# Abstract:Real-time Log Analysis Using Hadoop and Spark

The **Real-time Log Analysis** project is a Big Data solution designed to collect, process, and analyze system and application logs in real time using the **Hadoop** and **Apache Spark** ecosystem. It enables organizations to monitor application performance, detect anomalies or failures, and gain insights into user behavior or system usage patterns—all in near real-time for proactive decision-making.

## Key Features

- Collect streaming log data from servers or applications using tools like **Flume** or **Kafka**.
- Process logs in real-time using **Apache Spark Streaming**.
- Perform parsing, filtering, and aggregation of log records.
- Detect anomalies, frequent errors, or traffic spikes on the fly.
- Store processed logs in **HDFS** for batch analysis or historical trend tracking.
- Visualize trends, alerts, and system health using tools like **Grafana**, **Kibana**, or **Power BI**.

## Tools & Technologies Used

- **Apache Kafka** – for ingesting real-time logs
- **Apache Flume** – (alternative to Kafka) for log collection
- **Apache Spark Streaming** – for real-time processing
- **HDFS** – for storage of raw and processed logs
- **YARN** – for cluster resource management
- **Hive/Spark SQL** – for querying log data
- **Elasticsearch + Kibana** – (optional) for search and visualization
- **Scala or Python** – for Spark programming

## Input Data

- Web server logs (e.g., Apache/Nginx logs: IP address, timestamp, URL, response code)
- Application logs (e.g., error logs, transaction logs)
- System logs (e.g., CPU/memory usage, disk I/O logs)
- Streaming data from log-generating services or APIs

## Output/Reports

- Real-time error and event dashboards
- Aggregated reports: most visited URLs, error codes, user IPs
- Alerting on anomalies (e.g., traffic spikes, error surges)
- Hourly/daily log summaries stored in Hive/Spark tables
- Trend graphs and logs heatmaps for performance monitoring

## Processing Logic

- **Spark Streaming job** runs in a loop consuming batches of logs from Kafka every few seconds.
- Apply **filter conditions** (e.g., only ERROR or 500 status logs).
- Use **window operations** for time-based aggregations (e.g., every 10 mins).
- Store results in **HDFS** or **Elasticsearch**.

**24M11MC153**

**Salendram Sowmyasri**

**Aditya University**