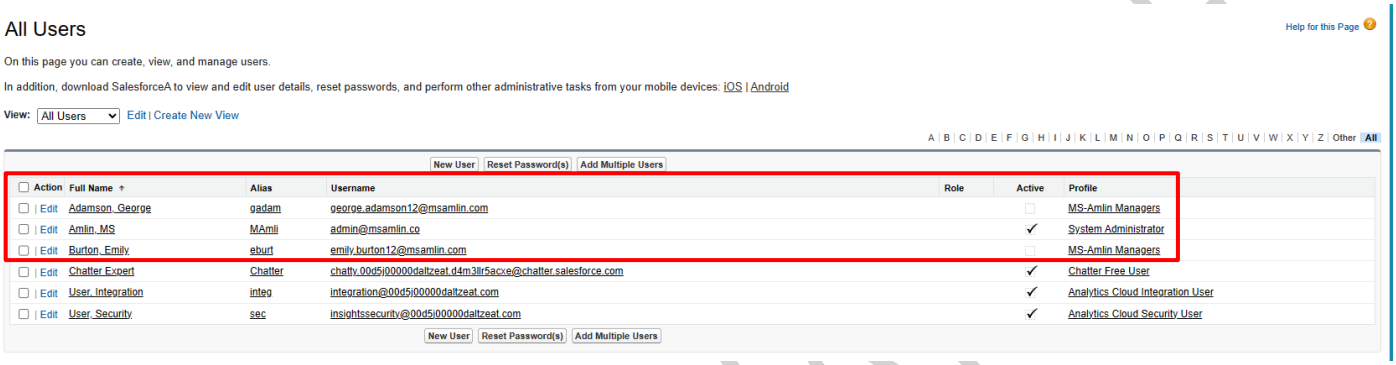
Salesforce Project

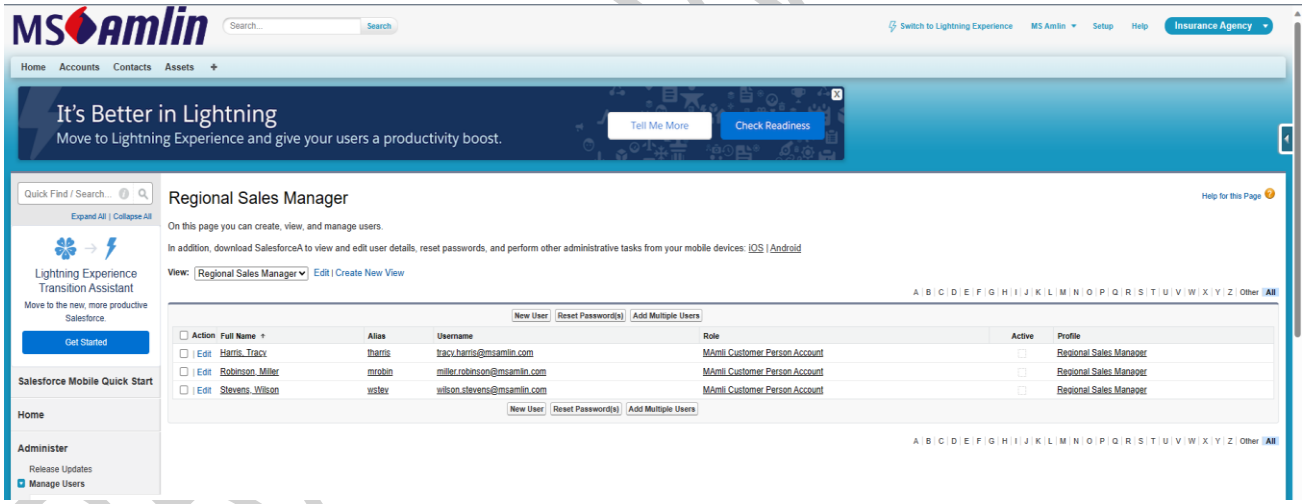
Name	Pranav Warade
Batch	23
Project Name	MS-Amlin Insurance Agency

1) Profile Name and Internal Users:

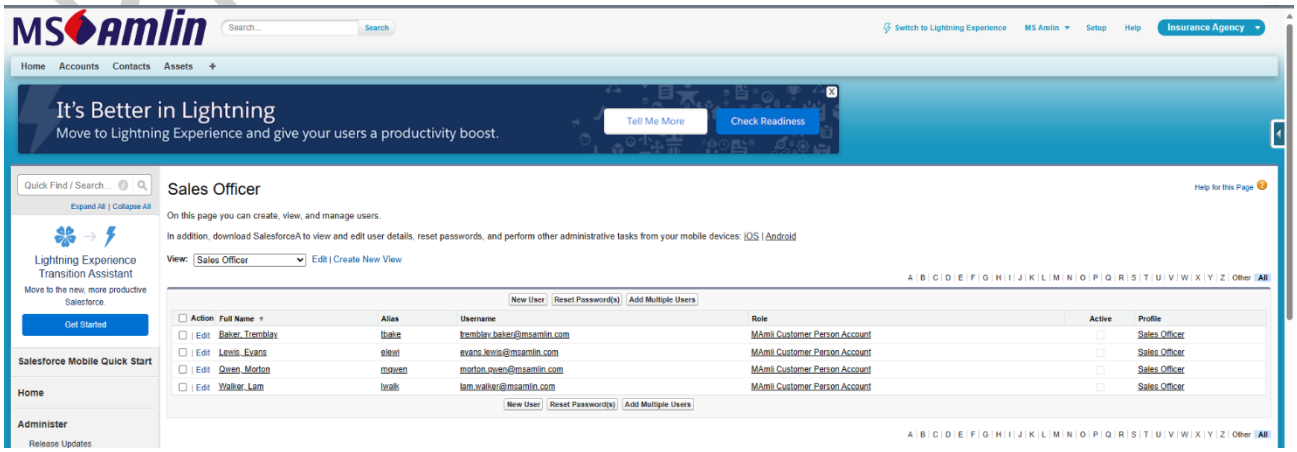


2) Community Portal Users:

Regional Sales Managers-



Sales Officer-



1. Account Object (Insurance Agency Details):

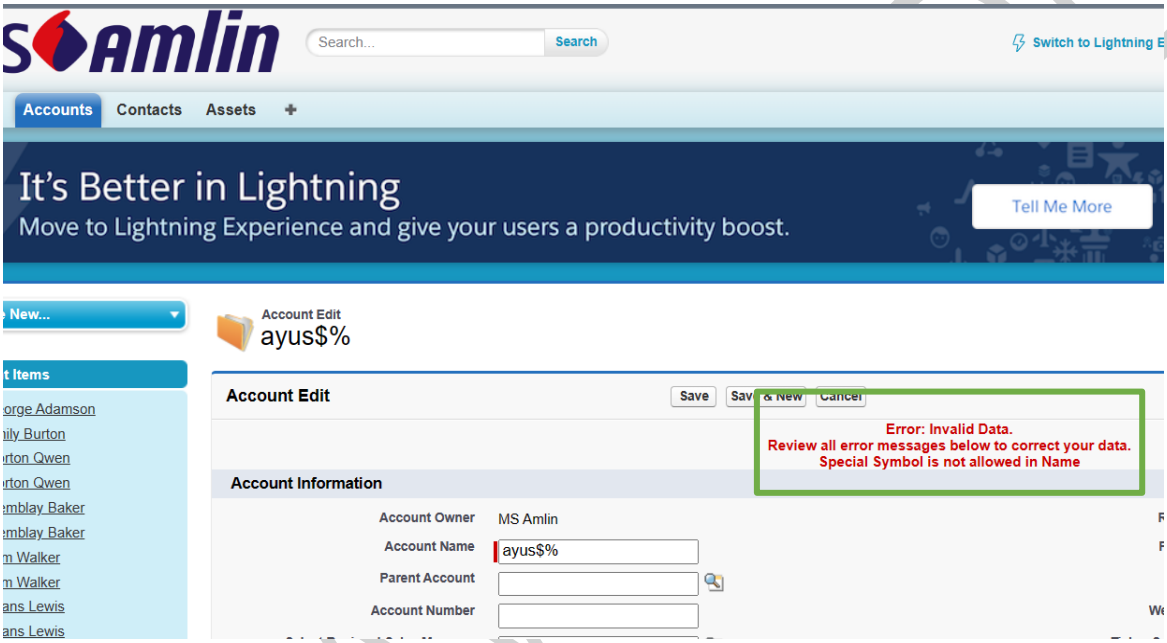
Create Custom fields→

Account Custom Fields & Relationships — New | Field Dependencies — Account Custom Fields & Relationships Help

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edit | Del | Replace	Active	Active__c	Picklist			MS Amlin, 21/09/2023, 10:38 pm
Edit | Del | Replace	Customer Priority	CustomerPriority__c	Picklist			MS Amlin, 21/09/2023, 10:38 pm
Edit | Del	Number of Locations	NumberofLocations__c	Number(3, 0)			MS Amlin, 21/09/2023, 10:38 pm
Edit | Del	Select Regional Sales Manager	Select_Regional_Sales_Manager__c	Lookup(Contact)	✓		MS Amlin, 23/09/2023, 12:58 am
Edit | Del | Replace	SLA	SLA__c	Picklist			MS Amlin, 21/09/2023, 10:38 pm
Edit | Del	SLA Expiration Date	SLAExpirationDate__c	Date			MS Amlin, 21/09/2023, 10:38 pm
Edit | Del	SLA Serial Number	SLASerialNumber__c	Text(10)			MS Amlin, 21/09/2023, 10:38 pm
Edit | Del	Total Contacts of 3-5 Rating	Total_Contacts_of_3_5_Rating__c	Number(10, 0)			MS Amlin, 23/09/2023, 1:05 am
Edit | Del | Replace	Upsell Opportunity	UpsellOpportunity__c	Picklist			MS Amlin, 21/09/2023, 10:38 pm

Requirements:

A] Prevent Special symbol in Acc Name.



Account Validation Rule

Back to Account Validation Rules

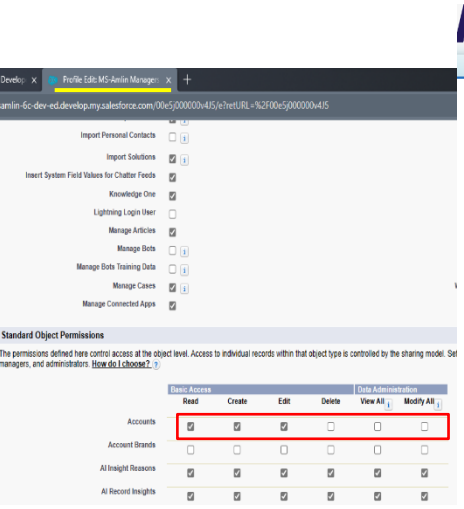
Validation Rule Detail

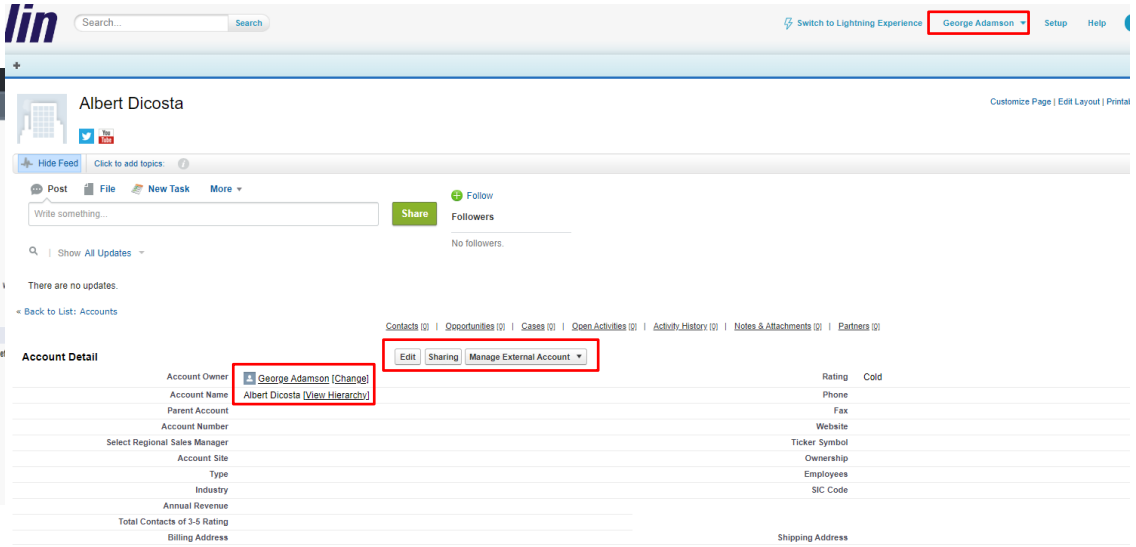
Edit | Clone

Rule Name	Prevent_Special_Sym_in_Acc_Name	Active	✓
Error Condition Formula	NOT(REGEX(Name , "^[A-z]*$"))		
Error Message	Special Symbol is not allowed in Name	Error Location	Top of Page
Description	No any special symbol in the account name		
Created By	MS Amlin, 25/09/2023, 5:19 pm	Modified By	MS Amlin, 25/09

Edit | Clone

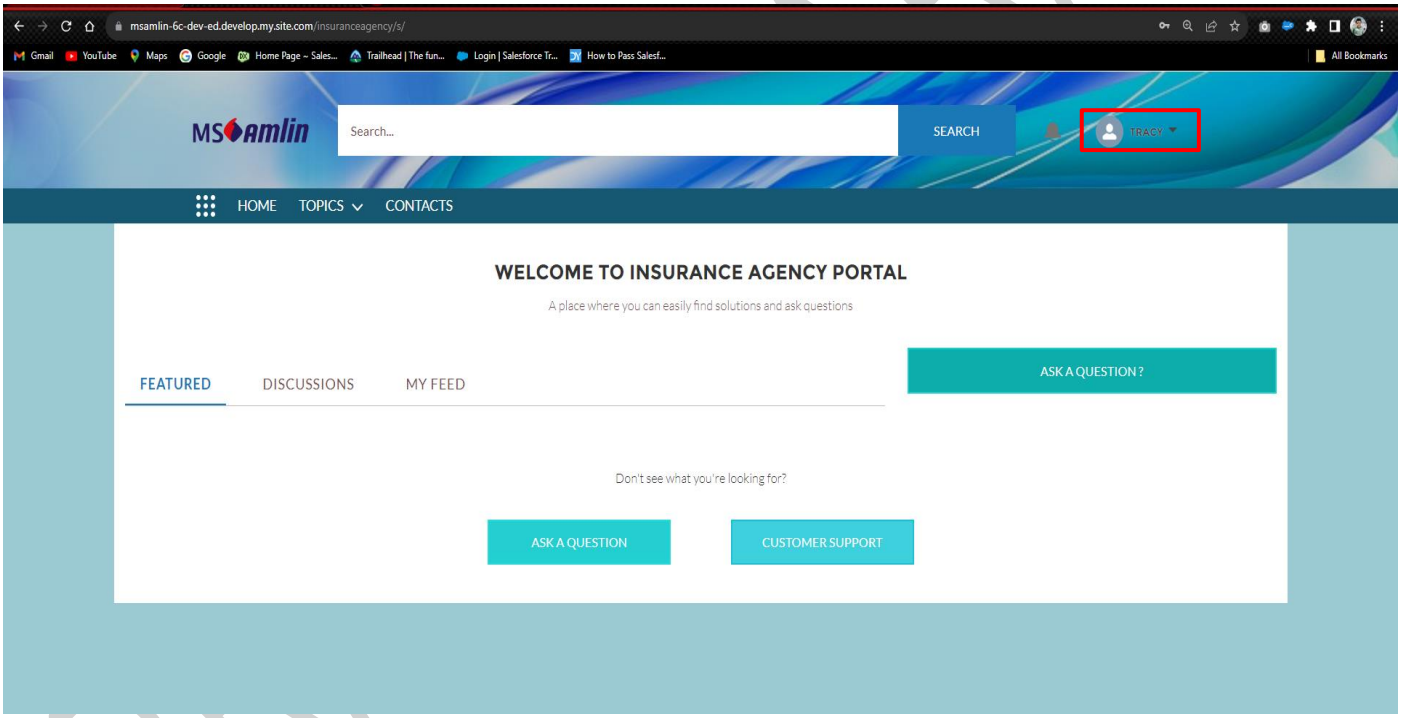
B] MS-Amlin profile user can only create, edit, update on the Account object.

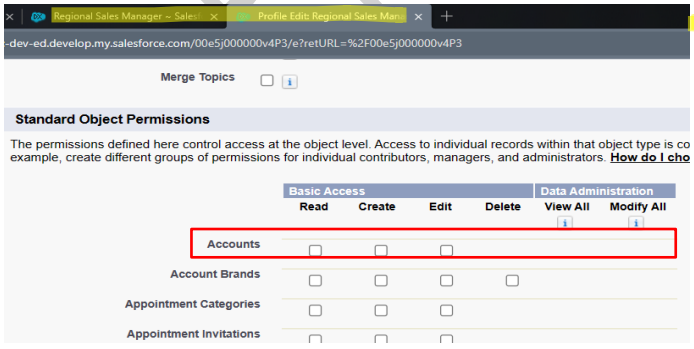


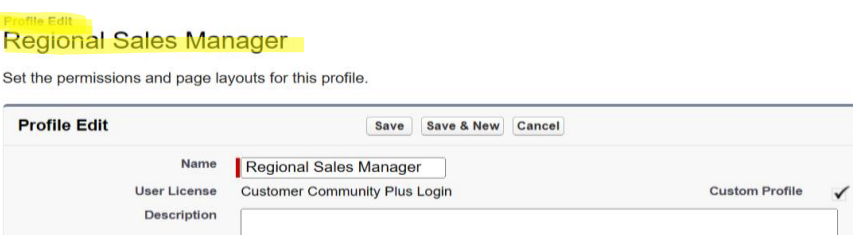


C] Regional Sales Managers and Sales Officers should not create, edit, update the Account.

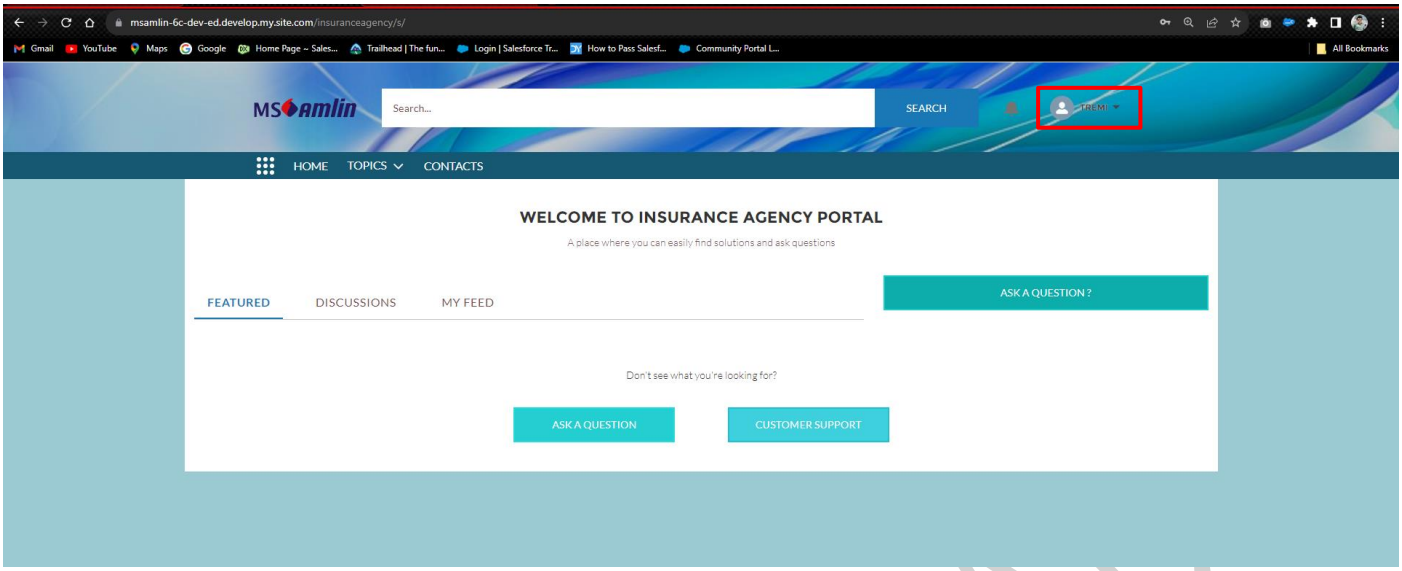
Regional Sales Managers:

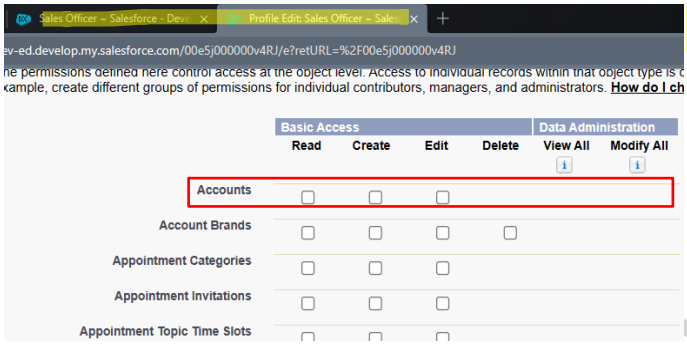


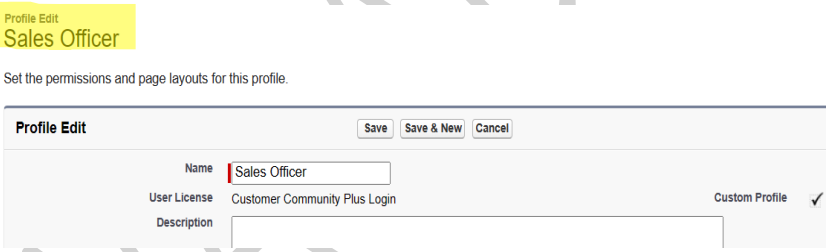




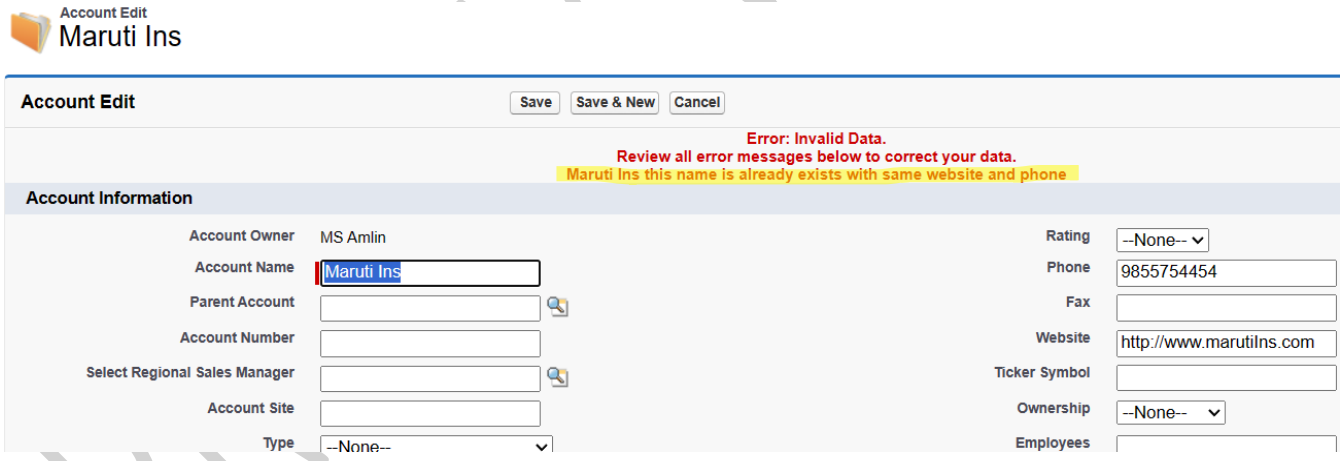
Sales Officer:

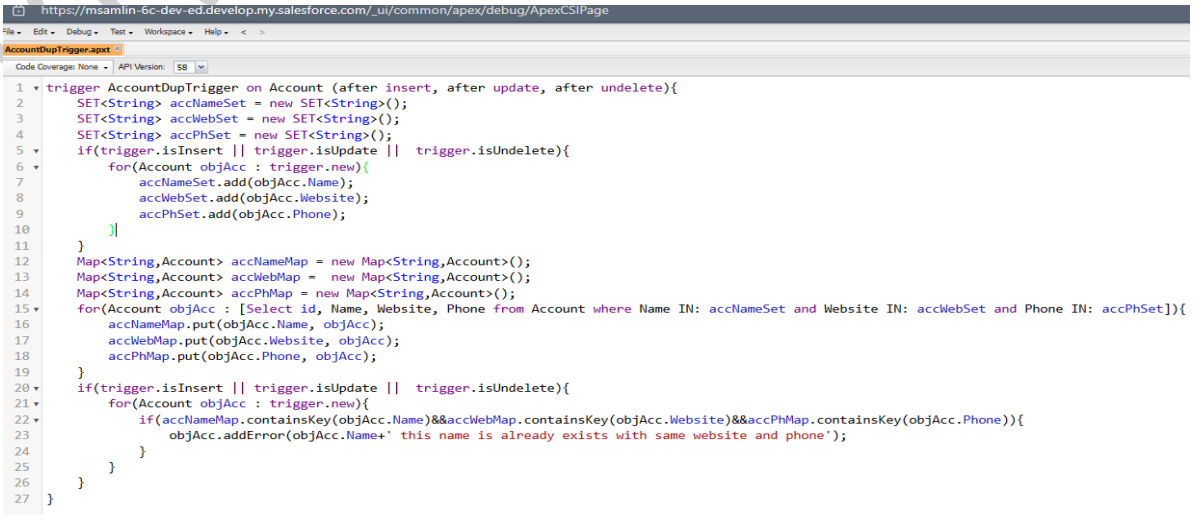






D] Prevent Duplicate Account Name with Website and Phone





```
trigger AccountDupTrigger on Account (after insert, after update, after undelete){
    SET<String> accNameSet = new SET<String>();
    SET<String> accWebSet = new SET<String>();
    SET<String> accPhSet = new SET<String>();
    if(trigger.isInsert || trigger.isUpdate ||  trigger.isUndelete){
        for(Account objAcc : trigger.new){
            accNameSet.add(objAcc.Name);
            accWebSet.add(objAcc.Website);
            accPhSet.add(objAcc.Phone);
        }
    }
    Map<String,Account> accNameMap = new Map<String,Account>();
    Map<String,Account> accWebMap =  new Map<String,Account>();
    Map<String,Account> accPhMap = new Map<String,Account>();
    for(Account objAcc : [Select id, Name, Website, Phone from Account where Name IN: accNameSet and Website IN: accWebSet and Phone IN: accPhSet]){
        accNameMap.put(objAcc.Name, objAcc);
        accWebMap.put(objAcc.Website, objAcc);
        accPhMap.put(objAcc.Phone, objAcc);
    }
    if(trigger.isInsert || trigger.isUpdate ||  trigger.isUndelete){
        for(Account objAcc : trigger.new){
            if(accNameMap.containsKey(objAcc.Name)&&accWebMap.containsKey(objAcc.Website)&&accPhMap.containsKey(objAcc.Phone)){
                objAcc.addError(objAcc.Name+' this name is already exists with same website and phone');
            }
        }
    }
}
```

```
trigger AccountDupTrigger on Account (after insert, after update, after undelete){
   SET<String> accNameSet = new SET<String>();
   SET<String> accWebSet = new SET<String>();
   SET<String> accPhSet = new SET<String>();
   if(trigger.isInsert || trigger.isUpdate ||  trigger.isUndelete){
      for(Account objAcc : trigger.new){
         accNameSet.add(objAcc.Name);
         accWebSet.add(objAcc.Website);
         accPhSet.add(objAcc.Phone);
      }
   }
   Map<String,Account> accNameMap = new Map<String,Account>();
   Map<String,Account> accWebMap =  new Map<String,Account>();
   Map<String,Account> accPhMap = new Map<String,Account>();
   for(Account objAcc : [Select id, Name, Website, Phone from Account where Name IN: accNameSet and Website IN: accWebSet and Phone IN: accPhSet]){
      accNameMap.put(objAcc.Name, objAcc);
      accWebMap.put(objAcc.Website, objAcc);
      accPhMap.put(objAcc.Phone, objAcc);
   }

   if(trigger.isInsert || trigger.isUpdate ||  trigger.isUndelete){
      for(Account objAcc : trigger.new){
         if(accNameMap.containsKey(objAcc.Name)&&accWebMap.containsKey(objAcc.Website)&&accPhMap.containsKey(objAcc.Phone)){
            objAcc.addError(objAcc.Name+' this name is already exists with same website and phone');
         }
      }
   }
}
```

2. Contact Object:

Create Record Type→

New Contact

Select Contact Record Type

Select a record type for the new contact. To skip this page in the future, change your record type settings on your personal setup page.

Select Contact Record Type

Record Type of new record: Regional Sales Manager

Regional Sales Manager
Sales Officer

Continue | Cancel

Available Contact Record Types

Record Type Name	Description
Regional Sales Manager	
Sales Officer	

Contact Record Type

Use this page to create and maintain record types for your organization. You can display differ

To use record types, add the Record Type field to your page layouts.

Contact Record Type

Action	Record Type Label
Edit | Del	Regional Sales Manager
Edit | Del	Sales Officer

Create contact custom field→ Cross formula field:

Account Detail — Edit | Delete | Sharing | Include

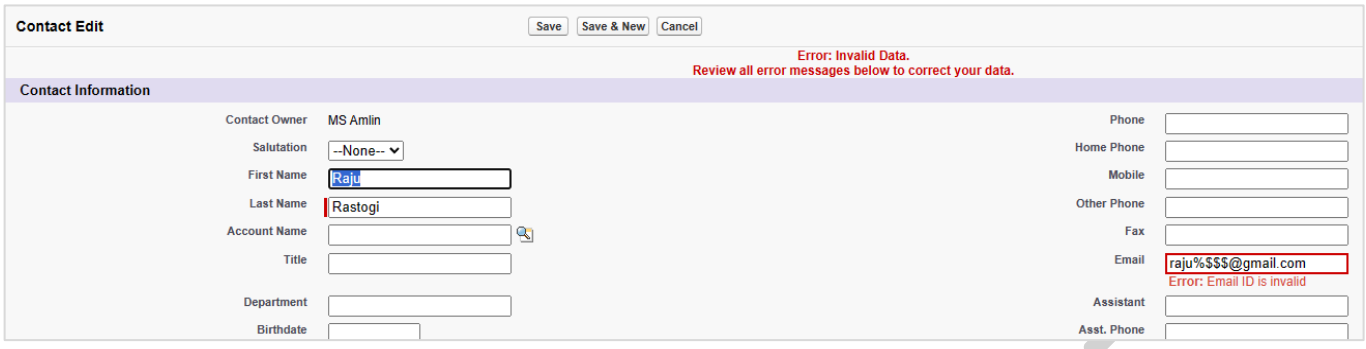
Account Owner	MS Amlin [Change]
Account Name	sForce [View Hierarchy]
Parent Account	
Account Number	
Select Regional Sales Manager	Tracy Harris
Account Site	
Type	
Industry	
Annual Revenue	
Total Contacts of 3-5 Rating	
Regional Sales Manager	Tracy Harris
Billing Address	San Francisco CA US
Customer Priority	
SLA Expiration Date	
Number of Locations	
Active	
Created By	MS Amlin, 21/09/2023, 10:38 pm

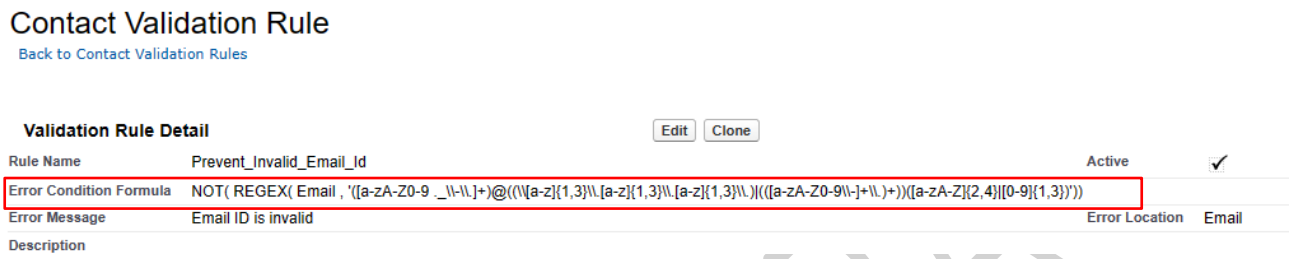
om Fields & Relationships — New | Field Dependencies

Field Label	API Name	Data Type
Active	Active__c	Picklist
Customer Priority	CustomerPriority__c	Picklist
Number of Locations	NumberofLocations__c	Number(3, 0)
Regional Sales Manager	Regional_Sales_Manager__c	Formula (Text)
Select Regional Sales Manager	Select_Regional_Sales_Manager__c	Lookup(Contact)
SLA	SLA__c	Picklist

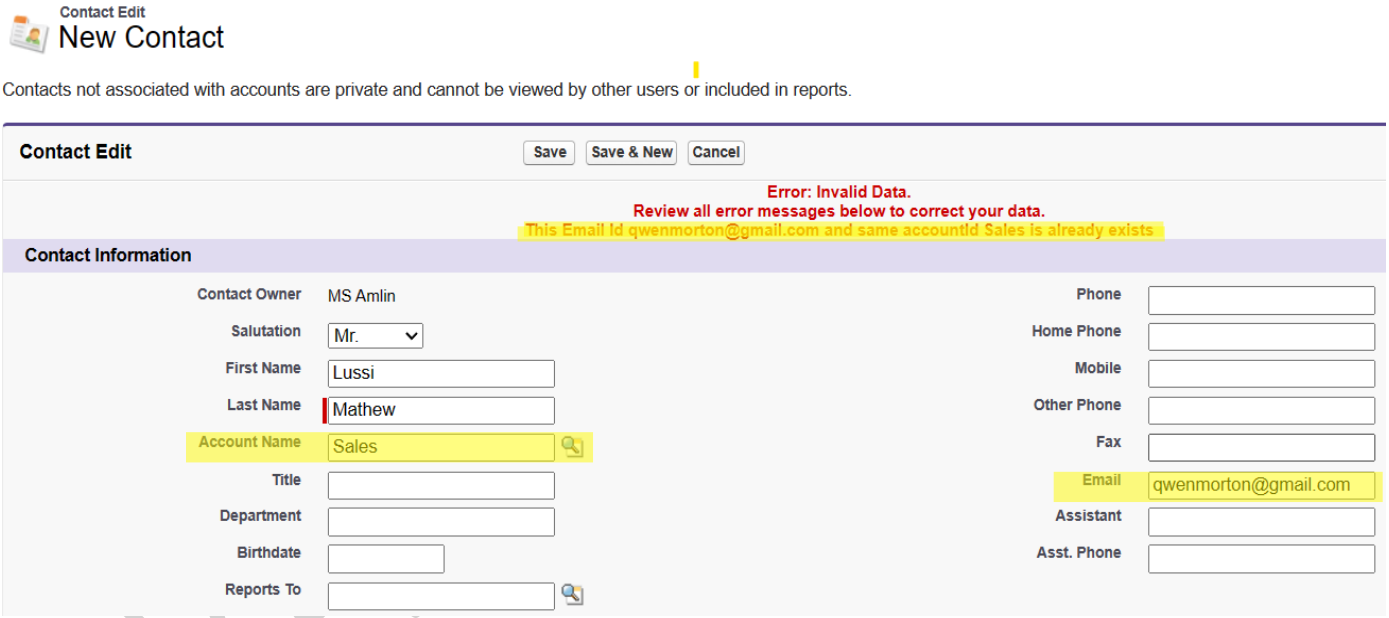
Requirements:

A] Prevent Invalid Email id.





B] Prevent Duplicate Contact with same Email Id from same Account.



DuplicateContactAccTrigger.apxt

```
trigger DuplicateContactAccTrigger on Contact (before insert, before update, after undelete){
    SET<ID> accIdSet = new SET<ID>();
    if( (trigger.isInsert || trigger.isUpdate || trigger.isUndelete) ){
        for(Contact objCon : trigger.new){
            if(trigger.isInsert || trigger.isUndelete){
                accIdSet.add(objCon.AccountId);
            }
            if(trigger.isUpdate){
                if(objCon.AccountId != trigger.oldMap.get(objCon.Id).AccountId){
                    accIdSet.add(objCon.AccountId);
                }
            }
        }
    }
    Map<Id,Account> accMap = new Map<Id, Account>();
    if(!accIdSet.isEmpty()){
        for(Account objAcc : [select Id, Name, (select Id, Name, Email, AccountId from Contacts) from Account where ID IN : accIdSet]){
            accMap.put(objAcc.Id,objAcc);
        }
    }
    if( (trigger.isInsert || trigger.isUpdate || trigger.isUndelete) ){
        for(Contact objCon : trigger.new){
            if(accMap.containsKey(objCon.AccountId)){
                List<Contact> ExistConList = accMap.get(objCon.AccountId).Contacts;
                for(Contact objExistingCon : ExistConList){
                    if(objCon.Email == objExistingCon.Email && objCon.AccountId == objExistingCon.AccountId){
                        objCon.addError('This Email Id '+objCon.Email+' and same accountId '+accMap.get(objCon.AccountId).Name+' is already exists');
                        break;
                    }
                }
            }
        }
    }
}
```

```
trigger DuplicateContactAccTrigger on Contact (before insert, before update, after undelete){
  SET<ID> accIdSet = new SET<ID>();
  if( (trigger.isInsert || trigger.isUpdate || trigger.isUndelete) ){
    for(Contact objCon : trigger.new){
      if(trigger.isInsert || trigger.isUndelete){
        accIdSet.add(objCon.AccountId);
      }
      if(trigger.isUpdate){
        if(objCon.AccountId != trigger.oldMap.get(objCon.Id).AccountId){
          accIdSet.add(objCon.AccountId);
        }
      }
    }
  }
  Map<Id,Account> accMap = new Map<Id, Account>();
  if(!accIdSet.isEmpty()){
    for(Account objAcc : [select Id, Name, (select Id, Name, Email, AccountId from Contacts) from Account where ID IN : accIdSet]){
      accMap.put(objAcc.Id,objAcc);
    }
  }
  if( (trigger.isInsert || trigger.isUpdate || trigger.isUndelete) ){
    for(Contact objCon : trigger.new){
      if(accMap.containsKey(objCon.AccountId)){
        List<Contact> ExistConList = accMap.get(objCon.AccountId).Contacts;
        for(Contact objExistingCon : ExistConList){
          if(objCon.Email == objExistingCon.Email && objCon.AccountId == objExistingCon.AccountId){
            objCon.addError('This Email Id '+objCon.Email+' and same accountId '+accMap.get(objCon.AccountId).Name+' is already exists');
            break;
          }
        }
      }
    }
  }
}
```

3. Applicant Details Object:

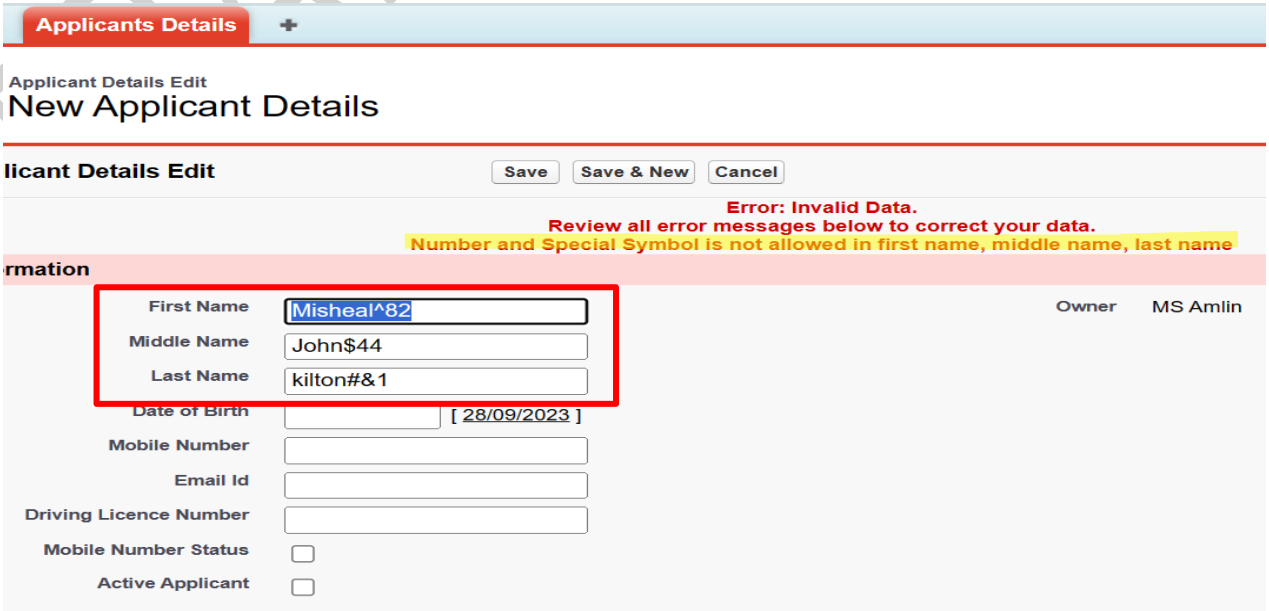
Create Applicant Details custom field→

Custom Fields & Relationships — New | Field Dependencies | Set History Tracking

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edit | Del	Active Applicant	Active_Applicant__c	Checkbox			MS Amlin, 27/09/2023, 1:50 pm
Edit | Del	Date of Birth	Date_of_Birth__c	Date			MS Amlin, 27/09/2023, 1:35 pm
Edit | Del	Driving Licence Number	Driving_Licence_Number__c	Text(20)			MS Amlin, 27/09/2023, 1:43 pm
Edit | Del	Email Id	Email_Id__c	Email			MS Amlin, 27/09/2023, 1:40 pm
Edit | Del	First Name	First_Name__c	Text(18)			MS Amlin, 27/09/2023, 1:19 pm
Edit | Del	Last Name	Last_Name__c	Text(18)			MS Amlin, 27/09/2023, 1:31 pm
Edit | Del	Middle Name	Middle_Name__c	Text(18)			MS Amlin, 27/09/2023, 1:20 pm
Edit | Del	Mobile Number	Mobile_Number__c	Phone			MS Amlin, 27/09/2023, 1:37 pm
Edit | Del	Mobile Number Status	Mobile_Number_Status__c	Checkbox			MS Amlin, 27/09/2023, 1:48 pm

Requirements:

A] Prevent Number and Symbol in First Name, Middle Name and Last Name.



Applicant Details Validation Rule

Back to Applicant Details

Validation Rule Detail Edit Clone

Rule Name	Prevent_Num_and_Sym_in_FN_LN_MN	Active	✓
Error Condition Formula	AND(NOT(REGEX(First_Name__c , "^[a-zA-Z]*$")), NOT(REGEX(Last_Name__c , "^[a-zA-Z]*$")), NOT(REGEX(Middle_Name__c , "^[a-zA-Z]*$")))		
Error Message	Number and Special Symbol is not allowed in first name, middle name, last name	Error Location	Top of Page
Description			
Created By	MS Amlin, 28/09/2023, 1:46 am	Modified By	MS Amlin, 28/09/2023, 2:42 am

B] First Name and Last Name are not same.

Applicants Details +

Applicant Details Edit

New Applicant Details

Save Save & New Cancel

Error: Invalid Data.
Review all error messages below to correct your data.
First Name and Last Name should not be same

First Name	Ramu
Middle Name	
Last Name	Ramu

Applicant Details Validation Rule

Back to Applicant Details

Validation Rule Detail Edit Clone

Rule Name	First_Name_and_Last_Name_not_same	Active	✓
Error Condition Formula	NOT(First_Name__c <> Last_Name__c)		
Error Message	First Name and Last Name should not be same	Error Location	Top o
Description			
Created By	MS Amlin, 28/09/2023, 12:03 pm	Modified By	MS A

C] Prevent Invalid Email Id.

Applicants Details +

Applicant Details Edit

New Applicant Details

Save Save & New Cancel

Error: Invalid Data.
Review all error messages below to correct your data.

First Name	Manoj
Middle Name	Raghu
Last Name	Sharma
Date of Birth	28/09/2023 [28/09/2023]
Mobile Number	
Email Id	manoj#%@gmail.com
	Error: This is an invalid Email ID
Driving Licence Number	

Applicant Details Validation Rule

Back to Applicant Details

Validation Rule Detail Edit | Clone

Rule Name	Prevent_Invalid_Email_Id	Active	✓
Error Condition Formula	NOT(REGEX(Email_Id__c , '([a-zA-Z0-9 ._\\-\\.]+)@((\\[a-z]{1,3}\\.[a-z]{1,3}\\.[a-z]{1,3}\\.)|(([a-zA-Z0-9\\-]+\\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})'))		
Error Message	This is an invalid Email ID	Error Location	Email Id
Description			
Created By	MS Amlin, 28/09/2023, 2:06 am	Modified By	MS Amlin, 28/09/2023, 12:23 pm

4. **Document Details Object**:

Custom Object
Document Details

Standard Fields [4] | Custom Fields & Relationships [0] | Validation Rules [0]
Record Types [0] | Apex Sha

Custom Object Definition Detail Edit | Delete

Singular Label	Document Details
Plural Label	Documents Details
Object Name	Document_Details
API Name	Document_Details__c

Custom Fields & Relationships New | Field Dependencies | Set History Tracking

Action	Field Label	API Name	Data Type
Edit | Del	Id Number	Id_Number__c	Text(20)
Edit | Del	Select Application	Select_Application__c	Master-Detail(Applicant Details)
Edit | Del | Replace	Select Identity Proof Document	Select_Identity_Proof_Document__c	Picklist

Requirements:

A] Prevent Duplicate Document Details record for an Applicant as per Identity Proof Document.

Assets | Applicants Details | Documents Details | +

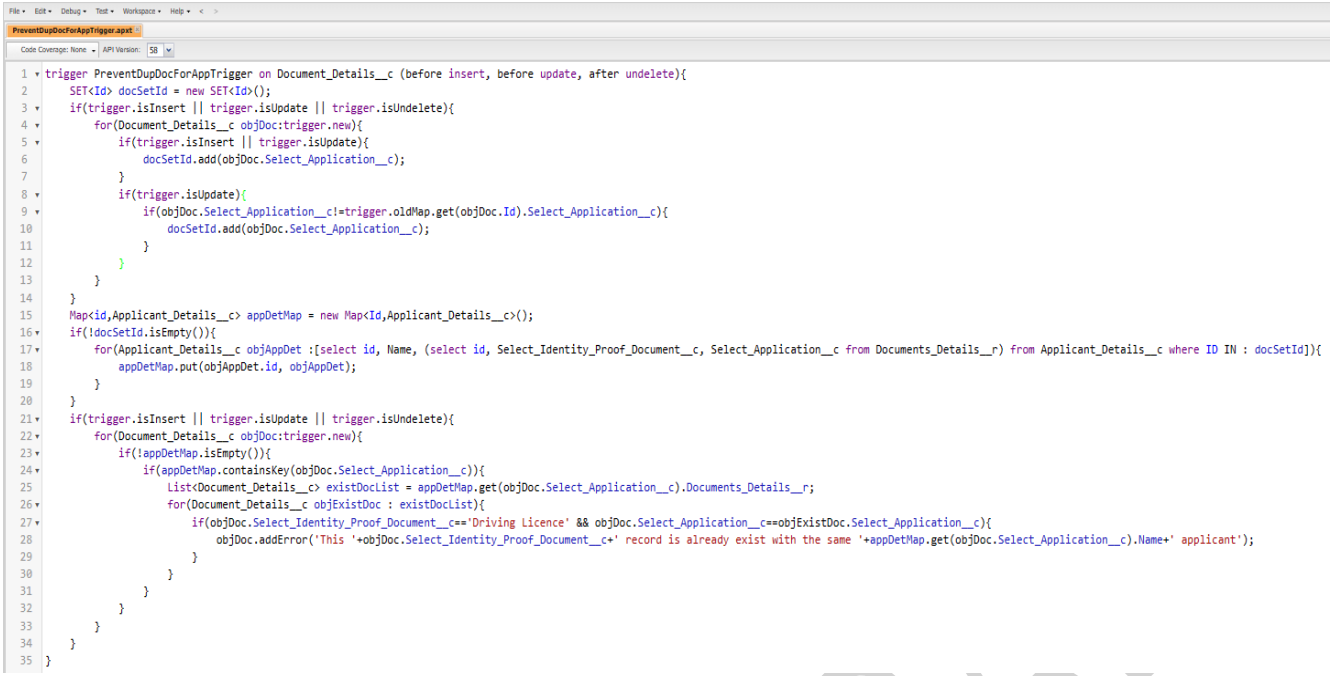
Document Details Edit
New Document Details

Document Details Edit Save | Save & New | Cancel

Error: Invalid Data.
Review all error messages below to correct your data.
This Driving Licence record is already exist with the same APPID-0003 applicant

Information

Select Application	APPID-0003
Select Identity Proof Document	Driving Licence
Id Number	34535



```
trigger PreventDupDocForAppTrigger on Document_Details__c (before insert, before update, after undelete){
    SET<Id> docSetId = new SET<Id>();
    if(trigger.isInsert || trigger.isUpdate || trigger.isUndelete){
        for(Document_Details__c objDoc:trigger.new){
            if(trigger.isInsert || trigger.isUpdate){
                docSetId.add(objDoc.Select_Application__c);
            }
            if(trigger.isUpdate){
                if(objDoc.Select_Application__c!=trigger.oldMap.get(objDoc.Id).Select_Application__c){
                    docSetId.add(objDoc.Select_Application__c);
                }
            }
        }
    }
    Map<id,Applicant_Details__c> appDetMap = new Map<Id,Applicant_Details__c>();
    if(!docSetId.isEmpty()){
        for(Applicant_Details__c objAppDet :[select id, Name, (select id, Select_Identity_Proof_Document__c, Select_Application__c from Documents_Details__r) from Applicant_Details__c where ID IN : docSetId]){
            appDetMap.put(objAppDet.id, objAppDet);
        }
    }
    if(trigger.isInsert || trigger.isUpdate || trigger.isUndelete){
        for(Document_Details__c objDoc:trigger.new){
            if(!appDetMap.isEmpty()){
                if(appDetMap.containsKey(objDoc.Select_Application__c)){
                    List<Document_Details__c> existDocList = appDetMap.get(objDoc.Select_Application__c).Documents_Details__r;
                    for(Document_Details__c objExistDoc : existDocList){
                        if(objDoc.Select_Identity_Proof_Document__c=='Driving Licence' && objDoc.Select_Application__c==objExistDoc.Select_Application__c){
                            objDoc.addError('This '+objDoc.Select_Identity_Proof_Document__c+' record is already exist with the same '+appDetMap.get(objDoc.Select_Application__c).Name+' applicant');
                        }
                    }
                }
            }
        }
    }
}
```

PRANAV WARA

5. **Assets Object:**

Create Assets custom field→

Asset Custom Fields & Relationships

New | Field Dependencies

Action	Field Label	API Name	Data Type	Indexed	Controlling Field
Edit | Del | Replace	Insurance Type	Insurance_Type__c	Picklist		

6. **Applicants-Assets Object:**

Create Assets custom field→

Custom Object

Applications-Assets

Standard Fields [4] | Custom Fields & Relationships [0] | Validation Rules [0] | Page Layc

Custom Object Definition Detail

Edit | Delete

Singular Label	Applications-Assets
Plural Label	Applications-Assets
Object Name	Applications_Assets
API Name	Applications_Assets__c

Custom Fields & Relationships

New | Field Dependencies | Set History Tracking

Action	Field Label	API Name	Data Type
Edit | Del	Down Payment	Down_Payment__c	Currency(18, 0)
Edit | Del	Finance Amount	Finance_Amount__c	Formula (Currency)
Edit | Del	Price	Price__c	Currency(18, 0)
Edit | Del	Select Application	Select_Application__c	Master-Detail(Applicant Details)
Edit | Del	Select Assets (Insurance)	Select_Assets_Insurance__c	Master-Detail(Asset)
Edit | Del	Tenure in Months	Tenure_in_Months__c	Number(18, 0)

Requirements:

A] Show Applicant Name in junction object.

Assets | Applicants Details | Documents Details | Applications-Assets

Applications-Assets

APP_ASSID-0001

« Back to List: Custom Object Definitions

Open Activities [0] | Activity H

Applications-Assets Detail

Edit | Delete | Clone

Applications-Assets ID	APP_ASSID-0001
Select Application	APPID-0003
Applicant Name	Rajiv Annaji Patil
Select Assets (Insurance)	Life Insurance
Tenure in Months	
Price	₹40,000
Down Payment	₹10,000

Field Information

Field Label	Applicant Name
Field Name	Applicant_Name
API Name	Applicant_Name__c
Description	
Help Text	
Data Owner	
Field Usage	
Data Sensitivity Level	
Compliance Categorization	
Created By	MS Amlin, 28/09/2023, 11:42 pm

Formula Options

Data Type: Formula

```
Select_Application__r.First_Name__c +' '+ Select_Application__r.Middle_Name__c +' '+ Select_Application__r.Last_Name__c
```

B] Show Assets Name and Serial Number in junction object.

Applications-Assets

APP_ASSID-0001

« Back to List: Custom Object Definitions

Applications-Assets Detail

Applications-Assets ID	APP_ASSID-0001
Select Application	APPID-0003
Applicant Name	Rajiv Annaji Patil
Select Assets (Insurance)	Life Insurance
Assets Name and Serial no	Life Insurance and INS2564
Tenure in Months	
Price	₹40,000
Down Payment	₹10,000
Finance Amount	₹30,000.0

Applications-Assets Custom Field

Assets Name and Serial no

Back to Applications-Assets

Custom Field Definition Detail

Field Information

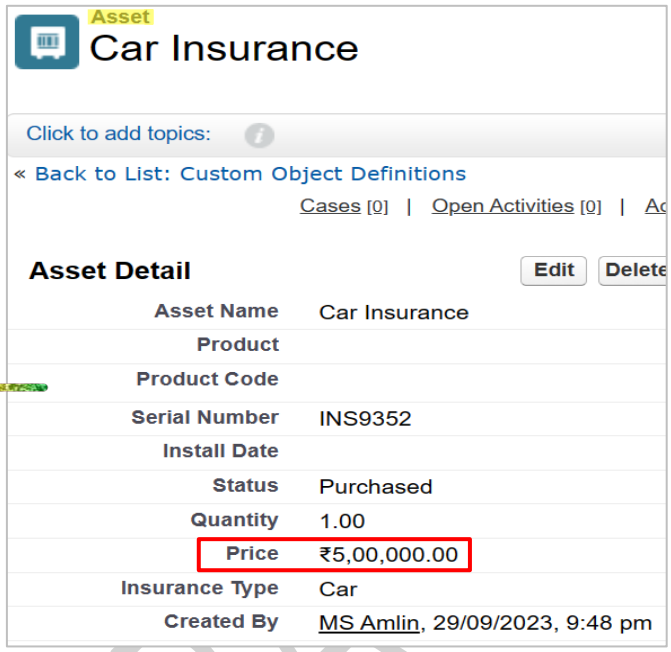
Field Label	Assets Name and Serial no
Field Name	Assets_Name_and_Serial_no
API Name	Assets_Name_and_Serial_no__c
Description	
Help Text	
Data Owner	
Field Usage	
Data Sensitivity Level	
Compliance Categorization	
Created By	MS Amlin, 29/09/2023, 12:30 am

Formula Options

Data Type: Formula

```
Select_Assets_Insurance__r.Name +' and '+ Select_Assets_Insurance__r.SerialNumber
```

C] Copy Asset Price field to the Application-Asset Price field.



AppAssToAssCopyPriceTrigger.apxt

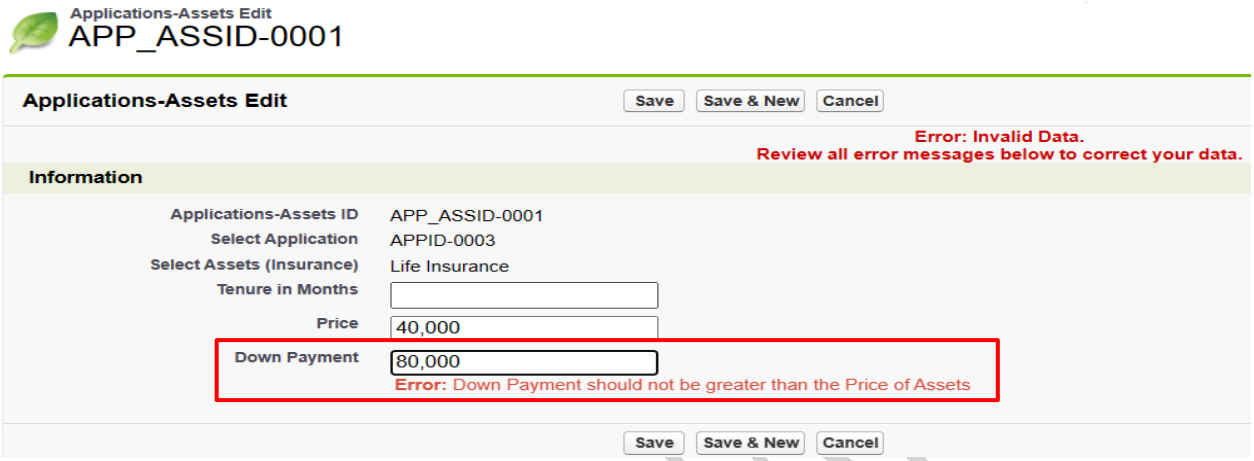
Code Coverage: None | API Version: 58

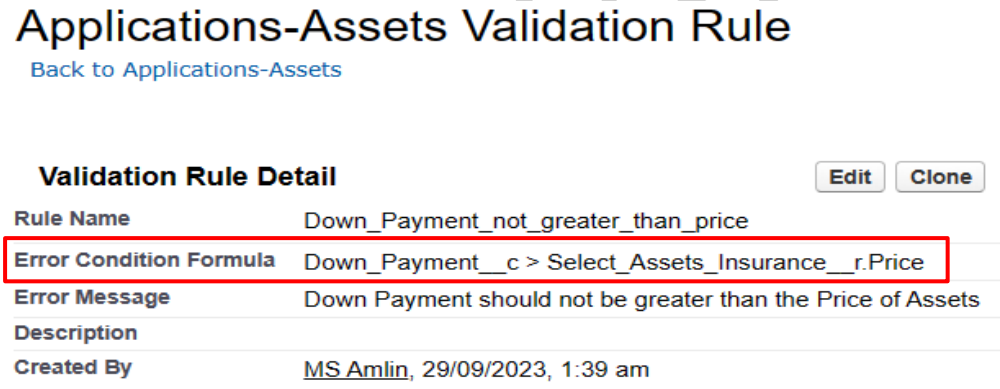
```
trigger AppAssToAssCopyPriceTrigger on Applications_Assets__c (before insert,before update){
    SET<id> assIdSet = new SET<id>();
    if(trigger.isInsert || trigger.isUpdate){
        for(Applications_Assets__c objAppAss : trigger.new){
            if(objAppAss.Select_Assets_Insurance__c!=null){
                assIdSet.add(objAppAss.Select_Assets_Insurance__c);
            }
            if(trigger.isUpdate){
                if(objAppAss.Select_Assets_Insurance__c!=trigger.oldMap.get(objAppAss.Id).Select_Assets_Insurance__c){
                    assIdSet.add(objAppAss.Select_Assets_Insurance__c);
                }
            }
        }
    }
    Map<id,Asset> assMap = new Map<id,Asset>();
    if(!assIdSet.isEmpty()){
        for(Asset objAss :[select id, Name, Price from Asset where ID IN : assIdSet]){
            assMap.put(objAss.id, objAss);
        }
    }
    if(trigger.isInsert || trigger.isUpdate){
        for(Applications_Assets__c objAppAss : trigger.new){
            if(!assMap.isEmpty()){
                if(assMap.containsKey(objAppAss.Select_Assets_Insurance__c)){
                    objAppAss.Price__c = assMap.get(objAppAss.Select_Assets_Insurance__c).Price;
                }
            }
        }
    }
}
```


```
trigger AppAssToAssCopyPriceTrigger on Applications_Assets__c (before insert,before update){
    SET<id> assIdSet = new SET<id>();
    if(trigger.isInsert || trigger.isUpdate){
        for(Applications_Assets__c objAppAss : trigger.new){
            if(objAppAss.Select_Assets_Insurance__c!=null){
                assIdSet.add(objAppAss.Select_Assets_Insurance__c);
            }
            if(trigger.isUpdate){
                if(objAppAss.Select_Assets_Insurance__c!=trigger.oldMap.get(objAppAss.Id).Select_Assets_Insurance__c){
                    assIdSet.add(objAppAss.Select_Assets_Insurance__c);
                }
            }
        }
    }
    Map<id,Asset> assMap = new Map<id,Asset>();
    if(!assIdSet.isEmpty()){
        for(Asset objAss :[select id, Name, Price from Asset where ID IN : assIdSet]){
            assMap.put(objAss.id, objAss);
        }
```

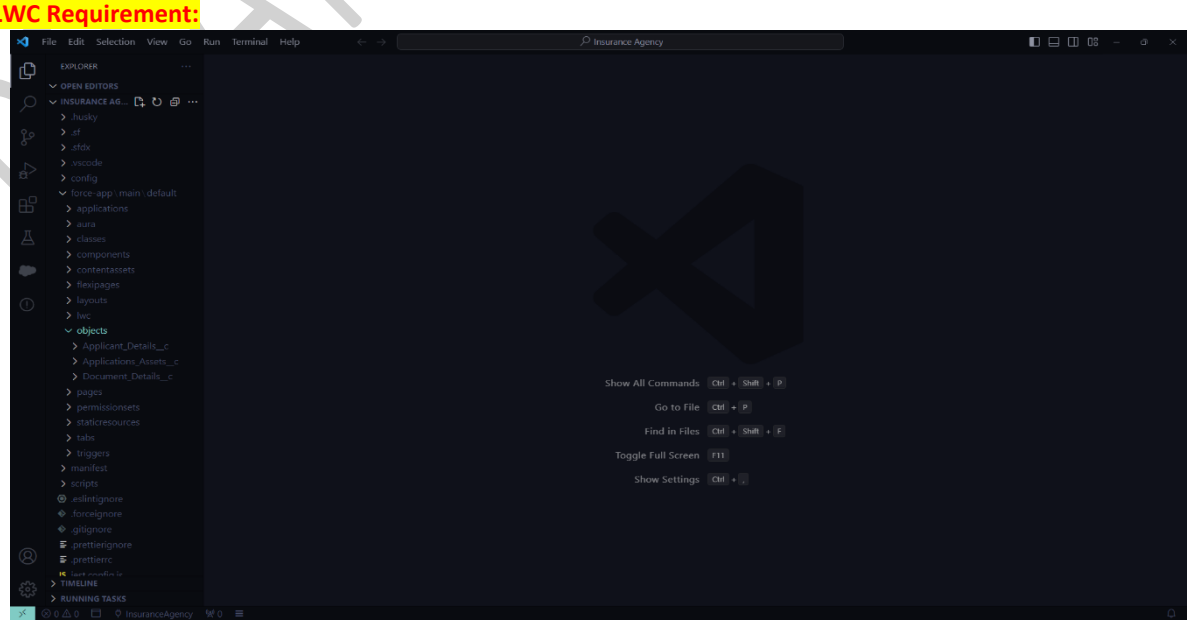
```
    }
    if(trigger.isInsert || trigger.isUpdate){
      for(Applications_Assets__c objAppAss : trigger.new){
        if(!assMap.isEmpty()){
          if(assMap.containsKey(objAppAss.Select_Assets_Insurance__c)){
            objAppAss.Price__c = assMap.get(objAppAss.Select_Assets_Insurance__c).Price;
          }
        }
      }
    }
}
```

D] Down Payment Should not greater than Price of Assets.



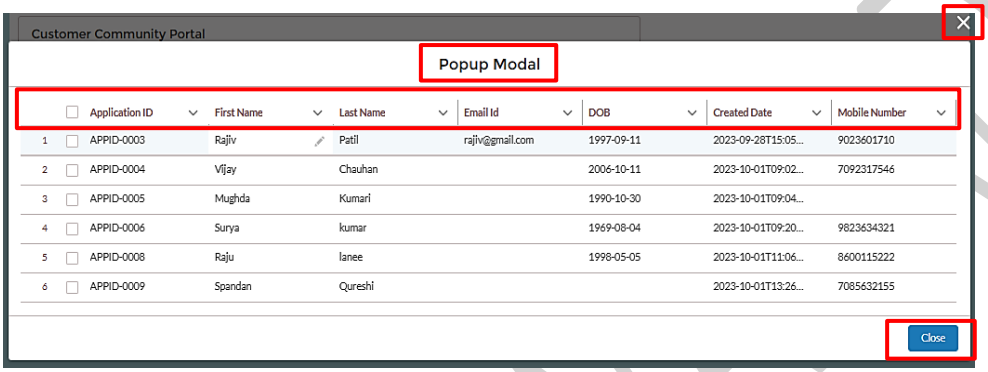


7. **LWC Requirement:**

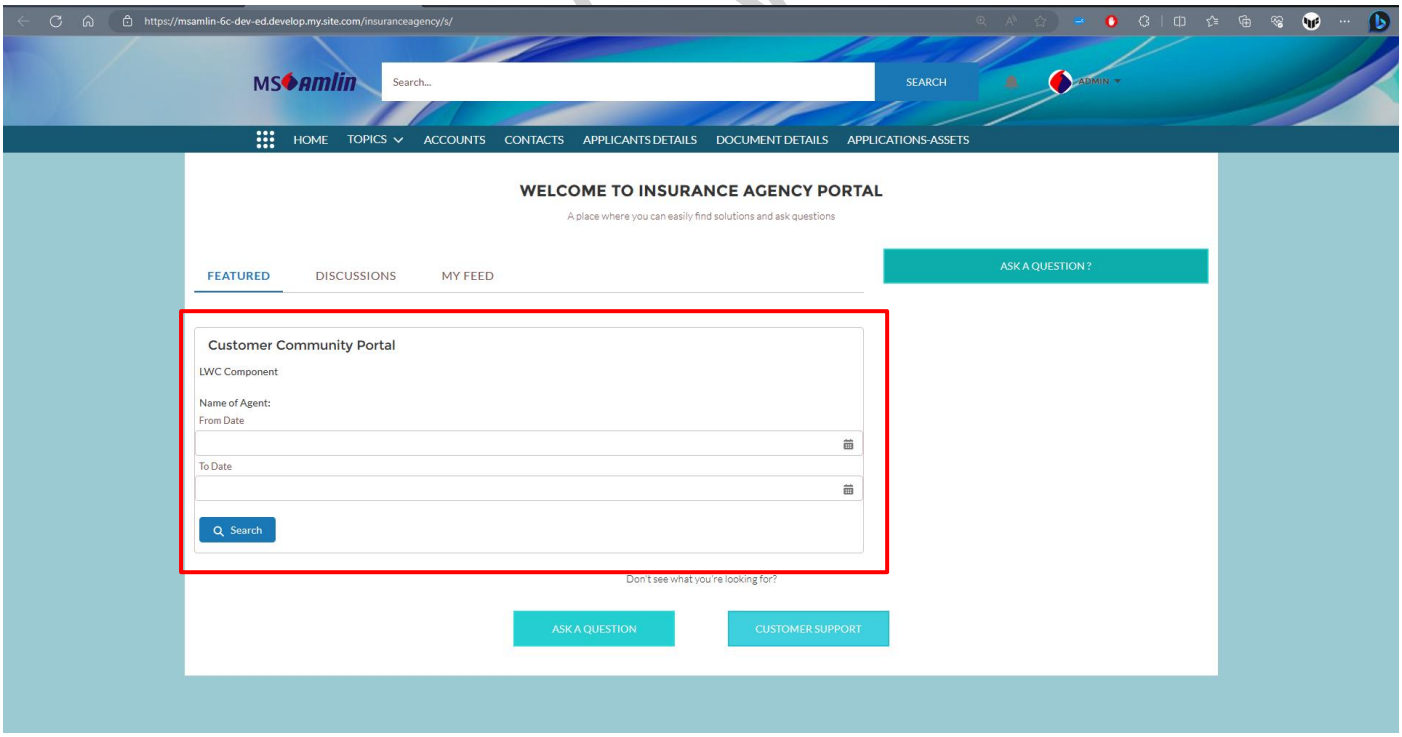


LWC Component:

Popup Modal:



Community Portal:



LWC Code for Popup-Modal: -

HTML ➔

```
<template>
    <lightning-card title="Customer Community Portal">
        <p>  LWC Component</p>
        <br />
          Name of Agent:
        <br />
        <lightning-record-edit-form object-api-name="Applicant_Details__c">
            <lightning-input type="Date" label="  From Date" value={start_date} name="startdate"
                onchange={startDateHandler}></lightning-input>
            <lightning-input type="Date" label="  To Date" value={end_date} name="enddate"
                onchange={endDateHandler}></lightning-input>
            <br />
              <lightning-button variant="brand" label="Search" icon-name="utility:search"
                onclick={buttonHandler}></lightning-button>


            <template lwc:if={showAppFlag}>
                <!--POP UP MODAL STARTS-->
                <section role="dialog" tabindex="-1" aria-modal="true" aria-labelledby="modal-heading-01"
                    class="slds-modal slds-fade-in-open">
                    <div class="slds-modal__container">
                        <button class="slds-button slds-button_icon slds-modal__close slds-button_icon-inverse"
                            title="Close" onclick={closeButtonHandler}>
                            <lightning-icon icon-name="utility:close" size="small" alternative-text="close"
                                variant="inverse"></lightning-icon>
                            <span class="slds-assistive-text">Cancel and close</span>
                        </button>
                        <div class="slds-modal__header">
                            <h1 id="modal-heading-01" class="slds-modal__title slds-hyphenate">Popup Modal</h1>
                        </div>
                        <div class="slds-modal__content slds-p-around_medium" id="modal-content-id-1">

                            <!-- DATA TABLE STARTS -->
                            <lightning-datatable
                            key-field="Id"
                            data={appList}
                            columns={appColumns}
                            draft-values={draftValues}
                            >
                            </lightning-datatable>
                            <!-- DATA TABLE END -->

                        </div>
                        <div class="slds-modal__footer">

                            <button class="slds-button slds-button_brand" onclick={closeButtonHandler}
                                aria-label="Cancel and close">Close</button>
                        </div>
                    </div>
                </section>
                <div class="slds-backdrop slds-backdrop_open"></div>
                <!--POP UP MODAL END-->
            </template>
        </lightning-record-edit-form>
    </lightning-card>
</template>
```

JS ➔

```
import { LightningElement, track } from 'lwc';
import searchAppAssRecords from '@salesforce/apex/ApplicationProvider.searchAppAssRecords';

export default class ApplicationPopupModal extends LightningElement {

    objApp = { 'sObjectType': 'Applicant_Details__c' };
    appList;
    start_date;
    end_date;

    showAppFlag = false;
    @track start_date;
    @track end_date;
```

```
    draftValues = [];

    startDateHandler(event) {
        console.log(event.target.value);
        this.start_date = event.target.value;
    }

    endDateHandler(event) {
        console.log(event.target.value);
        this.end_date = event.target.value;
    }

    appColumns = [
        { label: 'Application ID', fieldName: 'Name', editable: true },
        { label: 'First Name', fieldName: 'First_Name__c', editable: true },
        { label: 'Last Name', fieldName: 'Last_Name__c', editable: true },
        { label: 'Email Id', fieldName: 'Email_Id__c', editable: true },
        { label: 'DOB', fieldName: 'Date_of_Birth__c', editable: true },
        { label: 'Created Date', fieldName: 'CreatedDate', editable: true },
        { label: 'Mobile Number', fieldName: 'Mobile_Number__c', editable: true },
    ];

    closeButtonHandler(){
        this.showAppFlag = false;
    }

    buttonHandler() {
        searchAppAssRecords({ 'startDate': this.start_date , 'endDate' :  this.end_date })
            .then(result => {
                console.log(result);
                console.log(this.appList);
                this.appList = result;
                this.showAppFlag = true;
            })
            .catch(error => {
                console.log(error);
                this.showAppFlag = false;
            })
    }
}
```

CSS file ➔

```
.slds-modal__container {
    max-width: 80rem !important;
    width: 80% !important;
}
```

Meta-xml ➔

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>58.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightningCommunity__Page</target>
    </targets>
</LightningComponentBundle>
```

Apex Controller ➔

```
public with sharing class ApplicationProvider {

    @AuraEnabled
    public static List<Applicant_Details__c> searchAppAssRecords(Date startDate, Date endDate){
        System.debug('# Received Dates '+startDate+','+endDate);
        try{
            return [select id, Name, First_Name__c, Last_Name__c, Email_Id__c, Mobile_Number__c, Date_of_Birth__c, CreatedDate
from Applicant_Details__c where CreatedDate >=:startDate and CreatedDate <=:endDate];
        }
        catch(Exception e) {
```

```
            return null;
        }
    }
}
```

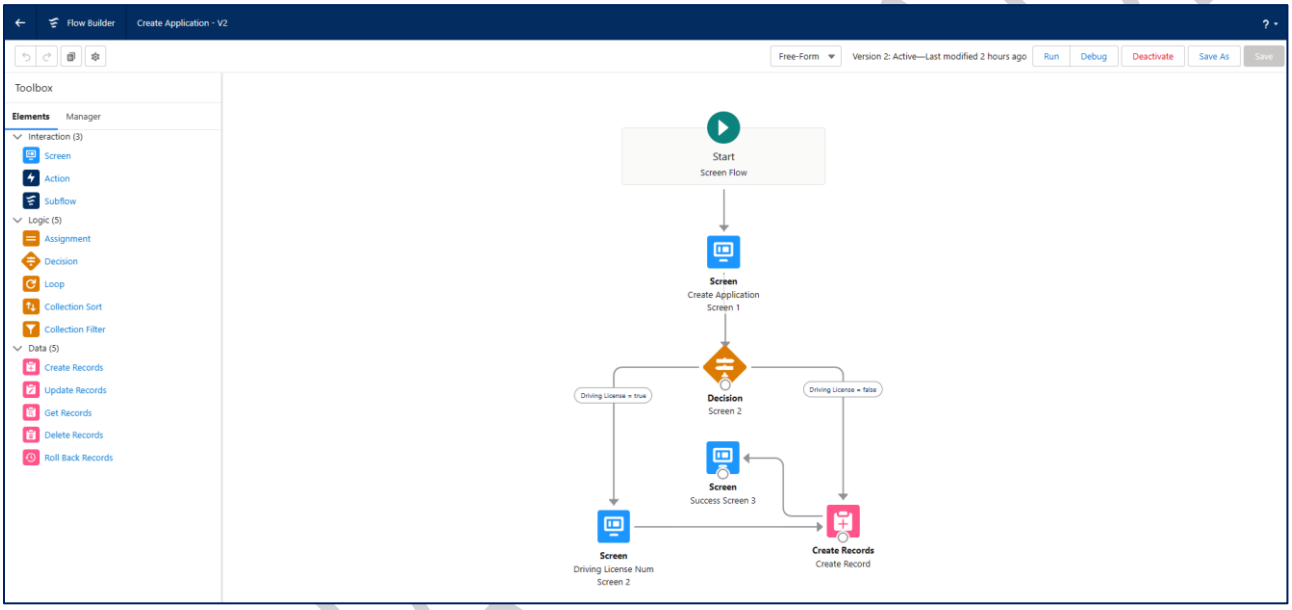
LWC Pop-up modal.mp4

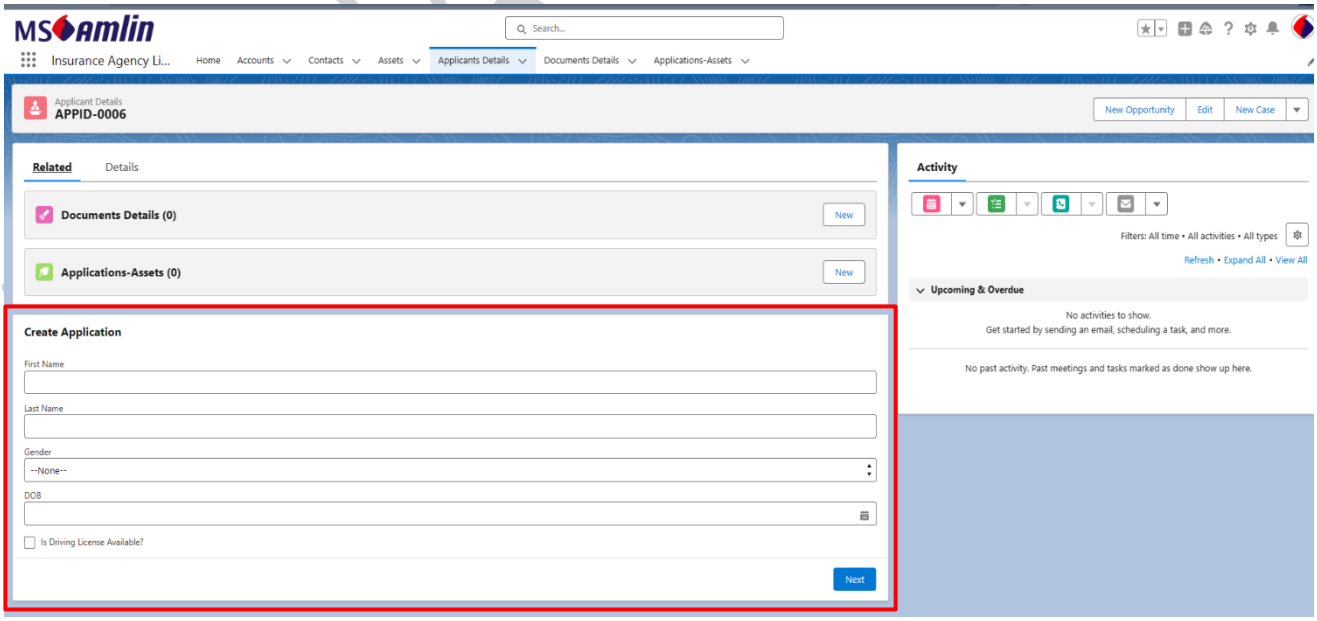
This is a video to see the output of LWC Component of Popup-Modal. (*Double Click on icon*)

8. Flow Builder:

JIRA TICKET: MSAMLIN-101

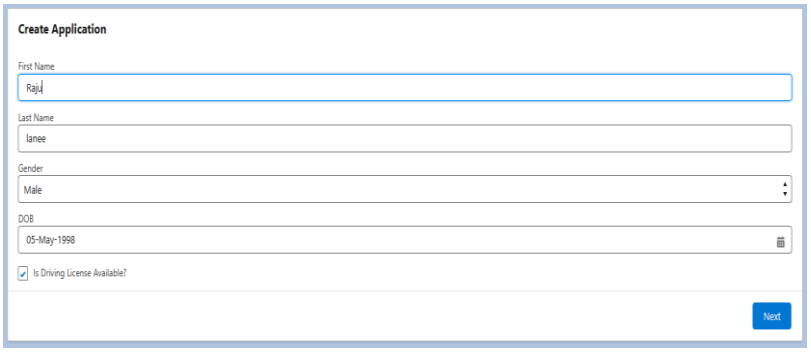
Create Application Record.



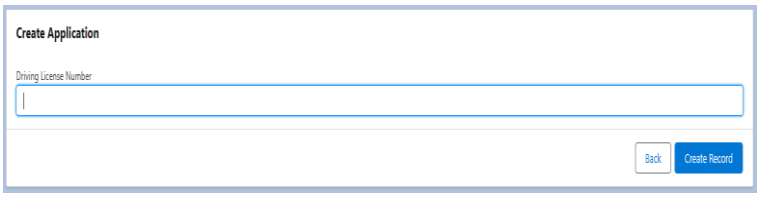


Requirements:

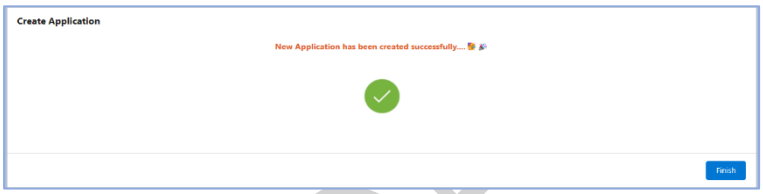
If driving license is available in screen 1 the only show screen 2.



Screen 1



Screen 2



Screen 3

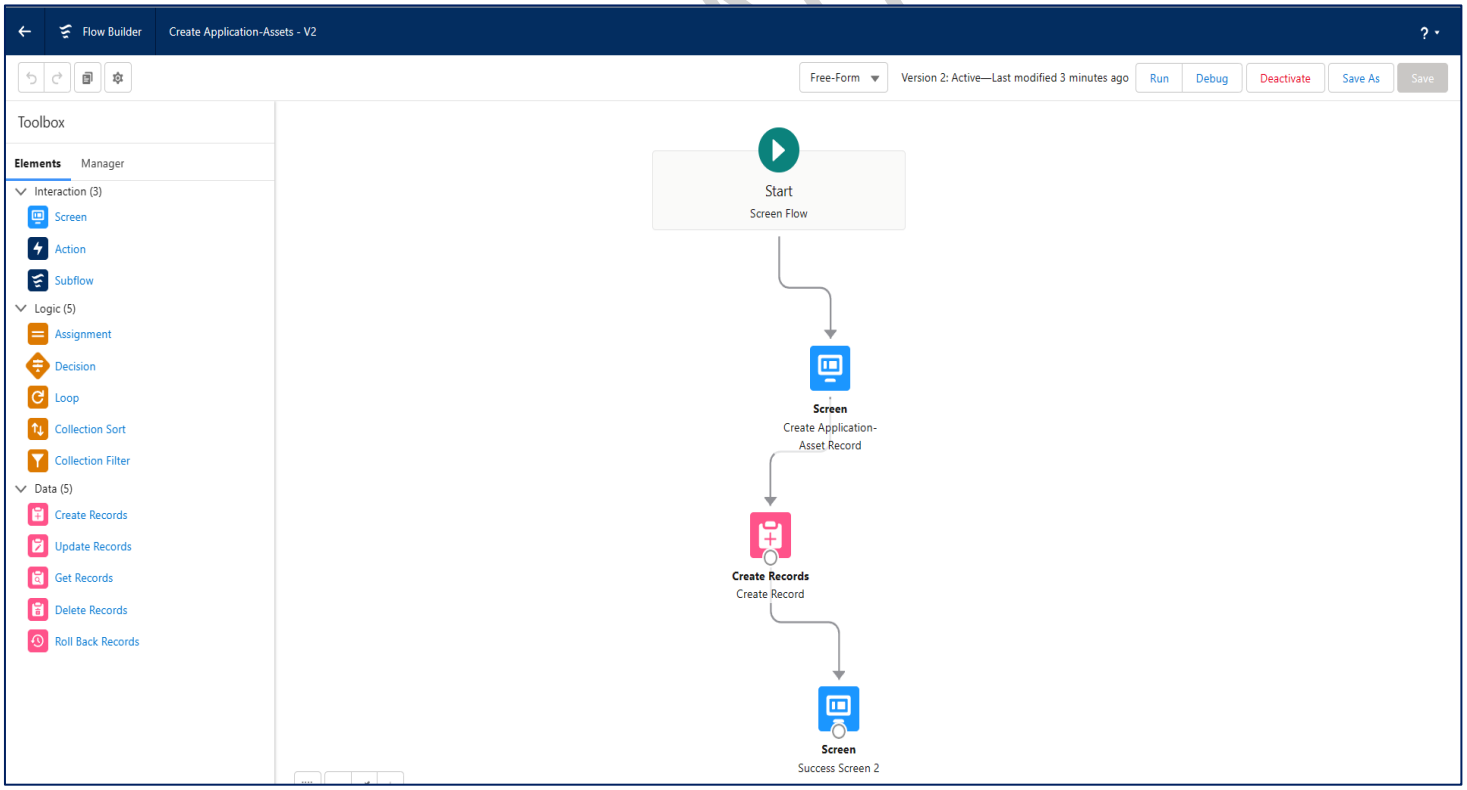
Screen flow
JIRA-101.mp4

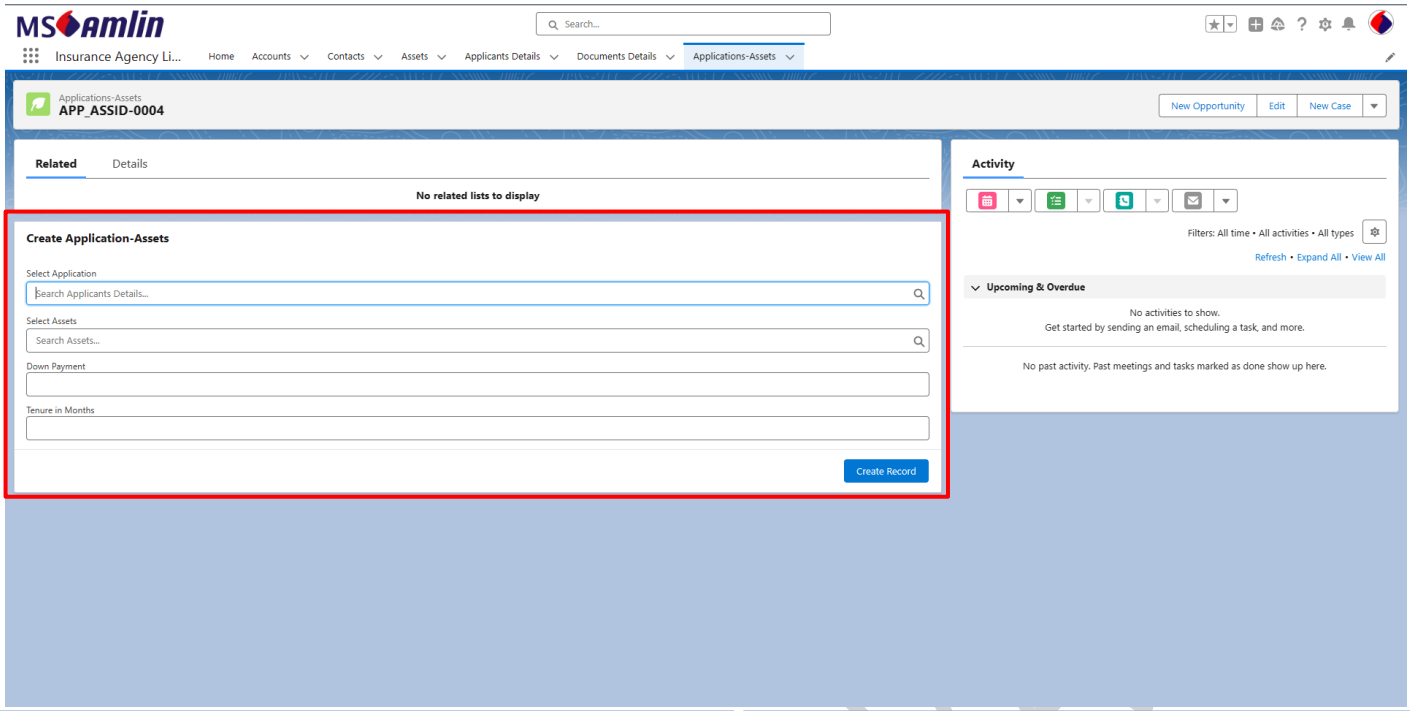
This is a video to see the output of Application Record flow. (*Double Click on icon*)

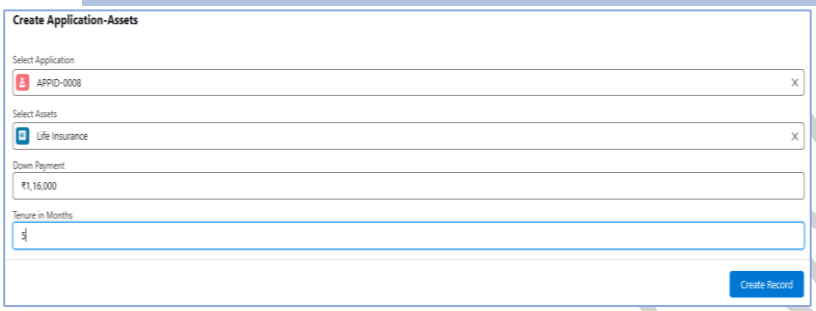
==========

JIRA TICKET: MSAMLIN-102

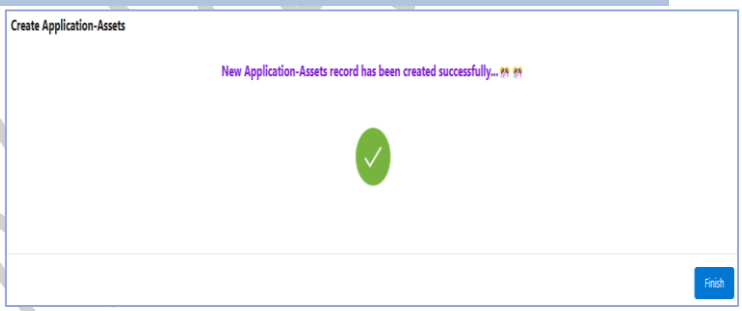
Create Application-Assets Record







Screen 1



Screen 2

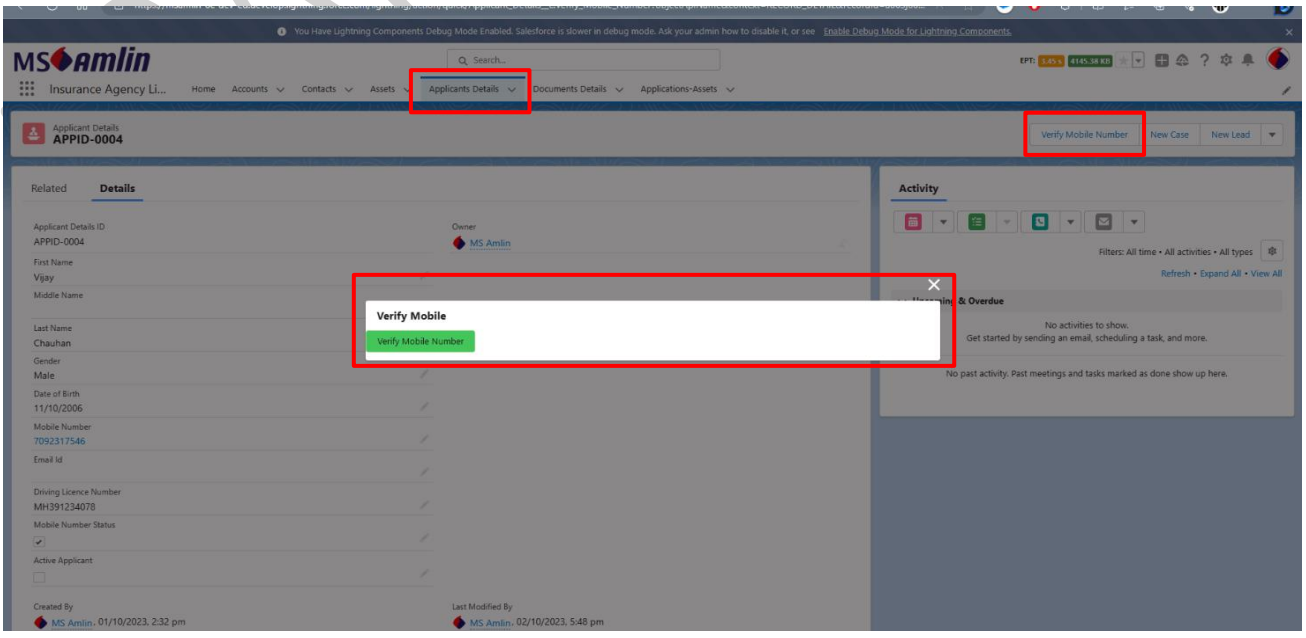
Screen Flow
JIRA-102.mp4

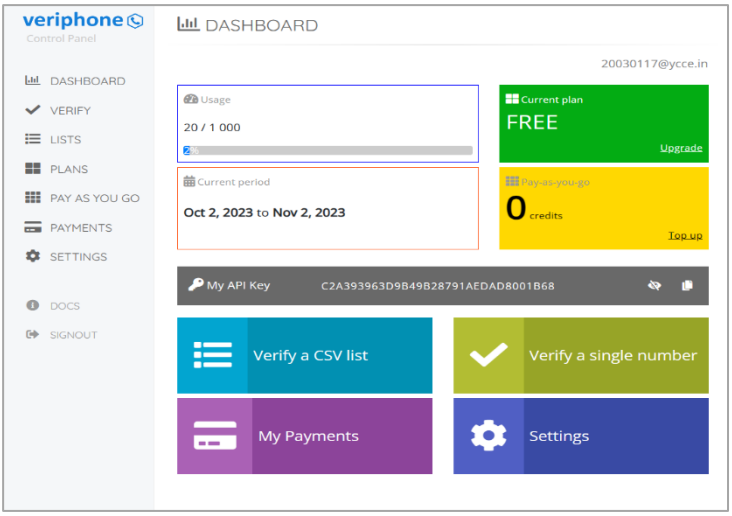
This is a video to see the output of Application-Assets Record flow. (*Double Click on icon*)

---

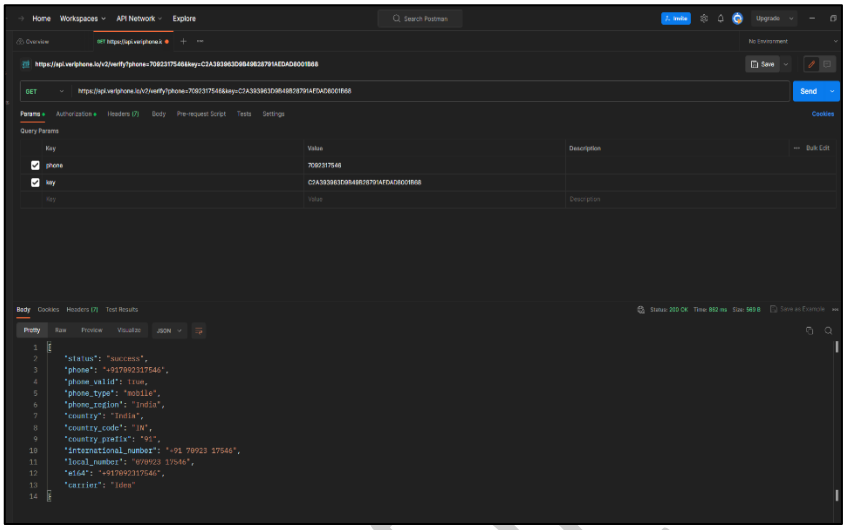
9. Integration Requirement:

Mobile Number verification in Application Details Object by Veriphone.





Veriphone.io



Postman

LWC Code for Mobile Number Verification: -

HTML ➔

```
<template>
    <lightning-card title="Verify Mobile">
        <lightning-button variant="success" label="Verify Mobile Number"
onclick={verifyMobHandler}></lightning-button>
    </lightning-card>
</template>
```

JavaScript ➔

```
import { LightningElement, api } from 'lwc';
import { CloseActionScreenEvent } from 'lightning/actions';
import verifyMob from '@salesforce/apex/ApplicationRequestAPI.verifyMob';

export default class ApplicationMobileVerify extends LightningElement {
    @api recordId;

    verifyMobHandler() {
        this.dispatchEvent(new CloseActionScreenEvent());
        window.location.reload();
        console.log(this.recordId);

        verifyMob({'appliRecordId' : this.recordId})
            .this(result=>{
                console.log(result);
                this.dispatchEvent(new CloseActionScreenEvent());

            })
            .catch(error=>{
                console.log(error);
                this.dispatchEvent(new CloseActionScreenEvent());
            })
    }
}
```

XML file ➔

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>58.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__RecordAction</target>
    </targets>
</LightningComponentBundle>
```

Request API class ➔

```
public with sharing class ApplicationRequestAPI {

    @AuraEnabled
    public static void verifyMob(ID appliRecordId){
        System.debug('Recevied Application Id '+appliRecordId);

        String mobileNumber = [select id, Mobile_Number__c from Applicant_Details__c where ID =:
appliRecordId].Mobile_Number__c;

        if(String.isNotBlank(mobileNumber)){
            System.debug('mobile '+mobileNumber);

            Http http = new Http();
            HttpRequest request = new HttpRequest();
            request.setEndpoint('https://api.veriphone.io/v2/verify?phone='+mobileNumber+'&key=C2A393963D9B49B28791AEDAD8001B6
8');
            request.setMethod('GET');

            HttpResponse response = http.send(request);
                System.debug('%%Status_Code = '+response.getStatusCode());

                if(response.getStatusCode() == 200){
                    System.debug('@@received body = '+response.getBody());

                    ApplicationRequestAPIWrapper objWrap = (ApplicationRequestAPIWrapper) JSON.deserialize(response.getBody(),
ApplicationRequestAPIWrapper.class);

                    Applicant_Details__c objAppl = new Applicant_Details__c(Id=appliRecordId);

                        if(objWrap.phone_valid){
                            objAppl.Mobile_Number_Status__c = true;
                        }
                        else{
                            objAppl.Mobile_Number_Status__c = false;
                        }

                        update objAppl;
                    }
                }
            }

            public class ApplicationRequestAPIWrapper{
                public Boolean phone_valid;
            }
        }
```

LWC mobile verify.mp4

This is a video to see the output of Application Detail Record flow. (*Double Click on icon*)

11. Batch Class Requirement:

- Create a Batch class to update all the Applications status Checkbox to Inactive

ApplicationStatusBatchClass.apxc | Applicant_Details__c@12:47 AM

```
select id, Name, First_Name__c, Last_Name__c, CreatedDate, Active_Applicant__c, Asset_Products__c from Applicant_Details__c where CreatedDate = LAST_MONTH
```

Query Results - Total Rows: 1

Id	Name	First_Name__c	Last_Name__c	CreatedDate	Active_Applicant__c	Asset_Products__c
a005j00000P5RYBAA3	APPID-0003	Rajiv	Patil	2023-09-28T15:05:41.00...	true	

Updated

ApplicationStatusBatchClass.apxc | Log executeAnonymous @3/10/2023, 12:50:21 am | Applicant_Details__c@12:50 AM

```
select id, Name, First_Name__c, Last_Name__c, CreatedDate, Active_Applicant__c, Asset_Products__c from Applicant_Details__c where CreatedDate = LAST_MONTH
```

Query Results - Total Rows: 1

Id	Name	First_Name__c	Last_Name__c	CreatedDate	Active_Applicant__c	Asset_Products__c
a005j00000P5RYBAA3	APPID-0003	Rajiv	Patil	2023-09-28T15:05:41.00...	false	

Apex Jobs

Help for this Page

Monitor the status of all Apex jobs, and optionally, abort jobs that are in progress.

View: All | Create New View

Action	Submitted Date	Job Type	Status	Status Detail	Total Batches	Batches Processed	Failures	Submitted By	Completion Date	Apex Class	Apex Method	Apex Job ID
	03/10/2023, 12:50 am	Batch Apex	Completed		1	1	0	Amlin, MS	03/10/2023, 12:50 am	ApplicationStatusBatchClass		7075j00005Ubi5E

ApplicationStatusBatchClass.apxc

```
global class ApplicationStatusBatchClass implements Database.Batchable<sObject>{

    global Database.QueryLocator start(Database.BatchableContext BC){
        return Database.getQueryLocator('select id, Name, First_Name__c, Last_Name__c, CreatedDate, Active_Applicant__c, Asset_Products__c from Applicant_Details__c where CreatedDate = LAST_MONTH and Active_Applicant__c = true');
    }
    global void execute(Database.BatchableContext BC, List<Applicant_Details__c> appList){
        if(!appList.isEmpty()){
            for(Applicant_Details__c objApp : appList){
                if(String.isBlank(objApp.Asset_Products__c)){
                    objApp.Active_Applicant__c = false;
                }
                Database.update(appList, false);
            }
        }
    }
    global void finish(Database.BatchableContext BC){

    }
}
```

Enter Apex Code

```
Database.executeBatch(new ApplicationStatusBatchClass(), 50);
```

Open Log | Execute | Execute Highlighted

- Run above batch class day 1 on every month.

ApplicationStatusBatchClass.apxc | ScheduleApplicationStatusBatchClass.apxc

Code Coverage: None | API Version: 58

```
public class ScheduleApplicationStatusBatchClass implements Schedulable{
    public void execute(SchedulableContext SC){
        Database.executeBatch(new ApplicationStatusBatchClass(), 50);
    }
}
```

Schedule Apex

Schedule an Apex class that implements the 'Schedulable' interface to be automatically executed on a weekly or monthly interval.

Cancel | Delete

Job Name: ScheduleApplicationStatusB

Apex Class: ScheduleApplicationStatusBatchClass

Schedule Apex Execution

Frequency: Weekly / Monthly — On day 1 of every month / On the 1st Sunday of every month

Start: 03/10/2023 [03/10/2023]

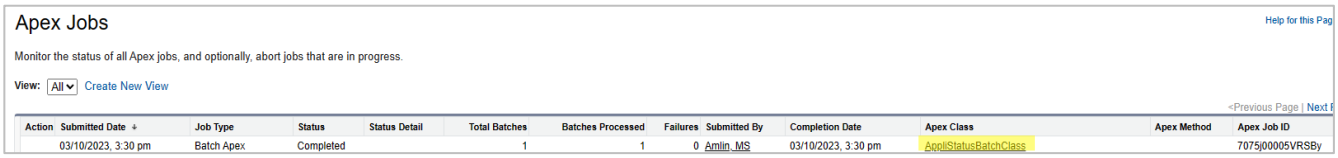
End: 03/11/2023 [03/10/2023]

Preferred Start Time: 8:00 am

Exact start time will depend on job queue activity.

Cancel | Delete

- Send success and failed record id.



Code➔

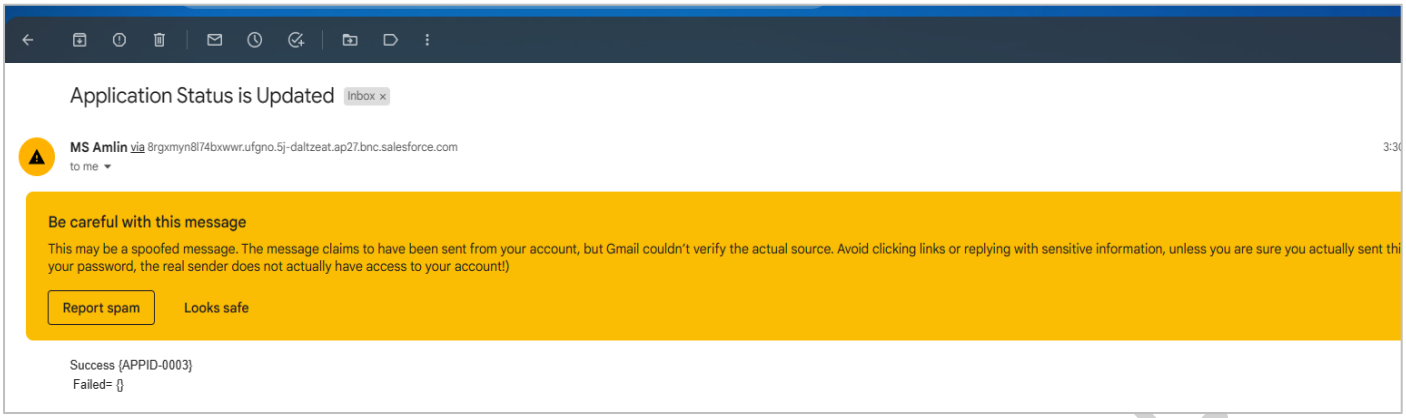
```
global class AppliStatusBatchClass implements Database.Batchable<SObject>, Database.Stateful{
    SET<String> successIdSet = new SET<String>();
    SET<String> failedIdSet = new SET<String>();
    global Database.QueryLocator start(Database.BatchableContext BC){
        return Database.getQueryLocator('select id, Name, First_Name__c, Last_Name__c, CreatedDate,
Active_Applicant__c, Asset_Products__c from Applicant_Details__c where CreatedDate = LAST_MONTH and
Active_Applicant__c = true');
    }

    global void execute(Database.BatchableContext BC, List<Applicant_Details__c> appList){
        if(!appList.isEmpty()){
            for(Applicant_Details__c objApp : appList){
                if(String.isBlank(objApp.Asset_Products__c)){
                    objApp.Active_Applicant__c = false;
                }
                Database.SaveResult[] result =  Database.update(appList, false);
                Integer index=0;

                for(Database.SaveResult sr: result){
                    if(sr.isSuccess()){
                        System.debug('Succes = '+appList[index].Name);
                        successIdSet.add(appList[index].Name);
                        index++;
                    }
                    else{
                        System.debug('Failed = '+appList[index].Name);
                        failedIdSet.add(appList[index].Name);
                        index++;
                    }
                    System.debug('Success= '+successIdSet);
                    System.debug('Failed = '+failedIdSet);
                }
            }
        }
    }
    global void finish(Database.BatchableContext BC){
        Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
        mail.setToAddresses(new String[] {'waradepranav99@gmail.com'});
        mail.setSenderDisplayName('MS Amlin');
        mail.setSubject('Application Status is Updated');
        mail.setPlainTextBody('Success '+successIdSet+'\n Failed= '+failedIdSet);
        Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail });

    }
}
```

- Email send System Admin.



- Write Test Class.

Batch Class➔

```
global class BatchClassAppStatus implements Database.Batchable<sObject>{

    global Database.QueryLocator start(Database.BatchableContext BC){
        return Database.getQueryLocator('select id, First_Name__c, Last_Name__c,Asset_Products__c,
Active_Applicant__c from Applicant_Details__c where Active_Applicant__c = true');
    }
    global void execute(Database.BatchableContext BC, List<Applicant_Details__c> appList){
        if(!appList.isEmpty()){
            for(Applicant_Details__c objApp : appList){
                if(String.isBlank(objApp.Asset_Products__c)){
                    objApp.Active_Applicant__c = false;
                }
            }
            Database.update(appList, false);
        }
    }
    global void finish(Database.BatchableContext BC){

    }
}
```

Test Class➔

```
@isTest(seeAllData=false)
public class TestClassOfAppStatus{

    @isTest
    public static void testAppBatchMethod(){

        List<Applicant_Details__c> appList = new List<Applicant_Details__c>();
        for(Integer i=1; i<=200; i++){
            Applicant_Details__c objApp = new Applicant_Details__c(First_Name__c='Shankar'+i,
Last_Name__c='Pratap'+i, Asset_Products__c= null, Active_Applicant__c = true);
            appList.add(objApp);
        }
        if(!appList.isEmpty()){
            Database.insert(appList, false);
        }
        Test.startTest();
            Database.executeBatch(new BatchClassAppStatus(), 200);
        Test.stopTest();

        List<Applicant_Details__c> actualAppList =[select id, First_Name__c, Last_Name__c, Active_Applicant__c,
Asset_Products__c from Applicant_Details__c where Active_Applicant__c=false];
        System.assertEquals(200, actualAppList.size());
```

```
    }
}
```

Apex Test Execution

Click Select Tests to choose one or more Apex unit tests and run them. To see the current code coverage for an individual class or your organization, go to the Apex Classes page.

Select Tests... | Developer Console | Options... | View Test History

Abort

Status	Class	Result
Test Run: 2023-10-03 23:08:34, admin@msamlin.co, (1 test class run)		
✔	[View] TestClassOfAppStatus	(1/1) Test Methods Passed

```
=== Test Summary
NAME                 VALUE
___________________  ____________________
Outcome              Passed
Tests Ran            1
Pass Rate            100%
Fail Rate            0%
Skip Rate            0%
Test Run Id          7075j00005VS6DM
Test Execution Time  1280 ms
Org Id               00D5j00000DAltZEAT
Username             admin@msamlin.co


=== Test Results
TEST NAME                                OUTCOME  MESSAGE  RUNTIME (MS)
_______________________________________  _______  _______  ____________
TestClassOfAppStatus.testAppBatchMethod  Pass              1280

23:08:40.267 ended SFDX: Run Apex Tests
```

Vs Code Output and Coverage

File Edit Debug Test Workspace Help

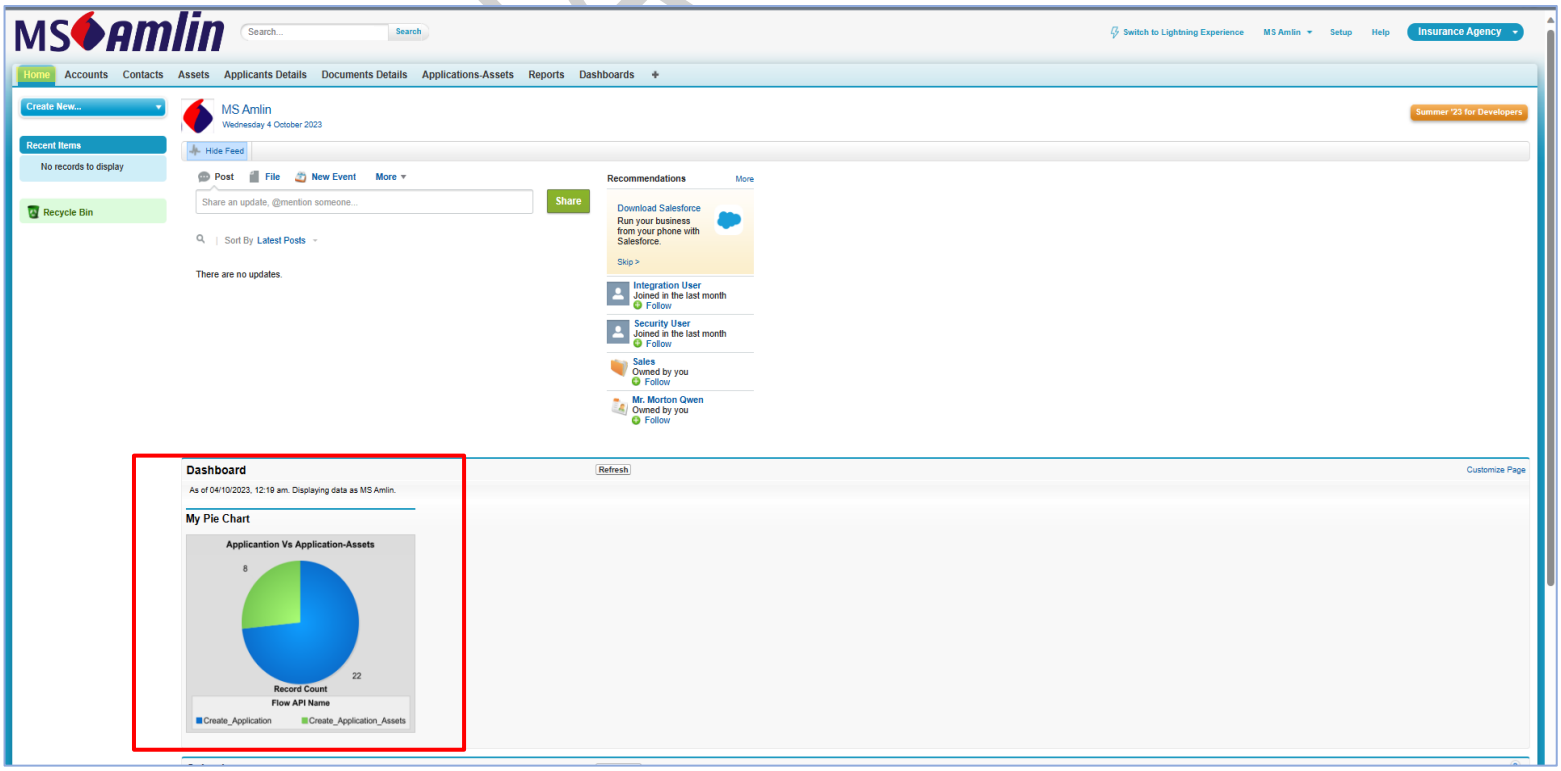
BatchClassAppStatus.apxc | TestClassOfAppStatus.apxc

Code Coverage: All Tests 100% | API Version: 58

None
All Tests 100%
TestClassOfAppStatus.testAppBatchMethod 100%

```
5       }
6       global void execute(Database.BatchableContext BC, L
7           if(!appList.isEmpty()){
8               for(Applicant_Details__c objApp : appList){
9                   if(String.isBlank(objApp.Asset_Products
10                      objApp.Active_Applicant__c = false;
11                  }
12              }
13              Database.update(appList, false);
14          }
15      }
16      global void finish(Database.BatchableContext BC){
17
```

Developer Console Coverage

12. Show Dashboard:

Show Dashboard (Pie Chart) on home page.