Title: IQchat
Author: Sagar Rajak, NIT Trichy
GitHub: https://github.com/Sagarrajak01/IQchat

Abstract — With the rapid digitization of communication, instant messaging platforms have become a key part of human interaction. WhatsApp generates massive volumes of unstructured chat data daily, which can reveal communication behaviour, temporal trends, and engagement dynamics. IQchat is a Python-based analytical system that extracts, cleans, and visualizes WhatsApp chat data using pandas and regular expressions. It transforms exported text chats into structured datasets and generates interactive visual insights through matplotlib and seaborn. The implementation, source code, and sample datasets are available on GitHub.

Keywords — WhatsApp, Chat Analysis, Python, pandas, regex, Visualization, Data Analytics

I. INTRODUCTION

WhatsApp is among the most widely used messaging applications, with billions of active users worldwide. Despite the vast amount of data exchanged daily, WhatsApp provides limited inbuilt analytics. IQchat aims to address this gap by enabling structured analysis of exported chat data. It provides metrics such as user activity distribution, daily and hourly message trends, and message-type ratios, enabling users to understand communication patterns in groups or personal chats.

II. RELATED WORK

Existing works like WhatAnalyzer and Chat provide basic visualization of WhatsApp chat data but lack flexible preprocessing or customization. IQchat extends functionality by integrating robust regex parsing, modular pandas-based processing, and dynamic visualizations. It allows handling of various chat formats and integrates seamlessly with extensions like NLP-based sentiment analysis.

III. SYSTEM ARCHITECTURE

IQchat consists of four primary modules: data input, regex-based parsing, analytical processing, and visualization.

Figure 1: System Architecture Diagram

- Input: WhatsApp Chat Export (.txt)
- Processing Layer: Regex Parsing → Data Cleaning → pandas DataFrame
- Analytics Layer: Statistical Computation → Visualization
- Output Layer: Graphs, Heatmaps, Word Clouds
  Each component is represented in boxes connected with arrows to show data flow.

IV. IMPLEMENTATION

The system is implemented using Python 3.10, leveraging open-source libraries such as pandas, NumPy, matplotlib, seaborn and re (regex).

Figure 2: Workflow Diagram
Flowchart sequence:
Raw Chat Export → Data Cleaning → Message Parsing → Statistical Analysis → Visualization → Report Generation.

Regex Parsing Example:

```
r'(\d{1,2}/\d{1,2}/\d{2,4},?\s\d{1,2}:\d{2}\s?[APap][Mm]\s-\s)'
```

This allows accurate identification of timestamp, sender, and message content.

V. DATA PROCESSING AND CLEANING

System notifications, empty lines, and placeholders like are filtered. The data is transformed into a panda Data Frame for analysis.

Figure 3: Data Parsing Example

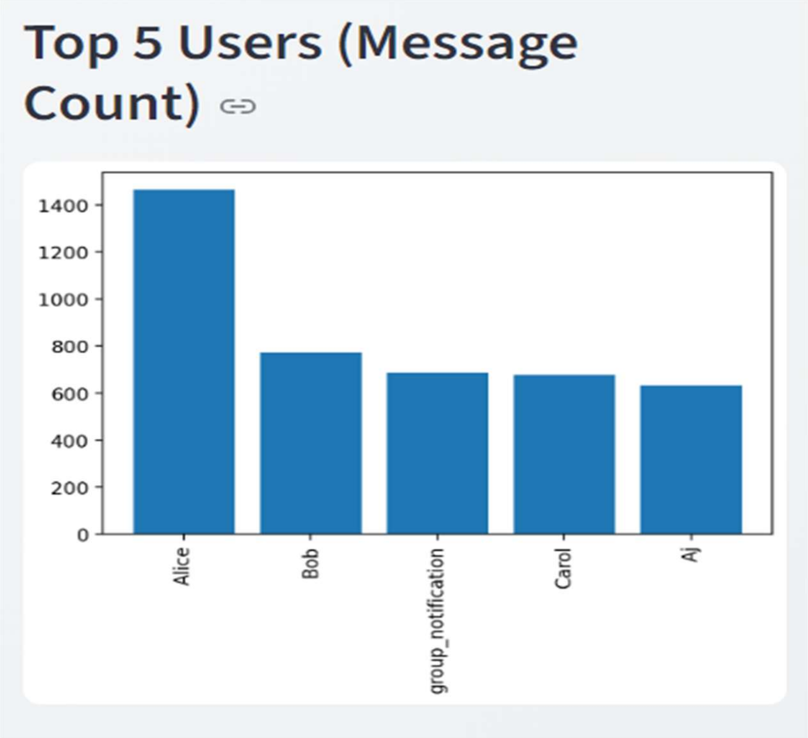
Date	Time	Sender	Message
12/03/2025	9:15 pm	John	Hello everyone!

VI. ANALYSIS AND VISUALIZATION

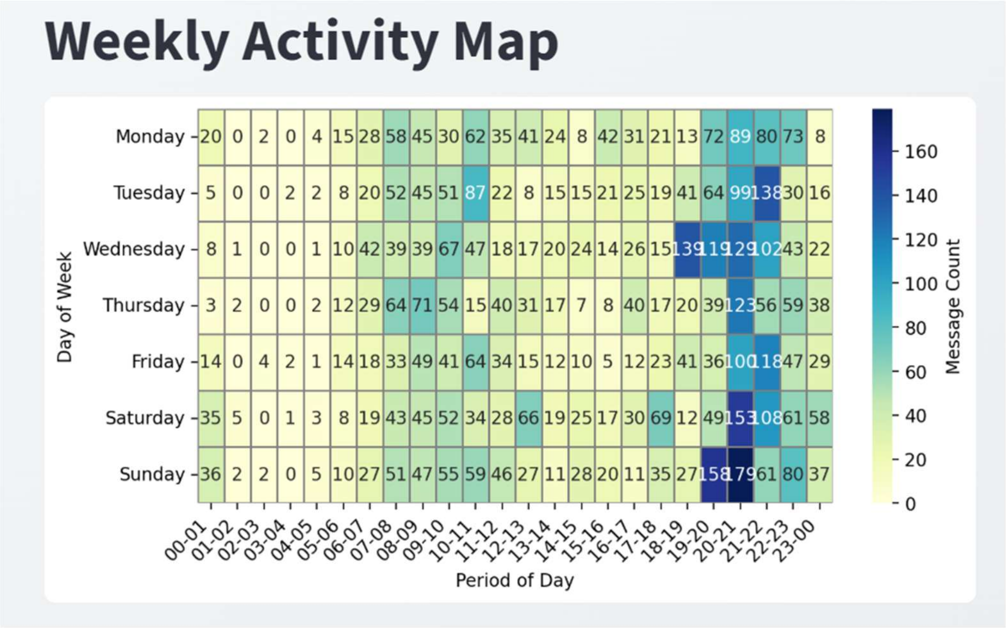
IQchat produces both tabular and graphical summaries. It computes message frequency, top participants, daily and hourly message distribution, and most frequent words. Visualizations generated include bar charts, word clouds, and activity heatmaps.

Figure 4: Graph and Output Samples
Include:

- (a) Bar chart: Top 5 most active users



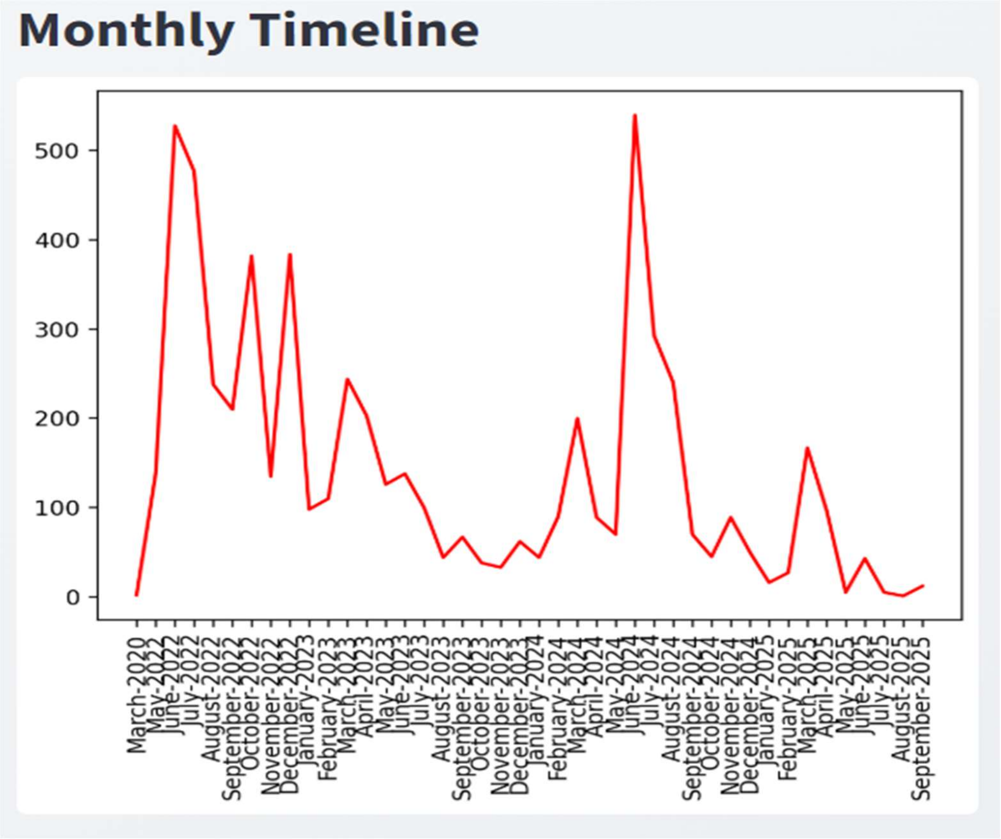
- (b) Heatmap: Daily vs Hourly activity



Weekly Activity Map

Day of Week	00-01	01-02	02-03	03-04	04-05	05-06	06-07	07-08	08-09	09-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21	21-22	22-23	23-00
Monday	20	0	2	0	4	15	28	58	45	30	62	35	41	24	8	42	31	21	13	72	89	80	73	8
Tuesday	5	0	0	2	2	8	20	52	45	51	87	22	8	15	15	21	25	19	41	64	99	138	30	16
Wednesday	8	1	0	0	1	10	42	39	39	67	47	18	17	20	24	14	26	15	139	119	129	102	43	22
Thursday	3	2	0	0	2	12	29	64	71	54	15	40	31	17	7	8	40	17	20	39	123	56	59	38
Friday	14	0	4	2	1	14	18	33	49	41	64	34	15	12	10	5	12	23	41	36	100	118	47	29
Saturday	35	5	0	1	3	8	19	43	45	52	34	28	66	19	25	17	30	69	12	49	153	108	61	58
Sunday	36	2	2	0	5	10	27	51	47	55	59	46	27	11	28	20	11	35	27	158	179	61	80	37

- (c) Monthly Timeline



Monthly Timeline

VII. RESULTS AND DISCUSSION

The tool was tested on chat datasets ranging from 1,000 to 100,000 messages. Parsing speed averaged under 10 seconds for 50,000 lines on a standard machine. Key insights include:

- Most active users and participation ratio
- Message volume trends across days and hours
- Media vs text communication ratio
- Frequently used words and emojis

VIII. CONCLUSION AND FUTURE WORK

IQchat provides an efficient, scalable solution for analysing WhatsApp chat data. It leverages Python's data science ecosystem to deliver accurate insights and engaging visuals. Future enhancements will focus on integrating natural language processing for sentiment and emotion analysis, multilingual chat support, and real-time dashboard deployment.

REFERENCES

1. WhatsApp Inc., "Export chat history," Meta Platforms, 2024. [Online]. Available: https://faq.whatsapp.com
2. Wes McKinney, *Python for Data Analysis*, O'Reilly Media, 2017.
3. Jake VanderPlas, *Python Data Science Handbook*, O'Reilly Media, 2016.
4. Matplotlib and Seaborn Documentation: https://matplotlib.org, https://seaborn.pydata.org