| Field Usage               |   |
| Data Sensitivity Level    |   |
| Compliance Categorization |   |
| Created By                | Ravi Teja Kuchipudi, 21/02/2023, 12:20 pm |
| Modified By               | Ravi Teja Kuchipudi, 21/02/2023, 12:20 pm |

**Roll-Up Summary Options**

| Roll-Up Summary    |  |
|--------------------|--|
| Data Type          | Roll-Up Summary  |
| Summarized Object  | Expense Item   |
| Field to Aggregate | Expense Item: Amount                                       |
| Filter Criteria    | All records should be included in the calculation and Save |

Summary Type: SUM

Figure 26

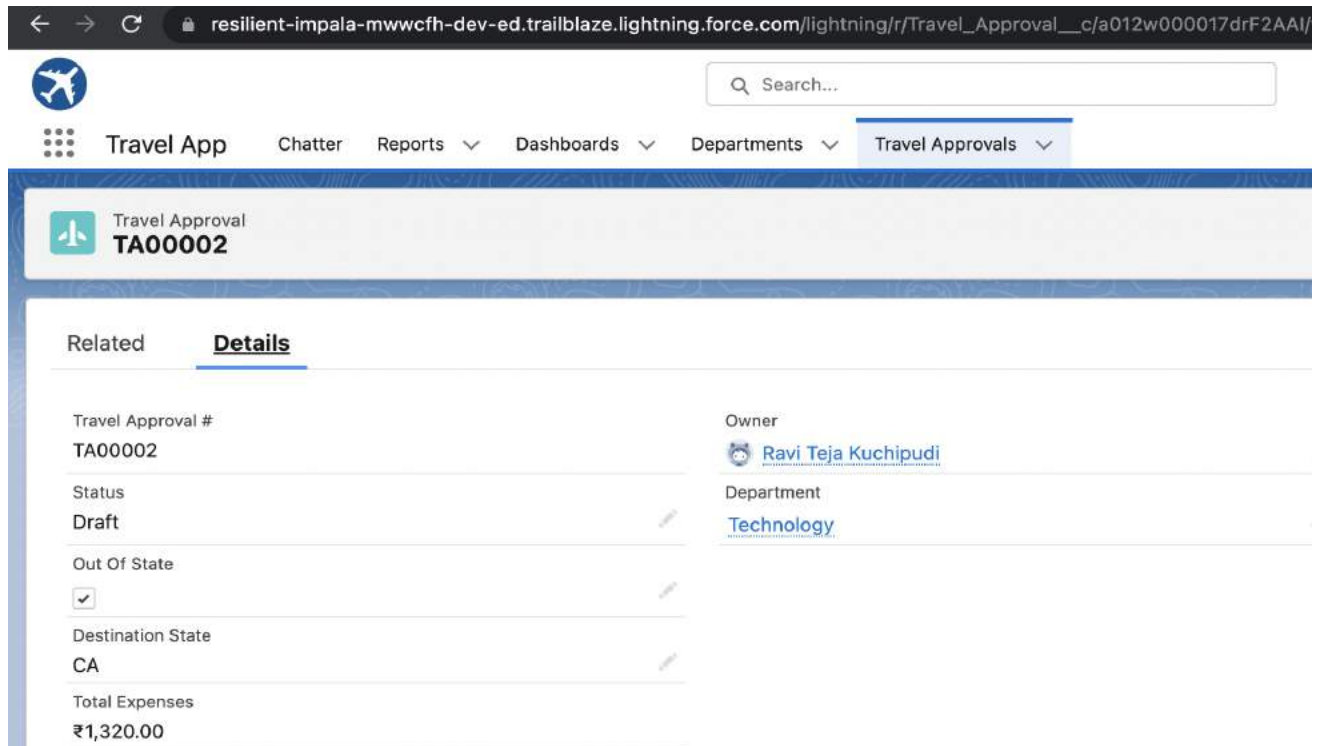


Figure 27

### Step 3:

1. Create Formula Fields.
2. Business Logic: Create a field that shows a visual indicator based on the value of the Status field.
3. Cache-Control = Public.
4. File = StatusImages.zip [upload it from the Project Folder]
5. Name = StatusImages
6. Setup | Custom Code | Static Resource | New
7. Save



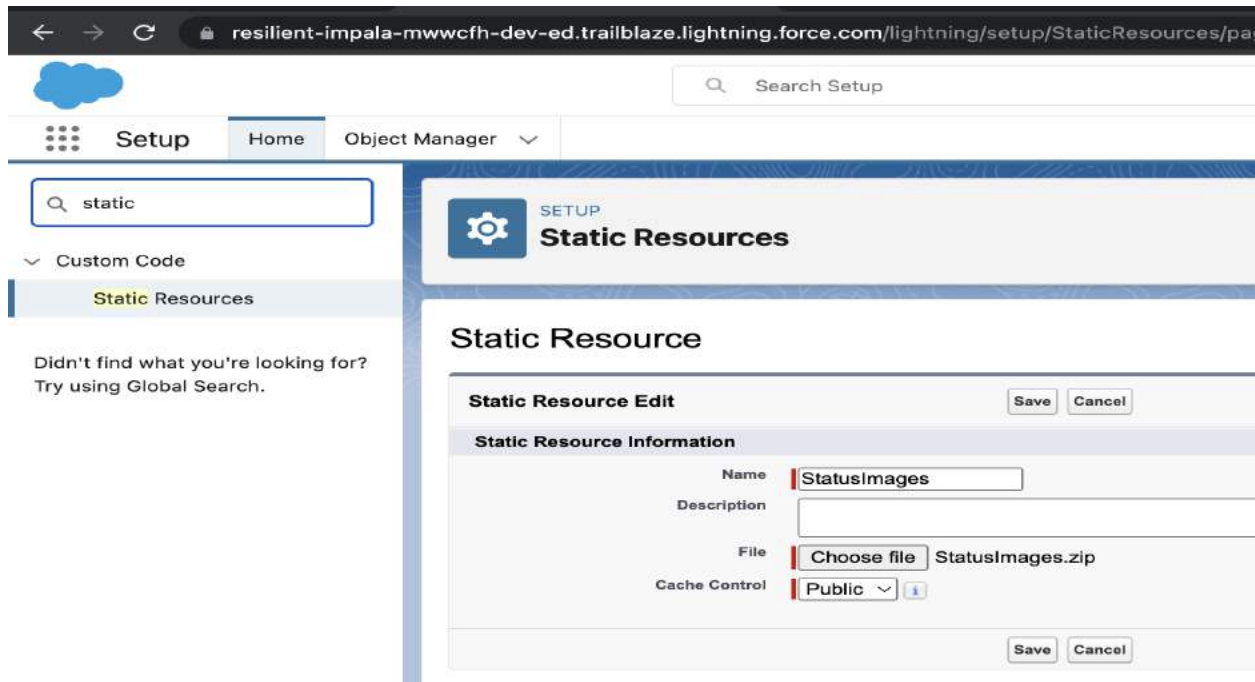


Figure 28

#### Step 4:

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

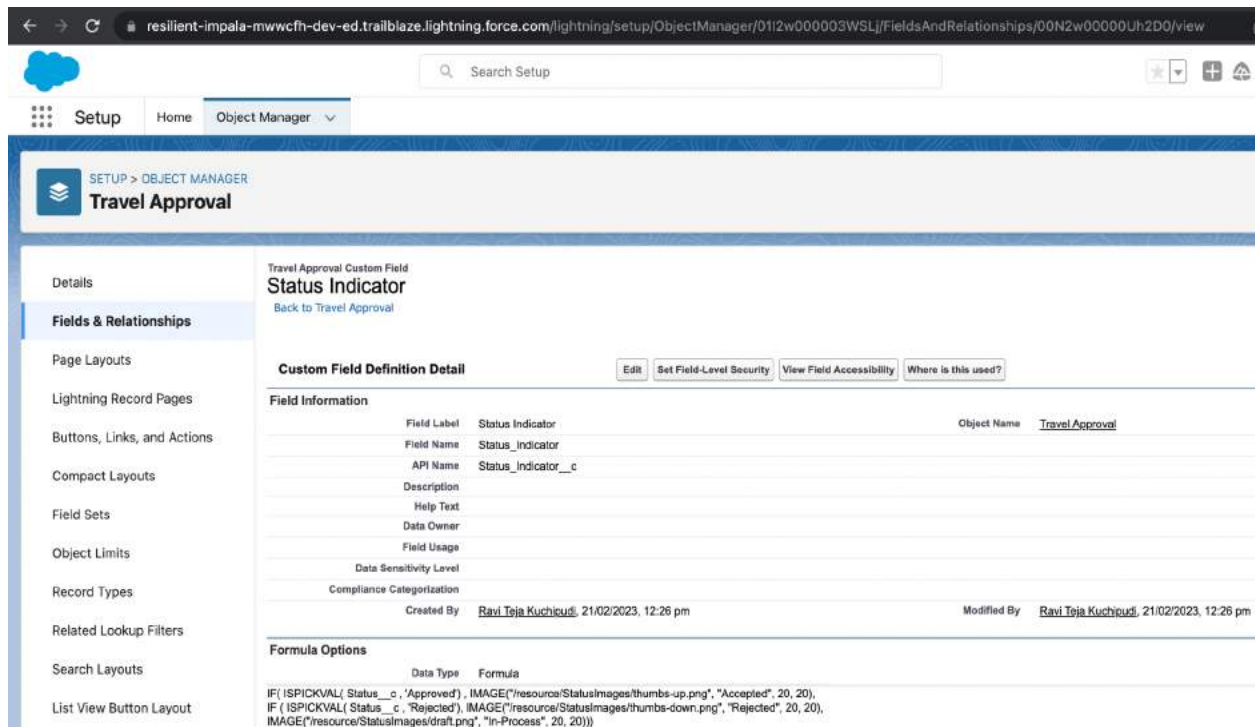


Figure 29



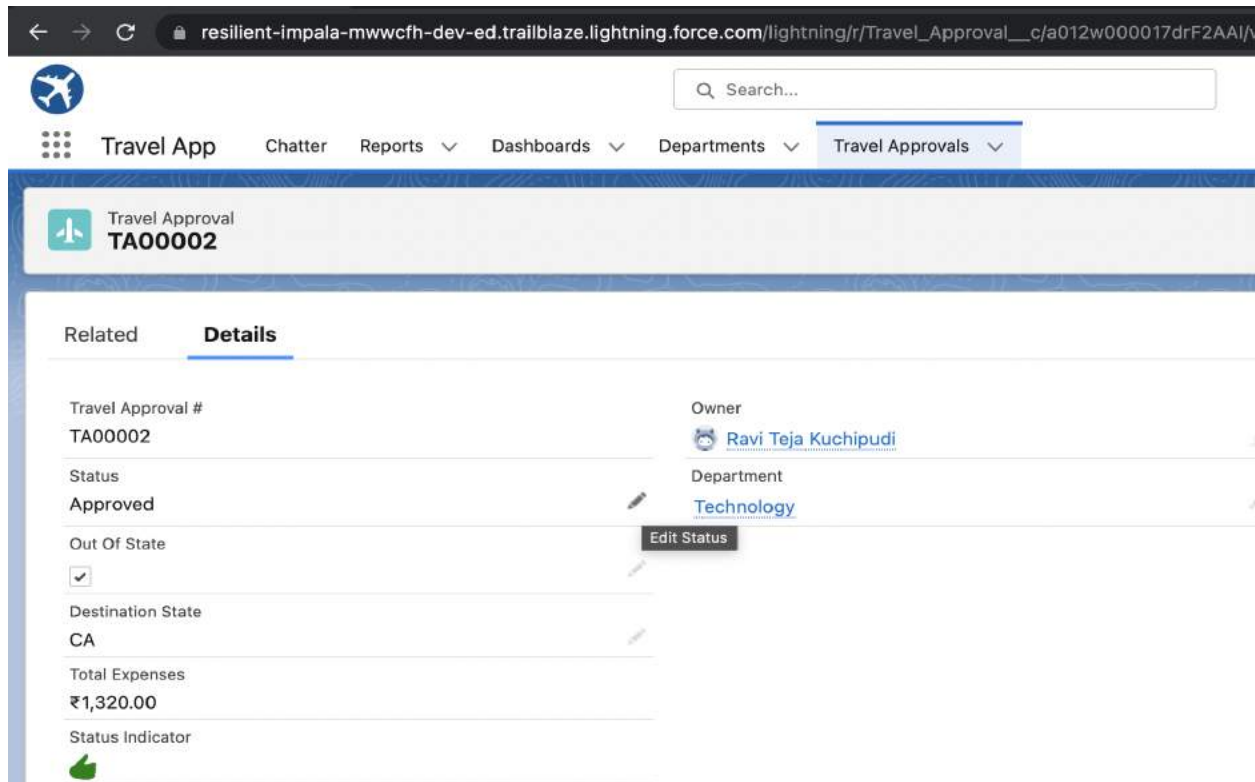


Figure 30

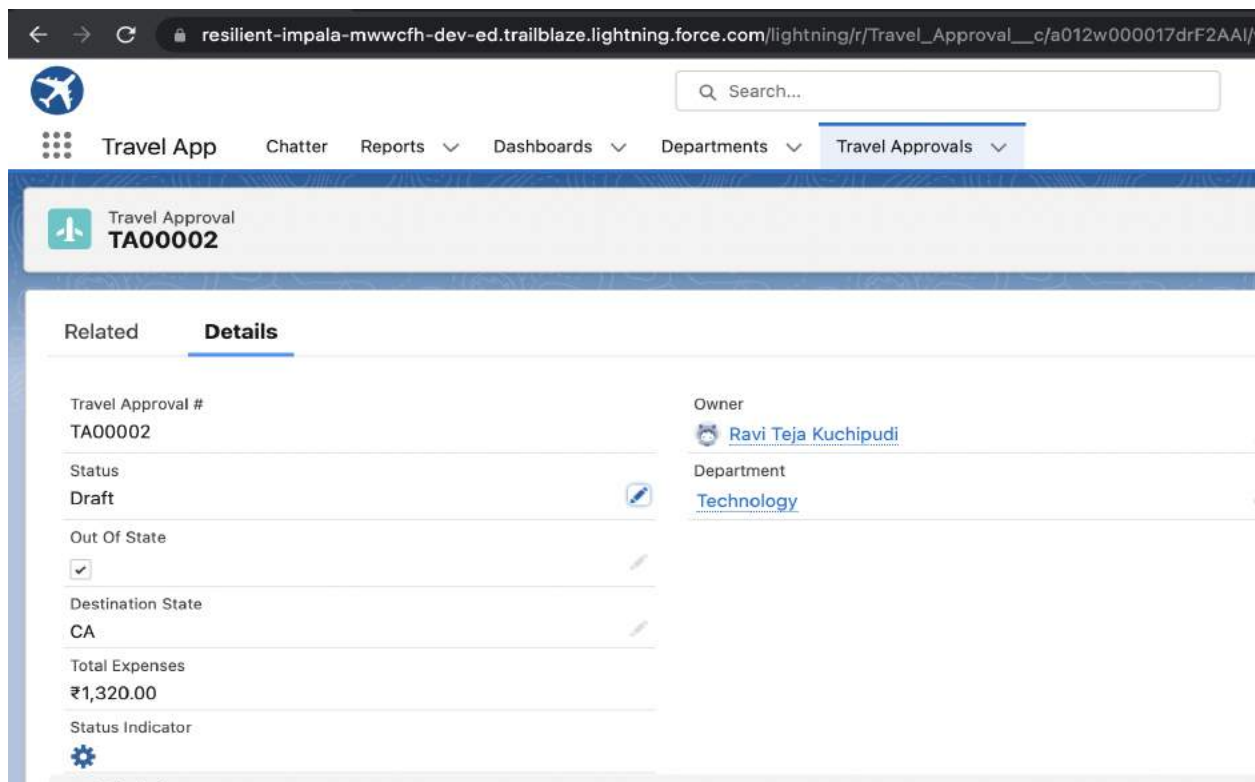


Figure 31

## Step 5:

### 1. Create a Record – Triggered Flow.

The screenshot shows the 'Configure Start' dialog in Salesforce Flow Builder. The title bar indicates the flow is 'Out of State Travel Flag - V1'. The dialog has a 'Travel Approval' field at the top. Below it, the 'Configure Trigger' section has a radio button group for 'Trigger the Flow When:' with options: 'A record is created', 'A record is updated', 'A record is created or updated' (selected), and 'A record is deleted'. The 'Set Entry Conditions' section includes a dropdown for 'Condition Requirements' set to 'None'. The 'Optimize the Flow for:' section has two options: 'Fast Field Updates' (selected with a checkmark) and 'Actions and Related Records'. The 'Fast Field Updates' option has a subtext: 'Update fields on the record that triggers the flow to run.' The 'Actions and Related Records' option has a subtext: 'Update any record and perform actions, like send an email.' At the bottom right are 'Cancel' and 'Done' buttons.

Figure 32

### 2. Add a Decision Element to the Flow.

The screenshot shows the 'Edit Decision' dialog in Salesforce Flow Builder. The title bar indicates the flow is 'Out of State Travel Flag - V1'. The dialog has a title 'Is Travel Out of State? (Is\_Travel\_Out\_of\_State)' with an edit icon. Below the title is the 'Outcomes' section with a subtext: 'For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.' The 'OUTCOME ORDER' section shows two outcomes: 'Yes Out of State' (selected) and 'In State'. The 'OUTCOME DETAILS' section for 'Yes Out of State' includes a 'Label' field with 'Yes Out of State', an 'Outcome API Name' field with 'Yes\_Out\_of\_State', and a 'Condition Requirements to Execute Outcome' dropdown set to 'All Conditions Are Met (AND)'. Below this is a table with columns 'Resource', 'Operator', and 'Value'. The first row has 'Record > Destination State' in the Resource column, 'Does Not Equal' in the Operator column, and 'TX' in the Value column. There is a '+ Add Condition' button below the table. The 'When to Execute Outcome' section has two radio buttons: 'If the condition requirements are met' and 'Only if the record that triggered the flow to run is updated to meet the condition requirements' (selected). At the bottom right are 'Cancel' and 'Done' buttons.

Figure 33

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

resilient-impala-mwcfh-dev-ed.trailblaze.lightning.force.com/builder\_platform\_interaction/flowBuilder.app?flowId=3012w000001CjLAAW

Flow Builder Out of State Travel Flag - V1

Free-Form Version 1: Active—Last modified a minute ago Run Debug View Tests Deactivate Save

### Edit Decision

Is Travel Out of State? (Is\_Travel\_Out\_of\_State)

Outcomes For each path the flow can take, create an outcome. For each outcome, specify the conditions that must be met for the flow to take that path.

OUTCOME ORDER +

- Yes Out of State
- In State**

Default Outcome

OUTCOME DETAILS

Delete Outcome

\* Label In State

\* Outcome API Name In\_State

Condition Requirements to Execute Outcome

All Conditions Are Met (AND)

Resource Operator Value

A \$Record > Destination State X Equals TX

+ Add Condition

When to Execute Outcome

☒ If the condition requirements are met

☐ Only if the record that triggered the flow to run is updated to meet the condition requirements

Cancel Done

Figure 34

4. Create an Action for the Flow Using Update Records Elements.

5. Value = \$GlobalConstant.True

resilient-impala-mwcfh-dev-ed.trailblaze.lightning.force.com/builder\_platform\_interaction/flowBuilder.app?flowId=3012w000001CjLAAW

Flow Builder Out of State Travel Flag - V1

### Edit Update Records

Update Salesforce records using values from the flow.

**Update Travel Approval Record (Update\_Travel\_Approval\_Record)**

\* How to Find Records to Update and Set Their Values

- ☒ Use the travel approval record that triggered the flow
- ☐ Update records related to 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 ← True X

+ Add Field

Cancel Done

Figure 35

6. Create an Action for the Flow Using Update Records Elements.
7. Value = \$GlobalConstant.False

resilient-impala-mwwcfh-dev-ed.trailblaze.lightning.force.com/builder\_platform\_interaction/flowBuilder.app?flowId=3012w000001ICjLAAW

builder Out of State Travel Flag - V1

### Edit Update Records

Update Salesforce records using values from the flow.

**Update Travel Approval Record (Update\_Travel\_Approval\_Recordd)**

**\* How to Find Records to Update and Set Their Values**

- ☒ Use the travel approval record that triggered the flow
- ☐ Update records related to 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

**i** 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 |

+ Add Field

Cancel Done

Figure 36

8. Make sure to save and activate the flow.
9. Flow Label = Out of State Travel Flag
10. How to find Records = Use the travel approval record that triggered the flow
11. Click Save.
12. Click Activate.

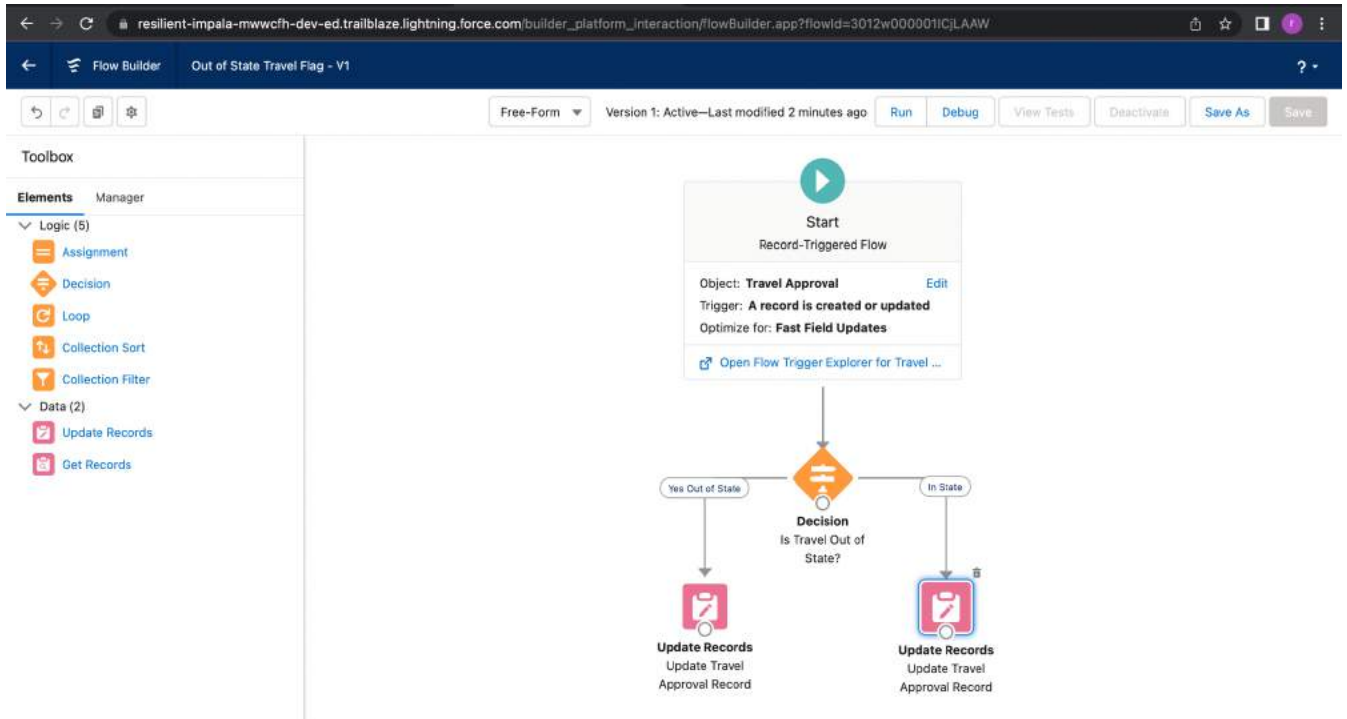


Figure 37

The screenshot shows the Salesforce Travel App interface. The top navigation bar includes the URL `resilient-impala-mwwcfh-dev-ed.trailblaze.lightning.force.com/lightning/r/Travel`. The 'Travel App' is selected in the navigation menu. The record is titled 'Travel Approval TA00002'. The 'Details' tab is active, showing the following fields:

| Field             | Value                               |
|-------------------|-------------------------------------|
| Travel Approval # | TA00002                             |
| Status            | Draft                               |
| Out Of State      | <input checked="" type="checkbox"/> |
| Destination State | TX                                  |

Owner: Ravi Teja Kuchipudi  
Department: Technology

Figure 38

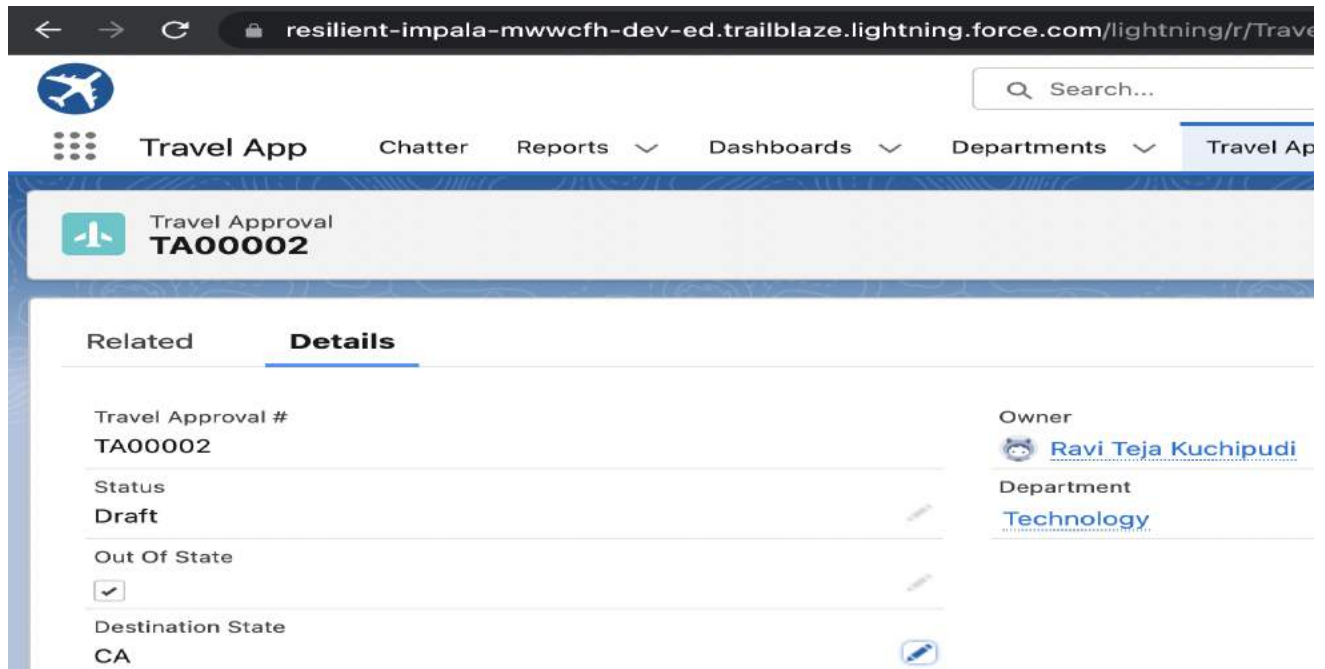


Figure 39

## Step 6:

1. Create an Approval Process to send Travel approvals to the Manager or Travel coordinator. Create an Approval Step for Out-of-State Travel.
2. Create Final Approval action.
3. Create Final Rejection action.

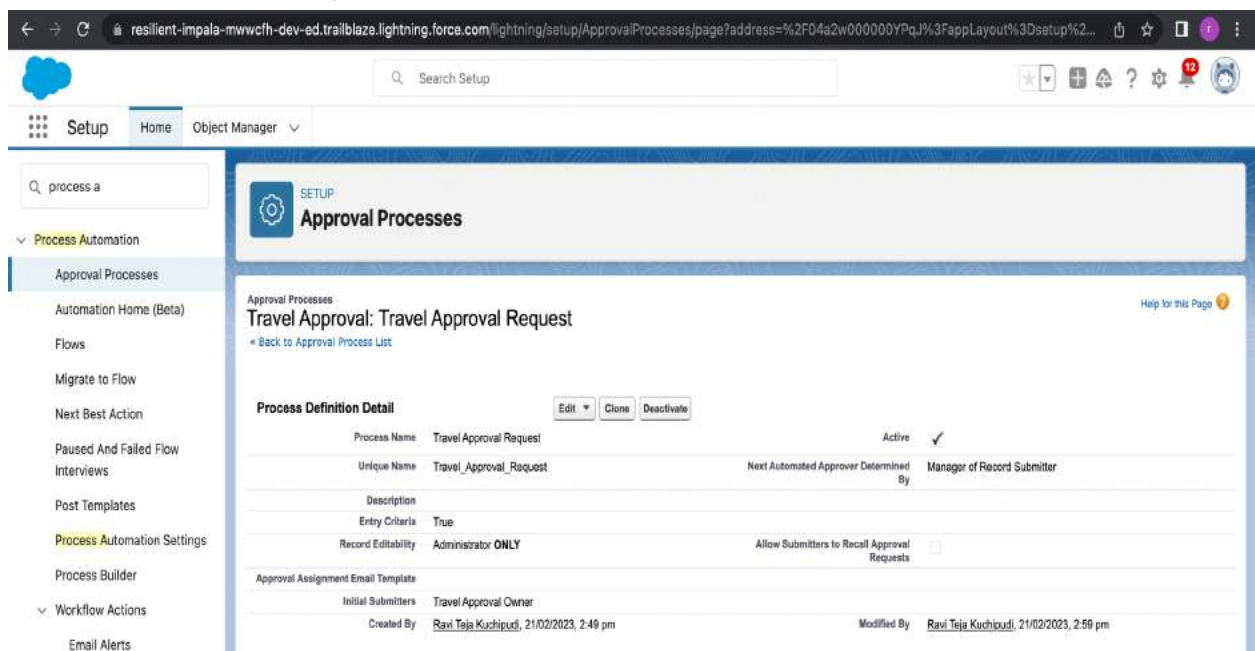
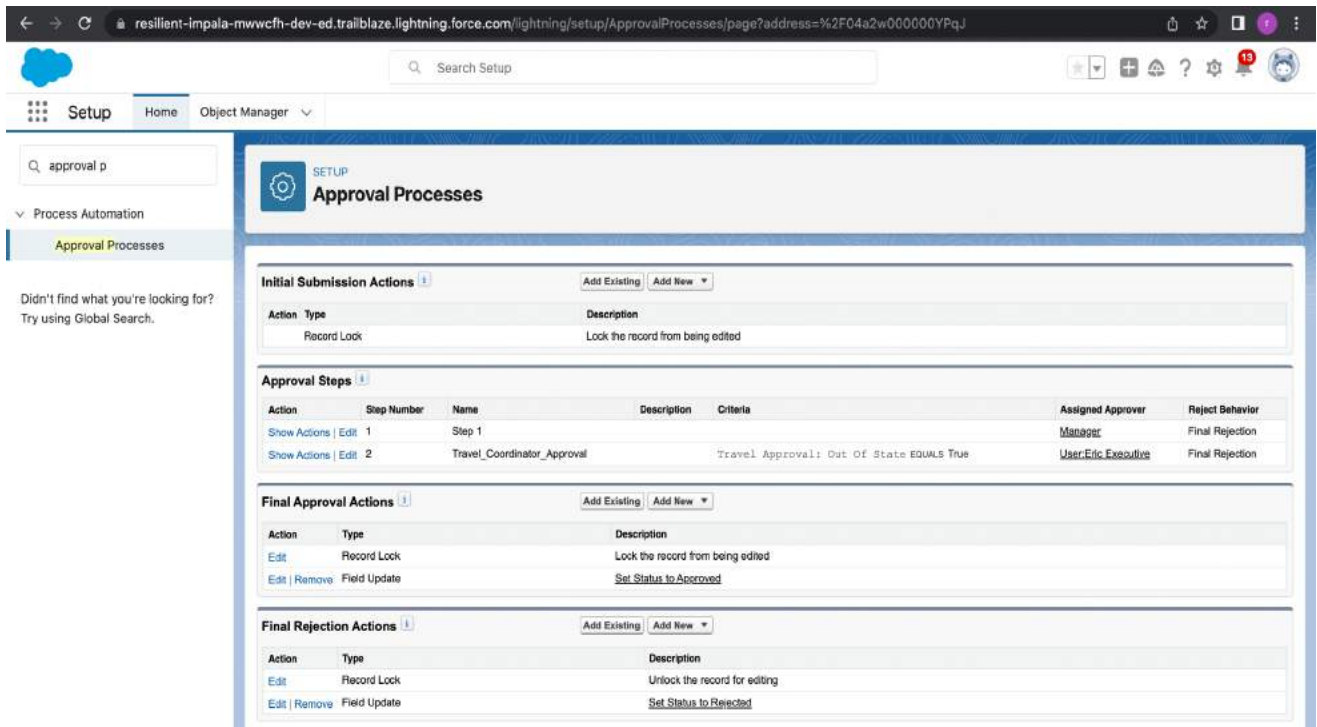


Figure 40





The screenshot shows the Salesforce Setup interface for Approval Processes. The left sidebar contains a search bar with "approval p" and a list of navigation items: Process Automation, Approval Processes, and a message "Didn't find what you're looking for? Try using Global Search." The main content area is titled "Approval Processes" and contains four sections: Initial Submission Actions, Approval Steps, Final Approval Actions, and Final Rejection Actions. Each section has a table of actions and their descriptions.

**Initial Submission Actions**

| Action      | Type | Description                       |
|-------------|------|-----------------------------------|
| Record Lock |      | Lock the record from being edited |

**Approval Steps**

| Action              | Step Number | Name                        | Description | Criteria                                  | Assigned Approver   | Reject Behavior |
|---------------------|-------------|-----------------------------|-------------|---|---------------------|-----------------|
| Show Actions   Edit | 1           | Step 1                      |             |   | Manager             | Final Rejection |
| Show Actions   Edit | 2           | Travel_Coordinator_Approval |             | Travel Approval: Out Of State EQUALS True | User:Eric Executive | Final Rejection |

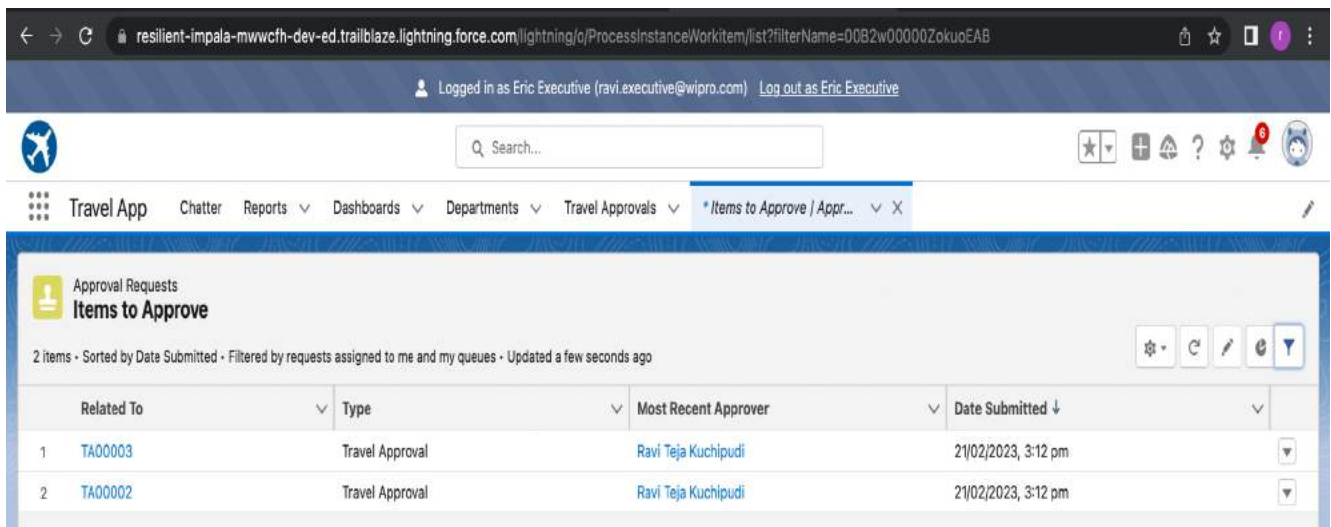
**Final Approval Actions**

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

**Final Rejection Actions**

| Action        | Type         | Description                   |
|---------------|--------------|-------------------------------|
| Edit   Remove | Record Lock  | Unlock the record for editing |
| Edit   Remove | Field Update | Set Status to Rejected        |

Figure 41



The screenshot shows the Salesforce Travel App interface for Items to Approve. The top navigation bar includes "Travel App", "Chatter", "Reports", "Dashboards", "Departments", "Travel Approvals", and a dropdown menu for "Items to Approve / Appr...". The main content area is titled "Approval Requests Items to Approve" and shows a list of 2 items. The list is sorted by Date Submitted and filtered by requests assigned to me and my queues. The table has columns: Related To, Type, Most Recent Approver, and Date Submitted.

**Approval Requests Items to Approve**

2 items • Sorted by Date Submitted • Filtered by requests assigned to me and my queues • Updated a few seconds ago

| Related To | Type            | Most Recent Approver | Date Submitted      |
|------------|-----------------|----------------------|---------------------|
| 1 TA00003  | Travel Approval | Ravi Teja Kuchipudi  | 21/02/2023, 3:12 pm |
| 2 TA00002  | Travel Approval | Ravi Teja Kuchipudi  | 21/02/2023, 3:12 pm |

Figure 42

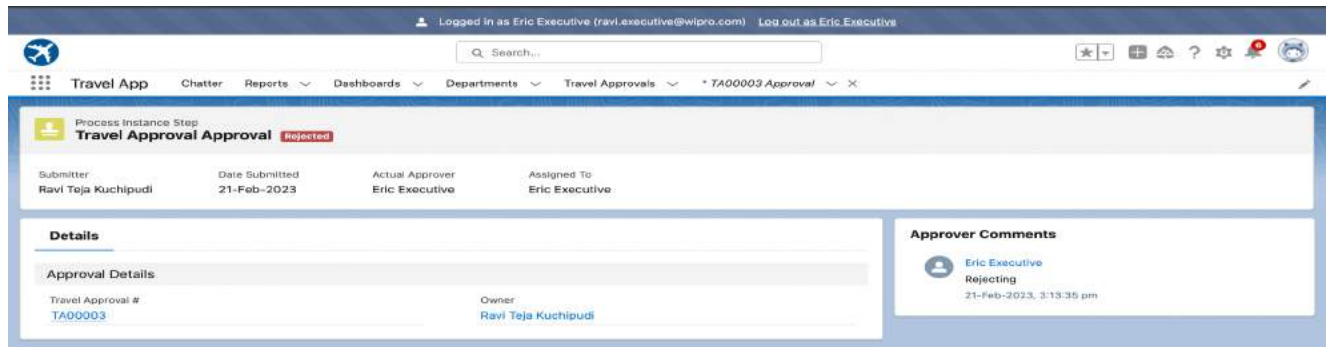


Figure 43

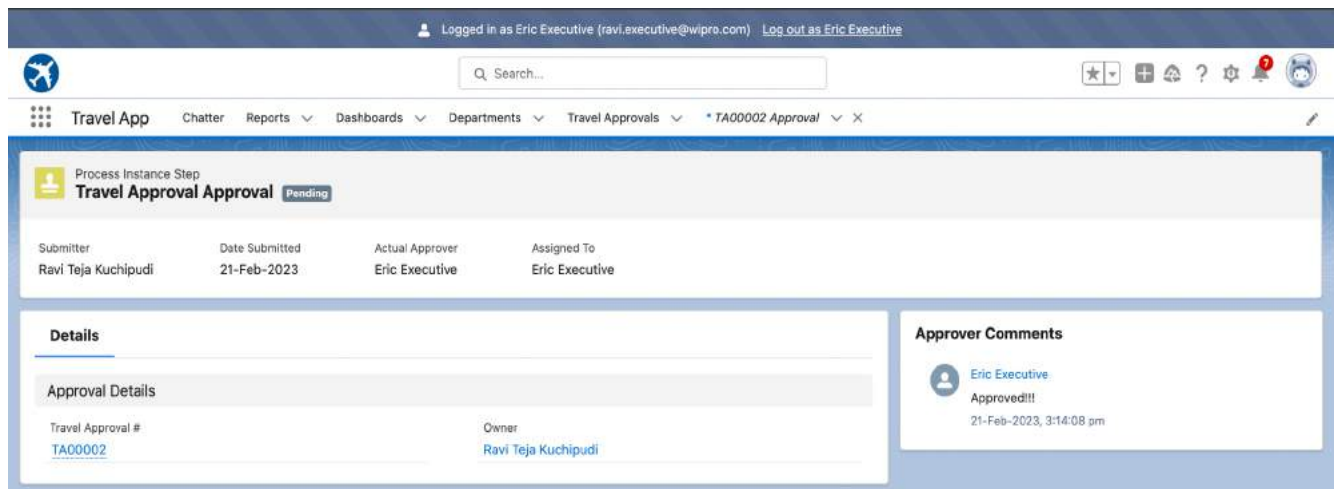


Figure 44

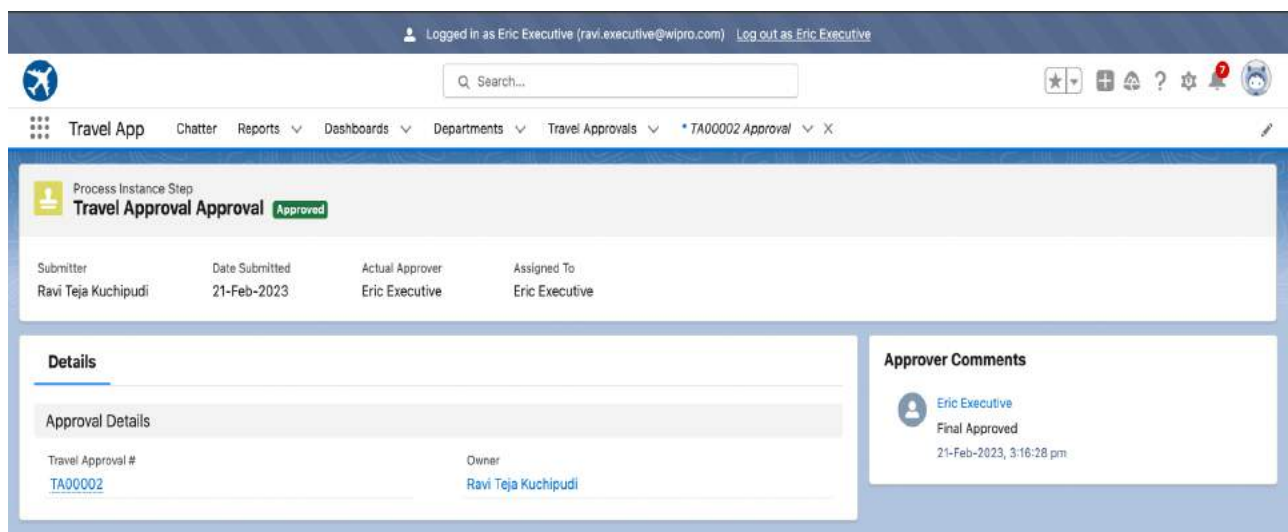
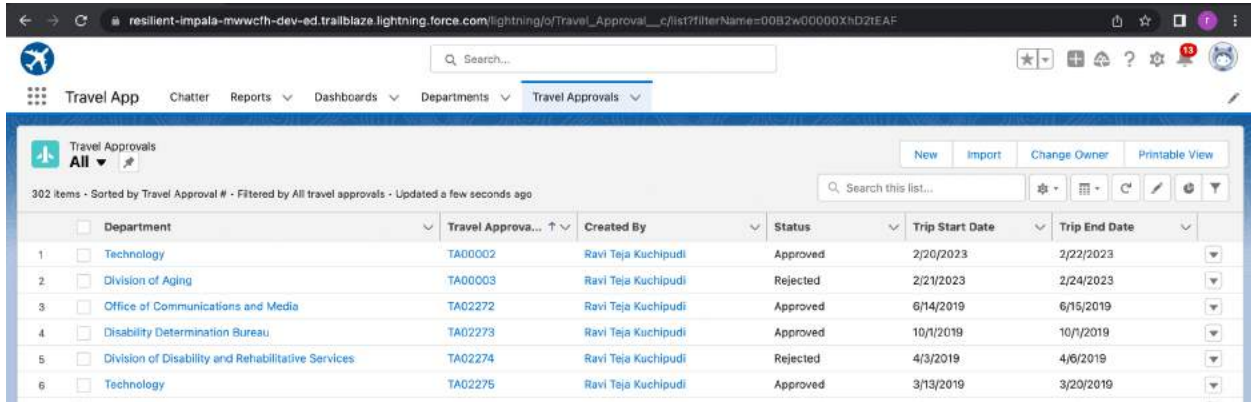


Figure 45



## Exercise 2:

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



|   | Department   | Travel Approval # | Created By          | Status   | Trip Start Date | Trip End Date |
|---|--|-------------------|---------------------|----------|-----------------|---------------|
| 1 | Technology   | TA00002           | Ravi Teja Kuchipudi | Approved | 2/20/2023       | 2/22/2023     |
| 2 | Division of Aging                                  | TA00003           | Ravi Teja Kuchipudi | Rejected | 2/21/2023       | 2/24/2023     |
| 3 | Office of Communications and Media                 | TA02272           | Ravi Teja Kuchipudi | Approved | 6/14/2019       | 6/15/2019     |
| 4 | Disability Determination Bureau                    | TA02273           | Ravi Teja Kuchipudi | Approved | 10/1/2019       | 10/1/2019     |
| 5 | Division of Disability and Rehabilitative Services | TA02274           | Ravi Teja Kuchipudi | Rejected | 4/3/2019        | 4/6/2019      |
| 6 | Technology   | TA02275           | Ravi Teja Kuchipudi | Approved | 3/13/2019       | 3/20/2019     |

Figure 46

**Step 2:** Create a Travel Request by Department Report.

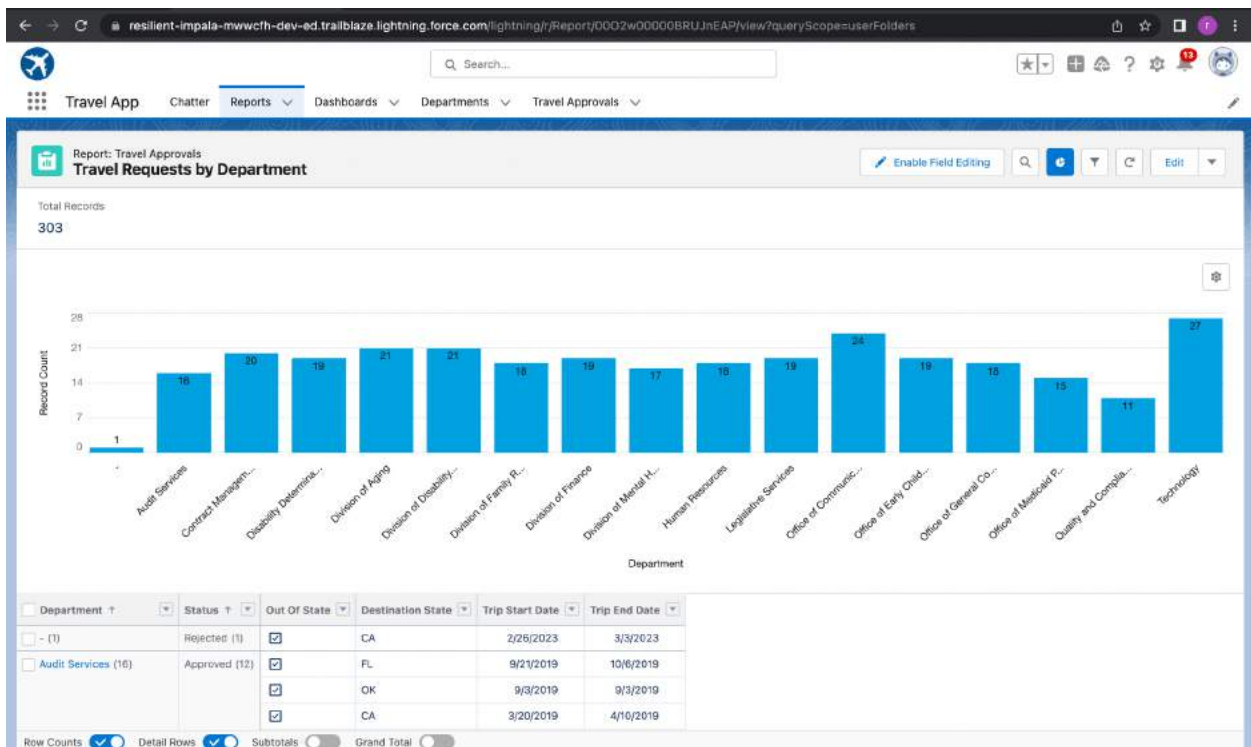


Figure 47

### Step 3: Create a Travel Request by Month Report.

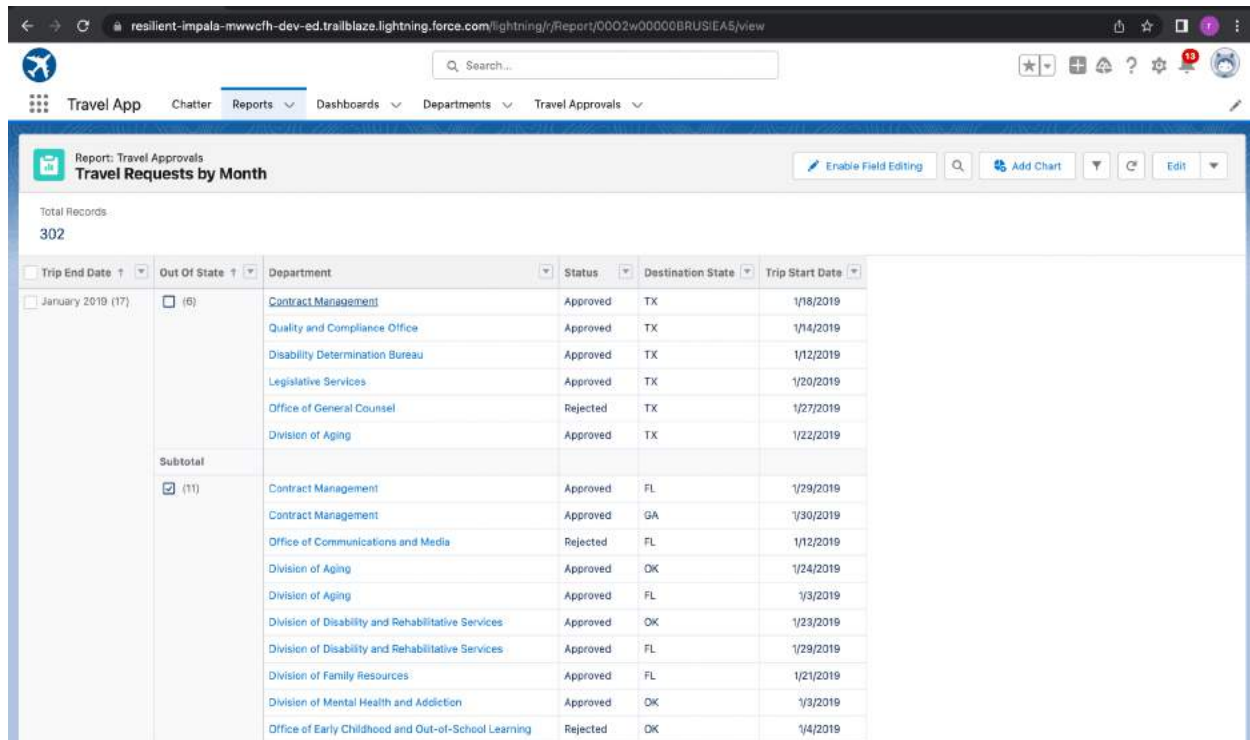


Figure 48

### Step 4: Create a Travel Approvals Dashboard.

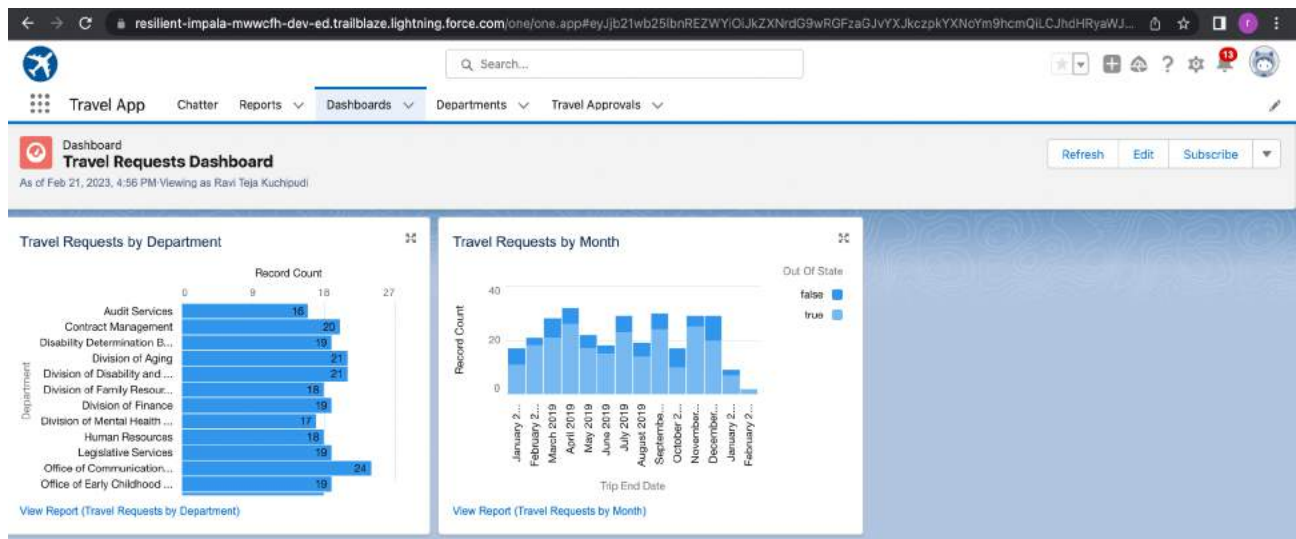


Figure 49

## Module 3

### Business Requirements:

- Demonstrate your Apex programming skills to:
- Write and execute codes from the Anonymous window.
- Demonstrate the use of Data types & Variables.
- Demonstrate the use of Control Flow Statement.
- Define relationships between objects. Data Manipulation using DML.
- Querying the Database using SOQL.
- Create Apex Classes, Objects & Interfaces
- Write Apex Triggers
- Write Test Classes.
- Create VisualForce Pages.

### Exercise 1:

1. Create a new custom lightning App, name: Code Playground.
2. Create a Customer Custom Object
3. Create a Custom fields for Customer Object
  - a. Label = Active, Checkbox, Save.
  - b. Label = Customer Type, Picklist, Values: Premium, Standard.
  - c. Label = Description, Text Area, Save.
  - d. Label = Billing, Master-Detail, Related To – Customer custom object.
4. Create a Billing Custom Object
5. Create Custom fields for Billing Objects.
  - a. Label = Amount Paid, Currency, Save.
  - b. Label = Customer Type, Picklist, Values: Premium, Standard.
  - c. Label = Status, Picklist, Values: Paid, Unpaid.

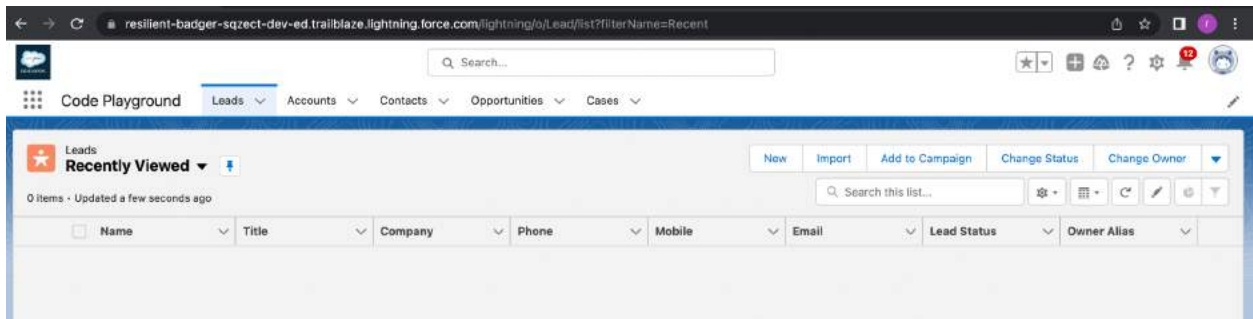


Figure 50

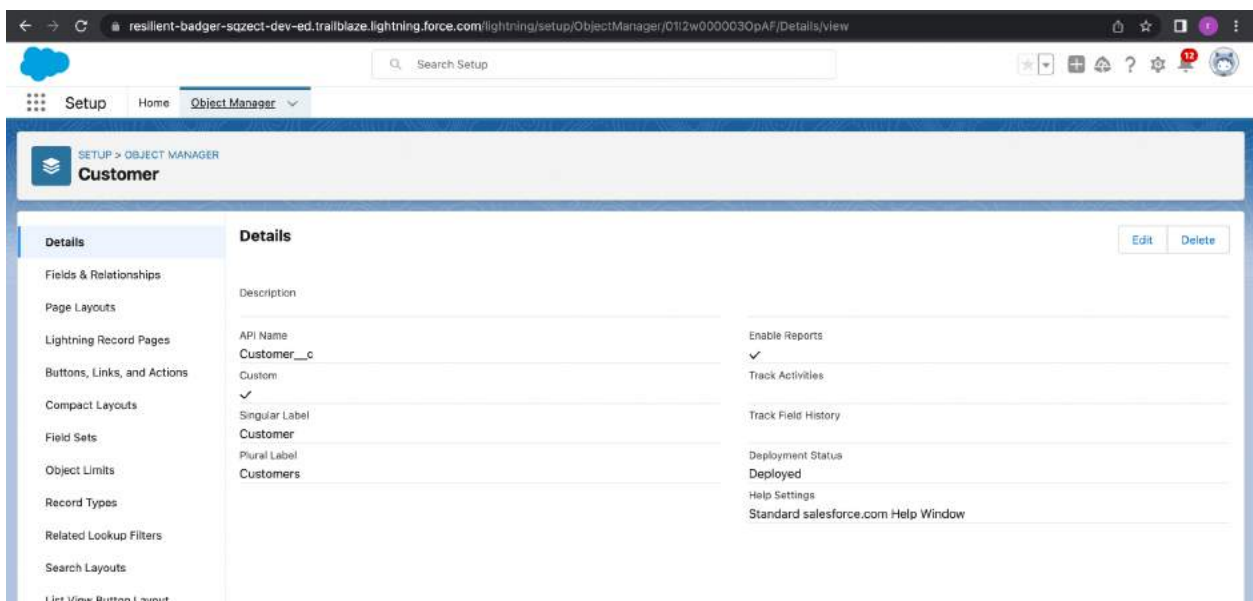


Figure 51

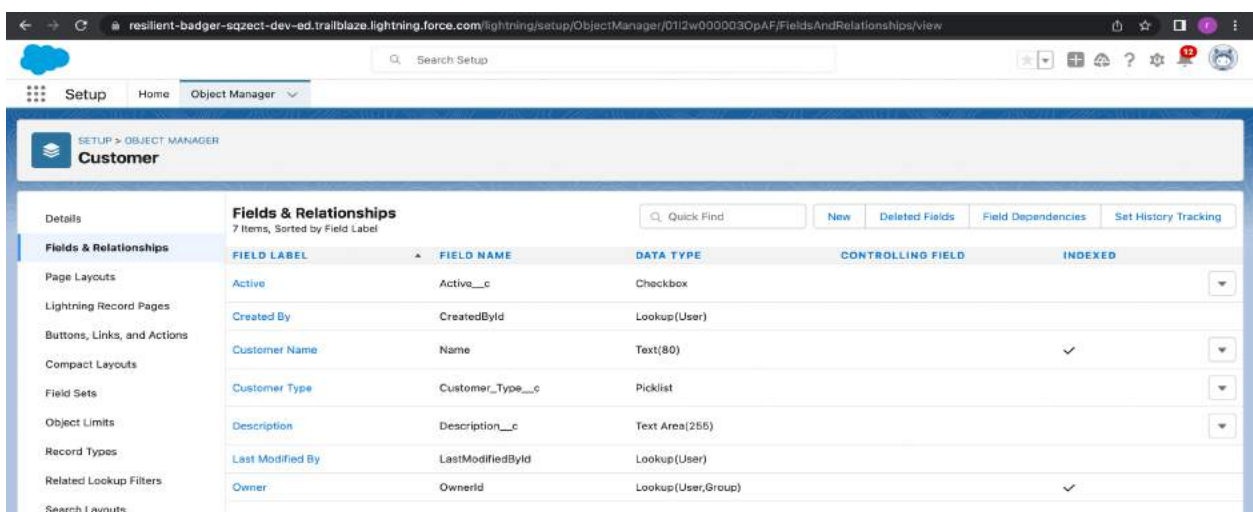


Figure 52

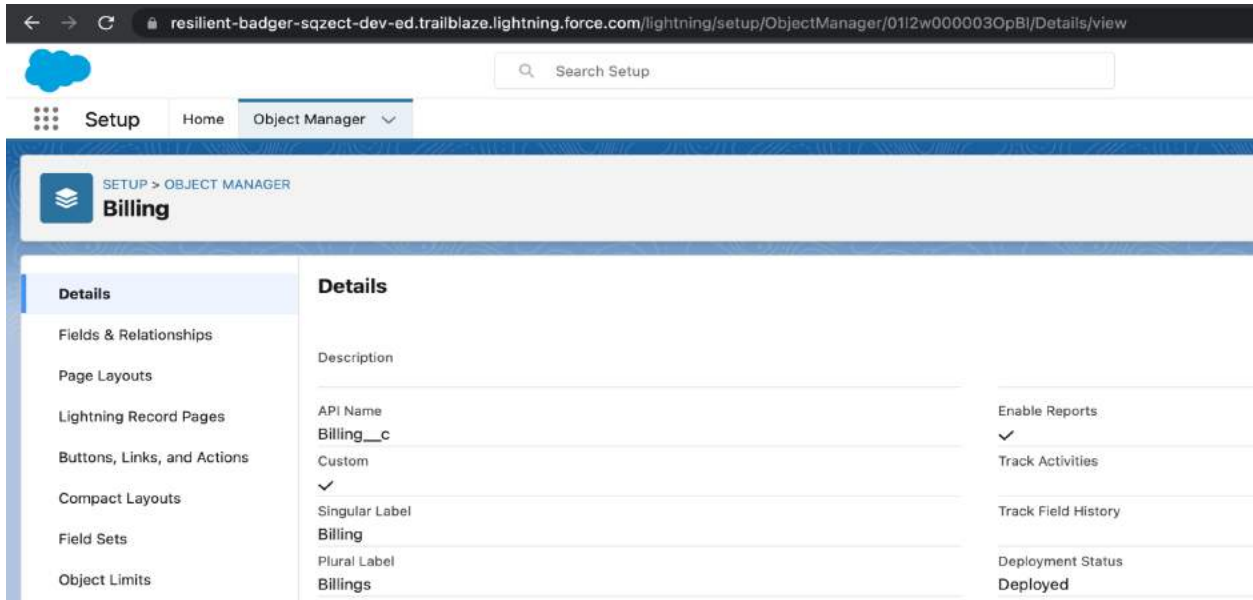


Figure 53

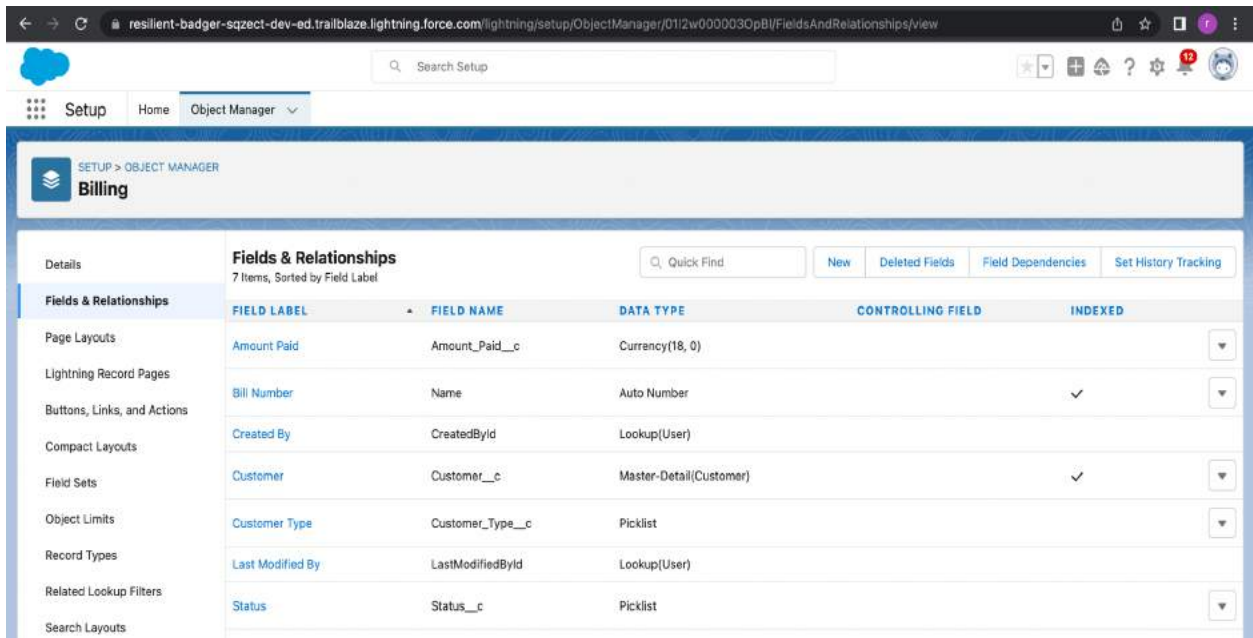


Figure 54

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

Figure 55

**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.

**Enter Apex Code**

```

1 String country = 'India';
2 System.debug(country.endsWith('ia'));
3 System.debug(country.endsWith('ab'));

```

☒ Open Log   Execute   Execute Highlighted

---

**Execution Log**

| Timestamp    | Event      | Details        |
|--------------|------------|----------------|
| 18:38:46:003 | USER_DEBUG | [2]DEBUG:true  |
| 18:38:46:003 | USER_DEBUG | [3]DEBUG:false |

---

☐ This Frame   ☐ Executable   ☒ Debug Only   ☐ Filter   [Click here to filter the log](#)

| User                | Application | Operation                              | Time                 | Status  |
|---------------------|-------------|--|----------------------|---------|
| Ravi Teja Kuchipudi | Unknown     | /services/data/v57.0/tooling/execut... | 21/02/2023, 18:38:46 | Success |
| Ravi Teja Kuchipudi | Unknown     | /services/data/v57.0/tooling/execut... | 21/02/2023, 18:38:10 | Success |

Figure 56



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

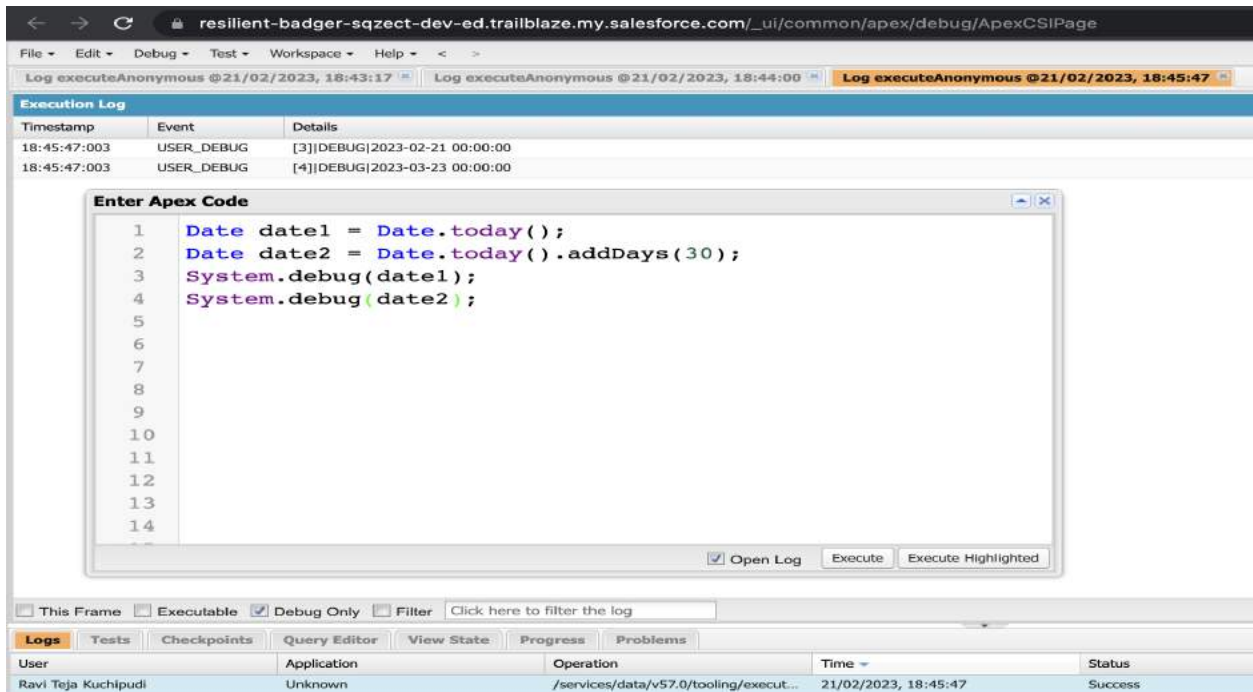


Figure 57

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

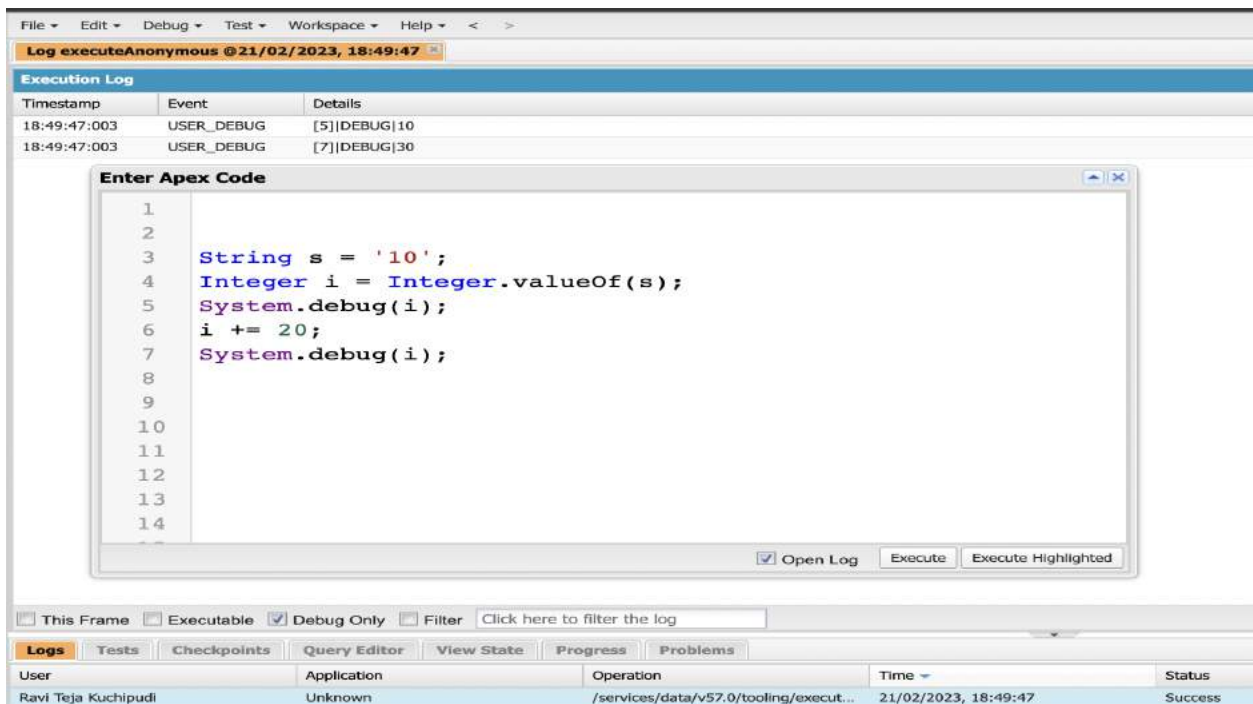


Figure 58

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

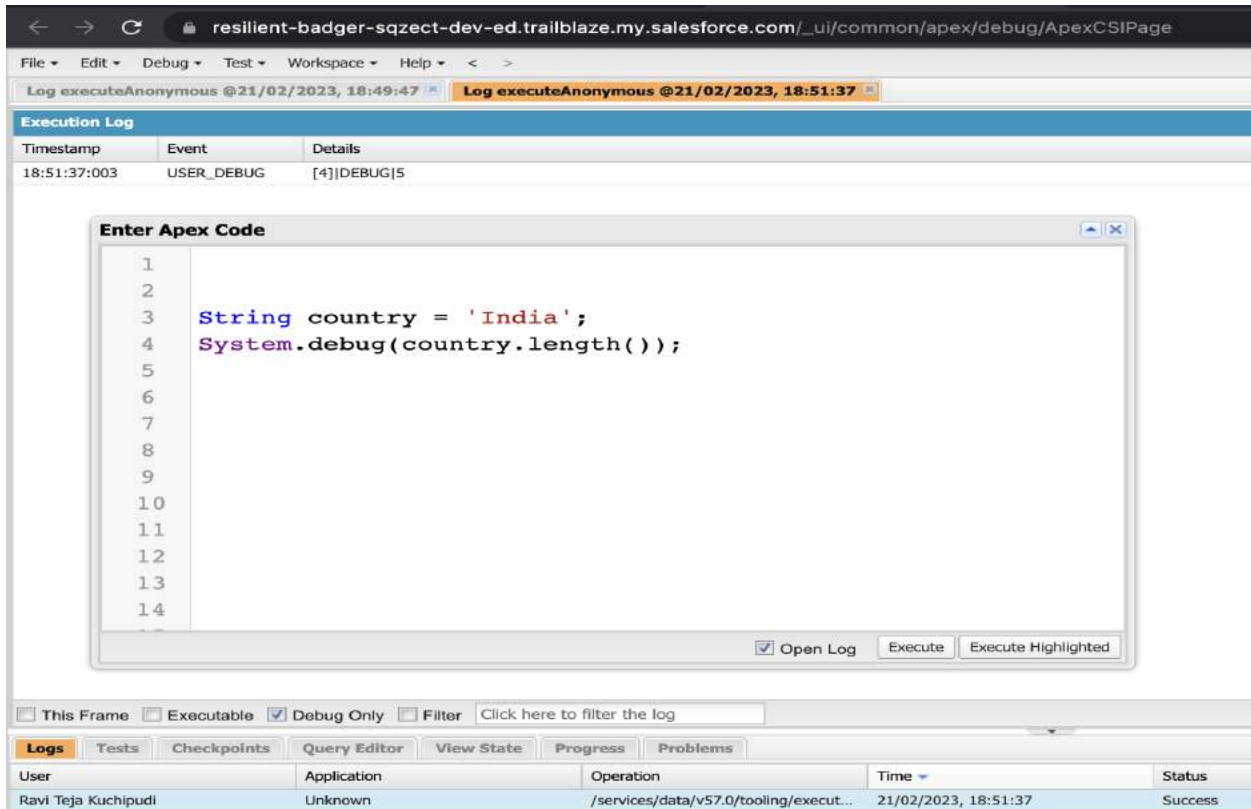


Figure 59

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



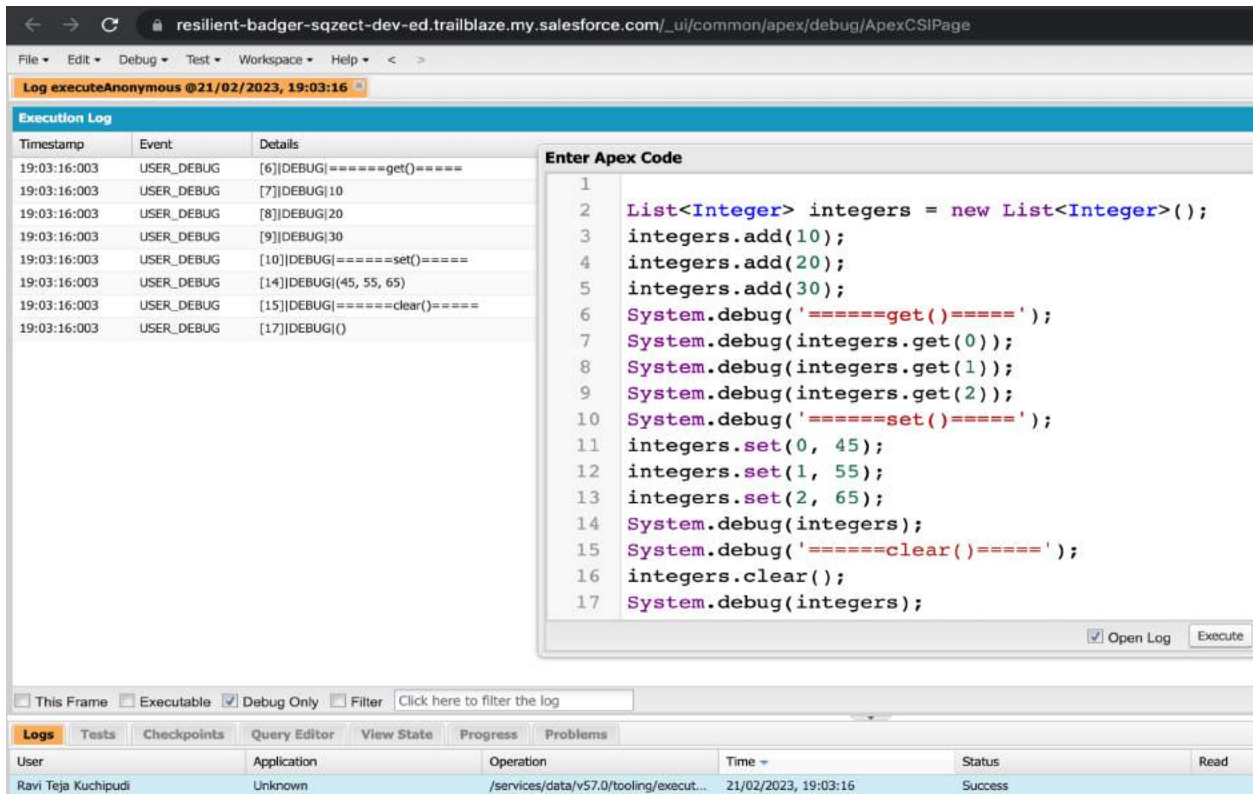


Figure 60

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

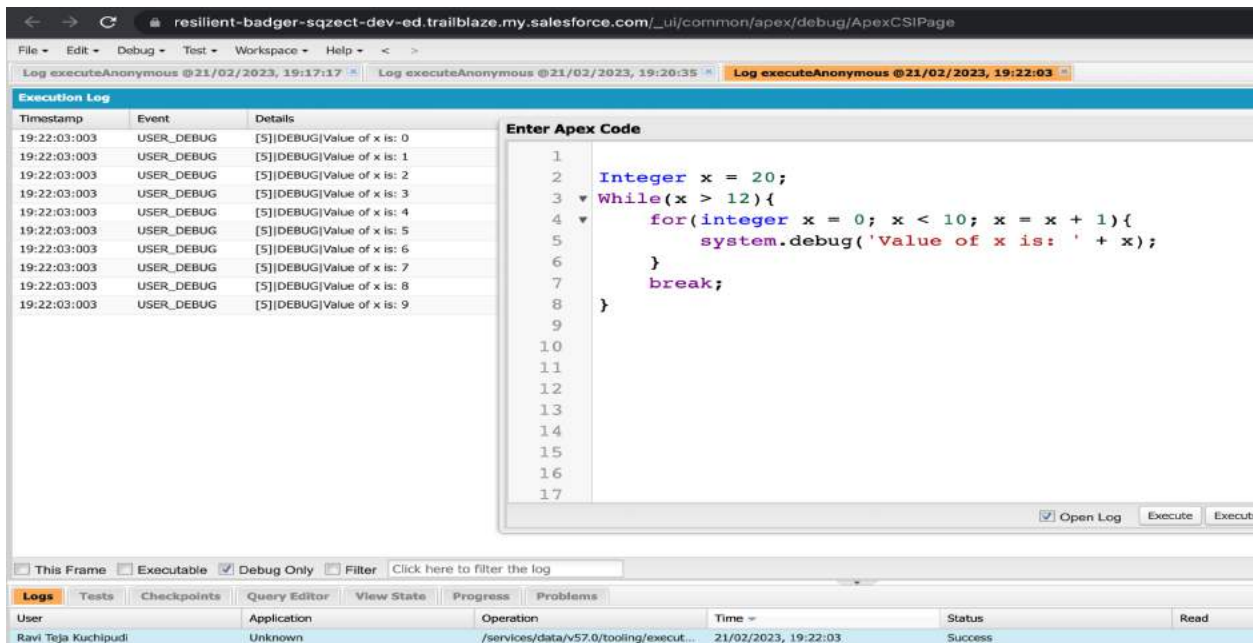


Figure 61

**Exercise 3:** Answer the following in True Or False: **(False)**

1. Integer myluckyNumber = 15; Integer myunlucklyNumber = 7;
2. myluckyNumber != myunlucklyNumber + 8.

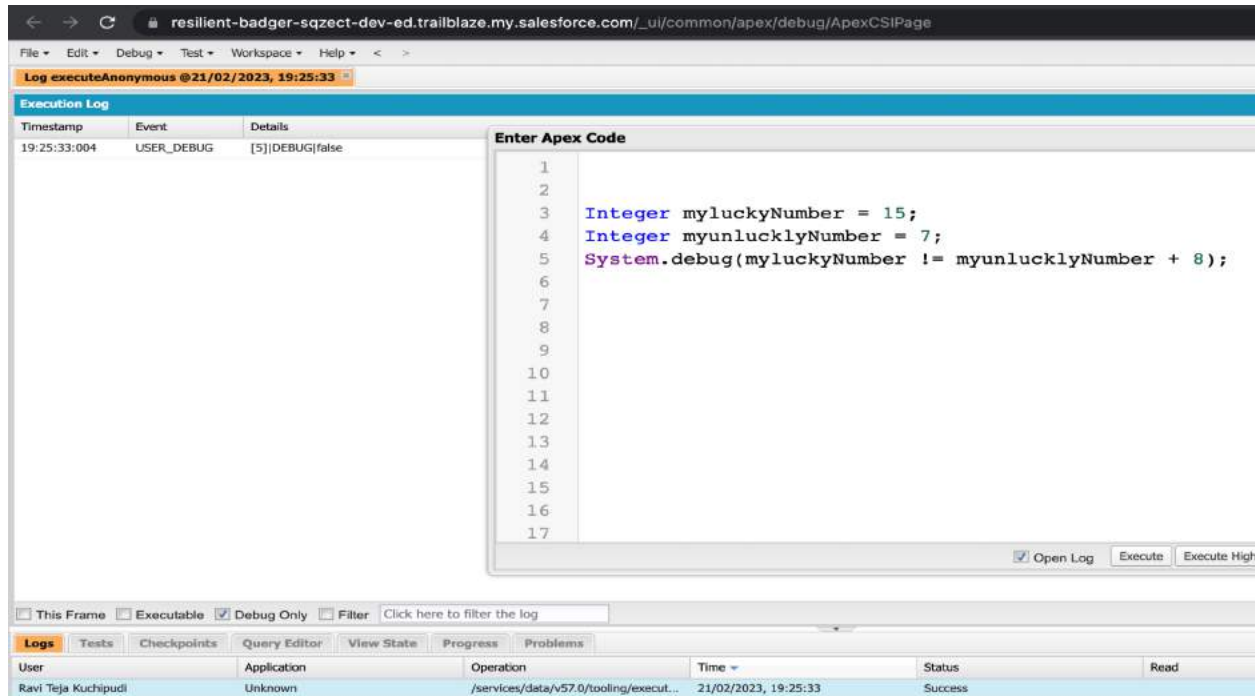


Figure 62

**Exercise 4:** Answer the following in True Or False: **(True)**

1. Boolean isTrue = True; Boolean isFalse = false; isTrue || isFalse

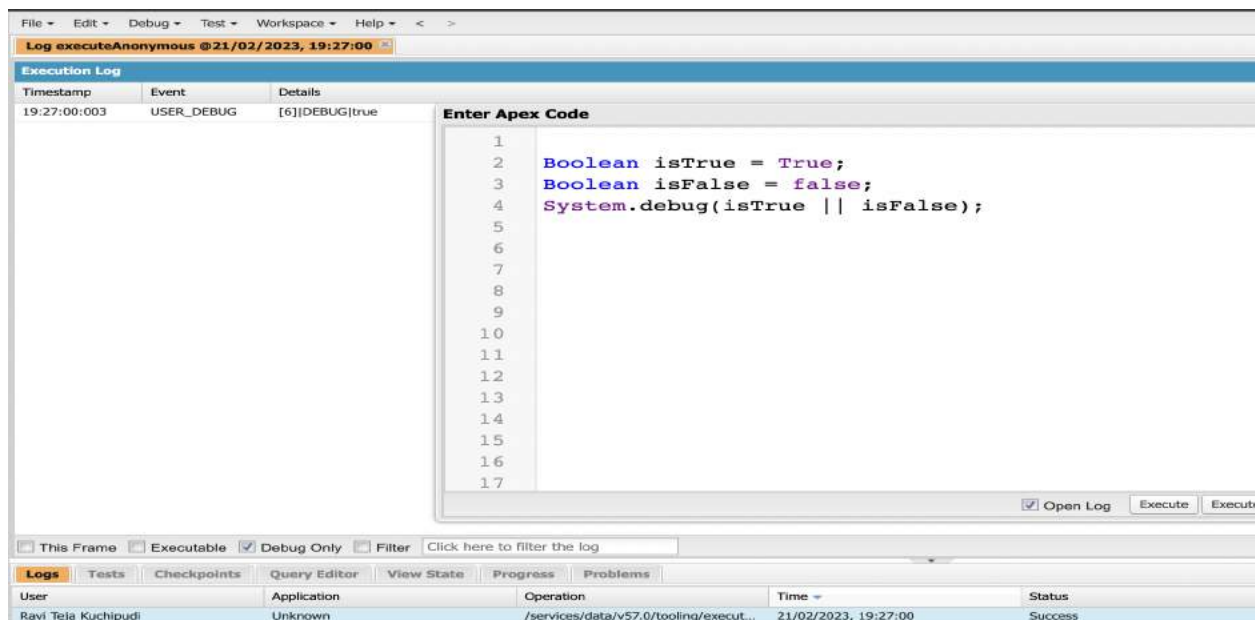


Figure 63

**Exercise 5:** Answer the following in True Or False: **(True)**

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

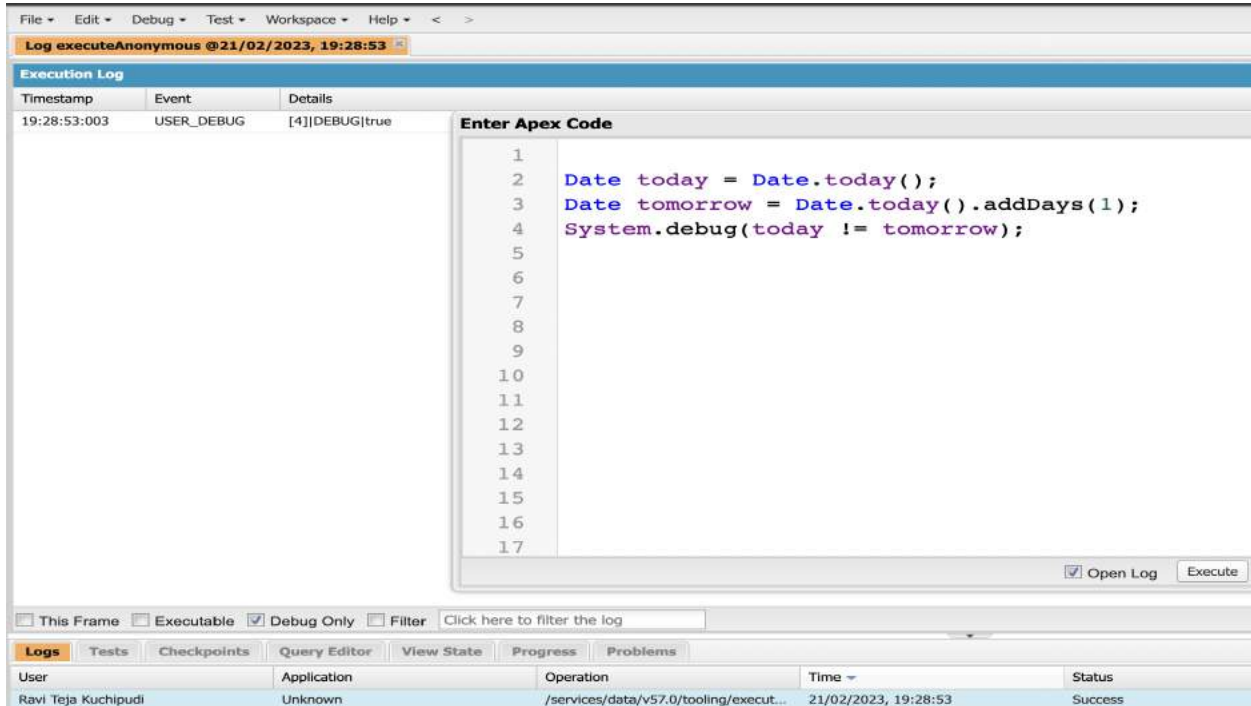


Figure 64

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

1. Define an Integer variable called Score.
2. System.debug (‘Grade: A+’), if Score is equal to 100.
3. System.debug (‘Grade: A’), if Score is greater than or equals to 90.
4. System.debug (‘Grade: B’), if Score is greater than or equals to 80.
5. System.debug (‘Grade: Failed’), define as final else.

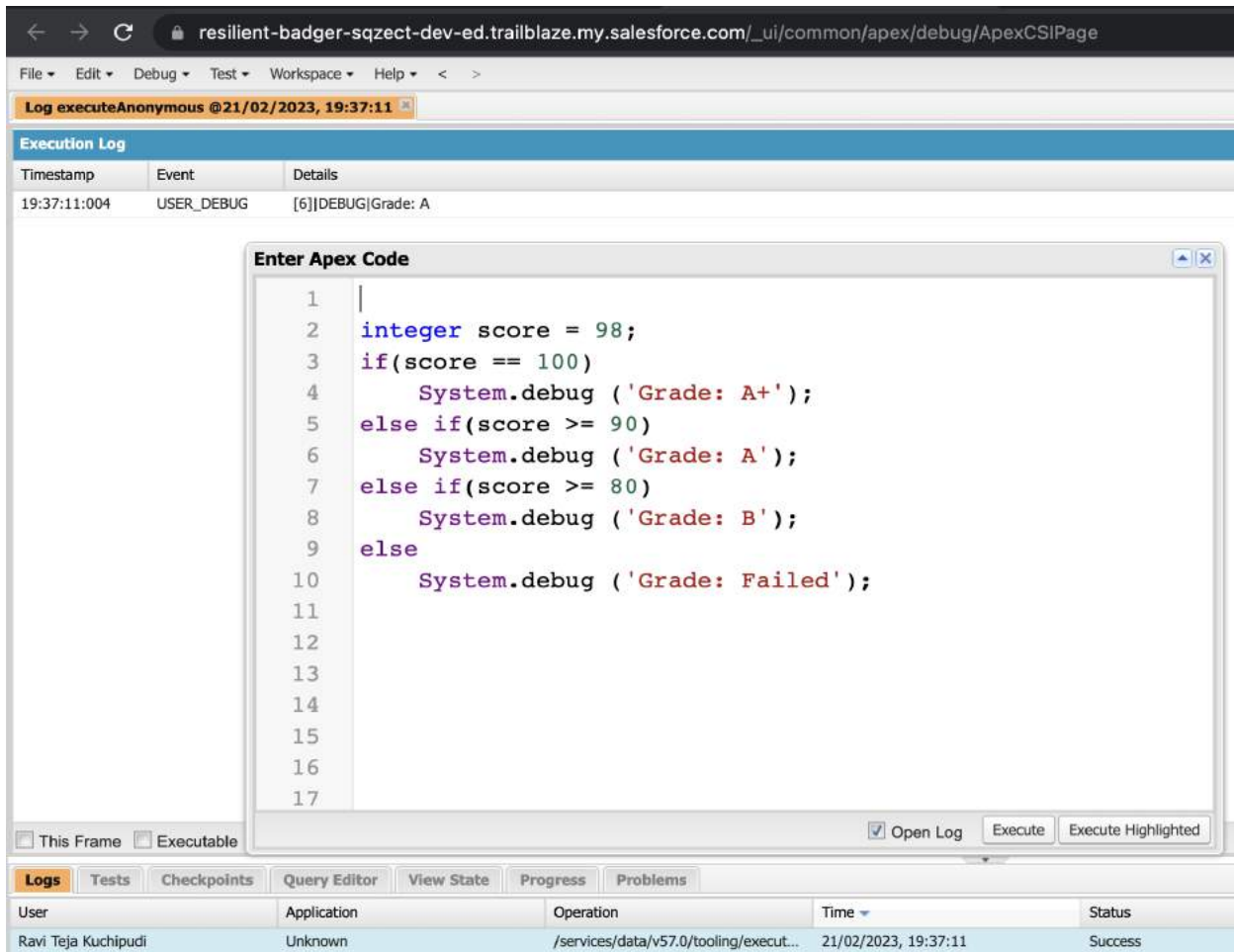


Figure 65

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

1. Initialize the Billing object list to store the Billing Records created today, use SOQL.
2. Use for Loop to iterate over the list and check if the Billings status = Paid.
3. `System.debug ('Value of Current Record in the Loop'+ Billing List);`
4. if Status is paid then add the Billings into a List of String.
5. `System.debug('Value of BillingList '+BillingList);`
6. Note: Create at least 2 Billing records with Status = Paid for this exercise.

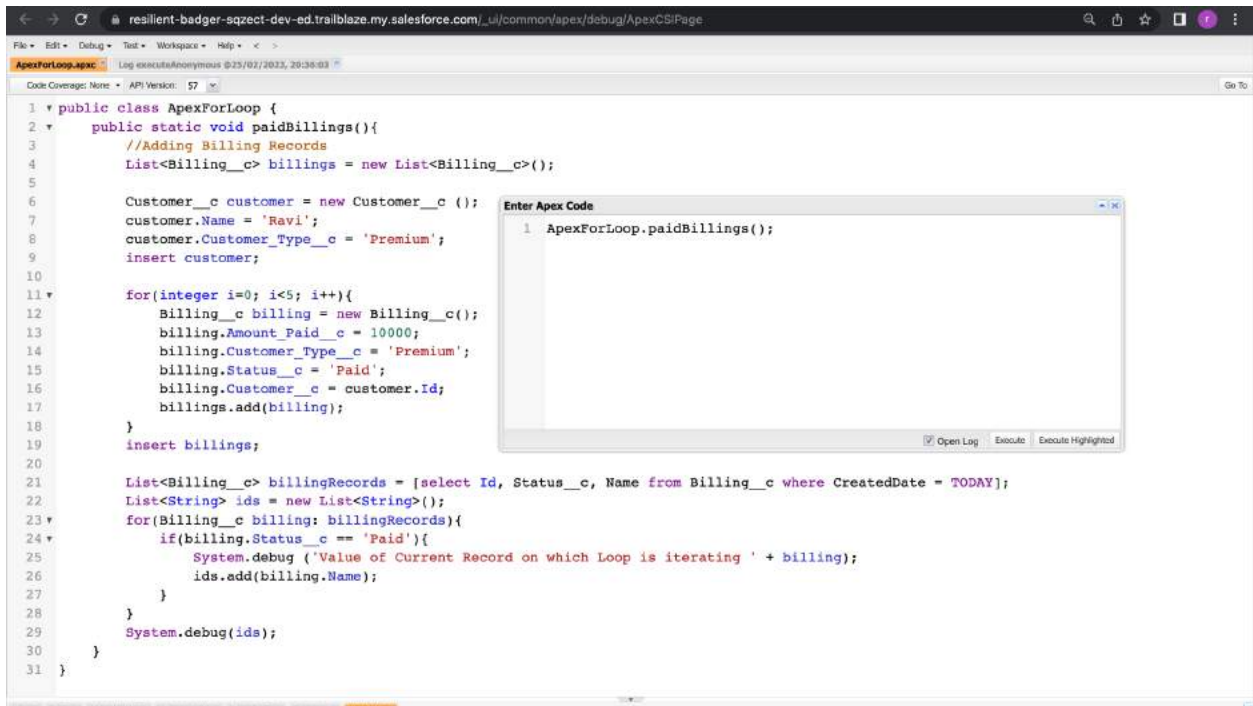


Figure 66

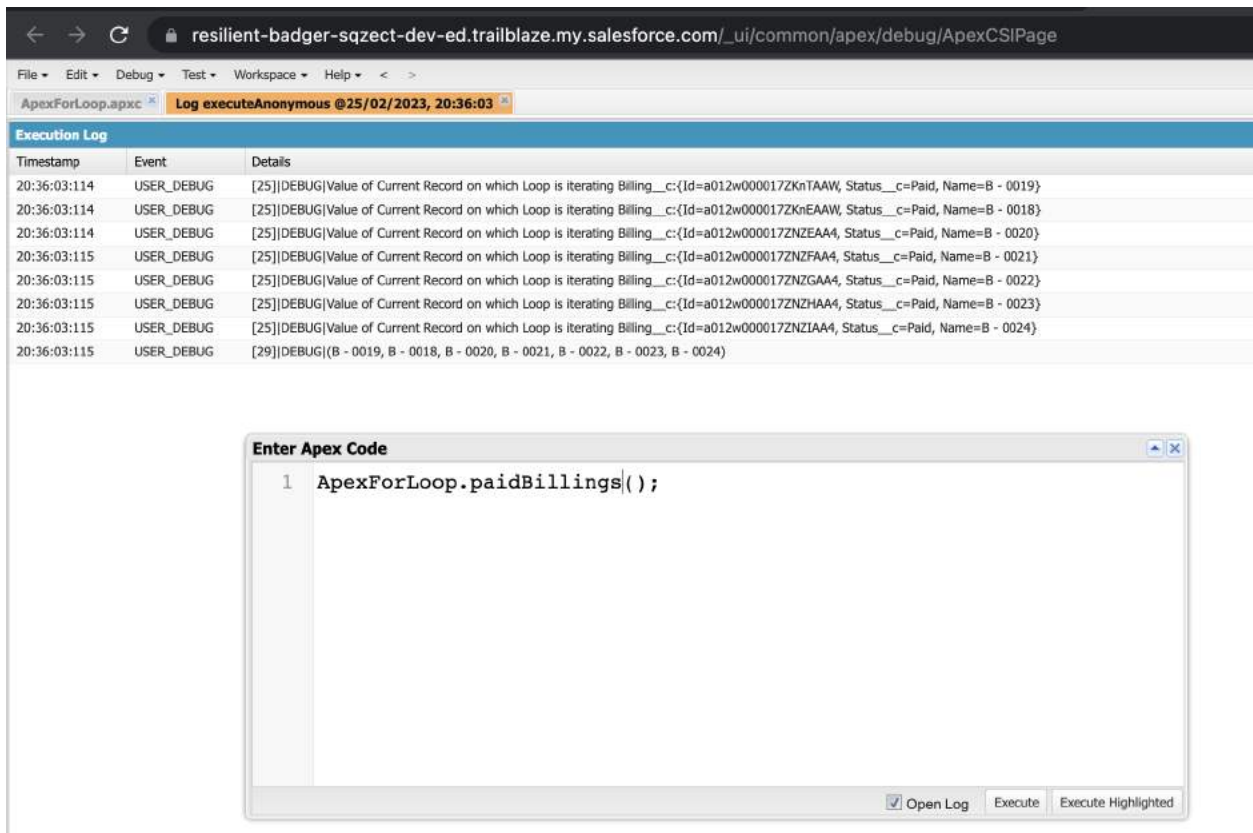


Figure 67

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

1. Class Name = DiscountClass.
2. Define a Final Decimal variable called “regularDiscount”.
3. Define a Decimal variable called “finalPrice”.
4. Define a Method called “calculateDiscount(Integer price)”, takes a Decimal data type.
5. finalPrice = price - price\*regularDiscount;
6. Don’t forget to return a Decimal data type.

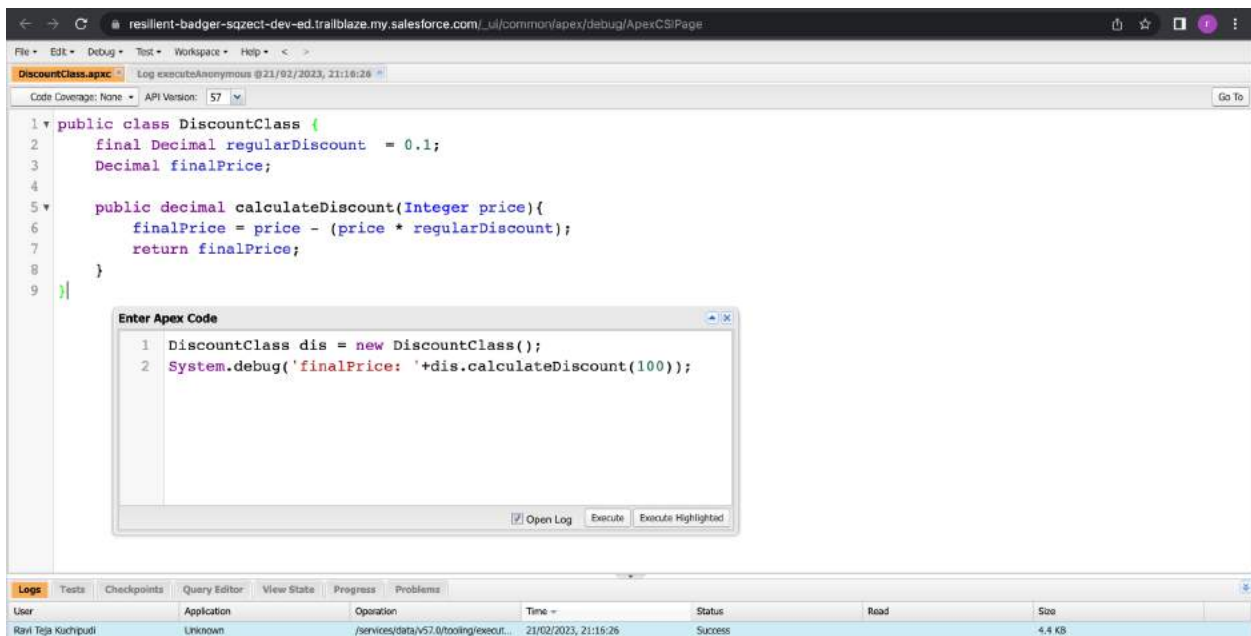


Figure 68

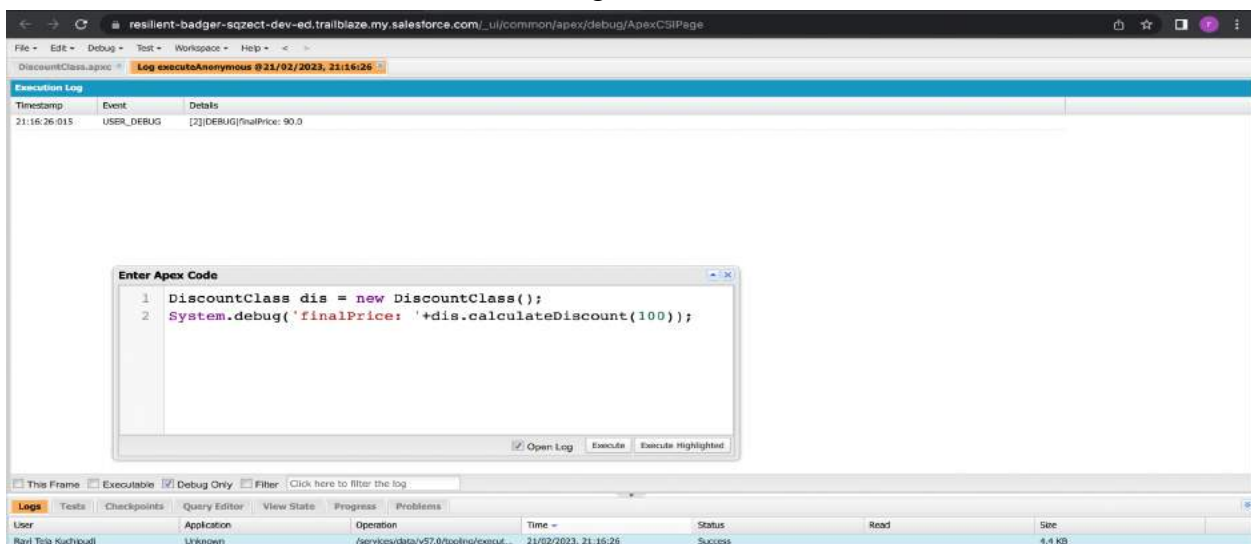


Figure 69



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

1. Interface Class Name = InterfaceExample. Method Signature = Double percentageDiscountTobeApplied();
2. Create an Apex Class to implement the Interface Class, Name = PremiumCustomer . Call the interface method to return 30% Discount.
3. Create another Apex Class to implement the Interface Class, Name = normalCustomer Call the interface method to return 10% Discount.

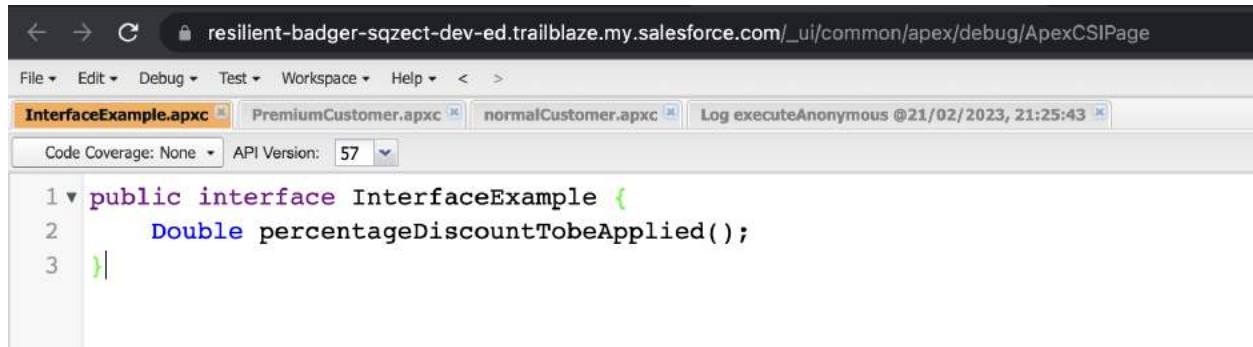


Figure 70

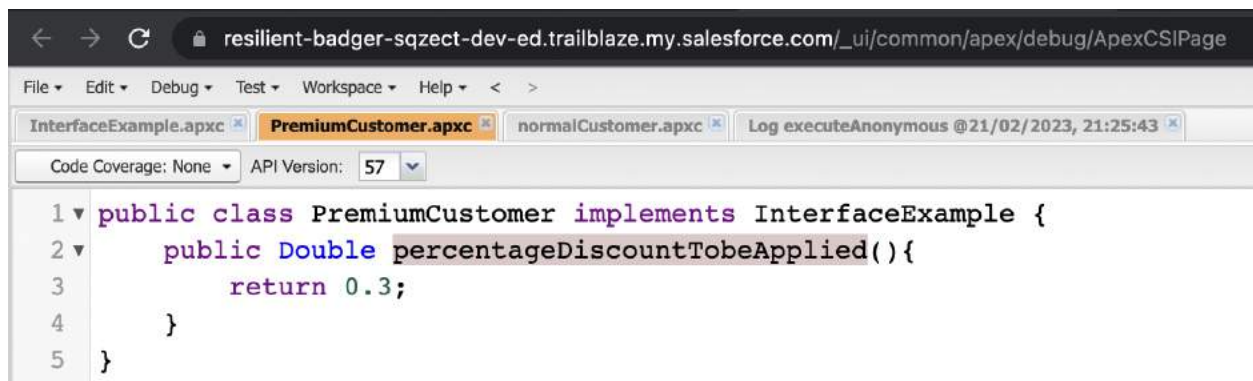


Figure 71

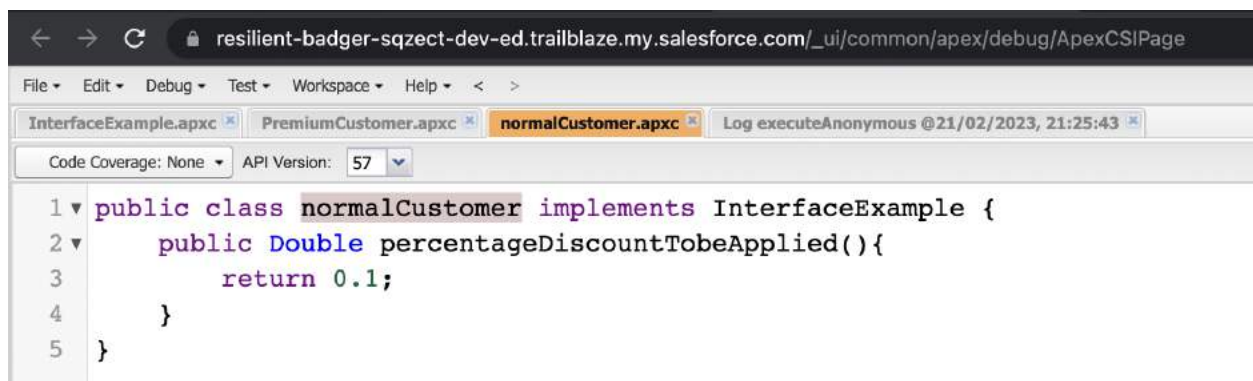


Figure 72



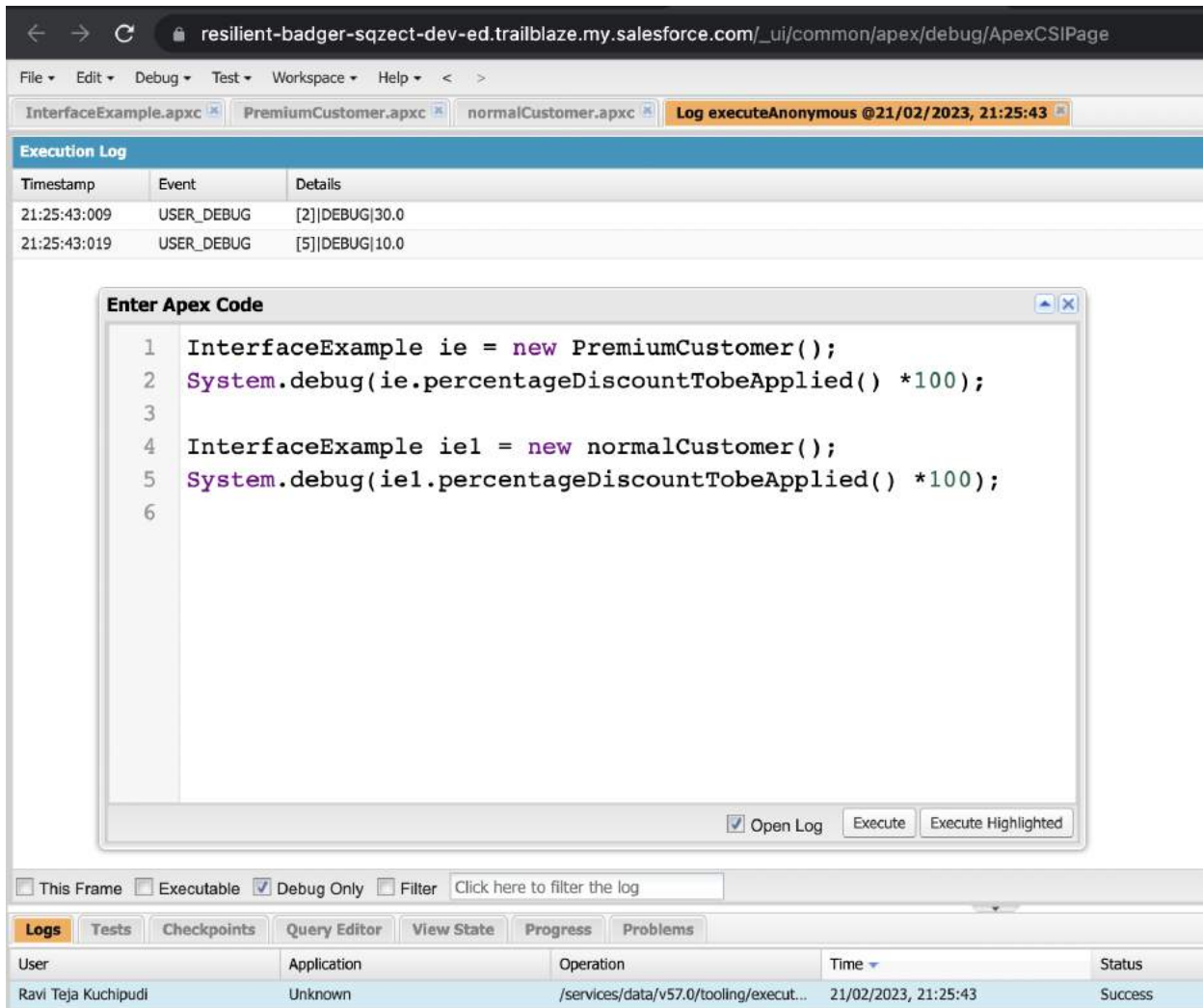


Figure 73

### 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.

The screenshot displays the Salesforce Apex IDE interface. The main editor shows the following Apex code:

```

1 public class DatabaseOperations {
2     public static void DBOps(){
3         //Insert a Customer Record First using simple DML Statement
4         Customer__c customer = new Customer__c ();
5         customer.Name = 'Wipro';
6         customer.Customer_Type__c = 'Premium';
7         insert customer;
8
9         Billing__c billing = new Billing__c ();
10        billing.Status__c = 'Paid';
11        billing.Amount_Paid__c = 5000000;
12        billing.Customer__c = customer.Id;
13
14        List<Billing__c> billings = new List<Billing__c>();
15        billings.add(billing);
16
17        //Insert Billing record using Database methods.
18        Database.SaveResult[] srList = Database.insert(billings,false) ;
19
20        for(Database.SaveResult sr : srList ){
21            if(sr.isSuccess())
22                System.debug(sr);
23            else{
24                Database.Error[] err = sr.getErrors();
25                for(Database.Error e: err){
26                    System.debug(e.getMessage());

```

An "Enter Apex Code" dialog box is open, showing the first line of code: `DatabaseOperations.DBOps();`. Below the code editor, the "Logs" tab is active, displaying a successful execution log:

| User                | Application | Operation                              | Time                 | Status  | Read |
|---------------------|-------------|--|----------------------|---------|------|
| Ravi Teja Kuchipudi | Unknown     | /services/data/v57.0/tooling/execut... | 22/02/2023, 07:39:41 | Success |      |

The bottom part of the image shows the continuation of the code from the previous block, including closing braces and additional debug statements:

```

27        System.debug(e.getStatusCode());
28        System.debug(e.getFields());
29
30    }
31 }
32 }
33 }
34 }

```

The "Logs" tab at the bottom also shows a successful execution log for the same user and operation.

Figure 74

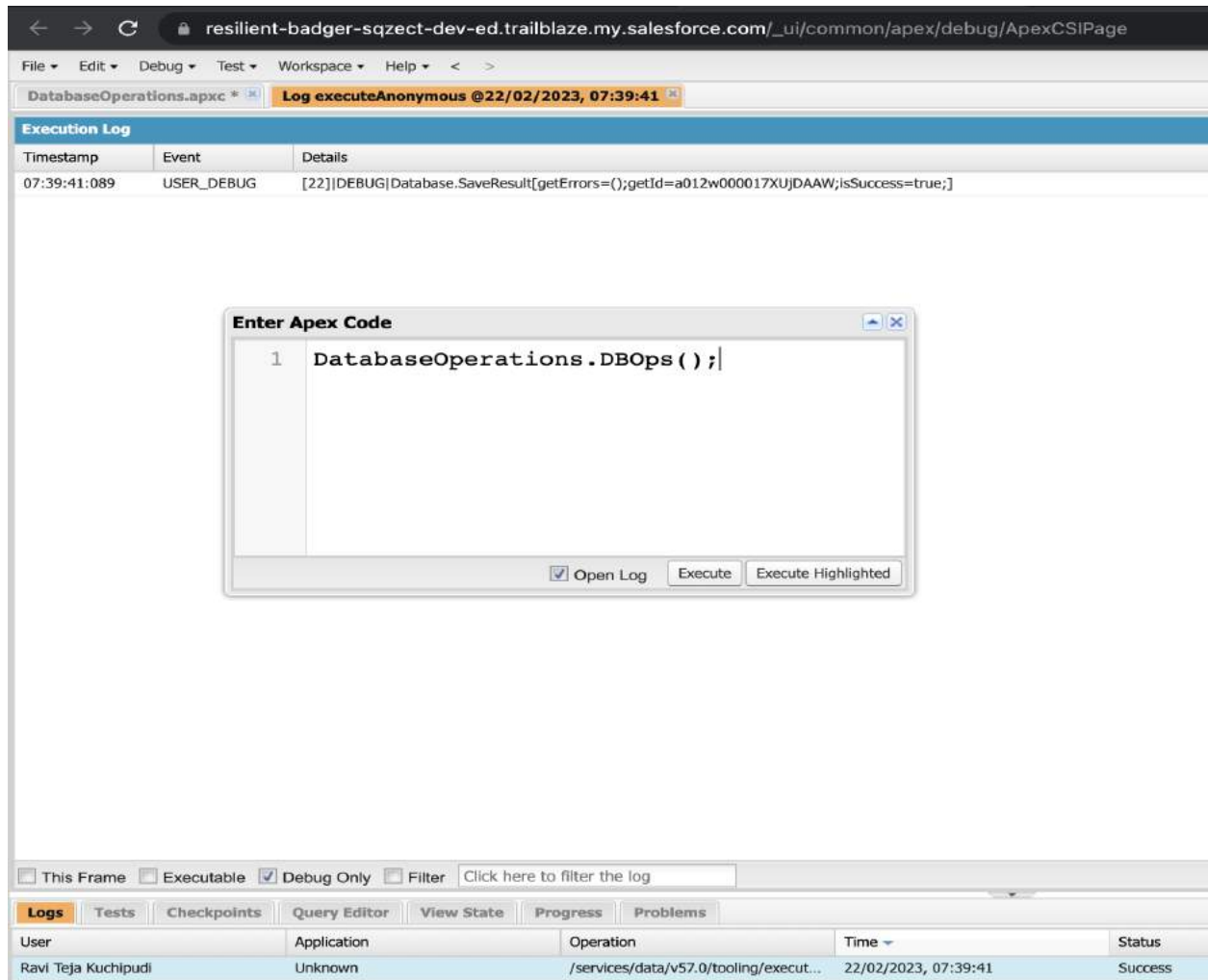


Figure 75

### 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”

The screenshot shows a Salesforce Apex CS page with the following query: `SELECT Id, Amount, StageName, AccountId, Account.Industry, Account.Website FROM Opportunity where Account.Industry = 'Energy' and Account.AnnualRevenue > 5000`. The results show 10 rows of data.

| Id                 | Amount | StageName            | AccountId          | Account.Industry | Account.Website    |
|--------------------|--------|----------------------|--------------------|------------------|--------------------|
| 0062w00000K43NnAAJ | 125000 | Negotiation/Review   | 0012w00001LdWjWAAV | Energy           | http://www.ucs.com |
| 0062w00000K43NnAAJ | 270000 | Proposal/Price Quote | 0012w00001LdWjWAAV | Energy           | http://www.ucs.com |
| 0062w00000K43NnAAJ | 120000 | Closed Won           | 0012w00001LdWjWAAV | Energy           | http://www.ucs.com |
| 0062w00000K43D0AAJ | 270000 | Negotiation/Review   | 0012w00001LdWjWAAV | Energy           | http://www.ucs.com |
| 0062w00000K43D3AAJ | 270000 | Closed Won           | 0012w00001LdWjWAAV | Energy           | http://www.ucs.com |
| 0062w00000K43D5AAJ | 915000 | Closed Won           | 0012w00001LdWjWAAV | Energy           | http://www.ucs.com |
| 0062w00000K43DAAZ  | 235000 | Closed Won           | 0012w00001LdWjWAAV | Energy           | http://www.ucs.com |
| 0062w00000K43D8AAZ | 440000 | Closed Won           | 0012w00001LdWjWAAV | Energy           | http://www.ucs.com |
| 0062w00000K43DAAZ  | 120000 | Closed Won           | 0012w00001LdWjWAAV | Energy           | http://www.ucs.com |
| 0062w00000K43DFAAZ | 675000 | Needs Analysis       | 0012w00001LdWjWAAV | Energy           | http://www.ucs.com |

Figure 76

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

1. Create a Billing Record when on Customer Object the 'Status' field changes to Active from Inactive.
2. Billing “Status” must be: Paid, Amount Paid = 1000000.
3. Perform the DML on a List.

```

1 trigger CustomerTrigger on Customer__c (after update) {
2     if (Trigger.isAfter && Trigger.isUpdate) {
3         List<Billing__c> billings = new List<Billing__c>();
4         for (Customer__c newC : Trigger.New) {
5             if (Trigger.OldMap.get(newC.Id).Active__c == false && newC.Active__c == true) {
6                 Billing__c billing = new Billing__c ();
7                 billing.Amount_Paid__c = 1000000 ;
8                 billing.Status__c = 'Paid';
9                 billing.Customer__c = newC.Id;
10                billings.add(billing);
11            }
12        }
13        if (billings.size() > 0) {
14            insert billings;
15        }
16    }
17 }

```

Figure 77

resilient-badger-sqzect-dev-ed.trailblaze.lightning.force.com/lightning/r/Customer\_\_c/a002w00000ZE1o3AAD/view

Code Playground Leads Accounts Contacts Opportunities Cases Billings Customers

Customer Ravi

Related Details

|               |  |                  |  |
|---------------|--|------------------|--|
| Customer Name | Ravi                                     | Owner            | Ravi Teja Kuchipudi                      |
| Active        | <input type="checkbox"/>                 |                  |  |
| Customer Type | Premium                                  |                  |  |
| Description   |  |                  |  |
| Created By    | Ravi Teja Kuchipudi, 22/02/2023, 7:25 am | Last Modified By | Ravi Teja Kuchipudi, 25/02/2023, 7:37 pm |

Figure 78

resilient-badger-sqzect-dev-ed.trailblaze.lightning.force.com/lightning/r/Customer\_\_c/a002w00000ZE1o3AAD/view

Code Playground Leads Accounts Contacts Opportunities Cases Billings Customers

Customer Ravi

Related Details

|                 |  |                  |  |
|-----------------|--|------------------|--|
| * Customer Name | <input type="text" value="Ravi"/>        | Owner            | Ravi Teja Kuchipudi                      |
| Active          | <input checked="" type="checkbox"/>      |                  |  |
| Customer Type   | Premium                                  |                  |  |
| Description     | <input type="text"/>                     |                  |  |
| Created By      | Ravi Teja Kuchipudi, 22/02/2023, 7:25 am | Last Modified By | Ravi Teja Kuchipudi, 25/02/2023, 7:37 pm |

Cancel Save

Figure 79

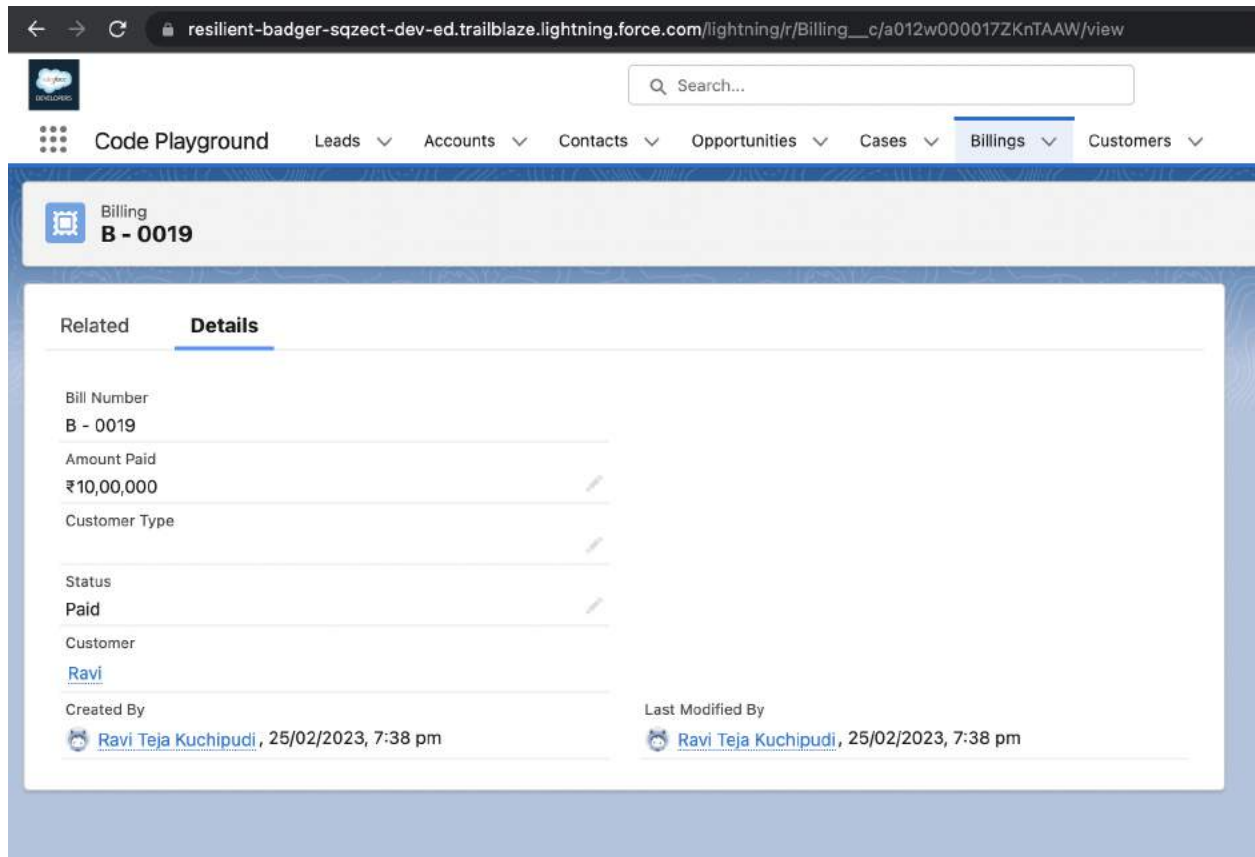


Figure 80

**Exercise 13:** Write a Test Class for CustomerTrigger.

1. Create the Data for the Customer custom Object
2. Use Test.startTest() & Test.stopTest().
3. Use assert statements to validate the output, Test Class must have full code coverage.



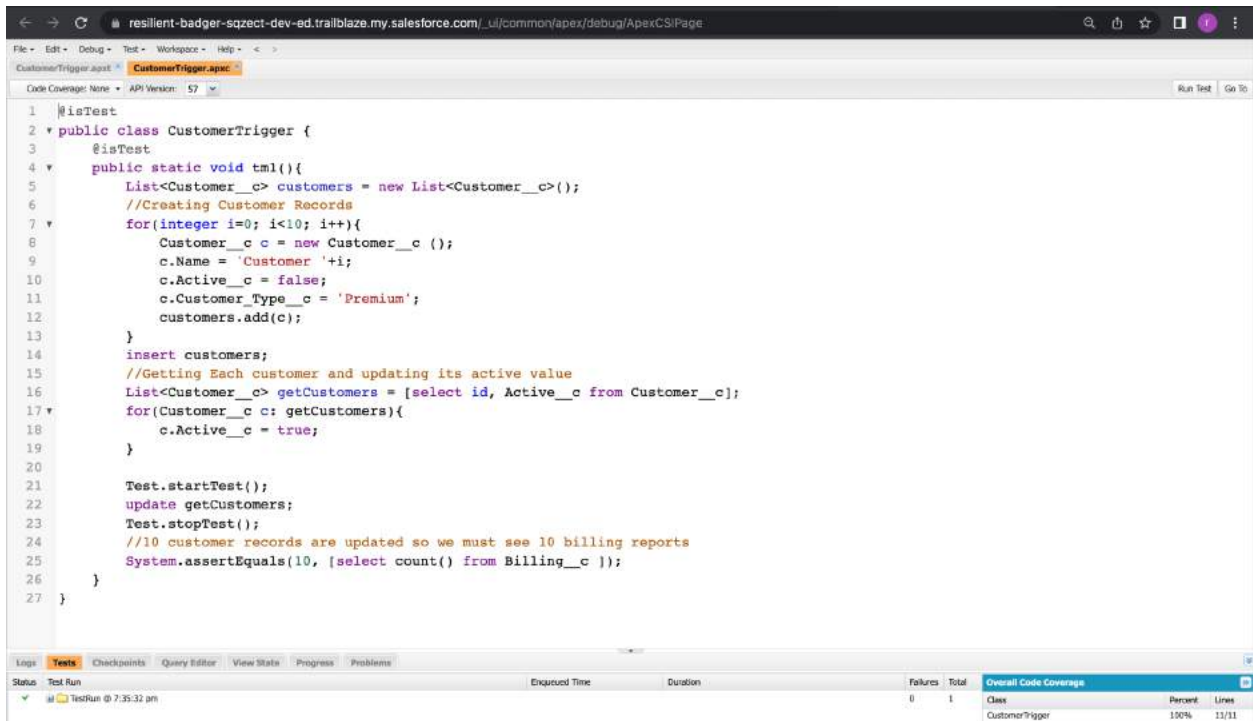


Figure 81

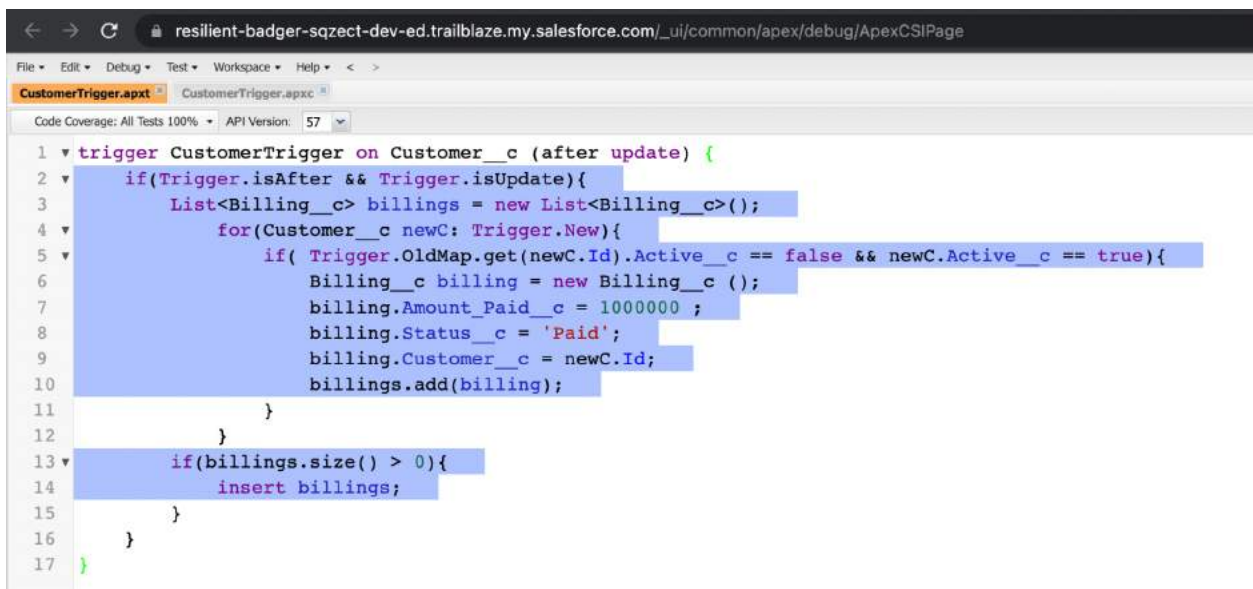
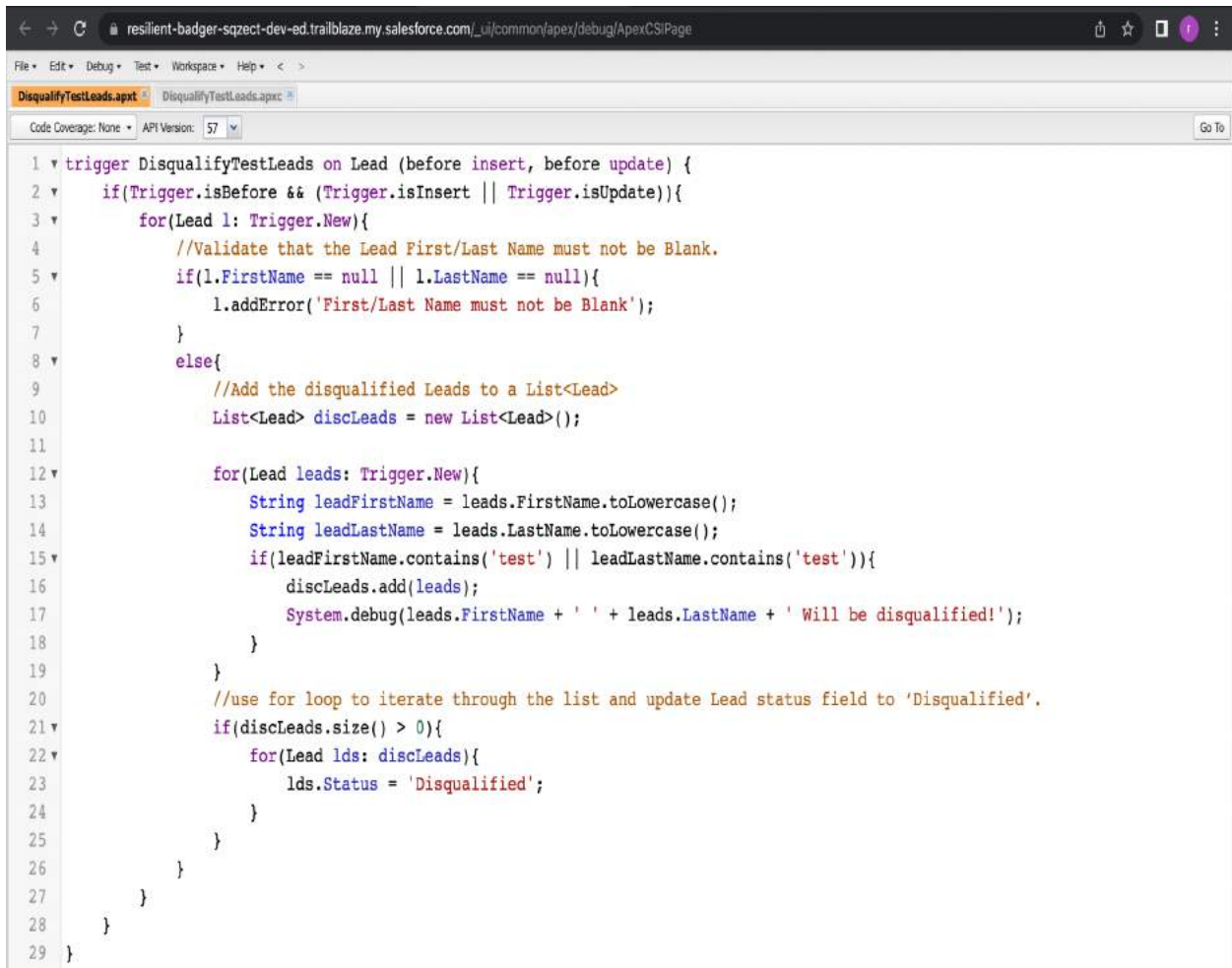


Figure 82



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

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



```
1 trigger DisqualifyTestLeads on Lead (before insert, before update) {
2     if (Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)) {
3         for (Lead l: Trigger.New) {
4             //Validate that the Lead First/Last Name must not be Blank.
5             if (l.FirstName == null || l.LastName == null) {
6                 l.addError('First/Last Name must not be Blank');
7             }
8             else {
9                 //Add the disqualified Leads to a List<Lead>
10                List<Lead> discLeads = new List<Lead>();
11
12                for (Lead leads: Trigger.New) {
13                    String leadFirstName = leads.FirstName.toLowerCase();
14                    String leadLastName = leads.LastName.toLowerCase();
15                    if (leadFirstName.contains('test') || leadLastName.contains('test')) {
16                        discLeads.add(leads);
17                        System.debug(leads.FirstName + ' ' + leads.LastName + ' Will be disqualified!');
18                    }
19                }
20                //use for loop to iterate through the list and update Lead status field to 'Disqualified'.
21                if (discLeads.size() > 0) {
22                    for (Lead lds: discLeads) {
23                        lds.Status = 'Disqualified';
24                    }
25                }
26            }
27        }
28    }
29 }
```

Figure 83

resilient-badger-sqzect-dev-ed.trailblaze.lightning.force.com/lightning/o/Lead/new?count=5&nooverride=1&useRecordTypeCheck=1&navigationLocation=LIST\_VI

Search...

New Lead

Lead Information

Lead Owner: Ravi Teja Kuchipudi

Phone:

Mobile:

Fax:

Email:

Website:

Lead Status: Open - Not Contacted

\* Name:  (highlighted)

Salutation: Mr. (dropdown)

First Name:  (highlighted)

\* Last Name:  (highlighted)

\* Company: RT Group (highlighted)

Title:

Lead Source: --None-- (dropdown)

Industry: --None-- (dropdown)

Annual Revenue:

We hit a snag.

Review the errors on this page.

- First/Last Name must not be Blank

Cancel Save & New Save

Figure 84

resilient-badger-sqzect-dev-ed.trailblaze.lightning.force.com/lightning/o/Lead/new?count=6&nooverride=1&useRecordTypeCheck=1&navigationLocation=LIST\_VI

Search...

New Lead

Lead Information

Lead Owner: Ravi Teja Kuchipudi

Phone:

Mobile:

Fax:

Email:

Website:

Lead Status: Open - Not Contacted

\* Name:  (highlighted)

Salutation: --None-- (dropdown)

First Name: tEsT (highlighted)

\* Last Name:  (highlighted)

\* Company: RT Group (highlighted)

Title:

Lead Source: --None-- (dropdown)

Industry: --None-- (dropdown)

Annual Revenue:

Cancel Save & New Save

Figure 85

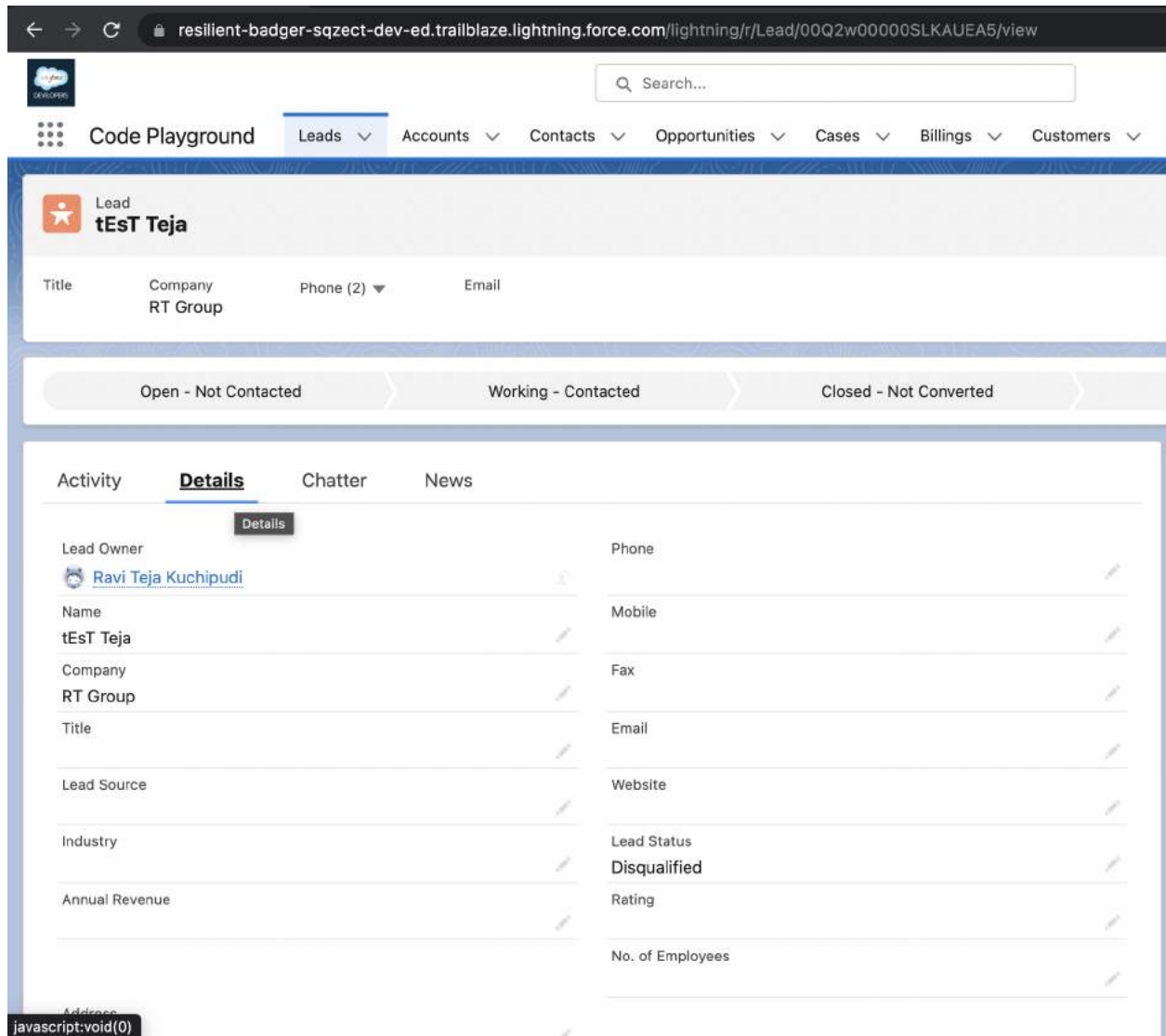


Figure 86

**Exercise 15:** Write a Test Class for `DisqualifyTestLeads`.

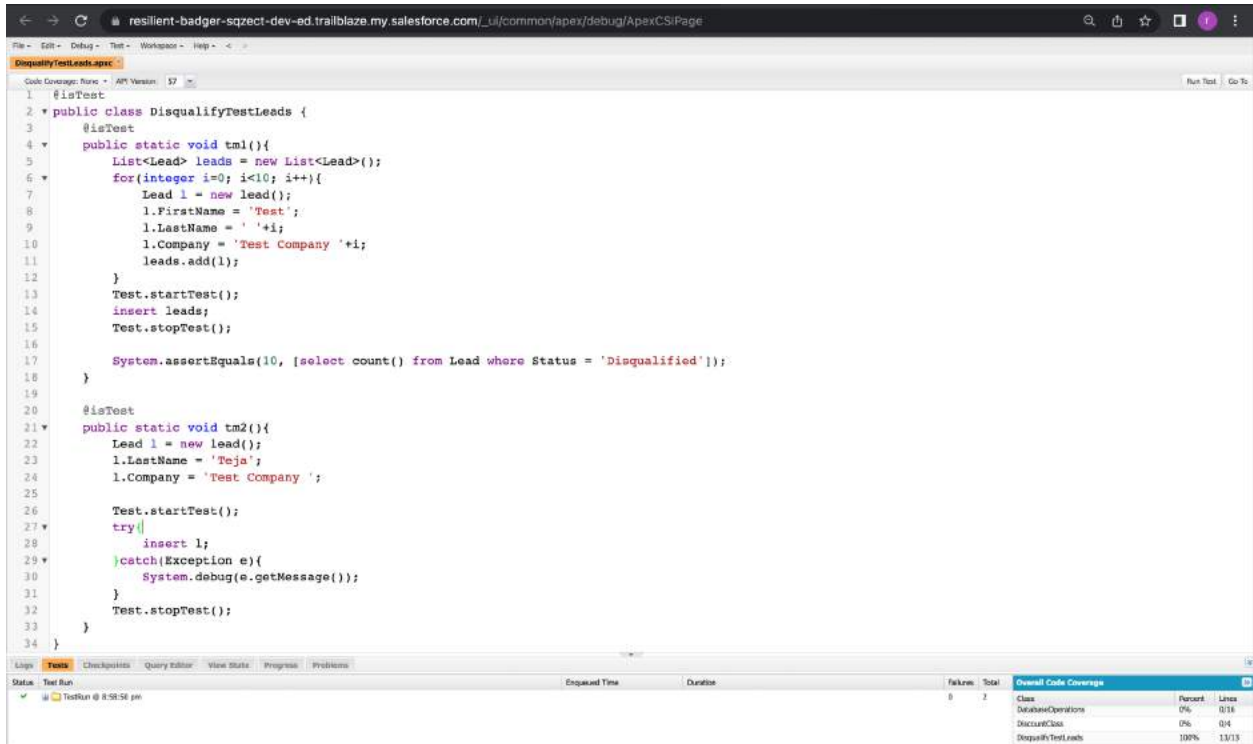


Figure 87

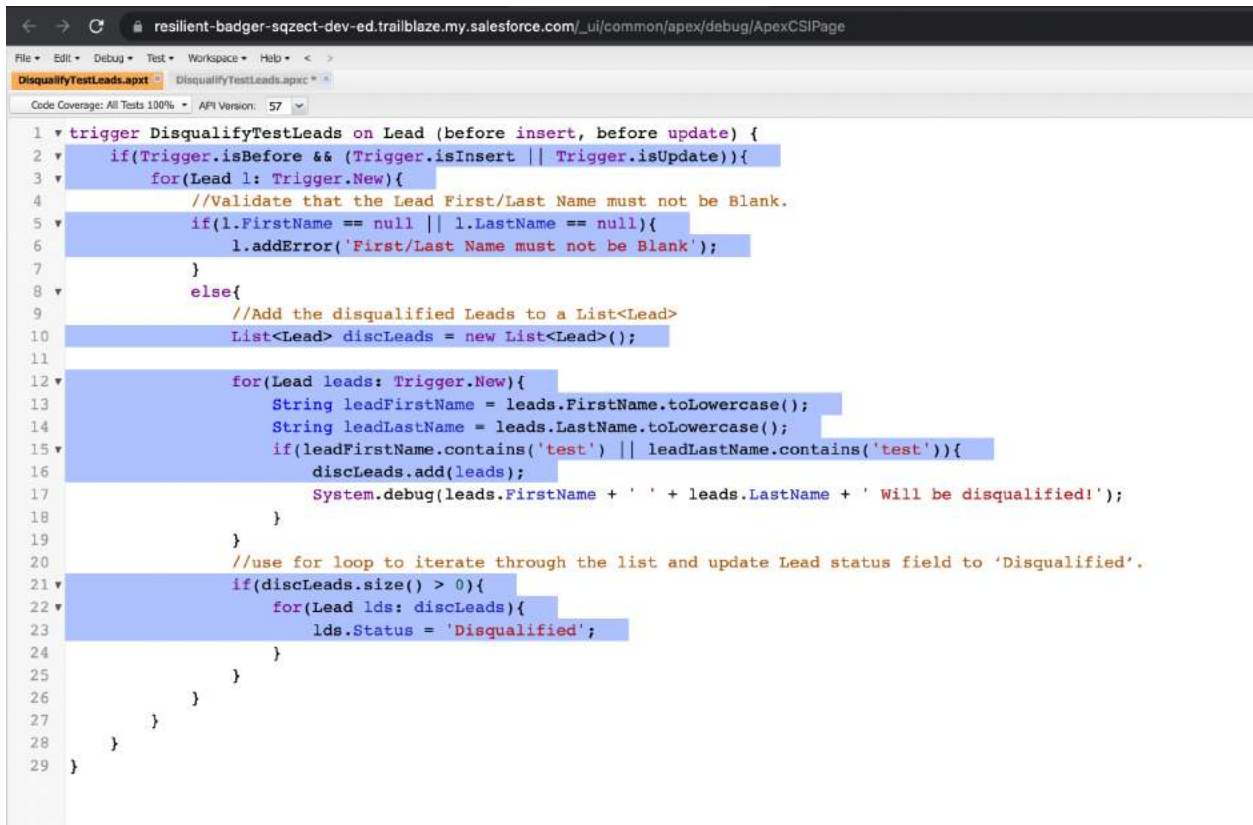


Figure 88

## Exercise 16:

1. Create a Visualforce page which displays Opportunity fields as output fields.
2. The page must be named OppView, It must reference the Opportunity standard controller.
3. Define 4 apex:outputField components: Opportunity Name, Amount, Close Date, Opportunity Account Name.

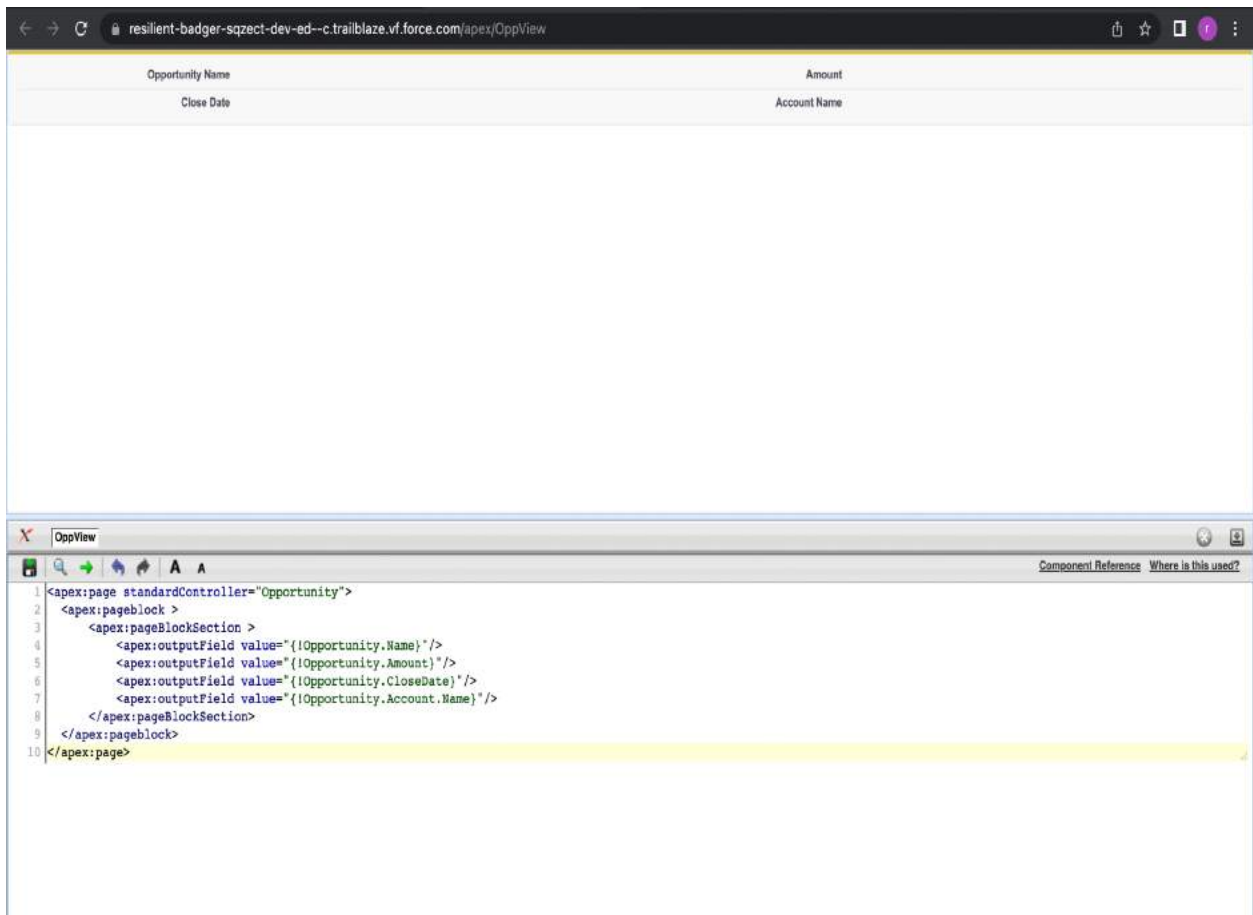


Figure 89

## Exercise 17:

1. Create a Visualforce page which shows a list of Accounts linked to their record pages
2. The page must be named AccountList
3. It must reference the Account standard controller.
4. It must have a recordSetVar attribute equal to accounts
5. It must have a Visualforce apex:repeat component, with the following:
6. Use the var attribute set to “a”, Use the <li> HTML list tag
7. Use the apex:outputLink component to link to the respective record detail page

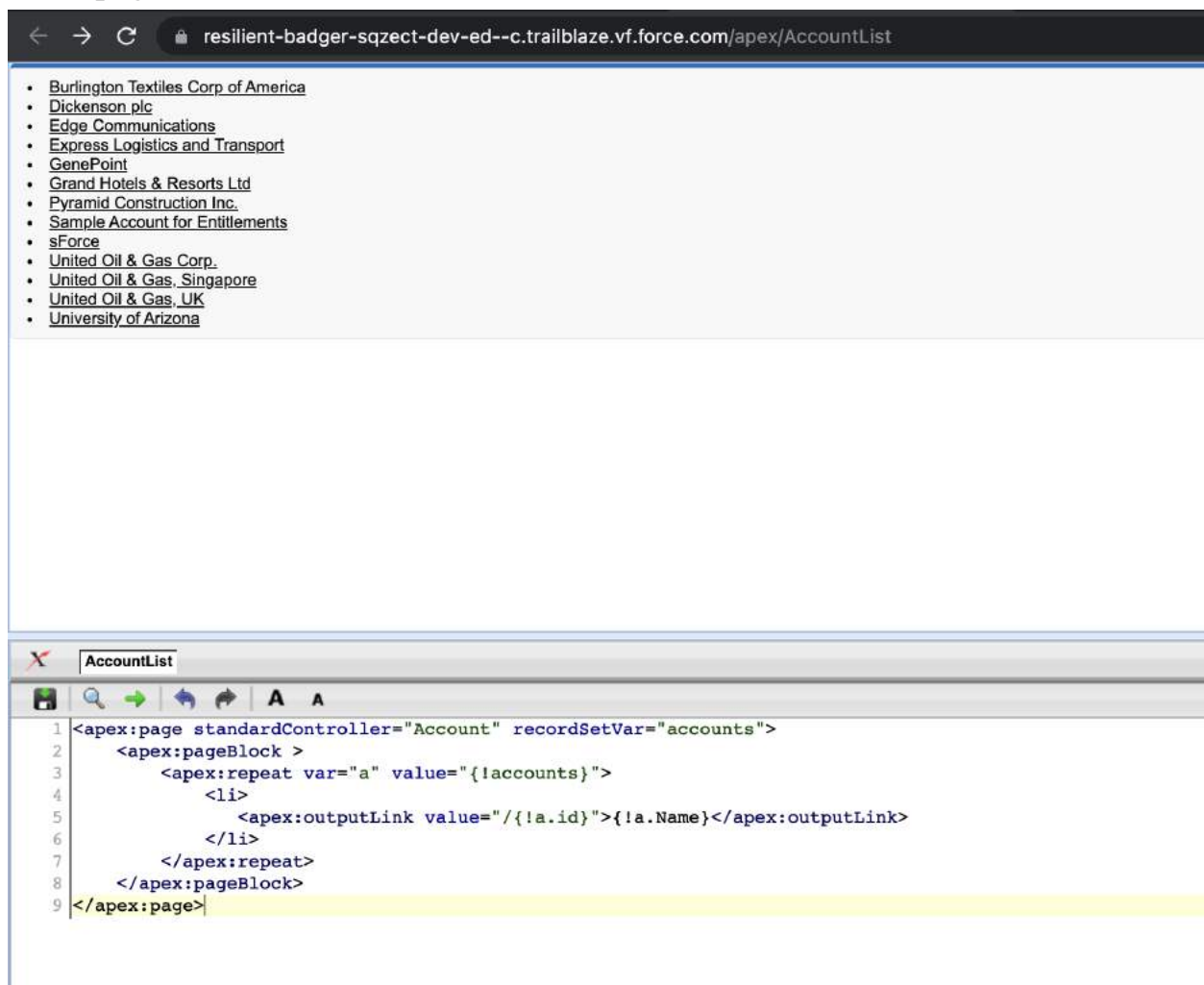
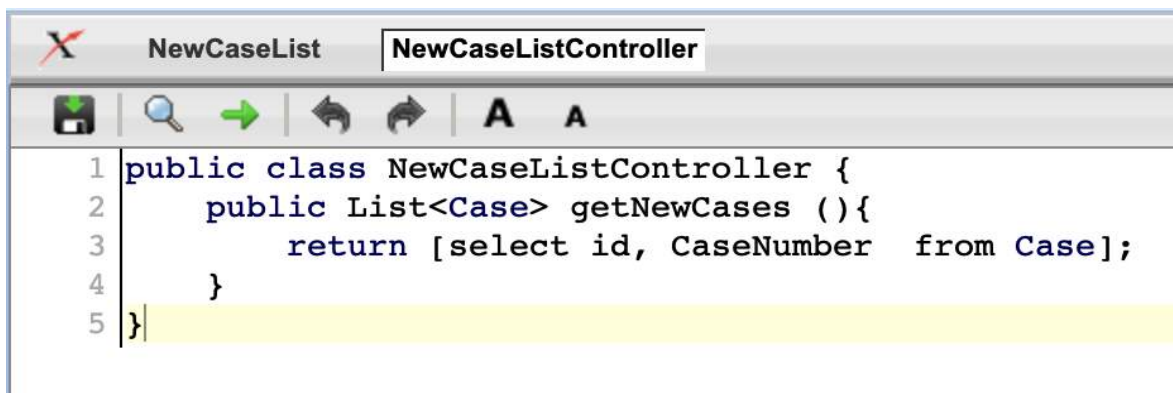


Figure 90

### Exercise 18:

1. Create a Visualforce page that uses a custom controller to display a list of cases with the status of 'New'. The page must be named NewCaseList.
2. The custom controller Apex class must be named NewCaseListController and include the following:
3. A publicly scoped method named getNewCases | Use the return type of List<Case>
4. Return a list of case records that includes the ID and CaseNumber fields..
5. Filter the results returned to only have a status of New.
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.\



The screenshot shows the Salesforce IDE interface. At the top, there are two tabs: 'NewCaseList' and 'NewCaseListController'. The 'NewCaseListController' tab is active. Below the tabs is a toolbar with icons for saving, searching, running, undo, redo, and text formatting. The main area displays the following Apex code:

```
1 public class NewCaseListController {  
2     public List<Case> getNewCases () {  
3         return [select id, CaseNumber from Case];  
4     }  
5 }
```

Figure 91



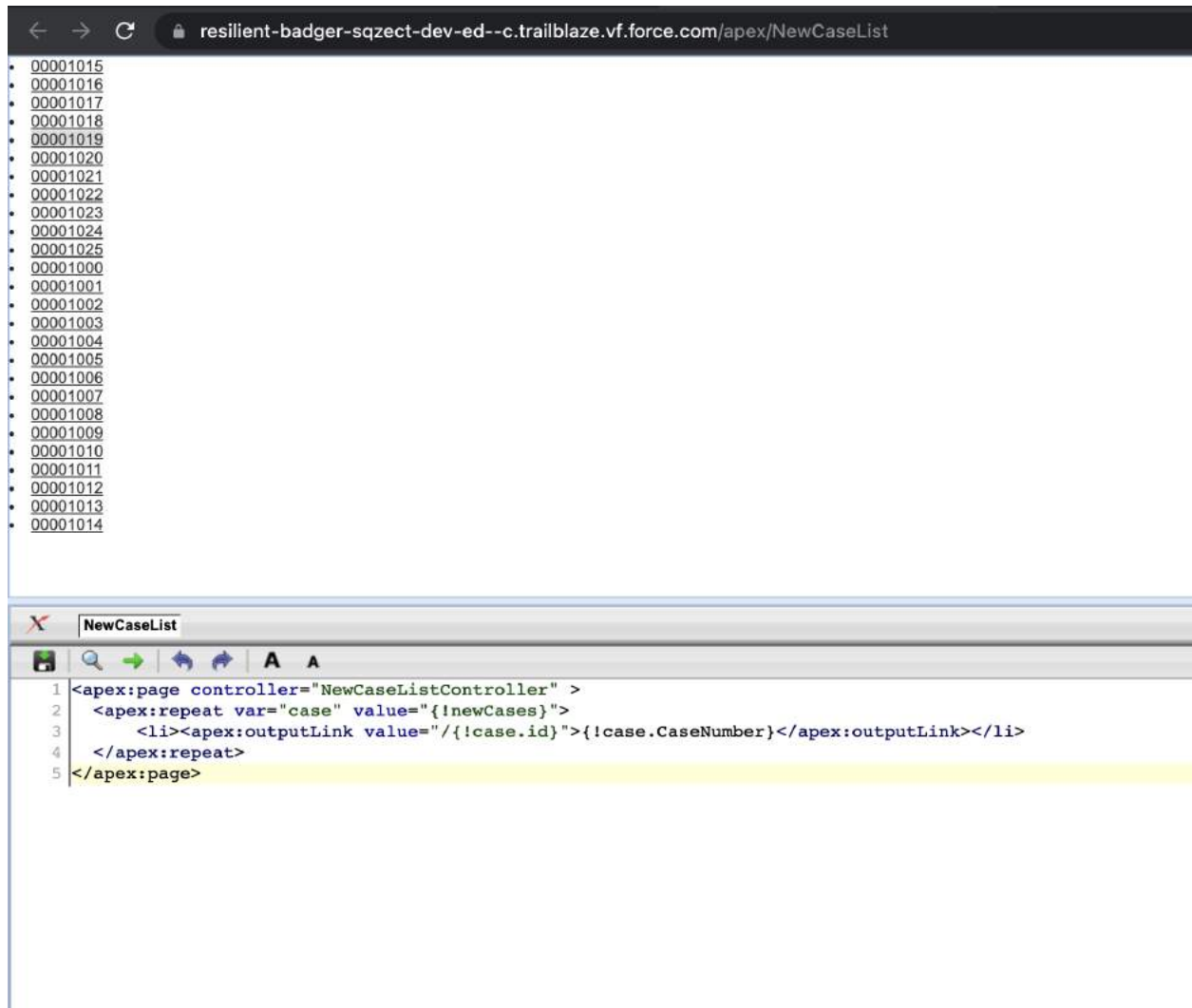


Figure 92

## References

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

## Github Link

[Github Link - Ravi Teja Kuchipudi](#)