1. Design Document (8)
2. Dockerization and Default Data (8)
3. Create a shell script which can go through order details and check if items ordered are in supplies. If items are found, then update demand\_type to SCHEDULE\_ORDER and if not let it bas as OPEN\_ORDER. Shell script should call a rest api which should perform all these business logic. (⅝)
   1. Search only for OPEN\_ORDERs.
   2. If items in orders are in new + onhand supplies, change to schedule\_order.
      1. For item across all shipnodes, quantity aggregate
   3. Return “success”
   4. Shell-Script: Make an API Call + Display (3)
4. On View Order page, if demand\_type is SCHEDULE\_ORDER, change update order link to Fulfill order else it should be update order only.
   1. Check demand type and change update button to fulfill button (3)
5. Fulfill order should update the order with actual delivery date and update the demand\_type to ALLOCATE\_ORDER.
   1. Front-end send actual delivery day in nice format + order id (3)
   2. Change demand type on back end (3)
6. Create a Shell script to check in order details for demand\_type =ALLOCATE\_ORDER and delivery\_date is today\_date, then update the demand\_type as COMPLETE\_ORDER and reduce/deduct the same in item supplies. Once the order is COMPLETE\_ORDER it should not appear on VIEW page.
   1. Query where demand\_type = ALLOCATE\_ORDER and DELIVERY\_DATE = today,
      1. If items available: Update demand\_type to COMPLETE\_ORDER and deduct from item supplies table
      2. Else: Reset demand\_type to OPEN\_ORDER (5/8)
   2. Update Search page (5)
7. Create a dashboard with below features,
8. Line Graph to show the order vs price of items.
   * + 1. Query to get completed orders (3)
       2. change order table to have actual price (3)
       3. Update create order page (5) -offer lower price
       4. Update view order page (3) -show new price
       5. Update update orders (3) -allow lower price
       6. Create dashboard page (5)
       7. Create a line graph (5)
9. Pie Chart showing the percentage of different category of items sold.
   * + 1. Create graph (8)