**Opply Data Assignment - (Ernest Amoako)**

**Insight Generation Segment:**

*Further insights into measures that could be taken to increase the request to order conversion ratio*:

* I believe a robust monitoring script that iterates over active requests and prompts action based on a determined threshold of ***Quote Delay,*** would be useful***.*** By this I mean that once we determine what an appropriate cutoff is (through statistical rigor) we could create a grouping of buyers who are likely to “churn”.
* Analysis on like-products to determine what an incoming product typically behaves as:
  + For instance if a buyer is after Cassava Flour and we have enough data to create an accurate profile, then we could alert the customer if their request is far outside the norm for usual successful requests.
  + Highlighting input fields to say that the Target Unit Price is far outside the norm may prompt a more reasonable choice.
  + Similar to the above for the price they are inputting. Perhaps prompting buyers to double check the unit price is the correct value being input.
* Creating successful conversion profiles of varying degrees.
* Active allocation and prompts to suppliers after decline from one buyer. Actively submit to them other buyers with similar needs and statistically determined profiles that they may be interested in. This is to prevent any potential stagnation.
* To expand on the tooltip example given in brief: we could possibly display information about products that are contextually available/successful in recent days. This expands on supplying context data as previously mentioned.

**Additional Question (Web Scraping):**

*Web scraping for prices for ingredients to know whether quotes are competitive*:

Several options exist at varying degrees of usability.

* For internal team members, we are able to set up automations through tools such as ParseHub to scan over pages of the website in question.
  + The exported results could be read into a collection and a script could be written to alert of major changes and highly competitive prices.
* A server-side script running a combination of python utils (requests, BeautifulSoup, and Selenium) would be able to perform this kind of monitoring.
  + Aligning with the site’s bot activity would be the main challenge here.