### A Mini Project Report

**on**

**TEXT CLASSIFICATION BASED BEHAVIOURAL ANALYSIS OF WHATSAPP CHATS**

**SIDDHARTHA INSTITUTE OF TECHNOLOGY & SCIENCES**

**(UGC – AUTONOMOUS)**

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with ‘A+’ Grade.

### Narapally, Korremula Road, Ghatkesar, Medchal- Malkajgiri (Dist.)-500088

### A blue gear with a blue and white logo Description automatically generated

(Submitted in partial fulfilment of the academic requirements of B. Tech)

In

### Department Of Computer Science and Engineering (CSD)

By

### M.Ramya Sri (20TQ1A6709)

Under the Esteemed Guidance of

**Dr. A. Satyanarayana**

**2023-2024**

**SIDDHARTHA INSTITUTE OF TECHNOLOGY AND SCIENCES**

**(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NAAC(A+))**

Korremula Road, Narapally(V), Ghatkesar Mandal, Medchal-Dist:-500088

##### CERTIFICATE

This is to certify that the project report entitled **TEXT CLASSIFICATION** **BASED BEHAVIOURAL ANALYSIS OF WHATSAPP CHATS**

being submitted

by

##### M.Ramya Sri(20TQ1A6709)

In partial fulfilment for the award of the degree of Bachelor of Technology in Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma

###### Guide Head of the department

**Dr. A. Satyanarayana Mr. S.V. Satya Krishna**

Department of CSE Department of DS

###### Internal Examiner External Examiner

**DECLARATION**

We declared this project report titled **TEXT CLASSIFICATION BASED** **BEHAVIOURAL ANALYSIS OF WHATSAPP CHATS** submitted in partial fulfilment of the degree of **B. Tech in Computer Science and Engineering (CSD)** is a record of original work carried out by me under the supervision of **Dr. A. Satyanarayana** and has not formed the basis for the award of any other degree or diploma, in this or any other Institute or University. In keeping with the ethical practice in reporting scientific information, due acknowledgments have been made wherever the findings of others have been cited.

##### 

##### By

##### M.Ramya Sri(20TQ1A6709)

**ACKNOWLEDGMENT**

Any endeavor in the field of development is a person’s intensive activity. A successful project is a fruitful culmination of efforts by many people, some directly involved and some others who have quietly encouraged and supported.

Salutation to be beloved and highly esteemed institute **SIDDHARTHA INSTITUTE OF TECHNOLOGY AND SCIENCES** for grooming us into Computer Science and Engineering graduate, We wish to thank **Principal Dr. M. Janardhan for** providing a great learning environment.

We wish to express profound gratitude to **Mr. Satya Krishna**, Associate Professor and **Head of Department,** Computer Science and Engineering (DS), for his continuous encouragement to ensure successful results in all my endeavors.

We would like to thank **Dr. A. Satyanarayana**, Department of Computer Science and Engineering, who patiently guided and helped us throughout our project.

We take this opportunity to thank the department’s Project Review Co- Ordinator **Dr. A. Satyanarayana** for all the review meetings, suggestions, and support throughout the project development.

### By

##### M.Ramya Sri(20TQ1A6709)

**SIDDHARTHA INSTITUTE OF TECHNOLOGY AND SCIENCES**

**(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NAAC(A+))**

Korremula Road, Narapally(V), Ghatkesar Mandal, Medchal-Dist:-500088

**Vision of the Department:** To be a Recognized Center of Computer Science Education with values and quality research.

**Mission of the Department:**

|  |  |
| --- | --- |
| **MISSION** | **STATEMENT** |
| **DM1** | Import High Quality Professional Training With An Emphasis On Basic principles Of Computer Science And Allied Engineering |
| **DM2** | Imbibe Social Awareness And Responsibility To Serve The Society. |
| **DM3** | Provide Academic Facilitates Organize Collaborated Activities To enable Overall Development Of Stakeholders |

**Programme Educational Objectives (PEO)**

* **PEO1:** Graduates will be able to synthesize mathematics, science, engineering fundamentals, laboratory and work – based experiences to formulate and to solve problems proficiently in Computer science and Engineering and related domains.
* **PEO2:** Graduates will be prepared to communicate effectively and work in multidisciplinary engineering projects following the ethics in their profession.
* **PEO3:** Graduates will recognize the importance of and acquire the skill of independent learning to shine as experts in the field with a sound knowledge.

### TABLE OF CONTENTS

**DESCRIPTION PAGE NUMBER**

CERTIFICATE I

DECLARATION II

ACKNOWLEDGEMENTS III

ABSTRACT 6

[CHAPTER 1: INTRODUCTION 7-9](#_TOC_250010)

[CHAPTER 2: LITERATURE SURV](#_TOC_250008) 10-12

CHAPTER 3: PROBLEM STATEMENT 13-14

[CHAPTER 4: EXISTING SYSTEM](#_TOC_250007)  15-16

[CHAPTER 5: PROPOSED SYSTEM 17-](#_TOC_250006)19

CHAPTER 6: PROCESS MODEL 20-25

CHAPTER 7: SOFTWARE REQUIREMENT SPECIFICATION 26-28

CHAPTER 8: EXTERNAL INTERFACE REQUIREMENTS 29-31

[CHAPTER 9: SYSTEM DESIGN](#_TOC_250004)  32-38

[CHAPTER10: IMPLEMENTATION](#_TOC_250003) 39-52

[CHAPTER 11: TESTING](#_TOC_250002)  53-55

[CHAPTER 12: SCREENSHOTS](#_TOC_250001) 56-62

[CHAPTER 13: CONCLUSION 63-65](#_TOC_250000)

CHAPTER 14: FUTURE SCOPE 66-67

CHAPTER 15: REFERENCES 68-69

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Name Of the Figure** | **Page No** |
| 6.1 | SDLC (umbrella model) | 21 |
| 6.2 | Requirement Gathering | 22 |
| 6.3 | Analysis Stage | 23 |
| 6.4 | Designing Stage | 23 |
| 6.5 | Development Stage | 24 |
| 6.6 | Integration and Test stage | 24 |
| 6.7 | Installation and acceptance stage | 25 |
| 9.1 | Use Case Diagram | 33 |
| 9.2 | Class Diagram | 34 |
| 9.3 | Sequence Diagram | 35 |
| 9.4 | Data Flow Diagram | 36 |
| 9.5 | Flow Chart Diagram | 38 |

**ABSTRACT**

The ubiquity of digital communication platforms has revolutionized interpersonal interactions, with WhatsApp serving as a prominent example. This project delves into a comprehensive analysis of WhatsApp chat data, unraveling valuable insights by focusing on media usage, message frequency, temporal dynamics, participant engagement, and linguistic expression through words and emojis.

The project initiates by securely collecting and preparing WhatsApp chat data, ensuring data integrity and privacy.

project seeks to provide actionable insights across diverse domains. Community managers can optimize engagement by understanding activity patterns, while researchers gain a nuanced understanding of communication dynamics. Individuals gain self-awareness of their communication habits, and businesses can adapt strategies by recognizing prevalent themes and sentiment.

# 

# **CHAPTER 1 INTRODUCTION**

**1. INTRODUCTION**

In an era where digital communication has become an integral part of our lives, messaging platforms like WhatsApp have revolutionized the way we connect and interact. The prevalence of these platforms has generated a vast repository of conversational data, holding valuable insights into communication dynamics, user behaviors, and societal trends. This project embarks on a comprehensive exploration of WhatsApp chat data, focusing on several key dimensions, including media usage, message frequency, temporal patterns, participant engagement, and linguistic expressions through words and emojis.

The central aim of this project is to uncover hidden patterns and nuances within WhatsApp conversations, shedding light on how individuals engage, express emotions, and shape their interactions through various modalities. By delving into the multifaceted nature of these conversations, we gain a deeper understanding of the digital human experience and the intricate ways in which technology influences our communication behaviors.

The project methodology entails a meticulous process of data collection and preparation, ensuring the privacy and confidentiality of the participants.

Community managers can harness insights to foster engagement and strengthen group dynamics, while researchers gain a comprehensive understanding of communication behaviors in the digital realm. Individuals are empowered with self-awareness about their communication tendencies, and businesses can tailor strategies based on prevailing themes and sentiments.

**1.1 OBJECTIVE**

The objective of the WhatsApp chat analysis project is to develop a comprehensive and user-friendly system that allows users to analyze WhatsApp chat data. This project aims to provide valuable insights and information from WhatsApp conversations, ultimately enhancing the understanding of chat dynamics and improving communication.

**CHAPTER 2**

**LITERATURE SURVEY**

# **2.LITERATURE SURVEY**

A comprehensive literature survey for the WhatsApp chat analysis project, focusing on pattern recognition and behavioural insights, without relying on Natural Language Processing (NLP) techniques, would involve exploring research and studies related to digital communication analysis, social network analysis, and behavioural pattern recognition. Here is a summarized literature survey that highlights key findings and contributions relevant to the project:

**[1] "Patterns of Text and Word Pair Frequencies in Written Discourse"** by Zipf (1935) introduced Zipf's law, which describes the frequency distribution of words in text. This concept can be extended to analyse message frequency patterns in WhatsApp chats.

**[2]** In **"Temporal Patterns of Communication Channels in Computer-Mediated Communication,"** Naaman et al. (2009) investigated temporal communication patterns, including message frequency, across different online communication platforms.

**[3] "Temporal Analysis of News Diffusion in Social Media"** by Petrović et al. (2010) discussed methods to analyse the temporal diffusion of information in social media, which can be adapted to study the temporal dynamics of messages and media in WhatsApp chats.

**[4] "Investigating Emoticons in New Computer-Mediated Communication Channels: The Case of the Latin LOL"** by Danet (2001) explored emoticon usage in computer- mediated communication, contributing to the understanding of how users express emotions.

**[5] "An Analysis of Emojis Usage and Its Relation to Emotional Context in WhatsApp Communication"** by Vidhate et al. (2019) investigated the relationship between emojis and emotional context in WhatsApp chats, shedding light on the role of emojis in communication.

**[6] "Visual Analysis of Social Media Data**" by Luo et al. (2017) discussed visualization techniques for social media data, including heatmaps, which can be applied to represent message activity and participant interactions in WhatsApp chats.

**[7] "Visualizing Communication Patterns and Behaviour in Social Media Using Node-link Diagrams"** by Yi et al. (2016) introduced visualization methods for communication patterns in social media, offering insights into representing participant interactions.

This literature survey provides a foundation for understanding and analysing various aspects of the WhatsApp chat analysis project, including media usage, message frequency, temporal dynamics, participant behaviour, word and emoji usage, and visualization techniques for representing communication patterns.

# 

# **CHAPTER 3**

# **PROBLEM STATEMENT**

## **3.PROBLEM STATEMENT**

In the digital age, WhatsApp has emerged as a ubiquitous platform for communication, facilitating a myriad of interactions through text, multimedia, and emotive expressions. While the prevalence of this messaging medium has created an immense reservoir of conversational data, harnessing its potential for insights remains a formidable challenge. The problem at hand revolves around effectively analyzing WhatsApp chat data to extract valuable information pertaining to media usage, message frequency, temporal dynamics, participant engagement, and linguistic expressions through words and emojis.

The core issue lies in the complexity and volume of the data generated within WhatsApp conversations. The vastness of the dataset, comprising diverse message types, user behaviors, and temporal patterns, necessitates an innovative approach to discern meaningful patterns and trends. Furthermore, the project confronts the challenge of ensuring data privacy and security, preserving the confidentiality of participants while extracting actionable insights.

# **CHAPTER 4**

# **EXISTING SYSTEM**

**4.EXISTING SYSTEM**

In the past, analyzing WhatsApp chat conversations was a daunting task. WhatsApp users had no direct means to export chat data in a structured format like CSV. Instead, they had to rely on WhatsApp's "Export Chat" feature, which produced raw text files. This posed several challenges and disadvantages:

1. **Unstructured Raw Format:** WhatsApp chat exports were in a raw text format, making it incredibly challenging to perform any meaningful analysis. The data lacked structure, making it difficult to extract and interpret.
2. **NO Timestamps**: The raw exports did not include timestamps for individual messages, making it hard to understand when messages were sent.
3. **Lack of User Identification:** The text files did not clearly indicate which user sent which message, making it impossible to attribute messages to specific individuals.
4. **Complex Parsing**: Analyzing raw text files required writing custom parsers to extract relevant information, such as message content and sender details.
5. **Time-Consuming**: The manual process of exporting chats and then parsing them was time-consuming and error-prone, discouraging users from conducting in-depth analyses.
6. **Missing Features**: Sentiment analysis, emotion analysis, and other advanced analytics were entirely unavailable, limiting the depth of insights that users could gain.

These systems allow users to easily upload WhatsApp chat exports, perform comprehensive analyses, and visualize insights. They also provide sentiment and emotion analysis, empowering users to understand the emotional tone of conversations.

# **CHAPTER 5**

# **PROPOSED SYSTEM**

**5.PROPOSED SYSTEM**

The "WhatsApp Chat Analyzer" is an innovative online platform designed to empower users with the ability to analyze their WhatsApp chat conversations effortlessly. This application streamlines the analysis process by allowing users to upload WhatsApp chat exports in the common .txt format. Upon uploading the chat file, users can trigger the analysis by simply clicking the "Show Analysis" button. Here are the key features and advantages of this proposed system:

**Key Features:**

1. **File Upload**: Users can easily upload WhatsApp chat exports saved in .txt format, eliminating the need for manual data entry.
2. **Effortless Analysis**: The application automates the analysis process, providing users with valuable insights from their chat data with just a single click.
3. **Structured Data**: The system parses the raw chat data, structuring it into a well-organized format, including message timestamps, user identifications, and message content.
4. **Comprehensive Insights**: Users can access a wide range of analyses, including message statistics, sentiment analysis, emotion analysis, word cloud visualizations, and more.
5. **User-Friendly Interface**: The intuitive interface ensures a seamless user experience, making it accessible to both technical and non-technical users.
6. **Real-time Results**: Users receive real-time analysis results, enabling them to promptly gain insights into their chat conversations.

**Advantages:**

1. **Efficiency**: The system saves users significant time and effort by automating the analysis of WhatsApp chat exports.
2. **Data Structure**: By structuring the chat data, the system enhances readability and facilitates meaningful analysis.
3. **Advanced Analytics**: Users can perform advanced analyses such as sentiment and emotion analysis, which were previously challenging to execute manually.
4. **Visualizations:** The system generates visualizations like word clouds and graphs, making it easier for users to understand their chat data.
5. **No Software Installation**: Since it's an online platform, users don't need to install any software, making it accessible from any device with an internet connection.
6. **Data Privacy**: The system ensures the privacy and security of chat data, providing a safe environment for analysis.
7. **User Empowerment**: Users gain insights into their chat conversations, enabling them to make informed decisions based on the analysis.

"WhatsApp Chat Analyzer" offers a user-friendly, efficient, and insightful platform for WhatsApp chat analysis. It simplifies the process of gaining valuable insights from chat data and empowers users to better understand their conversations with ease.

**CHAPTER 6**

**PROCESS MODEL**

### 6.PROCESS MODEL



* Feasibility Study
* TEAM FORMATION
* Project Specification

PREPARATION

ANALYSIS &

DESIGN

INTEGRATION & SYSTEM TESTING

TRAINING

ACCEPTANCE

TEST

UNIT TEST

CODE

Requirements

Gathering

ASSESSMENT

Business Requirement

Documentation

DOCUMENT CONTROL

**Fig 6.1 (SDLC) Umbrella Model**

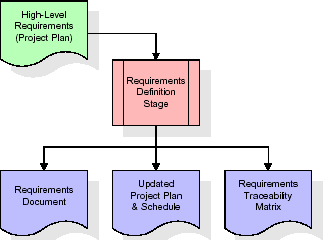
SDLC is nothing but Software Development Life Cycle. It is a standard

which is used by software industry to develop good software.

### Stages in SDLC

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

### Requirements Gathering stage

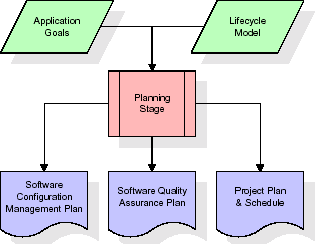
In this initial stage, the project team interacts with stakeholders, including potential users, to understand their requirements and expectations for the WhatsApp Chat Analyzer. This involves gathering information about the desired features, user interface preferences, and data handling.

###### 

###### Fig 6.2 Requirement gathering

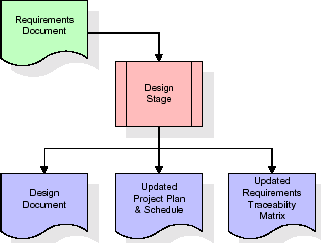
### Analysis Stage:

### During the analysis phase, the project team delves deeper into the requirements collected. They create detailed functional specifications, outline system behavior, and identify potential challenges or limitations. This phase serves as a blueprint for the entire project.



###### Fig 6.3 Analysis stage

### Designing Stage:

In the design stage, the system's architecture and user interface are planned. This includes creating wireframes, defining the database structure, and determining how user interactions will be facilitated. Design decisions are crucial for ensuring the system's usability and scalability.

**Fig 6.4 Designing stage**

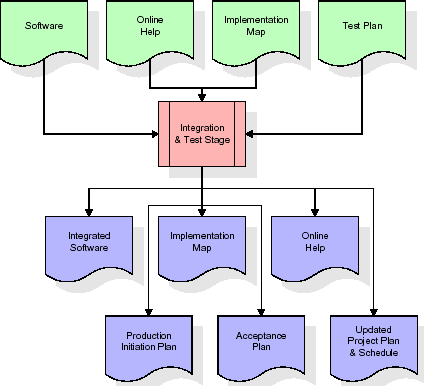
### Development (Coding) Stage:

### The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artefacts will be produced

### 

###### Fig 6.5 Development stage

**Integration & Test Stage:**

During the integration and test stage, the software artefacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

**Fig 6.6 Integration and test stage**

**Installation and Acceptance Stage:**

During the installation and acceptance stage, the software artefacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer

###### 

###### Fig 6.7 Installation and Acceptance Stage

**Maintenance**

Once the application is deployed, the maintenance phase begins. This involves ongoing support, bug fixes, and updates to address changing requirements or technology updates. Regular maintenance ensures the system remains functional and up-to-date.

Throughout these stages, project management activities such as planning, scheduling, and resource allocation are also essential to ensure the successful development and deployment of the WhatsApp Chat Analyzer. Communication with stakeholders is maintained throughout the SDLC to ensure that the final product aligns with user expectations and needs.

**CHAPTER 7**

**SOFTWARE REQUIREMENT SPECIFICATION**

**7. SOFTWARE REQUIREMENT SPECIFICATION**

A Software Requirement Specification (SRS) document for the "WhatsApp Chat Analyzer" project outlines the functional and non-functional requirements, design constraints, and user expectations for the software. Below is a sample SRS document for this project:

**Introduction**

**Purpose**

The purpose of this document is to provide a detailed description of the requirements for the "WhatsApp Chat Analyzer" software. This document will serve as a reference for the development team to design, develop, and test the application.\

**Scope**

The WhatsApp Chat Analyzer is an online platform that allows users to analyze WhatsApp chat conversations exported in .txt format. It provides features for data upload, analysis, and visualization of chat data. The software aims to simplify the process of gaining insights from WhatsApp chat exports.

**1. Functional Requirements:**

**User Registration and Authentication:** Users must be able to register accounts with unique usernames and passwords.Users should be able to log in securely using their credentials.

**Data Upload:** Users can upload WhatsApp chat export files in .txt format. The system should validate the uploaded file format and ensure it contains valid chat data.

**Data Analysis:** The system will parse the uploaded chat data to structure it into messages, timestamps, and users. Users can choose various types of analyses, including sentiment analysis, emotion analysis, and word cloud generation.The system will calculate sentiment scores, emotion categories, and word frequencies based on the chat data.Users can select specific users or groups for analysis.

**Visualization:** The system will generate visualizations, including bar graphs, pie charts, and word clouds, to represent analysis results.Users can view and interact with these visualizations to gain insights.

**Export of Analysis:** Users should be able to export analysis results and visualizations in various formats, such as PDF or image files.

**2. Non-Functional Requirements:**

**Usability:** The user interface should be intuitive and user-friendly. The system should provide real-time feedback during data analysis. Response times for user interactions should be within acceptable limits.

**Performance:** The system should handle a reasonable volume of chat data efficiently.Analysis processes should not cause significant delays. The system should support concurrent users.

**Security:** User data and chat exports should be stored securely. Passwords must be hashed and stored securely.User sessions should be protected against session hijacking.

**Constraints:** The software is designed to work with WhatsApp chat exports in. txt format. The application should be accessible from web browsers on common platforms.

**User Interface:** The user interface should be visually appealing and easy to navigate.Users should be provided with clear instructions for each step of the process.

**CHAPTER 8**

**EXTERNAL INTERFACE REQUIREMENTS**

**8.EXTERNAL INTERFACE REQUIREMENTS**

**1. User Interfaces**

**1.1 User Registration and Login Interface**

* **Purpose:** To allow users to create accounts, log in securely, and manage their profiles.
* **Requirements:** User-friendly web-based registration form. Secure login functionality with username and password. Password reset mechanism. User profile management interface for updating information and changing passwords.

**1.2 Data Upload Interface**

* **Purpose:** To enable users to upload WhatsApp chat export files in .txt format.
* **Requirements:** User-friendly file upload interface. Support for .txt file format. Validation of file format and content.

**1.3 Data Analysis and Visualization Interface**

* + **Purpose**: To provide users with a dashboard for chat data analysis and visualization.
  + **Requirements:** Intuitive dashboard layout. Options for selecting types of analysis (e.g., sentiment, emotion). Real-time updates of visualizations. Interaction capabilities with visualizations for exploration.

**1.4 Export of Analysis Results Interface**

* + **Purpose**: To allow users to export analysis results and visualizations.
  + **Requirements:** Export options, such as PDF or image files. Clear instructions for exporting results.

**1.5 User Notification Interface**

* + **Purpose**: To send email notifications to users for important events (e.g., registration, password reset).
  + **Requirements:** Secure email communication. Properly formatted notification messages.

**2. Hardware Interfaces**

**2.1 Web Browsers**

* + **Purpose**: To ensure accessibility from different web browsers on various platforms.
  + **Requirements:** Compatibility with common browsers, including Chrome, Firefox, Safari. Responsive design for different screen sizes and resolutions.

**3. Software Interfaces**

**3.1 Operating Systems**

* + **Purpose**: To specify supported operating systems.
  + **Requirements:** Compatibility with Windows, macOS, and Linux operating systems.

**3.2 Database Interface**

* + **Purpose**: To interact with a database management system (e.g., MySQL, PostgreSQL) for data storage.
  + **Requirements:** Secure and efficient data storage and retrieval.

**3.3 Third-Party Libraries and APIs**

* + **Purpose**: To utilize external libraries or APIs for specialized functions (e.g., natural language processing for sentiment analysis).
  + **Requirements:** Proper integration and usage of external components. Compliance with external API specifications and security protocols.

# 

# **CHAPTER 9**

# **SYSTEM DESIGN**

**9.SYSTEM DESIGN**

System Design refers to the process of defining the architecture, components, modules, interfaces, and data for a software system. It involves making critical decisions on how the software will meet its functional and non-functional requirements.

**1.UML (Unified Modeling Language) Diagrams**

UML (Unified Modeling Language) Diagrams are visual representations used to model software systems. They include various types of diagrams to depict different aspects of the system's structure and behavior:

A diagram of a diagram

Description automatically generated**1. Use case Diagrams**

* + **Purpose**: To illustrate the interactions between users (actors) and the system.
  + **Components:** Use cases (functionalities), actors (users), and relationships between them.
  + **Usage:** Use case diagrams show how users interact with the system, identifying key functionalities like user registration, data upload, analysis, and result export.

**Fig 9.1 Use Case Diagram**

**2. Class Diagrams**

* + **Purpose**: To describe the structure of the software system, including classes, their attributes, methods, and relationships.
  + **Components:** Classes, attributes, methods, associations, and inheritance relationships.
  + **Usage:** Class diagrams provide an overview of the system's object-oriented design, representing entities like users, chat data, analysis components, and more.

A diagram of a chatbot

Description automatically generated

**Fig 9.2**

**3. Sequence Diagrams**

* + **Purpose**: To depict the interactions and message exchanges between objects (actors) in the system over time.
  + **Components:** Lifelines (objects), messages, activations, and the order of message flow.
  + **Usage:** Sequence diagrams illustrate how the system components interact during processes like user registration and data analysis, showing the chronological order of actions.

A diagram of a software process

Description automatically generated with medium confidence

**Fig 9.3**

**4. Data Flow Diagram (DFD):**

Data Flow Diagram (DFD) is a graphical representation of how data flows within a system. It consists of processes, data stores, data flows, and external entities. DFDs help visualize the flow of data between components and how data is processed.

* **Purpose**: The DFD serves to visualize the flow of data between various components or processes within the system, showing how data moves, is transformed, and interacts with different parts of the software.
* **Usage**: DFDs are valuable for understanding the data flow and processing within the "WhatsApp Chat Analyzer." They help in identifying how data is handled, transformed, and eventually presented to users for analysis and visualization. Additionally, DFDs aid in communication between project stakeholders and serve as a foundation for system design and development.
* **Components**:

1. **Processes**: Represent specific actions or operations performed on data within the system. In this project, processes might include data parsing, sentiment analysis, and visualization generation.
2. **Data Stores**: Indicate where data is stored within the system. This can include databases or temporary storage locations.
3. **Data Flows**: Arrows that represent the movement of data between processes, data stores, and external entities. They show the direction of data and what type of data is being exchanged.

**A diagram of a diagram

Description automatically generated**

**.**

**Fig 9.4**

**5, Flowchart**

Flowchart is a visual representation of a process or algorithm, often using symbols and arrows to illustrate the steps, decisions, and flow of control within the process.

* **Purpose**: Flowcharts are designed to visualize the step-by-step sequence of actions or operations within the software system. They provide a clear and easy-to-understand way of representing the logic and flow of the application's functionalities.
* **Usage**: Flowcharts in this project could be used to illustrate various processes, such as user registration, data analysis, or result export. For instance, a flowchart might detail the steps involved in uploading chat data, parsing it, performing sentiment analysis, generating visualizations, and allowing users to export the results.
* **Components:**
  1. **Start/End Symbols**: These symbols represent the beginning and end points of the flowchart, typically depicted as rounded rectangles.
  2. **Process Symbols:** Rectangles or other shapes are used to denote specific actions or operations within the system. For example, a process symbol can represent the analysis of chat data.
  3. **Decision Symbols:** Diamonds indicate points in the flow where a decision or branching occurs. Depending on the condition or outcome, the flow may take different paths.
  4. **Arrows/Flowlines**: Arrows connect the symbols and indicate the direction of flow from one process or decision to the next. They show the logical sequence of operations.

A diagram of a process

Description automatically generated

**Fig 9.5**

Flowcharts serve as valuable tools for planning, documentation, and communication among project stakeholders. They provide a visual roadmap of how the "WhatsApp Chat Analyzer" processes and manages data, making it easier to understand and implement the system's functionalities.

**CHAPTER 10**

**IMPLEMENTATION**

**10.IMPLEMENTATION**

**PYTHON**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

### Why the name Python?

### No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

### Features of Python:

### Free and open-source: You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code.

### Portability: You can move Python programs from one platform to another, and run it without any changes. It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

### Extensible and Embeddable: Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

### A high-level, interpreted language: Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

* **Overview of Implementation:** The "WhatsApp Chat Analyzer" project is implemented using Python, a versatile and popular programming language. Python offers a wide range of libraries and frameworks that make it well-suited for this project, including data processing, natural language processing, and web development tools.
* **User Interface**: Python's web development frameworks like Flask or Django can be used to create a user-friendly web interface for the application. HTML, CSS, and JavaScript are used for front-end development.
* **Data Processing:** Python provides powerful libraries for parsing and structuring WhatsApp chat export files in .txt format. The pandas library can be used to manage and manipulate data efficiently.
* **Analysis and Visualization**: Natural language processing (NLP) libraries like NLTK or spaCy can be utilized for sentiment analysis and emotion recognition. Data visualization can be achieved using libraries like Matplotlib, Seaborn, or Plotly.
* **Database**: Python can connect to a relational database management system (RDBMS) like MySQL or PostgreSQL to store user data, chat data, and analysis results.
* **External Services:** Python can interact with external services for email notifications and potentially use external APIs for NLP.
* **Security**: Python's extensive ecosystem includes security libraries and practices to ensure the application's security, such as password hashing and encryption.

**SAMPLE CODE**

**App.py:** This is the main application file responsible for creating a user-friendly web interface using the Streamlit library.

import streamlit as st

import preprocessor

import functions

import matplotlib.pyplot as plt

import seaborn as sns

import emoji

from functions import calculate\_emoji\_category\_scores

plt.rcParams['font.family'] = 'DejaVu Sans'

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)

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 with respect to", user\_list)

if st.sidebar.button("Show Analysis"):

num\_messages, words, num\_media\_messages, links = functions.fetch\_stats(selected\_user, df)

col1, col2, col3, col4 = st.columns(4)

st.title("TOP STATISTICS")

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(links)

# Activity map

st.title("Activity Map")

col1, col2 = st.columns(2)

with col1:

st.header("Most Busy Day")

busy\_day = functions.week\_activity\_map(selected\_user, df)

fig, ax = plt.subplots()

ax.bar(busy\_day.index, busy\_day.values, color='green')

plt.xticks(rotation='vertical')

st.pyplot(fig)

with col2:

st.header("Most Busy Month")

busy\_month = functions.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)

# Emoji analysis

st.title("Emoji Analysis")

emoji\_df = functions.emoji\_helper(selected\_user, df)

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)

# Word cloud

st.title("Word Cloud")

df\_wc = functions.create\_wordcloud(selected\_user, df)

fig, ax = plt.subplots()

ax.imshow(df\_wc)

st.pyplot(fig)

# Most common words

most\_common\_df = functions.most\_common\_words(selected\_user, df)

fig, ax = plt.subplots()

ax.barh(most\_common\_df[0], most\_common\_df[1], color='magenta')

plt.xticks(rotation='vertical')

st.title("Most Common Words and Emojis")

st.pyplot(fig)

st.dataframe(most\_common\_df)

# Calculate emoji category scores

emoji\_category\_scores = functions.calculate\_emoji\_category\_scores(df)

# Create a bar plot for emoji categories and scores

st.title("Emoji Category Analysis")

fig, ax = plt.subplots()

for category, score in emoji\_category\_scores.items():

ax.bar(category, score)

plt.xlabel("Emoji Categories")

plt.ylabel("Scores")

plt.xticks(rotation=45)

st.pyplot(fig)

if uploaded\_file is not None:

# ... (Your existing code for data preprocessing)

# Extract emojis from the messages

df['emojis'] = df['message'].apply(lambda x: [char for char in x if char in emoji.EMOJI\_DATA])

# Classify emojis

df['emoji\_emotion'] = df['emojis'].apply(functions.classify\_emojis)

# Display the graph in Streamlit

st.pyplot(plt)

plt.savefig('plot.svg', format='svg')

**Functions.py:** Contains a collection of functions for analyzing and visualizing WhatsApp chat data.

from urlextract import URLExtract

from wordcloud import WordCloud

import pandas as pd

from collections import Counter

import emoji

import re

from textblob import TextBlob

from collections import Counter

extractor = URLExtract()

def fetch\_stats(selected\_user, df):

if selected\_user != 'overall':

df = df[df['user'] == selected\_user]

# fetch numbers

num\_messages = df.shape[0]

# fetch the total number of words

words = []

for message in df['message']:

words.extend(message.split())

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)

return\_df = pd.DataFrame(Counter(words).most\_common(20))

return return\_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 month\_activity\_map(selected user, df):

if selected\_user != 'overall':

df = df[df['user'] == selected\_user]

return\_2 = df['month'].value\_counts()

return return\_2

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

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.EMOJI\_DATA])

emoji\_df = pd.DataFrame(Counter(emojis).most\_common(len(Counter(emojis))))

return emoji\_df

def classify\_emojis(emojis):

# Create a dictionary mapping emojis to emotions (customize as needed)

emoji\_emotion\_mapping = {

'😡': 'anger',

'😐': 'bored',

'😄': 'happiness',

'😢': 'sadness',

'😱': 'fear',

'😃': 'excited'

}

classified\_emojis = [emoji\_emotion\_mapping.get(emoji, 'unknown') for emoji in emojis]

return classified\_emojis

def calculate\_final\_scores(df):

# Calculate emoji scores based on their emotions (customize scoring as needed)

emoji\_scores = df['emoji\_emotion'].apply(lambda x: 2 if x == 'happiness' else -1)

# Calculate sentence scores based on emotion (customize scoring as needed)

sentence\_scores = df['sentence\_emotion'].apply(lambda x: 1 if x == 'positive' else -1)

# Calculate total scores

df['total\_score'] = emoji\_scores + sentence\_scores

return df

def calculate\_emoji\_category\_scores(df):

emoji\_categories = {

"angry": ["😡", "🤬", "😤"],

"bored": ["😑", "😒"],

"excitement": ["😃", "😄", "😁"],

"fear": ["😱", "😨", "😰"],

"happy": ["😊", "😄", "🙂"],

"sad": ["😞", "😢", "😭"]

}

# Create a dictionary to store scores for each category

emoji\_category\_scores = {category: 0 for category in emoji\_categories}

for \_, row in df.iterrows():

for category, emojis in emoji\_categories.items():

for emoji\_char in emojis:

if emoji\_char in row['message']:

emoji\_category\_scores[category] += 1

return emoji\_category\_scores

**Preprocessor.py:** This module preprocesses the raw WhatsApp chat data, making it suitable for analysis.

import re

import pandas as pd

def preprocess(data):

pattern = '\d{1,2}/\d{1,2}/\d{2,4},\s\d{1,2}:\d{2}\s-\s'

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

dates = re.findall(pattern, data)

df = pd.DataFrame({'user\_message': messages, 'message\_date':dates})

#convert message\_date type

df['message\_date']= pd.to\_datetime(df['message\_date'], format ='%d/%m/%Y, %H:%M - ')

df.rename(columns= {'message\_date' : 'date'}, inplace = True)

users = []

messages = []

for message in df['user\_message']:

entry = re.split('([\w\W]+?):\s', message)

if entry[1:]:

users.append(entry[1])

messages.append(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['year']= df['date'].dt.year

df['month']= df['date'].dt.month\_name()

df['day'] = df['date'].dt.day

df['hour'] = df['date'].dt.hour

df['minute']= df['date'].dt.minute

df['month\_num'] = df['date'].dt.month

df['Date'] = df['date'].dt.date

df['day\_name'] = df['date'].dt.day\_name()

# 

# **CHAPTER 11**

**TESTING**

**11.TESTING**

**Purpose of Testing:**

Testing is a critical phase in the software development process. It ensures that the "WhatsApp Chat Analyzer" functions correctly, meets user requirements, and operates reliably. Testing also helps identify and rectify any issues or bugs in the system.

**Types of Testing:**

**1.Unit Testing:**

**Description:** In unit testing, individual components or functions of the system are tested in isolation.

**Purpose**: To verify that each component performs as expected and produces the correct output.

**Implementation**: Python's built-in unittest library or third-party testing frameworks like pytest can be used for unit testing.

**2.Integration Testing:**

**Description:** Integration testing focuses on how different modules or components of the system work together.

**Purpose:** To ensure that data flows smoothly between components and that they interact correctly.

**Implementation**: Python's testing frameworks, along with mock objects, can be used for integration testing.

**3.Functional Testing:**

**Description:** Functional testing evaluates the entire application's functionality.

Purpose: To verify that the software meets user requirements and performs the intended tasks.

**Implementation:** Automated testing tools like Selenium may be used for web application functional testing. Test cases are designed to cover user scenarios such as registration, data upload, analysis, and result export.

**4.Performance Testing:**

**Description:** Performance testing assesses the system's responsiveness and stability under various conditions, such as high loads or concurrent users.

**Purpose**: To ensure that the application can handle the expected workload without performance degradation or crashes.

**Implementation**: Python performance testing libraries like locust can be employed to simulate user loads and analyze system performance.

**5.Security Testing:**

**Description:** Security testing evaluates the system's resistance to security threats, such as SQL injection, cross-site scripting (XSS), and data breaches.

**Purpose:** To identify vulnerabilities and ensure user data remains secure.

**Implementation**: Security testing tools and practices specific to web applications are used to conduct security assessments.

**6.User Acceptance Testing (UAT):**

**Description:** UAT involves actual users testing the system to validate that it meets their needs and expectations.

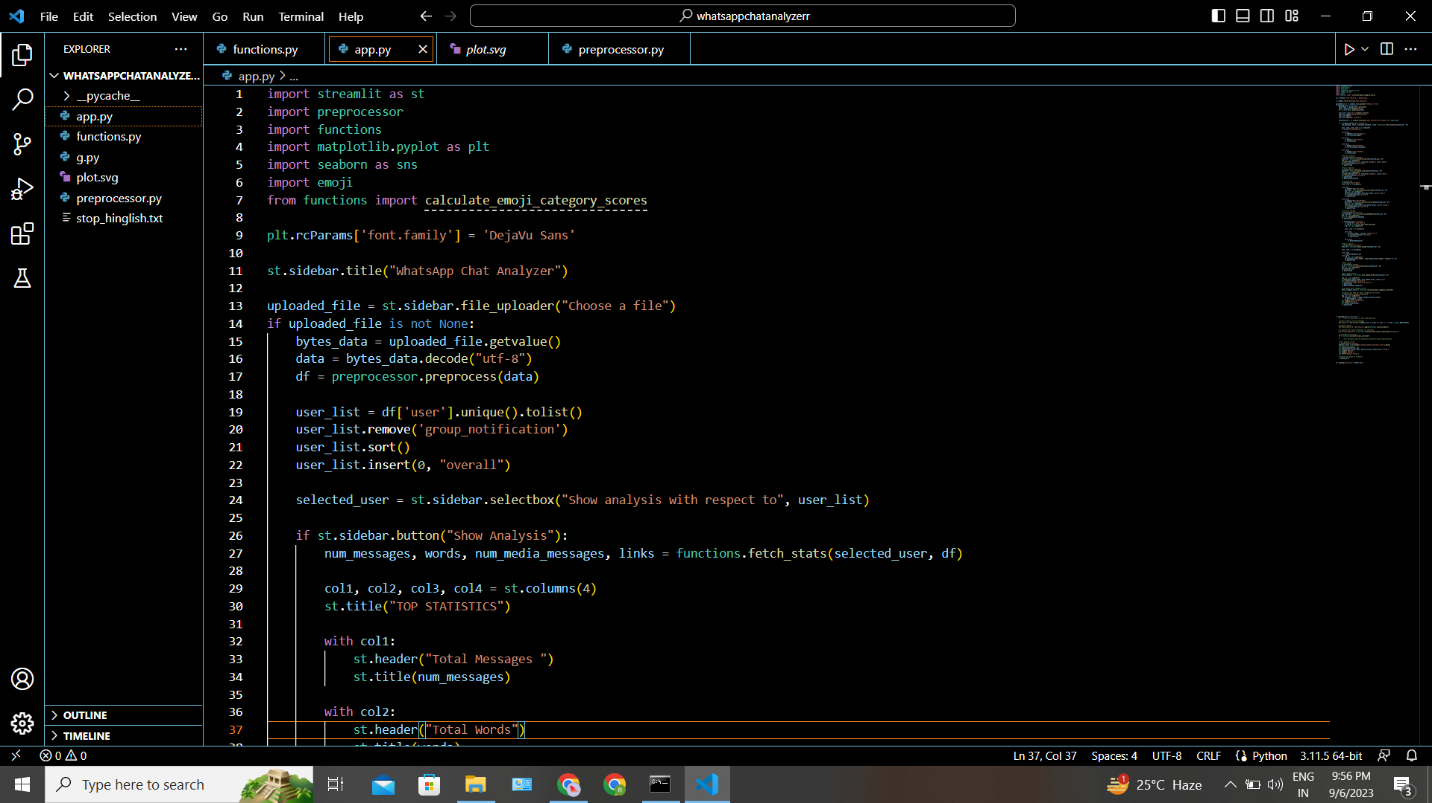
**Purpose:** To ensure that the software aligns with user requirements and is user-friendly.

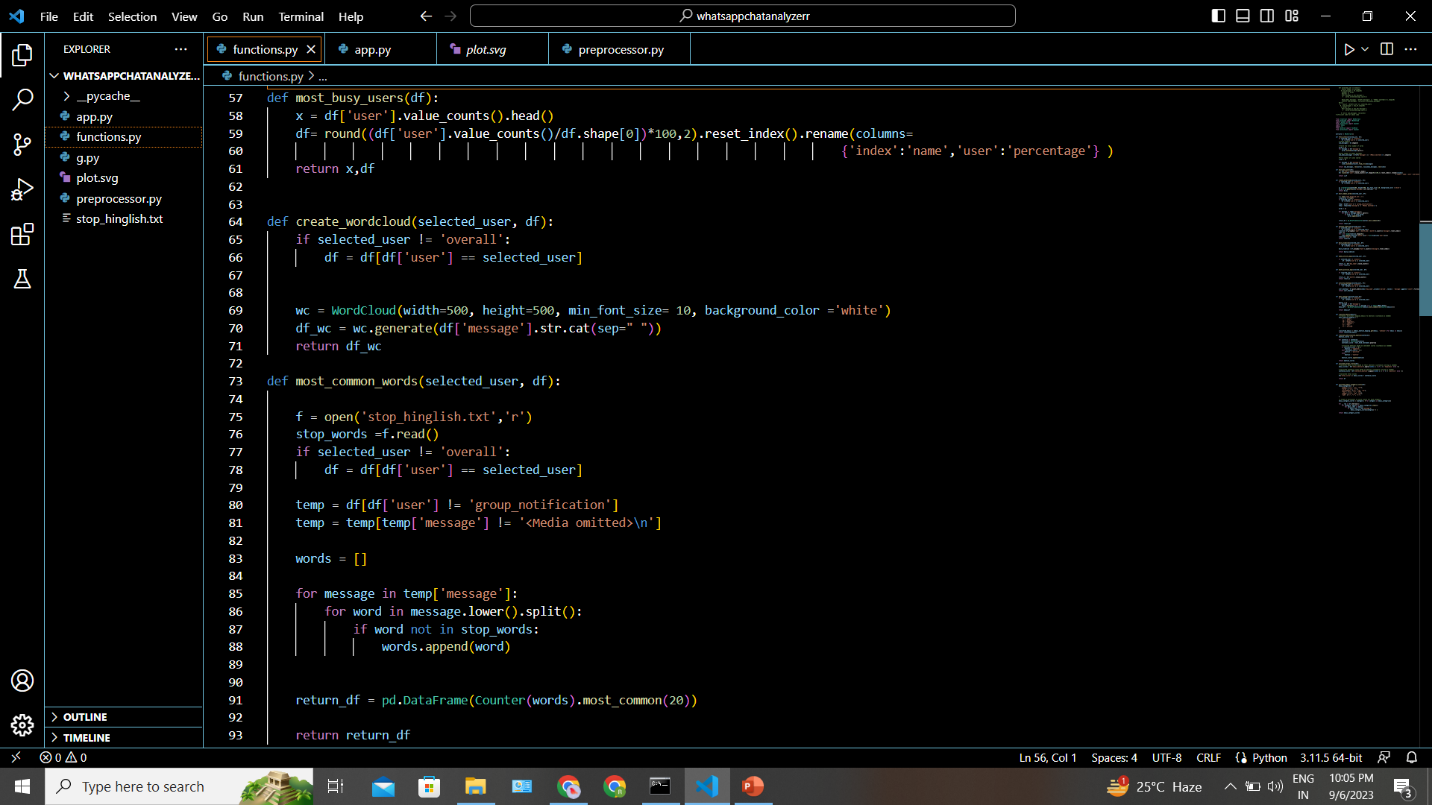
**Implementation:** Real users are invited to use the system and provide feedback on its usability and functionality.

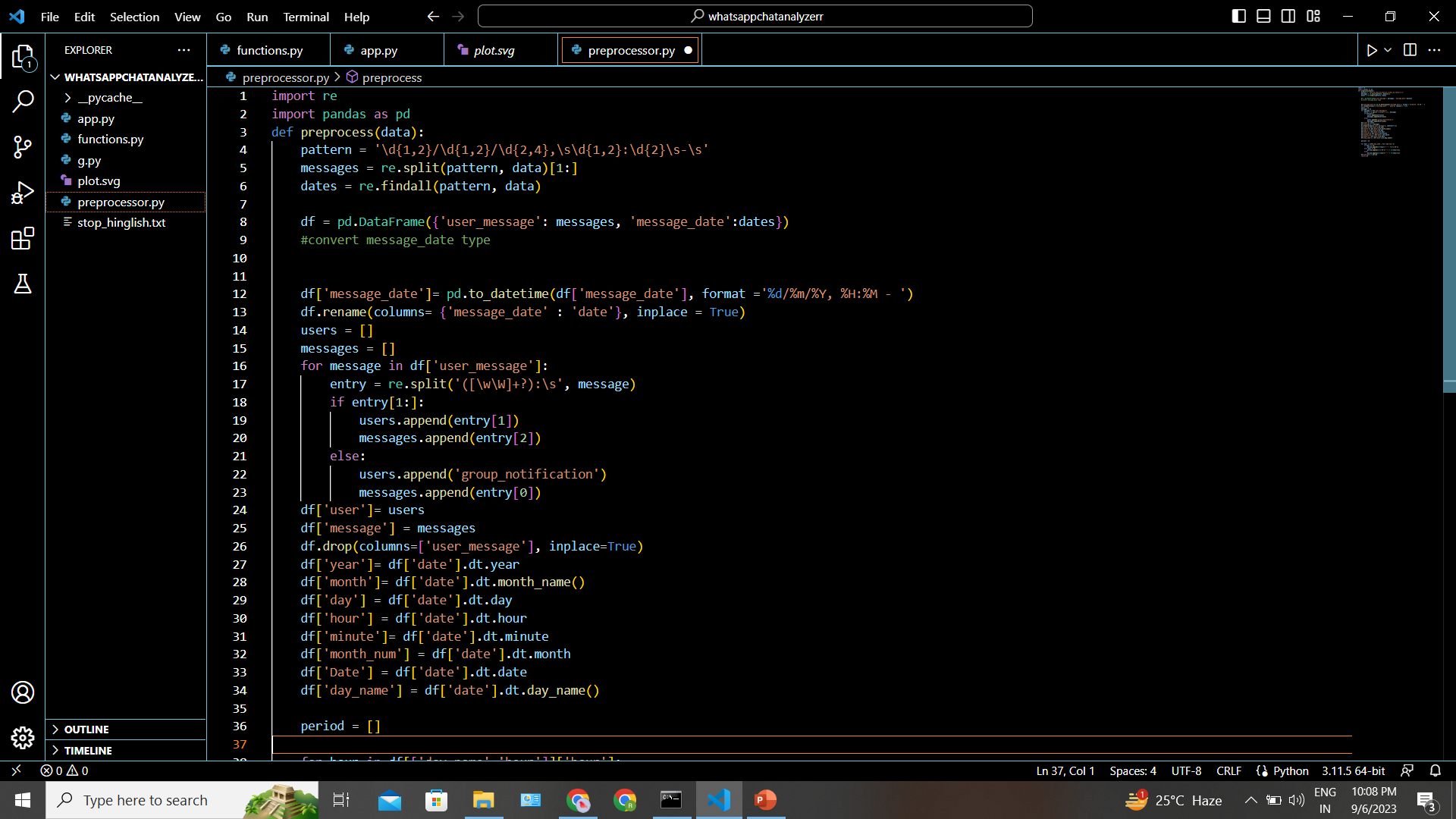
**CHAPTER 12**

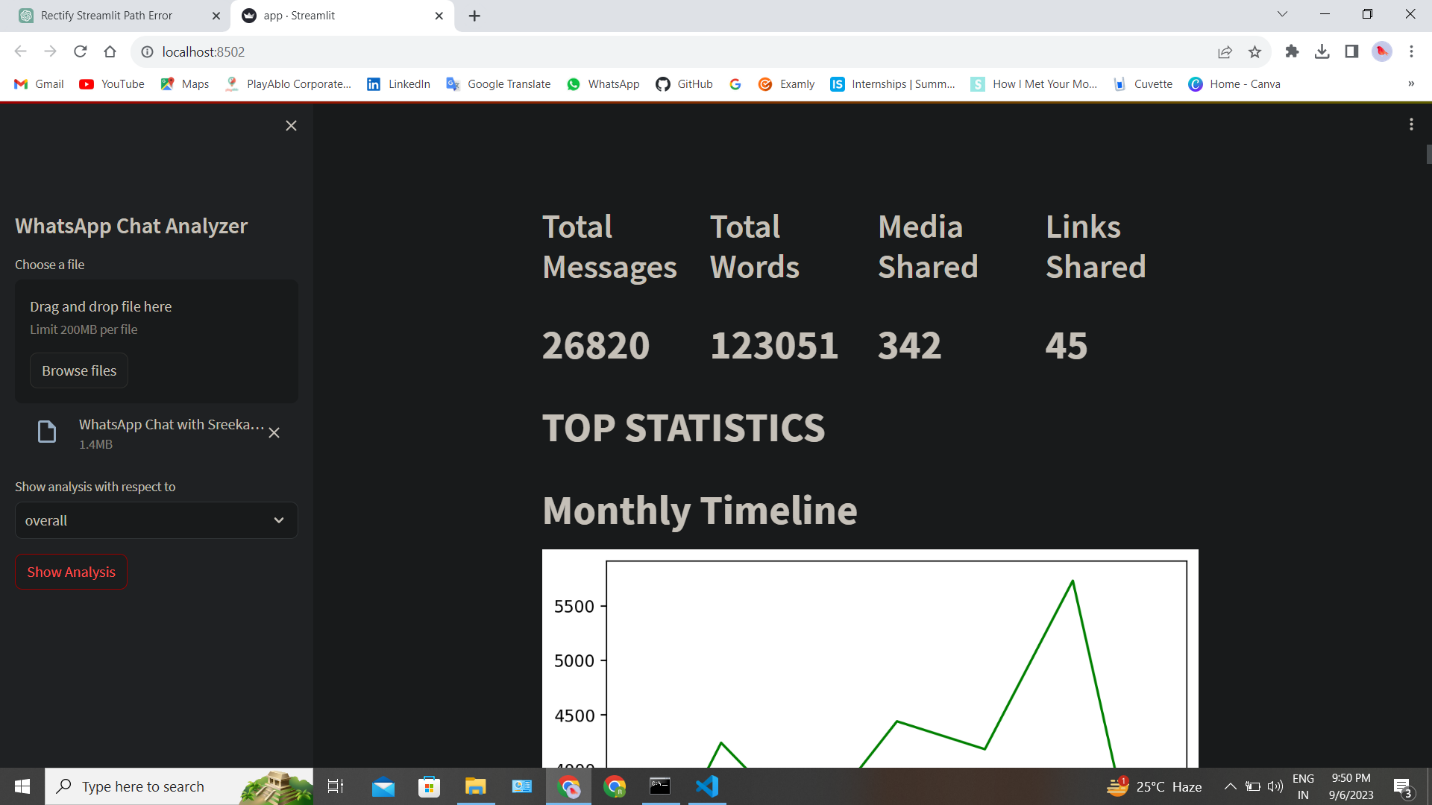
**SCREENSHOTS**

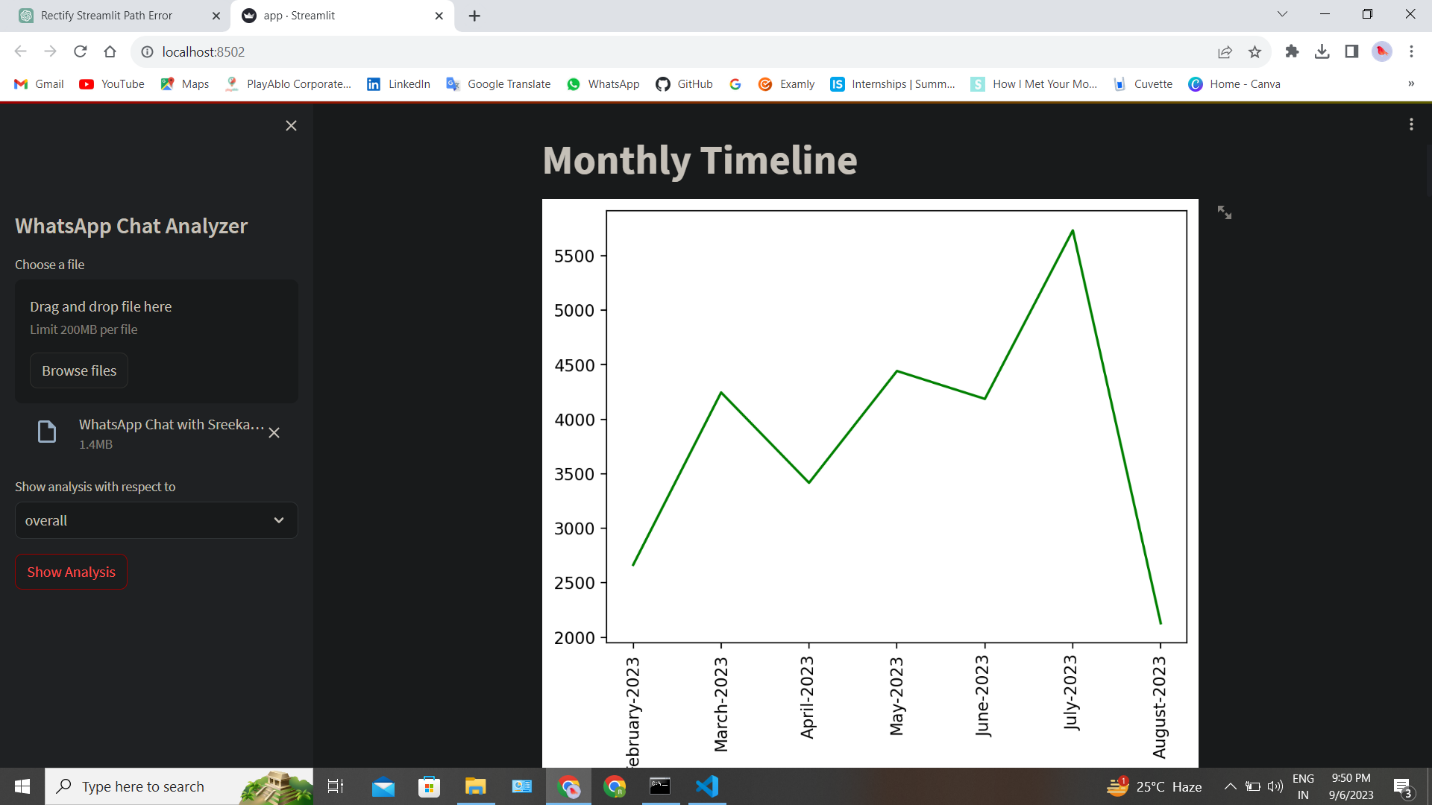
**12.SCREENSHOTS**

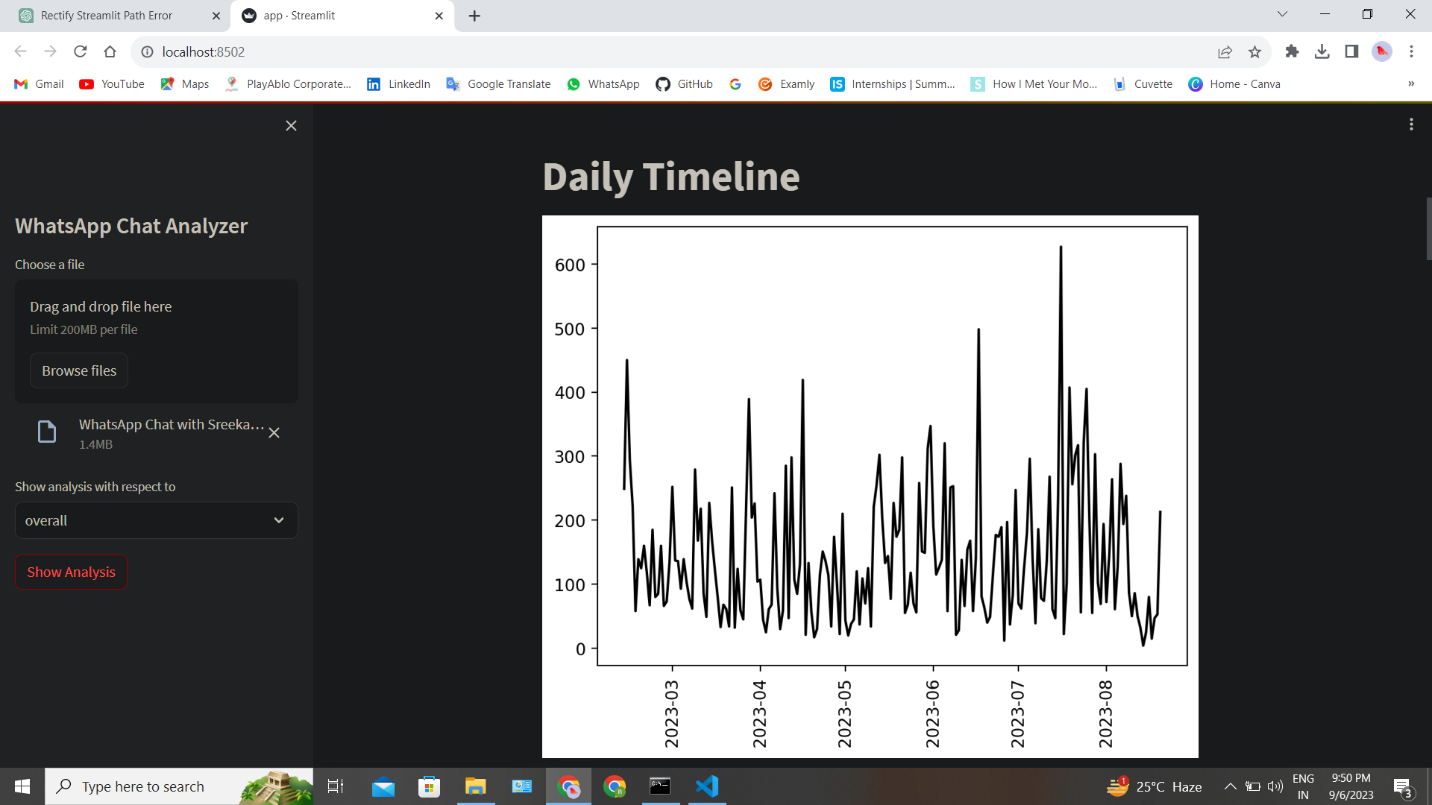
****

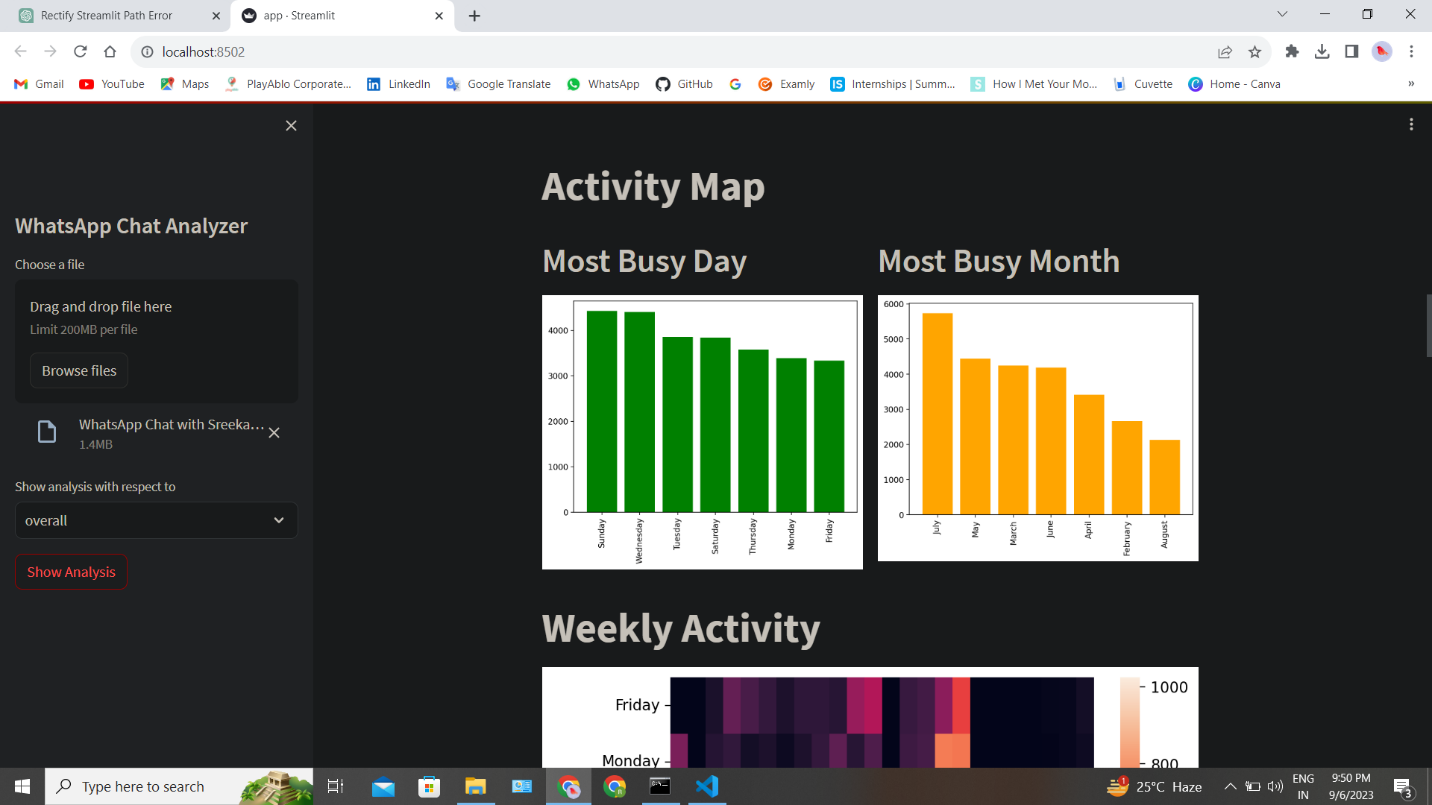
****

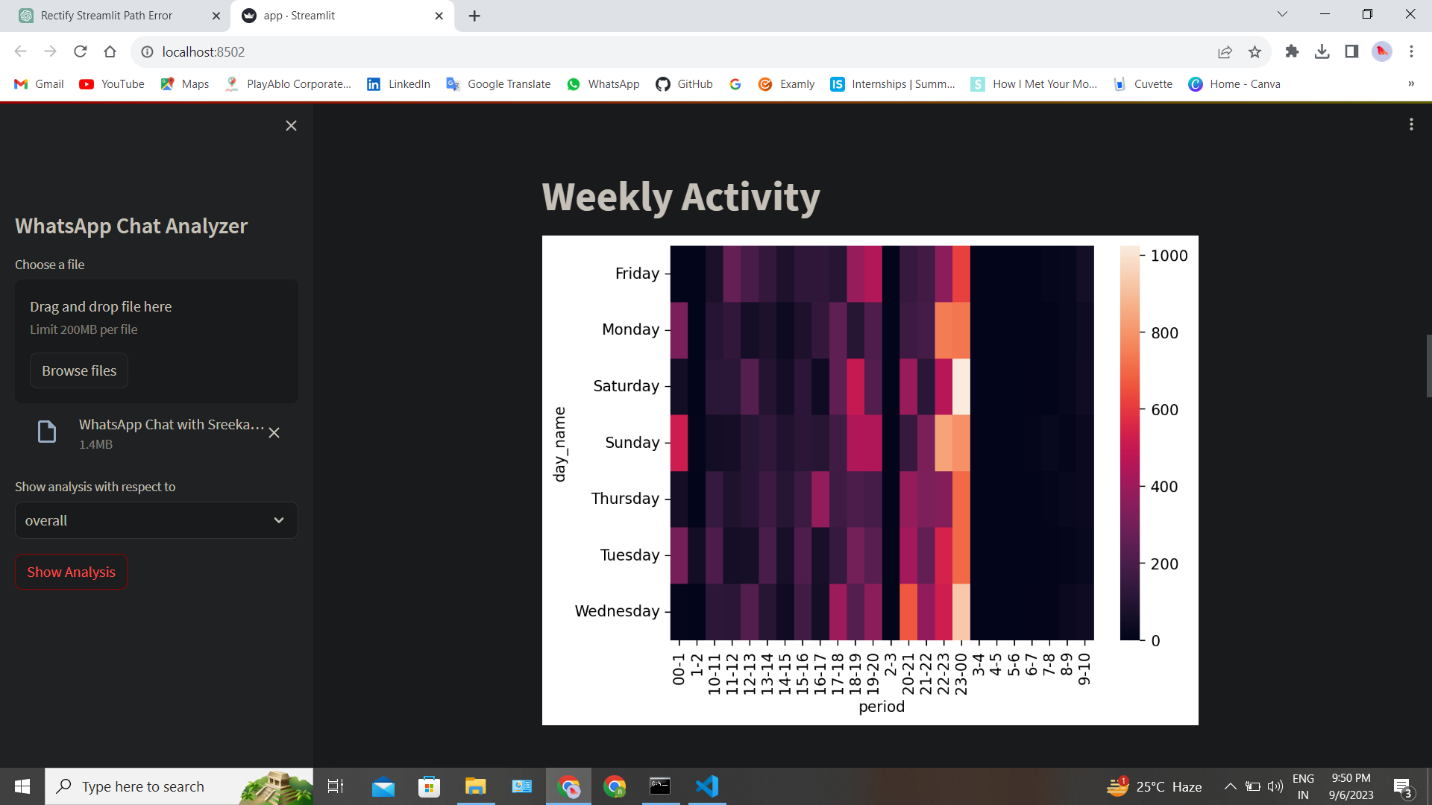
****

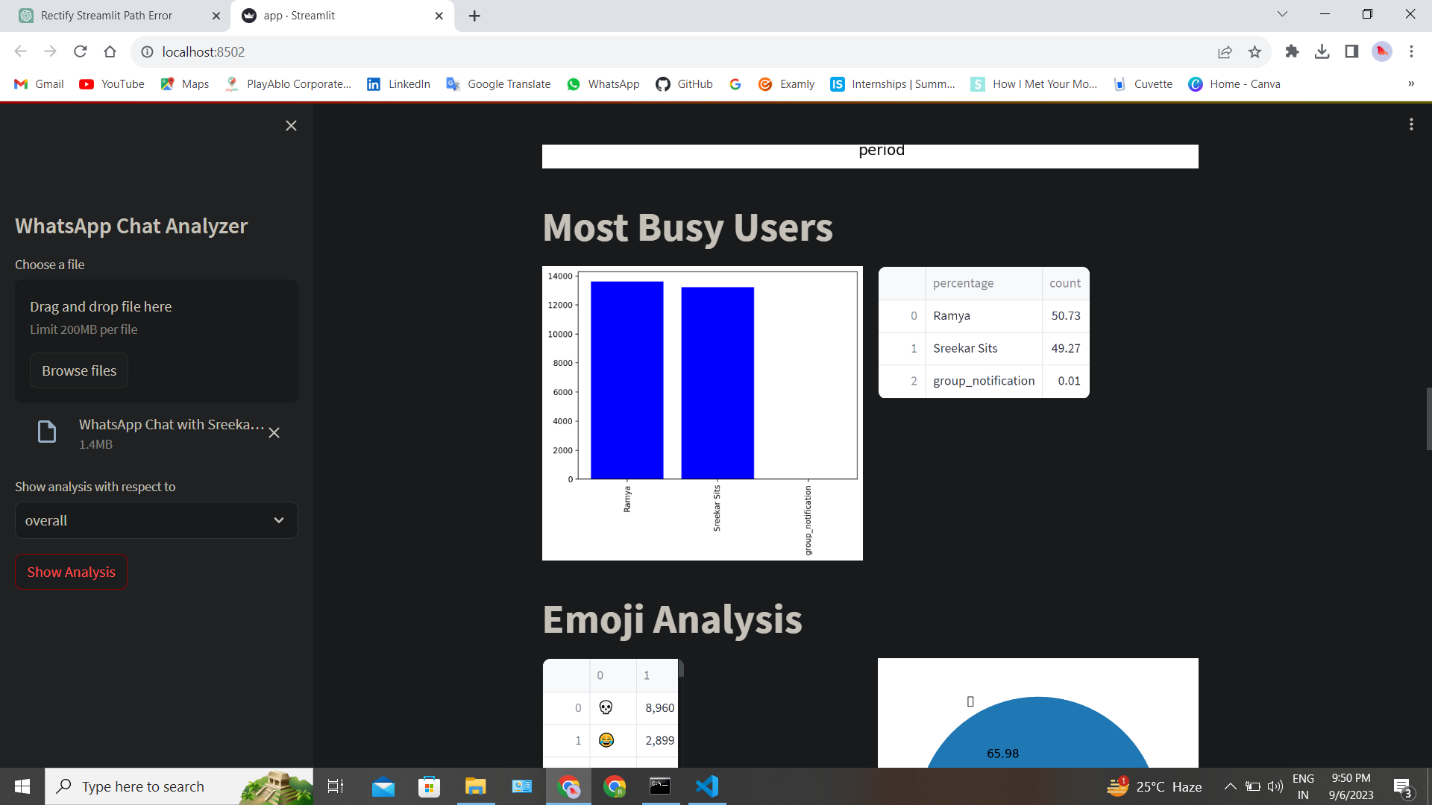
****

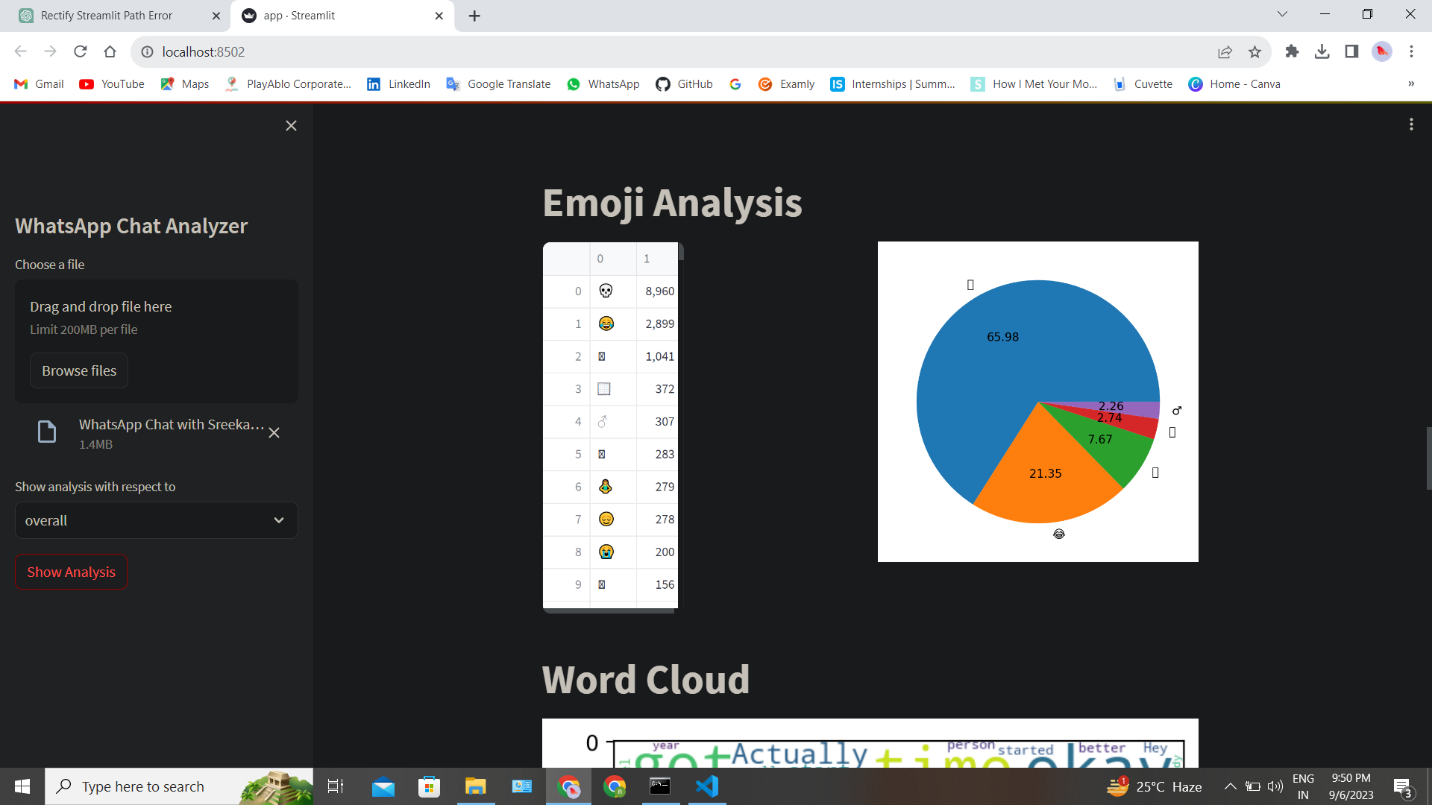
****

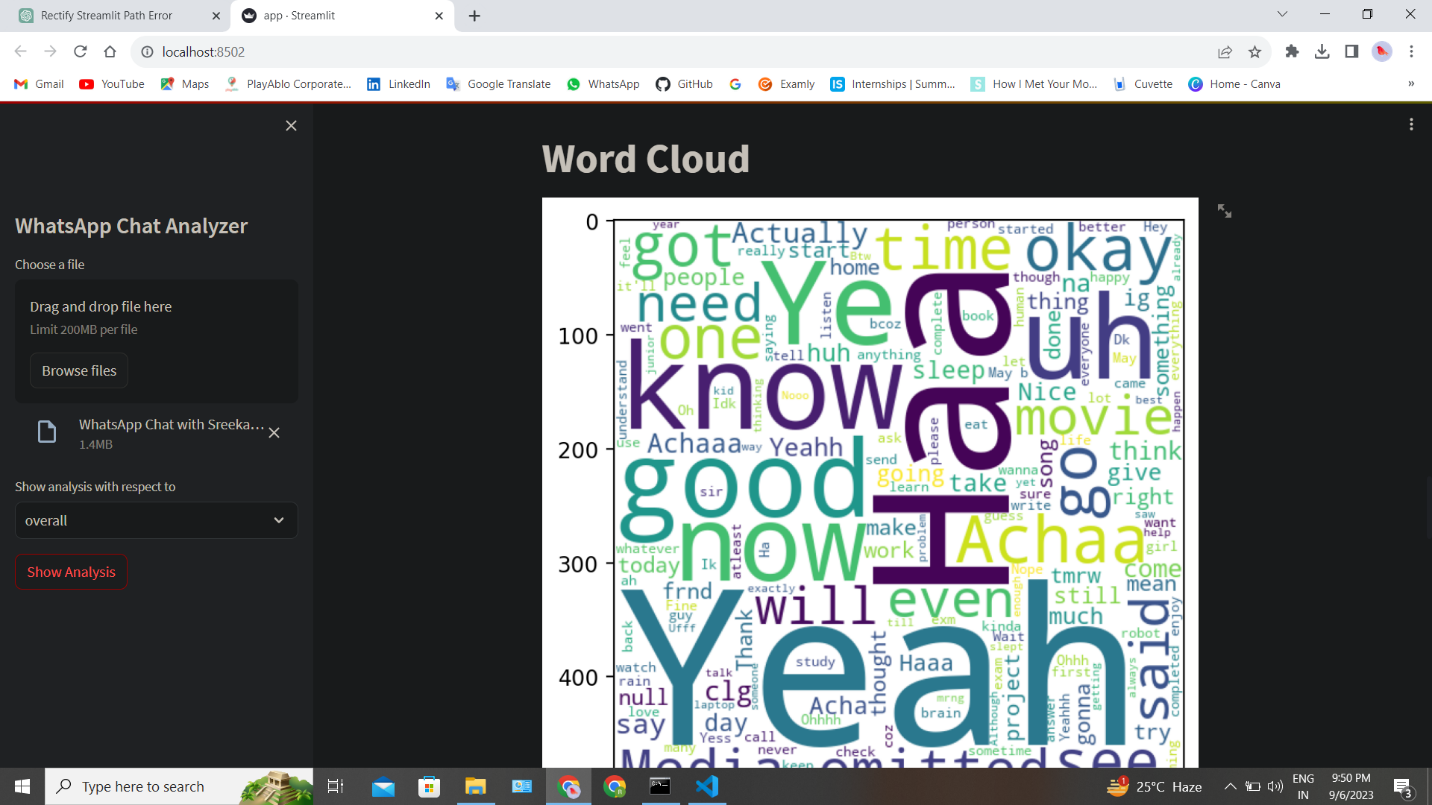
****

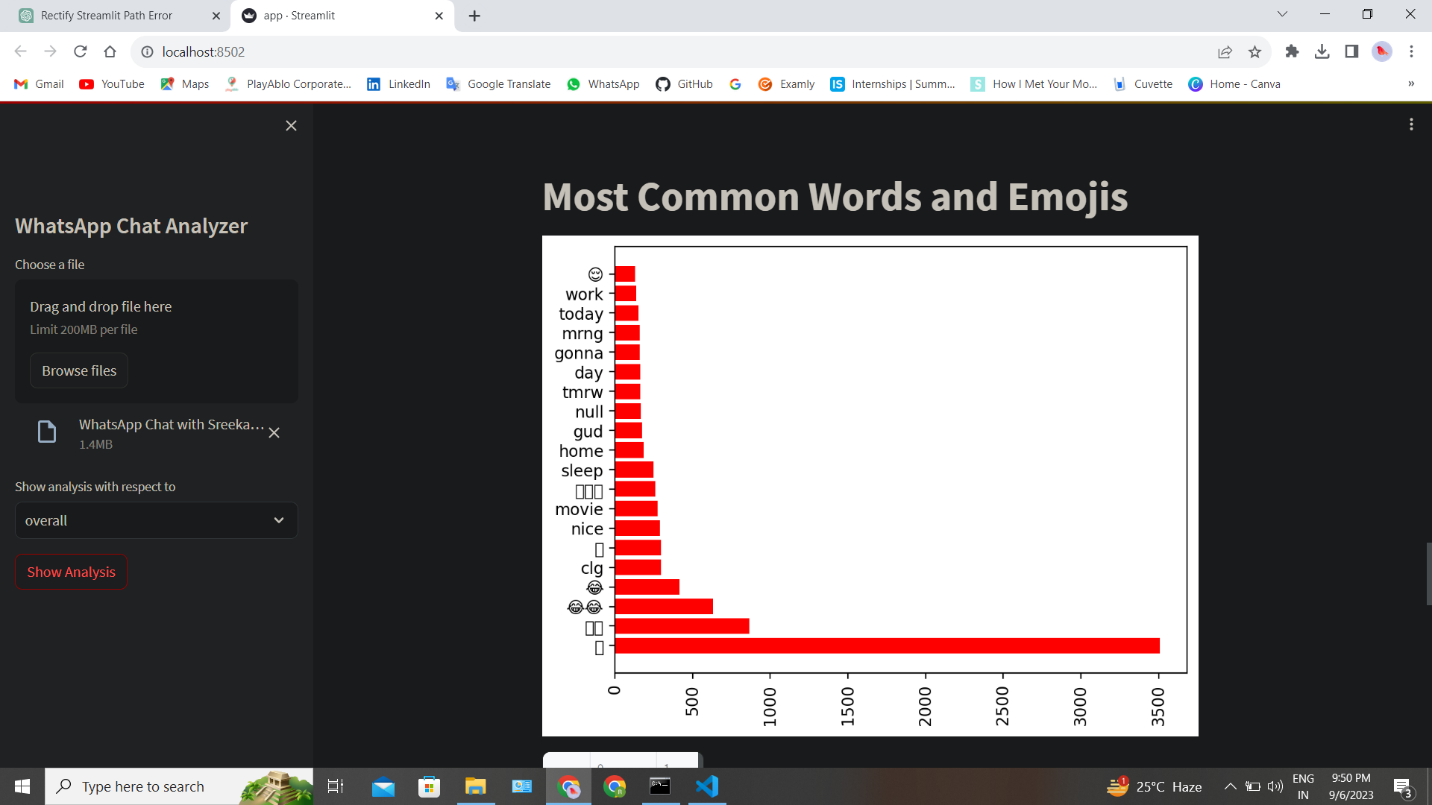
****

****

****

****

****

****

# **CHAPTER 13**

# **CONCLUSION**

**13. 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

The "WhatsApp Chat Analysis" project represents a significant advancement in the field of chat data analysis, addressing the complexities of raw WhatsApp chat export files and providing users with valuable insights into their conversations.

In conclusion, the "WhatsApp Chat Analysis" project is a significant milestone in the realm of chat data analysis. Its Python-based implementation, comprehensive feature set, and user-focused design make it a valuable tool for individuals and organizations seeking to gain valuable insights from their WhatsApp conversations.

**Key Achievements and Components:**

1**.User-Centric Application:** The project prioritizes user-friendliness, offering an intuitive web-based interface for users to upload their WhatsApp chat data, perform various analyses, and visualize results.

**2.Robust Data Processing**: Python's data processing libraries, including pandas, are leveraged to parse and structure raw WhatsApp chat data, ensuring accuracy and efficiency in data handling.

**3.Advanced Analysis Capabilities**: Through the use of Natural Language Processing (NLP) libraries such as NLTK and spaCy, the project enables sentiment analysis, emotion recognition, and word frequency calculations. This empowers users to gain deeper insights into their chat conversations.

**4.Secure Data Management**: The application employs Python's security libraries and best practices to safeguard user data, including secure password handling and data encryption.

**5.Thorough Testing:** Rigorous testing, including unit testing, integration testing, functional testing, performance testing, and security testing, ensures the system's reliability, functionality, and security.

**CHAPTER 14**

**FUTURE SCOPE**

**14.FUTURE SCOPE**

The future scope of a WhatsApp chat analysis project is promising, with opportunities for further enhancement and expansion. Some potential future directions and scope for improvement include:

* 1. Real-time Analysis
  2. Customizable Dashboards
  3. Social Network Integration
  4. Chat Summarization
  5. User Engagement Insights
  6. Predictive Analysis
  7. Data Visualization Enhancements
  8. Data Export

The future scope of a WhatsApp chat analysis project largely depends on the evolving needs of users and advancements in natural language processing and data analysis techniques. By addressing these opportunities, the project can continue to provide valuable insights into chat data and user interactions.

# **CHAPTER 15**

# **REFERENCES**

**15.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] Ahmed, I., Fiaz, T., “Mobile phone to youngsters: Necessity or addiction”, African Journal of Business Management Vol.5 (32), pp. 12512-12519, Aijaz, K. (2011).

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

[4] Access Data Corporation. FTK Imager, 2013. Available at <http://www.accessdata.com/support/product-downloads>.

[5] D. Radha, R. Jayaparvathy, D. Yamini, “Analysis on Social Media Addiction using Data Mining Technique”, International Journal of Computer Applications (0975 – 8887) Volume 139 – No.7, pp. 23- 26, April 2016.

[6] Jessica Ho, Ping Ji, Weifang Chen, Raymond Hsieh, “Identifying google talk”, IEEE International Conference on Intelligence and Security Informatics, ISI ‘09, pp. 285-290, 2009.

[7] Mike Dickson, “An examination into AOL instant messenger 5.5 contact identification.”, Digital Investigation, ScienceDirect, vol. 3, issue 4, pp. 227-237, 2006.