Summer Project On

Case Closed App

By

Sakshi Parkar (2021510046)
Devika Patil (2021510047)

Under the guidance of

Internal Supervisor

Dr. Aarti M. Karande

Department of Master Of Computer Application
Sardar Patel Institute of Technology
Autonomous Institute Affiliated to Mumbai University

2022-23

CERTIFICATE OF APPROVAL

This is to certify that the following students

Sakshi Parkar (2021510046)
Devika Patil (2021510047)

Have satisfactorily carried out work on the project entitled

"Case Closed App"

Towards the fulfilment of project, as laid down
by
Sardar Patel Institute of Technology
during year
2022-23.

Project Guide:
Dr. Aarti M. Karande

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Sakshi Parkar (2021510046)
Devika Patil (2021510047)

Have successfully completed the Project report on

"Case Closed App",

which is found to be satisfactory and is approved

at

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI

INTERNAL EXAMINER EXTERNAL EXAMINER

HEAD OF DEPARTMENT PRINCIPAL

Contents

Sakshi Parkar(2021510046)
Devika Patil(2021510047)

Abstract

Due to increase in criminal activities, the time taken by cops to get evidence to start the investigation process is more than expected. According to our survey, cops find analyzing the chat more helpful in the investigation process. To speed up this process chat analyzer helps to generate various statistics within fraction of seconds. The other use of this app is to detect if the suspect already has an record in the criminal area or it already exists in the database of the user irrespective of the appearance modifications (if any).

With the advancement of digital technology and the mergence of mobile phones in India, the communication scenario has completely changed. The increasing trend of smart phones and social networking applications in India has made communication faster and easier than at any time in history.

Objectives

The Android based Application "Placement App" is used

- To develop a web-app which can analyze a chat and generate various graphs.
- To detect if the suspects already exist in the database to ease the investigation process.

List of Figures

List of Tables

1 Introduction

1.1 Problem Definition

Case closed is a web-app which is a smart way into the investigation process. . This system basically saves time of the cops and speeds up the investigation process. The web-app consists of two main features, one is chat analyzer and the other one is face detector which detects if the suspect already exists in the database of the cops.

1.2 Objectives and Scope

1.2.1 Objectives

The application "Case Closed App" is

- To develop a web-app which can analyze a chat and generate various graphs.
- To detect if the suspects already exist in the database to ease the investigation process.

1.2.2 Scope

To achieve the primary goal of this application i.e. to analyze the chat and generate various statistics.
To detect if the suspects already exist in the database to ease the investigation process.
This application can be used by cops of the city.

1.3 Existing System

Currently no such kind same of application exists for the Crime Detection. Some of the disadvantages of existing system are as follows :

- Time consuming, Manual work
- Redundancy
- Loss of information

1.4 Proposed System

Case closed is a web-app which is a smart way into the investigation process. This system basically saves time of the cops and speeds up the investigation process. The web-app consists of two main features, one is chat analyzer and the other one is face detector which detects if the suspect already exists in the database of the cops.

Due to increase in criminal activities, the time taken by cops to get evidence to start the investigation process is more than expected. According to our survey, cops find analyzing the chat more helpful in the investigation process. To speed up this process chat analyzer helps to generate various statistics within fraction of seconds. The other use of this app is to detect if the suspect already has an record in the criminal area or it already exists in the database of the user irrespective of the appearance modifications (if any).

With the advancement of digital technology and the mergence of mobile phones in India, the communication scenario has completely changed. The increasing trend of smart phones and social networking applications in India has made communication faster and easier than at any time in history.

1.5 System Requirements

- Hardware Requirements

Table 1.5.1: Hardware Requirements

Processor	Dual Core Processor or Above
RAM	Minimum 4 GB RAM
Storage	Minimum 10 GB Hard Disk Space for smooth run

- Software Requirements

Table 1.5.2: Software Requirements

Operating System	OS Independent
Database	Firestore

2 Software Requirement Specification (SRS) and Design

2.1 Purpose

The purpose of this SRS Document is to present a description of project. This SRS outlines the process followed to gather the requirements for the project. This document will also describe how the requirement statements gathered from the stakeholders make their way into features of the system.
This document will, in addition, explain the scope, interfaces, and features as well as graphically describe the processes, functions and phases of the Software Development Life-cycle.

2.2 Definition

To build a Crime Detection Application so the cops can have an easy to go investigation process.

2.3 Overall Description

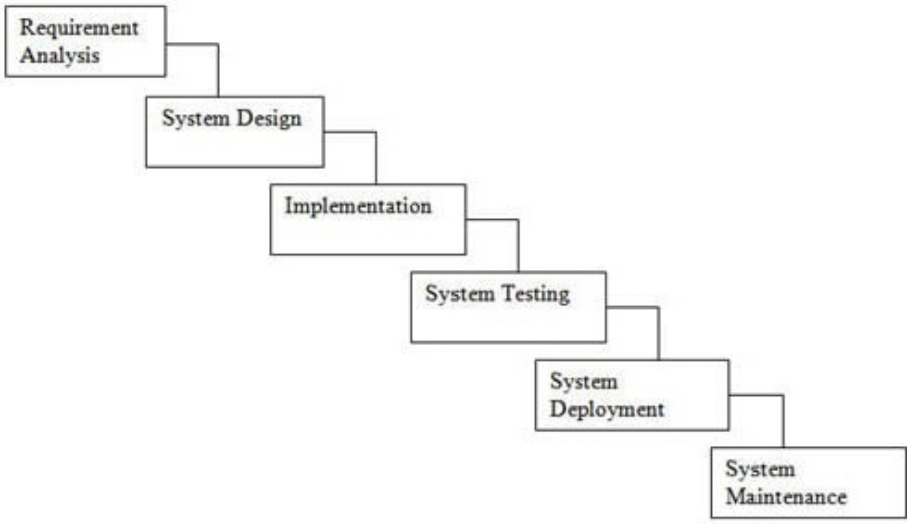
In this application user/cops have to upload the txt file of the chat they have to analyze and generate statistical analysis of. After clicking on show analysis button all the statistics will be shown on the screen. Another main feature of the app is to detect if the suspected criminal already holds any criminal history which can be useful in the investigation process. Cops have to upload the image of the suspect and if it already exists in the database then it will show the matched suspect with the respective name.

3 Project Analysis and Design

3.1 Methodologies Adapted

In Waterfall model, very less customer interaction is involved during the development of the product. Once the product is ready then only it can be demonstrated to the end users.

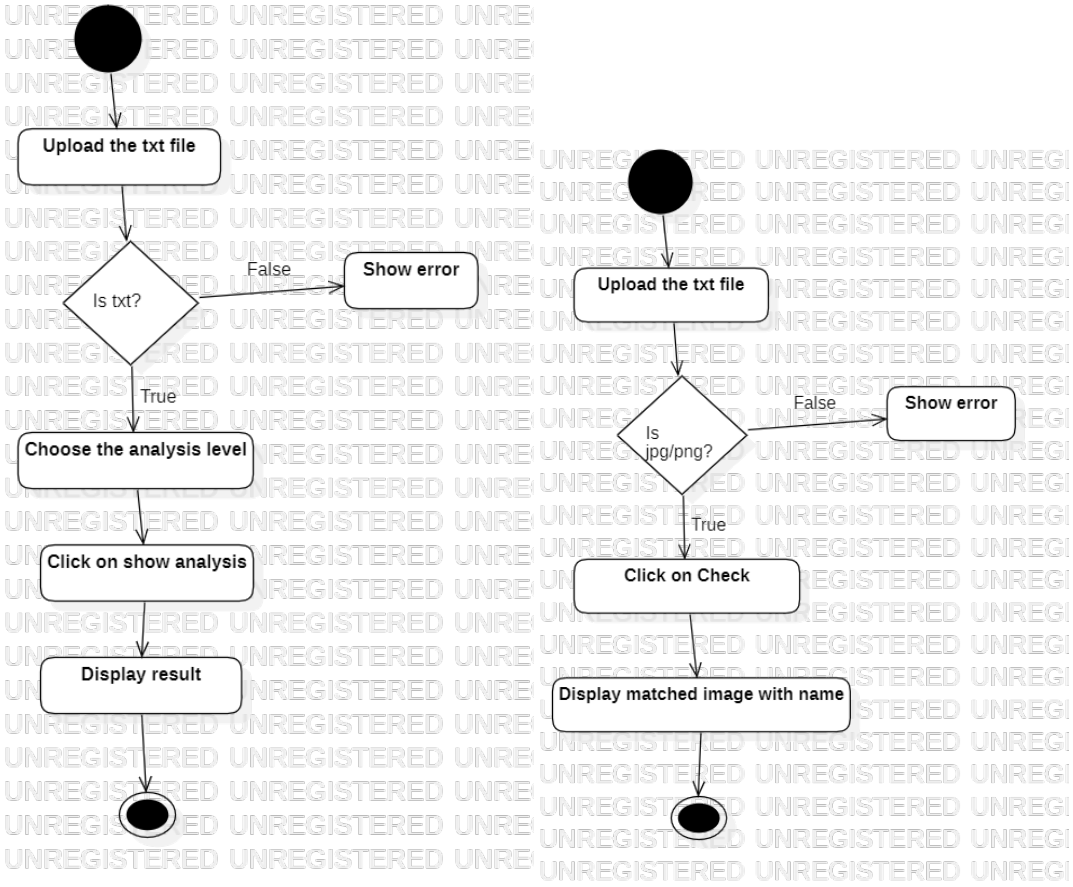
Once the product is developed and if any failure occurs then the cost of such issues is very high, because we need to update everything from document till the logic.



3.1.1: Diagrammatic Representation of Waterfall Model

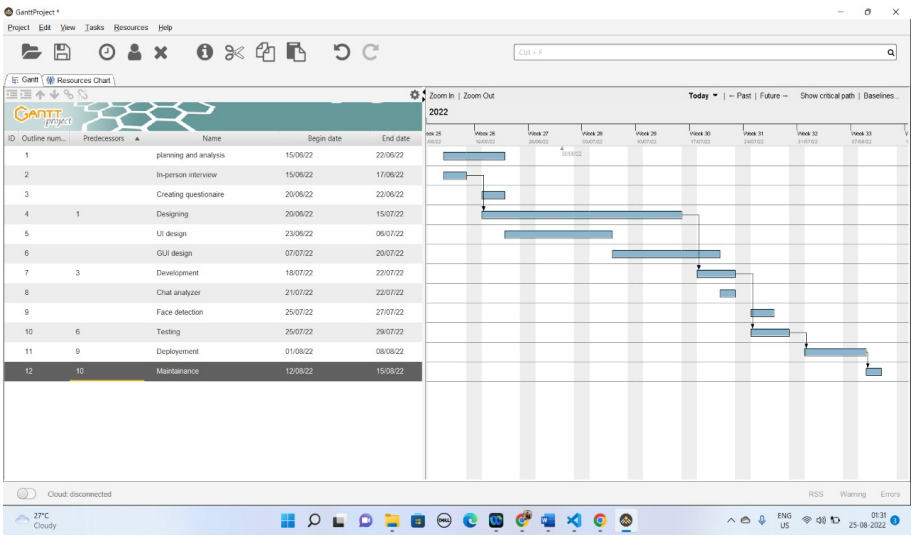
3.2 Modules

3.2.1 Activity diagram



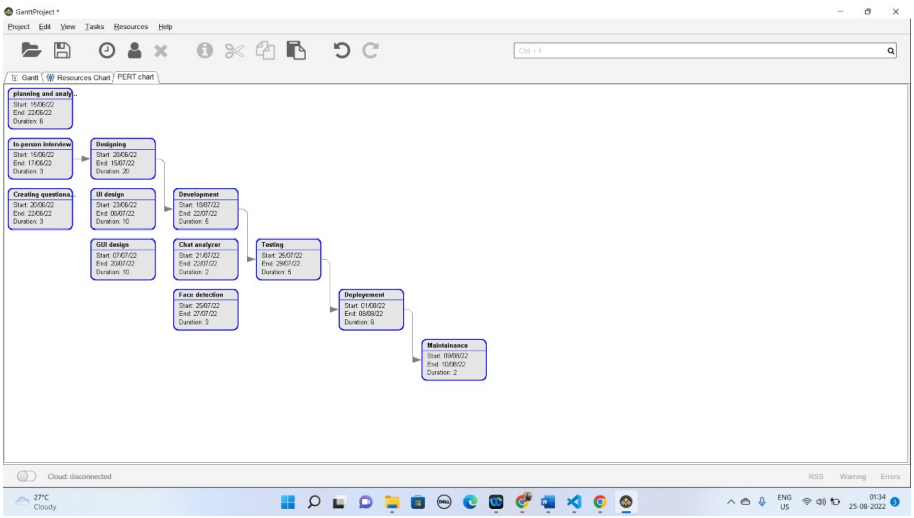
3.2.1: Activity Diagram

3.2.2 Work Breakdown Structure



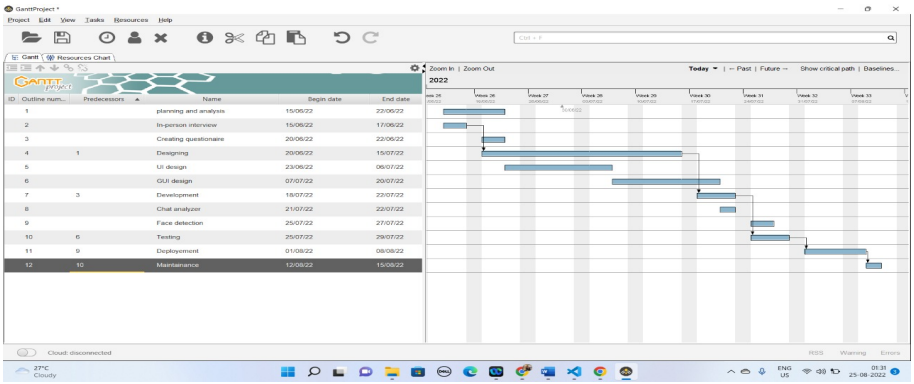
3.2.2: Work Breakdown Structure

3.2.3 PERT Chart



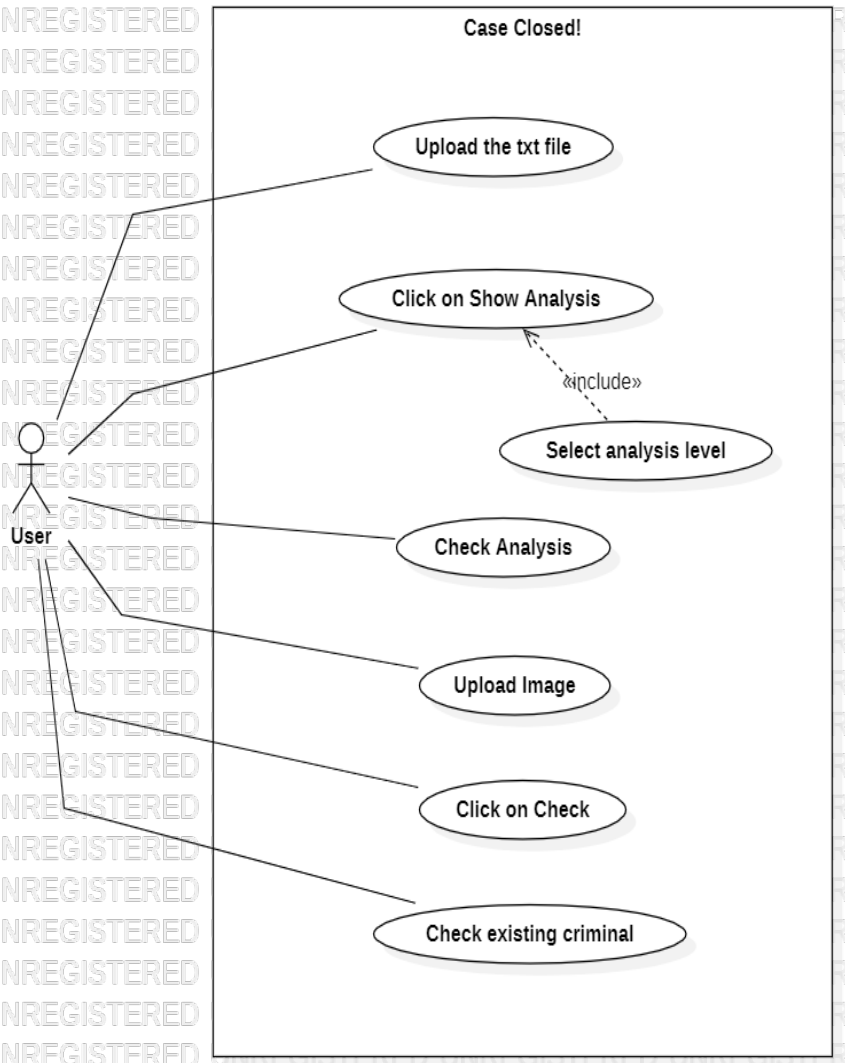
3.2.3: PERT Chart

3.2.4 Gantt Chart



3.2.4: Gantt Chart

3.2.5 Use-Case



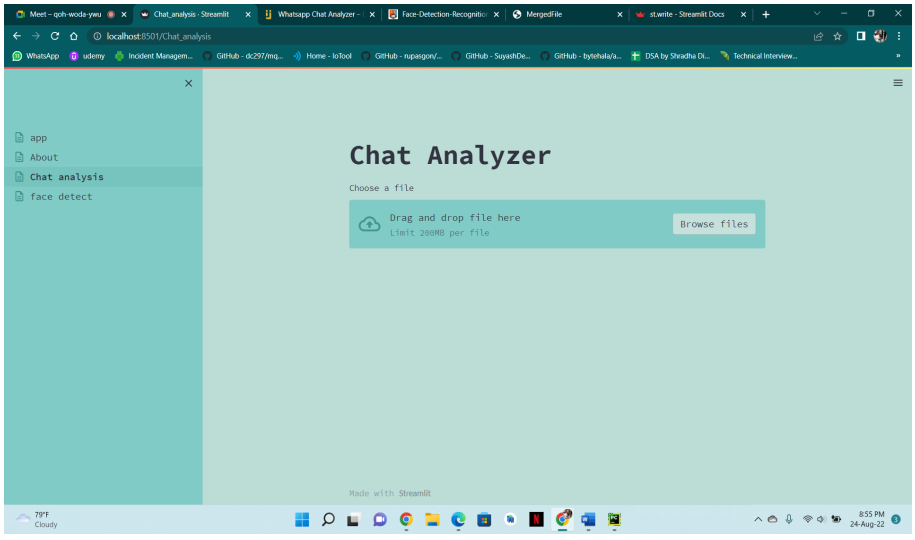
3.2.5: Use-Case Diagram

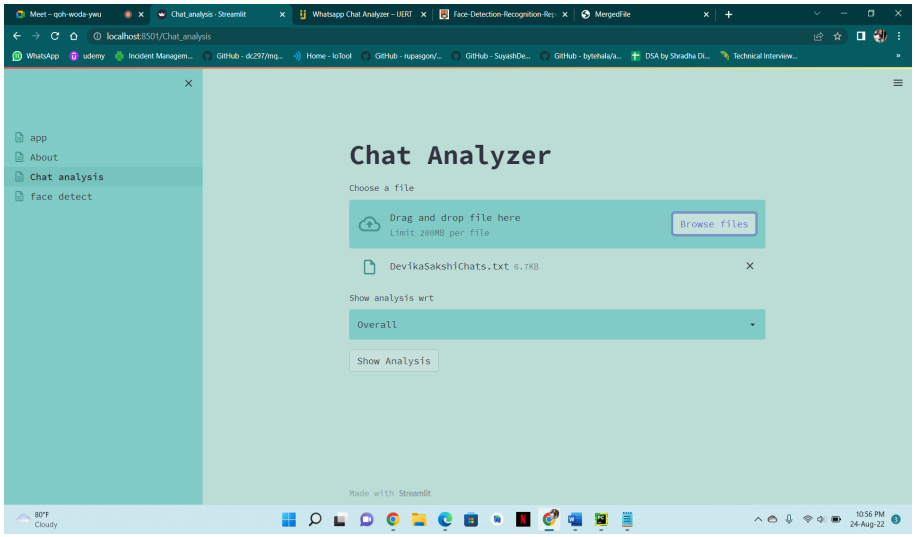
Use Cases:

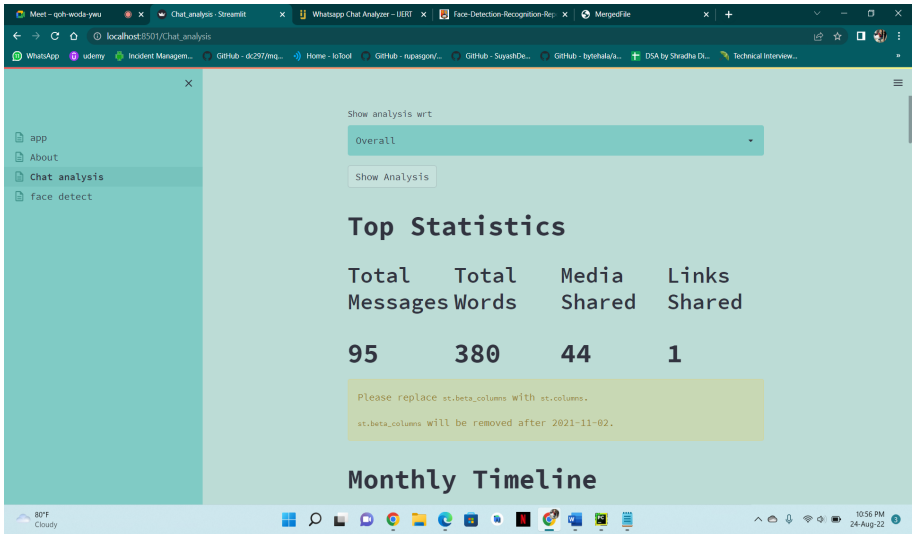
Table 4.2.1: Use Case Specifications

Actors	User
Description of Use Cases	
1	Upload the txt file- User needs to upload the txt file of the chat.
2	Click on show Analysis- User can click on show analysis button.
3	Select Analysis Levels – User can select the level at which they want the chat analysis
4	Check Analysis- Depending on the chat file, the app will result various modes of analysis.
5	Upload image- User can upload image of the suspect they want to check.
6	Click on Check- User can click on check to detect whether user already exists.
7	Check Existing Criminal- If matched, app will show the matched criminal image with the corresponding name.
Pre-condition	The file has to be of the .txt extension.
Includes	Select Analysis Levels

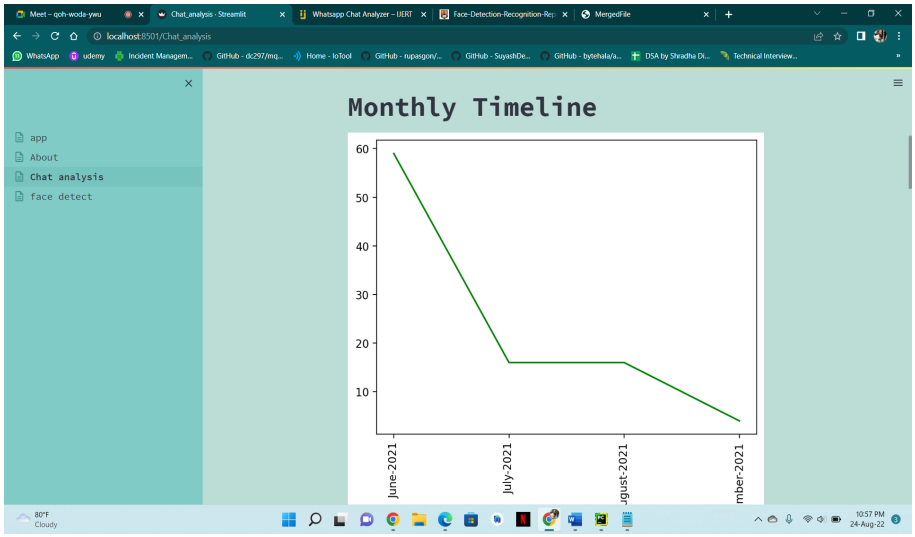
4 Project Implementation and Testing

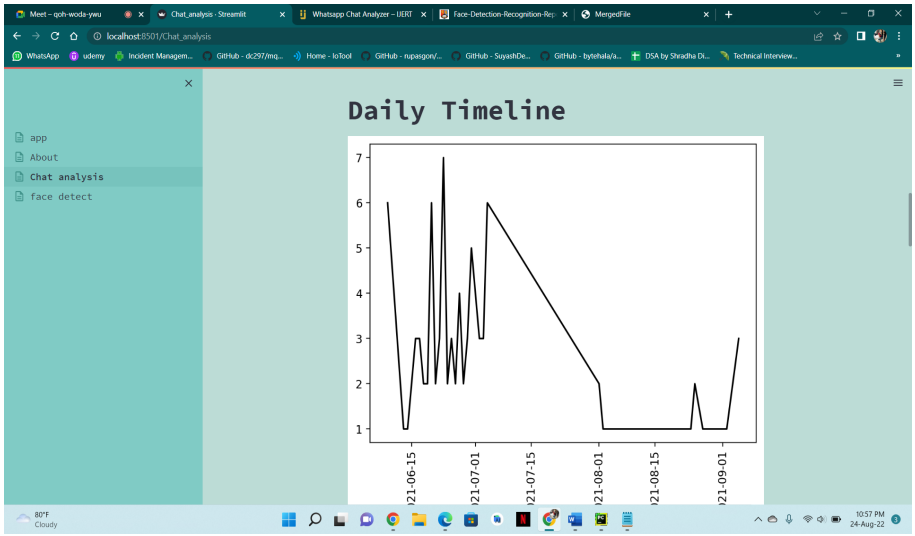


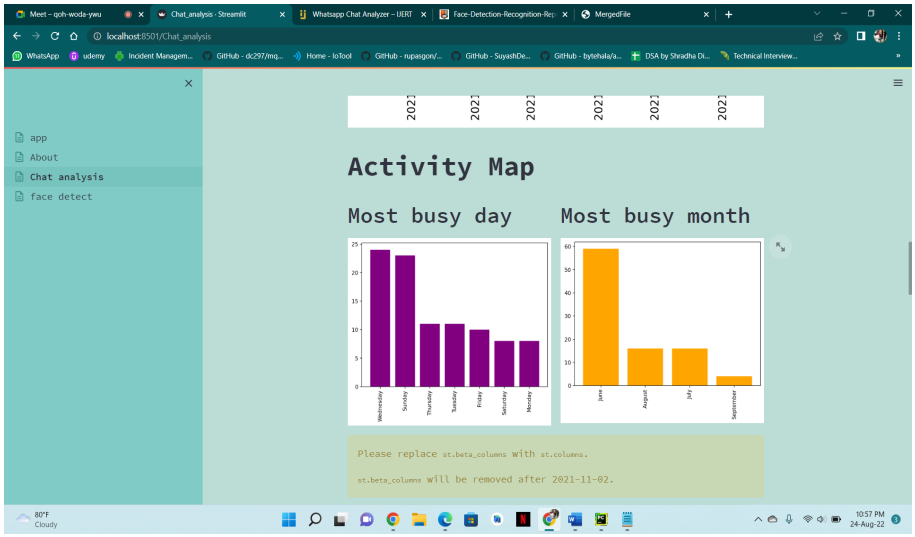

app
About
Chat analysis
face detect
Chat Analyzer
Choose a file
Drag and drop file here
Limit 200MB per file
Browse files
DevikaSakshiChats.txt 6.7KB
Show analysis wrt
Overall
Show Analysis
Made with Streamlit

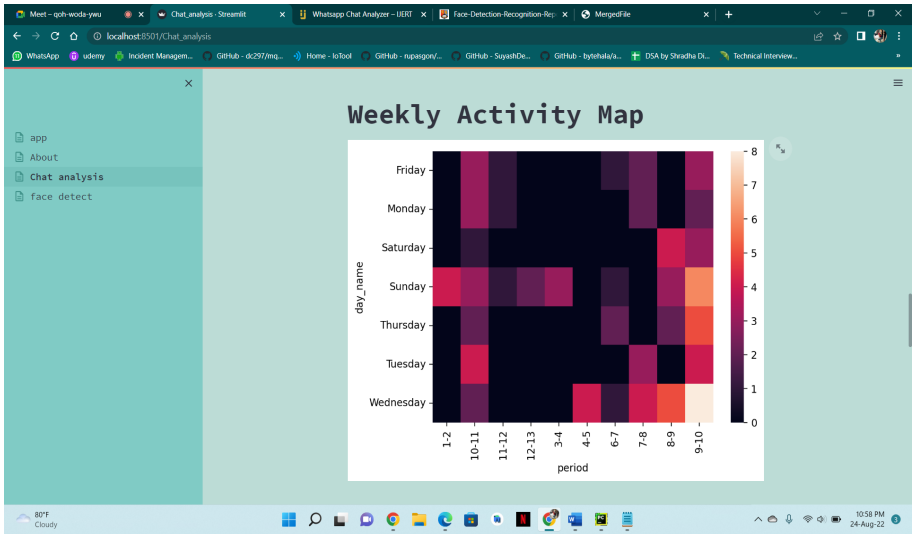

app
About
Chat analysis
face detect
Show analysis wrt
Overall
Show Analysis
Top Statistics
Total Messages
Total Words
Media Shared
Links Shared
95
380
44
1
Please replace st.beta_columns with st.columns.
st.beta_columns will be removed after 2021-11-02.
Monthly Timeline

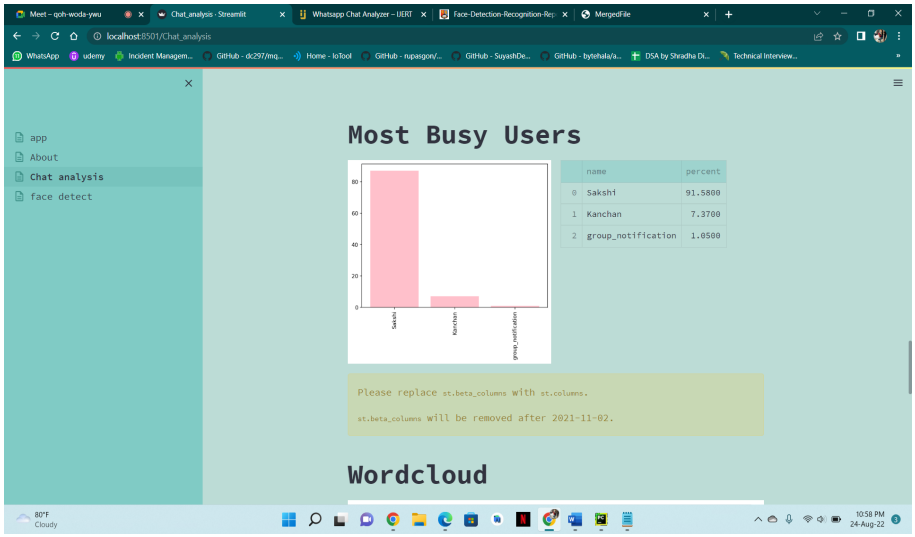
Sakshi Parkar(2021510046)
Devika Patil(2021510047)

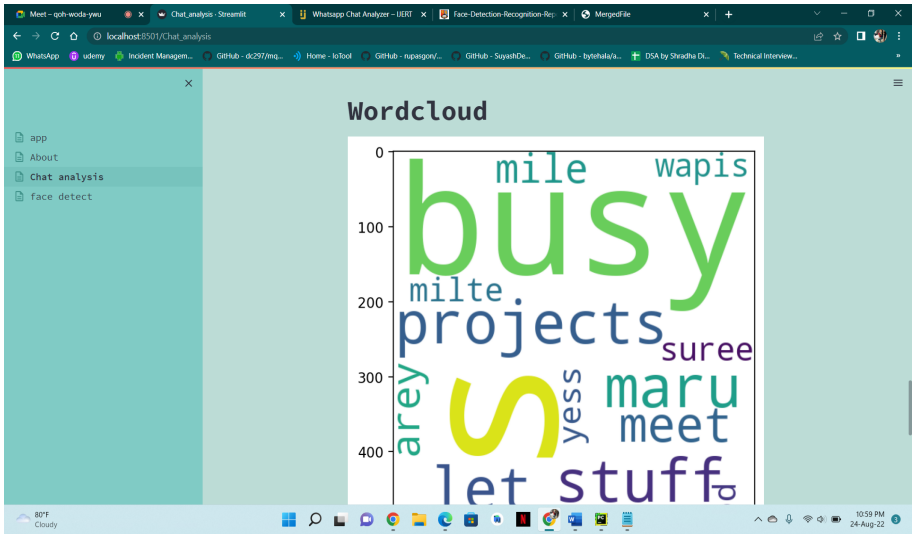


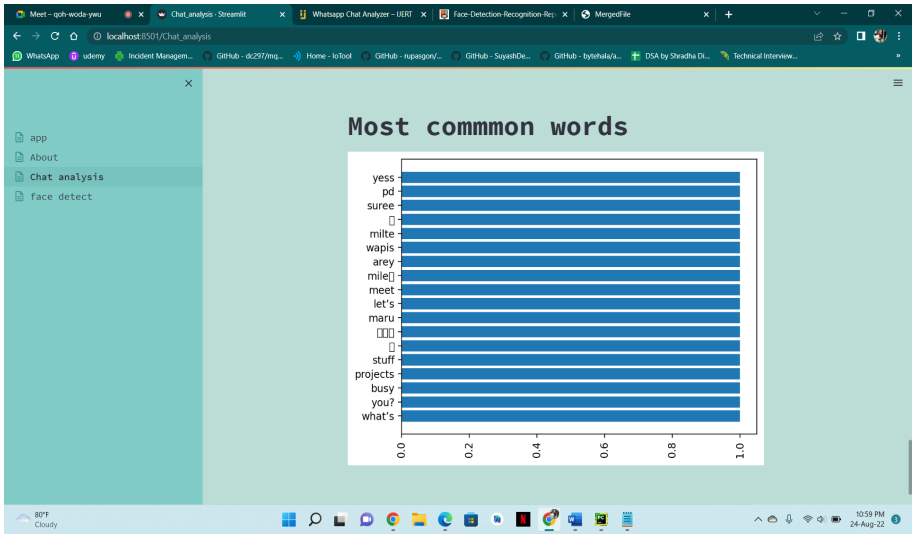


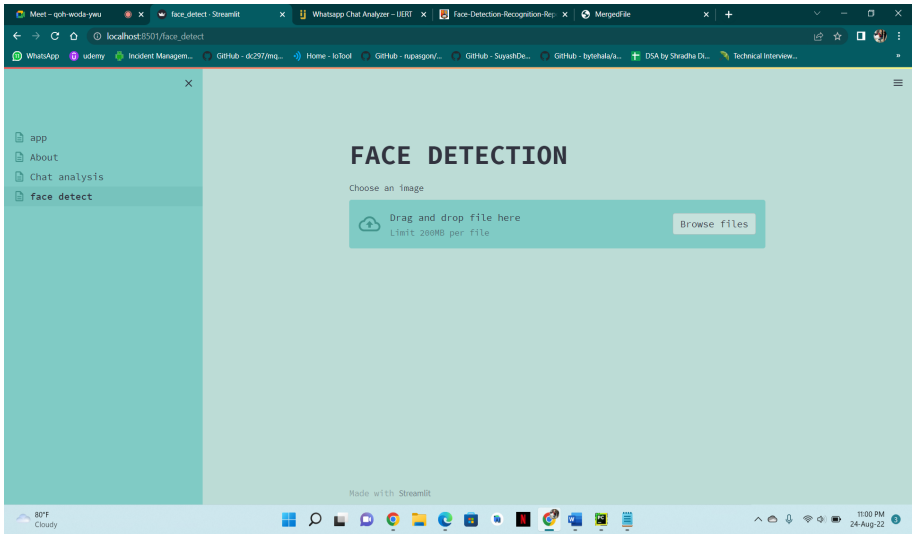

app
About
Chat analysis
face detect
Activity Map
Most busy day
Most busy month
Please replace st.beta_columns with st.columns.
st.beta_columns will be removed after 2021-11-02.

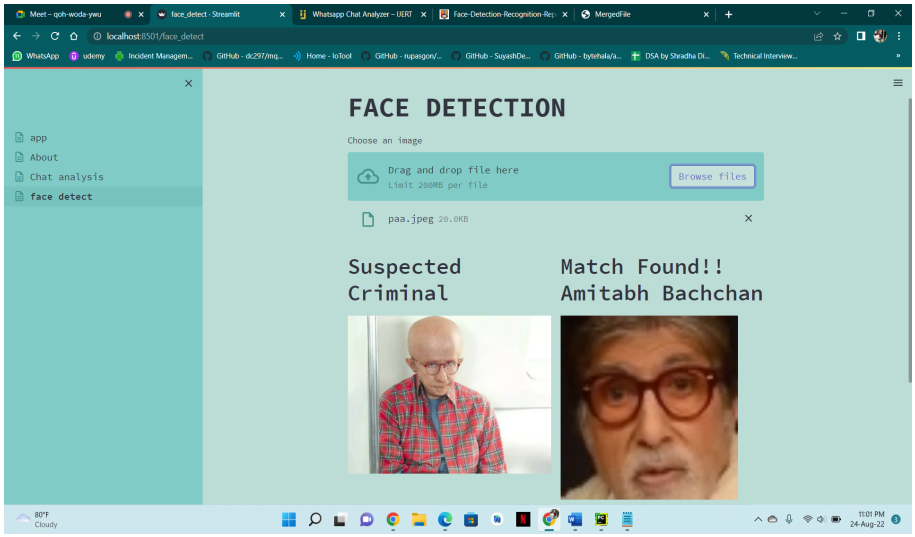

app
About
Chat analysis
face detect
Weekly Activity Map
day_name
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
period
1-2
10-11
11-12
12-13
3-4
4-5
6-7
7-8
8-9
9-10


Most Busy Users
name
percent
0 Sakshi 91.5800
1 Kanchan 7.3700
2 group_notification 1.0500
Please replace st.beta_columns with st.columns.
st.beta_columns will be removed after 2021-11-02.
Wordcloud
app
About
Chat analysis
face detect

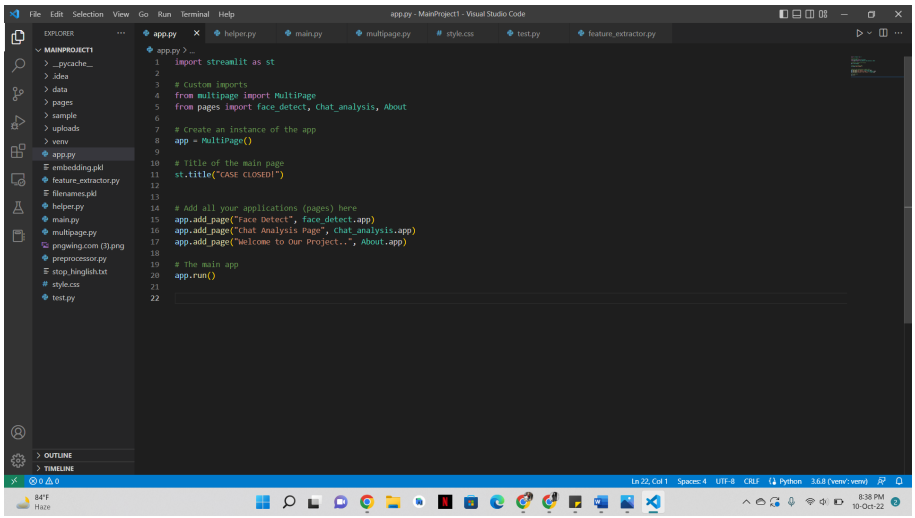

Wordcloud
mile
wapis
busy
milte
projects
suree
arey
yess
maru
meet
let
stuff
app
About
Chat analysis
face detect


Most commmon words
app
About
Chat analysis
face detect
yess
pd
suree
milte
wapis
arey
meet
let's
maru
stuff
projects
busy
you?
what's
0.0
0.2
0.4
0.6
0.8
1.0


FACE DETECTION
Choose an image
Drag and drop file here
Limit 200MB per file
Browse files
app
About
Chat analysis
face detect
Made with Streamlit


FACE DETECTION
Choose an image
Drag and drop file here
Limit 200MB per file
Browse files
paa.jpeg 20.0KB
app
About
Chat analysis
face detect
Suspected Criminal
Match Found!!
Amitabh Bachchan

4.1 Code 1



4.2 Code 2

multipage.py > ...

```python
import streamlit as st


# Define the multipage class to manage the multiple apps in our program
class MultiPage:
    """Framework for combining multiple streamlit applications."""

    def __init__(self) -> None:
        """Constructor class to generate a list which will store all our applications as an instance variable."""
        self.pages = []

    def add_page(self, title, func) -> None:
        """Class Method to Add pages to the project
        Args:
            title ([str]): The title of page which we are adding to the list of apps

            func: Python function to render this page in Streamlit
        """

        self.pages.append({

            "title": title,
            "function": func
        })

    def run(self):
        # Drodown to select the page to run
        page = st.sidebar.selectbox(
            'App Navigation',
            self.pages,
            format_func=lambda page: page['title']
        )

        # run the app function
        page['function']()
```

4.3 Code 3

```python
from tensorflow.keras.preprocessing import image
from keras_vggface.utils import preprocess_input
from keras_vggface.vggface import VGGFace
import numpy as np
import pickle
from tqdm import tqdm

filenames = pickle.load(open('filenames.pkl','rb'))

model = VGGFace(model='resnet50',include_top=False,input_shape=(224,224,3),pooling='avg')

def feature_extractor(img_path,model):
    img = image.load_img(img_path,target_size=(224,224))
    img_array = image.img_to_array(img)
    expanded_img = np.expand_dims(img_array,axis=0)
    preprocessed_img = preprocess_input(expanded_img)

    result = model.predict(preprocessed_img).flatten()

    return result

features = []

for file in tqdm(filenames):
    features.append(feature_extractor(file,model))

pickle.dump(features,open('embedding.pkl','wb'))
```

4.4 Code 4

```python
test.py > ...
import pickle
from sklearn.metrics.pairwise import cosine_similarity
import cv2
from mtcnn import MTCNN
from PIL import Image


feature_list = np.array(pickle.load(open('embedding.pkl','rb')))
filenames = pickle.load(open('filenames.pkl','rb'))

model = VGGFace(model='resnet50',include_top=False,input_shape=(224,224,3),pooling='avg')

detector = MTCNN()

sample_img = cv2.imread('sample/paa.jpeg')
results = detector.detect_faces(sample_img)

x,y,width,height = results[0]['box']

face = sample_img[y:y+height,x:x+width]


image = Image.fromarray(face)
image = image.resize((224,224))

face_array = np.asarray(image)

face_array = face_array.astype('float32')

expanded_img = np.expand_dims(face_array,axis=0)
preprocessed_img = preprocess_input(expanded_img)
result = model.predict(preprocessed_img).flatten()


similarity = []
for i in range(len(feature_list)):
    similarity.append(cosine_similarity(result.reshape(1,-1),feature_list[i].reshape(1,-1))[0][0])
```

4.5 Code 5

helper.py > ...

```python
from urlextract import URLExtract
from wordcloud import WordCloud
import pandas as pd

from collections import Counter
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())

    num_media_messages = df[df['message'] == '<Media omitted>\n'].shape[0]

    links = []
    for message in df['message']:
        links.extend(extract.find_urls(message))
                                    (variable) num_media_messages: Any
    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')
```

4.6 Code 6

helper.py > ...

```python
    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']:
```

4.7 Test Case 1

5.6 Test Cases

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	1.If Internet connection is on	Show chat analysis	Chat analysis will be shown.	Chat analysis	Pass
2	2.If Internet Connection is off	Don't show chat analysis	Chat analysis will not be shown.	Error	Pass
3	If Internet Connection is on	Alert warning	Alert warning won't popup	Alert message is not sent.	Pass
4	If Internet Connection is off	Alert warning	Alert warning will popup.	Alert warning pops up.	Pass

(For internet connection)

4.8 Test Case 2

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	If txt chat file is uploaded	Show chat analysis	Chat analysis will be shown.	Chat analysis	Pass
2	If any other extension file is uploaded	Don't show chat analysis	Chat analysis will not be shown.	Error	Pass

(For txt extension chat files)

4.9 Test Case 3

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	If uploaded picture matches with the database picture	Show matched result with the respective name.	Matched result with the name will be shown.	Matched result is shown.	Pass
2	2. If uploaded picture does not match with the picture	Don't Show matched result with the respective name.	Matched result with the name will not be shown.	Matched result is not shown	Pass
3	false recognition may occur as machine learning has not 100% accuracy.	Matching the pictures from both the end	Recognition might not happen.	Pictures are not recognized/matched	Pass

(Face detection)

5 Limitations

- All the users should have active internet connection.
- The supportive web browser is needed.
- Chat analyzer only supports .txt files

6 Future Enhancements

- Option to find out more information about the suspect other than the name.
- To make the face recognition more accurate.
- To analyze and give statistics about the media shared.

7 Bibliography

7.1 Web References

[1.] https://pypi.org/project/opencv-python/
[2.] https://mumbai.citizenmatters.in/accountability-how-to-raise-complaints-with-the-bmc2248
[3.] www.wikipedia.com
[4.] https://stackoverflow.com/