ENCS 6921 - Industrial Stage and Training

Winter 2023

March Monthly Report

Claim Explanation

Submitted By:
Manan Dineshkumar Paruthi - 40192620

Submitted To:
Dr. Rajagopalan Jayakumar
Director, Co-op Program, Associate Professor, Computer Science and Software Engineering

Introduction & Problem Statement

Before starting my Winter 2023 Coop, I had a discussion with my manager about the requirements of this course i.e ENCS 6921 Industrial Stage and Training and what can be done to aim to work above and beyond the scope of my internship.

So, he informed me that application logging is very important and critical in the industry, for Identifying and debugging issues in the production environment. As it contains sensitive and personal customer data, live debugging can not be done. Also, analysis of logs data helps us to gain information which is important to improve the quality of the applications.

Currently, the company is using Splunk and they are willing to migrate it to ELK Stack (Elastic Search, Logstash & Kibana) as it is open source so free of cost while Splunk is costly and requires a license.

But the issue with ELK Stack is that it is more difficult to set up and maintain. So, it would be great if I could make a POC (Proof Of Concepts) on the same which can be demoed to the senior management and based on that we can start the migration to ELK Stack.

The main reasons for migration are :

1) Cost: ELK Stack is open-source, which means it is free to use, whereas Splunk is proprietary software and requires a license, it makes ELK a more cost-effective option for organisations and helps them to reduce the budget expenditure in this economic crisis and layoff environment.
2) Flexibility: ELK Stack is highly customizable and can be tailored to specific needs, whereas Splunk's capabilities are more limited. With ELK, organizations can choose to use different components and add-ons, and can also create custom plugins to extend their functionality
3) Data Ownership: With ELK Stack, organizations have more control over their data and can decide where it is stored and how it is processed, whereas Splunk stores data on its own servers.

Solution Approach

This project is a web application that focuses on log management and data analysis. The application allows users to create accounts, log in to their accounts, and log out of their accounts. For every user action, events are logged, and log data is generated. This log data is then parsed and used to analyze and visualize trends and metrics using an informative dashboard.

Log management technologies enable businesses to extract important information from unstructured data such as response times, customer plan dollar values, and performance resource usage information. These technologies roll this information up into metrics dashboards

that give businesses insights into a wide range of trends that are happening across their systems and operations.

One of the advantages of log data is that it always maintains evidence of what happened. For example, if there is a sudden spike in the number of signups, businesses can quickly validate the change by checking the logs to see who signed up and when. Traditional metrics dashboards do not always maintain the source of data used to calculate the metrics, so validating sudden changes often requires a discussion with engineers, which can take days to complete.

By using this web application, businesses can gain insights into their operations by analyzing log data. The dashboard provides a visual representation of the data, making it easier for businesses to identify trends and patterns. The application makes it easy to log user actions and analyze the data, providing valuable insights for businesses to improve their operations and performance.

In summary, this project focuses on log management and data analysis, allowing businesses to gain insights from their log data. The application provides a user-friendly dashboard for visualizing the data, making it easier for businesses to identify trends and patterns. By using this application, businesses can make data-driven decisions to improve their operations and performance.

It is a web application that utilizes the MEAN stack for front-end and back-end development and the ELK stack for log management and data analysis.

The MEAN stack consists of four technologies: MongoDB, ExpressJS, AngularJS, and NodeJS. MongoDB is a NoSQL document-oriented database used to store and manage data. ExpressJS is a web application framework used to build APIs and web applications. AngularJS is a client-side framework used to build dynamic web applications. NodeJS is a server-side JavaScript runtime used to build scalable and efficient server-side applications.

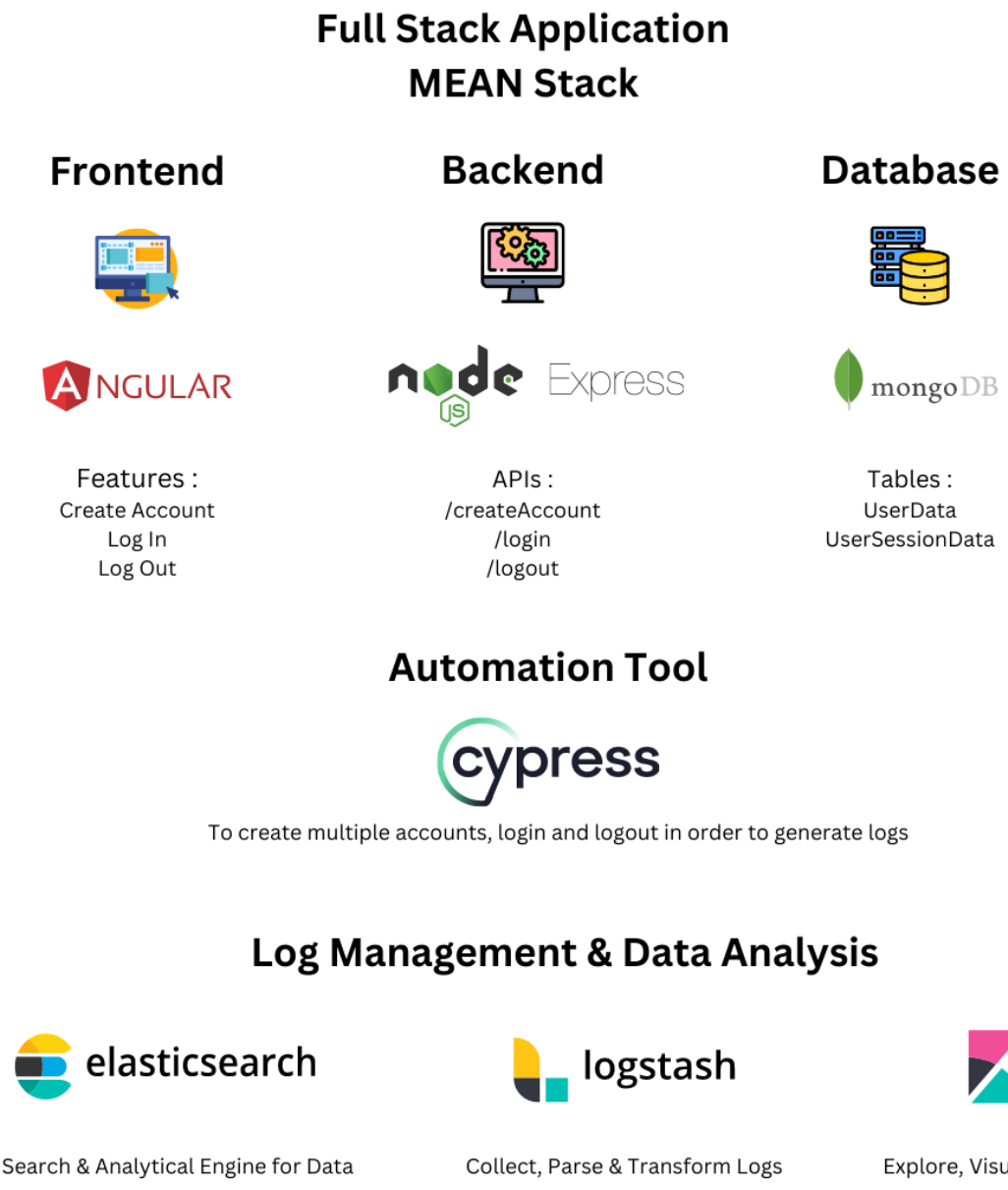
The front end of the application is built using Angular, a client-side framework that provides a seamless user experience. Angular makes it easy to build dynamic and interactive web applications. The back end of the application is built using NodeJS, ExpressJS, and MongoDB. NodeJS provides a scalable and efficient runtime environment for server-side JavaScript applications, while ExpressJS provides a robust framework for building APIs and web applications. MongoDB is used to store and manage data in a document-oriented format, making it easy to scale and maintain.

The Winston Logging tool is used for logging and saving application events. Winston is a JavaScript logging library that allows developers to log messages to various destinations such as the console, a file, or a database. By logging events, developers can analyze and assess threats and analyze errors before they disrupt broader business workflows.

Cypress is a front-end testing tool used to perform different actions that generate logs. These logs can be used for data analysis and visualization. Cypress is a JavaScript-based testing framework that allows developers to write and run automated tests for web applications.

The ELK stack is used to collect, store, and analyze log data from the application. The ELK stack consists of three technologies: Elasticsearch, Logstash, and Kibana. Logstash is used to collect log data from various sources, Elasticsearch is used to store the log data, and Kibana is used to provide visualizations and analytics. The ELK stack makes it easy to collect and analyze log data, providing insights into the performance and behaviour of the application.

Diagram



Implementation Details

To set up Elasticsearch on a local system, follow these steps:

1) Open the elasticsearch.yml file in the settings folder. To enable automatic index construction, add the following line: "action.auto create index:.monitoring*,.watches,.triggered watches,.watcher-history*,.ml*". Save and exit the file.

2) Go to the bin folder and execute "elasticsearch.bat." This will produce a Kibana password and enrollment token. This information should be copied and saved for future use.

3) Return to the configuration folder and re-open the elasticsearch.yml file. Under the "xpack.security.http.ssl" and "xpack.security.transport.ssl" sections, change the "enabled" option to false. Save and exit the file.

4) Go back to the bin folder and execute "elasticsearch.bat."

5) Check "http://localhost:9200/". You will be required to enter your login information. Use the following details: The username is elastic, and the password is the elastic password you copied before. To acquire a new password, navigate to the bin folder and run the command "elasticsearch-reset-password -u elastic."

6) Next, navigate to the bin folder and type "elasticsearch-reset-password -u kibana system" to copy and store the Kibana password.

To set up Kibana on a local system, follow these steps:

1) Open the config folder and find the kibana.yml configuration file. Uncomment the server.port, server.host, elasticsearch.host, elasticsearch.username, and elasticsearch.password settings. Replace the password with the Kibana password you already stored. Save and exit the file.

2) Go to the bin folder and run the "kibana.bat" programme.

3) Go to "http://localhost:5601." You will be required to enter your login information. Use the following information: The username is "elastic," and the password is the elastic password you previously copied.

To set up Logstash on a local system, follow these steps:

1) Go to the config folder and double-click the learn.conf file to open it. Replace the elastic password with the password you already stored.

2) Go to the bin folder and run the command "logstash -f.configlearn.conf —config.reload.automatic". The supplied configuration file will be used to start Logstash, and

the —config.reload.automatic flag will automatically reload the configuration file if changes are detected.

To configure a Kibana dashboard, please follow these steps:

1) Go to Stack Management, then Index Management on the Elasticsearch site. Copy the name of the file "logstsh index logdata-date>" with today's date.

2) After that, navigate to Stack Management and then Data Views. Choose "Create Data View." Enter "logstsh index logdata-date>" in the index pattern field and give the data view a name. "Save Data View to Kibana" should be selected.

3) Lastly, navigate to the Dashboard area and add a new dashboard. Click "Create Visualization" and choose the data view name you created in the previous step from the selection menu on the upper left. You may then construct data visualisations and upload them to the dashboard.

Grok Filter Implementation

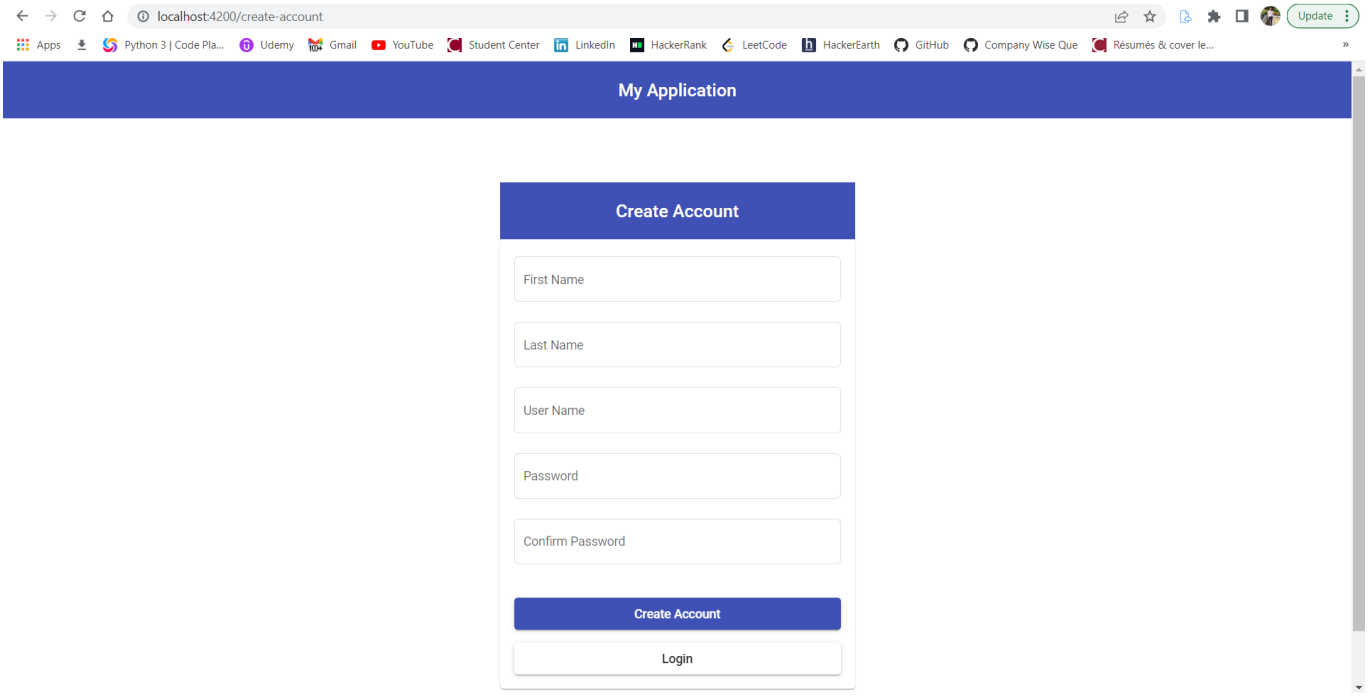
The Grok filter in Logstash provides a means to convert unstructured log data into structured data that can be readily indexed and searched in Elasticsearch. Grok identifies patterns in log data with regular expressions and then assigns those patterns to fields.

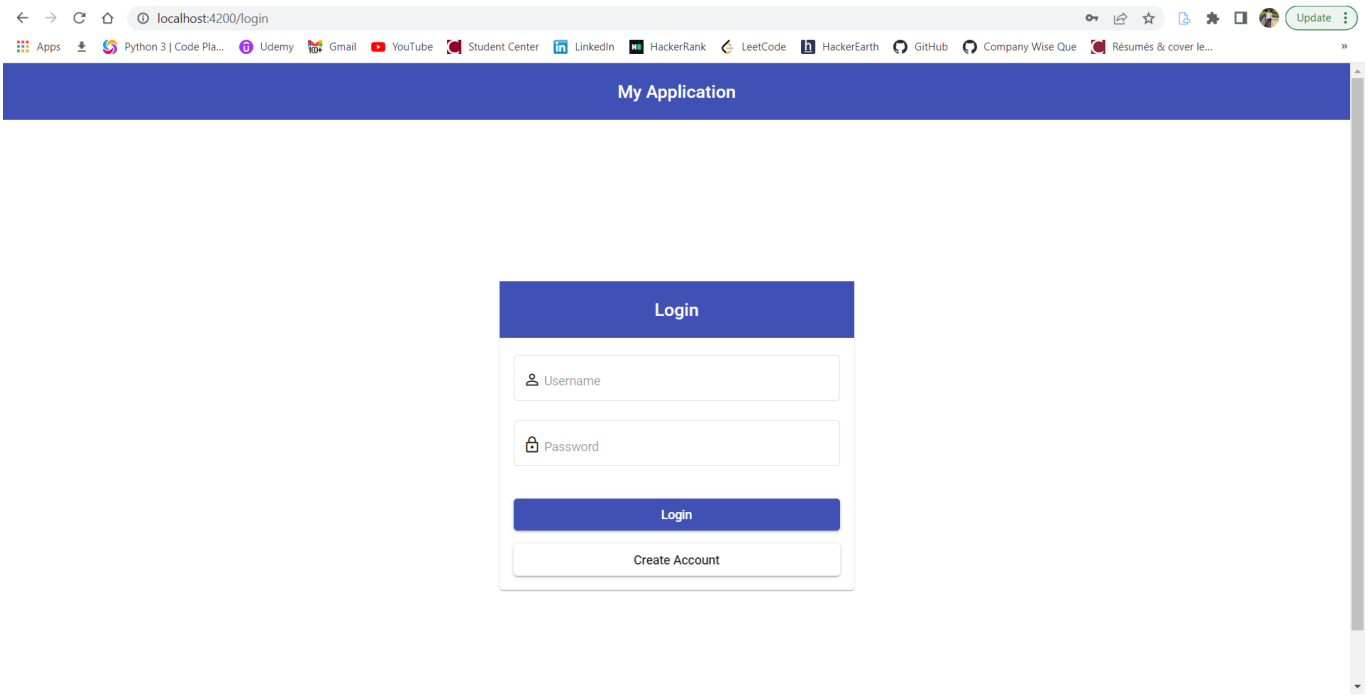
The Grok filter extracts fields from several forms of log data, including Apache logs, syslog, and NGINX logs. It may also parse bespoke log formats by specifying a pattern that fits the format.

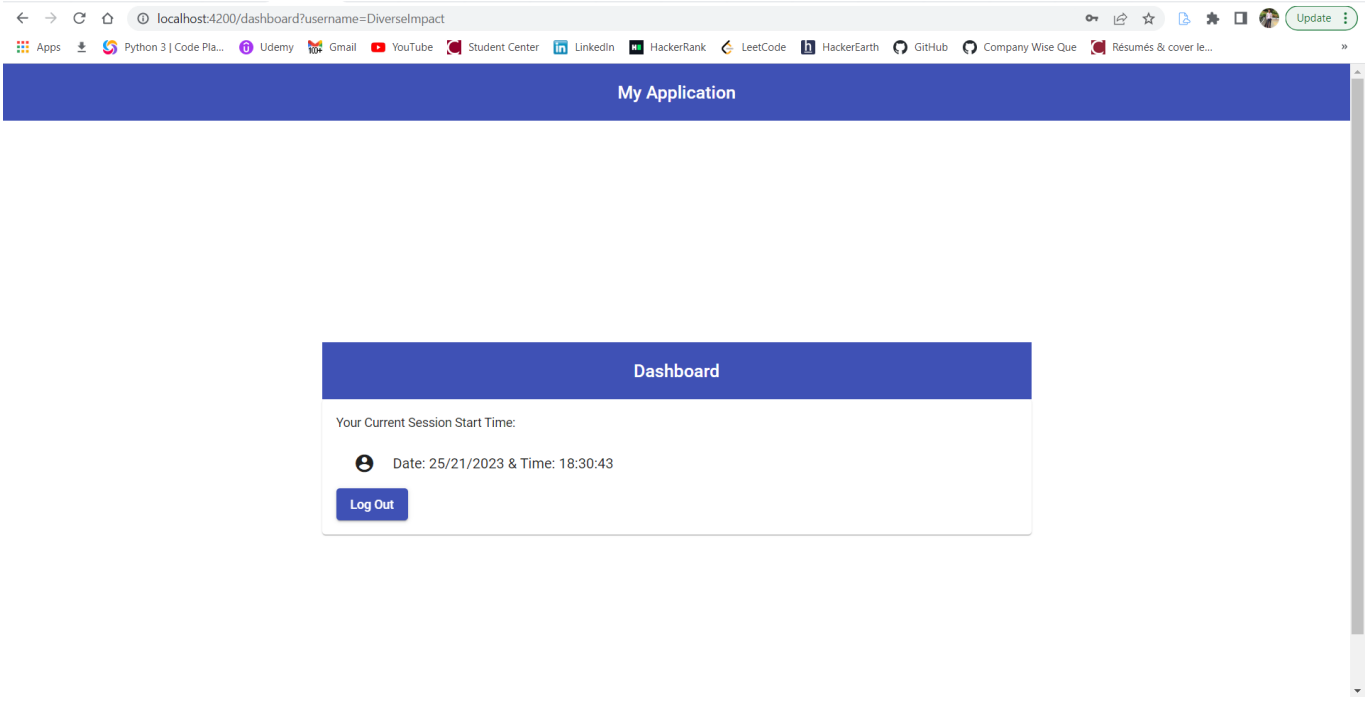
Grok patterns are set in a configuration file and can be tailored to the log data being processed. The extracted information may then be utilised in Logstash for additional processing or sent directly to Elasticsearch for indexing and searching. The Grok filter is a strong tool for converting unstructured log data into structured data that can be studied and acted upon.

```
filter {
  grok {
    patterns_dir => ["./patterns"]
    # logout, crete account, login
    match => { "message" =>
      [
        '%{TIMESTAMP_ISO8601:timeStamp} \[%{USERNAME:serviceName}\] %{WORD:logLevel}: %{GREEDYDATA:event} - %{GREEDYDATA:action} : \
{"%{WORD}":"%{WORD:firstName}","%{WORD}":"%{WORD:lastName}","%{WORD}":"%{WORD:userName}","%{WORD}":"%
{TIMESTAMP_ISO8601:loginDate}","%{WORD}":%{NUMBER:sessionTimeInSec},"%{WORD}":"%{GREEDYDATA:loginCountry}","%{WORD}":"%
{WORD:loginBrowser}"\}',
        '%{TIMESTAMP_ISO8601:timeStamp} \[%{USERNAME:serviceName}\] %{WORD:logLevel}: %{GREEDYDATA:event} - %{GREEDYDATA:action} : \
{"%{WORD}":"%{WORD:firstName}","%{WORD}":"%{WORD:lastName}","%{WORD}":"%{WORD:userName}","%{WORD}":"%{GREEDYDATA:password}"\}',
        '%{TIMESTAMP_ISO8601:timeStamp} \[%{USERNAME:serviceName}\] %{WORD:logLevel}: %{GREEDYDATA:event} - %{GREEDYDATA:action} : \
{"%{WORD}":"%{WORD:userName}","%{WORD}":"%{GREEDYDATA:password}"\}'
      ]
    }
  }
}
```

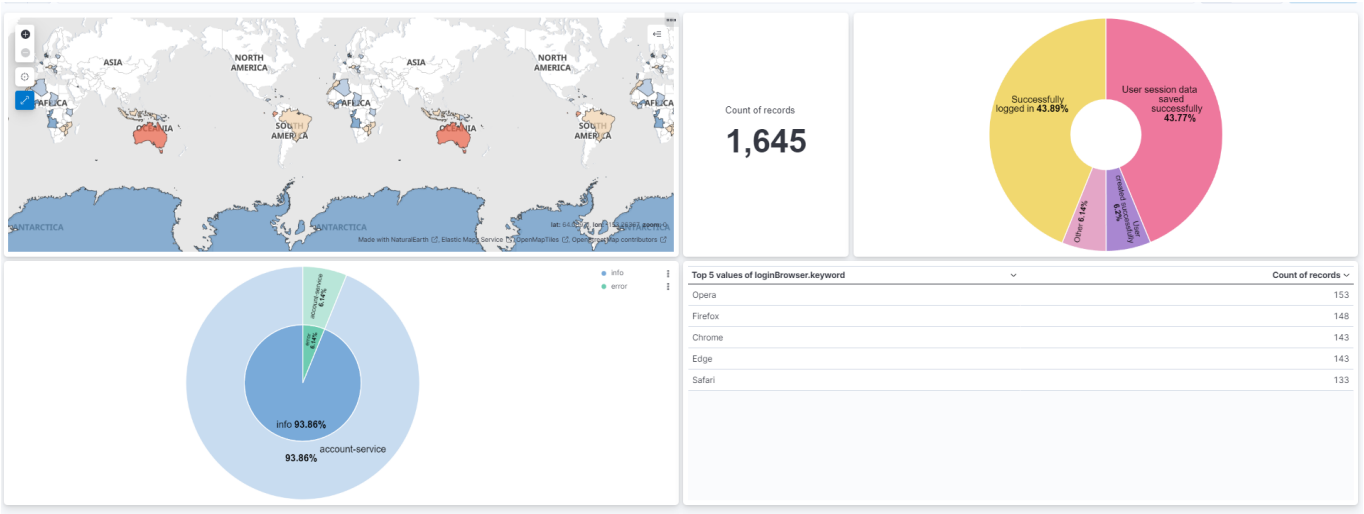
Screenshots of the Application

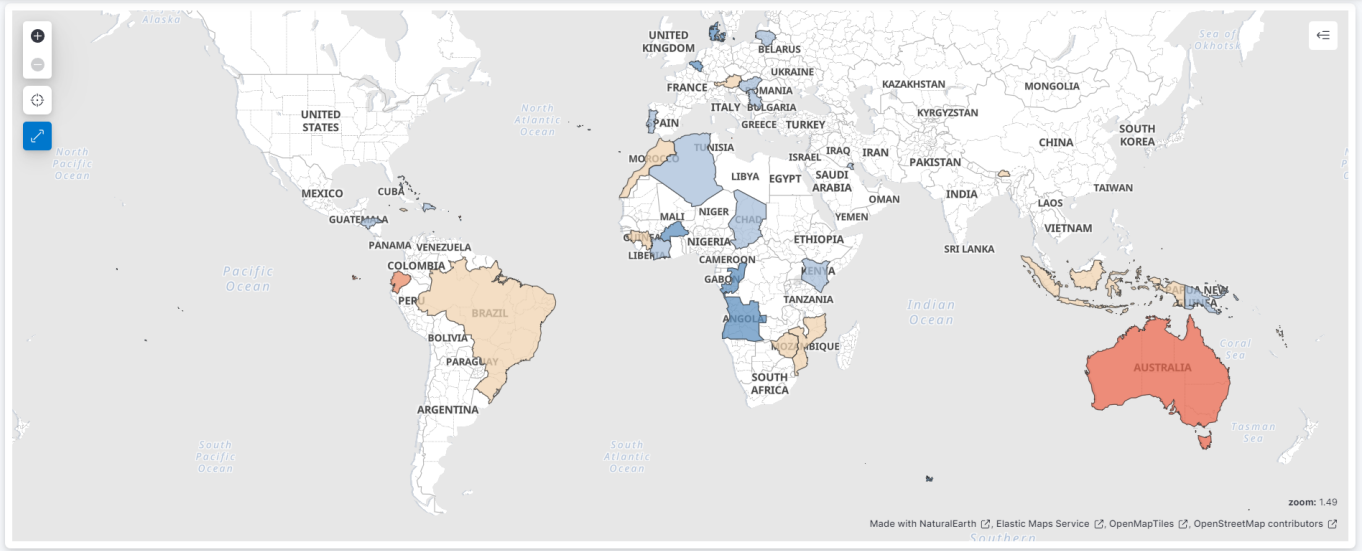


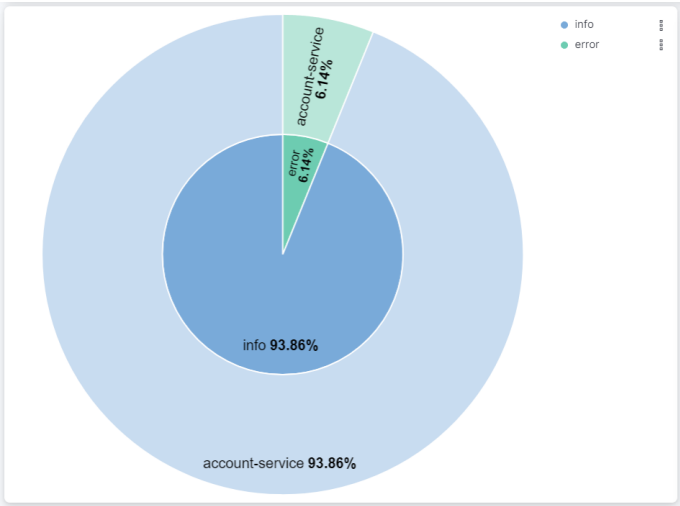


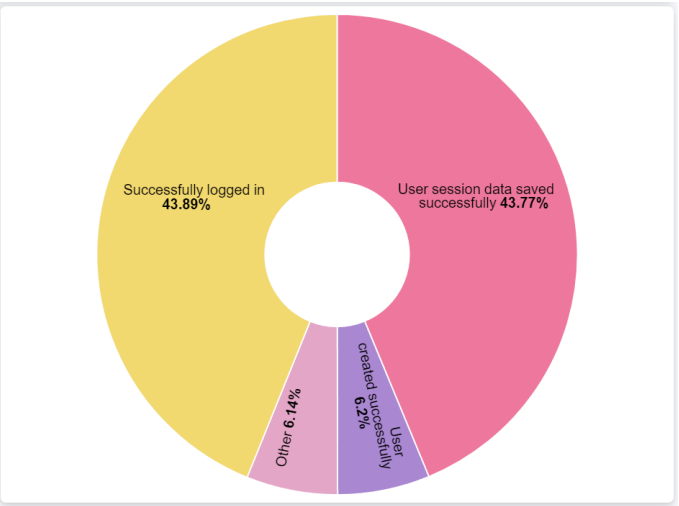


Screenshots of the Kibana Dashboard









Count of records

1,645

Top 5 values of loginBrowser.keyword	Count of records
Opera	153
Firefox	148
Chrome	143
Edge	143
Safari	133

Conclusion

Implementing a project utilising the ELK (Elasticsearch, Logstash, and Kibana) stack for log debugging and analysis, in conjunction with a full-stack application, was a good opportunity for me to demonstrate my talents to my management. In doing so, I exhibited not just my technical expertise but also my enthusiasm to acquire new technologies and my capacity to go above and beyond the requirements of my coop job.

It also aided in the implementation of an open-source ELK stack solution for log analysis, which will save the organisation money because it is license-free and will aid in the transition from pricey programmes such as Splunk. Utilizing the ELK stack assisted me in centralising and analysing logs created by various apps and servers, providing me with vital insights into the functioning of my system and spotting any issues that developed. I was able to demonstrate my abilities to design, create, and deploy a complete system, from the front-end to the back-end, by creating a full-stack application.

In conclusion, creating a project utilising the ELK stack and a full-stack application was an excellent method for me to exhibit my talents, initiative, and want to learn. As a result, I enhanced my visibility to my boss, boosted my prospects of converting my coop position to full-time employment, and positioned myself for future career advancement opportunities.