CRM Lab 7

Name. Harsha Hargunani
Batch. E1
Roll no. 09
Username. harsha@bts.com

Aim: Demonstrate the implementation of Apex triggers in Salesforce to automate processes and enforce business logic.

Theory:

- Trigger: It is a piece of code which executes automatically before or after an event for an operation. A Trigger is a segment of Apex code which executes before or after inserting or modifying a Salesforce record based on the condition provided.

- Events:
    1. before
    2. after

- Operations:
    1. Insert
    2. Update
    3. Delete
    4. Undelete

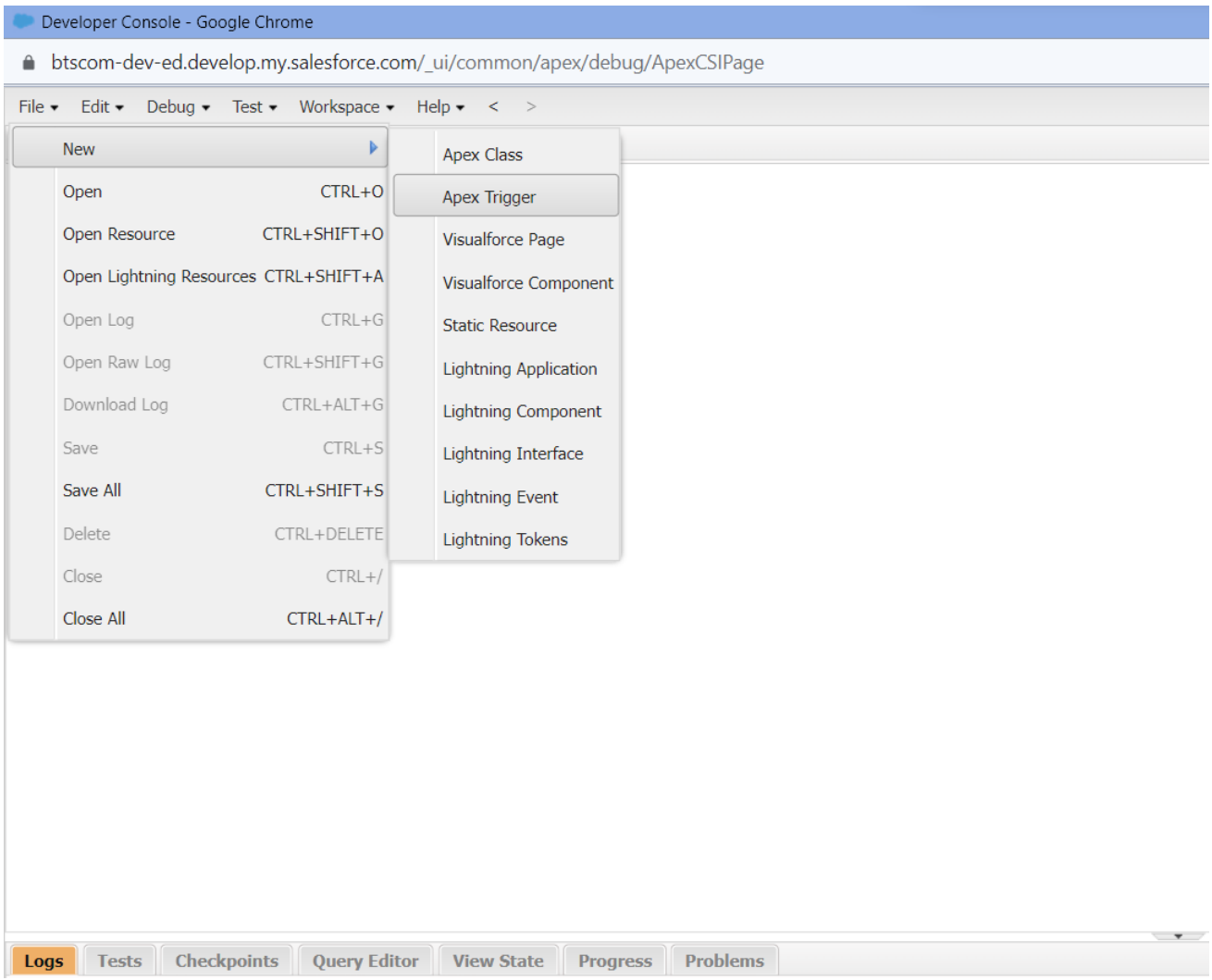
- Context Variables:
    1. trigger.new ➔ It returns a List of new version of the records (List<Object>).
    2. trigger.newMap ➔ It returns a Map of new version of the records (Map<Id,Object>).
    3. trigger.old ➔ It returns a List of old version of the records (List<Object>).
    4. trigger.oldMap ➔ It returns a Map of Old version of the records (Map<Id,Object>)
    5. trigger.isInsert ➔ It return True, if Operation is of Insert.
    6. Trigger.isUpdate ➔ It returns True, if Operation is of Update.
    7. trigger.isDelete ➔ It returns True, if Operation is of Delete.
    8. trigger.isUndelete ➔ It returns True, if Operation is of Undelete.
    9. trigger.isBefore ➔ It returns True, if Event is of Before operation.
    10. trigger.isAfter ➔ It returns True, if Event is of After operation.

Q1. (Task 1) Create / Update an Account Record. If Type is “Prospect”, then the Rating should be “Hot”. Otherwise, none.

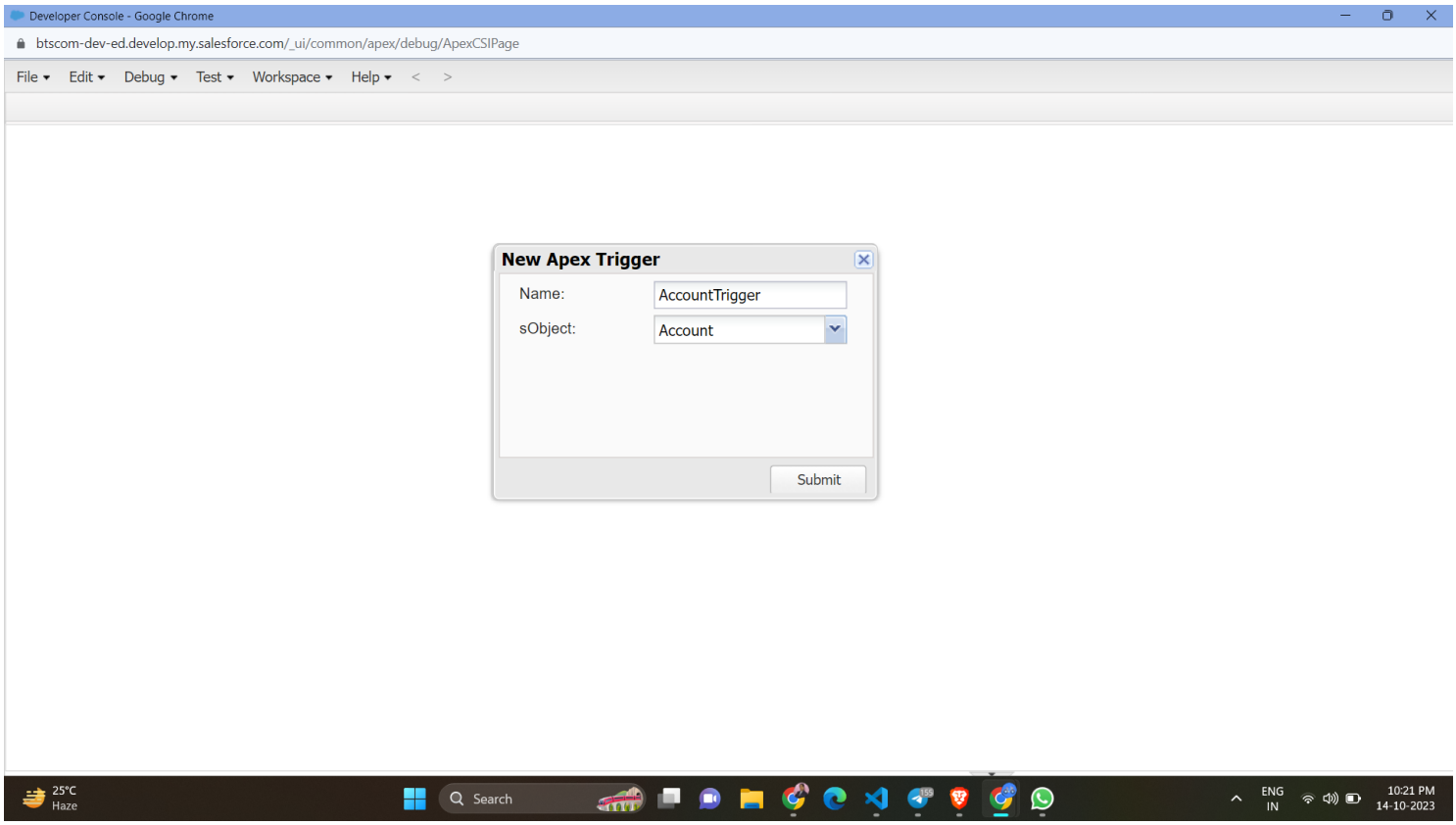
Ans.

EVENT	Before
OPERATIONS	Insert, Update
TRIGGER OBJECT	Account
AFFECTING OBJECT	Account

Step1: Developer Console | File | New | Apex Trigger



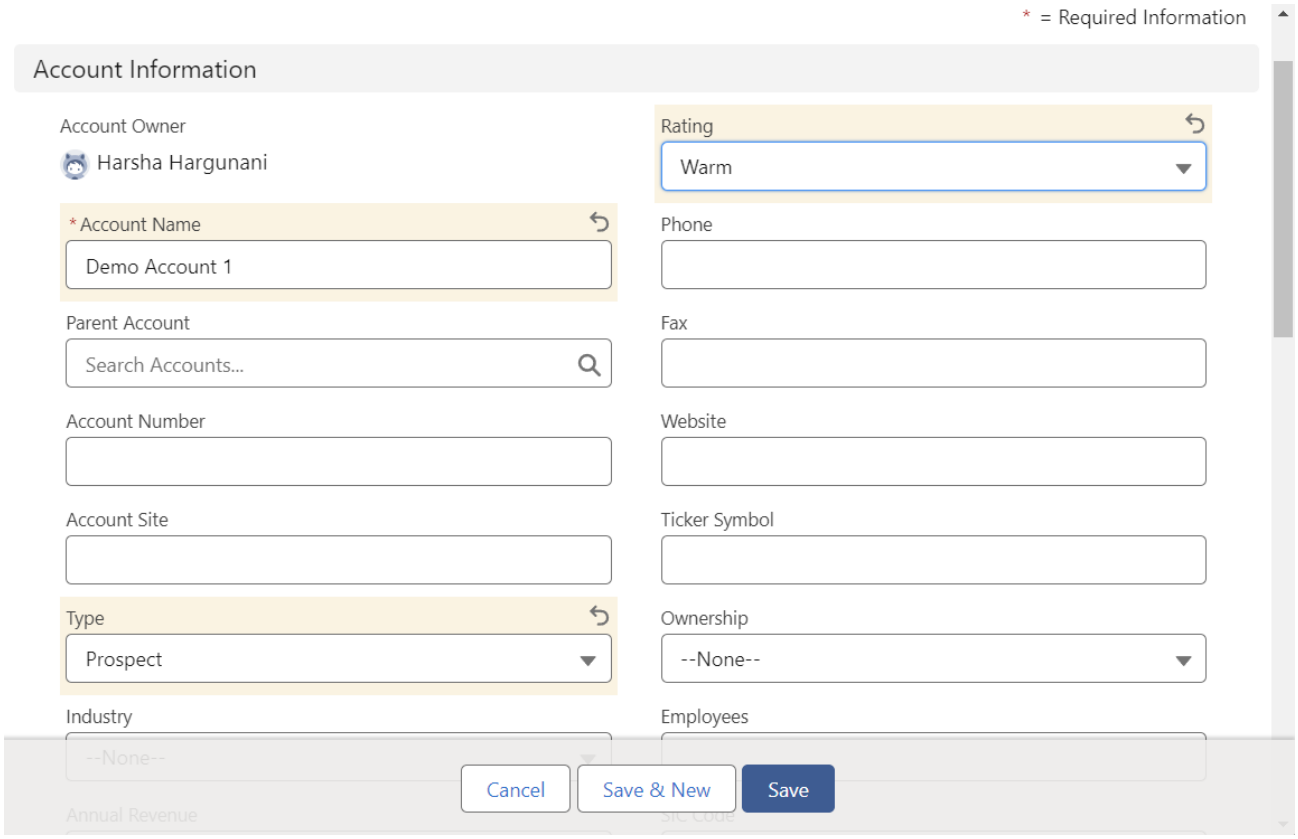
Step2: Enter Trigger name and Object name.

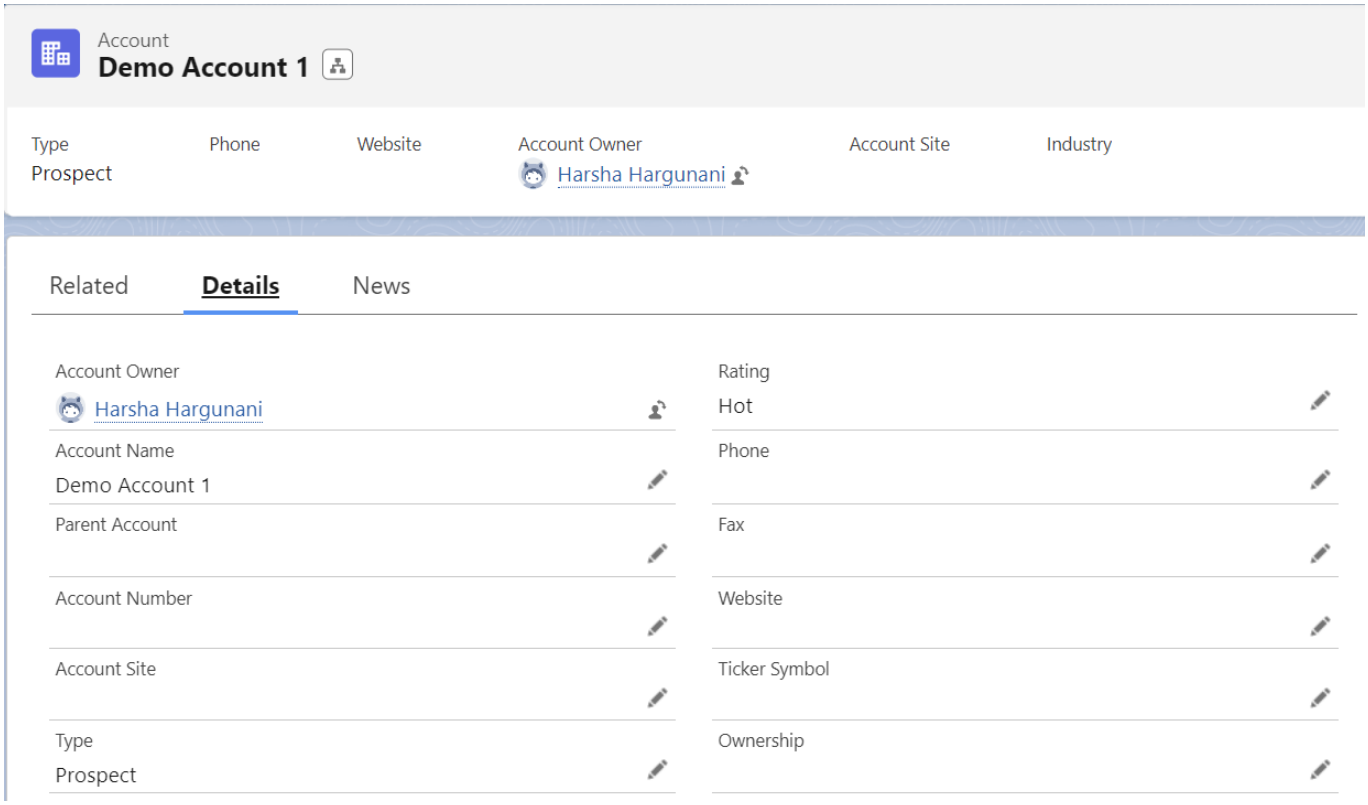


Step3: Enter Apex Code in AccountTrigger.apxt

```
trigger AccountTrigger on Account (before insert, before update) {
    for(Account objAcc : trigger.new){
        if(String.isNotBlank(objAcc.Type)){
            if(objAcc.Type ='Prospect'){
                objAcc.Rating = 'Hot';
            }
            else{
                objAcc.Rating = null;
            }
        }
    }
}
```

Output:


* = Required Information
Account Information
Account Owner
Harsha Hargunani
Rating
Warm
* Account Name
Demo Account 1
Phone
Parent Account
Search Accounts...
Fax
Account Number
Website
Account Site
Ticker Symbol
Type
Prospect
Ownership
--None--
Industry
--None--
Employees
Cancel
Save & New
Save


Account
Demo Account 1
Type
Prospect
Phone
Website
Account Owner
Harsha Hargunani
Account Site
Industry
Related
Details
News
Account Owner
Harsha Hargunani
Rating
Hot
Account Name
Demo Account 1
Phone
Parent Account
Fax
Account Number
Website
Account Site
Ticker Symbol
Type
Prospect
Ownership

Q2. (Task2) Create / Update Account Record.

A. If Type is “Prospect” and Rating is “Hot”, then SLA should be “Gold”.
B. If Type is “Customer - Direct” and Rating is “Cold”, then SLA should be “Silver”.
C. SLA should be none, otherwise.

Ans.

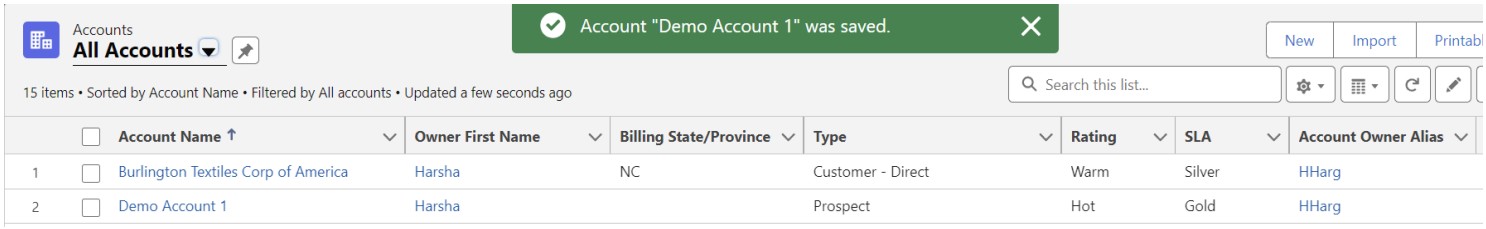
EVENT	Before
OPERATIONS	Insert, Update
TRIGGER OBJECT	Account
AFFECTING OBJECT	Account

AccountTrigger.apxt

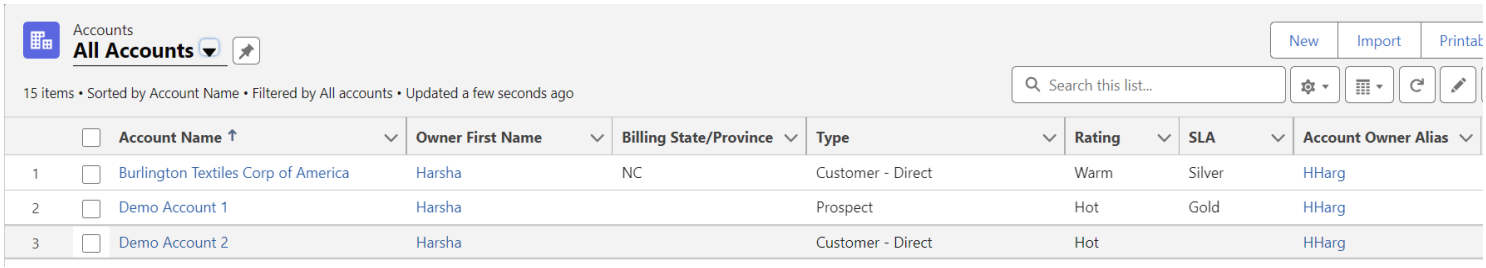
```
trigger AccountTrigger on Account (before insert, before update) {
    for(Account objAcc : trigger.new){
        if(String.isNotBlank(objAcc.Type) && String.isNotBlank(objAcc.Rating)){
            if(objAcc.Type ='Prospect' && objAcc.Rating ='Hot'){
                objAcc.SLA__c = 'Gold';
            }
            else{
                if(objAcc.Type == 'Customer - Direct' && objAcc.Rating ='Cold'){
                    objAcc.SLA__c = 'Silver';
                }
                else{
                    objAcc.SLA__c = null;
                }
            }
        }
    }
}
```

Output: Before-

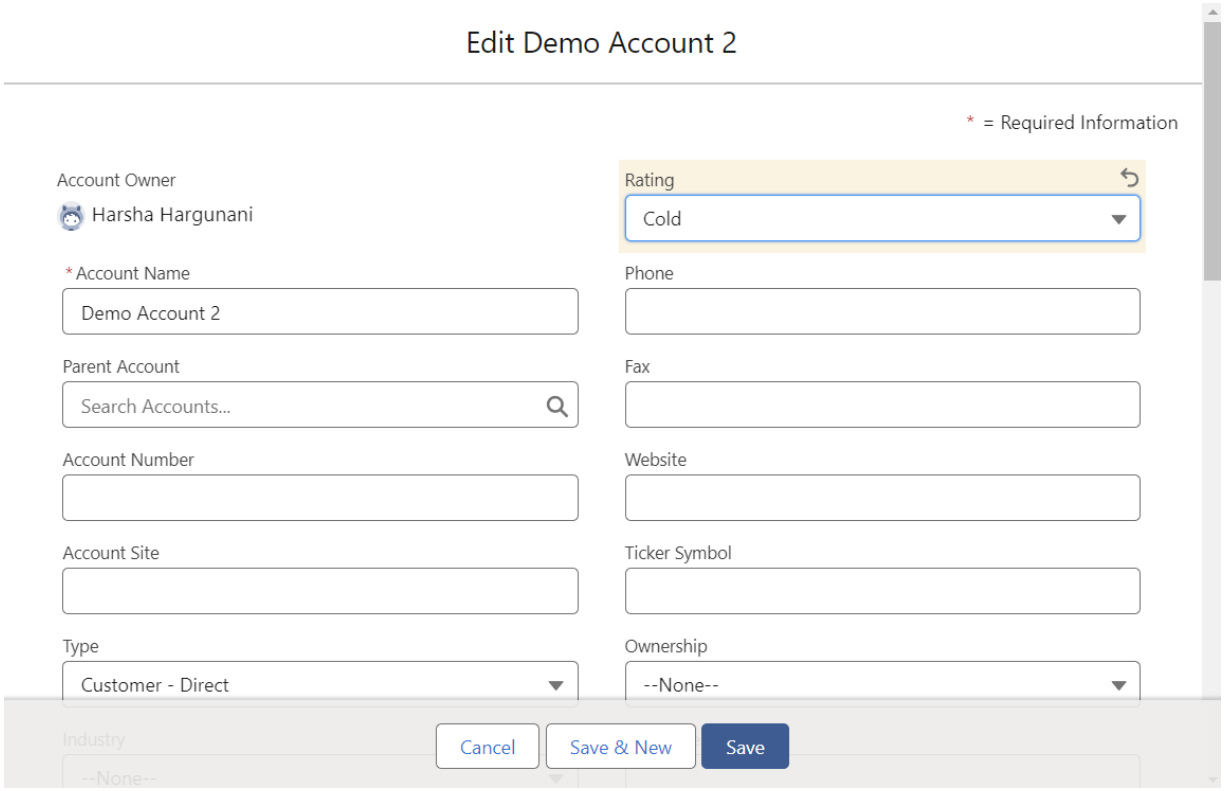
After Edit and Save without any changes-

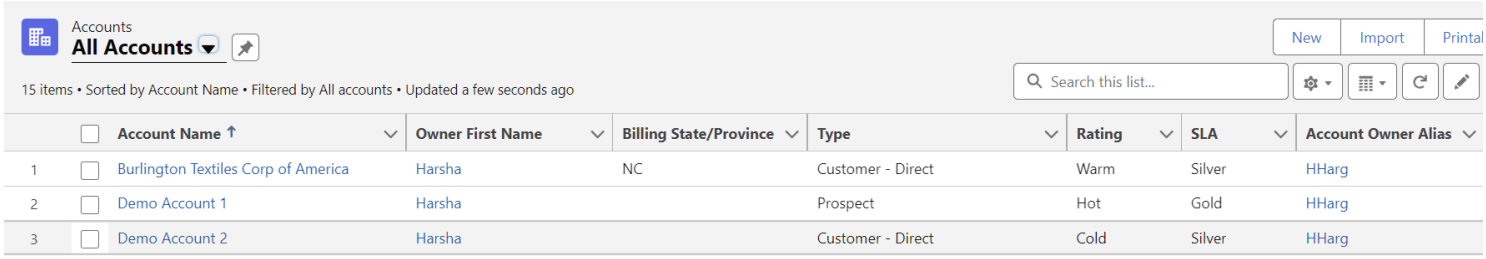


Before-



After Edit and changing Rating to Cold-





Q3. (Task3) Prevent Applicant Record if PAN Card is not available.

EVENT	Before
OPERATIONS	Insert, Update
TRIGGER OBJECT	Applicant
AFFECTING OBJECT	Applicant

Ans. ApplicantTrigger.apxt

```
trigger ApplicantTrigger on Applicant__c (before insert, before update) {
    for(Applicant__c objApp : trigger.new){
        if(String.isBlank(objApp.Pan_Card__c)){
            objApp.Pan_Card__c.addError('Pan Card is required');
        }
    }
}
```

Q4. (Task 4) Prefix Mr. or Ms. To the First Name of Applicant as per the Gender. If Gender is Transgender, then show an Error that 'Transgender is currently not allowed.

EVENT	Before
OPERATIONS	Insert, Update
TRIGGER OBJECT	Applicant
AFFECTING OBJECT	Applicant

Ans. ApplicantTrigger.apxt

```
trigger ApplicantTrigger on Applicant__c (before insert, before update) {
    for(Applicant__c objApp : trigger.new){
        if(objApp.Gender__c == 'Male' && !objApp.First_Name__c.startsWith('Mr.')){
            objApp.First_Name__c = 'Mr.'+objApp.First_Name__c;
        }
        else{
            if(objApp.Gender__c == 'Female' && !objApp.First_Name__c.startsWith('Ms.')){
                objApp.First_Name__c = 'Ms.'+objApp.First_Name__c;
            }
            else{
                if(objApp.Gender__c == 'Transgender'){
                    objApp.Gender__c.addError('Transgender is currently not allowed');
```

```
                }
            }
        }
    }
}
```

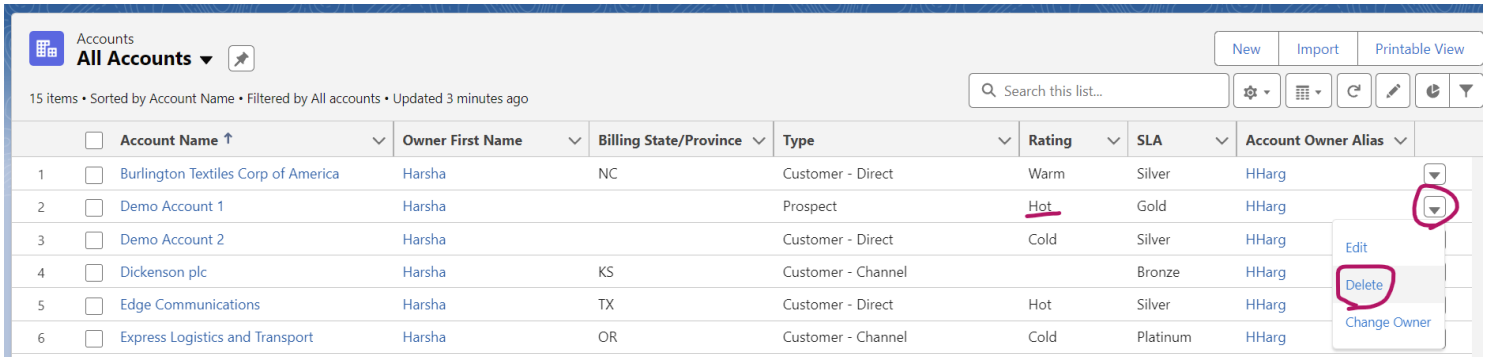
Q5. (Task 5) Prevent Account from Delete if Rating is ‘Hot’.

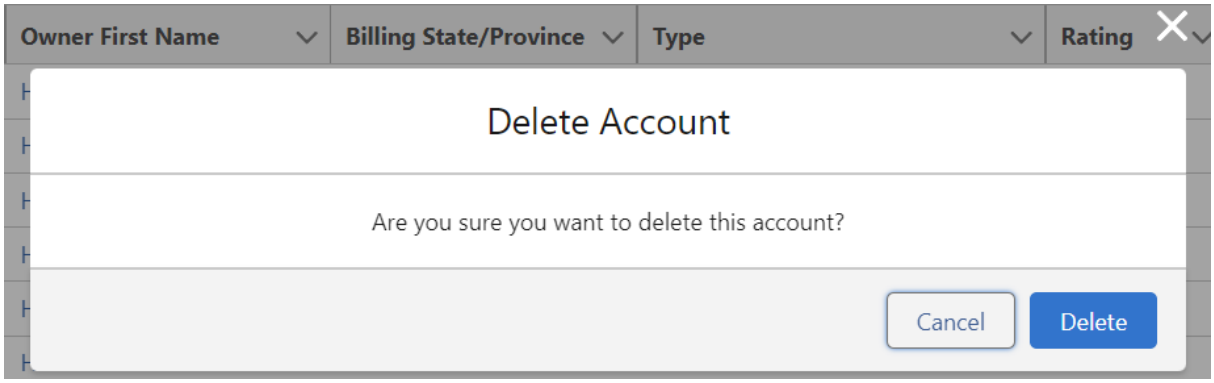
EVENT	Before
OPERATIONS	Insert, Update
TRIGGER OBJECT	Applicant
AFFECTING OBJECT	Applicant

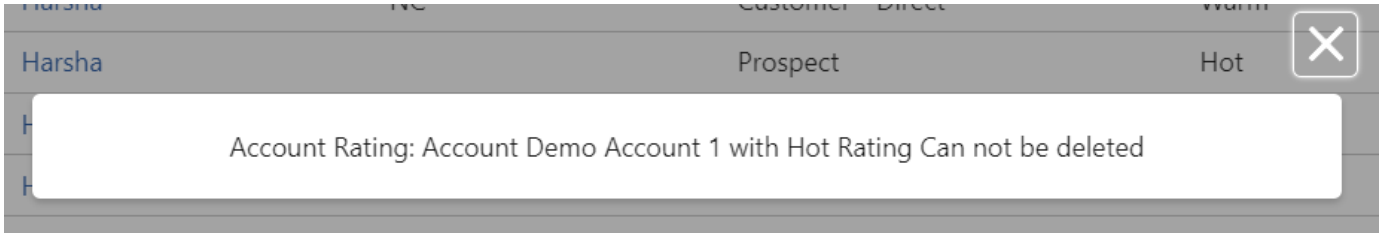
AccountTrigger.apxt

```
trigger AccountTrigger on Account (before delete) {
    for(Account objAcc: trigger.old){
        if(!String.isBlank(objAcc.Rating)){
            if(objAcc.Rating == 'Hot'){
                objAcc.Rating.addError('Account '+ objAcc.Name +' with Hot Rating Can not be deleted');
            }
        }
    }
}
```

Output:





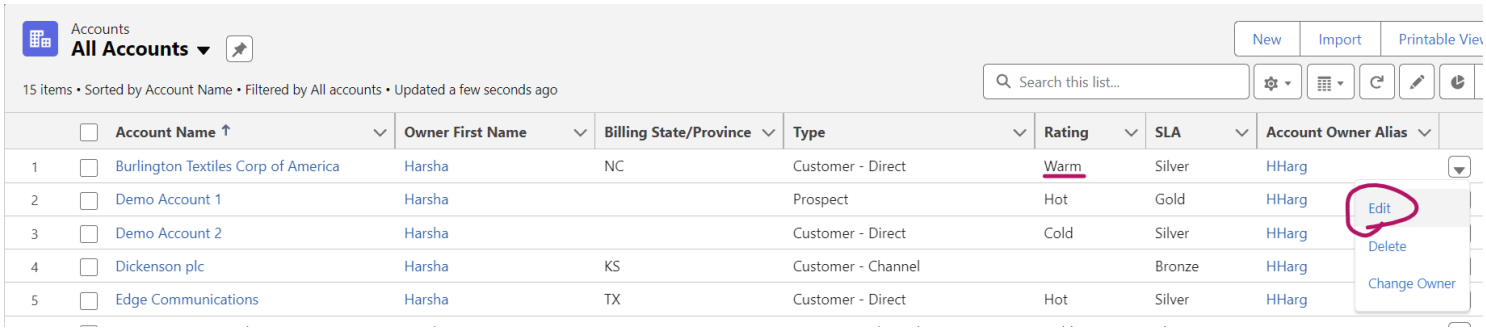


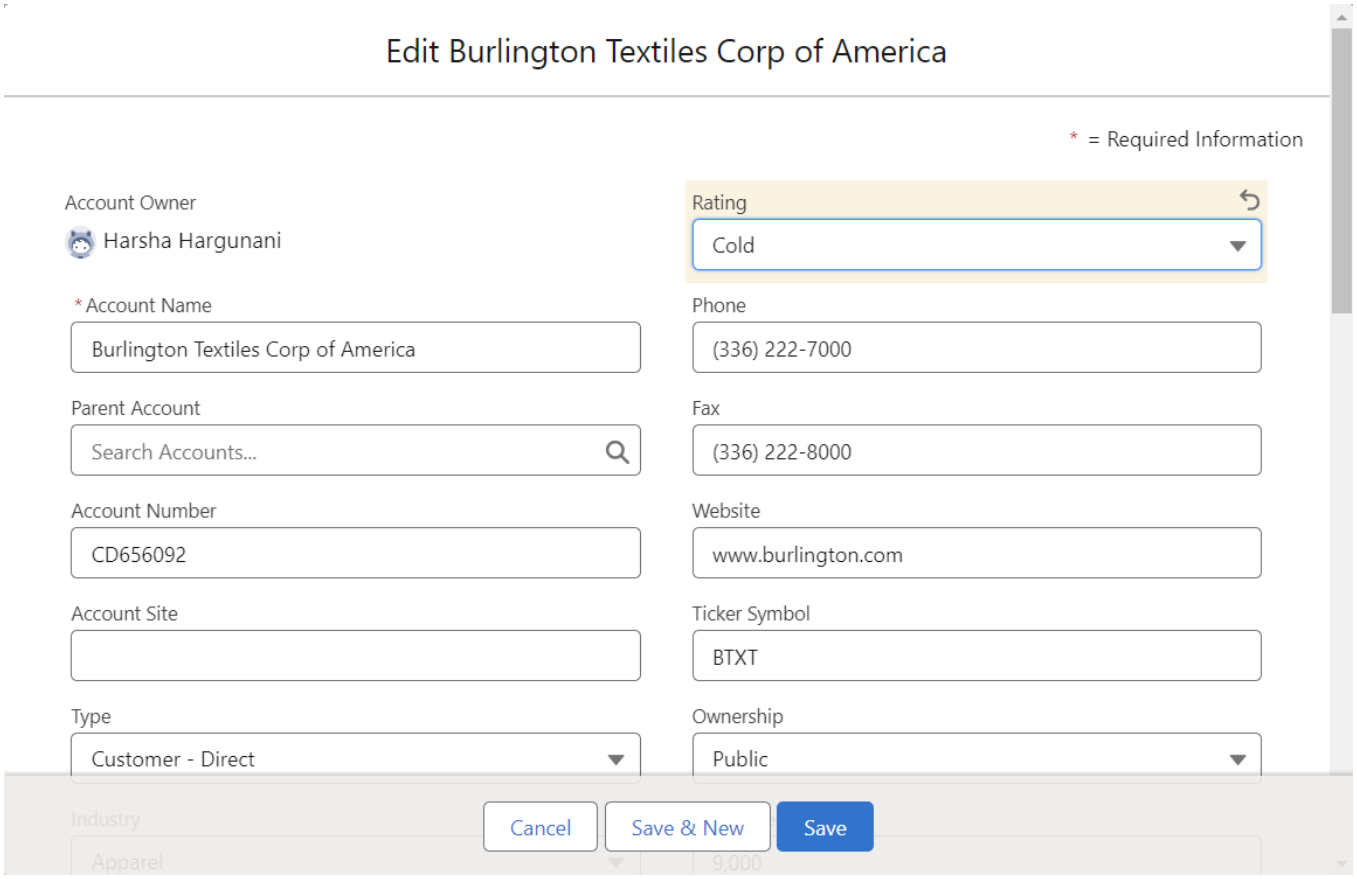
Q6. (Task 6) Prevent Account if Rating is updated from “Warm” to “Cold”.

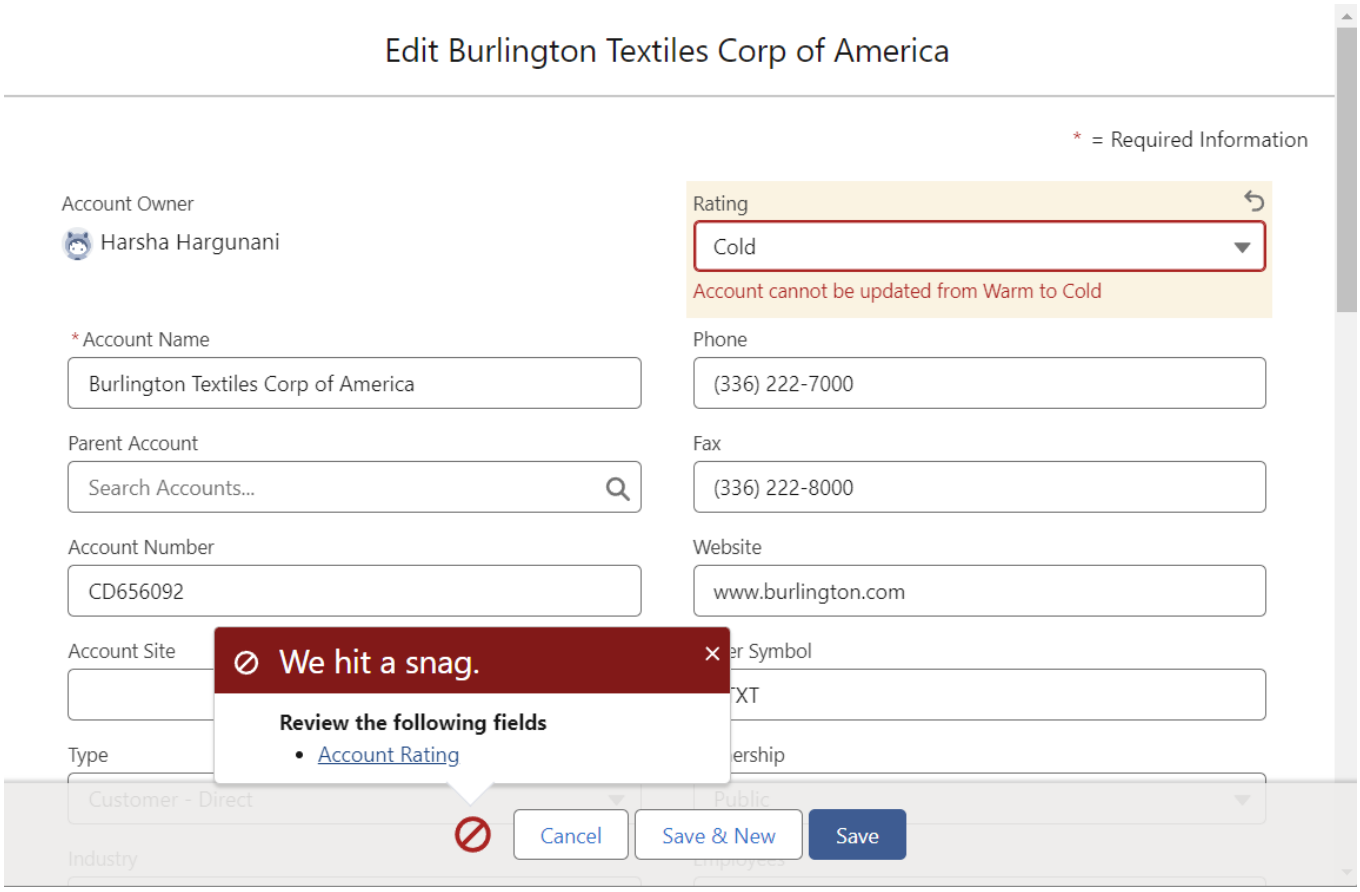
Ans. AccountTrigger.apxt

```
trigger AccountTrigger on Account (before update) {
    if(trigger.isUpdate && trigger.isBefore){
        for(Account objAcc: trigger.new){//Cinemax New Rating = Cold, Old Rating = Warm
            if(objAcc.Rating == 'Cold' && trigger.oldMap.get(objAcc.Id).Rating ==
'Warm'){
                objAcc.Rating.addError('Account cannot be updated from Warm to Cold');
            }
        }
    }
}
```

Output:




Edit Burlington Textiles Corp of America
* = Required Information
Account Owner
Harsha Hargunani
Rating
Cold
*Account Name
Burlington Textiles Corp of America
Phone
(336) 222-7000
Parent Account
Search Accounts...
Fax
(336) 222-8000
Account Number
CD656092
Website
www.burlington.com
Account Site
Ticker Symbol
BTXT
Type
Customer - Direct
Ownership
Public
Industry
Apparel
Cancel
Save & New
Save


Edit Burlington Textiles Corp of America
* = Required Information
Account Owner
Harsha Hargunani
Rating
Cold
Account cannot be updated from Warm to Cold
*Account Name
Burlington Textiles Corp of America
Phone
(336) 222-7000
Parent Account
Search Accounts...
Fax
(336) 222-8000
Account Number
CD656092
Website
www.burlington.com
We hit a snag.
Review the following fields
Account Rating
Account Site
Type
Customer - Direct
Public
Industry
Cancel
Save & New
Save

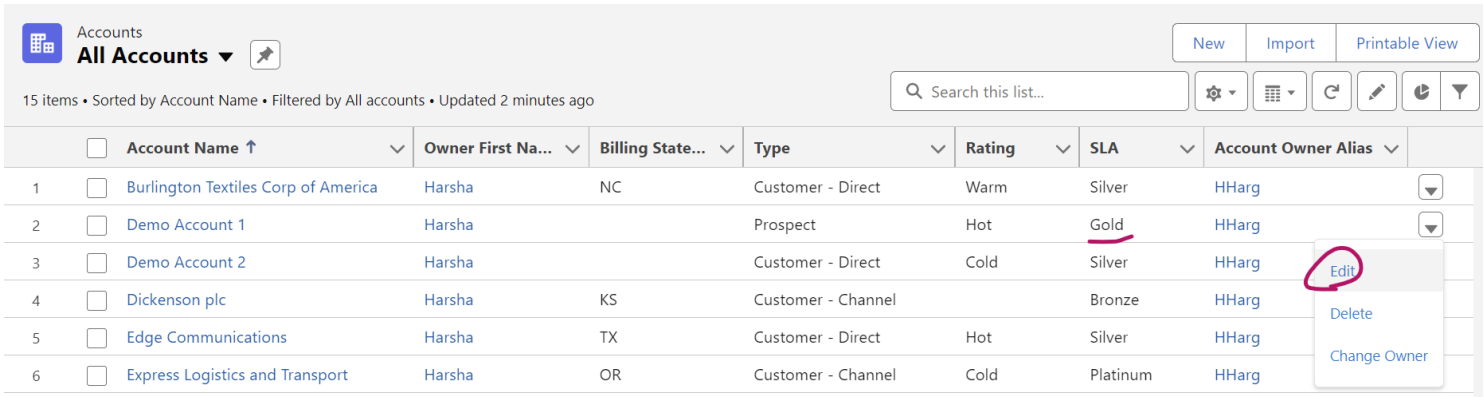
Q7. (Assignment 1) When

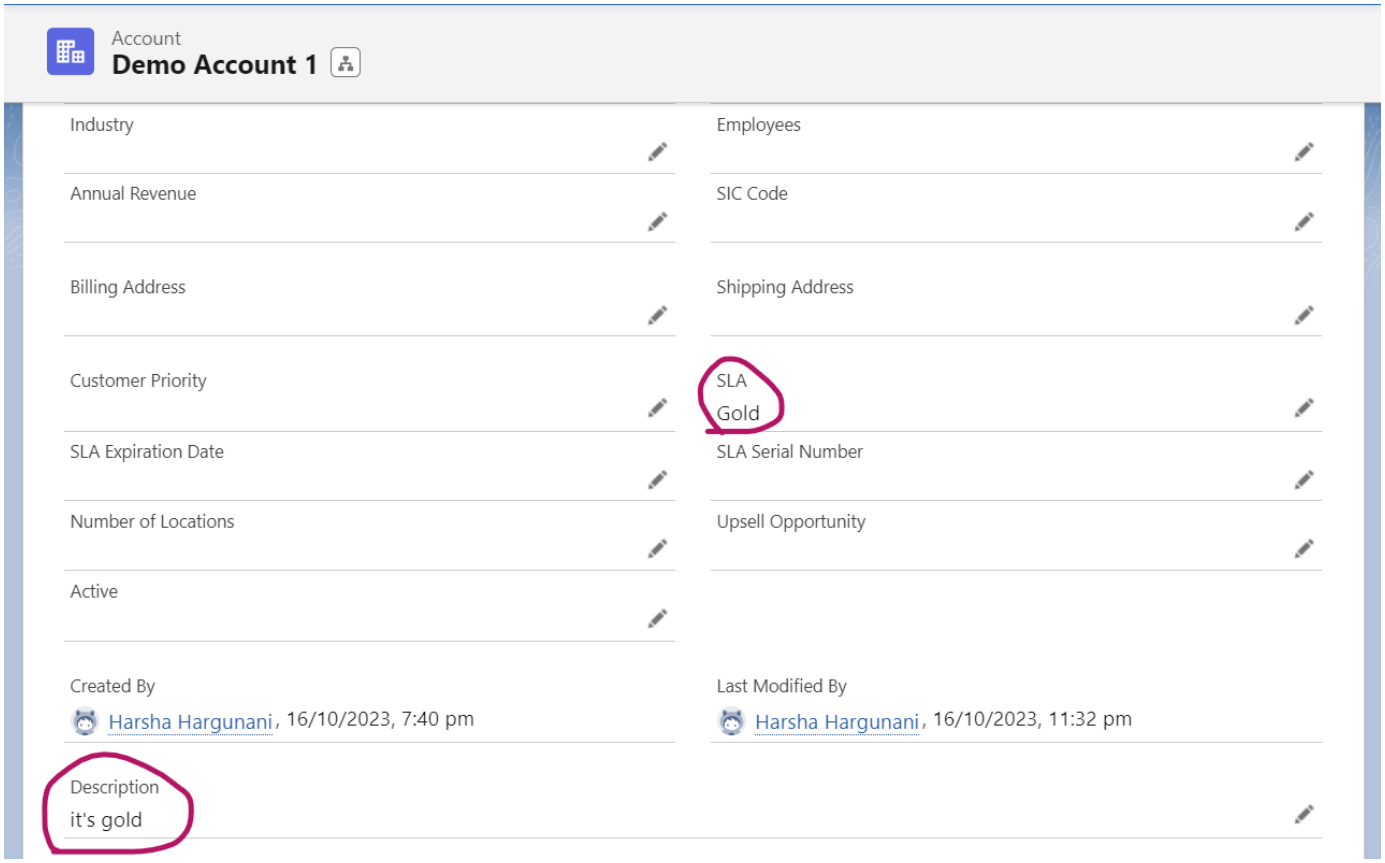
- Account SLA = Gold, then Description = 'it's gold'.
- SLA = Silver, Then Description ='it's a silver'
- Else Description = Blank

Ans. AccountTrigger.apxt

```
trigger AccountTrigger on Account (before insert, before update) {
    for(Account objAcc: trigger.new) {
        if(String.isNotBlank(objAcc.SLA__c)) {
            if(objAcc.SLA__c == 'Gold'){
                objAcc.Description = 'it\'s gold';
            }
            else{
                if(objAcc.SLA__c == 'Silver'){
                    objAcc.Description = 'it\'s a silver';
                }
                else{
                    objAcc.Description = '';
                }
            }
        }
    }
}
```

Output:





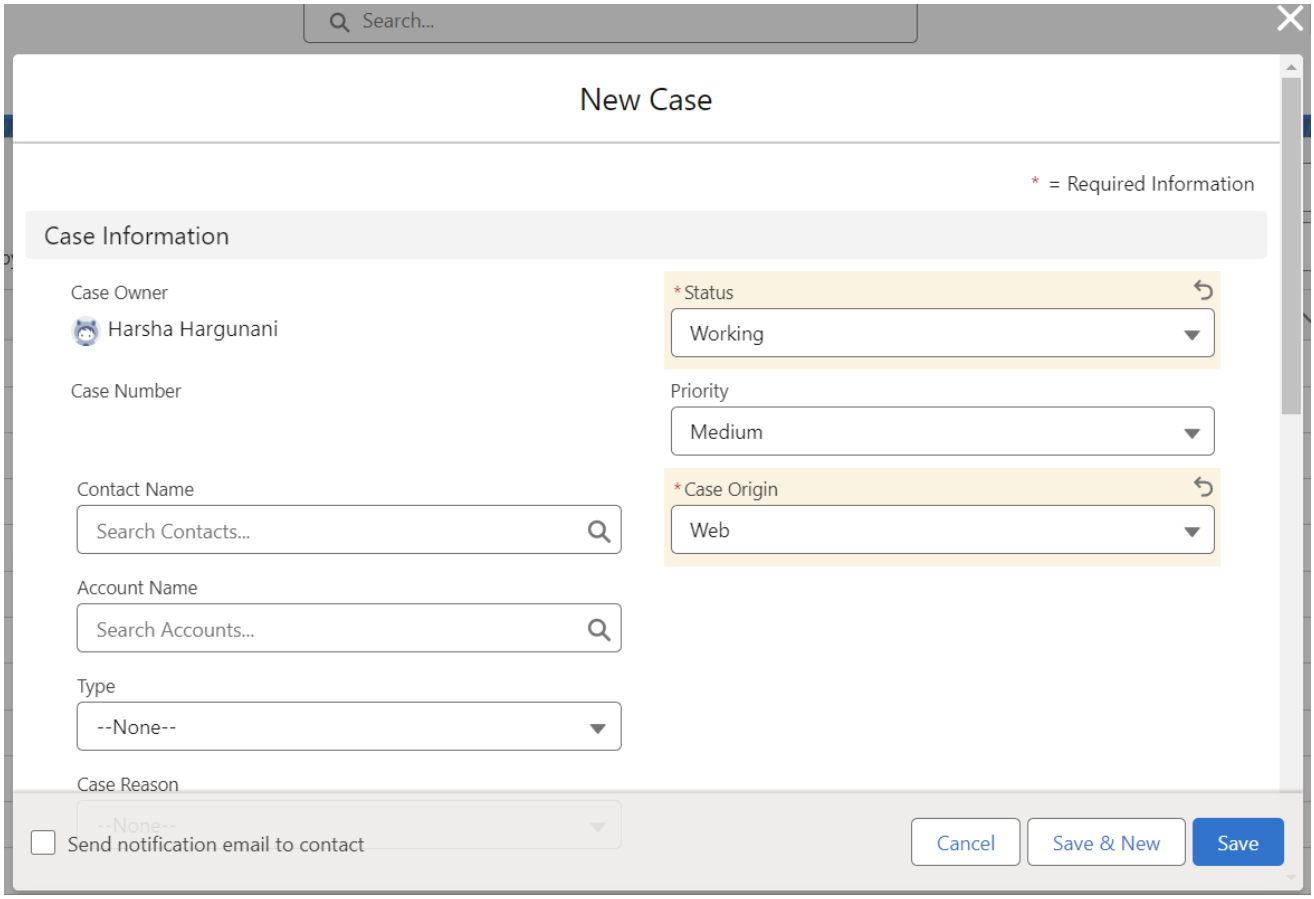
Q8. (Assignment 2)

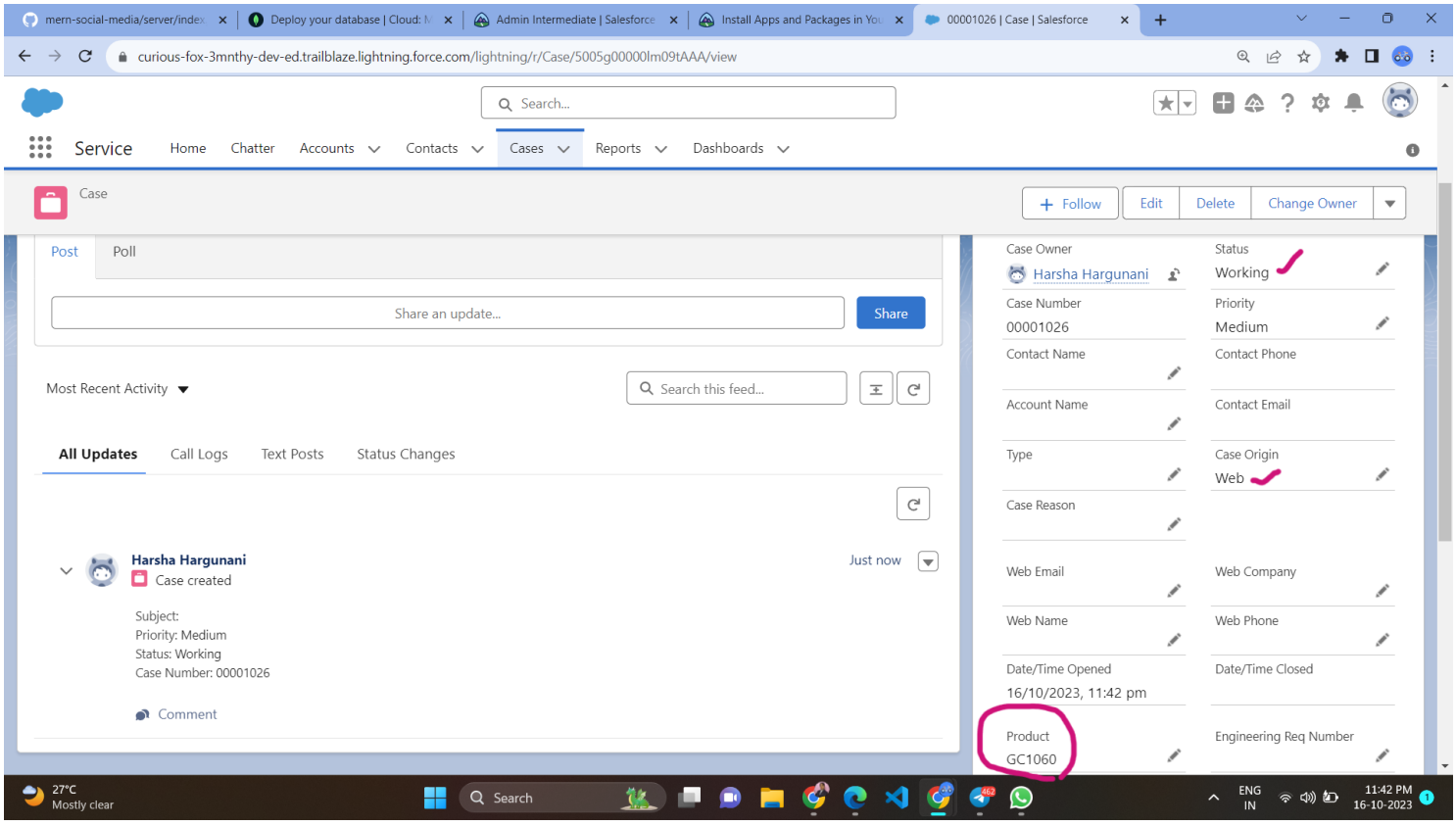
- If Case status is “New” and Origin is “Phone” ➔ Product should be “GC1020”.
- If Case status is “Working” and Origin is “Web” ➔ Product should be “GC1060”.
- If Case status is “Escalated” and Origin is “Email” ➔ Show Error “This case cannot be escalated”

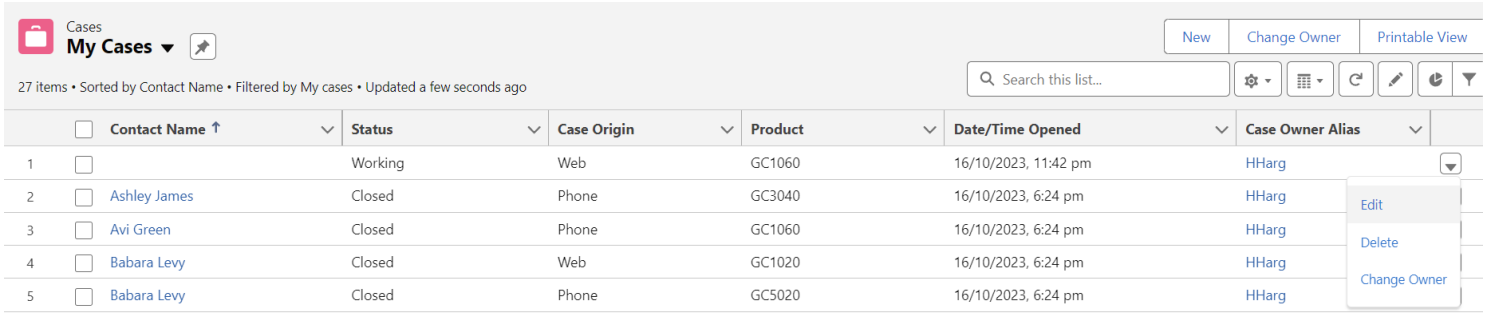
Ans. caseTrigger.apxt

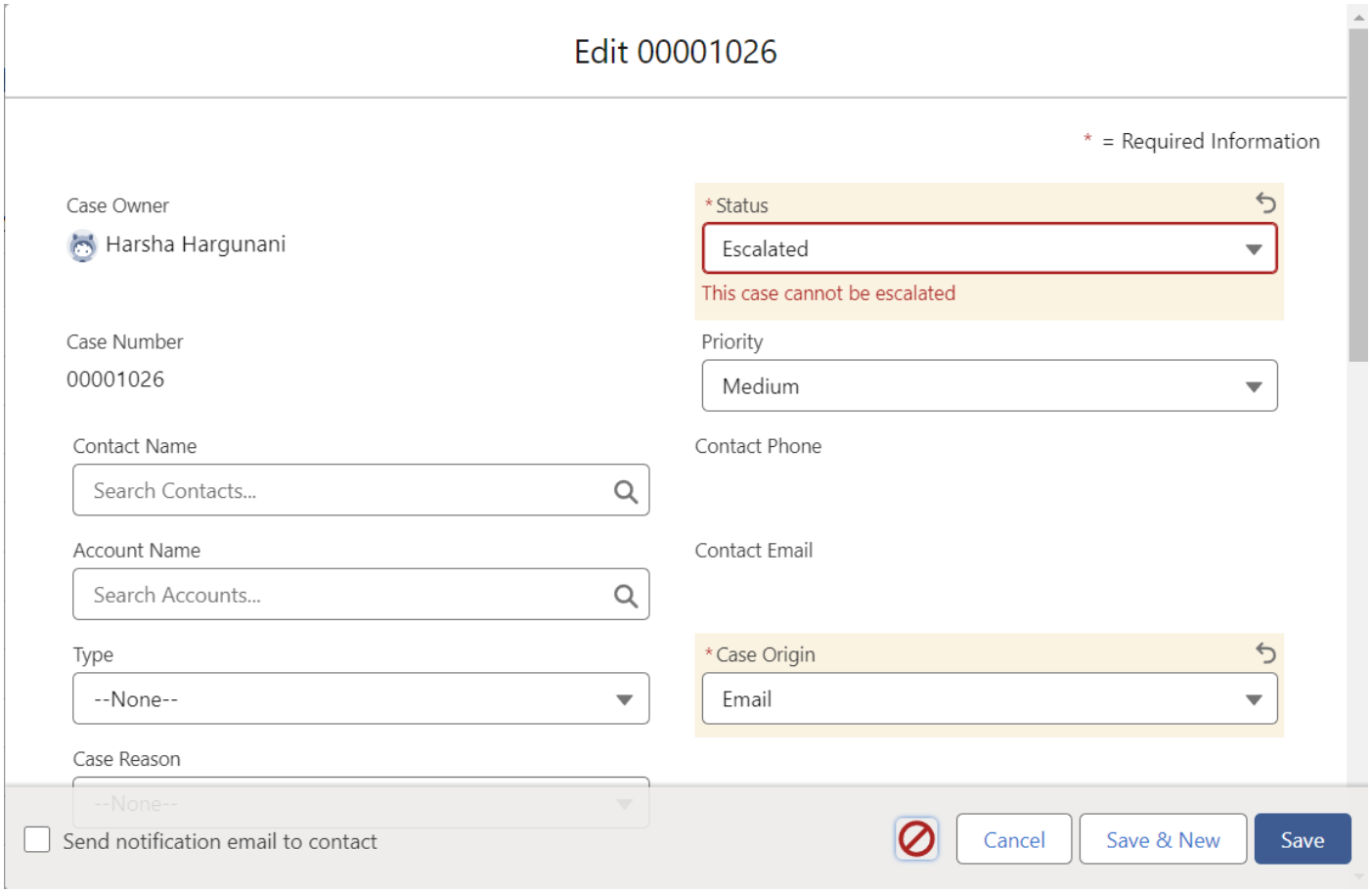
```
trigger caseTrigger on Case (before insert,before update) {
    for(Case objCase : trigger.new){
        if(objCase.Status == 'Escalated' && objCase.Origin == 'Email'){
            objCase.Status.addError('This case cannot be escalated');
        }
        else{
            if(objCase.Status == 'New' && objCase.Origin == 'Phone'){
                objCase.Product__c = 'GC1020';
            }
            else if(objCase.Status == 'Working' && objCase.Origin == 'Web'){
                objCase.Product__c = 'GC1060';
            }
        }
    }
}
```

Output:


Search...
New Case
* = Required Information
Case Information
Case Owner
Harsha Hargunani
Case Number
*Status
Working
Priority
Medium
*Case Origin
Web
Contact Name
Search Contacts...
Account Name
Search Accounts...
Type
--None--
Case Reason
Send notification email to contact
Cancel
Save & New
Save


curious-fox-3mnthy-dev-ed.trailblaze.lightning.force.com/lightning/r/Case/5005g00000lm09tAAA/view
Search...
Service
Home
Chatter
Accounts
Contacts
Cases
Reports
Dashboards
Case
Follow
Edit
Delete
Change Owner
Post
Poll
Share an update...
Share
Most Recent Activity
Search this feed...
All Updates
Call Logs
Text Posts
Status Changes
Harsha Hargunani
Case created
Just now
Subject:
Priority: Medium
Status: Working
Case Number: 00001026
Comment
Case Owner
Harsha Hargunani
Status
Working
Case Number
00001026
Priority
Medium
Contact Name
Contact Phone
Account Name
Contact Email
Type
Case Origin
Web
Case Reason
Web Email
Web Company
Web Name
Web Phone
Date/Time Opened
16/10/2023, 11:42 pm
Date/Time Closed
Product
GC1060
Engineering Req Number
27°C
Mostly clear
Search
ENG
IN
11:42 PM
16-10-2023





Q9. (Assignment 3) Prevent Applicant Record if EMAIL ID is not available.

Ans. ApplicantTrigger.apxt

```
trigger ApplicantTrigger on Applicants__c (before insert, before update) {
    for(Applicants__c objApp : trigger.new){
        if(String.isBlank(objApp.Email__c)){
            objApp.Email__c.addError('Email ID is required');
        }
    }
}
```

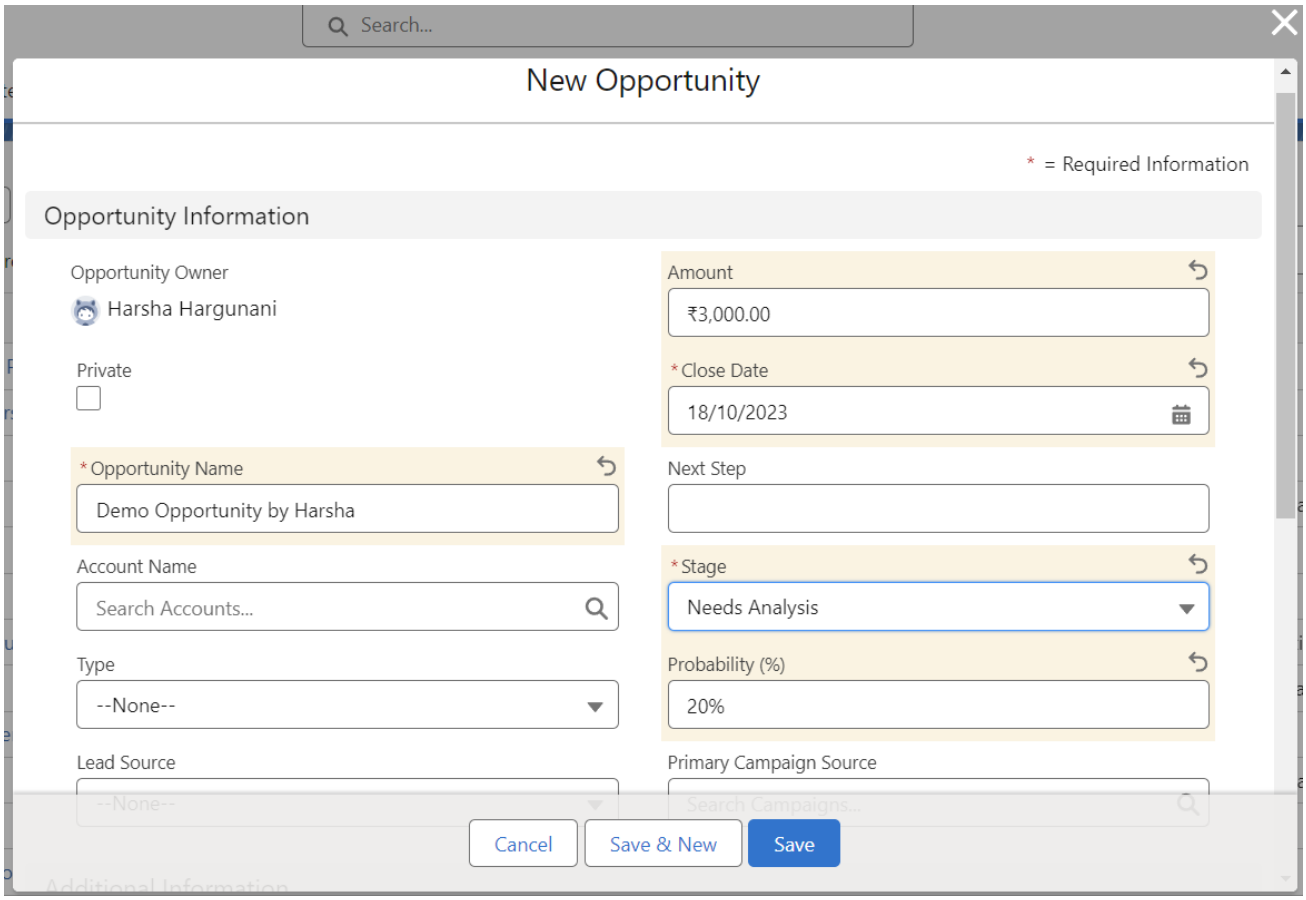
Q10. (Assignment 4) Create / Update Opportunity.

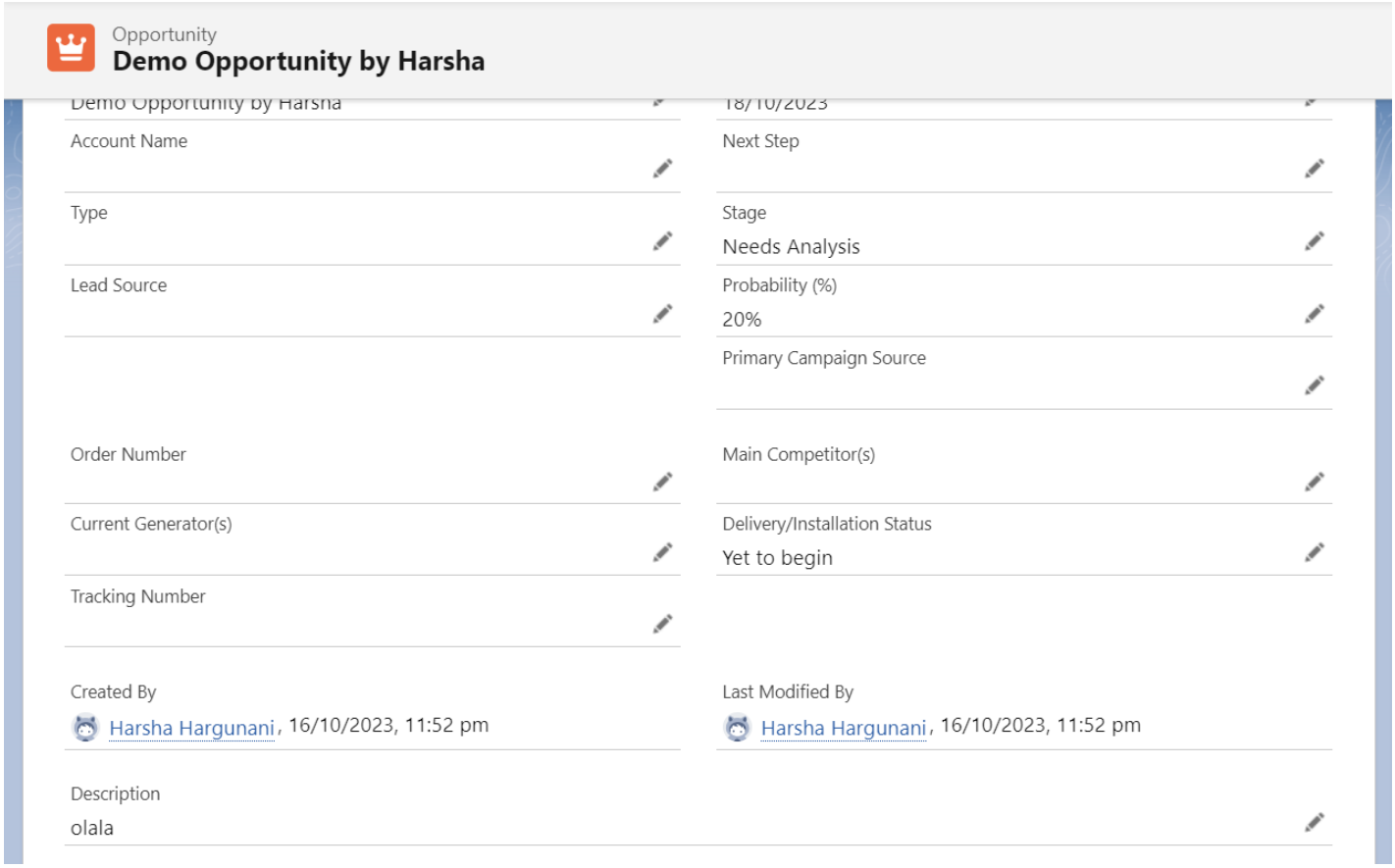
A. If Amount is less than 5000, then Delivery/Installation Status should be “Yet to Begin” and Description should be “olala”.
B. If Amount is Greater or equals to 5000 BUT less than 15000, then Delivery/Installation Status should be “In Progress” and Description should be “Yahooooo”.
C. Delivery/Installation Status should be None and Description should be none, otherwise.

Ans. OpportunityTrigger.apxt

```
trigger OpportunityTrigger on Opportunity (before insert, before update) {
    for(Opportunity objOpp : trigger.new){
        if(objOpp.Amount < 5000){
            objOpp.DeliveryInstallationStatus__c = 'Yet to Begin';
            objOpp.Description = 'olala';
        }
        else if(objOpp.Amount >= 5000 && objOpp.Amount < 15000){
            objOpp.DeliveryInstallationStatus__c = 'In Progress';
            objOpp.Description = 'Yahooooo';
        }
    }
}
```

Output:



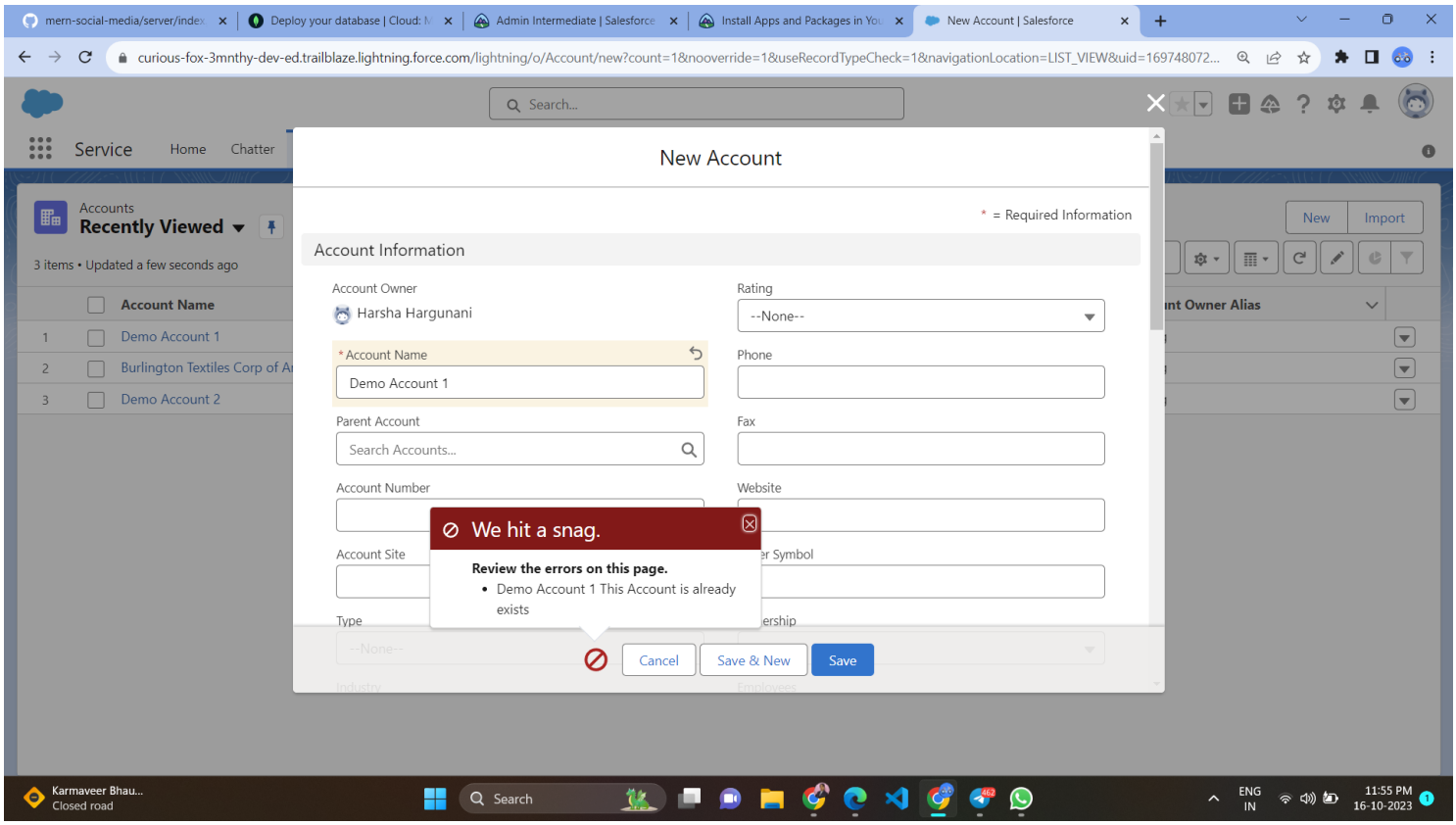


Q11. (Task 7) Insert an Account. If this Account Name already exists in the Account object, then prevent it.
Ans. Steps:

1. Create a SET of Duplicate Finding Field.
2. Add a new Record field in this Set.
3. Create a Map (Key (Duplicate Field),Value (Object)), SOQL over object and Put the Existing Records in the Map for Above SET.
4. Iterate Trigger once again and Check contains Key of the Fields.

AccountTrigger.apxt

```
trigger AccountTrigger on Account (before insert) {
    SET<String> accNameSet = new SET<String>();
    if(trigger.isInsert && trigger.isBefore){
        for(Account objAcc: trigger.new){
            if(!String.isBlank(objAcc.Name))
                accNameSet.add(objAcc.Name);
        }
    }
    Map<String,Account> accMap = new Map<String,Account>();
    for(Account objAcc: [select Id, Name from Account where Name IN: accNameSet]){//Chunk
= 200 Records
        accMap.put(objAcc.Name, objAcc);
    }
```

```
    if(trigger.isInsert && trigger.isBefore){
        for(Account objAcc: trigger.new){
            if(accMap.containsKey(objAcc.Name)){
                objAcc.addError(objAcc.Name + ' This Account is already exists');
            }
        }
    }
}
```

Output: