Section A

1. (2)
2. (1,3,4,5)
3. (2)
4. (1)
5. (1)
6. (2)
7. (3,4,5)
8. (2)
9. (3)
10. (4)
11. (4)
12. (2,3,4)
13. (4)
14. (4)
15. (4)
16. (4)
17. (1)
18. (2)
19. (3)
20. (4)

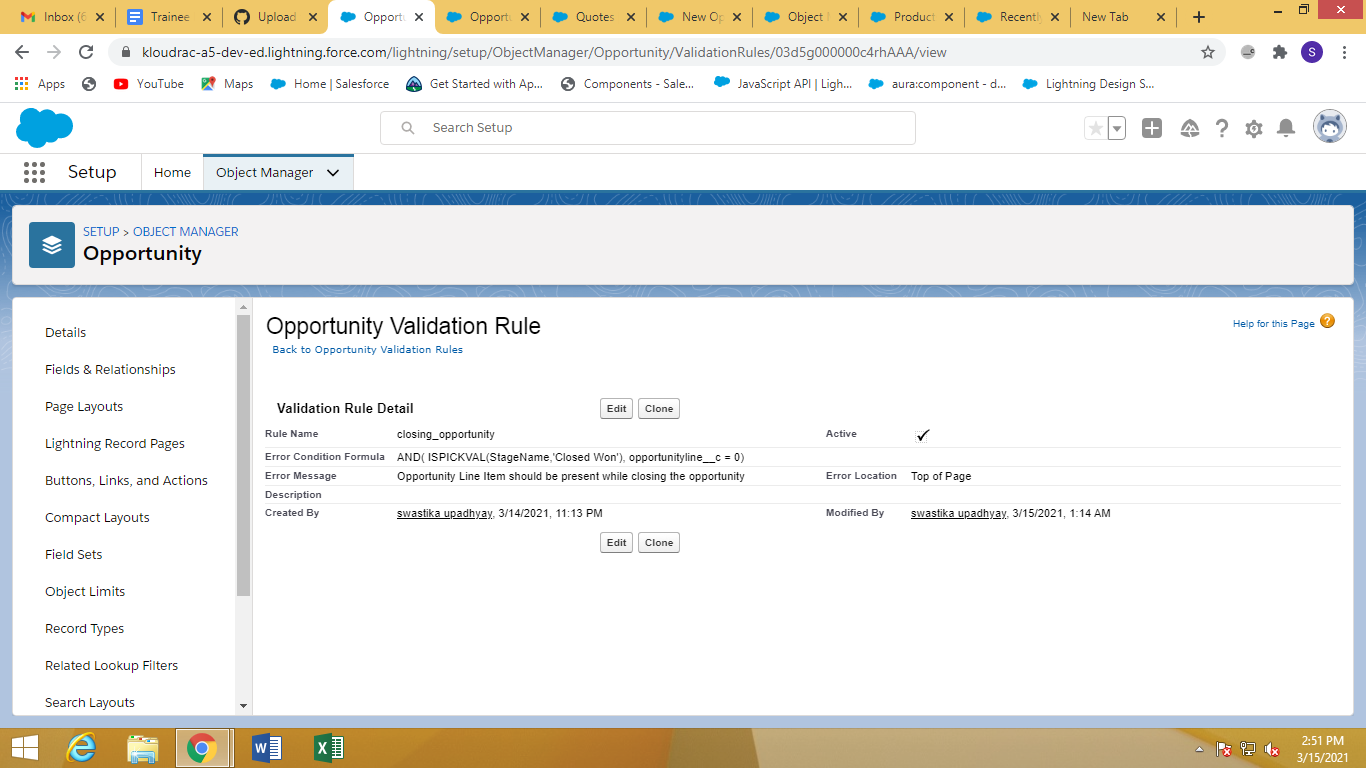
SECTION B

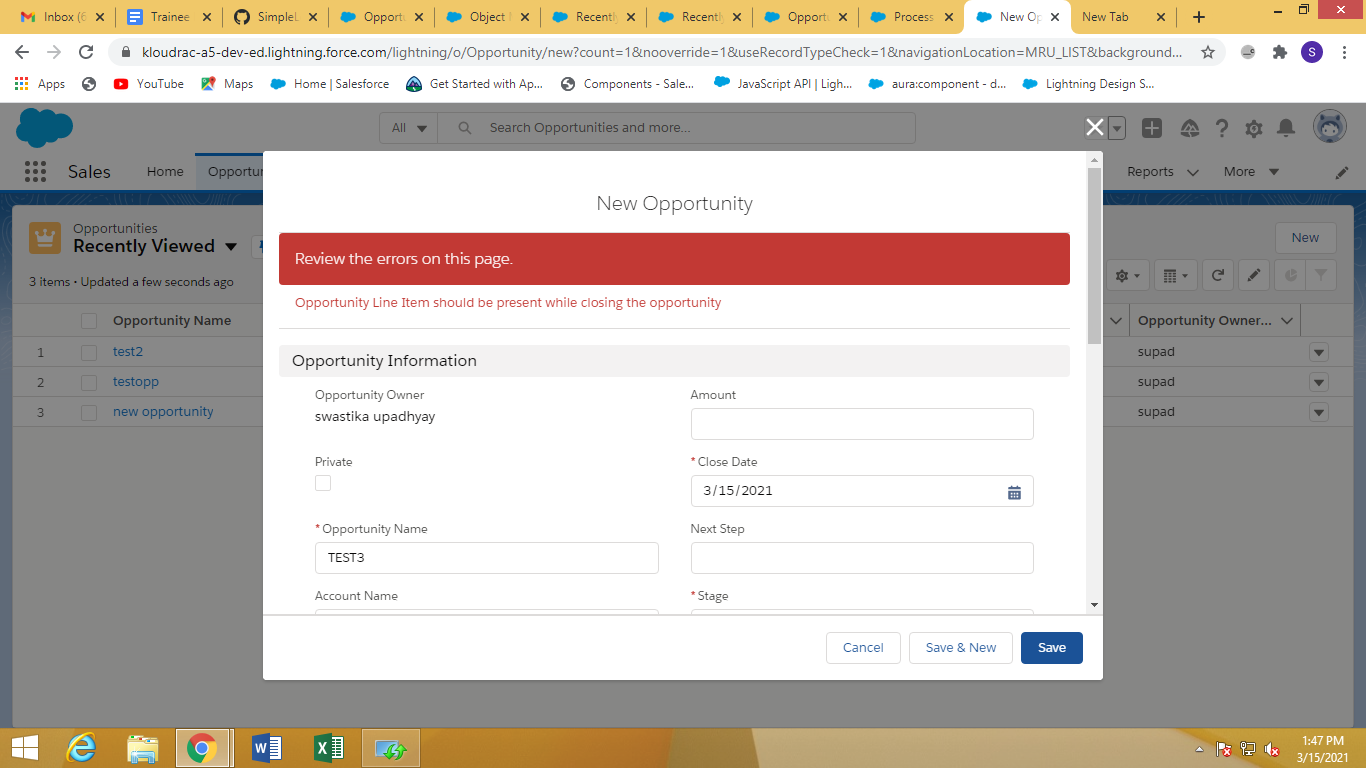
1. **Validation Rules (15 Marks)**

A)While closing the opportunity ( 'Closed Won') , an Opportunity Line Item

should be present. If  opportunity line item is not present under a opportunity ,

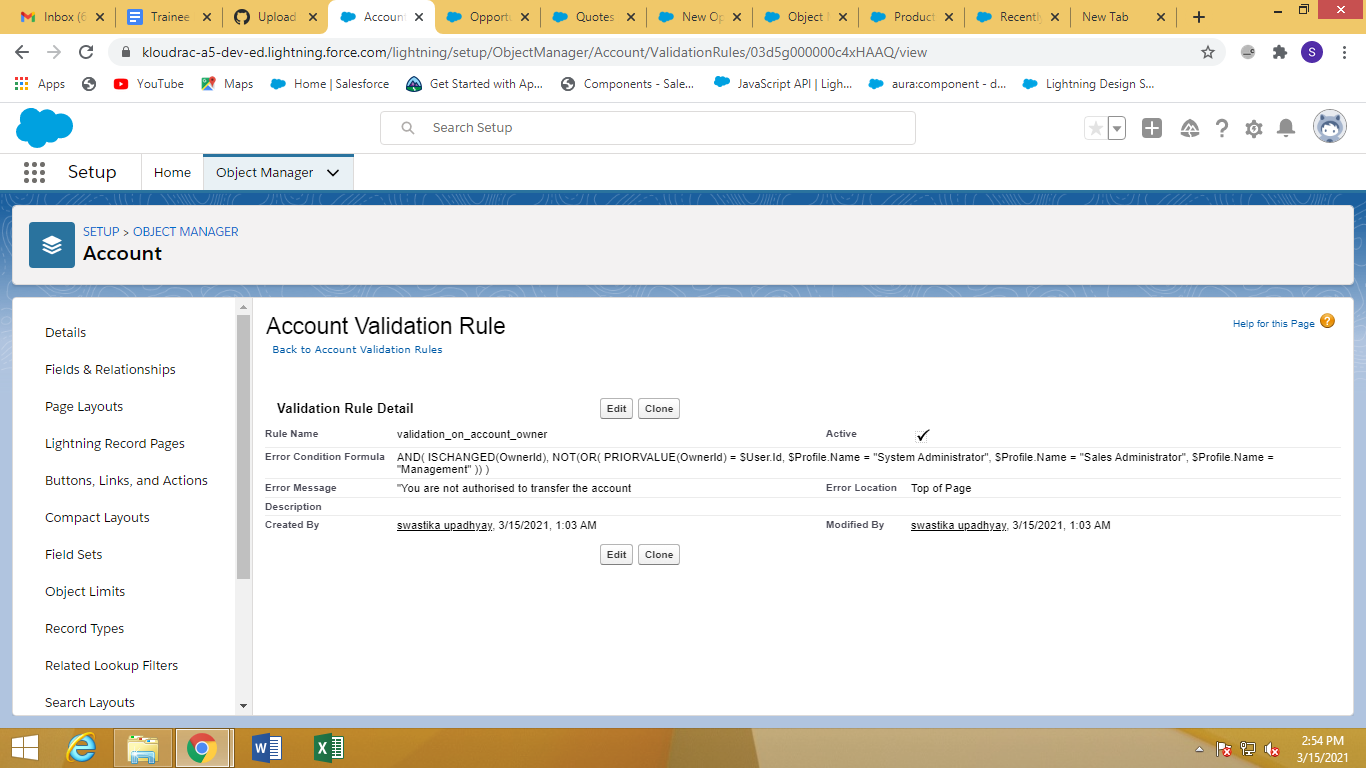
it should give the error message "Opportunity Line Item should be present while closing the opportunity.





1. Only Admins should be allowed to change the account owner.

Error Message : "You are not authorised to transfer the account."



1. **Workflow Rule (5 Marks)**

If Amount > $10,000 send an email alert on opportunity **closed won (Stage field value)** to pgoyal@kloudrac.com and msharma@kloudrac.com

Email Template .

Subject : Deal closed Alert - $Amount - Opportunity Name

Opportunity Name :

Account Name :

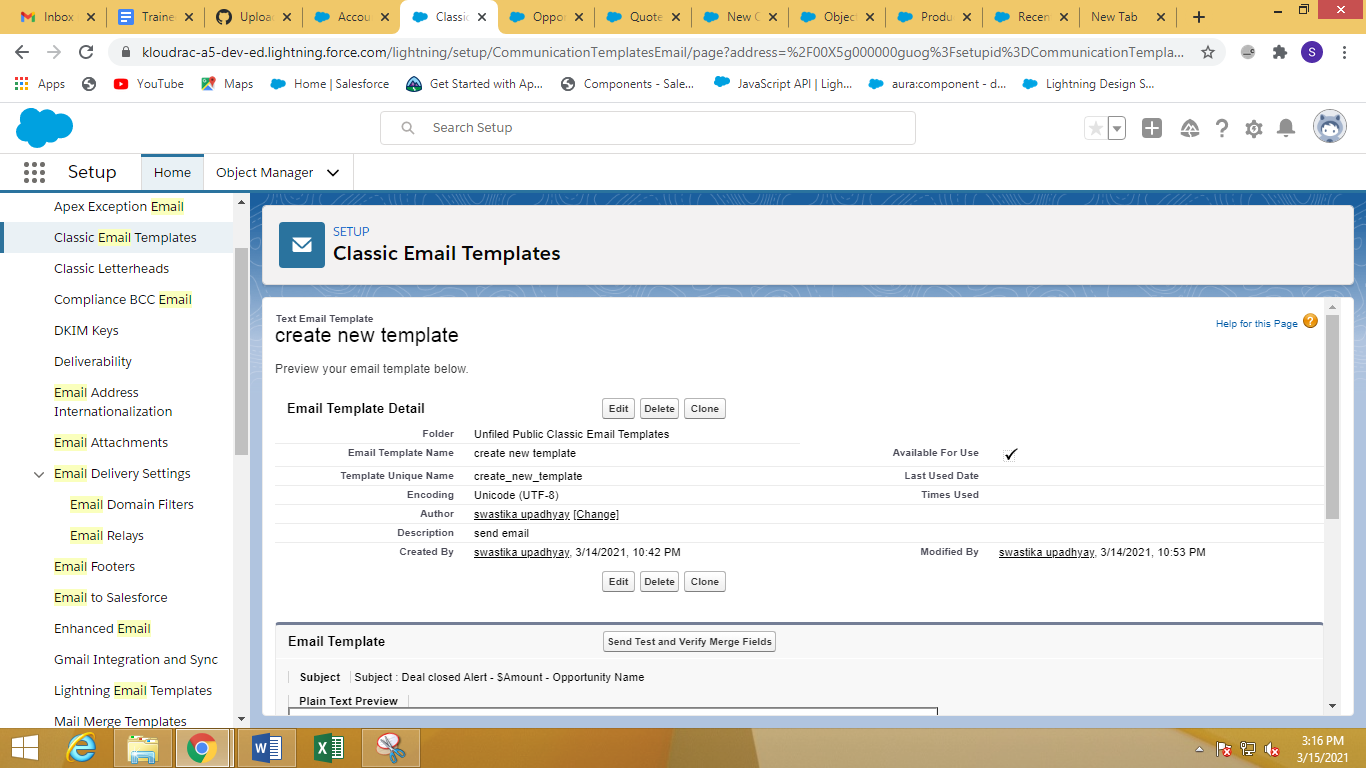
Account Owner :

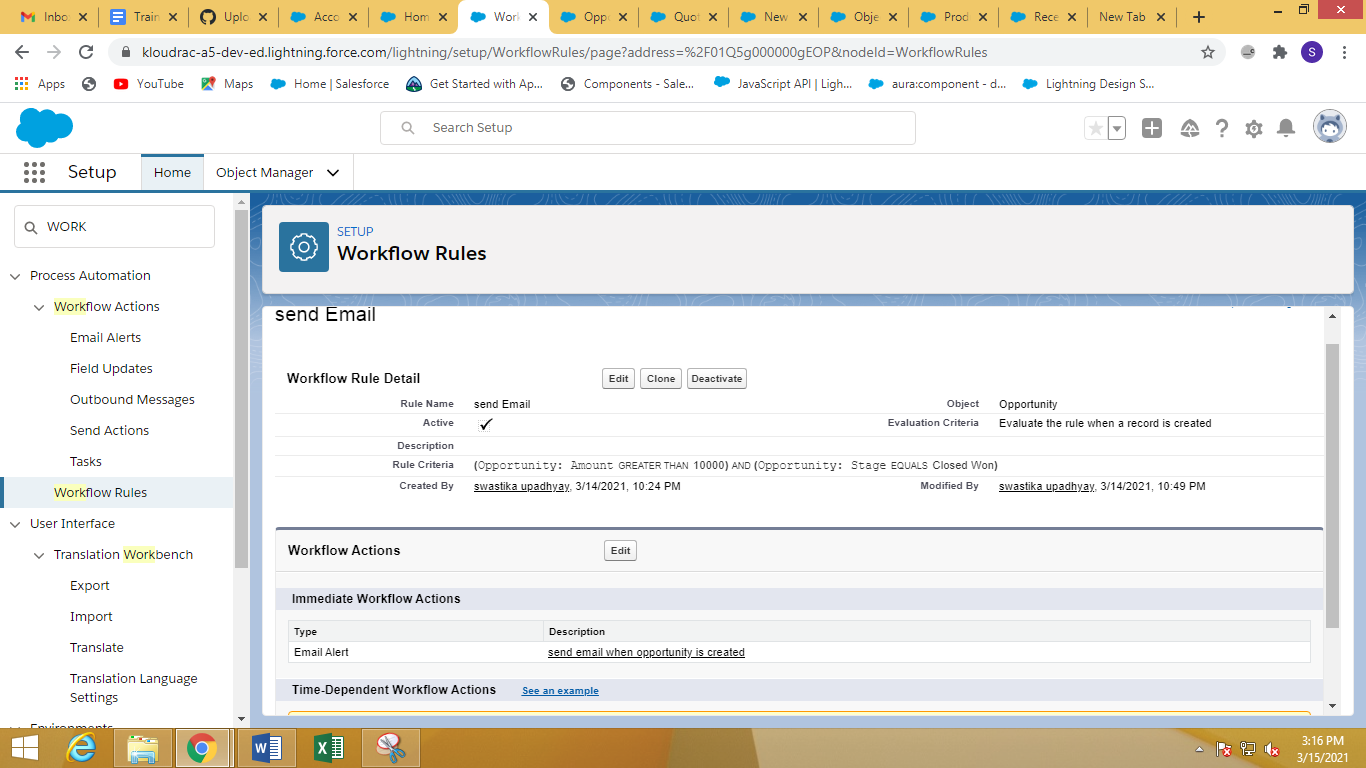
Amount :

Close Date :

Best Regards

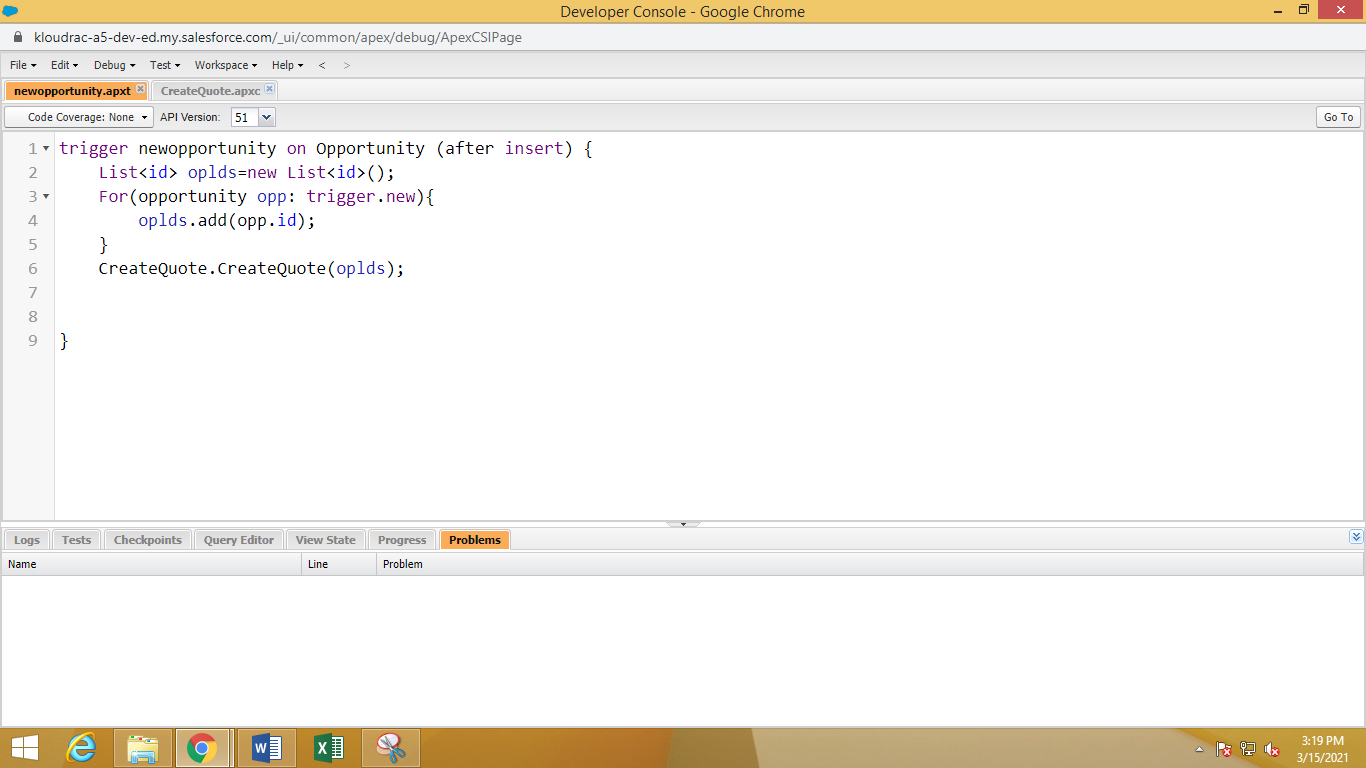
Salesforce Team - Kloudrac

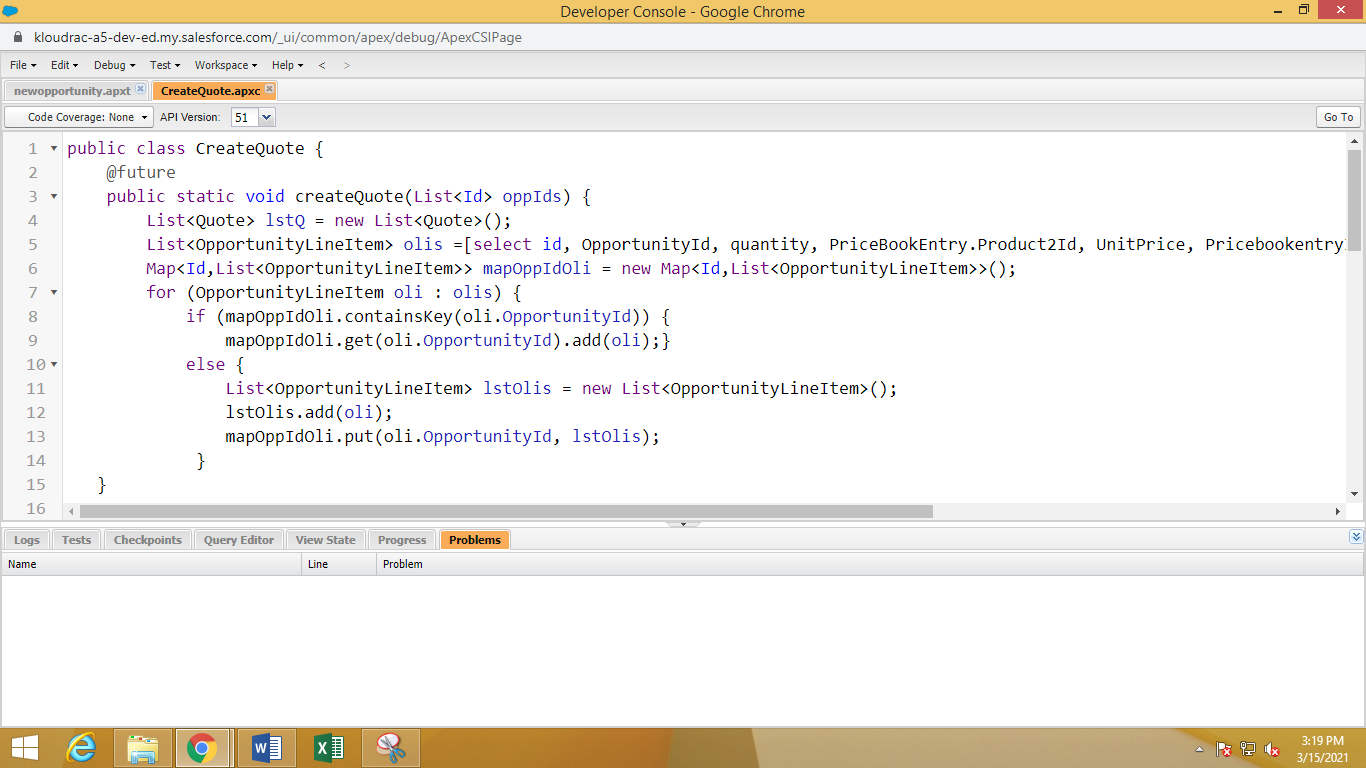


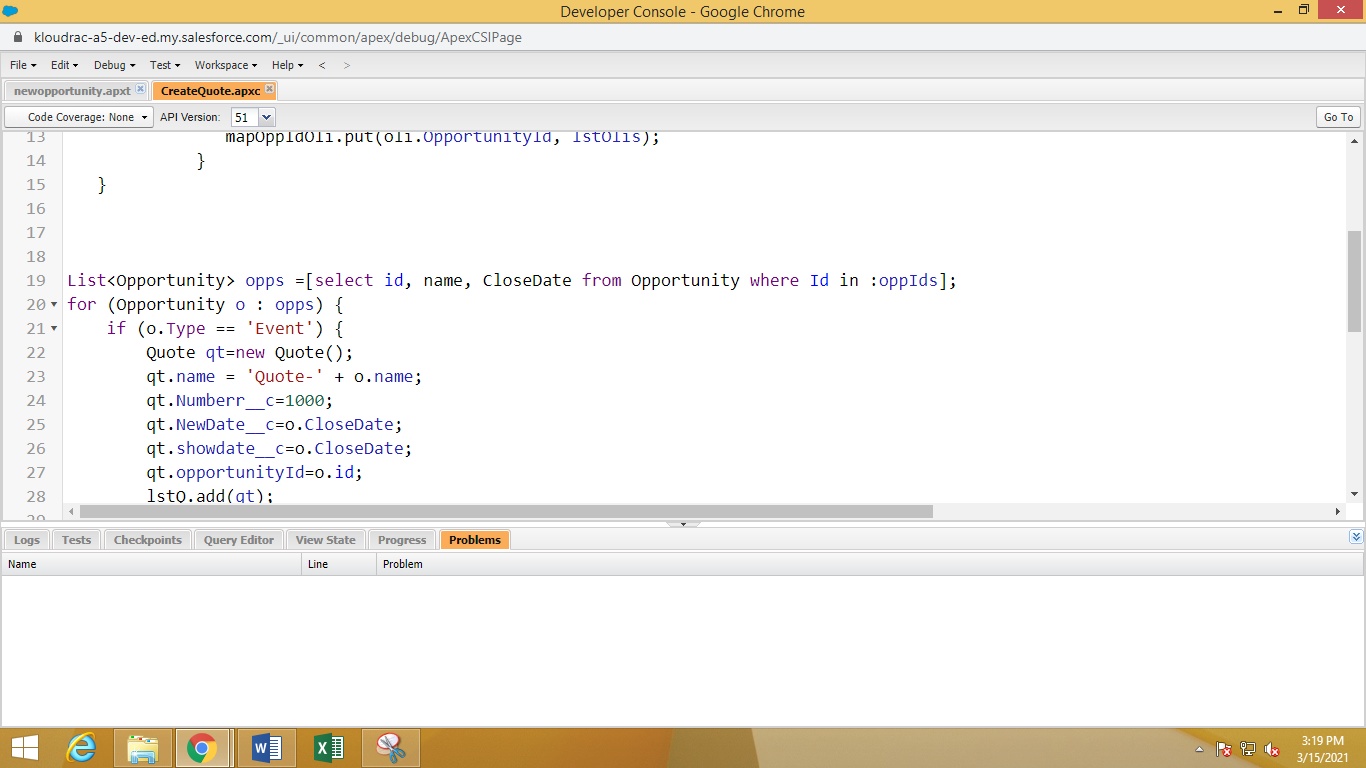


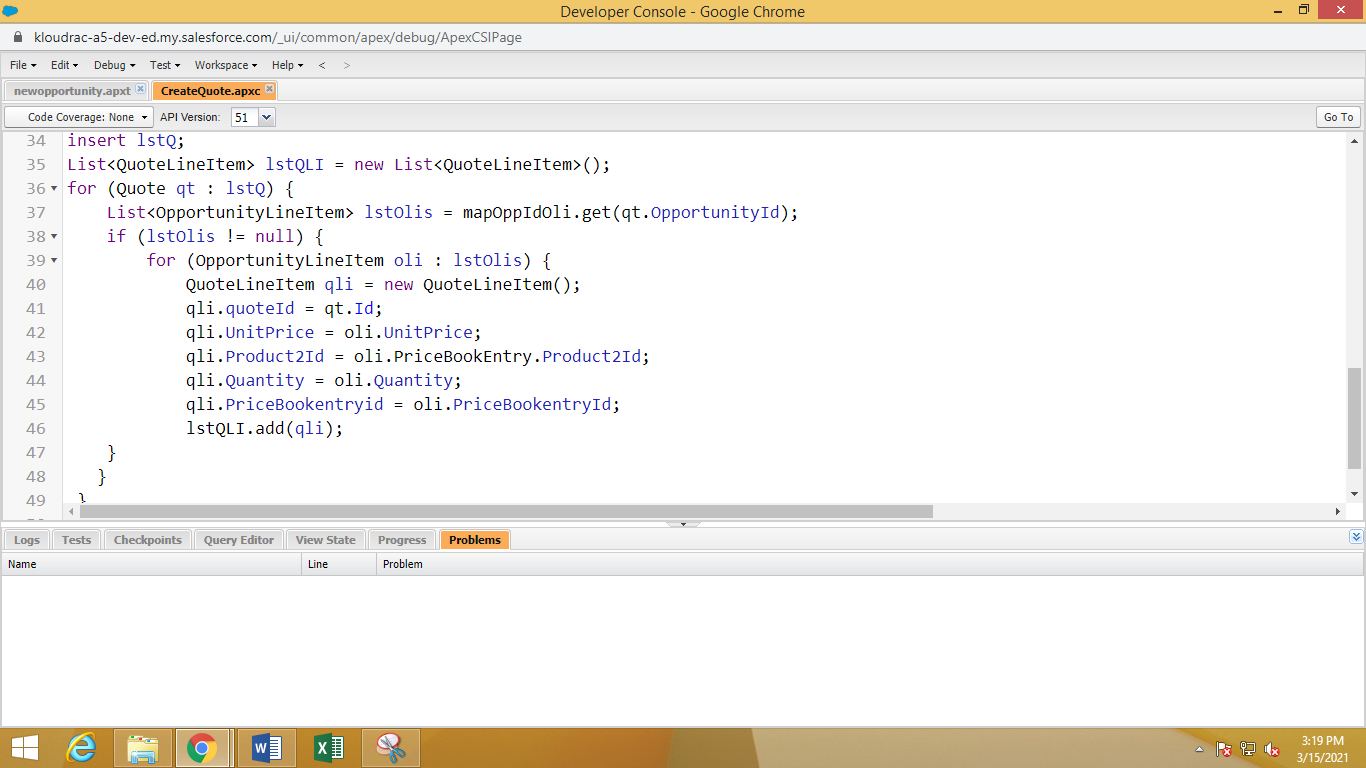
Section c

3)Quote and Quote Line Item should get auto created while Opportunity revenue is greater than 100000.









(Trigger Handler)

public class CreateQuote {

@future

public static void createQuote(List<Id> oppIds) {

List<Quote> lstQ = new List<Quote>();

List<OpportunityLineItem> olis =[select id, OpportunityId, quantity, PriceBookEntry.Product2Id, UnitPrice, PricebookentryId from OpportunityLineItem where OpportunityId in :oppIds];

Map<Id,List<OpportunityLineItem>> mapOppIdOli = new Map<Id,List<OpportunityLineItem>>();

for (OpportunityLineItem oli : olis) {

if (mapOppIdOli.containsKey(oli.OpportunityId)) {

mapOppIdOli.get(oli.OpportunityId).add(oli);}

else {

List<OpportunityLineItem> lstOlis = new List<OpportunityLineItem>();

lstOlis.add(oli);

mapOppIdOli.put(oli.OpportunityId, lstOlis);

}

}

List<Opportunity> opps =[select id, name, CloseDate from Opportunity where Id in :oppIds];

for (Opportunity o : opps) {

if (o.Type == 'Event') {

Quote qt=new Quote();

qt.name = 'Quote-' + o.name;

qt.Numberr\_\_c=1000;

qt.NewDate\_\_c=o.CloseDate;

qt.showdate\_\_c=o.CloseDate;

qt.opportunityId=o.id;

lstQ.add(qt);

}

}

insert lstQ;

List<QuoteLineItem> lstQLI = new List<QuoteLineItem>();

for (Quote qt : lstQ) {

List<OpportunityLineItem> lstOlis = mapOppIdOli.get(qt.OpportunityId);

if (lstOlis != null) {

for (OpportunityLineItem oli : lstOlis) {

QuoteLineItem qli = new QuoteLineItem();

qli.quoteId = qt.Id;

qli.UnitPrice = oli.UnitPrice;

qli.Product2Id = oli.PriceBookEntry.Product2Id;

qli.Quantity = oli.Quantity;

qli.PriceBookentryid = oli.PriceBookentryId;

lstQLI.add(qli);

}

}

}

}

}

(Trigger)

trigger newopportunity on Opportunity (after insert) {

List<id> oplds=new List<id>();

For(opportunity opp: trigger.new) {

oplds.add(opp.id);

}

CreateQuote.CreateQuote(oplds);

}