**INTERNSHIP REPORT**

A report submitted in fulfilment of the requirements for the Award of Degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

by

NARRA VARSHA, CSE, 3rd year, (160820733004)

SNEHA TULSIAN, CSE, 3rd year, (160820733121)

AKULA AMANAGANTI, CSE, 3rd year, (160820733129)

PULIMAMIDI BHAVYA SRI, CSE, 3rd year, (160820733130)

Under Supervision of

Dr. Santanu Chatterjee

RESEARCH CENTRE IMARAT-DRDO

(Duration -8th may, 2023 to 8th June, 2023)



**ACKNOWLEDGEMENT**

First we would like to thank Dr. Santanu Chatterjee, Scientist-“E”, Research Centre Imarat for giving us the opportunity to do an internship within the organization. We also would like to thank all the people that worked along with us with patience and openness, they created an enjoyable working environment.

**ABSTRACT**

Combining security information management (SIM) and security event management (SEM), security information and event management (SIEM) offers real-time monitoring and analysis of events as well as tracking and logging of security data for compliance or auditing purposes.

SIEM is a security solution that helps organizations recognize potential security threats and vulnerabilities before they have a chance to disrupt business operations. It surfaces user behaviour anomalies and automates many of the manual processes associated with threat detection and incident response and has become a staple in modern-day security operation centres (SOCs) for security and compliance management use cases.

Some SIEM tools may be designed to work better with either TCP or UDP logs, while others may be able to handle both protocols. Ultimately, the choice of protocol may depend on factors such as the network environment and the needs of the organization. Our aim is to develop a real time system that collects UDP syslog’s and create a robust parsing algorithm that converts the logs into a common format. These logs are stored in a database for further analysis. The users can view information based on custom filters through a user friendly interface.

**1. INTRODUCTION**

At the most basic level, all SIEM solutions perform some level of data aggregation, consolidation and sorting functions in order to identify threats and adhere to data compliance requirements. While some solutions vary in capability, most offer the same core set of functionality:

**Log Management**

SIEM captures event data from a wide range of source across an organization’s entire network. Logs and flow data from users, applications, assets, cloud environments, and networks is collected, stored and analysed in real-time, giving IT and security teams the ability to automatically manage their network's event log and network flow data in one centralized location.

Some SIEM solutions also integrate with third-party threat intelligence feeds in order to correlate their internal security data against previously recognized threat signatures and profiles. Integration with real-time threat feeds enable teams to block or detect new types of attack signatures.

**Incident Monitoring and Security Alerts**

Because they enable centralized management of on premise and cloud-based infrastructure, SIEM solutions are able to identify all entities of the IT environment. This allows SIEM technology to monitor for security incidents across all connected users, devices, and applications while classifying abnormal behaviour as it is detected in the network. Using customizable, predefined rules, administrators can be alerted immediately and take appropriate actions to mitigate it before it materializes into more significant security issues.

**Compliance Management and Reporting**

SIEM solutions are a popular choice for organizations subject to different forms of regulatory compliance. Due to the automated data collection and analysis that it provides, SIEM is a valuable tool for gathering and verifying compliance data across the entire business infrastructure. SIEM solutions can generate real-time compliance reports and reducing the burden of security management and detecting potential violations early so they can be addressed. Many of the SIEM solutions come with pre-built, out-of-the-box add-ons that can generate automated reports designed to meet compliance requirements.

**2. SYSTEM ANALYSIS**

**2.1. REQUIREMENT ANALYSIS**

**EXISTING SYSTEMS:**

There are a variety of existing SIEM (Security Information and Event Management) tools available in the market today. These include both paid and free solutions, as well as open-source tools. Some of the most popular SIEM tools include Solar Winds SIEM Security and Monitoring, Salesforce, ManageEngine Log360, and IBM QRadar. Some SIEM tools may be designed to work better with either TCP or UDP logs, while others may be able to handle both protocols.

**Existing SIEM tools for UDP logs**: There are many SIEM tools available that can handle UDP logs. Some examples of SIEM tools that can handle UDP logs include Datadog, IBM QRadar, and Splunk. These tools can be configured to tail log files or listen for logs sent over UDP. Additionally, provide log streaming capabilities that allow users to index, access, search, and correlate events using their existing Log Management and SIEM tools.

**Existing SIEM tools for TCP logs**: There are several SIEM tools available that can help with the analysis of TCP logs. Some popular examples include Splunk, IBM QRadar, ArcSight, LogRhythm, AlienVault, and Graylog. These tools can be used to monitor and analyze network traffic, detect anomalies and security threats, and provide alerts based on predefined rules and policies.

Some organizations may have specific requirements or preferences for which type of log transport protocol to use based on their specific needs or infrastructure. Ultimately, the choice between TCP and UDP for log transport will depend on a number of factors, including the specific use case and security requirements.

**2.2. METHODOLOGY:**

The main objective of this project is to develop a SIEM system that can parse logs, store in a database, automate the process, and create a UI for viewing the parsed logs.

**Parsing the Logs:**

The first step in developing a SIEM system is to collect and parse the logs. The logs are collected from various sources such as firewalls, access control systems, intrusion detection systems, and web servers. These logs are in different formats, so the first step is to convert them into a common format for analysis. In order to convert the logs into the common format we used regular expressions. We used Python Regex in the code which helps in filtering and parsing the logs to the desired destination.

**Obtaining the Database:**

Once the logs are parsed, they need to be stored in a database for further analysis. A database management system can be used for this purpose. The type of database used depends on the size and complexity of the data being collected.

There are various types of databases available such as SQL, NoSQL and Object-oriented databases. Each type of database has its own advantages and drawbacks. In this project, we will be using a NoSQL database such as MongoDB for storing the parsed logs.

**Automating the Process:**

After the logs are parsed and stored in the database, the next step is to automate the process. This involves configuring the SIEM system to perform certain actions automatically based on certain events. For example, syslog’s are automatically entered into a database without any intervention after being parsed as the logs incoming speed is very high we will also be needing certain open streaming platforms such as Kafka.

**Creating a UI for Viewing the Parsed Logs:**

The final step in developing a SIEM system is to create a UI for viewing the parsed logs. This UI should provide a clear picture of the security events happening in the system. It should be easy to navigate and provide the necessary information at a glance.

**3. SOFTWARE REQUIREMENTS SPECIFICATIONS**

**3.1. SYSTEM CONFIGURATIONS**

**Software Requirements:**

• Operating system : Windows 10 Pro.

• Coding Language : Python, JavaScript

• Mark Up Language: HTML, CSS

• Front-End : React.js

• Back-End : Node.js, Express.js

• Data Base : MongoDB

**Hardware Requirement:**

• System : Intel® Core™ i5.

• Hard Disk : 1TB.

• Ram : 4GB.

**4. CODE**

**4.1. Automation Code**

import socket

f1=open("window\_logs.csv","w")

f2=open("window\_logs2.csv","w")

c=0

p=0

sock=socket.socket(socket.AF\_INET,socket.SOCK\_DGRAM)

sock.bind(('0.0.0.0',514))

print('start process:')

count=0

while True:

try:

data,addr=sock.recvfrom(65535)

count+=1

print("in loop")

if c<3:

c=c+1

f1.write(addr[0]+" "+(data).decode('utf-8'))

f1.write("\n”)

print("1 write:",count)

if c==3:

p=0

f2.truncate()

f2.close()

f2=open("window\_logs2.csv","r+")

elif p<3:

p=p+1

f2.write(addr[0]+" "+(data).decode('utf-8'))

f2.write("\n")

print("1 write")

if p==3:

c=0

f1.truncate()

f1.close()

f1=open("window\_logs.csv","r+")

except:

print("in except 2")

f1.close()

f2.close()

sys.exit()

**4.2. Automated Parsing Code**

from pymongo import MongoClient

import re

import csv

client=MongoClient('localhost',27017)

db = client['project\_db']

collection1 = db['win2\_log']

sev\_list=["emerg(0)","alert(1)","crit(2)","err(3)","warning(4)","notice(5)","info(6)","debug(7)"]

sev\_dict={"E":"emerg(0)","A":"alert(1)","C":"crit(2)","err":"err(3)","W":"warning(4)","N":"notice(5)","I":"info(6)","D":"debug(7)"}

month\_list=["Jan","Feb","Mar","Apr","May","June","July","Aug","Sep","Oct","Nov","Dec"]

win2\_pattern ="(\d{2}.\d{2}.\d{3}.\d{2}) <(\d+)>(\d+) (\d{4})-(\d{2})-(\d{2})[T](\d{2}:\d{2}:\d{2})[+](\d{2}:\d{2}) (\d{2}.\d{2}.\d{3}.\d{2}) ([A-Z]\*) - (\w\*) - %([A-Z]\*)-([A-Z])-(\w\*): (.\*)"

win1\_pattern ="(\d{2}.\d{2}.(\d{3}|\d{2}).(\d{3}|\d{2})) <(\d+)>(\d+): ([\*|.]([a-zA-Z]\*)|([a-zA-Z]\*)) (.)([0-9]) (\d{2}:\d{2}:\d{2}(|([.]\d\*))): %([A-Z]\*)-([0-9])-(\w\*):(.\*)"

while True:

doc\_list=[]

doc\_dict={}

if len(list(csv.reader(open(r'window\_logs.csv'))))==3:

fp=open("window\_logs.csv","r+")

lines=fp.readlines()

fp.close()

fp=open("window\_logs.csv","w")

fp.close()

elif len(list(csv.reader(open(r'window\_logs2.csv'))))==3:

fp=open("window\_logs2.csv","r+")

lines=fp.readlines()

fp.close()

fp=open("window\_logs2.csv","w")

fp.close()

else:

continue

print(lines)

print("in loop-1")

try:

for l in lines:

try:

print('in try block')

matches1=re.findall(win2\_pattern, l)

matches2=re.findall(win1\_pattern,l)

if len(matches1):

print('in if block')

doc\_dict={

'ip\_addr':matches1[0][0],

'month':month\_list[int(matches1[0][4])-1],

'date' : matches1[0][5],

'time' :matches1[0][6],

'facility':matches1[0][11],

'mnemonic' :matches1[0][13],

'severity' :sev\_dict[(matches1[0][12])],

'message' :matches1[0][-1]

}

print(doc\_dict)

doc\_list.append(doc\_dict)

elif len(matches2):

print('in if block2')

doc\_dict={

'ip\_addr':matches2[0][0],

'month':matches2[0][5],

'date' : matches2[0][8] + matches2[0][9],

'time' :matches2[0][10],

'facility':matches2[0][13],

'mnemonic' :matches2[0][15],

'severity' :sev\_list[int(matches2[0][14])],

'message' :matches2[0][-1]

}

print(doc\_dict)

doc\_list.append(doc\_dict)

except:

continue

collection1.insert\_many(doc\_list)

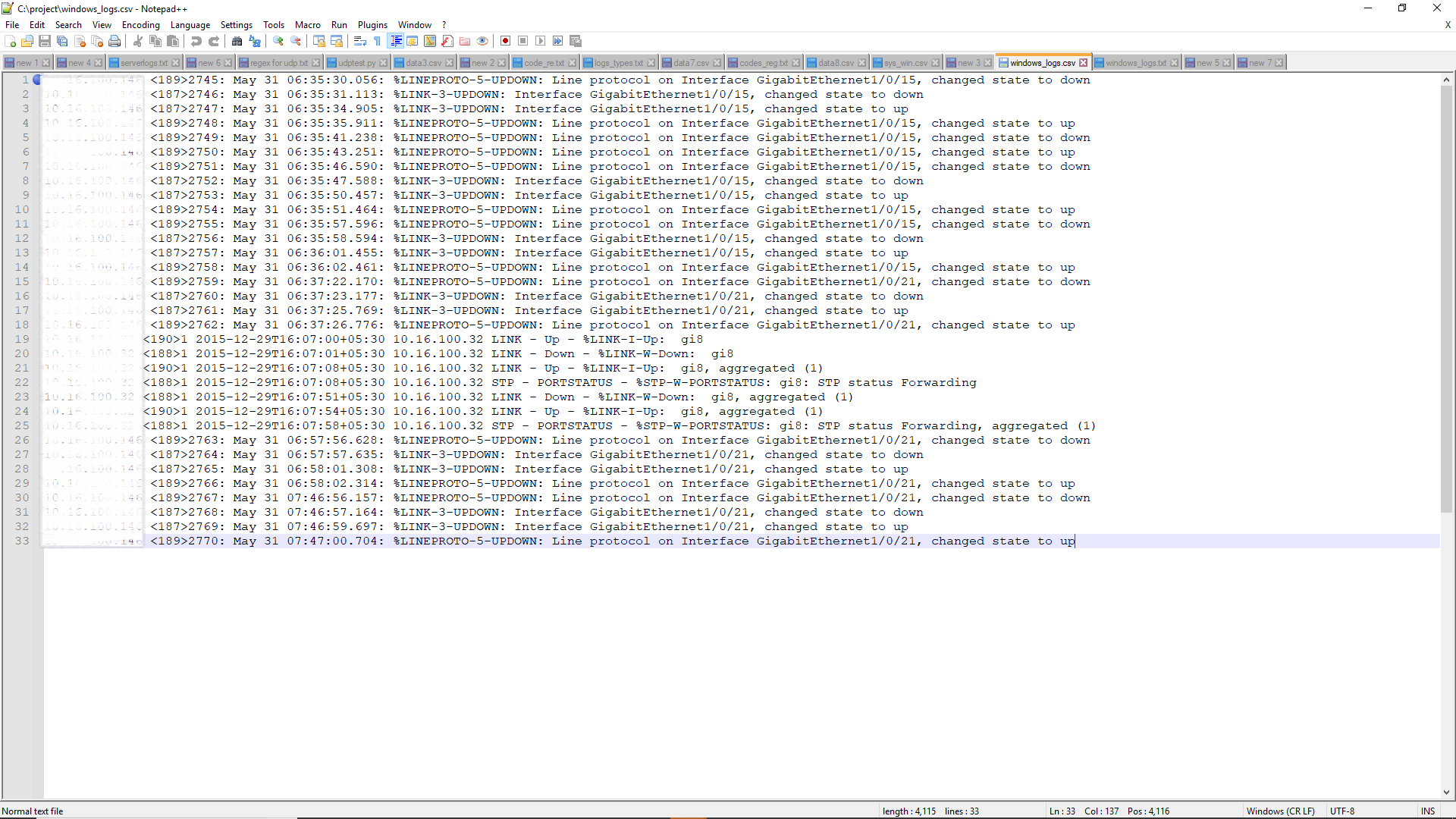
except:

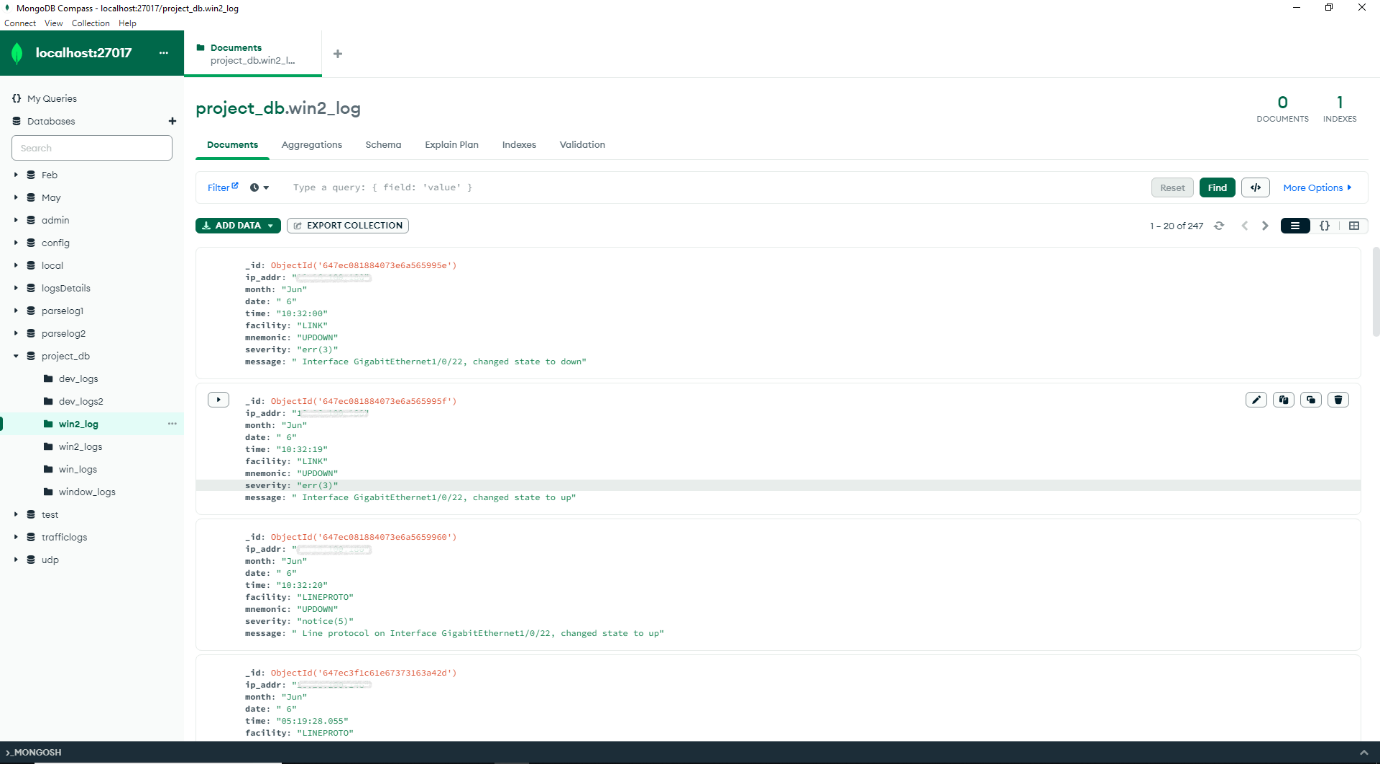
continue

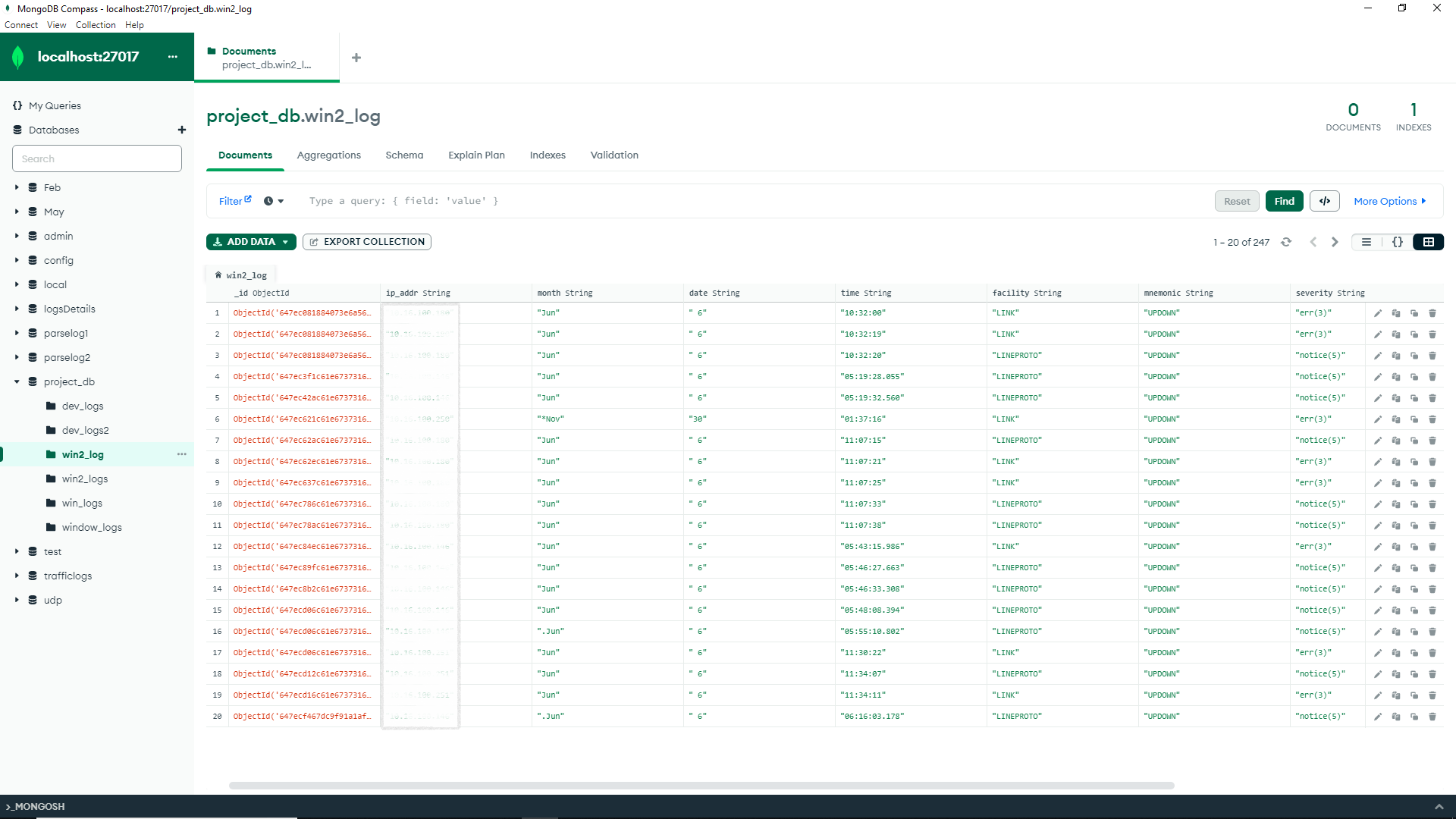
**4.3. UI CODE:**

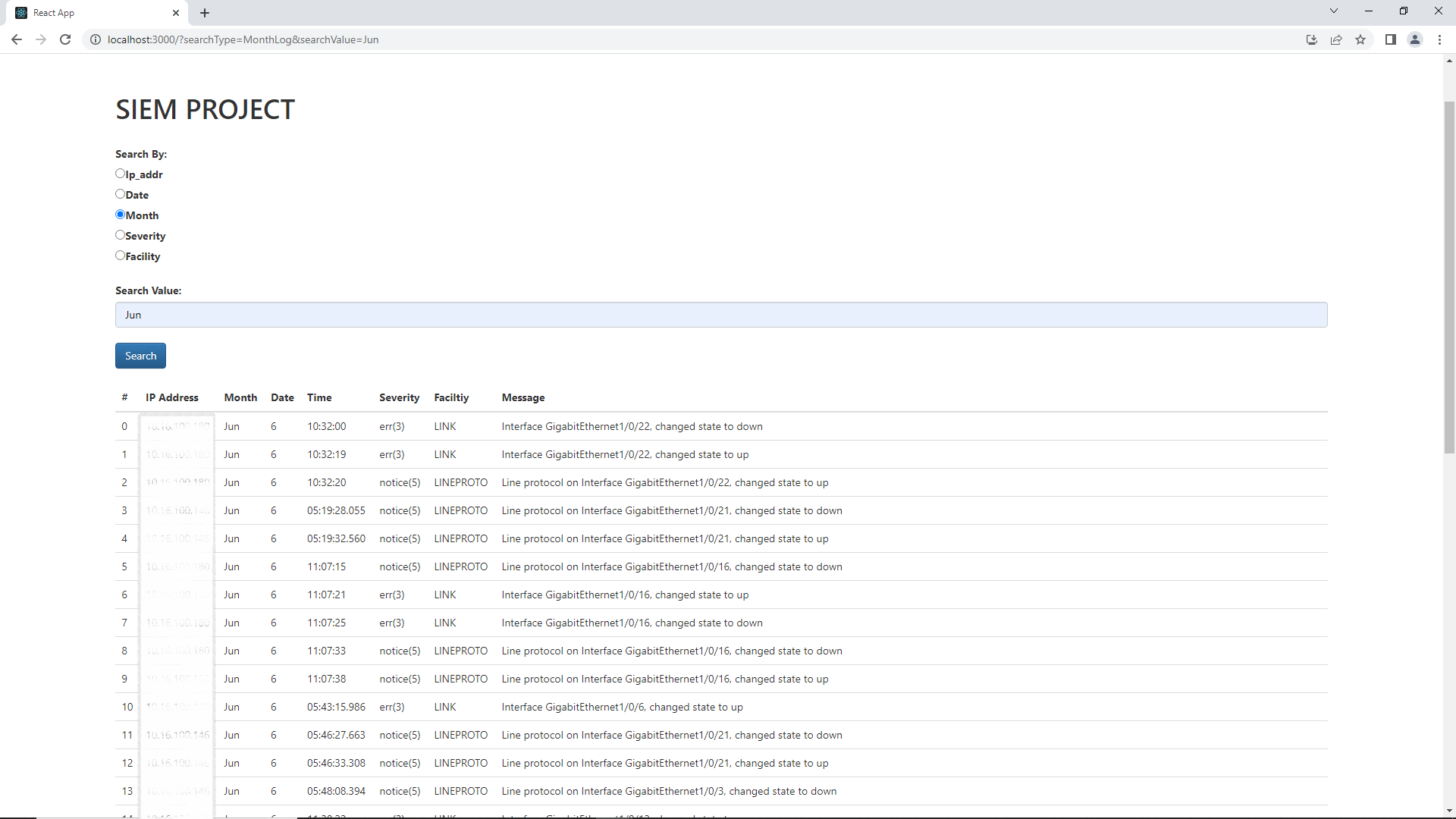
https://github.com/varsha1n/experiment

**5. SCREENSHOTS**

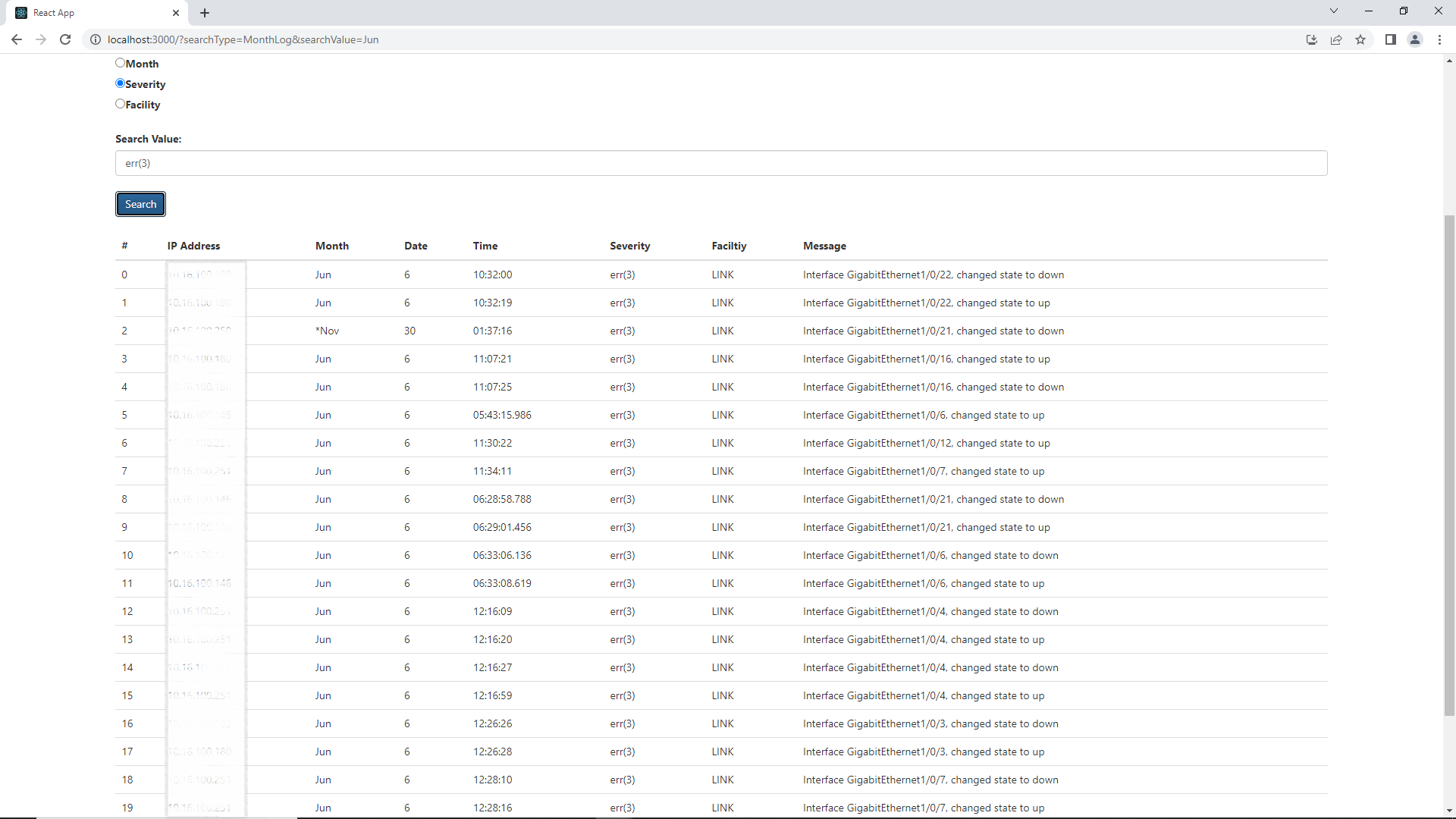
**Types of logs :**

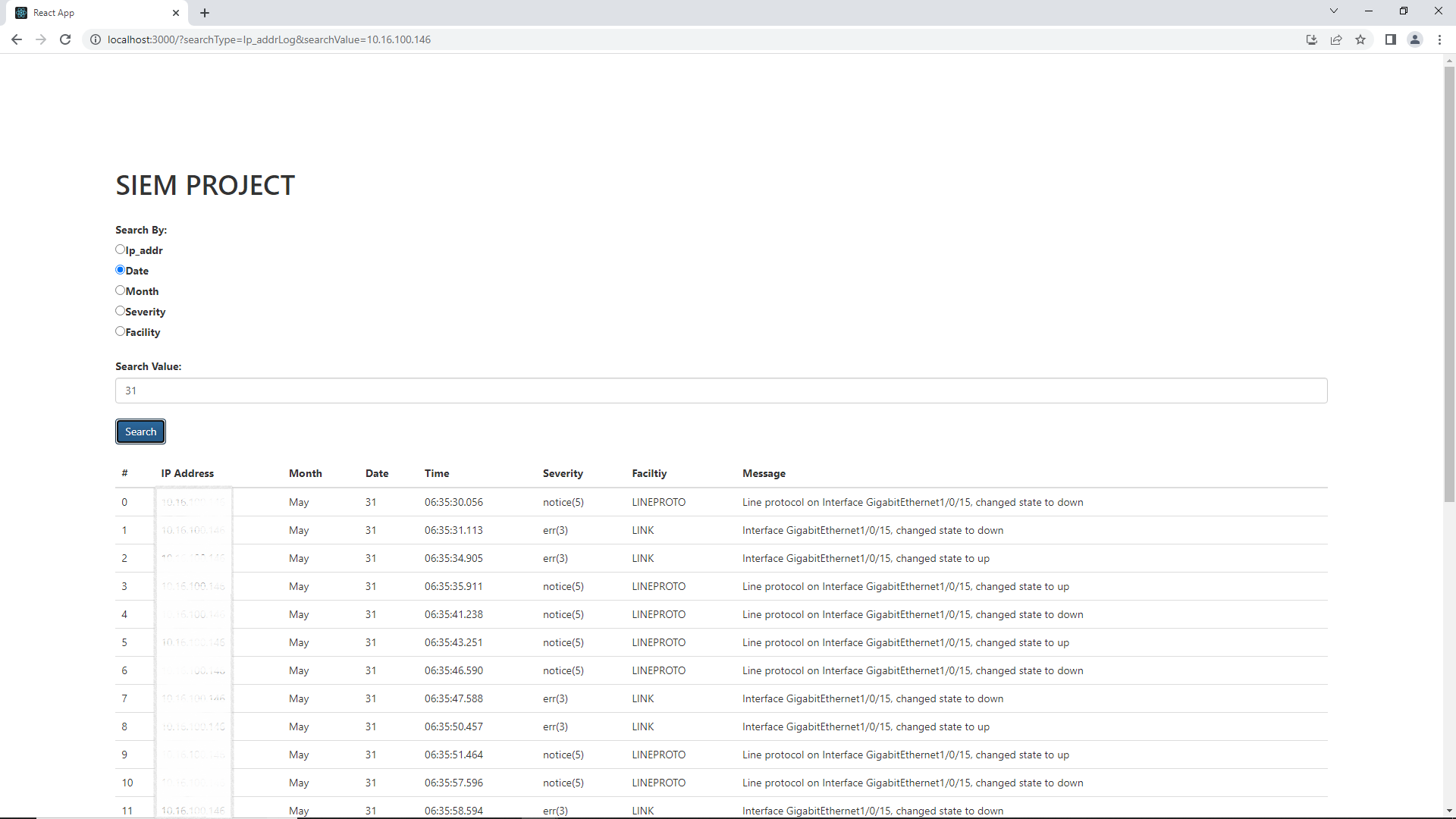
**Parsed Logs :**

**Logs in Database:**

**UI : Custom filter - Month**

**UI : Custom filter - Severity**

****

**UI : Custom filter - Date**

**6. CONCLUSION**

SIEM is an essential tool for any organization that wants to protect its assets and ensure the confidentiality, integrity, and availability of its data. It enables organizations to detect, prevent, and respond to security breaches and incidents in real-time, thereby minimizing the damage caused by such events. The implementation of a SIEM solution can be complex and time-consuming, but the benefits it offers in terms of improved security posture, regulatory compliance, and overall risk management make it a worthwhile investment. As security threats continue to evolve and become more sophisticated, SIEM will remain a critical component of any comprehensive security strategy.