MOHAMED DANISH M A

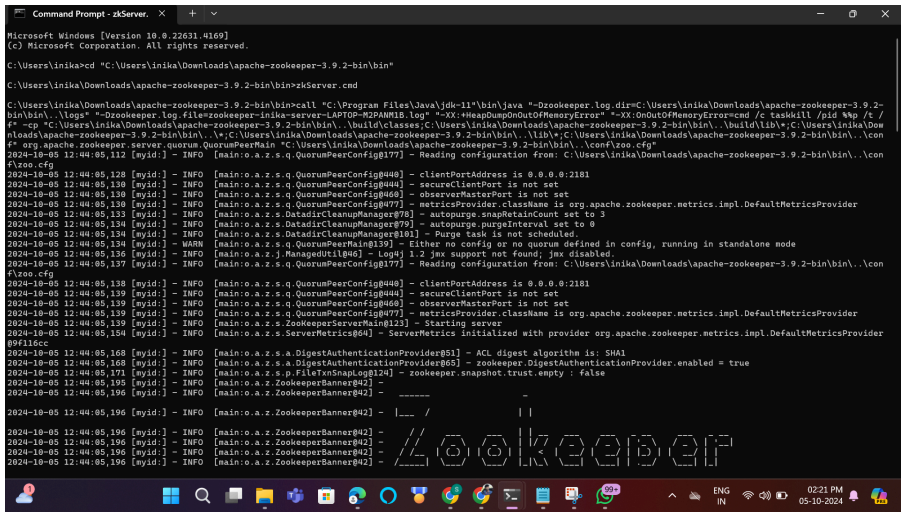
Git link : https://github.com/ThedanHisway/hashagile-asignment

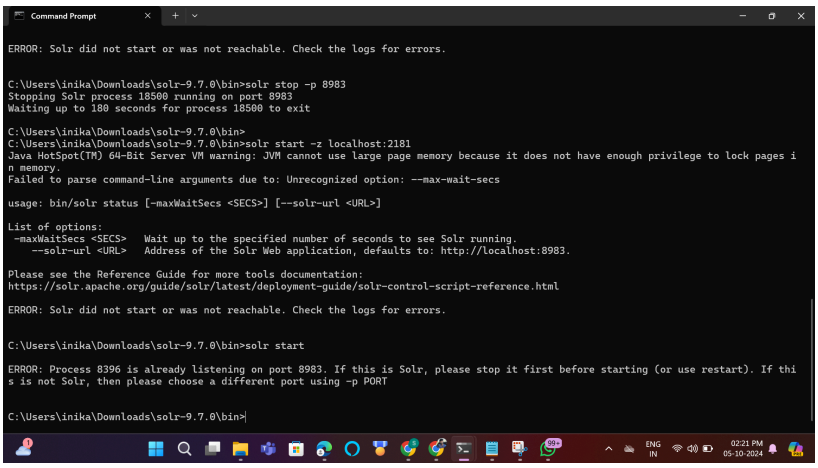
- First to run solr as standalone,
  Installed zookeeper and statered
- Started solr
- Exposed port using ngrok as i am using colab for further tasks and have to expose it globally

zkServer.cmd

solr start

ngrok http 8983





Command Prompt - ngrok ht

```
ngrok                                                                    (Ctrl+C to quit)

Share what you're building with ngrok https://ngrok.com/share-your-ngrok-story

Session Status                online
Account                       Mohamed Danish M A (Plan: Free)
Update                        update available (version 3.16.1, Ctrl-U to update)
Version                       3.16.0
Region                        India (in)
Latency                       1958ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://f10f-223-181-208-107.ngrok-free.app -> http://localhost:8983

Connections                   ttl     opn     rt1     rt5     p50     p90
                              154     0       0.00    0.00    1.63    90.39

HTTP Requests
-------------

14:15:39.895 IST GET  /solr/admin/collections   200 OK
14:15:39.893 IST GET  /solr/admin/cores         200 OK
14:15:38.283 IST GET  /solr/admin/info/system   200 OK
14:15:34.890 IST GET  /solr/                    200 OK
13:46:44.338 IST GET  /solr/Hash_Danish/facets  404 Not Found
13:46:43.715 IST GET  /solr/Hash_Danish/select  404 Not Found
13:46:43.115 IST GET  /solr/Hash_Danish/select  404 Not Found
13:46:42.355 IST GET  /solr/Hash_Danish/select  404 Not Found
13:46:41.728 IST POST /solr/Hash_Danish/update  404 Not Found
13:46:40.256 IST POST /solr/Hash_Danish/update  404 Not Found
```

localhost:8983/solr/#/~cloud

Solr

Refresh Show all details

Hosts 1 - 1 of 1. Filter by: Host/node name Show 10 hosts per page.

Host	Node	CPU	Heap	Disk usage	Requests	Collections	Replicas
localhost Windows 11 7.8Gb Java 11 Load: -1 show details...	8983_solr Uptime: 1h 12m Java 11.0.24 Solr 9.7.0 hide details...	1%	29% Max: 512.0Mb Used: 151.9Mb	528.4Kb Total #docs: 1.3k Avg size/doc: 428.8 Hash_MohamedDanish_s1r1: 528.4Kb Available disk: 117.0Gb free: 12.5Gb used by this node: 528.4Kb	RPM: 0.01 90%:	Hash_MohamedDanish	Hash_MohamedDanish_s1r1 (1.3k docs) deleted: 0 warmupTime: 0 avg size/doc: 428.8b

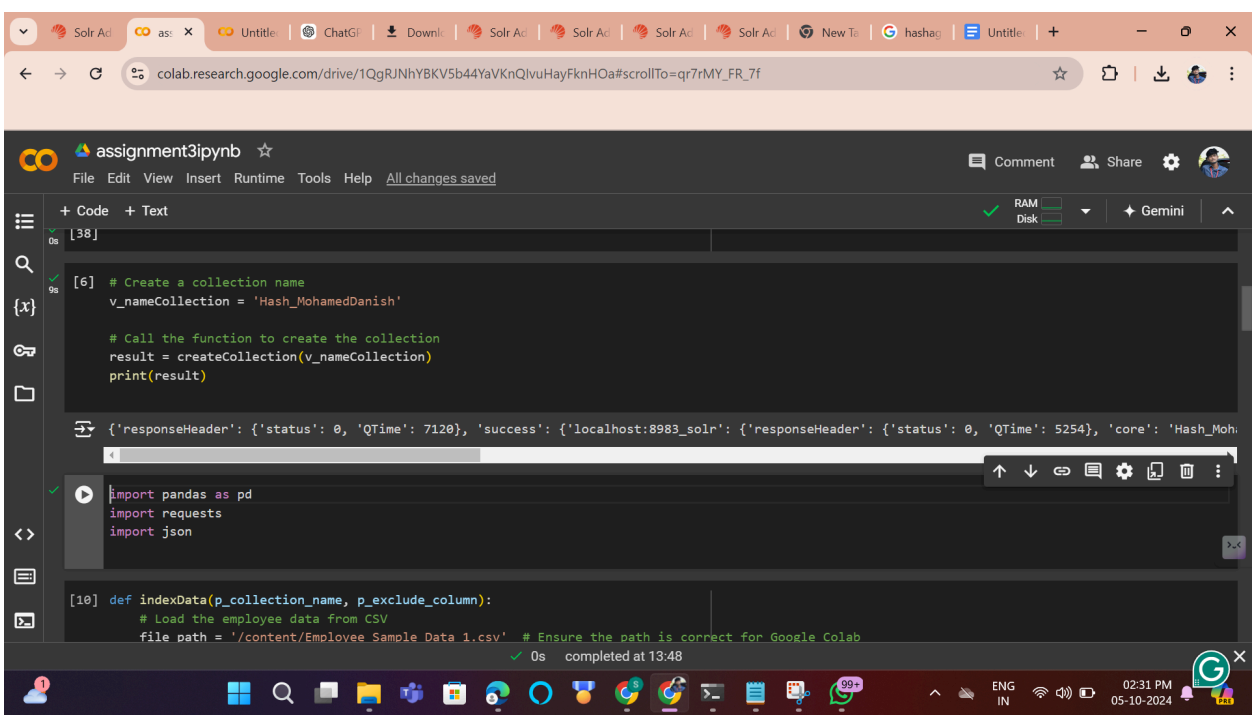
Dashboard
Logging
Security
Cloud
Nodes
Tree
ZK Status
Graph
Schema Designer
Collections
Java Properties
Thread Dump
Collection Selec...

Documentation Solr Query Syntax Community Issue Tracker Slack IRC

1] CREATING COLLECTIONS

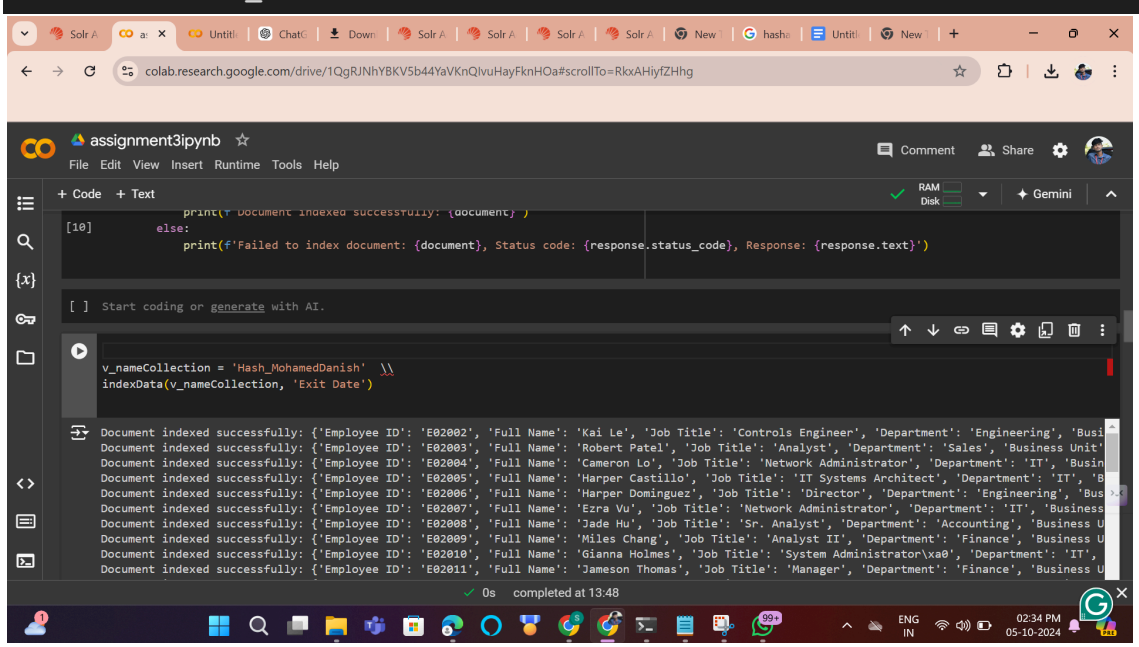
```
import requests

def createCollection(p_collection_name):
    ngrok_url =
"https://7c94-223-181-208-107.ngrok-free.app/solr/admin/collections"
    params = {
        "action": "CREATE",
        "name": p_collection_name,
        "numShards": 1,
        "replicationFactor": 1
    }
    response = requests.get(ngrok_url, params=params)
    return response.json()
```

Output



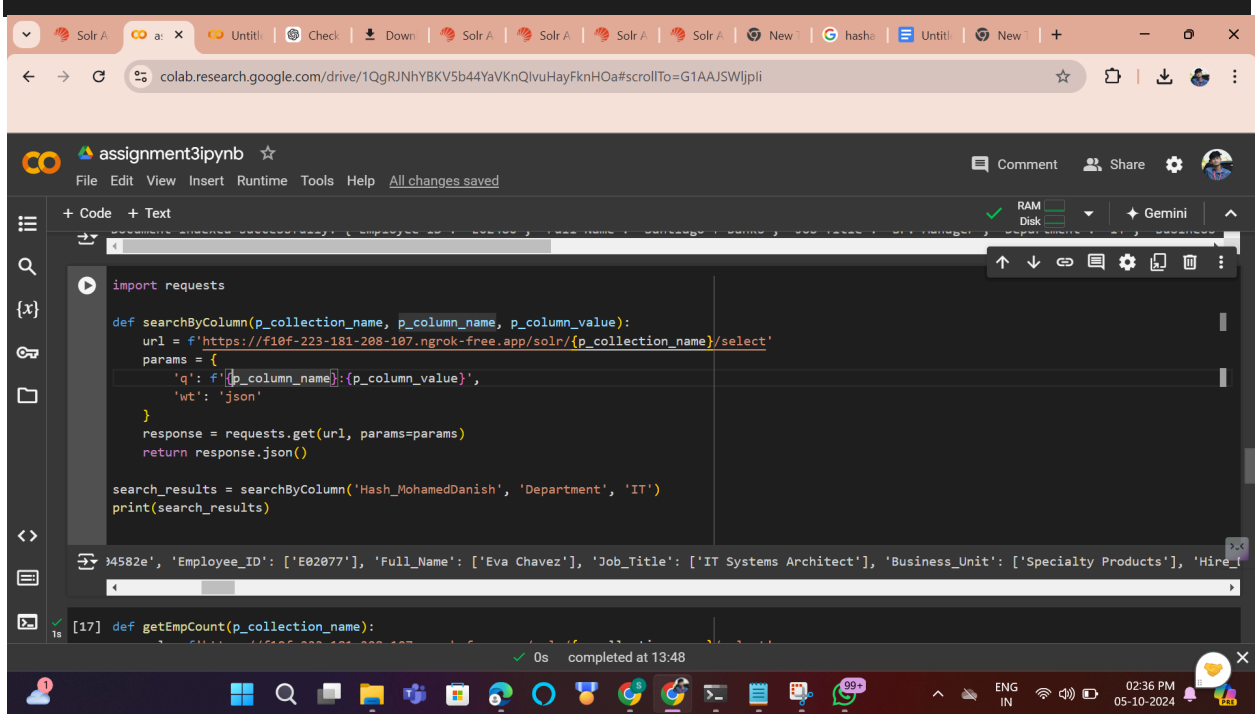
2] INDEXING

```
def indexData(p_collection_name, p_exclude_column):

    file_path = '/content/Employee Sample Data 1.csv'
    try:
        employee_data = pd.read_csv(file_path, encoding='utf-8')
    except UnicodeDecodeError:
        employee_data = pd.read_csv(file_path, encoding='ISO-8859-1')  #\
    if p_exclude_column in employee_data.columns:
        employee_data = employee_data.drop(columns=[p_exclude_column])

    for index, row in employee_data.iterrows():
        document = row.to_dict()
        response =
requests.post(f'https://7c94-223-181-208-107.ngrok-free.app/solr/{p_collection_nam
e}/update/json/docs',
                                  data=json.dumps(document),
                                  headers={'Content-Type': 'application/json'})
        if response.status_code == 200:
            print(f'Document indexed successfully: {document}')
        else:
            print(f'Failed to index document: {document}, Status code:
{response.status_code}, Response: {response.text}')
```



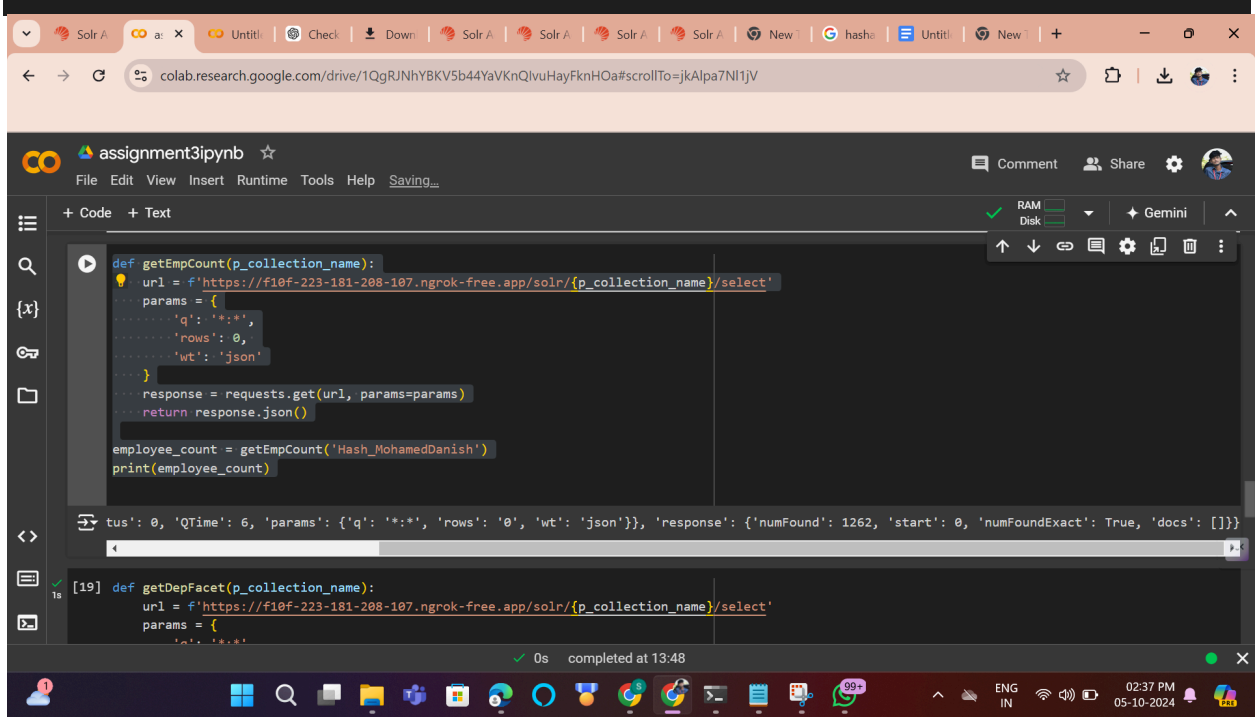
3] SEARCH BY COLUMN

```
import requests

def searchByColumn(p_collection_name, p_column_name, p_column_value):
    url =
f'https://f10f-223-181-208-107.ngrok-free.app/solr/{p_collection_name}/sel
ect'
    params = {
        'q': f'{p_column_name}:{p_column_value}',
        'wt': 'json'
    }
    response = requests.get(url, params=params)
    return response.json()

search_results = searchByColumn('Hash_MohamedDanish', 'Department', 'IT')
print(search_results)
```



4] GET EMP COUNT

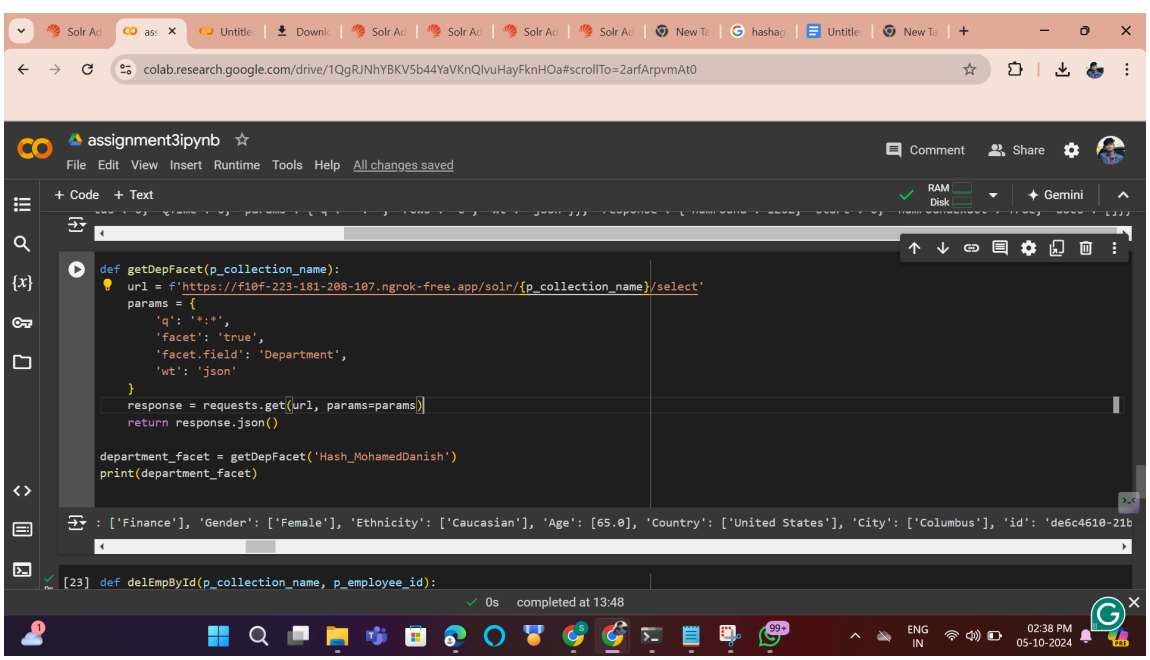
```
def getEmpCount(p_collection_name):
    url =
f'https://f10f-223-181-208-107.ngrok-free.app/solr/{p_collection_name}/sel
ect'
    params = {
        'q': '*:*',
        'rows': 0,
        'wt': 'json'
    }
    response = requests.get(url, params=params)
    return response.json()

employee_count = getEmpCount('Hash_MohamedDanish')
print(employee_count)
```



5] GET DEPT FACET

```
def getDepFacet(p_collection_name):
    url =
f'https://f10f-223-181-208-107.ngrok-free.app/solr/{p_collection_name}/sel
ect'
    params = {
        'q': '*:*',
        'facet': 'true',
        'facet.field': 'Department',
        'wt': 'json'
    }
    response = requests.get(url, params=params)
    return response.json()

department_facet = getDepFacet('Hash_MohamedDanish')
print(department_facet)
```

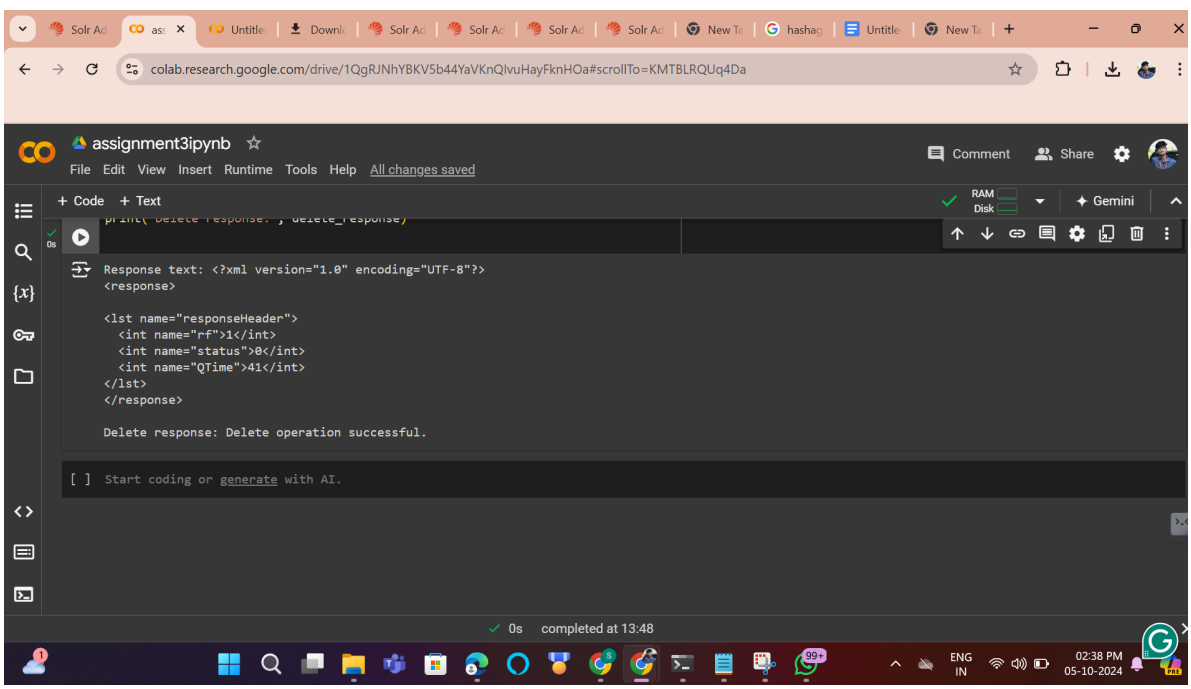
Output



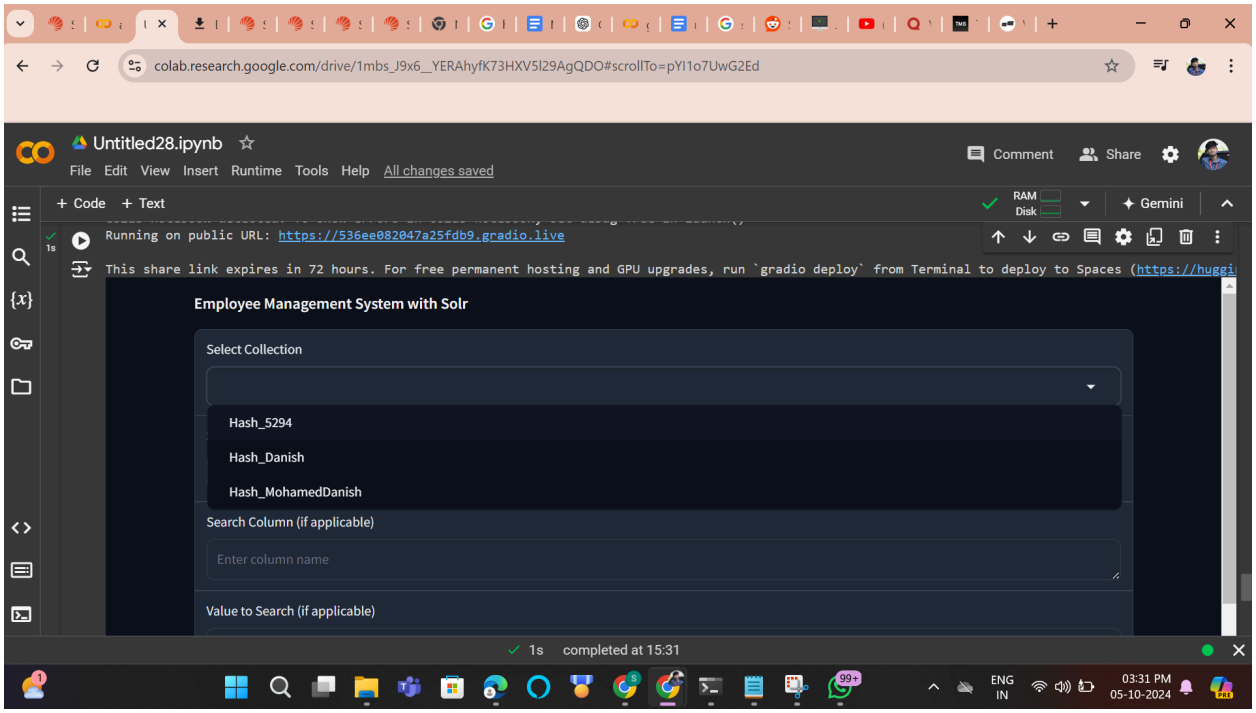
6] DELETE EMPLOYEE

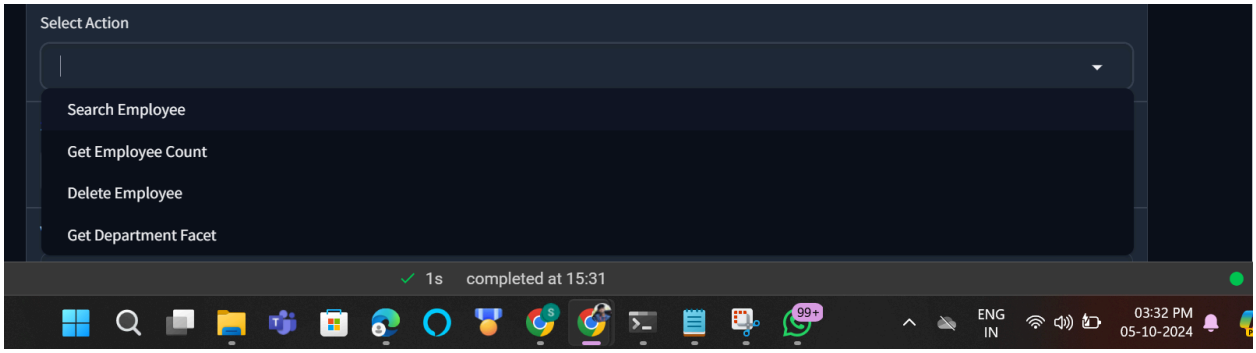
```
def delEmpById(p_collection_name, p_employee_id):
    url =
f'https://f10f-223-181-208-107.ngrok-free.app/solr/{p_collection_name}/upd
ate?commit=true'
    data = f'<delete><id>{p_employee_id}</id></delete>'
    headers = {'Content-Type': 'text/xml'}
    response = requests.post(url, data=data, headers=headers)

    print("Response text:", response.text)
    if response.status_code == 200:
        if '<int name="status">0</int>' in response.text:
            return "Delete operation successful."
        else:
            return "Delete operation failed."
    else:
        return f"Error: {response.status_code} - {response.text}"
delete_response = delEmpById('Hash_MohamedDanish', 'E02002')
print("Delete response:", delete_response)
```

Output



8] CREATED CUSTOM UI using gradio





Previously created gradio and a dashboard with aggregations like average salary, average age, max country, no. employees each dept.
Data fetched using requests from solr

Code

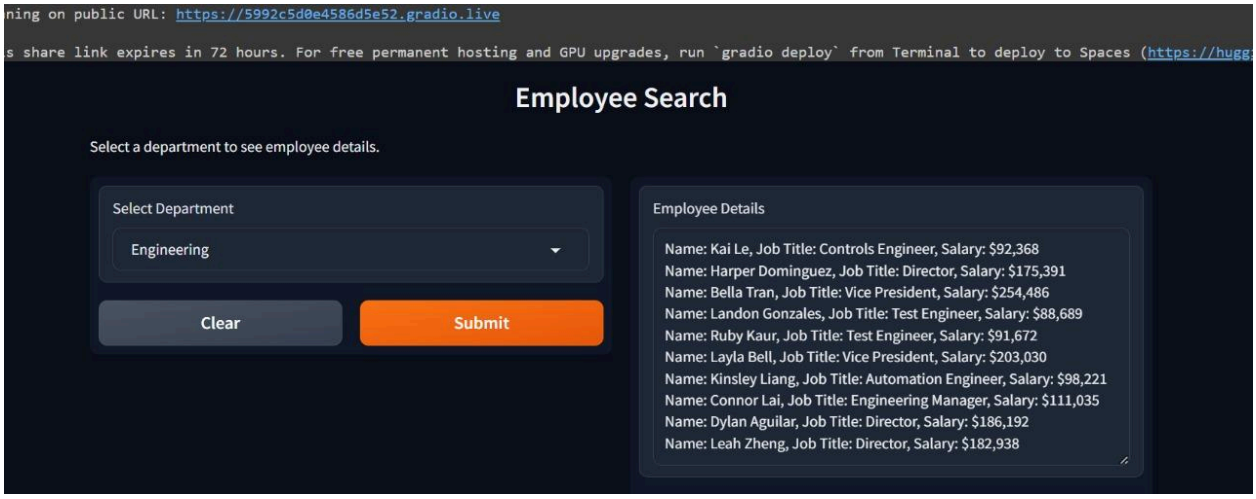
```
import gradio as gr
import requests
import numpy as np
```

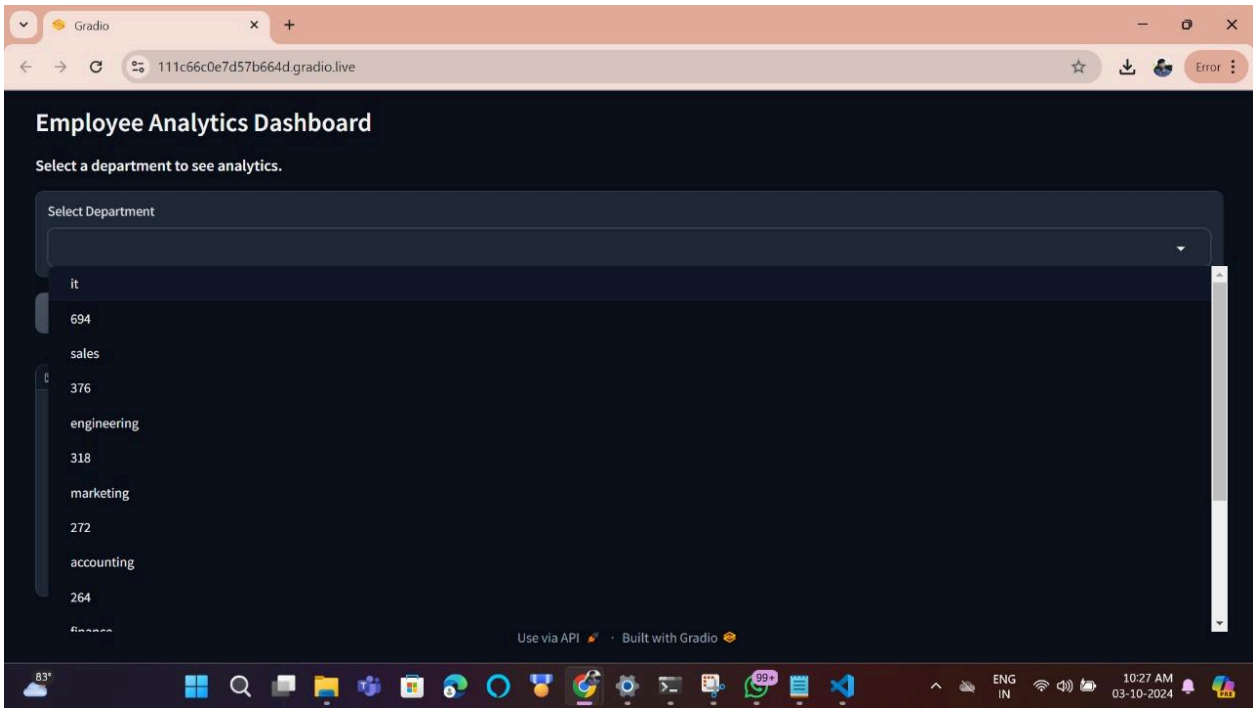
```
import matplotlib.pyplot as plt
import pandas as pd

def get_departments():
    ngrok_url = 'https://ac9d-106-222-122-100.ngrok-free.app'
    query_url =
f"{ngrok_url}/solr/employeess/terms?terms.fl=Department&terms.limit=-1&wt=
json"

    try:
        response = requests.get(query_url)
        response.raise_for_status()
        data = response.json()
        departments = data["terms"]["Department"]
        return departments if departments else []
    except Exception as e:
        print(f"Error fetching departments: {str(e)}")
        return []

def search_employees(department):
    ngrok_url = 'https://ac9d-106-222-122-100.ngrok-free.app'
    query_url =
f"{ngrok_url}/solr/employeess/select?q=Department:\"{department}\"&wt=json
&rows=100"

    try:
        response = requests.get(query_url)
        response.raise_for_status()

        data = response.json()

        if data["response"]["numFound"] == 0:
            return "No employees found for this department.", None

        employee_data = []
        for doc in data["response"]["docs"]:
            employee_data.append({
                "Job Title": doc['Job_Title'][0],
                "Salary": float(doc['Annual_Salary'][0].replace('$',
'').replace(',', '').strip()),
```

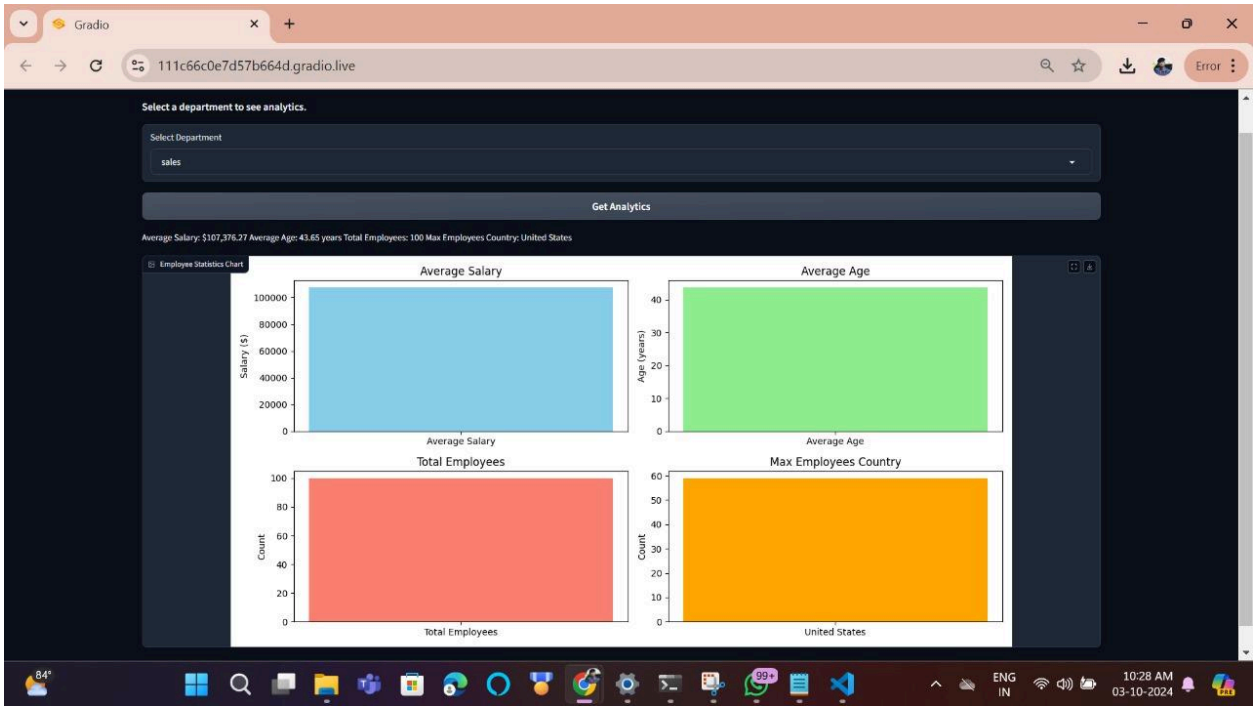
```
                "Age": doc['Age'][0],
                "Country": doc['Country'][0]
            })

        df = pd.DataFrame(employee_data)

        avg_salary = df["Salary"].mean()
        avg_age = df["Age"].mean()
        total_employees = len(df)
        max_country = df["Country"].value_counts().idxmax()

        plt.figure(figsize=(12, 6))

        plt.subplot(2, 2, 1)
        plt.bar(['Average Salary'], [avg_salary], color='skyblue')
        plt.ylabel('Salary ($)')
        plt.title('Average Salary')

        plt.subplot(2, 2, 2)
        plt.bar(['Average Age'], [avg_age], color='lightgreen')
        plt.ylabel('Age (years)')
        plt.title('Average Age')

        plt.subplot(2, 2, 3)
        plt.bar(['Total Employees'], [total_employees], color='salmon')
        plt.ylabel('Count')
        plt.title('Total Employees')

        plt.subplot(2, 2, 4)
        plt.bar([max_country],
[df["Country"].value_counts()[max_country]], color='orange')
        plt.ylabel('Count')
        plt.title('Max Employees Country')

        plt.tight_layout()
        plt.savefig("employee_statistics.png")
        plt.close()

        return (f"Average Salary: ${avg_salary:,.2f}\n"
                f"Average Age: {avg_age:.2f} years\n"
```

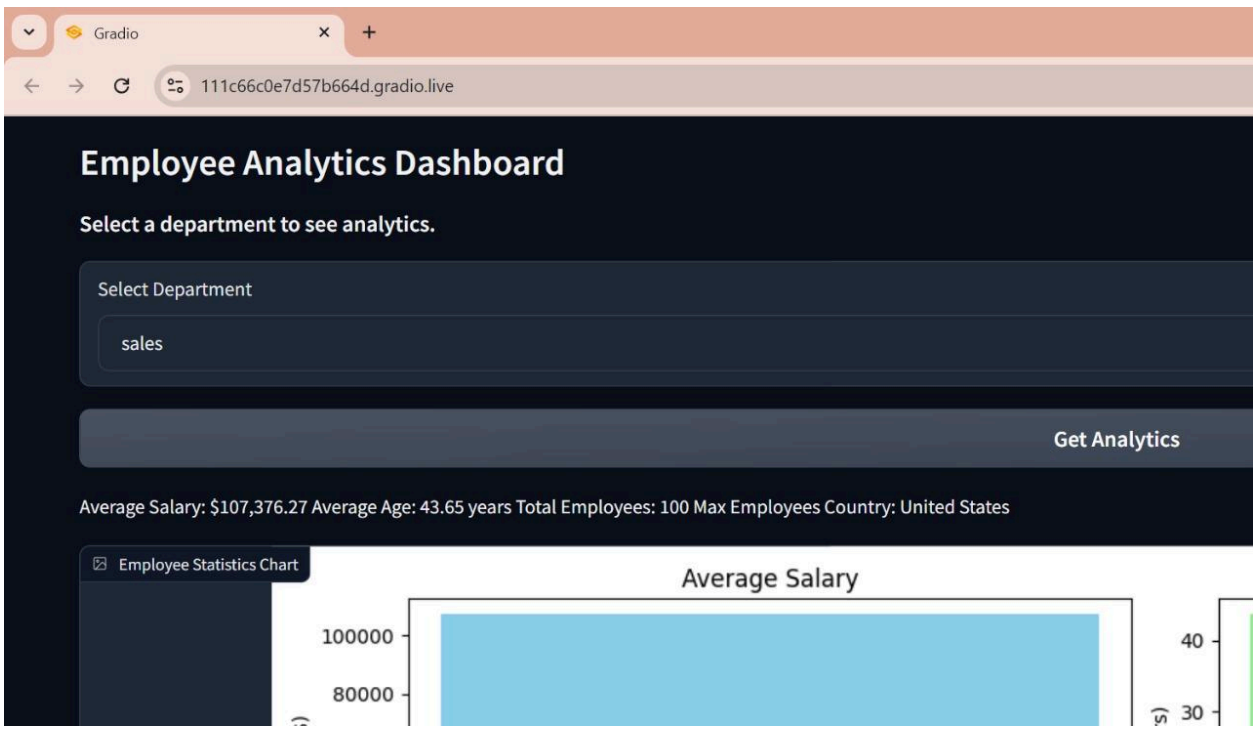
```
                f"Total Employees: {total_employees}\n"
                f"Max Employees Country: {max_country}"),
"employee_statistics.png"

    except Exception as e:
        return f"An error occurred: {str(e)}", None

with gr.Blocks() as interface:
    gr.Markdown("# Employee Analytics Dashboard")
    gr.Markdown("### Select a department to see analytics.")

    department_dropdown = gr.Dropdown(
        label="Select Department",
        choices=get_departments(),
        value=None
    )

    submit_btn = gr.Button("Get Analytics")

    output_text = gr.Markdown(label="Analytics Results",
elem_id="analytics-output")
    stats_chart = gr.Image(label="Employee Statistics Chart")

    submit_btn.click(fn=search_employees, inputs=department_dropdown,
outputs=[output_text, stats_chart])

interface.launch(share=True)
```

OUTPUT


ning on public URL: https://5992c5d0e4586d5e52.gradio.live
s share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from Terminal to deploy to Spaces (https://hugg
Employee Search
Select a department to see employee details.
Select Department
Engineering
Clear
Submit
Employee Details
Name: Kai Le, Job Title: Controls Engineer, Salary: $92,368
Name: Harper Dominguez, Job Title: Director, Salary: $175,391
Name: Bella Tran, Job Title: Vice President, Salary: $254,486
Name: Landon Gonzales, Job Title: Test Engineer, Salary: $88,689
Name: Ruby Kaur, Job Title: Test Engineer, Salary: $91,672
Name: Layla Bell, Job Title: Vice President, Salary: $203,030
Name: Kinsley Liang, Job Title: Automation Engineer, Salary: $98,221
Name: Connor Lai, Job Title: Engineering Manager, Salary: $111,035
Name: Dylan Aguilar, Job Title: Director, Salary: $186,192
Name: Leah Zheng, Job Title: Director, Salary: $182,938


Gradio
111c66c0e7d57b664d.gradio.live
Error
Employee Analytics Dashboard
Select a department to see analytics.
Select Department
it
694
sales
376
engineering
318
marketing
272
accounting
264
Use via API
Built with Gradio
83°
ENG
IN
10:27 AM
03-10-2024





Dockerizing My Gradio Project

To dockerize my Gradio application, I created a Dockerfile and built the Docker image

Step 1: Created a Dockerfile

- I created a new file named `Dockerfile`

Added the Following Code to the Dockerfile:

```
FROM python:3.9-slim

WORKDIR /app

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY employee_dashboard.py .
EXPOSE 7860
CMD ["python", "employee_dashboard.py"]
```

Step 2: Created a requirements.txt File

- I created a `requirements.txt` file in the same directory.

Added the Required Libraries:

```
gradio
requests
numpy
matplotlib
pandas
```

Step 3: Built the Docker Image

- I opened a terminal in the directory where my Dockerfile and Python script were located.

Ran the Following Command to Build the Docker Image:

```
docker build -t employee_dashboard .
```

Step 4: Ran the Docker Container

Once the image was built, I ran the container with this command:

```
docker run -p 7860:7860 employee_dashboard
```