OpportunityHandlerClass

```
public class OpportunityHandlerClass {


    public static void opportunityAutomobileQuantity(List<Opportunity> LstOpportunity, Map<Id,Opportunity>
OldMapOpportunity){
        set<Id> opportunityIds = new set<Id>();
        for(Opportunity opp : LstOpportunity){
            if(opp.StageName =='Closed Won' ){
                opportunityIds.add(opp.Id);
            }
        }
        Map<Id,Opportunity_Automobile__c> lstOpportunityAutomobile =new
Map<Id,Opportunity_Automobile__c>([SELECT Id, Opportunity__c, Automobile__c, Quantity__c, Unit_Price__c,
Total_Price__c FROM Opportunity_Automobile__c Where Opportunity__c IN: opportunityIds]);


        set<Id> AutoInformationIds = new set<Id>();
        for(Opportunity_Automobile__c OppAuto: lstOpportunityAutomobile.values()){
            if(OppAuto.Automobile__c != null){
                AutoInformationIds.add(OppAuto.Automobile__c);
            }
        }
        List<Automobile_Information__c> lstAutomobileInfomation = new List<Automobile_Information__c>();
        Map<Id,Automobile_Information__c> MapAutomobileInformation = New
Map<Id,Automobile_Information__c>([SELECT Quantity__c, Price__c, Name, Id FROM Automobile_Information__c
WHERE Id IN: AutoInformationIds]);
        For(Opportunity_Automobile__c AutoOpp : lstOpportunityAutomobile.Values()){
            decimal num = 0;
            if(AutoOpp.Automobile__c == MapAutomobileInformation.get(AutoOpp.Automobile__c).Id &&
OldMapOpportunity.get(AutoOpp.Opportunity__c).stagename != 'Closed Won'){


                num = MapAutomobileInformation.get(AutoOpp.Automobile__c).Quantity__c- AutoOpp.Quantity__c;
                MapAutomobileInformation.get(AutoOpp.Automobile__c).quantity__c = num;
                lstAutomobileInfomation.add(MapAutomobileInformation.get(AutoOpp.Automobile__c));
            }
        }
        If(!lstAutomobileInfomation.IsEmpty()){
            update lstAutomobileInfomation;
        }



    }
}
```

OpportunityAutomobileHandler

```
public class OpportunityAutomobileHandler {
  public static void quantityErrorOnAutomobileInformation(List<Opportunity_Automobile__c>
lstOpportunityAutomobile){
    set<Id> AutomobileIds = new Set<Id>();
    For(Opportunity_Automobile__c OppAutomobile : lstOpportunityAutomobile){
      if(oppAutomobile.Automobile__c != null){
        AutomobileIds.add(oppAutomobile.Automobile__c);
      }
    }
    Map<Id,Automobile_Information__c> lstAutomobileInformation = new map<Id,Automobile_Information__c>([SELECT
Id, CreatedById, Quantity__c, Price__c FROM Automobile_Information__c WHERE Id IN: AutomobileIds]);
    For(Opportunity_Automobile__c OppAutomobile : lstOpportunityAutomobile){
      If(OppAutomobile.Automobile__c == lstAutomobileInformation.get(OppAutomobile.Automobile__c).Id &&
lstAutomobileInformation.get(OppAutomobile.Automobile__c).Quantity__c < OppAutomobile.Quantity__c){
        OppAutomobile.addError('the Number of Automobile u want are not Available !! the Automobile are Available
Count is ' + lstAutomobileInformation.get(OppAutomobile.Automobile__c).Quantity__c );
      }
    }
  }
  }
```

OpportunityAutoMobileTrigger

```
trigger OpportunityAutoMobileTrigger on Opportunity_Automobile__c (before insert, before Update) {
  if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
    OpportunityAutomobileHandler.quantityErrorOnAutomobileInformation(trigger.new);
  }
}
```

InvoiceCreation

```
public class InvoiceCreation {
  public static void OpportunityClosedwonInvoiceGeneration(List<Opportunity> lstOpportunity,
Map<Id,Opportunity>OldMapOpportunity){
    set<Id> oppIds = new Set<Id>();
    For(Opportunity opp : lstOpportunity){
      if(Opp.StageName == 'Closed Won' && OldMapOpportunity.get(opp.Id).StageName != opp.StageName){
        oppIds.add(opp.Id);
      }
    }
    List<Opportunity_Automobile__c> lstOpportunityAutomobile = [SELECT Unit_Price__c, Total_Price__c,
Automobile__c, Quantity__c,Opportunity__c, Id FROM Opportunity_Automobile__c WHERE Opportunity__c IN: oppIds];
    List<Invoice__c> lstInvoice = new List<Invoice__c>();
    For(Opportunity_Automobile__c oppAuto : lstOpportunityAutomobile){
      Invoice__c i = new Invoice__c();
      i.Quantity__c = oppAuto.Quantity__c;
      i.Unit_Price__c = oppAuto.Unit_Price__c;
      i.Total_Price__c = oppAuto.Total_Price__c;
      i.Purchase_Date__c = date.today();
      i.Opportunity__c = oppAuto.Opportunity__c;
      lstInvoice.add(i);
    }
    if(!lstInvoice.isempty()){
      insert lstInvoice;
    }
  }
}
```

ContactRoleCheck

```
public class ContactRoleCheck {
 public static void CheckcontactRoleonOpportunity(List<Opportunity> lstOpportunity,
Map<Id,Opportunity>OldMapOpportunity){
    List<OpportunityContactRole> lstContactRole = [SELECT Id From OpportunityContactRole WHERE OpportunityId IN:
OldMapOpportunity.keyset()];
    For(Opportunity opp : lstOpportunity){
      if(Opp.StageName == 'Closed Won' && OldMapOpportunity.get(opp.Id).StageName != opp.StageName){
        If(lstContactRole.isempty()){
          opp.adderror('Please add contact Role on opportunity whenever Opportunity is Going to Closed Won.');
        }
      }
    }
  }
}
```

OpportunityTrigger

```
trigger OpportunityTrigger on Opportunity (before update, After Update) {
  if(trigger.isbefore && trigger.isUpdate){
     OpportunityHandlerClass.opportunityAutomobileQuantity(trigger.new, trigger.oldMap);
    ContactRoleCheck.CheckcontactRoleonOpportunity(trigger.new, trigger.oldMap);
  }
  IF(trigger.isafter && trigger.isupdate){
    InvoiceCreation.OpportunityClosedwonInvoiceGeneration(trigger.new, trigger.oldMap);
  }
}
```