Name: Sanyam Chandak

Salesforce Developer Certification

Contains Apex specialist (all apex codes) and process automation

APEX SPECIALIST SUPERBADGE

-

- **Challenge 1**

1.Go to the App Launcher -> Search How We Roll Maintenance -> click on Maintenance Requests -> click on first case -> click Details -> change the type Repair to Routine Maintenance -> select Origin = Phone -> Vehicle = select Teardrop Camper , save it.
2.Feed -> Close Case = save it..
3.Go to the Object Manager -> Maintenance Request ->Field & Relationships ->New ->Lookup Relationship -> next -> select Equipment ->next -> Field Label = Equipment ->next->next->next -> save it .
4.Now go to the developer console use below code.

MaintenanceRequestHelper.apxc :-

```
public with sharing class MaintenanceRequestHelper {
   public static void updateworkOrders(List<Case> updWorkOrders, Map<Id,Case>
nonUpdCaseMap) {
      Set<Id> validIds = new Set<Id>();


      For (Case c : updWorkOrders){
         if (nonUpdCaseMap.get(c.Id).Status != 'Closed' && c.Status == 'Closed'){
            if (c.Type == 'Repair' || c.Type == 'Routine Maintenance'){
               validIds.add(c.Id);


            }
         }
      }
```

```
    if (!validIds.isEmpty()){
        List<Case> newCases = new List<Case>();
        Map<Id,Case> closedCasesM = new Map<Id,Case>([SELECT Id, Vehicle__c,
Equipment__c, Equipment__r.Maintenance_Cycle__c,(SELECT
Id,Equipment__c,Quantity__c FROM Equipment_Maintenance_Items__r)
                                                FROM Case WHERE Id IN :validIds]);
        Map<Id,Decimal> maintenanceCycles = new Map<ID,Decimal>();
        AggregateResult[] results = [SELECT Maintenance_Request__c,
MIN(Equipment__r.Maintenance_Cycle__c)cycle FROM
Equipment_Maintenance_Item__c WHERE Maintenance_Request__c IN :ValidIds
GROUP BY Maintenance_Request__c];

    for (AggregateResult ar : results){
        maintenanceCycles.put((Id) ar.get('Maintenance_Request__c'), (Decimal)
ar.get('cycle'));
    }

        for(Case cc : closedCasesM.values()){
            Case nc = new Case (
                ParentId = cc.Id,
            Status = 'New',
                Subject = 'Routine Maintenance',
                Type = 'Routine Maintenance',
                Vehicle__c = cc.Vehicle__c,
                Equipment__c =cc.Equipment__c,
                Origin = 'Web',
                Date_Reported__c = Date.Today()

            );

            If (maintenanceCycles.containskey(cc.Id)){
                nc.Date_Due__c = Date.today().addDays((Integer)
maintenanceCycles.get(cc.Id));
            }

            newCases.add(nc);
        }

       insert newCases;
```

```
        List<Equipment_Maintenance_Item__c> clonedWPs = new
List<Equipment_Maintenance_Item__c>();
        for (Case nc : newCases){
            for (Equipment_Maintenance_Item__c wp :
closedCasesM.get(nc.ParentId).Equipment_Maintenance_Items__r){
                Equipment_Maintenance_Item__c wpClone = wp.clone();
                wpClone.Maintenance_Request__c = nc.Id;
                ClonedWPs.add(wpClone);

            }
         }
         insert ClonedWPs;
      }
   }
}
```

MaintenanceRequest.apxt :-

```
trigger MaintenanceRequest on Case (before update, after update) {
   if(Trigger.isUpdate && Trigger.isAfter){
      MaintenanceRequestHelper.updateWorkOrders(Trigger.New, Trigger.OldMap);
   }
}
```

1. After saving the code go back the How We Roll Maintenance ,
2. click on Maintenance Requests -> click on 2nd case -> click Details -> change the type Repair to Routine Maintenance -> select Origin = Phone -> Vehicle = select Teardrop Camper , save it.
3. Feed -> Close Case = save it..

- ## Challenge 2

- Setup -> Search in quick find box -> click Remote Site Settings -> Name = Warehouse URL , Remote Site URL = https://th-superbadge-apex.herokuapp.com , make sure active is selected.

- Go to the developer console use below code .

WarehouseCalloutService.apxc :-

```
public with sharing class WarehouseCalloutService {

    private static final String WAREHOUSE_URL = 'https://th-superbadge-
apex.herokuapp.com/equipment';

    //@future(callout=true)
    public static void runWarehouseEquipmentSync(){

        Http http = new Http();
        HttpRequest request = new HttpRequest();

        request.setEndpoint(WAREHOUSE_URL);
        request.setMethod('GET');
        HttpResponse response = http.send(request);


        List<Product2> warehouseEq = new List<Product2>();

        if (response.getStatusCode() == 200){
            List<Object> jsonResponse =
(List<Object>)JSON.deserializeUntyped(response.getBody());
            System.debug(response.getBody());

            for (Object eq : jsonResponse){
                Map<String,Object> mapJson = (Map<String,Object>)eq;
                Product2 myEq = new Product2();
                myEq.Replacement_Part__c = (Boolean) mapJson.get('replacement');
                myEq.Name = (String) mapJson.get('name');
                myEq.Maintenance_Cycle__c = (Integer) mapJson.get('maintenanceperiod');
                myEq.Lifespan_Months__c = (Integer) mapJson.get('lifespan');
                myEq.Cost__c = (Decimal) mapJson.get('lifespan');
                myEq.Warehouse_SKU__c = (String) mapJson.get('sku');
                myEq.Current_Inventory__c = (Double) mapJson.get('quantity');
                warehouseEq.add(myEq);
            }
```

```
            if (warehouseEq.size() > 0){
                upsert warehouseEq;
                System.debug('Your equipment was synced with the warehouse one');
                System.debug(warehouseEq);
            }

        }
    }
}
```

- **Challenge 3**

Go to setup -> Seacrh in Quick find box -> Apex Classes -> click Schedule Apex and Jb Name = WarehouseSyncScheduleJob , Apex Class = WarehouseSyncSchedule as it is below shown in the image

WarehouseSyncSchedule.apxc :-

```
global class WarehouseSyncSchedule implements Schedulable {
    global void execute(SchedulableContext ctx) {

        WarehouseCalloutService.runWarehouseEquipmentSync();
    }
}
```

- ## Challenge 4

MaintenanceRequestHelperTest.apxc :-

```
@istest
public with sharing class MaintenanceRequestHelperTest {

    private static final string STATUS_NEW = 'New';
    private static final string WORKING = 'Working';
    private static final string CLOSED = 'Closed';
    private static final string REPAIR = 'Repair';
    private static final string REQUEST_ORIGIN = 'Web';
    private static final string REQUEST_TYPE = 'Routine Maintenance';
    private static final string REQUEST_SUBJECT = 'Testing subject';

    PRIVATE STATIC Vehicle__c createVehicle(){
        Vehicle__c Vehicle = new Vehicle__C(name = 'SuperTruck');
        return Vehicle;
    }

    PRIVATE STATIC Product2 createEq(){
        product2 equipment = new product2(name = 'SuperEquipment',
```

```
                            lifespan_months__C = 10,
                            maintenance_cycle__C = 10,
                            replacement_part__c = true);
        return equipment;
    }

    PRIVATE STATIC Case createMaintenanceRequest(id vehicleId, id equipmentId){
        case cs = new case(Type=REPAIR,
                    Status=STATUS_NEW,
                    Origin=REQUEST_ORIGIN,
                    Subject=REQUEST_SUBJECT,
                    Equipment__c=equipmentId,
                    Vehicle__c=vehicleId);
        return cs;
    }

    PRIVATE STATIC Equipment_Maintenance_Item__c createWorkPart(id equipmentId,id
requestId){
        Equipment_Maintenance_Item__c wp = new
Equipment_Maintenance_Item__c(Equipment__c = equipmentId,
                                                Maintenance_Request__c = requestId);
        return wp;
    }


    @istest
    private static void testMaintenanceRequestPositive(){
        Vehicle__c vehicle = createVehicle();
        insert vehicle;
        id vehicleId = vehicle.Id;

        Product2 equipment = createEq();
        insert equipment;
        id equipmentId = equipment.Id;

        case somethingToUpdate = createMaintenanceRequest(vehicleId,equipmentId);
        insert somethingToUpdate;

        Equipment_Maintenance_Item__c workP =
createWorkPart(equipmentId,somethingToUpdate.id);
        insert workP;
```

```
        test.startTest();
        somethingToUpdate.status = CLOSED;
        update somethingToUpdate;
        test.stopTest();

        Case newReq = [Select id, subject, type, Equipment__c, Date_Reported__c,
Vehicle__c, Date_Due__c
                    from case
                    where status =:STATUS_NEW];

        Equipment_Maintenance_Item__c workPart = [select id
                                    from Equipment_Maintenance_Item__c
                                    where Maintenance_Request__c =:newReq.Id];

        system.assert(workPart != null);
        system.assert(newReq.Subject != null);
        system.assertEquals(newReq.Type, REQUEST_TYPE);
        SYSTEM.assertEquals(newReq.Equipment__c, equipmentId);
        SYSTEM.assertEquals(newReq.Vehicle__c, vehicleId);
        SYSTEM.assertEquals(newReq.Date_Reported__c, system.today());
    }

    @istest
    private static void testMaintenanceRequestNegative(){
        Vehicle__C vehicle = createVehicle();
        insert vehicle;
        id vehicleId = vehicle.Id;

        product2 equipment = createEq();
        insert equipment;
        id equipmentId = equipment.Id;

        case emptyReq = createMaintenanceRequest(vehicleId,equipmentId);
        insert emptyReq;

        Equipment_Maintenance_Item__c workP = createWorkPart(equipmentId,
emptyReq.Id);
        insert workP;

        test.startTest();
```

```
        emptyReq.Status = WORKING;
        update emptyReq;
        test.stopTest();

        list<case> allRequest = [select id
                                 from case];

        Equipment_Maintenance_Item__c workPart = [select id
                                                  from Equipment_Maintenance_Item__c
                                                  where Maintenance_Request__c = :emptyReq.Id];

        system.assert(workPart != null);
        system.assert(allRequest.size() == 1);
    }

    @istest
    private static void testMaintenanceRequestBulk(){
        list<Vehicle__C> vehicleList = new list<Vehicle__C>();
        list<Product2> equipmentList = new list<Product2>();
        list<Equipment_Maintenance_Item__c> workPartList = new
list<Equipment_Maintenance_Item__c>();
        list<case> requestList = new list<case>();
        list<id> oldRequestIds = new list<id>();

        for(integer i = 0; i < 300; i++){
          vehicleList.add(createVehicle());
           equipmentList.add(createEq());
        }
        insert vehicleList;
        insert equipmentList;

        for(integer i = 0; i < 300; i++){
           requestList.add(createMaintenanceRequest(vehicleList.get(i).id,
equipmentList.get(i).id));
        }
        insert requestList;

        for(integer i = 0; i < 300; i++){
           workPartList.add(createWorkPart(equipmentList.get(i).id, requestList.get(i).id));
        }
        insert workPartList;
```

```
        test.startTest();
        for(case req : requestList){
           req.Status = CLOSED;
           oldRequestIds.add(req.Id);
        }
        update requestList;
        test.stopTest();

        list<case> allRequests = [select id
                                  from case
                                  where status =: STATUS_NEW];

        list<Equipment_Maintenance_Item__c> workParts = [select id
                                                         from Equipment_Maintenance_Item__c
                                                         where Maintenance_Request__c in: oldRequestIds];

        system.assert(allRequests.size() == 300);
    }
}
```

- ## Challenge 5

WarehouseCalloutServiceTest.apxc:-

```
@isTest

private class WarehouseCalloutServiceTest {
@isTest
static void testWareHouseCallout(){
Test.startTest();
```

```
// implement mock callout test here
Test.setMock(HTTPCalloutMock.class, new WarehouseCalloutServiceMock());
WarehouseCalloutService.runWarehouseEquipmentSync();
WarehouseCalloutService apc = new WarehouseCalloutService();
System.enqueueJob(apc);
Test.stopTest();
System.assertEquals(1, [SELECT count() FROM Product2]);
}
}
```

WarehouseCalloutServiceMock.apxc:-

```
@isTest
global class WarehouseCalloutServiceMock implements HttpCalloutMock {
   // implement http mock callout
   global static HttpResponse respond(HttpRequest request){

      System.assertEquals('https://th-superbadge-apex.herokuapp.com/equipment',
request.getEndpoint());
      System.assertEquals('GET', request.getMethod());

      // Create a fake response
      HttpResponse response = new HttpResponse();
      response.setHeader('Content-Type', 'application/json');

response.setBody('[{"_id":"55d66226726b611100aaf741","replacement":false,"quantity
":5,"name":"Generator 1000
kW","maintenanceperiod":365,"lifespan":120,"cost":5000,"sku":"100003"}]');
      response.setStatusCode(200);
      return response;
   }
}
```

- ## Challenge 6

WarehouseSyncSchedule.apxc:-

```
global class WarehouseSyncSchedule implements Schedulable {
   global void execute(SchedulableContext ctx) {
```

```
        WarehouseCalloutService.runWarehouseEquipmentSync();
    }
}
```

WarehouseSyncScheduleTest.apxc:-

```
@isTest
public class WarehouseSyncScheduleTest {

    @isTest static void WarehousescheduleTest(){
        String scheduleTime = '00 00 01 * * ?';
        Test.startTest();
        Test.setMock(HttpCalloutMock.class, new WarehouseCalloutServiceMock());
        String jobID=System.schedule('Warehouse Time To Schedule to Test', scheduleTime,
new WarehouseSyncSchedule());
        Test.stopTest();

        CronTrigger a=[SELECT Id FROM CronTrigger where NextFireTime > today];
        System.assertEquals(jobID, a.Id,'Schedule ');


    }
}
```

MODULES

-

- ## Apex Triggers

AccountAddressTrigger.apxt:-

```
trigger AccountAddressTrigger on Account (before insert,before update) {

List<Account> acclst=new List<Account>();

  for(account a:trigger.new){
    if(a.Match_Billing_Address__c==true && a.BillingPostalCode!=null){
    a.ShippingPostalCode=a.BillingPostalCode;

    }

}
}
```

ClosedOpportunityTrigger.apxt:-

```
trigger ClosedOpportunityTrigger on Opportunity (after insert, after update) {

    List <Task> tasks = new List<Task>();
    for(Opportunity opp : [SELECT Id, StageName FROM Opportunity WHERE
StageName='Closed Won'
                           AND Id IN :Trigger.new]){
        tasks.add(new Task(Subject = 'Follow Up Test Task' , WhatId = opp.Id));
    }

    if(tasks.size() > 0){
        insert tasks;
    }
}
```

- **Apex Testing**

VerifyDate.apxc:-

```
public class VerifyDate {

    //method to handle potential checks against two dates
    public static Date CheckDates(Date date1, Date date2) {
        //if date2 is within the next 30 days of date1, use date2.  Otherwise use the end of the month
        if(DateWithin30Days(date1,date2)) {
            return date2;
        } else {
            return SetEndOfMonthDate(date1);
        }
    }

    //method to check if date2 is within the next 30 days of date1
    private static Boolean DateWithin30Days(Date date1, Date date2) {
        //check for date2 being in the past
     if( date2 < date1) { return false; }

     //check that date2 is within (>=) 30 days of date1
     Date date30Days = date1.addDays(30); //create a date 30 days away from date1
        if( date2 >= date30Days ) { return false; }
        else { return true; }
```

```
        }

        //method to return the end of the month of a given date
        private static Date SetEndOfMonthDate(Date date1) {
                Integer totalDays = Date.daysInMonth(date1.year(), date1.month());
                Date lastDay = Date.newInstance(date1.year(), date1.month(), totalDays);
                return lastDay;
        }

}
```

TestVerifyDate.apxc:-

```
@isTest
private class TestVerifyDate {
   static testMethod void TestVerifyDate() {
     VerifyDate.CheckDates(System.today(),System.today().addDays(10));
      VerifyDate.CheckDates(System.today(),System.today().addDays(78));
   }
}
```

RestrictContactByName.apxt:-

```
trigger RestrictContactByName on Contact (before insert, before update) {

	//check contacts prior to insert or update for invalid data
	For (Contact c : Trigger.New) {
		if(c.LastName == 'INVALIDNAME') {   //invalidname is invalid
			c.AddError('The Last Name "'+c.LastName+'" is not allowed for
DML');
		}

	}
}
```

TestRestrictContactByName.apxc:-

```
@isTest

private class TestRestrictContactByName {


   static testMethod void  metodoTest()

   {


      List<Contact> listContact= new List<Contact>();

      Contact c1 = new Contact(FirstName='Francesco', LastName='Riggio' ,
email='Test@test.com');

      Contact c2 = new Contact(FirstName='Francesco1', LastName =
'INVALIDNAME',email='Test@test.com');
```

```
        listContact.add(c1);
        listContact.add(c2);

        Test.startTest();
            try
            {
                insert listContact;
            }
            catch(Exception ee)
            {
            }
        Test.stopTest();
    }
}
```

RandomContactFactory.apxc:-

```
public with sharing class RandomContactFactory
{
	public static List<Contact> generateRandomContacts( Integer noOfContacts, String lastName )
	{
		List<Contact> contacts = new List<Contact>();

		for( Integer i = 0; i < noOfContacts; i++ )
		{
			Contact con = new Contact( FirstName = 'Test '+i, LastName = lastName );
			contacts.add( con );
		}

		return contacts;
	}
}
```

- ## Asynchronous Apex

AccountProcessor.apxc :-

```
public class AccountProcessor
{
  @future
  public static void countContacts(Set<id> setId)
  {
     List<Account> lstAccount = [select id,Number_of_Contacts__c , (select id from
contacts ) from account where id in :setId ];
     for( Account acc : lstAccount )
     {
        List<Contact> lstCont = acc.contacts ;

        acc.Number_of_Contacts__c = lstCont.size();
     }
     update lstAccount;
  }
}
```

AccountProcessorTest.apxc :-

```
@IsTest
public class AccountProcessorTest {
   public static testmethod void TestAccountProcessorTest()
   {
      Account a = new Account();
      a.Name = 'Test Account';
      Insert a;

      Contact cont = New Contact();
      cont.FirstName ='Bob';
      cont.LastName ='Masters';
      cont.AccountId = a.Id;
      Insert cont;

      set<Id> setAccId = new Set<ID>();
      setAccId.add(a.id);

      Test.startTest();
        AccountProcessor.countContacts(setAccId);
      Test.stopTest();

      Account ACC = [select Number_of_Contacts__c from Account where id = :a.id LIMIT
1];
      System.assertEquals ( Integer.valueOf(ACC.Number_of_Contacts__c) ,1);
  }

}
```

LeadProcessor.apxc :-

```
global class LeadProcessor implements Database.Batchable<sObject>, Database.Stateful
{
   global Integer recordsProcessed = 0;
   global Database.QueryLocator start(Database.BatchableContext bc) {
      return Database.getQueryLocator([SELECT ID, LeadSource FROM Lead]);
```

```
    }

    global void execute(Database.BatchableContext bc, List<Lead> scope) {
        for (Lead lead : scope) {
            lead.LeadSource = 'Dreamforce';
            recordsProcessed = recordsProcessed + 1;
            System.debug(lead.LeadSource);
        }
        update scope;
    }

    global void finish(Database.BatchableContext bc){
        System.debug(recordsProcessed + ' records processed. Shazam!');
    }
}
```

LeadProcessorTest.apxc :-

```
@isTest
private class LeadProcessorTest {

    @TestSetup
    static void setup(){
        List<Lead> leads = new List<Lead>();

        for (Integer i = 0; i < 200; i++) {
            leads.add(new Lead(LastName='Lead ' + i, Company='Company Number ' + i,
Status='Open - Not Contacted'));
        }

        insert leads;
    }

    static testMethod void test() {

        Test.startTest();
        LeadProcessor lp = new LeadProcessor();
        Id batchId = Database.executeBatch(lp);
        Test.stopTest();
```

```
        System.assertEquals(200, [select count() from lead where LeadSource =
'Dreamforce']);

    }
}
```

AddPrimaryContact.apxc :-

```
public class AddPrimaryContact implements Queueable
{
    private Contact c;
    private String state;
    public  AddPrimaryContact(Contact c, String state)
    {
        this.c = c;
        this.state = state;
    }
    public void execute(QueueableContext context)
    {
         List<Account> ListAccount = [SELECT ID, Name ,(Select id,FirstName,LastName from
contacts ) FROM ACCOUNT WHERE BillingState = :state LIMIT 200];
        List<Contact> lstContact = new List<Contact>();
        for (Account acc:ListAccount)
        {
                Contact cont = c.clone(false,false,false,false);
                cont.AccountId =  acc.id;
                lstContact.add( cont );
        }

        if(lstContact.size() >0 )
        {
           insert lstContact;
        }

    }
```

```
}
```

AddPrimaryContactTest.apxc :-

```
@isTest
public class AddPrimaryContactTest
{
    @isTest static void TestList()
    {
        List<Account> Teste = new List <Account>();
        for(Integer i=0;i<50;i++)
        {
            Teste.add(new Account(BillingState = 'CA', name = 'Test'+i));
        }
        for(Integer j=0;j<50;j++)
        {
            Teste.add(new Account(BillingState = 'NY', name = 'Test'+j));
        }
        insert Teste;

        Contact co = new Contact();
        co.FirstName='demo';
        co.LastName ='demo';
        insert co;
        String state = 'CA';

         AddPrimaryContact apc = new AddPrimaryContact(co, state);
         Test.startTest();
           System.enqueueJob(apc);
         Test.stopTest();
     }
 }
```

DailyLeadProcessor.apxc :-

```
global class DailyLeadProcessor implements Schedulable{
   global void execute(SchedulableContext ctx){
      List<Lead> leads = [SELECT Id, LeadSource FROM Lead WHERE LeadSource = ''];
```

```
        if(leads.size() > 0){
            List<Lead> newLeads = new List<Lead>();

            for(Lead lead : leads){
                lead.LeadSource = 'DreamForce';
                newLeads.add(lead);
            }

            update newLeads;
        }
    }
}
```

DailyLeadProcessorTest.apxc :-

```
@isTest
private class DailyLeadProcessorTest{

    public static String CRON_EXP = '0 0 0 2 6 ? 2022';

    static testmethod void testScheduledJob(){
        List<Lead> leads = new List<Lead>();

        for(Integer i = 0; i < 200; i++){
            Lead lead = new Lead(LastName = 'Test ' + i, LeadSource = '', Company = 'Test
Company ' + i, Status = 'Open - Not Contacted');
            leads.add(lead);
        }
        insert leads;

        Test.startTest();

        String jobId = System.schedule('Update LeadSource to DreamForce', CRON_EXP,
new DailyLeadProcessor());

        Test.stopTest();
    }
}
```

- ## Apex Integration Services

AnimalLocator.apxc :-

```
public class AnimalLocator
{
  public static String getAnimalNameById(Integer id)
   {
       Http http = new Http();
       HttpRequest request = new HttpRequest();
       request.setEndpoint('https://th-apex-http-callout.herokuapp.com/animals/'+id);
       request.setMethod('GET');
       HttpResponse response = http.send(request);
         String strResp = '';
          system.debug('******response '+response.getStatusCode());
          system.debug('******response '+response.getBody());
       // If the request is successful, parse the JSON response.
       if (response.getStatusCode() == 200)
       {
          Map<String, Object> results = (Map<String, Object>)
JSON.deserializeUntyped(response.getBody());
           // Cast the values in the 'animals' key as a list
          Map<string,object> animals = (map<string,object>) results.get('animal');
           System.debug('Received the following animals:' + animals );
           strResp = string.valueof(animals.get('name'));
           System.debug('strResp >>>>>>' + strResp );
       }
       return strResp ;
   }

}
```

AnimalLocatorTest.apxc :-

```
@isTest
private class AnimalLocatorTest{
   @isTest static  void AnimalLocatorMock1() {
       Test.SetMock(HttpCallOutMock.class, new AnimalLocatorMock());
```

```
        string result=AnimalLocator.getAnimalNameById(3);
        string expectedResult='chicken';
        System.assertEquals(result, expectedResult);
    }
}
```

ParkService.apxc :-

```
//Generated by wsdl2apex

public class ParkService {
    public class byCountryResponse {
        public String[] return_x;
        private String[] return_x_type_info = new
String[]{'return','http://parks.services/',null,'0','-1','false'};
        private String[] apex_schema_type_info = new
String[]{'http://parks.services/','false','false'};
        private String[] field_order_type_info = new String[]{'return_x'};
    }
    public class byCountry {
        public String arg0;
        private String[] arg0_type_info = new
String[]{'arg0','http://parks.services/',null,'0','1','false'};
        private String[] apex_schema_type_info = new
String[]{'http://parks.services/','false','false'};
        private String[] field_order_type_info = new String[]{'arg0'};
    }
    public class ParksImplPort {
        public String endpoint_x = 'https://th-apex-soap-
service.herokuapp.com/service/parks';
        public Map<String,String> inputHttpHeaders_x;
        public Map<String,String> outputHttpHeaders_x;
        public String clientCertName_x;
        public String clientCert_x;
        public String clientCertPasswd_x;
        public Integer timeout_x;
        private String[] ns_map_type_info = new String[]{'http://parks.services/',
'ParkService'};
        public String[] byCountry(String arg0) {
            ParkService.byCountry request_x = new ParkService.byCountry();
            request_x.arg0 = arg0;
            ParkService.byCountryResponse response_x;
```

```
        Map<String, ParkService.byCountryResponse> response_map_x = new
Map<String, ParkService.byCountryResponse>();
        response_map_x.put('response_x', response_x);
        WebServiceCallout.invoke(
          this,
          request_x,
          response_map_x,
          new String[]{endpoint_x,
          '',
          'http://parks.services/',
          'byCountry',
          'http://parks.services/',
          'byCountryResponse',
          'ParkService.byCountryResponse'}
        );
        response_x = response_map_x.get('response_x');
        return response_x.return_x;
      }
   }
}
```

ParkLocator.apxc :-

```
public class ParkLocator {
   public static string[] country(String country) {
      parkService.parksImplPort park = new parkService.parksImplPort();
      return park.byCountry(country);
   }
}
```

ParkLocatorTest.apxc :-

```
@isTest
private class ParkLocatorTest {
   @isTest static void testCallout() {

      Test.setMock(WebServiceMock.class, new ParkServiceMock());

      String country = 'Germany';
      String[] result = ParkLocator.Country(country);
```

```
        System.assertEquals(new List<String>{'Hamburg Wadden Sea National Park',
'Hainich National Park', 'Bavarian Forest National Park'}, result);
    }
}
```

AccountManager.apxc:-

```
@RestResource(urlMapping='/Accounts/*/contacts')
global with sharing class AccountManager{
    @HttpGet
    global static Account getAccount(){
        RestRequest req = RestContext.request;
        String accId = req.requestURI.substringBetween('Accounts/', '/contacts');
        Account acc = [SELECT Id, Name, (SELECT Id, Name FROM Contacts)
                       FROM Account WHERE Id = :accId];

        return acc;
    }
}
```

AccountManagerTest.apxc:-

```
@IsTest
private class AccountManagerTest{
    @isTest static void testAccountManager(){
        Id recordId = getTestAccountId();
        // Set up a test request
        RestRequest request = new RestRequest();
        request.requestUri =
            'https://ap5.salesforce.com/services/apexrest/Accounts/'+ recordId +'/contacts';
        request.httpMethod = 'GET';
        RestContext.request = request;

        Account  acc = AccountManager.getAccount();

        System.assert(acc != null);
    }

    private static Id getTestAccountId(){
```

```
        Account acc = new Account(Name = 'TestAcc2');
        Insert acc;

        Contact con = new Contact(LastName = 'TestCont2', AccountId = acc.Id);
        Insert con;

        return acc.Id;
    }
}
```

PROCESS AUTOMATION SPECIALIST SUPERBADGE

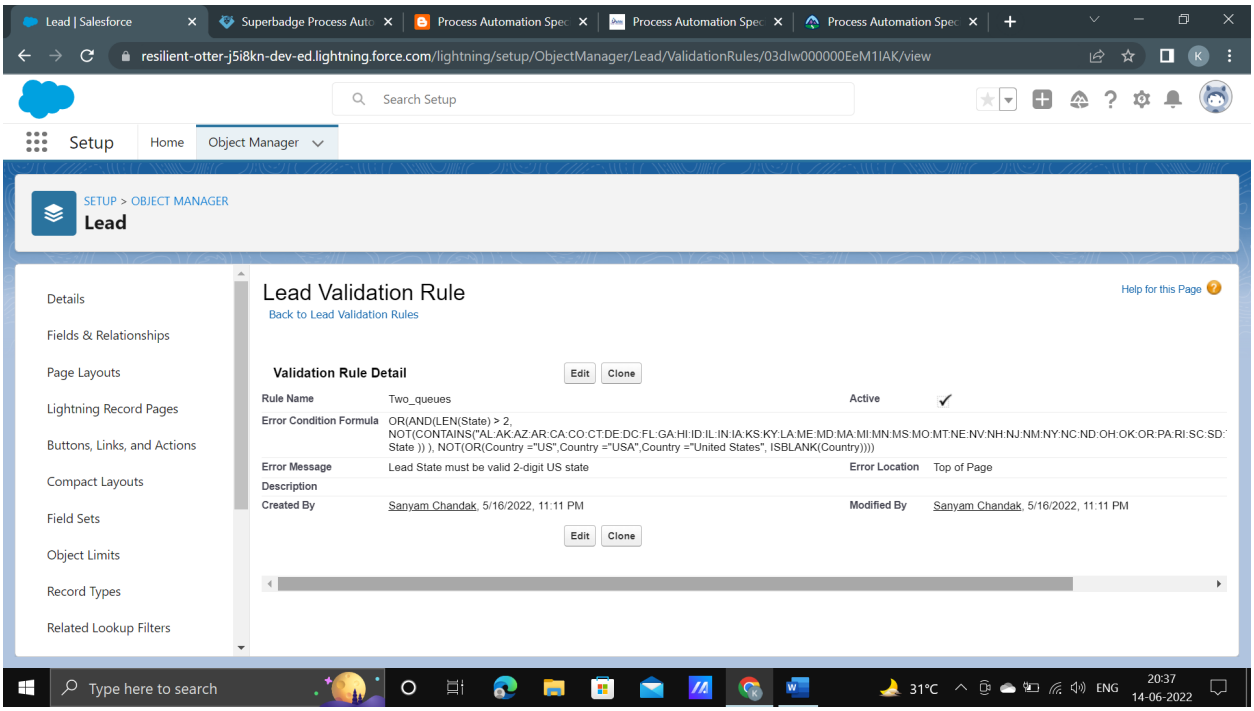
-

- ## Challenge 1: Automate Leads

Step 1: Creating Validation rule on leads

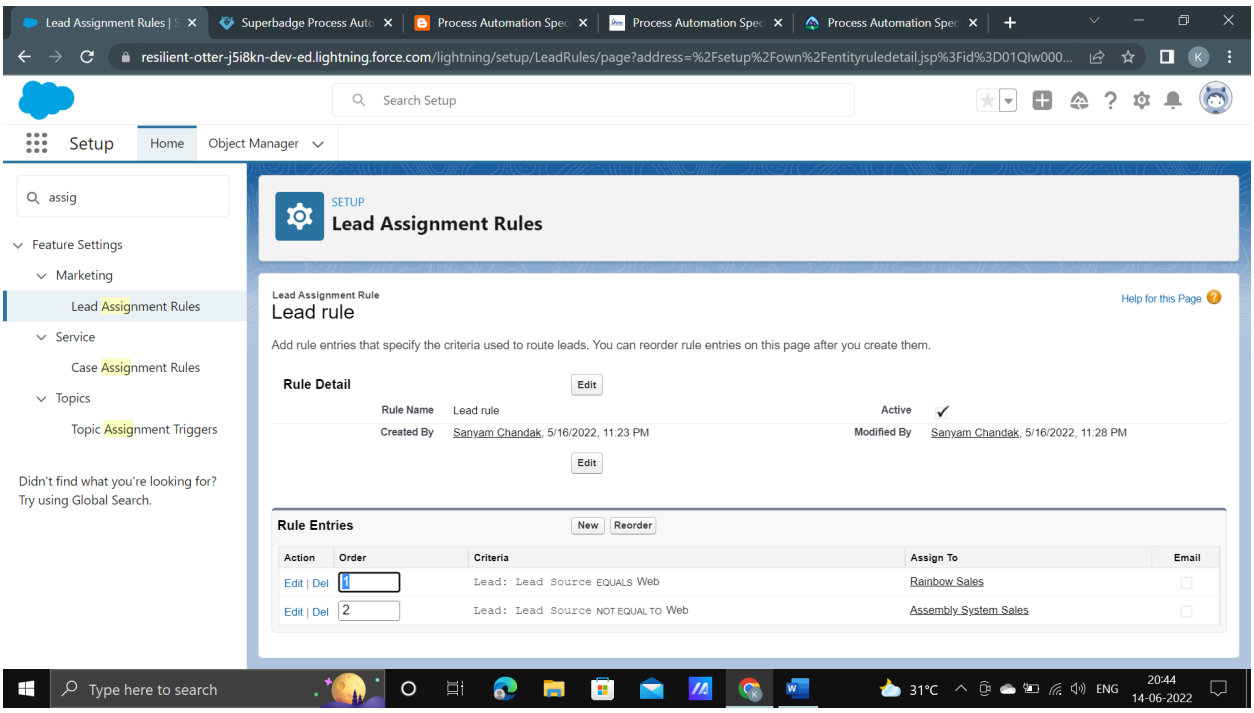
Error Condition Formula:

```
OR(AND(LEN(State) > 2,
NOT(CONTAINS("AL:AK:AZ:AR:CA:CO:CT:DE:DC:FL:GA:HI:ID:IL:IN:IA:KS:KY:LA:ME:MD:MA:MI:MN:MS:MO:MT:
NE:NV:NH:NJ:NM:NY:NC:ND:OH:OK:OR:PA:RI:SC:SD:TN:TX:UT:VT:VA:WA:WV:WI:WY", State )) ),
NOT(OR(Country ="US",Country ="USA",Country ="United States", ISBLANK(Country))))
```



Step 2: Creating two queues: Rainbow Sales and Assembly System Sales

Step 3: Creating assignment rule



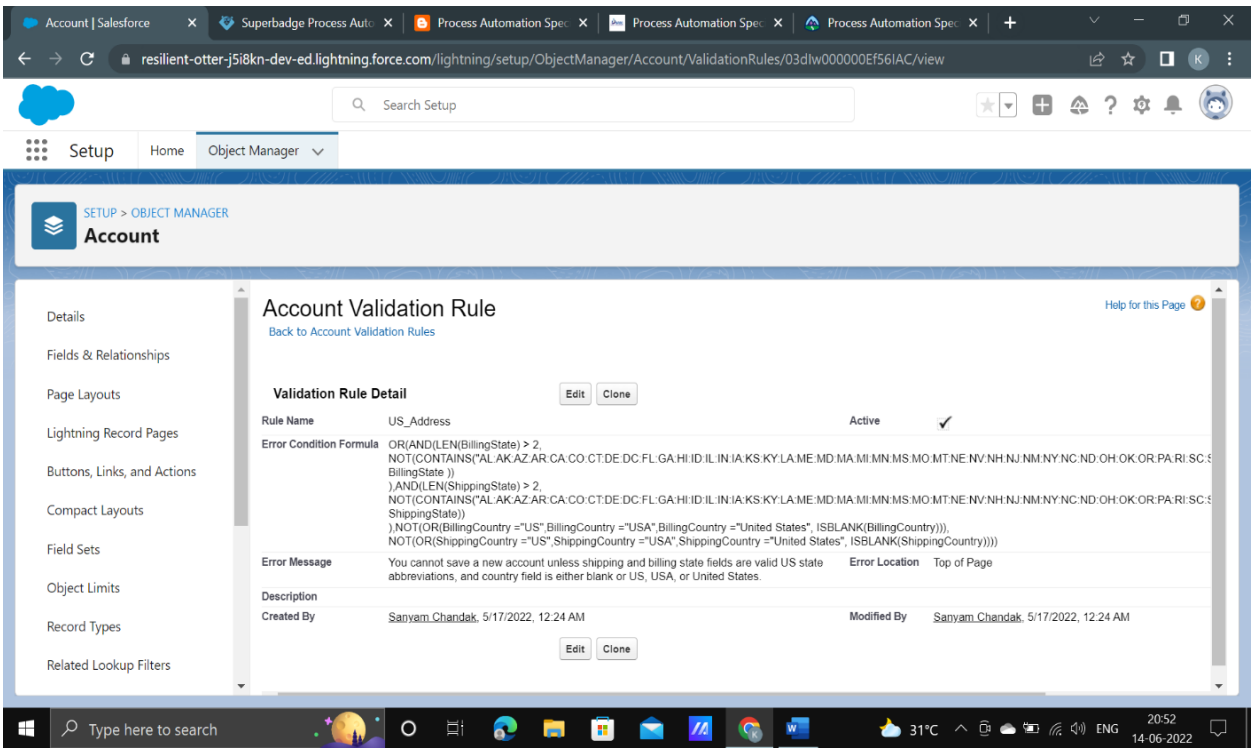
- ## Challenge 2: Automate Accounts

Step 1: Creating required fields

Step 2: Creating 1st validation rule on Account

Error Condition Formula:

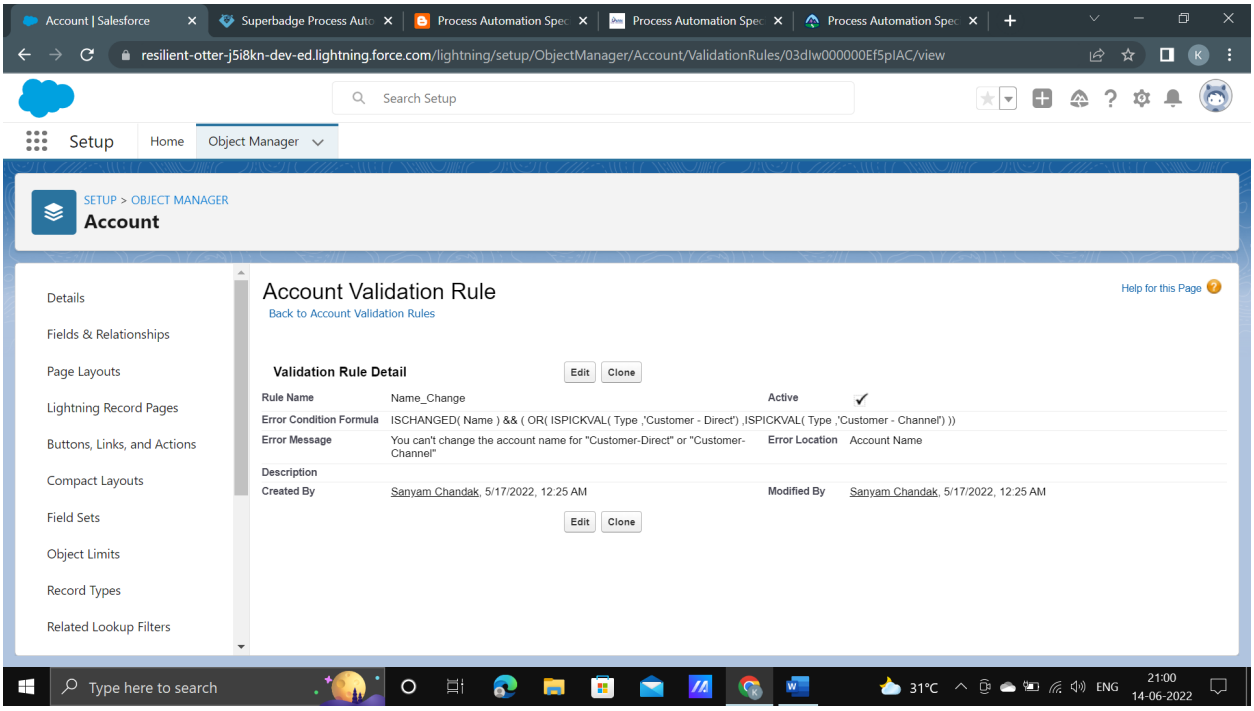
OR(AND(LEN(BillingState) > 2,
NOT(CONTAINS("AL:AK:AZ:AR:CA:CO:CT:DE:DC:FL:GA:HI:ID:IL:IN:IA:KS:KY:LA:ME:MD:MA:MI:MN:MS:MO:MT:NE:NV:NH:NJ:NM:NY:NC:ND:OH:OK:OR:PA:RI:SC:SD:TN:TX:UT:VT:VA:WA:WV:WI:WY", BillingState))
),AND(LEN(ShippingState) > 2,
NOT(CONTAINS("AL:AK:AZ:AR:CA:CO:CT:DE:DC:FL:GA:HI:ID:IL:IN:IA:KS:KY:LA:ME:MD:MA:MI:MN:MS:MO:MT:NE:NV:NH:NJ:NM:NY:NC:ND:OH:OK:OR:PA:RI:SC:SD:TN:TX:UT:VT:VA:WA:WV:WI:WY", ShippingState))
),NOT(OR(BillingCountry ="US",BillingCountry ="USA",BillingCountry ="United States",
ISBLANK(BillingCountry))),
NOT(OR(ShippingCountry ="US",ShippingCountry ="USA",ShippingCountry ="United States",
ISBLANK(ShippingCountry))))



Step 3: Creating 2nd validation rule on Account

Error Condition Formula:

ISCHANGED(Name) && (OR(ISPICKVAL(Type ,'Customer - Direct') ,ISPICKVAL(Type ,'Customer - Channel')
))



- ## Challenge 3: Create Robot Setup Object

Step 1: Creating an object named ***Robot_Setup*** with the data type AutoNumber and the display format ***ROBOT SETUP-{0000} starting with 0.***

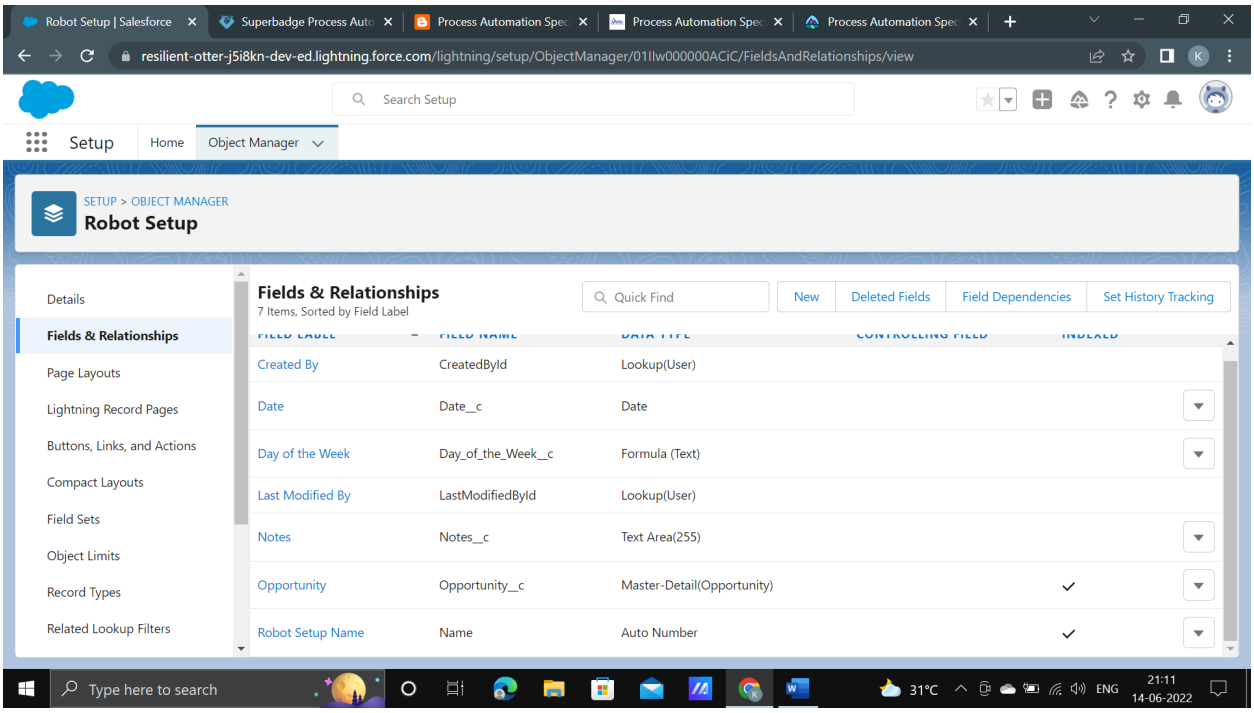
Step 2: Creating required fields

Date: - Date type field

Notes: - TextArea (255)

Day of the Week: - Formula (Text)

Opportunity: - Master-Detail Relationship with Opportunity.



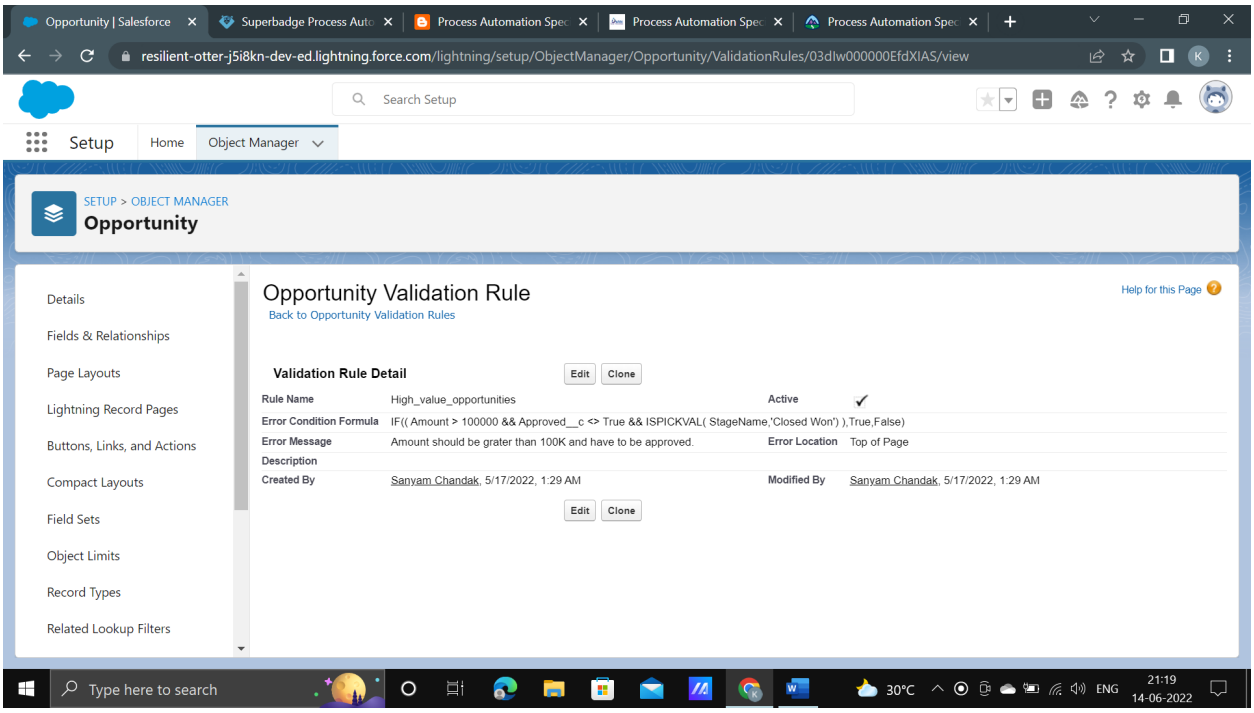
- ## Challenge 4: Create Sales Process and Validate Opportunities

Step 1: Creating ‘Approval’ checkbox field to opportunity object.

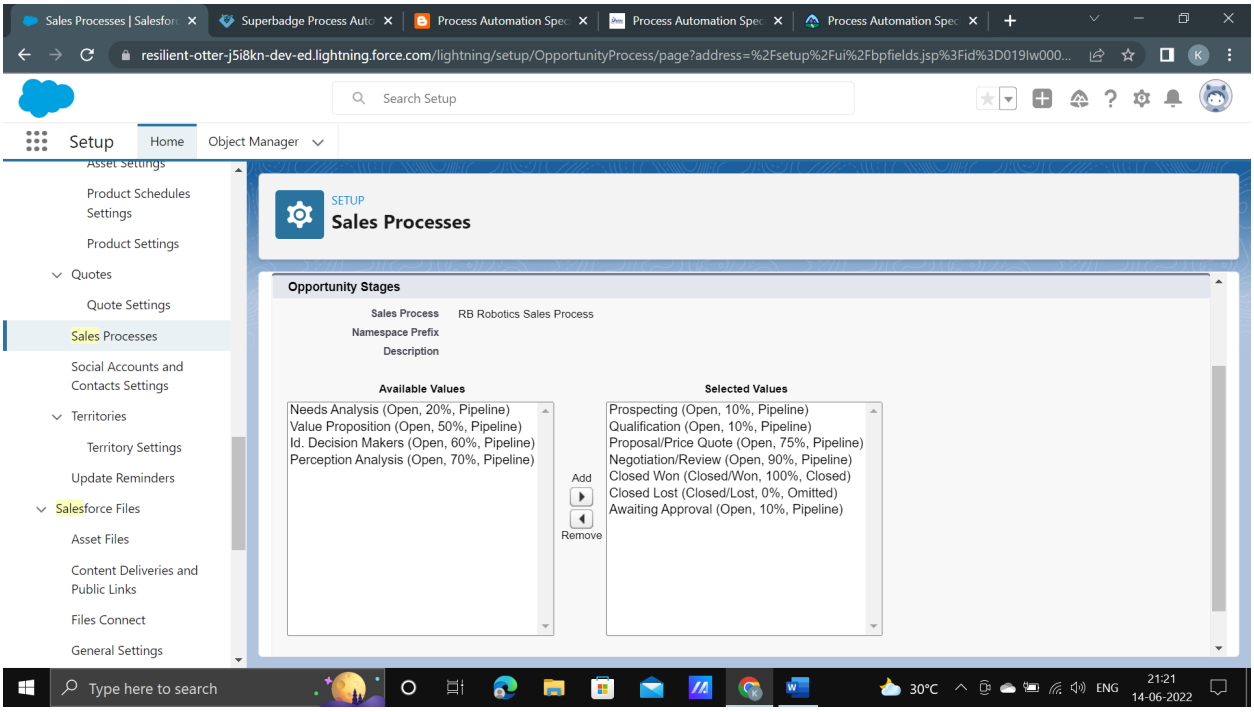
Step 2: Creating Validation rule on opportunity

Error Condition Formula:

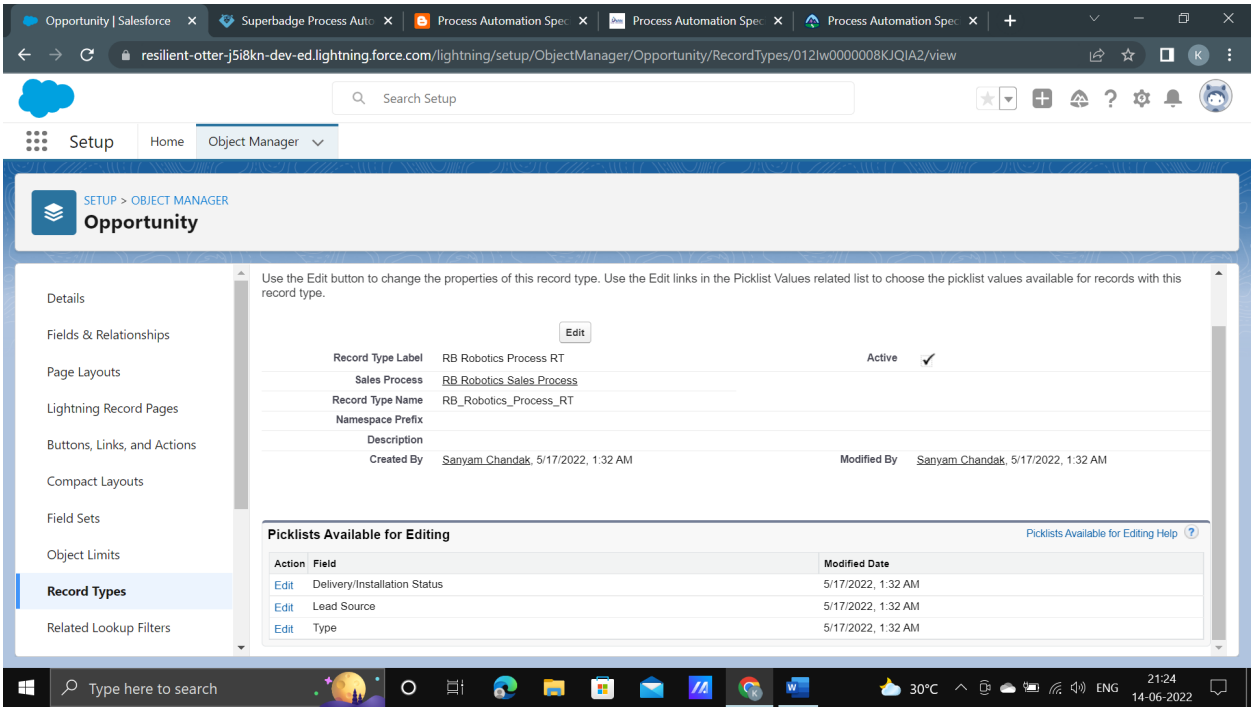
IF((Amount > 100000 && Approved__c <> True && ISPICKVAL(StageName,'Closed Won')),True,False)



Step 3: Creating sales process named "RB Robotics Sales Process"

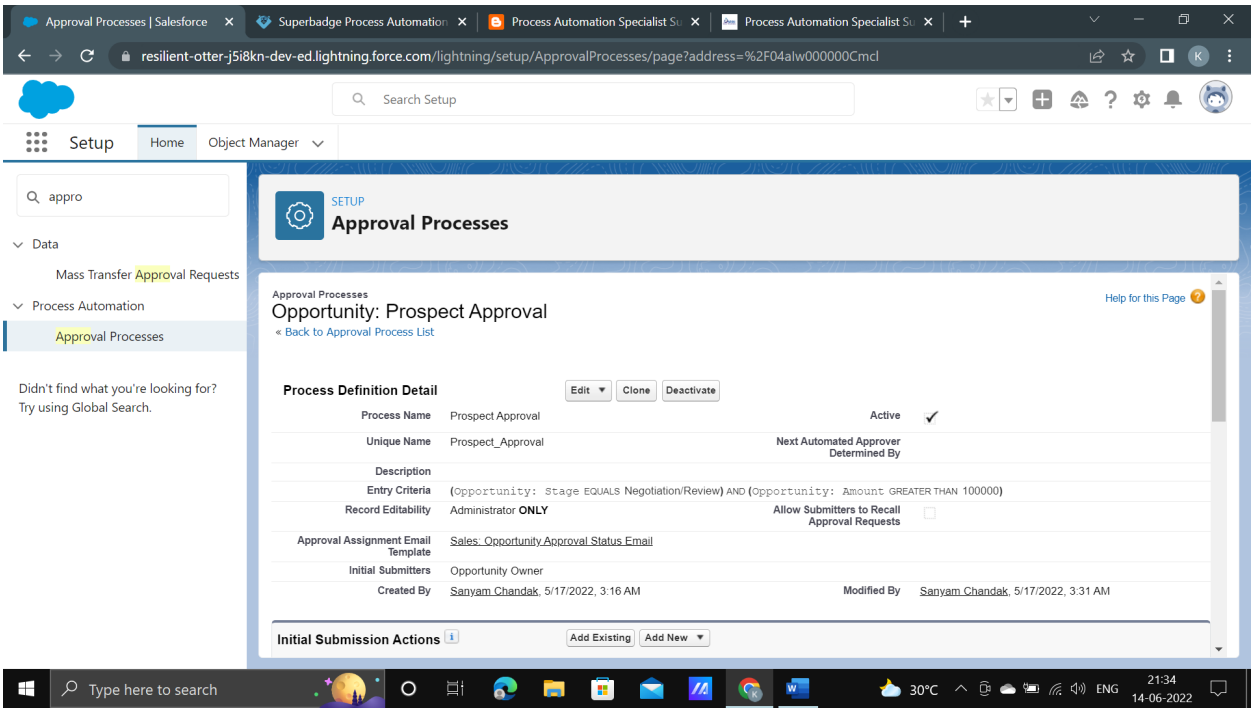


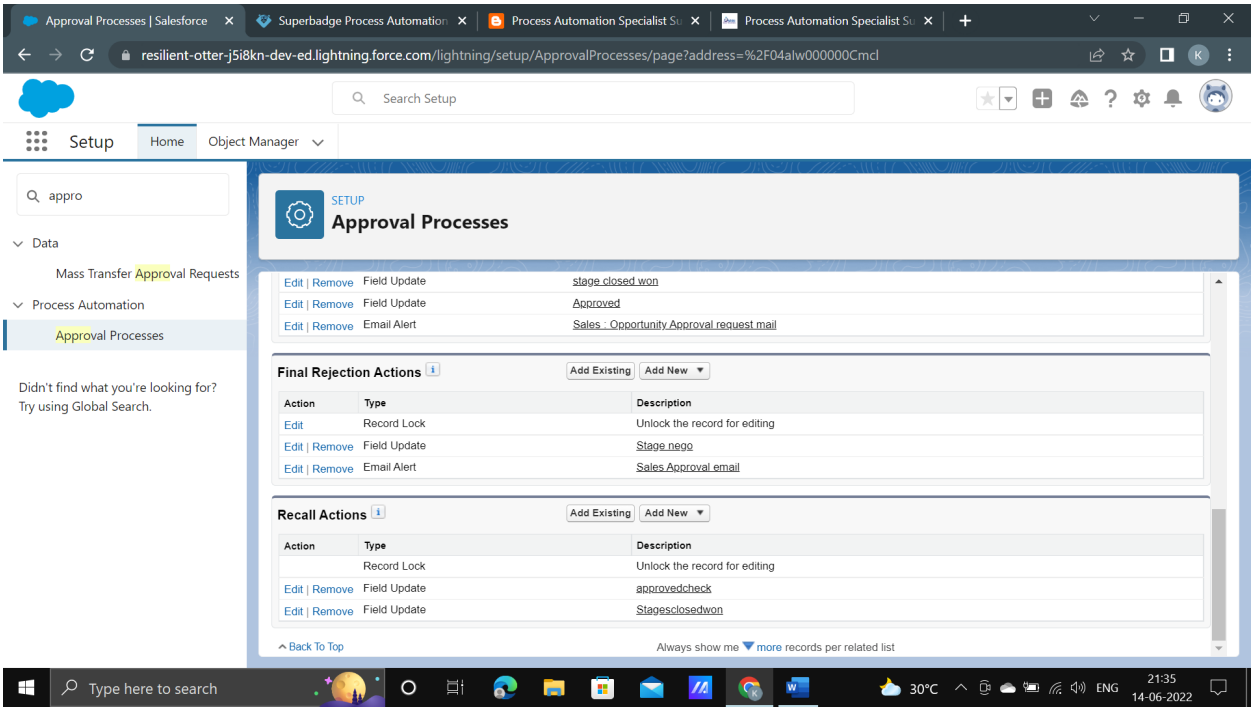
Step 4: Creating record type named "RB Robotics Process RT"



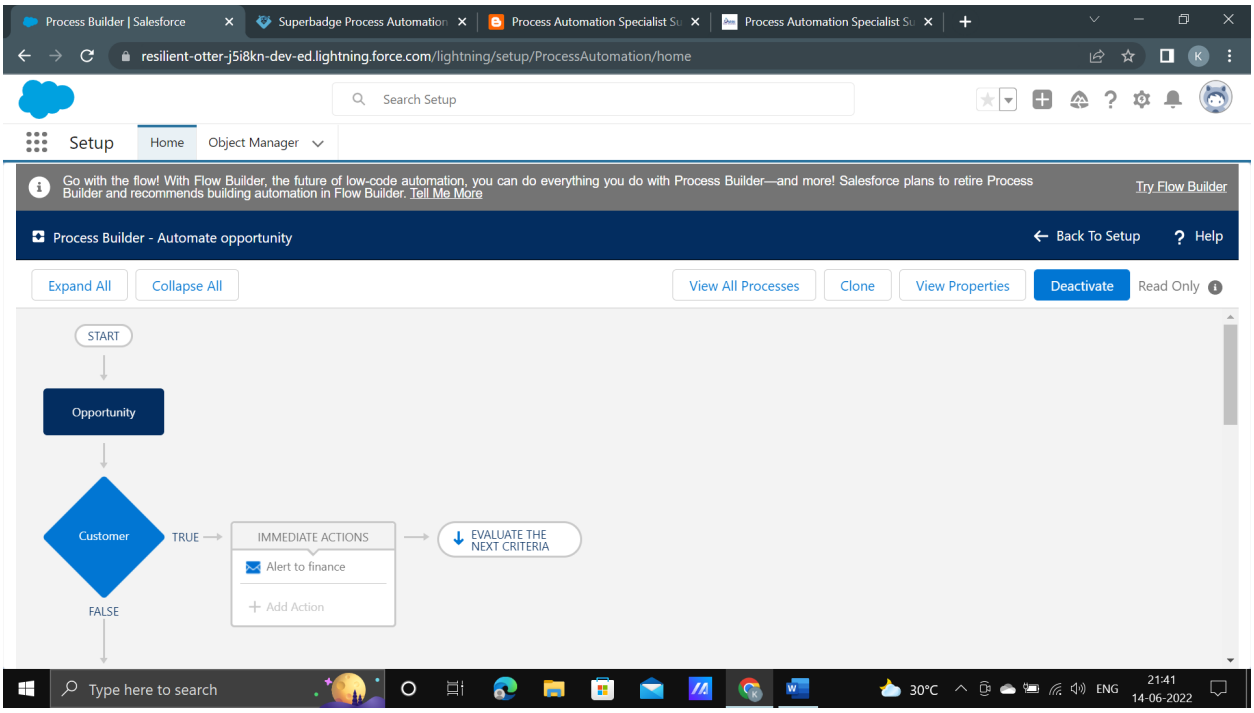
- **Challenge 5: Automate Opportunities**

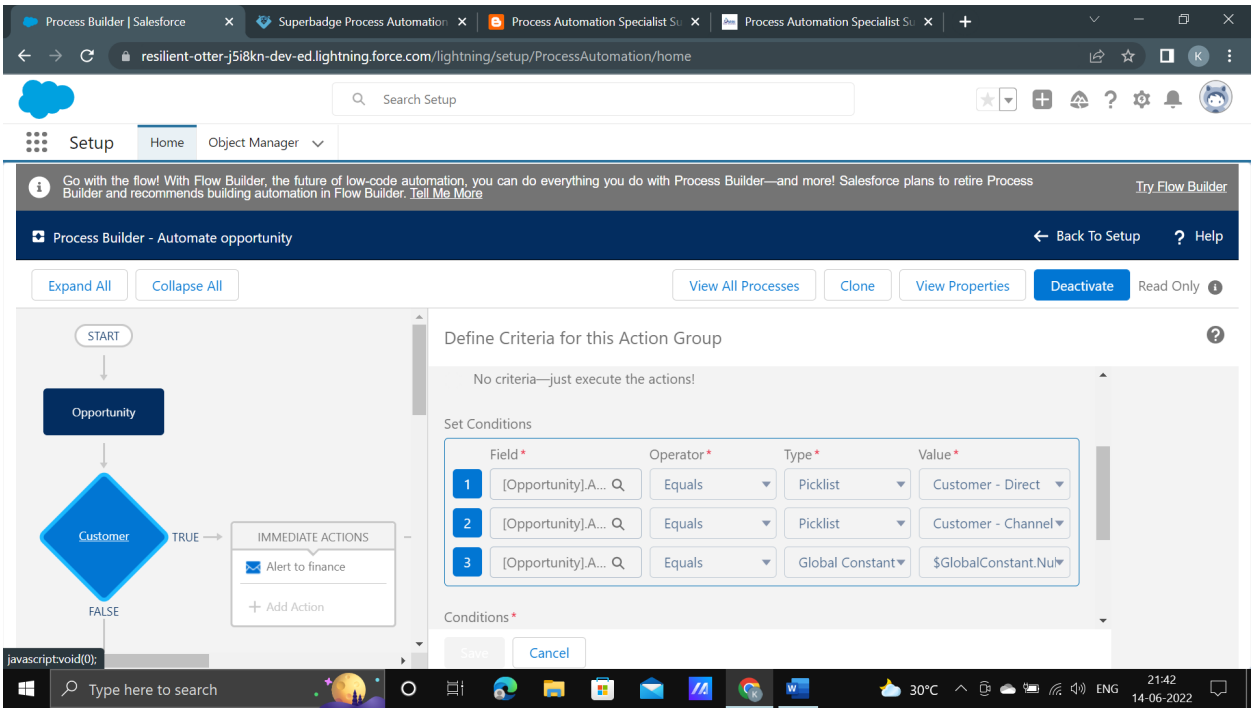
Step 1: Creating approval process named “Prospect Approval” on opportunity object.

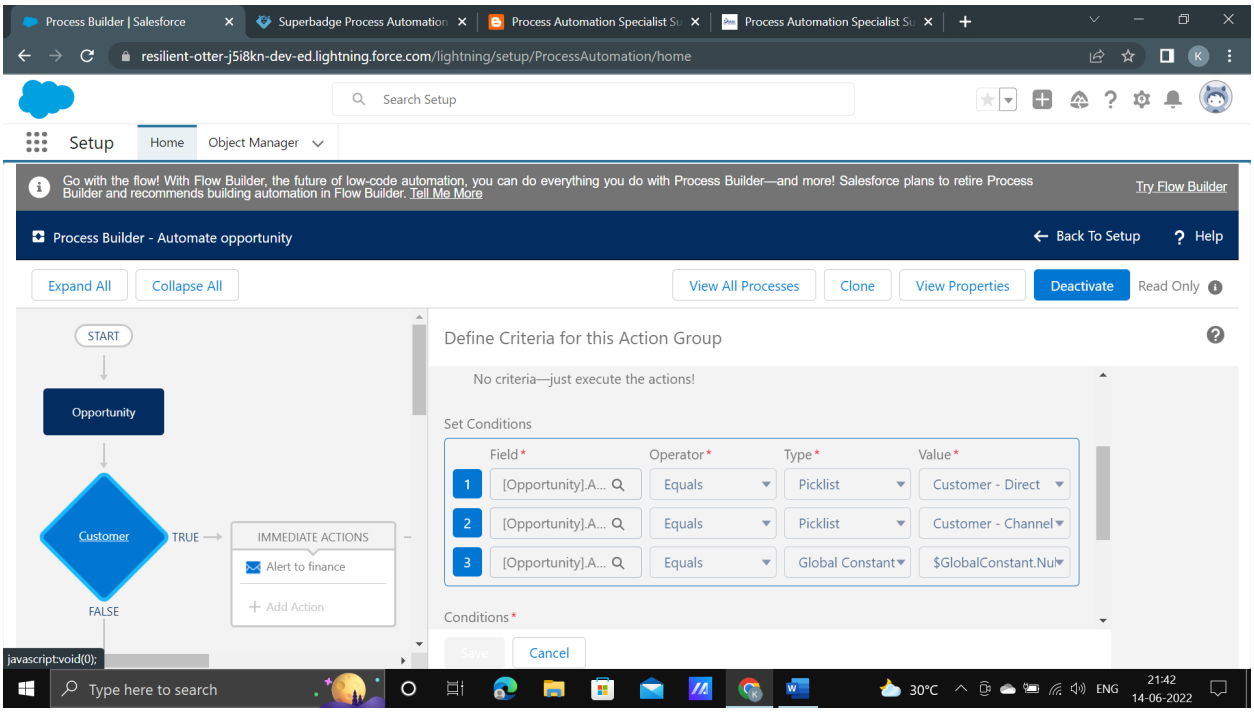

Approval Processes | Salesforce
Superbadge Process Automation
Process Automation Specialist Su
Process Automation Specialist Su
resilient-otter-j5i8kn-dev-ed.lightning.force.com/lightning/setup/ApprovalProcesses/page?address=%2F04alw000000Cmcl
Search Setup
Setup
Home
Object Manager
appro
Data
Mass Transfer Approval Requests
Process Automation
Approval Processes
Didn't find what you're looking for?
Try using Global Search.
SETUP
Approval Processes
Approval Processes
Opportunity: Prospect Approval
« Back to Approval Process List
Help for this Page
Process Definition Detail
Edit
Clone
Deactivate
Process Name
Prospect Approval
Active
Unique Name
Prospect_Approval
Next Automated Approver Determined By
Description
Entry Criteria
(Opportunity: Stage EQUALS Negotiation/Review) AND (Opportunity: Amount GREATER THAN 100000)
Record Editability
Administrator ONLY
Allow Submitters to Recall Approval Requests
Approval Assignment Email Template
Sales: Opportunity Approval Status Email
Initial Submitters
Opportunity Owner
Created By
Sanyam Chandak, 5/17/2022, 3:16 AM
Modified By
Sanyam Chandak, 5/17/2022, 3:31 AM
Initial Submission Actions
Add Existing
Add New
Type here to search
30°C
ENG
21:34
14-06-2022

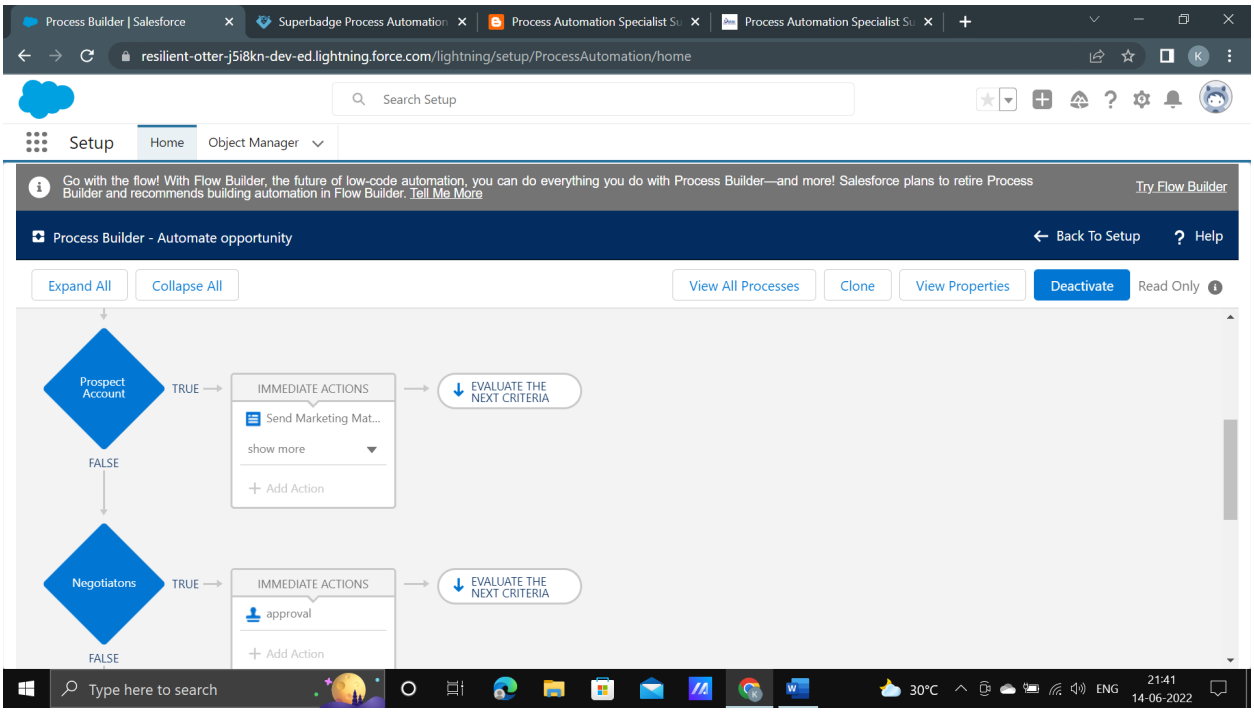

Approval Processes | Salesforce
Superbadge Process Automation
Process Automation Specialist Su
Process Automation Specialist Su
resilient-otter-j5i8kn-dev-ed.lightning.force.com/lightning/setup/ApprovalProcesses/page?address=%2F04alw000000Cmcl
Search Setup
Setup
Home
Object Manager
appro
Data
Mass Transfer Approval Requests
Process Automation
Approval Processes
Didn't find what you're looking for?
Try using Global Search.
SETUP
Approval Processes
Edit | Remove Field Update stage closed won
Edit | Remove Field Update Approved
Edit | Remove Email Alert Sales : Opportunity Approval request mail
Final Rejection Actions
Add Existing
Add New
Action
Type
Description
Edit Record Lock Unlock the record for editing
Edit | Remove Field Update Stage nego
Edit | Remove Email Alert Sales Approval email
Recall Actions
Add Existing
Add New
Action
Type
Description
Record Lock Unlock the record for editing
Edit | Remove Field Update approvedcheck
Edit | Remove Field Update Stagesclosedwon
Back To Top
Always show me more records per related list
Type here to search
30°C
ENG
21:35
14-06-2022

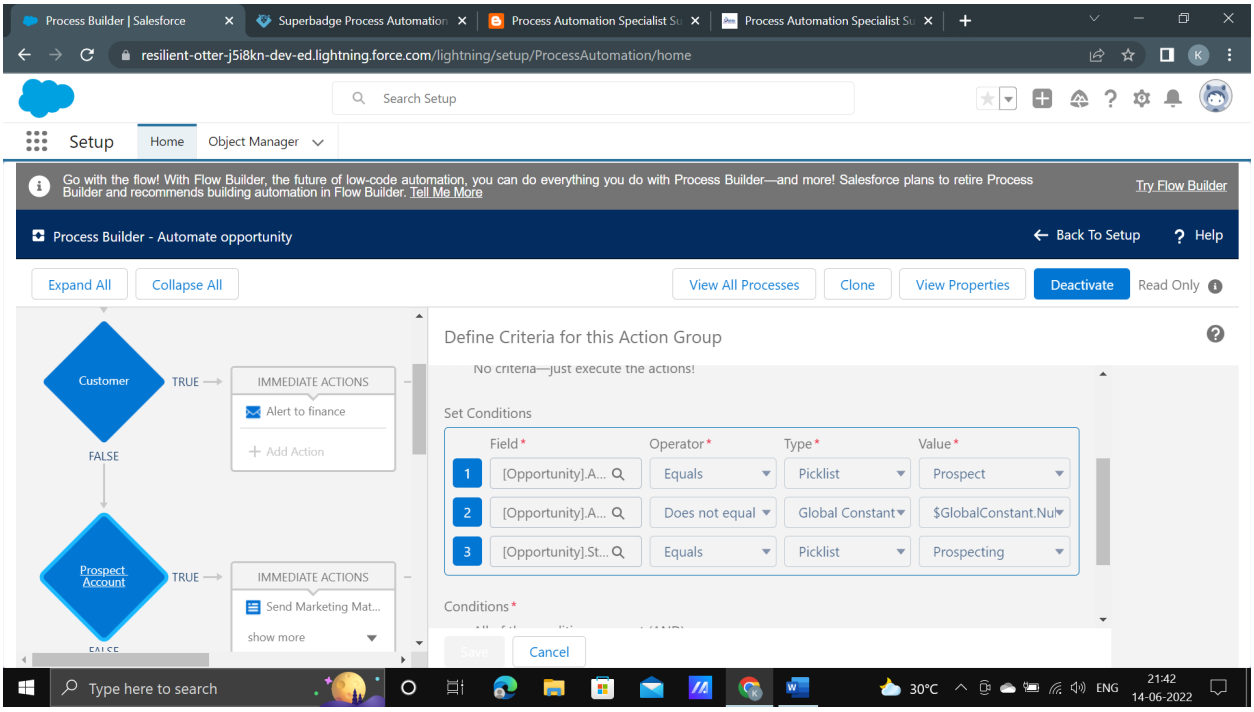
Step 2: Creating a process called “Automate opportunity” in process builder

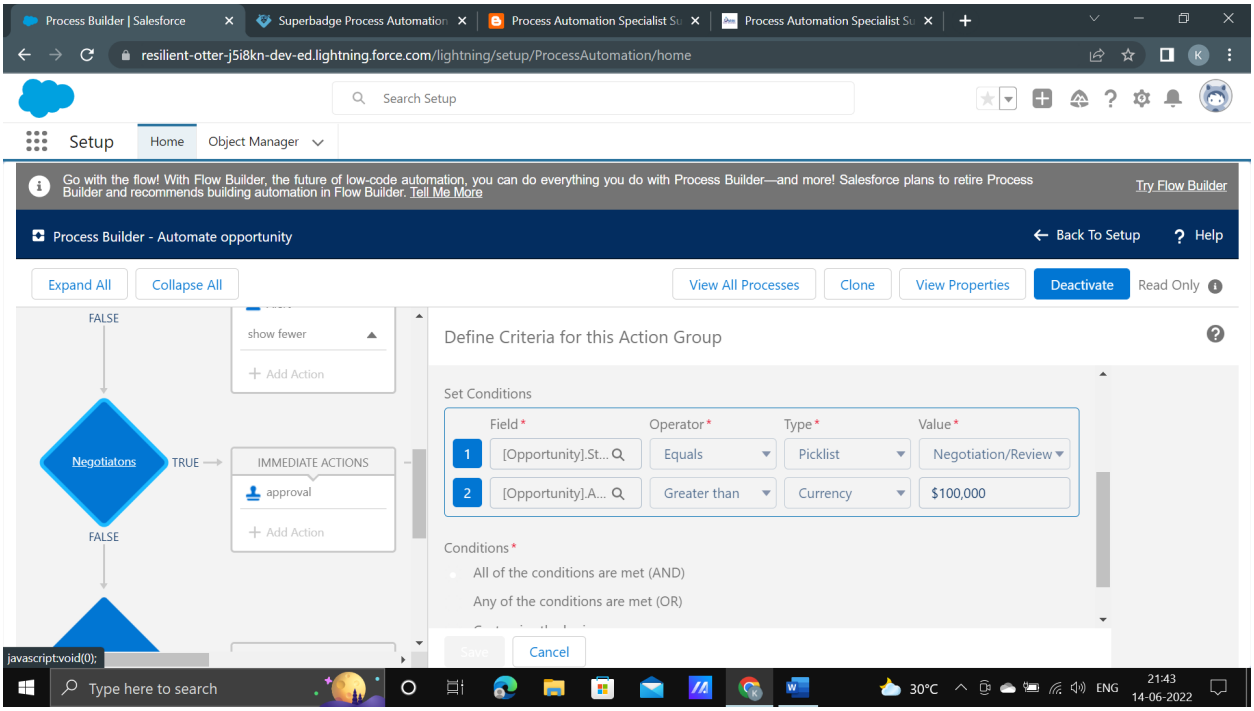


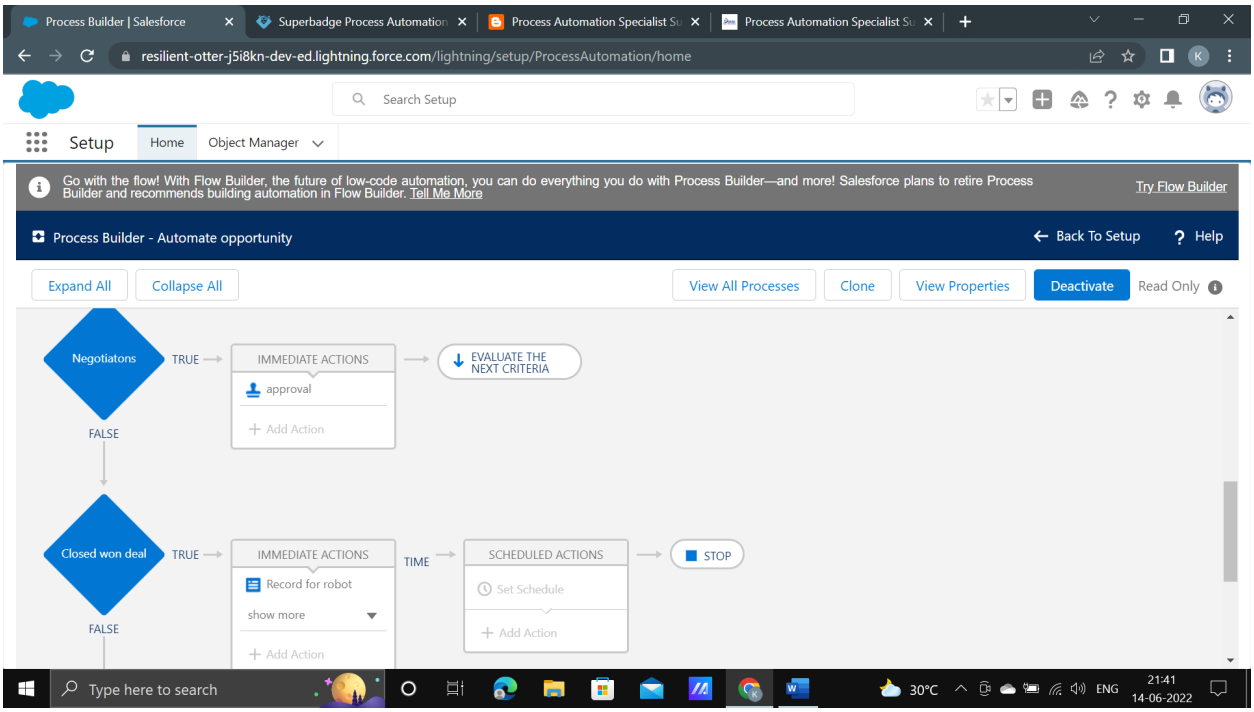

Process Builder | Salesforce
Superbadge Process Automation
Process Automation Specialist
Process Automation Specialist
resilient-otter-j5i8kn-dev-ed.lightning.force.com/lightning/setup/ProcessAutomation/home
Search Setup
Setup
Home
Object Manager
Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. Tell Me More
Try Flow Builder
Process Builder - Automate opportunity
Back To Setup
Help
Expand All
Collapse All
View All Processes
Clone
View Properties
Deactivate
Read Only
START
Opportunity
Customer
TRUE
FALSE
IMMEDIATE ACTIONS
Alert to finance
Add Action
Define Criteria for this Action Group
No criteria—just execute the actions!
Set Conditions
Field *
Operator *
Type *
Value *
1
[Opportunity].A...
Equals
Picklist
Customer - Direct
2
[Opportunity].A...
Equals
Picklist
Customer - Channel
3
[Opportunity].A...
Equals
Global Constant
$GlobalConstant.Nul
Conditions *
Save
Cancel
javascript:void(0);
Type here to search
30°C
ENG
21:42
14-06-2022

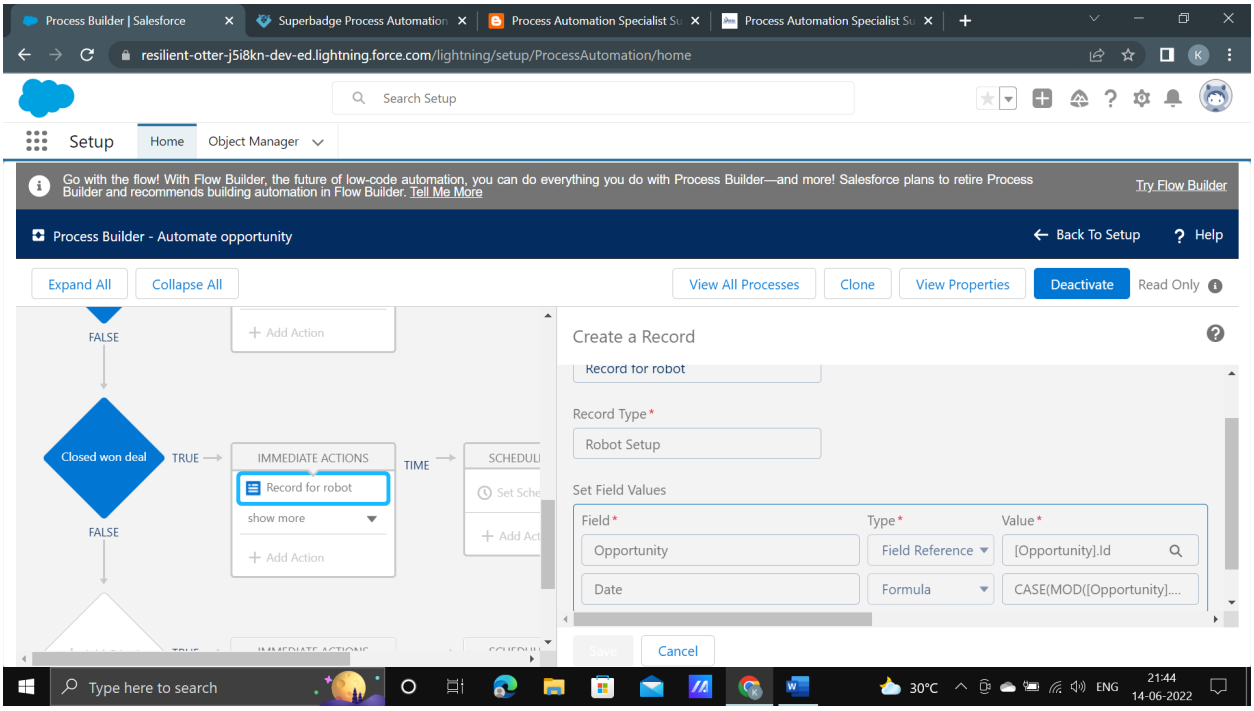

Process Builder | Salesforce
Superbadge Process Automation
Process Automation Specialist
Process Automation Specialist
resilient-otter-j5i8kn-dev-ed.lightning.force.com/lightning/setup/ProcessAutomation/home
Search Setup
Setup
Home
Object Manager
Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. Tell Me More
Try Flow Builder
Process Builder - Automate opportunity
Back To Setup
Help
Expand All
Collapse All
View All Processes
Clone
View Properties
Deactivate
Read Only
START
Opportunity
Customer
TRUE
FALSE
IMMEDIATE ACTIONS
Alert to finance
Add Action
Define Criteria for this Action Group
No criteria—just execute the actions!
Set Conditions
Field *
Operator *
Type *
Value *
1
[Opportunity].A...
Equals
Picklist
Customer - Direct
2
[Opportunity].A...
Equals
Picklist
Customer - Channel
3
[Opportunity].A...
Equals
Global Constant
$GlobalConstant.Nul
Conditions *
Save
Cancel
javascript:void(0);
Type here to search
30°C
ENG
21:42
14-06-2022

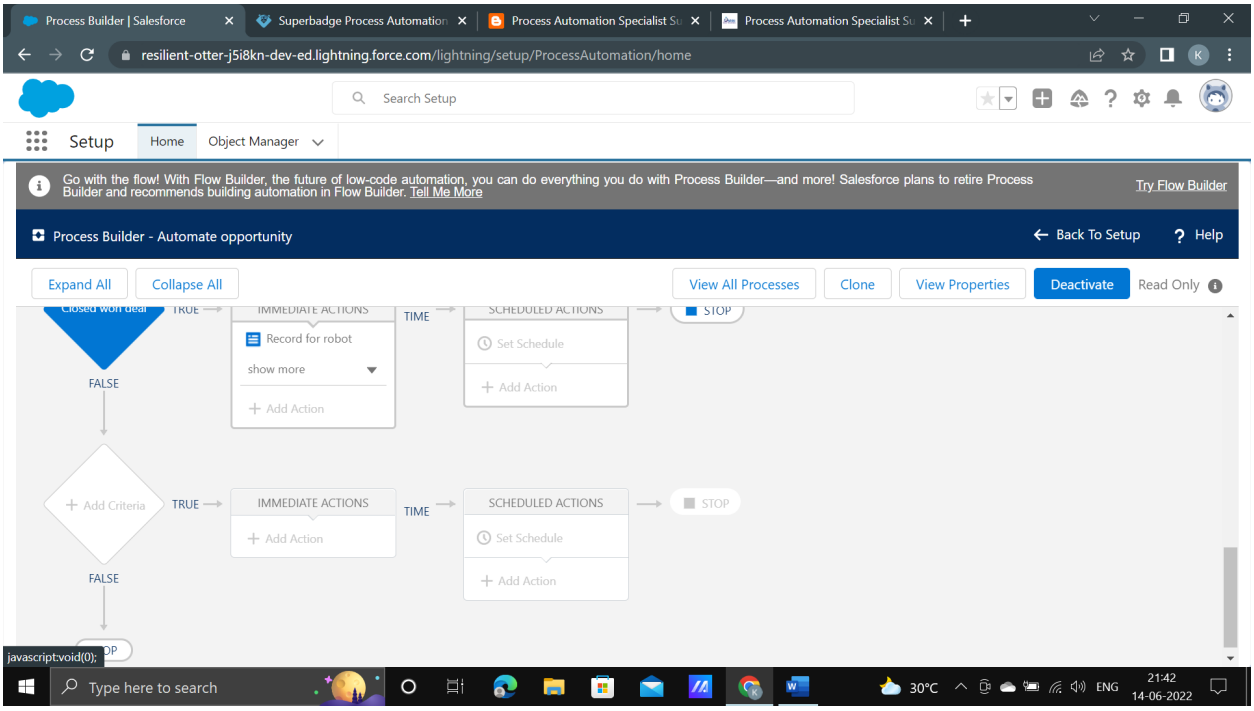

Process Builder | Salesforce
Superbadge Process Automation
Process Automation Specialist Su
Process Automation Specialist Su
resilient-otter-j5i8kn-dev-ed.lightning.force.com/lightning/setup/ProcessAutomation/home
Search Setup
Setup
Home
Object Manager
Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. Tell Me More
Try Flow Builder
Process Builder - Automate opportunity
Back To Setup
Help
Expand All
Collapse All
View All Processes
Clone
View Properties
Deactivate
Read Only
Prospect Account
TRUE
IMMEDIATE ACTIONS
Send Marketing Mat...
show more
Add Action
EVALUATE THE NEXT CRITERIA
FALSE
Negotiatons
TRUE
IMMEDIATE ACTIONS
approval
Add Action
EVALUATE THE NEXT CRITERIA
FALSE
Type here to search
30°C
ENG
21:41
14-06-2022


Process Builder | Salesforce
Superbadge Process Automation
Process Automation Specialist Su
Process Automation Specialist Su
resilient-otter-j5i8kn-dev-ed.lightning.force.com/lightning/setup/ProcessAutomation/home
Search Setup
Setup
Home
Object Manager
Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. Tell Me More
Try Flow Builder
Process Builder - Automate opportunity
Back To Setup
Help
Expand All
Collapse All
View All Processes
Clone
View Properties
Deactivate
Read Only
Customer
TRUE
IMMEDIATE ACTIONS
Alert to finance
Add Action
FALSE
Prospect Account
TRUE
IMMEDIATE ACTIONS
Send Marketing Mat...
show more
Define Criteria for this Action Group
No criteria—just execute the actions!
Set Conditions
Field *
Operator *
Type *
Value *
1
[Opportunity].A...
Equals
Picklist
Prospect
2
[Opportunity].A...
Does not equal
Global Constant
$GlobalConstant.Nul
3
[Opportunity].St...
Equals
Picklist
Prospecting
Conditions *
Save
Cancel
Type here to search
30°C
ENG
21:42
14-06-2022


Process Builder | Salesforce
Superbadge Process Automation
Process Automation Specialist Su
Process Automation Specialist Su
resilient-otter-j5i8kn-dev-ed.lightning.force.com/lightning/setup/ProcessAutomation/home
Search Setup
Setup
Home
Object Manager
Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. Tell Me More
Try Flow Builder
Process Builder - Automate opportunity
Back To Setup
Help
Expand All
Collapse All
View All Processes
Clone
View Properties
Deactivate
Read Only
FALSE
show fewer
Add Action
Negotiatons
TRUE
IMMEDIATE ACTIONS
approval
Add Action
FALSE
Define Criteria for this Action Group
Set Conditions
Field *
Operator *
Type *
Value *
1
[Opportunity].St...
Equals
Picklist
Negotiation/Review
2
[Opportunity].A...
Greater than
Currency
$100,000
Conditions *
All of the conditions are met (AND)
Any of the conditions are met (OR)
Save
Cancel
javascript:void(0);
Type here to search
30°C
ENG
21:43
14-06-2022

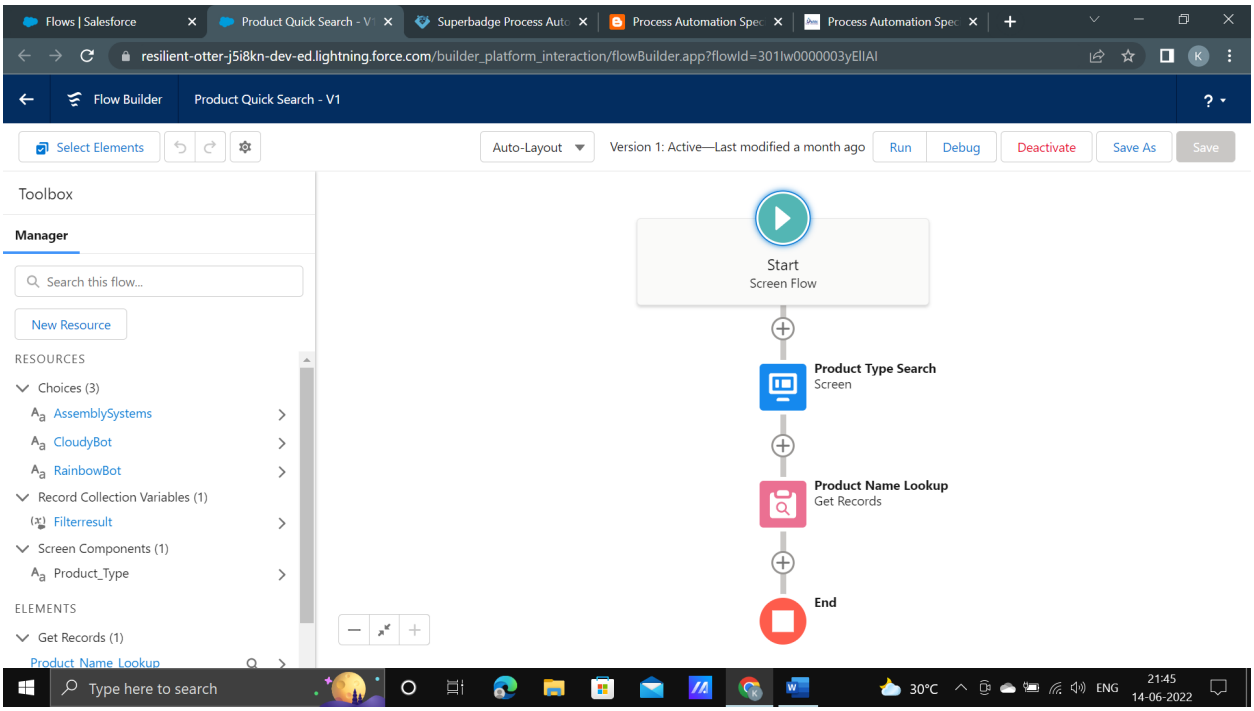

Process Builder | Salesforce
Superbadge Process Automation
Process Automation Specialist Su
Process Automation Specialist Su
resilient-otter-j5i8kn-dev-ed.lightning.force.com/lightning/setup/ProcessAutomation/home
Search Setup
Setup
Home
Object Manager
Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. Tell Me More
Try Flow Builder
Process Builder - Automate opportunity
Back To Setup
Help
Expand All
Collapse All
View All Processes
Clone
View Properties
Deactivate
Read Only
Negotiatons
TRUE
IMMEDIATE ACTIONS
approval
Add Action
EVALUATE THE NEXT CRITERIA
FALSE
Closed won deal
TRUE
IMMEDIATE ACTIONS
Record for robot
show more
Add Action
TIME
SCHEDULED ACTIONS
Set Schedule
Add Action
STOP
FALSE
Type here to search
30°C
ENG
21:41
14-06-2022

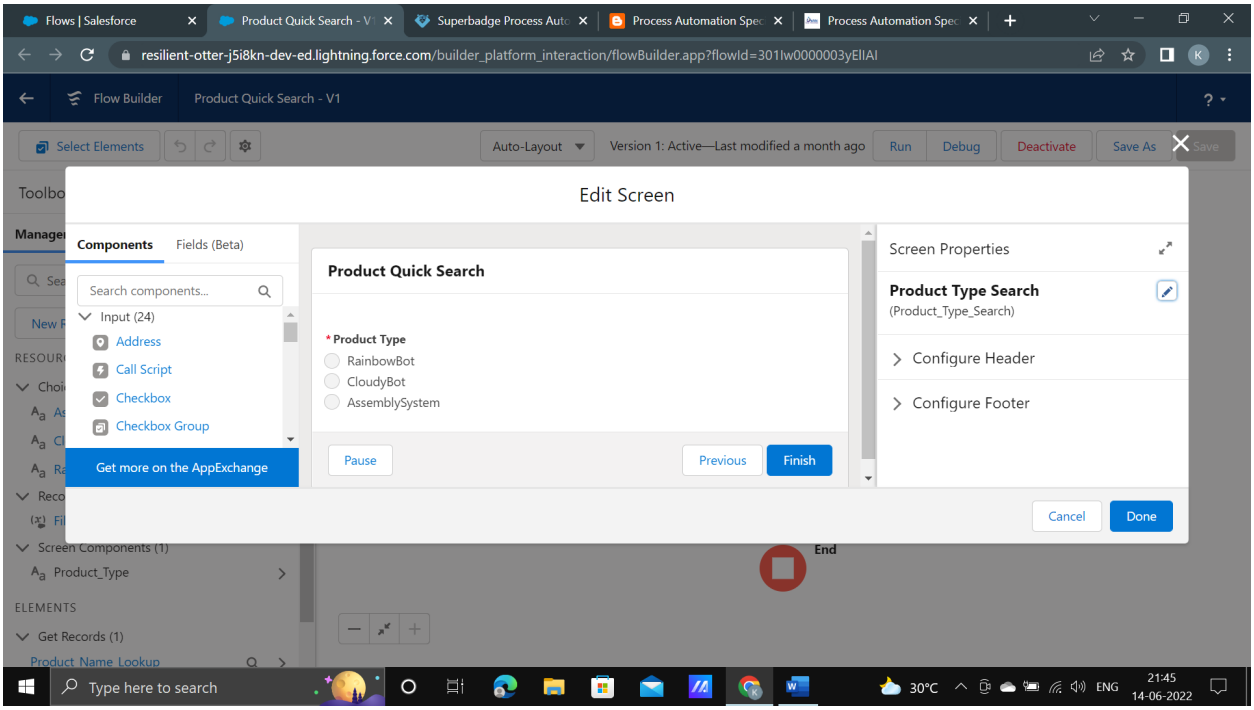


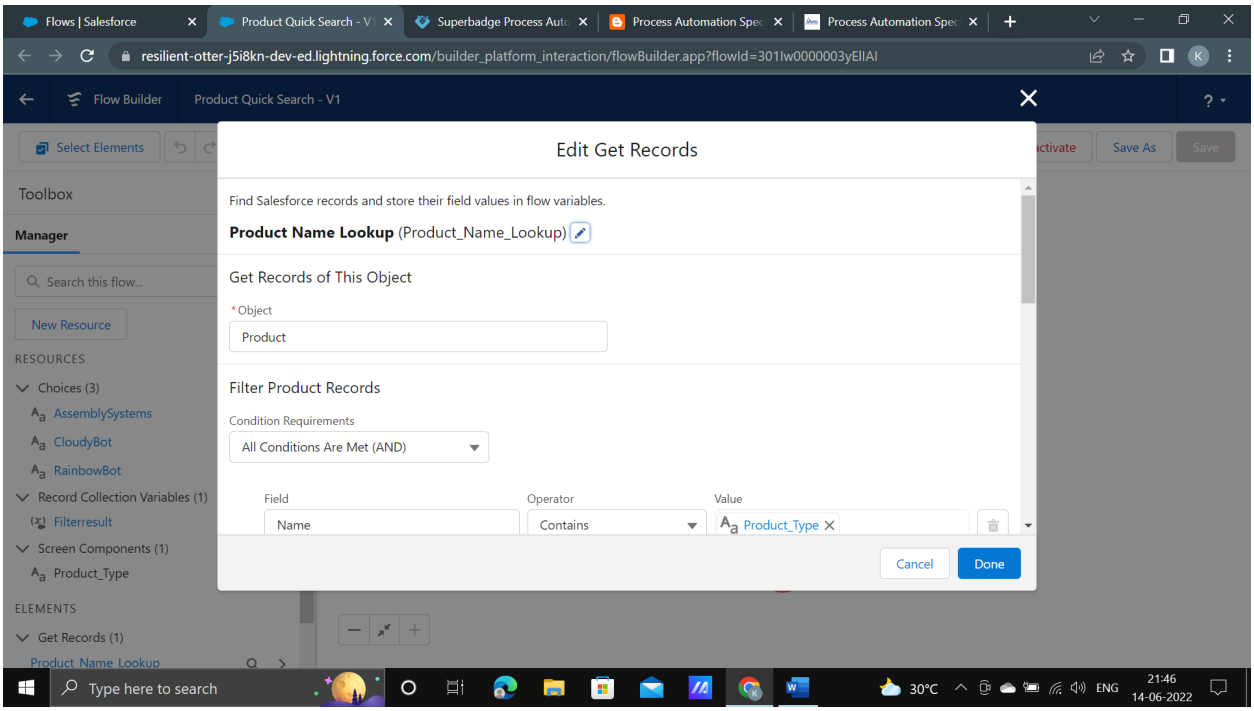


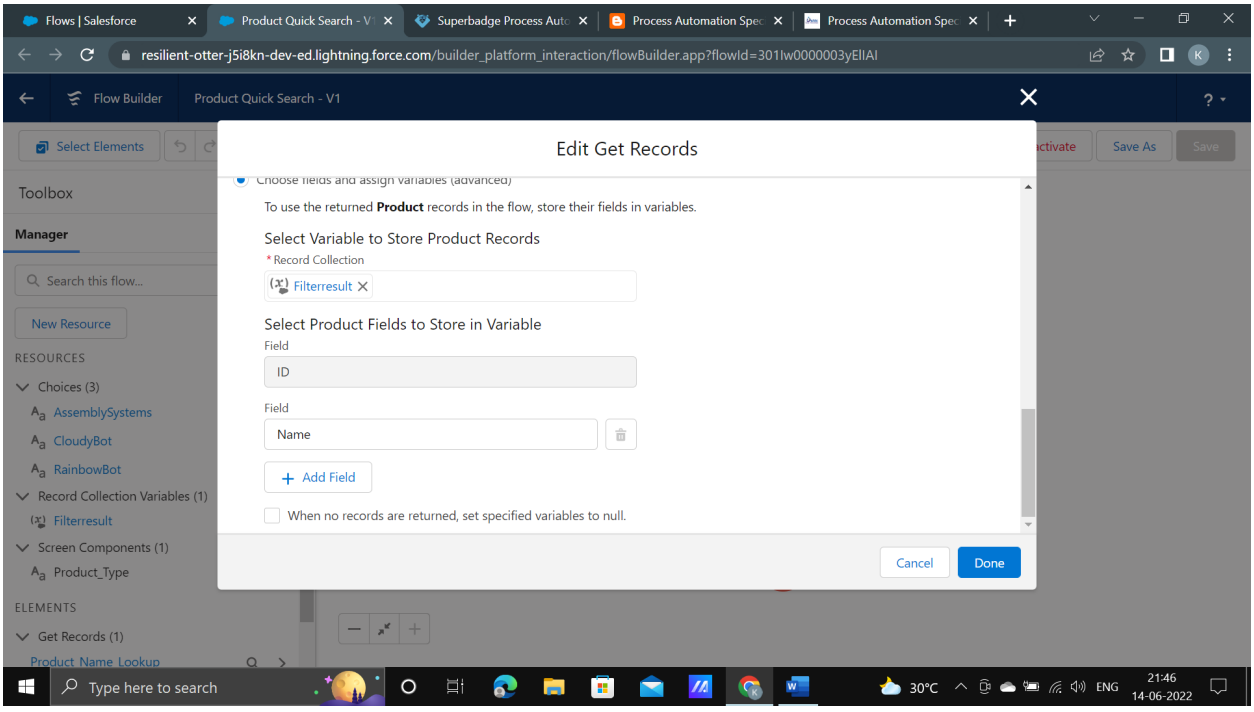
- **Challenge 6: Create Flow for Opportunities**

Step 1: Creating a flow named “Product Quick Search” in flow builder.





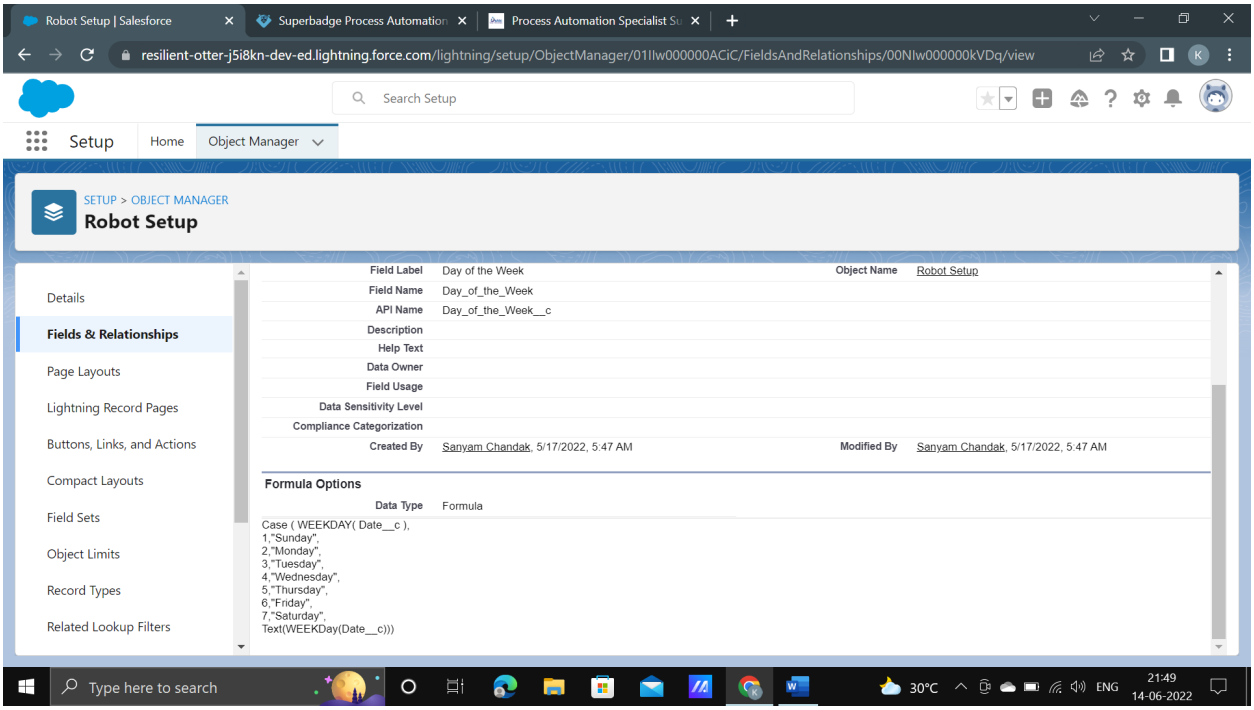




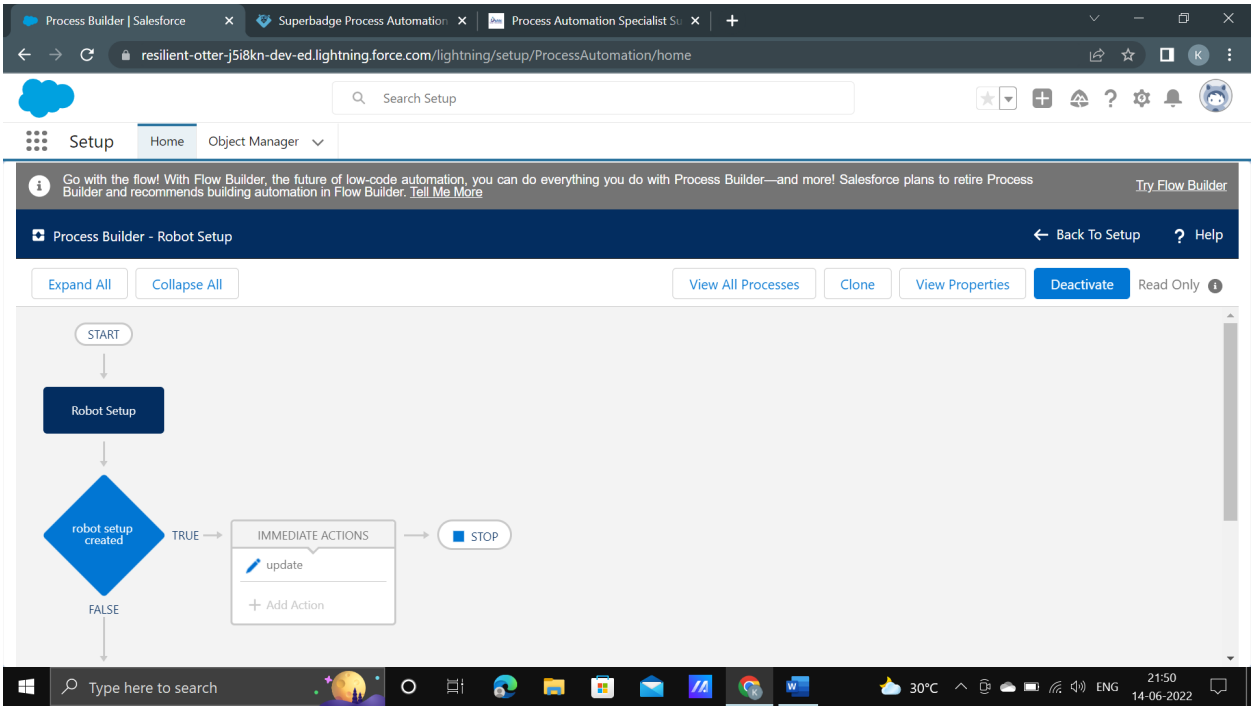
Step 2: Activating the flow and adding the flow to the opportunity screen using app builder.

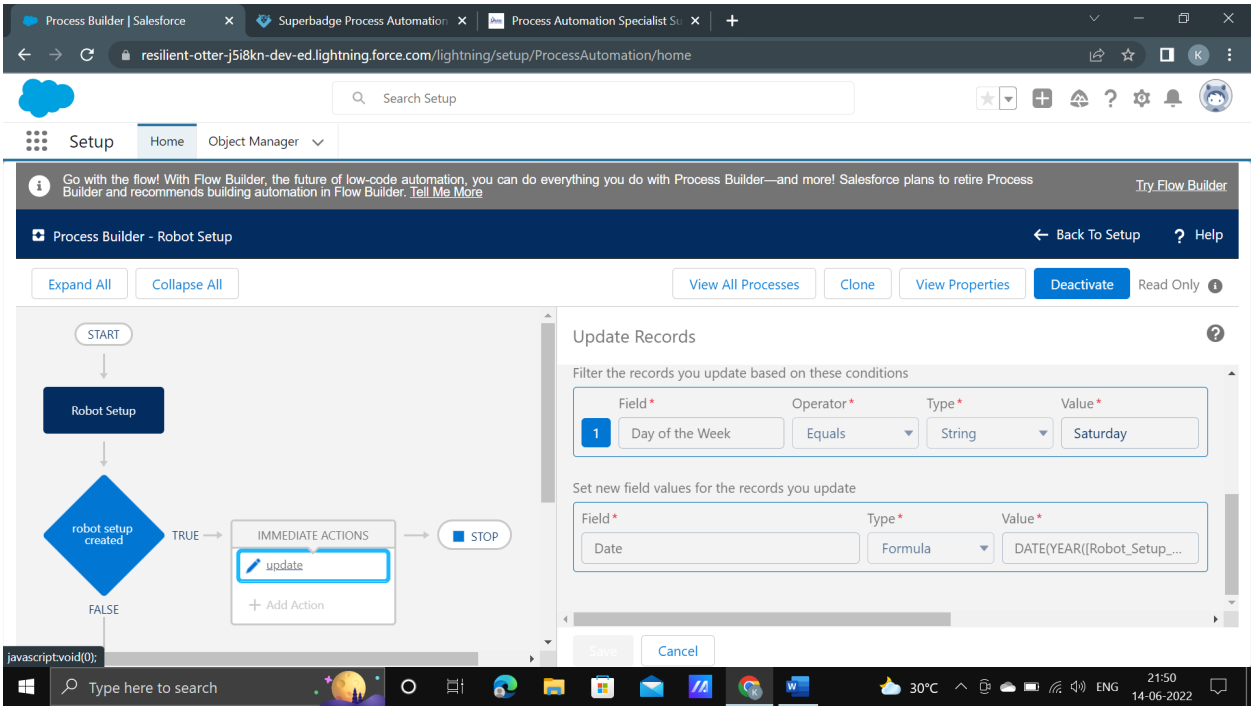
- ## Challenge 7: Automate Setups

Step 1: Changing the datatype for “Day of the week” field from TEXT to Formula (TEXT) and using the formula to get Day of the week.



Step 2: Creating process named “Robot Setup” in Process Builder.


Process Builder | Salesforce
Superbadge Process Automation
Process Automation Specialist
resilient-otter-j5i8kn-dev-ed.lightning.force.com/lightning/setup/ProcessAutomation/home
Search Setup
Setup
Home
Object Manager
Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. Tell Me More
Try Flow Builder
Process Builder - Robot Setup
Back To Setup
Help
Expand All
Collapse All
View All Processes
Clone
View Properties
Deactivate
Read Only
START
Robot Setup
robot setup created
TRUE
IMMEDIATE ACTIONS
update
Add Action
STOP
FALSE
Type here to search
30°C
ENG
21:50
14-06-2022


Process Builder | Salesforce
Superbadge Process Automation
Process Automation Specialist
resilient-otter-j5i8kn-dev-ed.lightning.force.com/lightning/setup/ProcessAutomation/home
Search Setup
Setup
Home
Object Manager
Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. Tell Me More
Try Flow Builder
Process Builder - Robot Setup
Back To Setup
Help
Expand All
Collapse All
View All Processes
Clone
View Properties
Deactivate
Read Only
START
Robot Setup
robot setup created
TRUE
IMMEDIATE ACTIONS
update
Add Action
STOP
FALSE
Update Records
Filter the records you update based on these conditions
Field *
Operator *
Type *
Value *
1
Day of the Week
Equals
String
Saturday
Set new field values for the records you update
Field *
Type *
Value *
Date
Formula
DATE(YEAR([Robot_Setup_...
Cancel
javascript:void(0);
Type here to search
30°C
ENG
21:50
14-06-2022