Project Report

On

“**Enterprise Search Using Elastic Search**”

Submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology in Computer Science

At

**Exploration & Development Directorate**

**Oil & Natural Gas Corporation Limited, Dehradun**



By:

**TEAM MEMBERS**

**Abhishek Bhardwaj**

**Vandit jain**

**Yash Tariyal**

**Anurag Kumar**

Graphic Era Deemed to be University, Dehradun



Under the guidance of Submitted to

**Mr. M K Mathur** **MR. Prabhadeep Singh**

Head Computer Services, E&D DTE, Assistant professor

ONGC, Dehradun GEU CSE

1

# *Copy of certificate*

# 

# *Certificate of Completion*

This is to certify that the project report entitled “**Enterprise Search Using Elastic**

**Search**” carried out by **Vandit jain,Abhishek Bhardwaj,Yash tariyal,Anurag kumar** student of **Graphic Era Deemed to be University, Dehradun** pursuing **B.Tech.** **IIIrd Year VIth Semester** is hereby accepted and approved as a credible work, submitted in the partial fulfilment for the requirement of degree of **B.Tech**. It is a bonafide record of the work done by a team under my supervision during their stay as a project trainee at **OIL AND NATURAL GAS CORPORATION LTD.,** from**June 1, 2022** to **July 31, 2022**.

**(MK MATHUR)**

**Head Computer Services**

**E&D Directorate**

**ONGC, Dehradun**

# ACKNOWLEDGEMENT

The project bears the imprints of the efforts extended by many people to whom, we are deeply indebted.

We are grateful to **Mr. M K Mathur, Mr. P Raja Sekhar, Mr. Amit Kumar, Mr. Mandeep Singh & Mr. Deepak Kumar**, ONGC, Dehradun for giving me the opportunity to work in E & D Directorate for the fulfilment of my project.

We are also grateful to the officers of **Computer Services, E&D Directorate**, who were always available for discussions at length on the various concepts that could be incorporated in the project. Their suggestions have been valuable and their teachings during the course of our discussions would continue to be a guiding principle in my works in the future as well.

We are grateful to the Exploration and Development Directorate department for providing me the opportunity and the environment to enhance my experience of industrial exposure in such short period of time.

## *DECLARATION*

We hereby declare that the project work entitled “**Enterprise Search Using Elastic**

**Search**” is an authenticated work carried out jointly in a team by me at **OIL AND NATURAL GAS CORPORATION LTD**. under the guidance of **Mr. M K Mathur, Head Computer**

**Services, E&D Directorate, Dehradun** for the partial fulfilment of the award of the degree of B.Tech.

Date: 30-july-2022

Place: Dehradun

**Abhishek Bhardwaj**

**Vandit jain**

**Yash Tariyal**

**Anurag kumar**

# INDEX

Table of Contents

**Introduction to Project** .......................................................................................................................................................... **7**

**Introduction to ONGC** ............................................................................................................................................................ **8**

Vision of ONGC.................................................................................................................................................................... 9

Mission of ONGC ............................................................................................................................................................... 10

Objectives of ONGC .......................................................................................................................................................... 11

Brief History ...................................................................................................................................................................... 12

**Project Description** ...............................................................................................................................................................**20**

Purpose of the Project ..................................................................................................................................................... 20

Goals of the Project .......................................................................................................................................................... 21

Given File Introduction ..................................................................................................................................................... 24

Implementation Environment .......................................................................................................................................... 25

**Developing Tools and Frameworks** ..................................................................................................................................... **26**

Python 3 ............................................................................................................................................................................26

Elastic Search .................................................................................................................................................................... 27

HTML-CSS ......................................................................................................................................................................... 27

XAMPP .............................................................................................................................................................................. 28

JAVA Script ........................................................................................................................................................................29

**Approach to the Problem ......................................................................................................................................... 30**

Data Flow Diagram ................................................................................................................................................................

Development-Environment Setup ........................................................................................................................................

Reading CSV file and Pushing into Server .............................................................................................................................

Creating GUI ..........................................................................................................................................................................

Performing Queries on database....………………..…………….…………….………………………………………………..……………. ss Multi-Field and Multi-Keyword Searching……..………………………………………………..………………………………..….……. aa Presenting data in tables.....................................................................................................................................

**Conclusion and Future Scope** ...................................................................................................................................................  **FinalProject** ..............................................................................................................................................................................

**References** .......................................................................................................................................................................... **37**

**Enterprise Search Using Elastic Search**

ONGC is a very big organization, it is among the few companies in the country which have very large number of employees. so, a efficient software is needed to maintain their database. Computer Services, E&D Dte., houses a number of in-house Web and Desktop based solutions and are being used in different offices of ONGC. These facilitates the end user to manage their data efficiently. Over the period, a large repository of data in terms of semi-structured files such as Word, Excel, Powerpoint, PDF etc.. documents and organized data stored in relational databases such as Oracle, mysql, postgres have grown manifold.

In the project the ability To build a google like search interface to search for keywords across Files (their content) and Databases (Oracle). It saves lots of time to search any field in the database and gives required information correctly.

The project requires us to use and apply the knowledge of Elastic Search to search faster on the present datasets. Since we have to make a web page, so we used HTML, CSS, Java Script to make webpage and manage it.

**Introduction To ONGC**

*“I would suggest ONGC to give world leadership in management of energy source, exploration of energy resources, diversification of energy sources, technology in*

*Underground Coal Gasification, an above all, finding new ways of tapping energy wherever it is, to meet the ever growing demand of the country”*

-**HON’BLE EX-PRESIDENT DR. APJ ABDUL KALAM**

Oil and Natural Gas Corporation Limited (ONGC) is an Indian multinational oil and gas company headquartered at Dehradun, India. It is one of the largest Asia-based oil and gas exploration and production companies and produces around **77% of India’s crude oil** (which is equivalent to around **30% of the country’s total demand**) and around **81% of its natural gas**.

It is one of the largest publicly traded companies by market capitalization in India. According to a studies conducted by Platts Energy Business Technology, ONGC is among top 250 Global Energy Companies. Since its inception ONGC has produced more than **600 million metric tonnes of crude oil** and supplied more than **200 billion cubic meters of gas**.

Today, **ONGC is India’s highest profit making corporate** and as per a survey conducted by the US-based magazine ‘Global Finance’ ONGC has been judged as **Asia’s best Oil & Gas company.**

ONGC leads the list of Indian companies listed in Forbes 400 Global Corporates and Financial Times Global 500 by Market Capitalization. Its only fully-integrated petroleum company in India, operating along the entire hydrocarbon value chain.

## VISION OF ONGC

“To be a World-Class Oil and Gas Company integrated in Energy Business with dominant Indian leadership and Global Presence.”

## STRATEGIC VISION: 2001-2021

Focusing on core business of E&P, ONGC has set strategic objectives of:

* Doubling reserves (i.e., accreting 7 billion tonnes of O+OEG) by 2021; out of these 5 billion tonnes are targeted from the Deep waters.
* Improving average recovery from 28 per cent to 40 per cent.
* Tie-up 20 MMTPA of equity Hydrocarbon from abroad.
* The focus of management will be to monetise the assets as well as to assetise the money.

## MISSION OF ONGC

* Dedicated to excellence by leveraging competitive advantages in R&D and technology with involved people.
* Imbibe high standards of business ethics and organizational values.
* Abiding commitment to health, safety and environment to enrich quality of Community

life.

* Foster a culture of trust, openness and mutual concern to make working a stimulating

&challenging experience for our people.

* Strive for customer delight through quality products and services.
* Integrated in Energy Business.
* Focus on domestic and international oil & gas exploration and production business opportunities.
* Providing value linkages in other sectors of energy business.
* Creating growth opportunities and maximize shareholder value.
* Retain dominant position in Indian Petroleum sector and enhance India's energy

availability.

## OBJECTIVES OF ONGC

* To maximize production of hydrocarbon, self-reliance in technology, promoting indigenous efforts to achieve
* self-reliance in technology, promoting indigenous efforts to achieve in all related equipment, material and services.
* Assist in conservation of oil, more efficient use energy and development of alternate source of energy.
* Environmental protection.
* Observe 100% safety in work.

## BRIEF HISTORY

ONGC was set up under the visionary leadership of **Pandit Jawahar Lal Nehru**, going against the wisdom of the then multinational oil companies operating in the country, who had almost written India off as a “Hydrocarbon Barren” country.

Pandit Nehru reposed faith in **Shri Keshav Dev Malviya** who laid the foundation of ONGC in the form of Oil and Gas division, under Geological Survey of India, in 1955. A few months later, it was converted into an Oil and Natural Gas Directorate.

The Directorate was converted into Commission and christened Oil & Natural Gas Commission on **14th August 1956**. In 1994, Oil and Natural Gas Commission was converted in to a Corporation, and in 1997 it was recognized as one of the **NAVRATNAS** by the Government of India. Subsequently, it has been conferred with **MAHARATNAS** status in the year 2010.

Over 50 years of its existence ONGC has crossed many a milestone to realize the energy dreams of India. The journey of ONGC, over these years, has been a tale of conviction, courage and commitment. ONGCs’ superlative efforts have resulted in converting earlier frontier areas into new hydrocarbon provinces.

From a modest beginning, ONGC has grown to be one of the largest E&P companies in the world in terms of reserves and production. ONGC as an integrated Oil & Gas Corporate has developed in-house capability in all aspects of exploration and production business i.e., Acquisition, Processing & Interpretation (API) of Seismic data, drilling, work-over and well stimulation operations, engineering & construction, production, processing, refining, transportation, marketing, applied R&D and training, etc.

Today, Oil and Natural Gas Corporation Ltd. (ONGC) is, the leader in Exploration & Production

(E&P) activities in India having 72% contribution to India’s total production of crude oil and 48% of natural gas. ONGC has established more than 7 Billion Tonnes of in-place hydrocarbon reserves in the country. In fact, six out of seven producing basins in India have been discovered by ONGC. **ONGC produces more than 1.27 million Barrels of Oil Equivalent (BOE) per day.** It also contributes over three million tonnes per annum of Value-Added-Products including LPG, C2 - C3, Naphtha, MS, HSD, Aviation Fuel, SKO etc.

**PRODUCTS & SERVICES**

* Crude Oil.
* Natural Gas.
* Liquefied petroleum Gas.
* Natural Gasoline.
* Ethane/Propane.
* Achromatic Naphtha.
* Superior Kerosene Oil

* Chief Drilling Services, Mumbai
* Chief Well Services, Mumbai
* Chief Geo- Physical Services, Dehradun
* Chief Logging Services, Baroda
* Chief Engineering Services, Mumbai
* Chief Offshore Logistics, Mumbai
* Chief Technical Services, Mumbai
* Chief Info-com Services, New Delhi
* Chief Corporate Planning, New Delhi
* Chief Human Resource Development, Dehradun
* Chief Employee Relations, Dehradun
* Chief Security, Dehradun
* Company Secretary, New Delhi
* Chief Marketing, New Delhi
* Chief Corporate Affairs &Co-ordination, New Delhi
* Chief Corporate Communication, New Delhi
* Chief Material Management, Dehradun
* Chief Technical Services, Dehradun
* Chief Health, Safety & Environment, Mumbai
* Chief Legal, New Delhi
* Chief Medical, Dehradun
* Chief Internal audit, New Delhi
* Chief Commercial, New Delhi
* Chief Exploration & Development, Dehradun

## Some Impressive facts about ONGC

### 1. State owned

One of the biggest advantages & strength of the company is that it is state owned. This led the company have great infrastructure with the governments support. The policymaking also becomes easier due to the same reason. Moreover, any undue and sustained pressure creates due impact on the government as well.

### 2. Growing demographics

ONGC went to global fields through its subsidiary, ONGC Videsh Ltd. (OVL). ONGC has made major investments in Vietnam, Sakhalin and Sudan and earned its first hydrocarbon revenue from its investment in Vietnam.

### 3. Top Technology

ONGC is the technological advancements that were implemented over the last few years. The advancements were substantial and improved the company's ability to extract the greatest amount of oil and gas.

### 4. Hard Industry for Competitor to enter

The oil sector is an industry wherein not many competitors can enter owing to the scale and government intervention.

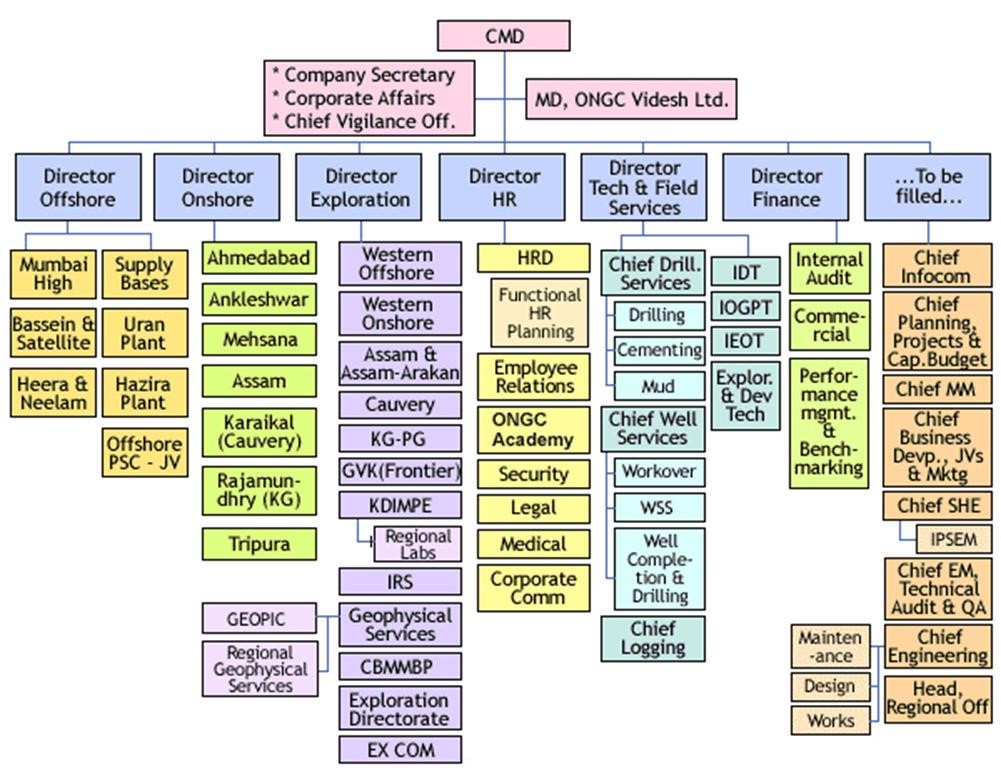
### 5. Strive to be environmentally friendly

The Company has in its guiding principles to cut down emission and become nature friendly in due course of time.

### 6. Strong Infrastructure

The company implemented some well needed improvements to the infrastructure and created a strength for the company.

### ORGANIZATIONAL STRUCTURE



“The structure is a strictly hierarchical organization with strictly defined roles and there is a unity of command which it makes it one of the most efficient companies of the country and best

MAHARATNA PSU.”

## EMPLOYEES & ENVIRONMENT

“To adopt and continuously innovate best-in-class HR practices to support business leaders through engaged, empowered and enthused employees.”

This shows that the ONGC nurtures best practices within the company environment and promotes innovation and the management is able to motivate the employees to perform their best. ONGC has 33,000 employees on its rolls. Of this number, 24,000 work across the company’s various offices, while 9,000 are employed on rigs.

**Employee management “Treating Employees like Customers”**

Companies should have a similar approach to employees and customers. A company should strive to retain an employee in the same way it tries to retain a customer.

ONGC is an organization which is a pride for the nation, the reason being its excellence in its performance. The reason behind this is its strategies outside as well as inside the organization. ONGC is a firm which knows how to deal with the internal as well as the external environment of its functioning.

If we deeply look into its smooth running work environment, then we would find the initiatives this giant corporation has taken for the well-being of its people. ONGC is committed to its value and promise of creating a healthy work environment for its employees, which facilitates their growth, builds up their confidence, and fosters in its employees a sense of belonging as well as job satisfaction.

A lot has already been done here at ONGC, many are in process and much more is planned, for the valuable employees of the renowned ONGC.

ONGC is a firm which believes that the non-financial benefits are as much important as the financial ones, and hence has thoughtfully kept a balance between both, as both are great motivating factors.

## CORPORATE SOCIAL RESPONSIBILITY

The mission of ONGC stated that the company would have an "abiding commitment to health, safety, and environment to enrich quality of community life." And this mission was reflected in its CSR activities. CSR at ONGC began as a philanthropic activity where the company contributed to several socio-economic developmental programs like building schools and hospitals, developing agriculture and cottage industry, building infrastructure facilities, etc., around its areas of operation on an ad hoc basis.

•ONGC is spearheading the United Nations Global Compact - World's biggest corporate citizenship initiative to bring Industry, UN bodies, NGOs, Civil societies and corporate on the same platform.

•During the year, your Company has undertaken various CSR projects at its work centres and corporate level.

•ONGC's CSR programs at the corporate level focused on disaster relief management and water management projects.

•ONGC and the Global Compact Program

The concept of 'global compact' was initiated by UN Secretary-General Kofi Annan in July 2000. The idea was to form an international corporate citizenship network for the advancement of universal social and environmental principles.

**ONGC IN INDIA**



1.ONGC Videsh Limited (OVL)

2.Mangalore Refinery and Petrochemicals Limited (MRPL)

3.ONGC Nile Ganga BV (ONG BV)

4.ONGC Mittal Energy Limited (OMEL)

5.ONGC Mittal Energy Services Limited (OMESL)

6.ONGC Tripura Power Company Pvt. Ltd. (OTPCL)

7. Kakinada Refinery & Petrochemicals Limited (KRPL)

8.Kakinada SEZ Limited

9.Mangalore SEZ Limited

10.Dahej SEZ Limited

11.Rajasthan Refinery Limited (RRL)

### ONGC VIDESH



ONGC Videsh Limited (OVL) is the international arm of ONGC. It was rechristened on 15 June 1989. It currently has 14 oil and projects across 15 countries. It's share of oil and gas production was 8.753 MMT of O+OEG in 2011-12 as against 0.253 MMT of O+OEG in 2002-03. OVL’s overseas cumulative investment has crossed USD 14 billion.

OVL currently has presence in 33 E&P projects in 15 countries, namely: Vietnam, Iraq, Libya, Syria, Sudan, South Sudan, Iran, Cuba, Brazil, Venezuela, Russia, Myanmar, Colombia, Nigeria and Kazakhstan.

Some of the leading alliance partners of OVL are BP, CNPC, Ecopetrol, ENI, Exxon, Statoil Hydro, PDVSA, Petrobras, Petronas, PetroVietnam, Repsol, Rosneft, Shell, Sinopec, Total and TPOC.

**1. Abstract Objective:**

To build a google like search interface to search for keywords across Files (their content) and Databases (Oracle).

Computer Services, E&D Dte., houses a number of in-house Web and Desktop based solutions and are being used in different offices of ONGC. These facilitates the end user to manage their data efficiently. Over the period, a large repository of data in terms of semi-structured files such as Word, Excel, Powerpoint, PDF etc.. documents and organized data stored in relational databases such as Oracle, mysql, postgres have grown manifold.

The objective is to build a Web-based intuitive solution to find each and every information stored in the files and databases by employing latest technologies.

**Components:**

1. **Elastic Search** - Search Database
2. **Data Ingestion Scripts** - Set of Python Scripts to prepare and push data into the Elastic Search Servers.
3. **Web application/portal** - Simple Web page like google interface to display the results of the search performed with additional facilities like filtering, paging, sorting, recent searches etc.

#### 2. Problem Statement

**Enterprise Search Using Elastic Search**

To build a google like search interface to search for keywords across Files (their content) and Databases (Oracle).

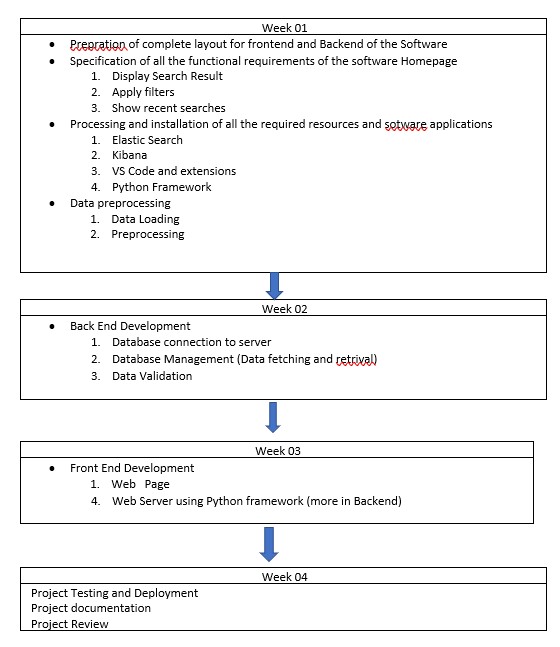
Computer Services, E&D Dte., houses a number of in-house Web and Desktop based solutions and are being used in different offices of ONGC. These facilitates the end user to manage their data efficiently. Over the period, a large repository of data in terms of semi-structured files such as Word, Excel, Powerpoint, PDF etc.. documents and organized data stored in relational databases such as Oracle, mysql, postgres have grown manifold.

#### 3. Scope and Objectives of Project

The objective is to build a Web-based intuitive solution to find each and every information stored in the files(Text, CSV) and databases(JSON, SQL) by applying latest technologies.

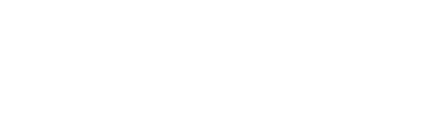
**4.Solution Design**

**Work Flow And Design**

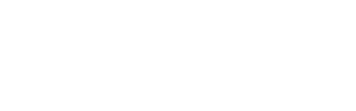




**Web Page**



**Script**



**ElasticSearch**



**Ingestion Scripts**

**5. Implementation Technologies and Platforms**

##### 1.Elastic search:-

Elastic search is a distributed, open-source search and analytics engine built on Apache Lucene and developed in Java.

Elasticsearch allows you to store, search, and analyze huge volumes of data quickly and in near real-time and give back answers in milliseconds.

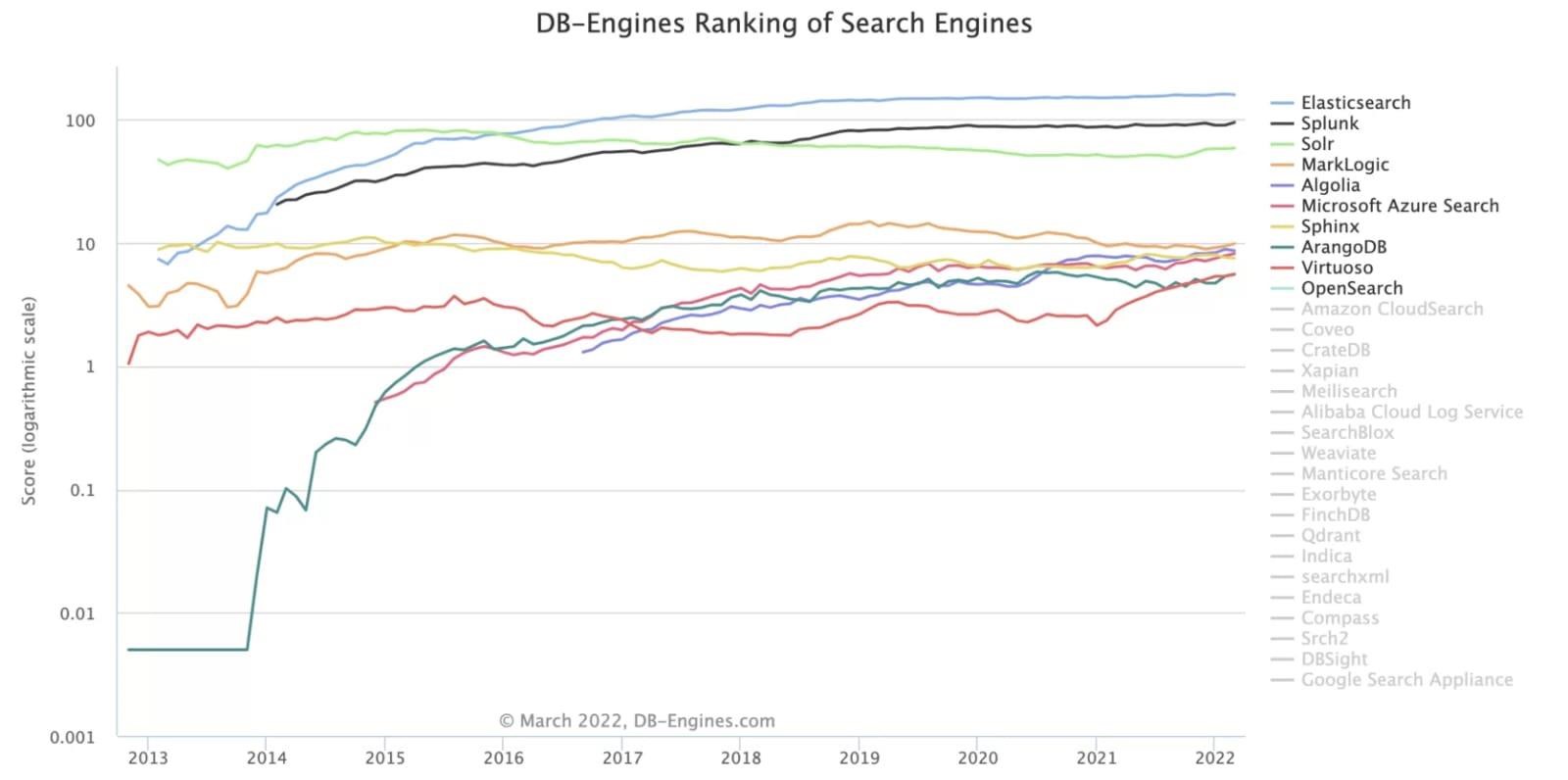
It’s able to achieve fast search responses because instead of searching the text directly, it searches an index. Advantages of elastic search

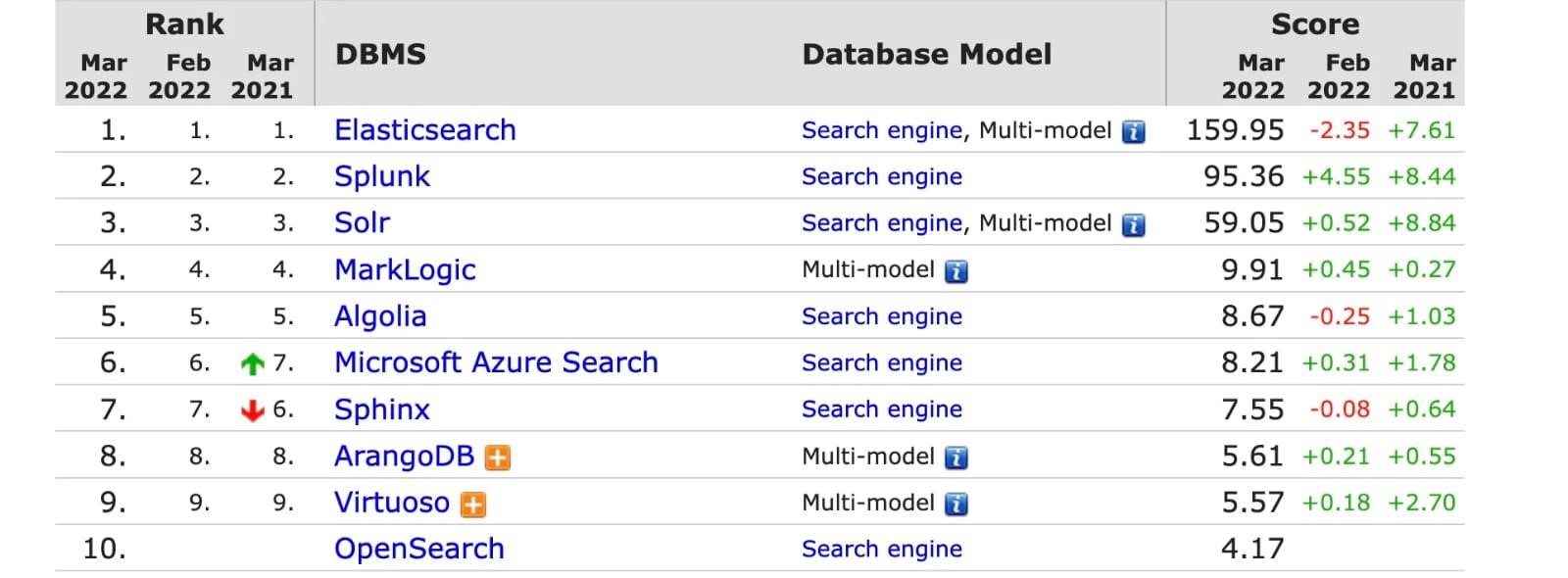
* Lots of search options.
* Document-oriented.
* Speed.
* Scalabilit

**Why to Use Elastic Search ?**

Advantages of ElasticSearch include the following:

* + **Lots of search options**.ElasticSearch implements a lot of features when it comes to search such as customized splitting text into words, customized stemming, faceted search, full-text search, autocompletion, and instant search. Also, fuzzy search is good for spelling errors. You can find what you are searching for even though you have a spelling mistake. Autocompletion and instant search refer to searching while the user types. It can be simple suggestions of existing tags, trying to predict a search based on search history, or just doing a completely new search for every keyword.
  + **Document-oriented**. ElasticSearch stores real-world complex entities as structured JSON documents and indexes all fields by default, with a higher performance result.
  + **Speed**. Speaking of performance, ElasticSearch is able to execute complex queries extremely fast. It also caches almost all of the structured queries commonly used as a filter for the result set and executes them only once. For every other request containing a cached filter, it checks the result from the cache.
  + **Scalability**. Software development teams favor ElasticSearch because it is a distributed system by nature and can easily scale horizontally, providing the ability to extend resources and balance the loading between the nodes in a cluster.
  + **Data record**. ElasticSearch records any changes made in transactions logs on multiple nodes in the cluster to minimize the chance of data loss.
  + **Query fine tuning**. Elastic search has a powerful JSON-based DSL, which allows development teams to construct complex queries and fine tune them to receive the most precise results from a search. It provides also a way of ranking and grouping results.
  + **RESTful API**. ElasticSearch is API-driven, so actions can be performed using a simple RESTful API.
  + **Distributed approach**. Indices can be divided into shards, with each shard able to have any number of replicas. Routing and rebalancing operations are done automatically when new documents are added.
  + **Multi-tenancy**. Often, you have multiple customers or users with separate collections of documents, and a user should never be able to search documents that do not belong to them. This often leads to a design where every user has their own index. Often, this leads to having too many indexes. One larger ElasticSearch index is actually better.





##### How Elastic Search works

To better understand how Elasticsearch works, let’s cover some basic concepts of how it organizes data and its backend components.

###### Documents

Documents are the basic unit of information that can be indexed in Elasticsearch expressed in JSON, which is the global internet data interchange format. You can think of a document like a row in a relational database, representing a given entity — the thing you’re searching for. In Elasticsearch, a document can be more than just text, it can be any structured data encoded in JSON. That data can be things like numbers, strings, and dates. Each document has a unique ID and a given data type, which describes what kind of entity the document is. For example, a document can represent an encyclopedia article or log entries from a web server.

###### Indices

An index is a collection of documents that have similar characteristics. An index is the highest level entity that you can query against in Elasticsearch. You can think of the index as being similar to a database in a relational database schema. Any documents in an index are typically logically related. In the context of an e-commerce website, for example, you can have an index for Customers, one for Products, one for Orders, and so on. An index is identified by a name that is used to refer to the index while performing indexing, search, update, and delete operations against the documents in it.

###### 2.Kibana:-

[Kibana](https://www.knowi.com/blog/grafana-vs-kibana/) is a data visualization and management tool for Elasticsearch that provides real-time histograms, line graphs, pie charts, and maps. It lets you visualize your Elasticsearch data and navigate the Elastic Stack. You can select the way you give shape to your data by starting with one question to find out where the interactive visualization will lead you. For example, since Kibana is often used for log analysis, it allows you to answer questions about where your web hits are coming from, your distribution URLs, and so on. If you’re not building your own application on top of Elasticsearch, Kibana is a great way to search and visualize your index with a powerful and flexible UI. However, a major drawback is that every visualization can only work against a single index/index pattern. So if you have indices with strictly different data, you’ll have to create separate visualizations for each.

**3.Postman:-** Postman is an API(application programming interface) development tool which helps to build, test and modify APIs. Almost any functionality that could be needed by any developer is encapsulated in this tool. It is used by over 5 million developers every month to make their API development easy and simple. It has the ability to make various types of HTTP requests(GET, POST, PUT, PATCH), saving environments for later use, converting the API to code for various languages(like JavaScript, Python).

**Some postman queries used in project:-**

**Put:http://localhost:9200/college for creating datasets**

**Post:http://localhost:9200/college/students/1 for inserting data**

**Get :http://localhost:9200/college/\_search for fetching data**

**Del:http://localhost:9200/college/students/7 for deleting data**

**get:http://localhost:9200/college/students/\_search?q=v1 for searching**

**4.HTML & CSS:-** HTML is used for creating the primary content of a webpage, giving it structure. You start by writing words, then apply tags or elements to these words. The web browser then reads this and can then understand the heading of a page, any paragraphs, and where the page starts and finishes, thus filling your web page with content.

CSS is used for background colour, styling, layout, borders, shadowing – all the essential design bits and bobs that make a webpage look slick and smart. CSS enables you to distinguish between presentation and content by modifying the design and display of HTML elements.

**5.Java script:-** is a lightweight, cross-platform, and interpreted compiled programming language which is also known as the scripting language for webpages. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for [**Client-side**](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments as well as [**Server-side**](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments. Javascript is both imperative and declarative type of language. JavaScript contains a standard library of objects, like [**Array**,](https://www.geeksforgeeks.org/arrays-in-javascript/) [**Date**,](https://www.geeksforgeeks.org/javascript-date-objects/) and [**Math**,](https://www.geeksforgeeks.org/javascript-math-object/) and a core set of language elements like **operators**, **control structures**, and **statements**.

We have added different functionality here like multi match for:

The multi\_match query builds on the match query to allow multi-field queries

i.e search data from given more than one specified fields it improves elastic search functionality

GET

/

\_search

{

"query"

:

{

"multi\_match"

:

{

"query"

:

"this is a test"

,

"fields"

:

[

"subject"

,

"message"

]

}

}

}

1.

The query string.

2.The fields to be queried.

There are different types of Multi Match types in elastic search are:

The way the multi\_match query is executed internally depends on the type parameter, which can be set to:

best\_fields

(default) Finds documents which match any field, but uses the \_score from the best field.

most\_fields

Finds documents which match any field and combines the \_score from each field.

cross\_fields

Treats fields with the same analyzer as though they were one big field. Looks for each word in any field.

phrase

Runs a match\_phrase query on each field and uses the \_score from the best field.

phrase\_prefix

Runs a match\_phrase\_prefix query on each field and uses the \_score from the best field

bool\_prefix

Creates a match\_bool\_prefix query on each field and combines the \_score from each field.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **best\_fields** | |  | | | | | |
| The For | best\_fields | | type is most useful when you are searching for multiple words best found in the same field. instance “brown fox” in a single field is more meaningful than “brown” in one field and “fox” in the other. | | | | |
|  | |
| The | best\_fields | | type generates a | [match](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query.html) | [query](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query.html) for each field and wraps them in a | [dis\_max](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-dis-max-query.html) | query, to find the |
| single best matching field. For instance, this query: | | | | | | | |
| GET /\_search {  "query": {  "multi\_match" : {  "query": "brown fox",  "type": "best\_fields",  "fields": [ "subject", "message" ],  "tie\_breaker": 0.3  }  }  } | | | | | | | |
|  | | | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| GET /\_search {  "query": {  "dis\_max": {  "queries": [  { "match": { "subject": "brown fox" }},  { "match": { "message": "brown fox" }}  ],  "tie\_breaker": 0.3  }  }  } | | | |
|  | | | |
| **most\_fields** | |  | |
| The | most\_field | | type is most useful when querying multiple fields that contain the same text analyzed in different ways. For instance, the main field may contain synonyms, stemming and terms without diacritics. A |
|  | |

would be executed as: second field may contain the original terms, and a third field might contain shingles.

This query:

|  |
| --- |
| GET /\_search {  "query": {  "multi\_match" : {  "query": "quick brown fox",  "type": "most\_fields", |

"fields"

:

[

"title"

,

"title.original"

,

"title.shingles"

]

}

}

}

would be executed as:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GET /\_search {  "query": {  "bool": {  "should": [  { "match": { "title": "quick brown fox" }},  { "match": { "title.original": "quick brown fox" }}, { "match": { "title.shingles": "quick brown fox" }}  ]  }  }  } | | | | |
| The score from each | match | clause is added together, then divided by the number of | match | clauses. |
|  |  |

Also, accepts analyzer, boost, operator, minimum\_should\_match, fuzziness, lenient, prefix\_length, max\_expansions , fuzzy\_rewrite, and zero\_terms\_query. **phrase and phrase\_prefix**

The phrase and phrase\_prefix types behave just like [best\_fields,](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-multi-match-query.html#type-best-fields) but they use a match\_phrase or match\_phrase\_prefix query instead of a match query.

This query:

GET

/

\_search

{

"query"

:

{

"multi\_match"

:

{

"query"

:

"quick brown f"

,

"type"

:

"phrase\_prefix"

,

"fields"

:

[

"subject"

,

"message"

]

}

}

}

would be executed as:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| GET /\_search {  "query": {  "dis\_max": {  "queries": [  { "match\_phrase\_prefix": { "subject": "quick brown f" }},  { "match\_phrase\_prefix": { "message": "quick brown f" }}  ]  }  }  } | | | | | | | | | | | | | |
| Also, accepts | analyzer, | boost, | lenient | | and | zero\_terms\_query | | as explained in [Match,](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query.html) as well as | | | slop | which is |  |
| explained in [Match phrase.](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query-phrase.html) Type | | | | phrase\_prefix | | | additionally accepts | | max\_expansions | . | | | |
|  | | |  |

###### cross\_fields

The cross\_fields type is particularly useful with structured documents where multiple fields **should** match. For instance, when querying the first\_name and last\_name fields for “Will Smith”, the best match is likely to have “Will” in one field and “Smith” in the other.

One way of dealing with these types of queries is simply to index the first\_name and last\_name fields into a single full\_name field. Of course, this can only be done at index time.

The cross\_field type tries to solve these problems at query time by taking a *term-centric* approach. It first analyzes the query string into individual terms, then looks for each term in any of the fields, as though they were one big field.

A query like:

|  |
| --- |
| GET /\_search {  "query": {  "multi\_match" : {  "query": "Will Smith",  "type": "cross\_fields", "fields": [ "first\_name", "last\_name" ],  "operator": "and"  }  }  } |

is executed as:

+(first\_name:will last\_name:will)

+(first\_name:smith last\_name:smith)

**cross\_field**

|  |  |  |
| --- | --- | --- |
| The | cross\_field | type can only work in term-centric mode on fields that have the same analyzer. Fields with the |
| same analyzer are grouped together as in the example above. If there are multiple groups, the query will use | | |

the best score from any group.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| For instance, if we have a | first | and | last | field which have the same analyzer, plus |
| a first.edge and last.edge which both use an edge\_ngram analyzer, this query: | | | | |
| GET /\_search {  "query": {  "multi\_match" : {  "query": "Jon",  "type": "cross\_fields",  "fields": [  "first", "first.edge",  "last", "last.edge"  ]  }  }  } | | | | |
|  | | | | |

would be executed as:

blended("jon", fields: [first, last])

| (

blended("j", fields: [first.edge, last.edge]) blended("jo", fields: [first.edge, last.edge])

blended("jon", fields: [first.edge, last.edge])

)

**bool\_prefix**

The

bool\_prefix

type’s scoring behaves like

[most\_field](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-multi-match-query.html#type-most-fields)

[s](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-multi-match-query.html#type-most-fields)

[,](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-multi-match-query.html#type-most-fields)

but using a

[match\_bool\_prefi](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-bool-prefix-query.html)

[x](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-bool-prefix-query.html)

[quer](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-bool-prefix-query.html)

[y](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-bool-prefix-query.html)

instead of

a

matc

h

query.

GET

/

\_search

{

"query"

{

:

"multi\_match"

{

:

"query"

:

"quick brown f"

,

"type"

:

"bool\_prefix"

,

"fields"

:

[

"subject"

,

"message"

]

}

}

}

The

analyzer

,

boos

t

,

operator

,

minimum\_should\_match

,

lenient

,

zero\_terms\_query

,

and

auto\_ge

n

erate\_synony

ms\_phrase\_query

parameters as explained in

[match quer](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query.html)

[y](https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query.html)

are supported.

The

fuzziness

,

prefix\_length

,

max\_expansions

,

fuzzy\_rewrite

, and

fuzzy\_transposition

s

parameters are

suppor

ted for the terms that are used to construct term queries, but do not have an effect on the prefix query

constructed from the final term.

The slop parameter is not supported by this query type.

Python Requests:-

query={

"\_source" : ["title"],

"size" : 10,

"min\_score":4.0,

"query": {

"bool": {

"must" : [

{

"match": {

"title": "terminator"

}

}],

"filter" : [],

"should" : [],

"must\_not" : []

}

}

}

a

=

requests

.

get

(

'http://localhost:9200/mynetflix/\_search'

,

json

=

query

)

query

={

"\_source"

:

[

"title"

]

,

"query"

:

{

"bool"

{

:

"must"

:

[

{

"match"

{

:

"title"

:

"Nazi"

}

}]

,

"filter"

[],

:

"should"

:

[],

"must\_not"

[]

:

}

}

}

a

=

requests

.

delete

(

'http://localhost:9200/mynetflix'

,

json

=

query

)

|  |
| --- |
| query={ "\_source" : ["title"], "query": {  "bool": {  "must" : [  ],  "filter" : [],  "should" : [],  "must\_not" : []  }  } }  myobj={'Name':'Prithviraj'} headers={'Accept':'application/json','Content-type':'application/json'} result=requests.post('http://localhost:9200/mynetflix/\_search',headers=headers,json=query) |

**6. Deployment and Testing of the Software**

**About Dataset**

Context

The Global dataset of oil and natural gas production, prices, exports, and net exports.

Content

Oil production and prices data are for 1932-2014 (2014 data are incomplete); gas production and prices are for 1955-2014; export and net export data are for 1986-2013. Country codes have been modified from earlier versions to conform to Correlates of War (COW) and Quality of Government (QOG) standards

**Deployed on Xampp Server for accessing through localhost server.**

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was developed by the **Apache Friends**, and its native source code can be revised or modified by the audience. It consists of **Apache HTTP Server, MariaDB, and interpreter** for the different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux. **Steps to deploy webpage on locsalhost:-**

1.Install Xampp on your system.

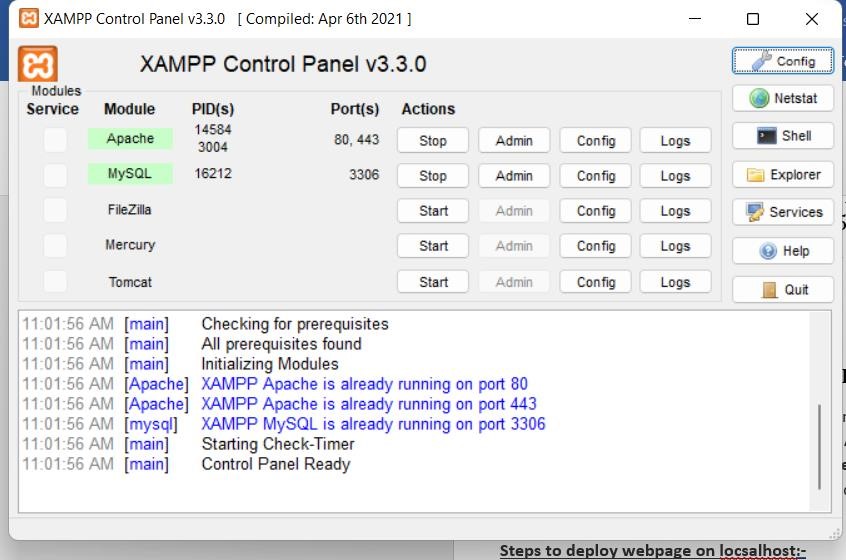
2.start Apache and MySql.

3.find your xampp folder where you installed it and go htdocs.

4.put your index file in htdocs and name it as index.php .

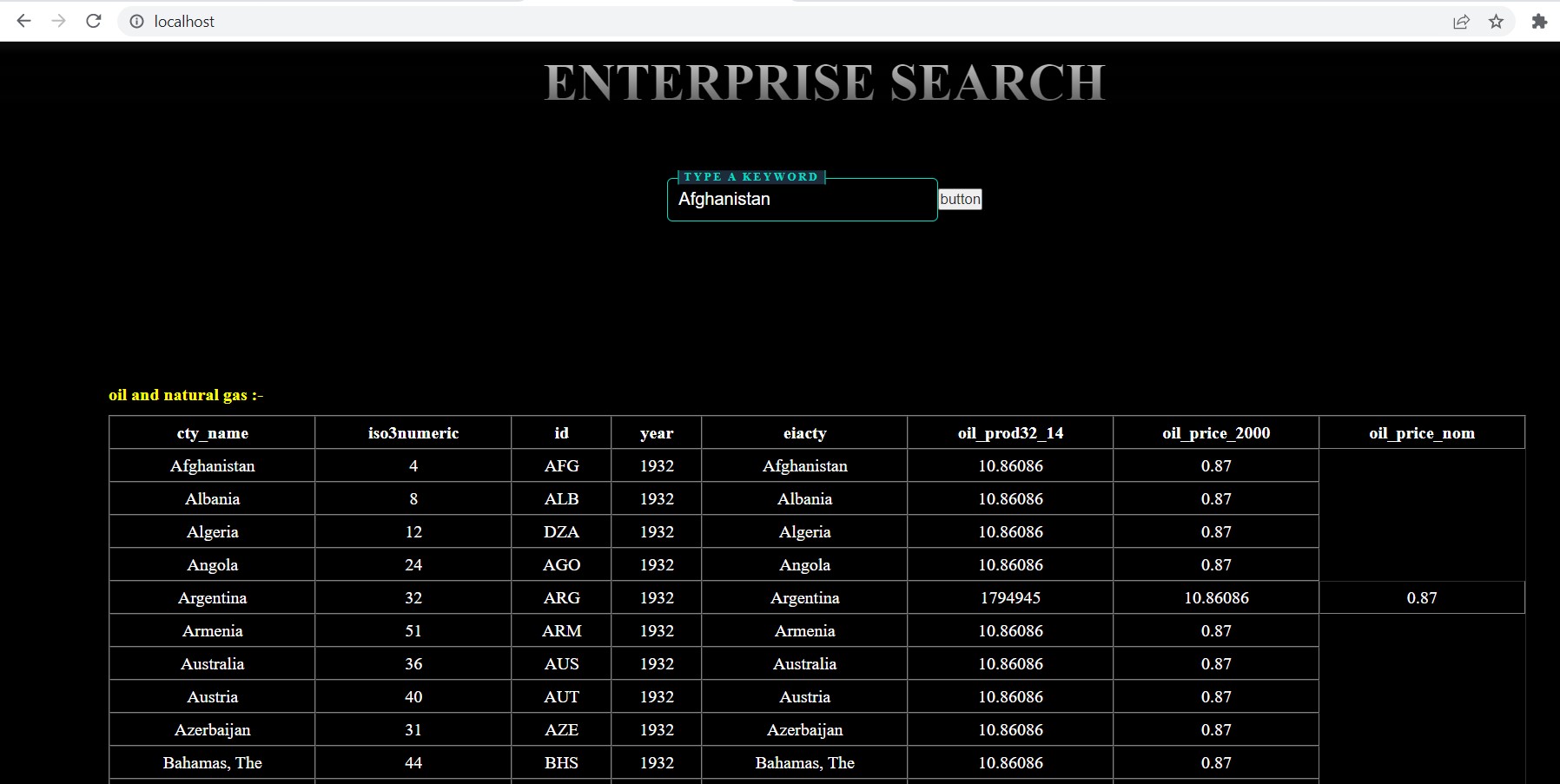
5.along with index file put all the css files also in htdocs.

6.Now,search localhost on your browser and your web page is deployed on localhost.



#### 7. Output Snapshots





**8. Future Scope**

* Multiple type database search
* Apart from text search,it will be able to search images and other searches.

#### 9. Conclusion

1.After completing this project i get to learn about some new technologies Like Elastic search, how it works and what are its application.

2.I get to know about how to create python scripts to load the huge datasets.

3.in this project in learn how javascript can be connected through elastic search.

# REFERENCES

## Reference 01: <https://www.elastic.co/>

**Reference 02:** [**https://www.w3schools.com/python/module\_requests.asp**](https://www.w3schools.com/python/module_requests.asp)

**Reference 03:** [**https://stackoverflow.com/questions/11593035/beginners**](https://stackoverflow.com/questions/11593035/beginners)[**guide-to-elasticsearch**](https://stackoverflow.com/questions/11593035/beginners-guide-to-elasticsearch)

**Reference 04:** [**https://www.kaggle.com/datasets/raspberrypie/oil-andgas**](https://www.kaggle.com/datasets/raspberrypie/oil-and-gas)