Interview Assignment Task

Navitha D

BE CSE(2024)

1.Function definition

```
import requests
import json
SOLR_URL = "http://localhost:8983/solr/"
def indexData(p_collection_name, p_exclude_column, employee_data)
    for record in employee_data:
        if p_exclude_column in record:
            del record[p_exclude_column]

    url = f"{SOLR_URL}{p_collection_name}/update?commit=true"
    headers = {"Content-Type": "application/json"}
    response = requests.post(url, data=json.dumps(employee_data), headers=headers)

    if response.status_code == 200:
        print(f"Data indexed successfully into collection '{p_collection_name}' excluding column
'{p_exclude_column}'.")
    else:
        print(f"Failed to index data: {response.text}")

def searchByColumn(p_collection_name, p_column_name, p_column_value)
    url = f"{SOLR_URL}{p_collection_name}/select?q={p_column_name}:{p_column_value}&wt=json"
    response = requests.get(url)

    if response.status_code == 200:
        data = response.json()
        print(f"Search Results for {p_column_name} = {p_column_value}:")
```

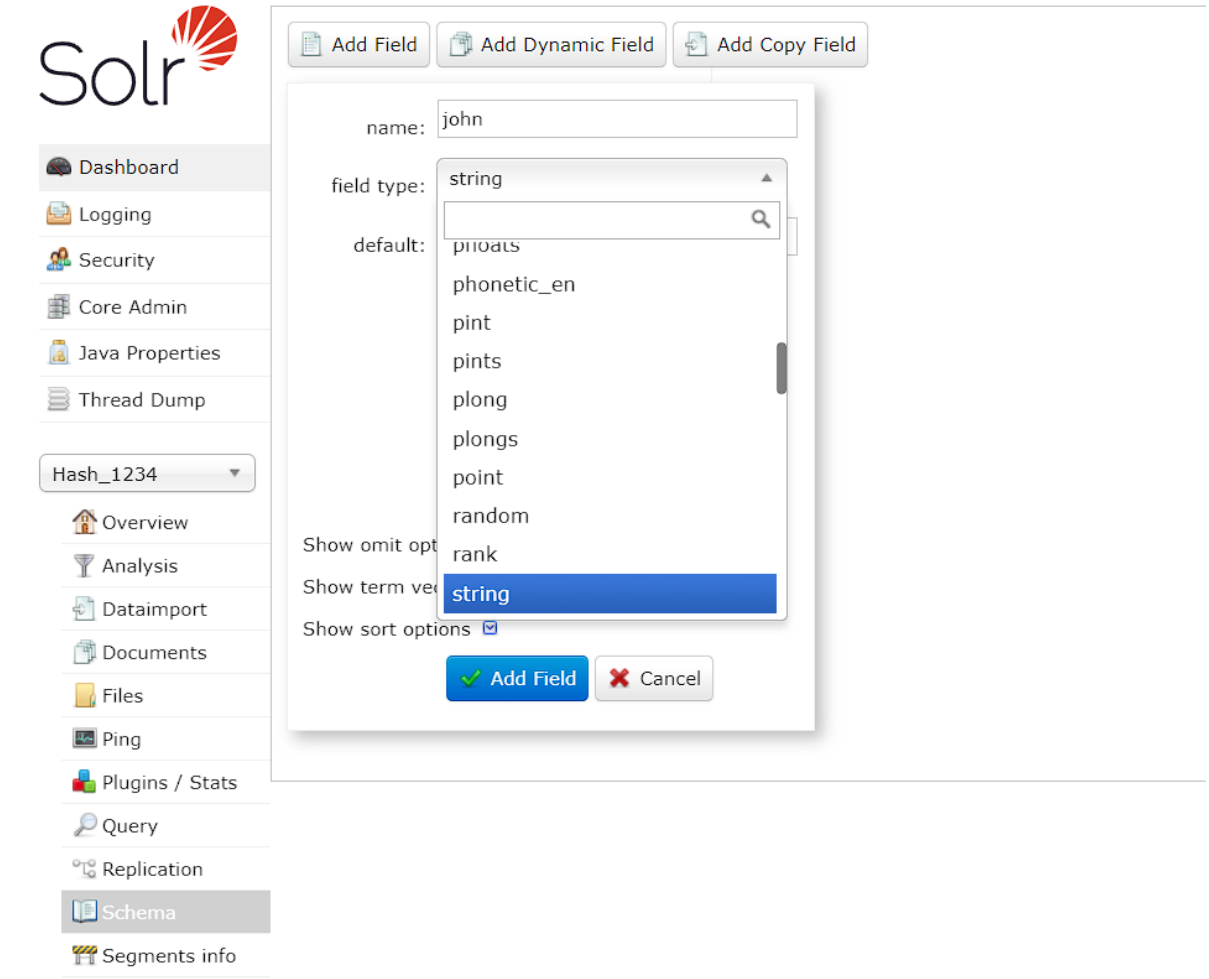
```
        for doc in data['response']['docs']:
            print(doc)
    else:
        print(f"Failed to search: {response.text}")

def getEmpCount(p_collection_name):
    url = f"{SOLR_URL}{p_collection_name}/select?q=*:*&rows=0&wt=json"
    response = requests.get(url)

    if response.status_code == 200:
        data = response.json()
        count = data['response']['numFound']
        print(f"Total number of employees in collection '{p_collection_name}': {count}")
    else:
        print(f"Failed to get employee count: {response.text}")

def delEmpById(p_collection_name, p_employee_id):
    url = f"{SOLR_URL}{p_collection_name}/update?commit=true"
    headers = {"Content-Type": "application/json"}
    delete_query = {"delete": {"id": p_employee_id}}
    response = requests.post(url, data=json.dumps(delete_query), headers=headers)

    if response.status_code == 200:
        print(f"Employee with ID '{p_employee_id}' deleted successfully.")
    else:
        print(f"Failed to delete employee: {response.text}")

def getDepFacet(p_collection_name):
    url =
f"{SOLR_URL}{p_collection_name}/select?q=*:*&facet=true&facet.field=department&rows=0&wt=js
on"
    response = requests.get(url)
```

```
    if response.status_code == 200:
        data = response.json()
        facets = data['facet_counts']['facet_fields']['department']
        print(f"Department-wise employee count for collection '{p_collection_name}':")
        for i in range(0, len(facets), 2):
            department = facets[i]
            count = facets[i + 1]
            print(f"{department}: {count}")
    else:
        print(f"Failed to retrieve department facet: {response.text}")
```

employee data to be indexed

```
employee_data = [
    {"id": "E001", "name": "John Doe", "department": "IT", "gender": "Male"},
    {"id": "E002", "name": "Jane Smith", "department": "HR", "gender": "Female"},
    {"id": "E003", "name": "Jim Brown", "department": "Finance", "gender": "Male"},
    {"id": "E004", "name": "Lucy Black", "department": "IT", "gender": "Female"}
]


collection_name = "Hash_YourName"
phone_collection = "Hash_1234"


# Create collection (normally done via Solr admin, not shown here)
# indexData(collection_name, 'department', employee_data)
# indexData(phone_collection, 'gender', employee_data
getEmpCount(collection_name
searchByColumn(collection_name, "department", "IT")
delEmpById(collection_name, "E003")
getEmpCount(collection_name)
getDepFacet(collection_name)
```

```
Microsoft Windows [Version 10.0.22631.4249]
(c) Microsoft Corporation. All rights reserved.

C:\Users\RATHISH>cd C:\Program Files\Java\jdk-21

C:\Program Files\Java\jdk-21>solr start
Java HotSpot(TM) 64-Bit Server VM warning: JVM cannot use large page memory because it does not have enough privilege to lock pages in memory.
Waiting up to 30 to see Solr running on port 8983
Started Solr server on port 8983. Happy searching!

C:\Program Files\Java\jdk-21>solr create -c Hash_YourName -shards 2 -replicationFactor 1
WARNING: Using _default configset with data driven schema functionality. NOT RECOMMENDED for production use.
         To turn off: bin\solr config -c Hash_YourName -p 8983 -action set-user-property -property update.autoCreateFields -value false

Created new core 'Hash_YourName'

C:\Program Files\Java\jdk-21>solr create -c Hash_1234 -shards 2 -replicationFactor 1
WARNING: Using _default configset with data driven schema functionality. NOT RECOMMENDED for production use.
         To turn off: bin\solr config -c Hash_1234 -p 8983 -action set-user-property -property update.autoCreateFields -value false

Created new core 'Hash_1234'
```



2.Function execution

```
import requests
import json

SOLR_URL = "http://localhost:8983/solr/"
def createCollection(p_collection_name):
    url = f"{SOLR_URL}admin/collections?action=CREATE&name={p_collection_name}&numShards=1&replicationFactor=1"
    response = requests.get(url)

    if response.status_code == 200:
        print(f"Collection '{p_collection_name}' created successfully.")
    else:
        print(f"Failed to create collection: {response.text}")
def indexData(p_collection_name, p_exclude_column, employee_data):
    for record in employee_data:
        if p_exclude_column in record:
            del record[p_exclude_column]
    url = f"{SOLR_URL}{p_collection_name}/update?commit=true"
    headers = {"Content-Type": "application/json"}
    response = requests.post(url, data=json.dumps(employee_data), headers=headers)

    if response.status_code == 200:
        print(f"Data indexed successfully into collection '{p_collection_name}' excluding column '{p_exclude_column}'.")
    else:
        print(f"Failed to index data: {response.text}")
```

```
def searchByColumn(p_collection_name, p_column_name, p_column_value):
    url =
f"{SOLR_URL}{p_collection_name}/select?q={p_column_name}:{p_column_value}&wt=json"
    response = requests.get(url)

    if response.status_code == 200:
        data = response.json()
        print(f"Search Results for {p_column_name} = {p_column_value}:")
        for doc in data['response']['docs']:
            print(doc)
    else:
        print(f"Failed to search: {response.text}")
def getEmpCount(p_collection_name):
    # Query to get the count of documents
    url = f"{SOLR_URL}{p_collection_name}/select?q=*:*&rows=0&wt=json"
    response = requests.get(url)

    if response.status_code == 200:
        data = response.json()
        count = data['response']['numFound']
        print(f"Total number of employees in collection '{p_collection_name}': {count}")
    else:
        print(f"Failed to get employee count: {response.text}")

def delEmpById(p_collection_name, p_employee_id):
    url = f"{SOLR_URL}{p_collection_name}/update?commit=true"
    headers = {"Content-Type": "application/json"}
    delete_query = {"delete": {"id": p_employee_id}}
    response = requests.post(url, data=json.dumps(delete_query), headers=headers)
```

```
    if response.status_code == 200:
        print(f"Employee with ID '{p_employee_id}' deleted successfully.")
    else:
        print(f"Failed to delete employee: {response.text}")


def getDepFacet(p_collection_name):
    url = f"{SOLR_URL}{p_collection_name}/select?q=*:*&facet=true&facet.field=department&rows=0&wt=json"
    response = requests.get(url)


    if response.status_code == 200:
        data = response.json()
        facets = data['facet_counts']['facet_fields']['department']
        print(f"Department-wise employee count for collection '{p_collection_name}':")
        for i in range(0, len(facets), 2):
            department = facets[i]
            count = facets[i + 1]
            print(f"{department}: {count}")
    else:
        print(f"Failed to retrieve department facet: {response.text}")


employee_data = [
    {"id": "E001", "name": "John Doe", "department": "IT", "gender": "Male"},
    {"id": "E002", "name": "Jane Smith", "department": "HR", "gender": "Female"},
    {"id": "E003", "name": "Jim Brown", "department": "Finance", "gender": "Male"},
    {"id": "E004", "name": "Lucy Black", "department": "IT", "gender": "Female"},
    {"id": "E02003", "name": "Tom White", "department": "IT", "gender": "Male"},
]
#order to execute the Functions
```

v_nameCollection = "Hash_YourName"

v_phoneCollection = "Hash_1234"

createCollection(v_nameCollection)

createCollection(v_phoneCollection)

getEmpCount(v_nameCollection)

indexData(v_nameCollection, 'department', employee_data)

indexData(v_phoneCollection, 'gender', employee_data)

delEmpById(v_nameCollection, 'E02003')

getEmpCount(v_nameCollection)

searchByColumn(v_nameCollection, 'department', 'IT')

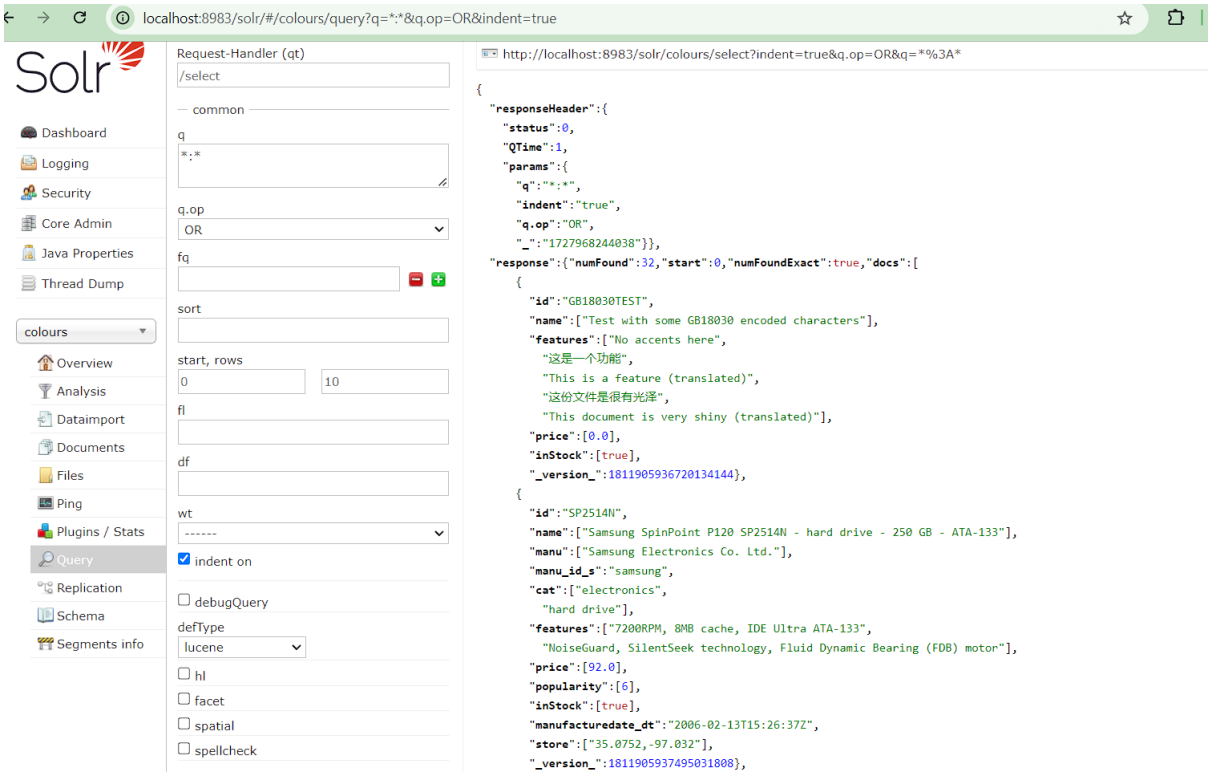
searchByColumn(v_nameCollection, 'gender', 'Male')

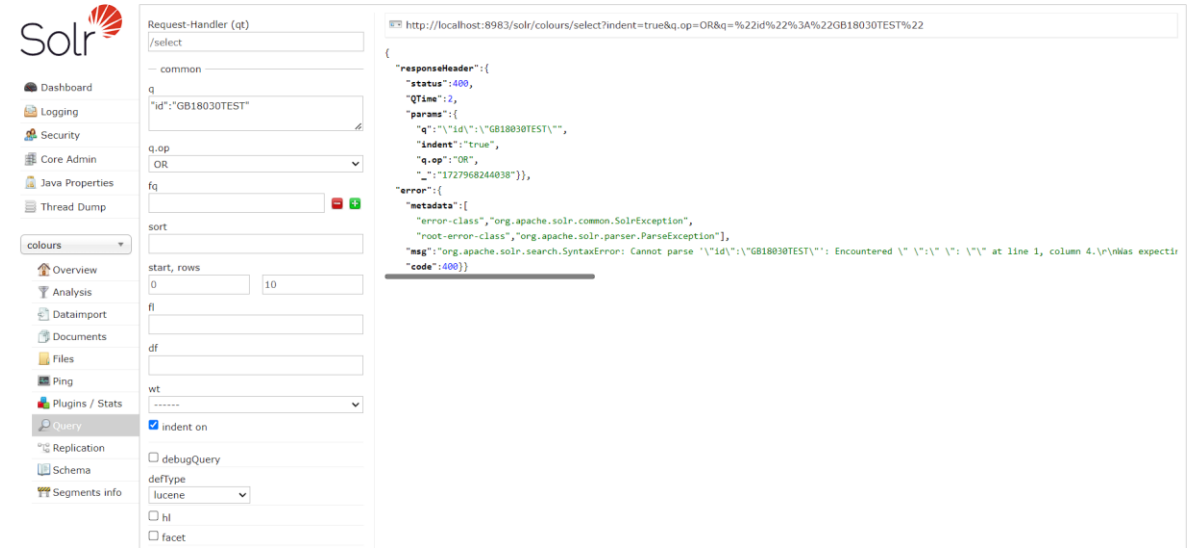
searchByColumn(v_phoneCollection, 'department', 'IT')

getDepFacet(v_nameCollection)

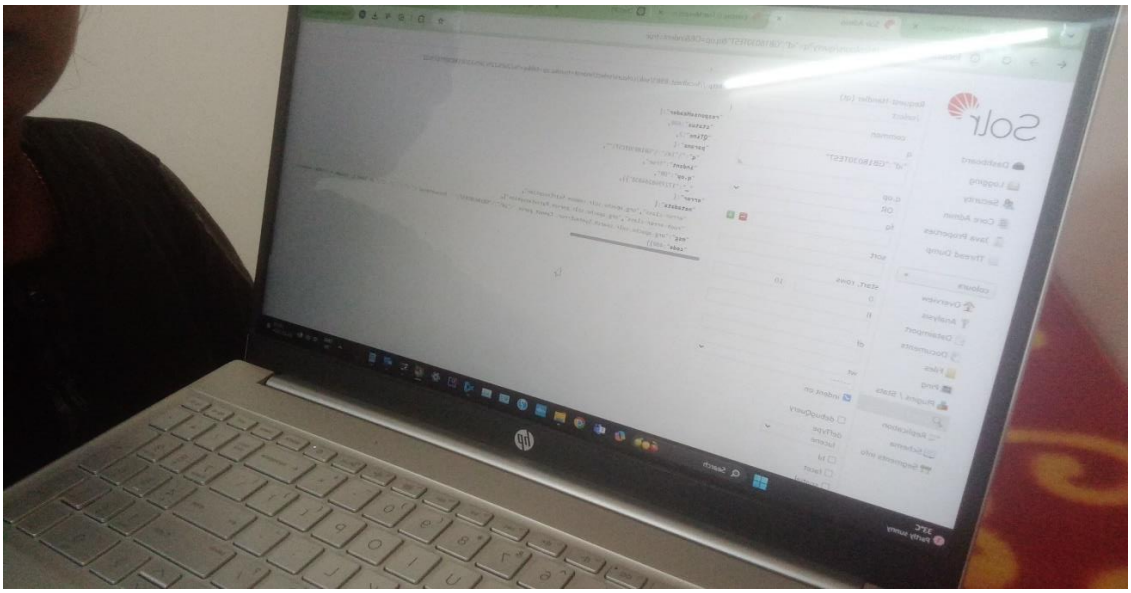
getDepFacet(v_phoneCollection)

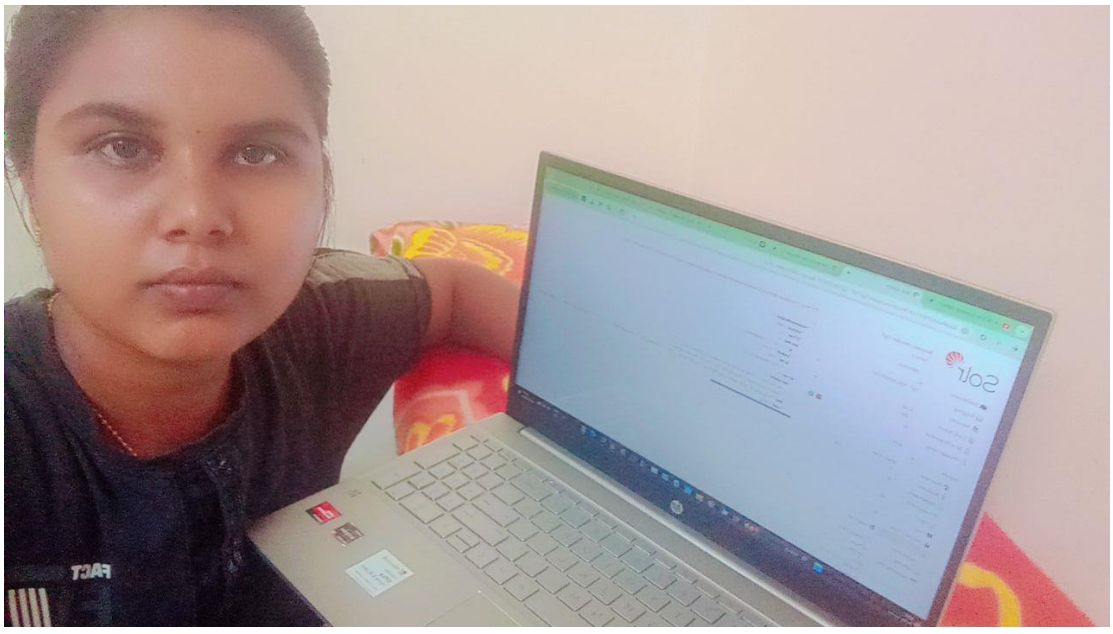
```
C:\Users\RATHISH\Downloads\solr-8.11.4\solr-8.11.4\example\exampledocs>java -Dc=colours -jar post.jar *.xml
SimplePostTool version 5.0.0
Posting files to [base] url http://localhost:8983/solr/colours/update using content-type application/xml...
POSTing file gb18030-example.xml to [base]
POSTing file hd.xml to [base]
POSTing file ipod_other.xml to [base]
POSTing file ipod_video.xml to [base]
POSTing file manufacturers.xml to [base]
POSTing file mem.xml to [base]
POSTing file money.xml to [base]
POSTing file monitor.xml to [base]
POSTing file monitor2.xml to [base]
POSTing file mp500.xml to [base]
POSTing file sd500.xml to [base]
POSTing file solr.xml to [base]
POSTing file utf8-example.xml to [base]
POSTing file vidcard.xml to [base]
14 files indexed.
COMMITting Solr index changes to http://localhost:8983/solr/colours/update...
Time spent: 0:00:02.822

C:\Users\RATHISH\Downloads\solr-8.11.4\solr-8.11.4\example\exampledocs>
```

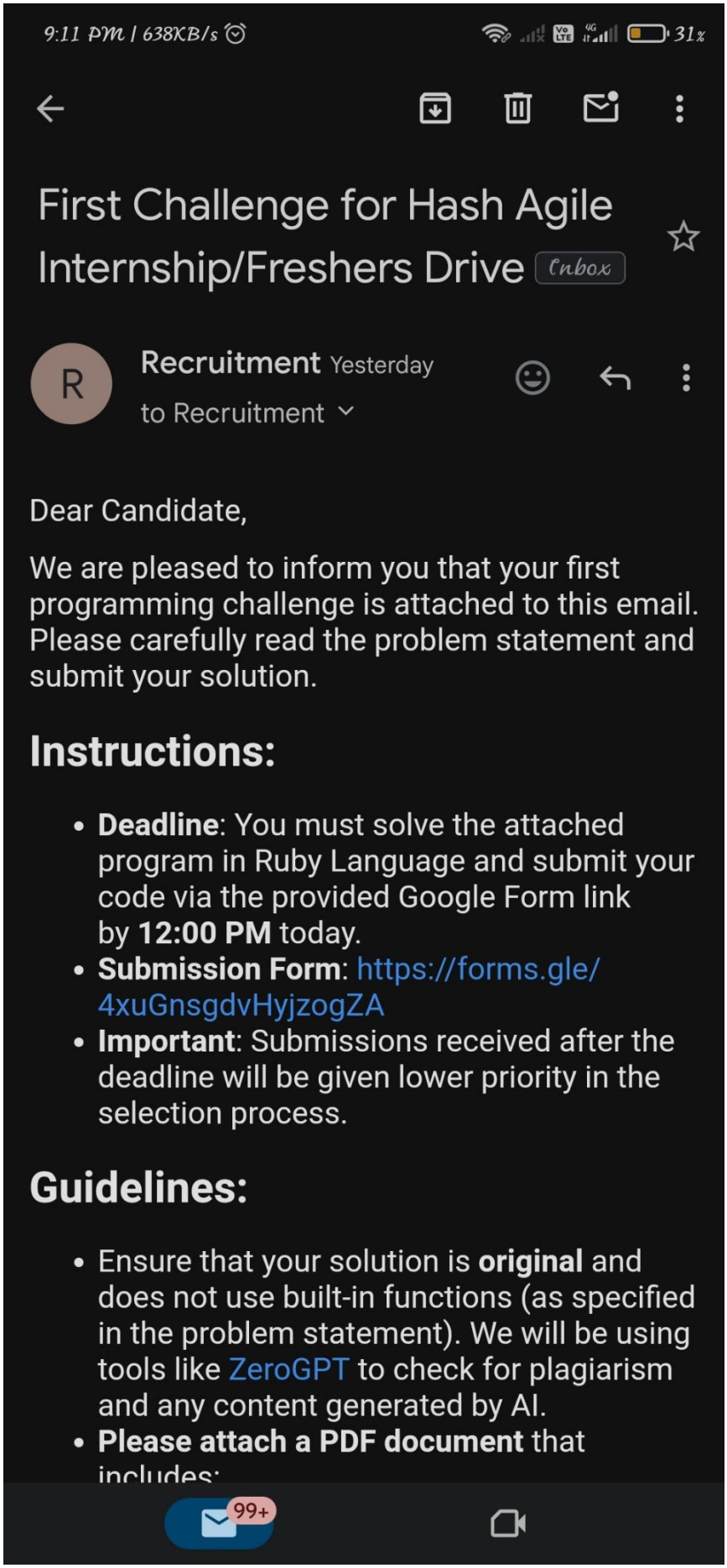

localhost:8983/solr/#/colours/query?q=*:*&q.op=OR&indent=true
Solr
Dashboard
Logging
Security
Core Admin
Java Properties
Thread Dump
colours
Overview
Analysis
Dataimport
Documents
Files
Ping
Plugins / Stats
Query
Replication
Schema
Segments info
Request-Handler (qt)
/select
common
q
:
q.op
OR
fq
sort
start, rows
0
10
fl
df
wt
------
indent on
debugQuery
defType
lucene
hl
facet
spatial
spellcheck
http://localhost:8983/solr/colours/select?indent=true&q.op=OR&q=*%3A*
{
"responseHeader":{
"status":0,
"QTime":1,
"params":{
"q":"*:*",
"indent":"true",
"q.op":"OR",
"_":"1727968244038"}},
"response":{"numFound":32,"start":0,"numFoundExact":true,"docs":[
{
"id":"GB18030TEST",
"name":["Test with some GB18030 encoded characters"],
"features":["No accents here",
"这是一个功能",
"This is a feature (translated)",
"这份文件是很有光泽",
"This document is very shiny (translated)"],
"price":[0.0],
"inStock":[true],
"_version_":1811905936720134144},
{
"id":"SP2514N",
"name":["Samsung SpinPoint P120 SP2514N - hard drive - 250 GB - ATA-133"],
"manu":["Samsung Electronics Co. Ltd."],
"manu_id_s":"samsung",
"cat":["electronics",
"hard drive"],
"features":["7200RPM, 8MB cache, IDE Ultra ATA-133",
"NoiseGuard, SilentSeek technology, Fluid Dynamic Bearing (FDB) motor"],
"price":[92.0],
"popularity":[6],
"inStock":[true],
"manufacturedate_dt":"2006-02-13T15:26:37Z",
"store":["35.0752,-97.032"],
"_version_":1811905937495031808},

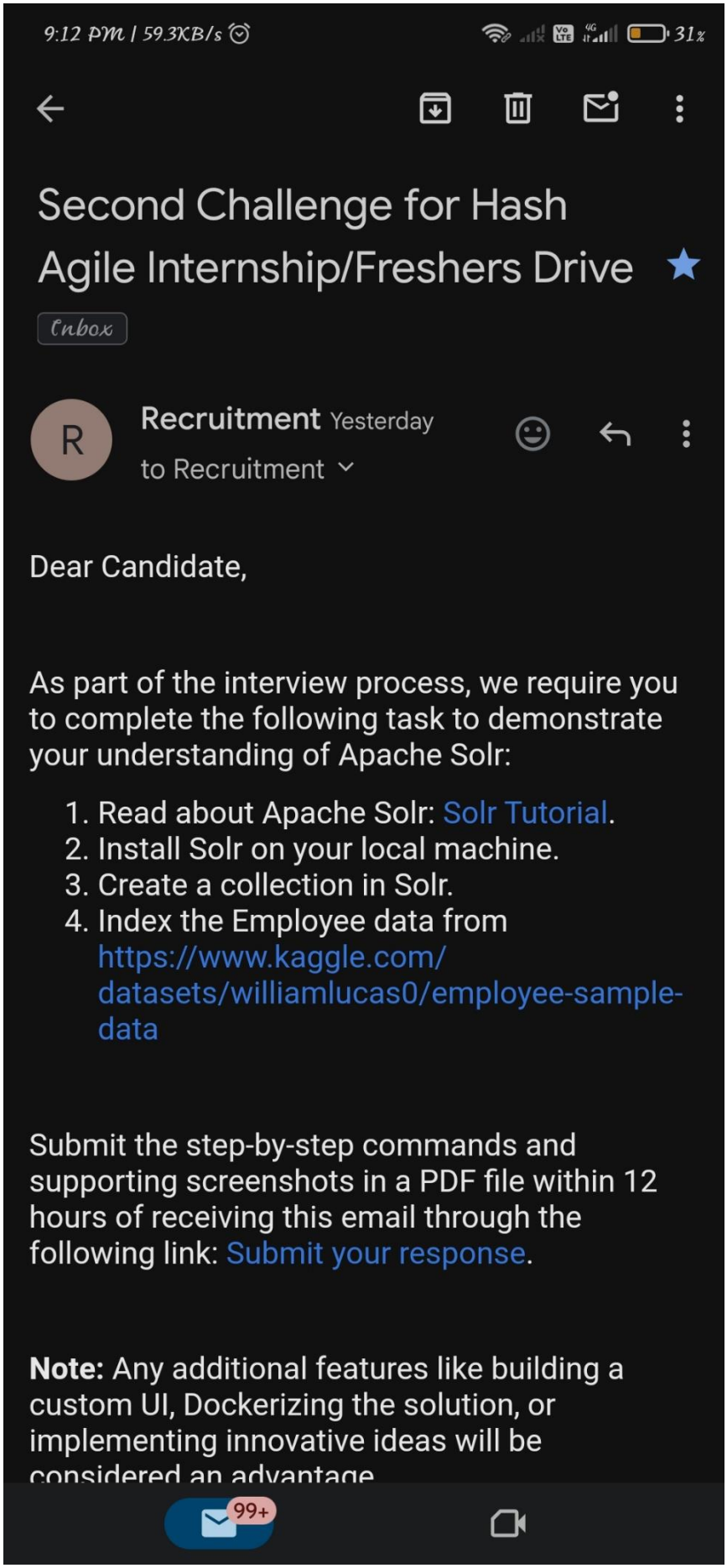

Solr
Dashboard
Logging
Security
Core Admin
Java Properties
Thread Dump
colours
Overview
Analysis
Dataimport
Documents
Files
Ping
Plugins / Stats
Query
Replication
Schema
Segments info
Request-Handler (qt)
/select
common
q
"id":"GB18030TEST"
q.op
OR
fq
sort
start, rows
0
10
fl
df
wt
------
indent on
debugQuery
defType
lucene
hl
facet
http://localhost:8983/solr/colours/select?indent=true&q.op=OR&q=%22id%22%3A%22GB18030TEST%22
{
"responseHeader":{
"status":400,
"QTime":2,
"params":{
"q":"\"id\":\"GB18030TEST\"",
"indent":"true",
"q.op":"OR",
"_":"1727968244038"}},
"error":{
"metadata":[
"error-class","org.apache.solr.common.SolrException",
"root-error-class","org.apache.solr.parser.ParseException"],
"msg":"org.apache.solr.search.SyntaxError: Cannot parse '\"id\":\"GB18030TEST\"': Encountered \" \":\" \": \"\" at line 1, column 4.\r\nWas expecti
"code":400}}

Selfie pics:






9:11 PM | 638KB/s
31%
First Challenge for Hash Agile Internship/Freshers Drive
Inbox
R
Recruitment Yesterday
to Recruitment
Dear Candidate,
We are pleased to inform you that your first programming challenge is attached to this email. Please carefully read the problem statement and submit your solution.
Instructions:
Deadline: You must solve the attached program in Ruby Language and submit your code via the provided Google Form link by 12:00 PM today.
Submission Form: https://forms.gle/4xuGnsgdvHyjzogZA
Important: Submissions received after the deadline will be given lower priority in the selection process.
Guidelines:
Ensure that your solution is original and does not use built-in functions (as specified in the problem statement). We will be using tools like ZeroGPT to check for plagiarism and any content generated by AI.
Please attach a PDF document that
99+


9:12 PM | 59.3KB/s
31%
Second Challenge for Hash Agile Internship/Freshers Drive
Inbox
R
Recruitment Yesterday
to Recruitment
Dear Candidate,
As part of the interview process, we require you to complete the following task to demonstrate your understanding of Apache Solr:
1. Read about Apache Solr: Solr Tutorial.
2. Install Solr on your local machine.
3. Create a collection in Solr.
4. Index the Employee data from https://www.kaggle.com/datasets/williamlucas0/employee-sample-data
Submit the step-by-step commands and supporting screenshots in a PDF file within 12 hours of receiving this email through the following link: Submit your response.
Note: Any additional features like building a custom UI, Dockerizing the solution, or implementing innovative ideas will be considered an advantage
99+

Repo Link:

https://github.com/Navitha05/solr_navitha05