**RELIANCE JIOMART INVENTORY MANAGEMENT**

**TASK1:**

We have written APIs for whole inventory management like getStockpoint(store), getItem(item), getInventory(quantity of item available in that store)

**TASK2:**

**301k** stores 17 millions of items crunching 3 million reads and 100 k writes per day,   
Managing the inventory in real time using kafka   
Inventory add: from store, Inventory deplete: when customer buys a produc

Whole logic is in backend, when an item goes to out of stock or comes to instock then a request is made to algolia basically an api call and algolia updates the status of item in PDP product display page.   
  
**TASK3:**Updating the the status of product to Google Merchant GMC and Facebook Ads through realtime kafka. Then they show the ads according to the availability of the item in particular store.   
  
**TASK4:**

Multiple teams like Promise team – gives an estimated Delivery time to the customer based on his location , and nearby available stores , and then the inventory available in that stores.  
Pricing TEAM which updates price of the item based on demand,   
Catalog TEAM which updates the products every in most of the categories  
all the teams use common Inventory Interface to get the consistent data of inventory of items – for this we have developed a website using Angular and ReactJS, then upgraded to NextJS

**TASK5:**

I have developed a reserve functionality feature reducing transaction overhead of money by company to customers and also for better customer experience.

Available qty-> reserved qty  
when user clicks on add to cart and buy now  
if payment success then move reserved qty to depleted qty  
if payment failure we move the reserved qty to available qty after 5 min which is expired time given  
and it is done by using cron job in java which runs every minute to fetch expired cart in java  
and fetching expired cart -> making cart inactive -> reverting reserved qty everything will be in one transaction   
and then I have JSON structure in MYSQL to store the store: item : qty objects