SECURE TRACE

PRACTICE SCHOOL-II

PREPARED BY

Anwesha Sinha (2015BTECHCSE029)

FACULTY GUIDE

DR. S TARUNA

ASSOCIATE PROFESSOR [CSE]

Department of Computer Science Engineering
Institute of Engineering and Technology (IET)
JK Lakshmipat University Jaipur

May 2019

SECURE TRACE

PRACTICE SCHOOL-II

Submitted in partial fulfilment of the requirements

for the degree of

Bachelor of Technology in Computer Science Engineering

By:

Anwesha Sinha (2015BTECHCSE029)

Faculty Guide:

Dr. S Taruna

ASSOCIATE PROFESSOR [CSE]

Department of Computer Science Engineering
Institute of Engineering and Technology (IET)
JK Lakshmipat University Jaipur

May 2019

CERTIFICATE

This is to certify that the Practice School-2 project work entitled ‘’SECURE TRACE VERSION (1.0) PHASE-II” submitted by Anwesha Sinha (2015BTECHCSE029) to- wards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Department of Computer Science & Engineering, JK Lakshmipat University Jaipur is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for Practice School-II examination.

....	....
Dr. S Taruna	**Dr. Sonal Jain**
Associate Professor	**Associate Professor**
Department of Computer Science & Engineering	**Department of Computer Science & Engineering**
institute of Engineering & Technology (IET)	**institute of Engineering & Technology (IET)**
JK Lakshmipat University Jaipur	**JK Lakshmipat University Jaipur**
Date: 16 May 2019	Date: 16 May 2019

....

Dr. Sanjay Goel
Director
Institute of Engineering& Technology
JK Lakshmipat University Jaipur
Date

ACKNOWLEDGEMENT

The internship opportunity I had with [Tekizma] was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me though this internship period.

Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to the CEO of [Tekizma] [Mr. Lakshman Narayanan] who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my project at their esteemed organization and extending during the training.

I express my deepest thanks to [Dr. S Taruna] [Ass. Prof. Computer Science, JKLU] for taking part in useful decision giving necessary advices and guidance and arranged all facilities to make life easier. I choose this moment to acknowledge her contribution gratefully.

It is my radiant sentiment to place on record my best regards, deepest sense of gratitude to [Mr. Yogesh], [Tech Lead], Ms. [Lavania] and Ms. [Vibha] for their careful and precious guidance which were extremely valuable for my study both theoretically and practically.

I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future,
Sincerely,

Name: Anwesha
Sinha.

Place: Bangalore.

Date: 16/05/2019

ABSTRACT

Secure Trace project is a NLP based project on Resume Parser. The goal is to get the candidate’s information after passing the resume through the parser which would eventually help in mapping the candidates to their desired job or mapping the job to the apt person. Resume Parsing is conversion of a free-form resume document into a structured set of information suitable for storage, reporting, and manipulation by software. It helps recruiters to efficiently manage electronic resume documents sent electronically.

Secure Trace Phase-I The concept behind phase 1 is to build a resume parser and extract valuable information from the resume and store it into microsoft sql database using spring maven, nlp core librariries. These fields falls are summarize under categories as:

- Personal Information
- Education
- Work Experience

Secure Trace Phase-II The concept behind phase II is to map the candidates to their desired job or mapping the job to the apt person. The requirement is to develop a fast retrieval search engine which would filter the candidates based on the job requirement and help the candidates filter the appropriate job for them based on the skills acquired by them. It is divided into 3 usecases:

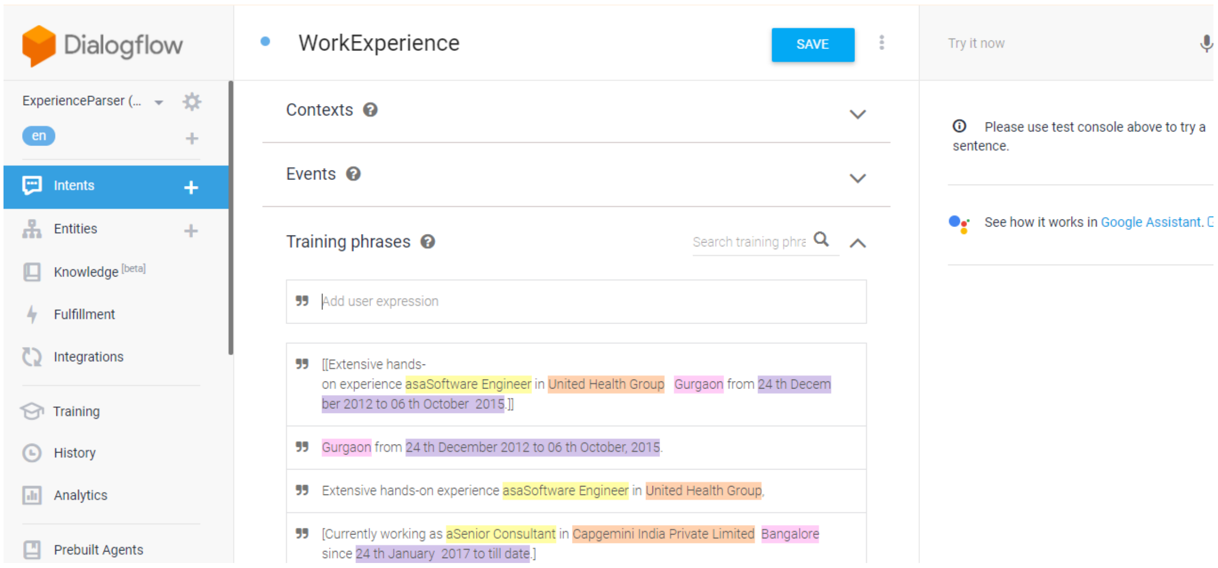
- Use case 1 - Candidate Search
- Use case 2 - Job Search
- Use case 3 - Exact Match

Purpose of Secure Trace is to parse the resume of a candidate and analyze the necessary information of the candidate. To build a resume parser with greater efficiency and the main purpose is not to serve as a database rather a search engine (server) with the main goal of indexing, searching, and providing real-time statistics on the data.

List of Figures

Contents

Contents



[[Extensive hands-on experience asaSoftware Engineer in United Health Group Gurgaon from 24 th December 2012 to 06 th October 2015.]]

Gurgaon from 24 th December 2012 to 06 th October, 2015.

Extensive hands-on experience asaSoftware Engineer in United Health Group.

[Currently working as aSenior Consultant in Capgemini India Private Limited Bangalore since 24 th January 2017 to till date.]

Please use test console above to try a sentence.

See how it works in Google Assistant.

CHAPTER 1 : ABOUT THE COMPANY

Tekizma is a private company that launched in 2015. It has a multidisciplinary team who is passionate about their mission to transform business through innovation, enhancing user efficiency and improving consumer access to the high standard they deserve.

As pioneers of technology, ***T ekizma*** leaders bring a successful track record of delivering solutions that increase consumer access to the high standard of quality they deserve. We employ cutting- edge, state-of- the-art software technologies and strive for quality. At the same time, it believes that customers are the most important asset. It believes in mentor ship and a technically challenging environment that encourages continuous learning. The organisation believe in sharing profits with employees and prioritizing employee growth as much as company growth.

1.1 Tekizma's Products

Tekizma's products streamline the delivery of quality services, establishing consumer confidence, satisfaction, and trust. They offer industry innovation through two product groups.

- ***TekizmaReconSuit*** of products equips systems managers with the means to proactively ensure crosssystem data integrity in health IT implementations, but can be deployed in any industry.
- ***M asterDataM anagement*[*M DM*]** Integrated Health products bring the power of in- formation to patients and clinicians, enabling them to be active partners in health while participating in the highest standard of care.
- ***T ekizmaLabs*** is the research and development arm of our company which design and build technologies which expand access to and elevate the norms of healthcare delivery.

1.2 Customers Partners

Its customers and partners are 9Zest, hCentive, NewYork State, Dell, MassHealth, KPMG, CSRA.

CHAPTER 2 : PROJECT- SECURE TRACE

Secure Trace project is a NLP based project on Resume Parser. The goal is to get the candidate's information after passing the resume through the parser which would eventually help in mapping the candidates to their desired job or mapping the job to the apt person. Resume Parsing is conversion of a free-form resume document into a structured set of information suitable for storage, reporting, and manipulation by software. It helps recruiters to efficiently manage electronic resume documents sent electronically.

2.1 Secure Trace Phase-I

The concept behind phase 1 is to build a resume parser and extract valuable information from the resume and store it into microsoft sql database using spring maven, nlp core librariries. These fields falls are summarize under categories as: 1. Personal Information 2. Education 3. Work Experience

2.2 Secure Trace Phase-II

The concept behind phase II is to map the candidates to their desired job or mapping the job to the apt person. The requirement is to develop a fast retrieval search engine which would filter the candidates based on the job requirement and help the candidates filter the appropriate job for them based on the skills acquired by them.

2.3 Purpose of Secure Trace

Purpose of secure trace is to parse the resume of a candidate and analyze the necessary information of the candidate. To build a resume parser with greater efficiency and the main purpose is not to serve as a database rather a search engine (server) with the main goal of indexing, searching, and providing real-time statistics on the data.

2.4 Design Approach

This section describes the principles and strategies to be used as guidelines when designing and implementing the system. ***Secure racePhase — I.***
This consists of 5 main modules.

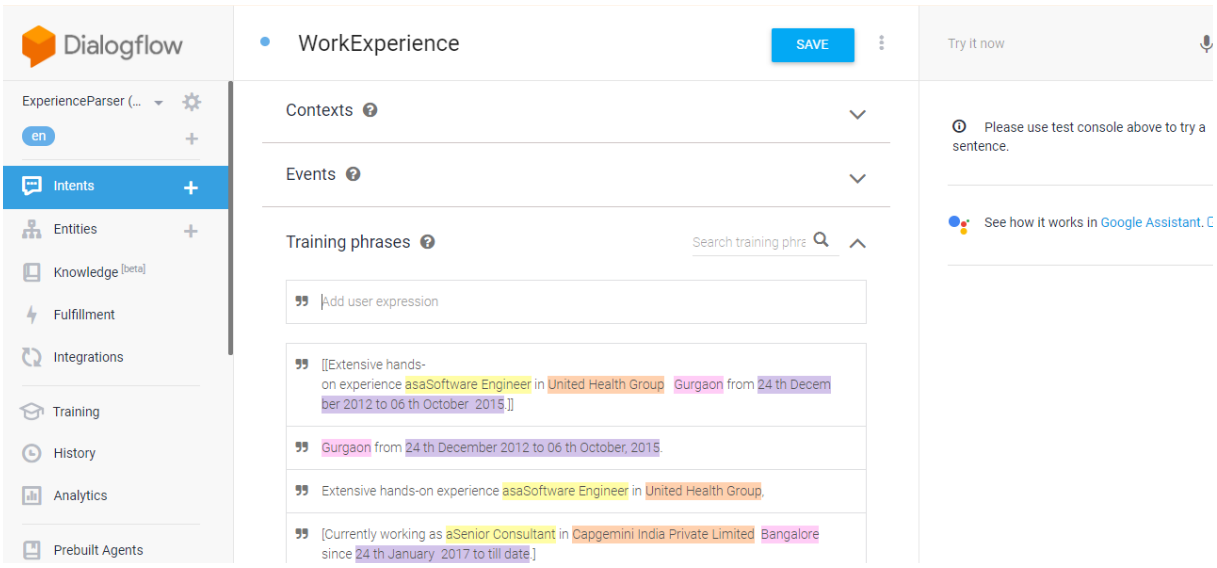
2.4.1 Core Module: This section will have all the common files and constants used throughout the project.
2.4.2 Web Module: This section is required from a user to communicate then we can design a portal.
2.4.3 Analyzer Module: In this section we will analyse or extract the data from a file or Resume and then copy that information to the list of beans.

2.4.4 DB Module: This section is used for database connectivity.

2.4.5 Trainer Module: In this section we are making our own modules and then we can use that module in the analyser with the default models.

For training purpose the requirement was to use and understand the correct training module. We analysed 3 NLP Libraries for the purpose:

1. Microsoft Luis: This was the best NLP for pattern recognition and formation.
2. Google Dialogue Flow: This could be trained best on the basis of Agent, Intents, Entities and even for Junk Data.
3. Stanford Core NLP : It was best for splitting and understanding of entities like Company name, college name.
4. Amazon Lex: It worked on Intents, Utterances, Slots. It could not serve the purpose Of training of data, found best for chatbot purposes.



[[Extensive hands-on experience asaSoftware Engineer in United Health Group Gurgaon from 24 th December 2012 to 06 th October 2015.]]

Gurgaon from 24 th December 2012 to 06 th October, 2015.

Extensive hands-on experience asaSoftware Engineer in United Health Group,

[Currently working as aSenior Consultant in Capgemini India Private Limited Bangalore since 24 th January 2017 to till date.]

Please use test console above to try a sentence.

See how it works in Google Assistant.

Fig-I NLP Library

2.5 Detailed Design

As mentioned in requirement document we are having different section in resume like Personal Information, Address, Education, Work Experience, Skills etc. So Basic flow of Secure trace design is described in picture below. It consists of

- 1. Analyzer
- 2. DB module
- 3. Trainer

Analyzer Module is a core of Secure trace. In this section we will analyze or extract the data from a file or resume and then copy that information to the list of bean. Analyzer will consist of

- 1. Analyze Interface
- 2. Analyze Implementation

Class diagram of Analyze Section

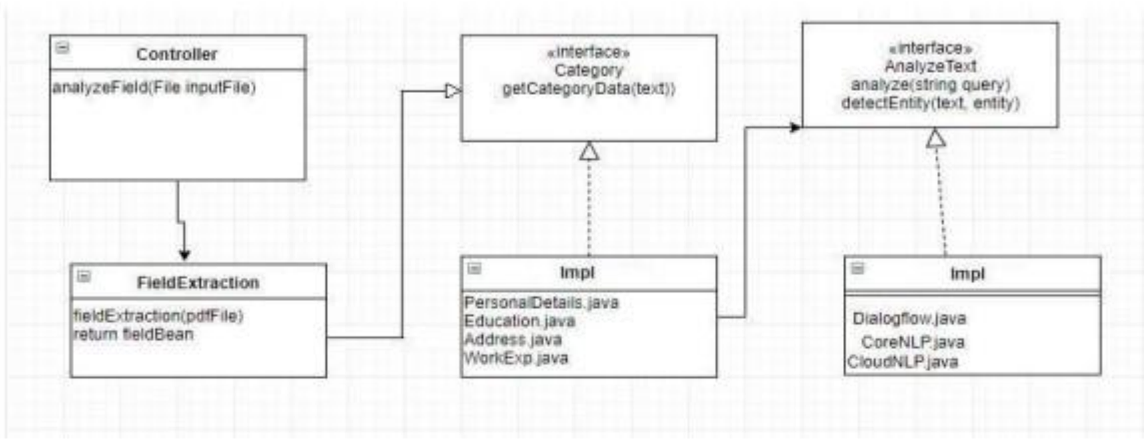


Fig-II Class Diagram

Here we will call analyze() method for analysis which will go into the analyze interface and having implementation in the Service Implementation. From Category we will call one method named getCategoryData() which is having functionality corresponding to all the categories like Personal Details, Address, Education etc. We will be using Category interface for getCategorydata() which will have implementation of Category as defined above education, personaldetails, work exp. etc. detectEntity() will detect the entity of the given string. For example: If we are passing “B. Thomas” then detectEntity() will detect B.Thomas as the “person” i.e. “entity”. fieldExtraction() will display the data of each field separately. It will seprate the fields as well from resumes Such as ‘Education’ is a field in resume and data inside the education will be displayed.

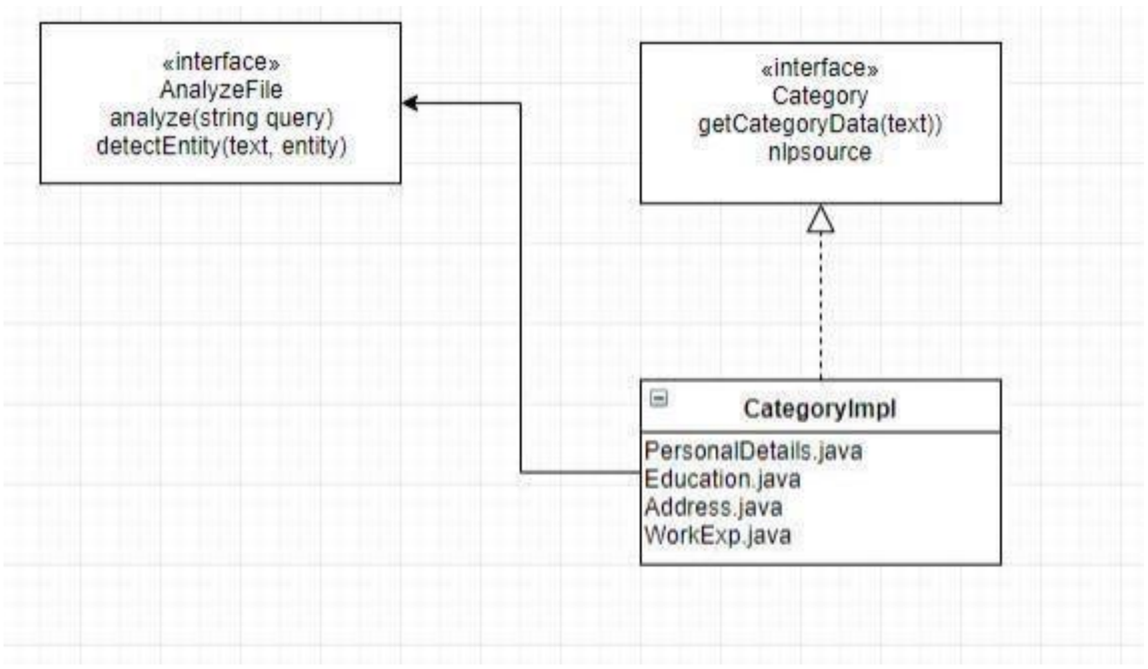


Fig –III Service & Service Impl

2.6 DB Module

DB module is database structure which we will be use in Secure trace. It will use to save the data from bean to the database.

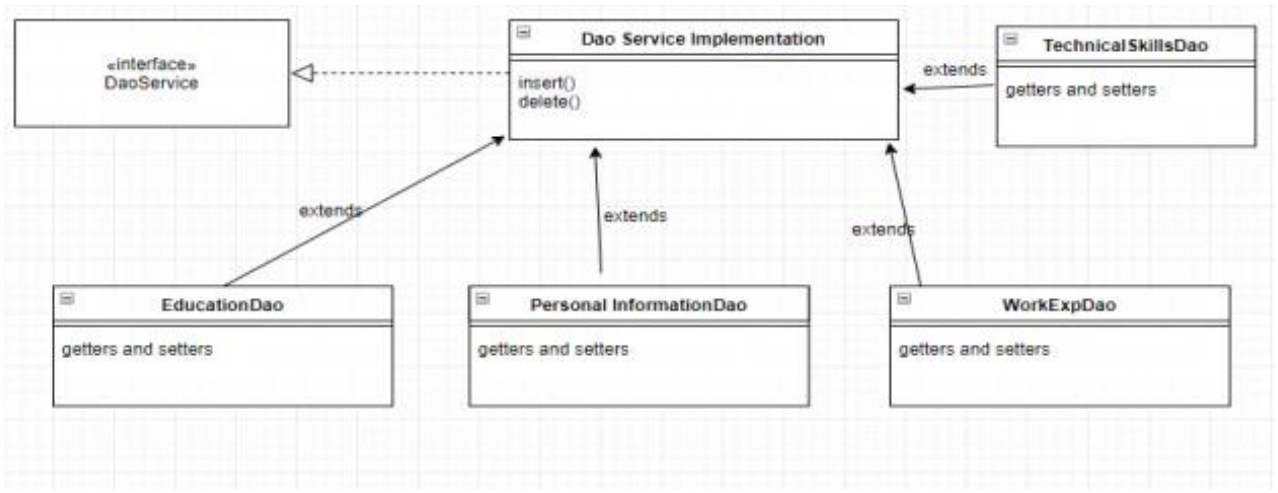


Fig –IV DAO Layer

Common functions like insert() update() delete() etc. will be defined in Dao Service implementation and extends by Category classes like Education, Address etc. Implementation will contain all the queries to get or send the data related to categories.

2.7 Trainer

Trainer is a section where we are making our own models and then we can use that model in the analyzer with the default models. Default models: These are the models which are already present in the API. For Example:- Organizations, Location, Date, Time etc. will get detect by the models which are in the APIs. Custom Models:- These are the models which are created by user and can be merge with the default model. For Example: Skills like Java, C, C++ etc. will be train in this trainer section and then can be used with the default models.

2.8 Technology Used:

- Spring Maven
- Hibernate
- MS SQL Studio
- Stanford Core NLP Library
- Dialogue Flow NLP Library

Chapter 3. Module in Secure Trace Phase-I

3.1 Purpose of Module

Initial work with secure trace was of making pattern recognisation for better resume parsing to be done with Microsoft Luis. The second step was to understand the project for its tesing which led to the study of spring maven, boot, hibernate. The purpose of my module was to increase the efficiency of the resume parser.

3.2 Modules

Modules built in secure trace:

1. Build Python Modules: of spellchecker and word spliiter which recognises itself that the 2 words are joined and it splits it and then pass it into spell checker. 2. Applied python to java connector through jython. My work increased the efficiency of the resume parser and led me to work in its 2nd phase

3.3 Description

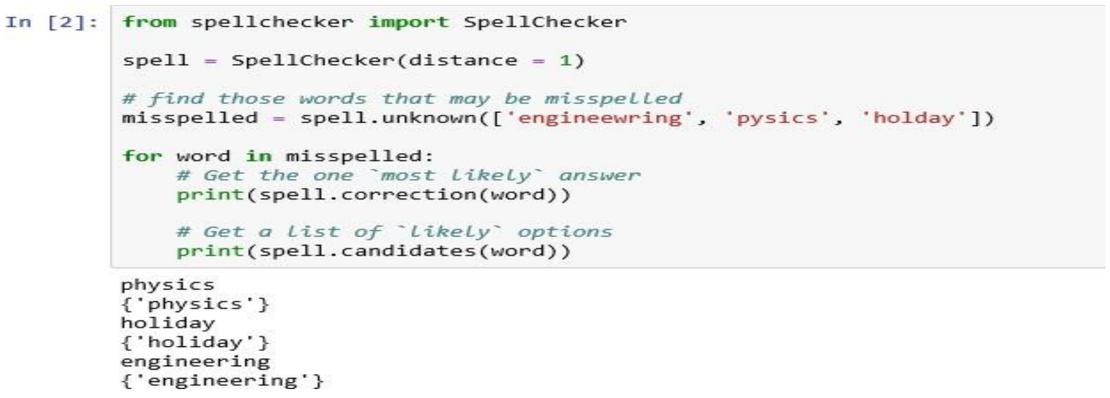
The work was to make a spelling correction checker which could even split the word which were a combination of 2 words.

Before applying the algorithm on spell checker and word splitter it was necessary to preprocess the .txt file by tokenizing each word and passing the document into the spell checker and word splitter.

- Spell Correction Checker
- Word Splitter

3.4 Spell Correction Checker

The SpellChecker library was used for the purpose. It would only return the wrong words after passing the .txt file and would give its most likely correct word for the same. In addition to that it would return all the possible combination of words which could be the correct word. It even considers the combination of 2 words as a wrong word and would return a possible correct word for it. Hence this problem could be solved by word splitter after passing the word as a wrong word it could be sent to the word splitter for further analysis. The added advantage of spellchecker is that it could even work for other languages than English.

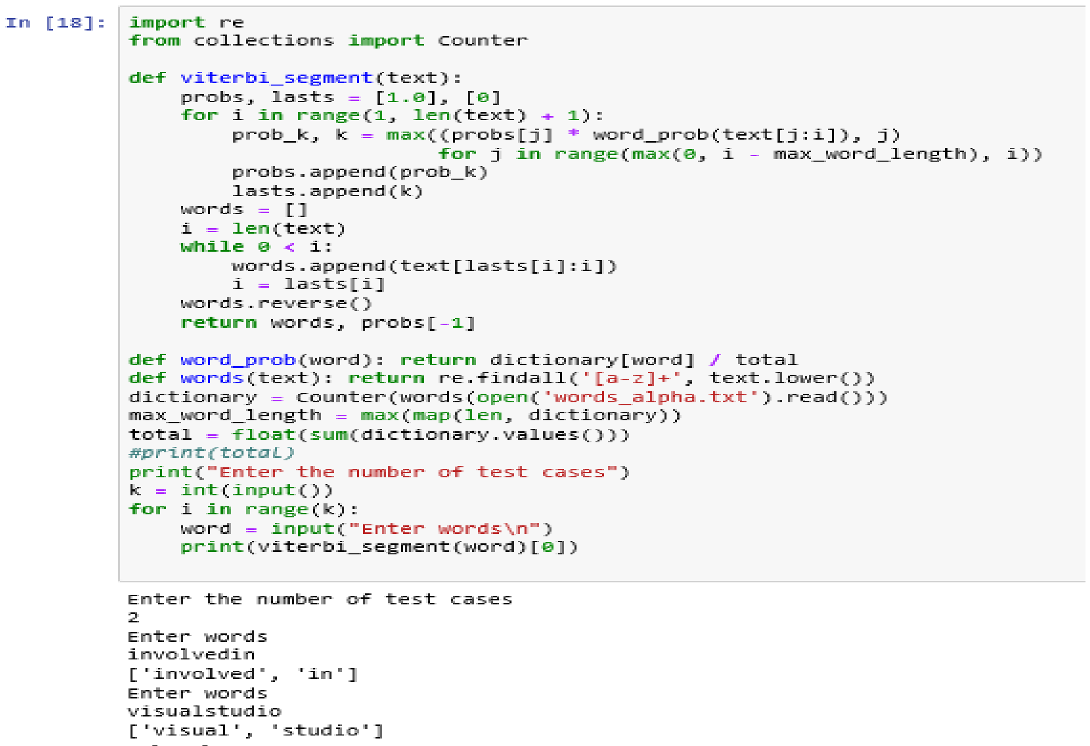


```
In [2]: from spellchecker import SpellChecker

        spell = SpellChecker(distance = 1)

        # find those words that may be misspelled
        misspelled = spell.unknown(['engineewring', 'pysics', 'holday'])

        for word in misspelled:
            # Get the one `most likely` answer
            print(spell.correction(word))

            # Get a list of `likely` options
            print(spell.candidates(word))

        physics
        {'physics'}
        holiday
        {'holiday'}
        engineering
        {'engineering'}
```

Fig –V SpellChecker

3.5 Word Splitter

After passing the words into spell checker function, the wrong words could be split if they are made of combination of 2 correct words. However, if the words are made of the combination of 2 wrong words, it won't split it directly into 2 words. For that purpose, the training file has to be trained to manage the wrong words for splitting them. It works on the probability check of 2 words. Reference: https://en.wikipedia.org/wiki/Viterbi$_{a}$*lgorithm*



```
In [18]: import re
         from collections import Counter

         def viterbi_segment(text):
             probs, lasts = [1.0], [0]
             for i in range(1, len(text) + 1):
                 prob_k, k = max((probs[j] * word_prob(text[j:i]), j)
                                 for j in range(max(0, i - max_word_length), i))
                 probs.append(prob_k)
                 lasts.append(k)
             words = []
             i = len(text)
             while 0 < i:
                 words.append(text[lasts[i]:i])
                 i = lasts[i]
             words.reverse()
             return words, probs[-1]

         def word_prob(word): return dictionary[word] / total
         def words(text): return re.findall('[a-z]+', text.lower())
         dictionary = Counter(words(open('words_alpha.txt').read()))
         max_word_length = max(map(len, dictionary))
         total = float(sum(dictionary.values()))
         #print(total)
         print("Enter the number of test cases")
         k = int(input())
         for i in range(k):
             word = input("Enter words\n")
             print(viterbi_segment(word)[0])

         Enter the number of test cases
         2
         Enter words
         involvedin
         ['involved', 'in']
         Enter words
         visualstudio
         ['visual', 'studio']
```

FIG-VI Word Splitter

3.6 Technology Used:

- Python
- Jython
- Microsoft Luis Pattern Recognisation, NLP Library

Chapter 4: SECURE TRACE PHASE-II

4.1 Purpose of Secure Trace Phase-II

To apply a searching algorithm and generate an API for the companies or organizations which can trace data or resume based on the query search, and generate the required response.

4.2 Flow of The Project

To generate rest API which had function inside it to do crud operation. The rest API will pull information from MS SQL Database and dump it into elastic search after indexing it into elastic search. Further elastic search will apply elastic search operation and pull out data from MS SQL database as output.

4.3Technology Used

- Node JS
- Elastic Search Algorithm
- Microsoft SQL Database

4.4Use case 1 - Candidate Search

To apply a searching algorithm and generate an API for the companies or organizations which can trace data or resume based on the query search, and generate response in the form of list of candidates to the query. The query search in their priority order is as follows:

- Required Skills
- Preferred Skills
- Minimum years of experience

The response of the query would be in the following form: Information of candidate will have following details:

- Name
- Candidate ID
- Overall Match Percentage
- Preferred SKills Matched Percentage
- Year of Experience

- Job Location

4.5 Use Case 2 - Job Search

To apply a searching algorithm and generate an API for the people searching for the job, and generate response. The query search in their priority order is as follows:

- Required Skills
- Preferred Skills
- Minimum years of experience

The response of the query would be in the following form: Information of Job will have following details:

- Job Title
- Job Id
- Job Desc
- Company Id
- Company Name
- Overall Match Percentage
- Preferred SKills Matched Percentage
- Min Year of Experience Required
- Job Location

4.6 Use Case 3 - Exact Match

Based on Candidate Location and Job Location, we will filter Jobs and Candidates with exact match of location. Response will be same like Use Case 1 and Use Case 2 for Candidate Search and Job Search respectively.

Chapter 5: OverView of Secure Trace Phase-II

5.1 Working

The approach is to use elastic search as the searching algorithm and create an API which will take out data from the database and index data to Elastic Search Server. We will also have a job which will keep refreshing new data on daily basis or when required.

To get search result, we will have an seperate API which will receive search parameters (eg: required skills, preferred skills and min experience required) and will filter out candidates. Pagination will also be handled by Elastic Search and count per page can be configured.

Elastic Search Elasticsearch is built on top of Apache Lucene, which is a high performance text search engine library. Although Elasticsearch can perform the storage and retrieval of data, its main purpose is not to serve as a database, rather it is a search engine (server) with the main goal of indexing, searching, and providing real-time statistics on the data.

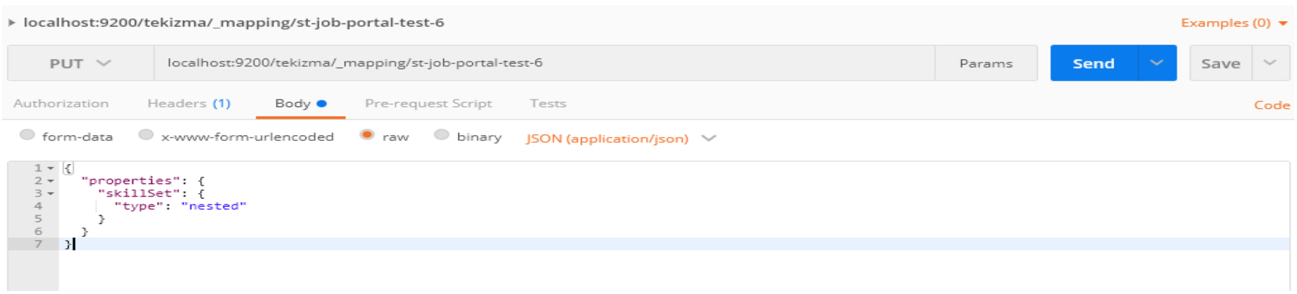
Elasticsearch has a distributed architecture that allows horizontal scaling by adding more nodes and taking advantage of the extra hardware. It supports thousands of nodes for processing petabytes of data. Its horizontal scaling also means that it has a high availability by rebalancing the data if ever any nodes fail.

When data is imported, it immediately becomes available for searching. Elasticsearch is schema- free, stores data in JSON documents, and can automatically detect the data structure and type.

5.2 Elasticsearch Analogy with Relational Databases:

Indices would be like a database, and types like a table in a database

Mapping In Elastic Search For the implementation of elastic search or indexing any data in elastic search engine. The proper mapping should be done that is letting the search engine understand the way we want to store the data. The mappings should be used to hit the API



PUT localhost:9200/tekizma/_mapping/st-job-portal-test-6

raw: JSON (application/json)

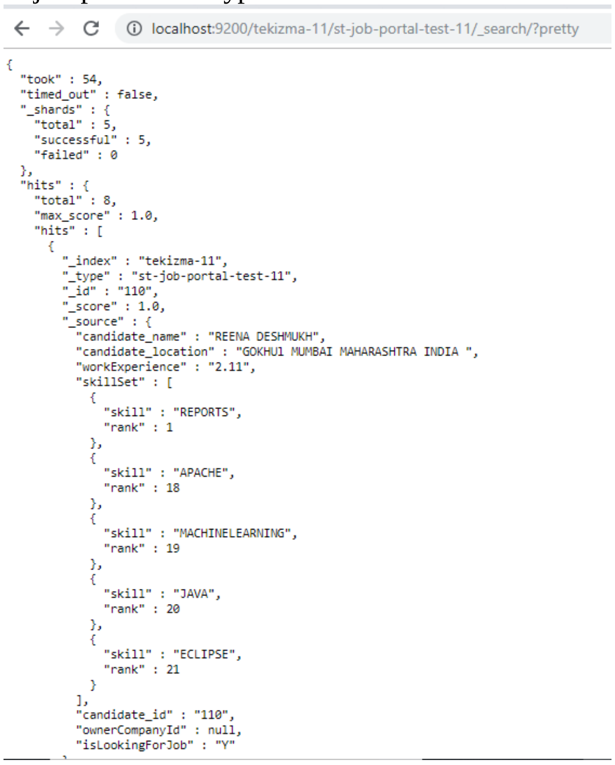
```
{
  "properties": {
    "skillSet": {
      "type": "nested"
    }
  }
}
```

FIG-VII Mapping of Nested Dataset

localhost:9200/tekizma/$_mapping/st - job - portal - test$

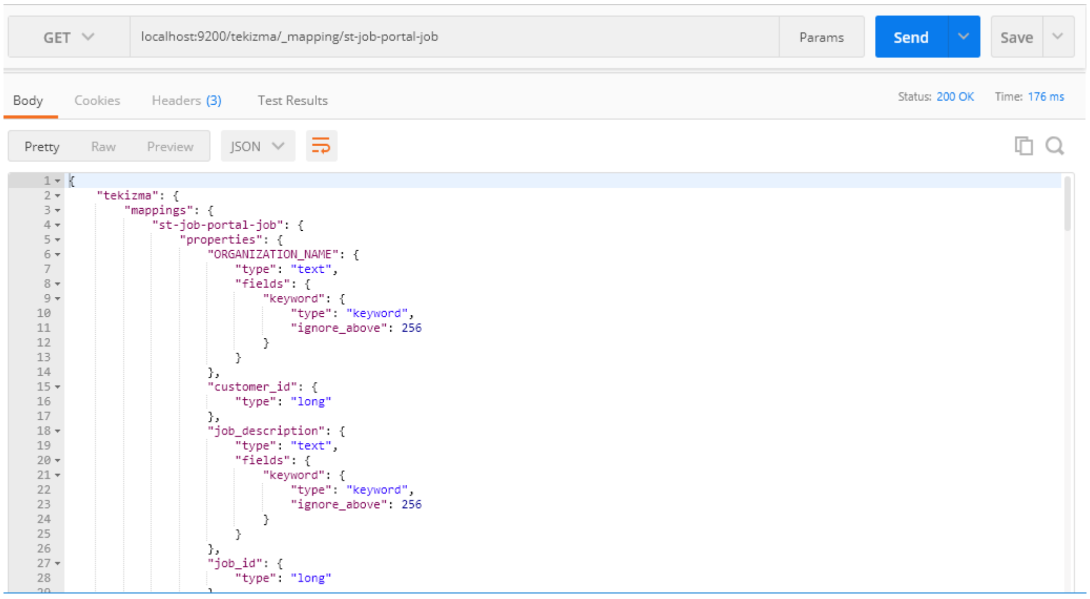
where tekizma: indexname

st-job-portal-test: typename


localhost:9200/tekizma-11/st-job-portal-test-11/_search/?pretty

```
{
  "took" : 54,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "hits" : {
    "total" : 8,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "tekizma-11",
        "_type" : "st-job-portal-test-11",
        "_id" : "110",
        "_score" : 1.0,
        "_source" : {
          "candidate_name" : "REENA DESHMUKH",
          "candidate_location" : "GOKHU1 MUMBAI MAHARASHTRA INDIA ",
          "workExperience" : "2.11",
          "skillSet" : [
            {
              "skill" : "REPORTS",
              "rank" : 1
            },
            {
              "skill" : "APACHE",
              "rank" : 18
            },
            {
              "skill" : "MACHINELEARNING",
              "rank" : 19
            },
            {
              "skill" : "JAVA",
              "rank" : 20
            },
            {
              "skill" : "ECLIPSE",
              "rank" : 21
            }
          ],
          "candidate_id" : "110",
          "ownerCompanyId" : null,
          "isLookingForJob" : "Y"
```

FIG-VIII Mapping leads to Skillset in Elasticsearch as Skill & Rank



```
{
    "tekizma": {
        "mappings": {
            "st-job-portal-job": {
                "properties": {
                    "ORGANIZATION_NAME": {
                        "type": "text",
                        "fields": {
                            "keyword": {
                                "type": "keyword",
                                "ignore_above": 256
                            }
                        }
                    },
                    "customer_id": {
                        "type": "long"
                    },
                    "job_description": {
                        "type": "text",
                        "fields": {
                            "keyword": {
                                "type": "keyword",
                                "ignore_above": 256
                            }
                        }
                    },
                    "job_id": {
                        "type": "long"
```

FIG-IX Original Mapping Vs Created Mapping in Elastic Search

5.3 Justify the Algorithm Used

Elastic search provides distributed, real time, search, analytic performance. It is scalable, fast, multilingual, provides features auto completion, instant search and is schema free.

To apply a searching algorithm and generate an API for the companies or organizations which can trace data or resume based on the query search, and generate the required response.

5.4 Elastic Search Algorithm

Elastic Search

- Node: Single running instance of Elastic Search.
- Cluster: Collection of one or more nodes (server). It provides collective indexing and searching across all nodes.
- Collection of different type of document and document properties. Always a name with all letters small.
- Type: within an index. Logical partition of your index.
- Type/Mapping: Collection of documents sharing a set of common fields present in an index.
- Document: Collection of fields specified in JSON Format.
- Shard: Indexes are horizontally divided into shards. Each shards contains all properties of document but number of json as compared to Index are less.
- Replicas: Copy of index and

shards. Installation:

- Download set up from https://www.elastic.co/downloads/elasticsearch
- Run bin/elasticsearch.bat file.
- It will host elastic search at

http://localhost:9200/ Data Indexing:

- 1. Create index :
  a. Request Method: PUT
  b. API : /¡index$_n$*ame* >
  c. RequestBody : number$_o$f$_s$hrds, number$_o$f$_r$eplicas

i.If notpassed; defaultvalueswillbetaken
d.Response : acknowledged : true

To see all indexes: http://localhost:9200/cat/indices?v

- 2. Create/Replacing Document
  a. Request Method: PUT
  b. API: /¡index$_n$ame > / < type/mapping > /
  < id >
  c. c.RequestBody :
  i.Jsonfilewithinputdocumentfields
  ii.Required
  iii.If < id > isnotpassed, elasticsearchwillgeneratearandomidandwilreturnitinresponse.
  d.Response : ”result” : ”created”, ”shards” : ”total” : 2, ”f ailed” : 0, ”successf ul” : 1,
  ”$_i$ndex” : ”customer”, ”created
- 3. Fetch Document:
  a. Request Method: GET
  b. API: /indexn*ame > / < type/mapping > / < id >*
  c. RequestBody : *NONE*
  d. Resonse : ”found” : true, ”index” : ”customer”, ”type” : ”external”,
  ”source” : , ”id” : ”1”, ”version” : 2
- 4. Delete Index:
  a. Request Method: DELETE
  b. API: /indexname >
  c.RequestBody : N ON E
  d.Response : ”acknowledged” :
  true
- 5. Updating Document
  a. Request Method: POST
  b. API: /¡index$_n$*ame > / < type/mapping > / < id*>
  /update
  c. c.RequestBody :
  i.”doc” : ”name” : ”Y ogeshttupta”, ”age” : ”26”
  d.Response :
  ”result” : ”updated”, ”$_s$hards” : ”total” : 2, ”f ailed” : 0, ”successf ul” : 1,
  ”$_i$ndex” : ”customer”, ”$_t$ype” : ”external”,
- 6. Deleting Document
  a. Request Method: DELETE
  b. API: /¡index$_n$ame > / < type/mapping > /
  < id > c.RequestBody : NONE
  d.Response :
  ”result” : ”deleted”, ”$_s$hards” : ”total” : 2, ”f ailed” : 0, ”successf ul” : 1,
  ”f ound” : true, ”$_i$ndex” : ”customer”, ”$_t$ype
- 7. Bulk Processing
  a. Request Method: POST
  b. API: /¡index$_n$*ame > / < type/mapping >*

/bulk
c. *c.RequestBody* :
”index” : ”id” : ”1””name” : ”JohnDoe””index” : ”id” : ”2”
”name” : ”JaneDoe” Request body json should be separate

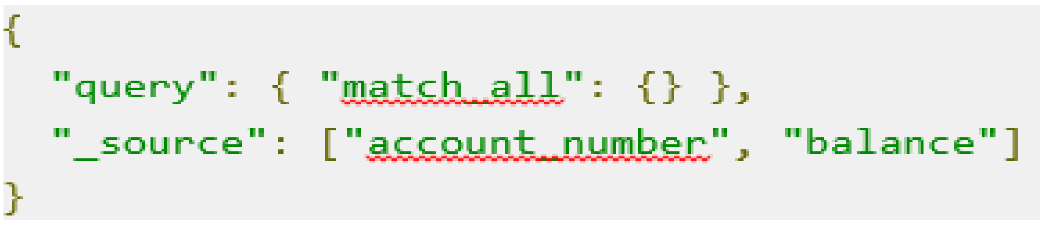
Response:
”took”:206,”items”:[”index”:”result”:”created”,”shards” : ”total” : 2,
”failed” : 0, ”successf ul” : 1, ”index” :
”customer”, ”created” : true, ”type” : ”internal1”, ”id” : ”1”, ”version” : 1,
”status” : 201, ”index” : ”result” : ”creat f alse

Search:
Using
URL:
API: /indexname > /search? < query >

Query Creation:
Search in all documents of index q is the query
parameter.
d. If sorting is required:
Sort = field*name* >: *asc/desc*

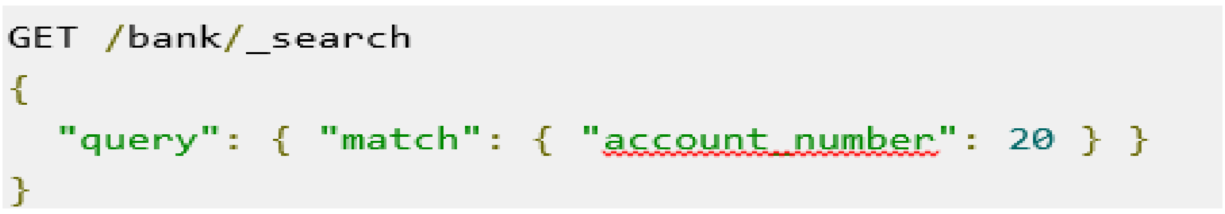
As for the response, we see the following parts:
• took – time in milliseconds for Elasticsearch to execute the search
• timedout–tells us if the search timed out or not
• shards–tells us how many shards were searched, as well as account of
the successful/f ailed searchedshards
hits–search results
hits.total–totalnumberof
documentsmatchingoursearchcriteria
hits.hits–actualarrayof searchresults

To get specific fields from document



```
{
  "query": { "match_all": {} },
  "_source": ["account_number", "balance"]
}
```

returns the account numbered 20:



```
GET /bank/_search
{
  "query": { "match": { "account_number": 20 } }
}
```

returns all accounts containing the term ”mill” in the address:

```
GET /bank/_search
{
  "query": { "match": { "address": "mill" } }
}
```

returns all accounts containing the term ”mill” or ”lane” in the address:

```
GET /bank/_search
{
  "query": { "match": { "address": "mill lane" } }
}
```

returns all accounts containing the phrase ”mill lane” in the address:

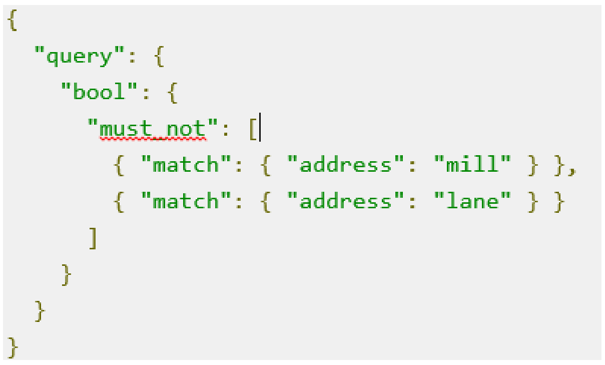
```
GET /bank/_search
{
  "query": { "match_phrase": { "address": "mill lane" } }
}
```

composes two match queries and returns all accounts containing ”mill” and ”lane” in the address:

```
{
  "query": {
    "bool": {
      "must_not": [
        { "match": { "address": "mill" } },
        { "match": { "address": "lane" } }
      ]
    }
  }
}
```

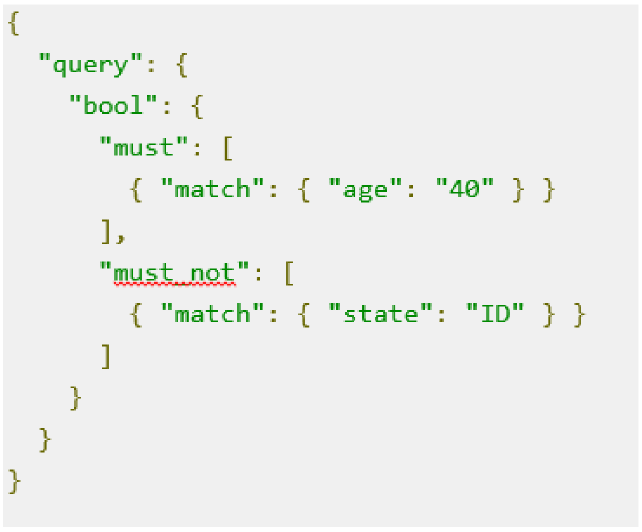
composes two match queries and returns all accounts containing ”mill” or ”lane” in the address: Same as above algorithm but the or query is written under should clause.

composes two match queries and returns all accounts that contain neither ”mill” nor ”lane” in the address:



```
{
  "query": {
    "bool": {
      "must_not": [
        { "match": { "address": "mill" } },
        { "match": { "address": "lane" } }
      ]
    }
  }
}
```

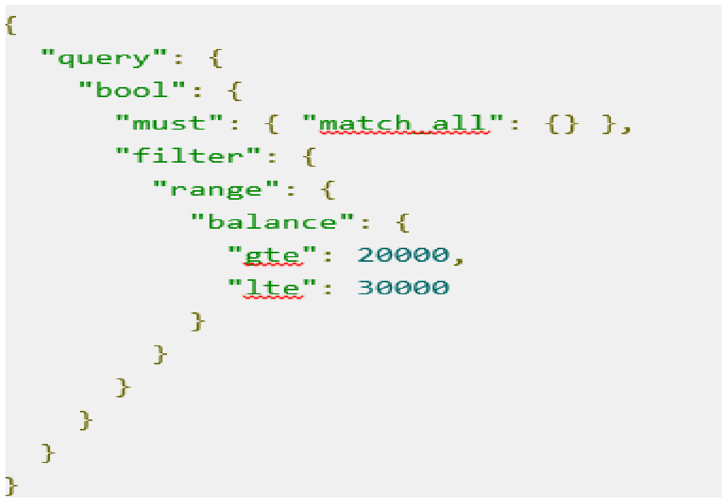
returns all accounts of anybody who is 40 years old but doesn’t live in ID:



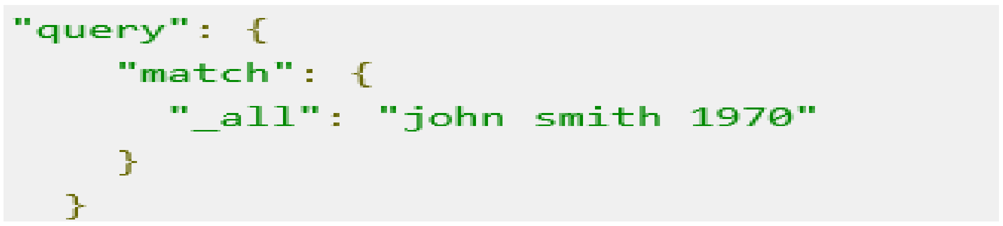
```
{
  "query": {
    "bool": {
      "must": [
        { "match": { "age": "40" } }
      ],
      "must_not": [
        { "match": { "state": "ID" } }
      ]
    }
  }
}
```

return all accounts with balances between 20000 and 30000,

inclusive. Search in all fie



```
{
  "query": {
    "bool": {
      "must": { "match_all": {} },
      "filter": {
        "range": {
          "balance": {
            "gte": 20000,
            "lte": 30000
          }
        }
      }
    }
  }
}
```



```
"query": {
     "match": {
        "_all": "john smith 1970"
     }
  }
```

FIG-X Elastic Search Algorithms

5.5 Build a Search Engine with Node.js and Elasticsearch

Node Javascript Node.js is an open source server environment. Node.js is free. Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.). Node.js uses JavaScript on the server. Node.js can generate dynamic page content. Node.js can create, open, read, write, delete, and close files on the server. Node.js can collect form data. Node.js can add, delete, modify data in your database.

5.6 Purpose of Using NodeJS

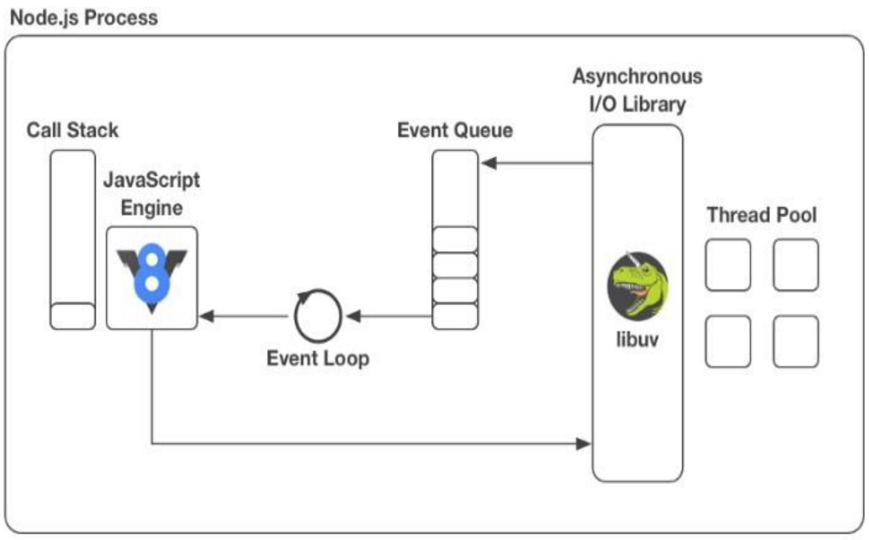
Node.js is a **standalone runtime environment** for JavaScript. That means, Node.js can exe- cute JavaScript code as an independent process on your computer. We can say that Node.js is to JavaScript what the JVM is to Java bytecode, or what the Python interpreter is to Python.

What's special about this is the standalone part. Because, traditionally JavaScript engines where exclusively embedded into web browsers. That means, you could execute JavaScript code only in the context of a website that is being displayed in a web browser. Node.js was one of the first JavaScript engines that was independent of a web browser and runs in its own process.

5.7 How does NodeJS Work

Node.js has the following three main ingredients:

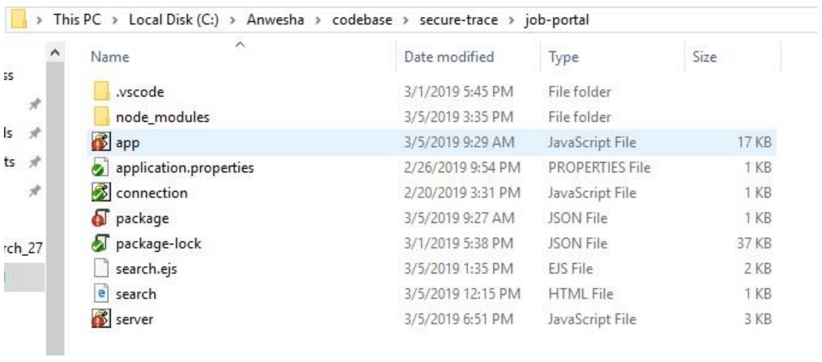
- JavaScript engine
- Asynchronous I/O library
- Event loop



Internal workings of Node.js.

FIG-XI Internal Working of NODEJS

5.8 Structure of the Project



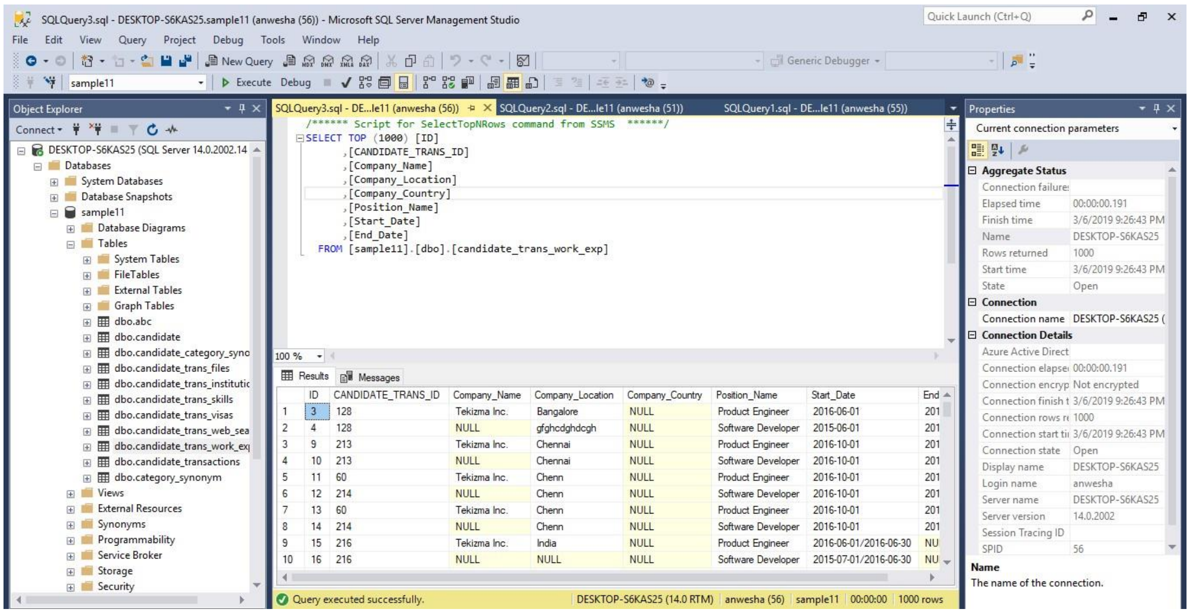
This PC > Local Disk (C:) > Anwesha > codebase > secure-trace > job-portal

Name	Date modified	Type	Size
.vscode	3/1/2019 5:45 PM	File folder	
node_modules	3/5/2019 3:35 PM	File folder	
app	3/5/2019 9:29 AM	JavaScript File	17 KB
application.properties	2/26/2019 9:54 PM	PROPERTIES File	1 KB
connection	2/20/2019 3:31 PM	JavaScript File	1 KB
package	3/5/2019 9:27 AM	JSON File	1 KB
package-lock	3/1/2019 5:38 PM	JSON File	37 KB
search.ejs	3/5/2019 1:35 PM	EJS File	2 KB
search	3/5/2019 12:15 PM	HTML File	1 KB
server	3/5/2019 6:51 PM	JavaScript File	3 KB

FIG-XII Structure Of The Project

5.9 Microsoft SqlStudio

SQL Server Management Studio (SSMS) is a software application first launched with Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools which work with objects and features of the server.


SQLQuery3.sql - DESKTOP-S6KAS25.sample11 (anwesha (56)) - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Debug Tools Window Help
New Query
Generic Debugger
sample11
Execute Debug
Object Explorer
Connect
DESKTOP-S6KAS25 (SQL Server 14.0.2002.14
Databases
System Databases
Database Snapshots
sample11
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.abc
dbo.candidate
dbo.candidate_category_syno
dbo.candidate_trans_files
dbo.candidate_trans_institutic
dbo.candidate_trans_skills
dbo.candidate_trans_visas
dbo.candidate_trans_web_sea
dbo.candidate_trans_work_ex
dbo.candidate_transactions
dbo.category_synonym
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
SQLQuery3.sql - DE...le11 (anwesha (56))
SQLQuery2.sql - DE...le11 (anwesha (51))
SQLQuery1.sql - DE...le11 (anwesha (55))
100 %
Results
Messages
Properties
Current connection parameters

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [ID]
      ,[CANDIDATE_TRANS_ID]
      ,[Company_Name]
      ,[Company_Location]
      ,[Company_Country]
      ,[Position_Name]
      ,[Start_Date]
      ,[End_Date]
  FROM [sample11].[dbo].[candidate_trans_work_exp]
```

	ID	CANDIDATE_TRANS_ID	Company_Name	Company_Location	Company_Country	Position_Name	Start_Date	End
1	3	128	Tekizma Inc.	Bangalore	NULL	Product Engineer	2016-06-01	201
2	4	128	NULL	gfghcdghdcgh	NULL	Software Developer	2015-06-01	201
3	9	213	Tekizma Inc.	Chennai	NULL	Product Engineer	2016-10-01	201
4	10	213	NULL	Chennai	NULL	Software Developer	2016-10-01	201
5	11	60	Tekizma Inc.	Chenn	NULL	Product Engineer	2016-10-01	201
6	12	214	NULL	Chenn	NULL	Software Developer	2016-10-01	201
7	13	60	Tekizma Inc.	Chenn	NULL	Product Engineer	2016-10-01	201
8	14	214	NULL	Chenn	NULL	Software Developer	2016-10-01	201
9	15	216	Tekizma Inc.	India	NULL	Product Engineer	2016-06-01/2016-06-30	NU
10	16	216	NULL	NULL	NULL	Software Developer	2015-07-01/2016-06-30	NU

Query executed successfully. | DESKTOP-S6KAS25 (14.0 RTM) | anwesha (56) | sample11 | 00:00:00 | 1000 rows

Aggregate Status	
Connection failure	
Elapsed time	00:00:00.191
Finish time	3/6/2019 9:26:43 PM
Name	DESKTOP-S6KAS25
Rows returned	1000
Start time	3/6/2019 9:26:43 PM
State	Open
Connection	
Connection name	DESKTOP-S6KAS25 (
Connection Details	
Azure Active Direct	
Connection elapse	00:00:00.191
Connection encryp	Not encrypted
Connection finish t	3/6/2019 9:26:43 PM
Connection rows re	1000
Connection start ti	3/6/2019 9:26:43 PM
Connection state	Open
Display name	DESKTOP-S6KAS25
Login name	anwesha
Server name	DESKTOP-S6KAS25
Server version	14.0.2002
Session Tracing ID	
SPID	56

Name
The name of the connection.

FIG-XIII Microsoft Sql Studio

5.10 Ajax

In order to give the SearchEngine a GUI, Website Using HTML, CSS, AJAX was developed. AJAX is the art of exchanging data with a server, and update parts of a web page without reloading the whole page. AJAX: It stands for asynchronous javascript XML It is a set of web technology and not any any language or framework It sends and receive data asynchronously. It do not interfere the current page, no requirement of reloading the web page.

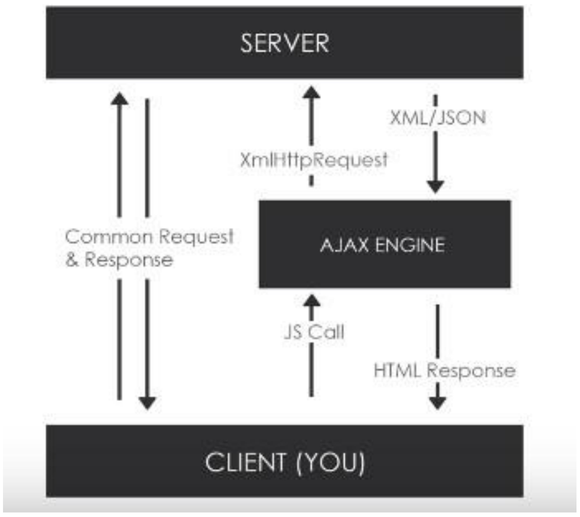


FIG-XIV AJAX

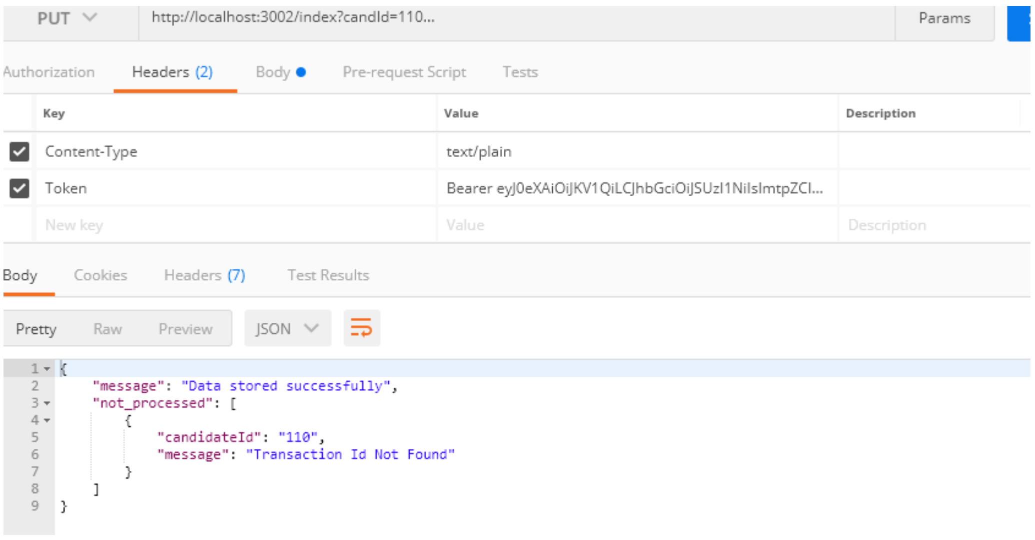
XmlHTTPRequest (XHR) Object: Through this API is sent in the form of an object(properties, methods). It is provided by the Browser's JS Environment. Methods transfer data between Ajax Engine Server.

Libraries other Methods- jQuery Axios SuperAgent Fetch API Prototype Node HTTP

5.11 Security & Authentication

Security & Authentication Bearer tokens are a type of access token. Authentication which uses bearer tokens is also known sometimes as application-only authentication or auth-only authentication. A bearer token is a byte array of unspecified format that you generate via a script like a curl command. Generated JSON WEB KEY USING HS-256 Algo
JSON WEB TOKEN: it has 3 parts: header, payload, signature
Header has 2 parts: Algorithm: SHA256, RSA, HMAC;
Type: JWT Signature ensures that the token wont changing while transferring from one body to other.
It provides more security. https://www.base64encode.org/
https://jwt.io/libraries

Further **DataValidation** was required for the security of the service which was provided in NodeJS through Cron Token bearer was created to be sent as header with each CRUD operation API. Validation of the Post requested was done through JOI.


PUT http://localhost:3002/index?candId=110... Params

Authorization | Headers (2) | Body | Pre-request Script | Tests

Key	Value	Description
✓ Content-Type	text/plain	
✓ Token	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI...	
New key	Value	Description

Body | Cookies | Headers (7) | Test Results

Pretty | Raw | Preview | JSON

```
{
    "message": "Data stored successfully",
    "not_processed": [
        {
            "candidateId": "110",
            "message": "Transaction Id Not Found"
        }
    ]
}
```

FIG-XV Security Through Token In The Header

5.12 Flow of the Project

Step 1: Node.js can generate dynamic page content. Node.js can create, open, read, write, delete, and close files on the server. Node.js can collect form data. Node.js can add, delete, modify data in your database.

Received by CRUD operations- Create, Read, Update Delete data from database. Created API for CRUD Operation.

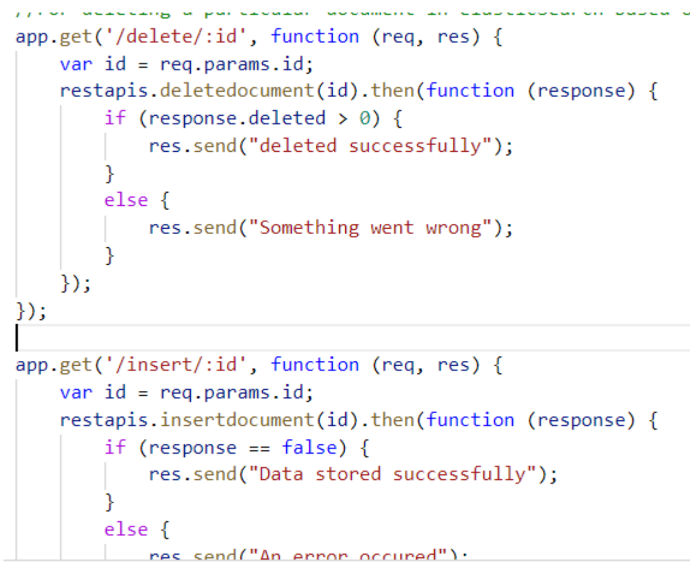
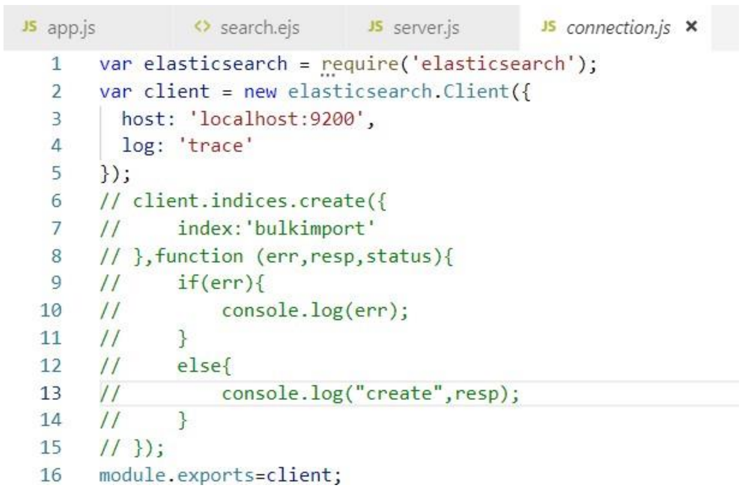

```
app.get('/delete/:id', function (req, res) {
    var id = req.params.id;
    restapis.deletedocument(id).then(function (response) {
        if (response.deleted > 0) {
            res.send("deleted successfully");
        }
        else {
            res.send("Something went wrong");
        }
    });
});

app.get('/insert/:id', function (req, res) {
    var id = req.params.id;
    restapis.insertdocument(id).then(function (response) {
        if (response == false) {
            res.send("Data stored successfully");
        }
        else {
            res.send("An error occured");
```

FIG-XVI API Creation

Step2: Calling of the CRUD operation function through their API creation. Ex: In figure restapis.insertdocument, it involves calling of insert function for create operation.

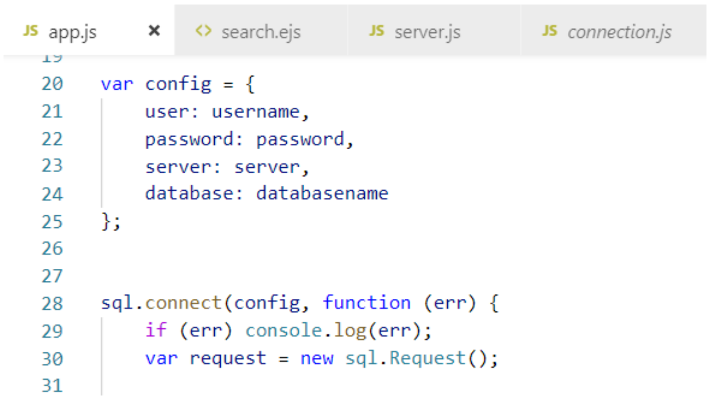
Step3: Setting up the connection between the localhost and elastic search.

Step4: Setting up the connection between localhost and database (Microsoft Sql Studio).



```
var elasticsearch = require('elasticsearch');
var client = new elasticsearch.Client({
  host: 'localhost:9200',
  log: 'trace'
});
// client.indices.create({
//     index:'bulkimport'
// },function (err,resp,status){
//     if(err){
//         console.log(err);
//     }
//     else{
//         console.log("create",resp);
//     }
// });
module.exports=client;
```

FIG-XVII Elastic Search Connection



```
var config = {
    user: username,
    password: password,
    server: server,
    database: databasename
};


sql.connect(config, function (err) {
    if (err) console.log(err);
    var request = new sql.Request();

```

FIG-XVIII Database Connection

Step5: Bulk indexing of data from localhost to elastic search, Importing of data from SqlStudio to Elastic Search through Bulk Import function. After cleaning of data and applying algorithm to buildName, buildAddress, buildSkill as in elastic search the data to be provided is in JSON form.

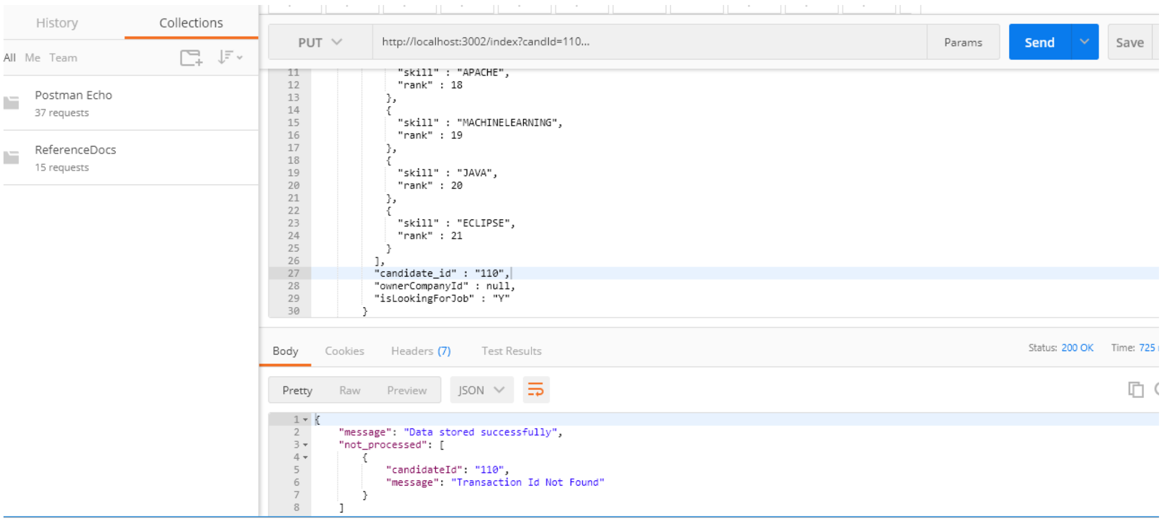
Step6: Perform elastic search, on the basis of 3 search operations required skills, preferred skills, minimum year of experience. Query based on bool, should, filter was applied. Pagination was done and scoring is done.

Step7: GUI was created to show the elaborated result.

Chapter 6: Use Cases In Secure Trace Phase-II

6.1 UseCase-I

Indexing the Candidate for Job Search:
Indexing could be done by CandidateId, TransactionId, JSON Body Passing

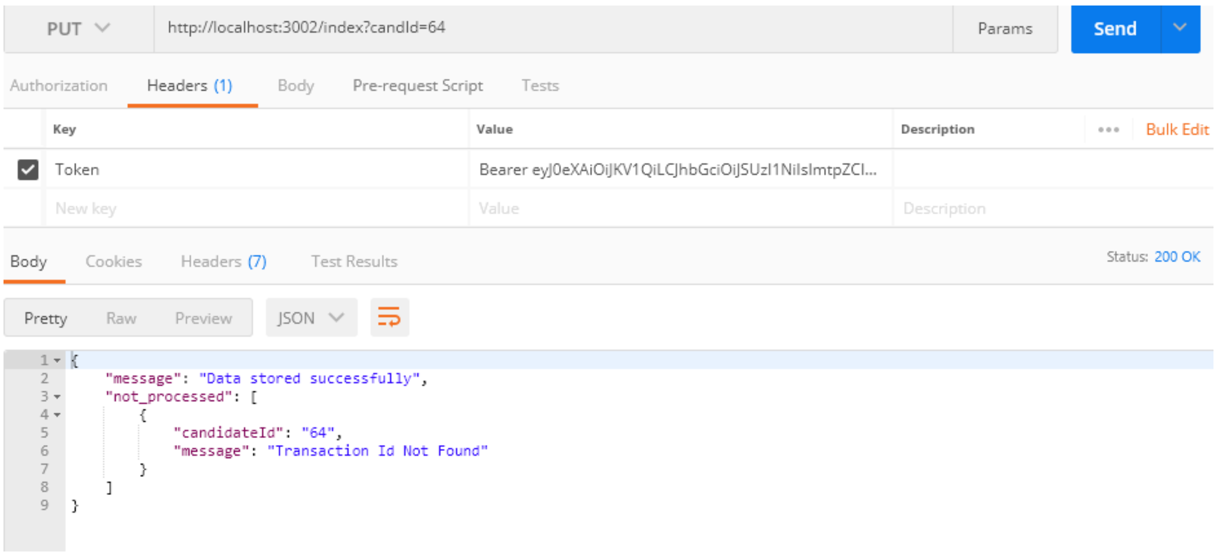


http://localhost:3002/index?candId=110...

```
          "skill" : "APACHE",
          "rank" : 18
        },
        {
          "skill" : "MACHINELEARNING",
          "rank" : 19
        },
        {
          "skill" : "JAVA",
          "rank" : 20
        },
        {
          "skill" : "ECLIPSE",
          "rank" : 21
        }
      ],
      "candidate_id" : "110",
      "ownerCompanyId" : null,
      "isLookingForJob" : "Y"
    }
```

```
{
    "message": "Data stored successfully",
    "not_processed": [
        {
            "candidateId": "110",
            "message": "Transaction Id Not Found"
        }
    ]
```

FIG-XIX UseCase-I (Hitting The API through JSON Body)

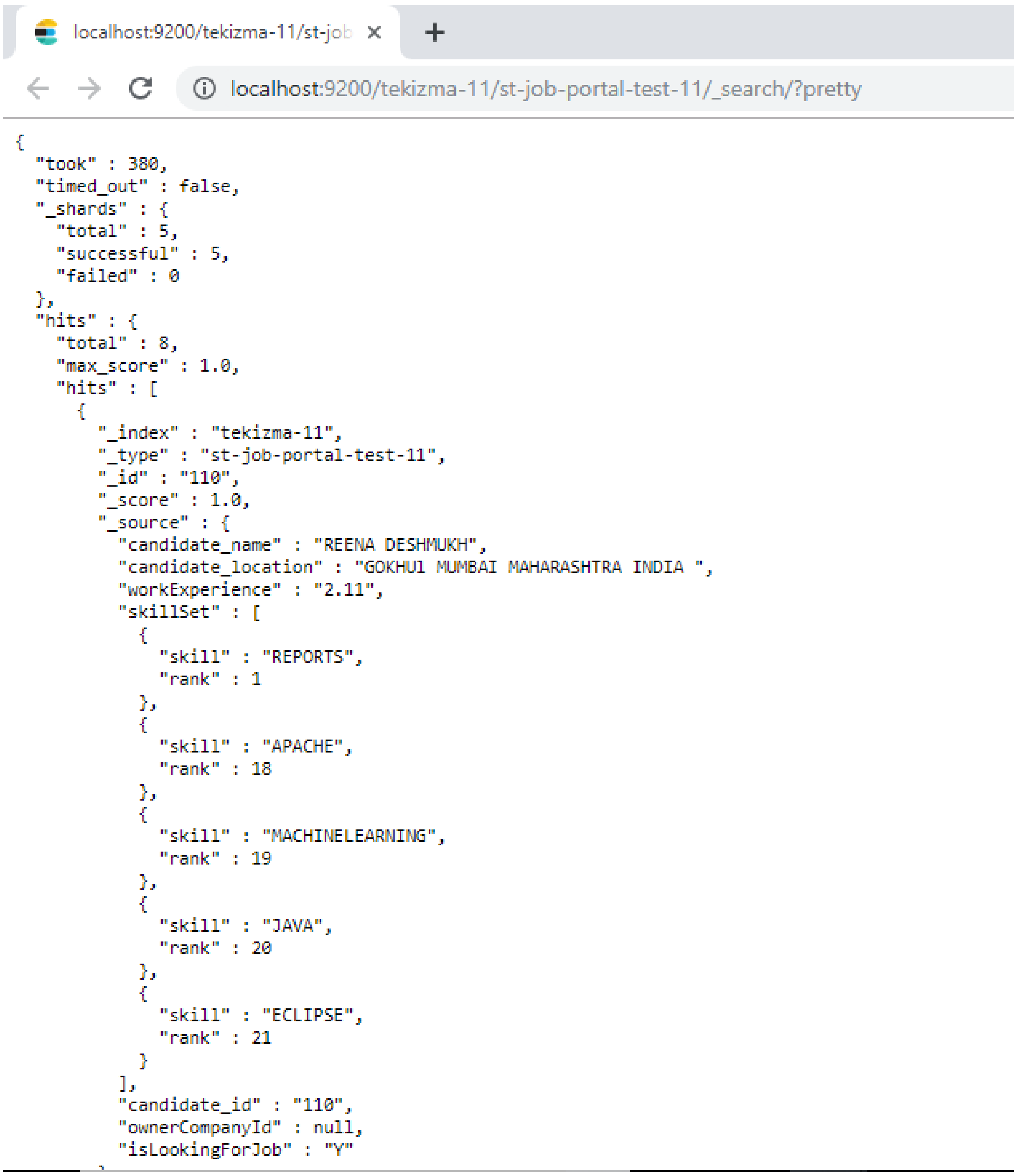

PUT
Params
Send
Authorization Headers (1) Body Pre-request Script Tests
Bulk Edit
Body Cookies Headers (7) Test Results
Status: 200 OK
Pretty Raw Preview JSON

http://localhost:3002/index?candId=64

Key	Value	Description
✓ Token	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI...	
New key	Value	Description

```
{
    "message": "Data stored successfully",
    "not_processed": [
        {
            "candidateId": "64",
            "message": "Transaction Id Not Found"
        }
    ]
}
```

FIG-XX UseCase-I (Hitting The API through CandidateId & No JSON Body)


localhost:9200/tekizma-11/st-job

localhost:9200/tekizma-11/st-job-portal-test-11/_search/?pretty

```
{
  "took" : 380,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "hits" : {
    "total" : 8,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "tekizma-11",
        "_type" : "st-job-portal-test-11",
        "_id" : "110",
        "_score" : 1.0,
        "_source" : {
          "candidate_name" : "REENA DESHMUKH",
          "candidate_location" : "GOKHU1 MUMBAI MAHARASHTRA INDIA ",
          "workExperience" : "2.11",
          "skillSet" : [
            {
              "skill" : "REPORTS",
              "rank" : 1
            },
            {
              "skill" : "APACHE",
              "rank" : 18
            },
            {
              "skill" : "MACHINELEARNING",
              "rank" : 19
            },
            {
              "skill" : "JAVA",
              "rank" : 20
            },
            {
              "skill" : "ECLIPSE",
              "rank" : 21
            }
          ],
          "candidate_id" : "110",
          "ownerCompanyId" : null,
          "isLookingForJob" : "Y"
```

FIG-XXI UseCase-I (Stored in Elastic Search)

Searching could be done on the indexed value on CompanyId, CandidateId, TransactionId, RequiredSkills, Preferred Skills if the candidate isLookingForJob is YES or TRUE

The search result further on the basis of overall rank appears first. The pagination is done through skill Rank. Elastic Search for searching takes into account must, should, filter queries in the match query. Required Skills are put in the must category, while preffered skills in the should category. Filtering is done on the basis of Work Experience and isLookingForJob. Pagination is done on the basis of the no. of Required, Preferred Skills Matched to get the overall Rank. WorkExperience is calculated from the start date of the First Job and the endDate of the Current Job. EndDate could be in the form of TillDate, Present and the date was of different format which was bring out to one format by MOMENT.JS.

SEARCH QUERY CASE-I:

```
{
 "from":0,
 "size":1000,
    "query": {
      "bool": {
        "must": [
          {
            "bool": {
              "should": [
                {
                  "nested": {
                    "path": "skillSet",
                    "query": {
                      "bool": {
                        "must": [
                          {
                            "match": {
                              "skillSet.skill": "ANGULAR JS"
                            }
                          }
                        ]
                      }
                    }
                  }
                },
                {
                  "nested": {
                    "path": "skillSet",
                    "query": {
                      "bool": {
                        "must": [
                          {
                            "match": {
                              "skillSet.skill": "HTML"
                            }
                          }
                        ]
                      }
                    }
                  }
                },
                {
                  "nested": {
                    "path": "skillSet",
                    "query": {
                      "bool": {
```

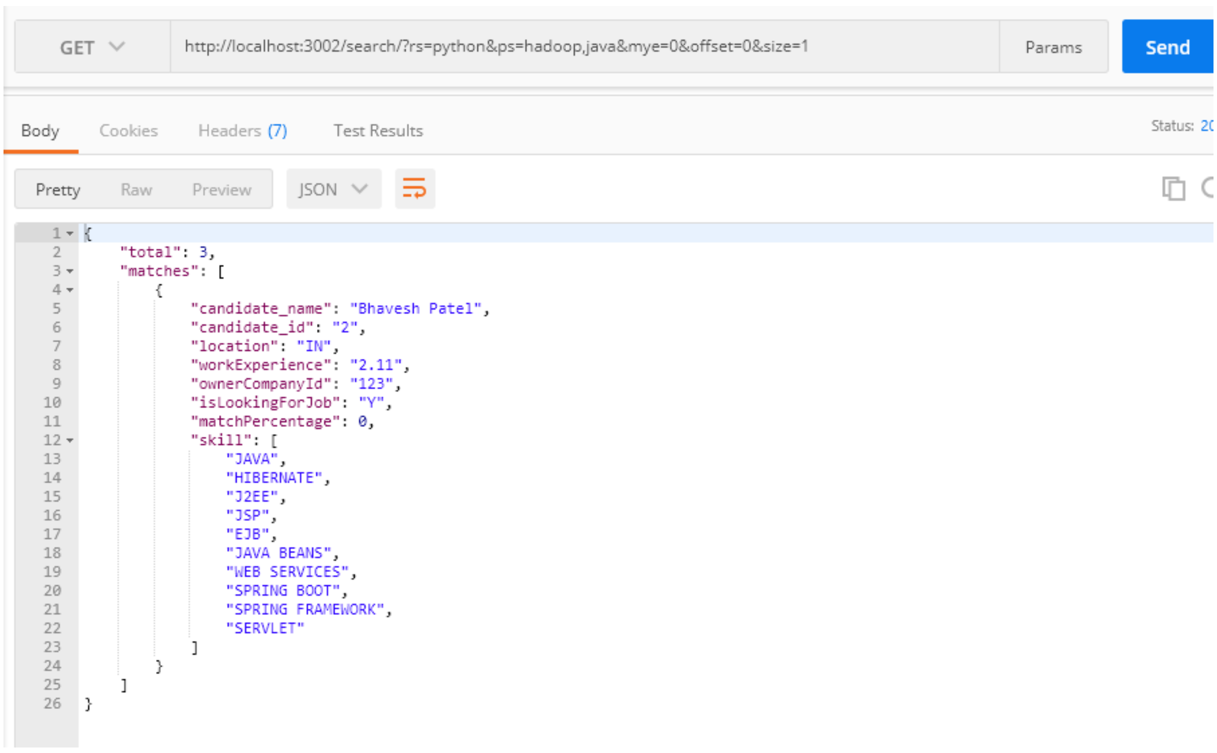
```
                    "must": [
                      {
                        "match": {
                          "skillSet.skill": "JIRA"
                        }
                      }
                    ]
                  }
                }
              }
            }
          ]
        }
      }
    ],
    "filter": [
      {
        "range": {
          "workExperience": {
            "gte": "0"
          }
        }
      },
     {"match":{"candidate_id":"2"}},
     {"match":{"isLookingForJob":"Y"}}

    ]
  }
}
```

Search Query-I CASE-II

```
{
 "from":0,
 "size":1000,
   "query": {
     "bool": {
       "must": [
         {
           "bool": {
             "should": [
               {
                 "nested": {
                   "path": "skillSet",
                   "query": {
                     "bool": {
                       "must": [
                         {
                           "match": {
                             "skillSet.skill": "tableau"
```

```
                    }
                  }
                ]
              }
            }
          }
        },
        {
          "nested": {
            "path": "skillSet",
            "query": {
              "bool": {
                "must": [
                  {
                    "match": {
                      "skillSet.skill": "stat"
                    }
                  }
                ]
              }
            }
          }
        },
        {
          "nested": {
            "path": "skillSet",
            "query": {
              "bool": {
                "must": [
                  {
                    "match": {
                      "skillSet.skill": "Reports"
                    }
                  }
                ]
              }
            }
          }
        }
      ]
    }
  }
],
"filter": [
  {
    "range": {
      "workExperience": {
        "gte": "0"
      }
    }
```

```
          },
          {
          "bool":{
          "should":[
                    {
                     "match": {
                       "candidate_id": "181"
                     }
                     },
                     {
                     "match": {
                       "candidate_id": "158"
                     }
                     }
                     ]
                     }
                     }
        ]
      }
    }
  }
```



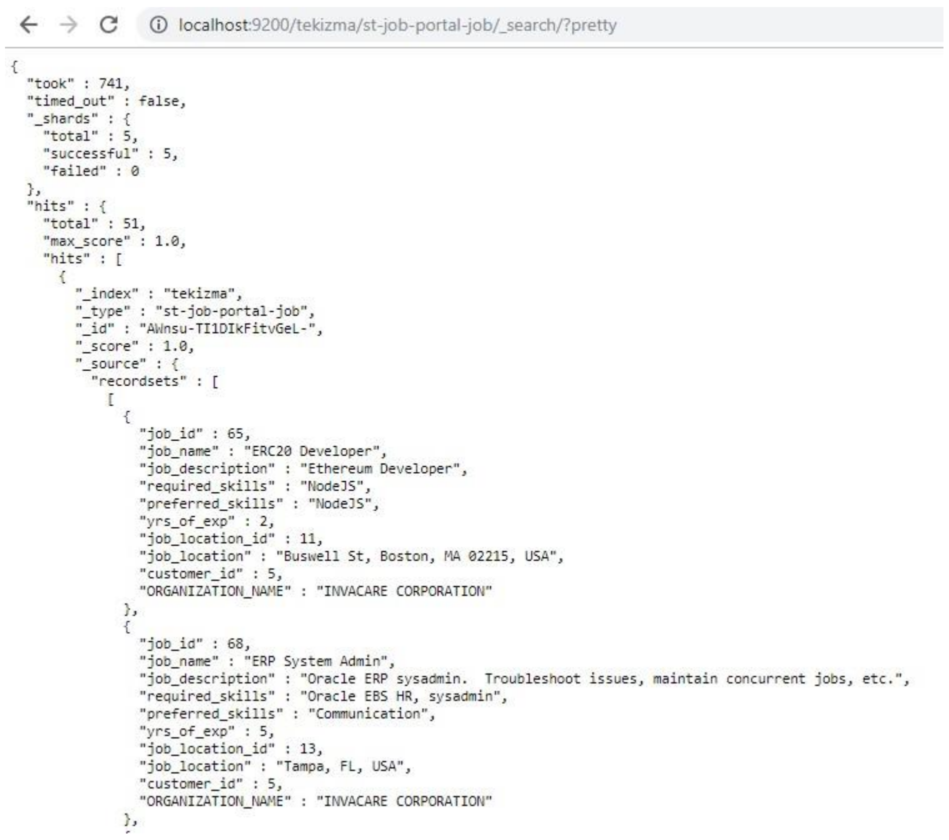
```
http://localhost:3002/search/?rs=python&ps=hadoop,java&mye=0&offset=0&size=1
```

```
{
    "total": 3,
    "matches": [
        {
            "candidate_name": "Bhavesh Patel",
            "candidate_id": "2",
            "location": "IN",
            "workExperience": "2.11",
            "ownerCompanyId": "123",
            "isLookingForJob": "Y",
            "matchPercentage": 0,
            "skill": [
                "JAVA",
                "HIBERNATE",
                "J2EE",
                "JSP",
                "EJB",
                "JAVA BEANS",
                "WEB SERVICES",
                "SPRING BOOT",
                "SPRING FRAMEWORK",
                "SERVLET"
            ]
        }
    ]
}
```

FIG-XXII SearchQuery Result

6.2 UseCase-II

In Use Case-I, candidates where indexed for the companies to search the candidates but here the jobs are indexed for the candidates to search the job. The indexed value included jobId, jobname, requiredSkills, preferredSkills, years of Experience, jobLocation, OrganisationName, customerId, jobDescription. The searching could be done on the basis of jobDescription, requiredSkills, preferredSkills, jobId, yearsofExp.



```
{
  "took" : 741,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "hits" : {
    "total" : 51,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "tekizma",
        "_type" : "st-job-portal-job",
        "_id" : "AWnsu-TI1DIkFitvGeL-",
        "_score" : 1.0,
        "_source" : {
          "recordsets" : [
            [
              {
                "job_id" : 65,
                "job_name" : "ERC20 Developer",
                "job_description" : "Ethereum Developer",
                "required_skills" : "NodeJS",
                "preferred_skills" : "NodeJS",
                "yrs_of_exp" : 2,
                "job_location_id" : 11,
                "job_location" : "Buswell St, Boston, MA 02215, USA",
                "customer_id" : 5,
                "ORGANIZATION_NAME" : "INVACARE CORPORATION"
              },
              {
                "job_id" : 68,
                "job_name" : "ERP System Admin",
                "job_description" : "Oracle ERP sysadmin.  Troubleshoot issues, maintain concurrent jobs, etc.",
                "required_skills" : "Oracle EBS HR, sysadmin",
                "preferred_skills" : "Communication",
                "yrs_of_exp" : 5,
                "job_location_id" : 13,
                "job_location" : "Tampa, FL, USA",
                "customer_id" : 5,
                "ORGANIZATION_NAME" : "INVACARE CORPORATION"
              },
```

FIG-XXIII UseCase-II (Stored in Elastic Search)

SEARCH QUERY FOR USE CASE-II

```
{
  "query": {
   "bool": {
    "should": [
     {
      "match": {
       "required_skills": "Javascript"
      }
     },
     {
      "match": {
       "preferred_skills": "Html"
      }
```

```
          },
          {
            "match": {
              "job_name": "Java"
            }
          }
        ],
        "filter":[
        {
            "range": {
              "yrs_of_exp": {
                "gte": "4"
              }
            }
          },
           {
            "match" : {
              "job_location": "Bengaluru, Karnataka, India"
          }

          }
        ]
      }
    }
  }
```

6.3 Exact Match

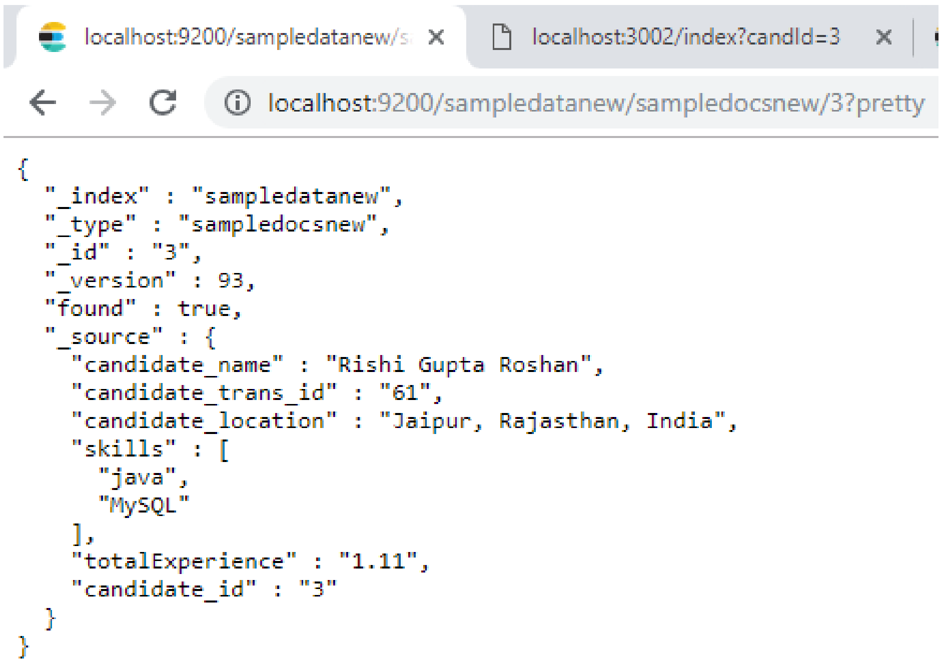
On the indexed values of UseCase-I & UseCase-II, if the searching is done on the exact values and not for therange of the values to come. It comes under the UseCase-III of the exact match.

Chapter:7 SUMMARY & Prsentable Output

7.1 Challenges

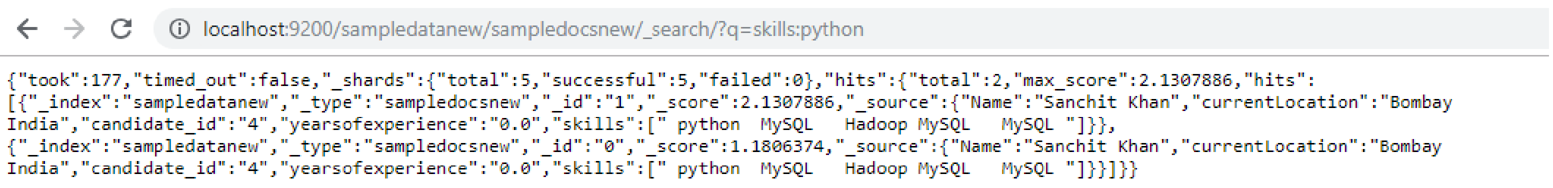
- Skills could be misspelt in the query due to typo error or fuzziness.
- Abbreviations or keywords might be used for the query.
- Continuous changes according to the client requirement in a live project,
- Local and production level deployment issues,
- Optimisation of algorithms
- Coding standard of the company.
- Meeting the deadlines.

7.2 Output


localhost:9200/sampledatanew/s
localhost:3002/index?candId=3
localhost:9200/sampledatanew/sampledocsnew/3?pretty

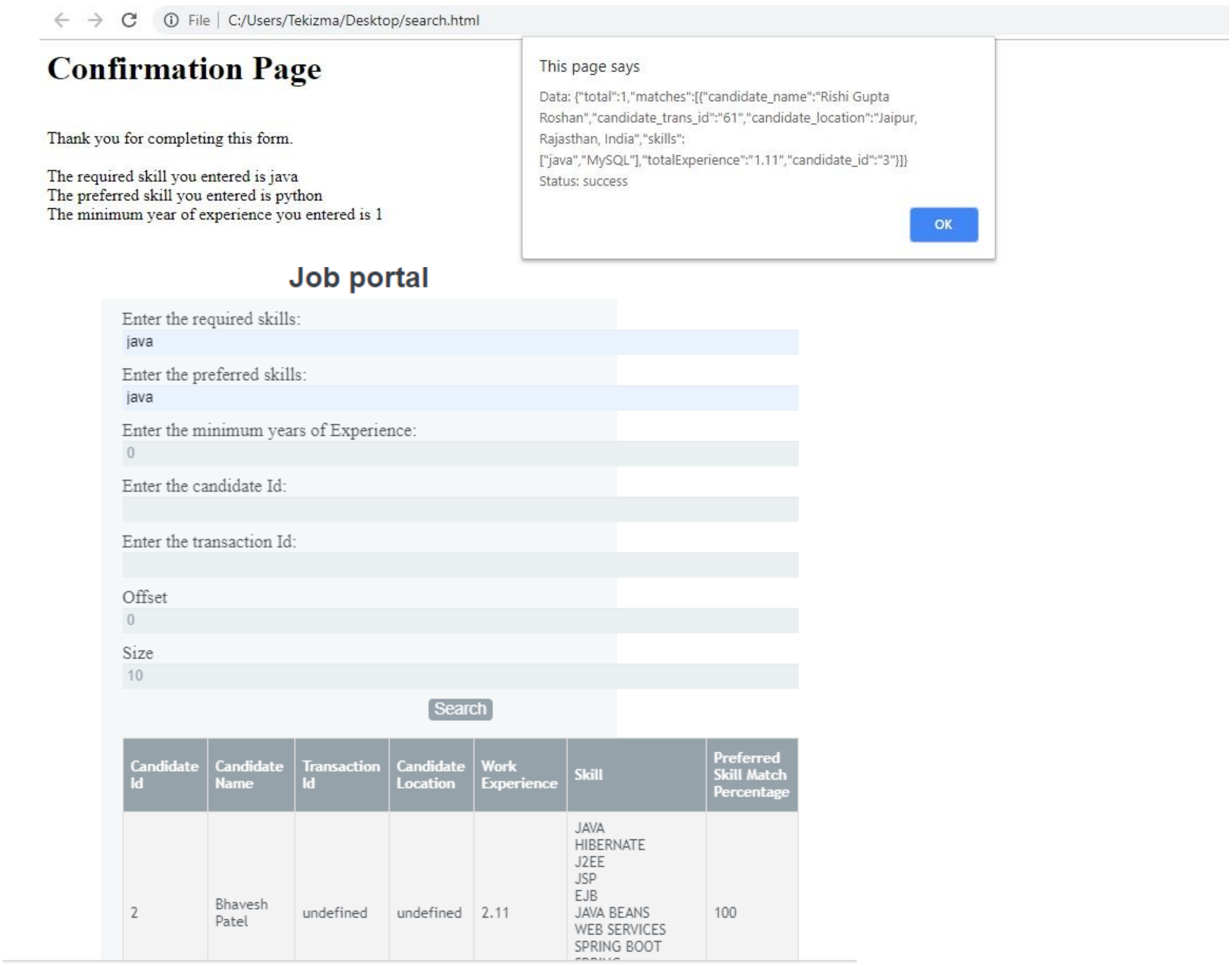
```
{
  "_index" : "sampledatanew",
  "_type" : "sampledocsnew",
  "_id" : "3",
  "_version" : 93,
  "found" : true,
  "_source" : {
    "candidate_name" : "Rishi Gupta Roshan",
    "candidate_trans_id" : "61",
    "candidate_location" : "Jaipur, Rajasthan, India",
    "skills" : [
      "java",
      "MySQL"
    ],
    "totalExperience" : "1.11",
    "candidate_id" : "3"
  }
}
```

FIG-XXIV Output In Elastic Search After Indexing


localhost:9200/sampledatanew/sampledocsnew/_search/?q=skills:python

```
{"took":177,"timed_out":false,"_shards":{"total":5,"successful":5,"failed":0},"hits":{"total":2,"max_score":2.1307886,"hits":
[{"_index":"sampledatanew","_type":"sampledocsnew","_id":"1","_score":2.1307886,"_source":{"Name":"Sanchit Khan","currentLocation":"Bombay
India","candidate_id":"4","yearsofexperience":"0.0","skills":[" python  MySQL   Hadoop MySQL   MySQL "]}},
{"_index":"sampledatanew","_type":"sampledocsnew","_id":"0","_score":1.1806374,"_source":{"Name":"Sanchit Khan","currentLocation":"Bombay
India","candidate_id":"4","yearsofexperience":"0.0","skills":[" python  MySQL   Hadoop MySQL   MySQL "]}}]}}
```

FIG-XXV Output In Elastic Search



Confirmation Page

Thank you for completing this form.

The required skill you entered is java
The preferred skill you entered is python
The minimum year of experience you entered is 1

This page says

Data: {"total":1,"matches":[{"candidate_name":"Rishi Gupta Roshan","candidate_trans_id":"61","candidate_location":"Jaipur, Rajasthan, India","skills":["java","MySQL"],"totalExperience":"1.11","candidate_id":"3"}]}
Status: success

Job portal

Enter the required skills:	java
Enter the preferred skills:	java
Enter the minimum years of Experience:	0
Enter the candidate Id:	
Enter the transaction Id:	
Offset	0
Size	10

Candidate Id	Candidate Name	Transaction Id	Candidate Location	Work Experience	Skill	Preferred Skill Match Percentage
2	Bhavesh Patel	undefined	undefined	2.11	JAVA HIBERNATE J2EE JSP EJB JAVA BEANS WEB SERVICES SPRING BOOT	100

FIG-XXVI Output Through GUI

7.3 Learnings

I got engaged with many technologies.

- Elastic Search
- NodeJS

  Url passing through Ajax in a dynamic way that connects the nodeJs application in a dynamic way.
- SpringBoot
- REST API
- Optimisation of Algorithms

  Continuous changes according to the client requirement in a live project & meeting the deadline.

Moreover, working in a live project is itself a giant opportunity to learn coding in a dynamic way so that when the client gives improvement only a small change is needed instead of changing the whole code.Coding in dynamic fashion is the beauty of live project.

7.4 Real World Elastic Search Use Cases

Here are some examples of real-world Elasticsearch use cases from the official Elastic website.

Wikipedia uses Elasticsearch to provide full-text search with highlighted search snippets, and search-as-you-type and did-you-mean suggestions.

The Guardian uses Elasticsearch to combine visitor logs with social -network data to provide real-time feedback to its editors about the public's response to new articles.

Stack Overflow combines full-text search with geolocation queries and uses more-like-this to find related questions and answers.

GitHub uses Elasticsearch to query 130 billion lines of code.

7.5 Summary

The work done in this project is on NLP Libraries GoogleDialogFlow, AmazonLex, Stanford NLP, Microsoft Luis. The algorithms are written on Elastic Search Indexing & Searching based on Client's requirement. To bring out the algorithm of Elastic search, its coded in NodeJs. Security & Authentication is done by passing token in the header. The Crud operations Are performed well to bring out the work. SpellChecker & Word Splitter Modules are built In Phase-I in Python to overcome the challenge in Phase-I. The python code was further integrated with Spring Maven Project through Jython. The prime importance is done on bringing out the correct or desired output irrespective of variety of Technologies used in the project. Besides Algorithm, Coding Standards and Optimisation of code is also put into practice. The work also includes the use of Loggers for production environment. Log.info, Log.error, Log.debug are put into practice. The code is written in a systematic pattern for its reusability In future.

Chapter:8 REFERENCES

1. Paper on Searching & Indexing Using Elastic Search, IJECS, Darshita Kalyani, Dr. Devarshi Mehta. Gujarat, India.
2. A Conceptual Review of Elastic Search – Survey Paper Subhani shaik1 , Nalamothu Naga Malleswara Rao2, International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887 Volume 5 Issue XI November 2017
3. Research Document Search using Elastic Search, R. Vidhya1* and G. Vadivu2, Indian Journal of Science and Technology, Vol 9(37), DOI: 10.17485/ijst/2016/v9i37/102108, September 2016
4. Use of Elastic Search for Intelligent Algorithms to Ease the Healthcare Industry, C. Bhadane, H. A. Mody, D. U. Shah, P. R. Sheth, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-6, January 2014
5. A Comparative Analysis of Node.js (Server-Side JavaScript), St. Cloud State University.
6. Spring Framework: A Companion to JavaEE Ankur Bawiskar1 , Prashant Sawant2 , Vinayak Kankate3 , B.B. Meshram4, IJCEM International Journal of Computational Engineering & Management, Vol. 15 Issue 3, May 2012 ISSN (Online): 2230-7893
7. Documentation of Microsoft Luis, Google Dialogue Flow, Amazon Lex, Stanford NLP, ElasticSearch, NodeJS, Spring Maven, MSSQLStudio
8. Building a Full Text Search Using Docker & Elastic Search, blog.patricktriest