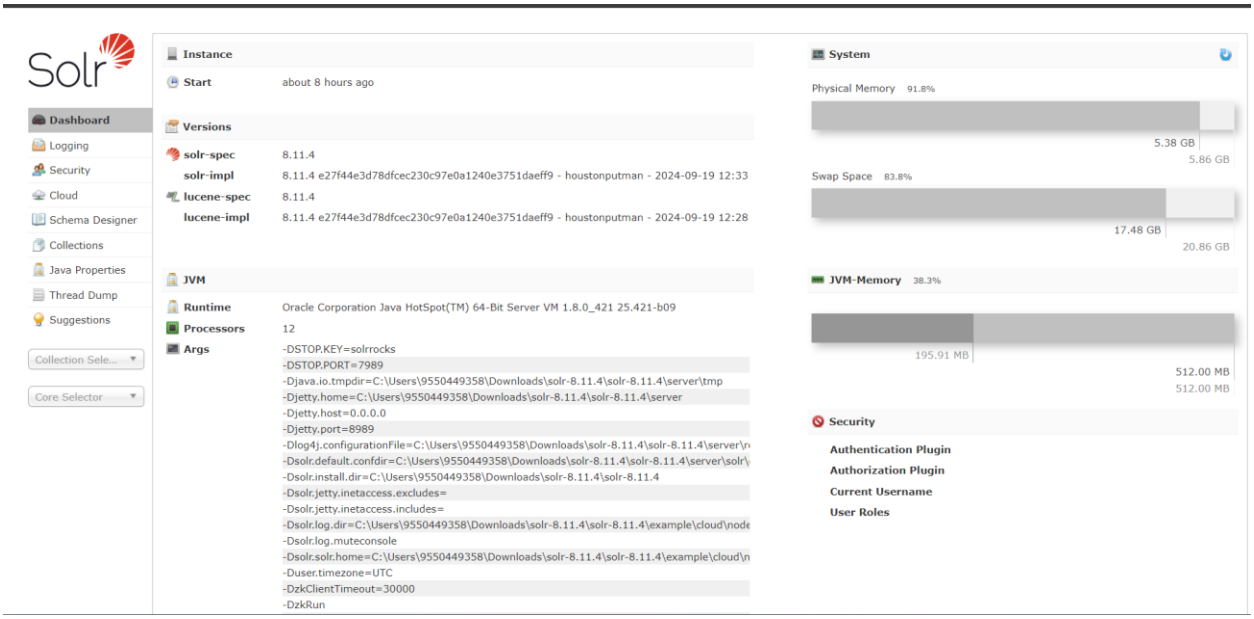
Name :Dharmendra Poondla

Selfie pic

```
C:\Users\9550449358\Downloads\solr-9.7.0\solr-9.7.0>Exception in thread "main" java.lang.UnsupportedClassVersionError: o
rg/eclipse/jetty/start/Main has been compiled by a more recent version of the Java Runtime (class file version 55.0), th
is version of the Java Runtime only recognizes class file versions up to 52.0
        at java.lang.ClassLoader.defineClass1(Native Method)
        at java.lang.ClassLoader.defineClass(Unknown Source)
        at java.security.SecureClassLoader.defineClass(Unknown Source)
        at java.net.URLClassLoader.defineClass(Unknown Source)
        at java.net.URLClassLoader.access$100(Unknown Source)
        at java.net.URLClassLoader$1.run(Unknown Source)
        at java.net.URLClassLoader$1.run(Unknown Source)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.net.URLClassLoader.findClass(Unknown Source)
        at java.lang.ClassLoader.loadClass(Unknown Source)
        at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
        at java.lang.ClassLoader.loadClass(Unknown Source)
        at sun.launcher.LauncherHelper.checkAndLoadMain(Unknown Source)
bin/solr -e cloud
```



```
def create_solr_collection(collection_name, num_shards=1, replication_factor=1):

    solr_url = f"http://localhost:8989/solr/admin/collections"


    params = {
        "action": "CREATE",
        "name": collection_name,
        "numShards": num_shards,
        "replicationFactor": replication_factor
    }


    try:
        response = requests.get(solr_url, params=params)


        if response.status_code == 200:
            print(f"Collection '{collection_name}' created successfully.")
        else:
            print(f"Failed to create collection '{collection_name}'. Response: {response.text}")

    except Exception as e:
        print(f"Error while creating collection: {e}")
'
```

Output

```
Collection 'my_new_collection' created successfully.
```

```
def read_excel_data(file_path, exclude_columns=None):
    """
    Reads data from an Excel file and returns it as a list of dictionaries.

    :param file_path: Path to the Excel file.
    :param exclude_columns: A list of columns to exclude from indexing.
    :return: List of dictionaries representing the data.
    """

    df = pd.read_csv(file_path,encoding='ISO-8859-1')


    if exclude_columns:
        df = df.drop(columns=exclude_columns, errors='ignore')


    data = df.to_dict(orient='records')
    return data

def index_data_to_solr(data, collection_name):
    """
    Index data to Solr.

    :param data: List of dictionaries containing the data to index.
    :param collection_name: The name of the Solr collection.
    """

    solr_url = f"http://localhost:8989/solr/{collection_name}/update?commit=true"


    headers = {
        'Content-Type': 'application/json'
    }
```

```
    try:
        response = requests.post(solr_url, headers=headers, data=json.dumps(data))


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

    except Exception as e:
        print(f"Error while indexing data: {e}")
```

Output

```
========== RESTART: C:/Users/9550449358/AppData/Local/Programs/Python/Python311/v2.py ==========
Data successfully indexed into collection 'my_new_collection'
```

```python
class SolrSearchRequest:
    def __init__(self, collection_name, column_name, column_value):
        self.collection_name = collection_name
        self.column_name = column_name
        self.column_value = column_value

def searchByColumn(solr_url, search_request: SolrSearchRequest):

    query = f'{search_request.column_name}:{search_request.column_value}'


    search_url = f"{solr_url}/{search_request.collection_name}/select"


    params = {
        'q': query,
        'wt': 'json',
        'indent': 'true'
    }

    try:

        response = requests.get(search_url, params=params)


        response.raise_for_status()


        data = response.json()


        if data['response']['numFound'] > 0:
            return data['response']['docs']
        else:
            return []

    except requests.exceptions.RequestException as e:
        print(f"Error occurred while searching: {e}")
        return None
```

```python
class SolrSearchRequest:
    def __init__(self, collection_name, column_name, column_value):
        self.collection_name = collection_name
        self.column_name = column_name
        self.column_value = column_value

def searchByColumn(solr_url, search_request: SolrSearchRequest):

    query = f'{search_request.column_name}:{search_request.column_value}'


    search_url = f"{solr_url}/{search_request.collection_name}/select"


    params = {
        'q': query,
        'wt': 'json',
        'indent': 'true'
    }

    try:

        response = requests.get(search_url, params=params)


        response.raise_for_status()


        data = response.json()


        if data['response']['numFound'] > 0:
            return data['response']['docs']
        else:
            return []

    except requests.exceptions.RequestException as e:
        print(f"Error occurred while searching: {e}")
        return None
```

Output

```
========== RESTART: C:/Users/9550449358/AppData/Local/Programs/Python/Python311/v3.py ==========
Search Results: [{'Gender': ['Male'], 'Ethnicity': ['Asian'], 'Age': [47.0], 'Country': ['United States'], 'City': ['Columbus'], 'id': '7175ca46-9829-4a68-a2e2-c8b62ec6516b', 'Employee_ID': ['E02002'], 'Full_Name': ['Kai Le'], 'Job_Title': ['Controls Engineer'], 'Business_Unit': ['Manufacturing'], 'Hire_Date': ['2/5/2022'], 'Annual_Salary': ['$92,368 '], 'Bonus__': ['0%'], 'Exit_Date': ['NaN'], '_version_': 1812465758751424512}, {'Gender': ['Male'], 'Ethnicity': ['Asian'], 'Age': [47.0], 'Country': ['NaN'], 'City': ['Columbus'], 'id': '07815269-2
2a1-494a-9bb4-46cf75e85755', 'Employee_ID': ['E02002'], 'Full_Name': ['Kai Le'], 'Job_Title': ['NaN'], 'Business_Unit': ['Manufacturing'], 'Hire_Date': ['2/5/2022'], 'Annual_Salary': ['$92,368 '], 'Bonus__': ['0%'], 'Exit_Date': ['NaN'], '_version_': 1812465760438583296}, {'Gender': ['Male'], 'Ethnicity': ['Asian'], 'Age': [47.0], 'Country': ['NaN'], 'City': ['NaN'], 'id': 'ce3c46e1-0615-4ff9-93ca-b5bf4d616f00', 'Employee_ID': ['E02002'], 'Full_Name': ['Kai Le'], 'Job_Title': ['Controls Engineer'], 'Business_Unit': ['Manufacturing'], 'Hire_Date': ['2/5/2022'], 'Annual_Salary': ['$92,368 '], 'Bonus__': ['0%'], 'Exit_Date': ['NaN'], '_version_': 1812465760517226496}]
```

```python
def getEmpCount(solr_url, p_collection_name):

    count_url = f"{solr_url}/{p_collection_name}/select"

    params = {
        'q': '*:*',
        'rows': 0,
        'wt': 'json',
        'indent': 'true'
    }

    try:

        response = requests.get(count_url, params=params)


        response.raise_for_status()


        data = response.json()


        return data['response']['numFound']

    except requests.exceptions.RequestException as e:
        print(f"Error occurred while fetching employee count: {e}")
        return None
```

Output

```
========== RESTART: C:/Users/9550449358/AppData/Local/Programs/Python/Python311/v4.py ==========
Employee Count in 'my_new_collection': 1262
```

```python
def delEmpById(solr_url, p_collection_name, p_employee_id):

    delete_url = f"{solr_url}/{p_collection_name}/update?commit=true"


    payload = {
        "delete": {
            "query": f"Employee_ID:\"{p_employee_id}\""
        }
    }

    try:

        response = requests.post(delete_url, data=json.dumps(payload), headers={'Content-Type': 'application/json'})


        response.raise_for_status()

        return response.json()

    except requests.exceptions.RequestException as e:
        print(f"Error occurred while deleting employee ID {p_employee_id}: {e}")
        return None
```

Output

```
======================================================== RESTART: C:/Users/9550449358/AppData/Local/Programs/Python/Python311/v5.py ========================================
Delete operation response: {'responseHeader': {'rf': 1, 'status': 0, 'QTime': 404}}
```

```python
def getDepFacet(solr_url, p_collection_name):

    facet_url = f"{solr_url}/{p_collection_name}/select"


    params = {
        'q': '*:*',
        'facet': 'true',
        'facet.field': 'Gender',
        'facet.mincount': 1,

    }

    try:

        response = requests.get(facet_url, params=params)


        response.raise_for_status()


        facet_counts = response.json().get('facet_counts', {}).get('facet_fields', {}).get('Gender', [])


        department_counts = {}
        for i in range(0, len(facet_counts), 2):
            department_name = facet_counts[i]
            count = facet_counts[i + 1]
            department_counts[department_name] = count

        return department_counts

    except requests.exceptions.RequestException as e:
        print(f"Error occurred while retrieving department facet counts: {e}")
        return None
```

Output

```
======================================================== RESTART: C:/Users/9550449358/AppData/Local/Programs/Python/Python311/v6.py ========================================
Department counts: {'female': 629, 'male': 581, 'nan': 49}
```

a) Var v_nameCollection = ‘Hash_<Your Name>’

b) Var v_phoneCollection =’Hash_<Your Phone last four digits’

c) createCollection(v_nameCollection)

d) createCollection(v_phoneCollection)

```
v_nameCollection = "Hash_Dharmendra"


v_phoneCollection = "Hash_4358"


create_solr_collection(solr_url, v_nameCollection)
create_solr_collection(solr_url, v_phoneCollection)
```

Output]

```
Collection 'Hash_Dharmendra' already exists.
Collection 'Hash_4358' already exists.
```

e) getEmpCount(v_nameCollection)

```
data = read_csv_data(file_path, exclude_columns=["Department"])
index_data_to_solr(data, solr_url, v_nameCollection)
```

Output

```
Data successfully indexed into collection 'Hash_Dharmendra'
```

f) indexData(v_nameCollection,’Department’)

g) indexData(v_ phoneCollection, ‘Gender’)

```
data = read_csv_data(file_path, exclude_columns=["Gender"])
index_data_to_solr(data, solr_url, v_phoneCollection)
```

Output

```
Data successfully indexed into collection 'Hash_4358'
```

h) getEmpCount(v_nameCollection)

```
emp_count = get_emp_count(solr_url, v_nameCollection)
print(f"Employee count in '{v_nameCollection}' after indexing: {emp_count}")
```

Output

```
Employee count in 'Hash Dharmendra' after indexing: 12590
```

i.delEmpById (v_ nameCollection ,'E02003')

```
del_emp_by_id(solr_url, v_nameCollection, 'E02003')
```

Output

```
Employee ID E02003 deleted successfully.
```

j) getEmpCount(v_nameCollection)

```
emp_count = get_emp_count(solr_url, v_nameCollection)
print(f"Employee count in '{v_nameCollection}' after deletion: {emp_count}")
```

Output

```
Employee count in 'Hash_Dharmendra' after deletion: 15108
```

k) searchByColumn(v_nameCollection,'Department','IT')

```
it_employees = search_by_column(solr_url, v_nameCollection, 'Department', 'IT')
print(f"Employees in IT department: {it_employees}")
```

Output

l) searchByColumn(v_nameCollection,'Gender' ,'Male')

```
male_employees = search_by_column(solr_url, v_nameCollection, 'Gender', 'Male')
print(f"Male employees: {male_employees}")
```

Output

m) searchByColumn(v_ phoneCollection,'Department','IT')

```
it_employees_phone = search_by_column(solr_url, v_phoneCollection, 'Department', 'IT')
print(f"IT employees in phone collection: {it_employees_phone}")
```

n) getDepFacet(v_ nameCollection)

o) getDepFacet(v_ phoneCollection)

```
dep_facet_counts = get_dep_facet(solr_url, v_nameCollection, 'Department')
print(f"Department facet counts in name collection: {dep_facet_counts}")
```

Output

```
Department facet counts in name collection: None
```

Git hub url: https://github.com/dharmendrapoondla123/SolrCode/tree/main