LABORATORY REPORT

Application Development Lab
(CS33002)

B.Tech Program in ECSc

Submitted By

Name:-Pruthibiraj Nayak

Roll No: 2230183

Kalinga Institute of Industrial Technology
(Deemed to be University)
Bhubaneswar, India

Spring 2024-2025

Table of Content

Experiment Number	8
Experiment Title	Sentiment Prediction API Using FastAPI and Youtube -Comments
Date of Experiment	26.03.2025
Date of Submission	31.03.2025

1. Objective:-

The objective of this lab experiment is to create a sentiment prediction API using FastAPI, which analyzes Twitter tweets for positive, negative, or neutral sentiment. This lab integrates natural language processing (NLP) techniques with a lightweight and high- performance API framework.

2. Procedure:- (Steps Followed)

1. Set up FastAPI Backend:

Install required libraries: pip install fastapi google-api-python-client textblob uvicorn Create a FastAPI server (main.py) to: Fetch YouTube comments using the Google API.Perform sentiment analysis on each comment using TextBlob.

2. Implement Sentiment Analysis:

The sentiment analysis is performed using TextBlob. The polarity score of each comment determines whether it is positive, negative, or neutral.The backend exposes an endpoint /sentiment/{video_id} to fetch and analyze comments from a YouTube video.

3. Create Frontend (HTML/CSS):

Create an HTML file (index.html) to: Take YouTube Video ID as input.Display the sentiment results categorically (Positive, Neutral, Negative).Use CSS to style the categories and improve UI.Use JavaScript to fetch data from the FastAPI backend and display the results dynamically on the frontend.

4. FastAPI Server Configuration:

Serve the HTML file using FastAPI's built-in file response system.Use the Uvicorn ASGI server to run the FastAPI application: uvicorn main:app --reload

5. Run the Application:

Start the FastAPI server - uvicorn main:app --reload. Visit http://127.0.0.1:8000/ in the browser to access the web interface.Enter a YouTube Video ID and click Analyze to view categorized comments based on sentiment.

Code:-

FLASK CODE

main.py

```
from fastapi import FastAPI
from fastapi.responses import FileResponse, JSONResponse
from googleapiclient.discovery import build
from textblob import TextBlob
from fastapi.staticfiles import StaticFiles

app = FastAPI()

# Mount static folder to serve HTML and CSS
app.mount("/static", StaticFiles(directory="static"), name="static")

API_KEY = "AIzaSyA5k8WuyYP_PUxTS9FD4vIKEGTwrTJ-onQ"  # Replace with your API
key

# Function to get YouTube comments
def get_youtube_comments(video_id):
    youtube = build('youtube', 'v3', developerKey=API_KEY)
    request = youtube.commentThreads().list(
        part="snippet",
        videoId=video_id,
        textFormat="plainText",
        maxResults=50
    )
    response = request.execute()
    return [item["snippet"]["topLevelComment"]["snippet"]["textDisplay"] for
item in response.get("items", [])]

# Function to analyze sentiment
def analyze_sentiment(comment):
    blob = TextBlob(comment)
    polarity = blob.sentiment.polarity
    if polarity > 0:
        return "positive"
    elif polarity < 0:
        return "negative"
    else:
        return "neutral"

# Serve the HTML page
@app.get("/")
def serve_frontend():
    return FileResponse("static/index.html")
```

```
# API endpoint for sentiment analysis
@app.get("/sentiment/{video_id}")
def sentiment_analysis(video_id: str):
    comments = get_youtube_comments(video_id)
    results = {"positive": [], "neutral": [], "negative": []}

    for comment in comments:
        sentiment = analyze_sentiment(comment)
        results[sentiment].append(comment)

    return JSONResponse(content=results)
```

app.py

```
from googleapiclient.discovery import build

API_KEY = "AIzaSyA5k8WuyYP_PUxTS9FD4vIKEGTwrTJ-onQ"
VIDEO_ID = "rcIA7QAhpH8"

def get_youtube_comments(video_id):
    youtube = build('youtube', 'v3', developerKey=API_KEY)

    request = youtube.commentThreads().list(
        part="snippet",
        videoId=video_id,
        textFormat="plainText",
        maxResults=50  # Adjust as needed
    )

    response = request.execute()

    comments = [item["snippet"]["topLevelComment"]["snippet"]["textDisplay"]
for item in response["items"]]
    return comments

# Example usage
comments = get_youtube_comments(VIDEO_ID)
print(comments)
```

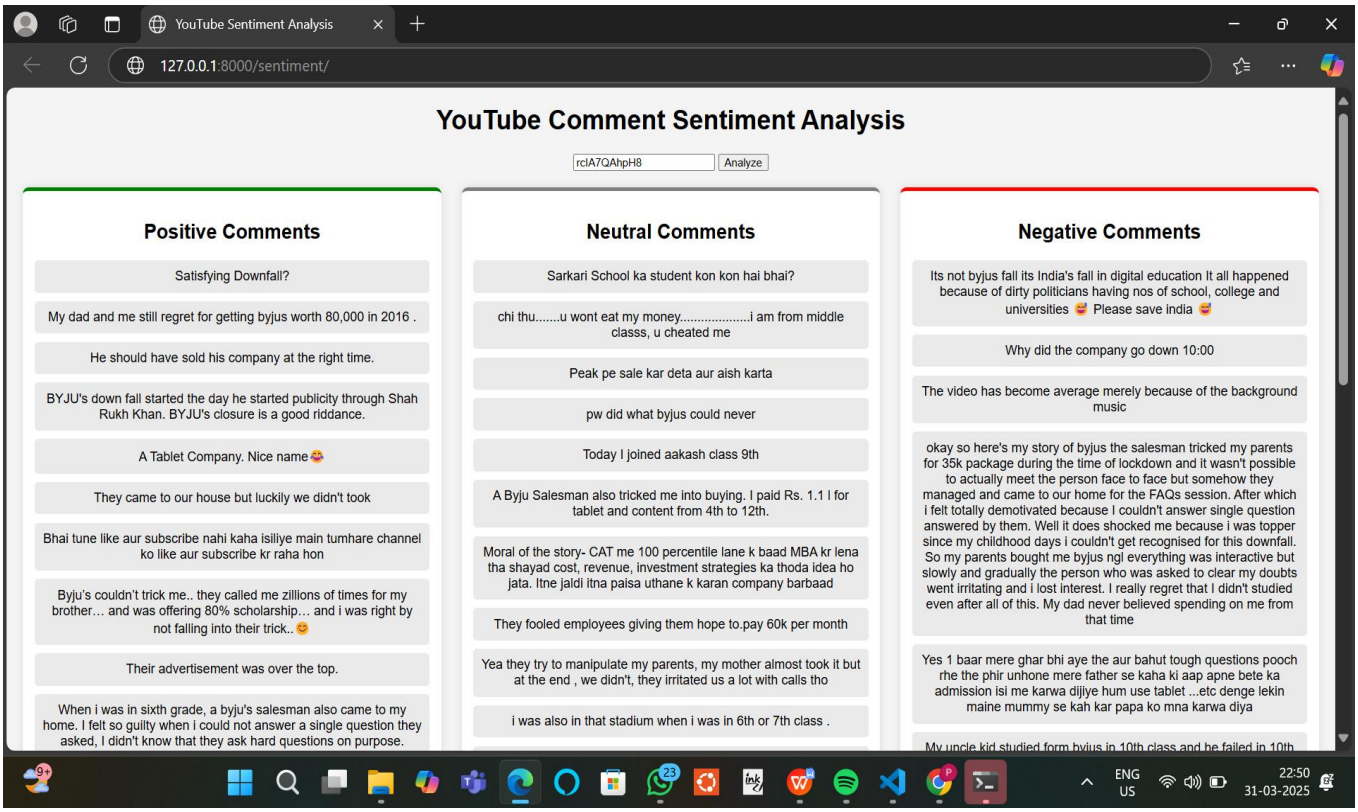
HTML CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>YouTube Sentiment Analysis</title>
    <style>
```

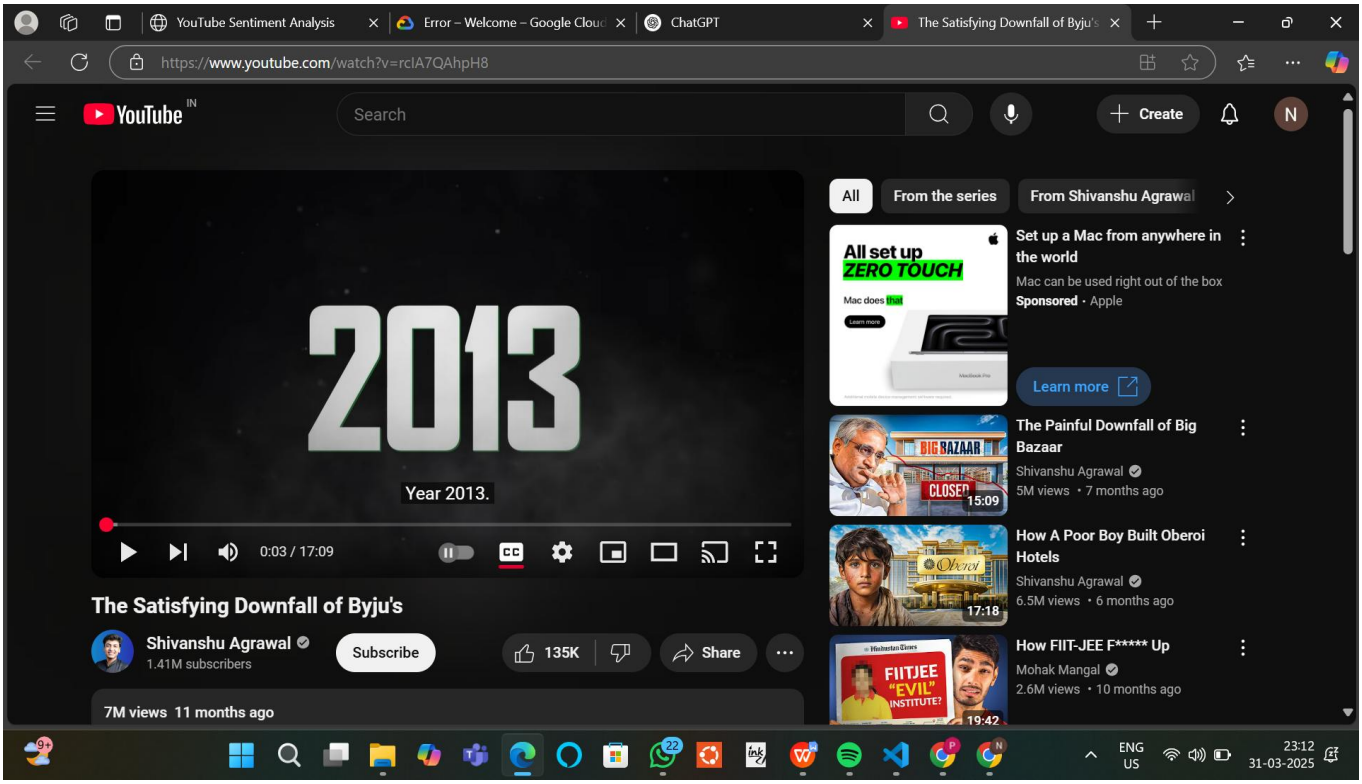
```
        body { font-family: Arial, sans-serif; text-align: center;
background-color: #f4f4f4; }
        h1 { margin-top: 20px; }
        .container { display: flex; justify-content: space-around; margin-top:
20px; }
        .category { width: 30%; background: white; padding: 15px; border-
radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); }
        .positive { border-top: 5px solid green; }
        .neutral { border-top: 5px solid gray; }
        .negative { border-top: 5px solid red; }
        .comment { background: #eaeaea; padding: 10px; margin: 10px 0;
border-radius: 5px; }
    </style>
</head>
<body>
    <h1>YouTube Comment Sentiment Analysis</h1>
    <input type="text" id="videoId" placeholder="Enter YouTube Video ID">
    <button onclick="fetchSentiment()">Analyze</button>

    <div class="container">
        <div class="category positive">
            <h2>Positive Comments</h2>
            <div id="positive-comments"></div>
        </div>
        <div class="category neutral">
            <h2>Neutral Comments</h2>
            <div id="neutral-comments"></div>
        </div>
        <div class="category negative">
            <h2>Negative Comments</h2>
            <div id="negative-comments"></div>
        </div>
    </div>
    <script>
        async function fetchSentiment() {
            const videoId = document.getElementById("videoId").value;
            if (!videoId) return alert("Please enter a Video ID");
            const response = await fetch(`/sentiment/${videoId}`);
            const data = await response.json();
            document.getElementById("positive-comments").innerHTML =
data.positive.map(c => `<div class="comment">${c}</div>`).join("");
            document.getElementById("neutral-comments").innerHTML =
data.neutral.map(c => `<div class="comment">${c}</div>`).join("");
            document.getElementById("negative-comments").innerHTML =
data.negative.map(c => `<div class="comment">${c}</div>`).join("");
        }
    </script>
</body>
</html>
```

3. Results/Output:- Entire Screen Shot including Date & Time



The video taken as input:



4. Remarks:-

In this experiment, we successfully developed a web application using FastAPI that fetches comments from YouTube videos, performs sentiment analysis on each comment using TextBlob, and categorizes the comments into Positive, Neutral, and Negative groups. The application integrates a Python backend to handle data processing and a user-friendly frontend to display the results in a visually appealing manner. By utilizing Google API for YouTube data and combining it with NLP techniques for sentiment analysis, this experiment demonstrated the effective use of modern web technologies for real-time data processing and sentiment evaluation. This approach can be expanded for more complex sentiment analysis applications and real-world use cases.

Pruthibiraj Nayak (2230183)

(Name of the Student) (Name of the Coordinator)