**Group 7: Explore the use of text search tools. Group Members:**

**1. Rishabh Jain (rjain35@hawk.iit.edu, A20495530)**
**2. Siddhant Bhatia (sbhatia14@hawk.iit.edu , A20500508 )**
**3. Utkarsha Malegaonkar (umalegaonkar@hawk.iit.edu, A20493621)**

# Project Draft - Search Engine with Elastic Search

## Introduction

Every day humans are generating billions of logs and all of them may or may not be important. Each log contains some key elements that are unique and will not be the same in the next one. Logs are having vital information in them, and each piece of information is not required every time. So, we will build a tool that can easily obtain data from them according to our dataset using AWS ElasticSearch.

For extracting the required text from a chunk of the data ElasticSearch will be used by hosting elastic search service on AWS integrated with kibana for more convenience while handling and understanding the logs. The sorted data will be used for other purposes too. For example, it can be used for the text auto completion feature, which helps the user while they are searching for a specific query.

## Abstract

Our team will be creating a full app search engine that will have the capacity of an auto-complete system. Apart from the ElasticSearch, its libraries and Kibana, we will be using other technologies like react and flask to create a simple front and backend for our project.

We have finalized that we will be using the Netflix data set for our project. This data set can be easily obtained from the Kaggle. We will be performing automation on the Netflix dataset, like if we are searching for a movie that starts with the letter A, then our system will automatically show all the other relevant movies as a result that starts with the letter A and it will also save the selected movie into our search so that the next time when we search, the previous search should automatically be pre-populated. This is a brief draft of what we will be doing in our project.

## High Level steps

**Data Collection**

Beats will be used for data collection. Beats is a lightweight data shipper. It is a free and open platform for single-purpose data shippers. They send data from hundreds or thousands of machines and systems to Logstash or Elasticsearch.

**Data Aggregation & Processing**

Logstash will be used to centralize, transform & stashing data. Logstash is a free and open server-side data processing pipeline that ingests data from a multitude of sources, transforms it, and then sends it to your favorite "stash."

**Indexing & Storage**

Elasticsearch is a distributed, RESTful search and analytics engine capable of addressing a growing number of use cases. As the heart of the Elastic Stack, it centrally stores your data for lightning-fast search, fine-tuned relevancy, and powerful analytics that scale with ease.

**Analysis & Visualization**

Kibana is a free and open frontend application that sits on top of the Elastic Stack, providing search and data visualization capabilities for data indexed in Elasticsearch. Commonly known as the charting tool for the Elastic Stack, Kibana also acts as the user interface for monitoring, managing, and securing an Elastic Stack cluster — as well as the centralized hub for built-in solutions developed on the Elastic Stack.

## Milestones:

| Sl. No. | Task | Due Date | Owned By |
|---|---|---|---|
| 1 | Starting off with the project, we will first analyze different aspects of our project and download all the necessary technologies that we will be requiring for our project like ElasticSearch, Kibana. | April 18 | Rishabh |
| 2 | Next step will be data set selection and to import the JSON data set that we will be using for the project and integrate the same with the ElasticSearch. | April 20 | Siddhant |
| 3 | After Successful installation, we will be checking whether everything is working fine. If so, then we should be able to see the localhost 9200 for the ElasticSearch. | April 22 | Rishabh & Utkarsha |
| 4 | Now we will be doing analysis stuff in the Kibana for our dataset using the GET command and using different aspects of the ElasticSearch like Search, mapping, aggregations, indexes, documents etc. | April 24 | Utkarsha |
| 5 | At this moment, we need to create an index in ElasticSearch and then read documents from the file and insert them into Elasticsearch. | April 26 | Siddhant & Utkarsha |
| 6 | We have our data in the ElasticSearch, but now we need to query the data from the local ElasticSearch, for this, we will be creating an API with an Endpoint. | April 28 | Rishabh |
| 7 | Since we want matching results related to the data that we are searching, for this we will be using a multi-match query which will give us all the documents where any of the provided fields has a prefix like the provided query text. | April 30 | Siddhant |
| 8 | To display all the above-mentioned functionalities, we will need a basic frontend, for this we will be using CSS, Java script and react to display our project content. | May 1 | Utkarsha |

## References


1. Amazon OpenSearch Service Documentation https://docs.aws.amazon.com/opensearch-service/?id=docs_gateway

2. Amazon Web Services, Amazon Elasticsearch Service: Developer Guide, Kindle Edition

3. Alberto Paro, Elasticsearch 7.0 Cookbook: Over 100 recipes for fast, scalable, and reliable search for your enterprise, 4th Edition

4. Anurag Srivastava, Learning Elasticsearch 7.x: Index, Analyze, Search and Aggregate Your Data Using Elasticsearch, 1st Edition

5. Darshita Kalyani, Dr. Devarshi Mehta, Paper on Searching and Indexing Using Elasticsearch