CHAT ANALYZER USING PYTHON

Submitted by

Anant Kansal [240810125006]

Kamesh Kushwah [240810125001]

Vivek Kumar [240810125011]

For the award of the

Post Graduate Diploma in Big Data Analytics (PG-DBDA)

Under the supervision of

Name: Mr. SUNIL KUMAR
Designation: SENIOR PROJECT ENGINEER

CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

Session: AUG-2024 To FEB-2025

INSTITUTIONAL AREA, JASOLA, NEW DELHI – 110025

DECLARATION

We declare that this written submission represents our ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources.

We also declare that we have adhered to all the principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will cause for disciplinary action by the Institute and so evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed. We further declare that if any violation of the intellectual property right or copyright occurs my supervisor and Institute should not be held responsible for the same.

Name: Anant Kansal

Signature:

PRN No.: 240810125006

Place: New Delhi

Date:

Name: Kamesh Kushwah

Signature:

PRN No.: 240810125001

Place: New Delhi

Date:

Name: Vivek Kumar

Signature:

PRN No.: 240810125011

Place: New Delhi

Date:

ACKNOWLEDGEMENT

"Enthusiasm is the feet of all progress, with it there is accomplishment and Without it there are only slits alibis."

Acknowledgment is not a ritual but is certainly an important thing for the successful completion of the project. At the time when we were made to know about the project, it was tough to proceed further as we were to develop the same on a platform, which was new to us. More so, the coding part seemed tricky that it seemed to be impossible for us to complete the work within the given duration.

We really feel indebted in acknowledging the organizational support and encouragement received from the CDAC Delhi.

The task of developing this system would not have been possible without the constant help of our faculty members and friends. We take this opportunity to express our profound sense of gratitude and respect to those who helped us throughout the duration of this project.

We express our gratitude to our supervisor **Mr. Sunil Kumar** for giving his valuable time and guidance to us.

Name: Anant Kansal

Date:

Name: Kamesh Kushwah

Date:

Name: Vivek Kumar

Date:

ABSTRACT

In today's digital age, instant messaging platforms have become a universal mode of communication, generating vast amounts of conversational data. These digital exchanges, often encompassing diverse topics and sentiments, hold a wealth of information ripe for analysis. This project introduces Chat Analyzer, a tool designed to provide in-depth analysis of chat data, irrespective of the specific subject matter discussed. While applicable to various chat sources, this iteration focuses on analyzing data exported from WhatsApp, a widely used messaging application. The core strength of Chat Analyzer lies in its ability to extract meaningful insights from this data, offering a comprehensive understanding of communication patterns and trends. Implemented using accessible Python modules like pandas, matplotlib, seaborn, and sentiment analysis tools, Chat Analyzer efficiently constructs data frames and generates diverse visualizations. These analytical outputs are then seamlessly integrated into a Streamlit Application (streamlit cloud), providing a user-friendly and resource-efficient interface. The lightweight nature of the underlying algorithms ensures that Chat Analyzer can be effectively applied to even the largest chat datasets, making it a powerful tool for understanding the complexities of digital conversations.

TABLE OF CONTENT

LIST OF FIGURES

OBJECTIVE

This project aims to develop a user-friendly chat analyzer for WhatsApp, extracting, analyzing, and visualizing conversation data to provide valuable insights. The analyzer will extract key information like timestamps, senders, and message content. It will analyze individual participant activity, calculating metrics such as message count and active hours. Overall chat statistics, including media messages and peak activity times, will be provided. Word clouds will visually represent frequent vocabulary, and emoji usage will be analyzed. Machine learning techniques will be used for sentiment analysis. Interactive visualizations will enable users to explore patterns and trends, offering a deeper understanding of WhatsApp conversations.

CHAPTER 1
INTRODUCTION

Instant messaging platforms like WhatsApp generate vast amounts of chat data, offering valuable insights into communication dynamics. Manually analyzing this data is impractical, so this project developed a user-friendly chat analyzer for WhatsApp exports. This tool empowers users to comprehensively understand conversations beyond simple reading. By automating analysis, it conveniently extracts key information, identifies trends, and visualizes patterns. Features include extracting message content, timestamps, and sender information for detailed participant activity analysis. The analyzer calculates metrics like message frequency and active hours, revealing individual contributions. Overall chat statistics are provided, highlighting peak activity and common vocabulary via word clouds. Emoji analysis reveals emotional expressions, and sentiment analysis (using machine learning) assesses the overall emotional tone. Interactive visualizations present results dynamically and intuitively. This project bridges the gap between raw chat data and meaningful insights, providing a valuable tool for understanding digital communication's complexities. It allows users to explore individual contributions, group dynamics, and emotional trends. The analyzer simplifies the process of gaining a deeper understanding of WhatsApp conversations. It offers a convenient and efficient way to extract key information from chat logs. The focus is on providing actionable insights through clear and concise visualizations.

CHAPTER 2
METHODOLOGY

This project developed a Chat Analyzer by processing exported logs to extract meaningful insights from WhatsApp conversations. The raw chat logs were exported from WhatsApp in .txt format. Initially, the preprocessing phase involved using regular expressions for pattern matching to separate timestamps and messages. This allowed for the structured extraction of timestamp and message content from the unstructured raw text. Group notification messages were identified and removed to focus on the actual conversation content. Feature engineering was then applied to generate new columns, such as the day, date, and hour, extracted from the timestamp. This transformation allowed for deeper analysis of communication patterns across different times of the day and days of the week. Unnecessary columns were dropped to streamline the dataset and retain only the essential features for analysis. Following preprocessing, common English stop words were removed to focus on more informative terms. Feature extraction was conducted by calculating word frequencies to identify the most common vocabulary used in the conversation. Additionally, emoji occurrences were counted to gauge emotional expression in the chat. Participant-specific metrics were calculated, such as message count and active hours, based on the timestamp data. Visualizations were then generated to represent the analysis: word clouds displayed the frequent words, revealing dominant themes, bar charts compared message frequency per participant, and line graphs illustrated chat activity over time, showing communication patterns. The project was implemented in Python, leveraging nltk for sentiment analysis and matplotlib and seaborn for visualizations. This analyzer provides clear, actionable insights into individual and group communication dynamics, simplifying the understanding of complex WhatsApp conversations through comprehensive visual representations of chat behaviour.

CHAPTER 3
REQUIREMENTS SPECIFICATIONS

A. Software Requirements

1. **Operating System:**
   - Windows or Linux
2. **Programming Language:**
   - Python
3. **Required Libraries:**
   - pandas, re(regex)
   - streamlit
   - matplotlib, seaborn
   - emoji, wordcloud
   - preprocessor, helper
   - urlextract, collections
4. **Development Environment:**
   - Jupyter Notebook
   - Pycharm
   - Streamlit Cloud use as a deployment environment.

B. Hardware Requirements

1. **Processor:**
   - Intel i3 processor or equivalent (or better)
2. **Memory (RAM):**
   - 4GB RAM or more
3. **Storage:**
   - Adequate storage space to accommodate:
     - Chat log files (e.g.: WhatsApp, size will vary depending on user data)
     - Python scripts and project files
     - Any intermediate data generated during analysis

CHAPTER 4
FLOW CHART (Chat Analyzer Project Flow)

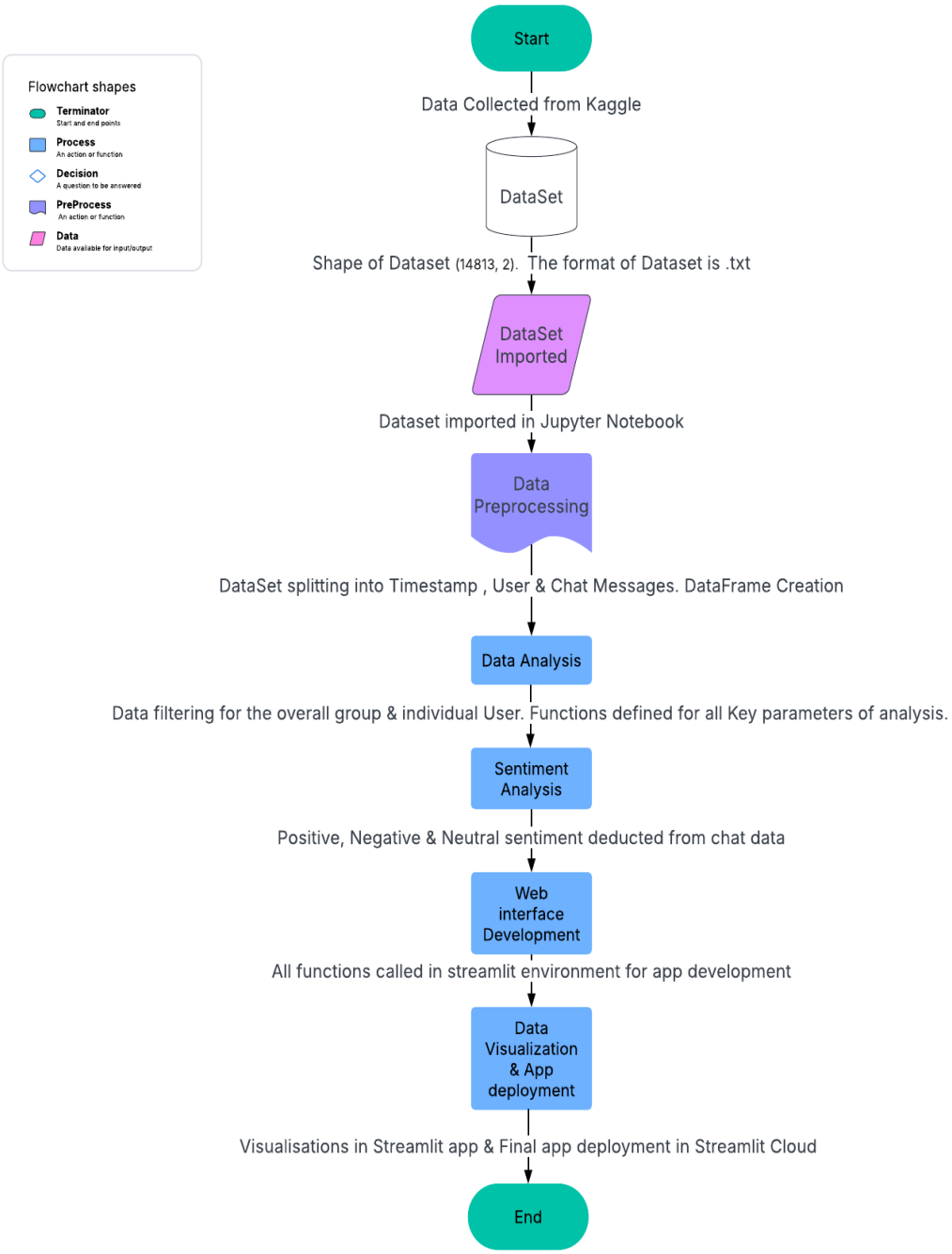

Fig. 1 Flow Chart

CHAPTER 5
DATA FLOW DIAGRAM

A. Activity Diagram: -

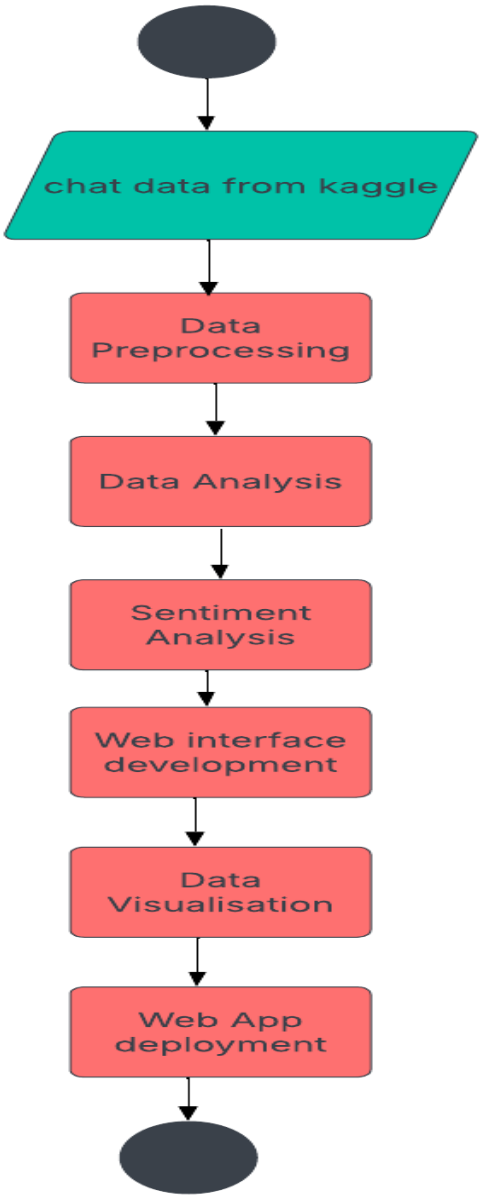

Fig. 2 Activity Diagram

B. Gantt Chart: -

This Gantt chart outlines the project timeline, detailing key tasks and their durations. It begins with initiation and ideation, followed by literature review and synopsis creation. Data collection and preprocessing occur alongside initial code implementation. Subsequent phases include code development, UI deployment, debugging, and report writing, culminating in a final code review and project presentation. The chart visually represents the project's progression, illustrating task dependencies and scheduling.

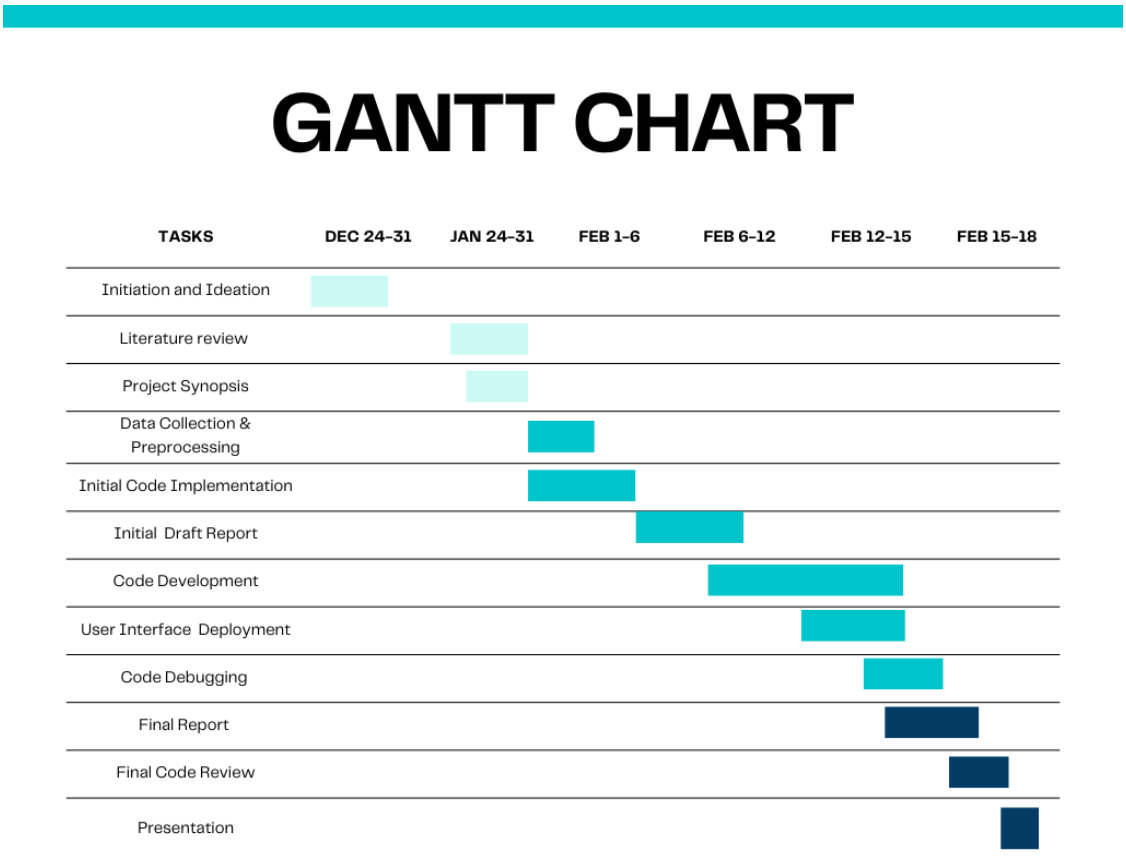

Fig. 3 Gantt Chart

CHAPTER 6
TECHNICAL CODE

This project analyzes Chats using Python and Streamlit, providing insights into communication patterns. It preprocesses data, extracts key features, and calculates metrics. Text and emoji analysis reveal themes and emotions. Time series and activity mapping visualize trends and behaviour. The Streamlit app enables data exploration, with potential for cloud deployment. Let's break down the WhatsApp Chat Analyzer project into technical aspects suitable for a project report, with each point containing three lines of explanation:

1. **Data Acquisition and Preprocessing:** Chat logs are acquired as text files uploaded by the user through the Streamlit interface. The preprocess function uses regular expressions to parse the text, separating messages from timestamps and user information. Data cleaning includes handling mismatched message/date counts and converting timestamps to datetime objects for analysis.
2. **Feature Engineering:** From the datetime objects, new features are extracted, including year, month, day, hour, minute, day of the week, and month name. A "period" feature is engineered to represent time slots (e.g., "10-11") for activity heatmap visualization. These features enhance the analysis by enabling time-based aggregations and activity pattern identification.
3. **Statistical Analysis:** The fetch_stats function calculates key metrics like total messages, word count, media messages, and shared links for overall and user-specific analysis. Most_busy_users identify top contributors based on message count, providing insights into group dynamics. These statistics offer a quantitative overview of chat activity and user engagement.
4. **Text Processing:** Stop words (from stop_hinglish.txt) are removed to focus on meaningful words in word cloud and frequency analysis. Create_wordcloud generates a visual representation of most frequent words, highlighting dominant themes in the conversation. Most_common_words identify the top 20 most frequent words after stop word removal, providing further textual insights.

5. **Emoji Analysis:** Emoji_helper extracts emojis from messages and counts their occurrences to understand emotional tone. The most frequent emojis are displayed in a DataFrame and visualized as a pie chart. This analysis reveals the prevalence of different emotions expressed in the chat.
6. **Time Series Analysis:** Monthly_timeline and daily_timeline functions aggregate message counts by month and day, respectively, creating time series data. These timelines visually represent chat activity trends over time, revealing peak periods and lulls. Matplotlib is used to plot these timelines, providing a clear picture of temporal patterns.
7. **Activity Mapping:** Week_activity_map and month_activity_map analyze message frequency by day of the week and month, respectively. Activity_heatmap creates a heatmap visualizing message frequency across time periods and days of the week, revealing activity hotspots. These functions help understand recurring activity patterns and user behavior.
8. **User Interface and Visualization:** Streamlit is used to create an interactive web application for data upload, user selection, and analysis display. Matplotlib and Seaborn are used for generating various plots, including line charts, bar charts, heatmaps, and pie charts. The app provides a user-friendly interface to explore chat data and gain insights through visualizations.
9. **Libraries and Tools:** Python libraries like Pandas, NumPy, Streamlit, Matplotlib, Seaborn, Word Cloud, URLExtract, and Emoji are used. Regular expressions are employed for text parsing and data extraction. The project demonstrates the use of data analysis and visualization techniques to understand communication patterns.
10. **Deployment:** The Streamlit app can be deployed on platforms like Streamlit Cloud or other cloud services for broader accessibility. In our case , we deployed the app on Streamlit Cloud it being a free platform. Deployment involves packaging the code, dependencies, and data (if any) and configuring the deployment environment. This step makes the analysis tool available to a wider audience.

CHAPTER 7
RESULT ANAYLSIS

1. Data Visualization

a. Heat Map for Chat Density Hour to Hour: -

This heatmap visualizes weekly chat activity, correlating days of the week with hourly periods. Warmer colors (lighter shades) indicate higher message frequency during specific day-time slots, revealing peak conversation times. For instance, Wednesday & Thursday afternoons and evenings (e.g.13-14, 17-18, 18-19) show a surge in messages. Conversely, cooler colors (darker shades) represent periods of less activity. This visualization helps pinpoint the most active days and times within the chat group, offering insights into communication patterns. It allows for quick identification of when the group is most engaged.

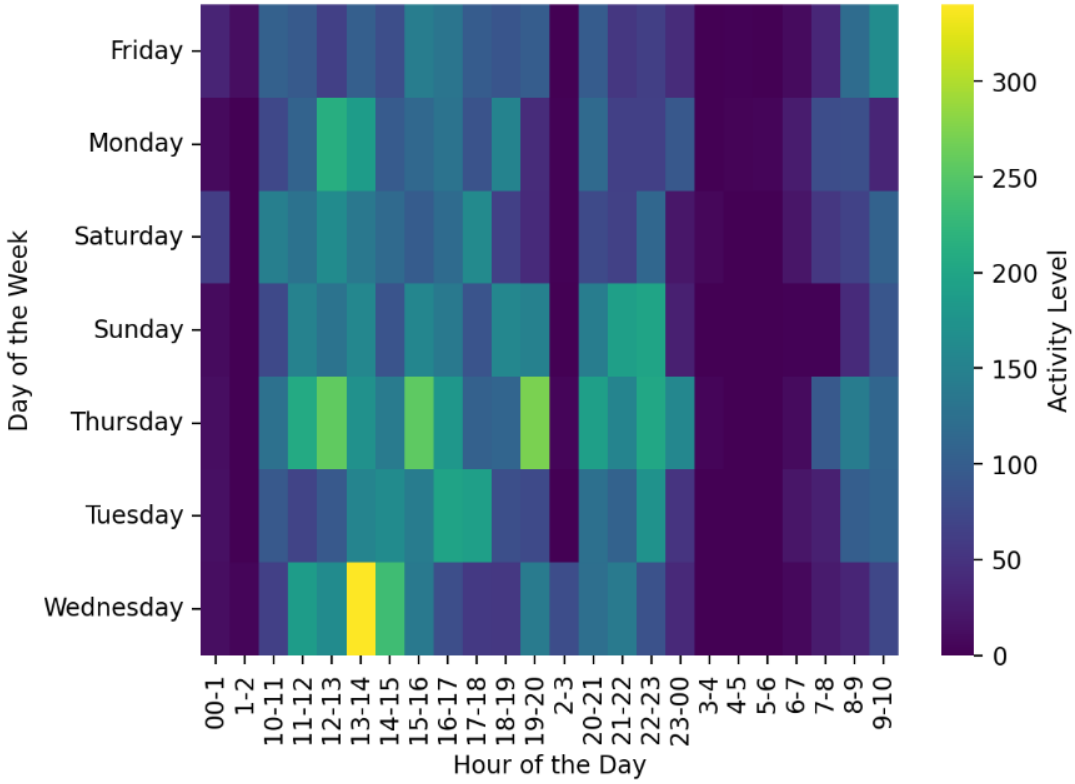

Fig. 4 Heat Map for Chat Density

b. Insights from Monthly Timeline Using Line Chart: -

This line chart depicts the monthly message activity within the chat, spanning from August to December 2024. The vertical axis quantifies the number of messages exchanged, while the horizontal axis represents the progression of months. The green line visually connects the data points, illustrating the fluctuations in message volume over time. A notable surge in activity is evident from August to November, indicating increasing engagement within the chat. The peak message volume is reached in November, suggesting the highest level of interaction during this month. Following the peak, a slight decrease in message activity is observed in December. This timeline provides a clear overview of how chat engagement evolved throughout the five-month period.

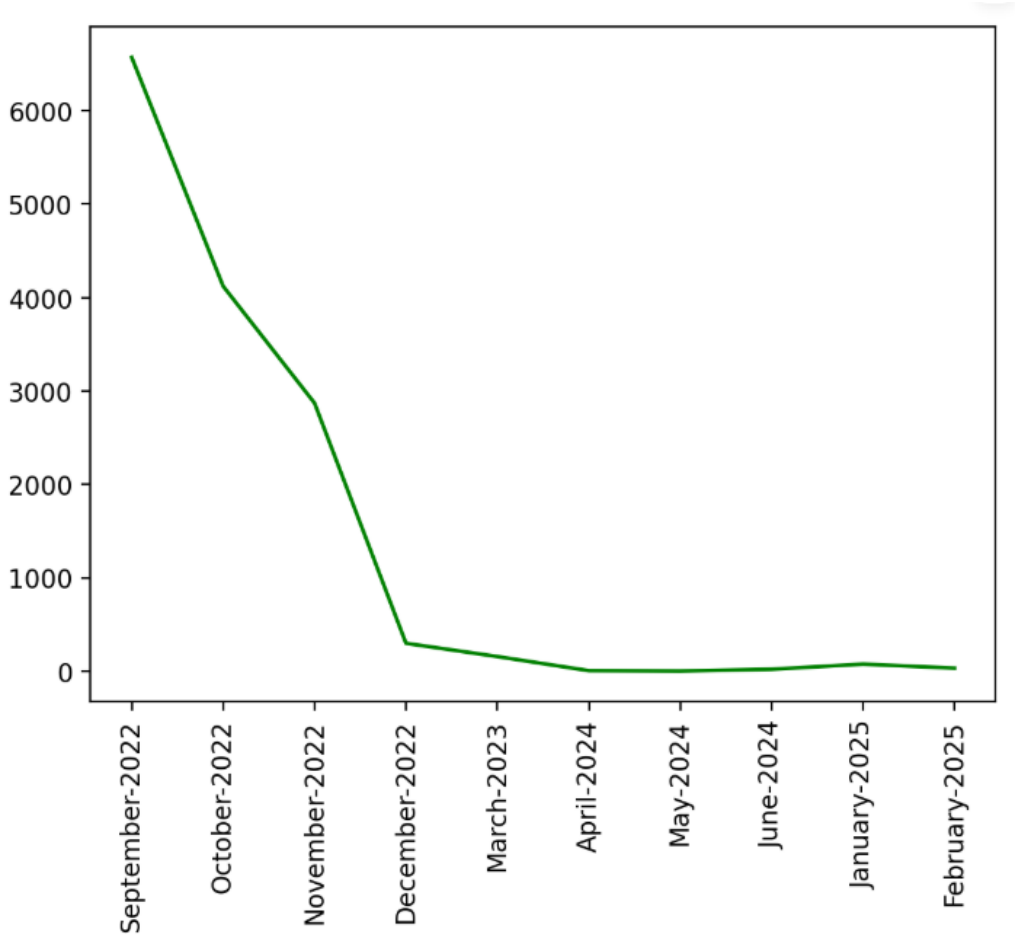

Fig. 5 Line Chart for Monthly Timeline

c. Insights of Most Busy Users by Using Bar Chart: -

This bar chart displays the top contributors to the chat, ranked by their message count. The vertical axisrepresentsthe number of messages, while the horizontal axis lists the users. Each bar corresponds to a user, with its height indicating their message frequency. The chart reveals the most active participants, highlighting their contribution to the overall conversation volume. It provides a clear visual comparison of user activity levels..

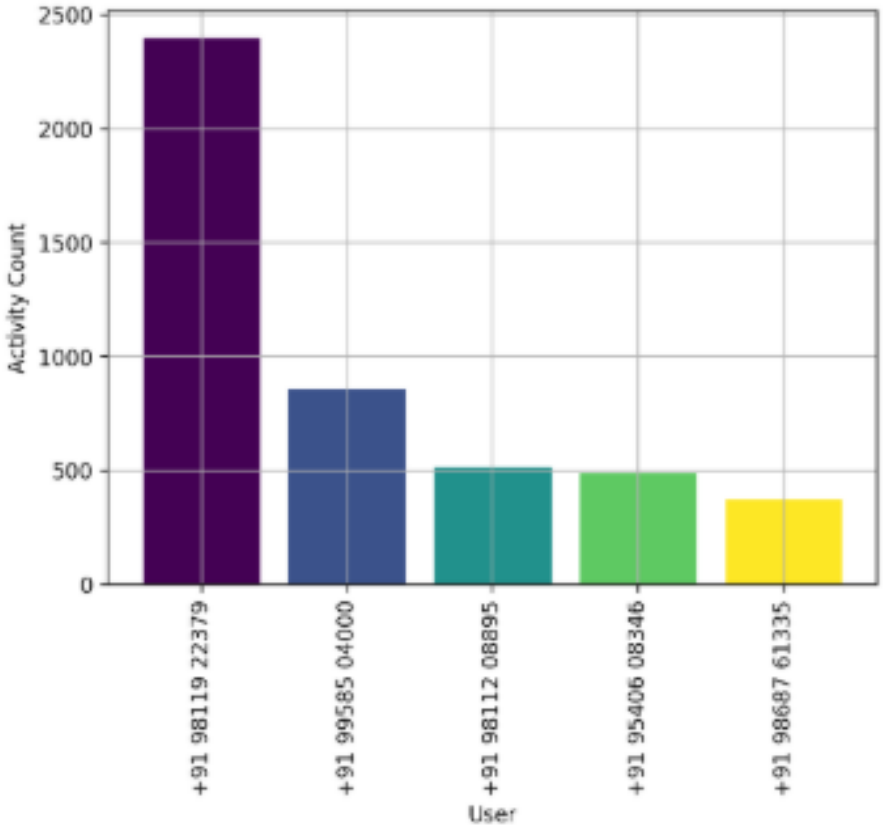

Fig. 6 Bar Graph Chart for Most Busy Users

d. Insights of Most Frequent Words Using Word Cloud: -

This word cloud visually represents the most frequent terms used in the chat conversations. The size of each word reflects its relative importance, with larger words indicating higher frequency. Dominant terms like "done," "map," "power," and "residents" suggest recurring topics or keywords within the discussions. The visualization provides a quick overview of prevalent themes and focal points in the chat. It helps identify the core vocabulary and subjects that dominate the conversation. The arrangement and size of the words offer an intuitive understanding of topic prominence.

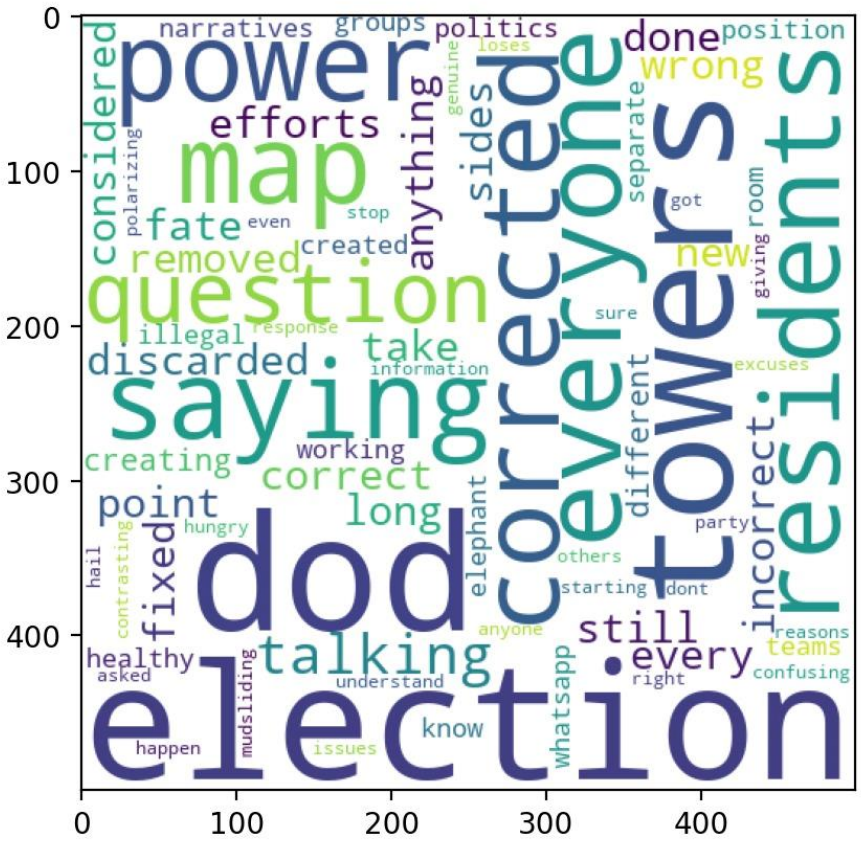

Fig. 7 Word Cloud for Most Frequent Words

e. Insights of Most Common Words Using Horizontal Bar Chart: -

This horizontal bar chart presents the frequency or count of different items related to the project, displayed along the vertical axis. The horizontal axis represents the magnitude or value associated with each item. Each bar's length corresponds to its respective value, allowing for easy comparison between items. The items listed (e.g., "dod", "elections", "power") appear to be keywords or phrases of some relevance. The chart helps visualize the relative importance or occurrence of these items. It provides a clear ranking of items based on their associated values.

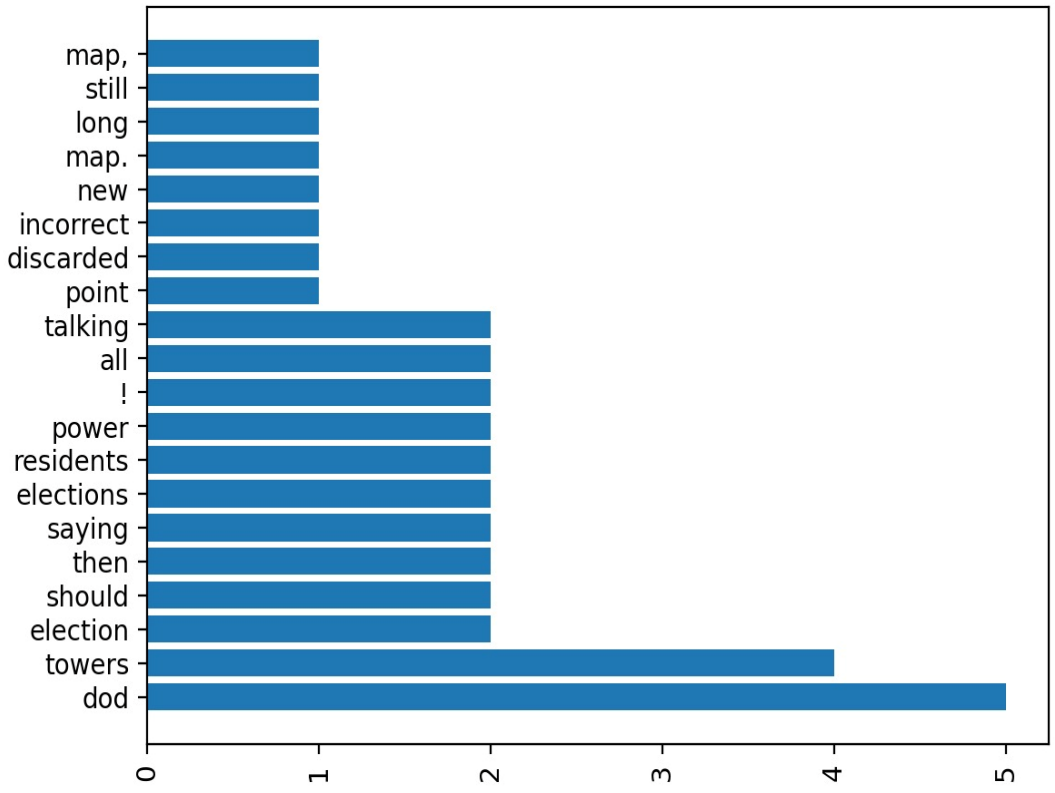

Fig. 8 Horizontal Bar Graph Chart for Most Common Words

f. **Insights of Most Busy Day & Month Using Bar Chart: -**

These bar charts depict chat activity, segmented by day of the week (left) and month (right). The vertical axes represent the message count, indicating the volume of conversations. The “Most Busy Day” chart reveals Thursday as the most active day, while Friday shows the least activity. The “Most Busy Month” chart highlights September as the peak month for communication. These visualizations offer a clear comparison of activity levels across different time frames. They help identify trends and patterns in chat engagement.

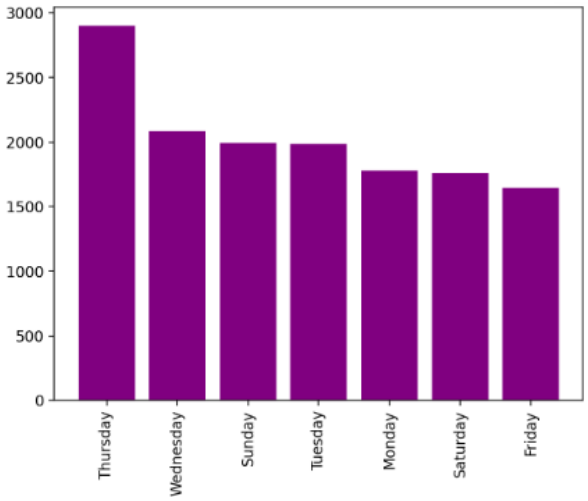

Fig. 9 Bar Graph for Most Busy Day

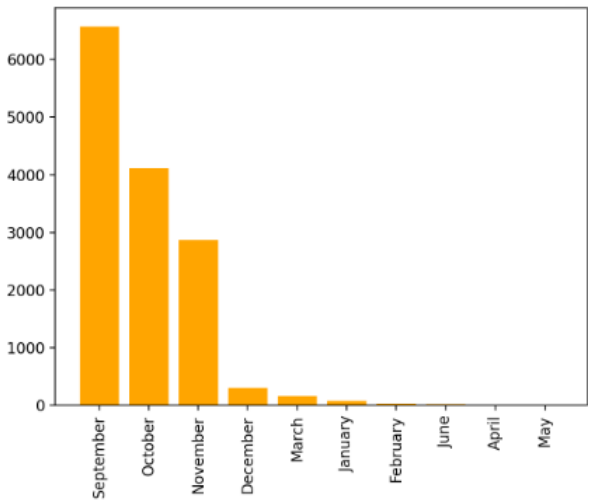

Fig. 10 Bar Graph Chart for Most Busy Month

g. Insights of Emoji Analysis Using Pie Chart: -

This pie chart visualizes the distribution of the top emojis used in the chat conversations. Each slice represents a unique emoji, and its size corresponds to the emoji's relative frequency. The percentages displayed indicate the proportion of each emoji's usage out of the total top emojis. The chart offers a quick overview of the most popular emojis and their relative prevalence. It helps identify the dominant emotional expressions or reactions conveyed through emojis.

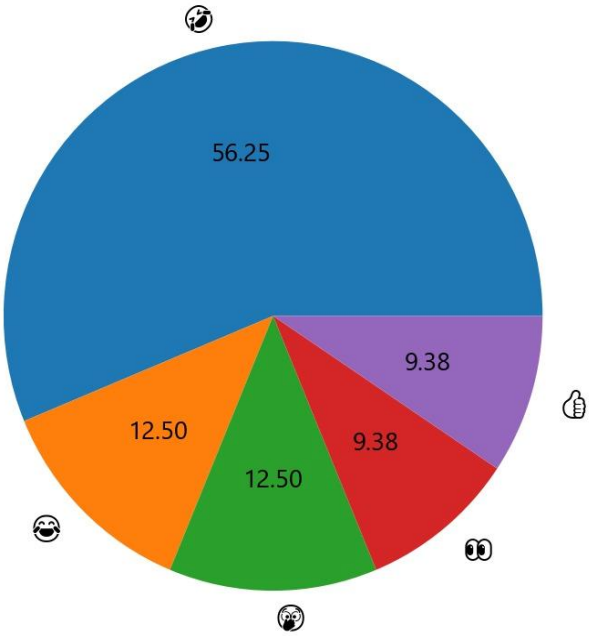

Fig. 11 Pie Chart for Emoji Analysis

h. Insights of Sentiment Score Using Bar Chart: -

This bar chart visualizes the distribution of sentiment scores categorized as positive, negative, and neutral. The vertical axis, labelled "Score," represents the magnitude or count associated with each sentiment. The horizontal axis denotes the sentiment categories. The varying heights of the bars illustrate the relative prevalence of each sentiment. The tallest bar, "Neutral," indicates the most dominant sentiment in the analyzed data. The chart provides a clear comparison of the overall emotional tone present in the text.

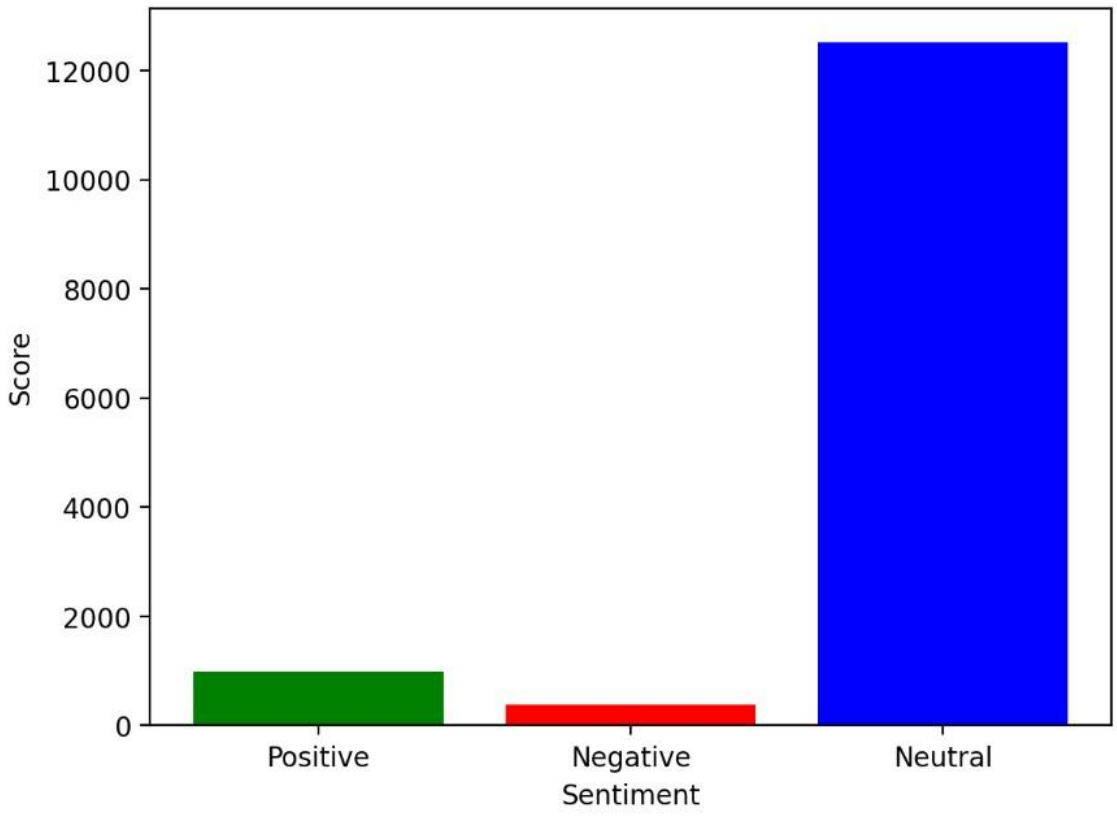

Fig. 12 Bar Graph Chart for Sentiment Score

i. **Insights of Positive and Negative Words Using Word Cloud: -**

These word clouds visually represent the most frequent words used in positive and negative sentiment messages within the chat. The size of each word corresponds to its frequency, with larger words indicating more common occurrences. Dominant terms in the positive cloud, like "resident," "society," and "election," suggest prevalent themes in positive conversations. Similarly, the negative cloud highlights words such as "RWA," "illegal," and "problem," revealing concerns or issues discussed in negative messages. These visualizations provide a quick understanding of the key topics and vocabulary associated with each sentiment. They help identify the emotional drivers and focal points within the chat.

Fig. 13 Word Cloud for Positive Words

Fig. 14 Word Cloud for Negative Words

j. Insights of Sentiments Trend Over Time Using Time Series: -

This time series chart visualizes sentiment trends over time, plotting sentiment scores against date. The horizontal axis represents the date, spanning from September to December, while the vertical axis indicates the sentiment score. Three lines represent positive, negative, and neutral sentiment, showcasing their fluctuations over the observed period. The chart reveals the evolution of each sentiment category, highlighting periods of increased or decreased positivity, negativity, or neutrality. It allows for the identification of temporal patterns and shifts in emotional tone within the chat. The consistent dominance of neutral sentiment is clearly visible.

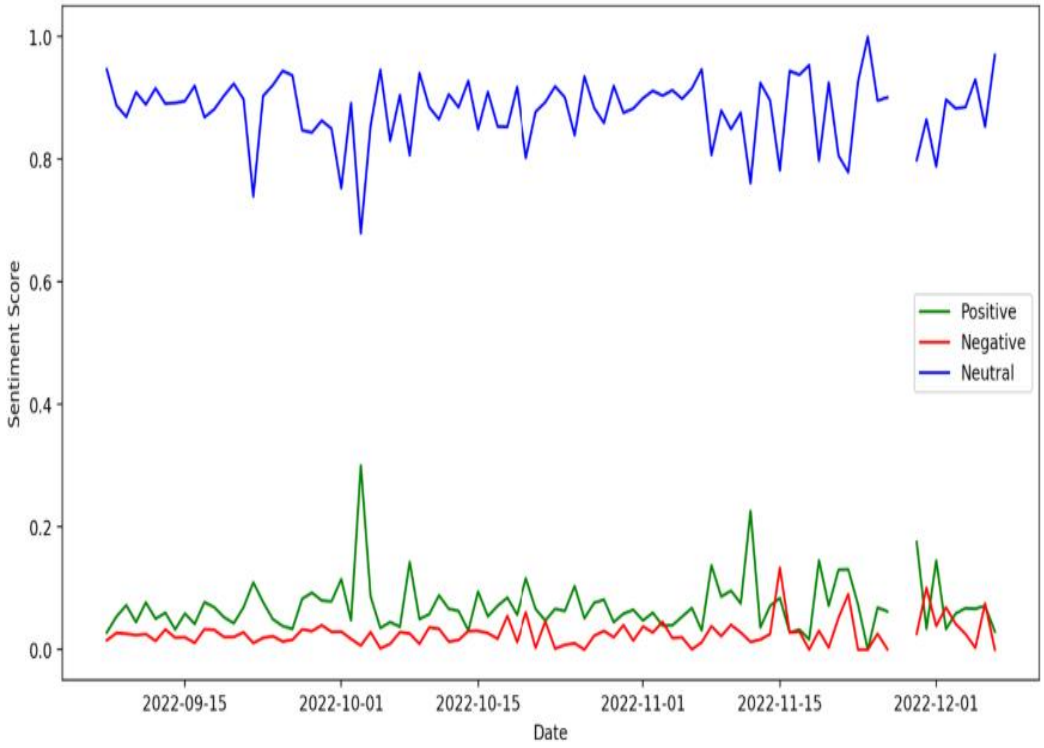

Fig. 15 Time Series Chart for Sentiments Trends Over Time

k. Insights of Web Page User Interface: -

This webpage showcases a "Chat Analyzer" application built using Streamlit, a Python library for creating interactive web apps. The interface is divided into a sidebar on the left and a main content area. The sidebar provides controls for uploading a chat file, selecting a user to analyze, and initiating the analysis. The main content area displays the results of the analysis, including "Top Statistics" and a "Monthly Activity Timeline" chart. The design is clean and user-friendly, with clear headings and intuitive controls, making it easy for users to upload their chat data and explore the insights.

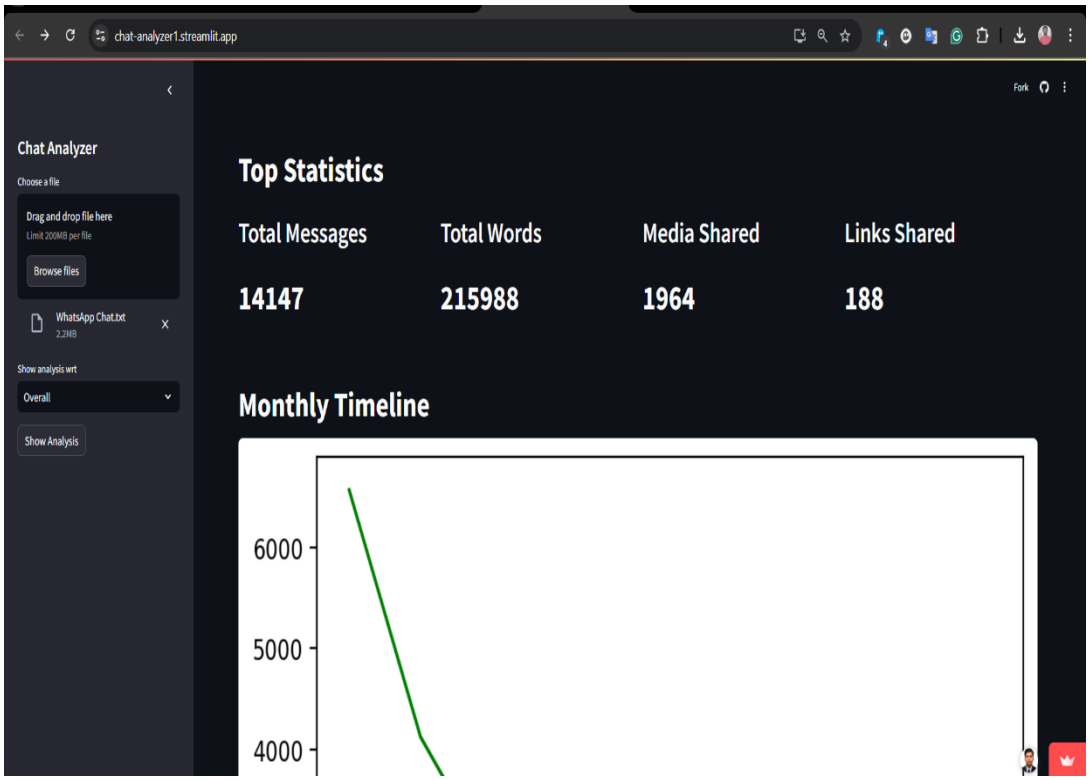

Fig. 16 Streamlit Cloud Web Page Interface

CHAPTER 8
CONCLUSION

This Chat Analyzer project successfully developed a user-friendly tool to extract, analyze, and visualize data from WhatsApp chat exports, providing valuable insights into conversations. The application effectively processes chat logs, extracting key information like timestamps, senders, and message content. It calculates individual and overall chat statistics, including message counts, active hours, media messages, and peak activity times. Furthermore, the integration of sentiment analysis and word cloud generation enhances the understanding of communication dynamics and emotional tone.

The interactive visualizations, including timelines, activity maps, and charts, empower users to explore patterns, trends, and individual contributions within their chats. The intuitive Streamlit interface makes the analysis accessible to users with varying technical backgrounds. By offering a comprehensive suite of analytical tools, this project bridges the gap between raw chat data and actionable insights, facilitating a deeper understanding of digital conversations. The project demonstrates the potential of leveraging data analysis and machine learning techniques to gain valuable knowledge from everyday communication.

In conclusion, this Chat Analyzer serves as a powerful tool for unraveling the complexities of WhatsApp conversations. It provides users with a convenient and informative way to explore their chat data, from individual behavior to overall group dynamics and sentiment. The project's success lies in its ability to transform raw chat logs into meaningful narratives, offering a unique perspective on digital interactions. Future enhancements could include expanding platform compatibility, incorporating advanced NLP techniques, and integrating additional analytical features to further enrich the chat analysis experience.

CHAPTER 9
FUTURE SCOPE

Future enhancements for this Chat Analyzer project could include expanding platform compatibility beyond WhatsApp to encompass other popular messaging apps. Integrating more advanced Natural Language Processing (NLP) techniques, such as topic modelling and named entity recognition, would provide deeper contextual insights. Developing user-defined filters and customizable analysis options would allow for more tailored explorations of chat data. Incorporating machine learning models for predictive analysis, like forecasting conversation trends, could add significant value. Implementing real-time chat analysis capabilities and exploring sentiment analysis at a more granular level (e.g., by message or speaker) are potential extensions. Adding support for multimedia content analysis (images, videos) and enhancing the user interface with interactive dashboards and reporting features would further improve the tool's utility. Finally, exploring API integrations for seamless data import and export would streamline the analysis workflow.

CHAPTER 10
REFERENCES

- McKinney, W. (2012). *Python for data analysis*. O'Reilly Media. (Or the official pandas documentation: https://pandas.pydata.org/docs/).

- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90-95. (Or the Matplotlib documentation: https://matplotlib.org/stable/contents.html).

- Mohammad, A. (2016). *WordCloud for Python*. (This might be cited as software, and you'd find details on how to cite it from the library itself or its repository.)

- Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious lexicon and rule-based sentiment analysis tool for social media text. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2014)*, 2188-2195.

- Gupta, A. (2021). *Streamlit: A Python library for creating interactive web applications*. (You'll need to find the most appropriate way to cite Streamlit - check their website or documentation for citation guidelines).

- Aharony, N., T., G., The Importance of the WhatsApp Family Group: An Exploratory Analysis. "Aslib Journal of Information Management, Vol. 68, Issue 2, pp.1-37" (2016).

- Mike Dickson, "An examination into yahoo messenger 7.0 contact identification", Digital Investigation, ScienceDirect, vol. 3, issue 3, pp. 159-165, 2006.