**BASAVARAJESWARI GROUP OF INSTITUTIONS**

**Ballari Institute of Technology & Management**

**AUTONOMOUS INSTITUTE UNDER VISVESVARAYA TECHNOLOGICAL UNIVERSITYJNANA SANGAMA, BELAGAVI 590018**

**INTERNSHIP**

**Report On**

# LOG FILE ANALYZER

Submitted in partial fulfillment of the requirements for the award of degree of

**Bachelor of Engineering**

**In**

**COMPUTER SCIENCE – DATA SCIENCE**

## Submitted by

**MALLIKARJUN HIREMATH**

**3BR23CD056**

## Internship Carried Out By

**EZ TRAININGS & TECHNOLOGIES PVT.LTD HYDERABAD**

**Internal Guide External Guide**

**PARVATHI Vishal Kumar**

**Asst.prof , CSE (DS) Technical Trainer**

ANUSHYA

**Asst. prof,CSE (DS)**

### BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution\*

**(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)**

**"Jnana Gangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India) Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197**

**2024-2025**

**BASAVARAJESWARI GROUP OF INSTITUTIONS**

## BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

**Autonomous institute under VISVESVARAYA TECHNOLOGICAL UNIVERSITYJNANA SANGAMA,**

**BELAGAVI 590018**

NACC Accredited Institution\*

**(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to**

**Visvesvaraya Technological University, Belagavi)**

### "JnanaGangotri" Campus, No.873/2,Ballari-HospetRoad,Allipur,

**Ballar1-583 104 (Karnataka)(India)**

**Ph: 08392 – 237100 / 237190, Fax: 08392 –237197**

**DEPARTMENT OF COMPUTER SCIENCE – DATA SCIENCE**

# CERTIFICATE

This is to certify that the Internship entitled **“ LOG FILE ANALYZER”** has been successfully completed by **MALLIKARJUN HIREMATH** bearing USN **3BR23CD056** a bonafide student of Ballari Institute of Technology and Management, Ballari. For the partial fulfillment of the requirementsfor the **Bachelor’s Degree in Computer Science and Engineering (DS)** of the VISVESVARAYA TECHNOLOGICAL UNIVERSITY,

Belagavi during the academic year 2024-2025.

**Signature of Internship Signature of HOD**

**Co-ordinator**

**PARVATHI Dr. ARADHANA**

**Asst. prof, CSE (DS) Prof. and HOD CSE(DS)**

**ANUSHYA**

**Asst. prof, CSE (DS)**

**DECLARATION**

I, **MALLIKARJUN HIREMATH** **,** second year student of Computer Science and Engineering (DS)Ballari Institute of Technology, Ballari, declare that Internship entitled **LOG FILE ANALYZER** is a part of Internship Training successfully carried out by **EZ TECHNOLOGIES & TRAININGS PVT. LTD, Hyderabad** at “**BITM, BALLARI”.** This report is submitted in partial fulfillment of the requirements for the award of the degree, Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi.

**Date : 28 Sep 2024 Signature of the Student**

**Place : Bellary**

**ACKNOWLEDGEMENT**

The satisfactions that a company the successful completion of my internship on “LOG FILE ANALYZER” would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is my privilege to express my gratitude and respect to all those who inspired me in the completion of my internship.

I am grateful to our respective coordinator **“PARVATHI (Asst.prof,CSE(DS)), ANUSHYA**

**(Asst. prof, CSE (DS))”** for his noble gesture, support co-ordination and valuable suggestions given to me in the completion of Internship.

I also thank **ARADHANA,** H.O.D. Department of **Computer science – Data Science Engineering** for extending all his valuable support and encouragement.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Chapter Name** | **Page No.** |
| **1** | **College Profile** | **01** |
| **2** | **Certificate** | **02** |
| **3** | **Declaration** | **03** |
| **4** | **Acknowledgement** | **04** |
| **5** | **Internship Report** | **06** |
| **6** | **Introduction** | **07** |
| **7** | **Code Implementation** | **09-10** |
| **8** | **Output** | **10-11** |
| **9** | **Conclusion** | **11** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Day** | **Date** | **Content Covered** | **Signature of the** |
| **faculty in-charge** |
| **1** | **09.09.24** | **Introduction to Python, Setup & Installation, First Python Program, Variables, Data Types,**  **`and Basic I/O** |  |
| **2** | **10.09.24** | **Control Structures: If-else, Loops, Functions and Modules** |  |
| **3** | **11.09.24** | **Lists, Tuples, and Dictionaries, File Handling** |  |
| **4** | **12.09.24** | **Exception Handling, Practice exercises on Python basics** |  |
| **5** | **13.09.24** | **Introduction to OOP, Classes, and Objects** |  |
| **6** | **14.09.24** | **Inheritance, Polymorphism, and Encapsulation** |  |
| **7** | **15.09.24** | **Abstract Classes and Interfaces** |  |
| **8** | **17.09.24** | **Practice exercises on OOP concepts** |  |
| **9** | **18.09.24** | **Introduction to DSA, Arrays, and Linked Lists** |  |
| **10** | **19.09.24** | **Introduction to Stacks and Queues** |  |
| **11** | **20.09.24** | **Practice exercise on basic concept(Reduce, Lambda function, List comprehension)** |  |
| **12** | **21.09.24** | **Introduction to Tree Data Structure** |  |
| **13** | **23.09.24** | **Introduction to Graph Data Structure** |  |
| **14** | **24.09.24** | **Searching Algorithm project building and presentation** |  |
| **15** | **25.09.24** | **Project Building & Presentations** |  |
| **16** | **26.09.24** | **Project Building & Presentations** |  |
| **17** | **27.09.24** | **Project Building & Presentations** |  |
| **18** | **28.09.24** | **Project Building & Presentations** |  |

**INTRODUCTION**

In today's data-driven landscape, the ability to analyze log files is crucial for understanding system performance, troubleshooting issues, and enhancing security. Log files, generated by applications, servers, and network devices, contain valuable information that, when properly analyzed, can reveal insights into system behavior, user activity, and potential anomalies.

This Log File Analyzer project aims to provide a robust tool for processing and interpreting log data efficiently. By leveraging advanced parsing techniques and data visualization methods, the project will enable users to identify trends, monitor system health, and detect irregularities in real time. The goal is to empower organizations to make informed decisions based on comprehensive log analysis, ultimately improving operational efficiency and security posture.

Through this project, we will explore various log formats, implement scalable processing algorithms, and develop an intuitive user interface to facilitate easy access to critical insights. By automating the log analysis process, we aim to save time and resources while providing actionable intelligence that supports proactive system management.

1. **Problem Statement**

To address the given problem Statement Log File Analyzer POC, we will create a simple system using object-oriented programming and data structure's. The system will perform CRUD operations On log entries Analyze the specific Log files For error massages and warnings And produces the summarises of the exstracted Information...

1. **Objectives**

Data Monitoring: Continuously monitor log files to track system performance and user activity.

Error Detection: Identify and analyze error messages to facilitate troubleshooting and improve system reliability.

Security Analysis: Detect unauthorized access attempts, vulnerabilities, and potential security breaches.

Usage Patterns: Analyze user behavior and system usage to inform capacity

1. **Technologies and Tools Used**

Programming Languages:

Python: Chosen for its powerful libraries and ease of use in data manipulation and analysis.

JavaScript: Utilized for front-end development to create a dynamic user interface.

Data Processing Libraries:

Pandas: For efficient data manipulation and analysis of structured data.

NumPy: For numerical operations and handling large datasets.

Log Parsing Libraries:

Loguru: A Python library that simplifies logging and log file parsing.

regex: For pattern matching and extracting relevant information from log entries.

Database:

SQLite or PostgreSQL: Used to store parsed log data for easy querying and retrieval.

Visualization Tools:

Matplotlib: For creating static, animated, and interactive visualizations in Python.

D3.js: A JavaScript library for producing dynamic, interactive data visualizations in web applications

**CODE IMPLEMENTATION:**

#LOG FILE ANALYZER….

from datetime import datetime

from collections import defaultdict

class LogEntry:

def \_init\_(self, timestamp, log\_level, message):

self.timestamp = timestamp

self.log\_level = log\_level

self.message = message

def to\_string(self):

return f"{self.timestamp} [{self.log\_level}] {self.message}"

class LogFileAnalyzer:

def \_init\_(self):

self.logs = []

def create\_log(self, log\_level, message):

timestamp = datetime.now().isoformat()

log\_entry = LogEntry(timestamp, log\_level, message)

self.logs.append(log\_entry)

def read\_logs(self):

return [log.to\_string() for log in self.logs]

def update\_log(self, index, log\_level, message):

if 0 <= index < len(self.logs):

self.logs[index].log\_level = log\_level

self.logs[index].message = message

return True

return False

def delete\_log(self, index):

if 0 <= index < len(self.logs):

del self.logs[index]

return True

return False

def analyze\_logs(self):

error\_counts = defaultdict(int)

for log in self.logs:

if log.log\_level == "ERROR":

error\_counts[log.message] += 1

return dict(error\_counts)

def summarize\_logs(self):

summary = {

'total\_entries': len(self.logs),

'error\_count': sum(1 for log in self.logs if log.log\_level == "ERROR"),

'warning\_count': sum(1 for log in self.logs if log.log\_level == "WARNING"),

}

return summary

def export\_logs(self, filename):

with open(filename, 'w') as f:

for log in self.logs:

f.write(log.to\_string() + '\n')

def import\_logs(self, filename):

try:

with open(filename, 'r') as f:

for line in f:

timestamp, rest = line.split(' [', 1)

log\_level, message = rest.split('] ', 1)

self.logs.append(LogEntry(timestamp, log\_level.strip(), message.strip()))

except Exception as e:

print(f"Error importing logs: {e}")

# Example usage

if \_name\_ == "\_main\_":

analyzer = LogFileAnalyzer()

while True:

print("\nMenu:")

print("1: Create logs")

print("2: Read logs")

print("3: Update logs")

print("4: Delete logs")

print("5: Analyze logs")

print("6: Summarize logs")

print("7: Export logs")

print("8: Import logs")

print("0: Exit")

choice = input("Enter your choice: ")

if choice == '1':

log\_level = input("Enter log level (INFO, WARNING, ERROR): ")

message = input("Enter log message: ")

analyzer.create\_log(log\_level, message)

elif choice == '2':

print("\nLog Entries:")

for log in analyzer.read\_logs():

print(log)

elif choice == '3':

index = int(input("Enter log index to update: "))

log\_level = input("Enter new log level: ")

message = input("Enter new log message: ")

if analyzer.update\_log(index, log\_level, message):

print("Log updated successfully.")

else:

print("Invalid index.")

elif choice == '4':

index = int(input("Enter log index to delete: "))

if analyzer.delete\_log(index):

print("Log deleted successfully.")

else:

print("Invalid index.")

elif choice == '5':

print("\nError Analysis:")

error\_analysis = analyzer.analyze\_logs()

print(error\_analysis)

elif choice == '6':

print("\nLog Summary:")

summary = analyzer.summarize\_logs()

print(summary)

elif choice == '7':

filename = input("Enter filename to export logs: ")

analyzer.export\_logs(filename)

print("Logs exported successfully.")

elif choice == '8':

filename = input("Enter filename to import logs: ")

analyzer.import\_logs(filename)

print("Logs imported successfully.")

elif choice == '0':

print("\nGoodbye!")

break

else:

print("Invalid option.")

Output:

============================= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python312/LOG.PY =============================

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

**Enter your choice: 1**

**Enter log level (INFO, WARNING, ERROR): INFO**

**Enter log message: ATTEMPT TO LOG-IN**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

**Enter your choice: 1**

**Enter log level (INFO, WARNING, ERROR): ERROR**

**Enter log message: FAILED TO LOG IN**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

**Enter your choice: 1**

**Enter log level (INFO, WARNING, ERROR): WARNING**

**Enter log message: DISK SPACE RUNNING LOW**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

Enter your choice: 2

**Log Entries:**

**2024-09-28T19:59:46.873182 [INFO] ATTEMPT TO LOG-IN**

**2024-09-28T20:00:01.385413 [ERROR] FAILED TO LOG IN**

**2024-09-28T20:00:27.497004 [WARNING] DISK SPACE RUNNING LOW**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

**Enter your choice: 3**

**Enter log index to update: 0**

**Enter new log level: ERROR**

**Enter new log message: INVALID CREDENTIALS**

Log updated successfully.

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

Enter your choice: 2

**Log Entries:**

**2024-09-28T19:59:46.873182 [ERROR] INVALID CREDENTIALS**

**2024-09-28T20:00:01.385413 [ERROR] FAILED TO LOG IN**

**2024-09-28T20:00:27.497004 [WARNING] DISK SPACE RUNNING LOW**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

**Enter your choice: 1**

**Enter log level (INFO, WARNING, ERROR): INFO**

**Enter log message: USER LOGGE IN**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

Enter your choice: 2

**Log Entries:**

**2024-09-28T19:59:46.873182 [ERROR] INVALID CREDENTIALS**

**2024-09-28T20:00:01.385413 [ERROR] FAILED TO LOG IN**

**2024-09-28T20:00:27.497004 [WARNING] DISK SPACE RUNNING LOW**

**2024-09-28T20:01:52.257702 [INFO] USER LOGGE IN**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

**Enter your choice: 4**

**Enter log index to delete: 3**

**Log deleted successfully.**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

Enter your choice: 2

**Log Entries:**

**2024-09-28T19:59:46.873182 [ERROR] INVALID CREDENTIALS**

**2024-09-28T20:00:01.385413 [ERROR] FAILED TO LOG IN**

**2024-09-28T20:00:27.497004 [WARNING] DISK SPACE RUNNING LOW**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

Enter your choice: 5

**Error Analysis:**

**{'INVALID CREDENTIALS': 1, 'FAILED TO LOG IN': 1**}

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

Enter your choice: 6

**Log Summary:**

**{'total\_entries': 3, 'error\_count': 2, 'warning\_count': 1}**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

**Enter your choice: 7**

**Enter filename to export logs: INVALID CREDENTIALS**

**Logs exported successfully.**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

**Enter your choice: 8**

**Enter filename to import logs: INVALID CREDENTIALS**

**Logs imported successfully.**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

**Enter your choice: 2**

**Log Entries:**

**2024-09-28T19:59:46.873182 [ERROR] INVALID CREDENTIALS**

**2024-09-28T20:00:01.385413 [ERROR] FAILED TO LOG IN**

**2024-09-28T20:00:27.497004 [WARNING] DISK SPACE RUNNING LOW**

**2024-09-28T19:59:46.873182 [ERROR] INVALID CREDENTIALS**

**2024-09-28T20:00:01.385413 [ERROR] FAILED TO LOG IN**

**2024-09-28T20:00:27.497004 [WARNING] DISK SPACE RUNNING LOW**

Menu:

1: Create logs

2: Read logs

3: Update logs

4: Delete logs

5: Analyze logs

6: Summarize logs

7: Export logs

8: Import logs

0: Exit

**Enter your choice: 0**

**Goodbye!**

**CONCLUSION :**

In conclusion, the log file analyzer project successfully demonstrates the ability to process and analyze large volumes of log data efficiently. By employing various data parsing, filtering, and visualization techniques, the project provides valuable insights into system performance, security issues, and user behavior. The results can help organizations make informed decisions, improve system reliability, and enhance security measures. Future enhancements could include integrating machine learning for predictive analytics and expanding support for additional log formats, further increasing the tool's versatility and effectiveness.

**REFERENCE**:

<https://chatgpt.com/>

<https://www.google.co.in/>