Project Progress Report

CS 410 – Text Information Systems

N. Wood (nadiaw2)

# Search engine for indoor environment data using ElasticSearch

NetID: nadiaw2. I will be working on this project individually.

Abstract:

In the age of sensors, devices and platforms collecting millions of datapoints every second, it comes necessary to be able to sift through all the data to develop insights efficiently Big data offers the solution for analyzing large amount of data and using the technique of Elasticsearch, access to data can be made faster.[[1]](#footnote-1)

I will be creating a web application to use ElasticSearch to search content from a set of documents of environmental datapoints collected by sensors (indoor air, humidity, temp etc). Currently, it is difficult to search for data in a RDMS database and it takes significant time using traditional SQL queries. The project will take the data transfer it to ElasticSearch server. The front end written in React will allow users to search for data. Measurable outcomes are going to be the amount of time it takes to run a query against a traditional RDMS database vs. using ElasticSearch. The planned architecture is shown in Figure1:

Diagram

Description automatically generated

Figure

The estimated time to complete this project is about 30-40 hours.

Tasks and time:

|  |  |  |
| --- | --- | --- |
| Task | Hours | Status |
| 1. Determine how to set up, configure and deploy ElasticSearch on a cloud platform :    1. The cloud platform determination will base on the cost difference between Azure, **Google** and AWS. | 15 hours | **Completed.**  Created an account on Google and hosted ElasticSearch as a hosted service on ElasticCloud. |
| 1. Migration of raw to json data to Elastic search server : | 3 hours | **Completed.**  Created a data ingestion pipeline to read in a file which contains json per line and convert the file into an array of jsons. Then uploaded the data to ElasticSearch Cloud. |
| 1. Development and deployment of Web application: | 10 hours | **Started.**  **8 hours’ worth of work left.**  Creating a CRUD application in REACT using the API to communicate with the ElasticSearch Engine. |
| 1. Test and measure outcomes: | 4 hours | Not started. |

## Issues:

Originally the plan was to develop a C#.NET application, but further research supported developing a front end using Node.js or React. I will be creating A CRUD (Create, read, update, delete) client using React.

## Setting up Elastic Search

Create a project on Google Cloud.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Data Ingestion to the engine

JSON used

{

"eventtimestamp": "2020-11-20T01:54:56.279Z",

"dataconnection": "humidity.monnit.humidity",

"datatype": "Relative Humidity",

"dataunits": "Percent RH",

"datavalue": "36.09",

"coord\_x": 0,

"coord\_y": 0,

"coord\_z": 0,

"coord\_phi": 0,

"coord\_theta": 0,

"clope": 0.9091,

"intercept": 9.8215,

"quadratic\_a": 0,

"quadratic\_b": 0,

"quadratic\_c": 0,

"batterycurrentvoltage": 3.42,

"batterymaximumvoltage": 3.4,

"batteryminimumvoltage": 2.7,

"datavaluecalibratedsi": 28.895061049389508,

"datavaluecalibratedq": 0

}

JSON field names have to be all lowercase or be separated by underscore

## Data Ingestion

Created a simple program in C# to convert existing files to a proper json file so that it can be imported into ElasticSearch

Create a web application

Using Visual Studio, create a web application

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, text, application, email, website

Description automatically generated

## Installing ElasticSearch.Net Client

Install the nuget package within the application

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

**Install npm packages to create a react app**

npm i -g [create-react-app@1.5.2](mailto:create-react-app@1.5.2)

**A picture containing text, indoor, green, screenshot

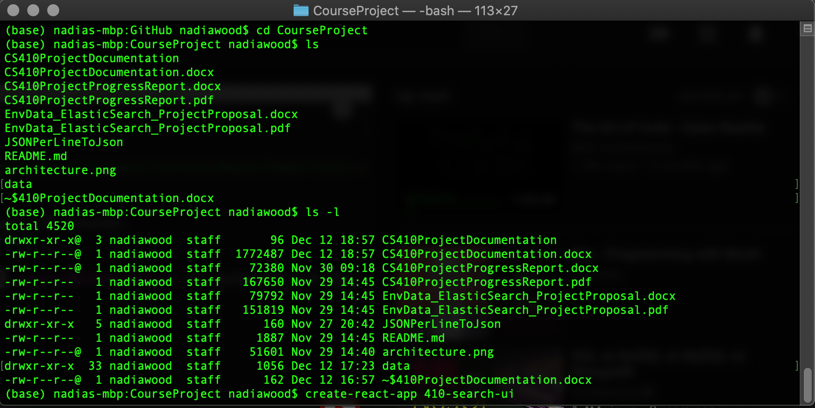
Description automatically generated**

**Setting up React Environment**

Visual Studio Code

Extensions – Simple React Snippets

Prettier Code formatter



**UI for search experience**

Install **React Search UI** and the **App Search** connector.

# Install React Search UI and a Connector, like the Elastic App Search Connector

npm install --save @elastic/react-search-ui @elastic/search-ui-app-search-connector

## Creating a search experience

1. Gujarat, India, Darshita Kalyani, and Dr. Devarshi Mehta, “Paper on Searching and Indexing Using Elasticsearch,” *International Journal Of Engineering And Computer Science*, June 30, 2017, https://doi.org/10.18535/ijecs/v6i6.45. [↑](#footnote-ref-1)