# Code review submission

Hello Jaiben,

I am glad to be shortlisted for the role and considered for assessment stages.

Coming to main point, I have read the code snipped bit by bit here are more observations and suggestions.

PS: **I have documented the errors on code file and prepared a correct version too. Please check it out.**

**“All my error findings and suggestions are based on observing this code snippet only, If I have little more info about the project story, I would definitely come up with more findings”**

**Logic flow:** I am explaining the flow assuming that there is no error in code, i.e. how this module should work ideally. Later, I will point out all error in next heading.

**In nutshell**, this middleware implements the feature in which we take user query of some medicine (either name or salt) and user’s location. Then we respond with matched products which are completely matched with query, and some are partial matched. we Also return the nearby pharmacies where these product is available, along with selling price.

**In details..**

1.This is search module which takes user query string and pharmacyIds. User string could be either name of the medicine like **Dolo** or salt like **paracetamol**. We are accommodating both the possibilities. Query could consist of many space separated names like **“Clindamycin Corex Nicotinamide”**.

2. we extract the query string from request object and pharmacyIds from request object’s body.

3.Now we will create a Elasticsearch query to search from salt index (salt database) and medicine index.

4.Above query use multi search API of Elasticsearch client instance to fetch data from two indexes with single http request to reduce network overhead.

5. then we are populating a response object which has three fields

(a). Salt suggestion array. In which each object consists of salt details and its array of pharmacy where salt is available.

(b). Medicine suggestion array. In which each object consists of medicine details and its array of pharmacy where medicine is available.

(c). Health suggestion array. In which each object consists of health product details and its array of pharmacy where this product is available.

. This populating part utilizes another two methods to get the work done.

All other methods are well commented to explain their functionality.

# Major error/mistakes findings.

Code snippet has many syntactical, logical errors which are clearly observable. These are..

1. Constants that are potentially to be environment variables are not written as process.env.<variable name>.

Such as-

Const OPENSEARCH\_ENDPOINT=OPENSEARCH\_URL.

Here OPENSEARCH\_URL is not initialized,

To initialize it as environment variable.  
 Const OPENSEARCH\_ENDPOINT=process.env.OPENSEARCH\_URL.

Now during application runtime, this variable will be initialized from .env file or command line.

Similarly other two variables are to be corrected similarly.

1. There are no instance for Elasticsearch client , database and redisclient. First import or require the needed dependencies, create instance then we can use them.

Like - for Elasticsearch

**const { Client } = require('@elastic/elasticsearch');**

**const client = new Client({ node: OPENSEARCH\_ENDPOINT,headers });**

for redisclient—

**import { createClient } from 'redis';**

**const redisClient = createClient();**

**redisClient.on('error', err => console.log('Redis Client Error', err));**

**await redisClient.connect({ url: <url of your redis server>});**

**// similarly we need to add database dependencies and create instance .**

1. **Create\_redis\_inv()** function is called even before its declaration, since this is a function expression, it is not hoisted. It means we must first write function expression then we can invoke it.

So this has to be moved above from **Create\_redis\_inv()** invocation. Or make it a declaration.

1. Similarly **getProduct()** and  **getNearestPharmacy()** are also invoked before the function expression.   
   Either make them function declaration or move the function expression top of its invocation so that they can be first initialized before invocation.
2. There is performance issue inside **extractSuggestion()** function which time complexity can be optimized from O(n+n) to O(n) only.

**I have commented all my findings on given code snippet and created a corrected version of it**