

WhatsappChat Analyzer using Python

A Project Report

**Submitted in partial fulfilment of the
Requirements for the award of the Degree of
Bachelor of Science in**

INFORMATION TECHNOLOGY

B. K. BIRLA COLLEGE OF ARTS, SCIENCE & COMMERCE (AUTONOMOUS), KALYAN

(Affiliated to University of Mumbai)

MAHARASHTRA – 421301

DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that the project entitled “**WhatsappChat Analyzer using Python**” is bonafied work of **Janhavi Sharad Hindurao** and **Priyanka Pravin Kedare** bearing seat number **3681590** and **3681479** respectively, submitted in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY** from University of Mumbai

Internal Guide

Coordinator

External Examiner

Date:

College Sea

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Janhavi Sharad Hindurao, 3681590, Priyanka Pravin Kedare, 3681479

Under the esteemed guidance of
Mrs. VANDANA MAURYA
Assistant Professors



**DEPARTMENT OF INFORMATION TECHNOLOGY B.K BIRLA
COLLEGE OF ARTS, SCIENCE & COMMERCE (AUTONOMOUS)**

(Affiliated to University of Mumbai)

**KALYAN, 421301
MAHARASHTRA
2022-2023**

Seat No.:

Rollno: 122,57

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

1. Name of Students: **Janhavi Sharad Hindurao**
Priyanka Pravin Kedare

2. Title of the Project : **WhatsappChat Analyzer using
Python**

3. Name of the Guide : **VANDANA MAURYA**

4. Teaching experience of the Guide Assistant
Professors

5. Is this your first submission?

Yes

☐

No

☐

Signature of the Student

Signature of the Guide

Date:

Date:

Signature of the Coordinator

Date:

TABLE OF CONTENT

| | | | |
|------------------|----------------------------------|---|--|
| CHAPTER 1 | INTRODUCTION | | |
| | 1.1 | Background | |
| | 1.2 | Objectives | |
| | 1.3 | Purpose, Scope, and Applicability | |
| | | 1.3.1 Purpose | |
| | | 1.3.2 Scope | |
| | | 1.3.3 Applicability | |
| | 1.4 | Achievements | |
| | 1.5 | Organization of Report | |
| CHAPTER 2 | SURVEY OF TECHNOLOGIES | | |
| | 2.1 | Existing System | |
| | 2.2 | Proposed System | |
| | 2.3 | Requirement Analysis | |
| | | 2.3.1. Matplotlib | |
| | | 2.3.2.Seaborn | |
| | | 2.3.3 Streamlit | |
| | | 2.3.4 NLP | |
| | 2.4 | Software Requirements | |
| | | 2.4.1 Python Programming Language | |
| | | 2.4.2 Pandas for data extraction and preparation | |
| | | 2.4.3 NumPy | |
| | 2.5 | Justification of selection of Technology | |
| CHAPTER 3 | REQUIREMENTS AND ANALYSIS | | |
| | 3.1 | Problem Definition | |
| | 3.2 | Requirements Specification | |

| | | | |
|------------------|-----------------------------------|--|--|
| | | 3.2.1. Functional requirements | |
| | | 3.2.2. Requirements of the Application | |
| | | 3.2.3. Functional requirement | |
| | 3.3 | Planning and Scheduling | |
| | 3.4 | Software and Hardware Requirements | |
| | 3.5 | FEASIBILITY STUDY | |
| | | 3.5.1. TECHNICAL FEASIBILITY | |
| | | 3.5.2. OPERATIONAL FEASIBILITY | |
| | | 3.5.3. ECONOMIC FEASIBILITY | |
| CHAPTER 4 | SYSTEM DESIGN | | |
| | 4.1 | Python: | |
| | | 4.1.a. Python Programming Language | |
| | | 4.1.b. Pandas for data extraction and preparation | |
| | | 4.1.c. NumPy | |
| | | 4.1.d. Matplotlib | |
| CHAPTER 5 | IMPLEMENTATION AND DESIGN | | |
| | 5.1 | Code | |
| | 5.2 | User interface design | |
| CHAPTER 6 | CONCLUSION AND FUTURE WORK | | |
| CHAPTER 7 | REFERENCES | | |

CHAPTER 1 : INTRODUCTION

1.1: Background:

Technology brings us to the virtual era in which information is available and accessible everywhere, and it makes us follow the development of technology for our future life. The development of this era created new product and technology which used by human being to communication along their life. This website or 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 People to communicate by using application which is familiar in human daily communication, for example WhatsApp. WhatsApp Messenger is an application used to send text messages and voice messages, make voice and video calls, stake the images, documents, user locations, and other media. WhatsApp application using two steps for access, first runs on mobile devices which is use cellular mobile number for register. And second, accessible from desktop computers which connected to the Internet, But the research focuses on chatting conversation on WhatsApp, because people will use chatting as communication.

1.2 Objectives:

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.

The purpose of WhatsApp groups is to **establish collective conversations with others**, but when you only place content, but never read or interact, the existence of such groups loses their purpose. No one likes monologues. Never send content, information or “news” that HASN'T been verified.

People form groups to use its numerous benefits. Members of a group **help each other in need, cooperate to reach goals, share resources**, and, last but not least, provide opportunities for social interaction, companionship, and support.

a lot of machine learning enthusiasts develop models which helps solve multiple problems the requirements of appropriate data are very large scale this project aims to provide a better understanding towards various types of chats. This analysis proves to be better input to machine learning models which essentially explore the chat data. These models require proper learning instances which provides better accuracy for these models. Our project ensures to provide an in-depth exploratory data analysis on various types of WhatsApp chats. Communication or conversation is one of types communication, to be successful in communication, conversation must be able cooperative each other.

Cooperation is a term utilized in the linguistic literature to appearance the human conduct in discussion. Person used WhatsApp for communication by talk to other using text message on WhatsApp.

1.3 Purpose, Scope, and Applicability:

1.3.1 Purpose:

In this decade the upcoming technologies are mainly dependent on data. This data can only be obtained if there is some research applied on the context of the requirements of the tool. Since a lot of machine learning enthusiasts develop models which helps solve multiple problems the requirements of appropriate data are very large scale this project aims to provide a better understanding towards various types of chats. This analysis proves to be better input to machine learning models which essentially explore the chat data. These models require proper learning instances which provides better accuracy for these models. Our project ensures to provide an in-depth exploratory data analysis on various types of WhatsApp chats.

1.3.2 Scope:

Data pre-processing, the initial part of the project is to understand implementation and usage of various python-built modules. The above process helps us to understand why different modules are helpful rather than implementing those functions from scratch by the developer. These various modules provide better code representation and user understandability. The following libraries are used such as NumPy, scipy pandas, csv, sklearn, matplotlib, sys, re, emoji, nltk seaborn etc. Exploratory data analysis, first step in this to apply a sentiment analysis algorithm which provides positives negative and neutral part of the chat and is used to plot pie chart based on these parameters. To plot a line graph which shows author and message count of each date, to plot a line graph which shows author and message count of each author, ordered graph of date vs message count, media sent by authors and their count, Display the message which is did not have authors, plot graph of hour vs message count.

1.3.3 Applicability:

There are various methodologies available for analysis but here matplotlib, streamlit, seaborn, re, pandas' libraries of python and some concept of NLP is used. This is the combination of machine learning and NLP. This whatsapp chat analyzer take import whatsapp chat file from user and analyze it and give different visualizations as a result.

CHAPTER 2 : SURVEY OF TECHNOLOGIES

2.1 Existing System:

There is a lot of development in the current system. In the older version there was no feature to display status, there was no feature to share documents and there was no feature to share location. In the current version, all of these features are available. In older version we couldn't share images through doc's format. In this system user is able to access WhatsApp in windows through WhatsApp web application, which can be connected through QR code. There is another feature called export chat where user can send or share or get the chat detail for data analysis through email, Facebook or some messenger application.

2.2 Proposed System:

Data pre-processing, the initial part of the project is to understand implementation and usage of various pythonbuilt modules. The above process helps us to understand why different modules are helpful rather than implementing those functions from scratch by the developer. These various modules provide better code representation and user understandability. The following libraries are used such as NumPy, SciPy pandas, csv, sclera, matplotlib, sys, re, emoji, not seaborn etc. Exploratory data analysis, first step in this to apply a sentiment analysis algorithm which provides positives negative and neutral part of the chat and is used to plot pie chart based on these parameters. To plot a line graph which shows author and message count of each date, to plot a line graph which shows author and message count of each author, ordered graph of date vs message count, media sent by authors and their count, Display the message which is di not have authors, plot graph of hour vs message count WhatsApp-Analyzer uses a number of open source projects to work properly.

2.3 Requirement Analysis:

2.3.1 Matplotlib –

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

A Python matplotlib script is structured so that a few lines of code are all that is required in most instances to generate a visual data plot. The matplotlib scripting layer overlays two APIs:

- The pyplot API is a hierarchy of Python code objects topped by *matplotlib.pyplot*
- An OO (Object-Oriented) API collection of objects that can be assembled with greater flexibility than pyplot. This API provides direct access to Matplotlib's backend layers.

Matplotlib was used for the data visualization of this system. It is a standard Python library used for creating 2D plots and graphs. It was imported in this work to create a graph of the dataset.

2.3.2 Seaborn –

Seaborn data visualization library was also imported in this work. It builds on Matplotlib's foundations. Being a higher-level library, it was able to expand the plot and better beautify it. It doesn't work alone hence it works on Matplotlib foundation.

2.3.3 Streamlet -

Streamlit is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc.

This library is used for creating beautiful web items and objects for representing WhatsApp chat analysis with different types of charts and visualizations on Streamlet.

2.3.4 NLP-

In this project, Features of NLP are used like Parsing Text, eliminating stop words and Analysing Text. Parsing text is used for splitting messages into words for analysis like total words and mostly used words. A file is used that contains all stop words which is given to the python program to show meaningful words only by eliminating all stop words. Text analysis is used to identify how many media are shared; how many links are shared.

[Urlextract](#) - URLExtract is python class for collecting (extracting) URLs from given text based on locating TLD.

urlextract is package with python class and command line script used for extraction of URLs from given text.

[WordCloud](#) - visual representations of words that give greater prominence to words that appear more frequently.

[Emoji](#) - Emoji for Python.

2.4 Software Requirements

2.4.1 Python Programming Language –

It is an interpreted, high-level general-purpose programming language. Created by Guido Van Rossum and first released in 1991. Its language constructs and objects-oriented approach aim to help programmer with clear, logical code for small and large-scale tools. Python is used for web development (server-side), software development, mathematics, it can be used alongside software to create workflows, it can connect to database systems, it can also read and modify files, it can be used to handle big data and perform complex mathematics and can be used for rapid prototyping, or for production-ready software development. Python is the programming language used for this work. It is a free opensource 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.

2.4.2 Pandas for data extraction and preparation –

pandas is an open source, library providing high-performance, easy-to-use data structures and data analysis tools for the Python

Pandas is a Python library that provides high-level data structures which are simple to use as well as intuitive. It was the tool that enabled the extraction of the data to be analysed. It was used to fetch the dataset in Python from the CSV, Excel, JSON files and manipulated the data to perform operations on it.

2.4.3 NumPy –

Python (NumPy) was the Python library used to handle the multidimensional arrays and functions that were needed for the classification of the chats into days, hours, minutes and seconds. In Python we have lists that serve the purpose of arrays, but they are slow to process. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called **ndarray**, it provides a lot of supporting functions that make working with **ndarray** very easy. Arrays are very frequently used in data science, where speed and resources are very important.

2.5 Justification of selection of Technology

The proposed system is developed by using Jupyter software. Jupyter is non-profit organization created to develop open-source software, open standards, and services for interactive computing across dozens of programming languages. The idea is to implement a data processing code using python to make better sense of WhatsApp group chat data. In economic feasibility, the most important is cost-benefit analysis. This project is not economical as it mainly depends on the sharing of data between two phones.

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem Definition:

WhatsApp-Analyzer is a statistical analysis tool for WhatsApp chats. Working on the chat files that can be exported from WhatsApp it generates various plots showing, for example, which another participant a user responds to the most. We propose to employ dataset manipulation techniques to have a better understanding of WhatsApp chat present in our phones.

WhatsApp has been the most used mode of communication and has been an efficient one too. It consists of many conversations in groups and individuals. So, there might be some hidden facts in them. This project takes those chats and provide a deep analysis of that data.

3.2. Requirements Specification

3.2.1. Functional requirements:

The functional requirements are organized in two sections First requirements of the application on any device and second requirements of the network resource to connect to the internal servers of WhatsApp. The requirements for the working of WhatsApp Application are organized in the following way. General requirements, requirements for authorization, and requirements for a transaction.

3.2.2. Requirements of the Application:

The internal servers get a request from a user account to process and send a message to a recipient. Functional requirement.

1: Description:

Internal servers checking if the user is authentic.

2.Input:

Request from the application about its authenticity.

3.Processing:

Check if the application is a valid application or not

4.Output:

Valid or invalid application

3.2.3. Functional requirement:

1.Description:

If the application is updated or not

2.Input:

Application Version

3.Processing:

Process Application Version

4.Output:

The Application gets the message “the” if it’s an older and unsupported application version.

3.2.4. Functional requirement:

1.Description:

The Application checks for user data after every update.

2.Input:

Check internal encrypted backup for user data

3.Processing: Check

User Data

4.Output:

Accepts User data or prompts to register again.

3.2.5. Functional requirement:

1.Description:

Changes in encryption codes

2.Processing:

Checks for changes in security encryption code.

3.Output:

Prompt regarding encryption changes.

3.3. Planning and Scheduling:

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. This document provides a scalable scheduling tool and associated schedule development, analysis, and monitoring methods to prepare, monitor, and report project schedules. Our Project is not that complex so we will not use very complex scheduling method.

[illegible]

3.4. Software and Hardware Requirements:

If software requirement analysis in the field of systems engineering and software engineering, encompasses those tasks that are used for a new or altered product or tool, taking account of the possibly conflicting requirements of the various stakeholders, documenting, validating and managing software or system requirements.

3.5. FEASIBILITY STUDY:

The main objective of the feasibility study is to treat the technical operational and economic feasibility of developing the application. Feasibility is the determination of whether or not project is worth doing. The feasibility study to be conducted for this project involves:

3.5.1. TECHINAL FEASIBILTY:

It is the measure of the specific technical solution and the availability of the technical resources and expertise. It is one of the first studies that must be conducted after tool has been identified. The proposed system is developed by using Jupyter software. Jupyter is nonprofit organization created to develop open-source software, open standards, and services for interactive computing across dozens of programming languages. The proposed system is developed by using Jupyter software. Jupyter is non-profit organization created to develop open-source software, open standards, and services for interactive computing across dozens of programming languages. The idea is to implement a data processing code using python to make better sense of WhatsApp group chat data.

3.5.2. OPERATIONAL FEASIBILITY:

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented, whether there will be resistance from the users which will affect the possible application benefits. It is the ability to utilize, support and perform the necessary tasks of a system or program. This system helps in many ways. It shows the number of users using WhatsApp and gives the data information of their sharing data. Which is organized in Pie-chart and Bar- chart.

3.5.3. ECONOMIC FEASIBILITY:

Economic feasibility is the most frequently used method for evaluating the effectiveness of the new system. Economic feasibility is the measure of the cost effectiveness of an information system solution. Without a doubt, this measure is most often and important one of the three. Information systems are often viewed as capital investments for the business, and, as such should be subjected to the same type of investment analysis as other capital investments. Economic analysis is used for evaluating the effectiveness of the proposed system. In economic feasibility, the most important is cost-benefit analysis. This project is not economical as it mainly depends on the sharing of data between two phones.

WhatsApp uses a protocol known as the XMPP (Extensible Messaging and Presence Protocol). This protocol is responsible for handling the message delivery task. According to, XMPP is an open extensible Markup Language (XML). XML, was designed to store and transport decamp is an XML technology for real-time communication, which powers a wide range of applications including instant messaging, presence and collaboration. The acronym XMPP can be explained as follows: Protocol which is a set of standards that allows systems to talk to each other. The protocol which stands for Presence, tells the servers that you are online, offline or busy. M — Messaging. The 'messaging' part of XMPP is 'what' you see; which is the Instant Message (IM) sent between clients. XMPP has been designed to send

all messages in real-time using a very efficient push mechanism. And then, the X which stands for extensible, explains the fact that it is an open standard and using an open systems approach of development and application, XMPP is designed to be extensible. In other words, it has been designed to grow and accommodate changes. Also, WhatsApp was built using Erlang. Erlang is a programming language used to build massively scalable, soft real-time systems with requirements on high availability. Erlang's runtime system has built-in support for concurrency, distribution and fault tolerance [11]. Erlang capabilities is what enables bug fixes and frequent updates in WhatsApp. Furthermore, WhatsApp also uses a technology known as Amnesia. Amnesia is a multi-user distributed database management system which enables quick response to requests. Amnesia works with the Erlang Runtime System. The Amnesia relational and object hybrid data model is what makes it suitable for developing distributed applications of any scale. In addition, WhatsApp uses Bootstrap front-end framework. The data analysis stage involves the Process of cleaning, transforming, inspecting and modelling the data that was used for this system. The goal of this work was to gather useful information about effective and functional users are in a WhatsApp group. Data analysis is a process for acquiring raw data and transforming it into information useful for decision making by users. In this work, the sample data that was analysed was gotten from the Unific Staff Community WhatsApp group chat. The groups were 3 in number as at the time of this research, with 256 members in each group. The objective of this work was to get the most active users in any WhatsApp group, to discover the most active days of week, to find the top 10 or top 20 most active users or more, as the case maybe, to also the activities of each user and the number of messages sent by the users of the group and also the analyses the word count of users on the platform.

Chapter 4: SYSTEM DESIGN

4.1. Python:

It is an interpreted, high-level general-purpose programming language. Created by Guido Van Rossum and first released in 1991. Its language constructs and objects-oriented approach aim to help programmer with clear, logical code for small and large-scale tools. Python is used for web development (server-side), software development, mathematics, it can be used alongside software to create workflows, it can connect to database systems, it can also read and modify files, it can be used to handle big data and perform complex mathematics and can be used for rapid prototyping, or for production-ready software development.

This stage involves the point where the WhatsApp data is collected. This was done by visiting the chat group to be analysed, to export the WhatsApp data file that was used. The procedures involved, visiting the WhatsApp group page, clicking on the settings, select export data and then select either add media or without media. This simply means to know whether you intend to export the file with the media or not. Note that exporting with the media, will lead to use of larger volume of data and waste of time for data collection.

4.1.a. 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.

4.1.b. Pandas for data extraction and preparation –

Pandas is Python library that provides high-level data structures which are simple to use as well as intuitive. It was the tool that enabled the extraction of the data to be analysed. It was used to fetch the dataset in Python from the CSV, Excel, JSON files and manipulated the data to perform operations on it.

4.1.c. NumPy –

Numerical Python (NumPy) was the Python library used to handle the multidimensional arrays and functions that were needed for the classification of the chats into days, hours, minutes and seconds.

4.1.d. Matplotlib –

Matplotlib was used for the data visualization of this system. It is a standard Python library used for creating 2D plots and graphs. It was imported in this work to create a graph of the dataset.

4.1.e. Seaborn –

Seaborn data visualization library was also imported in this work. It builds on Matplotlib's foundations. Being a higher-level library, it was able to expand the plot and better beautify it. It doesn't work alone hence it works on Matplotlib foundation.

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1. CODE

1.preprocessor.py:

```
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.m\s-\s'
    #'\d{1,2}/\d{1,2}/\d{2,4},\s\d{1,2}:\d{2}..\s-\s'

    #fixing unicode
    data = data.replace("\u202f", "")

    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,
    %I:%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
```

2.app.py:

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

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)
    # 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)
prop = FontProperties(fname='C:\Windows\Fonts\seguisym.ttf')

plt.rc('axes', unicode_minus=False)

plt.rcParams['font.family'] = prop.get_family()

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 common 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()
    patches, texts, autotexts =
ax.pie(emoji_df[1].head(), labels=emoji_df[0].head(), autopct="%0.2f")
plt.setp(autotexts, fontproperties=prop)
plt.setp(texts, fontproperties=prop)
st.pyplot(fig)
```

3.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']:
    # emojis.extend([c for c in message if c in emoji.UNICODE_EMOJI['en']])
    emojis.extend([e['emoji'] for e in emoji.emoji_list(message)])

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()

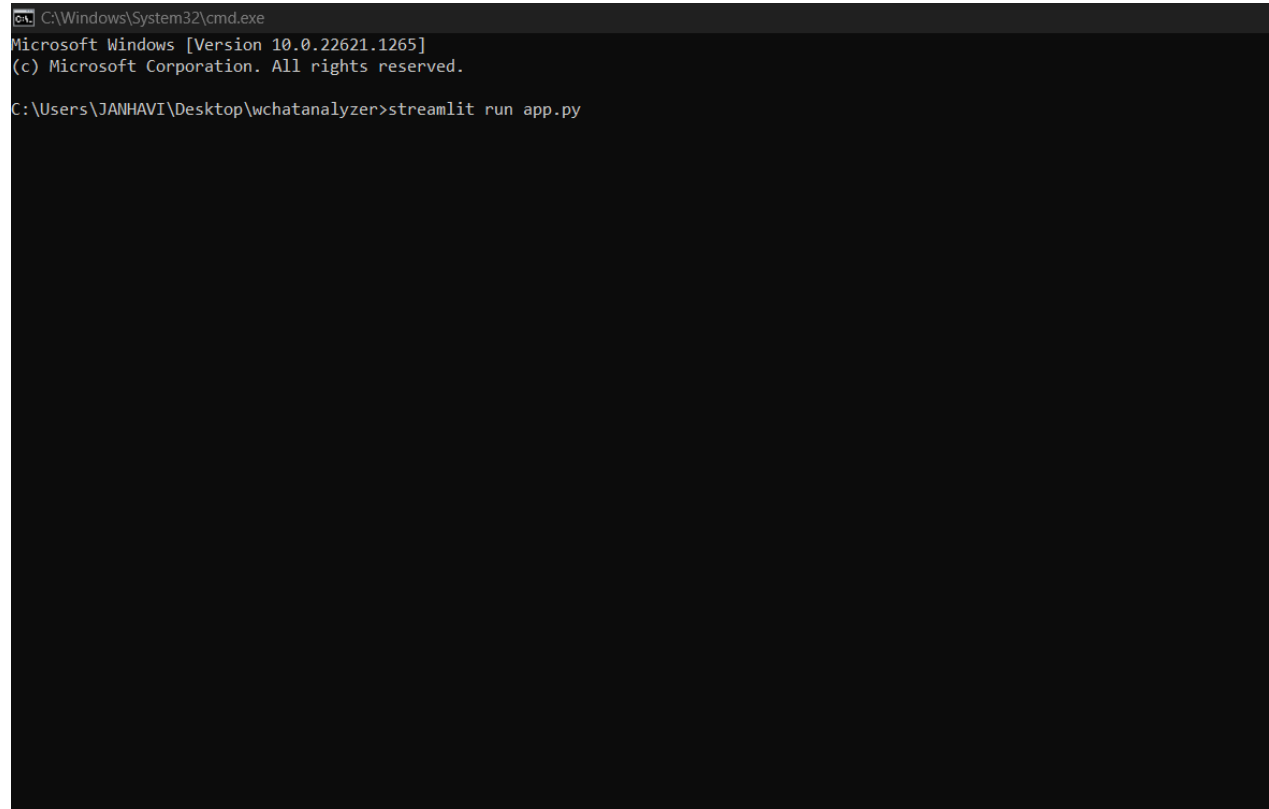
def activity_heatmap(selected_user,df):

    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
```

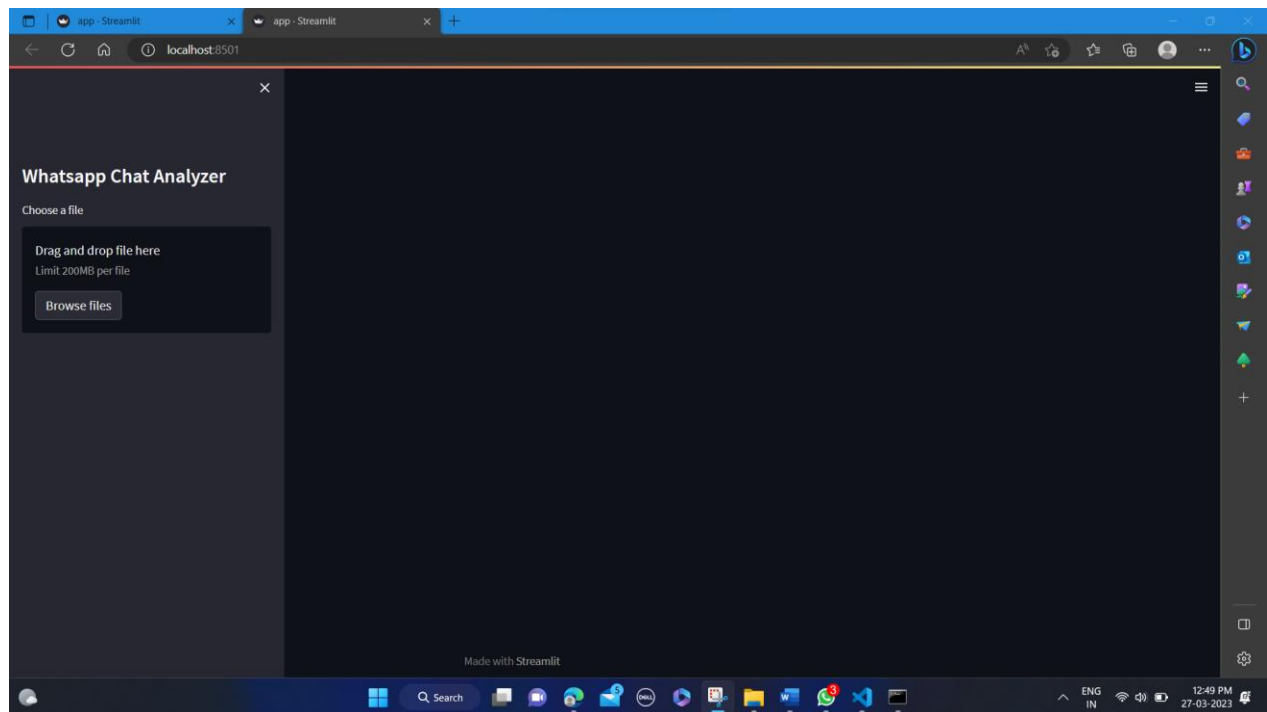
5.2. USER INTERFACE



```
cmd C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.1265]
(c) Microsoft Corporation. All rights reserved.

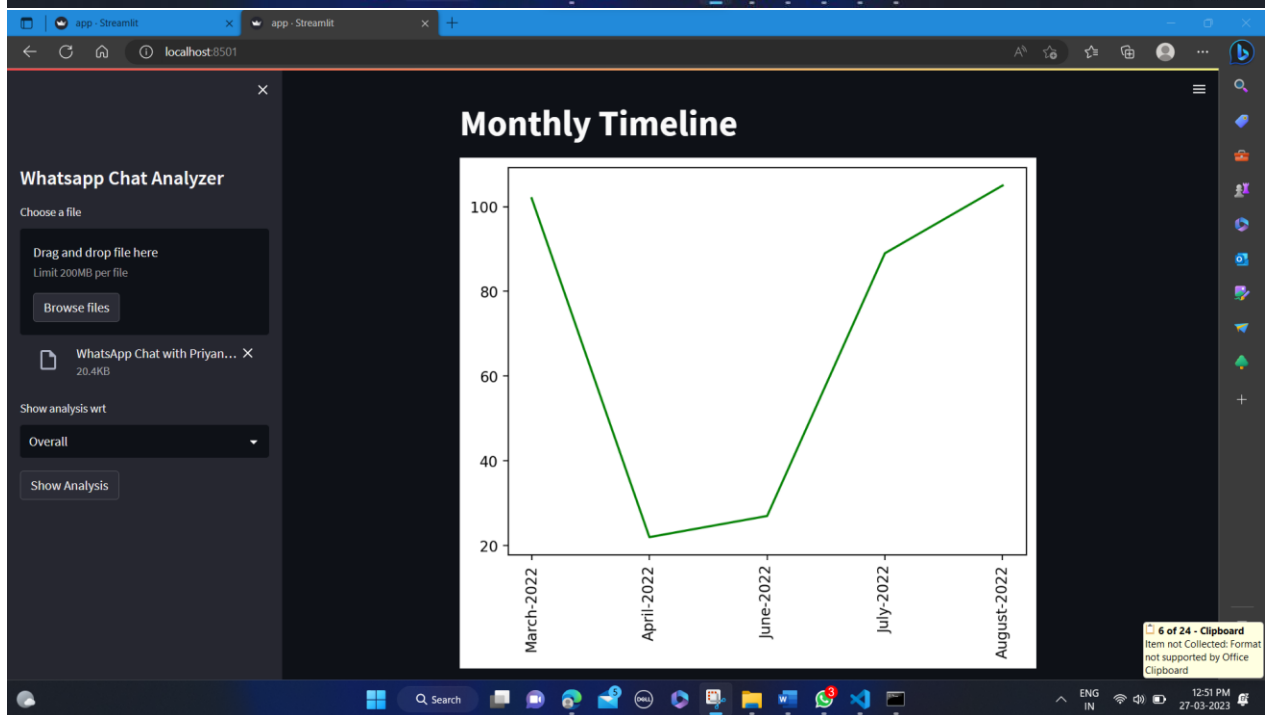
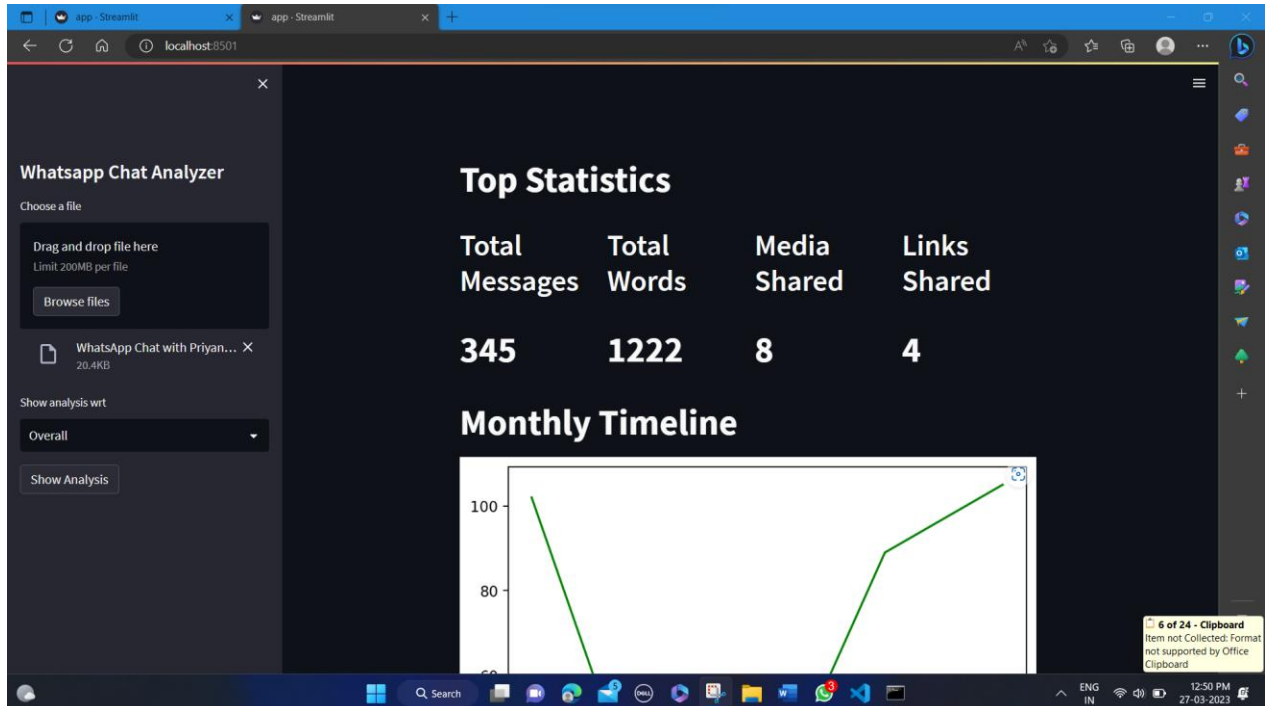
C:\Users\JANHAVI\Desktop\wchatanalyzer>streamlit run app.py
```

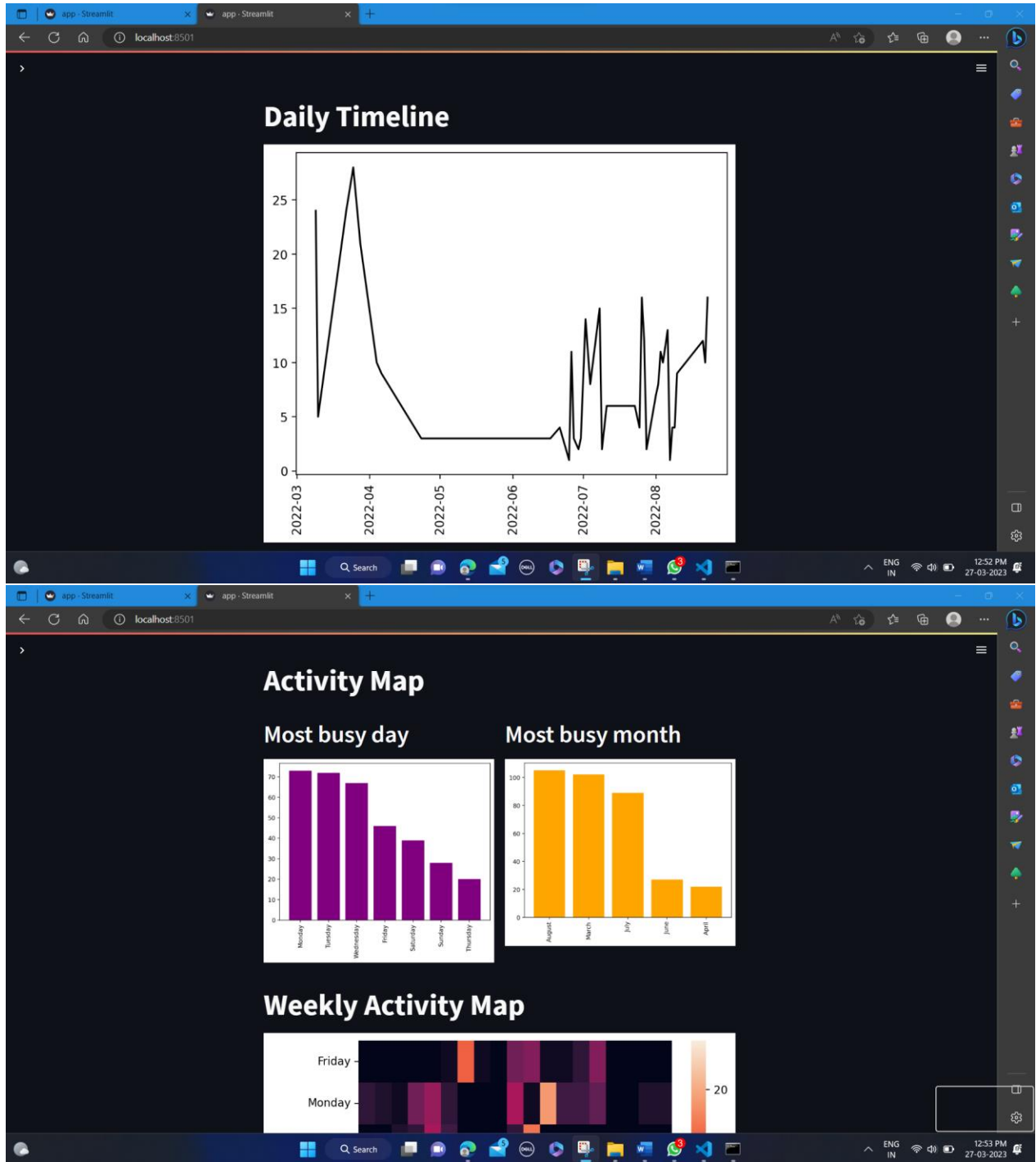
The image shows a Windows command prompt window. The title bar indicates the path C:\Windows\System32\cmd.exe. The window content shows the Microsoft Windows version (10.0.22621.1265) and copyright information. The user is in the directory C:\Users\JANHAVI\Desktop\wchatanalyzer and has executed the command streamlit run app.py.

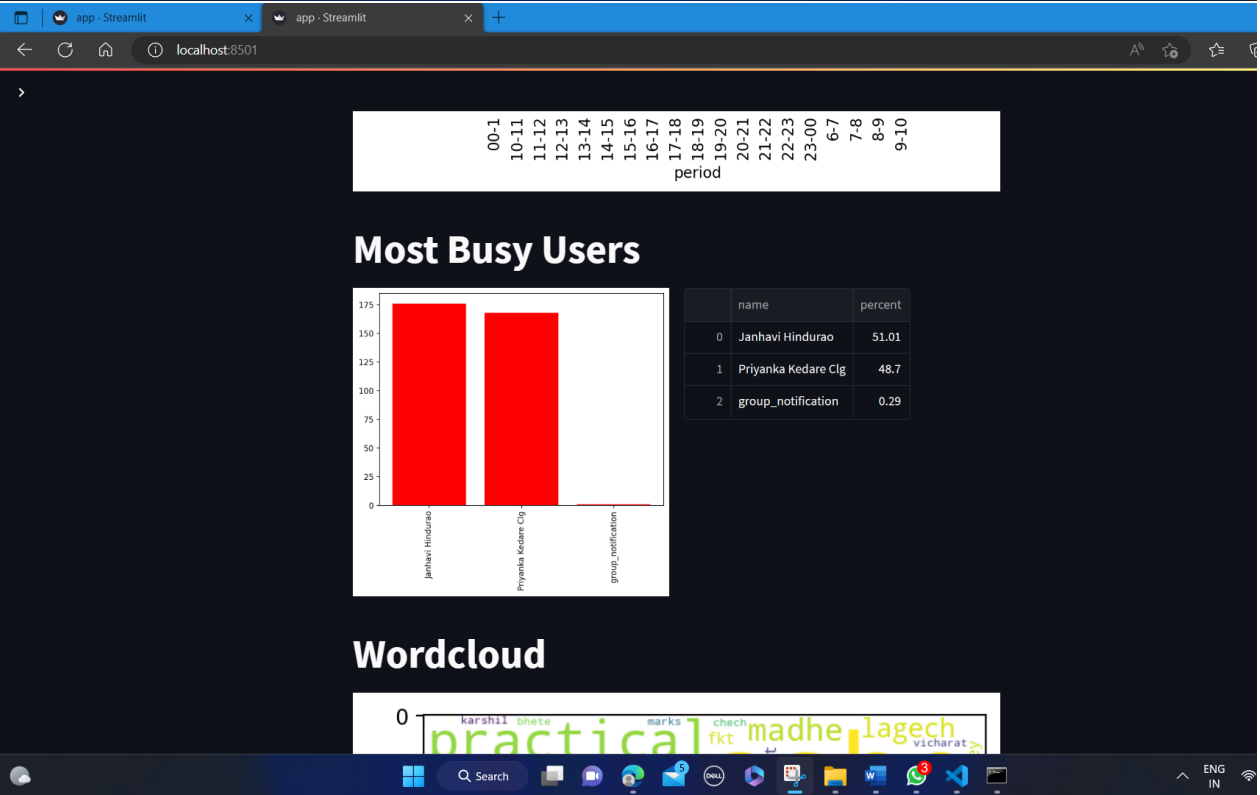
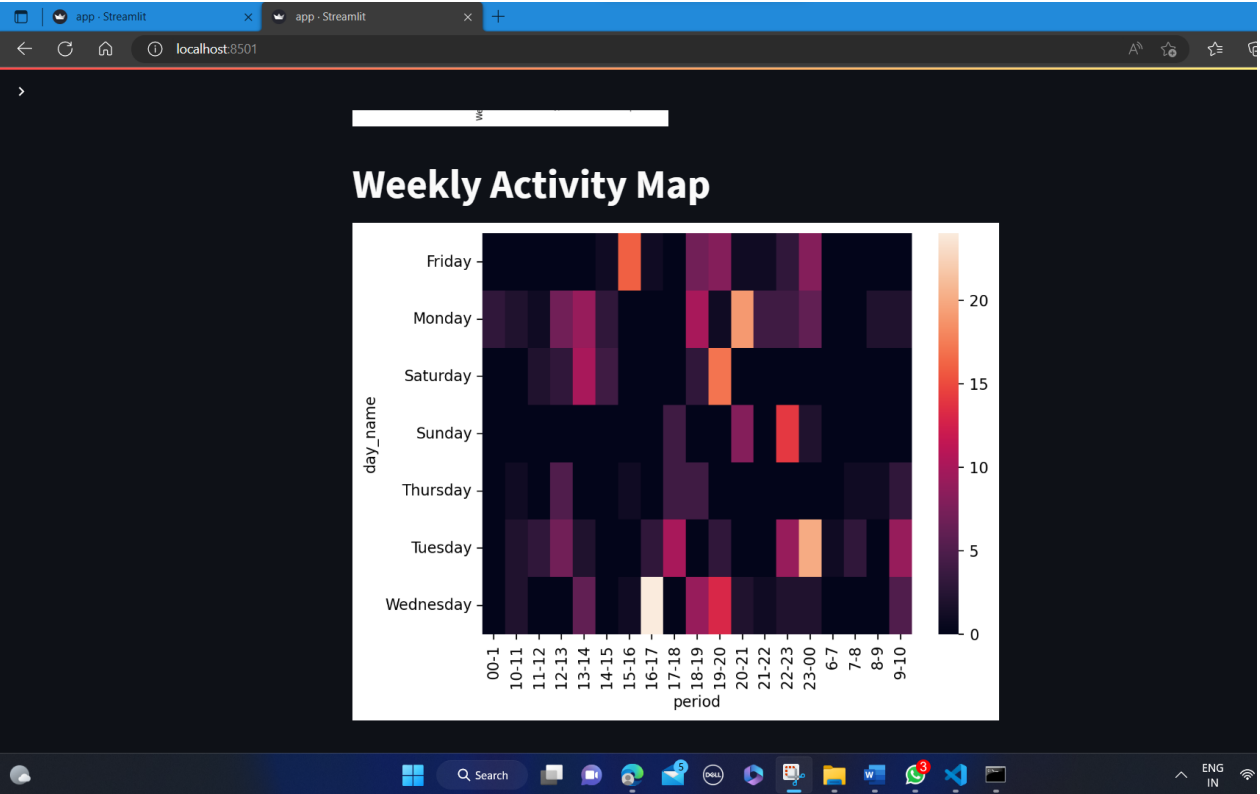


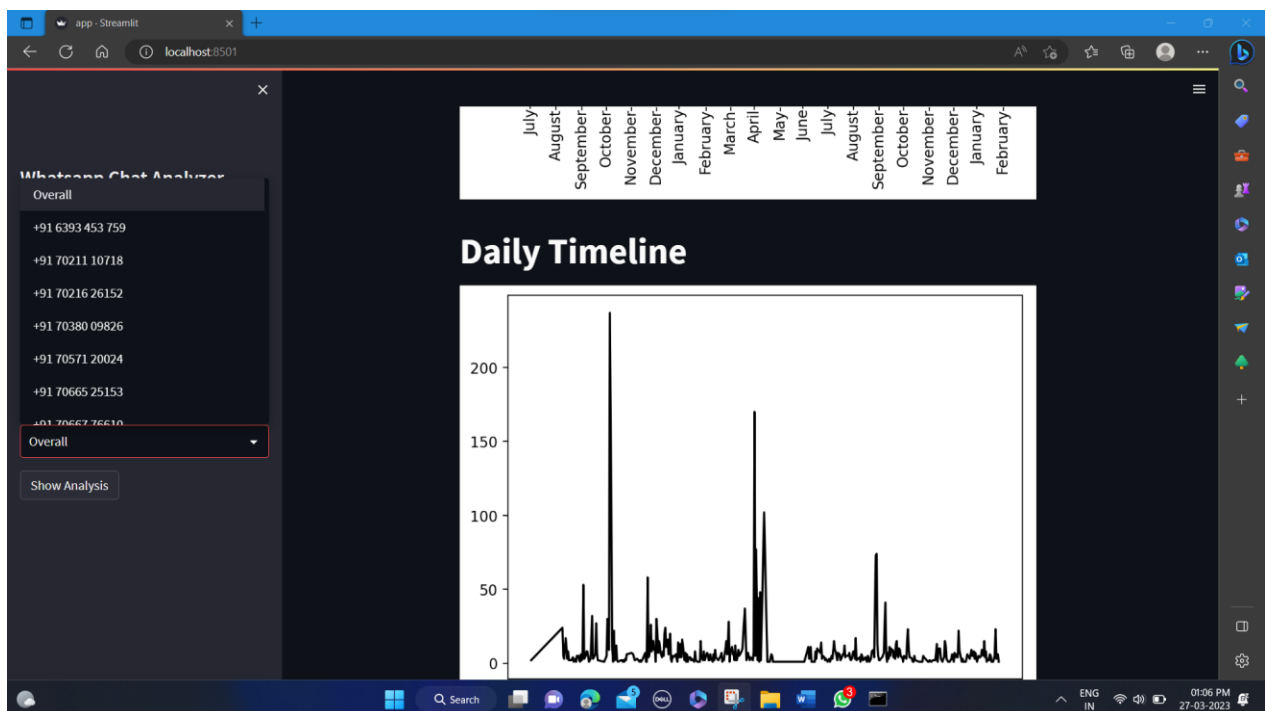
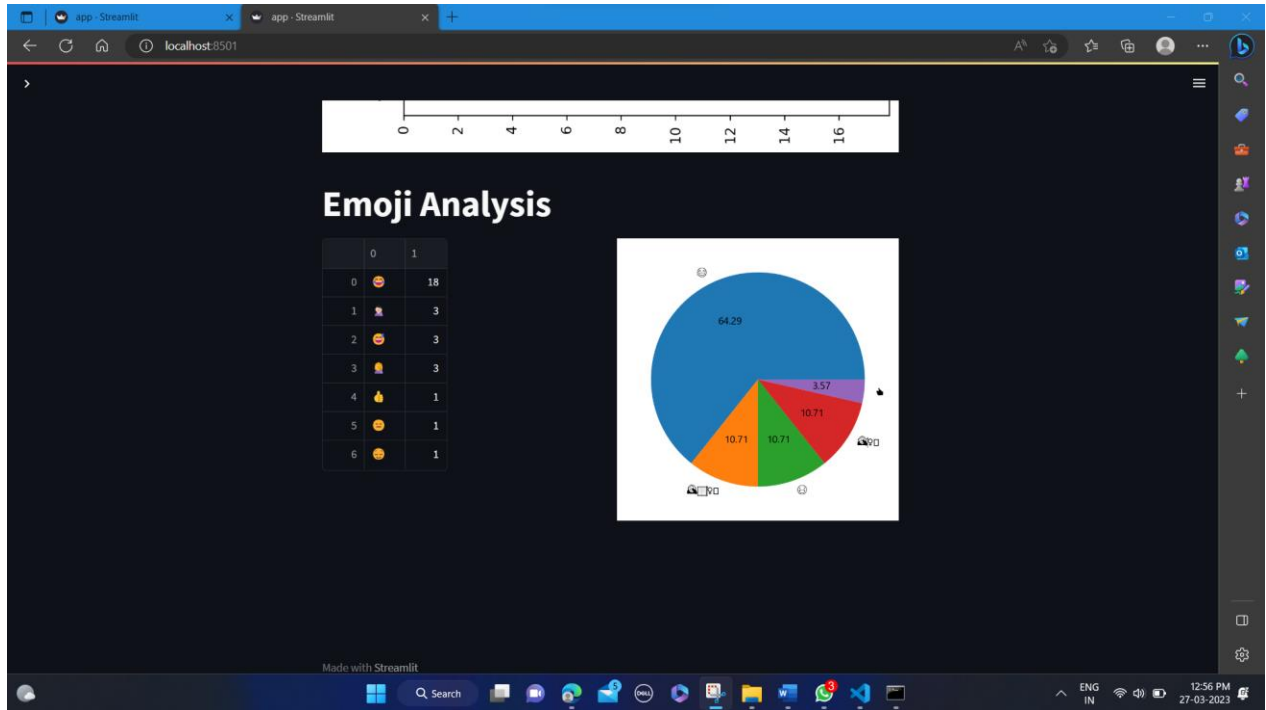
1.INDIVISUAL CHATS:-

Top statistics:



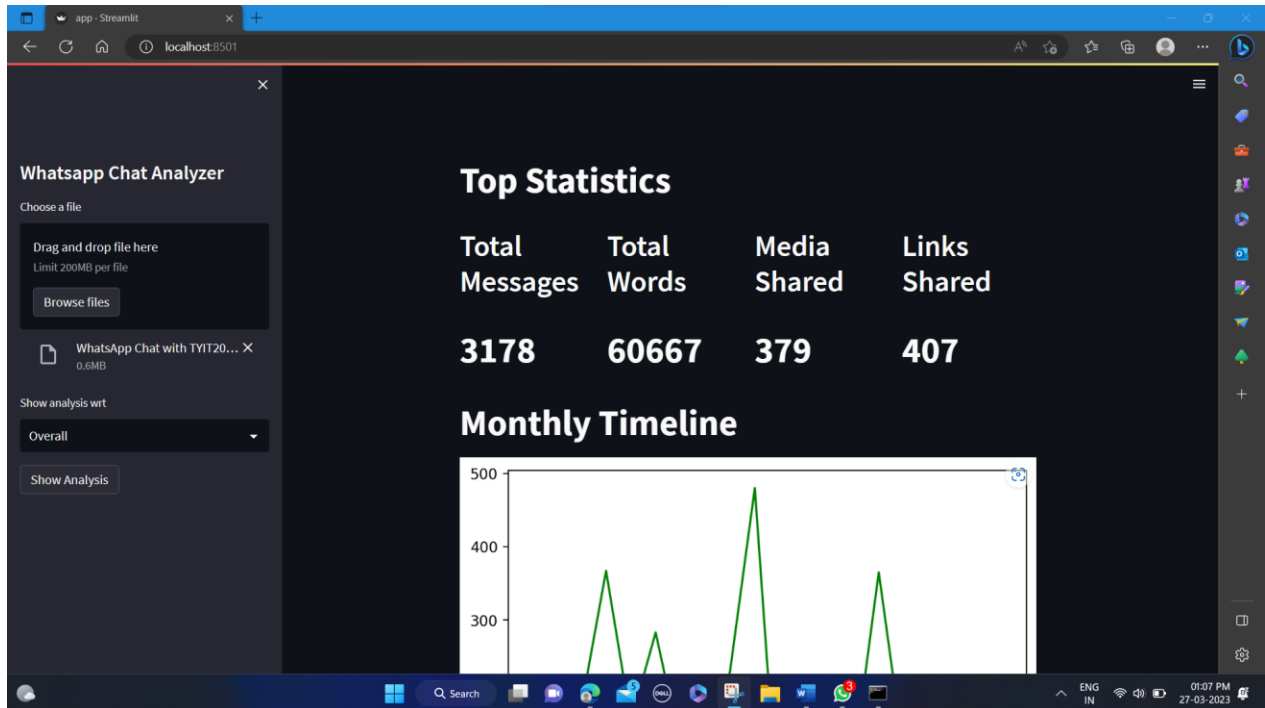


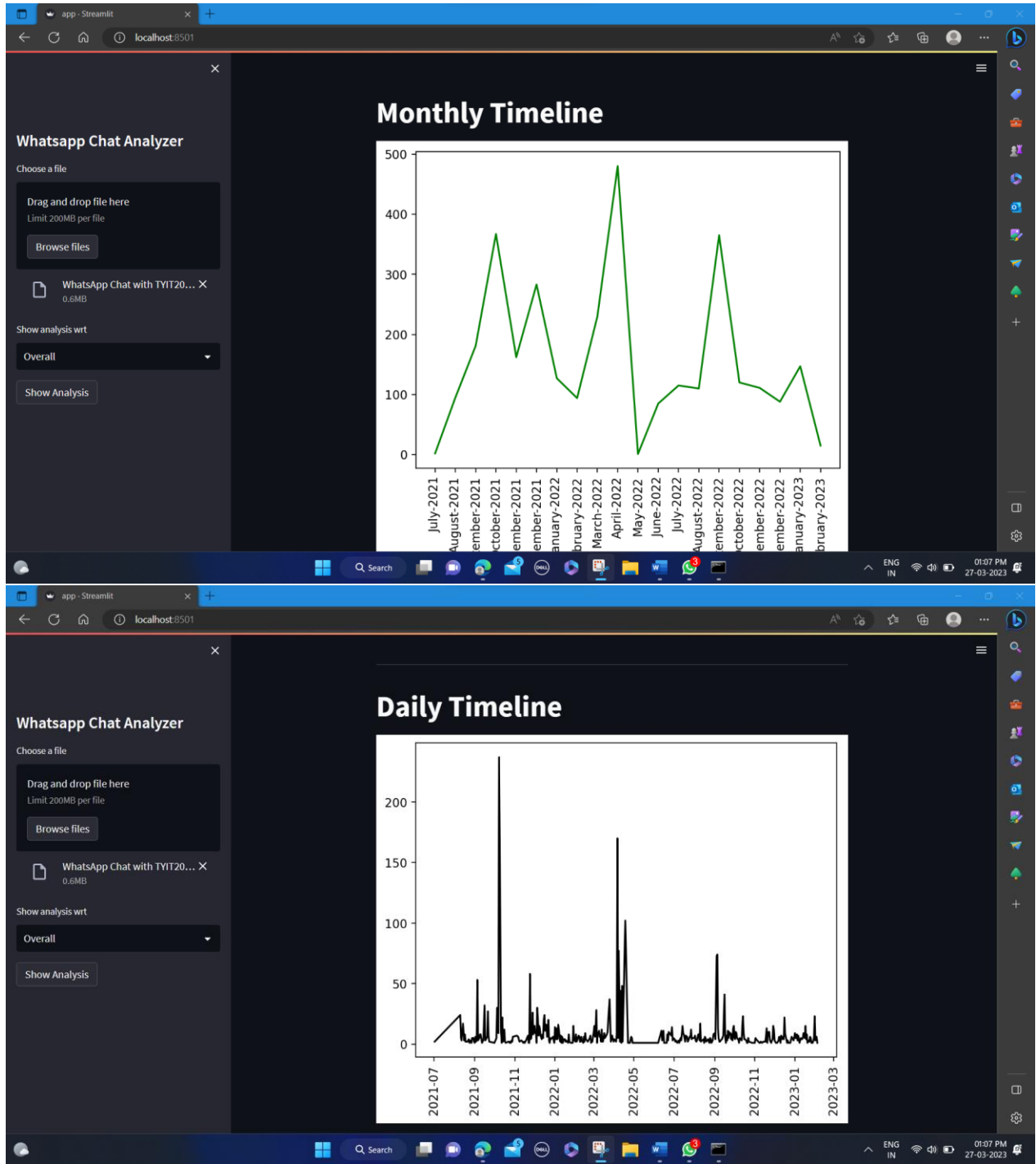


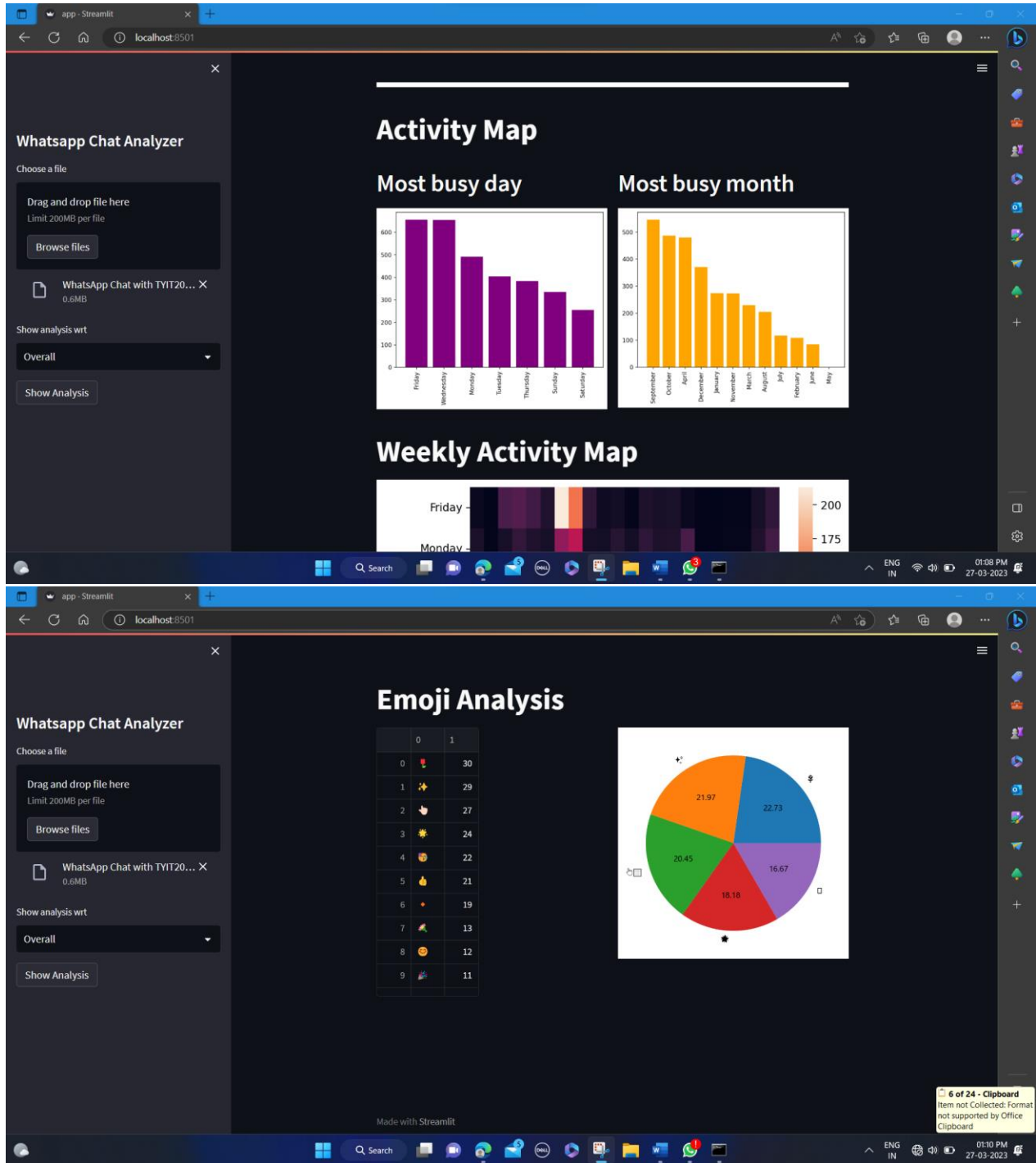


2.GROUP CHATS :-

Upload chats files and show analysis:







RESULTS AND DISCUSSION:

This project is created in python using streamlit and deployed on web. Working of project:

1. User go to sidebar and click on browse file .
2. Select whatsapp chat text file and import it for analysis .
3. User have choice for overall analysis or specific user analysis from whole group.
4. After selecting user, User click on show analysis button to analyze imported file.

5. It shows analysis of imported whatsapp text file.
6. User can see Total messages, words, media and link shared in the group.
7. then monthly and daily timeline for the message is shown using line charts.
8. Activity Map in which most busy month and day is shown by the bar charts.
9. then weekly activity map which shows hourly activity of users with corresponding day using heat map.
10. Top five busy users in group using graph and list of users with percentage of use.
11. WordCloud shows an interesting visualization of most common words.
12. Top twenty most common word represented by using bar chart.
13. List of Emojies with number of times it is used.
14. Pie chart which shows top five emojies percentage of use. This is the result of project and how project is working.

Chapter 6 :Conclusion

The major objective that has been decided in the initial phase of the requirement analysis is achieved successfully. After the implementation, the system provides reliable results. The system is totally menu and user friendly, which makes it easy for the users even with limited knowledge of computer environment to operate the developed system. The system avoids the drawbacks of the existing manual system and the validation facility of the system totally eliminates the chances of wrong data entry.

It has following features:

- User friendly.
- Time saving.
- Runs on any devices.
- Analyzes any WhatsApp imported file.
- Accuracy.
- Reliability.
- Easy to use.

Chapter 7 : references

[1] Available from: <http://www.statista.com/statistics/260819/numberof-monthly-active-WhatsApp-users>. Number of monthly active WhatsApp users worldwide from April 2013 to February 2016(in millions)

[2] [WhatsApp Chat Analysis Project | End to End Project with Heroku Deployment](#)



[3] J. Yeboah and G. D. Ewur, "The Impact of Whatsapp Messenger Usage on Students," Journal of Education and Practice, vol. Vol 5, no. 6, pp. 157-164, 2014

[4] WhatsApp, "About WhatsApp," 20 April 2019. [Online]. Available: <https://www.whatsapp.com/about/>.

