# Department of Computing

**Course Code: CS-432**

**Class: BSCS-12ABC**

Lab 11: Indexing, Importing and Searching data in Apache Solr

**Date: 25th April ,2025**

**Instructor: Dr. Khurram Shahzad, Dr. Shah Khalid**

**Name: Abdullah Imran**

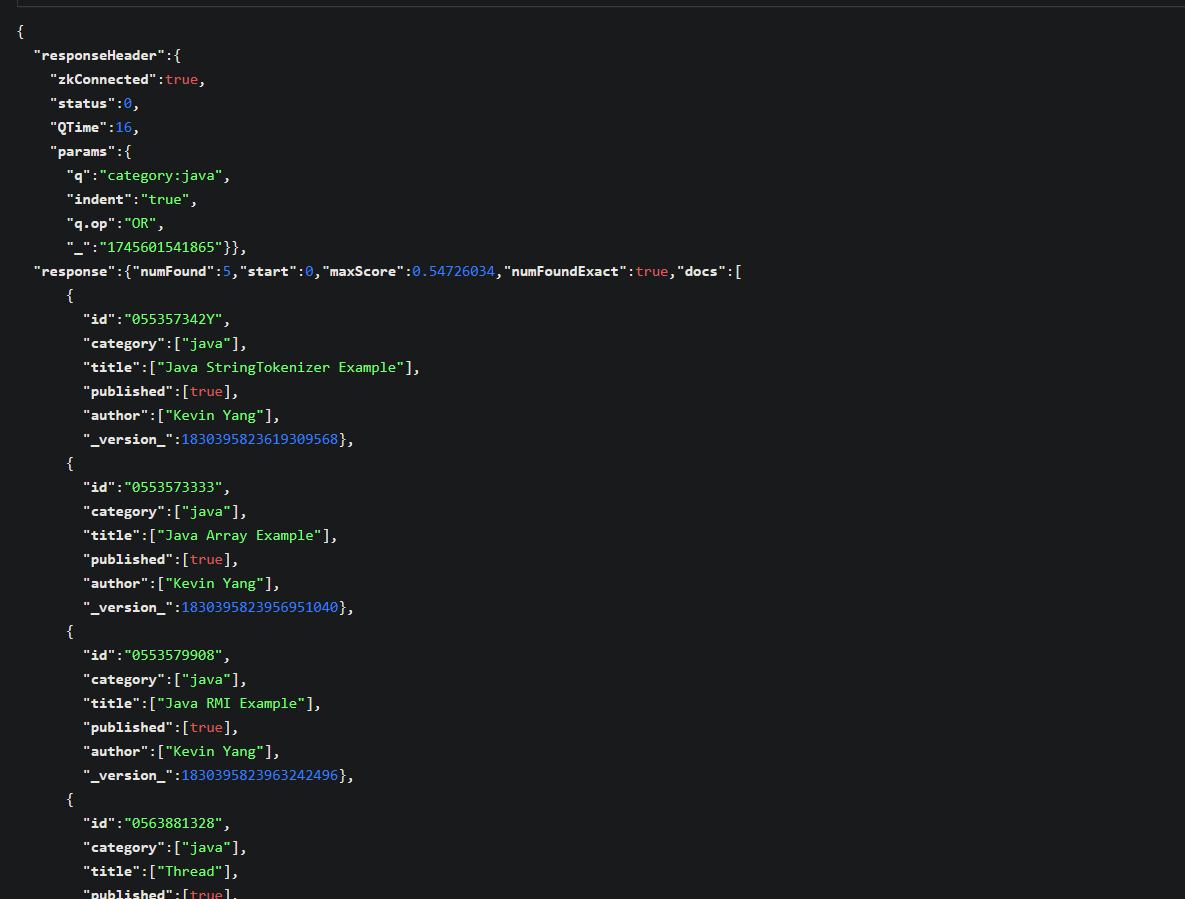
**Cms: 415833**

**Task  
  
Task 1**

You have already created a cluster of servers(shards) now take a sample data Index it, Import and perform searching on it.

Sample data is available on the following link.





Useful Links:   
<https://examples.javacodegeeks.com/apache-solr-clustering-example/>

### Task 2: Integrate Solr with a Web Interface

* Use a frontend framework (e.g., React, Flask, or Node.js) to build a search UI.
* Connect the UI with Solr backend via REST API.
* Display real-time search results with autocomplete and filters.

Flask Code

from flask import Flask, render\_template, request, jsonify

import requests

app = *Flask*(\_\_name\_\_)

SOLR\_COLLECTION = "abd"  # *replace with your collection name*

SOLR\_URL = *f*"http://localhost:8983/solr/{SOLR\_COLLECTION}/select"

@app.route('/')

*def* **index**():

    return *render\_template*('index.html')

@app.route('/search')

*def* **search**():

    query = request.args.*get*('q','\*:\*')

    category\_filter = request.args.*get*('category','')

    params = {

        'q': *f*'title:{query}\*' if query != '\*:\*' else '\*:\*',

        'wt': 'json',

        'rows': 10

    }

    if category\_filter:

        params['fq'] = *f*'category:{category\_filter}'

    solr\_response = requests.*get*(SOLR\_URL,params=params).*json*()

    results = solr\_response.*get*('response',{}).*get*('docs',[])

    formatted\_results = []

    for doc in results:

        formatted\_results.*append*({

'id':doc.*get*('id',''),

'title':doc.*get*('title',[''])[0],

'author':doc.*get*('author',[''])[0],

'category':doc.*get*('category',[''])[0],

'published':doc.*get*('published',[*False*])[0]

})

*print*(formatted\_results)

    return formatted\_results

if \_\_name\_\_ == '\_\_main\_\_':

    app.*run*(debug=*True*)

Html Code

<!DOCTYPE *html*>

<html>

<head>

    <title>Solr Search</title>

</head>

<body *style*="font-family: Arial, sans-serif; background: #f2f2f2; margin: 0; padding: 0;">

    <div *style*="width: 60%; margin: 50px auto; padding: 20px; background: white; border-radius: 8px;">

        <h1 *style*="text-align: center;">Search Solr</h1>

        <div *style*="margin-bottom: 10px;">

            <input *type*="text" *id*="search-input" *placeholder*="Search..." *autocomplete*="off"

*style*="padding: 10px; width: 60%; margin-right: 10px;" />

            <select *id*="filter" *style*="padding: 10px;">

                <option *value*="">All</option>

                <option *value*="category:news">News</option>

                <option *value*="category:tech">Tech</option>

            </select>

        </div>

        <ul *id*="suggestions" *style*="list-style: none; padding-left: 0; margin-top: 5px;"></ul>

        <div *id*="results"></div>

    </div>

    <script>

        const input = document.*getElementById*("search-input");

        const filter = document.*getElementById*("filter");

        const results = document.*getElementById*("results");

        const suggestions = document.*getElementById*("suggestions");

        input.*addEventListener*("input", () => {

*fetch*(`/autocomplete?prefix=${input.value}`)

                .*then*(res => res.*json*())

                .*then*(data => {

                    suggestions.innerHTML = "";

                    data.*forEach*(item => {

                        let li = document.*createElement*("li");

                        li.textContent = item;

                        li.onclick = () => {

                            input.value = item;

                            suggestions.innerHTML = "";

*performSearch*();

                        };

                        li.style.background = "#eee";

                        li.style.padding = "5px";

                        li.style.cursor = "pointer";

                        suggestions.*appendChild*(li);

                    });

                });

        });

        input.*addEventListener*("keypress", e => {

            if (e.*key* === "Enter") *performSearch*();

        });

        filter.*addEventListener*("change", performSearch);

        function **performSearch**() {

            const query = input.value;

            const selectedFilter = filter.value;

*fetch*(`/search?q=${query}&filter=${selectedFilter}`)

                .*then*(res => res.*json*())

                .*then*(data => {

                    results.innerHTML = "<h2>Results</h2>";

                    data.*forEach*(doc => {

                        let div = document.*createElement*("div");

                        div.innerHTML = `<pre>${JSON.*stringify*(doc, null, 2)}</pre>`;

                        div.style.marginTop = "10px";

                        div.style.padding = "10px";

                        div.style.background = "#f9f9f9";

                        div.style.borderLeft = "3px solid #007BFF";

                        results.*appendChild*(div);

                    });

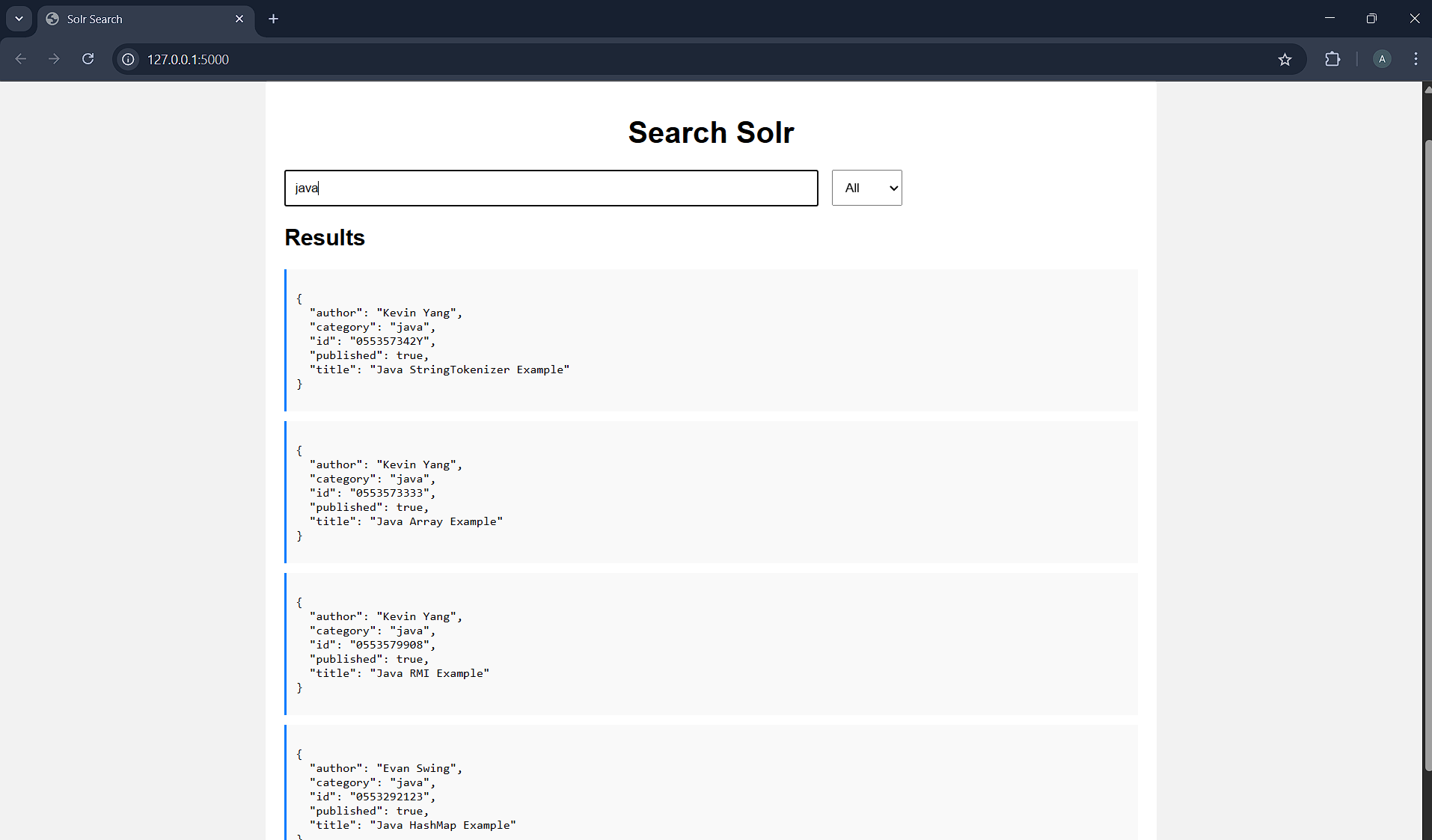
                });

        }

    </script>

</body>

</html>



**Deliverables**

Upload single word file with screenshots of your output on LMS.