STACKLINE SDE ASSESSMENT

Goal of this assignment is to build an api. The api should use the s3 file (https://s3-us-west-2.amazonaws.com/stackline-public/sample_product_data.tsv.gz) as the dataset.
The S3 file contains product data for 50k products. The index.html (provided) can be used to test your api visually.

1. Auto complete api for brand, category and product title data
    GET /api/product/autocomplete?type=<brandName>&searchTermPrefix=<term>
    Response Json Body: (Sample for autocomplete type brand)

```
{
     "type": "brand"
      "suggestions" : [ {
       "id": "<BrandId>",
        "name": "<BrandName>"
      }]
}
```

2. Search api that takes json requests (below) and searches the products loaded from the file. The conditions field in the request is an array with unique fieldname for each condition.

The returned products has to match all the conditions provided. The api has to also support pagination. You are not allowed to use any external search services to accomplish the task.
POST /api/product/search
Request Json Body

```
        {
 "conditions": [
  {
   "fieldName": "title",
   "values": [
    "val1",
    "val2"
   ]
  },
  {
   "fieldName": "brandId",
   "values": [
    "val1",
    "val2"
   ]
  },
  {
   "fieldName": "categoryId",
   "values": [
    "val1",
    "val2"
   ]
  }
 ],
 "pagination": {
   "from": 1
   "size": 50
```

```
 }
}
      Response Json Body:
           {
       "products" : [ {
          "productId": "<Title>",
          "title": "<Title>",
          "brandId": "<BrandId>",
          "brandName": "<Brand Name>",
          "categoryId": "<Category Id>",
          "categoryName":"<Category Name>",
          }
 ],
 "pagination": {
    "from": 1
    "size": 50
 }
          }
```

The completed assignment need to be a docker container image with the all the code and read me file in place.

Language you can use: NodeJS / C# /Java
What counts:
1. Working solution
2. Clean code
3. Optimized search operations
4. Data modeling and Object oriented design
5. Error handling

Time Limit: 8 hours.

SOLUTION

I have Designed and Developed this API using Java Language (Spring Boot Framework). This API is used to handle 2 operations mentioned in the goal. The API has been deployed and shared on Docker hub.
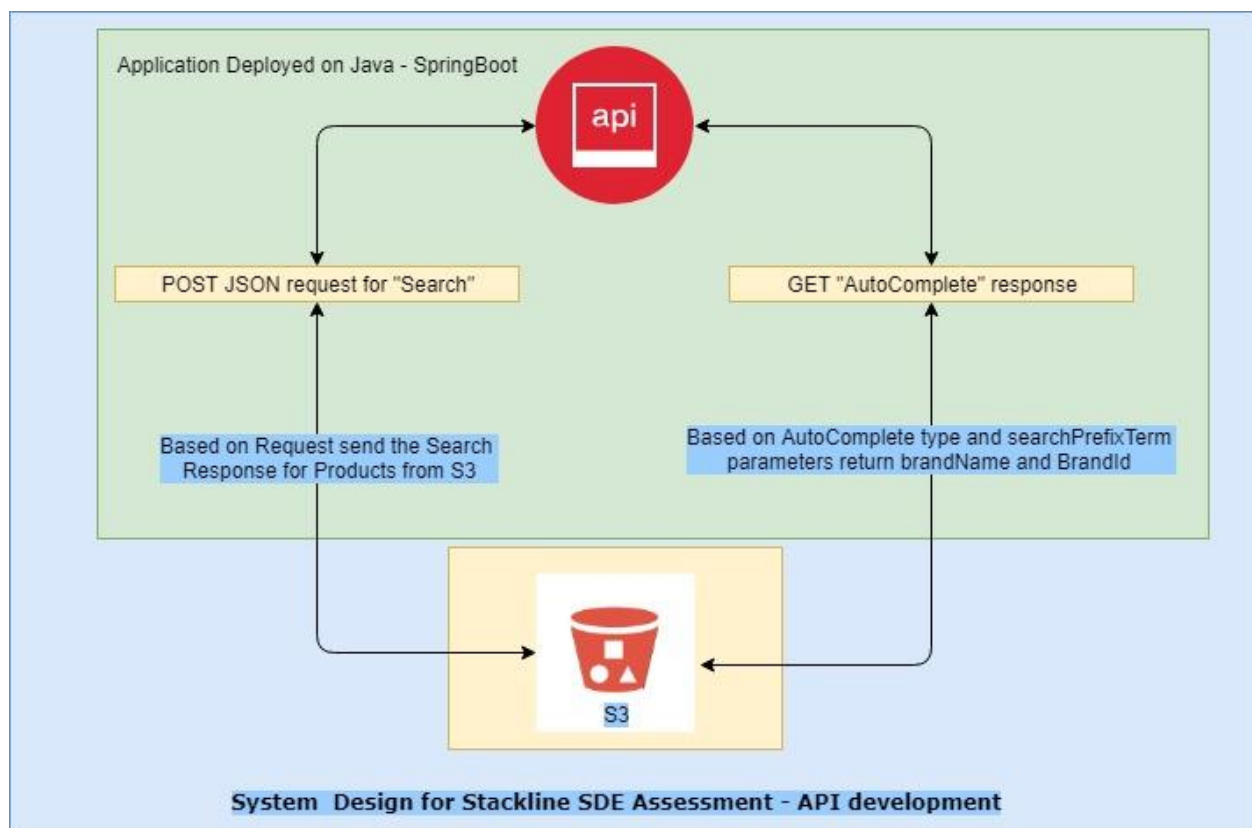
System Requirements:

- Java 8
- Postman / localhost:8080 to check the application response.
- **AWS Secret Key, Access Key, Bucket name, Key Name(File Name) is mandatory and should be inserted into the application.yml file under resources repository.**
  You should create a User in IAM with AmazonS3FullAccess to get the keys required and access required for S3.

I was supposed to implement index.html for UI but due to time for understanding some details of the API and lunch time I wasn't able to finish the design. However I have implemented it half way through which I will send again later this night.

System Design –

The assessment is challenging and interesting and I am very much excited to receive some more interesting challenges in my daily routines but I will definitely make sure to finish them in given time span.
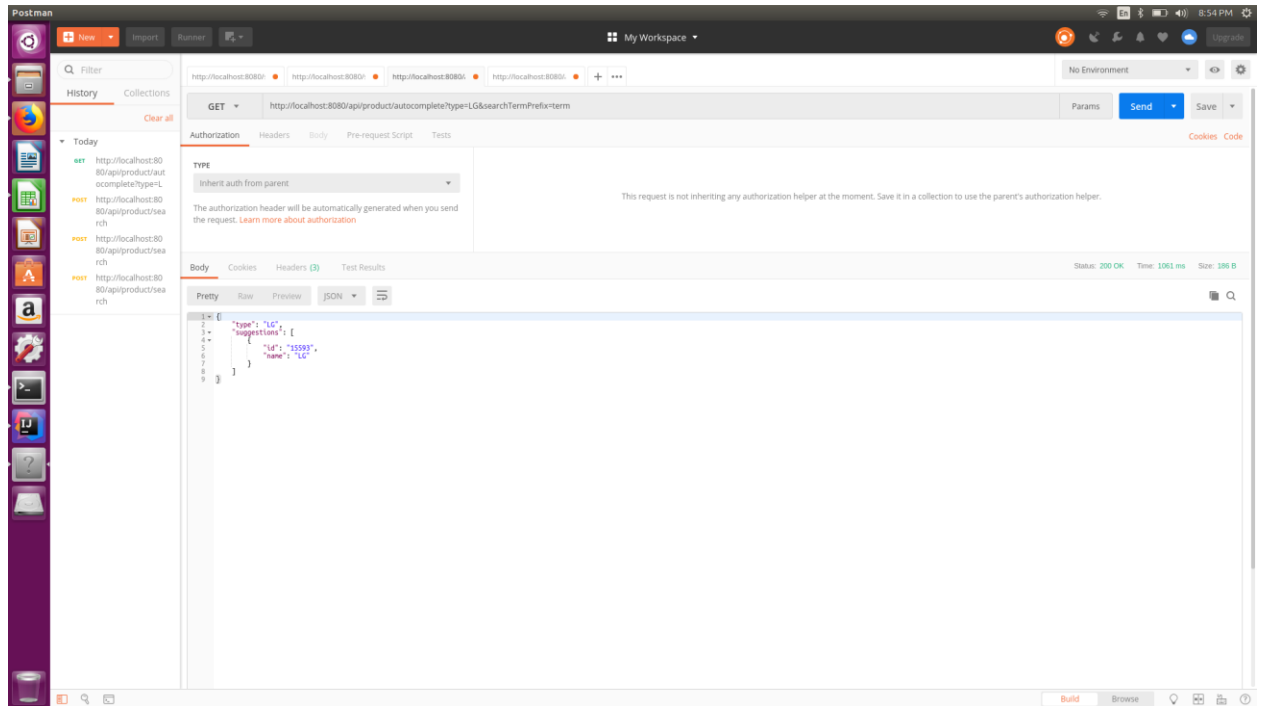
OUTPUT

POSTMAN OUTPUT –

1. **Autocomplete Search**

2. **Search**