SALESFORCE DEVELOPER CATALYST SELF LEARNING SUPER BADGES

TRAILHEAD LINK: https://trailblazer.me/id/pradosh0414

APEX SPECIALIST SUPERBADGE LINK -
https://trailhead.salesforce.com/content/learn/superbadges/superbadge_apex

CHALLENGE - 1 : QUIZ
CHALLENGE - 2 : AUTOMATE RECORD CREATION

CODE USED FOR THIS CHALLENGE:

MaintenanceRequestHelper Apex Class:

```
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()){
            Map<Id,Case> closedCases = 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'));
          }

          List<Case> newCases = new List<Case>();
          for(Case cc : closedCases.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));
             } else {
                nc.Date_Due__c = Date.today().addDays((Integer)
cc.Equipment__r.maintenance_Cycle__c);
             }

             newCases.add(nc);
          }

          insert newCases;

          List<Equipment_Maintenance_Item__c> clonedList = new
List<Equipment_Maintenance_Item__c>();
          for (Case nc : newCases){
             for (Equipment_Maintenance_Item__c clonedListItem :
closedCases.get(nc.ParentId).Equipment_Maintenance_Items__r){
```

```
                Equipment_Maintenance_Item__c item = clonedListItem.clone();
                item.Maintenance_Request__c = nc.Id;
                clonedList.add(item);
            }
        }
        insert clonedList;
    }
  }
}
```

MaintenanceRequest Apex Trigger :

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

CHALLENGE - 3 : SYNCHRONIZE SALESFORCE DATA WITH AN EXTERNAL SYSTEM

CODE USED FOR THIS CHALLENGE :

WarehouseCalloutService Apex Class:

```
public with sharing class WarehouseCalloutService implements Queueable {
   private static final String WAREHOUSE_URL = 'https://th-superbadge-
apex.herokuapp.com/equipment';

   @future(callout=true)
   public static void runWarehouseEquipmentSync(){
      System.debug('go into runWarehouseEquipmentSync');
      Http http = new Http();
      HttpRequest request = new HttpRequest();

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

      List<Product2> product2List = new List<Product2>();
      System.debug(response.getStatusCode());
      if (response.getStatusCode() == 200){
         List<Object> jsonResponse =
(List<Object>)JSON.deserializeUntyped(response.getBody());
         System.debug(response.getBody());

         for (Object jR : jsonResponse){
            Map<String,Object> mapJson = (Map<String,Object>)jR;
            Product2 product2 = new Product2();
            //replacement part (always true),
            product2.Replacement_Part__c = (Boolean) mapJson.get('replacement');
            //cost
            product2.Cost__c = (Integer) mapJson.get('cost');
            //current inventory
            product2.Current_Inventory__c = (Double) mapJson.get('quantity');
            //lifespan
            product2.Lifespan_Months__c = (Integer) mapJson.get('lifespan');
            //maintenance cycle
            product2.Maintenance_Cycle__c = (Integer)
mapJson.get('maintenanceperiod');
            //warehouse SKU
            product2.Warehouse_SKU__c = (String) mapJson.get('sku');

            product2.Name = (String) mapJson.get('name');
            product2.ProductCode = (String) mapJson.get('_id');
            product2List.add(product2);
         }

         if (product2List.size() > 0){
            upsert product2List;
            System.debug('Your equipment was synced with the warehouse one');
         }
      }
   }

   public static void execute (QueueableContext context){
      System.debug('start runWarehouseEquipmentSync');
      runWarehouseEquipmentSync();
      System.debug('end runWarehouseEquipmentSync');
   }

}
```

CHALLENGE - 4 : SCHEDULE SYNCHRONIZATION

CODE USED FOR THIS CHALLENGE :

WarehouseSyncSchedule Apex Class:

```
global with sharing class WarehouseSyncSchedule implements Schedulable{
   global void execute(SchedulableContext ctx){
      System.enqueueJob(new WarehouseCalloutService());
   }
}
```

CHALLENGE - 5 : TEST AUTOMATION LOGIC

CODE USED FOR THIS CHALLENGE :

MaintenanceRequestHelperTest Apex Class:

```
@isTest
public with sharing class MMJhwmfTYJVFm9TiUdmYYyrf6Pzpzf3YN1 {

   // createVehicle
   private static Vehicle__c createVehicle(){
      Vehicle__c vehicle = new Vehicle__C(name = 'Testing Vehicle');
      return vehicle;
   }

   // createEquipment
   private static Product2 createEquipment(){
      product2 equipment = new product2(name = 'Testing equipment',
                                  lifespan_months__c = 10,
                                  maintenance_cycle__c = 10,
                                  replacement_part__c = true);
      return equipment;
   }

   // createMaintenanceRequest
   private static Case createMaintenanceRequest(id vehicleId, id equipmentId){
      case cse = new case(Type='Repair',
                     Status='New',
                    Origin='Web',
                    Subject='Testing subject',
                    Equipment__c=equipmentId,
                    Vehicle__c=vehicleId);
        return cse;
    }

    // createEquipmentMaintenanceItem
    private static Equipment_Maintenance_Item__c createEquipmentMaintenanceItem(id
equipmentId,id requestId){
        Equipment_Maintenance_Item__c equipmentMaintenanceItem = new
Equipment_Maintenance_Item__c(
            Equipment__c = equipmentId,
            Maintenance_Request__c = requestId);
        return equipmentMaintenanceItem;
    }

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

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

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

        Equipment_Maintenance_Item__c equipmentMaintenanceItem =
createEquipmentMaintenanceItem(equipmentId,createdCase.id);
        insert equipmentMaintenanceItem;

        test.startTest();
        createdCase.status = 'Closed';
        update createdCase;
    test.stopTest();

    Case newCase = [Select id,
                    subject,
                    type,
                    Equipment__c,
                    Date_Reported__c,
                    Vehicle__c,
                    Date_Due__c
                   from case
                   where status ='New'];

    Equipment_Maintenance_Item__c workPart = [select id
                                              from Equipment_Maintenance_Item__c
                                              where Maintenance_Request__c =:newCase.Id];
    list<case> allCase = [select id from case];
    system.assert(allCase.size() == 2);

    system.assert(newCase != null);
    system.assert(newCase.Subject != null);
    system.assertEquals(newCase.Type, 'Routine Maintenance');
    SYSTEM.assertEquals(newCase.Equipment__c, equipmentId);
    SYSTEM.assertEquals(newCase.Vehicle__c, vehicleId);
    SYSTEM.assertEquals(newCase.Date_Reported__c, system.today());
}

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

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

    case createdCase = createMaintenanceRequest(vehicleId,equipmentId);
```

```
        insert createdCase;

        Equipment_Maintenance_Item__c workP =
createEquipmentMaintenanceItem(equipmentId, createdCase.Id);
        insert workP;

        test.startTest();
        createdCase.Status = 'Working';
        update createdCase;
        test.stopTest();

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

        Equipment_Maintenance_Item__c equipmentMaintenanceItem = [select id
                                                  from Equipment_Maintenance_Item__c
                                                  where Maintenance_Request__c = :createdCase.Id];

        system.assert(equipmentMaintenanceItem != null);
        system.assert(allCase.size() == 1);
    }

    @isTest
    private static void testBulk(){
        list<Vehicle__C> vehicleList = new list<Vehicle__C>();
        list<Product2> equipmentList = new list<Product2>();
        list<Equipment_Maintenance_Item__c> equipmentMaintenanceItemList = new
list<Equipment_Maintenance_Item__c>();
        list<case> caseList = new list<case>();
        list<id> oldCaseIds = new list<id>();

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

        for(integer i = 0; i < 300; i++){

equipmentMaintenanceItemList.add(createEquipmentMaintenanceItem(equipmentList.
get(i).id, caseList.get(i).id));
        }
        insert equipmentMaintenanceItemList;

        test.startTest();
        for(case cs : caseList){
           cs.Status = 'Closed';
           oldCaseIds.add(cs.Id);
        }
        update caseList;
        test.stopTest();

        list<case> newCase = [select id
                              from case
                              where status ='New'];



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

        system.assert(newCase.size() == 300);

        list<case> allCase = [select id from case];
        system.assert(allCase.size() == 600);
    }
}
```

CHALLENGE - 6 : TEST CALLOUT LOGIC

CODE USED FOR THIS CHALLENGE:

WarehouseCalloutServiceMock Apex Class:

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

        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"},{"_id":"55d66226
726b611100aaf742","replacement":true,"quantity":183,"name":"Cooling
Fan","maintenanceperiod":0,"lifespan":0,"cost":300,"sku":"100004"},{"_id":"55d66226726b6
11100aaf743","replacement":true,"quantity":143,"name":"Fuse
20A","maintenanceperiod":0,"lifespan":0,"cost":22,"sku":"100005"}]');
        response.setStatusCode(200);

        return response;
    }
}
```

WarehouseCalloutServiceTest Apex Class:

```
@IsTest
```

```
private class WarehouseCalloutServiceTest {
   // implement your mock callout test here
        @isTest
   static void testWarehouseCallout() {
      test.startTest();
      test.setMock(HttpCalloutMock.class, new WarehouseCalloutServiceMock());
      WarehouseCalloutService.execute(null);
      test.stopTest();

      List<Product2> product2List = new List<Product2>();
      product2List = [SELECT ProductCode FROM Product2];

      System.assertEquals(3, product2List.size());
      System.assertEquals('55d66226726b611100aaf741',
product2List.get(0).ProductCode);
      System.assertEquals('55d66226726b611100aaf742',
product2List.get(1).ProductCode);
      System.assertEquals('55d66226726b611100aaf743',
product2List.get(2).ProductCode);
   }
}
```

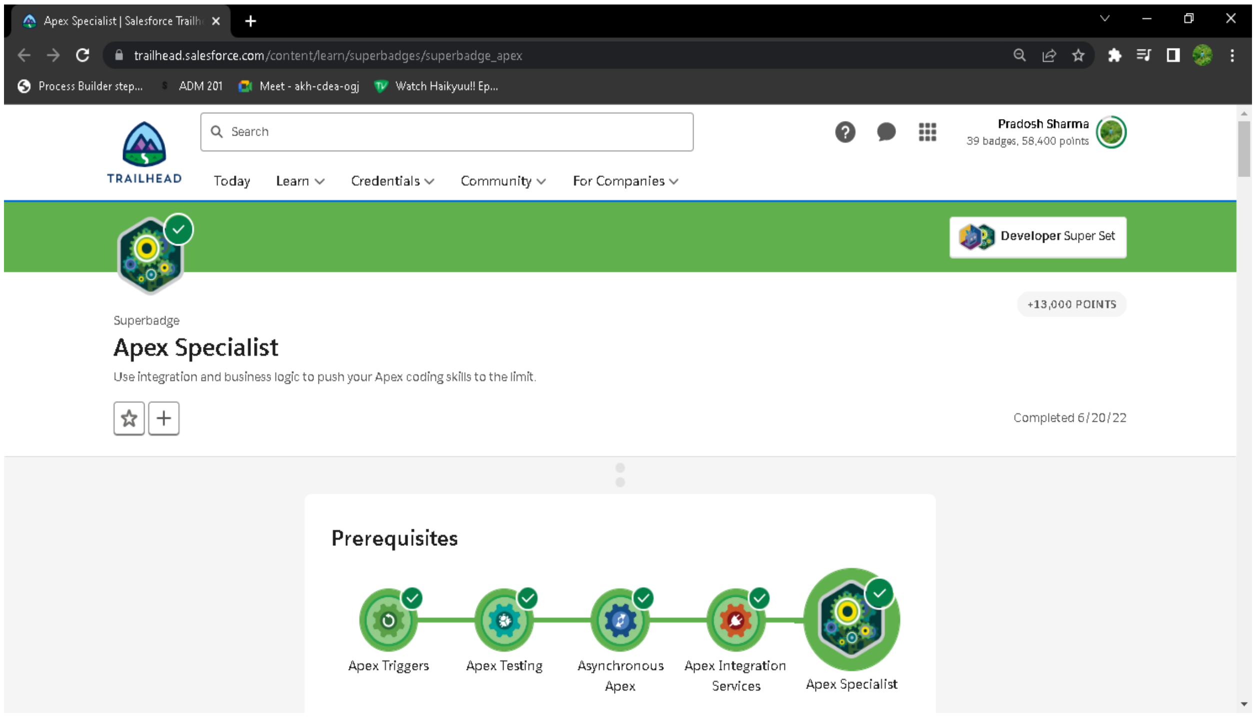
CHALLENGE - 7 : TEST SCHEDULING LOGIC

CODE USED FOR THIS CHALLENGE:

<u>WarehouseSyncScheduleTest Apex Class</u>:

```
@isTest
public with sharing class WarehouseSyncScheduleTest {
   // implement scheduled code here
   //
   @isTest static void test() {
      String scheduleTime = '00 00 00 * * ? *';
      Test.startTest();
      Test.setMock(HttpCalloutMock.class, new WarehouseCalloutServiceMock());
      String jobId = System.schedule('Warehouse Time to Schedule to test',
scheduleTime, new WarehouseSyncSchedule());
        CronTrigger c = [SELECT State FROM CronTrigger WHERE Id =: jobId];
        System.assertEquals('WAITING', String.valueOf(c.State), 'JobId does not match');

        Test.stopTest();
    }
}
```


Apex Specialist | Salesforce Trailh
trailhead.salesforce.com/content/learn/superbadges/superbadge_apex
Process Builder step... ADM 201 Meet - akh-cdea-ogj Watch Haikyuu!! Ep...
Search
Pradosh Sharma
39 badges, 58,400 points
TRAILHEAD
Today Learn Credentials Community For Companies
Developer Super Set
+13,000 POINTS
Superbadge
Apex Specialist
Use integration and business logic to push your Apex coding skills to the limit.
Completed 6/20/22
Prerequisites
Apex Triggers — Apex Testing — Asynchronous Apex — Apex Integration Services — Apex Specialist

PROCESS AUTOMATION SPECIALIST SUPERBADGE LINK - https://trailhead.salesforce.com/content/learn/superbadges/superbadge_process_automation

Challenge 1

Validation Rule

a. Check the function forLength.
b. Remember to check the NULL Values in Validation rule.

Queue Creation

c. This is straightforward normal Queue creation
d. Create Names with related to appropriate sales team.

Assignment Rule

e. Create new Assignment rule for this scenario(Do not use the standard rule).
f. Make sure that you rule is Active before you validate this step.

Challenge 2

Field Creations on Account Object

g. **Number of deals** Field should be a Roll-Up Summary take count of COUNT Opportunities
h. **Number of won deals** Field should be a Roll-Up Summary (COUNT Opportunity) with filter criteria of Closed Won
i. **Amount of won deals** Field should be a Roll-Up Summary (SUM Opportunity) with filter criteria of Closed Won
j. **Last won deal date** Field should be a Roll-Up Summary (MAX Opportunity)
k. **Deal win percent** Field should be a Formula(Percentage field) IF Number_of_deals c greater than 0 the , Number_of_won_deals c
/Number_of_dealsc otherwise Zero

l. **Call for Service** Field should be a Formula (Date)
IF(OR(TODAY() – 730 > Last_won_deal_date c , TODAY() + 7 < Last_won_deal_date c), 'Yes','No')

Validation Rules on Account Object

m. For Customer – Channel

ISCHANGED(Name) && ISPICKVAL(Type, "Customer – Channel")

n. For Customer – Direct

ISCHANGED(Name) && ISPICKVAL(Type, "Customer – Direct")

o. For Billing

State/Pro

vince

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))

p. For Billing Country

BillingCountry <> "US" && BillingCountry <> "USA" && BillingCountry <> "United States" && NOT(ISBLANK(BillingCountry))

q. For Shipping State/Province and Shipping Country

Don’t forget replicate For Shipping State/Province and Shipping Country same as Billing State/Province and Billing Country validation which I have mentioned above.

Challenge 3

It can be done easily:

r. Create a object and make sure the object name should be ***Robot_Setup__c***

s. Edit the Robot name(Standard field) switch the data type from Text to AutoNumber and make sure the display format should be ***ROBOT SETUP-{0000}***

t. Create following fields with

   correct data type:

   Date———————>Date

   c————————->DATE

   Notes——————-> Notes

   c————————>TEXT

Day of the Week——>Day_of_the_Weekc———>TEXT

Challenge 4

u. Create Sales Process in Opportunity; the name should be ***RB Robotics Sales Process***.

v. Create a record type; the name should be ***RB Robotics Process RT***.

W. Add ***Awaiting Approval*** value in opportunity Stage don’t forget to add RB Robotics Process RT record type.

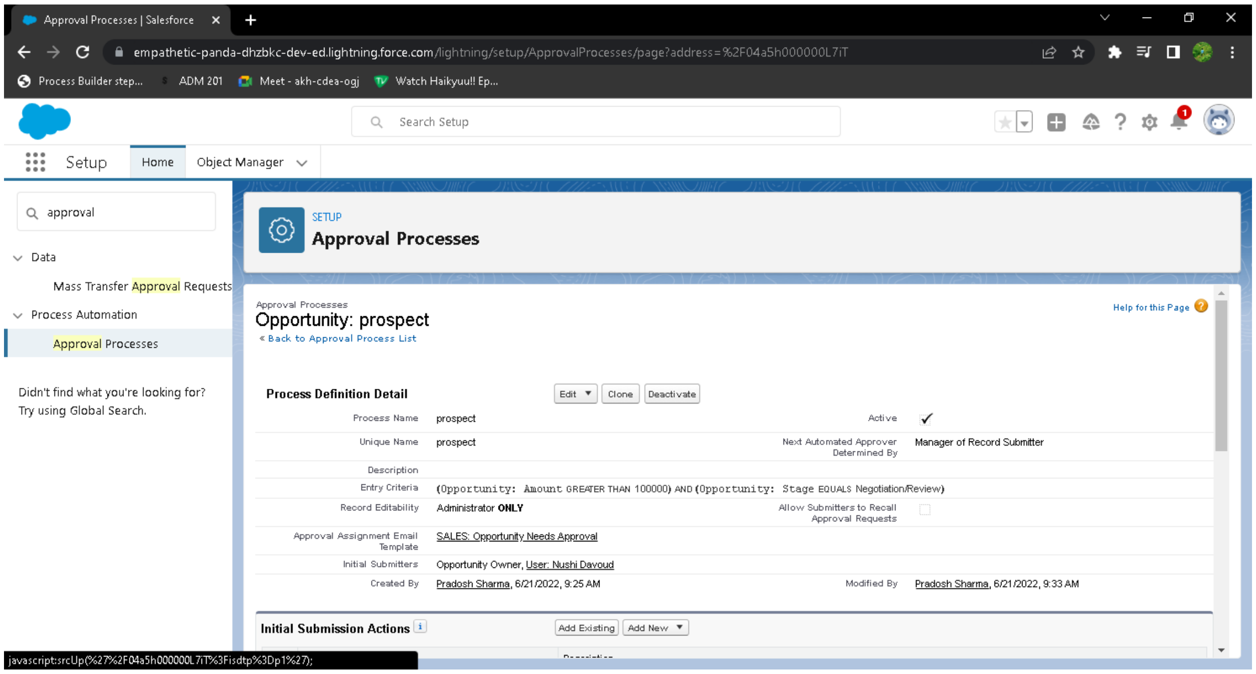
x. Create a Checkbox field and Name it ***Approved.***

y. Write a validation rule as below:

AND(Amount > 100000, Approvedc = False)

Challenge 5

Approval Process Definition Detail: See the screenshot below for details

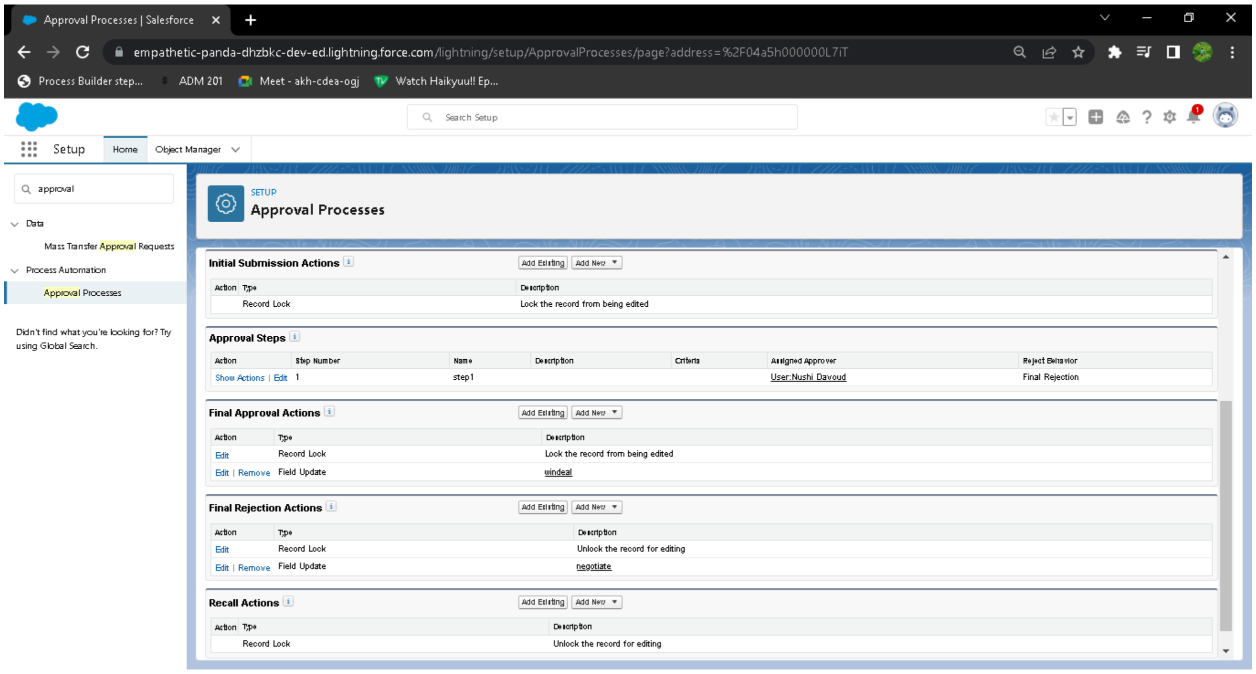


Approval Processes
Opportunity: prospect
« Back to Approval Process List

Process Definition Detail

Process Name	prospect	Active	✓
Unique Name	prospect	Next Automated Approver Determined By	Manager of Record Submitter
Description			
Entry Criteria	(Opportunity: Amount GREATER THAN 100000) AND (Opportunity: Stage EQUALS Negotiation/Review)		
Record Editability	Administrator **ONLY**	Allow Submitters to Recall Approval Requests	☐
Approval Assignment Email Template	SALES: Opportunity Needs Approval		
Initial Submitters	Opportunity Owner, User: Nushi Davoud		
Created By	Pradosh Sharma, 6/21/2022, 9:25 AM	Modified By	Pradosh Sharma, 6/21/2022, 9:33 AM

Initial Submission Actions



Initial Submission Actions

Action	Type	Description
	Record Lock	Lock the record from being edited

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions | Edit	1	step1			User:Nushi Davoud	Final Rejection

Final Approval Actions

Action	Type	Description
Edit	Record Lock	Lock the record from being edited
Edit | Remove	Field Update	windeal

Final Rejection Actions

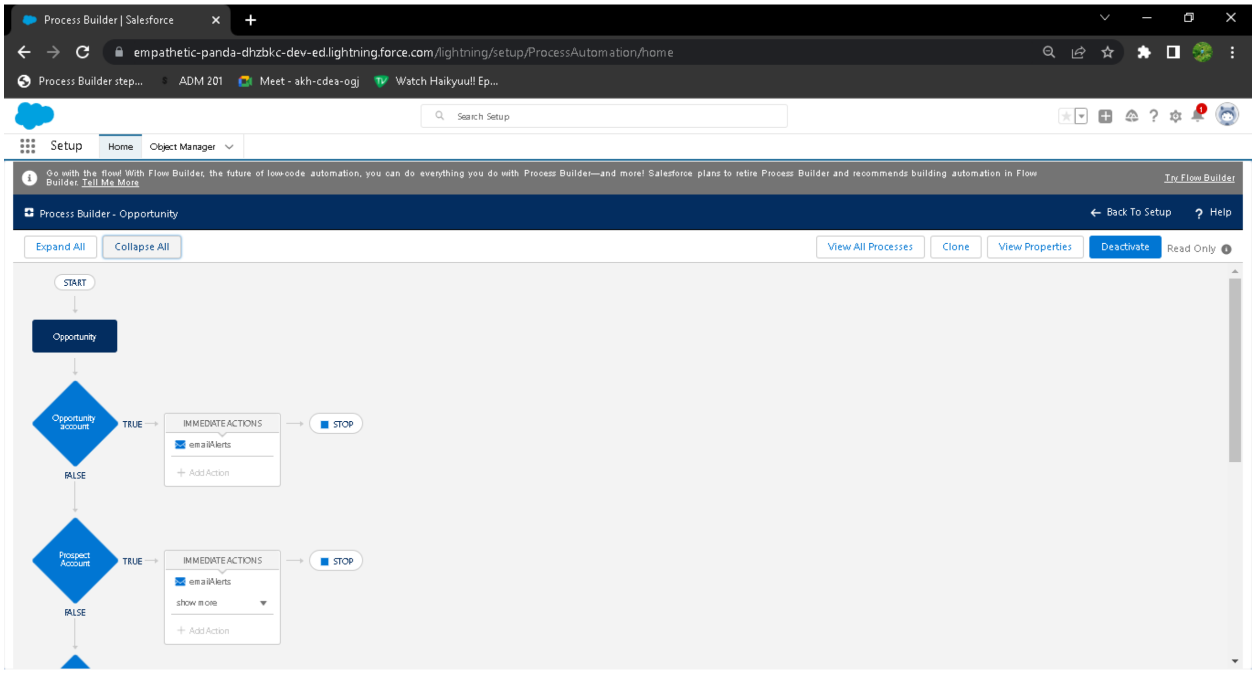
Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit | Remove	Field Update	negotiate

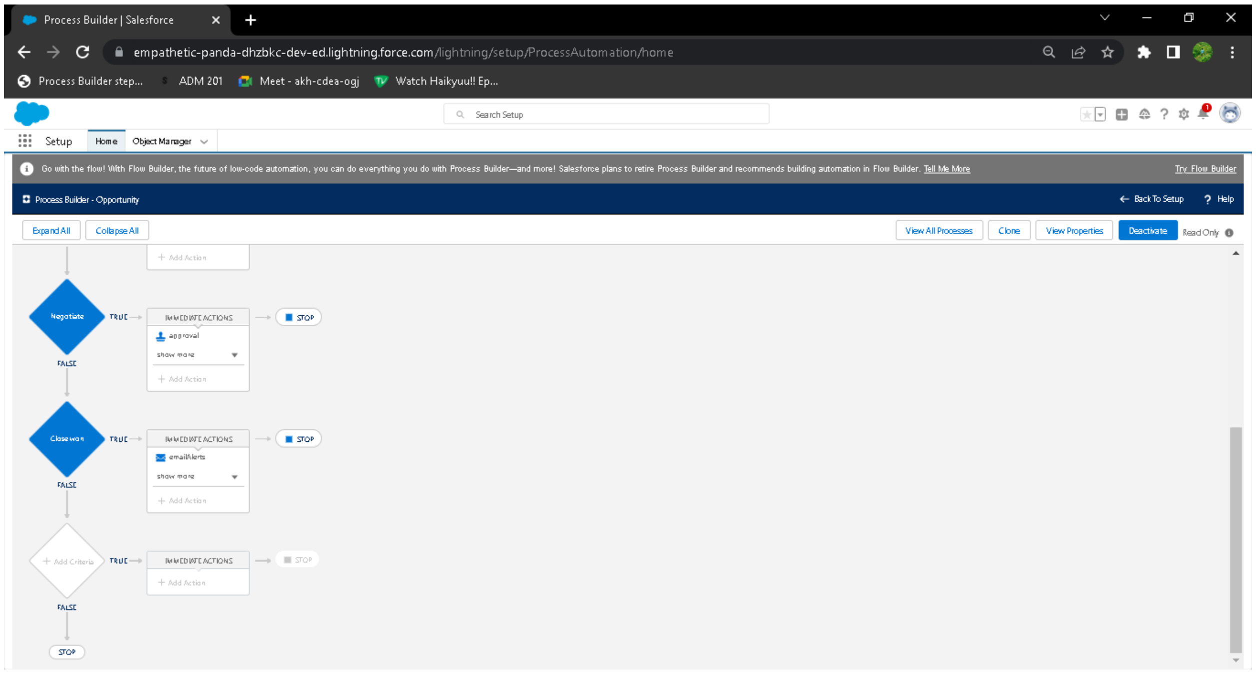
Recall Actions

Action	Type	Description
	Record Lock	Unlock the record for editing

It’s time to create **Process Builder**.

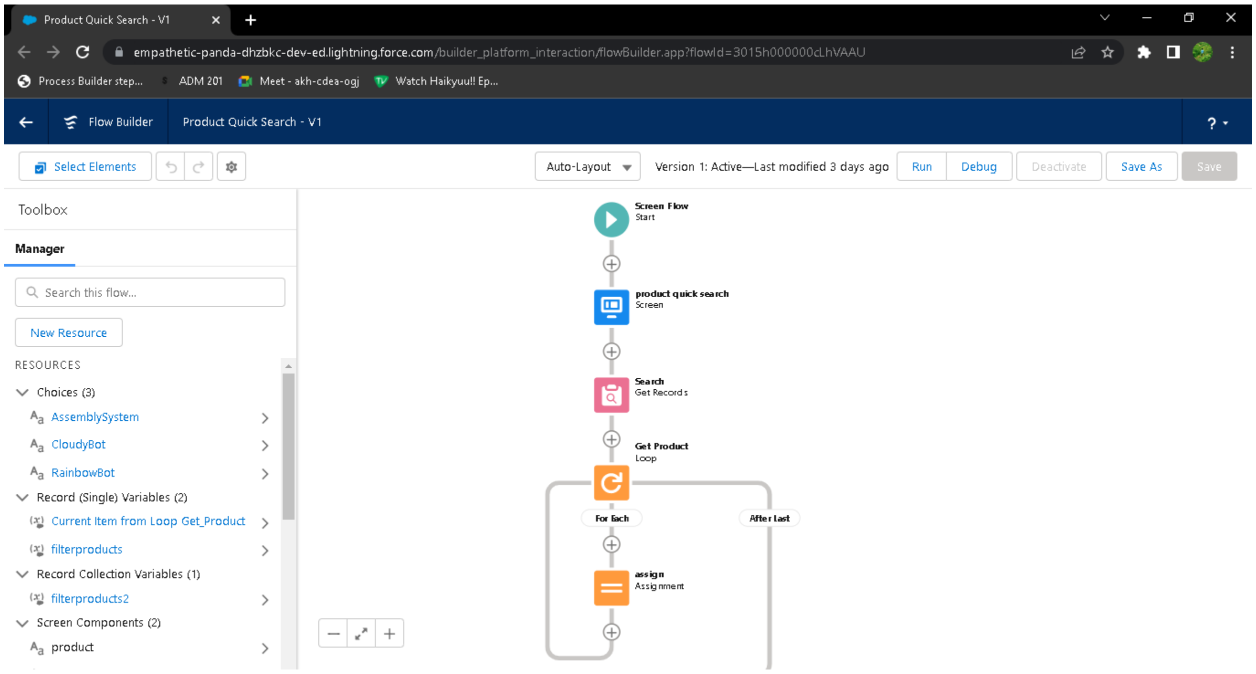
Name : Opportunity

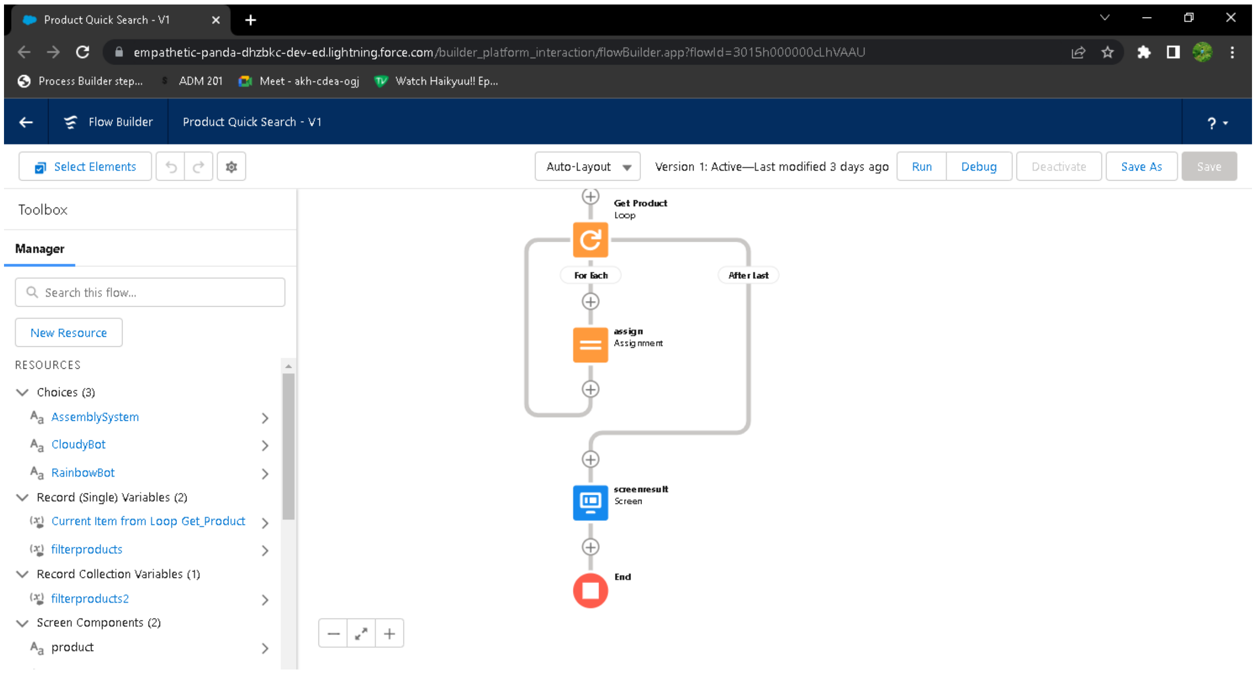



Process Builder | Salesforce
empathetic-panda-dhzbkc-dev-ed.lightning.force.com/lightning/setup/ProcessAutomation/home
Setup
Home
Object Manager
Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. Tell Me More
Try Flow Builder
Process Builder - Opportunity
Back To Setup
Help
Expand All
Collapse All
View All Processes
Clone
View Properties
Deactivate
Read Only
Negotiate
TRUE
FALSE
IMMEDIATE ACTIONS
approval
show more
Add Action
STOP
Closewon
emailAlerts
Add Criteria

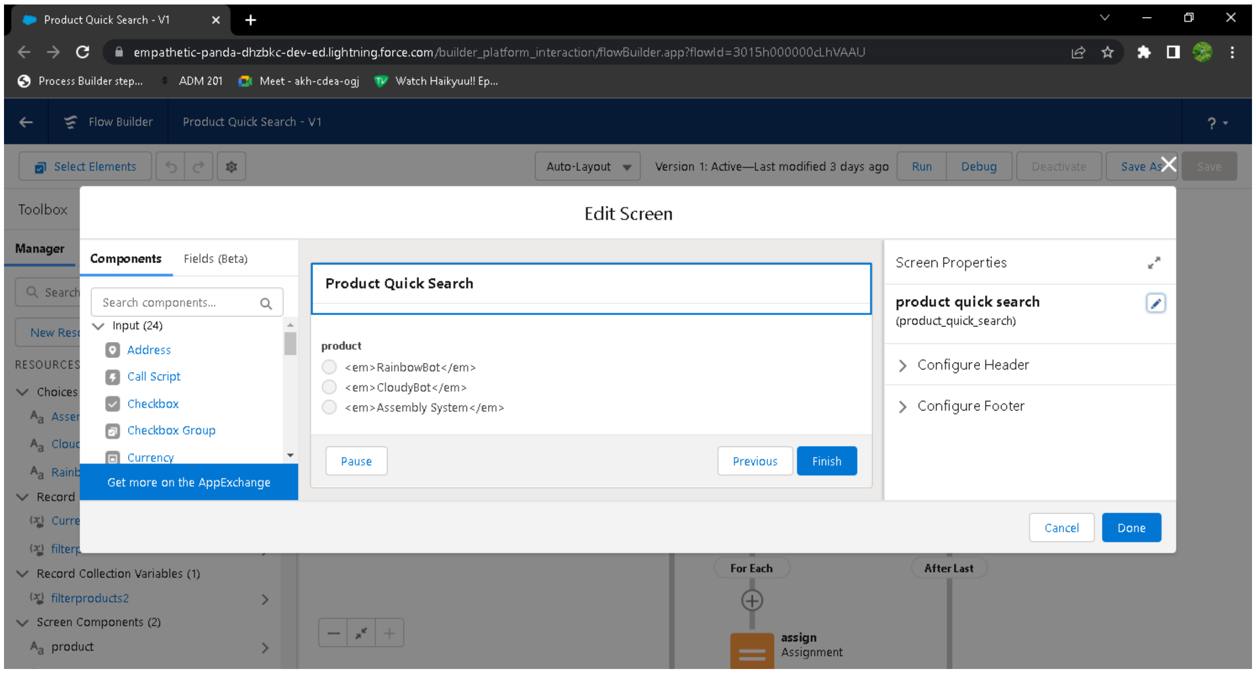
Challenge 6

Create the flow to display products.

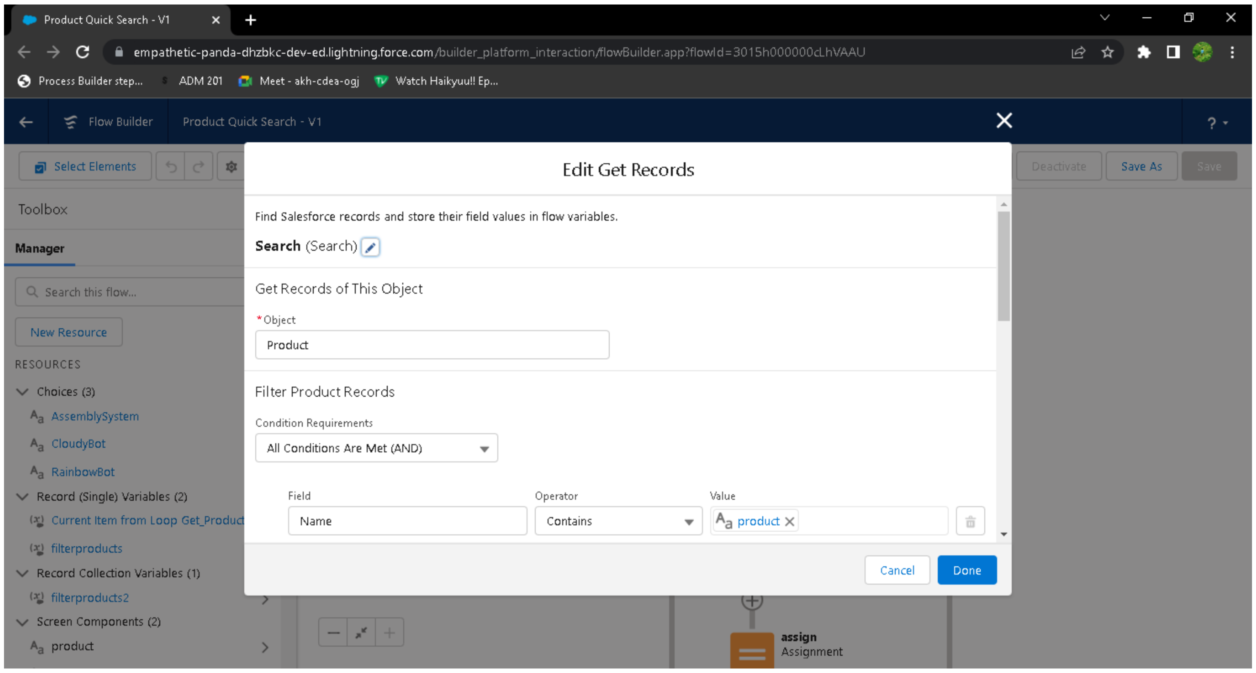



Product Quick Search - V1
empathetic-panda-dhzbkc-dev-ed.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=3015h000000cLhVAAU
Flow Builder
Product Quick Search - V1
Select Elements
Auto-Layout
Version 1: Active—Last modified 3 days ago
Run
Debug
Deactivate
Save As
Save
Toolbox
Manager
Search this flow...
New Resource
RESOURCES
Choices (3)
AssemblySystem
CloudyBot
RainbowBot
Record (Single) Variables (2)
Current Item from Loop Get_Product
filterproducts
Record Collection Variables (1)
filterproducts2
Screen Components (2)
product
Get Product
Loop
For Each
After Last
assign
Assignment
screenresult
Screen
End

SCREEN(PRODUCT TYPE SEARCH) Properties:



Get Records (Product Name Lookup) Properties:



Edit Get Records

Find Salesforce records and store their field values in flow variables.

Search (Search)

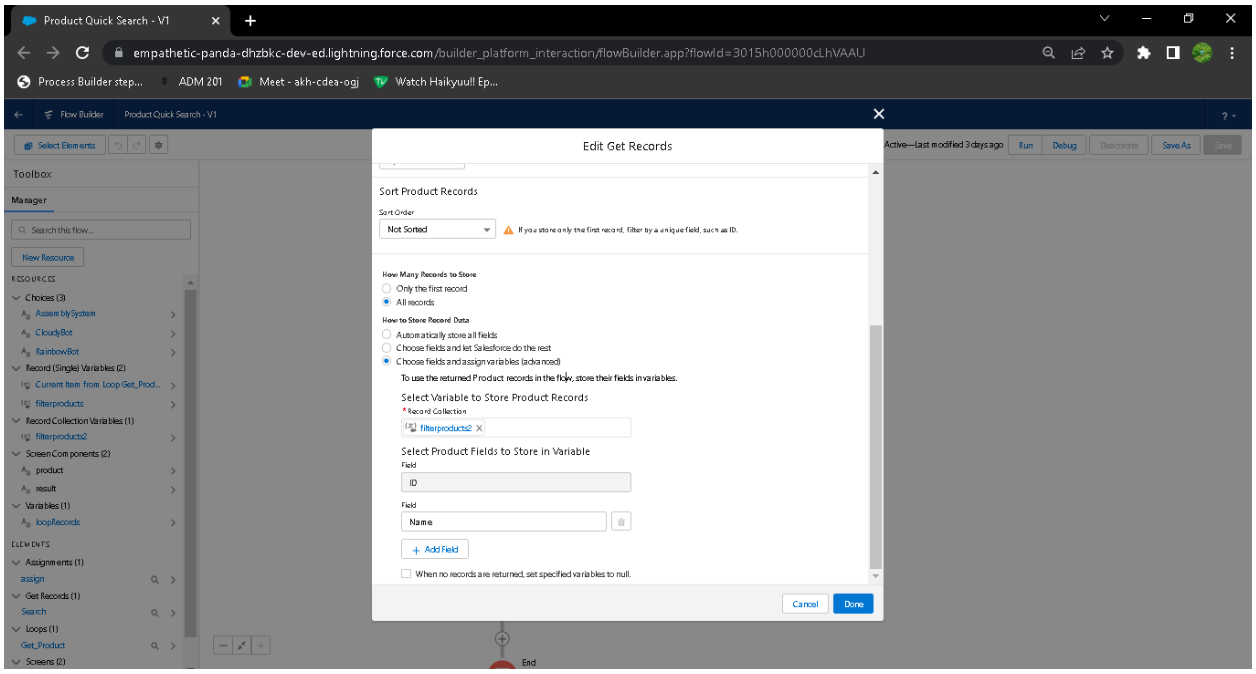
Get Records of This Object

* Object: Product

Filter Product Records

Condition Requirements: All Conditions Are Met (AND)

Field	Operator	Value
Name	Contains	product



Edit Get Records

Sort Product Records

Sort Order: Not Sorted

If you store only the first record, filter by a unique field, such as ID.

How Many Records to Store

( ) Only the first record
[✓ All records]

How to Store Record Data

( ) Automatically store all fields
( ) Choose fields and let Salesforce do the rest
[✓ Choose fields and assign variables (advanced)]

To use the returned Product records in the flow, store their fields in variables.

Select Variable to Store Product Records

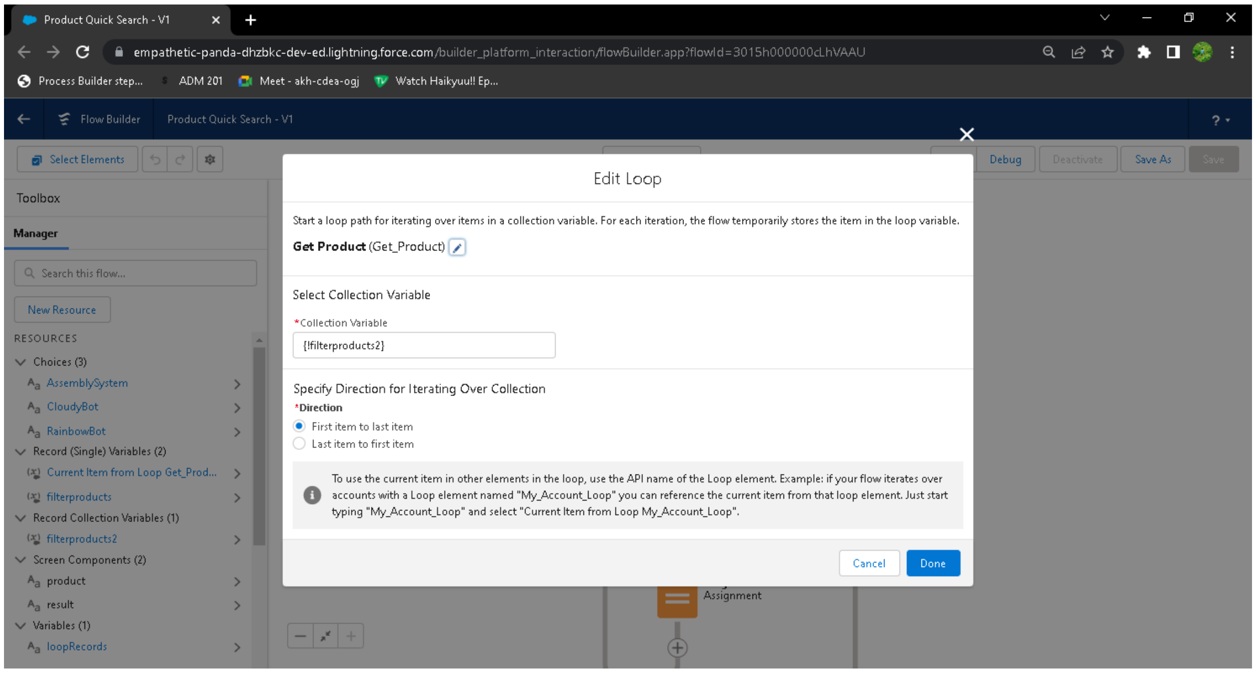
* Record Collection: filterproducts2

Select Product Fields to Store in Variable

Field: ID

Field: Name

[ ] When no records are returned, set specified variables to null.


Edit Loop

Start a loop path for iterating over items in a collection variable. For each iteration, the flow temporarily stores the item in the loop variable.

Get Product (Get_Product)

Select Collection Variable

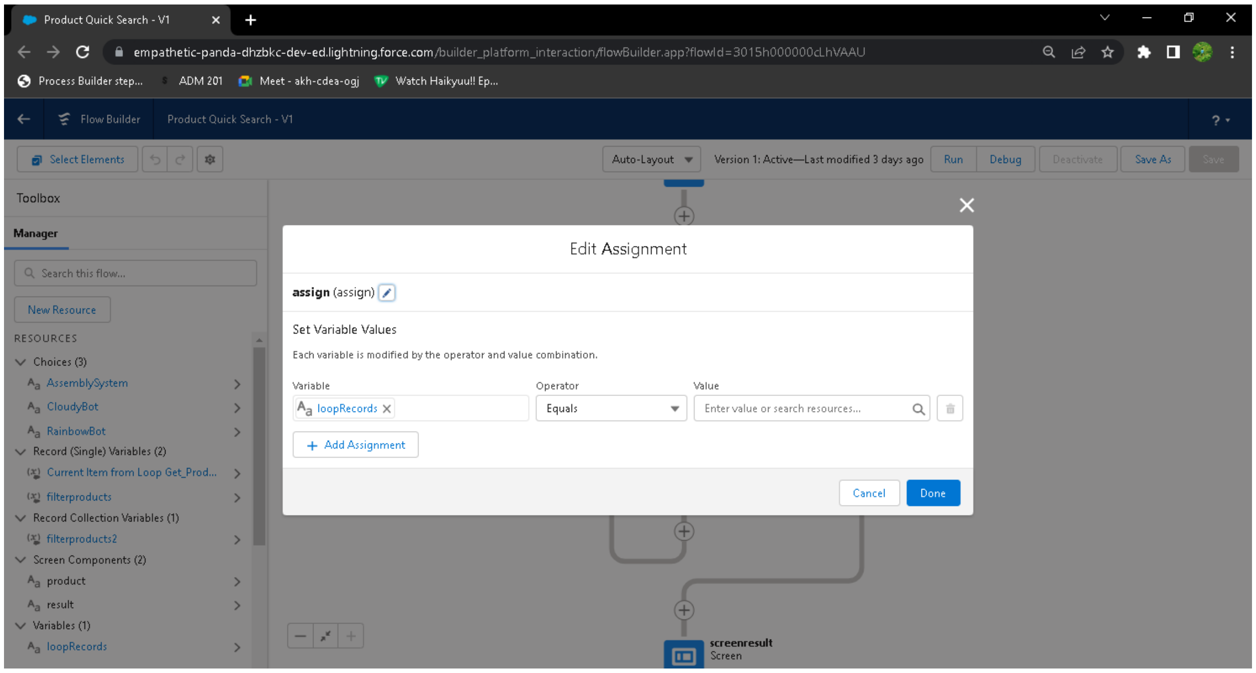
*Collection Variable: {!filterproducts2}

Specify Direction for Iterating Over Collection

*Direction

[✓ First item to last item]
Last item to first item

To use the current item in other elements in the loop, use the API name of the Loop element. Example: if your flow iterates over accounts with a Loop element named "My_Account_Loop" you can reference the current item from that loop element. Just start typing "My_Account_Loop" and select "Current Item from Loop My_Account_Loop".

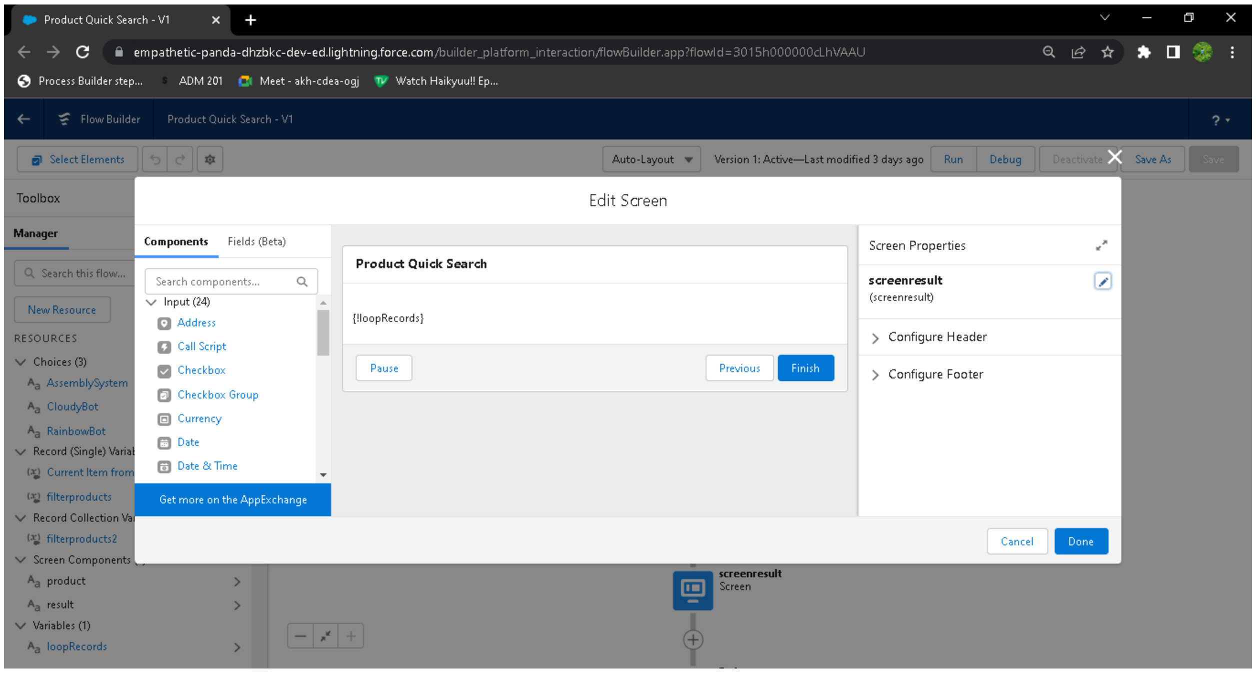

Edit Assignment

assign (assign)

Set Variable Values

Each variable is modified by the operator and value combination.

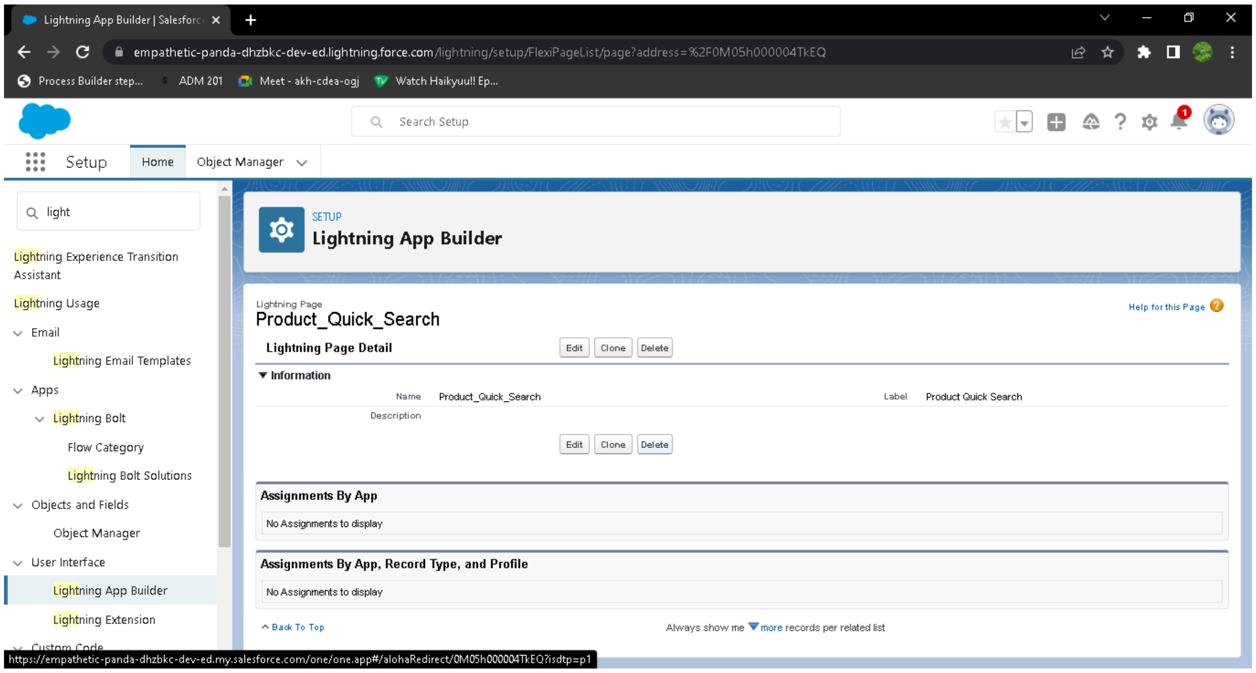
Variable	Operator	Value
loopRecords	Equals	Enter value or search resources...



a. Activate the flow

b. Add the flow to the opportunity screen using app builder.

Create a Record Page on Opportunity Object:
Go to Lightning App Builder page and click new. Record Page Properties are as follows:-


Lightning App Builder | Salesforce
empathetic-panda-dhzbkc-dev-ed.lightning.force.com/lightning/setup/FlexiPageList/page?address=%2F0M05h000004TkEQ
Search Setup
Setup
Home
Object Manager
light
Lightning Experience Transition Assistant
Lightning Usage
Email
Lightning Email Templates
Apps
Lightning Bolt
Flow Category
Lightning Bolt Solutions
Objects and Fields
Object Manager
User Interface
Lightning App Builder
Lightning Extension
Custom Code
SETUP
Lightning App Builder
Help for this Page
Edit
Clone
Delete
Back To Top
Always show me more records per related list
https://empathetic-panda-dhzbkc-dev-ed.my.salesforce.com/one/one.app#/alohaRedirect/0M05h000004TkEQ?isdtp=p1

Lightning Page
Product_Quick_Search

Lightning Page Detail

Information

Name	Product_Quick_Search	Label	Product Quick Search
Description			

Assignments By App
No Assignments to display

Assignments By App, Record Type, and Profile
No Assignments to display

- Add the component on newly created Opportunity Record Page.
- Please don’t forgot to Activate the page.

Challenge 7

- Change the datatype for “Day of the week” field from TEXT to Formula (TEXT) and use the following the formula to get Day of the week

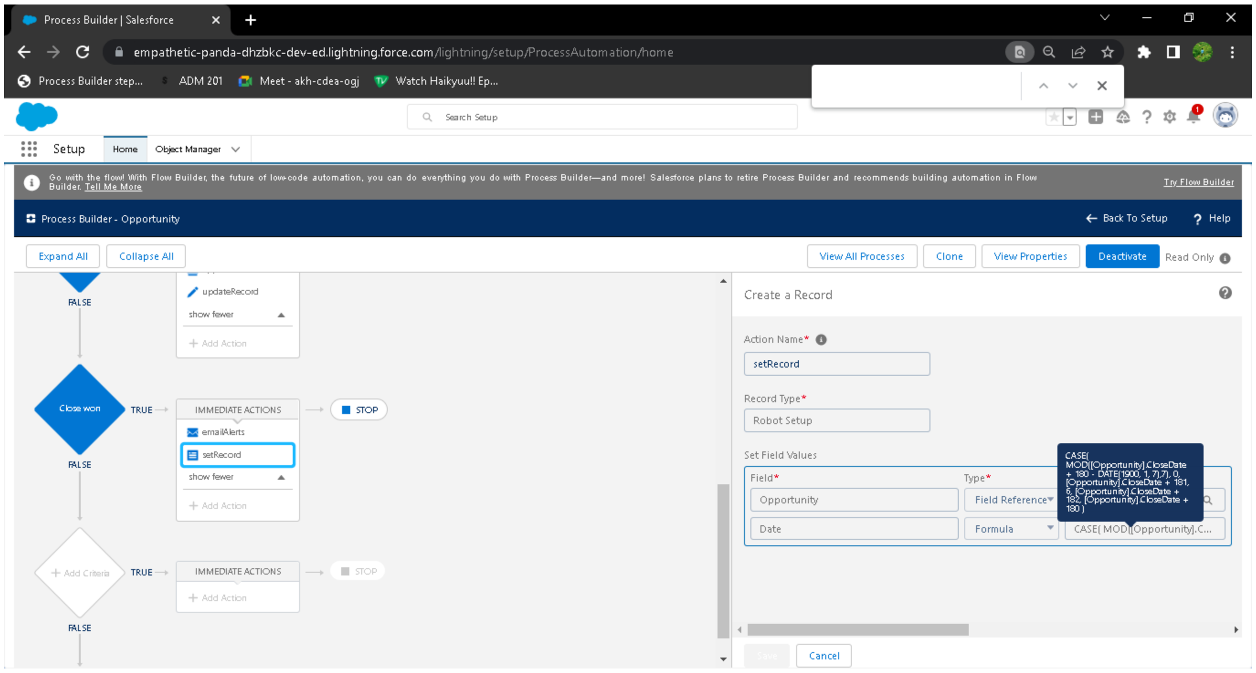
CASE(MOD(Datec – DATE(1900, 1, 7), 7), 0, “Sunday”, 1, “Monday”, 2, “Tuesday”, 3, “Wednesday”, 4, “Thursday”, 5, “Friday”, 6,

“Saturday”,”Error”) Or You can use this formula

also instead of above formula

CASE(WEEKDAY(Datec),
1, “Sunday”,
2, “Monday”,
3, “Tuesday”,
4, “Wednesday”,
5, “Thursday”,
6, “Friday”,
7, “Saturday”,
Text(WEEKDAY(
Datec)))

Update the process Opportunity to handle the robot setup saturday date case:


Process Builder | Salesforce
empathetic-panda-dhzbkc-dev-ed.lightning.force.com/lightning/setup/ProcessAutomation/home
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 - Opportunity
Back To Setup
Help
Expand All
Collapse All
View All Processes
Clone
View Properties
Deactivate
Read Only
updateRecord
show fewer
Add Action
FALSE
Close won
TRUE
IMMEDIATE ACTIONS
emailAlerts
setRecord
STOP
Add Criteria
Cancel

Create a Record

Action Name*: setRecord

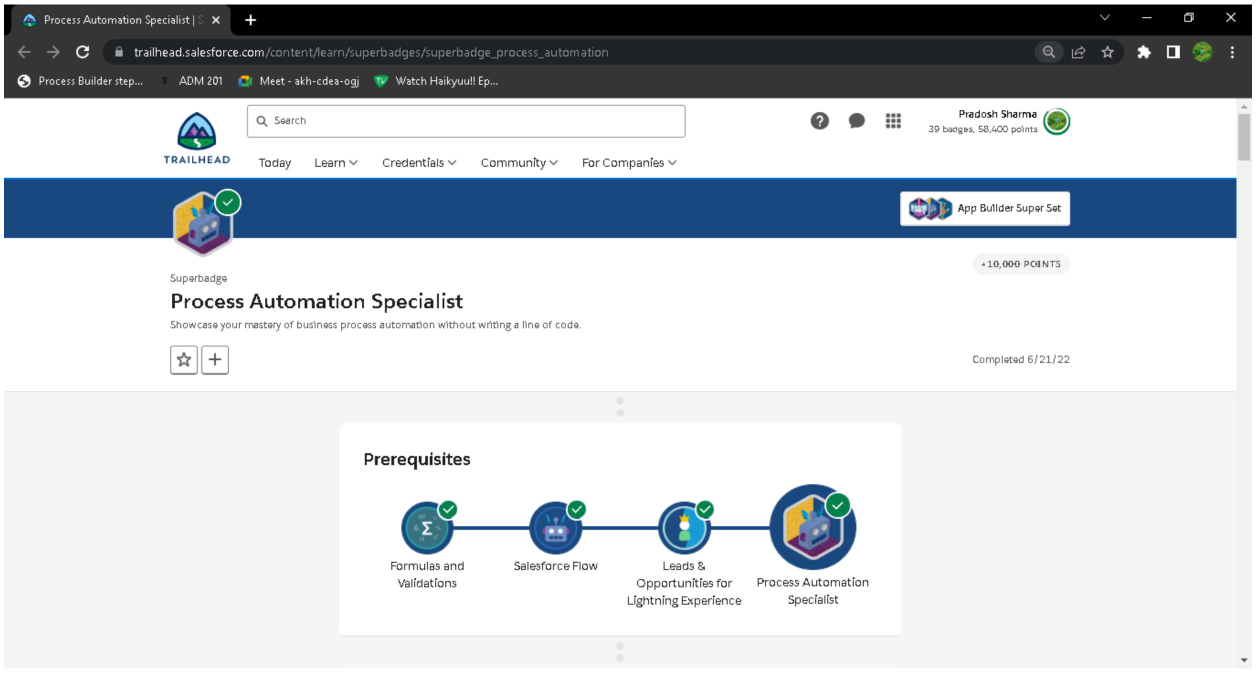
Record Type*: Robot Setup

Set Field Values

Field*	Type*	Value
Opportunity	Field Reference	
Date	Formula	CASE(MOD([Opportunity].C...

```
CASE(
MOD([Opportunity].CloseDate
+ 180 - DATE(1900, 1, 7),7), 0,
[Opportunity].CloseDate + 181,
6, [Opportunity].CloseDate +
182, [Opportunity].CloseDate +
180 )
```

Activate the Process and you are done!



Superbadge

Process Automation Specialist

Showcase your mastery of business process automation without writing a line of code.

Completed 6/21/22

Prerequisites

- Formulas and Validations
- Salesforce Flow
- Leads & Opportunities for Lightning Experience
- Process Automation Specialist