Order Trigger Handler:

```
public class OrderTriggerHandler {
    public static void validateOrderQuantity(List<HandsMen_Order__c> orderList) {
        for (HandsMen_Order__c order : orderList) {
            if (order.Status__c == 'Confirmed') {
                if (order.Quantity__c == null || order.Quantity__c <= 500) {
                    order.Quantity__c.addError('For Status "Confirmed", Quantity must be more than 500.');
                }
            } else if (order.Status__c == 'Pending') {
                if (order.Quantity__c == null || order.Quantity__c <= 200) {
                    order.Quantity__c.addError('For Status "Pending", Quantity must be more than 200.');
                }
            } else if (order.Status__c == 'Rejection') {
                if (order.Quantity__c == null || order.Quantity__c != 0) {
                    order.Quantity__c.addError('For Status "Rejection", Quantity must be 0.');
                }
            }
        }
        System.debug('All records validated successfully.');
    }
}
```

Order Total Trigger:

```
trigger OrderTotalTrigger on HandsMen_Order__c (before insert, before update) {
   Set<Id> productIds = new Set<Id>();


   for (HandsMen_Order__c order : Trigger.new) {
      if (order.HandsMen_Product__c != null) {
         productIds.add(order.HandsMen_Product__c);
      }
   }


   Map<Id, HandsMen_Product__c> productMap = new Map<Id, HandsMen_Product__c>(
      [SELECT Id, Price__c FROM HandsMen_Product__c WHERE Id IN :productIds]
   );


   for (HandsMen_Order__c order : Trigger.new) {
      if (order.HandsMen_Product__c != null &&
productMap.containsKey(order.HandsMen_Product__c)) {
         HandsMen_Product__c product = productMap.get(order.HandsMen_Product__c);
         if (order.Quantity__c != null) {
            order.Total_Amount__c = order.Quantity__c * product.Price__c;
         }
      }
   }
}
```

Inventory Batch Job:

```
global class InventoryBatchJob implements Database.Batchable<SObject>, Schedulable {
global Database.QueryLocator start(Database.BatchableContext BC) {
return Database.getQueryLocator(
'SELECT Id, Stock_Quantity__c FROM Product__c WHERE Stock_Quantity__c < 10'
);
}
global void execute(Database.BatchableContext BC, List<SObject> records) {
List<HandsMen_Product__c> productsToUpdate = new List<HandsMen_Product__c>();
// Cast SObject list to Product__c list
for (SObject record : records) {
HandsMen_Product__c product = (HandsMen_Product__c) record;
product.Stock_Quantity__c += 50; // Restock logic
productsToUpdate.add(product);
}
if (!productsToUpdate.isEmpty()) {
try {
update productsToUpdate;
} catch (DmlException e) {
System.debug('Error updating inventory: ' + e.getMessage());
}
}
}
global void finish(Database.BatchableContext BC) {
System.debug('Inventory Sync Completed');
}// Scheduler Method
global void execute(SchedulableContext SC) {
InventoryBatchJob batchJob = new InventoryBatchJob();
Database.executeBatch(batchJob, 200);
}
}
```

Stock Deduction Trigger:

```
trigger StockDeductionTrigger on HandsMen_Order__c (after insert, after update) {
  Set<Id> productIds = new Set<Id>();
  for (HandsMen_Order__c order : Trigger.new) {
    if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
      productIds.add(order.HandsMen_Product__c);
    }
  }
  if (productIds.isEmpty()) return;
  // Query Inventory where Product__c (Master-Detail to HandsMen Product) matches
  Map<Id, Inventory__c> inventoryMap = new Map<Id, Inventory__c>(
    [SELECT Id, Stock_Quantity__c, Product__c
     FROM Inventory__c
     WHERE Product__c IN :productIds]
  );
  List<Inventory__c> inventoriesToUpdate = new List<Inventory__c>()
  for (HandsMen_Order__c order : Trigger.new) {
    if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
      Inventory__c matchingInventory = inventoryMap.get(order.HandsMen_Product__c);
      if (matchingInventory != null) {
        matchingInventory.Stock_Quantity__c -= order.Quantity__c;
        inventoriesToUpdate.add(matchingInventory);
      }
    }
  }
  if (!inventoriesToUpdate.isEmpty()) {
    update inventoriesToUpdate;
  }
}
```

Order Confirmation Email Template:-

Dear {!HandsMen_Order_c.HandsMen_Customer_c},

Your order #{!HandsMen_Order__c.Name} has been confirmed!

Thank you for shopping with us.

Best Regards,

Sales Team

Low Stock Alert Email Template:-

Dear Inventory Manager,

This is to inform you that the stock for the following product is running low:

Product Name: {!Inventory_c.HandsMen_Product_c}

Current Stock Quantity: {!Inventory_c.Stock_Quantity_c}

Please take the necessary steps to restock this item immediately.

Best Regards,

Inventory Monitoring System

Loyality Points Email Template:-

Congratulations! You are now a {!HandsMen_Customer_c.Loyalty_Status_c} member and you are eligible for our Loyalty Rewards Program.

Enjoy exclusive discounts, early access to offers, and special member benefits.

Thank you for your continued Support.