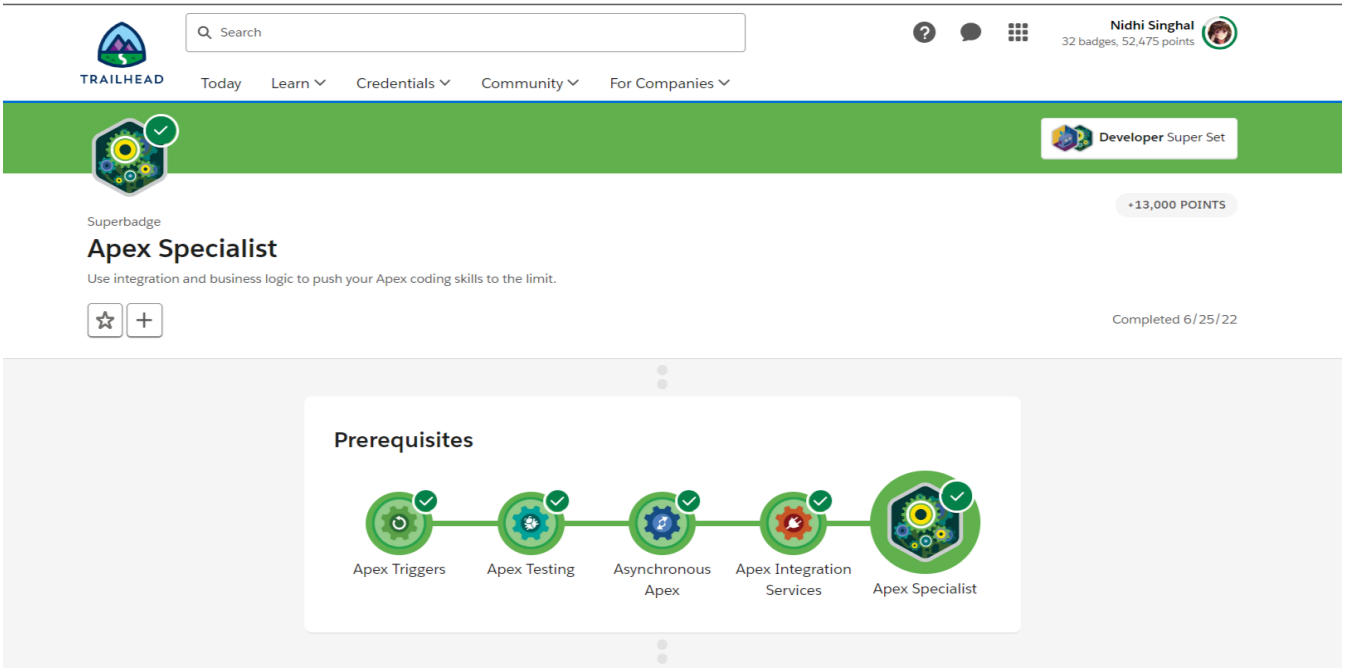
PROJECT REPORT (SALESFORCE DEVELOPER CATALYST)
SUPERBADGE 1
(APEX SPECIALIST)

PRE REQUISITES



1. CREDENTIAL SECURITY

In this module some pre setup for the challenge was done , like installing package , Editing some objects and creating required fieds to be used further

2. AUTOMATED RECORD CREATION

In this module we worked in developer console , worked in Apex class MAINTENANCEREQUESTHELPER and MAINTENANCEREQUEST below are the codes used:

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));
        } 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> 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); } }
```

3. SYNCHRONIZE SALESFORCE DATA WITH AN EXTERNAL SYSTEM:

In this module we worked in developer console , and as we also create a new Remote Site Setting named Warehouse. Also after working in WarehouseCalloutService.apxc we saved our code and then in anonymous window run this command.

```
System.enqueueJob(new WarehouseCalloutService());
```

4.SCHEDULE SYNCHRONIZATION USING APEX CODE :

In this module we worked in developer console . Also after working in WarehouseSyncShedule.apxc we saved our code and run them all.

WarehouseSyncShedule.apxc:

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

WarehouseCalloutService.apxc:

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

  //class that makes a REST callout to an external warehouse system to get a list of equipment that needs to be updated.
  //The callout's JSON response returns the equipment records that you upsert in Salesforce.

  @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());

      //class maps the following fields: replacement part (always true), cost, current inventory, lifespan, maintenance cycle, and
warehouse SKU
      //warehouse SKU will be external ID for identifying which equipment records to update within Salesforce
      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 = (Integer) mapJson.get('cost');
        myEq.Warehouse_SKU__c = (String) mapJson.get('sku');
        myEq.Current_Inventory__c = (Double) mapJson.get('quantity');
        myEq.ProductCode = (String) mapJson.get('_id');
        warehouseEq.add(myEq);
      }

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

  public static void execute (QueueableContext context){
    runWarehouseEquipmentSync();
  }
```

5.TEST AUTOMATION LOGIC :

In this module we worked in developer console . Also worked in three apex classes and saved and run them all . Respective code is provided below.

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); } }
```

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); }
```

6.TEST CALLOUT LOGIC :

In this module we worked in developer console . Also worked in three apex classes and saved and run them all . Respective code is provided below.

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); } } }
      }
```

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();
    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;
  }
}
```

7.TEST SCHEDULING LOGIC :

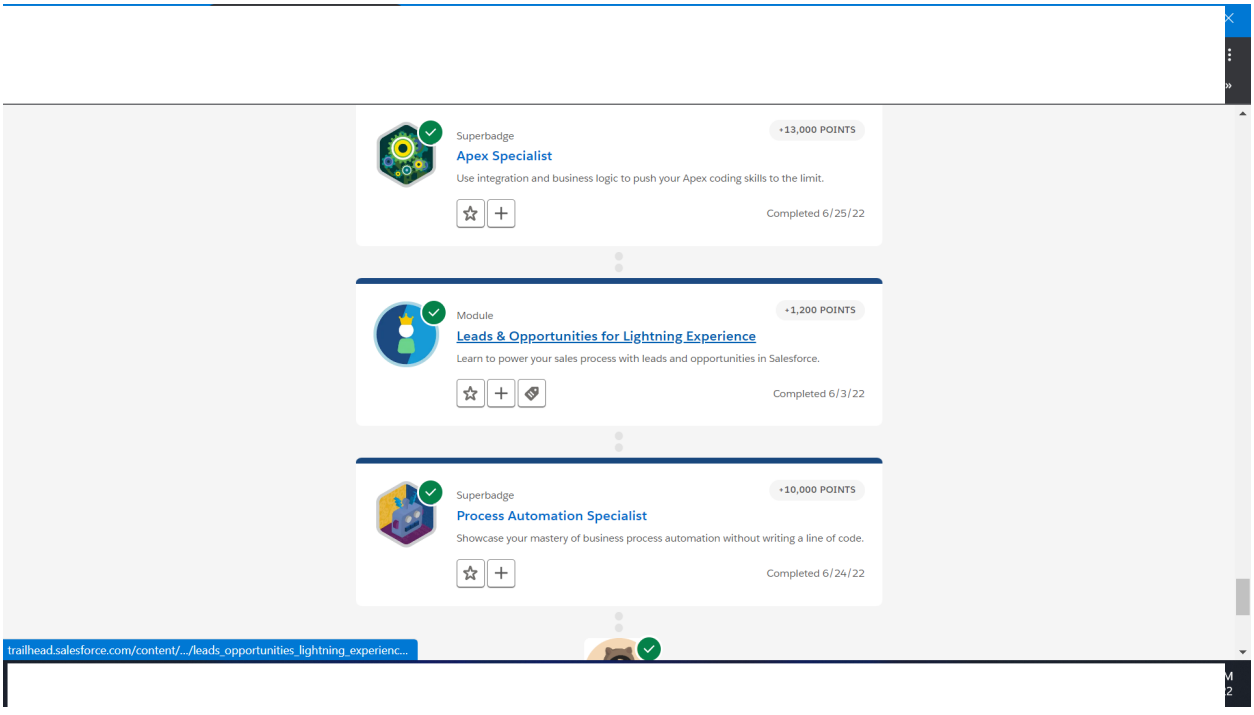
In this module also we worked in developer console . Also worked in two apex classes and saved and run them all . Respective code is provided below.

WarehouseSyncSchedule.apxc:

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

    WarehouseCalloutService.runWarehouseEquipmentSync();
  }
}
```

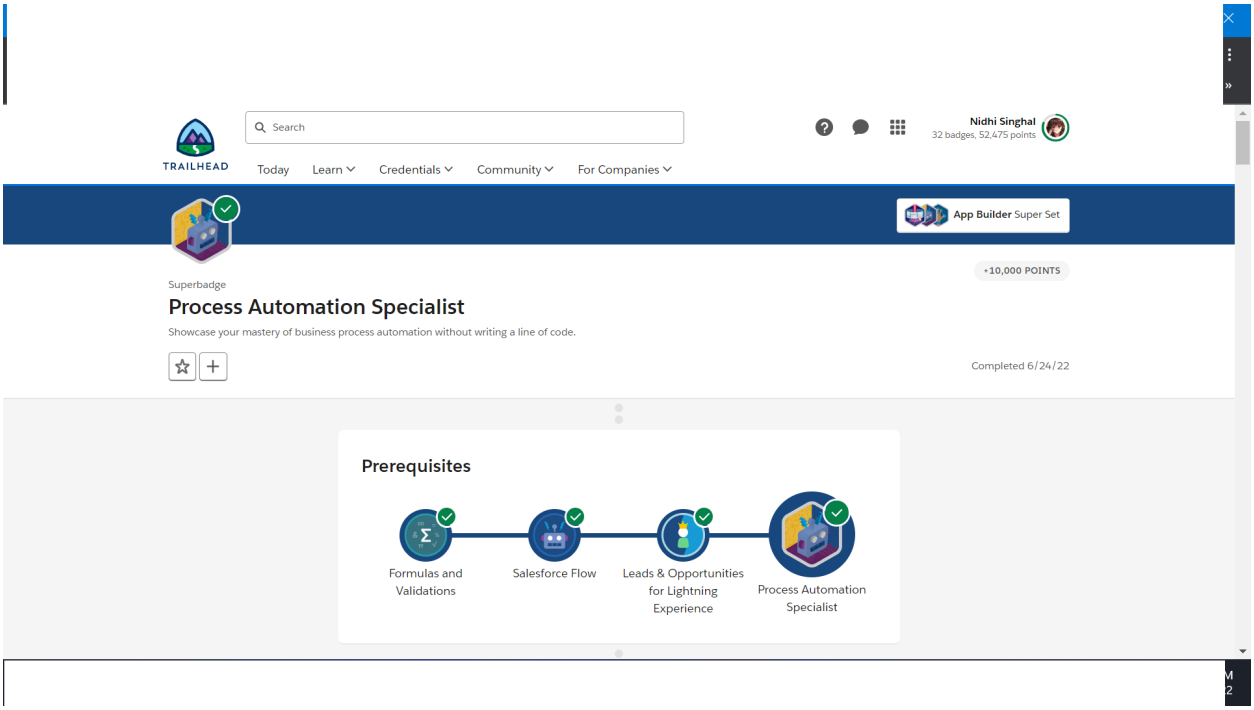
WarehouseSyncScheduleTest.apxc:

```
                    @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();
        //Contains schedule information for a scheduled job. CronTrigger is similar to a cron job on UNIX systems.
        // This object is available in API version 17.0 and later.
        CronTrigger a=[SELECT Id FROM CronTrigger where NextFireTime > today];
        System.assertEquals(jobID, a.Id,'Schedule ');


    }
}
```



SUPERBADGE 2
(PROCESS AUTOMATION SPECIALIST)

pre requisites:



1. CREDENTIAL SECURITY

In this module some pre setup for the challenge was done , like installing package , Editing some objects and creating required fieds to be used further.

2.AUTOMATE LEADS:

In this module we were asked to create validation rule and queues named RainbowSales and Assembly System Sales.
Here main error condition formula used was as follows:

```
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))))
```

3. AUTOMATE ACCOUNTS:

In this module we were asked to create 4 roll up summary fields which are : Number of deal , Number of won deals Amount of won deals, Last won deals date all over object opportunity, 2 normal fields were made named Deal win percent and Call for Service.
here two more validation rules were formed by us:
US_Address, Name Change
Here main error condition formula used was as follows respectively:

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))))

and

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

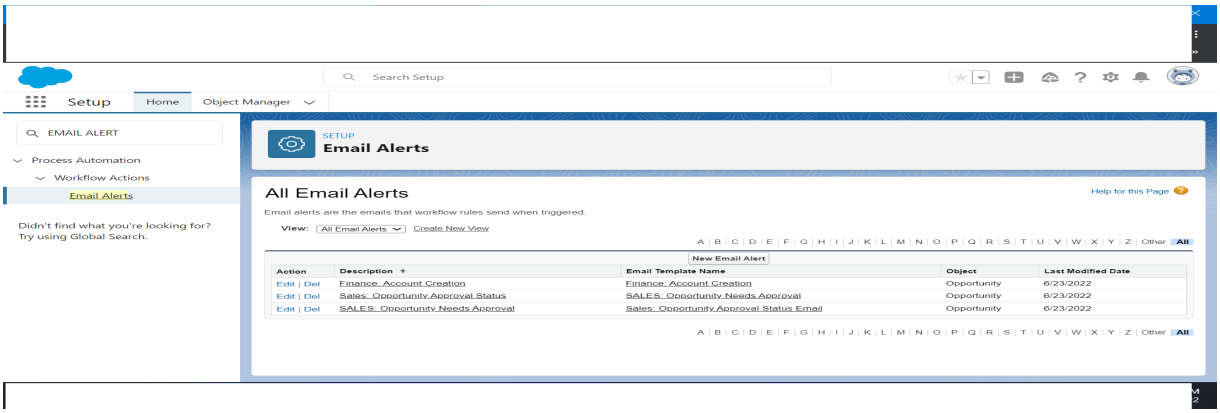
CREATE ROBOT SETUP OBJECT: **In master detail relationship with Opprotunoity it contains field names date, day of week , notes.**

4. CREATE SALES PROCESS AND VALIDATE OOPORTUNITIES:

In this for future reference we create a picklist value 'AWAITING APPROVAL' , create a opportunity validation formula and created a sales process.
Here main error condition formula used was as follows :

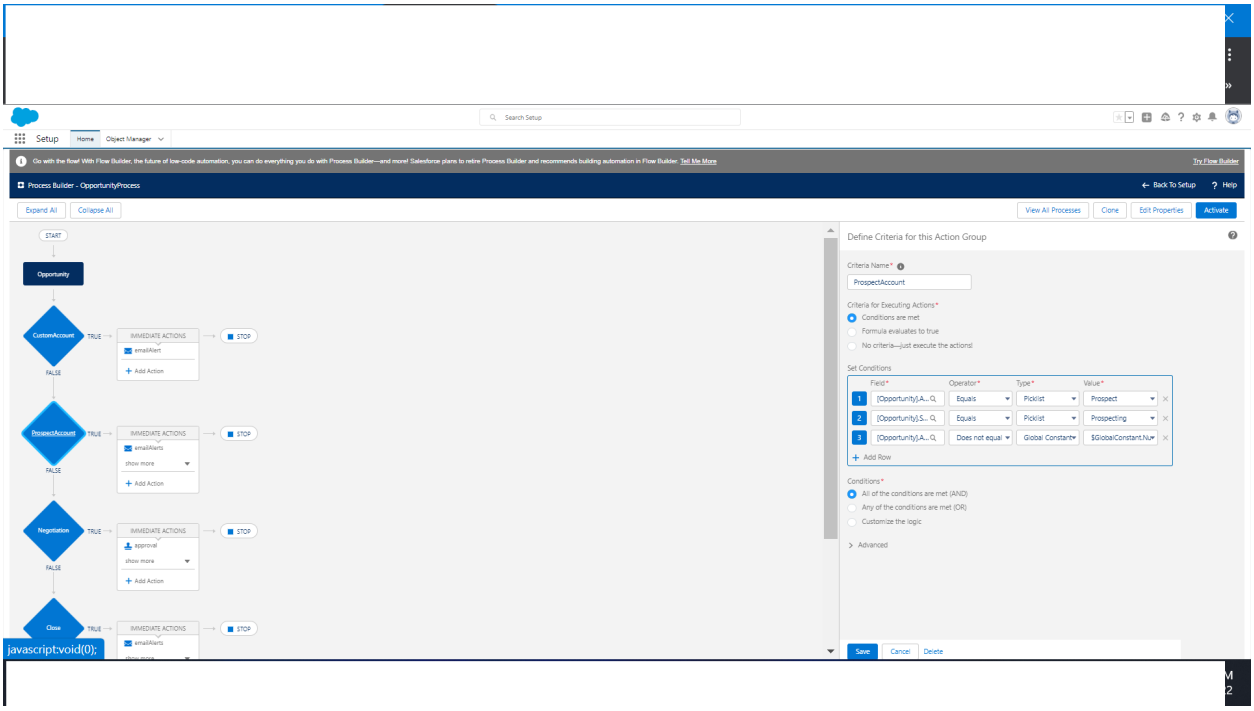
IF((Amount > 100000 && Approved__c <> True && ISPICKVAL(StageName,'Closed Won')),True,False)

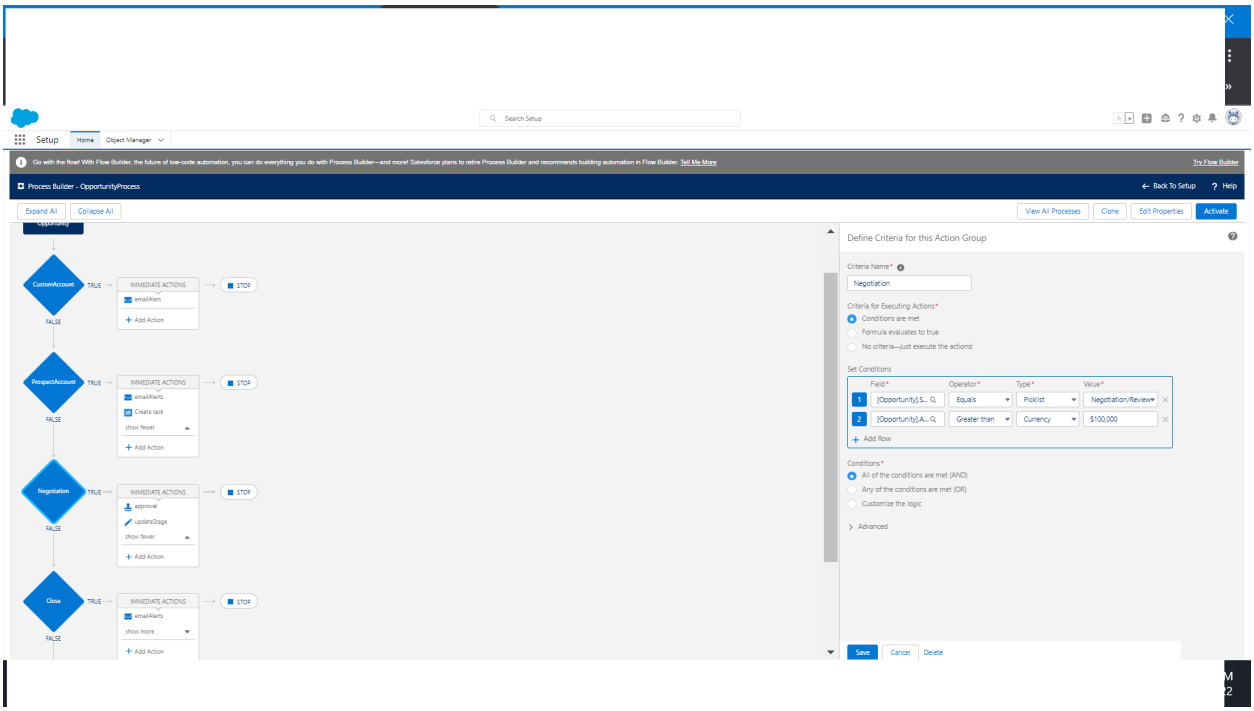
5. AUTOMATE OPPORTUNITIES: **In this we first created three email templates as alerts look at the adjacent photograph.**



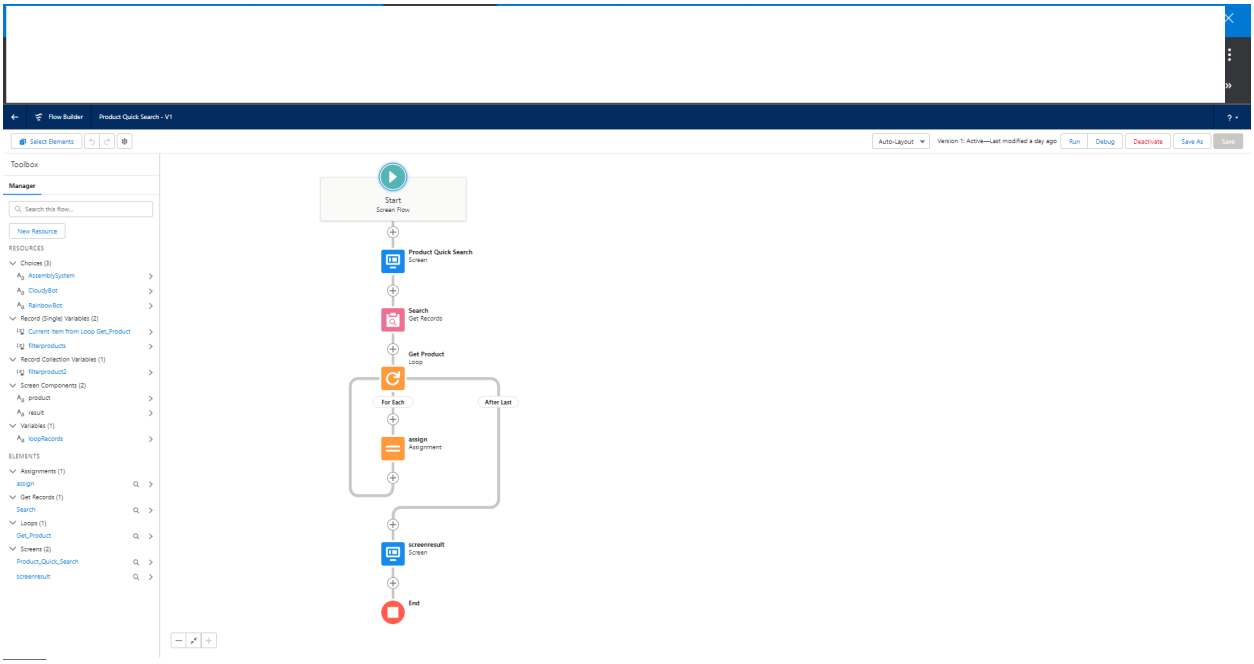
Now a process is created in the process builder

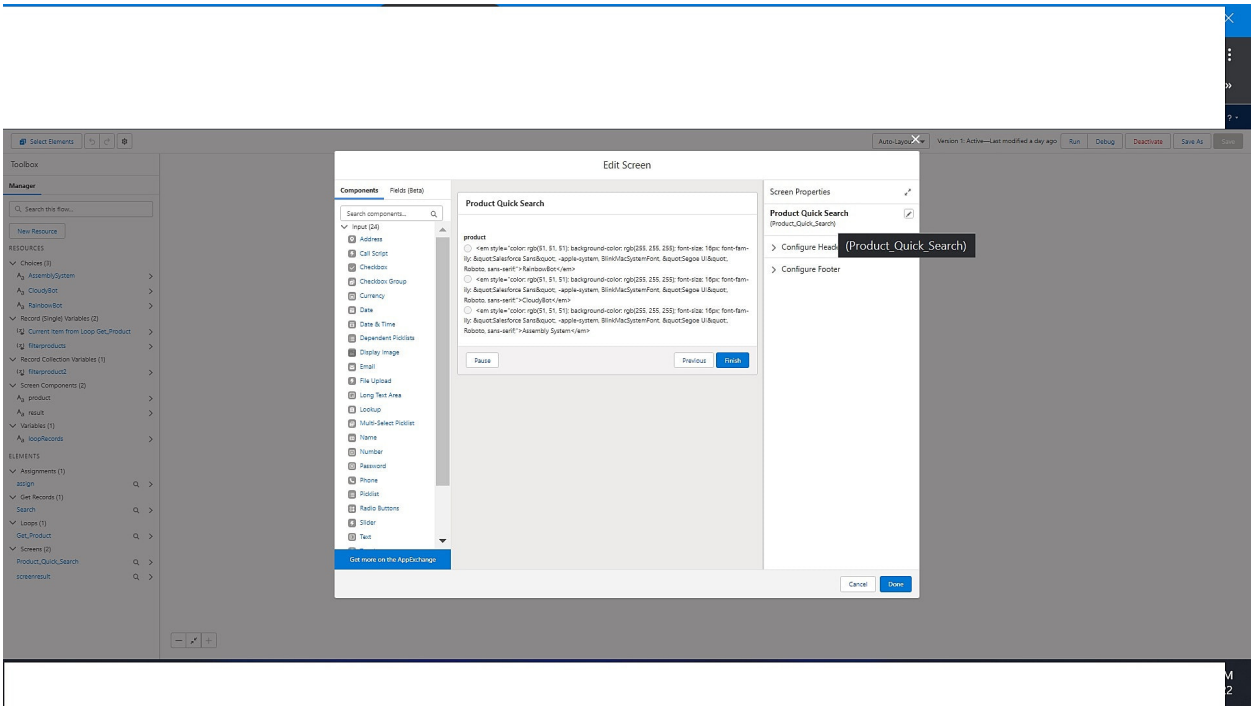
Here are the adjacent snapshot of same , there are used 4 criteria nodes .

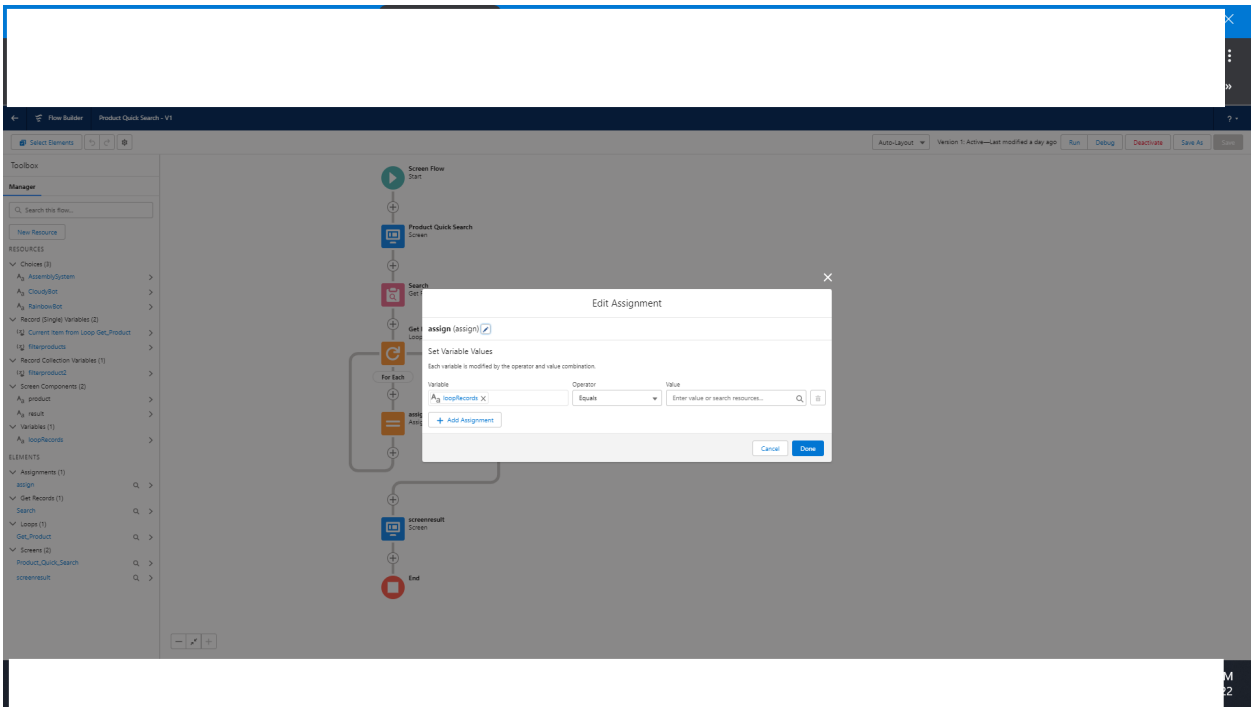


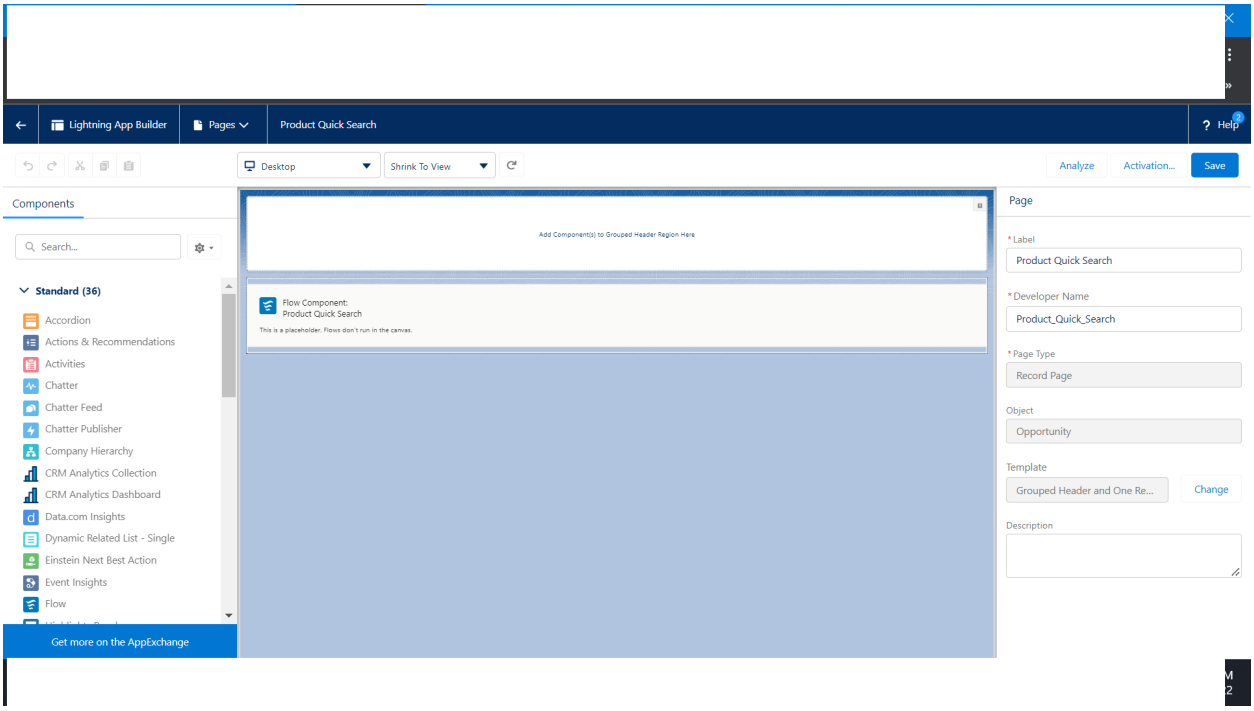

Setup
Home
Object Manager
Process Builder - OpportunityProcess
Back To Setup
Help
Expand All
Collapse All
View All Processes
Clone
Edit Properties
Activate
CustomAccount
ProspectAccount
Negotiation
Close
TRUE
FALSE
IMMEDIATE ACTIONS
STOP
Add Action
Define Criteria for this Action Group
Criteria Name*
Negotiation
Criteria for Executing Actions*
Conditions are met
Formula evaluates to true
No criteria—just execute the actions!
Set Conditions
Field*
Operator*
Type*
Value*
Equals
Picklist
Negotiation/Review
Greater than
Currency
$100,000
Add Row
Conditions*
All of the conditions are met (AND)
Any of the conditions are met (OR)
Customize the logic
Advanced
Save
Cancel
Delete

6. *CREATE FLOW FOR OOPORTUNITIES:* **Now a flow is created named PRODUCT QUICK SEARCH. And after creating it we use any template in Lightning app builder to save it at left pallete.**






Flow Builder
Product Quick Search - V1
Toolbox
Manager
Screen Flow
Start
Product Quick Search
Screen
Search
For Each
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...
Add Assignment
Cancel
Done
screenresult
Screen
End


Lightning App Builder
Pages
Product Quick Search
Help
Desktop
Shrink To View
Analyze
Activation...
Save
Components
Search...
Standard (36)
Accordion
Actions & Recommendations
Activities
Chatter
Chatter Feed
Chatter Publisher
Company Hierarchy
CRM Analytics Collection
CRM Analytics Dashboard
Data.com Insights
Dynamic Related List - Single
Einstein Next Best Action
Event Insights
Flow
Get more on the AppExchange
Add Component(s) to Grouped Header Region Here
Flow Component:
Product Quick Search
This is a placeholder. Flows don't run in the canvas.
Page
* Label
Product Quick Search
* Developer Name
Product_Quick_Search
* Page Type
Record Page
Object
Opportunity
Template
Grouped Header and One Re...
Change
Description

6. AUTOMATE SETUP: **Now in robot setup object in day of week field of formula type we use following formula:**

```
Case ( WEEKDAY( Date__c ),
1,"Sunday",
2,"Monday",
3,"Tuesday",
4,"Wednesday",
5,"Thursday",
6,"Friday",
7,"Saturday",
Text(WEEKDay(Date__c)))
```

And now in process builder we update date field with following formula:

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

AND SO, BOTH SUPERBADGES ALONG WITH COMPLETE SALEFORCE DEVELOPER CATALYST MODULE ARE DONE.

HERE IS MY TRAILHEAD PLAYGROUND LINK:

https://trailblazer.me/id/nsinghal46

THANK YOU!