# JobScraper

The Automatic Job Posting Tracker


- Chethan U Mahindrakar (002646783)
- Sayeed Ahmed (002191535)

# Problem Statement and Idea

**Problem Statement:**

Job seekers need to track job postings on multiple company websites, which can be time-consuming and inefficient.

**Idea:**

Develop a web scraping application that extracts job postings from company websites and notifies job seekers of matching postings using a Slack bot.

# Use Cases

**Use case 1: Job seeker looking for new opportunities**

Actor: Job seeker

Action: The web scraping application extracts job postings from various company websites and creates push notifications about new job postings that match the job seeker’s criteria through the Slack bot API.

Reaction: Job seeker would be able to quickly and easily find job openings that match their skills and interests without having to manually search multiple websites.

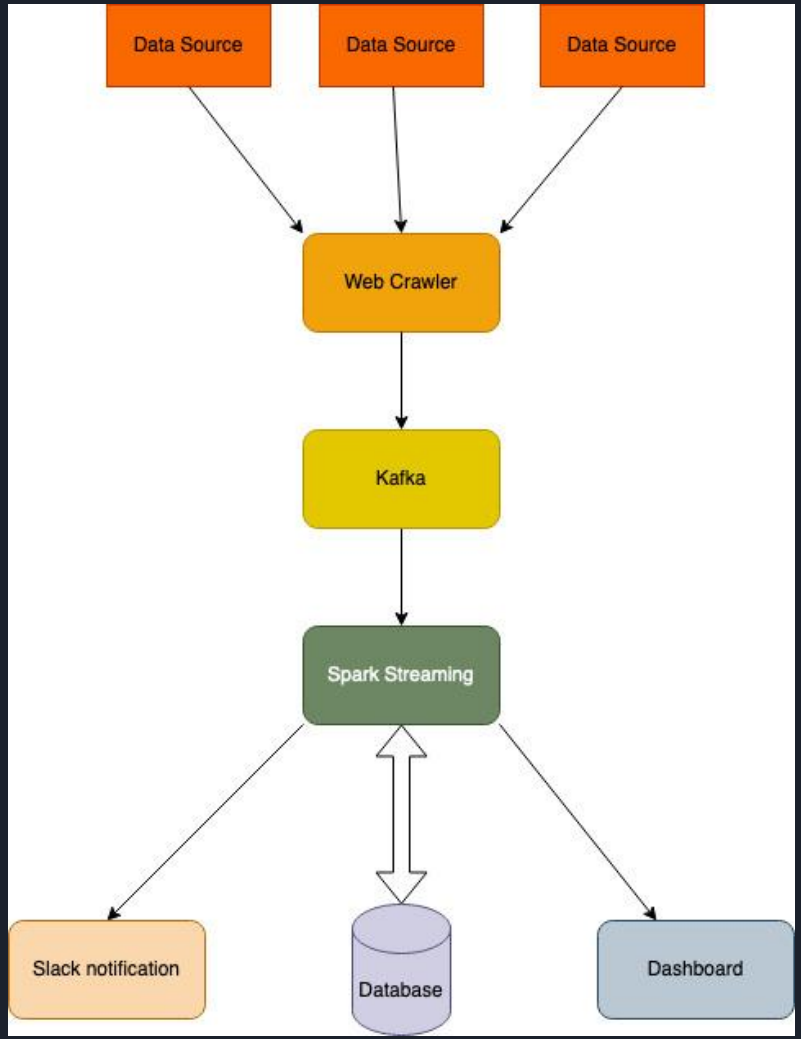
**Use case 2: Company looking to monitor their competitors' hiring strategies**

Actor: Company

Action: The company would use the application to extract job postings from their competitors' websites and analyze the data to identify trends and patterns in their hiring strategies.

Reaction: The company would gain insights into their competitors' hiring strategies, allowing them to adjust their own recruitment strategies as needed.

# Methodology



# Data Source

- Multiple company websites for job postings (unknown number of rows).
- Open-source datasets such as the Fortune Global 500 and Forbes Global 2000.

# Milestones/Sprints

Week 1: Develop web scraper and implement data extraction from company websites.

Week 2: Process and analyze the data using Spark.

Week 3: Implement Kafka for real-time processing and message queuing.

Week 4: Develop the Slack bot for user notifications and integrate it with the application.

Week 5: Optimize the system, conduct user testing, and finalize the project for submission.

# Scala Programming and Code Repository

- Web scraping application: Scala
- Database: SQL
- Kafka integration: Scala using Kafka APIs
- Slack bot: Scala for now / Any language that supports the Slack bot API, such as Python or JavaScript
- Spark: Scala
- Code repository: https://github.com/sayeedahmed01/JobScraper

# Acceptance Criteria

- Extraction of Job Postings: The web scraper application should be able to extract job postings from a company's website with a success rate of at least 80%.
- Data Processing and Analysis: Spark should be able to process and analyze job postings data with a processing speed of at least 500 rows per second.
- Periodic Retrieval and Storage: The application should be able to run periodically to retrieve new job postings and update the database with a success rate of at least 90%.

# Goals

- To provide a convenient tool for job seekers to monitor multiple company websites for job postings in real-time.
- Understanding of Web scraping, Apache Kafka, Apache Spark, Data Streaming, Slack Developer API’s.
- Provide insights through analysis on the Job Market.