I. Auto complete Suggestions

* Text entered

Wait for entry of at least 3 character

If less than 3 or blank cursor- only keyboard, if keyboard closed, show page with frequent searches.

If 3 or more then suggestions will be there, if keyboard closed that suggestions window will persist

Term as query

Ne ed for any pre-processing?

* Query updates with each addition or deletion (maintain at least 3 character rule)
* From elastic search get suggestions - top most will be the text entered, 5 text suggestions, 2 salt suggestions, 3 product suggestions

Example :

query: “para”

Text suggestion 1: “para”

5 more suggestions: “paracetamol”, “paracetamol tablet”, “paracetamol injection”, “paracetamol tablet 500mg”, “paracetamol 500mg”

2 salt suggestions: “Paracetamol”, Paracetamol+aceclofenac”

3 product suggestion: “Tablet Paracetamol 500mg - 10 tablets along with Salt or category (Manufacture name)

Tablet Paracetamol 500mg - 10 tablets

*Paracetamol (Cipla)*

Paradye Hair Shampoo - 100 ml

*Personal Care (Paradyes)*

* For salt suggestions- will get salt name and id from elastic search
* For product suggestions - get id, product name with pack, salt/category, mfg name, is\_healthproduct -yes/no
* For salt and product suggestions- clicking will redirect to salt/product page directly (here to redirect to medicine/health product page do we need health\_product yes/no or we can directly get that from id?

II. Search Listing Page

* When search icon clicked or enter pressed (in desktop) - query term along with location (lat/lon) send to backend
* Query term to elastic search
* lat/lon to our backend to calculate distance and filter pharmacies nearby along with products available
* By default, all page will be shown
* But if someone in salt page/product page-then modify/reenter search they will go back to the respective page
* Irrespective of the page- we will load all product and salt results from elastic search.
* from elastic search
* For products will get these- id, name\_with\_pack, salt/category, mfg name,
* For salt- salt\_id, name, Available forms, most\_common\_forms, JSON array of form\_strength\_packing, ids of medicines within each salt\_strength\_packing
* Get price from db
* we get relevant ids from elastic\_search and subset of nearby\_pharmacies (No.3 above)
* Check for the ids from products (7.a) as well as ids from salt tab (7.b) in inventory\_db of subset pharmacy
* Get MRP of each product, along with Price as array

{

Product id: xxx

MRP: xx

Price { pharmacy ID1 :Price1,Pharmacy ID2:Price2}

}

\*Note for salts- there will be multiple product ids- so we get price for all the product ids

* Display results
* In all page, medicine page, product page - first sort by availability (all available before not available), the sort by relevance score. Sample below

Available 1, rel score -10

Available 1, rel score - 8

Available 1, rel score - 2

Available 0, rel score -15

Available 0, rel score -10

Available 0, rel score -4

* For salts- availability in any nearby pharmacy inventory.
* Get the available form\_strength\_packing array along with product ids from elastic search
* For each product in the nested array - check if present in the filtered inventory db and associate - price along with pharmacy id of each product id and create a new array with ordering the forms,strengths and packing based on the occurrences of prices (this will have the available variants in front). We will call it mergedarray.
* Default selection will be the first listed variant (if we are ordering based on count of prices - available ones will be displayed at first)
* As the user changes the combination, prices and variants changes dynamically based on the information in mergedarray - to be stored as cache till new search term entered
* For salts not available nearby:
* No price or add to cart
* Display available forms- can select
* Default selection- one in most common

III. Product Details Page