# ASSUMPTIONS

USER SERVICE ASSUMPTIONS:

* Ideally the authentication service would be a separate microservice implemented with a standard Authentication library like OAuth 2, but for the scope of this project, I have used a Base64 encode/decode of the user\_id as a token, which is not session based. The token will be valid forever.
* LogOut of User is not implemented.
* User can access all the products of the inventory, but cannot access other user’s cart or orders.

PRODUCT SERVICE ASSUMPTIONS

* Product can be added or deleted in the inventory by an Admin user ONLY.
* While deleting a product, admin can only entirely remove the product from the inventory. He/she cannot delete a quantity of product.
* Product would be a separate Microservice, as there would be heavy read as well as write on the service. There would be frequent searching as well as updations as orders are placed.
* Also, many more enums can be created for unitOfMeasurements etc.
* Also, products can be grouped by merchants etc.

ORDER SERVICE ASSUMPTIONS

* An order can only be cancelled if it is placed.

CART SERVICE ASSUMPTIONS:

* User can access only one’s own cart.
* Once a user checks out, and order is created the product stock is decreased.
* A user and cart are associated, and there would be frequent access between the two, which is why it would be a separate microservice.

# IMPROVEMENTS:

There are a lot of improvements which can be done in future:

* Services can be divided into microservices, like User Service, ProductAccessService (which serves high number of read requests for products), Order & Payement Service.
* Products can be divided into various departments like Food, Beverages, Household items, Personal Wellness etc.
* Expiry date can be mandatory for food items but not for others.
* Various payment methods can be supported like UPI, Wallets etc
* For input to the API’s, entities should not be directly used, but POJO’s should be used.
* Shipment, Tracking and Refund functionalities can be added.
* Order can have various Statuses, rather than just created/placed/deleted.