**1. Objective**

* Purpose:   
  Develop a tool to monitor and analyze YouTube comments on social, economic, and political issues in real-time. This tool will collect comments, analyze sentiment, and provide a dashboard to visualize trends, aiding organizations and researchers in understanding public sentiment shifts on key issues.
* Core Goals:

1. Enable real-time data collection and analysis for specific social issues.
2. Perform accurate sentiment analysis on comments.
3. Visualize sentiment trends, keyword frequency, and engagement levels to provide actionable insights.

2. Scope

* In Scope:

1. Real-time collection of comments from targeted YouTube videos on issues like healthcare, climate change, and social justice.
2. Sentiment analysis using Python NLP libraries.
3. Development of an interactive dashboard to visualize trends over time.

* Out of Scope:

1. Detailed sub-topic classification within issues.
2. Prediction or machine learning-based forecasting for future sentiment.

3. Technology Stack

1. Data Collection: YouTube Data API v3 for accessing video metadata and comments.
2. Data Processing: PySpark for handling real-time data ingestion and transformations.
3. Programming Language: Python for API interactions, sentiment analysis, and data manipulation.
4. Sentiment Analysis Tools: VADER and TextBlob for sentiment scoring and keyword extraction.
5. Database: MySQL or SQLite for storing processed comments and metadata.
6. Visualization and Dashboard: Plotly or Dash for an interactive web-based dashboard.
7. Deployment Platform: Consider local deployment with options for cloud-based deployment if needed.

4. Detailed Implementation Phases

* Phase 1: Project Setup (1 Week)

Step 1: Register for the YouTube Data API through the Google Developer Console.

Step 2: Set up a local development environment: Install necessary Python libraries: PySpark, Pandas, VADER, TextBlob, Plotly, and Dash. Configure PySpark for distributed data processing.

Step 3: Create a GitHub repository for version control and documentation. Set up a README.md with an initial project overview and environment setup instructions.

* Phase 2: Data Collection and Storage (1-2 Weeks)

Step 1: Define the data schema for storing comments: Comment ID, Text, Timestamp, Video ID, Sentiment Score, Keyword, and Engagement Metrics (likes, replies).

Step 2: Create a script to connect to the YouTube Data API: Use keywords to search for videos related to specific issues. For each video, retrieve comments in real-time and store the raw data in a staging table within a database.

Step 3: Set up an ETL (Extract, Transform, Load) process in PySpark to clean and preprocess the data: Remove non-relevant comments (e.g., those without keywords).Filter based on recency and engagement metrics to prioritize popular comments.Store cleaned data in a primary table for sentiment analysis.

* Phase 3: Sentiment Analysis and Data Processing (2 Weeks)

Step 1: Develop a PySpark job for sentiment analysis:Use VADER for sentiment classification, categorizing comments as positive, neutral, or negative.Enhance analysis with TextBlob for keyword extraction, allowing insights into common themes and terms.

Step 2: Process the analyzed data: Aggregate sentiment scores by time intervals (e.g., hourly or daily) for each topic. Calculate keyword frequency and sentiment trends over time.

Step 3: Store processed results in the database with indexes on common query fields (e.g., timestamps, issue keywords).

* Phase 4: Dashboard Development and Visualization (1-2 Weeks)

Step 1: Design a user interface with Plotly or Dash:

Visual Elements:

Sentiment Distribution: A pie chart or bar graph displaying positive, neutral, and negative sentiment by topic.

Time-Based Sentiment Trends: A line chart showing sentiment trends for each topic over a defined period.

Keyword Frequency: A word cloud or bar chart showing the most discussed keywords for each topic.

Step 2: Add filtering and interaction capabilities:

Enable filtering by issue category, sentiment type, and date range.

Include tooltips or interactive data points to show detailed comment data when hovering over charts.

Step 3: Integrate data fetching with real-time updates:

Set up a refresh function to pull the latest sentiment data and update the dashboard at defined intervals (e.g., every 15 minutes).

Phase 5: Testing and Validation (1 Week)

Step 1: Conduct unit testing for each component (data ingestion, sentiment analysis, and visualization).

Step 2: Validate sentiment analysis accuracy with a sample dataset, checking for consistent and logical sentiment scoring.

Step 3: Test the dashboard for responsiveness and performance, especially under varying data loads.

Step 4: Collect user feedback to improve interface usability and functionality.

Phase 6: Documentation and Final Report (1 Week)

Step 1: Create detailed documentation:

API setup and configuration guide.

Explanation of each module (data collection, processing, analysis, visualization).

Usage instructions for running the dashboard locally or on a server.

Step 2: Write a final project report:

Summarize the project goals, methodologies, and results.

Include insights gained from the sentiment analysis and keyword trends.

Highlight potential applications and limitations.

5. Timeline

Total Duration: 8 weeks

Week 1: Setup and configuration

Weeks 2-3: Data collection and storage setup

Weeks 4-5: Sentiment analysis and processing

Weeks 6-7: Dashboard development and visualization

Week 8: Testing, documentation, and final reporting

6. Final Deliverables

Real-Time Data Pipeline: A PySpark-based pipeline that collects and processes YouTube comments.

Interactive Dashboard: A Plotly/Dash-based web interface that displays real-time sentiment trends, keyword frequency, and comment insights.

Documentation:

README file with setup and usage instructions.

Final project report summarizing the methodology, results, and insights.

7. Potential Applications

For Advocacy Groups: Understanding public sentiment on current social issues, helping refine campaigns and public messaging.

For Policymakers: Real-time feedback on recent legislative changes or policy announcements.

For Media and PR: Tracking public opinion on specific news topics, supporting timely and data-informed responses.

For Researchers: Analyzing trends in public discourse to identify shifts in sentiment on key societal issues.