**PROJECT: SENTIMENT ANALYSIS DASHBOARD**

**April 21, 2025**

Table of contents

[Sentiment Analysis and Comment Management System 1](#_Toc196331640)

[1. Introduction 1](#_Toc196331641)

[2. Objectives 1](#_Toc196331642)

[3. Methodology 1](#_Toc196331643)

[4. Implementation 2](#_Toc196331644)

[a. Sentiment analysis 2](#_Toc196331645)

[b. BST node structure 2](#_Toc196331646)

[c. Functionalities implemented 3](#_Toc196331647)

[d. Menu System 7](#_Toc196331648)

[5. Conclusion 9](#_Toc196331649)

[Limitations: 9](#_Toc196331650)

[Future improvements: 9](#_Toc196331651)

[6. References 9](#_Toc196331652)

# Sentiment Analysis and Comment Management System

## 1. Introduction

In the modern digital landscape, user-generated content is vital in shaping public opinion and decision-making. Understanding the sentiment behind these contributions is essential for platforms aiming to gauge user satisfaction. This project presents a C++ based system for analyzing and managing user comments using sentiment analysis and Binary Search Tree (BST) structures. It supports operations such as ranking comments by sentiment, user-specific comment searches, and basic sentiment statistics.

## 2. Objectives

The main objectives of this project were:

* To develop a sentiment analysis program that evaluates user comments based on a predefined set of positive and negative words.
* To store and manage comments efficiently using a BST based on sentiment scores.
* To implement user-friendly operations such as adding, deleting, and displaying comments.
* To provide statistical insights such as top comments, average sentiment scores, and listing users.

## 3. Methodology

The project was implemented using the following methodologies:

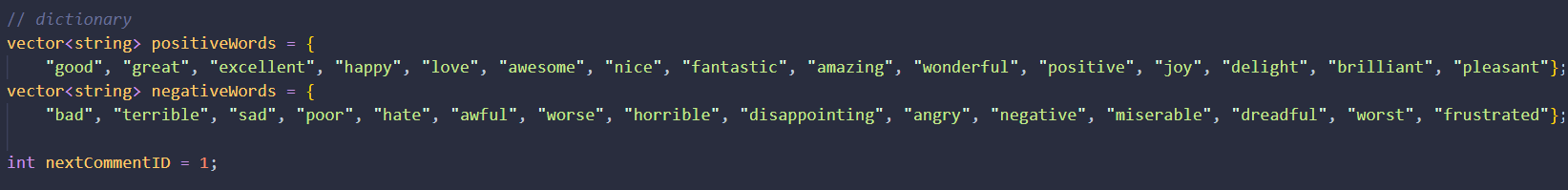
* **Sentiment Lexicon-Based analysis:** A simple dictionary-based sentiment scoring method using predefined vectors of positive and negative keywords. Each comment's sentiment score is calculated by counting positive words as +1 and negative words as -1.
* **Binary Search Tree (BST):** Comments are stored as nodes in a BST, where each node contains the comment ID, username, text, and sentiment score. Insertion is based on the sentiment score, enabling efficient ranking and retrieval.
* **File I/O operations:** The program supports storage by reading from and writing to a file using formatted text records.
* **User interaction:** A console-based menu system facilitates interaction with the system through multiple options, including addition, search, deletion, and viewing statistics.

## 4. Implementation

The project consists of the following major components:

### a. Sentiment analysis

* Sentiment is determined using vectors positiveWords and negativeWords.



* getSentimentScore() function tokenizes and evaluates each word in a comment.

A computer screen shot of text

AI-generated content may be incorrect.

### b. BST node structure

* Each node contains:
  + id (comment’s unique identifier)
  + user (commenter's name)
  + comment (actual text)
  + score (sentiment score)

A computer code on a dark background

AI-generated content may be incorrect.

### c. Functionalities implemented

* **Insertion (insert())**: Adds a new node to the BST.

A computer screen shot of text

AI-generated content may be incorrect.

* **Ranking (displayRanked())**: Displays comments in descending order of sentiment score.

A computer screen shot of a computer code

AI-generated content may be incorrect.

* **Search (searchByUser())**: Retrieves all comments by a particular user.

A screen shot of a computer code

AI-generated content may be incorrect.

* **Deletion (deleteByID())**: Deletes a comment node by its ID.

A computer screen shot of a program code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

* **Storage (saveTree() / loadTree())**: Saves and loads the BST from a file.

A computer screen shot of text

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

* **Top comments (topComments())**: Displays top 5 most positive comments.

A computer screen shot of code

AI-generated content may be incorrect.

* **User listing (collectUsers())**: Displays users alphabetically.

A screen shot of a computer code

AI-generated content may be incorrect.

* **Average score (collectScores())**: Calculates average sentiment per user.

A screen shot of a computer code

AI-generated content may be incorrect.

A computer code on a dark background

AI-generated content may be incorrect.

### d. Menu system

A menu-driven interface allows users to interact with the system by selecting from a range of options, including:

* Add comment
* Display ranked comments
* Search comments by username
* Delete comment
* Show all users
* Show top 5 comments
* Show average sentiment per user

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a program code

AI-generated content may be incorrect.

## 5. Conclusion

This project demonstrates how C++ can be used to implement a basic sentiment analysis and comment management system. The combination of sentiment lexicons and a BST provides a straightforward way to process and manage user-generated text. The system supports both interactive usage and storage, making it suitable for simple applications requiring sentiment-based ranking and analysis.

### Limitations:

* The sentiment analysis uses a simplistic keyword-based approach, which may not capture nuanced meanings or contextual sentiment.
* The use of BST does not handle balancing, which could lead to performance issues in skewed datasets.

### Future improvements:

* Integration of advanced sentiment analysis using natural language processing (NLP) techniques.
* Implementation of self-balancing BST (e.g., AVL Tree or Red-Black Tree).
* Development of a GUI for a better user experience.

## 6. References

* BST Concepts: GeeksforGeeks. Binary Search Tree (BST) <https://www.geeksforgeeks.org/binary-search-tree-data-structure>
* Lexicon-Based Sentiment Analysis: Bing Liu’s Opinion Lexicon
* File I/O in C++: [https://www.learncpp.com/](https://www.learncpp.com/" \t "_new)
* GitHub link: <https://github.com/tuyisengeaurele/SentimentAnalysis>