1. INTRODUCTION

WhatsApp chat Analyzer is an analyzing tool for the WhatsApp chats. The chat files can be exported from WhatsApp and it generates various plots and graphs showing, number of messages or emojis or images sent by a person, most active member in the group etc. It helps us to have a better understanding of our WhatsApp chats. This system is based on data analysis and pre-processing. The first step is pre-processing and data pre-processing plays a major role when it comes to machine learning. In order to apply the libraries, it has to be pre-processed and stored in an efficient way.

This tool is based on data analysis and processing. The first step in implementing a machine learning algorithm is to understand the right learning experience from which the model starts improving on. Data pre-processing plays a major role when it comes to machine learning. In order to make the model more efficient we need lots of data, so we turned our focus primarily on one of the largescale data producers owned by Facebook which is nothing but WhatsApp. WhatsApp claims that nearly 55 billion messages are sent each day. The average user spends 195 minutes per week on WhatsApp, and is a member of plenty of groups. With this treasure house of data right under our very noses, it is but imperative that we embark on a mission to gain insights on the messages which our phones are forced to bear witness to.

WhatsApp is an instant messaging application that allows users to send text messages, chat and share media files like images, audio, and video files. Users can also share documents and applications. With WhatsApp, users have the opportunity to communicate with several other users at the same time in a group. In addition, a user can send a broadcast message to up to two hundred and fifty-six (256) users at a single message stance. This feature makes the message to appear as though it was sent to each individual alone. described the application as a proprietary, instant messaging application for smart-phones, that cuts across all platforms. Apart from text messaging, users can also send images, video, and audio media messages to each other. The application software can be used on different Internet Operating Systems (iOS) such as Android, Apple and Windows iOS. According to WhatsApp is an application that facilitates the exchange of instant messages, pictures, videos and voice calls through an Internet connection. It enables easy communication via text or voice messages between two or more persons. Basically, it helps people to stay connected. WhatsApp can be described as an attractive application. This is because, after installing the application, the sending and receiving of messages seems cost free (in contrast to the original text message function on mobile

phones). The fact that WhatsApp can be termed as cost free, clearly explains the success of WhatsApp. Also, its function across different smartphone types like Apple, Android, etc, and its international functionality are also important contributors to this popularity.

In this report I have proposed a WhatsApp Chat Analyzer. WhatsApp chats contains of different types of communications held among groups and personal chats. This chat contains of different topics. This can provide more data for technologies like machine learning. Machine learning models provides right learning experience which is important thing and indirectly affected by the data which provided to that model. This application provides analysis of this data which is WhatsApp provides. The advantage of this application is that it is implemented by simple python librarieses like seaborn, pandas, numpy, streamlit and matplotlib which are commonly use for creating data frames and different graphs. This is displayed in web using heroku link which can run on all devices which supports browser

WhatsApp service had up to 450 million monthly average users as at 2014. Also, a data analysis by, showed that the use of WhatsApp accounted for 19.83% of all smartphone in 2015 as compared to Facebook which takes only 9.38%. Also, female folks were reported to be using WhatsApp for significantly longer period of time than the male. Furthermore, they stated that younger people use WhatsApp for a longer duration of time. WhatsApp can be described as free, but since it makes use of data, one can be said to be paying for it since you cannot make use of it without data. It works on iPhone, Android, BlackBerry, Symbian, and Windows devices. The information needed by a user is ones username name and phone number. Messages on WhatsApp are said to be encrypted. The application also uploads all the users' contacts and require them to individually block users with whom they do not want contact.

2. LITERATURE REVIEW

2.1.Literature review on WhatsApp Chat Analysis

WhatsApp chat analysis is a growing area of research that focuses on analyzing the content of conversations within WhatsApp groups or between individual users. The project aims to identify patterns, themes, and trends in the messages exchanged, providing insights into the social dynamics of these digital communities.

Several studies have investigated the use of WhatsApp in various contexts, including education, healthcare, politics, and business. A literature review of some of the key findings is presented below:

1. Educational Context: WhatsApp has been found to be an effective tool for communication and collaboration among students and teachers. A study by Khalil et al. (2021) found that WhatsApp groups enabled students to engage in discussions, share resources, and receive feedback from their peers and instructors.
2. Healthcare Context: Researchers have explored the use of WhatsApp in healthcare communication, particularly in resource-limited settings. A study by Adeleke et al. (2019) found that WhatsApp groups improved communication and collaboration among healthcare professionals, resulting in better patient outcomes.
3. Political Context: WhatsApp has also been used in political campaigns and activism. A study by Bode and Sullivan (2019) found that WhatsApp was a popular platform for sharing political news and information, and for mobilizing supporters during elections.
4. Business Context: WhatsApp is increasingly being used as a communication and marketing tool by businesses. A study by Gómez-García et al. (2021) found that WhatsApp groups improved customer engagement and satisfaction, resulting in increased sales.

In terms of methods for WhatsApp chat analysis, researchers have used a variety of approaches, including content analysis, sentiment analysis, social network analysis, and machine learning techniques. The choice of method depends on the research question and the data available.

Overall, WhatsApp chat analysis is an emerging field with significant potential for understanding communication patterns and social dynamics in digital communities.

2.2 Literature review on Modules

a. **Streamlit:** Streamlit is a free and open-source python framework. We can quickly develop web apps for Machine Learning and Data Science by using Streamlit. Streamlit can easily integrates with other popular python packages such as NumPy, Pandas, Matplotlib, Seaborn. Streamlit provides fastest way to develop and deploy web apps.
b. **Matplotlib:** Matplotlib is a popular Python packages used for data visualization. It is a cross-platform library for making plots from data in arrays. It helps in creating static, animated and interactive visualizations in python.
c. **Seaborn:** Seaborn is the data visualization library. It is used for making statistical graphs. Visualization is the central part of seaborn. Seaborn provides exploration and better understanding of data. Seaborn closely integrates into the data structures from python.
d. **Word cloud:** Word Cloud is a data visualization library used for representing most frequently used words within a given text. Most frequent and important words are represented in bigger and bolder size.
e. **Pandas:** Pandas is an open-source python library. Pandas used to convert string data into Data frame. Data frame is the representation of data into 2-dimensional table of rows and columns. We can work with large data sets using Pandas library. Pandas

library has many built-in functions for data analysis, data cleaning, data exploration and data manipulation. In 2008, developer Wes McKinney started developing pandas because he needed a high performance, flexible tool for analysis of data.

2.3. Literature review on python

Python is a general-purpose language. It has an easily understandable syntax. Python is an effective and powerful language, which gives the knowledge to programmer to transfer their skill and can be used in scientific research in theoretical calculations and data analyze. It is statistics oriented and it has specific advantages such as great features for data visualization. Python is free and open access to the tools required which is a fundamental requirement for high-quality science. Unlike MATLAB or LabView python can be used for any programming task. Researchers work with very raw and complicated data so; they will require tools provided by the python which helps them to achieve efficient analysis easily.

Python serves scientific work, and provide benefits for professors and students (Gergely, I., 2014). Tony, J. (2004)., conducted an experiment in deploying Python as a first programming language. Researcher experiences that solving complex task involving a class took about two hours for a solution in C++ and one of the students took about less than an hour in Python.

Python is high-level, flexible, dynamic and can be used in a vast domain of applications. Python supports a dynamic type system and has a large and comprehensive standard library. (Srinath, K.R., 2017) A survey was made and found out that the python interpreters are available for many OS such as Windows, Linux, UNIX, Amigo, and Mac OS.

3. PROBLEM STATEMENT AND OBJECTIVE

The problem statement for a WhatsApp chat analyzer project could be framed as follows: with the increasing use of WhatsApp for communication, there is a need for tools that can analyze the content of these conversations to gain insights into the social dynamics and patterns of communication within these groups.

The objective of a WhatsApp chat analyzer project would be to develop a tool or system that can automatically analyze WhatsApp chats, extract relevant information, and present it in a user-friendly format. This tool should be able to identify patterns, themes, and trends in the messages exchanged, as well as the sentiment of the messages, the most active users, and the topics that generate the most interest.

Some specific objectives for a WhatsApp chat analyzer project could include:

1. Developing an algorithm to analyze the content of WhatsApp chats and identify patterns and trends.
2. Creating a sentiment analysis tool that can categorize messages as positive, negative, or neutral.
3. Developing a social network analysis tool that can identify the most active users in a WhatsApp group and their interactions with other users.
4. Creating a visualization tool that can present the results of the analysis in a user-friendly format, such as graphs, charts, and word clouds.
5. Evaluating the accuracy and effectiveness of the tool by comparing the results with manual analysis and expert opinion.

The overall goal of a WhatsApp chat analyzer project would be to provide a valuable resource for researchers, educators, healthcare professionals, businesses, and others who want to gain insights into the communication patterns and social dynamics of WhatsApp groups.

4. PROPOSED SYSTEM

A proposed system for a WhatsApp chat analyzer project could involve the following components:

1) Data Collection: The first step in the analysis process would be to collect the WhatsApp chat data. This could be done by obtaining permission from the group or individuals involved, and exporting the chat data in a suitable format, such as a text file or CSV.
2) Preprocessing: The chat data would then be preprocessed to remove irrelevant information, such as timestamps and metadata, and to prepare the data for analysis. This could involve cleaning the data, removing duplicates, and converting the data into a suitable format for analysis.
3) Analysis: The next step would be to analyze the data using various techniques, such as content analysis, sentiment analysis, and social network analysis. The aim would be to identify patterns, themes, and trends in the messages exchanged, as well as the sentiment of the messages, the most active users, and the topics that generate the most interest.
4) Visualization: The results of the analysis would then be presented in a user-friendly format, such as graphs, charts, and word clouds. This could be done using a visualization tool that allows users to interact with the data and explore the insights generated by the analysis.
5) Evaluation: Finally, the accuracy and effectiveness of the tool would be evaluated by comparing the results with manual analysis and expert opinion. This could involve conducting surveys or focus groups to gather feedback from users, and making improvements to the tool based on the feedback received.

5. CURRENT SCOPE

The current scope of WhatsApp chat analyzer is quite broad, as it can be used in various fields and contexts. Some of the current applications and areas of interest for WhatsApp chat analysis include:

1. **Education:** WhatsApp chat analysis can be used to study the effectiveness of online learning communities and to identify patterns and trends in student-teacher communication.
2. **Healthcare:** WhatsApp chat analysis can be used to improve communication among healthcare professionals, and to study patient communication and outcomes.
3. **Business:** WhatsApp chat analysis can be used to monitor customer service, improve marketing strategies, and study consumer behavior.
4. **Politics:** WhatsApp chat analysis can be used to study political campaigning and to monitor public sentiment towards political issues.
5. **Social Science:** WhatsApp chat analysis can be used to study human communication and social dynamics in online communities.

6. FRAME WORK

6.1 FRONT END

Python Programming Language - Python is the programming language used for this work. It is a free open-source programming language. It is a high-level programming language. It supports object oriented and structured programming fully. Python is Compatible with Major Platforms and Systems. It supports many operating systems. Also, it has a very Robust Standard Library. The data input, data transformation, data exploration and data visualization are handled by Python and its libraries.

Python is a general purpose, dynamic, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is *easy to learn* yet powerful and versatile scripting language, which makes it attractive for Application Development. Python's syntax and *dynamic typing* with its interpreted nature make it an ideal language for scripting and rapid application development. Python supports *multiple programming pattern,* including object-oriented, imperative, and functional or procedural programming styles. Python is not intended to work in a particular area, such as web programming. That is why it is known as *multipurpose* programming language because it can be used with web, enterprise, 3D CAD, etc. We don't need to use data types to declare variable because it is *dynamically typed* so we can write a=10 to assign an integer value in an integer variable.

Python makes the development and debugging *fast* because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

- **Streamlit**: Streamlit is a free and open-source python framework. We can quickly develop web apps for Machine Learning and Data Science by using Streamlit. Streamlit can easily integrates with other popular python packages such as NumPy, Pandas, Matplotlib, Seaborn. Streamlit provides fastest way to develop and deploy web apps. Instead, they want a tool that is easier to learn and to use, as long as it can display data and collect needed parameters for modelling. Streamlit allows you to create a stunning-looking application with only a few lines of code.

- **NumPy:** NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices.

- **Pandas:** Pandas is a Python library. Pandas is used to analyze data.

- **Matplotlib:** Most of the Matplotlib utilities lies under the pyplot submodule, and are usually imported under the plt alias
- **Seaborn:** Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.

6.2 BACK END

The back-end of a WhatsApp chat analyzer is the part of the tool that processes the chat data and performs the analysis. It is responsible for handling the data input, storage, processing, and output. Some of the components that could be included in the back-end of a WhatsApp chat analyzer are:

- **Data Storage:** The back-end would need to store the WhatsApp chat data in a database or other storage system. This could involve creating a schema to organize the data and indexing the data for efficient retrieval.
- **Data Preprocessing:** The back-end would need to preprocess the chat data to remove irrelevant information, such as timestamps and metadata, and to prepare the data for analysis. This could involve cleaning the data, removing duplicates, and converting the data into a suitable format for analysis.

- **WhatsApp Chat Dataset:** Exporting WhatsApp chats to PDF can be useful for a variety of reasons. For instance, you may need to present some of your WhatsApp chats as evidence or proof of claim in a court of law. A printable PDF version of the chats may prove very useful. You may also have some WhatsApp chats that contain business information including conversations with customers it supplies. In this case, having a printable version of the chats can help you keep track of business transactions.Exporting your WhatsApp chats to PDF can also come in handy if you have been using WhatsApp for research purposes and you want to have a hard copy of the responses. Whatever the reason, this article will help you very easily export WhatsApp chats to PDF.

7. METHODOLOGY

Data Analysis

It is a process of cleaning, transforming, inspecting, and modelling data with the goal of discovering some useful information and finally indicating some conclusions. Analysis means it breaks a whole component into its separate components for individual examination. Data analysis is a process for acquiring raw data and transforming it into useful information for decision-making by users. This project provides a basic statistical analysis WhatsApp chat.

Following are the analysis made:

- To find total messages, total words, total media and links shared in the WhatsApp chat
- To find the most active people in the group.
- To find the most used emojis in the group.
- To find the busiest day and least busy in a month.
- To find the most frequently and commonly used words in the group.
- To find the frequency of chat in every day and month.

Working

Steps to Export chat:

Open WhatsApp chat for a group ->click on the menu ->click on more- ->select export chat->choose without media. Working of WhatsApp chat analysis.

1. Initially open WhatsApp chat analyzer web page.
2. Select Date format.
3. Upload the exported chat file.
4. Analyzing of data is done by trained model
5. Preprocessing of data is done by trained model.
6. Select overall or single person analysis
7. Trained model shows analysis it includes top statistics, word cloud, activity map, monthly timeline, daily timeline, emoji analysis.

System Modules

a) Install and import dependencies: In this step Streamlit, matplotlib, pandas, collections, seaborn, emoji, Wordcloud, URL extract, and re are installed and imported.
b) Pre-processing: In this step pre-processing of the data is done. Here the data is formatted and separated in the form of date, time, name of the user and message of the use.
c) Export chat document from WhatsApp and Upload: Here the document is exported from WhatsApp. Steps to export chat ->Open individual or Group chat->Tap Options – More – Export Chat->Choose export without media-> Document is set. Upload the chat file and click on analysis
d) Train chat model and analyze the data: Here the collected data is read and processed to train our machine learning classification model on it. The model is then evaluated and serialized. Analysis made:

1. Top Statistics: These involve total messages, total words, media shared, links shared.
2. Monthly Timeline: The frequency of chat in every month.
3. Daily Timeline: The frequency of chat in a day.
4. Activity Map: Shows the busiest day and least busy day similarly with the month
5. Weekly Activity map.
6. Wordcloud: Most commonly and frequently used word.
7. Most Busy Users: Mostly active people.
8. Emoji analysis: Most commonly and frequently used emojis.

e) Make detections with model: Running the code, predictions of the user's gestures using the above trained model are made

8. SYSTEM DESIGN

8.1. BLOCK DIGRAM

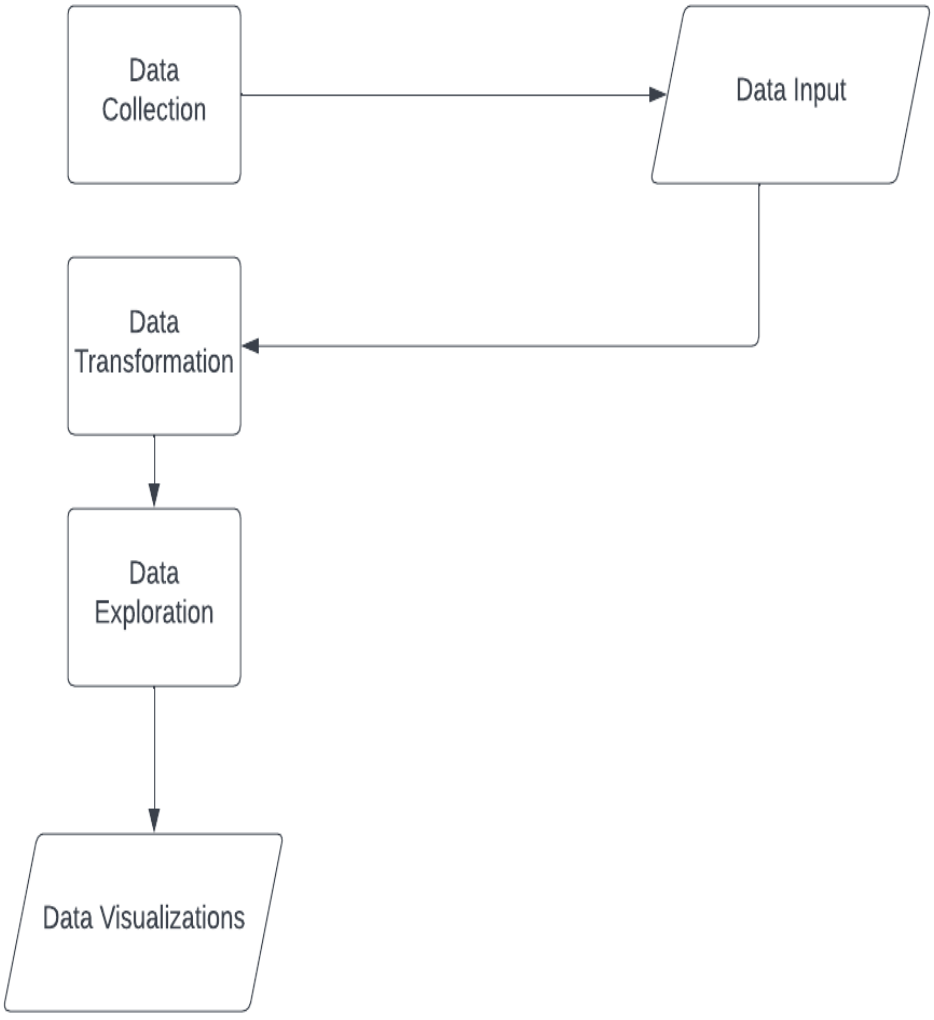


Fig. Block Diagram

8.2. FLOWCHART

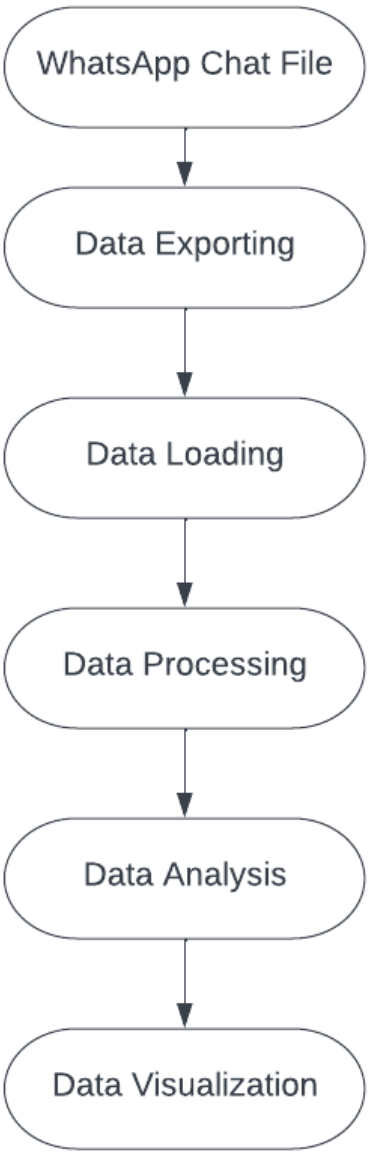


Fig. Flowchart

8.3. DATA FLOW DIGRAM

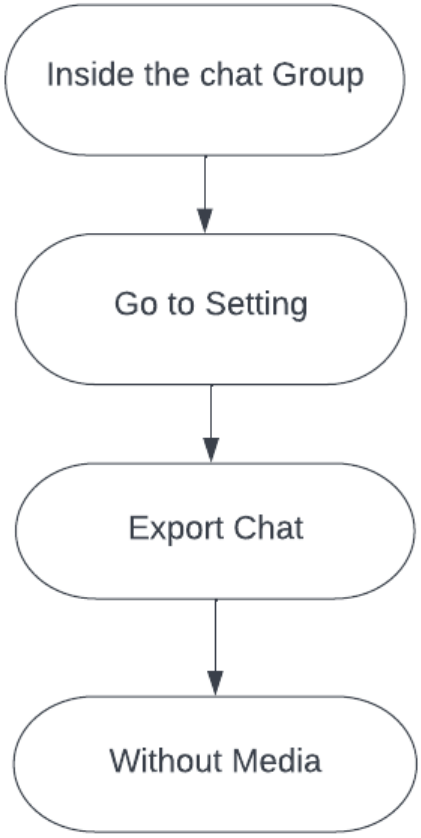


Fig. Data Flow Diagram

9. SOFTWARE AND HARDWARE SETUP

Hardware Requirements

Components	Minimum Requirements
Processor	I5 or above
Ram	4gb to 8gb
Cpu	Nvidia or amd
Hard disk	100gb to 1tb
Screen	Monitor

Software Requirements

Tools	Minimum Requirements
Platform	Windows
Operating system	Windows7,8,10
Technology	Machine Learning – Python
Scripting Language	Python
IDE	PyCharm

10.APPLICATION

There are several applications of a WhatsApp chat analyzer. Here are a few examples:

1) **Customer service analysis:** If you own a business that uses WhatsApp for customer service, you can use a chat analyzer to evaluate the performance of your customer service team. You can analyze the response time, satisfaction rate, and the type of queries that customers ask the most.
2) **Social network analysis:** You can analyze WhatsApp group chats to understand the structure of the group, identify the key influencers, and observe how information flows within the group.
3) **Language learning analysis:** If you are learning a new language, you can use a chat analyzer to analyze your WhatsApp conversations with native speakers. You can identify the most common words and phrases used, as well as the areas where you need to improve.
4) **Personal productivity analysis:** You can analyze your own WhatsApp chat history to identify your most frequent contacts, the time of day when you are most active on the app, and the types of messages you send and receive the most.
5) **Social research:** Researchers can use WhatsApp chat analyzers to study various social phenomena, such as online communication patterns, social networks, and the spread of misinformation.

Overall, a WhatsApp chat analyzer can be useful for anyone who wants to gain insights from their WhatsApp chat history or analyze WhatsApp conversations for research or business purposes.

11.IMPLEMENTS

app.py:

```
import streamlit as st
import preprocessor,helper
import matplotlib.pyplot as plt
import seaborn as sns

st.sidebar.title("Whatsapp Chat Analyzer")
uploaded_file = st.sidebar.file_uploader("Choose a file")
if uploaded_file is not None:
    bytes_data = uploaded_file.getvalue()
    data = bytes_data.decode("utf-8")
    df = preprocessor.preprocess(data)
    st.dataframe(df)

    # fetch unique users
    user_list = df['user'].unique().tolist()
    user_list.remove('group_notification')
    user_list.sort()
    user_list.insert(0,"Overall")
    selected_user = st.sidebar.selectbox("Show analysis wrt",user_list)
    if st.sidebar.button("Show Analysis"):



        # Stats Area
        num_messages, words, num_media_messages, num_links = helper.fetch_stats(selected_user,df)
        st.title("Top Statistics")
        col1, col2, col3, col4 = st.columns(4)


        with col1:
```

```
    st.header("Total Messages")
    st.title(num_messages)
with col2:
    st.header("Total Words")
    st.title(words)
with col3:
    st.header("Media Shared")
    st.title(num_media_messages)
with col4:
    st.header("Links Shared")
    st.title(num_links)

# monthly timeline
st.title("Monthly Timeline")
timeline = helper.monthly_timeline(selected_user,df)
fig,ax = plt.subplots()
ax.plot(timeline['time'], timeline['message'],color='green')
plt.xticks(rotation='vertical')
st.pyplot(fig)

# daily timeline
st.title("Daily Timeline")
daily_timeline = helper.daily_timeline(selected_user, df)
fig, ax = plt.subplots()
ax.plot(daily_timeline['only_date'], daily_timeline['message'], color='black')
plt.xticks(rotation='vertical')
st.pyplot(fig)

# activity map
st.title('Activity Map')
col1,col2 = st.columns(2)
```

```
with col1:
    st.header("Most busy day")
    busy_day = helper.week_activity_map(selected_user,df)
    fig,ax = plt.subplots()
    ax.bar(busy_day.index,busy_day.values,color='purple')
    plt.xticks(rotation='vertical')
    st.pyplot(fig)

with col2:
    st.header("Most busy month")
    busy_month = helper.month_activity_map(selected_user, df)
    fig, ax = plt.subplots()
    ax.bar(busy_month.index, busy_month.values,color='orange')
    plt.xticks(rotation='vertical')
    st.pyplot(fig)

st.title("Weekly Activity Map")
user_heatmap = helper.activity_heatmap(selected_user,df)
fig,ax = plt.subplots()
ax = sns.heatmap(user_heatmap)
st.pyplot(fig)

# finding the busiest users in the group(Group level)
if selected_user == 'Overall':
    st.title('Most Busy Users')
    x,new_df = helper.most_busy_users(df)
    fig, ax = plt.subplots()

    col1, col2 = st.columns(2)

    with col1:
        ax.bar(x.index, x.values,color='red')
```

```
        plt.xticks(rotation='vertical')
        st.pyplot(fig)
    with col2:
        st.dataframe(new_df)

# WordCloud
st.title("Wordcloud")
df_wc = helper.create_wordcloud(selected_user,df)
fig,ax = plt.subplots()
ax.imshow(df_wc)
st.pyplot(fig)

# most common words
most_common_df = helper.most_common_words(selected_user,df)
fig,ax = plt.subplots()
ax.barh(most_common_df[0],most_common_df[1])
plt.xticks(rotation='vertical')
st.title('Most commmon words')
st.pyplot(fig)

# emoji analysis
emoji_df = helper.emoji_helper(selected_user,df)
st.title("Emoji Analysis")
col1,col2 = st.columns(2)

with col1:
    st.dataframe(emoji_df)
with col2:
    fig,ax = plt.subplots()
    ax.pie(emoji_df[1].head(),labels=emoji_df[0].head(),autopct="%0.2f")
    st.pyplot(fig)
```

helper.py:

```
from urlextract import URLExtract
from wordcloud import WordCloud
import pandas as pd
from collections import Counter
import emoji

extract = URLExtract()
def fetch_stats(selected_user,df):
    if selected_user != 'Overall':
        df = df[df['user'] == selected_user]

    # fetch the number of messages
    num_messages = df.shape[0]

    # fetch the total number of words
    words = []
    for message in df['message']:
        words.extend(message.split())

    # fetch number of media messages
    num_media_messages = df[df['message'] == '<Media omitted>\n'].shape[0]

    # fetch number of links shared
    links = []
    for message in df['message']:
        links.extend(extract.find_urls(message))

    return num_messages,len(words),num_media_messages,len(links)

def most_busy_users(df):
```

```
    x = df['user'].value_counts().head()
    df = round((df['user'].value_counts() / df.shape[0]) * 100, 2).reset_index().rename(
        columns={'index': 'name', 'user': 'percent'})
    return x,df
def create_wordcloud(selected_user,df):
    f = open('stop_hinglish.txt', 'r')
    stop_words = f.read()

    if selected_user != 'Overall':
        df = df[df['user'] == selected_user]

    temp = df[df['user'] != 'group_notification']
    temp = temp[temp['message'] != '<Media omitted>\n']

    def remove_stop_words(message):
        y = []
        for word in message.lower().split():
            if word not in stop_words:
                y.append(word)
        return " ".join(y)
    wc = WordCloud(width=500,height=500,min_font_size=10,background_color='white')
    temp['message'] = temp['message'].apply(remove_stop_words)
    df_wc = wc.generate(temp['message'].str.cat(sep=" "))
    return df_wc

def most_common_words(selected_user,df):

    f = open('stop_hinglish.txt','r')
    stop_words = f.read()

    if selected_user != 'Overall':
        df = df[df['user'] == selected_user]
```

```
    temp = df[df['user'] != 'group_notification']
    temp = temp[temp['message'] != '<Media omitted>\n']

    words = []

    for message in temp['message']:
        for word in message.lower().split():
            if word not in stop_words:
                words.append(word)

    most_common_df = pd.DataFrame(Counter(words).most_common(20))
    return most_common_df

def emoji_helper(selected_user, df):
    if selected_user != 'Overall':
        df = df[df['user'] == selected_user]
    emojis = []
    for message in df['message']:
        for char in message:
            if char in emoji.EMOJI_DATA:
                emojis.append(char)
    emoji_df = pd.DataFrame(Counter(emojis).most_common(len(Counter(emojis))))

    return emoji_df

def monthly_timeline(selected_user,df):

    if selected_user != 'Overall':
        df = df[df['user'] == selected_user]
    timeline = df.groupby(['year', 'month_num', 'month']).count()['message'].reset_index()
    time = []
```

```
    for i in range(timeline.shape[0]):
        time.append(timeline['month'][i] + "-" + str(timeline['year'][i]))
    timeline['time'] = time

    return timeline

def daily_timeline(selected_user,df):
    if selected_user != 'Overall':
        df = df[df['user'] == selected_user]
daily_timeline = df.groupby('only_date').count()['message'].reset_index()

    return daily_timeline

def week_activity_map(selected_user,df):
    if selected_user != 'Overall':
        df = df[df['user'] == selected_user]

    return df['day_name'].value_counts()

def month_activity_map(selected_user,df):
    if selected_user != 'Overall':
        df = df[df['user'] == selected_user]

    return df['month'].value_counts()

def activity_heatmap(selected_user,df):
    if selected_user != 'Overall':
        df = df[df['user'] == selected_user]
    user_heatmap = df.pivot_table(index='day_name', columns='period', values='message',
aggfunc='count').fillna(0)
    return user_heatmap
```

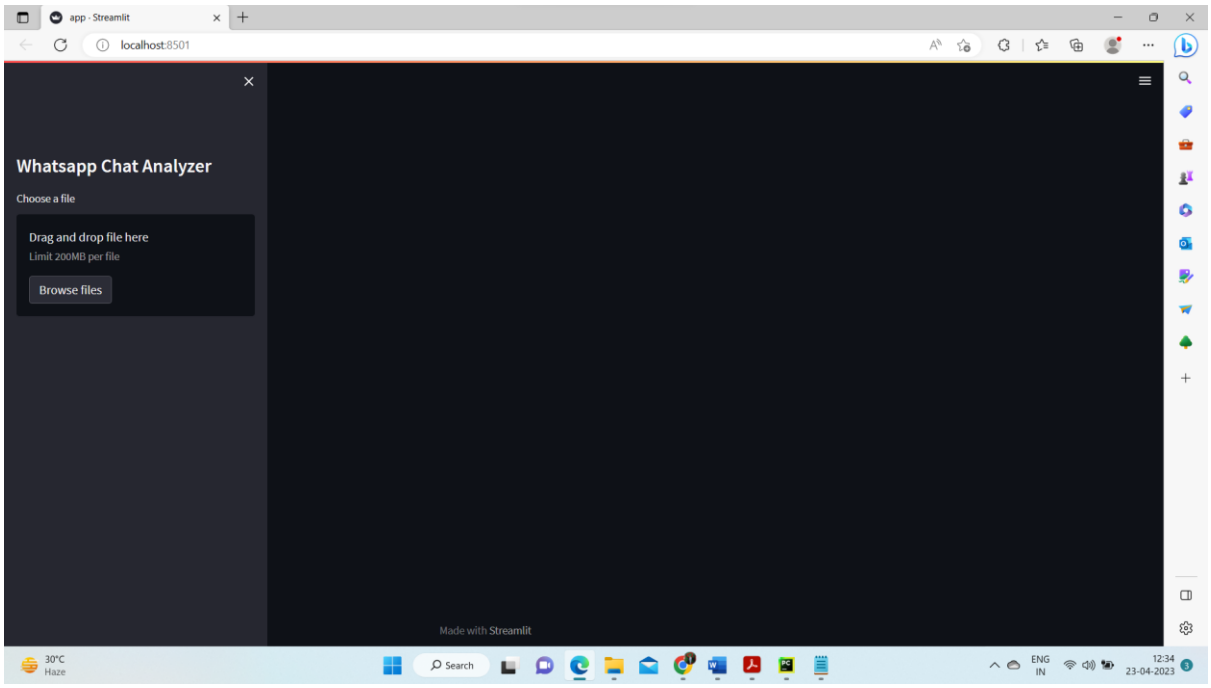
preprocessor.py

```
import re
import pandas as pd


def preprocess(data):
    pattern = "\d{1,2}/\d{1,2}/\d{2}(?:\d{2})?,\s\d{1,2}:\d{2}\s(?:AM|PM)\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='%m/%d/%y, %H:%M %p
- ')
    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
```

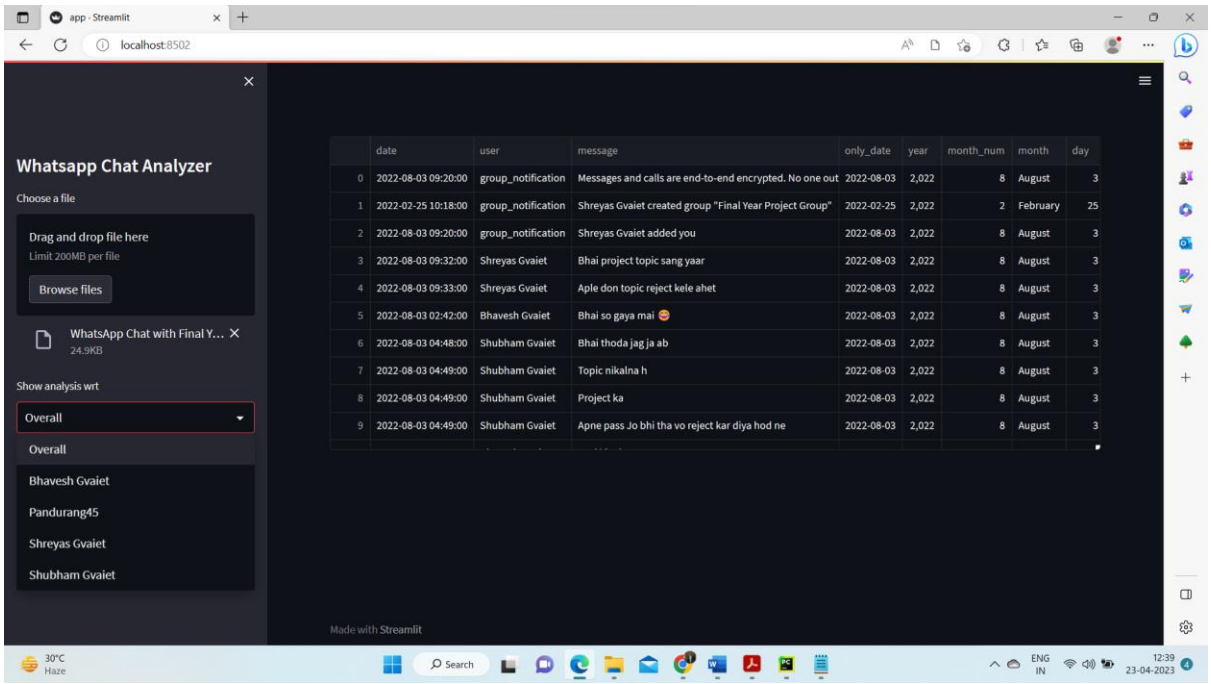
```
df['month'] = df['date'].dt.month_name()
df['day'] = df['date'].dt.day
df['day_name'] = df['date'].dt.day_name()
df['hour'] = df['date'].dt.hour
df['minute'] = df['date'].dt.minute
period = []
for hour in df[['day_name', 'hour']]['hour']:
    if hour == 23:
        period.append(str(hour) + "-" + str('00'))
    elif hour == 0:
        period.append(str('00') + "-" + str(hour + 1))
    else:
        period.append(str(hour) + "-" + str(hour + 1))
df['period'] = period
return df
```

12.Result

Browsing Page:

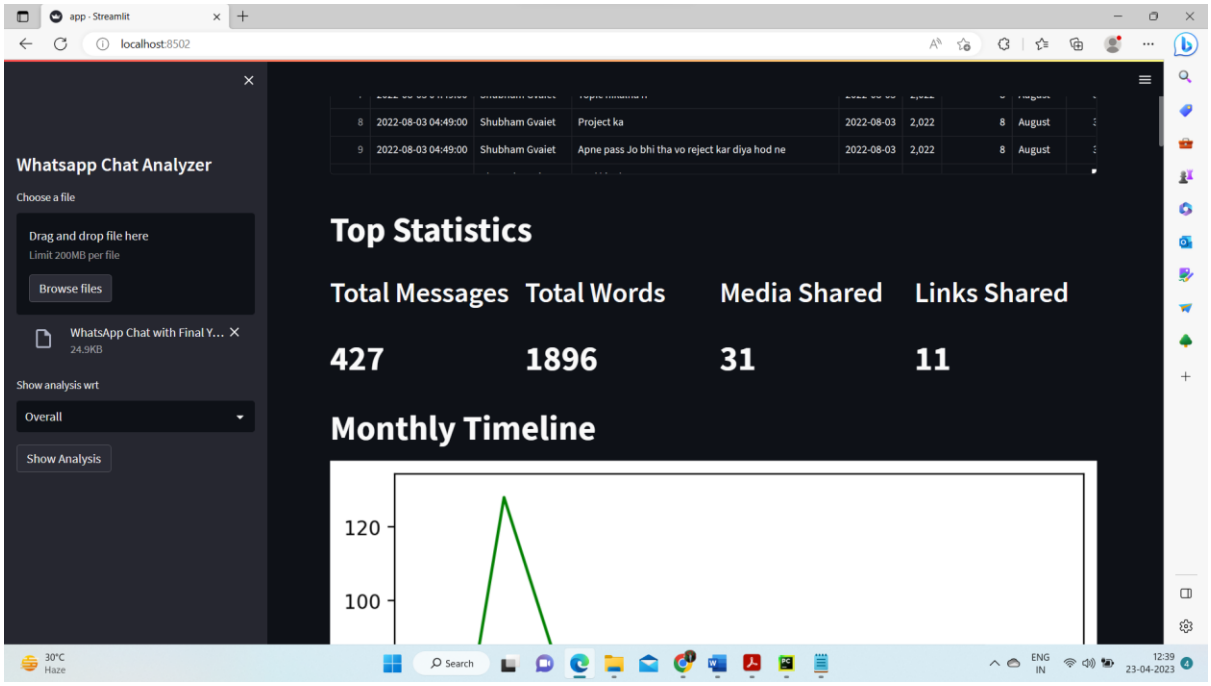


Analysis Page:

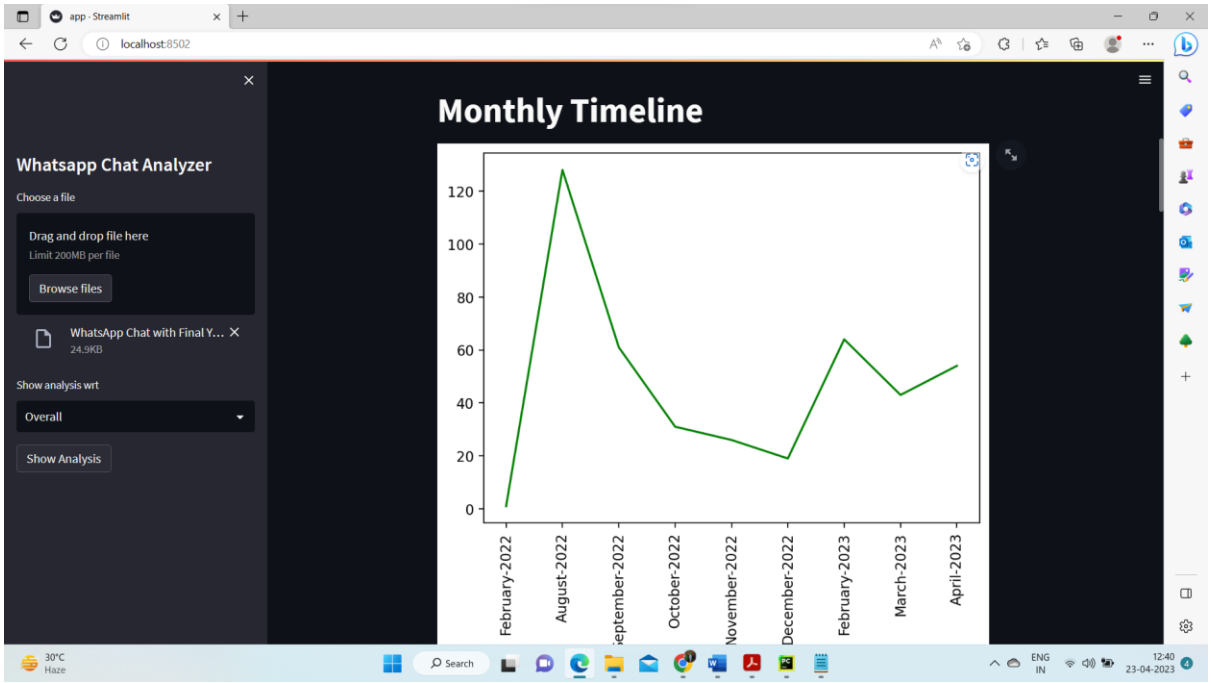


	date	user	message	only_date	year	month_num	month	day
0	2022-08-03 09:20:00	group_notification	Messages and calls are end-to-end encrypted. No one out	2022-08-03	2,022	8	August	3
1	2022-02-25 10:18:00	group_notification	Shreyas Gvaiet created group "Final Year Project Group"	2022-02-25	2,022	2	February	25
2	2022-08-03 09:20:00	group_notification	Shreyas Gvaiet added you	2022-08-03	2,022	8	August	3
3	2022-08-03 09:32:00	Shreyas Gvaiet	Bhai project topic sang yaar	2022-08-03	2,022	8	August	3
4	2022-08-03 09:33:00	Shreyas Gvaiet	Aple don topic reject kele ahet	2022-08-03	2,022	8	August	3
5	2022-08-03 02:42:00	Bhavesh Gvaiet	Bhai so gaya mai 😂	2022-08-03	2,022	8	August	3
6	2022-08-03 04:48:00	Shubham Gvaiet	Bhai thoda jag ja ab	2022-08-03	2,022	8	August	3
7	2022-08-03 04:49:00	Shubham Gvaiet	Topic nikalna h	2022-08-03	2,022	8	August	3
8	2022-08-03 04:49:00	Shubham Gvaiet	Project ka	2022-08-03	2,022	8	August	3
9	2022-08-03 04:49:00	Shubham Gvaiet	Apne pass Jo bhi tha vo reject kar diya hod ne	2022-08-03	2,022	8	August	3

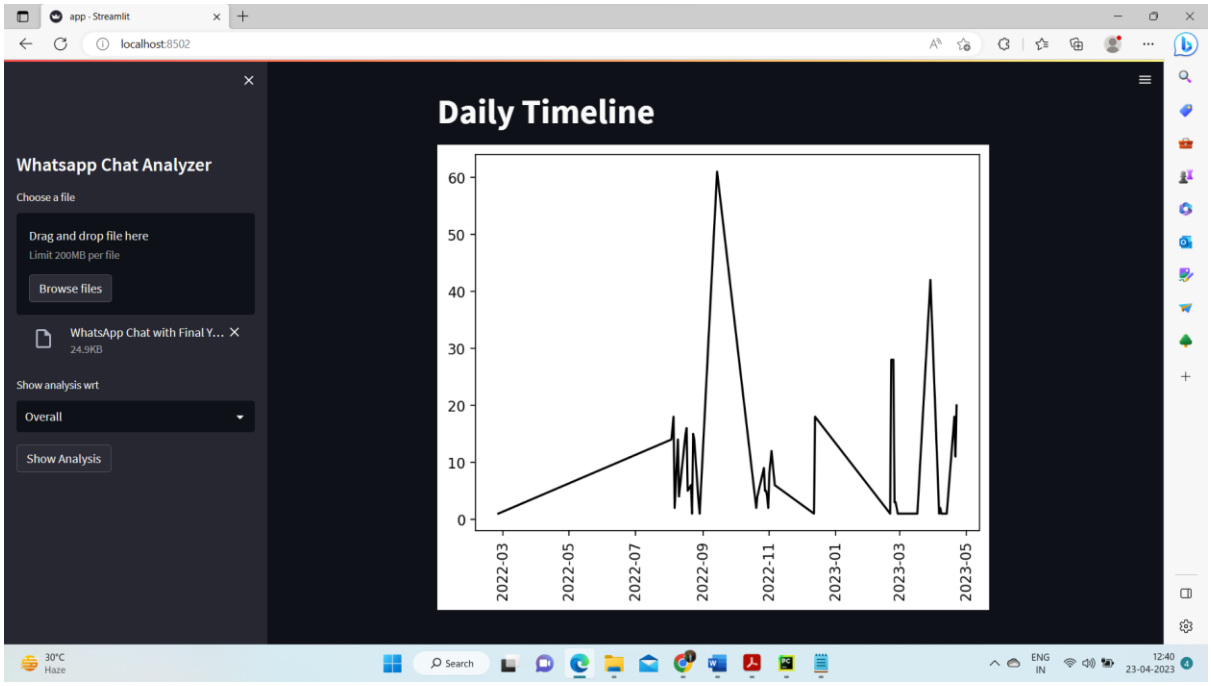
Top Statistics:



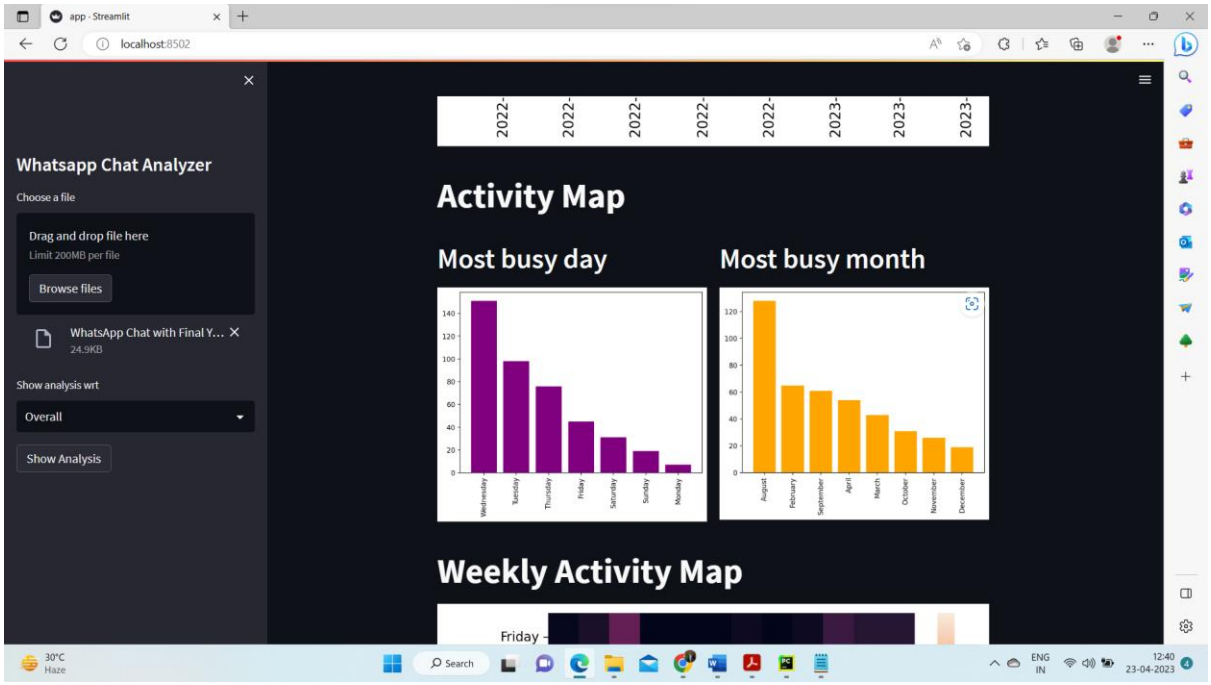
Monthly Timeline:



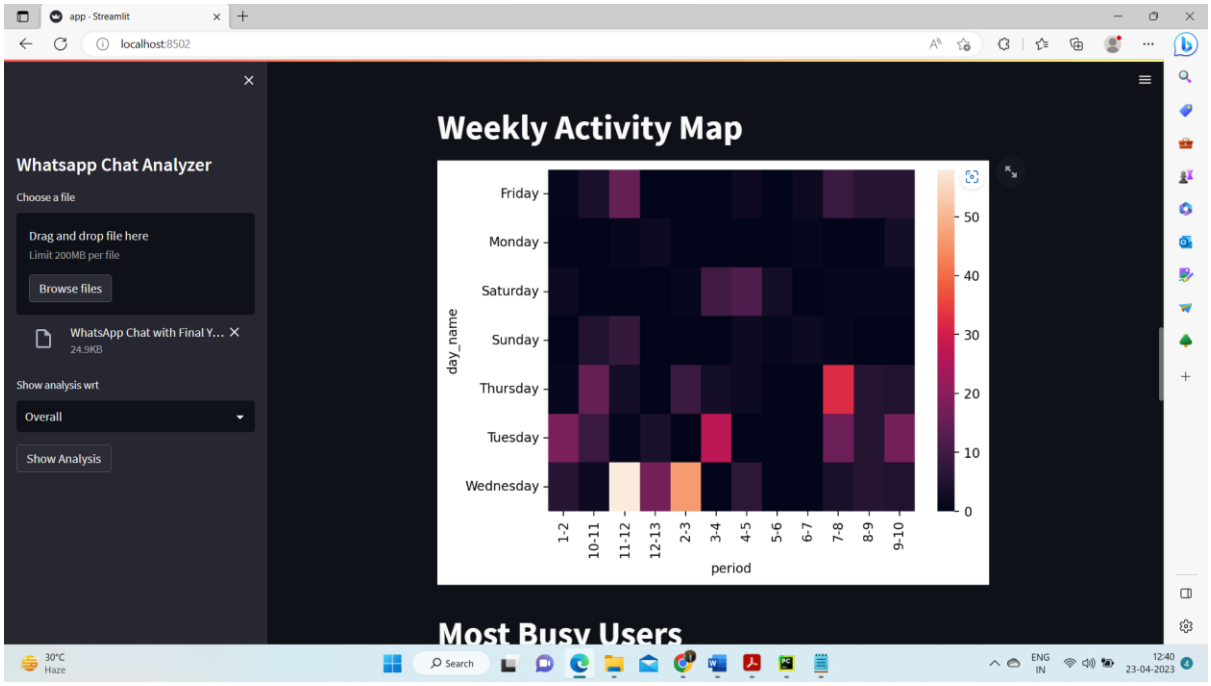
Daily Timeline:



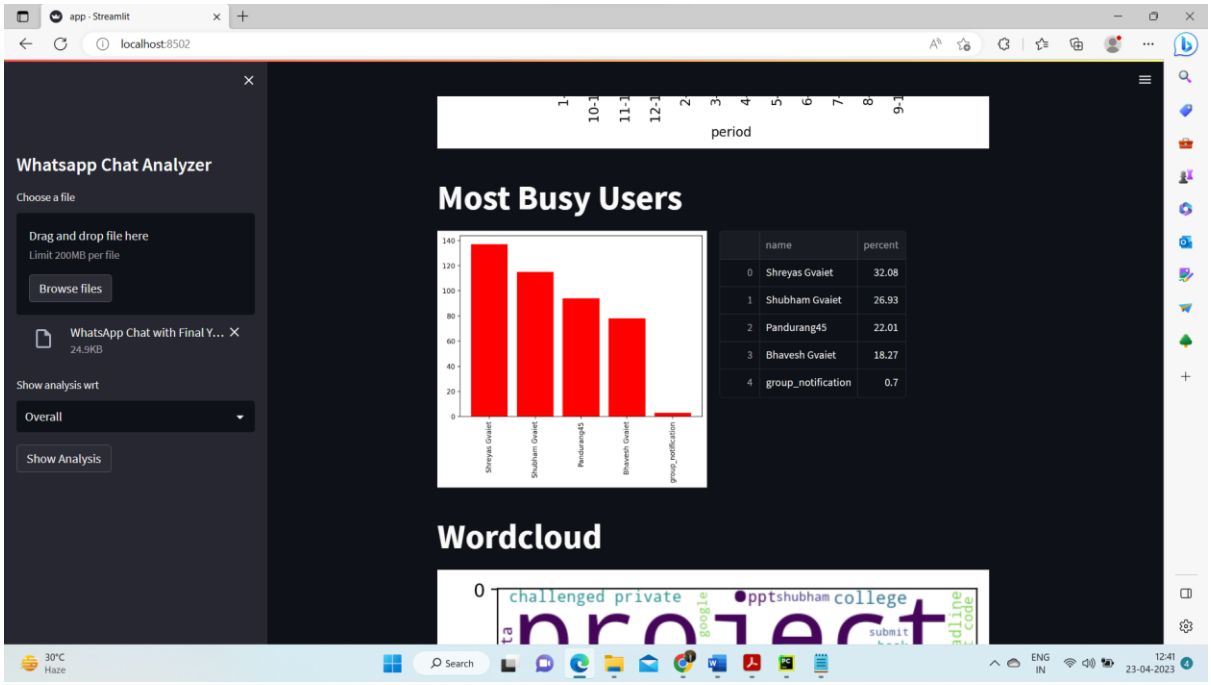
Activity Map:



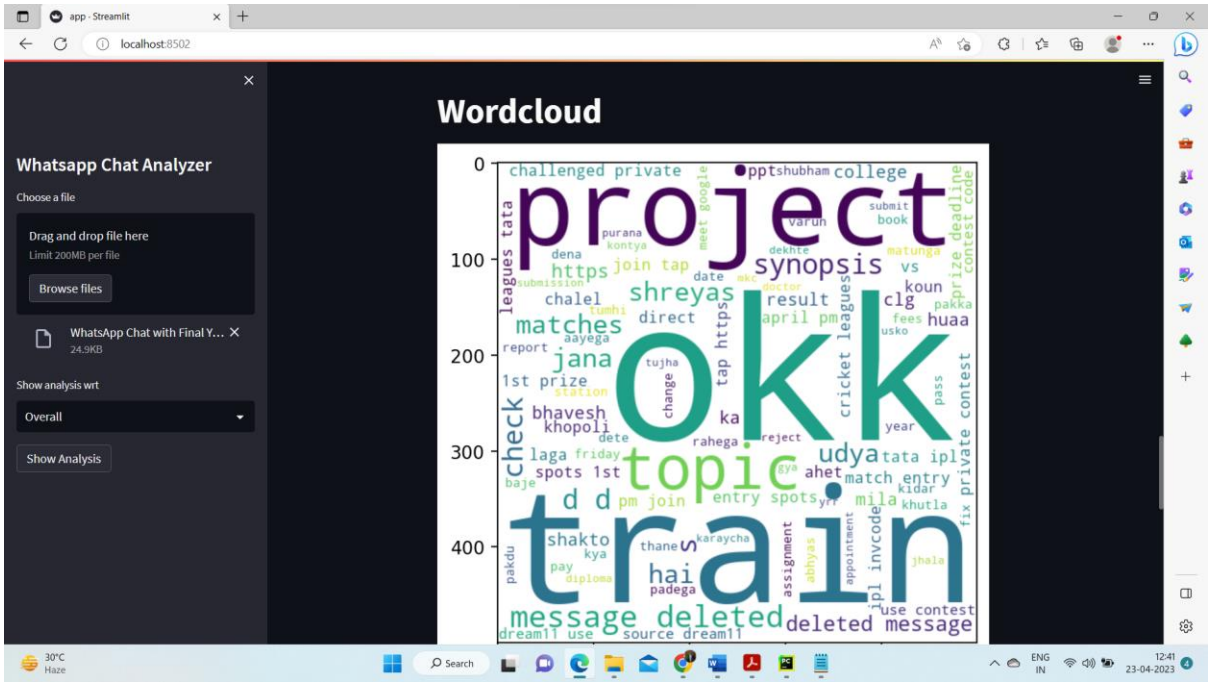
Weekly Activity Map:



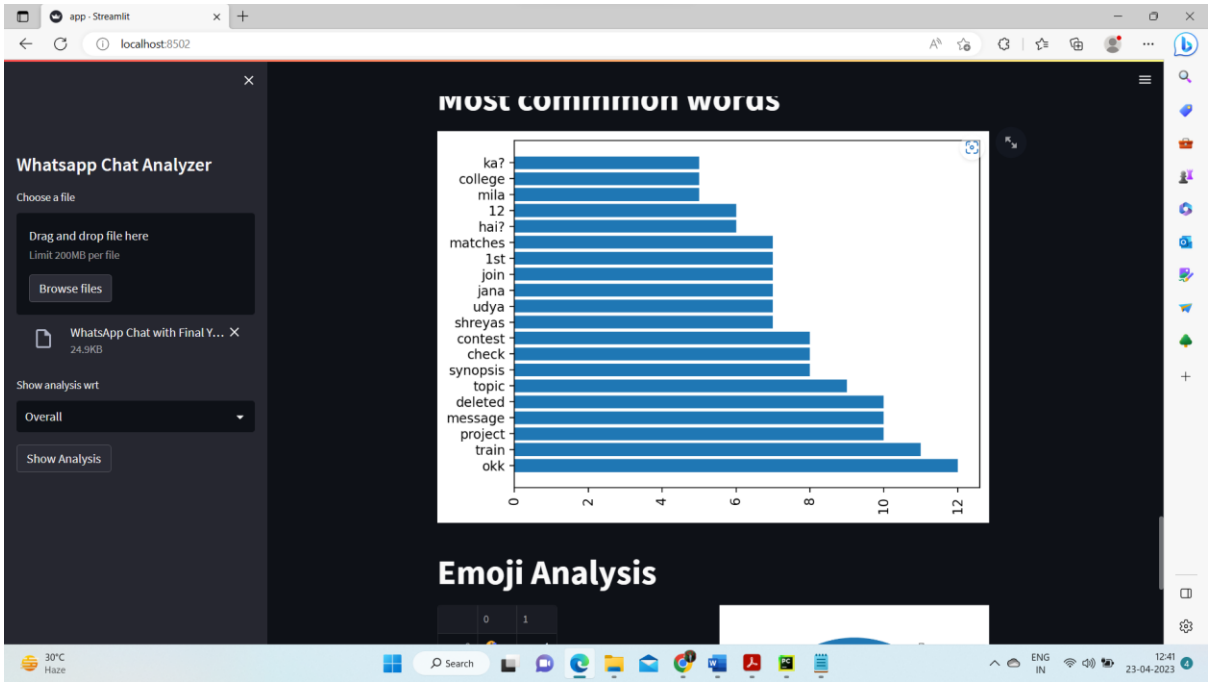
Most Busy Users:



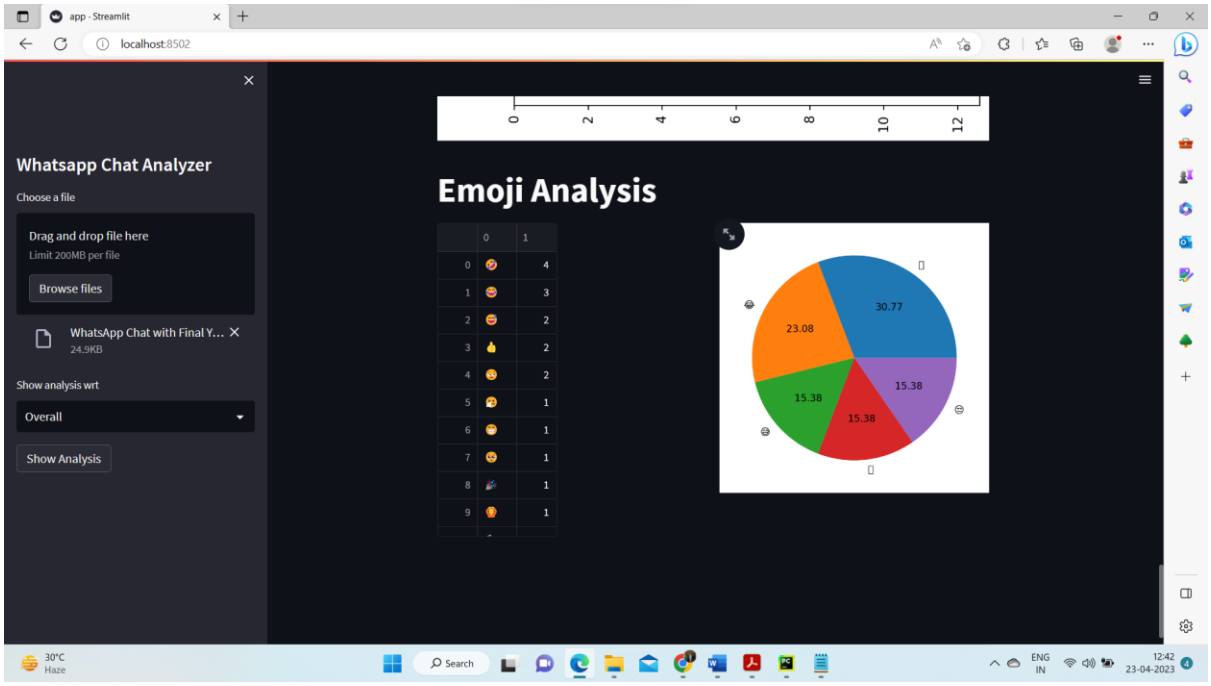
Wordcloud:



Most Common Words:



Emoji Analysis:



13.DISCUSSION

WhatsApp chat analyzer is a useful tool for analyzing WhatsApp chat history and gaining insights from the data. The process of setting up a WhatsApp chat analyzer involves installing the necessary software tools such as Python, Jupyter Notebook, and the required libraries such as Pandas, Matplotlib, and Seaborn. Once the necessary tools are installed, the next step is to import the WhatsApp chat data into Jupyter Notebook and analyze the data using various functions and techniques available in Pandas.

One of the key applications of WhatsApp chat analyzer is in the field of customer service analysis. Businesses can use the tool to evaluate the performance of their customer service team by analyzing response time, satisfaction rate, and the type of queries that customers ask the most. The tool can also be used for social network analysis to understand the structure of a WhatsApp group, identify key influencers, and observe how information flows within the group.

WhatsApp chat analyzer can also be used for personal productivity analysis by analyzing one's own WhatsApp chat history to identify the most frequent contacts, the time of day when they are most active on the app, and the types of messages they send and receive the most. Researchers can also use the tool for social research to study various social phenomena, such as online communication patterns, social networks, and the spread of misinformation.

However, it is important to note that the use of WhatsApp chat analyzer raises some privacy concerns as the tool accesses personal data, including messages and other information shared on the platform. Therefore, it is crucial to ensure that the data is collected and analyzed in accordance with relevant privacy regulations and that the analysis is carried out in an ethical manner.

14.CONSLUSION AND FUTURE SCOPE

14.1 CONSLUSION

The WhatsApp Chat Analyzer project report should conclude by summarizing the objectives, methodology, findings, and limitations of the project.

The conclusion should highlight the importance of the WhatsApp Chat Analyzer tool in helping users extract insights from WhatsApp chats. It should also emphasize the significance of the findings generated by the tool, which can help users better understand communication patterns and dynamics within their WhatsApp groups.

The limitations of the project should also be mentioned, such as the inability of the tool to analyze encrypted chats and the possible inaccuracies resulting from non-standardized formatting of chat data.

Finally, the conclusion should emphasize the potential applications of the WhatsApp Chat Analyzer, such as in social science research, marketing analytics, and business intelligence. It should also provide recommendations for future research and development of the tool, such as improving its accuracy, adding more features, and expanding its compatibility with different chat platforms.

In conclusion, the WhatsApp chat analyzer project is a useful and powerful tool that enables users to analyze their WhatsApp chat history and gain valuable insights. The project involves setting up the necessary software tools, importing the chat data into Jupyter Notebook, and analyzing the data using various functions and techniques available in Pandas.

The WhatsApp chat analyzer project has various applications, including personal productivity analysis, social research, and business analysis such as customer service analysis. It can help users understand communication patterns, identify key influencers, and improve customer service. Moreover, the tool is highly customizable, allowing users to apply various analytical techniques and methods to meet their specific needs.

Overall, the WhatsApp Chat Analyzer project report should highlight the tool's usefulness in generating insights from WhatsApp chats and its potential for further development and application.

14.2 Future Scope

The future scope of WhatsApp chat analyzer projects is vast and has the potential to be extended to various fields. Some of the possible areas where these projects can be useful are:

1) **Social Media Marketing:** Chat analyzer tools can be used to identify the interests and preferences of customers and to tailor marketing campaigns accordingly.
2) **Human Resource Management:** Chat analyzers can be used by organizations to understand the communication patterns of employees and to monitor their performance and productivity.
3) **Healthcare:** Chat analyzers can be used to analyze the chat histories of patients and doctors to identify patterns and to provide personalized care.
4) **Cybersecurity:** Chat analyzers can be used to identify potential threats and vulnerabilities in chat networks.
5) **Education:** Chat analyzers can be used to analyze the chat histories of students and to identify areas where they may need additional support.
6) **Public Opinion Analysis:** Chat analyzers can be used to understand the public opinion on various issues by analyzing conversations on social media.
7) **Sentiment Analysis:** The next step is to perform sentiment analysis on the chat data to identify the emotional tone of the messages. This could involve using natural language processing techniques to classify messages as positive, negative, or neutral.

In addition to the above areas, WhatsApp chat analyzer projects can be extended to various other fields based on the specific requirements and objectives of the project. With the continuous advancements in NLP and machine learning techniques, the accuracy and efficiency of chat analyzers are also expected to improve, leading to more insights and actionable outcomes.

15.REFERENCES

- https://www.irjmets.com/uploadedfiles/paper//issue_5_may_2022/24119/final/fin_irjmets1653279784.pdf
- https://ieeexplore.ieee.org/document/8611043
- https://www.ijltemas.in/DigitalLibrary/Vol.9Issue2/01-05.pdf
- https://www.researchgate.net/publication/350398931_Content_Analysis_of_Whatsapp_Conversations_An_Analytical_Study_to_Evaluate_the_Effectiveness_of_Whatsapp_Application_in_Karachi
- https://www.analyticsvidhya.com/blog/2021/04/whatsapp-group-chat-analyzer-using-python/
- https://github.com/campusx-official/whatsapp-chat-analysis
- https://www.researchgate.net/publication/341876645_Whatsapp_Chat_Analyzer
- https://docs.streamlit.io/
- https://ieeexplore.ieee.org/document/9388596