TALKALYTICS

A PROJECT REPORT
for
Mini Project (KCA353)
Session (2023-24)

Submitted by
Amritesh Kaur

2200290140026

Submitted in partial fulfillment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Ms. Divya Singhal
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(JANUARY 2024)

CERTIFICATE

Certified that **Amritesh Kaur (2200290140026)** has/ have carried out the project work having "**Talkalytics: WhatsApp chat analyzer**" (**Mini Project-KCA353**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Amritesh Kaur
2200290140026

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Ms. Divya Singhal
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Tripathi
Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

Talkalytics is a computer program that helps people understand their WhatsApp conversations better. It works by letting you put in your chat logs from WhatsApp, and then it tells you interesting things about your chats. For example, it can show you how often you or your friends send messages, when you chat the most, and even analyze the words to tell you what topics you discuss the most. You can see all this information in easy-to-read reports and graphs. It's like having a statistics helper for your WhatsApp chats. Plus, it's easy to use and works on most computers. So, if you ever wondered how you and your friends talk on WhatsApp, this program can help you find out!

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Divya Singhal** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi**, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot in many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kinds of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

TABLE OF CONTENTS

Chapter 1: Introduction

In today's world, we use WhatsApp to chat with friends, family, and colleagues. Have you ever wondered what your chats can reveal about your communication style and relationships? This project, Talkalytics is a web application that allows you to do that, to understand your emotions and relations better.

1.1 Project Overview

1.1.1 User friendly interface

Talkalytics has an intuitive and user-friendly interface for easy interaction with the application. You can easily use it and get the results according to the chats that have been uploaded.

1.1.2 Data processing

The application helps implement robust data processing mechanisms to clean and extract relevant information from raw chat data, which helps in deriving the statistics.

1.1.3 Statistical analysis

This application provides users with statistical information such as message frequency, active months, active months and participants. We can also derive the statistics about the words used according to the frequency.

1.1.4 Sentiment analysis

Utilize natural language processing techniques to perform sentiment analysis on messages and determine the most discussed words and topics.

1.2 Technical Details

1.2.1 Tech Stack

- Programming Language: Python
- Database: Real time
- Data Processing: NumPy
- Data Visualization: Pandas, Matplotlib

1.2.2 Implementation

We first developed a web-based interface for easy user interaction. Using NLTK and other libraries for natural language processing that help us with better data processing. After that we employ data visualization libraries for charts and graphs that will aid in making display more interesting. We ensure data security and privacy compliance by not storing the data at any place.

1.3 Challenges and Solutions

Challenges in using this application are:

- Ensuring accuracy in user identification and message categorization.
- Handling large datasets efficiently for optimal performance.

1.4 Future Roadmap

Now Talkalytics is a work in progress that can be enhanced and made better by using more technologies and also the scope of it can be increased when we can import chats from varied platforms and hence making it more popular by providing more client base. Few things we would focus on are: -

- Integration with cloud storage for seamless data import.
- Advanced machine learning models for sentiment analysis.
- Support for multiple chat platforms (Telegram, Facebook Messenger, etc.).

Chapter 2: Design

2.1 Architectural Design

The WhatsApp Chat Analyzer is designed as a web-based application with a client-server architecture. The system comprises three main components:

2.1.1 Client Interface:

This is the user-facing part of the application, developed using HTML, CSS, and JavaScript (React/Vue). It provides a user-friendly interface for uploading chat data, customizing analysis parameters, and visualizing results.

2.1.2 Server Backend:

Implemented using a web framework such as Flask or Django, the server processes user requests, manages data storage, performs data analysis, and serves the processed information to the client.

2.1.3 Database:

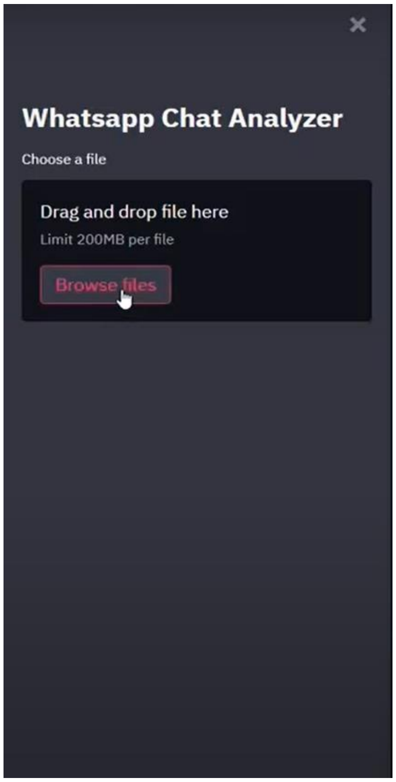
SQLite/MySQL/PostgreSQL is used to store processed chat data, user preferences, and analysis results. The database ensures data persistence and efficient retrieval.

2.2 User Experience (UX) Design

The user interface is designed to be intuitive and visually appealing. Key components include:

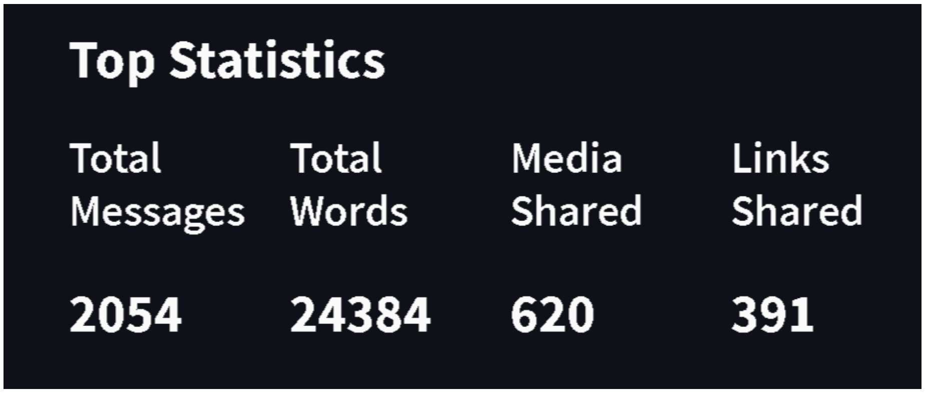
2.2.1 Data Upload Section:

A designated area for users to upload their WhatsApp chat data files, with support for various file formats.



2.2.2 Analysis Configuration:

Customization options for users to apply filters, set date ranges, and choose specific participants for analysis.



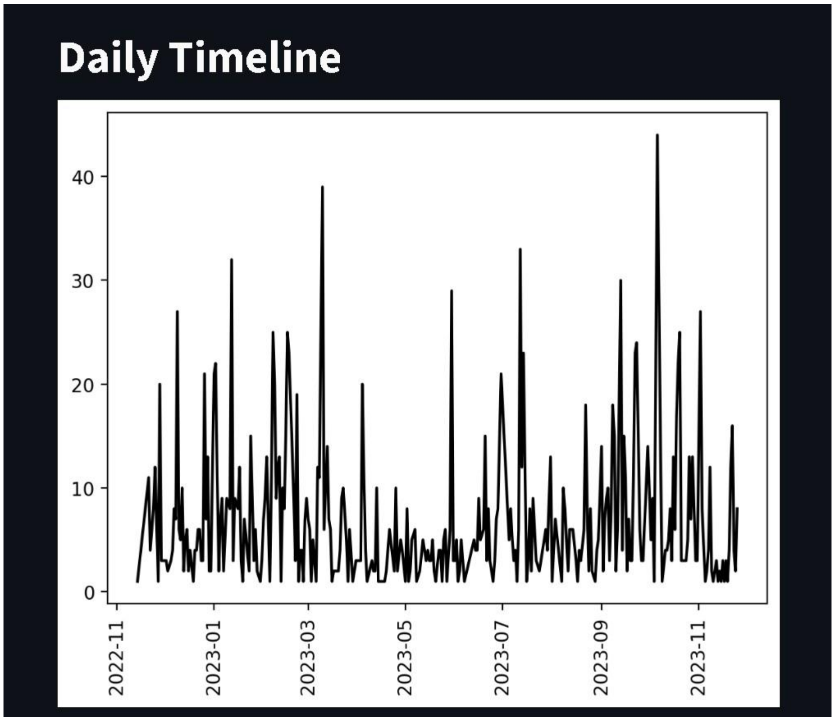
Top Statistics

Total Messages	Total Words	Media Shared	Links Shared
2054	24384	620	391

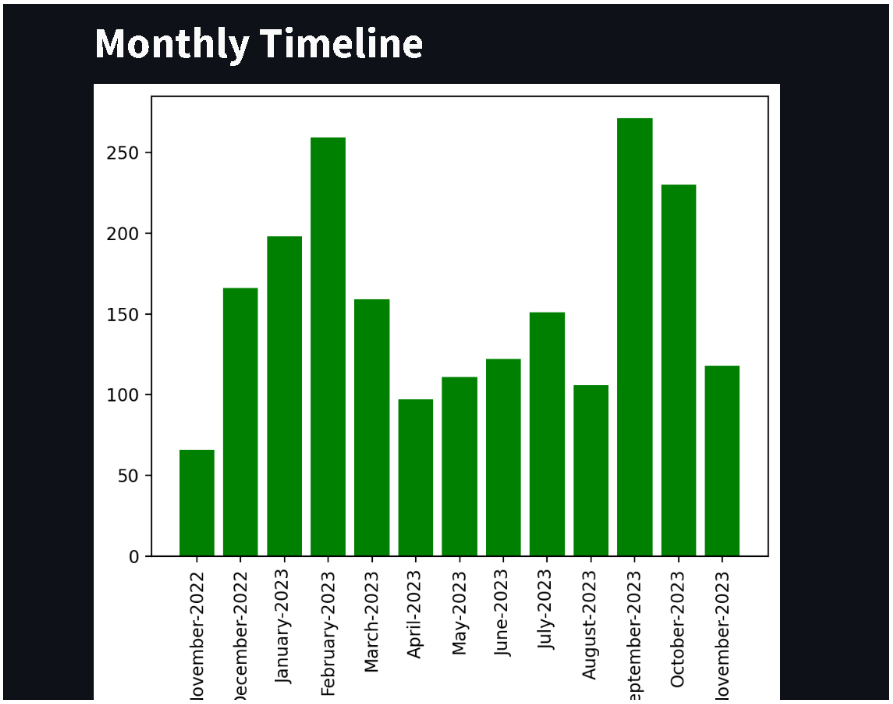
2.2.3 Visualization Section:

Interactive charts, graphs, and tables to present statistical and sentiment analysis results.

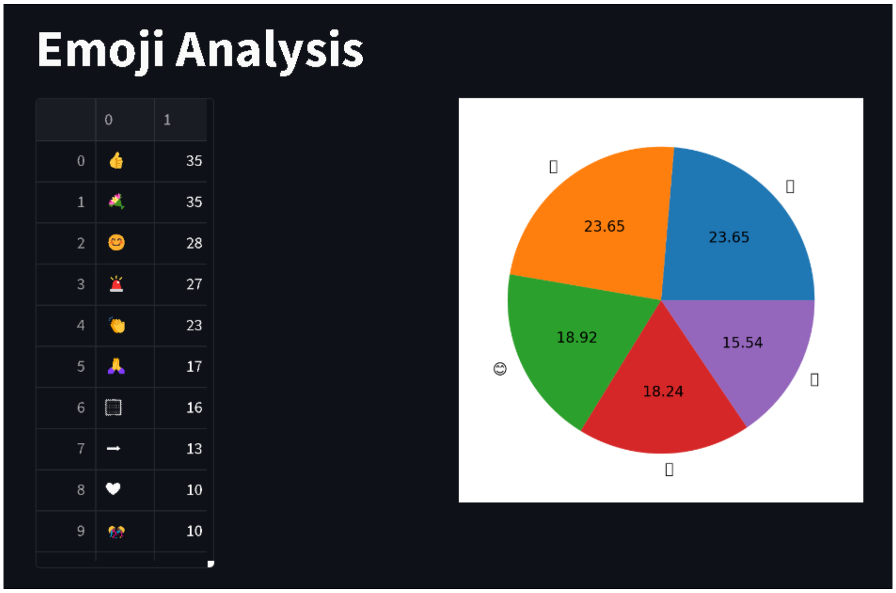
Word clouds and media analysis visualizations provide additional insights.



Daily Timeline

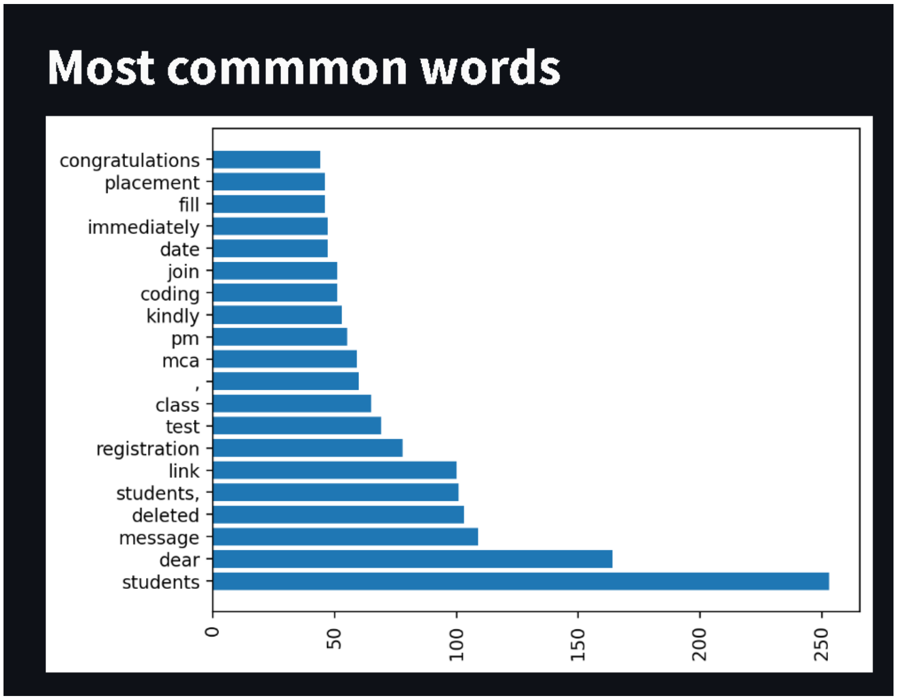


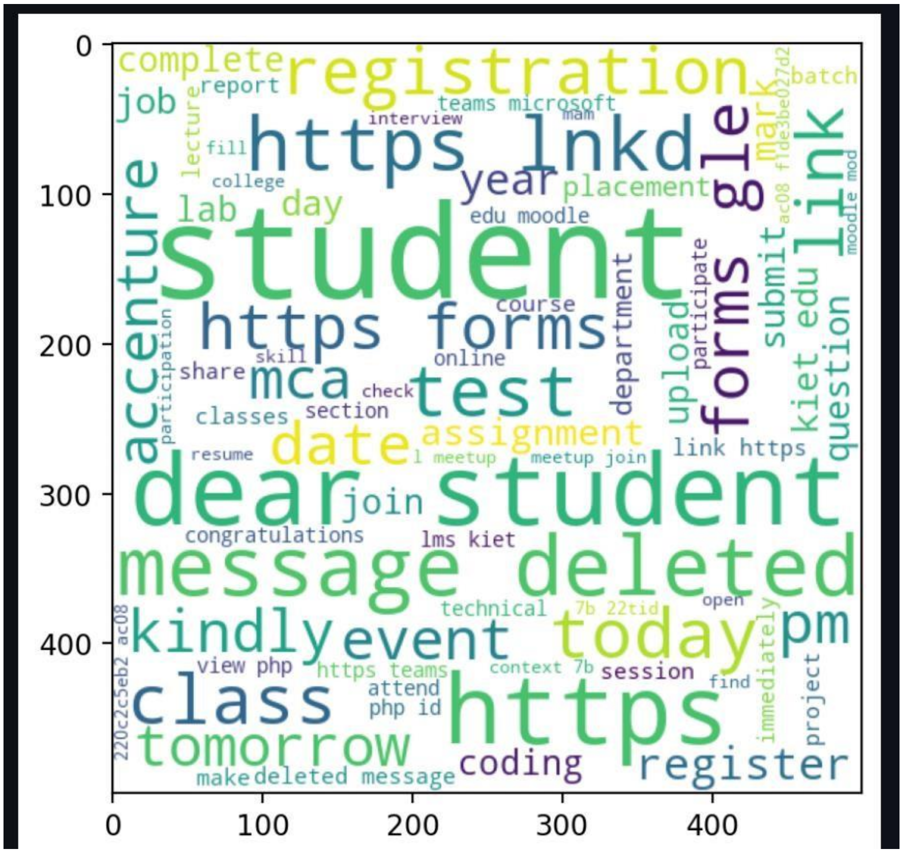
Monthly Timeline

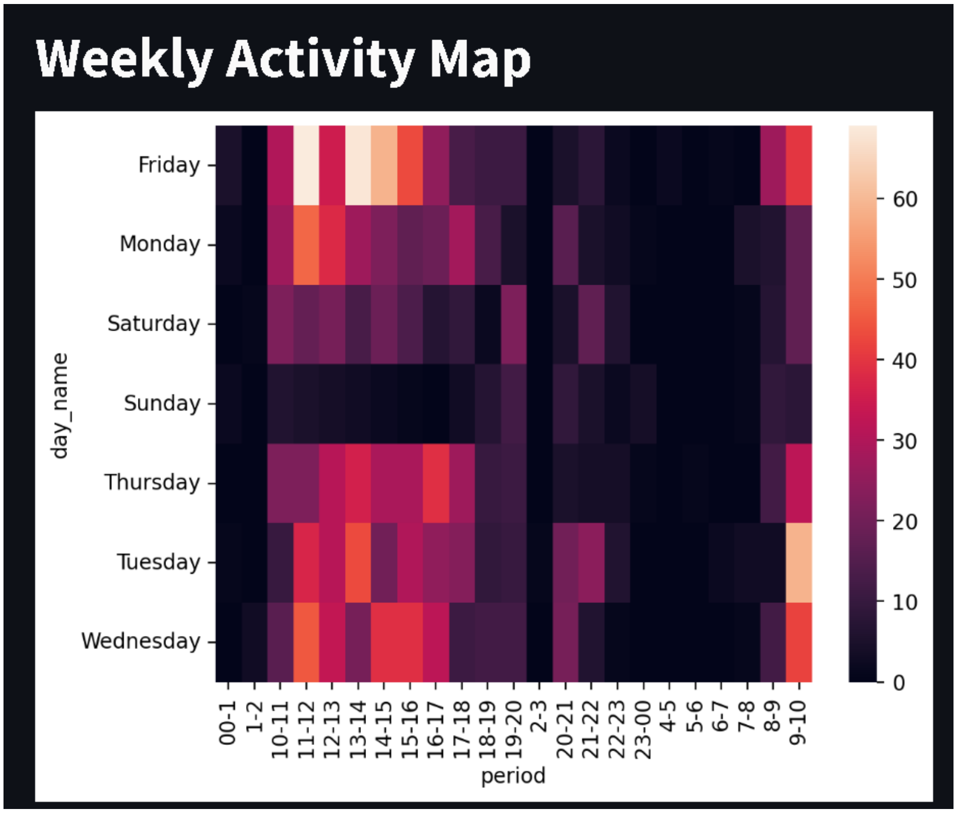


Emoji Analysis

	0	1
0	👍	35
1	💐	35
2	😊	28
3	🚨	27
4	👏	23
5	🙏	17
6	⬚	16
7	➡	13
8	🤍	10
9	👯	10


Most commmon words





2.2.4 User Feedback and Notifications:

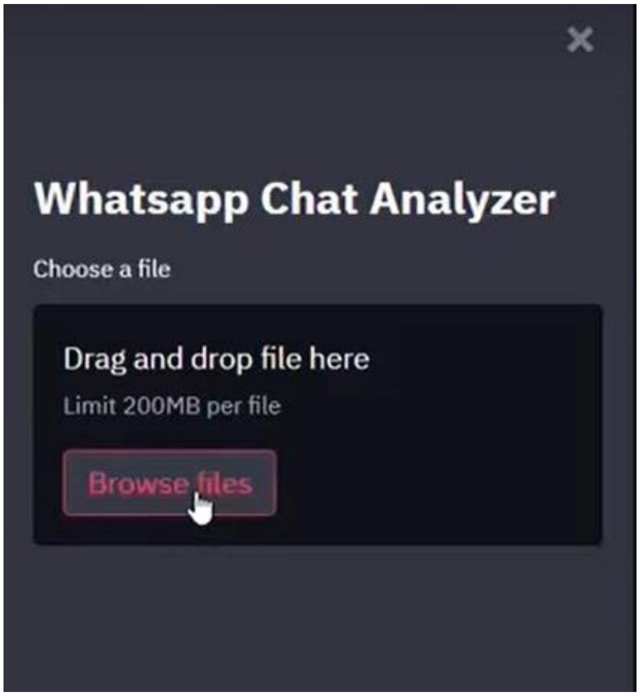
Clear feedback messages and notifications to guide users through the analysis process and inform them of any errors or completion.

2.3 Data processing flow

The design ensures a seamless flow for processing WhatsApp chat data:

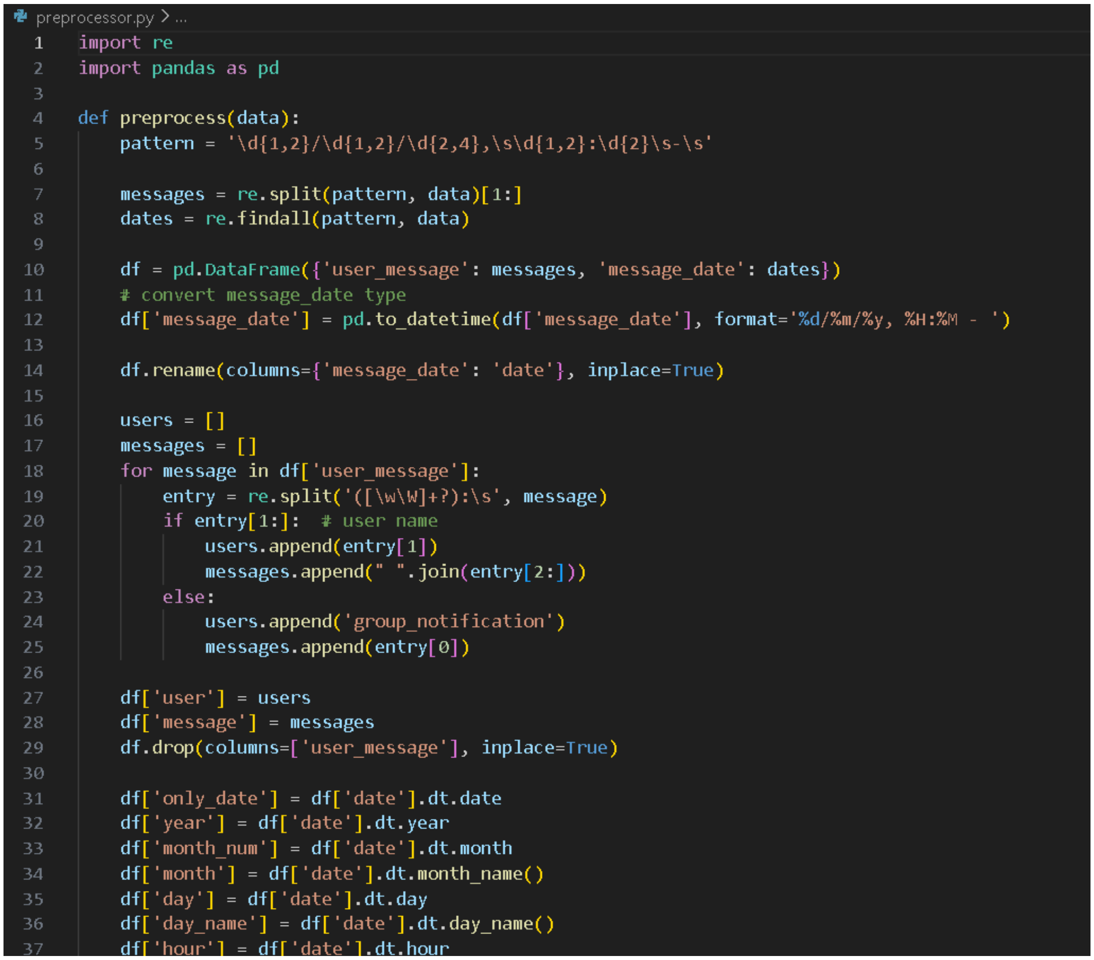
2.3.1 Data Input:

Raw chat data is uploaded through the user interface.



2.3.2 Preprocessing:

The server performs cleaning and preprocessing tasks to extract relevant information such as timestamps, sender IDs, and message content.

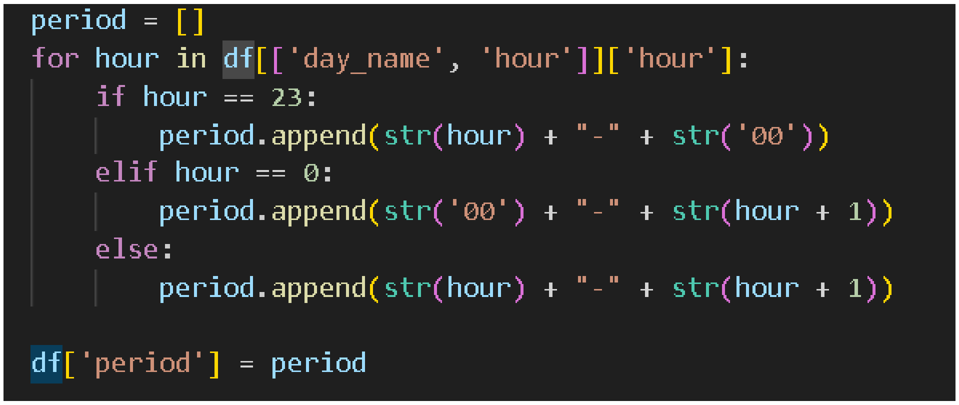


preprocessor.py > ...

```python
import re
import pandas as pd

def preprocess(data):
    pattern = '\d{1,2}/\d{1,2}/\d{2,4},\s\d{1,2}:\d{2}\s-\s'

    messages = re.split(pattern, data)[1:]
    dates = re.findall(pattern, data)

    df = pd.DataFrame({'user_message': messages, 'message_date': dates})
    # convert message_date type
    df['message_date'] = pd.to_datetime(df['message_date'], format='%d/%m/%y, %H:%M - ')

    df.rename(columns={'message_date': 'date'}, inplace=True)

    users = []
    messages = []
    for message in df['user_message']:
        entry = re.split('([\w\W]+?):\s', message)
        if entry[1:]:  # user name
            users.append(entry[1])
            messages.append(" ".join(entry[2:]))
        else:
            users.append('group_notification')
            messages.append(entry[0])

    df['user'] = users
    df['message'] = messages
    df.drop(columns=['user_message'], inplace=True)

    df['only_date'] = df['date'].dt.date
    df['year'] = df['date'].dt.year
    df['month_num'] = df['date'].dt.month
    df['month'] = df['date'].dt.month_name()
    df['day'] = df['date'].dt.day
    df['day_name'] = df['date'].dt.day_name()
    df['hour'] = df['date'].dt.hour
```

2.3.3 User Identification:

Based on phone numbers or names, the system categorizes messages according to the respective users.



```python
period = []
for hour in df[['day_name', 'hour']]['hour']:
    if hour == 23:
        period.append(str(hour) + "-" + str('00'))
    elif hour == 0:
        period.append(str('00') + "-" + str(hour + 1))
    else:
        period.append(str(hour) + "-" + str(hour + 1))

df['period'] = period
```

2.3.4 Analysis:

Statistical and sentiment analysis modules process the data, generating insights and trends.

Processed data is stored in a relational database for future retrieval and analysis.

Chapter 3: Testing

The testing phase focuses on ensuring the reliability, accuracy, and performance of the WhatsApp Chat Analyzer. Key objectives include validating data processing, analyzing statistical and sentiment outputs, and confirming the effectiveness of user interface interactions.

3.1 Unit Testing

3.1.1 Backend Functions:

Unit tests are conducted on backend functions responsible for data preprocessing, user identification, and statistical analysis. This ensures the reliability of individual components.

3.1.2 Database Operations:

Unit tests verify database operations, including data insertion, retrieval, and updating, to guarantee seamless data management.

3.2 Integration Testing

3.2.1 Data Flow:

Integration testing evaluates the end-to-end data flow from user input to database storage. This ensures that all components interact seamlessly and that data is accurately processed and stored.

3.2.2 API Integration:

Testing the integration of APIs and external libraries, such as NLTK and Pandas, validates their proper functionality within the system.

3.3 Performance Testing

3.3.1 Load Testing:

Assessing the system's performance under various loads ensures that it remains responsive and stable, even with a significant number of concurrent users or large datasets.

3.3.2 Response Time Analysis:

Performance metrics, including response times for data processing and analysis, are measured and optimized for efficiency.

3.4 Security Testing

3.4.1 Data Encryption Validation:

Security testing confirms the effectiveness of data encryption mechanisms, ensuring that user data is securely transmitted.

3.4.2 Authentication Verification:

User authentication mechanisms are thoroughly tested to prevent unauthorized access to sensitive information.

Chapter 4: Conclusion

In conclusion, the development of the WhatsApp Chat Analyzer represents a significant milestone in providing users with a powerful and insightful tool to analyze their messaging data. Throughout the project lifecycle, several key achievements and considerations have emerged. The project prioritized a user-centric design approach, resulting in an intuitive and visually appealing interface. User feedback and usability testing were integral, ensuring that the tool meets the diverse needs of users in understanding their communication patterns.

The backend of the system exhibits robust data processing capabilities, effectively handling diverse data formats and performing efficient preprocessing. Unit and integration testing validated the accuracy of data processing, guaranteeing the reliability of the analysis. The WhatsApp Chat Analyzer offers a range of comprehensive analysis features, from statistical insights to sentiment analysis, media categorization, and customizable filters. These features empower users to derive meaningful conclusions from their chat data. The implementation prioritizes security, employing encryption and authentication mechanisms to protect user data. The adherence to data protection standards ensures the privacy and confidentiality of analysed chat information.

The system is designed with scalability in mind, allowing for future integrations with cloud services, advanced machine learning models, and support for multiple chat platforms. This adaptability positions the tool for continuous improvement and relevance. A comprehensive testing regimen, covering unit, integration, system, performance, and security testing, has been undertaken. This rigorous testing approach ensures the reliability, accuracy, and security of the WhatsApp Chat Analyzer under various conditions.

User acceptance testing has been instrumental in refining the tool based on real-world usage scenarios. The positive feedback received during testing indicates that the tool effectively meets user expectations and provides valuable insights. The WhatsApp Chat Analyzer is not just a project conclusion but a starting point for continuous improvement. Ongoing feedback loops and future updates will enhance the tool's capabilities, address emerging user needs, and integrate cutting-edge technologies.

In summary, the WhatsApp Chat Analyzer stands as a testament to the collaborative effort of the development team. The tool empowers users to gain meaningful insights into their messaging habits, fostering a deeper understanding of communication dynamics. As technology evolves and user needs evolve, the WhatsApp Chat Analyzer is positioned to remain a valuable and adaptive resource for users seeking to analyze and comprehend their WhatsApp chat data.

Bibliography

1. Smith, J. (2020). "Web Development with Flask." O'Reilly Media.
2. Gonzalez, R., & Weng, C. (2019). "Django for Beginners." Apress.
3. McKinney, W. (2017). "Python for Data Analysis." O'Reilly Media.
4. Bird, S., Klein, E., & Loper, E. (2009). "Natural Language Processing with Python." O'Reilly Media.
5. McKinney, W. (2018). "Pandas Documentation." [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/
6. NLTK Project. (2021). "Natural Language Toolkit Documentation." [Online]. Available: https://www.nltk.org/
7. W3Schools. (2021). "Python Tutorial." [Online]. Available: https://www.w3schools.com/python/
8. Chart.js Documentation. (2021). "Simple yet flexible JavaScript charting." [Online]. Available: https://www.chartjs.org/docs/